NTNU
Norwegian University of
Science and Technology

SeaRhythm - A Rhythm Game with the ROLI Seaboard

Innovation in Game Controllers

Tore Bjørlo

Master of Science in Computer Science
Submission date: February 2018
Supervisor: Alf Inge Wang, IDI

Norwegian University of Science and Technology
Department of Computer Science

Abstract

More than 60 years ago, the first video games started appearing, and game controllers followed. Since then, the video and computer games market has become one of the largest entertainment industries in the world. Millions of dollars are spent on research and development of new game controllers, and the most popular ones have followed a similar design over the past two decades. This report studies non-conventional gaming controllers and how they might be used in creative ways to create new gaming experiences.

The first part of this study explores different game controllers, their evolutions, and their parallels. It aims to explore innovative ways for a non-controller device to be used as a game controller. Frameworks for game enjoyment are studied, and form the basis for several game concepts. One of these concepts is made into a rhythm-game which uses a non-controller as the primary gaming device. This game was tested using the frameworks for game enjoyment to ascertain the quality of the game.

Results from the tests indicate that the use of a non-conventional game controller can affect the gaming experience in a positive fashion. However, several flaws have been discovered in the design, but these can be fixed to further improve the positive effect.

Sammendrag

For mer enn 60 år siden begynte de første dataspillene å dukke opp, og spillkontrollere fulgte. Siden den gang har markedet for dataspill blitt et av de største underholdningsindustriene i verder. Millioner av dollar blir brukt på forskning of utvikling av nye spillkontrollere, og de som er mest populære har fulgt et lignende design de siste to tiårene. Denne rapporten studerer ikke-konvensjonelle spillkontrollere og hvordan de kan bli brukt på kreative måter for å lage nye spillopplevelser.

Den første delen av denne rapporten utforsker forskjellige spillkontrollere, hvordan de har utviklet seg, og deres paralleller. Den utforsker innovative måter for en ikke-kontrollenhet til å bli brukt some en spillkontroller. Rammeverk for gode spillopplevelser blir studert, og danner grunnlaget for flere spillkonsepter. Ett av disse konseptene blir laget til et rytmespill som bruker en ikke-kontrollenhet som en primær spillkontroller. Dette spillet ble testet ved bruk av rammeverkene for gode spillopplevelser for å fastslå kvaliteten til spillet.

Resultatene fra testene indikerer at bruken av en ikke-konvensjonell spillkontroller kan påvirke spillopplevelsen på en positiv måte. Derimot, så har flere feil blitt oppdaget i designet, men disse kan bli fikset for å ytterligere forbedre den positive effekten.

Acknowledgements

I would like to express my profound gratitude to Professor Alf Inge Wang, my supervisor, for his patience, valuable suggestions, and constructive advice. Through our discussions, he has helped me by providing both perspective and clarity throughout this project.

I would also like to give special thanks to Arne Bjørlo for his helpful feedback and assistance in proofreading this report, and also to the rest of my family for their encouragement and support.

Lastly, I give sincere thanks to everyone who participated in the testing process and provided feedback. You were an invaluable part of the culmination of this project.

Contents

List of Figures

List of Tables

Part I
Introduction

This chapter serves as an introduction to this project, beginning with a brief explanation of the motivation behind the project and my personal motivation for undertaking it. Next, the project and its primary phases are described, followed by the research questions this project is looking to answer. Finally, there is an outline describing the main themes of the remainder of the report.

Chapter 1

Motivation

Video games and computer games are constantly evolving. A key factor for this evolution is through the production of newer and better hardware, and consequently, software. New avenues for games are being explored, with virtual reality being one of the current frontiers. Although virtual reality is not a new concept, dating back to the 1950s or even earlier, it is only in recent years it has become more widely available to the public.

However, it is not only through brand new technology that we discover new gaming experiences, but also through innovation and unconventional approaches. This project takes an interesting approach to game development by taking a device that was not designed to be a game controller and developing a game concept using this particular device as a controller.

My own great interest for video games is my motivation for undertaking this challenge. My goals are to learn more about computer game design and development, to explore new ways of creative thinking, and to develop my computer game programming skills.

Chapter 2

Project Description

This project is the continuation of a previous specialization project, conducted in the spring of 2017 (Bjørlo, 2017). In the specialization project I did a prestudy into various game controllers and their evolution, and into what aspects of a game affect game enjoyment. The product of the specialization report was a concept for a game. In this report the text in Part II is included from the specialization project. Sections 7.2 and 7.6 have received minor changes, while extra information has been added to Chapter 8. Chapter 12 in Part III is also from the previous project, while Chapters 10, 11, and 13 - 15 are based on it.

Since this master's thesis is a continuation of the specialization project, the problem description is the same for both:

> **Music Game with the ROLI Seaboard**
> In this project, the goal is to create an innovative music game prototype which uses the ROLI Seaboard RISE 25 as input to control the game. The Seaboard is a new type of musical keyboard which makes it possible to change the expression of individual tones in a unique way, through the use of a touch-sensitive surface.
>
> The first part of the project will be to study and research technology and literature related to this project. The second part will be to develop a prototype of a musical game that uses the Seaboard as a controller, and the third part will be to evaluate the prototype with users.

The first phase of the project was completed in the specialization project, while the two remaining phases have been carried out as part of this thesis. The second phase began by creating my own game based on the concept and early design from the previous project. Some unforeseen technical problems occurred during the implementation which delayed the project and forced me to abandon my original plan of using C#. Instead, I had to use C++ which, although similar to C#, I had very little experience with and resulted in a decision to scale down the scope of the prototype. In the end, after roughly 3 500 lines of code, *SeaRhythm* was created.

After the development was over, it was time for the final phase: evaluation. This phase began by deciding on an approach for measuring the game enjoyment of the players of *SeaRhythm*. A user testing period was initiated and a survey was created asking the players various questions about what they thought of the game and the Seaboard as a controller. The testing period lasted for one week, during which I had players play *SeaRhythm* and collected answers for the aforementioned survey. After testing was done, I spent the remainder of my time writing this thesis.

Chapter 3

Research Questions

This project uses a research approach based on the Goal, Questions, Metrics paradigm by Basili (1992). The GQM approach is defined in a top-down fashion. Goals are specified first, then quantifiable questions are defined, and finally a set of metrics are created, in order to help answer the questions [1].

Research goal - Develop a game with the Seaboard as the controller and investigate how the Seaboard affects game enjoyment.

Based on the research goal, I have decided on the three following research questions:

RQ1 - *In which way did the use of the Seaboard as a controller influence the player's opinion of the game?*
This research question studies the players' experience of using the Seaboard as a game controller. It is important that the players find the game to be enjoyable, and to which degree the Seaboard affects the experience.

RQ2 - *To which degree does* SeaRhythm *fulfill the criteria of game enjoyment?*
This research question seeks to answer whether *SeaRhythm* is an enjoyable game based on the theory behind game enjoyment. It is important to determine whether the potential enjoyment *SeaRhythm* offers has a solid basis in theory established through measurable and tried metrics.

RQ3 - *Does the players' experiences of* SeaRhythm *support the design choices that were made to increase game enjoyment?*
While similar to **RQ2**, this research question takes into consideration the players' opinions and seeks to find a correlation between theory and practice. It is important to determine whether the theory of game enjoyment holds up in practice and what improvements can be made.

Chapter 4

Report Outline

This report is divided into four parts. Each of the last three parts covers a milestone of the project: research, contribution, and conclusion.

Part I introduces the motivation for the project, the project itself, the research questions this report aims to answer, and the research approach.

Part II contains a prestudy on the subject of game controllers. This includes a general history of game controllers as well as a review of several notable modern controllers. The Seaboard is presented, before several frameworks for game enjoyment are introduced.

Part III begins with an evaluation of the Seaboard as a game controller. Next, a few games that were used as inspiration are presented, together with some thoughts on which of their game mechanics can fit the Seaboard as a controller. Several game concepts are described, with one concept being chosen for further development. Next follows the game design, technology, requirements, software architecture, and testing.

Part IV details the user test strategy before going into the results of said user test.

Part V discusses the results of the project, and provides evaluations of the design, results, and the project as a whole.

Part VI presents the conclusion to the project, and ends with a description of future work to be done.

Part II
Prestudy

This part contains a prestudy, which is performed in order to gain an overview of game controller history and design, as well as how to design an enjoyable game. An understanding of these attributes will lay the foundation for what is required to come up with a concept for a game suited for the Seaboard.

First, I will explain what a game controller is, and then describe how they have evolved since they were first introduced until present day. After, I will present several modern game controllers and their designs, before investigating the Seaboard and its properties. Finally, I introduce the concept of game enjoyment and some frameworks for how it can be evaluated.

