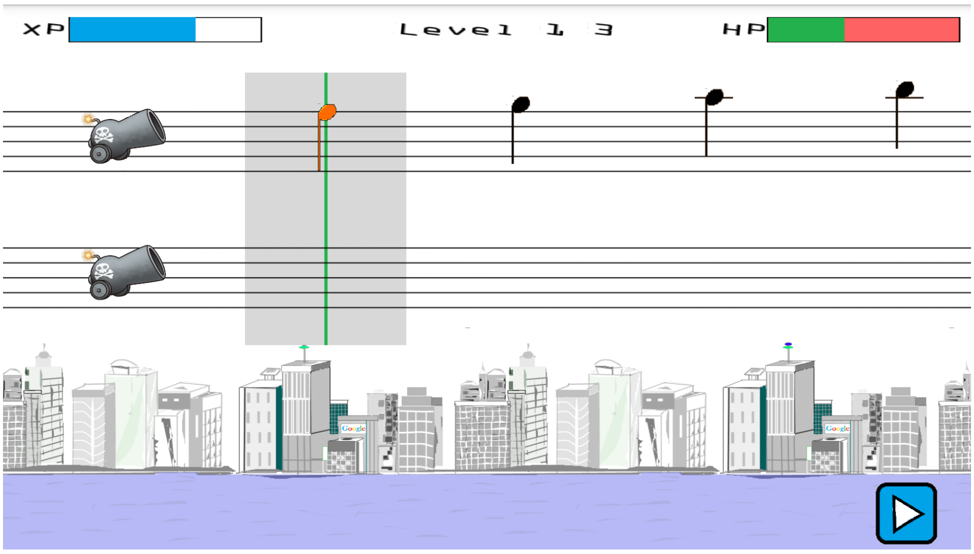
Multi-Pitch Sound Analysis on Low Quality Microphones: An Approach to a Game-based Android Application for the Piano



Jacqueline Staub

Bachelor Thesis
March 2014

Supervisor:
Prof. Dr. Otmar Hilliges

ETH zürich

Advanced Interactive Technologies

Abstract

Mobile phones are spread around the world. The number of smart phone users increases massively. Statistical projections predict a number of 1.75 billion smart phones possessed by seven billion people on earth by the end of the year 2014, a way to reach many people is to program applications for the smart phone. A common way to interact to smart phones is to touch the screen. We tested another way of interaction, using environmental sound analysis. We developed a music game for Android phones to support students during the process of learning to play the piano. We assume the user to know the notation of notes and to play an instrument that complies with the theory of harmonics. Our application solves the frequency detection problem using the short time fourier transform. In a controlled comparison two prototypes based on differing game ideas were compared. A final prototype was implemented with the favoured functional elements of the controlled comparison experiment. In a cognitive walkthrough we tested the final prototype for usability issues. The results pointed out some issues with the memorability of the game concept. We achieved an average score of 71% for the memorability. High values were measured for the user satisfaction with an average value of 81%. In another experiment we measured the volume of noise in three environments (a silent room, outside, and close to a street). A mechanism to filter noisy signals was implemented based on the measurements. The application works best (with a hit rate of 100%) for instruments with a range of notes between D3 and E7. We reach a hit rate of 70% for the piano, whose pitch range starts at A1 and ends at G7. The main difference between our application and related research topics is, that our application works o low quality microphones and runs only on the limited resources provided by the phone. Our results demonstrate that the hardware of today's mobile phones is already fast enough to perform time-consuming operations. Even fast enough to give the user the impression of an instantaneous feed-back.

Acknowledgment

I would like to give thanks to Prof. Dr. Otmar Hilliges for supervising this project. Thanks to him I had the opportunity to get to know many fields in computer science that were unfamiliar to me half a year ago. The last semester was a very challenging and rewarding one. His profound knowledge in many areas was very helpful to me.

Moreover I would like to express my appreciation and deep gratitude to Christine Ryer-Bringold, my former music teacher. She supported me generously with many hours of supportive advices and conversations about education and music.

Someone else who deserves my deep gratitude is my mother. The innumerable sacrifices she made to support me whenever she could are highly appreciated and I cannot say how thankful I am for everything she did.

Contents

List of Figures ix

List of Tables xi

1. Introduction 1

1.1. Problem Statement . . . 1
1.2. Goals . . . 2
1.3. Structure of this Document . . . 3
1.4. Conventions . . . 3

2. Background 5

2.1. Music . . . 5
2.1.1. Physical Tones . . . 5
2.1.2. Partials . . . 7
2.1.3. Instrumental Differences . . . 8
2.1.4. Noise . . . 9
2.2. Android . . . 10
2.2.1. Android Design Principles . . . 10
2.3. Education Theory . . . 12
2.3.1. Key Components in the Game Idea . . . 13
2.4. User Study and Evaluation . . . 13
2.4.1. Determination of System Values . . . 13
2.4.2. Choice on Functionality of the System . . . 14
2.4.3. Usability Testing on the final Prototype . . . 14

3. Related Work 17

3.1. Zero Crossings Algorithm 17
3.2. Related Research 19
3.2.1. Decay Characteristics of Piano Tones 19
3.2.2. Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals 20
3.2.3. The Design with Intent Method: A design tool for influencing user behaviour 20

4. Technical Aspects **23**
4.1. Hardware 23
4.2. Requirements 23
4.3. Common Setup 24
4.4. Involved Libraries 24

5. Implementation **27**
5.1. Frequency Analysis 27
5.2. Software Components 29
5.3. Graphics 29
5.4. Sound 30
5.4.1. Notes 30
5.4.2. Songs 30
5.4.3. Timing 31
5.4.4. MIDI Playback 31
5.4.5. Noise Detection 32
5.5. Game Controller 33
5.5.1. Player and Score 33
5.6. User Interface 33

6. Evaluation **37**
6.1. Methodology 37
6.2. Experiment: Comparison of two Prototypes 38
6.2.1. Setup 38
6.2.2. Results 39
6.3. Experiment: Cognitive Walkthrough 41
6.3.1. Setup 41
6.3.2. Results 42
6.4. Long term study 44
6.4.1. Setup 44
6.5. Limitations 44
6.5.1. Problems at the bottom of the range 45
6.5.2. Problems at the top of the range 46
6.5.3. Problems with different instruments 46
6.5.4. Assumption of loudest fundamental wave 47
6.5.5. Dependency on environmental factors 47

7. Conclusion and Future Work **49**
7.1. Future Work 50

7.1.1. Instructions 50
7.1.2. Feed-back after finishing the game 50
7.1.3. Support for transposed notation and multiple instruments 50
7.1.4. Interface to create levels 50

A. Demographic Questionnaire **51**

B. Post-study Questionnaire **55**

C. Tables for Notes **59**
C.1. Notes on the Piano, the complete Table 59

Bibliography **63**

List of Figures

2.1. Duration . . . 6
2.2. Scales . . . 7
2.3. Partials . . . 8
2.4. Transposition . . . 9

3.1. Zero crossings algorithm . . . 18
3.2. Issues in the zero crossings algorithm . . . 19

4.1. Setup . . . 24

5.1. Mirroring and partials for A4 . . . 28
5.2. Software Components . . . 29
5.3. Notes . . . 30
5.4. Songs . . . 31
5.5. Timing . . . 31
5.6. User Interface . . . 35

6.1. Prototype A . . . 38
6.2. Prototype B . . . 39
6.3. Controlled Comparison Experiment . . . 40
6.4. Results for the cognitive walkthrough experiment . . . 43
6.5. Problems at the bottom of the range . . . 46
6.6. Problems at the top of the range . . . 46
6.7. Problems in the model . . . 47

List of Tables

5.1. Noise measurements in three environments 32

6.1. Measurements hit rate 45

C.1. Notes, Frequency, Wavelength, Semitone number 59

1

Introduction

During the past ten years mobile phones with advanced computational capacity (**smart phones**) emerged. One noticeable difference between them and their predecessors (**feature phones**) is the possibility of a multitude of different **interaction styles**. While traditional mobile phones offered interaction using hardware keys only, their successors offer a larger scope of possible interaction styles. Very often the interaction between user and phone bases on skin contact on the screen. In the year 2011 Apple presented the first mobile phone that could be controlled by voice. (Even though the idea of a system using speech recognition already existed a couple of decades before [Pinola Mar. 2014]) Speech control is a very natural way to interact with a system, since it is a common ability of the human to speak. On the other hand such a system poses many difficulties. Speech disorder, noise, non native speakers, and different accents are some specific issues that have to be handled in a systems controlled by voice. The project presented in this paper does not deal with voice recognition but a related interaction style, namely interaction with sound. It is an approach to an educational music game developed for Android smart phones, to support children or adults during the process of learning how to play an instrument. In comparison to most of the other music games available in the Play Store that are controlled by touching an *illustration* of an instrument on the screen, this system is controlled by a *real instrument*.

1.1. Problem Statement

Motivation is a key component to learn effectively. Experiments discovered the fact that the attention of a person listening to a presentation drops massively after a time period of about ten minutes [Medina 2009]. Lets assume we were able to increase that time period by keeping the person more motivated and therefore attentive. We could expect better results as an effect of higher motivated learning. The educational game presented in this document tries to motivate its user using attractive animations, moving objects, and audible responses.

Another problem of the traditional way to learn to play an instrument is the missing instantaneous feed-back if the student is practising himself. Unrecognised and uncorrected faults will be strengthened during the exercise. Iteration is a very effective way to remember and automate learned actions. Unfortunately iteration enables automated faults as well, resulting in much harder effort to correct the fault, once it was learned. The game can substitute the role of the teacher while the student is practising, correcting every fault at the very moment it was made. In the broader sense we are able to prohibit the automated learning of faults by immediately correcting any improper action. We expect more effective learning, less frustration and an overall better result with the proposed educational game controlling the student while exercising.

The commitment of a music teacher is not cheap, leading to probably unused potential of children, which grow up in an poorly living environment. Thanks to industrialisation and bulk production of hardware elements, smart phones are no longer reserved for the affluent society, but affordable for everyone. Therefore the use of an educational game on a smart phone could offer a great potential for people who cannot or do not want to afford a teacher.

1.2. Goals

The project was designed with three preliminary goals in mind:

1. The first goal is the need for **correctness**, which can be understood in the given context as: Whenever the user produces a tone, the system must be able to determine what tone it was.
2. The second goal is **completeness**: All tones that are recognised, are produced by the user. A major issue in this concern is noise. Even very gentle sounds would be recognised and analysed. Any body movement, the voices of people in the adjoining rooms or even the breathing of the user would result in some measurements by the system. The issue of noise was solved by a simple approach of comparing the magnitude of a signal with a predefined threshold (we will call this procedure **thresholding**). Measurements with a sound volume below some chosen threshold are not handled because they are assumed to be noise. The exact choice of this threshold and the implications of this approach will be explained in Chapter 5.
3. A third goal was to design a system capable of **detecting multiple pitches** at the same time. The system was designed for one single instrument, the piano. The fact that the player is able to produce multiple pitches on a piano was the driving reason for this goal. Initially we restricted the use to one single instrument, namely the piano. We arrived at this decision due to the fact of possibly very distinct structures of partials for different instruments. It turned out to be not that different to turn the measurements unusable for the instruments we checked. We were able to excel the primary goal in this aspect to multiple instruments, which obey the theory of harmonics. The system was tested for the following instruments: Piano, e-piano, soprano recorder, alto recorder, accordion, pure sine waves, and voice.

1.3. Structure of this Document

In this section we will present an overview of the following chapters:

Chapter 2 will describe the required background information needed to understand the decisions made in the project. Three orthogonal areas of science span the range touched by this project. In particular we will go through the basics of music, Android programming principles and education theory.

Chapter 3 addresses the problem of sound analysis and presents related work. A majority of projects solving a sound analysis problem are using some kind of fourier transform, but this is not the only way to solve sound analysis problems. We will cover one technique, called the zero crossings algorithm and shortly describe the difficulties of this approach and name some reasons why the fourier transform was chosen in the project instead of the zero crossings algorithm. Afterwards we will present related research papers that were supportive for this project.

Chapter 4 describes technical aspects that are relevant for the design of the project. This includes a discussion about hardware requirements, the idea of a common setup, how the application is supposed to be used, involved libraries and what they are needed for.

Chapter 5 demonstrates aspects about the implementation. This includes discussions about the graphical part, the part about sound analysis, the game controller, as well as the user interface. In this chapter we will give an introduction on the problem of frequency analysis.

Chapter 6 presents the results and the setup of our three user experiments. Two of them were conducted and will be discussed in detail. The last study was not conducted and will therefore only be presented in the theoretical point of view.

Chapter 7 concludes the project and summarizes shortly the most important points. We will finish with some ideas about possible future work and improvements of the application.

1.4. Conventions

In the remaining section of this chapter we will introduce some conventions made for the sake of readability and briefness of the document. Throughout this document, we will adhere the following conventions:

- To reduce redundancy and increase readability, we will refer to 'the user' (or other third parties) as a male person. We do not intend to discriminate anybody.
- Code examples will be given in Java, due to the fact that the project was written in Java. This will be the case for the entire document if not specified differently.
- We sometimes use the term screen size for the effective size of the screen but more often as the resolution in pixels. The intended use should be clear from the context.

