Bard: software development for algorithmic composition

- making music out of chess games in real time -

Master Thesis Report

Emmanouil K. Papageorgiou

Aalborg University
Architecture , Design and Media Technology

AALBORG UNIVERSITY
STUDENT REPORT

Architecture, Design and Media Technology
Aalborg University
http://www.aau.dk

Title:
Bard: Software Development for Algorithmic Composition

Theme:
Music Interaction

Project Period:
Spring Semester 22

Project Group:
XXX

Participant(s):
Emmanouil Papageorgiou

Supervisor(s):
Sofia Dahl

Copies: 1

Page Numbers: 49

Date of Completion:
April 7, 2022

Abstract:

This project investigates the creative interpolation of music into the game of chess. The combination of the data coming from a chess notation system with a chess engine's calculations, constitutes a sufficient amount of input information for the development of a method for algorithmic composition that could provide a meaningful musical narration to a chess game being played. If the processing of these chess-related pieces of data is taken as a function of the music tonal system, then the sonified output could evoke feelings that appeal to how chess players experience the positions of their game. To test the algorithmic composition method that was developed for such a concept, this project juxtaposed a variation of the software to the participants of the experiment, one that does not really interact with their moves on the chessboard, but with another, prerecorded sum of moves. The participants to the experiment were asked the same questions about their experience with interacting with the two systems without knowing which variation truly interacts with their moves. After comparing, contrasting and analyzing the collected data, the interacting algorithm scored higher than its predefined counterpart in all proposed questions, thus indicating that the musical outcome can be meaningfully perceived.

Contents

Preface

This paper details the development of an application for making music out of chess games in real-time. It would not be possible without prof. Sofia Dahl's guidance, the supervisor for this project.

Aalborg University, April 7, 2022

Emmanuel Papageorgiou
Emmanouil Papageorgiou
Study No. 20190995
epapag19@student.aau.dk

Chapter 1

Introduction

This paper details and elaborates on the development of a method for algorithmically composing music out of the game of chess in real time.

Algorithmic composition (or automated composition) was defined by Adam Alpern in 1995 as the "process of using some formal process to make music with minimal human intervention" [1]. Before giving a name to this concept, throughout the history of music, there has a been a plethora of efforts to automate the process of composition; either for solidifying and showing off the capabilities of a theoretical system of a whole aesthetic era (canons, sonatas, fugues etc.), for breaking through such a consolidated theoretical system (Schoenberg's serialism, Xenakis's stochastic music etc.) or even just for the sake of creating a musical game (e.g. Mozart's "Dice Music").

The game of chess, just like music, has been subjected to an unprecedented dehumanization due to modern day technology and all the various useful applications that go along with them. Artificial Intelligence has long surpassed the human intellect when it comes to pondering the way through a winning position in chess. That said, it is a common misconception that chess is purely a game of logic. There is some kind of style in chess, as there is in art and music. Chess Masters can be so easily categorized by their playing style that when someone replays a tournament game, they can guess which players played that game [7]. In fact, as in real life, a player is asked to decide between a certain number of alternatives of equal value. Although rational thinking is a priority, imagination and intuition are equally important - patterns are the player's arsenal. In complex situations there comes a point where a decision has to be made and it can be obvious when one move is better than all the other alternatives. However, most of the time, chess is an intuitive game where personal taste and emotions, as well as external factors, influence one's decisions. Therefore, factors such as the desire to take risks, to define one of the many, determine the style of each player.

Proportionately, the same can be said for a music performer or composer. Alicia de Larrocha (1923-2009), one of the leading pianists of the last century said in an interview: "Every composer is possessed by his own atmosphere and his own character. The style accompanies the composer's era and his own humanism". In the same interview she continues: "I believe that loudness is the portrait of the personality of every pianist or musician. Every artist is like the shape of the face, meaning it has its own sound, something very, very personal [4].

The common use of the word "play" when referring to chess and music is not accidental. To play means to act. The action takes place within the same framework. As in mathematics, music and chess have a precise language of symbols. A piece of music or a game of chess can thus be reproduced by musical instruments with or without a chessboard. One of the skills of a good player is abstract reasoning: musicians are able to "listen" to music simply by looking at the score and chess players are able to "see" a sequence of moves by simply reading the notation.

This very notation, with which all possible information with regards to a chess game is recorded, in combination with a chess engine's (AI) evaluation of the positions, can provide a set of data descriptive enough to render a musical composition that follows the dynamics of the opposing sides' clash on the chessboard. The musical notation (as well as the entirety of the measurable musical components) has already been translated to an integer number based language (MIDI protocol). If the chess data were also to be transcribed to a an integer based system, then theoretically, the emotional expression that the music tonal system has incorporated in its structure could be rendered from the events of a chess game through the operations of an algorithm.

The algorithm created for this project is called "***Bard***", named after the lyrical poets composing songs and reciting the epic battles throughout the Middle Ages. In the following sections, this paper lists projects affiliated or related to *Bard*, details the development of the composition's method, elaborates on the implementation of the software, presents the experiment designed to test the application's efficiency and provides the outcome of the evaluation.

Chapter 2

Related Work

2.1 Serialism & Coalition Chess

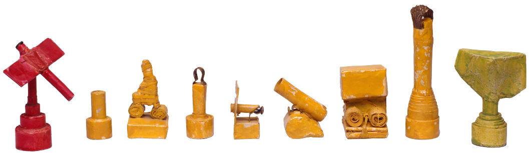


Figure 2.1: Schoenberg's Coalition Chess pieces

The software developed for this project creates musical outcomes based on melodic series that derive from the incoming input of chess notation. This project's method of composition is essentially an application of Serialism, a method in which every musical element (pitch, rhythm, timbre, dynamics) is arranged by a predetermined sequence of numbers, a series. The idea of serialism was primarily implemented by Arnold Schönberg (1874-1951) with his 12-tone technique which uses the unique occurrences of all the subdivisions of the octave to formulate a melodic sequence of 12 steps that sets up the conditions for a custom harmony and structural progression.

Being a multi-talented personality, Schönberg expanded the scope of his activities in the teaching of composition, painting and inventions. Some of his students stand out being great personalities such as Alban Berg, Anton Webern, Hanns Eisler, Nikos Skalkotas and John Cage. One of the games invented by the prolific composer was *coalition chess*. Christian Meyer reports: "It cannot be mere luck that exactly at the same period that Schönberg was working on the series of his twelve-tone technique, he was also developing the revolutionary game of coalition chess." Coalition chess is a variation of traditional chess, played by four players in 10 by 10 squares chessboard. From the Arnold Schönberg Center website: "By inventing Coalition Chess in the early 1920's, Arnold Schönberg defied the rules of chess. By inventing the method for composition with the 12 tones which are only driven by their own ratio, he shook the foundations of the traditional rules of Western Music."

In his book, *Theory of Harmony*[13], Schoenberg calls upon terms of war and revolution to describe the transposing functions from one tonality to another, comparing the tonic degree with a tyrant and the dominant and subdominant with rebels ready to rise and take over. It is possible that this interpretation of musical clash and collision is what Schoenberg based the rules of Coalition Chess on[8].

2.2 REUNION

On the 5th of March 1968, in Toronto, John Cage and Marcel Duchamp played a game of musical chess. This event, by the title "**Reunion**", attracted an audience of hundreds in Ryerson Theater, where these two iconic figures would trigger an auditory experience through a custom electronic chessboard with their every move.

Marcel Duchamp, a French-American lecturer of Dada and an artist that along with Picasso and Henry Matisse is said to have defined the evolution of the plastic arts in the early 20th century, was deemed by Grand Master Emmanuel Lasker as one of the 25 strongest chess masters in the U.S.A.

John Cage, a composer, music theorist and pioneer of musical indeterminacy and electroacoustic music, charmed from Duchamp's artistic persona, was asking him -as an excuse- for chess lessons, until he conceived the idea of a musical chess performance (Tomkins 1966)[3]

The functions of the chessboard were based on covering and uncovering 64 photoresistors, one for each square. An oscilloscope was transmitting images in customized TV-screens, thus allowing the visualization of some of the auditory events triggered by the chessboard. Four composers, Lowell Cross, Gordon Mumma, David Behrman and David Tudor designed the sound generators. The project's goal was to define the shape and the form of the performance's acoustic environment via chess moves.

Lowell Cross mentions that Cage asked him to develop an electronic chessboard which would choose and distribute sounds around the audience while the chess game would progress. He told him that this performance would be called "Reunion" because he wanted to bring artists with whom he had collaborated in the past together in a familiar location, suitable for concerts[3].

2.3 Music for 32 Chess Pieces

Music for 32 Chess Pieces is a system that allows users to create music by playing chess using networked computers. A Server takes the input of the player's move commands and configures the changes made on the user interface concerning the composition parameters. A set of game-to-sound plugins correlate the chess positions to musical phrases and sends open sound control (OSC) messages to a platform for tone generation[12]. The main difference in the design of this approach with *Bard* is the element of meaningful musical narration.

2.4 Apollo Meets Caissa

Music and Chess: Apollo meets Caissa [15] is an all around study of the dipole of music and chess, from its historic research to the technical deconstruction of its components. Zographos juxtaposes the music elements in respect to their chess counterparts and describes the dynamics of their dialectic relationship. Many of this work's principals have been incorporated in *Bard*'s algorithm design.

2.5 Bard

2.5.1 Sample-Based Algorithmic Composition

The first installment of *Bard* [11] worked with the same components but with a completely different design. The chess notation was triggering prerecorded sample: each piece bore a specific sound sample and each combination of departure-arrival square was forming a specific chord. The chess engine's evaluation of the position was functioning as a gain slider that determined whose side's sound amplitude would prevail.

2.5.2 Sound Location for an 8x8 Matrix (Chessboard)

In an other version of this algorithm, the chess engine was suggesting the best move for each new move to be played in a chess game and provided a cue in a sound map designed in two different approaches:

1. **HRTF** version: using a head related transfer function, this approach was placing the user in the center of the chessboard. Tee directionality of the sound was fluctuating in three axes: left-right, up-down, forwards-backwards.

2. **Speaker-to-Speaker Crossfade** version: using a dynamic panning technique which adjusted the amplitude of the sound in proportion to the square's distance from the user for the x-axis, this approach was placing the user at the center of the bottom of the chessboard. The y-axis was perceived by tuning the pitch of the sound: the higher the pitch the higher the row of the matrix, the lower the pitch, the closer to the user.

2.5.3 Current Project's Problem Statement

In contrast to the previous works mentioned in this chapter, this installment of Bard aims to utilize the real-time evaluation of a chess engine in a manner that not only would it partake in the development of the method of algorithmic composition, but it would also convey black and white's dialectic relationship on the board, as well as the events of the game in a meaningful way through music.

Due to the fact that the theory of the harmony of tonal music has been imprinted in the listening perceptions of the western audience, the emotional range can be quite clearly expressed to that audience (of western music)[5].

If this expressive universality of the harmony of music is put to react with an other integrated system, such as the Universal Chess Interface (UCI), then it would be possible to produce a meaningful musical commentary on the chess game being played, or in other words, program music (as in symphonic poems) which renders an extra-musical narrative musically[6].

Chapter 3

Algorithmic Composition Methodology

Below follows a conceptual algorithm that describes the methodology behind this project's implementation of the software's development in four steps.

1. **Isolating the Chess Components**

   The classic algebraic chess notation provides all the data necessary to enlist all chess events: type of moved piece, capturing, castling, checking, position of the piece inside the chess matrix. For example, the expression *R1xa8+* means that the rook from row 1 captured a piece at a8 and checked the king. White always opens the game, so by counting the moves, the turn of the player to play is also known. After isolating all different types of chess data, the chess engine's evaluation is added to the list of the chess components. Once the set of chess components is integrated, there starts their association to the musical components.

