Universiteit Utrecht

Master Thesis

Game and Media Technology

Music Interaction Techniques in a Touchless Environment

S. L. Geertsema

3944980

Supervisor

R. C. Veltkamp

23 April 2015

Abstract

Music is often used for setting a mood or relaxing. But it could also be used as a method for expressing an emotion for people who have trouble to do so. Since about twenty percent of modern teenagers (in the Netherlands) experience this problem, a solution is needed. One such solution is an environment where teenagers can use several sense-based methods to express their emotions. The Moodroom is such an environment. It uses colours to express emotions. This project is a first attempt to come to an expressive user interface to interact with music.

The goal of this project is to implement and verify various techniques for music interaction. Possible interaction techniques are gesture-based menus, predefined gestures and undefined gestures. With interaction techniques the user is able to control the certain parameters, such as tempo and attack. These are used to provide corresponding music. Possible techniques for providing fitting music are either with recorded music or music that is algorithmically composed. Recorded music can be found with the use of a database containing pre-calculated parameters, or by adjusting a certain song to the parameters. Generated music is generated in real-time, based on the parameters.

A qualitative evaluation has been performed to verify which techniques (gesture-based menus and predefined gestures for interaction and database-based recorded music and generated music for music) are the most suitable for this purpose. With the current implementation, 77% of the participants prefers a gesture-based menu over a predefined gesture interface, and 80% prefers the recorded music selected with the use of a database over the generated music.

Contents

1 Introduction

Music is often used to express one's feelings or emotion, either by playing a musical instrument or listening to music. Both ways have their own strengths and weaknesses. Playing live music gives the player freedom to adjust music to their own taste, or play something completely new, reflecting his feelings. However, it requires skill and knowledge to play. No particular skill is necessary in order to listen to music. However, it is a less personal expression of one's emotion if it is created by another person.

Music can be used for more than only relaxing or reflecting ones mood. At least one in every five teenagers experiences physical or emotional problems [1]. In order to treat these problems, it is important that they are able to express their feelings. However, it is often difficult for teenagers to do this, so different sorts of therapies are developed to ease this process. One such method is with the use of expressive arts [2]. This is a creativity based form of therapy, where participants use all sorts of art to express themselves. Usually they choose one art at a time, but to provide a more immersive experience, researchers are looking for ways to combine arts. One method could be to create an art-based environment that reflects their emotions. Three years ago an environment, the *Moodroom* (see section 2.1), was developed for this purpose. Although users can only adjust the room visually, evaluations showed that users were able to express their emotions. This effect could probably be reinforced by adding more sense-experience, since humans use all their senses to experience emotion. For instance sounds or smells can bring memories back and induce strong emotions.

For this project sound in the form of music is chosen. It will concentrate on the interaction techniques in both interaction and music needed for an intuitive and immersive experience.

Music is usually played with a musical instrument or from a recording. But since the music should be interactive, a different approach is needed. For over a century, various forms of musical instruments that are played differently than the usual ways (keys or strings) are developed.

1.1 Goal and Contribution

The aim of this thesis is to improve the interaction with these sense-based environments by comparing interaction methods with different methods for creating music. This is achieved by creating a software program implementing different techniques for generating music and interaction with the system. This comparison can then be used in projects where users want to make music with a non-touching interface, as used in the Moodroom.

A small experiment with 30 participants is created where they are asked to rate different techniques implemented for this project. Concluding from this project, most people (76.7%) prefer a gesture-based menu over a predefined gesture environment. 80% of the participants prefer recorded music over generated music. For a Moodroom-like project, a clear menu in combination with real music would be the best implementation.

1.2 Structure

The following chapter will provide background information on different techniques used in this thesis, the interactive environment Moodroom and the Kinect, and on generated music.

In Chapter 3 the methodology of the thesis will be explained.

Chapter 4 will elaborate on how different features have been implemented. This chapter is split in three sections: one governing the interaction with the system, one focusses on playing and generating music, and finally how it all is implemented.

Chapter 5 discusses the performed experiment and the results.

Chapter 6 presents the results from this thesis and experiment, and according to these, a conclusion will be drawn. It will also discuss some possible future work.

2 Related work

In this chapter the techniques used in this thesis will be discussed, placing them in the proper context of previous research.

2.1 The Moodroom

In 2012, the Waag Institute developed an interactive installation called the Moodroom. With the use of embodied learning, the Moodroom is a (fun) way to provide teenagers a tool for showing emotions, with a final purpose of improving their happiness and wellbeing [3].

The Moodroom is an environment that consists of three screens, projected by three beamers and one Kinect, controlled by a laptop. An example of the layout is shown in Figure 1 . With the use of gestures, the 'emotion' of the room is adjusted. This emotion is visualised with the use of colours, shapes and point clouds.

With the use of the Kinect it is possible to adjust the environment completely intuitively to represent the internal world of experiences. The various settings of the Moodroom are changed using a menu, navigated with the use of gestures (waving, swiping, pointing, etc.). For instance, when selecting a background colour, waving one's hand from right to left will slide through the spectrum, until the correct colour is found. Waving up or down will alter the saturation of the colour.

There are three variables:

1. **Background:** the overall colour of the Moodroom.
2. **Pattern:** a pattern consists of several elemental shapes, triangles, rectangles, ovals, dots and plus-signs.
3. **Point cloud:** the participant can also be shown on the screens, represented by a point cloud.

The only sense used in the current Moodroom is visual; none of the other senses are implemented. To provide a more immersive and possibly better experience, more senses could be used.

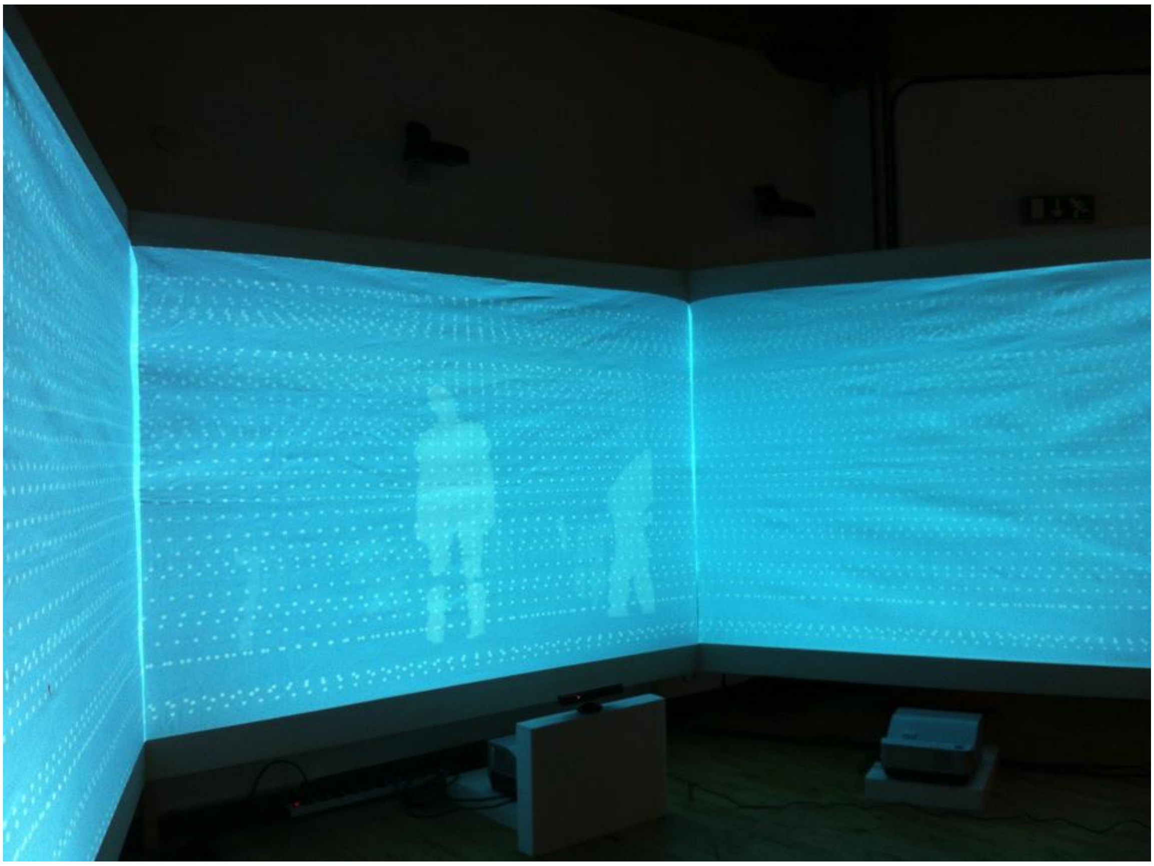

Figure 1.The Moodroom.

2.2 Music Interaction

Music is one of the most brain-demanding activities. Most other activities, such as vision and language, use primarily localised parts of the brain, but music seems to involve it almost completely [4]. It is also the sole artistic activity that is unanimous for the whole earth. Societies without writing or visual arts exist, but music is always part of a society. So it is no wonder that many research into characteristics (for instance whether music is able to influence humans, see section 2.2.2) and properties of music is done, or inventing new musical instruments (such as the Theremin, see section 2.5), creating new methods to interact with instruments (with interactive setups like the Brain Opera, see section also 2.5). Most of these research is combined in one field that is Music and Human-Computer Interaction.

Music Interaction (short for Music and Human-Computer Interaction) covers the design, refinement, evaluation, analysis and use of interactive systems that use the computer for any kind of musical activity [5] (chapter 1). It is a broad field, since all sorts of musical roles exist, with each their own requests, qualities and goals: performer, composer, listener, teacher, designer, and many more.

In the next section some of these research that is related to this thesis will be discussed. Section 2.2.1 will focus on interfaces, how open and easy the interface should be.

Section 2.2.2 then is about the influence of music on human beings, how easy to manipulate we actually are.

Besides manipulation, we can use music to express our emotions. But not all emotions are expressible in music, and some terms describe this emotional music more correctly than others. Section 2.3 will discuss the Geneva Emotional Music Scale and some related problems.

Finally music generation will be discussed in section 2.4.