Chapter 5

What is a Game Controller?

A game controller is a device intended to provided input to and interact with a computer or video game, usually by controlling a character or an object in the game. Normally, game controllers are connected externally by a cord or by wireless technology to the games or entertainment system they are being used with. However, in the case of hand-held entertainment devices, such as the Nintendo 3DS or Sony PSP, the controllers are integrated into the device itself. Some of the more common input devices to be used as game controllers include keyboards, mouse devices, gamepads, and joysticks [2].

Computer and video games are often separated; computer games are played on personal computers (PCs), while video games are usually played using a TV and a video game console. Video game consoles are built around a standardized set of hardware and software, which is relatively inexpensive and easy to mass produce [3]. Comparatively, there are many PCs with different hardware on the market. They can also be customized extensively, allowing customers to choose parts themselves, instead of buying a pre-built PC.

Traditionally, computer games are played using a mouse and keyboard, while consoles use a controller designed for that specific platform [3]. One thing to note is that even though most video game controllers used to be incompatible with PCs because of the hardware and software restrictions on consoles, this is no longer the case. Today's consoles are becoming more like PCs, and many modern controllers work both on the system they were designed for and on PCs [4].

Chapter 6

Evolution of Game Controllers

Since they first started appearing in the early 1950's and 1960's, the video and computer games market has grown to become one of the largest entertainment industries in the world. Figures 6.1 and 6.2 shows the economic growth of the video game industry in the United States over the past few years. In such a large industry, a lot of money is invested not into just games and consoles, but also controllers. For instance, Microsoft spent over $100 million on research and development for their Xbox One controller [5].

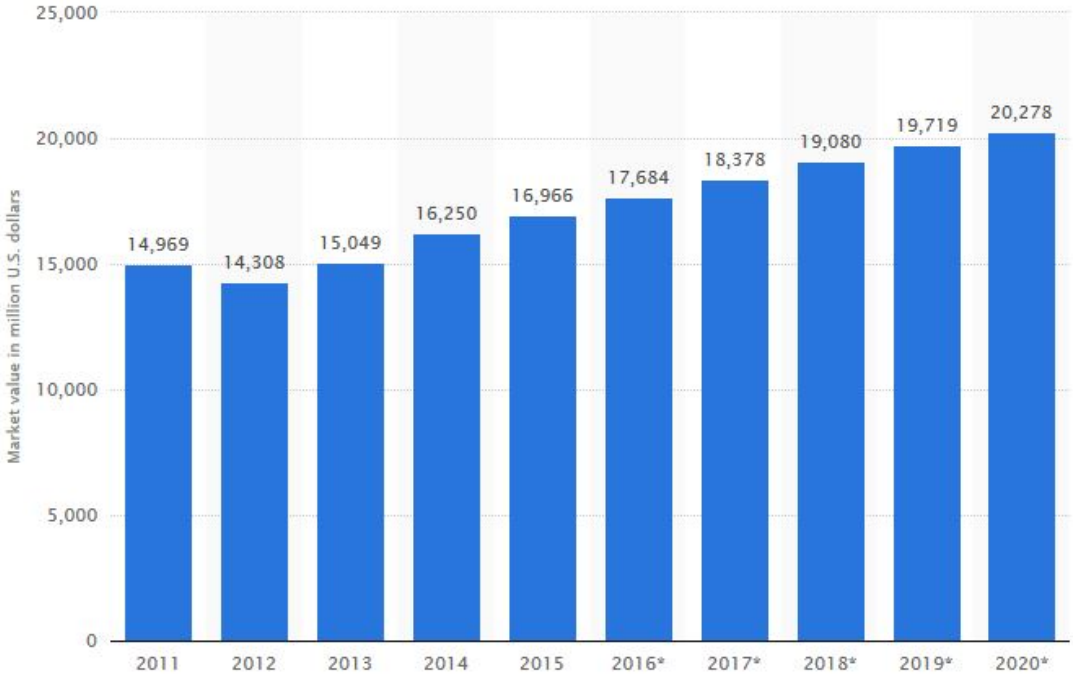


FIGURE 6.1: Value of the video game market in the United States [6].

Overmars (2012) states that the games of the present are rooted in the games of the past, and I believe the same is true for controllers. Although the first computer

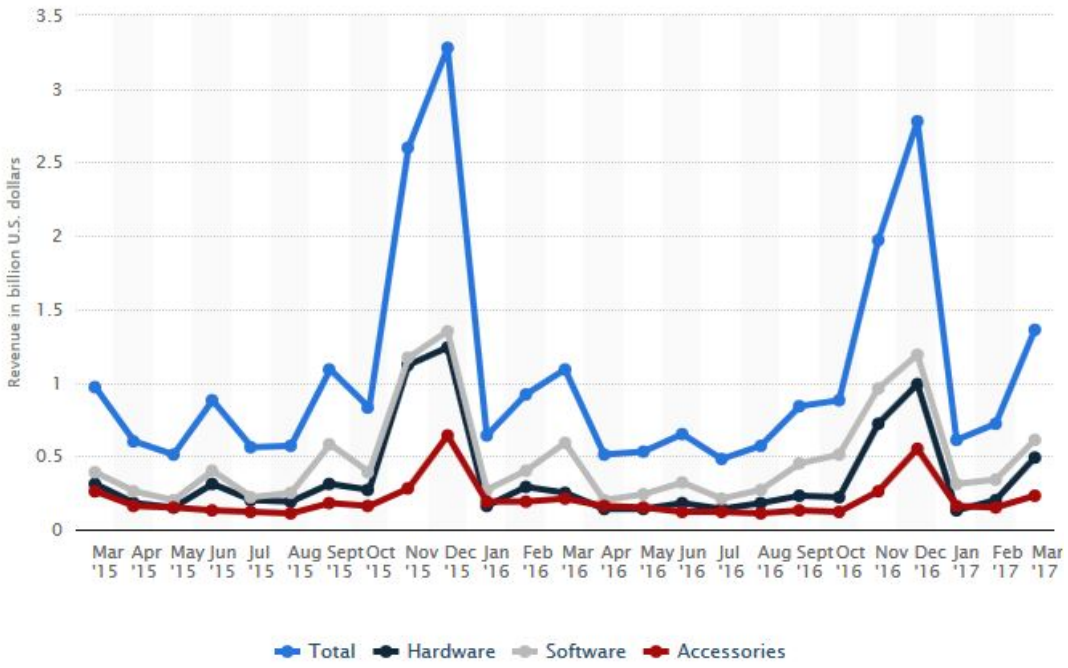


FIGURE 6.2: Total and segment revenue of the U.S. video game industry [7].

games were cutting-edge technology at the time, by today's standards they are very simple and so are their controllers. Early controllers had a few buttons and a dial, allowing limited interactivity. Nowadays controllers have many more buttons, multiple analog sticks, and some even use motion detection technology [8]. The two next sections will provide an overview of how some of these changes came to pass.

6.1 Early Controllers

In the infancy of computer games there was no hardware that had been designed specifically to interact with them. Digital switches or knobs connected to various resistors were used for input. One of the first game controllers to be developed was for *Spacewar!* (1961) on the DEC PDP-1. The toggle switches that were part of the PDP-1 proved to be clumsy, so the developers made a box with two two-way switches and a button, as seen in Figure 6.3 in order to make the game easier to play [3].

According to Cummings (2007), *Tennis for Two* (1958) was the first computer game. It was played using a paddle, a simple device with a knob and a button, connected to the computer through a wire. The knob could be rotated to change the angle of the

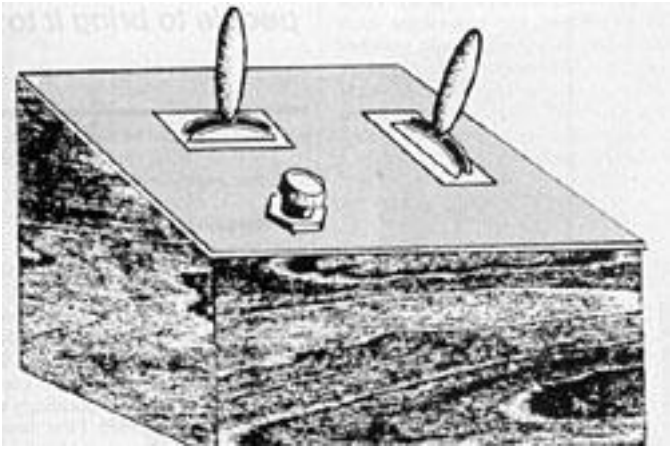


FIGURE 6.3: *Spacewar!* controller sketch

return of the ball, and pressing the button would return the ball across the net. The paddle would go on to be used as the controller for the Magnavox Odyssey (1972), the first video game console, and later for the popular Atari game *Pong* (1975) [3, 9].

