NTNU
Norwegian University of
Science and Technology

DJ Game

Exploring New Interactions in Mobile Music
Games

Max Elias Melander

Master of Science in Informatics
Submission date: May 2017
Supervisor: Alf Inge Wang, IDI
Co-supervisor: Sigurd Saue, IM

Norwegian University of Science and Technology
Department of Computer Science

Abstract

This thesis explores the possibility of creating new interactions in mobile music games. It begins with the creation of a research goal and research questions. These questions were aimed at the discovery of how a game implementing these interactions might affect user perception of both music and gameplay elements. By conducting a thorough prestudy, it was found that there might be some new ground to break when it comes to simplification of complex actions on small touch screens. And on the gameplay side of things, there was found to be an obvious overweight in games using rhythm-pattern interactions, where players react to on-screen prompts through simple mechanics to cause some musical output. Few games attempt to flip this music-gameplay relation on its head, having manipulation of music be the central gameplay mechanic. With this prestudy as a backdrop, a game prototype was designed and developed. The interaction design for this prototype was largely inspired by how DJs interact with and manipulate pre-recorded music. Gameplay mechanics were then designed to support these player interactions. By conducting observations and interviews, and analyzing the collected data, the prototype was found to be successful in several areas. Participants showed a better understanding of musical structure and music production and performance, indicating value in simplification when mapping complex real-world interactions to mobile screens. Some of the participants were observed to reach something close to a state of flow while playing the game, confirming the viability of music interaction and manipulation as a core gameplay mechanic. The thesis concludes with a discussion of these findings, and lastly presents potential future work.

Sammendrag

Denne oppgaven utforsker muligheten for å lage nye interaksjonsformer i musikkspill for mobiltelefoner. Oppgaven begynner med å presentere et forskningsmål med tilhørende forskningsspørsmål. Spørsmålene ble laget for å finne ut hvordan et spill som implementerer slike nye interaksjonsformer kan påvirke brukerens opplevelse av både musikalske og spillmessige elementer. Gjennom utførelsen av et grundig forstudium, ble det funnet plass til videre arbeid innen forenkling av komplekse brukerhandlinger for små touch-skjermer. På spillsiden ble det funnet en tydelig overvekt av spill som bruker rytmemønser-interaskjoner, hvor spilleren reagerer på hva som blir vist på skjermen med enkle handlinger og dette skaper et musikalsk resultat. Få spill setter dette musikk-spill-forholdet på hodet ved å bruke manipulasjon av musikk som sin sentrale spillmekanikk. Med dette forstudiet som grunnmur, ble en spillprototype designet og utviklet. Interaksjonsdesignet for prototypen var i stor grad inspirert av hvordan en DJ interagerer med og manipulerer ferdig innspilt musikk. Spillmekanikk ble så designet for å disse interaksjonsformene. Gjennom å utføre observasjoner og intervjuer, og gjennom analyse av innsamlet data, ble prototypen vist å være vellykket på flere områder. Forskningsdeltagerne viste en bedre forståelse av låtstruktur, musikkproduksjon og musikkutøvelse, noe som indikerer en verdi i forenkling av komplekse handlinger til mobilskjermer. Det ble observert at noen av deltakerne nådde noe lignende en flow-tilstand når de spilte spillet. Noe som bekrefter potensialet ved å bruke musikkinteraksjon som en sentral spillmekanikk. Denne oppgaven konkluderer med en diskusjon rundt disse funnene, og presenterer til slutt potensielt videre arbeid.

Acknowledgements

I would like to thank my supervisors, Professor Alf Inge Wang and Associate Professor Sigurd Saue at the Norwegian University of Science and Technology (NTNU), for their support and guidance throughout this project.

I would also like to thank Blank and all of its employees for supporting me with both guidance, office space, and food, during the last half of this school year.

Lastly, I would like to thank all of the participants of the testing phase of this project. Without them there would be no study to present.

Table of Contents

List of Figures

Part I
Introduction

1 Motivation

Internet access from mobile is one of the fastest growing technology platforms in the developing world. This includes the development of low-cost smartphones that are distinct from the high-end devices that are better known. Projects like One Laptop per Child are at risk of being outpaced by market developments - some pointing out that the world is rapidly moving towards “One mobile per person.” [1] [2]. These trends point out the potential broader relevance of the project presented in this report. [3]

With broad take up of advanced mobile technology outside the developed world, creative forms of musical engagement may find resonance in non-western musical cultures [4]. If information systems are accessed primarily through mobile devices, then research on music interaction on small form factors will prove to be highly pertinent [3].

In addition to this, the author has a personal motivation for doing this project. Having experience with both music performance, and music production, the author sees real value in letting non-musicians experience even a taste of what performing music feels like. The author was also approach some time ago by a friend, who is a well-known musician and producer, to create a mobile game. This was seen as an opportunity to try to create something original and new in the genre of music games, and became the motivational backbone for this entire research project.

2 Research Methodology and Research Questions

This chapter will present what research methodology was used to formulate a research goal, and from this goal derive relevant research questions. Under each research question is a short description of how it will be answered.

2.1 Methodology

In such an exploratory project as this, a structured and concrete measurement mechanism for feedback and evaluation was required. To achieve this, the Goal Question Metric Paradigm (GQM) was chosen as a general guideline [5].

The GQM approach is based upon the assumption that for and organization to measure in a purposeful way, it must first specify the goals for itself and its projects, then it must trace those goals to the data that are intended to define those goals operationally, and finally provide a framework for interpreting the data with respect to the stated goals. Basically, a project needs goals to define what informational needs that project has, thereby quantifying these needs for information, and making it easier to analyze the results as to weather or not the goals were achieved [5].

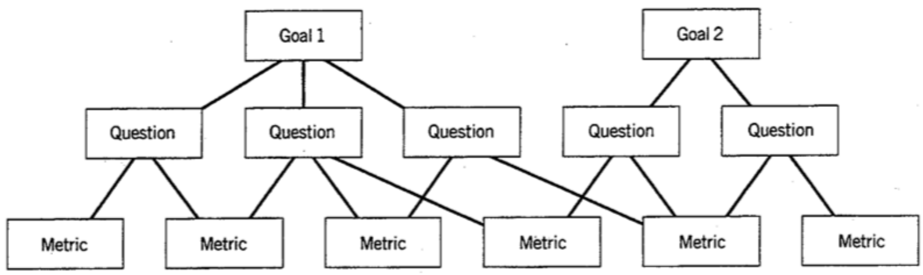

Figure 2-1: GQM Approach

In GQM the measurement is defined in a top-down fashion as shown in Figure 2-1, starting with the Conceptual Level (GOAL). The process of setting a goal is done by considering the goal's three coordinates, issue, object, viewpoint, and purpose. The creation of goals using this process, allows us to derive meaningful research questions that characterize that goal in a quantifiable way. The derived questions should at least encompass these three groups of questions:

1. How can we characterize the object with respect to the overall goal of the specific QGM model?
2. How can we characterize the attributes of the object that are relevant with respect to the issue of the specific GQM model?
3. How do we evaluate the characteristics of the object that are relevant with respect to the issue of the specific GQM model?

After the questions have been developed, they are associated with appropriate metrics [5].

The GQM paradigm was originally created to be well suited for defining quality and productivity improvement goals within an organization, with very process specific questions and metrics. However, this approach is also very useful in more exploratory research projects. It helps with the organization of project goals, research questions and metrics, which is useful when analyzing the results, to better understand if the questions have been answered and goals have been reached, even if these results are more qualitative in nature. Therefore, as this was an exploratory project, the metrics presented below are based on qualitative data collection methods, such as literature study, observation, and one-on-one interviews, rather than qualitative methods such as questionnaires [6] [5].

2.2 Research Questions

Research Goal: Explore the possibility of creating new interactions in mobile music games, and how a game implementing these interactions might affect user perception of both music and gameplay elements.

Using the GQM paradigm described above, this research goal was turned into the following research questions:

RQ1: What is the current state of the art in mobile music interaction?

This research question will be answered by conducting a thorough prestudy.

RQ2: How is the player's understanding and appreciation of musical structure affected by the game?

Participants will be presented with some relevant material both before and after playing a game prototype. Through the analysis of interview data, this research question will be answered.

RQ3: How is the player's understanding and appreciation of music production and performance techniques affected by the game?

Participants will be presented with some relevant material both before and after playing a game prototype. Through the analysis of interview data, this research question will be answered.

RQ4: How can interaction with pre-recorded music make the player feel like they are taking part in the musical performance?

This research question will be answered through analyzing the results of observations and interviews.

RQ5: How does the implemented music interaction affect the player's enjoyment of the game?

This research question will also be answered through analyzing the results of observations and interviews.

3 Research Process

This chapter will present all the steps of the research project, describe how they were conducted, and potential problems with each step.

3.1 Prestudy

In order to answer the research questions above, there was first conducted a prestudy. As the project was exploratory and open-ended, the prestudy phase focused on gathering previous studies, works, and solutions in the field of music interaction, with a focus in games on the mobile platform. This information was then used to create structure to the problem space, making it easier to reach a conclusion on where the next part of the study should focus its efforts. It also informed which technologies where to be used in developing and testing different solutions.

3.2 Prototype Development

In order to understand how different kinds of music interaction in mobile games affects players, there was a need for actual applications to test. Two different prototypes were designed and developed based on the information gathered in the prestudy. The early stages of each development process loosely followed the steps laid out in the book Sprint: How to Solve Big Problems and

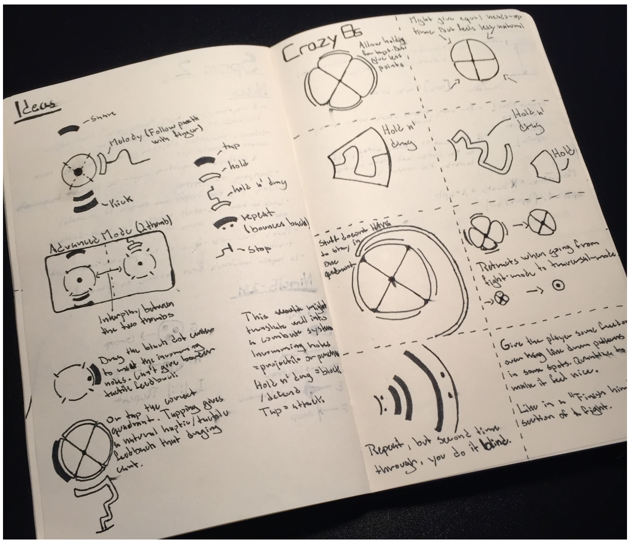

Figure 3-1: Design Ideas

Test New Ideas in Just Five Days by Jake Knapp [7]. The book is mostly written to be used in bigger teams, but even when ignoring the parts meant for teams, it was a big help in getting ideas out fast. The choice to make a prototype instead of a finished product was made based on the time available and the fact that the author of this report was the only developer.

A focus group test of an early version of the prototype was conducted to get rid of any obvious design problems before moving on to the data collection phase of the project. This test was conducted at the author's workplace with 3 of the author's colleagues.

3.3 Data Collection

Qualitative research methods were used to collect data on the usage of the prototypes. Observations as well as semi structure interviews were conducted. Two or more data collection methods create what is known as triangulation. Having data from different viewpoint creates potential for better analysis, giving the presented results more validity.

3.3.1 Observation

To gain a better understanding of how player's actually use and perceive the prototypes, which might be different from what they report when questioned, overt participant observation were conducted.

In overt participant observation the test subjects know that they are being observed, as the researcher is in the room with them, and to some degree takes part in the situation under study. In this case, the participants were allowed to ask the researcher any questions they might have about the use of the prototypes. This was done to create a more casual atmosphere, and quickly get passed any technical hang-ups, as this was not meant to be a usability test [8].

An advantage of conducting overt observations is that the test subjects can give their consent, making the observations more ethical. But there are also disadvantages of doing overt instead of covert observations. The researcher may intrude upon the social setting and potentially interfere with the research subject's normal behavior when not being observed. This is called the

Hawthorn Effect [9]. They also have to get used to being observed, and how to treat the researcher [8]. This can cause stress, and make them uncomfortable or defensive. The more casual atmosphere created by participatory observation was done to hopefully alleviate some of these disadvantages.

Observation was conducted of people playing the game in the test subject's homes, to create an as natural setting as possible. To be able to observe how music affects gameplay and vice versa, different versions of the same prototype was given to the test subject with different feature turned on and off. Each major prototype was given 10 minutes of observation. Making the whole observation process of one subject 30 minutes long, including 10 minutes for setup.

During the observations, the observer took down as many and as detailed notes as possible. This included both things the observer observed, as well as his thoughts on the research process as well as emerging analysis. If felt necessary by the observer, notes were also taken on their role in the process. For example if they felt that they affected the situation in any meaningful way [8].

Since there was only one researcher doing observations, some questions on the validity of the observation data might come into question. Every person has selective recall, selective perception, and accentuated perception [8]. In short, this means that some things are more important to people than others. One observer might perceive and remember certain things as much more important than another observer would have. Because of this validity problem, the observer tried to be reflective under the observations, and note anything that came to mind about them affecting the situation, what they were taking for granted, and what assumptions they were making. As mentioned above, triangulation was used to try to confirm the findings derived from observation by other data collection methods. The observations themselves also used data triangulation as more than one person was observed.

