NTNU – Trondheim
Norwegian University of
Science and Technology

Designing and Evaluating a Tablet Application for Guitar Effect Control

Lars Kinn Ekroll

Master of Science in Informatics
Submission date: June 2015
Supervisor: Maria Letizia Jaccheri, IDI
Co-supervisor: Rune Aalberg Alstad, Aalberg Audio
Aleksander Torstensen, Aalberg Audio

Norwegian University of Science and Technology
Department of Computer and Information Science

Summary

The company Aalberg Audio is developing a system of audio effects aimed at guitar players, and wants a software application for tablets that can control these effects wirelessly. This study tried to determine whether existing research into the design and evaluation of digital musical instruments could be applied to the design of such an application; another aim was to design and develop a working application prototype.

An iterative design science research method was used. A series of prototypes were designed and developed using strategies from the field of digital musical instrument design. Using an experiment with 7 participants, the final prototype was evaluated from the performer perspective using techniques from digital musical instrument evaluation, to see whether it increased discoverability, playability, and enjoyment.

Using digital musical instrument design strategies gave rise to mostly the same properties in the tablet prototype as one would expect in a digital musical instrument. For the participants, the final prototype achieved higher discoverability and somewhat higher enjoyment than traditional guitar effects, whereas playability was hindered by some interaction and design issues.

In conclusion, the existing research into the design and evaluation does indeed seem to be applicable to the design and evaluation of tablet applications for effect control.

Contents

Chapter 1

Introduction

Specialized portable audio effects units—called stomp boxes or effect pedals—are central to the sound of performers in many modern musical genres. The start-up company Aalberg Audio, which came out of NTNU's School of Entrepreneurship program, is developing a new and innovative effect pedal system, specifically aimed at electric guitar players, which includes the ability to control effects wirelessly.

As a part of their system, Aalberg Audio envision a tablet application able to control their effect pedals in real time. Little research seems to be available to guide the design and development of such an application. However, there is a large body of research into the creation of digital musical instruments (DMIs), which are control interfaces for digital audio synthesis units.

The electric guitar is not a digital musical instrument. It can, however, be described as a multi-component system that includes both the guitar, the associated effect pedals, and the amplifier as part of a larger instrument (Tanaka, 2010). Allowing the effects pedals to be controlled from a tablet would thus make the instrument partly digital. Seen in this light, the research into DMI design should have relevance also for the design of audio effect control interfaces for guitar players.

1.1 Aims of study

This study has two main aims.

1. Determining whether the existing research into digital musical instrument design and evaluation is applicable to the design and evaluation of interfaces for audio effect control.
2. Designing and developing a prototype tablet application that can control Aalberg Audio's guitar effects.

One important design decision for a DMI is the relationship between the control interface and the synthesis unit—specifically, the relationship between parameters of the control interface and parameters of the synthesis unit. This relationship can be described in terms of *mapping strategies* (see section 2.1). Research shows that different mapping strategies give rise to different properties in the DMI. It would seem that mapping strategies could also be used to describe the relationship between a control interface and an audio effect unit. This leads to the first research question:

RQ1: Compared to a digital musical instrument, does the use of a particular mapping strategy give rise to the same properties in an audio effect control interface?

To answer this question, the prototype application was designed based on mapping strategies for digital musical instruments.

Another important aspect of DMI creation lies in evaluating the final instrument. This is also important when creating an interface for audio effect control. The literature suggests that evaluation methods from human-computer interaction may not be directly applicable to DMIs. Because of this, DMIs have their own, distinctive perspective on evaluation (see section 2.3). This leads to the second research question:

RQ2: Are evaluation methods for digital musical instruments also useful for evaluating audio effect control interfaces?

To answer this question, the prototype application was evaluated using methods from the digital musical instrument literature.

1.2 Case background

Aalberg Audio have envisioned a guitar effects system consisting of three parts: a series of effect pedals, a wireless controller called *Aero*, and a software application for the Android and iOS platforms. The first two parts of this system have at the time of writing been realized. The effect pedals can be remotely controlled using the wireless Bluetooth low energy standard (Bluetooth Smart[1]). At the start of

[1] http://www.bluetooth.com/Pages/Bluetooth-Smart.aspx

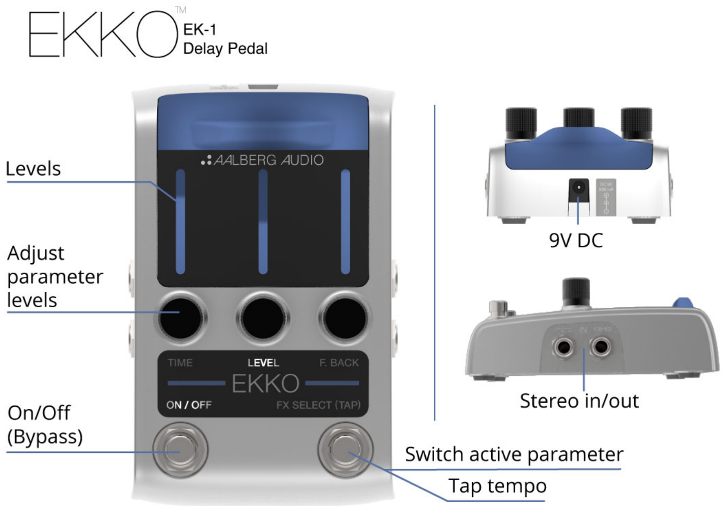


FIGURE 1.1: Layout of the Ekko delay pedal.

this study, the delay effect pedal *Ekko* and the wireless controller *Aero* were in their final prototype stages. The reverb effect *Rom* and the tremolo effect *Trym* were only on the drawing board.

1.2.1 The effect pedals

As seen in figure 1.1, *Ekko* has three adjustable parameters—time, level, and feedback—each with its own physical rotary encoder. Additionally, the time parameter can be set by tapping the bottom right foot switch in the desired tempo. This foot switch can also be used to change the currently active parameter, the relevance of which will be described later. An on/off foot switch on the bottom left side rounds off the interaction possibilities of the pedal itself. In the *off* state, the audio signal passes unaltered through the pedal. This is known as *bypass*.

Aalberg Audio's three effect pedals are shown in figure 1.2 on the next page. The general layout of each is similar, though each has different parameters. In particular, all three pedals have the following in common:

1. The top of each pedal has a strip in an identifying colour.

FIGURE 1.2: The three effect pedals Ekko, Rom, and Trym.

2. Each pedal has three parameters, each of which is associated with a vertical level indicator and a rotary encoder. The level indicators use the pedal's identifying colour.
3. The leftmost parameter of each pedal has to do with time, and can be controlled using the tap tempo foot switch.
4. Each pedal has the same foot switches at the bottom for controlling bypass (on/off), selecting the active parameter, and using tap tempo.
5. Above the foot switches, each pedal has a label area containing labels for the parameters and foot switches, as well as the name of the pedal.

The unique properties of each pedal will now be described. This includes a description of each pedal's purpose, and of the effect's parameters. Note that the descriptions of audio effects are meant to be brief and in layman's terms, to provide context for those unfamiliar with these types of effects, and are thus not entirely accurate.

Ekko

Ekko is a delay effect, which allows a user to create a sound similar to a repeating, decaying echo. A digital delay effect generally delays the input signal for a given time, then plays it back. The delayed signal is then fed back into the effect, repeating the process. In the *Ekko*, the unprocessed signal is also played back at the same time.

Ekko has the parameters time, level, and feedback.

Time is a value ranging from 0 to 2000 milliseconds. It determines how long the input signal is delayed.

Level is a value ranging from 0 to 100%. It determines the sound level of the processed signal.

Feedback is a value ranging from 0 to 100%. It determines the amount of the signal which is fed back into the delay process.

Rom

Rom is a reverb effect, which allows a user to simulate the reverberation of sound that occurs in rooms of various sizes. Reverberations occur because sound waves reflect differently off of different room surfaces. Multiple reflections build up, all of which arrive at the listener at different times after the direct sound. Surfaces also absorb sound waves, meaning the number of reflections will decay over time. The time between the listener hears the direct sound until the reflected sound waves start arriving is called the *pre-delay*. *Reverberation time* is the time it takes for the sound to decay a standard amount, usually 60 dB. Reverberation is differentiated from echo: with echo, the individual reflections are clearly distinguishable, whereas reverberation has so many reflections that the listener is unable to distinguish them.

Rom has the parameters pre-delay, mix, and size. At the time of writing, the final parameter ranges were not finalized.

Pre-delay is a value ranging from 0 to 200 milliseconds. It determines the pre-delay time, as described above.

Mix is a value ranging from 0 to 100%. It determines the balance between the direct signal and the delayed signals. A mix of 0% represents only the direct signal, a mix of 50% has an equal balance, and a mix of 100% represents only the delayed signals.

Size is a value ranging from 0 to 100%. It determines the size of the simulated room, which influences the reverberation time. A value of 100% represents a very large room, like a cathedral, whereas a value of 0% represents an infinitely small room, i.e., no reverberation.

Trym

Trym is a tremolo effect. This effect produces a “trembling” sound, caused by periodic variations in the sound level. The periodic variation is achieved by using a low frequency oscillator (LFO)—a signal generator producing a periodic signal at a frequency below human hearing—to modulate the input signal's amplitude.

Trym has the parameters speed, shape, and depth. At the time of writing, the final parameter ranges were not finalized.

Speed is a value ranging from 50 to 1250 milliseconds. It determines the period of the LFO[1]—that is, the time from peak to peak of the produced signal.

Shape is a value ranging from 0 to 100%. It determines the shape of the signal produced by the LFO. At 0%, the LFO produces a sine wave; at 100%, the LFO produces a square wave. The values between 0 and 100 represent a gradual change from the sine wave shape to square wave shape. In essence, the shape of the LFO signal determines the abruptness of the variation in the input signal's amplitude. With a sine wave the variation occurs gradually, whereas a square wave causes a virtually instant change.

Depth is a value ranging from 0 to 100%. It determines how much influence the LFO's signal has on the input signal. At 0%, there is no change in the input signal; at 100%, the LFO signal will periodically completely mute the input signal.

1.2.2 The wireless controller

Aero, the wireless controller, is the second part of the system. This controller has five buttons, an on/off switch, and a rotary encoder which can also be pressed to function as a sixth button. *Aero* can control *Ekko* and future Aalberg Audio effect pedals, and claims a range of up to 30 metres.

The default configuration for controlling the *Ekko* pedal can be seen in figure 1.3 on the facing page, and is as follows:

- The rotary encoder adjusts the currently *active parameter*.
- Two buttons on the left side of the figure are used to select the previous or next active parameter.
- Pressing the rotary encoder allows users to set the time parameter of the current effect, through the tap tempo functionality.
- Pressing and holding the rotary encoder toggles the *bypass* mode—i.e. the pedal's on/off state.

[1] The parameter can equivalently be represented as the LFO's frequency, in Hz, which would give the parameter a range of 20 Hz (at the bottom) and 0.8 Hz (at the top). Since it seems more natural to have lower numbers at the bottom of the level indicator, and the pedal itself has no indication of the unit used, the author chose the millisecond representation.

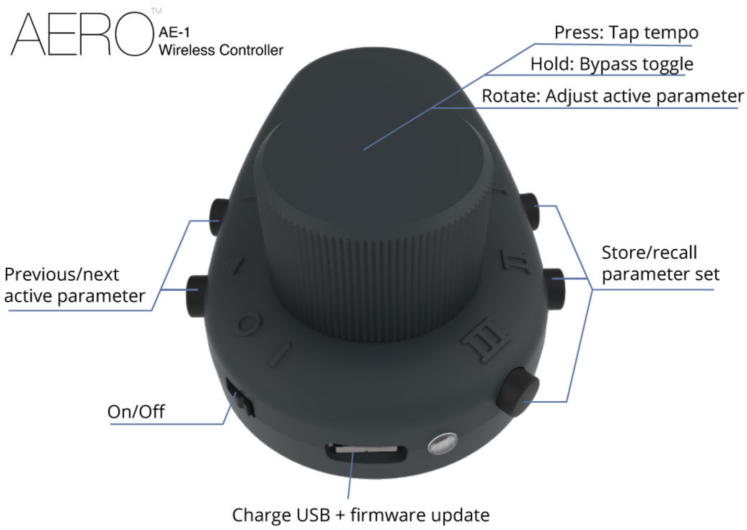


FIGURE 1.3: Layout of the Aero wireless controller.

- The final three buttons are used for storing and recalling *presets* (sets of parameter values).

Both the *Ekko* effect pedal and the *Aero* wireless controller have been through several rounds of prototyping, and are currently in a state where they can be—and have been—used together during a live concert performance.

1.2.3 The software application

The final and as yet unimplemented third part of the system was envisioned as a software application for mobile devices, with a focus on tablets running the Android and iOS operating systems. The application would communicate both with the effect pedals and with the wireless controller. Some preliminary work towards such an application was already under way at the start of this study, through the development of a cross-platform communication API. However, no work on the actual user interaction had been done.

Aalberg Audio and the author together decided that the goals for a prototype application would be to extend and enhance the real-time control of Aalberg Audio's effect pedals, and to ease the creation and management of presets[1]. The

[1] A *preset* is a set of parameter settings that can be stored and retrieved simultaneously.

application should ideally be cross-platform for iOS and Android.

Chapter 2

Theory

In this chapter, some important theoretical concepts are introduced. Section 2.1 describes the concept of mapping, which is one of the central design considerations for digital musical instruments. Section 2.2 clarifies the language used to talk about prototyping, an approach that has been used in this study. Finally, section 2.3 describes how digital musical instruments can be evaluated.

2.1 An overview of mapping theory

The concept of mapping is introduced, after which the systems-oriented and functional perspectives on mapping are presented. Finally, empirical results in the literature are presented and discussed.

2.1.1 What is mapping?

When the means of controlling an audio effect or synthesis unit is separable from the unit itself, a question arises of how these are related. In the field of computer music, the concept of *mapping* is used to describe this relationship (Keislar, 2014). The gestural interface and the synthesis or audio effect unit are said to be related through *mapping strategies*, which refer to the way input gestures are mapped onto sound parameters (M. Wanderley & Depalle, 2004).

2.1.2 Viewpoints on mapping

Several viewpoints can be taken on mapping. Some articles have taken a *systems-oriented* view (Van Nort, Wanderley, & Depalle, 2014). This describes mapping as a correspondence between control parameters and sound parameters, i.e. what to map where. The complexity of these mappings has generally been described as *one-to-one, one-to-many,* and *many-to-one* [1] (Rovan, Wanderley, Dubnov, & Depalle, 1997; Hunt & M. M. Wanderley, 2002). Some combinations of these can be called *many-to-many* (Hunt & M. M. Wanderley, 2002). Examples include the interfaces described as multiparametric, which combine one-to-many and many-to-one mappings (Hunt & Kirk, 2000; Hunt & M. M. Wanderley, 2002; Hunt, Wanderley, & Paradis, 2003).

One-to-one mappings are claimed to be the least expressive (Rovan et al., 1997). One-to-many mappings are described as expressive on a macro level, but do not allow access to the individual parameters. Many-to-one mappings, on the other hand, are claimed to be highly expressive, but much harder to master than one-to-one mappings (Rovan et al., 1997).

Also of interest is how the mappings respond to input. In the systems-oriented view, the response can be thought of as being modified by some transfer or warping function (Van Nort et al., 2014). For instance the response can be linear, or follow a logarithmic, quadratic, or exponential curve.

The *functional* view of mapping draws on concepts from geometry and topology (Van Nort et al., 2014). In terms of what to map where, a mapping is defined by the explicit point-wise associations between an n-dimensional space of control parameters, the “control space”, and an m-dimensional space of sound parameters, the “sound space” (Van Nort et al., 2014). The response of the mapping is then defined by the rules governing the association of entire subregions of these parameter spaces that are not explicitly mapped, which can be said to be the geometric structure of the mapping (Van Nort, 2010; Van Nort et al., 2014).

When the control space has fewer dimensions than the sound space—that is, the number of control parameters is lower than the number of sound parameters, which is usually the case—we can view mapping as an embedding of a lower-dimensional control space in a higher-dimensional sound space (Van Nort et al., 2014). Using most approaches, this leads to parts of the sound space being unreachable. This means that a mapping must be carefully designed to make sure the desirable parts of the sound space are reachable. However, what constitutes

[1] One-to-many and many-to-one mappings are also known as divergent and convergent mappings, respectively.

a desirable combination of sound parameter values is subjective. Designing mappings to be used by multiple people is thus, arguably, very hard.

Comparing these viewpoints, it can be said that the systems-oriented view is well suited for analysing how individual input parameters influence individual sound parameters. On the other hand, the functional view is concerned with how the overall state of the input parameters affects the state of the sound parameters. Thus, the systems-oriented view can be described as analytical, whereas the functional view is holistic (Van Nort et al., 2014).

2.1.3 Empirical results in the literature

One-to-one mappings have been shown to be the least engaging for users (Hunt & Kirk, 2000). They are easy to learn for simple tasks, but users show little improvement over time beyond this (Hunt & Kirk, 2000). Some people seem to prefer one-to-one mappings, and it is suggested that these people may prefer to think analytically rather than holistically (Hunt & Kirk, 2000). Hunt and Kirk define the *analytical* mode of thought as breaking the sound down into its individual parameters, which can be studied one at a time. The *holistic* mode of thought is defined as thinking of the resulting sound as a whole, with details being of lesser importance.

