Exploring Pitch Perception Thresholds in Very Short Tones: Minimum Tone Duration for Perception of Pitch in 2-Tone and 3-Tone Sequences.

Russell Burton, BEd, GradDipMusEd, MMus.

School of Psychology

December 2021

A thesis submitted in fulfilment of the requirement for the degree of Doctor of Philosophy.

THE UNIVERSITY of ADELAIDE

Blank page

Table of Contents

List of Tables

List of Figures

Abstract

To teach students the elements of music, comprehensive understanding of each element is essential. The thesis aimed to improve understanding of pitch perception in short contiguous tone sequences.

A study with 20 first-year psychology students established that the ability to order tones does not mean their pitch can be identified. With timbre varied between stimuli, longer durations were required to identify the order of 2-tone sequences, compared to when timbre was the same for all stimuli ($p < .01$, effect size = 0.72). It was concluded that identification of the shortest stimuli relied on sound analogue comparison (holistic representations of sound sequences), not pitch perception.

A method to establish the Minimum Duration for Perception of Pitch (MDPP) in short contiguous tone sequences was created using target and test tones of different timbre and duration. Tone durations under 40ms were asynchronously modified by frequency level with lower frequencies retaining longer durations. Using a 3-forced-choice paradigm, four experienced musicians (18-58 years; 3 males) matched short sine wave target tones against long piano test tones as either the same pitch, or a semitone higher or lower. Mean overall MDPP was 8ms for single tones, 24ms for 2-tone sequences and 47ms for 3-tone sequences. Results for single tones matched those from previous research. Minimum durations varied extensively, depending on each tone's position in 2-tone and 3-tone sequences. Longer minimum durations were recorded to identify low compared to high tones in the first position of 2-tone sequence (13ms difference, 95% CI [1ms, 25ms], $p = .043$) and in both the first position (54ms difference, 95% CI [2ms, 104ms], $p = .046$.) and the second position (53ms difference, 95% CI [19ms, 88ms], $p = .014$) of 3-tone sequences. It was suggested that differences were caused by (i) preconscious adjustment of low tones to accommodate subsequent high tones as harmonics of the lower tones, causing identification errors in low tones; and (ii) stream segregation of higher tones, enabling easier identification of high tones. Tones positioned last were unaffected by the relative pitch of previous tones. Further work will be required to establish the duration at which the number of segments in a sequence of tones no longer affects the MDPP by increasing the number of segments until the average minimum duration of tone segments reaches asymptote.

Results were used to select durational limits for an aural training program to teach students musical intervals. The program used temporal occlusion (the reduction of tone durations to improve focus and concentration) to improve interval recognition skills for the first four intervals of a major scale (tone durations 1 second--20ms). Participants were 84 Year 8 students (aged 12-14, music experience 2-5 years) The study compared the new program to the school's current aural training program; both were computer-based, self-explanatory, and modified training activities based on success. The new method was not significantly better than the current program. Suggested improvements included creating targeted videos for explaining tasks, improving "game like" aspects and providing opportunities for person-to-person interaction.

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree. I give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of funding from the Catholic Education Office.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Russell Burton

Signed:

1st December 2021

Acknowledgements

I would like to express my most sincere gratitude to my chief supervisor Emeritus Professor Ted Nettelbeck for his commitment to getting me through this eight-year process, for all of the help and guidance he has given me over those years and also for inspiring me to keep going when I needed it. I would also like to thank my supervisors Dr Ian Zajak and Associate Professor Carl Crossin OAM for their help and guidance. I would like to thank my wife Vera for the support she has given me over the years and my three children Ivan, Greg and Tim for their patience and support. I would like to thank the teaching staff and students at Cabra Dominican College for their assistance and support with the research studies that I undertook at the school with their help and participation. I would also like to thank the members of the Australian Society of Music Education and the student body at the University of Adelaide who took part in my studies.

Preface

When I first started this research, my goal was to isolate a single, indivisible unit of sound from which all music was constructed. This search was inspired by the Australian Curriculum, Assessment and Reporting Authority's new definition of 'the elements of music' as: "Rhythm, pitch, dynamics and expression, form and structure, timbre, and texture" (ACARA, 2015). After comparing ACARA's list of elements with the American education system's list of elements (NAfME, 2015) and the English education system's list (www.gov.uk, 2013), it was clear that there was no agreed list of elements of music. Many elements were common amongst at least two of the lists, but all lists differed from each other in some respects. I wanted my students to be able to access an authoritative, universal list of the elements of music that they could reliably use for reference in their education. I decided the best way to do this was to identify and isolate single, indivisible, units of sound, which could then be considered the 'building blocks' from which music was created. From these building blocks, the true 'elements of music' could be extracted.

I had theorised that a very short segment of sound, perhaps 1/50 of a second in duration (20ms), would be cognitively indivisible and that all the elements of sound - "pitch, loudness, duration, timbre, sonic texture and spatial location" (Burton, 2015)[1] would be perceptible within this unit. My thinking was that isolating basic building blocks of music (which I had imagined as chunks or units of sound) would lead to improved methods for teaching and understanding music. Additionally, should these basic building blocks also

[1] This paper was prepared and presented during my first 18 months of candidature. It is not directly relevant to the final direction of this thesis but has been included here as Appendix 1.

apply to language and speech, this might lead to improved methods for teaching and understanding language and speech as well.

However, during the early phase of reviewing past literature that had addressed duration thresholds for the perception of sound, I found evidence that what I had envisaged was not possible. Different aspects of sound perception required different minimum durations for perception, which meant that there was no single time span within which all elements of sound were both perceptible and non-divisible. Researchers had already found that it was possible to identify accurately the arrival times of sounds at each ear for interaural time differences as short as 10 microseconds (i.e. 1/100,000 of a second), as summarised and confirmed by Thavam & Dietz, (2019). I also found evidence that changes in frequency and loudness can be discerned at durations as short as 1 ms per sound segment (Wier & Green, 1975) and that differences in timbre could be identified in sounds of even shorter durations (Robinson & Patterson, 1995). On the other hand, much longer durations were required for accurate perception of pitch (Doughty & Garner, 1947). Moreover, it was obvious from these studies that substantial individual differences existed around any mean value established. Taken together, these results clearly demonstrated that sound cannot be divided into units of a duration (or duration range) from which no further division would be discernible and within which all elements of sound perception would be both available and stable.

Nonetheless, reviews of brain wave studies revealed that a form of sound ‘chunking’ was indeed apparent in sound cognition and that some aspects of sound, such as pitch perception, can be treated as chunks or units (See Chapter 2, “Discussion” section relating to beta and theta brain waves for an extended account of this realisation). I therefore changed the direction of my study, from attempting to identify units of sound, to identifying units of one aspect of sound; perception of pitch. Researchers had already established that the minimum duration for perception of pitch (MDPP) is not constant but is dependent on a variety of factors, including the frequency of the sound, the complexity of the sound and the context in which the sound is heard. However, little work has been done on the perception of pitch in short contiguous sequences, which became the focus of my research.

Why focus on such a small area of music such as identifying pitch in short contiguous sequences? When teaching music, the inconsistency of important musical terms, such as ‘pitch’, makes it very hard to provide students with an accurate account of the effect music has and how it is achieved. The reason for the inconsistency in relation to pitch, is that how pitch perception works is not fully understood. For example, one would expect that if a sound with an identifiable pitch was played, all the students in the room would hear the sound as the same pitch and be able to match it on a keyboard. This has been found to be not so; the same sound (in some instances) can be identified as different pitches by different people (Renken, 2004). Explaining this to students requires a knowledge of pitch perception that most teachers cannot provide. Another area of pitch perception that is still not fully understood is that of perception of pitch in short contiguous sequences, and this is the area of focus for this thesis. I believe that the research provided in this thesis will assist in our understanding of the limitations of pitch perception and provide insights into what happens when sounds change so quickly that the pitch of each change is no longer discernible. These insights are not designed for the education of beginner music students, rather they are for advanced students with a desire to understand and explore every aspect of music.

Additionally, to help improve music education for struggling students, as part of this thesis I created an aural training program. For many music students, especially those who are less skilled than others, aural training is a boring, repetitive, unrewarding, and seemingly pointless task. I created a computer program that used very short tones to improve learning efficiency, with the goal of making interval training more interesting for students while making progress more incremental, thus allowing slower learners more time to adapt to new tasks. I tested it with year 8 students to see if they achieved better results using it than another computer based aural training program. Although not convincingly successful, some aspects of this attempt were promising. The results of this work are included in Chapter 6.

Chapter 1: Introduction

This thesis combines a reflexive ethnographic approach with empirical methodology. Reflexive ethnographic research has been shown to be a successful method for improving music education outcomes in a variety of contexts (Campbell, 1999; Partti & Karlsen, 2010; Williams, 2005). Personal insights into music pedagogy are explored using a reflexive interrogation of historical and current research combined with exploratory empirical studies. The focus of this thesis was to examine pitch perception thresholds in short sounds and how they can be used to improve educational outcomes, as well as to increase understanding of sound and its relationship to music. To do this, known sound ordering thresholds were examined for relevance to pitch perception. Because it was highly unlikely that a single minimum duration would be appropriate for explaining variability of perception limits, known causes of variability in perception results were examined. These include differences in information masking levels of tones in different positions (first, second or third) and relative pitches (highest, middle or lowest), as well as variability in individual results. The minimum duration for perception of pitch (MDPP) in 2-tone and 3-tone sequences was established by experimental means.

The benefits of increasing our understanding of pitch perception by identifying the MDPP (minimum duration for perception of pitch) in short contiguous sequences include enabling teachers to more accurately explain how pitch perception works, and resolving anomalies between pitch perception, frequency change detection, and timbre perception. This exploration could also lead to improvements in hearing implants, hearing implant training, computerised speech recognition and computerised speech production, as well as enhance our understanding of masking and how the brain processes sound. Further to this, this research should lead to an improved understanding of how masking levels change depending upon the relative position of tones in a sequence.

Additionally, to optimise training outcomes, strategies based on short tone identification should be developed that both a) place higher demands on the nervous system that handles pitch perception than is required under normal listening conditions and b) requires effortful deployment of auditory attention (Patel, 2012). To this end, minimum duration levels for perception of pitch were used to create a computer-based aural training program. The aim of this program was to help struggling students learn to recognise musical intervals, which is an exercise that forms part of the students' educational curriculum. The intention was to gradually decrease durations of the two tones comprising each interval to ensure increased attention is given to the increasingly limited information provided, with the additional benefit of maintaining a modest challenge to ensure appropriate motivation. It was reasoned that this would provide increased exposure to a small range of simple intervals to maximise retention, without repetition becoming boring.

Outline of Prior Short Tone Research

When two or more tones are played contiguously (i.e. one straight after the other), distinct pitches are perceived if the duration of each tone is sufficient to allow the cognitive separation of the tones. However, if the tones are very short, smearing of the tones prevents or distorts the perception of the pitch of the tones (Ciocca & Darwin, 1999; Grose, Hall, & Buss, 2002; Hawkins, Thomas, Presson, Cozic, & Brookmire, 1974).

The minimum duration for establishing a sense of pitch from a single isolated tone has been found to range from 4ms to 16ms, depending upon the frequency of the tone being heard (Doughty & Garner, 1947). Low tones have required longer durations for presentation than high tones. For isolated tones presented in the context of a melody, minimum durations for identification of pitch increased to between 8 ms and 34 ms, depending upon the frequency of the tone being heard (Patterson et al., 1983; Robinson & Patterson, 1995).

Expressed in terms of wave cycles, it takes around eight wave cycles of a sine wave tone to establish the pitch of an isolated tone, to the point that it can accurately be identified in a melody. (Patterson et al., 1983). This figure equates to around 8 milliseconds (ms) for higher frequency (1000 Hz) tones, lengthening to around 34 ms for lower frequency (200 hz) tones.

All research referred to so far relates to isolated tones, where tone segments are not immediately preceded and/or followed by other sounds. In the context of tones that form part of a sequence that does not include gaps between tones (i.e. contiguous sequences), little research has been done that requires accurate identification of the pitch of those short tones used in the studies. However, some research has implied that the identification of pitch was likely used in the identification of stimuli. Thus, for example, Warren, Gardner, Brubaker, and Bashford Jr (1991) found that identifying short well-known melodies played in a continuous cycle required durations of 160ms per tone. It can be assumed that the pitch of the tones must have been perceivable for the melodies presented to be correctly identified. In addition to this research, many masking studies have used sine wave tones that started either contiguously or 20ms apart (Pierre L Divenyi & Hirsh, 1975; Leshowitz & Cudahy, 1973; Massaro, 1970; Massaro, 1971, 1973; Massaro, 1975; Massaro, Cohen, & Idson, 1976; Massaro & Idson, 1978). In most of the studies participants were asked to select a tone as either ‘high’ or ‘low’ and in some studies the interval between the two tones was less than a semitone. Although these studies were not designed to identify the minimum duration for perception of pitch (MDPP), they are useful for examining why minimum durations for identification change depending upon the position of the tone in the sequence. As will be explained later, the methods used in these studies are generally not suitable for detecting minimum durations for identification of pitch because there are a variety of strategies other than pitch perception that can be used to identify tones in these studies.

In contrast, minimum durations for *ordering* sequences of tones has been studied extensively. In these studies, results have varied dramatically depending on the study conditions. The order of presentation of sequences consisting of two contiguous sine wave tones can be identified at durations as short as 1ms per segment (Wier & Green, 1975). The order of presentation of sequences consisting of three contiguous sine wave tones can be identified at durations as short as 2 ms per segment (Divenyi & Hirsh, 1974; Nickerson & Freeman, 1974). As will be noted later in this thesis, these durations are shorter than the MDPP for a single tone. They therefore cannot be the result of pitch perception (see Chapter 3 for a study dealing with this issue). There seems to be no research relating to short isolated sequences longer than three tones.

Minimum duration results for identifying tone order in cyclic sequences varied enormously depending on the type of sounds being presented to participants. Minimum durations ranged from 50ms for tones in a simple sequential order (Schouten, 1962) to 670ms for ordering the sequence of tones in disparate ranges and timbres (Warren & Obusek, 1972). These studies did not necessarily require participants to accurately identify the pitch of tones being ordered, although it is highly likely that accurate pitch identification was possible for many of the stimuli being used in these studies. Evidence from studies of brain wave functioning, combined with information about stream segregation, provides reasons for these diverse results.

The theta wave is responsible for moving sound information to the auditory cortex, as well as to short-term memory, and lasts between 125 and 250ms (Galambos, Makeig, & Talmachoff, 1981; Luo & Poeppel, 2012; Zanto, Large, Fuchs, & Kelso, 2005). This range matches closely the range of minimum durations found for ordering similar sounds (125ms – 200ms). Thus, unless sound seqeunces form a very simple pattern such as a scale (Schouten

1962), sound sequence segments require the duration of a theta wave to be successfully ordered. However, dissimilar sounds have been found to require a duration of more than that of a single theta wave for ordering. This is because the disparate nature of these sounds requires a shift in attention amongst different classes of stimuli (Thomas & Fitzgibbons, 1971; van Noorden, 1975). This shift takes time and ordering dissimilar sounds has been found to require up to 670ms (Warren & Obusek, 1972), which equates to around three theta waves.

The gamma wave is closely linked to local processing and extracting information from afferent nerves (Gu & Liljenstrom, 2007; Luo & Poeppel, 2012; Steinschneider et al., 2011; VanRullen et al., 2014). The gamma wave resets (i.e. drops previous signal information) as soon as a change in a sound is noticed and immediately starts processing the new information. When this happens frequently, processing of individual sounds slows to the rate of beta waves. Beta waves emerge as subharmonics of gamma waves when quick changes occur regulary. These waves last around 50ms (Haenschel et al., 2000; Kisley & Cornwell, 2006). The duration barrier descibed by Schouten and others, where the sounds of tones change from discrete sounds to blurred sounds, matches the duration of a beta wave. Tones in a continuous sequence shorter than a beta wave blur together, as more than one sound is processed within the same beta wave. The result of this blurring is usually perceived as a slide in pitch rather than a discrete change.

There were two areas of interest that emerged from examining short-tone research. Firstly, except for single tone studies, there were no studies found that required participants explicitly to identify the pitch of each tone. Secondly, a proposition realised as a consequence of this observation was that the MDPP for tones in short contiguous sequences has not been established; participants in the 2-tone and 3-tone studies referred to above could not have been using pitch perception to differentiate between tone sequences because the minimum durations achieved were shorter than the MDPP found for a single tone. Therefore, I located my research in this area.

Examining Short Tone Sequences

Although an extensive literature review was undertaken, no research was found that specifically delineates pitch identification in contiguous tone sequences, other than a study by Warren et al. (1991) (see above), where participants were required to identify familiar melodies repeated in a continuous loop (minimum duration was 160ms). As will be seen in the literature review below, sound ordering research has been covered comprehensively. The research into short sequential tones was mainly carried out by researchers focused on language perception and was designed to identify the minimum duration required for the ordering of sounds, rather than to identify pitch. However, from an examination of tone ordering research results, it was concluded that it is not possible that the minimum duration for perception of pitch is the same as the minimum duration for ordering of tones. This is because the order of presentation of sequences consisting of three contiguous sine wave tones can be identified at durations as short as 2 ms per segment (Divenyi & Hirsh, 1974; Nickerson & Freeman, 1974). However, this duration is far shorter than the 8ms minimum duration required to identify the pitch of a single isolated tone (Patterson et al., 1983; Robinson & Patterson, 1995). It stands to reason that a tone in a sequence of three cannot be identified at a shorter duration than that required to identify a single tone and therefore the MDPP in short contiguous sequences has not yet been established as part of sound ordering research. There does not seem to have been any researchers (past or present) interested in identifying the pitch of sequential tones, other than to ensure that sound ordering results were accurate.

The lack of research into short tone pitch perception begs the question; how relevant or important is it to discover the MDPP in short contiguous tone sequences? In addition to this, is it important to discover if minimum durations for perception of pitch are consistent for tones in different positions in a sequence and of different relative pitch heights? It is suggested here that there are several areas of current research interest that might benefit from this knowledge. Improved knowledge of short-tone pitch perception would surely provide insights into how the brain processes sound and thereby improve our knowledge of brain functioning. Such knowledge could also expand our understanding of information masking and help to identify more accurately which sounds are masked by other sounds and which are not, as well as understanding why this masking might occur. As described in detail in Chapter 5, the amount of interference caused by tones other than the target in a 2-tone or 3-tone sequence changes depending upon the pitch height of the two tones relative to each other, as well as on their position in the sequence.

It is also envisaged that improved knowledge of the perceptual limits of pitch identification has potential to improve teaching methods for developing pitch identification and pitch differentiation skills in students, especially struggling students who find aural training unmotivating. This might be achieved by using tones of decreasing duration to an identified perceptual limit that is outlined by these studies.

Finally, it is possible that understanding differences in minimum durations for perception of pitch (and the reasons for these differences) may eventually lead to improvements in areas such as computer-generated speech synthesis, voice recognition, sound compression and sound processing for hearing aids. Such knowledge may be of assistance in the future, when finer details, like conveying emotional content through pitch modulation and providing details relating to information masking in sound compression, become important, rather than the current focus on energetic masking (Allen, 2008; Hajiaghababa, Marateb, & Kermani, 2018). Regarding computer-generated music, by understanding how perception changes depending on the position of a tone in a sequence, melodies might be created that are just on the edge between clarity and blurring, which may be a desired effect for some composers. The potential relevance for this research will be discussed further in the Conclusions section (Chapter 7).

Objectives

1. To confirm that the minimum duration for ordering tone sequences is less than the minimum duration for perception of pitch.
2. To establish the minimum duration for perception of pitch (MDPP) in 2-tone and 3-tone sequences.
3. To investigate the effect of different tone positions on the MDPP.
4. To use the information gained from this study to improve computer-based aural training methods and to create a new aural training system.

Research Questions,

1. Does the ability to order a sequence of tones mean that the pitch of the tones can be identified?
2. What is the shortest duration at which the pitch of a tone is identifiable when presented in a 2-tone or 3-tone sequence?
3. Is there a logical progression in minimum duration for identification of pitch from a single tone through to continuous tone sequences?
4. Does the presence of other tones affect (or mask) the perception of short tones equally, regardless of the relative position of the masking tone or tones? Specifically:

a. Does the position (first second or third) of the target tone affect the minimum duration required for perception of pitch in 2-tone and 3-tone sequences?

b. Does the relative pitch (low middle or high) of the target tone affect the minimum duration required for perception of pitch in 2-tone and 3-tone sequences?

5. If so, what mechanisms cause the differences in the minimum duration for perception of pitch in sequences of contiguous tones?

6. Can the information gained from this research be used to assist the education of secondary music students?

Chapter 2: Literature Review

The aim of this literature review is to report on studies that have examined the minimum duration for perception of pitch (MDPP) in a contiguous melodic context or that have examined areas of sound perception that might shed light on pitch perception in short tones.

Defining Pitch and Pitch Change

Pitch is a cognitive construct that is strongly linked to the frequency of vibration of a sound (Nave, 1999; Pierce, 1992). It has been defined by the Acoustical Society of America Standards as: “That attribute of auditory sensation by which sounds are ordered on the scale used for melody in music. . . The pitch of a sound may be described by the frequency of that simple tone having a specified sound pressure level that is judged by listeners to produce the same pitch” (ASA, 2021). Change in frequency during the presentation of a sound usually equates to a proportional change in the position of a sound on this pitch scale. However, this may not be the case for very short sounds. Although changes in frequency during a sound are usually associated with the perception of changes in pitch, it is also used in perception of timbre and speech (Gay, 1968; Saldanha & Corso, 1964; Seashore, 1938); thus, changes in frequency at very short durations also affect timbre and speech perception.

Describing stimuli in this Review

It should be noted here that all stimuli discussed in this literature review (with the exception of studies of single tones) have been made up of combinations of sound segments. **References to stimuli will always mean the sound segment combination that is presented to the participant**. Except for Patterson and Green (1970), **durations indicated in research results reported here refer to the duration of each sound segment within the specified**

stimulus. The duration of the entire stimulus can therefore be calculated by multiplying the segment duration by the number of segments in the specified stimulus. Some studies, such as Patterson and Green (1970), Green (1973), (Warren, 1974) and (Wier & Green, 1975) involved the presentation of two stimuli during each trial for the purpose of comparison. In these cases, both stimuli consisted of combinations of sound segments (except for Patterson & Green, 1970) and the two stimuli were separated by a comparatively long gap. Patterson and Green's (1970) stimuli, called Huffman sequences, consisted of single sounds which incorporate delayed frequency bands within each single sound. Even though changes happened within this sound, Patterson and Green's stimuli will be reported by their total duration.

No studies have specifically targeted the MDPP in short contiguous sequences. To gain a full understanding of current and historical knowledge in this area, studies that used sine wave tones (and other pitched tones) will be examined. The first studies reviewed were those examining minimum durations for perception of pitch in isolated tones. The following research was used to establish a base minimum duration comparison with other minimum durations.

Single Tone Research

Bürck, Kotowski, and Lichte (1936) were the first to establish a continuous range of minimum durations depending upon the frequency of tones being examined. Their results (using novice participants) formed a truncated 'u' shaped (or 'ski jump') function, with minimum durations for a 50 Hz tone requiring around 60ms (for a 1000 Hz around 15ms; for a 10,000 Hz around 20ms).

Doughty and Garner (1947) used two highly trained participants on two tasks. They determined that a sense of pitch (as opposed to a click sound) first emerged at durations ranging from 18ms, for a 125 Hz sine wave tone, to around 4ms for tones in the range of 1000 Hz to 8000 Hz. For a sense of pitch to predominate over any click sound, tone durations ranged from 24ms for a 125 Hz tone to around 10ms for tones in the range of 1000 Hz to 8000 Hz.

Patterson et al. (1983) sought to identify the threshold duration for *melodic pitch*. Participants selected which of four isolated tones changed pitch on a second presentation of a 4-tone melody. The four participants ranged in age from 24 to 44. All had normal hearing and had received diverse musical training. The duration between the onset of each tone was a constant 660 ms. The duration of each tone ranged from 10ms to 80 ms. In the second presentation of the sequence, one tone was raised or lowered by one ‘scale step’ (i.e. either one or two semitones). The tones were all drawn from a single diatonic scale. In the test example provided in the publication, the second tone in the repeated sequence was two semitones lower. The Figure presented for sine wave tone identification accuracy (p. 3) showed that 62.5% accuracy (i.e. more than 50% above chance for a 4afc study) was achieved at 10ms for 900 Hz tones, 20ms for 400 Hz tones and 40ms for 200 Hz tones. Patterson et al. (1983) commented that, based on the results, it takes about 7 wave cycles of a tone in the frequency range being tested to support ‘melodic pitch’. However, based on calculations from the results in the Figure, participants achieved 62.5% accuracy once tone durations reached 8 wave cycles.

Defining the Minimum Duration for Perception of Pitch for a Single Tone

Although the term “pitch” has already been defined above, within this definition there is a wide variety of more specific meanings, such as “click pitch”, “tone pitch” (Doughty & Garner, 1947), “minimal pitch” (Pollack, 1968) and “indefinite pitch” (Souza, Batista, & Souza-Filho, 2015). The online Encyclopedia Britannica defines pitch:

In music, position of a single sound in the complete range of sound. Sounds are higher or lower in pitch according to the frequency of vibration of the sound waves producing them. A high frequency (e.g., 880 hertz [Hz; cycles per second]) is perceived as a high pitch and a low frequency (e.g., 55 Hz) as a low pitch. (Encyclopedia_Britannica, 2020)

In the studies that follow, the critical dependent variable will be a measure of the construct "accurate pitch perception", adopted from Patterson et al.'s (1983) research. In their study they identified "melodic pitch" as "the stimulus duration required to know which member of a four-note melody has been changed by one note on the second presentation of the melody" (Patterson et al., 1983, p. 322). In this thesis, accurate pitch perception will be identified as *the exposure duration required to place a sine wave tone in the range of tones to within a semitone of its true position.*

The minimum duration for perception of melodic pitch (MDPP) in this study has been taken from Patterson et al.'s (1983) results. The simplest way of referring to this duration is by citing it as a figure relating to the number of sine waves required to establish melodic pitch: 8 sine waves. In terms of actual duration for the frequency range being studied, this equates to between 34 ms for the lowest tone (C4 or 261.6 Hz) to 8ms for the highest tone (C6 or 1046.5 Hz).

Pitch Perception of Tones in Short Sequences

From studying single tones, researchers branched out into three other areas that required participants to identify the pitch of tones: identifying short cyclic melodies, identifying the tones as 'high' and 'low' in masking studies, and identifying pitch changes in tone sequences which included gaps. This work is reviewed in the sections that now follow.

Short Cyclic Melody Identification

Despite a wide search, only two studies have been found that used contiguous sequences of tones and required the identification of the pitch of the tone. Both involved identifying melodies from continuously cycled sequences of tones. Identification of short cyclic melodies was used by the researchers to identify the limits of melodic perception (Warren, Gardner, Brubaker and Bashford Jr., 1991) and to study cognitive decline as we age (Andrews, Dowling, Bartlett and Halpern ,1998). Warren et al.'s research (mentioned above and outlined below) will be used in Chapter 5 as the standard for the MDPP in continuous cyclic tones.

Warren et al. (1991) studied perception of short familiar melodies, to determine the shortest duration at which familiar melodies can be identified, when played in a continuous cycle. They created eight stimuli consisting of isochronous versions of familiar melodies, using sine waves with note pitches between D#4 and D# 5 (311 to 622 Hz). Isochronous versions of familiar melodies are created by changing all the notes in the melodies to the same duration, thereby removing rhythmic cues. Warren et al. reported that the overall or grand median for the lower limit of melodic recognition was 160 ms. They also reported that at very brief tone durations it was not possible to identify the cycled melodies, even with musical training. However, the rapid series of tones appeared to form a distinctive pattern, although the pattern was not melodic. In order to examine this effect, a further study was undertaken, which will be reported later in this review (see section: “1-10 ms: Analogue analysis of sequences” last paragraph).

Andrews et al. (1998) adopted the task developed by Warren et al. (1991) to identify thresholds for speeded and slowed melodies and used it to test for differences between age groups. They found an average minimum duration of 204 ms per tone for inexperienced listeners (who averaged 0.3 years of music lessons) but a significantly shorter 162 ms per tone for participants who were professional musicians, (who had generally been practicing music actively and continuously from about 6 years of age). Moreover, mean recognition time for younger music professionals was even shorter (123 ms), which was around the same result reported by Thomas and Fitzgibbons (1971) for success in identifying sequences of pure tones.

These results both fall within the “125 – 200 ms duration band identified in the “Cyclic Presentation of Sound Sequences (single stream)” section presented below. This means that the MDPP for cyclic melodies is approximately the same as the MDPP for identifying the order of cyclic sequences of similar sounds. From this information it can be inferred that if the order of continuously cycled tones is perceptible, melodies that those tones create are understandable.

Masking

Many studies on masking used pitch perception as their dependent variable (Kallman & Massaro, 1979; Leshowitz & Cudahy, 1973; Massaro, 1970; Massaro, 1971; Massaro, 1972a; Massaro, 1973; Massaro, 1975). These studies provided useful information for this thesis in two areas. Firstly, by providing information relating to relative levels of masking that might be expected in 2-tone and 3-tone sequences depending upon the position (first second or third) and relative pitch (highest tone, middle tone, or lowest tone) of the tone. Secondly, by providing possible reasons as to why these masking differences might occur.

There are two classes of masking. They were originally called ‘detection masking’ and ‘recognition masking’ (Massaro, 1972a) but, more recently, have been termed ‘energetic masking’ and ‘informational masking’ (Middlebrooks & Simon, 2017). Detection (or energetic) masking relates to the overall sound becoming more difficult to hear. When a sound undergoes full detection (energetic) masking, the target (or masked) sound cannot be perceived from amongst the other sounds being heard (Patterson & Green, 2012). This effect is contrasted with recognition or informational masking, where the target sound is heard, but some aspect of the sound cannot be recognised accurately. Aspects of sounds that can be masked include loudness (i.e. energetic masking) pitch, spatial location and timbre. Because the focus of this thesis is on informational masking of pitch, studies relevant to the masking of pitch will now be examined, following some general comments.

Although the intent of masking studies was to provide information on the masking of a target tone, many of the studies involved identifying the first tone (or second tone) of a series of two tones (Kallman & Massaro, 1979; Leshowitz & Cudahy, 1973; Massaro, 1970; Massaro, 1971; Massaro, 1972a; Massaro, 1973; Massaro, 1975). On the occasions that the masking tone was presented with no gap, the resulting sequence is very similar to the 2-tone sequences used in this thesis (see "Chapter 5 Study 2: 2-tone condition").

Stimulus Onset Asynchrony. Several studies have identified differences in stimuli using Stimulus Onset Asynchrony (SOA). The term Stimulus Onset Asynchrony (SOA) is avoided throughout this thesis because it can cause confusion. For example, an SOA of 70ms could mean a 10ms tone with a 60ms gap (Leshowitz & Cudahy, 1973), a 20 ms tone with a 50ms gap (Massaro, 1972), a 60ms tone with a 10ms gap (Massaro, 1972), a 70ms tone with no gap (such as the stimuli used in the studied conducted for this thesis) or a 500ms and a 430ms tone started 70ms apart and stopped simultaneously (Hirsh, 1959). All these 70ms SOA stimuli resulted in different outcomes.

Specifying the Type of Masking Being Studied.

There are several different ways by which sounds can be masked. In temporal terms, sounds can be masked by other sounds either occurring simultaneously, or following the target sound (backward masking), or before the target sound (forward masking). In terms of sound quality of maskers, broad spectrum sounds like white or pink noise can create masking, but single or multiple individual tones can also mask sounds. Sounds with similar frequency, and especially sounds within the ‘critical band’ (sounds that stimulate the same area on the basilar membrane), generally produce the most marked masking. However, sounds that do not contain frequencies in this crucial area can also mask a sound (Neff & Callaghan, 1988). Because the intent of this thesis is to examine the identification of pitch in contiguous sine wave sequences, *the types of masking of interest here are forward and backward informational masking, caused by contiguous but non-overlapping sine wave tones.*

In a 3-tone study, such as the 3-tone study condition reported in chapter 5, the type of masking that could affect perception of a particular tone changes with the position of the tone in a sequence. The first tone in a 3-tone sequence is backward masked by two single contiguous sine wave tones. The last tone in the sequence is forward masked by two single contiguous sine wave tones and the middle tone is both forward and backward masked by a single tone each.

Several studies (for example: Massaro, 1970) have explored masking using single sine wave tones as maskers for other sine wave tones. Most have used 10ms - 20ms test tones masked by a sine wave tone of different pitch and, usually, different duration. In all studies examined in this thesis, two test tones were used: one labelled “high” and the other “low”. Participants were generally asked to identify which of the two test tones (high or low) was presented. In most of these studies, the extent of masking (judged by the percentage of correct answers using a method of constant stimuli) was varied by placing a silent gap (usually ranging from 20ms to 500 ms) between the end of the test tone and the start of the masking tone (Massaro, 1970). In addition to varying gap duration, some studies have also modified the pitch difference between two test tones (Leshowitz & Cudahy, 1973). Other studies have used a comparison paradigm where both high and low tones were presented but the order of presentation was randomised (Leshowitz & Cudahy, 1973; Ronken, 1972); and one study used a comparison paradigm but only masked one of the tones in each trial (Kallman & Massaro, 1979). Other researchers have used variables including changing the duration of the masking tone and varying the number of masking tones presented, but this research will not be covered here.

Research results have been presented in one of two forms. The first, and most common, is the percentage of correct responses for each variable at each level. Variables reported have included gap duration, masker duration, masker pitch and frequency difference between test tones. A different method was used by Leshowitz and Cudahy (1972). They varied pitch difference between the two test tones as well as gap duration and identified the minimum frequency difference for 75% correct frequency ID at each gap duration.

Masking studies have been used for a variety of reasons, including understanding how masking works (Greenwood, 1961; Wegel & Lane, 1924), exploring memory limitations (Massaro, 1970), examining speech intelligibility in a noisy environment (Kidd & Colburn, 2017), understanding sound compression strategies and limitations (Painter & Spanias, 2000), for intelligence tests (Deary, Head, & Egan, 1989; Grudnik & Kranzler, 2001; Nettelbeck, 1987), and for understanding how sound perception works (van Dijk & Backes, 2003). Consequently, sound masking is an important subject to explore; and understanding more fully how pitch perception is affected by tones in different positions (i.e. the intent of this thesis), would make a valuable contribution to knowledge. The research reviewed below has been grouped into the types of masking used, beginning with backward and forward masking of tones by other tones; the method used in this thesis.

Backward masking

Massaro (1970) used masking studies to test for the existence of ‘echoic memory’, also known as ‘sensory memory’. Echoic memory relates to an immediate storage system that retains the acoustic properties of sounds (Cowan, 1984; Li & Cowan, 2014). He used the masking effect to establish the duration of interference to information processing resulting from a ‘masking tone’. In his first study, Massaro examined forward and backward masking using single tones. He theorised that the presentation of a second (masking) tone immediately following a target tone would halt processing of the first tone in echoic memory, thereby limiting participants’ ability to identify the tone. This assumption matches findings from brain wave research that reported that both beta and gamma waves reset when new sound information is presented (Buzsáki, 1989, 2005; Galambos et al., 1981; Klimesch, 1999; Zanto et al., 2005). Massaro reasoned that sound ‘maskers’ presented later than the duration limits of echoic memory would have no effect on tone identification and, by varying the duration between target and masking tones, Massaro aimed to establish the maximum duration of echoic memory.

In Massaro’s first experiment, the test tones were two 20ms sine wave tones. The high tone was 870 Hz and the low tone was 770 Hz. The masking tone in the first experiment was 820Hz and lasted 500 ms. The silent inter-tone interval in all experiments in this study were 0, 20, 40, 80, 160, 250, 350, or 500 milliseconds (ms). The three participants were asked to identify the target tone as either high or low. They were expected to use their memory of both sounds (from prior presentations and from practice) to differentiate between them. They practiced for about 15 hours over two weeks before completing experimental trials. Results showed identification improvement up to 250ms, at which point identification success reached asymptote. From this, Massaro concluded that echoic memory lasts for around 250 ms.

As will be seen (and discussed) later in this chapter, this duration matches that of the Theta brain wave responsible for transmitting information to higher processing areas and for memory storage (Buzsáki, 1989, 2005; Klimesch, 1999).

Ronken (1972) undertook a study on both forward masking and backward masking (see the *backward masking* section below) using a different method to Massaro. He also used high and low test tones, but without gaps between the test tone and masker and both high and low test tones were presented at every trial run, with a 500 ms gap between high and low tone presentations. Participants chose whether the higher frequency tone was presented first or second. The frequency difference between the two test tones varied from 4 Hz to 512 Hz. The two test tones (one high and one low) were centred around the 1000 Hz frequency (the frequency of the masker). Stimuli were presented in one of three conditions: both test tones forward masked, both test tones backward masked, and both test tones with no mask. Four high school students had at least 9000 practice trials before undertaking the study. Ronken reported that the presence of a trailing tone (i.e. backward masker) decreased the discriminability of the signal. Although not reported in the study, upon examining Figure 2, a frequency difference of 64 Hz was required for a discrimination accuracy of 75% when stimuli were presented with no masking tones, whereas a frequency difference of 128 Hz was required for a discrimination accuracy of 75% when stimuli were backward masked.

Leshowitz and Cudahy (1973) undertook a study using a method similar to Ronken (above), but also included gaps between maskers and target tones. They claimed that their results contradicted Massaro's (1970) findings. They started with 20ms test tones with 500ms maskers but added 10ms test tones because they found no changes in masking effects when they tried to replicate Massaro's results using 20ms test tones.

In their first experiment they used a comparison paradigm where two sets of 20ms test tones, each followed by a masker, were presented to participants (N = 4). One test tone was lower than the other and the participants were asked to identify in which interval the lower tone occurred. Leshowitz and Cudahy used two variables: changing the frequency difference between the two test tones; and changing the gap duration between the test tone and the masker. Instead of making identification success the dependent variable (as in the Massaro experiments), they kept identification success as a constant and tested for the minimum frequency difference for 75% accuracy in each of the four gap durations. Leshowitz and Cudahy did not find a significant difference in minimum frequency levels for any of the different gap durations. They then repeated the experiment, replacing 20ms test tones with 10ms tones but found no statistically significant differences for gap durations. However, examining their results (Figure 2, p. 884), a large change is apparent in minimum frequency differences at different gap durations in all three backward masking conditions. It is likely that the -20ms SOA result in the 800 Hz test tone centre frequency with the 1000 Hz interference tone (the middle graph in Figure 2) substantially affected their reported results; this result ran counter to the other results, in that participants were able to identify the stimuli at an average of 5 Hz in this test condition, comparted to around 20 Hz for all other conditions, for SOAs up to 40 ms. This 5 Hz result should be treated with caution; it was one of the shortest results in the whole study and was the only time all four participants had an identical minimum frequency difference. This result was also shorter than any other backward masking result, including at the longest gap duration in the 20ms test tone (experiment 1), where the test tone was double the duration of the current tone and there was a 120ms gap before the masking tone. The reason for this result may have been a glitch in the experimental method, or that the pitch of the marker affected the result because of the closeness in time of the masker to the test tone. In any case, without this one result, there is a clearly a masking effect of the 10ms test tone duration, which diminished as gap durations increased. Therefore, putting the problematical 5 Hz result aside, Leshowitz and Cudahy's results clearly supported Massaro's claim of decreasing effect of masking as gap durations increase, rather than contradicting it.

Forward Masking

In his second experiment, Massaro (1970) used the same method as his first backward masking study for a forward masking study. The masking tone was presented and then followed by a silent inter-tone interval of 0 – 500ms, after which the test tone was played. Massaro found no evidence of any form of forward masking under these conditions. Unfortunately, he did not present an unmasked tone for a baseline comparison so he could not check the forward masking results against a baseline. However, the next study reported (below) did include an unmasked tone in the study.

In the forward masking component of Ronken's (1972) study (see above for further details), he reported that a frequency difference of just 34 hertz was required for a discrimination accuracy of 75% in the forward masking condition of his study, compared to 64Hz for the no-mask condition. Ronken observed that the forward masking condition improved identification. This was a common result across other forward masking studies considered here, where unmasked tones were presented for comparison (Leshowitz and Cudahy, 1973; Massaro, 1973). It clearly meant that, instead of reducing the amount of information available to a participant, a masking tone placed before a test tone provided information that could be used to assist test tone identification, thus improving results in the forward masking condition, compared to the no-mask condition. This is likely to be because the masking tones used in these studies were centred between the frequencies of the two test tones. Thus, the frequency of the stimulus was perceived as ascending when the high test-tone followed the middle-placed masking tone and descending when the low test-tone followed the middle-placed masking tone. Previous research by Whitfield and Evans (1965) found that, at least in the brain of un-anaesthetised cats, some neurons only fired if a frequency went from low to high, whereas others fired if a frequency went from high to low. They also reported that more neurones fire in these two conditions than fire when unchanging tones were presented, thus making frequency change direction easier to perceive than the pitch of steady state tones. This response to frequency change has been noted in more recent studies and labelled ‘mismatch negativity’ (MMN) (Molholm et al., 2005). The theory presented here is that discrimination of frequency change in the ‘up/down’ direction, described above, was used to identify stimuli in the forward masking condition of the studies reported here, rather than identifying the different pitches of the test tones. Thus, stimuli were more easily identified in the forward masked condition than in the ‘no mask’ condition.

Masker Differences with Different Frequencies

One of the aims of this thesis is to discover if the relative pitch and position of tones has any effect on the MDPP of tones in a sequence. If (for example) the MDPP of low tones is strongly affected by higher tones that follow them in a contiguous sequence of tones, one would expect a similar result from masking studies that used backward maskers that were of a higher frequency than test tones.

In his fourth experiment, Massaro (1970) used two masking tones of differing frequencies. All other conditions were the same as for his first backward masking experiment (described above). Massaro found that the higher tone (999 Hz), which was higher than both the ‘high’ and ‘low’ target tones, produced more overall interference with pitch identification than did the lower (820 Hz) tone, which was placed between the frequencies of the two target tones. Massaro decided to study this effect further. His final experiment (experiment V) was designed to check if the frequency of maskers affected the result, when presented in a more random condition than that from experiment IV. The pitch of the backward masking tone was randomly varied between four frequencies. This contrasted with the previous experiment IV, where each masker was presented in different trial runs. Massaro reasoned that the previous results could have been caused by participants using different strategies for the two-masking tone conditions. In experiment V, one of four masking tones (540, 660, 720 and 999 Hz) were randomly presented to participants. Massaro found no significant differences in identification success between the four maskers. This result ran counter to his experiment IV and counter to results from his later studies (see below). It is noteworthy that none of the masking tones presented was of a lower frequency than the test tones. Thus, any possible difference between the effects of high and low masking tones could not be identified. It is also possible that the lack of effect by the different masking tones was influenced by the (unknown) training instructions to participants.

Divenyi and Hirsh (1975) appended a 40ms tone to the 3-tone sequences that they had used in an earlier sound ordering study. This fourth tone was added to act as a masker for the other tones. This effectively made four tone sequences, but only the first three tones were used for identifying the sequence (Divenyi & Hirsh, 1974). The masking tone (i.e. the last tone) was randomly selected from one of nine frequencies located both above and below the pitch of the other tones, as well as from within the same pitch range. Participants were asked to identify the order of the first three tones, in a similar manner to his 1974 study (see below for details of this study). Divenyi & Hirsh reported that identification indices of all four subjects were most depressed when the frequency of the masking tone lay 1/6 to 1/3 octave higher than the highest pattern frequency. Put simply, the initial 3-tone sequence was harder to identify if the fourth tone was higher in frequency than the other tones in the sequence.

Massaro et al. (1976) completed further backward and forward masking experiments using different frequency masking tones. He used maskers both higher than and lower than the test tones, as well as a masking tone pitched between the pitches of the two test tones. The 20ms test tones were 860 Hz (high) and 790 Hz (low). The 20ms masking tone was either high (900 Hz), middle (825 Hz), or low (750 Hz). The gap duration between the masking tone and the two test tones was 0, 20, 40, 80, 160, 250, or 350 ms. Participants were considered relatively naïve and were tested an hour a day for five days. Massaro et al. reported that the middle frequency masking tone produced about 4% less masking than the higher or lower frequency tones in the backward masking condition. There was also some forward masking apparent at the zero-gap masking condition, which runs counter to results from other studies completed in the same decade.

In this study, all experimental conditions were randomized within a given session so the participants could not adopt different processing strategies under the different experimental conditions. Unfortunately, results reported did not include comparisons between high masking tones and low masking tones, so it is impossible to tell if either high or low masking tones had a stronger backward masking effect than the other.

Massaro et al. (1976) considered the possibility that the reason for the lower masking rate in the middle frequency was that observers sometimes recognized the masking tone as the middle frequency and compared the pitch of the test tone directly to the mask, to determine whether the test was higher or lower. However, they argued against this interpretation stating that, if observers were simply making relative judgments, they would expect forward masking to be symmetrical with backward masking and this did not occur. In addition, performance should have been easier under the masking conditions at long SOAs than under the no-mask condition, but this was not the case. They concluded that participants did not appear to make relative judgments in the pitch identification task and the reason for the slightly reduced masking with the middle-frequency masker must be found elsewhere.

However, studies such as Wier and Green's (1975) show that it is possible to differentiate between high/low stimuli, similar to those used by Massaro et al. (1976), at tone durations as short as 2 ms. Therefore Massaro et al.'s conclusion that relative judgements could not have been used were clearly inaccurate. The fact that participants in the Massaro et al. study made so many mistakes in this condition, is surprising considering the Wier and Green result and most likely relates to the instructions given to participants and their relative inexperience in the task.

Kallman and Massaro (1979) made three experiments, examining various aspects of backward masking. Nineteen first-year psychology students participated after two days of practice. In experiment 1, a 20ms standard 1000Hz tone was presented followed by 500 ms gap. This was then followed by 20ms test tone, which was followed by a 100ms masking tone after gap durations of 10, 20, 40, 80, 160, 250, 350, or 500 ms. There were seven different masking frequencies. The frequency of one of the maskers was the same frequency as the standard tone. Other frequencies were both higher and lower than the standard and test tones. Results showed that performance improved from short to long masking gaps. Kallman and Massaro also reported that maskers of similar frequency were harder to identify than other maskers and that low tones with high maskers and high tones with low maskers were most inaccurate. Another effect apparent in their Figures -- but not reported by Kallman and Massaro (possibly because it was a relatively small effect) -- was that the test tones revealed greater backward masking from higher frequency masking tones than from lower frequency masking tones.

Based on the research reported above, there is some evidence that low tones are more affected by masking than high tones, but this result is not universal.

Summary of Masking Research.

The masking research presented above suggests two different results: one for forward masking and one for backward masking. Regarding backward masking, studies have found that sine wave tones played after a target tone make it harder to identify the target tone. Regarding forward masking, many of the studies found that tones played before a target tone made it easier to identify the target tone. This seems intuitively unlikely, and this result is not supported by findings from other non-masking studies. As discussed above, the placement of the forward masking tone between the two test-tones in these studies, may have assisted participants identify ‘high’ and ‘low’ test-tones, because the sound would seem to rise if the high test-tone was presented and descend if the low test-tone was presented. Consequently, it is difficult to assess the real masking effect of forward maskers on sine wave tones. However, research to be presented in Chapters 4 and 5 of this thesis should contribute to our understanding in this area.

It is worth noting at this point, that research into brain wave functioning has supported Massaro’s viewpoint regarding echoic memory. As will be explored later (see “Sound Cognition Research” below), the gamma wave (which is the primary information processing brain wave trace) stops a processing cycle as soon as new information becomes available (Galambos et al., 1981; Zanto et al., 2005). This supports Massaro’s theory that the presence of another sound disrupts processing of an earlier sound. Further to this, the theta wave, which is responsible for moving information to memory, lasts between 125 and 250ms.

Processing is still carried out throughout this process but stops as soon as the wave cycle concludes (between 160ms and 250ms after a stimulus arrives) (Buzsáki, 1989, 2005; Klimesch, 1999). This explains Massaro's findings, that delay to delivery of a masking sound longer than 250ms does not influence a participant's ability to identify a sound.

Gapped Tone Research

Watson et al. (1975) studied semi-contiguous sequences of 40ms tones. Each sequence had ten 40ms tones and there were 40 ms gaps after the third and fifth tones. Only tones in four of the 10 positions were changed during the experiment. The tones that were changed were the first and the last tones, the fifth tone (which occurred just before a gap) and the sixth tone (which had a gap before it but not after it). Only one of the tones was changed in each trial. For all experiments, the standard tone/gap pattern was presented on each trial, followed by a comparison pattern. The four participants, aged between 16 and 27, and with mixed musical experience, were given between 15 and 60 hours training. Participants decided if the second tone sequence in each trial was the same as the standard sequence or was a modification of that standard. After each selection, participants were given feedback for accuracy. Results were given as the minimum detectable frequency change for each tone position.

Watson et al. (1975) found that the changes to lower tones, relative to the other tones in the sequence, required larger frequency differences for identification than higher tones. They also found that changes to tones presented immediately before a 40ms gap were more easily identified than all other tones, excepting the last tone. While experimenting with the duration of the silent gap, they found that most participants reached asymptote for identifying the tone after the gap duration was increased to 80 ms, although one of the four participants required more than 120ms to reach asymptote. The frequency difference required to identify the tone before the gap was similar to that required to identify the last tone; and both tones required significantly lower frequency differences for identification that tones in the other positions. Watson et al. (1975) commented that, instead of the well-known ‘recency’ effect being the reason that the last tone in a sequence was easily recognised, it was because the last tone has access to more undisturbed processing time than other tones in the sequence.

Watson et al. (1975) stated that high-frequency components were more easily identified than lower ones, suggesting that the highs may be given a pre-emphasis in storage, or some other reason caused them to be particularly resistant to interference. This suggestion is consistent with other findings that have reported that masks lower in frequency than the target tone produced less interference in backward recognition masking than masks higher in frequency than the target tone.

The results from the above research into masking and gapped tones may help predict results for pitch perception of 2-tone and 3-tone sequences. To summarise, high tones (relative to other tones in a sequence) should be more easily identified than low tones and the last tone, having more processing time available, should be more easily identified than tones in any other position.

Tone Sequence Ordering

Except for the masking studies, the studies on short cyclic melody recognition, and the study on gapped tone sequences (all detailed above), most studies that included sine wave tones as stimuli focussed on identifying the order of the sequence of tone segments, without specifically relating these segments to pitch perception. Because pitch has not been identified in these studies, it cannot be assumed that pitch perception was used, or was even available to use, to identify stimuli in some (or all) of these studies. Alternative strategies that can be used to order sounds (other than pitch perception) include sound analogue comparison and, at least for short sequences, frequency change direction detection.

Sound analogues, according to Neisser (1967), are holistic representations of a sound sequence consisting of a set of the distinguishing features of that sequence. In contrast, *strings* consist of discrete (and usually verbalized) representations of each sound segment. If sufficient time is available for individual processing of each segment, strings are created, and the pitch of each tone should be perceptible. If the available information is not sufficient to identify each of tone separately (owing to the brief time each tone is available for processing), an analogue of the entire tone sequence is created. This means that identifying the sequence order of tone segments can be achieved, regardless of whether the sequence is perceived as a string of discrete sounds or as an analogue of the overall sequence. If an analogue of a sound sequence is required to discriminate sound order, the pitch of each sound is clearly not available for identification. As will be discussed later in this chapter, another strategy for identifying short sequences of tones is frequency change direction detection. Just knowing if the frequency of two tones went from low to high, or high to low can be enough to identify the stimuli in some studies. In other studies (discussed below), sequences were too long for this simple strategy, but sound analogue comparison was still available as an alternative to pitch detection.

Similar explanations to that given above for sound analogue detection have been given by Massaro (1975) and Warren and Bashford (1993), but instead of describing the effect as sound analogues, they have described it as creating a ‘gestalt’ of the sequence, where an integration mechanism or ‘gestalt’ provides an overall impression of a sound sequence that can be used to differentiate between sound sequences.

Further evidence that it is not necessary for pitch identification to be used in tone sequence ordering is provided by the minimum durations required for ordering tones. Some of the studies reported below have included minimum durations shorter than 4ms for identifying the order of sounds. However, because the minimum duration required to establish a sense of pitch is 4ms (Doughty & Garner, 1947), it stands to reason that some of the sound ordering results cannot possibly have been achieved through pitch identification.

Even at segment durations longer that the minimum 4ms required for establishing a sense of pitch in a single tone, it is uncertain if the pitch of the tones being examined was identifiable, or not, in most of the studies, because of the limiting effect of masking on pitch identification. Conversely, some studies (reported below) have required over half a second per tone segment to identify the order of the tone sequence; and it is highly likely that pitch identification was possible at these durations. The important point is that being able to identify the order of sequences is not proof of pitch identification for tones in that sequence. Speculations as to the reasons for the wide variety in minimum durations for tone sequence ordering will be examined in the discussion section at the end of this chapter.

Tone sequence studies can be categorised as falling into four duration bands depending on the independent variables being studied. The bands are; 15 - 75 ms to identify the order of two sounds in isolation; 125 – 200 ms to identify order within a cyclically repeated series of similar sounds; 300 – 700 ms to identify order within a cyclically repeated series of disparate sounds (when the sounds are perceived as being from different sound sources) and; 1 – 10 ms to differentiate between a set number of patterns of different sequential sounds. Table 2.1 (below) presents a summary of aims, methods, and results for studies that have examined sound ordering.

Table 2. 1

Aims, Methods and Results of Short Tone Studies

15-75 ms. the 50 ms threshold and single presentation of sound sequences

Author	Aim	No. of segments	Stimuli type	C (Cyclic) S (Single)	No of choices	No of participants	Training/ Experience	Response Mode	Result ms) per sound segment
Hirsh (1959)	To discover the shortest duration at which the temporal order of two sounds can be identified.	2	Twelve different sound pairs. Duration between onset times: 0 ms to 60 ms	S	2	5	Low / A panel of five listeners	Identify the order of sound presentation	20 ms

125 – 200 ms. Cyclic presentation of sound sequences (single stream)

Author	Aim	No. of segments	Stimuli type	C (Cyclic) S (Single)	No of choices	No of participants	Training/ Experience	Response Mode	Result ms) per sound segment
Thomas, Hill, Carroll, & Garcia, (1970)	To discover the minimum rate at which the order of vowel sounds can be identified.	4	sequences of vowels. Duration: 75 ms to 300 ms	C	6	25	Low / Students with no known hearing loss	Name the order of vowells	125 ms
Thomas & Fitzgibbons, (1971)	Shortest duration for four sine wave tones (within a perfect 4th)	4	Sine waves a perfect 4th or less apart. Durations unknown (possibly 75 ms to 500 ms)	C	6	not give n	not given	Identify the order of sound presentation	125 ms

Author	Aim	No. of segments	Stimuli type	C (Cyclic) S (Single)	No of choices	No of participants	Training/ Experience	Response Mode	Result ms) per sound segment
Warren, Gardner, Brubaker, & Bashford Jr, (1991) Exp. 1	To determine the shortest duration in which familiar melodies can be identified when played in a continuous cycle	8	Familiar melodies: "Camptown Races," "Yankee Doodle," "Rock-a-bye Baby," "God Rest Ye Merry Gentlemen," "Happy Birthday," "Twinkle Twinkle Little Star," "Skip to My Lou," "Love Me Tender". Duration: 40 ms to 3.6 seconds	C	8	30	Low / University students with no muscial training	Name the molody	160 ms
Andrews, Dowling, Bartlett, & Halpern, (1998)	To identify differences between older and younger participants in identified speeded and slowed melodies. Melodies: "Happy Birthday," "London Bridge," "Yankee Doodle" , "Frere Jacques," "Twinkle, Twinkle, Little Star," "Old Macdonald," "Swanee River," "Rock of Ages," and "Rudolph, the Red-Nosed Reindeer"	8	Familiar melodies Duration: 31 ms to 4 seconds	c	8	80	Low / From low to high experience	Name the molody	186 ms
		8	Familiar melodies with gaps between each repetition. Duration: 31 ms to 4 seconds	C	8	24	Low / Professional musicians		145 ms
						31	Low / Low experience		198 ms
		8	Familiar melodies with original rhythms. Duration: 31 ms to 4 seconds	C	8	80	Low / From low to high experience		198 ms
Nickerson & Freeman, (1974)	Examining differences between the ability to report order and to discriminate it. The premise was that having to verbalise increased requisite durations.	4	Sine wave sets. Set 1: 469 Hz, 586 Hz, 732 Hz, and 916 Hz. Set 2: 300 Hz 375 Hz 1,144 Hz and 1,431Hz, Set 3: 586 Hz, 631 Hz, 680 Hz, and 732 Hz. Duration: 200 ms set rate	C	6	6	25 test runs with each frequency set / paid high school students	Identify the order of sound presentation	200 ms

300+ ms. Cyclic presentation of sound sequences (with stream segregation)

Author	Aim	No. of segments	Stimuli type	Cyclic/Single	No of choices	No of part.	Training/ Experience	Response Mode	Result ms) per sound segment
Warren 1969	identify the order of four 200 ms disparate sounds presented continuously in cyclic form		40 Hz square wave, 1,000 Hz sine wave , 796 Hz sine wave, 2,000 Hz ocatave band noise, Duration: 200 ms only						
Warren & Obusek, (1972)	To test if ordering unrelated sounds is possible at 200 ms.	3	40 Hz square wave, 1,000 Hz sine wave, 2,000 Hz ocatave band noise. Duration: 200 ms only	C	2	100	Low / Pool of 520 general undergratuate psychology students. Each student gave 1 response to a single stimuli	Participant ordered cards to identify sequence	found to be not possible at 200 ms
	To test if the order of partially related sounds can be identified at 200 ms.	4	40 Hz square wave, 1,000 Hz sine wave , 796 Hz sine wave, 2,000 Hz ocatave band noise, Duration: 200 ms only	C	6	120		Participant ordered cards to identify sequence	possible only with successive sine wave segments
	To identify the shortest duration at which the order of unrelated sounds can be identified.	4	40 Hz square wave, 1,000 Hz sine wave, 2,000 Hz ocatave band noise, the vowel /i/. Duration: 200 ms, 300 ms, 450 ms, 670 ms, 1000 ms	C	6	150		Participant ordered cards to identify sequence	300 ms
						150		Verbal description of sound order	670 ms
Thomas & Fitzgibbons, (1971)	To identify the shortest duration at which the order of four sine wave tones (wider than a perfect 4th) and dissimilar sounds can be identified.	4	Tones more widely spaced than a perfect 4th / dissimilar sounds: "e.g., pure tones, white noise, etc." Durations unknown (possibly 75 ms to 500 ms)	C	6	not given	not given	Identify the order of sound presentation	more than 125 ms and up to 500 ms

1-10 ms. Analog analysis of sequences

Author	Aim	No. of segments	Stimuli type	C (Cyclic) S (Single)	No of choices	No of participants	Training / Experience	Response Mode and other info	Result per seg
Patterson & Green, (1970)	To identify the minimum temporal auditory acuity the ability to differentiate sound order) for phase changes	2	Changed phase spectra "Huffman sequences". 1.25 ms, 2.5 ms, 5 ms. 10 ms	S	2	3	High / Uni students	Differentiate between 2 stimuli after hearing both stimuli	2.5 ms
Green, (1973)	To Identify the minimum temporal auditory acuity for volume changes	2	Volume differences in sine waves 90 dB to 100 dB and 70 dB to 80 dB. Duration set at 1 ms per segment	S	2	3	High / Extremely well practiced on Huffman sequences	Differentiate between 2 stimuli after hearing both stimuli	1 ms
Wier & Green, (1975)	To test minimum temporal auditory acuity for frequency changes	2	1000 Hz and 2000 Hz sine waves, Durations: 1 ms to 128 ms per segment	S	2	3	High / Not given	Differentiate between 2 stimuli after hearing both stimuli	1 ms
Nickerson & Freeman, (1974)	Exp 2: to discriminate between stimuli containing sounds segments of different orders.	4	Sine waves of frequencies 469 Hz, 586 Hz, 732 Hz, and 916 Hz. Duration 2 ms - 200 ms	S	6	1	High / Second researcher	Differentiate between 6 stimuli	2 ms
Divenyi & Hirsh, (1974)	To identify the shortest segment duration that the order of three tones can be identified and to test if frequency differences affect this result	3	Sine waves: 706 Hz, 1000 Hz and 1413 Hz. Duration: 1 ms - 30 ms.	S	6	7	High / Some musical training	Differentiate between 6 stimuli	6 ms

Author	Aim	No. of segments	Stimuli type	C (Cyclic) S (Single)	No of choices	No of participants	Training / Experience	Response Mode	Result ms) per sound segment
Warren, (1974)	Distinguishing between permeated orders of non-related sounds.	3	Hiss, square wave, sine wave. Duration: 5 ms to 100 ms. Interstimulus interval (ISI) either 50 ms, 1.5 seconds or 3 seconds	C	2	5	High / Graduate students	Differentiate between two stimuli after hearing both (choose "same" or "different")	5 ms for ISI up to 1.5 seconds. 10 ms for ISI of 3
		3	Hiss, square wave, sine wave. Duration: 5 ms to 100 ms	S	2			Differentiate between two stimuli after hearing both (choose "same" or "different")	10 ms
		2	Hiss, square wave. Duration: 5 ms to 100 ms	S	2			Differentiate between 2 stimuli	5 ms
		3	Hiss, square wave, sine wave. Duration: 5 ms to 100 ms	S	2			Differentiate between 2 stimuli	10 ms
Warren, Bashford, & Gardner, (1990)	Examining participants ability to differentiate between different arrangements of vowels	3	Vowels. Duration: 10 ms to 100 ms	C	2	36	Low / University students who passed audimetric screening	Differentiate between two stimuli after hearing both (choose "same" or "different")	10 ms

Author	Aim	No. of segments	Stimuli type	C (Cyclic) S (Single)	No of choices	No of participants	Training / Experience	Response Mode	Result ms) per sound segment
Warren, Gardner, Brubaker, & Bashford Jr, (1991) Exp 2.	To examine if sequences of *non-melodic tones* can be differentiated at very short durations	3	Sequence of tones with different pitches: 988 Hz, 1661 Hz and 1794 Hz. Duration 10 ms to 5000 ms	C	2	36	Low / University students who passed audiometric test	Differentiate between two stimuli after hearing both (choose "same" or "different")	10 ms
	To examine if sequences of tones with *different timbres* can be differentiated at very short durations	3	Sequence of tones with different timbres: 800-Hz pulse train, 800-Hz sinusoidal tone, 800-Hz square wave. Duraton: 10 ms to 5000 ms	C	2	36		Differentiate between two stimuli after hearing both (choose "same" or "different")	10 ms

Note: Most headings are self-explanatory. However, the heading "C(Cyclic) S (Single)" may cause confusion. C (Cyclic) means that the sound segments were looped continuously without gaps. S (Single) means that the sound sequence was presented once. The number of individual sounds in each sequence is given in the column "No of segments".

Of all the research listed above, no study has been found that specifically identifies the pitch of tones as a means for testing the perceptual limits of short sound change. This is probably because most studies considered here were looking at limitations to speech perception, and so the cognitive strategy that participants used to identify stimuli was not considered important. Supporting evidence for this suggestion comes from Robinson and Patterson (1995), who found that the identification of the pitch of a tone is not essential for speech identification.

15-75 ms: The 50 ms Threshold and Single Presentation of Sound Sequences

During the 1930s most researchers of sound perception studied single, isolated sounds. The prevalence of single sound research is reflected in Seashore's (1938) description of the four "psychological attributes of sound". These were: "pitch, time, loudness and timbre" (p. 3). These "elemental components" only fully encompass monophonic presentations from single sound sources because they do not include the sonic information contained in multiple sound sources, or the spatial location information inherent in all natural sounds. (See Appendix 1 for a paper on this topic that was written during an early part of thesis candidature and published by the Australian Society for Music Education). However, even though sound perception researchers were studying single sounds, other researchers were examining speech perception and some of their research included examinations of pitch and how it changes.

They found that sounds occurring regularly at shorter durations than 50ms smeared together and made speech and other perceptual tasks difficult to understand; such as the minimum duration of phenomes for intelligible speech (Cowan, 1936; Joos, 1948), the durational limits for morse code (Joos, 1948), and the longest frequency of sound waves and click trains that provides a sense of pitch, rather than perception of discrete sounds for each wave or click (Guttman & Julesz, 1963; Whitfield, 1979; Winckel, 1967). Additionally, sonic reflections that arrive within 50 ms of the original sound blend together (i.e. the reflections are heard as a reverberation of the original sound), whereas a sonic reflection arriving later than 50 ms is heard as a discrete sound or echo (Winckel, 1967). Of particular relevance to this thesis, sequences of contiguous tones with segments shorter than 50ms blur together and sound as if they are smeared (Schouten, 1962; Van Noorden, 1975).

Taken together, this all seems to indicate that there is a limitation in cognitive processing that affects our ability to separate individual sounds at rates faster than around 20 times a second (i.e. every 50 ms). This conclusion will be further examined in the discussion section (below), but there is evidence to suggest that this effect is connected to the duration of cognitive beta waves (Haenschel, Baldeweg, Croft, Whittington, & Gruzelier, 2000).

Ordering tones at this duration is also probelematical. Schouten (1962) demonstrated that it was possible to recognise a unidirectional scale pattern played in a cyclic sequence if the duration of each note is 50ms or longer. However, this unidirectional scale pattern was found to be a special condition, because, once this simple pattern was disrupted by changing just a single note, it was no longer possible to discern sound order at this note duration. Winckel (1967) went so far as to state: “when two short tones of different pitch are struck one right after the other too rapidly, at least during the 50 ms period, they cannot be perceived surely in the right order” (1967, p. 52). However, Hirsh (1959) had already disproved this. In fact, as will be seen later in the review, none of the above effects applies when sound segments are not immediately replaced by other sounds. As can be seen in Table 2.1, for single presentations of a short series of tones, temporal order could be perceived when the duration of each component sound was 20 msec (Hirsh; 1959; Leshowitz and Hanzi, 1972).

Van Noorden (1975) examined 3-tone sequences at durations slightly above 50 ms per segment. He found that when the third sound of the series continued in the same direction as the second sound, pitch direction was successfully identified, at even shorter onset asynchronies than for two sounds. However, when the third tone changed to the opposite direction, judgements appeared to be much harder. Successful identification required double the SOA of judgements where the stimuli change remained in the same direction. This result has direct implications for this thesis as it may indicate that longer durations may be required for identification of pitch when sequences start or end on a middle tone.

In a more recent study, Bader, Schröger, and Grimm (2017) studied participants' ability to identify changes in contiguous sequences of 50ms tones. These researchers were looking for links between various changes in brain waves and participants' ability to identify changes in random patterns of tones under two conditions: a fixed pitch condition and a continually transposing condition. Participants could identify changes in tone sequences successfully, in both conditions, although participants were much more accurate in the absolute condition. In the fixed pitch condition, participants reached 75% accuracy after two repetitions, whereas in the transposed condition participants achieved a 25% hit rate after two repetitions and gradually increased up to 40% hit rate after 12 repetitions. Considering Schouten's demonstration of the difficulty in placing 50ms tones in a sequence (discussed above) and the research which has found that it takes over 125 ms per segment to order continuous sequences of tones (described below), it is likely that sound analogue analysis was used to identify the changes in this study.

125 – 200 ms: Cyclic Presentation of Sound Sequences (Single Stream)

By 1969, pitch change research had moved into two separate areas. The first area involved examining single presentations of sound segments and the second area involved examining cyclically repeated sound segments. Interest in cyclically repeated sounds was driven by the observation that, because speech uses continuous sounds, cyclically repeating a limited number of sounds more closely represented limitations to understanding of speech than would single presentations. Cyclically repeated sound research was further divided into two categories; single stream research and segregated sound research. Single stream research involved studying sequences of similar sounds, whereas segregated sound stream research involved studying sequences of sounds that did not sound similar and were identified as "disparate sounds".

As can be seen by the research results in Table 2.1, similar sounds can be successfully ordered if durations are longer than 125ms and the sounds are similar enough to be preconsciously allocated to the same sound source (Thomas, Hill, Caroll, & Garcia, 1970; Thomas & Fitzgibbons, 1971; Warren et al, 1991; Andrews et al, 1998; Nickerson & Freeman, 1974). Regarding perception of pitch, two of these studies used recognition of melodies as their identification criteria (see above for details) and it can be concluded that pitch perception in this duration band is sufficiently accurate to identify well known melodies for durations of 160ms or greater; but it is not known if pitch perception is possible in cyclic continuous sequences at durations shorter than 160 ms.

300+ ms: Cyclic Presentation of Sound Sequences (with stream segregation)

As can be seen from the "125 – 200 ms" section in Table 2.1 above, Thomas and Fitzgibbons (1971) found that the minimum duration necessary for the correct identification of the temporal order of equal segments of four pure tones in a continuously repeated sequence was 125 ms if the tones were all within a perfect 4th. However, with intervals wider than a perfect fourth, up to 500 ms was required to identify the tone sets. Furthermore, in their brief report, Thomas and Fitzgibbons found that "for a mixture of 'different' or disparate stimuli (e.g., pure tones plus white noise plus square wave tones) minimum segment durations were considerably greater than 125 msec" (p. 87). They postulated that much of the increase in segment time required for correct order perception resulted from the need to shift attention among different classes of stimuli. This seems to indicate that these stimuli had split into different streams and was the cause of the longer durations for successful identification. Thomas and Fitzgibbons conclusions have been supported by research from Van Noorden (1975), who made a comprehensive study of this effect and discerned a linear relationship between fission (where sounds split into different streams) and pitch distance.

Disparate sounds do not sound as if they come from the same sound source and they are therefore processed differently from similar sounds. As will be discussed later (see the "Discussion" section below), disparate sounds are preconsciously separated into different sound streams (e.g. all the oboe sounds are placed in one cognitive sound stream and all the clarinet sounds are placed in another cognitive sound stream). This enables listeners to pick out, and focus on, individual instruments within an ensemble (or sound sources in a normal environment) and to process these streams of sounds separately.

As can be seen from Table 2.1 above, **w**hen sounds are preconsciously allocated to different sound sources, up to 670ms is required to identify accurately the presentation order of cyclically repeated sounds (Warren, 1969; Warren & Obusek, 1972; Thomas & Fitzgibbons, 1971). The accuracy of pitch perception was not reported in any of the studies reviewed in the foregoing. However, as pitch perception is likely possible at 160ms per segment, it is highly likely that the pitch of each of the tones in this section was individually identifiable.

1-10 ms: 'Sound Analogue' or 'Gestalt' Analysis of Sequences

As can be seen from Table 2.1, when participants are well trained and are asked to choose between a small number of possibilities, discrimination between sound sequences is possible for durations less than 10ms and as short as 1ms. (Patterson & Green, 1970; Green, 1973; Wier & Green, 1975, Nickerson & Freeman, 1974; Divenyi & Hirsh, 1974; Warren, 1974; Warren, Bashford, & Gardner, 1990; Warren Gardner, Brubaker & Bashford Jr, 1991). The basis for this form of sound differentiation has not been established and is the subject of the first study in this thesis, presented in Chapter 3. Most studies reported that it was not possible to distinguish individual sounds at the shortest durations being identified, but the blurred sequences were reported as having identifiably different qualities.

Nickerson and Freeman's (1974) investigation into cyclic presentations of short tones (see the "125 – 200 ms" section above) included an examination of the limits of short tone duration perception for highly trained participants. With intense training the second author was able to reduce successful identification durations of continuously repeated sound order patterns down to 2 ms per segment. Reporting on the two experiments, Nickerson and Freeman noted that stimuli were most likely identified by temporal order at 200 ms but, as durations decreased, "the discrimination is based primarily on steady state spectral information" (1974, p. 476). For the very short stimuli, it was noted that: "one hears a steady state sound that has the quality of a tonal buzz" (p. 476). The authors concluded that identification was based on the detection of spectral differences between the stimuli. However, Patterson and Green (1970) and Wier and Green (1975) were able to eliminate spectral differences in their stimuli and yet participants in their study were able to discriminate between stimuli where individual components of a stimuli were as short as 1 ms. It seems therefore more likely that order information was the reason for identification success in Nickerson and Freeman's study, rather than spectral differences, even though the differences were perceived as different timbres. As the duration of this task decreased from 200 ms to 2 ms, the task seems to have crossed from identifying the order of four cyclic pitches to selecting between six different possible sound analogues; a task which no longer involves identifying changes in the pitch of sound segments.

Divenyi and Hirsh (1974) aimed to identify the shortest segment duration at which the order of three contiguous tones can be identified. Participants were able to differentiate between the six possible segment orders at durations as short as 2 ms for the unidirectional sequences (stimuli that went sequentially from low to high or high to low) and as short as 10 ms for the sequences that changed direction. The relevance of this study to the thesis lies in the increased ease of recognising unidirectional sequences, as opposed to sequences that change direction. Therefore, it may be the case that the pitch of tones in sequences that go straight up or straight down may be easier to identify than sequences that start or end on a middle tone. This effect has also been reported by Van Noorden (1975).

Wier and Green (1975) investigated frequency changes in very short sounds[2]. They aimed to find the shortest duration at which changing the order of frequencies can be differentiated. Stimuli segments consisted of a sine wave tone of 1000 Hz and a sine wave tone of 2000 Hz. Each trial consisted of two stimulus observation intervals with a 500 ms pause between them. To create a forced 2-choice task, Wier and Green arranged each trial, so that if one segment order occurred in the first observation interval, the opposite occurred in the second. Stimuli with segment durations of 132 ms, 64 ms, 32 ms, 16 ms, 8 ms, 4 ms, 2 ms and 1 ms were presented to participants from the longest duration to the shortest duration. There were three participants (of unknown origin or experience level) in the study and each of them was trained for a minimum of 20 hours before undertaking the experiment. Participants were required to declare if the stimuli that had the frequencies changing from low to high occurred in the first or second observation interval. Feedback for accuracy was given after each response, both during practice and during the experiment. During the

[2] This study is reported in detail because it has been replicated as part of Study 1 reported in Chapter 3.

experiment, participants were presented with repeated trials of the same duration until the group as a whole appeared to reach a stable level of performance (p. 1513). Their last 200 trials were then used for reporting differentiation accuracy in the experiment. There is no indication as to what was considered a stable level of performance but, based on the study results (Wier & Green, 1975, p. 1513: Figure 1), it is likely that, for durations longer than 8 ms, participants' results stabilised at or near a correct response level of 100%. For durations of 6 ms, 4 ms and 2 ms, practice was continued until the participants showed no further improvement in identification accuracy. Reported accuracy remained above 90% correct identification for all duration levels being tested. The results reported for the experiment represent the mean of the three participants' performances on the last 200 trials, registered by each participant at each of the eight segment duration levels.

Wier and Green (1975) reported that trained participants can differentiate between sounds that incorporate changes in frequency for durations as short as 1 ms per sound segment. Although this result seems to indicate that frequency changes in stimuli as short as 1 ms per sound segment can be identified as going either from 'low to high' or 'high to low' (phrases which are normally associated with changes in pitch), it is unlikely that pitch perception for the two discrete segments was the method used by participants to identify the stimuli. This issue will be discussed later in this review and will form the basis of Study 1 (see Chapter 3).

Summary.

From the review above, it is clear that identifying the pitch of short contiguous tones, such as will be examined in this thesis, should require no more than 125 – 200 ms per tone segment because all of the sound segments are similar. It is quite possible that they will require much less than this duration because of the additional time available in echoic memory. However, the minimum duration of these tone segments would have to be longer than the minimum duration for perception of a single pitch, which, in a melodic context has been found to be around 8 wave cycles (between 8ms and 34 ms in the frequency range being studied). Within these durational boundaries, there are other clues available to aid in tone duration estimations. If identifying the pitch of each tone in short sequences of sine wave tones was as easily achieved as ordering short sequences of sine wave tones, one would expect minimum durations for perception of pitch to be 10ms or below. However, this is extremely unlikely because of the known masking effects of other tones. A more realistic figure might be the 20ms, which Hirsh (1959) identified as the minimum duration required for identifying which of two sounds was presented before the other. Also, it is likely that tones in different positions within a sequence will require different minimum durations for perception of pitch. In research studying tone sequences including gaps (Watson et al., 1975), high tones were found to be more easily identified than low tones and tones immediately followed by a gap (which includes the natural gap which follows the last tone) were more easily identified than tones immediately followed by other tones.

These results can assist in formulating hypothesis creation, by using these results to make predictions for the current research. They are:

- The last tone in a sequence will be more easily identified than other tones
- High tones will be more easily identified than low tones
- Sequences that go straight up or straight down should be more easily identified than sequences that include a change of direction.
- The MDPP in short tone sequences should be between 4ms (the shortest duration for perception of pitch of a single tone (Doughty & Garner, 1947)) and 160ms (the shortest duration for identifying melodies in a continuous cyclic sequence).

Sound Cognition Research

To provide an explanation as to why the above limitations occur, the link between the perception limitations identified above and what is known of brain functioning will now be considered. The duration of beta waves, gamma waves and theta waves match the sound ordering duration groupings and masking effects identified in the research reported above and may help provide reasons for these limitations. Potential limitations to pitch perception that have not been covered by prior research are also examined and a theory relating to echoic memory is proposed. Two different methods of pitch perception are described and then related to current knowledge about minimum durations for perception of pitch.

Sources of information reported in this area include basic perceptual studies, MEG studies, EEG studies and more invasive study methods such as accessing cochlear nerves in animals and manipulating the input of cochlear implants in humans. The review begins with cyclic sounds and concludes with the primary area of focus for the current thesis, being single presentations of two very short sound segments. Finally, a study will be proposed to examine change detection strategies in very short sounds.

Theta Band Waves and Cyclic Sounds

The movement of sound information from the Basilar Membrane (BM) to the auditory cortex involves cyclic brain waves known as theta waves (Galambos, Makeig, & Talmachoff, 1981; Luo & Poeppel, 2012; Zanto, Large, Fuchs, & Kelso, 2005). The theta band of around 5 Hz (a wave every 200 ms) is used to transmit pre-processed information to higher cognitive centres such as the auditory cortex and short-term auditory memory. It is also used as the fundamental unit for long-term memory storage (Buzsáki, 1989, 2005; Klimesch, 1999).

Within the relatively slow theta wave patterns, other processing cycles occur. The most prevalent of these are the gamma waves. This form of wave has been linked to local sound processing (Gu & Liljenstrom, 2007; Luo & Poeppel, 2012; Steinschneider, Liégeois-Chauvel, & Brugge, 2011; VanRullen, Zoefel, & Ilhan, 2014). Both the theta and gamma wave bands phase lock with changes in sound signals, which means that when a sound signal starts, stops or changes (above the threshold levels for detection), gamma and theta waves reset by starting a new cycle (Galambos et al., 1981; Luo & Poeppel, 2012; Zanto et al., 2005). However, this resetting is only possible within a range of durations. The quickest a theta wave can reset is 125 ms and the longest a theta wave can last is around 250 ms. These minimum and maximum durations also match the measured capacity range of echoic memory and encompass the duration limitations seen in identifying continuous similar sound patterns (i.e. the 125 – 200ms grouping for cyclic presentations of similar sounds).

The bandwidth of the theta wave ranges from around 4 Hz to 8 Hz (250 ms to 125 ms cycle length) (Luo & Poeppel, 2012; Lütkenhöner & Poeppel, 2011) and the duration of sound information carried by the theta wave is long enough to include basic language content such as syllables. In musical terms, the most common 5 Hz theta wave relates to the duration of a semiquaver at a tempo of 75 crotchet beats per minute (i.e. five sounds per second). The theta band wave duration varies in order to track speech signal dynamics, pitch changes and other noticeable changes in sounds. Because theta waves phase-lock with new sound signals, it is likely that the duration limitations seen in identifying continuous sound patterns (125 to 200 ms per segment) are linked to theta wave patterns. The theory advanced here is that continuous cyclic sounds require segment durations at least as long as a theta wave to permit correct identification of the order of sound segments. This is because the minimum duration required to order cyclic presentations of sounds, which is 125 ms for both vowel sounds (Thomas et al., 1970) and pure tones (Thomas & Fitzgibbons, 1971), is the same duration as the minimum duration of theta waves (125 ms). Further, the range of minimum durations identified for ordering cyclic patterns of sound (125 to 200 ms) lies within the theta wave cycle duration band (125 ms to 250ms).

A complication for this theory is that there are pieces of music that are played at a faster rate than the theta band processing speed seems to allow. However, based on the research reported above, although the sounds would not blur together as they would if each sound was shorter than 50 ms, it would not be possible (for example) for a transcriber to notate a melody accurately after a single presentation at such a fast presentation rate. This is because the order of notes would not be discernable, despite each note being clearly and distinctly heard. This effect has been described by Thomas et al:

> Many of our subjects reported verbally (and in some cases wrote on their answer sheets) that their last two answers [referring to segment durations of 100 ms and 75 ms] were guesses, although they were confident that they had heard all four vowels "separately". (Thomas et al., 1970, p. 1011)

Echoic Memory. The duration range of the theta wave (125 – 200 ms) is also very close to the duration range results from echoic memory research. Echoic memory has been linked to the processing of syllables (Massaro, 1974) and closely follows the duration limitation boundary found for cyclic ordering of similar sounds (see 125 – 200 ms section heading above). The exact location of the echoic memory system within the brain is unknown and may involve multiple and diffuse areas rather than a specific location. However, the duration of echoic memory has been the subject of several studies. The theory proposed to explain this was that theta waves and echoic memory are different ways of explaining the same effect. In other words, the theta wave is the electronic trace of echoic memory. This is because both theta waves and echoic memory have very similar durational limits, and both represent first level analysis of sounds. The difference in reported durations for echoic memory (ranging from 120 to 250 ms) is likely caused by the types of studies undertaken, because some studies have explored onset/offset synchrony to discover the shortest duration in which sound order can be observed (Abel, 1972; Efron, 1970a; Efron, 1970b; Efron, 1970c), whereas other studies have examined the maximum duration at which one sound can affect the perception of another sound (Massaro, 1974).