3.3.2 Semi-structured Interview

After the observations were completed, the subjects were interviewed. The interview method used was a semi-structured interview. In this kind of interview there is an incomplete script, which causes a need for improvisation [8]. In this case, it provided what was believed to be the best possible setting to gain as much information as possible, as the goal was to "discover" new information, not to "check" already preconceived notions. This type of interview has the advantage of letting the subject explain him or herself to clarify ambiguities, and go in-depth on personal accounts and feelings. There was no time limit set for the interview, in the hope of removing some of the pressure off of the interviewee of having to create an opinion inside some deadline.

However, this kind of qualitative interview also presents many potential pitfalls. With regard to the problem of artificiality of the interview and lack of trust, our interview subjects were not complete strangers, but acquaintances. This also helped mitigate ambiguity of language, as there already existed communication experience between the subjects and the interviewer. However, one might argue that this presented partiality or familiarity bias, where the interviewee would ”read into” the questions to provide the interviewer with what he or she believed were desired answers. Also, because the subjects were chosen from the researchers social circle, one might argue that there was some elite bias. Meaning that only certain types of people of high status were chosen as test subjects, creating overweight in data from articulate, well-informed informants [10].

In order to ensure that the interviewer was able to focus fully on the task of interviewing the subject, the interview was recorded, removing the need to take notes while asking questions. Before starting the recording, the interviewer asked for consent from the interview subject. After the interview, the recording was used to write a transcription. The researcher that performed the interview verified the transcription by listening to the recording while reading through it. When the researcher was content with the quality of the transcription, it was sent to the interviewee for his/her verification.

3.3.3 Video and Music Presentation

Before the participants were allowed to play the prototype, they where asked to listen to a piece of music, and watch a video of a DJ performance. Then some interview questions were asked to the participants, found in chapters 14.3 and 14.4. The piece of music and video was played for the participants once again, after the play session was completed, and the questions previously asked about them were revisited.

3.4 Data Analysis

The methods described above produce qualitative data. Qualitative data is descriptive data not measurable with numerical results. After completing the interviews, the results were analyzed in the following fashion.

The researcher started by skimming through the interviews to get a sense of the structure, main points, and general ideas. After this, the researcher began the filtering process, thoroughly reading through the material, trying to identify segments of text that were relevant to the research questions, while simultaneously removing segments bearing no relation to the overall research purpose. After this, the coding phase began.

Once more, the researcher read through the interviews, this time labeling each segment from the filtering step with a descriptive word, describing the theme presented by that unit of data. All the labels were then written on a white-board, and the researcher grouped the labels into higher-level concepts. To start with, the researcher used an inductive approach [Oates 2005] to categorize the labels, trying to observe the data with an open mind, clear of all previous experiences, learning, and prejudice. After this first step, the categories were refined. Merging the ones that were too small. Each concept was then assigned a color, and each previously coded segment was marked with that code's concept color to make readability and further analysis easier.

Lastly, the researcher looked for themes and inter-connections between segments and categories, as well as patterns across multiple interviews.

Part II
Prestudy

4 Mobile Music Interaction

The rise of smartphones quickly gave birth to a new, still emergent research field, called Mobile Music, which focuses on the combination of music and mobile technology [11]. Many new studies in this field are now presented at the yearly NIME (International Conference on New Interfaces for Musical Expression) conference. One paper presented at the NIME conference in 2010, presents four musical interaction patterns, which were used as a backbone in the mobile music interaction part of this prestudy. All of the four proposed interaction patterns address, in different ways, the general problem of "How may humans manipulate music and musical information using everyday mobile devices?" The writers of the paper encourage developers to mix several of these patterns into one product, using the parts one finds valuable [12].

4.1 Natural Interaction/Natural Behavior

This pattern corresponds to musical interaction that imitates real interaction with sound producing objects. Thus, it encompasses all musical gestures that might be regarded as "natural". Striking, scrubbing, shaking, plucking, bowing, blowing, etc. It should also be mentioned that the visual and auditory representation and result is equally important. One should strive for a response to the user input that is as natural and expected as possible [12].

One of the greatest benefits using this design pattern is user familiarity [13]. If the product is similar to something the user is already familiar with or has already learn, the learning curve for using the product is reduced [13]. But user familiarity also has its drawbacks in the case of mobile music as a result of the lack of haptic feedback. As shown in [14], the presence of haptic feedback can improve a player's ability to learn the behavior of a virtual music instrument, but if the product is designed to simulate a real instrument, the haptic feedback must be of high quality if it is to promote transfer of skill from the real to the virtual domain [14]. On a flat touchscreen, this is more often than not, simply not possible to achieve. You cannot accurately simulate the feel of guitar strings, a spinning vinyl record under your fingers, or the placement of the buttons on a saxophone. But in the last few years, haptic feedback on mobile devices has seen some progress. This is presented in the Technologies section of the prestudy.

4.1.1 Existing Solutions

Virtuoso Piano Free 2

There are an endless amount of piano apps for the iPhone. This is one of them. It displays piano keys in the range of one octave (12 notes), with the ability to show higher and lower ranges through the six available octaves. It gives the player the ability to slide their finger over the keys to play them, which makes it possible to quickly change between two notes. This is known as a trill in music theory. Where the app diverges from a real piano is in the feel of the keys, the range of the keys (without having to scroll), and the use of sustain pedals. Haptically, on the iPhone, you cannot control how hard the keys are struck, and you cannot physically feel where on the keyboard your fingers are without looking on the screen [15].

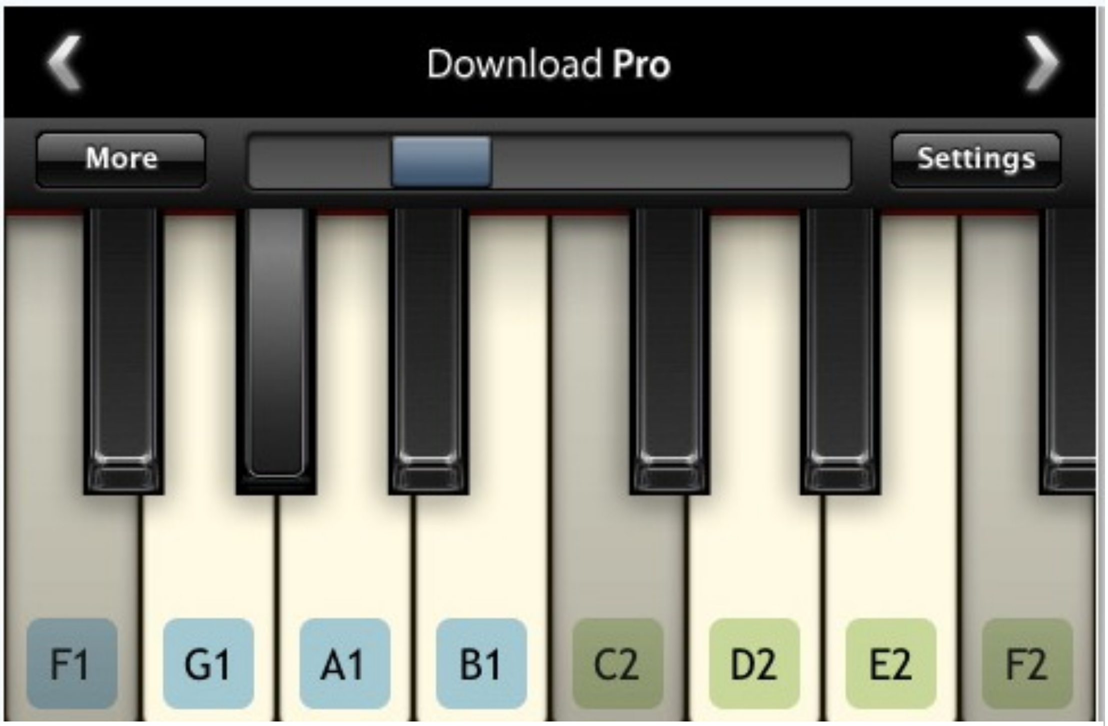

Figure 4-1: Virtuoso Piano Free 2 Screenshot

Djay 2

This app attempts to map a DJ setup to the iPhone and iPad. It has two different views, a classic view showing two turntables, and a modern view showing waveforms. In this section the interesting part is the classic view. This concept of natural interaction with virtual turntables on a touch screen was researched in [16]. The haptic feedback provided by touch surfaces was found not to be good enough for Scratch- DJs, in particular when compared to the sensory feedback of the Traditional/Hybrid setups [16] [17].

Figure 4-2: Djay 2 Screenshot

Drum Meister

As the name suggests, this is a drumming app. The player has the ability to set up their own drum kit, both choosing from different drums and cymbals, and placing these where they want them. Tapping the different drums and cymbals causes them to create a sound. Tapping different locations on the same drum can also create different sounds. Again, mobile touch screens do not register how hard the screen is tapped, although there exists some work in this field (see 8), removing the ability to play with any musical dynamics [18].

4.2 Event Sequencing

The next interaction pattern presented in [12] is event sequencing. This pattern allows the user to access the timeline of the musical piece, and to "schedule" musical events in this timeline, making it possible for them to arrange a whole set of events at once. A design pattern like this can be useful on small mobile screens where real time, precise actions can be difficult to perform. It allows the user to schedule events asynchronously of the sounds playing in real-time, which can be seen as allowing epistemic actions - actions performed to uncover information that is hidden or hard to compute mentally as a complement to pragmatic actions on the system [12] [19].

4.2.1 Existing Solutions

There are many apps that incorporate event sequencing in their design. Most notably the many Digital Audio Workstation (DAW) apps available today.

iMaschine 2

Created by Native Instruments, iMaschine 2 is a fully-fledged DAW for the iPhone. It allows for event sequencing of single drum hits and instrument notes, as well as sequencing of loops and entire sections of a song [20]. There are many DAWs for mobile phones, including Garageband and Samsung Soundcamp.

LP-5

LP-5 is a pure sequencer and mixer, which allow the user to import audio files from multiple different sources, including recording from the hardware input and other apps. It then lets you sequence and mix these audio files in a grid [21].

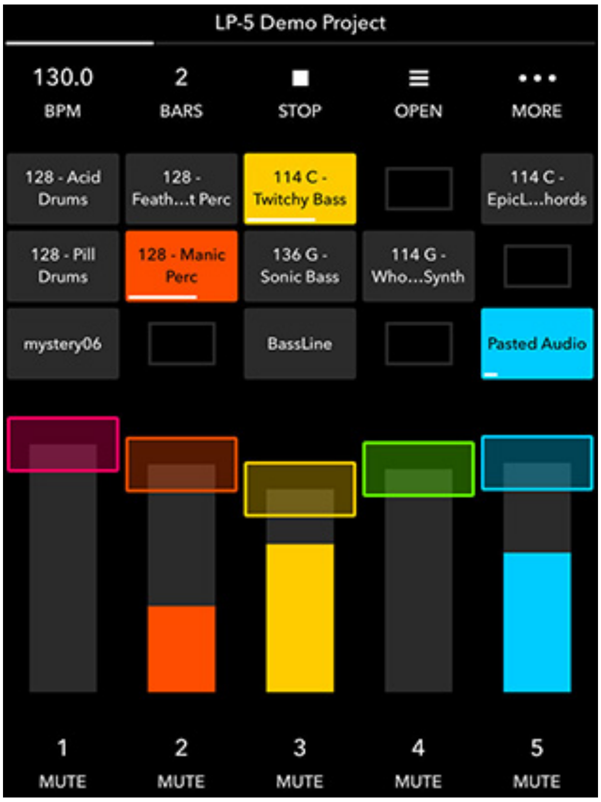

Figure 4-3: LP-5 Screenshot

4.3 Process Control

This pattern aims to free the user from real-time and event-by-event music manipulation by letting them control a process that generates musical events or musical material [12]. Rather than controlling a small sett of limited interaction features that directly causes musical events to happen, the user controls a set of parameters in a musical generation process. The musical content is created by generative algorithms automatically, and the user indirectly controls the music by manipulating the input parameters.

One of the greatest benefits of using this interaction pattern is having the ability to create complex musical results through simple interactions. However, one might argue that the loss of creative control over what the resulting musical elements are, takes away some of the feeling of mastery and motivation [22] as well as a feeling of flow [23] [24] from the player. Concepts explored more in chapter 7.

4.3.1 Existing Solutions

Bloom

Bloom is an iPhone app created by ambient musician Brian Eno, and software designer Peter Chilvers. The player can tap the screen where and how many times they like. These taps are used as input parameters for the music generation algorithm behind the scenes, which then outputs an endless stream of music. Tapping is the only input, aside from shaking the screen to clear it from past inputs. The user is also allowed to change one other parameter, called the mood setting. The moods have very non-descriptive names, like Neroli, Benzoin, and Tolu. In addition to having the user control the input parameters, one can also choose to have the app generate music without any input from the player [25].

NodeBeat

Obviously inspired by Bloom, NodeBeat is a mobile application that lets the user place a number of different nodes on the screen and connect them to each other. The app then uses these nodes as inputs for an underlying generative algorithm that outputs music. The user can also tweak this algorithm by changing rhythm, tempo, and key/scale. The nodes that can be placed come in two varieties. Generators and Notes. Generators can either generate rhythmic or melodic content. The Note nodes are connected to the Generator nodes and play in sequence based on the distance from the Generator node it is connected to [26].