It has also been shown that users find *multiparametric* many-to-many interfaces—which use a combination of one-to-many and many-to-one mappings—more engaging than interfaces using only one-to-one mappings (Hunt & Kirk, 2000; Hunt et al., 2003). Multiparametric interfaces are also shown to be harder to learn initially, but with training end up being more effective at complex tasks (Hunt & Kirk, 2000; Hunt et al., 2003).

It has been claimed that the lack of access to individual parameters makes one-to-many mappings of limited use on their own (Rovan et al., 1997). This has some support from a recent experiment which compares an innovative interface based on one-to-many mapping; an interface which simply uses sliders for one-to-one mapping; and an interface which presents both simultaneously (Tubb & Dixon, 2014a).

Seen from a systems perspective, the first of these interfaces consists of two one-to-many mappings. The input from the controller is a position on a 2D plane, and the x and y coordinates each map to five different sound parameters. Using the functional perspective, the mapping can be described as a function from a 2-dimensional controller space to a 10-dimensional sound space. What makes the interface innovative is that it uses a Hilbert curve, which is a type of space-filling curve, for each control dimension. Each Hilbert curve visits every point in its respective 5-dimensional sound space. This makes every point in the sound space reachable using only these two control dimensions.

The combined approach was seen by users as better than either of the individual interfaces (Tubb & Dixon, 2014a). A likely reason is that the interfaces are good for different things. The slider interface was perceived as better at fine-tuning sounds and gave a feeling of more control, which are qualities useful in the idea selection or "convergent" stage of creativity. On the other hand, the interface using Hilbert curves was better for quickly discovering interesting sounds and generating new ideas, and felt more creative, which are qualities that are useful in the idea generation or "divergent" stage of creativity. It was also deemed as being better than a randomizer for this purpose (Tubb & Dixon, 2014a).

Some conclusions can be drawn from these results. First, taking a systems-oriented view, it seems that an interface which combines mappings of different complexities will work better than one which uses only one-to-one, only one-to-many, or only many-to-one (Hunt & Kirk, 2000; Tubb & Dixon, 2014a). Second, different situations or tasks—like idea generation, idea selection, and live performance—call for different mapping strategies (Tubb & Dixon, 2014a).

2.2 Prototyping

As described later, this study used a prototyping approach to design and development. Beaudouin-Lafon and Mackay (2012) define a prototype in the context of human-computer interaction as a concrete representation of part or all of an interactive system. The authors describe how prototypes and prototyping techniques can be characterized along four orthogonal dimensions: *representation, precision, interactivity,* and *evolution.*

2.2.1 Prototype characteristics

Representation describes the form of the prototype. Examples include paper sketches and computer simulation. Representations can fall into two basic groups: *offline* prototypes, which can be created quickly and cheaply, usually at early stages of design; or *online* prototypes, which run on a computer, are usually more costly to produce, and are usually not appropriate for early design stages.

Precision describes the level of detail at which the prototype is to be evaluated. For instance, a prototype could be informal and rough, or highly polished. Different parts will be more or less precise depending on the purpose of the prototype:

high precision makes relevant details clear, whereas low precision leaves irrelevant details open for interpretation and further exploration.

Interactivity describes the extent to which a user can actually interact with the prototype. A prototype can be classified as *fixed*, with no possible interaction; *fixed-path*, with a small set of interaction paths; or *open*, supporting a large set of interactions. In offline prototypes, interactivity is often achieved by having one or more people playing the role of the interactive system, presenting information and responding to the user's actions. Online prototypes often implement parts of the system, making the computer handle interactivity for some parts while leaving others under human control.

Evolution describes the expected life cycle of the prototype. A prototype can be *rapid*, meaning inexpensive, easy to produce, and disposable; *iterative*, meaning it evolves through several iterations to work out details or explore alternatives; or *evolutionary*, meaning it evolves to become a part of the final system.

2.2.2 Prototyping strategies

We can also talk about *prototyping strategies.* These decide which role prototypes play with respect to the overall system, and the order in which different aspects of the prototypes are created. Beaudouin-Lafon and Mackay (2012) describe four strategies: *horizontal, vertical, task-oriented,* and *scenario-based.*

Horizontal prototypes represent one entire layer of the design, and are used to get an overall picture of the system from the perspective of the users. Issues that can be addressed include consistency, functional coverage, and redundancy.

Vertical prototypes fully implement a feature from the user interface layer down to the system layer. They are often used to assess the feasibility of a feature described using one of the other prototyping strategies.

Task-oriented prototypes are based on a task analysis which identifies the individual tasks users want the system to accomplish. These prototypes combine features of horizontal and vertical prototypes, in that they are broad enough to cover all the required functionality yet deep enough to enable detailed analysis of how the tasks can be supported.

Scenario-based prototypes share many similarities with task-oriented prototypes. However, instead of focusing on individual and independent tasks, they follow more realistic design scenarios of real-world usage.

2.3 Evaluating digital musical instruments

O'Modhrain (2011) provides a framework for evaluating digital musical instruments (DMIs) from the perspective of different stakeholders: the audience, the performer/composer, the designer, and the manufacturer. These stakeholders have varying goals, among which are enjoyment, playability, robustness, and achievement of design specifications. O'Modhrain's framework will be used to frame the discussion of how to evaluate musical user interfaces.

Few, if any, projects will be able to dedicate time to pursuing all of these perspectives. In this study, the author decided to focus on the performer's perspective.

Playability, robustness, and enjoyment are described by O'Modhrain as possible evaluation goals from the performer's perspective. These, along with the additional goal of discoverability, are described next.

2.3.1 Playability

The playability of a musical interface is obviously important to the performer. After all, playing it—that is, using it in a performance context—is often the end goal in and of itself (McDermott, Gifford, Bouwer, & Wagy, 2013).

Playability is arguably affected most by the mapping strategy used. One concern is how well the mapping is communicated to the performer, or in other words, how well the performer's mental model of the interface matches the implementation. A performer with an erroneous understanding of the system may not be able to predict the system's response. Anything that reduces the fluidity of interaction will also reduce playability, this includes technical issues like jitter and latency (Wessel & Wright, 2002).

2.3.2 Robustness

Robustness is important for performers, as they must know they can rely upon their instruments when performing. Robustness is affected by both hardware and software defects, which may for instance make the system respond incorrectly or not at all. Additionally, the technical issues mentioned under playability also negatively affect robustness.

Although this goal would be very important in a finished commercial system, it is of limited importance for a prototype. A prototype does, for instance, not need to support a variety of performance situations or setups. The finished prototype should, however, work reliably within the context of the evaluation.

2.3.3 Enjoyment

Enjoyment is important if we want performers to keep using a musical interface, and several factors may affect it. First, one could argue that the playability as a whole affects enjoyment. Second, it seems that musicians tend to enjoy overcoming obstacles; mastering an instrument should be challenging, in order to engage musicians in developing virtuosity (O'Modhrain, 2011). McDermott et al. (2013) similarly argue that activities involving musical instruments must be continuously challenging in order to remain engaging, and that the challenge must be at an appropriate level. This leads to a tension between *ease of use*—a common goal in human-computer interaction—and *long-term engagement*. McDermott et al. (2013) continue to explore which aspects of music interaction benefit from difficulty, and which aspects are negatively affected by it. With this insight it should be possible to evaluate whether a musical interface is more difficult than it needs to be, without falling into the trap of making it user-friendly but uninteresting.

2.3.4 Discoverability

Expanding on the possible goals given by O'Modhrain (2011), we can add discoverability: being able to discover interesting sounds and explore the possibilities of the interface. Based on the results from Tubb and Dixon (2014a, 2014b), it would seem like the chosen mapping strategy also influences this goal: the one-to-many interface was better suited to the idea generation stage of creativity. The ideas about analytical and holistic thinking, presented by Hunt and Kirk (2000), may also play into discoverability. A mapping strategy that mainly supports analytical thinking, by clearly presenting each sound parameter, might cause a user to rely on preconceptions about how the parameters should be set. Making the parameters less explicit, thus supporting holistic thinking, might thus let a user try parameter combinations he or she had not previously thought of.

2.3.5 Evaluation methods

O'Modhrain (2011) outlines some methods for evaluating the achievement of goals.

Goals can be evaluated by giving users simple musical tasks placed within a musical context, with tasks aiming to test features related to the goals. In contrast with normal practice in human-computer interaction usability testing, it is suggested that think-aloud strategies—where users comment on their thought process as they perform tasks—are detrimental to musical tasks, since users are both performing with, listening to the audio output of, and evaluating the system under consideration. Therefore, it is better to ask users to reflect on their experiences after the event. Reminding users about actions they made or problems they faced when performing the tasks seems to be effective. This suggests a qualitative approach using observation and semi-structured interviews.

It is also normal to capture quantitative performance data from the interface while users perform tasks, including timing and accuracy data. This does not, however, capture the user experience. A quantitative approach to analysing the user experience is to monitor various physiological data from users while they perform tasks, though this is inherently difficult, and out of scope for this study.

Finally, longitudinal studies can be useful, because musical instruments and interfaces both have a learning curve and need to be enjoyable over the long term.

Chapter 3

Methodology

This chapter is concerned with the methodology used in the study. The chosen research method—design science research—is described first in section 3.1, after which the design and development of the horizontal prototypes and the vertical prototypes are described in more detail in sections 3.2 and 3.3. The final prototype is then examined in section 3.4. Finally, the method used to evaluate said prototype is detailed in section 3.5.

3.1 Design science research

The study followed an iterative design science research method, as defined by Peffers, Tuunanen, Rothenberger, and Chatterjee (2007). Using this framework, the research consisted of six steps or activities: problem identification and motivation; definition of the objectives for a solution; design and development; demonstration; evaluation; and communication. The steps were not followed in strict sequence, but formed an iterative process which can be described as learning via making (Oates, 2006, p. 111).

Each of the steps will now be described.

3.1.1 Problem identification and motivation

The identified problem was, in essence, discovering how—or, indeed, if—a touch-enabled tablet device could enhance the experience of using Aalberg Audio's effect system. The motivation for the research came from Aalberg Audio, who had requested Master's students who could investigate the design and implementation of an effect control application on tablets.

3.1.2 Definition of the objectives for a solution

At the start of the study, the objectives for a solution were simply that it should

1. be an application for tablets;
2. be developed using Xamarin, to make future cross-platform support for Android and iOS easier;
3. communicate with the Aalberg Audio's effect pedals using Bluetooth Smart; and
4. extend and enhance the control of Aalberg Audio's effect pedals.

In order to develop for iOS you need access to Apple hardware both to compile the code (an Apple computer) and to run the application (an Apple tablet, or an iOS emulator running on an Apple computer). Since the author lacked such hardware, a decision was made to develop the solution with cross-platform compatibility in mind, but with a design specific to Android.

The fourth objective, while a noble goal, is fairly vague and not very testable. Accordingly, as the study progressed, more specific objectives were decided upon. The author initially tried asking Aalberg Audio for both some quantifiable objectives, e.g., users should be able to do some task more quickly with the tablet interface than without; and for qualitative objectives, e.g., that users should report increased creativity, efficiency, satisfaction, or some other subjective measure. However, the company could not express any concrete objectives, leading to the investigation into musical user interface evaluation presented in section 2.3. This investigation guided much of the subsequent design and evaluation.

In the end, the author decided that the main objectives should be to increase the discoverability, playability, and enjoyment of the effect system. This was based on the observed limitations of the existing control possibilities in the system: Both the controls on the pedals and the wireless Aero controller places focus on one effect parameter at a time. The theory suggests this limits playability and enjoyment, and the author would argue it limits discoverability. This is especially the case with the Aero controller, because it only has a single rotary encoder. The author thus saw this as the area which a tablet interface could most significantly improve upon.

3.1.3 Design and development

With the objectives defined, an artefact can be designed and developed to solve the problem. In this study, the main artefact was an *instantiation*: specifically, a prototype software application running on an Android tablet allowing wireless control of guitar effects.

The design and development followed a prototyping approach, in which several iterations of prototypes were designed, implemented, and analysed (Oates, 2006, pp. 113–114). As the analysis step of prototyping is closely intertwined with the evaluation step of the research process, it will be covered there.

There were two main phases of design and development. Early iterations focused on horizontal, iterative prototypes; whereas later iterations focused on vertical, evolutionary prototypes. All prototypes were online and interactive. The horizontal prototypes had fairly high precision, and the vertical prototypes had very high precision. These two phases of design and development are described in sections 3.2 and 3.3.

3.1.4 Demonstration

After developing a prototype, its ability to solve the problem has to be demonstrated. This involves placing the prototype in a suitable context, which in this study means allowing guitar players to use the prototype to control Aalberg Audio's effect pedals.

Being able to place the prototype in this context relied on the Bluetooth Smart communication to be properly specified and implemented on the effect pedals by Aalberg Audio. During the first half of the study, it looked like this would be done. However, as time passed it became clear that the implementation would not be ready until after the study had concluded. This meant an alternative context had to be considered.

As the main function of the prototype would be controlling audio effects, it was seen as essential that users of the prototype would get some form of audible response to actions performed in the user interface. A "Wizard of Oz" approach could have been used, in which the effects were actually controlled by a human based on the users' actions. However, this would lead to a response time that was considered unacceptable for an accurate evaluation of the prototype. Thus, a suitable replacement for Aalberg Audio's effects had to be found or developed. After some consideration, the author concluded that developing a software replacement running on the computer to be the easiest solution. This had the advantage that the effects could be made similar in sound to Aalberg Audio's effects, and use the same parameters with the same maximum and minimum values, meaning the results from the evaluation would be more applicable. Development of the replacement effects is covered in section 3.3.

3.1.5 Evaluation

If a prototype is able to solve the problem, the next question is: how well? An answer to this can be found by observing users solving the problem using the prototype, and interviewing them about the experience.

Evaluation and analysis during design and development was mostly informal. It consisted of demonstrations for, testing by, and subsequent conversations with Aalberg Audio staff; as well as the author's own testing and assessment. The former was very important to prioritize features, define scope, and steer the development in the right direction. The latter was especially important in later stages of development, as frequent small changes had to be made to the various touch interactions to make them feel right.

After developing the final prototypes, a formal experiment was set up to compare the prototypes to each other and systematically gather qualitative opinions. This is covered in detail in section 3.5.

3.1.6 Communication

After a final artefact has been developed and evaluated, the final step of the design research process is communicating the results. In this case, the means of communication is the Master's thesis you are reading at this very moment.

3.2 Horizontal prototype design and development

This section covers the design and development of the horizontal prototypes mentioned earlier.

The horizontal prototypes were designed and implemented as interactive wireframes using Justinmind Prototyper[1].

A screen capture of the last early prototype is shown in figure 3.1 on the next page. The design closely follows Google's *material design* guidelines (Google Inc., n.d.). The features of this design will now be described.

[1] http://www.justinmind.com

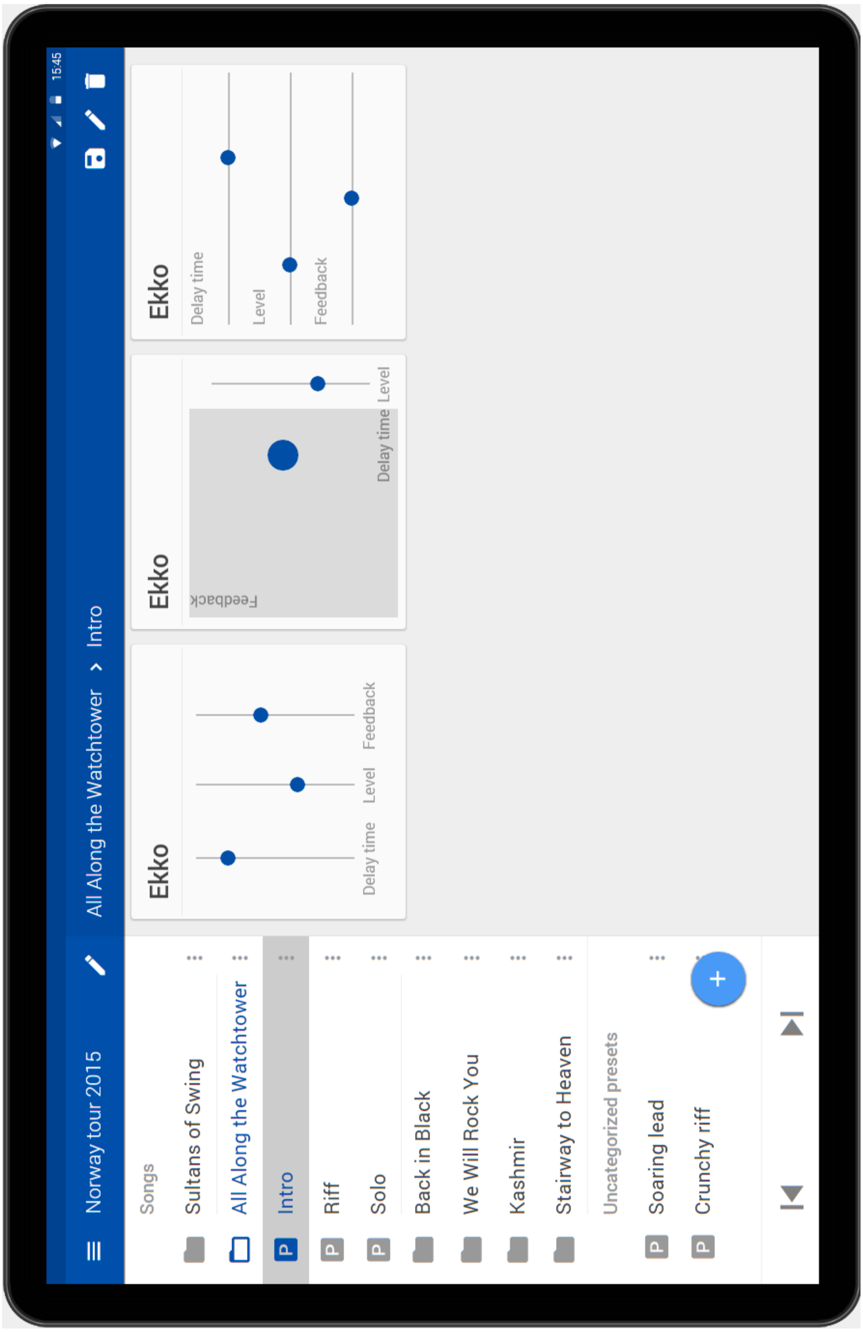


FIGURE 3.1: Early horizontal prototype: set list on the left side, effect control area on the right side.

