2017

MUSE: a Music Sandbox Environment for Novices

Popa, Iulius

UNIVERSITY OF CALGARY

MUSE: a Music Sandbox Environment for Novices

by

Iulius Aurelian Tiberiu Popa

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTATIONAL MEDIA DESIGN

CALGARY, ALBERTA

DECEMBER, 2017

Abstract

Collaborative musical interfaces for novices allow people with limited to no musical education access to a “walk-up and play” group musical experience. However, research shows that the ease-of-use of an interface tends to be inversely proportional with its creative affordances. My research aims at increasing the level of creative controls in interfaces for novices without compromising the “walk-up and play” characteristic or negatively affecting the perceived quality of the musical output. This thesis proposes a new design paradigm – a “game of music” and presents the design, implementation, and evaluation of MUSE – a real time, collaborative musical interface for novices. Participants in user studies of MUSE found learning and using the interface easy, described the music as pleasant, and reported having creative control over the music output. Further research could explore this new paradigm and its potential to maximize creative affordances when designing new collaborative musical interactions for novices.

Thesis contributions

MUSE was accepted for oral presentation and published in the proceedings to the 12th Sound and Music Computing Conference in Maynooth, Ireland (Popa, Boyd, and Eagle 2015). MUSE was showcased at the Computer Science Industry Day in May 2015 at the University of Calgary and also ran as interactive exhibit in the Computational Media Design Gallery FM space between March 30th and April 9th 2015.

Table of Contents

List of Figures and Illustrations

Chapter 1 Introduction

Overview

Music in the digital age has become more readily available and abundant than ever. We are one click away from listening to any piece of music ever recorded. Music listening is one of the leisure activities that have seen a significant growth in recent years due to the pervasiveness of computer and mobile technology. According to Nielsen's Music 360 2016 study, 90% of Canadians listens to music, spending an average of 24 hours each week listening to their favorite tunes (Nielsen 2016). The time spent listening to music has increased 20% since 2013, while for teens it has gone up 40%. However, the number of people actively engaged in music creation and/or performance is drastically lower than that of music listeners, with some statistics reporting that less than 1% of U.S. adults played a musical instrument in 2012 ("Number of Adults Participating in the Performing Arts in the United States in 2012" 2016). Factors responsible for the low numbers of active participants in music making include a limited or absent music education background, or – in the case of people who do have some basic music training –a lack of interest in pursuing long-term music education (Cremaschi et al. 2015).

The intellectual, personal and social benefits in the development and wellbeing of those who pursue music education have long been supported by a large body of work (Hallam 2010; Schellenberg 2005). Children who study music show improved language development, literacy, numeracy, and overall increased cognitive performance (Schellenberg 2004). At a personal level, music education fosters emotional sensitivity, fine motor skills, and relaxation among others. Moreover, music education provides the skills and means to participate in group music making which has long-lasting benefits on the personal and social wellbeing of those actively engaged.

Participants in group music making report feeling a strong sense of belonging due to their active contribution to the group outcome, as well as increased self-confidence, improved social skills and overall deeper intrinsic motivation (Hallam 2010). The self-determination theory supports these reports in that the intrinsic motivation of individuals stems from fulfilling three psychological needs: *competency* (i.e. controlling the outcome of an activity and experiencing mastery), *autonomy* (i.e. a universal urge towards agency of one's life), and *relatedness* (i.e. the need for interaction, connectedness, and belonging) (Deci and Ryan 2002). Group music making is a great facilitator in fulfilling these needs that give one a sense of wellbeing and happiness. The sense of wellbeing is elevated in participants in a group activity due to the "emotional contagion" of positive, happy feelings experienced among the group members (Hatfield, Cacioppo, and Rapson 1993).

Although the benefits of group music making are evident, traditional music education in the Western tradition requires long-term commitment and practice to attain a level of musical proficiency allowing one to participate in group music activities. In the absence of music education, for most music consumers who otherwise love music – but only interact with it passively through the act of listening – the experience and benefits of music participation in general and group music making in particular are usually out of reach.

Apart from bringing music to us closer than ever, ubiquitous computer technology has also facilitated the development of collaborative musical interfaces providing non-traditional means through which novices can experience to some degree performing, creating, and collaborating musically without the need for traditional music training or education. Computer-based collaborative musical interfaces for wide audience draw on to the users' multi-modal digital literacy to foster creativity and collaboration, and ultimately to encourage formal music training

(Kylie et al. 2011). It has been suggested that computer-mediated games enhance music education and provide a doorway into formal skill development and music participation (Cassidy and Paisley 2013).

Collaborative Musical Interfaces for Novices

Research on collaborative musical interfaces for novices shows that the "walk-up and play" characteristic (i.e. short time-span to a positive participatory experience) takes precedence when designing for novices (Bengler and Bryan-Kinns 2013; Cassidy and Paisley 2013; Hansen, Andersen, and Raudakoski 2011; Blaine and Fels 2003; Weinberg 2003; Jordà 2003). Moreover, it has been suggested that the challenge in designing for novices is handling the apparent trade-off between:

1. A low, entry-level skill allowing novices quick access to positive collective experiences, and
2. Creative freedom leading to opportunities for musical skill and mastery with the interface.

A low, entry-level skill that leads quickly to positive user experiences is a predominant feature of everyday games and frequently found in other collaborative interactions such as tabletop and multiplayer computer games. Collaborative musical interfaces that focus on this feature in their design usually do so at the expense of available musical controls.

Creative freedom – generally associated with musical instruments – does provide users increased opportunities for long-term engagement and skill development but usually requires prior musical knowledge or lengthy and steep learning curves, which prevents novices from having positive experiences and compromises the "walk-up and play" requirement.

Ease of Use vs. Creative Affordances

The ease-of-use and the creative freedom tend to become mutually exclusive in applications where the design approach is deeply rooted in either the game domain or the music domain. As I discuss in detail in Chapter 3, there are two predominant approaches to designing collaborative interfaces for novices:

1. "Game-oriented designs" that focus on short time spans towards positive novice experiences similar to most casual games and video games.
2. "Music-oriented designs" that focus on creative affordances towards freedom of musical expression similar to most traditional and non-traditional musical instruments.

I believe that it is possible to design collaborative interfaces for novices that feature increased creative affordances without compromising the "walk-up and play" characteristic. In this thesis, I propose a new design paradigm, that of a "game of music" that draws on the particularities, limitations, and expectations of novice users within a live, collaborative music context. The focus of this thesis is the design, implementation, and evaluation of a working prototype of a computer-mediated application for collaborative music creation and performance in real-time for novices, named MUSE. MUSE stands for MUsic Sandbox Environment, and aims at providing novices a musically safe, "designated" area for creative collaborative play within the larger music universe playground of the Western music tradition.

Thesis Objectives

The objectives of this thesis are:

- To present the design and implementation of a computer-mediated application for collaborative music creation and performance in real-time for novices.
- To provide the research background context for the discussion and relevance of this new application.
- To present the evaluation results of a number of user studies and discuss the proposed design in the light of these results.
- To discuss other uses of this platform and potential future research explorations based on this design.

Thesis Structure

This thesis comprises six chapters as follows: Chapter 1: In this chapter, I present my research hypothesis and the motivation for pursuing it, and indicate the general direction in which I will orient the discussion. In Chapter 2, I start by examining the music literature through a "game" lens and present an overview on the use of chance or indeterminacy in music creation and performance. I then place MUSE as a musical composition within this context, and end with a discussion on a number of approaches to music indeterminacy and how they informed the design of MUSE. I also explore the "collaborative musical experiences for novices" domain, identify common design approaches, and propose the "game of music" paradigm as a foundation for achieving my goals. In Chapter 3, I examine the idiosyncrasies of the music and game domains, discuss the "game of music" paradigm, present the game design methodology used, and provide an in-depth overview of the rules, game components, and game-play of MUSE. In Chapter 4, I present the technical and implementation details of MUSE, describe the system design in relation with game components and user actions at interaction level. I also mention a number of system design challenges I encountered during MUSE's implementation. In Chapter 5, I present the results of two studies evaluating MUSE's proposed design goals, discuss MUSE in the light of the study results, and reflect on ideas for future explorations. Appendix A lists the web links to videos of MUSE and Appendix B contains the documents and results of the user study.

Chapter 2 Background

Within the “game of music” paradigm, MUSE is as much a musical composition as it is a participatory experience. As a musical composition, MUSE was informed by, and builds upon a number of approaches to composition and performance from the music literature that make use of chance, or indeterminacy. The Music Context: Chance section in this chapter presents a selected number of compositions in which aleatoric methods are employed in their creation and/or realization. As a participatory experience, MUSE is an interactive game of music for novices situated within a computer-mediated context. In the Collaborative Musical Experiences for Novices section I discuss several commercial and academic projects exploring real-time collaborative music making experiences and interfaces.

Music Context: Chance

In a general and traditional sense, a music composition is a set of instructions in the form of a musical score meant to produce a determined, fixed musical result. Although the rendition of the work is unique for every musician who performs it, the differences between renditions are usually subtle, requiring an equally discriminating and educated audience. When a composer leaves one or more musical aspects to chance, the composition transforms into a probability space, able to generate a number of significantly distinct, but nonetheless related musical outputs. The higher the amount of musical elements left to chance, the larger the degree of musical variation between realizations or versions of a composition. In this section I present first, a number of ways in which chance has been employed in music creation and/or performance in works ranging from the fifteenth century to contemporary times and second, an examination of the ways the design of MUSE was informed by these approaches to chance in music.

Early music.

Though a wider spread of indeterminacy as compositional device emerged only during the twentieth century and evolved into what is commonly referred to as *aleatoric music* (Meyer-Eppler 1958) or *aleatory music* (Kennedy and Bourne Kennedy 2007), limited degrees of chance had been used as early as the 15th century. Johannes Ockeghem wrote *Missa Cuiusvis Toni* ("Mass in any mode"), a four-part choral work that can be sung in either the Dorian, Phrygian, Lydian or Mixolydian musical modes (Allsen 1994). The choice of performing the piece in a given mode is left to the musicians. The rendition of the piece is thus not limited to only one possibility, but four different ones due to the change of musical mode. During the 17th and 18th century, baroque music features the use of "basso continuo", a composition method and form of musical accompaniment in which the composer indicates a figured bass line in the score, that specifies the harmony (i.e. chords) upon which the accompanist can improvise. Figure 1 shows an example of a figured bass line: the numbering below the bass note indicates the chord on which the accompaniment improvisation should occur.

Figure 1: Figured bass notation in baroque music (Wikimedia Commons)

Although the improvisation was bound to the stylistic constraints of the period, no two accompaniments would be identical. Nonetheless, the voice-leading and the underlying harmony would always be consistent.

Musical dice games.

Musical dice games, or *Musikalisches Würfelspiel* are composition "games" for the layperson dating back to the 18th century. These games were usually designed by composers and offered a music system consisting of a set of rules allowing laypeople to "compose" their own musical piece. Johann Philipp Kirnberger designed the earliest documented musical dice game titled "*Der allezeit fertige Menuetten- und Polonaisencomponist*" or "*The ever-ready minuet and polonaise composer*", in 1757 (Nierhaus 2009). To create a "new" composition, a user would roll a die whose result indicated a particular musical variation to be used for the current time unit in the piece (i.e. bar). Therefore, the game consisted in a predetermined musical form plus a number of musical choices (mostly bars) for each particular time unit of the piece. The design of such a game required musical skill in preparing the musical choices in a manner that took into account all the possible combinations and the implications of the harmonic and voice-leading aspects. In his introduction to the game Kirnberger wrote: "Anyone who is familiar only with dice and numbers and can write down notes is capable of composing as many of the aforesaid little pieces as he desires." (Hedges 1978). Other dice games of well-known composers are: "*Einfall einen doppelten Contrapunct in der Octave von sechs Tacten zu machen ohne die Regeln davon zu wissen*" or "A method for making six bars of double counterpoint at the octave without knowing the rules" by Carl Philipp Emanuel Bach in 1758 and "*Table pour composer des minuets et des Trios `a l'infinie; avec deux dez `a jouer*" or "A table for composing minuets and trios to infinity, by playing with two dice", created by Maximilian Stadler around 1780. Although a number of dice games have been attributed to Haydn and Mozart, their authorship has been disputed. It has been suggested that their names were simply used to boost sales (Hedges 1978). Without strong evidence proving otherwise, musical dice games are still being published under their names. A more recent musical dice game is attributed to ragtime music composer Scott Joplin (c.1876-1917). "Scott Joplin Melody Dicer" was published in 1974 by Carousel Publishing Corp. advertising that anybody can "make over a million ragtime tunes" in no time. Similar to its 18th century counterparts, this game employs the use of two dice, a chart, and a number of cards containing musical fragments.

The design of these musical dice games relies on a systematic re-organizing of musical fragments resulting in a large number of possible combinations. Overall, chance affects only the combination of written out musical fragments within a fixed, predetermined musical form. Though the user does not require musical knowledge in order to create a new piece – other than being able to perform the resulting composition – the probability space of the system always produces a musical work that sounds stylistically correct and musically pleasing.

Modern Music.

Overview.

During the twentieth century, modern music composers begin implementing higher degrees of chance in their works, leading to a wider range of rendition possibilities for a given work.

The use of chance can be separated into two broad approaches: the first is concerned with the use of random procedures during the compositional process, usually referred to as "indeterminacy of composition" (Simms 1996). This approach employs random methods for the generation of one or more musical aspects such as musical material (e.g. pitch, duration, rhythm, timbre, etc.), musical form, orchestration, and instrumentation in order to produce a fixed, determined composition. The second approach describes works in which the composer leaves one or more musical aspects to be determined by musicians during the performance of the work – "indeterminacy of performance". This section focuses mainly on the second category, and without attempting to offer an exhaustive exploration of this topic, it does present a number of selected compositions using various approaches to chance that informed the design of MUSE. Figure 2 illustrates a four-set diagram, with each set representing an "and/or" collection of indeterminate musical aspects employed in the compositions examined in the Modern Music section. I joined related musical aspects using the "and/or" grouping in order to provide a clearer, simplified visualization of the distribution of compositions in this context.

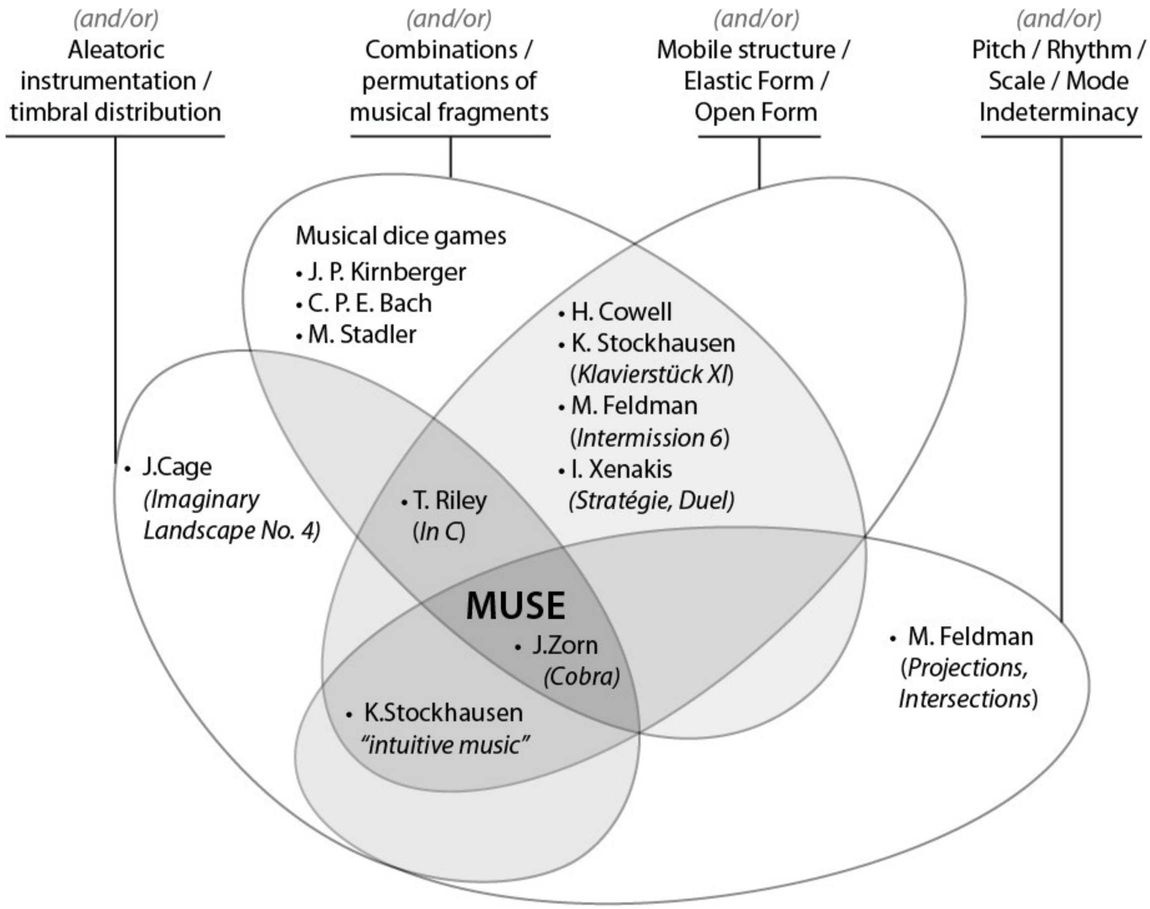


Figure 2: MUSE within a musical context defined by compositional approaches to indeterminacy of performance.

The diagram does not indicate the variation in the *degree* of chance of a particular category employed by individual compositions. Its purpose is to offer a broad visual depiction of the musical context within which MUSE is situated. I will present examples ranging from early compositions where only one particular musical aspect is left to chance to contemporary works where almost all possible musical aspects are determined during performance.

Henry Cowell is among the first modern composers to use chance in his works as "elastic form music". Examples include the *Mosaic Quartet* (1935, String Quartet No. 3), *Sound Form No. 1*

(1937) for five players, and *26 Simultaneous Mosaics* (1963). "Elastic form music" emerged within a dance context, as a means to provide choreographers with an efficient solution to adapting new dance pieces to music (Gagné 2011) allowing the expansion and contraction of melodic phrases, treating musical sections as "block units" and rearranging and /or repeating them thus altering the form and the duration of the piece (Griffiths 2001).

One of Henry Cowell's students was John Cage, who is considered a pioneer of both chance composition and aleatoric music. Cage is probably best known for his controversial, "silent" three-movement composition, *4'33"* written in 1952. Cage strips the piece of all possible musical aspects, except its duration, which is always 4 minutes and 33 seconds. The performer is not required to do anything. Depending on one's perspective, *4'33"* can be regarded as either an extremely determinate composition or a highly aleatoric one. From a performer's point of view, the "performing" of 4 minutes and 33 seconds of silence is identical no matter who plays the piece. Without any means for the performer to alter the music content of the work, the performance cannot be evaluated through its quality of rendition. However, from a listener's viewpoint, the incidental sounds occurring in the environment during the silent performance make for a unique listening experience every single time, placing *4'33"* among the most aleatoric compositions possible. Another interesting use of chance is found in Cage's *Imaginary Landscape No. 4* (1951) for 12 radio receivers. The score has a fixed form and indicates the succession and duration of specific radio events to be performed, such as frequency tuning, volume, and tone. The piece relies on radio broadcast as musical material. The indeterminacy of performance of this work is somewhat unusual, in that although each rendition is significantly different (due to the radio programs being broadcast at a given time), no indeterminate aspect is left for performers to control.

In 1951, Cage writes *Music of Changes*, which is considered to be the first instrumental piece composed entirely using chance. This works exemplifies the “indeterminacy of composition” approach (Simms 1996). Musical aspects such as sound events, durations, dynamics, tempo, and density of voices are each selected using chance. Cage uses 8 by 8 cell charts to store 64 entries for each of these musical aspects which are later used towards the creation of musical fragments. The selection of entries is based on random numbers produced through aleatoric methods indicated in the ancient Chinese text used for divination titled *I Ching* or *Classic of Changes* (Pritchett 1996).

Indeterminacy of performance.

Indeterminacy of performance is a characteristic of a musical work in which the composer leaves one or more musical aspects to chance, to be determined during the performance of the piece. Morton Feldman wrote a number of works taking this approach. In his *Intersections* and *Projections* pieces (1950 - 1953) Feldman uses chance for pitch and duration parameters by laying out the score on graph-paper indicating “time-units” on the horizontal axis and relative pitch level (i.e. high, middle, low) on the vertical axis. The choice of pitches and rhythms is left to the musicians. These works are among the first modern compositions for traditional instruments to use graphic notation as an approach to performance indeterminacy.

In his *Intermission 6* (1953) for one or two pianos, Feldman does indicate the pitches – consisting of fifteen very short musical fragments – but the form here is mobile, not only allowing the pianist to play the fragments in any order, but also not giving any indications regarding the repetition of fragments or duration of the piece. The complete instructions read: “Composition begins with any sound and proceeds to any other. With a minimum of attack, hold each sound until barely audible. Grace notes are not played too quickly. All sounds are to be played as soft as possible."

Similarly, Karlheinz Stockhausen uses a mobile structure in his *Klavierstück XI* (1956) for piano, in which the performer chooses to start with any one of nineteen fragments that are laid out on the page, and play through the fragments in any order untill a fragment has been played three times, at which moment the piece ends (Maconie 1976).

Terry Riley's *in C* (1964) is an example of an aleatoric composition that relies on the indeterminate juxtaposition of written-out musical fragments. *In C* is written for an ensemble of variable size and instrumentation and consists of a sequence of 53 musical fragments of varying lengths. The performance of the piece starts with all musicians playing the first fragment. The fragments have to be played in order, but the number of repetitions each fragment can be played is a choice made by the performer. Dropping out and listening is not only allowed but encouraged as indicated in the written instructions to the score. This leads to dynamic musical form and a virtually infinite number of possible performances due to the ongoing change over time of polyrhythms, polyphonic texture, and timbral distribution (Carl 2009).

Chance as probability space.

The higher the level of indeterminacy of performance in a composition, the larger and more diverse the probability space for the rendition of the work becomes. Compositions that give performers a high degree of choice can almost be conceptualized as *games*. While written out musical aspects in the score can be regarded as *game components*, in that they are clearly determined by the composer as discrete entities to be manipulated, the performance instructions (and/or the lack thereof) become the counterpart of *game rules*, acting as interaction facilitators between the performers and the composition (i.e. gameplay, in the game domain) (Brathwaite and Schreiber 2008). The rules thus provide the framework within which the musical experience is constructed using the given components.

Iannis Xenakis was largely critical of aleatoric approaches granting too much freedom to the performers, stating that “chance needs to be calculated” (Harley 2004). Xenakis approaches chance in some of his compositions through “game theory”, by implementing an external conflict into the performance of music. Examples include works such as *Duel* (1959) and *Stratégie* (1962) for orchestra. The composition is comprised of a set of musical modules such as modules for strings (e.g. short, sustained, and glissandi sounds), for percussion, for winds, and a silence module. The orchestra is divided into two groups, each one having its own conductor. A conductor starts by communicating to his group what module to play and proceeds to performing it. The other conductor chooses to play the next module as a counter-action, based on a table of points awarded for certain sequences of module combinations. The overall succession of modules follows a carefully designed table of probabilities, as designed by Xenakis, who used the point awarding approach to increase the likelihood of certain combinations of modules to occur while discouraging others. The performance concludes with a conductor “winning” the game by achieving the highest amount of points. In both *Duel* and *Stratégie*, the actual game “player” is the conductor. The musicians have no control over the musical modules being played or the order of events.

Karlheinz Stockhausen removes the role of the conductor and gives the performers full freedom over the indeterminate aspects in some of his compositions as detailed further. In the light of the analogy with games – due to the high probability space of indeterminacy of performance –

although theoretically the musicians are only "performing" the piece, from a participatory and psychological perspective they become active creators.

The availability of multiple options to choose from in a real-time performing situation engages the performers at a deeper psychological level, similar to players facing game decisions in real time. The weighing of available options, the anticipation of the overall impact of a musical action on the whole, the performing of that action followed by the evaluation of the effect that action had on the music and the performer, are aspects that significantly increase the involvement of the musician in the performance experience. The engagement of the musician deepens further when performing chance music in ensembles. The musical context becomes more unpredictable, since more than one musician participates in its building. A musician's action will now have an impact on the other musicians' choices, creating a motivational and inspirational continuum. Listening to the others becomes absolutely indispensable, as one's contribution is always part of a bigger musical whole. The performance dynamics – the interaction of each performer with the composition and the interactions that emerge among performers – are unique to every performance and thus responsible for producing original musical results every single time.

Stockhausen's *intuitive music* builds heavily on the psychological investment of the musician in the performance of ensemble music using chance. *Aus den sieben Tagen*, ("From the Seven Days*")* is a collection of fifteen texts for intuitive music written in 1968. The "score" contains only the written text; no musical materials are provided. The first text, titled *Richtige Dauern* ("Right Durations") for "circa 4 players" indicates:

> "*Play a sound. Play it for so long, until you feel that you should stop. … Stop when you feel that you should stop. But whether you play or stop, keep listening to the others. At best, play when people are listening. Do not rehearse."* (Stockhausen 1972).

The second text, *Unbegrenzt* ("Unlimited") for ensemble reaches even deeper into both the conscious and the subconscious of the performers, almost into metaphysical cosmology territory: "Play a sound with the certainty that you have an infinite amount of time and space." The performance context of intuitive music does not provide nor require any written out musical materials. The texts act as "super-charged" *rules*; their interpretation by the players produces instant, unique musical material and therefore the entire musical experience. Although the audience may perceive the composition as highly aleatoric, Stockhausen states that intuitive music "is not indeterminacy, but intuitive determinacy!" He argues that the musical result is not at all indeterminate, but "absolutely direct, from concrete experience" (Kurtz 1992), implying that the interpretation of the text by a player is not subject to chance, but highly determined by all the experiences that shaped that player's intuition up to and including the moment of performance. Whereas the philosophical statement of Cage's *4'33"* challenges the audience's perception on what constitutes music and what does not, Stockhausen's intuitive music compels the listener into reflecting on where chance in performance ends and concrete experience begins. Unlike Stockhausen's approach on using very simple rules to create a theoretically infinite probability space for the composition, John Zorn uses a more robust set of rules in his work *Cobra* (1984) to achieve a similar degree of indeterminacy of performance. *Cobra* is a "game piece" composition for an ensemble of variable size and instrumentation, includes a leader (i.e. *prompter*) and features a complex set of rules. As shown in the four-set diagram in Figure 2 this work features the most indeterminate aspects of performance compared to the other compositions examined so far. *Cobra* is similar to Xenakis' *Duel* or *Stratégie* in that the rules enable emergent gameplay during performance. There are however some important distinctions as well: whereas Xenakis' game rules (enforced by the winning condition) ensure a calculated probability for certain sequences of musical modules to take precedence, since in Cobra there is no winner, the distribution of events depends exclusively on the personal preference of the decision maker. Moreover, unlike *Duel* and *Stratégie* where only the conductors are allowed to make game choices, in Cobra, along with the "leader", every musician of the ensemble can affect to a high degree the gameplay of the performance. Zorn describes this performance context as both "democratic", due to the performers' involvement at gameplay level, as well as "fascist", due to the presence of a leader free to allow or thwart the choices made by performers.

Though a thorough description of the rules of Cobra is beyond the scope of this paper, its overall concept significantly informed the design of MUSE and therefore warrants a short overview. The "score" of Cobra consists of a page indicating all the actions allowed in the game, which correspond with events of significant musical impact.

The actual musical material is at the discretion of players. The performance unfolds as follows: a musician requests the leader (i.e. prompter) to allow a specific game event to take place using a bodily motion cue (e.g. hand, eye signals, pointing) associated with that event (as per the rules). Although multiple musicians may request events at the same time, the prompter has the freedom to choose what request to grant. This is done by facing a large card containing the symbol for that event to all the musicians. The subsequent lowering of the card constitutes a "down-beat" and indicates the triggering of the chosen event. Examples of events include: "Pool" requested by pointing one finger to the mouth, indicating that performers not currently playing can enter performance while the ones performing either stop or change significantly whatever they are playing. “Cartoon trades” is an action where “single notes/sounds/events [are] passed extremely quickly from one player to another” (Brackett 2010). This is achieved by a musician making eye contact with another musician while “passing the cartoon trade”. If a performer wishes to “lock in” whatever the ensemble is currently playing, he/she can request the “sound memories” event, requesting the leader to indicate all players to memorize whatever they are playing at that moment for later recall (with a maximum of three “sound memories” allowed). A player can dramatically impact the gameplay by becoming a guerilla (with the acknowledgement of the prompter). As guerilla, a player has the freedom to do almost anything they wish. If two more players join the guerilla, they become a squad and take the control of the prompter for a limited time. The end of performance is also an event requested by players, and executed once the prompter lowers the card as either: sudden stop, six to ten seconds “Coda”, or “hold and fade” endings.

Computer Music.

The 80s brought about technological innovations that opened up new possibilities for professional and amateur musicians. In 1983 the MIDI communication protocol was standardized (“MIDI History” 2016). MIDI (musical instrument digital interface) digital instruments and computers could now communicate with each other using sixteen dedicated channels through which they transmitted music parameter data such as pitch, velocity, and duration as well as control change parameters such as volume, sustain, and panning. Several connected MIDI devices share the same timing clock allowing for synchronized playback between them. Since a MIDI file does not store the sounds themselves, but information consisting of parameter values of timed sound events, a musical score takes as little as a few kilobytes of computer disk space to store. That same year, Yamaha released the DX-7 ("Yamaha Products History" 2017), the first commercially successful digital synthesizer keyboard which featured a MIDI interface, followed a year later by Apple releasing their MIDI compatible Macintosh computer. Software for MIDI-enabled computer music creation became available in 1986 starting with C-Lab's *Notator*. Although most software was geared towards enabling the recording of musical ideas as a sequence of event parameters (i.e. music sequencers), MIDI offered composers and performers the ability to control music in real-time.

The Intelligent Music company lead by Joel Chabade developed a computer program designed for interactive composing and performing using the MIDI protocol. During 1986-1987, by leveraging the computer's graphic display, mouse, and MIDI interface, Chabade together with John Offenhartz, Antony Widoff and David Zicarelli designed *M* and *Jam Factory*, tools that provided an environment for real-time music interactivity (Zicarelli et al. 2017). Whereas other music sequencer software only allowed the recording of predetermined musical events, M and Jam Factory gave the composer/performer the ability to creatively modify those musical events in real time by means of controlling a large number of variables. In essence, *M* is comprised of four sets of note events that loop simultaneously. Thus, the composition process consists of planning out these "building" blocks which act as musical material to be not only rendered but modified in real-time. During playback, these patterns loop continuously while the user changes various note event parameters. Zicarelli explains that "in designing M and Jam Factory, various assumptions – effectively musical decisions – were made in order to promote a user's ability to "get at" the music and change it interactively while maintain significant control" (Zicarelli 1987). M employs indeterminacy in higher level controls over the order of pitches, note durations and note velocities. Whereas in a regular software sequencer the changing of the order of pitches in a pattern would require stopping the playback, re-organizing the pitches in the preferred order, and playing back the result, in M this is achieved with one mouse click by selecting an ordering from the following options: Original Order, Cyclic Random, and Utterly Random. In M, musical change and variation is achieved by real-time control of the degree of randomness of any parameter. The change is applied to the block immediately and the user receives instant musical feedback of his action. Apart from incremental change of a given parameter, since each parameter can be stored in one of six available configurations (i.e. presets), the user can quickly and drastically change the musical character of a voice by switching between contrasting configurations of a given parameter. Furthermore, due to the MIDI's low memory footprint, music snapshots of the current state of all voices and parameters (similar to Zorn's "sound memories") can be stored for recall at a later time. Control-wise, M receives mouse input "gestures" through the graphical interface as well as MIDI messages triggered through pre-assigned keys on a MIDI keyboard. From an interaction perspective, in M the line between composing and performing is blurred by the affordances of the interface. Zicarelli emphasizes that M is more than just a music production tool, and that users reported having "a great time with your program" more than "I'm making great music with your program". He goes on to say that the process of discovery and the pleasure of performance along with the music being produced with M are what provide enjoyment to the user.

