Making a game of chess sound

By Elizabeth Catharina du Toit

Thesis presented in fulfilment of the requirements for the degree of Master of Music at Stellenbosch University

Supervisor: Dr Ralf Alexander Kohler

December 2016

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the authorship owner thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Signature: ....

Date: ....

Abstract

This thesis discusses the correlations between chess and music via a mathematical basis. The reason for this study was to explore the ontological status of Western Classical Music representation by substituting it with the representation of chess, in order to see whether Western Classical Music could be represented in other logical systems. This study resulted in the sounding of the game of chess. It explores the music notation by comparing it to and exchanging it with chess representation. The way in which this was accomplished was by creating musical variables to represent chess moves, notation and other entities of chess. Thereafter the sounds were implemented into a computer programme which was created in the Java language.

The variables were coded into the programme which was created for this thesis and was named *The Sound of Chess Programme.* Three chess games including Scholars Mate, an intermediate game and lastly a game by Kasparov were coded and made to sound via *The Sound of Chess Programme.* The sounds of all three games were analysed. In the final sounds one could hear positional structure of the chess games, as well as which chess pieces were moving.

The results of this study showed that any non-musical logical system can represent musical phenomena. This means that 21st century musicians may have many representational systems at their disposal via the use of various logical systems which could be made to sound. The addition of sound to logical representations can also add new methods to understanding of musical systems, as by hearing the entities one can interact with the content differently.

Opsomming

Hierdie navorsing ondersoek die ooreenkomste tussen skaak en musiek deur gebruik van ’n wiskundige raamwerk. Die rede vir hierdie studie is om die ontologiese status van musiekvoorstelling te deurgrond deur dit te vervang met die voorstelling van skaak om sodanig vas te stel of musiek deur ander logiese sisteme voorgestel kan word. Hierdie studie het gelei tot die klank van skaak. Standaard Westerse Musieknotasie is vergelyk en verruil met skaaknotasie. Musiekveranderlikes is geskep om skaakskuiwe, skaaknotasie en ander elemente van skaak voor te stel, waarna klanke toegevoeg is tot ‘n Java rekenaarprogramtaal.

Die veranderlikes is in die program *The Sound of Chess Programme*, wat vir hierdie studie ontwerp is, ingevoer. Drie skaakspelle insluitende “Scholars Mate”, ’n intermediêre program en ’n spel van Kasparov is gekodeer en verklank deur die program.

Die klankbaan van al drie spelle is ontleed. Posisionele strukture, skuiwe en identiteite van die stukke kon in die klanke waargeneem word.

Die resultate van die studie toon dat enige nie-musikale logiese sisteem musiekeienskappe kan oordra. Dit beteken dat 21ste eeuse musici deur die gebruik van logiese sisteme wat verklank word, verskeie voorstellingstegnieke tot hul beskikking het.

Deur na die eienskappe van die klanke te luister, kan daar op verskeie maniere interaksie met die inhoud plaasvind.

Acknowledgments

I would like to express my sincere gratitude to my supervisor Dr Ralf Alexander Kohler for the continuous support of my MMus research study as well as his motivation and the vast contribution of his knowledge. I have thoroughly enjoyed lectures and supervisions with Dr Kohler and I have learned copious numbers of new study, research and writing methods.

My sincere thanks also go to Misha Beare and Nicholas Hilarides who provided me with many hours of help and guidance with regards to the computer programming for this study.

I would like to thank Prof Caroline van Niekerk, Badja Zaacks and Jennifer Ritchie for assistance with the editing of my thesis.

Thank you to my family: my parents, aunts, uncles, godfather and grandfather for supporting me spiritually and financially throughout the writing of my thesis and for always encouraging me with regards to my music and chess endeavours.

Lastly, I am immensely appreciative to Wendy Ackerman for generously paying for my Masters Degree. Learning is the most incredible gift that one can receive and I thank her greatly.

Table of Contents

List of tables

Chapter 1

Introduction and personal motivation, statement of the problem, theoretical reference points, methodology and chapter outline

1.1 Introduction and personal motivation

I was exposed to both chess and music before I started school. Over the years the similarities between chess and music became more apparent to me, although I was troubled by the fact that throughout my music studies I could never three-dimensionally touch a note. For example, I could never pick up middle C and move it around, as I could do with a piece on the chess board. I continued to look for more correlations between the two entities as I progressed to secondary and tertiary level education. The reason I have added this personal data to my introduction is to contextualise the beginning of my journey for the reader regarding this topic, because people throughout this process have asked me what made me think of linking chess and music. Some of the correlations that I encountered between chess and music include the following: fundamental numerical ties, benefits of participation, their mathematical linkage and their logical basis (Godlovitch 2002 pp. 14-20, Vogel 2013 pp. 119-122).

Regarding the differences between the two entities, it should be noted that chess does not contain sound while Western Classical Music theory cannot be manipulated. Apart from these factors, there is very little that cannot be linked between chess and music. There are three fundamental links between chess and music. The first is neurological and the second is logical. The third link is that both align on a representative level due to the fact that each has its own notational disciplines (Wilson 2003 pp. 224-245, Chess-insights 2015, Siegel 2015 pp. 246-247).

In my second year at university, I always attended chess coaching after my aural lecture. This led me to the realisation that after attending aural training, my eye was already in focus on the chess board. I decided to swap my chess coaching to the hour before my aural lecture and I found that after attending chess, my ears could hear chordal progressions immediately. Focussing the optical nerves and focussing the ear both require the cerebral and pre-frontal cortices in the brain. Hence one can deduce that participation in both chess and music requires the same neurological processes. Furthermore, participating in a game of chess requires intense logical thinking. In a game of chess, the player at the beginning of the game is in complete control of his or her fate and nothing is left to luck (Hallman 2013 pp. 314-316). The player with the better logic, mind set, theory knowledge, creativity and intuition has the competitive edge (Wilson 2003 pp. 224-245, Chess-insights 2015, Siegel 2015 pp. 246-247).

A music performance correlates to the above elements in terms of what it requires and involves. The elements surrounding the performer are critical and directly impact the quality of a musical performance (Bell and Davis 2011 p. 46).

Logic is a sub-discipline of philosophy, which links logic to the study of thought. In the Greek language, the etymology of the word logic, “logos” means reasoning art, which can be taken literally when speaking about chess and music (Sfetcu 2014 pp. 6-8). The core of chess creation is logically based. Logical principles, too, are embedded with music and it is these two forms of logic that caught my interest (Bader 2009 pp. 181-198).

With regards to the aspect of representation, both chess notation and Standard Western Music Notation have adequacies and inadequacies. Adequacies include the following: Chess includes an infinite variety of moves that a player can make as well as the ability to notate those moves (Burgun 2012 pp. 41-43). Music includes the same infinite variations of sounds that can be used so that no melody has to sound the same. Furthermore, Standard Western Music Notation has the ability to show a large variety of information (Levin 2009 pp. 22-23).
Chess notation contains certain inadequacies. It cannot notate tensions between chess pieces and the passing through of other squares. For example, if a Pawn

moves from e2 to e4, its passing through or over e3 is not notated. Chess also lacks a notation system that can express the emotions of two opponents in order to understand their reasoning with regard to different moves made in a game.

Western Classical Music excludes many aspects of the sounding phenomenon. Standard Western Music Notation does not harness enough symbolism to accurately describe what the composer intended, which is why a musical score can be under-notated (Delaere and Crispin 2009 pp. 40-41).

According to Kuijken the practice of interpretation can easily misguide a musician, especially throughout time (Kuijken 2013 pp. 12-35). It is likely that if Bach had to listen to one of his cantatas today, he would not recognise his own work, as his cantatas were under-notated with regards to phrasing, dynamics and slurs (Goeth 2015). These are just a few examples of the adequacies and inadequacies in both disciplines. Obvious links exist to prove that chess and music are related neurologically; logically and via notation. A worthwhile question would be whether it would be possible to merge the two entities.

1.2 Chess and music

For this study it was important to understand the ways in which chess and music could be linked, as I needed to assess whether there were similarities between chess and music. The research only transfers' chess notation to musical sound and/or notation, it does not deal with music notation/composition (musical sound) to chess notation. Links would be used as variables to create a sounding chess game. If there were not enough ways in which to integrate the two entities, then it would not have been possible to create and integrate exchangeable variables for the end result of this thesis. This section explores some evidence of the alignment that exists between chess and music.

Leisure and music have a long standing co-existence that dates back to the Stone Age. Trigger, Washburn and Adams discuss the presence of games in the Stone Ages whilst Bulliet and Crossley explain how musical instruments were created and used during this time. Discoveries of musical instruments and board games prove that even in the time when man was at his most primitive, board games and music have existed alongside one another (Trigger, Washburn and Adams 1996 p. 7, Bulliet and Crossley 2011 pp. 38-39).

Relationships between games, music and mathematical concepts have existed throughout history. Linkage between the two entities took place as far back as the time of the mathematician Pythagoras, who discovered ratios in music. He used these ratios to develop the octave scale. Pythagoras believed music to be a science (Taruskin 2010 pp. 60-63).

The correlations between music and mathematics as well as between mathematics and chess have been scrutinised separately. Both chess and music share logical elements which are tied to mathematics. According to Petrović, chess is a logical game that is supported by a vast mathematical core (Petrović 1997 pp. 1-2).

In a game of chess one is faced with many options and choices: the aim is to find the best outcome. This requires a logical interaction with each choice that the player faces. When composing music we are also faced with many options and choices: however, when performing Western Classical Music the notation is rather exact in pitch and rhythm, but flexible to a degree with tempo and dynamics. Chess like music also requires creative thinking. Horowitz and Rothenberg explain that creativity and originality can be taught through chess (Boom and Rychtar 2013 pp. 1-3, Horowitz and Rothenberg 2015 pp. 64-75).

Studies show that when comparing advanced chess players to beginners their memories differed according to the situation. When both were faced with remembering a randomly set up position, their memories and position recall tested the same: however, when faced with memorising games from start to finish the chess masters tested above average. This demonstrated that novice chess players are using pattern recognition rather than their strategic knowledge.

In fact, once any chess player has cultivated the skill of identifying sequences and pattern, strategic thinking plays a small role with regards to winning a chess game (Sharma 2005 pp. 177-178, de Groot 2008 pp. 8-33).

Griggs (2013) explains that in chess there are two decision making factors when making a move: the first is pattern recognition and the second is calculation. A master in chess uses 40% calculation and 60% pattern recognition when playing, whereas a beginner chess player uses only 5% pattern recognition and 95% calculation when moving a piece. According to Schwartz and Griffin a chess player also requires good intuition in order to gain positional advantages (Schwartz and Griffin 2012 pp. 176-180, Griggs 2013).

Griggs reflects this data onto composers of music, and he theorises that a master composer has more experience in sound pattern recognition than a beginner composer. One could deduce from this that the art of music is also based on various pattern recognitions (Griggs 2013).

The relationships that musicians and chess players can see, in terms of their knowledge of pattern recognition in their fields, is what sets them apart. A beginner musician tends to deal with composition one note at a time, while a beginner chess player plans one move at a time. Neither sees the bigger picture as they are deciphering the smaller movements and changes (López 2007 p. 80, Latour and Woolgar 2013 pp. 248-249).

Both Gladwell and Griggs state that, in order to be considered a master in one's field, it takes an average of ten years or 10 000 hours of consistent study (Gladwell 2008 pp. 40-41, Griggs 2013).

Western Classical Music can be seen in two parts-the technical part and the creative part. Musicians require both mathematics and logic to be technically competent. In order to create Western Classical Music within stylistically correct limits requires an application of analytical and logical thinking (du Sautoy 2011, Assayag 2013 pp. 93-103).

A quotation by nineteenth century mathematician J.J. Sylvester puts the relationship between music and mathematics into perspective: *May not Music be described as the Mathematic of sense, Mathematic as Music of the reason? The soul of each the same!* (Harkleroad 2009 pp. 1-3).

There are obvious and perhaps trivial correlations between chess and music. Chess and Western Classical Music are two logical systems that correlate in many different ways. Some similarities that I have encountered include the following: The 8x8 chess board layout and 8 notes to an octave scale; the 64 squares on a chess board and 64 pulses in Western Classical Music. There are an infinite number of different chess games and music melodies or chess compositions and music compositions, black ink notes on white paper, piano keys, black and white chess pieces and black and white squares. The same terminology, such as tempo, crescendo, dynamics and technique, is also used in both chess and Western Classical Music. Both chess and Western Classical Music have their own notational systems (Downs 2013 pp. 67-71, Czelakowski 2015 pp. 65-67).

Besides these superficial similarities, more tangible correlations include: harmony, composition, creation, time, space, relationships and emotion, and concentration. Benefits of chess and music are also on par as both stimulate optimum brain function, concentration and endurance, and creativity (Črnčen and Wilson 2006 pp. 579-594, Lawrence and Alburt 2010 pp. 2-4, Boden McGrill and King 2013 pp. 106-107).

Alloway and Alloway and Kaufman and Sternberg all agree that chess and music can raise a participant's IQ, especially in young children (Alloway and Alloway 2013 pp. 119-124, Kaufman 2013 pp. 236-237).

The two disciplines have simultaneously interested many of the same men and women. Famous music composers who were masters at chess as well as music include: Philidor, Schoenberg, and Schumann, who according to Daverio and Macdonald, taught Brahms how to play chess (Daverio 2002 p. 157, McClain 2010, Macdonald 2012 p. 131).

Prokofiev is also famous for having been a chess master and a musician. One of Prokofiev's greatest chess wins was against grandmaster José Raúl Capablanca in a rated chess game in 1914. This makes one eager to hear if he created chess positions as beautifully as he composed music (Prokofiev and Phillips 2008 p. 692, Linder 2013 pp. 62-66, Winter 2014).

Schoenberg created the four person chess game, commonly referred to as Schoenberg's Coalition Chess, in the 1920s, whilst being in the military during World War One. His chess game included pieces of four colours: yellow, black, red and green. Yellow and black consist of twelve pieces per player, whilst green and red consist of six pieces per player. Only the yellow and black colours have Kings. It is said that Schoenberg picked these colours because they were the colours of the flag that belonged to the Weimar Republic. Some of the characters of Schoenberg's chess pieces include motorcycles, machine guns, tanks, guards and planes, but the classic Queen and King remained unaltered. The game harnesses the basis of classical chess, but it is played on a ten by ten chess board, which has 100 squares, rather than an eight by eight chess board, which consists of 64 squares (Paquette 2007, Lin 2012 pp. 1-12).

1.3 Research statement

This study attempts to challenge the traditional representation of Western Classical Music in a playful thought experiment which will translate chess notation into Standard Western Music Notation in order to understand the ontological status of both disciplines. It will not investigate whether the two notational systems are mutually interchangeable; instead some parameters of chess will be matched to Western Classical Music.

The aim of the study is to investigate whether a game of chess could be made to sound through coding music variables logically in order to provide an accurate, rather than musically sounding representation of chess moves and characteristics.

1.4 Goals and objectives

The goals and objectives for this thesis are to gain insight surrounding the relationship between chess and music as well as the ontological status of both entities. It will be evident that other games or logical systems could be made to sound. It will be suggested to the reader that a sounding game of chess could aid easier music study by introducing an interactive dimension. Lastly, the reader will be able to hear acoustic representation of a chess game, which will suggest that a person could play chess by ear.

1.5 Theoretical framework

The concept of musical representation goes back to early records of man transcribing sounds and resultant music notation. Today the Standard Western Music Notation of the world is still evolving.

The demand for music representation can be traced back to Gregorian chant in the Medieval Era. During this time Western musicians were expected to perform and improvise using their understanding of monophonic chants and to set the mood for the task at hand, with their aural ability. The representational functions of music notation emanated from the desire to improve teaching and for documentation purposes, although after the inception of music notations, there was a build-up in interest with regards to developing the representations for broader, everyday use (Selfridge-Field 1997 pp. 3-5).

According to Albert Frantz (2015), most music students give up when doing aural training as they become lazy, given the principle that it is often intrinsic to human nature to seek the easier method, which is to read music notation. He explains the importance of developing the inner ear to hear pitches, prior to performing the physical action when playing one's instrument. Other problems that arise when music is not learnt by ear include: memory lapses, misunderstanding of the melodic line, interpretation problems, and believing that one sounds better than one really does. Frantz says that musicians cripple themselves when they do not develop their ear training and those teachers who do not develop their students' auditory music learning, "fail to create authentic musicians" (Frantz 2015).

Selfridge-Field does not explain the impact that external musical representations have had on the auditory abilities of musicians, but with all the continuous changes to music notation and fast moving technological developments, it is possible that musicians have become more dependent on their notation and music's internal mental representation. Thus one might conclude that aural semantics may be deteriorating in musicians.

Selfridge-Field described the development of music representation throughout history and discussed the changes in symbolisms and notational formats linked to modern computer technology. She predicted that these would become independent of hardware configurations and software processes. According to the author the recent history can be described as follows:

Selfridge-Field comments that the establishment in the late 1980s of the Musical Instrument Digital Interface (MIDI) made tools for musical sound accessible to all levels of musicians. According to Selfridge-Field this should be discussed in three concepts of musical information, namely: phonological, which pertains to all the entities of sound; graphical notation, which pertains to all notated representations of music; and semantic, which has to do with the interpretation of music.

Furthermore, she points out that at the heart of musical information are the attributes of sound and visual representation.

Selfridge-Field discusses issues that arise with regards to the way in which software processes musical information, as the software regularly makes automatic programming assumptions about the input values or music variables. Practicality is often an end goal in Standard Western Music Notation; however, certain aspects of computerised music programmes can also complicate or misinterpret the operational instructions that process music automatically.

Theories of processing order often determine the optimum format for musical information. Broadly speaking, all systems of representation have been created with a specific application in mind and these in turn have influenced basic design features.

Assumptions with regard to end use are what determine which elements of information will be considered vital. Selfridge-Field points out that Standard Western Musical Notation is not logically self-consistent: in fact visual grammar is open-ended, and context-dependent interpretations abound (Selfridge-Field 1997 pp. 5-20).

With regards to my research this meant that because representation contains undeniable relativity, representation of entities needed to be applied in accordance with what was considered most important for one's desired end goals. This pertained mainly to the way in which the representations of the pitch variables were implemented for the sounding chess games at the end of this study.

1.6 Methodology

This thesis is the result of a two year qualitative research study. Initially at the proposal stage of this project I wanted to demonstrate that chess and music could be integrated through mathematics. The idea was to prove that this could be done by representing chess moves into variables that could be plugged into a mathematical algorithm, which then resulted in sounding chess music. Thereafter I was going to implement the variables into three chess games, a beginner chess game, an intermediate chess game and an advanced chess game.

Prior to doing this I had to explore the correlations between chess and Western Classical Music in order to understand if such algorithms and variables could exist. After I found that chess and Western Classical Music linked in a broad spectrum of ways, especially mathematically, I investigated the ways in which both chess and Western Classical Music, as separate entities, had been represented throughout history. This made it possible to delineate the ways in which chess and Western Classical Music could be represented as one entity with regards to the end result. It also highlighted certain ways in which to best represent the chess moves into sound with regards to representations that are more likely not to change for a prolonged period of time in both disciplines.

After finding possible representations, I needed to explore ways in which to reflect and showcase the data via mathematics at the end of this study. In Chapter Two, whilst researching computerised music and computer chess, I realised that creating an algorithm in a computer programme would best reflect the final data for three specific reasons: Firstly, programming is a modern and ever growing mathematical art, which would link chess and Standard Western Music Notation through mathematics as per the initial plan. Secondly I could better understand the variables that were created, if I were able to hear them, manipulate them and interact with them on a practical level. Lastly this also made it possible to hear the three games at the end of my study.

It was important to choose a programming language in order to start coding the chess variables into music. As a result three different programming languages were investigated as possible applications, in order to code the final variables into a computer programme. These options were Lisp, Python and Java.

I learnt how to programme in all three computer programming languages, via the use of tutorials and assistance from acquaintances who are programmers. After this I decided to investigate which of the three programming languages would best suit the study. I needed a programming language which would best showcase the representation of the variables that were created. Other elements that were relevant for this project include: a programming language that could aid further future developments of the final code; the chosen programming language needed to be popular among both chess informatics developers and music technology users, worldwide; I wanted a programming language that was cross-platform and easily accessible; and finally I wanted a programming language that could run software within a World Wide Web browser.

In accordance with the above wants and needs, the Java programming language was chosen as the front runner pertaining to this thesis. I coded the programme using jGRASP, which is an Integrated Development Environment that is used to code in the Java programming language.

After establishing a programming language and the Integrated Development Environment, I worked on creating an 8x8 grid where each co-ordinate was represented by an assigned pitch. This chess board grid made it possible to calculate the pitch variables for each move that needed to be implemented within the computer programme. After this the variables were designed and established according to which best represented the chess moves into sounds, within the jGRASP application. I coded each chess piece in isolation and I assigned each piece with six random moves in order to hear the intervals of the individual chess pieces. The final programme was then created and it was named *The Sound of Chess Programme*. In Chapter Four I discuss in detail where the variables were implemented within the code, as well as the functions of each line of code within *The Sound of Chess Programme*.

Three chess games were decided upon and the programme was then applied to the three games; the first game was Scholars Mate. The second chess game was by a Western Province Chess player, Anant Dole, against his opponent Sapar Batyrov, who is a chess grandmaster. The third game coded was by Garry Kasparov against the IBM chess computer which is most commonly known as Deep Blue. The variables of each chess game were calculated after which the variables were substituted into *The Sound of Chess Programme*.

The three sounding chess games were then compared and analysed and finally *The Sound of Chess Programme* was annotated. The five questions below were discussed, answered and concluded in the second half of Chapter Four:

- Could the sounding game of chess aid a new musical notation system and if so would such a system be easier than Standard Western Music Notation for musicians to use?
- Could other games or logical systems be used for themes in music composition?
- Could a sounding game of chess aid easier music study by introducing an interactive dimension?
- Could chess games be made to sound melodically?
- Could a person play chess by ear?

The contribution as well as the significance of the study was established. The conclusion discussed the main findings of the thesis as well as the reason for certain applications to the variables and methods that were used to create the final sounding chess games. The results and effects of the three sounding chess games regarding their audible characteristics in correlation with their positional happenings were portrayed. Limitations regarding the research were explored and recommendations for possible future developments with regards to the multifaceted areas of both chess and music were addressed.

Lastly, the thesis was concluded in accordance with findings and thoughts surrounding the existential features, which were explored in this project as well as other observations and outcomes that came to the fore regarding the interchanging of chess with musical representations.

1.7 Limitations of the research

The final sounding chess games revealed that there were certain entities that were lacking within sounds that could ultimately enhance the representational translations regarding the beauty of the game of chess. Although I could hear many of the positional happenings within the sounding chess games, the inclusion of other entities and adjustments of created variables could contribute to the auditory understanding of what is happening within the chess games.

Early on in my research I struggled to gain a clear understanding of how to implement rhythm within the variables. This was the main difficulty with regards to making the game of chess sound, as implementing rhythm is very important in Western Classical Music, but for the representation of chess, rhythm is not important. Thus I was caught between the importance of representing rhythm for music in order for chess to truly reflect musical variables, whilst at the same time risking the set structure of chess and ultimately misrepresenting the external function of rhythm within chess, which has no affiliation with the actual documentation of chess. As a result I decided to compromise by attempting to marry the idea of rhythm in chess with the necessity of rhythm in music.

In this study chess was made to sound, rather than made to inspire music, but assigning what is defined as musical variables to a logical system, such as chess, made it clear that there are elements in music notation which are too complex to define. It was established early on in this study that chess does not attempt to notate varied entities; instead it only notates the logical factors and in turn abandons external elements such as rhythm and individual intent. However, it was important that the linkage between the represented external factors in music be matched with the unaccounted for representations of the external forces that exist in chess. The merging of this metaphysical component within the new confinements of a computer programme helped me to better assess their representational importance.