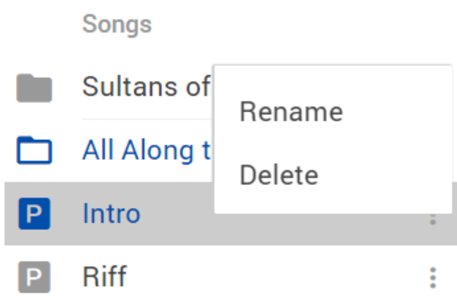


FIGURE 3.2: Action overflow menu opened for the song "Sultans of Swing" in the set list.

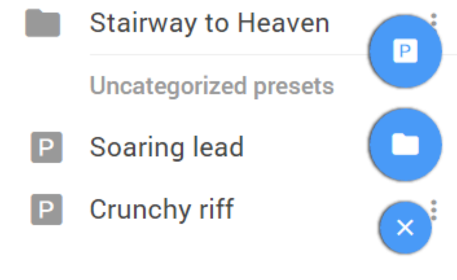


FIGURE 3.3: Pressing the floating action button to add an item allows a choice between adding a preset (top) or song (middle), or cancelling the action (bottom).

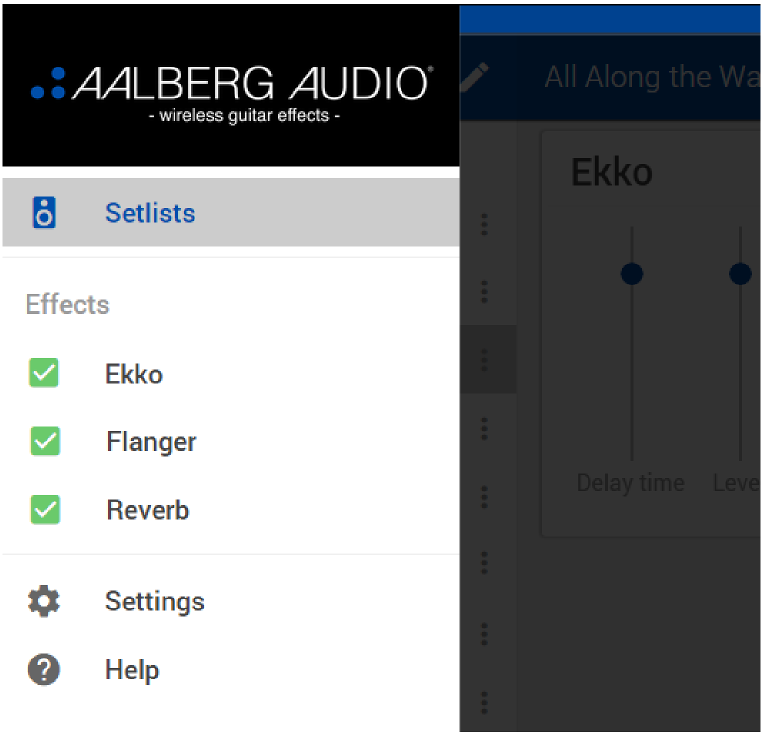


FIGURE 3.4: Navigation drawer for navigation and effect overview.

3.2.1 The set list

On the left side, approximately a quarter of the screen space is used for a list for organizing presets (called a *set list*). The idea was to use the set list—a document commonly used by musicians to list what songs they intend to play during a performance—as a metaphor for managing presets.

Users would use drag-and-drop to group presets in folders (here called *songs*). Drag-and-drop would also be used to reorder items in the set list. The figure shows a set list with six songs, one of which is opened to show three presets; with two unorganized presets near the bottom. Each item in the set list has a button consisting of three vertical dots, which Google calls an *action overflow*. Pressing this button would allow a user to rename or delete an item directly from the set list (see figure 3.2 on page 22).

Back and forward buttons on the bottom of the set list would allow a user to jump quickly to the next or previous preset. Also at the bottom is a *floating action button* (blue), which would allow a user to add a new song or preset to the set list (see figure 3.3 on page 22).

The intent was for a user to be able to have separate set lists for different occasions. Creating, browsing, and selecting set lists was however not shown in the prototype. The toolbar, in blue, at the top of the set list area shows the name of the active set list (here "Norway tour 2015") and has an action button for renaming the set list (pencil icon).

3.2.2 Effect control

The remaining screen space consists of an area for displaying controls for the various connected effects. Three very rudimentary variations are shown for the Ekko effect. This part of the prototype was not very fleshed out, because it turned out to be very hard to prototype this kind of touch interaction using Justinmind Prototyper.

The toolbar, in blue, at the top of this area shows the song and preset name (here "All Along the Watchtower" and "Intro", respectively) for the currently selected preset. Three actions are also defined using icons in the top right: save this preset (floppy disk icon), rename this preset (pencil icon), and delete this preset (trash can icon).

3.2.3 Navigation drawer

Pressing the "hamburger icon" (three horizontal lines) in the top left corner brings out the navigation drawer, shown in figure 3.4 on page 23. This drawer would allow navigation to various areas of the application, such as the application settings and a help section. The drawer also shows a list of owned pedals with connection status (green for connected, grey for not connected), and would allow the user to quickly toggle each effect on or off using checkboxes.

3.2.4 Early prototype analysis

The feedback from Aalberg Audio focused on two areas: prototype scope and looks.

Defining what functionality the prototype should include is an important part of the design and development activity. Aalberg Audio thought the preset management features to be of lesser priority than the real-time control features, and asked for the rest of the prototype iterations to focus the latter.

Aalberg Audio also thought the effect controls looked too dissimilar to their effect pedals, and asked for the controls to more closely follow the pedal designs.

Based on the experiences with using Justinmind Prototyper, and from the feedback provided by Aalberg Audio, it was clear that a new design approach was needed.

3.3 Vertical prototype design and development

In order to focus on the effect control, a decision was made to switch to prototypes focusing solely on the control experience. This implied a vertical prototyping approach. The experience with Justinmind Prototyper had shown it to be ineffective for prototyping touch interaction for the effect control interfaces, and the design work could not be easily reused in the actual Android application. Thus, the iterative, throwaway prototyping approach was dropped in favour of doing design and development of evolutionary prototypes directly in the Xamarin development environment. At the same time, it had become clear that communicating with Aalberg Audio's guitar pedals using Bluetooth Smart would not be possible within the time frame of the study.

The new approach to design and development therefore involved

- coming up with designs based on findings from the mapping theory discussed in section 2.1, while evoking a design similarity with Aalberg Audio's pedals;
- developing an application architecture that would simplify future cross-platform support;
- implementing the designs as an Android application, including development of user interface elements that could not be effectively expressed as standard Android input controls;
- developing a suitable substitute for Aalberg Audio's effects, which could be used for demonstration and evaluation; and
- implementing communication with the effect substitute.

These tasks were not performed one after another. Rather, they were done mostly concurrently and iteratively. However, the process started with formulating an initial design idea and ended with a final prototype. The initial design idea is covered first, in section 3.3.1; a description of the final prototype follows in section 3.4.

3.3.1 Initial design idea based on mapping theory

With the objectives of the prototype being increased enjoyment, discoverability, and playability for the performer, the question remained of how this could be done. Using the findings from the review of mapping theory, an initial design idea was formed. This idea sought to combine the following feature goals:

FG1: Since multiparametric interfaces have been shown to be more engaging, a mapping strategy combining one-to-many and many-to-one mappings should ideally be used. Such a strategy results in an interface that is inherently more complex than a simplistic one-to-one mapping, providing users with a degree of challenge and possibility of mastery. This should improve enjoyment. By virtue of reducing the reliance on analytical thinking—that is, thinking of the resulting sound in terms of the individual parameters—it should also improve discoverability.

FG2: It should be possible to easily control multiple simultaneous sound parameters. Extending the control in this way should improve playability, as it would be easier to quickly change a sound or make small variations while playing. This quick, fluid movement through sound space should also improve discoverability. This goal implies the use of one-to-many mappings, and/or leveraging touch gestures that can be performed simultaneously.

FG3: There are some situations in which one-to-one mappings seem to be preferred, including the idea selection stage of creativity, and some people who just happen to like one-to-one mappings better. Thus, an interface based on one-to-one mapping should be included, and it should be easy to switch from one interface to the other.

FG4: Because the application will be used by guitar players, it has to be designed for one-handed use. Guitar players actually use both hands when playing, but may in certain situations be able to spare one hand (e.g., when holding long, sustained notes the strumming/plucking hand may be spared; when playing on an open string the fretting hand may be spared).

FG5: Even when the strumming/plucking hand can be spared, guitar players often use the thumb and index finger of this hand to hold a plectrum with which to pluck the strings. Because of this, all actions in the application should be possible using a single finger.

FG6: The interface should take design inspiration from Aalberg Audio's pedals, to satisfy their demands for design coherence across the hardware and software.

It is worth noting that the inclusion of two separate interfaces with different mappings should help alleviate a problem, described by Tohidi, Buxton, Baecker, and Sellen (2006), that can occur when evaluating a single interface. Showing only a single prototype to a participant provokes higher ratings and more positive comments than showing several prototypes, whereas showing several prototypes leads to increased criticism and a decrease in positive comments. Tohidi et al. suggest that showing several prototypes sends a clear message that the designers have not yet made up their minds, and means that the participants do not have to worry about "disappointing" the designers. It also allows participants to criticize without being negative.

The next section describes how these features were included in the final design.

3.4 The final prototype

The end result of the design and development process was a prototype software application running on an Android tablet. This Android application has a user interface which allows wireless control of software imitations of the three different Aalberg Audio effect pedals described in section 1.2.1, running on a laptop computer. For each effect, the prototype allows a user to switch between two modes of control, which have been called the *sliders interface* and the *circle interface*.

The unique properties of these two interfaces, along with their connection to the desirable feature goals FG1 to FG6 on pages 26–27, are discussed first in sections 3.4.1 and 3.4.2. Subsequently, features that the two have in common are described in section 3.4.3. Finally, the prototype system's overall architecture is described in section 3.4.4. Implementation is glossed over, since it is not the focus of this thesis.

3.4.1 The sliders interface

Figure 3.5 on the next page shows the sliders interface for each of the three effects. Its design and usage is now described.

Design

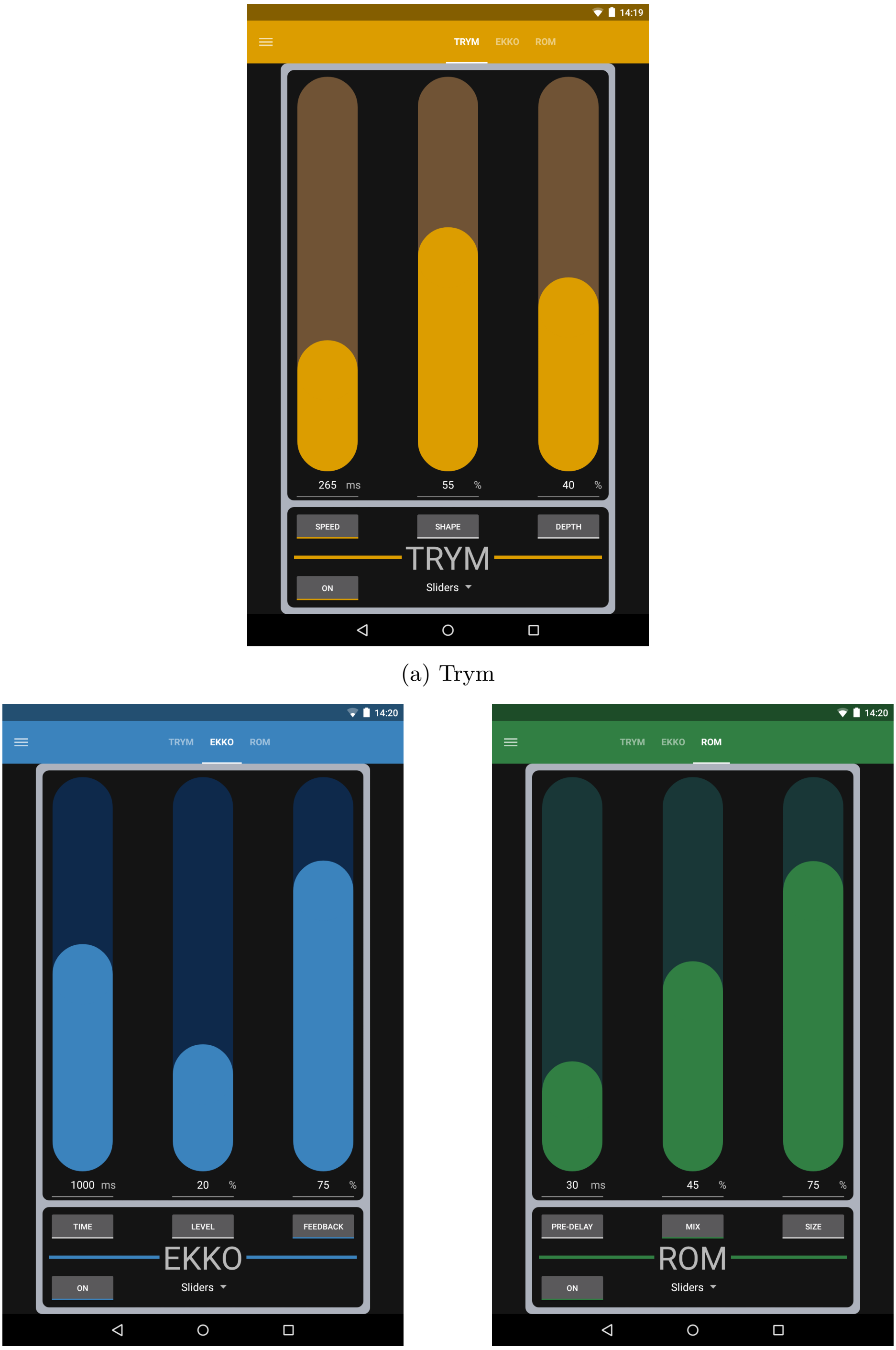
The sliders interface uses three vertical sliders to control the three sound parameters. It uses a one-to-one mapping, where the vertical position of each slider controls exactly one sound parameter. The interface should be easy to learn because of the one-to-one mapping. Because sliders are commonly used in many music technology devices, the interface should also feel familiar and intuitive to many users. The inclusion of this interface satisfies FG3.

Each slider can be said to move within a slider track. The slider has a bright colour, whereas the slider track has a darker colour. The sliders are designed to look similar to the level indicators used in Aalberg Audio's pedals, supporting FG6.

Each slider is placed directly above and is the same width as the numeric display and label/activation button for the parameter it controls. This provides a visual grouping, and should help make the user interface easy to understand.

Usage

Using the slider interface is very simple. Touching a finger anywhere within the area of each slider makes the slider jump there instantly. Moving the finger vertically while touching will also move the slider. The interface supports multi-touch, and thus all the sliders can be moved simultaneously using separate fingers, supporting feature FG2. The nature of the human hand does however make it difficult to move two sliders in different directions using two fingers on the same hand. Thus, FG4 has only partial support. FG5 can only be said to be supported when moving a single slider at a time.