In 1990, Zicarelli developed *Max*, a software for interactive music based on the research work done by Miller Puckette at IRCAM in the mid-1980. Since then, Max has evolved into a powerful tool for creativity by harnessing the power of the computer to provide a modular environment for creating interactive applications embedding digital audio processing, sound synthesis, and video processing, controllable through MIDI or other control protocols. Other approaches to computer music interactivity include live-coding, or “on-the-fly programming”, where creative individuals use programming languages such as *ChucK* (“ChucK” 2017) and *SuperCollider* (“SuperCollider” 2017) to create and transform digital and synthesized sound during performance by typing instructions in a laptop’s command-line prompt.

Tools like Max have also been used in live performance to augment the capabilities of acoustic instruments. The combination of dedicated software, communication protocol standards and the processing power of the computer has led to the design and creation of numerous novel electronic musical instruments. Since 2002, musicians and researchers of interactive music have been sharing their instrument designs at the international New Interfaces for Musical Expression conference (“NIME” 2017).

Conclusion.

In the previous section I presented selected musical works using chance ranging from early music examples, through mid-twentieth century aleatoric compositions to more recent game pieces. Apart from the obvious increase in the level of chance employed in these works over time, the compositional focus also shifts from the musical result as evaluated by the audience, more towards the participatory experience of the performers. Works such as the intuitive music of Stockhausen and Zorn’s game pieces act as open-ended, socio-musical systems where the real-time interaction among performers within the rules of the work *are* the piece just as much as the actual musical result.

The works examined so far provide the musical context for situating MUSE as a musical composition. MUSE sits at the intersection of approaches to indeterminacy of performance as shown in Figure 2 due to the implementation of these compositional methods in the design of my application. The grouping and situating of works in this diagram does not take into account the *degrees of chance* employed by either selected compositions or MUSE. However, MUSE does implement indeterminacy of performance for each category of musical aspects indicated in the diagram. The players (i.e. performers) of MUSE are given full control over the distribution of musical instruments across the musical range, virtually unlimited access to combinations of musical fragments, and full control over the musical form. However, since the target audience of MUSE comprises of users with no musical training, chance is controlled to a certain degree (by design) in regards to these aspects and to a slightly higher degree at pitch, rhythm, dynamics, timbre, and musical scale level.

Similar to many of the compositions presented here, MUSE is accompanied by a number of interaction/performance rules which lay some boundaries for what is aesthetically, stylistically, and technically possible. Nonetheless, the physical implementation of MUSE, adds another layer of affordances as well as limitations. These aspects will be discussed in greater detail in GUI and game components and Gameplay sections.

MUSE shares some important similarities with the early musical games: first, the *musically pleasing* and *novice-friendly* characteristics of musical dice games – represented by a system always producing a consistent musical output regardless of the user's musical education – is also shared by MUSE, which presents itself as a musically "safe" environment allowing experimentation with no need for composition skills and without concern for wrong notes. Second, the *gameful* character of using chance to determine some aspect of the musical output is implemented in MUSE as affordances whose type of outcome is presented clearly to the user, though the exact rendition of the result is left to chance.

The re-arranging of pre-determined, short musical entities in new configurations found in Feldman's *Intermissions* is shared by MUSE in the "modular" approach taken in its design. The constituent components of MUSE (i.e. musical blocks) can be added, removed, arranged, and even modified within the limitations imposed by the system.

Terry Riley's *In C* informed the minimalist musical style used in creating MUSE based on its design goal of appealing to large audiences within a Western music tradition. This led to the employment of a musical style broadly characterized by tonal harmony, steady metric pulse, repetition of motifs, and gradual change (Bernard 1993). However, MUSE also affords musical changes that are not necessarily characteristic to the minimalistic style, as discussed in the Gameplay section.

Some parallels can also be drawn between the overall approach to MUSE and Stockhausen's works employing chance, in that MUSE is a system of change-affording interactions focused on the internal processes of the music being created from the user experience perspective. MUSE allows users to observe and affect the musical changes as they occur over time, to listen to other players, to form mental relationships between their action and musical "reaction", to reflect on emerging musical trajectories, and to anticipate how personal contributions alter a trajectory, among others. Moreover, MUSE did not expand into a tool for real-time composition and performance for external audiences. Its main focus is not on the resulting musical output as evaluated by outside observers, but on a meaningful novice user engagement with music, in a real-time collaborative experience.

A degree of similarity with Xenakis' approach to calculated chance based on probability is found in MUSE in the controlled randomness of certain musical parameters through design and implementation. Since MUSE was created for novices, considerable effort has been made in anticipating most possible game states, and while the game allows users to create unique, emergent gameplay and musical experiences, a number of musical parameters are either concealed from the user's direct control, or are randomly generated according to predetermined probability rules.

Lastly, John Zorn's game pieces informed the design of MUSE in two important aspects:

1. Game rules that allow any player the possibility to become the ensemble leader for a certain amount of time. In MUSE this translates in the time-limited, turn-taking game mechanic, where although each player is still an ensemble member, once during their own turn, they can control, if they wish, the entire course of the music being created, not only their own contributions.
2. Simple cues matched to complex, significant, and often sudden musical changes. MUSE provides its players with "one-to-many" actions that translate to several musical parameters being changed at only one touch of a gamepad button.

At system level, MUSE shares a lot of features with Zicarelli's M software, such as the use of repeating loops of notes, using degrees of randomness to affect progressive and sudden change and musical variation at the three main sound parameter levels: pitch, duration and velocity. At the time of designing MUSE, I had already been using Max for simple projects but I was not aware of the existence of M (although it had been designed almost 30 years before). The most important aesthetic aspects M and MUSE have in common are the real-time nature of the interaction, the "get at" the music approach in the design of the interface, and the focus on the quality of the interactive experience rather than the musical end result as perceived by external audiences. This section concludes situating MUSE within the musical context of chance and indeterminacy.

Collaborative Musical Experiences for Novices

From an interaction perspective, MUSE is a co-located, multiplayer computer game situated within the broad Human Computer Interaction (HCI) domain. A number of projects spanning several research areas informed the design of MUSE. Though these works provided valuable HCI insight, they do not fall under the scope of this thesis and thus they will not be analyzed in detail here. In providing the research background against which MUSE's contribution can be appropriately evaluated, and to situate my project within clearly defined contextual boundaries, the related works presented in this chapter are situated in the larger HCI field and cover some or all of the following research areas: collaborative music, music creation and performance in real-time, music games, music interfaces for novices. The discussion will take place from a *collaborative interactive music system* perspective and start by situating each system within the larger context, followed by an examination of the system's implementation of its constituent components, and by looking at the system's behavior and its impact on the user interaction and user experience. Moreover, I will briefly touch upon the pros and cons of each design and their influence onto the approach taken in the design of MUSE.

Introduction.

Blaine and Fels (Blaine and Fels 2003) offer the most recent literature examination on the design and implementation of collaborative musical interfaces for users with no formal musical background and cover mostly academic projects published prior to the year 2003. Most of the examined interfaces share a common "walk-up and play" design feature usually requiring users between five and ten minutes of learning time.

Projects are discussed within the following contextual and design constraints:

1. *Focus*: the intended audience of the musical output: participants or external audience
2. *Location*: e.g. installations for public exhibition
3. *Media*: what role sound and/or image play in the interface
4. *Scalability*: how scalable a particular design in allowing more users to collaborate
5. *Player interaction*: interface design and its impact on “learning by observing”
6. *Musical range*: approaches taken in minimizing a chaotic musical output in real-time
7. *Physical interface*: affordances and modes of interaction
8. *Directed interaction*: distributed leadership and turn-taking behaviours
9. *Learning curve*: the time required to achieve a general sense of musical agency with the interface
10. *Pathway to expert performance*
11. *Level of physicality between players.*

This literature review acknowledges that the main challenge in designing for collaborative experiences for novices is managing the trade-off between a low entry-level skill affording novice users access to a collective experience (associated with a highly restricted musical control) and the creative freedom leading to opportunities for skill and musical mastery with the interface (associated with the need for musical knowledge). The authors conclude that the “easy to learn and play” characteristic should take priority in the design of collaborative musical interfaces for novices.

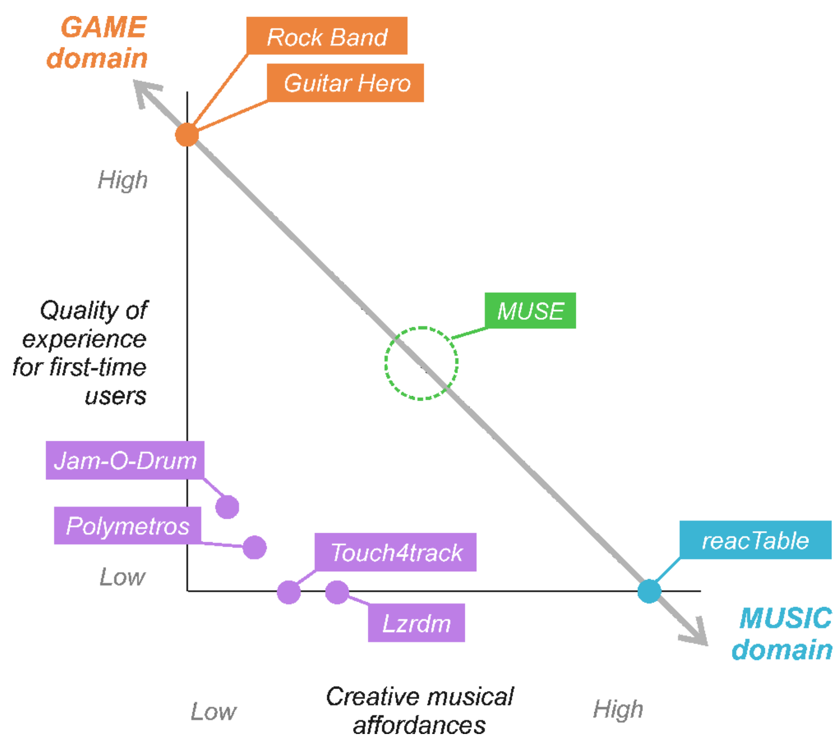


Figure 3: MUSE within the context of related collaborative music systems

I believe that one of the main contributions of my research is the *design approach* taken in the creation of MUSE – a real-time, collaborative music application for novices. Therefore, the projects examined here are situated in the larger context of *collaborative music systems* based on the design approach employed rather than the features they exhibit. Figure 3 shows a number of more recent commercial and academic projects and indicates the tendency of these systems to either gravitate towards the music domain (i.e. creative musical affordances) or the game domain (i.e. quality of experience for first time users). Prior to starting the research for MUSE, while still in the ideation stage, I strongly believed that there should be a way to reach an *ideal* intersection between these two domains. The investigation of these related collaborative music systems revealed that an alternate – if not completely different – approach to the design of such a system was required if increased creative affordances in music interfaces for novices were to be achieved. My design approach is discussed in greater detail in Chapter 3 MUSE.

In the next section I examine the collaborative music systems shown in Figure 3, discuss the approach to their design and highlight the impact this has on the trade-off proposed by Blaine and Fels as the main requirement in designing collaborative musical experiences for novices.

Commercial applications.

Game-oriented design.

Game-oriented music systems are designed for positive experiences for the musically untrained user and are mostly characterized by playful interaction, and ease of use and learning of the interface and gameplay.

Rock Band and *Guitar Hero* are two examples of game-oriented music applications for novices that evolved out of MIT research exploring easy-to-use interfaces designed for encouraging music participation among users regardless of their musical experience and ability (Miller 2009). These two music games share many similarities, one being the use of custom made controllers that resemble musical instruments (e.g. guitar, drum kit). The game uses pre-recorded songs as musical material taken from a large repertoire of rock, punk and metal released from 1960 to the present. Overall, players perform sequences of input control actions on these instruments simulating playing the actual song while the original recording of the song is played back. A song is usually split into four parts rendered as four audio tracks during gameplay. A player can take any part – lead guitar, bass guitar, drums, or voice (through the addition of a microphone in karaoke mode), and can either play alone or with other players taking on the other parts. The basic game mechanic relies on the player reading the simple on-screen notation which indicates what frets on the guitar (i.e. a button on the custom guitar controller) to finger or what drum component to hit at what time. By correctly matching the visual cue to the indicated input control in real-time, the associated sound of that particular instrument track is played back through the speakers and the player is rewarded game points. If the player misses an input action, whatever the associated sound was to that action – comprised of a note or groups of notes on the audio track, depending on the game difficulty – is muted and the player loses points. The success of these games relies on the simple mechanic of rewarding eye-hand skill coordination, while giving the players the impression that they are actually performing the music. Miller (2009) goes on to say that *Rock Band* and *Guitar Hero* add a new dimension to the notion of "schizophonia" introduced by R. Murray Schafer, that of "schizophonic performance" where a previously recorded sound is not only dissociated from its source, but is now combined with the physical gesture of live performance. By providing an engaging user experience characterized by an easy-to-use interface and game mechanic, a musical output that is not only recognizable but appreciated by players, a game system that rewards skill mastery, and the opportunity to share this experience in a collocated setting with other players, *Rock Band* and *Guitar Hero* achieve the goal of providing novices the opportunity for collaborative, participatory music performance. However, the performance aspect is related only to the theatrical character of music performance, and not to the actual rendering of a musical piece. Most importantly, the opportunity for creative input is virtually non-existent.

MUSE also focuses on the quality of the user experience in its design, but takes a step further in contributing to the player's sense of autonomy and competency by providing a music system that allows for real performance of musical changes and for creative exploration of musical possibilities in real time.

Music-oriented design.

The design of most *music-oriented* applications focuses on the implementation of music controls, and aims at providing the user with many opportunities to explore and express their musical creativity similar to most musical instruments.

reacTable (Sergi 2010) is an interface that encompasses the characteristics of a music-oriented application design. Built for the production of live electronic dance music, reacTable features a modular synthesizer approach in implementing both its audio engine and user interface. One or more users can produce music in real time by connecting and controlling sound modules such as oscillators, filters, and sequencers and their associated parameters through the manipulation of physical objects placed onto a tangible tabletop display.

Figure 4: reacTable (Sergi 2010)

The interface provides users with immediate visual feedback of the state of each individual musical module and its relation to the entire music system at any time. This implementation, along with the considerable amount of musical and sound parameter controls available, leads to an engaging human-computer interaction usually associated with traditional musical instruments. The creators state that reacTable is suitable for both novices and experts alike. However, I argue that the heavy reliance on the electronic music paradigm, such as modular synthesis techniques and its associated graphic symbols and metaphors render reacTable unsuitable for novices, in that the quality of both the user interaction and of the musical result is mostly unsatisfactory. In the case of a multi-user session, reacTable offers the same open interface, with no territorial boundaries between personal space and communal space. Without formulating a plan of action or a pre-determined score to attribute scopes and roles to each user during a collaborative session, this design approach can lead to questionable results even when experienced electronic musicians are involved. This lowers the validity of the "real-time" collaborative aspect of the interface. Because *reacTable* was primarily designed for DJs as a solo musical instrument, the quality of the overall music output in a multi-user setup is directly impacted by the users' knowledge and mastery of electronic music techniques, and their group playing experience.

One of the main design goals with MUSE was to minimize or exclude, the need for prior knowledge of music concepts or techniques without the associated drop in the quality of the music produced and thus of the overall user experience. This was partially facilitated by a design choice focused primarily on meaningful collaborative interactions, with secondary, optional single user capabilities.

Academic projects.

The "easy to learn and play" characteristic – proposed by Blaine and Fels as the main design priority for music interaction experiences for novices (Blaine and Fels 2003) – is also addressed by a number of other academic projects. These projects make further attempts at achieving both *user-friendliness* and *creative freedom* in their interfaces.

However, as shown shortly, these two objectives are implemented in a sequential fashion instead of being addressed concurrently in the design stage. This approach results in projects focused either on *creative musical affordances* first (music-oriented designs) and *positive novice user experience* second (characteristic of game-oriented designs), or vice-versa. The following subsection highlights some of the issues that occur as a result of this approach and their impact on the quality of user experience.

Creative musical affordances first.

Touch4track (Xambò, Laney, and Dobbyn 2011) is an academic project investigating collaborative music making in real-time through the prototyping of a musical interface built on a multi-touch display table-top. The design approach of Touch4track resembles the one employed by reacTable – not only does it address the musical controls first, but its multi-user interface is also based on a tested prototype of a single-user interface which did not take into account the implications of a collaborative setting. The authors later addressed the user experience aspect based on the premise that a division of labour is needed in order to support the ***group awareness*** guideline as defined in the field of Computer Supported Collaborative Work (Dourish and Bellotti 1992).

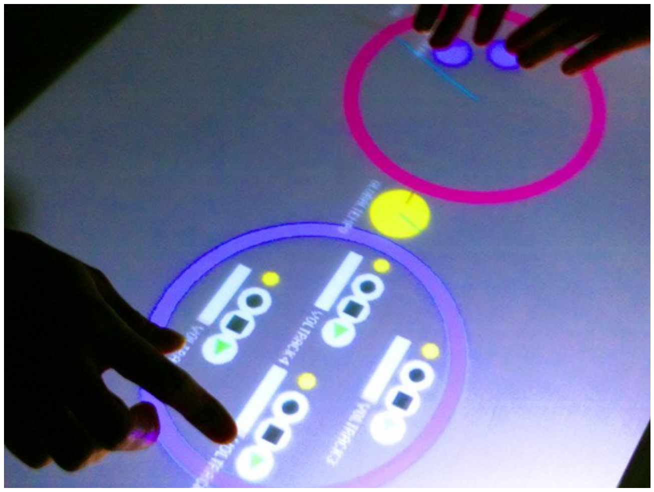


Figure 5: Touch4Track tabletop interface (Xambò, Laney, and Dobbyn 2011)

A user of Touch4track can only use one of the following controls at any given time: playback control (of four recorded samples), timbre control (i.e. filter) and a global pitch shift. While the state of the entire application is visible to all players at all times, and an audible feedback occurs when a user performs a musical change, the “democratic” approach of allowing users to perform actions simultaneously – in a desire to facilitate a concurrent collaborative output – compromises the goal of achieving group awareness. The visual and auditory senses of a novice user performing a musical contribution are more likely to be focused on his actions and not on the other’s creative input. The aspect of group awareness in MUSE is best illustrated by the jazz music practice, where although everybody plays all the time, only one musician can be in the spotlight at a given moment. This method not only allows the “passive” players to observe the active musician’s contributions but also leads to a creative flow, where the musical input of a player is dependent on the group and subsequently impacts the creative direction of the other players during the performance.

Lzrdm is a “co-located collaborative music improvisation” interface, as described by its creators (Klügel et al. 2011). Similar to touch4Track, *Lzrdm* uses a touch-table display as input control. The design of the interface follows a graphical patching of logical units, similar to the data-flow language used in applications such as Pure-Data or MaxMSP and aims at providing controls and metaphors characteristic to a broad electronic music genre.

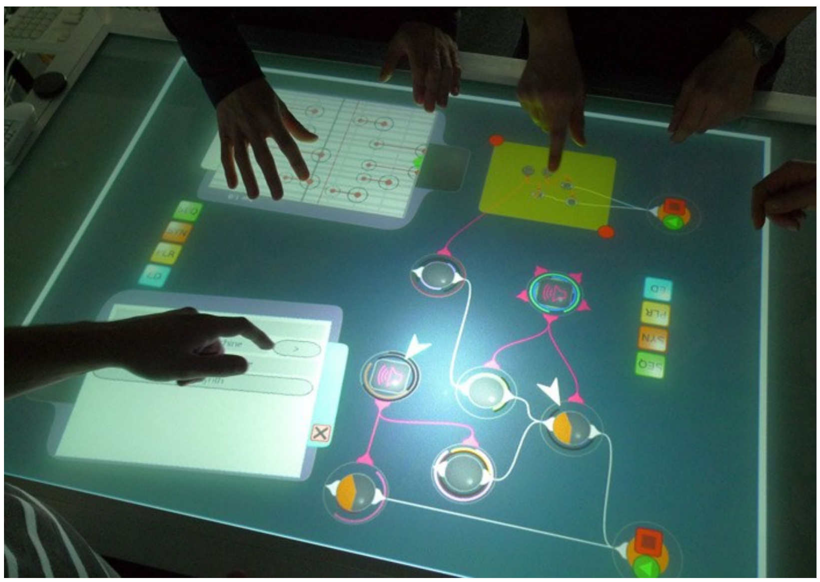


Figure 6: Lzrdm (Klügel et al. 2011)

The novice user friendliness is at best achieved by a smaller amount of musical controls compared to reacTable. The interface however, does not address territoriality which leads to the same problems of identifying individual contributions faced by *reacTable* and *touch4track*. Moreover, the data-flow approach discourages to some extent the implementation of one-to-many mappings of interface controls to musical parameters. This limits the diversity of musical change over time and could lead to aural fatigue and poor user experience. Mapping strategies used in the design of performance controls for computer music performance, as proposed by Rovan et al (Rovan et al. 1997) are classified into three categories:

1. One-to-One
2. Divergent, and
3. Convergent.

Whereas *one-to-one* mapping assigns each individual input to one musical parameter, the *divergent*, or one-to-many mapping assigns an action to a macro-level musical event consisting of several musical parameters. To avoid musical monotony, MUSE implements several *divergent* controls (or One-to-Many relationships) which allow for both continuous changes as well as drastic (distinct), global musical shifts. The availability of such affordances not only reduces aural fatigue but also increases the opportunities for creative, individual expression.

Positive novice user experience first.

Jam-O-Drum (Blaine and Perkis 2000) is a computer mediated "drumming circle" implemented on a seven-foot diameter circular projection surface which uses drum triggers to activate its sounds. Up to twelve users can gather around the table and participate in a collaborative musical experience. Its design goal was primarily to "facilitate group interaction in a public environment" aiming at a positive first-time user experience for a mixed audience of novices and experts alike. The interaction mode is based on a "call and response" approach with the system suggesting a drumming pattern followed by users taking turns in trying to reproduce that pattern or improvise upon it. Jam-O-Drum uses projected visual animations on the table surface to accompany the drum sounds which enhances the positive engagement of participants as reported by its authors. Still, the interface does not really capitalize on the computer mediated system, considering that the overall experience could be easily reproduced with real acoustic instruments and a group leader. Although the system is user friendly through its simplicity, it does not provide creative opportunities to its novice users. The authors acknowledged that there was an increased risk for musical chaos to emerge and went on to developing *Jam-O-World* in which a number of "game-like" interaction modes were added to the system in an effort to give more structure to the experience. One such mode, the "*HexaPong*" aimed at exploring the relations between moving, bouncing balls and musical events (pitch and duration included) in a six player pong game. Although this approach led to a stronger collaboration among players which improved the overall user experience, the actual music being produced emerged as a by-product of the pong-game. The authors concluded that "restricting the player's control over musical elements is enacted to achieve a more sociable environment" (Blaine and Perkis 2000).

Similar to Jam-O-World, *Polymetros* (Bengler and Bryan-Kinns 2013) was designed as an "experience for exhibitions, festivals or public workshops" and aimed at a target audience of various skills, ages, and experience (Figure 7).

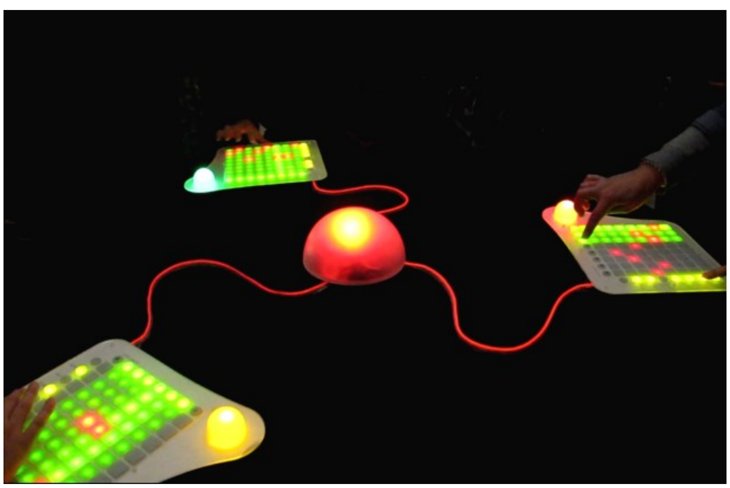


Figure 7: Polymetros (Bengler and Bryan-Kinns 2013)

The focus of this project was to provide novices a positive first-time user experience through an opportunity to "experience and maintain control over their musical activities". Informed by the context of territoriality in collaborative settings, and more specifically, by the need to establish a clearly defined, personal space for interaction, *Polymetros* uses custom hardware controllers instead of a table-top display. The controllers, which the authors call "*instruments*" provide an eight by eight grid of buttons with which users create and edit a musical phrase in a manner similar to a step sequencer, with all musical phrases metrically synchronized. Each player is free to control his eight note long, repeating musical phrase, by adding or removing notes and selecting pitches for these notes, thus adding their personal contribution to the overall musical output.

Despite providing users with individual controllers, the overall group awareness is still missing due to the lack of a communal space for visual feedback of the current state of the entire system. The authors acknowledge this flaw in their user study, observing that players reported difficulty identifying their own contributions among the overall music produced. The "visibility of system status" is the first of ten usability heuristics in the Human Computer Interaction field as proposed by Jakob Nielsen (Nielsen 1995), and probably the most important rule in User Interface Design. I argue that a visual representation of the state of all constituent components and their inter-relationship is mandatory if the personal music contributions are to go beyond random, disparate actions towards becoming meaningful choices that relate to the collaborative music effort. Moreover, the absence of a visual state of the system combined with a largely "free", *un-structured* interaction with the system, leads to an almost nonexistent collaborative engagement between participants. Each user becomes far more preoccupied with exploring their own "instrument" than considering how their contribution affects the whole.

Observations.

The hypothesis proposed by Blaine and Fels (Blaine and Fels 2003) in their literature review, that the *ease-of-use* associated with *game-oriented* interactive music systems and the *creative freedom* provided by *music-oriented* designs tend to be mutually exclusive, is supported by all the projects presented in the previous section as illustrated in Figure 3.

Interestingly enough, *game-oriented,* commercial systems like *Rock Band* and *Guitar Hero* suggest that music is a substantial design component of their system. Similarly, *reacTable*, which has the most musical capabilities among the projects presented here, claims that *ease-of-use* is a significant characteristic of their interface (Jordà 2003). In reality, none of these systems truly integrate what is the most representative characteristic of the other domain into their designs. *Rock Band* and *Guitar Hero* – whose design resides deeply in the game domain – employ music only as a shallow layer (through the theatrical character of rock music performance) added to an already functional game system. Likewise, the novice friendliness of *reacTable* – a fully-fledged musical instrument rooted in electronic music concepts and metaphors – is superficially portrayed by its playful-looking interface of tangible objects on a colorful tabletop display. Although these applications bring forth a layer of innovation in their designs, this innovation does not occur at a functional, system level where games and music might actually intersect. They are successful in their own right as either well-designed games or well-designed musical instruments.

The academic projects taking a "c*reative musical affordances first*" approach – *Touch4Track* and *Lzrdm* – attempt to achieve novice friendliness by considerably reducing the amount of musical controls compared to *reacTable*. Although this results in a simpler interface and interaction, the fact that these projects are designed from a music-oriented perspective (focused on creative affordances) results in a quality of user experience still bounded to the level of musical knowledge of the user. Without taking into account the lack of musical training of their audience in the design stage, these applications provide the user with yet another "*musical instrument*" that presumes understanding of music concepts, and – in the case of collaborative interaction – requires extensive group practice to provide satisfactory musical results in real-time. Not only do these interfaces fail to provide a positive first-time user experience but risk to emerge as poorly designed musical instruments.

Polymetros and *Jam-O-World* focus on achieving a positive novice user experience and approached this by incorporating a certain level of *playfulness* in their designs. However, the oversimplification of the system combined with an unstructured approach to interaction – in order to gain a positive *first-time* user experience – can transform a music system into a musical *toy*. The distinction between playing with toys and playing within a system relates to the research in game studies done by Caillois (Caillois 1961), who clearly separates *paidia* – as relating to spontaneity, excitement and improvisation – from *ludus* – which refers to system-defined, rule-based, organized play. Although novice-friendly, neither of these designs offer long term engagement and opportunity for skill mastery.

Conclusion.

I presented in this section both commercial applications and academic projects addressing collaborative musical experiences for novices and focused on how the approach to designing these systems determines the existence and/or the degree of the ideal trade-off between quality of experience for first time users and creative musical opportunities. I also showed that by approaching the design from either of these two premises usually fails in balancing them, with one of them turning the final system away from the ideal intersection of games and music towards either a musical toy or a musical instrument.

I propose in the next chapter that providing increased creative affordances without compromising the "walk-up and play" requirement needs a new design paradigm, one that merges the game and the music domains together at a core level. Moreover, I suggest that the resulting collaborative music system could feature both playfulness (i.e. *paidia*) within an organized, rule-based play environment (i.e. *ludus*).

Chapter 3 MUSE

Introduction

In this chapter I discuss the design of MUSE and the approach I took towards providing a playful, positive first-time user experience *and* an increased level of creative affordances within a collaborative music setting for novices. I start by examining the idiosyncrasies of both the game and the music domains and discuss the new proposed paradigm, continue with a discussion of MUSE's design and implementation goals, and end by presenting the approach taken in achieving those objectives.

The main design goals for MUSE as a collaborative musical interface for novices are:

1. The interface's purpose is facilitating collaborative music creation and performance for novices in real time,
2. The interface should be easy to learn to play (i.e. 10 – 15 minutes of instruction and no musical knowledge required),
3. The interface should give the users a sense of musical agency by giving them control over the music output (maximizing creative affordances), and
4. The perceived quality of the music output as observed by users should be pleasant or above.

Furthermore, the players' end goal with the interface should be the active participation in contributing to the ongoing, real-time collaborative creation and performance of music.

A New Paradigm

Music and games both lie in rule-based domains, and the quality of experience depends on conforming to the rules that define that domain. However, the rules of a game are considerably smaller in scope when compared to those that define the world of music. Virtually anybody can learn a game suitable to their age prior to playing, grasp the rules, the concepts, and the goals of the game well, and have a positive playing experience on their first try. An equivalent quality of experience for a music performer or creator – from the moment of learning "*the rules*" to the actual musical act – is usually achieved over a much longer time span.