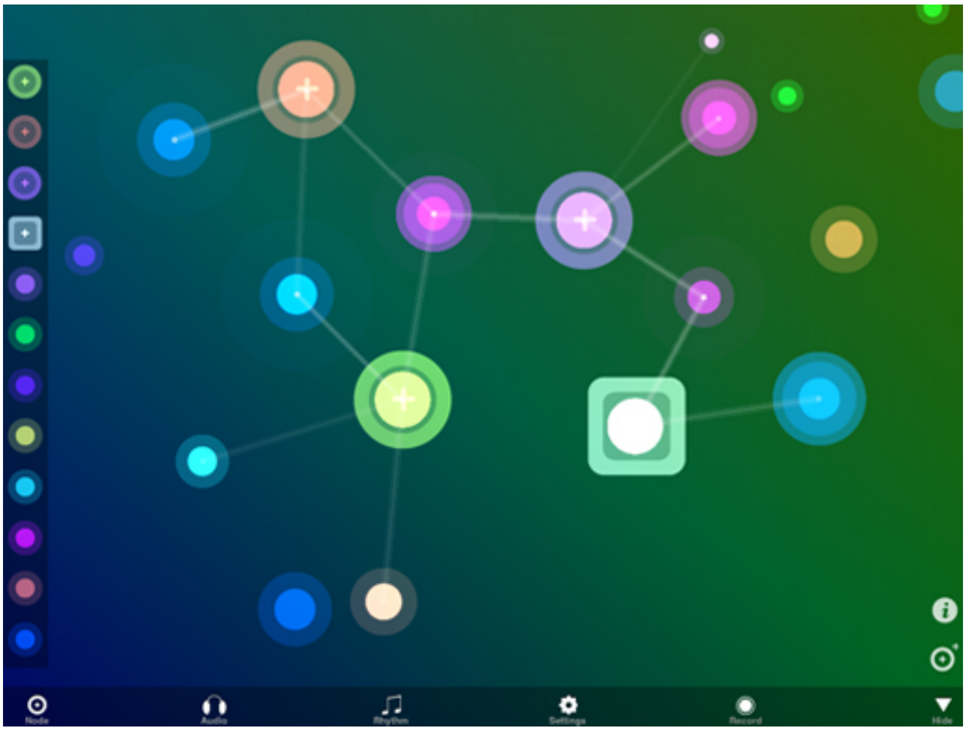

Figure 4-4: NodeBeat Screenshot

4.4 Sound Mixing

This pattern consists in selecting and triggering multiple sounds, so that they may play simultaneously [12]. If two tracks are triggered at the same time their sounds mix and play together, hence the name of the pattern. This can be viewed as a real-time version of the event-sequencing pattern. Musical elements or structures of any duration are triggered in real-time.

As with Process Control, this pattern aims to avoid the note-by-note paradigm of musical control, which is very difficult to implement on mobile devices. Each musical input from the user has the potential to trigger a complex result. The focus of the user will be in combining layers of sound, not necessarily composing anything from scratch [12].

4.4.1 Existing Solutions

Here we can refer back to the existing solutions under Event Sequencing. Both of these examples also allow for real-time triggering of musical structures. iMaschine [20] is the mobile version of a famous sampler and DAW. Each pad of the sampler can be loaded with a sound of any duration, and can be triggered both in real time and in a sequencing mode. Meaning that Native Instruments implemented both the event sequencing, and sound-mixing pattern very closely tied.

Another example of this is Apple's Garageband app [27], which some elements from all of the patterns presented here. Mixing by real-time triggering and modeling of an actual sound mixing board, event sequencing both inside separate instrument tracks and bigger song structures, process control in the form of arpeggiators, which generate musical notes based on simple user parameter control, and lastly natural interaction in the form of modeling many different kinds of instruments, like piano and guitar.

5 Music Interaction in Games

As well as looking at design patterns for music interactions on mobile devices, we need to look at how music can be used in video games specifically. There has been some research done on classifying types of player-music interaction in video games. Pichlmair and Kayali propose seven criteria for analyzing or categorizing the music game genre [28]. These criteria are: active score, rhythm action, quantization, synesthesia, play as performance, free-form play, and sound agents. In the paper, these criteria are only applied to games of the "music game" genre [28]. Also it does not categorize its findings, it simply shows what music games implement what criteria. McAlphine, et al presents a more general view of video game music [29]. From its use in different settings, to its ability to evoke emotion in the player. The paper does not, however, look at the player as someone who can affect the games musical flow.

In this section, we will look at the seven types of player-music interaction presented by Alex Wroten in his master thesis [30], and try to create a connection to the design patterns presented in the previous chapter. Lastly, we will summaries what has been presented so far, and conclude in what areas further research might provide valuable or interesting results.

5.1 Filtered-Preferential Interaction

Wroten describes this type of video game music interaction as one where the player has explicit control over the musical content. Meaning that the player can personalize his/her game experience by choosing its music and controlling when and how the music is presented [30].

This interaction type also includes games that let the player control volume levels of the music independent of the game's sound effects. As such one might view this as an implementation of the sound mixing design pattern. Letting the player play different musical content of their choosing on top of already existing sound [30].

5.1.1 Existing Solutions

Many game on the Microsoft Xbox and Sony PlayStation allow for custom soundtracks. Letting the player replace the game's own soundtrack with songs stored on their consol. Rockstar's Grand Theft Auto series lets players choose from different in-game radio stations when driving vehicles. Audiosurf and Vib-Ribbon generate levels based on music the player provides. These two last examples also use the rhythm-pattern interaction described in 5.3.

5.2 Cinematic-narrative and Cinematic-situational Interaction

In Wroten's thesis, these two types describe cinematic music in games. In cinematic-narrative, the music reflects where the player is in the games narrative progression. It is pre-determined, and the player cannot affect it in any other way than progressing through the game. Cinematic-situational takes on a more dynamic approach, reacting to the player’s gameplay choices. But Wroten warns against using the terms "dynamic" and "non-linear" [31]. Arguing that these terms are too restrictive [30].

As these two types of interaction are not really music interaction per say, but musical content chosen by a program to enhance player action, it is difficult to tie it to any of the design pattern presented above, and as such will not be a point of focus in this thesis.

5.2.1 Existing Solutions

Cinematic-narrative interaction has been very popular in story-driven video games for a very long time. Examples of this are games in the Zelda franchise and Uncharted franchise. But as video game genres blend more and more, typical action games now often introduce player choice to its story line, increasing the need for cinematic-situational interaction to reflect the player's choice musically. Examples of this can be found in Dishonered, The Walking Dead, and Deus Ex.

5.3 Rhythm-Pattern Interaction

If you ask someone to imagine a rhythm or music game, most likely they will think of a game using this pattern. Players are required to react to on-screen triggers in musical time to prevent discontinuities in the game's soundtrack. Often the players are scored based on how "on-beat" they are and are given score bonuses based on un-broken "streaks" where no mistakes are made [30].

Rhythm-pattern interaction is in many ways closely connected to the natural interaction/natural behavior design pattern. Even though these types of games do not always simulate real instruments or dancing, the process of hitting a button in musical time based on on-screen prompts is arguably very similar to playing a pre-written piece of music in a non-improvisational manner. The games not mapping directly from an actual instrument could be argued to more closely follow the sound-mixing pattern. But player inputs in these types of games do not necessarily cause a sound to occur. Rather, correct player input means that the music already playing will continue to play. Which breaks from both of the mentioned design patterns.

5.3.1 Existing Solutions

This style of player interaction has its roots in Bear and Morrisons's classic memory game, Simon, released by Milton Bradley in 1978 [32] [30]. PaRappa The Rapper was among the first games to have rhythm-pattern interaction as part of its gameplay. Later, this interaction pattern was made popular and brought into the mainstream by games such as Dance Dance Revolution, Guitar Hero, and Rock Band.

Figure 5-1: PaRappa The Rapper Screenshot

5.4 Triggered-Incidental Interaction

Wroten describes this type of music interaction as one where player input triggers a discrete musical event. The difference between this and a simple sound effect is its connection to the musical content already playing in the game. These player-triggered musical events are quantized rhythmically and pitched melodically to fit in with the games other musical layers. One can view this as the player composing a part of the musical content in real time, although the music created is mostly a side-effect of the gameplay and not something the player is tasked with creating directly [30].

Again, this type of interaction can be tied to the natural interaction/natural behavior design pattern. Hitting a button on an input device, causing a musical result to occur in real-time, is similar to playing an instrument. But the player does not always control the musical output other than when it should occur, breaking with the control one might expect from a real instrument. Because of this, this interaction type is more closely following the sound mixing design pattern, where the player can trigger a musical elements at any time, but in most cases the game chooses which exact musical element that will be played, taking away most of the musical control from the player.

5.4.1 Existing Solutions

On example of this, brought up in many different papers on video game music including Wroten's thesis, is Toshio Iwai's Otocky. Here the player controls a spaceship in 2D space, and can shoot in any of eight different directions at any time. The projectiles cause a sound-effect that is both quantized rhythmically and pitched melodically to fit in with the background music's harmonies. Another example, which takes on a quite different approach to this, is the fighting game Killer Instinct. In the game, players can perform special finishing moves on their opponents, called Ultra Combos. During an Ultra Combo, the player performs an extended string of attacks on their defeated opponent. Ultra Combos are accompanied by rhythmic beats that sound off each time the character hits their opponent, the melody of which echoes the music theme that was playing during the fight [33].

Figure 5-2: Killer Instinct Screenshot

5.5 Freeform-Representational Interaction

If triggered-incidental interaction is a step lower in the abstraction of musical control than filtered-preferential interaction, then this interaction pattern takes this downward trend in abstraction to its logical conclusion. Here the player has direct control over a game's musical environment. This often causes the game to loose a lot of its "game-like" properties, like rules, goals, and win/loose states [30].

Because of this lack of abstraction, this type of interaction has the potential to follow any of the design patterns presented above. It all depends on what type of musical environment the game presents to the player.

5.5.1 Existing Solutions

A very popular example of this is Iwai's Electroplankton. Different gameplay modes allow the player to control a sequencer-like instrument in different ways. With no real winning or loosing states, this is a very exploratory game. This is very similar to the existing solutions mentioned under the process control mobile music interaction design pattern. Fract OSC uses this type of interaction in conjunction with other. It is a first person puzzle game where solving musically based puzzles and progressing through the game unlocks more and more electronic music instruments and controllers for your home base. These instruments and controllers can be used just like their real life counterparts only through the abstraction of controlling them through an in-game character.

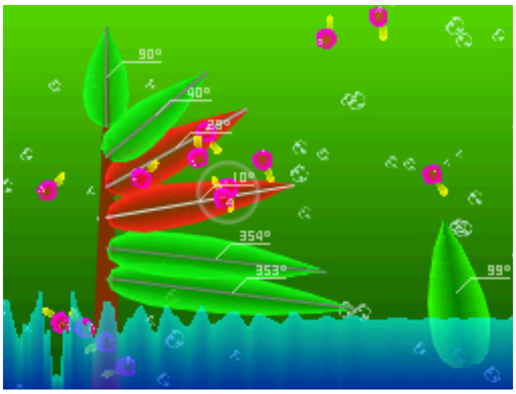

Figure 5-3: Electroplankton Screenshot

5.6 Enqueued-Incidental Interaction

Enqueued-incidental interaction is based on the concept of triggered-incidental interaction. But instead of the user's input causing an immediate auditory result, the player inputs are stored for later use in affecting the game's musical content [30].

This type of interaction can be tied to the event sequencing design pattern. Rather than letting the player have real-time control of musical output, their actions causes musical structures to be sequenced for later playback. One can also tie it to the process control design pattern, looking at player inputs as parameters for a music generator.

5.6.1 Existing Solutions

No existing solutions were found to implement this type of interaction. There might exist solution where player inputs are enqueued in the music generating process without telling the player about it, but such a solution would be very hard to find without the developers having shared some information about it.

6 Summary and Conclusions

The work presented so far shows that a lot of work already has been done both in the terms of music interaction on mobile devices and player-music interaction in video games. But in music interaction design a lot of effort seems to be put into the mapping of real-world musical phenomena to multi-touch screens without big efforts in abstractions. This is shown especially in DJ applications for mobile phones, where both turntables and mixers are mapped directly to virtual representations of the same hardware. Therefore, there might be some new ground to break when it comes to simplification of complex actions on small screens, but still having these actions be performed in real time, instead of as event sequencing or as control of parameters in generative algorithms.

On the game side, there is an obvious overweight in games using rhythm-pattern interactions. It is a form of interaction where one can easily add game rules and scoring, but it does not give the player any real control of music manipulation or any real choice in what to do. It boils down to simple reaction. Also, when including triggered-incidental interaction, one can see a trend forming. Most music games, aside from those using freeform-representational interaction, uses music as a side-effect output of what the player is doing in the game. Gameplay mechanics that would have worked without the music, be it pressing a button when a note aligns with the strum bar in Guitar Hero, or shooting an enemy in Otocky, causes sound to happen as a side-effect. There seems to be a lack of games that flip this gameplay-music relationship on its head, where control of music is the gameplay mechanic in and of itself, and manipulation of music is what causes gameplay side-effects. Games where the player is given a set of tools to manipulate music, and is required to use these tools to progress through the game.

7 Game Design Theory

Making a game solely based on previous works in music interaction on mobile devices and player-music interaction concepts in games do not necessarily mean you'll end up with a successful and engaging product. What makes a game fun and motivating in and of themselves is a popular research topic. Even though making a complete game was outside the scope of this project, some previous works in creating engaging gaming experiences was used as a backdrop when creating the prototypes presented in the next part of this thesis.

This section will present different concepts in game design and the evaluation of what makes a game fun to play. There is no summary at the end of this section, as these terms and their potential value is discussed throughout the presentation of prototype designs.