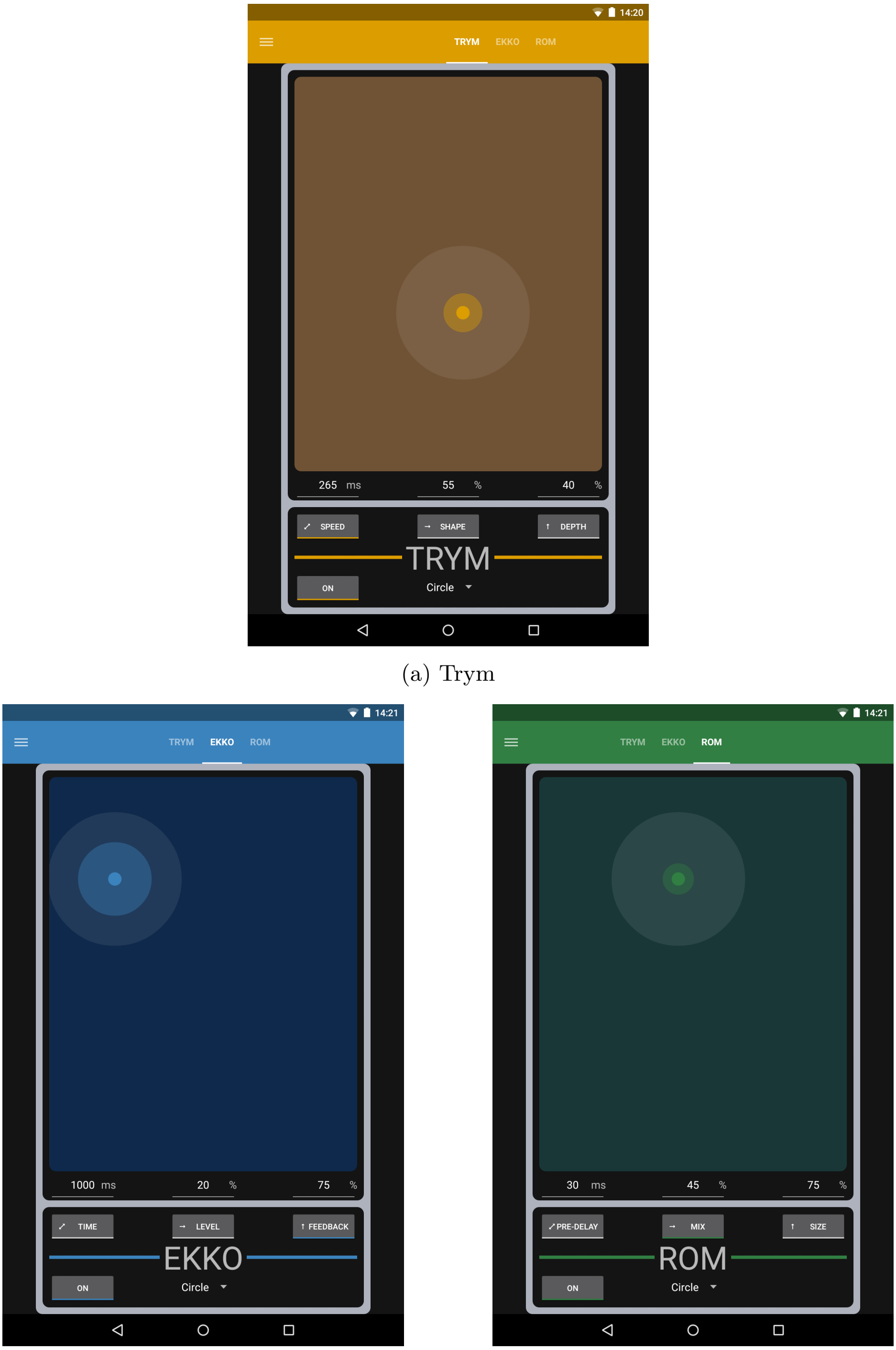


(a) Trym

(b) Ekko

(c) Rom

FIGURE 3.5: The sliders interface, shown for each of the effect three effects.



(a) Trym

(b) Ekko

(c) Rom

FIGURE 3.6: The circle interface, shown for each of the effect three effects.

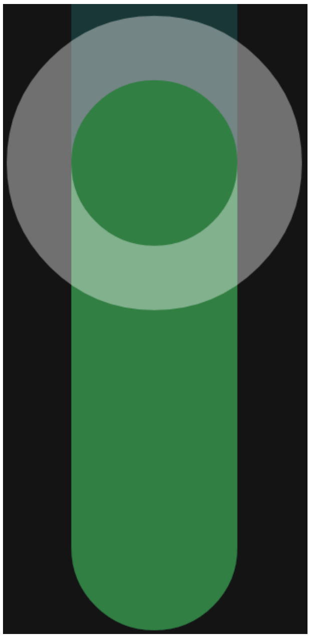


FIGURE 3.7: A slider being touched. Notice the touch indicator—a translucent band surrounding the top of the slider.

FIGURE 3.8: A detail view of the bottom area of the circle interface. The top button row illustrates the use of arrow icons to communicate control-to-sound mappings.

While the user is touching a slider, the top of the slider gets highlighted with a translucent white band called the touch indicator. This is shown in figure 3.7. The touch indicator serves as visual confirmation that the user's touch had an effect. When the finger is released, the touch indicator fades away.

3.4.2 The circle interface

Figure 3.6 on page 30 shows the circle interface for each of the three effects. Its design and usage is now described.

Design

In the circle interface, a circle is placed inside a rectangular area. The circle's position and radius is used to control the three sound parameters. A combination of one-to-one and one-to-many mappings are used: the circle's radius maps onto a single sound parameter, whereas the circle's position maps onto the two remaining sound parameters. The latter is done through a simple layered mapping: the circle position maps onto a horizontal and a vertical value, each of which maps onto a single sound parameter.

As shown in figure 3.8, icons are paired with sound parameter names to show the mappings. The diagonal, two-headed arrow symbolizes circle radius; the horizontal arrow symbolizes horizontal circle position; and the vertical arrow symbolizes vertical circle position.

Although the circle interface is not truly multiparametric—it lacks many-to-one mappings—it should provide many of the same benefits, supporting FG1. It is

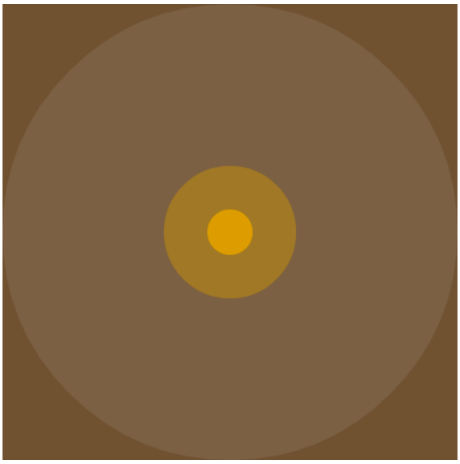


FIGURE 3.9: A detail view of the circle, illustrating the three parts: the outer touch area indicator, the resizeable level indicator, and the inner position indicator.

more complex, thus represents a challenge and allows mastery. Additionally, the relationship between control parameter and sound parameter is less clear, even with the use of icons. This should make it easier to enter the holistic mode of thought, thinking of a sound not in terms of individual parameters but as a whole.

The design allows simultaneous control of two sound parameters by using a single finger to move the circle around, supporting FG2 (partly), FG4, and FG5. The original plan was to allow the user to resize and move the circle at the same time, thus allowing simultaneous control of all three sound parameters. However, this feature was hard to implement properly, and had to be dropped due to time constraints.

The circle itself consists of three differently-shaded parts, shown in figure 3.9. The innermost and brightest part is a small, fixed-size *position indicator* circle which serves to make the position of the circle more explicit by clearly defining the circle's centre. It also serves as a lower bound for the level indicator (described next), clearly marking the minimum possible radius.

Around the position indicator a band in a more muted shade can be seen. This is the *level indicator* circle, which renders underneath the position indicator. It is this circle that actually grows and shrinks when a user resizes the circle or otherwise changes the associated parameter. Its radius is always equal to or larger than the position indicator, and smaller than or equal to the touch area indicator (described next).

Finally, the outermost *touch area indicator* circle is rendered underneath the level indicator. Its main function is to clearly mark the area which a user can touch to move the circle around. It also serves to clearly mark the maximum possible radius of the level indicator.

Usage

As mentioned, the user controls sound parameters by moving the circle and by changing its size. Each of these actions can be done in several ways.

Moving the circle There are two ways to move the circle:

1. A user can touch anywhere *within* the touch area indicator, and then drag the circle to the desired position. The touch area is thus suitably large even though the circle might be at its minimum size. Touching the touch area indicator also gives it a white highlight similar to the slider interface's touch indicator (figure 3.7), which gives the user visual feedback that the action had an effect.
2. A user can also touch anywhere *outside* the touch area indicator and hold the finger roughly in the same place for a little under a second. While the user does this, a new, small touch indicator circle starts growing under his or her finger. When this new indicator reaches the full touch area indicator size, the circle immediately moves into its centre. If the user lifts the finger or moves it outside the growing indicator's area before that point, the action is cancelled and the new indicator fades away.

Both of these actions are done using a single finger of one hand, supporting FG4 and FG5.

Changing the size The size or radius can also be changed in two ways:

1. A user can use the standard pinch-in and pinch-out gestures anywhere within the rectangle describing the circle position bounds. Pinch-in involves placing two fingers on the screen and moving them towards each other. This makes the circle smaller. Pinch-out similarly involves placing two fingers on the screen, but this time the fingers are moved away from each other. This makes the circle larger. Allowing the gesture to be done anywhere on the rectangle means there is enough room to go from minimum to maximum circle size, and vice versa.

FIGURE 3.10: The toolbar, with a button to open the navigation drawer and a tab for each connected effect. The *Ekko* effect is currently selected.

2. A user can also use the *quick scale* gesture, which was standardized in Android 4.4, anywhere within the touch indicator area. This involves tapping twice on the touch indicator area in quick succession and, on the second tap, holding the finger down. As long as the finger is held down, moving the finger upward makes the circle smaller. Similarly, moving the finger downward makes the circle larger.

Both of these actions can be done using one hand, supporting FG4; the second is done using a single finger, supporting FG5.

3.4.3 Common features

Some features are shared between both interfaces. This includes the control interfaces' top toolbar, navigation gestures, and bottom area; and the general design cues taken from Aalberg Audio's hardware effect pedals. All the actions described here can be done with a single finger, thus supporting FG4 and FG5.

Toolbar

The top of the interface consists of a toolbar, shown in figure 3.10. This toolbar has a "hamburger icon" for opening a navigation drawer, and a tab for each of the connected effects. The currently visible effect is highlighted and takes on that effect's primary colour, as shown in figures 3.5 to 3.6 on pages 29–30. Pressing one of the other tabs slides the control interface for that effect into view.

The navigation drawer was planned to be similar to the one in the horizontal prototypes (see section 3.2.3), but no functionality was implemented in this prototype.

Navigation gestures

In addition to the tabs in the toolbar, a user can navigate between effect by swiping (almost) anywhere on the screen with a single finger. Swiping left-to-right moves the entire effect interface to the right, while gradually revealing another effect from the left; vice versa when swiping right-to-left. When lifting the finger, the view snaps onto the effect that occupies the most screen real estate at the time, moving the other effect out of view.

The three exceptions to the swipe-anywhere rule are: the user cannot initiate a swipe from anywhere within the circle interface's touch area indicator[1]; nor can a swipe be initiated after a slider has started moving; finally, swiping from the left edge of the screen opens the navigation drawer.

Bottom area

Both the effect control interfaces share a common bottom area, shown in figure 3.12 on page 38. As can be seen, the only difference is the inclusion of arrow icons on the parameter label buttons for the circle interface.

The top row of the common area is used to display the numerical value for each of the parameters. The three numeric displays are actually number input fields, and allow a user to enter precise parameter values using the tablet's software keyboard overlay.

The row below contains three parameter label buttons. These serve two roles. First, they function as labels for the numeric displays, and for the sliders in the sliders interface. Second, they show which parameter, if any, is selected as the *active parameter* for control using the Aero controller (see section 1.2.2). This is indicated using a vertical strip of the effect's primary colour. In figure 3.12, *Mix* is currently the active parameter. Pressing a label button that is not currently selected will activate that parameter and deactivate any other; whereas pressing a label button that is currently selected will deactivate it.

The third row simply displays the name of the effect being controlled.

Finally, the bottom row contains a button for toggling the effect off (putting it in bypass mode) or on, and a so-called spinner for changing the interface mode of the current effect. The spinner is illustrated in figure 3.13. The on/off button is labelled with the state it is currently in: when the effect is on, the button displays "On" and has a vertical strip in the effect's primary colour. When the effect is off, it displays "Off" and has a grey coloured strip.

An earlier prototype had a *tap tempo* button on the right side of the bottom row. This was removed before the final evaluation, to focus more on the differences of control between the two modes. If the software is to be released to users, the button should be included again.

[1] Allowing swiping from the circle's touch area would make it difficult to move the circle

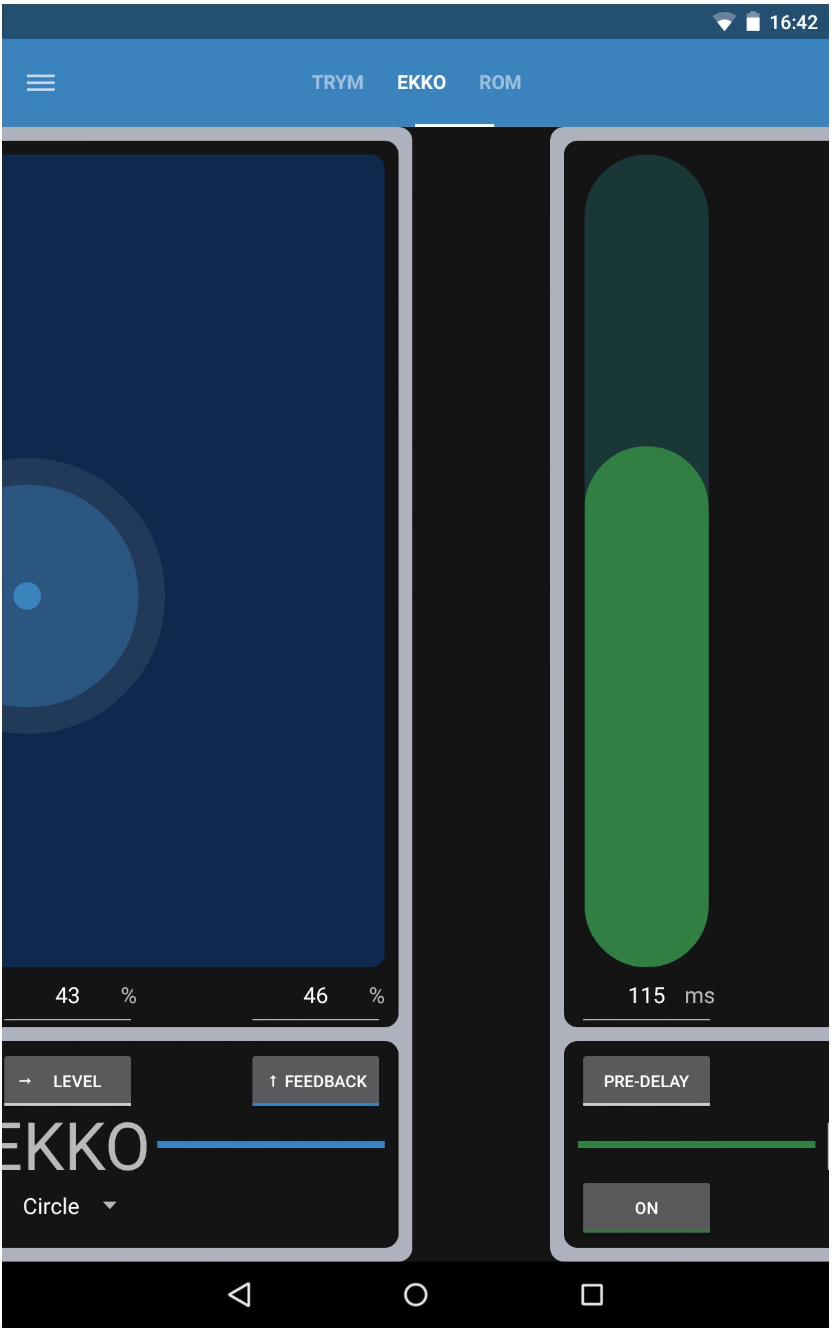
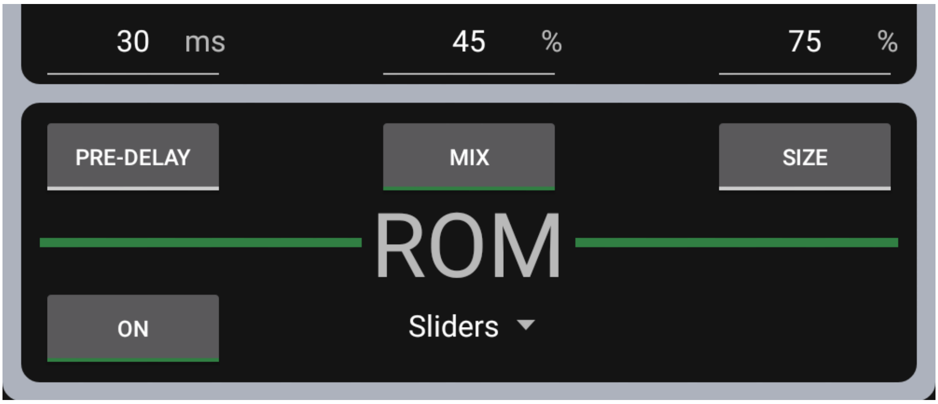
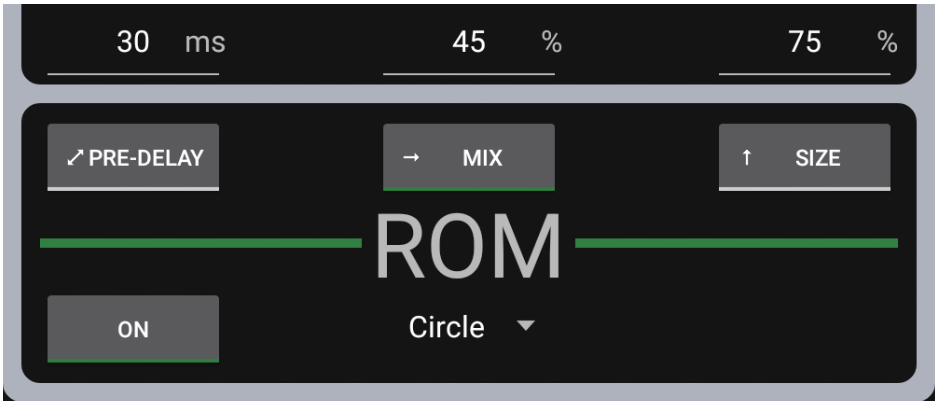


FIGURE 3.11: The interface in the middle of a right-to-left swipe gesture from the *Ekko* effect, revealing the *Rom* effect to the right.