Another important difference between games and music is that games allow users to make their own decisions from the first interaction with the system, which leads to a strong sense of agency. Achieving a similar sense of agency in traditional music creation and/or performance requires significantly more time and effort spent in mastering the basics.

Going back to the context of computer-mediated music interaction for novices in real-time, a few apparent questions arise: how can the time span – from learning the rules to achieving a positive first-user experience in the world of music – be shortened to the time scale of everyday games? Is it possible to provide a novice user with a set of music rules that can be easily grasped just before playing, rules that would lead to a positive interactive and musical experience, while still allowing the user to be in charge of his actions and decisions? I propose that the answer to these questions is yes. With MUSE I approached the design of such an interface / application anchored in a new theoretical paradigm, that of a *game of music*.

MUSE as Game of Music

Although the proposed concept of a *game of music* can be thought of having a similar meaning with that of a *music game* – as represented by the game-oriented designs discussed in the previous chapter – I argue that there is a fundamental difference between the two. It is this difference that inspired the approach taken in the design of MUSE, which will be discussed in this chapter.

A *music game*, as represented by *Rock Band* or *Guitar Hero*, is comprised of a self-contained set of components, rules, and interface that can co-exist outside of the music domain as a stand-alone game. The music component is not an integral part of the mechanics and dynamics of the game, but rather acts as a non-functional, aesthetic layer. The game system itself is not integrated within the music domain and does not form any deep-level relationships with the rules and dynamics that define music making and performing.

A *game of music*, however, is a game where a user plays *with* music, meaning that all the constituent game components are anchored into the music domain and provide a unified game system. More importantly, the game system cannot exist outside of the music domain, since its economy depends on the music's own rules and principles. This orients the discussion towards the topic of bridging the domains of game and music.

Gamification as Design Context

To better illustrate the fundamentally different approach of designing a *game of music* compared to that of a *music-game*, I will briefly examine the concept of *gamification* and how it relates to the discussion of merging games and music. In a general sense, gamification is described as the integration of game design elements into a non-game domain (Deterding et al. 2011).

The term *gamification* first appears in 2002 (Marczewski 2013), but started gaining more popularity around 2010 in various contexts such as education, health, marketing, finance, and productivity. Depending on context, gamification is often referred to as “playful design” (Ferrara 2012), “game layer” (Priebatsch 2010), “applied gaming” (Christensen, Griffiths, and Farrer 2009), or “funware” (Zichermann and Linder 2010). All these terms capture to a certain extent the main trait of gamification which is improving user engagement with a given activity relying on the inherent play element of culture and society as proposed by Huizinga in *Homo Ludens* (Ehrmann, Lewis, and Lewis 1968).

Gamification experiment.

During my initial research stages, I experimented with the gamification of music by incorporating an existing, simple game system *within* the music domain by mapping musical parameters (e.g. pitch, timbre, and duration) to an engine written for the game of Reversi (also known as Othello). Although interesting from a sonification perspective, as “collaborative music interface for novices” (with a focus on increased creative affordances), this experiment failed on a number of levels as discussed below.

Within a game system, the strategy employed and the choice of actions trigger a variety of emotional responses in the player, and the intensity of these emotions changes in relation to the impact a given action has on the internal economy of the game system. Similarly, within the domain of music creation and performance in a real-time setting, the actions performed by music players elicit emotional responses based on factors related to the internal economy of music making and performing. The setting of the performance, the musical context, the specific role each player has in the ensemble, the amount and value of creative input of a player, all impact the range and quality of emotional responses in the participant, and thus affect the overall user experience. All the possible interactions a player can have with an artefact – also called motivational affordances (Zhang 2008) – are the perceived opportunities that lead to a rewarding game experience, as perceived by the player in relationship with the game. It follows that the internal economy of a game system shapes the motivational affordances out of which the players' intrinsic motivation emerges. This leads to the conclusion that motivational affordances and the user satisfaction they elicit cannot exist outside of the internal economy of the system that generates them.

This explains why the motivational affordances responsible for generating engaging, emergent gameplay in a game system become meaningless when translated into the music domain. Moreover, the translation of the time domain from one system to another can significantly impact the quality of the user experience. The passing of time, as dictated by the game rules, can make a game system be perceived as "fun" to play when players enter the psychological state of "flow"(Csikszentmihalyi 2014). The timing of what is perceived as good "flow" in a particular game system can have a very different impact on the creation and perception of music in real-time.

Situated motivational affordances.

The initial experiment of "matching" the economics of a game system with the economics of real-time collaborative music proved unfeasible due to the unique idiosyncrasies of these two mediums. Deterding (Deterding 2011) also supports this view on gamification in that game elements that are successful within a game system do not necessarily maintain their attributes when translated to a new context. Deterding argues that these perceived opportunities should be conceptualized as *situated motivational affordances*, in that they are both *artifactual* – object specific– and *situational* – context or system specific. These properties of affordances explain why the approach of bundling a few musical controls in an interface will not result in a music system generating positive first-use experiences. Within the "collaborative music interfaces for novices" context, the affordances of the system should not replicate affordances present in models of music creation and performance that rely on music knowledge. Instead, they should emerge from a context that focuses on providing access to positive, creative musical interactions to users with no musical education. Thus, the design of the game interface, components, rules and the interactions they afford needs to be approached from the perspective of the user's interaction with the game.

User-centered design.

The design process by which the means to achieve a particular goal derive from the envisioned, desired end result is known as "*backwards design*" in education, and as "*user-centered design"* or "*user-driven development*" in product or service development. In education, this process is used in curriculum design by first setting the educational goals to be achieved by students, followed by defining the methods that show acceptable evidence of knowledge, which in turn determine the learning experiences and instruction that would lead to the proposed goals (McTighe and Wiggins 2012). Similarly, the development of services or products using this design approach focus on optimizing the ways the user is able, wants, or needs to use the product in favor of imposing predefined ways of interaction and thus constraining the user to change their behavior (Norman and Draper 1986).

Applying “user-centered design” to collaborative music interfaces for novices requires drawing on the particularities, limitations and expectations of the target users within a live, collaborative music context. I present below some of these aspects that provided me the foundation for designing my application.

Music perception.

An aspect I believe, should play a central role in the user-centered design process, is the novices' exposure to music through listening, and more specifically the process of music perception. When listening to music, novices and trained musicians alike build an individual *personal inner music library* which is acquired throughout one's lifespan and comprises “all the music ever heard, collected and stored in the mind and body of that person” (Folkestad 2012). However, novices assimilate, decode, and store music in their personal library differently than musicians do.

It has been proposed that music cognition involves the use of codes to evoke conscious experiences (Bharucha, Curtis, and Paroo 2006). The “code” designation is used here to describe the ability to transmit and preserve information. These codes are *acoustic* (i.e. enabling information to be transmitted through air as acoustic signals that are registered by the auditory system) and *representational* such as auditory and cognitive. The brain maps acoustic codes onto auditory codes which are then stored as parsed representations of the acoustic signals. At this point, the conscious experience of listening can also be accompanied by affective experiences. All these processes occur in both novices and musicians alike. However, this is where the music perception process changes for novices compared to musicians.

For professional and amateur musicians, along with conscious affective responses (i.e. what the music "feels" like), the auditory codes can be further mapped onto cognitive codes. The occurrence of this process depends on the listener's directed attention while its complexity is determined by the listener's level of musical expertise. This process contributes to the expansion of the trained musician's *personal music library*, and provides the basis for evaluating other music works through auditory, cognitive and affective processes, performing established works according to a particular style, and creating new musical ideas by means of improvisation and composition.

For novices, who lack the knowledge and skills to evaluate and assess the "*what"* and "*why"* behind the music they listen to, the conscious mapping of auditory codes onto cognitive ones does not occur, except at a very basic level (providing that the listener's attention is directed towards this process) involving music aspects such as tempo (slow vs. fast), pitch (high vs. low), and amplitude (loud vs. soft). For the most part, the internal music library of novices consists of basic auditory codes and their corresponding affective qualities, if any. However, research shows that novices can tell chords and melodies apart without knowing the actual difference between them. This is a result of the partly different neural mechanisms involved in processing melodic vs. harmonic information (Koelsch and Jentschke 2010). A study shows that novices not only can discriminate between major (i.e. "happy") and minor (i.e. "sad") modes, but also that the lack of cognitive evaluation (i.e. musical training) of novices leads to increased emotional responsiveness compared to trained musicians (Pallesen et al. 2005). This aspect might speed up the process of retrieving previous auditory codes for appraisal of new music. Although through repeated exposure to music novices become accustomed to basic music concepts and any affective responses they might elicit, they never develop a complex cognitive or visual representation of the music they listen to (i.e. such as in the form of musical notation or as motor memory from playing a musical instrument). In a user-centered design approach, these insights become invaluable as a foundation for creating a collaborative music interface for novices.

Music interaction.

In terms of real-time, collaborative music interaction, novices would expect:

- To perform game actions that have an immediate musical result
- Game actions that have an audible and visual feedback
- Game actions that have a predictable result and can be reproduced
- An easy to understand, intuitive interface and controls
- Easy controls that are well matched to game actions
- Simple game rules, if any
- Fair access to game actions among participants

The interface and components of such an interface should provide the missing visual representation for novices to map auditory codes (i.e. the sounds and music they hear) onto visual codes (i.e. immediate, consistent visual feedback of their actions and the overall state of the music). The mapping of game actions onto controls should result in controls that do not require a high level of dexterity.

Music creation.

From a music creation in a collaborative context perspective, novices would expect to:

- Be able to produce musical results related in style/genre to the music they listen to

- Perform musical changes they would expect for the most part.
- Have control over the music output in a creative way without having to know music
- Need very little time to learn to play.
- Produce overall “pleasant” music.
- Have a positive first-time experience.

Based on these particularities, limitations and expectations, the following user-centered design aspects emerge: the overall gameplay and rules should facilitate quick learning and playing, the music output should satisfy wide audiences, and the gameplay (with all the possible actions and interactions) and the musical results should be defined in terms of both the novices’ process of music perception and their expectations of the game. Overall, novices would expect to be able to interact in real-time with at least the basic music concepts they (may know or not) they are aware of, and at most with concepts that would produce musical results not exceeding their aesthetic expectations.

Design framework.

Though the desired user experience envisioned for the proposed *game of music* can be broadly defined as “fun”, playful interaction, Gaver et al. (2004) best describe MUSE’s aesthetic goal with the term “*ludic engagement*” as an activity “motivated by curiosity, exploration, and reflection”. The field of gamification benefits from the development of a number of frameworks for designing for positive user experiences, such as *PLEX* and *MDA*.

PLEX (Arrasvuori et al. 2011), standing for Pleasurable Experience Framework, offers twenty-two desirable user experiences derived from the work of Costello and Edmonds (Costello and Edmonds 2007) – who identified thirteen “*pleasures of play*” – as well as from conducting their own user studies on video game play. In addressing the desirable user experiences, Arrasvuori et al. make the distinction between *functional experiences*, such as the perceived usability and usefulness, and *emotional experiences* which relate to the emotional aspect of the user experience. PLEX mainly focuses towards *playful experiences*, thus covering the emotional end of the UX spectrum. The purpose of the PLEX framework is to aid designers in brainstorming desirable user responses to lay the foundation for a unified game experience.

MDA is another user-centered design framework proposed by Hunicke et al. (Hunicke, LeBlanc, and Zubek 2004). Although it features only eight broad categories of desirable user emotions, it also provides a directed approach to designing for experience. MDA, or *mechanics-dynamics-aesthetics*, emphasizes the difference between a designer's perspective on a game and that of a player: whereas designers work with *mechanics* such as game interface, rules, and components (Brathwaite and Schreiber 2008) to create interactions that lead to desirable emotional responses in users, players perceive the game from the aesthetic, experience side, and in so doing, they infer understanding of the interactions and mechanics responsible for the quality of the gameplay. The MDA framework suggests that by constantly keeping the player's perspective in mind during the design process, designers have more control over, and opportunities to enhance the user experience in their games. More specifically, instead of taking a *feature-driven* approach, as seen in the musical interfaces for novices presented in Game-oriented design the design process should be driven by the *aesthetics* and unfold as follows:

1. choosing the desirable emotional responses – *aesthetics* – the designer hopes to achieve
2. defining the interactions – *dynamics* – that can elicit those emotions
3. creating the game design elements – *mechanics* – that afford the interactions to occur

Methodology

As already mentioned, my initial vision for MUSE was to provide a collaborative platform offering a *pleasant, creative music exploration* experience for novices in real-time. These preliminary aesthetic goals already exist as broad categories of desirable user emotions provided by the MDA framework. The eight categories comprising the MDA's aesthetic component are:

- *Sensation* – game as sense-pleasure
- *Narrative* – game as unfolding story
- *Fellowship* – game as social framework
- *Expression* – game as self-discovery
- *Fantasy* – game as make-believe
- *Challenge* – game as obstacle course
- *Submission* – game as pastime

Therefore, the aesthetic goals I chose, that best reflected my initial vision for MUSE are in order of importance: *Sensation*, *Expression*, and *Submission*.

Design goals.

To provide a broader context for a later discussion on the implementation aspects without going into technical detail, I will briefly explain the relationship between the aesthetics, dynamics, and mechanics of MUSE. According to the MDA framework, the overall *dynamics* of game play – comprising all the user interactions and all the game states at run time – must be chosen such that they elicit the established desirable emotions in users. The first aesthetic goal, *Sensation*, is determined by the musical quality of the real-time sound output. The supporting game dynamic I envisioned was that of a "no-wrong-notes" game state at any time, where the music hovered above a subjective, "non-disturbing" quality for most of the game play session. Some preliminary mechanics able to support this game dynamic involved limiting the novice users' access to low-level musical parameters and controls that could negatively impact the musical output. The second goal, *Expression*, or game as self-discovery, becomes possible in an environment that allows users a level of musical creativity, to create and perform musical contributions in the established "no-wrong-notes" environment. This dynamic required an approach to interface and controls that abstracted musical concepts and presented them as easy-to-grasp and easy-to-use game components. The third and last aesthetic goal, *Submission*, or game as past-time, emerges from an unrushed, curiosity driven exploration of the musical possibilities of the game. This quality of interaction was made possible by approaching MUSE not as a game of *progression*, where the flow of events is scripted by the designer, but as a game of *emergence* (e.g. similar to sandbox games and some simulator games) where the game state continuously changes as a result of the players' decisions and interactions (Juul 2002). Moreover, as opposed to games of progression, which offer very little re-playability due to the known outcome, games of emergence provide users with increased re-playability value due to the high probability space they offer. This fact informed the design decision to approach MUSE as an open-ended, sandbox environment defined by a no winning condition or reward system awarding points or other incentives to the players. The only incentives in MUSE are the ones afforded by the interaction with, and control of the music in a collaborative setting as detailed in the Gameplay section.

Implementation overview.

Staying consistent with the MDA's user-centered approach, I will briefly describe MUSE from a user perspective.

Figure 8: A four player setup of MUSE consists of a computer running the game, a large shared screen, a pair of loudspeakers, and four gamepad controllers.

Similar to most multiplayer console games, MUSE's graphical interface is displayed on a large TV screen or projected image, and shared by up to four players. Unlike most local multiplayer console games though, the screen image is not split to the number of players, but rather provides a view of the entire game state, much like a traditional tabletop game. The music heard during a game session of MUSE is entirely created in real time by the players themselves and is played back through a pair of loud speakers. During gameplay, players take turns in changing the musical output, thus controlling the game, by using traditional wireless game pads as input controllers.

Though the game actions available to players and the possible interactions are described in the Gameplay section, an analogy with a musical ensemble might provide a better idea of the game's dynamics. MUSE is similar to a four instrument, improvisational musical ensemble that has no conductor, follows no score, but has some clearly established rules as of what is musically allowed. The music starts by having the first musician play whatever they wish (i.e. within the constraints of the rules) for a limited time, and once their time is up they are required to keep repeating the last grouping of notes they just played. In this way, when the following musician joins the ensemble, there is already some musical background against which he or she can personally contribute musically. This continues untill all musicians have joined the ensemble. On subsequent turns, each musician can modify whatever they were playing on their last turn, and thus gradually change the overall music output. However, on a turn, a musician can make more powerful changes by taking the role of conductor of the improvisational ensemble and dictate global changes affecting all the players including him or herself. This can result in global chord progressions, changes of register, of tempo, of the musical material currently being played by the other players, to even discarding one or more players for a given time, among others. For a better understanding of MUSE's gameplay, I recommend watching a short video of a demo gameplay (see Appendix A.4 MUSE Gameplay Demo)

GUI and game components.

The user interface was designed following the ten-foot interaction approach guidelines (Google 2015) aimed at large screen interaction (i.e. TV) with users positioned at approximately ten feet away from the screen. MUSE's UI features large graphic elements and text labels, minimal visual clutter, in order to facilitate easy grasping of the relationship between visual representation of the music system and the musical output at any time. From a color scheme perspective, the background consists of shades of dark grey, while vivid colors are only applied to game elements that users actually interact with. The game interface consists of an eight by eight grid that provides a communal game space for all four players. The only game elements that players interact with, by moving them on the game grid, are called "sound blocks". (Figure 10)

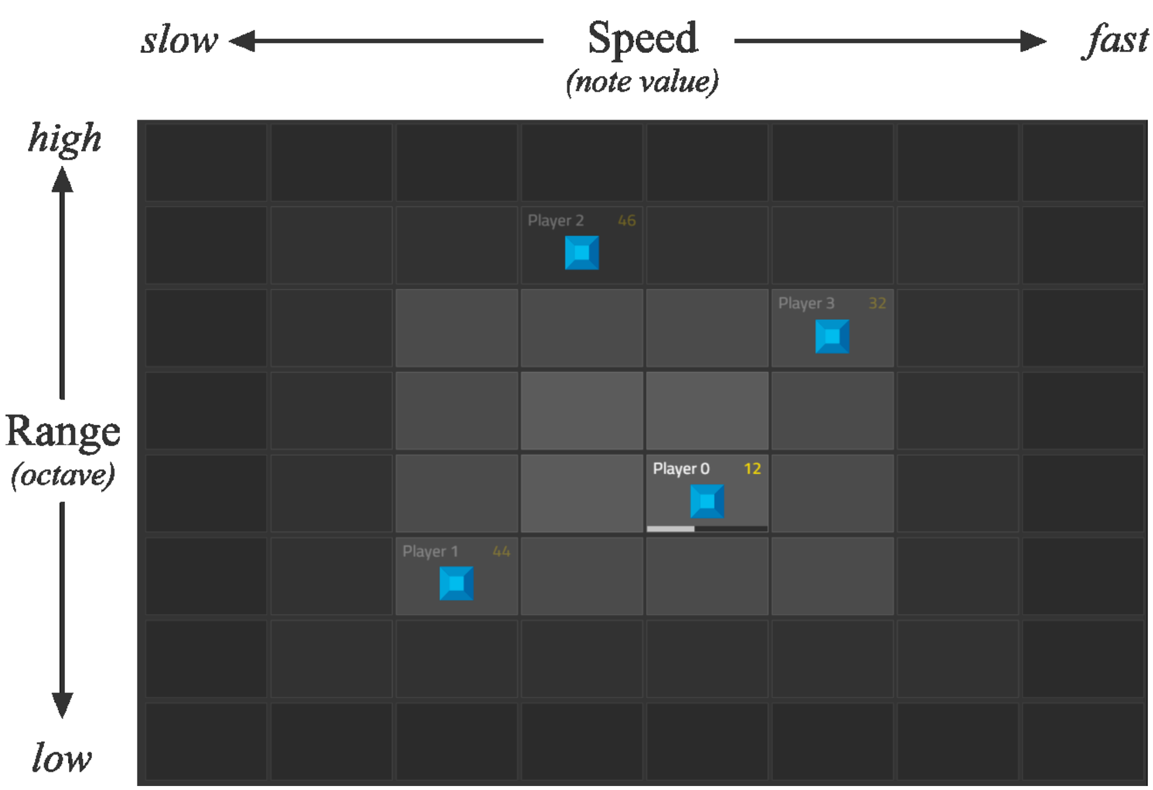


Figure 9: The 8 by 8 grid user interface of MUSE

A sound block (Figure 10) represents a collection of eight musical notes that once placed on the grid, will start playing back its constituent notes and repeating unless another game action affects the sound block.

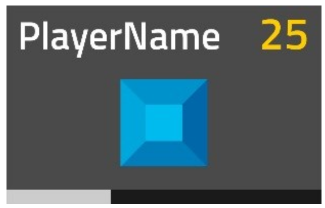


Figure 10: The Sound Block, the main game component in MUSE

At the beginning of the game each player is assigned a sound block that will play back through a distinct musical instrument timbre for the rest of the session. A sound block can occupy only one tile on the grid at any given time, and no two sound blocks can share a tile. The position of a sound block on the grid affects its sounding range and playing speed (Figure 9). The grid's rows correspond to musical octaves, with higher sounding octaves higher up the grid, while its columns determine the sound blocks' playback speed. The speed distribution ranges from very slow on the left side of the grid to very fast to the right, with sound blocks halving or doubling their play back speed for every column to the left and right respectively.

Gameplay.

The core of the user experience in MUSE lays in the probability space of the game system, in which the gameplay emerges from the unpredictable succession of choices made by players in real time. This section presents the game rules and game actions (i.e. mechanics) that make

MUSE not only a safe, musical "playground" for novices to explore, but also a music creation and performance sandbox allowing for creative expression.

Game rules.

To allow novice users to better grasp the relationship between their game actions and the musical results, MUSE implements the time limited, *turn-taking* game mechanic. This approach also enables users to infer gameplay knowledge and stimulate their creativity through observation. Learning by observing and "pitching in" have been proven to be an important informal learning experience that only complements traditional education (Paradise and Rogoff 2009). The turn taking mechanic gives each player the opportunity to be the "active player" for a limited time, time during which he or she is the only one controlling the game's musical output. The game starts with players taking turns "activating" their sound block, which turns on the audio for that specific block. This first round is not time limited. However, on all subsequent turns, a time counter is displayed in the top-right corner of each player's *Sound Block* (Figure 10). A player receives 25 seconds of "action time" every turn, time he or she can use to perform the game actions of their choice when they are the "active player". Once the timer reaches zero, the active player's turn ends immediately and it is the next player's turn to take control of the game. If a player wishes to have more than 25 seconds available on their turn, they can use the *END TURN* game action before their timer runs out. Since they receive 25 seconds each turn, if players wish, they can *END TURN* for several consecutive turns and thus accumulate up to a maximum of 60 seconds of available *action time*.

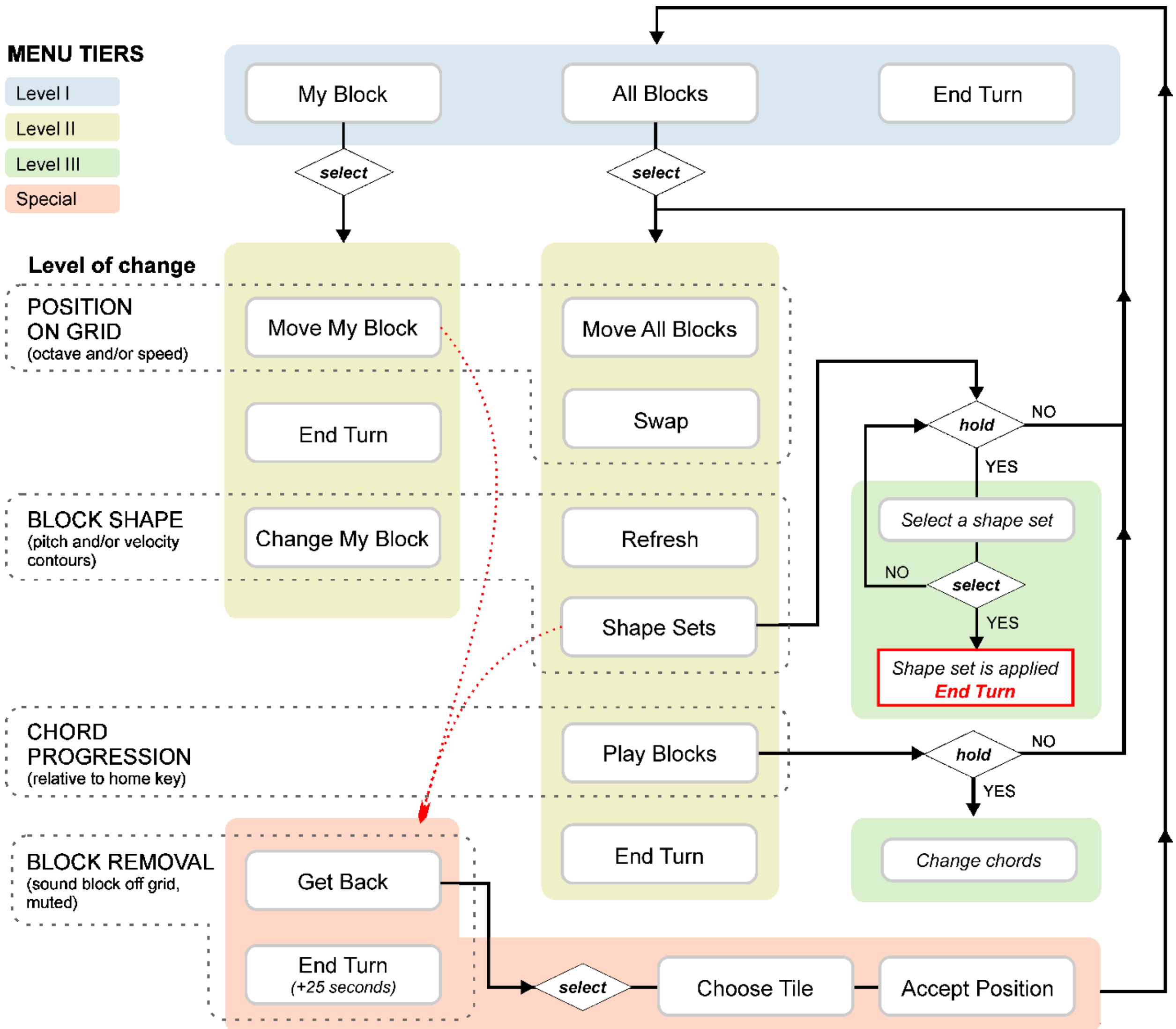


Figure 11: The Menu Actions / Navigation in MUSE

Action menu.

The game actions available in MUSE are organized in a three-tier deep navigational structure. The first tier menu (Figure 12) offers players the option of choosing one of the following actions:

- Perform actions on their own sound block – *My Block.*
- Perform actions affecting all the sound blocks in the game – *All Blocks*.
- Skip their turn – *End Turn*, and in so doing, adding 25 more seconds to their action time which they can use on their next turn.

Figure 12: Tier 1 Menu

The *My Block* option accesses a second tier menu (Figure 13) containing the following game actions:

- *Move my Block*: allows a user to move his own sound block horizontally and vertically on the grid, thus changing the block's speed and register respectively. These musical changes happen in real time. Although a sound block is composed of an eight-note loop, when the block moves to a new tile, the musical change (i.e. note value and/or octave) occurs instantly, regardless of whether the loop has finished a playback iteration or not. This action allows moving a block anywhere on the grid for as many times as the active player wishes (limited only by how much action time they have left on that turn).
- *Change my Block*: allows a user to change the musical character of their sound block, by changing the block's graphical representation to one of the four geometrical shapes available, as shown in Figure 13. The four block shapes are: *Square*, *Triangle*, *Circle*, and *Pentagon*.
    - The *Square* shape renders all the notes of a block at the same pitch and velocity, which makes it sound static and repetitive.
    - The *Triangle's* notes still have the same pitch but they also have a velocity contour, which means that some of the notes will be loud, some soft, and some even silent. This gives the Triangle a rhythmic quality.
    - The *Circle* has a pitch contour, resulting in a more melodic character, but its velocity stays at a fixed level for all its constituent notes.
    - The *Pentagon* has both a pitch and a velocity contour, which makes it the most complex sounding shape in the game.

- *End Turn*: allows the active player to end their turn at any time, even before their action time runs out. Any unused time is preserved till the next turn, when it is added to the regular 25 seconds of action time per turn.

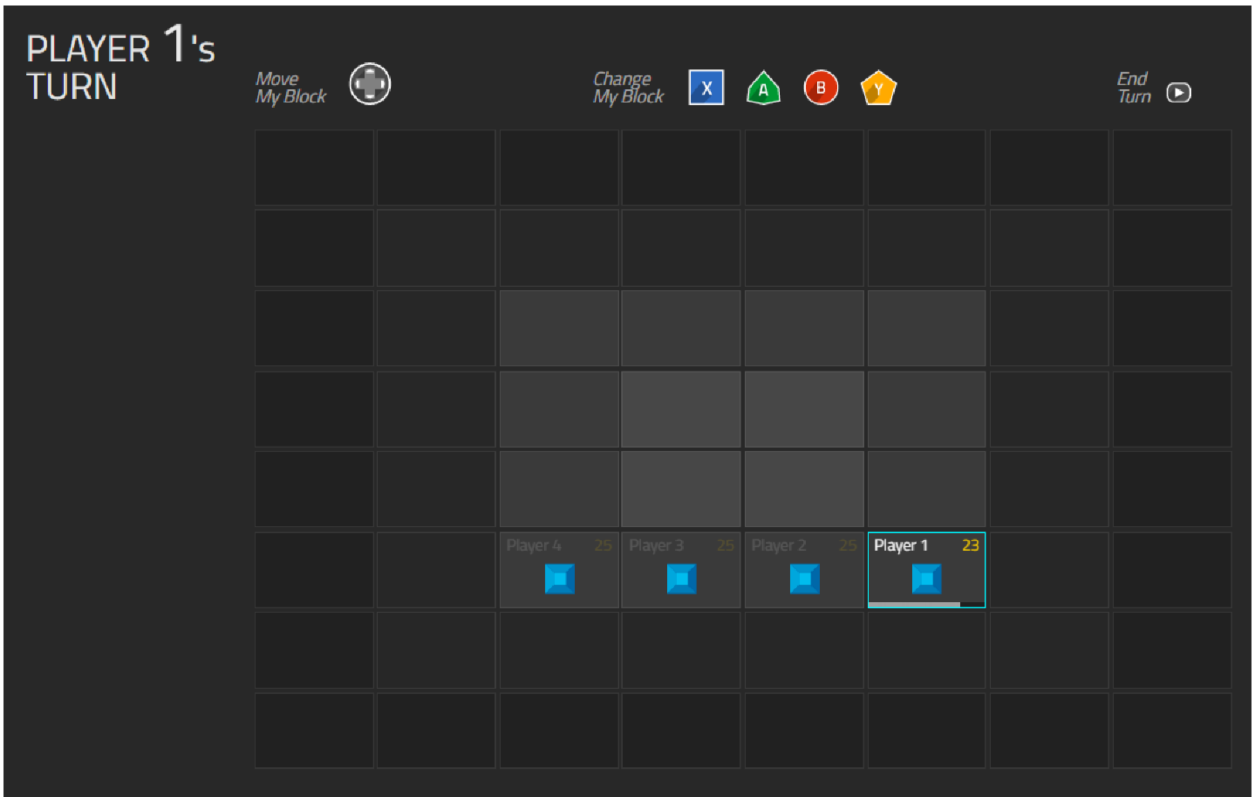


Figure 13: The *My Block* game action, 2nd tier Menu

These actions comprise all the available controls a user has on their own sound block. Before presenting the actions affecting all the blocks in the game, some aspects require clarification. The shape in MUSE is the main building block for the music of the game. The entire sound output is the result of the countless number of configurations between the blocks' shapes and their position on the game grid, all occurring in real-time. As discussed, each type of sound block has a distinct musical character that is also portrayed through the choice of geometrical shape they have in the game. Figure 14 illustrates the characteristics of all these shapes:

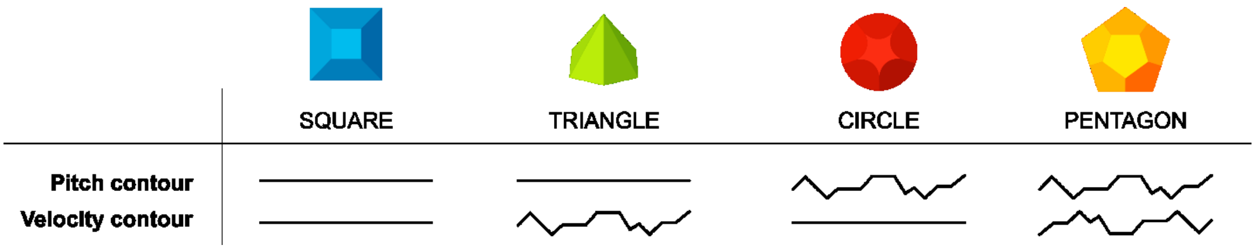


Figure 14: The sound block shapes in MUSE

The game engine creates a pseudo-random collection of pitches and/or velocities (depending on shape) every time a user selects the *Change my Block* action. This ensures a degree of musical variation between multiple instances of the same shape. For example, selecting the *Triangle* ten times in a row will produce ten different rhythmic patterns. Moreover, a user can repeatedly change the shape of their block while moving the block on the grid at the same time and achieve a higher degree of melodic and rhythmic variation.