7.1 Flow and GameFlow

Flow is an experience "so gratifying that people are willing to do it for its own sake, with little concern for what they will get out of it, even when it is difficult or dangerous" [24]. To achieve flow, the experience must consist of these eight elements:

1. A task that can be completed
2. The ability to concentrate on the task
3. That concentration is possible because the task has clear goals
4. That concentration is possible because the task provides immediate feedback
5. The ability to exercise a sense of control over actions
6. A deep but effortless involvement that removes awareness of the frustrations of everyday life
7. Concern for self disappears, but sense of self emerges stronger afterward
8. The sense of the duration of time is altered

The combination of these elements causes a sense of deep enjoyment so rewarding that people feel that expending a great deal of energy is worthwhile simply to be able to feel it [24]. Additionally, an important precursor to a flow experience is a match between the person's skills and the challenges associated with the task, with both being over a certain level.

In their paper, Penelope Sweetser and Peta Wyeth present a model for designing, evaluating, and understanding player enjoyment in games [23]. They call this model GameFlow. It maps the eight elements of flow to eight gameplay elements, which each includes a set a set of criteria for achieving enjoyment in games. These eight elements and their related criteria can be seen in Table 7-1.

Element	Criteria
Concentration Games should require concentration and the player should be able to concentrate on the game	- Games should provide a lot of stimuli from different sources - Games must provide stimuli that are worth attending to - Games should quickly grab the players' attention and maintain their focus throughout the game - Players shouldn't be burdened with tasks that don't feel important - Games should have a high workload, while still being appropriate for the players' perceptual, cognitive, and memory limits - Players should not be distracted from tasks that they want or need to concentrate on
Challenge Games should be sufficiently challenging and match the player's skill level	- Challenges in games must match the players' skill levels - Games should provide different levels of challenge for different players - The level of challenge should increase as the player progresses through the game and increases their skill level - Games should provide new challenges at an appropriate pace
Player Skills	- Players should be able to start playing the

Games must support player skill development and mastery	game without reading the manual - Learning the game should not be boring, but be part of the fun - Games should include online help so players don't need to exit the game - Players should be taught to play the game through tutorials or initial levels that feel like playing the game - Games should increase the players' skills at an appropriate pace as they progress through the game - Players should be rewarded appropriately for their effort and skill development - Game interfaces and mechanics should be easy to learn and use
Control Players should feel a sense of control over their actions in the game	- Players should feel a sense of control over their characters or units and their movements and interactions in the game world - Players should feel a sense of control over the game interface and input devices - Players should feel a sense of control over the game shell (starting, stopping, saving, etc.) - Players should not be able to make errors that are detrimental to the game and should be supported in recovering from errors - Players should feel a sense of control and impact onto the game world (like their actions matter and they are shaping the game world) - Players should feel a sense of control over the actions that they take and the strategies that they use and that they are free to play the game the way that they want (not simply discovering actions and strategies planned by the game developers)
Clear Goals Games should	- Overriding goals should be clear and presented early

provide the player with clear goals at appropriate times	- Intermediate goals should be clear and presented at appropriate times
Feedback Players must receive appropriate feedback at appropriate times	- Players should receive feedback on progress toward their goals - Players should receive immediate feedback on their actions - Players should always know their status or score
Immersion Players should experience deep but effortless involvement in the game	- Players should become less aware of their surroundings - Players should become less self-aware and less worried about everyday life or self - Players should experience an altered sense of time - Players should feel emotionally involved in the game - Players should feel viscerally involved in the game
Social Interaction Games should support and create opportunities for social interaction	- Games should support competition and cooperation between players - Games should support social interaction between players (chat, etc.) - Games should support social communities inside and outside the game

Table 7-1: GameFlow elements

7.2 Challenge, Fantasy and Curiosity

Similarly to Sweetser and Wyeth, Thomas W. Malone presents a set of heuristics or guidelines for designers of video games in his paper What Makes Things Fun to Learn? He organizes this into three categories: challenge, fantasy and curiosity [22].

For a game to be challenging, it must provide a goal whose attainment is uncertain. According to Malone, the best goals are practical or fantasy goals (like reaching the moon in a rocket), rather than simply goals of using a skill (like doing arithmetic problems). The players must also be able to tell whether they are getting closer to the goal. Malone proposes four ways of making the outcome of a game uncertain for players: Variable difficulty levels, multiple level goals, hidden information, and randomness [22].

Fantasies often make computer games more interesting. Malone differentiates between intrinsic and extrinsic fantasies. Most extrinsic fantasies depend only on whether or not the skill is used correctly. Did the player answer the math questions right enough times for the man not to get hanged? In intrinsic fantasies however, the skill also depends on the fantasy. The player gets to se an actual graphical representation of their skill in use, meaning that the problems are presented in terms of the elements of the fantasy world. In a tennis game, the player is required to use tennis specific skills (mapped to a controller). If the player misses, he/she can see by how much and in what direction. Malone argues that intrinsic fantasies in general are both more interesting and more instructional than extrinsic fantasies. When the fantasy in a game is intimately related to the material being learned, the players are able to exploit analogies between their existing knowledge about the fantasy world and the unfamiliar things they are learning [22].

Curiosity in the player/learner can be achieved by providing environments that have an optimal level of informational complexity [34] [35]. Optimal complexity is achieved when the player know enough to have expectations about what will happen, but where those expectations are sometimes unmet. Malone presents two types of curiosity: Sensory and cognitive curiosity. Sensory curiosity involves the attention attracting value of changes or patterns in the sensory stimuli of an environment. Cognitive curiosity however, comes from the desire to bring better "form" to one's knowledge structures. The designer can achieve this by presenting just enough information to make the player's existing knowledge seem incomplete, inconsistent, or inparsimonious [22].

7.3 Motivation

Denis and Jouvelot reinforce many of the concepts and ideas presented above in their paper on motivation in educational games. Following is their description of four best practices, which promote optimal motivation in the player [36].

1. Reify values into rules. Game designers must translate the values the game should express into rules. You don't have a game if you don't have any rules.
2. Give power. Players must be provided expressive ways to confront with and test rules, experiencing meaningful feedback to their input.
3. Tune usability. Entry barriers that go against the players' urge to practice the game should be leveled.
4. Derail the gameplay. Designers should provide gamers with alternatives ad space instead of constraining them in a predefined trajectory that hinder audacity, creativity and exploration - key aspects of fun, and learning.

8 Technology

This chapter will present different technologies, and explain why some were chosen over others. Relevant technological advancements are then presented. These are technologies that might be used in future work.

8.1 Game Engine, Platform and Frameworks

When choosing what technology to use for development, there were many pros and cons to take into consideration. Because the author was working on this project alone, and already had experience developing specifically for iOS using Apple's Swift programming language, it was chosen to develop specifically for the iOS platform. This choice meant that working prototypes could be developed very quickly. Something that was of great importance in such an exploratory project. One negative aspect of the choice of going platform specific, especially on the iOS platform, is in market share. According to the IDC, Android had 86.8% of the smartphone market share in unit shipments in the third quarter of 2016. iOS only had 12.5% market share [37]. Releasing a product like the one presented here, only on iOS, means missing out on most of the potential market. This in turn would necessitate porting the app to the Android platform after it was finished, leading to a lot of extra work.

There are many game engines that support cross-platform development. Some of the most popular ones being, Unity Mobile, Unreal, and Cocos2D-x. Using one of these would have made porting the visuals and game logic extremely easy, but it would have necessitated being fluent in the C++ programing language, which the author was not at the start of this project. The same goes for the use of an audio framework. I knew I didn't want to learn either Core Audio by Apple or OpenGL ES for Android, because this would take up a lot of time in learning something already made easy by many different frameworks, and in most cases, I would only need simple audio playback. Again, I knew the Swift programming language well, so I chose the most popular audio framework for iOS, AudioKit.

If I were to go forward with the last presented prototype, making it into a complete product, I would invest the time in learning C++, and moving development over to using the Cocos2d-x game framework, together with the Superpowered cross-platform audio framework. But again, given the need to be able to make working prototypes quickly, I chose to develop on the iOS platform, using the Swift programming language together with SpriteKit and AudioKit.

8.1.1 SpriteKit

Apple's SpriteKit framework helps with the creation of 2D sprite-based games. It claims to make it easy to create high performance, battery-efficient games. It supports custom OpenGL ES shaders and lighting and advanced physics effects and animations.

SpriteKit is a graphics rendering and animation infrastructure that you can use to animate arbitrary textured images, otherwise known as sprites. SpriteKit provides a traditional rendering loop that alternates between determining the contents of and rendering frames. You determine the contents of the frame and how those contents change. SpriteKit does the work to render that frame efficiently using graphics hardware. SpriteKit is optimized for applying arbitrary animations or changes to your content. This design makes SpriteKit more suitable for games and apps that require flexibility in how animations are handled.

Having had experience with creating applications from scratch with both plain OpenGL and with the help of SpriteKit, the latter was chosen to help quicken the process form idea to working prototype.

8.1.2 AudioKit

AudioKit is an audio synthesis, processing, and analysis framework for iOS, macOS, and tvOS. It is built upon the AVFoundation framework created by Apple, and aims to significantly simplify audio programming on iOS devices. The key concept for this framework is that everything is built up from nodes. Nodes are interconnectable signal processing components. Each node has at least an output, and most likely parameters. If it processes another signal, the node will also have an input. This means that the developer is free to use and connect these audio-processing components in any way they please.

I did not have any experience with audio programming or signal processing before starting this project. A framework like this made it possible for me to explore in prototyping, without having to worry about low-level signal processing.

8.2 Relevant Technology Advancements

As mentioned before, a lot of design issues stem from the fact that mobile devices cannot, in many cases, give good physical feedback. Recently, advancements have been made in haptic feedback technologies for phones. Starting from the iPhone 6s, Apple included their Taptic Feedback Engine in phones. Using a linear actuator, the Taptic Engine can reproduce the sensation of motion or generate new and distinct tactile experiences, often reinforced by both visual and auditory feedback. This technology is accessed via Apple's UIFeedbackGenerator class. The prototypes of this project was created for and tested on an iPhone 6. As a result, there was no access to this new technology.

In the case of recognizing tap strength on mobile screens, some work has been done to use the mobile devices accelerometer measure this. Anthony Picciano created a subclass of Apple's UIGestureRecognizer, which attempts to do this. On a tap, the accelerometer is accessed and a variable named pressure is set to a value between 0.0 and 2.0 [38].

Part III
Design and Development

9 Design

This prototype was designed to try to tackle two of the least explored researched topics found in the prestudy phase of this project. Namely, simplifying complex actions without taking away too much creative control from the player, and using manipulation of music as a game mechanic in and of itself.

Loosely following the steps laid out in Sprint [7], many different solution sketches were made. With the first problem I tried to address being how to let the player control prerecorded music. I did not want to use the Guitar Hero solution of correct player actions simply turning the music on or off. After many weeks of failed ideas, a potential solution came to mind. Controlling pre-recorded music is exactly what DJs do. They create live remixes of songs by jumping between two different songs and mixing different parts from different songs together. The main motivation behind this prototype then became how to simplify and map DJ actions to a mobile screen, and use these actions as the main gameplay mechanics of a game. Meaning that the game should not dictate exactly what button the player should press at what time. Rather, the player should have full access to all of these mechanics, and be able to use them at any time to solve whatever gameplay challenges are presented to them.

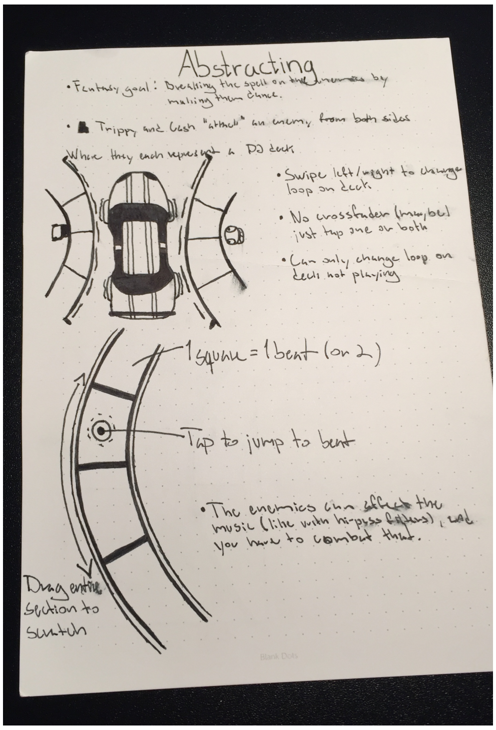

Figure 9-1: Final Design Sketch

This chapter first presents how DJ specific actions were mapped to a mobile screen, and then goes through the gameplay rules designed to allow the use of this mapping as game mechanics.

9.1 DJ Controls Design

This section will present the prototype's mapping of DJ equipment and actions into a simplified mobile representation.

