The Thereminic Room Interface

Interfacing a Room sized Instrument for Multiple Players

Alireza Mahzoon
Abstract

The Thereminic Room Interface

Alireza Mahzoon

A Theremin is essentially an electronic synthesizer, which reacts to the distance between a hand and an antenna. The music is created from an internal oscillator which reacts to an induction field, when it is disturbed by the layer.

In the Thereminic room, small groups of pupils with serious impairments will be able to create music expressions together, by interacting in various ways with the different antennas and the accelerometer. In this way the participants will be inspired to move in rhythms and patterns, thus promoting physical exercise for the pupils involved.

In what follows two ideas are presented, interfacing the room with the antennas and defining appropriate variables for the system based on the players' movements, designing an interface to help teachers to create willing sound by setting variables in the system in the room. The concepts are proposed based on an existing system.

The outcome of the thesis is a design solution to assign proper note values to a distance which make it possible to play different set scales based on the players' movement in the room, in addition, the suggested idiomatic user interface design could map the configuration of the four antennas and their activities to the appropriate Musical Instrument Digital Interface (MIDI) signals for the software synthesizer.
8 Discussion and Future work

8.1 Variable Based Design ............................................. 35
8.2 Design for iPad ..................................................... 37
8.3 Improving the User Experience ............................. 38

9 Conclusion ......................................................... 39
1 Introduction

In this thesis, I will describe the development of a user interface to an artifact being developed in a project under the name “Thereminic rooms”, at the department of Information Technology at Uppsala University. The thesis describes one part of the project, namely the design and implementation of the user interface. The other part of the project, including the physical and software implementation (apart from the user interface) of the Thereminic rooms has been published in a parallel thesis by Rahmanur Hermanawan: Dancing with Theremins [18].

1.1 Background

Over time, many efforts have been made to help disabled and impaired people managing their everyday life, such as building special ramps, designing doorways and restrooms so as to accommodate wheelchairs and providing Braille signs. However, there are still many challenges left to address. One general tendency is that people with impairments tend to lose their motivation to participate in activities due to all the remaining barriers. There are now ongoing projects in special and training schools to include children with even severe impairments to get engaged in different activities, both physical and cultural. One of the challenges of the above mentioned projects is to motivate pupils with impairments having a low desire and motivation to do physical activities. One possibility is to interleave physical activities with other activities that promotes participation. One such motivating factor can be to provide means for actively playing music, which has been shown in a music project at Årsta special training school. If it were possible to use music as a motivating factor for an embodied interaction with an instrument, this could be one way of activating children with physical impairments to participate in physical activities.

One of the devices used at the Årsta school has proven itself to be of great interest in this respect, is the Theremin, see figure 1. The original Theremin was invented in 1917 by the Russian musician Leonard Theremin, and it is one of the earliest electronic instruments constructed. The instrument has been later been acquired by the Moog company, which recently released a modified version, the Theremini. One major addition to the instrument was a pitch control that makes it much easier to play than the original instrument. Some may consider this “cheating”, but as a matter of fact, this simple change has made it perfectly suitable for a music project at the Årsta special training school in Uppsala. The Theremin and the Theremini are both played without being touched physically. Essentially, it is an electronic synthesizer, which reacts to the distance between a hand and an antenna. The music is created from an internal oscillator which reacts to an induction field as it is disturbed by an inductive element (such as an arm or a finger) moving within the induction field. In the MUMIn-project (Maximal Output - Minimal Interaction) at Årsta special training school, in Uppsala [4] a Theremin has been used among other special instruments as a means for allowing children with very severe impairments to play music, despite their impairments.

The use of the Theremini at the training school was very successful, and in a followup to the MUMIn-project there are now plans for the construction of a room-sized version of this instrument, which is based on the conceptual idea
in the Theremin. The project, which also serves as the research context within which this thesis is being written, is called the “Thereminic room”. In the Thereminic room, small groups of pupils with more or less serious impairments will be able to create music expressions together, by interacting in various ways with the different antennas and the accelerometer. In this way, the participants will be inspired to move in rhythms and patterns, thus promoting physical exercise for the pupils involved.

In a related Master thesis [18], the technical implementation of the instrument has been carried out, but there still remains a problem for the intended users – music therapists, music teachers, etc. – to manage the configuration of the instrument in a way that is uncomplicated but still provides full control over the settings.

Since induction fields follow a logarithmic scale when it comes to intensity vs. distance, a room-sized Theremin would require very strong fields in order to reach all over the area, and it was therefore decided to use ultrasound sensors, that are distributed on four separate antennae. Furthermore, the ultrasound sensors are also coordinated with a set of wearable accelerometers. Altogether, the Thereminic room can be used by several users at the same time, collaborating in the task of creating music through embodied interaction.

The technical aspects of the construction of the physical ultrasound antennas are addressed in a parallel thesis by Rhaman Hermanawan: Dancing with Theremins [18] and the thesis project described here is addressing creation of a user interface for the teachers, allowing them to easily configure and manage this complex instrument. The implementation of a proper user interface will be a key factor in establishing the usability of the instrument. The intention behind all this is that the teacher (user) shall be able to combine various synthesizer sounds and effects and map these to different parts of the Thereminic room components. How to apply this intention of the teacher in a simple and transparent way constitutes the challenge for this thesis.

1.2 Purpose and Research Question

Historically, there are three main dominant paradigms for user interface design: implementation centric, metaphoric and idiomatic [15].

- To be able to use an implementation centric interface, the user first needs
to know how the actual system works internally and only then the user can work with the interface, the focus of design is on understanding how things work in the system. Based on this, learning how to handle it.