The three chess games which were coded into *The Sound of Chess Programme* could be described as sounding primitively in rhythmic structure and tonal quality, although the pitches integrate well. The result of it sounding uninteresting and distant, with a computer tone quality, pleased me, as this proves that although the elements of aesthetics were accounted for with regards to the rhythm, the representation's end result via another non-living device reiterated that external variables are too complex to represent within man-made systems. What one could delineate from the final sounds is that representation is a simplified logic of all entities, which needs a human element to breathe life into it.

The end result of the final sounding chess games could have been random in sound or it could have aided no cognitive satisfaction. This was likely, but unsystematically based compositions are a common 21st century entity. There are under-notations in both chess and music, but for this study this reiterated that the way in which an under-notated piece of music or chess is interpreted depends solely on the performance and this is when its true meaning is defined.

It is clear that there are many similarities between the logic of chess and the logic of music, but accounting for the metaphysical factors of chess and music via the created variables did not contribute to the final sounds. It is difficult to deduce more information about the limitations of Western Classical Music Notation via the sounding chess games, as the variables of the games ultimately represent chess and not music and *The Sound of Chess Programme* is still at its basic stages of development.

This study has found one way of making a chess game sound instead of making a chess game sound musical and this sounding has very little to do with what music is.

1.8 Chapter outline

I. Introduction and personal motivation, statement of the problem, theoretical reference points, methodology and chapter outline

A Introduction and personal motivation

1 The same neurological processes are utilised when participating in chess and music

a Fundamental links between chess and music

2 There are possibilities of merging chess and music entities logically and via notation

a. Logical linkage

B Chess and music

1 Both chess and music can be linked via mathematics

a Chess and mathematics

b Music and mathematics

2 Knowledge of pattern recognition determines the level of competence in both chess and music. Participation in chess and music can raise IQ

a Participation and IQ

C Research statement

1 Parameters of chess are matched to Standard Western Music principles to determine whether a game of chess could sound

a Using music and chess entities for sound

2 The linkage of chess and mathematics as well as music and mathematics makes it possible to link both chess and music via mathematics and computer programming

a Coding can link music and chess

D Goals and objectives

1 Gaining insight into the value of making a game of chess sound as well as exploring if a person could play chess by ear or simplify music study by introducing an interactive dimension

a Benefits of sounding chess

b Possibilities

E Theoretical framework

1 Standard Western Classical Music Notation is still evolving.

a Evolving music

b Representation of pitch variables

F Methodology

1 The journey of the thesis

a The field work is discussed

b Creating the variables

c Choosing a programming language

d Implementing the variables

e Three chess games

f Creating the algorithm

g Five questions

h Analysing the results

G Limitations

1 Implementation of rhythm, and under-notations are identified

a Rhythm and under-notations

2 Interpretation defines performance in both chess and music

a. Interpretation

II. Correlations and comparisons between chess and music

A History of chess

1 Origin and developmental changes in chess provides guidance for exploring variables within the chess board

a Origin of chess

b Stages of development

B Standard Western Music Notation

1 The development of Standard Western Music Notation presented in four stages

a Gregorian Chant

b Tablature

c Current Standard Western Music Notation

d Computerised Music

2 A final analysis of the four stages of development of Standard Western Music Notation

a Music history contributes to creating variables

3 Restrictions regarding presentation and under-notation could influences interpretation

a Restrictions in representation

b under-notation and influences

C Chess notation

1 A brief history of chess notation

a Philip Stamma and Standard Algebraic Notation

b François Philidor and Standard Chess Notation

c Computerised chess notations

E Computer chess

1 History of computer chess engines and computer programmes that have linked chess and music

a The Turk

b Matthias Wüllenweber

c Fritz

d Ludwig

2 Three computer programming languages considered for *The Sound of Chess Programme*

a Chess moves to sounding variables

F Sounding Chess

1 Music has been linked to sound previously

a Chess being linked with sounds and music

b Compositions about chess

c Non algorithmically based correlations

III. Finding a conversion formula

A Computer programming languages

1 Studying the background of the three programming languages considered for the final computer programme

a Lisp, Python, Java

b Advantages of each programming language

c Disadvantages of each programming language

d Java works best for this study

B Creating the Variables

1 The Implementations of variables are explained

a Variables as it pertains to the chess board

b Variables as it pertains to the chess pieces

c Decision making process and aesthetics

d Representation of musical rhythm

C The South of Chess Programme

1 Coding the pitches and rhythmic variables as well as the coding of chess manipulations

a Pitches are coded into the board

b Rhythm and the aesthetics of chess

c Castling and check

D Three sounding chess games

1 The background regarding the three chosen chess games which were put to sound

a Scholars Mate the beginner game

b Anant Dole

c Kasperov's loss to Deep Blue

IV. Results: presentation and discussion

A The final *Sound of Chess Programme* with annotations

1 The code is annotated and explained in detail

a Annotations

b Explanations

B Five Questions

1 Looking at the advantages and disadvantages surrounding a sounding chess game and its impact on music study

a Musical chess notation

b Making other logical systems sound

c Chess's impact on music study

d Chess and melody

e Playing chess by ear

V. Conclusion and Recommendations

A Research statement and contribution

1 Sounding chess aids characteristics regarding the ontological status of both chess and music
  - a Chess and music are under-notated
  - b Chess represented in Standard Western Music Notation

B Main Findings

1 The sounds of all three chess games are analysed and the findings are discussed
  - a All representations are relative
  - b Reflection on the variables

2 Sound for capturing, castling and positional qualities could be identified within *The Sound of Music Programme*
  - a Unique sounds found

C Outlook

1 Visual and aural patterns were recognised in all three pieces and future adjustments
  - a Visual and aural patterns

2 Future adjustments and developments
  - a Future research

D Conclusion and Recommendations

1 Chess and music are compatible via logical linkage
  - a Endless variations
  - b Other logical systems can be linked

2 Limitations of the research
  - a Monotonous tones
  - b Lack of rhythmic elements
  - c Lack of variables

3 Exchanging sound entities with representational systems adds new dimensions to the interactions and understanding of various methodical systems

a Table of Elements

b 21st century composers

c Linking scientific concepts with musical phenomena

Chapter 2

Correlations and comparisons between chess and music

The content in Chapter Two expands on the relationships between chess and music. It explores ways in which to approach the creation of the variables that would represent chess moves as sound. This was done by investigating links between chess and music via the history of music and chess notation as well as the development of other chess and music representations. Possible ways in which to present the final variables for the end goal of this study were also discussed.

This chapter includes a brief historical overview of chess and music as well as an in-depth discussion about music notation, chess notation, and music composition software and computer chess engines.

2.1 History of chess

In this section the history and development of chess is discussed. The origin of chess has created much debate throughout the centuries. The development of the chess board as well as the changes in the movement of chess pieces seemed to have greatly impacted the way in which chess as a whole is represented globally today.

There are many theories about the origin of chess. Earlier research believed chess to have originated in China, prior than in India. Chess in Chinese is named Chaturanga, which is the original Western Indian name for the game. Cox Forbes was one of the theorists who believed chess to have originated from an earlier Chinese board game, namely Chaturaji (Forbes 1860 pp. 1-46). Another researcher, David Li, found that an earlier form of chess existed in 203BC, although there is various written evidence that contests Li's theories (Paul 1936 p. 1). According to Harold Murray, a researcher on the history of chess, all theories regarding the origin of the chess game were incorrect and he contested them in a book which he published in 1913. Murray's research found that chess did in fact originate in India, which is what most chess historians still teach today. He explained that chess later made its appearance in China where Chaturanga evolved into another form of chess named Xiangqi (Murray 2015 pp. 57-120). An interesting fact about Xiangqi is that the chess pieces were placed on top of the lines that intercepted one another on the chess board, instead of being placed on top of squares as in modern chess (Remus 2015).

Chess can be traced back to at least 500 AD, although the modern chess which is played today has been played since the sixth century. The game of chess has undergone many changes including: the names of the pieces; the directions in which the pieces move; the layout of the board and even the name "chess". The development can be seen in five stages from 700 AD up to the game of chess that is played today. These stages are: The Sanskrit Period, the Persian Period, the Arabic Period, the Medieval Period and the Modern Period (Bird 2010 pp. 7-85).

Davidson (2012) explains the five stages of chess development: The Sanskrit Period of Chess which existed in Western India was called Chaturanga and the pieces were as follows: Rajah (King), Mantri (Queen), Gaja was an elephant (Bishop), Asva (Knight), Ratha was a boat or chariot (Rook) and lastly the Padati was a foot soldier (Pawn).

In the Persian Period of Chess, it was known as both Chatrang and Shatranj. At this time chess was played in India and Persia, where it became famous among Persian Royalty. The names of the pieces were altered greatly: Shah (King), Farzin (Queen), Asp (Knight), Pil (Elephant/Bishop), Rukh (Boat/Chariot/Rook) and Piyadah (Pawn).

The Arabic Period of Chess adopted the name of Shatranj and chess was now played in various Middle Eastern countries. The names of the chess pieces changed once again: Shah (King), Firz (Queen), Fil (Elephant/Bishop), Baidaq (Pawn) the, but Rukh (Boat/Chariot/Rook) stayed the same.

The Medieval Period of Chess introduced the game to Europe for the first time in the early 13th century. The names of the pieces saw adaptations which included English words, which still apply today. These include the King, Pawn, Knight and Rook (at this stage the Rook was not yet in the form of a castle; instead it symbolised both a chariot and a bird). The Ferz (Queen) and Alifil (Bishop) still retained their Middle Eastern names.

The Modern Era of Chess introduced set rules and structure to the game. These rules still apply today. Previously, the game of chess required players to capture all their opponents' pieces on the board before a game was considered over. This was changed to the King's entrapment (Checkmate) as being the only way to complete a game of chess, regardless of the number in captured pieces. The Queen Pawn, the concept of castling, the increased movement of the Bishop, the increased movement of the Queen, the option of skipping a square on a Pawn's first move as well as *en passant* were all established during this period (Davidson 2012 pp. 19-36).

The development of chess was relevant for this study as I needed to explore ways in which to implement chess moves into music variables. The movements which chess pieces make, and have made in the past, aided new possibilities and methods by which to approach the representation of musical intervals via the use of chess moves.

Chess pieces have developed rapidly throughout history and as a result, I realised that, if sounds were implemented within the movements of the chess pieces, I would have to define each piece in accordance with its assigned sound. I calculated that I would not be able to give logical justifications for making a chess piece sound a certain way, for example why should a Knight sound as a "G"? As a result I made a supposition that it would be better to represent musical pitches within the chess board, rather than implementing them into the chess pieces. Creating musical pitches within the chess board would make it more accessible to play different versions of chess, such as “Doubles” (this requires four players, two chess boards, and all the captured pieces which are given to the opposition’s team mate. These captured pieces may be placed anywhere on the board at any point in the game). It is also not likely that the chess board will change in the future. As a result, this section guided me to further explore the variables within the chess board.

2.2 Western music notation

In this section a brief development of Standard Western Music Notation is represented in four different stages: Gregorian chant, Tablature, current Standard Western Music Notation and lastly, computerised music. Although Standard Western Music Notation has an intricate development process, these four groups aid a clear view of the changes that notation has undergone through time. Unsystematically based 21st century composition is not discussed as electronically based music challenges an absolute work of art, thus it does not form part of my experiment. When analysing the four stages more closely, it seems that even with modern technology, music notation has resisted alterations from its original notational systems and that through our history music notation has shifted according to social needs. Very few attempts have been made in the modern world to find an entirely new representation for Standard Western Music Notation based on an existing logical system, for example the Table of Elements. After analysing these four core notational stages, it seemed that music notation has remained close to its point of inception.

2.2.1 Gregorian chant

Originally music was learnt aurally from the composer but, to ensure the preservation of music, a notational system was in demand. Gregorian chant was an important development as it was the first step in defining a mental representational system or reference which gave practical music its first relation to a visual dimension. Over 300 forms of a cappella singing developed in the Roman Catholic churches across Europe, Africa and the Middle East. During the eighth century, Gregorian chant became the most popular form of chant in Europe. It is debatable whether the bulk of the melodic constructions in Gregorian chant originated from France or from Rome (Taruskin 2010 pp. 89-93). Gregorian chant is named after the sixth and seventh century Pope St. Gregory I (the Great). It is said that Pope George the Great stated that Gregorian chant was inspired by the Holy Spirit (Taruskin 2010 pp. 34-35).

Books consisting of notated Gregorian chant only appeared in the tenth century. Prior to the tenth century it was a luxury to notate music, as paper and quills were expensive supplies and very few people outside of the church were educated, thus they could not notate music (Taruskin 2010 p. 115). Chant notation was represented in neumes, symbols that indicated the movement of notes going higher or lower. These were indicated with signs such as the "virga", which indicated moving higher in pitch with the voice and the "porrectus" (N), which indicated singing higher, then lower and then higher again. There were many more neumes that were initially used for choral music, also known as Gregorian chant or Plainchant. The neumes were originally placed above the text. Neumes did not indicate exact pitch (Taruskin 2010 pp. 293-300). An anonymous document called the *Alia musica* depicted the first staff line in red ink and named it F, which is still the case in our bass clef today. Soon thereafter, a yellow line was added above the red line, named middle C (Taruskin 2010 pp. 117-119).

In the tenth century the letters of the tetrachord, D; E; F; G were recorded and named the Dasia notation. Neumes did not indicate exact pitch until the 11th century and oral tradition was required (Taruskin 2010 pp. 116-117). Thereafter a six lined staff indicating the spaces between the lines was created by Hucbald and rather than neumes he used text syllables in the spaces. He also indicated whole tones and semitones for the first time, which he marked with a "T" (Taruskin 2010 pp. 120-124). Guido of Arezzo was the first person to create the four lined staff with neumes written on top of and on the lines. The four lined staff is still recognised today as the common Plainchant Staff. The grand staff at a point consisted of eleven lines that linked the treble and the bass clef as one entity. The five lined staff was created in the 12th century, but was not recognised until the 17th century. Guido created the tonic Sol-fa. Originally his Sol-fa system was written as: Ut, Re, Mi, Fa, Sol, La, Sa. He also created the Hexachords which were six tone scales that required a B-flat. The Hexachord without the B-flat was called Hexachordum Naturale, which indicated that the scale was natural; this was the inception of accidentals. Neumes altered after Guido's time, into square shapes rather than movement symbols. with the "virga" becoming a square with a tail (Taruskin 2010 pp. 148-155).

In the beginning of the 12th century, music notation was the first notation that could be interpreted without oral instruction as it represented exact pitches, even though there were no recorded rhythmic measures. By the late 12th century there was a demand for a new realm of music notation named "ars mensurabilis". The entomology of this word is measured time, or rhythm as it is called today (Taruskin 2010 pp. 293-296). Music notation that included rhythm appeared in the 13th century. Although it did provide single note rhythms, the notation was mainly based on the ligatures. Rhythm developed steadily throughout the 13th century and by the 14th century, while signs such as the brevis, semibrevis, longa, duplex longa and maxima were commonly used to represent rhythm in music notation (Taruskin 2010 pp. 251-252, Metzinger 2015).

By the 15th century more signs were created and set values were assigned to tones; by the end of the 15th century, Josquin des Prés created the first sharp signs that were originally indicated as a crossed out "B" (Taruskin 2010 pp. 649-651).

This section has highlighted the fact that the journey of music representation was a slow but steady one. What I found most interesting was which elements of practical music were historically thought to have been the most important to represent via notation. Pitches were focussed on for many centuries prior to the rhythms being explored and implemented in music notation.

I interpreted this with regards to my study as the way in which I was going to create variables that represented chess moves into sound. I surmised that if I started creating variables with the same steady progressions as in the Gregorian chant era, other developments such as rhythm could be designed and implemented, in a similar logical succession, at a later stage. As a result I established that it was best to first focus on creating the variables which would represent chess moves translated into pitch.

2.2.2 Tablature

Tablature notation is an example of a different way in which music (playing/sounding) was represented at the beginning of the development of music notation (writing down). This section continues to discuss the variety of ways in which new musical concepts required written representations and how these musical concepts (sounds) were implemented within the developing musical composition of this time.

Up until the 17th century the tablature, which was a vocal notation, was also used for melodic instruments such as the viol and oboe. Chordal instruments such as the organ used a different version of tablature, where other signs were used in place of the common tablature notation. According to Latham, tablature was used mainly by musicians who could not read music notation. It is a complex notational system that uses lines and symbols that are not on a staff, but it included bar lines. In the 17th century these bar lines were accepted into Standard Western Music Notation (Latham 2003 pp. 1253-1254, Taruskin 2009 pp. 606-626).

By the 17th and 18th centuries the shapes of notes were altered to round shapes, as they were easier to print than the then existing squares. Although expression markings were used in the time of early neume notation, they were lost, but made their reappearance in the 17th century. The 18th century Mozart indicated the first non-legato instruction with dots. In the 18th century ornamentations were written out as it should be played in music scores. Ornamentations were indicated with only signs in the Baroque Era. According to Thompson, the 19th and 20th centuries showed few notational changes, and almost came to a standstill since before the ninth century (Thompson 1985 pp. 1519-1521).

Applying this to my study meant that if I first started to create variables for only one entity of representation for chess moves they were more likely to advance to their optimum potential in the future, as new representations could be found according to users' needs at a later stage. Assigning all the variables at one time could be disastrous if they were not all implemented correctly, which in this case could ultimately result in misrepresenting music or chess entities. The Gregorian chant and Tablature sections aided me with an overall method. As a result, the first variables which I focused on were the ways in which to represent pitch within the chess board.

2.2.3 Standard Western Music Notation

Standard Western Music Notation tries to depict auditory phenomena in spatial terms. In the 19th century expression became an important aspect to notate. This resulted in an influx in self expression within musical compositions that demanded newer notational symbols. This introduced an increase in symbols which had to depict more complex rhythms, dynamic markings and a greater variety of pitch. Rhythmic values increased and decreased with the inclusion of irregular time signatures. Some composers created works which were overly fast and other times too slow resulting in their music becoming incoherent. Dynamic markings of the Classical Era, such as fortissimo and pianissimo, were intensified in the Romantic Era. This resulted in pianississimo and fortississimo and sometimes up to four forte and four piano markings being indicated. There was also an increase in the usage of crescendos and decrescendos within musical works during this time. The pitches altered to express twelve tones as well other frequencies that are not included in the chromatic scale (Taruskin 2009 pp. 343-411).

In the Romantic Era broader harmonic structures developed and as a variety of complex harmonies came into existence, music notation required modern performance directions such as accelerando, ritardando, fermatas, as well as other expression markings which both sustained and accelerated musical tempi. Most importantly *Tempo Rubato* was created. This is a tempo marking that translates into English as stolen time. In order to depict a variety of different expressions within the music, timbres become increasingly important and included modernised sound qualities. Orchestras doubled in size as more tonal colours were in demand for compositions which consisted of bigger chordal structures. This orchestral development resulted in high and low ranging instruments being utilised in order to perform a greater range of pitches. Standard Western Music Notation expanded in length as composers created works which were longer in duration. This resulted in these pieces becoming cohesive via recurring melodic themes throughout various movements. Music that was composed in the 19th and early 20th centuries included Nationalist music, such as lullabies, folk music and dance music that depicted a vast array of cultural identities. Although many 19th century composers created profound musical works, some chose to compose music with characteristics of the Classical Period. Currently, Standard Western Music Notation is understood and used by the Western World as well as non Western countries (Taruskin 2009 pp. 443-479).

2.2.4 Computerised music and its challenges

The 20th century introduced the computer. This section discusses how the advancements of computer technology have impacted the development of music, especially with regards to seeking new representations of Western Classical Music and future notational system developments. As Iannis Xenakis was one of the frontrunners in computerised music technology his work is also specifically explored.

Huge technological advances have come about in the 20th and 21stcenturies. Music composition programmes such as GarageBand for Mac, Sibelius, Common Music, Ludwig and Hummingbird are all at the disposal of modern composers. These technologies aid composers with a faster turnover time. Sibelius offers more sequencing options than computerised representation of music. Standard Western Music Notation has still shown few advances. This has resulted in some attempts that have been made to alter current Standard Western Music Notation via the use of computer programming. One software programme that attempted this alteration is New Music Notation. However, their results simply replaced the shapes of accidentals with diamond shapes. Other changes of music representations via programming include avant-garde artistic drawings that depict various emotions (Sadie 1980 pp. 415-421, West 2013, Nelson 2014 pp. 219-221).

Some of the most successful attempts at linking music with computer technology would be those of Iannis Xenakis, who was renowned for using his advanced mathematics and music skills, and linking them with computer science. He created new compositions via the use of machines and computers and aided the world with interesting music notations. His initiative in seeking modern music representations is what is lacking in the 21st century's Standard Western Music Notation. Xenakis best linked mathematics and music by using UPIC (Unité Polyagogique Informatique CEMAMu), which is a computer composition programme that he created. The problem lies in the fact that most new music notational systems aim to substitute the already existing Standard Western Music Notation with new symbolic information and representations, rather than seeking an entirely new system, as Xenakis did. One could conclude that Western Classical Music demands a notation system that can both cater for pre-existing systems as well as those of modern demands (Xenakis 1992 pp. 329-389, Harley 2004 preface, Luque 2009 pp. 77-84).

This research made me aware of the advantages and disadvantages of creating an entirely new music notation or of seeking new representations for music practice. For example if someone were to play from a score containing chess notation it could aid an entirely new notational system, which could require less reading and be quick to notate. Alternatively, if the representation of the pitch variables was too complex and was implemented incorrectly, the data could be misrepresented.

I am interested in future developments with regards to a sounding chess game. This implies keeping the pitch variables as basic as possible, as the chess notation needs to be easier to understand and read if I want a long term goal of introducing chess music notation to practical music and music composition in the future.

2.2.5 Changing Standard Western Music Notation

In order to better understand what Western Classical Music representation harnessed and what it lacked, I investigated the restrictions of Standard Western Music Notation. I attempted to grasp what changes would need to take place with regards to the effect that Standard Western Music Notation has on the representation of music today, and what that means for the aural practice in Western Classical Music.

This section explored three predominant Standard Western Music Notation restrictions, as follows: Firstly, that the way in which composers wanted their music interpreted has become lost as a result of under-notation. Secondly, that the aural abilities of musicians are deteriorating as a result of being overly dependent on Standard Western Music Notation. Lastly, that Standard Western Music Notation is too outdated to represent the musical advances of the 21st century and those of the future.

If the Standard Western Music Notation system were to change, it would probably be altered via the use of modern technology, yet some musicians are content with the current system. Patterson believes that the future of Standard Western Music Notation is secure (Patterson 2015). Others believe that we are not developing our Standard Western Music Notation systems as consistently as was done during previous centuries (Nelson 2014 pp. 222-232). It is difficult to believe that Western Classical Music Notation from hundreds of years ago is meeting the needs of a 21st century Western Classical Musician, but this is nevertheless what happens, for within the constraints of the current system musicians still find new ways to express themselves.

Johnson explains that musicians who listen to hours of music are more musical than musicians who do not listen to music frequently (Johnson 2002 pp. 11-37). Taube believes that a recording of music does not contain any performance information (Taube 2013 pp. 3-4). Perhaps this is not visually the case, but according to Klickstein, it certainly does aurally. No study has ever proven that a musician who sight-reads for hours on end is more musical: in fact Klickstein explains that listening to music frequently will improve one's sight-reading. The only way to understand stylistic elements in music is by listening to music (Klickstein 2009 pp. 14-25; 98-102). The lingering question remains: why make something aural visual, when there are performance directions in recordings which the neurological pathways in the brain detect and the inner ear can retain as a memory?