Short Term Memory. Short term auditory memory has also been referred to as "synthesized auditory memory" (Li & Cowan, 2014) and comprises the results of the pre-cognitive processing in echoic memory. Information that reaches the higher levels of the auditory cortex, including the main pitch processing areas (i.e. the lateral half of Herschl's gyrus, the superior temporal gyrus and Planum polare (Patterson, Uppenkamp, Johnsrude, & Griffiths, 2002)), is already in the form of an abstracted or synthesised version of the original sound stimuli (Li & Cowan, 2014). Because there is no direct link between the synthesised version and the original sound vibrations, the synthesised versions can often be substantially different from the sound waves that caused them (Patterson, 1973; Patterson et al., 1992). Storage in short-term auditory memory can last somewhere between 1.5 and 30 seconds, which allows sufficient time to compare disparate sound information (Li & Cowan, 2014). Different aspects of sounds are stored in various processing areas, including the auditory cortex. The content of this short-term auditory memory system is available for conscious review as well as further processing (Janata, 2012). It is likely that the ordering of disparate sounds (see "300+ Cyclic Presentation of Sound Sequences (with stream segregation)" above) require processing in short term memory to link the sounds back into the correct temporal sequence after being preconciously segregated into different streams.

Dissimilar Sounds and Stream Segregation. Dissimilar sounds, as well as very fast or widely spaced similar sounds, are often grouped into different cognitive streams (van

Noorden, 1975). This is termed “stream segregation”, although Van Noorden used the phrase ‘sound fission’ to describe this effect. vanNoorden completed a study examining the boundary of sound fission (i.e. stream segregation) for similar sounds (such as sine wave tones) and found that, for faster tones and wider gaps, participants were unable to report all sounds in one stream. It would seem from Van Noorden’s results that the duration of a single theta wave (125 – 250ms) is not sufficient for analysis and recognition of dissimilar sounds in a cyclic pattern. This is because they require a minimum of 300 ms and up to 700 ms between sound onsets for their order to be identified. This duration is the equivalent of between two and four theta wave cycles. Thomas and Fitzgibbons (1971) postulated that the need to shift attention amongst different classes of stimuli is the reason that such long processing times are required. For these kinds of sounds, the extended use of short-term auditory memory seems necessary for successful order identification. A possible reason for this is that the cognitive grouping of results (i.e. “clumping”) is not available for dissimilar sounds. In the same way that a long string of numbers can be memorised by grouping or clumping numbers so that they can be identified by a single cue, similar sounds could be clumped to extend memory capacity. However, in the case of dissimilar sounds, this clumping may not be possible, requiring separate recognition of each sound.

Stream segregation can be caused by a number of factors. van Noorden (1975) reported that how sounds are heard (i.e. in single or multiple streams) varies, due to the duration of the sounds, the differences between the various sounds (loudness, pitch and/or timbre), the perceptual set of the listener (what the listener is trying to hear) and the listener’s level of experience. Van Noorden concluded that, generally, the faster the changes in sounds (i.e. the shorter the duration between tone onsets in a sequence) and the greater the disparity in sounds (i.e. the greater the difference in pitch, loudness and/or timbre), the more likely it is that sound fission (stream segregation) occurs.

An important assumption underpinning the current thesis, and drawn from the foregoing account, is that if a sound sequence is segregated into different streams, the sounds in each stream can be isolated and studied without interference from other sounds. This may affect the MDPP for tone in particular positions and relative pitch heights if sounds in 2-tone and 3-tone sequences can be isolated and processed separately.

Beta and Gamma Band Waves and Isolating Single Sounds

Gamma wave oscillations are most closely linked to local processing and extracting information from afferent nerves (Gu & Liljenstrom, 2007; Luo & Poeppel, 2012; Steinschneider et al., 2011; VanRullen et al., 2014). One of the primary features of the gamma wave is that every time a new sound signal occurs, gamma wave activity resets by starting a new cycle (Galambos et al., 1981; Zanto et al., 2005). The gamma waves that phase lock with sound input are restricted to a range of between 20 and 25 ms and are primarily involved in matching nerve patterns against previously learnt patterns and also calculating pitch and loudness from position, rate and intensity information (Colgin et al., 2009).

A change in gamma wave phase occurs if novel information is detected in a sound (such as a change in the frequency, timbre or loudness). When very quick changes occur regularly, beta waves emerge as sub-harmonics of the primary gamma waves (Haenschel et al., 2000; Kisley & Cornwell, 2006). Although beta waves range from 12 to 30 Hz, Haenschel et. al. identified the "slow beta" range of between 12 and 20 hz (50 ms – 83 ms per wave) as those used to gather all the available information about a rapidly changing stimulus (Haenschel et al., 2000). The 50 ms (or 20 Hz) beta wave, which is the fastest of the 'slow beta' range associated with sound processing, matches the 50 ms sound processing transition points described earlier in this review. These points include the transition from discrete sound pulses to a sense of pitch, the transition from two sounds blurring together to being able to hear two discrete sounds and changing from hearing a repetition of a sound as a reverberation to hearing it as a distinct echo; (see the section entitled "15-75 ms: the 50 ms threshold and single presentation of sound sequences"). It is assumed here that the beta wave rate of oscillation limits the speed at which changes in frequency can be individually identified. Further, the information within a single beta wave cycle is summarised and a single pitch selected as a 'best fit' from information contained in the beta wave. This theory is supported by studies that have found that, if the frequency of a sound changes more quickly than every 50 ms, the sound is reported as blurred or smeared (Joos, 1948; Schouten, 1962). This is not to say that a rough pitch is not pre-cognitively selected to represent the sound, but a sense of movement (up or down in pitch) is also included in the perception of the sound. A similar effect is found in vision, where if an object is travelling very fast, it seems blurred, but the direction of travel is known (Burr & Morgan, 1997) and the general location of the object is still available. This effect is discussed further in the "analogues, frequency change detection and very short sounds" section (below). However, the relevance to 2-tone and 3-tone sequences may be limited because, as it has already been pointed out (see above), short sequences do not seem to be affected by beta wave limitations. This is likely because beta waves do not form until multiple changes have occurred. Therefore, although not directly affecting this study, these limitations may become increasingly important as tone sequences increase from 2-tone sequences through to continuous sequences.

The masking of tones by other tones is very relevant to this thesis. The link between gamma and theta waves and masking limitations outlined by Massaro (1970) seems very strong. Sparks (1976) reflected on how the masking effect was achieved. He summarised Massaro’s theory by saying:

“In the first hypothesis, consider the possibility that pitch identifications are mediated by a perceptual processor requiring a finite time to analyze signal details. Assume that this operation can be interrupted by a second signal following closely in time. Analysis of the first signal, then, will be incomplete and an impoverished version of the signal will be transferred to a short-term store. Assume further that the duration of this "analyzing time" [i.e. the minimum time required for an observer to identify the pitch of a tone] would vary as a function of the listening experience of the observer. The data of Fig. 1 could be interpreted as supporting such a hypothesis since all psychometric functions show performance drops as IBI is reduced from 200 msec, and overall performance would appear to have a direct relation to listener experience.” Sparks (1976, p. 1349)

Massasaro’s hypothesis, as outlined by Sparks (1976) is supported by research into gamma and theta brain waves. To put it in terms of brain waves, a 20ms tone followed by a 20ms gap would have the benefit of at least one complete gamma wave cycle. For experienced musicians, and with wide intervals, that is probably all that is needed for accurate pitch recognition. When the subsequent masking tone arrives, the second 25ms gamma wave, which (had no other sound arrived) would have been utilised to process further the test tone, dumps its information and starts processing the masking tone. However, contrary to Massaro’s theory, the information gained in the first gamma wave cycle is still retained and progresses up the auditory nerve via a theta wave, which eventually brings it to the central processing areas. At this point information arrives in a pre-processed synthesised form and a decision is made as to whether it matches the previously memorised ‘high’ or ‘low’ tone.

A point if interest in relation to masking research was the use of 500ms maskers. At this duration, the maskers last at least two theta waves. More masking was found when 500ms maskers were used compared to 20ms and 10ms maskers. The implication of this is that the interference of the continuing tone made it more difficult to decide the pitch of the target tone.

Sparks found that novice performance showed much more masking effect than the performance of trained musicians, but the former became similar to the trained musicians after considerable training (approximately 200 hours). This result supports methods used in later research by Massaro (1975) and Massaro and Idson (1978), where identification difficulty was modified, depending upon the skill of the participant, to minimise floor and ceiling effects. The implications for research in this thesis is that in order to gain a true indication of minimum durations for perception of pitch, either highly trained musicians should be used or considerable time should be expended training novice participants.

Masking Effect Differences Caused by Relative Pitch and Position of Tones

All studies examined in this literature review have found that a sound placed after a target tone affects identification of that tone. Some researchers have also reported that sounds occurring before a test tone affect identification of that tone. In addition, some researchers have reported that the relative pitch of tones interfering with a test tone did not change identification difficulty, whereas other researchers have found that higher tones (relative to the other tones in the sequence) were less affected by other tones than were lower tones. This issue will be explored in Chapters 4 and 5. Based on masking research; it is theorised that:

- tones occurring at the end of sequences (or after a gap) will be less influenced by masking than tones followed by another tone,

- lower pitch tones (relative to the other tones) will require longer durations for identification compared to higher tones,
- high tones (relative to other tones in the sequence) will remain relatively unaffected by the masking effect of other tones.

Analogues, Frequency Change Detection and Very Short Sounds

It seems that short duration sound segments are subject to more intense pitch extraction processes. Evidence for this includes the differently shaped encephalograph (EEG) wave cycles that short segments of sound evoke, compared to continuous sounds using the Event-Related Potential (ERP) procedure. These short duration wave cycles include the N100 and P200 wave peaks, which have been related to sound processing activity. If changes come particularly quickly, the N100 and P200 waves disappear, indicating that the amount of processing done on continuous sounds is less comprehensive than for short sound segments. This reduction in processing would be necessary in order to accommodate continuous flows of new information (Lütkenhöner & Poeppel, 2011). In terms of short sound research, echoic memory for sounds processed during a theta wave may account for the ability to identify pitch changes in very short tones, even for durations shorter than 50 ms per segment. This theory therefore suggests that these very short sounds can be held in echoic memory and processed multiple times. For example, two 30 ms sound segments, lasting a total of 60 ms, could be held in echoic memory for around 240 ms, according to Massaro’s (1974) research. This would allow around three times the processing time compared to that available for continuous sounds. If the exact location of the change in frequency can be located, it would enable the identification of each sound segment individually, regardless of its duration. This is supported by research that has found that presentations of two or three sound segments in isolation can be identified at much shorter segment durations than cyclic sounds (Hirsh, 1959;

van Noorden, 1975; Wier & Green, 1975). If this is the case, it stands to reason that the mechanism that limits continuous sound perception and identification does not apply to identifying the order (and possibly the pitch) of two or three sound segments presented in isolation. The aim of the research reported in Chapters 4 and 5 of this thesis was to identify the shortest duration at which it is possible to identify the pitch of short sequences of contiguous tones of different frequencies.

Frequency Change Detection vs Sound Analogue Comparison. At durations below the minimum duration required for perception of the pitch of a single tone, it is still possible to discriminate between different sound orders (Nickerson & Freeman, 1974; Wier & Green, 1975). However, there is doubt as to whether the results relating to research into changes in very short sounds, in the range of 1 – 4 ms, are a product of any pitch or sound order identification process at all.

This question will be examined from three directions. First, the threshold of pitch identification will be examined, to consider if it is even possible that steady state pitch information can be used at such short durations. Secondly, the relationship between frequency change and identification of stimuli will be examined, to look for a more likely identification method. Thirdly, the possibility that analogues of sound sequences are being used to make these identifications will be examined and a study will be undertaken to resolve this issue.

Pitch Perception Models and Very Short Sounds. Mechanical sound waves pressing on the Basilar Membrane (BM) create nerve pulses that travel along the auditory nerve. Each nerve pulse provides three pieces of information. These are: (i) the stimulated nerve's physical position along the BM; (ii) the nerve impulse's temporal position, relative to other nerve firings; and (iii) the overall rate of firing of each nerve. Two theories of pitch perception have been developed to describe how this information is utilised to extract pitch information. The theory relating to the stimulated nerve's position along the BM is known as the "Place" theory (Corwin, 2009). The theory relating to the number of times nerves fire and the position in the wave cycle at which they fire is known as the "Rate" theory (depts.washington.edu, 2013). Regarding the pitch identification method explained by the Place theory (i.e. pitch identificaiton from the position of stimulus on the BM), the frequency spread caused by sounds shorter than 4 ms would make it impossible to judge the pitch of very short sounds accurately using this method (Winkel, 1967). Regarding the pitch identification method explained by the Rate theory (i.e. the timing of nerve firings), a minimum of 4 cycles of any specified frequency is required to gain any sense of the pitch from a sound (Robinson & Patterson, 1995). Weir and Green's 1 ms stimuli segments only create one wave cycle for the 1000 Hz tone and two wave cycles for the 2000 Hz tone. Clearly, neither the Place theory or the Rate theory would indicate that pitch identification was available at 1ms or for two wave cycles. In confirmation of this conclusion, one or two wave cycles has been found to be insufficient to create a sense of pitch (Robinson & Patterson, 1995).

Rather than one of the above theories providing sufficient explanation for pitch identification for all circumstances, most writers have agreed that both forms of pitch perception (i.e. both 'place' and 'rate' theories) can be utilised, depending on frequency constraints (De Cheveigne, 2005; Evans, 1978; Fearn, Carter, & Wolfe, 1999; Greenberg, Marsh, Brown, & Smith, 1987; Krumbholz et al., 2003; Pollack, 1990). These authors have suggested that temporal matching (i.e. 'rate' theory) is used exclusively for analysing frequencies lower than around 150 Hz (Miller, 1948) but position mapping (i.e. 'place' theory) is used exclusively for higher frequencies (above 2000 - 4000 Hz) (depts.washington.edu, 2013). Between 150 Hz and 4000 Hz (the frequency range of all the research reported in this thesis), it is likely that a combination of both forms of pitch perception is used (Greenberg et al., 1987). However, combining both strategies would be of little or no assistance. This is because there is insufficient information, using either of the two perception strategies, to allow improvement in perception by combining the two sources. Therefore, it can be concluded that pitch information has not been used to analyse sounds in those studies presented with durations within the 1 – 4 ms range. In addition to this finding, although some of the other results categorised in this review in the 1 – 10 ms grouping have included minimum durations required for pitch recognition in the range 5 - 10 ms (Nickerson & Freeman, 1974; Warren, 1974; Warren et al., 1990; Warren et al., 1991), evidence from the wider body of research -- and comments from the above researchers on their own studies -- suggests that participants in these studies were also using something other than steady state pitch analysis, in order to complete the tasks that they encountered.

Frequency Change Information. It is proposed here that an alternative to the use of steady state pitch identification is to use frequency change information. According to this theory, the mind has an inherent capacity to detect change and it is possible to identify direction of change without identifying any other features of the stimulus (Molholm, Martinez, Ritter, Javitt, & Foxe, 2005). Semal and Demany (2006) found that the ability to identify the direction in a change of frequency between two sound stimuli is independent of the ability to identify the pitch of each frequency. Zatorre and Schönwiesner (2011) stated that it is likely that change in pitch direction uses different sets of processors than those used for pitch detection.

In a much earlier, but comprehensive and seemingly conclusive study, Whitfield and Evans (1965) examined neuronal responses in 70 un-anaesthetised cats. They found that some neurones did not respond to steady state frequencies but, instead, responded to frequency changes in a particular direction. In other words, some auditory neurones did not fire when the frequency stayed the same, did not fire when frequency changed to a higher level, but only fired when the frequency changed downwards. This seems strong evidence that pitch identification is not only unnecessary for frequency change identification, but immaterial to this identification. Whitfield and Evans stated that there exists in the brains of cats many more neurones that respond solely to pitch change of a specific direction than there are neurones that respond to steady state frequency. This response to frequency change has been noted in more recent studies and labelled 'mismatch negativity' (MMN) (Molholm et al., 2005). Although Molholm et al. acknowledged that MMN indicates that a change has occurred and provides some information as to nature of that change, the specific directionality of the response found in Whitfield and Evans' study seems to have been overlooked.

Sound Analogue Comparison. It seems clear, therefore, that the identification of steady state pitch is not necessary to identify the direction of a frequency change. However, it remains to be determined whether participants in studies that require the detection of changes in extremely short sounds can use frequency change detection at all. Two possibilities present themselves as potential cognitive strategies for the identification of these very fast frequency changes: frequency change detection and sound analogue comparison. If analogues are being used, then it may not be necessary for frequency change direction information to be accessible to the participant at all. As described above, analogues are holistic representations of a sound sequence consisting of a set of distinguishing features of that sequence. Although differences in the order of nerve firings are discernible (possibly down to the order in which nerve firings arrive at the processing centres), the nerve firings may not be perceived as frequency changes. Van Noorden stated: “differences in order can be detected in tone sequences which are so fast that the order itself can no longer be identified” (1975, p. 65).

Conclusions and Introduction to the First Study

A way of testing if sound analogue comparison or frequency change detection is being used to differentiate between stimuli, is to change some distinguishing features of stimuli, in addition to frequency, and test whether this disrupts identification. *If frequency change information is being used to identify the stimuli, changing the timbre of the sequence will have no significant effect on the result. However, if analogues of sounds are being used to differentiate between stimuli, then changing the timbre of the sequences (e.g. presenting the sequence first using a sine wave timbre and then using a ‘clarinet’ timbre) should disrupt this process*. Given that pitch and frequency change information are clearly available at longer segment durations, but harder (or impossible) to access at shorter durations, it is hypothesised that changing the timbre between frequency change sequences significantly and increasingly disrupts identification of changes in contiguous frequencies as stimuli segment durations are reduced. A study testing this hypothesis will be reported in Chapter 3.

Pitch Identification in Contiguous Sequences. Regarding pitch identification, the theory presented here is that it requires longer durations to identify the pitch of each tone in a short sequence of tones, than it takes to identify the order of a similar sequence of tones; and therefore, the MDPP cannot be established using sound ordering studies. This theory can be tested, first, by establishing the minimum duration required for the ordering of sounds in different conditions, by examining the research reports from sound ordering studies (see below) and then, secondly, by creating a test that uses similar tone sequences but ensures pitch perception is used to identify each tone. The results from these studies could then be compared with sound ordering studies, to see if sound ordering or pitch perception requires longer segment duration.

In the case of very short contiguous tones, one method for ensuring that tone sequences are identified as strings rather than analogues, is to require participants to match each segment of a pure tone (i.e. sine wave) sequence correctly against single tones created using a different timbre and duration. This would disrupt analogue matching because there would be many differences between a pure tone contiguous sequence and a single tone created using a different timbre (e.g. a piano tone).

In light of the foregoing review, the studies reported in the following chapters have been predicated on assumptions that:

1. identifying the pitch of contiguous tones requires longer exposure durations than identifying the pitch of isolated tones.
2. it is not possible to identify accurately the pitch of a tone at very short (below 4ms) segment durations.

Chapter 3: Study 1. Adaptive Methods for Frequency Change Detection

The minimum duration for perception of pitch (MDPP) in short contiguous sequences has not been established by pitch identification, but it had previously been assumed that the ability to order short sine wave tone sequences implies that the pitch of each segment can be identified (Robinson, 1995). From a review of literature, it was established that the minimum duration for ordering tones was 1 ms for 2-tone sequences (Wier & Green, 1975), and between 2ms to 7ms for 3-tone sequences (Divenyi & Hirsh, 1974). From this, it was immediately obvious that pitch identification was not being used to order these sequences because the minimum duration required to establish a sense of pitch is 4ms for a 1000 Hz tone, extending to 18ms for a 125 Hz tone (Doughty & Garner, 1947). Therefore, it has been concluded that minimum durations for perception of pitch in short contiguous sequences has not been established by any means. However, the strategy used to order very short 2-tone and 3-tone sequences was unclear, so further research was undertaken to establish the basis for identification of ordering of sine wave stimuli. Several studies have established that it is not necessary to identify steady state pitch in order to identify the direction of a frequency change in a sound consisting of two tones played at different frequencies (Evans, 1978; Molholm et al., 2005; Semal & Demany, 2006; Zatorre & Schönwiesner, 2011). This is because different cognitive processes and neural pathways are used for detection of changes in frequency, compared to detection of steady state pitch (Whitfield & Evans, 1965; Zatorre & Schönwiesner, 2011). However, it remains to be determined whether studies requiring participants to differentiate between two extremely short frequency change stimuli (e.g. the shortest durations successfully identified in Wier and Green (1975)) use any form of frequency analysis processing to differentiate between stimuli. An alternative cognitive strategy to analysing the direction of frequency change is to make sound analogue comparisons and use these to differentiate stimuli. Analogues, as defined by Neisser (1967), are holistic representations of a *full sound sequence*, consisting of a set of the distinguishing features of that sequence. When a sound or sequence of sounds is heard, these distinguishing features are pattern-matched against cognitive templates that have been established through prior exposure to similar sounds (Hopfield, 1995). In this way successful identification is achieved through pattern matching, rather than by analysis of pitch direction. This effect has been identified as an ‘integration mechanism’ or ‘gestalt’ by Massaro (1975), to describe an anomaly in his data where contiguous tones were more easily identified than tones spaced slightly apart. Massaro commented that subjects could be slightly more accurate by comparing gestalt units in the first and second presentation than comparing stimuli where a gestalt is less likely to form. Sound analogue comparison (i.e. ‘gestalt’ comparison) can be used when there is only minimal information available and would therefore be the last strategy to be lost due to decreasing information caused by shortening stimuli durations. If analogues are being used, then it may not be necessary for information about the direction of frequency change to be accessible to the participant at all. This suggestion has been supported by Van Noorden, who stated: “differences in order can be detected in tone sequences which are so fast that the order itself can no longer be identified” (van Noorden, 1975, p. 65). The perception strategies used to detect differences in order in these very short sounds have yet to be established by research, even though it is accepted that this form of stimuli can be differentiated.

Wier and Green (1975) demonstrated that trained participants can differentiate between sounds that incorporate changes in frequency for durations as short as 1 ms per tone duration. However, Wier and Green did not identify the cognitive strategy used to discern these differences, stating that their test “does not specify the basis of the discrimination”

(Wier & Green, 1975, p. 1515). Of the three possible strategies for identifying frequency changes (pitch identification, frequency change detection and sound analogue comparison), pitch identification can be immediately excluded from the shortest durations identified in Wier and Green's study. As is clear from the review in the previous chapter, a sense of pitch cannot be extracted from a single tone shorter than 4 ms in duration (Doughty & Garner, 1947) and therefore it was clearly not possible for pitch change detection to be used for identifying the shortest stimuli successfully identified in Wier and Green's (1975) study. This leaves frequency change detection and sound analogue comparison as possible candidates for this form of frequency discrimination. The study now to be described in this chapter was designed to exclude the strategy of sound analogue comparison.

For participants to reach successful identification in the area of very short sounds (i.e. below 10 ms per tone), there is clear evidence that very short sound differentiation is subject to dramatic improvement through learning (Green, 1973; Nickerson & Freeman, 1974; Patterson & Green, 1970; Wier & Green, 1975). It has also been acknowledged that improvement through learning takes place in many experiments from other fields: "Most experimenters acknowledge that absolute stability of the underlying psychometric function generally is not a realistic assumption, since subjects typically experience some perceptual learning during the course of an experiment, reducing the true threshold" (Leek, 2001, p. 1288). It is the case therefore that researchers studying very short sound differentiation consistently use extended practice before undertaking their studies (Divenyi & Hirsh, 1974; Green, 1973; Nickerson & Freeman, 1974; Patterson & Green, 1970; Warren, 1974; Wier & Green, 1975). A review of this field using key words including "pitch change" "pitch perception", "frequency change" "frequency detection" and "pitch duration", amongst others, has found no study that did not use extended practice in some form where participants were able to identify stimuli at or shorter than an exposure duration of 10 ms per tone. Therefore, the study to follow incorporated extended practice within its design.

It is also worth noting that the studies reviewed above have all used very small numbers of participants: between one and seven participants. This practice reflects the lengthy time demands that have been required of participants in this kind of research. Although such small participant numbers are insufficient to draw firm conclusions about perceptual capacities in the general population, a small sample has been regarded as sufficient to establish minimum durations for the various forms of perception that were tested in these studies.

A way of identifying whether frequency change detection or sound analogue comparison is used to identify changes in very short sounds, such as those used in Wier and Green's (1975) study, is now described. The key to achieving this is to disrupt sound analogue perception by varying an aspect of the stimuli other than frequency and test to see if this new independent variable negatively affects perception of frequency change stimuli, compared to a baseline study of non-varied stimuli. Thus, if only frequency change information is being used to identify the stimuli, randomly changing the timbre between stimuli should not affect the minimum perception duration. However, if analogues of sounds are being used to detect frequency change, then changing the timbre of the sequence, without changing the frequency information, should make it far more difficult to identify stimuli at very short durations than the baseline condition of non-varied stimuli. This is because sound analogue comparison would no longer be possible to use for stimuli identification in this condition.

Because the shortest duration at which two contiguous tones with differing frequencies can be identified has been established by Wier and Green's (1975) study, their method was used as a basis for the design of the current study. Weir and Green used 1,000 Hz and 2,000 Hz sine wave tones. In the study to be described now, these same sine wave stimuli were recreated and used for comparison with 1,000 Hz and 2,000 Hz tones that were created from 86 different timbres. For example: one stimulus was created using a 'flute' 1000 Hz tone followed by a 'flute' 2000 Hz tone. Another stimulus was created using a 'clarinet' 1000 Hz tone followed by a 'clarinet' 2000 Hz tone. Thus, 87 sets of stimuli were created (one sine wave set plus 86 sets of other timbres). These sets consisted of stimuli of a single timbre with 1,000 Hz and 2,000 Hz segments. Segment durations ranged from 128ms to 1ms, with durations halving at each progressive step. For each segment duration, two stimuli were created: one starting with the 1000 Hz tone, followed by the 2000 Hz tone (the 'up' stimulus) and one starting with the 2000 Hz tone, followed by the 1000 Hz tone (the 'down' stimulus). It is worth noting that at 1ms, a 1,000 Hz tone lasts for one wave cycle and a 2,000 Hz tone lasts for two wave cycles. Because it is impossible to establish a 'frequency' with just one wave cycle (for the 1,000 Hz tone), it is difficult to accept that the participants in Wier and Green's (1975) study used a frequency analysis strategy to differentiate between the1ms stimuli. It is also worth noting that the 1ms duration becomes a natural floor level for stimuli identification. It is impossible to present 1000 Hz stimuli at shorter durations than 1ms per segment because 1ms is the duration of a single 1000 Hz sound wave.

Given that frequency change information is clearly available at longer tone durations, but assuming that it is harder (or impossible) to access at shorter durations, it was hypothesised that: *the minimum duration for identifying change in direction (either low-high or high-low) in stimuli consisting of two contiguous tones with different frequencies, is significantly longer when each stimulus is created using one of 86 different timbres, compared to when a sine wave timbre is used to create all stimuli.*

Additional Considerations

Prior consideration was given to other possible reasons for differences in identification success rates between the single sine wave timbre condition and the randomly selected (varied) timbre condition. It is possible that it might be harder to identify the pitch of tones using different timbres such as flute, clarinet, guitar and organ than using sine wave tones because the different timbres selected for the varied timbre study might be individually harder to identify than the sine wave timbre used in Wier and Green's study. It was therefore decided to include both a sine wave single timbre set and a series of single timbre sets, randomly selected from the pool of 86 timbres used in the 'varied timbre' condition, for the individual timbre condition in the study. It was therefore expected that: *the minimum duration for identifying change in direction (either low-high or high-low) in stimuli consisting of two contiguous tones with different frequencies, would not be significantly longer when a single timbre randomly selected from a pool of timbres is used to create stimuli (condition 1B), compared to when a single sine wave timbre is used to create stimuli (condition 1A).* If the two conditions were found to be significantly different, with the randomly selected timbres being harder to identify than the sine waves timbre, it would not be logically possible to test the hypothesis for this study using this method. This is because any difference found between the single timbre condition and the varied timbre condition could be caused by the increased difficulty of identifying stimuli created using the 86 different timbres. However, should the sine wave timbre condition and the randomly selected individual timbre be found to be not significantly different, then both conditions can be grouped together and used without differentiation as the 'single timbre' condition for the purpose of comparing to the 'varied timbre' condition in the study proper.

In addition to the effect of learning, the issue of adaptation can also affect results. This issue seems particularly relevant to the current research, because it has been theorised that three different methods for identifying a frequency change are available to participants. Should some participants be able to adapt to use different methods while others cannot adapt, it can be inferred that, not only has learning taken place, but a series of adaptations has enabled some participants to identify stimuli at much shorter durations than other participants.

Method

Participants

Participants were 20 university students (18-24 years; 8 males) who were enrolled in first-year Psychology at the University of Adelaide. Participants were not paid but received course credits for participation. There were no selection criteria and participants were not screened for hearing acuity. Participants' experience level was unknown and no prior training was provided.

Design

A modified adaptive test model was developed whereby progress through the duration levels was slowed (compared to the standard one up, two down model) in order to maximise learning at each level[3]. Progression to a shorter duration was determined by the participant correctly identifying all stimuli in a block of five trials. If one error was made (indicating

---

[3] Owing to limitations of participant availability, it was not possible to include extended practice sessions as part of preparation for the study, as had been used by previous researchers. As an alternative, it was decided to use the study to maximise learning in the most efficient way possible. It was important to ensure that some form of training took place to assist participants to reach the very short durations that were the focus of this study.

80% success in that block) the same tone duration level was presented again. If the participant made two errors in a block of 5 (indicating less than 75% correct), the participant was presented with longer tone durations. Feedback was given at the end of each trial, so the participants knew when they were correctly identifying the stimuli.

At each tone duration level there were two independent variables: the type of stimuli (either varied timbres or single timbre) and the stimuli order (low-high followed by high-low, or high-low followed by low-high). The dependent variable was correct identification of the stimuli. Constants were the sound pressure level (70 dB) and the frequencies of the stimuli (1000 Hz and 2000 Hz). Participants were randomly allocated to two different conditions. Condition 1 consisted of a batch of single timbre stimuli followed by a batch of varied timbre stimuli. Condition 2 reversed the order of condition 1 (i.e. a varied timbre batch followed by a single timbre batch). This was done to balance the effects of learning, adaptation, and fatigue across the two stimuli conditions.

Testing for Differences Between Single Timbre Identification Rates. To test if rates for identification success between the single sine wave timbre condition and the randomly selected single timbre condition were significantly different, half of the participants were presented with sine wave timbre sets as the single timbre condition and half were presented with a single timbre randomly selected from the 86 timbre sets as the single timbre condition. For the randomly selected single timbre condition, each participant was given a different randomly selected single timbre set. Ten such randomly selected timbre sets were used in this study.

Materials

Stimuli Creation. Stimuli were created using the following description by Wier and Green (1975). Each stimulus consisted of two contiguous tones of equal duration: a 1000 Hz (low) tone and a 2000 Hz (high) tone. At each duration level, and for each timbre, two stimulus pairs were created. Each pair consisted of two stimuli, each containing two contiguous tones of differing frequencies, divided by a 500 ms gap. Stimulus pairs were arranged as follows:

Pair 1: (1000 Hz tone – 2000 Hz tone) – 500 ms gap - (2000 Hz tone – 1000 Hz tone)

Pair 2: (2000 Hz tone – 1000 Hz tone) – 500 ms gap - (1000 Hz tone – 2000 Hz tone)

In terms of a verbal description; Pair 1 goes up in pitch then down; Pair 2 goes down in pitch then up.

The stimulus pairs described above were produced for each of 87 different timbres (86 timbres for the varied timbre condition plus the sine wave timbre) at each of the following tone durations: 128ms, 64ms, 32ms, 16ms, 8ms, 4ms 2ms and 1 ms; the durations used by Wier and Green (1975). In short, stimulus durations were calculated by halving the previous (longer) tone durations; this method produces linear steps on a logarithmic scale. This is an appropriate step selection method for studying the duration required for accurate perception because duration perception has been found to be logarithmic in nature (Amenedo & Escera, 2000). In all 1,248 stimuli were created. In order to report the data as a linear function rather than a logarithmic curve, the durations were converted to base 2 logarithms. Thus, in $base_2$ 128ms = 7 (i.e. $2^7 = 128$), 64ms= 6, 32ms = 5, 16ms = 4, 8ms =3, 4ms = 2, 2ms = 1 and 1 ms = 0. An alpha level of $p \leq .05$ was set for all statistical tests.

Apparatus. Individual stimuli were created using an IBM compatible computer running the sound recording and creation software “Sonor” (Cakewalk, 2009). Sine waves for the stimuli were created using the program “Audacity” (FreeSoftwareFoundation, 1991) and then imported into Sonor for editing. Apart from the sine wave timbre, the 86 other timbres were created with a Roland XP-50 synthesiser using the on-board General Midi sounds. The keyboard was recorded using Sonor through a Conexand CX20671 sound input device on a Toshiba Satellite L630 computer. All stimuli at all durations were started and ended at zero crossing. Four IBM compatible computers running the presentation program "Inquisit 4" (Millisecond, 2014) were used to present the stimuli pairs to participants and record their responses (See Appendix 3 for the source code redone with JavaScript). The participants listened to the stimuli pairs through Sennheiser HD 518 headphones and responded by clicking one of two buttons representing the two alternative choices. The sound presentation levels were tested using a RadioShack analogue 33-4050 sound meter. Sounds were presented at 70 Db SPL.

Procedure

Procedure was created using the Inquisit (Millisecond, 2014) computer program. Each participant completed 600 trials consisting of blocks of five trials in six batches of 20 blocks. The large number of stimuli pairs presented to each participant was necessary to ensure that a sufficient number of stimuli pairs were identified at the shortest duration levels. Each batch of trials consisted exclusively of either single timbre trials or varied timbre between trails.

Participants were initially given safety instructions and introduced to the general procedure. They were told that the program they would use included more specific instructions, but they were verbally instructed that the purpose of the study was to judge if the sound went up then down (1000 Hz-2000Hz followed by 2000 Hz-1000Hz) or down then up and to respond with a button press. A visual representation of these two options was presented on the buttons (see Figure 3.1) including the text: "Low-high First" and "Low-high Second" to match more closely Weir and Green's instructions, where subjects were asked to identify in which interval the stimulus went from a lower to a higher frequency.

Before commencing data collection, participants were presented with an instruction screen on the computer followed by a practice screen. Participants read the instructions and then went on to the practice screen. During the practice session, blocks of five sine wave single timbre trials were presented to them at the tone duration of 128 ms. When all five trials in a block were correctly identified, participants exited the introduction section and continued to the study proper.

Figure 3. 1

Response Selection Buttons.

Note: Participants were asked to press the button corresponding to the stimuli pairs presented. “Low-high First” meant the sound went up then down (1000 Hz-2000Hz followed by 2000 Hz-1000Hz). “Low-high Second” meant the opposite.

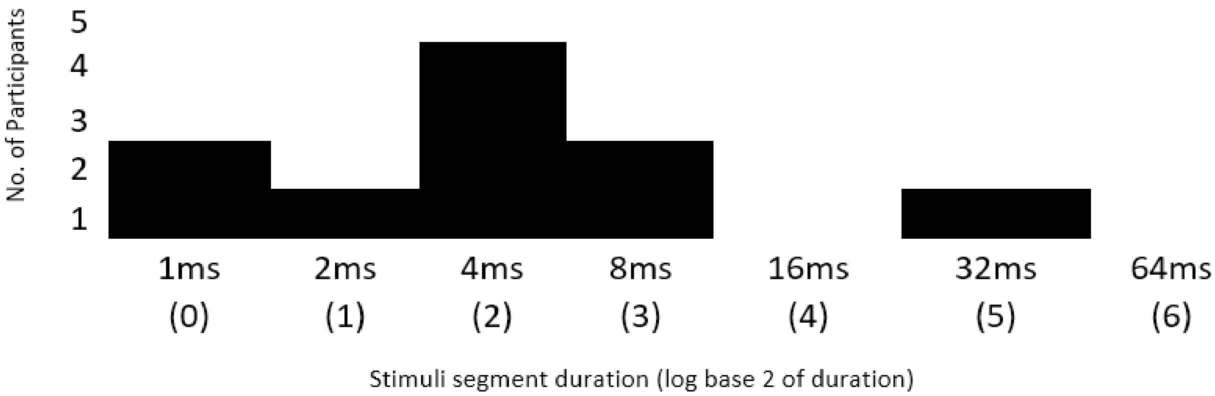
Using a 2-alternatives forced-choice procedure, stimuli pairs were presented, starting at the 128 ms per tone level. The stimuli pairs were presented in blocks of five. If all five stimuli were correctly identified, stimuli with shorter segment durations were presented. If one error was made, stimuli of the same duration were presented. If two errors were made in any block of five, stimuli of a longer duration were presented. If two errors were made at the 128ms level (the longest segments in the study), the 128ms stimuli were presented again. If all five stimuli were correctly identified at the 1ms level (the shortest segment duration in the study), the 1ms stimuli were presented again. Stimuli presentations continued until all 600 trials (300 for each condition) had been completed. This took participants around an hour to complete. The threshold duration for each participant was taken as the shortest duration that the participant successfully completed four out of the five stimuli in a single block.

Results

First, the two single timbre conditions (condition 1A: sine wave. condition 1B: random single timbre) were tested for difference (see Figures 3.2 and 3.3).

Figure 3. 2

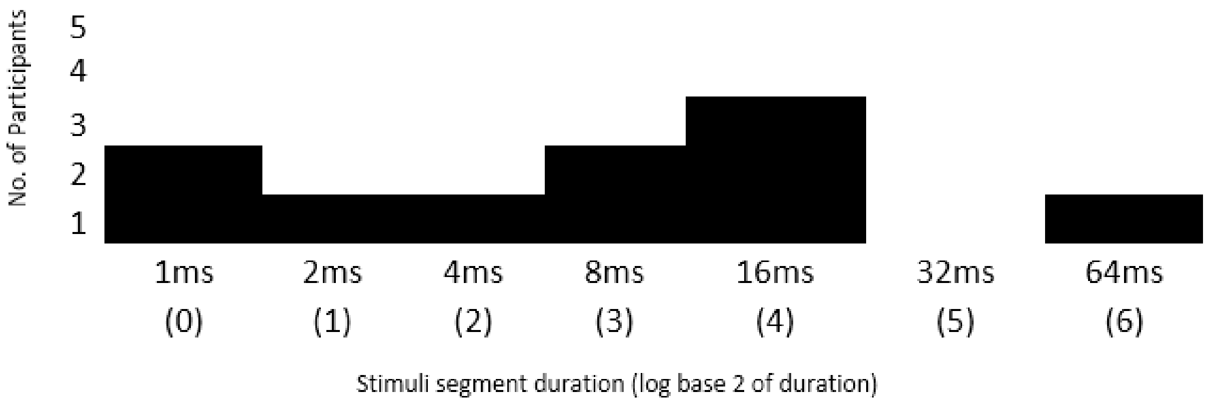
Histogram of Randomly Selected (non-sine wave) Single Timbre Stimuli (condition 1A).



Note: Overall mean (for log base 2): 2.0. Standard deviation 1.41.

Figure 3. 3

Histogram of Sine wave Single Timbre Stimuli (Condition 1B).



Note: Overall mean (for log base 2): 2.7. Standard deviation 1.95.

A one-tailed t-test (independent samples assuming unequal variance), found no significant difference between the two single timbre conditions ($t = 1.03$, $df = 13$, $p = .32$, n.s.). Therefore, this result supported the expectation that the minimum duration for identifying change in direction (up or down) in stimuli consisting of two contiguous tones with different frequencies, is not significantly longer when a single timbre, randomly selected from a pool of timbres, is used to create stimuli (condition 1B), compared to when a single sine wave timbre is used to create stimuli (condition 1A). This result is also supported by Patterson et al. (1983), who found that the pitch of complex tones, such as those used for this study in the varied timbre condition, were identifiable at shorter durations than sinusoidal tones.

The results of the main study were then examined. The shortest duration that participants were able to reach and achieve 75% accuracy (for each of the two conditions), using the modified adaptive step model, is presented in Table 3.1. The results (for log base2 milliseconds) were tested for statistical significance using a one-tailed paired t-test; $t = 3.22$, $df = 19$, $p < .01$, effect size = 0.72; 95% confidence level for difference in minimum identification duration between the two conditions 0.218 to 1.205, mean difference = 0.85. (log base 2). (i.e. 1.6ms to 8.4ms, mean difference = 5ms [4])

This result therefore supported the hypothesis that the minimum duration for identifying change in direction (up or down), in stimuli consisting of two contiguous tones with different frequencies, is significantly longer when different timbres are used to create stimuli, compared to when the same timbre is used to create stimuli. It should also be noted

[4] Durations in milliseconds have been added in order to ensure the minimum identification duration is clear.

that none of the participants was able to reach the 1ms per tone level in the varied timbre condition.

Table 3. 1

Minimum Duration for Identification of Stimuli for Participants in Varied timbre and Single Timbre Stimuli Conditions (matched pairs)

Participant Id No.	Varied timbre Condition ms(log base 2)	Single Timbre Condition ms(log base 2)
1	16ms (4)	16ms (4)
2	4ms (2)	1ms (0)
3	64ms (6)	64ms (6)
4	8ms (3)	4ms (2)
5	16ms (4)	2ms (1)
6	8ms (3)	8ms (3)
7	8ms (3)	4ms (2)
8	8ms (3)	4ms (2)
9	4ms (2)	1ms (0)
10	4ms (2)	4ms (2)
11	16ms (4)	16ms (4)
12	4ms (2)	8ms (3)
13	32ms (5)	8ms (3)
14	2ms (1)	4ms (2)
15	16ms (4)	8ms (3)
16	8ms (3)	1ms (0)
17	4ms (2)	1ms (0)
18	2ms (1)	2ms (1)
19	16ms (4)	16ms (4)
20	64ms (6)	32ms (5)
Mean log2 (SD)	3.2 (1.4)	2.35 (1.7)

Discussion

This study has found that the minimum duration for identifying change in direction (up or down) in stimuli consisting of two contiguous tones with different frequencies, is significantly longer when different timbres are used to create stimuli, compared to when the same timbre is used. It can therefore be concluded that the reason for the increase in duration required for successful identification of varied timbre stimuli is because frequency change detection becomes unavailable for identifying stimuli at very short durations. In addition to this conclusion, because no varied timbre stimuli were able to be successfully identified at the 1 ms per tone duration, it was concluded that no method other than sound analogue detection was available at the 1ms per tone duration. Therefore, the minimum duration at which it is possible to detect the direction of a change in frequency is 2ms per tone, for 1000 Hz and 2000 Hz frequencies. This result is supported by observations during the trial period, where some participants reported that the 1 ms per tone stimuli sounded like a single ‘click’, rather than a sound with two segments.

The number of participants who failed to reach the shorter durations was a concern in this study. Of the 20 participants, only nine were able to identify any stimuli at the 2 ms level and only four were able to identify single timbre stimuli at the 1 ms per tone level. There were three participants who were unable to identify any stimuli below even the 16 ms per tone duration. In searching for reasons as to why some participants had such trouble identifying stimuli at the longer durations, two different lines of reasoning were developed.

Firstly, it was noted that Schneider et al. (2005) had found that two different strategies are normally used to detect changes in frequency and that participants in their study demonstrated a cognitive preference for one form or the other. The different preferences have been described as “synthetic” listening and “spectral” or “analytic” listening (Terhardt, 1974). Synthetic listening involves using the overall sound to make a pitch judgement, including making use of prior experience of pitch change recognition. Spectral listening, in contrast, involves analysing individual components of a sound and basing pitch direction judgements on this information (Schneider et al., 2005; Terhardt, 1974). This effect was established through a study reported by Schneider et al. (2005). Participants were required to listen to stimuli consisting of tones that had the fundament (f_0) removed. The tones were created in such a way that the fundamental tone (had it been present) would have moved in one direction, but the harmonics that were presented to participants moved predominantly in the opposite direction. Participants who favoured synthetic listening identified the pitch change direction predominantly in the direction of the (missing) fundamental tone movement, whereas participants who favoured spectral listening identified the pitch direction as moving in the opposite direction.

Synthetic listening seems to align with what has been describe in the introduction as ‘sound analogue’ analysis, whereas spectral listening seems to align with frequency change detection. It is plausible that participants who exhibit a preference for synthetic listening would have more trouble with varied timbre stimuli. This being so, it is possible that, in this study, participants who had trouble identifying the very short duration stimuli preferred synthetic listening.

However, a second line of reasoning for the high number of participants unable to identify the shorter stimuli has been drawn from research into brain waves. There were two participants in the study (participants 3 and 20) who were not able to identify stimuli shorter than 64ms in one or both conditions (both conditions for participant 3 and 64ms in the varied timbre condition and 32ms in the single timbre condition for participant 20). This result could be explained by an inability to adopt a more intense processing strategy than the strategy that was initially adopted.

As identified in the literature review (see Chapter 2), when each tone in a sequence lasts longer than the minimum duration for a theta wave (125ms), the tones are processed separately, and a single pitch result is forwarded to the auditory cortex (Thomas & Fitzgibbons, 1971). This makes it relatively simple to use the common method of identifying the pitch of each tone and then deciding if the second pitch is higher or lower than the first pitch. As durations shorten, both tones occur during the length of a single theta wave and must be analysed more thoroughly in order to separate the pitch of each of the two tones. This is possible for isolated short duration sounds because interference from other sounds does not occur. This means more time can be spent processing the two tones in each stimulus (Lütkenhöner & Poeppel, 2011). It is therefore possible that those participants, who were unable to identify stimuli at durations shorter than 64ms per tone, were either unused to processing sounds more intensely (and thus made many mistakes), or were unable to do this form of intense processing because of cognitive limitations. This is speculative but plausible; and in either case, these participants would consequently be unable to identify stimuli effectively at shorter durations.

As durations decrease to below 50ms per tone, the two separate tones in each stimulus appear smeared (Joos, 1948; Schouten, 1962), and it becomes increasingly more difficult to identify the pitch of each tone. However, even below 50ms, each sound in the pair can still be phased locked to gamma waves. This means it is still possible to process both sounds separately. As explained in the literature review, one of the primary features of the gamma wave is that every time a new sound signal occurs, gamma wave activity resets by starting a new cycle (Galambos et al., 1981; Zanto et al., 2005). The gamma waves that phase lock with sound input are restricted to a range of between 20ms and 25ms and are primarily involved in matching nerve patterns against previously learnt patterns and also calculating pitch and loudness from position, rate and intensity information (Colgin et al., 2009). Once durations shorter than 20ms per tone are reached it becomes very difficult to analyse the tones separately, and different strategies, such as relying on detection of direction of frequency change, or sound analogue comparison, are needed to identify these very short frequency change stimuli. Two of the participants were unable to identify stimuli shorter than 16ms per tone and the theory presented here is that they were unable to modify their identification strategy any further in order to help them reach even shorter stimuli levels.

Many participants (n = 15) reached stimulus durations shorter than 10ms per tone and it is this area where it was theorized that participants are likely to use sound analogue comparison for identification of stimuli. Most of the participants who reached this level identified shorter duration stimuli in the single timbre condition than in the varied timbre condition, which provides support for the theory that sound analogue comparison, rather than frequency change detection (or pitch identification) was being used to identify these very short stimuli. However, it is only at the very shortest duration level of 1ms per tone that it would seem necessary for sound analogue comparison to be used exclusively (because of a lack of pitch information) and it is worth noting that none of the participants was able to identify stimuli in the varied timbre condition at the duration of 1ms per tone. This result provides further support for the theory that sound analogue comparison, rather than frequency change detection (or pitch identification), is exclusively used to identify stimuli at the duration of 1 ms. It also leads to a conclusion that the shortest duration at which it is possible to identify the direction (low-high or high-low) of a change in frequency of 1,000 Hz and 2,000 Hz tones is 2ms, which is two wave cycles for the lowest tone.

Implications

When intending to study frequency (pitch) identification, and especially to find the minimum duration for identification of the pitch of sequential tones, the results of this study suggest that it is important to minimize the possibility that other strategies, such as sound analogue comparison, might be used to identify stimuli. The best method to ensure that study results relate exclusively to pitch identification, is to vary the pitch of stimuli and to require the participant to identify each tone by its pitch. If studying frequency change direction detection, this study suggests that the best method to ensure that frequency change direction detection is being used is to vary the timbre of stimuli being used.

Limitations

There were no criteria for acceptance into the study and it is obvious that a wide variety of detection ability levels were encountered. Some participants (n = 5) were unable to reach stimulus durations shorter than 10ms per tone, at which point it was theorised that participants were likely to use sound analogue comparison for identification of stimuli. Their results did not therefore contribute to the main intent of the research.

Recommendations for Further Study

In a study of this type there is a clear need to screen participants beforehand, based on their ability to identify changes in frequency at exposure durations as short as at least 8 ms. Although current results have supported the hypothesis tested by this study, future researchers may wish to consider a similar study but with participants selected for the ability to pass a screening test for short tone frequency detection, following extended practice.

An issue that was raised by this research is that, only by making adaptation impossible through the use of identification restrictions (such as forcing pitch identification), can the duration threshold of pitch perception for very short contiguous tones be determined. Because none of the studies reported above has ensured that pitch perception was being used exclusively, the minimum duration at which it is possible to identify the pitch of two or three contiguous tones has not been established. This is the subject of the next chapter.

Chapter 4: Study 2: MDPP Background and Pilot Study

It has been established that the pitch of discrete, very short sine wave tones can be identified within a melodic context when participants are presented with as few as eight wave cycles of a target tone (Patterson et al., 1983; Robinson & Patterson, 1995). For a 1046.5 Hz, tone (two octaves above middle C, or C6), this represents a duration of around 8ms per tone. However, it is unclear if tones as short as these are identifiable by their pitch when the tones are played contiguously in a melodic sequence. The effects of forward and backward masking, where tones required longer time for identification when they were preceded by another tone (forward masking) or followed by another tone (backward masking) (Massaro, 1975), suggests that longer durations might be required to identify the pitch of tones that are played contiguously with other tones. That just eight wave cycles (duration between 8ms and 34ms depending upon the frequency of the tone) are required to identify the pitch of a single tone, but 160ms are required to identify the pitch of a tone in a cyclically repeated melodic sequence (Warren et al., 1991), leads to the conclusion that the minimum duration of sound required for identification of pitch in short contiguous tone sequences lies somewhere between 8ms and 160ms. In addition, results from the literature review would suggest that tones of differing relative frequencies and temporal positions would require differing durations for identification. The aim of this study was to establish the MDPP for tones in all positions and relative pitch levels, as well as overall minimum durations, for 3-tone and 2-tone contiguous sequences.

Although little work has been done relating to short contiguous tone *identification* in melodic sequences (i.e. the accurate perception of the pitch of individual tones within a short sequence of tones), much research has sought to establish the limits of perception in relation to the *ordering* of short contiguous tones in melodic sequences. As already stated above, this research has primarily been concerned with the perceived importance of tone ordering to language comprehension (Divenyi & Hirsh, 1974; Warren & Bashford, 1993). Thus, Hirsh (1959) found that participants could successfully identify the order of presentation of two sounds (including sine wave tones) for onset duration differences as short as 15 – 20ms. Divenyi and Hirsh (1974) found that participants could successfully identify the order of presentation of sequences consisting of three contiguous tones for even shorter tone durations (2 – 7ms per tone), and Wier and Green (1975) found that discriminating between two ‘fixed frequency’ tones, the only difference between which was the order of presentation of the two tones, was possible for tone durations as short as 1ms. Although it is clear from these studies that discriminating the order of tones can be achieved at extremely short durations, as the study in Chapter 3 established, it is unlikely that any form of pitch identification or pitch change analysis was used to discriminate between the shortest stimuli used in Wier and Green’s and Divenyi and Hirsh’s studies (for example). This is because pitch perception is not possible at durations shorter than 4ms per tone. In short, there is not enough information in tones shorter than 4ms to enable accurate pitch identification (Doughty & Garner, 1947; Robinson & Patterson, 1995). As has been detailed above, alternative methods that can be used for identifying contiguous tone sequences at these very short durations include detection of direction in frequency change, where the direction of frequency change (higher or lower) can be discerned without the need to identify the pitch of either tone (Evans, 1978; Molholm et al., 2005; Semal & Demany, 2006; Zatorre & Schönwiesner, 2011); and sound analogue comparison, where distinguishing features of a sequence are pattern-matched against cognitive templates that have been established through prior exposure to similar sounds (Hopfield, 1995).

This is the critical, central proposition that underpins this thesis. To emphasize this point, ***it cannot be assumed that the ability to order individual tones within a short sequence implies that the pitch of each of the individual tones is perceptible or identifiable, because it is possible to place sine wave tones in sequence order without identifying the pitch of the tones*** (Divenyi & Hirsh, 1974; Warren, 1974). Therefore, ***the minimum duration for accurate perception of pitch in a short sequence of tones cannot be implied from the results of the studies reviewed.*** Noting the difference between the ability to discriminate order of sounds and to identify pitch correctly, Warren and Bashford (1993) clarified that sound discrimination can be made using the global characteristics of a sound sequence, rather than by identifying the individual sounds within a sequence.

In order to outline the extent of this gap in knowledge, a detailed review of the field of pitch perception was undertaken, using a wide variety of key words including “pitch change” “pitch perception”, “frequency change” “frequency detection” and “pitch duration”. Of the pitch perception studies found, most examined isolated tones. These tones had either been presented as a single entity (Doughty & Garner, 1947; Pollack, 1967; Robinson & Patterson, 1995), in a sequence with half-second gaps between each tone (Patterson et al., 1983) or with masking tones played before or following target tones (Leshowitz & Cudahy, 1973; Massaro, 1975; Massaro, Cohen, & Idson, 1976).

It is possible that a study by Warren et al. (1991) did require participants to identify the pitch of fast contiguous tones. This study involved participants identifying by name short familiar melodies that were presented in a continuous loop without the start or end tones being identified. Results were that melodies presented in these brief cyclic patterns were not identifiable below 160ms per tone. However, because the task was to identify known melodies rather than the pitch of specific tones, the identification of the pitch of each tone was not a requirement and therefore it is not known if the participants in the study used pitch recognition, or the overall “gestalt” of each melody, to identify the melodies correctly. However, if it is assumed that pitch identification was used in this study, then this would mean that the MDPP for a looped sequence of 6 to 9 tones is around 160ms per tone. This duration is similar to the duration required to order vowel sounds and sine wave tones presented in a continuous loop (Thomas & Fitzgibbons, 1971; Thomas et al., 1970).

As already established above, the gap in knowledge relating to the minimum duration required for identification of pitch in contiguous tone sequences can be estimated as being between 8 wave cycles (i.e. 8-34ms) and 160ms. A theory to account for this large difference involves consideration of brain wave processing. To summarise conclusions drawn in Chapter 2, the cyclic processes (i.e., brain waves) that are primarily used to recognise pitch last around 25ms and are classified as Gamma waves. In continuous sequences individual processing slows down and Beta waves emerge as sub-harmonics of the primary Gamma waves. Beta waves persist for around 50ms and are used to process sound information for longer sequences of tones. In some conditions, such as simple scalic patterns, it is possible to identify the order of cyclically repeated tones at segment durations of 50ms, which is the duration of the type of Beta wave commonly used for analysing sound. However, as soon as tones are placed more randomly in a sequence (i.e. not ordered in a simple scalic pattern) the 50ms tones are not able to be ordered correctly. A tone duration of at least 125ms (i.e. a full Theta wave) is required for each sound segment before sound order in non-scalic sequences can be identified. The minimum duration required for ordering sine wave tones in a looped sequence is 160ms per tone segment (Warren et al., 1991), which falls within the Theta wave band of 125ms to 240ms per wave.

In contrast to this, isolated sounds, such as two short contiguous sine wave tones, can be ordered successfully at durations well below 50ms because short segments of sound are processed more thoroughly than sounds in a series. In terms of short sound research, echoic memory may account for the ability to identify pitch changes in very short tones. Echoic memory, as discussed in the literature review, lasts as long as a Theta wave and sounds can be reprocessed during this time unless other sounds disturb the process.

This theory suggests that these very short sounds can be held in echoic memory and processed multiple times. For example: two 30 ms sound segments, lasting a total of 60 ms, could be held in echoic memory for around 240 ms (according to Massaro's (1974) research). This would give around three times the processing time compared to that available for continuous sound sequences. It therefore stands to reason that the mechanism that forms the limiting factor for continuous sound perception and identification (i.e. the duration of a theta wave), does not apply to identifying the order (and possibly the pitch) of two or three very short contiguous sound segments presented in isolation. Therefore, minimum durations for perception of pitch should be much shorter for short sequences of contiguous tones.

Results from sound ordering studies suggest that minimum durations for identification of pitch might vary depending on different characteristics of both the relative frequency and position of the tone being investigated. Thus, Divenyi and Hirsh (1974) found that 3-tone sequences that changed sequentially *from low to high or high to low* were more easily identified than those that included a reversal in direction. They also found that sequences *ending with the highest and lowest tones* were more easily recognized than sequences ending within this high/low boundary. Warren (1972) found that the *first and last sounds in a sequence* were more easily identified than the middle sounds. Massaro (1970) found that *a second tone, played after a target tone*, interfered with participants' ability to identify the pitch of target tones within a duration of 240ms. This effect was later named “backward recognition masking” (Massaro, 1975). In a similar study, Leshowitz and Cudahy (1973) found that a target tone *preceded* by a leading tone was, for some participants, more easily identified than a single tone by itself. Watson et al. (1975) found that the *lower tones*, relative to the other tones in the sequence, required larger frequency differences for identification than higher tones. They also found that *tones that came immediately before a 40ms gap* were more easily identified than all of the other tones, with the exception of the last tone.

This knowledge is summarised below. According to the above studies, short contiguous tone sequences are more readily identifiable if:

- The tone is part of a unidirectional sequence,
- The last tone is the highest or lowest in the sequence,
- The tone is either first or last in a sequence,
- The tone is the second of two tones.
- Short contiguous tone sequences are less readily identifiable if:
    - The tone is followed by other tones,
    - The tone is a ‘middle pitch’ in a sequence (i.e. not highest or lowest).

Results from some of these reports conflict with each other. The most conflicting results relate to the first tone of a sequence. Research from Massaro (1970) would indicate that the first tone (that was immediately followed by other sounds) would require longer durations to identify because of the masking effect. whereas Warren (1972) found that the first sound was easier to identify than other tones.

Based on the review of previous research, the following research questions were addressed:

1. What is the minimum duration for perception of pitch (MDPP) in 2-tone and 3-tone sequences?
2. Does the position of the target tone (first second or third) affect the minimum duration required for perception of pitch in 2-tone and 3-tone sequences?
3. Does the relative pitch (low middle or high) of the target tone affect the minimum duration required for perception of pitch in 2-tone and 3-tone sequences?

Sections of this study.

To ensure reliability and comprehensiveness, the following studies included:

- A pilot study to test the effectiveness of method and to confirm that the position (first second or third) and relative pitch of a tone (highest tone, middle tone, or lowest tone) in a sequence affects the ability to perceive the pitch of a tone accurately at short durations.
- A set of practice activities, whereby prospective participants could improve their skills in identifying short tone stimuli, before participating in the acceptance test.
- An acceptance test whereby prospective participants were required to identify single tone stimuli at durations of 10ms or less before progressing to the main study, which included the following three conditions:
- A single tone condition, where study participants were presented with single tone sine wave stimuli to be matched against piano tones. This condition was designed to establish the shortest duration at which single short tone stimuli could be identified using this method, and secondly, to test for differences between MDPP in isolated tones and tones in 3-tone contiguous sequences.
- A 3-tone condition.
- A 2-tone condition

What follows in this chapter describes the background and pilot work required. The main study with three conditions is then reported in Chapter 5.

Definitions

In this thesis, the word 'tone' always refers to a sound with a specific pitch (or primary frequency). Although other definitions of the word 'tone' include "the largest difference in pitch between two notes that are next to each other in the western musical scale" and "the quality of sound of a musical instrument or singing voice" (Dictionary.cambridge.org, 2019), these definitions of the word tone have not been used here. The word 'semitone' always refers to a frequency ratio of around 18:17 (or exactly 100 cents) between two tones. 'Stimulus' refers to a contiguous 3-tone or 2-tone sequence of sine wave tones, or a single tone. The phrase 'test tone' refers to the first, second or third tone of the contiguous 3-tone or 2-tone sequence (i.e. the stimulus) that is required to be matched against a target tone. To emphasise this point, 'test tone' and 'stimulus' have different meanings in this study. The 'test tone' is one of the two or three contiguous tones that make up the stimulus (or the only tone, in the single tone condition) whereas 'target tone' refers to a piano tone played after a stimulus that is used for comparison with the test tone (which is either the first, second or third tone of the stimulus). A "trial" refers to the presentation of a single stimulus for identification. A trial run refers to a series of stimuli presentations that finishes (in the main study) at the first identification error after 50 stimuli have been presented.

Equivalent Perception Durations for Different Frequencies.

In the following studies, durations below 40ms were referenced to the tone B5. This is because minimum durations required for identification of single tones change, depending on the frequency of the tone. Researchers studying short pitch perception have reported that durations required for identification of pitch in very short tones increases as the frequency of tones decrease (Doughty & Garner, 1947; Patterson et al., 1983; Robinson & Patterson, 1995). Patterson et al. (1983), in their research on very short tones in a melodic context, found a direct relationship between MDPP and the number of wave cycles required at each frequency tested. The minimum number of wave cycles required for identification of tones between 900Hz and 200 Hz remained relatively consistent at 8 wave cycles, even though the duration required to identify the tones changed from 10ms for 900 Hz tones to 40ms for 200 Hz tones (Patterson et al., 1983 Figure 2a). Therefore, perception of melodic pitch in this frequency range can be more accurately reported using wave cycles, rather than milliseconds. Based on this observation, Robinson and Patterson (1995) used wave cycles for their short duration pitch identification study.

However, there was an aspect of the Patterson et al. research that was not accommodated in Robinson and Patterson (1995). In their 1983 study, Patterson et al. also noted that, for tone durations longer than 40 ms, pitch identification success was similar, regardless of the frequency for all tones with five or more harmonics (Patterson et al., 1983, p. 5) and for sine wave tones with frequencies above 100 Hz (Patterson et al., 1983 - see chart p. 4). By not taking this into consideration, Robinson and Patterson included tone durations that were clearly not equivalent, such as a short F4 tone (186ms; 64 wave cycles at 349 Hz), being compared to a much longer C1 tone (1,952ms; 64 wave cycles at 33 Hz).

Based on the research findings outlined above, segment durations in this study were adjusted in two different ways. For stimulus durations from 6ms to 40 ms, tone durations were adjusted based on their wave cycles count. For stimulus durations from 40ms to 200ms, tones were created as close to the indicated stimulus duration as possible (i.e. ± 2ms). The reason for this variation (± 2ms) is that all stimuli in this study were started and ended at positive zero crossing. This meant that stimuli durations were slightly different for all tones.

Therefore, stimuli were created using different criterial over three temporal zones. Stimuli were created with the same segment durations from 200ms to 40ms (i.e. for tone durations of 200ms, 100ms, 70ms, 60ms 50ms 45ms and 40ms). In the transition area between 40ms and 10ms, tone durations were only reduced if they had not yet reached 10 wave cycles. Once each tone reached 10 wave cycles, the tone's duration was maintained until all tone durations reached 10 wave cycles. This meant that the lowest pitched tone (C4) maintained a duration of 38.2ms (i.e. 10 wave cycles) until the highest frequency tone (C6) reached a duration of 9.6ms (i.e. 10 wave cycles). Once tones of all frequencies reached 10 wave cycles (at the 10ms duration level), all stimuli were again reduced simultaneously, but this time at a rate of one wave cycle per duration level.

To establish a baseline for measurement and comparison, the tone 'B5' was used as the reference tone for reporting durations. The B5 tone was selected because it vibrates at approximately one cycle per millisecond and thus the relationship between the duration level reported and the actual duration of the tone remained consistent throughout all duration levels (See Appendix 2 for a Table of durations for all stimuli at all duration levels). Therefore, in this study, tone durations below 40ms have been reported as 'B5 equivalent' (B5eq), meaning that the durations reported are referenced to the duration of the B5 tone at each duration level.

Method Selection Rationale for Testing Pitch Perception Accuracy

To ensure pitch perception (and not analogue detection or another strategy) was being used to identify the pitch of each tone, and to ensure that minimum durations for perception of pitch at each position (first second and third) and relative pitch (highest, middle and lowest) was quantifiable, the following criteria were created:

- the pitch of each tone in the sequence should be identified,
- sound analogue matching should not be possible for stimulus identification,

- pitch perception accuracy should be measurable to within a semitone,
- the answers should not be guessable from previous questions and answers,
- the task should be easy to understand.

Methods that had previously been used to ascertain the accuracy of pitch perception at short durations were examined for appropriateness for this study. Four studies were found that used pitch detection tests for identifying very short tones:

1. Patterson et al. (1983) required participants to identify which of four tones changed pitch on a second presentation of a 4-tone melody. The duration between the onset of each tone was a constant 660 ms. The duration of each tone ranged from 10ms to 80 ms. All tones were located within the diatonic scale being used. In the repetition sequence, one tone was raised or lowered by one scale step (i.e. either one or two semitones). In the test example provided by Patterson et al., the second tone was two semitones lower than the original presentation.
2. Robinson and Patterson (1995) required participants to identify the pitch chroma (i.e. either a ‘C’, a ‘D’, an ‘E’ or an ‘F’) of tones presented in one of four octave ranges. After each selection participants were given feedback for accuracy. To identify the pitch chroma of the tones successfully, participants needed to retain the pitch of one or all of the four possible tones in memory. They obtained this pitch information from either their warmup activities or the success of their previous selections in the study.
3. Doughty and Garner (1947) used two participants who adjusted a dial that regulated the duration of a tone that was repeated once every second until a selection was made. Participants were given two tasks. First, they were required to select the duration at which a sense of tone first emerged from the click sound of a very short tone burst (result: 4.4ms B5 equivalent) and, secondly, the point at which the tone became more dominant than the

click (result: 10.2ms B5 equivalent). Each participant practiced until they consistently chose the same duration at each frequency level and loudness level. The final reported duration was the mean of 40 selections over four sessions.

4. Warren et al. (1991) required participants to identify which of eight possible short familiar melodies was presented. These melodies were started at random points in the melody and were continuously looped until a selection was made. Tone durations ranged from 40ms to 320 ms. This was the only study of the four reviewed here that used contiguous sequences of tones, rather than either single isolated tones or tones that were spaced evenly apart.

The methods of these studies were all examined, for possible adoption for use in Study 2 but all were rejected for the reasons listed below:

1. Although the definition for pitch perception used in Study 2 was adopted from Patterson et al. (1983), their pitch identification method used spaced tones rather than contiguous tones and adopting their method for this study would require participants to listen to sequences up to 100 times faster than sequences presented in the Patterson et al. (1983) study. Their method was therefore considered not appropriate for this study.
2. The Robinson and Patterson (1995) method of matching stimulus tones to memorised tones could easily lend itself to analogue matching rather than pitch identification. This is because the participant was asked to memorise the sound of four tones and then match one of them with an identical tone with identical timbre. This method was therefore rejected.
3. The Doughty and Garner (1947) method did not include a means for measuring the accuracy of pitch perception for the tone at the selected duration. The participant was

simply asked to identify at what duration a sense of pitch was more pronounced than a click sound for each tone being presented. This method was therefore rejected.

4. The Warren et al. (1991) study required participants to identify contiguous sequences of tones, which is similar to the requirements for this study. However, their method is susceptible to analogue matching because there were only eight melodies from which to choose. With a single presentation of two or three tones, the correct phrase could be identified simply by ordering the relative pitch of each tone. A participant in a study by Nickerson and Freeman (1974) was able to discriminate between six 4-tone cyclic sequences down to segment durations of 2ms. Because this duration is shorter than the MDPP (Doughty & Garner, 1947), it is clear that pitch perception was not the method used to identify pitch sequences in this form of study. The Warren et al. (1991) method was therefore rejected.

Therefore, because no suitable method found for identifying the pitch of short contiguous tones was found, a new method was created.

Pilot Study.

To test the viability of the new method for identifying the pitch of each tone in the sequence, a pilot study was completed.

Method

Participants. There was one participant, the author (aged 58 years), who has had more than 50 years' experience as a musician and who, in preparation, practiced short tone interval recognition for more than 20 hours.

Design. To ensure each tone in each sequence was identified by its pitch, the participant matched the tones in each stimulus against a tone with different timbre and duration. Sine waves tones were selected as the stimulus timbre because of the long history of using sine waves in studies of this kind and because they provide an unambiguous sense of pitch. A piano sound was selected as the test timbre because it is the most used timbre in aural training and because the piano timbre is familiar to most people (Loh, 2007). Using a sine wave timbre for stimuli and a piano timbre for test tones minimises the possibility of sound analogue matching because there are many differences between the two timbres, making it unlikely that anything other than the frequency of the tones could be matched.

To avoid the possibility of participants guessing the answers based on similarity to previous answers, the stimuli were created using a variety of musical intervals and a variety of starting tones. The frequency ratios (musical intervals) between the tones were randomly selected from a range between four semitones and nine semitones (inclusive). This ensured all intervals were wider than three semitones and all intervals except the ‘jump intervals’ (i.e. the highest and lowest tones when they occur contiguously in 3-tone sequences) were less than 10 semitones. The jump intervals ranged from eight semitones to 18 semitones. Contiguous tones needed to be at least a minor third apart (20% of centre frequency) to avoid potential issues with ‘critical bands’, where different frequencies can stimulate the same nerves in the cochlear (Fastl & Zwicker, 2007, p. 150).

Materials. Materials for this study included stimuli, test tone and apparatus.

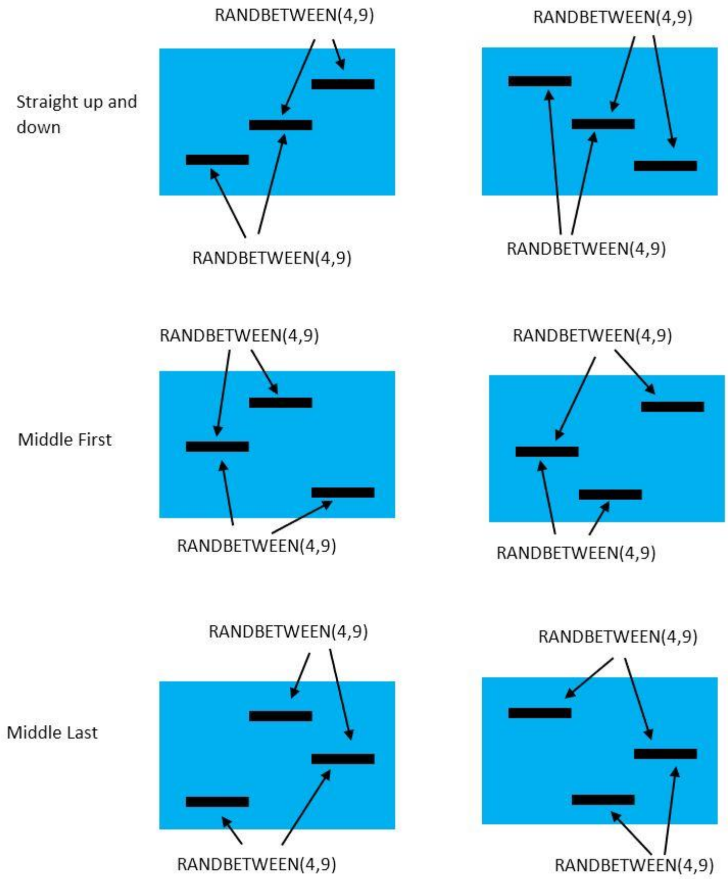
*Stimuli***.** Stimuli were created with the aim of minimizing the possibility of identification by any means other than by identifying the pitch of the stimulus tone. Two methods were used for this. The frequency ratios (musical intervals) between the three tones in each 3-tone and 2-tone stimulus were randomly selected (with certain constraints identified below) to ensure the participant could not use prior recognition of a melodic pattern to help identify the tone; and the starting pitch of the first tone of each stimulus was randomly selected to ensure the participant could not use recognition of a previously identified starting tone to help identify the target tone. Target tones and test tones were presented using different timbres. Target tones were presented using a sine wave timbre and test tones were presented using a piano timbre. This was done to ensure that matching wave forms could not be used to help with identification of target tones.

Stimuli were created from sine waves in the frequency range 250 Hz to 1050 Hz. This frequency range was selected because it is within the range of most musical instruments (making it a commonly heard and therefore a familiar frequency range) and the tones in the lower end of the selected range were still perceptible at reasonably short durations (see “equivalent perception durations” above for more detail). The selected frequency range also encompassed the frequency range used by Patterson et al. (1983), from which the definition of pitch perception used in this study was taken. The frequencies were selected from tones in the equal tempered scale of which the tone A4: 440 Hz is a member, within the indicated frequency band. There was a total of 25 equal tempered scale tones in the frequency band from 250 Hz to 1050 Hz. The pitch of the selected sine wave tones ranged chromatically from C4 to C6 (261.6 Hz to 1046.5 Hz). Tone durations in the pilot study ranged from 70ms to 10ms. The sine waves for the stimuli were created using the program “Audacity” (FreeSoftwareFoundation, 1991) and then imported into the sound recording and creation software “Sonor” (Cakewalk, 2009) for editing. They were then prepared by shortening them to the desired length, starting and stopping the sound waves at zero crossing. Raised cosine ramps to start and end stimuli were considered but not used because of a concern that ramps favour the higher frequency tones. At the 25ms duration level, a raised-cosine ramp would affect 20% of a C4 tone but only 5% of a C6 tone. Additionally, several researchers have reported no significant difference in identification accuracy between gated and ramped stimuli (Ronken, 1972; Watson, Wroton, Kelly, & Benbassat, 1975).

Stimuli were created at only six duration levels: 10ms, 15ms, 20ms, 30ms, 50ms and 70ms. At each duration, 30 stimuli were created: five stimuli for each of the six pitch orders that are possible for three tones of differing frequencies (see Figure 4.1).

Figure 4. 1

Stimulus Tone Sequence Direction Combinations



The sine wave tones were joined with other sine wave tones at zero crossing to create 3-tone sequences using a method described below. Each stimulus consisted of three contiguous sine wave tones and was ordered in one of six ways: low-middle-high, high-middle-low, middle-low-high, middle-high-low, low-high-middle and high-low-middle. These can be represented visually using lines to represent pitches (see Figure 4.1).

The 180 stimuli were created subject to three constraints:

Constraint 1: Contiguous tones needed to be at least a minor third apart (20% of centre frequency) to avoid potential issues with ‘critical bands’, where different frequencies can stimulate the same nerves in the cochlear (Fastl & Zwicker, 2007, p. 150).

Constraint 2: Tones needed to have different interval relationships and interval orders, to avoid selection based on prior experience of the interval being presented (see Chapter 2).

Constraint 3: At each duration level, stimulus tones needed to be evenly spread across the selected frequency spectrum (250 Hz to 1050 Hz).

Tone frequencies and tone orders used to create stimuli were selected randomly using the “RANDBETWEEN (x,y)” method in Excel. This method creates a random integer between two given numbers (provided in brackets), inclusive of the numbers. For example: RANDBETWEEN (4,9) will produce a random integer between 4 and 9 (inclusive). The starting tone of each stimulus was selected by using RANDBETWEEN (1, 25) to produce a number corresponding to a pitch between C3 (tone ‘1’) and C6 (tone ‘25’). Once the first tone was selected, RANDBETWEEN (4, 9) was used twice to select the number of semitones between the tones.

To create stimuli that included tones in all possible relative frequency positions, the two random numbers between 4 and 9 were used in three different ways (see Figure 4.1 for a visual representation of this process). This process ensured all intervals were wider than three semitones (for the reason given above), and all intervals except the ‘jump intervals’ (i.e. the highest and lowest tones when they occur contiguously) were less than 10 semitones. The jump intervals ranged from eight semitones to 18 semitones.

For the straight up and straight down stimuli, the randomly generated numbers were used to create intervals between the first and second tones and the second and third tones. For the ‘straight up’ stimuli, the first random number was added to the pitch number of the first tone to select the pitch of the second tone. The second random number was then added to the second tone to select the pitch number of the third tone. For the ‘straight down’ stimuli, the random numbers were subtracted instead of added.

For stimuli that started with the middle tone, the two random numbers were used between the first and second and first and third pitch numbers. For stimuli that ended with the middle tone, the two random numbers were used between the first and third and second and third pitch numbers.

If the lowest or highest tone in any stimulus went outside the band of 25 notes used in the study (i.e. if the highest number was higher than 25 or the lowest number was lower than 1), the stimulus was discarded, and another created. This process was repeated until eight stimuli had been successfully created for each of the sixteen duration levels and the six sequence orders (see Figure 4.1 for a visual description of the six sequence orders).

To ensure a balance of frequency levels, in each of the 36 stimulus groups created (six duration levels and six sound order variants) eight number sequences were created, of which five stimuli were selected for use. To select these five stimuli, the three numbers representing the pitch of each tone were averaged and then:

- The stimuli with the lowest and highest averaged number sequence were selected.
- The stimulus with the averaged number that was closest to 12.5 (the middle point) was selected.
- Two more stimuli, one with an averaged number above middle point and one below middle point, were selected from the remaining five stimuli. Selection of these two stimuli was done with the aim of maximising the variety of musical intervals in the selection of stimuli. If an interval number (i.e. one of the random 4 – 9 numbers) was

already represented three times, this interval number would be avoided (if possible) for the final two stimuli.

- The remaining three number sequences were discarded.

Test Tones. Test tones were created by recording a chromatic scale from C4 to C6 in minims that had been notated in the computer program Sibelius (Finn, 1993). The tones were recorded using a 'piano' timbre and saved as individual wave files. During the study, all sounds were accessed and played by a HTML program designed for this purpose.

Apparatus. Stimuli were presented via the internet site https://intervalstore.a2hosted.com/, created for this purpose. The participant listened to each stimulus through Sennheiser HD 518 headphones and responded by clicking one of three buttons representing the three alternative choices (higher, the same or lower). The study participant interface was created using HTML (WHATWG, 2017) with data recording using SQL (Chamberlin & Boyce, 2016). The data were initially stored on a server controlled by A3Hosting and then downloaded at the end of the study. PHP (PHP, 2018) was used to communicate with the server and transmit study information between the user and the database. See Appendix 3 for the code that was created for this and other studies in this thesis.

Procedure. The pilot study used the method of constant stimuli. The 180 stimuli (30 stimuli at 6 duration levels) were presented three times for a total of 540 trials. For the first 180 trials the participant was required to identify the first tone in the sequence of three. The sequence of stimulus presentations was then reordered and presented a second time for 180 trials. This time the participant was required to identify the second tone in the sequence of three. The reordering and presentation process were repeated, and the participant was then required to identify the third tone in the final 180 trials. A three-alternatives forced choice method was used for all presentations. The choices were: (1) the test tone (piano tone) is the same frequency as the target tone (sine wave tone); (2) the test tone is a semitone lower than the target tone; (3) the test tone is a semitone higher than the target tone. A semitone was chosen as an appropriate interval to test accurate pitch identification because it is the smallest interval used by Patterson et al. (1983) in their study on melodic pitch, from which the definition of melodic pitch used in this study was derived. Patterson et. al., in their 4-choice, fixed choice study, used a 62.5% success rate as the accepted threshold level for 50% identification success. as a 1 in 4 study, the guess rate was 25% and 50% of the remaining 75% of trials was 37.5%. Therefore, a minimum of 62.5% (i.e. 25% plus 37.5%) success was required to ensure half of the stimuli were correctly identified without the benefit of guessing (Prins, 2016). This study used a cut-off rate of 66.7% because it was a 3-choice, fixed choice study with a guess rate of 33.33’% (Prins, 2016, p. 62). An identification success rate of 66.7% or greater was therefore selected as the cut-off criteria for successful identification of stimuli at each duration level. Stimuli were created with the aim of minimizing the possibility of identification by any means other than by identifying the pitch of the stimulus tone. Two methods were used for this. The frequency ratios (musical intervals) between the three tones in each 3-tone stimulus were randomly selected (with certain constraints identified above) to ensure that the participant could not use prior recognition of a melodic pattern to help identify the tone; and the starting pitch of the first tone of each stimulus was randomly selected to ensure the participant could not use recognition of a previously identified starting tone to help identify the target tone. In addition, the test tone used a piano timbre, rather than a sine wave timbre, to ensure sound analogue comparison was not being used to match the test tone against the target tone.

The participant was seated at a computer with headphones, with sound self-adjusted to comfortable loudness (approximately 65 – 75 dB). The participant was presented with a set of instructions indicating which tone to identify and which buttons to press. After reading the instructions, the participant clicked on the "Next tone" button. The first stimulus was then presented. After each stimulus was presented, the participant clicked on the "compare tone" button and the test (piano) tone was presented. One of the three options was selected, depending on if whether the participant thought that the "test tone" was higher, the same or lower than the target tone. After making a choice, the participant received feedback about accuracy (correct or incorrect). After a short 2-sec pause the next stimulus was presented. The cycle was repeated until the participant had responded to all 180 stimuli. At this point the participant was directed to attend to the second tone and the 180 stimuli were repeated. Finally, the participant was directed to attend to the third tone and the stimuli were repeated for a third time. As explained above, threshold level for identification success was set at 66.7% correct for this 3-choice, forced-choice study (Prins, 2016, p. 62).

Results

One hundred and eighty 3-tone contiguous sequences were presented three times to a single participant, who identified the pitch of the first, second and third tone in that order. Table 4.1 shows the percentage of correct identifications for each tone position at each duration level. Thirty stimuli were presented at each tone position at each duration level. At a cut-off percentage of 66.7% for this three-choice, forced choice study, identification of the first tone was only possible at 70 ms duration (see Table 4.1). Identification of the second tone was possible for tone durations as short as 30ms and identification of the third tone was possible for tone durations as short as 10ms (the shortest duration being tested).

To test whether the position of a tone in a sequence affects the ability to perceive the pitch of a tone at short durations accurately, results were analysed by a Friedman Test, rather than a one-way repeated measures ANOVA., because of an extreme outlier at the 15 ms position in the third tone identification column.