2

Background

Three orthogonal branches (music, education, and mobile phone applications) span the range involved in this project. For each of them we will shortly describe the most important concepts, enabling a deeper understanding of fundamental mechanisms we used in the project. An application in the intersection of several fields of research offers many possibilities but at the same time it brings possibly many difficulties together, sometimes aggravating each other in unpredictable ways. Therefore we will handle each topic individually and meet the most challenging issues related to these areas. We briefly show in what way these problems affected the work, and how they were resolved.

2.1. Music

We will discuss aspects related to the music first. We will cover the notational representation and the physical meaning of notes and sound. We will start with the physical properties, followed by notational and graphical aspects of music.

2.1.1. Physical Tones

A tone is an acoustical phenomenon produced by a sound emitting subject. We are surrounded by molecules that underlie an atmospheric pressure, induced by the weight of the molecules above our head, that are attracted by the gravity. This pressure is constantly varying, as a consequence of constant movement of the molecules in the air. The human is not able to detect minuscule changes in the air pressure if the particular level of pressure surrounding him varies less than the smallest sensation magnitude perceivable, he experiences silence. Sound, on the other hand, is experienced in the case of a rapid variation in the average level of pressure surrounding a human. These variations in the density of the molecules are propagating through the air with a particular speed and magnitude. These two attributes are known as characteristic properties for waves. Thus, we can describe sound as waves in the density of the atmospheric pressure propagated through the air [Indiana-University Mar. 2014].

A **tone** usually is a sound induced by an instrument, whereas **sound** can be the effect of a natural phenomenon. Wind would be an example for a sound. The stroke of a key on the piano clearly produces a tone.

A tone is defined by the following properties:

1. **duration**, time instances the tone is audible
2. **pitch**, frequency of the most prominent wave included in the tone
3. **quality**, conclusion of all waves involved in the sound pattern

The duration of a tone defines the time a tone is perceivable by the human audition system. Music theories propose a quantization to predefined durations. Some examples of these predefined durations are: full notes, half notes, quarter notes, eighth note and so on. A concept to notate an atomic musical element (notes and breaks) was proposed.

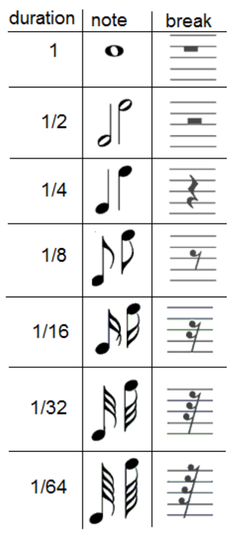


Figure 2.1.: *Duration quantization for notes and breaks [Gitarren-Kids Feb. 2014]*

From a physical point of view a tone is just a set of overlapping sine or cosine waves. Each of these waves has a concrete frequency and magnitude. The instrument affects which waves (what frequencies and magnitudes) are present when a particular tone was played. The *material*, the *temperature*, the *length* and the *shape* of the instrument have a major influence on the frequencies that can be produced by a particular instrument. The quality of a tone is given by the set of all interfering waves. Every instrument produces a unique sound impression, induced by its individual wave constellation with specific frequencies and magnitudes. Sound patterns can be very diverse for different instruments, nevertheless the human is able to detect prominent features in the sound and is able to identify whether two tones originating from different instruments are similar.

A **pitch** is the most prominent frequency in the sound structure of a tone (the frequency with the largest absolute magnitude). Also it can be seen as the degree of highness or lowness of a tone, which is ultimately defined by the frequencies involved. We use a notation defining a octave to contain 12 semitones. A **note** is a tone labelled with a name. (C, Cis, D, Dis, E, F, Fis, G, Gis, A, H, B are commonly known names for the 12 semitones in an octave.) We append the number of the according octave to the note to distinguish a particular note from its namesake one octave above or below: C0, C1, C2, C3, etc.

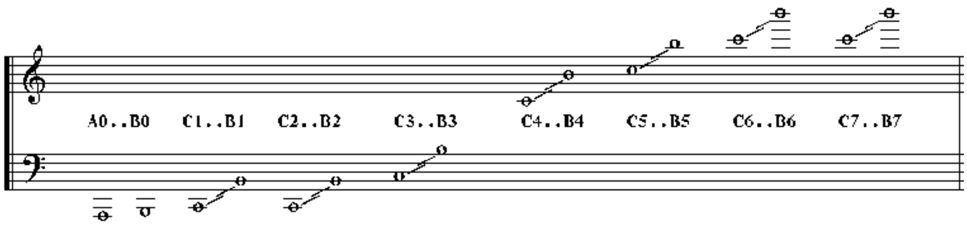


Figure 2.2.: *Scales in multiple octaves [BelowXpectation Feb. 2014]*

The physical correlation between the same notes in different octaves is as follows: The frequency of a particular note one octave higher than its namesake is exactly twice the frequency of the note one octave below. Example: The tuning fork that many musicians use to calibrate their instrument has the frequency 440 Hz, the name of this note is A4. The frequency of A5 is 880 Hz, and the one of A3 therefore must be 220 Hz. A formula to compute the frequency of any note using the frequency of a reference note and the distance between the note and the reference note is shown in Equation 2.1.

$$f_n = f_0 \cdot (\sqrt[12]{2})^n \tag{2.1}$$

Where f_0 is the frequency of a reference note (e.g. 440 Hz for A4) and n denotes the distance in semitones to the reference note (positive if higher, negative if lower). $\sqrt[12]{2}$ is the multiplicative factor by which the frequency increases from one note to the next. It is the 12th root of 2, since a whole octave doubles the frequency and the value has to be shared among 12 semitones. Equation 2.2 shows this effect:

$$f_{i+12} = \sqrt[12]{2}^{12} \cdot f_i = 2 \cdot f_i \tag{2.2}$$

A complete table presenting the name, the wavelength, the frequency, and the number (according to the standard known from the Musical Instrument Digital Interface, MIDI) for every note relevant for the piano is given in the Appendix C.1.

2.1.2. Partials

A tone normally involves numerous waves of different frequencies and magnitudes. The wave with the most prominent magnitude is called **fundamental tone**. The pitch with the highest magnitude is accompanied by some other waves with smaller but still not negligible magnitudes. These waves are called **partials**. The magnitude and the frequency of them determines the timbre of an instrument. Every family of instruments has a different peculiarity of their partials. (Which is the reason why the sound of a particular note on a piano is clearly distinguishable from the sound of the same note played on a clarinet. This fact is an illustration of the asimilar structure of the pianos and the clarinets partials).

The magnitude of the fundamental wave depends on its frequency, the instrument it was played on, and how the tone was played. Therefore, the fundamental wave is not guaranteed to have a higher magnitude than a partial of a much louder fundamental wave (of another tone) that was played simultaneously. The following formula originates from the **theory of harmonics** and defines the frequency of possible partials for a certain fundamental frequency f_0:

$$f_i = i \cdot f_0 \tag{2.3}$$

Possible candidates for partials are waves with a integral multitude of the fundamental frequency. We call the partial with a frequency that is n times higher than the referred frequency the nth partial. Using Equation 2.3 it is possible to inspect partials during the analysis and to filter them from the signal [Wikipedia Mar. 2014].

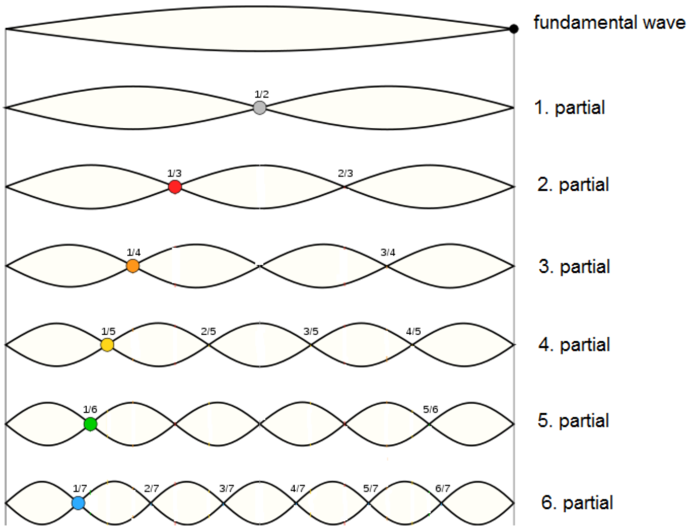


Figure 2.3.: *Fundamental frequency and up to the sixth partial [Wikipedia Feb. 2014]*

2.1.3. Instrumental Differences

The theory of harmonics does not apply for every instrument. It was useful for the piano and the flute, but turned out unsatisfactory for the accordion. The application is not guaranteed to work properly in every situation, since it cannot guarantee to filter all partials correctly if the harmonic structure in the sound pattern is asimilar to the one proposed by the theory of harmonics.

Another issue encountered during the elaboration of this project was the fact that not every instrument has the same spectrum of tones reachable. A piano holds 88 keys, therefore it is able to produce an amount of 88 distinct single tones. The soprano recorder has a maximal number of 31 tones. In addition to the fact that not all instruments have the same **gamut** (range of pitches), they do not necessarily intersect all at the same pitches. The notation of notes is a function with the pitch as a parameter. Every pitch has its own notational representation. For the ease of simplicity to read the notes, they are usually transposed towards the notes without **ledger lines** (additional lines outside of the scope of the usual five lines). Whether a transposition is necessary or not depends on the instrument. If the instrument is capable of producing the notes around C4, no transposition is needed. For instruments like the soprano recorder, whose lowest tone is C5 a transposition of one octave is conducted. The notation for this effect includes a tiny 8 right above the tremble clef. In the case of the guitar, whose lowest tone is a C3, we transpose one octave towards C4. A tiny 8 below the tremble clef indicates this transposition.

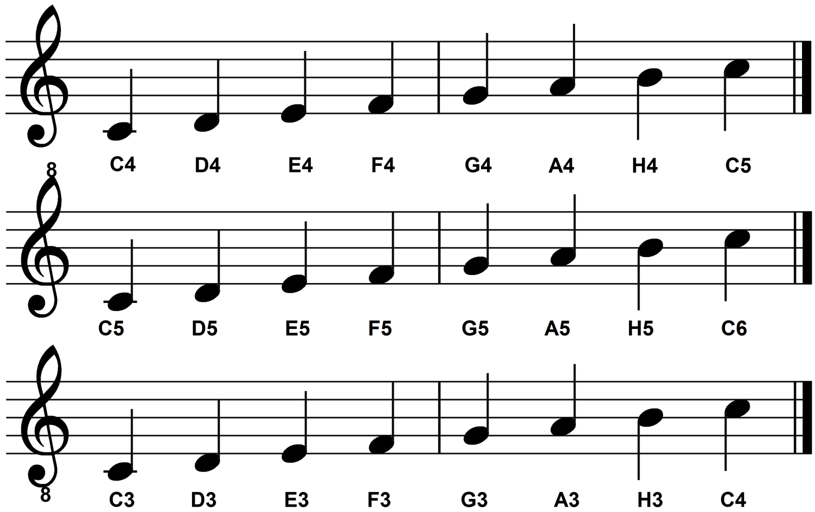


Figure 2.4.: *Transposition: without, downwards, upwards*

Whether or not a transposition has to be performed depends on the instrument. To support multiple instruments the knowledge about their notational nature concerning transposition is crucial. This affects the usability of the application because of the restricted use for certain instruments (whose notes do not need a transposition). The project is constrained for instruments that obey the theory of harmonics and do not use a transposed notation. As a continuation of the project this aspect could be widened to more instruments. Especially the instruments in need of a transposition, but still obeying the theory of harmonics.

2.1.4. Noise

Noise is unwanted sound, in the context of a music game it appears as ambient sound that is not considered to be a signal for the game. Noisy signals originate from the environment, rather than from the user. They occur naturally and interfere with the signal originating from the user. We cannot prevent the existence of noise (and the microphone from measuring it), so we have to filter the noisy signal from the remaining, desired signal. The volume of the signal produced by the user is usually much stronger than the signal produces by noise. Filtering works good enough to separate the signal originated from the user from the noise.

2.2. Android

In this section we will reveal the design principles Android proposes to their developers and how they were applied in our project [Android Feb. 2014].

2.2.1. Android Design Principles

Android makes use of many design concepts, which will be presented here. We indicate in what way the design principles influenced the design choices we made in the application:

- ▷ Sound effects
  The well-chosen usage of sound effects has a motivating effect on the user. We combined the termination of the game with a sound effect. In the case of a success a success sound effect is audible, in the case of a failure, a failure sound effect is played.

- ▷ Choice of icons
  A good choice of icons is very important to reduce the cognitive effort needed to perform a task. Well known icons are understood intuitively, whereas the cognitive effort to interact with a unknown icon is much higher. Our game makes use of three buttons at the end of the game (one to replay, one to skip, and another one to get back to an overview of all levels. These elements appear in many contexts. A comparison of icons used in well-known games to perform the mentioned tasks, lead to the icons we used in our game.

- ▷ No double effort
  Users do not want to enter the same information twice. The application should remember the users preferences and instead offer an interface to change the preferences. In our game we offer three speed categories to the user: Slow, Moderate, and Fast. The preference can be set in the SettingsActivity. Initially the value is set to moderate. When changed, it will remember the preferred value, even when the application is closed and relaunched.