Because the sound complexity of these shapes increases from *Square* to *Pentagon*, in order to control the range of musical complexity in the game, and thus the perceived quality of the music output, I treated these shapes as a limited game resource and reduced their availability to a maximum of: 4 *Squares*, 3 *Triangles*, 2 *Circles*, and 1 *Pentagon*. For example, if two players have already chosen the *Circle* as their block's shape, a different player who is currently the active player can only choose the *Square*, the *Triangle* or the *Pentagon*, since the Circle maximum limit is two. However, if the active player performs *Move my Block* and thus lands on another's player tile, that player is temporarily removed from the game, its block's sound muted, and the shape it used to have becomes available to the current player (Figure 15).

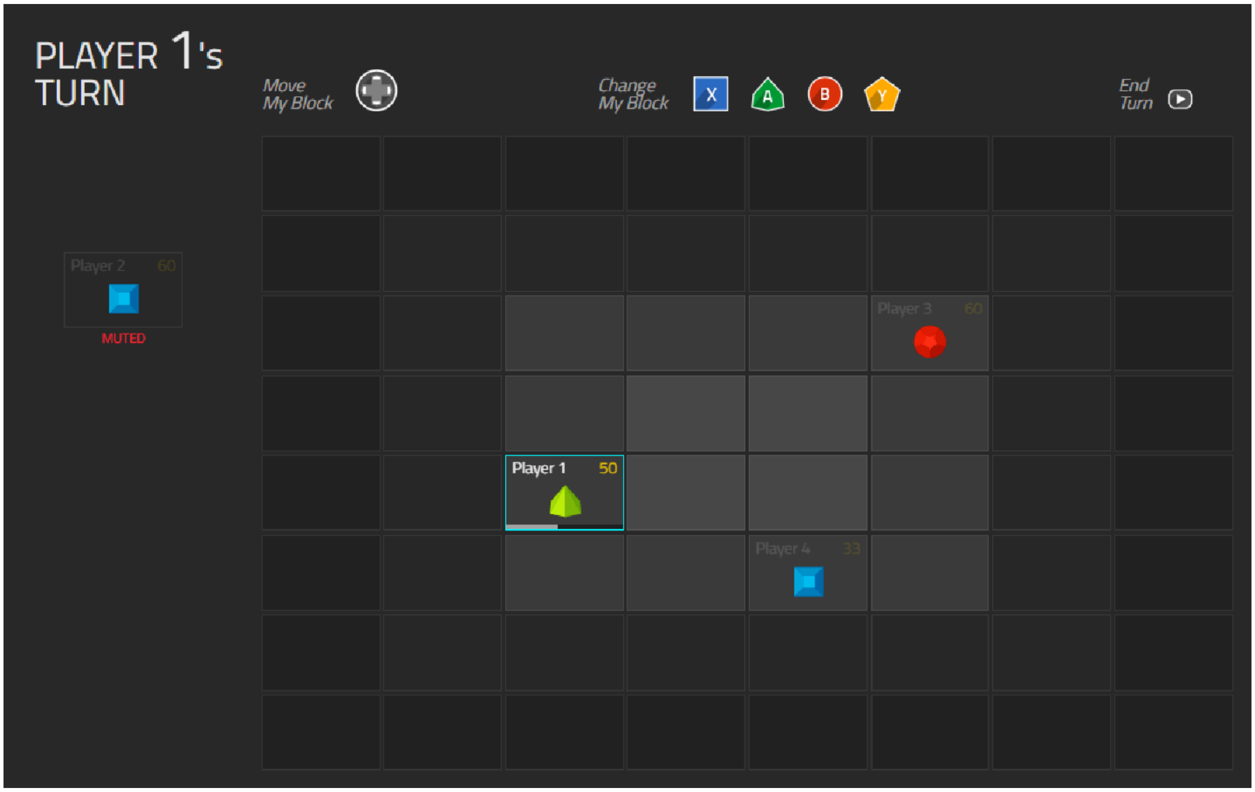


Figure 15: Gameplay showing a removed (muted) player

When the removed player becomes again the active player – as shown in Figure 11–, he or she can either *End Turn* (and gain a +25 seconds bonus for later use) or *[Get Back - Choose Tile - Accept Position]* which will place them back on the grid, turn their block's sound on, and reset their shape to *Square* (which is always available due to the Square's maximum availability of four).

Lastly, the edges of the grid do not impede the movement of a sound block. For example, a block situated at the right edge of the grid that moves horizontally will wrap on the first column on the left. Thus, a block can move in one step between the fastest speed column and the slowest one, as well as between the highest and the lowest octaves. The advantage to this "wrap-around" implementation becomes more evident in the *All Blocks* section discussed below.

The *All Blocks* option reveals a second tier menu allowing the active player to control more blocks than only his own. The actions, as shown in Figure 16 are:

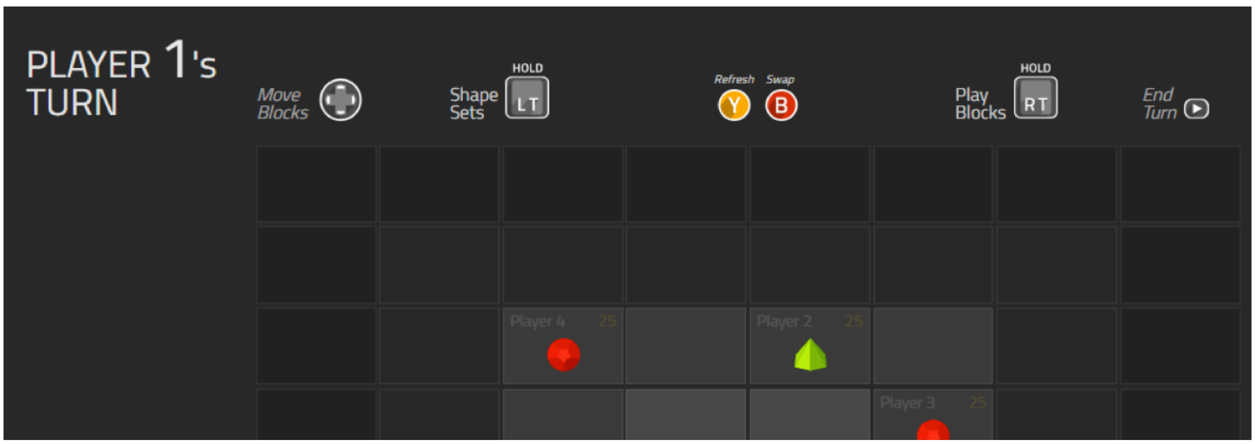


Figure 16: The *All Blocks* menu action, 2nd tier menu

- *Move all Blocks*: this action allows moving all the blocks on the grid as a *block group*. The distance between blocks is preserved, and when the block group crosses a grid edge, the blocks crossing the edge will re-appear on the opposite side of the grid due to the "wrap-around" grid implementation. This action can be performed repeatedly.
- *Swap*: The blocks currently on the grid exchange their places as well as their shapes. The location of the *block group* on the grid remains unchanged. New pseudo-randomly generated pitch and/or velocity contours are applied to the shapes in use every time the *Swap* action is performed. This action can be performed repeatedly.
- *Refresh*: this action generates new pitch and/or velocity contours (according to each block's shape) for all blocks on the grid. This action can be performed repeatedly.

- *Shape Sets*: by holding down the left trigger (LT) gamepad button, a 3rd tier game option becomes available as illustrated in Figure 17.

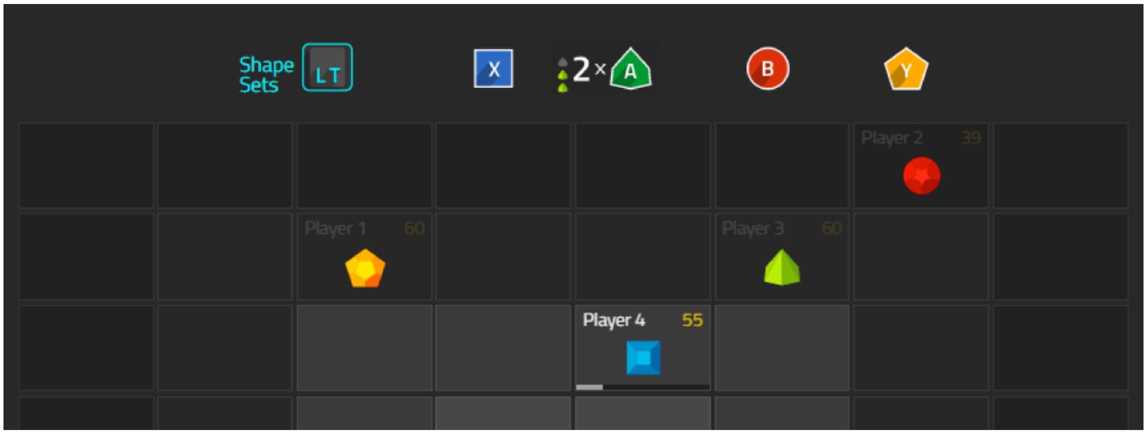


Figure 17: The *Shape Sets* menu action

The active player can select one of the ten shape set options, which will be applied to the current block group on the grid. For example, the shape set selection in Figure 17 will randomly choose two blocks on the grid and apply the *Triangle* shape to them, while the other two blocks will be temporarily removed and muted. The Shape Set options are: 1, 2, 3 or 4 *Squares*, 1, 2 or 3 *Triangles*, 1 or 2 *Circles*, and 1 *Pentagon*. Any shape set containing more shapes to be applied than the current number of blocks on the grid is not displayed as an option. Moreover, once a Shape Set has been applied, the active player's turn ends.

- *Play Blocks*: by holding down the right trigger (RT) gamepad button, another 3rd tier game option becomes available as illustrated in Figure 18.

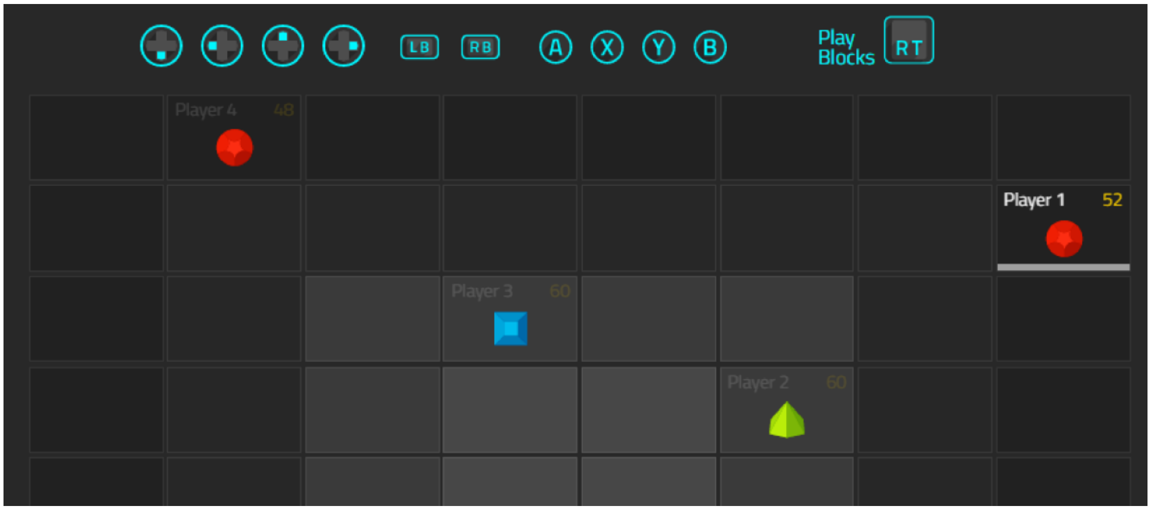


Figure 18: The *Play Blocks* menu action

The ten gamepad buttons illustrated above correspond to ten musical scales built on the diatonic steps of the current musical scale (i.e. key). Pressing any of these buttons will transpose the notes of all blocks to the notes of the corresponding scale. This action results in the performance of a real-time chord progression. The release of the right trigger button (RT) brings the user back to the *All Blocks* menu. In this way, the *Play Blocks* action can be alternated with any other action available in *All Blocks*.

- *End Turn*: ends the active player's turn saving any unused action time for later use.

With the exception of *Shape Set* and *End Turn* actions, a user can perform all actions repeatedly. However, at this proof-of-concept stage, once a player has selected the My Block or All Blocks option, he or she cannot go back and switch between these two options. In other words, once a player has chosen to perform actions on their own block they cannot perform actions on all blocks during the same turn.

Chapter 4 Implementation

Choosing the software and hardware platform for the development and evaluation of MUSE required adhering to the same backwards approach used in designing the actual game experience. To correctly evaluate the feasibility of my proposed game design of MUSE, the deployment platform had to be easy-to-use and familiar enough to most users, in order to allow me to efficiently examine the degree to which the design goals have been achieved. Since I expected my contribution to contain some novel design ideas bridging music and games, I did not want the evaluation of those ideas to be hindered by extraneous or unnecessarily exotic platform features. The platform would thus have to facilitate efficient development, deployment and user testing as follows:

- *Design and Development*

  the platform should enable the overall development of a co-located, collaborative, creative experience in real-time and allow for fast prototyping and testing (mature technology, widely available, strong developer community support, and low-cost or free)
- *Deployment and User testing*

  the platform should be widely accessible, free or low cost, and offer users an easy-to-use, simple, and familiar human-computer interaction medium with no custom-made input-control devices, or the need for additional software or hardware components.

Most of the co-located collaborative applications discussed in the Collaborative Musical Experiences for Novices section use tabletop displays as medium for human-computer interaction. These devices offer a high degree of flexibility in designing novel user-interfaces and controls due to their usually large screen size and access to controls from all four sides of the display. However, I felt that although most users are very fluent in touch-screen and gesture control interaction through the rise in the use of portable devices (e.g. smart phones, tablets), interacting with large tabletop devices in collaborative settings is not a familiar experience to many. Moreover, the high cost and reduced portability of this interface made it a less than ideal option.

The platform that is usually associated with a *co-located collaborative application* by most people is the gaming console (e.g. xBox, PlayStation, Wii). Most of these devices use the TV screen as shared video output source and gamepads for user input control. Though gaming consoles are somewhat wide spread and offer familiar human-computer interaction, developing and testing require additional expertise and costs.

The medium that best fulfilled all the implementation requirements set for MUSE turned out to be more ubiquitous than even the gaming console. The web browser is steadily becoming the dominant platform for all kinds of applications across multiple hardware platforms. The multitude and maturity of web technologies and application programming interfaces (APIs) available for the web-browser allows for free or low-cost, fast prototyping and testing, and in the particular development of MUSE – due to strong developer community support – for a research focused on actual design and development rather than on troubleshooting countless technical issues usually surrounding more exotic, experimental platforms.

Software and Hardware Platform

MUSE employs the *javascript* programming language for all the logic and scripting involved. Since it is an interpreted language, javascript does not need to be compiled, and thus runs in any web browser regardless of the computer's hardware. The layout, graphics and user interface components were developed using the HTML5 markup language, CSS3 style sheet language (Cascade Style Sheets) and the jQuery library – a free, open-source javascript library that simplifies client-side scripting in the browser (“jQuery” 2017). These technologies also benefit from the dedicated developer tools embedded in most modern web browsers which make testing, debugging, and prototyping even faster than before.

The audio engine of MUSE was built using the relatively young Web Audio API which provides developers tools to build powerful and complex audio applications for the web browser (“Web Audio API” 2017). In a nutshell, Web Audio API provides an audio context within which an audio signal chain is built starting with a sound source (e.g. oscillators or wave sound files), linked to one or more audio nodes (e.g. gain, reverb, delay, etc.) and ending with an audio destination, usually the computer’s audio interface output.

Lastly, the input control uses the Gamepad API, which although an experimental technology whose specifications have not yet stabilized, it is supported in several major browsers such as Google’s Chrome (“Using the Gamepad API” 2017). As its name suggests, Gamepad API allows web browsers to read gamepad controller events similarly to mouse or keyboard events.

Timing System

From a system implementation perspective, MUSE is closely related to a step sequencer which is usually found in software or hardware machines that generate drum, bass and groove patterns. At its core, a step sequencer is a fixed-length set of “empty” notes, or events – of the same note value – to which a user can add actual pitches and velocities. When playing back, the step sequencer plays through the set of note elements and repeats when it reaches the end of the set. A note that has no pitch assigned to it will sound silent, just as a musical rest. MUSE uses sampled sounds to populate the notes of all the musical loops (i.e. Sound Blocks in the game). A Sound Block is a collection of eight notes (i.e. the fixed length set of notes in a step sequencer) that is repeatedly played back unless a game action is performed which affects the musical characteristics of that Block.

I programmed the step sequencer of MUSE from the ground up using the Web Audio API and core javascript. Because the implementation of a precise step sequencer for the browser is not necessarily straight forward I will describe here the steps taken in designing the step sequencer of MUSE. Hopefully, other designers and developers will find this information helpful, helping them speed up the prototyping of their own browser-based music projects.

Brower-based event timers.

At the core of any rhythmic sequencer lays an internal clock (i.e. a metronome) that allows for accurate scheduling of note events for future playback. The web browser allows event scheduling through the use of JavaScript's own timers. *`setTimeout()`* executes after a specified delay in milliseconds and *`setInterval()`* executes at a regular time interval. Although useful, these functions fall short when used in a rhythmic sequencer because they share the main JavaScript execution thread with all the additional scripting logic, user input, and graphic rendering. To exemplify, let us assume that *`setInterval()`* is used to call a sound playback function every 500 milliseconds, to create a steady beat. If the execution of `setInterval()` is stalled by other computational processes, the beat may arrive several milliseconds to tens of milliseconds later. These potential execution delays are undesirable in a rhythmic sequencer, leading to the need for a reliable clock that is not hindered by the other computation processes in the browser.

The Web Audio API provides developers just the solution, by revealing the internal hardware clock of the audio interface through the `AudioContext.currentTime` parameter. This property is stored as a floating-point number with around fifteen decimal digits of precision. The clock starts ticking at zero when the web page's `AudioContext` is first initialized and cannot be paused, stopped or reset. The `currentTime` parameter returns the time in seconds elapsed since `AudioContext` was initialized. The Web Audio API also provides a `BufferSource.start()` function for playing any sound loaded in that buffer. This function can take a time parameter – represented on the `AudioContext`'s internal clock time line – allowing playback events to be scheduled in the future. A simplistic approach to creating a steady beat using only the Web Audio API's clock is to statically schedule a number of future events using multiple `BufferSource.start()` calls with different `AudioContext` times passed to them (e.g. start(now), start(now + 1sec), start(now + 2sec), etc.). While possible, this crude approach needs all the events to be scheduled before the start of the application and does not allow for dynamic, on-the-fly tempo changes.

Audio event scheduler.

The solution to building a reliable metronome in the browser is to use JavaScript's `setTimeout()` to schedule only the next Web Audio API's `BufferSource.start()` (i.e. buffer playback) event as suggested by Chris Wilson (Wilson 2013). Figure 19 illustrates the components and the functioning of this event scheduler.

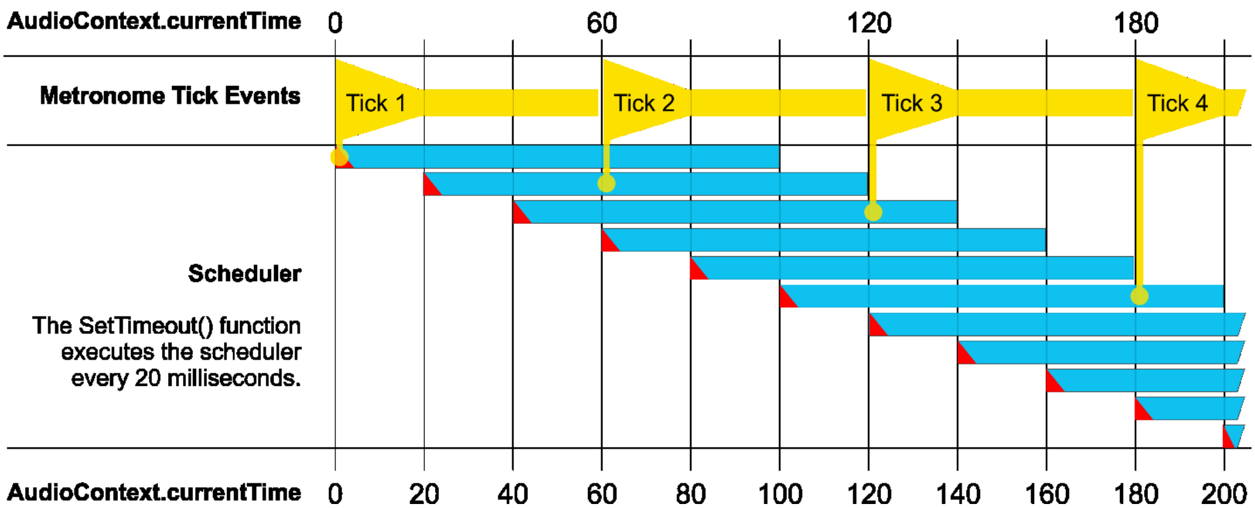


Figure 19: MUSE's Event Scheduling using `SetTimeout()` and `AudioContext.currentTime`

Metronome design.

I approached the design of MUSE's step sequencer by building a basic metronome and using its "tick" to schedule all audio events for later playback. The "Scheduler" function is called at a pre-determined time interval by the JavaScript's `setTimeout()`. As mentioned before, JavaScript timers can be stalled by other concurrent processes, therefore the scheduler needs to execute at a time interval significantly smaller than MUSE's metronome "tick" value. This is to ensure that multiple scheduler calls take place before the next tick playback time. If the execution of the first scheduler gets stalled by other processes, other subsequent calls will still occur before the playback time of the next metronome tick. A detailed explanation of the actual process is described below.

On its first iteration the scheduler checks whether the next tick's playing time is within the "look ahead" window. This window is represented by the blue bar in Figure 20 area A, and overlaps events happening from the `AudioContext.currentTime` (represented by the red triangle in Figure 20, area A) to 100 milliseconds ahead. In this case, the first tick occurs at exactly `AudioContext.currentTime` (i.e. zero).

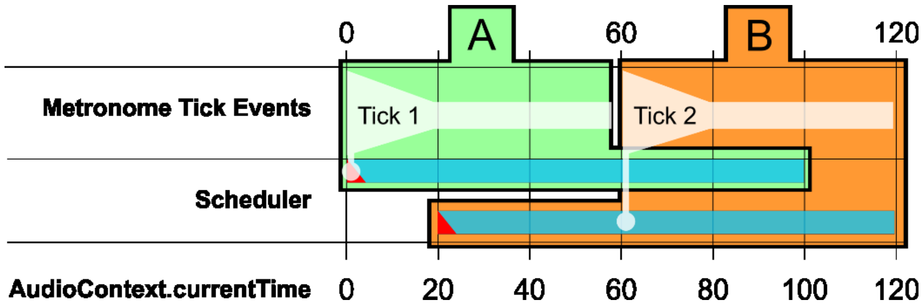


Figure 20: Event Scheduler: detail of the metronome's tick scheduling.

The scheduler then calls the `BufferSource.start()` to play the first tick at its specified time (i.e. zero). Since the first tick's playback time is equal to the present time, the tick plays instantaneously. The scheduler updates the playback time for the next tick, which in this case equals the currentTime plus a fixed duration of the tick, 60ms. On its second execution, twenty milliseconds later, the scheduler (represented by the red triangle in Figure 20 area B), checks if the second tick's playback time is within the "look ahead" window. The system still has 40 milliseconds before it needs to play the second tick. This is where the Web Audio API's `BufferSource.start()` function comes into play. Since `BufferSource.start()` allows events to be scheduled for playback in the future, the second tick is scheduled to occur when `AudioContext.currentTime` reaches 60ms.

Delay execution prevention.

The use of a longer "look ahead" window (i.e. 100ms) combined with a short `setTimeout()` interval (i.e. 20ms) results in more time overlap of the scheduler's calls. This in turn prevents late or missed tick events caused by potential stalling of `setTimeout()` in the main execution thread. As illustrated in Figure 21, the fifth `SetTimeout()` call to the scheduler (dark blue rectangle) comes more than 40ms after its expected time. Although late, the scheduler's current time (i.e. 125ms) is still almost 60ms before of the next tick's playback time (i.e. tick 4), enough to successfully schedule that event to occur at time 180ms.

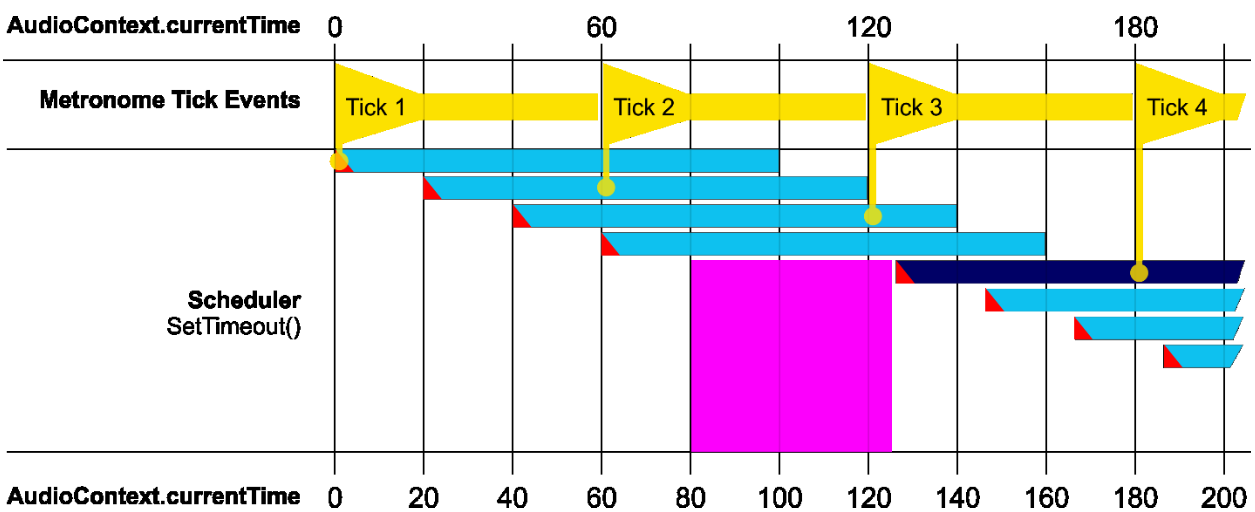


Figure 21: Scheduler Delay: representation of the benefits of a large scheduler overlap in the case of a delayed execution of one call of JavaScript's SetTimeout().

The metronome system described above lays the foundation for the scheduling of all available note values in MUSE as illustrated in Figure 22. Although at a tempo of 60bpm the value of MUSE's tick is 62.5 milliseconds (Figure 22), the illustrations above used a rounded value of 60ms for simplicity.

Real-time speed change in MUSE.

MUSE maps note speed to eight columns on the GUI grid, and as illustrated in Figure 22, ranges from very slow to the left of grid to very fast to the right, with Sound Blocks (i.e. loops) doubling or halving their speed depending on the column they are positioned on. A loop of quarter notes that is moved to the right becomes a loop of eighth notes thus doubling its perceived speed.

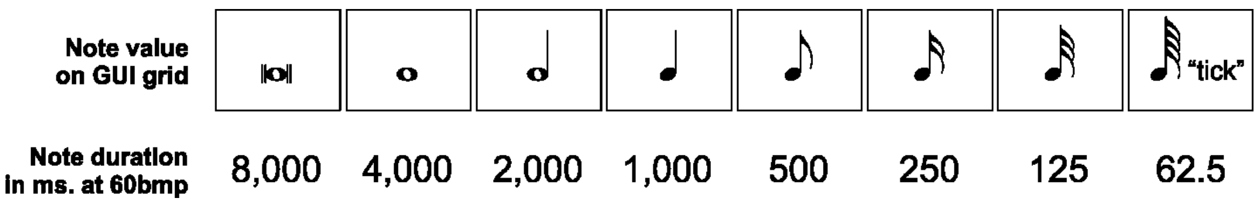


Figure 22: Speed distribution: each note value corresponds to a column in MUSE.

Initial approach.

The initial implementation of speed change in MUSE – corresponding to a horizontal displacement of a Sound Block on the grid – changed the note value of a Sound Block only after completing the current iteration of the loop. As exemplified in Figure 23 if a user requested a note value change when the loop was playing the third of the eight component notes, the change in speed would occur only after all eight notes had been played.