- Metaphoric interfaces are based on intuition, where the interface is a replication of the real world and there are visual metaphors in the interface that represent the functions, so the user can understand the metaphors’ imagery. This would alleviate the user from a too high barrier of learning, while still adhering to the aspects of the real world. In idiomatic interfaces there are no metaphorical visual or behavioral idioms to guide the user, they are based solely on learning. The user needs to interact with the interface to learn how to use the interface and the usability depends largely to ease of learning and remembering.

Most of the interfaces we see around us today are more idiomatic and it is prominent that the user learn easily how to work with them. However, most music applications use metaphoric interfaces for music creation to match the musical instrument in the real world. It is therefore interesting to study the situation in the user interface of the therminic room. Here we can not use the same method since we do not generate music by regular music instruments. The concept which we consider here to explore the usability of the interface is the term affordance. It is explained by Donald Norman in his book “the design of everyday things” as “a relationship between the properties of an object and the capabilities of the agent that determine just how the object could possibly be used” [23].

The main research questions discussed in this thesis are:

1. How we can design a highly usable interface based on the variables that the computer receives from the antenna to fulfill expectations of affordances?

In addition: The meaning of the term “interface” in the therminic room is not limited to the API (Application Programming Interface), but the stage and players’ movements are part of the interface. This suggest a secondary research question to be answered in this thesis:

2. How can the player’s minimum interaction with the antennas in the room be defined in relation with the variables to create music? In the other words, what movements in the room can be matched to the variables to generate music?

1.3 Delimitations

The end product is the manifestation of a proof of concept, which incorporates as much of the intended functionality as possible, and provides a finalized interface to the Arduino-based electronic instrument. Essentially, the combined instrument should be possible to play in a real setting, with minor adjustments, using the software that is presented here. Part of the work could be included as an evaluation of the interface by representatives from the intended target user group (special teachers).
1.4 Thesis Structure

In chapter one, the general background for the thesis idea is described. In chapter two, the technical background, the products which are converting body movement to sound are introduces. In chapter three the component of the Theremin systems and also the Musical Instrument Digital Interface (MIDI) protocol are described. Some design principles which influenced the first ideas behind the prototyping are declared in the fourth chapter. In chapter five and six, the method used for design and a conceptual design of the user interface is illustrated. In chapter seven, the conceptual solution for designing a complete virtual room as a metaphoric representation that connects the user interface to the players’ movements is proposed. There are discussions about the other solutions for design, design ideas for iPad and user experience perspective in chapter eight. The final chapter is the conclusion.

2 Technical Background

There are several different technologies that can be used to convert body movement to sound and music. In this chapter we will present some of these technologies as a background to the later discussions in the thesis.

Software like Kagura [3] and Nagualsound [5] use 3D-motion tracking technology to convert the body movement to the music. In addition there are products which use motion and position sensors to capture the body movement.

Soundbeam is a product which uses ultrasound sensors to give the users the possibility to play music regardless of their impairments. It has two prominent criteria, first one is the sensitivity of the beam. It means that even the most unreachable or immobilised individuals can ‘play’ independently; secondly, electronic technology makes available a huge palette of possible soundworlds, releasing the player from the traditional limitations of percussion-based activity [8]. see figure 2.

Figure 2: The figure shows Soundbeam, it can have up to 4 sensors attached, though the most popular choice to start with is two [9].

“The first prototype of Soundbeam was built in 1984, and by 1989 the machine was being produced commercially by Cornwall-based Electronic Music Studios (EMS), developers of the VCS3 (Voltage Controlled Studio) – the first
synthesiser to be widely used in schools and universities, and by the more adventurous rock groups” [9].

It is also possible to translate body movement to music and sound by the use of infrared sensors. Dodecaudion [10] is a gesture based instrument which uses infrared sensors to capture the body movement and generate music. The shape of product is like dodecahedron. see figure 3.

Figure 3: Dodecaudion has 12 infrared sensors and has been designed using a dodecahedron shape [10].

The product uses infrared sensors, Arduino, Bluetooth, processing, OSC (Open Sound Control) and a synthesiser to generate music [1]. “The idea is to further the creation of new means of expression in electronic live music”. Jakub Koźniewski who talks about the product in TEDx (Technology, Entertainment, Design) conference believes that the regular devices which are used as musical controllers are not compelling in terms of gestural expression, but with Dodecaudion player can use the whole body and even go behind the controller [2].

Remidi T8 is another product which turns the whole human hand to a music instrument. It is consist of a glove and wristband. There are five pressure sensor pads on the fingertips and three sensors on the palm. The wristband is the brain of the T8, it records the signals from the glove sensors and tracks the motion of the hand, then it sends the information over Bluetooth MIDI. Motion Tracking and gesture recognition allow control of different parameters: delay, glide, filter, release, etc. Remidi ios app make it possible for the player to remix and perform music then share it in different platforms [6].

The Thereminic room is different from the examples shown in this section, in that it does not require anyone to carry an instrument (possibly a little similar to the Soundbeam, but it is still extended to larger areas). In this chapter the layout of the whole thereminic room system is presented, although without any detailed description of the system components in order to keep the focus on the system and user interface, not the technical aspects.

3 The Technological Setup

In order to understand the design problems addressed in this thesis, it is necessary to get a general grasp of the underlying technology in some detail. The design of the basic hardware/software instrument is specific to this project and
is the conceptual base which we will have to relate to in our design. Therefore, we will describe the technical background to the current implementation of the Thereminic room in this chapter.