- ▷ Pictures rather than text
  To read is not a natural ability of the human, it has to be learned. We cannot expect all users to be able to read. Furthermore the cognitive effort to react on text is much higher than to react on pictures. Therefore the users get tired earlier. To prevent this, it is important to use written elements as rarely as possible. Only the main menu, the store, and the overview of levels and difficulties contain text. In our application only the main menu consists of pure text elements. The store, the level overview and the difficulty overview use it in combination with pictures.

- ▷ The user is the boss
  People like to feel to be in control of something. On the other hand they feel lost when they have too many choices. A good balance between ceding the control to the user and automatic execution of certain tasks is best. As an example from our application: The mechanism to delete the user account runs automatically, but before running it the user is asked to confirm the deletion.

- ▷ Hiding unnecessary information
  People are overwhelmed when they see too much information at once. We try to break a complex task into multiple independent chunks. Instead of presenting all level at once (which might become quite a few), we assigned them to five difficulty categories. The user first has to decide which difficulty he wants to play and then which level. Another example of hiding irrelevant information is the option to delete the user account. We assume this task to be executed on rare occasions. So it does not need to be apparent on the main screen. Instead we put it into the SettingsActivity.

- ▷ Similar appearance, similar functionality
  The human mind is very powerful in detecting similarities. This can be used to reduce the cognitive effort to understand the functionality of an element, by using similar icons for elements with similar interaction patterns. As an example we make use of three types of objects in the store. Either an extra level is locked, it is too expensive or it can be bought. We indicated these three types with similar icons, differing only in the color and possible additional elements (a lock) on the icon.

- ▷ Fault resistance
  Humans make mistakes. We have to be aware of this and be prepared to recover from all possible mistakes a user could make. In our application the only mistake a user could possibly make is to delete some files we need to run the game. For every level a text file containing the notes and a MIDI file is written to the memory the first time the application launches. If such a file were deleted we could not load the information needed to start. On account of this the application would crash. To prevent this, we check whether all files are located on the memory when the application starts. If this is not the case, we rewrite the missing files.

- ▷ Feedback
  The user needs a feedback for every action he performs to know whether it succeeded or not. Our application reacts visually, all signals that were detected by the application are analysed and are shown to the user. Another feedback we give is a sound effect indicating the success or failure at the end of a level.

- ▷ Speed
  The actions a users performs the most should happen fast. In our application the user can start the game he wants within three clicks. After the first launch he gets within one click to the next or to the same level again. Only two clicks are needed for a level in the same difficulty category (except the next or the current level) and within three clicks to whatever level he wants outside of the current difficulty. We assume it to be very likely that a user wishes to play the current level again or the next one. On the other side we assume the situation where the user wants to play a level outside of the current difficulty level to be unlikely.

2.3. Education Theory

The nature of the human brain is very complex, many theories exist about how people learn best. An experienced music teacher (for piano and flute) answered in an expert interview to questions about the art to support children and adults during the process of learning to play an instrument. We will present the results of this interview in this section.

Christine Ryser-Bringold, a music teacher at a primary school in a suburban village near Biel is experienced to work with children and adults. We conducted an expert interview on how to teach people. We will present the key statements on the aspects of learning.

1. **Is there a difference between the work with children and adults?**
   "Yes, there is a tremendous difference between children and adults. Adults often have coordinative problems they have to overcome, whereas children have to learn how to focus and diligently to practice their exercises. Their educational books are quite different in respect of structure and songs. Children are pleased with colourful pictures, catchy and simple melodies. Adults prefer visually more decent textbooks. The size of their hands affects the choice of songs. Children are often unable to play certain chords, due to the dimension of their body."

2. **How often do you advice your students to practice?**
   "It depends on the age of the student. Mainly the concentration is the limiting factor in this question. Children are not used to stay focused for a very long time, therefore I usually begin with exercises of about five minutes. The period can be increased as the child gets used to it to ten, fifteen, thirty minutes per day. There is no limit, more practice is never a handicap. The driving factor is not how long they are practising per day, but how frequently per week. I advice my students to practice every day, even if it is only for five minutes."

3. **Is it possible to overdo in practising?**
   "As long as they do not constantly make the same faults, no. I made the observation, that students which had difficulties with some part of the song when they were practising at home often learned the fault by heart so that is was even harder to get it right afterwards."

4. **What failures do you observe the most, related to the learning process?**
   "To play music is not only about how it sounds, but also about the technique. Many people have a bad bearing, which influences the quality of their music. It is important to relax the muscles from time to time, sitting upright, and breathing constantly. In my role as a teacher I often have to remind my students to concentrate on their breathing and bearing."

5. **How do you motivate your students?**
   "Feedback is an important factor for the motivation. Everybody likes to get a praise, when they did a good job. If some exercise did not work out, it is time to find the barrier and to work on it. Usually I divide the song into multiple pieces and let the student practice all of them separately, we rejoin it after the barrier was mastered. I think for most of the students it is motivating to master some piece and to get a new song as a reward."

6. **In what way can you assist the students, where do they need help?**

> *"I see my job as a teacher as an assistance for critical sections in a song. The students get a direct feedback, whether they did a job or not, I correct their bearing if it is necessary, I motivate them and I guide their progress."*

2.3.1. Key Components in the Game Idea

Many design choices for the project were established on the statements gathered in the expert interview. In this section we will present concepts and ideas from the game and explain why these choices were made.

Choice 1: *Target group*
Target group of our project are adolescents, children with hands that are already grown-up or adults. People who would like to learn to play an instrument using a (guided) self-study. We expect them to be able to read the tremble and the bass clef, but do not expect any former experience on the piano.

Choice 2: *Learning assistance*
The game provides as assistance a direct feedback about whether a note was correctly or incorrectly played. It is not supposed to correct the bearing or the technique of playing. We are unable to detect which hand was used to play a particular note. Likewise we are unable to detect a cramped position of the body. The application was designed to prohibit a learning of faults by heard by giving a direct feedback for correct or incorrect notes.

Choice 3: *Motivation*
Another advantage of the game we would like to present is the easy way of motivating the user. Games are designed to please people, which directly influences the motivation. Sound effects and the concept to collect points for exercising are supposed to have a positive influence on the users motivation. The user earns points for successfully solved levels and for correctly played notes. Correct notes with a bad timing give a little less points than correct notes at a good timing. These points can be used to buy extra levels. Therefore diligent practice is rewarded with new songs.

2.4. User Study and Evaluation

Several experiments were conducted in order to improve the application. Two of them were user experiments and one was used to find an appropriate value for a system variable, used to filter noisy signals. We will present the three experiments and the respective results in this section.

2.4.1. Determination of System Values

One experiment was used to determine a system value used as threshold in the application to filter noisy signals. We run the experiment in three different environment. One in a relatively silent environment, one in a rather loud (and therefore noisy) environment and one in between.

In all three situations we conducted the fourier transform. An inspection of the resulting magnitude values revealed information about the severity of the apparent noisy signal.

Our application includes a filtering mechanism after the fourier transform to get rid of the noisy signals. We cannot prevent noise from happening, therefore we need to filter it. The problem is, that a too high threshold will not measure any noise, but maybe it will not measure real signals (originating from the user) too. This effect is very annoying for the user and has to be prevented whenever possible. On the other hand, if we set the threshold too low, we can detect all signals originating from the user, but maybe we would analyse noise as well. We want to analyse as little noise as possible. Therefore a well-chosen threshold value is important. We run the experiment and decided on a value based on the results we got by the experiment.

2.4.2. Choice on Functionality of the System

Another experiment we conducted with 12 participants was to determine what functional elements a possible user might prefer the most. For this purpose we arranged two prototypes with different game principles. The first prototypes main purpose was to provide a solid feedback to the user. The feedback included information about whether a note was detected or not and what note is was (including the information whether the note was correct, too high or too low). The concept of this first prototype did not include any rule on the timing. The user could play a note as often and whenever he wanted. The second prototype provide only the information whether a signal was correct or incorrect. Any further information (too high or too low or whether a signal was detected at all) was not provided. This prototype forced the user to play a note at a particular time. We presented the functionality of both prototypes to the participants and checked the understanding of all participants regarding the differences between the two prototype. After a demonstration of all functional elements each participant was confronted with a set of 31 attributes. The participants had to assign either prototype A or prototype B to all attribute individually. Depending on their opinion, they chose the prototype that convinced them more in the respective aspect. Every attribute was an aspect of one of these key aspect we were interested in: *Motivation*, *Intuition*, and *Effectiveness* in the process of learning to play an instrument. We made three hypothesis we wanted to examine in this experiment. They were:

1. The participants prefer prototype B in consideration of an intuitive handling
2. The participants think prototype A is more effective in the learning process
3. The participants prefer prototype A in consideration of the motivation

We were able to validate the last two hypothesis, but the first one was falsified. A clear trend towards an overall preference of prototype A over prototype B was observed. Therefore we proceeded to improve prototype A with the improvement ideas of the participants.

2.4.3. Usability Testing on the final Prototype

A second user experiment was conducted on the final prototype. The goal of this second user experiment was to detect possible usability issues. The system used in this experiment included all functional elements of prototype A, and had an additional element to include the timing. We had a contribution of eight participants. The cognitive walkthrough they had to traverse confronted the participants with all functional elements used in the game. The execution was observed by an experimenter. Whenever a user did not behave as assumed, the experimenter tried to identify the reason for the observed usability issue, classify it as one of the four reasons: *Misunderstanding of the task*, *visual defect*, *misleading appearance* (names, shapes) or *logical defect*.

We detected one visual and one logical defect. The visual defect was caused by a color problem, where the background had the same color as the text in front of it. The logical defect was caused by an inconsistent behaviour of elements with similar appearances. These issues were fixed or can be found in the listing of possible improvements in Chapter 7.

3

Related Work

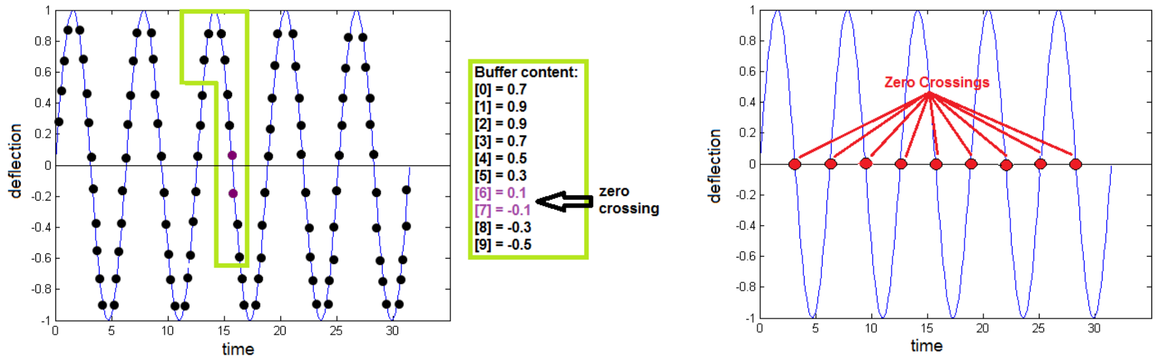
In this chapter we will present the environmental research topics. Three papers influenced the choices we made in the project. In addition we will give an example of a fundamental different approach to the problem of frequency analysis.

3.1. Zero Crossings Algorithm

The zero crossings algorithm is a method to determine the frequency of a given signal. We assume the sampling frequency to be known and constant. The point where a function is traversing the x-axis (and therefore the y value becomes zero) is called **zero crossing**. The approach used in the algorithm proposed here is to count the number of zero crossings in a certain time interval and to estimate the frequency of a wave from these values. The frequency of a wave can be described completely by the number of zero crossings, since the frequency is defined to be the number of impulses per time unit. The number of these impulses is equal to the number of zero crossings.

The microphone usually writes into a buffer of shorts or bytes. Whether shorts or bytes are used has an influence on the accuracy of the measurements read by the microphone. The buffer has a certain size. The bigger the buffer, the larger is the interval which can be read and analysed and consequently the longer is the time period that is stored in the buffer.

The algorithm goes through the buffer and inspects the sign of the value stored in the buffer entry. We found a zero crossing whenever the sign of a entry is different than the sign of the previous entry (since waves are assumed to be continuous and according to the *intermediate value theorem* we found a zero crossing if the sign of subsequent samples from a continuous wave change). The only thing needed to know after counting the number of zero crossings in



(a) Sampling *(b) Zero Crossings*

Figure 3.1.: *Zero crossings algorithm*

the buffer is the duration of the time period. This can be found using the number of entries in the buffer and the sampling frequency.

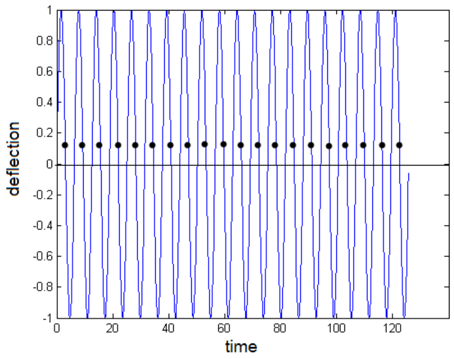
The following code demonstrates the algorithm for a buffer audio data and reading bytes:

```
private int zerocrossings(Byte[] audioData, int samplingfrequency){
    //initially the number of crossings is 0
    int numCrossings = 0;

    //traverse the buffer and count number of crossings
    for(int p=0; p< audioData.length-1; p++){
        if (audioData[p]>0 && audioData[p+1]<=0) numCrossings++;
        if (audioData[p]<0 && audioData[p+1]>=0) numCrossings++;
    }

    //compute the frequency using the number of zero crossings and the buffer size
    int frequency = (samplingfrequency/audioData.length)*numCrossings/2;
    return frequency;
}
```

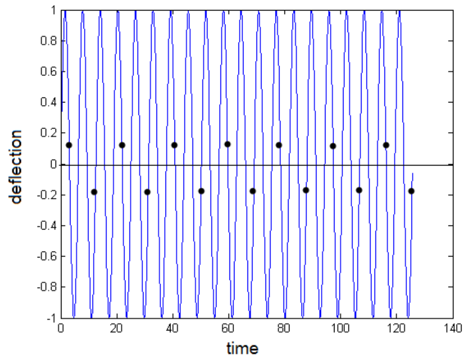
The advantage of this approach is its speed. It runs in $\mathcal{O}(n)$, with n being the buffer size. Whereas the algorithm we used (the fast fourier transform) runs in $\mathcal{O}(n \cdot log(n))$.