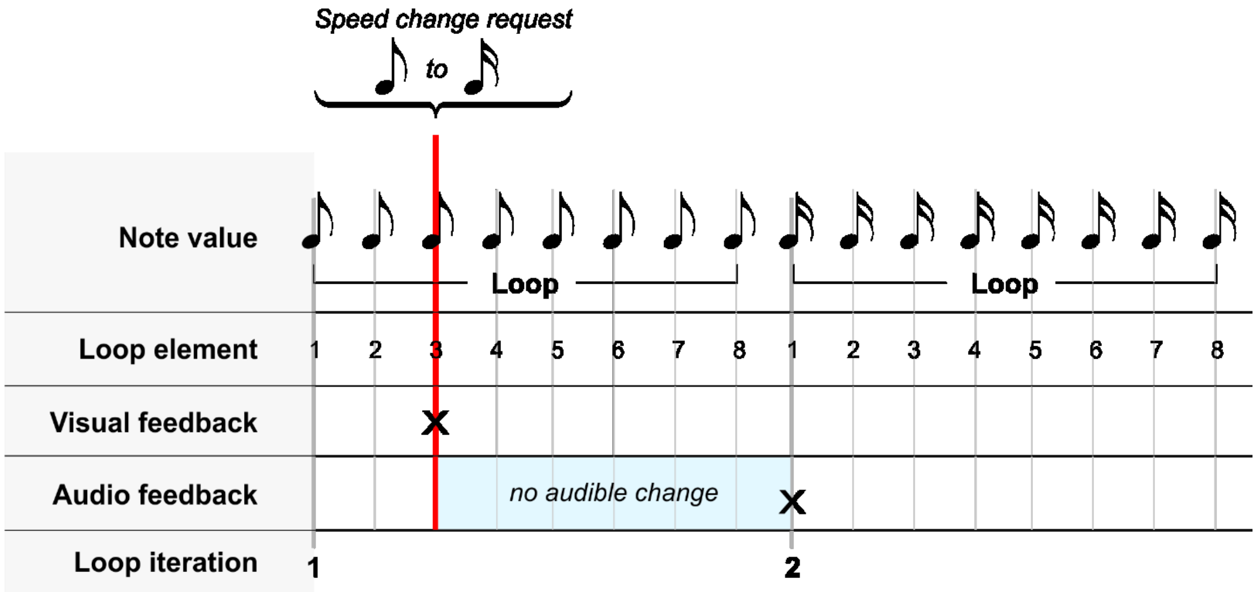


Figure 23: Initial approach to the Speed Change Request (horizontal move on the grid)

This approach negatively impacted the music output by making it sound even more cookie-shaped and mechanical, a music characteristic already imposed by the step sequencer approach. From a gameplay perspective, this made for a less than desirable user experience, in that the action of moving the Sound Block on the speed axis did receive an instant visual feedback (i.e. the graphical representation of the loop moved on the grid instantly with the press of the gamepad button) but a delayed musical one. Moreover, in the case of loops located on the very slow, left-side of the grid, a player would have to wait a considerable amount of time – till the loop finished playing the remaining notes at the previous speed – before actually hearing the musical change associated with his game action.

The next logical step in addressing these issues was making the speed change possible at any time during the iteration of the loop. In other words, the newly requested note value – as determined by the horizontal displacement on the gird – would be applied to the remaining notes in the loop following the pressing of the gamepad button, regardless of where in the loop this request occurred. Still, this generates a de-synchronization between multiple loops as shown in the example illustrated in Figure 24, but produces some very interesting off-beat elements that add rhythmic variety to the overall music. However, during the course of a play session, the off-beat elements resulting from the speed changes of all players, overlap and grow into an extreme rhythmic complexity that changes from being musically “interesting” to becoming entirely chaotic.

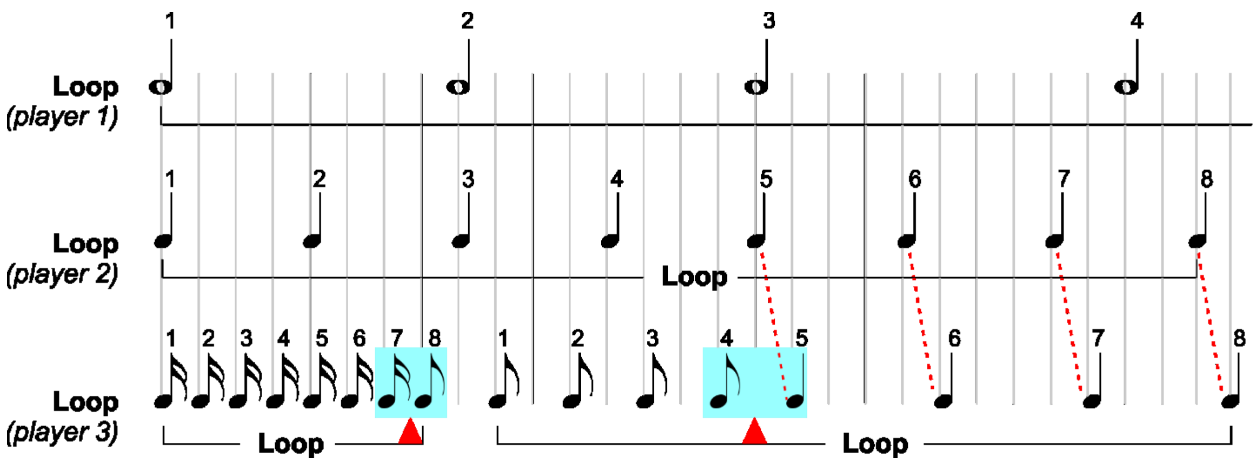


Figure 24: Loop De-synchronization: speed change requests are highlighted in blue. Red dotted lines show de-synchronization of two loops playing quarter notes (i.e. located on the same grid column).

Actual implementation.

This section describes the solution to creating a step sequencer that allows for real time note value changes without synchronization issues. As previously described, the event scheduler stores the playback time of the next metronome tick note. The tick value is equal with the shortest note value available in MUSE (i.e. the 1/64th found on the right most column on the GUI grid, Figure 22). The relationship between the tick and the other note values is stored as a multiplier (Figure 25).

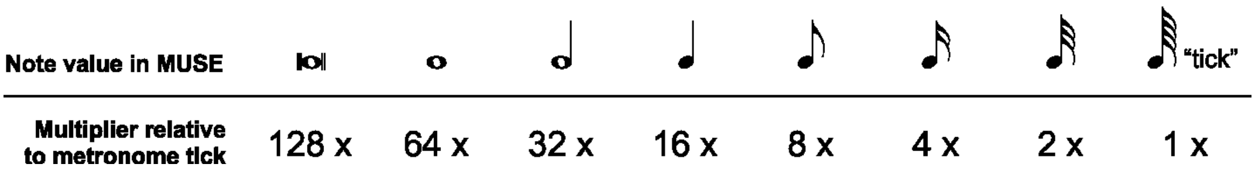


Figure 25: Note Value Multiplier: relationship between metronome ticks and all the note values available in MUSE.

On the very first tick of the metronome, the *Scheduler* stores the playback times for all the note values to zero (i.e. instantly), as illustrated in Figure 19 and Figure 20. This means that any loop, regardless of its speed (i.e. the grid column it is positioned on) will start playing at time 0ms. When the Scheduler schedules the next metronome tick, it also checks whether any other note values (e.g. equal to or longer than the tick value) should be scheduled as well. This is achieved by performing a simple modulo operation between the metronome tick counter and the note value multiplier. If the remainder of the division of the tick counter by the note value multiplier equals zero, then the next note of that particular value is scheduled to occur at the same time with the next metronome tick.

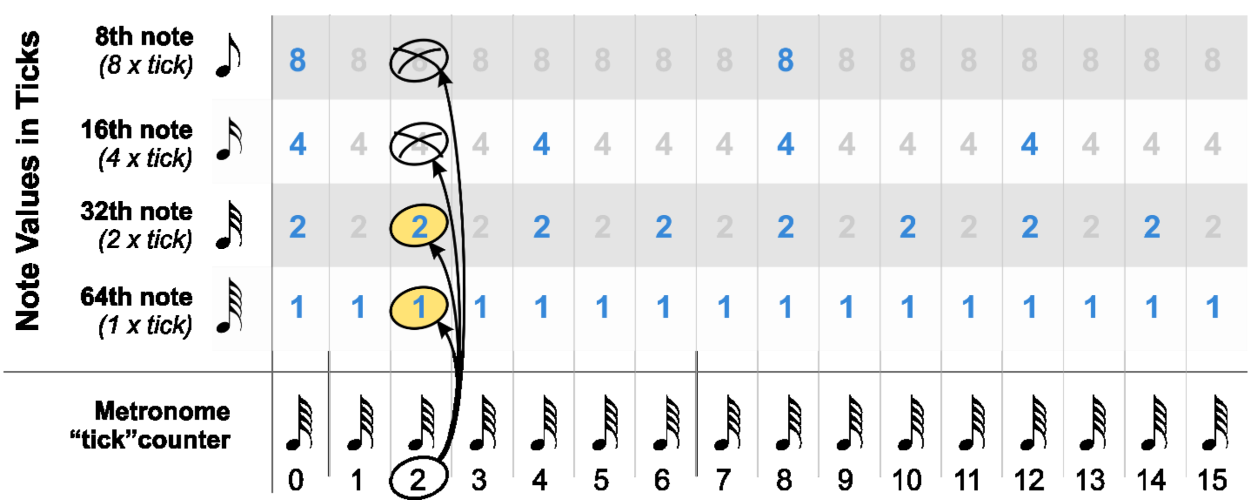


Figure 26: Note Value Multipliers: table illustrating the scheduling of note values for synchronized speed changes. Not all MUSE note values are shown.

For example, when scheduling the 2nd metronome tick (Figure 26), MUSE also schedules the 64th and 32nd note values to take place with the tick, since the modulo of tick counter 2 / the 64th multiplier of 1 = 0 as well as the modulo of tick counter 2 / the 32nd multiplier of 2 = 0. No other longer note values will be scheduled to occur at this metronome tick playback time. We see then, that a speed change action can be performed regardless of where in the loop it takes place (i.e. on which note out of all eight notes). All loops thus stay in sync even when they change speed because the scheduled playback times for all note values are vertically "locked" to the metronome tick. Although the possibility for syncopation to occur as result of the grid movement action (i.e. speed change) does not take place anymore, MUSE still allows for a lesser amount of "off-beat" rhythmic variety to happen through the use of velocity contours for shapes such as the Triangle and Pentagon (see GUI and game components)

Audio System

As described in the Gameplay section, MUSE's sound output is comprised of four ongoing loops of eight notes each. Each of these loops is assigned a distinct musical instrument (i.e. timbre) for the entire duration of a play session. Although Web Audio API allows creating powerful browser-based synthesizers, I felt that MUSE would not benefit enough from it to be worthwhile building one at this prototyping stage. Instead, I recorded studio quality sound samples from a Yamaha MOXF6 hardware synthesizer using a small Max/MSP patch and the Audacity software for additional envelope editing. These samples comprise the basis for the sound banks of the sampler specifically coded for the game.

Instrumentation.

MUSE's sound banks store four distinct sounding timbres: two plucked string and two pitched percussion instruments. Since I anticipated that later on I might need to apply additional sound processing to these samples at a global level – same processing scripting regardless of the instrument – I chose sounds with similar wave envelope characteristics (i.e. attack, decay, sustain, and release). I also felt that my non-musician target users would find that instruments with a clear attack are easier to identify in a four voice ensemble. Moreover, I chose predominantly acoustic sounding instruments based on their larger timbral spectrum across the amplitude range.

In choosing the instruments to sample I deliberately avoided creating an ensemble resembling common instrumentations in order to orient the users towards musical exploration free from associations with pre-existing musical arrangements representative of a particular instrumentation or genre. For the same reason I sampled a full eight-octave range of the guitar and marimba sounds, well beyond their natural range, as generated by the physical modeling engine of the Yamaha MOXF synthesizer. The other two sampled instruments (i.e. a plucked string/percussive instrument and a bell sound) have a somewhat semi-acoustic character, due to some additional synthetic timbral characteristics. The timbre prominence across the full range differs for each of the sampled instruments, as illustrated in Figure 27.

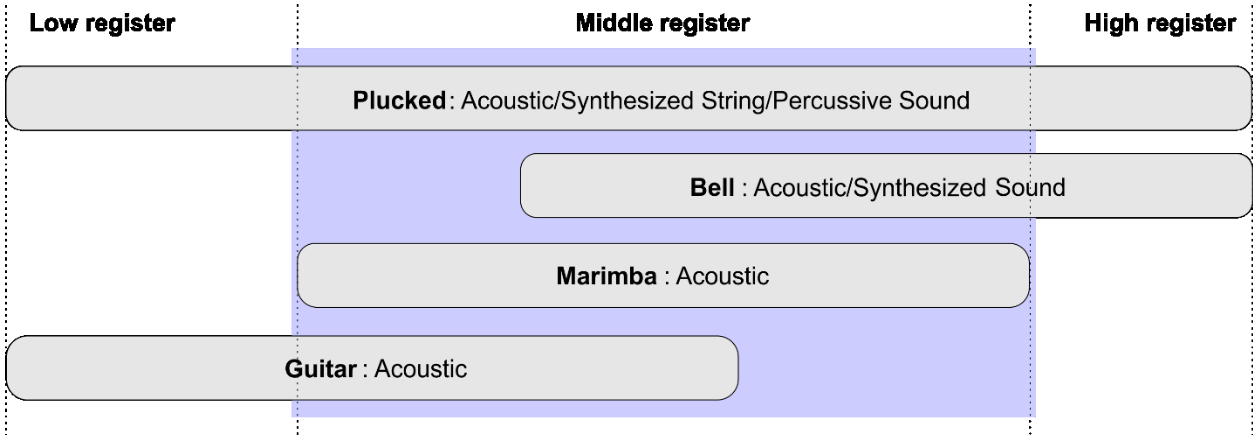


Figure 27: Sampled instruments and their relative timbre prominence across MUSE's eight-octave range

As shown above, all four instruments have relatively equal timbre prominence in the middle register, while only two stand out in the low or high registers. The choice of percussive sounds and the use of the large eight-octave range posed some challenges when addressing the audio mixing of the sound output, which are described later in the Challenges and Solutions section.

Sound sampling.

The downloading and decoding of high quality audio samples in the web browser can take a significant amount of time. In order to speed up this process and still preserve some timbral characteristics of the instruments based on the sample's velocity, I sampled only the C note for every octave at three distinct velocity levels – soft, medium, and loud corresponding to the MIDI velocity values of 50, 80, and 110. These values were chosen based on the significant timbral difference of the instrument between these fixed velocities rather than on the amplitude difference alone. Velocity values are stored in MUSE's sound banks as follows: 0 is silent (no sample), 1 is soft, 2 is medium, and 3 is loud. Each sound bank thus stores 24 samples recorded at 16bit / 44100Hz as illustrated in Figure 28.

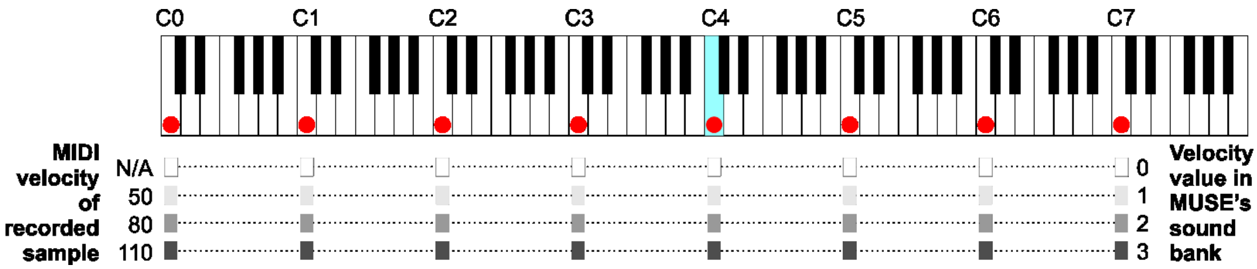


Figure 28: Sampled sounds: recorded MIDI velocity to MUSE velocity mapping

The need for the zero velocity value in the sound banks (i.e. silent note) is further explained in the Musical Entities section. The intermediate notes located between the sampled C's are transposed in real time by MUSE; the application retrieves the sampled C sound closest to the requested pitch and through the use of the Web Audio API's `playbackRate` plays it back at a speed value corresponding to the interval ratios of an equal-tempered scale as illustrated in Figure 29.

Interval	Unison	Minor 2nd	Major 2nd	Minor 3rd	Major 3rd	Perfect 4th	Augm. 4th	Perfect 5th	Minor 6th	Major 6th	Minor 7th	Major 7th	Octave
Note	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C
Ratio	1.0	1.059	1.122	1.189	1.259	1.334	1.414	1.498	1.587	1.681	1.781	1.887	2.0

Figure 29: Note Sampling: the equal-tempered scale ratios applied at runtime by MUSE

Music System

The "no-wrong-notes" quality of the music output – as derived from the main aesthetic goal of the game – is a reflection of what a novice user, familiar with Western music through exposure to popular media, would find relatively pleasant, but not disturbing. Within MUSE, this subjective quality is facilitated by the tonal music system on which the game is built, and more specifically, through the approach taken in the selection, treatment, and implementation of the musical scales that comprise the tonal environment of the game.

Tonality.

The music created in the game is based on a collection of ten musical scales. These scales are built on the first six diatonic scale degrees of the tonal centre of MUSE (I, II, iii, IV, V, and vi), which by default is the key of C major. Although MUSE is designed so that it can use any of the twelve major keys as tonal centres, no game action is currently implemented to handle real modulation to a new key. Chord progressions are however possible as explained later. Figure 30 lists all the scales available in the game and their relationship to the home key.

	Scale degree (relative to the home key / tonic)	Scale type / scale pitches	Base Pitch / relative to TBP (tonic base pitch)	Scale Set distribution of scale steps in semitones (value added to this scale's base pitch)											
Above tonic	Dominant **V**	*major pentatonic / G, A, C, D, E*	*(TBP) + 7*	0	0	0	2	2	4	4	7	7	9	9	12
Above tonic	Subdominant **IV**	*major pentatonic / F, G, A, C, D*	*(TBP) + 5*	0	0	0	2	2	4	4	7	7	7	9	9
Above tonic	Mediant **iii**	*minor pentatonic / E, G, A, B, D*	*(TBP) + 4*	0	0	0	3	3	5	5	7	7	10	10	12
Above tonic	Supertonic **ii**	*minor natural, no 6th / D, E, F, G, A, C*	*(TBP) + 2*	0	0	0	0	2	3	3	5	7	7	7	10
	TONIC **I** (home key)	*major pentatonic / C, D, E, G, A*	***TBP***	0	0	0	2	2	4	4	4	7	7	9	12
Below tonic	Relative minor **vi**	*minor natural, no 6th / A, B, C, D, E, G*	*(TBP) - 3*	0	0	0	0	2	3	3	5	7	7	7	10
Below tonic	Dominant **V**	*major pentatonic / G, A, B, D, E*	*(TBP) - 5*	0	0	0	2	2	4	4	4	7	7	9	12
Below tonic	Subdominant **IV**	*major pentatonic / F, G, A, C, D*	*(TBP) - 7*	0	0	0	2	2	4	4	7	7	7	9	9
Below tonic	Mediant **iii**	*minor pentatonic / E, G, A, B, D*	*(TBP) - 8*	0	0	0	3	3	5	5	7	7	10	10	12
Below tonic	Supertonic **ii**	*dorian, no 2nd / D, F, G, A, B, C*	*(TBP) - 10*	0	0	0	3	3	5	5	7	7	9	10	12

LEGEND

TBP	The Tonic Base Pitch, the first scale degree of the tonal centre (i.e. default key).
Base Pitch	The first scale degree of a given scale.

Figure 30: Distribution of scales and steps in MUSE's tonal system

Music scales.

As shown in Figure 30, in the Scale type column, all scales are either five or six pitches in length. However, a scale is stored differently in the memory of the computer as illustrated in the Scale Set column. These entries (i.e. numbers from 0 to 12) represent the number of semitones between the base pitch of that scale (i.e. its first scale degree) and a given scale degree. A Scale Set has a predetermined number of entries (i.e. twelve), and as illustrated, usually contains more than one entry for each constituent scale degree. The reason for this implementation approach, is that in MUSE, the user does not have direct control over the pitches comprising his or her Sound

Block (i.e. loop). The eight pitches that make up a loop are randomly selected from the Scale Set currently in use and because of the weighted distribution of constituent pitches in a scale, the pitches with more than one entry will have a better chance of being selected. The weight distribution always starts with the tonic of a scale as the most weighted scale degree and usually decreases towards the third and/or fifth of the scale, followed by the other constituent scale degrees.

It is worth noting that at any given time only one Scale Set is used to populate the pitches of all the loops in the game. Since all four loops play notes belonging to one scale only, and because of the weight distribution of pitches within the scale, the harmony produced by the overlapping of voices will have a predominantly consonant quality. This approach also makes chord progressions possible, by giving users control over which Scale Set is currently being used and allowing them to perform a considerable amount of progressions using the available Scale Sets, while within the safe boundaries of the tonal music system employed by MUSE. The distribution of pitches within scales was tested extensively and went through many iterations during the development of MUSE.

Musical Entities

At the core of the musical system of MUSE lays the loop, or as implemented in gameplay, the Sound Block. All the music produced in the game results from the large number of possible combinations of the four Sound Blocks on the MUSE's grid.

Loops / Sound Blocks.

A loop, or Sound Block, is a collection of eight elements (i.e. notes) with each of these elements having a Pitch property and a Dynamic property. The timbral difference among the four Sound Blocks depends on the musical instrument assigned to each player's Block. However, the musical difference among the Sound Block's shapes available in the game (i.e. Square, Triangle, Circle, and Pentagon) results from the combination of pitch and velocity contours as illustrated in Figure 31:

	SQUARE	TRIANGLE	CIRCLE	PENTAGON
Pitch property	**Fixed Pitch**	**Fixed Pitch**	**Random pitches**	**Random pitches**
Dynamic property	**Fixed Dynamic** *(soft, value = 1)*	**Random Dynamics** *(varies, values 0 to 3)*	**Fixed Dynamic** *(medium, value = 2)*	**Random Dynamics** *(varies, values 0 to 3)*

Figure 31: Detail of Pitch and Dynamic Contours of all shapes in MUSE

Pitch and velocity contours / shapes.

A Square shape, as shown in Figure 32, has the Pitch property populated with the Base Pitch of the current scale for all the notes of the loop while the Dynamic property stores the value of 1, corresponding to a soft dynamic (see the Sound sampling section). The musical result of the Square shape is a steady, static repetition of the scale's tonic at low volume.

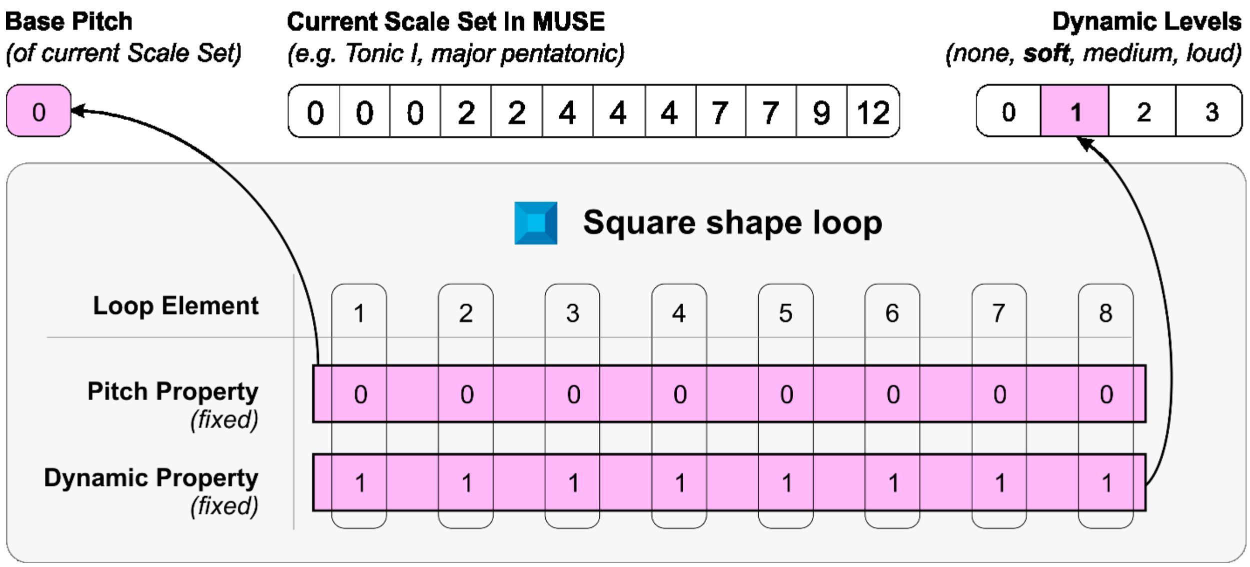


Figure 32: Detail of the Square shape in MUSE

When a player changes their Sound Block shape into a Circle, MUSE populates the Pitch Property of that loop with eight Scale Steps randomly selected from the Current Scale Set. A Circle shape has the Dynamic property set to a fixed value of 2, which corresponds to a medium velocity, as shown in Figure 33 below.

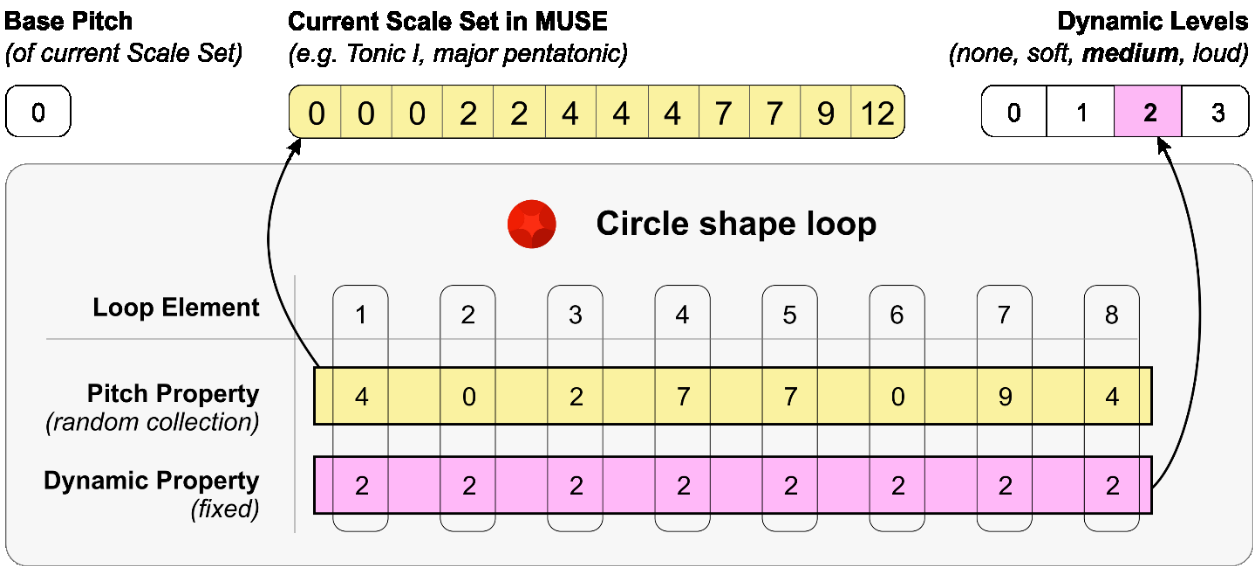


Figure 33: Detail of the Circle shape in MUSE

Apart from having a melodic character due to the variation in pitches and steady dynamic contour, the Circle shape also enables the rendering of sustained sounding chords, as explained in the Reverb section. It is worth mentioning that the random selection of pitches for a loop may or may not contain all the constituent pitches of the Current Scale Set (e.g. in Figure 33 the scale step 12 – the octave – is not part of the selected pitches for the loop).

The Triangle shape is the complement of the Circle in that it contains a pseudo-random collection of dynamics while having the Pitch Property fixed on the base pitch of the Current Scale Set. As shown in Figure 34, the Dynamic levels available can take values from zero to three. A loop element (i.e. note) with a velocity value of zero does not produce any sound. The Triangle shape can thus create off-beat rhythmic figures due to the alternation of silent note elements with accented notes within the loop. Since it only plays the Base Pitch (i.e. tonic) for all its Scale Steps, the Triangle has a predominantly rhythmic, percussive character.

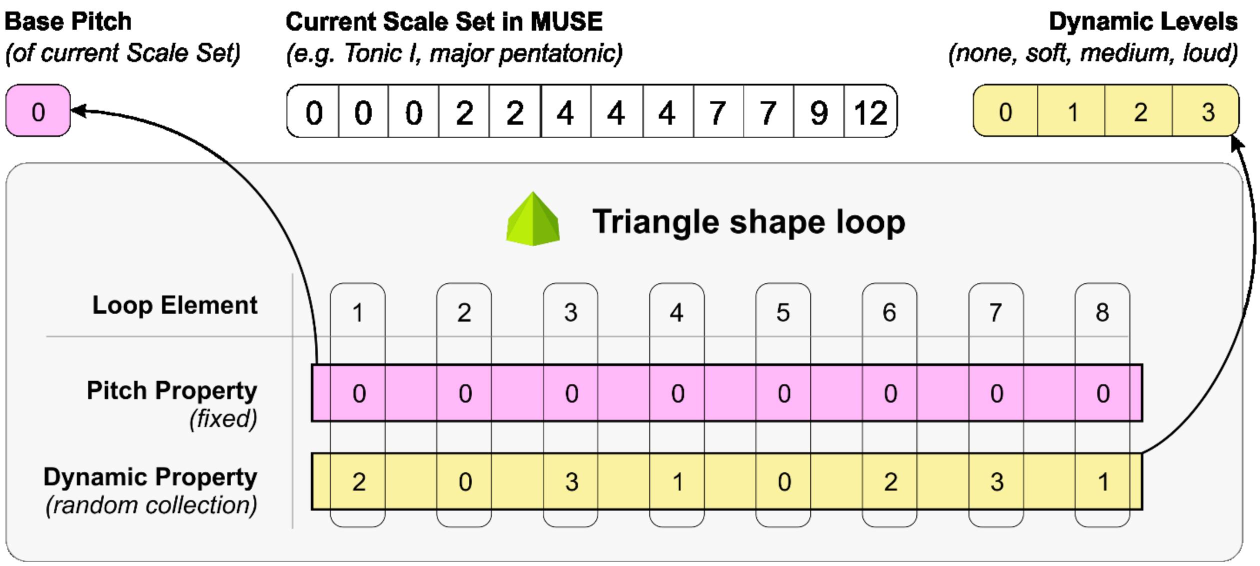


Figure 34: Detail of the Triangle shape in MUSE

The treatment of silent notes in a loop is further discussed in the next section. The last shape available in MUSE is the Pentagon (Figure 35) which has both the Pitch and the Dynamic collections pseudo-randomized, creating the most animated pitch/velocity contours in MUSE.