2. **Handling Time**

   The greatest problem with real time interactive algorithmic composition is that the musical component of time is not up to the composer's disposal. For that reason, there has to be found a way to constantly sonify a static state which in this case, is each position of the game between two consecutive moves. To solve this problem, Bard utilizes the concept of step sequencers whose function is to periodically iterate the elements of a series in a given duration for each step. That way, even if a player gets stuck in a position, pondering their next move, the music will not stop. Subsequently, this leads to another problem: the auditory outcome of the game at that time period will be monotonously repetitive. Here comes the chess engine's evaluation as a deus ex machina: the floating number representing the opposing sides' advantage ratio configures the step sequencer's metronome (the duration provider to the beating steps), turning it into a pulsar sequencer which generates complex but periodic musical rhythms. So, although the engine's calculations do not yet meaningfully partake in the composition, they already secured the auditory variance and singularity of each possible chess position.

3. **Melody & Chord Generation**

   In Western Music, each musical mode is a specific sequence of seven intervals between eight pitches. The fact that a chessboard is an 8x8 square matrix comes in handy: after the initial modality is determined, the pieces' moves coordinates are being used as indices to create a series of notes chosen out from the applied mode. This series, as it is later explained in section 5.2.1, functions as the composition's melody. Consequently, after establishing the composition's modality, the field for generating harmonies is set: the notes that appear in each series make up for the selection of each chord's elements with intervals of 3rds, 5ths and their reversals.

Up to this point, this conceptual algorithm has integrated all the basic elements of a music composition: melody, harmony and rhythm. Now, there needs to be found a way to express the musical components with regards to the events of the chessboard.

4. **Infusing "meaning" to the mix**

   Besides the chess events (captures, castling, threats to the king, etc.) which are provided by the notation, some additional inferences can be added to the extra-musical set of concepts to be sonified. The chess engine calculates the advantage ratio of the opposing sides at each given position. That means that the quality of each new move can be calculated on the spot, thus categorizing each move in four distinct types: correct, inaccuracy, mistake and blunder.

   In Western Music, the harmonic progressions and the tonal transpositions are known to have a certain emotional impact on listeners. Transpositions to the relative key or to the dominant and/or subdominant scale degree are considered to be very natural, thus expressing a set of emotions that generally fall into the spectrum of serenity and anticipation. On the contrary, transpositions to neighboring keys are considered to be rough and fall in the emotional spectrum of disorientation and angst or agony. Consequently, the moves' type can be correlated with transposing accordingly to a different tonal center, to a different musical mode and thus appealing to the analogous emotion.

Chapter 4

Design

4.1 Concept

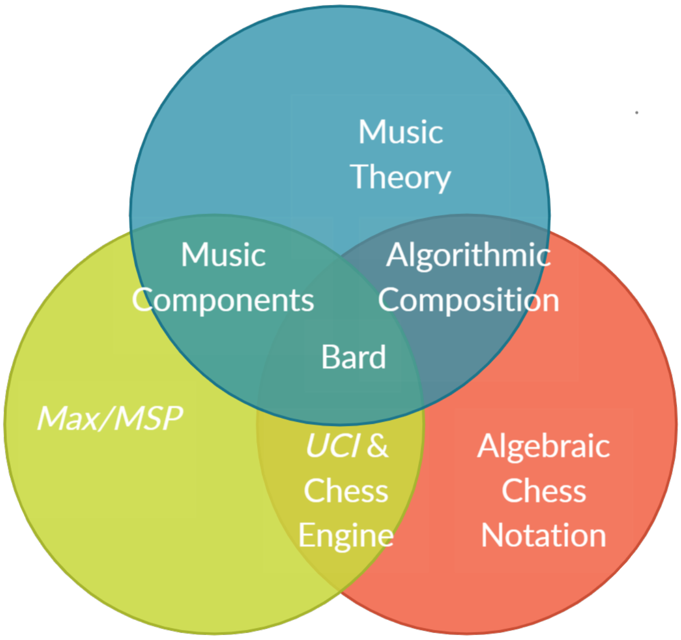


Figure 4.1: *Bard*'s Venn Diagram of Concept Design

Figure 4.1 shows the concept design for this project. Contextually, *Bard* is the result of the conjunction of the programming language of Max, the rules of Western Music's theory and the classic algebraic chess notation. Within this conjunction, the chess engine works in synergy with the musical components to define the algorithmic method for composition designed for this project.

4.2 Workflow

Figure 4.2 describes the input and output of Bard's operations. The users' moves trigger the UCI (Universal chess interface) object to generate the chess notation which undergoes a process of data conversion and when combined with the chess engine's evaluation, they make up the input of the algorithm. The output is an auditory outcome that features an ongoing melody, chords, key changes, modal changes as well as ongoing rhythm progressions.

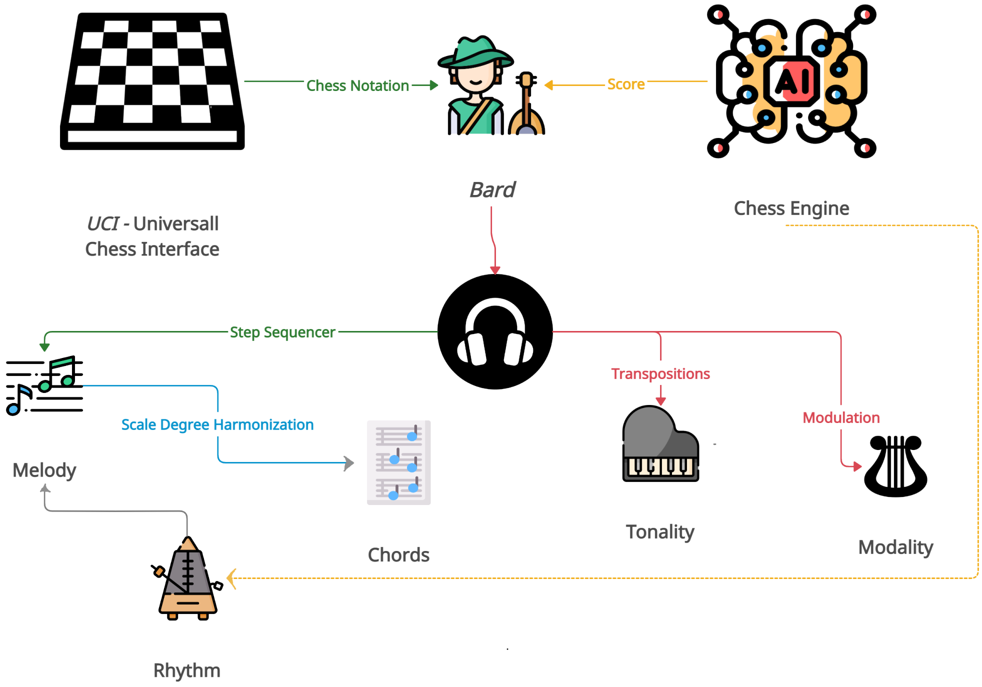


Figure 4.2: *Bard*'s Input and Output

4.3 Experiment

Figure 4.3 describes the experiment's repeated measures for related samples design in which the research study addresses to a single sample group to test different treatment conditions. In this project's case, one treatment condition has the sample testing the standard version of the software and the second condition has the sample testing the same software but without actually interacting with it. In order to deal with this design's order effects (changes in scores may occur from participating in earlier treatment), the order of participating in each condition had to be counterbalanced. The quantitative data deriving from the questionnaire were evaluated according to the question/statement's nature and the ones suitable for analysis and comparison, underwent the t-testing procedure to determine if they differed significantly.

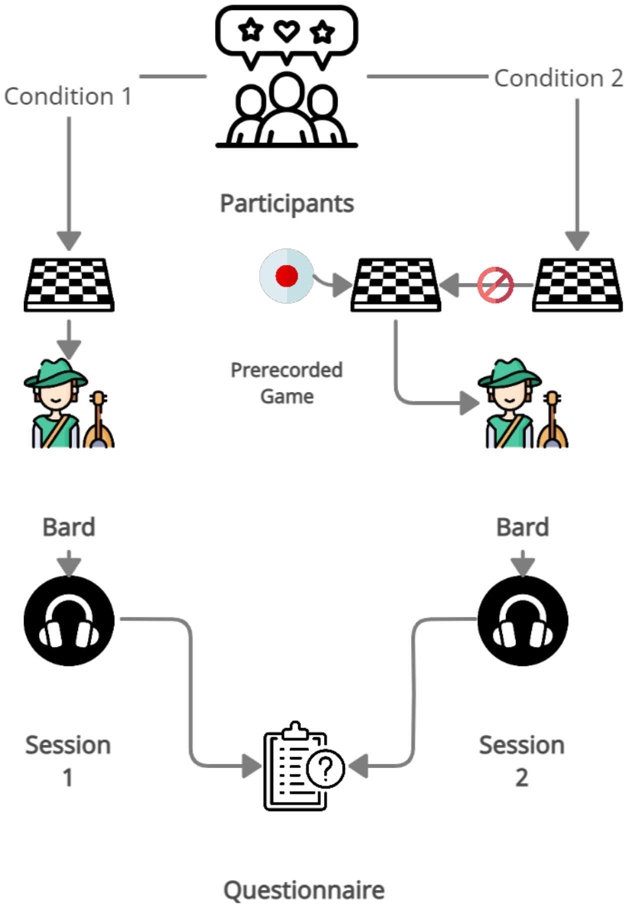


Figure 4.3: Related Samples - Repeated Measures Design

Chapter 5

Implementation

The development of *Bard* has taken place in the visual programming language of Max/MSP. This project utilized two external components: multimedia composer and programmer Jeremy Bernstein's *uci* object [2] and Joseph Vincent Manzo's *modal object library*. [9].

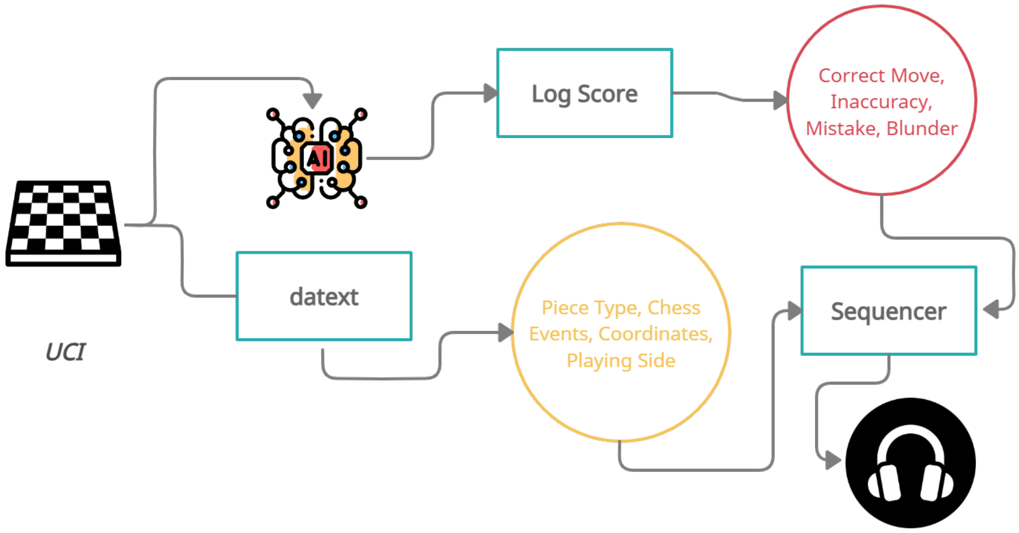

Log Score
Correct Move, Inaccuracy, Mistake, Blunder
datext
Piece Type, Chess Events, Coordinates, Playing Side
Sequencer
UCI

Figure 5.1: *Bard*'s General Operation Algorithm

5.1 Chess Components

The processing of *uci*'s stream of data led to the implementation of the patcher named "**datext**" (in earlier work), the function of which is to disintegrate the incoming classic algebraic notation and break its messages down to separate elements.