In my Honours thesis I argued that dyslexic music students need to hear their music before being able to decipher its notation and that only after hearing their music does the notation make sense to them. There is a constant debate between the necessity of music notation and its downfalls of music notation. The speed with which music needs to be produced in the 21st century typically does not enable the musician of today to learn music by ear, as according to Kaschub and Smith, the average musician nowadays does not have the aural ability of musicians of previous eras. Musicians today depend on music notation with more intensity and with the intent of learning their music, rather than to catalogue or teach (Kaschub and Smith 2014 pp. 29-30). Although Standard Western Music Notation may be frowned upon by practical music purists, it definitely has its place in this era. The irony of this is that there is a huge emphasis on sight-reading in Western Culture, yet there is an even greater force of musicians protesting against the changing of the old fashioned music notation. They insist that this cannot be what our predecessors could have foreseen (Roads 1996 pp. 708-724).

The Standard Western Music Notation system poses many restrictions and challenges. Limitations arise when interpreting ornamentation or other stylistic indications of notational systems from previous centuries as they cannot be accurately notated or delineated in Standard Western Music Notation. As a result, other field work is required by the performer in order to understand what the composer intended. In modern non-tonal compositions most of the notes require accidentals which make reading unnecessarily complex. Standard Western Music Notation always fulfils the function of transferring music from composer to performer and from performer to listener. Throughout history, music notation has not been uniform, as some compositions are notated well and others are not (Chung, Hinde and Moonis 2003 pp. 93-99).

According to Scholes, there are musical elements which notation will never be able to represent: “There are nuances which paper and ink afford no means of representing and without which the music does not ‘live’; and ‘tempo rubato’ entirely eludes the art of writing” (Scholes 1955 p. 696).

There have been attempts made to find a new notational system, but all of them take the existing Standard Western Music Notation and alter it with new symbolisms and designs that represent the information in a variety of interesting ways. Why not just use or create another system all together? Western Classical Music is a logical system that can be represented by any non musical logical system.

When one takes on the responsibility to represent something, one can either represent it well or misrepresent it. This illustrates that it is preferable to create one variable well, such as sound, rather than 50 variables, badly. This section helped me to comprehend what Standard Western Music Notation was lacking. As a result, I was aware of these missing elements which, in turn, aided me with ways in which to seek solutions with regards to creating the representations of chess moves made to sound.

2.3 Chess notation

The development of chess notation is represented in five different stages: Firstly, a brief history of chess notation, secondly, Philipp Stamma, thirdly, François Andrè Philidor, fourthly, Standard Chess Notation, and lastly computerised chess notation.

Chess notation has had an extensive development, but the following five sections offer a description with regards to the development chess notation has undergone through the ages. After summarising these five stages, it was clear that the development of chess representation had not undergone as many changes as Standard Western Music Notation; in fact the way in which the game's notation is currently applied has remained similar since the 17th century. As in Western Classical Music, chess has its own notational system. Algebraic Chess Notation is the transcription of each chess move made by both black and white pieces throughout the game. It can also describe a position at any point of the game.

2.3.1 A brief history of chess notation

This brief history shows two versions of chess notation, which ultimately describe the same entity, that of the order and representation of chess moves in a game. These two notations were publically received conflictingly at different stages of their developments. One could say that it took a chess game to decide the fate of the currently used chess notation.

Although the Algebraic notations have existed alongside the Descriptive Chess notation since before the 17th century, two men from opposite sides of the world competed with their short hand systems of both notations. Philipp Stamma published his book *The Noble Game of Chess* in 1737; it depicted a short hand Algebraic Chess Notation system which he developed. François Andrè Philidor published his short hand Descriptive Notation book in 1747. They were not only rivals with regards to chess notation; they also competed in practical chess. Stamma lost to Philidor in a tournament in 1747, the same year as Philidor's book was published. This resulted in Philidor selling many of his books and becoming an overnight sensation. It is believed that had it not been for Stamma losing the match that the Descriptive Chess Notation would not have survived until the 1970s (McCrary 2015). Although Stamma lost the chess game, his notation stood the test of time as it is the official chess notation used today. It is currently referred to as Standard Algebraic Notation (SAN). Ironically there is no known notation of Stamma and Philidor's game against one another (Golombek 1976 pp. 240-241).

This is a prime example of how one form of representation can be accepted for various reasons at one point in time, and another form of representation can later be accepted for alternate reasons. For this study, I needed to take into account long term reasons for the way in which I implemented variables. As an illustration of this, it might be surmised that a popular outcome for my study, in present times, could become redundant in future if the variables only pertain to an aspect such as being aesthetically pleasing.

2.3.2 Philipp Stamma

Philipp Stamma from Syria published his short hand system that was based on Algebraic Chess Notation in France in 1737 (Stamma 1737 pp. xxi-xxiv). It was translated and published in English in 1745 (Stamma 1745 pp. xxi-xxiv). Algebraic notation was used in the Middle East prior to it being discovered by the Western World through publications by Stamma. If it were not for Stamma, Middle Eastern and European chess notation would not have had one notation today. Stamma is famous not only for his notations, but for his analysis of the end game. Originally his system described which pieces were moving by naming them after their starting file position (Hf3) instead of the object's name (Nf3) as is done today. Although his file position was replaced, it is still used when two of the same pieces of the same colour, such as two black Knights, can move to the same square. The file name is then added to identify which Knight has moved (Nh5-f6). Today this Algebraic short hand notation is the recognised notation of the World Chess Association, FIDE (Federation Internationale des Echecs), and competitive chess notation (Williams 2001 pp. 36-37, Schafroth 2002 pp. 100-103).

2.3.3 François Andrè Philidor

François Andrè Philidor was both a music and a chess prodigy. He performed his first music composition in front of King Louis XV and composed 17 operas. Philidor was also a renowned blindfold chess player. In the early 17th century the Western World used Descriptive Chess Notation. Descriptive notation used full sentences to describe the movement of a piece. It took immensely long to write down and today very little is understood from archived, descriptive notations. Here is an example of Descriptive Chess Notation: "The white King commands his owne Knight into the third house before his owne Bishop". In 1777 he wrote a book in which he explained his chess notation system, which was based upon the originally used Descriptive Chess Notation which consisted of full sentences to describe every move that was made. In the preface Philidor explained that the Chinese game of chess is not the same as his nation's game and he also believed that he was the first man from the Western World to cultivate the practical and theoretical study of chess (Philidor 1777 preface para.1). Today this is still referred to as Descriptive Chess Notation, which was normally written in

English and Spanish. People who were illiterate could not understand this notational form and with its other disadvantage being time consumption, it became increasingly unpopular. Different versions were used in English speaking countries up until the 1970s, after which it was no longer permitted as a chess notation in FIDE rated chess tournaments and was then replaced with the Standard Algebraic System (Golombek 1976 pp. 117-119, Hearst and Knott 2009 pp. 21-25, Pritchard 2012 pp. 35).

In general this can be regarded as an example of a scenario where too many representations can become overbearing and over-notated, which could result in frustration for the user interacting with the content. To reflect on the notations of Stamma and Philidor helped me analyse the way in which variables that represent the same entity could be implemented in various ways, even though they still achieved the same outcome.

2.3.4 Standard Algebraic Notation

Standard Algebraic Notation (SAN) is the notation that is now used globally to represent chess. This notation is used later in this study, as all of the chess moves which were made to sound for this study are notated in SAN (Pritchard 2012 p. 36).

SAN is the written notation that is also referred to as international notation. Both players are required to notate throughout the game, unless there is less than five minutes left on their clock. Each move is written down in chess notation pads, which consist of two lines for each move, one for white and one for black. Whilst notating, the chess player is expected to follow FIDE rules which state that a player may only notate a move after moving the piece properly and removing his or her hand from the chess piece. The move is then notated and the player is then required to press their clock with the same hand with which they have touched their piece. Only then is the move complete and the opponent may answer with their next move. The chess board is numbered along the sides with numbers 1-8 and is horizontally marked with letters A-H. A1, the black square in the corner, faces the player who is playing white on their left hand side and A8 is the white square on the right hand side of the player who is playing black. The player has to notate the square that the piece has moved to as well as the first letter of the piece's name. When capturing a piece the "x" sign is used. Castling requires "0-0" for Kingside castling and "0-0-0" for Queenside castling. Check is notated with a "+" and checkmate with double "++". When a game is complete, a "1" is placed under the moves of the winning colour and a "0" is placed under the moves of the losing colour. Both players are required to sign one another's notation sheets and a result card must also be filled in by both players before the result is entered into the FIDE database (FIDE 2015).

After exploring chess notation the next step was to analyse what was not portrayed by Standard Algebraic Notation. I compared what is being depicted in chess to what is being represented in Western Classical Music and discovered that Standard Algebraic Notation lacks three variables which are important for the accurate rendition of Western Classical Music: firstly, Standard Algebraic Notation does not describe where a piece was prior to being moved, as it takes into account that visually the player can delineate the origin of a piece. Secondly, the Standard Algebraic Notation does not aid one with a way in which to describe emotions or external entities which could affect the decision making process in a chess game. Lastly, Standard Algebraic Notation does not require representations for the time it takes for chess players to make each move nor does it represent the length of a game in general.

This meant that I would have to re-define the representation of Standard Algebraic Notation if I wanted it to represent musical principles, as Standard Western Music Notation does aid representations for rhythm and intervals, which Standard Algebraic Notation does not. I also found that both chess and Western Classical Music lacked representations for entities such as interpretation.

2.3.5 Computerised chess notation

Due to the advancements in computer technology, online chess content required new technological representations of Standard Algebraic Chess Notation. These representations were quickly developed using a method from 1846, Figurine Algebraic Notation.

Figurine Algebraic Notation (FAN) is the notation used for computerised chess. It is based on the system that was created by Philipp Stamma (SAN). An icon, representing the piece, substitutes the first letter of the chess piece's name. For example instead of (Nf6) it becomes (♞f6). This makes for a more inclusive chess notation, as it can be understood without language barriers (Golladay 2007 pp. 115-156). In 1846 Count Robiano was the first person to write about and use Figurine Algebraic Notation which was originally referred to as Notation Parlante (Hooper and Whyld 1996 p. 134). Currently FAN is used in online chess tournaments, online chess databases, newspapers and published chess books, but it is not permitted by FIDE in non-online chess tournaments (FIDE 2015).

This gives the chess notation, which I used to represent new music variables, online representations that are already in place. Furthermore, this means that there are already existing advancements if sounding chess notation were to materialise. Representations are adapted in accordance with the users' needs for that time, much like the notational developments of Gregorian Chant and Tablature. I also came to the conclusion that when comparing Western Classical Music composition and music notational software with computerised chess notations, not much has been altered or developed with regards to new ways in which to represent both the current Standard Western Music Notation and the current Standard Algebraic Chess Notation. As a result of this computer notational link between chess and Western Classical Music, I also wanted to explore the linkage between computer chess software and computerised music software, as I believed that programming could produce an algorithm in which to implement the variables that I have created.

2.4 Computer chess

Computer chess has been used to test and determine technological developments since its inception. In this section I wanted to evaluate how computer chess has developed in order to delineate whether it could be integrated with a computer programme which produced sound. Coding a computer programme could serve as an algorithm in which to substitute the variables that I have to implement in my thesis. These variables include: the chess board and the chess pieces in correlation with musical pitch; the order of chess moves represented via a melodic line; musical rhythm translating a chess player's decision making process with regards to time usage. Programming plays a vital role in both the future developments of chess and music. In this section it became apparent to me that using computer programming to link chess and music aided a modern mathematical foundation between the two entities. To recognise the complexity of this process the development of the chess engine is further explored towards the end of this section.

In 1770 the Turk, invented by Wolfgang von Kempelen, was a machine that played chess against people. Von Kempelen was an inventor as well as a diplomat. The Turk consisted of a life sized wooden cabinet with arms that resembled those of a human. Von Kempelen first presented it to Marie Theresa, the Empress of Austria, as a magic trick. The machine became an overnight sensation as it very seldom lost any of its games. In 1807 the Turk was sold by Von Kempelen's son to Johann Maelzel. The Turk travelled throughout Europe and the United States of America and played against famous people such as Catherine the Great, Benjamin Franklin, Charles Babbage and it even resigned while playing against Napoleon, as he was fooling around and cheating against the device, in order to test its abilities. Some people believed the Turk to be witchcraft and magic whilst others believed it to be a hoax. Edgar Allan Poe wrote an essay in 1836 depicting his suspicious theories behind the mechanics of the Turk automaton (Computer History Museum 2011a).

The Turk even fooled the associates of the Academy of Science in France, who could not figure out how the contraption worked or whether it truly was a hoax. The secret to the Turk, that fooled Europe and the United States of America for 85 years, was that participants were actually playing against an anonymous chess grandmaster who was sitting inside the cabinet. He gained access through a trap door and controlled the arms of the Turk from the inside with magnets, strings and ropes. Eventually the anonymous grandmaster passed away in Cuba and the owner of the Turk, Maelzel, also died on his return from Cuba in the late 1830s. The Turk was placed in a museum in Philadelphia that burnt down in 1854. The Turk did not survive the fire (Levitt 2006 pp. 1-258).

Alan Turing theorised the world's first computerised chess programme in 1947. Although computers did not yet exist, Turing used chess to demonstrate the abilities that computers could potentially harness. In 1952 Turing tested his hypothetical computer programme's prototype, "Turbo Champ", by playing a match with his friend Alick Glennie. Glennie beat Turner, but after only 30 moves, which according to experts is quite remarkable. Glennie is still recognised as the first man in the world to beat a computer programme at chess. There are publishings of Turner's chess programme, but in spite of this, his programme was never completed (Computer History Museum 2011b, Peddie 2013 pp. 140-142).

During 1958 in the United States of America, another development was made with regards to the then hypothetical computer chess system. Claude Shannon is well known for writing the world's first article about computer chess which was published in a philosophy magazine in 1950 (Shannon 1950 pp. 1-18). Whilst he worked for Bell Telephone Laboratories, Shannon created two theories that are still in use today. His first theory, "Type A", considered the computation of variations for different moves and his second theory, "Type B", included a strategy for the computer to outsource the most valuable moves to the position at hand, using existing chess theory (Newborn 2014 pp. 8-18).

The first fully working chess computer programme was created by the British doctor Dietrich Prinz in 1951. The chess programme had severe limitations compared to what we have today. It could detect the superior moves for all two move checkmates but the programme could not manage a full game. Just seven years later, in 1958, the infamous computer programmer, Alex Bernstein, became the first man to develop a fully working computer chess programme. It was tested against International Chess Master (IM) Edward Lasker and even though he beat the chess programme, it was a huge triumph for Bernstein. Lasker admitted that the chess programme had the ability to beat amateur chess players (Levy and Newborn 2012 pp. 54-59, Newborn 2014 pp. 22-24).

From 1958–1967 the Massachusetts Institute of Technology (MIT) was the frontrunner in developing a computer programme that could beat mediocre chess players. Four first-year students decided to develop the programme and by the time they graduated in 1962, it worked. Richard Greenblatt was also a computer programmer at MIT, was intrigued by the four students' work and as a result he developed their system further. In 1967 Greenblatt created the first chess programme that competed in a chess tournament. The programme was named "Mack Hack VI" and it had a rating of 1400 (Computer History Museum 2011b, Lasar 2011, Levy and Newborn 2012 p. 8).

The 1970s brought faster machines and computer chess grew more popular. Even private hobbyists and homebound engineers tried to create their own chess programmes, but computer chess was still not available to the general public. Much like the piano, which was initially exclusively owned by the rich, playing chess on a personally owned computer was a luxury that only the wealthy could afford. Microcomputers were the world's first privately owned computers and were developed by Peter Jennings in the mid 1970s. The three most popular 1970s chess computer brands included CHESS, Belle and KAISSA and all three of their chess programmes ran off software that was microprocessor based (Computer History Museum 2011c, Computer History Museum 2011d).

The 1980s marked the first World Microcomputer Chess Championships where a diverse group of programmers created their versions of chess software and hardware that could compete against the world's best players. This made chess increasingly popular among big investors and top programmers. It was becoming more frequent for computer programmes to beat human players as well as other chess programmes. Fidelity produced their first Chess challenger which was created by Sidney Samole and Ron Nelson. It was a popular, compact, electronic device that average chess players could play against, in the privacy of their homes. IBM (International Business Machines) became interested in developing a chess programme that could outsmart an international grandmaster. In the late 1980s an IBM team, Thomas Anantharaman, Murray Campbell and Feng-Hsiung Hsu, created a programme called "Deep Thought" that played three times against International Grandmaster Garry Kasparov. In 1989 Kasparov beat "Deep Thought" (Computer History Museum 2011e, Newborn 2011 pp. 1-6, Marshland and Schaeffer 2012 pp. 55-79).

In 1996 Kasparov was challenged again by the IBM creation, when he lost two of the six games that he played against the chess programme, but ultimately did not lose the match. The last and most famous "man versus computer" challenge left Kasparov the loser against the then renamed "Deep Blue". This made him the first Grandmaster in history to lose a match to a computer chess programme. After the triumph of "Deep Blue", it became more frequent for computer programmes such as the Fritz group, Hydra and Deep Junior to beat international grandmasters (Hsu 2002 pp. 3-261, Kasparov 2010, Hale 2013 pp. 13-25).

Today chess computers are referred to as chess engines and the most popular engines that are competing nowadays include Komodo 9, Fritz 14, Gull 2.8b, Fire 4, Houdini 4, and Stockfish 6. Now that computers have beaten humans at chess, the 21st century brings new adventures for the chess engine. Currently chess engine engineers and programmers are focussed on competing against one another. They also have high ambitions for creating a new machine that can teach and tutor chess to humans in the place of human chess coaches (Jaap van den Herik 2008 p. 192, Hartmann 2014).

Currently the 2015 winner of the 21st World Computer Chess Championships is Johannes Zwanzger from Germany, whose programme “Jonny” beat all of the other chess engines competing in the Netherlands (Krabbenbos 2015).

The development of chess engines excited me because I found similar representations in computer music developments as there could be in computer chess developments. This meant that chess and music could be linked via computer programming which could serve as the final algorithm into which chess variables could be substituted in order to create sounds, as programming is a modern form of mathematics.

After exploring the developments of both computer chess engines and computerised music, I decided to select computer programming languages that would serve as the mathematical linkage between chess and music. This provided the algorithm into which the variables that I created, to represent chess moves into sound, were substituted. I also wanted to explore if there were existing methods in which chess and music have been linked via computer programming in the past.

2.5 Matthias Wüllenweber

After deciding that I was going to code the created variables into a computer programme, I explored ways in which chess and music have been linked via computer programming. This section discusses the works of software developer Matthias Wüllenweber, who has created both chess engine software and music composition software.

Matthias Wüllenweber was born in 1961 and is a world renowned software developer and physicist. He is most popular for three different chess programmes: firstly for developing and launching the world's first chess database programme, ChessBase, in partnership with Frederic Friedel, a well known chess journalist. Secondly for developing Fritz, the computer chess engine that won the 1995 World Computer Chess Championships and lastly for Ludwig, the computer programme that combines chess algorithms with music composition software. He studied physics at three different universities, namely the University of Bonn (1983-1986), the University of Edinburgh (1984-1985) and the University of Bremen (1991-1993), where he also studied music theory. Wüllenweber is a very good pianist and can also play the flute. He has been the CEO of ChessBase GmbH since its inception in 1986. Wüllenweber currently resides in Hamburg, Germany, where the ChessBase headquarters are based (Kniep 2006, Friedel 2012).

2.6.1 Fritz

In order to comprehend the ways in which I could integrate the computer programming of both chess and music, I decided to explore chess and music linkage via computer programming. I achieved this by further researching Fritz and Ludwig.

"Fritz" is one of the most popular computer chess engines of our time. It was originally created and published by Wüllenweber, in 1991. Mathias Feist was responsible for the programme development of the Fritz chess engine. Fritz is an incredibly fast working programme and it has the ability to automatically change its own game strength. This aspect allows Fritz to play against players of all strengths, to help improve their playing strength as well as to make games more fair and enjoyable for opponents. Currently Deep Fritz 14 is the newest version of the computer chess engine and was developed by Gyula Horváth. Fritz has its own virtual world. It has its own user interface and players can choose their own board. It also has a database where chess players can revise games and even receive coaching via video chat with their coach (Wüllenweber, Friedel and Feist 2006 pp. 208-213, USCF 2007 pp. 149-183, Wall 2014).

2.6.2 Ludwig

Wüllenweber created a music composition programme that works using the same algorithms as Fritz and named this music composition programme Ludwig, after the composer Ludwig van Beethoven. Ludwig is a programme that synthesises music and chess utilising search techniques based upon Artificial Intelligence (AI). Ludwig composes by itself, but it can also be used by composers to programme their own compositions. Ludwig was first incorporated within Fritz 9, where chess players each receive their own customised theme song based on the username that they enter. The programme also has the ability to teach practical music and provides entertainment for beginner and advanced musicians. Ludwig composes pieces of music based on information that the participant provides; this includes the level of the participants' playing, their preferred genre and lastly the instrument that they wish to play. The computer programme considers all variables and then composes an appropriate piece for the user to play. An accompaniment ranging from a full orchestra to a rock band can also be added, once the user has learnt the music. Wüllenweber presented Ludwig in Bonn, which was the birthplace of Ludwig Van Beethoven, at the Man versus Machine conference in 2006. The latest version of the programme is Ludwig 3.0 (Friedel 2006, ChessBase 2011, Fischer 2015).

Both Fritz and Ludwig programmes consist of various features that aid more representations of, and interactions with, music and chess entities. I found two concepts in this section that caught my attention with regards to developing a computer programme for the end goal of my study: firstly, Wüllenweber's explanation of how he used highly advanced programmes to create Fritz and Ludwig and secondly, the method which was implemented for the creation of Ludwig via the algorithms used to create Fritz. This made me realise that I had to explore programming languages in order to delineate which programming language would best suit my study, whilst taking into account possible future developments such as adding features or assigning new variables. As a result I considered three programming languages as candidates in which to create the computer programme that would represent chess moves into sounding variables. These computer programmes are discussed in Chapter Three.

2.6 Sounding chess

This section investigates the way in which chess has been linked to sound, in three different instances. Firstly, through a film called Nummer Twaalf, and secondly, via a concert called Reunion by John Cage. Lastly, Slonimsky's thematic parallels between chess and Western Classical Music are discussed. It was important for my thesis to explore in which way the methodologies of these projects were different from mine.

In 2009 Nummer Twaalf film was released by Guido van der Werve. The film was about the unfolding relationship between chess and music. Guido was neither a chess player nor a musician, but in 2007 he needed to create a scene for a play, Nummer Elf, in which he had to compose his own music. He taught himself to compose via reading different materials. At the time he was also learning to play chess and soon he decided to write the scene about a chess grandmaster playing sounding chess. Guido taught himself about the piano and decided to merge the chess board with a piano keyboard. He summonsed chess grandmaster Leonid Yudasin to compose a chess game with the King's Gambit opening. Yudasin created a game called the King's Gambit Accepted. Guido then composed the piano score based on Yudasin's chess game and in the opening scene of Nummer Twaalf, as Yudasin played the scripted chess game against Guido, the piano would play the assigned notes. The first scene for his 2009 movie, Nummer Twaalf, was shot at the Marshall, a New York based chess club. This scene includes a string quartet accompanying the piano chess, Yudasin and Guido. Guido aimed to build a new instrument out of chess and music rather than making a game of chess sound (Luhring Augustine 2010).

In this case music was improvised, based on a deconstructed chess game; but for this study, the purpose was not to create music that was inspired by the characteristics of chess, but instead to represent prefixed variables which made all the moves in any chess game, sound. Regardless of whether this resulted in beautiful musical compositions or not, the aim was to use a logical method to create accurate representations of how chess would sound by implementing music variables. For example, instead of representing chess sounds via one instrument such as the piano, it was decided rather to represent a larger variety of pitch variables. As a result, the range for the pitch variables represented eight octaves, which were implemented according to the 8x8 chess board.