3.1 General Overview

The system in a Thereminic room incorporates several different physical hardware components that are used to detect and communicate various different body movements via a series of microcontrollers to a software synthesizer (on a regular computer) and produce sound. In the prototypical Thereminic room, there is a total of four stationary antennas which are provided with four ultrasound sensors each of the antennas are placed in a specific pattern in order to get data from the body movement and distance of one or more simultaneous users. For the structural description of the hardware I will refer to the scheme in figure 5. The ultrasound sensors are programmed to detect both distance and angular movement in relation to each antenna. There is, apart from the “antennas” also one other set of motion sensors that detects the acceleration and position of the limbs of a person. The antennas and the motion sensors will transmit data in a raw form to the Base Station (BS) where the data is either translated by a separate microcontroller (Arduino-based) into the MIDI format, or transferred straight to a separate software in the computer that relates the raw data to a larger set of MIDI signals, which can be modified dynamically by the master user, which will in most cases be a teacher. In this thesis we will now focus on the conceptualization of this API in the personal computer (PC) part (between the controllers and the synthesizer). The standard setup in the bare-bones system is static, and any change to the produced MIDI code needs a reprogramming of the central Arduino unit.

The user interface presented here is thus filling the gap between the raw data coming from the antenna and the output to the synthesiser software. The interface is implemented in order to make it simpler (or even just possible) for the teachers to interact more easily with the system. More details about the user choices in the interface are explained in the Design section. The various antennas K, L, M, N are communicating with the Base Station over the XBee protocol, which allows for a larger distance connection than for example Bluetooth. The Base Station coordinates the raw distance data from the antennas (K, L, M, and N) and generates a sequence of MIDI signals (but which might not necessarily
have been transformed into music scores yet) which is then sent over a USB (Universal Serial Bus) connection to the PC. On the PC, the interface software will allow the teachers to translate the signals from the antennas into the right MIDI signals for use in a MIDI-capable synthesizer. In the following sections we will describe the different parts in the technical setup: the antennas, the communication protocol and the hardware used on the antennas.

Figure 5: An overview of the hardware and software setup for the Thereminic room [18]

3.2 Antenna

There are three main parts in the hardware of each antenna: a microcontroller ATMega328p as a central processor to calculate the distance, an XBee module that accommodating ZigBee protocol as communication device and ultrasound sensors to detect distances. Details about distance calculation is discussed in the Dancing with Theremins thesis [18]. On the hardware boxes, see figure 7, in each antenna there are two Lighting Entity Diodes (LED)s and one potentiometer. One LED is notifying if the user is in the detection range or not, the other LED is to show XBee communication status and the potentiometer can change the maximum distance detection of the antenna.

In the software part, FreeRTOS (Real Time Operating System) is implemented. It is a real time operating system kernel and suitable for using in the embedded systems with microprocessor or microcontroller. There is Arduino FreeRTOS library which is compatible with several Arduino boards and libraries to manage ultrasound sensors and ZigBee communication via XBee. Two tasks are designed in the antenna software, first task Blinked to check if the user is out of range or not, second, task distance to send the detected distance to the Base Station. Antenna L and N not only detect the distance, but also detect the user position against the antenna.

In figures 6, 7 and 8 the antenna hardware with ultrasound sensors, as well as the antenna wireless communication and Base Station are shown respectively. The wireless communications through the antenna are used for transmitting data to the Base Station. Note in figure 6, that the ultrasound sensors are mounted in a 90 circular fashion on the antenna base. This makes it possible to get not only the distance readings but also the angular direction of the person’s position. In this way, the functionality of the antennas becomes more versatile.
than on the original Theremin itself, in that they can also detect the direction of the player.

Figure 6: Antenna hardware with ultrasound sensors

Figure 7: Antenna wireless station for transmitting data to the Base Station. The XBee module is located under the green square in the picture

Figure 8: Base Station. Note the row of LED-indicators on the top left, used to display the transmission status from the antennas to the Base Station
The antennas on the pictures are from the first physical prototypes, made out of cardboard tubes, duct tape, and similar simple materials. The final product is expected to have a much sleeker design with more physical stability.

3.3 The ZigBee Protocol

For the communication between antennas and Base Station, it was chosen to use the ZigBee wireless protocol over XBee antennas. This communication protocol is based on the IEEE 802.15.4 protocol specifications that have been applied to make low rate network. It is used when low power or low bandwidth is needed, it is also suitable for personal area networks. IEEE 802.15.4 is a standard to define low-rate wireless personal area networks functionality. ZigBee does have some restrictions in capacity, but is overall a very capable protocol, which is implemented by the XBee modules mounted on top of the control boxes, see figures 7 and 8 [26].

3.4 The Arduino Microcontroller

A microcontroller is a small central processing unit which has a high potential to control and monitor the sensors, the distance calculations and also the communications over the ZigBee network. It is used for embedded applications and communication with other peripherals that exist in the same chip. Typical examples of popular microcontrollers are Arduino and its various cousins, as well as smaller versions of these. They have a high capacity for calculation (the programs are written in C++) and do also incorporate the functionality of converting analog signals to digital (through programming) and vice versa. This also makes them very suitable for the construction of the antennas here, since the ultrasound sensors (see below) only produce analog voltage-variations, which need to be decoded into a binary code, which is then further translated into MIDI signals. This translation capacity becomes even more useful in the context of this project, since the MIDI library provides the essential parts of the general communication with the computer as well.

3.5 Ultrasound Sensors

An ultrasound sensor is essentially a combination of a special loudspeaker and a sensitive microphone, where the main difference from a normal stereo system is that the sounds are in the ultrasound frequency interval. The sounds are therefore not possible for a human to detect (or be irritated by) and the sensors will in most cases not be affected by other, potentially disturbing, sound sources either. The sensor uses ultrasonic waves to measure the distance to objects. There are two transducers (speaker and microphone) in each sensor, and the device is sending a sound wave to the object and listening for that exact same sound wave to bounce back, so it can calculate the distance between the sensor and the object using the speed of sounds as a measuring stick. These transducers are for example, used in robotics and automation for different applications such as detecting the position of an object, measuring the distance between the sensor and an object, counting the objects which pass through the front of the sensor and tracking an object. In the current system, the sensors are used to detect both the position and the distance of the player relative to the antennas. Figure
9 illustrates an ultrasound sensor. Each sensor has an effective reach distance between five centimeters and four meters. However, the precision of the distance detection at both ends of this spectrum is lower than for the in between distances [11].