Table 4.1

Pitch Identification success at differing tone durations and positions for 3-tone sequences

Duration (ms)	first tone (%)	second tone (%)	third Tone (%)	Mean (SD)
10	21	47	**79**	49 (23.7)
15	50	61	52	54 (4.8)
20	32	59	**82**	58 (20.4)
30	50	**75**	**87**	71 (15.4)
50	61	**92**	**83**	79 (13.0)
70	**74**	**75**	**95**	81 (9.7)
Mean (SD)	48(17.5)	68 (14.4)	80 (13.4)	

Note: Each percentage figure represents percent correctly identified out of 30 trials. Identification success above 66.7% are in boldface.

The Friedman test indicated that there was a significant difference in identification success, depending upon the position of the tone being identified, n= 6, $\chi2(2) = 9.333$, p= .009. Pairwise comparisons were performed (SPSS Statistics, 2017) with a Bonferroni correction for multiple comparisons. Pitch identification success was significantly higher when the tone being identified was the third tone compared to the first tone ($p = .012$).

In addition to tone position differences, pitch ‘height’ was also theorised to affect perception success. Comparisons were made across all stimuli between the lowest tone in the sequences, the middle tone and the highest tone. A 2-way repeated measures ANOVA test found no significant difference in identification scores between the three pitch heights (highest, middle, lowest). There was also no significant interaction between tone position and tone duration.

Identification success for the first tone was examined further. As indicated above, identification success was only found to be possible at the 70ms duration level. However, dividing the pitch of the first tone of the stimuli into highest, middle and lowest tone in the 3-tone sequence, when the first tone occurred as the highest tone, it was identifiable down to 50ms (90% accuracy: n=20 for identification at 70ms and 50ms) whereas when the first tone occurred as the lowest tone, it was not identifiable even at the longest duration being tested (70ms). The middle tone was identifiable at 70 ms but not at 50ms (see Table 4.2). None of the stimuli was identifiable below 50ms.

A chi-square test of homogeneity was conducted between relative pitch height type and identification success for stimuli 50ms and above (i.e. 50ms and 70ms). All expected cell counts were greater than five.

Table 4.2

First tone Pitch Identification success at differing durations and pitch heights in 3-tone sequences

Duration (ms)	Low tone (%)	Middle tone (%)	High tone (%)
50	38	50	**90**
70	58	**82**	**90**
Mean (SD)	48 (10)	66 (16)	90 (0)

Note: Identification success above 66.7% are in boldface.

Group sizes were unequal. Sixty-one stimuli were presented for identification of the first tone at durations of 50ms or longer. Of those, 20 consisted of the lowest tone in the sequence, 21 of the middle tone and 20 consisted of the highest tone in the sequence. Eighteen of the highest tones (90%) were correctly identified compared to 10 of the lowest tones (50%) and 14 of the middle tones (67%) a statistically significant difference in proportion (p = .023).

Discussion

In this study, the first tone of the three-tone contiguous sequence required significantly longer durations for identification success than the third tone. Identification of the third tone was possible down to the shortest duration sequences being presented (10 ms). This is not surprising in view of results from the many studies done on the ‘recency’ effect (Davelaar, Goshen-Gottstein, Ashkenazi, Haarmann, & Usher, 2005) and also the results from forward masking studies of sine wave tones (Leshowitz & Cudahy, 1973; Massaro, 1970). Supporting Massaro’s results, and contrary to results reported by Warren and Obusek (1972), the first tone in the series of three required longer to identify. This result was complicated by the finding that when the first tone was the highest tone, it was significantly easier to identify at the longer durations of the ones being studied (50ms - 70ms), than when the first tone was the lowest tone.

Based on these findings, the decision was taken to examine differences in minimum identification duration for both tone position (first, second or third) and pitch height (high, middle, low). However, instead of continuing with the Method of Constant Stimuli (MCS) used in the pilot, the staircase version of the Method of Limits (ML) was used for the main studies (see Chapter 5), because it is more efficient, concentrating most trials close to threshold. Moreover, because early trials are easy, it is likely that the observer will not make mistakes and will gain confidence. Also, a lack of mistakes when completing the initial longer duration stimuli will confirm that the observer has understood what is required.

Chapter 5: Study 2: MDPP Main Study

Three study conditions were tested in the main study. A single tone condition checked if the results from this method are comparable to other short tone studies, such as those done by Patterson et al. (1983), Robinson and Patterson (1995) and Doughty and Garner (1947). This was followed by a 3-tone condition and then a 2-tone condition. The rationale for completing the 3-tonecondition before the 2-tone condition was that, if there was no significant difference between the minimum duration for the identification of pitch of isolated tones and the minimum duration for the identification of pitch of 3-tone sequences, it would be pointless doing the 2-tone condition.

Method

Participants. Participants were first tested for their ability to identify the pitch of very short tones. They were required to identify sine wave tones consistently at durations of 10ms (B5 equivalent) before continuing in the study. In preliminary work it was found that participants who could not discriminate accurately at the 10ms duration level for isolated tones were not able to reach durations that were short enough to differentiate between the masking effects of high tones compared to low tones. For example, many potential participants who could not discriminate accurately at the 10ms duration level for isolated tones consistently scored around 100ms MDPP (i.e., random chance) for all tone positions (first, second and third) and relative pitches (highest, middle, and lowest).

Participants completed prior training to achieve this 10ms identification level, using a training program created for this purpose. They were asked to use the training program as much as possible before undertaking the study. This study requires a specific skill: being able to extract a tone from a tone series, convert it to a synthesised memory, hold the sound in memory and compare it to a later sound of a different timbre and duration. A participant’s level of innate and acquired skill undoubtedly affects the amount of practice needed to reach asymptote for the specific task required for this study. The length of practice undertaken ranged from 20 minutes for one participant to around 20 hours (completed by the author of this thesis, who served as a participant), and all participants were trained musicians. Micheyl, Delhommeau, Perrot, and Oxenham (2006) reported that between four and eight hours of extra training was required to bring non-musician participants up to level of trained musicians in psychoacoustic trials. Sparks (1976) reported a far greater extra training duration of up to 200 hours. There was no possibility for recruiting participants willing to undergo such extensive extra training, so trained musicians were targeted for participation. Participants were recruited through an email sent to all South Australian branch members of the Australian Society of Music Education (ASME) (approximately 300 people) as well as through personal recruitment from local educational institutions (15 people). Although more than 20 people agreed to participate in the study and undertook training, within the six-month period allowed to determine participation, only four reached the accuracy required at 10ms duration for identifying a single tone. Two reasons for this low participation rate are proposed. Firstly, as working music teachers, potential participants who had not already acquired a high level of auditory acuity would be unable to dedicate the amount of time required to achieve the pre-requisite level of skill needed to participate in the study. Secondly, as this study was intended to identify a limit to human perception, it was not surprising (given the wide spread of innate and acquired abilities) that only a small percentage of even a highly skilled pool would be able to identify tones at the shortest duration humanly possible.

Participants were four experienced musicians (aged 18 to 58 years; 3 males and 1 female). They all indicated "seven or more years" music experience in the participation questionnaire and were not paid for their participation.

Design. For all study conditions, a three-alternatives, forced-choice method was used. The choices were: (1) the test tone (piano tone) is the same frequency as the target tone (sine wave tone); (2) the test tone is a semitone lower than the target tone; (3) the test tone is a semitone higher than the target tone. A semitone was chosen as an appropriate interval to test accurate pitch identification because it was the smallest interval used by Patterson et al. (1983) in their study on melodic pitch, from which the definition of accurate pitch perception was derived (see Chapter 4).

Stimuli durations were decreased or increased using a two-down one-up staircase procedure (Leek, 2001). This was selected as the most appropriate method to identify minimum perception durations because of its simplicity, its capacity to deliver most trials close to threshold and its relative insensitivity to lapsing errors. Stimulus segment durations below 40ms were referenced to the tone B5 (see "Equivalent Perception Durations for Different Frequencies" in the "Definitions" section in Chapter 4). The stimulus was initially presented at 200 ms duration. After two successive correct identifications, stimulus durations were decreased to the next level (e.g 100ms following 200ms). After one incorrect identification, stimulus durations were increased. The stimulus duration levels for the single tone condition were: 200ms, 100ms, 70ms, 60ms, 50ms, 45ms, 40ms 35ms, 30ms, 25ms ,20ms, 15ms, 10ms, 8ms, 7ms, 6ms, 5ms and 4ms. The minimum duration level for each study condition was taken as the mean of the duration levels of the last five reversals in each trial run, counting backward from the final reversal. Reversals were recorded following an increase in exposure durations after exposure durations had decreased from longer durations or following a decrease in exposure durations after exposure durations had increased from shorter durations.

The procedure was stopped on the first reversal caused by an error after a minimum of 50 trials. Trial runs ranged from 53 trials to 68 trials. This procedure was preferred as sensible practice, rather than stopping data collection after the first five reversals, the method initially followed during further pilot work, after changing from the method of constant (see “Pilot Study”), consistent with common practice with the staircase method of limits. However, pilot work found that threshold in such instances (i.e. using common practice) could always be subsequently established at markedly shorter exposures and it was therefore assumed that such early errors reflected nothing more than an attentional lapse. The new procedure reduced the likelihood that the program would cease before the participant’s responses were near-threshold. Stimuli were randomly selected from the pool of 25 single-tone stimuli for each duration level.

Materials and Procedure. The stimuli, test tones, apparatus and procedure were the same as those created for the Pilot Study (see Chapter 4 above).

Single Tone Condition

To calibrate this investigation of 2-tone and 3-tone contiguous sequences with other studies, such as those done by Patterson et al. (1983), Robinson and Patterson (1995) and Doughty and Garner (1947), the minimum duration for identification of the pitch of a single tone using this method was established.

Results

The mean minimum duration was 8 wave cycles with a standard deviation of 2 (see Table 5.1). This is essentially the same as the 8 wave cycles established by Patterson et al. (1983) as the minimum duration for melodic pitch. The results from the 3-tone and 2-tone study conditions (to follow) are therefore directly comparable to previous research into single tone perception.

Table 5.1

Minimum Duration (ms) for identification of short sine wave tones made by four individual participants

Participant ID No.	single tone Minimum Duration (ms)
1	7
2	6
3	10
4	10
Mean (SD)	8.25 (2.06)

3-Tone Condition

The gap in knowledge relating to the minimum duration required for identification of pitch in contiguous tone sequences has been identified as lying between the minimum duration required for the identification of pitch of single tones (8ms – 34ms depending on frequency (Patterson et al., 1983)), and the minimum duration required for identification of continuous cyclic sequences (160ms (Warren et al., 1991)).

Even though it seems reasonable *a priori* that the MDPP in 3-tone sequences is longer than that required to identify single tones and shorter than the required for cyclic sequences of tones, the point at which there will be no difference in the MDPP with the addition of a further tone to a sequence has yet to be established.

It was hypothesised that the MDPP in short, non-cyclic 3-tone contiguous sequences, requires longer durations for identification of pitch than single tones (Hypothesis 1). This hypothesis was tested by comparing results from the single-tone condition against results from the 3-tone condition. It was also hypothesised that the MDPP in short, non-cyclic 3-tone contiguous sequences, can be identified at shorter durations than the median 160 ms required for identifying songs in a cyclic presentation as found by Warren et al. (1991) (Hypothesis 2). This hypothesis was tested by comparing results from this study against results from the Warren et al. (1991) study.

Based on the pilot study and results from masking studies (see literature review), it was hypothesised that *the lowest tone, when presented as the first tone in a 3-tone contiguous sequence requires longer durations for identification than the highest tone, when presented as the first tone in a 3-tone contiguous sequence (Hypothesis 3).*

It was also hypothesised, based on forward masking results from Leshowitz and Cudahy (1973) and Massaro (1970) that *the third tone in a sequence of three tones requires less time for perception of pitch than tones in the first or second position (Hypothesis 4).*

Materials for this condition were the same as described for the pilot study (above), except that stimuli were created at 16 durations. These durations were: 200ms, 100ms, 70ms, 60ms, 50ms, 45ms, 40ms 35ms, 30ms, 25ms ,20ms, 15ms, 10ms, 8ms, 7ms, and 6ms. The general procedure was the same as that used for the single tone condition. However, instead of a single stimulus set, nine stimuli sets were presented to the participants, thus making nine trial runs. Each time, different combinations of sound orders and note positions were presented for identification. The four participants were randomly allocated to two different presentation treatments (A and B) to minimise the effects of learning, adaptation, and fatigue. Participants in both treatment groups completed the same trial runs but in a different order.

The two participants in presentation treatment group A were presented with the series of trial runs in the following order:

- Identify the first tone:
    - when it is the middle tone,

  - when it is the highest tone
  - when it is the lowest tone
- Identify the second tone:
  - when it is the middle tone,
  - when it is the highest tone
  - when it is the lowest tone
- Identify the third tone:
  - when it is the highest tone
  - when it is the lowest tone
  - when it is the middle tone,

The two participants in treatment group B were presented with the series of trial runs in reverse pitch height order, but the same order for identifying the tone position:

- Identify the first tone:
  - when it is the lowest tone,
  - when it is the highest tone
  - when it is the middle tone
- Identify the second tone:
  - when it is the lowest tone,
  - when it is the highest tone
  - when it is the middle tone
- Identify the third tone:
  - when it is the middle tone,
  - when it is the lowest tone
  - when it is the highest tone

The target tone that required pitch matching was identified before each study phase started. Participants were instructed to identify either the first tone, the second tone or the third tone.

To establish if there was a significant difference between the minimum duration required for identification of 3-tone sequences and continuously looped sequences, the results from the Warren et al. (1991) study were used for comparison with the results from this study. In the Warren et al. (1991) study participants listened to a looped series of tones that were all the same duration but had different pitch orders. The tone sequences were created based on eight common folk songs: “Camptown Races”, “Yankee Doodle,” “Rock-a-bye Baby”, “God Rest Ye Merry Gentlemen”, “Happy Birthday”, “Twinkle Twinkle Little Star”, “Skip to My Lou” and “Love Me Tender”. Each loop was started from the third or fourth tone in each melody and stopped after the participant had made a selection as to the correct melody. The durations used were: 40 ms, 57 ms, 80 ms, 113 ms, 160 ms, 226 ms, and 320 ms (steps increased by a factor of the square root of 2, so that every two steps doubled in duration). An issue of concern was whether the two results were truly comparable, because the 3-tone study condition required participants to isolate tones, whereas Warren et al.’s looped study required participants to identify a whole tone sequence. However, no other study provided a more appropriate comparison, and the looped study appeared to be an easier test than identifying the pitch of each individual tone, so the comparison was accepted as appropriate.

Results

Nine trial runs of 3-tone sequences were presented to four participants. Table 5.2 (below) displays the results for all four participants for each of the nine trial runs completed. Table 5.2 also includes the results from the single tone condition (see above) for the purpose of comparison. As indicated above, because the method of sine wave tones being matched against pre-recorded piano was different to other methods used for pitch identification, it was considered important to check single tone condition results from this study against MDPP results found by other researchers (Patterson et al., 1983; Robinson & Patterson, 1995), and specifically that found by Patterson et al. (1983) as the minimum duration required for ‘melodic pitch’. As indicated in the Single Tone Results (above), the mean MDPP using the selected method (8ms) was directly comparable to previous research into single tone perception. Therefore, the results from the 3-tone condition (to follow) are directly comparable to previous research into single tone perception.

To compare minimum durations for identification of pitch of a single isolated tone against tones in a contiguous 3-tone sequence, a one-way repeated measures ANOVA was completed. There were no outliers in the data, as assessed by inspection of boxplots. Pitch identification was normally distributed ($p > .05$) except for the second position middle tone ($p = .001$), as assessed by Shapiro-Wilk's test of normality on the studentized residuals. The normal distribution violation was caused by one participant being able to identify the second tone in middle position at far shorter durations (Participant 2= 15ms) than any of the other participants could (Participant 1: 70ms, Participant 3: 70ms, Participant 4: 70ms). However, this result was left unmodified in the dataset because it did not increase the likelihood of a significant outcome. Because the assumption of sphericity was violated, a Greenhouse & Geisser (1959) correction was applied ($\varepsilon = 0.288$) and was used to correct the one-way repeated measures ANOVA. Minimum duration was statistically significantly different for the different tone positions, $F(2.589, 7.659) = 7.357$, $p = .013$. partial $\eta^2 = .75$.

A planned contrast was completed comparing the mean minimum duration for identification of a single isolated tone, (m= 8.3ms SD= 2.1) with the mean minimum durations of all of the contiguous tone positions: First/high (m= 18.25ms SD= 7.18), First/middle (m= 52.5ms SD= 23.6), First/low (m= 72.5ms SD= 16.39), second/high (m= 24.25ms SD= 13.66), second/middle (m= 56.25ms SD= 23.82), second/low (m= 77.5ms SD= 24.87), last/high (m= 32.5ms SD= 16.77), last middle (m= 46.25ms SD= 21.03), last low (m= 41.75ms SD= 36.49). There was a significant mean difference of 39.11ms, 95% CI [10.52, 67.7] ms, p = .022, $\eta^2 = .86$.

Therefore, the hypothesis that the MDPP in short, non-cyclic 3-tone contiguous sequences, requires longer durations for identification of pitch than single tones (Hypothesis 1) was supported.

Table 5.2

Minimum Duration (ms) for Perception of Pitch of Tones in a Series of Three Tones

P. No.	*Single*	*First tone (of 3)*			*Second tone (of 3)*			*Third tone (of 3)*		
	Tone	*Low*	*Mid*	*High*	*Low*	*Mid*	*High*	*Low*	*Mid*	*High*
1	7.0	70.0	50.0	25.0	70.0	70.0	15.0	7.0	15.0	25.0
2	6.0	60.0	20.0	8.0	40.0	15.0	7.0	15.0	40.0	15.0
3	10.0	100.0	70.0	15.0	100.0	70.0	35.0	100.0	60.0	60.0
4	10.0	60.0	70.0	25.0	100.0	70.0	40.0	45.0	70.0	30.0
Mean	8.3	72.5	52.5	18.3	77.5	56.3	24.3	41.8	46.3	32.5
SD	2.1	18.9	23.6	8.3	28.7	27.5	15.8	42.1	24.3	19.4

Note: P. No. = Participant's identification number. Low, Mid and High refers to the relative pitch (low middle or high) of the target tone.

To test hypothesis 2, the results from the Warren et al. (1991) study were used for comparison here. For this comparison, the results for the lower quartile of participants in Warren et al. were compared with average minimum duration results from the 3-tone study condition. It was reasoned that the four participants in the 3-tone study condition were more highly trained than most of the participants in the Warren et al. (1991) study and therefore it was considered reasonable that only the results from the more skilled participants would be used. Table 5.3 shows the median, lower quartile, and upper quartile results for their study, as well as the number of participants in the study.

Table 5.3

Average Minimum Duration (ms) for Identification of a Melody (from Warren et al. (1991, p. 280))

Melody	Median	Lower Quartile	Upper Quartile	No. Participants
God Rest Ye Merry Gentlemen	113	113	226	29
Camptown Races	160	113	160	30
Happy Birthday	160	113	226	29
Rock-a-bye Baby	160	160	226	30
Yankee Doodle	160	113	226	30
Love Me Tender	226	160	320	29
Skip to My Lou	226	160	320	25
Twinkle Twinkle Little Star	320	226	320	29
Overall (mode)	160	113	320	
Overall Mean (SD)	191 (64)	145 (40)	253 (60)	

An independent-samples t-test was run to determine if there was a difference between minimum durations for 3-tone sequences, compared to looped sequences. There were no outliers in the data, as assessed by inspection of a boxplot for values greater than 1.5 box-lengths from the edge of the box. Pitch identification was normally distributed ($p > .05$) for the 3-tone means, but not for the looped melodies ($p = .018$), as assessed by Shapiro-Wilk's test of normality on the studentized residuals. However, the comparison was completed because the t-test is relatively resilient to non-normal results. There were eight looped sequences and nine 3-tone sequences. Minimum durations were shorter for the 3-tone sequences (M= 47ms SD= 6.7ms) than the looped sequences (M= 145ms, SD= 40.2). There was homogeneity of variances, as assessed by Levene's test for equality of variances ($p = .072$). Mean minimum identification duration difference was 97ms, 95% CI [65.5 to 130.2] shorter for 3-tone sequences than looped sequences. Therefore hypothesis 2 -- *the MDPP in* *short, non-cyclic 3-tone contiguous sequences, can be identified at shorter durations than the median 160 ms required for identifying songs in a cyclic presentation as found by Warren et al. (1991)*-- was supported.

3-tone Results. One of the main aims of this study was to establish the MDPP in 3-tone contiguous sequences. Nine trial runs were completed by all participants. The mean minimum duration for identifying the pitch of a tone in a contiguous 3-tone sequence over all nine trial runs, was 47ms. However, within this result there were very wide differences associated with the tone's position in the sequence. Therefore, a more accurate summary is that the minimum duration for the identification of pitch in a 3-tone sequence ranges from 18ms to 78ms, depending upon the position within the sequence of the tone being identified. The shortest MDPP in a 3-tone contiguous sequence was 18ms to identify the first tone when it was the highest pitch of three contiguous tones. The longest MDPP in a 3-tone contiguous sequence was 78ms to identify the second tone when it was the lowest pitch of three contiguous tones. See Table 5.2 for details of the results for all trial runs.

A two-way repeated measures ANOVA was run to determine the effect of the position (first, second or third) and the relative pitch height (low, middle or high) on minimum durations for identification of pitch of sine wave tones. There were no outliers, as assessed by examination of studentized residuals for values greater than ±3. Pitch identification was normally distributed ($p > .05$) except for the second position middle tone ($p = .001$), as assessed by Shapiro-Wilk's test of normality on the studentized residuals. As mentioned above, the normal distribution violation was caused by one participant being able to identify the second tone in middle position at far shorter durations (Participant 2 = 15ms) than could any of the other participants (Participant 1: 70ms, Participant 3: 70ms, Participant 4: 70ms). This result was left unmodified in the dataset because it did not increase the likelihood of a significant outcome. Mauchly's test of sphericity indicated that the assumption of sphericity was met for position, $\chi 2(2) = .807$, $p = .661$ and Pitch $\chi 2(2) = .685$, $p = .775$. Regarding the two-way interaction, with a Mauchly’s W result of .000, Sphericity was assumed to be violated. However, neither the Greenhous-Geisser test (.607) nor the Huynh-Feldt test (1.000) indicated a problem, so the Huynh-Feldt results were used for analysis of the two-way interaction.

The main effect of pitch showed a statistically significant difference in minimum identification duration between trials, $F(2, 6) = 15.42$, $p = .004$, with minimum durations decreasing from low tones (M = 63.9ms SD = 19.4ms) to middle tones (M= 51.7ms SD = 4.4ms) to high tones (M= 25.0ms SD = 7.2ms). Post hoc analysis with a Bonferroni adjustment revealed that pitch recognition minimum durations were statistically significantly shorter for high tones than middle tones (M = 26.7ms 95% CI [2.6, 50.7], $p = .038$). There was no significant difference in minimum durations for the main effect of position.

There was a statistically significant two-way interaction between relative pitch height and position, $F(4, 12) = 3.46$, $p = .042$ (see Figure 5.1). Therefore, simple main effects were run.

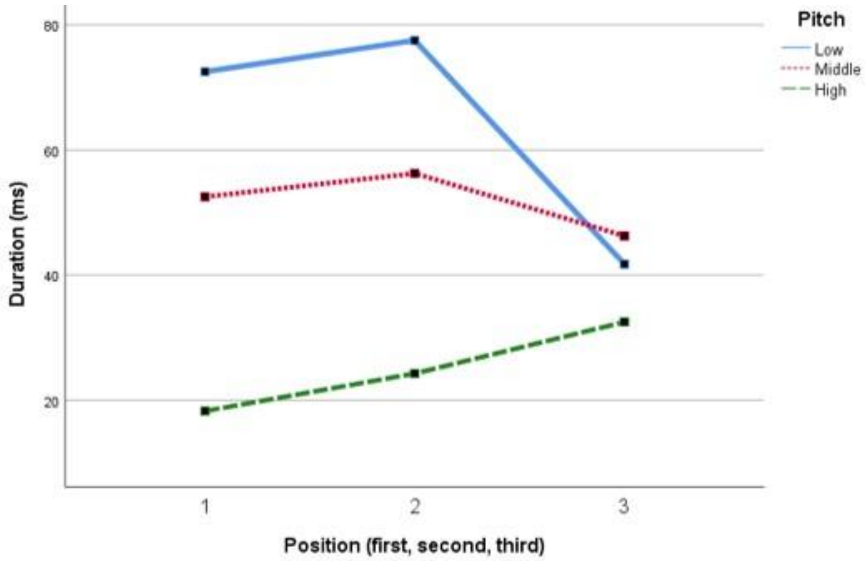
In the first position, longer durations were required to identify low tones (M = 72.5ms, SD = 18.9ms) compared to high tones (M = 18.2ms, SD = 8.3ms), a statistically significant difference of 54ms, 95% CI [2ms, 104ms], $p = .046$. Therefore, regarding hypothesis 3 -- *the lowest tone, when presented as the first tone in a 3-tone contiguous sequence requires longer durations for identification than the highest tone, when presented as the first tone in a 3-tone contiguous sequence* -- this hypothesis was supported by the data.

In the second position, longer durations were required to identify low tones (M = 77.5ms, SD = 28.7ms) compared to high tones (M = 24.3ms, SD = 15.9ms), a statistically significant difference of 53ms, 95% CI [19ms, 88ms], p = .014.

In the third position, identification duration was not statistically different depending on the relative pitch height (low, middle or high) of the test tone.

Figure 5.1

Line Graph showing Mean Minimum Durations for 3-Tone Sequences



Note: The line graph shows mean minimum durations for identification of pitch for tones in first, second and third position for relative pitch heights of low, middle, and high relative pitch.

Identification duration was not statistically different depending on the position (first second or third) of the test tone for any of the three relative pitch heights. Therefore, regarding hypothesis 4 -- *the third tone in a sequence of three tones requires less time for perception of pitch than tones in the first or second position* – this hypothesis was not supported by the data.

Low Tone and High Tone Identification Differences. To confirm that there was not a similar disparity between low tone and high tone identification in single tones, all trial results from the single tone condition were collated and divided into two groups. Stimuli requiring participants to identify tones above B4 were placed in one group (high tones) and stimuli using tones B4 and below in a second group (low tones). See Table 5.4 for the results of this tabulation. Mean rate for accuracy in pitch identification were not significantly different between high tones and low tones (t = 1.7, df = 3, p = .181, n.s.).

Table 5.4

Accuracy Identifying Low Tones vs High Tones

	Accuracy (percent correct N= 217)	
Participant	Low tones (n=117)	High tones (n=100)
participant 1	76.9	82.1
participant 2	81.0	90.0
participant 3	87.0	85.2
participant 4	71.4	92.9
mean	79.1	87.5
SD	6.6	4.8

Differences between Octaves, Major 7ths and Major 9ths. Considering other aspects of the overall data in more detail, identification differences based on the intervals between tones emerged. It was apparent that, where sequences included octaves, these were more often identified correctly than sequences that included intervals a semitone higher or lower than the octave. Data were collected on the success rate for identification of the three intervals for all tests (see Table 5.5).

A one-way repeated measures ANOVA was conducted to determine whether there was a statistically significant difference between octaves, major 7ths, and minor 9ths. There were no outliers in the data, as assessed by inspection of a boxplot. Identification errors were normally distributed at each interval level, as assessed by Shapiro-Wilk's test (p > .05).

Mauchly's test of sphericity indicated that the assumption of sphericity had not been violated, $\chi 2(2) = 4.43$, $p = .11$. Identification success was statistically significantly different for the three intervals $F (2, 10) = 11.98$ $p = .002$, partial $n^2 = .71$.

Table 5.5

Identification Error Rates (errors/total presentations) for Identification of Sequences that Included Major Seventh and Minor Ninth Intervals, Compared with Sequences that Included Octave Intervals Within 3-Tone Sequences

	Major 7th		Octave		Minor 9th	
Section	n	Rate	n	Rate	n	Rate
First tone low	61	0.52	111	0.22	80	0.48
First tone middle	40	0.21	25	0.25	2	*
First tone high	62	0.38	44	0.07	54	0.38
Second tone low	43	0.65	8	0.14	0	*
Second tone middle	36	0.33	107	0.29	87	0.45
Second tone high	27	0.35	11	0.10	9	0.29
Third tone low	45	0.22	89	0.25	64	0.33
Third tone middle	51	0.24	18	0.38	1	*
Third tone high	38	0.27	33	0.10	40	0.25
Mean error rate (SD)		*0.35 (0.15)*		*0.20 (0.10)*		*0.36(0.09)*

**Note*: intervals with n < 8 were not included in the analysis.

Using a complex contrast with Bonferroni adjustment to test for differences between octaves and intervals a semitone either side of the octaves, there was a statistically significant difference in identification success between octaves ($M = .20$, $SD= .10$) and the mean for the major seventh interval ($M = .35$, $SD= .15$) and minor ninth interval ($M = 36$, $SD= .09$), a mean difference of 0.18, 95% CI [0.04, 0.32], $p =.019$. Because of the random allocation of pitches in the stimuli, intervals between the stimuli varied across all duration levels and pitch orders (see the ‘N’ columns in Table 5.5).

2-Tone Condition

Results from the 3-tone condition of the current study (see above) found that the minimum duration required to identify the pitch of short 3-tone contiguous sequences ranged from 18ms to 77ms with a mean minimum duration of 48ms. Low tones required longer durations for identification than high tones when they occurred in the first or second position of the three tone positions.

As anticipated, the MDPP of 3-tone contiguous sequences fell between the 8 wave cycles (i.e. 8.0ms B5eq) required for identifying melodic pitch in an isolated condition (Patterson et al., 1983) and the 160ms required for recognition of a cyclic melody five to nine tones in length (Warren et al., 1991). It would therefore seem likely that the MDPP of tones in a 2-tone sequence, is longer than that required for identifying the pitch of single tones presented alone, but shorter than that required for identifying pitch of a single tone within 3-tone sequences. Possible reasons why tones in 2-tone sequences might require longer durations for perception than single tones include the effect of masking (both forward and backward masking) (Massaro, 1975), the integration of the two sounds (where the second tone is perceived as a continuation of the first tone) (Ciocca & Darwin, 1999) and the smearing of tones (Schouten, 1962) (the "Discussion" section for a detailed explanation of these effects).

Following on from the 3-tone study condition, the following research questions were addressed:

1) What is the minimum duration for perception of pitch (MDPP) in 2-tone sequences?
2) Does the position (first or second) of the target tone affect the minimum duration required for perception of pitch in 2-tone sequences?

3) Does the relative pitch (low, middle or high) of the target tone affect the minimum duration required for perception of pitch in 2-tone sequences?

Materials for this condition were the same as described for the pilot study (above), except that stimuli were created at 18 durations. These durations were: 200ms, 100ms, 70ms, 60ms, 50ms, 45ms, 40ms 35ms, 30ms, 25ms ,20ms, 15ms, 10ms, 8ms, 7ms, 6ms, 5ms and 4ms. The general procedure was the same as that used for the single tone condition. However, instead of a single stimulus set, four stimuli sets were presented to the participants, thus making four trial runs. Each time, different combinations of sound orders and note positions were presented for identification. The four participants were randomly allocated to two different presentation treatments (A and B) to minimise the effects of learning, adaptation, and fatigue. Participants in both treatment groups completed the same trial runs but in a different order.

The two participants in treatment group A were presented with the series of trial runs in the following order: Identify the first tone when it is the highest tone, then when it is the lowest tone, and then identify the second tone when it is the highest tone, then when it is the lowest tone. The two participants in treatment group B were presented with the series of trial runs in the same order for identifying the tone position but in reverse pitch height order i.e. identify the first tone when it is the lowest tone, then when it is the highest tone, and Identify the second tone when it is the lowest tone, then when it is the highest tone.

The target tone that required pitch matching was identified before each phase of this condition started. Participants were instructed to identify either the first tone or the second tone.

Results

The aim of this study condition was to establish the MDPP in 2-tone contiguous sequences. Four trial runs were completed by all participants. The mean minimum duration for identifying the pitch of a tone in a contiguous 2-tone sequence over all four trial runs, was 24 ms. However, within this result there were wide differences associated with the tone's position in the sequence. Therefore, a more accurate summary is that the minimum duration for the identification of pitch in a 2-tone sequence ranges from 17ms to 35ms depending upon the position within the sequence of the tone being identified. The shortest mean MDPP in a 2-tone contiguous sequence was 17ms to identify the second tone when it was the lowest pitch of two contiguous tones. The longest mean MDPP in a 2-tone contiguous sequence was 35ms to identify the first tone when it was the lowest tone. See Table 5.6 for details of the results for all trial runs.

Table 5.6

Minimum Duration (ms) for Identification of Tones in a Series of Two Tones

P.No	*1stLow*	*1stHigh*	*2ndLow*	*2ndHigh*
1	35	20	5	15
2	10	8	8	15
3	50	30	20	30
4	45	30	35	35
Mean	35	22	17	24
SD	18	10	14	10

Note: "P.No" = Participant number. "1st", "2nd", and "3rd" refer to the order of the identified tone in the sequence. "Low" and "High" refers to the relative pitch (low or high) of the target tone.

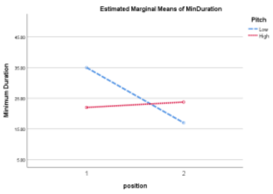
There were no outliers, as assessed by examination of studentized residuals for values greater than ±3. Minimum durations were normally distributed, as assessed by Shapiro-Wilk's test of normality on the studentized residuals ($p > .05$). There was a statistically significant two-way interaction between pitch height and position, $F(1, 3) = 17.29$, $p = .025$. Therefore, simple main effects were run.

In the first position, longer durations were required to identify low tones (M = 35ms, SD = 17.79ms) compared to high tones (M = 22ms, SD = 10.46ms), a statistically significant difference of 13ms, 95% CI [1ms, 25ms], $p = .043$. There were no other statistically significant differences. There was also no main effect for either pitch height or position.

Although not shown in the results as significant, it can be seen from the dashed blue (or ‘Low’) line in Figure 5.2 that there is a weak trend towards low tones being “easier” to identify (i.e. at shorter exposure durations) in the second position than the first position.

Figure 5.2

Line Graph showing Mean Minimum Durations for 2-Tone Sequences



High tones remain relatively unchanged between position 1 and 2, whereas low tone minimum durations change from 35 ms to 17 ms: a decrease in Mean minimum identification durations of 50%.

Discussion

The minimum duration participants required to identify the pitch of an isolated tone (using the method of matching a test tone against a target tone) was very similar to results found by other researchers (Patterson et al., 1983; Robinson & Patterson, 1995). The results from the 3-tone and 2-tone study conditions are therefore directly comparable to previous research into single tone perception.

Although mean overall MDPP in a 3-tone sequence in this study was 47ms, a more accurate description is that minimum durations ranged from 18ms (B5eq) to 78ms, depending upon the position within the sequence of the tone being identified. Lower tones located early in the sequence required the longest duration for identification (73ms for the lowest tone in the first position and 78ms for the lowest tone in the second position) and higher tones located early in the sequence required the shortest durations for identification (18ms for the highest tone in the first position and 24ms for the highest tone in the second position). The minimum duration required for identification of pitch in 3-tone contiguous sequences was found to be longer than the 4ms minimum duration required for identification of the pitch of a single tone in the 1khz to 4khz frequency range and shorter than the 160ms minimum duration required for identification of pitch in continuous cyclic sequences.

As anticipated, the MDPP of 2-tone contiguous sequences (24ms) was between the 8 wave cycles (i.e. 8.0ms B5eq) required for identifying melodic pitch in an isolated condition (Patterson et al., 1983) and the mean 48ms required for identification of pitch in 3-tone sequences. As was the case for the 3-tone condition of the study, low tones (relative to the other tones in the sequence) that were positioned early in a contiguous sequence of tones required longer durations for identification than high tones occurring early in the sequence. It was concluded from these results that perception of the pitch of low tones is masked by higher frequency tones played immediately after the low tone.

For high tone mean results (see Figure 5.2), there appears to be little difference between identification in first position and second position, although there is a small increase in minimum duration identification. In the 3-tone condition, there was evidence of a straightforward progression from shorter to longer minimum identification durations for high tones when identifying the first, second and third tone (although individual scores did not reflect this). This theory, although not strongly supported, is not contradicted by the results of the 2-tone condition (compared to the 3-tone condition) because three of the four participants also required longer durations to identify the high tone in second position than in the first position.

The minimum duration required to identify the last tone in a sequence seems less affected by the pitch of the preceding tones than tones in other positions. This indicates that forward masking is not relative pitch height dependent as backward masking seems to be.

As expected, the MDPP for tones in all positions of the 3-tone sequences was longer than for single isolated tones. However, the difference between tone identification in 2-tone sequences and single isolated tones was not statistically significant. This may be an accurate result but may be a type II error, resulting from low power due to small participant numbers. However, it is worth noting that some participants were able to identify the pitch of tones in the 2-tone sequences at a duration similar to the MDPP of a single isolated tone, which implies that the result of no significant difference between the MDPP for 2-tone sequences and the MDPP for single tones is accurate.

All minimum duration results for perception of pitch in 2-tone and 3-tone sequences were much shorter than the minimum duration required for identifying the pitch of tones in a cyclic sequence. As predicted, the limitations related to theta waves are clearly not applicable for tones in 2-tone and 3-tone sequences. As outlined in the literature review, the theta band of around 5 Hz (a wave every 200 ms) is used to transmit pre-processed information to higher cognitive centres such as the auditory cortex and short-term auditory memory. The shortest time by which a theta wave can reset is 125 ms and the longest a theta wave can last is around 250 ms. Tones in a cyclic sequence required the duration of at least a theta wave for identification of pitch (Warren, Gardner, Brubaker, & Bashford Jr, 1991) as well as for tone ordering (Thomas & Fitzgibbons, 1971). However, tones in 2-tone and 3-tone sequences were identified at shorter durations than 125ms in all positions (first, second and third) and relative pitches (highest, middle, and lowest).

Differences in Levels of Masking Between High Tones and Low Tones

In the 3-tone condition, low tones in the first position required nine times the minimum duration required for identification of the pitch of a single isolated tone but high tones in the first position only required twice the duration of a single isolated tone for identification of pitch. As can be seen from this result, the amount of interference, or masking, of tones in this study has differed significantly, depending upon the position of the target tones in the sequences. In this study, lower tones were subject to higher levels of masking than were higher tones in the first and second position in both the 2-tone and 3-tone conditions. Because these two positions (first and second position for 3-tone sequences and the first position for 2-tone sequences) all involved backward masking by the following tones, it can be inferred that, in the backward masking condition, lower tones required longer durations for identification than higher tones. Moreover, because lower tones in this study were always followed by higher tones, it can be concluded that higher tones have a stronger backward masking effect on the preceding tone than lower tones. This result supports findings from Kallman and Massaro (1979), Watson, Wroton, Kelly, and Benbassat (1975), and Divenyi and Hirsh (1975), who found that test tones in a backward masking study were harder to identify if they were followed by a higher frequency tone than if they were followed by a lower frequency tone.

Three potential explanations for this effect are considered. They are not mutually exclusive, and all may have contributed to the minimum duration results from this study.

Preconscious Adjustment for Perceived Harmonics. Ciocca and Darwin (1999) reported that higher frequency tones presented after a primary tone can affect the perceived pitch of the primary tone to a greater extent than higher frequency tones presented before a primary tone. In their study, the perceived pitch of the primary (or lowest) tone heard first in a sequence was perceived as higher or lower that its actual pitch, depending upon if the higher tones following the primary tone were higher or lower than the frequency that would be expected from the natural harmonic series of the lower (or primary) tone. In Ciocca and Darwin's study, the non-simultaneous components were added immediately following the target tone in one experiment, and between 20ms and 160ms after the target tone in the second experiment. In all but the 160ms condition, perceived pitch of the primary tone was changed by the non-simultaneous components, although the effect gradually decreased as gap durations increased. In both experiments the target tone and non-simultaneous components were each 90ms duration with 5ms ramps.

This effect would account for the longer MDPP of lower frequency tones when they occurred first or second in the sequence, because they would likely be affected by higher tones. As tone durations increased, the pitch of the following tones had a decreasing effect on the lower frequency tones. When the lower frequency tones appeared in the last position, there was much less of an effect on the MDPP, because there were no higher frequency tones following them to cause a skewing of pitch perception. Conversely, a lower frequency tone would be unlikely to affect perception of a high frequency tone that followed (or preceded) it because lower frequency tones are never harmonics of higher frequency tones and therefore would not be preconsciously allocated to that role. This means that, under this theory, a lower frequency tone is less likely to affect the perceived pitch of a high frequency tone. This was confirmed in the study, where higher tones were comparatively unaffected by lower tones in the sequence.

These findings are supported by results from other studies. Grose et al. (2002) reported that non-simultaneous tones can be integrated and perceived as if the two tones were a single tone, even when the tones are presented up to 45ms apart. In an earlier study on masking, Hawkins et al. (1974) reported that the tonal information participants used to identify test tones contained components derived both from the test tone and from a retroactive masker. This was especially evident when the relative pitch of the masker was unknown.

Expected Intervals Causing Errors. There were statistically significantly fewer errors identifying tones in sequences that included octaves than with tones in sequences that included jumps either a semitone short of an octave or a semitone beyond the octave (i.e. major sevenths and minor nineths - see results section: Table 5.5). The octave interval is the only interval that is prevalent in music from around the globe and thus is recognised by all humans (McDermott & Oxenham, 2008, p. 3). It is therefore likely an inbuilt characteristic in humans and, consistent with this, there is evidence for a cognitive “Octave template” in the structure of the brain (Demany & Semal, 1988). In addition, Graves and Oxenham (2017) found that learned tonal hierarchies influence the accuracy of pitch interval perception. Given Graves and Oxenham's findings, the current results lend support to the theory that learnt material that regularly includes an octave will prime participants to expect octave intervals (Krumhansl, 1995). Therefore, participants would tend to choose a tone that matched an octave interval, if selection of the correct tone was in doubt. To avoid this problem in future, stimuli could be prepared that did not include the octave interval, thus ensuring equal difficulty for all stimuli.

Stream Segregation Assisting High Tone Identification. Another possible reason for the large difference between responses to lower tones and higher tones in the first and second position is the likelihood that high tones can be isolated as a different sound stream and processed individually, whereas low tones cannot. Sounds having some feature that distinguishes them from other sounds (such as the highest tone in a sequence) can be consciously grouped into a different cognitive stream, allowing the tones in that stream to be analysed independently from the other sounds (vanNoorden, 1975). This phenomenon has been called the 'cocktail effect', where an individual person can be heard and understood, despite a large amount of background noise. This extra, undisrupted, processing time would explain why high tones in the first position of the 3-tone sequences were identified by all participants at durations of 25ms or less. For some participants, minimum durations for the highest tones in the first position were close to the MDPP of a single isolated tone. If these high tones had been processed together with the other tones, it is unlikely that they could be identified at such short durations. Schouten (1962) demonstrated that tone sequences with segments shorter than a 50ms beta wave sound like they are smeared together, and this would make identifying the pitch of these short contiguous tones almost impossible. Stream segregation might therefore explain why the high tones in this study did not suffer from this smearing effect; and the minimum durations recorded for pitch percepiton of the highest tones (occurring in the first position), in the 2-tone and 3-tone conditions, provide evidence that these tones can be isolated and processed independently. When tones sound similar, such as in sine wave sequences as used here, streaming can only be achieved through a conscious effort (Van Noorden, 1975). This requires practice and may explain why not all participants were able to achieve minimum durations close to the MDPP of a single tone. In short, participants were not sufficiently skilled at stream segregation to isolate the high tone successfully at such short durations.

Conversely, low tones are strongly affected by other tones, making it very difficult for them to be isolated and placed in a separate stream. Therefore, low tones in the first position are unlikely to benefit from stream segregation and thus require much longer durations for identification of pitch.

Forward masking

In the current study, forward masking was evident for all tones in the second and last positions. Although tones in the last position suffered exclusively from forward masking (having no tones being played after them, and thus no backward masking), responses to these still showed evidence of masking. This result runs contrary to results from studies that found that forward maskers assisted target tone recognition (Leshowitz & Cudahy, 1973; Massaro, 1970; Ronken, 1972). However, as discussed in the literature review, the earlier findings likely relate to the special condition of the masking tone being placed between the frequencies of the test tones, allowing participants to make an ‘up’ ‘down’ judgement rather than a ‘high’ or ‘low’ judgement. The results from this research may reflect the higher accuracy of the method used in this study over the ‘high vs low’ method, which was used in the above-mentioned studies reporting that forward masking assisted target tone identification.

High tones in the last position required slightly longer durations for identification than did high tones in the first and second positions. This may indicate that high tones are influenced more by forward masking than by backward masking. It may be that, for high tones, pitch perception is disrupted by information arriving before the tone more than information arriving after it. This may be a result of participants identifying pitch based on expected intervals from pervious tones (see “Expected intervals causing errors” above).

The Benefit of Silence (or a Gap) After a Tone.

As described in the literature review, Watson et al. (1975), commented that the last tone in a sequence, as well as tones that came immediately before a 40ms gap, were more easily identified than all the other tones. They stated that the reason for this is that these tones have access to more undisturbed processing time than other tones in a sequence.

In the pilot study, tones in the last position were significantly easier to identify than tones in the other two positions. However, this effect was not found in the main study. A possible reason for this disparity is that, in the pilot study, the relative height of the tone (highest, middle or lowest) was not known to the participant prior to the presentation of each stimulus. This meant that the participant had to focus attention more generally because he did not know in what direction (up, down or changing direction) the tone sequence would go and whether the target tone would be the highest tone, the middle tone, or the lowest tone. This more general focus may have had the effect of magnifying the benefit of silence for identifying the last tone. In the main study, participants had the benefit of knowing beforehand the relative pitch (low, middle or high) of the target tone, thus allowing them to focus on the expected pitch area for identification of the tone. This may have reduced the benefit of the silence after the last tone, for the study involving four participants.

The minimum duration required to identify the last tone in a sequence did not seem affected by the pitch of the preceding tones. Two reasons are proposed to account for this. The first is that the gap after the last tone allowed for extra processing of the last tone, thus enabling more accurate results for all relative pitches. Secondly, the issue of preconscious adjustment of low tones to accommodate subsequent high tones as harmonics does not arise for the last tone because there are no subsequent tones to cause this effect. Because the last tone was only affected by forward masking, it was concluded that forward masking levels are not affected by the frequency of the target tone, compared to the other tones in each stimulus.

Although commonly known as a “recency effect”, the benefit from being the last tone in a sequence is more likely a processing benefit from the silence after the last tone, as described above. This is because at the shorter durations (below 50ms), all tones in the stimulus would reach the memory areas of the brain at around the same time (as part of the same theta wave). Therefore, because all the tones would be processed simultaneously, the classic recency effect would not apply.

Tone Ordering Results that Conflict with Results from this Study

Regarding research by Warren (1972), who found that the first and last sounds in a sequence were more easily identified than the middle sounds, this was not generally the case here. Rather, in general terms, the highest tone was the most easily identified. Contrary to Warren’s results, the first tone (when it was the lowest frequency tone in the sequence) was one of the hardest tones to identify. Additionally, as mentioned before, Massaro’s (1970), Ronken’s (1972) and Leshowitz and Cudahy’s (1973) findings, that forward masking improves pitch perception accuracy, were not supported by this study.

Confirming the Method Used to Ensure Equivalent Durations for High and Low Tones

The method used in this study to ensure perception equivalence for all frequencies was to use the same durations for stimuli of every pitch from segment durations from 200ms to 40 ms and then to reduce durations asymmetrically until wave cycle parity was reached at 10 wave cycles (10ms B5eq), after which stimuli were reduced by step according to the number of waves. To ensure this method was effective, results from the single tone condition were checked to confirm that there was no disparity between low tone and high tone identification. Mean rates for accuracy in pitch identification were not statistically significantly different between high tones and low tones and it was therefore concluded that this method had successfully provided parity for tones of all frequencies in the study. This outcome also provides evidence that the longer durations required for identification of pitch in lower tones in the first and second position were not caused by a disparity in minimum recognition durations. Further evidence of this is provided by results from the third position in the 3-tone study. In the third position in this study, lower tones averaged shorter durations for recognition than middle tones (see Figure 5.1). Additionally, there is evidence from other studies; thus, Doughty and Garner (1947, p. 355) reported in their pitch threshold study, that sine wave tones of the same frequency as the low range of this study (250 Hz) required only around four wave cycles (16.8ms) to identify pitch of single isolated tones, compared to 10 wave cycles (10.2ms) to identify pitch of single isolated tones in the B5 equivalent frequency range (1000 Hz). The four wave cycles minimum for low tones is considerably shorter than the equal number of waves cycles for all frequencies provided in this study.

Based on all the information provided above, it was concluded that it is unlikely that, when compared to higher tones, low tones require more than the same number of wave cycles for pitch identification.

Conclusions

The method used for identification of short tones in this study is directly comparable with the results reported by other researchers.

The mean overall minimum duration for perception of pitch (MDPP) in this study was 24ms for 2-tone sequences and 47ms for 3-tone sequences. However, because of the effect of differences in masking levels depending upon the relative pitch and position of each tone, a more accurate description is that minimum durations ranged from 17ms (B5eq) to 35ms for 2-tone sequences and from 18ms (B5eq) to 78ms for 3-tone sequences. The relative pitch and position in a sequence of each tone affects minimum durations in predictable ways. In both cases, lower tones located early in the sequence required the longest durations for identification and higher tones located early in the sequence were identified at the shortest durations indicated (above).

It was concluded that backward masking of low tones by high tones was primarily caused by preconscious adjustment of low tones, to accommodate subsequent high tones as harmonics of the lower tones. Two other factors were identified as potentially accounting for differences in minimum duration for identification of high and low tones. The first was the possibility of stream segregation of higher tones, which allowed the high tones to be isolated and processed separately. The second was misidentification of tones based on the preconscious assumption that the tone sequences would follow common patterns or comprise common musical intervals.

The type of masking (forward masking or backward masking) that tones were subjected to clearly influenced minimum durations for high and low tones. Backward masking had a greater effect on low tones than high tones, but the minimum duration required to identify the last tone in a sequence was less affected by the pitch of the preceding tone.

Two reasons are proposed to account for this. The first is that the gap after the last tone allows for extra processing of the last tone, thus enabling more accurate results. The second reason is that preconscious adjustment of low tones to accommodate subsequent high tones as harmonics is not relevant to the last tone in a sequence because there are no subsequent tones. Therefore, it was concluded that forward masking levels are not affected by the relative pitch (highest, middle or lowest) of the target tones.

Based on this study and other research, the known sequence of minimum durations for perception of pitch from single tones to cyclically repeated tones is unidirectional but irregular. The average MDPP tripled from single tone identification to identification of tones in 2-tone sequences and then doubled from 2-tone sequences to 3-tone sequences. It is presumed that, at some point, asymptote will be reached at 160ms per tone segment, but because this progression is neither linear nor exponential, there is too little information to be able to predict minimum durations for 4-tone and 5-tone sequences from this sequence of minimum durations.

The highest tone in 2-tone and 3-tone sequences is not easily masked and can be identified at comparatively short durations, although high tones occurring early were identified at slightly shorter durations than high tones occurring in the last position.

If it is important for lower tones to be perceived accurately, they would either need to be placed last in a sequence or be surrounded by tones that matched the lower tone's harmonic series. Conversely, if a desired effect was to modify the pitch perception of a lower tone, then placing a high tone that is outside of the low tone's harmonic series immediately after the low tone would likely affect the perceived pitch of the low tone.

Limitations

Although four participants is generally considered a very small participant sample, for many short tone studies, small participant numbers (of between one and seven) have been assumed to be sufficient-to-purpose (Divenyi & Hirsh, 1974; Doughty & Garner, 1947; Green, 1973; Massaro, 1974; Patterson et al., 1983; Robinson & Patterson, 1995; van Noorden, 1975; Wier & Green, 1975). It is likely that the use of small numbers of participants has been due to the difficulty of recruiting participants, considering the large commitment of time required for attaining the prerequisite skills to approach minimum durations for identification – as was the case here also. It is also recognised that reliance on a small sample limits generalisation of any results. Nonetheless, it was considered sufficient to have four highly musically skilled participants for this study, given that such expertise should at least assist to test the lower limits of perceptual capacities. It is worth noting here that despite the small numbers of participants, the effect size found in the results was large enough to generate significant results for even four participants.

Because of the small number of participants in this study the likelihood of type II errors (i.e. false negative results) was high. For example, the second tone in the middle pitch position was identified at minimum durations similar to low tones in the first and second position except for one participant. If this result was an outlier caused by chance, then there would also be a significant difference between middle tones and high tones in the second position. Another example is the non-statistically significant finding for overall position in the 3-tone study condition. If results from a larger pool of participants followed the same trend of minimum durations for identification of pitch, the average MDPP of tones in the third position would be significantly shorter than for the other two positions.

The amount of practice completed by individual participants before testing may have affected the result. However, the amount of practice required may also depend on the prior experience and training of a participant. As a skills-based study (i.e. being able to extract a tone from a tone series, convert it to a synthesised memory, hold the sound in memory and compare it to a later sound of a different timbre and duration), a participant's level of innate and acquired skill will have undoubtedly affected the amount of practice of the specific task required for this study. Practice ranged from around 20 minutes (albeit from the participant who scored the lowest minimum durations in most tone positions) to over 20 hours (anecdotal reports). Therefore, minimum durations may not necessarily relate to the amount of practice completed. However, it is likely that practice will improve results for most people and thus the true limen for MDPP in 3-tone contiguous sequences may be lower than the results from this study have suggested, because some participants may not have had sufficient practice to reach identification asymptote. Nonetheless, results were sufficiently consistent to generate statistically significant differences.

Because of the need to avoid identification strategies such as sound analogue comparison, stimuli were created from random frequencies and intervals. Some intervals were harder to identify than others, as has been established by the difference in identification percentages between octaves and intervals a semitone above and below this level. There may be other intervals that interfered with the data in a similar fashion, but that did not appear as significant in this study but may appear with larger numbers of participants.

This study only covered tones from C4 to C6 (261.5Hz to 1046.5 Hz) and tones in a different frequency range may give different duration results. It is theorised that those tones consisting of frequencies lower than the range tested will retain the same B4 equivalent duration (i.e. equal wave cycle durations). Conversely, it is likely that tones consisting of frequencies higher than the range tested will require more wave cycles, but minimum durations should remain similar to the B4 tone. This theory is based on results from single tone studies that tested a broader frequency range (Doughty & Garner, 1947).

Because this study used a small, highly skilled group of participants, population mean minimum durations will likely be very different to the minimum durations found in this study. In the lead up to this research, more than 20 people initially agreed to participate but only four were able to identify single tone stimuli consistently, at or below the 10ms cut-off duration for participation in the study. This implies that the population average for untrained participants will be much longer than durations identified in this study.

Recommendations for Further Study

The duration at which the number of segments in a sequence of tones no longer affects the MDPP is not known and could be established using similar methods by increasing the number of segments in a sequence until average minimum duration of tone segments reaches 160ms, at which point the MDPP should asymptote.

It may be worthwhile examining the extent to which integration of surrounding tones (especially higher tones) affects pitch perception, using a study focused on this effect. This may be of benefit in aural training because it could provide insight into mistakes students make when identifying intervals. Studying this effect may also assist in improving sound compression techniques, speech recognition and speech synthesis (This will be discussed in greater detail in the Conclusions Section).

To avoid in the future the issue of participants instinctively preferring octave intervals, stimuli should be prepared that do not include the octave interval, thus ensuring a more equal difficulty level for all stimuli.

Chapter 6 Study 3. Aural Training Study

In Chapter 5, the minimum duration for perception of pitch in 2-tone sequences was identified as ranging from 17 ms to 35ms depending upon the position of the tone in the sequence. In this chapter this information was used to create an aural training program designed to assist with learning of musical intervals. The aim was to improve educational outcomes for a group of year 8 general music students. It was theorised that the use of short tones would improve students' skill in identifying musical intervals. The benefits of a system that used tone durations that approach the limits of perception include temporal occlusion training, where decreasing amounts of stimuli help focus attention on important details; more regular repetition of stimuli due to increased presentation rates; and greater motivation to stay on task due to interesting, appropriately challenging, and varied exercises. It was hypothesised that *students using the new aural training program designed for this research would learn musical intervals more effectively than students using the current computer-based interval training program*.

Rationale for Providing an Improved Aural Training Method

A central motivation for identifying the limits of pitch perception for very short tone-durations is that pedagogical strategies are most likely to be effective if high demands are placed on the neural mechanisms of pitch perception (Patel, 2012). In the case of this research, decreased tone durations, compared to those normally experienced in tone sequences, provide these higher demands on the nervous system that handles pitch perception. This conclusion was based on Patel's OPERA hypothesis, which sought to explain findings by other researchers that music training enhances neural encoding of speech sounds in the auditory brain stem. Patel reasoned that musical training required auditory processing networks to function with higher precision than generally needed for ordinary speech communication and that this higher precision, through the vehicle of functional neuroplasticity, drove improvements in language skills. Using this reasoning it can be predicted that for pedagogical training strategies to benefit the neural encoding of pitch, training should place higher demands on the nervous system that handles pitch perception than is required under normal listening conditions.

Aural training has long been considered an integral and highly important part of music education (Buehrer, 2000; Thackray, 1975). McPherson (2005) also emphasized the importance of aural abilities to musical practice. He identified playing by ear, improvising, and playing by memory as valuable skills for musicians. An important area of aural training is pitch matching and interval recognition. Interval recognition is the main process by which melodies are identified (Dowling & Bartlett, 1981). Aural pitch matching is a requirement of the year 3-4 Australian Curriculum (acara.edu.au, 2018, p. 34) and recognising musical intervals is included in the year 7-8 Australian Music Curriculum Document. Interval recognition is also important for music teachers and conductors. Stambaugh and Nichols (2020) found a strong correlation ($r = .75$) in preservice teachers between skill in identifying intervals and skill in identifying errors in melodies. Error detection of this type is an important skill for identifying and correcting problems during ensemble rehearsals. Therefore, creating an improved aural training program was deemed to be a worthwhile pursuit.

However, aural training is a repetitive activity, and students have told me repeatedly that they find it boring. In general, progress is incremental, and it involves being repeatedly exposed to a variety of unrelated musical excerpts to improve your ability to recognise, analyse and replicate accurately variations in pitch, rhythm, dynamics, timbre, texture, and form. Despite these issues, it is generally recognised that aural training is a valuable activity for learning music and having ‘a good ear’ (the goal of aural training) is an essential aspect of musical ability (Buehrer, 2000; McPherson, 2005; Thackray, 1975). The aim of this study was to provide an interesting game-like method that can adjust to every level and rate of improvement, include optimum training methods, and thus maximise learning. The targets for this study were students who struggle with one of the fundamental requirements of aural training; recognising simple musical intervals.

There are several ways in which students can be taught interval recognition skills. A widely applied method utilizes the computer program *Auralia* (Rising, 2016). This program provides graded aural training, including training of musical interval recognition. Its computer-based method can be contrasted to the more traditional approach of aural training, where exercises are provided to a whole class at the same difficulty level. This whole-class approach restricts the ability of a teacher to provide developmentally appropriate material to all students because of the wide variety of achievement levels found in most classes. While some students will find a particular task appropriate for their development level, most students in the class will find the task either too easy or too hard. A computer-based training program can therefore provide benefits over this whole-class approach because progress can be modified for each student, thereby continuously providing training at or close to an individual’s current development level. However, the program *Auralia* has limitations in providing training at a very basic level. Although it is reported by the creators to be “the most comprehensive ear training software available” (MusicEdNet, 2018), some students are not able to achieve successful results at even the most basic level provided by the *Auralia* computer program (see the “development of the aural training program” section below).

Limited research has been published on the effectiveness of computer-based aural training studies. Kariuki and Ross (2017) randomly divided a group (N=20) of the first author's private music students into two groups. The treatment group was taught for five 10-minute sessions using a computerised ear training program, while the control group was taught for five sessions using traditional, non-computerized ear training methods. At the end of the five sessions, students were tested. Kariuki and Ross found that the treatment group performed significantly better than the control group after the five session (Kariuki & Ross, 2017). The effect size of the method d = 1.62, was very large compared to other education studies, as will be seen below. However, a larger study (N=58) found no significant difference in improvement scores between computer-based aural training and standard aural teaching (Ozeas, 1992). Moreover, when the data were examined more closely, Ozeas found that students with low initial scores did very poorly with the computer training program compared to the teacher-based method. In short, it is possible that success with different forms of training is mediated by initial level of skill. The implications of this will be discussed further below and in the conclusions section of the thesis. Hofstetter (1981) compared the results from computer based aural training against audiotape-based training (N=33). He also reported a significantly better outcome for computer-based training, but the effect size was not reported. Thus, because only three studies were found that have directly investigated computer-based aural training, with mixed results reported, computer-based training from other areas of education was also examined, to try to ascertain to what extent computer-based training could be expected to be more effective, if at all, than classroom face-to-face teaching. This was done using information from Hattie's (2008) meta-study.

Hattie completed a meta study of education meta-studies, including computer-based training (computer assisted instruction) and reported effect sizes. Effect size was calculated using two methods. For more effective outcomes in treatment vs control studies, effect size was calculated as: d = [Mean treatment – Mean control]/SD. For measurement of student improvement over time using pretests and posttests, effect size was calculated using: *d* = [Mean treatment – Mean control]/SD and *d*= [Mean end of treatment – Mean beginning of treatment]/SD. In both cases SD was the pooled sample standard deviation (Hattie, 2008, p. 8). Hattie commented:

> An effect size of *d* = 1.0 indicates an increase of one standard deviation on the outcome—in this case the outcome is improving school achievement. A one standard deviation increase is typically associated with advancing children's achievement by two to three years, improving the rate of learning by 50%, or a correlation between some variable (e.g., amount of homework) and achievement of approximately r = 0.50 (Hattie, 2008, p. 7)

Hattie reported an overall effect size of 0.37 for computer-based training, compared with other traditional methods, was based on the combination of 81 meta-studies covering over 4000 studies. He recommended that an effect size for a substantive improvement in student outcomes be 0.4, deriving this figure from the mean effect size of all variables in his meta-analysis and suggesting that this be used as the benchmark to judge the success or otherwise of new educational methods when compared with pre-existing method.

On the basis of the foregoing discussion of computer-based training it seemed possible that some negative aspects of computer-based training might outweigh the putative benefit of individualised progress. To counter this issue, the new training program to be reported now, sought to improve the effectiveness of a computer-based approach by using a strategy that incorporates a variety of benefits (as described below).

How Students Develop Aural Skills

There are a variety of methods and strategies that students can use to develop their skills in musical interval training. One popular strategy is to use familiar melodies, where the first two notes in a familiar melody form one of the intervals from the training session. Common ‘interval melodies’ include: Major second: *Happy Birthday,* Minor third: *Greensleeves,* Perfect fourth: *Mexican Hat Dance,* Perfect fifth: *Star Wars* theme. Students try to hear the song in their head and match the first two tones to the interval being presented in the training session.

Another method is to use a major or minor scale and attempt to locate the interval on that scale. For example, If a perfect fifth is presented in the training session, the student might start on the lowest note and sing the musical scale up to the fifth note (i.e.: “1, 2, 3, 4, 5”). The assumption underpinning this method is that the student will recognise that the first and fifth note in the scale are the same as the two tones in the interval example presented.

Both of these methods are effective ways of learning intervals in controlled conditions, but when faced with obstacles, such as presenting tones using unfamiliar instruments and/or outside the singing range of the participant, they are of less functional use. A more efficient method is to become so familiar with the intervals that recognition aids are no longer necessary. Reaching this level is one of the primary purposes of interval training; and, consistent with discussion outlined on p. 155, the current author concluded that temporal occlusion training should increase the rate at which students achieve this ideal level. Specifically, it was theorised that constraining the duration of an interval stimulus would force participants to focus on the skills of frequency identification and ratio matching (e.g. a perfect fifth contains two tones with a frequency ratio of 2:3), rather than using alternative strategies such as singing along with the stimulus and then attempting to match their singing with various interval songs.

Development of a New Aural Training Program

Young people come to music classes with a variety of abilities. Across many years I have regularly encountered students who could not identify intervals at the simplest level. With such cases I had found that, when using the aural training program “Auralia”, no improvement would be apparent despite weeks of practice. Struggling students would continue to score at a rate consistent with random guessing (around 50% correct at the first level of training). This is consistent with results from Ozeas (1992), who found that lower skilled students fared poorly when using computer-based methods. The intent for developing the program for this study was to structure computer-based training in a more incremental way to facilitate progress for all students. As will be seen later, this met only limited success.

There are several possible reasons for students’ failure to progress. The first is that the students were tone-deaf. However, Cuddy, Balkwill, Peretz, and Holden (2005) found that most students who are unable to match pitches or identify intervals have no physiological reason for not being able to do so. These researchers found that, by university-entrance age, the percentage of adolescents believing that they are ‘tone deaf’ is around 17%, whereas only 4% of people experience tone-deafness for which a neuroanatomical cause can be identified (Kalmus & Fry, 1980). Such cases may suffer from a condition termed *amusia*, which renders them unable to recognize (and therefore identify) small changes in pitch. However, there is evidence that poor sensitivity to differences in pitch can reflect learning opportunities. Pfordresher and Brown (2007), in their study on poor pitch singing, have confirmed that some people do not suffer from amusia and yet have trouble discriminating tones. However, Santos, Joly-Pottuz, Moreno, Habib, and Besson (2007) established that pitch sensitivity can be improved with training. It is unfortunate that a significant percent of students will believe for the rest of their lives that they are tone deaf just because they have not had the opportunity to learn how to recognise pitch changes or to sing in tune.

To attempt to address this issue, I designed a sequence of aural training activities, with graded difficulty, to help students develop pitch identification skills. Previous research into computer based aural training found that students who started with poor aural skills fared significantly worse than higher skilled students (Ozeas, 1992), so the program was designed to be as gradual and sequential as possible. The easiest level began with a pitch matching program, whereby tones were presented to students who had to match the pitch of the tone presented (i.e. the test tone) by finding the tone of the same pitch on a keyboard (the target tone). At the initial level the same timbre (piano) was used for both test and target, but later levels introduced differences in this regard. The program incorporated constraint theory in the exercises, in order to increase focus and maintain motivation on this very repetitive pitch matching activity. If tones were correctly matched twice in succession, the duration of the test tone was shortened. Continuing success resulted in the test tones reducing to 20ms duration, at which point the program indicated to the student that s/he had ‘won’ and should move to a more challenging activity, such as interval recognition.

This program was trialled on a student in my class whose sensitivity to pitch change in a tuneful sense was obviously very poor. However, after preliminary testing I established that the student was not tone deaf, because she was able to differentiate (higher or lower) between tones closer than a quarter of a tone (Albouy et al., 2013). I therefore was confident that the reason the student could not identify tones was a training issue rather than a neuroanatomical problem. However, further testing established that the student could not match single pitches and the new program was therefore trialled with this student. Initially, this still proved too difficult for the student, and I therefore added a “hint” function to the program, so that the student would be told if the pitch was higher or lower. Following further practice, the hint function was withdrawn after three successful attempts. With some individual assistance, this strategy was successful, and the student was able to match tones unaided. This student’s subsequent progress matching pitches confidently was rapid, but she continued to struggle for a time with interval identification when returning to the ‘Auralia’ training program. Eventually this student did complete the Auralia exercises successfully. This student’s improvement was accepted as evidence for the efficacy of the new procedure and the strategies used to help that student were therefore incorporated into the development of the full interval training program used in the study now to be described.

Piloting the New Aural Training Program

In the next phase of development of the program, students were required to match the pitch of two tones by identifying them on an ‘html’ keyboard. Following this identification, students were then required to identify the interval which these two tones created. The program was designed to create random diatonic intervals up to 12 semitones apart. Therefore, there were seven possible intervals for students to identify. As before, two successful identifications led to decreased durations for test tones to follow. This procedure was trialled on a group of year 12 students (N = 5). The students reported that they enjoyed the challenge of trying to reach the shortest duration levels but that the interval identification exercise (matching the tones on a keyboard) was difficult and took a long time.

Based on this feedback, the task was then reduced to just recognising four musical intervals, with additional duration levels introduced to promote more gradual progress. This procedure was further trialled on a class of year 9 students (N = 16). However, some students continued to indicate that, although they found the task interesting, they needed more instructions to be able to use the program successfully. Further video instructions were therefore created and html ‘xylophone’ tones matching the pitches of the intervals were added to the starting or ‘base’ level of the program. After successfully trialling this modified program on a different group of year 9 students (N = 23), the program was finalized and made ready for the study by including a logon page and creating a storage area for test results.

The Training Study

The aim of this study was to test the new training method designed to facilitate the development of aural perception of students aged 12-14 years old, who were known to struggle to match pitch and identify musical intervals accurately. This study examined the effect of this new aural training program on the development of students’ aural capabilities, compared to the performances of a control group that continued with the current training method used at the school; i.e. they continued using the computer-based aural training program “Auralia” (Rising, 2016).

Background Development and Preliminary Pilot Work

The training method that evolved from several trials and developmental modifications incorporated many potential benefits over current aural training methods. The primary differences involved using shorter tone durations, incorporating a game-like interface and the addition of more steps in the learning sequence. Benefits of this system included starting with simpler exercises, using practice time more efficiently, keeping the difficulty level at a constant modest challenge, making the interaction and interface more game-like and using constraint training to focus attention. These benefits all stemmed from the use of very short tones in the training sessions.

The new training program started with a very simple exercise. In this first exercise the starting tone remained the same for all sequence presentations and the higher second tone was pitched either a perfect fourth or a perfect fifth higher than the first tone. These two intervals were repeated until the user chose to move on to a more difficult exercise. Repeated testing such as this has been found to improve memory retention of concepts and skills (Karpicke & Roediger, 2008; Lynch & Maclean, 2000). Repetition fatigue was minimised by constantly changing (increasing or decreasing) the duration of the two tones. Tone durations were increased or decreased based on a ‘two down one up’ staircase method, where durations were decreased after two successful identifications but increased after each identification error. This procedure provided the great majority of trials at durations close to current threshold and therefore kept the exercises at a level of difficulty appropriate to the student’s current skill level. As Brophy has emphasized, maintaining a constantly challenging difficulty level helps to ensure ideal learning conditions (Brophy, 1987).

A ‘game’ aspect was also incorporated within this program. The program was designed so that students could reach the achievable goal of identifying two intervals in a row at the 20 ms duration level. This training program presented nine difficulty levels that modified both the starting tone and the number of intervals being presented. The ‘game’ goal of each level was to reach the shortest duration: 20 ms. As the student progressed to shorter exposure durations, the level difficulty was visually coded and changed in colour from blue (default), through orange (started but not progressed below 100ms) and through light green (the participant has reached below 100 ms) to dark green (level achieved). There was also a scoring system, where the student could compare high scores, as has been recommended by research that had found that computer games improved learning outcomes by both improving motivation and improving rate of learning (Papastergiou, 2009).

Decreasing tone durations may have other benefits. Constraint training, where available information is restricted to focus cognitive powers on a specific area, has been found to improve learning outcomes. For example, Farrow and Abernethy (2002) used temporal occlusion when displaying examples of tennis players serving. This involved reducing the amount of information available to the participant by blanking a video screen after short durations of a video showing tennis players serving. They found this method significantly improved participants' skills for predicting direction of serve, compared to the performance of participants given explicit instruction or general training. Although this research involved reduced visual information rather than aural, it was assumed here that temporal occlusion training is transferable to aural activities, on grounds that there are many similarities between visual processing and auditory processing (Marks, 1987). Although shorter duration tones were presented in this study, instead of shorter duration videos, it was reasonable to assume that the use of temporal occlusion of audio information should lead to improved performance in interval recognition skills in high school students.

Another benefit of using short-tone duration training is that trainees would complete more training exercises per minute. In limited time conditions, such as during school lessons where a short amount of time is allocated to aural training, the presentation of more training trials could improve the efficiency and effectiveness of the training program.

Method

Participants

Participants were drawn from among 220 Year 8 students enrolled in the compulsory music program at Cabra Dominican College. This group included children who had varying degrees of musical training, but all would have had a minimum of two years of classroom music education as part of the prescribed educational program. Prior aural training had therefore been delivered, either when children were in earlier years at Cabra Dominican College or at the primary feeder schools located in the region from which some of these children had come. Prior to Year 8, the participants would have been taught music education using a variety of face-to-face classroom methods. The task described below formed part of the curriculum and the results from what is described below as the “post-test” contributed to the term’s assessment. However, ethic approval in the full study required informed consent, so that participants were recruited by way of an information letter addressed to the students and parents and an introductory demonstration by the researcher. Letters providing detailed information relating to the study were given to each student to take home and discuss with their parents. The students’ ages ranged from 12 to 14 years and there were 44 females and 40 males. Of the 220 year 8 students at the school, 98 students returned the signed forms, indicating willingness to participate. Of these, 84 students were accepted for the study. Three students were excluded because their pretest showed they had already established high level skills in the targeted training area (recognition of major second, major third, perfect fourth, and perfect fifth intervals). Nine students were excluded from the study because they had not completed both the pretest and the posttest.

Design

The study was designed to (i) test for improvement over pretest levels; and (ii) test for differences between two different computer-based procedures. The participants were given a pretest at the beginning of the school year. This pretest therefore established base-line aural skills already achieved as a consequence of previous exposure to aural training. Following the pre-test, participants completed 10 minutes per week of computer-based training, using either the “Aural Train” or “Auralia” program as described below, for six weeks. At the end of the training sessions, participants were given a post-test. The sessions all involved presenting intervals for students to identify. The intervals were: major second, major third, perfect fourth, and perfect fifth. A four-alternatives forced choice method was used for the pre-test and the post-test. The four intervals (described above) were selected because students were already familiar with them through their work on major scales.

Pre-test/ post-test design. For both pre-test and post-test, students completed a 40-question two-tone interval test. Half of the interval tests were presented using a sine wave timbre and half were presented using a piano timbre (see below for the reason for this). The post-test used the same intervals and timbres as the pre-test, but the stimuli were presented in reverse order to minimise any possibility of students replicating memorised sections of the test. The frequency range and tone selection and creation process were the same as that used in previous chapters and these have been described fully in Chapter 4. The pitch of the tones being used ranged from C4 to C6. Both the intervals and the starting tone were randomly selected using the computer program Excel using the "Randbetween" method (described in Chapter 4). The starting tone was randomly selected between C4 and F5 to ensure all intervals could be played within the C4 to C6 range. The reason for the smaller range for starting tone selection (C4 to F5 instead of C4 to C6) was that if a tone higher than F5 was selected as the starting tone, and (for example) the perfect fifth was selected as the interval, a tone above C6 would have been required to create the interval, which is higher than the highest tone created for the study. The intervals used in the tests were randomly selected from the four available intervals: major second, major third, perfect fourth, and perfect fifth intervals.

Training session design. The same intervals were used in the training sessions as in the pre-test and post-test. However, choices in both training programs were initially restricted to two intervals from a single starting tone and then gradually expanded, depending upon the student's success rate and personal selections. In the training sessions, choices ranged from a two-choice forced choice training exercise to a forced 4-choice exercise, depending upon the level of difficulty selected by the participant. Half of the participants used the program 'Auralia' (Rising, 2016) and half of the students used the program 'Aural Train' (the program created for this study). A problem that needed to be overcome in regard to consistency and equality of testing in the pre-test and post-test is that the program Auralia used a piano timbre to present stimuli in its training sessions, but the Aural Train program used sine waves. This was overcome by using both sine wave and piano timbres for the intervals in both pre-test and post-test (see above).

The Aural Train training exercises ranged from forced 2-choice exercises (top left - see Figure 6.2) to forced 4-choice exercises. This range is the same as Auralia, but the exercises increased in difficulty in two different ways. For Aural Train, the difficulty increased across the page by changing the number of different starting notes (see Figure 6.2). The left-hand choices started on the same note, the middle choice started on two different notes and the right-hand choices started on random tones within a 1.5 octave range. Down the page, difficulty increased by interval options changing from two intervals (easiest) through three intervals to four intervals (hardest). Students were asked to start at the easiest exercise (top left: "single start note with two intervals") and then they were left to choose the manner in which they progressed to the bottom right hand (hardest) exercise. Students selected the degree of difficulty by clicking on a button with a description of the level. The exercises on the bottom right were the same level of difficulty as those that were presented in the pre-test and post-test.

In "Auralia" students could control the number of intervals being presented but not the pitch of the starting tone, which was semi-randomly generated starting with a small pitch range and gradually increasing in range as more intervals were presented.

Materials

Stimuli. Sine wave stimuli were created using the same frequency range and in the same way as in Chapter 4. Sine wave tones were joined with other sine wave tones at zero crossing to create the four musical intervals. All stimuli were saved as individual wave files. This was accomplished using the digital audio workstation program Sonor (2009). Stimuli were created at sixteen duration levels. The levels created for the practice sessions were: 200ms, 100ms, 70ms, 60ms, 50ms, 45ms, 40ms 35ms, 30ms, 25ms ,20ms.

For the pre-test and post-test, piano tones were created in the same way as described in Chapter 4. These piano tones were only used for half of the musical intervals in the pre-test and post-test. As explained above, this is because the Auralia program uses piano tones and Aural Train uses sine wave tones to present their intervals. Therefore, to provide equality of testing conditions, half of the intervals were presented using sine wave tones and the other half were presented using piano tones. All intervals were presented at a slow speed, comparable to the default duration setting in Auralia. This equated to around 500ms per tone. The intervals were randomly selected (from among second, third, fourth, and fifth intervals) and the starting tone was also randomly selected.

Pre-test/ Post-test Apparatus. Stimuli for pre-test and post-test were presented to participants via the internet site https://intervalstore.a2hosted.com/ which was created for this purpose (See appendix 3 for the source code of this program). Participants listened to each stimulus through their own computer and responded by clicking one of four buttons representing the four alternative choices (second, third, fourth, and fifth intervals) (see Figure 6.1 below).

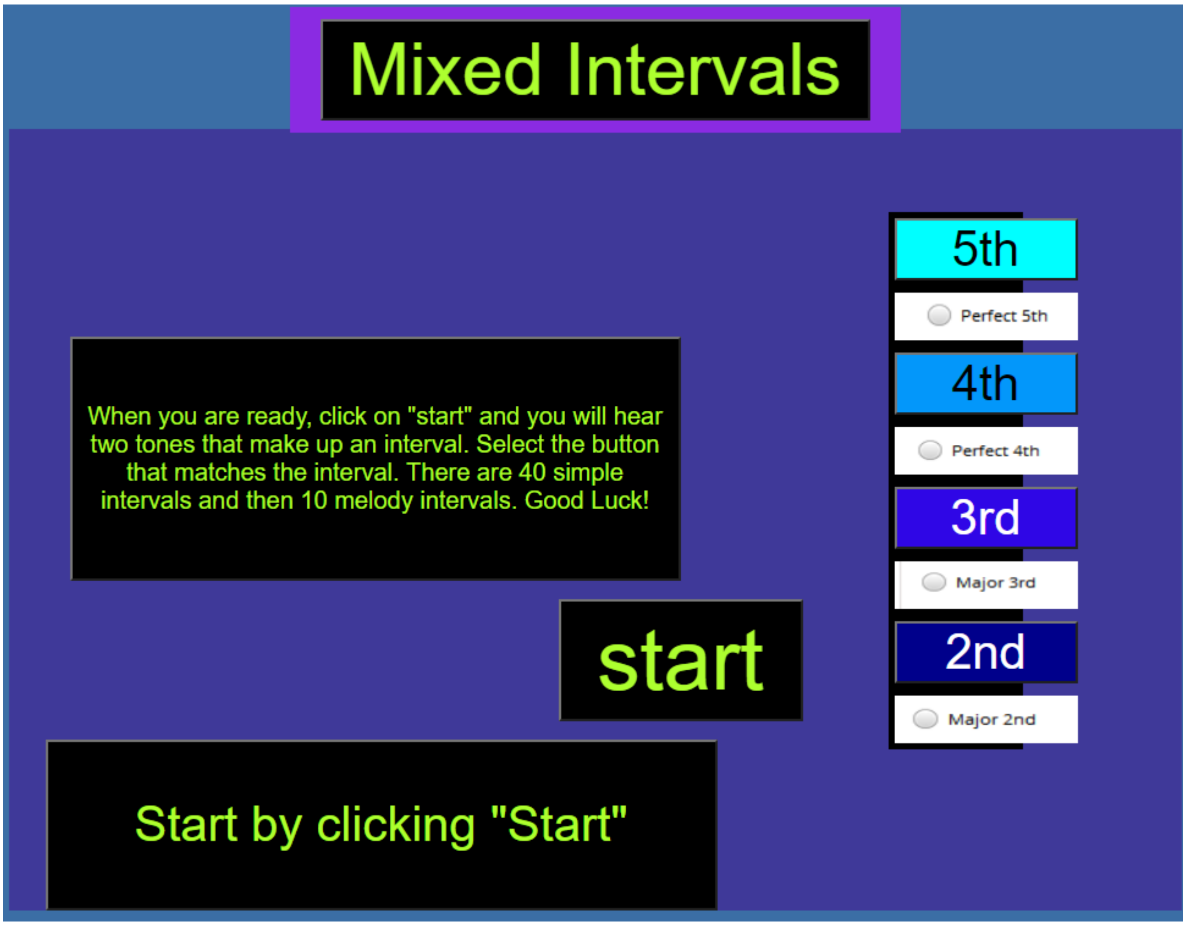
The study participant interface was created using HTML (WHATWG, 2017) with data recording using SQL (Chamberlin & Boyce, 2016). The data were initially stored on a server controlled by A3Hosting and then downloaded at the end of the study. PHP (PHP, 2018) was used to communicate with the server and transmit study information between the user and the database.

Training Session Apparatus. Stimuli for the training sessions were presented to participants via two different internet sites. Aural Train users accessed the internet site https://intervalstore.a2hosted.com/ which was created for this purpose. Auralia users accessed the site: https://www.risingsoftware.com/auralia, which is the site students currently use at the school for access to aural training.

For the Aural Train program, graduated exercises were provided in a progress grid (see Figure 6.2 below). The grid progresses from the simplest exercise (two intervals from a single base tone) at the top left corner of the grid to the hardest exercise (four intervals from random base tones) on the bottom right.