By the late 1970s and early 1980s, many consoles had been developed and so had many controllers. The popular Atari consoles from this period all used a joystick as their main controller, an example of which is shown in Figure 6.4. The joystick is a more advanced version of the controller for *Spacewar!*, combining the two sticks into one multi-directional stick. The joystick became quite prevalent, meaning most games had to adapt their gameplay to work with it. This was fine for the simple 2D games of the Atari, but with the introduction of the gamepad, use of the joystick dwindled [3].

FIGURE 6.4: Atari 2600 joystick

The gamepad became popular with the release of the Nintendo Entertainment System (1983), although it was not the first gamepad in existence. The NES controller consisted of a directional pad (D-pad) and two buttons, as seen in Figure 6.5. One of the advantages of the D-pad is that it requires less movement than the joystick and was less prone to being damaged. It supports movement in eight directions (up, down, left, right, and diagonal), which made it perfect for the 2D platform games the NES is known for, such as *Super Mario Bros.* [3]. The gamepad went through many different designs across many different consoles, but some of the most notable development occurred after the transition from two-dimensional to three-dimensional games [3]. This transition marked the beginning of what we today consider modern game controllers.

FIGURE 6.5: NES controller

6.2 Modern Controllers

When 3D consoles were first introduced, the controller design was still based on that of controllers which were used for 2D games. The issues were that D-pads do not work well in a 3D environment, because player movement would be limited, and there also needed to be some form of camera control. With the introduction of the Nintendo 64's (N64, 1996) controller and the analog stick, the first problem was solved. Players could now move in any direction, and also at different speeds since analog sticks are pressure sensitive and allow for degrees of movement. Having the camera automatically follow the player in a 3D environment is hard, and obstacles can frequently block the player's view. The N64 controller allowed for some degree of camera control through the use of four buttons which would adjust the camera around the player, alleviating the worst of the problem. In 1997, Sony released the Dual Analog controller for PlayStation (1994). It featured two analog sticks, one of which could be used to control the camera more easily while also allowing proper 3D movement [3, 10]. These two controllers are described further in Sections 7.1 and 7.2.

Following the analog stick was the development of analog buttons. Same as the analog stick, these buttons could provide a degree of input that was often used in driving games to provide variable acceleration.

Although several types of peripheral and novelty controllers had been around for many years, such as light guns and special controllers for arcade games, rhythm games experienced a surge of popularity with the release of games such as *Dance Dance Revolution* (DDR, 1998) and later *Guitar Hero* (2005). These type of games simplified a more complex musical task to a level that a computer can understand. In the case of DDR, a dance mat was developed that looks a lot like a D-pad. The player would stand on the mat and hit the correct arrows at the correct time using their feet [3].

Touch screen technology was introduced to the public gaming market through the Nintendo DS (2004). It is a handheld game console that uses a touch screen as one of its inputs, a feature Nintendo also incorporated into the 3DS (2011), the Wii U (2012), and the Switch (2017) [11, 12, 13]. Furthermore, games with touch screen controllers have become quite common in the mobile game market since the appearance of smartphones, with over 600 000 available games on the AppStore [14].

The Nintendo Wii (2006) was a great success and the Wiimote was quite a development within controller design with its addition of motion control. It can detect movement and also where the player is pointing the controller, allowing for new ways of playing games [3, 8]. Further details on the Wiimote can be found in Section 7.4.

One of the latest uses of motion control in games is within the realm of virtual reality. Using a head-mounted display, players can fully immerse themselves into a virtual world and interact with it using handheld motion controllers. In the virtual world, players can look around by moving their heads and they can reach out with their hands and manipulate objects using the controllers. Pictured in Figure 6.6 are the handheld controllers for the HTC Vive (2016), one of the VR devices currently on the market.

6.3 Summary of Evolution of Game Controllers

During the past 60 years, we have gone from simple controllers made from knobs and buttons to touch screens and motion detection. The first controllers were made from whatever was at hand, with games equally as simple, and now every last detail is designed from the ground up as part of a multi-billion dollar industry. Although still limited by computing power, games became more complex with the creation of the gamepad, sparking the creation of many popular 2D platform games. When 3D was introduced, the analog stick was designed, providing players with the tools

FIGURE 6.6: HTC Vive controllers

to direct their characters in this new environment. The analog stick has become a standard in nearly every modern controller, providing both character and camera control. Motion controls made their mark with the innovative Wiimote, creating an influx of games that were controlled in a new way. For the past few years, the evolution of motion control has continued through the development of VR, making games more immersive than ever.

Through this study, I have gained some insight into the drive behind the evolution of game controllers. Not only do the games affect the design of controllers, but the controllers also affect the design of games. In my case, I need to consider the Seaboard's constraints carefully in order to develop a game concept which fits the controller.

Chapter 7

Game Controllers

In this chapter I will further discuss controllers that were defining or innovative. I look for trends and parallels between different controllers, which might help me understand how to use the Seaboard as a controller.

7.1 Nintendo 64 Controller

The design team behind the controller for the Nintendo 64 were told to try out new ideas that would break from the standard mold of gaming controllers [15]. The result was the M-shaped controller in Figure 7.1.

FIGURE 7.1: Nintendo 64 controller

The controller is designed to be held in three different ways. First, it can be held by the two outer grips. This allows access to the right-hand side side buttons, the D-pad, and the "L" and "R" shoulder buttons, making it the ideal way to play 2D games. Second, it can be held by the center and right grips, which provides access to the analog stick, the right-hand side buttons, the "R" shoulder button, and the "Z" trigger on the controller's underside. This grip was intended for 3D games. Last, the controller can be held by the center and left grips. This allows the use of the D-pad, the analog stick, the "L" shoulder button, and the "Z" trigger, which was used for the

popular game *GoldenEye 007* [16]. The design is rather controversial, since not all the buttons are available at the same time by the nature of its shape.

Also included in the N64 controller is a cartridge slot on the controller's underside. This slot could be used for the Rumble Pak or the Controller Pak. The Rumble Pak was designed to provide haptic feedback while playing, in an effort enhance game immersion. The Controller Pak is the N64's external memory card, providing storage beyond the game cartridges' battery-backed memory. Whereas other consoles used a memory card that plugged into the console itself, Nintendo decided to plug it into the controller instead. This design choice allowed players to bring their own controllers to friends while keeping their own settings and controller configurations.

Despite its unconventional design, the N64 controller was slightly ahead of its time, being among the first controllers to feature an analog stick. Although the concept of the Controller Pak never became popular beyond the N64, the analog stick was quickly adopted into other gamepads and is still a standard to this day.

7.2 PlayStation and Xbox Controllers

PlayStation 4 (2013) and Xbox One (2013) are the two largest brands of current generation consoles on the market [17]. Both consoles have gone through several earlier iterations (PS, PS2, PS3; Xbox, Xbox 360), and so have their controllers. What designs have two giants in the video console market used for their controllers and how have they changed from one generation to the next?

FIGURE 7.2: Controllers for each generation of PlayStation. Top left: PlayStation Controller. Top right: DualShock 2. Bottom left: DualShock 3. Bottom right: DualShock 4

Figure 7.2 illustrates the PlayStation controllers over the past four console generations. The core design of the original PlayStation controller has been kept largely intact, except for the addition of analog sticks to all later models. Not pictured is the original DualShock controller which was released for the original PlayStation in 1997, three years after the console's release. Cosmetically, it looks almost identical to the DualShock 2. In fact, the only difference worth mentioning between the two is the fact that the DualShock 2's buttons are analog. DualShock 3 became wireless, but it wasn't until DualShock 4 that there were some larger cosmetic changes. The main difference, besides cosmetics, was the inclusion of a touchpad on the front of the controller.

FIGURE 7.3: Controllers for each generation of Xbox.
Top left: Xbox Controller. Top right: Xbox Controller S.
Bottom left: Xbox 360 controller. Bottom right: Xbox One controller

The different Xbox controllers are shown in Figure 7.3. The top two controllers were both made for the original Xbox (2001). The Xbox Controller S replaced the larger Xbox Controller as the standard controller by popular demand in 2002 [18]. The Xbox 360 controller featured a slimmer design, but besides the movement of a few auxiliary buttons, it was quite similar to the Xbox Controller S. Even though Microsoft invested $100 million into research and development for the Xbox One controller, creating more than 200 prototypes [19], in the end they settled on releasing what is essentially an improved version of the Xbox 360 controller, featuring improved textures, grips, and battery life.