![Image of ultrasound sensor](image)

Figure 9: Ultrasound sensor

### 3.6 MIDI

The acronym MIDI stands for *Musical Instrument Digital Interface* which is a standard protocol that makes the communication between different electronic music instruments and computers possible. “MIDI can be divided into three separate entities: the language it uses, also known as its protocol, the hardware interface it uses to transmit and receive its information (such as connectors and wires); and its distributions formats, such as Standard MIDI Files (SMF)” [17]. The MIDI signals are sent over a serial interface, and thus require very little bandwidth, but instead a high communication speed. The MIDI protocol is an old communication protocol and it has only been changed a few times since its introduction in the 1980s. However, despite this most music hardware and software still use MIDI as an interface language between the instruments. MIDI also has the advantage that it can easily be decoded in a human-readable form.

#### 3.6.1 MIDI Channels

Channel usage in MIDI is like what we have on television or radio. It is basically for isolating part of the information. Each message has a channel number, so the synthesizer will analyse the events on the basis of the channel number and the audio will be produced corresponding to the notes on the selected channel.

#### 3.6.2 Note On and Note Off

The format of MIDI messages is binary values. The messages for note playing are categorized as note off and note on which are referring to the musicians performance for each single note. If we take the action of a musician when playing a musical instrument like turning on and off a switch, note off and on have the same meaning. Actions, like pulling a bow, pluck a string, blow air or press a key are considered as a note on in MIDI and stopping the actions will be the note off. see figure 10. This way of representing the activity also means that some MIDI sequences need to be balanced. A note on code needs to be followed by a note off signal, since otherwise the current tone will be played continuously.
until the synthesizer receives a note off, or gets shut down. To make this in a
transparent way is partly a responsibility of the user interface. In the case of
the Thereminic room, there is an intricate problem to decide when to send the
note off signal. There are three major alternatives:

1. Send a note off, whenever the tone value changes. This allows for a legato
type of playing, where notes can be shorter or longer, depending on when
the user moves.

2. Send a note off at regular intervals, e.g., the notes will be more or less
the same length, which could be easier to handle, but might give a more
monotone performance.

3. The note off and on could be triggered by the motion sensors, essentially
giving one person the role of a conductor. The controller base for this
version has not yet been implemented in the physical antenna, so it is a
remaining case to implement.

The available alternatives should also be possible to configure over the user
interface, and the setting of these options as well adds to the complexity of the
interface, see figure 10.

![Figure 10: Note on and note off in MIDI messages](image)

### 3.6.3 MIDI and Other New Communication Instructions

There is MIDI in, MIDI thru and MIDI out in the devices which are using MIDI.
MIDI in is the input where the data is coming to the device and MIDI out is for
sending out the message from the device or software which it is generated from.
MIDI thru is used for chain devices when we want to have the same MIDI in
the output to be transmitted as an input to another device. Hence, MIDI thru
is used for this transmission instead of MIDI out.

### 3.6.4 Data Messages in MIDI

MIDI messages are divided into two categories: status and data. The status
byte of the MIDI shows the information about the event, and it identifies the
channel of the event. It also shows what kind of event that is being transferred.
An event can be note off or note on, for example, a status byte can tell that
an event is pressing keyboard and belongs to channel x. The data byte shows
the value of a message, for example when a key is pressed, the status byte
would identify the event which is pressing the key. The data byte tells the value
number of the key and velocity level of hitting the key. “The status byte uses
numbers ranging from 128 to 255 (1000 0000 to 1111 1111 in binary numbers),
and the data byte uses numbers ranging from 0 to 127 (0000 0000 to 0111 1111
in binary numbers)” [17]. Figure 11 graphically shows the order and content of
data byte and status byte in MIDI messages. The function is called Polyphonic
Aftertouch, it is identified in the first byte. The second byte shows the value of
the function which is a pressed key. The level of this function is determined in
the third byte, it is 95. The encoding of the notes may seem complex, however,
the notation remains the same, regardless of how complex music you want to
write.

Figure 11: Data byte and status byte in MIDI messages [17]

4 Defining an Interface for Creating Music

The meaning of interface in therminic room is not limited to the software user
interface but the stage is part of the interface for music creation. The movements
and gestures in the stage are part of the process for music generation. In this
chapter a background related to generating music by using data from players’
movement is discussed as well as some principles for designing user interface.

4.1 Electronic Systems Depending on Movement

“The world caught up with Leon Theremin in the 1960s and 1970s when several
composers rediscovered the exploration of movement to create electronic mu-
sic” [27]. In the Experimental Music book by Michael Nyman [24], the chapter
named Electronic Systems explores some works which created by composers after they were used electronics into experimental music, it was not just adding new equipments for the performance but “inventing and adapting a portable electronic technology which was easily accepted into the ever-open world of performance indeterminacy”. One of the examples of mentioned electronic systems is the one which is used in Cage’s variation v(1965).

Figure 12: Figure Cage’s Variations V, a collaborative work featuring music by John Cage, choreography by Merce Cunningham and Gordon Mumma [24].

In this sound system creating music were more dependent on the dancer’s movement rather than manipulation of musicians. “The entire floor was transformed into a musical instrument responsive to movement throughout the space” [24]. Two sets of electronic sensors are implemented in the system, first is a set of focused photocells which the dancers movement interrupts the falling lights on them, the second is a group of five-foot high antennas that respond to the distance of the dancers from each other, to the proximity of the dancers from the antennas, and to the number of dancers on the stage [24].

One part of the work is choreography for designing movement in the room and the other part is sound production. So in the whole process there is a need for a choreographer to designing sequence of movements of bodies and a sound engineer for balancing and adjusting sound sources. The novel idea in therminic room is that by designing a user interface the teacher in the room can create music by manipulating variables, setting effects or using presets. In the following sections in this chapter some general user design principles are discussed.