John Cage produced Reunion in 1968. He used a chess board with photo resisters to produce sounds while a chess game was being played. Four electronic compositions simultaneously played as chess pieces moved. A recreation of Reunion in 2012 used computer technology to modify the sounds of eight electronic musicians. These electronic sounds were accompanied by images of light beams on a screen and these images were influenced by the body movements of the people present.

Cage used the chess board to produce the sounds. These sounds were not in any logical sequential order as, if the same move was played twice in the game, it would produce a different sound. Therefore the chess move or chess piece had no consistent correlation to the sound, but rather the board produced electronic inputs consisting of random compositions every time a piece was moved. These sounds were then amplified through the concert venue with strategically placed amplifiers. The sounds became fewer as the pieces were taken off the chess board. There is no chess notational record of the actual games that were played during either of the Reunion concerts that took place in 1968 or in 2012.

In neither of the productions of Reunion was there an attempt to link the actual movements of a chess piece to a predetermined pitch, neither was there any attempt to notate the sounds or music produced during the chess games (Cage and Duchamp 2009 pp. 1-14).

Nicolas Slonimsky composed a chess prelude based on what he finds the characteristics sound like the chess opening called Giuoco Piano. He played this for chess Grandmaster Capablanca. Capablanca in turn recited the opening moves. The music was an impressionistic expression of his perception of the game and there was no logical or accurate correlation between the actual pitches in the music and the position of the pieces on the board. Slonimsky explored the thematic parallels between chess and music. He linked positional play to the characteristics of Classical Music, combinational play to Romantic Music and unconventional openings and gambits to Modern Music. These links are interpretative rather than algorithmically based. This is thus another example of composition that fits in the category of music about chess (Slonimsky 2005 pp. 318-322).

2.6.1 “Chess” the musical

Whilst researching sounding concepts of chess, the following section did not initially seem important to the study, but upon further exploration of the Chess musical’s storyline, I realised that there is an entire dimension missing in chess representation with regards to the energies and emotions which surround chess players. These entities directly affect a chess player’s decision making processes.

“Chess” is a famous Broadway musical set at a chess tournament that was taking place during the time of the Cold War. There are two versions of the musical, the original Broadway version and the London version. Both use the same music, but differ slightly in storyline. The original tells the story about a love-triangle involving two grandmaster chess players, an American grandmaster, Freddie Trumper, whose character is inspired by Bobby Fisher and a Russian grandmaster, Anatoly Sergievsky. The story unfolds as Freddie arrives at the chess tournament with his manager and love interest, Florence Vassy, who is introduced to Anatoly. The two grandmasters have an altercation during their chess match, which results in a defaulted game. Florence is left to sort out the embarrassing argument that was witnessed by the press. She later has an affair with Anatoly, who is a married man and the musical ends in a tragedy (Rosenfield 2010, Roberts 2010, Rigsbee 2015).

The music was written by the former ABBA singer Björn Ulvaeus and Benny Andersson. The lyrics were written by Tim Rice and the librettist was Richard Nelson. The music was recorded in 1983 in Stockholm, whilst the orchestral parts were being recorded separately in London by the London Symphony Orchestra. The composers decided to release the music to the public prior to creating the production and the music was an instant hit. The songs made it to the top of many worldwide hit music charts and the "Chess" album also reached the top 10 most popular albums in South Africa. Thereafter the production was launched and the original Broadway production premiered in 1988 at the Imperial Theatre in New York, and was directed by Trevor Nunn (Hartston, Rice and Andersson 1986 pp. 1-49, Scimia 2015).

When observing a round of chess at a chess tournament it is said that one is surrounded by silence, but one can hear chess clocks ticking, sounds of chess players pressing their clocks at different tempi, and the sounds of chess pieces being picked up, put down and captured, all at their calculated time. Everyone plays their games at different speeds. Some players are intensely nervous, others are overly confident; some are annoyed and so on. These varying emotions can affect a chess player's decision making process. For example if a chess player is nervous to play against their opponent, they might be more cautious when making chess moves and as a result it takes longer for them to decide where to place their next move. This means that the atmosphere in chess dictates the rhythms, but the atmosphere in Western Classical Music does not affect the rhythm.

After much consideration, I decided that this element of chess was important to represent and if it could not be represented in variables, I calculated that it could be implemented within the lack of certain variables. For example if I did not create a rhythm variable, but rather coded the sounds to play in a random succession, and instructed that the sounds be played *ad lib* for practical music performance or compositional reasons, this could represent the decision making process which pertains to the individuals' calculations at the time of engaging with the content, in accordance with their surroundings. If this were to be performed, it would also allow musicians to create their own rhythms, based on their interpretations of the given sounds and possible external factors.

The information researched in Chapter Two pertained to this study, as it helped to establish logical methods with which to approach the creation of the final variables. This also initiated the idea of presenting the final data via computer programming. Chapter Two highlighted the importance of being aware of the representations that could be portrayed within the final *The Sound of Chess Programme*, which was created for this study.

Chapter 3

Finding a conversion formula

This chapter includes an investigation of three programming languages, namely Lisp, Python and Java. These are the languages which were used to code the final computer programme. The advantages and disadvantages of the three computer programming languages were explored and discussed and as a result of this research, the final programme was coded in the Java programming language. The discoveries that were made at the end of the previous chapter were applied to the final variables and algorithm, which were then further researched. The final variables were established and a computer programme was created, into which the variables were substituted. The conclusion of this chapter analyses the three sounding chess games that were coded in to the final computer programme, which is named *The Sound of Chess Programme*.

3.1 Computer programming languages

In Chapter Two I explored methods with regards to a conversion formula which would reflect and present the final data for this study. As a result computer programming served as the conversion formula into which the final variables were substituted.

This section focuses on the development and system structure of the three programming languages namely, Lisp, Python and Java and their integrated development environments: Common Music, PyCharm and jGRASP. The reason for this investigation was to find the programming language which best satisfied the demands for the representations and future developments of this study.

3.1.1 Lisp Programming Language

Lisp was examined for this study because its commands are simple to grasp, and Lisp represents mathematical functions that solve all computable equations by manipulating the variables.

Lisp is a computer programming language that is, by one year, the second oldest high level computer language in the world. The creator of Lisp, John McCarthy, launched Lisp in 1958, just one year after the first computer language, FORTRAN, was launched. McCarthy was influenced by Lambda calculus notation, which was founded in the 1930s by Alonzo Church. Lambda expressions are explicated in an unspecified way, via mathematical computation that is then decomposed to input values in order to create output values. Lisp does not use anonymous Lambda expressions; instead, Lisp aids the ability to name a function in order to assist a user to identify the aim of the function in the equation. This may not produce a result, but it can be used for the creation of interesting implementations. Lisp was originally created as a practical mathematical notational system that was intended for the research of artificial intelligence. Although Lisp has altered with the passing of time, it has always been able to represent mathematical functions in a way that solves all computable equations. The most prominent Lisp dialects are Common Lisp, which is a substantial design, and Scheme, which is a modest Lisp adaptation (Cope 2004 pp. 151-234).

Lisp spearheaded Objective Scientific Computer Systems such as Automatic Storage Management, tree data structures, recursion and self-hosting compilers. A unique trait that Lisp harnesses is that it derives from its own Lisp processing, which essentially means that Lisp is written in and exists within Lisp itself. The software has the ability to manipulate variables for an idealistic outcome. The source code of Lisp consists of lists which are made up of units named Cons (Taube 2013 pp. 7-12).

All the values in Common Music are prefixed with a mathematical function, for example, a basic Common Lisp function is written (+ 2 2) or (print “Hello World”), which is the terminology that has been adopted by various programming languages such as C++ and PHP. Lisp contains an infinite number of representations for functions and operations such as addition, multiplication, subtraction, division, binding, variable, eval, fboundp, boundp, quote, defvar, symbol functions, print list and setf. Each of these fundamental functions is based on basic mathematical equating. Lisp gives a beginner programmer the tools to create complex algorithms without having to understand computer programming in great depth. The basic syntax is simple and can be adapted for use for today's computer systems (Simoni and Dannenberg 2013 pp. 1-27).

3.1.2 Common Music Software System

Common Music is used for algorithmic music composition and it is an object oriented music composition environment, which exists within the Lisp programming language.

Common Music is a programme created by Heinrich Taube in 1989 and is based upon the computer programme Lisp. As an undergraduate student, Taube saw the demand for a computerised compositional programme. He did not excel at mathematics in secondary school and he had no computer science experience, but as a music composition student at Stanford University he attended a course in artificial intelligence. There he learned to code in two languages, Lisp and Stanford Artificial Intelligence Language, which were both created by John McCarthy. As his studies progressed, much of the software that he used for music compositional purposes was discarded as computers developed into compacted, household items. Finding a substantial music composition software programme became impossible. Taube yearned to create a common music software system that was mobile, could be used on any hardware and that could be altered according to the user's needs. For his Masters degree he created Common Music (CM), a music compositional software system which changes algorithmic representations that are of a high level. Common music offers a variety of different compositional tools such as MIDI, Common Lisp Music, Music

Kit, C Music and Common Music Notation, one of which is called Stella, which is a music notational editor. The programme produces sounds by manipulating a portrayal of altered music metalevels and substitutes it into a mixture of arrangements. At the 1996 Ler Concours International de Logiciels Musicaux, Taube's Common Music software won first prize (Nierhaus 2009 pp. 63-64, Taube 2014).

Common Music exists within Common Lisp. Grace (Graphical real-time algorithmic composition environment) is the primary user application for Common Music. Grace consists of two languages that are used for musical algorithms. These are Sal and S7 Scheme. Sal and S7 Scheme are algorithmic music compositional programmes that are simple to grasp and to implement. S7 Scheme was created by Bill Schottstaedt who programmed Scheme into an updated programme to enable audio use. Sal and S7 Scheme consist of three main entities namely control statements, definitional forms and infix expressions. Within these entities, the programmes provide a variety of control statements such as set (appoints an array of presented variables, values); if (calculates which clause, "else" or "then", an affirmative equation consists of); loop (explains repetitive sequences) and begin (the final expression in a sequential format, which turns into the full begin value). Definitions consist of three different statements: these are variable (calculates all variables), function (re-specifies the statements' function) and process (calculates different settings in the function and loops them into one body) (Taube 2013 pp. 2-28, Schottstaedt 2015).

3.1.3 Python Programming Language

The reasons for exploring Python included the fact that it is a young programming language that is developing rapidly. Creating programmes in Python is time efficient because it has many built-in functions and uses a simple syntax (Python 2015).

Python was created by Guido van Rossum in early 1991. Compared to the other programming languages that I learnt for this study, Python is the youngest, and it is a relatively new programming language in general. Python got its name from Monty Python's Flying Circus, which was a popular British television show. Prior to its inception, Van Rossum worked for The National Research Institute for Mathematics and Computer Science (CWI). He worked with systems calls which required him to work with a computer programming language named ABC. ABC consisted of a platform programming language, which was an operating system named Amoeba, developed by Andrew Tanenbaum. ABC did not provide all of the functions that Van Rossum needed in order to complete his daily tasks at work, and as a result he decided in December 1981 to code a new language that would improve many of the flaws which ABC had. The new programming language required a module that could be added onto his code. Van Rossum decided to seek the help of programming developers at Modula-3, because he had worked with an earlier version of their module (Modula-2+).Modula offered an extension to his code, which provided error handling feedback (Bird, Klein and Loper 2009 pp. 1-34).

In 1995 Van Rossum moved to Virginia to start working for the Corporation for National Research Initiatives (CNRI), whilst continuing working on the development of Python and its distribution. In 2000 Van Rossum, along with his recruited programmers and developers, merged with a team of developers at BeOpen computer laboratories where they worked on further improving Python. The team made many improvements to Python based on popular features of various other programming languages. Haskell is one of the programming languages that was used to alter Python with regards to its List Comprehensions, which are expressions that run applications consisting of multiple variables. By 2001 Python Software Development (PSF) was formed for copyright reasons (Telles 2008 pp. 2-5, Lutz 2010 pp. 1-8).

Python harnesses many built-in functions. Python can be used in the C or C++ programming language. Users can change an object type at any point in time during a code. This makes it quicker to write the code, but slower to run on the computer. Python is available on various platforms. It uses very simple syntax and includes a variety of built-in functions. Python aids a faster turnover time for programmers to write programmes, as they do not need to declare data types. Python offers advanced types of objects; data type examples include integers (numbers), strings (text), dates and arrays (Martelli 2006 pp. 3-13, PSF 2015).

3.1.4 PyCharm Integrated Development Environment

PyCharm offers World Wide Web support which means that codes can be integrated with online advancements. PyCharm also includes features such as an intelligent code completion as well as a progressive debugging system.

PyCharm is an integrated development environment (IDE) which was created by a company named JetBrains. JetBrains is a software development company that was originally named IntelliJ and was founded by Sergey Dmitriev, Valentin Kipiatkov and Eugene Belyaev in Prague in the year 2000. Their first product was named IntelliJ Renamer. At the time that PyCharm was created, Sergey Dmitriev was the CEO of JetBrains. Dmitriev stepped down as CEO in 2012 to continue with research in the field of bioinformatics. The team of programmers who developed PyCharm are based in the Czech Republic and launched the first version of PyCharm in 2010. PyCharm was created as a development environment for the programming language Python and its web application framework named Django. PyCharm quickly became the preferred IDE software for programmers in Python and Django. The PyCharm software can be used to programme music within the Python computer programming language. Currently the JetBrains Software Development Company consists of over 500 employees and has two CEOs namely Oleg Stepanov and Maxim Shafirov. The JetBrains headquarters are in the Czech Republic and the company has expanded globally, with office branches in Russia, America and Bavaria (Jaiswal and Kumar 2015 pp. 33-41).

PyCharm was created to be an exclusive integrated development environment for the Python programming language, although it can also be used as an integrated development environment for other programming languages. PyCharm is cross platform and supports many web frameworks, not only Django, and it also supports many library features. PyCharm's test support can help users to test their codes within the PyCharm Integrated Development Environment and features syntax highlighting (a feature that categorises texts according to different fonts and colours) and code completion (a feature that detects errors such as typos, which are then reduced). This is especially helpful for substantial code bases. Users can also add their desired tools through the Python language plug-ins and still write scripts within the PyCharm Integrated Development Environment. JetBrains frequently updates the PyCharm software with new features (Corbett 2014 p. 8, Sale 2014 pp. 10-12).

3.1.5 Java Programming Language

There are three reasons for further explorations of the Java Programming Language: firstly, the fact that Java is a high level language; secondly, Java is one of the most in-demand programming languages in the world, and thirdly, Java is also integrated with internet browsers, which makes further code developments simpler.

Java was developed by James Gosling and Patrick Naughton and their team of engineers who worked for Sun Microsystems. Together they were referred to as "The Green Team". Originally the Green Team worked on a system that would integrate various electronics with computer technology. By 1991 the team created a programming language which was called "Oak". The team used this programme to create a television remote control that existed within a box. This looked similar to today's decoder systems. Their new product was named "*7", but its launch in 1992 was a disaster. For two years after its launch, Sun Microsystems tried to sell their product to various companies, but no one was interested. The Oak was later renamed Java as the registration of the name Oak was already being used by another company's computer programming language. In 1994 Sun Microsystems saw a new opportunity for Java with regards to redirecting their programme to the ever growing World Wide Web. They decided to link Java with an internet browser, collaborated with Netscape Navigator Internet Browser and Java launched in December 1995. As a result they became an overnight success. Currently Java is the most commonly used programming language for various user applications and electronic devices in the world. Java has been used to code games, car radios, televisions, mobile phones and global positioning systems among many other applications (Doke 1999 pp. 3-6, Oracle 2015).

Java is an object orientated language, which means that Java can define data structures and influence objects within its code. Unlike some other programming languages, which run programmes directly from the user's hardware, Java uses its own language, the Java Virtual Machine. When a programmer compiles a project in Java, it converts the newly written code into Java bytecode, after which the developer runs the code and it compiles the bytecode at runtime. These independent, cross platform perks are why Java is so popular and as a result, has the slogan "Write once, Run anywhere", which was established in 1995 by Sun Microsystems. Java is a high level language that can run on most platforms. The Java Development Kit (JDK) contains a software application named Java Runtime Environment (JRE) which helps programmers run their Java software in a World Wide Web browser. The Java Runtime Environment integrates the Java Virtual Machine application (Bryant 2011 pp. 15-35, Viescas 2015).

3.1.6 jGRASP Integrated Development Environment

jGRASP has won many grants and is widely used by South African programming developers. It includes an advanced debugging system and its user interface has a simple layout. jGRASP has won many prestigious grants including one from NASA.

jGRASP is part of a research project which was created in 1999 in the Department of Computer Science and Software Engineering at the Samuel Ginn College of Engineering at the University of Auburn in Alabama. The software was created by Larry Borowski and James Cross. The original version of jGRASP was named GRASP (Graphical Representation of Algorithms, Structures and Processes), which was a development environment tool that the team created in 1988. Cross wanted to make it cheaper for developers to download coding software and as a result they created GRASP. GRASP is a code examination tool which received funding from many prestigious foundations, namely the NASA Marshall Space Flight Centre (1988-1995), the Department of Defence Advanced Research Projects Agency (ARPA) (1995-1996), the National Science Foundation (1998-2012) and the Defence Information Systems Agency (DISA). The team wanted to further develop GRASP and after re-writing and re-designing GRASP in the Java computer programming language, they later named their new software jGRASP. In order to make the appropriate changes to GRASP, Cross decided to perform a research study on computer science students. The results showed that the majority of the students preferred to develop their codes using a control structure diagram (CSD). Besides this change, the team made many other changes to jGRASP and the latest version (2.01-09) was released in October 2015. JGRASP has over 3.2 million users in 188 countries around the world, of which South African jGRASP users rank third (Garrido 2009 pp. 418-423, jGRASP 2015).

jGRASP is an Integrated Development Environment (IDE), which aids programming languages such as Java to compile and run programmes. JGRASP can also be used for alternative programming languages (other than Java), as a source code editor. It uses operating systems which are cross-platform and is also popular for its debugging system. jGRASP has more functionality than GRASP, as it does not only include the features that GRASP harnessed, such as CSD, Complexity Identification and Debugging, but it also includes a unified modelling language (UML).This is a graphical language that has the ability to specify the limits of a software intensive system. jGRASP is simple and quick to install; after users' install the Java Development Kit (JDK), they just need to install the jGRASP software which can be downloaded for free online. The new version of jGRASP (version 2.0.2) is expected for release in 2016 (Thao, Klang and Talberg 2006 pp. 1-8, Litvin and Litvin 2015 pp. 1-11).

Having further explored the three programming languages, it became apparent that I needed to learn how to programme in all three languages in order to make an informed decision as to which programming language would best accommodate the developments of this study.

3.2 Advantages and disadvantages of Common Music, Python and Java

After I learned to programme in Lisp, Python and Java, the characteristics of these languages were compared with one another, in order to establish which programme best pertained to the end goal of this study.

Some of the required features for the final programming language, in accordance with possible future developments, were as follows: firstly, that the programming language was cross platform. Secondly, it was easily accessible. Thirdly, it could be integrated with online advancements by running applications in a World Wide Web browser. Finally, it needed to be in demand for both music technology and chess computer technology users.

3.2.1 Advantages and disadvantages of Lisp

After I learned how to code in Lisp it became apparent that it is a language that lends itself well to music composition. Common Music is an application which exists within Lisp which unfortunately means that one would need far too many dependencies to implement programmes at an application level. Lisp is an old programming language. This makes it very difficult to find technological support, as the continued use of the older programming languages has been overtaken with broader and more inclusive options such as Java or Python. Modern languages are technologically less challenging to use on a broad range of devices. Common Music cannot aid a code that is easily accessible to the modern world. In order to listen to the chess sounds, individuals would have to download Grace (the Common Music user interface), thereafter pasting the set codes into Lisp, SAL or S7 Scheme. This could become problematic and frustrating as the files cannot be accessed online.

The fact that Common Music is an old programme can also hinder the growth of further improvements with regards to this thesis, as I would have to change to another programming language in order to further develop *The Sound of Chess Programme*. Although Common Music is designed for music composition and can solve all equitable algorithms, the aim of this study is not to compose music, but rather to produce representations of chess into sound. The development of this study could include integrations with applications, for example an interactive chess board, which would make coding the variables in Common Music impractical (Van Rossum 1997, Simoni 2003).

Disadvantages of the Common Lisp programming language include the following aspects:

- Common Lisp cannot be used unless the prerequisites are installed such as Grace for Common Music. This prerequisite hinders support and the expansion of the Lisp programming language.

- The Common Lisp programme is not as widely accessible as Java or Python and can only be run on a limited number of devices.

- Common Lisp cannot be integrated with other technologies as easily as Java or Python, as it has limited input types such as text files.

- Common Lisp coding can only be executed whilst running the application in Graphical Real-time, whereas Java and Python offer a broader variety of compilers.

- There are many versions of Lisp, but unfortunately most of them are incompatible with one another.

- Common Music Lisp is an old programming language which makes it difficult to implement complex programming that is outside of the fixed framework of Lisp (Van Rossum 1997).

When comparing Common Music Lisp to what was required for this study, it turned out that it was not modern enough, and as a result it could hinder future music programming developments with regards to *The Sound of Chess Programme*. Common Music pertains to music technology programmers, which would make it difficult to integrate with chess technologies. As a result it was decided not to programme *The Sound of Chess Programme* in Common Music Lisp, as it did not harness the majority of the required features.

3.2.2 Advantages and disadvantages of Python

Python is easy to grasp for beginner programmers, as the coding is simple due to relaxed rules. It uses fewer lines of code compared to those of Lisp and Java programming languages. The reduction in code can be attributed to Python's built-in high level data types (different types of variable objects) and its dynamic typing (the ability to switch between variable object types throughout the code).

When the speed of Python and Java are compared, Java harnesses a faster run-time. The Python developer does not need to declare any variables or data types at the start of the code because, whilst it runs the code, the application will need to examine any variables and look up their types automatically. For example one could have a line of code X+Y: whilst Python would have to look up the data type for X and Y before running the code, Java will start running the code immediately, as the known data types are already declared. Declaring the variables beforehand results in an increase of speed at which the application can perform. With an increased use of functions, the difference in speed between the two languages becomes more noticeable. Although Python programmes generally run more slowly than Java programmes, Python programmes do take less time to learn and develop, due to the simplicity of the code. Java supports more features such as graphics, networking and object oriented programming (Flanagan 2005 pp. 3-6, PSF 2015).

Other disadvantages of the Python programming language include:

- Python requires a Python Enable Compiler, which is not available to all operating systems or devices.

- Python is not as commonly used with regards to computer and mobile devices compared to other programming languages such as Java.

- Meta programming which is available in Lisp and Java is not suitable for Python's data types, which means that Python does not have the ability to read languages' programming as if they were data.

- Python runs more slowly than Java due to the high run-time of Java.

- Python will bind the number to a binary variable which loses the exact value often needed in complex musical notes, whereas Java can perform exact integer or floating point (for example: 0.01) functions.

- Python requires advanced code as it does not have a built in function which can run concurrent tasks (Ferg 2009, Mikoluk 2013).

All the features which this study required were present in the Python programming language although when further compared with Java, Python had a slower run-time. Python cannot read meta programming as if it is data, and lastly with regards to music, Python could not always perform exact integer functions, which was problematic for certain musical rhythms which could hinder possible future developments with regards to later implementations of rhythm. As a result *The Sound of Chess Programme* was not programmed in Python.

3.2.3 Advantages and disadvantages of Java

Due to the popularity of Java it is guaranteed to run on most devices, for example it can run on televisions, appliances and mobile devices which can run the Java virtual machine. Java is faster than Python due to its run-time.