Both PlayStation and Xbox controllers have kept to a standard design they seem unwilling to depart from. Although they spend millions on research, there have been few major changes. Maybe that is because there is little to improve upon.

7.3 Steam Controller

The Steam Controller was released by Valve in November 2015 for PC. It is rather unique due to the fact that it is designed to be used as both a traditional game controller, and as a substitute for a mouse and keyboard. The controller is mainly meant to be used within Steam's Big Picture mode, which is designed for use with TVs and game controllers, where controls can be mapped on a per-game basis. Outside of Big Picture mode, it operates as a normal controller [20].

Instead of the traditional double analog sticks, the Steam Controller features two clickable trackpads and only one analog stick. It also has a total of sixteen buttons spread across the face, shoulders, and underside of the controller. Motion sensing within the controller is used to track its relative orientation.

FIGURE 7.4: Steam Controller

By June 2016, 500 000 units had been sold, reaching nearly 1 million units sold by October of the same year [21]. Comparatively, PlayStation 4 and Xbox One sold 10.2 million and 5.5 million units[1] during their first seven months on the market, respectively [22].

Commercially, the Steam Controller has not been a success, but it is an attempt to bridge the gap between video and computer games. For the time being, however, it seems that the Steam Controller has not broken any significant barriers.

[1]The sales figures for PlayStation and Xbox are for controllers that come bundled with console purchases, and do not include sales of separate controllers. The Steam Controller sales include controllers that came bundled with the Steam Machine.

7.4 Wii Remote

The Wii Remote, more commonly known as the Wiimote, was another innovative controller from Nintendo. It is the main controller for the Nintendo Wii (2006) and one of its main features is motion control. The Wiimote can be used separately or together with a Nunchuk, both of which are shown in Figure 7.5. Besides motion controls, it features a D-pad, five buttons on top, and one trigger on the bottom. Additionally, it supports basic audio and rumble functionality, and if held sideways it works much like an NES controller. When connected, the Nunchuk adds an analog stick and two more buttons, providing functionality similar to a gamepad [23].

FIGURE 7.5: Wiimote (right) with attachable Nunchuk (left)

The Wiimote stands in stark contrast to other more typical controllers. Its one-handed remote control-looking design suits motion control since we are used to pointing with remote controls. To use the pointer functionality realiably, the player must be relatively close to the infrared sensor bar connected to the console in order to pinpoint where the player is pointing the controller. For motion tracking, the Wiimote uses an accelerometer with which it is able to sense acceleration along three axes. The tracking is communicated wirelessly to the Wii through Bluetooth, allowing players to mimic actual game actions, such as swinging a sword.

The Wiimote might have put a constraint on video game developers, but it did facilitate the creation of many new types of games. Even though Nintendo has incorporated motion controls into later controllers, it is no longer the primary feature. However, Sony's motion controller, the PlayStation Move, has spiked in sales following the release of the PlayStation VR, proving that virtual reality is a viable market

for motion controllers [24].

7.5 Wii U GamePad

The GamePad for the Wii U (2012) is particularly interesting because it combines standard controller elements, motion control, and touchscreen controls. The touchscreen can be used as a supplement to games by providing an additional screen, allowing players an alternate view or extra functionality. Additionally, it can be used much like a handheld device since games can be played directly on the GamePad's screen, without the use of a TV. Nevertheless, it must remain in close proximity of the Wii U to work.

The GamePad, displayed in Figure 7.6, has incorporated some of the more traditional controller elements into its design. It features two analog sticks, a D-pad, four action buttons, and four shoulder buttons. Also, it has motion detection, a camera, rumble support, and a touchscreen [12].

FIGURE 7.6: Wii U Gamepad

As seems to be the norm for Nintendo, some of the GamePad's functions were iterated upon, and incorporated for the Nintendo Switch. However, the GamePad has had seemingly no impact outside of Nintendo.

7.6 Joy-Con

Joy-Con are the primary controllers of the Nintendo Switch (2017). It consists of two individual units that can be used together by one player, or separately by two players. One can be held in each hand, or they can be connected to the Joy-Con Grip to create a more traditional-looking controller. They can also be attached to the sides of the Switch console unit, making it look similar to the Wii U GamePad except

that it can be used as a fully portable device since the GamePad had an external console.

Most of the Joy-Con's features are shown in Figure 7.7, but it also includes motion tracking similar to that of the Wiimote, and the HD Rumble. In one of the minigames of *1-2 Switch*, the player holds one of the Joy-Cons and guesses how many virtual balls are inside of it. The HD Rumble provided a rumble sensation that allowed players to accurately guess the number of virtual balls [25].

FIGURE 7.7: Joy-Con controller set for the Nintendo Switch

The Switch and Joy-Con are very new, so it is hard to say anything about how impactful they might prove to be. Since its launch, the Switch has had strong sales figures. Looking at Nintendo's history in controllers they are likely to keep the innovation within their controller design going.

7.7 Rhythm Game Controllers

Many variants of rhythm and music games exist, as well as the many different types of controllers designed for them. Some of the most popular music rhythm games are part of the *Guitar Hero* series, which uses the guitar-shaped controller displayed in Figure 7.8. It is a fairly simple controller, featuring five fret buttons, a strum bar, and a whammy bar. These two series became very popular, but the publishers oversaturated the market. 21 games were released between the *Guitar Hero* and *Rock Band* series over a period of five years [26].

FIGURE 7.8: *Guitar Hero* controller

This type of controller is designed for a singular purpose. It is supposed to feel natural and simulate playing a real instrument. Since the Seaboard is a musical instrument, this is not a promising indication for its viability as a game controller. However, the Seaboard is a lot more advanced than a plastic guitar with a few buttons.

7.8 Summary of Game Controllers

Although the reviewed controllers span over a 20 year time period, and represent significant developments they have several common features. Standard design consists of two analog sticks, one left and one right, each of which can be pressed as a button; a D-pad on the left side; four action buttons on the right side; a bumper and a trigger button on both the left- and right-hand backsides; a small number of auxiliary buttons located near the middle ("Start", "Menu", etc.). This design has become so refined over the past 15 years, that in order for it to change significantly, something truly exceptional must happen.

Additionally, elements like headphone jacks, Bluetooth technology, motion sensors and vibrating grips are becoming integrated parts of the communication and feedback between the game and the player. Other changes are mostly of ergonomic nature like textured grips, movement of auxiliary buttons for more natural finger movements and ergonomic shapes.

This research has provided me with additional insight into the capabilities and constraints of various state of the art controllers. One takeaway here is that some controllers are highly specialized and were designed to do only one thing. Other controllers are designed to be as good as possible for multiple games and hardware. By identifying constraints and advantages for traditional controllers, I gain knowledge which might transfer to the Seaboard.

Chapter 8

ROLI Seaboard

The Seaboard is a new type of keyboard by ROLI, a company that specializes in creating new music-making devices. It is designed to be a music-creation device which merges traditional keyboard design with modular technology [27]. There are several variants of the Seaboard, but the one used in this project is the Seaboard RISE 25 [28].

The RISE might look quite similar to a normal keyboard, but there are no separate keys. Instead, it consists mostly of a single pressure-sensitive surface with waves of key-like geometry [29]. When used as an instrument with Equator, a software synthesizer and sound engine from ROLI, each of the controls marked in Figure 8.1 work as follows:

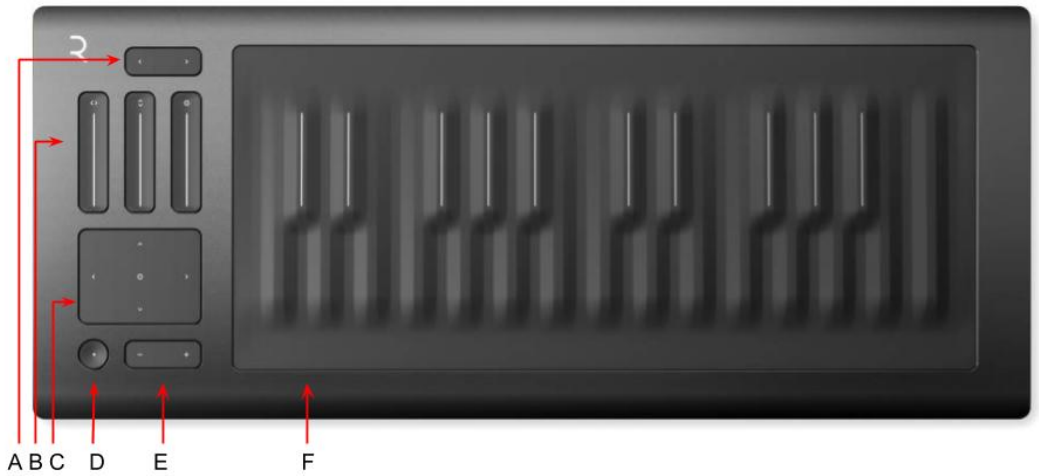


FIGURE 8.1: Seaboard RISE 25

A **Preset switch**: Select presets within Equator.