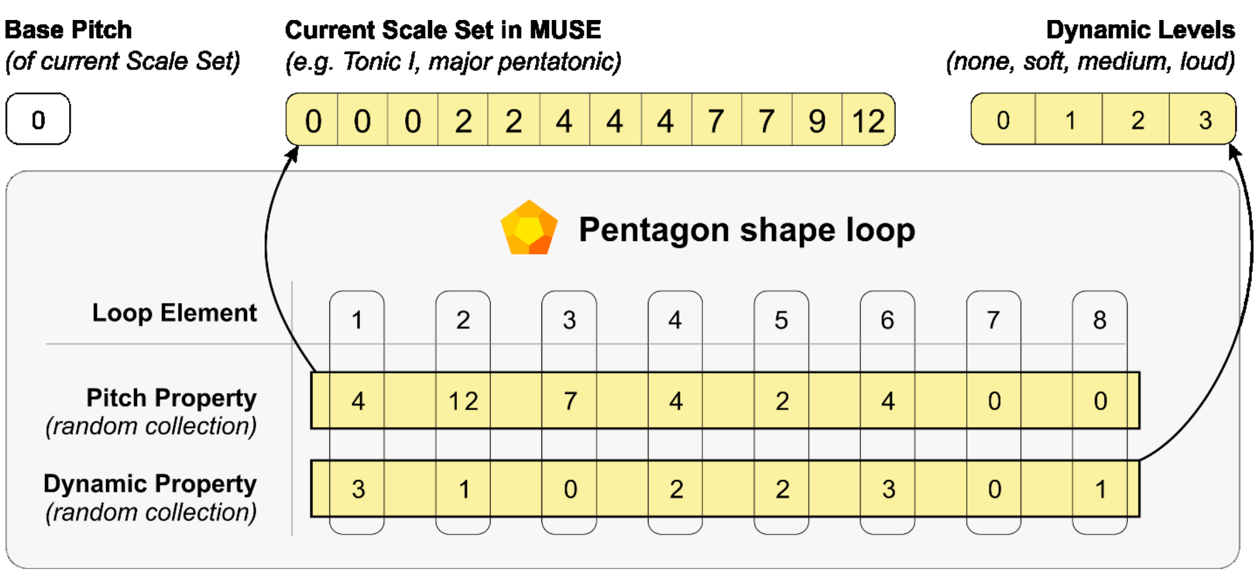


Figure 35: Detail of the Pentagon shape in MUSE

Challenges and Solutions

The design choices of using a grid layout together with the step sequencer approach using sampled percussive sounds, led to some aesthetic issues impacting the sound output and thus the overall game experience. This section explains the issues encountered in implementing the audio engine of MUSE and the approach taken in addressing these challenges.

Amplitude buildup.

At very slow speeds (i.e. long note values within a loop) the percussive/plucked sounds of MUSE decay naturally before the note value's ends. At high speeds however, sounds need to be trimmed to the note value's length to prevent an unwanted amplitude build up. This is achieved through the Web Audio API's `setTargetAtTime` method, which allows scheduling an amplitude event at a specific time. This method takes three arguments: target value (i.e. desired amplitude), start time, and a time constant parameter. The time constant affects the exponential decay rate of this method (i.e. the larger the value, the slower the transition).

These amplitude parameters are calculated in real-time for every Sound Block and depend on its current shape and speed (i.e. note value). The Square shape, which repeats the same pitch over and over again, has to have minimal overlapping of sounds, in order to prevent undesirable artefacts at high speeds (i.e. short note values). The Triangle shape loop also needs its notes to be cut short through a clean, fast fadeout in order to bring out the Triangle's rhythmic character. The Circle and the Pentagon get a slightly different amplitude treatment at high speeds due to their more melodic character. As already shown, Circles and Pentagons have a randomly generated collection of pitches. Therefore, their fadeout's start time and decay rate are calculated so they allow the loop's sounds to overlap slightly, merging into each other, thus adding to the more melodic character of these shapes. At very high speeds, Circles and Pentagons become almost harmonic in character due to the intentional overlap of notes through the amplitude fadeout. Another implementation feature responsible for this harmonic effect is the use and implementation of reverb as discussed in the Reverb section.

However, regardless of shape, the fadeout parameters are directly proportional with the loop's note duration based on its current speed. The shorter the note value (i.e. the loop's speed), the sooner the fade out will start and the shorter it will be.

Syncopation treatment.

Another use of the fadeout function concerns the rhythmic character shared by both the Triangle and the Pentagon shapes. As already shown, a loop element (i.e. note) can take a velocity value of zero, which means that no sound is produced at that particular time/position in the loop. Although "silent" notes within the loop lead to the creation of rhythmic patterns, testing of MUSE showed that pure silence (i.e. musical rests) within these patterns sounded somewhat mechanical and artificial. Therefore, the length of a sounding note (i.e. velocity value 1 - 3) followed by silence (i.e. velocity value of 0) is dynamically prolonged based on the silence length. Figure 36 illustrates the velocity values of a loop (for Player 2) and the perceived musical output after applying the modified fadeout to it; the indicated velocity values represent: 0 – no sound, 1 – soft, 2 – medium, and 3 – loud. As a side note, as specified in the Sound sampling section, the velocity value is related more to the sound's color than its amplitude. Although a velocity value of 3 produces the loudest sound, the timbre of that sound is also the brightest compared to lower velocities.

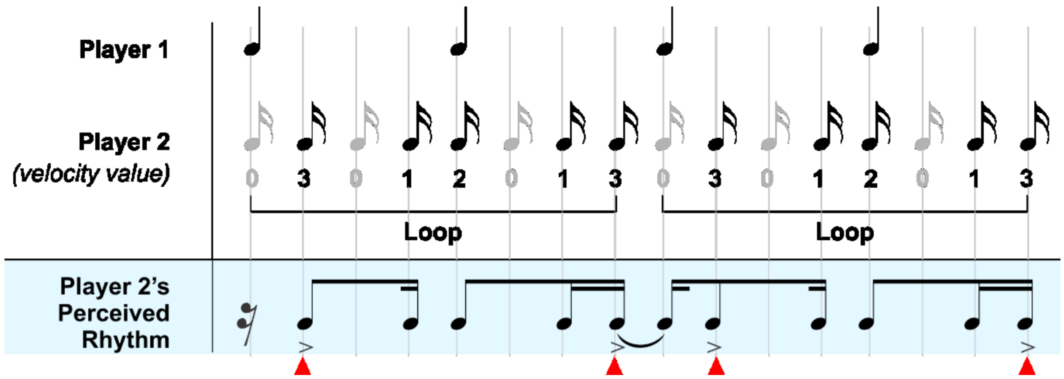


Figure 36: Detail of rhythm syncopation in MUSE. Red triangles represent off-beat accents.

If more than one silent note follows a sounding note (i.e. velocity > 0), the fadeout start time of the sounding note is delayed till the end of the last silent element. The fadeout function also addresses the situation where several consecutive silent notes span across two iterations of the loop as shown in Figure 37.

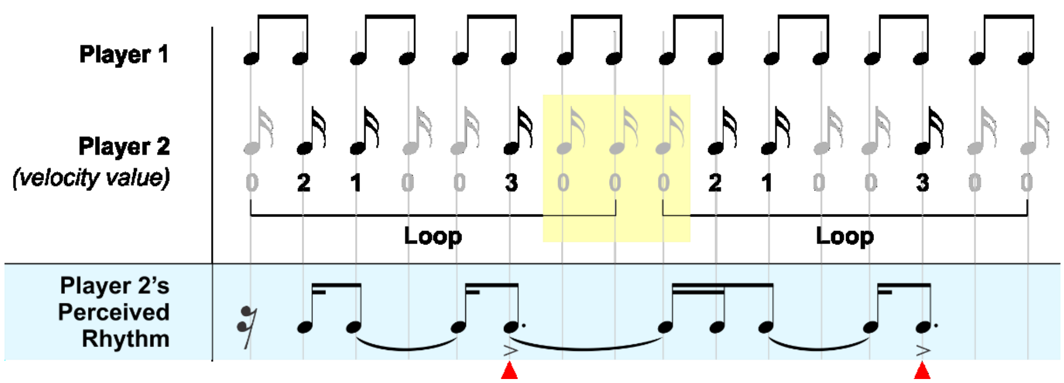


Figure 37: Detail of syncopation occurring over two iterations of the loop

Sound mix.

Along with the obvious changes in a loops' speed (i.e. note values) and its range (i.e. octaves) due to the displacement of a Sound Block on the grid, there is also another level of audio processing that is less evident to the player which involves the amplitude (i.e. volume) and the reverb levels of a Sound Block.

Perceived loudness.

During initial stages of design I strongly considered implementing game actions allowing players direct control over the amplitude and the time-domain signal processing (e.g. reverb, delay, etc.) of their sound loops. Preliminary tests of this approach revealed that novice users might quickly steer towards a less-than-desirable musical output if given these controls due to the finer musical skills required in manipulating these parameters in real time.

These tests also showed that the quality of the overall musical output was influenced by the loops' location on the grid. The farther from the centre of the grid loops were displaced (i.e. extreme octave ranges and/or loop speeds), the more they negatively affected the flow of the music and thus its quality. The reason behind this issue was due to a combination of two design choices I initially made: MUSE's grid design and the selection of sound samples (i.e. instruments) for the loops.

A change in the speed of a loop (i.e. across the columns on the grid) impacted its perceived amplitude with respect to the other loops, in that the faster the loop played, the "louder" the sound appeared to be. Similarly, the slower the speed (i.e. the longer the note values), the softer the loop seemed to sound compared to the other sounding loops. As already presented in the Audio System section, all the sounds in the game were sampled from either pitched percussion or plucked instruments. These sounds share a similar amplitude envelope: a fast, loud attack with a short decay, followed by a moderate-length gradual sustain and release. Since the loudest part of any of these envelopes is the sharp attack, it explains why the perceived increase in amplitude occurred when a loop played at a fast speed. Likewise, since these envelopes decay rather quickly (i.e. non-sustained sounds) it shows why at low speed the loop seemed to lose amplitude. Moreover, the range or octave a loop currently plays at affects the length of the sounds; the higher the octave, the shorter the sample and vice-versa. This is a direct consequence of the pitched percussion and plucked string instrument samples; low frequency sounds last longer, while high-frequency ones end rather quickly. This resulted in an unbalanced distribution of the perceived amplitudes across the grid and made for rather abrupt changes in the flow of the music when a user displaced his Sound Block on the grid. Since giving the control over these parameters to the user threatened the main design goal of the game (that of producing a subjective but pleasant musical experience for novices), I created a fixed table of gain offset values for all the grid tiles of MUSE (i.e. for each range / speed combination) in order to achieve a relatively balanced mix of the sound output. These values (Figure 38) were chosen through extensive game play by testing countless layouts and combinations of Sound Block shapes on the grid.

Highest Octave

Lowest Speed

Highest Speed

+1.8	+1.0	+0.6	0	-0.2	-0.6	-1.2	-1.4
+1.6	+1.0	+0.5	0	-0.2	-0.6	-1.1	-1.3
+1.5	+0.8	+0.4	0	-0.2	-0.6	-1.0	-1.2
+1.2	+0.6	+0.3	0	-0.2	-0.6	-0.8	-1.0
+1.0	+0.5	+0.2	0	-0.2	-0.6	-1.0	-1.0
+0.4	+0.2	0	-0.2	-0.4	-0.6	-1.0	-1.2
+0.2	0	0	0	-0.3	-0.7	-1.0	-1.3
0	0	0	0	0	-0.6	-1.0	-1.4

Lowest Octave

Red : positive gain offset **Green** : no gain offset **Blue** : negative gain offset

Figure 38: Gain Offset Values as stored in MUSE's engine

Reverb.

Following the implementation of amplitude control within MUSE's audio engine, I chose to enhance the color of the sound output by adding a convolution reverb to the audio signal chain using Web Audio API. Just as a specific gain offset value corresponds to each individual tile in the game, a pair of values representing dry/wet (reverb) signals is also applied in real-time to a Sound Block depending on its position on the grid. There are eight distinct pairings of dry/wet signal available as illustrated in Figure 39:

Setting	1	2	3	4	5	6	7	8
Dry	1.0	1.0	1.0	1.0	0.95	0.85	0.6	0
Wet	0.05	0.15	0.2	0.25	0.3	0.4	0.6	0.8

Figure 39: Reverb Dry/Wet values as stored in MUSE's audio engine

The use of reverb aimed at enhancing the goal partially achieved by the gain offset: to create a balanced mix among all voices, regardless of the configuration and/or shapes of the Sound Blocks on the grid, and within the current design and implementation constraints. Since the early prototyping stages of MUSE I was preoccupied with the design challenge of producing harmonic sound from a Sound Block (i.e. instrument) alone. The challenge was to achieve this while keeping the interface and game play as simple as possible for my novice users. Through a serendipitous discovery during game testing, while applying a high level of reverb to the Sound Blocks, I found that a decently pleasant harmonic effect was achieved in the high speed area of the grid (i.e. very short note values) when using the Circle shape (i.e. a collection of pitches but only one fixed velocity). This discovery led to the creation of a matrix of wet/dry reverb value pairs to complement the gain offset on each tile.

Highest Octave

Lowest Speed

+1.8 7	+1.0 7	+0.6 6	0 5	-0.2 5	-0.6 6	-1.2 7	-1.4 8
+1.6 7	+1.0 7	+0.5 6	0 5	-0.2 5	-0.6 6	-1.1 7	-1.3 8
+1.5 7	+0.8 7	+0.4 6	0 5	-0.2 5	-0.6 6	-1.0 7	-1.2 8
+1.2 7	+0.6 7	+0.3 6	0 5	-0.2 5	-0.6 6	-0.8 7	-1.0 8
+1.0 7	+0.5 6	+0.2 5	0 4	-0.2 5	-0.6 6	-1.0 6	-1.0 7
+0.4 6	+0.2 5	0 4	-0.2 3	-0.4 4	-0.6 5	-1.0 6	-1.2 6
+0.2 5	0 4	0 3	0 2	-0.3 2	-0.7 3	-1.0 4	-1.3 5
0 4	0 3	0 2	0 1	0 1	-0.6 2	-1.0 3	-1.4 4

Highest Speed

Lowest Octave

Figure 40: Reverb and Gain Matrix: blue numbers show wet/dry signal pairings as presented in Figure 39 (with higher numbers representing higher reverb levels) while grey numbers show the gain offset as specified in Figure 38.

This matrix shows that the middle to high registers in both very slow and very fast speed extremes on the grid receive the highest amount of reverb (values 7 and 8 in the table shown in Figure 39). In the very slow/high register regions of the grid where MUSE's percussive sounds decay sooner compared to the note's value will benefit from the high amount of reverb tail. The fastest column on the grid in its mid-high register applies dry/wet pair number 8 which contains only wet signal (i.e. 0/0.8 dry/wet pair) to the loop. In the absence of a dry signal, each note in the loop is rendered without any of its percussive attack, allowing only the sustain and release part of the envelope to be prolonged so that it overlaps with the next occurrence of the same note element in the loop. Since all notes in the loop overlap on themselves, the perceived effect is that of a sustained chord made up of all the notes in the loop.

Moreover, the distribution of scale pitches within a "sustained" loop influences somewhat its timbral quality. Even when two consecutive Circle instances (no change in tile position) end up having the same pitches selected from the Scale Set, the quantity of pitches in the loop affects the perceived timbral quality of the sustained "chord" as shown in Figure 41. The reason for the timbral difference is due to the random amplitude applied to the constituent pitches.

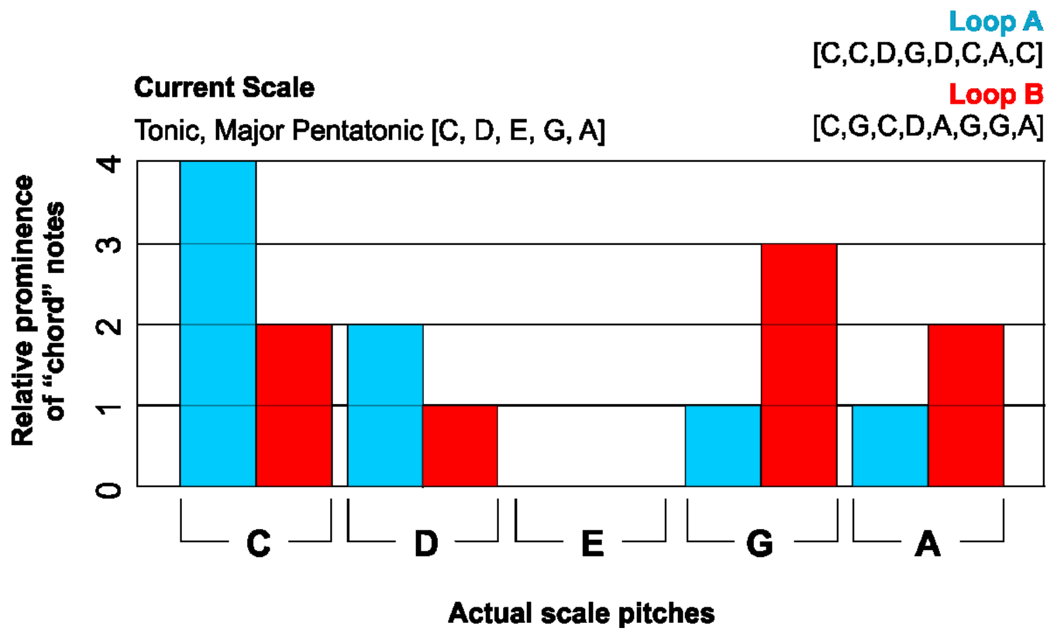


Figure 41: Detail of Timbral differences of similar sustained sound loops in MUSE

Chapter 5 Evaluation and Conclusion

User Studies

To evaluate MUSE against my proposed design goals, I ran two user studies that took place a few weeks apart. No changes or improvements were made to the application between these studies. In each study, the group size was the maximum allowed for the application, of four participants. The group in the first study (group A) consisted of participants of mixed age groups (two 15 – 24, one 31 – 40, and one 10 - 14). The second study had two groups of participants (group B and C), all aged 10 – 14.

Study structure.

The study unfolded as follows:

- The participants watched a 10 min. video tutorial on MUSE (Appendix A.1 MUSE Video Tutorial)
- The participants played a 20 minutes session of MUSE (Appendix A.2 MUSE Study Gameplay (group A))
- The participants completed a 25 point / three section questionnaire (Appendix B.1 MUSE User Study Questionnaire):
    - Section I: gender, age range, music training level, group musical experience, familiarity with gamepad controllers,
    - Section II comprising of questions related to the direct experience of playing MUSE, and

- Section III: short quiz on the basic features of MUSE as presented in the video tutorial and featured in-game.

Results.

Walk-up and play.

The results of the studies indicate that MUSE achieved the "walk-up and play" objective – the main requirement of collaborative music interfaces for novices. 66.7% of participants reported understanding of the game actions as "easy" and "very easy" after watching the video tutorial and participating in the game session. On the "ease on making" music in MUSE, 33.3% reported that it was "very easy", 58.3% - easy, and 8.3% - somewhat easy. Most participants reported they would remember all or most game actions (75%) if they were to play again. On the matching of game actions to gamepad controls, 91% of participants reported they were perfectly or well matched. The results of section III of the questionnaire (short quiz on the basic features of MUSE) indicate that the majority of participants (83%) recalled correctly the main actions available in MUSE as well as their musical behavior in the game.

MDA aesthetic goals.

The results indicate that the aesthetic goals for MUSE based on the MDA design framework were achieved as follows:

1. *Sensation* (a no-wrong-notes, non-disturbing quality of the music output as reported by participants): 91% of participants rated the music created in the game as "very pleasant" and "pleasant". When asked how often they thought the music sounded off-key (wrong

notes, out of tune) 66.6% reported “never” or “not often”, while 25% reported “sometimes”.

2. *Expression* (game as self-discovery, access to creative controls): most participants reported “always” looking forward to their own turn in the game (83.3%). On wanting to try out a new game action on their turn, 16.7% reported “always”, 50% - “often”, and 33.3% - “sometimes”. On what they enjoyed most in MUSE, participant reports include: “I liked the controls and how easy they were to move / change the music”, “Playing with friends and seeing how they were going to change the music”, “I enjoyed how much variety there was in types of sounds, rhythms and music”, “Interacting with players and affecting sounds”, “The ability to control other blocks”.
3. *Submission* (game as past-time, unrushed exploration of musical possibilities, and game flow in a real-time collaborative setting): 75% of participants found the turn time to be “fair” (i.e. not too short or too long), while 16.7% found it somewhat short. Most participants reported being curious in seeing and hearing how the other players changed the music while waiting for their turn (83%).

Participants reported having a positive first-time user experience with MUSE and described it using the following adjectives: “fun” – 66.7%, “exciting” – 41.7%, and “interesting” – 33.3%. The self-rated music training level of the participants was: “none” – 25%, “beginner” – 16.7%, “intermediate” – 50%. The group musical experience (choir, band, ensemble, or orchestra) was reported as: “none” – 33.3%, “beginner” – 33.3%, “intermediate” – 25%. Most participants reported their familiarity with gamepad controllers as “not at all” or “not much” (66.6%). In terms of being acquainted with each other, all participants in group B knew each other before the study, three participants in group C were already acquainted, while group A had only two participants who knew each other prior to playing MUSE.

Discussion

The aim of this research was to explore the potential of collaborative musical interfaces for novices to offer increased creative affordances without compromising the “walk-up and play” characteristic or negatively affecting the perceived quality of the musical output.

Overview.

I suggested that achieving increased creative controls in collaborative interfaces for novices is possible by anchoring the design not solely in the game domain or the music domain but in a new conceptual paradigm, that of a “game of music” that draws upon and takes into account the particularities, limitations and expectations of novice users in a live, collaborative music context. I identified a number of strengths and weaknesses of novice users in regards to music perception, as well as novices’ expectations of a collaborative music-making context and used them as a basis for the user-centered approach taken in creating MUSE. Throughout multiple design and testing iterations I strove to place the users and their experiences and interaction with the interface (both individual and collaborative) at the core of the design process. By following the MDA design framework (mechanics, dynamics, and aesthetics), I set out to achieve three distinct aesthetic goals: sensation, expression, and submission. I derived the game interactions, and subsequently the mechanics, rules, and components from the proposed aesthetics goals while constantly evaluating the “walk-up and play” characteristic as well as the quality of the music output as perceived by participants.

Sensation as perceived quality of music output.

According to the results of the study, the participants found the music they created in MUSE pleasant for the entire duration of the play session (20 minutes). I present below the most important factors that I believe contributed significantly to the perception of the music created as "pleasant".

The generally consonant harmonies of MUSE are based on the tonal music system which is equally familiar to people of all music training skills, be they beginners, intermediate, or experts. Popular music is almost entirely based on the tonal music system, and thus novices are exposed to its predominant consonant harmonies almost on a daily basis. This general familiarity with the tonal music system contributed to perceiving the music created in game as "pleasant".

MUSE's emerging musical form resembles, to a certain extent, the minimal music genre in that the main musical elements of MUSE are repeating groups of notes, or "loops" and that musical change tends to occur rather gradually. Although the music in MUSE can be radically changed in one game action, in general, the presence of "loops" as main musical materials ensures a balanced degree of ongoing musical transformation, and provides participants enough time to get a "feel" for the musical character at any given time.

The limited availability of the number of types of "shapes" at a given time can be found in many tabletop and computer strategy games as limited game resources. By limiting availability, players give more weight to the significance of each resource and choose more carefully when to use a particular resource. However, I made this design choice based on music composition techniques. The character of the music can be greatly changed simply by altering the distribution of music roles (i.e., foreground, middle ground, and background) among the available instruments of the ensemble. Whereas foreground musical levels are usually more animated, both at pitch and dynamic levels, background levels are less complex and serve as support against which the foreground is presented. The availability of a given shape in MUSE (i.e. maximum of four squares, three triangles, two circles, and one pentagon) is inversely proportional to its relative sounding complexity. I implemented this feature in order to ensure a balanced distribution of music roles for any possible combinations of shapes on the grid and to keep the perceived musical quality from getting too close to the "unpleasant" end of the spectrum for my novice users. I would also note that very interesting musical effects can definitely be achieved by implementing a different availability matrix (e.g., inversed availability). It would require musical knowledge and access to lower-level musical parameters to achieve results that would still be considered musically "pleasant" by wide audiences.

The reports of a small percentage of participants that the music "sometimes" sounded off-key (wrong notes, out of tune) coincides with a small number of instances in which most sound blocks were placed towards the edges of the grid, at extreme speeds and/or ranges. These less-frequently occurring configurations indeed brought novelty to the music flow, but posed some challenges to novice users in perceiving them as consonant harmonies (which they were in actuality) due to their extreme pitch and speed characteristics. What is important to note here however, is that no participant rated the music created in the game "unpleasant" or "very unpleasant" but "pleasant" – 58.3%, "very pleasant" – 33.3%, with only 8.3% rating it as "acceptable".

Expression as musical creativity.

All three groups exhibited a similar behavior curve in exploring MUSE's controls and actions during gameplay. Participants usually started by timidly moving their block on the grid in the central regions, followed by a change of shape and ending their turn. Later in the game, they progressively increased the range of motion, moving their block around the edges of the grid, and started changing and/or "refreshing" their block shape more often, as well as exploring the "All Blocks" actions category. Some players exhibited "cautious" gameplay by taking relatively long to explore game actions with higher musical impact while other players not only looked forward to going through all the game possibilities as soon as possible, but started quite early performing action combinations which resulted in more complex musical change. One example of expressing their creativity was alternating quickly between the "All Blocks" chord change, the "Move All Blocks", and "Refresh" all the blocks on their turn. A small number of players explored in depth what could be described as a "solo" feature, which although not explicitly defined in MUSE, results from the rapid alternation between, and combination of "Move Block", "Change Shape", and refreshing of the shape. This "combo" action produces a very animated, complex musical foreground presence of that instrument (considerably less static compared to the otherwise stationary repeating loop), making it stand out against the supporting musical background provided by the other blocks (i.e. instruments). I believe that by giving users the possibility to use their personal creativity and skill in performing a certain sequence of combinations as per their own choice, MUSE gives players a strong sense of musical / gameplay agency that increases the potential for long-term engagement with the interface.

The "solo" action mentioned above sometimes resulted in taking other block's place on the grid, which meant the removal and muting of that block. I implemented the muting feature for two reasons: first, to provide a way for a player to remove another player's block if they found that block to negatively impact the music quality at a given point in time, and second, to provide the ability to remove more than one voice (i.e. block) as a creative control in changing the musical output. From an interaction perspective, the intentional or accidental first-time removing (i.e. muting) of another player's block triggered an interesting reaction in participants: although no points are gained or lost in MUSE, most players who had been "muted" by others took "revenge" on their turn and muted back the player who had previously muted them. While players expressed their enjoyment of this little game of "cat-and-mouse" by laughing and commenting out, the general attitude changed quite rapidly from a democratic building of the music output to a social game for dominance. The focus changed from playing "with" music to playing "with" or rather against other players. The feature of controlling not only one's block, but all blocks including the option of temporarily muting one or more players was inspired by my participation in directed improvisational ensembles and by John Zorn's Cobra where although there is a "leader", members of the ensemble can form a "guerilla" and take over the course of the action for a limited time. I believed that MUSE's creative space would benefit from presenting itself as a democratic setting where performers have the means to rise to the composer / conductor (or even dictator) status just as quickly as they can lose it. All the groups in both studies went through the "cat-and-mouse" emergent play. However, none of the groups spent more than a few minutes in this revengeful interaction. Usually, at the end of a session of "cat-and-mouse" – when only one block was on the grid (i.e. being heard) – the participants got back on the grid on their turn and began building back the music, and started showing interest again in choosing the tile where to place their block and the shape of the block with respect to the other active voices. I can only assume that the reason this conflict was usually short-lived is due to the real-time nature of MUSE: the gameplay constantly produces some musical output. From a music perception point of view, playing "cat-and-mouse" resulted in some sort of musical relief (i.e. less active voices) but its nature was rather singular and possibly felt to participants (at least subconsciously) as not being rewarding enough musically to be entertained for a long time. Lastly, an interaction observed frequently at the beginning of play was imitation of actions: in one instance after a player placed their block in a corner (i.e. extreme pitch and speed range) the following player placed their own block in a different corner. Also, once a player explored the "solo" action, more players tried combining actions on their turn. The turn-taking mechanic and the co-location of players (i.e. same physical space) allowed participants to see not only the other players' actions on the screen but the actual manipulation of gamepad controls, thus learning from and getting inspired by each other.

Submission as pastime.

I was pleasantly surprised when users from all groups informally reported at the end of the play session that the playing felt much shorter than 30 minutes. This suggests that participants experienced some level of psychological "flow" during gameplay (Csikszentmihalyi 1990) – the mental state characterized by a high degree of immersion and involvement where users have a sense of agency and pleasure in the activity itself.

These studies did not aim to measure the level of psychological flow of the participants. However, it is worth pointing out how the six factors of "flow" relate to MUSE as platform or to the actual playing experience as reported by the participants:

Flow factors	**Relatedness to MUSE as platform or to the playing experience as reported by participants**

1. Intense and focused concentration on the present moment	- Facilitated through real-time creation / performance of music
2. Merging of action and awareness	- Real-time interaction with immediate visual and audible feedback
3. A loss of reflective self-consciousness	- N/A (see below)
4. A sense of agency over the activity	- Creative control over the music output - “No-wrong-notes” environment, easy to understand actions and controls (supported by the study results)
5. A distortion of temporal experience	- Participants informally reported that the session felt shorter than it was
6. Experience of the activity as intrinsically rewarding	- Participants described the experience as “fun”, “exciting”, and “interesting” - The perceived quality of the music was rated as pleasant (58.3%) and very pleasant (33.3%)

The loss of reflective self-consciousness refers to the psychological state of losing the awareness of oneself as a social actor. This state may have oscillated between a higher degree during “solo” endeavors (i.e. combining game actions resulting in an increased amount of continuous musical change) and a lower degree while fighting for grid dominance (i.e. the “cat-and-mouse” game of eliminating/muting other players).

Supported by the study results, the immersion of participants in the first-time experience of playing MUSE relates to a number of factors that derive from the design approach: the anticipation of one’s turn, the unpredictable game state changes (i.e. grid configurations), the emerging musical and player interactions, the balance between the perceived challenges of the game and the perceived abilities of players to overcome those challenges, the “non-punishing” quality of musical results regardless of user actions, but nonetheless the opportunity to produce music output of increased appeal.

Participant suggestions.

A number of suggestions came from participants who self-rated themselves as having intermediate musical skill:

- An increased number of players (6 - 8), forming “teams”,
- The option to store a moving pattern / block shape change pattern and use it to replace one’s loop while that player waits for their turn,
- The option of single-player composition in real-time with MUSE,
- The ability (even if for short periods of time) for all players to be able to control their blocks simultaneously – “free-for-all”,
- The option to control one’s own block audio volume, and
- The option to save one or more grid configurations (block shapes and their positions) for later retrieval.

To no surprise, these suggestions show that users with intermediate musical training are looking for more creative options and freedom to create and express musical ideas compared to novices. This shows some potential of MUSE to expand into two directions: as composition tool (e.g. storing a series of events, saving grid configurations) and as performance instrument (e.g. the “free-for-all” mode, own block volume control, the “team” approach).

Future ideas.