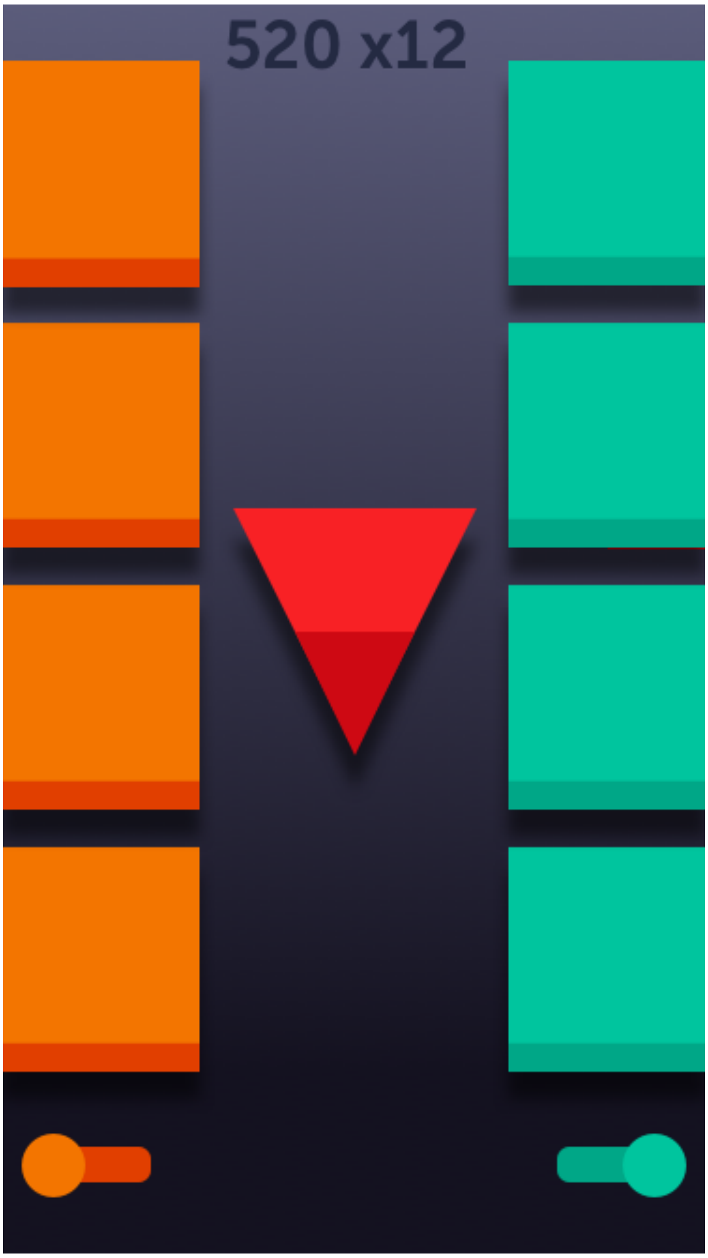

Figure 9-2: Prototype Main View Screenshot

9.1.1 Decks

As with a conventional DJ setup in most cases consists of two decks and one mixer. A deck in DJing refers to an audio player of some sort. Usually a turntable or a Pioneer CDJ. Aside from the mixer, the two decks are what provide the DJ with music manipulation abilities. They control playback speed, what part of the audio that should be played and so forth.

In this solution, each side of the screen is identical and represents one deck each.

9.1.2 Loops

A DJ deck typically lets the DJ play audio files, CDs or vinyl. In this solution, on deck holds four different loops. Each loop is four beats long. By swiping the decks toward the middle of the screen, a visual representation of all available loops on that deck is shown. By swiping the same direction again over a loop the player can change which loop each deck is playing at any time. Each loop is represented by a unique color.

Figure 9-3: Prototype Change Loop Screenshot

9.1.3 Segments

Typical digital DJ setup lets the DJ store queue points in a song. Queue points are markers in a song that can be instantly jumped to, without the need to scrub through the track. Queue points are used in many different situations. Hitting a queue point while scratching with digital vinyl or platters means that the performer never looses track of where they are in the track, something that was not possible with analog DJ setups. Setting up queue points on a kick drum and a snare hits, makes it possible for the performer to play these queue points like a sampler or drum machine. Effectively creating a live instrument on the spot. Queue points can also be used to simply loop a track by continuously jumping backward to a previous point in the song.

This prototype divides each loop up into four equal segments, visually represented by one colored button each. The top button holds the first segment, continuing down to the bottom button, which holds the last segment. By tapping a button, the player can jump to that segment at any time. This then, effectively works like the more traditional queue points, other than that the player cannot place these points manually.

9.1.4 Mixer

A traditional DJ setup has a mixer between the two decks. The does all signal processing on both input channels, including volume control, EQ, and FX. The most interesting feature of a DJ mixer in out case is the crossfader. The crossfader is a volume slider between the two decks. Moving the slider all the way to the right means that you only hear the right deck. Putting the slider in the middle results in you hearing equal amounts of both decks. The crossfader can be used to transition between songs by sliding it slowly from one side to the other, jumping from one song to the other, and so forth. For turntabalism, the crossfader is used in a much more artistic way. On most mixers, you can set the crossfader curve. This option decides how the crossfader's position translates to volume levels. Turntabalism and scratching requires a sharp crossfader curve, allowing the performer to move between full and no volume with very little movement. Volume cutting, in different patterns, done together with turntable manipulation, is what creates the sounds people recognize as scratching.

In this solution, there is no mixer. Volume control between the two decks is simplified. A player can choose to trigger only segments from one deck, causing the other deck to be silent, or the player can trigger segments in both decks at once, causing the two decks to play at equal volume.

9.1.5 Rate control

A traditional deck, be it a turntable, CDJ, or digital controller, usually has some way of controlling the playback rate. This can for instance be a slider that sets the playback speed between 0.25 and 2 times the normal playback speed. It can also be in the form of letting the DJ move the actual physical vinyl or some representation of it, to have total control of the playback, both forwards and backwards.

In this solution, the player can slide their finger along a deck to change its current playback rate. This rate can be set to any value between 0.25 and 2 times the normal playback rate. There was also work done to implement full vinyl-like control over the playback, but the chosen audio framework unfortunately did not allow for this to be done in a satisfying way. As a result, full vinyl like control was scrapped for this prototype. If it had been successfully implemented, the player would have been able to slide their finger along a deck in any direction to cause playback to follow the finger's movements. Using more than one finger while "scratching" like this would cause automatic crossfader movements, like in DeJay [17]. But again, this was not implemented in this prototype, as changing between forward and reverse playback caused to many audio glitches using the selected framework.

9.1.6 Quantization

Quantization in this case, means to snap musical elements to a rhythmic grid, so that the triggering of such an element can never happen off musical time.

Tapping a loop segment in this prototype, queues that segment to be played at the next beat. Effectively quantizing the player input. By tapping the toggle button under each deck, the player can also choose to turn off quantization, resulting in segments being played as soon as they are tapped. This can result in un-rhythmic results, but also gives the player much more creative control.

9.2 Game Design

9.2.1 Fantasy / Story Premise

Even though simplified, the player controls described above are very close to real DJ controls, making the app so far feel more like a tool than a game. To avoid a game with intrinsic fantasy [22], a story premise was constructed to give the controls a natural place in the game world.

The player's avatar is Trippy Turtle. Trippy Turtle is a DJ, and also a turtle. The game starts in Trippy Turtle's hometown on the west coast of Norway. All Trippy wants to do is play music for his friends and go on an adventure. But the evil corporate overlord, D. Bag, has other plans. He has taken away all emotion and love from the world. Making all of the worlds inhabitants grey and dull. Trippy realizes that he has the power to save the world. By making people dance, he can break D. Bag's spell. Trippy goes on the adventure of a lifetime, traveling to New Jersey to stop the evil D. Bag, saving as many people as he can along the way.

This story premise was not presented to the player in the first iteration of the prototype, but was told to the participants of the observations by the observer before they started playing.

9.2.2 Goal

The goal of the game is to save the enemies, discovering how to use the musical abilities at your disposal to make them happy. If the music stops, or the player can't save the enemy in time, the game is over. Subgoals are to save the enemies in as few moves as possible, and to create score streaks by stringing together enemy-relevant actions without using any unnecessary actions in between.

9.2.3 Enemy Design

This prototype includes one enemy type, which can come in many different variations. The enemy is a triangle. It has two sides, each corresponding to the deck on its side. The sides can have different colors, with each color corresponding to a specific loop. The sides can be flipped vertically, and have different y-axis positions relative to each other. Each enemy variation requires some specific player action to be defeated. If multiple actions are required, the player does not have to perform them all at the same time, but each required action must be executed for a full loop length (four beats).

Following are examples of different enemy layouts, with a description of what player actions they require.

Example 1

Play the first loop on the left deck, and the second loop on the right deck.

Figure 9-4: Enemy Layout 1

Example 2

Play the right deck in half tempo relative to the left deck. Both decks playing the first loop.

Figure 9-5: Enemy Layout 2

Example 3

Play the segments on both decks in reverse order, with the left deck playing the third loop, and the right deck playing the second loop.

Figure 9-6: Enemy Layout 3

Example 4

Play the segments on the right deck in reverse order, and the segments on the left deck in normal order, with the left deck playing the first loop and the right deck playing the fourth loop.

Figure 9-7: Enemy Layout 4

Example 5

Play the segments on the left deck in reverse order and in half tempo relative to the right deck. Play the segments on the right deck in normal order. Play the second loop on both decks.

Figure 9-8: Enemy Layout 5

In addition to the static properties described above, the enemy also has the potential to animate, indicating more rhythm-based actions for the player to perform. One or both of the sides can flash in a rhythmic pattern, indicating to the player to unlock the deck's quantization and play the pattern on the segments of their choosing.

9.2.4 Game Progression

The game is split up into four levels, with only the first level being unlocked at the start of the game. Each level includes five different enemy configurations, which must be solved by the player for the next level to unlock. If the player fails at any point during a level, the whole level must be restarted. Each new level comes with its own set of loops. The master tempo for each new level is faster than the previous one.

9.2.5 Rules and Rewards

The game is designed to be played in relatively short sittings. Less than ten minutes a sitting. To promote this, the amount of player actions required to win or loose are kept to a minimum. After the player triggers the first sound, if at any point the music stops for more than one beat, the game is over. If the player does no action suggested by the enemy design for more than four loops (16 beats), the game is over. To defeat or "solve" an enemy, the player must perform each action suggested by its design for the duration of one loop. These actions can be performed at the same time or sequentially in any order.

A score and "streak multiplier" is always visible at the top of the screen. Every enemy motivated action the player successfully completes awards them ten points times the current streak multiplier times how many actions they completed at ones. Every completed action also makes the streak multiplier go up by one. If an entire loop goes by without the player successfully executing an enemy motivated action, the streak multiplier is reset to zero.

After completing a level, the player is presented with a summary screen. This screen first gives a three start rating on every enemy in that level, based on how long it took for the player to defeat that enemy versus the optimal solve time. These star ratings are then used as additional score multiplier for the final score.

9.3 Focus Group

After the first iteration of the prototype was completed, a focus group test was hosted at the offices of Blank with 3 of the author's colleagues. The game was first presented on a projection screen. The participants were given a description of design concepts and what parts were not yet implemented. After this short presentation, the participants played the game.

Some design and implementation problems were revealed during this test. The prototype did not show which segment was currently playing and which segment was queued to be played next. There was no indication of what caused the player to loose the game, and no good feedback on when an enemy was successfully defeated. A software bug was found that caused the game to crash when loop four was chosen in specific situations.

As the author's colleagues are software developers and interaction designers, a bunch of useful design and implementation suggestions were also collected. These suggestions helped development of future iterations of the prototype.

10 Implementation and Software Architecture

Following is a class diagram of the prototype with a description of each class. Shown in Figure 10-1. Some of the Apple specific boilerplate classes are left out.

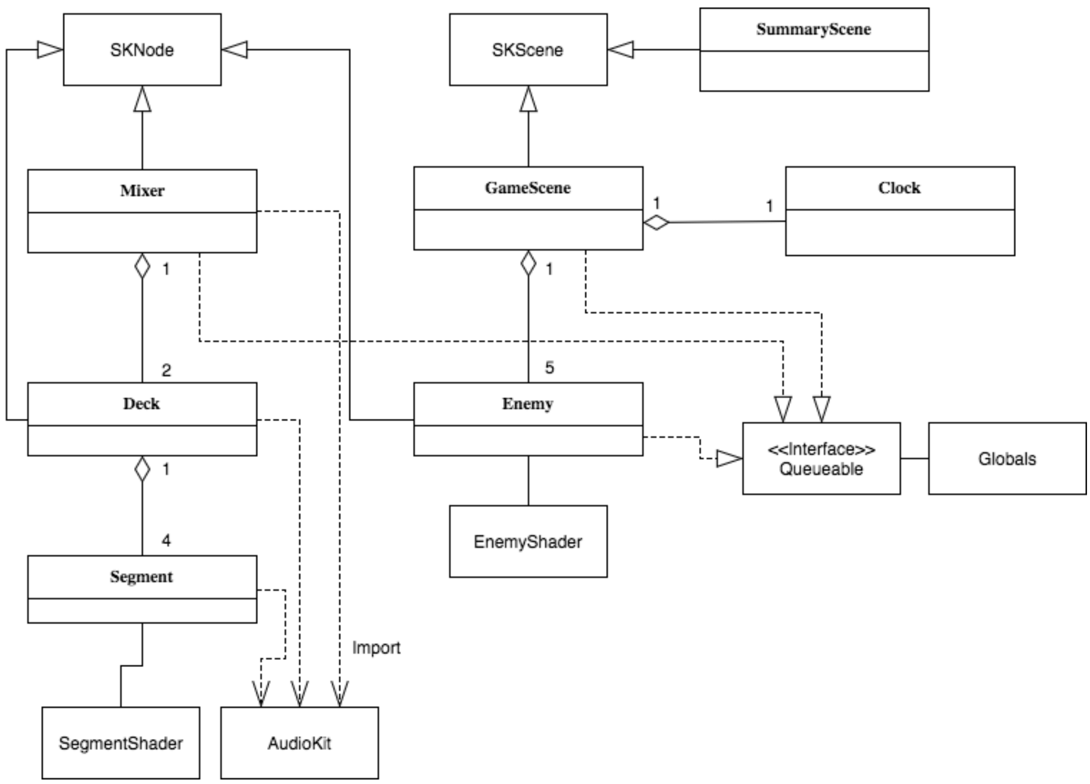

Figure 10-1: Class Diagram

10.1 Clock

This is arguably the most important class. It works like a metronome, calling a callback function in its parent class (in this case GameScene) on every beat of some given tempo in beats per minute. The first iteration of this class used the standard Apple getCurrentTime method and ran in the main update loop of the game. This resulted in an average delay of 40ms from when the callback function should have been called to when it actually was. To solve this, the class was rewritten to run on its own thread, decoupling it from rendering calls and game logic. Also the CACurrentMediaTime method was used to get the current time. This method returns the current absolute time, in seconds, derived by calling mach_absolute_time. mach_absolute_time returns CPU ticks since the device was last rebooted, and proved to be much more accurate. The average error of this class is now around 30ns on an iPhone 6.