These alphanumeric elements then got sorted out and coded into ASCII values so that they could ultimately represent all possible pieces of information that chess notation can provide in the form of integer numbers. These pieces of data function as separate entities in the development of the algorithm and their components refer to:

1. the piece type (K ♔ , Q ♕, R ♖, B ♗, N ♘ and pawns)
2. the act of capturing a piece
3. the act checking the king
4. the act mating the king
5. the act castling
6. the coordinates of the square occupied before and after a piece moves
7. black or white's turn to play
8. the move counter

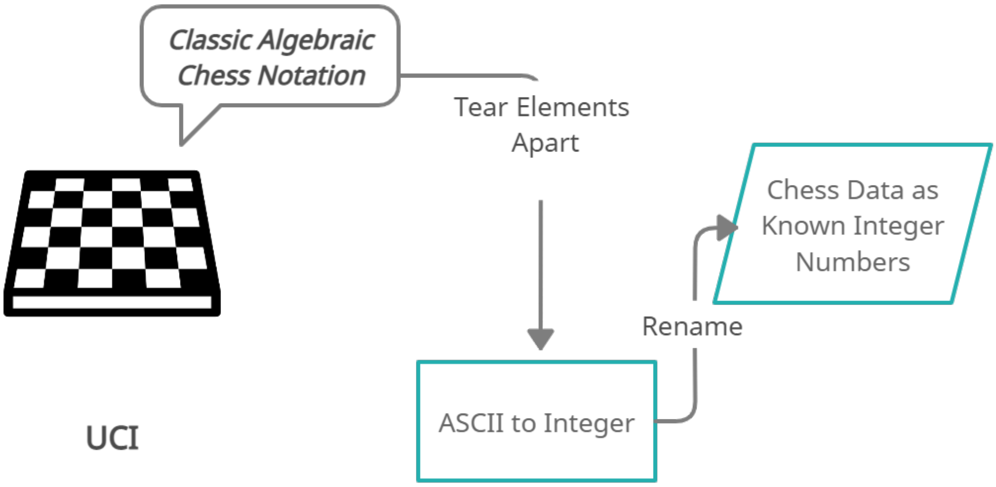

Classic Algebraic Chess Notation
Tear Elements Apart
Chess Data as Known Integer Numbers
Rename
ASCII to Integer
UCI

Figure 5.2: Converting the Chess Data

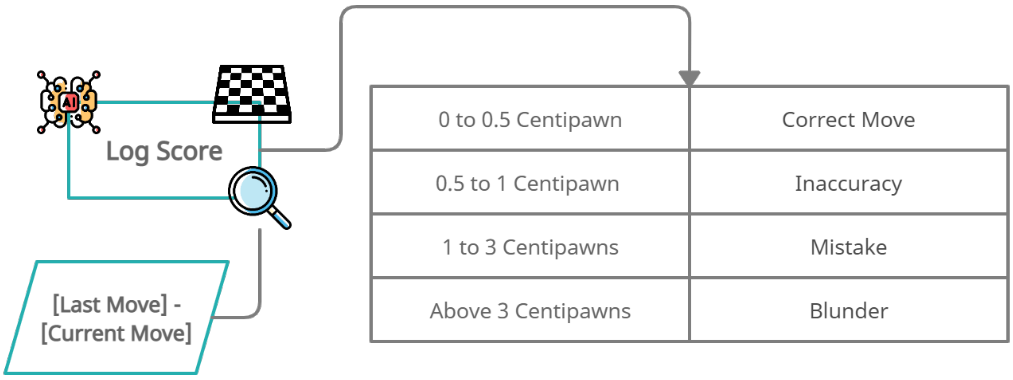
5.1.1 The Chess Engine

The *uci* object can load to different chess engines: Stockfish or Critter . The chess engine provides a real time evaluation of each current position of the chessboard in **centipawns**, a measurement unit which is defined as a real number and can roughly range from -60 to +60. The point of reference is 0, where the concept of a balanced chess position is indicated (white is attributed with a ≈ 0.2 centipawn advantage at the starting position - in most chess engines). The more the engine's evaluation tends towards the positive values, the bigger the advantage for white, and vice versa; the more the evaluation tends towards the negative values, the bigger the advantage for black.

5.1.2 Patcher Score Log

A chess move can be characterized as correct, inaccuracy, mistake or blunder. In chess programming, the "centipawn" measurement unit is also being used for the categorization and definition of the moves' quality.

The "score log" patcher keeps a catalogue of the moves' index, accompanied by the engine's evaluation of the position at the given instance. Then, it outputs the difference of the current position's score with the next to last position's score.



0 to 0.5 Centipawn	Correct Move
0.5 to 1 Centipawn	Inaccuracy
1 to 3 Centipawns	Mistake
Above 3 Centipawns	Blunder

Figure 5.3: Calculating the Moves' Quality

When this numeric difference is

1. less than 0.5 centipawn, the move is deemed as **correct**
2. between 0.5 and 1 centipawn, the move is deemed as an **inaccuracy**
3. between 1 and 3 centipawns, the move is deemed as a **mistake**
4. greater than 3 centipawns, the move is deemed to be a **blunder**

Every time that a move is played on the board, the algorithm calculates the difference between the last and next to last evaluations and outputs a *bang* accordingly, so as to determine the incoming move's quality.

Patcher Best Suggested Move

Identically to how the chess components derive from the chess notation (mentioned above), there goes a similar process for the extraction of the engine's best move suggestion. When the engine analyzes the position for the purpose of outputting the score and/or deciding what to play, it calculates a series of possible answers to the last played move. It , then, ponders its opponent's best move and comes to a final decision based on that assessment. This pondering of the opponent's best move can be therefore extracted from the analytical process and be consequently treated as a separate entity in the algorithm's functions.

5.2 Musical Components

The algorithm runs two corresponding music data producing processes: one that outputs melodies, one that outputs chords. From the *modal object library*, the *modal change* object was utilized for reading music tonalities and modalities: it contains a library of the interval sequences of 27 different modalities for the twelve different tones in midi values. The *modal triad* object extracts the midi values for the chords of the selected tonality's scale degrees.

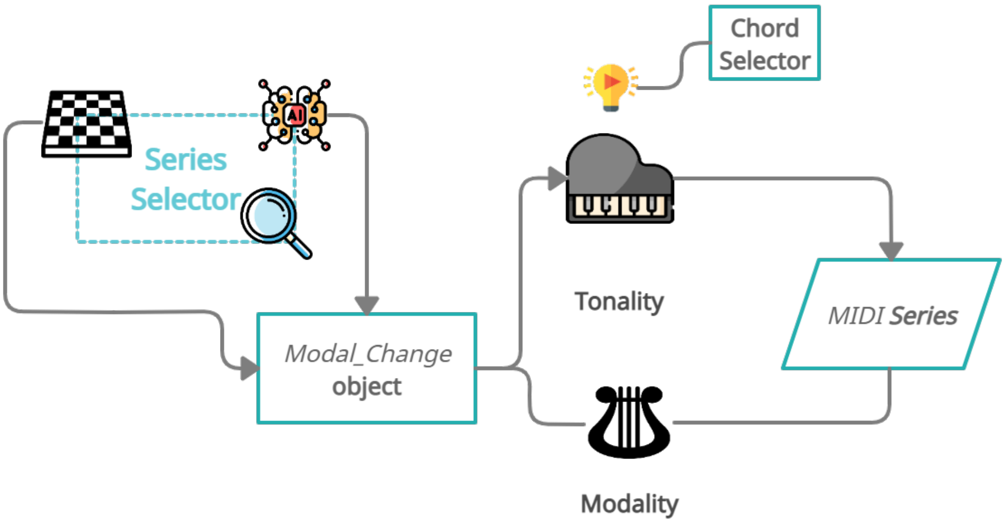


Figure 5.4: Series Selector Patch - Choosing the MIDI series

5.2.1 Series Selector

The series selector patch is responsible for outputting the melody of the algorithmic composition.

Until the first move is played , there is no input for the *modal object library*, hence there is no tonality and modality to transpose to. For the first move, and for the first move only, an abstract set of tonalities and modalities were manually mapped to the 20 first possible chess moves as an initiator to the whole process.

After the tonality and modality have been determined for the first time, the object starts providing the interval sequences indicated in midi values which essentially represent music scales or music modes. Then, the calculation of the remainder of the division of the midi value by 12, is defining the pitch class set of the tonality and the pitch class of each produced midi value.

The creation of the actual melody derives from the set of 4 different coordinates on the chessboard: the square which the moving piece departed from, the square that the piece arrived to, as well as the 2 coordinates of the engine's best suggested move (departure square - arrival square). These coordinates (and in that sequence) function as indices, value selectors from the music scale sequence that was previously determined.

5.2.2 Scale Degree selector

Depending on the piece type (K, Q, R, B, N or pawn), the algorithm chooses to play specific scale degrees:

1. The King ♔ along with the pawns trigger the scale degree of the tonic (I)

2. The Queen ♕ triggers the scale degree of the subtonic (VII)

3. The Rooks ♖ trigger the scale degree of the dominant (V)

4. The Bishops ♗ trigger the scale degree of the sub-median (VI)

5. The Knights ♘ trigger the scale degree of the subdominant (IV)

Chord Generator

The above mentioned scale degree selector is embedded in patch called *makechord* and sends its output to the *modaltriad* object which then picks the relative chord out from each current tonality. It then procceeds to output its data to the octave selector.

5.2.3 Octave Selector

The *Octave Selector* patch determines the octave range within which the incoming chords will sound. This range unfolds from the second through the sixth octave of a theoretical 88-key piano keyboard. This configuration of octaves is based on the engine's evaluation of each current position. It is the only feature of the algorithm that actually gives specific information away about the score to the players. When the engine's evaluation fluctuates in the determined range for inaccuracies (including correct moves), the *makechord* patch sends the pitch classes to be sonified at the fourth octave (middle octave). Consequently, the slight and the clear advantage for black sends the pitch classes to the third and second octave respectively, while the slight and clear advantage for white sends them to the fifth and sixth octave respectively.

5.3 Step Sequencer

Given that the previous sub-patches are all in embedded in this "*sequencer*" patch, this is the final destination of the functions of the algorithm; the memory space within which the chess components are interpolated with the musical components and provide the integrated set of midi values which constitute the outcome of the algorithmic composition.

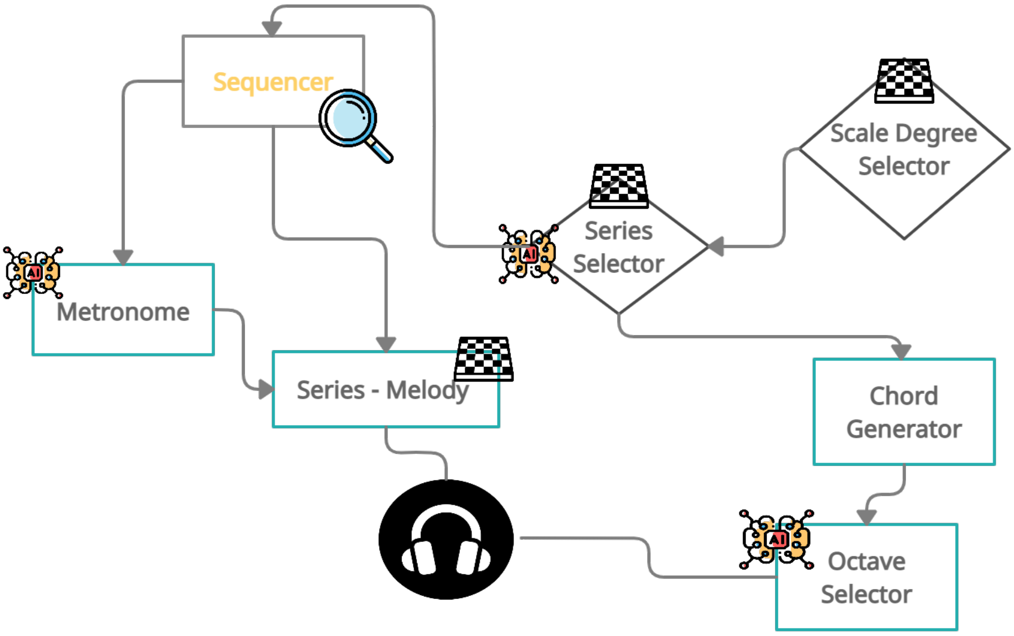


Figure 5.5: The Sequencer - related Operations

5.3.1 Creating Melodies

As explained above, the series selector patch outputs the melody in the form of a series of midi values that derive from the indicated notes that belong to the given tonality, provided by the coordinates of the last played move and the coordinates of the best next suggested move. This set of eight midi values per play turn can contain repetitions of a note as the coordinates from where it derives from might share elements.