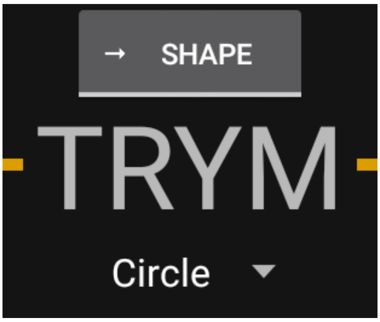


(a) Sliders interface

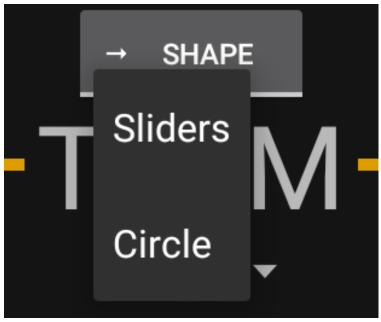


(b) Circle interface

FIGURE 3.12: The common bottom area. The top row displays numerical values, the second row displays parameter label buttons, the third row displays the name of the effect, and the bottom row contains a bypass switch and an interface mode selection.



(a) Closed spinner. Pressing it opens a menu allowing mode selection.



(b) Opened spinner. Pressing one of the choices activates that interface mode.

FIGURE 3.13: The interface mode spinner, which allows changing between the slider interface and the circle interface for the current effect.

General design cues

To promote a coherent and recognizable design between the software application and the hardware effects, several design cues were borrowed from the latter.

- The effect interfaces feature a metal grey outline with rounded corners, and use a black background for the main control area and the bottom area.
- The bottom area of each effect interface is laid out similarly to the bottom area of the hardware effects. The interface's parameter label buttons are in the same place as the effect pedal's parameter labels; the effect's name is displayed in the same way; and the interface's on/off button and (when it was present) tap tempo button are in the same position as the respective hardware foot switches
- Each effect has a unique colour scheme based on the colours shown in figure 1.2 on page 4. The colour scheme for the currently visible effect is used on the Android status bar and the application's toolbar, and as also features as an accent in some way when interacting with most user interface elements.

All of this should ensure that FG6 is satisfied.

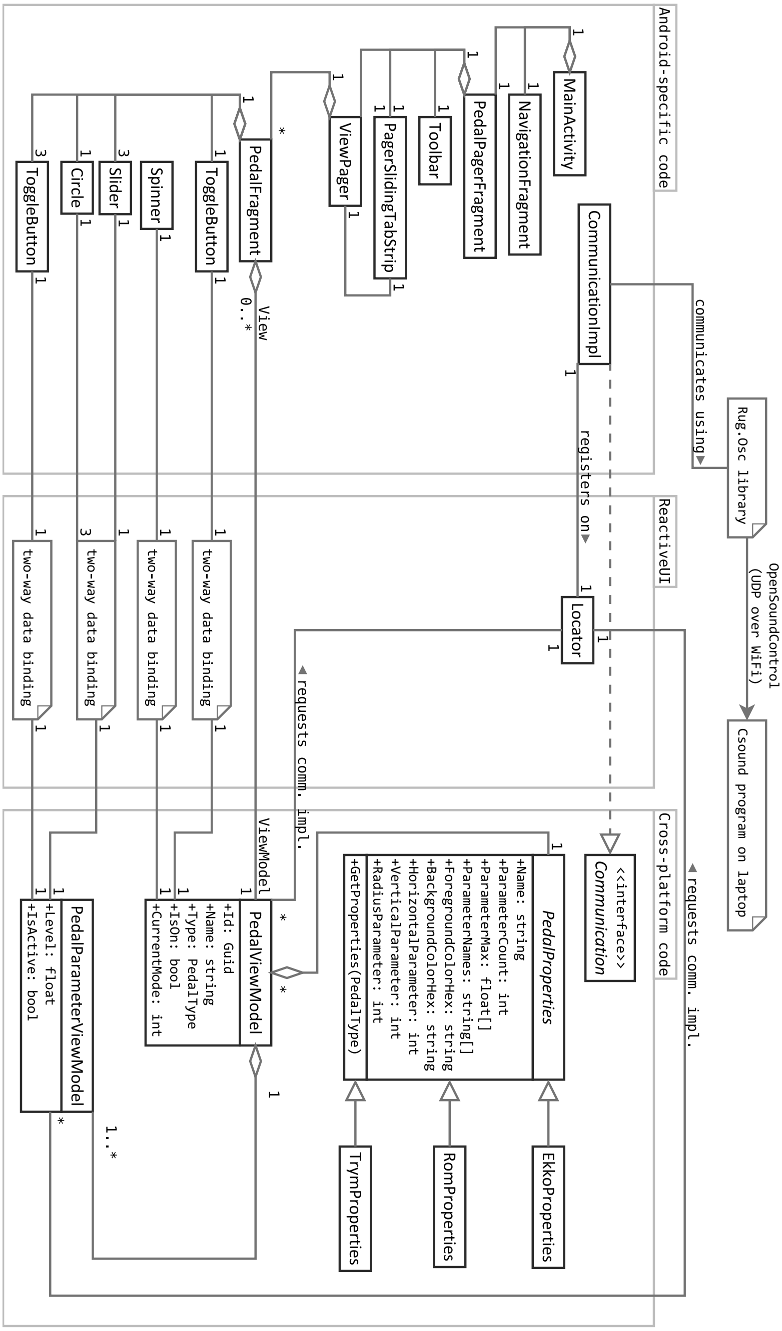


FIGURE 3.14: Simplified diagram of the final prototype's architecture. Standard UML notation is used.

3.4.4 Prototype system architecture

Figure 3.14 on the preceding page shows a simplified view of the final prototype's architecture, using standard UML notation. The main part of the diagram describes the software application, which consists of the two UML packages *Android-specific code* (left) and *Cross-platform code* (right), with the UML package for the *ReactiveUI* framework functioning as glue between them. Above these three packages, the wireless communication (using OpenSoundControl) with the audio processing software (implemented with Csound) is indicated.

In the following sections, the application architecture will be described first. Then, the technology used to implement the architecture is briefly mentioned. Finally, the implementation of the audio processing software is described.

Application architecture

The application is structured using a *Model View ViewModel (MVVM)* architecture. As implemented here, *views* represent platform-specific user interface code, whereas *view models* represent cross-platform abstractions of the user interface. A single view model can be used by multiple views. *Models* represent the actual state of the system. The final, crucial part of a MVVM architecture is data binding, keeping the view and view model synchronized.

The ReactiveUI framework[1] is used to implement the MVVM architecture. Among other things, it facilitates declarative data binding between views and view models, and also enables view model behaviour to be specified declaratively.

The most important view is shown in the diagram, namely the `PedalFragment`, which represents the collection of UI controls for a single effect pedal. The two most important view models, the `PedalViewModel` and the `PedalParameterViewModel`, are also shown. A `PedalFragment` binds each of the UI controls to properties of a view model. ReactiveUI's two-way data binding ensures changes to a property, no matter the source, is propagated to all controls bound to that property.

The application contains no actual model, as the system state that is being controlled lies outside the application. However, the communication layer between the view models and the effects can be seen as the model. The view models use a platform-specific implementation of the `Communication` interface to propagate view model changes to the effects. In this prototype, communication is one-way (view model to effects). Communication would, however, have to go both ways in a finished system.

[1] http://reactiveui.net

Each of Aalberg Audio's effects are described by a subclass of the abstract `PedalProperties` class, and specify things like effect name, colour scheme, sound parameter names, maximum values for each sound parameter, and the mapping between circle interface control parameters and sound parameters. This makes it easy to extend the system with new effect pedals, without having to code new platform-specific user interfaces.

Because of differences in various platforms' network stacks, communication has to be implemented separately on each platform. To let the cross-platform view models access the communication layer, a platform-independent `Communication` interface is exposed, which all platforms must implement. The *service locator pattern* is used to let each platform register its implementation when the application starts. A cross-platform view model then simply requests a suitable implementation from ReactiveUI's `Locator` when it needs to communicate.

Technology used in the application

Both the platform-specific and the cross-platform parts of the application are implemented in the C♯ programming language by using Xamarin Platform[1]. The Android user interface itself uses a Xamarin.Android port of PagerSlidingTabStrip[2] to implement the tabs for each effect. The Android implementation of the Communication interface uses Rug.Osc[3] library for wireless communication via OpenSoundControl (OSC). It uses the User Datagram Protocol (UDP) to transfer OSC messages to an OSC server on the network. Finally, the ReactiveUI framework is used, as described earlier.

Audio processing software

The actual audio effects are implemented as a software programming running on a computer, using the audio programming language Csound[4]. Csound's built-in OSC server functionality is used to allow communication with the tablet application. Each of Aalberg Audio's effect pedals described in section 1.2.1 are mimicked, and have the same sound parameters and the same maximum parameter values (varying minimum values are not implemented—all parameters ranges start at zero).

[1] http://xamarin.com/platform
[2] https://github.com/jamesmontemagno/PagerSlidingTabStrip-for-Xamarin.Android
[3] https://bitbucket.org/rugcode/rug.osc
[4] http://csound.github.io/about.html

These effect implementations make use of various standard Csound *opcodes*. All of them use `OSClisten`[1] to receive OSC messages for each sound parameter, including bypass.

The tremolo effect uses `ftgen`[2] to generate sine and square wave function tables. Two `oscil`[3] opcodes function as the LFOs, generating low-frequency sine and square signals of the correct *speed* using the function tables. By default, the signals vary between -1 and 1, but they are then scaled to vary between $1 - (depth)$ and 1. Morphing between sine and square wave shapes is done by using `ntrpol`[4] to calculate a weighted mean value of the two LFO signals, with *shape* as the weight. Finally, the input audio signal is multiplied with the morphed LFO signal to achieve the tremolo effect.

The delay effect uses `delayr`[5], `deltap3`[6], and `delayw`[7] to respectively read from, tap at a given offset *time*, and write to a delay line. The input signal and the delayed signal obtained using `deltap3` is written back to the delay line, with the latter multiplied with the *feedback* coefficient. Finally, the input signal is summed with the delayed signal, the latter multiplied by the *level* coefficient, and sent to the output.

The reverb effect feeds the input signal through the `reverbsc`[8] opcode, which implements reverb using a feedback delay network (FDN). A lowpass filter internal to the opcode is set with a cutoff frequency of 8000 Hz. The *size* parameter is mapped to the opcode's feedback level, but scaled so that it has a more useful response using the `logcurve`[9] and `scale`[10] opcodes. The reverb signal is then given the specified *pre-delay* by using a `delayr`, `deltap3`, and `delayw` delay line. Note that this does *not* involve feeding the delayed signal back into the delay line, as was done for the previous effect. The delayed reverb signal (the "wet" signal) is finally mixed with the "dry" input signal using `ntrpol` to calculate a weighted mean, with the *mix* parameter used as weight.

Taking a higher view, the program reads an audio signal from an input channel on a sound card; routes it through the three effects, with output from one effect fed as input into the next; and writes the resulting signal to a sound card output

[1] http://www.csounds.com/manual/html/OSClisten.html
[2] http://www.csounds.com/manual/html/ftgen.html
[3] http://www.csounds.com/manual/html/oscil.html
[4] http://www.csounds.com/manual/html/ntrpol.html
[5] http://www.csounds.com/manual/html/delayr.html
[6] http://www.csounds.com/manual/html/deltap3.html
[7] http://www.csounds.com/manual/html/delayw.html
[8] http://www.csounds.com/manual/html/reverbsc.html
[9] http://www.csounds.com/manual/html/logcurve.html
[10] http://www.csounds.com/manual/html/scale.html

channel. The signal flow is:

$$\text{Input} \rightarrow \text{Tremolo} \rightarrow \text{Delay} \rightarrow \text{Reverb} \rightarrow \text{Output}.$$

3.5 Evaluating the final prototype

This section describes the methods used to evaluate the final prototype. An overview of the process is given in section 3.5.1, followed by a description of the participant selection in section 3.5.2. Next, the experimental context is described in section 3.5.3. Finally, the experiment itself is detailed step-by-step in section 3.5.4. The results of the evaluation are presented in chapter 4 on page 55.

3.5.1 Overview of the evaluation process

Based on the literature on evaluating digital musical instruments (section 2.3), an evaluation process was constructed. The process featured an experiment where users were given musical tasks with appropriate contexts. Qualitative data were collected through observation of users while performing the tasks, and interviews afterwards. The interviews were recorded. For some tasks, quantitative data were also collected during task performance. Finally, the data were analysed.

The tasks, contexts, and interview questions were designed to compare the playability, discoverability, and enjoyment of the two user interface modes, in order to ascertain whether the design principles taken from the literature on digital musical instruments had similar effects on an interface for guitar effect control. This will be described in more depth in later sections.

3.5.2 Participant selection

The evaluation was done with the help of 7 participants. All of them were electric guitar players of varying experience. Four of the participants were selected by a convenience strategy, i.e. they were people to whom the author had easy access. These were either friends of the author or people who worked for Aalberg Audio in some capacity. The remaining three participants were professional musicians whom Aalberg Audio had contacted in order to get feedback on the final prototype.

Selecting participants in this way resulted in a group with wide-ranging backgrounds. However, all the participants ended up being male. Although this author does not believe the results of the evaluation would be very different, it is a flaw of the process worth mentioning.

3.5.3 Experimental context

The experimental context can be said to consist of the information given to participants in advance of the evaluation; the environment in which the experiment was performed; and the setup of the prototype system being used in the experiment. These three parts are now described.

Information given in advance of the experiment

Participants were sent an email thanking them for their participation; describing the purpose of the experiment; and broadly outlining the different parts of the experiment and their approximate time requirements. These parts will be described in section 3.5.4. The email also expressed the wish to record audio from the interviews, and gave assurances that any data collected would be anonymized.

The experiment environment

For the first 5 of the 7 participants, the experiment was performed in a room in the offices of Aalberg Audio. Only the author and a single participant were in the room for each run of the experiment.

For participants 6 and 7, the experiment was performed in their respective homes. For participant 6, no one besides the author and the participant was present. In the case of participant 7, the participant's significant other was also present, though not focused on the experiment.

In all cases, participants were allowed to sit or stand as they wanted during the experiment. When not giving instructions, the author sat to the side and observed silently.

Prototype system setup

For the purpose of the evaluation, the prototype system was set up as shown in figure 3.15. To control the audio effects, each participant used touch gestures to interact with the prototype application running on a 7-inch tablet—a Google Nexus 7[1] running Android 5.0.2—which was mounted on a microphone

[1] http://en.wikipedia.org/wiki/Nexus_7_(2013_version)

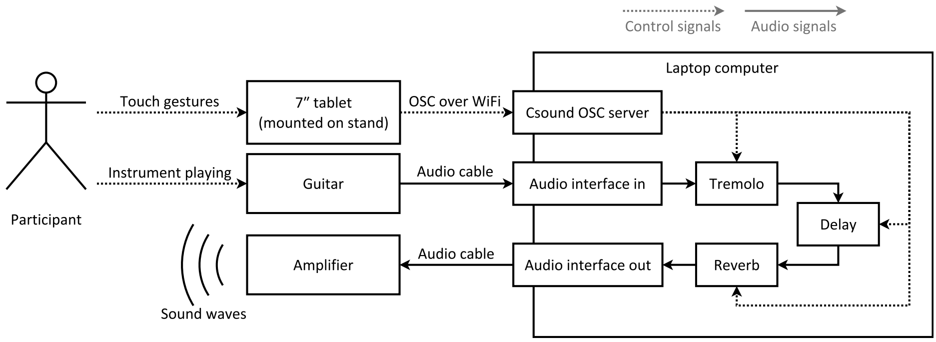


FIGURE 3.15: How the prototype system was set up during evaluation.

stand using a König & Meyer tablet PC holder (model 19740)[1]. The resulting OpenSoundControl messages were sent over the wireless network to the Csound program running on the laptop computer. While controlling the effects, each participant also played an electric guitar, which was plugged into an M-Audio Fast Track Ultra USB audio interface connected to the laptop computer. This input signal was routed through the audio effects implemented in Csound, whose output was sent to a guitar amplifier.