4.2 User Interface Design Principles

The final prototype is essentially a single-page idiomatic interface. In this section, application of the mental model concept in design, the idea of using conventions and the reason for having the single-page interface is discussed.

4.2.1 Mental Models

One of the principles of user interface design is to focus on user’s mental model rather than what is called implementation model [15]. The implementation
model is derived mainly from how the application has been programmed in the implemented code. For developers, it could, e.g., be logical to have a button for every function or have different dialogue boxes for setting the translation of the MIDI codes, but for the users this might instead be completely confusing. Instead, users can be expected to prefer to follow an intuitive and task-based process to accomplish the goal of the task. One good example of an interface based on the mental models is what we see in the design of Adobe Photoshop Express Software. In this software, the user does not need to manipulate the numerical values, and the edited image is shown as one of a number of thumbnails showing alternative settings that the user can pick from, see Figure 13.

However, ease of use can often result in a loss of granularity in the settings. This is a compromise that might be difficult in many cases. In Adobe Photoshop, users have to work with hue and saturation or alternatively change other numbers to see the wanted visual effects in the image. When we design these kinds of interfaces it is still necessary to provide alternative means in order to circumvent the higher granularity. A user might gradually become so proficient in his or her skills that the given granularity is experienced to be too crude to achieve the desired goal. This means to design for personal development.

4.2.2 Conventions in Interface

According to Steve Krug, in his book “Do Not Make Me Think”, one of the ways to make the user understand the design is to use existing conventions and take advantage of what users already know [19]. The design which is used to catch drivers’ attention in stressful situations is an example that follows existing conventions. Standardized stop signs in the streets with the same pattern or the location spots of gas and brake pedals in cars are employing the same concept. The gas is to the right and the brake pedal is in the left side. There are also the same type of convention-based design in web design. For example, the navigation controls are at the top of all web pages and the users also expect to see a search box, site ID and other common utilities at the top of the web page. Some icons like the image of a shopping cart is an example of a convention in

Figure 13: Adobe Photoshop Express for iPad, an example of software design to match user mental models [15]. At the bottom we see a row of sample adjustments that have been selected to provide the most common changes to a picture.
online stores to guide you to see what you have bought.

Most conventions that exist are created from habit and generalisation of behaviours. Some of the conventions have also been developed for purely practical reasons, such as that most exit doors open outwards, so that in case of emergency it will become easier to escape. On the other hand, most banks have doors that open inwards, as a means to make it more difficult for bank robbers to leave the bank quickly.

This also means that the use of conventions can be a powerful construct in the creation of an intentional design, where all design decisions are made explicitly.

4.2.3 Complex Usable Interface

Simplicity in the design is one of the criteria which strengthens transparency and usefulness, but in interacting with a user interface, it is also important to think about the user’s memory load, stress levels and even the way in which he or she interacts with the artifact. By adhering to the principles of simplicity [20], it is relatively easy to ease the load on the user during the work. For example, if the user needs to enter the same information in different dialogues, it is easy both to forget the information and to make the user collect the information over and over again, since this kind of information is normally only stored temporarily in the user’s working memory. The recurring entry of information is also increasing the risk of entering conflicting information by mistake.

Designing a system interface with one single page which collects all the information that the user needs is one possible way to help the user completing the tasks efficiently. If an interface needs to be complex due to a necessity of displaying a large amount of information it can still fulfill the definition of International Standardization Organization (ISO) 9241 [13] and usefulness [22] since they are regularly being used by professionals. However, even when used by professionals, there is a need to support the user so that he or she does not miss or misinterpret important information. There are many examples of problems and even disasters that occur due to small signs or signals being overlooked.

Possible means of support are the display of structures, the organisation of data on the screen and the support for observing patterns in the complex collection of information. Calibrated scales are one such device, implying that the dials on a control panel are calibrated in such a way that they all point in the same direction during normal circumstances. As soon as a measure deviates, the corresponding dial will immediately be easy to observe. Some of these mechanisms have been devised in the design of this software.
5 Method

In this chapter the background behind the user interface design is detailed and the process of prototyping the interface is explained. The very first ideas for the interface design and the limitation in the hardware part which effected on the design elements in the interface is declared. The sketch and prototype in this level are using metaphors.

5.1 Metaphoric vs. Idiomatic Interfaces

As it mentioned in the introducing chapter, the paradigms of user interface design can be divided into implementation centric interfaces, metaphoric interfaces (also known as skeumorphic interfaces) and idiomatic interfaces [14]. However, nowadays most interfaces are idiomatic. There are exceptions, for example within music creation software which are still remaining metaphoric. In this section I will give two examples of metaphoric and idiomatic interfaces for music creation.

Metaphoric interfaces use real world connections to visualize different functions on the interface, for instance tiny pair of scissors on button indicating cut. We grasp the meaning of the metaphoric elements of an interface because we mentally connect them with other things we have previously learned but the problem with metaphoric interfaces is that all metaphors do not scale well and sometimes it is even hard to find metaphors from real world for the functions in the interface. In addition, users do not always have the same cultural background which makes it difficult to understand the metaphors. Figure sunrizer shows an example of metaphoric interface. Here knobs are used for setting some parameters. However, from an interaction perspective the knobs are very difficult to use, and should probably be better shaped as sliders.

Figure 14: Sunrizer is an iPad synthesizer that resembles a hardware original. The interaction with the software is also designed to be similar to the interaction with the original hardware.

One other problem with metaphoric interfaces is that are often designed to be completely truthful to the original, and not implementing the best interaction styles for the new medium. The knobs are once again good examples of this problem. Moving the mouse or touching the screen to operate a knob is much less sensitive than a physical knob. Most prominent is the disappearance of
muscle memory, that develops quickly with the physical controls, but is lacking in the computer interface [25].