2.2.1 Interfaces

Since the prehistory, humans have created musical instruments. From bone-flutes and animal-skin drums to synthesizers and electric guitars, numerous different layouts, shapes and sounds exist. Before the invention of digital instruments the inventers were dependent of physical boundaries and properties of material and humans. Besides musical instruments, also the design of other devices like mixers and synth-bays are based on physical boundaries: shapes of sliders, knobs, location of wires, etc. With the coming of digital interfaces, important questions arise: should designers of digital interfaces retain the layout of the original device (called *design metaphors* [6])? Or would it be better to research newer and more suitable layouts? Stowell et al. suggest that keeping layouts the same for the sake of familiarity is a bad idea, since that will constrain interactions and would be unintuitive in a digital world [5] (chapter 8). Music is defined as "rich" and "open". Rich means that the expressiveness of music is not limited to a single instance. A musical piece can have many different (both emotional and non-emotional) responses with listeners. It can also be rich with structure, like themes and layers. Open means that music can be used as "music", a structured medium with predefined rules (chords, genres, etc.), or without these rules, resulting in new genres, or other uses of music.

To not limit one in this rich and open nature of music, the interface that is used to create or edit music should also be rich and open. Various examples of interfaces are discussed in [5] (chapter 8), where the focus is always placed at the user, to create an interface where the performer has the freedom and possibilities to perform freely, without feeling limited by the interface.

One way of implementing this is proposed by Stowell et al., a novel way for creating interfaces, based on an abstract grammar. This interface can then be adjusted by the musician to his or her own style, while retaining the underlying features that are defined in the grammar. By doing this, it retains the rich nature that is stored in the grammar, and it leaves room for a unique interpretation of the performer.

As discussed before, a musical interface should be designed thoughtfully. It should be rich and open, but is there a trade-off between a complete interface or an easy to use interface? In [4] (chapter 2) McDermott et al. defend their claim that a (musical) interface should not be too easy. Getting used to the interface should be hard, although preferable with an interface that is quick to understand. They argue that interacting with musical instruments is different than interacting with non-musical instruments, since the goal is different. Most performers play an instrument for fun, for entertainment, to relax. There is no goal to achieve: the only 'goal' is to play [7] [8]. This is where it is different than other instruments, or tools. A tool is made to be useful, to help the user with an otherwise difficult task [9]. For this reason, it would be annoying for the user if this tool is hard to use: it should enlighten the task, not making it more difficult.

With a musical instrument however, this is different. Playing an instrument becomes more interesting and rewarding when it is harder to play [10] [11]. One feels more content or proud when a difficult piece of music is finally performed correctly. Or maybe incorrectly, though deliberate played 'wrong' (not as imagined by the composer), where it results in a unique version of the original piece.

The freedom of this 'personal touch' could be limited when the interface is created in such a way that it is simplified too much. This simplification is often done for the sake of usability, as a warm bath for new users. When one becomes more experienced, one could desire for more possibilities. If the interface is too simple, there is no room for more possibilities, since these would distract.

Various dimensions of difficulty are explained: physical [12], dexterity and coordination, going from imagination to realisation, layouts of interfaces, polyphony, repertoire, tuning and concepts. Simplifying on one (or more) of these terms leads to a completely different experience, where often the initial goal or idea for the instrument is lost.

Many reasons for a difficult interaction are provided.

First, open-endedness and long-term engagement. When an interface is difficult, the user has to invest time to understand it (a *learning curve* [13] [14]). This leads to a stronger 'relationship' between the user and the interface, since the user never will fully understand it, providing an ongoing goal [15].

Secondly, creativity. A difficult interface provides more possibilities, no options are left behind for the sake of simpleness, hiding it behind menus [11]. This leaves the user with a lot of freedom to perform exactly like he means to.

Thirdly, displaying virtuosity [16]. A performer receives respect (or feels proud) when a very difficult piece is performed [17]. A certain level of prestige is associated with the commitment of time and energy to attain mastery [18]. A simple interface might be complicated to design or use in a professional way, but an audience often will not see this, thus lowering the level of prestige [19] [20].

And finally communicating effort. Audiences are used to the fact that an obvious complex task requires more skill than a simple task, since a complex task often requires more time to master [21].

But besides difficult interaction, sometimes it is okay to be simple.

First in the case when there is no need to perform complicated musical pieces. Sometimes the user just wants to express something without difficulty, for instance in a video game.

Second tasks that are peripheral or technical. Most performers do not want to be bothered by tasks that are repetitive or annoying, such as tuning strings; they want to perform.

Thirdly to learn a new instrument. New performers would be greatly helped if tools would exist that ease the first steps in learning a new musical instrument. For instance limiting the keys of a piano to only white keys. When the skill of the performer increases, these limitation can be lifted gradually.

Finally when teaching. A digital tutor could allow for a more personal approach, with no limits to time or skill.

Concluding from this: a musical interface should be:

- Rich, to provide the performer with means to actually express his self.
- Open, to ensure enough freedom for the performer to create his personal piece of music
- Hard to use, for a stronger relationship between the performer and the instrument
- Yet simple enough to not be annoying.

2.2.2 Effect of music on humans

Besides entertaining, music can be used to provide information to the listener. For instance by providing auditory feedback when typing on a mobile phone [22], or alerting users with an alarm [23] However, these are often based on short sound clips. In [5] (chapter 4), Bramwell-Dicks et al. state that the goal of these clips is to transport information. Various implementations of this are present in personal computers and other devices, where short sound clips are used to inform the user of an error, or a new email [24]. Music fairly plays a role in this context, although it could provide additional feedback. They suggest that (ambient) music might provide a useful feedback, since humans are very sensitive for music.

Music is often used to influence the behaviour of humans. Various examples from a broad research field are given, with possible HCI uses.

For instance in a supermarket, where slow music makes the shopper walk slower, thus inviting them to buy more and spend more money [25]. Musical genres do influence people: classical music makes people buy more expensive products [26]. The nationality of the music can influence the choice of wine: with French music humans tend to buy French wine, with German music humans tend to buy German wine [27]. Maybe this could be extended to online shopping?

Or while dining in a restaurant, where the tempo of the music seems to influence the speed of dining ([28], contrasting with [29]). And although an online equivalent of dining is absent, the idea of tempo influencing the speed of actions could be useful.

Another location where the genre of music is important is in the arcade [30]. Easy Listening attracted more older, female players, while Rock or Dance attracted young, male players. Although the focus of the player is on the game, and the music is merely in the background, it does have an impact.

Athletes and in other sports performed better while listening music, probably because it distracts the attention of the discomfort [31] [32] [33]. So maybe the discomfort of a user can also be negated using music?

Instrumental music with a fast tempo and a high intensity has a negative influence on the comprehension of reading for students, although slow music has no significant effect [34].

And finally in the field of psychology, where music can be used to elicit an emotional response from the patient. Although it is not entirely sure that music does directly influence emotions, some evidence does point in that direction [35]. The problem with measuring this effect is that although people are often in contact with music, most of the time this is merely in the background. And listening to a song while performing another task creates a less strong response than when in a concert. So although researches are doing research in this field, it is hard to get strong evidence.

Besides these potential research ideas, there is some research done with the affective properties of music in the HCI field.

First in the field of computer games. Various studies have been done with the effect of music on the gamers: music creates a more immersing experience [36]. Background music increases the stress level [37]. And a game is more liked when the music was chosen by the gamers themselves [38].

Another study, from the 1930's, showed a relation between the typing speed and music [39]. Jazz had a positive effect on the speed, however more errors were present.

Two studies are done in the field of online gambling [40]. Both showed that fast music does not necessarily increase the bets or the risks taken, but the speed of betting does increase. Also fast music could potentially increase the addictiveness of the game.

[41] shows that when listening to the correct, congruent music, more facts were remembered.

Bramwell-Dicks et al. did "*one of the first pieces of empirical research investigation how music can affect someone's emotional experiences, rather than behaviour*". For this they developed an email client with continuously playing ambient music, Ambient Music Email (AME) [42]. At the arrival of an email, a short fitting phrase is played to notify the user. Tests with the system showed that AME was less boring than no music at all, although, due to some problems with the music, the negative impact was larger. But the system itself does show potential, so carefully designed further research in interfaces with music is needed.

2.3 GEMS

Not only has music such a strong influence on the behaviour of human, also emotionally humans are often affected. These music-evoked emotions could be the same as emotions experienced in everyday situations, but no hard evidence is found for this. Measurement of emotions induced by music is hard, since most scales are based on general dimensional or categorical emotion models instead of domain-specific models, and since music-induced emotions seem to differ from emotions experienced in everyday situations, a different model is needed. Zenter et al. created such a domain-specific model, the *Geneva Emotional Music Scale* (GEMS) [43]. To create this model, four different studies were performed to create an extensive, yet not too long, list with terms that can be used to describe experienced or felt emotion.

For the first study a list of 515 (French) terms was gathered from various sources, where all terms were selected to have an affective value. Participants answered one question for each term, whether this term was better suited to describe a certain affective "colour" than another. The resulted list contained 146 terms and was used in the second study.

The goal for the second study was verifying the relation between the terms, emotions and music. For this, 262 participants were asked to listen to songs from their favourite musical genre (limited to classical, jazz, pop/rock, Latin American, and techno), and then rate the 146 terms for three different situations: how often they felt the emotion, how often they perceived the emotion and how often they experienced that emotion in their everyday life. With this ranking a new list was compiled with ten factors that was featured in every list, and besides this list, the terms were re-valued, based on various tests and genres.

The third study was similar to the second study, except on a larger scale (considering participants, 262 vs 801). The list with terms was reduced to make it more manageable to 66 terms. People at a music festival were asked to rate these terms according to how they experienced the music they heard at the festival. With this data, the researchers created a new list with terms, and selected three sets of terms in increasing size, with three, nine and forty items. Most of these emotions are positive, as music most often does induce one positive, since the listener is not in a threatening or potentially sad position.

The fourth and final study was performed to verify the previous found results and compare the list with two other scales: discrete and dimensional. 238 participants listened to sixteen instrumental, classical excerpts and had to rate the music again, with the participants divided in two groups: one with emotions from the discrete and dimensional scales, and one with the GEMScale. Their conclusions are that the GEMScale outperformed both other scales in terms (which emotions are better suited?), agreement (do participants chose the same emotions?) and discrimination (is music better divided by emotions?).

The final list contains 45 terms that can adequately be used to describe induced emotions by music. Smaller sets were also created with 25 and 9 terms (see Table 1). Further tests were held and a significant fit was shown for classical music, although the model is designed to capture emotions induced by music in general.