Disadvantages are high requirements on the environment. The algorithm turns out unusable when noise is apparent. Likewise it can only be used to determine the frequency for a single wave. It is not usable to analyse the frequencies of two waves that were audible at the same time. And moreover the algorithm supports only some frequencies well. Assume a wave with a very high frequency and relatively low sampling rate. If we detect a zero crossing, we cannot know if the wave had just one zero crossing in the particular interval, or maybe three, five, or seven zero crossings. Even worse: If we detect no zero crossing, it might be the case that the wave had actually an even number of zero crossings without being able to detect them.

The issue of a limited accuracy when estimating the frequency of a continuous signal using a constant sampling frequency f is widely known. A relation between the sampling frequency and the maximal frequency, which can be estimated on the samples is given in the **Nyquist-Shannon sampling theorem**. It says that the maximal frequency which can be estimated from



(a) Issue: No crossings found



(b) Issue: Not all crossings found

Figure 3.2.: *Issues in the zero crossings algorithm*

a sequence of samples is:

$$f_{nyquist} = \frac{1}{2} \cdot f_{sampling} \tag{3.1}$$

The implication of this theorem becomes visible in Figure 3.2, where the frequency of the signal is faster than the sampling rate. The theorem claims that we need in every oscillation of a wave at least two samples (which will have different signs) to detect every zero crossing. The same issue was present for our solution of the frequency determination problem, but it had no influence on the correctness of the results, since the highest pitch on the piano has a frequency of 3 kHz. And at the time of writing smart phones usual in trade have already a higher sampling frequency than 6 kHz. The microphone of our tester phone (Samsung Galaxy S4) had a sampling frequency of 48 kHz (far beyond the frequency of 6 kHz).

3.2. Related Research

In this section we will present two papers that had an influence on the implementation of our application.

3.2.1. Decay Characteristics of Piano Tones

The first paper is about the decay characteristics for different notes on the piano [Shi 2011]. It was written 2011 by Zhengshan Shi at the university of New York. He investigated the speed of decay for different notes and found a two staged process. The first stage of the decay is exponential, afterwards (in the second stage) it continues in a linear manner. He called the first stage **immediate sound** and the second **resonance**. Decay rates of 1.5 dB/s - 40 dB/s for the first stage and 0.3 dB/s - 30 dB/s for the second stage were measured.

In another measurement he discovered variations in the speed of the decay for different frequencies. The non-linear behaviour for the decay time varies with the frequency, a higher note has a decay time significantly shorter than a deeper note. Higher pitches have a much faster decay than lower pitches. He assumes the reason for this behaviour to be the capability of the air to absorb different energy spectra. Low frequencies have less energy than high frequencies and can preserve it better.

The observation made by this paper is that it is difficult to tell how long a particular key on the piano was pressed. The question of the optimal buffer size arises. For high notes it would be preferable to have a short buffer size, since the decay time is probably very short and we do not want noise to overcome the signal. On the other hand it would be preferable to have larger buffer sizes for lower notes, due to their longer decay characteristics an exacter measurement is expected for a larger buffer size. We decided to inspect the note (or the notes) to be played and decide depending on their pitches, how long the buffer size should be.

3.2.2. Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals

Another paper that affected the choices for our project was the one by Anssi Klapuri, Tuomas Virtanan, and Jan-Markus Holm from the university of Tampere in Finland [Klapuri 2000]. It was written in the year 2000. They implemented a method to estimate multiple pitches of concurrent sounds. Polyphonic pitch recognition is much complexer than single pitch recognition, because of probably very different ranges of pitches and the number of notes that are played simultaneously. They established a method to support multiple instruments. Measurements with up to six simultaneously played noted were conducted. Additional to the frequency estimation they used a mechanism to perform a sound separation. We used the same algorithm for the frequency estimation as they did: Using short time fourier transform they analysed the buffer entries, and performed a sound spectrum estimation. They filtered all partials and started over again. Initially a filter is used with a particular threshold to reject spectrum estimation where no note was played, but only noise was measured. The algorithm supported waves with frequencies between 50 Hz and 6 kHz.

3.2.3. The Design with Intent Method: A design tool for influencing user behaviour

In a paper written by Dan Lockton, David Harrison, and Neville A. Stanton in the year 2009, they present several helpful design principles to influence the users behaviour in a way the system designer intends [Lockton 2009]. Most of these principles are based on environmental and ecological psychology. The user does not necessarily need to be aware of this influence in every case. In our case the goal is clearly to make the user producing the correct notes. As this goal is clear to the user as well, it does not matter whether he is aware of the system influencing him or not. The goal of the project presented in the paper was to reduce the likelihood of a customer leaving his card in the ATM. Two modes are presented to achieve the goal. The first mode is called *inspiration mode*. The designer makes use of some design patterns presented in six lenses (architectural, error-proofing, persuasive, visual, cognitive, and security). The second mode is called *prescription mode*. In this mode the designer defines the desired user behaviours. For each target behaviour he decides the most applicable design patterns (again the six lenses presented before) and implements those that apply the most. In our application the design patterns for the visual and the cognitive lens applied the most.

4

Technical Aspects

This chapter presents the technical aspects of the project. We will present the hardware the application was design for and tested with. Proceeding with a presentation of all requirements, that is the permissions necessary to run the game as well as expectations on the operation system and its version on the target device. The common setup and a listing of all involved libraries will follow.

4.1. Hardware

The application was designed for a Samsung Galaxy S4 as a target device. Its display has a size of five inches, the resolution is 1920 x 1080 pixels (1080p). A quadcore processor (Qualcomm) with a clock frequency of 1.9 GHz (Snapdragon 600 APQ8064T) is integrated [Samsung Feb. 2014].

4.2. Requirements

The application requires two Android permissions. The first one is used to record audio. The necessity of this requirement is due to the use of the microphone to measure the sound produced by the user. A second permission is to write to the external storage. This permission is used to write levels and preferences of the user to the secure digital memory card (SD card) of his mobile phone. Initially the levels are stored in the assets of the application. After the first launch all regular levels are written to the SD card. Extra levels still stay in the assets, they are written to the SD card at the time when they are bought by the user.

Minimal requirement to the Android API running on the device is version 13 (equal to 3.2), target version is API 19, the newest version at the time of writing. The minimal requirement was set to 13 because of the use of a concurrent data structure for queues introduced in API 13. Such a concurrent queue was used to implement a producer-consumer paradigm between two types of threads (microphone listener treads and the threads calculating the fast fourier transform on this data).

4.3. Common Setup

The setup to use the application was intended to be in a reasonably quiet room with a piano and a mobile phone. The mobile phone replaces the sheets containing the notes. The phone is supposed to be placed somewhere near the body of the instrument, such that it is able to measure the sound produced by the user (for instance directly on the beam).

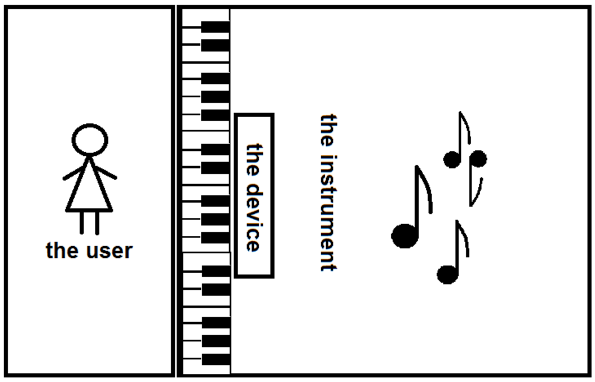


Figure 4.1.: *Common setup to use the application*

4.4. Involved Libraries

The application was developed in Java, beside the general Android library used for activities, views, and the detection of events, two other libraries were used. We will explain them and show in which context they were used and why.

1. **jMusic**
   jMusic is a music library [R.Brown Feb. 2014]. We used it to generate MIDI files. The format of a song was adapted to the format expected in jMusic to generate MIDI files. At the first launch of the application all levels are written to the SD card and read into the library to produce a MIDI representation of each level, stored on the SD card as well. The user gets the possibility to listen to the levels additional to the possibility to just playing them. This possibility is useful at least for those users who like to get a audible version of the level. If the user gets stuck somewhere they have the possibility to get an acoustical sample solution. This might be helpful to detect possible faults.

2. **gdx**
   Another library we used was gdx [Fede Oct. 2013]. It is a part of the library lidgdx containing a version of the fast fourier transform (fft). The fourier transform was used to translate a sound signal from spatial domain into the frequency domain, where the frequency analysis took place.

5

Implementation

We will present the basic ideas and implementation details to the project in this chapter. The topics to be discussed are divided into the following sections: Frequency Analysis, Software Components, Graphics, Sound Analysis, Game Controller, and User Interface.

5.1. Frequency Analysis

Frequency analysis is the problem to find the frequency components in a signal consisting of samples. Sampling is the process of measuring some values (in the case of the microphone it is the ambient pressure level of molecules in the air). One way to transform the information stored in samples to information about the frequency was presented in chapter 3, the zero crossings algorithm. This algorithm suffers from inflexibility and becomes unusable if noise is present in the samples.

The approach we used to solve the problem was the **short time fourier transform**. This algorithm is similar to the fast fourier transform in the sense of its operations. The main difference between the two algorithms is the number of samples they take into account. The fast fourier transform usually just uses the whole range of samples it is given. This returns reliable results for images. If a signal is likely to change over time (sound is fading if the signal is not regenerated repeatedly), the approach of the fast fourier transform is not satisfactory. The short time fourier transform takes a shorter time interval into account and computes the fast fourier transform on this chunk, which we will call *window*.

The fourier transform needs a set of samples (we will call them x) of the length $2n$. It returns a set of frequencies of the size $2n$ as well, using the following formula:

$$f_m = \sum_{k=0}^{2n-1} x_k \cdot e^{-\frac{2\pi i}{2n} mk} \tag{5.1}$$

Where m is in the range $0, .., 2n - 1$.

The fourier transform returns an array. Every entry in this array corresponds to the measurements of the air pressure for a particular frequency. The measurements in f_m are complex numbers representing the magnitude of the respective frequency. In a system where the first entry has index 0, we can compute the frequency of the bin with index i (between 0 and $n - 1$) as shown in Equation 5.2. We define l to be the size length of the array, fs the sampling frequency, and i the index of the respective bin.

$$f_i = \frac{(i + 1) \cdot fs}{l} \tag{5.2}$$

The volume of a particular frequency can be computed using the complex entries of a bin with index i (between 0 and $n - 1$) as follows:

$$volume_i = \sqrt{real_i^2 + imag_i^2} \tag{5.3}$$

Where $real_i$ is the real part of the complex entry with index i, whereas $imag_i$ is the imaginary part of the same entry. Equation 5.3 holds only for $i \in \{0, n - 1\}$. The inspection of values e_i with $i \in \{n, 2n - 1\}$ reveals no new information, the same results as for the first part are shown in reversed order.

A survey of all volume values will point out frequencies, which were measured at a higher volume. The highest volume is assumed to be the fundamental frequency.

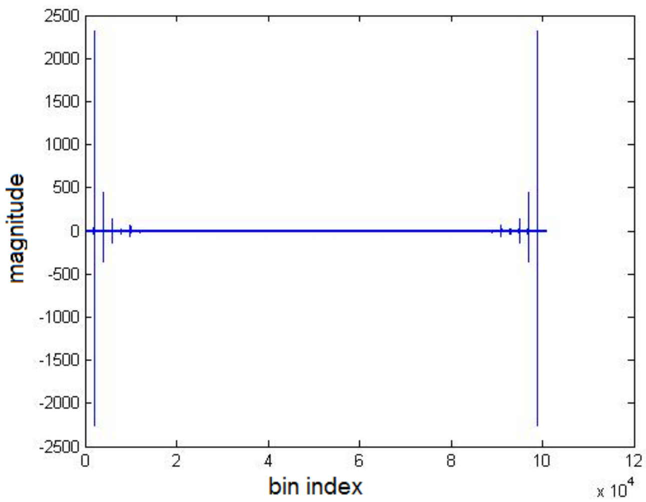


Figure 5.1.: *Array entries (magnitude) for measurements of A4*

The peaks in Figure 5.1 show the result of the fourier transform followed by the computation of the volume for a particular sound signal. The biggest peak belongs to the fundamental frequency, which is at 440 Hz in the case of an A4. The peak next to the biggest peak belongs to the first partial, with a frequency of 880 Hz. The peak next to this one is the second partial with frequency of 1320 Hz. The environment of this measurement was very quiet, as can be seen on the noise level, which is negligible.