3.5.4 The experiment

The participants were assigned to one of two groups: a "Sliders" group and a "Circle" group. The "Sliders" group consisted of participants 1, 3, 5, and 7, whereas the "Circle" group consisted of participants 2, 4, and 6. A counterbalanced within-subjects design was used for the experiment: Each group tested both interfaces, but the order in which the interfaces were used was varied. Everyone in the "Sliders" group were shown the slider interface first, and used that interface first for each part of the test. Conversely, participants in the "Circle" group were shown the circle interface first, and each part of the test was first done using that interface. This was done to minimize the influence of any potential effects, such as learning, which might arise from the order in which the interfaces were tested.

Each experiment session consisted of 5 parts. Following a short introduction to the procedure, there was an explanation and short demonstration of the two interfaces. The bulk of the session consisted of a structured exploration of the

[1] http://produkte.k-m.de/en/Multimedia-Equipment/Holders/19740-Tablet-PC-holder-black

interfaces, a set of target acquisition tasks, and finally an interview. Each part will now be described

Introduction

The session started with introductions and small talk, to make the participant feel at ease. The participant was thanked for his willingness to take part in the experiment. Next, the participant was informed about the purpose of the experiment: comparing two different user interfaces for remotely controlling guitar effects. It was made clear that the focus was on the control experience, not on sound quality; and that the interfaces was what was being tested, not the participant.

After this, the participant was given a brief overview of the remaining parts of the experiment, and was encouraged to ask questions if anything was unclear.

The allotted time for this part was 2 minutes.

Explanation and demonstration

It was explained that the two interfaces were prototypes, lacking the polish and features of a finished application. Next, the interfaces were presented. Participants in the "Sliders" group was shown the sliders interface first; those in the "Circle" group was shown the circle interface first. For each of the interfaces, the method of control and the mapping was explained. The participant was asked to perform each of the touch gestures used in each interface, to confirm that they understood how to use them. The participant was also briefly introduced to the three different effects (tremolo, delay, reverb).

The allotted time for this part was 3 minutes.

Structured exploration tasks

Upon completing the explanation and demonstration, the participant was asked to pick up the guitar and explore both the effects and the interface. For each interface (sliders or circle) and effect combination, the participant was tasked with finding new and interesting sounds, with the context that he was trying to find a sound for a new song he was making.

The purpose of this part of the experiment was to simulate the idea generation stage of creativity, which should provide the participant with experiences regarding the prototype's discoverability. In finding new sounds, the participant had to

TABLE 3.1: Target parameter values on the delay effect for the target acquisition tasks.

Task	Time	Level	Feedback
1	300 ms	80%	60%
2	1240 ms	30%	70%
3	600 ms	50%	50%
4	150 ms	40%	0%

play with the prototype as a part of the larger instrument, which should provide experiences regarding playability. Finally, the general use of the prototype should give the participant a sense of its enjoyment.

The exploration tasks was structured so that the participant was given 2.5 minutes to explore each effect, in order, with the interface he was shown first. Before starting the exploration of an effect, the parameters of all effects were reset to zero, so that only the effect being explored had any influence on the sound. This took a total of 7.5 minutes. Then the same procedure was repeated, but with the interface he was shown last. In total, this amounted to 15 minutes of structured exploration.

Target acquisition tasks

After the structured exploration tasks came a set of four target acquisition tasks. To provide context for the tasks, the participant was asked to imagine a situation in which he knew what sound he wanted, but did not know what the parameter values were. The tasks tried to simulate such a situation by using the following procedure:

1. The effect parameters were set using a program on the computer, without the user interface being updated.
2. The participant was given 30 seconds to play the guitar and listen to how the effect influenced the sound.
3. The effect parameters were reset to zero.
4. The participant was given 1 minute and 30 seconds to recreate the sound using one of the interfaces.

In total, this part of the experiment took 8 minutes.

All four tasks involved only the delay effect. The parameters used for each task are given in table 3.1.

Participants in the “Sliders” group performed tasks 1 and 2 using the sliders interface, and tasks 3 and 4 using the circle interface. Conversely, participants in the “Circle” group performed tasks 1 and 2 using the circle interface, and tasks 3 and 4 using the sliders interface.

The purpose of this part of the experiment was, as mentioned, to simulate a situation in which a desired sound is known but the parameter values are not. Such a situation might for instance occur when switching from one planned sound to another during a concert performance. This should provide the participant with experiences regarding the prototype’s playability. As before, the general use of the prototype should also give the participant a sense of its enjoyment.

Interview

Finally, the participant was asked to put away the guitar and, if he had done the previous tasks while standing, to sit down. He was then informed of the purpose of the interview: collecting background information about him, and getting his feedback on the prototype’s two user interfaces. Then he was asked permission to record the interview using an audio recording device. Assurances were given that the recordings were confidential and that the transcript would be anonymized.

The interview then commenced. It was a semi-structured interview. Such an interview allows a combination of pre-determined questions and prompts; impromptu questions about actions performed or problems met during the tasks; and clarifying follow-up questions and discussion. The tone was largely conversational throughout. To avoid a confrontational setting that can occur when directly facing each other, chairs were oriented so that the interviewer and the participant were sitting at approximately a 90 degree angle.

After the interview was concluded, the participant was again thanked for taking part. He was again assured of confidentiality and anonymity, and was asked if he wanted the transcript sent to him for checking.

Interview questions

The following questions were used for the interviews. They have been translated from Norwegian, the language in which the interviews were conducted.

1. How old are you?
2. What is your occupation?
3. Can you briefly tell me about your musical background?

- Probe: How long have you played the guitar?

4. How would you describe your knowledge about the way guitar effects work?
   - Probe: For instance, maybe you know only a little about what parameters do, or maybe you know how different effect parameters influence the sound. Maybe you even know how signal processing works and how to make implement guitar effects.
5. How do you relate to new technology and new products on the guitar and guitar effects market?
   - Probe: For instance, maybe you are a "gear hound" who wants to try everything new, or maybe you are satisfied with your current setup and not interested in changing anything.
6. Can you briefly describe your experience with the slider interface?
   - Probe: Ask about observed actions and problems
7. Can you briefly describe your experience with the circle interface?
   - Probe: Ask about observed actions and problems
8. How did the interfaces compare to regular guitar pedals?
9. How would you compare the two interfaces regarding
   (a) how easy they were to use?
   (b) how good they were for finding specific sounds?
   (c) how good they were for finding new sounds?
   (d) how well you liked them?
10. Which interface do you think would be most suitable
    (a) for a concert? Why?
    (b) for practice? Why?
    (c) when composing music? Why?
11. If you had the opportunity, would you use either or both of these interfaces yourself? Why or why not?
12. Is there anything you feel is relevant and would like to add?

The purpose of questions 1 through 5 was to gather background information about the participants. Additionally, questions 1 and 2 are easy to answer, whereas questions 3 through 5 get progressively more reliant on reflection. This was done in order to warm up the participants to the interview situation and get used to having to think about their answers. For questions 3 through 5, probes were used if necessary to ensure some degree of similarity between the information gathered for each participant.

Questions 6 and 7 were used to get the participants to reflect on their experiences with using the interfaces. The answers given here should help with evaluating the playability, discoverability, and enjoyment of the prototype. For participants in the "Circle" group, the order of these two questions was reversed.

Question 8 was used to see whether participants found any advantages or disadvantages with controlling effects using the prototype interfaces versus the controls on regular guitar pedals. The purpose was to see whether participants actually found the prototype useful compared to the guitar effect systems they were used to using.

The purpose of question 9 was to find out whether the chosen mapping strategies, when used for a guitar effects control system, led to the results one would expect for a digital musical instrument. In the latter case, one would expect the one-to-one mapping (sliders interface) to be easier to use (9a). One would also expect it to be better for finding specific sounds (9b), due to its more direct support for analytical thinking. In contrast, the multiparametric mapping (circle interface) would be expected to perform better during idea generation (9c). The expectation for overall preference (9d) is less clear: it would be expected that a few would prefer the sliders, although most would prefer the circle or would use both.

Question 10 was used to find out which interface worked best in the situations of a concert, practice, and composition. The thinking here was that these situations might impose different requirements on the user interface. For instance, during a concert playability and efficiency is paramount. Composition would probably benefit from increased discoverability, while having room for interactions taking more time. A practice situation might be a midpoint between the two, as it might require a musician to both do quick, pre-planned adjustments; and to engage in creative acts like jamming, trying out new sounds, and group composition. In all cases, enjoyment is important.

Question 11 was used to find out whether the participants actually saw any value in remotely controlling guitar effects from a tablet.

Finally, question 12 was an invitation for the participant to share thoughts and knowledge which did not fit into any of the previous questions.

3.5.5 Data collection

Data from the experiment were collected in three ways. First, each interview was recorded using a Zoom H4n portable audio recorder. Second, notes were taken of events of special interest that occurred during the exploration and target acquisition parts of the experiment. This included things that the participants had problems with, and remarks made by participants. Finally, the parameters reached at the end of each target acquisition task were noted.

3.5.6 Data analysis

The qualitative interview data and the quantitative task parameter data were analysed in different ways.

Qualitative data analysis

The recorded interviews were transcribed manually. After reading through the transcripts, two of the participants' answers were deemed to be missing some background information. These participants were contacted via email, and the answers were appended to their interview transcripts. All transcripts were then imported into the analysis software QDA Miner Lite[1]. Using this software, the transcripts were coded following an open coding strategy. This involved grouping statements made during the interview using codes in the participants' own words. Codes that described similar statements were merged. Finally, the coded statements were grouped in different categories.

An overview of the codes and categories are given in appendix A on page 77.

The observation notes ended up being very sparse because most of the researcher's focus was on administrating the experiment correctly. The notes are summarized in the results section.

Quantitative data analysis

The numerical data collected during the target acquisition tasks were compared to the target parameter values, and a normalized error value for each parameter in each individual task was calculated. The normalized error was defined as

$$\frac{dist(v_t, v_a)}{v_{max}} \times 100\%,$$

[1] http://provalisresearch.com/products/qualitative-data-analysis-software/freeware

where v_t is the target value, v_a is the actual value reached, v_{max} is the maximum possible value for the given parameter, and $dist(x, y) = |x - y|$ is the Euclidean distance between two values.

Box plots comparing the normalized error values of the two interface modes were then created for each of the parameters on the delay effect used for the target acquisition tasks.

The calculations and plots were achieved by using the R software environment[1].

[1] http://www.r-project.org

Chapter 4

Results

This chapter presents the results of the experiment. Section 4.1 introduces the participants, then the quantitative and qualitative data from the experiment are presented in sections 4.2 and 4.3, respectively.

4.1 Introducing the participants

A short introduction to the participants follows; a summary is given in table 4.1 on page 57. In that table, P# is short for participant number.

Participant 1

A male 27-year-old music technology student and amateur musician.

Musical background: He plays several instruments. He has played the guitar for 13 years, although he mostly plays the keyboard. He plays heavy, guitar based music, and electronic music. Although he used to play in a band, these days he only plays in a studio setting.

Knowledge of effects: Very high. In addition to studying music technology, he has experience with implementing digital signal processing on microcontrollers.

Participant 2

A male 26-year-old music technology student and amateur musician.

Musical background: He has played the guitar for 11 years, and used to play in a metal band. These days he plays in a rock band, but only in a studio setting, never live.

Knowledge of effects: High. Has had courses on digital signal processing for music, and feels he has a good grasp of how effects work.

Participant 3

A male 32-year-old professional musician and composer.

Musical background: Has played the guitar for 15 years, and has played in two well-known rock bands who toured extensively. He also composes music for film and theatre.

Knowledge of effects: Very high practical knowledge. He considers guitar effects a central part of his sound, and is highly experienced with controlling them live while playing.

Participant 4

A male 34-year-old professional musician and music producer.

Musical background: He has played the guitar for 23 years. As a freelance musician he has worked with several famous artists, and has been a member of a popular rock artist's backing band for the past 10 years.

Knowledge of effects: High practical knowledge. He is keenly interested in guitar effects, uses them when playing, and is also highly experienced with using effects in the studio as a producer and technician. In addition, he reviews effects for an industry magazine.

Participant 5

A male 42-year-old professional musician.

Musical background: He has played the guitar for 31 years, and has played in a popular rock band for the past 23 years.

Knowledge of effects: Fair practical knowledge. He has long experience using analogue effect pedals, and now uses mostly digital effects and software. With his experience, he says he knows how various effect parameters affect the sound, although he doesn't know much about how they work.

Table 4.1: Summary of experiment participants.

P#	Group	Age	Occupation	Played	Effects knowledge
1	Sliders	27	Music technology student	13 years	Very high
2	Circle	26	Music technology student	11 years	High
3	Sliders	32	Musician / composer	15 years	Very high
4	Circle	34	Musician / producer	23 years	High
5	Sliders	42	Musician	31 years	Fair
6	Circle	35	Publishing editor	10 years	Fair
7	Sliders	32	College lecturer	12 years	Some

Participant 6

A male 35-year-old publishing editor and amateur musician.

Musical background: He is self-taught on the guitar, and has played for 10 years. He plays the guitar and sings in an indie folk band.

Knowledge of effects: Fair practical knowledge. He is very interested in analogue guitar pedals, and is fairly sure he knows how most of them influence his guitar sound, but he has no experience with and is sceptical towards digital effects.

Participant 7

A male 32-year-old college lecturer and amateur musician.

Musical background: He is a self-taught guitarist, and has played the guitar for 12 years. He plays a variety of guitars and guitar-like instruments in an indie folk band.

Knowledge of effects: Some knowledge. He describes his knowledge as “touch and go”. He uses only a limited set of effects, and says he “very rarely has the patience to really familiarize [himself] with how things work”.

4.2 Results from the quantitative data analysis

Tables 4.2 and 4.3 list the final parameter values reached by each participant for each task, grouped by which interface was used for the task. The tables also show the normalized error values as described in section 3.5.6. In these tables P# is the participant number, T is the reached time value, L is the reached level value,

TABLE 4.2: Parameter values and calculated errors for tasks performed with the sliders interface.

Task	P#	T (ms)	L (%)	F (%)	T_{err} (%)	L_{err} (%)	F_{err} (%)
1	1	290	29	69	0.5	51	9
1	3	307	56	52	0.35	24	8
1	5	314	57	50	0.7	23	10
1	7	264	64	62	1.8	16	2
2	1	1200	30	80	2	0	10
2	3	1097	45	63	7.15	15	7
2	5	1235	45	82	0.25	15	12
2	7	1410	43	35	8.5	13	35
3	2	563	36	51	1.85	14	1
3	4	619	46	43	0.95	4	7
3	6	556	76	41	2.2	26	9
4	2	125	32	9	1.25	8	9
4	4	147	50	0	0.15	10	0
4	6	243	10	10	4.65	30	10

and F is the reached feedback value. T_{err}, L_{err}, and F_{err} are the normalized error values for time, level, and feedback, respectively. The target values were described in table 3.1 on page 48.

The data were used to create the box plots in figure 4.1 on the next page, which compare the normalized error for each parameter across the two interfaces. The bottom and top of each box represent the first (lower) and third (upper) quartiles, respectively, whereas the second quartile (the median) is indicated using a thicker band inside the box. Lines extending vertically from each box represent variability outside the upper and lower quartiles.

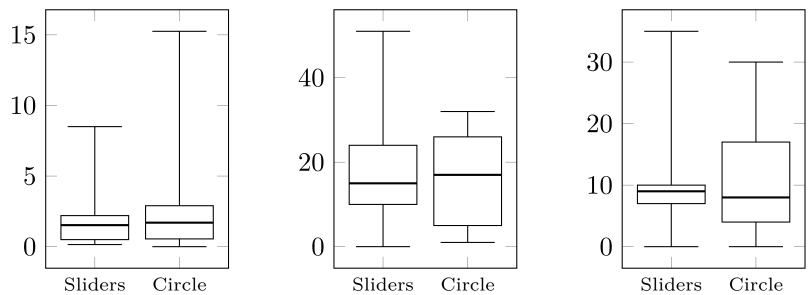
Of particular note in the plots are the medians of the normalized error for each parameter on each interface, which are shown in table 4.4 on the facing page.

4.3 Results from the qualitative data analysis

In the following sections, the observations notes from the experiments are first summarized. Then, results from the interviews are presented.

TABLE 4.3: Parameter values and normalized error for tasks performed with the circle interface.