Various experiments have been conducted to verify the quality of GEMS as a model. In 2010, Vuoskoski et al. performed a comparison of three emotional models (again GEMS versus two general emotion models: discrete and dimensional) [44]. Sixteen excerpts from movie soundtracks were used. GEMS's consistency was less than the other two models, although it did show that it is suitable for music in a wider sense. In 2013, Lykartsis et al. did research in the terms used in GEMS. Originally GEMS was developed in French, and translated to English. For this research they translated it to German.

TERM	**FEELING**
Wonder	Amazement, curious
Transcendence	Perfect, supreme, great
Power	Strong, heroic, full of energy
Tenderness	Soft, sweet
Nostalgia	Nostalgic, remembering
Peacefulness	Calm, relaxed
Joyful activation	Dancing, active, amused
Sadness	Depressed, mourning
Tension	Nervous, stressed

Table 1. GEMS terms.

Conclusions are that the German translation sometimes performed better, probably due to better or better understandable terms and that it does confirm the suitability of the model on a basic structure level.

These experiments and others did use small datasets of audio. To perform an experiment on a larger scale, Aljanaki developed a game with a purpose. This game contains 400 songs in four different genres: classical, electronic, pop and rock. During their research, they found out that some of the GEMS terms are unclear, so those are changed: Wonder to Amazement, Transcendence to Solemnity and Peacefulness to Calmness. They also found out that induced emotions depend on whether participants liked a song, their mood, and in a lesser degree their gender.

The music and the parameters computer from this music used in Aljanaki's research project are the basis for this thesis' project (see section 4.2).

2.4 Music generation

The previous sections were about humans and music and how the two are connected, by either interaction with music, or being influenced by music. The details about the making of music were quite direct: the user performs an action and the music(interface) responds immediately, which resulted in music. But besides this direct approach, it is possible to use the computer to create music in a less direct way, by providing the computer with some parameters, and then let it compose (or generate) music on its own. In [5] (chapter 13) two classes of music generation are described:

Evolutionary computation (EC). This is a way of composing based on the evolution of things. The computer uses some parameters to generate a song. Some selection happens, either by a human listener [45] or by some fitness formula [46]. Some parameters survive, some are removed, and some mutation might happen. These new resulting parameters form the basis for the next iteration.

Generative music (GM). Where EC generates a score, or some parameters, GM does not, but instead it uses some rules or an algorithm to generate music [47]. The idea is that the computer actually makes art (music) on its own, without the help of a human.

One thing both methods have in common is that the final result could have been created using "normal" methods (e.g. by a human).

Although the computer is able to do the generation work on its own, a human listener might provide some useful feedback, as the computer cannot really judge music. This results in some typically HCI problems:

- Humans are slow, a so called fitness evaluation bottleneck [48]. Therefore, EC is often restricted to only a few generations, since a piece of music of 60 seconds does actually take 60 seconds to rate.
- A badly designed graphical user interface that has certain elements that have to be used a lot in the process can slow the user really down [49] [50].
- Users become bored, fatigues and annoyed [51].
- Users cannot remember every previously generated piece [52].
- Users assume the computer should know that a generated piece is bad, or not.
- Since EC is generated randomly, some bad generations might be present.
- The iteration become worse every step
- A typical fitness formula is closed and cannot be adjusted.

Since the problems are HCI problem, the potential solutions can be so too: Simple GUIs, archives of good individuals [53], combining human and computer strength [54] [55], keyboard shortcuts, expressive interfaces (see section 2.2.1), mnemonic visualisations, better representations.

Various interfaces with these solutions in mind have been developed, either based on higher-level grammatical representations [45], time or creative transformations [56].

2.5 Examples of various HCIs

The previous sections were all about new techniques and research in the field of HCI. However, there are many examples of devices that are invented the last few decades. And some of these new devices are based on older inventions.

Figure 2. A Theremin.

One of the first electronic instruments is the Theremin. The Theremin (see Figure 2), invented in Russia in 1928 by Léon Theremin, is an electronic music instrument that is played without physical contact. It features two antennas which conduct electricity. The instrument is played based on the distance of the hands to those antennas. One hand controls pitch, the other amplitude. Although it is touchless, it is hard to play, but the gesture-based layout is intuitive, when the height of the hand increases, so does the volume, or pitch.

A newer approach of this idea is the Brain Opera. This is a production where users are given multiple non-traditional instruments, like the Rhythm Tree (see Figure 3), Melody Easels and Digital Baton [57]. The audience walks in a room where all these instruments are and they play with the various instruments during about an hour. They need to interact with the instruments in various ways: using speech to record samples or use touch and gestures to manipulate sound structures. All instruments are connected with a computer system and after an hour, some musicians will perform the piece of music as it is created by the audience, also on non-traditional instruments.

On a smaller scale provides the Reactable an intuitive solution for interaction with music [58]. On a table a screen is projected providing the user with an interface that can be interacted with by touch of by placing physical objects on the table (see Figure 4). The software recognizes the objects and responds accordingly. Objects are used to generate sound-waves, play sound-loops or to control the generators and other objects.

These described devices are all musical instruments, they need performers to make music. This next device does not, except a user that provides some parameters.

The Oramics (see Figure 5) is a device that is developed in 1957 by Daphne Oram. It features 35mm film strips which have been drawn on that are used to generate sound. The drawings control four parameters: pitch, vibrato, reverberation level and volume [59]. This is similar with the approach taken in this project. These four parameters are fed to a synthesizer that generates a song.

The Moodroom is a modern version of this concept.

Besides physical set-ups, digital solutions are also possible.

Hyperscore is a program that implements the idea of using parameters to generate music on the computer [60]. Its ultimate goal is to provide an intuitive and interactive graphical environment for creating and editing compositions. The basis for each composition are user-created free-formed curves

Figure 3. The Brain Opera, with the Rhythm Trees.

Figure 4. The Reactable.

(see Figure 6), which the software maps to structural and gestural elements. Motives can be designed and are automatically adjusted to the overall sound, so recognisable elements are available.
For more control many different software music synthesizers exist, like CSound, Chuck, Max and SuperCollider [61]. Instead of a graphical user interface, these programs are text-based and require coding skills to generate music. For this project SuperCollider is chosen as it is free, the most dynamic, and object-oriented [62]. The first release of SC was in 1996 and many updates have been released since. Nowadays it is one of the most used sound synthesis environments. It features a programming language that can be used for anything, but is mostly a framework for real-time audio synthesis and algorithmic composition.

2.6 Conclusion

In the previous sections many music interaction techniques have been discussed that are relevant for this project. But since the Moodroom is not meant as a musical instrument, but merely as a tool for expressing ones emotions, some of the research should be adjusted to the needs. For instance, in [5] (chapter 2) it is stated that a more complicated interface results in a longer learning curve, though leading to a more interesting and longer experience. However, for the Moodroom, a short learning curve would be better, since people should be able to use it fast. And there is (hopefully) no need for them to use it more than a few times. So the interface should be easy, yet provide enough freedom for a rich experience [5] (chapter 8).

In [5] (chapter 4), the influence of music on humans is discussed. As mentioned, music has a strong influence on humans, often unconsciously. For the Moodroom, participants have to select music that expresses how they feel, this is the other way around. But the music can be selected with that goal. Also the music has to be selected on the emotion it induces on people.

The Moodroom is closely related to the Oramics: with a few parameters, music is generated. However, modern techniques provide a richer experience, with touchless interaction, better music and more possibilities.

Therefore, the goal is to implement an environment that has an easy to use interface that provides enough freedom for the user to reach his or her goal: create or find music that expresses their emotions.

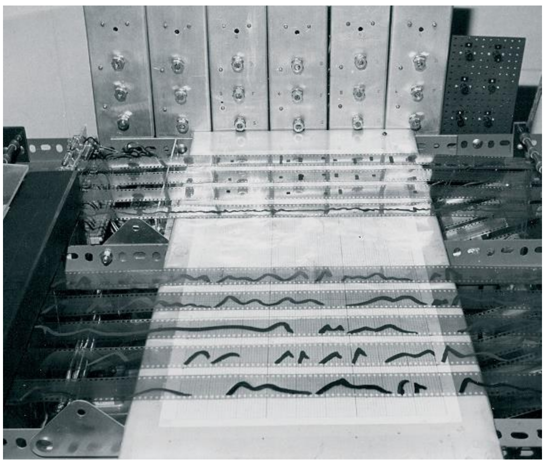

Figure 5. The Oramics Machine. Visible are curves used to describe the music.

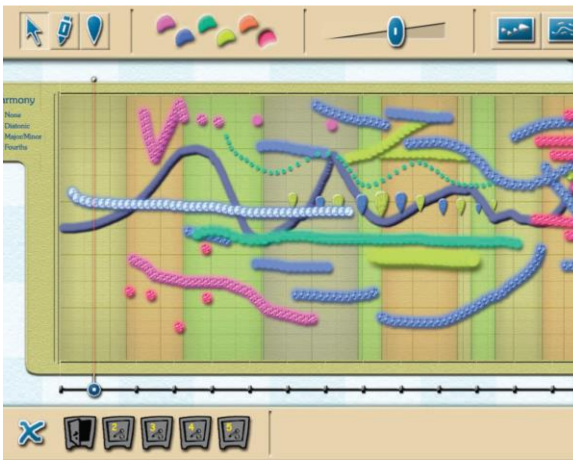

Figure 6. A screenshot of the free-formed curves in Hyperscore.

3 Approach

The Moodroom as developed by de Waag Institute makes use of a menu interface [3]. With this menu, the user adjusts the look and feeling of the Moodroom. The interface consists of multiple menus and submenus, all containing different buttons. The menu is navigated with the use of several gestures and by pointing with the hands.

For this thesis' project, various interaction techniques are compared in combination with music selection. To verify this, three interaction methods are discussed and compared.

For music to be as flexible and adjustable as needed for this project, various music techniques are discussed and compared.

3.1 Microsoft Kinect

The basis of the interaction techniques as used in the Moodroom and discussed in this thesis is the Microsoft Kinect.

Following Nintendo's Wii (2006) and PlayStation's Move (2010), Microsoft released their own wireless input device: they released the *Kinect* in the fall of 2010. The main difference between the Kinect and other input devices is the lack of a physical controller in the hands of the user. It is able to "see" the user and interpret its movements, or as Microsoft's slogan says: "You're the controller!"

The Kinect uses a standard colour camera, a depth sensor and a multi array microphone to provide full body 3D motion capture, facial and voice recognition [63].

For this project we used the Software Development Kit (or SDK) for Kinect for Windows. This SDK provides a basic Natural User Interface library for interaction with the Kinect.