As a designer, I believe that in its current form, MUSE is a successful prototype of a collaborative music interface for novices that offers increased creative controls and fulfills my initial design objectives. As a musician however, throughout the countless testing sessions of the prototype I grew somewhat tired by the relatively low expressive power of MUSE and its current creative opportunities. The expanding of MUSE through the implementation of more creative controls is something I have constantly reflected upon during the design of this prototype. However, while a new feature could improve the user experience it would nonetheless have to be thoroughly user tested to ensure that none of the design goals are being compromised or the achieved user experience is diminished. I believe that a better approach to increase the creative affordances of MUSE (other than just adding new controls and functionality to the game) is to consider MUSE as a platform for collaborative interactions with the current prototype being just the "novice-ready, play it now" game mode. New game "modes" can be easily created for this platform with each mode focusing on specific target users and/or uses, not only "walk-up and play" for novices. Examples include:

- *Real-time performance for an ensemble of intermediate users*: based either on a predetermined plan or open ended interaction, each player would control their block in real-time (i.e. no turn taking), multiple players on a tile are allowed (i.e. no eliminating players from the grid) in an effort to produce a more complex musical output. This mode can also feature increased expressive controls, make use of the analog joysticks of the gamepad for amplitude and timbre control. The players would still benefit from the immediate visual feedback and the state of the entire ensemble during the performance.
- *Single-player interactive tutorial*: this could be implemented so that a user can learn the interface at their own pace. This game mode can present each feature of MUSE progressively through levels, give the user time to practice the controls, suggest and explain from a music point of view why there are differences in the moods of the music

based on the grid configurations (shapes and their placement), award achievement points for completing each of these tutorial steps, and unlock more advanced control features (amplitude, timbre).

- *Team-oriented gameplay*: this mode would have a clear winning goal and require group cooperation in other to succeed. The mode would present itself as a pre-established series of events that the users have to perform in real time in order to win. In other words, the mode is a MUSE "score" that users follow and perform in real-time for points.

MUSE in other applications.

The music "engine" behind MUSE was also used in a research project exploring the use of music in facilitating emotion recognition and expression in autistic children. For this project I programmed MUSE to create an ongoing musical output matching a specific emotion as per the arousal / valence model for emotion classification (Figure 42).

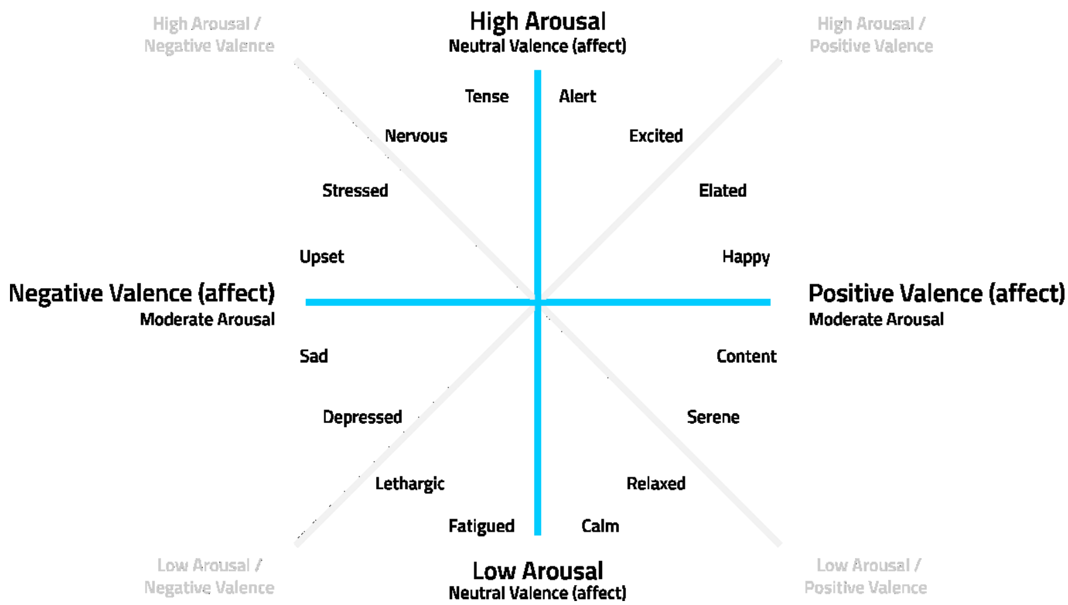


Figure 42: The emotion classification arousal / valence graph

A link to a video demo of the use of MUSE for this project can be found in the Appendix A.5 MUSE System as Music Emotion Mapper / Elicitor.

Upon receiving a new emotion request (from either sensors or an external tablet application) MUSE would progressively transition to a new grid configuration that would produce a music output corresponding with the requested emotion. The transition path to the new emotion would go through intermediate emotions, for example: from Alert (top centre) to Calm (bottom centre), the music "emotion" would pass through the middle of the graph (Neutral emotion).

The music itself would gradually change in a musically meaningful manner, with sound blocks moving to their new grid configuration (locations and shapes) in discrete steps matching the expected emotion transition. As this project required the rendition of no more than nine distinct musical "states" corresponding to nine emotions, I expanded MUSE in several aspects including the availability of musical scales to include other variations of major, minor, diminished and augmented scales, creating variations of the current block shapes to allow finer control over the variation in pitch and velocity inside the generated loop, and removing the limit of shapes on the board (i.e. any combinations of shapes at any time). An example of shape variation is the “soft” shape which changed the percussive character of the original sound samples to a “softer” character. This was achieved by reducing the attack amplitude of the sound sample and indicating that with the letter “s” placed next to the block shape. This relatively small number of expressive enhancements to MUSE’s core system allowed the creation of a larger, richer variety of musical moods and characters.

Summary.

This thesis presented the design, implementation and evaluation of MUSE, a computer-mediated application for collaborative music creation and performance in real-time for novices. In Chapter One I presented my research hypothesis and the motivation for pursuing it, and indicated the general direction in which I would orient the discussion. In Chapter Two, I examined the music literature and presented an overview of the use of chance in music creation and performance. I then placed MUSE as a musical composition within this context, and ended with a discussion on a number of approaches to music indeterminacy and how they informed the design of MUSE. I also explored the “collaborative musical experiences for novices” domain, identified common design approaches, and proposed the “game of music” paradigm as a foundation for achieving my goals. In Chapter Three, I examined the idiosyncrasies of the music and game domains, discussed the “game of music” paradigm from a user-centered design perspective, presented the game design methodology used, and provided an in-depth overview of the rules, game components, and game-play of MUSE. In Chapter Four, I presented the technical and implementation details of MUSE, described the system design in relation with game components and user actions at interaction level. I also mentioned a number of system design challenges I encountered during MUSE's implementation. In Chapter Five, I presented the results of two studies evaluating MUSE's proposed design goals, discussed MUSE in the light of these studies, and reflected on ideas for future explorations.

I believe the approach I took in designing MUSE as a "game of music" – with its mechanics, dynamics, and aesthetics, achieved the proposed goals of fostering immersion and engagement in the direct, collaborative interaction in playing "with" music for wide audiences. I also hope that MUSE as a platform will provide inspiration for further research in providing novices opportunities for participatory music activities towards fostering musical creativity and hopefully, future engagement in formal music training and practices.

References

Allsen, J Michael. 1994. “Johannes Ockeghem,‘ Missa Cuiusvis Toni’, Ed. George Houle.” *Notes* 51 (1). Music Library Association.: 410.

Arrasvuori, Juha, Marion Boberg, Jussi Holopainen, Hannu Korhonen, Andr’es Lucero, and Markus Montola. 2011. “Applying the PLEX Framework in Designing for Playfulness.” In *Proceedings of the 2011 Conference on Designing Pleasurable Products and Interfaces*, 24:1-24:8. DPPI ’11. Milano, Italy: ACM. doi:10.1145/2347504.2347531.

Bengler, Ben, and Nick Bryan-Kinns. 2013. “Designing Collaborative Musical Experiences for Broad Audiences.” In *Proceedings of the 9th ACM Conference on Creativity and Cognition*, 234–42. Sydney, Australia: ACM. doi:10.1145/2466627.2466633.

Bernard, Jonathan W. 1993. “The Minimalist Aesthetic in the Plastic Arts and in Music.” *Perspectives of New Music*. JSTOR, 86–132.

Bharucha, Jamshed J., Meagan Curtis, and Kaivon Paroo. 2006. “Varieties of Musical Experience.” *Cognition* 100 (1): 131–72. doi:10.1016/j.cognition.2005.11.008.

Blaine, Tina, and Sidney Fels. 2003. “Collaborative Musical Experiences for Novices.” *Journal of New Music Research* 32 (4). Taylor & Francis: 411–28.

Blaine, Tina, and Tim Perkis. 2000. “The Jam-O-Drum Interactive Music System: A Study in Interaction Design.” In *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, 165–73. ACM.

Brackett, John. 2010. “Some Notes on John Zorn’s Cobra.” *American Music* 28 (1). University of Illinois Press: 44–75.

Brathwaite, Brenda, and Ian Schreiber. 2008. *Challenges for Game Designers*. Vol. 1. Rockland, MA, USA: Charles River Media, Inc.

Carl, Robert. 2009. *Terry Riley's in C*. Oxford University Press.

Cassidy, Gianna G, and Anna M J M Paisley. 2013. "Music-Games: A Case Study of Their Impact." *Research Studies in Music Education* 35 (1). SAGE Publications: 119–38.

Christensen, Helen, Kathleen M Griffiths, and Louise Farrer. 2009. "Adherence in Internet Interventions for Anxiety and Depression: Systematic Review." *Journal of Medical Internet Research* 11 (2). JMIR Publications Inc., Toronto, Canada: e13.

"ChucK." 2017. Accessed September 23. http://chuck.cs.princeton.edu/.

Costello, Brigid, and Ernest Edmonds. 2007. "A Study in Play, Pleasure and Interaction Design." In *Proceedings of the 2007 Conference on Designing Pleasurable Products and Interfaces*, 76–91. ACM.

Cremaschi, Alejandro M, Ksenia Ilinykh, Elizabeth Leger, and Nathan Smith. 2015. "Students Who Quit Music Lessons: Recent Research And Recommendations For Teachers." *MTNA E-Journal* 6 (3): 15–26. http://ezproxy.lib.ucalgary.ca/login?url=http://search.proquest.com/docview/1683484556?accountid=9838.

Csikszentmihalyi, Mihaly. 1990. *Flow: The Psychology of Optimal Experience*. Vol. 1st. Harper & Row.

———. 2014. "Toward a Psychology of Optimal Experience." In *Flow and the Foundations of Positive Psychology*, 209–26. Dordrecht: Springer Netherlands. doi:10.1007/978-94-017-9088-8_14.

Deci, Edward L, and R M Ryan. 2002. "Overview of Self-Determination Theory: An Organismic Dialectical Perspective." *Handbook of Self-Determination Research*, 3–33.

Deterding, Sebastian. 2011. "Situated Motivational Affordances of Game Elements: A

Conceptual Model." In *Gamification: Using Game Design Elements in Non-Gaming Contexts, a Workshop at CHI*.

Deterding, Sebastian, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. "From Game Design Elements to Gamefulness: Defining Gamification." In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, 9–15. MindTrek '11. Tampere, Finland: ACM. doi:10.1145/2181037.2181040.

Dourish, Paul, and Victoria Bellotti. 1992. "Awareness and Coordination in Shared Workspaces." In *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work*, 107–14. ACM.

Ehrmann, Jacques, Cathy Lewis, and Phil Lewis. 1968. "Homo Ludens Revisited." *Yale French Studies*, no. 41. JSTOR: 31–57.

Ferrara, John. 2012. *Playful Design*. Rosenfeld Media.

Folkestad, Goran. 2012. "Digital Tools and Discourse in Music: The Ecology of Composition." *Musical Imaginations: Multidisciplinary Perspectives on Creativity, Performance and Perception* 193: 205.

Gagné, Nicole V. 2011. *Historical Dictionary of Modern and Contemporary Classical Music*. Scarecrow Press.

Gaver, William W, John Bowers, Andrew Boucher, Hans Gellerson, Sarah Pennington, Albrecht Schmidt, Anthony Steed, Nicholas Villars, and Brendan Walker. 2004. "The Drift Table: Designing for Ludic Engagement." In *CHI'04 Extended Abstracts on Human Factors in Computing Systems*, 885–900. ACM.

Google. 2015. "Building TV Games." Accessed December 5. https://developer.android.com/training/tv/games/index.html.

Griffiths, Paul. 2001. “Aleatory.” *The New Grove Dictionary of Music and Musicians* 2.

Hallam, Susan. 2010. “The Power of Music: Its Impact on the Intellectual, Social and Personal Development of Children and Young People.” *International Journal of Music Education* 28 (3). SAGE PublicationsSage UK: London, England: 269–89. doi:10.1177/0255761410370658.

Hansen, Anne-Marie Skriver, Hans Jørgen Andersen, and Pirkko Raudakoski. 2011. “Play Fluency in Music Improvisation Games for Novices.” In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 220–23.

Harley, James. 2004. *Xenakis: His Life in Music*. Routledge.

Hatfield, Elaine, John T. Cacioppo, and Richard L. Rapson. 1993. “Emotional Contagion.” *Current Directions in Psychological Science* 2 (3). SAGE PublicationsSage CA: Los Angeles, CA: 96–100. doi:10.1111/1467-8721.ep10770953.

Hedges, Stephen A. 1978. “Dice Music in the Eighteenth Century.” *Music & Letters* 59 (2). JSTOR: 180–87.

Hunicke, Robin, Marc LeBlanc, and Robert Zubek. 2004. “MDA: A Formal Approach to Game Design and Game Research.” In *Proceedings of the AAAI Workshop on Challenges in Game AI*. Vol. 4.

Jordà, Sergi. 2003. “Interactive Music Systems for Everyone: Exploring Visual Feedback as a Way for Creating More Intuitive, Efficient and Learnable Instruments.” In *Proceedings of the Stockholm Music Acoustics Conference (SMAC03), Stockholm, Sweden*. Citeseer.

“jQuery.” 2017. Accessed July 30. https://jquery.com/.

Juul, Jesper. 2002. “The Open and the Closed: Games of Emergence and Games of Progression.” In *CGDC Conf.*

Kennedy, Michael, and Joyce Bourne Kennedy. 2007. *The Concise Oxford Dictionary of Music*. Oxford University Press. doi:10.1093/acref/9780199203833.001.0001.

Klügel, Niklas, Marc René Frieß, Georg Groh, and Florian Echtler. 2011. “An Approach to Collaborative Music Composition.” In *NIME*, 32–35.

Koelsch, Stefan, and Sebastian Jentschke. 2010. “Differences in Electric Brain Responses to Melodies and Chords.” *Journal of Cognitive Neuroscience* 22: 2251–62. doi:10.1162/jocn.2009.21338.

Kurtz, Michael. 1992. “Stockhausen: A Biography, Trans.” *Richard Toop (London and Boston: Faber and Faber, 1992)* 13: 1952–67.

Kylie, Peppler, Downton Michael, Lindsay Eric, and Kenneth Hay. 2011. “The Nirvana Effect: Tapping Video Games To Mediate Music Learning and Interest,” February. International Journal of Learning and Media, MIT Press. https://scholarworks.iu.edu/dspace/handle/2022/14637.

Maconie, Robin. 1976. *The Works of Karlheinz Stockhausen*. Oxford University Press.

Marczewski, Andrzej. 2013. *Gamification: A Simple Introduction*. Andrzej Marczewski.

McTighe, Jay, and Grant Wiggins. 2012. “Understanding by Design{®} Framework.” *Alexandria, VA: Association for Supervision and Curriculum Development*.

Meyer-Eppler, Werner. 1958. “Statistic and Psychologic Problems of Sound.” *Die Reihe* 1: 55–61.

“MIDI History.” 2016. Accessed August 15. https://www.midi.org/articles/categories/midi-history.

Miller, Kiri. 2009. “Schizophonic Performance: Guitar Hero, Rock Band, and Virtual Virtuosity.” *Journal of the Society for American Music* 3 (4): 395–429.

doi:10.1017/S1752196309990666.

Nielsen. 2016. "Nielsen Music 360 Report." http://www.nielsen.com/us/en/press-room/2012/music-discovery-still-dominated-by-radio--says-nielsen-music-360.html.

Nielsen, Jakob. 1995. "10 Usability Heuristics for User Interface Design." https://www.nngroup.com/articles/ten-usability-heuristics/.

Nierhaus, Gerhard. 2009. *Algorithmic Composition: Paradigms of Automated Music Generation*. Springer Science & Business Media.

"NIME." 2017. Accessed September 23. http://www.nime.org/.

Norman, Donald A, and Stephen W Draper. 1986. "User Centered System Design." *New Perspectives on Human-Computer Interaction, L.Erlbaum Associates Inc., Hillsdale, NJ*.

"Number of Adults Participating in the Performing Arts in the United States in 2012." 2016. Accessed June 1. https://www.statista.com/statistics/381497/number-of-adults-participating-in-performing-arts-by-type-us/.

Pallesen, Karen Johanne, Elvira Brattico, Christopher Bailey, Antti Korvenoja, Juha Koivisto, Albert Gjedde, and Synnöve Carlson. 2005. "Emotion Processing of Major, Minor, and Dissonant Chords: A Functional Magnetic Resonance Imaging Study." *Annals of the New York Academy of Sciences* 1060: 450–53. doi:10.1196/annals.1360.047.

Paradise, Ruth, and Barbara Rogoff. 2009. "Side by Side: Learning by Observing and Pitching in." *Ethos* 37 (1). Wiley Online Library: 102–38.

Popa, Iulius A T, Jeffrey E Boyd, and David Eagle. 2015. "MUSE: A Music-Making Sandbox Environment for Real-Time Collaborative Play." In *The 12th Sound and Music Computing Conference, Music Technology Research Group, Dept. of Computer Science, Maynooth University, Maynooth, Co. Kildare, Ireland*, 7–14. Maynooth.

http://smcnetwork.org/system/files/SMC2015_submission_91.pdf.

Priebatsch, Seth. 2010. "TED Talks: The Game Layer on Top of the World." https://youtu.be/-n3buCOXiY8.

Pritchett, James. 1996. *The Music of John Cage*. Vol. 5. Cambridge University Press.

Rovan, Joseph Butch, Marcelo M Wanderley, Shlomo Dubnov, and Philippe Depalle. 1997. "Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance." In *Kansei, The Technology of Emotion. Proceedings of the AIMI International Workshop*, 68–73. Genoa: Associazione di Informatica Musicale Italiana, October.

Schellenberg, E. Glenn. 2004. "Music Lessons Enhance IQ." *Psychological Science* 15 (8). SAGE PublicationsSage CA: Los Angeles, CA: 511–14. doi:10.1111/j.0956-7976.2004.00711.x.

———. 2005. "Music and Cognitive Abilities." *Current Directions in Psychological Science* 14 (6). SAGE PublicationsSage CA: Los Angeles, CA: 317–20. doi:10.1111/j.0963-7214.2005.00389.x.

Sergi, Jordà. 2010. "The Reactable: Tangible and Tabletop Music Performance." In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, 2989–94. ACM.

Simms, Bryan R. 1996. *Music of the Twentieth Century: Style and Structure*. Cengage Learning.

"SuperCollider." 2017. Accessed September 23. http://supercollider.github.io/.

"Using the Gamepad API." 2017. Accessed July 30. https://developer.mozilla.org/en-US/docs/Web/API/Gamepad_API/Using_the_Gamepad_API.

"Web Audio API." 2017. Accessed July 30. https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API.

Weinberg, Gil. 2003. “Interconnected Musical Networks--Bringing Expression and Thoughtfulness to Collaborative Music Making.” In *Massachusetts Institute of Technology Media Laboratory*. Citeseer.

Wilson, Chris. 2013. “A Tale of Two Clocks - Scheduling Web Audio with Precision.” https://www.html5rocks.com/en/tutorials/audio/scheduling/.

Xambó, Anna, Robin Laney, and Chris Dobbyn. 2011. “TOUCHtr4ck: Democratic Collaborative Music.” In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*, 309–12. ACM.

“Yamaha Products History.” 2017. Accessed September 23. https://www.yamaha.com/en/about/history/products/#genre.

Zhang, Ping. 2008. “Technical Opinion: Motivational Affordances: Reasons for ICT Design and Use.” *Commun.ACM* 51 (11). New York, NY, USA: ACM: 145–47. doi:10.1145/1400214.1400244.

Zicarelli, David. 1987. “M and Jam Factory.” *Computer Music Journal* 11 (4). The MIT Press: 13. doi:10.2307/3680237.

Zicarelli, David, Joel Chadabe, John Offenhartz, Antony Widoff, and Richard Lainhart. 2017. “M An Intelligent Musical Instrument.” Accessed September 16. www.cycling74.com.

Zichermann, Gabe, and Joselin Linder. 2010. “Game-Based Marketing.” *Inspire Customer Loyalty through Rewards, Challenges, and Contests* 21 (6): 19.109-128.

Appendix A: Video links

Below are the links to videos featuring MUSE including the tutorial presented to the study participants, gameplay sessions, video presentation of MUSE, and a video demo of the application developed during the research of using music in emotion recognition and expression for autistic children.

A.1 MUSE Video Tutorial

This video was presented to all groups of participants in the study before playing MUSE.

https://youtu.be/nOw2OKDTij0

A.2 MUSE Study Gameplay (group A)

This is a full recording of the gameplay during the first study (group A). The names of participants as shown in game have been blurred out to protect their privacy.

https://youtu.be/oZqrviGPahM

A.3 Video Presentation of MUSE

This video is based on the oral presentation given at the 2015 Sound and Music Computing conference in Maynooth, Ireland.

https://youtu.be/LJP1lIRo05c

A.4 MUSE Gameplay Demo

This video shows a gameplay of MUSE recorded to accompany the oral presentation at the SMC 2015 in Maynooth, Ireland.

https://youtu.be/vwYRiVdCYmo

A.5 MUSE System as Music Emotion Mapper / Elicitor

This is a demo of the application developed during the research of music in emotion recognition and expression for autistic children. This application uses MUSE's music "system" modified to support the requirements of this project.

https://youtu.be/Xu992JSNFWM

Appendix B: MUSE study

B.1 MUSE User Study Questionnaire

Section A

1. Indicate your gender

☐ Male ☐ Female ☐ Prefer not to say

2. Select your age range

☐ 10 – 14 ☐ 15 – 18 ☐ 19 – 25 ☐ 26 – 30 ☐ 31 – 40 ☐ 41 – 50

3. Rate you music training level

☐ none ☐ beginner ☐ intermediate ☐ advanced ☐ expert

4. Rate your musical experience with a choir, band, ensemble, or orchestra

☐ none ☐ beginner ☐ intermediate ☐ advanced ☐ expert

5. You have used a gamepad controller before (such as Xbox or PlayStation controllers)

☐ not at all ☐ not much ☐ quite a bit ☐ often ☐ a lot

Section B

1. Select what best describes what playing MUSE felt like (or write your own answer)

☐ interesting ☐ unusual ☐ exciting ☐ weird ☐ pleasant ☐ boring ☐ fun ☐ annoying
☐ other: ____

2. Rate the music created in the game

☐ very pleasant ☐ pleasant ☐ acceptable ☐ unpleasant ☐ very unpleasant

3. How often do you think the music sounded off-key (wrong notes, out of tune)?

☐ never ☐ not often ☐ sometimes ☐ often ☐ almost always

4. I was looking forward to my own turn in the game

☐ always ☐ often ☐ sometimes ☐ not often ☐ never

5. I was curious to see and hear how the other players changed the music

☐ always ☐ often ☐ sometimes ☐ not often ☐ never

6. I wanted to try out a new game action on my turn

☐ always ☐ often ☐ sometimes ☐ not often ☐ never

7. I found the turn time to be:

☐ very short (*not enough time*) ☐ somewhat short ☐ fair (*just about right*) ☐ somewhat long ☐ very long (*too much waiting*)

8. How difficult was it to *understand* the game actions (what you could do in the game)?

☐ very difficult ☐ difficult ☐ not too difficult ☐ easy ☐ very easy

9. How easy was it for you to *make music* in MUSE?

☐ not easy at all ☐ not too easy ☐ somewhat easy ☐ easy ☐ very easy

10. How many game actions do you think you would remember if you were to play again?

☐ none of them ☐ a few ☐ some of them ☐ most of them ☐ all of them

11. Rate how well the gamepad controls were matched to the game actions (simple and intuitive)

☐ perfectly matched ☐ well matched ☐ somewhat matched ☐ not really matched ☐ not matched at all

12. Please write in a few words what you enjoyed most in MUSE

____

13. Please write in a few words what you really disliked in MUSE

____

Section C

1. Moving a block (a sound maker) to the right on the board makes it:

☐ play slower ☐ play faster ☐ sound lower ☐ sound higher ☐ not sure

2. Moving a block up on the board makes it:

☐ play slower ☐ play faster ☐ sound lower ☐ sound higher ☐ not sure

3. A block (a sound maker) placed in the bottom-left corner sounds:

☐ high and fast ☐ low and slow ☐ high and slow ☐ low and fast ☐ not sure

4. A Triangle shape makes a block sound:

☐ Jumpy ☐ Tune-y ☐ Steady ☐ Jumpy-Tune-y ☐ not sure

5. Two blocks can be on the same tile on the board at the same time:

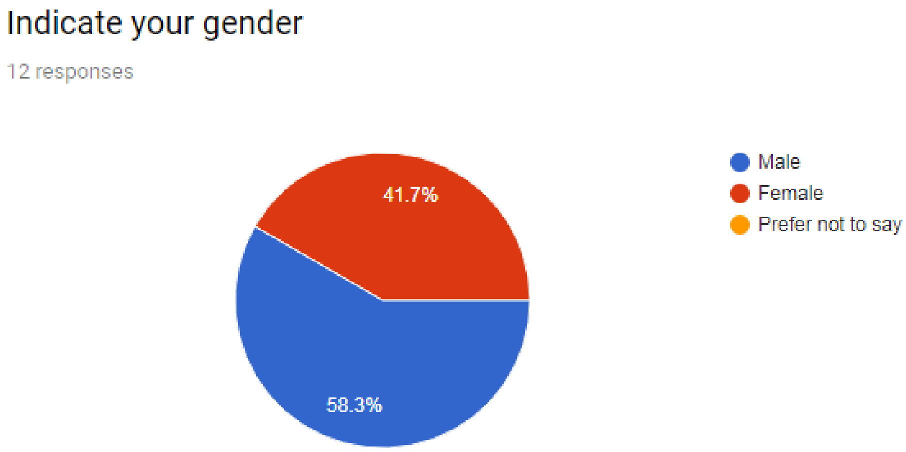
☐ true ☐ false ☐ not sure

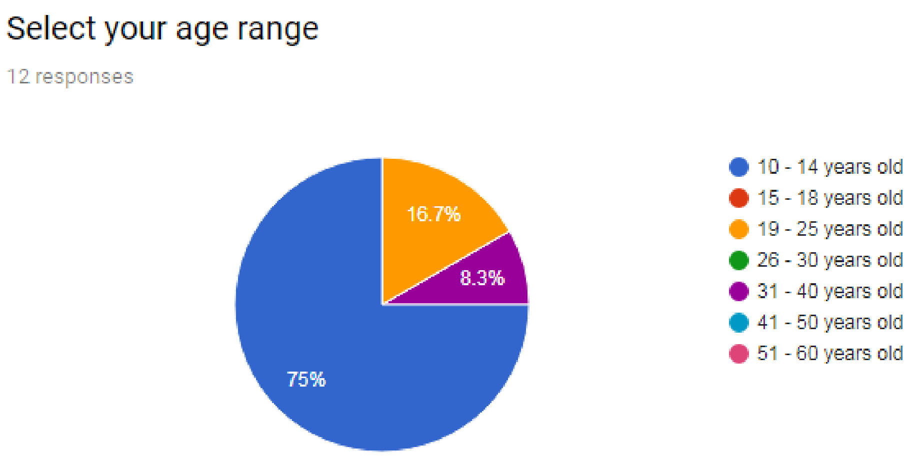
6. You can move all the blocks on your turn:

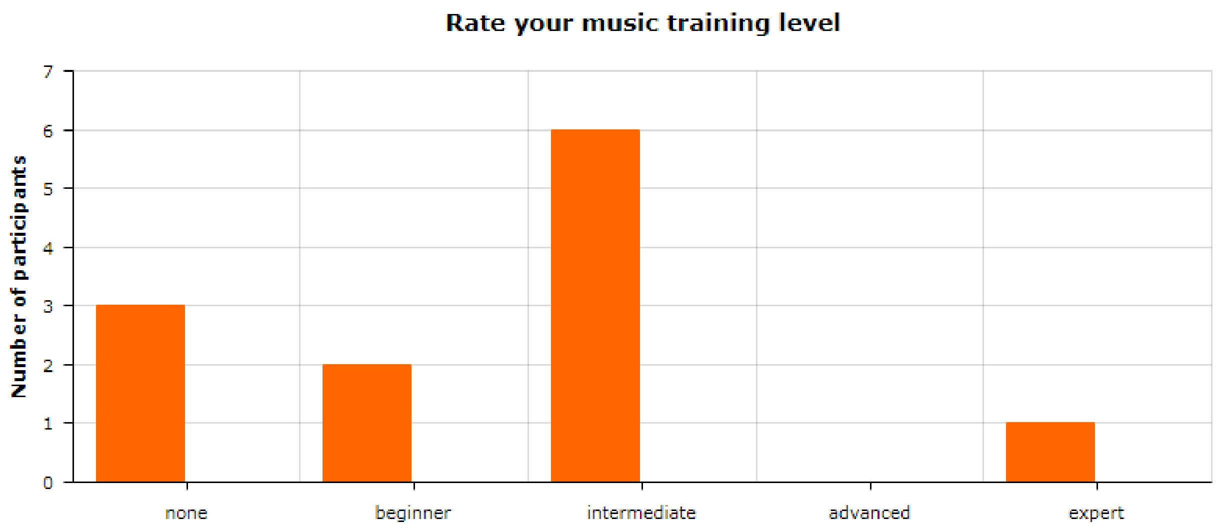
☐ true ☐ false ☐ not sure

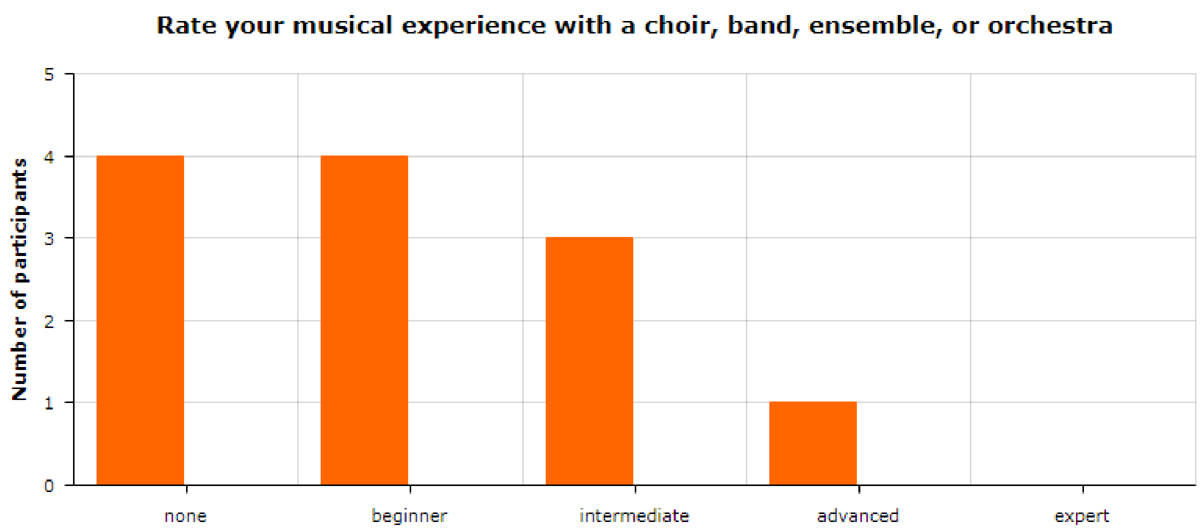
7. Please write below any thoughts, feedback, or suggestions about MUSE you might have.