In Java one must declare the variable only once, at the start. It has fewer high level data types (for example: integer, string, date), but it is still suitable for most applications. This distinct difference between Python and Java can be noted in the length of code required to write various programmes that perform the same function. Java is known to require roughly three to five times more lines of code to compile as compared to Python. Other than this, there are many more advantages to coding in the Java programming language (Van Rossum 1997, Dale and Weems 2008 pp. 25-46).

Advantages of Java are as follows:

- Java runs similarly to the programming language C++, but Java is renowned for posing fewer problems than the C++ language.

- In order to run applications, the device is required to have the Java Runtime Environment (JRE) installed. Java by Oracle is the most popularly used of these runtime environments and they are installed in website browsers such as Internet Explorer, Chrome and Firefox.

- Java Virtual Machine (JVM) is used in the development of applications to run the capabilities of the code run on a wide range of machines and/or devices. Another benefit of Java is that it is platform independent which means that the Java Virtual Machine (JVM) receives a specific set of instructions that are capable of running on any platform which supports Java.

- Other coding languages are dependent on fixed systems that do not allow their code to run on other systems. The JVM runs the code against all the known Java supporting systems to make sure it is executable and that the code runs smoothly.

- The Java language is object orientated which means that everything that is written in the language is an object or has descended from a main object or object class.

- Advanced classes are supported in Java adding more appeal for users who are coding advanced applications. Java is capable of handling multi-faceted classes such as strings, arrays and threads as well as providing exception (error) handling feedback.

- Java has the ability to handle input and output classes to get or send information to a huge number of mediums. Some of these mediums include different types of computer files, websites, databases, electronic inputs and physical inputs, for example sensors or buttons.

- Java is able to handle network classes that allow communication between two or more network devices over a local area network or across the Internet.

- Java comes with the Java Virtual Machine which precedes the Abstract Window Toolkit for creating platform-independent applications. The JVM allows one to run the code across multiple platforms and various devices upon which the Java code is installed.

- Java is ideal for web programming and provides an application programme class that lets a programmer create Java programmes that can be downloaded and run on a browser. The applet class reduces the strain on the network connection and allows the code to run smoothly and effectively, regardless of the connection speed. This can be seen in many game applications where the Java code is downloaded before the game starts.

- Java allows for large applications to run on the World Wide Web.

- Another benefit to the functionality of Java is that it does not require programmers to free allocated memory before the application is executed, which makes the language easier to code in and less likely to run into errors.

- Java has its own standard library which includes various classes and functions which are written in the core language and form part of the Java programme itself. The Swing Library provides a programmer with many options and functionality when creating applications. The Swing Library consists of different components which are platform independent as they are written in Java. Components found in Swing include scroll panes, tables, applications, lists, message boxes and trees (Thompson 2008 pp. 15-24, Boyarsky and Selikoff 2014 pp. 39-42).

The Java programming language had all of the required features which met all the needs that were identified at the beginning of this section. Creating programmes in Python was quicker, but it was not as scalable as Java. The benefits of programming in Java with regards to functionality, Java support and future advancements pertaining to this study far outweighed the other programming languages. It was decided that *The Sound of Chess Programme* would be developed in Java.

3.3 Creating the variables

This section explains the way in which all the final variables were implemented. Prior to creating the variables it was important to establish the entities of chess which required Western Classical Music variables. The three main entities of chess for this study are as follows: firstly, the chess board that serves as a grid, which gives the coordinates of the pieces; secondly, the pieces and the ways in which they move; lastly, the decision making process, which is the time a chess player takes before making each move, which for this study includes the external factors that can attribute to those decisions.

3.3.1 The chess board

At the beginning of this study it was assumed that the pitch variables would be assigned to each chess piece, but upon further exploration of the history of the chess pieces in Chapter Two, it became impossible to produce logical explanations as to why a chess piece had to give a particular sound. There was no justification for making a Queen sound a certain way. As a result the logical method was to assign the pitch variables to the chess board. This also makes it easier to recognise patterns aurally and visually in the experiment, which is important as both chess and Western Classical Music require pattern recognition in order for a person to grasp the content at a proficient level. The chess board is used as a grid on which to find the co-ordinates of each chess piece, as only one piece can stand on a square at a time. For this study, the chess board remained a grid that plays the co-ordinates of each chess piece via sound.

The chess board consists of 64 squares that are laid out as an 8X8 grid. Horizontally the board is alphabetised, from A through to H and vertically it is numbered, from 1 through to 8. In Western Classical Music if one were to apply the octave scale to the horizontal rows of the chess board, then the first block would start on the note A and the last block, which is H on the chess board, would end on the note A an octave higher. For example:

Table 3.1

Chess:	A	B	C	D	E	F	G	H
Music:	A	B	C	D	E	F	G	A (an octave higher)

When deciding on the representations for the vertical columns, I was seeking numerical musical variables, in alignment with the vertical representations on the chess board. After much thought, octave ranges were added to the columns. I chose to represent the variables for pitches diatonically because it is too advanced for some beginner musicians to aurally distinguish between semitones and quarter-tones and it was important to use pitch representations which would make chess moves easy to hear for further analysis. Distinguishing between diatonic tones stimulates pattern recognition and calculation. Griggs explains how both calculation and pattern recognition is used differently when comparing beginner and advanced chess players and musicians (Griggs 2013). By using diatonic chords a master chess player, with an untrained aural ability, is more likely to be able to recognise chess patterns in the diatonic sounds compared to quarter-tones, for example. The perfect octave pitches were assigned to the horizontal ranks of the chess board. The implemented diatonic octave scale spans eight octaves in total over the chess board files. The eight octaves aid an untrained ear with high pitches and low pitches. When applying the ranges it was noted that if the numbered ranges were to have started on A1, then they would have ended on A9, which would have been too far out of range. As a result they were implemented as A0 (27.5 HZ) through to A8 (7040 HZ):

Table 3.2

	a	b	c	d	e	f	g	h	
8	**A7**	**B7**	**C8**	**D8**	**E8**	**F8**	**G8**	**A8**	8
7	**A6**	**B6**	**C7**	**D7**	**E7**	**F7**	**G7**	**A7**	7
6	**A5**	**B5**	**C6**	**D6**	**E6**	**F6**	**G6**	**A6**	6
5	**A4**	**B4**	**C5**	**D5**	**E5**	**F5**	**G5**	**A5**	5
4	**A3**	**B3**	**C4**	**D4**	**E4**	**F4**	**G4**	**A4**	4
3	**A2**	**B2**	**C3**	**D3**	**E3**	**F3**	**G3**	**A3**	3
2	**A1**	**B1**	**C2**	**D2**	**E2**	**F2**	**G2**	**A2**	2
1	**A0**	**B0**	**C1**	**D1**	**E1**	**F1**	**G1**	**A1**	1
	a	b	c	d	e	f	g	h	

After establishing the pitches of each square on the chess board, these pitches needed to be coded into jGRASP, which is Java's Integrated Development Environment. When coding in a computer programme that uses Musical Instrument Digital Interface (MIDI), each key is assigned a key number. For example: Middle C (c4) is equal to the key number 60, hence when c4 is played on the chess board, the coded key number 60 is assigned to sound in jGRASP, as Middle C.

This chess board includes the assigned key numbers for each square of the chess board:

Table 3.3

	A	b	c	d	e	f	g	h	
8	**105** (A7)	**107** (B7)	**108** (C8)	**110** (D8)	**112** (E8)	**113** (F8)	**115** (G8)	**117** (A8)	8
7	**93** (A6)	**95** (B6)	**96** (C7)	**98** (D7)	**100** (E7)	**101** (F7)	**103** (G7)	**105** (A7)	7
6	**81** (A5)	**83** (B5)	**84** (C6)	**86** (D6)	**88** (E6)	**89** (F6)	**91** (G6)	**93** (A6)	6
5	**69** (A4)	**71** (B4)	**72** (C5)	**74** (D5)	**76** (E5)	**77** (F5)	**79** (G5)	**81** (A5)	5
4	**57** (A3)	**59** (B3)	**60** (C4)	**62** (D4)	**64** (E4)	**65** (F4)	**67** (G4)	**69** (A4)	4
3	**45** (A2)	**47** (B2)	**48** (C3)	**50** (D3)	**52** (E3)	**53** (F3)	**55** (G3)	**57** (A3)	3
2	**33** (A1)	**35** (B1)	**36** (C2)	**38** (D2)	**40** (E2)	**41** (F2)	**43** (G2)	**45** (A2)	2
1	**21** (A0)	**23** (B0)	**24** (C1)	**26** (D1)	**28** (E1)	**29** (F1)	**31** (G1)	**33** (A1)	1
	a	b	c	d	e	f	g	h	

3.3.2 The chess pieces

In this section, the movements of the chess pieces were implemented with music variables.

In chess notation the co-ordinates of each piece are recorded, for example e4. This means that the Pawn has been moved to e4. The senses of touch and vision help one to comprehend where a chess piece was positioned prior to being moved to a new square, which is why chess notation does not indicate the original position of a chess piece. The notation only states where a piece is newly placed. For example if a piece moves to e4, then the chess notation does not record that the piece moved from e2 through e3 to get to e4. For this study, the journey was interpreted as intervals in Western Classical Music.

Western Classical Music requires the distance between sounds in order to create, play or trace the harmonic and melodic lines. When analysing a chess game, one has to retrace what was recorded in the notation, by re-playing each move on a chess board. Some advanced players can understand where the pieces are located without consulting a chess board, as they have acquired a visual memory of the board in accordance with where the pieces are on the squares. This is similar to perfect pitch in music. In music, if the notes or variables are not all present, it is difficult to grasp the harmonic or melodic composition. The relationships between the notes create the structure of the music.

Chess notation lacks the distances which the pieces have travelled, and as a result it was important to make the journey of the chess pieces easy to trace both theoretically as well as audibly. In order to detect the movements of the chess pieces, via intervals in sound, it was decided that two notes would be made to sound for every chess move made. For example, a move such as e4 was coded as e2, which sounded the second octave's E (e2), which was then followed by e4 that played the fourth octave's E (e4). As a result two notes were now sounding consecutively and these two notes represented one move. This made it simple to hear the interval of the chess piece as it moved through the third octave's E (e3), even although e3 did not sound. This made it possible to hear that the chess piece had jumped two octaves.

The chess piece variables do not indicate which piece is moving. All that is indicated is the note of the square on which the piece is standing, followed by the note of the square to which it has moved. I wanted to further explore if each piece's intervals could be distinguished aurally in the final *The Sound of Chess Programme.* In Appendix A and presented on the accompanying disc (see Appendix D for instructions to listen to *The Sound of Chess Programme*) a list of random standard moves are provided for each chess piece in order to hear their particular intervals in isolation.

3.3.3 The decision making process

This section is presented in two parts, namely, the aesthetics of the decision making process, and secondly the external factors which can influence a chess player's decisions (morale).

3.3.3.1 Aesthetics of chess

In this study I referred to the decision making process of a chess player (rhythm of chess) as the aesthetics of chess. The aesthetics of a chess move can be described in two stages. In competitive chess tournaments a player has to move the piece that they first touch, which is why the first stage is when a player picks up the chosen piece. The piece is then off the grid whilst the player decides where it needs to go. The chess piece may remain in the chess player's hand until the second stage, which is when the player puts the piece down on the chosen square. The move is only considered over once the player lets go of the piece, notates the move and then presses their clock.

I decided that making one sound per move did not express the suspense that a player internalises whilst playing chess. Therefore the aesthetics of a chess move require two actions, picking up and putting down. This part of the decision making process was automatically implemented in the chess piece's variables in the previous section, which produced two sounds for every one chess move: one for the square on which the piece was picked up, and another for the square from which the piece was put down. From this the first part of the aesthetics with regards to the decision making process was already solved and applied.

3.3.3.1 The decision making process

The second part of the decision making process was briefly discussed at the end of Chapter Two. This pertained to the surroundings or morale of a chess player, which can directly affect his or her decision making process.

Competitive chess players are accustomed to pressure, because if a player makes one incorrect decision it could ultimately cost them the game. Some chess players might relate to the scenario of being distracted by a spectator during the decision making process of their next intended move. As a result of this distraction, chess players will often not move until the spectator walks away or is removed by an arbiter. The reason for this is that any distraction in chess can impact the result of the game, which is why no talking is permitted during a round of chess. This is similar to music where it is difficult to perform when external factors are distracting you.

The time it takes for a chess player to make a move, whether it is prior to picking up a piece, whilst holding the piece off the grid or concentrating during their opponent's move, consists of moments where external variables can affect the speed with which the next move is implemented. For example if a player feels intimidated by their opponent, they might be more cautious in executing their next move. These variations of speed with regards to the implementation, in which the decided moves are played, were translated into musical rhythm for this study.

Chess notation does not record the times of each move made in a game of chess, but these various speeds are recognised as an entity of chess for this study. The rhythm of chess requires a personal decision making process with regards to an individual's surroundings and confidence in committing to the intended move. As a result the variables for the tempo in which to play the chess sounds were made *ad lib*. Therefore if the chess sounds were to be performed, the musician could decide at which time to play the next notes in accordance with their environment and morale. The results of the implementations of the variables for the three entities of chess were as follows: firstly, that the chess board variables were represented as pitch in music. Secondly, that the chess pieces and the aesthetics of the decision making process were depicted as musical intervals. Lastly, that the second part of the decision making process, which includes the external components surrounding chess, was assigned a variable of musical rhythm, which was to be played *ad lib*.

3.4 The Sound of Chess Programme

At the proposal stage of this thesis, I was not going to programme the chess sounds myself. The aim was exclusively to translate the chess moves into sound variables. As the study progressed, it became more apparent that in order to better understand whether the implementations of the created variables were being represented properly, one would have to hear them. Consequently I programmed *The Sound of Chess Programme* for this thesis.

3.4.1 The coded pitches

After creating the code in jGRASP, which is the Integrated Development Environment for the Java programming language, the variables were substituted into the programme.

The pitch variables, which represent the chess board, along with the interval variables, which represented the movements of the chess pieces as well as the decision making process with regards to the characteristics of chess, were integrated. In order to programme these two separate variables, they needed to merge within the code. Hence the chess moves that were used to test the codes were inserted into a text document with their MIDI key numbers, which were assigned to the chess boards in the previous section. For example if a Pawn moved to e4, then the text document needed e2 and e4 pre-reflected with the MIDI key numbers, 64 and 40 in it. This resulted in the programme automatically playing the text file variables, when the code was instructed to run.

After coding the variables into *The Sound of Chess Programme*, it was easier to explore the different possibilities of the chess sounds. When the variables were played in a monotone succession of one note at a time, it sounded much like a modern form of Gregorian chant, but it did not make logical sense to have two white pieces' notes, followed by two black pieces' notes, play in succession. The code was thus restructured so that the variables would sound two notes at a time, in a chord-like fashion. The relationship between the two sounds made logical sense to the ear and the chords were more intriguing. Hearing the distances between the two notes at once made it easier to identify the intervals between the sounds of each moving piece. Out of interest, a few random Knight; Bishop; Rook; Queen and King moves were created in jGRASP text folders, in order to hear the intervals of the pieces' moves in isolation, and determine whether the intervals of each piece could be distinguished aurally.

3.4.2 The coded rhythm

The rhythm variables, which represented the decision making process with regards to surroundings and internal emotional content, were implemented as *ad lib* when the games are being performed and were coded to play at a random rhythm in *The Sound of Chess Programme*, for the following reasons:

In the beginning of this study I wanted to time each move made by every player, in a random chess game at a local tournament which I then wanted to simplify into musical measurement ratio. This would result in there being only three variants of rhythm for *The Sound of Chess Programme* in jGRASP. This would not accurately capture the variation which each individual chess game harnesses, and would cause misrepresentations of chess games. Apart from that the time sequences indicate large intervals that, on their own, would be hard to grasp. Furthermore, one has to differentiate between the time spent picking up and putting down a chess piece and the time spent in the decision process which is governed by a player’s emotions prior to picking up and moving a piece.

Chess does not represent time; as a result, the time a player takes is naturally integrated with their surroundings and morale, whereas in Western Classical Music, rhythm and interpretation are dealt with as separate entities within the representation of music. As a result it was not possible to accurately implement rhythm into music variables. The closest representation in Western Classical Music, to what the rhythm and interpretation of chess harnessed, was to play the sounds *ad lib*. I decided that in order to best represent *ad lib* within *The Sound of Chess Programme*, a function had to be written that played the chess notes at a random time. The time frame is much tighter in music with its regular pulses/meter and rhythm that fits into this. This worked but it became apparent that the external phenomenon and interpretation for both chess and Western Classical Music cannot be represented within a computer programme.

3.4.3 Chess manipulations

There are four main chess manipulations which will be discussed in the following order: capturing a piece, castling, check and checkmate.

When a piece is captured off the board, it is coded as one semitone higher in *The Sound of Chess Programme*. For example, if a piece was captured on c4, then instead of MIDI key number 60 sounding, the key number 61 would sound. The reason for this was to have some aural indication that something had happened. Selfridge-Field explained that when a MIDI key number is played, there is no way of knowing the difference between certain intervals (Selfridge-Field 1997 p. 6). For example if C is coded as 60, 60 could also mean that B sharp or D double flat are sounding.

Therefore, I calculated that it would not matter if the sound of the captured piece indicated up or down, as it had no specific attachment to sharps or flats. The decision for the pitch to go up was in order to mimic the elation which is felt when one captures a piece; hence it was assigned to go up.

Thereafter came the programming which would indicate and represent when a player castles. Castling is when a player swops their King with their Rook in order to protect the King. This can happen firstly if the King has not yet moved in the game and secondly, if there are no pieces standing between the Rook and the King. It is a basic rule that one may not castle through check (a piece threatening check on any of the squares which the King would need to intercept).

The Sound of Chess Programme produces two sounds as one chord, for every chess move which is made. As a result, a chess move makes the two squares that the piece touched sound as one chord. When a piece castles it requires two pieces and four squares, thus it was coded that when the King castles, all four squares will sound as one chord in *The Sound of Chess Programme*. This means that the Rook and the King touch four squares in one move. These castling chords can only be a variation of four possible chords. The instances of this follow in the next paragraph.

When white castles, the sounding MIDI key numbers are either (28-24 and 21-26) or (28-31 and 33-29), which is dependent on which side the King castles (King's side castle or Queen's side castle). This means that if black were to castle, the two possible chords that would sound include (112-115 and 117-113) or (112-108 and 105-110). The programme was designed to recognise these successions automatically, regardless of where this is situated within the playing of the coded game. After playing the castling chords, the programme mechanically continues to play the rest of the two note chords.

For the purposes of this study, check and checkmate were eliminated from being made to sound, for two reasons: firstly, there was a variety of ways in which to code both check and checkmate, but in order to find appropriate representations for check and checkmate, this study required more research. Secondly, it is not a prerequisite for a player to say check under tournament conditions, hence *The Sound of Chess Programme* not being required to say check.

Other possibilities that were explored for *The Sound of Chess Programme* included a code which could detect which colour won the game as well as if the outcome was a draw. At Nationals a player has to indicate which colour won, by placing the winning colour's King in the middle of the board and if the game ended in a draw, both Kings are placed in the centre. In order to recreate this gesture into music, one could indicate that the game was over by playing an e1 (28) or an e8 (112) at the end of the piece as the e8 would indicate that the black King had won and the e1 would indicate that the white King had won. The sounding of both e1 and e8 would indicate a draw. Thus if white won, the note would be the lower sounding E while if black won, it would be the higher sounding E. In order to write the code for this action it had to be expressed in a way that if the game ended with an odd number of moves, white had won and if the game ended with an even number of moves played, black had won. Unfortunately problems arose when trying to create a function which would detect a draw.

When I started to create *The Sound of Chess Programme*, Scholars Mate was used to test different programming techniques. For the first attempt at hearing a chess game sound, each note was coded separately with an assigned duration. Although this was primitive, it was a wonderful experience to hear the desired pitches for the first time. After all the variables were translated into code, and *The Sound of Chess Programme* was complete, Scholars Mate was again the first chess game which was notated into the jGRASP text file and used to sound in *The Sound of Chess Programme*.

3.5 Three sounding chess games

This section discusses the three chess games which were made to sound in *The Sound of Chess Programme.* Refer to Appendix B for the chess notation and correlating MIDI pitch numbers of all three chess games. Appendix C shows these games translated into Standard Western Music Notation. The sounds of all three these notated chess games can be heard on the accompanying disc. The three games range from beginner to master level and it was decided that the games that would be programmed were: Scholars Mate, A. Dole against S. Batyrov and G. Kasparov against Deep Blue.

The beginner game made to sound in this thesis is Scholars Mate. Scholars Mate is a chess game commonly referred to as “cheap chess”. It is most frequently played by beginner chess players and it is forbidden to play Scholars Mate in a FIDE rated tournament. The coding of Scholars Mate resulted in a short piece, as it is a short game; therefore the text file is shorter to run than the other two chess games take to run in jGRASP.

The intermediate chess game that was coded to sound was supplied by Anant Dole. The game is by Anant Dole and his opponent S. Batyrov, who was a Grandmaster at the Parsvnath International Open in 2008. Dole was one of the top South African Junior Chess players and he has represented South Africa at many world chess championships. Currently Dole is studying to obtain a Masters Degree in Electrical and Computer Engineering at Dartmouth University in New Hampshire. There are a vast number of chess players in the world: the pool is immense with regards to beginner chess players, intermediate chess players, international grandmasters and world champions. As a result, although Anant Dole was one of the best players who represented the South African Junior Chess team abroad, when compared to the novice game of Scholars Mate and former World Chess Champion Grandmaster Garry Kasparov, Anant Dole was calculated to be the median for this study.

The advanced chess game that was programmed to sound for the study left Garry Kasparov the loser against “Deep Blue”, an IBM chess engine. In 1996 the “man versus computer” challenge made Kasparov the first chess Grandmaster in history to lose to a computer. After “Deep Blue” it became more frequent for computer programmes to beat international grandmasters.

Although the three games programmed in *The Sound of Chess Programme* have not undergone a research survey, there are entities which can be assessed. All of the sounding games harnessed chords which I found pleasing to the ear, as there was a melodic line I could sing along with. Prior to executing this study, I assumed that the pieces would produce out of context sounds, but after analysing the chess sounds further, did not sound as strange as I had initially anticipated. Rhythm in chess with regards to *The Sound of Chess Programme* is represented as *ad lib*. A performer of the chess music can play the pitches *ad lib* in order to mimic the decision making process of a chess player (rhythm of chess); this process is what I referred to in this thesis as the aesthetics of chess. If the sounding chess pieces were to be performed by a musician, it is probable that with human interpretation of tempo/rhythm, the decision making processes experienced by chess players could come to life. After working with the coded sounds, I realised that I had developed an auditory memory of all three chess games, when they were played via *The Sound of Chess Programme*.

Chapter Three helped to establish the final variables, as well as the programming language that was used to code an algorithm into which the variables were substituted. The chess sounds were heard for the first time in *The Sound of Chess Programme* and as a result possible future developments were made clearer in this chapter.

Chapter 4

Results: presentation and discussion

This chapter provides the annotated version of the final *The Sound of Chess Programme* and it is explained with extensive detail.

The five questions, which were established in the Methodology (Chapter 1.6), are presented, discussed, answered and concluded in accordance with the findings of this study in this chapter.

The questions include the following topics: Firstly, could Algebraic Chess Notation be used as a new music notation? Secondly, could *The Sound of Chess Programme* aid other games to sound? Thirdly, what influences could a sounding chess game have on music study? Fourthly, could the chess game be made to sound in a melodic structure? Lastly, could a sounding chess game result in an individual learning to play chess by ear?

4.1 The final Sound of Chess Programme with annotations

This section explains each function in *The Sound of Chess Programme*.