3.2 Interaction techniques

A formal definition of interaction techniques is given by A.B. Tucker in his handbook on Computer Science: "An interaction technique is the fusion of input and output, consisting of all software and hardware elements, that provides a way for the user to accomplish a task." [64]

When a user wants to accomplish a task on the computer, he can make use of several elements, either hardware or software: the mouse, keyboard, voice or touch. Or more specialized controllers like the Kinect. Much research is going on in this field of human-computer interaction [65].

In this section, we will discuss several techniques that are relevant for this project.

3.2.1 Gesture-based menu

An established and often used interaction model is the menu. It is a structured overview of choices and is used in situations where the user has the ability to choose an item, where multiple items are present.

Operating Systems offer a lot of choices for settings, programs to run, operations to execute, etc. The first OS interfaces were text based, where the user could select an option by typing in a number, or use move a selection-marker up and down. When graphical user interfaces arrived, menus were transformed to more mouse-friendlier solutions. Countless different layouts of menus were launched, some better specialized for specific interfaces, others more general [66] [67].

For this project we looked into several menu structures, but since we are in a touchless, pointer-less environment, many of the existing layouts are not suitable and a simple layout is created.

3.2.2 Predefined gestures

To communicate with each other, humans developed languages and use their mouths to speak. However, using only speech can lead to miscommunication: the meaning of a word can differ on how it was pronounces, or how the speaker moved while uttering it. One wants to use non-verbal cues to assist in clarifying what is meant. The whole body plays a role in the communication.

Often small movements with the hands, arms or other parts are commonly used [68]. For instance waving when saying goodbye, pointing a thumb up in agreement or clapping hands to show joy. These gestures are often universal, although regional differences exist.

The use of gestures for interaction with computer systems is becoming more popular. With the arrival of touch, cameras and other input techniques, as well as faster computers, the ease of using a gesture is increasing. A gesture can be used to perform actions that would otherwise require multiple actions, e.g. moving the mouse, clicking a button, etc.

3.2.3 Undefined gestures

In the previous section, gestures are discussed. These are predefined, often small, moves of one (or in special cases more than one) limb. However, people tend to use their whole body when they communicate with others. Their mood has a strong influence on posture and movements [69], and this can be used as input.

Instead of defining some gestures, every motion the participant makes can be considered by the software. Small moves will have less impact than broad moves, a move of the hand will influence another parameter than a move of the legs, etc. Also the overall posture of the participant can be taken into account, and possible even his facial expressions, although the Kinect is not precise enough for the latter.

3.2.4 Comparing these techniques

Each technique discussed has its own strengths and weaknesses. We will shortly discuss each technique and compare them to the other techniques.

The menu has the ability to show which options are present in a (if well designed) clear overview. However, menus work better when used in combination with manual input, such as a mouse or keyboard, since it requires precise pointing and selecting of buttons. It is unviable for pointer-less interaction: it is hard to keep hands still when pointing at the screen from a distance, and, due to the lack of physical feedback, to know when a button is selected.

The gesture technique solves the pointing problem, but introduces others. The user has to memorize every gesture that is available in the program or at least the ones he should regularly use. Gesture detection with the Kinect is hard, since it depends on the sensor, which introduces noise. But apart from the noise, recognizing a gesture is less difficult than precise pointing.

Interpretation of movement solves most problems in the other techniques. However, this technique is hard to implement, since there is no established method. And as every person behaves different it requires a flexible implementation, without fixed rules for certain behaviours.

3.3 Music techniques

Since the music should reflect the mood of the user, there needs to be a great variation of music samples, or samples that can be adjusted to fit the required mood.

To be able to reflect ones mood, five parameters are defined on which the music is selected. See Table 2 for an overview of these parameters.

For this thesis' project these three approaches are considered:

1. Music that is already recorded, chosen from a database.
2. Music that is already created (not necessarily recorded), but can be adjusted.
3. Music that does is not recorded, but generated.

Using recorded music samples could be considered the simplest method. Find a large dataset of music samples, compute parameters for it and then use a query to find a song that is the closest match

PARAMETER	DESCRIPTION	LOW VALUE	HIGH VALUE
Attack	Time a note takes to reach its peak	Notes are dragged	Notes are sharp
Loudness	Volume	Calm	Loud
Noisiness	Harmonics, how much noise or random notes	Harmonious	Distorted
Pitch	Frequencies that appear	Low frequencies	High frequencies
Tempo	Speed	Slow	Fast

Table 2. Parameters and descriptions.

for that query. However, there are innumerable combinations of parameters, so to meet all parameters as close as possible, an enormous number of samples is needed.

Alternatively, instead of having a song for every combination, a song could be adjusted slightly, to fit the parameters more closely.

Finally we could just generate the music we need. This eliminates the need for a large database, where searching for a particular song could take time. Although it is hard to generate a complete song (music with a defined motive, repetitions, dynamics, etc.), it is possible to generate continuing music conform the selected parameters.

3.4 Combining interaction and music

The goal of the project is to determine which interaction technique is the best for which music production method. To validate this, an experiment where each combination is tested and rated should be held. This will result in the following, for now empty, matrix:

Music Gestures	Recorded	Adaptive	Generated
Menu			
Predefined			
Undefined			

The goal of this project is to fill this matrix, so it can be used to assign a rating to a combination of techniques. This can be useful when one wants to implement this.

To fill the matrix, first the different techniques need to be implemented. When they are implemented, an experiment will be performed to rate the various techniques. With this rating, the matrix can be filled and a conclusion can be drawn.

4 Implementation

In this chapter we will explain how the theoretical ideas from the previous chapter are implemented, and we will shortly discuss strengths and weaknesses of these implementations.

4.1 Interacting with the system

In this section it will be explained how the user can interact with the system and select the parameters that are used to play music. Both gesture-based menus and predefined gestures are implemented. The third technique, undefined gestures, is not implemented, since it is very complex.

4.1.1 Gesture-based menu

The following two approaches use a menu structure for users to interact with the system: one where the parameters are represented in movable blocks and one where the parameters are shown as sliders.

4.1.1.1 Approach 1: Blocks

Big blocks are placed on the right side of the screen, one for each parameter. The user can select a block, pick it up, move it around and release it. Each block can be moved individually. If the user picks up a block, he can move it to the centre of the screen, where it will turn green, to indicate that the block is selected. The user then releases the block, and the height of the block when released is an indication of the value for that particular parameter. As soon as the first block is selected, a list with songs that satisfy the condition is displayed. The more parameters are set, the shorter the list with songs.

Although the noise introduced from the Kinect sensor prevents a very precise placement of blocks, this does not appear to be a really significant issue since values are relative (e.g. a high value for tempo, instead of exactly setting a bpm), but it could distract from the final purpose.

There are two possible ways to use these blocks. Either the user has to place all blocks for a song to be selected, or the user places a few blocks and has to select a song manually.

With the current implementation of parameters this approach does not work well. The generated music needs all parameters to be able to generate music. This could be solved with the use of default values, but that limits the influence of the user, since certain parameters are pre-set: the idea of this approach is that instead of all parameters, a single or a few parameters can be set. Figure 7 shows a screenshot of this approach.

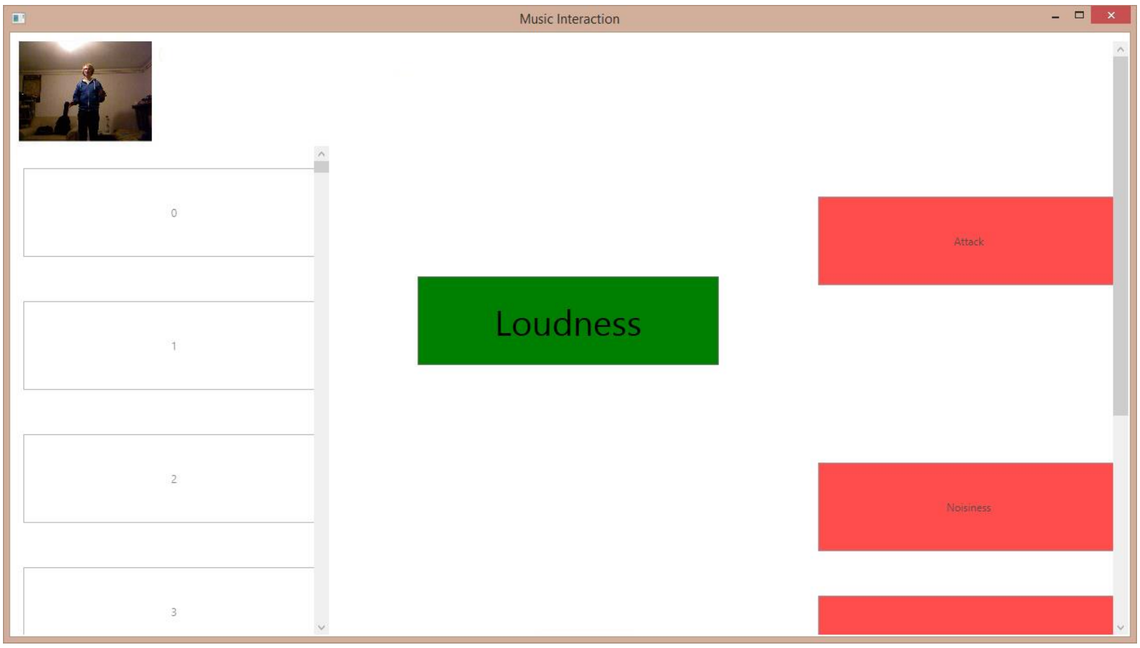

Figure 7. Screenshot of the Block layout. Loudness is dragged right now.

4.1.1.2 Approach 2: Sliders

For each parameter a slider is placed on the screen. This slider represents the value of that parameter for the current selected song. If the value for a slider is changed, a new song is located in the database.

Using sliders to represent values improves the visual feedback a lot. It is very clear to see what value for a parameter a song has.

This approach gesture friendly, through directly mapping the movement of the hands to locations on screen, using swipes to select a slider, and the distance between hands to set the value. This method deals better with noise interference. See Figure 8 for a screenshot of the implementation of this approach.

4.1.2 Predefined gestures

Menus are often used, but as touchscreens and other interaction inputs are getting more and more used, different approaches for easily handling input from these modalities are needed. Combining a Kinect and a menu works, but does not necessarily feel intuitive or easy to use since no physical feedback is provided. One has to hover hands over buttons (either large, thus cluttering the screen space, or small, thus hard to select) or point at the screen.