Figure 6. 1

Pre-test/post-test board



Note: After clicking the "start" button, the stimulus was presented. Students clicked one of the four blue buttons, or four white buttons, according to the interval they believe was presented. To minimise any advantage from using buttons that matched one program and not the other, the four blue buttons matched the buttons in Aural Train and the four white buttons matched the buttons in Auralia. After selecting an interval, students were given feedback for accuracy. The feedback was presented as changes in colour of the buttons. The correct interval changed colour to green and if the student's selection was different to the correct interval, their selection changed to red. Students again clicked the start button and all buttons were reset to their blue colour and a new stimulus was presented.

Figure 6.2

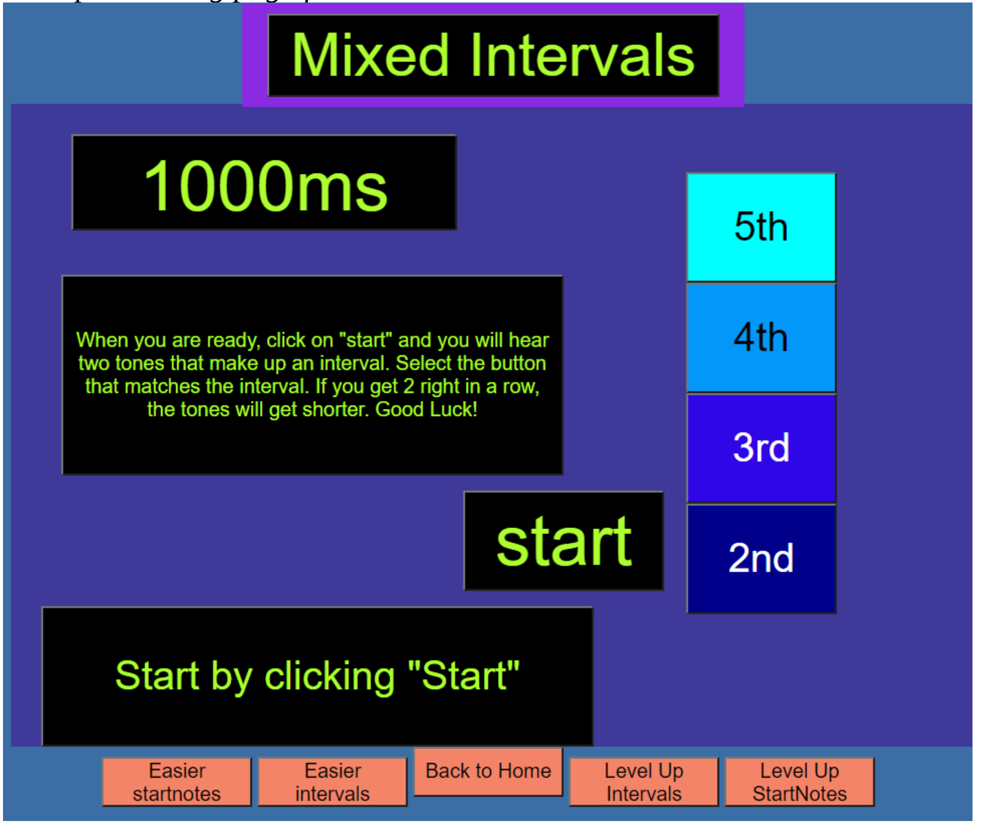
Starting screen for the Aural Train program

Note: The nine levels increase in difficulty both downward and left to right. Going downward, students progress from two intervals to four intervals. Across the columns, students progress from a single start note to random or “mixed” start notes. Progress at each level is shown by colour. Blue means ‘not started’, orange means ‘started but duration achieved is greater than 100ms’, light green means ‘Duration achieved is between 30ms and 70ms’ and dark green means ‘Level Completed’.

An example of the webpages used for training in the Aural Train program is shown in Figure 6.3 below. This figure shows the webpage for the hardest level: four types of intervals with random start notes. A description of the procedure for progressing through the stimuli is presented in the “notes” section of Figure 6.3.

Figure 6.3

Example training page for Aural Train

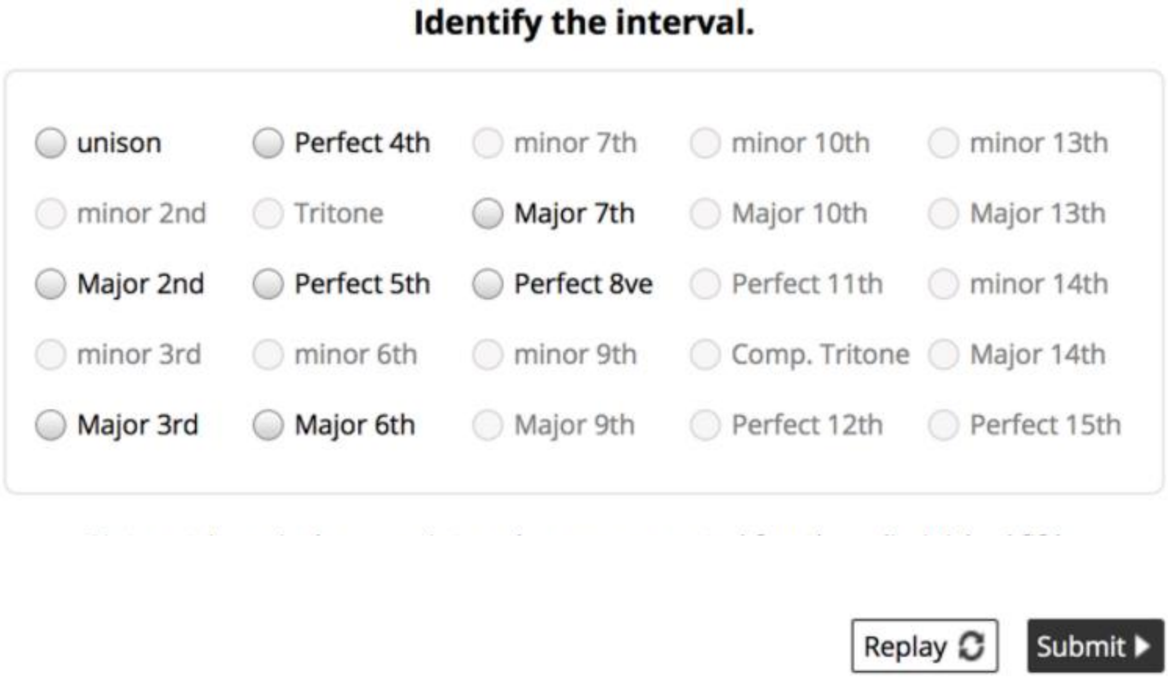


Note: On the webpage used by the students, only the intervals students were learning were visible and able to be clicked. Level 1 started with just the perfect fourth and perfect fifth, Level 2 added the major second and Level 3 (the current screen) added the major third. The orange buttons at the bottom of the screen were used to move between levels and to exit the program.

An example of the webpages used for training for “Auralia” is shown in Figure 6.4 below. A description of the procedure for progressing through the stimuli is presented in the “notes” section of Figure 6.4.

Figure 6.4

Example training page for Auralia.



Note: On the webpage used by the students, only two to four intervals were in bold and able to be clicked. Level 1 started with just the perfect fourth and perfect fifth, Level 2 added the major second and Level 3 added the major third.

Procedure

In this study, participants were given a pre-test, then completed 10 minutes of aural training per week for six weeks using one of two different aural training programs. They were then given a post-test. The procedure for the pre-test and post-test was as follows. The participant was seated at a table with their computer and headphones. The sound was adjusted to a comfortable loudness (approximately 65 – 75 dB). The participant was presented with a set of instructions indicating the procedure to follow and which buttons to press. After reading the instructions, the participant clicked on the "start" button. The first stimulus was then presented. After each stimulus had been presented, the participant clicked the button corresponding to the interval each participant identified. After pressing start, the start button changed its name and function to a "repeat" button. Participants could press this button up to three times to hear a repeat of the interval. This procedure was repeated until all 40 stimuli had been presented. There was no feedback given during the test, but the overall results were given to the participant at the end of the test.

Results

Figure 6.5 sets out histograms, divided into students who trained with Auralia and students who trained with Aural Train, showing the number of participants (out of 84 participants) who achieved the scores shown out of 40 for the pre-test and post-test. Pre-test and post-test means and SDs for Auralia were 13 (3.1) and 14 (4.1), respectively; and 13 (3) and 15 (4.7), respectively, for Aural Train.

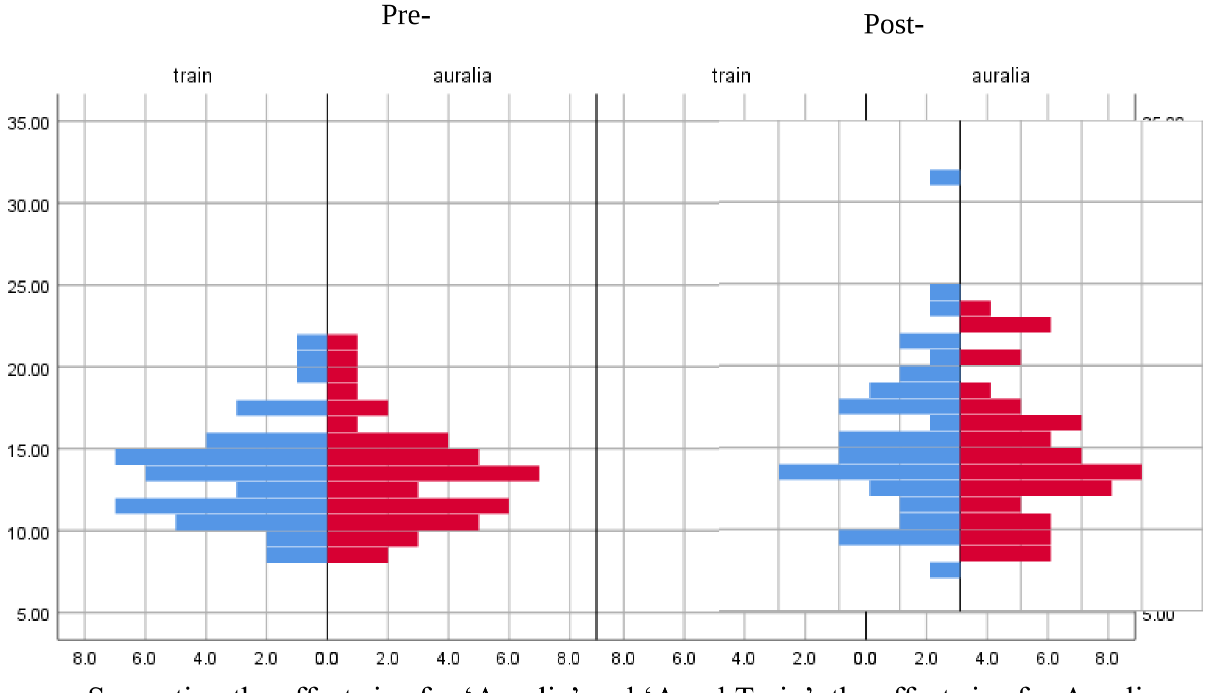
These results were analysed using a two-way mixed ANOVA method. There were no outliers, as assessed by examination of studentized residuals for values larger than ±3. Scores were normally distributed, as assessed by Normal Q-Q Plot. There was homogeneity of variances, as assessed by Levene's test of homogeneity of variance ($p > .05$). There was homogeneity of covariances, as assessed by Box's test of equality of covariance matrices ($p = .78$). There was no statistically significant interaction between the training programs and time on interval recognition ($F(1, 82) = 1.17$, $p = .283$, partial $\eta 2 = .110$). The main effect of time showed a statistically significant difference in interval recognition from the start of the trial to the end, $F(1, 82) = 10.19$, $p = .002$, partial $\eta 2 = .11$. There was no statistically significant difference between the two groups ($F(1, 82) = .603$, $p = .439$, partial $\eta 2 = .007$).

It is worth noting that the top score for the 'Aural Train' post-training results (Figure 6.5: blue mark at the '32' point level) is not statistically an outlier. Further analysis was completed to examine the effect size of the training programs both grouped together and separately.

The effect size of improvement with both computer-based training programs combined was calculated using Hattie's recommended method for measuring effect sizes (i.e. the difference between the pre-test and post-test group means, divided by the pooled standard deviation) (Hattie, Biggs, & Purdie, 1996). This effect size was 0.40. This can be considered to be a successful outcome because it is higher than the average effect size of 0.34 for the introduction of drill and practice programs against standard teaching methods (Hattie, 2008, p. 224) and equal to Hattie's recommended effect size for a substantive improvement in student outcomes.

Figure 6.5

Histogram of Pre-training and Post-training scores for 'Aural Train'(train) and 'Auralia'.



Separating the effect size for 'Auralia' and 'Aural Train', the effect size for Auralia was 0.28 whereas the effect size for the 'Aural Train' program was 0.50.

Discussion

Two aural training computer programs were compared for their effectiveness in improving interval identification skill in year 8 students. Both programs achieved statistically significant improvement from pre-test to post-test but there was no significant difference between the two programs. The difference in effect size between the two programs was moderately large (Auralia effect size: 0.28, Aural Train effect size: 0.50) but it would require a sample size of 300 for this difference to be statistically significant at $p < .05$. This number of participants is beyond the candidate pool available at the current school but could be achieved if the study was run over successive years or a larger cohort of students was used (perhaps across multiple schools). This will be discussed further in the concluding chapter.

It is also the case that a majority of participants completing these training programs showed no improvements at all, which is obviously a major concern for the effectiveness of both programs. Nonetheless, the considerable work involved in developing the ‘Aural Train’ program has generated ideas about how to increase its effectiveness and these will be discussed in the ‘recommendations for further research’ section below and expanded further in the conclusions chapter.

Implications

Although encouraging, the computer-based ‘Aural Train’ program that was created for this study was not demonstrated to be significantly better than current aural training methods. Therefore, it would be inappropriate to recommend that this program be used as a replacement method in an education setting. However, further improvements on the program might be possible.

Limitations

This study used a school environment where there were many uncontrolled variables. Student motivation levels varied and likely changed on a weekly basis, which meant that students' commitment to identifying intervals accurately might have changed because of these different personal conditions. The length of time spent on the training program might have varied for different lessons and for different students. Some students settle and start work more quickly than others, which would give them more time for training. There were technical difficulties with both the internet and some students' computing equipment, which limited their ability to access the training program at times. There were also students who missed some of the training sessions and tests for reasons including sport, other school-based activities and through illness. This would certainly have had a limiting effect on the effectiveness of both programs by reducing time 'on-task' during training. Available time was also very limited because of class scheduling. Thus, only a limited amount of time could be allocated for aural training and school constraints meant that training times could not be extended. However, especially given time limitations, it remains plausible that more time could deliver better progress; six 10-minute sessions, each separated by a week, obviously does not constitute an optimum for training of this kind and further work is required to provide a thorough consideration of training effectiveness.

It would have been desirable to include a no-training control group, to ensure that the improvement was not simply a function of taking the test for a second time. However, the curriculum at the school required that students be provided with aural training and the study was permitted, consistent with this. Within the limited timeframe available (one semester) it was not possible to provide a 'no-training' control group. With a longer timeframe, it would have been possible to delay one group's training to later in the year and limit initial involvement to a pre-test and post-test, coincident with those completed by the other students participating in the intervention conditions.

Conclusions

The program using a new method for interval training designed for this study did not deliver a statistically significantly better outcome than 'Auralia'. Therefore, using current methods, the prediction that placing higher demands on the nervous system that handles pitch perception than is required under normal listening conditions did not result in greater benefit for the neural encoding of pitch. However, based on current results, with a study cohort of 300, the difference between the two programs would likely be statistically significant, in favour of the new methods described here. Moreover, it is likely that longer training times are necessary if either of these training programs is to produce improvements to the performance of a majority of trainees. It is therefore cautiously suggested that the comparatively high effect size of 0.5 for the program designed for this study may well indicate that the use of short tones for aural training does show sufficient promise as a method for improving student outcomes to warrant further consideration. Possible ways for improving the training program are outlined in the section to follow.

Recommendations for further study

An improved method of aural training might be achieved by (i) creating targeted videos for explaining tasks; (ii) improving the 'game like' aspects of the training program; (iii) providing opportunities for person-to-person interaction; and (iv) designing a larger study using the world wide web. These four areas for improvement will be discussed in detail in the Conclusions section of this thesis. In addition, the effectiveness of the study might be improved by using elective music students rather than general music students and by increasing the number of participants. These areas for improvement will also be discussed in detail in the Conclusions section of this thesis.

Chapter 7 Conclusions

The aim of this thesis was to explore pitch perception of very short sounds, in order to understand this area more thoroughly and to use this knowledge to improve music education. The first objective was to test the hypothesis that the minimum duration for ordering tone sequences is shorter than the minimum duration for perception of pitch. Given support for this hypothesis, the second objective was to establish the minimum duration for perception of pitch (MDPP) in 2-tone and 3-tone sequences, and to explore the effect of different tone positions on the MDPP. The third objective was to use the information gained from these investigations to improve computer-based aural training methods, by creating an improved aural training computer-based program.

A series of research questions were created (see p. 8) and studies were designed and implemented to answer these questions. A summary of these studies follows.

Adaptive Methods for Frequency Change Identification

Prior to the research reported in this thesis, the minimum duration for perception of pitch (MDPP) in short contiguous sequences of tones had not been established, although the order of sine wave tones had been found to be identifiable at very short durations (less than 4ms per segment). The first study (see Chapter 3) demonstrated that when sound analogue comparison was prevented by varying timbre between stimuli (while retaining the same two frequencies for all stimuli), the minimum duration for identifying change in direction (up or down) was significantly longer, compared to when the same timbre was used. It was concluded that, in the past, sound analogue comparison had been used to differentiate between frequency change stimuli at the 1ms per tone segment level. In short, it is not necessary to be able to perceive the frequency of tone segments, or even the direction of frequency change between tones, to distinguish between different orders of tones.

The Minimum Duration for Perception of Pitch in Short Contiguous Tone Sequences

A method that ensured the use of pitch identification when identifying the MDPP in short contiguous tone sequences was created. This method was guided by conclusions from the previous study, together with information from the literature review about information masking (see chapter 2). Participants matched a single tone segment from a sequence of tones against a single tone with different timbre and duration.

Following necessary pilot work, the MDPP for single tone, 2-tone, and 3-tone sequences were measured, using four experienced musicians as participants. Results showed that the mean MDPP for single isolated tones was 8 wave cycles, essentially the same as the 8 wave cycles established by Patterson et al. (1983) as the minimum duration for melodic pitch. The mean MDPP for 3-tone contiguous sequences was 48ms, ranging from 18ms to 77ms depending upon the position and relative pitch-height of the tone in the sequence. For tones in the first or second position (of the three positions), the lowest tones in each sequence required longer durations for identification than the highest tones. Mean MDPP in 2-tone contiguous sequences was 24ms, ranging from 17ms to 35ms depending upon the position and relative pitch-height of the tone in the sequence. Again, longer durations were required to identify low tones in the first position compared to high tones.

This study confirmed that the effects of masking sine wave tones by other sine wave tones is not consistent across frequency levels. High tones (relative to other tones in a sequence) created a stronger backward masking effect than low tones. In the forward masking condition (i.e. when identifying the last tone in a sequence), minimum durations for perception of pitch were relatively consistent across frequencies but still significantly longer than for identifying single tones. Reasons for these differences will be discussed in the "Responses to Research Questions" section below.

Minimum durations for perception of pitch in this research were considerably longer than for studies that have reported minimum durations for ordering similar short sequences of tones. This study therefore confirmed previous results (see Study 1), that longer durations are required for perception of pitch of tones in short sequences than is required to identify tone order.

It was concluded from the MDPP for isolated tones, 2-tone and 3-tone sequences and looped melodies, that minimum durations for perception of pitch increase in a stepwise fashion, from a single tone to a continuous sequence. From current information it appears that the average MDPP increases at a diminishing rate, from tripling minimum durations between single tones and 2-tone sequences to doubling minimum durations from 2-tone sequences to 3-tone sequences. The point where the MDPP for short contiguous sequences would reach the MDPP for continuously looped stimuli is therefore difficult to estimate and this question would therefore benefit from further research.

Aural Training Study

The results of the MDPP study were incorporated into an aural training computer program. This relied on temporal occlusion training to enhance educational outcomes of year 8 secondary school students by improving their interval recognition skills. The new training program was compared to a commercially available aural training program. However, although promising in some regards, the new method did not produce a statistically significantly better outcome.

Responses to Research Questions

Information from the above studies has been collated and used to create responses to the research questions proposed at the beginning of the thesis:

1. Does the ability to order a sequence of tones mean that the pitch of the tones can be identified?

The ability to order tone sequences does not mean that pitches of tones in the sequence are identifiable. The minimum duration for identification of any of the pitches in a 3-tone sequence is 18ms (for the first tone when it is the highest relative pitch), whereas 3-tone sequences can be ordered for segment durations less than 10 ms. Therefore, there is no direct relationship between the ability to order tones and the ability to identify the pitch of each tone,

2. What is the shortest duration at which the pitch of a tone is identifiable when presented in a 2-tone or 3-tone sequence?

The mean overall minimum duration for perception of pitch (MDPP) in this study was 24ms for tones within a 2-tone sequence and 47ms for 3-tone sequences. However, because the effect of differences in masking levels depends on the relative pitch and position of each tone, a more accurate description is that minimum durations ranged from 17ms (B5eq) to 35ms for tones within 2-tone sequences and from 18ms (B5eq) to 78ms for 3-tone sequences. The relative pitch of each tone, and its position within a sequence, affects minimum durations in predictable ways. In both cases, lower tones located early in the sequence required the longest durations for identification and higher tones located early in the sequence were identified at the shortest durations.

3. Is there a logical progression in the MDPP from a single tone through to continuous tone sequences?

There is insufficient research currently to identify a logical progression in the MDPP. The sequence of MDPP in contiguous sequences, from single tones to cyclically repeated tones, is unidirectional but irregular. Average MDPP tripled from single tone identification to identification of tones in 2-tone sequences and then doubled from 2-tone sequences to 3-tone sequences. It may be that, at some point, asymptote will be reached at around 160ms per tone segment, but because the progression from 1-to-2-3 has not been found to be linear or exponential, there is insufficient information to permit prediction of minimum durations for 4-tone and 5-tone sequences.

4. Does the presence of other tones affect (or mask) the perception of short tones equally, regardless of the relative position of the masking tone or tones? Specifically:
    a. Does the position (first second or third) of the target tone affect the minimum duration required for perception of pitch in 3-tone sequences?

There was no significant difference found that affects the average MDPP depending on the position of the target tone. However, this may be a type II error, resulting from small participant numbers.

b. Does the relative pitch (low middle or high) of the target tone affect the minimum duration required for perception of pitch in 3-tone sequences?

In both 2-tone and 3-tone studies, low tones early in the sequence required longer durations for identification than high tones located early. There was no significant difference between high and low tones for the last tone in the sequence. Although not statistically significant, there was a tendency for higher pitch tones to require longer durations in the forward masking condition than in the backward masking condition.

5. If so, what mechanisms cause the differences in the minimum duration for perception of pitch in sequences of contiguous tones?

It has been suggested that backward masking of low tones by high tones is primarily caused by preconscious adjustment of low tones to accommodate subsequent high tones as harmonics of the lower tones, although this suggestion is speculative. Two other possibilities were suggested as potentially accounting for differences in minimum duration for identification of high and low tones. The first was that stream segregation of higher tones may allow the high tones to be isolated and processed separately. The second was misidentification of tones based on a preconscious assumption that the tone sequences would follow common patterns or comprise common musical intervals. Clearly, the plausibility of these suggestions requires further research.

The minimum duration required to identify the last tone in a sequence did not appear to be affected by the pitches of the preceding tones. Two reasons are proposed to account for this. The first is that the gap after the last tone allowed for extra processing of the last tone, thus enabling more accurate results for all relative pitches. Secondly, the issue of preconscious adjustment of low tones to accommodate subsequent high tones as harmonics does not arise for the last tone because there are no subsequent tones to cause this effect. Because the last tone was only affected by forward masking, it was concluded that forward masking levels are not affected by the frequency of the target tone; compared to the other tones in each stimulus.

6. Can the information gained from this research be used to assist the education of secondary music students?

The new interval training method was about as effective as the commercially available “Auralia” computer-based program. This was not the outcome expected. Nonetheless, subsequent consideration of the many problems encountered with completing this study has suggested that the new program may hold promise if those problems can be overcome. This matter is discussed below.

Possible benefits from research undertaken in this thesis

Study 1: Adaptive Methods for Frequency Change Detection

Understanding how identification strategies change as segment durations shorten contributes to understanding individual cognitive limitations. If some participants are unable to change from pitch detection to frequency change detection and then to sound analogue comparison as durations shorten, this may suggest cognitive limitation. Of course, care would need to be taken to ensure that sufficient training is given before testing, to give participants every opportunity to learn the skills required for each level of frequency change detection, if they are cognitively able to do so. After further validation, the test program used in this study could provide a testing procedure for identifying this form of cognitive limitation,

Study 2: The MDPP in Short Contiguous Tone Sequences

Understanding differences in minimum durations for perception of pitch could lead to further improvements in hearing implants, computerised speech recognition, speech synthesis, sound compression, auditory scene analysis and computer-generated music. Aural training methods might also be improved by application of improved understanding of the strengths and limitations of pitch perception. Improved knowledge of short-tone pitch perception could also lead to improved understanding of how the brain processes sound and thereby improve knowledge of brain functioning. Such knowledge could also expand our knowledge of masking. Each of these possibilities is discussed, below.

Improvements in Hearing Aids. Although improvements in sound compression for hearing aids and other purposes currently focus on *energetic* masking, *information* masking, where aspects of a sound are masked such as the sound’s pitch, may in the future contribute toward sound compression improvements.

Hearing implant technology is improving rapidly in the areas of volume compression techniques, frequency band selection and real time processing (Hajiaghababa et al., 2018). Future challenges include the elimination of reverb (in the processed signal) and the successful processing of compressed signals, such as telephone speech (Allen, 2008). Once these issues have been resolved through further research and increases in computer processing power, information masking and the importance of understanding how well low and high tones are processed at different points in a phrase may become important. Alternatively, understanding the effects of information masking may improve hearing outcomes more effectively than some of the issues mentioned above. For example, knowing that high sounds at the start of a phrase are more likely to be clearly heard than other tones may help in the creation and refining of algorithms used in sound analysis and compression techniques.

As implants improve through the addition of more electrodes and more sophisticated nerve stimulation methods, minimum duration for short tone data could inform manufacturers of the limits of sound perception beyond which it is pointless to proceed. For example, providing accurate pitch information for low tones at short exposure durations (e.g. less than 70ms) at the start of a sequences is pointless because this information is masked by the following tones. However, providing accurate pitch information for low tones with short exposure durations on the final tone could be very important, because accurate pitch identification is possible for exposure durations as short as 40ms, and some important information could be lost if this sound was not clearly heard by the wearer of a cochlear implant.

Improved training for cochlear implant patients. Both Galvin III, Fu, and Nogaki (2007) and Lo, McMahon, Looi, and Thompson (2015) found that gradually increasing task demands towards threshold levels using a similar design to this aural training program was an effective way of improving aural perception, as well as improving speech perception in cochlear implant patients. Further, Lo et al. (2015) found that melodic contour training, where melodies were increased or decreased in tone duration depending upon successful identification, conferred benefits for training cochlear implant recipients to better perceive speech. In their study, Lo et al decreased segment durations down to 50ms for all tones. However, based on results from this thesis, it may have been more effective if durations were asynchronously modified based on the position of the tone in the sequences, the tone's relative pitch, and the number of tones in the sequence. This would ensure that there were no sequences that were outside the perceptual range of the listener.

Improvements in Computerised Speech Recognition. Speech recognition requires the analysis of a single (or stereo) sound wave trace for key factors that indicate that certain vowels and consonants have been spoken. Knowing what aspects of this sound trace are important for speech recognition and what aspects are inconsequential is an important part of this process. For example, the base frequency (or fundamental frequency of the voice tone) that is used to speak is irrelevant, because this can vary enormously amongst speakers (Ladefoged, 2018). However, the direction of frequency change and the dominant frequency of the harmonics in the 'speech range' of 200 Hz – 4000 Hz are critical for understanding speech. Understanding which aspects of speech are perceivable (and therefore possibly important), compared to which aspects are not perceivable (and therefore irrelevant), can improve the efficiency of speech recognition systems. For example, changes in higher frequency tones may require segmented analysis as short as 25ms, whereas changes in low frequency tones can be broken into far longer segments for analysis.

Improvements in Speech Synthesis Techniques. In the same way that speech recognition requires the analysis of key factors of a speech trace, speech synthesis requires a comprehensive understanding of the same key factors to create a realistic speech trace. Therefore, understanding which aspects of speech are perceivable (and therefore possibly important), and which aspects are not perceivable (and therefore irrelevant), can improve the efficiency of speech synthesis systems. Furthermore, as the provision of emotional content in synthesised speech becomes more important (for example, providing an 'urgent voice' for time critical information), pitch information will become more important and details relating to pitch perception and comprehension will become more valuable.

Improvements in Sound Compression Techniques. As mentioned above, although improvements in sound compression currently focus on *energetic* masking, *information* masking (such as the minimum duration differences in tone identification reported in this thesis) may in the future contribute toward sound compression improvements and the research reported in this thesis may assist in this process. Sound compression utilizes what is currently known about masking to simplify sound waves without causing audible loss in sound quality. An improved understanding of the masking process should lead to higher quality and more efficient algorithms for improving sound quality in compressed files, such as MP3 files. Some of the short-tone research outlined in this thesis has relevance to information masking effects, especially in relation to perception changes in different frequency ranges.

Improvements in Auditory Scene Analysis. Auditory scene analysis is the process of creating an imaginary 'scene' based on the sounds being received by a person (Bregman, 1994). For example, a low pitch rumbling might be a chair being scraped or the start of an earthquake. Various characteristics of the rumble will allow the listener to refine their 'scene' until it is as close to reality as possible. Understanding the limitations of hearing, such as minimum durations for perception of pitch in different conditions, has potential to assist us in understanding which aspects of sounds are being used to discriminate between the various possible physical causes of audible sounds. For example, the relative clarity of the highest tone in a sequence may serve to clarify the auditory scene and thus be important information to include in a scene analysis. Alternatively, having the high tones ‘pop out’ of a sequence may lead to greater confusion and might require minimising when analysing a scene. This area might be worthwhile examining in a study designed to test what frequency ranges provide the most relevant information for auditory scene analysis. Additionally, as auditory scene analysis moves into the field of artificial intelligence, high levels of understanding as to how human sound perception works will become increasingly important.

Improvements in Computer-Generated Music. There are virtually no limits to the sounds computers can create. However, understanding the limitations to hearing will assist programmers and composers to create music that is, for example, intentionally blurred by using tones shorter than 50ms, clear but confusing, by using tones between 50ms and 160ms, or creating trackable (and therefore memorable) phrases, using tones longer than 160 ms. Additionally, by understanding how perception changes depending on the position of a tone in a sequence, melodies can be created that are just on the edge between clarity and blurring, which may be a desired effect for some composers.

Improvements in Game Creation. In combination with auditory scene analysis, the realism of games can be improved by our understanding of perception. Understanding the limitations of hearing, such as minimum durations for perception of pitch in different conditions, will assist in understanding which aspects of sounds are being used to discriminate between the various possible physical causes of audible sounds. Consequently, this understanding can be used to create the impression of (for example) an earthquake that sounds realistic. Realistic auditory scenes created with such a process will enhance the attractiveness of a game.

Improved Understanding of How the Brain Processes Sound. To understand better how the brain processes sound, the limitations of perception need to be determined. In this thesis, to attempt to explain why these limitations occur, consideration has been given to possible links between brain waves and limitations in sound perception. Theories have been presented linking theta waves, echoic memory, and limits in minimum durations for perception of cyclic tone sequences. Theories have also been presented in relation to extra processing time being available for the analysis of short, isolated sequences of tones during theta wave cycles. Although the relationships between tone sequences and theta waves have not been studied in this thesis, methods for testing the relationship of cyclic tones and short sequences of tones with theta waves have been proposed (see below).

Computer-based Aural Training

It was envisaged that knowledge of the perceptual limits of pitch identification gained from these studies would improve aural training methods for high school students. The main strategy to achieve this was by using temporal occlusion training. Temporal occlusion training involves reducing minimum durations of tone segments by incremental steps to improve focus and concentration. In addition to the benefits of temporal occlusion training, the use of durational limits allowed a more game-like environment and had the added potential benefits of increasing presentation rates for training events and closely matching an individual student's 'modest challenge' level by constantly adjusting difficulty level, based on identification success. Although not found to be statistically significantly better than current methods at this stage, the new program reported in Chapter 6 showed promise and seems worthy of a retrial, after identified aspects of the program have been improved, as set out in some detail in the section headed “An improved method for aural training”.

Reflections on the Process and Product of this Thesis

Objectives and Research Questions

As described in the preface, I had previously anticipated that a very short segment of sound, perhaps 1/50 of a second in duration, would be cognitively indivisible and that all elements of sound - pitch, loudness, duration, timbre, sonic texture, and spatial location would be perceptible within this unit. However, research outlined in the Introduction clarified that this was not so. Nonetheless, minimum durations for identification of pitch within specific boundaries or criteria could be established.

The objectives of this thesis were therefore to establish the minimum duration required for accurate perception of pitch in 2-tone and 3-tone sequences and to use this information to create an improved aural training system. The first objective was achieved. but the aural training procedure (as designed and tested in this thesis) was not successful.

Literature Review

The literature review focused on masking of the pitch of short tones, the different bands of limitations to short tone ordering resulting from the use of different study methods (see below), and the cognitive limitations that most likely cause the limits in perception that were found in this and in earlier short tone perception research.

As a consequence of a thorough study of the area of sound perception, and specifically my research into the cochlear and sound processing that occurs during the transition from the cochlear to other areas of the brain, my understanding of how sound perception works was changed. Accordingly, my teaching in this area also changed substantially. I came to understand that there are two methods of gathering pitch information, which are suited to different frequency ranges. The well understood and commonly taught method is the ‘place’ method. The other, method, the ‘rate’ method of pitch perception, is not well understood or used in the field of education or in information found on the internet. An internet search, using the search terms “how does the ear work?” and “process of hearing”, found that the ‘rate’ method of gathering pitch information was not mentioned in most of the websites found, even though some of them specifically mentioned the ‘place’ method of gathering pitch information (HearingLink, 2018; UHAC, 2019). Moreover, even when the rate method was mentioned specifically, such as on the Encyclopedia Britannica online website (Hawkins, 2018), it was given only two sentences out of three paragraphs dedicated to the cochlear. This is despite the rate method being used exclusively for hearing low frequency tones (below 150 Hz) and being an important method of pitch analysis for the whole vocal and instrumental pitch range. However, from a personal perspective, with the new information gained from studying the physiology of hearing, I can now provide to the students whom I teach, a far more comprehensive picture of how sound perception works.

My research into sound perception was used to inform my rationale for concluding that perception of pitch was not possible at the durations being successfully identified in some of the studies that were examined (those with results below 4ms per tone segment) and led to my exploration of different methods for identifying the stimuli that might have been used to identify stimuli in these studies.

While researching the different theories of sound perception, I was disturbed by the way some authors strongly associated themselves with a particular theory and fought against evidence to the contrary to justify their belief. While acknowledging the value of the law of parsimony as a guide to research method, it seems likely to me that, in this case, rather than limiting sound processing to one method or another, the brain would use all methods available to gather information that might assist in making a judgement about pitch. An analogy I have developed to explain my misgiving about this conclusion is that of a person who can walk (an analogy for the rate theory) or ride a bicycle (an analogy for the place theory) to a destination (a pitch decision). If a person had access to both means of transport, but one of them is more suited to a particular environment – for example, crossing a muddy field (an analogy for discerning the pitch of a low-frequency sound), surely the person would not limit her/himself to walking (i.e. using the rate method of sound analysis), if s/he came across a firm patch of ground (an analogy for higher frequency content). Surely the person would ride the bike, if doing so made the destination easier to reach. In the same way, for a low-pitched sound (which supposedly uses 'rate' analysis exclusively), that also incorporated high frequency components, surely the brain would not ignore this aspect of the sound and only use rate analysis. It seems far more likely that both rate and place analysis would be used to gain a more comprehensive picture and help to provide a more accurate and comprehensive decision in relation to pitch.

Minimum Perception Bands in Short Tone Research: The Grouping of Perception Categories. As my early investigations of short tone ordering research progressed, it became apparent that results from the research into tone sequence ordering could be grouped into four categories or duration bands, based on the type of independent variables being studied and the identification criteria provided to the participants. The bands described in Chapter 2 were therefore developed, based on the minimum duration for identification of stimuli for each study. Studies with similar research objectives and types of stimuli fell clearly into the identified bands. It is noteworthy that there were no exceptions to these groupings, which suggests that these bands are real and based on cognitive limitations.

Of those detailed in Chapter 2, the 15ms – 75ms band was the most diverse, with a wide variety of cognitive effects found in this duration band. This is also the duration band within which the current research fell. Further research enabled me to outline reasons for the limitations outlined above, which have been detailed below.

Tone Sequence Identification Studies. There were three areas of research that required participants to identify the pitch of sequences of tones. They were short cyclic melody identification, masking studies and gapped tone research.

Short cyclic melody identification was found to be possible at durations of 160ms per tone and was compared to the MDPP for 3-tone sequences. The rationale for this decision seems justified because (as pointed out in Chapter 5), although pitch perception was not mandated in Warren et al.'s (1991) cyclic melody study, it seemed highly likely that pitch perception was being used in the study and so was a valid comparison for the 3-tone study.

Masking studies and gapped tone studies provided information as to what differences in minimum durations to expect in the 2-tone and 3-tone studies. Watson et al. (1975) and Kallman and Massaro (1979) found that lower tones, relative to the other tones in sequences, required longer durations for identification than higher tones. Additionally, some researchers (e.g. Massaro 1970 and Ronken1072) found that 'masking tones' placed before test tones enabled the test tones to be more easily identified. From these results it seemed important that the position of each tone in a sequence was isolated and examined separately. Separating tone identification in this way meant that the research results in Chapter 5 could confirm that lower tones were harder to identify when followed by higher tones, but tones placed before a target tone did not assist in identification of the target tone.

Echoic Memory, the Theta Wave, and the Perception of Sounds in a Cyclic Sequence. The duration band of 125 ms to 200 ms included results from studies that required participants to order cyclic patterns of syllables, tone sequences that had a relatively small range of frequencies, and sequences of numbers. In searching for reasons for this durational limit, two areas of research showed distinct similarities: psychological theory about echoic memory and the theta wavelength band of brain waves. The theory developed to explain this apparent correlation was that theta waves and echoic memory are different ways of explaining the same effect. In other words, the theta wave is the electronic trace of echoic memory.

A way of testing for a relationship between the theta wave and echoic memory, would be to design a study that tests whether the end points of echoic memory and the theta wave match. As theta waves are known to phase lock with changes in signals, it is predicted that both the theta wave and echoic memory will start at the same time. Study methods that established the boundaries of echoic memory could be repeated, but with the addition of ECG electrodes. These electrodes would firstly test that a theta wave started with a particular stimulus (i.e. testing for phase lock) and, if it did, then establish if the echoic memory effect lasted longer than the theta wave associated with the sound. This could be done by marking the end point of the theta wave and the time point at which masking no longer affects perception. If the duration of the echoic memory effect was less than the duration of the theta wave, the theory that the theta wave is the electronic trace of echoic memory would be supported, because it would establish that echoic memory exists entirely within the duration of a theta wave and therefore the theta wave is the physical manifestation of echoic memory.

A hypothesis to test the assumption that minimum durations for sound ordering is linked to the theta wave would be that *the point at which the ordering of sounds is no longer possible is the same point at which theta waves lose their phase lock with each individual tone*. A study to test for this could be designed by creating a sequence of contiguous tones of durations ranging from 50 ms to 200 ms. The participant should wear an ECG head net, positioned so that the theta waves related to sound perception were clearly readable. The participant should identify the order of cyclic sequences of tones, in the same manner as in previous research. The brain wave activity and the tone sequences should be recorded together, and the results examined to see if the loss of phase lock coincides with the participant's inability to order tones.

If these points matched, this would be strong evidence that the loss of phase locking with individual sounds causes the inability to order continuous cyclic sequences of tones at durations shorter than 125 ms and is therefore responsible for the limitation in MDPP in cyclic sequences.

The Beta Wave and the 50ms Barrier. In addition to limitations to sound perception imposed by the theta wave, that are also limitations that are related to the duration of beta waves. Although beta waves range from 12 to 30 Hz, Haenschel et. al. identified the "slow beta" range of between 12 and 20 hz (50 ms – 83 ms per wave) as those used to gather all the available information about a stimulus (Haenschel et al., 2000). The 50 ms (or 20 Hz) beta wave, which is the fastest of the 'slow beta' range associated with sound processing, matches the 50 ms sound processing transition points described earlier in this review. These points include the transition from discrete sound pulses to a sense of pitch, the transition from two sounds blurring together to being able to hear two discrete sounds and changing from hearing a repetition of a sound as a reverberation to hearing it as a distinct echo (see the Chapter 2 section entitled "15-75 ms: the 50 ms threshold and single presentation of sound sequences"). Based on this information, I assumed that the beta wave rate of oscillation limits the speed at which changes in frequency can be heard as individual sounds, at least in continuous sequences. Further, for tone durations shorter than a beta wave, information within a single beta wave cycle is summarised and a single pitch selected as a ‘best fit’ from information contained in the beta wave. This theory is supported by studies that have found that if the frequency of a sound changes more quickly than every 50 ms, the sound is reported as blurred or smeared (Joos, 1948; Schouten, 1962).

Proposed test to confirm the link between the beta wave and the 50ms barrier. A hypothesis for this assumption that could be tested would be that *the point at which sounds start to blur together will match the point at which beta waves lose their lock with each individual tone*.

A study to test for this could be constructed by creating a sequence of contiguous tones which, during the course of a trial run, are gradually shortened to segment durations shorter than 40ms and lengthened to segment durations longer than 60ms. The auditory effect would be that of the tone sequences speeding up and slowing down. The participant should wear an ECG head net positioned so that gamma and beta waves were clearly readable. The participant should be asked to press a button at the point at which sounds start to blur together as tone durations decrease and the point where sounds become clearly differentiated as tone durations increase. To match responses with tone sequences and brain waves, all data (the brain wave activity, the button press points and the tone sequences) should be recorded together. The results should be examined to check if the loss of phase lock of beta waves coincides with the point at which the sounds start to blur together, and that the phase lock is regained at the point that the participant indicates sounds could be clearly differentiated. The matching of these points would be strong evidence that the loss of phase locking with individual sounds causes the sounds to blur together.

Adaptive Methods for Frequency Change Detection.

Research has established that frequency change detection is a very important aspect of sound perception to the point that (at least in cats) four times as many nerve pathways are dedicated to frequency change detection than pitch identification. Therefore, for tone sequences of increasingly short durations, it is highly likely that frequency change detection is retained even after pitch perception is lost. However, the first study outlined in this thesis (see Chapter 3) established that, at very short durations (around 1ms), discriminating between changes of frequency of tones does not require either pitch identification or frequency change detection. It was suggested that, instead of detection of direction for frequency change, sound analogue comparison was the method used to differentiate between stimuli in studies where very short duration sounds were being studied. The study in Chapter 3 found that, using stimuli with constant timbre, differences in the direction of a frequency change were perceptible in tones as short as 1ms. However, once the timbre of stimuli was changed (even though identical frequency changes were used), it was no longer possible for participants to identify the direction of frequency change in stimuli with segment durations as short as 1ms.

It seems that if changes in frequency are too fast to identify individually, we still perceive these differences, but they are identified as changes in ‘quality’ rather than in frequency. In these very short frequency change studies, the only difference between stimuli was the order in which the tones were presented. An example of this is Wier and Green’s (1975) study. The authors stated that, at the shortest durations being studied, one stimulus sounded like a ‘tik’ whereas the other sounded like a ‘tok’. This can be perceived as the brain trying to identify the overall sound (i.e. the sound analogue) and presenting the words ‘tik’ and ‘tok’ as the best match for the sequences.

This relationship between the identification of changes in frequency at very short durations and describing differences between stimuli as different timbres resonates with other research in short tone perception. What we know as ‘timbre’ perception (i.e. differentiating between different sound sources, such as a clarinet and an oboe) may be the strategy used to differentiate between these very short sounds. Comments relating to ‘changes in the quality of the sound’ appear in other research into cyclic sounds. In discussing the way in which their participants differentiated between very short stimuli, Nickerson and Freeman (1974) noted that the stimuli in their study had the quality of a tonal buzz and that the different timbres of buzzes were able to be matched with one of the six stimuli that the participants had been required to differentiate. Warren et al. (1991) also stated that discrimination in their study was based on recognition of differences in quality between the two sound bursts, rather than by identifying the order of individual sound segments.

The method used to discriminate between these very short stimuli was described as sound analogue comparison. However, based on the verbal reports (above) it seems reasonable to postulate that sound analogue comparison is another way of describing the process of timbre identification, where the overall characteristics of a sound are matched against memories of similar characteristics. This would explain why instances describing frequency change stimuli as differences in timbres were found in research reports of frequency change studies.

Methods for Identification of Pitch in Very Short Sine Wave Tones.

Prior research has provided mixed results as to the masking effects of sine wave tones on other sine wave tones of varying relative pitch heights and temporal positions. It was therefore uncertain as to whether tones in these different positions required longer or shorter minimum durations for identification of pitch. Study 2 was designed to ensure not only an average minimum duration for tones in 3-tone and 2-tone sequences was derived but also minimum durations for tones in varying relative pitch heights and temporal positions.

Because there was no suitable method found for identifying the pitch of short contiguous tones, a method was created specifically for this purpose. The method of identifying the pitch of a tone, by matching a test tone against one of three piano tones either the same pitch, a semitone higher or a semitone lower, was found to be an effective way of testing pitch perception at short durations; and results were directly comparable with results from other researchers.

I am confident that only pitch perception was used to identify the tones in all study conditions outlined in Chapters 4 and 5. However, there were some issues in relation to other aspects of the method. Because intervals ranging in semitones from a minor third to an augmented 11th were used, some intervals were harder to identify than others. One cause for concern was the finding that octave intervals were identified more consistently than major sevenths and minor nineths. This is not surprising, given the prevalence of octaves in European music and the importance of the octave in sound perception (known as ‘pitch height’). However, because tone sequences were randomly allocated at each duration level (with constraints listed in Chapter 4), some duration levels may have been harder than others because of a prevalence of more difficult intervals in the tone sequences. Multiple tries might have been needed to move to lower durations at a hard duration level. This may have affected the results by making it harder to reach shorter durations within the allocated 50 stimulus presentations for each trial run. To rectify this issue in future research, consideration should be given to balancing interval difficulty at duration levels. This would of course need to be done while still ensuring that intervals did not become predictable or that information, other than the pitch of each individual segment, could not be used to assist with pitch identification.

Equivalent Durations for High and Low Frequency Tones. Researchers have found that melodic pitch was identifiable in the context of isolated tones when the duration of each tone reached 8 waves cycles. The method in this thesis, of using equal numbers of waves below 40ms (in which the duration of each tone differed depending upon the frequency of the tone) and equal duration for tones above 40ms, worked well in ensuring that tones of all frequencies could be identified equally at all duration levels. This was confirmed in Chapter 4, where no significant difference between high tone identification and low tone identification was found. However, this issue requires further consideration because, in a larger study (such as eight participants), the results for average accuracy of 79% for low tones and 87% for high tones would indicate that low tones were significantly harder to identify than high tones. This is despite exposure times for some low tones being over three times longer than the equivalent exposure times for high tones. However, this result runs counter to those of other researchers, like Doughty and Garner (1947), who found that 250Hz tones required less than twice the duration of high (1000 Hz) tones to establish a sense of pitch. Regardless, future researchers should consider using even longer durations for low frequency tones than equal number of sound waves, in order to fully balance out differences in minimum exposure duration times for perception of tones ranging from high frequency to low frequency tones. In summary, the duration ratios for high and low tones selected for this study were appropriately based on best practice and effectively countered identification differences in low and high tones, but consideration should be given in future studies to increasing the duration difference between high tones and low tones below 40ms.

The effect of small participant numbers on the reliability of the study. As outlined in the Limitations section of Chapter 5, reliance on a small sample of participants and the use of a single trial run for each participant limits generalisation of any results.

Nonetheless, it was considered sufficient to have four highly musically skilled participants for this study, given that such expertise should at least assist to test the lower limits of perceptual capacities.

The Minimum Duration for Perception of Pitch in Short Contiguous Tone Sequences

Short contiguous sequences are common both in music and in nature. Examples include arpeggiated chords in music and birdsong in nature. As previously discussed, although cyclically presented tones require exposure durations equivalent to theta waves for identification of pitch, short tones require far less time. This is because the unanalysed auditory trace of short, isolated tones can be stored in echoic memory and then processed more thoroughly. It seems a valuable and useful task to identify the MDPP in such short contiguous sequences. To address this gap in knowledge, many studies would need to be completed. In addition to the current study examining 2-tone and 3-tone sequences, sequence segment numbers would need to be increased until sufficient segments had been added to allow minimum durations for identification of pitch of the tone segments to equal the average MDPP in continuous cyclic tone sequences (i.e. 160 ms per tone segment).

In general, it is clear that tones that are presented both before and after target tones have a masking effect on target tones and that high tones positioned early in the sequence are least affected by masking. Conversely, the identification of low tones positioned early in the sequence is most affected by masking. A tendency noted here was that minimum exposure durations for identification of pitch tended to be shorter for low pitched tones in the last position (second for 2-tone sequences; or third for 3-tone sequences). Conversely, with high tones there was a small but consistent tendency across the two studies, whereby identification of pitch for high tones in the last position required longer minimum exposure durations than identification of pitch for high tones in the first position. This suggests that forward masking may have a greater effect on the highest tone in a sequence than backward masking. This effect may be related to the reasons behind the strong effect that high tones have on low tones that precede them. As previously described from research by Ciocca and Darwin (1999) and Grose, Hall, and Buss (2002), non-simultaneous tones can be integrated and perceived as if the two tones were a single tone and higher frequency tones presented after a primary tone can affect the perceived pitch of the primary tone to a greater extent than higher frequency tones presented before a primary tone. This may have a reciprocal (but clearly smaller) effect on high tones; high tones sounded after a low tone can be affected by the frequency of the low tone, in that the perceived pitch of the high tone is preconsciously adjusted to fit more closely with the pitch of the low tone. This would explain the longer durations required for identification of pitch of high tones in the last position.

Theta Wave Limitations for Short Tone Processing. As noted previously, shorter sounds can be reprocessed for extra information within the time period equating to a theta wave cycle. This implies that, as the number of tones in sequences increase, there will be a decrease in processing time for perception of pitch that relates to maximum duration of a theta wave (250ms). This is because, as more tones are added, the additional tones take up some of the limited processing time available for each tone in the segment sequence. For example, in a 2-tone sequence, each tone could be processed for half the 250ms available (assuming the correlated theta wave lasts the maximum duration recorded for theta waves). This means each tone could be processed for a maximum of 125ms. A 3-tone sequence would only have a maximum of 80ms available for processing each tone. This means that, to reach the same result for pitch identification, more information would need to be available to enable faster processing times. This is most likely achieved through increased exposure durations of each tone because, even though extra processing time is more effective than extra exposure time, longer exposure times still lead to improved identification with the same duration available for processing (Massaro, 1972b). This was indeed one of the findings of this research. The average minimum duration for identification of the pitch of an individual tone increased from 8 ms for single tones, through 24ms for tones within 2-tone sequences, to 47 ms for tones within 3-tone sequences. Based on this progression, 4-tone sequences might require average minimum durations of 100ms per segment; and that, for 5-tone sequences, the 160ms minimum duration for continuous cyclic patterns might be reached. However, it may also be the case that minimum durations stabilise at around 50ms because, at this point, tones no longer smear together. This alternative theory might mean that minimum durations might only gradually increase from this level to the cyclic tone identification minimum of 160 ms. Consistent with this suggestion, it should be noted that Schouten (1962) reported that, at 50ms, participants found the pitch of tones to be identifiable in continuous sequences, but the sequence order remained impossible to identify unless presented in a simple scalic sequence. Results from the 3-tone study reported in Chapter 5 run counter to this, however, because low tones positioned early in a sequence required up to 70ms for identification, which is longer than the 50ms mentioned in Schouten's report. It may be that some tones remain identifiable at comparatively short durations, such as high tones positioned early and the last tone in a sequence, whereas low and middle tones in any position, except the last position, require increasing durations for identification.

There is some evidence for increased minimum exposure durations being linked to increases in numbers of segments in a sequence in some of the cyclic sequence studies reported in the literature review. Within the duration bands identified in the literature review, there were irregularities, such as the 125ms minimum duration for identifying the order of four cyclic tones, compared to the 160ms minimum duration of identifying melodies with six to eight cyclic tones per presentation. It may be that the additional time required for identification of the melodies was needed to order the increased number of sounds in short-term memory. Thus, with extended cyclic sequence presentations, it may require even longer durations for perception of pitch. However, as sequence segment durations increase past 160 ms, this falls outside the bounds of the area of short-tone research that was the focus of this thesis.

Participant numbers and significant results. The problems encountered when recruiting participants for the studies in Chapter 5 have already been described in detail there and is not discussed further here. However, with only four participants in the 2-tone and 3-tone studies, it was encouraging to find such a strong and statistically significant difference between minimum durations for the identification of tones located at different positions in the sequences. It was noteworthy that, for 2-tone sequences, some participants were able to identify pitch at the same duration (or shorter) than the average MDPP when only single tones were presented. This may mean that, in some conditions, depending upon the position of the tone in relation to other tones, tones in 2-tone sequences are just as easy to identify as single isolated tones.

An Improved Method for Aural Training (Interval Recognition)

Both the interval recognition training method incorporating very short tones that was created for this thesis, ("Aural Train") and the alternative computer-based method "Auralia" (Rising, 2016). produced interval recognition over baseline measures to about the same extent. Nonetheless, the effect size of 0.5 for "Aural Train" was larger than for "Auralia" and it exceeded the minimum recommended by Hattie (2008) for a substantive improvement in student outcomes ('*d'* = 0.4). Moreover, the study was underpowered for a direct comparison with "Auralia". Despite there not being a statistically significant difference between the two computer-based programs, there were promising aspects to these results that suggested that "Aural Train" might provide a better option, if improvements were made to the training program.

The main benefit of computer-based instruction is that it can be designed to focus on the needs of each student. Advantages claimed for this approach to training include control by the learner over difficulty of content; rate and timing of delivery; place of learning; enhancement and efficiency for development of learning and reasoning skills; and cost savings (Chumley-Jones, Dobbie, & Alford, 2002). Intuitively, therefore, computer-based instruction would seem to offer the means for creating an improved learning environment. Nonetheless, there is clear evidence that poorly designed computer-based instruction that, for example, fails to take account of an initial poor level of skill (Ozeas, 1992), can fail to deliver optimal outcomes (Chumley-Jones et al., 2002). One reason for this is a failure to provide appropriate and timely instructions that are tailored to student needs (Chumley-Jones et al., 2002). Other critics have pointed to the lack social interaction in a one-to-one computer-based learning environment, which can adversely affect student learning (Shatri, 2020).

The aural training program "Aural Train" developed for this thesis did not overcome all the issues identified above. Although results were promising, further development would be needed to cater for the diverse range of needs of students in a classroom environment. Post-study consideration of participants' performances have suggested four areas for improvement: (i) creating targeted videos to improve instruction effectiveness, that are triggered by events such as constant inaccuracies; (ii) adding more 'game like' aspects to the training program to improve interest and motivation; (iii) providing opportunities for person-to-person interaction, to improve motivation and enable collaborative learning; and (iv) designing a larger study using the world wide web to provide a more accurate check of the program's effectiveness.

Creating Targeted Videos. The training program might be improved by referring students to additional instruction videos where they do not successfully identify intervals presented early during training. This would be especially important for students whose base-line skills are poor (Ozeas, 1992), because the added instructions would be targeted at individual students who struggled to understand the concepts required for successful completion of the exercises. A series of videos could be created that describe the difference between intervals in a variety of different ways, varying from presenting acoustic differences between the intervals in greater depth, to providing multiple models of students correctly (and possibly incorrectly) completing the tasks. For example, if a student fails to differentiate between the first two intervals presented at the most basic level, a video could be presented that introduces the concept of high and low pitch, describes how intervals are created and provides strategies for identifying the two intervals that were incorrectly identified (see Chapter 6 "how students develop aural skills" for an explanation of some of these strategies). Once the initial level was completed successfully, a second video, explaining how intervals can remain the same even though the pitch of the two tones change, could be automatically presented to struggling students but not played for those who did well in the first section.

It is important to provide teaching in a variety of modes and styles to help students with different learning styles master tasks (Rosenfeld & Rosenfeld, 2008). These different modes and styles can include the use of diagrams and animations for visual learners, detailed verbal explanations for auditory learners, modelling of correct identification as well as modelling incorrect identification and immediately providing strategies for correcting inaccurate choices for kinaesthetic learners. Videos designed to cater for a variety of learning styles could help maximise student learning. In the training study reported in Chapter 6, this was attempted in a limited way, in that a second explanation video was created and presented to all year 8 students after three weeks of training, because it was apparent that some students were not progressing. The video included real life examples of the intervals (i.e. presentations of recordings of music in which the target interval was prominent) and an explanation of how the change of an interval can affect the music. To take this further, specialised video might be added automatically at the start of each new exercise, so that students are only presented with examples and explanations of one or two intervals at a time at the start of each training period. Students would then be introduced to more intervals as they progressed through the exercises and displayed readiness for further complexity. Other videos would be triggered by students failing to identify intervals successfully that are covered in the targeted training activity.

Adding More 'Game-Like' Activities. Making the training exercises more game-like also has potential to improve the program. This could be achieved by adding an aiming component for selecting the correct answer and/or creating animated sequences between levels to help students visualize their progress. Completing one level successfully might lead to an animated sequence, such as going from a small town to a larger town or city to confer an increased sense of progress. It might also be possible to create a scenario where interval recognition is required to progress through the game. For example, locked doors might be opened by identifying a series of intervals. An alien echo locator might provide distance readings by sounding two tones with different frequencies (i.e. each musical interval equals a particular distance in the game). This might provide information that would enable the player to jump more accurately, identify the correct stepping distance for a puzzle, or provide targeting instructions for a gun.

Overcoming the isolation of individual workstations. Using computer-based training programs frequently requires that students focus on a computer screen and therefore exclude themselves from interacting with other students. While observing classes undertaking the aural training exercises, I noticed some students announcing their progress and scores, but they appeared disappointed when other students did not respond. Some students also remarked that they would prefer to work with a partner. Others said that they found the activities boring. However, the provision of dual headphone sets (or plugs) would enable students to work in pairs. This simple modification to procedure could overcome the absence of social interaction that is otherwise generally characteristic of a one-to-one computer-based learning environment. Collaborative learning has been found to improve critical thinking skills (Gokhale, 1995), so that permitting interaction in the way proposed may enable collaboration and therefore faster learning. Students could be encouraged by instruction to discuss their reasons for choosing a particular interval and learn from each other's ideas. However, care would need to be taken to ensure that one student did not dominate the pair and simply answer all the questions, instead of sharing responses between the two partners.

Access to the Aural Training Program for Large Numbers. Because the training program is web-based, it would not be difficult to create a website that could be made available to anyone with access to the internet. This would enable many more people to access the program and take advantage of this innovative method. Having large numbers of participants would also make it easier to test improvements, because data on usage rates, rate of improvements in skill level and suggestions for improvements to the training program would be available on a continuing basis.

Improving Study Methods for the Aural Training Study

As noted above, many students made little if any progress with either the "Aural Train" or "Auralia" program. On completion, students were questioned about the programs, with particular attention to the extent to which they felt engaged or found participation "interesting". Many students reported finding the aural training activities 'boring', although some did report that they found the "Aural Train" program fun. It would be worthwhile attempting to ascertain whether there are differences between the two programs in the extent to which students found them motivational. To accomplish this, a different study design would be more appropriate. This modified study would require students to try both programs sequentially (half starting with one program and half starting with the other one) and then to respond to a questionnaire designed to test their enjoyment and motivation regarding their progress through the exercises. Data could be collected on both their responses and their progress while using each program. These data would permit a comprehensive examination of this issue. Of course, it would also be necessary to check that a more motivational procedure did also deliver better training outcomes.

It is the case that the time required for such a complex study would require time outside that normally available for aural training within a secondary school's general music program. Considerable time would be necessary for students to join and to learn to use two different training programs. Additionally, time would be required to provide sufficient training time on each program to achieve some progress, as well as having participants complete a pre-test, an interim test and a post-test. Whether special arrangements might be made to overcome this problem is a matter for future consideration.

Using Elective Students. It is possible that this form of study might be more appropriate for elective music students in higher year levels or at university level. These students are more highly motivated to learn musical skills and usually have more time allocated to music in the timetable each week. However, because of the smaller numbers of students available at senior levels (because fewer students enrol in music educational programs), such a study would need to be managed across a long period of time, or in many schools within the same year.

Expanding Study Numbers. It has been recognised above that the training study was under-powered, recruiting insufficient numbers of students to establish a statistically significant result compared with the commercially available program. However, the power required in this particular case of around 300 participants should not, in the future, prove insurmountable. The need for larger participant numbers could be addressed by using the internet to promulgate the program, making it available for large numbers of participants to try. This procedure could be promoted to enable large numbers of appropriate participants to engage with the program and the accumulated results could then be tested for significant differences in skill levels achieved with different programs or for improvement over base-line levels. Although a comparison study could not be done in this way, a simple test for the extent of improvement over baseline could be established and, should the final effect size exceed 0.4 (the minimum recommended by Hattie (2008) for a substantive improvement in student outcomes), this outcome should be considered to be a successful result. Permission would be needed, and ethics approval would have to be sought, but there are many examples of this method of research on the web.

Final Comment

This thesis has aimed to improve understanding of pitch perception in short contiguous tone sequences by

- investigating whether contiguous tone sequences can be ordered at shorter segment durations than the tones can be identified by pitch,
- proposing that judgements about the shortest stimuli in past tone ordering studies have relied on sound analogue comparison rather than pitch perception,
- identifying the MDPP in 2-tone and 3-tone sequences,
- confirming that low tones require longer durations for identification when followed by high tones, and
- using this knowledge to develop improved method for training pitch perception among secondary school students.

The aims of the first study were met by establishing that frequency change direction detection was not possible for the shortest stimuli in Wier and Green’s (1975) study. The plausible alternative of sound analogue detection has been proposed, but further study is needed to confirm this theory. The success of the 2-tone and 3-tone studies was limited by the small number of participants and/or lack of multiple repetitions of trial runs. Nonetheless, these studies have provided a foundation for a feasible and comprehensive future research program, with an effective method for ensuring that stimuli are identified by their pitch and for investigating further the strong indication reported here of masking differences between tones in different positions. The last aim, to develop an improved aural training system, was less successfully met but that study has provided the basis for improved method, which requires further validation but does hold promise that the aim is achievable.

References

Abel, S. M. (1972). Discrimination of temporal gaps. *The Journal of the Acoustical Society of America, 52*(2B), 519-524. doi:10.1121/1.1913139

ACARA. (2015). Music glossary. v7.5. Retrieved from http://www.australiancurriculum.edu.au/the-arts/music/glossary

acara.edu.au. (2018). Australian Curriculum: The Arts. Foundation to Year 10. Retrieved from http://australiancurriculum.edu.au/f-10-curriculum/the-arts/music/?year=12747&capability=ignore&capability=Literacy&capability=Numeracy&capability=Information+and+Communication+Technology+%28ICT%29+Capability&capability=Critical+and+Creative+Thinking&capability=Personal+and+Social+Capability&capability=Ethical+Understanding&capability=Intercultural+Understanding&priority=ignore&priority=Aboriginal+and+Torres+Strait+Islander+Histories+and+Cultures&priority=Asia+and+Australia%E2%80%99s+Engagement+with+Asia&priority=Sustainability&elaborations=true&elaborations=false&scotterms=false&isFirstPageLoad=false

Albouy, P., Mattout, J., Bouet, R., Maby, E., Sanchez, G., Aguera, P. E., . . . Tillmann, B. (2013). Impaired pitch perception and memory in congenital amusia: the deficit starts in the auditory cortex. *Brain, 136*(Pt 5), 1639-1661. doi:10.1093/brain/awt082

Allen, J. B. (2008). Nonlinear cochlear signal processing and masking in speech perception. *Springer handbook of speech processing*, 27-60.

Amenedo, E., & Escera, C. (2000). The accuracy of sound duration representation in the human brain determines the accuracy of behavioural perception. *European Journal of Neuroscience, 12*(7), 2570-2574. doi:10.1046/j.1460-9568.2000.00114.x

Andrews, M. W., Dowling, W. J., Bartlett, J. C., & Halpern, A. R. (1998). Identification of speeded and slowed familiar melodies by younger, middle-aged, and older musicians and nonmusicians. *Psychology and aging, 13*(3), 462-471. doi:10.1037//0882-7974.13.3.462

ASA (2021) Pitch. In *Standards Store* (Vol. 11.01). https://asastandards.org/Terms/pitch/: Acoustical Society of America.

Bader, M., Schröger, E., & Grimm, S. (2017). How regularity representations of short sound patterns that are based on relative or absolute pitch information establish over time: An EEG study. *PloS one, 12*(5), e0176981.

Bregman, A. S. (1994). *Auditory scene analysis: The perceptual organization of sound*: MIT press.

Bregman, A. S., & Campbell, J. (1971). Primary auditory stream segregation and perception of order in rapid sequences of tones. *Journal of Experimental Psychology, 89*(2), 244-249. doi:10.1037/h0031163

Brophy, J. (1987). Synthesis of research on strategies for motivating students to learn. *Educational leadership, 45*(2), 40-48.

Buehrer, T. E. (2000). *An alternative pedagogical paradigm for aural skills: An examination of constructivist learning theory and its potential for implementation into aural skills curricula.* Indiana University Bloomington, IN,

Bürck, W., Kotowski, P., & Lichte, H. (1936). Frequenzspektrum und Tonerkennen. *Annalen der Physik, 417*(5), 433-449.

Burr, D. C., & Morgan, M. J. (1997). Motion deblurring in human vision. *Proceedings of the Royal Society B: Biological Sciences, 264*(1380), 431-436. doi:10.1098/rspb.1997.0061

Burton, R. L. (2015). *The elements of music: What are they, and who cares?* Paper presented at the Music: Educating for Life. ASME XXth National Conference Proceedings, Adelaide.

Buzsáki, G. (1989). Two-stage model of memory trace formation: a role for "noisy" brain states. *Neuroscience, 31*(3), 551-570. doi:10.1016/0306-4522(89)90423-5

Buzsáki, G. (2005). Theta rhythm of navigation: link between path integration and landmark navigation, episodic and semantic memory. *Hippocampus, 15*(7), 827-840. doi:10.1002/hipo.20113

Cakewalk. (2009). Sonor (Version 8.5). Boston: Cakewalk.

Cariani, P., & Micheyl, C. (2012). Toward a theory of information processing in auditory cortex. In *The Human Auditory Cortex* (pp. 351-390): Springer.

Campbell, M. R. (1999). Learning to teach music: A collaborative ethnography. Bulletin of the Council for Research in Music Education, 12-36.

Carl, D., & Gutschalk, A. (2013). Role of pattern, regularity, and silent intervals in auditory stream segregation based on inter-aural time differences. *Experimental brain research*, *224*(4), 557-570. doi:10.1007/s00221-012-3333-z

Chamberlin, D. D., & Boyce, R. F. (2016). SQL (Version SQL:2016) [Database program]. https://www.mysql.com/: SO/IEC.

Chumley-Jones, H. S., Dobbie, A., & Alford, C. L. (2002). Web-based learning: sound educational method or hype? A review of the evaluation literature. *Academic medicine*, *77*(10), S86-S93.

Ciocca, V., & Darwin, C. (1999). The integration of nonsimultaneous frequency components into a single virtual pitch. *The Journal of the Acoustical Society of America*, *105*(4), 2421-2430.

Colgin, L. L., Denninger, T., Fyhn, M., Hafting, T., Bonnevie, T., Jensen, O., . . . Moser, E. I. (2009). Frequency of gamma oscillations routes flow of information in the hippocampus. *Nature, 462*(7271), 353-357. doi:10.1038/nature08573

Corwin, J. (2009). The auditory system. Retrieved from http://www.medicine.virginia.edu/basic-science/departments/neurosci/research/corwin-lab/educationresearch/handouts/Auditory-System-handout-revised-for-2009-by-JTC.pdf website:

Cowan, M. (1936). *Pitch and intensity characteristics of stage speech*. Retrieved from

Cowan, N. (1984). On short and long auditory stores. *Psychological bulletin, 96*(2), 341-370. doi:10.1037/0033-2909.96.2.341

Cuddy, L. L., Balkwill, L. L., Peretz, I., & Holden, R. R. (2005). Musical difficulties are rare: a study of "tone deafness" among university students. *Annals of the New York Academy of Sciences, 1060*(1), 311-324. doi:10.1196/annals.1360.026

Davelaar, E., Goshen-Gottstein, Y., Ashkenazi, A., Haarmann, H., & Usher, M. (2005). The demise of short-term memory revisited: empirical and computational investigations of recency effects. *Psychological Review, 112*(1), 3-42. doi:10.1037/0033-295X.112.1.3

Deary, I. J., Head, B., & Egan, V. (1989). Auditory inspection time, intelligence and pitch discrimination. *Intelligence, 13*(2), 135-147.

De Cheveigne, A. (2005). Pitch perception models. *Pitch*, 169-233. doi:10.1007/0-387-28958-5_6

Demany, L., & Semal, C. (1988). Dichotic fusion of two tones one octave apart: Evidence for internal octave templates. *The Journal of the Acoustical Society of America, 83*(2), 687-695.

depts.washington.edu. (2013). The Auditory Nerve Response. Retrieved from http://depts.washington.edu/sphsc461/auditory%20nerve%20response/auditory%20nerve%20response.pdf

Dictionary.cambridge.org. (2019). TONE | meaning in the Cambridge English Dictionary. [online] Retrieved from https://dictionary.cambridge.org/dictionary/english/tone

Divenyi, P. L., & Hirsh, I. J. (1974). Identification of temporal order in three-tone sequences. *The Journal of the Acoustical Society of America*, *56*(1), 144-151. doi:10.1121/1.1903245

Divenyi, P. L., & Hirsh, I. J. (1975). The effect of blanking on the identification of temporal order in three-tone sequences. *Perception & Psychophysics*, *17*(3), 246-252.

Doughty, J. M., & Garner, W. R. (1947). Pitch characteristics of short tones; two kinds of pitch threshold. *The Journal of Experimental Psychology, 37*(4), 351-365. doi:10.1037/h0061516

Dowling, W. J., & Bartlett, J. C. (1981). The importance of interval information in long-term memory for melodies. *Psychomusicology: A Journal of Research in Music Cognition, 1*(1), 30.

Efron, R. (1970a). Effect of stimulus duration on perceptual onset and offset latencies. *Perception & Psychophysics, 8*(4), 231-234. doi:10.3758/BF03210211

Efron, R. (1970b). The minimum duration of a perception. *Neuropsychologia, 8*(1), 57-63. doi:10.1016/0028-3932(70)90025-4

Efron, R. (1970c). The relationship between the duration of a stimulus and the duration of a perception. *Neuropsychologia, 8*(1), 37-55. doi:10.1016/0028-3932(70)90024-2

Elliott, L. L. (1970). Pitch memory for short tones. *Perception & Psychophysics, 8*(5), 379-384.

Encyclopedia_Britannica. (2020). Pitch. Retrieved from https://www.britannica.com/art/pitch-music

Eshraghi, A. A., Nazarian, R., Telischi, F. F., Rajguru, S. M., Truy, E., & Gupta, C. (2012). The cochlear implant: historical aspects and future prospects. *The Anatomical Record: Advances in Integrative Anatomy and Evolutionary Biology, 295*(11), 1967-1980.

Evans, E. F. (1978). Place and time coding of frequency in the peripheral auditory system: some physiological pros and cons. *International Journal of Audiology, 17*(5), 369-420. doi:10.3109/00206097809072605

Farrow, D., & Abernethy, B. (2002). Can anticipatory skills be learned through implicit video based perceptual training? *Journal of sports sciences, 20*(6), 471-485.

Fastl, H., & Zwicker, E. (2007). *Psychoacoustics: Facts and models* (Vol. 22): Springer Science & Business Media.

Fearn, R., Carter, P., & Wolfe, J. (1999). The perception of pitch by users of cochlear implants: possible significance for rate and place theories of pitch. *Acoustics Australia, 27*(2-41).

Finn, B. F., J. (1993). Sibelius (Version 8.7). Burlington, Massachisetts: Avid.

FreeSoftwareFoundation. (1991). Audacity (Version 2.0.2). Boston: Free Software Foundation.

Galambos, R., Makeig, S., & Talmachoff, P. J. (1981). A 40-Hz auditory potential recorded from the human scalp. *Proceedings of the National Academy of Sciences of the United States of America, 78*(4), 2643-2647. doi:10.1073/pnas.78.4.2643

Galvin III, J. J., Fu, Q.-J., & Nogaki, G. (2007). Melodic contour identification by cochlear implant listeners. Ear and hearing, 28(3), 302.

Gay, T. (1968). Effect of speaking rate on diphthong formant movements. *The Journal of the Acoustical Society of America, 44*(6), 1570-1573. doi:10.1121/1.1911298

Gokhale, A. A. (1995). Collaborative learning enhances critical thinking.

Graves, J. E., & Oxenham, A. J. (2017). Familiar tonal context improves accuracy of pitch interval perception. *Frontiers in psychology*, *8*, 1753.

Green, D. M. (1973). Temporal acuity as a function of frequency. *The Journal of the Acoustical Society of America, 54*(2), 373-379. doi:10.1121/1.1913587

Greenberg, S., Marsh, J. T., Brown, W. S., & Smith, J. C. (1987). Neural temporal coding of low pitch. I. Human frequency-following responses to complex tones. *Hearing Research*, *25*(2-3), 91-114. doi:10.1016/0378-5955(87)90083-9

Greenwood, D. D. (1961). Auditory masking and the critical band. *The Journal of the Acoustical Society of America, 33*(4), 484-502.

Grose, J. H., Hall, J. W., 3rd, & Buss, E. (2002). Virtual pitch integration for asynchronous harmonics. *The Journal of the Acoustical Society of America, 112*(6), 2956-2961. doi:10.1121/1.1514934

Grudnik, J. L., & Kranzler, J. H. (2001). Meta-analysis of the relationship between intelligence and inspection time. *Intelligence, 29*(6), 523-535. doi:10.1016/s0160-2896(01)00078-2

Gu, Y., & Liljenstrom, H. (2007). A neural network model of attention-modulated neurodynamics. *Cognitive Neurodynamics, 1*(4), 275-285. doi:10.1007/s11571-007-9028-7

Guttman, N., & Julesz, B. (1963). Lower limits of auditory periodicity analysis. *The Journal of the Acoustical Society of America, 35*(4), 610-610. doi:10.1121/1.1918551

Haenschel, C., Baldeweg, T., Croft, R. J., Whittington, M., & Gruzelier, J. (2000). Gamma and beta frequency oscillations in response to novel auditory stimuli: A comparison of human electroencephalogram (EEG) data with in vitro models. *Proceedings of the National Academy of Sciences, 97*(13), 7645-7650. doi:10.1073/pnas.120162397

Hajiaghababa, F., Marateb, H. R., & Kermani, S. (2018). The design and validation of a hybrid digital-signal-processing plug-in for traditional cochlear implant speech processors. *Computer methods and programs in biomedicine, 159,* 103-109.

Hattie, J. (2008). *Visible learning: A synthesis of over 800 meta-analyses relating to achievement*. Abingdon, Oxon: Routledge.

Hattie, J., Biggs, J., & Purdie, N. (1996). Effects of learning skills interventions on student learning: A meta-analysis. *Review of educational research, 66*(2), 99-136.

Hawkins, H. L., Thomas, G. B., Presson, J. C., Cozic, A., & Brookmire, D. (1974). Precategorical selective attention and tonal specificity in auditory recognition. *The Journal of Experimental Psychology, 103*(3), 530-538. doi:10.1037/h0037138

Hawkins, J. E. (2018). Human ear - Transmission of sound within the inner ear. Retrieved from https://www.britannica.com/science/ear

HearingLink. (2018). How the ear works. Retrieved from https://www.hearinglink.org/your-hearing/about-hearing/how-the-ear-works/

Hirsh, I. J. (1959). Auditory perception of temporal order. *The Journal of the Acoustical Society of America, 31*, 759-767. doi:10.1121/1.1907782

Hofstetter, F. T. (1981). *Applications of the GUIDO system to aural skills research, 1975-80.* Paper presented at the College music symposium.

Hopfield, J. J. (1995). Pattern recognition computation using action potential timing for stimulus representation. *Nature, 376*(6535), 33-36.

Joos, M. (1948). Acoustic phonetics. *Language, 24*(2), 5-136. doi:10.2307/522229

Kalmus, H., & Fry, D. B. (1980). On tune deafness (dysmelodia): frequency, development, genetics and musical background. *Ann Hum Genet, 43*(4), 369-382. doi:10.1111/j.1469-1809.1980.tb01571.x

Kallman, H. J., & Massaro, D. W. (1979). Similarity effects in backward recognition masking. *Journal of Experimental Psychology: Human Perception and Performance, 5*(1), 110.

Karpicke, J. D., & Roediger, H. L., 3rd. (2008). The critical importance of retrieval for learning. *Science, 319*(5865), 966-968. doi:10.1126/science.1152408

Kariuki, P. N., & Ross, Z. R. (2017). The Effects of Computerized and Traditional Ear Training Programs on Aural Skills of Elementary Students. *Online Submission*.

Kidd, G., & Colburn, H. S. (2017). Informational masking in speech recognition. *The auditory system at the cocktail party*, 75-109.

Kisley, M. A., & Cornwell, Z. M. (2006). Gamma and beta neural activity evoked during a sensory gating paradigm: effects of auditory, somatosensory and cross-modal stimulation. *Clinical Neurophysiology, 117*(11), 2549-2563. doi:10.1016/j.clinph.2006.08.003

Klimesch, W. (1999). EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain Research Reviews, 29*(2-3), 169-195. doi:10.1016/s0165-0173(98)00056-3

Krumhansl, C. L. (1995). Music psychology and music theory: Problems and prospects. *Music Theory Spectrum, 17*(1), 53-80

Krumbholz, K., Patterson, R., Seither-Preisler, A., Lammertmann, C., & Lütkenhöner, B. (2003). Neuromagnetic evidence for a pitch processing center in Heschl's gyrus. *Cerebral Cortex, 13*(7), 765-772. doi:10.1093/cercor/13.7.765

Ladefoged, P. N. (2018). Phonetics. Retrieved from https://www.britannica.com/science/phonetics/Suprasegmentals

Leek, M. R. (2001). Adaptive procedures in psychophysical research. *Perception & Psychophysics*, *63*(8), 1279-1292. doi:10.3758/bf03194543

Leshowitz, B., & Cudahy, E. (1972). Masking with continuous and gated sinusoids. *The Journal of the Acoustical Society of America, 51*(6), 1921-1929. doi:10.1121/1.1913051

Leshowitz, B., & Cudahy, E. (1973). Frequency discrimination in the presence of another tone. *The Journal of the Acoustical Society of America, 54*(4), 882-887.

Leshowitz, B., & Hanzi, R. (1972). Auditory pattern discrimination in the absence of spectral cues. *The Journal of the Acoustical Society of America*, *52*(1A), 166-166.

Li, D., & Cowan, N. (2014). Auditory Memory. In *Encyclopedia of Computational Neuroscience* (pp. 1-3): Springer.

Lo, C. Y., McMahon, C. M., Looi, V., & Thompson, W. F. (2015). Melodic contour training and its effect on speech in noise, consonant discrimination, and prosody perception for cochlear implant recipients. Behavioural Neurology, 2015.

Loh, C. S. (2007). Choice and effects of instrument sound in aural training. *Music Education Research, 9*(1), 129-143.

Luo, H., & Poeppel, D. (2012). Cortical oscillations in auditory perception and speech: evidence for two temporal windows in human auditory cortex. *Frontiers in psychology, 3*, 170. doi:10.3389/fpsyg.2012.00170

Lütkenhöner, B., & Poeppel, D. (2011). *From tones to speech: magnetoencephalographic studies*. In *The Auditory Cortex* (pp. 597-615). doi:10.1007/978-1-4419-0074-6_28

Lynch, T., & Maclean, J. (2000). Exploring the benefits of task repetition and recycling for classroom language learning. *Language Teaching Research, 4*(3), 221-250.

Marks, L. E. (1987). On cross-modal similarity: Auditory–visual interactions in speeded discrimination. *Journal of Experimental Psychology: Human Perception and Performance, 13*(3), 384.

Massaro, D. W. (1970). Preperceptual auditory images. *The Journal of Experimental Psychology, 85*(3), 411-417. doi:10.1037/h0029712

Massaro, D. W. (1971). Effect of masking tone duration on preperceptual auditory images. *Journal of Experimental Psychology, 87*(1), 146.

Massaro, D. W. (1972a). Preperceptual images, processing time, and perceptual units in auditory perception. *Psychol Rev, 79*(2), 124-145. doi:10.1037/h0032264

Massaro, D. W. (1972b). Stimulus information vs processing time in auditory pattern recognition. *Perception & Psychophysics, 12*(1), 50-56. doi:10.3758/BF03212841

Massaro, D. W. (1973). A comparison of forward versus backward recognition masking. *Journal of Experimental Psychology, 100*(2), 434.

Massaro, D. W. (1974). Perceptual units in speech recognition. *The Journal of Experimental Psychology, 102*(2), 199-208. doi:10.1037/h0035854

Massaro, D. W. (1975). Backward recognition masking. *The Journal of the Acoustical Society of America, 58*(5), 1059-1065. doi:10.1121/1.380765

Massaro, D. W., Cohen, M. M., & Idson, W. L. (1976). Recognition masking of auditory lateralization and pitch judgments. *The Journal of the Acoustical Society of America, 59*(2), 434-441. doi:10.1121/1.380887

Massaro, D. W., & Idson, W. L. (1978). Target-mask similarity in backward recognition masking of perceived tone duration. *Perception & Psychophysics*, *24*(3), 225-236.