Idiomatic interfaces instead focus on the “learning of simple non-metaphorical visual and behavioral idioms to accomplish goals and tasks”. Samples of visual idioms in graphical user interfaces are in operation systems like, close boxes, title bars, screen splitters, hyperlinks we need to learn about them. TC11 is a sample of application which has idiomatic interface [15].

Figure 15: Two screenshots of the TC-11 performance interaction view [12]

“Instead on relying on knobs or faders to control the sonic output, users are able to use multi-touch gestures, as well as the iPad’s accelerometer, gyroscope, and compass to control the app. When using the multi-touch components of the app, they see a variety of geometric shapes that correspond to their interactions. Users are also able to record their interactions” [12]. “The user must learn by exploring the tonal and visual effects created by touching and gesturing” [15]. However, this also means that the interface is less intuitive than the metaphoric version, whereas the interaction with the controls is better adjusted to the interface quality and properties.

5.2 Initial Sketches

The system performance is sending the antennas data to the synthesizer. Looking at the system in a simplest way, it is like a switchboard that on one side there is antennas and on the other side there is synthesizer. The user need to connect this two sides and make a desired sound. see figure 16 for our initial sketches of this structure.

The functionality can also be associated with a mixer with various channels that allows the sound engineers to create desired mix. In the input there is stream of different values and the output is the stream in MIDI format. see figure 17.
Figure 16: The left box is layout of the room and the right box is the sketch of the first idea behind interface

Figure 17: Showing the functionality of the interface in the sketch
5.3 Prototyping Ideas

Using Adobe Photoshop software, Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), Javascript and jQuery some of the design ideas implemented to have better understanding of the functionality of the interface in the web browser. The design ideas in the following figures are based on the antennas that send two variables, angle and distance. The user can set the variables in the the main page, see figure 18. The user set the angle and distance between a person and antenna in the top left box. The top right box shows the room layout. The players’ angel and distance to all antennas will be shown in bottom boxes. The user need to choose the person number in the combo box to see the information. The user can see the angels to antennas in bottom left box and the distances to antennas in bottom right box.

![Figure 18: Entering numbers or changing knobs to set variables in the main page](image)

In addition, there are different sensors on the antennas that can be chosen by the user. The user can set the effects on the sensors as well. see figure 19. Antenna one is chosen and sensor one on the antenna is selected. In this scenario the user picked an effect in the bottom left box. The value of the effect can be set in the bottom left box.

The antennas in the project did not have the capability of using different sensors or setting them to cover different angels. The options were limited, so the final design is different from the sketches. The design made in Adobe XD and it is presented in chapter 6. For implementing the user interface on the hardware Tkinter library which is a Graphic User Interface (GUI) module for Python programming language will be used.
6 Design

In section 6.1, the data which will be received by computer from the antenna is presented. In section 6.2, different parts of the interface are shown separately. To show the functionality of the interface, the task of defining an effect is illustrated in section 6.3.

6.1 Design Requirements

There are three types of data that are sent to the PC. L and N antennas send numbers that represent the position of the player in relation to the antenna. The numbers are 900 for the center, 1000 for the right and 800 for the left position. K and M antennas are sending the distance value of the player from the antenna. The sensors take a value between 0 and 400 cm. In addition, there are two accelerator sensors that send b and c. Once the function takes the data, in MIDI out there are some effects based on the data that the PC is receiving.

The interface should give the user the possibility to define the variables for each antenna, seeing the values in antennas separately and can see the information from other antennas. The user also needs to know what combination of data will produce the effect.

6.2 Design Rationale

The prototype is designed as one main page and all the information and all functions are accessible from the main page. The main task for the user is to choose an antenna and select the player’s position and values for the maximum distance to the antenna. As mentioned, there are two antennas that send the
distance values and two antennas that send the position data. The main interface consists of a navigation board and six boxes. The process for setting up the antennas has been implemented as an “open wizard” type of sequencing. This means that the user has to follow a certain path forward through the setup, but will be able to change previous decisions with random access, i.e., he or she does not have to reverse the path once the selection is made. The user starts the task by choosing the room setting in the top left box.

The user configures the room setting from a combobox which is located in the top left box in the interface. The combobox provides two options to the user. The antennas can be located either in the center of the room or in the respective corners. In the next step, the user chooses the antenna and afterward, he/she adjusts the position and distance data for the antennas by interacting with “Position” and “Distance” boxes, respectively.

After choosing the room settings, the “Choose Antenna” box will be activated. The user activates the relevant settings for an antenna by clicking on the antennas’ icons in the “Choose Antenna” box. The antennas can be selected one by one. The “Choose Position” box will be activated by the selection of either L or N antenna. Similarly, the “Choose Distance” box will be activated by choosing either K or M antenna.

The selected data will be recorded in an “Overview of the Antennas Data” box. As a result, the whole chosen setting appears in the antenna data in the box. The effects which are produced by distance and position data will be highlighted in the “Effect” box. The user can neither interact with the selected data in the “Overview of the Antennas Data” box, nor with the effects in the “Effect” box.

The functions in the “Navigation” box are not directly connected to boxes in the main page, but they provide the choices that the user can do, in other words, navigation is a section to handle the metafunctions which are not directly connected to how the user set the settings such as: distance and position of the user.

In the following parts, from section 6.2.1 to 6.2.5, the figure of each box is illustrated and explained in some more detail.

6.2.1 Navigation

In the navigation bar there are buttons that will allow the user to perform certain meta activities: Play, Scale, Redo, Save and Note Off, see figure 20. The user can choose a scale from the navigation and by defining an effect in the interface a sound will be generated. It is also essential to have a note off button in the navigation since there is a delay in producing sound. In some cases the MIDI signals can come “out of sync” which may cause a tone to be sent continuously, since the “note off” signal wasn’t picked up by the MIDI receiver. Using note off, it is possible to have a control over what we send and what we hear.