To improve this, an interaction structure is created that interprets the moves, instead of directly applying them, for instance special gestures for certain actions.

A gesture is a predefined movement that the user can perform to interact with the system, for instance a gesture like waving with the right hand that could lead to playing the selected song.

Multiple gestures are defined and linked to different actions. For basic gestures (waving, swiping and some two hand gestures like bringing two hands together) a gesture library is used [70]. More complicated gestures are developed for this thesis.

Most of the more complicated gestures are time-dependant. For instance, when detecting a tempo it is important to have a correct measurement of the speed of the hand.

Moving a hand up or down is used to determine the tempo. A timer is started that during two seconds will look for and locate the hand every 30 milliseconds. These positions are stored and are later used to calculate the tempo. The following pseudo code shows the method to do this:

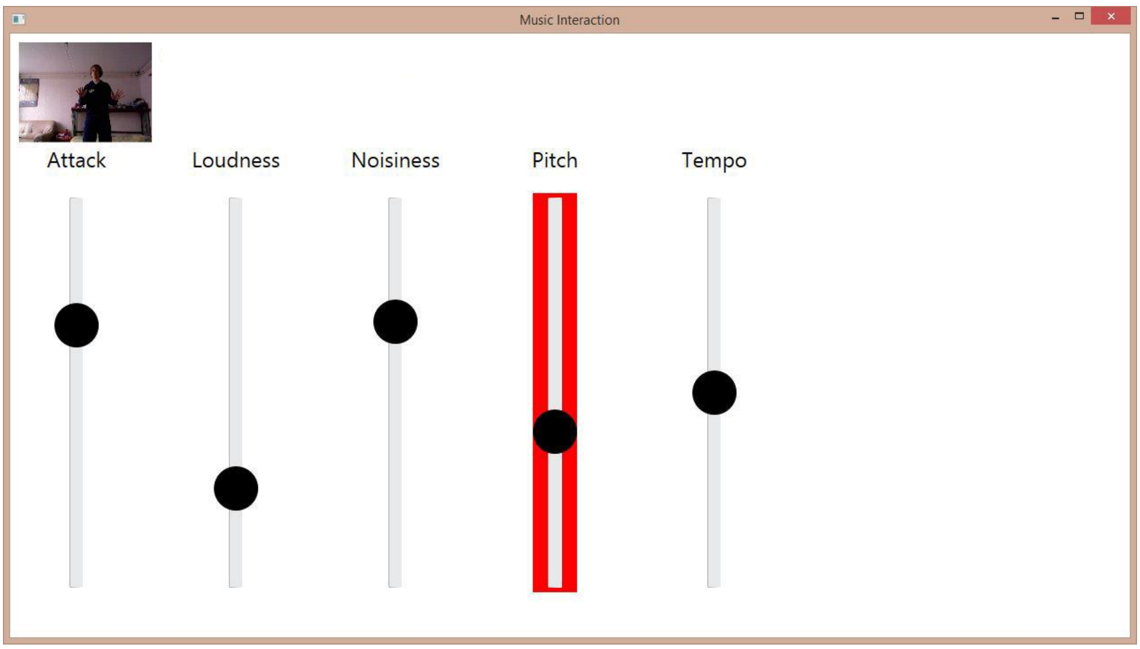

Figure 8. Screenshot of the Slider layout.

1. Compute the mean over all stored positions
2. Determine for each position whether it is above or under this mean
3. Count how many consecutive positions are at the same side of the mean
4. Compute the mean over all these lengths, either above or under
5. Compute the BPM with the following formula: $\frac{30\,ms}{mean\_length \cdot 2} * 60$

Besides tempo, the stored position of the hand can be used for multiple other parameters, although timing might not be important for all.

This list shows how each parameter can be derived from the body:

- **Attack.** If the movements of the arms are short and with sharp angles, the attack value is high. If the movements are smoother, the value is low.
- **Loudness.** The wider the movements are, the higher the value.
- **Noisiness.** If movements are repeated, this value is low. If each next movement is different, this value is high.
- **Pitch.** The distance between the knees and head.
- **Tempo.** How fast the movements of the arms are.

When using gestures there is no need to select items on the screen, this could prove to be more user-friendly. The user is free to stand anywhere in the range of the camera and can move freely around. He can make large movements, or make small gestures.

Most of the gestures are intuitive. However, the Kinect or library sometimes fails to recognize a gesture, which results in performing the same gesture over and over. That breaks the immersion of the system.

Figure 9 shows a screenshot of the implementation of this approach.

4.2 Playing music

The song selection is based on different parameters. These parameters are described in Table 2.

For this project we have used the same song set that was used in the research by Aljanaki [71]. This set contains 400 songs from the recording company *Magnatune* [72]. Four genres are used: classical, electronic, pop and rock. From each genre 100 songs are randomly selected. For all these songs the described parameters are computed and the results are stored in a database.

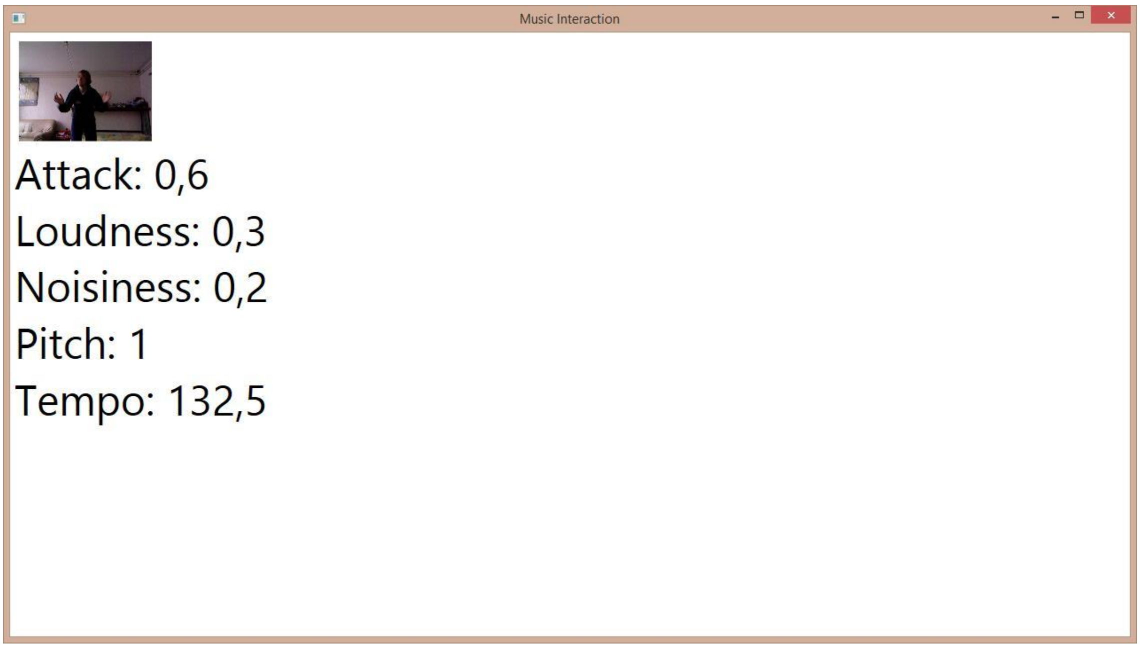

Figure 9. Screenshot of the Predefined Gesture layout.

With one of the interaction methods described in the last chapter, the user is able to assign values to these parameters. These values can then be used in different implementations. Two different approaches are tested: recorded music, where the parameters are used to locate a recorded song stored in a database and generated music, where the parameters are used to generate a song.

4.2.1 Collection based

In the previous section the different interaction methods are described, but each of these methods provides the same functionality. It uses input from the user to generate values for the parameters.

The values are stored in a parameter class. These parameters are used to locate a song in the database. Two different approaches have been implemented, however only the multiple parameter approach remains in the final version, as explained below.

4.2.1.1 Approach 1: A single parameter

A parameter is assigned a value between 0 and 1. When a song needs to be played, a parameter is selected by the user and the song located at $value\ for\ that\ parameter \cdot number\ of\ songs$ in the database is returned.

One of the advantages of this approach is that the user has the opportunity to find out what a certain parameter is. If he increases the value for tempo, the returned song will be faster. This will (probably) improve the understanding of the parameters, especially if the terms are unfamiliar to the user.

A disadvantage of this approach is that it is not possible, or at least very hard, for the user to select a song that is e.g. both fast and calm. This is because the songs are ordered on a single selected parameter, and by design it is not possible to select a second parameter.

This approach was originally implemented to be used with the block and slider approach. For use with the blocks approach, this method was used multiple times, once for each parameter, since the user selected parameters in a certain order. But for the slider and other approaches, this order is not available, thus this method lost its need.

4.2.1.2 Approach 2: Multiple parameters

To solve this problem, another approach is added: using all parameters at once. All parameters are assigned a value, either by default or by the user. The query is a single point in a 5 dimensional space. To select a song that fulfils all parameters, we have to find a song that is the closest to the query in the space. The following pseudo code shows how:

```
1. For each parameter, do:
      Compute for all songs the distance to the query point
2. X = 200
3. For each parameter, do:
      Select the X songs with the smallest distance
4. If the intersection of all sets is empty, increase X with 10 and repeat step 3
5. If the intersection contains one or more songs, select one
```

A good value for X has been set at 200 after some research. A lower value could be used, but that would limit the freedom of the parameters, since fewer songs are found. Experiments showed that users expect a different song after changing the parameters, and with only one song in the set, that is not possible.

This approach is more than multiplying the previous approach, since there is no explicit order in parameters. A user could update a parameter, but this does not make this parameter more important than the other parameters.

4.2.2 Generating music

Using recorded music has one problem: it is inflexible. If a user wants a slow, calm and noisy song it has to be in the database. If no such song is present in the set, it cannot be played.

This problem can be solved with the use of generated music. With SuperCollider (SC) it is possible to generate a song, with the use of the five described parameters. A script used to do this works as follows: it generates music in duple meter. Random melodic patterns consisting of four notes in a two octave span are repeated twice with different random rhythmic patterns. The patterns may include pauses, as long as they fit the metric structure. Each pattern is harmonized with chords or intervals. These are randomly chosen and depend on the musical scale: harmonic notes fitting the melody are chosen with a bigger probability than dissonant notes.

The five described parameters in Table 2 are implemented and can be adjusted. To change these parameters a link has been made between SC and our program. A pipe is used to send commands to SC, which responds accordingly.