B **Touch faders**: Control dynamics of glide, slide, and press.

C **XY touchpad**: Control sound parameters in Equator.

D **Power button**: Indicates whether the RISE is on or off, and battery charge status.

E **Octave shift**: Switch octaves.

F **Touch sensitive surface**: Keywaves are the raised surfaces in the middle, while ribbons are the flat surfaces above and below the keywaves.

The Seaboard operates through the following five dimensions of touch: strike, press, glide, slide, and lift [30].

- **Strike**: The velocity and force with which a keywave is struck.
- **Press**: The pressure and continuous touch applied to the keywave after it is struck.
- **Glide**: Horizontal movements from side to side on a keywave or along the ribbons.
- **Slide**: Vertical movements up and down along a keywave.
- **Lift**: The velocity with which a keywave is released.

Combining the five dimensions of touch in different ways gives rise to a multitude of possibilities. In games, for instance, each dimension could be tied to a command while different combinations of dimensions could allow the player to perform more complex actions. Couple this with the fact that there are 27 distinct surfaces on the Seaboard (25 keywaves and the upper and lower ribbons) and the potential for many distinct actions increases dramatically.

Chapter 9

Game Enjoyment

When playing a game, it is imperative that the player receives some measure of enjoyment from the experience. A game might offer some form of value, but if the players are not enjoying the game, they will not play it [31]. For instance, a game might offer educational value, but will be largely ignored if it is not fun. What people find enjoyable can vary greatly between individuals, and no game will appeal to everyone. However, there are certain criteria which, when met, can increase a player's enjoyment of the game.

But how can enjoyment be measured? In this section, we study two frameworks for evaluating game enjoyment: Sweetster and Wyath's (2005) GameFlow model, and Malone's (1980) heuristics for designing instructional computer games. Additionally, we will take a look at a study which examines how video game interactivity can affect game enjoyment by Stalski *et al.* (2011). To create an enjoyable game, I will use the criteria and guidelines from these articles when developing the game concept.

9.1 GameFlow

The GameFlow model was created by Sweetster and Wyath for evaluating player enjoyment in games. It was adapted to games from Csiszentmihalyi's (1990) research into what makes experiences enjoyable and the resulting concept of *flow*. In short, flow is an experience of deep enjoyment that people are willing to expend a great deal of effort simply to feel it [31]. The elements that make up flow provided the basis for the GameFlow model. The model consists of the following eight elements: concentration, challenge, skills, control, clear goals, feedback, immersion, and social interaction.

9.1.1 Concentration

Sweetster and Wyath state that in order to enjoy a game, it requires concentration. The more the player concentrates on the game, the more absorbed they will become. When the player is concentrating so much that they are completely absorbed, they do not notice anything outside of the game.

In order to maintain the player's interest, game should grab their attention fast, and maintain it for as long as the game is played. One way to capture the player's attention is by providing something worth paying attention to, for instance by the use of a rich and interesting game world, tempting the player to explore it. From the beginning of a game until the end, it it important to increase the workload in order to keep the player concentrated. If the game becomes too easy, concentration is lost. However, it is important that the workload does not exceed an appropriate amount, such as not to overly frustrate the player. Balance is key.

Games should not require the player to complete tasks that do not feel important. One example could be minimal effort gathering quests, or so-called fetch quests. If the famous hero charged with defeating the evil dragon is asked to round up an entirely unrelated farmer's escaped chickens, he should be able to say no and carry on with his dragon-slaying duties.

Lastly, players should not be distracted by with nongame-related actions, such as setting options, while playing. This includes the game interface, which should be as non-intrusive as possible, providing the player with the maximum amount screen space for gameplay.

9.1.2 Challenge

The most important aspect of good game design is challenge, according to Sweetster and Wyath. A game should be appropriately challenging, match the player's abilities, vary difficulty, and keep a good pace. If a game is too hard it can be frustrating or discouraging, but if it is too easy it can be boring. Therefore, a game's difficulty should gradually increase as the player learns how to play the game.

By testing the limits of the player's performance, through distinct and challenging situations, games create enjoyment. A positive experience, such as the successful completion of a challenge, makes the player wants to keep playing. Satisfaction comes from completing difficult tasks, surpassing opponents, mastering skills, and reaching desired goals. Progress is its own reward, through feelings of personal triumph.

9.1.3 Player Skills

Skill development and mastery is key to enjoyment in games. It is important for the player to feel like they are improving at the game, and that their newly acquired skills are put to the test. Games should provide a learning curve, and introduce new skills gradually over time.

To begin with, players should be taught the basic mechanics of the game through a tutorial. The tutorial should absorb the player and teach them quickly and easily. Alternatively, players can learn as they play, learning and practicing skills as they need them and in context. Rewards are also an important part of learning to play a game. By improving their skill, the player can take on harder challenges, thus earning better rewards.

When designing a game there are more indirect ways of allowing a player to learn the game. By following platform conventions, trends in game design, and controller layout, players can translate skills from similar games. Game controls and interface should be consistent, simple, and non-intrusive to provide easy access to the game.

9.1.4 Control

Players need to experience a sense of control over what actions they perform in the game in order to experience flow. This also includes actions regarding the game interface, menus, and input devices. In short, what the player intends to do should be translated to actions in-game. Basic control should be easy to learn, providing quick access to core game mechanics.

Menus should be easy to use and intuitive, prioritizing functionality over aesthetics. Through the menus, players should be allowed to either remap controls or the game should support different styles of play. This provides the player with the option of choosing a playstyle that fits them, further improving their sense of control. Being able to save and load is also important, allowing players to explore the world at their own pace.

Actions that the player performs should have some impact on the game world. This provides some measure of control in the sense that the player's actions have consequences, and that they are taking part in shaping the world they are in. Similarly, the game should not burden the player with choices that are unimportant and have no impact on their progress or the world.

9.1.5 Clear Goals

Games should establish an overarching clear goal early which should be presented to the player in a clear and straightforward manner. When a goal has been reached, a new one should be provided at an appropriate time. Besides an overarching goal, games should also have multiple smaller goals. These can be goals that contribute towards the completion of the main goal, or optional goals that provides a reward for the player but which is not crucial to the progression of the game. Goals can be established through the use of the story within the game, or by providing a "briefing" explaining the mission parameters and measures of success.

9.1.6 Feedback

Feedback must be provided to the player at appropriate times. To provide the player with information about their progress towards objectives, feedback must be frequent. Games can use scores to show players that they are improving, and encourage continued play. Another way to provide feedback is to have the player be able to identify their status in the game through interface and sound. Health bars can be used to see injured units, or and audio cue can tell the player that they've been noticed by an enemy. Another important aspect is to provide immediate feedback for player actions. When a player interacts with an object, reacts to an event, or gives an order there should be an immediate reaction from the character or units the player controls.

9.1.7 Immersion

Deep and effortless involvement in the game is a part of flow. It can result in loss of concern for self, or for your surroundings, and can alter perception of time. Due to the time and effort invested, players also often feel deep emotional investment in a game. While playing, the game is the center of the player's attention and directly affects their emotions. When immersed, the player forgets that they are outside of the game and are only interacting with the world through a controller and an interface.

People play games to calm down and escape everyday worries, be it work-related or other problems. Games allow players to live in a world with different restrictions than the real world, where they can do things they would otherwise not be able to do. By enticing the player, the game keeps them in the game world and involves them emotionally. Immersive games also affect players through audio and narrative to draw them in further. Ambient sounds help the player feel like they are present, while music can evoke emotion or build suspense. A good narrative gives the player

context for who their character is and what is happening in the game world, which immerses the player in the story of the game.

9.1.8 Social Interaction

While social interaction is not an element of flow as it can interrupt immersion through the contact of people outside the game, it is still a strong element of enjoyment in games. People play games to socialize, even if they would not normally enjoy the game. Social interaction is not part of playing the game like the other elements of GameFlow are, but the game is rather a means to provide social interaction.

Social interaction is supported in games through player competition, cooperation, and connection. Game enjoyment can be enhanced with player-to-player interaction, both inside and outside of the game. People enjoy being social and interacting while playing games, watching others play games, or talking about games. Competition is also an important aspect of social interaction, since players gain satisfaction from competing against and defeating other people.