![Figure 20: Navigation](image-url)
6.2.2 Room Setting

The place where the antennas are located in the room can be either in the center of the room or in the corners, depending on how it is desired to engage the performers. Both setups have their respective advantages, although the cornered version probably will be the most often used. This setting is also selected as the default option. The desired room setting can be chosen from a combobox, see figure 21.

![Figure 21: Choosing the room setting. In this figure the centered positioning has been selected.](image)

6.2.3 Set Antennas Data

There are icons for each of the four antennas in “Choose Antenna” box, this is shown in figure 22. After choosing the room, the user can choose an antenna and set a variable which could be either position or distance of the person to the antenna.

![Figure 22: Choosing the antenna](image)

For antenna L and N, the position of the player to the antenna can be chosen, figure 23. If the user chooses L or N, “Choose Position” box will be activated.

For antenna K and M the user can choose the distance of the player to the antenna, figure 24. If K or M are selected, “Choose Distance” box will be activated.
6.2.4 Overview of the Antennas Data

“Overview” box shows the data for all antennas that exists, after setting a variable, data will be recorded and illustrated in this box, as shown in figure 25.

6.2.5 Effects

Here is a list of the effects. The activated effect will be highlighted in this box. The variable should be set for all antennas to activate an effect.
6.3 Defining an Effect in the Interface

In this section, we will go through the process of setting up the system through the software in a little more detail. Mentioned above, the interface is constructed somewhat in the shape of an open wizard walkthrough. However, in contrast to standard wizard routines, the user will have a visual representation of the choices made throughout the setup process. He or she can also go back directly to a previous choice and redo it without having to back up step by step. In this way, the interface will be helping the user to proceed in an appropriate sequence, while still not being completely rigid and forcing the user into repeating steps to change previous selections. The first step to define a function is to select the room setting. Figure 27 shows the first screen the user sees. In the next steps, the user needs to set variables for all antennas.

After choosing the room setting, the user can pick an antenna and give data to the antenna. The gray boxes are not clickable (i.e. still inactive). After choosing the room setting the second box will be activated and the user can pick an antenna, see figure 28.

In the next step of the scenario which is illustrated in the figures, the user pick antenna L. When the antenna L is selected, the box with the choosable
entities for antenna L will be activated, see figure 29.

When the user chooses the position of the player to antenna L, the position will be recorded in the overview section, see figure 30.

Figure 28: Main page after choosing the room setting

Figure 29: Main page before giving data to antenna L
After setting all variables, an effect will be activated. Figure 31 shows the variables in the “Overview” box which activate Effect1.

The antennas have two dimensions of expression, distance and lateral position of the person. Since the pupils are going freely around the scene, they can each affect pitch and some other effects on the synthesizer. This makes the interface both powerful and efficient, which on the other hand does not mean that it becomes easy to play the instrument. It is important that the instrument provides a challenge, and that it allows for the continuous development of playing skills. This can be achieved also by combining different antennas with different effects. This is also why it is necessary to be able to save the current settings, so that you don’t have to reprogram the interface every time. It will
probably be necessary to delimit the pitch and effects to digital signals, and use the same pitch control scheme as on the Theremini, see above.

7 Interfacing the Room

The players’ movements can be regarded as the main sound producing elements. Each person in the field can play notes based on the two variables: his or her distance and position in relation to the various antennas. To get a better appreciation of how this works I will start in this section, to introduce some necessary definitions of several terms in the theory of music, and next, a way to define the interface to produce note based on players’ movements is illustrated.

7.1 Note, Octave and Scale

Notes are symbols which are used to write sound music. Notes can be written on, between, under or above the staff. Staff is five horizontal lines to show the notes, see figure 32.

Figure 32: Five horizontal lines (i.e. staff), which are used to represent different pitches, the note value of each line position is further determined by a clef, either the G or Base clef (normally).

In the theory of music, four criteria have been defined for sound: pitch, intensity (in MIDI terms this is often referred to as “velocity”), duration and tone color. The intensity is the volume of the tone and it shows the strength of the hearing music. The duration is the amount of time that a note or musical compositions lasts, it is a time that one can hear the music. The characteristic of sound that makes it clear for us is called color tone, for instance the voice of two persons or the sound of two musical instrument, pitch is the most prominent of the qualities of the sound that can be shown on staff. The staff shows a pitch range vertically from low in the bottom to high at the top. The difference between the pitch or frequency of two sounds is defined as a musical interval.

The short intervals called half step and the long ones are one whole step. The interval which includes five whole steps and two half steps is called perfect octave, see figure 33.

The notes which are connected (conjunct) and are located in an octave interval are called a scale.

7.2 Players’ Movements in the Room

In the Thereminic room, there are two of the antennas that have three options for the position of the person to the antenna: right, left and center. Each position can represent a category of notes with different pitch octaves. In figure 34 each color shows one category. Hence if one chooses to move to the right side in the
Figure 33: The interval with five whole steps and two half steps which is called octave.

interface the black range will be selected, for the center and left options, two other categories are defined.

Figure 34: Three categories of notes with different pitches. Each category will be activated by choosing the position of the person to the antenna in the interface.

There are two antennas to detect the distance. The maximum distance between the antenna and a player is four meters and can be divided by 7 to show different notes. As a result, seven parts exist and each part, as shown in figure 35 represents one note. Consequently, one has a combination of notes with four categories, two based on distance and two based on position, respectively.