10.2 GameScene

SpriteKit renders its content based on a node tree, where the top node always is an SKScene class. The GameScene class inherits from SKScene, and controls all the main game logic, as well as handling all user input. Every class that needs to render something to the screen is a child node of GameScene. This class also holds a callback queue for use with the clock. Every class that wants to do something in rhythmic time has to implement the Queueable protocol. Then the specific object calls the queueCallback function of the GameScene with itself as a parameter. On the next beat, triggered by the clock, every object in the callback queue of the GameScene is called and removed from the queue.

10.3 SummaryScene

This class holds all the code for the score summary view, shown at the end of a successfully completed level. It is transitioned to by the GameScene, and takes the current level, score, multiplier and star ratings for each enemy as initial parameters. After rendering and animating the player's level results, it handles inputs from the player to transition to the next level.

10.4 Mixer

This class is the top level of the group of classes implementing audio processing using the AudioKit framework. It has the responsibility of wiring up the two decks to an AudioKit output, and handling and sending user input to the appropriate functions of its two decks. It controls which loop and segment each deck should play when triggered.

10.5 Deck

This class handles the loading of audio files into its array of loops, the splitting of these loops up into player triggerable segments, the playback of these segments, playback rate control, and volume control. It holds an AKMixer class as an output, and wires every segment it holds into this one output, which in turn is connected to the Mixer class's output. This class also inherits from the SpriteKit SKNode class, and holds an array of SKSpriteNode classes as children. This array of sprite nodes render the segment buttons to the screen. One great advantage of using SpriteKit's node hierarchy system is the ease at which animations can be applied to multiple children at once. The mixer can easily apply position translation with an easing function to one of its decks when the user wants to change loops and have this animation affect all of the decks children with one line of code.

10.6 Segment

This class controls playback of segments. The main reason for not doing this directly in the Deck class comes from problems using the AudioKit framework. AudioKit provides the ability to jump to any point in an audio file during playback, but this feature does not perform as well as needed. Therefor every Segment class stores two identical audio files. Every time the play function of this class is called, which audio file is the active one switch. Thereby avoiding AudioKit having to jump to the beginning of an audio file while it is playing. This class provided a good abstraction for this solution.

10.7 Enemy

This class handles all things enemy. It generates the enemy variation, handles rendering, animations and player input, and keeps track of player scoring. It holds a list of required player actions, and keeps track of which of these has been completed and when they where completed. The result of time passed and the player's actions is then passed to the GameScene, which controls win/loose states and score tallying.

10.8 Shaders

To give the player better visual feedback of the audio and rhythmic timing, special fragment shaders to be used by the segment button and enemy sprite were written. SpriteKit uses OpenGL ES, so these shaders were naturally written in GLSL. These shaders both take the current audio amplitude and beat tick as uniform inputs, and use these values to change the colors of the sprite's fragments.

11 Challenges

The biggest implementation challenge by far came as a result of using the AudioKit framework. It required many hacks, like the one in the Segment class, to get what seemed like simple functionality to work in a satisfying way. Since the game is all about listening to and manipulating audio, any small glitches in the audio playback proved to be extremely noticeable.

Gameplay and interaction design was by far the hardest part of this whole project for me. I spent multiple months just writing down and drawing idea after idea without getting any real results I was happy with. Using the Sprint method [7] to quickly get the ideas on paper and exploring different approaches to that same solution definitely helped, but arriving at the design solution presented above was a painful process for me.

Part IV
Results

12 Test Population

The test population consisted of ten people. To some degree, convenience sampling [8] was used to select the test subjects, because all of the test subjects were easily recruitable acquaintances. Even so, subjects were selected based on their gender, age, musical background, and gaming experience to get the best possible population representation in as little time as possible.

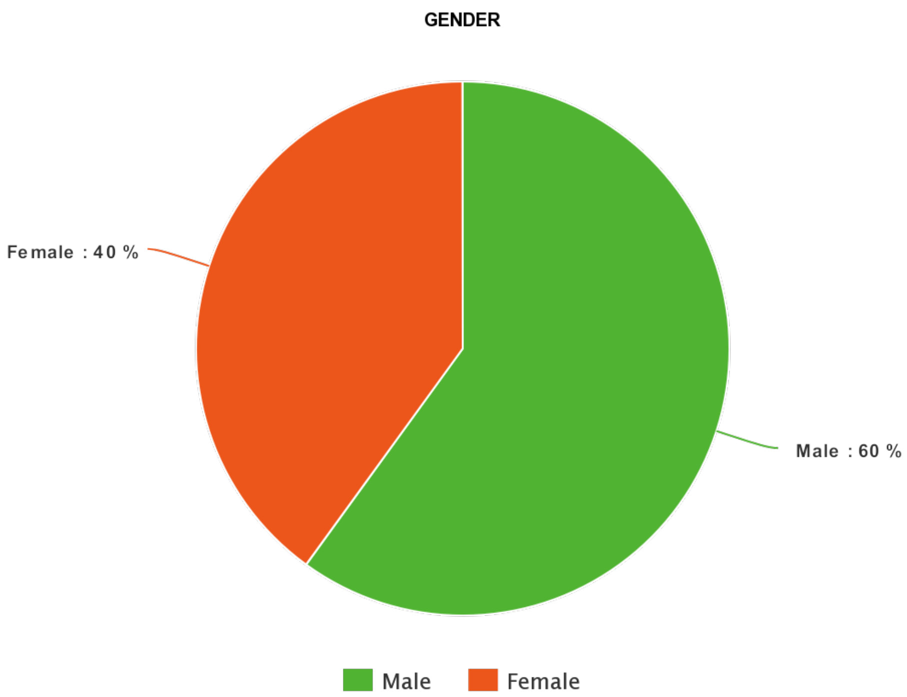

Figure 12-1: Gender Pie Chart

Out of the 10 people in the test population, six were men and four were women.

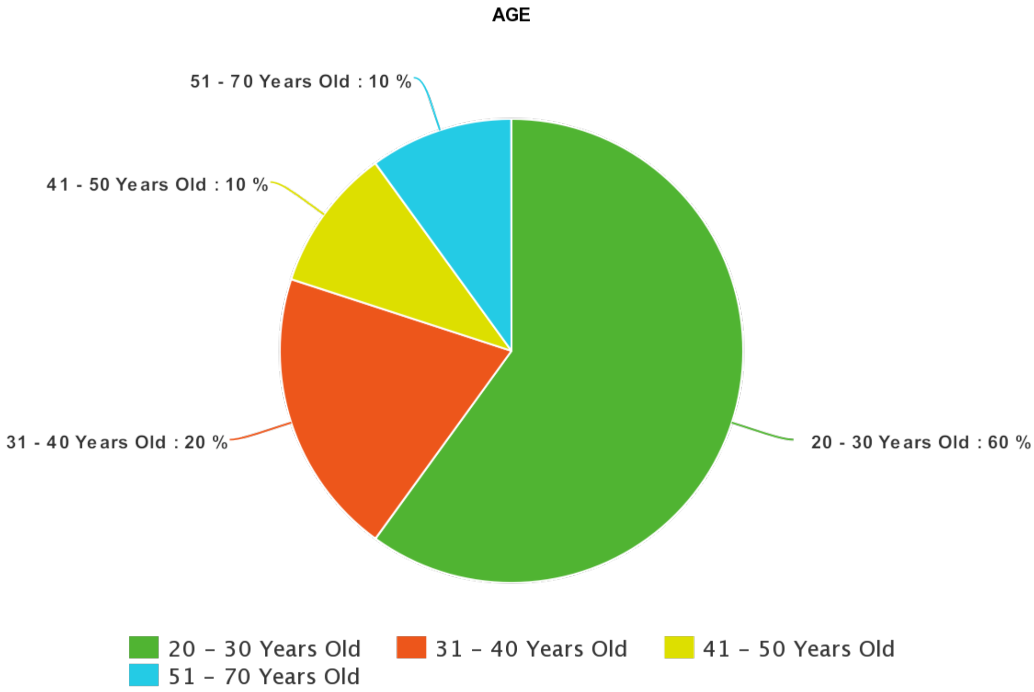

Figure 12-2: Age Pie Chart

The age of the population ranged from 25 to 62 years old.

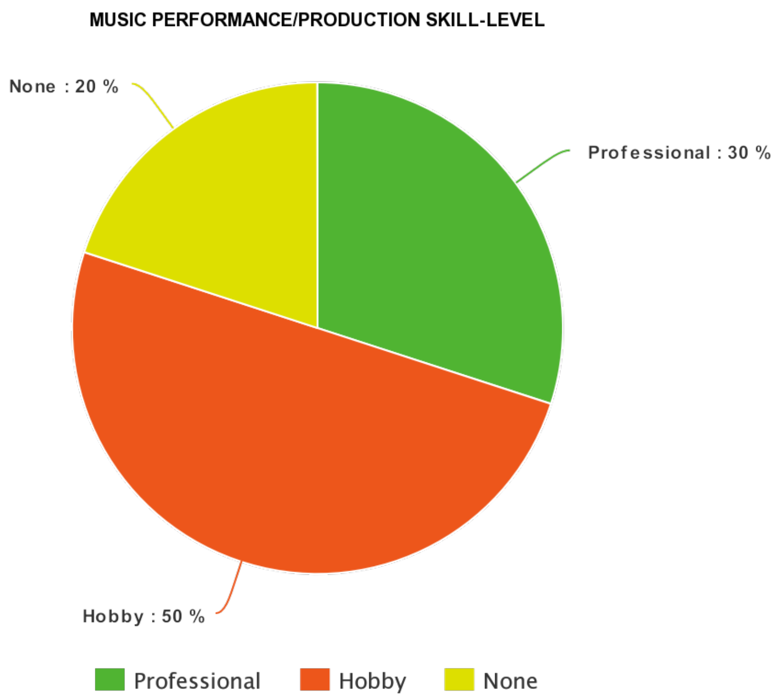

Figure 12-3: Music Skill Pie Chart

Two of the test subjects worked as professional musicians, one of these two also worked as a music producer, one worked as a professional DJ, and the rest had varying degrees of hobby level musical knowledge.

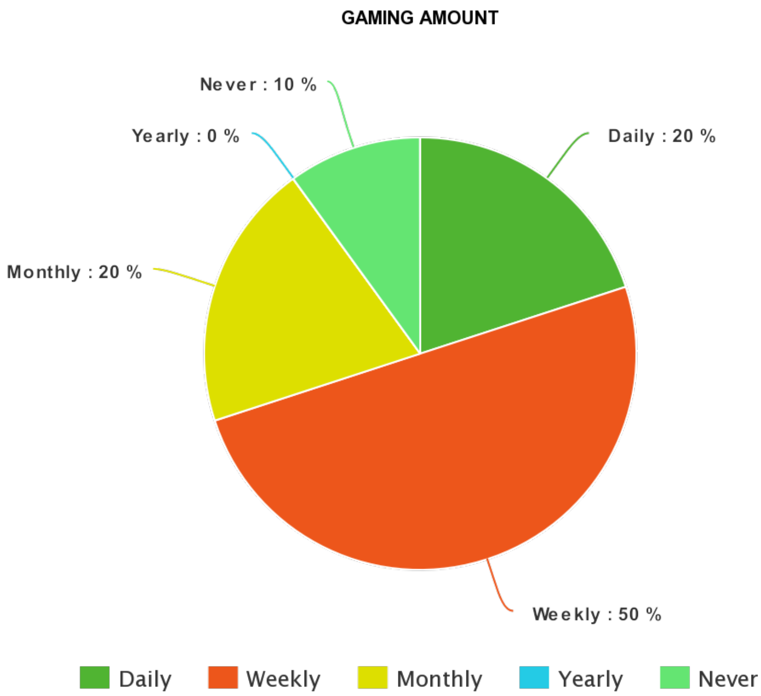

Figure 12-4: Gaming Amount Pie Chart

Every participant except one played video games to some degree, ranging from daily to monthly gaming. This chart does not represent what devices the gaming is happening on. Resulting in someone playing mobile games in 10-minute chunks daily to be grouped with someone playing computer role-playing games for hours every day.

13 Observations

Following is a summary of the observations made. The observer wrote down everything observed, including thoughts concerning his own impact on the subjects and reasons why some observations were made. These thoughts on impact and reasons behind specific behavior in the test subjects are revisited more thoroughly in chapter 15.

The concept of queuing a segment to be played next took time for some of the participants to understand. Non-musicians especially, seemed to expect that hitting a segment button would immediately result in some sound output. Even after the observer telling them that hitting a segment queues it to be played on the next beat, some participants would still wait, and try to hit the button exactly on beat, sometimes causing them to hit it too late and loose the game. The early parts of this learning curve looked to cause some frustration in the test subjects, with five subjects making loud groans after failing a level. After understanding what the enemy wanted from them in terms of player actions, some test subjects needed a second to stop and think about how to actually do the actions required. This pause in some cases was long enough for the music to stop and the subject to loose the game. However, after a couple of tries, all participants started to naturally queue segments early. The visual feedback of showing which segment is currently playing and which is queued to play next seemed to help quicken this learning process, as all the struggling participants asked the observer if their understanding of this visual feedback was correct. Which in all cases, it was.