This series is then driven into the *multislider* object along with the input of the clock (metronome) which determines the duration of each of the eight steps of the sequence in beats per minute (BpM). The *zl.scramble* object then shuffles the list's elements every two iterations of the sequencing cycle so that the melody would not remain static when players take their time to ponder their next move.

Consequently, the output of the *multislider* is one integer number (midi value) per sequenced step and it is being utilized by the algorithm in two ways. The first obvious function is to immediately sonify this number by treating it as what it really is: a midi value. The second function of this datum is what turns this system into a closed loop.

5.3.2 Sequencing the Clock

Inspired by analogue sequencing and applied to digital sequencing for *Bard*, the function explained below is a very simple implementation of one of Allen Strange's multiplications [14].

First, the floating point arithmetic (decimal number) representing the position's deviation from balance is converted to its absolute value, so that it would reflect the general tension of the chessboard without taking winning sides into consideration. Then, it gets scaled from the range that it would normally fluctuate between (-60 to +60, but 0 to +60 after absolute value conversion), to a numeric range representing time (BpM), a set of time values that would make sense in terms of making music. The criteria by which this scaling process was configured, were subjected to the author's personal aesthetics and the product of empirical experimentation on evaluating different variations of the time-scaling.

The outcome of the multiplication of the midi value (see previous section) with this interpolated version of the chess engine's evaluation of the position is then sent as input to the system's metronome, thus determining each step's duration; a dialectic conjugation that ultimately serves as the inherent musical rhythm of the game of chess being conducted.

For the purpose of musically commenting on the events of the chess board, when a piece is captured, the duration of the steps drops down in half and t is defined as:

$$t = \frac{Midi \times Sc}{6}$$

while when no captures take place, t is defined as:

$$t = \frac{Midi \times Sc}{12}$$

$$Midi \in [0...12] - Sc \in [50...750]$$

5.4 Transposing Function

Incoming data from the *score log* patch, defining the last played move as either correct, inaccuracy , mistake or blunder, also determine the harmonic progression of the composition: when the last move played is

1. correct, the tonal center remains the same.
2. an inaccuracy, the tonal center shifts to the dominant scale degree (V)
3. a mistake, the tonal center shifts to sub-median (VI)
4. a blunder, the tonal center is transposed up a semitone

This is simply implemented by adding the relevant number of semitones to the *modal change* object when the corresponding type of move occurs.

5.5 Sonification

The overall system's output has to do with a stream of MIDI values. In order to turn this stream of data into sound, it has to be sent to a virtual instrument processor.

For this reason, *Ableton*, a digital audio workstation (DAW) came to play, as this project revolves around the methodology of an algorithmic composition based on chess; its instrumentation would constitute a whole new different project.

Concerning the sonification of the data in *Ableton*, the VSTs used and the programme's settings are elaborated on in the following chapter.

Chapter 6

Evaluation

The experiment that was designed for the evaluation of this project studies the capability of the developed software "*Bard*" to make music that could evoke feelings that appeal to how chess players experience the positions of their game, or in other words, its capability of providing meaningful musical commentary with regards to chess playing.

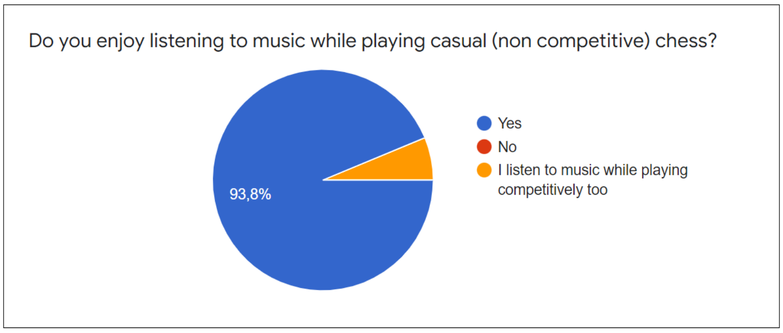


Figure 6.1: Accompanying chess with music - Users' feedback

In this section, all references to "**Game 1**" stand for the **non interacting version** of the algorithm and all references to "**Game 2**" stand for the **interacting version**

6.1 Experiment

In order to test that hypothesis, a variation of the program had to be developed; such one that would give the impression to the participants that they are interacting with the algorithm, while they practically are not doing so. The concept of the experiment's design is as follows:

"A chess player is shown to two different versions of *Bard*: version one is actually interacting with their moves and version two is interacting with a different set of moves. They are asked to try out both versions without them knowing with which version they are playing. After each session, they are asked if their interaction with the interface produced a musical outcome that was contextually following the development of their moves in the chessboard.

Bard would be **failing to meet** its objective if:

1. Participants respond negatively for both versions
2. Participants respond positively for both versions
3. Participants respond negatively for the interacting version and positively for the non-interactive version

   Bard would be **meeting** its objective if:

4. Participants respond positively for the interacting version and negatively for the non-interactive version"

The participants were asked to navigate the interface's chessboard, essentially playing against themselves - moving both black and white's pieces, for two games, 4-5 minutes each. One game would use the interacting interface, the other would show them their moves but it would practically be sonifying a prerecorded set of moves. They were asked to play in a manner that would examine all possible fluctuations of a chess position's dynamics: maintaining the balance (by playing correct moves / inaccuracies), disturbing the balance (actively making mistakes or blunders). The participants would not know which game is which and after they had completed each session, they were asked to provide answers to a questionnaire (identical for both games) regarding their experience.

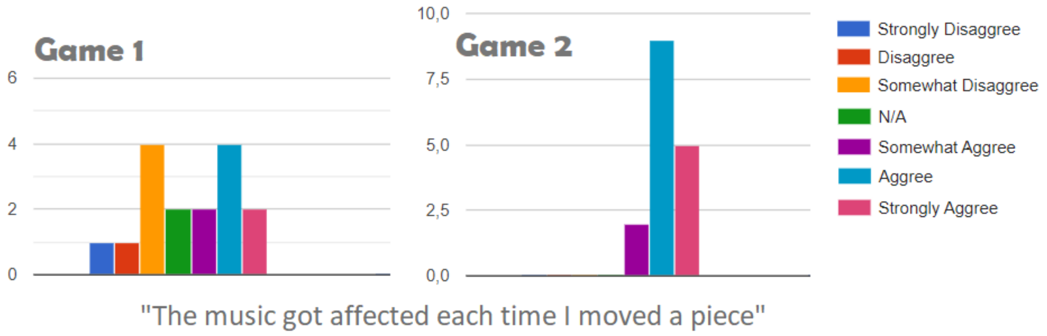


Figure 6.2: Answers to question/statement #1 for games 1 and 2

6.1.1 The questionnaire

Game 1 and Game 2

Half of the participants tried out the interacting version of the application first, proceeded to provide the associated feedback and then do one more iteration for the non-interacting version of the application; vice versa for the other half. This adjustment to the experimental procedure was applied in order to counterbalance the effects of first-time exposure prejudice.

Participants provided answers in a 7-point Likert scale in which the options were listed in the following sequence:

1. Strongly Disagree
2. Disagree
3. Somewhat Disasgree
4. N/a
5. Somewhat Agree
6. Agree
7. Strongly Agree

The questions/statements concerning the participants' experience with using *Bard* for games 1 and 2 were listed as follows:

1. The music was affected each time I moved a piece
2. When a piece was captured on the board I could hear it in the changes in the music
3. The transitions between different stages (the opening, the middle game and the endgame) could be heard in the music
4. When the balance of the position was getting disrupted, I could hear it in the music
5. The music reflected the correctness, inaccuracy, mistake, or blunder from a move made on the board.

After providing feedback on their interactive experience with the software, the interviewees were asked if they were familiar with the famous "Opera Game", the 1858 chess game, played at an opera house in Paris between the American master Paul Morphy against the German noble Karl II, Duke of Brunswick and the French aristocrat Comte Isouard de Vauvenargues. They were ,then, shown to a video recording of it being played on *Bard* and were asked to evaluate on what degree did they feel that the sonification was musically describing the dynamic's of the game's positions on a scale from 1 (to a low degree) to 10 (to a high degree).

The participants then proceeded on writing down a description of their overall experience from playing the two games with the system and were then asked if they would be interested in using such an application foe casual chess play.

For the sake of better understanding of the enclosed data, the participants were asked to also provide information about their chess skills on various available measurable sources: FIDE (International Chess Federation) ratings, chess.com ratings and lichess.org ratings. As far as music is concerned, participants were asked if they enjoy listening to music while playing casual chess and were then asked to provide information that configure the GMSI , Goldsmith's Music Sophistication Index.

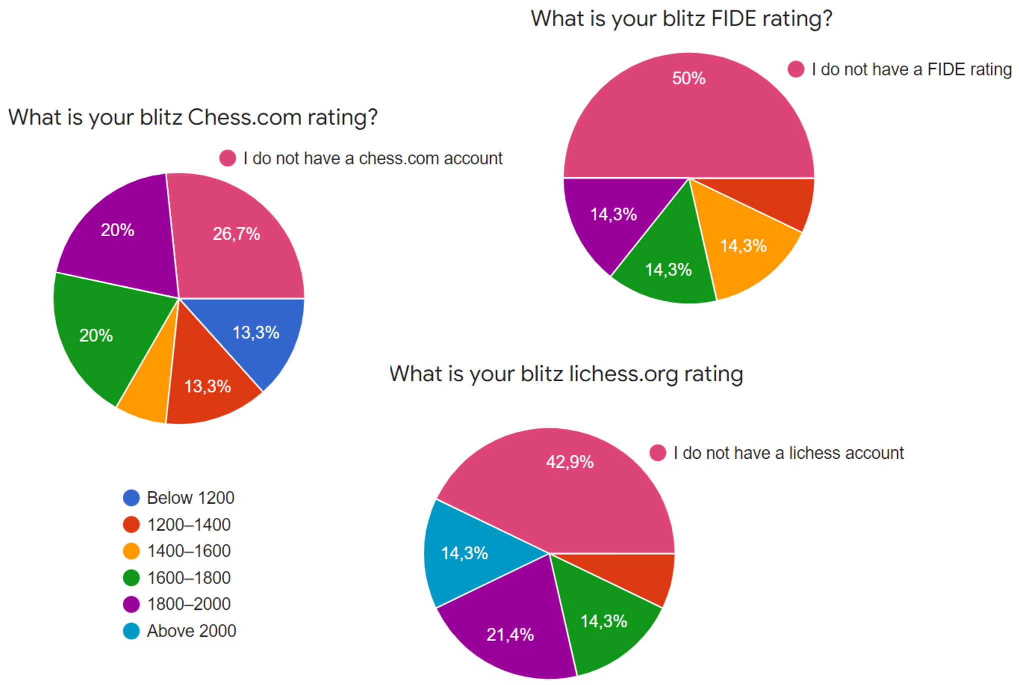


Figure 6.3: Chess Skill Level of the Participants as per ELO ratings

6.1.2 Participants

This project's target group is chess players of all different levels of skill who enjoy listening to music when playing chess. Consequently, the recruiting of the participants to the experiment concerned chess players regardless of age, gender, level of chess skill or musical education. Nevertheless, for the sake of data analysis and the results' evaluation, participants were asked to provide the experiment's conductor with information concerning their age, their chess skills and their level of involvement with music as listeners and/or scholars.

All participants to the experiment were informed about the academic purpose of the project, the nature of the data being collected (**age, name, ELO ratings and statements concerning music and the project**) as well as their right to interrupt the procedure and revoke any or all data provided, at any point in time.