5.2. Software Components

The application was built using the **model-controller-view** pattern. The **view** is managed by the graphic thread, which gets the input to do the visualisation from its **controller** (the GameActivity). Other parts that are connected with the controller are two **models**. One is used to control the microphone, to write the measurements into a buffer which is read and written into a queue that is shared with the analysis component, the FFT-thread. The GameActivity creates a listeningThread, used to read the microphone buffer and to write the measurements into a shared linked blocking queue. This is a data type very useful when used in parallel programs to share data. It has a built-in mechanism to ensure thread-safe accesses to critical data. The queue contains Byte arrays, written there by the listening thread. Another thread continuously reads the data from the queue before removing it and performing an fft-analysis on it. We implemented it in respect of the **producer-consumer** paradigm. The GameActivity decides when a thread should be active and when to stop. The threads are running only when the game is actively used. More information on these three categories will be given in the respective sections of this chapter.

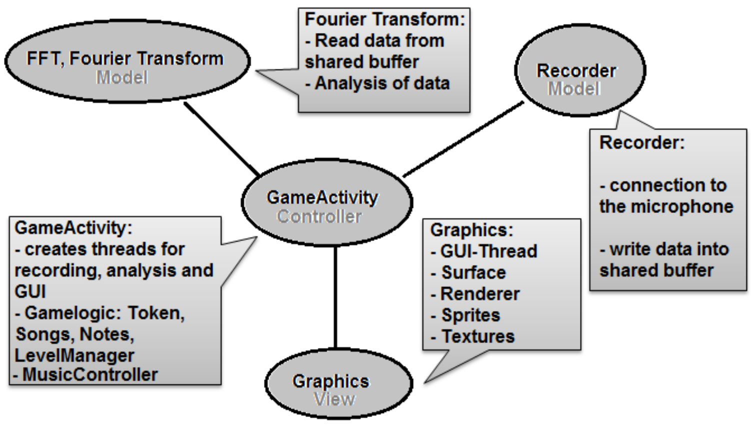


Figure 5.2.: *Controller, Model, View*

5.3. Graphics

We implemented all graphical aspects of the game using openGL ES (open graphics library for embedded systems) version 1.0 designed for Android. All activities except the GameActivity use the proposed Android graphic widgets. The GameActivity itself defines a customised surface to get informed about occurrences of touch-events on the screen. A general class created to support any kind of graphical item. The renderer we established initiates all sprites and draws elements, depending on the state of the game (running, successfully finished, unsuccessfully finished).

5.4. Sound

In this section we will present software parts related to sound. We will demonstrate the data types for notes and songs, how we handled the aspect of timing in the game, which algorithm we used to detect noise and in what way the creation and playback of MIDI files was accomplished.

5.4.1. Notes

We assume a note to be uniquely defined with the pitch and the duration of a sound. The fast fourier transform returns a frequency spectrum, which can be analysed. The analysis of the frequency with the highest magnitude reveals the pitch.

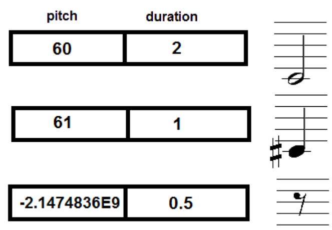


Figure 5.3.: *Data Type for Notes*

Pitches between 33 and 109 are possible to play on the piano. A pause does not have a pitch, the jMusic library we used to generate MIDI files defined pauses to have a pitch of -2.1474836E9 we used this value to be able to use the library without the need of of a conversation of notes. The duration for pauses can attain any of the following values: 0.25, 0.5, 1, 2, 4. The duration for non-pauses can attain any of the following values: 0.25, 0.375, 0.5, 0.75, 1, 1.5, 2, 3, 4, 6.

5.4.2. Songs

A monophonic song (i.e. a song with only one phrase) is semantically the same as a series of consecutive notes and pauses, each with its individual pitch and duration. Such a series is called a **phrase**. Every phrase is implemented as a ArrayList of Notes. Songs on the piano usually are polyphonic (i.e they can possibly containing multiple phrases). The notes written in the tremble clef usually build one phrase and the notes in the bass clef another one. We used exactly as many phrases as the song has notes that are played simultaneously. The whole data type for a song consist of an ArrayList of ArrayLists of Notes.

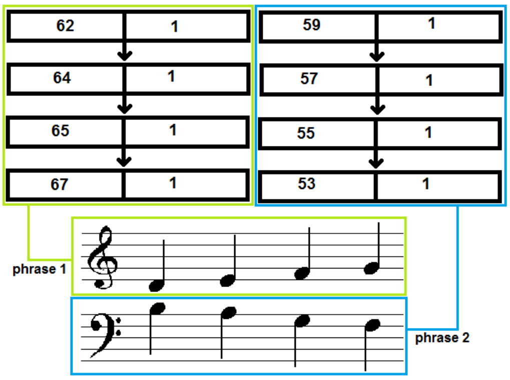


Figure 5.4.: Data Type for Songs

5.4.3. Timing

We enforce the user to obey the correct timing using an indentation corresponding to the duration of the note. That has the effect that all notes, which are supposed to be played at the same time are written below each other, at the same indentation.

Another method we used is to reward the user when he follows the timing we intended. The gaining of points is five times as high when the player plays at the correct moment, whereas he only gains 1 point for the correct note at the wrong timing.

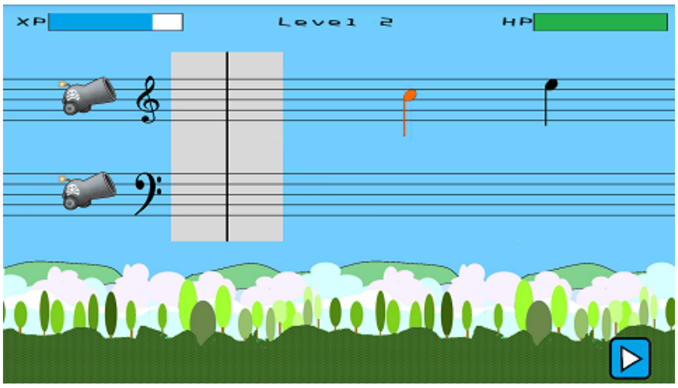


Figure 5.5.: The grey bar indicates the time when to play a note

The game includes a grey bar with a central line to indicate the perfect timing to play a note.

5.4.4. MIDI Playback

Our class LevelManager implements the JMC interface, which is the library of jMusic. In this library a method exists to write MIDI files under the condition that all phrases are already present. The user gets to choose what instrument and what speed the song should have. We decided on piano and 100 beats per minute. When launching the game the LevelManager gets asked whether all predefined levels are present on the SD card. If not (in the case of the first launch or if the user deleted a file), a method is invoked to (re)write the missing files. The playback during the game, if the user wants to hear an audio example for a certain song is set

Environment	*Average silent*	*Average loud*
Office with 3 people	16	661
Outside, not close to a street	24	722
Outside, close to a street	82	790

Table 5.1.: *Noise values in three environments*

up with a MediaPlayer object. The MediaPlayer is able to stream ordinary music files from the SD card of the mobile phone. Since all levels are written to the SD card (*secure digital memory card*) as text files containing the notes and as a MIDI file, we can just call a MediaPlayer instance on the respective MIDI file.

5.4.5. Noise Detection

Noise is a signal we are not interested in. We cannot prevent it from happening in nature and therefore we need to filter it. To do so we analysed the attributes of the desirable signals and the attributes of noise. The sound originating from an instrument has to obey the rules imposed by the sound characteristics of a particular instrument. It cannot exceed the range of pitches reachable by the instrument and it is unlikely to overshoot or to fall below a certain volume. To address the first property, we applied a bandpass filter in the frequency domain after the fourier transform. The buffer might include more frequencies than we actually need, because certain frequencies cannot be produced on the piano (pitches which are too low or too high to be played on a piano). We only analyse the band of frequencies reachable on the piano. The second property became useful as well. Noise usually occur in all frequencies (and therefore also within the band of frequencies that can occur on the piano), but not very prominent in respect of the volume. The sound produced by the user on the other hand is very likely to have a noticeable difference in the volume for certain frequencies. The volume of noise depends on the environment. A unwanted sound signal (e.g. measurements close to some street) can have a much louder volume than a noisy signal in a soundproof room without any distracting influences. We conducted a measurement on the volume of noise in various environments before deciding on a concrete threshold value. The results of these measurements are shown in Table 5.1.

The unit of the entries in the array after we performed the fourier transform is identical to the volume (in decibel) that was measured for the particular frequency, where the respective frequency of a bin with index i is determined as shown in Equation 5.4.

$$f_i = i \cdot \frac{samplingfrequency}{arraysize} \tag{5.4}$$

The first insight we made was that there are more bins than frequencies, therefore we have to match all frequencies in the array to a frequency that can be produced on the piano. We used a rounding algorithm to perform this task. (Example: The frequency 441 Hz cannot be produced on the piano directly, but 440 Hz can be produces, by playing a A4. We count the volume measurements for the frequency 441 Hz to the ones for 440 Hz, as well as all other measurements for frequencies that round to the frequency 440 Hz.) That is the reason for the relatively high values in Table 5.1. The unit is decibel (dB), but it is a measurement for every frequency individually, so the value computed does not have the meaning of a decibel measurement anymore. Therefore we divide by the number of bins which counter for a particular frequency, getting the average volume per pitch.

The second insight we made was that the distribution of frequencies in the array is constant, whereas the frequencies which are possible on the piano increase in an exponential manner. The gap between existing frequencies is increasing, which means that (since all gaps in the array are filled by measurements) there are much more measurements for higher frequencies than for lower ones and it is not accurate to just adding all frequencies that round to a particular frequency.

We decided on a value of 80 as threshold value to detect noise. The reason for this choice (of a rather high threshold value for noise) is our opinion that a wrongly measured signal, which does not originate from the user is worse than the possibility that the user might play a note too gently and that the corresponding signal might not be analysed since it was identified as noise.

5.5. Game Controller

In this section we will present two aspects corresponding to the game controller, namely the concept of the player and the score.

5.5.1. Player and Score

The player or user in our game is individual in the sense of his previous experience. Every attempt to play the game, be it successful or unsuccessful, is rewarded with experience points. The experience increases with every note that was played correctly and with every level that was successfully accomplished. The timing has an influence on the score. If the note is played correctly but not at the perfect timing, the user gains one experience point. If he does so at the right moment, he gains five experience points. Every song has an individual reward. The more difficult a level, the more points can be gained for a successful completion of a game. The points a player earned during the process of practising the game can be utilized in the store as payment method. The store offers extra songs that can be activated when the user decides to use the experience points he gained for successfully finishing a level or playing the correct notes to buy an extra level. The extra levels are supposed to be ordered according to their difficulty.

5.6. User Interface

We would like to present the user interface in this section.

The main screen offers four possible choices. The first choice is to inspect the store. In the store there are three possible types of items: Either an extra can be bought, is not yet active or it would be available but the user does not have enough points.

As a second choice the user could inspect the acknowledgements or the settings. Within the settings the user can choose one of three speed graduations (at the beginning it is set to medium) or he can delete the user account. If he decides to do so he has to confirm this choice and gets back to the main screen. If he decides not to delete the account he just gets back to the settings.

The most important part of the game follows on the choice to start the game. The overall difficulty of the level can be chosen in the first screen. Afterwards he has to decide which level in the respective difficulty category he wants to play. During the active game the microphone is active and the game is controlled by music. The user can pause and restart the active game by clicking the pause button. After finishing the game, the user either was successful or unsuccessful. The user can go back to the overview of levels and retry the level in both situations. To go directly to the next level it is essential to finish the current game successfully.