Figure 35: Dividing the distance of the performers to the antenna by seven to define seven different notes. Through a proper assignment of note values to a distance, it might be possible to play different set scales, such as a major or minor scales, or even harmonic subscales, such as blues, or pentatonic scales.
7.3 Music21 for Producing Note

The important question is whether it is now to be able to transfer the data that is streamed from the four antennas into MIDI format or not. It turns out that this is surprisingly easy, due to the library Music21, which is available in the library from Python programming language that makes it possible to define notes and make a list of them to be played in MIDI format. In the code, it is needed to import the library and define notes in the variables. For playing the notes it is not feasible to have a list, as a result, a stream is defined which appends the notes. As an example, in the following code, it is shown how three notes are appended to the stream and in the end they will be played by the last command. show(‘midi’)[16].

```python
from music21 import
t1 = note . Note ("F1")
t2 = note . Note ("F2")
t3 = note . Note ("F3")
stream1 = stream . Stream ( )
stream1 . append ( t1 )
stream1 . append ( t2 )
stream1 . append ( t3 )
stream1 . show( ‘ midi ’ )
```

As can be seen here, the conversion is very simple, and it allows for a very wide variety of expressions being programmed. Exactly how far it is possible to extend this playing model remains to see. Hopefully it will be possible to facilitate a large number of possible effects and note expressions. Currently, we have decided on providing a proof of concept, with a limited set of possibilities. However, it would be interesting to pursue this issue in the future. We will discuss this further in the next chapter.

8 Discussion and Future work

In the following sections, two suggestions for future design is discussed. In section 8.1, the design of two interfaces based on variables other than position and distance is shown. In section 8.2, some interaction methods that are applicable to the iPad version of the presented interface are highlighted. In section 8.3, some aspects of user experience improvement by using technology for converting body movement to music is discussed.

8.1 Variable Based Design

The presented interface design uses selected icons and idioms to guide the user. However, the target music applications, the synthesizers, are in most cases displaying a design that is based on the metaphor from the real world, and gives the user a sense of using the actual instrument, by the provision of buttons, knobs and sliders. The reason for the initial idiomatic design is that it is based on the technical (non-musical) variables of position and distance which can not be shown by knobs. In this section we will instead show two alternative designs of the main page, although with the same purpose as presented application but
they use different variables. The first design is based on octave and velocity variables, see figure 36.

The second example is for a Theremin which has different sensors on each antenna to give the user various options to choose from, see figure 37. This might be a very useful extension to the basic functionality, in that it might allow four polyvoices, that is up to four different notes being played simultaneously, one per antenna. We might also add chord based MIDI output, and other more advanced options.

Figure 36: Conceptual design of metaphoric interface with velocity, note, and octave as variables

Figure 37: Conceptual design of metaphoric interface with antennas having more than one sensor
8.2 Design for iPad

The original interface has been designed for a regular PC computer. However, carrying a PC around makes the setup less flexible, and might lead to less usage. Also, the functionality is not so extensive that it needs a PC for its implementation (possibly with exception for the synthesizer). Therefore the possibility to use an iPad or tablet has been considered for future development.

It turns out that there are very powerful synthesizer implementations developed for tablets, and implementing the application for iPad would therefore be a very interesting development, both with regards to cost and flexibility. We have looked at some possible inspiration sources for the implementation of an iPad version of the software. There are iPad applications with different interaction ideas that are applicable for iPad version of the presented application. The “Addictive Synth” [sic] is an iPad music synthesizer that allows the user to draw wave lines with finger, it is also possible to manipulate audio effects with finger movement and see the result real-time. Another example is TC-11 iPad application which has a built-in patch editor. The interface can be divided into different areas by the user. See figure 38 and figure 40.

Considering two variables for one antenna in the current system make it possible to have an iPad version with more interactions. If there are four antennas, the interface can be divided into four areas, each area represents one antenna with two variables (X - Y). One or both variables will change the parameters for one antenna through the movement of one finger.

In addition to the options for designing interface, a good idea for future work is to add other musical instruments to the systems and define some presets to make a willing music. Applying the ideas above are only possible on a real-time system with the minimum collision between the sensors.
8.3 Improving the User Experience

As it mentioned in chapter 2 there are different technologies which are used to convert body movement to music. The usage of such technology is not limited to the performances or music generation in schools by pupils, but it can be used in public places to enrich user experience. Sound Forest (Ljudskogen) in Scenkonstmuseet [7] in Stockholm is one of the good examples of room sized musical instrument. Visitors can use the entire body to create the music. It is a live audiovisual environment with vibrating tiles in the floor and playable strings from floor to ceiling to change sound and light [21].

Figure 40: An installation in Scenkonstmuseet which user can produce music and change lighting in the room by interaction with the strings in the room and tiles in the floor

One further idea for extending the room sized interface is to use sensors on the floor or other surfaces in the room to create music, with the hypothesis that the addition of visual expressions would give the user better experience while interacting with a room sized musical instrument. In this case it would be very interesting to also consider a completely new interaction theme that allows both sound and imagery to be configured through a software interface.
9 Conclusion

There are two basic ideas presented in this paper. One is the general principle for interfacing the room with its antennas, defining appropriate variables for the system, based on the players’ movements, and the other one is a prototype for interface design.

The interface prototype in this report is based on the properties of the physical implementation of the Thereminic room, as made by Rahmanu. However, the prototype shows that the implementation of the interface can be modeled on any sequence of MIDI signals coming from the Base Station Arduino. The idiomatic interface implementation maps the configuration of the four antennas and their activities to the appropriate MIDI signals for the software synthesizer.

The current interface is only set for the use of four antennas, but there are possibilities to add the motion sensors that were planned as part of the system, as well as up to four more antennas, without major reprogramming.

Currently the complete system incorporates the four antennas, a Base Station and a PC with the interface software presented in the thesis, which can be used in a working setup serving as a proof of concept. What remains, in order to producing a final product, is the final redesign of the antennas, and automating the process of setting up the whole system including the wi-fi connections. Today, this has to be made by manually connecting the antennas and the computer.
References