McDermott, J. H., & Oxenham, A. J. (2008). Music perception, pitch, and the auditory system. *Current opinion in neurobiology, 18*(4), 452-463. doi:10.1016/j.conb.2008.09.005

McPherson, G. E. (2005). From child to musician: Skill development during the beginning stages of learning an instrument. *Psychology of music, 33*(1), 5-35.

Micheyl, C., Delhommeau, K., Perrot, X., & Oxenham, A. J. (2006). Influence of musical and psychoacoustical training on pitch discrimination. Hearing Research, 219(1-2), 36-47. doi:10.1016/j.heares.2006.05.004

Middlebrooks, J. C., & Simon, J. Z. (2017). Ear and brain mechanisms for parsing the auditory scene. In *The auditory system at the cocktail party* (pp. 1-6): Springer.

Miller, G. A. T., W. G. (1948). The Perception of Repeated Bursts of Noise. *The Journal of the Acoustical Society of America, 20*(2), 171-182. doi:10.1121/1.1906360

Millisecond. (2014). Inquisit (Version 4.0.5). Seattle: Millisecond.

Molholm, S., Martinez, A., Ritter, W., Javitt, D. C., & Foxe, J. J. (2005). The neural circuitry of pre-attentive auditory change-detection: an fMRI study of pitch and duration mismatch negativity generators. *Cerebal Cortex, 15*(5), 545-551. doi:10.1093/cercor/bhh155

Moore, B. C. (2012). *An introduction to the psychology of hearing*: Brill.

MusicEdNet. (2018). Auralia. Retrieved from https://www.musicednet.com/Products/Software/Auralia-Cloud-Edition.aspx

NAfME. (2015). Core music standards glossary. Retrieved from http://www.nafme.org/my-classroom/standards/core-music-standards/

Nave, C. R. (1999). Hyperphysics -Pitch. *Nave, C R.* Retrieved from http://hyperphysics.phy-astr.gsu.edu/hbase/sound/pitch.html

Neff, D. L., & Callaghan, B. P. (1988). Effective properties of multicomponent simultaneous maskers under conditions of uncertainty. *The Journal of the Acoustical Society of America, 83*(5), 1833-1838.

Neisser, U. (1967). *Cognitive psychology*. New York: Appleton-Century-Crofts.

Nettelbeck, T. (1987). Inspection time and intelligence.

Nickerson, R., & Freeman, B. (1974). Discrimination of the order of the components of repeating tone sequences: Effects of frequency separation and extensive practice. *Perception & Psychophysics, 16*(3), 471-477. doi:10.3758/BF03198574

Ozeas, N. L. (1992). The effect of the use of a computer assisted drill program on the aural skill development of students in beginning solfege

Painter, T., & Spanias, A. (2000). Perceptual coding of digital audio. *Proceedings of the IEEE, 88*(4), 451-515.

Papastergiou, M. (2009). Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. *Computers & Education, 52*(1), 1-12.

Partti, H., & Karlsen, S. (2010). Reconceptualising musical learning: New media, identity and community in music education. Music Education Research, 12(4), 369-382.

Patel, A. D. (2012). The OPERA hypothesis: assumptions and clarifications. Annals of the New York Academy of Sciences, 1252(1), 124-128.

Patterson, J., & Green, D. (1970). Discrimination of transient signals having identical energy spectra. *The Journal of the Acoustical Society of America, 48*, 894. doi:10.1121/1.1912229

Patterson, R. (1973). The effects of relative phase and the number of components on residue pitch. *The Journal of the Acoustical Society of America, 53*(6), 1565-1572. doi:10.1121/1.1913504

Patterson, R., & Green, D. M. (2012). Auditory masking. *Handbook of perception, 4*, 337-361.

Patterson, R., Peters, R., & Milroy, R. (1983). *Threshold duration for melodic pitch.* Paper presented at the Proceedings of the 6th International Symposium on Hearing, Bad Nauheim, Germany.

Patterson, R., Robinson, K., Holdsworth, J., McKeown, D., Zhang, C., & Allerhand, M. (1992). Complex sounds and auditory images. *Auditory physiology and perception, 83*, 429-446.

Patterson, R., Uppenkamp, S., Johnsrude, I., & Griffiths, T. (2002). The processing of temporal pitch and melody information in auditory cortex. *Neuron, 36*(4), 767-776. doi:10.1016/s0896-6273(02)01060-7

Pfordresher, P. Q., & Brown, S. (2007). Poor-pitch singing in the absence of" tone deafness". *Music Perception: An Interdisciplinary Journal, 25*(2), 95-115.

PHP, T. G. (2018). PHP: Hypertext Preprocessor. Retrieved from http://php.net/

Pierce, J. R. (1992). *The science of musical sound*. New York: Scientific American Library

Plomp, R. (1964). Rate of decay of auditory sensation. *The Journal of the Acoustical Society of America, 36*(2), 277-282. doi:10.1121/1.1918946

Pollack, I. (1967). Number of pulses required for minimal pitch. *The Journal of the Acoustical Society of America, 42*(4), 895. doi:10.1121/1.1910663

Pollack, I. (1968). The apparent pitch of short tones. *The American Journal of Psychology, 81*(2), 165-169. doi:10.2307/1421260

Pollack, I. (1990). Detection and discrimination thresholds for auditory periodicity. *Percept Psychophys, 47*(2), 105-111. doi:10.3758/bf03205974

Prins, N. (2016). *Psychophysics: a practical introduction*: Academic Press.

Rising. (2016). Auralia. Fairfield, Victoria: Rising Software.

Robinson, K., & Patterson, R. (1995). The duration required to identify the instrument, the octave, or the pitch chroma of a musical note. *Music Perception*, 1-15. doi:10.2307/40285682

Ronken, D. A. (1972). Changes in frequency discrimination caused by leading and trailing tones. The Journal of the Acoustical Society of America, 51(6B), 1947-1950.

Rosenfeld, M., & Rosenfeld, S. (2008). Developing effective teacher beliefs about learners: The role of sensitizing teachers to individual learning differences. *Educational Psychology, 28*(3), 245-272.

Saldanha, E., & Corso, J. F. (1964). Timbre cues and the identification of musical instruments. *The Journal of the Acoustical Society of America, 36*(11), 2021-2026. doi:10.1121/1.1937169

Santos, A., Joly-Pottuz, B., Moreno, S., Habib, M., & Besson, M. (2007). Behavioural and event-related potentials evidence for pitch discrimination deficits in dyslexic children: Improvement after intensive phonic intervention. *Neuropsychologia, 45*(5), 1080-1090.

Schneider, P., Sluming, V., Roberts, N., Scherg, M., Goebel, R., Specht, H. J., . . . Rupp, A. (2005). Structural and functional asymmetry of lateral Heschl's gyrus reflects pitch perception preference. *Nature neuroscience, 8*(9), 1241-1247. doi:10.1038/nn1530

Schouten, J. (1962). *On the perception of sound and speech.* Paper presented at the Proceedings of the 4th International Congress on Acoustics (Copenhagen).

Seashore, C. E. (1938). *Psychology of music*: New York: Dover Publications.

Selezneva, E., Gorkin, A., Budinger, E., & Brosch, M. (2018). Neuronal correlates of auditory streaming in the auditory cortex of behaving monkeys. *European Journal of Neuroscience, 48*(10), 3234-3245.

Semal, C., & Demany, L. (2006). Individual differences in the sensitivity to pitch direction. *The Journal of the Acoustical Society of America, 120*(6), 3907-3915. doi:10.1121/1.2357708

Shatri, Z. G. (2020). Advantages and Disadvantages of Using Information Technology in Learning Process of Students. *Journal of Turkish Science Education, 17*(3), 420-428.

Souza, V. M., Batista, G. E., & Souza-Filho, N. E. (2015). *Automatic classification of drum sounds with indefinite pitch.* Paper presented at the 2015 International Joint Conference on Neural Networks (IJCNN).

Sparks, D. W. (1976). Temporal recognition masking—or interference? *The Journal of the Acoustical Society of America, 60*(6), 1347-1353.

Stambaugh, L. A., & Nichols, B. E. (2020). The relationships among interval identification, pitch error detection, and stimulus timbre by preservice teachers. *Journal of Research in Music Education, 67*(4), 465-480.

Steinschneider, M., Liégeois-Chauvel, C., & Brugge, J. F. (2011). Auditory evoked potentials and their utility in the assessment of complex sound processing. In *The auditory cortex* (pp. 535-559): Springer.

Terhardt, E. (1974). Pitch, consonance, and harmony. *The Journal of the Acoustical Society of America, 55*(5), 1061-1069. doi:10.1121/1.1914648

Thackray, R. (1975). Some thoughts on aural training. *Australian Journal of Music Education*(17), 25.

Thavam, S., & Dietz, M. (2019). Smallest perceivable interaural time differences. *The Journal of the Acoustical Society of America, 145*(1), 458-468.

Thomas, I. B., & Fitzgibbons, P. J. (1971). Temporal order and perceptual classes. *The Journal of the Acoustical Society of America, 50*(1A), 86-87. doi:10.1121/1.1977756

Thomas, I. B., Hill, P. B., Carroll, F. S., & Garcia, B. (1970). Temporal order in the perception of vowels. *The Journal of the Acoustical Society of America, 48*(4), 1010-1013. doi:10.1121/1.1912221

UHAC. (2019). How We Hear: 6 Basic Steps to Hearing Retrieved from https://www.uhac.ca/hearing-loss/how-we-hear/

van Dijk, P., & Backes, W. H. (2003). Brain activity during auditory backward and simultaneous masking tasks. *Hearing Research, 181*(1-2), 8-14.

van Noorden, L. (1975). Temporal coherence in the perception of tone sequences. *Ph. D. Thesis*.

VanRullen, R., Zoefel, B., & Ilhan, B. (2014). On the cyclic nature of perception in vision versus audition. *Philosophical Transactions of the Royal Society of London, 369*(1641), 20130214. doi:10.1098/rstb.2013.0214

Warren, R. M. (1972). Perception of temporal order: Special rules for the initial and terminal sounds of sequences. *The Journal of the Acoustical Society of America, 52*(1A), 167-167.

Warren, R. M. (1974). Auditory temporal discrimination by trained listeners. *Cognitive Psychology, 6*(2), 237-256. doi:10.1016/0010-0285(74)90012-7

Warren, R. M., & Bashford, J. A. (1993). When acoustic sequences are not perceptual sequences: The global perception of auditory patterns. *Perception & Psychophysics, 54*(1), 121-126.

Warren, R. M., Bashford, J. A., & Gardner, D. A. (1990). Tweaking the lexicon: Organization of vowel sequences into words. *Perception & Psychophysics*, *47*(5), 423-432. doi:10.3758/BF03208175

Warren, R. M., Gardner, D. A., Brubaker, B. S., & Bashford Jr, J. A. (1991). Melodic and nonmelodic sequences of tones: Effects of duration on perception. *Music Perception*, 277-289. doi:10.2307/40285503

Warren, R. M., & Obusek, C. J. (1972). Identification of temporal order within auditory sequences. *Perception & Psychophysics*, *12*(1), 86-90. doi:10.3758/BF03212848

Warren, R. M., Obusek, C. J., Farmer, R. M., & Warren, R. P. (1969). Auditory sequence: Confusion of patterns other than speech or music. *Science*, *164*(3879), 586-587. doi:10.1126/science.164.3879.586

Watson, C. S., Wroton, H. W., Kelly, W. J., & Benbassat, C. A. (1975). Factors in the discrimination of tonal patterns. I. Component frequency, temporal position, and silent intervals. *The Journal of the Acoustical Society of America, 57*(5), 1175-1185.

Wegel, R., & Lane, C. (1924). The auditory masking of one pure tone by another and its probable relation to the dynamics of the inner ear. *Physical review*, *23*(2), 266.

WHATWG, W. C. (2017). HTML (Version HTML5) [Hyper Text Markup Language]. https://www.w3.org/html/ https://whatwg.org/: W3C & WHATWG.

Whitfield, I. C. (1979). Periodicity, pulse interval and pitch. *Audiology, 18*(6), 507-512. doi:10.3109/00206097909072641

Whitfield, I. C., & Evans, E. F. (1965). Responses of Auditory Cortical Neurons to Stimuli of Changing Frequency. *Journal of neurophysiology, 28*, 655-672. doi:10.1152/jn.1965.28.4.655

Wier, C. C., & Green, D. M. (1975). Temporal acuity as a function of frequency difference. *The Journal of the Acoustical Society of America, 57*, 1512-1515. doi:10.1121/1.1914184

Winckel, F. (1967). *Music, Sound and Sensation* (T. Binkley, Trans.). New York Dover.

Williams, L. F. (2005). Reflexive Ethnography: An Ethnomusicologist's Experience as a Jazz Musician in Zimbabwe. Black Music Research Journal, 25(1/2), 155-165.

www.gov.uk. (2013). Music programmes of study. *Statutory guidance National curriculum in England:*. Retrieved from https://www.gov.uk/government/publications/national-curriculum-in-england-music-programmes-of-study/national-curriculum-in-england-music-programmes-of-study

Zanto, T. P., Large, E. W., Fuchs, A., & Kelso, J. S. (2005). Gamma-band responses to perturbed auditory sequences: evidence for synchronization of perceptual processes. *Music Perception, 22*(3), 531-547.

Zatorre, R. J., & Schönwiesner, M. (2011). Cortical speech and music processes revealed by functional neuroimaging. In *The Auditory Cortex* (pp. 657-677): Springer.

Zwislocki, J. J. (1969). Temporal summation of loudness: an analysis. *The Journal of the Acoustical Society of America*, *46*(2), 431-441. doi:10.1121/1.1911708

Appendix 1 The Elements of Music: What Are They and Who Cares?

Abstract

One of the issues music teachers face in their efforts to educate students is the inconsistency of important musical terms. Until now this has not been a particular concern, because every music teacher uses the definitions they are most comfortable with. However with the advent of the National curriculum, with its mandated curriculum concepts and teacher evaluations based on successful implementation of set curricula, there seems to be a greater sense of urgency that the documents we are directed to teach from are accurate and pedagogically sound. One definition which seems particularly important is the definition of "the elements of music", which the ACARA document defines as: rhythm, pitch, dynamics and expression, form and structure, timbre and texture (acara.edu.au, 2013). These elements however are far from universally accepted and do not match the curriculum documents of either the USA or UK (Education.gov.uk, 2011; Musiced.nafme.org, 2013). In addition, one very important point that seems to have been overlooked is that Music is an art form, and any list of elements of music should reflect this undeniable fact. The theory proposed in this paper is that as sound consists of the same elements regardless of whether it is music, speech or naturally produced, the elements of sound should be listed separately as: pitch, dynamics, rhythm, texture, timbre and spatial location. Music requires more than just sound however, so this paper proposes that the elements of music be: sound, structure and artistic intent or effect.

Keywords: music, elements, cognition, education, curriculum.

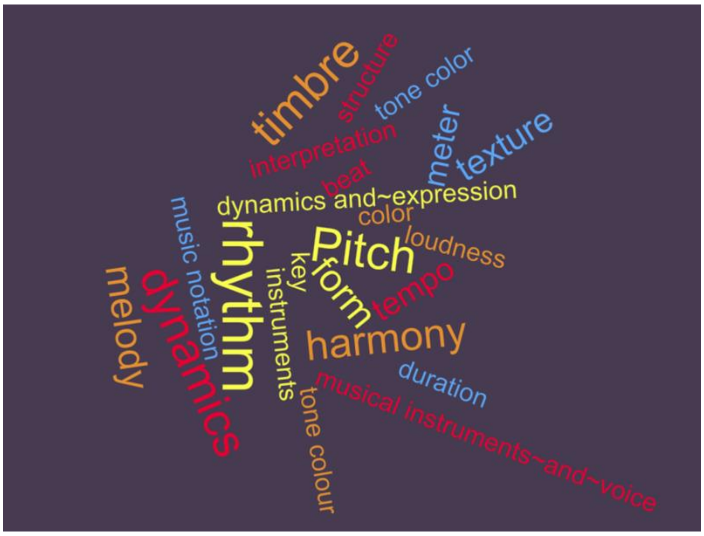
One of the issues music teachers face in their efforts to educate students is the inconsistency of important musical terms. Until now this has not been a particular concern, because every music teacher uses the definitions they are most comfortable with. However, with the advent of the National curriculum with its mandated curriculum concepts and teacher evaluations based on successful implementation of set curricula, there seems to be a greater sense of urgency that the documents we are directed to teach from are accurate and pedagogically sound.

One definition which seems particularly important, as evidenced by the number of times it is mentioned in the ACARA curriculum document, is the definition of "the elements of music". These elements are listed as: rhythm, pitch, dynamics and expression, form and structure, timbre and texture (acara.edu.au, 2013). However, after extensive research, there seems to be little evidence as to the veracity of this or any other list of the elements of music.

One would assume that the phrase "the elements of music" would bring to everyone's mind an identical list of words in the same way that "the table of elements" does in the area of Chemistry. However this is not the case. There are widely disparate definitions of the elements of music both in curriculum documents and on the internet (see figure 1). The title of every list: "*The* elements of music", implies each list is absolute and comprehensive. This is clearly not the case.

Figure 1

Words used in lists of the elements of music



The elements of music feature prominently in the current music curriculums not only of Australia, but also USA and UK (acara.edu.au, 2013; Education.gov.uk, 2011; Musiced.nafme.org, 2013). Unfortunately, some of the concepts identified as the elements of music by the three curriculum documents do not match. All three curriculums identify “pitch”, “dynamics” and “timbre” as elements, but the other identified elements of music are far from universally agreed. Different words are used to describe concepts relating to changes in note length: USA and Australia use the term “rhythm”, while UK uses the terms: “duration” and “tempo” to identify the time-based element of music. The words “form” and “structure” seem to be used interchangeably, with UK using “structure”, USA using “form” and Australia using both “structure” and “form”. There also seems to be complete disagreement amongst the curriculum documents as to the inclusion of “harmony” (USA only) and “expression” (Australia only) as elements of music. As the elements of music feature so strongly in these curriculum documents, it seems a worthwhile pedagogical goal to establish an agreed basis for the identification of the elements of music. Unfortunately, complications to this goal arise immediately.

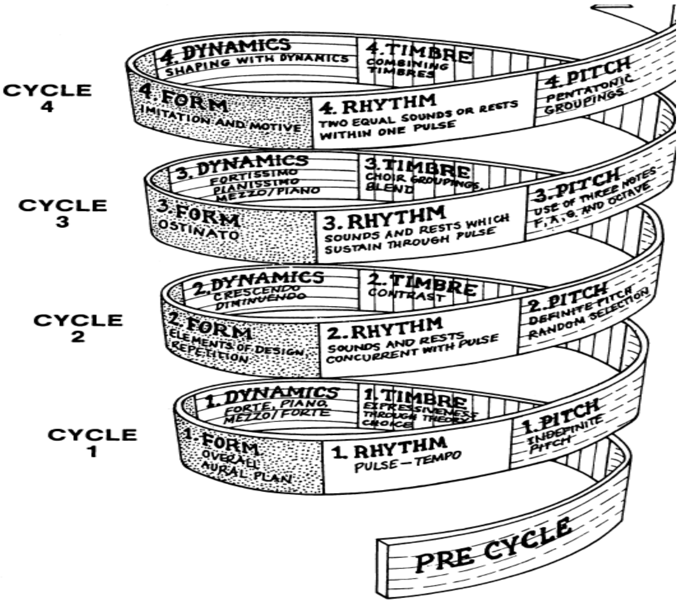
It seems the phrase “the elements of music” has been used in a number of different contexts both in contemporary times and over the years. In the 1800s, the phrase “the elements of music” and the phrase “the rudiments of music” were used interchangeably (Clementi, 1974; Niecks, 1884). The elements described in these documents might nowadays be considered more the elements of music performance than the elements of music, because they refer to aspects of music that are needed in order to become a musician. Recent writers such as Estrella (Estrella, 2015) seem to be using the phrase “elements of music” in a manner similar to that used in the 1800’s.

Since the emergence of the study of psychoacoustics, most lists of elements of music have related more to how we hear music than how we learn to play it. Seashore, in his book “Psychology of Music” (Seashore, 1938), identified four “psychological attributes of sound”. These were: “pitch, loudness, time, and timbre” (p. 3). He did not call them the “elements of music” but referred to them as “elemental components” (p. 2). Nonetheless these elemental components seem to link very precisely with four of the most common musical elements: Pitch links with pitch, timbre with timbre, loudness links with dynamics, and time links with the time based concepts of rhythm, duration and tempo.

Since Seashore’s time a number of authors have introduced lists of elements of music. By far the most well-known and influential was the Manhattanville Music Curriculum Program (Thomas, 1970). This group of curriculum experts introduced the elements of music to the American education system. Their selection of pitch, rhythm, dynamics, timbre and form closely mirrored Seashore’s elemental components, with the addition of form. This list was so influential that most American schools and Universities adopted them and their related spiral curriculum (see figure 2 below). Since 1970, the concepts generally selected as elements of music have expanded to include: “form”, “texture”, “harmony” and a very recent addition: “spatial location” (Levitin, 1999).

Figure 2

America’s spiral curriculum



Source: Ronald B. Thomas, *Manhattanville Music Curriculum Program Final Report, Part 1* (United States Office of Education, ED 045 865, August 1970), p.39.

In the goal of adopting a single set of elements of music for common usage, it is necessary to discover why lists of elements have included concepts as disparate as “expression” and “musical instruments” (acara.edu.au, 2013; Estrella, 2015). The main issue seems to be related to the definition of the word “element”. Webster’s New 20th Century Dictionary defines an element as: “a substance which cannot be divided into a simpler form by known methods” (Webster, 1947). This definition seems to align with Seashore’s “elemental compounds” as well as most of the educational institutions’ lists of elements. Other lists of elements seem more closely aligned with definitions such as: “the rudimentary principles of an art, science, etc.: the elements of grammar” (Dictionary.com 2013b). These two disparate ways of defining the word “element” can be differentiated by describing them as a) the “perceptual elements of music” and b) the “rudimentary elements of music”. This paper will be referring exclusively to the “perceptual elements of music”.

Now that word definition issues have been examined, and in a first step towards defining the elements of music, it would be beneficial to explore the elements of sound. Sound is readily dividable into two simple and irreducible elements: pressure and time. These fundamental elements form the basis of sound waves and can describe in absolute terms every sound we hear. However, this very basic level of analysis is not very helpful. It is like saying that the chemical elements of matter consists of electrons, protons and neutrons. Basically true, but it does not help us in understanding the properties of different types of atoms. A further level of analysis is required.

In relation to chemistry, this next level of chemical analysis involves understanding how many of each fundamental element (electron, proton and neutron) make up each particular type of atom. This information can then be used to understand how different atoms behave. In relation to sound, the next level of analysis involves understanding how the fundamental elements of sound (pressure and time) are transformed and analysed by the process of perception and cognition.

It is important at this point to decide if this next level of analysis refers to the elements of sound or the elements of music. In this age of jack hammers and typewriters as musical instruments, it is becoming clear that any sound can be either musical or unmusical, depending on the context in which the sound is heard. If a sound is intended to be music, or is heard as music, it can be considered to be music. If a sound is heard as noise, (for example: the annoying sound of a neighbour’s radio), the sound is not music to the listener, although it could still be music to someone else. Put a different way, as Alten (2011) explains in his book

Recording and producing audio for media: "All sound - speech, sound effects, and music - is made up of the same basic elements" (p 358). It is the use to which the elements of sounds are put, that makes the difference between music and other forms of sound. This paper therefore proposes creating two separate lists: one being the elements of sound (i.e. elements that are common to all sounds), and the other being the elements of music (i.e. elements that are exclusive to music). The elements of music will be examined first.

Any list of the elements of music must hold true for a symphony, a snare drum solo and John Cage's " 4' 33" ". In creating a list of the elements of music, one very important point that seems to have been overlooked in all previous lists, is that Music is an art form, and any list of elements should reflect this undeniable fact. As an art form, one of the elements must refer to: "the disposition or modification of things by human skill to answer the purpose intended" (Webster 1941). In this context, the art of the music composer is to affect the listener by structuring or organising sound. The art of the (music) listener is to be affected by the organisation of sound; whether this organization is intentional, as when listening to a symphony, or environmental, such as when listening to the gentle, rhythmic tinkle of wind chimes or John Cage's 4' 33". This paper proposes that the elements essential for music are: 1) Artistic intent or effect, 2) sound and 3) structure.

In order to identify the perceptual elements of sound, the absolute elements of sound (pressure and time) must now be divided into elements relating to cognitive perception and analysis. A limiting factor when choosing words to represent this second level of elements is that words relating to the lower level (ie pressure and time) cannot be used to describe elements in the level above this absolute level. It is for this reason that the word "duration", which is almost synonymous with "time", is not used in most recent lists of elements of music. As every element of sound must have duration to exist, duration cannot be considered a perceptual element of sound. The word “rhythm”, which relates to changes in duration, is the most widely accepted term for the time based element of sound (acara.edu.au, 2013; Musiced.nafme.org, 2013). Similarly “loudness”, which is strongly related to pressure (being a perceptual measurement of pressure changes over time (Goldstein & Kramer, 1960)), is also not used in most recent lists of elements of music. The word “dynamics”, which relates to changes in loudness, is the most widely accepted term for the loudness based element of sound.

Pitch, the most widely accepted element of all, can be considered a discrete element because it represents how the mind perceives the cyclic repetitiveness aspect of sound (De Cheveigne, 2005). Timbre, also cited in almost every list of elements, represents how the mind perceives the harmonic relationship and relative loudness of the vibrations of a single sound source, as well as the way the sound changes over time (Matthews, 1999). Texture seems to be an essential addition to the list of elements, as the source of every sound heard is automatically located and identified through a number of cognitive processes.

An element that has only recently been suggested is “spatial location”. Spatial location involves the cognitive placement of a sound in an environmental context, and includes placement of a sound on a horizontal and a vertical plane as well as placement in a reverberant environment (Levitin, 1999). Spatial location is considered so important that two processing areas are used exclusively to identify the horizontal location of a sound (Corwin, 2009) and much cognitive effort is expended processing the different arrival times of sonic reflections, in order to make sense of a sound’s acoustic environment (Levitin, 1999). Based on the amount of processing power the brain spends on this aspect of sound, “spatial location” seems a vital element to be included in any list of perceptual elements of sound.

In relation to other potential elements, the British inclusion of "harmony" seems unnecessary, because harmony can be considered one of the many possible textures of sound: one that involves two or more simultaneous pitches. The Australian inclusion of "expression" as an adjunct to dynamics seems inappropriate from a number of aspects. Firstly, if the use of the term relates to expression markings, then the placement of the term as an adjunct to dynamics is not appropriate. Not only do expression markings involve dynamics (eg: crescendo, decrescendo, tremolo etc.), they involve pitch (vibrato, grace notes, trills etc) and timbre (eg "dolce", "brightly" etc.) as well. Secondly, it seems that expression is really the whole purpose of music making. Rather than being an element of music, expression is really the end product of the music making process itself.

In conclusion, in order to provide a pedagogically sound list of elements of music, this paper proposes that the elements of music and the elements of sound be considered separately and that the perceptual elements of sound are: pitch, timbre, dynamics, rhythm, texture and spatial location and the elements of music are: artistic intent or effect, sound and structure.

Table 1

The elements of sound and music

The elements of sound:
- Pitch
- Dynamics
- Rhythm
- Texture
- Timbre
- Spatial location

The elements of music:
- Artistic intent or effect
- Sound
- Structure

References

acara.edu.au. (2013). Australian Curriculum: The Arts. Foundation to Year 10. Retrieved from http://www.acara.edu.au/verve/_resources/DRAFT_Australian_Curriculum_The_Arts_Foundation_to_Year_10_July_2012.pdf

Clementi, M. (1974). *Notes on. Introduction to the art of playing on the piano forte: containing the elements of music, preliminary notions on fingering, and fifty fingered lessons*: Da Capo Pr.

Corwin, J. (2009). The auditory system. Handout (sourced from the internet). Jcorwin@virginia.edu

De Cheveigne, A. (2005). Pitch perception models. In *Pitch* (pp. 169-233): Springer.

Education.gov.uk. (2011). Music - Schools. Retrieved from http://www.education.gov.uk/schools/teachingandlearning/curriculum/primary/b00199150/music

Estrella, E. (2015). The Elements of music. Retrieved from http://musiced.about.com/od/beginnerstheory/a/musicelements.htm

Goldstein, R., & Kramer, J. C. (1960). Factors affecting thresholds for short tones. *Journal of Speech, Language and Hearing Research, 3*(3), 249.

Levitin, D. J. (1999). Memory for musical attributes. In P. R. Cook (Ed.), *Music, cognition, and computerized sound: An introduction to psychoacoustsics* (pp. 105 - 127). Cambridge, Massachusetts: The MIT press.

Matthews, M. (1999). Introduction to timbre. In P. R. Cook (Ed.), *Music, cognition, and computerized sound: An introduction to psychoacoustsic* (pp. 79-88). Cambridge, Massachusetts: The MIT press.

Musiced.nafme.org. (2013). NAfME – Music Education – The School Music Program: A New Vision. Retrieved from http://musiced.nafme.org/resources/the-school-music-program-a-new-vision/

Niecks, F. (1884). A Concise Dictionary of Musical Terms. To Which Is Prefixed an Introduction to the Elements of Music by Frederick Niecks. *The Musical Times and Singing Class Circular, 25*(498), 473. doi:10.2307/3357513

Seashore, C. E. (1938). *Psychology of music*: New York: Dover Publications.

Thomas, R. B. (1970). *MMCP synthesis : a structure for music education*. Bardonia, N.Y: Bardonia, N.Y : Media Materials.

Webster, N. (Ed.) (1947) Webster's New Twentieth Century Dictionary. Clevelend Ohio: The World Publishing Company.

Appendix 2 Duration Table for Sine Wave Stimuli Creation

												10 wave	9 waves	8 waves	7 waves	6 wave	5 waves	4 waves
		200ms	100ms	70ms	50ms	40ms	35ms	30ms	25ms	20ms	15ms	10ms	9ms	8ms	7ms	6ms	5ms	4ms
C4	(261.63 Hz)	200	100	70	50	40	38.2	38.2	38.2	38.2	38.2	38.2	34.4	30.6	26.8	22.9	19.1	15.3
C#4	(277.18 Hz)	200	100	70	50	40	36.1	36.1	36.1	36.1	36.1	36.1	32.5	28.9	25.3	21.6	18.0	14.4
D4	(293.66 Hz)	200	100	70	50	40	35	34.1	34.1	34.1	34.1	34.1	30.6	27.2	23.8	20.4	17.0	13.6
D#4	(311.13 Hz)	200	100	70	50	40	35	32.1	32.1	32.1	32.1	32.1	28.9	25.7	22.5	19.3	16.1	12.9
E4	(329.63 Hz)	200	100	70	50	40	35	30.3	30.3	30.3	30.3	30.3	27.3	24.3	21.2	18.2	15.2	12.1
F4	(349.23 Hz)	200	100	70	50	40	35	30	28.6	28.6	28.6	28.6	25.8	22.9	20.0	17.2	14.3	11.5
F#4	(369.99 Hz)	200	100	70	50	40	35	30	27.0	27.0	27.0	27.0	24.3	21.6	18.9	16.2	13.5	10.8
G4	(392 Hz)	200	100	70	50	40	35	30	25.5	25.5	25.5	25.5	23.0	20.4	17.9	15.3	12.8	10.2
G#4	(415.3 Hz)	200	100	70	50	40	35	30	25	24.1	24.1	24.1	21.7	19.3	16.9	14.4	12.0	9.6
A4	(440 Hz)	200	100	70	50	40	35	30	25	22.7	22.7	22.7	20.5	18.2	15.9	13.6	11.4	9.1
A#4	(466.16 Hz)	200	100	70	50	40	35	30	25	21.5	21.5	21.5	19.3	17.2	15.0	12.9	10.7	8.6
B4	(493.88 Hz)	200	100	70	50	40	35	30	25	20.2	20.2	20.2	18.2	16.2	14.2	12.1	10.1	8.1
C5	(523.25 Hz)	200	100	70	50	40	35	30	25	20	19.1	19.1	17.2	15.3	13.4	11.5	9.6	7.6
C#5	(554.37 Hz)	200	100	70	50	40	35	30	25	20	18.0	18.0	16.2	14.4	12.6	10.8	9.0	7.2
D5	(587.33 Hz)	200	100	70	50	40	35	30	25	20	17.0	17.0	15.3	13.6	11.9	10.2	8.5	6.8
D#5	(622.25 Hz)	200	100	70	50	40	35	30	25	20	16.1	16.1	14.5	12.9	11.2	9.6	8.0	6.4
E5	(659.26 Hz)	200	100	70	50	40	35	30	25	20	15.2	15.2	13.7	12.1	10.6	9.1	7.6	6.1
F5	(698.46 Hz)	200	100	70	50	40	35	30	25	20	15	14.3	12.9	11.5	10.0	8.6	7.2	5.7
F#5	(739.99 Hz)	200	100	70	50	40	35	30	25	20	15	13.5	12.2	10.8	9.5	8.1	6.8	5.4
G5	(783.99 Hz)	200	100	70	50	40	35	30	25	20	15	12.8	11.5	10.2	8.9	7.7	6.4	5.1
G#5	(830.61 Hz)	200	100	70	50	40	35	30	25	20	15	12.0	10.8	9.6	8.4	7.2	6.0	4.8
A5	(880 Hz)	200	100	70	50	40	35	30	25	20	15	11.4	10.2	9.1	8.0	6.8	5.7	4.5
A#5	(932.33 Hz)	200	100	70	50	40	35	30	25	20	15	10.7	9.7	8.6	7.5	6.4	5.4	4.3
B5	(987.77 Hz)	200	100	70	50	40	35	30	25	20	15	10.1	9.1	8.1	7.1	6.1	5.1	4.0
C6	(1046.5 Hz)	200	100	70	50	40	35	30	25	20	15	9.6	8.6	7.6	6.7	5.7	4.8	3.8

Note: All numbers are in milliseconds. The transition area from set tone durations to set wave cycles is highlighted in yellow. B5 equivalent is highlighted in green.

Appendix 3 Programs Written for Study 1 and Study 2

Study 1

Main Program

```
 <!DOCTYPE html>
<html>
<?php

 if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}
print $idin;
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testdb";
$mysqli = new mysqli($servername, $username, $password, $database);
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}
$sql="SELECT * FROM participant WHERE participantNo=$idin";
if ($result=mysqli_query($mysqli,$sql))
  {
  // Fetch one and one row
  while ($row=mysqli_fetch_row($result))
    {
       $recordDur= $row[23];
    }
       mysqli_free_result($result);
    }
print $recordDur;
?>
<head>
    <meta charset= "utf-8">
    <title>Identify Frequency Change Direction (from same note)</title>
    <link rel="stylesheet" href="Study2CSS.css">
<style>
</style>
</head>
<body>
<div id= "holder">
<h1 id = "heading"><u>Up or Down Pitch Direction Training</u></h1>
<button id = "home" onclick="gohome()"> Back to Home Page</button>
<img src= "upButton.png" id= "up" alt= "up button" onclick= "upPressed()"></img>
<img src= "downButton.png" id= "down" alt= "down button" onclick=
"downPressed()"></img>
<img src="Blank.png" alt="Wrong" id = "wrong"></img>
<button id = "starter" onclick="stimPres()"> When ready, click once to start</button>
```

```
<button id = "harder" onclick="hardLevel()"> Go to Hard level</button>
<p id = "ID">your ID should appear here</p>
<div id= "levelDiv">
<p id = "Level">300 ms</p>
</div>
<h2 id= "previousBest"> Previous best</h2>
<p id = "Score">
 Remaining Tries: 40
</p>
<div id= "explainer" >
<h2 >Instructions: Up Down Pitch Training    (hint: if you don't know, just guess)</h2>
<p>When you press the blue "When ready, click once to start" button (above), you will hear
two sine wave tones: a 1000 Hz tone and a 2000 Hz tone. These two tones make the interval
of an octave. Depending upon the order of presentation, the sequence will either sound like it
goes ‘up’ (low to high) or ‘down’ (high to low). You need to press the button with an arrow
indicating the direction of frequency change (up or down). If you are correct, a tick with a
green background will flash on the screen for ½ second. If your selection was wrong, a cross
with a red background will flash on the screen for ½ second. As you improve your skill, the
task will become more difficult by decreasing the duration of the two tones. This method of
improving skills is called temporal occlusion training. If it becomes too difficult, making
mistakes will make it easier again.</p>
<p>Try to get below 10ms in the 40 'tries' before the exercise stops (click the 'refresh' button
to try again).
<h2 >Rationale: Up Down Pitch Training </h2>

</div>
</div>
<div id= "ending">
<p id= "congrats"> Congratulations<p>
<button id = "tryAgain" class= "endings" onclick="againOrContinue()"> Try Again</button>
<button id = "hardLevel" class= "endings" onclick="window.location.href =
'../Study2Var/StudyD200.php?idin=' + idin">Hard Level</button>
<button id = "Leave" class= "endings" onclick="window.location.href =
'../../HomePage.php?idin=' + idin">Back to Home Page</button>
</div>
<script src= "Study2Java.js"></script>
<script src="jquery-3.0.0.js"></script>
<script>
var idin = "<?php echo $idin; ?>";
var lowLevel= "<?php echo $recordDur; ?>";
document.getElementById('previousBest').innerText= "Your previous best: "+ lowLevel +
"ms";
document.getElementById('ID').innerText = "Your ID is: " + idin;
</script>
</body>
</html>
```

Program Extra CSS (cascading style sheet)

```
#wrong {position: absolute;
```

```
    top: 55%;
  left: 45%;
  -ms-transform: translate(-50%, -50%);
  transform: translate(-50%, -50%);
     width: 350px;
    height:350px;
    z-index: 0;
    background-color: rgb(200,200,200);
    display: none;
}
#Score {position: relative;
    top: -220px;
    left: 30px;
    font-size:300%;
    z-index: 5;
    visibility: hidden;
}
#heading {position: relative;
    top: 10px;
    text-align: center;
    font-size: 2em;
    z-index: 5;}
#up {
    width: 250px;
    height: 150px;
    position: relative;
    top: 90px;
    left: 500px;
    z-index: 7;}

#down {
    width:250px;
    height: 150px;
    position: relative;
    top: 250px;
    left: 250px;
    z-index: 7;}
#ID {position: relative;
     left: 50px;
     top: -405px;
    z-index: 5;}

#Level {position: relative;
        z-index: 5;
        top:  0px;
        left:  140px;
        font-size:  500%
    }
```

```
  #levelDiv {
  position: relative;
  top:  -360px;
     left:  150px;
     width: 500px;
  background-color: rgb(200,200,200);
  z-index: 6;
}
#holder {
  background-color: white;
  position: relative;
  width: 1000px;
  height: 640px;
  margin: auto;
  border-width: 2px;
  z-index: 3;
}
#home {
  background: rgb(200,50,50);
  width:200px;
  height: 80px;
  font-size:200%;
  position: relative;
  top: -150px;
  left:770px;
  z-index: 6;
}
#starter {
  background: rgb(10,203,238);
  width:500px;
  height: 50px;
  font-size:200%;
  position: relative;
  top: 220px;
  left:150px;
  z-index: 5;

}
#harder {
  background: red;
  width:250px;
  height: 100px;
  font-size:200%;
  position: relative;
  top: 220px;
  left:200px;
  z-index: 6;
```

```
}
#nextOne {
   display: none;
   background: rgb(10,203,238);
   width:500px;
   height: 50px;
   font-size:200%;
   position: relative;
   top: 10px;
   left:40px;
   z-index: 5;
}
#explainer {background-image: url('herringbone.png');
   padding: 20px;
   position: relative;
   top: -250px;
}
p{
   font-size: 130%;
   margin-left: 10px;
   margin-right: 10px;

}
#previousBest{
   position: relative;
   top: -440px;
   left: 20px;
}
#ending{
   visibility: hidden;
   position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  width: 850px;
  height: 500px;
  background: rgb(209, 105, 105);
  padding: 20px;
  z-index: 7;
}
.endings{
   font-size: 120%;
}
#congrats {
   font-size: 40px;
   position: relative;
   top: 10px;
   left: 20px;
```

```
    z-index: 5;
    margin: 10px;
}
#tryAgain {
    position: relative;
    width: 250px;
    height: 100px;
    left: 20px;
    top: 50px;
    border:2 px rgb(250, 37, 55);
    background-color: lightgreen;
    border-radius: 15px;
  z-index: 7;
}
#hardLevel {
    position: relative;
    width: 250px;
    height: 100px;
    left: 40px;
    top: 50px;
    border:2 px rgb(240, 243, 42);
    background-color: rgb(211, 224, 26);
    border-radius: 15px;
  z-index: 7;
}
#Leave{
    position: relative;
    width: 250px;
    height: 100px;
    left:60px;
    top: 50px;
    border:2 px rgb(250, 37, 55);
    background-color: rgb(248, 150, 158);
    border-radius: 15px;
  z-index: 7;
}
body {background-color: lightskyblue;}
```

Extra Javascript

```
var duration= 17;
var lowDuration= 17;
var direction= 0;
var choice = 0;
var score= 0;
var goes= 40;
var TotScore=0;
var correct= 0;
var incorrect= 0;
var answ= 0;
```

```
var elapsedTime=0;
var i = 0;
var startTime= 0;
var changePic = document.getElementById("wrong");
var currentLevel= document.getElementById('Level');
var currLevel= "300ms";
var endChoice= document.getElementById("congrats");
var lowestDuration= 0;
var threeHundredUp = new Audio("study3/330001.wav");
function threeHUp() {threeHundredUp.play();}
var threeHundredDown = new Audio("study3/330002.wav");
function threeHDn() {threeHundredDown.play();}
var oneFiftyUp = new Audio("study3/315001.wav");
function oneFiveUp() {oneFiftyUp.play();}
var oneFiftyDown = new Audio("study3/315002.wav");
function oneFiveDn() {oneFiftyDown.play();}
var eightyUp = new Audio("study3/308001.wav");
function eightOUp () {eightyUp.play();}
var eightyDown = new Audio("study3/308002.wav");
function eightODn() {eightyDown.play();}
var fourtyUp = new Audio("study3/304001.wav");
function fourtOUp() {fourtyUp.play();}
var fourtyDown = new Audio("study3/304002.wav");
function fourtODn() {fourtyDown.play();}
var twentyUp = new Audio("study3/302001.wav");
function twentOUp() {twentyUp.play();}
var twentyDown = new Audio("study3/302002.wav");
function twentODn() {twentyDown.play();}
var fifteenUp = new Audio("study3/301501.wav");
function fiftUp() {fifteenUp.play();}
var fifteenDown = new Audio("study3/301502.wav");
function fiftDn() {fifteenDown.play();}
var twelveUp = new Audio("study3/301201.wav");
function twelvUp() {twelveUp.play();}
var twelveDown = new Audio("study3/301502.wav");
function twelvDn() {twelveDown.play();}
var teneUp = new Audio("study3/301001.wav");
function tenUp() {teneUp.play();}
var teneDown = new Audio("study3/301002.wav");
function tenDn() {teneDown.play();}
var nineUp = new Audio("study3/300901.wav");
function ninUp() {nineUp.play();}
var nineDown = new Audio("study3/300902.wav");
function ninDn() {nineDown.play();}
var eighteUp = new Audio("study3/300801.wav");
function eightUp() {eighteUp.play();}
var eighteDown = new Audio("study3/300802.wav");
function eightDn() {eighteDown.play();}
```

```
var seveneUp = new Audio("study3/300701.wav");
function sevenUp() {seveneUp.play();}
var seveneDown = new Audio("study3/300702.wav");
function sevenDn() {seveneDown.play();}
var sixeUp = new Audio("study3/300601.wav");
function sixUp() {sixeUp.play();}
var sixeDown = new Audio("study3/300602.wav");
function sixDn() {sixeDown.play();}
var fiveUp = new Audio("study3/300501.wav");
function fivUp() {fiveUp.play();}
var fiveDown = new Audio("study3/300502.wav");
function fivDn() {fiveDown.play();}
var foureUp = new Audio("study3/300401.wav");
function fourUp() {foureUp.play();}
var foureDown = new Audio("study3/300402.wav");
function fourDn() {foureDown.play();}
var threeUp = new Audio("study3/300301.wav");
function threUp() {threeUp.play();}
var threeDown = new Audio("study3/300302.wav");
function threDn() {threeDown.play();}
var twoeUp = new Audio("study3/300201.wav");
function twoUp() {twoeUp.play();}
var twoeDown = new Audio("study3/300202.wav");
function twoDn() {twoeDown.play();}
var oneUp = new Audio("study3/300101.wav");
function onUp() {oneUp.play();}
var oneDown = new Audio("study3/300102.wav");
function onDn() {oneDown.play();}
var currentDuration= 300;
switch (duration){
   case 17: currentDuration= 300; break;
   case 16: currentDuration= 150; break;
   case 15: currentDuration= 80; break;
   case 14: currentDuration= 40; break;
   case 13: currentDuration= 20; break;
   case 12: currentDuration= 15; break;
   case 11: currentDuration= 12; break;
   case 10: currentDuration= 10; break;
   case 9: currentDuration= 9; break;
   case 8: currentDuration= 8; break;
   case 7: currentDuration= 7; break;
   case 6: currentDuration= 6; break;
   case 5: currentDuration= 5; break;
   case 4: currentDuration= 4; break;
   case 3: currentDuration= 3; break;
   case 2: currentDuration= 2; break;
   case 1: currentDuration= 1; break;
}
```

```
function stimPres(){
if (score==0){document.getElementById('starter').style.visibility= 'hidden'};
startTime= new Date();
randnum = Math.random();
if(duration== 17){
if (randnum < 0.5){threeHUp(); direction= 1;}
else {threeHDn(); direction= 0;}
currLevel=  "300 ms";
currentLevel.innerText= currLevel;}
if(duration== 16){
if (randnum < 0.5){oneFiveUp(); direction= 1;}
else {oneFiveDn(); direction= 0;}
currLevel=  "150 ms";
currentLevel.innerText= currLevel}

if(duration== 15){
if (randnum < 0.5){eightOUp(); direction= 1; }
else {eightODn(); direction= 0;}
currLevel=  "80 ms";
currentLevel.innerText= currLevel}
if(duration== 14){
if (randnum < 0.5){fourtOUp(); direction= 1;}
else {fourtODn(); direction= 0;}
currLevel=  "40 ms";
currentLevel.innerText= currLevel}
if(duration== 13){
if (randnum < 0.5){twentOUp(); direction= 1;}
else {twentODn(); direction= 0;}
currLevel=  "20 ms";
currentLevel.innerText= currLevel}
if(duration== 12){
if (randnum < 0.5){fiftUp(); direction= 1;}
else {fiftDn(); direction= 0;}
currLevel=  "15 ms";
currentLevel.innerText= currLevel}
if(duration== 11){
if (randnum < 0.5){twelvUp(); direction= 1;}
else {twelvDn(); direction= 0;}
currLevel=  "12 ms";
currentLevel.innerText= currLevel}
if(duration== 10){
if (randnum < 0.5){tenUp(); direction= 1;}
else {tenDn(); direction= 0;}
currLevel=  "10 ms";
currentLevel.innerText= currLevel}
if(duration== 9){
if (randnum < 0.5){ninUp(); direction= 1;}
else {ninDn(); direction= 0;}
```

```
currLevel=  "9 ms";
currentLevel.innerText= currLevel}
if(duration== 8){
if (randnum < 0.5){eightUp(); direction= 1;}
else {eightDn(); direction= 0;}
currLevel=  "8 ms";
currentLevel.innerText= currLevel}
if(duration== 7){
if (randnum < 0.5){sevenUp(); direction= 1;}
else {sevenDn(); direction= 0;}
currLevel=  "7 ms";
currentLevel.innerText= currLevel}
if(duration== 6){
if (randnum < 0.5){sixUp(); direction= 1;}
else {sixDn(); direction= 0;}
currLevel=  "6 ms";
currentLevel.innerText= currLevel}
if(duration== 5){
if (randnum < 0.5){fivUp(); direction= 1;}
else {fivDn(); direction= 0;}
currLevel=  "5 ms";
currentLevel.innerText= currLevel}
if(duration== 4){
if (randnum < 0.5){fourUp(); direction= 1;}
else {fourDn(); direction= 0;}
currLevel=  "4 ms";
currentLevel.innerText= currLevel}
if(duration== 3){
if (randnum < 0.5){threUp(); direction= 1;}
else {threDn(); direction= 0;}
currLevel=  "3 ms";
currentLevel.innerText= currLevel}
if(duration== 2){
if (randnum < 0.5){twoUp(); direction= 1;}
else {twoDn(); direction= 0;}
currLevel=  "2 ms";
currentLevel.innerText= currLevel}
if(duration== 1){
if (randnum < 0.5){onUp(); direction= 1;}
else {onDn(); direction= 0;}
currLevel=  "1 ms";
currentLevel.innerText= currLevel}
}
var upButton= document.getElementById('up');
var downButton= document.getElementById('down');
function upPressed(){
upButton.src="upPressed.png";
choice= 1;
```

```
check();}
function downPressed(){
downButton.src="downPressed.png";
choice= 0;
check();}
function check(){
   answ= 0;
   if (direction == choice){
      score++;
      correct++;
      changePic.src = "Correct.png";
      changePic.style.display = "block"
      answ= 1;
      if (score>1){duration--; score= 0; if (duration < 1) {duration++;}}
      goes--;
      document.getElementById('Score').innerText= "Remaining Tries " + goes;
   }
   if (direction != choice) {
      duration++;
      incorrect++;
      score= 0;
      changePic.src = "Wrong.png";
      changePic.style.display = "block";
      if (duration > 17) {duration--;}
      goes--;
      document.getElementById('Score').innerText= "Remaining Tries: " + goes;
      }

   if (duration<lowDuration){lowDuration=duration; }
   if (goes< 1){
      nextStudy(); return;
   }
   document.getElementById('Score').style= "visibility: visible";
   setTimeout(pictureOff,500);
}
function pictureOff(){
   changePic.style.display = "none"
   setTimeout(stimPres,500);
   upButton.src= "upButton.png"
   downButton.src= "downButton.png"
}
function gohome(){
   if (lowDuration<17){
      var go= confirm("Do you really want to leave before the test is complete?          Click
'OK' to leave the test and return to the home page, or'cancel' to continue the Up Down
activity")
         if (go== true){
      switch (lowDuration){
```

```
            case 17: lowestDuration= 300; break;
            case 16: lowestDuration= 150; break;
            case 15: lowestDuration= 80; break;
            case 14: lowestDuration= 40; break;
            case 13: lowestDuration= 20; break;
            case 12: lowestDuration= 15; break;
            case 11: lowestDuration= 12; break;
            case 10: lowestDuration= 10; break;
            case 9: lowestDuration= 9; break;
            case 8: lowestDuration= 8; break;
            case 7: lowestDuration= 7; break;
            case 6: lowestDuration= 6; break;
            case 5: lowestDuration= 5; break;
            case 4: lowestDuration= 4; break;
            case 3: lowestDuration= 3; break;
            case 2: lowestDuration= 2; break;
            case 1: lowestDuration= 1; break;
            }

            $.ajax({
                type: "POST",
                url: 'topScore1.php',
                data: {'idin' : idin, 'lowestLevel' : lowestDuration, 'prevLevel' : lowLevel},
                });
            window.location.href = "../../HomePage.php?idin=" + idin;
    }}else {window.location.href = "../../HomePage.php?idin=" + idin}
}
var continueCurrent=0;
var tryAgainT = document.getElementById("tryAgain")
function hardLevel(){
    if (lowDuration<17){
        continueCurrent=1;
            finalScore();
    }else {window.location.href = "../Study2Var/StudyD200.php?idin=" + idin}
}
function finalScore(){
    switch (lowDuration){
    case 17: lowestDuration= 300; break;
    case 16: lowestDuration= 150; break;
    case 15: lowestDuration= 80; break;
    case 14: lowestDuration= 40; break;
    case 13: lowestDuration= 20; break;
    case 12: lowestDuration= 15; break;
    case 11: lowestDuration= 12; break;
    case 10: lowestDuration= 10; break;
    case 9: lowestDuration= 9; break;
    case 8: lowestDuration= 8; break;
    case 7: lowestDuration= 7; break;
```

```
    case 6: lowestDuration= 6; break;
    case 5: lowestDuration= 5; break;
    case 4: lowestDuration= 4; break;
    case 3: lowestDuration= 3; break;
    case 2: lowestDuration= 2; break;
    case 1: lowestDuration= 1; break;
    }
    $.ajax({
        type: "POST",
        url: 'topScore1.php',
        data: {'idin' : idin, 'lowestLevel' : lowestDuration, 'prevLevel' : lowLevel},
        });
    document.getElementById("ending").style= "visibility: visible;";
    endChoice.innerText = "You were able to reach \n\n" + lowestDuration + "ms.  \n\nSelect
from the buttons below:";
    endChoice.style= "font-size: 40px;";
    if (continueCurrent==1){
        tryAgainT.innerText= "Continue";
    }
}

function againOrContinue(){
    if (continueCurrent=1){
    document.getElementById("ending").style= "visibility: hidden;"
    continueCurrent=0;
    tryAgainT.innerText= "Try Again";
    stimPres();
    }else{window.location.href = "Study2.php?idin=" + idin}
}
function nextStudy(){
    finalScore();
}
PHP to send to database
 <!DOCTYPE html>
<html>
<?php

 if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}
print $idin;
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testdb";
$mysqli = new mysqli($servername, $username, $password, $database);
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}
$sql="SELECT * FROM participant WHERE participantNo=$idin";
```

```
if ($result=mysqli_query($mysqli,$sql))
  {
  // Fetch one and one row
  while ($row=mysqli_fetch_row($result))
    {
       $recordDur= $row[23];
    }
       mysqli_free_result($result);
    }
print $recordDur;
?>
<head>
    <meta charset= "utf-8">
    <title>Identify Frequency Change Direction (from same note)</title>
    <link rel="stylesheet" href="Study2CSS.css">
<style>
</style>
</head>
<body>
<div id= "holder">
<h1 id = "heading"><u>Up or Down Pitch Direction Training</u></h1>
<button id = "home" onclick="gohome()"> Back to Home Page</button>
<img src= "upButton.png" id= "up" alt= "up button" onclick= "upPressed()"></img>
<img src= "downButton.png" id= "down" alt= "down button" onclick=
"downPressed()"></img>
<img src="Blank.png" alt="Wrong" id = "wrong"></img>
<button id = "starter" onclick="stimPres()"> When ready, click once to start</button>
<button id = "harder" onclick="hardLevel()"> Go to Hard level</button>
<p id = "ID">your ID should appear here</p>
<div id= "levelDiv">
<p id = "Level">300 ms</p>
</div>
<h2 id= "previousBest"> Previous best</h2>
<p id = "Score">
 Remaining Tries: 40
</p>
<div id= "explainer" >
<h2 >Instructions: Up Down Pitch Training    (hint: if you don't know, just guess)</h3>
<p>When you press the blue "When ready, click once to start" button (above), you will hear
two sine wave tones: a 1000 Hz tone and a 2000 Hz tone. These two tones make the interval
of an octave. Depending upon the order of presentation, the sequence will either sound like it
goes ‘up’ (low to high) or ‘down’ (high to low). You need to press the button with an arrow
indicating the direction of frequency change (up or down). If you are correct, a tick with a
green background will flash on the screen for ½ second. If your selection was wrong, a cross
with a red background will flash on the screen for ½ second. As you improve your skill, the
task will become more difficult by decreasing the duration of the two tones. This method of
improving skills is called temporal occlusion training. If it becomes too difficult, making
mistakes will make it easier again.</p>
```

```
<p>Try to get below 10ms in the 40 'tries' before the exercise stops (click the 'refresh' button
to try again).
<h2 >Rationale: Up Down Pitch Training </h3>
<p>Describing fast air-vibrations (i.e. sounds) as ‘high’ pitched sounds, and slow air-
vibrations as ‘low’ pitched sounds is not instinctive. It is a learnt behaviour. Calling pitched
sounds ‘High’ and ‘Low’ are not as natural as we now think. In the ancient Greek days,
common string instruments were constructed with the longer, lower pitched strings at the top
of the instrument. This led to low frequency sound to be called ‘high’ tones and vice versa.
However this ancient practice is actually less natural than our current system because there
are some physiological foundations for the current 'high/low' identification practice. ‘Low’
frequency sounds occur more often close to the ground (e.g rumblings and thuds) while high
frequency sounds (e.g. snapping twigs and bird sounds) appear high up in the environment.
Thus ‘high’ sounds occur more commonly high up in our environment and low sounds appear
low near the ground. Therefore there is some logic to the current 'high/low' identification
practice.  <br><br>
Because identifying sounds as ‘high’ and 'low' is a learnt (rather than instinctive) behaviour,
it needs to be taught. As a high school teacher, I can attest that at least one or two students in
every class struggle with this seemingly intuitive concept. I have therefore developed a
training and practice method for developing skills in the fundamental musical concept "high
and low sounds."<br><br>
</p>
</div>
</div>
<div id= "ending">
<p id= "congrats"> Congratulations<p>
<button id = "tryAgain" class= "endings" onclick="againOrContinue()"> Try Again</button>
<button id = "hardLevel" class= "endings" onclick="window.location.href =
'../Study2Var/StudyD200.php?idin=' + idin">Hard Level</button>
<button id = "Leave" class= "endings" onclick="window.location.href =
'../../HomePage.php?idin=' + idin">Back to Home Page</button>
</div>
<script src= "Study2Java.js"></script>
<script src="jquery-3.0.0.js"></script>
<script>
var idin = "<?php echo $idin; ?>";
var lowLevel= "<?php echo $recordDur; ?>";
document.getElementById('previousBest').innerText= "Your previous best: "+ lowLevel +
"ms";
document.getElementById('ID').innerText = "Your ID is: " + idin;
</script>
</body>
</html>
<!DOCTYPE html>
<html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
```

```
$database = "interval_testdb";
if (isset($_POST["idin"])) {
    $idin = intval($_POST["idin"]);

    $lowLevel = intval($_POST["lowestLevel"]);

    $prevLevel = intval($_POST["prevLevel"]);
    print $idin;
    print $lowLevel;
    print $prevLevel;
    // Create connection
    $mysqli = new mysqli($servername, $username, $password, $database);
    //MySqli Insert Query
    $insert_row = $mysqli->prepare("INSERT INTO freqdirection (Participant, Duration)
VALUES(?, ?)");
    $insert_row->bind_param("ii", $idin, $lowLevel);

    $insert_row->execute();
if($lowLevel < $prevLevel) {
        $sql = "UPDATE participant SET upDn1= '$lowLevel' WHERE participantNo= '$idin'";
        if (mysqli_query($mysqli, $sql)) {
            echo "New record created successfully";
        } else {
            echo "Error: " . $sql . "<br>" . mysqli_error($conn);
        }
    }

    $mysqli->close();

}
?>
<body>
<p id= "checker"> test this script</p>
<script>
var IDNo= "<?php echo $idin; ?>";
document.getElementById("checker").innerText= "the ID No is: " + IDNo;
</script>
</body>
</html>
<!DOCTYPE html>
<html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_GET["idin"])) {
    $idin = intval($_GET["idin"]);
```

```
    $lowLevel = intval($_GET["lowestLevel"]);

    $prevLevel = intval($_GET["prevLevel"]);
    print $idin;
    print $lowLevel;
    print $prevLevel;
    // Create connection
    $mysqli = new mysqli($servername, $username, $password, $database);
    //MySqli Insert Query
    $insert_row = $mysqli->prepare("INSERT INTO freqdirection (Participant, Duration)
VALUES(?, ?)");
    $insert_row->bind_param("ii", $idin, $lowLevel);

    $insert_row->execute();
if($lowLevel < $prevLevel) {
        $sql = "UPDATE participant SET upDn1= '$lowLevel' WHERE participantNo= '$idin'";
        if (mysqli_query($mysqli, $sql)) {
            echo "New record created successfully";
        } else {
            echo "Error: " . $sql . "<br>" . mysqli_error($conn);
        }
    }

    $mysqli->close();

}
?>
<body>
<p id= "checker"> test this script</p>
<script>
var IDNo= "<?php echo $idin; ?>";
document.getElementById("checker").innerText= "the ID No is: " + IDNo;
</script>
</body>
</html>
```

SQL code

```
-- phpMyAdmin SQL Dump
-- version 4.8.4
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Jun 18, 2021 at 04:28 PM
-- Server version: 10.1.37-MariaDB
-- PHP Version: 7.3.0
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
```

```
SET time_zone = "+00:00";
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
/*!40101 SET NAMES utf8mb4 */;
--
-- Database: `interval_auraltrain`
--
CREATE DATABASE IF NOT EXISTS `interval_auraltrain` DEFAULT CHARACTER
SET utf8mb4 COLLATE utf8mb4_unicode_ci;
USE `interval_auraltrain`;
-- --------------------------------------------------------
--
-- Table structure for table `freqdirection`
--
CREATE TABLE `freqdirection` (
  `Participant` int(11) NOT NULL,
  `Duration` int(11) NOT NULL,
  `ansone` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `answ` int(11) NOT NULL,
  `elapsedTime` int(11) NOT NULL,
  `datetime` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `participant`
--
CREATE TABLE `participant` (
  `firstName` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `surname` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `usernm` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `passwd` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `Homeclass` int(11) DEFAULT NULL,
  `training` text COLLATE utf8mb4_unicode_ci,
  `instrument` text COLLATE utf8mb4_unicode_ci,
  `participantNo` int(11) NOT NULL,
  `dateTime` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `preInt` int(11) NOT NULL,
  `preMel` int(11) NOT NULL,
  `postInt` int(11) NOT NULL,
  `postMel` int(11) NOT NULL,
  `topScore` int(11) NOT NULL DEFAULT '20',
```

```
  `single2` int(11) NOT NULL DEFAULT '1000',
  `single3` int(11) NOT NULL DEFAULT '1000',
  `single4` int(11) NOT NULL DEFAULT '1000',
  `double2` int(11) NOT NULL DEFAULT '1000',
  `double3` int(11) NOT NULL DEFAULT '1000',
  `double4` int(11) NOT NULL DEFAULT '1000',
  `mixed2` int(11) NOT NULL DEFAULT '1000',
  `mixed3` int(11) NOT NULL DEFAULT '1000',
  `mixed4` int(11) NOT NULL DEFAULT '1000',
  `upDn1` int(11) NOT NULL DEFAULT '0',
  `upDn2` int(11) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
--
-- Dumping data for table `participant`
--
INSERT INTO `participant` (`firstName`, `surname`, `usernm`, `passwd`, `Homeclass`,
`training`, `instrument`, `participantNo`, `dateTime`, `preInt`, `preMel`, `postInt`, `postMel`,
`topScore`, `single2`, `single3`, `single4`, `double2`, `double3`, `double4`, `mixed2`,
`mixed3`, `mixed4`, `upDn1`, `upDn2`) VALUES
('df', 'fd', 'rburton', '3324', 1, 'veryLittle', 'keyboard', 1, '2020-12-11 10:53:56', 0, 0, 0, 0, 1240,
1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 70, 0, 0),
('Test1', 'Test1', 'Test', 'test1', 0, '4to6years', 'woodwind', 39, '2020-12-22 12:25:34', 0, 0, 0, 0,
20, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 0, 0),
('General', 'Test', 'Test', '123', NULL, '1to3years', 'none', 40, '2020-12-24 15:37:09', 0, 0, 0, 0,
20, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 0, 0);
--
-- Indexes for dumped tables
--
--
-- Indexes for table `freqdirection`
--
ALTER TABLE `freqdirection`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `participant`
--
ALTER TABLE `participant`
  ADD PRIMARY KEY (`participantNo`);
--
-- AUTO_INCREMENT for dumped tables
--
--
-- AUTO_INCREMENT for table `freqdirection`
--
ALTER TABLE `freqdirection`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `participant`
```

```
--
ALTER TABLE `participant`
  MODIFY `participantNo` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=41;
COMMIT;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Database for all Short tone studies

```
-- phpMyAdmin SQL Dump
-- version 4.8.4
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Jun 18, 2021 at 04:47 PM
-- Server version: 10.1.37-MariaDB
-- PHP Version: 7.3.0
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
/*!40101 SET NAMES utf8mb4 */;
--
-- Database: `interval_testdb`
--
-- --------------------------------------------------------
--
-- Table structure for table `auraltrain`
--
CREATE TABLE `auraltrain` (
  `Participant` int(11) NOT NULL,
  `id` int(11) NOT NULL,
  `activityNo` int(11) NOT NULL,
  `lowDuration` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `content`
--
CREATE TABLE `content` (
  `id` int(11) NOT NULL,
  `cid` int(11) DEFAULT NULL,
```

```
  `data_file` varchar(200) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `data` longblob
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `freqdirection`
--
CREATE TABLE `freqdirection` (
  `Participant` int(11) NOT NULL,
  `Duration` int(11) NOT NULL,
  `datetime` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `ID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
--
-- Dumping data for table `freqdirection`
--
INSERT INTO `freqdirection` (`Participant`, `Duration`, `datetime`, `ID`) VALUES
(1, 80, '2021-02-08 08:58:41', 1),
(1, 1, '2021-04-21 06:34:32', 2),
(1, 1, '2021-04-21 08:17:42', 3),
(1, 1, '2021-04-21 22:15:26', 4);
-- --------------------------------------------------------
--
-- Table structure for table `freqdirectioncng`
--
CREATE TABLE `freqdirectioncng` (
  `Participant` int(11) NOT NULL,
  `Duration` int(11) NOT NULL,
  `datetime` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `ID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `notename`
--
CREATE TABLE `notename` (
  `Participant` int(11) NOT NULL,
  `duration` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `curTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
--
-- Dumping data for table `notename`
--
INSERT INTO `notename` (`Participant`, `duration`, `ID`, `curTime`) VALUES
(2, 45, 1, '2021-02-11 23:24:46'),
(2, 100, 2, '2021-02-11 23:29:13'),
(2, 100, 3, '2021-02-12 00:01:34'),
```

```
(1, 100, 4, '2021-02-15 07:10:00'),
(1, 202, 5, '2021-02-15 16:09:14');
-- --------------------------------------------------------
--
-- Table structure for table `noteshortname`
--
CREATE TABLE `noteshortname` (
  `Participant` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `section1` int(11) NOT NULL DEFAULT '10',
  `section2` int(11) NOT NULL DEFAULT '10',
  `section3` int(11) NOT NULL DEFAULT '10',
  `section4` int(11) NOT NULL DEFAULT '10',
  `section5` int(11) NOT NULL DEFAULT '10',
  `section6` int(11) NOT NULL,
  `datetime` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `participant`
--
CREATE TABLE `participant` (
  `firstName` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `surname` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `usernm` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `passwd` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `Homeclass` int(11) DEFAULT NULL,
  `training` text COLLATE utf8mb4_unicode_ci,
  `instrument` text COLLATE utf8mb4_unicode_ci,
  `participantNo` int(11) NOT NULL,
  `dateTime` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `preInt` int(11) NOT NULL DEFAULT '0',
  `preMel` int(11) NOT NULL DEFAULT '0',
  `postInt` int(11) NOT NULL DEFAULT '0',
  `postMel` int(11) NOT NULL DEFAULT '0',
  `topScore` int(11) NOT NULL DEFAULT '20',
  `single2` int(11) NOT NULL DEFAULT '1000',
  `single3` int(11) NOT NULL DEFAULT '1000',
  `single4` int(11) NOT NULL DEFAULT '1000',
  `double2` int(11) NOT NULL DEFAULT '1000',
  `double3` int(11) NOT NULL DEFAULT '1000',
  `double4` int(11) NOT NULL DEFAULT '1000',
  `mixed2` int(11) NOT NULL DEFAULT '1000',
  `mixed3` int(11) NOT NULL DEFAULT '1000',
  `mixed4` int(11) NOT NULL DEFAULT '1000',
  `upDn1` int(11) NOT NULL DEFAULT '300',
  `upDn2` int(11) NOT NULL DEFAULT '200',
  `keyboard` int(11) NOT NULL DEFAULT '200'
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
--
-- Dumping data for table `participant`
--
INSERT INTO `participant` (`firstName`, `surname`, `usernm`, `passwd`, `Homeclass`,
`training`, `instrument`, `participantNo`, `dateTime`, `preInt`, `preMel`, `postInt`, `postMel`,
`topScore`, `single2`, `single3`, `single4`, `double2`, `double3`, `double4`, `mixed2`,
`mixed3`, `mixed4`, `upDn1`, `upDn2`, `keyboard`) VALUES
('Russell', 'Burton', 'fifth', 'test', NULL, '7yearsplus', 'wind', 1, '2021-02-07 22:13:04', 0, 0, 0,
0, 20, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1, 200, 100),
('russell', 'burton', 'rburton', '3324', NULL, '7yearsplus', 'wind', 2, '2021-02-12 09:47:28', 0, 0,
0, 0, 20, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 300, 200, 45);
-- --------------------------------------------------------
--
-- Table structure for table `threefirstint`
--
CREATE TABLE `threefirstint` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `correcttwo` int(11) NOT NULL,
  `correctthree` int(11) NOT NULL,
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `threefirststrdn`
--
CREATE TABLE `threefirststrdn` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `correcttwo` int(11) NOT NULL,
  `correctthree` int(11) NOT NULL,
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-- --------------------------------------------------------
--
-- Table structure for table `threefirststrup`
--
CREATE TABLE `threefirststrup` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `correcttwo` int(11) NOT NULL,
  `correctthree` int(11) NOT NULL,
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `threesecondexhit`
--
CREATE TABLE `threesecondexhit` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `correcttwo` int(11) NOT NULL,
  `correctthree` int(11) NOT NULL,
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `threesecondext`
--
CREATE TABLE `threesecondext` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `correcttwo` int(11) NOT NULL,
```

```
  `correctthree` int(11) NOT NULL,
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `threesecondint`
--
CREATE TABLE `threesecondint` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `correcttwo` int(11) NOT NULL,
  `correctthree` int(11) NOT NULL,
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
--
-- Dumping data for table `threesecondint`
--
INSERT INTO `threesecondint` (`Participant`, `datetime`, `duration`, `participantAns`,
`corrAns`, `correctone`, `correcttwo`, `correctthree`, `Correct`, `eltime`, `ID`,
`AveMinScore`) VALUES
(0, '2021-05-11 01:54:37', 200, 2, 2, 25, 18, 13, 1, 8, 1, NULL),
(0, '2021-05-11 01:54:42', 200, 2, 2, 4, 13, 17, 1, 3, 2, NULL),
(0, '2021-05-11 01:54:50', 100, 0, 0, 4, 13, 19, 1, 6, 3, NULL),
(0, '2021-05-11 01:55:00', 100, 2, 2, 19, 13, 4, 1, 6, 4, NULL),
(0, '2021-05-11 01:55:12', 70, 0, 1, 14, 7, 3, 0, 7, 5, NULL),
(0, '2021-05-11 01:55:20', 100, 1, 1, 4, 11, 16, 1, 4, 6, NULL),
(0, '2021-05-11 01:55:26', 100, 1, 1, 22, 13, 5, 1, 3, 7, NULL),
(0, '2021-05-11 01:55:32', 70, 2, 2, 3, 7, 14, 1, 5, 8, NULL),
(0, '2021-05-11 01:55:38', 70, 0, 2, 13, 6, 2, 0, 3, 9, NULL),
(0, '2021-05-11 01:55:44', 100, 0, 0, 10, 18, 22, 1, 4, 10, NULL),
(0, '2021-05-11 01:55:48', 100, 1, 1, 4, 13, 19, 1, 3, 11, NULL),
(0, '2021-05-11 01:55:54', 70, 0, 2, 3, 7, 14, 0, 5, 12, NULL),
(0, '2021-05-11 01:56:19', 100, 2, 2, 4, 11, 16, 1, 5, 13, NULL),
(0, '2021-05-11 01:56:23', 100, 2, 2, 10, 18, 22, 1, 2, 14, NULL),
(0, '2021-05-11 01:56:27', 70, 0, 0, 13, 6, 2, 1, 2, 15, NULL),
(0, '2021-05-11 01:56:30', 70, 0, 1, 12, 16, 25, 0, 2, 16, NULL),
(0, '2021-05-11 01:56:38', 100, 0, 2, 19, 13, 4, 0, 5, 17, NULL);
-- --------------------------------------------------------
```

```
--
-- Table structure for table `threethirdint`
--
CREATE TABLE `threethirdint` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `correcttwo` int(11) NOT NULL,
  `correctthree` int(11) NOT NULL,
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `threethirdstrdn`
--
CREATE TABLE `threethirdstrdn` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `correcttwo` int(11) NOT NULL,
  `correctthree` int(11) NOT NULL,
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `threethirdstrup`
--
CREATE TABLE `threethirdstrup` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `correcttwo` int(11) NOT NULL,
  `correctthree` int(11) NOT NULL,
```

```
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `tonedeaftest`
--
CREATE TABLE `tonedeaftest` (
  `Participant` int(11) NOT NULL,
  `Duration` int(11) NOT NULL,
  `datetime` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `twofirstdn`
--
CREATE TABLE `twofirstdn` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `correcttwo` int(11) NOT NULL,
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `twofirstup`
--
CREATE TABLE `twofirstup` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `correcttwo` int(11) NOT NULL,
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
```

```
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `twosecond`
--
CREATE TABLE `twosecond` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `correcttwo` int(11) NOT NULL,
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `twosecondup`
--
CREATE TABLE `twosecondup` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `correcttwo` int(11) NOT NULL,
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- --------------------------------------------------------
--
-- Table structure for table `visits`
--
CREATE TABLE `visits` (
  `Participant` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `dateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci
COMMENT='Number of visitors';
--
-- Dumping data for table `visits`
```

```
--
INSERT INTO `visits` (`Participant`, `ID`, `dateTime`) VALUES
(1, 1, '2021-02-15 22:57:26'),
(2, 2, '2021-02-15 22:57:50'),
(1, 3, '2021-02-15 23:03:35'),
(1, 4, '2021-02-15 23:17:23');
-- --------------------------------------------------------
--
-- Table structure for table `zonenote`
--
CREATE TABLE `zonenote` (
  `Participant` int(11) NOT NULL,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `duration` int(11) NOT NULL,
  `participantAns` int(11) NOT NULL,
  `corrAns` int(11) NOT NULL,
  `correctone` int(11) NOT NULL,
  `Correct` tinyint(4) NOT NULL,
  `eltime` int(11) NOT NULL,
  `ID` int(11) NOT NULL,
  `AveMinScore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
--
-- Indexes for dumped tables
--
--
-- Indexes for table `auraltrain`
--
ALTER TABLE `auraltrain`
  ADD PRIMARY KEY (`id`);
--
-- Indexes for table `content`
--
ALTER TABLE `content`
  ADD PRIMARY KEY (`id`);
--
-- Indexes for table `freqdirection`
--
ALTER TABLE `freqdirection`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `freqdirectioncng`
--
ALTER TABLE `freqdirectioncng`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `notename`
--
```

```
ALTER TABLE `notename`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `noteshortname`
--
ALTER TABLE `noteshortname`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `participant`
--
ALTER TABLE `participant`
  ADD PRIMARY KEY (`participantNo`);
--
-- Indexes for table `threefirstint`
--
ALTER TABLE `threefirstint`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `threefirststrdn`
--
ALTER TABLE `threefirststrdn`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `threefirststrup`
--
ALTER TABLE `threefirststrup`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `threesecondexhit`
--
ALTER TABLE `threesecondexhit`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `threesecondext`
--
ALTER TABLE `threesecondext`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `threesecondint`
--
ALTER TABLE `threesecondint`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `threethirdint`
--
ALTER TABLE `threethirdint`
  ADD PRIMARY KEY (`ID`);
--
```

```
-- Indexes for table `threethirdstrdn`
--
ALTER TABLE `threethirdstrdn`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `threethirdstrup`
--
ALTER TABLE `threethirdstrup`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `tonedeaftest`
--
ALTER TABLE `tonedeaftest`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `twofirstdn`
--
ALTER TABLE `twofirstdn`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `twofirstup`
--
ALTER TABLE `twofirstup`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `twosecond`
--
ALTER TABLE `twosecond`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `twosecondup`
--
ALTER TABLE `twosecondup`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `visits`
--
ALTER TABLE `visits`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `zonenote`
--
ALTER TABLE `zonenote`
  ADD PRIMARY KEY (`ID`);
--
-- AUTO_INCREMENT for dumped tables
--
--
```

```
-- AUTO_INCREMENT for table `auraltrain`
--
ALTER TABLE `auraltrain`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `content`
--
ALTER TABLE `content`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `freqdirection`
--
ALTER TABLE `freqdirection`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;
--
-- AUTO_INCREMENT for table `freqdirectioncng`
--
ALTER TABLE `freqdirectioncng`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `notename`
--
ALTER TABLE `notename`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;
--
-- AUTO_INCREMENT for table `noteshortname`
--
ALTER TABLE `noteshortname`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `participant`
--
ALTER TABLE `participant`
  MODIFY `participantNo` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=3;
--
-- AUTO_INCREMENT for table `threefirstint`
--
ALTER TABLE `threefirstint`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `threefirststrdn`
--
ALTER TABLE `threefirststrdn`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `threefirststrup`
--
```

```
ALTER TABLE `threefirststrup`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `threesecondexhit`
--
ALTER TABLE `threesecondexhit`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `threesecondext`
--
ALTER TABLE `threesecondext`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `threesecondint`
--
ALTER TABLE `threesecondint`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=18;
--
-- AUTO_INCREMENT for table `threethirdint`
--
ALTER TABLE `threethirdint`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `threethirdstrdn`
--
ALTER TABLE `threethirdstrdn`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `threethirdstrup`
--
ALTER TABLE `threethirdstrup`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `tonedeaftest`
--
ALTER TABLE `tonedeaftest`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `twofirstdn`
--
ALTER TABLE `twofirstdn`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `twofirstup`
--
ALTER TABLE `twofirstup`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
```

```
-- AUTO_INCREMENT for table `twosecond`
--
ALTER TABLE `twosecond`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `twosecondup`
--
ALTER TABLE `twosecondup`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `visits`
--
ALTER TABLE `visits`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;
--
-- AUTO_INCREMENT for table `zonenote`
--
ALTER TABLE `zonenote`
  MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
COMMIT;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Single Tone study