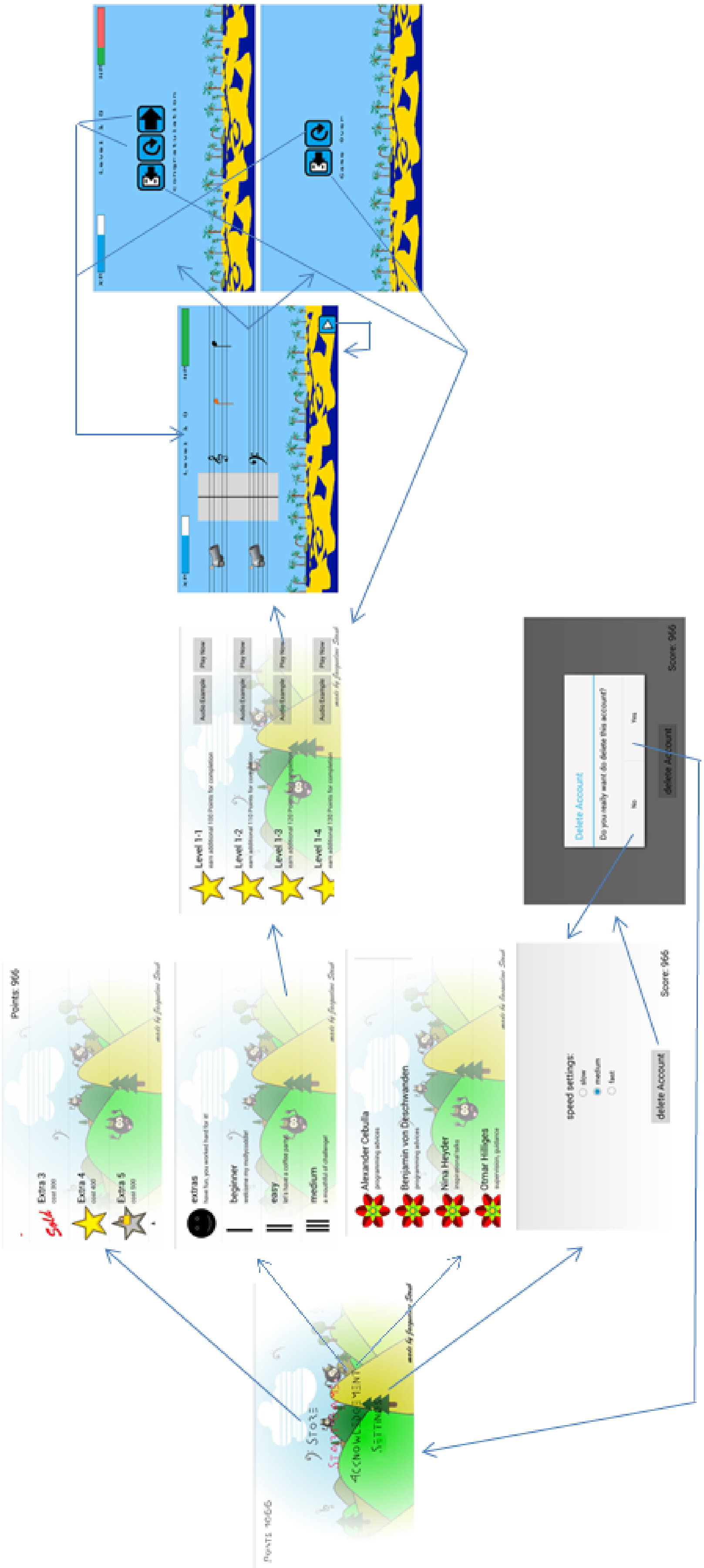


Figure 5.6.: *Storyboard of the game*

6

Evaluation

In this chapter we will evaluate the results gathered in our two user experiments. To begin we will present the methodology of the experiments, followed by a presentation of the two studies, setup and results each. We will finish the chapter by discussing a potential long-term study. Conclusion of all results and a listing of the general limitations will follow.

6.1. Methodology

The first user experiment tested the satisfaction factor of the subsequent aspects in the form of a controlled comparison for two prototypes:

1. Motivation
2. Intuition
3. Effectiveness in the learning process

The second user experiment tested the following aspects in the form of a cognitive walkthrough for the resulting prototype from the improved favourite from the first user study:

1. Acceptance
2. Learnability
3. Memorability
4. User Satisfaction

6.2. Experiment: Comparison of two Prototypes

In a first user experiment two applications fulfilling the same functional methods were presented to the participants. A demonstration of all methods and explanation of the key concepts as well as the differences between the two prototypes were realised. After completing a first questionnaire (presented in Appendix A), used to collect the demographic user information, the participants were asked to name their favourite prototype in consideration of 31 facets of the three factors motivation, intuition, and effectiveness in the learning process.

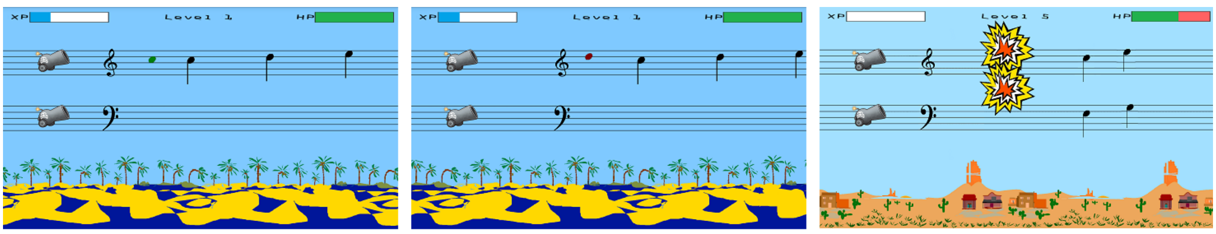
We established the following hypotheses to be validated in the user experiment:

1. A majority of the participants prefer prototype B in consideration of the criteria intuition
2. A majority of the participants prefer prototype A in consideration of effectiveness in the learning process
3. A majority of the participants prefer prototype A in consideration of motivation

6.2.1. Setup

Five men and seven women attended the first user experiment. All of them aged between 17 and 60 years (with an average value 38). Half of them with moderate experience (none to five years practice with some instrument, self-assessing less talented or experienced than the average), the other half with advanced experience in playing music (six to eighteen years of experience, self-assessing above the average). We will now describe the two prototypes and in what sense they differ.

Prototype A:
This prototype is analysing the signal received by the microphone and reports the result to the user. A note, corresponding to the signal, is produced and shown on the screen. Using this approach, the user gets a feedback, whether the note he produced was too high, too low, or too gently. But there is no need to play the note at a specific time, it can be played whenever the player wants, as long as it is played before the note collides with the token, which results in a damage.



a) correct b) incorrect c) explosion

Figure 6.1.: *Screen shots for the first prototype*

In Figure 6.1 three typical situations are presented. If the user produced the note that was expected, a green dot is shown on the same height as the note of the sheet. The notes the user is playing are flying from the left to the right side of the screen, whereas the notes the player should play are flying in the opposite direction. They are crossing in the middle of the screen and either eliminate each other or cause an explosion and correlated damage for the user. A correctly played note by the user (green dot) eliminates the note the farthest on the left. An incorrectly played note creates a red dot on the same height as the corresponding note was that was played by the user. This information can be used to learn whether the note was too high or too low and to make a new attempt to correct the previously failed note. The red dot produces an explosion when colliding with the first note, lead to damage of 10 points for the user (out of 100). A note that was not correctly played up to the time when the note collides with the token results in a damage of 25 health points.

Prototype B:

The second prototype works similarly to the first one, except that it requires the user to play the note at a particular moment, namely when the note to be played is situated directly in front of the tokens mouth. If the player produces a sound at a time where no note is directly in front of the token, the players health points are reduced by 25 points. If there is a note in front of the token and the user produces the expected tone, the note is eliminated without damaging the player, the only information the user gets is a green "OK" sign on the screen, that the note was successfully eliminated. If the user produces an incorrect tone, the health points are reduced by 10 points, the user gets informed by a "X" sign on the screen, but the note is eliminated as well. Note, that in this setup the user gets no response, whether the tone he produces was too high or too low and he has only one chance to get it right, since the note is eliminated in either case.

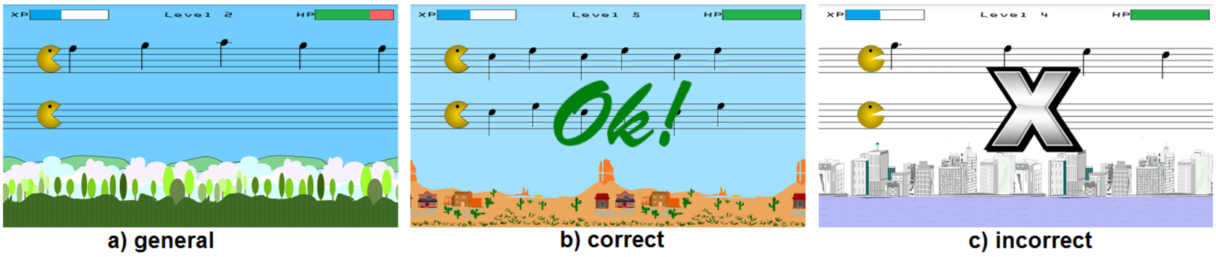


Figure 6.2.: *Screen shots for the second prototype*

After the description and demonstration of both prototypes the participants of the study had to decide which prototype they preferred for 31 aspects, associated to different aspects of the three main criteria *intuition*, *motivation* and *effectiveness in the learning process*.

The last question they had to answer was which prototype they preferred in general, and how that prototype could be improved. Which prototype provides the better potential for further enhancements in their opinion.

6.2.2. Results

The results of the user experiment are presented in the diagram below. The possible outcomes of every measurement were in the set ’a’,’b’,’u’. The measurements for ’a’ and ’b’ are nominal data, they cannot be compared. If someone prefers prototype A over prototype B, we registered it as ’a’. If a participant cannot name a favourite for some aspect, it is registered as ’u’. The three main categories intuition, motivation and effectiveness in the learning process were split into several subtopics. These subtopics were presented to the participants and analysed individually. At the end the results for the related subtopics were combined as a measurement for the main categories.

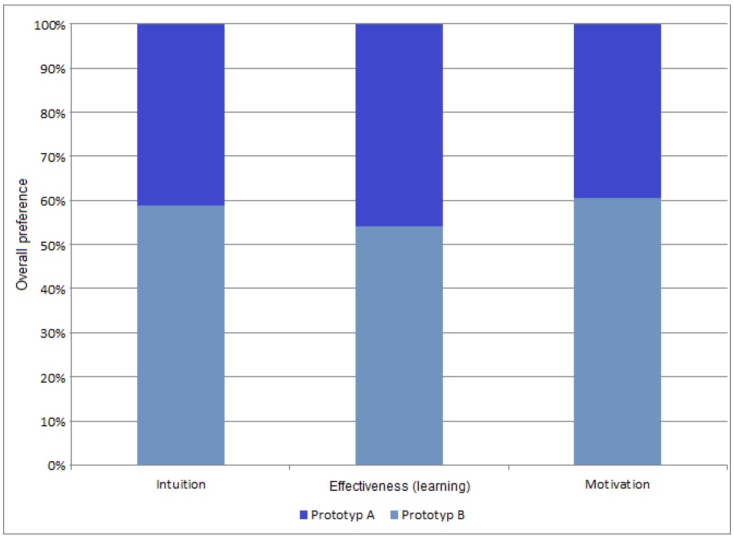


Figure 6.3.: *Results for the controlled comparison experiment*

These were the hypotheses we made initially:

- ▷ A majority of the participants prefer prototype B in consideration of the criteria intuition
- ▷ A majority of the participants prefer prototype A in consideration of effectiveness in the learning process
- ▷ A majority of the participants prefer prototype A in consideration of motivation

Our first hypothesis was clearly falsified, as we can see in Figure 6.3. The second and third hypothesis were verified. Eight out of twelve participants remarked an overall preference for prototype A. Only four out of twelve participants said, they would continue the development of prototype B. Therefore we selected prototype A.

In this listing we will present the advices the participants made to improve prototype A:

1. Enforce timing using a bar to indicate the window for the perfect timing
2. Offer the user to choose between three speed levels
3. Allow the user to halt and continue the game
4. Play sound effect when successfully/unsuccessfully finishing the game
5. Buy extras in a predefined order
6. Button to delete account from MainActivity into SettingsActivity
7. How-to-play, Instructions
8. Verification of the notes played by the user

The first six advices were realised for the second experiment.

6.3. Experiment: Cognitive Walkthrough

The second experiment was a cognitive walkthrough. It was performed to detect issues with the user-control of the application. The participants filled out a questionnaire (presented in Appendix A) used to gather demographic data. After a session of instructions they got three minutes of time to make themselves familiar with the interface before tackling ten tasks that were specified and observed by the experimenter. After the accomplishment of these tasks the participants filled out a second questionnaire (presented in Appendix B). This questionnaire was used to measure the participants satisfaction in the following criteria: Acceptance, learnability, memorability, user satisfaction.

6.3.1. Setup

In total five women and three men participated in this second user experiment. They were aged between 17 and 58 (with an average value of 32). All of them were experienced musicians, but not all of them had experience on the piano. Three participants were able to read the tremble clef only, the others were capable of reading both tremble and bass clef. Three participants did not own a smart phone, one additional person was not familiar with the Android operating system. Four participants stated that the time they spend to interact with their smart phones adds up to less than three hours per week. The others used their smart phones during a period between 3 and 24 hours per week for various activities. One person was color-blind and another person had an uncorrected kind of reduced vision.

After a short time of three minutes to get familiar with the user interface, without asking any questions, the participants were asked to fulfil the following tasks:

1. Adjust the speed to medium
2. Play level 1 of the difficulty beginner
3. Play the next level
4. Retry the current level
5. Play level 2 of the difficulty easy
6. Buy the first extra
7. Play the first extra
8. Stop the running game
9. Listen to the audio example for level 5 in difficulty hard
10. Take a look at the acknowledgements
11. Delete the user account

The experimenter surveyed the execution of the tasks. In the case of a faulty action he had to identify the cause of the fault. It was distinguished between four reasons: Misunderstanding of the task, visual defect (unnoticed choices), misleading name or presentation, concept unclear. The experimenter asked the following questions to identify the cause of the fault:

- What task were you trying to accomplish?
- Were you aware of the presence of the component?
- Were the name and the presentation of the component an indication for its behaviour and how it is used?
- Do you see that the action leads to the desired effect, that was required for the task?

After the tasks were finished the participants were asked to fill out a questionnaire to gather their opinion on four spheres of interest: Acceptance, Learnability, Memorability, and User Satisfaction. Five different aspects for every topic were evaluated and added up to an aggregate value for the topic.