The coding below is how one programmes the loading of all the required pre-built in functions:

```
importjavax.sound.midi.MidiSystem;

importjavax.sound.midi.Synthesizer;

importjavax.sound.midi.MidiChannel;

import java.io.*;

importjavax.swing.*;

importjava.awt.*;

importjava.util.*;
```

```
importjava.applet.Applet;
importjava.awt.Graphics;
```

The class name below is a blueprint for individual objects. It saves time so that one does not have to declare processes for each individual object:

```
public class ChessSoundDynamic01
{
```

Below is the code for a new object that holds all the variables. In this case it is holding the variables of the chess moves:

```
ArrayList<Integer> storeMoves = new ArrayList<Integer>();
```

This declares the file name which will be implemented later on in the code. “Test” is assigned as the filename so that the programme will run without exceptions (errors), if no filename is provided by the user.

```
String fileName = "Test";
```

This code terminates the main process, should any exceptions occur:

```
public static void main( String[] args ) throws FileNotFoundException
  {
```

The “new” keyword is a Java operator (JavaScript function), which creates the object initialization. The “new” operator is then followed by a call to a constructor, which in this case is the class that was declared earlier:

```
new ChessSoundDynamic01();
  }
```

This code executes the menu function:

```
ChessSoundDynamic01() throws FileNotFoundException
  {
```

```
menu();
  }
```

The menu executes the function below:

```
//Menu Start.
void menu()throws FileNotFoundException
  {
```

This executes the welcome message:

```
JOptionPane.showMessageDialog(null,"Welcome to Sound of Chess.");
    String inputMenu=(JOptionPane.showInputDialog("Sound of Chess \n 1: Specify File Name \n 2: Update from Text File. \n 3: Play Chess Game \n 10: Quit Application "));//Creates the menu for the application in a re-occurring switch statement.
```

The code below instructs the application to get (fetch) the input, which is the chosen menu:

```
int menu = Integer.parseInt(inputMenu);
```

If there is no menu choice, then the application must automatically run the coding below:

```
if (inputMenu==null)
    {
```

This code orders the application to close:

```
System.exit(0);
```

If a user selects “cancel” then it must be depicted on the user’s screen:

```
System.out.println("User hit cancel.");
    }
```

If the length of the input is 0, which means that nothing has been selected, then the programme will notify the user and will re-show the menu options:

```
else if(inputMenu.length() == 0)
    {
JOptionPane.showMessageDialog(null,"You need to enter a value, please try again.");
inputMenu = (JOptionPane.showInputDialog("Sound of Chess \n 1: Specify File Name \n 2: Update from Text File. \n 3: Play Chess Game \n 10: Quit Application "));
    }
```

If the menu is not equal to ten, then the application below must run automatically:

```
while(menu != 10) //Exits Menu at Value of 10.
    {
```

This is the function that is used to run a menu selection:

```
switch(menu)
    {
```

If a file name is selected then the programme runs the function called FileNameGrabber. This requires the user to select and insert the name of the file which they would like to see in the jGRASP integrated development environment. In this case the chess game's text documents (files containing the variables for the chess pitches) include: Scholars_Mate, A_Dole vs S_Batyrov, G_Kasparov vs Deep_Blue as well as the individual chess moves. These can each be selected and run in jGRASP:

```
case 1: fileNameGrabber();
break;
```

After selecting and inserting a text document as described above, the second application in the JOptionPane is run. Once this second option is selected by the user, this code will load all the values from the chosen text document into the jGRASP environment:

```
case 2: moves();
```

```
break;
```

The user then has to select the third option in the JOptionPane window, which will then run the "play function". This means that the chess music audio will play:

```
case 3: play(); //Sounds Loaded Noted & Applies Necessary Logic.

break;

        }

inputMenu = (JOptionPane.showInputDialog("Sound of Chess \n 1: Specify File Name \n 2: Update from Text File. \n 3: Play Chess Game \n 10: Quit Application "));

menu = Integer.parseInt(inputMenu);

if (inputMenu==null)

{

System.exit(0);

        }

else if(inputMenu.length() == 0)

        {

JOptionPane.showMessageDialog(null,"You need to enter a value, please try again.");

inputMenu = (JOptionPane.showInputDialog("Sound of Chess \n 1: Specify File Name \n 2: Update from Text File. \n 3: Play Chess Game \n 10: Quit Application "));

  }

  //Menu End.
```

This function instructs the application to read the file which is selected and is implemented after the user selects Option One in the JOptionPane window. It reads the file name and assigns it to its variable:

```
//Get File Name Start.
Void fileNameGrabber()
{
```

This shows the text in the JOptionPane window, after which it reads the selected input and then assigns it to the variable called filename:

```
fileName = (JOptionPane.showInputDialog("Enter file name excluding extension"));
}
//Get File Name End.
```

Here the function moves are run:

```
//Text File Load Start.
void moves()throws FileNotFoundException
{
```

This closes the code that may cause an exception demand:

```
try
{
```

A “scanner” breaks its input into tokens using a delimiter pattern:

```
Scanner movesFile = new Scanner (new File (fileName + ".txt"));
```

This reads all the lines of the code:

```
while(movesFile.hasNextInt())
{
storeMoves.add(movesFile.nextInt());
System.out.println(storeMoves);
```

```
}
movesFile.close();
}
catch (FileNotFoundException ex)// File Not Found Error Catch.
{
```

This code sets the default file name "test" in order to avoid the programme giving errors:

```
fileName = ("Test");
}
}
//Text File Load End.
//Audio Play Start.
```

Here the code runs the “PLAY” function, which means that the audio must be played:

```
void play()throws FileNotFoundException
{
int instrumentType = 0; // 0 is a piano.
int timeRandom = 120; // between 0 & 127
int duration = 200; // Time in milliseconds
int sustain = 0;
Random rand = new Random();
String state = "RegularPlayState";
int winner = 0;
int end = 0;
```

```
int holder = 0;
```

The code “storeMoves.Size” means that the variables stored in the text document are declared (in this case the size of the text file). “i” is declared as counted with a value of 0. When “i” is smaller than the total number of “moves(storeMoves.Size)”, then it runs the code play function which plays the sound. “i” is increased by one digit (i++) after each single sound is played, which means that the application runs until “i” is larger than the total number of “moves(storeMoves.Size)”, this plays all the moves in the record, therefore the full chess game is audible:

```
for (inti = 0; i<storeMoves.size(); i++)
    {
```

Below are debug comments that do not affect the code:

```
    //Debug.
    // System.out.println(end);
    // System.out.println(storeMoves.size());
    //System.out.println(i);
    //Debug.
```

The code below is specially designed to detect when a castling move is made. This starts the castle logic. Castling consists of four possible variables - these four possibilities were coded below:

```
if(storeMoves.get(i) == 28 && storeMoves.get(i+1) == 31 && storeMoves.get(i+2) == 33 && storeMoves.get(i+3) == 29 || storeMoves.get(i) == 112 && storeMoves.get(i+1) == 115 && storeMoves.get(i+2) == 117 && storeMoves.get(i+3) == 113 || storeMoves.get(i) == 112 && storeMoves.get(i+1) == 108 && storeMoves.get(i+2) == 105 && storeMoves.get(i+3) == 110 || storeMoves.get(i) == 28 && storeMoves.get(i+1) == 24 && storeMoves.get(i+2) == 21 && storeMoves.get(i+3) == 26 )//Detect Unique Moves.
```

{

This declares the variable which is to be read later:

state = "CastleState";

}

//Detect Castle Logic End.

//Action Castle Logic Start.

If the variable is set to "CastleState", then the case sound will be played. The code automatically detects when one of the four variables of castling occur in a chess game. The earlier "i" codes are then overrun by the "CastleState" programme:

if (state == "CastleState")

{

try {

Below is the first of two places within the overall code where rhythm is implemented. This "timeRandom" code sets the rhythm at a random time between the desired musical measurements:

timeRandom = rand.nextInt(500) + 250; //Random Time Between 1000ms & 100ms.

Synthesizer synth = MidiSystem.getSynthesizer();//Opening of Instrument.

synth.open();

MidiChannel[] instrument = synth.getChannels(); //Get Instrument Type.

This pulls data and plays it within the Java instrument function:

instrument[instrumentType].noteOn(storeMoves.get(i), 120); //Note in Current Array Section 01 On.

instrument[instrumentType].noteOn(storeMoves.get(i+1), 120);

instrument[instrumentType].noteOn(storeMoves.get(i+2), 120);

instrument[instrumentType].noteOn(storeMoves.get(i+3), 120); //Note in Current Array Section 02 On.

Thread.sleep(timeRandom); //Random Time Delay Between Current Note On and Current Note End.

instrument[instrumentType].noteOff(storeMoves.get(i), 120);

instrument[instrumentType].noteOff(storeMoves.get(i+1), 120);

instrument[instrumentType].noteOff(storeMoves.get(i+2), 120);

instrument[instrumentType].noteOff(storeMoves.get(i+3), 120);

System.out.println("Notes Currently Playing: "' + storeMoves.get(i) + "' "' + storeMoves.get(i+1)+ "' "' + storeMoves.get(i+2)+ "' "' + storeMoves.get(i+3) + ""); //Visualise what Notes are Sounding.

i = i+4; //Cycle to Next Four Array Blocks, this Ensures Notes are Not Played Separately Afterwards.

Thread.sleep(timeRandom); //Random Time Delay Between This Note End and New Note Start.

synth.close(); //Close Instrument.

}

If there is any errors with regards to the variables, then the programme will automatically run the code below:

catch (Exception e)

{

This automatically displays the errors in the user interface:

e.printStackTrace();

```
}
```

If the state of the variables is not in “CastleState”, then the programme will play the variables in the “RegularPlayState” below:

```
state = "RegularPlayState";

    }

    //Action Castle Logic End.

if (state == "RegularPlayState")// Continue Regular Sound Dynamic.

    {

try { //Note Playing.
```

Here is the second time in the code where the rhythm functions are re-implemented, as the code had to be repeated in order to be executed within the “RegularPlayState”:

```
timeRandom = rand.nextInt(500) + 250; //Randomising Time.

        Synthesizer synth = MidiSystem.getSynthesizer();//Opening of Synth.

synth.open();

MidiChannel[] instrument = synth.getChannels(); //Get Instrument Type.


instrument[instrumentType].noteOn( storeMoves.get(i), 120 );  //Note Array Section 01 On.

instrument[instrumentType].noteOn( storeMoves.get(i+1), 120 ); //Note in Array Section 02 On.

Thread.sleep(timeRandom); //Insert Time Break.

instrument[instrumentType].noteOff( storeMoves.get(i), 120);
```

```
instrument[instrumentType].noteOff( storeMoves.get(i+1), 120);

System.out.println("Notes Currently Playing: "' + storeMoves.get(i) + "'    "' + storeMoves.get(i+1) + ""); //Debugging.

i = i+1; //Cycle to  Next Two Array Blocks.

Thread.sleep(timeRandom); //Time Break.

synth.close(); //Close Instrument.

        }

catch (Exception e)

        {

e.printStackTrace();

        }

      }

    }

    //Audio Play End. (End process)

  }

}
```

This concludes the annotations of *The Sound of Chess Programme*.

4.2 Musical chess notation

Could the sounding game of chess aid a new musical notation system and if so would such a system be easier than Standard Western Music Notation for musicians to use?

After creating *The Sound of Chess Programme,* it became apparent that it would be possible to play music from chess notation; hence it could aid a new form of music notation.

If music were to be played from the current chess notation, new and established musicians would have to learn an entirely new notation. The use of Algebraic Chess Notation, for music in the current context of *The Sound of Chess Programme*, would involve much less reading when compared to Standard Western Music Notation because it excludes rhythmic notation, ledger lines, accidental signs, clefs, performance directions and as it consists of no more than two letters and one number at a time, it is quick to read and simple to notate. As a result it would save time if music were to be composed with Algebraic Chess Notation when used in *The Sound of Chess Programmes* context, although the entities of interpretation and rhythm are solely determined by the performer and not the composers, with regards to the variables created for this study. Therefore composer would either need to add dynamic markings above the chess sounds, or they would have to state in writing what they would like to convey regarding interpretation. Rhythm is not represented in chess notation; hence the entity of rhythm in Western Classical Music terms would have to stay *ad lib*. In this case chess notation could be faster to use, but it does not account for one of the most vital entities in Western Classical Music representation, which is rhythm. If a composer does not want to notate rhythms then chess notation is ideal, but for the larger scope of musical compositional intentions, chess notation would require a method in which to notate musical rhythm.

As a result the notation would be easier to read, but if music were to be composed using chess notation, regarding the variables discovered for this study, then music would have to be composed via representations of chess. This means that there are boundaries for a composer, because chess sounds are geared by, and exist within, the elements of chess. These variables can be altered to suit musical entities, but as a result they could not be variables that represent the game of chess, but rather music that is represented within chess notation, which was not the goal for this project.

The three final chess games' sounds are represented for documentation purposes in this project, via Algebraic Chess Notation. The reason for this was in order to understand the assigned variables for the sounds in relation to the chess moves. Therefore chess notation can represent musical elements, as it accomplished in representing the sounds that were played via *The Sound of Chess Programme* for this thesis.

In accordance with the research, the prerequisite of composing with chess sounds is in direct alignment with the representations of chess entities, instead of music entities. Standard Western Music Notation requires a reader to look in many different directions in order to understand all of the performance instructions. Chess notation does not require as much tracking; this is why with regards to reading and composing, chess notation is less time consuming and as a result it could aid simpler usage in general.

4.3 Making other games sound

Could other games or logical systems be used for themes in music composition?

One should be able to translate any logical system or game into sound, but it is dependent on the number of entities that it harnesses, which can correspond with, and be designed into, representing variables, as well as the method with which these variables are applied in relation to logical representations.

In order to represent logical systems and games one would need to understand which entities have to be present in order for the game or logical system to be well represented. For example, in this study, chess harnessed the following main entities: the board, the pieces, the notation and the decision making process of a player. As a result there were four elements needing to be represented in order for the end goal to reflect the aesthetics and functions of chess. Hence if one were trying to make a card game sound, the number of cards in the deck, along with the numbers reflected on the cards, would be two of the entities in the game. The way in which these entities are represented within sound are the key factors. In order for the variables to sound in accordance with a card game, they need to be based on the accuracy of logical representations with regards to linking sound variables with the card entities. Chess does not involve luck. The structure of chess is purely logical and this could also play a vital role, when attempting to make other recreational systems sound, as it is difficult to implement logical variables with something which is contingent.

As a result other games or logical systems can only be made to sound if they have elements which can be translated into sounds based on logical principles and accurate representations with regards to the attributes of the chosen system. When considering specific games which are more likely to be sounded, the games that require chance are not probable for aiding substantial themes in composition; in fact they are more likely to sound sporadically out of context, due to the fact that they do not harness logical structures. Further study could determine whether random variables sound more pleasing than logically structured variables when they are compared to commonly used theme structures in Western Classical Music. I, however, figure that logical variables will aid more pleasing themes for composition.

4.4 Chess's impact on music study

Could a sounding game of chess aid easier music study by introducing an interactive dimension?

A sounding game of chess could contribute to various aspects of music study. The ways in which it could be used include as a teaching tool for beginners to music, an interpretation tool for music as well as for certain aspects of theory and compositional studies. The fact that the sounding chess board is three dimensional and combines three senses - sight, sound and touch - is what makes it an interactive tool. This means that people who are stronger visual learners can see their musical content, and as a result could benefit from learning music via the chess board as well as via music notation. Auditory learners could also better understand their music if they could hear the visually demanding content in Western Classical Music, such as theory.

Children that are too young to play a musical instrument could learn to play music via playing a sounding game of chess. They could learn basic music principles and develop cognitively. The chess board could serve as an instrument, which could enable children to play with the sounds by moving the chess pieces in various directions.

A sounding game of chess could be used as a tool that enables students to reason with their motives in music. When chess moves are made, players have to have an understanding of their moves. Musicians also require a good comprehension of their musical intent, but on a much larger scale than chess.

For music theory studies, the visual interaction with the game of chess could add a new practical music affiliation to or familiarity with the theory content, as the students would be able to interact with theory concepts via a three dimensional apparatus, which can also sound. This could make it easier to relate to the subject matter.

The sounding game of chess could also be used as a compositional tool, as instead of producing music via computer programming or on paper, composers could play their compositions on a three dimensional object. As a result it is possible that the sounding game of chess could be integrated with different aspects of music study and it could make those subjects easier to comprehend, by adding another dimension to music. The integration of the sounding game of chess regarding music study would require more development and research.

4.5 Chess and melody

Could chess games be made to sound melodically?

Chess could be made to sound melodically, as there are enough common entities between chess and music which would enable other possibilities with regard to combining chess with music. For this study, the aim was to translate the functions and aesthetics of the game of chess into sound, rather than to create music inspired by chess; hence the purpose of this study was not to make chess sound melodically, for example, music inspired by chess such as the Broadway play, Chess. If the purpose of this project were swopped around, and music had to be represented into chess variables with the aim of the variables reflecting melodic structures, which could then have been integrated with a computer programme, it is highly likely that the result could have sounded melodically. This means, depending on the way in which the variables are implemented, that chess games could be made to sound melodically.

In this thesis melodic lines were established and as a result they could be used as themes for improvisation or compositions. Therefore chess is already able to sound and I surmise that if the variables were altered to reflect a more defined outcome of melody that chess games could be made to sound melodically.

4.6 Playing chess by ear

Could a person play chess by ear?

Whilst working with *The Sound of Chess Programme*, it became apparent that I was remembering the chords and I would sing along. This means that it is possible to remember the pitches and the intervals. This aural memory also made it possible to recognise sounds of certain moves which were commonly made, for example e4. I was also able to hear the difference in intervals regarding each individual piece, which made it easy to aurally distinguish which piece was moving, and lastly, I could tell by ear which chess games were being played.

Therefore as it stands, it is possible for a musician who can hear and understand a basic level of intervals to in fact learn to play chess by ear. If one takes the most commonly played opening moves of a chess game, it is highly likely that these moves could be memorised by chess players and musicians alike.

Chapter Four encouraged further reflections on the final *The Sound of Chess Programme* and assisted in gaining a better comprehension of how sounding chess could coexist with music.

Chapter 5

Conclusion and Recommendations

This chapter includes a focused discussion about the following headings pertaining to the overall research: firstly, the research statement and original contribution to the field of music; secondly, the main findings of the study; thirdly, the limitations of the final research; fourthly, the possible future developments and lastly, the conclusion of the research.

Chapter Five focuses on the outcome of the three final chess games and discusses what the sounding chess games reveal about the development of the chess positions within each game. After this the shortcomings of the sounding variables are assessed and the possible future developments for *The Sound of Chess Programme* are established, followed by the conclusions drawn from the body of this thesis.

5.1 Research statement and contribution

The study aspired to test the representations of Standard Western Music Notation in a logical and ludic experiment. The sounding of chess representations aided a better comprehension of the existential characteristics of both entities. This was conducted by making a game of chess produce sound, which in turn proved that chess and music can be integrated via commonsense principles.

The significance of this study includes two aspects: firstly, it was shown that chess can be notated in Standard Western Music Notation, and it could be possible to substitute other notational systems with sound representations. Secondly, it demonstrated that music and chess are under-notated in different ways. Standard Western Music Notation lacks accurate representations for a composer’s intentions and chess notation lacks representations for external elements that influence the moves that a player implements, as well as the time taken for each move made in a game of chess.

5.2 Main findings

In a game of chess one is faced with many options; the aim is for a player to find the best outcome. This requires logical interactions with each choice that the player is faced with for every move in the game. As a result I realised that all representations are relative, but if the application of the variables made logical sense to me and my interaction with the content, then the variables for this study would be applied in accordance with a desired outcome for this thesis. The variables that are reflected in *The Sound of Chess Programme* include pitch, intervals and rhythm played at random.

The reason why I chose to represent the variables for pitches as diatonic representation, instead of eight quarter-tones, for instance, was due to the fact that it is highly unlikely for the untrained ear to distinguish between quarter-tones. As it is too advanced for some beginner musicians to conceptualise semitones and quarter-tones, it did not make sense to use pitches which are too difficult to hear. This could lead to only auditorily developed musicians being able to engage with the end product. For a project that was introducing the game of chess to musical entities it was important to use pitch representations which would make chess moves easy to analyse within sound variables. This would also aid an inclusive outcome which is audible to both music and chess communities.

For this study the diatonic octave scale was implemented on a grid that spans eight octaves in total. This makes it easier for an untrained ear to distinguish between the sounds as not only is the diatonic scale easier to hear, but the eight octaves involved also aid an untrained ear with high pitches and low pitches. This helps the listener to establish whether white pieces or black pieces are playing at the beginning of a chess game. As one enters the middle game, one starts to hear the high notes and low notes integrate in the middle register. Using the diatonic scale also gave the outcome a way in which to analyse whether the intervals of the chess moves and identification of the chess pieces moving could be heard. As a result wide spanning octaves and diatonic pitches were chosen for the integration of chess with Western Classical Music concepts, with regards to the tools needed to hear and analyse particular positions within the three chess games, in correlation with the final assigned sounds of the three sounding chess games.

The rhythmical aspects were chosen to be represented and performed as *ad lib*, rather than being represented within the 64 squares of the chess board and implemented as hemidemisemiquaver beats. I am aware that 64th notes can be implemented within the 64 squares on a chess board if the aim is to represent music variables. The reason I chose not to apply the 64 squares and rhythmic combinations within the rhythmic variables for the chess games was due to the fact that these rhythms did not translate in accordance with the representations of chess entities. The rhythm of chess is determined by external variables which are independent of the 64 squared chess board. A rhythmical dimension (time and the differentiation of time) is one of the pillars of Western Classical Music and not prescribing a specific duration is by no means excluding the rhythmical dimension if performed *ad lib*.

The other option that I assessed for the integration of musical rhythm with chess rhythm was the timing of each move made in a random game of chess and condensing these times into musical rhythm. Upon further evaluation I realised that this would not work, as the condensed times would not be a direct representation of the three games used for this project; instead this would result in enforcing the calculated times of the alternative chess game onto the three chosen games for this thesis. As a result I calculated that if timing other games' moves was going to result in the misrepresentation of rhythm in chess, then making *The Sound of Chess Programme* play the sounds at random would be on par with the condensed time outcome. As a result the rhythmic dimension is missing in *The Sound of Chess Programme*, as it left out the musical reality by performing it *ad lib* and the sounds are set at random within *The Sound of Chess Programme*, so that it could be re-evaluated and developed in future.

Ways in which to write code for check and checkmate were also considered. The most plausible option was to make the surrounding squares where the King cannot go, due to attack and then sound at the same time as the last move made. For example, if a Queen checked the King, the move that the Queen made would sound along with each square surrounding the King. This would enable the listener to hear when the King was under attack. Although this was a valid idea, it could not be logically explained in accordance with chess representation. I thus decided that this is an investigation to be explored further at a later stage.

There are no prescriptions that dictate how to apply representations to subject matter. One can only make decisions based on logical aspects which are relevant at the time, such as in the development stages of Gregorian chant, society ultimately demanded the changes needed for the representations of their time. Although notation needs to represent the material at hand, it does not need to explain notational systems.

The final three sounding chess games reflect much about the character of their individual positional qualities. The results with regards to the sounds in accordance with the actual happenings of the positions in each game are as follows:

Scholars Mate is a short and basic game consisting of roughly four moves for each black and white, which constitutes a bishop's opening. The results of the sounds reflect the simplicity of the first four opening moves: both white's and black's e file Pawns followed by both white's and black's Bishops moving to the c file; all four of these moves can be heard as their sounds are the same pitches, which are octaves apart. The white Queen travels from d1 to a5 which causes the pitches to move four octaves higher and sounds a compound perfect 5th up. The following move, Knight to f6, is a mistake for black. Instead black should attempt to stop the checkmate that the white Queen is threatening on f7. One can strangely hear that this Knight's move protrudes quite prominently within the succession of all the sounds, almost as if the listener can detect danger within the sounds.