Task	P#	T (ms)	L (%)	F (%)	T_{err} (%)	L_{err} (%)	F_{err} (%)
1	2	298	50	63	0.1	30	3
1	4	255	54	73	2.25	26	13
1	6	605	48	66	15.25	32	6
2	2	1232	59	53	0.4	29	17
2	4	1182	49	40	2.9	19	30
2	6	1011	47	48	11.45	17	22
3	1	—	—	—	—	—	—
3	3	572	54	42	1.4	4	8
3	5	705	57	69	5.25	7	19
3	7	579	51	46	1.05	1	4
4	1	150	20	5	0	20	5
4	3	188	29	0	1.9	11	0
4	5	139	43	0	0.55	3	0
4	7	116	45	10	1.7	5	10



(a) Time error (%) (b) Level error (%) (c) Feedback error (%)

FIGURE 4.1: Box plots comparing normalized errors on the two interfaces for each of the three delay parameters.

TABLE 4.4: Median of normalized errors for each interface

	Sliders (%)	Circle (%)
Median time error	1.525	1.7
Median level error	15	17
Median feedback error	9	8

4.3.1 Observations

Only participants 2 and 7 spent much time using the double-tap-and-drag gesture for adjusting the circle size in the circle interface. The other five participants tried it during the demonstration, but did not use it during exploration or target acquisition tasks.

All the participants tried to do the pinch gestures for the circle interface with the touch area indicator circle or the level indicator circle as the starting point, even though it was explicitly stated during the explanation that the gesture could be done anywhere within the rectangle.

Participants 2 and 3 had problems with resizing the circle when it was in a corner.

4.3.2 Interviews

Statements made by the participants during the interviews were coded and categorized. The results that can be said to have some degree of agreement are presented below. For the full results, including all codes and code categories, along with an overview of their occurrence among the participants, refer to appendix A.

The sliders interface

Positive statements about the sliders interface were fairly similar. As can be seen in table A.4 on page 80, most participants (6 out of 7) described the sliders interface as intuitive or easy to understand. There was also wide agreement (5 out of 7) that it felt familiar, or was similar to a normal effect pedal. 5 of the 7 participants also described this interface as being visually clear. Additionally, 4 participants said the interface made it easy to "dial in" parameters (to precisely set parameters to some predetermined values).

Few negative comments were made specifically about this interface. In table A.5 on page 81, these are shown. There was no wide agreement on any of them.

The circle interface

Table A.6 on page 82 shows the positive statements regarding the circle interface. The responses are more varied than those about the sliders interface. Of note is that 5 of the 7 participants said the circle interface made it possible to discover sounds you otherwise wouldn't find, and 3 participants said it made adjusting multiple simultaneous parameters easy. The interface also prompted the use of various positive language like “fun”, “innovative”, “liberating”, “promising”, and “very cool”.

Some statements could be categorized as neutral. These are shown in table A.7 on page 83. There was no wide agreement.

The circle interface also elicited a variety of fairly specific negative statements. Most importantly, 4 of the 7 participants found it hard to adjust the circle’s size; 3 of these specified that it was hard to fine-tune the sound parameter mapped to the circle’s size. 3 participants also described that it was hard to know what parameters they were controlling. The interface was also described by 3 participants as having a learning curve.

Both interfaces

Some general statements regarding both interfaces were also made. These are shown in tables A.9 to A.11 on pages 85–86, grouped into positive, neutral, and negative categories respectively.

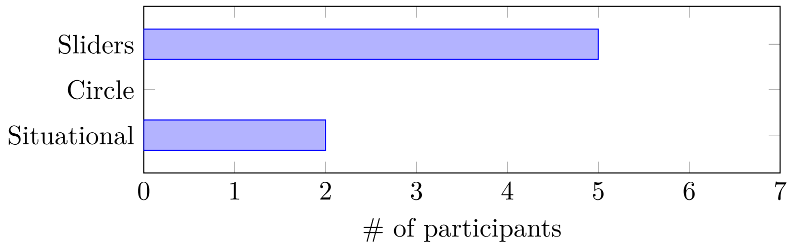
Of the positive statements, the most notable was that 3 participants thought it was nice to not have the effect controls on the ground.

The most notable of the negative statements was that 4 of the 7 participants described, in some way or another, that some controls would benefit from having a non-linear response rather than the linear response currently implemented.

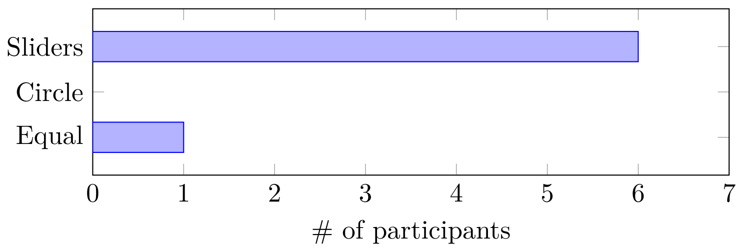
Comparing the interfaces

During the interview, participants were asked to compare the interfaces in various areas (question 9). In their answers, the participants expressed some preferences. The results are summarized in figure 4.2 on the next page, and are based on tables A.12 to A.15 on pages 86–87.

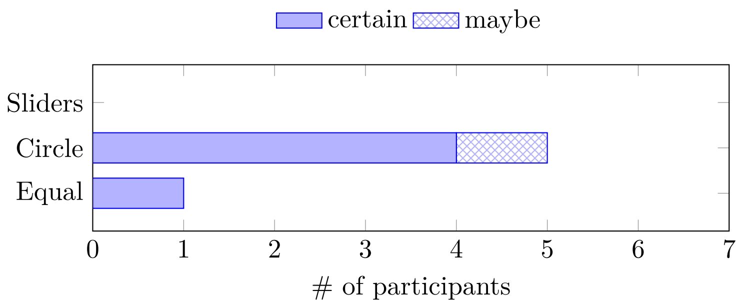
Participants were also asked which interface would be most suitable in various situations (question 10), and whether they would use the application themselves (question 11). These results are summarized in figures 4.3 and 4.4 on pages 63–64, and are based on tables A.16 to A.19 on pages 87–88.



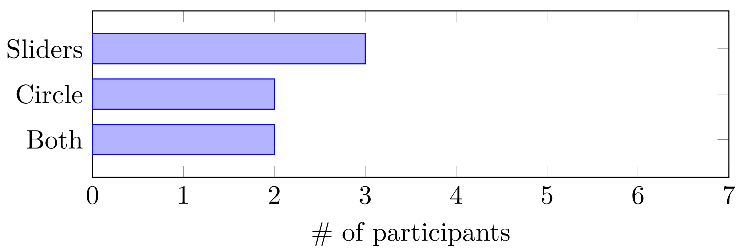
(a) Which interface was easiest to use?



(b) Which interface was best for finding specific sounds?

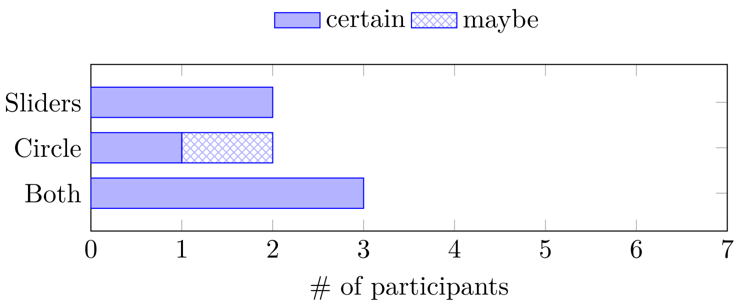


(c) Which interface was best for finding new sounds?

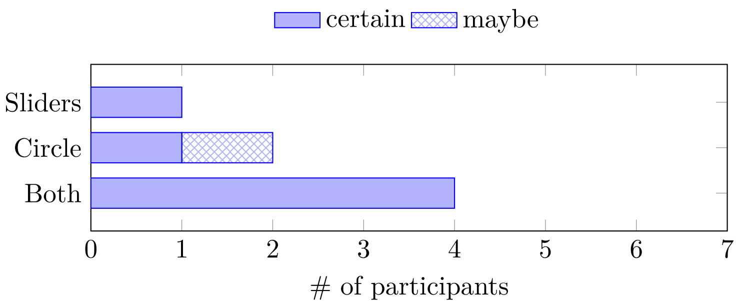


(d) Which interface was preferred overall?

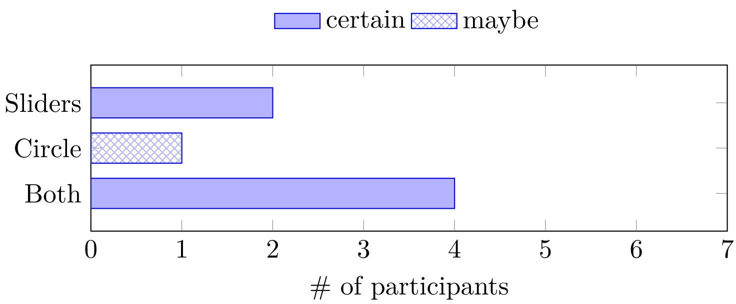
FIGURE 4.2: Preferences expressed in answers to interview questions 9 (a)–(d).



(a) Which interface was most suitable for concert use?

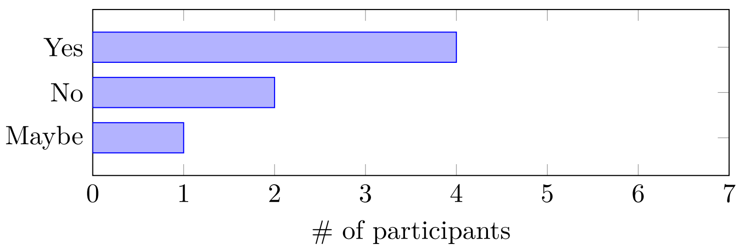


(b) Which interface was most suitable for practice?



(c) Which interface was most suitable for composition?

FIGURE 4.3: Summary of answers to interview questions 10 (a)–(c).



Would you use the application yourself?

FIGURE 4.4: Summary of answers to interview question 11.

Chapter 5

Discussion and conclusion

This chapter discusses the implications of the results presented in the previous chapter. First, the results are viewed from the performer perspective, and the achievement of goals for discoverability, playability, and enjoyment explored, in section 5.1. Then, the research questions posed at the beginning are answered in section 5.2. Flaws in the study are mentioned in section 5.3, and further work is outlined in section 5.4. Finally, section 5.5 concludes the study.

5.1 The results from the performer perspective

This section discusses what the results say about the prototype system's discoverability (section 5.1.1), playability (section 5.1.2), and enjoyment (section 5.1.3) from the perspective of performers.

Based on the results, it is hard to assess whether the prototype achieved increased playability, discoverability, and enjoyment compared to Aalberg Audio's existing control possibilities (*Aero* wireless controller and controls on the pedals). This is mainly due to limitations of the experiment. The Bluetooth Smart protocol had not yet been implemented in Aalberg Audio's effect pedals in a way that allowed communication with a tablet. The author thus had to implement software effects that mimicked the effect pedals. While this worked well for creating an audible response to the performers' actions in the interface, it meant that the same tasks could not be performed on Aalberg Audio's effect pedals in a controlled manner.

However, using the participants' responses, it is possible to compare the prototype with traditional guitar pedals. The two interface modes (*sliders* and *circle*) can also be compared to each other.

5.1.1 Discoverability

The most clear-cut difference between traditional guitar effects and the prototype seems to be in discoverability. The circle interface was the reason for this difference. 5 of the 7 participants said the circle interface made it possible to discover sounds you otherwise wouldn't find, whereas no such statements were made about the sliders interface.

Explaining why this was the case is somewhat harder. The following statements from participants 4 and 5 indicate that the circle interface may reduce the possibility of analytical thinking (breaking the sound down into its individual parameters) by using a one-to-many mapping:

> You're not using the visual aspect [of the sliders interface]... where you can, for some reason, become stuck with a thought of "that's how it should be set, because that works". You're on thin ice, and that means you just move [the circle] around until it sounds good. (Participant 4)

> It was very fun to, kind of as "art by accident", end up on different variants—maybe sounds that I wouldn't find by dragging a slider up and down—because you can change multiple parameters with a single movement. (Participant 5)

On the other hand, statements from participants 3 and 6 indicate that the reason may simply be unfamiliarity with the circle interface:

> Because you're not used to the surroundings, you might get lost more quickly. And that's a good thing when it comes to finding new sounds. (Participant 3)

> I imagine that it might make you think in a different way, and in that sense it can be useful. [...] I'm a guy who will tune a string one tone down, because I then do completely different things creatively. (Participant 6)

Repeated usage of the circle interface would necessarily make it more familiar over time. If unfamiliarity was the main reason for increased discoverability, the effect might therefore be less pronounced in the long run. In the short term, however, discoverability was definitely increased for most participants.

5.1.2 Playability

There is some basis for saying that playability was increased compared to traditional guitar effects. Participants 2 and 4 expressed that it was good to be able to move multiple sliders at once; while participants 4, 5 and 6 expressed that the circle interface made it easy to adjust multiple parameters simultaneously. Being able to move multiple simultaneous parameters increases the fluidity of interaction, since the performer does not have to move focus from parameter to parameter. Because of this, playability can be said to be enhanced.

However, playability is hurt when performers are unable to perform the actions they want. Both the sliders and circle interfaces had some problems in this area. 2 participants expressed that it was hard to find-tune parameter values with the sliders. That is, small adjustments were hard. In this respect, the circle interface was generally better. No one claimed it was hard to adjust the circle's position. But one control parameter, the circle size, was described as hard to adjust by 4 participants; 3 of these specified that fine-tuning in particular was hard.

Why were the sliders hard to do small adjustments with, while the circle's position did not have this problem? The author believes the reason to be a difference in how the touch interactions were handled. For the sliders interface, the top of the slider was simply moved to the location of the detected touch event. Because of this, any attempt to "grab" the top of the slider would result in it being moved, unless you hit exactly the same point on the touch screen. Contrast this with the circle interface. There, all movement was made relative to the initial touch point, meaning the circle could be "grabbed" without being moved. The user would have to explicitly push or drag the circle in some direction to move it, making small adjustments much easier.

What about the problems with adjusting the circle size? All participants seemed to assume that pinch-in and pinch-out gestures had to be initiated from the level indicator or the touch area indicator (see figure 3.9 on page 33). This reduced the available range in which the fingers could move, especially for the pinch-in gesture. However, pinch gestures could in fact be initiated anywhere within the rectangular area in which the circle could be moved. One explanation for the participants' assumption could be that the primary gesture for moving the circle, namely dragging it around, had to be initiated in the circle's touch area indicator. Making it possible to drag the circle by touching any point on the rectangle could have made the gestures more consistent. Another explanation is that it simply made no sense for the users to touch anywhere other than the object being manipulated. Using the terms popularized by Norman (1988), the first explanation indicates an inconsistent *conceptual model*; the second explanation indicates that the circle did not have a perceived *affordance* indicating it could be resized in any other way.

Interestingly, the quick scale gesture for resizing the circle was used very little by the participants. In the author's opinion, it did not have any of the pinch gestures' problems. The lack of usage may simply be because the gesture is not yet well-established among users.

Another problem for the playability was that not enough attention had been paid to the response of each mapping. 4 of the 7 participants expressed in various ways that the responses of various mappings should not be linear. For example, it would have been wise if the mappings for the time parameter followed a quadratic or exponential curve. Participant 4 put it this way:

> When you're working with low [time] values, like with a slapback echo, it is currently very sensitive. When you start to get higher values, you don't really hear a difference between 670 and 675 milliseconds. [...] But when you're operating down there [with low values], it makes quite a big difference.

To summarize, it seems that playability was somewhat enhanced by the prototype; though problems with the handling of touch interactions and the mapping response also made playability worse. It is hard to say whether the positives outweighed the negatives. However, the problems outlined above should be possible to fix in a finished application.

5.1.3 Enjoyment

Based on figure 4.4 on page 64, it would seem that the prototype was somewhat more enjoyable to use than traditional guitar effects. 4 of the 7 participants said they would use the application themselves given the opportunity, while 1 would maybe use it. 3 participants also expressed that they enjoyed having the controls off of the ground.

During the short time that the participants used the prototype, the increased challenge inherent in the circle interface's mapping did not seem to increase the enjoyment of the circle interface compared to the sliders interface. Besides the comments regarding unfamiliarity and discoverability, no one mentioned enjoying the challenge or overcoming obstacles. Indeed, figure 4.2d on page 62 shows that more participants preferred the sliders interface than the circle interface overall, though the only a single participant separated them. This is interesting in light of the positive language generally used when describing the circle interface. It is, however, possible that preferences would change if the participants were allowed to use the prototype over a longer time period.

5.2 Answering the research questions

With the results presented in chapter 4, it should be possible to answer the research questions of the study.

5.2.1 Research question 1

RQ1: Compared to a digital musical instrument, does the use of a particular mapping strategy give rise to the same properties in an audio effect control interface?

In a digital musical instrument, one would expect a one-to-one interface to be easier to use. Figure 4.2a on page 62 shows that this also seems to be the case for the sliders interface, which uses a one-to-one mapping. One would also expect it to be more appropriate for analytical thinking, thus making it easier to find specific sounds. Figure 4.2b on page 62 shows that this, too, seems to true for the sliders interface. It is worth noting that the quantitative data shown in the box plots of figure 4.1 on page 59 do not indicate a significant difference between the two interfaces. However, those data did not take into account how quickly each task was completed, which might have changed the result.