6.3.2. Results

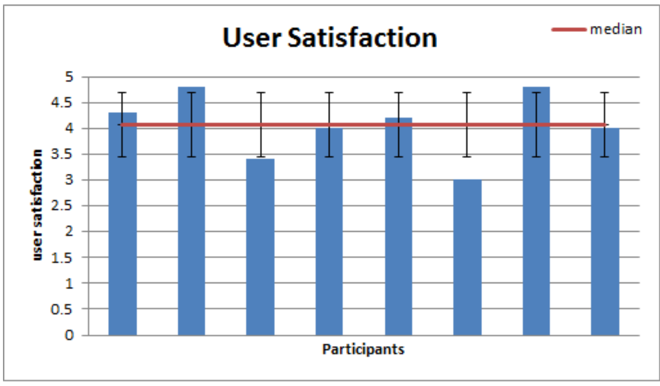
In this section we will present the results of the second user experiment. Acceptance, Learnability, Memorability, and User Satisfaction results will be demonstrated individually. The following diagrams demonstrate the opinion for every subject for every topic. The results are in the range from 0 to 5, where 0 is the worst and 5 is the best score. In blue the answers for all individuals can be seen, the red line represents the median, and the black sign on the top of every result indicates the standard deviation for all answers.

All but three subjects accomplished the tasks without any fault. A serious technical issue became apparent during the execution of the experiment: The threshold value for the noise detection was set too high for most of the subjects on the piano. For notes that were played rather gently the measurement was not loud enough, which is why the signal was neglected and no analysis on those signals took place. After the experiment the threshold value was adjusted.

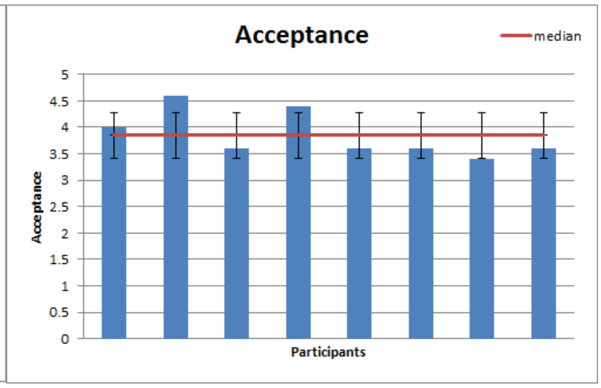
Two subjects were unable to fulfil task number 6. The reason for it was a visual defect due to a badly chosen text color for a button, that was indistinguishable from the background. Another subject did not see the particular button, due to the arrangement of the buttons on the screen, which resulted in a unsuccessful execution of task number 6. We eliminated these usability issues by editing the background image and rearranging the buttons on the MainActivity.

Two subjects failed at task number 7, due to a misconception of the functionality of the store and the platform to actually play a certain level. This usability issue could be resolved by a button for every extra level that was bough with the indication to play it directly from the store.

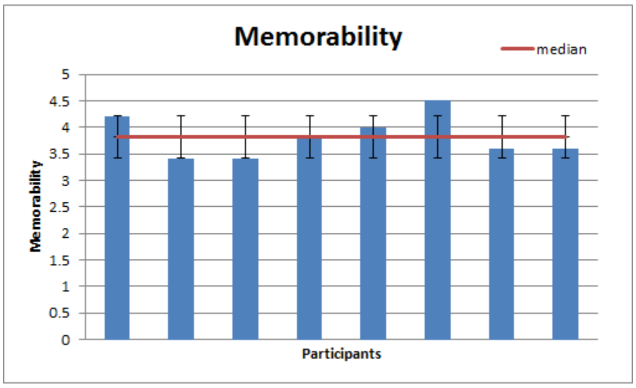
The best results were measured for the user satisfaction values. An average value of 4.06 (out of maximal 5 points) with a standard deviation of 0.4 are evidences to suggest that the participants were rather satisfied than unsatisfied by the game, even though most of them had difficulties to survive the levels due to the threshold for the noise detection that was set too high. Most of them were not frustrated but motivated after failing at a level, because of the direct feedback,



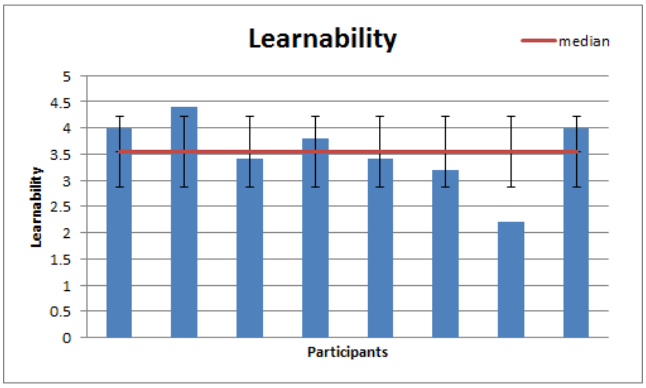
(a) User Satisfaction



(b) Acceptance



(c) Memorability



(d) Learnability

Figure 6.4.: *Results for the cognitive walkthrough experiment*

sound effects, and the possibility to retry the level right away.

The outcome for the acceptance values showed an average value slightly below 4 (3.85) with a value of 0.2 for the standard deviation. The issue with the threshold that was set too high might have a major influence on this outcome. Likewise it might have an affect on the fact that the control of the game was felt complex. The memorability scores had an average value of 3.8 with a standard deviation of 0.15, which was the measurement with the tightest spread. Primarily responsible for this value was the fact that participants with no experience with Android were irritated by the missing back button on the screen. The technical knowledge presumed was felt higher for those people. Android phones usually have three soft keys for operations that are used frequently. There is no need for an additional button for the same functionality, but not all people are aware of this discrepancy. The outcome for the learnability measured an average value of 3.5 and a value of 0.45 for the standard deviation. Main reason for this result might be the issue of the threshold that was set too high. A majority of the participants thought that game was rather difficulty to play, which might be affected by the faulty setting of the threshold value.

6.4. Long term study

In the context of this project we prepared a long-term study, which was not conducted for a lack of time. The results are missing due to this issue, but we will present the idea and the setup for the long-term study.

6.4.1. Setup

The hypothesis we would try to evaluate is the following:
A group using the application would be more motivated and exercise more often and for a longer time, as a result they would have a faster progress than the subjects from the reference group.

We would recruit a large group of music students, randomly assign them to the group which is supposed to exercise using the application or to the reference group. We assume them to know the tremble and the bass clef both. They would be given a set of songs they had to learn individually. In a weekly meeting we would ask them which song they are currently practising and how motivated they are. These weekly meetings would take place for about a year. At the end their statements would be evaluated and compared to the statements of the reference group.

6.5. Limitations

In an experiment we evaluated the hit ratio the application reaches in playing all notes possible for a certain instrument. This experiment was conducted using *pure sine waves*, *a piano*, *a soprano recorder*, *an alto recorder*, and *an accordion* as test instruments. We will summarize the results in the following table:

Instrument	*Hit Rate*	*Range*	*Problems*
Soprano Recorder	100%	72-96	none
Alto Recorder	100%	65-89	none
Accordion	98%	48-104	48 was measured as 49
Pure sine wave	83%	33-104	33 to 42 and 45 to 46 were measured as one note higher than it actually was
Piano	71%	33-104	33-49 measured the first overtone and 101-104 were too quiet

Table 6.1.: *Hit Rate for five instruments.*

Main difficulties occur at the border of the interval supported by the application. The maximum range of notes supported by the application is the same range as the piano has. The piano is one of the instruments with the widest range. Many commonly known instruments lie in this range or have the same range. We will now demonstrate the five most serious problems:

1. Problems at the bottom of the range
2. Problems at the top of the range
3. Problems with different instruments
4. Assumption of loudest fundamental wave
5. Dependency on environmental factors

Some of these issues can be addressed, we will propose some ideas to improve the application in Chapter 7.

6.5.1. Problems at the bottom of the range

The problems at the bottom of the range (see the problems for the sine wave) originate probably from a property of the fourier transform. The bins resulting from the fourier transform are uniformly allocated for all frequencies in the desired range as described in Table 6.1. The distribution of frequencies in a sound pattern has an exponential manifestation (see Table C.1). The result of this linear distribution of non-linear data is a uneven distribution of measurements. We have much more bins for high notes than for a lower note. This behaviour is critical, since an instrument is not guaranteed to produce exactly the correct frequency, so it can happen, that some measurement is mistakenly added into another frequency-window than it actually belongs for deep notes, since there is for most of them only one bin per frequency.

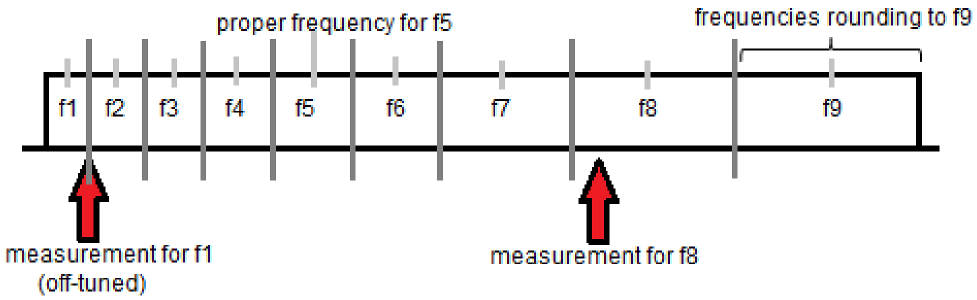


Figure 6.5.: *Problems at the bottom of the range*

6.5.2. Problems at the top of the range

The problem with high notes, as explained in Chapter 3, is that their decrease is faster than the one for deeper notes. The magnitude for lower tones is influenced by many factors - the note currently playing with all its partials, the note played before and all its partials, and all of them might be lingering. It is very likely to top the threshold of the volume and therefore to be detected as note (instead of noise). But this is not guaranteed in the scenario for high notes, where the descent is very fast and it is unlikely to hear the note that was played previously if it was a high note as well. In such a scenario it can happen that a note alone is not able to top the volume threshold and to be heard by the application. A trade-off between completeness (all notes analysed are originating from the instrument) and correctness (all pitches can be detected) is inevitable.

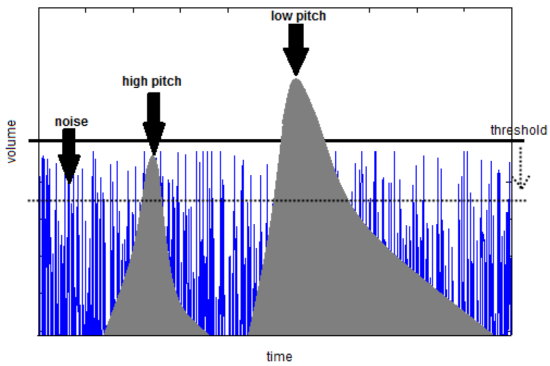


Figure 6.6.: *Problems at the top of the range*

6.5.3. Problems with different instruments

Another limitation is the support for different instruments. They may differ in various aspects. The overall sound intensity, partial structure, and notation of notes for a particular instrument are three aspects that cause problems when supporting multiple instruments.

6.5.4. Assumption of loudest fundamental wave

Our sound analysis bases on the assumption that the fundamental frequency is the one with the loudest magnitude. This holds not always true for the piano, as shown in Table 6.1.

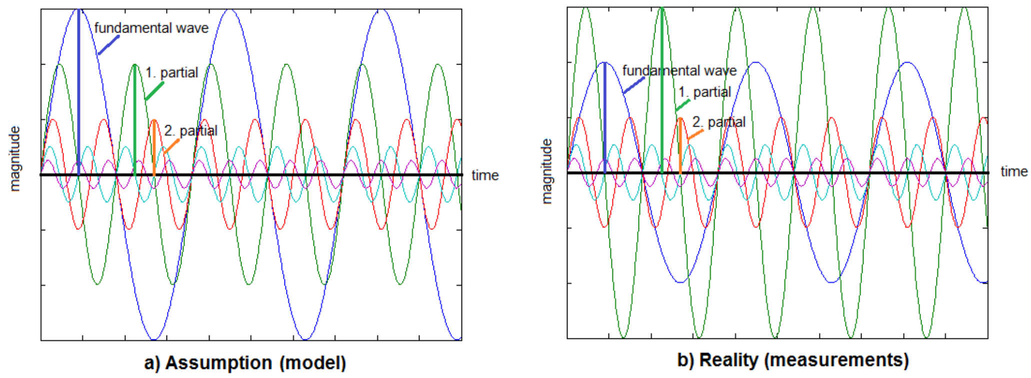


Figure 6.7.: *Discrepancies between model and measurements*

6.5.5. Dependency on environmental factors

The sound of an instrument depends on the temperature of the instrument and the environment. Experiments showed striking discrepancies for the hit rate when the instrument has not the same temperature as the environment or is not warmed up.

7

Conclusion and Future Work

In this bachelor thesis we proposed an application developed to support music students learning and exercising their instrument. We highlight the following aspects:

1. **Estimation for multiple pitches**
   The Android application we developed analyses sound measured by the microphone. We are trying to identify the most prominent pitches and to match them with the excepted notes. We are able to identify multiple pitches at once, which allows us to use the application for instruments that can produce more than one note at a time (e.g. piano)

2. **Support for multiple instruments**
   In addition to the proposed functionality for the piano we are able to use the application for other instruments too, upon the condition that the instrument follows the theory of harmonics. Some instruments that were testers for this purpose are the piano, the soprano recorder, and the alto recorder.