Generating music based on the provided parameters provides flexibility that is not possible with recorded music. This gives a number of advantages: possibility to adjust certain parameters and keep others the same, no memory lost to an enormous music database and the fun of creating one's "own" music.

A big disadvantage however so far is the 'simplicity' of the music. It consists of randomly chosen notes, without a more songlike structure. The timbre of the instrument is also very basic. In this aspect the recorded music provides better structured music, with more different and familiar characteristics. This could be solved with a more complicated script.

4.3 Coding

The basis for this project is the Kinect, with the Kinect SDK. This SDK is written to be used with either C++ or C#, where C# is the preferred one. So for this project, we choose to use C#.

Figure 10 shows the Class Diagram for this project.

MusicInteractionWindow. The main class of the project. It initializes and controls the Kinect and it initializes all other interaction classes.

KinectSettings. A small class that is used to control two Kinect options: 1) the angle of the camera and 2) which image is used (either colour or depth, defined in the enumeration **ImageView**).

DummyInteractionClient. A small class that is needed to be able to detect whether the hands are open or closed.

IInteraction. An abstract class used to create an interface for the interaction between the musical part and the interaction part of the program. The **Menu** and **Gesture** classes are implementations of this class. **Hand** is a small class where both hands are defined, including whether the hands are open or closed. The Gesture class makes use of **SkeletonPoints**, a class that stores a Skeleton, where the location of each point can be retrieved. As the default C# is not precise, **RealTimeTimer** is a timer that is optimized to be precise. It is used to compute the tempo.

IMusic. Another abstract class. It is used to control the musical production. **Existing** and **Generated** are implementations for this class, where Existing defines a database for recorded songs and

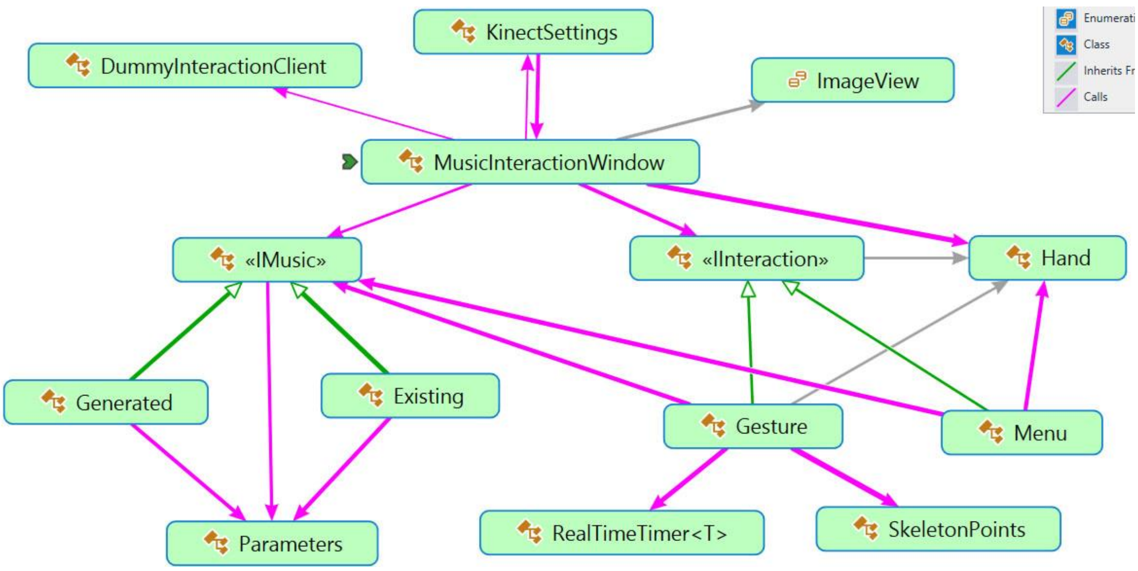

Figure 10. Class Diagram.

Generated connects between this program and SuperCollider. The whole musical part uses **Parameters**, a class where the mentioned parameters are stored. The used IInteraction class makes changes to these parameters, where the used IMusic class responds.

The use of abstract classes for both music and interaction provides freedom. If another technique is added, it simply needs to implement the mandatory methods from the abstract class, so no changes have to be made to the code in the rest of the program.

5 Experiment

An experiment was performed to demonstrate the capabilities of the solution, as well as to find out how well the solution works.

In section 3.4 a matrix is defined that visualizes how all techniques are combined. But it contains three techniques for interaction and three techniques for music, but since two interaction techniques have been implemented, the gesture based menu and the predefined gestures and two music techniques have been implemented, recorded music and generated music, the matrix is updated like this:

Music Gestures	Recorded	Generated
Menu		
Predefined		

This experiment is used to fill in the matrix with ratings for each combination:

1. Menu with recorded music
2. Menu with generated music
3. Predefined with recorded music
4. Predefined with generated music

Two different alternatives of the experiment are created to test against learning. Both alternatives contain the same exercises and questions, but the exercises are mirrored. At least 30 participants are needed for a valid statistical value. The test takes around 10 minutes.

The layout of the experiment is as follows:

1. The participant answers some basic questions about themselves: age, gender, education and whether they have experience with the Microsoft Kinect, Nintendo Wii or PlayStation Move.
2. The participants are given some time to get used to the software and to understand the parameters.
3. They are given several exercises, where they have to set the parameters as described in 30 seconds. The exercises are shown in Table 3. This is where the two alternatives differ. Half of the participants will start with exercise 1 and continue with 2, 3, etc. The second half will start with exercise 8 and continue with 7, 6, etc.
4. They are asked a few questions to verify their experiences so far.
5. Since the purpose of the thesis' project is to provide users with a tool to express their emotions, we will test this. They are asked to think of an emotion they felt yesterday and how they would express this in terms of the five given parameters. They can choose between one of the four combinations and then execute this.
6. After they have succeeded, they are asked some questions to research their experiences, whether they found they succeeded, and if it was easy and intuitive.
7. They are asked to rate their experiences on the music and interaction techniques on a Likert scale.

EXERCISE	ATTACK	LOUDNESS	NOISINESS	PITCH	TEMPO	INTERACTION COMBINATION
1	**High**	**High**	**High**	**High**	**High**	2: Menu with generated music
2	*Low*	*Low*	**High**	**High**	*Low*	
3	**High**	**High**	*Low*	**High**	**High**	3: Gesture with recorded music
4	*Low*	*Low*	**High**	**High**	*Low*	
5	*Low*	**High**	**High**	*Low*	*Low*	4: Gesture with generated music
6	**High**	*Low*	*Low*	**High**	**High**	
7	*Low*	**High**	**High**	*Low*	**High**	1: Menu with recorded music
8	*Low*	*Low*	*Low*	*Low*	*Low*	

Table 3. Overview of the exercises. Bold and italics are used for clearness.

From these questions, a quality degree will be computed like this: the number of positive reactions divided by the total number of reactions. This will indicate whether people like the software. Their answers will provide more details on how their experiences were.

5.1 Results

The experiment was done with the help of 30 participants. An overview of some statistics about the participants is shown in Table 4. The participants are grouped by experience with either the Kinect, Wii or Move, 0 being no experience at all, 4 being a lot of experience. The education is either university (U), university of professional education (HBO), Vocational Education (MBO) or a combination of U and HBO. However, no correlation has been found between these characteristics and their preferred technique.

The participants were equally divided over the two alternatives of the experiment, however, no differences that can be explained due to these versions are found.

All of the participants preferred the menu over the gestures, although many of them explained that the reason for that is lack of training. There are too many gestures to be handled at once, which makes them hard to control. But giving more time, and perhaps without a specific task or exercise, it does

EXPERIENCE		GENDER		AGE GROUP		EDUCATION	
0	*8*	Female	*5*	18-20	*2*	U	*5*
		Male	*3*	21-23	*3*	HBO	*2*
				24-28	*3*	U/HBO	*0*
						MBO	*1*
1	*2*	Female	*2*	18-20	*0*	U	*1*
		Male	*0*	21-23	*2*	HBO	*1*
				24-28	*0*	U/HBO	*0*
						MBO	*0*
2	*12*	Female	*8*	18-20	*4*	U	*6*
		Male	*4*	21-23	*7*	HBO	*4*
				24-28	*1*	U/HBO	*2*
						MBO	*0*
3	*6*	Female	*1*	18-20	*3*	U	*3*
		Male	*5*	21-23	*1*	HBO	*3*
				24-28	*2*	U/HBO	*0*
						MBO	*0*
4	*2*	Female	*0*	18-20	*1*	U	*1*
		Male	*2*	21-23	*1*	HBO	*1*
				24-28	*0*	U/HBO	*0*
						MBO	*0*

Table 4. Information about the participants. Experience (0 none to 4 a lot), Gender, Age and Education. The number of participants in a group is shown in italics.

VERSION		INTERACTION COMBINATION		RECORDED	GENERATED	MENU	GESTURES
A	*15*	Menu, recorded	*9*	3.6	2.0	4.2	2.3
		Menu, generated	*3*	3.3	3.0	4.0	3.0
		Gestures, recorded	*3*	4.3	2.3	3.7	4.0
B	*15*	Menu, recorded	*8*	3.6	2.4	4.1	2.6
		Menu, generated	*3*	4.3	2.3	3.7	1.7
		Gestures, recorded	*4*	3.8	2.8	3.8	3.5

Table 5. Rating per interaction technique. 1 is very negative, 5 very positive. The size of the groups in shown in italics.

have potential to be an intuitive and fun way of interacting, even better than the menu, where one has to divide an emotion in multiple parameters.

Table 5 shows how the participants rated the various techniques. Recorded music is rated higher than generated music: with an average of 3.7 against 2.4. The menu is rated higher than the gestures, with an average of 4 against 2.7. Also most participants, 17, chose to use a combination of these two techniques. Six participants used the menu in combination with the generated music, and seven participants chose the gestures, in combinations with recorded music.

The choice for the menu is often explained: it was far more easy to use than the gestures. Some participants did prefer the gestures, since it required less attention to specifically setting parameters, resulting in a more intuitive experience. However, the current implementation was not easy to use: some gestures were too much alike, they required a lot of practise and all parameters had to be set at the same time.

Although participants often complained about not hearing any clear difference when changing parameters with recorded music, they still preferred it over generated music. The reason for this is that the generated music is difficult to listen to, since it lacks repeated patterns and multiple layers of instruments: it is less pleasing to the ears than recorded music. So although it would lead to a more explicit and precise change in parameters, it was barely chosen.