Anant Dole's game consists of an opening called the Queen's Gambit Declined, Slav Defence Dutch Variation (D18). This opening is very advanced although not all chess players enjoy the d5 opening. The opening requires the player to bring out all their pieces and ready the battlefield first with strong possibilities for both Kings' and Queens' side attacks. The opening consists of an elegant succession of prerequisite moves. In this particular game after the first few moves, Anant delves into a very creative middle game attacking both whites' Kings and the Queen's side, until he breaks through on the King's side and ends the game with his opponent resigning one move prior to being checkmated. At the beginning of the sounds one can hear all the pieces being developed and the pre-determined positional structure can be heard in the sounds almost as if it were intentionally composed in this melodic structure. After white castles with a chord, which is easy to identify within the sounds, the game becomes more creative. One can hear sounds that are not as connected as in the beginning of the game, almost as if the answers between the black and white pieces are gaining new momentum. Anant's positional attacks are audible on the Queen's side followed by attacks in sound on the King's side. One can also hear when a piece is being taken as there is a semitone lift. In bar 20 a piece has been captured and one can hear an interval of an octave, but as Selfridge-Field explained when a MIDI key number is played, there is no way of knowing the difference between certain intervals (Selfridge-Field 1997 p. 6). If middle C is coded as 60, 60 could also mean that B sharp or D double flat are sounding. In the sounds of the last six chords one can delineate to some extent that Anant saw the checkmate ahead of time and played out the final moves accordingly. One can hear the King being boxed in as the variation of the sounds becomes smaller and tighter. The piece even ends with a perfect cadence.

The opening played in the last game, by Kasparov and Deep Blue, is the Sicilian Defence, Alapin Variation, Barmen Defence Modern Line (B22). Deep Blue started the game with a brute force approach which is used by computers. Kasparov decided to respond with the Sicilian opening of c5. Unlike with Scholars Mate and in Anant's game one can hear the difference in the opening compared to when the Pawns answer on the same files; this opening used different files in the opening. Instead of e4 and e5 or d4 and d5 one can hear the e4 followed by c4. The fifth move after the white Knight develops to f3. Black pins the Knight with its Bishop; one can hear this via the pitches moving up a whole tone. Many times in this game there are swap offs, which are also audible such as in moves three, nine, ten, 23 and 24 the pitches stay the same. One can also hear Kasparov trying to re-route his Bishop onto the white King's c7 diagonal. White stops this by placing a Knight on b5 and as a result one can hear the Bishop's sounds being interrupted by the Knight's sound. The end of the game is also very interesting as similarly to Anant's game the sounds of the pitches start to move towards a final cadence. Unfortunately Kasparov resigned, but one can hear that white's sounds are closing in on Kasparov's King. It is also possible to hear the discovered check in the last two moves. The black King moves in the line of a discovered check after which Deep Blue removes his Knight which is standing in between the white Rook and the Kasparov's King, and exposes the check which Rook is threatening. This is audible by a semitone playing from an F sharp to a G whilst the D in the bass stays sounding in both chords; this is similar to something being unwrapped or in this case, by moving the Knight the check is unveiled within the sounds.

There are many more sounds that come to the fore in the three chess games, for instance: which pieces are taking one another during exchanges, which colours are castling on which side of the board, cadences that can be heard for example in Kasparov's game, when Deep Blue moves his Rook out of danger to c7. As a result it is possible to hear certain developments with regards to unique positional qualities that each of the three chess games harness, within the final sounds.

I acknowledge that there are many more aspects regarding the music analysis of these three sounding chess games, but this will require further exploration in future research.

5.3 Outlook

If one looks at the music notation in Appendix C one can recognise a pattern visually as well as aurally in all three pieces. The pieces all start with the alternating chords sounding a large distance apart: this happens because white and black are on opposite sides of the board. At the beginning of the games, when white plays, one hears and sees low sounding chords and when black answers one hears and sees higher sounding chords. As the pieces progress one can see that the chords played by white and the chords played by black sound closer together. Eventually these chords start to integrate or overlap. Pattern recognition is vital to become proficient at both chess and music study.
The variables for the chess moves require many more alterations in order to be better synthesised with music. If the sounding chess games could sound more musical it would better show off the beauty of the game of chess.

The quality of the sounds needs to be further explored and addressed as it will make it easier for the listener to interact with the sounds. Adding dynamics to the sounds will also add a sense for the positional structure of the game as well as add depth to the sounds. Most importantly the implementation of musical rhythm could add another dimension to the chess sounds, as sustained notes and quick rhythms could make certain positional tensions more obvious to the listener and could ultimately enhance the assigned pitches.

Other possible future side adaptations which could make the chess sounds more musical within *The Sound of Chess Programme* include: firstly, integrating the programme of sounding chess with an online analysis chess board could aid chess players and composers with the opportunity to create their own sounding chess games. Secondly, further explorations for variables that represent check, checkmate, winner detection and perhaps an interactive rhythmic element which allows a player to implement the next chord via pressing Enter. Lastly, finding ways in which to improve the tonal quality of the sounds within the programming could make the games sound both more interesting and musical.

A few other possibilities which could make for interesting research regarding what a sounding chess game could contribute to music study include: an investigation into re-interpreting the variables for this study in order to develop a sounding chess board which suits the needs of music teaching or study. This could be used as a teaching tool that is interactive, three dimensional, music representational, and could ultimately integrate visual and auditory strengths of learning, combined into one entity. This could also teach beginners music more swiftly by adding the elements of interaction and three dimensions, via the use of chess pieces on a chess board. As a child moves the chess pieces up and down a board, music intervals, pleasing to the ear and compatible with Standard Western Music Notation, albeit in a different format, are sounding. Representation of music on a chess board could make notation easier to understand and faster to read and write. Music notation and music theory could also be simplified for musicians who struggle to read Standard Western Music Notation, via a sounding chess board.

5.4 Conclusion

Chess and music are compatible via logical linkage. This was validated by making a game of chess sound via computer programming. The reason for undertaking this was to explore the representations of Standard Western Music Notation in an analytical and playful experiment, via the substitution of both music and chess representations in order to delineate their metaphysical status.

Initially when I observed that my aural skills were improving after I attended chess coaching, I believed it to be coincidental, but after further exploring the relationship between chess and music, it is clear that it was the correlations between the entities that helped me to learn faster.

Master chess players and advanced musicians use pattern recognition in order to decipher their music and positions on a chess board, but pattern recognitions could intensify when participating in sounding chess. Western Classical Music, being time bound, does not allow for reflection, but needs immediate visual pattern recognition before any action is taken. Chess players are coping with complicated pattern recognition, but have time to reflect unless it is during blitz chess where there are strict time constraints. Sight and hearing are linked in the brain and both chess and music study stimulate optimum brain function. This means that it is highly likely that the integration of chess and music could stimulate brain activity to function at an even higher level.

The correlations between chess and music have intensified for me as a result of this thesis, as one can now touch a musical note by picking it up in the form of a chess piece, after which one can alter its identity by moving it across a chess grid in three dimensions. This uncovered many new aspects regarding the integration of chess and music, as by blueprinting the two entities, they revealed one another's weaknesses and enhanced one another's strengths. It was immensely exciting for me as this is a link which I have longed to experience within music and chess.

The outcome of *The Sound of Chess Programme* lacked certain entities, but the co-existence of the infinite variety of chess moves and the infinite variations of sounds demonstrated that there is no shortage of multiplicand patterns in both chess and music, which in turn makes it simple to superimpose the two entities in different ways. This makes it apparent that the connections between chess and music are endless and that they are clearly not as superficial as black and white chess pieces, chequered chess board squares and black and white piano keys. Instead the two entities are interconnected and the integration of their correlations could benefit different aspects in both chess and music fields. My thesis demonstrates that there is a connection between chess and music. It is evident that what is played makes musical sense, but how it plays is to be further explored in future research.

5.5 Recommendations

The representations of chess and music have many links, which will suffice for various methods in which the two entities could be affiliated, regarding differently set outcomes.

Throughout history music has been represented in various ways. The notational possibilities for music are endless. For this project chess and music connected via logic, but these entities can be linked with anything. The possibilities are much like the variations regarding the choices of openings to play in a game of chess. There are also many ways in which to develop strategies. The varieties of openings are often frightening for a beginner, but because chess is logical, it is possible to develop strategies in order to cope with the numbers of choices with which one is faced. Other ways in which music and chess can be linked include via their creativity, intuitional phenomenon, terminologies and concentration.

The main limitations of *The Sound of Chess Programme* are as follows: firstly, that the monotonous tones make it difficult to engage with the pitches; making the quality of the sounds better within the computer programme could make listening to the games more enjoyable. Secondly, the absence of rhythmic structures within *The Sound of Chess Programme* causes problems when trying to delineate which section of a game I am listening to. Lastly, it is likely that it would have been more satisfactory to have had beautiful musical qualities that show off the positional tensions and excitements. Although some of these variables are lacking in my project, I am convinced that future developments could account for all the variables being integrated and implemented to better reflect the characteristics of chess games within musical sounds, whilst still remaining true to the logical translations of the representations of chess entities.

The process of the ways in which I implemented the chess moves to musical variables taught me that exchanging sound entities with various representational systems could add new dimensions to the interactions and understanding of different methodical systems. For example, if one could comprehend a logical system such as the Table of Elements by hearing it, then this would result in an alternative way in which to study exclusive systems. The merging of chess and music could be seen as an example of the possibilities regarding other rational systems being made to sound. If it is possible that any logical system can be represented by musical entities and vice versa, a 21st century composer is spoiled for choice. The game of chess has interested and stimulated many famous composers from previous eras, which makes it possible that the integration of chess and music is also likely to fascinate future composers. This could mean that the two entities which have co-existed since the Stone Age may exist as one entity in the future.

By linking scientific concepts with musical phenomena via their representations, many fields of study could be integrated. Most importantly if chess and music were compatible and resulted in a cohesive outcome for my thesis, then the following quotation from Leonardo da Vinci (Chaucer 2012) reiterates the importance of such connections:

To develop a complete mind: Study the science of art; and learn the art of science. Realize that everything connects to everything else.

Appendices

Appendix A: Intervals of each chess piece in isolation

The following sounds are available on the accompanying disc.

Random Knight moves:

Nb8-Nc6	107-84
Ng8-Nf6	115-89
Ng1-Nf3	31-53
Nb1-Nc3	23-48
Nd6-Ne4	86-64
Ng4-Ne3	67-52

Random Bishop moves:

Bc1-Bg5	24-79
Bf1-Bd3	29-50
Bf8-B37	113-100
Bc8-Ba3	108-57
Be3-Bh6	52-93
Bb3-Bd5	47-74

Random Rook moves:

Ra1-Rd1	21-26
Ra1-Ra4	33-69
Rh8-Rd8	117-110
Ra8-Ra3	105-45
Rd1-Rd5	26-74
Rb6-Rh6	83-93

Random Queen moves:

Qd8-Qf6	110-89
Qd8-Qd5	110-74
Qd1-Qb3	26-47

Qd1-Qd4	26-62
Qg3-Qc2	55-36
Qa7-Qg1	93-31

Random King moves:

Ke8-Kf8	112-113
Ke1-Ke2	28-40
Kd4-Ke4	62-64
Kf3-Ke2	53-40
Kb7-Kc6	95-84
Kd6-Ke5	86-76

Appendix B: Chess and pitch notation of all three chess games

Included in Addendum D are Scholars Mate, Anant Dole's game against a Grandmaster at the Parsvnath International open in 2008 and Garry Kasparov's first match against Deep Blue in 1996.

The pitching codes are indicated in the column on the right of every move played. Each move contains two sounds. A particular chess move which is referred to as castling contains four notes as two pieces (The King and the Rook) which are being played as one move. Please note that "(S)" means sharp and as a result the pitch adds one digit to its coding value (semitone); this happens when a piece is taken off the board.

This game is the beginner chess game used for this study:

White: Unknown		**Black:** Unknown	
e4	40-64	**e5**	100-76
Bc4	29-60	**Bc5**	113-72
Qh5	26-81	**Nf6**	115-89
Qxf7++	81-102 (S)		
1	White Won	**0**	Black Lost

This game is the intermediate level chess game used for this study:

Event: 6th Parsvnath International Open 2008 **White:** Batyrov. S (GM: Grandmaster)		**Date:** 19/01/2008 **Round:** 9 **Result:** 0-1 **Black:** Dole. A	
d4	38-62	**d5**	98-74
Nf3	31-53	**Nf6**	115-89
c4	36-60	**c6**	96-84
Nc3	23-48	**dxc4**	74-65 (S)
a4	33-57	**Bf5**	108-77
e3	40-52	**e6**	100-88
Bxc4	29-61 (S)	**Bb4**	113-59

0-0	28-31 + 33-29	**Nb-d7**	107-98
Nh4	53-69	**Bg6**	77-91
f4	41-65	**Nd5**	89-74
Nxg6	69-92 (S)	**hxg6**	105-92 (S)
Bd2	24-38	**N7-b6**	98-83
Bb3	60-47	**a5**	93-69
Rc1	21-24	**Nf6**	74-89
Qe2	26-40	**Qe7**	110-100
e4	52-64	**Rd8**	105-110
Bc2	47-36	**Rxd4**	110-63 (S)
Be3	38-52	**Bc5**	59-72
Bxd4	52-63 (S)	**Bxd4+**	72-63 (S)
Kh1	31-33	**Nxe4**	89-65 (S)
Rf3	29-53	**Qh4**	100-69
g4	43-67	**Nf2+**	64-41
Rxf2	53-42 (S)	**Bxf2++**	62-42 (S)
0	White Lost	**1**	Black Won

This game is the advanced chess game used for this study:

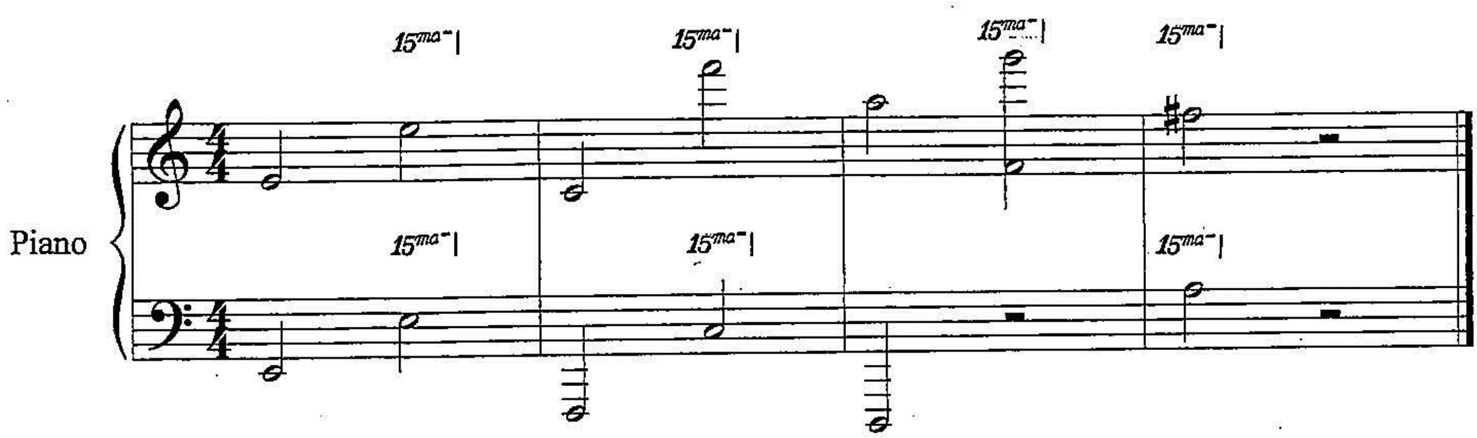
Event: Man versus Computer in Philadelphia, Pennsylvania **White:** Deep Blue (IBM Chess Computer)		**Date:** 10/02/1996 **Round:** 1 **Result:** 1-0 **Black:** Kasparov. G (GM: Grandmaster)	
e4	40-64	**c5**	96-72
c3	36-48	**d5**	98-74
exd5	64-75 (S)	**Qxd5**	110-75 (S)
d4	38-62	**Nf6**	115-89
Nf3	31-53	**Bg4**	108-67
Be2	29-40	**e6**	100-88
h3	45-57	**Bh5**	67-81
0-0	28-31 + 33-29	**Nc6**	107-84
Be3	24-52	**cxd4**	72- 63 (S)

cxd4	48-63 (S)	**Bb4**	113-59
a3	33-45	**Ba5**	59-69
Nc3	23-48	**Qd6**	74-86
Nb5	48-71	**Qe7**	86-100
Ne5	53-76	**Bxe2**	81-41 (S)
Qxe2	26-39 (S)	**0-0**	112-115 +117-113
Ra-c1	21-24	**Ra-c8**	105-108
Bb5	40-71	**Bb6**	69-83
Bxf6	79-90 (S)	**gxf6**	103-90 (S)
Nc4	76-60	**Rfd8**	113-110
Nxb6	60-84 (S)	**axb6**	93-84 (S)
Rf-d1	29-26	**f5**	89-77
Qe3	40-52	**Qf6**	100-89
d5	62-74	**Rxd5**	110-75 (S)
Rxd5	26-75 (S)	**exd5**	88-75 (S)
b3	35-47	**Kh8**	115-117
Qxb6	52-84 (S)	**Rg8**	108-115
Qc5	83-72	**d4**	74-62
Nd6	71-86	**f4**	77-65
Nxb7	86-96 (S)	**Ne5**	84-76
Qd5	72-74	**f3**	65-53
g3	43-55	**Nd3**	76-50
Rc7	24-96	**Re8**	115-112
Nd6	95-86	**Re1+**	112-28
Kh2	31-45	**Nxf2**	50-42 (S)
Nxf7+	86-102 (S)	**Kg7**	117-103
Ng5	101-79	**Kh6**	103-93
Rxh7+	96-106 (S)	**Resigns**	
1	White Won	**0**	Black Lost

Appendix C: The sheet music for *The Sound of Chess* games

Please be aware that some of the chess sounds are out of range for the piano. The piano notation is only used for a visual idea of what one is hearing when listening to *The Sound of Chess Programme*. *The Sound of Chess Programme* is not designed to the pitch parameters of any specific musical instrument.

Scholars_Mate



A_Dole vs S_Batyrov

G_Kasparov vs Deep_Blue


Pno.
15ma
Pno.
15ma
15ma
15ma
15ma
15ma
15ma

Appendix D: Instructions to listen to *The Sound of Chess Programme* on the accompanying disc

1. The accompanying disc consists of one file named Sound of Chess. Click on the file and you will see three folders: Programme, Code and Requirements.
2. Open the Requirements folder and instal jGRASP.
3. Once it is installed, run the programme then click on the icon of the red man running on the menu bar of the user interface.
4. A message welcoming you to *The Sound of Chess* will pop up, click ok.
5. Thereafter a three step process will appear on the screen.
6. Press 1 followed by Enter. The programme will then ask you which game you would like to run.
7. Thereafter copy the name of the chess game that you would like to listen to from your programme folder and paste it into the slot on the window.
8. Press 2 followed by Enter. The game will now load in the user interface.
9. Press 3 followed by Enter and listen to *The Sound of Chess Programme*.
10. When you are finished press 10 followed by enter to load another chess game or to exit the programme.

Bibliography

Alloway. T.P, Alloway. R.G. (2013). *Working Memory: The Connected Intelligence.* Have: Psychology Press.

Ancient Music Ireland> (2015). *Instruments Through the Ages.* <www.ancientmusicireland.com> Accessed 10/07/2015.

Anonymous. (1813). *An Easy Introduction to the Game of Chess*. London: Reese and Brown.

Arnold. D. (1994). *The New Oxford Companion to Music*. London: Oxford University Press.

Assayag. G. (2013). *Mathematics and Music: A Diderot Mathematical Forum*. Berlin: Springer.

Assiya. U. (2013). *Chess and Maths* Saarbrücken: Lambert Academic Publishing.

Bader. R. (2009). *Musical Acoustics, Neurocognition and Psychology of Music.* Frankfurt: Peter Lang.

Banaschak. P. (2000). *The Eastward Diffusion of Chess: Why this History Cannot Yet be Written*. Lawndale: Meissenburg.

Beard. M. (2012). *The Olympics – Ancient and Modern.* <www.theguardian.com> Accessed 02/07/2015.

Bell. J, Davis. HP. (2011). *Developing Rhythmic Sensitivity: A Study Designed for All Musicians.* Bloomington: Trafford Publishing.

Bell. R.C. (1979). *Discovering Chess.* London: Shire Publications.

Bird. H.E. (2010). *Chess History and Reminiscences.* Whitefish: Kessinger.

Bird. S, Klein. E, Loper. E. (2009). *Natural Language Processing with Python.* Sebastopol: O'Reilly.

Boden McGrill. C.J, King. K.P. (2013). *Developing and Sustaining Adult Learners.* Charlotte: IAP Publishing.

Boom. M, Rychtar. J. (2013). *Game-Theoretical Models in Biology.* Boca Raton: CRC Press.

Boyarsky. J, Selikoff. S. (2014). *OCA: Oracle Certificate Associate Java SE8 Programmer.* Hoboken: John Wiley and Sons.

Bronstein. D. (2013). *Zurich International Chess Tournament 1953.* North Chelmsford: Courier Corporation.

Brown. L.L. (2013). *The Benefits of Music Education*. <www.pbs.org> Accessed 10/10/2013.

Bryant. J. (2011). *Java 7 for Absolute Beginners.* New York: Apress.

Bulliet. R, Crossley. P. (2011). *The Earth and Its Peoples, Brief Edition, Volume 1.* Boston: Cengage Learning.

Burgun. K. (2012). *Game Design Theory: A New Philosophy for Understanding Games.* Boca Raton: CRC Press.

Cage. J, Duchamp. M. (2009). *Reunion: John Cage, Marcel Duchamp, Electronic Music, and Chess*. <www.johncage.org> Accessed 02/02/2015.

Cairns. G. (2002). Pillow Chess: *Mathematics Magazine.* Vol. 75, No. 3, pp. 173-186.

Celone. J. (2001). *Educational Technologies.* <www.edutechchess.com> Accessed 04/02/2014.

Chaucer H. (2012). *A Creative Approach to the Common Core Standards: Da Vinci Curriculum*. Lanham: Rowman & Littlefield Education.

Chernev. I. (1998). *Logical Chess: Move by Move*. London: Batsford.

ChessBase. (2011). *Ludwig 3.0 – Write Great Songs with Automatic Arrangements.* <www.write-music.com> Accessed 05/07/2015.

Chess-insights. (2015). *Improve Your Thinking Process in Chess.* <www.chess-insights.com> Accessed 09/07/2015.

Chung. P, Hinde. C, Moonis. A. (2003). *Developments in Applied Artificial Intelligence.* Berlin: Springer.

Computer History Museum. (2011a). *Mastering the Game: A History of Computer Chess.* <www.computerhistory.org> Accessed 03/07/2015.

Computer History Museum. (2011b). *Opening Moves: Origins of Computer Chess.* <www.computerhistory.org> Accessed 03/07/2015.

Computer History Museum. (2011c). *Middle Game: Computer Chess comes of Age.* <www.computerhistory.org> Accessed 04/07/2015.

Computer History Museum. (2011d). *Chess for Everyone.* <www.computerhistory.org> Accessed 04/07/2015.

Computer History Museum. (2011e). *Endgame.* <www.computerhistory.org> Accessed 04/07/2015.

Cope. D. (2004). *Virtual Music: Computer Synthesis of Musical Style.* Cambridge: MIT Press.

Corbett. P. (2014). *The absolute basics: Python 3.* New York: Phil Corbett.