9.2 Challenge, Fantasy, and Curiosity

Although focused on instructional computer games, Malone states that "these same ideas can be applied to other educational environments and life situations. In a sense, the categories I will describe constitute a general taxonomy of *intrinsic motivation* of what makes an activity fun or rewarding for its own sake rather than for the sake of some external reward" [32].

9.2.1 Challenge

In order for a game to be challenging, the outcome must be uncertain. A goal that is possible to fail needs to be provided. The very notion of a game implies that there is an objective to be completed.

Obvious goals should be provided in simple games. They can be made obvious by the use of visual effects or fantasy. More complex games can allow for more complex or abstract goals. If a game has no built-in goals, the player should be able to easily create one. Also, players must be able to tell if they are progressing towards their goal.

Uncertain outcomes are important because they build suspense. If players are guaranteed to win, the game is likely to become boring. One way to add uncertainty is

by variable difficulty levels. Difficulty can be adjusted by letting the player choose, or it can be determined automatically based on the player's skill, or the player can be matched with an opponent of similar skill. Through the use of multiple level goals, the challenge increases. Even though the completion of one goal can be fairly certain, the completion of several is more uncertain. These extra goals can be meta-goals or subgoals. While beating a level can be a goal, beating the level within a certain time can be a more challenging subgoal. Hidden information can also make some outcomes uncertain. By selectively revealing information, the player does not see the whole picture, but can make educated guesses or deductions.

Success when challenged improves self-esteem. It makes the player feel that they accomplished something and makes them want to play more. Consequently, failure can lead to loss of desire to play more. Challenge should be inviting, not discouraging.

9.2.2 Fantasy

Many associate fantasy with dragons, magic, elves, and the like. In this case, however, fantasy refers to the setting and objects of the game world, though those elements can certainly be a part of the world. Malone divides fantasy into to categories: *intrinsic* and *extrinsic* fantasy.

Extrinsic fantasies rely solely on whether or not the skill is used correctly, for instance if an answer is right or wrong. This is the game simply reacting to the player's input. In intrinsic fantasies the fantasy depends on the skill, and vice versa. The player receives some sort of response to their actions which might impact how they choose to act next. Intrinsic fantasies are generally more interesting, because they can be applicable to real-life situations, thus being more immersive. Malone also argues that fantasy helps satisfy emotional needs in players. It is difficult to know what emotional needs players may have, but games with many different fantasies will likely have a broader appeal.

9.2.3 Curiosity

Curiosity is the motivation to learn and explore, independent of goals or fantasies. Game worlds should be interesting and new, but not incomprehensible, to provoke the player's curiosity. Malone distinguishes between *sensory curiosity* and *cognitive curiosity*.

Sensory curiosity attracts attention through audio and visual effects. Changes in environment, sound, and light can focus attention. These effects can be used as decoration, fantasy enhancers, rewards, or as a representation system. Decorations are

sound and visuals which play out regardless of what the player does. It provides some interest, but too much can be boring. Fantasy enhancers are similar to decorations, but also provides atmosphere and immersion because of its relation to the fantasy. By using sounds or visual effects when the player triumphs or receives a reward, it increases player satisfaction. Graphics and sound can be used as a representation system to convey information more easily than with words or numbers. Many games uses certain music when the player is in combat, so the player knows that there are enemies nearby when that music plays.

Cognitive curiosity is about piquing the player's interest to explore or learn more about the game world or something in the world. One way to go about it is to present just enough information so that the player's knowledge is incomplete and thus causing the player to feel compelled to fill that gap in their knowledge. Another method is to use inconsistencies in information that is presented to the player. The player may hear conflicting arguments from different characters and decide to investigate to find the truth of the matter. Games should be responsive to the player's actions, and provide feedback. To engage curiosity, feedback should be surprising. Events that occur in a game may seem random and surprising at first, but as the player gains more knowledge they find that there was some underlying logic to the events.

9.3 Natural Mapping in Game Controllers

In the context of natural mapping, natural refers to the extent to which players perceive the interactivity to the game to be predictable or logical. Whether a controller is natural also depends on both technology and individual differences. People can be conditioned to find certain interfaces highly natural, such as a mouse and keyboard. Today, most people with access to a computer know that mouse movement on an XZ-plane translates to pointer movement on the screen in an XY-plane because it has become natural for them. Realistic controllers such as handlebars, steering wheels, and flight sticks correspond to real-life modes of vehicle operation. As such, they would feel highly natural if operating a fitting vehicle in a game. Traditional controllers, like the standard controllers for PlayStation, interact with the game not through real movements, but by pressing buttons or other actions that are not strongly connected to real-life actions. According to Stalski *et al.*, such restrictive physical involvement lessens immersion.

Mapping refers to how the actions performed by the player through the controller translates to what happens within the game. The action the player performs can range from arbitrary (unrelated to reaction in game) to natural (related to action in game). The extent to which mapping is present in a controller affects the player's perceived controller naturalness.

More naturally mapped controllers should be more intuitive to use, because they are based on a real-life counterpart. People already have a mental model of how such a controller might be used. The player can use experience from real-life and apply it to the game which helps facilitate spatial presence. Spatial presence is the sense of being physically present in a virtual setting [33].

Skalski *et al.* discuss four types of natural mapping, in order from most arbitrary to most natural: directional natural mapping, kinesic natural mapping, incomplete tangible natural mapping, and realistic tangible natural mapping.

9.3.1 Directional Natural Mapping

Directional natural mapping is the most basic way to increase natural mapping in controllers. It can be produced by creating a correspondence between directions from controller to the results shown in the game. Many computer games played with a mouse and keyboard have this type of mapping. Moving the mouse to the left or right will cause the player character in the game to turn in the corresponding direction. Moving the mouse forwards or backwards will cause the character to look up or down, respectively. Games that uses the mouse to make the character look around often has character movement bound to the 'W', 'A', 'S', and 'D' keys on the keyboard. 'W' and 'S' move the character forward or backward, while 'A' and 'D' causes the player to strafe left or right, respectively. Without a minimum of natural directions, the player may be confused or frustrated, since unnatural actions are counterintuitive to the player's existing knowledge.

9.3.2 Kinesic Natural Mapping

By using body movements that correspond to real-life actions without having a realistic and tangible controller we get kinesic natural mapping. This is basically what Kinect, a motion peripheral for Xbox 360 and Xbox One, aims to do. Take the game *Kinect Star Wars*, for example. It has no physical controller, but the Kinect's camera captures movements. In one of the game's modes, players take on the role of a Jedi and can wield a lightsaber and use the Force through hand gestures, although players did not find the game to feel very natural because of poor controls [34, 35]. Normally, close mapping of actual body movements translated to the same movement in a game is perceived would be perceived as fairly natural. However, missing a tangible controller reduces naturalness.

9.3.3 Incomplete Tangible Natural Mapping

The Wiimote, which was discussed in Section 7.4, is an example of incomplete tangible natural mapping. It is a controller that partially feels like and simulates the behavior of objects in the game. Players grasp the Wiimote similarly to how they would grasp objects in the game, like a tennis racket or baseball bat. As in kinesic natural mapping, the player performs realistic motions, but they also have a tangible controller which they use similarly to how they would perform a real-life action. However, the tangibility is limited. In the *Wii Sports* bowling mini-game, players hold the Wiimote in much the same way as they would a bowling ball, but the controller does not have the same weight or shape of a real bowling ball, making the experience incomplete.

9.3.4 Realistic Tangible Natural Mapping

This type of natural mapping adds a realistic and tangible element to provide a higher level of natural mapping. These type of controllers are very close in resemblance and function to their real-life counterparts, and they appear naturally suited to perform the tasks of the game. A driving game with a steering wheel, pedals, and a gear lever will feel highly realistic, because the actions of the player corresponds directly to the movement of the car in the game that the player is playing. Additionally, the player's experience and knowledge of driving a real-life car translates to how the game is played, allowing their mental model of driving a car to close the gap between the controller and what is showed in the game. The result is an enhanced spatial presence, immersing the player further into the game.

9.4 Summary of Game Enjoyment

When playing a game, it is important that the player enjoys the experience. Through Sweetster and Wyath's GameFlow model, Malone's heuristics for designing instructional computer games, and the effect of video game interactivity on game enjoyment by Stalski *et al.*, some key criteria for enjoyable computer games may be established.

Sweetster and Wyath's GameFlow model focus on eight main elements: concentration, challenge, skills, control, clear goals, feedback, immersion, and social interaction. These are very much related to the player involvement when playing, requiring concentration and developing skills to master the game, challenges that needs to be resolved and unpredictable outcomes creating some element of risk. The player needs to experience a sense of control, and basic control should be easy to learn. Further, the game must provide some clear goals to be achieved and provide feedback

towards the completion of these goals. Immersion into the game, and leaving the real world for a while is also part of the flow, although this is probably more a result of concentration and focus on playing rather than a game criterion. An element of social interaction may add to the enjoyment as competition or sharing experiences creates more interest and may add to the sense of skill. When these criteria are fulfilled the player will experience what has been defined as flow, an experience of deep enjoyment that people are willing to expend a great deal of effort simply to feel.