3. **MIDI files as audio examples**
   An auxiliary functionality to support the user with multiple ways of learning is the approach to provide audio examples for every level. We generate MIDI files for all levels and offer them to the user at the moment when picking the level he wants to play. We expect good results for people who prefer audio stimuli rather than visual representations to learn.

4. **Simplicity**
   Our application is designed to be self-explanatory. In two user experiments we gathered information about different parts of the application and observed the interaction of people with them. Some usability issues were detected and a remedy of these issues took place.

7.1. Future Work

We would like to suggest some ideas how to improve the project. The first enhancement originates from the user feedback of the first and second user experiment.

7.1.1. Instructions

Instructions to clarify all functionalities of the application for users which are interested might be helpful. These instructions would have to explain the use of the graphical elements like red or green dots, notes, the grey bar, the line in the middle of the bar, the tokens, and what it means if an explosion is shown. Other elements which would have to be explained are: When exactly the game is won and when it is lost, how to buy an extra and how to play these extras.

7.1.2. Feed-back after finishing the game

One possibility of additional information would be a direct evaluation of the game after it was completed. An overview of all notes and whether they were played correctly or incorrectly and more information about the timing when a note was played. The information could increase the awareness for how many points were collected and for what reason.

7.1.3. Support for transposed notation and multiple instruments

An enhancement towards the support of more instruments has to include the possibility to transpose the notation. In a first step a list of supported instruments and the respective transposition must be introduced. Another possibility would be to involve the user and to ask him to play a certain note. The transposition of the instrument could be concluded from the measurement for this reference note. We could calibrate the application using only some reference notes. This would enable the application to be used either to calibrate the instrument or to be used with massively off-tuned instruments.

7.1.4. Interface to create levels

To allow a higher level of personalisation it would be appreciated to have a user interface to produce levels for the application. So every user could prepare the songs they would like to exercise on. We thought of a web-platform for students to publish and share their songs. Such a platform could improve the usability, since no duplicates have to be created. A database could offer a search functionality whether a certain song exists already. This way a community could be initiated. The community would evolve dynamically.

A

Demographic Questionnaire

Pre-Study Questionnaire

General questions

1. What is your birthdate?
   ________________ (DD.MM.YYYY)
2. What is your gender?
   - ☐ Male
   - ☐ Female
3. Do you have any kind of uncorrected visual defect?
   - ☐ Blindness
   - ☐ Colour-blindness
   - ☐ Cataract
   - ☐ glaucoma
   - ☐ other, namely: ___________________
4. Do you take any kind of medication taking influence on the concentration/temper/vision?
   Namely: ______________________
5. Did you drink any alcohol today?
   - ☐ Yes
   - ☐ No
6. Are you wide awake?
   - ☐ Yes
   - ☐ No
7. Are you distracted right now, physically or psychically?
   - ☐ Yes
   - ☐ No
8. Are you physically and mentally able to play an instrument?
   - ☐ Yes
   - ☐ No

Musical questions

9. Do you play the piano?
   - ☐ Yes
   - ☐ No
10. How many years of practice do you have?
    ________ years
11. How often do you practice?
    ________ times per week
12. Can you read notes written in the tremble clef?
    - ☐ Yes
    - ☐ No
13. Can you read notes written in the bass clef?
    - ☐ Yes
    - ☐ No

Technical information

14. Do you have a smartphone? (a phone controlled by touching the display)
    - ☐ Yes
    - ☐ No
15. Is the smartphone running Android?
    - ☐ Yes
    - ☐ No
16. How long ago did you buy your first smartphone?

    ___________ years
17. How often per week do you use the phone for the following purposes:
    a. Gaming ___________ h
    b. Writing sms ___________ h
    c. phone calls ___________ h
    d. Surfing on the internet ___________ h
    e. Else ___________ h

A. Demographic Questionnaire

B

Post-study Questionnaire

Post-Study Questionnaire

On a scale from 0 to 5 (where 0 is not at all, 5 is very much) how do you answer the following questions:

Acceptance	**Not at all**					**very much**
1. how often would you use the game	0	1	2	3	4	5
2. how complex is the system	0	1	2	3	4	5
3. how cumbersome to use is the game	0	1	2	3	4	5
4. how consistent is the game logic	0	1	2	3	4	5
5. how much did you like the game idea	0	1	2	3	4	5

Learnability

1. how easy to play is the game	0	1	2	3	4	5
2. how intuitive is the game concept	0	1	2	3	4	5
3. how clear is the structure of the app	0	1	2	3	4	5
4. how intuitive is the function of the graphical elements	0	1	2	3	4	5
5. how much help did the game provide	0	1	2	3	4	5

Memorability

1. how easy to remember is the structure of the game	0	1	2	3	4	5
2. how easy to remember is the function of the graphical elements	0	1	2	3	4	5
3. how supportive was the game to learn an instrument	0	1	2	3	4	5
4. how familiar did you feel the second time you played the game	0	1	2	3	4	5
5. how much background knowledge in technical and musical aspects is needed to play the game	0	1	2	3	4	5

User satisfaction

1. how fun is the game	0	1	2	3	4	5
2. how emotionally fulfilling is the game	0	1	2	3	4	5
3. how rewarding is the game	0	1	2	3	4	5
4. how aesthetically is the game	0	1	2	3	4	5
5. how motivating is the game	0	1	2	3	4	5

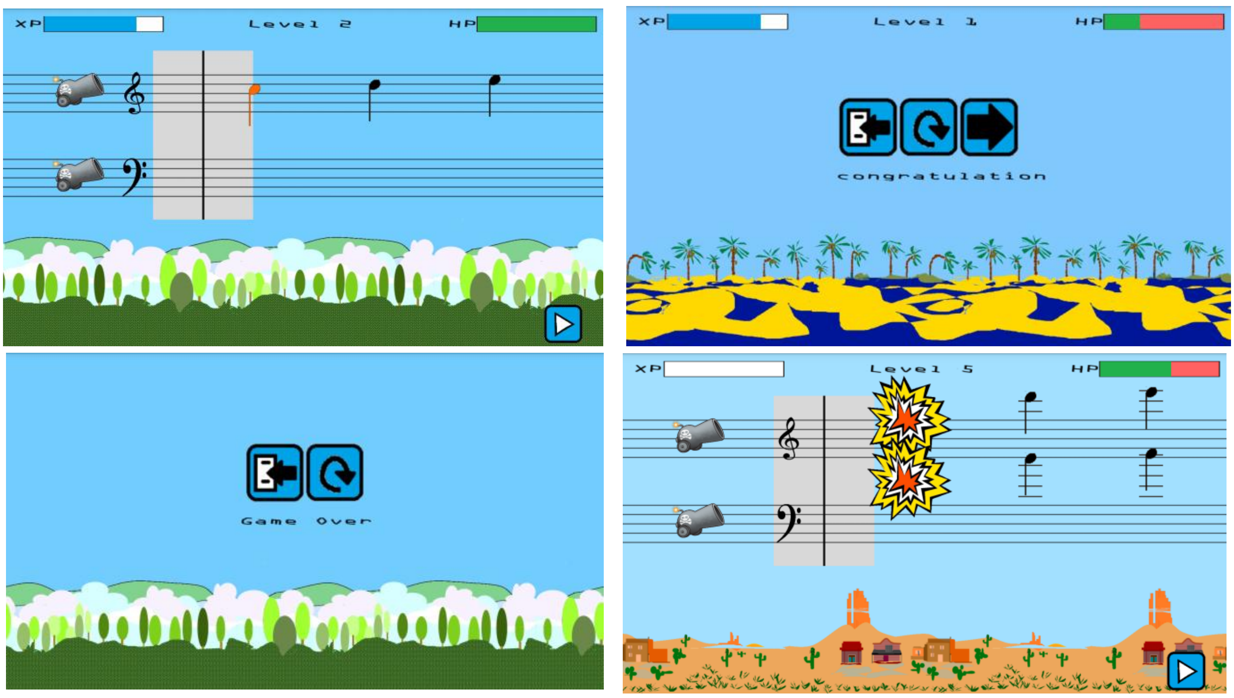
This page will be filled out by the experimenter.

1. What do the following items stand for:

correct	incorrect	
		Upper 5 lines
		Lower 5 lines
		Notes coming from the right
		Orange note
		Green circles from the left
		Red circles from the left
		X popping up in the middle of the window from time to time
		XP-bar top left
		HP-bar top right
		Level number in top middle
		Canon
		Left Item in congratulation screen
		Middle item in congratulation screen
		Right item in congratulation screen
		Grey bar

2. Answer the following questions

correct	incorrect	
		When do you get to the congratulation screen?
		When do you get to the game over screen?
		What are the points you can earn for completing a level
		What can they be used for
		Points in the top left corner of the MainActivity
		When is it possible to buy an extra
		When is it not possible to buy an extra (how is that indicated)
		Where do we find the extra levels we already bought



3. Any tips what could be improved:

Concerning graphics:

Concerning game logic:

Is something missing that would improve the game in your opinion:

Is something unneeded and the absence of this element would improve the game in your opinion:

C

Tables for Notes

A complete overview of all notes available on the piano with their wavelengths, frequency, name and half tone number

C.1. Notes on the Piano, the complete Table

Table C.1.: *table containing notes, frequency, wavelength and semitone number*

Note	Frequency (Hz)	Wavelength (cm)	Semitone Number
A1	55	627	33
B1	58.27	592	34
H1	61.74	559	35
C2	65.41	527	36
Cis2	69.3	498	37
D2	73.43	470	38
Dis2	77.78	444	39
E2	82.41	419	40
F2	87.31	395	41
Fis2	92.5	373	42
G2	98	352	43
Gis2	103.83	332	44
A2	110	314	45

continued on next page

Table C.1.: *(Continued)*

Note	Frequency (Hz)	Wavelength (cm)	Semitone Number
B2	116.54	296	46
H2	123.47	279	47
C3	130.81	264	48
Cis3	138.59	249	49
D3	146.83	235	50
Dis3	155.56	222	51
E3	164.81	209	52
F3	174.61	198	53
Fis3	185	186	54
G3	196	176	55
Gis3	207.65	166	56
A3	220	157	57
B3	233.08	148	58
H3	246.94	140	59
C4	261.63	132	60
Cis4	277.18	124	61
D4	293.66	117	62
Dis4	311.13	111	63
E4	329.63	105	64
F4	349.23	98.8	65
Fis4	369.99	93.2	66
G4	392	88	67
Gis4	415.3	83.1	68
A4	440	78.4	69
B4	466.16	74	70
H4	493.88	69.9	71
C5	523.25	65.9	72
Cis5	554.37	62.2	73
D5	587.33	58.7	74
Dis5	622.25	55.4	75
E5	659.26	52.3	76
F5	698.46	49.4	77
Fis5	739.99	46.6	78
G5	783.99	44	79
Gis5	830.61	41.5	80
A5	880	39.2	81
B5	932.33	37	82
H5	987.77	34.9	83
C6	1046.5	33	84
Cis6	1108.73	31.1	85
D6	1174.66	29.4	86

continued on next page

Table C.1.: *(Continued)*

Note	Frequency (Hz)	Wavelength (cm)	Semitone Number
Dis6	1244.51	27.7	87
E6	1318.51	26.2	88
F6	1396.91	24.7	89
Fis6	1479.98	23.3	90
G6	1567.98	22	91
Gis6	1661.22	20.8	92
A6	1760	19.6	93
B6	1864.66	18.5	94
H6	1975.53	17.5	95
C7	2093	16.5	96
Cis7	2217.46	15.6	97
D7	2349.32	14.7	98
Dis7	2489.02	13.9	99
E7	2637.02	13.1	100
F7	2793.83	12.3	101
Fis7	2959.96	11.7	102
G7	3135.96	11	103
Gis7	3322.44	10.4	104
A7	3520	9.8	105
B7	3729.31	9.3	106
H7	3951.07	8.7	107
C8	4186.01	8.2	108
Cis8	4434.92	7.8	109

Bibliography

ANDROID, Feb. 2014. http://developer.android.com/design/get-started/principles.html.

BELOWXPECTATION, Feb. 2014. http://www.belowxpectation.com/glop/av/ipn.gif.

FEDE, D. D., Oct. 2013. http://libgdx.badlogicgames.com/.

GITARREN-KIDS, Feb. 2014. http://www.gitarren-kids.de/img/show-noten-und-pausen-tabelle.gif.gif.

INDIANA-UNIVERSITY, Mar. 2014. http://www.indiana.edu/ emusic/acoustics/sound.htm.

KLAPURI, VIRTANAN, H., 2000. Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals.

LOCKTON, HARRISON, S., 2009. The design with intent method: A design tool for influencing user behaviour.

MEDINA, D. J., 2009. Brain rules: 12 principles for surviving and thriving at work, home, and school.

PINOLA, M., Mar. 2014. http://www.itbusiness.ca/news/history-of-voice-recognition-from-audrey-to-siri/15008.

R.BROWN, A., Feb. 2014. http://explodingart.com/jmusic/.

SAMSUNG, Feb. 2014. http://www.samsung.com/de/consumer/mobile-device/mobilephones/smartphones/gt-i9505zkadbt-spec.

SHI, Z., 2011. Decay characteristics of piano tones.

WIKIPEDIA, Feb. 2014. http://commons.wikimedia.org/wiki/file:moodswingerscale.svg.

WIKIPEDIA, Mar. 2014. http://en.wikipedia.org/wiki/harmonic.