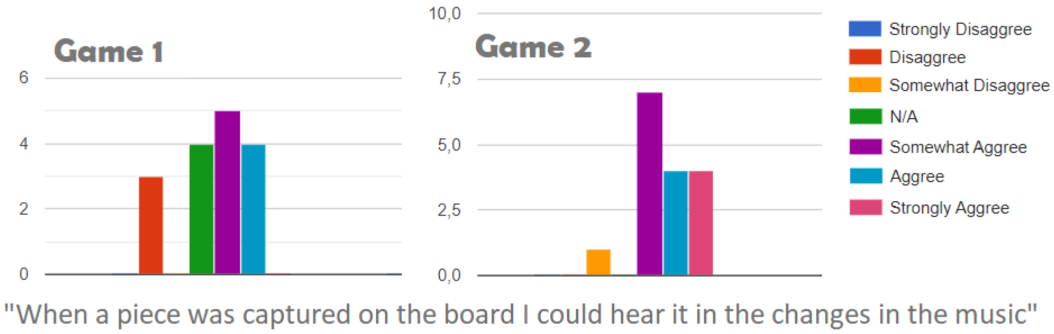


Figure 6.4: Answers to question/statement #2 for games 1 and 2

6.1.3 Materials - Experimental Setup

In version 1 of the software, a chessboard interface was depicting the moves that the user was making, while each move was in actuality triggering the *Opera Game*'s (a classic Paul Morphy game) sequence of moves in the background. This game was chosen because of its slow, steady and tenacious build up of white's advantage. In principal, it would contradict the auditory outcome of the software if the guidelines about trying out fluctuations of the chess positions' dynamics were followed.

In Version 2 of the software, the interface showing the players' moves on the board would send the actual data to the algorithm and the musical outcome would be solely based on them. here you report what settings your program had, what headphones used etc

All sessions were carried out with the participants to the experiment wearing a closed type noise-cancelling Bluetooth headset (JBL LIVE650BTNC).

6.2 Data Analysis

The set of data gathered from the questionnaire are both qualitative and quantitative. Some answers were evaluated based on the author's inferences and observations about the final outcome of the algorithm, while another part of the data was used to examine the null hypothesis of the algorithm's musical outcome having no effect whatsoever on the users' perception of the chess positions' dynamics.

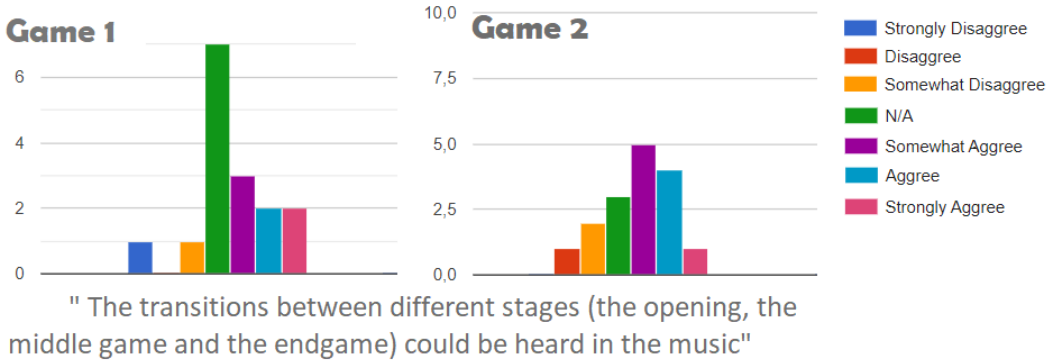


Figure 6.5: Answers to question/statement #3 for games 1 and 2

6.2.1 Game 1 and 2

Two out of five questions concerning game 1 and 2 were designed to evaluate the trustworthiness of the set of answers. Specifically, question/statement #1: "The music was affected each time I moved a piece" was expected to be leaning towards agreeableness for both games' sessions as the algorithm was indeed reacting to new input every time a move was played. The only difference between the two variations was that in one case the algorithm was interacting with the user's moves and in the other case, the prerecorded set of moves. An accumulation around "Not Applicable" or a tendency towards disagreeableness was expected concerning question/statement #3: "The transitions between different stages (the opening, the middle game and the endgame) could be heard in the music". That is because the participants were essentially being asked to provide feedback for a function of the algorithm that does not exist. This question was formulated to test if the interviewees were biased towards providing positive feedback to the project, but also because this is a function of the algorithm to be implemented in the future.

The function of the algorithm concerning the sonification of the act of capturing a piece has to do with manipulating the tempo, similar to a slow motion effect. This very subtle change in the sonification process, makes question/statement #2: "When a piece was captured on the board I could hear it in the changes in the music" more about a provision of a supplementary insight with regards to defining a "meaningful musical commentary". Thus, this question was also deemed as not a crucial one to extract strong inferences about the project's hypothesis.

t - testing

The questions/statements that made up the strongest possible inferences out of the experiment, were entries #4 and #5 concerning the algorithm's ability to musically express the disruptions of the positional balance and its ability to reflect the quality of the moves played on the chessboard (correct, inaccuracy , mistake, blunder).

The evaluation of the set of answers provided to the above mentioned entries to the questionnaire was based on the repeated measures *t*-testing procedure for two related samples, which is used to determine if two sets of data (interacting algorithm feedback / non-interacting algorithm feedback) are significantly different. If the *t* value falls beyond the cut-off threshold $\alpha = 0.05$ (indicating that in 5 out of 100 iterations of data collection the null hypothesis would be true) drawn from the *t*-table that shows the critical values of the *t* distribution, **then the null hypothesis can be rejected**. That would mean that there is a significant difference of the data provided for the interacting and the non-interacting algorithm.

To do that, the Likert-type data had to be converted to numeric values (from [strongly disagree, ..., strongly agree] to [1, 2, ..., 7]) so as to calculate the formula:

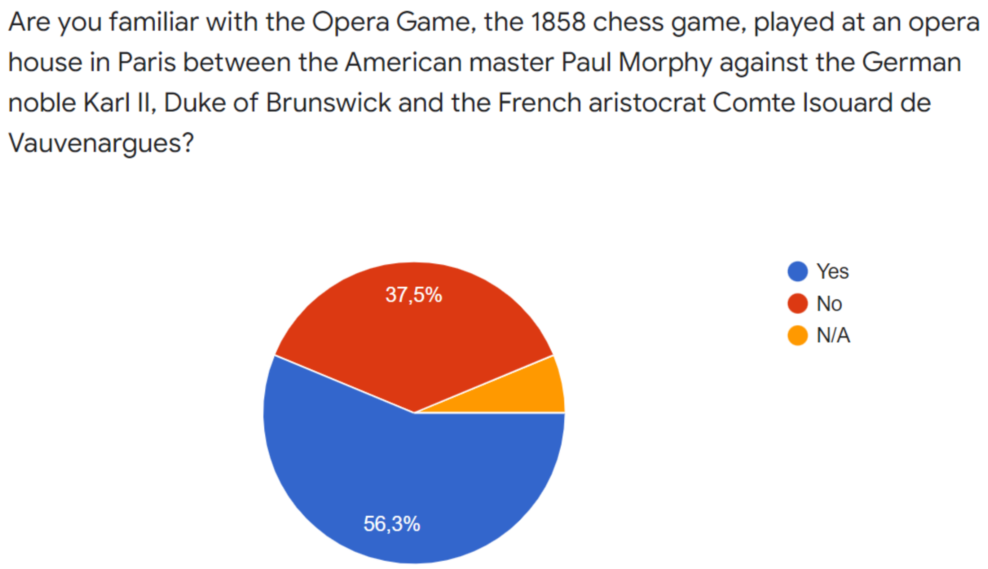
$$t = \frac{\overline{X} - \mu_{expected}}{\sqrt{\frac{s^2}{n}}}$$

where:

- The Null Hypothesis $H_0 = 0$ for no effect of the condition and the Alternative Hypothesis $H_A > 0$ for the null hypothesis being rejected
- The cut-out threshold $\alpha = 0.05$ for evaluating the null hypothesis
- $\mu_{expected} = 0$ when comparing with the null hypothesis
- n =number of subjects
- degrees of freedom $D_f = n - 1$
- mean score difference $\overline{X} = \frac{\sum_1^n [Score_n]_2 - \sum_1^n [Score_n]_1}{n}$
- variance of the sample $SS = \sum_1^n (D - \overline{X})^2$ for D being the difference of scores
- population variance $s^2 = \frac{SS}{D_f}$

6.2.2 The Opera Game

The next part of the questionnaire refers to *Bard*'s musical rendering of chess master Paul Morphy's famous "Opera Game". The purpose of this question ("to what degree do you feel that the sonification was musically describing the dynamics of the game's positions?") was to draw inferences about the software's ability to musically match the sensations that have already been evoked from a known and well-studied game. For that reason, the participants' answers who were not familiar with the particular game were filtered out from the evaluation.



Are you familiar with the Opera Game, the 1858 chess game, played at an opera house in Paris between the American master Paul Morphy against the German noble Karl II, Duke of Brunswick and the French aristocrat Comte Isouard de Vauvenargues?

Figure 6.6: Pie Chart of the sample's familiriaty to the *Opera Game*

6.2.3 Chess Ratings & Music Sophistication Index

The interviewees provided feedback about their chess skills based on their ELO ratings in FIDE, chess.com and lichess.org. These ratings were then categorized in six different levels of skill where 1 stands for "novice" and 6 for "master". The sum of ratings for each platform divided by the population of answers provides the mean average of the sample's category of level of skill.

The official webpage for Goldsmiths Musical Sophistication Index (Gold-MSI) [10] provided the questions with regards to the participants' perceptual abilities, their musical training, but also the "emotions factor" which covers behaviours related to emotional responses to music. The GMSI Scoring App was used to calculate the mean average of these categories of musical sophistication , as well as the general music sophistication index of every participant.

6.3 Results

6.3.1 Interpretation of feedback on question/statements #1, #2 and #3

As mentioned in section 6.2.1, question/statement #1 was expected to draw feedback leaning towards agreeableness for both versions of the algorithm, due to the fact that both versions were actually outputting new musical data with every triggering of the move counter. However, the overall assumption about the sample's feedback for game 1 points to uncertainty, as shown in figure 6.2. On the contrary, the overall feedback on question #1 about game 2 suggests a clear comprehension of the music changing after each move.

The results in figure 6.4 about question #2 are -as expected- indecisive. This is because ,in game 1, the players' captures and the prerecorded game's captures could either coincide or not. That said, the results show a deviation between the two sessions with the participants being slightly more agreeable to the statement about the interactive version of the algorithm.

As shown in figure 6.5 about question #3, both graphs form a normal distribution - even with a population sample of 16 persons- where game 1's peak is drawn at "N/a" and game 2's peak at "Somewhat Agree". As explained on section 6.2.1, this question's purpose was to indicate the possibility of the participants being positively biased towards the project as there is no function of the algorithm that takes the stages of the game into account. Although there was an occurrence of a participant stating with a written comment that "*[...] and there was a difference from opening to end game*, the fact that the answers were normally accumulated around the "N/a" area suggests that users provided reliable data.