Code for Program

```
<!DOCTYPE html>
<html>
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
   top: 340px;
   left: 250px;
   visibility: hidden}
   #nextstim {position: absolute;
   top: 340px;
   left: 150px;}
#higher {position: absolute;
   top: 150px;
   left: 500px;}
#same {position: absolute;
   top: 250px;
   left: 500px;}
#lower {position: absolute;
   top: 350px;
   left: 500px;}
```

```
#wrong {position: absolute;
   top: 400px;
   left:820px;}
#Score {position: absolute;
   top: 380px;
   left:820px;
   font-size:300%;
   z-index: 2;}
#heading {position: absolute;
   top: 1px;
   left:350px;
   font-size: 300%}
#Answer {position: absolute;
   top: 1px;
   left:250px;
   font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page (practice only)</button>
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
200px; padding: 5px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen to
the sine wave stimuli. <br> 2) click the green "Compare" button. <br>  3) choose whether the
keyboard tone is the <i>same </i> pitch as the sine wave stimuli or <i>higher </i> or
<i>lower </i>.</button>
<button id = "starter" onclick="stimPres()" style = "background: rgb(10,203,238);
width:500px; height: 50px; font-size:100%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%;">Next Tone</button>
```

```
<!-- Stimuli -->
<audio src= "50ms/420001.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/420002.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/420003.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/420004.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/420005.wav" preload= "auto" id= "sound5"></audio>
<audio src= "50ms/420006.wav" preload= "auto" id= "sound6"></audio>
<audio src= "50ms/420007.wav" preload= "auto" id= "sound7"></audio>
<audio src= "50ms/420008.wav" preload= "auto" id= "sound8"></audio>
<audio src= "50ms/420009.wav" preload= "auto" id= "sound9"></audio>
<audio src= "50ms/420010.wav" preload= "auto" id= "sound10"></audio>
<audio src= "50ms/420011.wav" preload= "auto" id= "sound11"></audio>
<audio src= "50ms/420012.wav" preload= "auto" id= "sound12"></audio>
<audio src= "50ms/420013.wav" preload= "auto" id= "sound13"></audio>
<audio src= "50ms/420014.wav" preload= "auto" id= "sound14"></audio>
<audio src= "50ms/420015.wav" preload= "auto" id= "sound15"></audio>
<audio src= "50ms/420016.wav" preload= "auto" id= "sound16"></audio>
<audio src= "50ms/420017.wav" preload= "auto" id= "sound17"></audio>
<audio src= "50ms/420018.wav" preload= "auto" id= "sound18"></audio>
<audio src= "50ms/420019.wav" preload= "auto" id= "sound19"></audio>
<audio src= "50ms/420020.wav" preload= "auto" id= "sound20"></audio>
<audio src= "50ms/420021.wav" preload= "auto" id= "sound21"></audio>
<audio src= "50ms/420022.wav" preload= "auto" id= "sound22"></audio>
<audio src= "50ms/420023.wav" preload= "auto" id= "sound23"></audio>
<audio src= "50ms/420024.wav" preload= "auto" id= "sound24"></audio>
<audio src= "50ms/420025.wav" preload= "auto" id= "sound25"></audio>
  <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
```

```
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
<script>
var durchange= [200, 100, 50, 45, 40, 35, 30, 25, 20, 15, 10, 9, 8, 7, 6, 5, 4];
var durNo= 0;
var duration= 200;
var i = 0;
var n = 0;
var f = 0;
var score = 0;
var total = 0;
var TotScore=0;
var correctAnswer = 1;
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
var finalStudy= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
var minPercept= 0;
var perceptAdd=0;
var idin = "<?php echo $idin; ?>";
document.getElementById('heading').innerText = duration + "ms";
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20, 21, 22, 23, 24];
function shuffleArray(array) {
   for (var b = array.length - 1; b > 0; b--) {
        var j = Math.floor(Math.random() * (b + 1));
        var temp = array[b];
        array[b] = array[j];
        array[j] = temp;
    }
    if (TotScore==0){setTimeout(myFunction, 50);};
    return array;
}
```

```
var test256= [
document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
document.getElementById("sound6"),
document.getElementById("sound7"),
document.getElementById("sound8"),
document.getElementById("sound9"),
document.getElementById("sound10"),
document.getElementById("sound11"),
document.getElementById("sound12"),
document.getElementById("sound13"),
document.getElementById("sound14"),
document.getElementById("sound15"),
document.getElementById("sound16"),
document.getElementById("sound17"),
document.getElementById("sound18"),
document.getElementById("sound19"),
document.getElementById("sound20"),
document.getElementById("sound21"),
document.getElementById("sound22"),
document.getElementById("sound23"),
document.getElementById("sound24"),
document.getElementById("sound25"),
];
var keyboardPlay= [
document.getElementById("N1"),
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
document.getElementById("N19"),
document.getElementById("N20"),
```

```
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
function changeSound() {
   for (f = 0; f< 25; f++) {
         var a= duration * 100+f+1+400000;
         var l= duration * 100+f+1+490000;
      if (duration<10){
         test256[f].src= "50ms/" + l + ".wav"
      }
      else {
      if (duration< 15) {
         if (f >14){test256[f].src= "50ms/" + a + ".wav"}}
      else if (duration< 20) {
         if (f >11){test256[f].src= "50ms/" + a + ".wav"}}
      else if (duration< 25) {
         if (f >4){test256[f].src= "50ms/" + a + ".wav"}}
      else if (duration< 30) {
         if (f >1){test256[f].src= "50ms/" + a + ".wav"}}
      else {test256[f].src= "50ms/" + a + ".wav"}
      }
   }
   document.getElementById('heading').innerText = duration + "ms";
}
function randomAnswer() {
    var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
      switch (rAns){
   case 0: if (n >23) {note1--; correctAnswer=0;} else {note1++; correctAnswer=2;} break;
    case 1:  correctAnswer = 1; break;
   case 2:  if (n <1) {note1++; correctAnswer=2;} else {note1--; correctAnswer=0;} break;
   }}
function myFunction() {
   n= array[i];
   note1= n;
   randomAnswer()
   document.getElementById('nextstim').style.visibility= 'visible';
}
function nextStim(){
   playstim();
   keypress= 0;
   startTime= new Date();
   i++;
   document.getElementById('nextstim').style.visibility= 'hidden';
   document.getElementById('compare').style.visibility= 'visible';
```

```
    if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
    keyboardPlay[note1].play();
    keypress=1;
    document.getElementById('compare').style.visibility= 'hidden';
    }}
function playstim(){
    test256[n].play();
    setTimeout (finstim, 500)}

function finstim() {silence= 0;
    }
function higher (){
    answer=2;
    checkAnswer();
    }
function same (){
    answer=1;
    checkAnswer();
    }
function lower (){
    answer=0;
    checkAnswer();
    }
    function stimPres(){
document.getElementById('starter').style.visibility= 'hidden';
}
function checkAnswer() {
answ = 0;
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
setTimeout(clearPic,1000);}
$.ajax({
            type: "POST",
            url: 'Study1Ans.php',
            data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : n, corrYN : answ,   eltime : elapsedTime},
        });
if (answ === 1) {score ++;
```

```
   if (last==1){dirCh.unshift(durNo); last= 2;}
   if (last==0){last=1}};
if (answ === 0) {score=0;
   if (last==2){dirCh.unshift(durNo); last= 0;}
   if (last==1){last=0;}
   durNo--;
   if(durNo<1){durNo=0};
   duration= durchange[durNo];
   if (TotScore>50){
      for (i=0; i<5; i++){
      perceptAdd= dirCh[i];
      minPercept= minPercept+ perceptAdd;
      }
      minPercept= Math.round(minPercept/5);
      perceptAdd= durchange[minPercept];
      $.ajax({
               type: "POST",
               url: 'Study1Res.php',
               data: {idin : idin, aveMinScore : perceptAdd},
             });



   var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Please record this result. Click OK to start again and Cancel to go to the home screen");
   if (con== true){window.location.href = "Study1.php?idin=" + idin;}
   else{gohome();};
   }
   changeSound();
};
TotScore++;
Score.innerText = TotScore;
if (score>1) {
   durNo++;
   if(durNo>16){durNo=16};
   duration= durchange[durNo];
   score=0;
   changeSound();
   }
correctAnswer = 1;
answer = 0;
if (i>23){i=0; shuffleArray(array); myFunction();}
else {myFunction();};
}
function clearPic(){
changePic.src = "Blank.png";
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
```

```
</script>
</body>
</html>
```

Recording Results to SQL

```
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
$corrone = intval($_POST["corrone"]);
$corrYN = intval($_POST["corrYN"]);
$eltime = intval($_POST["eltime"]);
$tableName = intval($_POST["tableName"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO zOneNote(Participant, duration,
ParticipantAns, corrAns, correctone, Correct, eltime) VALUES(?, ?, ?, ?, ?, ?, ?)");
$insert_row->bind_param("iiiiiii", $tester, $duration, $partans, $corrans, $corrone, $corrYN,
$eltime);
$insert_row->execute();
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO zOneNote(Participant, AveMinScore)
VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

Two-Tone Study

First Tone Low

```
<!DOCTYPE html>
<html>
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
   top: 340px;
   left: 250px;
   visibility: hidden;}
#nextstim {position: absolute;
top: 340px;
left: 150px;}
#higher {position: absolute;
   top: 150px;
   left: 500px;}
#same {position: absolute;
   top: 250px;
   left: 500px;}
#lower {position: absolute;
   top: 350px;
   left: 500px;}
#wrong {position: absolute;
   top: 400px;
   left:820px;}
#Score {position: absolute;
   top: 380px;
   left:820px;
   font-size:300%;
   z-index: 2;}
#heading {position: absolute;
   top: 1px;
   left:350px;
   font-size: 300%}
#Answer {position: absolute;
   top: 1px;
   left:250px;
   font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
```

```
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page (practice only)</button>
<button id = "home" onclick="nextEx()" style = "background: rgb(200,200,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 440px; left:40px;">
Practice Next Exercise</button>
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
220px; padding: 15px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen
for the first tone of the two tone sine wave stimuli. <br> 2) click the green "Compare" button.
<br>  3) choose whether the keyboard tone is the <i>same </i> pitch as the <u>first</u> sine
wave stimuli or <i>higher </i> or <i>lower </i>.</button>
<button id = "starter" onclick="stimPres()" style = "background: rgb(10,203,238);
width:500px; height: 50px; font-size:150%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%; visibility:hidden">Next Tone</button>
<!-- Stimuli -->
<audio src= "50ms/22000109.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/22000308.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/22000411.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/22000413.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/22000513.wav" preload= "auto" id= "sound5"></audio>
<audio src= "50ms/22000702.wav" preload= "auto" id= "sound6"></audio>
<audio src= "50ms/22000802.wav" preload= "auto" id= "sound7"></audio>
<audio src= "50ms/22000904.wav" preload= "auto" id= "sound8"></audio>
<audio src= "50ms/22001018.wav" preload= "auto" id= "sound9"></audio>
<audio src= "50ms/22001105.wav" preload= "auto" id= "sound10"></audio>
<audio src= "50ms/22001106.wav" preload= "auto" id= "sound11"></audio>
<audio src= "50ms/22001205.wav" preload= "auto" id= "sound12"></audio>
<audio src= "50ms/22001410.wav" preload= "auto" id= "sound13"></audio>
<audio src= "50ms/22001420.wav" preload= "auto" id= "sound14"></audio>
<audio src= "50ms/22001510.wav" preload= "auto" id= "sound15"></audio>
<audio src= "50ms/22001621.wav" preload= "auto" id= "sound16"></audio>
<audio src= "50ms/22001822.wav" preload= "auto" id= "sound17"></audio>
<audio src= "50ms/22002011.wav" preload= "auto" id= "sound18"></audio>
<audio src= "50ms/22002216.wav" preload= "auto" id= "sound19"></audio>
```

```
<audio src= "50ms/22002518.wav" preload= "auto" id= "sound20"></audio>
<audio src= "50ms/22001621.wav" preload= "auto" id= "sound21"></audio>
  <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
<script src="2ToneComb.js"></script>
<script>
var durchange= [200, 100, 50, 45, 40, 35, 30, 25, 20, 15, 10, 8, 7, 6, 5, 4, 3, 2, 1];
var durNo= 0;
var i = 0;
var n = 0;
var score = 0;
var total = 0;
var correctAnswer = 1;
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
```

```
var note2= 10;
var TotScore= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
var minPercept= 0;
var perceptAdd=0;
var idin = "<?php echo $idin; ?>";
var heading= document.getElementById('heading');
if(Number.isInteger(idin/2)) {heading.innerText = "Part 1 of 2: ID First (low) note: " +
duration + "ms";}
    else{heading.innerText = "Part 2 of 2: ID First (low) note: " + duration + "ms";};
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20];
function shuffleArray(array) {
   for (var b = array.length - 1; b > 0; b--) {
        var j = Math.floor(Math.random() * (b + 1));
        var temp = array[b];
        array[b] = array[j];
        array[j] = temp;
    }

    setTimeout(myFunction, 50);
    return array;
}
var test256= [
document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
document.getElementById("sound6"),
document.getElementById("sound7"),
document.getElementById("sound8"),
document.getElementById("sound9"),
document.getElementById("sound10"),
document.getElementById("sound11"),
document.getElementById("sound12"),
document.getElementById("sound13"),
document.getElementById("sound14"),
document.getElementById("sound15"),
document.getElementById("sound16"),
document.getElementById("sound17"),
document.getElementById("sound18"),
document.getElementById("sound19"),
document.getElementById("sound20"),
```

```
document.getElementById("sound21"),
];
var keyboardPlay= [
document.getElementById("N1"),
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
document.getElementById("N19"),
document.getElementById("N20"),
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
function randomAnswer() {
     var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
        switch (rAns){
    case 0: if (note1 >23) {note1--; correctAnswer=0;} else {note1++; correctAnswer=2;}
break;
     case 1:  break;
    case 2:  if (note1 <1) {note1++; correctAnswer=2;} else {note1--; correctAnswer=0;}
break;
    }}
function myFunction() {
    if(i>21){shuffleArray(array); i=0; return;}
            n= array[i];
    changeStim(duration);
    if(correctOne > correctTwo){
        i++;
        myFunction();
        return;
    }
```

```
    note1= correctOne-1;
    note2= correctTwo-1;
    randomAnswer();
    document.getElementById('nextstim').style.visibility= 'visible';
}
    function nextStim(){
    playstim();
    keypress= 0;
    startTime= new Date();
    i++;
    document.getElementById('nextstim').style.visibility= 'hidden';
    document.getElementById('compare').style.visibility= 'visible';
    if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
    keyboardPlay[note1].play();
    keypress=1;
    document.getElementById('compare').style.visibility= 'hidden';
    }}
function playstim(){
    test256[n].play();
    setTimeout (finstim, 500)}
function finstim() {silence= 0;
    }
function higher (){
    answer=2;
    checkAnswer();
    }
function same (){
    answer=1;
    checkAnswer();
    }
function lower (){
    answer=0;
    checkAnswer();
    }
function checkAnswer() {
answ = 0;
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
```

```
setTimeout(clearPic,1000);}
$.ajax({
                type: "POST",
                url: 'Study2FirstUpAns.php',
                data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : correctOne, corrtwo : correctTwo, corrYN : answ,  eltime : elapsedTime},
              });
if (answ === 1) {score ++;
   if (last==1){dirCh.unshift(durNo); last= 2;}
   if (last==0){last=1}};
if (answ === 0) {score=0;
      if (last==2){dirCh.unshift(durNo); last= 0;}
   if (last==1){last=0;}
   durNo--;
   if(durNo<1){durNo=0};
   duration= durchange[durNo];
   if (TotScore>50){
      for (i=0; i<5; i++){
      perceptAdd= dirCh[i];
      minPercept= minPercept+ perceptAdd;
      }
      minPercept= Math.round(minPercept/5);
      perceptAdd= durchange[minPercept];
      $.ajax({
                type: "POST",
                url: 'Study2FirstUpRes.php',
                data: {idin : idin, aveMinScore : perceptAdd},
              });



   var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Press OK to go to the next section");
   if (con== true){
      if(Number.isInteger(idin/2)) {window.location.href = "Study2FirstDn.php?idin=" +
idin;}
   else{gohome();};
   }}
};
//TotScore = Math.round((answ/duration*100),0) + TotScore;
TotScore++;
Score.innerText = TotScore + "   " + dirCh;
if (score>1) {
   durNo++;
   if(durNo>18){durNo=18;};
   duration= durchange[durNo];
   score=0;
   }
```

```
document.getElementById('heading').innerText = duration + "ms";
correctAnswer = 1;
answer = 0;
if (i>20){i=0;  shuffleArray(array);}
else {myFunction();};
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
function nextEx(){if(Number.isInteger(idin/2)) {window.location.href =
"Study2FirstDn.php?idin=" + idin;}
    else{gohome();};}
function clearPic(){
changePic.src = "Blank.png";
}
function stimPres(){
if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
</script>
</body>
</html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
$corrone = intval($_POST["corrone"]);
$corrtwo = intval($_POST["corrtwo"]);
$corrYN = intval($_POST["corrYN"]);
$eltime = intval($_POST["eltime"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO TwoFirstUp(Participant, duration,
ParticipantAns, corrAns, correctone, correcttwo, Correct, eltime) VALUES(?, ?, ?, ?, ?, ?, ?,
?)");
$insert_row->bind_param("iiiiiiii", $tester, $duration, $partans, $corrans, $corrone, $corrtwo,
$corrYN, $eltime);
$insert_row->execute();
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
```

```
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO TwoFirstUp(Participant, AveMinScore)
VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

First Tone High

```
<!DOCTYPE html>
<html>
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
   top: 340px;
   left: 250px;
   visibility: hidden;}
#nextstim {position: absolute;
top: 340px;
left: 150px;}
#higher {position: absolute;
   top: 150px;
   left: 500px;}
#same {position: absolute;
   top: 250px;
   left: 500px;}
#lower {position: absolute;
   top: 350px;
   left: 500px;}
#wrong {position: absolute;
   top: 400px;
   left:820px;}
#Score {position: absolute;
   top: 380px;
   left:820px;
   font-size:300%;
   z-index: 2;}
#heading {position: absolute;
   top: 1px;
```

```
    left:350px;
    font-size: 300%}
#Answer {position: absolute;
    top: 1px;
    left:250px;
    font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page (practice only)</button>
<button id = "home" onclick="nextEx()" style = "background: rgb(200,200,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 440px; left:40px;">
Practice Next Exercise</button>
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
220px; padding: 15px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen
for the first tone of the two tone sine wave stimuli. <br> 2) click the green "Compare" button.
<br>  3) choose whether the keyboard tone is the <i>same </i> pitch as the <u>first</u> sine
wave stimuli or <i>higher </i> or <i>lower </i>.</button>
<button id = "starter" style = "background: rgb(10,203,238);
width:500px; height: 50px; font-size:150%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%;">Next Tone</button>
<!-- Stimuli -->
<audio src= "50ms/22000109.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/22000308.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/22000411.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/22000413.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/22000513.wav" preload= "auto" id= "sound5"></audio>
<audio src= "50ms/22000702.wav" preload= "auto" id= "sound6"></audio>
```

```
<audio src= "50ms/22000802.wav" preload= "auto" id= "sound7"></audio>
<audio src= "50ms/22000904.wav" preload= "auto" id= "sound8"></audio>
<audio src= "50ms/22001018.wav" preload= "auto" id= "sound9"></audio>
<audio src= "50ms/22001105.wav" preload= "auto" id= "sound10"></audio>
<audio src= "50ms/22001106.wav" preload= "auto" id= "sound11"></audio>
<audio src= "50ms/22001205.wav" preload= "auto" id= "sound12"></audio>
<audio src= "50ms/22001410.wav" preload= "auto" id= "sound13"></audio>
<audio src= "50ms/22001420.wav" preload= "auto" id= "sound14"></audio>
<audio src= "50ms/22001510.wav" preload= "auto" id= "sound15"></audio>
<audio src= "50ms/22001621.wav" preload= "auto" id= "sound16"></audio>
<audio src= "50ms/22001822.wav" preload= "auto" id= "sound17"></audio>
<audio src= "50ms/22002011.wav" preload= "auto" id= "sound18"></audio>
<audio src= "50ms/22002216.wav" preload= "auto" id= "sound19"></audio>
<audio src= "50ms/22002518.wav" preload= "auto" id= "sound20"></audio>
<audio src= "50ms/22001621.wav" preload= "auto" id= "sound21"></audio>
  <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
<script src="2ToneComb.js"></script>
<script>
var durchange= [200, 100, 50, 45, 40, 35, 30, 25, 20, 15, 10, 8, 7, 6, 5, 4, 3, 2, 1];
var durNo= 0;
var i = 0;
var n = 0;
```

```
var score = 0;
var total = 0;
var correctAnswer = 1;
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
var note2= 10;
var TotScore= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
var minPercept= 0;
var perceptAdd=0;
var idin = "<?php echo $idin; ?>";
var heading= document.getElementById('heading');
if(Number.isInteger(idin/2)) {heading.innerText = "Part 2 of 2: ID First (high) note: " +
duration + "ms";}
   else{heading.innerText = "Part 1 of 2: ID First (high) note: " + duration + "ms";};
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20];
function shuffleArray(array) {
  for (var b = array.length - 1; b > 0; b--) {
      var j = Math.floor(Math.random() * (b + 1));
      var temp = array[b];
      array[b] = array[j];
      array[j] = temp;
   }

   setTimeout(myFunction, 50);
   return array;
}
var test256= [
document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
document.getElementById("sound6"),
document.getElementById("sound7"),
```

```
document.getElementById("sound8"),
document.getElementById("sound9"),
document.getElementById("sound10"),
document.getElementById("sound11"),
document.getElementById("sound12"),
document.getElementById("sound13"),
document.getElementById("sound14"),
document.getElementById("sound15"),
document.getElementById("sound16"),
document.getElementById("sound17"),
document.getElementById("sound18"),
document.getElementById("sound19"),
document.getElementById("sound20"),
document.getElementById("sound21"),
];
var keyboardPlay= [
document.getElementById("N1"),
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
document.getElementById("N19"),
document.getElementById("N20"),
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
function randomAnswer() {
    var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
       switch (rAns){
   case 0: if (note1 >23) {note1--; correctAnswer=0;} else {note1++; correctAnswer=2;}
break;
```

```
     case 1:  break;
    case 2:  if (note1 <1) {note1++; correctAnswer=2;} else {note1--; correctAnswer=0;}
break;
    }}
function myFunction() {
    if(i>21){shuffleArray(array); i=0; return;}
            n= array[i];
    changeStim(duration);
    if(correctOne<correctTwo){
        i++;
        myFunction();
        return;
    }
    note1= correctOne-1;
    note2= correctTwo-1;
    randomAnswer();
    document.getElementById('nextstim').style.visibility= 'visible';
}
function nextStim(){
    playstim();
    keypress= 0;
    startTime= new Date();
    i++;
    document.getElementById('nextstim').style.visibility= 'hidden';
    document.getElementById('compare').style.visibility= 'visible';
    if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
    keyboardPlay[note1].play();
    keypress=1;
    document.getElementById('compare').style.visibility= 'hidden';
    }}
function playstim(){
    test256[n].play();
    setTimeout (finstim, 500)}
function finstim() {silence= 0;
    }
function higher (){
    answer=2;
    checkAnswer();
    }
function same (){
    answer=1;
    checkAnswer();
    }
function lower (){
```

```
    answer=0;
    checkAnswer();
    }
function checkAnswer() {
answ = 0;
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
setTimeout(clearPic,1000);}
$.ajax({
                type: "POST",
                url: 'Study2FirstDnAns.php',
                data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : correctOne, corrtwo : correctTwo, corrYN : answ,  eltime : elapsedTime},
              });
if (answ === 1) {score ++;
    if (last==1){dirCh.unshift(durNo); last= 2;}};
    if (last==0){last=1}
if (answ === 0) {score=0;
    if (last==2){dirCh.unshift(durNo); last= 0;}
    if (last==1){last=0;}
    durNo--;
    if(durNo<1){durNo=0};
    duration= durchange[durNo];
    if (TotScore>50){
        for (i=0; i<5; i++){
        perceptAdd= dirCh[i];
        minPercept= minPercept+ perceptAdd;
        }
        minPercept= Math.round(minPercept/5);
        perceptAdd= durchange[minPercept];
    $.ajax({
                type: "POST",
                url: 'Study2FirstDnRes.php',
                data: {idin : idin, aveMinScore : perceptAdd},
              });



    var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Press OK to go to the next section");
    if (con== true){
        if(Number.isInteger(idin/2)) {gohome();}
```

```
    else{window.location.href = "Study2FirstUp.php?idin=" + idin;};
    }}
};
//TotScore = Math.round((answ/duration*100),0) + TotScore;
TotScore++;
Score.innerText = TotScore + "   " + dirCh;
if (score>1) {
    durNo++;
    if(durNo>18){durNo=18;};
    duration= durchange[durNo];
    score=0;
    }
document.getElementById('heading').innerText = duration + "ms";
correctAnswer = 1;
answer = 0;
if (i>20){i=0;  shuffleArray(array); }
else {myFunction();};
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
function nextEx(){if(Number.isInteger(idin/2)) {gohome();}
    else{window.location.href = "Study2FirstUp.php?idin=" + idin;};}
function clearPic(){
changePic.src = "Blank.png";
}
</script>
</body>
</html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
$corrone = intval($_POST["corrone"]);
$corrtwo = intval($_POST["corrtwo"]);
$corrYN = intval($_POST["corrYN"]);
$eltime = intval($_POST["eltime"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO TwoFirstDn(Participant, duration,
ParticipantAns, corrAns, correctone, correcttwo, Correct, eltime) VALUES(?, ?, ?, ?, ?, ?, ?,
?)");
```

```
$insert_row->bind_param("iiiiiiii", $tester, $duration, $partans, $corrans, $corrone, $corrtwo,
$corrYN, $eltime);
$insert_row->execute();
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO TwoFirstDn(Participant, AveMinScore)
VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

Last Tone High

```
<!DOCTYPE html>
<html>
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
   top: 340px;
   left: 250px;
   visibility: hidden;}
#nextstim {position: absolute;
top: 340px;
left: 150px;}
#higher {position: absolute;
   top: 150px;
   left: 500px;}
#same {position: absolute;
   top: 250px;
   left: 500px;}
#lower {position: absolute;
   top: 350px;
   left: 500px;}
#wrong {position: absolute;
```

```
    top: 400px;
    left:820px;}
#Score {position: absolute;
    top: 380px;
    left:820px;
    font-size:300%;
    z-index: 2;}
#heading {position: absolute;
    top: 1px;
    left:350px;
    font-size: 300%}
#Answer {position: absolute;
    top: 1px;
    left:250px;
    font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page (practice only)</button>
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
220px; padding: 15px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen
for the second tone of the two tone sine wave stimuli. <br> 2) click the green "Compare"
button. <br>  3) choose whether the keyboard tone is the <i>same </i> pitch as the
<u>second</u> sine wave stimuli or <i>higher </i> or <i>lower </i>.</button>
<button id = "starter" onclick="stimPres()" style = "background: rgb(10,203,238);
width:500px; height: 50px; font-size:150%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%;">Next Tone</button>
<!-- Stimuli -->
```

```
<audio src= "50ms/22000109.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/22000308.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/22000411.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/22000413.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/22000513.wav" preload= "auto" id= "sound5"></audio>
<audio src= "50ms/22000702.wav" preload= "auto" id= "sound6"></audio>
<audio src= "50ms/22000802.wav" preload= "auto" id= "sound7"></audio>
<audio src= "50ms/22000904.wav" preload= "auto" id= "sound8"></audio>
<audio src= "50ms/22001018.wav" preload= "auto" id= "sound9"></audio>
<audio src= "50ms/22001105.wav" preload= "auto" id= "sound10"></audio>
<audio src= "50ms/22001106.wav" preload= "auto" id= "sound11"></audio>
<audio src= "50ms/22001205.wav" preload= "auto" id= "sound12"></audio>
<audio src= "50ms/22001410.wav" preload= "auto" id= "sound13"></audio>
<audio src= "50ms/22001420.wav" preload= "auto" id= "sound14"></audio>
<audio src= "50ms/22001510.wav" preload= "auto" id= "sound15"></audio>
<audio src= "50ms/22001621.wav" preload= "auto" id= "sound16"></audio>
<audio src= "50ms/22001822.wav" preload= "auto" id= "sound17"></audio>
<audio src= "50ms/22002011.wav" preload= "auto" id= "sound18"></audio>
<audio src= "50ms/22002216.wav" preload= "auto" id= "sound19"></audio>
<audio src= "50ms/22002518.wav" preload= "auto" id= "sound20"></audio>
<audio src= "50ms/22001621.wav" preload= "auto" id= "sound21"></audio>
 <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
```

```
<script src="2ToneComb.js"></script>
<script>
var durchange= [200, 100, 50, 45, 40, 35, 30, 25, 20, 15, 10, 8, 7, 6, 5, 4, 3, 2, 1];
var durNo= 0;
var i = 0;
var n = 0;
var score = 0;
var total = 0;
var correctAnswer = 1;
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
var note2= 10;
var TotScore= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
var minPercept= 0;
var perceptAdd=0;
var idin = "<?php echo $idin; ?>";
var heading= document.getElementById('heading');
if(Number.isInteger(idin/2)) {heading.innerText = "Part 2 of 2: ID Second (low) note: " +
duration + "ms";}
   else{heading.innerText = "Part 1 of 2: ID Second (low) note: " + duration + "ms";};
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20];
function shuffleArray(array) {
  for (var b = array.length - 1; b > 0; b--) {
      var j = Math.floor(Math.random() * (b + 1));
      var temp = array[b];
      array[b] = array[j];
      array[j] = temp;
   }

   setTimeout(myFunction, 50);
   return array;
}
var test256= [
document.getElementById("sound1"),
```

```
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
document.getElementById("sound6"),
document.getElementById("sound7"),
document.getElementById("sound8"),
document.getElementById("sound9"),
document.getElementById("sound10"),
document.getElementById("sound11"),
document.getElementById("sound12"),
document.getElementById("sound13"),
document.getElementById("sound14"),
document.getElementById("sound15"),
document.getElementById("sound16"),
document.getElementById("sound17"),
document.getElementById("sound18"),
document.getElementById("sound19"),
document.getElementById("sound20"),
document.getElementById("sound21"),
];
var keyboardPlay= [
document.getElementById("N1"),
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
document.getElementById("N19"),
document.getElementById("N20"),
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
```

```
function randomAnswer() {
     var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
        switch (rAns){
    case 0: if (note2 >23) {note2--; correctAnswer=0;} else {note2++; correctAnswer=2;}
break;
     case 1:  break;
    case 2:  if (note2 <1) {note2++; correctAnswer=2;} else {note2--; correctAnswer=0;}
break;
    }}
function myFunction() {
    if(i>21){shuffleArray(array); i=0; return;}
            n= array[i];
    changeStim(duration);
    if(correctOne<correctTwo){
        i++;
        myFunction();
        return;
    }
    note1= correctOne-1;
    note2= correctTwo-1;
    randomAnswer()
    document.getElementById('nextstim').style.visibility= 'visible';
}
    function nextStim(){
    playstim();
    keypress= 0;
    startTime= new Date();
    i++;
    document.getElementById('nextstim').style.visibility= 'hidden';
    document.getElementById('compare').style.visibility= 'visible';
    if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
    keyboardPlay[note2].play();
    keypress=1;
    document.getElementById('compare').style.visibility= 'hidden';
    }}
function playstim(){
    test256[n].play();
    setTimeout (finstim, 500)}
function finstim() {silence= 0;
    }
function higher (){
    answer=2;
    checkAnswer();
```

```
    }
function same (){
    answer=1;
    checkAnswer();
    }
function lower (){
    answer=0;
    checkAnswer();
    }
function checkAnswer() {
answ = 0;
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
setTimeout(clearPic,1000);}
$.ajax({
            type: "POST",
            url: 'Study2LastAns.php',
            data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : correctOne, corrtwo : correctTwo, corrYN : answ,  eltime : elapsedTime},
        });
if (answ === 1) {score ++;
    if (last==1){dirCh.unshift(durNo); last= 2;}};
    if (last==0){last=1}
if (answ === 0) {score=0;
        if (last==2){dirCh.unshift(durNo); last= 0;}
    if (last==1){last=0;}
    durNo--;
    if(durNo<1){durNo=0};
    duration= durchange[durNo];
    if (TotScore>50){
        for (i=0; i<5; i++){
        perceptAdd= dirCh[i];
        minPercept= minPercept+ perceptAdd;
        }
        minPercept= Math.round(minPercept/5);
        perceptAdd= durchange[minPercept];
        $.ajax({
            type: "POST",
            url: 'Study2LastRes.php',
            data: {idin : idin, aveMinScore : perceptAdd},
        });
```

```
    var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Press OK to go to the next section");
    if (con== true){
        if(Number.isInteger(idin/2)) {gohome();}
    else{window.location.href = "Study2LastUp.php?idin=" + idin;}}
    }
};
//TotScore = Math.round((answ/duration*100),0) + TotScore;
TotScore++;
Score.innerText = TotScore;
if (score>1) {
    durNo++;
    if(durNo>18){durNo=18;};
    duration= durchange[durNo];
    score=0;
    }
document.getElementById('heading').innerText = duration + "ms";
correctAnswer = 1;
answer = 0;
if (i>20){i=0;  shuffleArray(array); }
else {myFunction();};
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
function clearPic(){
changePic.src = "Blank.png";
}
function stimPres(){
if (TotScore==0){ document.getElementById('starter').style.visibility= 'hidden';}
shuffleArray(array);
}
</script>
</body>
</html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
$corrone = intval($_POST["corrone"]);
$corrtwo = intval($_POST["corrtwo"]);
$corrYN = intval($_POST["corrYN"]);
```

```
$eltime = intval($_POST["eltime"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO TwoSecond(Participant, duration,
ParticipantAns, corrAns, correctone, correcttwo, Correct, eltime) VALUES(?, ?, ?, ?, ?, ?, ?,
?)");
$insert_row->bind_param("iiiiiiii", $tester, $duration, $partans, $corrans, $corrone, $corrtwo,
$corrYN, $eltime);
$insert_row->execute();
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO TwoSecond(Participant, AveMinScore)
VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

Last Tone Low

```
<!DOCTYPE html>
<html>
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
   top: 340px;
   left: 250px;
   visibility: hidden;}
#nextstim {position: absolute;
top: 340px;
left: 150px;}
#higher {position: absolute;
   top: 150px;
   left: 500px;}
```

```
#same {position: absolute;
   top: 250px;
   left: 500px;}
#lower {position: absolute;
   top: 350px;
   left: 500px;}
#wrong {position: absolute;
   top: 400px;
   left:820px;}
#Score {position: absolute;
   top: 380px;
   left:820px;
   font-size:300%;
   z-index: 2;}
#heading {position: absolute;
   top: 1px;
   left:350px;
   font-size: 300%}
#Answer {position: absolute;
   top: 1px;
   left:250px;
   font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page (practice only)</button>
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
220px; padding: 15px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen
for the second tone of the two tone sine wave stimuli. <br> 2) click the green "Compare"
button. <br>  3) choose whether the keyboard tone is the <i>same </i> pitch as the
<u>second</u> sine wave stimuli or <i>higher </i> or <i>lower </i>.</button>
<button id = "starter" onclick="stimPres()" style = "background: rgb(10,203,238);
width:500px; height: 50px; font-size:150%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
```

```
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%;">Next Tone</button>
<!-- Stimuli -->
<audio src= "50ms/22000109.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/22000308.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/22000411.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/22000413.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/22000513.wav" preload= "auto" id= "sound5"></audio>
<audio src= "50ms/22000702.wav" preload= "auto" id= "sound6"></audio>
<audio src= "50ms/22000802.wav" preload= "auto" id= "sound7"></audio>
<audio src= "50ms/22000904.wav" preload= "auto" id= "sound8"></audio>
<audio src= "50ms/22001018.wav" preload= "auto" id= "sound9"></audio>
<audio src= "50ms/22001105.wav" preload= "auto" id= "sound10"></audio>
<audio src= "50ms/22001106.wav" preload= "auto" id= "sound11"></audio>
<audio src= "50ms/22001205.wav" preload= "auto" id= "sound12"></audio>
<audio src= "50ms/22001410.wav" preload= "auto" id= "sound13"></audio>
<audio src= "50ms/22001420.wav" preload= "auto" id= "sound14"></audio>
<audio src= "50ms/22001510.wav" preload= "auto" id= "sound15"></audio>
<audio src= "50ms/22001621.wav" preload= "auto" id= "sound16"></audio>
<audio src= "50ms/22001822.wav" preload= "auto" id= "sound17"></audio>
<audio src= "50ms/22002011.wav" preload= "auto" id= "sound18"></audio>
<audio src= "50ms/22002216.wav" preload= "auto" id= "sound19"></audio>
<audio src= "50ms/22002518.wav" preload= "auto" id= "sound20"></audio>
<audio src= "50ms/22001621.wav" preload= "auto" id= "sound21"></audio>
  <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
```

```
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
<script src="2ToneComb.js"></script>
<script>
var durchange= [200, 100, 50, 45, 40, 35, 30, 25, 20, 15, 10, 8, 7, 6, 5, 4, 3, 2, 1];
var durNo= 0;
var i = 0;
var n = 0;
var score = 0;
var total = 0;
var correctAnswer = 1;
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
var note2= 10;
var TotScore= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
var minPercept= 0;
var perceptAdd=0;
var idin = "<?php echo $idin; ?>";
var heading= document.getElementById('heading');
if(Number.isInteger(idin/2)) {heading.innerText = "Part 1 of 2: ID Second (high) note: " +
duration + "ms";}
   else{heading.innerText = "Part 2 of 2: ID Second (high) note: " + duration + "ms";};
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20];
function shuffleArray(array) {
  for (var b = array.length - 1; b > 0; b--) {
       var j = Math.floor(Math.random() * (b + 1));
       var temp = array[b];
       array[b] = array[j];
       array[j] = temp;
```

```
    }
    setTimeout(myFunction, 50);
    return array;
}
var test256= [
document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
document.getElementById("sound6"),
document.getElementById("sound7"),
document.getElementById("sound8"),
document.getElementById("sound9"),
document.getElementById("sound10"),
document.getElementById("sound11"),
document.getElementById("sound12"),
document.getElementById("sound13"),
document.getElementById("sound14"),
document.getElementById("sound15"),
document.getElementById("sound16"),
document.getElementById("sound17"),
document.getElementById("sound18"),
document.getElementById("sound19"),
document.getElementById("sound20"),
document.getElementById("sound21"),
];
var keyboardPlay= [
document.getElementById("N1"),
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
document.getElementById("N19"),
document.getElementById("N20"),
```

```
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
function randomAnswer() {
    var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
      switch (rAns){
   case 0: if (note2 >23) {note2--; correctAnswer=0;} else {note2++; correctAnswer=2;}
break;
    case 1:  break;
   case 2:  if (note2 <1) {note2++; correctAnswer=2;} else {note2--; correctAnswer=0;}
break;
   }}
function myFunction() {
   if(i>21){shuffleArray(array); i=0; return;}
         n= array[i];
   changeStim(duration);
   if(correctOne>correctTwo){
      i++;
      myFunction();
      return;
   }
   note1= correctOne-1;
   note2= correctTwo-1;
   randomAnswer()
   document.getElementById('nextstim').style.visibility= 'visible';
}
   function nextStim(){
   playstim();
   keypress= 0;
   startTime= new Date();
   i++;
   document.getElementById('nextstim').style.visibility= 'hidden';
   document.getElementById('compare').style.visibility= 'visible';
   if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
   keyboardPlay[note2].play();
   keypress=1;
   document.getElementById('compare').style.visibility= 'hidden';
   }}
function playstim(){
   test256[n].play();
```

```
   setTimeout (finstim, 500)}
function finstim() {silence= 0;
   }
function higher (){
   answer=2;
   checkAnswer();
   }
function same (){
   answer=1;
   checkAnswer();
   }
function lower (){
   answer=0;
   checkAnswer();
   }
function checkAnswer() {
answ = 0;
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
setTimeout(clearPic,1000);}
$.ajax({
               type: "POST",
               url: 'Study2LastUpAns.php',
               data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : correctOne, corrtwo : correctTwo, corrYN : answ,  eltime : elapsedTime},
             });
if (answ === 1) {score ++;
   if (last==1){dirCh.unshift(durNo); last= 2;}};
   if (last==0){last=1}
if (answ === 0) {score=0;
      if (last==2){dirCh.unshift(durNo); last= 0;}
   if (last==1){last=0;}
   durNo--;
   if(durNo<1){durNo=0};
   duration= durchange[durNo];
   if (TotScore>50){
      for (i=0; i<5; i++){
      perceptAdd= dirCh[i];
      minPercept= minPercept+ perceptAdd;
      }
      minPercept= Math.round(minPercept/5);
      perceptAdd= durchange[minPercept];
```

```
      $.ajax({
            type: "POST",
            url: 'Study2LastUpRes.php',
            data: {idin : idin, aveMinScore : perceptAdd},
          });



   var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Press OK to go to the next section");
   if (con== true){
      if(Number.isInteger(idin/2)) {window.location.href = "Study2LastDn.php?idin=" +
idin;}
   else{gohome();};
   }}
};
//TotScore = Math.round((answ/duration*100),0) + TotScore;
TotScore++;
Score.innerText = TotScore;
if (score>1) {
   durNo++;
   if(durNo>18){durNo=18;};
   duration= durchange[durNo];
   score=0;
   }
document.getElementById('heading').innerText = duration + "ms";
correctAnswer = 1;
answer = 0;
if (i>20){i=0;  shuffleArray(array); }
else {myFunction();};
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
function clearPic(){
changePic.src = "Blank.png";
}
function stimPres(){
if (TotScore==0){ document.getElementById('starter').style.visibility= 'hidden';}
shuffleArray(array);
}
</script>
</body>
</html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
```

```
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
$corrone = intval($_POST["corrone"]);
$corrtwo = intval($_POST["corrtwo"]);
$corrYN = intval($_POST["corrYN"]);
$eltime = intval($_POST["eltime"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO TwoSecondUp(Participant, duration,
ParticipantAns, corrAns, correctone, correcttwo, Correct, eltime) VALUES(?, ?, ?, ?, ?, ?, ?,
?)");
$insert_row->bind_param("iiiiiiii", $tester, $duration, $partans, $corrans, $corrone, $corrtwo,
$corrYN, $eltime);
$insert_row->execute();
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO TwoSecondUp(Participant, AveMinScore)
VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

Three Tone Study

Stimuli Change calls

```
var duration= 200;
function changeStim(){
switch(duration){
case 200:
test256[0].src= "50ms/3200070210.wav";
test256[1].src= "50ms/3200142012.wav";
test256[2].src= "50ms/3200151019.wav";
```

```
test256[3].src= "50ms/3200182516.wav";
test256[4].src= "50ms/3200221624.wav";
if (n === 0){correctOne = 7; correctTwo = 2; correctThree = 10};
if (n === 1){correctOne = 14; correctTwo = 20; correctThree = 12};
if (n === 2){correctOne = 15; correctTwo = 10; correctThree = 19};
if (n === 3){correctOne = 18; correctTwo = 25; correctThree = 16};
if (n === 4){correctOne = 22; correctTwo = 16; correctThree = 24};
break;
case 100:
test256[0].src= "50ms/3100070210.wav";
test256[1].src= "50ms/3100110513.wav";
test256[2].src= "50ms/3100141017.wav";
test256[3].src= "50ms/3100162112.wav";
test256[4].src= "50ms/3100221623.wav";
if (n === 0){correctOne = 7; correctTwo = 2; correctThree = 10};
if (n === 1){correctOne = 11; correctTwo = 5; correctThree = 13};
if (n === 2){correctOne = 14; correctTwo = 10; correctThree = 17};
if (n === 3){correctOne = 16; correctTwo = 21; correctThree = 12};
if (n === 4){correctOne = 22; correctTwo = 16; correctThree = 23};
break;
case 70:
test256[0].src= "50ms/3070070314.wav";
test256[1].src= "50ms/3070101607.wav";
test256[2].src= "50ms/3070152213.wav";
test256[3].src= "50ms/3070161020.wav";
test256[4].src= "50ms/3070182412.wav";
if (n === 0){correctOne = 7; correctTwo = 3; correctThree = 14};
if (n === 1){correctOne = 10; correctTwo = 16; correctThree = 7};
if (n === 2){correctOne = 15; correctTwo = 22; correctThree = 13};
if (n === 3){correctOne = 16; correctTwo = 10; correctThree = 20};
if (n === 4){correctOne = 18; correctTwo = 24; correctThree = 12};
break;
case 60:
test256[0].src= "50ms/3060070111.wav";
test256[1].src= "50ms/3060100618.wav";
test256[2].src= "50ms/3060111605.wav";
test256[3].src= "50ms/3060132207.wav";
test256[4].src= "50ms/3060172509.wav";
if (n === 0){correctOne = 7; correctTwo = 1; correctThree = 11};
if (n === 1){correctOne = 10; correctTwo = 6; correctThree = 18};
if (n === 2){correctOne = 11; correctTwo = 16; correctThree = 5};
if (n === 3){correctOne = 13; correctTwo = 22; correctThree = 7};
if (n === 4){correctOne = 17; correctTwo = 25; correctThree = 9};
break;
case 50:
test256[0].src= "50ms/3050051102.wav";
test256[1].src= "50ms/3050130818.wav";
test256[2].src= "50ms/3050152011.wav";
```

```
test256[3].src= "50ms/3050172413.wav";
test256[4].src= "50ms/3050211623.wav";
if (n === 0){correctOne = 5; correctTwo = 11; correctThree = 2};
if (n === 1){correctOne = 13; correctTwo = 8; correctThree = 18};
if (n === 2){correctOne = 15; correctTwo = 20; correctThree = 11};
if (n === 3){correctOne = 17; correctTwo = 24; correctThree = 13};
if (n === 4){correctOne = 21; correctTwo = 16; correctThree = 23};
break;
case 45:
test256[0].src= "50ms/3045090512.wav";
test256[1].src= "50ms/3045111904.wav";
test256[2].src= "50ms/3045130615.wav";
test256[3].src= "50ms/3045151017.wav";
test256[4].src= "50ms/3045202416.wav";
if (n === 0){correctOne = 9; correctTwo = 5; correctThree = 12};
if (n === 1){correctOne = 11; correctTwo = 19; correctThree = 4};
if (n === 2){correctOne = 13; correctTwo = 6; correctThree = 15};
if (n === 3){correctOne = 15; correctTwo = 10; correctThree = 17};
if (n === 4){correctOne = 20; correctTwo = 24; correctThree = 16};
 break;
case 40:
test256[0].src= "50ms/3040070210.wav";
test256[1].src= "50ms/3040101507.wav";
test256[2].src= "50ms/3040110614.wav";
test256[3].src= "50ms/3040172315.wav";
test256[4].src= "50ms/3040211524.wav";
if (n === 0){correctOne = 7; correctTwo = 2; correctThree = 10};
if (n === 1){correctOne = 10; correctTwo = 15; correctThree = 7};
if (n === 2){correctOne = 11; correctTwo = 6; correctThree = 14};
if (n === 3){correctOne = 17; correctTwo = 23; correctThree = 15};
if (n === 4){correctOne = 21; correctTwo = 15; correctThree = 24};
break;
case 35:
test256[0].src= "50ms/3035030701.wav";
test256[1].src= "50ms/3035061203.wav";
test256[2].src= "50ms/3035140916.wav";
test256[3].src= "50ms/3035182314.wav";
test256[4].src= "50ms/3035201422.wav";
if (n === 0){correctOne = 3; correctTwo = 7; correctThree = 1};
if (n === 1){correctOne = 6; correctTwo = 12; correctThree = 3};
if (n === 2){correctOne = 14; correctTwo = 9; correctThree = 16};
if (n === 3){correctOne = 18; correctTwo = 23; correctThree = 14};
if (n === 4){correctOne = 20; correctTwo = 14; correctThree = 22};
 break;
case 30:
test256[0].src= "50ms/3030120519.wav";
test256[1].src= "50ms/3030131708.wav";
test256[2].src= "50ms/3030150817.wav";
```

```
test256[3].src= "50ms/3030181320.wav";
test256[4].src= "50ms/3030182411.wav";
if (n === 0){correctOne = 12; correctTwo = 5; correctThree = 19};
if (n === 1){correctOne = 13; correctTwo = 17; correctThree = 8};
if (n === 2){correctOne = 15; correctTwo = 8; correctThree = 17};
if (n === 3){correctOne = 18; correctTwo = 13; correctThree = 20};
if (n === 4){correctOne = 18; correctTwo = 24; correctThree = 11};
 break;
case 25:
test256[0].src= "50ms/3025050109.wav";
test256[1].src= "50ms/3025091507.wav";
test256[2].src= "50ms/3025182516.wav";
test256[3].src= "50ms/3025191225.wav";
test256[4].src= "50ms/3025202416.wav";
if (n === 0){correctOne = 5; correctTwo = 1; correctThree = 9};
if (n === 1){correctOne = 9; correctTwo = 15; correctThree = 7};
if (n === 2){correctOne = 18; correctTwo = 25; correctThree = 16};
if (n === 3){correctOne = 19; correctTwo = 12; correctThree = 25};
if (n === 4){correctOne = 20; correctTwo = 24; correctThree = 16};
 break;
case 20:
test256[0].src= "50ms/3020041001.wav";
test256[1].src= "50ms/3020070312.wav";
test256[2].src= "50ms/3020140721.wav";
test256[3].src= "50ms/3020152211.wav";
test256[4].src= "50ms/3020221524.wav";
if (n === 0){correctOne = 4; correctTwo = 10; correctThree = 1};
if (n === 1){correctOne = 7; correctTwo = 3; correctThree = 12};
if (n === 2){correctOne = 14; correctTwo = 7; correctThree = 21};
if (n === 3){correctOne = 15; correctTwo = 22; correctThree = 11};
if (n === 4){correctOne = 22; correctTwo = 15; correctThree = 24};
 break;
case 15:
test256[0].src= "50ms/3015050112.wav";
test256[1].src= "50ms/3015051201.wav";
test256[2].src= "50ms/3015101408.wav";
test256[3].src= "50ms/3015171223.wav";
test256[4].src= "50ms/3015172312.wav";
if (n === 0){correctOne = 5; correctTwo = 1; correctThree = 12};
if (n === 1){correctOne = 5; correctTwo = 12; correctThree = 1};
if (n === 2){correctOne = 10; correctTwo = 14; correctThree = 8};
if (n === 3){correctOne = 17; correctTwo = 12; correctThree = 23};
if (n === 4){correctOne = 17; correctTwo = 23; correctThree = 12};
 break;
case 10:
test256[0].src= "50ms/3010070213.wav";
test256[1].src= "50ms/3010071302.wav";
test256[2].src= "50ms/3010132008.wav";
```

```
test256[3].src= "50ms/3010211725.wav";
test256[4].src= "50ms/3010212517.wav";
if (n === 0){correctOne = 7; correctTwo = 2; correctThree = 13};
if (n === 1){correctOne = 7; correctTwo = 13; correctThree = 2};
if (n === 2){correctOne = 13; correctTwo = 20; correctThree = 8};
if (n === 3){correctOne = 21; correctTwo = 17; correctThree = 25};
if (n === 4){correctOne = 21; correctTwo = 25; correctThree = 17};
break;
case 8:
test256[0].src= "50ms/3008070314.wav";
test256[1].src= "50ms/3008171124.wav";
test256[2].src= "50ms/3008110618.wav";
test256[3].src= "50ms/3008091803.wav";
test256[4].src= "50ms/3008061201.wav";
if (n === 0){correctOne = 7; correctTwo = 3; correctThree = 14};
if (n === 1){correctOne = 17; correctTwo = 11; correctThree = 24};
if (n === 2){correctOne = 11; correctTwo = 6; correctThree = 18};
if (n === 3){correctOne = 9; correctTwo = 18; correctThree = 3};
if (n === 4){correctOne = 6; correctTwo = 12; correctThree = 1};
break;
case 7:
test256[0].src= "50ms/3007201325.wav";
test256[1].src= "50ms/3007070116.wav";
test256[2].src= "50ms/3007081504.wav";
test256[3].src= "50ms/3007152109.wav";
test256[4].src= "50ms/3007182511.wav";
if (n === 0){correctOne = 20; correctTwo = 13; correctThree = 25};
if (n === 1){correctOne = 7; correctTwo = 1; correctThree = 16};
if (n === 2){correctOne = 8; correctTwo = 15; correctThree = 4};
if (n === 3){correctOne = 15; correctTwo = 21; correctThree = 9};
if (n === 4){correctOne = 18; correctTwo = 25; correctThree = 11};
break;
case 6:
test256[0].src= "50ms/3006191423.wav";
test256[1].src= "50ms/3006130720.wav";
test256[2].src= "50ms/3006050109.wav";
test256[3].src= "50ms/3006101604.wav";
test256[4].src= "50ms/3006081303.wav";
if (n === 0){correctOne = 19; correctTwo = 14; correctThree = 23};
if (n === 1){correctOne = 13; correctTwo = 7; correctThree = 20};
if (n === 2){correctOne = 5; correctTwo = 1; correctThree = 9};
if (n === 3){correctOne = 10; correctTwo = 16; correctThree = 4};
if (n === 4){correctOne = 8; correctTwo = 13; correctThree = 3};
 break;}
}
```

First tone Low

```
<!DOCTYPE html>
<html>
```

```
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
   top: 340px;
   left: 250px;
   visibility: hidden;}
#nextstim {position: absolute;
top: 340px;
left: 150px;}
#higher {position: absolute;
   top: 150px;
   left: 500px;}
#same {position: absolute;
   top: 250px;
   left: 500px;}
#lower {position: absolute;
   top: 350px;
   left: 500px;}
#wrong {position: absolute;
   top: 400px;
   left:820px;}
#Score {position: absolute;
   top: 380px;
   left:820px;
   font-size:300%;
   z-index: 2;}
#heading {position: absolute;
   top: 1px;
   left:350px;
   font-size: 300%}
#Answer {position: absolute;
   top: 1px;
   left:250px;
   font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page</button>
<button id = "home" onclick="nextEx()" style = "background: rgb(200,200,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 440px; left:40px;">
Practice Next Exercise</button>
```

```
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
220px; padding: 15px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen
for the first tone of the three tone sine wave stimuli. <br> 2) click the green "Compare"
button. <br>  3) choose whether the keyboard tone is the <i>same </i> pitch as the
<u>first</u> sine wave stimuli or <i>higher </i> or <i>lower </i>.</button>
<button id = "starter" onclick="stimPres()" style = "background: rgb(10,203,238);
width:500px; height: 50px; font-size:150%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%;">Next Tone</button>
<!-- Stimuli -->
<audio src= "50ms/3200010913.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/3200030816.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/3200041119.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/3200041317.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/3200051317.wav" preload= "auto" id= "sound5"></audio>
  <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
```

```
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
<script src="3ToneCombStrUp.js"></script>
<script>
var durchange= [200,100,70,60,50,45,40,35,30,25,20,15,10,8,7,6];
var durNo= 0;
var tableName= 1;
var i = 0;
var n = 0;
var score = 0;
var total = 0;
var correctAnswer = 1;
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var correctThree=0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
var note2= 0;
var note3= 0;
var TotScore= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
var minPercept= 0;
var perceptAdd=10;
var idin = "<?php echo $idin; ?>";
var heading= document.getElementById('heading');
if(Number.isInteger(idin/2)) {heading.innerText = "Part 4 of 4: ID the First note: " + duration
+ "ms";}
   else{heading.innerText = "Part 1 of 4: ID the First note: " + duration + "ms";};
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4];
function shuffleArray(array) {
  for (var b = array.length - 1; b > 0; b--) {
```

```
        var j = Math.floor(Math.random() * (b + 1));
        var temp = array[b];
        array[b] = array[j];
        array[j] = temp;
    }

    if (TotScore==0){setTimeout(myFunction, 50);};
    return array;
}
var test256= [
document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
];
var keyboardPlay= [
document.getElementById("N1"),
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
document.getElementById("N19"),
document.getElementById("N20"),
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
function randomAnswer() {
     var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
        switch (rAns){
```

```
    case 0: if (note1 >23) {note1--; correctAnswer=0;} else {note1++; correctAnswer=2;}
break;
     case 1:  break;
    case 2:  if (note1 <1) {note1++; correctAnswer=2;} else {note1--; correctAnswer=0;}
break;
    }}
function myFunction() {
        n= array[i];
    changeStim(duration);
    note1= correctOne-1;
    note2= correctTwo-1;
    note3= correctThree-1;
    randomAnswer()
    document.getElementById('nextstim').style.visibility= 'visible';
}
function nextStim(){
    playstim();
    keypress= 0;
    startTime= new Date();
    i++;
    document.getElementById('nextstim').style.visibility= 'hidden';
    document.getElementById('compare').style.visibility= 'visible';
    if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
    keyboardPlay[note1].play();
    keypress=1;
    document.getElementById('compare').style.visibility= 'hidden';
    }}
function playstim(){
    test256[n].play();
    setTimeout (finstim, 500)}
function finstim() {silence= 0;
    }
function higher (){
    answer=2;
    checkAnswer();
    }
function same (){
    answer=1;
    checkAnswer();
    }
function lower (){
    answer=0;
    checkAnswer();
    }
```

```
function checkAnswer() {
answ = 0;
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
setTimeout(clearPic,1000);}
$.ajax({
                type: "POST",
                url: 'Study3FirstUpAns.php',
                data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : correctOne, corrtwo : correctTwo, corrthree : correctThree, corrYN : answ, eltime :
elapsedTime, tableName : tableName},
              });
if (answ === 1) {score ++;
   if (last==1){dirCh.unshift(durNo); last= 2;}};
   if (last==0){last=1}
if (answ === 0) {score=0;
   if (last==2){dirCh.unshift(durNo); last= 0;}
   if (last==1){last=0;}
   durNo--;
   if(durNo<1){durNo=0};
   duration= durchange[durNo];
   if (TotScore>50){
      for (i=0; i<5; i++){
      perceptAdd= dirCh[i];
      minPercept= minPercept+ perceptAdd;
      }
      minPercept= Math.round(minPercept/5);
      perceptAdd= durchange[minPercept];
   $.ajax({
                type: "POST",
                url: 'Study3FirstUpRes.php',
                data: {idin : idin, aveMinScore : perceptAdd, tableName : tableName},
              });



   var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Press OK to return to the home screen");
   if (con== true){
      if(Number.isInteger(idin/2)) {gohome();}
   else{window.location.href = "Study3FirstExt.php?idin=" + idin;};
   }}
```

```
};
//TotScore = Math.round((answ/duration*100),0) + TotScore;
TotScore++;
Score.innerText = TotScore + "   " + dirCh;
if (score>1) {
   durNo++;
   if(durNo>15){durNo=15;};
   duration= durchange[durNo];
   score=0;
   }
document.getElementById('heading').innerText = duration + "ms";
correctAnswer = 1;
answer = 0;
if (i>4){i=0;   shuffleArray(array); myFunction();}
else {myFunction();};
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
function nextEx(){if(Number.isInteger(idin/2)) {gohome();}
   else{window.location.href = "Study3FirstExt.php?idin=" + idin;};    }
function clearPic(){
changePic.src = "Blank.png";
}
function stimPres(){
if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
</script>
</body>
</html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
$corrone = intval($_POST["corrone"]);
$corrtwo = intval($_POST["corrtwo"]);
$corrthree = intval($_POST["corrthree"]);
$corrYN = intval($_POST["corrYN"]);
$eltime = intval($_POST["eltime"]);
$tableName = intval($_POST["tableName"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
```

```
$insert_row = $mysqli->prepare("INSERT INTO ThreeFirstStrUp(Participant, duration,
ParticipantAns, corrAns, correctone, correcttwo, correctthree, Correct, eltime) VALUES(?, ?,
?, ?, ?, ?, ?, ?, ?)");
$insert_row->bind_param("iiiiiiiii", $tester, $duration, $partans, $corrans, $corrone,
$corrtwo, $corrthree,$corrYN, $eltime);
$insert_row->execute();
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeFirstStrUp(Participant,
AveMinScore) VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

First Tone Middle

```
<!DOCTYPE html>
<html>
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
   top: 340px;
   left: 250px;
   visibility: hidden;}
#nextstim {position: absolute;
top: 340px;
left: 150px;}
#higher {position: absolute;
   top: 150px;
   left: 500px;}
#same {position: absolute;
   top: 250px;
   left: 500px;}
#lower {position: absolute;
```

```
    top: 350px;
    left: 500px;}
#wrong {position: absolute;
    top: 400px;
    left:820px;}
#Score {position: absolute;
    top: 380px;
    left:820px;
    font-size:300%;
    z-index: 2;}
#heading {position: absolute;
    top: 1px;
    left:350px;
    font-size: 300%}
#Answer {position: absolute;
    top: 1px;
    left:250px;
    font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page</button>
<button id = "home" onclick="nextEx()" style = "background: rgb(200,200,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 440px; left:40px;">
Practice Next Exercise</button>
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
220px; padding: 15px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen
for the first tone of the three tone sine wave stimuli. <br> 2) click the green "Compare"
button. <br>  3) choose whether the keyboard tone is the <i>same </i> pitch as the
<u>first</u> sine wave stimuli or <i>higher </i> or <i>lower </i>.</button>
<button id = "starter" onclick="stimPres()" style = "background: rgb(10,203,238);
width:500px; height: 50px; font-size:150%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
```

```
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%;">Next Tone</button>
<!-- Stimuli -->
<audio src= "50ms/3200010913.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/3200030816.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/3200041119.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/3200041317.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/3200051317.wav" preload= "auto" id= "sound5"></audio>
  <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
<script src="3ToneComb1stInt.js"></script>
<script>
var durchange= [200,100,70,60,50,45,40,35,30,25,20,15,10,8,7,6];
var durNo= 0;
var tableName= 1;
var i = 0;
var n = 0;
var score = 0;
var total = 0;
var correctAnswer = 1;
```

```
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var correctThree=0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
var note2= 0;
var note3= 0;
var TotScore= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
var minPercept= 0;
var perceptAdd=10;
var idin = "<?php echo $idin; ?>";
var heading= document.getElementById('heading');
if(Number.isInteger(idin/2)) {heading.innerText = "Part 1 of 4: ID the First note: " + duration
+ "ms";}
   else{heading.innerText = "Part 4 of 4: ID the First note: " + duration + "ms";};
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4];
function shuffleArray(array) {
  for (var b = array.length - 1; b > 0; b--) {
      var j = Math.floor(Math.random() * (b + 1));
      var temp = array[b];
      array[b] = array[j];
      array[j] = temp;
   }

   if (TotScore==0){setTimeout(myFunction, 50);};
   return array;
}
var test256= [
document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
];
var keyboardPlay= [
document.getElementById("N1"),
```

```
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
document.getElementById("N19"),
document.getElementById("N20"),
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
function randomAnswer() {
     var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
        switch (rAns){
    case 0: if (note1 >23) {note1--; correctAnswer=0;} else {note1++; correctAnswer=2;}
break;
     case 1:  break;
    case 2:  if (note1 <1) {note1++; correctAnswer=2;} else {note1--; correctAnswer=0;}
break;
    }}
function myFunction() {
            n= array[i];
    changeStim(duration);
    note1= correctOne-1;
    note2= correctTwo-1;
    note3= correctThree-1;
    randomAnswer()
    document.getElementById('nextstim').style.visibility= 'visible';
}
function nextStim(){
    playstim();
    keypress= 0;
    startTime= new Date();
```

```
   i++;
   document.getElementById('nextstim').style.visibility= 'hidden';
   document.getElementById('compare').style.visibility= 'visible';
   if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
   keyboardPlay[note1].play();
   keypress=1;
   document.getElementById('compare').style.visibility= 'hidden';
   }}
function playstim(){
   test256[n].play();
   setTimeout (finstim, 500)}
function finstim() {silence= 0;
   }
function higher (){
   answer=2;
   checkAnswer();
   }
function same (){
   answer=1;
   checkAnswer();
   }
function lower (){
   answer=0;
   checkAnswer();
   }
function checkAnswer() {
answ = 0;
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
setTimeout(clearPic,1000);}
$.ajax({
          type: "POST",
          url: 'Study3FirstIntAns.php',
          data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : correctOne, corrtwo : correctTwo, corrthree : correctThree, corrYN : answ, eltime :
elapsedTime, tableName : tableName},
        });
if (answ === 1) {score ++;
```

```
    if (last==1){dirCh.unshift(durNo); last= 2;}};
    if (last==0){last=1}
if (answ === 0) {score=0;
    if (last==2){dirCh.unshift(durNo); last= 0;}
    if (last==1){last=0;}
    durNo--;
    if(durNo<1){durNo=0};
    duration= durchange[durNo];
    if (TotScore>50){
        for (i=0; i<5; i++){
        perceptAdd= dirCh[i];
        minPercept= minPercept+ perceptAdd;
        }
        minPercept= Math.round(minPercept/5);
        perceptAdd= durchange[minPercept];
    $.ajax({
                type: "POST",
                url: 'Study3FirstIntRes.php',
                data: {idin : idin, aveMinScore : perceptAdd, tableName : tableName},
              });



    var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Press OK to continue");
    if (con== true){
        if(Number.isInteger(idin/2)) {window.location.href = "Study3FirstDn.php?idin=" +
idin;}
    else{gohome();}
    }}
};
//TotScore = Math.round((answ/duration*100),0) + TotScore;
TotScore++;
Score.innerText = TotScore + "   " + dirCh;
if (score>1) {
    durNo++;
    if(durNo>15){durNo=15;};
    duration= durchange[durNo];
    score=0;
    }
document.getElementById('heading').innerText = duration + "ms";
correctAnswer = 1;
answer = 0;
if (i>4){i=0;  shuffleArray(array); myFunction();}
else {myFunction();};
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
```

```
function nextEx(){if(Number.isInteger(idin/2)) {window.location.href =
"Study3FirstDn.php?idin=" + idin;}
   else{gohome();}     }
function clearPic(){
changePic.src = "Blank.png";
}
function stimPres(){
if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
</script>
</body>
</html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
$corrone = intval($_POST["corrone"]);
$corrtwo = intval($_POST["corrtwo"]);
$corrthree = intval($_POST["corrthree"]);
$corrYN = intval($_POST["corrYN"]);
$eltime = intval($_POST["eltime"]);
$tableName = intval($_POST["tableName"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeFirstInt(Participant, duration,
ParticipantAns, corrAns, correctone, correcttwo, correctthree, Correct, eltime) VALUES(?, ?,
?, ?, ?, ?, ?, ?, ?)");
$insert_row->bind_param("iiiiiiiii", $tester, $duration, $partans, $corrans, $corrone,
$corrtwo, $corrthree,$corrYN, $eltime);
$insert_row->execute();
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
```

```
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeFirstInt(Participant, AveMinScore)
VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

First tone High

```
<!DOCTYPE html>
<html>
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
   top: 340px;
   left: 250px;
   visibility: hidden;}
#nextstim {position: absolute;
top: 340px;
left: 150px;}
#higher {position: absolute;
   top: 150px;
   left: 500px;}
#same {position: absolute;
   top: 250px;
   left: 500px;}
#lower {position: absolute;
   top: 350px;
   left: 500px;}
#wrong {position: absolute;
   top: 400px;
   left:820px;}
#Score {position: absolute;
   top: 380px;
   left:820px;
   font-size:300%;
   z-index: 2;}
#heading {position: absolute;
   top: 1px;
   left:350px;
   font-size: 300%}
#Answer {position: absolute;
   top: 1px;
   left:250px;
```

```
   font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page</button>
<button id = "home" onclick="nextEx()" style = "background: rgb(200,200,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 440px; left:40px;">
Practice Next Exercise</button>
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
220px; padding: 15px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen
for the first tone of the three tone sine wave stimuli. <br> 2) click the green "Compare"
button. <br>  3) choose whether the keyboard tone is the <i>same </i> pitch as the
<u>first</u> sine wave stimuli or <i>higher </i> or <i>lower </i>.</button>
<button id = "starter" onclick="stimPres()" style = "background: rgb(10,203,238);
width:500px; height: 50px; font-size:150%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%;">Next Tone</button>
<!-- Stimuli -->
<audio src= "50ms/3200010913.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/3200030816.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/3200041119.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/3200041317.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/3200051317.wav" preload= "auto" id= "sound5"></audio>
  <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
```

```
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
<script src="3ToneCombStrDn.js"></script>
<script>
var durchange= [200,100,70,60,50,45,40,35,30,25,20,15,10,8,7,6];
var durNo= 0;
var tableName= 1;
var i = 0;
var n = 0;
var score = 0;
var total = 0;
var correctAnswer = 1;
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var correctThree=0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
var note2= 0;
var note3= 0;
var TotScore= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
```

```
var minPercept= 0;
var perceptAdd=10;
var idin = "<?php echo $idin; ?>";
var heading= document.getElementById('heading');
if(Number.isInteger(idin/2)) {heading.innerText = "Part 2 of 4: ID the First note: " + duration
+ "ms";}
   else{heading.innerText = "Part 3 of 4: ID the First note: " + duration + "ms";};
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4];
function shuffleArray(array) {
  for (var b = array.length - 1; b > 0; b--) {
      var j = Math.floor(Math.random() * (b + 1));
      var temp = array[b];
      array[b] = array[j];
      array[j] = temp;
   }

   if (TotScore==0){setTimeout(myFunction, 50);};
   return array;
}
var test256= [
document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
];
var keyboardPlay= [
document.getElementById("N1"),
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
```

```
document.getElementById("N19"),
document.getElementById("N20"),
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
function randomAnswer() {
     var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
        switch (rAns){
    case 0: if (note1 >23) {note1--; correctAnswer=0;} else {note1++; correctAnswer=2;}
break;
     case 1:  break;
    case 2:  if (note1 <1) {note1++; correctAnswer=2;} else {note1--; correctAnswer=0;}
break;
    }}
function myFunction() {
            n= array[i];
    changeStim(duration);
    note1= correctOne-1;
    note2= correctTwo-1;
    note3= correctThree-1;
    randomAnswer()
    document.getElementById('nextstim').style.visibility= 'visible';
}
function nextStim(){
    playstim();
    keypress= 0;
    startTime= new Date();
    i++;
    document.getElementById('nextstim').style.visibility= 'hidden';
    document.getElementById('compare').style.visibility= 'visible';
    if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
    keyboardPlay[note1].play();
    keypress=1;
    document.getElementById('compare').style.visibility= 'hidden';
    }}
function playstim(){
    test256[n].play();
    setTimeout (finstim, 500)}
function finstim() {silence= 0;
    }
```

```
function higher (){
    answer=2;
    checkAnswer();
    }
function same (){
    answer=1;
    checkAnswer();
    }
function lower (){
    answer=0;
    checkAnswer();
    }
function checkAnswer() {
answ = 0;
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
setTimeout(clearPic,1000);}
$.ajax({
                type: "POST",
                url: 'Study3FirstDnAns.php',
                data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : correctOne, corrtwo : correctTwo, corrthree : correctThree, corrYN : answ, eltime :
elapsedTime, tableName : tableName},
              });
if (answ === 1) {score ++;
    if (last==1){dirCh.unshift(durNo); last= 2;}};
    if (last==0){last=1}
if (answ === 0) {score=0;
    if (last==2){dirCh.unshift(durNo); last= 0;}
    if (last==1){last=0;}
    durNo--;
    if(durNo<1){durNo=0};
    duration= durchange[durNo];
    if (TotScore>50){
        for (i=0; i<5; i++){
        perceptAdd= dirCh[i];
        minPercept= minPercept+ perceptAdd;
        }
        minPercept= Math.round(minPercept/5);
        perceptAdd= durchange[minPercept];
    $.ajax({
                type: "POST",
```

```
              url: 'Study3FirstDnRes.php',
              data: {idin : idin, aveMinScore : perceptAdd, tableName : tableName},
            });



   var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Press OK to return to the home screen");
   if (con== true){
      if(Number.isInteger(idin/2)) {window.location.href = "Study3FirstExt.php?idin=" +
idin;}
   else{window.location.href = "Study3FirstInt.php?idin=" + idin;}
   }}
};
//TotScore = Math.round((answ/duration*100),0) + TotScore;
TotScore++;
Score.innerText = TotScore + "   " + dirCh;
if (score>1) {
   durNo++;
   if(durNo>15){durNo=15;};
   duration= durchange[durNo];
   score=0;
   }
document.getElementById('heading').innerText = duration + "ms";
correctAnswer = 1;
answer = 0;
if (i>4){i=0;   shuffleArray(array); myFunction();}
else {myFunction();};
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
function nextEx(){if(Number.isInteger(idin/2)) {window.location.href =
"Study3FirstExt.php?idin=" + idin;}
   else{window.location.href = "Study3FirstInt.php?idin=" + idin;} }
function clearPic(){
changePic.src = "Blank.png";
}
function stimPres(){
if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
</script>
</body>
</html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
```

```
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
$corrone = intval($_POST["corrone"]);
$corrtwo = intval($_POST["corrtwo"]);
$corrthree = intval($_POST["corrthree"]);
$corrYN = intval($_POST["corrYN"]);
$eltime = intval($_POST["eltime"]);
$tableName = intval($_POST["tableName"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeFirstStrDn(Participant, duration,
ParticipantAns, corrAns, correctone, correcttwo, correctthree, Correct, eltime) VALUES(?, ?,
?, ?, ?, ?, ?, ?, ?)");
$insert_row->bind_param("iiiiiiiii", $tester, $duration, $partans, $corrans, $corrone,
$corrtwo, $corrthree,$corrYN, $eltime);
$insert_row->execute();
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeFirstStrDn(Participant,
AveMinScore) VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

Second Tone Low

```
<!DOCTYPE html>
<html>
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
```

```
    top: 340px;
    left: 250px;
    visibility: hidden;}
#nextstim {position: absolute;
top: 340px;
left: 150px;}
#higher {position: absolute;
    top: 150px;
    left: 500px;}
#same {position: absolute;
    top: 250px;
    left: 500px;}
#lower {position: absolute;
    top: 350px;
    left: 500px;}
#wrong {position: absolute;
    top: 400px;
    left:820px;}
#Score {position: absolute;
    top: 380px;
    left:820px;
    font-size:300%;
    z-index: 2;}
#heading {position: absolute;
    top: 1px;
    left:350px;
    font-size: 300%}
#Answer {position: absolute;
    top: 1px;
    left:250px;
    font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page</button>
<button id = "home" onclick="nextEx()" style = "background: rgb(200,200,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 440px; left:40px;">
Practice Next Exercise</button>
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
220px; padding: 15px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen
for the Second tone of the three tone sine wave stimuli. <br> 2) click the green "Compare"
button. <br>  3) choose whether the keyboard tone is the <i>same </i> pitch as the
<u>Second</u> sine wave stimuli or <i>higher </i> or <i>lower </i>.</button>
```

```
<button id = "starter" onclick="stimPres()" style = "background: rgb(10,203,238);
width:500px; height: 50px; font-size:150%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%;">Next Tone</button>
<!-- Stimuli -->
<audio src= "50ms/3200041911.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/3200070210.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/3200101813.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/3200110509.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/3200120509.wav" preload= "auto" id= "sound5"></audio>
<audio src= "50ms/3200142012.wav" preload= "auto" id= "sound6"></audio>
<audio src= "50ms/3200151019.wav" preload= "auto" id= "sound7"></audio>
<audio src= "50ms/3200182516.wav" preload= "auto" id= "sound8"></audio>
<audio src= "50ms/3200201116.wav" preload= "auto" id= "sound9"></audio>
<audio src= "50ms/3200221624.wav" preload= "auto" id= "sound10"></audio>
 <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
```

```
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
<script src="3ToneComb2ndExt.js"></script>
<script>
var durchange= [200,100,70,60,50,45,40,35,30,25,20,15,10,8,7,6];
var durNo= 0;
var tableName= 1;
var i = 0;
var n = 0;
var score = 0;
var total = 0;
var correctAnswer = 1;
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var correctThree=0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
var note2= 0;
var note3= 0;
var TotScore= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
var minPercept= 0;
var perceptAdd=10;
var idin = "<?php echo $idin; ?>";
var heading= document.getElementById('heading');
if(Number.isInteger(idin/2)) {heading.innerText = "Part 3 of 3: ID the Second note: " +
duration + "ms";}
    else{heading.innerText = "Part 1 of 3: ID the Second note: " + duration + "ms";};
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4,5,6,7,8,9];
function shuffleArray(array) {
   for (var b = array.length - 1; b > 0; b--) {
```

```
        var j = Math.floor(Math.random() * (b + 1));
        var temp = array[b];
        array[b] = array[j];
        array[j] = temp;
    }

    setTimeout(myFunction, 50);
    return array;
}
var test256= [
document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
document.getElementById("sound6"),
document.getElementById("sound7"),
document.getElementById("sound8"),
document.getElementById("sound9"),
document.getElementById("sound10"),
];
var keyboardPlay= [
document.getElementById("N1"),
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
document.getElementById("N19"),
document.getElementById("N20"),
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
```

```
function randomAnswer() {
     var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
        switch (rAns){
    case 0: if (note2 >23) {note2--; correctAnswer=0;} else {note2++; correctAnswer=2;}
break;
     case 1:  correctAnswer=1; break;
    case 2:  if (note2 <1) {note2++; correctAnswer=2;} else {note2--; correctAnswer=0;}
break;
    }}
function myFunction() {
    n= array[i];
    if(i>9){shuffleArray(array); i=0; return;}
    changeStim(duration);
    note1= correctOne-1;
    note2= correctTwo-1;
    note3= correctThree-1;
    if(correctOne<correctTwo){
        i++;
        myFunction();
        return;
    }
    randomAnswer()
    document.getElementById('nextstim').style.visibility= 'visible';
}
function nextStim(){
    playstim();
    keypress= 0;
    startTime= new Date();
    i++;
    document.getElementById('nextstim').style.visibility= 'hidden';
    document.getElementById('compare').style.visibility= 'visible';
    if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
    keyboardPlay[note2].play();
    keypress=1;
    document.getElementById('compare').style.visibility= 'hidden';
    }}
function playstim(){
    test256[n].play();
    setTimeout (finstim, 500)}
function finstim() {silence= 0;
    }
function higher (){
    answer=2;
```

```
    checkAnswer();
    }
function same (){
    answer=1;
    checkAnswer();
    }
function lower (){
    answer=0;
    checkAnswer();
    }
function checkAnswer() {
answ = 0;
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
setTimeout(clearPic,1000);}
$.ajax({
            type: "POST",
            url: 'Study3SecondLoAns.php',
            data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : correctOne, corrtwo : correctTwo, corrthree : correctThree, corrYN : answ, eltime :
elapsedTime, tableName : tableName},
          });
if (answ === 1) {score ++;
    if (last==1){dirCh.unshift(durNo); last= 2;}};
    if (last==0){last=1}
if (answ === 0) {score=0;
    if (last==2){dirCh.unshift(durNo); last= 0;}
    if (last==1){last=0;}
    durNo--;
    if(durNo<1){durNo=0};
    duration= durchange[durNo];
    if (TotScore>50){
        for (i=0; i<5; i++){
        perceptAdd= dirCh[i];
        minPercept= minPercept+ perceptAdd;
        }
        minPercept= Math.round(minPercept/5);
        perceptAdd= durchange[minPercept];
    $.ajax({
            type: "POST",
            url: 'Study3SecondLoRes.php',
            data: {idin : idin, aveMinScore : perceptAdd, tableName : tableName},
```

```
        });



   var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Press OK to continue");
   if (con== true){
      if(Number.isInteger(idin/2)) {gohome();}
   else{window.location.href = "Study3SecondHi.php?idin=" + idin;};
   }}
};
//TotScore = Math.round((answ/duration*100),0) + TotScore;
TotScore++;
Score.innerText = TotScore + "   " + dirCh;
if (score>1) {
   durNo++;
   if(durNo>15){durNo=15;};
   duration= durchange[durNo];
   score=0;
   }
document.getElementById('heading').innerText = duration + "ms";
correctAnswer = 1;
answer = 0;
if (i>9){i=0;   shuffleArray(array);}
else {myFunction();};
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
function nextEx(){if(Number.isInteger(idin/2)) {gohome();}
   else{window.location.href = "Study3SecondHi.php?idin=" + idin;};}
function clearPic(){
changePic.src = "Blank.png";
}
function stimPres(){
if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
</script>
</body>
</html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
```

```
$corrone = intval($_POST["corrone"]);
$corrtwo = intval($_POST["corrtwo"]);
$corrthree = intval($_POST["corrthree"]);
$corrYN = intval($_POST["corrYN"]);
$eltime = intval($_POST["eltime"]);
$tableName = intval($_POST["tableName"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeSecondExt (Participant, duration,
ParticipantAns, corrAns, correctone, correcttwo, correctthree, Correct, eltime) VALUES(?, ?,
?, ?, ?, ?, ?, ?, ?)");
$insert_row->bind_param("iiiiiiiii", $tester, $duration, $partans, $corrans, $corrone,
$corrtwo, $corrthree,$corrYN, $eltime);
$insert_row->execute();
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeSecondExt (Participant,
AveMinScore) VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

Second Tone Middle

```
<!DOCTYPE html>
<html>
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
   top: 340px;
   left: 250px;
   visibility: hidden;}
#nextstim {position: absolute;
```

```
top: 340px;
left: 150px;}
#higher {position: absolute;
   top: 150px;
   left: 500px;}
#same {position: absolute;
   top: 250px;
   left: 500px;}
#lower {position: absolute;
   top: 350px;
   left: 500px;}
#wrong {position: absolute;
   top: 400px;
   left:820px;}
#Score {position: absolute;
   top: 380px;
   left:820px;
   font-size:300%;
   z-index: 2;}
#heading {position: absolute;
   top: 1px;
   left:350px;
   font-size: 300%}
#Answer {position: absolute;
   top: 1px;
   left:250px;
   font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page</button>
<button id = "home" onclick="nextEx()" style = "background: rgb(200,200,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 440px; left:40px;">
Practice Next Exercise</button>
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
220px; padding: 15px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen
for the Second tone of the three tone sine wave stimuli. <br> 2) click the green "Compare"
button. <br>  3) choose whether the keyboard tone is the <i>same </i> pitch as the
<u>Second</u> sine wave stimuli or <i>higher </i> or <i>lower </i>.</button>
<button id = "starter" onclick="stimPres()" style = "background: rgb(10,203,238);
width:500px; height: 50px; font-size:150%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
```

```
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%;">Next Tone</button>
<!-- Stimuli -->
<audio src= "50ms/3200010913.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/3200030816.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/3200041119.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/3200041317.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/3200051317.wav" preload= "auto" id= "sound5"></audio>
<audio src= "50ms/3200010913.wav" preload= "auto" id= "sound6"></audio>
<audio src= "50ms/3200030816.wav" preload= "auto" id= "sound7"></audio>
<audio src= "50ms/3200041119.wav" preload= "auto" id= "sound8"></audio>
<audio src= "50ms/3200041317.wav" preload= "auto" id= "sound9"></audio>
<audio src= "50ms/3200051317.wav" preload= "auto" id= "sound10"></audio>
  <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
```

```
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
<script src="3ToneCombStr.js"></script>
<script>
var durchange= [200,100,70,60,50,45,40,35,30,25,20,15,10,8,7,6];
var durNo= 0;
var tableName= 1;
var i = 0;
var n = 0;
var score = 0;
var total = 0;
var correctAnswer = 1;
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var correctThree=0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
var note2= 0;
var note3= 0;
var TotScore= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
var minPercept= 0;
var perceptAdd=10;
var idin = "<?php echo $idin; ?>";
var heading= document.getElementById('heading');
if(Number.isInteger(idin/2)) {heading.innerText = "Part 1 of 3: ID the Second note: " +
duration + "ms";}
   else{heading.innerText = "Part 3 of 3: ID the Second note: " + duration + "ms";};
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4,5,6,7,8,9];
function shuffleArray(array) {
  for (var b = array.length - 1; b > 0; b--) {
      var j = Math.floor(Math.random() * (b + 1));
      var temp = array[b];
      array[b] = array[j];
      array[j] = temp;
```

```
    }

    setTimeout(myFunction, 50);
    return array;
}
var test256= [
document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
document.getElementById("sound6"),
document.getElementById("sound7"),
document.getElementById("sound8"),
document.getElementById("sound9"),
document.getElementById("sound10"),
];
var keyboardPlay= [
document.getElementById("N1"),
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
document.getElementById("N19"),
document.getElementById("N20"),
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
function randomAnswer() {
     var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
        switch (rAns){
```

```
    case 0: if (note2 >23) {note2--; correctAnswer=0;} else {note2++; correctAnswer=2;}
break;
     case 1:  correctAnswer=1; break;
    case 2:  if (note2 <1) {note2++; correctAnswer=2;} else {note2--; correctAnswer=0;}
break;
    }}
function myFunction() {
            n= array[i];
    changeStim(duration);
    note1= correctOne-1;
    note2= correctTwo-1;
    note3= correctThree-1;
    randomAnswer()
    document.getElementById('nextstim').style.visibility= 'visible';
}
function nextStim(){
    playstim();
    keypress= 0;
    startTime= new Date();
    i++;
    document.getElementById('nextstim').style.visibility= 'hidden';
    document.getElementById('compare').style.visibility= 'visible';
    if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
    keyboardPlay[note2].play();
    keypress=1;
    document.getElementById('compare').style.visibility= 'hidden';
    }}
function playstim(){
    test256[n].play();
    setTimeout (finstim, 500)}
function finstim() {silence= 0;
    }
function higher (){
    answer=2;
    checkAnswer();
    }
function same (){
    answer=1;
    checkAnswer();
    }
function lower (){
    answer=0;
    checkAnswer();
    }
```