All participants were asked to express an emotion. In Figure 11 these emotions are shown, as well as the values of the parameters used to express them. Four emotions were chosen multiple times: enthusiasm two times, sadness three times, angry six times and happy eleven times. Although many participants chose the same emotion, almost none of them set the parameters at the same values as shown in Figure 11. None of the coloured bars are the same, with one exception: enthusiasm, every value at 100.

Table 6 provides an overview per emotion. For the emotions that have been chosen multiple times the mean and the standard deviation are calculated. The standard deviation shows how much the participants disagreed over a parameter. For instance for happy, one participant set attack to 10, others to 100.

Although everyone did succeed, some had trouble finding the settings they thought would express the emotion. This was either due to difficulty with finding the right values for the parameters, or with the fact that songs would continuously change while changing parameters. Some emotions were, presumably, easier than others. For instance, most people thought happiness an easy emotion, hence

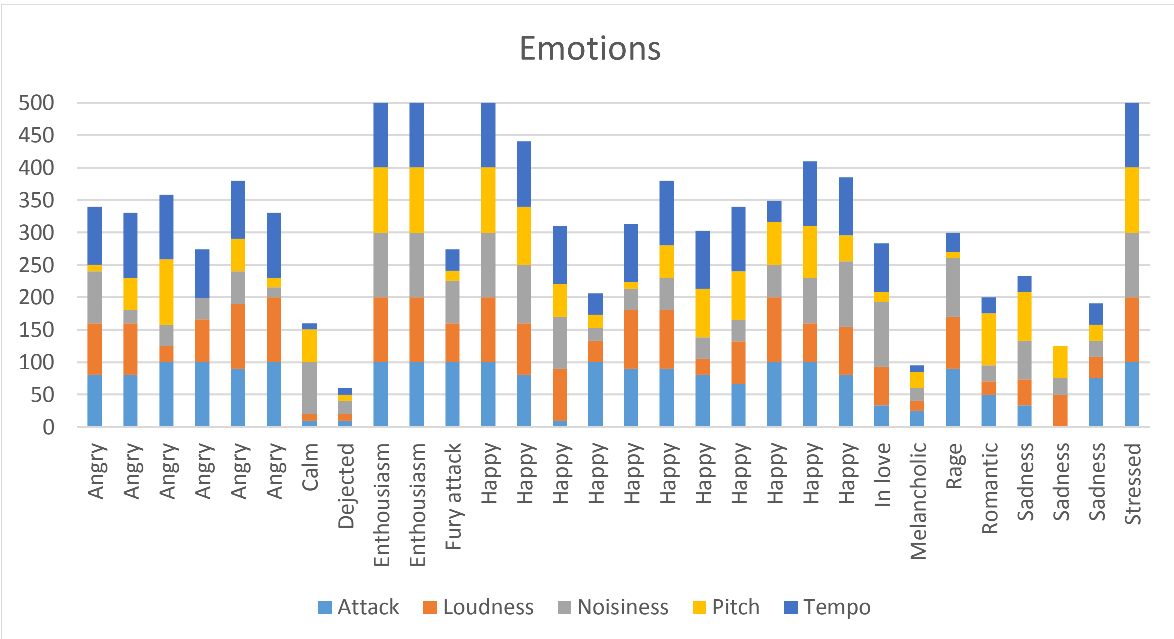

Figure 11. Values per emotion.

the eleven variants. Angry and sadness were also deemed easy. Yet nobody did really agree on how to set the parameters. This might be due to different genres in music, and the constant change of tracks while setting parameters, or to humans experiencing emotions quite different.

Participants were also asked whether they would use a system like this to express emotions. 25 participants think it is possible, although some assume that for more difficult emotions it would be hard to find the right values. Others think it would depend per person, as some people like to use movements to express themselves, while others are more subdued.

The other five participants think it is impossible, or at least with the implementation as it currently is. This is due to the limitations of the software (noise, gestures or songs) or the overall idea of using music or gestures as a mean to express an emotion.

EMOTION		**ATTACK**		**LOUDNESS**		**NOISINESS**		**PITCH**		**TEMPO**	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Angry	*6*	91.7	9.0	75.2	25.4	38.5	21.6	37.5	33.9	92.5	9.0
Calm	*1*	10.0	0.0	10.0	0.0	80.0	0.0	50.0	0.0	10.0	0.0
Dejected	*1*	10.0	0.0	10.0	0.0	20.0	0.0	10.0	0.0	10.0	0.0
Enthusiasm	*2*	100.0	0.0	100.0	0.0	100.0	0.0	100.0	0.0	100.0	0.0
Fury attack	*1*	100.0	0.0	60.0	0.0	66.0	0.0	15.0	0.0	33.0	0.0
Happy	*11*	81.5	25.0	72.6	23.9	59.9	27.9	59.6	27.1	84.2	24.5
In love	*1*	33.0	0.0	60.0	0.0	100.0	0.0	15.0	0.0	75.0	0.0
Melancholic	*1*	25.0	0.0	15.0	0.0	20.0	0.0	25.0	0.0	10.0	0.0
Rage	*1*	90.0	0.0	80.0	0.0	90.0	0.0	10.0	0.0	30.0	0.0
Romantic	*1*	50.0	0.0	20.0	0.0	25.0	0.0	80.0	0.0	25.0	0.0
Sadness	*3*	36.0	30.7	41.0	7.0	36.7	16.5	50.0	20.4	19.3	14.1
Stressed	*1*	100.0	0.0	100.0	0.0	100.0	0.0	100.0	0.0	100.0	0.0

Table 6. Settings for each parameter, per emotion. μ is the mean value assigned, σ the standard deviation. The higher a mean is, the more important it is for that emotion. However, a higher standard deviation means that participants chose very different values. The number of participants expressing a certain emotion is shown in italics.

6 Conclusion

The goal of this project was to compare different interaction techniques with different techniques for making music. This was done by implementing some of these techniques. For the touchless interaction the Microsoft Kinect was used, for music both a database with recorded music and a music generating script in SuperCollider. In the related work section it is stated that the interface should be easy, while providing enough freedom to express ones emotions. To verify this, a small experiment was created to find out what participants like, and if they are able to work with the program. 30 participants were asked to practise and use it, and were then given the task of expressing an emotion with it. Although everyone succeeded in expressing an emotion, five participants think this method is not suited for expressing emotions. 76.7% of the participants preferred the gesture-based menu, over the predefined gestures. The gestures were judged intuitive, although having to perform them at the same time was hard. The menu received an overall rating of 4 out of 5, the gesture 2.7.

80% of the participants preferred the recorded music over the generated music, since the generated music did not really sound like music, but more like a collection of random notes.

With these ratings it is possible to fill the matrix introduced in section 3.4 (and updated in chapter 5). These ratings are based on how participants rated the several techniques, and which were preferred by them. 5 is the most preferred, 1 is the least.

Music Gestures	Recorded	Generated
Menu	5	4
Predefined	3	1

From the matrix it can be concluded that the combination of a gesture-based menu and recorded music is rated the best. So for a Moodroom-like setting, it would be best to look into a well-designed menu, in combination with recorded music, since that is easy to use and understand, while providing enough controls for freedom and playing around.

In the conclusion of the related work (see section 2.6) the goal of the project was stated as:

> *"(…) to implement an environment that has an easy to use interface that provides enough freedom for the user to reach his or her goal: create or find music that expresses their emotions."*

The gesture-based menu interface was preferred, and easy to use. The predefined gesture-based interface was not as easy, yet provided more freedom. The music that was found with the system was expressing their emotions, so the goal has been reached at least to a certain extent.

6.1 Future work

The experiment showed that participants were enthusiastic about the possibilities of 'dancing' in front of a camera, with music generated from their movements. However, only some predefined gestures were considered by the software. Undefined gestures would lead to more freedom, and probably in a more immersive experience. A recent example of this is the Nagual Dance [73]. This project uses a (invisible) grid on the floor, where the location of the user in that grid defines which sound is adjusted: drums and bass, melody, or combinations. The intensity and curves of the gestures is the basis for the music.

However, even more freedom should be possible. No predefined gestures, or grids, or anything. Just a user and a piece of software and from the movements a piece of music arises. Based on the speed of dancing, the way of moving, maybe even facial expressions. Using this, the participant of a Moodroom would not be confronted with parameters or difficult gestures, but merely the way he moves indicates what emotions he is experiencing.

For a more immersive, and probably richer, experience, the script used for generating music with SuperCollider could be expanded. When more structure and more different voices are used, it will provide a more appealing song, which could actually improve the expressivity of the system.

Finally, more research could be done in adding more senses besides the visual and auditory ones. Adding smell, taste and touch might improve the overall experience, since humans use all their senses to experience the world. So providing more ways to express them might increase the effect and freedom to do so.

7 Bibliography

1 Meijer S, Postma S, Schoemaker C, Zwaanswijk M. Nationaal Kompas Volksgezondheid. [Internet]. 2014 [cited 2014 July 25]. Available from: http://www.nationaalkompas.nl/gezondheid-en-ziekte/functioneren-en-kwaliteit-van-leven/psychisch-functioneren/psychisch-functioneren-jeugd/hoe-vaak-komen-psychische-problemen-bij-jeugd-voor/.

2 About. [Internet]. [cited 2015 February]. Available from: http://www.ieata.org/about.html.

3 Buijs L. Spelen met emoties. Utrecht University; 2013.

4 Ball P. The music instinct: How music works and why we can't do without it. London: Bodley Head; 2010.

5 Holland S, Wilkie K, Mulholland P, Seago A, editors. Music and Human-Computer. London: Springer; 2013.

6 Carroll JM, Mack RL, Kellogg WA. Interface metaphors and user interface design. Vol Handbook of human-computer interaction. New York: Elsevier; 1988.

7 Rowe R. Machine musicianship. Cambridge, Massachusetts: MIT Press; 2001.

8 Tanaka A. Musical performance practice on sensor-based instruments. In Trends in. Centre Pompidou: IRCAM; 2000.

9 Karlstrom P. Existential phenomenology and design - Why "ready-to-hand" is not enough. [Internet]. 2007 [cited 2015 March 23]. Available from: http://petter.blogs.dsv.su.se/files/2011/02/Existential_phenomenolgyand-HCI.pdf.

10 Gentner D, Nielsen J. The Anti-Mac interface. Communications of the ACM. 1996;39(8):70–82.

11 Nielsen J. Progressive disclosure. [Internet]. 2006 [cited 2015 March 23]. Available from: http://www.useit.com/alertbox/progressive-disclosure.html.