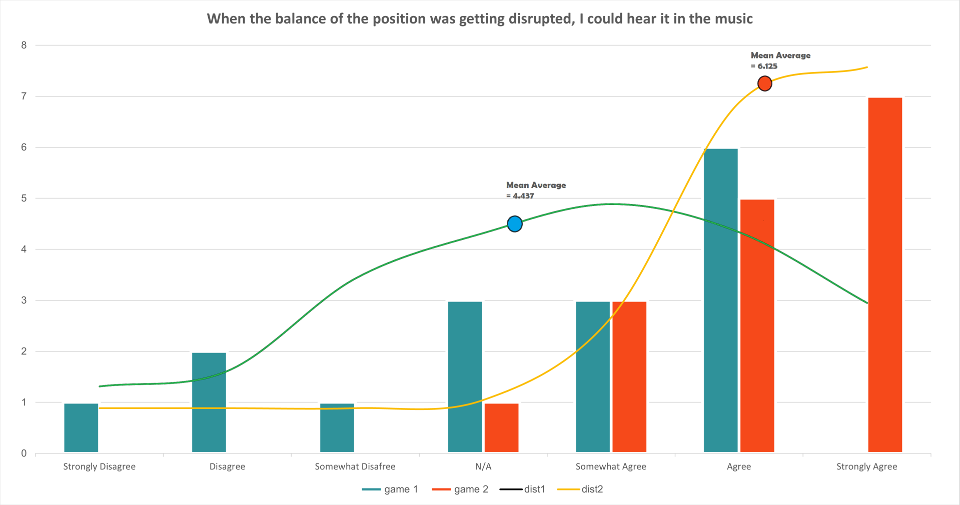


Figure 6.7: Answers to question/statement #4 for games 1 and 2

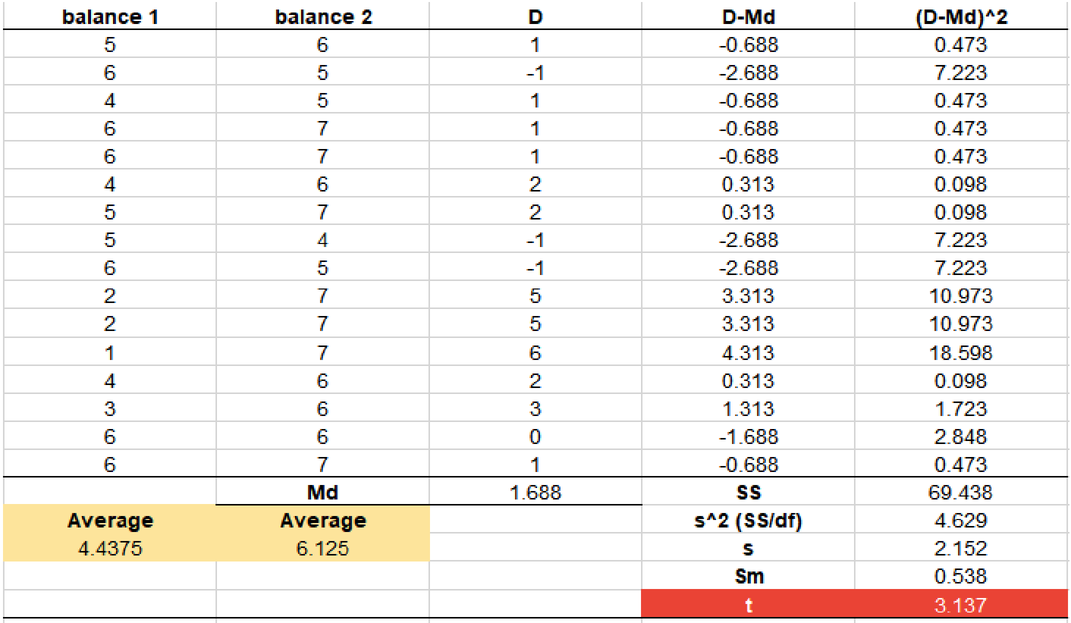
6.3.2 Comparing feedback on questions #4 & #5

In order to evaluate and compare the feedback on question/statements #4 & #5, the most vital questions on this project's hypothesis, there had to be determined whether there is a significant statistical difference between the results. As shown in section 6.2.1, the calculation of the t value will either reject or prove the null hypothesis of no effect of condition which, in this case, is that the interacting and non-interacting versions of the algorithm have the same impact on the participants' perceptions of the sonification with regards to the chess positions' dynamics.

The tables in figures 6.8 and 6.11 show that the t value for question #4 and question #5 is $t_b = 3.137$ and $t_t = 3.954$ respectively. Referring to the t-distribution table for an alpha level of the t-test of $\alpha = 0.05$ and degrees of freedom $D_f = 15$, the critical value is $t_{cr} = 2.131$.

$$t_{cr} < t_b \quad \& \quad t_{cr} < t_t$$

The null hypothesis gets rejected, thus indicating that there is a significant difference between the participants' feedback for both games 1 & 2.



balance 1	balance 2	D	D-Md	(D-Md)^2
5	6	1	-0.688	0.473
6	5	-1	-2.688	7.223
4	5	1	-0.688	0.473
6	7	1	-0.688	0.473
6	7	1	-0.688	0.473
4	6	2	0.313	0.098
5	7	2	0.313	0.098
5	4	-1	-2.688	7.223
6	5	-1	-2.688	7.223
2	7	5	3.313	10.973
2	7	5	3.313	10.973
1	7	6	4.313	18.598
4	6	2	0.313	0.098
3	6	3	1.313	1.723
6	6	0	-1.688	2.848
6	7	1	-0.688	0.473
	Md	1.688	**SS**	69.438
Average	**Average**		**s^2 (SS/df)**	4.629
4.4375	6.125		**s**	2.152
			Sm	0.538
			t	3.137

Figure 6.8: t value calculation for question/statement #4 for games 1 and 2

6.3.3 Feedback on the *Opera Game* Sonification

Nine out of sixteen participants were familiar with chess master Paul Morphy's *Opera Game*. This question's purpose was to get an insight on *Bard*'s performance in a game in which people had already experienced a certain set of emotional effects (from previously studying it), so as to see if this method can appeal to their experience. As shown in figure 6.9, the mean average of the interviewees familiar to the suggested game falls beyond 7.5 when asked if the sonification was musically describing the dynamics of the game;s positions, in a scale from 1 (to a low degree) to 10 (to a high degree).

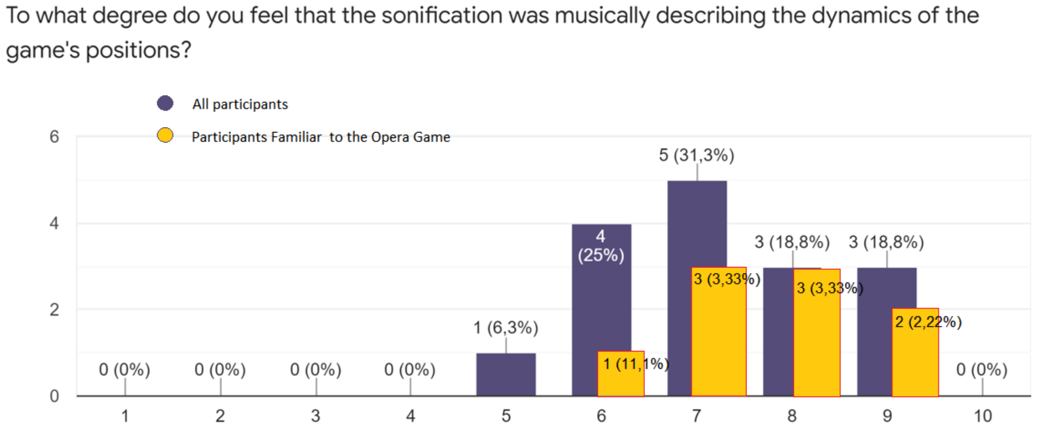

To what degree do you feel that the sonification was musically describing the dynamics of the game's positions?

Figure 6.9: From a low degree (1) to a high degree (10)

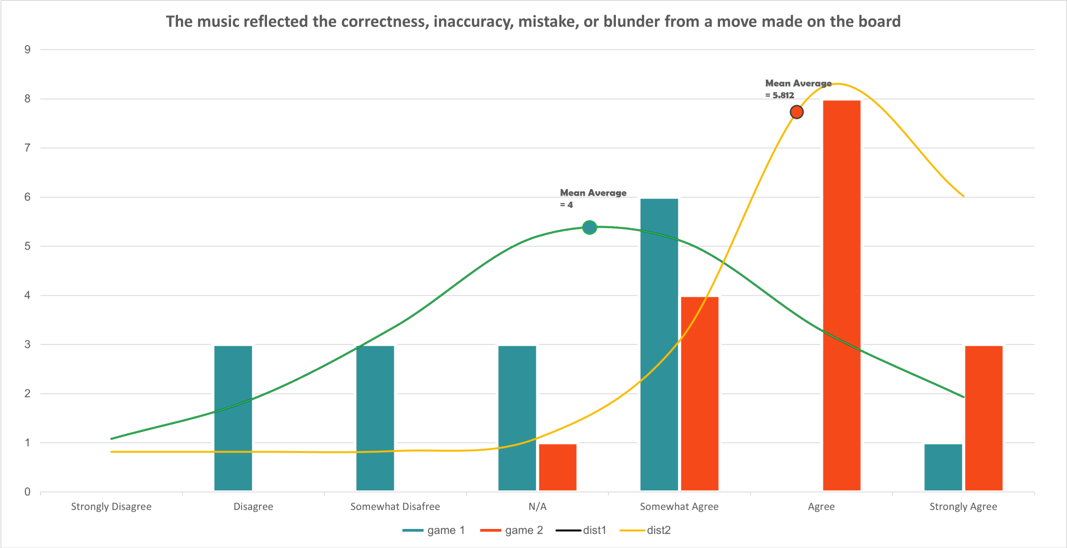
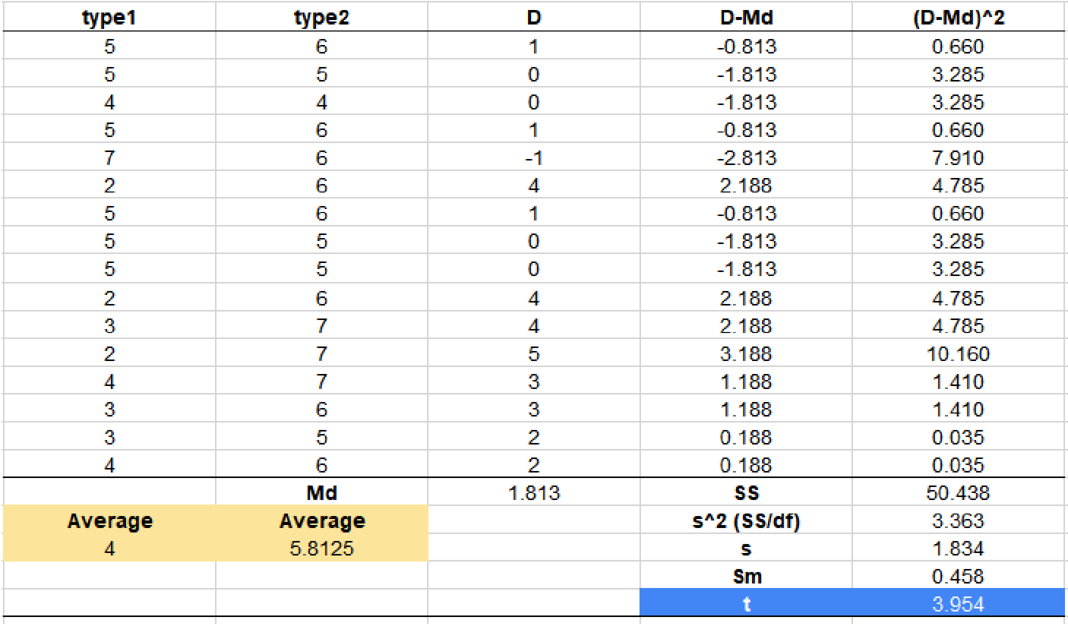


Figure 6.10: Answers to question/statement #5 for games 1 and 2

6.3.4 Remarkable Comments

1. " *First game somewhat made sounds depending on moves, second game for sure made sounds depending on moves and there was a difference from opening to end game*
2. *Little stressful. I'm not that good at playing, so have never played against myself before.*
3. "*It was really fun but I feel like the first game's music wasn't as accurate as the second game*"
4. "*Interesting concept, game 2 was following the game I was playing. I would like more variety on the sound*"
5. "*Both games the music did change, however in the first one I felt a disconnect between my own moves and the changes of music. Indifferent moves to the position produced fairly severe tune changes*
6. "*It was the first time I played a chess game with an interactive music carpet that matched, modern gaming sound experience, stimulating feelings of anxiety, balance or confidence. Brilliant.*"