```
function checkAnswer() {
answ = 0;
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
setTimeout(clearPic,1000);}
$.ajax({
		type: "POST",
		url: 'Study3SecondIntAns.php',
		data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : correctOne, corrtwo : correctTwo, corrthree : correctThree, corrYN : answ, eltime :
elapsedTime, tableName : tableName},
	});
if (answ === 1) {score ++;
   if (last==1){dirCh.unshift(durNo); last= 2;}};
   if (last==0){last=1}
if (answ === 0) {score=0;
   if (last==2){dirCh.unshift(durNo); last= 0;}
   if (last==1){last=0;}
   durNo--;
   if(durNo<1){durNo=0};
   duration= durchange[durNo];
   if (TotScore>50){
      for (i=0; i<5; i++){
      perceptAdd= dirCh[i];
      minPercept= minPercept+ perceptAdd;
      }
      minPercept= Math.round(minPercept/5);
      perceptAdd= durchange[minPercept];
   $.ajax({
		type: "POST",
		url: 'Study3SecondIntRes.php',
		data: {idin : idin, aveMinScore : perceptAdd, tableName : tableName},
	});



   var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Press OK to continue");
   if (con== true){
      if(Number.isInteger(idin/2)) {window.location.href = "Study3SecondHi.php?idin=" +
idin;}
      else{gohome();};
```

```
    }}
};
//TotScore = Math.round((answ/duration*100),0) + TotScore;
TotScore++;
Score.innerText = TotScore + "   " + dirCh;
if (score>1) {
    durNo++;
    if(durNo>15){durNo=15;};
    duration= durchange[durNo];
    score=0;
    }
document.getElementById('heading').innerText = duration + "ms";
correctAnswer = 1;
answer = 0;
if (i>9){i=0;  shuffleArray(array);}
else {myFunction();};
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
function nextEx(){if(Number.isInteger(idin/2)) {window.location.href =
"Study3SecondHi.php?idin=" + idin;}
        else{gohome();};}
function clearPic(){
changePic.src = "Blank.png";
}
function stimPres(){
if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
</script>
</body>
</html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
$corrone = intval($_POST["corrone"]);
$corrtwo = intval($_POST["corrtwo"]);
$corrthree = intval($_POST["corrthree"]);
$corrYN = intval($_POST["corrYN"]);
$eltime = intval($_POST["eltime"]);
$tableName = intval($_POST["tableName"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
```

```
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeSecondInt (Participant, duration,
ParticipantAns, corrAns, correctone, correcttwo, correctthree, Correct, eltime) VALUES(?, ?,
?, ?, ?, ?, ?, ?, ?)");
$insert_row->bind_param("iiiiiiiii", $tester, $duration, $partans, $corrans, $corrone,
$corrtwo, $corrthree,$corrYN, $eltime);
$insert_row->execute();
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeSecondInt (Participant,
AveMinScore) VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

Second Tone High

```
<!DOCTYPE html>
<html>
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
   top: 340px;
   left: 250px;
   visibility: hidden;}
#nextstim {position: absolute;
top: 340px;
left: 150px;}
#higher {position: absolute;
   top: 150px;
   left: 500px;}
#same {position: absolute;
   top: 250px;
   left: 500px;}
```

```
#lower {position: absolute;
   top: 350px;
   left: 500px;}
#wrong {position: absolute;
   top: 400px;
   left:820px;}
#Score {position: absolute;
   top: 380px;
   left:820px;
   font-size:300%;
   z-index: 2;}
#heading {position: absolute;
   top: 1px;
   left:350px;
   font-size: 300%}
#Answer {position: absolute;
   top: 1px;
   left:250px;
   font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page</button>
<button id = "home" onclick="nextEx()" style = "background: rgb(200,200,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 440px; left:40px;">
Practice Next Exercise</button>
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
220px; padding: 15px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen
for the Second tone of the three tone sine wave stimuli. <br> 2) click the green "Compare"
button. <br>  3) choose whether the keyboard tone is the <i>same </i> pitch as the
<u>Second</u> sine wave stimuli or <i>higher </i> or <i>lower </i>.</button>
<button id = "starter" onclick="stimPres()" style = "background: rgb(10,203,238);
width:500px; height: 50px; font-size:150%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
```

```
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%;">Next Tone</button>
<!-- Stimuli -->
<audio src= "50ms/3200041911.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/3200070210.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/3200101813.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/3200110509.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/3200120509.wav" preload= "auto" id= "sound5"></audio>
<audio src= "50ms/3200142012.wav" preload= "auto" id= "sound6"></audio>
<audio src= "50ms/3200151019.wav" preload= "auto" id= "sound7"></audio>
<audio src= "50ms/3200182516.wav" preload= "auto" id= "sound8"></audio>
<audio src= "50ms/3200201116.wav" preload= "auto" id= "sound9"></audio>
<audio src= "50ms/3200221624.wav" preload= "auto" id= "sound10"></audio>
  <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
<script src="3ToneComb2ndExt.js"></script>
<script>
var durchange= [200,100,70,60,50,45,40,35,30,25,20,15,10,8,7,6];
var durNo= 0;
```

```
var tableName= 1;
var i = 0;
var n = 0;
var score = 0;
var total = 0;
var correctAnswer = 1;
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var correctThree=0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
var note2= 0;
var note3= 0;
var TotScore= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
var minPercept= 0;
var perceptAdd=10;
var idin = "<?php echo $idin; ?>";
var heading= document.getElementById('heading');
if(Number.isInteger(idin/2)) {heading.innerText = "Part 2 of 3: ID the Second note: " +
duration + "ms";}
   else{heading.innerText = "Part 2 of 3: ID the Second note: " + duration + "ms";};
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4,5,6,7,8,9];
function shuffleArray(array) {
  for (var b = array.length - 1; b > 0; b--) {
      var j = Math.floor(Math.random() * (b + 1));
      var temp = array[b];
      array[b] = array[j];
      array[j] = temp;
   }

   setTimeout(myFunction, 50);
   return array;
}
var test256= [
document.getElementById("sound1"),
document.getElementById("sound2"),
```

```
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
document.getElementById("sound6"),
document.getElementById("sound7"),
document.getElementById("sound8"),
document.getElementById("sound9"),
document.getElementById("sound10"),
];
var keyboardPlay= [
document.getElementById("N1"),
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
document.getElementById("N19"),
document.getElementById("N20"),
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
function randomAnswer() {
     var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
        switch (rAns){
    case 0: if (note2 >23) {note2--; correctAnswer=0;} else {note2++; correctAnswer=2;}
break;
     case 1:  correctAnswer=1; break;
    case 2:  if (note2 <1) {note2++; correctAnswer=2;} else {note2--; correctAnswer=0;}
break;
    }}
function myFunction() {
    n= array[i];
```

```
    if(i>9){shuffleArray(array); i=0; return;}
    changeStim(duration);
    note1= correctOne-1;
    note2= correctTwo-1;
    note3= correctThree-1;
    if(correctOne>correctTwo){
        i++;
        myFunction();
        return;
    }
    randomAnswer()
    document.getElementById('nextstim').style.visibility= 'visible';
}
function nextStim(){
    playstim();
    keypress= 0;
    startTime= new Date();
    i++;
    document.getElementById('nextstim').style.visibility= 'hidden';
    document.getElementById('compare').style.visibility= 'visible';
    if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
    keyboardPlay[note2].play();
    keypress=1;
    document.getElementById('compare').style.visibility= 'hidden';
    }}
function playstim(){
    test256[n].play();
    setTimeout (finstim, 500)}
function finstim() {silence= 0;
    }
function higher (){
    answer=2;
    checkAnswer();
    }
function same (){
    answer=1;
    checkAnswer();
    }
function lower (){
    answer=0;
    checkAnswer();
    }
function checkAnswer() {
answ = 0;
```

```
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
setTimeout(clearPic,1000);}
$.ajax({
            type: "POST",
            url: 'Study3SecondHiAns.php',
            data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : correctOne, corrtwo : correctTwo, corrthree : correctThree, corrYN : answ, eltime :
elapsedTime, tableName : tableName},
          });
if (answ === 1) {score ++;
   if (last==1){dirCh.unshift(durNo); last= 2;}};
   if (last==0){last=1}
if (answ === 0) {score=0;
   if (last==2){dirCh.unshift(durNo); last= 0;}
   if (last==1){last=0;}
   durNo--;
   if(durNo<1){durNo=0};
   duration= durchange[durNo];
   if (TotScore>50){
      for (i=0; i<5; i++){
      perceptAdd= dirCh[i];
      minPercept= minPercept+ perceptAdd;
      }
      minPercept= Math.round(minPercept/5);
      perceptAdd= durchange[minPercept];
   $.ajax({
            type: "POST",
            url: 'Study3SecondHiRes.php',
            data: {idin : idin, aveMinScore : perceptAdd, tableName : tableName},
          });



   var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Press OK to continue");
   if (con== true){
      if(Number.isInteger(idin/2)) {window.location.href = "Study3SecondLo.php?idin=" +
idin;}
   else{window.location.href = "Study3SecondInt.php?idin=" + idin;};
   }}
};
```

```
//TotScore = Math.round((answ/duration*100),0) + TotScore;
TotScore++;
Score.innerText = TotScore + "   " + dirCh;
if (score>1) {
   durNo++;
   if(durNo>15){durNo=15;};
   duration= durchange[durNo];
   score=0;
   }
document.getElementById('heading').innerText = duration + "ms";
correctAnswer = 1;
answer = 0;
if (i>9){i=0;  shuffleArray(array);}
else {myFunction();};
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
function nextEx(){if(Number.isInteger(idin/2)) {window.location.href =
"Study3SecondLo.php?idin=" + idin;}
   else{window.location.href = "Study3SecondInt.php?idin=" + idin;};   }
function clearPic(){
changePic.src = "Blank.png";
}
function stimPres(){
if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
</script>
</body>
</html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
$corrone = intval($_POST["corrone"]);
$corrtwo = intval($_POST["corrtwo"]);
$corrthree = intval($_POST["corrthree"]);
$corrYN = intval($_POST["corrYN"]);
$eltime = intval($_POST["eltime"]);
$tableName = intval($_POST["tableName"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
```

```
$insert_row = $mysqli->prepare("INSERT INTO ThreeSecondExHit (Participant, duration,
ParticipantAns, corrAns, correctone, correcttwo, correctthree, Correct, eltime) VALUES(?, ?,
?, ?, ?, ?, ?, ?, ?)");
$insert_row->bind_param("iiiiiiiii", $tester, $duration, $partans, $corrans, $corrone,
$corrtwo, $corrthree,$corrYN, $eltime);
$insert_row->execute();
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeSecondExHit (Participant,
AveMinScore) VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

Third Tone Low

```
<!DOCTYPE html>
<html>
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
   top: 340px;
   left: 250px;
   visibility: hidden;}
#nextstim {position: absolute;
top: 340px;
left: 150px;}
#higher {position: absolute;
   top: 150px;
   left: 500px;}
#same {position: absolute;
   top: 250px;
   left: 500px;}
#lower {position: absolute;
```

```
    top: 350px;
    left: 500px;}
#wrong {position: absolute;
    top: 400px;
    left:820px;}
#Score {position: absolute;
    top: 380px;
    left:820px;
    font-size:300%;
    z-index: 2;}
#heading {position: absolute;
    top: 1px;
    left:350px;
    font-size: 300%}
#Answer {position: absolute;
    top: 1px;
    left:250px;
    font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page</button>
<button id = "home" onclick="nextEx()" style = "background: rgb(200,200,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 440px; left:40px;">
Practice Next Exercise</button>
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
220px; padding: 15px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen
for the Third tone of the three tone sine wave stimuli. <br> 2) click the green "Compare"
button. <br>  3) choose whether the keyboard tone is the <i>same </i> pitch as the
<u>Third</u> sine wave stimuli or <i>higher </i> or <i>lower </i>.</button>
<button id = "starter" onclick="stimPres()" style = "background: rgb(10,203,238);
width:500px; height: 50px; font-size:150%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
```

```
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%;">Next Tone</button>
<!-- Stimuli -->
<audio src= "50ms/3200010913.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/3200030816.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/3200041119.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/3200041317.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/3200051317.wav" preload= "auto" id= "sound5"></audio>
  <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
<script src="3ToneCombStrUp.js"></script>
<script>
var durchange= [200,100,70,60,50,45,40,35,30,25,20,15,10,8,7,6];
var durNo= 0;
var tableName= 1;
var i = 0;
var n = 0;
var score = 0;
var total = 0;
var correctAnswer = 1;
```

```
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var correctThree=0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
var note2= 0;
var note3= 0;
var TotScore= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
var minPercept= 0;
var perceptAdd=10;
var idin = "<?php echo $idin; ?>";
var heading= document.getElementById('heading');
if(Number.isInteger(idin/2)) {heading.innerText = "Part 2 of 4: ID Third note: " + duration +
"ms";}
   else{heading.innerText = "Part 3 of 4: ID Third note: " + duration + "ms";};
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4];
function shuffleArray(array) {
   for (var b = array.length - 1; b > 0; b--) {
       var j = Math.floor(Math.random() * (b + 1));
       var temp = array[b];
       array[b] = array[j];
       array[j] = temp;
   }

   if (TotScore==0){setTimeout(myFunction, 50);};
   return array;
}
var test256= [
document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
];
var keyboardPlay= [
document.getElementById("N1"),
```

```
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
document.getElementById("N19"),
document.getElementById("N20"),
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
function randomAnswer() {
    var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
      switch (rAns){
   case 0: if (note3 >23) {note3--; correctAnswer=0;} else {note3++; correctAnswer=2;}
break;
    case 1:  break;
   case 2:  if (note3 <1) {note3++; correctAnswer=2;} else {note3--; correctAnswer=0;}
break;
   }}
function myFunction() {
        n= array[i];
   changeStim(duration);
   note1= correctOne-1;
   note2= correctTwo-1;
   note3= correctThree-1;
   randomAnswer();
   document.getElementById('nextstim').style.visibility= 'visible';
}
function nextStim(){
   playstim();
   keypress= 0;
   startTime= new Date();
```

```
   i++;
   document.getElementById('nextstim').style.visibility= 'hidden';
   document.getElementById('compare').style.visibility= 'visible';
   if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
   keyboardPlay[note3].play();
   keypress=1;
   document.getElementById('compare').style.visibility= 'hidden';
   }}
function playstim(){
   test256[n].play();
   setTimeout (finstim, 500)}
function finstim() {silence= 0;
   }
function higher (){
   answer=2;
   checkAnswer();
   }
function same (){
   answer=1;
   checkAnswer();
   }
function lower (){
   answer=0;
   checkAnswer();
   }
function checkAnswer() {
answ = 0;
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
setTimeout(clearPic,1000);}
$.ajax({
            type: "POST",
            url: 'Study3ThirdUpAns.php',
            data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : correctOne, corrtwo : correctTwo, corrthree : correctThree, corrYN : answ, eltime :
elapsedTime, tableName : tableName},
          });
if (answ === 1) {score ++;
```

```
   if (last==1){dirCh.unshift(durNo); last= 2;}};
   if (last==0){last=1}
if (answ === 0) {score=0;
   if (last==2){dirCh.unshift(durNo); last= 0;}
   if (last==1){last=0;}
   durNo--;
   if(durNo<1){durNo=0};
   duration= durchange[durNo];
   if (TotScore>50){
      for (i=0; i<5; i++){
      perceptAdd= dirCh[i];
      minPercept= minPercept+ perceptAdd;
      }
      minPercept= Math.round(minPercept/5);
      perceptAdd= durchange[minPercept];
   $.ajax({
            type: "POST",
            url: 'Study3ThirdUpRes.php',
            data: {idin : idin, aveMinScore : perceptAdd, tableName : tableName},
          });



   var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Press OK to continue");
   if (con== true){
      if(Number.isInteger(idin/2)) {window.location.href = "Study3ThirdExt.php?idin=" +
idin;}
   else{window.location.href = "Study3ThirdDn.php?idin=" + idin;};
   }}
};
//TotScore = Math.round((answ/duration*100),0) + TotScore;
TotScore++;
Score.innerText = TotScore + "   " + dirCh;
if (score>1) {
   durNo++;
   if(durNo>15){durNo=15;};
   duration= durchange[durNo];
   score=0;
   }
document.getElementById('heading').innerText = duration + "ms";
correctAnswer = 1;
answer = 0;
if (i>4){i=0;  shuffleArray(array); myFunction();}
else {myFunction();};
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
```

```
function nextEx(){if(Number.isInteger(idin/2)) {window.location.href =
"Study3ThirdExt.php?idin=" + idin;}
   else{window.location.href = "Study3ThirdDn.php?idin=" + idin;}  }
function clearPic(){
changePic.src = "Blank.png";
}
function stimPres(){
if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
</script>
</body>
</html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
$corrone = intval($_POST["corrone"]);
$corrtwo = intval($_POST["corrtwo"]);
$corrthree = intval($_POST["corrthree"]);
$corrYN = intval($_POST["corrYN"]);
$eltime = intval($_POST["eltime"]);
$tableName = intval($_POST["tableName"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeThirdStrUp(Participant, duration,
ParticipantAns, corrAns, correctone, correcttwo, correctthree, Correct, eltime) VALUES(?, ?,
?, ?, ?, ?, ?, ?, ?)");
$insert_row->bind_param("iiiiiiiii", $tester, $duration, $partans, $corrans, $corrone,
$corrtwo, $corrthree,$corrYN, $eltime);
$insert_row->execute();
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
```

```
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeThirdStrUp(Participant,
AveMinScore) VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

Third Tone Middle

```
<!DOCTYPE html>
<html>
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
   top: 340px;
   left: 250px;
   visibility: hidden;}
#nextstim {position: absolute;
top: 340px;
left: 150px;}
#higher {position: absolute;
   top: 150px;
   left: 500px;}
#same {position: absolute;
   top: 250px;
   left: 500px;}
#lower {position: absolute;
   top: 350px;
   left: 500px;}
#wrong {position: absolute;
   top: 400px;
   left:820px;}
#Score {position: absolute;
   top: 380px;
   left:820px;
   font-size:300%;
   z-index: 2;}
#heading {position: absolute;
   top: 1px;
   left:350px;
   font-size: 300%}
#Answer {position: absolute;
   top: 1px;
   left:250px;
```

```
    font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page</button>
<button id = "home" onclick="nextEx()" style = "background: rgb(200,200,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 440px; left:40px;">
Practice Next Exercise</button>
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
220px; padding: 15px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen
for the Third tone of the three tone sine wave stimuli. <br> 2) click the green "Compare"
button. <br>  3) choose whether the keyboard tone is the <i>same </i> pitch as the
<u>Third</u> sine wave stimuli or <i>higher </i> or <i>lower </i>.</button>
<button id = "starter" onclick="stimPres()" style = "background: rgb(10,203,238);
width:500px; height: 50px; font-size:150%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%;">Next Tone</button>
<!-- Stimuli -->
<audio src= "50ms/3200010913.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/3200030816.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/3200041119.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/3200041317.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/3200051317.wav" preload= "auto" id= "sound5"></audio>
  <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
```

```
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
<script src="3ToneComb1stExt.js"></script>
<script>
var durchange= [200,100,70,60,50,45,40,35,30,25,20,15,10,8,7,6];
var durNo= 0;
var tableName= 1;
var i = 0;
var n = 0;
var score = 0;
var total = 0;
var correctAnswer = 1;
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var correctThree=0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
var note2= 0;
var note3= 0;
var TotScore= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
```

```
var minPercept= 0;
var perceptAdd=10;
var idin = "<?php echo $idin; ?>";
var heading= document.getElementById('heading');
if(Number.isInteger(idin/2)) {heading.innerText = "Part 4 of 4: ID Third note: " + duration +
"ms";}
   else{heading.innerText = "Part 1 of 4: ID Third note: " + duration + "ms";};
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4];
function shuffleArray(array) {
  for (var b = array.length - 1; b > 0; b--) {
      var j = Math.floor(Math.random() * (b + 1));
      var temp = array[b];
      array[b] = array[j];
      array[j] = temp;
   }

   if (TotScore==0){setTimeout(myFunction, 50);};
   return array;
}
var test256= [
document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
];
var keyboardPlay= [
document.getElementById("N1"),
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
```

```
document.getElementById("N19"),
document.getElementById("N20"),
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
function randomAnswer() {
    var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
      switch (rAns){
   case 0: if (note3 >23) {note3--; correctAnswer=0;} else {note3++; correctAnswer=2;}
break;
    case 1:  break;
   case 2:  if (note3 <1) {note3++; correctAnswer=2;} else {note3--; correctAnswer=0;}
break;
   }}
function myFunction() {
        n= array[i];
   changeStim(duration);
   note1= correctOne-1;
   note2= correctTwo-1;
   note3= correctThree-1;
   randomAnswer();
   document.getElementById('nextstim').style.visibility= 'visible';
}
function nextStim(){
   playstim();
   keypress= 0;
   startTime= new Date();
   i++;
   document.getElementById('nextstim').style.visibility= 'hidden';
   document.getElementById('compare').style.visibility= 'visible';
   if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
   keyboardPlay[note3].play();
   keypress=1;
   document.getElementById('compare').style.visibility= 'hidden';
   }}
function playstim(){
   test256[n].play();
   setTimeout (finstim, 500)}
function finstim() {silence= 0;
   }
```

```
function higher (){
   answer=2;
   checkAnswer();
   }
function same (){
   answer=1;
   checkAnswer();
   }
function lower (){
   answer=0;
   checkAnswer();
   }
function checkAnswer() {
answ = 0;
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
setTimeout(clearPic,1000);}
$.ajax({
               type: "POST",
               url: 'Study3ThirdIntAns.php',
               data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : correctOne, corrtwo : correctTwo, corrthree : correctThree, corrYN : answ, eltime :
elapsedTime, tableName : tableName},
             });
if (answ === 1) {score ++;
   if (last==1){dirCh.unshift(durNo); last= 2;}};
   if (last==0){last=1}
if (answ === 0) {score=0;
   if (last==2){dirCh.unshift(durNo); last= 0;}
   if (last==1){last=0;}
   durNo--;
   if(durNo<1){durNo=0};
   duration= durchange[durNo];
   if (TotScore>50){
      for (i=0; i<5; i++){
      perceptAdd= dirCh[i];
      minPercept= minPercept+ perceptAdd;
      }
      minPercept= Math.round(minPercept/5);
      perceptAdd= durchange[minPercept];
   $.ajax({
               type: "POST",
```

```
          url: 'Study3ThirdIntRes.php',
          data: {idin : idin, aveMinScore : perceptAdd, tableName : tableName},
        });



  var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Press OK to continue");
  if (con== true){
    if(Number.isInteger(idin/2)) {gohome();}
  else{window.location.href = "Study3ThirdExt.php?idin=" + idin;};
  }}
};
//TotScore = Math.round((answ/duration*100),0) + TotScore;
TotScore++;
Score.innerText = TotScore + "   " + dirCh;
if (score>1) {
  durNo++;
  if(durNo>15){durNo=15;};
  duration= durchange[durNo];
  score=0;
  }
document.getElementById('heading').innerText = duration + "ms";
correctAnswer = 1;
answer = 0;
if (i>4){i=0;  shuffleArray(array); myFunction();}
else {myFunction();};
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
function nextEx(){if(Number.isInteger(idin/2)) {gohome();}
  else{window.location.href = "Study3ThirdExt.php?idin=" + idin;};}
function clearPic(){
changePic.src = "Blank.png";
}
function stimPres(){
if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
</script>
</body>
</html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
```

```
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
$corrone = intval($_POST["corrone"]);
$corrtwo = intval($_POST["corrtwo"]);
$corrthree = intval($_POST["corrthree"]);
$corrYN = intval($_POST["corrYN"]);
$eltime = intval($_POST["eltime"]);
$tableName = intval($_POST["tableName"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeThirdInt(Participant, duration,
ParticipantAns, corrAns, correctone, correcttwo, correctthree, Correct, eltime) VALUES(?, ?,
?, ?, ?, ?, ?, ?, ?)");
$insert_row->bind_param("iiiiiiiii", $tester, $duration, $partans, $corrans, $corrone,
$corrtwo, $corrthree,$corrYN, $eltime);
$insert_row->execute();
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeThirdInt(Participant, AveMinScore)
VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

Third Tone High

```
<!DOCTYPE html>
<html>
<head>
   <meta charset= "utf-8">
   <title>Burton R Short note pitch study</title>
<style>
#compare {position: absolute;
   top: 340px;
   left: 250px;
```

```
   visibility: hidden;}
#nextstim {position: absolute;
top: 340px;
left: 150px;}
#higher {position: absolute;
   top: 150px;
   left: 500px;}
#same {position: absolute;
   top: 250px;
   left: 500px;}
#lower {position: absolute;
   top: 350px;
   left: 500px;}
#wrong {position: absolute;
   top: 400px;
   left:820px;}
#Score {position: absolute;
   top: 380px;
   left:820px;
   font-size:300%;
   z-index: 2;}
#heading {position: absolute;
   top: 1px;
   left:350px;
   font-size: 300%}
#Answer {position: absolute;
   top: 1px;
   left:250px;
   font-size: 150%}
</style>
</head>
<body onLoad = shuffleArray(array)>
<h1 id = "heading">200 ms</h1>
<?php $idin = intval($_GET["idin"]);
print $idin; ?>
<button id = "home" onclick="gohome()" style = "background: rgb(200,50,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 340px; left:40px;"> Back
to Home Page</button>
<button id = "home" onclick="nextEx()" style = "background: rgb(200,200,50);
width:130px; height: 80px; font-size:100%; position: absolute; top: 440px; left:40px;">
Practice Next Exercise</button>
<button id = "instructions" style = "background: rgb(200,200,240); width:350px; height:
220px; padding: 15px; font-size:100%; position: absolute; top: 110px; left:40px;"> 1) Listen
for the Third tone of the three tone sine wave stimuli. <br> 2) click the green "Compare"
button. <br>  3) choose whether the keyboard tone is the <i>same </i> pitch as the
<u>Third</u> sine wave stimuli or <i>higher </i> or <i>lower </i>.</button>
<button id = "starter" onclick="stimPres()" style = "background: rgb(10,203,238);
```

```
width:500px; height: 50px; font-size:150%; position: absolute; top: 470px; left:170px; z-
index: 5;">
When ready, click "next Tone" to start</button>
<img src="Blank.png" alt="Wrong" id = "wrong" style= "width: 150px; height:150px;">
<p id = "Answer"> </p>
<p id = "Score"> score</p>
<p id = "demo"> </p>
<button id = "higher" onclick="higher()" style = "background: rgb(50,203,255);
height:100px; width:230px; font-size:150%">Keyboard Higher</button>
<button id = "same" onclick="same()" style = "background: rgb(200,240,17);
height:100px; width:230px; font-size:150%">Keyboard Same</button>
<button id = "lower" onclick="lower()" style = "background: rgb(220,100,70);
height:100px; width:230px; font-size:150%">Keyboard Lower</button>
<button id = "compare" onclick="compare()" style = "background: rgb(25,240,17);
height:100px; width:230px; font-size:150%">Compare</button>
<button id = "nextstim" onclick="nextStim()" style = "background: rgb(147,103,200);
height:100px; width:230px; font-size:150%;">Next Tone</button>
<!-- Stimuli -->
<audio src= "50ms/3200010913.wav" preload= "auto" id= "sound1"></audio>
<audio src= "50ms/3200030816.wav" preload= "auto" id= "sound2"></audio>
<audio src= "50ms/3200041119.wav" preload= "auto" id= "sound3"></audio>
<audio src= "50ms/3200041317.wav" preload= "auto" id= "sound4"></audio>
<audio src= "50ms/3200051317.wav" preload= "auto" id= "sound5"></audio>
  <!-- Piano Sounds -->
<audio src= "keyboardSounds/c4.mp3" preload= "auto" id= "N1"></audio>
<audio src= "keyboardSounds/csh4.mp3" preload= "auto" id= "N2"></audio>
<audio src= "keyboardSounds/d4.mp3" preload= "auto" id= "N3"></audio>
<audio src= "keyboardSounds/dsh4.mp3" preload= "auto" id= "N4"></audio>
<audio src= "keyboardSounds/e4.mp3" preload= "auto" id= "N5"></audio>
<audio src= "keyboardSounds/f4.mp3" preload= "auto" id= "N6"></audio>
<audio src= "keyboardSounds/fsh4.mp3" preload= "auto" id= "N7"></audio>
<audio src= "keyboardSounds/g4.mp3" preload= "auto" id= "N8"></audio>
<audio src= "keyboardSounds/gsh4.mp3" preload= "auto" id= "N9"></audio>
<audio src= "keyboardSounds/a4.mp3" preload= "auto" id= "N10"></audio>
<audio src= "keyboardSounds/ash4.mp3" preload= "auto" id= "N11"></audio>
<audio src= "keyboardSounds/b4.mp3" preload= "auto" id= "N12"></audio>
<audio src= "keyboardSounds/c5.mp3" preload= "auto" id= "N13"></audio>
<audio src= "keyboardSounds/csh5.mp3" preload= "auto" id= "N14"></audio>
<audio src= "keyboardSounds/d5.mp3" preload= "auto" id= "N15"></audio>
<audio src= "keyboardSounds/dsh5.mp3" preload= "auto" id= "N16"></audio>
<audio src= "keyboardSounds/e5.mp3" preload= "auto" id= "N17"></audio>
<audio src= "keyboardSounds/f5.mp3" preload= "auto" id= "N18"></audio>
<audio src= "keyboardSounds/fsh5.mp3" preload= "auto" id= "N19"></audio>
<audio src= "keyboardSounds/g5.mp3" preload= "auto" id= "N20"></audio>
<audio src= "keyboardSounds/gsh5.mp3" preload= "auto" id= "N21"></audio>
<audio src= "keyboardSounds/a5.mp3" preload= "auto" id= "N22"></audio>
<audio src= "keyboardSounds/ash5.mp3" preload= "auto" id= "N23"></audio>
<audio src= "keyboardSounds/b5.mp3" preload= "auto" id= "N24"></audio>
```

```
<audio src= "keyboardSounds/c6.mp3" preload= "auto" id= "N25"></audio>
<script src="jquery-3.0.0.js"></script>
<script src="3ToneCombStrDn.js"></script>
<script>
var durchange= [200,100,70,60,50,45,40,35,30,25,20,15,10,8,7,6];
var durNo= 0;
var tableName= 1;
var i = 0;
var n = 0;
var score = 0;
var total = 0;
var correctAnswer = 1;
var answer = 0;
var correctOne = 0;
var correctTwo = 0;
var correctThree=0;
var answ = 0;
var keypress = 0;
var startTime= 0;
var finishTime= 0;
var elapsedTime = 0;
var silence= 0;
var note1= 0;
var note2= 0;
var note3= 0;
var TotScore= 0;
var dirChange= 0;
var dirCh=[];
var last= 2;
var minPercept= 0;
var perceptAdd=10;
var idin = "<?php echo $idin; ?>";
var heading= document.getElementById('heading');
if(Number.isInteger(idin/2)) {heading.innerText = "Part 1 of 4: ID Third note: " + duration +
"ms";}
   else{heading.innerText = "Part 4 of 4: ID Third note: " + duration + "ms";};
 /* Randomize array element order in-place.
 * Using Durstenfeld shuffle algorithm.
 */
var array = [0,1,2,3,4];
function shuffleArray(array) {
   for (var b = array.length - 1; b > 0; b--) {
       var j = Math.floor(Math.random() * (b + 1));
       var temp = array[b];
       array[b] = array[j];
       array[j] = temp;
   }
```

```
    if (TotScore==0){setTimeout(myFunction, 50);};
    return array;
}
var test256= [
document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
];
var keyboardPlay= [
document.getElementById("N1"),
document.getElementById("N2"),
document.getElementById("N3"),
document.getElementById("N4"),
document.getElementById("N5"),
document.getElementById("N6"),
document.getElementById("N7"),
document.getElementById("N8"),
document.getElementById("N9"),
document.getElementById("N10"),
document.getElementById("N11"),
document.getElementById("N12"),
document.getElementById("N13"),
document.getElementById("N14"),
document.getElementById("N15"),
document.getElementById("N16"),
document.getElementById("N17"),
document.getElementById("N18"),
document.getElementById("N19"),
document.getElementById("N20"),
document.getElementById("N21"),
document.getElementById("N22"),
document.getElementById("N23"),
document.getElementById("N24"),
document.getElementById("N25"),
];
function randomAnswer() {
     var rAns = Math.floor(Math.random() * 3);
//demo.innerText= rAns;
        switch (rAns){
    case 0: if (note3 >23) {note3--; correctAnswer=0;} else {note3++; correctAnswer=2;}
break;
     case 1:  break;
    case 2:  if (note3 <1) {note3++; correctAnswer=2;} else {note3--; correctAnswer=0;}
break;
    }}
function myFunction() {
```

```
        n= array[i];
    changeStim(duration);
    note1= correctOne-1;
    note2= correctTwo-1;
    note3= correctThree-1;
    randomAnswer();
    document.getElementById('nextstim').style.visibility= 'visible';
}
function nextStim(){
    playstim();
    keypress= 0;
    startTime= new Date();
    i++;
    document.getElementById('nextstim').style.visibility= 'hidden';
    document.getElementById('compare').style.visibility= 'visible';
    if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
var changePic = document.getElementById("wrong")
function compare () {
if (keypress== 0) {
    keyboardPlay[note3].play();
    keypress=1;
    document.getElementById('compare').style.visibility= 'hidden';
    }}
function playstim(){
    test256[n].play();
    setTimeout (finstim, 500)}
function finstim() {silence= 0;
    }
function higher (){
    answer=2;
    checkAnswer();
    }
function same (){
    answer=1;
    checkAnswer();
    }
function lower (){
    answer=0;
    checkAnswer();
    }
function checkAnswer() {
answ = 0;
finishTime= new Date();
elapsedTime= ((finishTime - startTime)/1000);
if (answer === correctAnswer){answ = 1}
if (answ === 1) {changePic.src = "Correct.png";
Answer.innerText= " ";
```

```
setTimeout(clearPic,1000);}
else if (answ === 0) {
changePic.src = "Wrong.png";
setTimeout(clearPic,1000);}
$.ajax({
            type: "POST",
            url: 'Study3ThirdDnAns.php',
            data: {idin : idin, duration : duration, pAns : answer, correctAns : correctAnswer,
corrone : correctOne, corrtwo : correctTwo, corrthree : correctThree, corrYN : answ, eltime :
elapsedTime, tableName : tableName},
          });
if (answ === 1) {score ++;
   if (last==1){dirCh.unshift(durNo); last= 2;}};
   if (last==0){last=1}
if (answ === 0) {score=0;
   if (last==2){dirCh.unshift(durNo); last= 0;}
   if (last==1){last=0;}
   durNo--;
   if(durNo<1){durNo=0};
   duration= durchange[durNo];
   if (TotScore>50){
      for (i=0; i<5; i++){
      perceptAdd= dirCh[i];
      minPercept= minPercept+ perceptAdd;
      }
      minPercept= Math.round(minPercept/5);
      perceptAdd= durchange[minPercept];
   $.ajax({
            type: "POST",
            url: 'Study3ThirdDnRes.php',
            data: {idin : idin, aveMinScore : perceptAdd, tableName : tableName},
          });



   var con= confirm("Well done! your averaged minimum score was " + perceptAdd + "ms.
Press OK to continue");
   if (con== true){
      if(Number.isInteger(idin/2)) {window.location.href = "Study3ThirdUp.php?idin=" +
idin;}
   else{gohome();}
   }}
};
//TotScore = Math.round((answ/duration*100),0) + TotScore;
TotScore++;
Score.innerText = TotScore + "   " + dirCh;
if (score>1) {
   durNo++;
```

```
    if(durNo>15){durNo=15;};
    duration= durchange[durNo];
    score=0;
    }
document.getElementById('heading').innerText = duration + "ms";
correctAnswer = 1;
answer = 0;
if (i>4){i=0;  shuffleArray(array); myFunction();}
else {myFunction();};
}
function gohome(){window.location.href = "../StudyPractice.php?idin=" + idin;}
function nextEx(){if(Number.isInteger(idin/2)) {window.location.href =
"Study3ThirdUp.php?idin=" + idin;}
    else{gohome();} }
function clearPic(){
changePic.src = "Blank.png";
}
function stimPres(){
if (total==0){ document.getElementById('starter').style.visibility= 'hidden';}
}
</script>
</body>
</html>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$duration = intval($_POST["duration"]);
$partans = intval($_POST["pAns"]);
$corrans = intval($_POST["correctAns"]);
$corrone = intval($_POST["corrone"]);
$corrtwo = intval($_POST["corrtwo"]);
$corrthree = intval($_POST["corrthree"]);
$corrYN = intval($_POST["corrYN"]);
$eltime = intval($_POST["eltime"]);
$tableName = intval($_POST["tableName"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeThirdStrDn(Participant, duration,
ParticipantAns, corrAns, correctone, correcttwo, correctthree, Correct, eltime) VALUES(?, ?,
?, ?, ?, ?, ?, ?, ?)");
$insert_row->bind_param("iiiiiiiii", $tester, $duration, $partans, $corrans, $corrone,
$corrtwo, $corrthree,$corrYN, $eltime);
$insert_row->execute();
```

```
$mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testDB";
if (isset($_POST["idin"])) {
$tester = intval($_POST["idin"]);
$aveMinScore = intval($_POST["aveMinScore"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO ThreeThirdStrDn(Participant,
AveMinScore) VALUES(?, ?)");
$insert_row->bind_param("ii", $tester, $aveMinScore);
$insert_row->execute();
$mysqli->close();
}
?>
```

Aural Training Study

Pre/Post Test

```
var choice=0;
var intNum=0;
var melNum=41;
var Num=0;
var stan= 0;
var corrAns= 0;
var noCorrect= 0;
var goesAtLevel= 0;
var stopper= 4;
var intNo=0;
var counter=0;
var score=0;
var melScore=0;
var intScore=0;
var melPlay=0;
var results = document.getElementById("results");
var startButton= document.getElementById("starter");
var explain= document.getElementById("explain");
function starter(){
  blank();
  if(stopper>1){sounds[Num].play(); stopper--;};
    startButton.style.fontSize= "25px";
    startButton.innerText="Repeat interval";
    results.style.visibility= 'visible';
```

```
        explain.style= "font-size: 70px;"
        explain.innerText= "Single Intervals";
    }
function fifth(){choice=5;checker();}
function fourth(){choice=4;checker();}
function third(){choice=3; checker();}
function second(){choice=2; checker();}
function checker(){
    if(choice==intNo){
        noCorrect++;
        goesAtLevel++;
        corrAns=1;
        results.style= "background-color:greenYellow; color: black;"
        results.innerText= "well done: "+ noCorrect + "/" + goesAtLevel;
        if(melPlay==1){melScore++;}
        else{intScore++;}
    }
    else{
        goesAtLevel++;
        results.style= "background-color:Tomato;"
        results.innerText= "Bad luck. The correct answer was a " + corrAns+ ". (score:"+
noCorrect + "/" + goesAtLevel+ ")";
    }
    $.ajax({
        type: "POST",
        url: 'Study4PretestPost.php',
        data: {idin : idin, intNo : intNo, melInt : melNum, answer : choice, correct : corrAns,},
        });
    counter++;
    melPlay=0;
    corrAns=0;
if(goesAtLevel<41){
    if(counter==5){melPlayer();melPlay=1;}
    else{intNum++; Num=intNum;
        setTimeout(blank, 1000);
        randNo(Num);
        stopper=3;
        if(counter==1 && goesAtLevel>2){startButton.style= "font-size: 70px";
startButton.innerText="Start";}
        else{setTimeout(player, 1000);}
    }}
else{postRes();
    var con= confirm("Well done! your score was " + noCorrect + "/"+ goesAtLevel + ". Press
OK to return to the home screen");
    if (con== true){window.location.href = "../homeBoard.php?idin=" + idin;}
    else{window.location.href = "../homeBoard.php?idin=" + idin;}
    }
}
```

```
function melPlayer(){
   Num=melNum; melNum++;
   setTimeout(blank, 1000);
   randNo(Num);
   stopper=4;
}
function player(){sounds[Num].play();}


function blank(){
   results.style= "background-color:Black; color:greenyellow; fontSize:40px;"
   if (counter ==5){
      counter= 0;
      explain.style= "font-size: 20px";
      if (melNum==41||melNum==45||melNum==46||melNum==50){
explain.innerText="You will now hear a melody. Identify the interval between the LAST two
tones of the melody. Press 'Start' when you are ready"
         startButton.style= "font-size: 70px";
         startButton.innerText="Start";
         results.innerText= "listen for the interval at the END"; }
      else{explain.innerText="You will now hear a melody. Identify the interval between the
FIRST two tones of the melody. Press 'Start' when you are ready"
         startButton.style= "font-size: 70px";
         startButton.innerText="Start";
         results.innerText= "listen for the interval at the START"; }
   }
   else if (counter==1){results.innerText= "Press 'Start' when ready"; stopper=4; }
   else {results.innerText= "listen for next tone"; }
}
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="utf-8">
<?php if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}; ?>

 <title>Mixed Intervals</title>
 <link rel="stylesheet" href="Study4prepost.css">
 <style>
 </style>
 </head>
 <body>

<div id= "backGround">
      <div id= "head" >
            <button id= "intro">Mixed Intervals</button>
   </div>
   <div id= "ansButtons">
      <button class="ansButton" id= "fifth" onclick= "fifth()">5th</button>
```

```
        <img src= "p5.PNG" class="ansButton" id= "fifth" onclick= "fifth()">
        <button class="ansButton" id= "fourth" onclick= "fourth()">4th</button>
        <img src= "p4.PNG"  class="ansButton" id=  "fourth" onclick= "fourth()">
        <button class="ansButton" id= "third" onclick= "third()">3rd</button>
        <img src= "Maj3.PNG"  class="ansButton" id= "third" onclick= "third()">
        <button class="ansButton" id= "second" onclick= "second()">2nd</button>
        <img src= "Maj2.PNG"  class="ansButton" id= "second" onclick= "second()">
    </div>

    <button id= "explain">When you are ready, click on "start" and you will hear two tones
that make up an interval. Select the button that matches the interval. There are 40 simple
intervals and then 10 melody intervals. Good Luck!</button>
    <button id= "results">Start by clicking "Start"</button>
    <button id= "starter" onclick= "starter()">start</button>
</div>
<audio src= "intervalsNoMel/Int32nds.wav" preload= "auto" id= "sound1"></audio>
<audio src= "intervalsNoMel/45001014L3rd.wav" preload= "auto" id= "sound2"></audio>
<audio src= "intervalsNoMel/Int45ths.wav" preload= "auto" id= "sound3"></audio>
<audio src= "intervalsNoMel/45000207L4th.wav" preload= "auto" id= "sound4"></audio>
<audio src= "intervalsNoMel/45001012L2nd.wav" preload= "auto" id= "sound5"></audio>
<audio src= "intervalsNoMel/Int33rds.wav" preload= "auto" id= "sound6"></audio>
<audio src= "intervalsNoMel/45001015L4th.wav" preload= "auto" id= "sound7"></audio>
<audio src= "intervalsNoMel/45001017L5th.wav" preload= "auto" id= "sound8"></audio>
<audio src= "intervalsNoMel/Int43rds.wav" preload= "auto" id= "sound9"></audio>
<audio src= "intervalsNoMel/Int13rds.wav" preload= "auto" id= "sound10"></audio>
<audio src= "intervalsNoMel/45001724L5th.wav" preload= "auto" id= "sound11"></audio>
<audio src= "intervalsNoMel/45000209L5th.wav" preload= "auto" id= "sound12"></audio>
<audio src= "intervalsNoMel/Int12nds.wav" preload= "auto" id= "sound13"></audio>
<audio src= "intervalsNoMel/45001219L5th.wav" preload= "auto" id= "sound14"></audio>
<audio src= "intervalsNoMel/45000204L2nd.wav" preload= "auto" id= "sound15"></audio>
<audio src= "intervalsNoMel/Int53rds.wav" preload= "auto" id= "sound16"></audio>
<audio src= "intervalsNoMel/Int52nds.wav" preload= "auto" id= "sound17"></audio>
<audio src= "intervalsNoMel/Int62nds.wav" preload= "auto" id= "sound18"></audio>
<audio src= "intervalsNoMel/45001318L4th.wav" preload= "auto" id= "sound19"></audio>
<audio src= "intervalsNoMel/Int55ths.wav" preload= "auto" id= "sound20"></audio>
<audio src= "intervalsNoMel/Int63rds.wav" preload= "auto" id= "sound21"></audio>
<audio src= "intervalsNoMel/Int64ths.wav" preload= "auto" id= "sound22"></audio>
<audio src= "intervalsNoMel/45001317L3rd.wav" preload= "auto" id= "sound23"></audio>
<audio src= "intervalsNoMel/Int42nds.wav" preload= "auto" id= "sound24"></audio>
<audio src= "intervalsNoMel/45001721L3rd.wav" preload= "auto" id= "sound25"></audio>
<audio src= "intervalsNoMel/45001722L4th.wav" preload= "auto" id= "sound26"></audio>
<audio src= "intervalsNoMel/45001315L2nd.wav" preload= "auto" id= "sound27"></audio>
<audio src= "intervalsNoMel/Int54ths.wav" preload= "auto" id= "sound28"></audio>
<audio src= "intervalsNoMel/45001719L2nd.wav" preload= "auto" id= "sound29"></audio>
<audio src= "intervalsNoMel/45001216L3rd.wav" preload= "auto" id= "sound30"></audio>
<audio src= "intervalsNoMel/45001217L4th.wav" preload= "auto" id= "sound31"></audio>
<audio src= "intervalsNoMel/Int65ths.wav" preload= "auto" id= "sound32"></audio>
<audio src= "intervalsNoMel/Int44ths.wav" preload= "auto" id= "sound33"></audio>
```

```
<audio src= "intervalsNoMel/45000206L3rd.wav" preload= "auto" id= "sound34"></audio>
<audio src= "intervalsNoMel/Int15ths.wav" preload= "auto" id= "sound35"></audio>
<audio src= "intervalsNoMel/45001320L5th.wav" preload= "auto" id= "sound36"></audio>
<audio src= "intervalsNoMel/Int34ths.wav" preload= "auto" id= "sound37"></audio>
<audio src= "intervalsNoMel/45001214L2nd.wav" preload= "auto" id= "sound38"></audio>
<audio src= "intervalsNoMel/Int35ths.wav" preload= "auto" id= "sound39"></audio>
<audio src= "intervalsNoMel/Int14ths.wav" preload= "auto" id= "sound40"></audio>
<audio src= "intervalsMelody/iLMa3b.wav" preload= "auto" id= "sound41"></audio>
<audio src= "intervalsMelody/i1Ma2b.wav" preload= "auto" id= "sound42"></audio>
<audio src= "intervalsMelody/i1p4.wav" preload= "auto" id= "sound43"></audio>
<audio src= "intervalsMelody/i1Ma3a.wav" preload= "auto" id= "sound44"></audio>
<audio src= "intervalsMelody/iLp4.wav" preload= "auto" id= "sound45"></audio>
<audio src= "intervalsMelody/ilast3a.wav" preload= "auto" id= "sound46"></audio>
<audio src= "intervalsMelody/i1Ma3b.wav" preload= "auto" id= "sound47"></audio>
<audio src= "intervalsMelody/i1p5.wav" preload= "auto" id= "sound48"></audio>
<audio src= "intervalsMelody/i1Ma2a.wav" preload= "auto" id= "sound49"></audio>
<audio src= "intervalsMelody/iLp5.wav" preload= "auto" id= "sound50"></audio>

    <script src="pretestCode.js"></script>
    <script src="jquery-3.0.0.js"></script>
<script>
var idin= <?php echo $idin; ?>;
if (idin<1) {window.location.href= "wrongUsername.html"}
var bonus= 70;
var studyNo= 9;
var sounds = [
    document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
document.getElementById("sound6"),
document.getElementById("sound7"),
document.getElementById("sound8"),
document.getElementById("sound9"),
document.getElementById("sound10"),
document.getElementById("sound11"),
document.getElementById("sound12"),
document.getElementById("sound13"),
document.getElementById("sound14"),
document.getElementById("sound15"),
document.getElementById("sound16"),
document.getElementById("sound17"),
document.getElementById("sound18"),
document.getElementById("sound19"),
document.getElementById("sound20"),
document.getElementById("sound21"),
document.getElementById("sound22"),
```

```
document.getElementById("sound23"),
document.getElementById("sound24"),
document.getElementById("sound25"),
document.getElementById("sound26"),
document.getElementById("sound27"),
document.getElementById("sound28"),
document.getElementById("sound29"),
document.getElementById("sound30"),
document.getElementById("sound31"),
document.getElementById("sound32"),
document.getElementById("sound33"),
document.getElementById("sound34"),
document.getElementById("sound35"),
document.getElementById("sound36"),
document.getElementById("sound37"),
document.getElementById("sound38"),
document.getElementById("sound39"),
document.getElementById("sound40"),
document.getElementById("sound41"),
document.getElementById("sound42"),
document.getElementById("sound43"),
document.getElementById("sound44"),
document.getElementById("sound45"),
document.getElementById("sound46"),
document.getElementById("sound47"),
document.getElementById("sound48"),
document.getElementById("sound49"),
document.getElementById("sound50"),
]
function randNo(Num){
   switch(Num){
   case 0: corrAns= "2nd"; intNo=2; break;
   case 1: corrAns= "3rd"; intNo=3; break;
   case 2: corrAns= "5th"; intNo=5; break;
   case 3: corrAns= "4th"; intNo=4; break;
   case 4: corrAns= "2nd"; intNo=2; break;
   case 5: corrAns= "3rd"; intNo=3; break;
   case 6: corrAns= "4th"; intNo=4; break;
   case 7: corrAns= "5th"; intNo=5; break;
   case 8: corrAns= "3rd"; intNo=3; break;
   case 9: corrAns= "3rd"; intNo=3; break;
   case 10: corrAns= "5th"; intNo=5; break;
   case 11: corrAns= "5th"; intNo=5; break;
   case 12: corrAns= "2nd"; intNo=2; break;
   case 13: corrAns= "5th"; intNo=5; break;
   case 14: corrAns= "2nd"; intNo=2; break;
   case 15: corrAns= "3rd"; intNo=3; break;
   case 16: corrAns= "2nd"; intNo=2; break;
```

```
    case 17: corrAns= "2nd"; intNo=2; break;
    case 18: corrAns= "4th"; intNo=4; break;
    case 19: corrAns= "5th"; intNo=5; break;
    case 20: corrAns= "3rd"; intNo=3; break;
    case 21: corrAns= "4th"; intNo=4; break;
    case 22: corrAns= "3rd"; intNo=3; break;
    case 23: corrAns= "2nd"; intNo=2; break;
    case 24: corrAns= "3rd"; intNo=3; break;
    case 25: corrAns= "4th"; intNo=4; break;
    case 26: corrAns= "2nd"; intNo=2; break;
    case 27: corrAns= "4th"; intNo=4; break;
    case 28: corrAns= "2nd"; intNo=2; break;
    case 29: corrAns= "3rd"; intNo=3; break;
    case 30: corrAns= "4th"; intNo=4; break;
    case 31: corrAns= "5th"; intNo=5; break;
    case 32: corrAns= "4th"; intNo=4; break;
    case 33: corrAns= "3rd"; intNo=3; break;
    case 34: corrAns= "5th"; intNo=5; break;
    case 35: corrAns= "5th"; intNo=5; break;
    case 36: corrAns= "4th"; intNo=4; break;
    case 37: corrAns= "2nd"; intNo=2; break;
    case 38: corrAns= "5th"; intNo=5; break;
    case 39: corrAns= "4th"; intNo=4; break;
    case 40: corrAns= "3rd"; intNo=3; break;
    case 41: corrAns= "2nd"; intNo=2; break;
    case 42: corrAns= "4th"; intNo=4; break;
    case 43: corrAns= "3rd"; intNo=3; break;
    case 44: corrAns= "4th"; intNo=4; break;
    case 45: corrAns= "3rd"; intNo=3; break;
    case 46: corrAns= "3rd"; intNo=3; break;
    case 47: corrAns= "5th"; intNo=5; break;
    case 48: corrAns= "2nd"; intNo=2; break;
    case 49: corrAns= "5th"; intNo=5; break;
    }

}
randNo(0);
function postRes(){
    $.ajax({
        type: "POST",
        url: 'Study4PretestPostFinal.php',
        data: {idin : idin, intScore : intScore, melScore : melScore},
        });
}
 </script>
 </body>
 </html>
```

```
     body{background-color:rgb(59, 110, 165)}
     #head{
       width:500px;
       position: relative;
       background-color:blueviolet;
       margin: auto;
       text-align: center;
       top:-100px;
   }
   #intro{
       background-color:black;
       color:greenyellow;
       width: 450px;
       font-size:60px;
       padding: 5px;
       margin: 10px;
   }
#backGround{
    background-color:rgb(63, 57, 153);
    position: absolute;
    top: 0%;
    bottom:0%;
    left:0%;
    right:0%;
    margin: auto;
    width:960px;
    height:640px;
}
#ansButtons{
     display: flex;
     flex-direction:column;
     position: relative;
     left:75%;
     top: -35px;
    background-color: black;
    width: 110px;
    height:440px;
}
.ansButton{
    background-color:darkgoldenrod;
    width:150px;
    height:110px;
   font-size:40px;
   margin: 5px;
}
    #fifth{ background-color:aqua;   }
   #fourth{ background-color:rgb(3, 151, 250);}
   #third{background-color:rgb(47, 6, 230);color:white;}
```

```
  #second{background-color:darkblue;color:white;}
  #duration{
   position: absolute;
      left: 50px;
      top:20px;
      background-color:black;
      color:greenyellow;
      width: 40%;
      height:15%;
      font-size:70px;
      padding: 10px;
      margin: 10px;
  }
  #explain{
      background-color:black;
      color:greenyellow;
      position: absolute;
      left: 50px;
      top:170px;
      width: 500px;
      height:200px;
      font-size:20px;
      padding:10px;
      resize:none;
      overflow: hidden;
  }
  #results{
   background-color:black;
   color:greenyellow;
   position: absolute;
   top:500px;
   left: 30px;
   font-size:40px;
      padding:10px;
      width:550px;
      min-height:140px;
      resize:none;
      visibility:visible;
}
#starter{
      position: absolute;
      top: 385px;
     left: 450px;
      background-color:black;
      color:greenyellow;
      width: 200px;
      min-height:100px;
      font-size:70px;
```

```
      resize:none;
    }
    #autoStart{
     position: absolute;
     top: 400px;
    left: 240px;
     background-color:black;
     color:greenyellow;
     width: 200px;
     min-height:70px;
     font-size:20px;
     resize:none;
     visibility:hidden;
}
    #backToBoard{
     position: absolute;
     top:640px;
    left:0px;
    right:0px;
    margin:auto;
     width: 1000px;
     height: 60px;
     visibility:visible;
}
.levels{
    position: relative;
    width:150px;
    height:50px;
    background-color:rgb(243, 132, 104);
    font-size:20px;
    left: 90px;
    right:0px;
    margin: auto;
}
    textarea{resize: none;}
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_auraltrain";
if (isset($_POST["idin"])) {
    $idin = intval($_POST["idin"]);
    $intScore = intval($_POST["intScore"]);
    $melScore = intval($_POST["melScore"]);
    // Create connection
    $mysqli = new mysqli($servername, $username, $password, $database);
    //MySqli Insert Query
    if ($mysqli->connect_error) {
```

```
        die("Connection failed: " . $mysqli->connect_error);
    }
    $updt="UPDATE Participant SET preInt= $intScore WHERE ParticipantNo= $idin";
    if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else {
echo "Error updating record: " . $mysqli->error; }
    $updt="UPDATE Participant SET preMel= $melScore WHERE ParticipantNo= $idin";
    if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else {
echo "Error updating record: " . $mysqli->error; }
    $mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_auraltrain";
if (isset($_POST["idin"])) {
$idin = intval($_POST["idin"]);
$intNo = intval($_POST["intNo"]);
$melInt = intval($_POST["melInt"]);
$answer = intval($_POST["answer"]);
$correct = intval($_POST["correct"]);
    // Create connection
    $mysqli = new mysqli($servername, $username, $password, $database);
    //MySqli Insert Query
    if ($mysqli->connect_error) {
        die("Connection failed: " . $mysqli->connect_error);
    }
    $insert_row = "INSERT INTO pretest(participantNo, intervalNo, melodicInt, answer,
correct) VALUES($idin, $intNo, $melInt, $answer, $correct)";
    if ($mysqli->query($insert_row) === TRUE) { echo "Record updated successfully"; } else
{ echo "Error updating record: " . $mysqli->error; }
    $mysqli->close();
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="utf-8">
<?php if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}; ?>

 <title>Mixed Intervals</title>
 <link rel="stylesheet" href="Study4prepost.css">
 <style>
 </style>
 </head>
 <body>
```

```
<div id= "backGround">
        <div id= "head" >
                <button id= "intro">Mixed Intervals</button>
    </div>
    <div id= "ansButtons">
        <button class="ansButton" id= "fifth" onclick= "fifth()">5th</button>
        <img src= "p5.PNG" class="ansButton" id= "fifth" onclick= "fifth()">
        <button class="ansButton" id= "fourth" onclick= "fourth()">4th</button>
        <img src= "p4.PNG"  class="ansButton" id=  "fourth" onclick= "fourth()">
        <button class="ansButton" id= "third" onclick= "third()">3rd</button>
        <img src= "Maj3.PNG"  class="ansButton" id= "third" onclick= "third()">
        <button class="ansButton" id= "second" onclick= "second()">2nd</button>
        <img src= "Maj2.PNG"  class="ansButton" id= "second" onclick= "second()">
    </div>

    <button id= "explain">When you are ready, click on "start" and you will hear two tones
that make up an interval. Select the button that matches the interval. There are 40 simple
intervals and then 10 melody intervals. Good Luck!</button>
    <button id= "results">Start by clicking "Start"</button>
    <button id= "starter" onclick= "starter()">start</button>
</div>
<audio src= "intervalsNoMel/Int32nds.wav" preload= "auto" id= "sound1"></audio>
<audio src= "intervalsNoMel/45001014L3rd.wav" preload= "auto" id= "sound2"></audio>
<audio src= "intervalsNoMel/Int45ths.wav" preload= "auto" id= "sound3"></audio>
<audio src= "intervalsNoMel/45000207L4th.wav" preload= "auto" id= "sound4"></audio>
<audio src= "intervalsNoMel/45001012L2nd.wav" preload= "auto" id= "sound5"></audio>
<audio src= "intervalsNoMel/Int33rds.wav" preload= "auto" id= "sound6"></audio>
<audio src= "intervalsNoMel/45001015L4th.wav" preload= "auto" id= "sound7"></audio>
<audio src= "intervalsNoMel/45001017L5th.wav" preload= "auto" id= "sound8"></audio>
<audio src= "intervalsNoMel/Int43rds.wav" preload= "auto" id= "sound9"></audio>
<audio src= "intervalsNoMel/Int13rds.wav" preload= "auto" id= "sound10"></audio>
<audio src= "intervalsNoMel/45001724L5th.wav" preload= "auto" id= "sound11"></audio>
<audio src= "intervalsNoMel/45000209L5th.wav" preload= "auto" id= "sound12"></audio>
<audio src= "intervalsNoMel/Int12nds.wav" preload= "auto" id= "sound13"></audio>
<audio src= "intervalsNoMel/45001219L5th.wav" preload= "auto" id= "sound14"></audio>
<audio src= "intervalsNoMel/45000204L2nd.wav" preload= "auto" id= "sound15"></audio>
<audio src= "intervalsNoMel/Int53rds.wav" preload= "auto" id= "sound16"></audio>
<audio src= "intervalsNoMel/Int52nds.wav" preload= "auto" id= "sound17"></audio>
<audio src= "intervalsNoMel/Int62nds.wav" preload= "auto" id= "sound18"></audio>
<audio src= "intervalsNoMel/45001318L4th.wav" preload= "auto" id= "sound19"></audio>
<audio src= "intervalsNoMel/Int55ths.wav" preload= "auto" id= "sound20"></audio>
<audio src= "intervalsNoMel/Int63rds.wav" preload= "auto" id= "sound21"></audio>
<audio src= "intervalsNoMel/Int64ths.wav" preload= "auto" id= "sound22"></audio>
<audio src= "intervalsNoMel/45001317L3rd.wav" preload= "auto" id= "sound23"></audio>
<audio src= "intervalsNoMel/Int42nds.wav" preload= "auto" id= "sound24"></audio>
<audio src= "intervalsNoMel/45001721L3rd.wav" preload= "auto" id= "sound25"></audio>
<audio src= "intervalsNoMel/45001722L4th.wav" preload= "auto" id= "sound26"></audio>
<audio src= "intervalsNoMel/45001315L2nd.wav" preload= "auto" id= "sound27"></audio>
```

```
<audio src= "intervalsNoMel/Int54ths.wav" preload= "auto" id= "sound28"></audio>
<audio src= "intervalsNoMel/45001719L2nd.wav" preload= "auto" id= "sound29"></audio>
<audio src= "intervalsNoMel/45001216L3rd.wav" preload= "auto" id= "sound30"></audio>
<audio src= "intervalsNoMel/45001217L4th.wav" preload= "auto" id= "sound31"></audio>
<audio src= "intervalsNoMel/Int65ths.wav" preload= "auto" id= "sound32"></audio>
<audio src= "intervalsNoMel/Int44ths.wav" preload= "auto" id= "sound33"></audio>
<audio src= "intervalsNoMel/45000206L3rd.wav" preload= "auto" id= "sound34"></audio>
<audio src= "intervalsNoMel/Int15ths.wav" preload= "auto" id= "sound35"></audio>
<audio src= "intervalsNoMel/45001320L5th.wav" preload= "auto" id= "sound36"></audio>
<audio src= "intervalsNoMel/Int34ths.wav" preload= "auto" id= "sound37"></audio>
<audio src= "intervalsNoMel/45001214L2nd.wav" preload= "auto" id= "sound38"></audio>
<audio src= "intervalsNoMel/Int35ths.wav" preload= "auto" id= "sound39"></audio>
<audio src= "intervalsNoMel/Int14ths.wav" preload= "auto" id= "sound40"></audio>
<audio src= "intervalsMelody/iLMa3b.wav" preload= "auto" id= "sound41"></audio>
<audio src= "intervalsMelody/i1Ma2b.wav" preload= "auto" id= "sound42"></audio>
<audio src= "intervalsMelody/i1p4.wav" preload= "auto" id= "sound43"></audio>
<audio src= "intervalsMelody/i1Ma3a.wav" preload= "auto" id= "sound44"></audio>
<audio src= "intervalsMelody/iLp4.wav" preload= "auto" id= "sound45"></audio>
<audio src= "intervalsMelody/ilast3a.wav" preload= "auto" id= "sound46"></audio>
<audio src= "intervalsMelody/i1Ma3b.wav" preload= "auto" id= "sound47"></audio>
<audio src= "intervalsMelody/i1p5.wav" preload= "auto" id= "sound48"></audio>
<audio src= "intervalsMelody/i1Ma2a.wav" preload= "auto" id= "sound49"></audio>
<audio src= "intervalsMelody/iLp5.wav" preload= "auto" id= "sound50"></audio>

   <script src="posttestCode.js"></script>
   <script src="jquery-3.0.0.js"></script>
<script>
var idin= <?php echo $idin; ?>;
if (idin<1) {window.location.href= "wrongUsername.html"}
var bonus= 70;
var studyNo= 9;
var sounds = [
   document.getElementById("sound1"),
document.getElementById("sound2"),
document.getElementById("sound3"),
document.getElementById("sound4"),
document.getElementById("sound5"),
document.getElementById("sound6"),
document.getElementById("sound7"),
document.getElementById("sound8"),
document.getElementById("sound9"),
document.getElementById("sound10"),
document.getElementById("sound11"),
document.getElementById("sound12"),
document.getElementById("sound13"),
document.getElementById("sound14"),
document.getElementById("sound15"),
document.getElementById("sound16"),
```

```
document.getElementById("sound17"),
document.getElementById("sound18"),
document.getElementById("sound19"),
document.getElementById("sound20"),
document.getElementById("sound21"),
document.getElementById("sound22"),
document.getElementById("sound23"),
document.getElementById("sound24"),
document.getElementById("sound25"),
document.getElementById("sound26"),
document.getElementById("sound27"),
document.getElementById("sound28"),
document.getElementById("sound29"),
document.getElementById("sound30"),
document.getElementById("sound31"),
document.getElementById("sound32"),
document.getElementById("sound33"),
document.getElementById("sound34"),
document.getElementById("sound35"),
document.getElementById("sound36"),
document.getElementById("sound37"),
document.getElementById("sound38"),
document.getElementById("sound39"),
document.getElementById("sound40"),
document.getElementById("sound41"),
document.getElementById("sound42"),
document.getElementById("sound43"),
document.getElementById("sound44"),
document.getElementById("sound45"),
document.getElementById("sound46"),
document.getElementById("sound47"),
document.getElementById("sound48"),
document.getElementById("sound49"),
document.getElementById("sound50"),
]
function randNo(Num){
   switch(Num){
   case 0: corrAns= "2nd"; intNo=2; break;
   case 1: corrAns= "3rd"; intNo=3; break;
   case 2: corrAns= "5th"; intNo=5; break;
   case 3: corrAns= "4th"; intNo=4; break;
   case 4: corrAns= "2nd"; intNo=2; break;
   case 5: corrAns= "3rd"; intNo=3; break;
   case 6: corrAns= "4th"; intNo=4; break;
   case 7: corrAns= "5th"; intNo=5; break;
   case 8: corrAns= "3rd"; intNo=3; break;
   case 9: corrAns= "3rd"; intNo=3; break;
   case 10: corrAns= "5th"; intNo=5; break;
```

```
    case 11: corrAns= "5th"; intNo=5; break;
    case 12: corrAns= "2nd"; intNo=2; break;
    case 13: corrAns= "5th"; intNo=5; break;
    case 14: corrAns= "2nd"; intNo=2; break;
    case 15: corrAns= "3rd"; intNo=3; break;
    case 16: corrAns= "2nd"; intNo=2; break;
    case 17: corrAns= "2nd"; intNo=2; break;
    case 18: corrAns= "4th"; intNo=4; break;
    case 19: corrAns= "5th"; intNo=5; break;
    case 20: corrAns= "3rd"; intNo=3; break;
    case 21: corrAns= "4th"; intNo=4; break;
    case 22: corrAns= "3rd"; intNo=3; break;
    case 23: corrAns= "2nd"; intNo=2; break;
    case 24: corrAns= "3rd"; intNo=3; break;
    case 25: corrAns= "4th"; intNo=4; break;
    case 26: corrAns= "2nd"; intNo=2; break;
    case 27: corrAns= "4th"; intNo=4; break;
    case 28: corrAns= "2nd"; intNo=2; break;
    case 29: corrAns= "3rd"; intNo=3; break;
    case 30: corrAns= "4th"; intNo=4; break;
    case 31: corrAns= "5th"; intNo=5; break;
    case 32: corrAns= "4th"; intNo=4; break;
    case 33: corrAns= "3rd"; intNo=3; break;
    case 34: corrAns= "5th"; intNo=5; break;
    case 35: corrAns= "5th"; intNo=5; break;
    case 36: corrAns= "4th"; intNo=4; break;
    case 37: corrAns= "2nd"; intNo=2; break;
    case 38: corrAns= "5th"; intNo=5; break;
    case 39: corrAns= "4th"; intNo=4; break;
    case 40: corrAns= "3rd"; intNo=3; break;
    case 41: corrAns= "2nd"; intNo=2; break;
    case 42: corrAns= "4th"; intNo=4; break;
    case 43: corrAns= "3rd"; intNo=3; break;
    case 44: corrAns= "4th"; intNo=4; break;
    case 45: corrAns= "3rd"; intNo=3; break;
    case 46: corrAns= "3rd"; intNo=3; break;
    case 47: corrAns= "5th"; intNo=5; break;
    case 48: corrAns= "2nd"; intNo=2; break;
    case 49: corrAns= "5th"; intNo=5; break;
    }

}
randNo(39);
function postRes(){
    $.ajax({
        type: "POST",
        url: 'Study4PosttestPostFinal.php',
        data: {idin : idin, intScore : intScore, melScore : melScore},
```

```
        });
}
 </script>
 </body>
 </html>
var choice=0;
var intNum=39;
var melNum=48;
var melNum2=48;
var Num=39;
var stan= 0;
var corrAns= 0;
var noCorrect= 0;
var goesAtLevel= 0;
var stopper= 4;
var intNo=0;
var counter=0;
var score=0;
var melScore=0;
var intScore=0;
var melPlay=0;
var results = document.getElementById("results");
var startButton= document.getElementById("starter");
var explain= document.getElementById("explain");
function starter(){
    blank();
    if(stopper>1){sounds[Num].play(); stopper--;};
        startButton.style.fontSize= "25px";
        startButton.innerText="Repeat interval";
        results.style.visibility= 'visible';
        if (counter > 0){
            explain.style= "font-size: 70px;"
            explain.innerText= "Single Intervals";}
    }
function fifth(){choice=5;checker();}
function fourth(){choice=4;checker();}
function third(){choice=3; checker();}
function second(){choice=2; checker();}
function checker(){
    if (counter < 1){
        explain.style= "font-size: 70px;"
        explain.innerText= "Single Intervals";}
    if(choice==intNo){
        noCorrect++;
        goesAtLevel++;
        corrAns=1;
        results.style= "background-color:greenYellow; color: black;"
        results.innerText= "well done: "+ noCorrect + "/" + goesAtLevel;
```

```
        if(melPlay==1){melScore++;}
        else{intScore++;}
    }
    else{
        goesAtLevel++;
        results.style= "background-color:Tomato;"
        results.innerText= "Bad luck. The correct answer was a " + corrAns+ ". (score:"+
noCorrect + "/" + goesAtLevel+ ")";
    }
    $.ajax({
        type: "POST",
        url: 'Study4PosttestPost.php',
        data: {idin : idin, intNo : intNo, melInt : melNum, answer : choice, correct : corrAns,},
        });
    counter++;
    melPlay=0;
    corrAns=0;
if(goesAtLevel<41){
    if(counter==5){melPlayer();melPlay=1;}
    else{intNum--; Num=intNum;
        setTimeout(blank, 1000);
        randNo(Num);
        stopper=3;
        if(counter==1 && goesAtLevel>2){startButton.style= "font-size: 70px";
startButton.innerText="Start";}
        else{setTimeout(player, 1000);}
    }}
else{postRes();
    var con= confirm("Well done! your score was " + noCorrect + "/"+ goesAtLevel + ". Press
OK to return to the home screen");
    if (con== true){window.location.href = "../homeBoard.php?idin=" + idin;}
    else{window.location.href = "../homeBoard.php?idin=" + idin;}
    }
}
function melPlayer(){
    Num=melNum; melNum--; melNum2= melNum+2;
    setTimeout(blank, 1000);
    randNo(Num);
    stopper=4;
}
function player(){sounds[Num].play();}


function blank(){
    results.style= "background-color:Black; color:greenyellow; fontSize:40px;"
    if (counter ==5){
        counter= 0;
        explain.style= "font-size: 20px";
```

```
      if (melNum2==41||melNum2==45||melNum2==46||melNum2==50){
explain.innerText="You will now hear a melody. Identify the interval between the LAST
TWO TONES of the melody. Press 'Start' when you are ready"
         startButton.style= "font-size: 70px";
         startButton.innerText="Start";
         results.innerText= "listen for the interval at the END"; }
      else{explain.innerText="You will now hear a melody. Identify the interval between the
FIRST TWO TONES of the melody. Press 'Start' when you are ready"
         startButton.style= "font-size: 70px";
         startButton.innerText="Start";
         results.innerText= "listen for the interval at the START"; }
   }
   else if (counter==1){results.innerText= "Press 'Start' when ready"; stopper=4; }
   else {results.innerText= "listen for next tone"; }
}
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_auraltrain";
if (isset($_POST["idin"])) {
$idin = intval($_POST["idin"]);
$intNo = intval($_POST["intNo"]);
$melInt = intval($_POST["melInt"]);
$answer = intval($_POST["answer"]);
$correct = intval($_POST["correct"]);
   // Create connection
   $mysqli = new mysqli($servername, $username, $password, $database);
   //MySqli Insert Query
   if ($mysqli->connect_error) {
      die("Connection failed: " . $mysqli->connect_error);
   }
   $insert_row = "INSERT INTO posttest(participantNo, intervalNo, melodicInt, answer,
correct) VALUES($idin, $intNo, $melInt, $answer, $correct)";
   if ($mysqli->query($insert_row) === TRUE) { echo "Record updated successfully"; } else
{ echo "Error updating record: " . $mysqli->error; }
   $mysqli->close();
}
?>
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_auraltrain";
if (isset($_POST["idin"])) {
   $idin = intval($_POST["idin"]);
   $intScore = intval($_POST["intScore"]);
   $melScore = intval($_POST["melScore"]);
```

```
   // Create connection
   $mysqli = new mysqli($servername, $username, $password, $database);
   //MySqli Insert Query
   if ($mysqli->connect_error) {
      die("Connection failed: " . $mysqli->connect_error);
   }
   $updt="UPDATE Participant SET postInt= $intScore WHERE ParticipantNo= $idin";
   if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else {
echo "Error updating record: " . $mysqli->error; }
   $updt="UPDATE Participant SET postMel= $melScore WHERE ParticipantNo= $idin";
   if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else {
echo "Error updating record: " . $mysqli->error; }
   $mysqli->close();
}
?>
```

Main Training Program

```
<!DOCTYPE html>
<html>
<meta charset="utf-8" />
<?php
if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testdb";
if (isset($_GET["idin"])) {
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
if ($mysqli->connect_error) {
   die("Connection failed: " . $mysqli->connect_error);
}
$updt="UPDATE participant SET topScore= 0, single2= 1000, single3= 1000, single4=
1000,double2= 1000, double3= 1000, double4= 1000, mixed2= 1000, mixed3= 1000,
mixed4= 1000 WHERE participantNo= 1";
if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else { echo
"Error updating record: " . $mysqli->error; }
$sql="SELECT * FROM participant WHERE participantNo=$idin";
echo "User Name and Id no:";
if ($result=mysqli_query($mysqli,$sql))
  {
  // Fetch one and one row
  while ($row=mysqli_fetch_row($result))
   {
   printf ("%s (%s)\n",$row[2],$row[7]);
 $idinput= $row[7];
$topScoreD= $row[13];
$single2D= $row[14];
$single3D= $row[15];
```

```
$single4D= $row[16];
$double2D= $row[17];
$double3D= $row[18];
$double4D= $row[19];
$mixed2D= $row[20];
$mixed3D= $row[21];
$mixed4D= $row[22];
    }
     $rowcount= $result->num_rows;
echo $double3D, $mixed2D;
  // Free result set
  mysqli_free_result($result);
}
 $mysqli->close();
if($rowcount<1){echo "Wrong Username";}
}
?>
<head>
     <meta charset= "utf-8">
     <title>Burton R Identify Short Notes</title>
<style>
.buttons1{
position: relative;
border-radius:5px;
width:300px;
height: 75px;
font-size:200%;
background-color: rgb(10,180,210);
box-shadow: 5px 5px 2px grey;
left: 5px;}
h1 {font-size:250%;}
.buttons2{
position: relative;
border-radius:5px;
width:300px;
height: 75px;
font-size:200%;
background-color: rgb(10,180,210);
box-shadow: 5px 5px 2px grey;
left: 15px;}
.borders{
     display: flex;
     flex-direction:column;
background-color: rgb(150, 209, 67);
border:30px;
border-radius:25px;
border: 2px solid #73AD21;
width: 360px;
```

```
margin: 20px;
margin-top:70px;
}
#centreH{
text-align: center;
font-size:190%;
}
#div1{
background-color: lightskyblue;
margin: 0px;
border-radius:5px;
}
#title1{
position: relative;
text-align: center;
padding: 10px;
}
#leadInstructions{
position: absolute;
font-size: 300%;
left:550px;
background-color: rgb(240, 240, 240);
margin: auto;
}
#divCenter{
    display: flex;
    flex-direction:row;
width: 1200px;
margin: auto;
background-color: rgb(240, 240, 240);
}
#titleDiv{
width: 600px;
margin: auto;
border-radius:25px;
background-color: white;
}
body {background-color: lightskyblue;}
#explanation {
    position: relative;
    background-image: url('../herringbone.png');
    padding: 20px;
    width: 800px;
    margin: auto;
    top: -100px;
}
#colorKey {
    position: relative;
```

```
background-color: rgb(150, 209, 67);
border-radius:25px;
border: 2px solid #73AD21;
width: 190px;
margin: 10px;
text-align: auto;
padding: 20px;
top: -420px;
left: -260px;
}
#video {
    position: relative;
    top: -200px;
}
#videoTrain {
    font-size: 130%;
    font: bold;
}
</style>
</head>
<body>
<div id= "titleDiv">
<h1 id= "title1">Musical Interval Training</h1>
</div>
<div id= "div1" class= "borders1">
<div id= "divCenter">
    <div id= "leadInstructions">Choose your level</div>
<div class = "borders" style = " height: 400px; position: relative; top: 0px;">
<h1 id = "centreH"> <u>Single Start-note</u></h1>
<button id = "single2" class= "buttons1" onclick= "singleTwo()" style = "margin : 5px; top:
5px;background-color:rgb(228,80,60);">Two Intervals</button>
<button id = "single3" class= "buttons1" onclick= "singleThree()" style = "margin : 5px; top:
5px;">Three Intervals</button>
<button id = "single4" class= "buttons1" onclick= "singleFour()" style = "margin : 5px; top:
5px;">Four Intervals</button>
</div>
<div class = "borders" style = " height: 400px; position: relative; top: 0px;">
        <h1 id = "centreH"> <u>Double Start-note</u></h1>

        <button id = "double2" class= "buttons1" onclick= "doubleTwo()" style = "margin :
5px; top: 5px;">Two Intervals</button>
        <button id = "double3" class= "buttons1" onclick= "doubleThree()" style = "margin :
5px; top: 5px;">Three Intervals</button>
        <button id = "double4" class= "buttons1" onclick= "doubleFour()" style = "margin :
5px; top: 5px;">Four Intervals</button>

        </div>
<div id = "div2" class = "borders" style = " height: 400px;">
```

```
<h1 id = "centreH"> <u>mixed Start-notes </u></h1>
<button id = "mixed2" class= "buttons2" onclick= "mixedTwo()" style = "margin :
5px;">Two Intervals </button>
<button id = "mixed3" class= "buttons2" onclick= "mixedThree()" style = "margin : 5px;
">Three Intervals </button>
<button id = "mixed4" class= "buttons2" onclick= "mixedFour()" style = "margin : 5px;
">Four intervals </button>
</div>
</div>
<p id= "check"></p>
</div>
<button id= "saveClose" onclick= "goHome()"  style= "position: fixed; right: 100px; top:
100px; width: 200px; height: 100px; font-size: 30px; background-color: rgb(237, 41,
57);">Back to Home Page</button>
<div id= "explanation">
<h2>Instructions: Interval Recognition </h2>
<p>
Like many other aural training programs, this program starts with choosing between a perfect
4th and 5th. However, in order to make the exercise more interesting (and also to include
temporal occlusion training, the duration of the tones are decreased while staying with the
same two intervals. <br> <br>
<p id= "videoTrain">
If you are not too sure how intervals work, please see the video (below - scroll down). Please
note: I created the video for my Year 7 music class, so it may sound a little
'different.'<br></p>
The board (above) shows a variety of exercises that you can choose from. The levels start
with the simplest on the top left corner (in red), and progress to the heardest in the bottom
right. Difficulty is increase by increasing the number of intervals and number of different
starting notes. The most difficult exercise involves differentiating between seconds, thirds,
fourths and fifths using random starting tones.
<br>
</p>
<h2>Rationale: Interval Recognition </h2>
<P> Interval recognition is a gateway skill to notation. Without this skill it is impossible to
audiate (hear in your head) written music or to sing written music without first having to hear
other people singing it.
</p>
<h2> Musical Intervals Explanation Video</h2>
<Div id= "colorKey">
<h3>Color Key</h3>
<button class= "colorKeyCl" style= "background-color: rgb(10,180,210); width:
170px;">Not Started</button>
<button class= "colorKeyCl" style= "background-color: Moccasin; width:
170px;">Started</button>
<button class= "colorKeyCl" style= "background-color: lightGreen; width: 170px;">Going
well</button>
<button class= "colorKeyCl" style= "background-color: Green; width: 170px; color:
white;">Completed</button>
```

```
</div>
<video id= "video" width="320" height="240" controls>
  <source src="IntervalTrainingVideo.mp4" type="video/mp4">
Your browser does not support the video tag.
</video>
</div>
<script>
var idin= <?php echo $idin; ?>;
var single2= <?php echo $single2D; ?>;
var single3= <?php echo $single3D; ?>;
var single4= <?php echo $single4D; ?>;
var double2= <?php echo $double2D; ?>;
var double3= <?php echo $double3D; ?>;
var double4= <?php echo $double4D; ?>;
var mixed2= <?php echo $mixed2D; ?>;
var mixed3= <?php echo $mixed3D; ?>;
var mixed4= <?php echo $mixed4D; ?>;
if (idin<1) {window.location.href= "wrongUsername.html"}
check.innerText= idin;
var btnSingle2= document.getElementById("single2")
var btnSingle3= document.getElementById("single3")
var btnSingle4= document.getElementById("single4")
var btnDbl2= document.getElementById("double2")
var btnDbl3= document.getElementById("double3")
var btnDbl4= document.getElementById("double4")
var btnmixed2= document.getElementById("mixed2")
var btnmixed3= document.getElementById("mixed3")
var btnmixed4= document.getElementById("mixed4")
if (single2>900){
 btnSingle2.style= "width:300px; height:100px; background-color: red; z-index: 2"
 btnSingle2.innerHTML = "Please start with this button first"
}
if (single2<1000){btnSingle2.innerText="Sngl2int: " + single2 + "ms";
btnSingle2.style="font-size:20px;";} ;
if(single2<30) {btnSingle2.style="background-color:Green;font-size:20px; color: white;"}
if(single2<100){if(single2>30){btnSingle2.style="background-color:LightGreen;font-
size:20px;"}}
if(single2<1000){if(single2>100){btnSingle2.style="background-color:Moccasin;font-
size:20px;"}}
if (single3<1000){btnSingle3.innerText="Sngl3int: " + single3 + "ms";
btnSingle3.style="font-size:20px;"}
if(single3<30) {btnSingle3.style="background-color:Green;font-size:20px; color: white;"}
if(single3<100){if(single3>30){btnSingle3.style="background-color:LightGreen;font-
size:20px;"}}
if(single3<1000){if(single3>100){btnSingle3.style="background-color:Moccasin;font-
size:20px;"}}
if (single4<1000){btnSingle4.innerText="Sngl4int: " + single4 + "ms";
btnSingle4.style="font-size:20px;"}
```