Non-musicians also took the longest to complete the entire first level, with the oldest participant with no gaming experience never being able to defeat all five enemies of the first level before observation time was up. Non-musicians seemed to have some trouble with observing the enemy and figuring out how to defeat it while simultaneously keeping the music going. The observer might have caused some performance anxiety in some of the participants here, since the observer himself was known by all the participants to have a musical background. However, these struggling participants showed a strong desire to quickly restart the game after failing a level, wanting to do better and get further with every try. Some participants also wanted to retry a level even after having completed it, because they did not get a three star rating on every enemy in that level. Musicians did not show this same difficulty with observing and performing at the same time. This might come from experience with reading sheet music or communicating with other musicians while playing an instrument at the same time.

Several participants expressed how cool they thought the game was. These exclamations were observed to often come straight after they discovered a new feature/gameplay possibility or were able to complete a set of actions resulting in satisfying musical output. One feature that seemed to elicit this type of response most often, was the ability to change the rate of audio playback while the audio was playing, creating a kind of vinyl slow-down or speed-up effect on the audio. After coming to grips with the controls, the test subjects looked to be very engrossed in the game, especially the ones who chose to where headphones. Two test subjects quickly restarted levels multiple times without ever looking up from the phone. In the first two levels of the game, the professional musicians and DJ did not focus fully on the game in the same manner as the non-musicians, making comments and asking questions to the observer while playing.

When playing the game with the audio muted, players found it very hard to keep the music going, loosing track of where they were in the music's progression. The game looked to loose its entire grip on the players, making them loose focus and becoming disinterested.

The input required to change a loop, swiping the deck to the side once, and swiping again in the same direction over a loop to select it, proved to be unnatural for almost all test subjects. All test subjects tried to swipe the other way the second time to hide the loops again, which only caused the loops on the other deck to be revealed.

14 Interviews

Following are the questions used as a guide for the semi-structured interviews. Under each question is a summary of the answers given by the test subjects, presenting both the general consensus as well as original and interesting points.

14.1 Music Background

1. In what settings do you most often listen to music?

Most subjects answered that they listen to music while doing other things, like walking somewhere, taking the buss, cleaning their apartment, or as background music in social settings. Very few people talked about actively listening to music for the sole purpose of listening, but when probed, everyone said that they enjoyed going to concerts.

2. How do you consume music? Albums, playlists, radio etc.

Playlists and charts on streaming services like Spotify was by far the most popular way in which the interviewees consumed music. The ones that previously stated that they actively listen to music all said that they prefer listening to albums in an active listening setting, but even they preferred streaming services and playlists when listening while doing other things.

3. Do you follow any specific artist? If so, how did you come to follow that artist?

All interviewees except one could mention specific artists that they like. When probed about how they came to know that artist, there was a theme of not finding them through curated playlists or via recommendations from friends. Some people said that they actively watched out for new music from their favorite artists, while others said that who their favorite artists were changed often, and was more connected to liking a specific song than the artist in general.

4. Have you played or do you play any instruments?

A few people had started playing an instrument as a child, but stopped before reaching high school. Three people actively played instruments today. All of these three people are multi-instrumentalists to different degrees.

5. Have you every studied music theory?

All of the three people actively playing instruments today had studied music in high school, and two of them had continued studying music through higher education. Other people mentioned having music classes in school as children, but when probed on their music theory knowledge, none of these people showed any real knowledge on the subject.

6. Do you have any experience with writing music, producing music, or DJing?

Five out of the ten interviewees had tried their hand at writing a song. Seven interviewees had tried producing music to some degree, ranging from using music production apps on their phones to professional grade software on their computers. When probed, many of these people had quickly given up on music production, as there is an extreme amount of knowledge in both music composition theory, music production theory, and knowledge about the music production software needed to get a good result. But everyone that had tried music production said that they found the experience satisfying, and whished they had spent more time with it before giving up. Only two of the interviewees had tried DJing, and both of them now do it professionally.

14.2 Gaming Background

7. Do you play video games? If so, what kinds and how often?

All except one interviewee had played video games in some form. Ranging from console and computer games multiple times a day to phone games once a month.

8. What motivates you to play games?

The interviewees with most gaming experience mentioned having a fun experience as one of the biggest motivating factors of playing games, while the interviewees who mostly played games on their phone said that it was a good pass-time activity and a fun thing to do when they were bored. One person said that they mostly played puzzle games on their phone because they enjoyed the mental challenge.

9. What are your thoughts on and experience with music games?

Everyone who had played video games had also tried some music games. The seemingly most popular ones being Sing Star and Guitar Hero. When further questioned, none of the interview subjects said that they actively sought out music games in particular. Many also indicated that they though music games in most cases meant having to buy physical peripherals in order to play it.

14.3 DJ Video Questions

10. What did you think of what you just watched?

There were some different opinions on the video, but the majority of interview subjects liked what they saw. For two of them, the music was too hectic and hard to follow to be enjoyable. Many interviewees also mentioned that they had never seen DJing like this before.

11. Can you explain what the DJ was doing?

As expected, the two people with DJing experience could point out a lot of what the DJ was doing. But people with music production experience, and even the musician with no DJing experience, had trouble accurately describing the techniques used by the DJ in the video, mentioning only that he manipulated the vinyl and crossfader to create scratching sounds. When probed on what they thought the buttons he was hitting did, most people answered something to the affect of "the buttons play sounds".

12. What are your thoughts on DJing in general?

All participants had had some experience with DJs, be it through going to nightclubs or watching DJing videos on YouTube. Three people said that they thought Spotify playlists could do an equally good job at parties, but that after watching this video they appreciated that some DJs do more than just switching songs. Others said that they enjoyed DJs at night clubs, and that they could tell the difference between good DJs and bad DJs not only in their song choices, but in how they transitioned between songs and mixed songs together on the fly.

14.4 Music clip questions

13. What did you think of the music you just heard?

The general consensus seemed to be that the music was "ok" or "cool", or that they liked it in general. None of the interview subjects really had too much else to say about it, even when probed.

14. Could you try to talk about specific parts of the song? Song structure, specific instruments, etc.

All three musicians mentioned that they liked the build-up to the second part, and how the second part was a nice release to that build-up. The two DJs, including some of the participants who had dabbled in music production talked about the drums and bass, mentioning genres like Trap and Dance Hall. Many of the non-musicians had trouble with talking about specifics in the music, other than general phrases including "I liked the vocals", "I though it had good reggae vibes", and "the singer had a nice voice".

14.5 Gameplay Questions

15. What are your thoughts on what you just played? What did you like? What didn't you like?

The responses to this question were mostly positive. Subjects said that they liked the music, and the feeling of "doing those cool dj things from the video". Some mentioned that they liked making the enemies happy, and that they thought the story premise (told to them by the interviewer/observer) was a cute way of presenting such a non-traditional game. The game being quite hectic was seen as a positive aspect by some, and a negative aspect by others. One participant said that because failing the levels could happen so, failing just made her just want to try again even more, sort of like the famous game Flappy Bird. Another participant said the hectic and quick nature of the levels made him stressed out, but that this stress felt like it turned into a deep focus as he got better and better at the game.

16. What are your thoughts on your creative freedom and control of the music in the game?

One interviewee said that they felt a creative freedom to some degree, but because the play session was so short, they didn't feel like they had fully explored the possibilities available to them. Another participant said that they enjoyed the feeling of stringing together enemy relevant actions building up the streak multiplier, and that they felt that they could achieve this in many different ways. Several participants said that they felt a real control over the sound output of the game. That they were the ones choosing what sounds that should be played at any given time. The two professional DJs were less positive to the creative freedom, saying that it felt limited compared to a traditional DJ setup. Digging a bit deeper on those comments revealed that those feelings mostly came from not being able to play their own songs and setting up their own queue points. Also they though that the DJ-controls were missing some way to apply audio effects or at the very least eq filtering. Lastly with the game being so punishing, some participants said that they didn't feel like they had the time to explore what was possible, but conceded that playing the game longer, and getting to more advanced levels probably would help.

17. How would you compare this game to other mobile games you might have played?

One participant said that to her, it felt like a mixture between a puzzle game, and something hectic like Flappy Bird or Super Hexagon. Several participants said that they thought the concept was quite original, arguing that many of the mobile games they had played were mobile versions of existing game genres originally made for other types of gaming devices. "This feels like it's designed for phones".

18. How would you compare this game to other music games you have played?

"This game feels different in that it doesn't force me to do any specific actions, like for instance Guitar Hero does" was one of the responses to this question. Many of the subjects said that it did not feel like any music game they had played before, and that it felt more like a puzzle/action game than a classical rhythm game. One interviewee said that other music games had made him feel like he was playing an actual instrument more than this game did, but that this game in turn made him feel like he had an actual impact on the sound output compared to other games.

19. How would you compare the actions you were performing to past musical experiences you might have had?

All musicians mentioned that after getting to grips with the controls and mechanics of the game, they sometime got into a state of intense focus, a feeling similar to playing a difficult piece of music on an instrument, or being "in the groove" with a band. But the two DJs also mentioned the their past DJing experience made the controls available in the feel somewhat limited.

20. What are your feelings on playing the game with the audio muted?

Several participants said that the game lost all of its meaning with the audio off. "The fun of the game is hearing the musical results of what you are doing, not just pressing the buttons and watching them flash in different colors".

21. Do you have any thoughts on what could be improved in future iterations?

Many of the participants mentioned that they would like better visual feedback on actions like when they had successfully completed one part of an enemies required actions. The more experienced gamers also mentioned better feedback on when you were "on a roll" or had a long streak without failing. The test subjects also mentioned a wish to be able to use their own songs in the game. "Having the game automatically create loops from your own songs, and having the enemies force you to mix them in new and interesting ways would be awesome!". One of the participants who mentioned not feeling total creative freedom mentioned turning off all quantization as a possible solution. "The parts of the game where I unlocked the quantization of a deck made me feel much more in control of what was happening. Doing something more with those parts might be good."

14.6 DJ Video Revisited

22. Have your feelings changed at all with regards to this video? If so, how?

A majority of the test subjects said that they had a better appreciation for what the DJ was doing. One of the interviewees that had previously said that the music was too hectic to follow said that they now had an easier time "following the action".

23. Can you explain what the DJ was doing?

Six of the test subjects that previously couldn't say anything specific now said that the DJ must be able to jump to different parts of the song with the buttons he is hitting, and also talked about him having different songs on the two different "sides" of the mixer. Two of the interviewees still used very non-specific terms when talking about what they had seen.

24. Have your thoughts on DJing in general changed at all? If so, how?

Again, many of the participants said that they had a better appreciation for what the DJ in the video was doing. One participant who had dabbled in music production said that he wanted to try his hand on DJing after playing the game and watching the video. He previously had not though of DJing as being such a creative thing.

14.7 Music Clip Revisited

25. What are your thoughts on the music you just heard? Have they changed since last time?

The general consensus was that the participants felt like they knew the song better, they could hum along to the melodies and tap along to the drum rhythm. One interviewee pointed out that this could simply be the case because they had heard different parts of the song many times and not because they had played with it in a game.

26. Could you try to talk about specific parts of the song? Song structure, specific instruments, etc.

There was no real difference in the responses to this question, other than that more of the participants talked about the difference in the two parts of the song rather than just about the song as a whole.

Part V
Discussion and Conclusion

15 Discussion

This chapter will discuss different parts of the project, presenting both positive and negative sides of each part.

15.1 Design Process

Originally, I had the thought of doing many small prototypes to test out different music interaction solutions and how they might affect the player in different ways. But trying to come up with original design solutions for both music interaction and music games proved to be one of the toughest challenges of this entire project.

Sprint by Knapp [7] bases most of its steps on feedback from a group of people with an investment in the problem at hand. There was no outside feedback during the design phase of this project. This lack of feedback might have caused potentially good solutions to be ignored, or too much time to be spent on bad ideas.

At first, I tried to design the game as a whole, thinking about gameplay and music interaction as one thing. But as the pre-study shows, there is a clear divide in previous works, where one side focuses on pure music interaction and the other focuses on music games without any real attention given to the interaction between the player and the device the game is played on. Acknowledging this divide and splitting the design up into two phases, where the first phase focused on music interaction design and the second phase focused on gameplay design proved to be very helpful. After reaching the interaction design solution presented in this report, designing the gameplay part became much easier.

Conducting a focus group with work colleagues helped get rid of many small design quirks that might have unnecessarily impacted the test results in a negative way.

15.2 Interaction and Gameplay

All in all, the reaction to the game was positive. The test subjects said that they enjoyed the feeling of controlling the music, and that the concept felt like a fresh take on the music game genre, aligning with my findings that most music games today focus on rhythm pattern interactions. There were some problems with the music interaction. Some of the participants had trouble wrapping their heads around the concept of queuing up segments to be played on the next beat, finding it tricky to remember to trigger a segment before the actual beat hits. This could probably have been aided by better visual feedback and a slower ramp up in difficulty, which would remove some of the entry barriers that Denis and Jouvelot warns about in their paper [36]. Another problem showed itself in switching between loops. The required player input of swiping in the same direction twice did not feel natural for most participants. But moving away from natural interaction and one-to-one mappings of real-world DJ equipment seemed to work out well in that the participants did manage to do some relatively complex operations like beat-matching, beat-juggling, and real-time triggering of queue points without having any previous knowledge about how DJ equipment typically works.