"



type1	type2	D	D-Md	(D-Md)^2
5	6	1	-0.813	0.660
5	5	0	-1.813	3.285
4	4	0	-1.813	3.285
5	6	1	-0.813	0.660
7	6	-1	-2.813	7.910
2	6	4	2.188	4.785
5	6	1	-0.813	0.660
5	5	0	-1.813	3.285
5	5	0	-1.813	3.285
2	6	4	2.188	4.785
3	7	4	2.188	4.785
2	7	5	3.188	10.160
4	7	3	1.188	1.410
3	6	3	1.188	1.410
3	5	2	0.188	0.035
4	6	2	0.188	0.035
	Md	1.813	SS	50.438
Average	Average		s^2 (SS/df)	3.363
4	5.8125		s	1.834
			Sm	0.458
			t	3.954

Figure 6.11: *t* value calculation for question/statement #5 for games 1 and 2

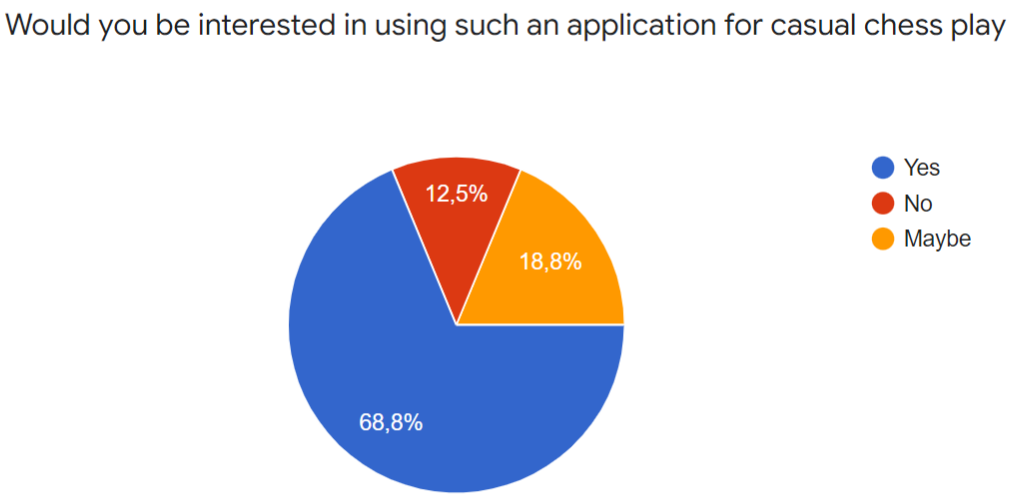


Figure 6.12: Revisiting the music-making out of chess concept

Chapter 7

Discussion

The results of the experiment conducted to test this project's hypothesis, that Bard can provide a meaningful musical narration to a game of chess, were as decisive as the experiment's design allowed. Trying to prove that a piece of music is meaningful is a controversy on its own. Even at program music's and symphonic poems' prime, it was very common to hand program notes (actual text, drawings, symbols etc.) out to the audience in order to help people contextualize the music they were listening to. In this project's case, the program note is the chessboard.

The fact that participants who were familiar with the opera game (and/ or have studied it) rated the sonification higher than those who were not, is quite encouraging. This means that people who had already formed an opinion about that game, who had already experienced a particular set of emotions about it, felt like revisiting this emotional spectrum.

An interesting observation while conducting the experiment was that the younger the participant the more prone they were to exploring the application's capabilities; although referencing the participant's age in the "results" section was not deemed necessary for the evaluation of the data received.

Also, the fact that the trap question/statement #3 came out with with a neutral feedback, indicates that the participants did not answer for the sake of complimenting the work from one hand, and that the initial hypothesis can be further supported from the other.

What can be said with certainty, judging by the evaluation's results, is that there is clear perception of interactivity. In both treatment conditions, participants felt like the music was changing with every move they played. Whether this change was in a direction that followed the game's dynamics is answered by the statistical analysis that questions/statements #4 & #5 were a subject of. Based on that, it comes out that there is indeed a significant difference between the two groups of answers.

Although this research may not clearly prove that the music outcome is meaningful or that bard really does make a non-verbal, story-telling musical narration on the chess games, it does show that the participants felt like the music of the actually interacting session was following their game. That fact alone, is very encouraging for the future of this project.

There is a set of musical assets that has not yet been explored by the method. Assets like interactive orchestration / sound design and black/white pieces musical distinction could improve Bard's voice and upgrade its epic-battle-singing skills.

Chapter 8

Conclusion

From the beginning of music's recorded history, musicians have always experimented with finding ways of automating the music making procedure. Chess, being a game almost as old as music itself has found its way into being a subject of study for musicians in many occasions throughout history. Today, with the available technology, besides the set of rules that chess provides, musicians have also access to the objective truth of the chessboard on command through chess engines, chess positions' calculators.

This project's hypothesis was that if this system of chess's rules is combined with the chess engine's calculations and then brought to synergy with a musical system, then we can expect a meaningful musical outcome with regards to game's dynamics.

The software developed to test this hypothesis, Bard, inputs the chess notation and the computer's evaluation and outputs a musical outcome which integrates melodies, rhythms, harmonic progressions and modalities.

Then, a variation of the software was implemented in which there was no actual interaction but only a sonification of a prerecorded game's moves. This variation was juxtaposed to the standard version of the algorithm in a related samples experiment which was conducted to test if there is a significant difference between the sample's experience with one and the other.

The evaluation of the participants' quantitative feedback showed that there is indeed a significant difference, while the qualitative data collected pointed towards the method having achieved its purpose.

Figures **4.2, 4.3, 5.1, 5.2, 5.3, 5.4, 5.5** have been designed using resources from **Flaticon.com**

Bibliography

[1] Adam Alpern. “Techniques for algorithmic composition of music”. In: *On the web: http://hamp. hampshire. edu/adaF92/algocomp/algocomp* 95.1995 (1995), p. 120.

[2] *Chess for Max*. `https://cycling74.com/forums/chess-for-max`. Accessed: /today.

[3] Lowell Cross. “Reunion: John Cage, Marcel Duchamp, electronic music and chess”. In: *Leonardo Music Journal* 9 (1999), pp. 35–42.

[4] D. Elder. *Pianists at Play*. Kahn and Averill, 1994.

[5] Patrik N Juslin and John Sloboda. *Handbook of music and emotion: Theory, research, applications*. Oxford University Press, 2011.

[6] Jonathan Kregor. *Program music*. Cambridge University Press, 2015.

[7] Edward Lasker. *The Adventure of Chess*. Dover, 1959.

[8] Bennett Lin. “Constraint and Craft in Schoenberg’s Coalition Chess”. In: (2012).

[9] Vincent Joseph Manzo. *Implementing Modality in Algorithmic Composition*. 2007.

[10] Daniel Müllensiefen et al. “The musicality of non-musicians: an index for assessing musical sophistication in the general population”. In: *PloS one* 9.2 (2014), e89642.

[11] EKPA Music Department Papageorgiou Emmanouil. “Chess Music: Software Development for the exploration of possibilities in music composition”. In: (2018).

[12] Dale E Parson. “Chess-Based Composition and Improvisation for Non-Musicians.” In: *NIME*. 2009, pp. 157–158.

[13] Arnold Schoenberg. *Theory of harmony*. Univ of California Press, 1983.

[14] Allen Strange and Gordon Mumma. *Electronic music: systems, techniques, and controls*. WC Brown Company, 1972.

[15] Achilleas Zographos. *Music and Chess: Apollo Meets Caissa*. SCB Distributors, 2017.

Appendix A

Snapshots of the the algorithm in Max

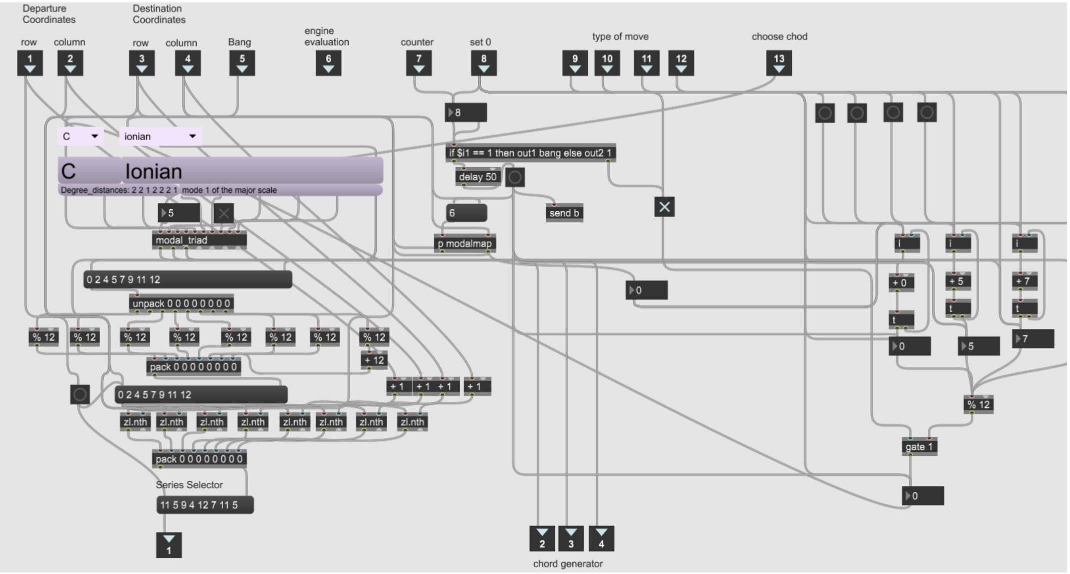
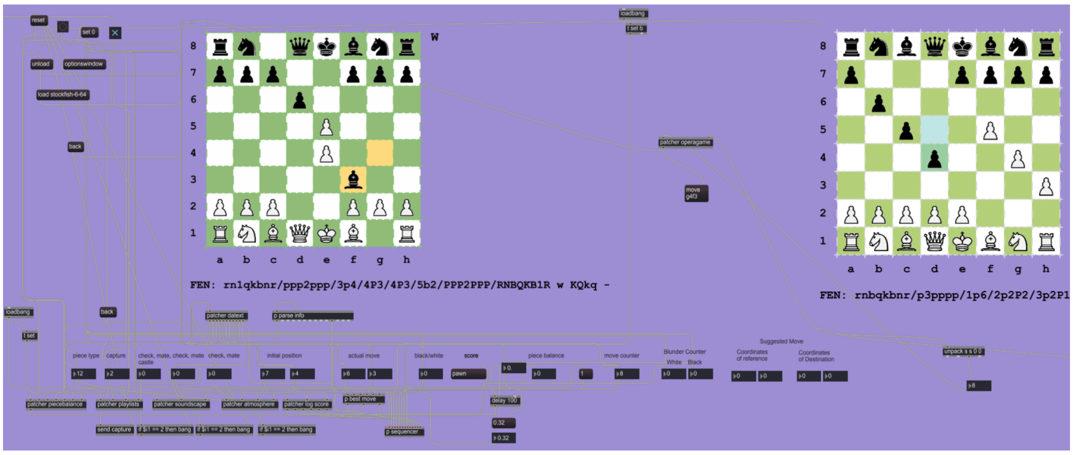