The sliders interface uses something similar to what has been called a multiparametric mapping. In a digital musical instrument, one would expect this to be more engaging and enjoyable than a one-to-one mapping, leading to most people preferring the multiparametric mapping. This did not seem to be the case for the prototype, where no real difference in engagement or enjoyment could be seen between the two interfaces, and the sliders interface was preferred overall by a narrow margin (figure 4.2d on page 62).

One would also expect a multiparametric mapping to be better than a one-to-one mapping for the idea generation stage of creativity. As shown in figure 4.2c on page 62, this does seem to apply to the prototype as well.

Finally, one would expect an interface that combines a one-to-one mapping and a multiparametric mapping to be better than either on its own, because they support different phases of creativity. This is supported by figure 4.3 on page 63, which shows that having both the sliders and the circle interfaces would be preferable in the contexts of concert use, practice use, and composition use.

Based on this, the answer to the research question is not a clear “yes” or “no”, but rather a “mostly”. It does seem like a mapping strategy gives audio effect control interfaces many of the same properties one would expect in digital musical instrument. However, the property of engagement seems to be different. This study cannot answer concretely *why* engagement is different. One possibility is that, as suggested by Hunt and Kirk (2000), the advantages of multiparametric interfaces will only be revealed over time.

5.2.2 Research question 2

RQ2: Are evaluation methods for digital musical instruments also useful for evaluating audio effect control interfaces?

Based on the author's experiences with using the evaluation framework of O'Modhrain (2011), the answer to this question would have to be "yes". The evaluation sessions based on digital musical instrument evaluation strategies resulted in useful qualitative data, as presented earlier. The framework also provided a useful set of goals by which the prototype could be evaluated from a performer's perspective, as seen in section 5.1.

5.3 Flaws in the study

This study has some flaws that are worth noting.

First, all the participants of the study ended up being male. A more carefully planned participant selection could clearly have been used to ensure a balanced gender mix among participants. The limited number of participants also means that the results cannot be taken as more than an indication, and are not applicable to the population at large.

Second, Aalberg Audio's effect pedals could not be included in the experiment because of technical issues, described earlier. If communication with the effect pedals had been possible, it would have been much easier to evaluate whether the prototype achieved its goals compared to the existing Aalberg Audio system.

Finally, it took a long time before the prototypes were testable in a proper context with audible results. Part of the reason was that the author assumed communication with the effect pedals would be easy, thus no fall-back for how to test the prototypes had been planned. When the effects had to be mimicked in software, it consumed time that could have been used to test the prototype in a proper context before the final evaluation. This would probably have identified some of the problems with touch interaction and mapping response that had negative influence on playability.

5.4 Further work

There are several avenues of research that could be pursued after this study.

One possibility is to do a longitudinal study, allowing participants to use the prototype system repeatedly over a longer period. This could for instance be done by lending the prototype system out to users, by having participants come back for regular testing sessions, or a combination of both. This could be used to determine the effects of practice with the system. For instance one could find out if engagement with the circle interface would increase over time, and whether discoverability would be negatively influenced by more familiarity.

Another possibility would be to try to design a mapping strategy for controlling parameters across several effect pedals. For instance, with 3 effect pedals you have a total of 9 parameters to control. The author suspects the full usefulness of multiparametric mappings to only reveal itself when the number of parameters becomes too large to handle efficiently one by one. Using a strategy like the one presented by Tubb and Dixon (2014b) to control all parameters of all effects simultaneously would, in the author's opinion, be very interesting.

Several features were also suggested by participants during the interviews (see table A.20). Among these are the inclusion of presets. In addition to the direct usefulness of quickly being able to change from one sound to another, having preset management in the prototype would also open up for a range of functional mapping strategies. These are defined based on point-wise associations between an n-dimensional control space and an m-dimensional sound space. Presets could be used to define these associations, by for instance positioning presets on a 2-dimensional grid. Using one of the functional mapping strategies described by Van Nort et al. (2014), the performer could then "morph" between the presets. This would allow performers to specify their own, personal mappings.

It was also suggested that it would be useful to have the touch interface on a phone, which could be attached to the guitar. That would also open up the possibility for using the phone's accelerometer or other sensors as control inputs, for instance by performing small variations based on how the guitar is tilted.

5.5 Conclusion

The first aim of the study, as detailed in (section 1.1 on page 1), was to determine whether existing research into DMI design and evaluation was applicable for designing and evaluating interfaces for audio effect control. Having answered the research questions, we can conclude that this to a large degree is the case. The use of mapping strategies from the realm of digital musical instrument design mostly led to the expected properties in the final prototype; and the evaluation methods laid out for digital musical instruments proved useful for evaluating the final prototype from a performer's perspective.

Through answering the research questions, the second aim of the study—designing and developing a prototype tablet application that could control Aalberg Audio's guitar effects—was also met, with one exception: the prototype was not able to control the actual effect pedals, only software imitating those effects. This was due to implementation issues on Aalberg Audio's end. However, the imitations and the original effects should be sufficiently similar for the results of the prototype evaluation to be valuable.

For the 7 participants in the study, the prototype application achieved clear gains in discoverability compared to traditional guitar effects. However, gains in playability were to a large degree negated by problems with the touch interaction and the mapping response. Finally, enjoyment was slightly increased compared to traditional guitar effect control.

Bibliography

Beaudouin-Lafon, M. & Mackay, W. E. (2012, May 7). Prototyping tools and techniques. In *Human-computer interaction handbook* (3rd ed., pp. 1081–1104). Human Factors and Ergonomics. CRC Press.

Google Inc. (n.d.). Introduction - material design [Google design guidelines]. Retrieved May 25, 2015, from http://www.google.com/design/spec/material-design/introduction.html

Hunt, A. & Kirk, R. (2000). Mapping strategies for musical performance. In M. Wanderley & M. Battier (Eds.), *Trends in gestural control of music*. Paris, France: IRCAM—Centre Pompidou.

Hunt, A. & Wanderley, M. M. (2002, August). Mapping performer parameters to synthesis engines. *Organised Sound*, *7*(2), 97–108.

Hunt, A., Wanderley, M. M., & Paradis, M. (2003, December 1). The importance of parameter mapping in electronic instrument design. *Journal of New Music Research*, *32*(4), 429–440.

Keislar, D. (2014). About this issue. *Computer Music Journal*, *38*(3), 1–2.

McDermott, J., Gifford, T., Bouwer, A., & Wagy, M. (2013, January 1). Should music interaction be easy? In S. Holland, K. Wilkie, P. Mulholland, & A. Seago (Eds.), *Music and human-computer interaction* (pp. 29–47). Springer Series on Cultural Computing. Springer London.

Norman, D. A. (1988). *The psychology of everyday things*. New York, NY, USA: Basic Books.

Oates, B. J. (2006). *Researching information systems and computing*. London, United Kingdom: SAGE Publications.

O'Modhrain, S. (2011). A framework for the evaluation of digital musical instruments. *Computer Music Journal*, *35*(1), 28–42.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, *24*(3), 45–77.

Rovan, J. B., Wanderley, M. M., Dubnov, S., & Depalle, P. (1997). Instrumental gestural mapping strategies as expressivity determinants in computer music performance. In *Proceedings of kansei - the technology of emotion workshop.*

Tanaka, A. (2010). Mapping out instruments, affordances, and mobiles. In *Proceedings of the international conference on new interfaces for musical expression (NIME)* (pp. 88–93). New interfaces for musical expression. Sydney, Australia: NIME.

Tohidi, M., Buxton, W., Baecker, R., & Sellen, A. (2006). Getting the right design and the design right. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 1243–1252). CHI '06. New York, NY, USA: ACM.

Tubb, R. & Dixon, S. (2014a). A zoomable mapping of a musical parameter space using hilbert curves. *Computer Music Journal*, *38*(3), 23–33.

Tubb, R. & Dixon, S. (2014b). The divergent interface: supporting creative exploration of parameter spaces. In B. Caramiaux, K. Tahiroglu, R. Fiebrink, & A. Tanaka (Eds.), *Proceedings of the internation conference on new interfaces for musical expression* (pp. 227–232). New interfaces for musical expression. London, United Kingdom: Goldsmiths, University of London.

Van Nort, D. (2010). *Modular and adaptive control of sound processing* (Doctoral dissertation, McGill University, Montreal, Canada).

Van Nort, D., Wanderley, M. M., & Depalle, P. (2014). Mapping control structures for sound synthesis: functional and topological perspectives. *Computer Music Journal*, *38*(3), 6–22.

Wanderley, M. & Depalle, P. (2004). Gestural control of sound synthesis. *Proceedings of the IEEE*, *92*(4), 632–644.

Wessel, D. & Wright, M. (2002). Problems and prospects for intimate musical control of computers. *Computer Music Journal*, *26*(3), 11–22.

Appendices

Appendix A

Categories and codes

This appendix lists all codes used when coding the interviews, grouped in code categories. Additionally, the occurrence of each code among the interviewed participants is indicated.

Table A.1: Musical background.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Plays other instruments than guitar	✓			✓		✓	✓	4
Plays or played guitar in well-known band			✓	✓	✓			3
Plays or played in a band	✓	✓				✓		3
Professional musician			✓	✓	✓			3
Self-taught		✓				✓	✓	3
Only plays in a studio context	✓	✓						2
Composed for film and theatre			✓					1
Done a lot of freelance work				✓				1
Plays most guitar-like instruments				✓				1
Plays mostly keyboard	✓							1

Table A.2: Effect knowledge.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Does music production work	✓	✓		✓	✓			4
Does not know how to implement effects			✓		✓	✓		3
Highly knowledgeable	✓		✓	✓				3
Uses mostly software effects	✓	✓			✓			3
Does not usually adjust effects while playing				✓		✓		2
Fairly knowledgeable					✓	✓		2
Has multiple effect setups for different kinds of work			✓	✓				2
Knows how to implement effects	✓	✓						2
Long experience with using guitar effects			✓		✓			2
Studies music technology	✓	✓						2
Feels digital effects are harder to control						✓		1
Knowledgeable about analogue effects only						✓		1
Somewhat knowledgeable							✓	1
Uses a MIDI-controlled mixing matrix for effects			✓					1
Uses software effects, but prefers hardware effects live			✓					1

Table A.3: Attitude toward new guitar and effects technology.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Has a fairly stable effects setup		✓		✓		✓	✓	4
Keeps up with the latest developments	✓	✓	✓	✓				4
Open to experimentation with the setup			✓	✓		✓	✓	4
Positively inclined towards new technology		✓	✓	✓	✓			4
Does not keep up to date					✓		✓	2
Doesn't like digital effects						✓		1
Has to be simple						✓		1
Is somewhat up to date on effects						✓		1
Is used to having a tablet on stage					✓			1
Likes guitar effects						✓		1
New technology has to actually innovate				✓				1
Old-fashioned						✓		1

Table A.4: Slider interface - positive.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Sliders are easy to understand / intuitive	✓	✓		✓	✓	✓	✓	6
Similar to a pedal / familiar	✓		✓	✓		✓	✓	5
Sliders are more visually clear	✓	✓		✓		✓	✓	5
Easy to dial in parameters	✓	✓		✓	✓			4
Sliders are easier than circle	✓				✓	✓		3
Good to be able to move multiple sliders at once		✓		✓				2
Sliders are easier to get started with	✓	✓						2
Sliders worked well				✓		✓		2

Table A.5: Slider interface - negative.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Hard to fine-tune with sliders			✓				✓	2
Slider control could as easily be done on the pedal					✓			1
Sliders are boring	✓							1
Sliders were "jumpy"			✓					1

Table A.6: Circle interface - positive.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
You can discover sounds you otherwise wouldn't find	✓		✓	✓	✓	✓		5
Circle makes it easy to adjust multiple parameters simultaneously				✓	✓		✓	3
Circle is much more expressive		✓		✓				2
Circle was intuitive / easy to understand		✓		✓				2
Fun	✓				✓			2
Unfamiliarity may enhance creativity			✓			✓		2
A creative tool					✓			1
Circle easier to use live	✓							1
Circle makes you discover pedals in a new way					✓			1
Circle would work really well in the studio				✓				1
Good to have two ways to resize circle		✓						1
Innovative				✓				1
It was nice that you could instantly move the circle			✓					1
Liberating		✓						1
Promising				✓				1
Useful in the early creative phase					✓			1
Very cool				✓				1

Table A.7: Circle interface - neutral.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Circle interface makes you listen more	✓			✓				2
Circle is very different compared to usual guitar pedal control				✓			✓	2
Double-tap-and-drag a bit easier than pinch							✓	1
Experimental	✓							1

Table A.8: Circle interface - negative.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Hard to adjust circle size	✓				✓	✓	✓	4
Hard to fine-tune using circle size	✓				✓		✓	3
Hard to know what you are controlling		✓	✓		✓			3
Has a learning curve		✓			✓	✓		3
Circle was hard to hit when in a corner		✓						1
Circle was less intuitive than slider							✓	1
Didn't like having delay time mapped to circle size			✓					1
Double-tap-and-drag reacted opposite to expectation							✓	1
Felt circle might move while changing its size		✓						1
Felt you had to start pinch gesture at the current value						✓		1
Hard to use circle size to adjust small values				✓				1
Tap-and-hold to move is a bit too slow					✓			1
The effects used didn't benefit from being able to change them simultaneously			✓					1
The three circle parameters aren't equal	✓							1
Was not familiar with double-tap-and-drag gesture							✓	1

Table A.9: Both interfaces - positive.

Code	1	2	3	4	5	6	7	Freq
	Participant number							
Nice to have the controls off the ground			✓			✓	✓	3
Having access to both is what makes it appealing				✓	✓			2
Better than controlling effects with a mouse		✓						1
Good looking		✓						1
It's the best way to find and explore sounds and effects		✓						1
It works really well in some situations				✓				1
Would combine with regular effects to get benefits from both							✓	1

Table A.10: Both interfaces - neutral.

Code	1	2	3	4	5	6	7	Freq
	Participant number							
Feels equal to guitar pedals			✓					1
May be more suited to studio or home (not concert/practice)						✓		1

Table A.11: Both interfaces - negative.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Some controls should be non-linear			✓	✓		✓	✓	4
Lacks the physical element						✓	✓	2
Doesn't want anything digital in the effect setup						✓		1
Generally hard to fine-tune			✓					1
Hard to set very small values				✓				1
Lacks the easy overview you get with physical pedals							✓	1
Learning to use and integrate the app in an existing workflow could take some time				✓				1
More complicated to switch focus between effects							✓	1
No advantages over Aero controller			✓					1
Not really easier to use than pedals	✓							1
Unused potential of tablet and touch			✓					1

Table A.12: Ease of use.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Slider is easiest to use			✓		✓	✓	✓	4
Both are easy to use	✓			✓				2

Table A.13: Finding specific sounds.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Sliders are better for finding specific sounds	✓		✓	✓	✓	✓		5
Equally good for finding specific sounds							✓	1

Table A.14: Finding new sounds.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Circle is better for finding new sounds	✓		✓	✓	✓			4
Circle might be better for finding new sounds		✓				✓		2
Equally good for finding new sounds		✓					✓	2

Table A.15: Overall preference.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Preferred the slider interface			✓			✓	✓	3
Liked both		✓		✓				2
Preferred the circle interface	✓				✓			2

Table A.16: Concert suitability.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Both interfaces have their uses during concert		✓		✓	✓			3
Circle more suitable for concert	✓						✓	2
Sliders more suitable for concert			✓			✓		2

Table A.17: Practice suitability.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Both are good for practice		✓	✓	✓	✓			4
Circle more suitable for practice	✓						✓	2
Sliders more suitable for practice						✓		1

Table A.18: Composition suitability.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Both are useful for composition	✓	✓		✓	✓			4
Slider is more useful for composition			✓				✓	2
Circle might be better for composition						✓		1
You have more time, so changing multiple parameters simultaneously is less useful							✓	1

Table A.19: Would use app?

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Would use app	✓	✓		✓	✓			4
Would not use app			✓			✓		2
Would maybe use app							✓	1

Table A.20: Suggestions.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Named presets			✓		✓			2
Phone-size interface attached to guitar	✓	✓						2
Should be possible to change all three circle parameters simultaneously				✓	✓			2
Values and labels should be placed so that they easy to see when using the interface		✓	✓					2
Circle - Double tap somewhere to move there instantly			✓					1
Full screen mode for circle interface	✓							1
Have the least important parameter on circle size	✓							1
Important to be able to easily change between effects in app without pressing small buttons							✓	1
List of songs with presets for different parts of songs			✓					1
Tablet may be better for keyboardists	✓							1
Which parameter you controlled could be determined by how many fingers you touch the screen with			✓					1

Table A.21: General statements.

	Participant number							
Code	1	2	3	4	5	6	7	Freq
Sound quality is important				✓		✓		2
Changing parameters on pedals does not often happen	✓							1
Practice as if it were a concert							✓	1
User-friendliness is important				✓				1
You need to learn your tools to use them well				✓				1