The story premise told to the participants by the observer seemed to resonate with people. Including this more into both the graphics and through something like cut-scenes and dialogue would most likely help to achieve what Malone calls intrinsic fantasy. By not telling the player how to defeat the different enemy variation, I feel like some curiosity, as described by Berlyne and Piaget and what Malone calls cognitive curiosity, was achieved. One problem with this lack of information given however, was that the players that sometimes failed the levels too fast, not realizing that just keeping the music going, without doing the enemy suggested action was still a viable option for a limited time. The three star ratings given for each enemy defeated, in addition to the overall score given for each completed level, created a nice subgoal for the player. Even though it did not have any real affect on gameplay, some participants showed a willingness to go back to get the maximum number of starts on each level. One could have improved and expanded upon this concept by giving the players rewards for completing a level with the maximum amount of start.

In regards to music affecting the gameplay, there is still much more work to be done. Without changing or adding to the code of the prototype, one could simply add musical loops that don't fit together from the get go. Requiring the player to change rate and pitch to match different loops together. One could also look at song structure, requiring the player to take this into account, starting with intro parts, and progressing naturally through a traditional song structure. The enemy design and animation could also take the music currently playing as input parameters, having different frequencies or rhythms changing its behavior.

15.3 Technology

The choice of using the AudioKit framework caused some problems in the development of the prototype. Some features that should have been easily implemented, like jumping to different location in an audio file while playing, caused auditory glitches not acceptable in a music game. This necessitated the implementation of hacky solutions to overcome these problems. These solutions are not something I would whish to be a part of a potential product released to the public, as they introduce bugs and other performance issues. Loading duplicates of every audio file in some cases caused the game to slow down and finally crash after being played for longer than 10 minutes. Setting the rate of audio playback using AudioKit provided functions would sometime not "take", causing some segments to be stuck at a slower rate than the rest of the segments. This problem would eventually fix itself, but caused confusion among the players nonetheless. Looking back, it might have been smart to invest some time in learning low-level audio programming for the iPhone in order to get exactly the results I wanted without having to resort to hacky solutions. If I continue working on this prototype towards a finished product, I will definitely look into other audio frameworks, as mentioned in the technology section of the pre-study.

There were no problems in using the iPhone as both a development and testing platform. Hardware wise, the iPhone 6 handled everything thrown at it very well. Using an old iPhone meant that I could not explore haptic feedback as mentioned in the technology section of the pre-study. I believe that haptic feedback has great potential in music interaction, and would have loved to explore this more.

The SpriteKit framework also worked out very well. The prototype ran at a solid 60fps at all times, with no slow-down. In early stages of development, the audio processing done by AudioKit ran on the main thread. This meant that glitches in the audio framework would impact the frame rate of the prototype. Moving all audio processing to separate threads solved this problem. SpriteKit has tremendous support for physics, animations, custom shaders, and lighting, creating an almost unlimited improvement potential to the visual part of the game.

15.4 User Perception

The prototype managed to get the participants into a state of flow to some degree. This was backed up by both the observations and the interviews. Two participants were observed to restart levels multiple times, without regards for the observer or the outside world. Similar, but not as intense behavior was also observed in several other participants. The musicians also said in the interviews that the game had at some points made them feel focused in a way similar to the focus they felt when playing instruments.

The prototype not having a proper challenge and skill ramp-up caused some players to have to grind against the game for some time before reaching a skill level where they felt that they had a sense of control over their actions. This goes against the three elements presented by Sweetser; Challenge, Player Skills, and Control. But after this initial hurdle, the levels progression in challenge looked to support the players' skill development in a satisfying way. Clear goals was something the game excelled at, having both the long term goal of completing an entire level, the intermediate goals of defeating each enemy, and the subgoal of getting a three star rating on every enemy. One of the games weaker points was visual feedback. As I am not an artist or animator, and there was limited development time, the enemy especially did not always give the player the feedback they required, sometimes making them uncertain if they had completed a set of actions successfully.

The prototype did not support or create any opportunities for social interaction, which is one of the criteria of GameFlow. Being a creative game, one could have looked at the possibilities of collaboration between players, or sharing highlights or high scores together with the musical output created by the player.

Through both the observations and the interviews, the participants did seem motivated to keep playing the game to get better at it. Both in terms of the game itself and in terms of DJing, where one participant stated that he was going to look into DJing as something to do as a hobby.

16 Conclusion

This chapter will for each research question present a conclusion based on the results relevant to that question.

RQ1: What is the current state of the art in mobile music interaction?

Through the prestudy, different works on both mobile music interaction and player-music interaction in video games were presented. It was shown that there was work left to be done in the abstraction of real-world music equipment to multi-touch screens. On the game side, it was shown that most music games focus on rhythm-pattern interactions, and that there is a lack of games where music manipulation is what affects gameplay, and not just having musical output as a side-effect of what is happening in the game.

RQ2: How is the player's understanding and appreciation of musical structure affected by the game?

By playing a piece of music to the participants before and after playing the prototype and asking the same questions after each listen, a conclusion to this question can be drawn. The interviews revealed that some appreciation of music structure might have been gained from playing the game, changing the way in which some participants talked about song structure. However, it is uncertain whether this new appreciation came from playing the game, or just having listened to the same song multiple times.

RQ3: How is the player's understanding and appreciation of music production and performance techniques affected by the game?

Similarly to the music played to the participants, a video of a DJ performance was also shown before and after playing the prototype. A majority of the participants showed an increased knowledge in DJing in general, talking about having two songs playing on "each side", and how the DJ must be able to jump to different parts of the songs be pressing different buttons. Without ever telling the player that what they were doing is similar to DJing, and using an interface not resembling a traditional DJ setup, the players still gained knowledge on the subject. This shows potential in simplifying and abstracting interactions for mobile devices, using the knowledge people already have regarding the use of these devices instead of requiring other outside knowledge.

RQ4: How can interaction with pre-recorded music make the player feel like they are taking part in the musical performance?

Through both observations and interviews it was found that some participants entered a flow state similar to that of performing music while playing the prototype. Using the GameFlow model, there is potential to tune the game in future iterations to help more players reach this state. Participants also expressed that this prototype gave them the feeling of more creative freedom than what they got from more traditional rhythm games. Creating a feeling of flow together with maximizing creative freedom easily accessible to non-musicians looks to be a good solution for creating feelings similar to those experienced when performing music, in players.

RQ5: How does the implemented music interaction affect the player's enjoyment of the game?

The music interaction design presented in this project affects the player's enjoyment in different ways. Some found it motivating, expressing that their increasing skill level made them want to keep playing the game. Others, with less experience in music and gaming, found the mixture of controlling music and solving gameplay challenges at the same time to be hectic and sometimes frustrating. Letting the participants play the game with the music turned off, revealed that the music interaction and manipulation is what made the game fun to play.

17 Future Work

There are countless exiting possibilities for improvements and expansion on the presented solutions, including:

- Exploring new audio frameworks to be able to implement all the desired functionality, like full forwards and backwards rate control of playback.
- Creating better visual feedback through enemy animation and animations in general.
- Creating new challenges not only from improved enemy design but also from the use of initially incompatible music loops.
- Allowing the players to use their own music, automatically creating loops and segments from the music provided by the player.
- Incorporating story into game
- Incorporating the fantasy and story premise through the design of enemies and visual player controls
- Social interaction between players, through sharing highlights, high scores, or collaboration.

In terms of continuing the research work, future work would include:

- More extensive testing on a larger sample size of potential users
- Testing totally different design solutions, and how these might give more insight into the research questions.

18 References

[1] Michael Bull, *Sound moves: iPod culture and urban experience.*: Routledge, 2015.

[2] Angela Chang and Hiroshi Ishii, "Sensorial interfaces," in *Proceedings of the 6th conference on Designing Interactive systems*, 2006, pp. 50-59.

[3] Koray Tahiroğlu, Atau Tanaka, Adam Parkinson, and Steve Gibson, "Designing musical interactions for mobile systems," in *Proceedings of the Designing Interactive Systems Conference*, 2012, pp. 807-808.

[4] Lilly Irani, Janet Vertesi, Paul Dourish, Kavita Philip, and Rebecca E. Grinter, "Postcolonial computing: a lens on design and development," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2010, pp. 1311-1320.

[5] Victor R. Basili-Gianluigi Caldiera and H. Dieter Rombach, "Goal question metric paradigm," *Encyclopedia of Software Engineering*, vol. 1, pp. 528-532, 1994.

[6] Victor R. Basili, "Software modeling and measurement: the Goal/Question/Metric paradigm," Tech. rep. 1992.

[8] Briony J. Oates, *Researching information systems and computing.*: Sage, 2005.

[7] Jake Knapp, John Zeratsky, and Braden Kowitz, *Sprint: how to solve big problems and test new ideas in just five days.*: Simon and Schuster, 2016.

[9] Andrea Fontana and James H. Frey, "The interview: From structured questions to negotiated text," *Handbook of qualitative research*, vol. 2, pp. 645-672, 2000.

[10] Michael D. Myers and Michael Newman, "The qualitative interview in IS research: Examining the craft," *Information and organization*, vol. 17, pp. 2-26, 2007.

[11] Lalya Gaye, Lars Erik Holmquist, Frauke Behrendt, and Atau Tanaka, "Mobile music technology: Report on an emerging community," in *Proceedings of the 2006 conference on New interfaces for musical expression*, 2006, pp. 22-25.

[12] Luciano V. Flores, Marcelo S. Pimenta, Eduardo R. Miranda, Eduardo A. A. Radanovitsck, and Damián Keller, "Patterns for the design of musical

interaction with everyday mobile devices," in *Proceedings of the IX Symposium on Human Factors in Computing Systems*, 2010, pp. 121-128.

[13] Jef Raskin, "Intuitive equals familiar," *Communications of the ACM*, vol. 37, pp. 17-19, 1994.

[14] Maura Sile OMódhrain, *Playing by feel: incorporating haptic feedback into computer-based musical instruments.*: Stanford University, 2001.

[15] Peter Nagy, Virtuoso Piano Free 2, 2010.

[16] Pedro Lopez, Alfredo Ferreira, and J. A. Madeiras Pereira, "Battle of the DJs: an HCI Perspective of Traditional, Virtual, Hybrid and Multitouch DJing.," in *NIME*, 2011, pp. 367-372.

[18] greySox, Drum Meister, 2010.

[17] algoriddim GmbH, Djay 2, 2013.

[19] David Kirsh and Paul Maglio, "On distinguishing epistemic from pragmatic action," *Cognitive science*, vol. 18, pp. 513-549, 1994.

[20] Native-Instruments, iMaschine 2, 2015.

[21] Markus Waldboth, LP5, 2013.

[22] Thomas W. Malone, "What makes things fun to learn? Heuristics for designing instructional computer games," in *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*, 1980, pp. 162-169.

[23] Penelope Sweetser and Peta Wyeth, "GameFlow: a model for evaluating player enjoyment in games," *Computers in Entertainment (CIE)*, vol. 3, pp. 3-3, 2005.

[24] Mihaly Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New, York, NY: Harper Perennial, March 1991. [Online]. http://www.amazon.com/gp/product/0060920432/ref=si3_rdr_bb_product/104-4616565-4570345

[25] Opal Limited, Bloom, 2008.

[26] Seth Sandler, Node Beat, 2011.

[28] Martin Pichlmair and Fares Kayali, "Levels of sound: On the principles of interactivity in music video games," in *Proceedings of the Digital Games Research Association 2007 Conference" Situated play*, 2007.

[27] Apple, Garage Band, 2011.

[29] Kenneth B. McAlpine, Matthew Bett, and James Scanlan, "Approaches to creating real-time adaptive music in interactive entertainment: A musical

perspective," in *Audio Engineering Society Conference: 35th International Conference: Audio for Games*, 2009.

[30] Alexander Wroten, "Developing Musical Video Games Through Principles of Player-Music Interaction A Thesis Submitted to the Faculty," DARTMOUTH COLLEGE Hanover, New Hampshire, Ph.D. dissertation 2011.

[31] Karen Collins and others, "An introduction to the participatory and non-linear aspects of video games audio," *Essays on sound and vision*, pp. 263-298, 2007.

[32] Owen Edwards, "Simonized: In 1978 a new electronic toy ushered in the era of computer games," *Smithsonian Magazine*, 2006.

[33] Ultra Combo, Accessed: 2017-04-20.

[34] Daniel E. Berlyne, "Structure and direction in thinking.," 1965.

[35] Jean Piaget and Margaret Cook, *The origins of intelligence in children.*: International Universities Press New York, 1952, vol. 8.

[36] Guillaume Denis and Pierre Jouvelot, "Motivation-driven educational game design: applying best practices to music education," in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, 2005, pp. 462-465.

[38] Anthony Picciano, Tap pressure strength detection using accelerometer, Accessed: 2017-04-23.

[37] Smartphone OS Market Share, 2016 Q3, Accessed: 2017-04-26.

[39] Greg Schiemer and Mark Havryliv, "Pocket Gamelan: Tuneable trajectories for flying sources in Mandala 3 and Mandala 4," in *Proceedings of the 2006 conference on New Interfaces for Musical Expression*, 2006, pp. 37-42.

[40] Eduardo Reck Miranda and Marcelo M. Wanderley, *New digital musical instruments: control and interaction beyond the keyboard.*: AR Editions, Inc., 2006, vol. 21.

[41] Tina Blaine and Sidney Fels, "Contexts of collaborative musical experiences," in *Proceedings of the 2003 conference on New interfaces for musical expression*, 2003, pp. 129-134.