Important motivating elements are challenge, fantasy, and curiosity as suggested by Malone. Challenge is related to the feeling of risk when the outcomes are uncertain, making the feeling of success greater when the challenges have been overcome. It is also suggested that games with many different fantasies will likely have a broader appeal. Finally, elements that create sensory curiosity through audio and visual effects or cognitive curiosity through exploring or learning the game world improve immersion and sparks interest in the game.

Stalski *et al.* argue that the naturalness of a controller affects game enjoyment. They describe four ways in which a controller can be naturally mapped, from least to most natural: directional natural mapping, kinesic natural mapping, incomplete tangible natural mapping, and realistic tangible natural mapping. The more natural the controller, the more immersed the player will be, because they forget that they are participating through a controller.

There are several parallels in these theories of what makes a game interesting and enjoyable, and the challenge in my project will be to create a game concept that utilizes the Seaboard as a controller in a way that will enhance the game experience. The physical limitations of the Seaboard will impose restrictions as to which level of mapping that is possible to obtain. Identifying what the most natural operations are that can be performed with the Seaboard is critical for meeting the criteria and guidelines provided in this section.

Part III
SeaRhythm

This part describes the process of developing my findings from the prestudy, combining it with various game concepts and deeper knowledge of the Seaboard into a proposed game concept.

Initially, I describe the capabilities and limitations of the Seaboard as a controller. Then I present some inspirational games that I believe have some functionality that could be handled by the Seaboard, and that may indicate which concepts and game mechanics that are achievable. Based on these findings, I present some game concepts that may fit the Seaboard, before making my selection and outlining the chosen game concept that I have called *SeaRythm*. Next, the game's design is described by connecting concept of *SeaRhythm* to the framework for game enjoyment. A description of the relevant technology that has been utilized for the project is presented before supplying a list of requirements for the game and an architectural overview. Finally, the testing approach is discussed.

Chapter 10

Advantages and Limitations of the Seaboard

In this project, I will attempt to design a game concept which will use the Seaboard as a game controller. Part of the task is to evaluate and identify what type of games fits the advantages and limitations that the Seaboard offers. Can it be used to play traditional 2D or 3D games, or does it impose too much of a disadvantage on the player compared to a traditional controller? Just like it was difficult to navigate a 3D environment with a D-pad, so may it be difficult to navigate a 2D environment with the Seaboard. With the layout and functionality of the Seaboard, it seems difficult to control multi-directional motion and speed in an intuitive and natural manner that traditional controllers do with ease.

As the Seaboard is based on new hardware and software, there is limited information and availability of useful third party software which adds to the challenge of creating a fitting game. The features of the Seaboard are described in Section 8, and the functionality of strike, press, glide, slide, and lift could open for some elements of dynamic control. Actions can be scaled up or down using the slide functionality. For instance, it could be possible to "shoot" by sliding fast up on a keywave.

The strengths of the Seaboard seem to be multi-touch functions, as different combinations of keys can easily be operated. The keys are grouped by octaves, and located in two rows, providing possibly eight groups that can be assigned different capabilities. In total, twenty-five keys provides the ability to map a lot of different abilities. However, this can also be a practical challenge as the player would need to remember 25 different key bindings. This could be made more intuitive by grouping keys by similar functionality.

It seems that Seaboard easily can utilize the concept of controlling lane oriented activity, and its possibilities for numerous different commands can allow for various types of activities within each lane. It may be casting spells, shooting, putting out traps, and more. Combining keys can add to this, and possibly raise the challenge

as the player's skill increases.

In summary, it seems the Seaboard would best benefit from a game with simple movements or no movement at all required by the player. The game can incorporate complex actions because of the multi-touch functions and the different dimensions of touch. However, I must be careful not to make the controls frustrating to use or learn by adding so many combinations that the player is impeded in any way. This can be mitigated somewhat by grouping similar actions together.

Chapter 11

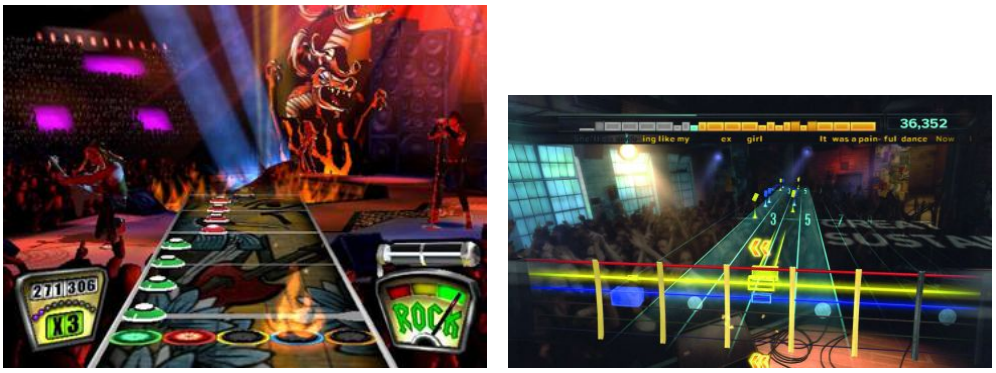
Inspirational Games

In this chapter, I present a few games that I took inspiration from when working on the concepts in Chapters 12 and 13. I was also inspired by other games, but the games showcased here are those whose mechanics feature most prominently in the concepts I came up with.

11.1 Guitar Hero and Rocksmith

Guitar Hero (2005) is a music and rhythm game where the player plays a guitar-shaped controller to simulate playing music. *Rocksmith* (2011) is a similar music game, but instead of a guitar-shaped controller, it integrates a real electric guitar or bass guitar as a controller.

Both games play similarly with vertically scrolling lanes and colored notes paced to the music. A comparison of the two games is shown in Figure 11.1. In *Guitar Hero*, the controller has five fret-like buttons on the neck that match the color and



(A) *Guitar Hero* (B) *Rocksmith*

FIGURE 11.1: Side-by-side comparison of *Guitar Hero* and *Rocksmith*

position of the lanes in the game. It also has a strum bar and a whammy bar on the body. To play a note, the player must hold the correct buttons and hit the strum bar at the same time as the notes passes the marked area near the bottom of the screen. *Rocksmith*, however, uses the full range of frets on the guitar and looks a bit different. Here, each string is represented by a color, and each lane is equivalent to a fret. In the case of Figure 11.1b, the note to be played is blue and on the second lane, which is equivalent to playing the second fret of the fourth string from the bottom (the "D" string). Both games award the player a score on each song they have played based on how many notes they managed to hit.

Rocksmith provides dynamic difficulty, adjusting the note density automatically while playing. The better a player does, the higher the density gets, until they are playing the entirety of the song. Should the player miss a lot of notes, the density decreases. Songs are divided into several parts. One part could be a riff where you play the same sequence several times in a row, while another could be a solo which is much less repetitive but likely harder. If a song has several parts with the same riff it could start off with a low note density and if the player does well

11.2 Crypt of the NecroDancer

Crypt of the NecroDancer (2015) is a single-player beat-matching rhythm game with roguelike elements. Roguelikes are a subgenre of role-playing games and are characterized by procedurally generated levels, tile-based graphics, and permanent death of the player's character [36]. The player views the game world, consisting of different types of tiles, from a top-down view. To be able to act, the player has to match the beat of the music in the game. The game can be played using a mouse and keyboard, a gamepad (like an Xbox 360 controller), or even a dance mat [37].

In the game, players move through several floors of a labyrinthine crypt filled with monsters and treasure. The goal is to progress through all the floors and defeat the final boss at the end. Players can move or attack in four directions, but can only do so if they execute the action on the beat of the game's music. The player attacks monsters by moving into either the tile the monster is standing on, or the one it is moving to. Monsters also move to the beat, and can harm the player in the same way. Each type of monster has a unique, but consistent movement pattern. For instance, the blue slime monster in Figure 11.2 will switch between moving up or down every two beats.