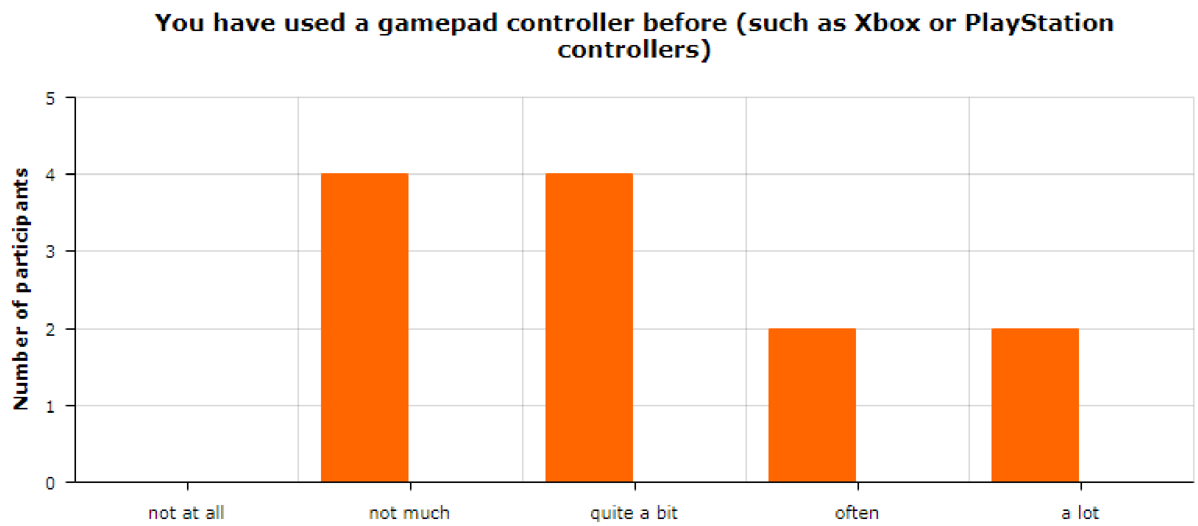
B.2 MUSE User Study Results





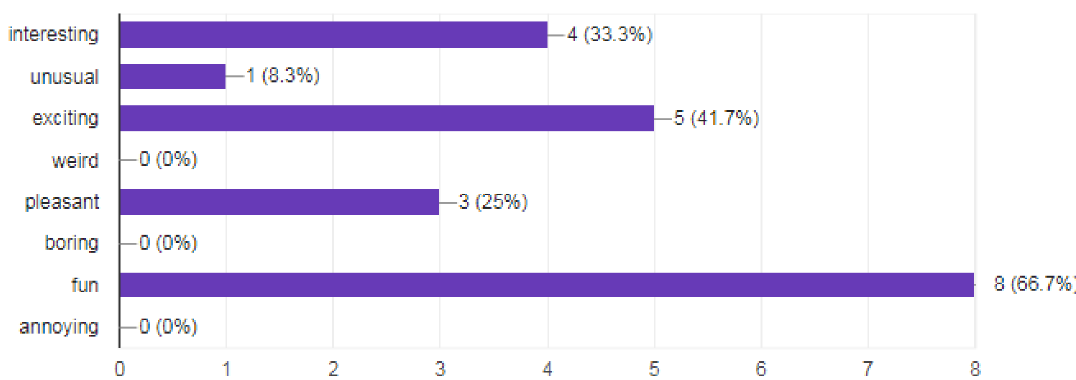

Rate your music training level
Number of participants
0
1
2
3
4
5
6
7
none
beginner
intermediate
advanced
expert

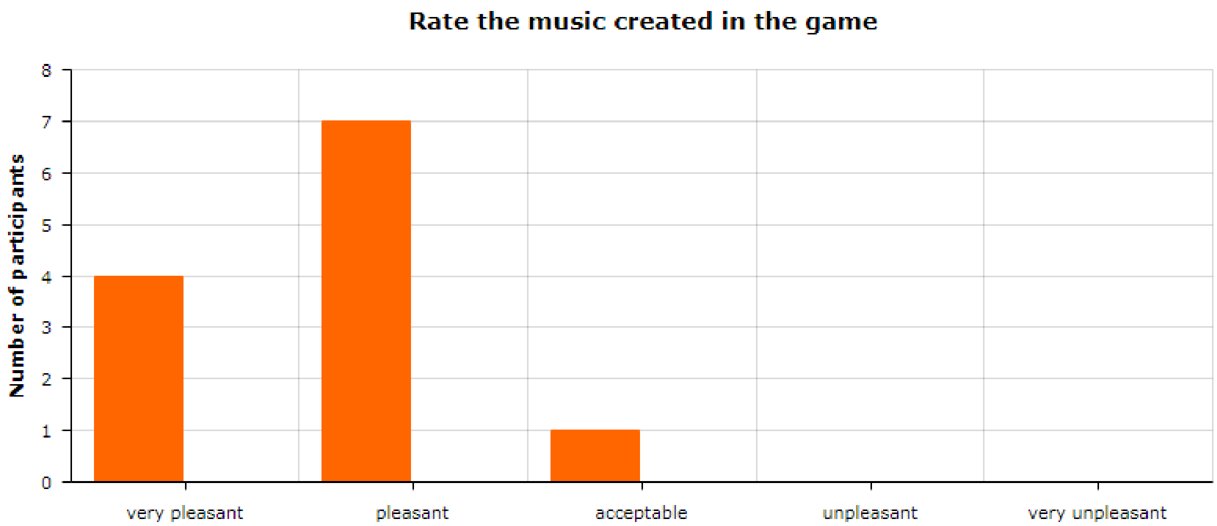

Rate your musical experience with a choir, band, ensemble, or orchestra
Number of participants
0
1
2
3
4
5
none
beginner
intermediate
advanced
expert

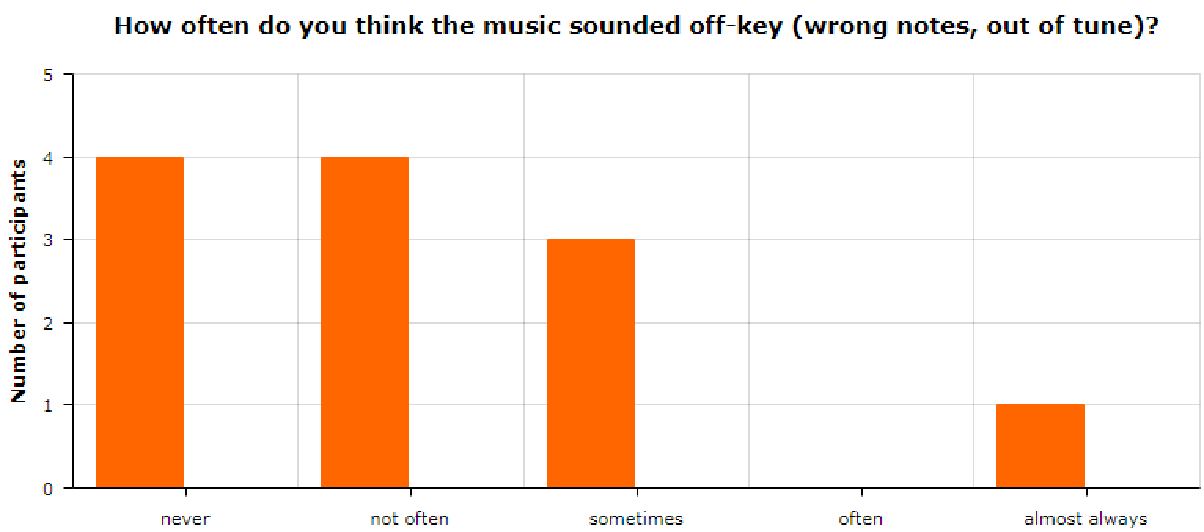


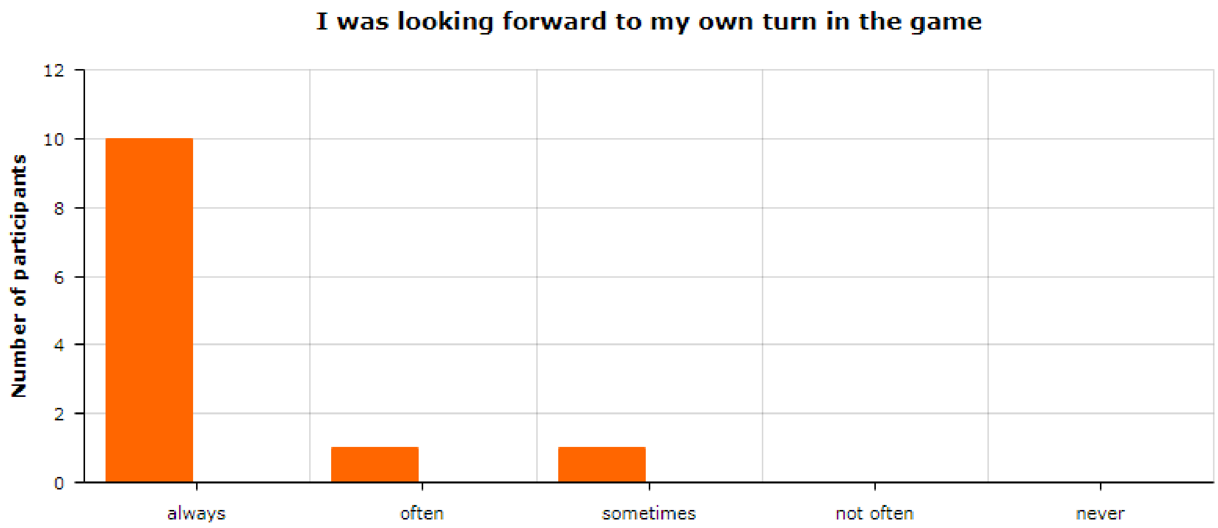
Select what best describes what playing MUSE felt like (or write your own answer)

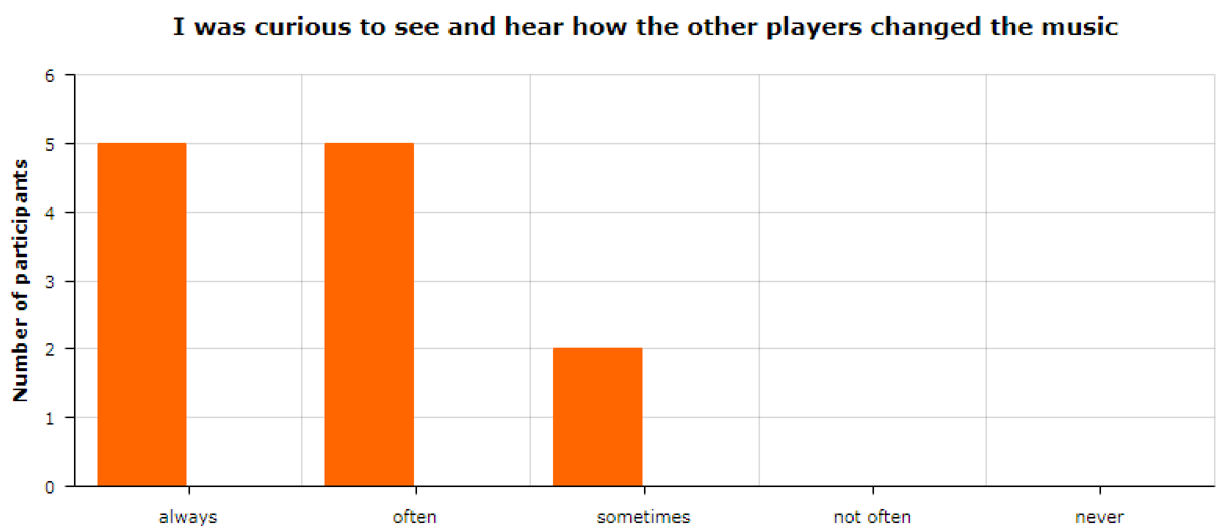
12 responses

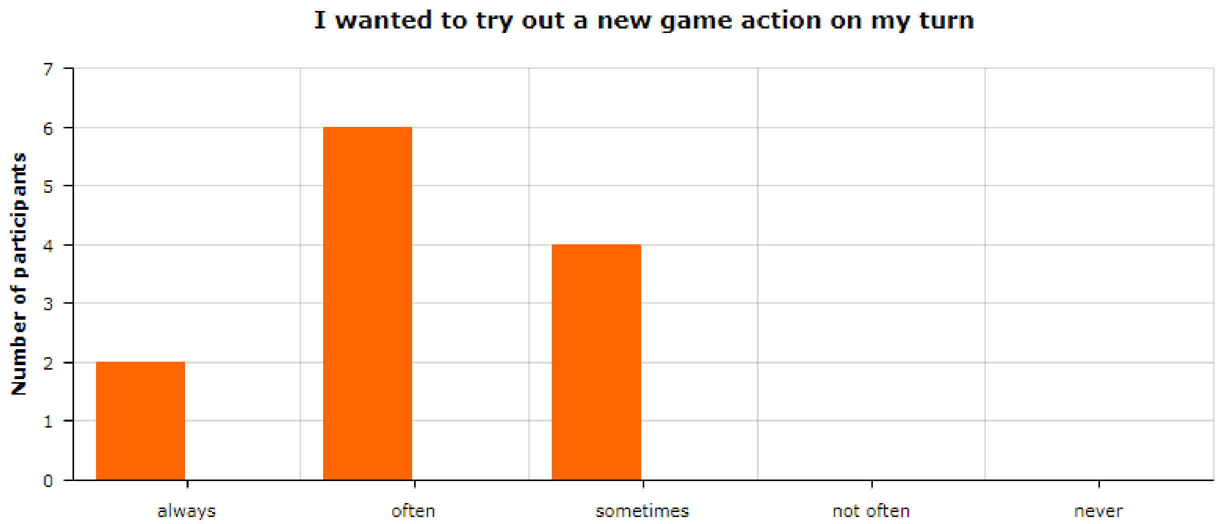


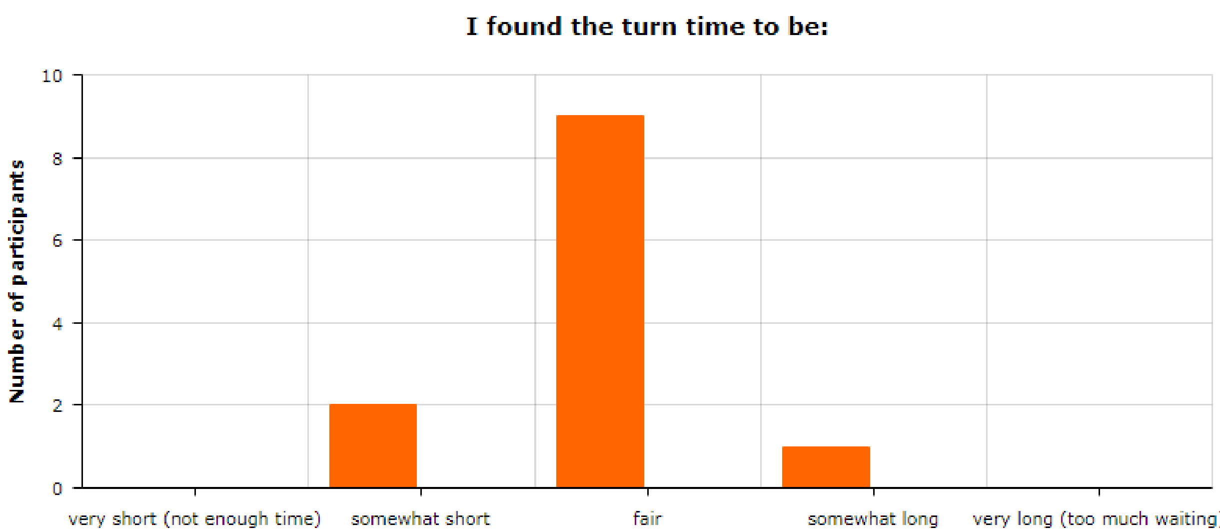

Rate the music created in the game
Number of participants
0
1
2
3
4
5
6
7
8
very pleasant
pleasant
acceptable
unpleasant
very unpleasant

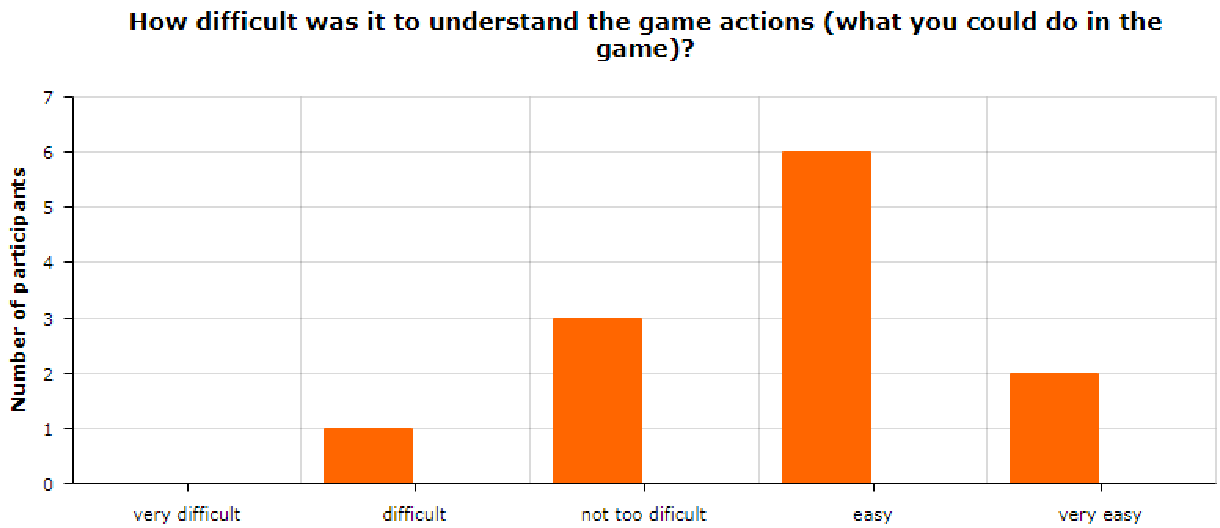

How often do you think the music sounded off-key (wrong notes, out of tune)?
Number of participants
0
1
2
3
4
5
never
not often
sometimes
often
almost always

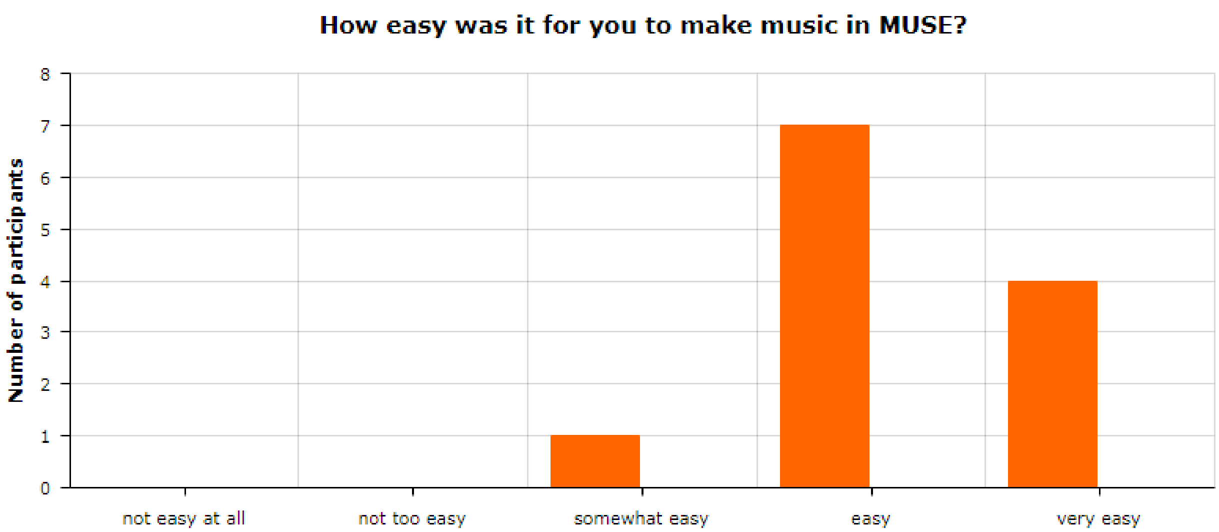

I was looking forward to my own turn in the game
Number of participants
0
2
4
6
8
10
12
always
often
sometimes
not often
never

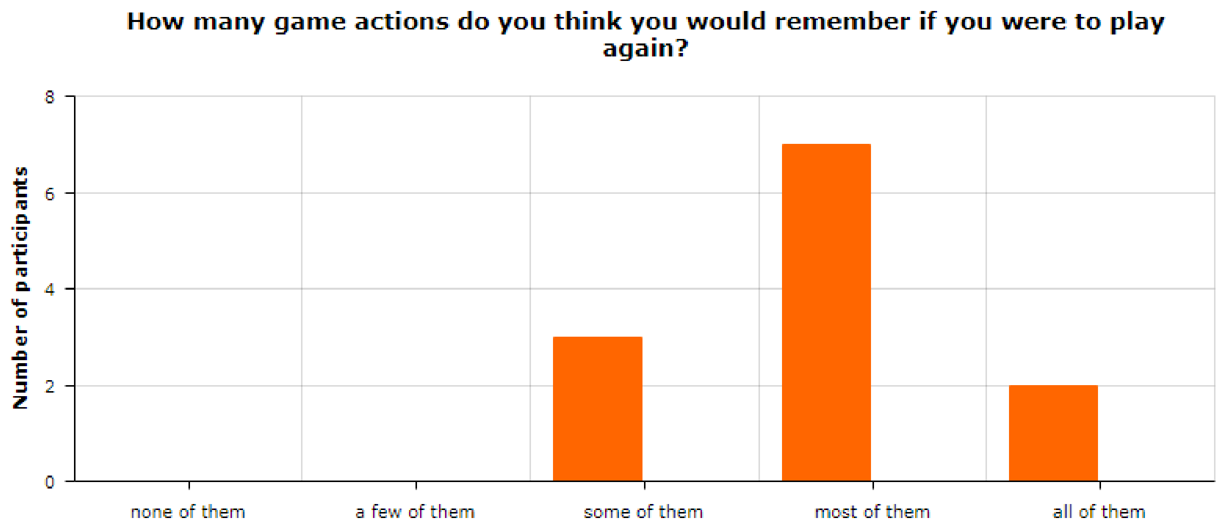

I was curious to see and hear how the other players changed the music
Number of participants
0
1
2
3
4
5
6
always
often
sometimes
not often
never

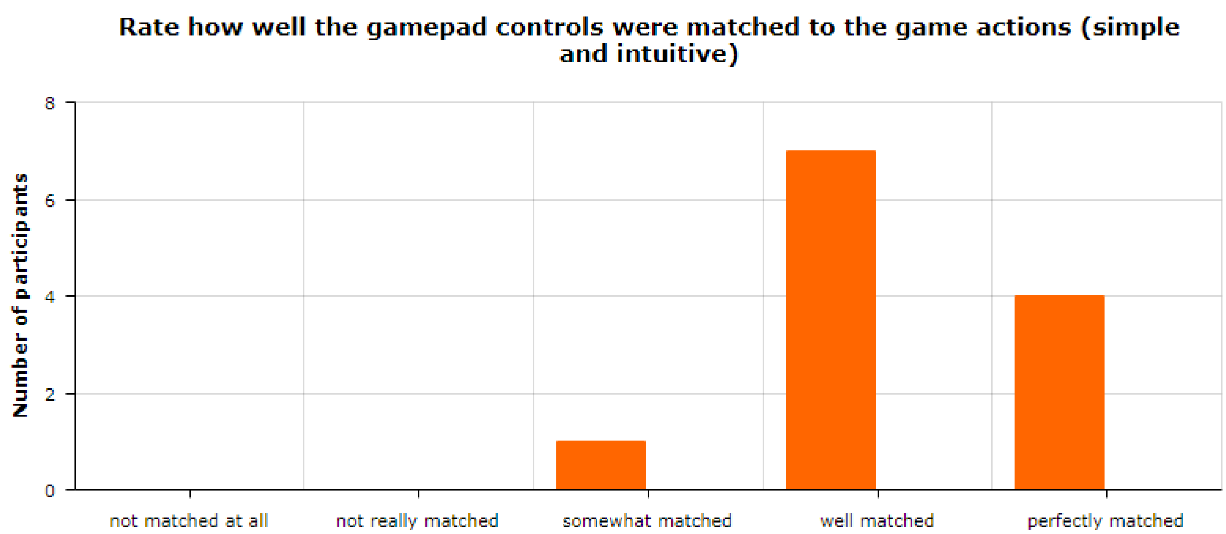

I wanted to try out a new game action on my turn
Number of participants
0
1
2
3
4
5
6
7
always
often
sometimes
not often
never


I found the turn time to be:
Number of participants
0
2
4
6
8
10
very short (not enough time)
somewhat short
fair
somewhat long
very long (too much waiting)


How difficult was it to understand the game actions (what you could do in the game)?
Number of participants
0
1
2
3
4
5
6
7
very difficult
difficult
not too dificult
easy
very easy


How easy was it for you to make music in MUSE?
Number of participants
0
1
2
3
4
5
6
7
8
not easy at all
not too easy
somewhat easy
easy
very easy


How many game actions do you think you would remember if you were to play again?
Number of participants
0
2
4
6
8
none of them
a few of them
some of them
most of them
all of them


Rate how well the gamepad controls were matched to the game actions (simple and intuitive)
Number of participants
0
2
4
6
8
not matched at all
not really matched
somewhat matched
well matched
perfectly matched

Please write in a few words what you enjoyed most in MUSE

12 responses

Everything (2)
The ability to control other blocks
Interacting with players and affecting sounds
Most of the time it sounded good. It was fun!
Easy to play, yet fun.
I enjoyed to try different notes and tune
I liked the sounds in the game and it's so fun
I enjoyed how much variety there was in types of sounds, rhythms and music
Hearing the music off-tune
Playing with friends and seeing how they were going to change the music
I liked the controls and how easy they were to move / change

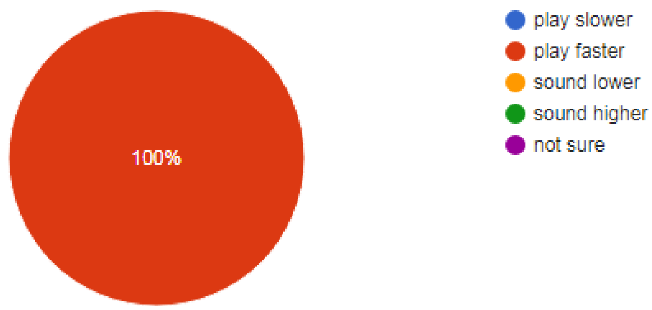
Please write in a few words what you really disliked in MUSE

12 responses

Nothing (4)
Nothing, I loved the game
Getting back from "muted" state was not intuitive for me
Hard / a bit confusing to remember actions
Perhaps more functions, teams?
Nothing that I can think of
I wish we had a little longer for each turn
One person muting everyone
I didn't like it when someone muted everyone

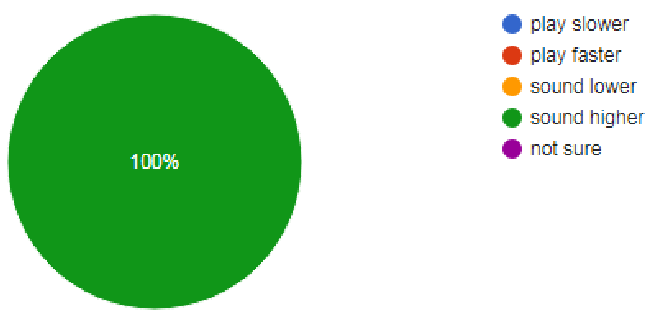
Moving a block (a sound maker) to the right on the board makes it:

12 responses



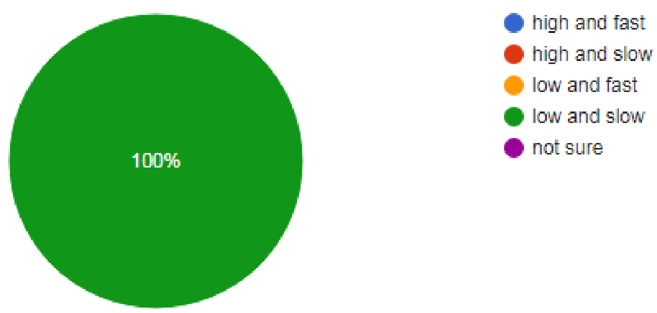
Moving a block up on the board makes it:

12 responses



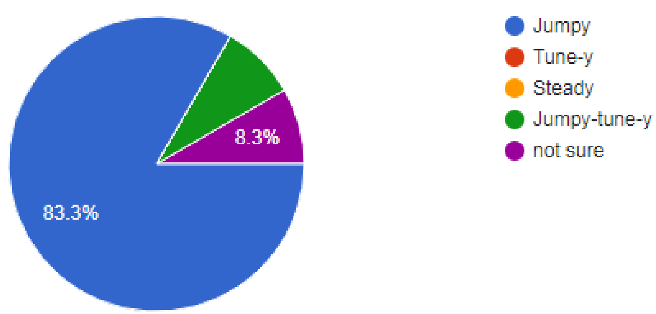
A block (a sound maker) placed in the bottom-left corner sounds:

12 responses



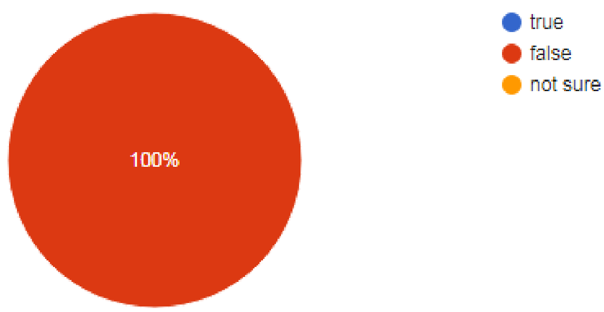
A Triangle shape makes a block sound:

12 responses



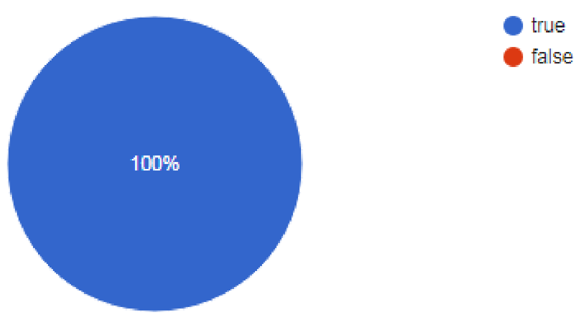
Two blocks can be on the same tile on the board at the same time:

12 responses



You can move all the blocks in one action on your turn

12 responses



Appendix C: MUSE repository

https://bitbucket.org/juliuspopa/muse