12 Smith M, Burke C, Draper S, Potter C. The energy cost of rock drumming: a case study. In: European College of Sport Science (ECSS) 13th annual Congress; 2008; Estoril, Portugal.

13 Bills AG. The curve of learning. New York: Longmans, Green and Co; 1934.

14 Ritter FE, Schooler LJ. The learning curve. Amsterdam: Pergamon; 2002. p. 8602–8605.

15 Swift B. Chasing a feeling: Experience in computer supported jamming. In: Holland S, Wilkie K, Mulholland P, Seago A. Music and human-computer interaction. London: Springer. p. 85–100.

16 Wagner R. The virtuoso and the artist. 1830.

17 Wallis I, Ingalls T, Campana E, Vuong C. Amateur musicians, long-term engagement, and HCI. In: Holland S, Wilkie K, Mulholland P, Seago A. Music and human-computer interaction. London: Springer; 2013. p. 49–66.

18 Levitin D. This is your brain on music: The science of a human obsession. East Rutherford: Penguin Putnam; 2006.

19 Stuart C. The object of performance: Aural performativity in contemporary laptop music. Contemporary Music Review. 2003;22(4):59–65.

20 Paine G. Gesture and morphology in laptop music performance. In: Dean R. The Oxford handbook of computer music. New York: Oxford University Press; 2009. p. 214–232.

21 Krefeld V, Waisvisz M. The hand in the web: An interview with Michel Waisvisz. Computer Music Journal. 1990;14(2):28-33.

22 Brewster SA. Overcoming the lack of screen space on mobile computers. Personal and. 2002;6(3):188–205.

23 Sanderson P. The multimodal world of medical monitoring displays. Applied Ergonomics. 2006;37(4):501–512.

24 Gaver WW. Auditory icons: Using sound in computer interfaces. Human Computer Interaction. 1986;2(2):167–177.

25 Milliman RE. Using background music to affect the behavior of supermarket shoppers. The Journal of Marketing. 1982;46(3):86–91.

26 Areni CS,&KD. The influence of background music on shopping behavior: Classical versus Top-Forty music in a wine store. Advances in Consumer Research. 1993;20(1):336–340.

27 North AC, Hargreaves DJ, McKendrick J. The influence of in-store music on wine selections. Journal of Applied Psychology. 1999;84(2):271–276.

28 Caldwell C, Hibbert SA. The influence of music tempo and musical preference on restaurant patrons' behavior. Psychology and Marketing. 2002;19(11):895–917.

29 Milliman RE. The influence of background music on the behavior of restaurant patrons. Journal of Consumer Research. 1986;13(2):286–289.

30 Griffiths M, Parke J. The psychology of music in gambling environments: An observational research note. Journal of Gambling Issues. 2005;13.

31 Bishop DT, Karageorghis CI, Georgios L. A grounded theory of young tennis players' use of music to manipulate emotional state. Journal of Sport & Exercise Psychology. 2007;29:584–607.

32 Waterhouse J, Hudson P, Edwards B. Effects of music tempo upon submaximal cycling performance. Scandinavian Journal of Medicine & Science in Sports. 2010;20(4):662–669.

33 Simpson SD, Karageorghis CI. The effects of synchronous music on 400-m sprint performance. Journal of Sports Sciences. 2006;24(10):1095–1102.

34 Thompson WF, Schellenberg EG, Letnic AK. Fast and loud background music disrupts reading comprehension. Psychology of Music. 2011.

35 Livingstone SR, Mühlberger R, Brown AR, Loch A. Controlling musical emotionality: An affective computational architecture for influencing musical emotion. Digital Creativity. 2007;18(1):43–53.

36 Sanders T, Cairns P. Time perception, immersion and music in videogames. Paper presented at BCS HCI, University of Abertay, Dundee. 2010 6–10 Sept.

37 Hébert S, Béland R, Dionne-Fournelle O, Crête M, Lupien SJ. Physiological stress response to video-game playing: The contribution of built-in music. Life Sciences. 2005;76(20):2371–2380.

38 Cassidy GG, MacDonald RAR. The effects of music on time perception and performance of a driving game. Scandinavian Journal of Psychology. 2010;51(6):455–464.

39 Jensen MB. The influence of jazz and dirge music upon speed and accuracy of typing. Journal of Educational Psychology. 1931;22(6):458–462.

40 Dixon L, Trigg R, Griffiths M. An empirical investigation of music and gambling behaviour. International Gambling Studies. 2007;7(3):315–326.

41 Richards D, Fassbender E, Bilgin A, Thompson WF. An investigation of the role of background music in IVWs for learning. Research in Learning Technology. 2008;16(3):231–244.

42 Bramwell-Dicks A. Towards AME: Exploring the use of ambient music within an email interface. Dissertation, University of York, York. 2010.

43 Zentner M, Grandjean D, Scherer K. Emotions evoked by the sound of music: Characterization, classification, and measurement. Emotion. 2008;8:494–521.

44 Vuoskoski JK,ET. Measuring music-induced emotion: A comparison of emotion models, personality biases, and intensity of experiences. Musicae Scientiae. 2011;15(2):159-173.

45 McDermott J, O'Neill M, Griffith NJL. EC control of sound synthesis. Evolutionary Computation Journal. 2010;18(2):277-303.

46 McDermott J, Griffith NJL, O'Neill M. Evolutionary computation applied to sound synthesis. In: Romero J, Machado P, editors. The art of artificial evolution: A handbook on evolutionary art and music. Berlin/Heidelberg: Springer; 2007. p. 81–101.

47 Collins N. The analysis of generative music programs. Organised Sound. 2008;13(3):237–248.

48 Biles JA. GenJam: A genetic algorithm for generating jazz solos. In: Proceedings of the 1994 international computer music conference. Danish Institute of Electroacoustic Music, Denmark: International Computer Music Association; 1994. p. 131–137.

49 Fitts PM. The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology. 1954;47(6):381–391.

50 MacKenzie IS. Fitts' law as a research and design tool in human–computer interaction. Human–Computer Interaction. 1992;7:91–139.

51 Takagi H. Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. Proceedings of the IEEE. 2001;89(9):1275–1296.

52 Miller GA. The magical number seven, plus or minus two: Some limits on our capacity for processing information. The Psychological Review. 1956;63:81-97.

53 Dahlstedt P. On the role of temporary storage in interactive evolution. In: M. Giacobini ea, editor. Applications of evolutionary computing: EvoWorkshops 2009, no. 5484 in LNCS. New York: Springer; 2009. p. 478–487.

54 Baluja S, Pomerleau D, Jochem T. Towards automated artificial evolution for computer-generated images. Connection Science. 1994;6(2-3):325-354.

55 McDermott J, Griffith NJL, O'Neill M. Evolutionary GUIs for sound synthesis. In: Giacobini M, editor. Applications of evolutionary computing (vol. 4448 in LNCS). New York: Springer; 2007. p. 547–556.

56 Hoover AK, Stanley KO. Exploiting functional relationships in musical composition. Connection Science. 2009;21(2):227-251.

57 Paradiso JA. The brain opera technology: New instruments and gestural sensors for musical interaction and performance. Journal of New Music Research. 1999;28(2):130-149.

58 Jordà S, Geiger G, Alonso M, Kaltenbrunner M. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In: Proceedings of the 1st international conference on Tangible and embedded interaction; 2007; ACM. p. 139-146.

59 'Oramics' Daphne Oram. UK, 1959. [Internet]. [cited 2014 December]. Available from: http://120years.net/oramicsdaphne-oramuk1959-2/.

60 Farbood M. Hyperscore: a new approach to interactive, computer-generated music. Massachusetts Institute of Technology; 2001.

61 Comparison of audio synthesis environments. [Internet]. [cited 2014 September]. Available from: http://en.wikipedia.org/wiki/Comparison_of_audio_synthesis_environments.

62 McCartney J. SuperCollider. [Internet]. [cited 2014 December]. Available from: http://supercollider.github.io/.

63 Microsoft Corporation: Kinect for Windows Programming Guide version 1.6. [Internet]. [cited 2015 January]. Available from: http://msdn.microsoft.com/en-us/library/hh855348.aspx.

64 Tucker AB. Computer Science Handbook, Second Edition. Chapman & Hall/CRC; 2004.

65 Hewett TT, Baecker R, Card S, Carey T, Gasen J, Mantei M, Perlman G, Strong G, Verplank W. ACM SIGCHI Curricula for Human-Computer Interaction. [Internet]. 1996 [cited 2015 January]. Available from: http://old.sigchi.org/cdg/cdg2.html#2_1.

66 Menus (Windows). [Internet]. [cited 2015 January]. Available from: http://msdn.microsoft.com/en-us/library/dn742392.aspx.

67 Bailly G,LE,NL. MenUA: A Design Space of Menu Techniques. [Internet]. 2009 [cited 2015 January]. Available from: http://www.gillesbailly.fr/menua/.

68 Kendon A. Gesture: Visible Action as Utterance. Cambridge: Cambridge University Press; 2004.

69 Chartrand TL, Bargh JA. The unbearable automaticity of being. American Psychologist. 1999 July;54(3):462-479.

70 Pterneas. LightBuzz/Vitruvius. [Internet]. 8 [cited 25 July 2014]. Available from: https://github.com/lightbuzz/vitruvius.

71 Aljanaki A, Wiering F, Veltkamp RC. Collecting annotations for induced musical emotion via online game with a purpose. University Utrecht; 2014. ISSN 0924-3275.

72 Magnatune. [Internet]. [cited 2014 July]. Available from: http://magnatune.com/.

73 Nagual Dance. [Internet]. [cited 2014 September]. Available from: http://www.nagualsounds.com/.

74 Edworthy J, Waring H. The effects of music tempo and loudness level on treadmill exercise. Ergonomics. 2006;49(15):1597–1610.

75 Spenwyn J, Barrett D, Griffiths M. The role of light and music in gambling behaviour: An empirical pilot study. International Journal of Mental Health and Addiction. 2010;8(1):107–118.

76 Bramwell-Dicks A, Petrie H, Edwards ADN, Power C. Affecting user behaviour and experience with music: A research agenda. BCS HCI workshop “When words fail: What can music interaction tell us about HCI?” 8 May 2011. Northumbria University, Newcastle Upon Tyne. 2011.

77 McDermott J, O’Reilly UM. An executable graph representation for evolutionary generative music. Dublin: In GECCO ’11; 2011.