```
if(single4<30) {btnSingle4.style="background-color:Green;font-size:20px; color: white;"}
if(single4<100){if(single4>30){btnSingle4.style="background-color:LightGreen;font-
size:20px;"}}
if(single4<1000){if(single4>100){btnSingle4.style="background-color:Moccasin;font-
size:20px;"}}
if (double2<1000){btnDbl2.innerText="Dbl2int: " + double2 + "ms"; btnDbl2.style="font-
size:20px;"}
if(double2<30) {btnDbl2.style="background-color:Green;font-size:20px; color: white;"}
if(double2<100){if(double2>30){btnDbl2.style="background-color:LightGreen;font-
size:20px;"}}
if(double2<1000){if(double2>100){btnDbl2.style="background-color:Moccasin;font-
size:20px;"}}
if (double3<1000){btnDbl3.innerText="Dbl3int: " + double3 + "ms"; btnDbl3.style="font-
size:20px;"}
if(double3<30) {btnDbl3.style="background-color:Green;font-size:20px; color: white;"}
if(double3<100){if(double3>30){btnDbl3.style="background-color:LightGreen;font-
size:20px;"}}
if(double3<1000){if(double3>100){btnDbl3.style="background-color:Moccasin;font-
size:20px;"}}
if (double4<1000){btnDbl4.innerText="Dbl4int: " + double4 + "ms"; btnDbl4.style="font-
size:20px;"}
if(double4<30) {btnDbl4.style="background-color:Green;font-size:20px; color: white;"}
if(double4<100){if(double4>30){btnDbl4.style="background-color:LightGreen;font-
size:20px;"}}
if(double4<1000){if(double4>100){btnDbl4.style="background-color:Moccasin;font-
size:20px;"}}
if (mixed2<1000){btnmixed2.innerText="mixed2int: " + mixed2 + "ms";
btnmixed2.style="font-size:20px;"}
if(mixed2<30) {btnmixed2.style="background-color:Green;font-size:20px; color: white;"}
if(mixed2<100){if(mixed2>30){btnmixed2.style="background-color:LightGreen;font-
size:20px;"}}
if(mixed2<1000){if(mixed2>100){btnmixed2.style="background-color:Moccasin;font-
size:20px;"}}
if (mixed3<1000){btnmixed3.innerText="mixed3int: " + mixed3 + "ms";
btnmixed3.style="font-size:20px;"}
if(mixed3<30) {btnmixed3.style="background-color:Green;font-size:20px; color: white;"}
if(mixed3<100){if(mixed3>30){btnmixed3.style="background-color:LightGreen;font-
size:20px;"}}
if(mixed3<1000){if(mixed3>100){btnmixed3.style="background-color:Moccasin;font-
size:20px;"}}
if (mixed4<1000){btnmixed4.innerText="mixed4int: " + mixed4 + "ms";
btnmixed4.style="font-size:20px;"}
if(mixed4<30) {btnmixed4.style="background-color:Green;font-size:20px; color: white;"}
if(mixed4<100){if(mixed4>30){btnmixed4.style="background-color:LightGreen;font-
size:20px;"}}
if(mixed4<1000){if(mixed4>100){btnmixed4.style="background-color:Moccasin;font-
size:20px;"}}
function singleTwo(){window.location.href = "intervalsOneTone45.php?idin="+idin};
```

```
function singleThree(){window.location.href = "intervalsOneTone245.php?idin="+idin};
function singleFour(){window.location.href = "intervalsOneTone.php?idin="+idin};
function doubleTwo(){window.location.href = "intervalsTwoTone45.php?idin="+idin};
function doubleThree(){window.location.href = "intervalsTwoTone245.php?idin="+idin};
function doubleFour(){window.location.href = "intervalsTwoTone.php?idin="+idin};
function mixedTwo(){window.location.href = "intervalsMixed45.php?idin="+idin};
function mixedThree(){window.location.href = "intervalsMixed245.php?idin="+idin};
function mixedFour(){window.location.href = "intervalsMixed.php?idin="+idin};
function postTest(){window.location.href = "PrePostTest/Posttest.php?idin="+idin}
function goHome(){window.location.href =  "../HomePage.php?idin="+idin}
</script>
</body>
</html>
<!DOCTYPE html>
<html lang="en">
 <meta charset="utf-8" />
 <title>Mixed Intervals</title>
<?php
if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testdb";
$mysqli = new mysqli($servername, $username, $password, $database);
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}
$sql="SELECT * FROM participant WHERE participantNo=$idin";
if ($result=mysqli_query($mysqli,$sql))
  {
  // Fetch one and one row
  while ($row=mysqli_fetch_row($result))
    {
        $recordDur= $row[20];
    }
        mysqli_free_result($result);
    }
?>
 <head>
 <link rel="stylesheet" href="Study4and5.css">
 <style>
 </style>
 </head>
 <body>

<div id= "backGround">
        <div id= "head" >
                <button id= "intro">Mixed Intervals</button>
```

```
    </div>
    <div id= "ansButtons">
        <button class="ansButton" id= "fifth" onclick= "fifth()">5th</button>
        <button class="ansButton" id= "fourth" onclick= "fourth()">4th</button>
        <button class="ansButton" id= "third" onclick= "third()">3rd</button>
        <button class="ansButton" id= "second" onclick= "second()">2nd</button>
    </div>
    <button id= "duration">200ms</button>
    <button id= "explain">When you are ready, click on "start" and you will hear two tones
that make up an interval. Select the button that matches the interval. If you get 5 right in a
row, you will go to the next level. In "Easy" mode you can listen to the tones (the bars on the
right) and also repeat the interval. Good Luck!</button>
    <button id= "results">Start by clicking "Start"</button>
    <button id= "starter" onclick= "starter()">start</button>
    <button id= "autoStart" onclick= "auto()">click for automatic starting</button>
</div>
<button id= "saveClose" onclick= "Home()" >Save and Close</button>
<audio src= "Stimuli/49991012.wav" preload= "auto" id= "int2fromA3"></audio>
    <audio src= "Stimuli/49991014.wav" preload= "auto" id= "int3fromA3"></audio>
    <audio src= "Stimuli/49991015.wav" preload= "auto" id= "int4fromA3"></audio>
    <audio src= "Stimuli/49991017.wav" preload= "auto" id= "int5fromA3"></audio>
    <audio src= "Stimuli/49991315.wav" preload= "auto" id= "int2fromC4"></audio>
    <audio src= "Stimuli/49991317.wav" preload= "auto" id= "int3fromC4"></audio>
    <audio src= "Stimuli/49991318.wav" preload= "auto" id= "int4fromC4"></audio>
    <audio src= "Stimuli/49991320.wav" preload= "auto" id= "int5fromC4"></audio>
    <audio src= "Stimuli/49990204.wav" preload= "auto" id= "int2from2"></audio>
    <audio src= "Stimuli/49990206.wav" preload= "auto" id= "int3from2"></audio>
    <audio src= "Stimuli/49990207.wav" preload= "auto" id= "int4from2"></audio>
    <audio src= "Stimuli/49990209.wav" preload= "auto" id= "int5from2"></audio>
    <audio src= "Stimuli/49990709.wav" preload= "auto" id= "int2from7"></audio>
    <audio src= "Stimuli/49990711.wav" preload= "auto" id= "int3from7"></audio>
    <audio src= "Stimuli/49990712.wav" preload= "auto" id= "int4from7"></audio>
    <audio src= "Stimuli/49990714.wav" preload= "auto" id= "int5from7"></audio>
    <audio src= "Stimuli/49991719.wav" preload= "auto" id= "int2from17"></audio>
    <audio src= "Stimuli/49991721.wav" preload= "auto" id= "int3from17"></audio>
    <audio src= "Stimuli/49991722.wav" preload= "auto" id= "int4from17"></audio>
    <audio src= "Stimuli/49991724.wav" preload= "auto" id= "int5from17"></audio>


    <audio src= "Stimuli/A3Tone.wav" preload= "auto" id= "A3Tone"></audio>
    <audio src= "Stimuli/B3Tone.wav" preload= "auto" id= "B3Tone"></audio>
    <audio src= "Stimuli/Csh4Tone.wav" preload= "auto" id= "Csh4Tone"></audio>
    <audio src= "Stimuli/D4Tone.wav" preload= "auto" id= "D4Tone"></audio>
    <audio src= "Stimuli/E4Tone.wav" preload= "auto" id= "E4Tone"></audio>

    <script src="commonCode.js"></script>
    <script src="allSounds.js"></script>
    <script src="jquery-3.0.0.js"></script>
```

```
<script>
var idin= <?php echo $idin; ?>;
var recordDur = <?php echo $recordDur; ?>;
if (idin<1) {window.location.href= "wrongUsername.html"}
var bonus= 40;
var studyNo= 7;
if(score>0){explain.style= "font-size: 70px;"; explain.innerText= "Your best so far: " +
recordDur+ "ms";}
var sounds = [
    document.getElementById("int2fromA3"),
    document.getElementById("int3fromA3"),
    document.getElementById("int4fromA3"),
    document.getElementById("int5fromA3"),
    document.getElementById("int2fromC4"),
    document.getElementById("int3fromC4"),
    document.getElementById("int4fromC4"),
    document.getElementById("int5fromC4"),
    document.getElementById("int2from2"),
    document.getElementById("int3from2"),
    document.getElementById("int4from2"),
    document.getElementById("int5from2"),
    document.getElementById("int2from7"),
    document.getElementById("int3from7"),
    document.getElementById("int4from7"),
    document.getElementById("int5from7"),
    document.getElementById("int2from17"),
    document.getElementById("int3from17"),
    document.getElementById("int4from17"),
    document.getElementById("int5from17"),
]
function randNo(){
    if (duration<recordDur){lowestDur=duration;}
    randNum = Math.floor((Math.random()*20));
    switch(randNum){
        case 0: corrAns= "2nd"; intNo=2; break;
        case 1: corrAns= "3rd"; intNo=3; break;
        case 2: corrAns= "4th"; intNo=4; break;
        case 3: corrAns= "5th"; intNo=5; break;
        case 4: corrAns= "2nd"; intNo=2; break;
        case 5: corrAns= "3rd"; intNo=3; break;
        case 6: corrAns= "4th"; intNo=4; break;
        case 7: corrAns= "5th"; intNo=5; break;
        case 8: corrAns= "2nd"; intNo=2; break;
        case 9: corrAns= "3rd"; intNo=3; break;
        case 10: corrAns= "4th"; intNo=4; break;
        case 11: corrAns= "5th"; intNo=5; break;
        case 12: corrAns= "2nd"; intNo=2; break;
        case 13: corrAns= "3rd"; intNo=3; break;
```

```
        case 14: corrAns= "4th"; intNo=4; break;
        case 15: corrAns= "5th"; intNo=5; break;
        case 16: corrAns= "2nd"; intNo=2; break;
        case 17: corrAns= "3rd"; intNo=3; break;
        case 18: corrAns= "4th"; intNo=4; break;
        case 19: corrAns= "5th"; intNo=5; break;
    }
    if(intNo===2|intNo===3){randNo()}
}
var playA= document.getElementById("A3Tone");
var playB= document.getElementById("B3Tone");
var playCsh= document.getElementById("Csh4Tone");
var playD= document.getElementById("D4Tone");
var playE= document.getElementById("E4Tone");
function eTone(){playE.play();}
function dTone(){playD.play();}
function cshTone(){playCsh.play();}
function bTone(){playB.play(); }
function aTone(){playA.play(); }
randNo();
 </script>
 </body>
 </html>
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="utf-8" />
 <title>Mixed Intervals</title>
<?php
if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testdb";
$mysqli = new mysqli($servername, $username, $password, $database);
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}
$sql="SELECT * FROM participant WHERE participantNo=$idin";
if ($result=mysqli_query($mysqli,$sql))
  {
  // Fetch one and one row
  while ($row=mysqli_fetch_row($result))
    {
        $recordDur= $row[21];
    }
        mysqli_free_result($result);
    }
```

```
?>
 <link rel="stylesheet" href="Study24and5.css">
 <style>
 </style>
 </head>
 <body>

<div id= "backGround">
        <div id= "head" >
                <button id= "intro">Mixed Intervals</button>
    </div>
    <div id= "ansButtons">
        <button class="ansButton" id= "fifth" onclick= "fifth()">5th</button>
        <button class="ansButton" id= "fourth" onclick= "fourth()">4th</button>
        <button class="ansButton" id= "third" onclick= "third()">3rd</button>
        <button class="ansButton" id= "second" onclick= "second()">2nd</button>
    </div>
    <button id= "duration">200ms</button>
    <button id= "explain">When you are ready, click on "start" and you will hear two tones
that make up an interval. Select the button that matches the interval. If you get 5 right in a
row, you will go to the next level. In "Easy" mode you can listen to the tones (the bars on the
right) and also repeat the interval. Good Luck!</button>
    <button id= "results">Start by clicking "Start"</button>

    <button id= "starter" onclick= "starter()">start</button>
    <button id= "autoStart" onclick= "auto()">click for automatic starting</button>
</div>
<button id= "saveClose" onclick= "Home()" >Save and Close</button>
<audio src= "Stimuli/49991012.wav" preload= "auto" id= "int2fromA3"></audio>
    <audio src= "Stimuli/49991014.wav" preload= "auto" id= "int3fromA3"></audio>
    <audio src= "Stimuli/49991015.wav" preload= "auto" id= "int4fromA3"></audio>
    <audio src= "Stimuli/49991017.wav" preload= "auto" id= "int5fromA3"></audio>
    <audio src= "Stimuli/49991315.wav" preload= "auto" id= "int2fromC4"></audio>
    <audio src= "Stimuli/49991317.wav" preload= "auto" id= "int3fromC4"></audio>
    <audio src= "Stimuli/49991318.wav" preload= "auto" id= "int4fromC4"></audio>
    <audio src= "Stimuli/49991320.wav" preload= "auto" id= "int5fromC4"></audio>
    <audio src= "Stimuli/49990204.wav" preload= "auto" id= "int2from2"></audio>
    <audio src= "Stimuli/49990206.wav" preload= "auto" id= "int3from2"></audio>
    <audio src= "Stimuli/49990207.wav" preload= "auto" id= "int4from2"></audio>
    <audio src= "Stimuli/49990209.wav" preload= "auto" id= "int5from2"></audio>
    <audio src= "Stimuli/49990709.wav" preload= "auto" id= "int2from7"></audio>
    <audio src= "Stimuli/49990711.wav" preload= "auto" id= "int3from7"></audio>
    <audio src= "Stimuli/49990712.wav" preload= "auto" id= "int4from7"></audio>
    <audio src= "Stimuli/49990714.wav" preload= "auto" id= "int5from7"></audio>
    <audio src= "Stimuli/49991719.wav" preload= "auto" id= "int2from17"></audio>
    <audio src= "Stimuli/49991721.wav" preload= "auto" id= "int3from17"></audio>
    <audio src= "Stimuli/49991722.wav" preload= "auto" id= "int4from17"></audio>
    <audio src= "Stimuli/49991724.wav" preload= "auto" id= "int5from17"></audio>
```

```
    <audio src= "Stimuli/A3Tone.wav" preload= "auto" id= "A3Tone"></audio>
    <audio src= "Stimuli/B3Tone.wav" preload= "auto" id= "B3Tone"></audio>
    <audio src= "Stimuli/Csh4Tone.wav" preload= "auto" id= "Csh4Tone"></audio>
        <audio src= "Stimuli/D4Tone.wav" preload= "auto" id= "D4Tone"></audio>
    <audio src= "Stimuli/E4Tone.wav" preload= "auto" id= "E4Tone"></audio>


    <script src="allSounds.js"></script>
    <script src="commonCode.js"></script>
    <script src="jquery-3.0.0.js"></script>
<script>
var idin= <?php echo $idin; ?>;
var recordDur = <?php echo $recordDur; ?>;
if (idin<1) {window.location.href= "wrongUsername.html"}
if(score>0){explain.style= "font-size: 70px;"; explain.innerText= "Your best so far: " +
recordDur+ "ms";}
var bonus= 50;
var studyNo= 8;
var sounds = [
    document.getElementById("int2fromA3"),
    document.getElementById("int3fromA3"),
    document.getElementById("int4fromA3"),
    document.getElementById("int5fromA3"),
    document.getElementById("int2fromC4"),
    document.getElementById("int3fromC4"),
    document.getElementById("int4fromC4"),
    document.getElementById("int5fromC4"),
    document.getElementById("int2from2"),
    document.getElementById("int3from2"),
    document.getElementById("int4from2"),
    document.getElementById("int5from2"),
    document.getElementById("int2from7"),
    document.getElementById("int3from7"),
    document.getElementById("int4from7"),
    document.getElementById("int5from7"),
    document.getElementById("int2from17"),
    document.getElementById("int3from17"),
    document.getElementById("int4from17"),
    document.getElementById("int5from17"),
]
function randNo(){
    if (duration<recordDur){lowestDur=duration;}
    randNum = Math.floor((Math.random()*20));
    switch(randNum){
        case 0: corrAns= "2nd"; intNo=2; break;
        case 1: corrAns= "3rd"; intNo=3; break;
        case 2: corrAns= "4th"; intNo=4; break;
```

```
        case 3: corrAns= "5th"; intNo=5; break;
        case 4: corrAns= "2nd"; intNo=2; break;
        case 5: corrAns= "3rd"; intNo=3; break;
        case 6: corrAns= "4th"; intNo=4; break;
        case 7: corrAns= "5th"; intNo=5; break;
        case 8: corrAns= "2nd"; intNo=2; break;
        case 9: corrAns= "3rd"; intNo=3; break;
        case 10: corrAns= "4th"; intNo=4; break;
        case 11: corrAns= "5th"; intNo=5; break;
        case 12: corrAns= "2nd"; intNo=2; break;
        case 13: corrAns= "3rd"; intNo=3; break;
        case 14: corrAns= "4th"; intNo=4; break;
        case 15: corrAns= "5th"; intNo=5; break;
        case 16: corrAns= "2nd"; intNo=2; break;
        case 17: corrAns= "3rd"; intNo=3; break;
        case 18: corrAns= "4th"; intNo=4; break;
        case 19: corrAns= "5th"; intNo=5; break;
    }
    if(intNo===3){randNo()}
}
var playA= document.getElementById("A3Tone");
var playB= document.getElementById("B3Tone");
var playCsh= document.getElementById("Csh4Tone");
var playD= document.getElementById("D4Tone");
var playE= document.getElementById("E4Tone");
function eTone(){playE.play();}
function dTone(){playD.play();}
function cshTone(){playCsh.play();}
function bTone(){playB.play(); }
function aTone(){playA.play(); }
randNo();
 </script>
 </body>
 </html>
<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8" />
<?php
if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testdb";
$mysqli = new mysqli($servername, $username, $password, $database);
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}
```

```
$sql="SELECT * FROM participant WHERE participantNo=$idin";
if ($result=mysqli_query($mysqli,$sql))
  {
  // Fetch one and one row
  while ($row=mysqli_fetch_row($result))
    {
      $recordDur= $row[15];
    }
      mysqli_free_result($result);
    }
?>
 <title>Intervals</title>
 <link rel="stylesheet" href="Study24and5.css">
 <style>
 </style>
 </head>
 <body>

<div id= "backGround">
      <div id= "head" >
            <button id= "intro">Intervals</button>
    </div>
    <div id= "ansButtons">
      <button class="ansButton" id= "fifth" onclick= "fifth()">5th</button>
      <button class="ansButton" id= "fourth" onclick= "fourth()">4th</button>
      <button class="ansButton" id= "third" onclick= "third()">3rd</button>
      <button class="ansButton" id= "second" onclick= "second()">2nd</button>
    </div>
    <div id= "marimba">
      <button class= "bars" style= "width:60px;" onclick= "eTone()">G</button>
      <button class= "bars" style= "width:70px;" onclick= "dTone()">F</button>
      <button class= "bars" style= "width:80px;" onclick= "cshTone()">E</button>
      <button class= "bars" style= "width:90px;" onclick= "bTone()">D</button>
      <button class= "bars" style= "width:100px;" onclick= "aTone()">C</button>
    </div>
    <button id= "duration">200ms</button>
    <button id= "explain">When you are ready, click on "start" and you will hear two tones
that make up an interval. Select the button that matches the interval. If you get 5 right in a
row, you will go to the next level. In "Easy" mode you can listen to the tones (the bars on the
right) and also repeat the interval. Good Luck!</button>
    <button id= "results">Start by clicking "Start"</button>
    <button id= "starter" onclick= "starter()">start</button>
    <button id= "autoStart" onclick= "auto()">click for automatic starting</button>
</div>
<button id= "saveClose" onclick= "Home()" >Save and Close</button>
<audio src= "Stimuli/49991012.wav" preload= "auto" id= "int2fromA3"></audio>
    <audio src= "Stimuli/49991014.wav" preload= "auto" id= "int3fromA3"></audio>
    <audio src= "Stimuli/49991015.wav" preload= "auto" id= "int4fromA3"></audio>
```

```
    <audio src= "Stimuli/49991017.wav" preload= "auto" id= "int5fromA3"></audio>

    <audio src= "Stimuli/A3Tone.wav" preload= "auto" id= "A3Tone"></audio>
    <audio src= "Stimuli/B3Tone.wav" preload= "auto" id= "B3Tone"></audio>
    <audio src= "Stimuli/Csh4Tone.wav" preload= "auto" id= "Csh4Tone"></audio>
    <audio src= "Stimuli/D4Tone.wav" preload= "auto" id= "D4Tone"></audio>
    <audio src= "Stimuli/E4Tone.wav" preload= "auto" id= "E4Tone"></audio>

    <script src="commonCode.js"></script>
    <script src="singleSounds.js"></script>
    <script src="jquery-3.0.0.js"></script>
<script>
var idin= <?php echo $idin; ?>;
var recordDur= <?php echo $recordDur; ?>;
if (idin<1) {window.location.href= "wrongUsername.html"}
var bonus= 20;
var studyNo= 2;
var sounds = [
    document.getElementById("int2fromA3"),
    document.getElementById("int3fromA3"),
    document.getElementById("int4fromA3"),
    document.getElementById("int5fromA3"),
]
function randNo(){
    if (duration<recordDur){lowestDur=duration}
    randNum = Math.floor((Math.random()*4));
    switch(randNum){
        case 0: corrAns= "2nd"; intNo=2; break;
        case 1: corrAns= "3rd"; intNo=3; break;
        case 2: corrAns= "4th"; intNo=4; break;
        case 3: corrAns= "5th"; intNo=5; break;
    }
    if(intNo===3){randNo()}
}
if(score>0){explain.style= "font-size: 70px;"; explain.innerText= "Lowest Duration so far: "
+ recordDur+ "ms";}
var playA= document.getElementById("A3Tone");
var playB= document.getElementById("B3Tone");
var playCsh= document.getElementById("Csh4Tone");
var playD= document.getElementById("D4Tone");
var playE= document.getElementById("E4Tone");
function eTone(){playE.play();}
function dTone(){playD.play();}
function cshTone(){playCsh.play();}
function bTone(){playB.play(); }
function aTone(){playA.play(); }
randNo();
 </script>
```

```
 </body>
 </html>
<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8" />
<?php
if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testdb";
$mysqli = new mysqli($servername, $username, $password, $database);
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}
$sql="SELECT * FROM participant WHERE participantNo=$idin";
if ($result=mysqli_query($mysqli,$sql))
  {
  // Fetch one and one row
  while ($row=mysqli_fetch_row($result))
    {
        $recordDur= $row[14];
    }
        mysqli_free_result($result);
    }
?>
 <title>Intervals</title>
 <link rel="stylesheet" href="Study4and5.css">
 <style>
#instructAnsButtons{
    position: relative;
    top: 20px;
    margin: 10px;
    background-color: yellow;
    padding: 10px;
    font-size: 20px;
}
 </style>
 </head>
 <body>

<div id= "backGround">
        <div id= "head" >
                <button id= "intro">Intervals</button>
    </div>
    <div id= "ansButtons">
        <button class="ansButton" id= "fifth" onclick= "fifth()">5th</button>
```

```
<button class="ansButton" id= "fourth" onclick= "fourth()">4th</button>
<button class="ansButton" id= "third" onclick= "third()">3rd</button>
<button class="ansButton" id= "second" onclick= "second()">2nd</button>
<div id= "instructAnsButtons">Click These buttons to Answer</div>
</div>
<div id= "marimba">
<button class= "bars" style= "width:60px;" onclick= "eTone()">G</button>
<button class= "bars" style= "width:70px;" onclick= "dTone()">F</button>
<button class= "bars" style= "width:80px;" onclick= "cshTone()">E</button>
<button class= "bars" style= "width:90px;" onclick= "bTone()">D</button>
<button class= "bars" style= "width:100px;" onclick= "aTone()">C</button>
</div>
<button id= "duration">200ms</button>
<button id= "explain">**Please be careful about the volume level ** When you are ready,
click on "start" and you will hear two tones that make up an interval. Select the button that
matches the interval. If you get 5 right in a row, you will go to the next level. In "Easy" mode
you can listen to the tones (the bars on the right) and also repeat the interval. Good
Luck!</button>
<button id= "results">Start by clicking "Start"</button>

<button id= "starter" onclick= "starter()">start</button>
<button id= "autoStart" onclick= "auto()">click for automatic starting</button>
<img src="intArrows.png" style= "position:absolute; left: 930px; top: 85px; height: 510px;
width: 200px">
<button id= "Playbar"style= "position:absolute; left: 840px; top: 590px; height: 80px;
width: 200px; background-color: yellow; color: black;">Click to play the bars. 'C' is the first
tone you hear (the starting tone). Try and match the second tone by clicking the bars
</button>
<button id= "Playbar"style= "position:absolute; left: -30px; top: 60px; height: 50px; width:
100px; background-color: black; color: white;">Try to get to 20ms</button>
</div>
<button id= "saveClose" onclick= "Home()" >Save and Close</button>
<audio src= "Stimuli/49991012.wav" volume= ".5" preload= "auto" id=
"int2fromA3"></audio>
<audio src= "Stimuli/49991014.wav" volume= ".5" preload= "auto" id=
"int3fromA3"></audio>
<audio src= "Stimuli/49991015.wav" volume= ".5" preload= "auto" id=
"int4fromA3"></audio>
<audio src= "Stimuli/49991017.wav" volume= ".5" preload= "auto" id=
"int5fromA3"></audio>

<audio src= "Stimuli/A3Tone.wav" volume= ".5" preload= "auto" id= "A3Tone"></audio>
<audio src= "Stimuli/B3Tone.wav" volume= ".5" preload= "auto" id= "B3Tone"></audio>
<audio src= "Stimuli/Csh4Tone.wav" volume= ".5" preload= "auto" id=
"Csh4Tone"></audio>
<audio src= "Stimuli/D4Tone.wav" volume= ".5" preload= "auto" id= "D4Tone"></audio>
<audio src= "Stimuli/E4Tone.wav" volume= ".5" preload= "auto" id= "E4Tone"></audio>
```

```
    <script src="commonCode.js"></script>
    <script src="singleSounds.js"></script>
    <script src="jquery-3.0.0.js"></script>
<script>
var idin= <?php echo $idin; ?>;
var recordDur = <?php echo $recordDur; ?>;
if (idin<1) {window.location.href= "wrongUsername.html"}
var bonus=0;
var studyNo= 1;
var sounds = [
    document.getElementById("int2fromA3"),
    document.getElementById("int3fromA3"),
    document.getElementById("int4fromA3"),
    document.getElementById("int5fromA3"),
]
function randNo(){
    if (duration<recordDur){lowestDur=duration;}
    randNum = Math.floor((Math.random()*4));
    switch(randNum){
        case 0: corrAns= "2nd"; intNo=2; break;
        case 1: corrAns= "3rd"; intNo=3; break;
        case 2: corrAns= "4th"; intNo=4; break;
        case 3: corrAns= "5th"; intNo=5; break;
    }
    if(intNo===2|intNo===3){randNo()}
}
if(score>0){explain.style= "font-size: 70px;"; explain.innerText= "Your best so far: " +
recordDur+ "ms";}
var playA= document.getElementById("A3Tone");
var playB= document.getElementById("B3Tone");
var playCsh= document.getElementById("Csh4Tone");
var playD= document.getElementById("D4Tone");
var playE= document.getElementById("E4Tone");
function eTone(){playE.play();}
function dTone(){playD.play();}
function cshTone(){playCsh.play();}
function bTone(){playB.play(); }
function aTone(){playA.play(); }
randNo();
 </script>
 </body>
 </html>
<!DOCTYPE html>
<html lang="en">
 <meta charset="utf-8" />
 <title>Intervals</title>
<?php
if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}
```

```
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testdb";
$mysqli = new mysqli($servername, $username, $password, $database);
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}
$sql="SELECT * FROM participant WHERE participantNo=$idin";
if ($result=mysqli_query($mysqli,$sql))
  {
  // Fetch one and one row
  while ($row=mysqli_fetch_row($result))
    {
        $recordDur= $row[16];
    }
        mysqli_free_result($result);
    }
?>
 <head>
 <link rel="stylesheet" href="Study4Abs.css">
 <style>
 </style>
 </head>
 <body>

<div id= "backGround">
        <div id= "head" >
                <button id= "intro">Intervals</button>
    </div>
    <div id= "ansButtons">
        <button class="ansButton" id= "fifth" onclick= "fifth()">5th</button>
        <button class="ansButton" id= "fourth" onclick= "fourth()">4th</button>
        <button class="ansButton" id= "third" onclick= "third()">3rd</button>
        <button class="ansButton" id= "second" onclick= "second()">2nd</button>
    </div>
    <div id= "marimba">
        <button class= "bars" style= "width:60px;" onclick= "eTone()">G</button>
        <button class= "bars" style= "width:70px;" onclick= "dTone()">F</button>
        <button class= "bars" style= "width:80px;" onclick= "cshTone()">E</button>
        <button class= "bars" style= "width:90px;" onclick= "bTone()">D</button>
        <button class= "bars" style= "width:100px;" onclick= "aTone()">C</button>
    </div>
    <button id= "duration">200ms</button>
    <button id= "explain">When you are ready, click on "start" and you will hear two tones
that make up an interval. Select the button that matches the interval. If you get 5 right in a
row, you will go to the next level. In "Easy" mode you can listen to the tones (the bars on the
right) and also repeat the interval. Good Luck!</button>
```

```
    <button id= "results">Start by clicking "Start"</button>
    <button id= "starter" onclick= "starter()">start</button>
    <button id= "autoStart" onclick= "auto()">click for automatic starting</button>
</div>
<button id= "saveClose" onclick= "Home()" >Save and Close</button>
<audio src= "Stimuli/49991012.wav" preload= "auto" id= "int2fromA3"></audio>
    <audio src= "Stimuli/49991014.wav" preload= "auto" id= "int3fromA3"></audio>
    <audio src= "Stimuli/49991015.wav" preload= "auto" id= "int4fromA3"></audio>
    <audio src= "Stimuli/49991017.wav" preload= "auto" id= "int5fromA3"></audio>

    <audio src= "Stimuli/A3Tone.wav" preload= "auto" id= "A3Tone"></audio>
    <audio src= "Stimuli/B3Tone.wav" preload= "auto" id= "B3Tone"></audio>
    <audio src= "Stimuli/Csh4Tone.wav" preload= "auto" id= "Csh4Tone"></audio>
    <audio src= "Stimuli/D4Tone.wav" preload= "auto" id= "D4Tone"></audio>
    <audio src= "Stimuli/E4Tone.wav" preload= "auto" id= "E4Tone"></audio>

    <script src="commonCode.js"></script>
    <script src="singleSounds.js"></script>
    <script src="jquery-3.0.0.js"></script>
<script>
var idin= <?php echo $idin; ?>;
var recordDur = <?php echo $recordDur; ?>;
if (idin<1) {window.location.href= "wrongUsername.html"}
var bonus= 40;
var studyNo= 3;
var sounds = [
    document.getElementById("int2fromA3"),
    document.getElementById("int3fromA3"),
    document.getElementById("int4fromA3"),
    document.getElementById("int5fromA3"),
]
function randNo(){
    if (duration<recordDur){lowestDur=duration;}
    randNum = Math.floor((Math.random()*4));
    switch(randNum){
        case 0: corrAns= "2nd"; intNo=2; break;
        case 1: corrAns= "3rd"; intNo=3; break;
        case 2: corrAns= "4th"; intNo=4; break;
        case 3: corrAns= "5th"; intNo=5; break;
    }

}
if(score>0){explain.style= "font-size: 70px;"; explain.innerText= "score: " + score;}
var playA= document.getElementById("A3Tone");
var playB= document.getElementById("B3Tone");
var playCsh= document.getElementById("Csh4Tone");
var playD= document.getElementById("D4Tone");
var playE= document.getElementById("E4Tone");
```

```
function eTone(){playE.play();}
function dTone(){playD.play();}
function cshTone(){playCsh.play();}
function bTone(){playB.play(); }
function aTone(){playA.play(); }
randNo();
 </script>
 </body>
 </html>
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="utf-8" />
<?php
if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testdb";
$mysqli = new mysqli($servername, $username, $password, $database);
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}
$sql="SELECT * FROM participant WHERE participantNo=$idin";
if ($result=mysqli_query($mysqli,$sql))
  {
  // Fetch one and one row
  while ($row=mysqli_fetch_row($result))
    {
       $recordDur= $row[22];
    }
       mysqli_free_result($result);
    }
?>

 <title>Mixed Intervals</title>
 <link rel="stylesheet" href="Study4Abs.css">
 <style>
 </style>
 </head>
 <body>

<div id= "backGround">
        <div id= "head" >
                <button id= "intro">Mixed Intervals</button>
    </div>
    <div id= "ansButtons">
        <button class="ansButton" id= "fifth" onclick= "fifth()">5th</button>
```

```
        <button class="ansButton" id= "fourth" onclick= "fourth()">4th</button>
        <button class="ansButton" id= "third" onclick= "third()">3rd</button>
        <button class="ansButton" id= "second" onclick= "second()">2nd</button>
    </div>
    <button id= "duration">200ms</button>
    <button id= "explain">When you are ready, click on "start" and you will hear two tones
that make up an interval. Select the button that matches the interval. If you get 5 right in a
row, you will go to the next level. In "Easy" mode you can listen to the tones (the bars on the
right) and also repeat the interval. Good Luck!</button>
    <button id= "results">Start by clicking "Start"</button>

    <button id= "starter" onclick= "starter()">start</button>
    <button id= "autoStart" onclick= "auto()">click for automatic starting</button>
</div>
<button id= "saveClose" onclick= "Home()" >Save and Close</button>
<audio src= "Stimuli/49991012.wav" preload= "auto" id= "int2fromA3"></audio>
    <audio src= "Stimuli/49991014.wav" preload= "auto" id= "int3fromA3"></audio>
    <audio src= "Stimuli/49991015.wav" preload= "auto" id= "int4fromA3"></audio>
    <audio src= "Stimuli/49991017.wav" preload= "auto" id= "int5fromA3"></audio>
    <audio src= "Stimuli/49991315.wav" preload= "auto" id= "int2fromC4"></audio>
    <audio src= "Stimuli/49991317.wav" preload= "auto" id= "int3fromC4"></audio>
    <audio src= "Stimuli/49991318.wav" preload= "auto" id= "int4fromC4"></audio>
    <audio src= "Stimuli/49991320.wav" preload= "auto" id= "int5fromC4"></audio>
    <audio src= "Stimuli/49990204.wav" preload= "auto" id= "int2from2"></audio>
    <audio src= "Stimuli/49990206.wav" preload= "auto" id= "int3from2"></audio>
    <audio src= "Stimuli/49990207.wav" preload= "auto" id= "int4from2"></audio>
    <audio src= "Stimuli/49990209.wav" preload= "auto" id= "int5from2"></audio>
    <audio src= "Stimuli/49990709.wav" preload= "auto" id= "int2from7"></audio>
    <audio src= "Stimuli/49990711.wav" preload= "auto" id= "int3from7"></audio>
    <audio src= "Stimuli/49990712.wav" preload= "auto" id= "int4from7"></audio>
    <audio src= "Stimuli/49990714.wav" preload= "auto" id= "int5from7"></audio>
    <audio src= "Stimuli/49991719.wav" preload= "auto" id= "int2from17"></audio>
    <audio src= "Stimuli/49991721.wav" preload= "auto" id= "int3from17"></audio>
    <audio src= "Stimuli/49991722.wav" preload= "auto" id= "int4from17"></audio>
    <audio src= "Stimuli/49991724.wav" preload= "auto" id= "int5from17"></audio>

    <audio src= "Stimuli/A3Tone.wav" preload= "auto" id= "A3Tone"></audio>
    <audio src= "Stimuli/B3Tone.wav" preload= "auto" id= "B3Tone"></audio>
    <audio src= "Stimuli/Csh4Tone.wav" preload= "auto" id= "Csh4Tone"></audio>
    <audio src= "Stimuli/D4Tone.wav" preload= "auto" id= "D4Tone"></audio>
    <audio src= "Stimuli/E4Tone.wav" preload= "auto" id= "E4Tone"></audio>

    <script src="commonCode.js"></script>
    <script src="allSounds.js"></script>
    <script src="jquery-3.0.0.js"></script>
<script>
var idin= <?php echo $idin; ?>;
var recordDur= <?php echo $recordDur; ?>;
```

```
if (idin<1) {window.location.href= "wrongUsername.html"}
var bonus= 70;
var studyNo= 9;
var sounds = [
   document.getElementById("int2fromA3"),
   document.getElementById("int3fromA3"),
   document.getElementById("int4fromA3"),
   document.getElementById("int5fromA3"),
   document.getElementById("int2fromC4"),
   document.getElementById("int3fromC4"),
   document.getElementById("int4fromC4"),
   document.getElementById("int5fromC4"),
   document.getElementById("int2from2"),
   document.getElementById("int3from2"),
   document.getElementById("int4from2"),
   document.getElementById("int5from2"),
   document.getElementById("int2from7"),
   document.getElementById("int3from7"),
   document.getElementById("int4from7"),
   document.getElementById("int5from7"),
   document.getElementById("int2from17"),
   document.getElementById("int3from17"),
   document.getElementById("int4from17"),
   document.getElementById("int5from17"),
]
function randNo(){
   if (duration<recordDur){lowestDur=duration}
   randNum = Math.floor((Math.random()*20));
   switch(randNum){
      case 0: corrAns= "2nd"; intNo=2; break;
      case 1: corrAns= "3rd"; intNo=3; break;
      case 2: corrAns= "4th"; intNo=4; break;
      case 3: corrAns= "5th"; intNo=5; break;
      case 4: corrAns= "2nd"; intNo=2; break;
      case 5: corrAns= "3rd"; intNo=3; break;
      case 6: corrAns= "4th"; intNo=4; break;
      case 7: corrAns= "5th"; intNo=5; break;
      case 8: corrAns= "2nd"; intNo=2; break;
      case 9: corrAns= "3rd"; intNo=3; break;
      case 10: corrAns= "4th"; intNo=4; break;
      case 11: corrAns= "5th"; intNo=5; break;
      case 12: corrAns= "2nd"; intNo=2; break;
      case 13: corrAns= "3rd"; intNo=3; break;
      case 14: corrAns= "4th"; intNo=4; break;
      case 15: corrAns= "5th"; intNo=5; break;
      case 16: corrAns= "2nd"; intNo=2; break;
      case 17: corrAns= "3rd"; intNo=3; break;
      case 18: corrAns= "4th"; intNo=4; break;
```

```
        case 19: corrAns= "5th"; intNo=5; break;
    }

}
if(score>0){explain.style= "font-size: 70px;"; explain.innerText= "Lowest Duration so far: "
+ recordDur+ "ms";}
var playA= document.getElementById("A3Tone");
var playB= document.getElementById("B3Tone");
var playCsh= document.getElementById("Csh4Tone");
var playD= document.getElementById("D4Tone");
var playE= document.getElementById("E4Tone");
function eTone(){playE.play();}
function dTone(){playD.play();}
function cshTone(){playCsh.play();}
function bTone(){playB.play(); }
function aTone(){playA.play(); }
randNo();
 </script>
 </body>
 </html>
<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8" />
<?php
if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testdb";
$mysqli = new mysqli($servername, $username, $password, $database);
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}
$sql="SELECT * FROM participant WHERE participantNo=$idin";
if ($result=mysqli_query($mysqli,$sql))
  {
  // Fetch one and one row
  while ($row=mysqli_fetch_row($result))
    {
        $recordDur= $row[17];
    }
        mysqli_free_result($result);
    }
?>
 <title>2Note Intervals</title>
 <link rel="stylesheet" href="Study4and5.css">
 <style>
```

```
 </style>
 </head>
 <body>

<div id= "backGround">
      <div id= "head" >
            <button id= "intro">2Note Intervals</button>
   </div>
   <div id= "ansButtons">
      <button class="ansButton" id= "fifth" onclick= "fifth()">5th</button>
      <button class="ansButton" id= "fourth" onclick= "fourth()">4th</button>
      <button class="ansButton" id= "third" onclick= "third()">3rd</button>
      <button class="ansButton" id= "second" onclick= "second()">2nd</button>
   </div>
   <button id= "duration">200ms</button>
   <button id= "explain">When you are ready, click on "start" and you will hear two tones
that make up an interval. Select the button that matches the interval. If you get 5 right in a
row, you will go to the next level. In "Easy" mode you can listen to the tones (the bars on the
right) and also repeat the interval. Good Luck!</button>
   <button id= "results">Start by clicking "Start"</button>
   <button id= "starter" onclick= "starter()">start</button>
   <button id= "autoStart" onclick= "auto()">click for automatic starting</button>
</div>
<button id= "saveClose" onclick= "Home()" >Save and Close</button>
<audio src= "Stimuli/49991012.wav" preload= "auto" id= "int2fromA3"></audio>
   <audio src= "Stimuli/49991014.wav" preload= "auto" id= "int3fromA3"></audio>
   <audio src= "Stimuli/49991015.wav" preload= "auto" id= "int4fromA3"></audio>
   <audio src= "Stimuli/49991017.wav" preload= "auto" id= "int5fromA3"></audio>
   <audio src= "Stimuli/49991315.wav" preload= "auto" id= "int2fromC4"></audio>
   <audio src= "Stimuli/49991317.wav" preload= "auto" id= "int3fromC4"></audio>
   <audio src= "Stimuli/49991318.wav" preload= "auto" id= "int4fromC4"></audio>
   <audio src= "Stimuli/49991320.wav" preload= "auto" id= "int5fromC4"></audio>

   <audio src= "Stimuli/A3Tone.wav" preload= "auto" id= "A3Tone"></audio>
   <audio src= "Stimuli/B3Tone.wav" preload= "auto" id= "B3Tone"></audio>
   <audio src= "Stimuli/C4Tone.wav" preload= "auto" id= "C4Tone"></audio>
   <audio src= "Stimuli/Csh4Tone.wav" preload= "auto" id= "Csh4Tone"></audio>
   <audio src= "Stimuli/D4Tone.wav" preload= "auto" id= "D4Tone"></audio>
   <audio src= "Stimuli/E4Tone.wav" preload= "auto" id= "E4Tone"></audio>
   <audio src= "Stimuli/F4Tone.wav" preload= "auto" id= "F4Tone"></audio>
   <audio src= "Stimuli/G4Tone.wav" preload= "auto" id= "G4Tone"></audio>

   <script src="commonCode.js"></script>
   <script src="sounds.js"></script>
   <script src="jquery-3.0.0.js"></script>
<script>
var idin= <?php echo $idin; ?>;
var recordDur = <?php echo $recordDur; ?>;
```

```
if (idin<1) {window.location.href= "wrongUsername.html"}
var bonus= 20;
var studyNo= 4;
var sounds = [
   document.getElementById("int2fromA3"),
   document.getElementById("int3fromA3"),
   document.getElementById("int4fromA3"),
   document.getElementById("int5fromA3"),
   document.getElementById("int2fromC4"),
   document.getElementById("int3fromC4"),
   document.getElementById("int4fromC4"),
   document.getElementById("int5fromC4"),
]
function randNo(){
   if (duration<recordDur){lowestDur=duration;}
   randNum = Math.floor((Math.random()*8));
   switch(randNum){
      case 0: corrAns= "2nd"; intNo=2; break;
      case 1: corrAns= "3rd"; intNo=3; break;
      case 2: corrAns= "4th"; intNo=4; break;
      case 3: corrAns= "5th"; intNo=5; break;
      case 4: corrAns= "2nd"; intNo=2; break;
      case 5: corrAns= "3rd"; intNo=3; break;
      case 6: corrAns= "4th"; intNo=4; break;
      case 7: corrAns= "5th"; intNo=5; break;
   }
   if(intNo===2|intNo===3){randNo()}
}
if(score>0){explain.style= "font-size: 70px;"; explain.innerText= "Your best so far: " +
recordDur+ "ms";}
var playA= document.getElementById("A3Tone");
var playB= document.getElementById("B3Tone");
var playC= document.getElementById("C4Tone");
var playCsh= document.getElementById("Csh4Tone");
var playD= document.getElementById("D4Tone");
var playE= document.getElementById("E4Tone");
var playF= document.getElementById("F4Tone");
var playG= document.getElementById("G4Tone");
function eTone(){if(randNum<4){playE.play();}else{playG.play()}}
function dTone(){if(randNum<4){playD.play();}else{playF.play()}}
function cshTone(){if(randNum<4){playCsh.play();}else{playE.play()}}
function bTone(){if(randNum<4){playB.play();}else{playD.play()}}
function aTone(){if(randNum<4){playA.play();}else{playC.play()}}
randNo();
 </script>
 </body>
 </html>
<!DOCTYPE html>
```

```
<html lang="en">
 <meta charset="utf-8" />
 <title>2Note Intervals</title>
<?php
if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testdb";
$mysqli = new mysqli($servername, $username, $password, $database);
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}
$sql="SELECT * FROM participant WHERE participantNo=$idin";
if ($result=mysqli_query($mysqli,$sql))
  {
  // Fetch one and one row
  while ($row=mysqli_fetch_row($result))
    {
        $recordDur= $row[18];
    }
        mysqli_free_result($result);
    }
?>
 <head>
 <link rel="stylesheet" href="Study24and5.css">
 <style>
 </style>
 </head>
 <body>

<div id= "backGround">
        <div id= "head" >
                <button id= "intro">2Note Intervals</button>
    </div>
    <div id= "ansButtons">
        <button class="ansButton" id= "fifth" onclick= "fifth()">5th</button>
        <button class="ansButton" id= "fourth" onclick= "fourth()">4th</button>
        <button class="ansButton" id= "third" onclick= "third()">3rd</button>
        <button class="ansButton" id= "second" onclick= "second()">2nd</button>
    </div>
    <button id= "duration">200ms</button>
    <button id= "explain">When you are ready, click on "start" and you will hear two tones
that make up an interval. Select the button that matches the interval. If you get 5 right in a
row, you will go to the next level. In "Easy" mode you can listen to the tones (the bars on the
right) and also repeat the interval. Good Luck!</button>
    <button id= "results">Start by clicking "Start"</button>
```

```
    <button id= "starter" onclick= "starter()">start</button>
    <button id= "autoStart" onclick= "auto()">click for automatic starting</button>
</div>
<button id= "saveClose" onclick= "Home()" >Save and Close</button>
<audio src= "Stimuli/49991012.wav" preload= "auto" id= "int2fromA3"></audio>
    <audio src= "Stimuli/49991014.wav" preload= "auto" id= "int3fromA3"></audio>
    <audio src= "Stimuli/49991015.wav" preload= "auto" id= "int4fromA3"></audio>
    <audio src= "Stimuli/49991017.wav" preload= "auto" id= "int5fromA3"></audio>
    <audio src= "Stimuli/49991315.wav" preload= "auto" id= "int2fromC4"></audio>
    <audio src= "Stimuli/49991317.wav" preload= "auto" id= "int3fromC4"></audio>
    <audio src= "Stimuli/49991318.wav" preload= "auto" id= "int4fromC4"></audio>
    <audio src= "Stimuli/49991320.wav" preload= "auto" id= "int5fromC4"></audio>

    <audio src= "Stimuli/A3Tone.wav" preload= "auto" id= "A3Tone"></audio>
    <audio src= "Stimuli/B3Tone.wav" preload= "auto" id= "B3Tone"></audio>
    <audio src= "Stimuli/C4Tone.wav" preload= "auto" id= "C4Tone"></audio>
    <audio src= "Stimuli/Csh4Tone.wav" preload= "auto" id= "Csh4Tone"></audio>
    <audio src= "Stimuli/D4Tone.wav" preload= "auto" id= "D4Tone"></audio>
    <audio src= "Stimuli/E4Tone.wav" preload= "auto" id= "E4Tone"></audio>
    <audio src= "Stimuli/F4Tone.wav" preload= "auto" id= "F4Tone"></audio>
    <audio src= "Stimuli/G4Tone.wav" preload= "auto" id= "G4Tone"></audio>

    <script src="commonCode.js"></script>
    <script src="sounds.js"></script>
    <script src="jquery-3.0.0.js"></script>
<script>
var idin= <?php echo $idin; ?>;
var recordDur = <?php echo $recordDur; ?>;
if (idin<1) {window.location.href= "wrongUsername.html"}
var bonus= 40;
var studyNo= 5;
var sounds = [
    document.getElementById("int2fromA3"),
    document.getElementById("int3fromA3"),
    document.getElementById("int4fromA3"),
    document.getElementById("int5fromA3"),
    document.getElementById("int2fromC4"),
    document.getElementById("int3fromC4"),
    document.getElementById("int4fromC4"),
    document.getElementById("int5fromC4"),
]
function randNo(){
    if (duration<recordDur){lowestDur=duration;}
    randNum = Math.floor((Math.random()*8));
    switch(randNum){
        case 0: corrAns= "2nd"; intNo=2; break;
        case 1: corrAns= "3rd"; intNo=3; break;
        case 2: corrAns= "4th"; intNo=4; break;
```

```
        case 3: corrAns= "5th"; intNo=5; break;
        case 4: corrAns= "2nd"; intNo=2; break;
        case 5: corrAns= "3rd"; intNo=3; break;
        case 6: corrAns= "4th"; intNo=4; break;
        case 7: corrAns= "5th"; intNo=5; break;
    }
    if(intNo===3){randNo()}
}
if(score>0){explain.style= "font-size: 70px;"; explain.innerText= "Your best so far: " +
recordDur+ "ms";}
var playA= document.getElementById("A3Tone");
var playB= document.getElementById("B3Tone");
var playC= document.getElementById("C4Tone");
var playCsh= document.getElementById("Csh4Tone");
var playD= document.getElementById("D4Tone");
var playE= document.getElementById("E4Tone");
var playF= document.getElementById("F4Tone");
var playG= document.getElementById("G4Tone");
function eTone(){if(randNum<4){playE.play();}else{playG.play()}}
function dTone(){if(randNum<4){playD.play();}else{playF.play()}}
function cshTone(){if(randNum<4){playCsh.play();}else{playE.play()}}
function bTone(){if(randNum<4){playB.play();}else{playD.play()}}
function aTone(){if(randNum<4){playA.play();}else{playC.play()}}
randNo();
 </script>
 </body>
 </html>
<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8" />
<?php
if (isset($_GET["idin"])) {$idin = intval($_GET["idin"]);}
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testdb";
$mysqli = new mysqli($servername, $username, $password, $database);
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}
$sql="SELECT * FROM participant WHERE participantNo=$idin";
if ($result=mysqli_query($mysqli,$sql))
  {
  // Fetch one and one row
  while ($row=mysqli_fetch_row($result))
    {
        $recordDur= $row[19];
```

```
    }
        mysqli_free_result($result);
    }
?>
 <title>2Note Intervals</title>
 <link rel="stylesheet" href="Study4Abs.css">
 <style>
 </style>
 </head>
 <body>

<div id= "backGround">
        <div id= "head" >
                <button id= "intro">2Note Intervals</button>
    </div>
    <div id= "ansButtons">
        <button class="ansButton" id= "fifth" onclick= "fifth()">5th</button>
        <button class="ansButton" id= "fourth" onclick= "fourth()">4th</button>
        <button class="ansButton" id= "third" onclick= "third()">3rd</button>
        <button class="ansButton" id= "second" onclick= "second()">2nd</button>
    </div>
    <button id= "duration">200ms</button>
    <button id= "explain">When you are ready, click on "start" and you will hear two tones
that make up an interval. Select the button that matches the interval. If you get 5 right in a
row, you will go to the next level. In "Easy" mode you can listen to the tones (the bars on the
right) and also repeat the interval. Good Luck!</button>
    <button id= "results">Start by clicking "Start"</button>

    <button id= "starter" onclick= "starter()">start</button>
    <button id= "autoStart" onclick= "auto()">click for automatic starting</button>
</div>
<button id= "saveClose" onclick= "Home()" >Save and Close</button>
<audio src= "Stimuli/49991012.wav" preload= "auto" id= "int2fromA3"></audio>
    <audio src= "Stimuli/49991014.wav" preload= "auto" id= "int3fromA3"></audio>
    <audio src= "Stimuli/49991015.wav" preload= "auto" id= "int4fromA3"></audio>
    <audio src= "Stimuli/49991017.wav" preload= "auto" id= "int5fromA3"></audio>
    <audio src= "Stimuli/49991315.wav" preload= "auto" id= "int2fromC4"></audio>
    <audio src= "Stimuli/49991317.wav" preload= "auto" id= "int3fromC4"></audio>
    <audio src= "Stimuli/49991318.wav" preload= "auto" id= "int4fromC4"></audio>
    <audio src= "Stimuli/49991320.wav" preload= "auto" id= "int5fromC4"></audio>

    <audio src= "Stimuli/A3Tone.wav" preload= "auto" id= "A3Tone"></audio>
    <audio src= "Stimuli/B3Tone.wav" preload= "auto" id= "B3Tone"></audio>
    <audio src= "Stimuli/C4Tone.wav" preload= "auto" id= "C4Tone"></audio>
    <audio src= "Stimuli/Csh4Tone.wav" preload= "auto" id= "Csh4Tone"></audio>
    <audio src= "Stimuli/D4Tone.wav" preload= "auto" id= "D4Tone"></audio>
    <audio src= "Stimuli/E4Tone.wav" preload= "auto" id= "E4Tone"></audio>
    <audio src= "Stimuli/F4Tone.wav" preload= "auto" id= "F4Tone"></audio>
```

```
    <audio src= "Stimuli/G4Tone.wav" preload= "auto" id= "G4Tone"></audio>

    <script src="commonCode.js"></script>
    <script src="sounds.js"></script>
    <script src="jquery-3.0.0.js"></script>
<script>
var idin= <?php echo $idin; ?>;
var recordDur = <?php echo $recordDur; ?>;
if (idin<1) {window.location.href= "wrongUsername.html"}
var bonus= 50;
var studyNo= 6;
var sounds = [
    document.getElementById("int2fromA3"),
    document.getElementById("int3fromA3"),
    document.getElementById("int4fromA3"),
    document.getElementById("int5fromA3"),
    document.getElementById("int2fromC4"),
    document.getElementById("int3fromC4"),
    document.getElementById("int4fromC4"),
    document.getElementById("int5fromC4"),
]
function randNo(){
    if (duration<recordDur){lowestDur=duration;}
    randNum = Math.floor((Math.random()*8));
    switch(randNum){
        case 0: corrAns= "2nd"; intNo=2; break;
        case 1: corrAns= "3rd"; intNo=3; break;
        case 2: corrAns= "4th"; intNo=4; break;
        case 3: corrAns= "5th"; intNo=5; break;
        case 4: corrAns= "2nd"; intNo=2; break;
        case 5: corrAns= "3rd"; intNo=3; break;
        case 6: corrAns= "4th"; intNo=4; break;
        case 7: corrAns= "5th"; intNo=5; break;
    }

}
if(score>0){explain.style= "font-size: 70px;"; explain.innerText= "Your best so far: " +
recordDur+ "ms";}
var playA= document.getElementById("A3Tone");
var playB= document.getElementById("B3Tone");
var playC= document.getElementById("C4Tone");
var playCsh= document.getElementById("Csh4Tone");
var playD= document.getElementById("D4Tone");
var playE= document.getElementById("E4Tone");
var playF= document.getElementById("F4Tone");
var playG= document.getElementById("G4Tone");
function eTone(){if(randNum<4){playE.play();}else{playG.play()}}
function dTone(){if(randNum<4){playD.play();}else{playF.play()}}
```

```
function cshTone(){if(randNum<4){playCsh.play();}else{playE.play()}}
function bTone(){if(randNum<4){playB.play();}else{playD.play()}}
function aTone(){if(randNum<4){playA.play();}else{playC.play()}}
randNo();
 </script>
 </body>
 </html>
var duration = 1000;
function changeStim(){
switch(duration){
   case 1000:
sounds[0].src= "Stimuli/49991012.wav";
sounds[1].src= "Stimuli/49991014.wav";
sounds[2].src= "Stimuli/49991015.wav";
sounds[3].src= "Stimuli/49991017.wav";
break;
case 500:
sounds[0].src= "Stimuli/45001012.wav";
sounds[1].src= "Stimuli/45001014.wav";
sounds[2].src= "Stimuli/45001015.wav";
sounds[3].src= "Stimuli/45001017.wav";
break;
case 250:
sounds[0].src= "Stimuli/42501012.wav";
sounds[1].src= "Stimuli/42501014.wav";
sounds[2].src= "Stimuli/42501015.wav";
sounds[3].src= "Stimuli/42501017.wav";
break;
case 125:
sounds[0].src= "Stimuli/41251012.wav";
sounds[1].src= "Stimuli/41251014.wav";
sounds[2].src= "Stimuli/41251015.wav";
sounds[3].src= "Stimuli/41251017.wav";
break;
case 70:
sounds[0].src= "Stimuli/40701012.wav";
sounds[1].src= "Stimuli/40701014.wav";
sounds[2].src= "Stimuli/40701015.wav";
sounds[3].src= "Stimuli/40701017.wav";
break;
case 35:
sounds[0].src= "Stimuli/40351012.wav";
sounds[1].src= "Stimuli/40351014.wav";
sounds[2].src= "Stimuli/40351015.wav";
sounds[3].src= "Stimuli/40351017.wav";
break;
case 20:
sounds[0].src= "Stimuli/40201012.wav";
```

```
sounds[1].src= "Stimuli/40201014.wav";
sounds[2].src= "Stimuli/40201015.wav";
sounds[3].src= "Stimuli/40201017.wav";
 break;
}
}
var duration = 1000;
function changeStim(){
switch(duration){
   case 1000:
sounds[0].src= "Stimuli/49991012.wav";
sounds[1].src= "Stimuli/49991014.wav";
sounds[2].src= "Stimuli/49991015.wav";
sounds[3].src= "Stimuli/49991017.wav";
sounds[4].src= "Stimuli/49991315.wav";
sounds[5].src= "Stimuli/49991317.wav";
sounds[6].src= "Stimuli/49991318.wav";
sounds[7].src= "Stimuli/49991320.wav";
sounds[8].src= "Stimuli/49990204.wav";
sounds[9].src= "Stimuli/49990206.wav";
sounds[10].src= "Stimuli/49990207.wav";
sounds[11].src= "Stimuli/49990209.wav";
sounds[12].src= "Stimuli/49990709.wav";
sounds[13].src= "Stimuli/49990711.wav";
sounds[14].src= "Stimuli/49990712.wav";
sounds[15].src= "Stimuli/49990714.wav";
sounds[16].src= "Stimuli/49991719.wav";
sounds[17].src= "Stimuli/49991721.wav";
sounds[18].src= "Stimuli/49991722.wav";
sounds[19].src= "Stimuli/49991724.wav";
break;
case 500:
sounds[0].src= "Stimuli/45001012.wav";
sounds[1].src= "Stimuli/45001014.wav";
sounds[2].src= "Stimuli/45001015.wav";
sounds[3].src= "Stimuli/45001017.wav";
sounds[4].src= "Stimuli/45001315.wav";
sounds[5].src= "Stimuli/45001317.wav";
sounds[6].src= "Stimuli/45001318.wav";
sounds[7].src= "Stimuli/45001320.wav";
sounds[8].src= "Stimuli/45000204.wav";
sounds[9].src= "Stimuli/45000206.wav";
sounds[10].src= "Stimuli/45000207.wav";
sounds[11].src= "Stimuli/45000209.wav";
sounds[12].src= "Stimuli/45001214.wav";
sounds[13].src= "Stimuli/45001216.wav";
sounds[14].src= "Stimuli/45001217.wav";
sounds[15].src= "Stimuli/45001219.wav";
```

```
sounds[16].src= "Stimuli/45001719.wav";
sounds[17].src= "Stimuli/45001721.wav";
sounds[18].src= "Stimuli/45001722.wav";
sounds[19].src= "Stimuli/45001724.wav";
break;
case 250:
sounds[0].src= "Stimuli/42501012.wav";
sounds[1].src= "Stimuli/42501014.wav";
sounds[2].src= "Stimuli/42501015.wav";
sounds[3].src= "Stimuli/42501017.wav";
sounds[4].src= "Stimuli/42501315.wav";
sounds[5].src= "Stimuli/42501317.wav";
sounds[6].src= "Stimuli/42501318.wav";
sounds[7].src= "Stimuli/42501320.wav";
sounds[8].src= "Stimuli/42500406.wav";
sounds[9].src= "Stimuli/42500408.wav";
sounds[10].src= "Stimuli/42500409.wav";
sounds[11].src= "Stimuli/42500411.wav";
sounds[12].src= "Stimuli/42500810.wav";
sounds[13].src= "Stimuli/42500812.wav";
sounds[14].src= "Stimuli/42500813.wav";
sounds[15].src= "Stimuli/42500815.wav";
sounds[16].src= "Stimuli/42501618.wav";
sounds[17].src= "Stimuli/42501620.wav";
sounds[18].src= "Stimuli/42501621.wav";
sounds[19].src= "Stimuli/42501623.wav";
break;
case 125:
sounds[0].src= "Stimuli/41251012.wav";
sounds[1].src= "Stimuli/41251014.wav";
sounds[2].src= "Stimuli/41251015.wav";
sounds[3].src= "Stimuli/41251017.wav";
sounds[4].src= "Stimuli/41251315.wav";
sounds[5].src= "Stimuli/41251317.wav";
sounds[6].src= "Stimuli/41251318.wav";
sounds[7].src= "Stimuli/41251320.wav";
sounds[8].src= "Stimuli/41250507.wav";
sounds[9].src= "Stimuli/41250509.wav";
sounds[10].src= "Stimuli/41250510.wav";
sounds[11].src= "Stimuli/41250512.wav";
sounds[12].src= "Stimuli/41250911.wav";
sounds[13].src= "Stimuli/41250913.wav";
sounds[14].src= "Stimuli/41250914.wav";
sounds[15].src= "Stimuli/41250916.wav";
sounds[16].src= "Stimuli/41251517.wav";
sounds[17].src= "Stimuli/41251519.wav";
sounds[18].src= "Stimuli/41251520.wav";
sounds[19].src= "Stimuli/41251522.wav";
```

```
break;
case 70:
sounds[0].src= "Stimuli/40701012.wav";
sounds[1].src= "Stimuli/40701014.wav";
sounds[2].src= "Stimuli/40701015.wav";
sounds[3].src= "Stimuli/40701017.wav";
sounds[4].src= "Stimuli/40701315.wav";
sounds[5].src= "Stimuli/40701317.wav";
sounds[6].src= "Stimuli/40701318.wav";
sounds[7].src= "Stimuli/40701320.wav";
sounds[8].src= "Stimuli/40700608.wav";
sounds[9].src= "Stimuli/40700610.wav";
sounds[10].src= "Stimuli/40700611.wav";
sounds[11].src= "Stimuli/40700613.wav";
sounds[12].src= "Stimuli/40701113.wav";
sounds[13].src= "Stimuli/40701115.wav";
sounds[14].src= "Stimuli/40701116.wav";
sounds[15].src= "Stimuli/40701118.wav";
sounds[16].src= "Stimuli/40701820.wav";
sounds[17].src= "Stimuli/40701822.wav";
sounds[18].src= "Stimuli/40701823.wav";
sounds[19].src= "Stimuli/40701825.wav";
break;
case 35:
sounds[0].src= "Stimuli/40351012.wav";
sounds[1].src= "Stimuli/40351014.wav";
sounds[2].src= "Stimuli/40351015.wav";
sounds[3].src= "Stimuli/40351017.wav";
sounds[4].src= "Stimuli/40351315.wav";
sounds[5].src= "Stimuli/40351317.wav";
sounds[6].src= "Stimuli/40351318.wav";
sounds[7].src= "Stimuli/40351320.wav";
sounds[8].src= "Stimuli/40350507.wav";
sounds[9].src= "Stimuli/40350509.wav";
sounds[10].src= "Stimuli/40350510.wav";
sounds[11].src= "Stimuli/40350512.wav";
sounds[12].src= "Stimuli/40350911.wav";
sounds[13].src= "Stimuli/40350913.wav";
sounds[14].src= "Stimuli/40350914.wav";
sounds[15].src= "Stimuli/40350916.wav";
sounds[16].src= "Stimuli/40351416.wav";
sounds[17].src= "Stimuli/40351418.wav";
sounds[18].src= "Stimuli/40351419.wav";
sounds[19].src= "Stimuli/40351421.wav";
break;
case 20:
sounds[0].src= "Stimuli/40201012.wav";
sounds[1].src= "Stimuli/40201014.wav";
```

```
sounds[2].src= "Stimuli/40201015.wav";
sounds[3].src= "Stimuli/40201017.wav";
sounds[4].src= "Stimuli/40201315.wav";
sounds[5].src= "Stimuli/40201317.wav";
sounds[6].src= "Stimuli/40201318.wav";
sounds[7].src= "Stimuli/40201320.wav";
sounds[8].src= "Stimuli/40200709.wav";
sounds[9].src= "Stimuli/40200711.wav";
sounds[10].src= "Stimuli/40200712.wav";
sounds[11].src= "Stimuli/40200714.wav";
sounds[12].src= "Stimuli/40201416.wav";
sounds[13].src= "Stimuli/40201418.wav";
sounds[14].src= "Stimuli/40201419.wav";
sounds[15].src= "Stimuli/40201421.wav";
sounds[16].src= "Stimuli/40201820.wav";
sounds[17].src= "Stimuli/40201822.wav";
sounds[18].src= "Stimuli/40201823.wav";
sounds[19].src= "Stimuli/40201825.wav";
 break;
}
}

     body{background-color:rgb(59, 110, 165)}
     #head{
       width:500px;
       position: relative;
       background-color:blueviolet;
       margin: auto;
       text-align: center;
       top:-100px;
   }
   #intro{
       background-color:black;
       color:greenyellow;
       width: 450px;
       font-size:60px;
       padding: 5px;
       margin: 10px;
   }
#backGround{
    background-color:rgb(63, 57, 153);
    position: absolute;
    top: 0%;
    bottom:0%;
    left:0%;
    right:0%;
    margin: auto;
    width:960px;
```

```
    height:640px;
}
#marimba{
     display: flex;
     flex-direction:column;
     position: relative;
     left:835px;
     top: -500px;
     align-items:center;
     justify-content:space-evenly;
    background-color: black;
    width: 110px;
    height: 600px;
}
.bars{
    background-image: url("marimbaBarH.png");
    width:70px;
   font-size:40px;
}
#ansButtons{
     display: flex;
     flex-direction:column;
     position: relative;
     left:65%;
     top: -35px;
    background-color: red;
    width: 180px;
    height:440px;
    order: -1;
    border-style: solid;
    border-color: yellow;
    border-width: 5px;
}
.ansButton{
    background-color:darkgoldenrod;
    width:150px;
    height:110px;
   font-size:50px;
   position: relative;
}
    #fifth{ background-color:aqua;   left: 20px; top: 10px;}
   #fourth{ background-color:rgb(3, 151, 250);  left: 20px; top: 20px;}
   #third{background-color:rgb(47, 6, 230);color:white;display:none;  left: 20px; top: 30px;}
   #second{background-color:darkblue;color:white;display:none;  left: 20px; top: 40px;}
   #duration{
    position: absolute;
       left: 50px;
       top:20px;
```

```
        background-color:black;
        color:greenyellow;
        width: 40%;
        height:15%;
        font-size:70px;
        padding: 10px;
        margin: 10px;
    }
    #explain{
        background-color:black;
        color:greenyellow;
        position: absolute;
        left: 50px;
        top:170px;
        width: 500px;
        height:200px;
        font-size:20px;
        padding:10px;
        resize:none;
        overflow: hidden;
    }
    #results{
     background-color:black;
     color:greenyellow;
     position: absolute;
     top:500px;
     left: 30px;
     font-size:40px;
        padding:10px;
        width:550px;
        min-height:140px;
        resize:none;
        visibility:visible;
}
#starter{
        position: absolute;
        top: 385px;
       left: 450px;
        background-color:black;
        color:greenyellow;
        width: 200px;
        min-height:100px;
        font-size:70px;
        resize:none;
    }
    #autoStart{
     position: absolute;
     top: 400px;
```

```
  left: 240px;
   background-color:black;
   color:greenyellow;
   width: 200px;
   min-height:70px;
   font-size:20px;
   resize:none;
   visibility:hidden;
}
  #backToBoard{
   position: absolute;
   top:640px;
  left:0px;
  right:0px;
  margin:auto;
   width: 1000px;
   height: 60px;
   visibility:visible;
}
.levels{
   position: relative;
   width:150px;
   height:50px;
   background-color:rgb(243, 132, 104);
   font-size:20px;
   left: 90px;
   right:0px;
   margin: auto;
}
  textarea{resize: none;}
#saveClose{
   position: fixed;
   right: 40px; top: 40px;
   width: 250px;
   height: 120px;
   font-size: 30px;
   background-color: rgb(237, 41, 57);
}

    body{background-color:rgb(59, 110, 165)}
    #head{
      width:500px;
      position: relative;
      background-color:blueviolet;
      margin: auto;
      text-align: center;
      top:-100px;
  }
```

```
  #intro{
      background-color:black;
      color:greenyellow;
      width: 450px;
      font-size:60px;
      padding: 5px;
      margin: 10px;
  }
#backGround{
    background-color:rgb(63, 57, 153);
    position: absolute;
    top: 0%;
    bottom:0%;
    left:0%;
    right:0%;
    margin: auto;
    width:960px;
    height:640px;
}
#marimba{
     display: flex;
     flex-direction:column;
     position: relative;
     left:835px;
     top: -500px;
     align-items:center;
     justify-content:space-evenly;
    background-color: black;
    width: 110px;
    height: 600px;
}
.bars{
    background-image: url("marimbaBarH.png");
    width:70px;
   font-size:40px;
}
#ansButtons{
     display: flex;
     flex-direction:column;
     position: relative;
     left:70%;
     top: -35px;
     background-color: red;
     width: 180px;
     height:440px;
     order: -1;
     border-style: solid;
     border-color: yellow;
```

```
    border-width: 5px;
}
.ansButton{
   background-color:darkgoldenrod;
   width:150px;
   height:110px;
   margin: auto;
  font-size:40px;
}
   #fifth{ background-color:aqua;   }
  #fourth{ background-color:rgb(3, 151, 250);}
  #third{background-color:rgb(47, 6, 230);color:white;visibility:hidden;}
  #second{background-color:darkblue;color:white;}
  #duration{
   position: absolute;
     left: 50px;
     top:20px;
     background-color:black;
     color:greenyellow;
     width: 40%;
     height:15%;
     font-size:70px;
     padding: 10px;
     margin: 10px;
  }
  #explain{
     background-color:black;
     color:greenyellow;
     position: absolute;
     left: 50px;
     top:170px;
     width: 500px;
     height:200px;
     font-size:20px;
     padding:10px;
     resize:none;
     overflow: hidden;
  }
  #results{
   background-color:black;
   color:greenyellow;
   position: absolute;
   top:500px;
   left: 30px;
   font-size:40px;
     padding:10px;
     width:550px;
     min-height:140px;
```

```
       resize:none;
       visibility:visible;
}
#starter{
       position: absolute;
       top: 385px;
      left: 450px;
       background-color:black;
       color:greenyellow;
       width: 200px;
       min-height:100px;
       font-size:70px;
       resize:none;
   }
   #autoStart{
    position: absolute;
    top: 400px;
   left: 240px;
    background-color:black;
    color:greenyellow;
    width: 200px;
    min-height:70px;
    font-size:20px;
    resize:none;
    visibility:hidden;
}

   #backToBoard{
    position: absolute;
    top:640px;
   left:0px;
   right:0px;
   margin:auto;
    width: 1000px;
    height: 60px;
    visibility:visible;
}
.levels{
    position: relative;
    width:150px;
    height:50px;
    background-color:rgb(243, 132, 104);
    font-size:20px;
    left: 90px;
    right:0px;
    margin: auto;
}
   textarea{resize: none;}
```

```
  #saveClose{
   position: fixed;
   right: 40px; top: 40px;
   width: 250px;
   height: 120px;
   font-size: 30px;
   background-color: rgb(237, 41, 57);
}
<?php
$servername = "localhost";
$username = "interval_fred";
$password = "Gs[+[2X2f%OI";
$database = "interval_testdb";
if (isset($_POST["idin"])) {
$idin = intval($_POST["idin"]);
$studyNo = intval($_POST["studyNo"]);
$lowDur = intval($_POST["lowDur"]);
$lowestDur = intval($_POST["lowestDur"]);
// Create connection
$mysqli = new mysqli($servername, $username, $password, $database);
//MySqli Insert Query
$insert_row = $mysqli->prepare("INSERT INTO auraltrain(Participant, activityNo,
lowDuration) VALUES(?, ?, ?)");
$insert_row->bind_param("iii", $idin, $studyNo, $lowDur);
$insert_row->execute();
if ($lowestDur<1000){
switch($studyNo){
   case 1:
      $updt="UPDATE participant SET single2= $lowDur WHERE participantNo= $idin";
      if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else {
echo "Error updating record: " . $mysqli->error; }
       break;

   case 2:
      $updt="UPDATE participant SET single3= $lowDur WHERE participantNo= $idin";
      if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else {
echo "Error updating record: " . $mysqli->error; }
       break;

   case 3:
      $updt="UPDATE participant SET single4= $lowDur WHERE participantNo= $idin";
      if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else {
echo "Error updating record: " . $mysqli->error; }
       break;

   case 4:
      $updt="UPDATE participant SET double2= $lowDur WHERE participantNo= $idin";
```

```
        if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else {
echo "Error updating record: " . $mysqli->error; }
        break;

    case 5:
        $updt="UPDATE participant SET double3= $lowDur WHERE participantNo= $idin";
        if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else {
echo "Error updating record: " . $mysqli->error; }
        break;

    case 6:
        $updt="UPDATE participant SET double4= $lowDur WHERE participantNo= $idin";
        if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else {
echo "Error updating record: " . $mysqli->error; }
        break;

    case 7:
        $updt="UPDATE participant SET mixed2= $lowDur WHERE participantNo= $idin";
        if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else {
echo "Error updating record: " . $mysqli->error; }
        break;
    case 8:
        $updt="UPDATE participant SET mixed3= $lowDur WHERE participantNo= $idin";
        if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else {
echo "Error updating record: " . $mysqli->error; }
        break;
    case 9:
        $updt="UPDATE participant SET mixed4= $lowDur WHERE participantNo= $idin";
        if ($mysqli->query($updt) === TRUE) { echo "Record updated successfully"; } else {
echo "Error updating record: " . $mysqli->error; }
        break;
}}
$mysqli->close();
}
?>

     body{background-color:rgb(59, 110, 165)}
     #head{
       width:500px;
       position: relative;
       background-color:blueviolet;
       margin: auto;
       text-align: center;
       top:-100px;
    }
    #intro{
       background-color:black;
       color:greenyellow;
```

```
      width: 450px;
      font-size:60px;
      padding: 5px;
      margin: 10px;
   }
#backGround{
    background-color:rgb(63, 57, 153);
    position: absolute;
    top: 0%;
    bottom:0%;
    left:0%;
    right:0%;
    margin: auto;
    width:960px;
    height:640px;
}
#marimba{
     display: flex;
     flex-direction:column;
     position: relative;
     left:835px;
     top: -500px;
     align-items:center;
     justify-content:space-evenly;
    background-color: black;
    width: 110px;
    height: 600px;
}
.bars{
    background-image: url("marimbaBarH.png");
    width:70px;
   font-size:40px;
}
#ansButtons{
     display: flex;
     flex-direction:column;
     position: relative;
     left:70%;
     top: -35px;
    background-color: black;
    width: 110px;
    height:440px;
    background-color: red;
    width: 180px;
    height:440px;
    order: -1;
    border-style: solid;
    border-color: yellow;
```

```
    border-width: 5px;
}
.ansButton{
    background-color:darkgoldenrod;
    width:150px;
    height:110px;
    margin: auto;
   font-size:40px;
}
    #fifth{ background-color:aqua;   }
   #fourth{ background-color:rgb(3, 151, 250);}
   #third{background-color:rgb(47, 6, 230);color:white;}
   #second{background-color:darkblue;color:white;}
   #duration{
    position: absolute;
       left: 50px;
       top:20px;
       background-color:black;
       color:greenyellow;
       width: 40%;
       height:15%;
       font-size:70px;
       padding: 10px;
       margin: 10px;
   }
   #explain{
       background-color:black;
       color:greenyellow;
       position: absolute;
       left: 50px;
       top:170px;
       width: 500px;
       height:200px;
       font-size:20px;
       padding:10px;
       resize:none;
       overflow: hidden;
   }
   #results{
    background-color:black;
    color:greenyellow;
    position: absolute;
    top:500px;
    left: 30px;
    font-size:40px;
       padding:10px;
       width:550px;
       min-height:140px;
```

```
        resize:none;
        visibility:visible;
}
#starter{
        position: absolute;
        top: 385px;
       left: 450px;
        background-color:black;
        color:greenyellow;
        width: 200px;
        min-height:100px;
        font-size:70px;
        resize:none;
   }
   #autoStart{
    position: absolute;
    top: 400px;
   left: 240px;
    background-color:black;
    color:greenyellow;
    width: 200px;
    min-height:70px;
    font-size:20px;
    resize:none;
    visibility:hidden;
}
   #backToBoard{
    position: absolute;
    top:640px;
   left:0px;
   right:0px;
   margin:auto;
    width: 1000px;
    height: 60px;
    visibility:visible;
}
.levels{
    position: relative;
    width:150px;
    height:50px;
    background-color:rgb(243, 132, 104);
    font-size:20px;
    left: 90px;
    right:0px;
    margin: auto;
}
   textarea{resize: none;}
   #saveClose{
```

```
   position: fixed;
   right: 40px; top: 40px;
   width: 250px;
   height: 120px;
   font-size: 30px;
   background-color: rgb(237, 41, 57);
}
var duration = 1000;
function changeStim(){
switch(duration){
   case 1000:
sounds[0].src= "Stimuli/49991012.wav";
sounds[1].src= "Stimuli/49991014.wav";
sounds[2].src= "Stimuli/49991015.wav";
sounds[3].src= "Stimuli/49991017.wav";
sounds[4].src= "Stimuli/49991315.wav";
sounds[5].src= "Stimuli/49991317.wav";
sounds[6].src= "Stimuli/49991318.wav";
sounds[7].src= "Stimuli/49991320.wav";
break;
case 500:
sounds[0].src= "Stimuli/45001012.wav";
sounds[1].src= "Stimuli/45001014.wav";
sounds[2].src= "Stimuli/45001015.wav";
sounds[3].src= "Stimuli/45001017.wav";
sounds[4].src= "Stimuli/45001315.wav";
sounds[5].src= "Stimuli/45001317.wav";
sounds[6].src= "Stimuli/45001318.wav";
sounds[7].src= "Stimuli/45001320.wav";
break;
case 250:
sounds[0].src= "Stimuli/42501012.wav";
sounds[1].src= "Stimuli/42501014.wav";
sounds[2].src= "Stimuli/42501015.wav";
sounds[3].src= "Stimuli/42501017.wav";
sounds[4].src= "Stimuli/42501315.wav";
sounds[5].src= "Stimuli/42501317.wav";
sounds[6].src= "Stimuli/42501318.wav";
sounds[7].src= "Stimuli/42501320.wav";
break;
case 125:
sounds[0].src= "Stimuli/41251012.wav";
sounds[1].src= "Stimuli/41251014.wav";
sounds[2].src= "Stimuli/41251015.wav";
sounds[3].src= "Stimuli/41251017.wav";
sounds[4].src= "Stimuli/41251315.wav";
sounds[5].src= "Stimuli/41251317.wav";
sounds[6].src= "Stimuli/41251318.wav";
```

```
sounds[7].src= "Stimuli/41251320.wav";
break;
case 70:
sounds[0].src= "Stimuli/40701012.wav";
sounds[1].src= "Stimuli/40701014.wav";
sounds[2].src= "Stimuli/40701015.wav";
sounds[3].src= "Stimuli/40701017.wav";
sounds[4].src= "Stimuli/40701315.wav";
sounds[5].src= "Stimuli/40701317.wav";
sounds[6].src= "Stimuli/40701318.wav";
sounds[7].src= "Stimuli/40701320.wav";
break;
case 35:
sounds[0].src= "Stimuli/40351012.wav";
sounds[1].src= "Stimuli/40351014.wav";
sounds[2].src= "Stimuli/40351015.wav";
sounds[3].src= "Stimuli/40351017.wav";
sounds[4].src= "Stimuli/40351315.wav";
sounds[5].src= "Stimuli/40351317.wav";
sounds[6].src= "Stimuli/40351318.wav";
sounds[7].src= "Stimuli/40351320.wav";
break;
case 20:
sounds[0].src= "Stimuli/40201012.wav";
sounds[1].src= "Stimuli/40201014.wav";
sounds[2].src= "Stimuli/40201015.wav";
sounds[3].src= "Stimuli/40201017.wav";
sounds[4].src= "Stimuli/40201315.wav";
sounds[5].src= "Stimuli/40201317.wav";
sounds[6].src= "Stimuli/40201318.wav";
sounds[7].src= "Stimuli/40201320.wav";
 break;
}
}
```