Figure A.1: *Bard*'s Series Selector Patch in Max



FEN: rn1qkbnr/ppp2ppp/3p4/4P3/4P3/5b2/PPP2PPP/RNBQKB1R w KQkq -

FEN: rnbqkbnr/p3pppp/1p6/2p2P2/3p2P1

Figure A.2: *Bard*'s false interaction variation in Max

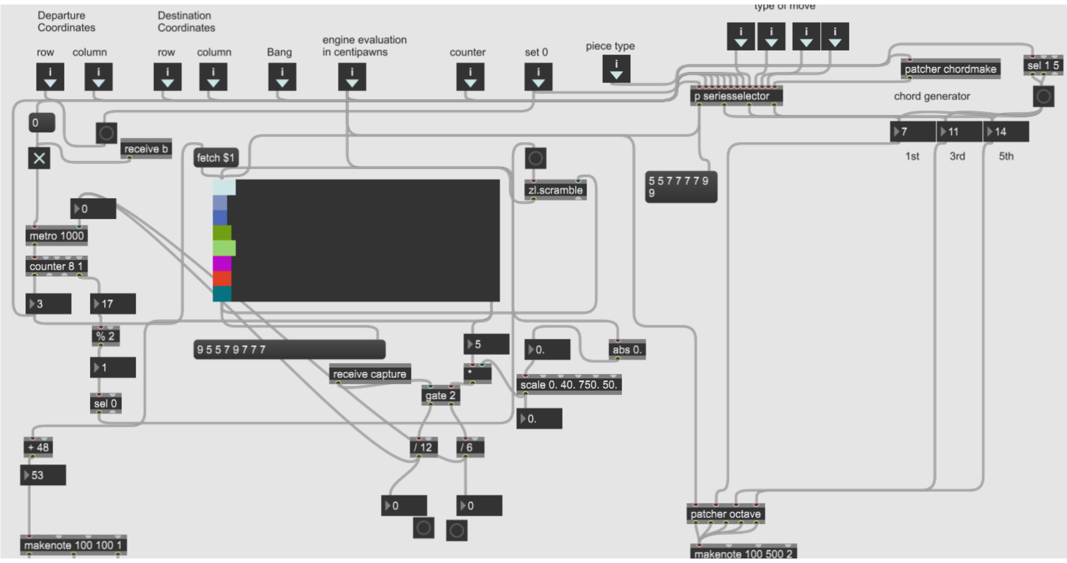


Figure A.3: *Bard*'s Step Sequencer Patch in Max

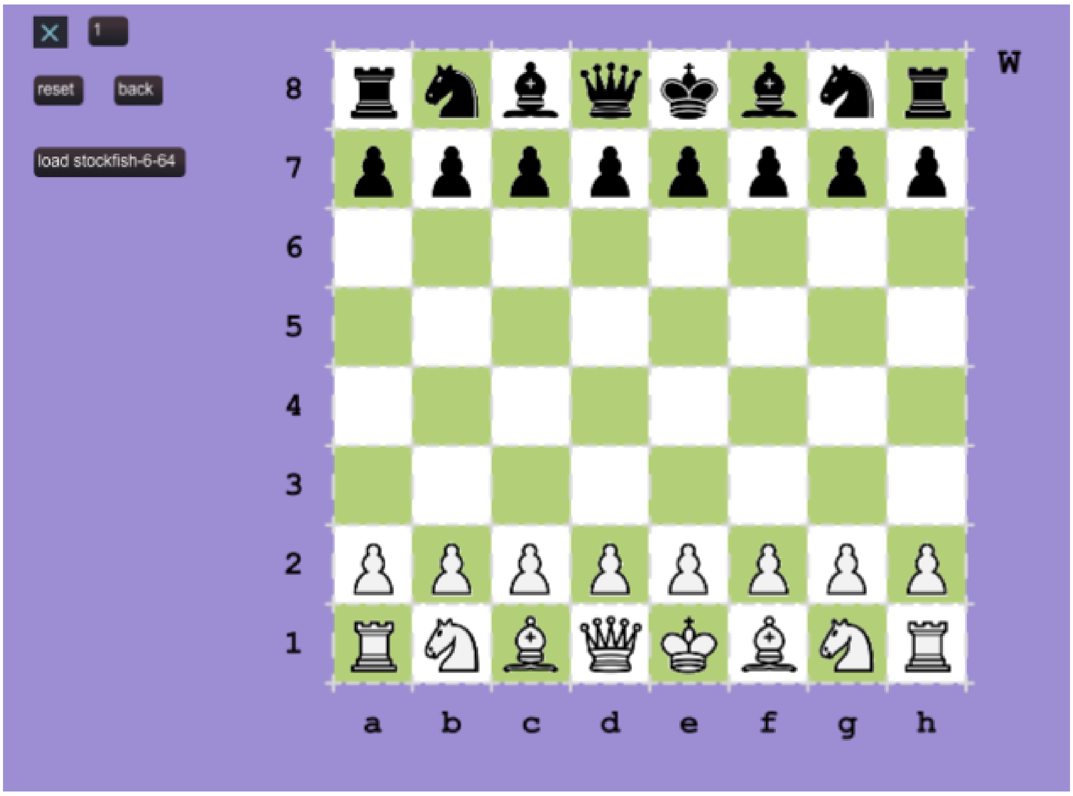
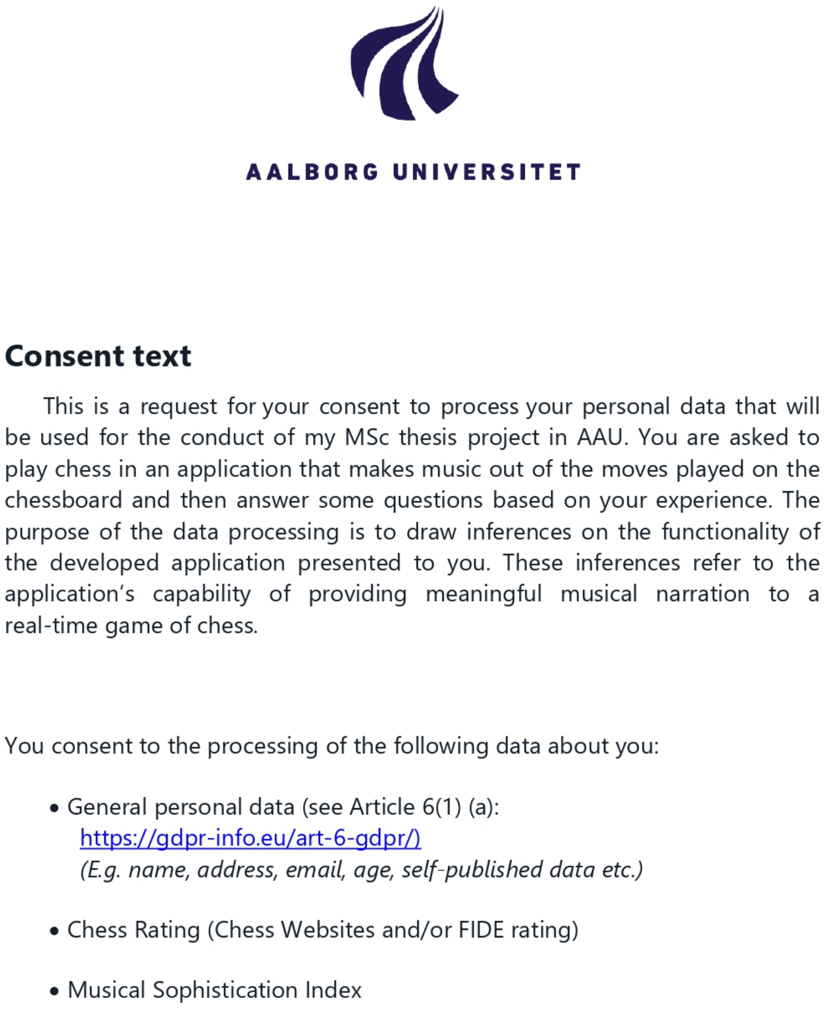


Figure A.4: *Bard*'s User Interface

Appendix B

Consent form



AALBORG UNIVERSITET

Consent text

This is a request for your consent to process your personal data that will be used for the conduct of my MSc thesis project in AAU. You are asked to play chess in an application that makes music out of the moves played on the chessboard and then answer some questions based on your experience. The purpose of the data processing is to draw inferences on the functionality of the developed application presented to you. These inferences refer to the application's capability of providing meaningful musical narration to a real-time game of chess.

You consent to the processing of the following data about you:

- General personal data (see Article 6(1) (a): https://gdpr-info.eu/art-6-gdpr/) *(E.g. name, address, email, age, self-published data etc.)*
- Chess Rating (Chess Websites and/or FIDE rating)
- Musical Sophistication Index

I, Emmanuel Papageorgiou, is the data controller of your data.Your data will be stored securely, and I will solely use the data for the above purpose. You always have the right to change your consent and stop the procedure at any time. If you wish to change your consent later on, you can reach out to epapag19@student.aau.dk, manopapageorgiou@gmail.com or call +45 91709482. The General Data Protection Regulation entitles you to obtain information that you find in the email you receive from me later on.

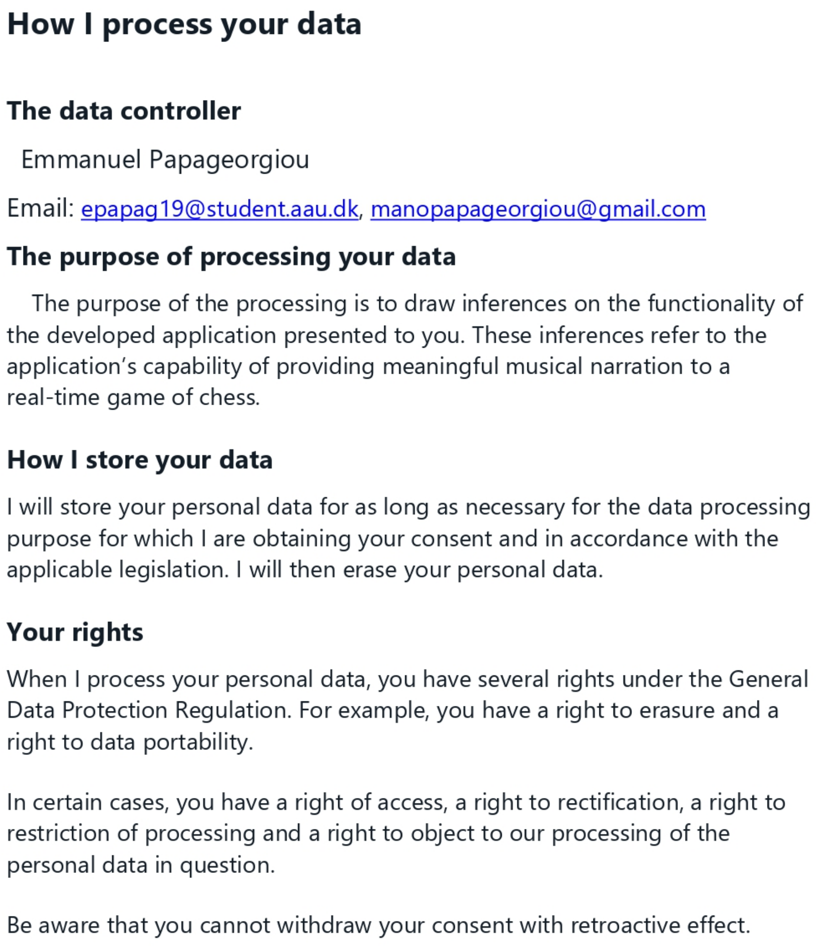
☐ I hereby consent to Emmanuel Papageorgiou processing my data in accordance with the above purpose and information.

Date:

Name:

____

Signature



How I process your data

The data controller

Emmanuel Papageorgiou

Email: epapag19@student.aau.dk, manopapageorgiou@gmail.com

The purpose of processing your data

The purpose of the processing is to draw inferences on the functionality of the developed application presented to you. These inferences refer to the application's capability of providing meaningful musical narration to a real-time game of chess.

How I store your data

I will store your personal data for as long as necessary for the data processing purpose for which I are obtaining your consent and in accordance with the applicable legislation. I will then erase your personal data.

Your rights

When I process your personal data, you have several rights under the General Data Protection Regulation. For example, you have a right to erasure and a right to data portability.

In certain cases, you have a right of access, a right to rectification, a right to restriction of processing and a right to object to our processing of the personal data in question.

Be aware that you cannot withdraw your consent with retroactive effect.

Do you want to complain?

If you believe that I do not meet my responsibility or that I do not process your data according to the rules, you may lodge a complaint with the Danish Data Protection Agency at dt@datatilsynet.dk.

However, I encourage you also to contact us, as I want to do me utmost to accommodate your complaint.

Disclosure to and from third parties

Your data (or parts of your data) may be transferred to Aalborg University Copenhagen.