Cottrell. W.C. (1984). *The choral music of Paul Patterson.* Champaign: Illinois University Press.

Črnčen. R, Wilson. S.J. (2006). The Cognitive and Academic Benefits of Music to Children: Facts and Fiction. *Educational Psychology.* Vol. 26, No. 4, pp. 579-594.

Czelakowski. J. (2015). *Freedom and Enforcement in Action: A Study in Formal Action Theory.* Berlin: Springer.

Dale. N, Weems. C. (2008). *Programming and Problem Solving with Java.* Burlington: Johnes & Bartlett Learning.

Daverio. J. (2002). *Crossing Paths: Schubert, Schumann, and Brahms*. Boston: Boston University Press.

Davidson. H.A. (2012). *A Short History of Chess.* Philadelphia: Crown/Archetype Publications.

de Bruin. J.P.C, Uylings. H.B.M, van Eden. C.G, Corner. M.A, Feenstra. M.G.P. (1991). *The Prefrontal Cortex: Its Structure, Function and Pathology.* Amsterdam: Elseviers Science Publishers.

de Groot. A. (2008). *A Thought of Choice in Chess.* Amsterdam: Amsterdam University Press.

Delaere. M, Crispin. D. (2009). *Unfolding Time: Studies in Temporality in Twentieth-Century Music.* Leuven: Leuven University Press.

DeMaio. J. (2011). Which Chess Boards have a closed Knight's Tour within the Rectangular Prism? *The Electronic Journal of Combinatorics*. Vol. 18, pp. 1-14.

Di Felice. G. (2010). *Chess Periodicals: An Annotated International Bibliography 1836-2008.* Jefferson: McFarland & Company.

Doke. E.R. (1999). *Java for the COBOL Programmer*. London: Cambridge University Press.

Downs. P. (2013). *Schoolhouse Shams: Myths and Misinformation in School Reform.* Lanham: Rowman & Littlefield.

Du Sautoy. M. (2011). *Listen by Numbers: Music and Maths*. <www.thegaurdian.com> Accessed 08/05/2013.

Ferg. S. (2009). *Python Conquers the Universe: Python and Java: A side-by-side Comparison.* <www.pythonconquerstheuniverse.wordpress.com> Accessed 03/10/2015.

FIDE. (2015). *Regulations for the 2014 -2015 FIDE World Chess Grand Prix Series.* <www.fide.com> Accessed 01/07/2015.

Fischer. J. (2015). *Ein Gespräch mit Matthias Wüllenweber über Schach, Computer, Mathematik und Musik.* <www.Karlonline.org> Accessed 06/07/2015.

Flanagan. D. (2005). *Java in a Nutshell.* Sebastopol: O'Reilly Media.

Forbes. D. (1860). *The History of Chess: From the time of the early invention of the game in India till the period of its establishment in western and central Europe.* London: W.H. Allen and Company.

Frantz. A. (2015). *Ear Training: Is Your Musical Brain Wired Backwards.* <www.key-notes.com> Accessed 05/06/2015.

Friedel. F. (2006). *Ludwig – a synthesis of chess and music*. <www.chessbase.com> Accessed 14/09/2013.

Friedel. F. (2012). *Ask the Experts: with Matthias Wüllenweber and Mathias Feist*. <en.chessbase.com> Accessed 05/07/2015.

Friedel. F. (2015). *Kasparov and Thirty Years of Computer Chess.* <en.chessbase.com> Accessed 05/07/2015.

Garrido. J.V. (2009). *Object Oriented Simulation: A Modelling and Programming Perspective.* New York: Springer.

George. S.A. (2005). *Visual Perception of Music Notation: On Line and Off Line Recognition.* London: IRM group.

Ghannam. T. (2011). *The Mystery of Numbers: Revealed Through their Digital Root.* New York: Create Space Independent Publishing.

Gier. N.F. (1981). *Wittgenstein and Phenomenology: A comparative Study of the Later Wittgenstein, Husserl, Heidegger, and Merleau-Ponty.* New York: Suny Press.

Gladwell. M. (2008). *Outliers: The Story of Success.* New York: Little, Brown and Company.

Gobet. F, Retschitzki. J. (2004). *The Psychology of Board Games.* Hove: Psychology Press.

Godlovitch. S. (2002). *Musical Performance: A Philosophical Study*. New York: Routledge.

Goeth. P. (2015). *JS Bach's Well Tempered Clavier: Notable Performances.* <www.bachwelltemperedclavier.org> Accessed 09/07/2015.

Golladay. S.M. (2007). *Los Libros de Acedrex Dados E Tablas: Historical, Artistic and Metaphysical Dimensions of Alfonso's X's "Book of Games".* Ann Arbor: ProQuest.

Golombek. H. (1976). *A History of Chess.* London: Routledge and Kegan Paul.

Green. L. (2013). *How Popular Musicians Learn: A Way Ahead for Music Education.* Farnham: Ashgate Publishing.

Griggs. J. (2013). *The Secret Link between Chess and Playing Music by Ear*. <www.hearandplay.com> Accessed 12/12/2014.

Haire. M. (2015). *Should Chess be an Olympic Sport?* <www.content.time.com> Accessed 01/07/2015.

Hale. B. (2013). *Philosophy looks at Chess.* Chicago: Open Court Publishing.

Hallman. J.C. (2013). *The Chess Artist: Genius, Obsession, and the World's Oldest Game.* London: Macmillan.

Harkleroad. L. (2009). *The Math Behind the Music*. New York: Cambridge University Press.

Harley. J. (2004). *Xenakis: His Life in Music.* New York: Routledge.

Harmon. C. (2002). *A Peoples History of the World: From the Stone age to the New Millennium*. London: Verso Books.

Hartmann. J. (2014). *Choosing a Chess Engine.* <en.chessbase.com> Accessed 04/07/2015.

Hartmut Metz. F.M. (2000). *Diktator Kasperov verhalf zu mehr Demokratie: Matthias Wüllenweber über die Computerisierung des Schachs in der Ära des 13. Weltmeisters.* <www.scrkuppenheim.de> Accessed 06/07/2015.

Hartston. W.R, Rice. T, Andersson. B. (1986). *Chess: The Making of a Musical.* Hove: Pavilion.

Hearst. E; Knott. J. (2009). *Blindfold Chess: History Psychology, Techniques, Championships, World Records and Important Games.* Jefferson: McFarland.

Hooper. D.V. (1992). *The Oxford Companion to Chess*. London: Oxford University Press.

Hooper. D.V; Whyld. K. (1996). *The Oxford Companion to Chess.* London: Oxford University Press.

Horowitz. I.A; Rothenberg. P.L. (2015). *Personality of Chess:* New York: Ishi Press.

Horstmann. C.S. (2001). *Core Java 2: Fundamentals.* Upper Saddle River: Prentice Hall Professional.

Hsu. F.H. (2002). *Deep Blue Building the Computer that Defeated the World Chess Champion.* Princeton: Princeton University Press.

Isacoff. S. (2012). *The Instrument, The Music, The Musicians from Mozart to Modern Jazz and Everything in Between*. London: Souvenir.

Jaap van den Herik. J. (2008). *Computers and Games: 6th International Conference.* New York: Springer.

Jaiswal. S, Kumar. R. (2015). *Learning Django Web Development.* Birmingham: Packt.

JGRASP. (2015). *An integrated Environment with Visualizations for Improving Software Comprehensibility.* <www.jgrasp.org> Accessed 03/10/2015.

Johnson. J. (2002). *Who Needs Classical Music?* London: Oxford University Press.

Kamien. R. (2010). *Music: An Appreciation.* New York: McGraw-Hill.

Kaschub. M, Smith. J. (2014). *Promising Practices in 21st Century Music Teacher Education.* London: Oxford University Press.

Kasparov. G. (2003). *My Great Predecessors: Part 1.* New York: Random House.

Kasparov. G. (2010). *The Chess Master and the Computer.* <www.nybook.com> Accessed 09/07/2015.

Kaufman. J.C, Sternberg. R.J. (2006). *The International Handbook of Creativity.* London: Cambridge University Press.

Kaufman. S.B. (2013). *The Complexity of Greatness: Beyond Talent or Practice.* London: Oxford University Press.

Klickstein. G. (2009). *The Musician's Way*: Oxford: Oxford University Press.

Kniep. A. (2006). *Schach: Der Faktor Mensch.* <www.spiegel.de> Accessed 06/07/2015.

Krabbenbos. J. (2015). *WCCC 2015.* <www.icga.org> Accessed 04/07/2015.

Kuijken. B. (2013). *The Notation is Not the Music: Reflections on Early Music Practice and Performance.* Bloomington: Indiana University Press.

Lasar. M. (2011). *Brute Force or Intelligence? The Slow Rise of Computer Chess: Computer Chess isn't just about Playing Games.* <www.arstechnica.com> Accessed 03/07/2015).

Latham. A. (2003). *The New Oxford Companion to Music*. London: Oxford University Press.

Latour. B, Woolgar. S. (2013). *Laboratory Life: The Construction of Scientific Facts*. Princeton: Princeton University Press.

Lawrence. A, Alburt. L. (2010). *Knack Chess for Everyone: A Step-by-Step Guide to Rules , Moves & Winning Strategies.* Lanham: Rowman & Littlefield.

Lawson. C. (2013). "The Notation is Not the Music: Reflections on Early Music Practice and Performance." By Barthold Kuijken: *Performers Practice Review.* Vol. 18, No. 1, Article. 8, pp. 1-4.

Levin. F.R. (2009). *Greek Reflections on the Nature of Music.* London: Cambridge University Press.

Levitt. G. (2006). *The Turk, Chess Automaton.* Jefferson: McFarland.

Levy. D, Newborn. M. (2009). *How Computers Play Chess.* New York: Ishi Press.

Levy. D, Newborn. M. (2012). *All about Chess and Computers: Chess and Computers and more Chess and Computers.* New York: Springer.

Levy. D. (2013). *Computer Chess Compendium*. New York: Springer.

Lin. B. (2012). *Constraint and Craft in Schoenberg's Coalition Chess.* <www.bobtailyearlings.com> Accessed 16/07/2015.

Linder. I. (2013). *Jose Raul Capablanca: Third World Chess Champion (Chesscafe World Chess Champion Series) (The World Chess Champion Series).* Milford: SCD Distributors.

Litvin. M, Litvin. G. (2015). *Getting Started with jGRASP for Java.* Andover: Skylight.

Lochhead. J. (2015). *Reconceiving Structure in Contemporary Music: New Tools in Music Theory Analysis.* London: Routledge.

López. J. (2007). *Critical Information Infrastructures Security: First International Workshop.* Berlin: Springer.

Luhring Augustine (2010). *Guido van der Werve- Nummer Twaalf.* <www.luhringaugustine.com> Accessed 23/01/2015.

Luque. S. (2009). The Stochastic Synthesis of Iannis Xenakis: *Leonardo Music Journal*. Vol. 19, pp. 77-84.

Lutz. M. (2010). *Programming Python.* Sebastopol: O'Reilly.

Macdonald. H. (2012). *Music in 1853: The Bibliography of a year.* Woodbridge: The Boydell Press.

Marshland. T.A, Schaeffer. J. (2012). *Computers, Chess, and Cognition.* New York: Springer.

Martelli. A. (2006). *Python in a Nutshell.* Sebastopol: O'Reilly.

Maurer. J.A. (1999). *A Brief History of Algorithmic Composition*. <ccrma.stanford.edu> Accessed 09/10/2013.

McClain. D.L. (2010). *Mathematicians, Musicians and Chess Masters*. <www.nytimes.com> Accessed 08/11/2013.

McClain. D.L. (2011). *Making the Case for or Against, Chess as an Olympic Sport.* <www.nytimes.com> Accessed 02/07/2015.

McCrary. R.J. (2015). *The History of Notation.* <www.Knightschessclub.org> Accessed 02/07/2015.

McDonald. N. (2013). *Chess: The Art of Logical Thinking: From the first move to the last.* London: Batsford.

Metzinger. J. (2015). *The Dawn of Western Christianity.* <www.liturgica.com> Accessed 18/07/2015.

Michel. L. (2012). *The Grandmaster Hoax.* <www.theparisreview.org> Accessed 03/07/2015.

Mikoluk. K. (2013). *Python vs Java: Key Differences.* <www.blog.udemy.com> Accessed 4/10/2015.

MonRoi. (2009). *Celebrities Play Chess.* <www.monroi.com> Accessed 12/07/2015.

Montagu. J. (2004). How Old is Music?: *The Galpin Society Journal.* Vol. 57, pp. 171-182.

Mouton. J. (2004). *How to succeed in your Master's and Doctoral Studies.* (7thEd). Pretoria: Van Schaik.

Murray. H.J.R. (2015). *A History of Chess: The Original 1913 Edition*. New York: Skyhorse Publishing.

Nelson. R. (2014). *Foundations of Music*. Boston: Cengage Learning.

New Word Encyclopedia. (2014). *Gregorian Chant.* <www.newworldencyclopedia.org> Accessed 17/07/2015.

Newborn. M. (2011). *Beyond Deep Blue: Chess in the Stratosphere.* New York: Springer.

Newborn. M. (2012). *Kasparov versus Deep Blue: Computer Chess comes of Age*. New York: Springer.

Newborn. M. (2014). *Computer Chess ACM Monograph Series.* Cambridge: Academic Press.

Newsome. B. (2012). *Beginning Visual Basics 2012.* New Jersey: Wiley.

Nierhaus. G. (2009). *Algorithmic Composition: Paradigms of Automated Music Generation*. New York: Springer.

Oracle. (2015). *The History of Java Technology.* <www.oracle.com> Accessed 28/08/2015.

Paquette. M. (2007). *Online Schoenberg Coalition Chess.* <www.schoenbergchess.com> Accessed 16/07/2015.

Patterson. J. (2015). *History of Music Notation*. <www.mfiles.co.uk/music-notation-history.htm> Accessed 14/07/2015.

Paul. A.L. (1936). The Origin of Chess: *Western Chess Chronicle.* Vol. 1, No. 9.p. 1.

Paul. E.S, Kaufman. S.B. (2014). *The Philosophy of Creativity: New Essays.* New York: Oxford University Press.

Peddie. J. (2013). *The History of Visual Magic in Computers.* New York: Springer.

Pein. M. (2015). *Chess at the Olympics?* <www.telegraph.co.uk> Accessed 02/07/2015.

Peplow. M. (2004). *Science Secret of Grand Masters Revealed.* <www.nature.com> Accessed 05/08/2014.

Petrović. M. (1997). *Mathematics and Chess*. New York: Dover Publications.

Philidor. F.A. (1777). *Analysis of the Game of Chess.* 2nd ed. London: P. Elmsley.

Phillips. D. (2010). *Algorithmic Music Composition with Linux Part 2: Linux Journal.* <www.linuxjournal.com> Accessed 16/02/2013.

Pilgrim. M. (2009). *Dive into Python 3.* New York: Springer.

Pitt. D. (2012). *Stanford Encyclopaedia of Philosophy: Mental Representation*. <plato.stanford.edu> Accessed 04/06/2015.

Poe. E.A. (2015). *Maelzel's Chess-Player.* Brighton: Arsalan Ahmed.

Pritchard. D. (2012). *The Right Way to Play Chess.* London: Hachette.

Prokofiev. S, Phillips. A. (2008). *Sergey Prokofiev diaries, 1915-1923: Behind the Mask.* New York: Cornell University Press.

PSF. (2015). *Floating Point Arithmetic: Issues and Limitations.* <www.docs.python.org> Accessed 05/10/2015.

Python. (2015). *Python Course.*<www.python-course.eu> Accessed 03/10/2015.

Quine. V.W. (2009). *The Philosophy of Logic. 2nd Edition.* Cambridge: Harvard University Press.

Rastall. R. (2008).*The Notation of Western Music: An Introduction.* London: Travis and Emery.

Reilly. E.D. (2003). Milestone in Computer Science and Information Technology. West Port: Greenwood Publishing.

Remus. H. (2015). *The Origin of Chess and the Silk Road***.** <www.silkroadfoundation.org> Accessed 20/08/2015.

Renold. M. (2015). *Intervals, Scales, Tones.* Forest Row: Temple Lodge Publishing.

Rigsbee. V. (2015). *Chess.* <www.broadwaymusicalhome.com> Accessed 06/07/2015).

Roads. C. (1996). *The Computer Music Tutorial.* Cambridge: MIT Press.

Roberts. K. (2010). *Chess Synopsis: London Original Version.* <www.musicals.net> Accessed 06/07/2015.

Rosenfield. B. (2010). *Chess Synopsis: Broadway Version.* <www.musicals.net> Accessed 06/07/2015.

Rossant. C. (2014). *IPython: Interactive Computing and Visualization Cookbook.* Birmingham: Packt.

Sadie. S. (1980). *The New Grove Dictionary of Music and Musicians Volume 14.* London: Macmillan.

Saint Louis Art Museum. (2012). *New Media Series – Guido van der Werve: Number twelve: Variations on a Theme*. <www.slam.org> Accessed 23/01/2015

Sale. D. (2014). *Testing Python: Applying Unit Testing, TDD, BDD and Acceptance Testing.* Hoboken: John Wiley.

Saulnier. D, Berry. M. (2009) .*Gregorian Chant: A Guide to the History of Liturgy.* Brewster: Paraclete Press.

Schafroth. C. (2002). *The Art of Chess.* New York: Harry. N. Abrahams.

Scholes. P.A. (1955). *The Oxford Companion to Music*. London: Oxford University Press.

Scholes. P.A. (1983). *The Oxford Companion to Music*. London: Oxford University Press.

Schottstaedt .B. (2015). *S7*. <www.ccrma.stanford.edu> Accessed 02/01/2015.

Schwartz. S; Griffin. T. (2012). *Medical Thinking: The Psychology of Medical Judgement and Decision Making*. Berlin: Springer.

Schwenk. A.J. (1991). Which Rectangular Chessboards have a Knight's Tour? *Mathematics Magazine.* Vol. 16, No. 5, pp. 325-332.

Scimia. E. (2015). *Chess: The Musical.* <www.chess.about.com> Accessed 06/07/2015.

Sedgewick. R; Wayne. K. (2011). *Algorithms.* 4th Ed. New York: Pearson Education.

Selfridge–Field. E. (1997). *Beyond MIDI.* Cambridge: MIT Press.

Sfetcu. N. (2014). *The Game of Chess.* Craiova: Craiova Publishers.

Shannon. C. (1950). Programming a Computer for Playing Chess: *Philosophical Magazine.* Series. 7, Vol. 41, No. 314, pp. 1-18.

Sharma. T.C. (2005). *The Motivation for Learning and Teaching.* New Delhi: Sarup and Sons.

Shenk. D. (2011). *The Immortal Game.* London: Souvenir Press.

Siegel. D.J. (2015). *The Developing Brain: How Relationships and the Brain Interact to Shape Who We Are.* New York: Guilford.

Sierra. K, Bates. B. (2005). *Head First Java.* Sebastopol: O'Reilly.

Simoni. M. (2003). *Algorithmic Composition: A Gentle Introduction to music composition using Common Lisp and Common Music.* <www.quad.lib.umich.edu> Accessed 08/04/2014.

Simoni. M, Dannenberg. R.B. (2013). *Algorithmic Composition: A Guide to Composing Music with Nyquist.* Michigan: Michigan University Press.

Slonimsky. N. (2005). *Nicolas Slonimsky Writings on Music IV: Slonimskyana*. New York: Routledge.

Staff Writers. (2012). *10 Big Brain Benefits of playing chess*. <www.onlinecollegecourses.com> Accessed 10/10/2013.

Stamma. P. (1737). *Essai Sur Le Jeu Des Echecs: où l'on donne quelques regles pour le bien joüer, & remporter l'avantage par des coups fins & subtils, que l'on peut appeller les secrets de ce jeu / par le sieur Philippe Stamma natif d'Alep en Syrie.* Paris: P. Emery.

Stamma. P. (1745). *The Noble Game of Chess.* London: J. Brindley.

Stratton. J. (1999). *Critical Thinking for College Students*. Lanham: Rowman & Littlefield.

Tamassia. R. (2013). *Handbook of Graph Drawing and Visualization.* Boca Raton: CRC Press.

Taruskin. R. (2009). *Music in the Nineteenth Century: Oxford History of Western Music.* London: Oxford University Press.

Taruskin. R. (2010). *Music from the Earliest Notations to the Sixteenth Century*. London: Oxford University Press.

Taube. H.K. (2013). *Notes from the Metalevel*. London: Taylor & Francis.

Taube. R. (2014). *Common Music 3 Dictionary*. <www.commonmusic.sourceforge.net> Accessed 13/01/2015.

Telles. M.A. (2008). *Python Power!: The Comprehensive Guide.* Boston: Course Technology PTR 2008.

Thao. T, Klang. C, Talberg. B. (2006). *JGRASP: a Code Analyzing Tool.* <www.citeseerx.ist.psu.edu.com> Accessed 05/10/2015.

Thompson. O. (1985). *The International Cyclopaedia of Music and Musicians.* New York: Dodd, Mead.

Thompson. T.J. (2008). *Bluetooth Application Programming with the Java API's Essentials Edition.* Burlington: Morgan Kaufmann.

Trigger. B.G, Washburn. W.E. Adams. R.E.W. (1996). *The Cambridge History of the Native Peoples of the Americas.* London: Cambridge University Press.

USCF. (2007). Chess Life. *United States Chess Federation*. Issues.1-6, Vol. 62, pp. 149-183.

van der Dussen. A, Burd. E. (2002). *Proceedings.* Michigan: IEEE Computer Society.

van Rossum. G. (1997). *Comparing Python to Other Languages.* <www.python.org> Accessed 03/10/2015.

Viescas. A. (2015). *How Does Java Work?* <www.smallbusiness.chron.com> Accessed 04/10/2015.

Vogel. M. (2013). *Media of Reason: A Theory of Rationality.* Columbia: Columbia University Press.

Wall. B. (2014). *Chess Engines and Programs.* <www.chessmaniac.com> Accessed 05/07/2015.

Watters. A, van Rossum. G. (1996). *Internet Programming with Python.* Redwood: M&T Books.

Weber. J.F. (2015a). *Early Western Chant.* <www.liturgica.com> Accessed 18/07/2015.

Weber. J.F. (2015b). *Gregorian Chant.* <www.liturgica.com> Accessed 18/07/2015.

West. B. (2013). *Hummingbird Notation.* <www.humingbirdnotation.com> Accessed 17/10/2014.

Williams. G. (2001). *Chess: An Easy–to–Follow Illustrated Guide to Playing this Popular Game of Skill.* London: Parragon.

Wilson. J.F. (2003). *Biological Foundations of Human Behaviour.* Belmont: Wadsworth Publishing.

Winter. E. (2014). *Sergei Prokofiev and Chess*. <www.chesshistory.com> Accessed 22/01/2015.

Wise. P, Van Der Spuy. M. (1986). *Musical History and General Knowledge of Music.* Cape Town: Nasou Limited.

Wolf. M. (2002). *Er tut Dinge, mit denen Fritz nicht rechnet. <www.berliner-zeitung.de*> Accessed 06/07/2015.

Wüllenweber. M, Friedel. F, Feist. M. (2006). Kramnik vs Deep Fritz: Computer Wins Match: *ICGA Journal.* Vol. 29, No. 4, pp. 208-213.

Xenakis. I. (1992). *Formalized Music: thought and Mathematics in Composition.* New York. Pendragon.

Yang. Y.H. (2011). *Music Emotion Recognition: Multimedia Computing, Communication and Intelligence.* Boca Raton: CRC Press.

Yung. Y.H. (2011). Ranking– Based Emotion Recognition for Music Organization and Retrieval. *IEE Transitions on Audio, Speech, and Language Processing*. Vol. 19, No. 4.

Zelle. J.M. (2004). *Python Programming: An Introduction to Computer Science.* Portland: Franklin Beedle and Associates.