As the player defeats monsters they receive gold and a gold multiplier. Miss the beat, and the multiplier resets. The player also does not move or attack until they begin hitting the beat again. Gold can be used to purchase items that enhance the player character. Throughout the crypt there are also treasure chests and diamonds. The music goes faster and monsters get tougher as the player moves to each new level,

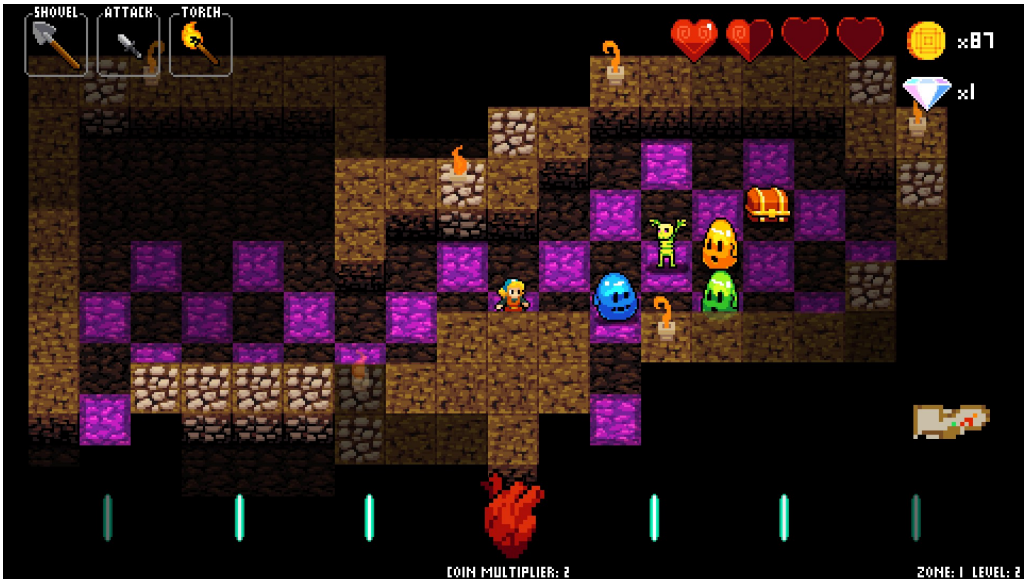


FIGURE 11.2: *Crypt of the NecroDancer*

making the game progressively harder. Should the player be defeated, they return to the game lobby and lose all their items except for the diamonds. The diamonds can be spent to buy permanent upgrades for the player character or to unlock new treasures within the crypt. Players can also unlock new characters by completing certain goals in the crypt, providing replay value to the game.

11.3 Magicka

Magicka (2011) is an action-adventure game for one to four players set in the fantasy world of Midgård. The players are tasked with deafeating an evil sorcerer that is terrorizing the world. Compared to other games of a similar type, *Magicka* prioritizes adventuring and combat, with less emphasis on items as rewards. Aiming and shooting spells, as well as movement, are controlled with a mouse. Creating spells is controlled with a keyboard [38].

There are eight elements with which the player can create spells: water, life, shield, cold, lightning, arcane, earth, and fire. Two additional elements can be created by combining fire and water to create steam, or by combining water and cold to create ice. Thus, the players have access to a total of ten elements. By combining up to five of these, players can create a large variety of spells. Some elements complement each other, while others, like fire and cold, cancel each other out. For the most part players can experiment with spells, but there are certain special combinations with

unique effects called "Magicks" in the game which can only be cast if the player has acquired the corresponding spellbook.

Spells can be be cast in four different ways: as a projectile or beam, as an area of effect, as a weapon enhancement, or on the player's themselves. All elements normally deal damage, except for life which heals, and shield which creates barriers. In order to cast a spell, the player sequentially presses the buttons for what elements they want to use. The sequence they are pressed in does not matter except for "Magicks". Then the player chooses one of the four ways to cast the spell using various mouse buttons. Figure 11.3 show players attacking monsters with spells. The icons in the bottom left corner represents each of the elements the player can use to craft spells.

FIGURE 11.3: *Magicka*

11.4 Space Invaders

Space Invaders (1978) is a 2D fixed shooter game where the player controls a cannon and uses it to defend against attacking aliens. The cannon can be moved horizontally across the bottom of the screen shown in Figure 11.4, and can shoot straight up at the aliens coming from the top of the screen. Players earn points by destroying aliens, with aliens in rows further back being worth more. The goal is to destroy all of the aliens, but there can be several waves depending on the version. As more aliens are destroyed, they begin to move faster, which can carry over to subsequent waves [39].

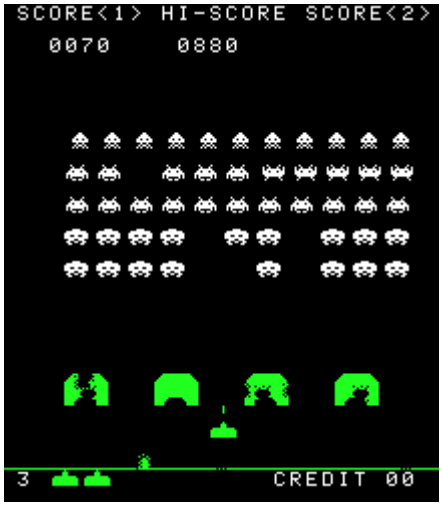


FIGURE 11.4: *Space Invaders*

The aliens shoot towards the bottom of the screen, attempting to destroy the cannon. Several destructible bunkers are placed in front of the cannon, and provide partial protection from the aliens. If aliens reach the bottom or if the player's last cannon is destroyed, the invasion is successful and the game is over.

11.5 Summary of Inspirational Games

My inspiration from existing games has been taken from *Guitar Hero*, *Rocksmith*, *Crypt of the NecroDancer*, *Magicka* and *Space Invaders*. Although these games span from 1978 to 2015, they all have elements that can possibly be performed by using the Seaboard as a game controller.

Guitar Hero and *Rocksmith* are music and rythm games, and play similarly with vertically scrolling lanes and colored notes paced to the music. Both games awards the player a score on each song they have played based on how many notes they managed to hit. In *Rocksmith*, the songs increase in difficulty as the player's skills improve. The concept of lanes and using a beat seem very appropriate for the Seaboard.

Crypt of the NecroDancer is a beat-matching rythm game where the players have to move through a labyrinthine crypt filled with monsters and treasures. Players can move or attack in four directions, but can only do so if they execute the action on the beat of the game's music. Enemies also move to the beat, each with a consistent and unique movement pattern, and can harm the player. Players are rewarded when defeating the monsters, but are punished if they miss the beat. As players progress

through the game, it gets harder. The concepts of using varying rhythm, different types of monsters, and random rewards are all elements that may be used with the Seaboard.

Magicka is an action-adventure game with a priority on adventuring and combat, with less emphasis on rewards. Creating spells is a main feature of the game, and is controlled with a mouse and keyboard. There are eight elements which the player can combine to create spells. Some elements complement each other, while others do not work together. There are four ways to modify how the spells are cast, using various mouse buttons. The concept of using spells in both an attacking and a protecting manner, seem to be possible with the Seaboard.

Space Invaders was an early 2D game where the player controls a cannon by moving horizontally and shooting up at the aliens coming from the top of the screen. The aliens shoot down, attempting to destroy the cannon which is partly protected by bunkers. The speed of the remaining aliens increases as more of them are destroyed. The simple movement of both player and enemies seem to complement the Seaboard, since it is limited in ways to adequately control a character in a more complex environment.

Chapter 12

Game Concepts

While carrying out the prestudy, I devised a few concepts for games that might fit the Seaboard. These concepts were derived from the games presented in Chapter 11; specifically from some of the mechanics of those games that I found interesting.

12.1 Rhythm Invaders

This idea originated from a conversation with my supervisor, Alf Inge Wang. The game would work much like *Space Invaders* except that the player would hit keys on the Seaboard to move and shoot the cannon. I iterated upon the idea and included elements like lanes from *Guitar Hero* and the rhythm-based combat from *Crypt of the NecroDancer*.

Enemies would act similarly to how they act in *Space Invaders*, and instead of one large group there would be several smaller groups. Each group would be constricted to one lane, and the player can move the cannon from lane to lane. A group of keys would be assigned to each lane, and within that group each key would correspond to a position in that lane. To shoot, the player would need to press the key corresponding to the position of the cannon. Look at Figure 12.1 and imagine that the five left-most keys corresponds to positions 1 through 5 in Lane A. If the cannon is in A1 but wants to shoot from A5, the player would tap the key corresponding to A5 twice to the beat. Once to move, and again to shoot. The red lines are range indicators. The cannon's projectile will travel from the bottom black line to the first red line in one beat, and then to the second red line the next beat and so on. If the player knows how far the enemies move per beat, they can predict when and where they need to be to hit their shots. Additionally, difficulty could be adjusted by having each group of enemies behave differently. One group could be large and tough, requiring several hits to destroy; while another could be small and fast, moving several positions per beat.

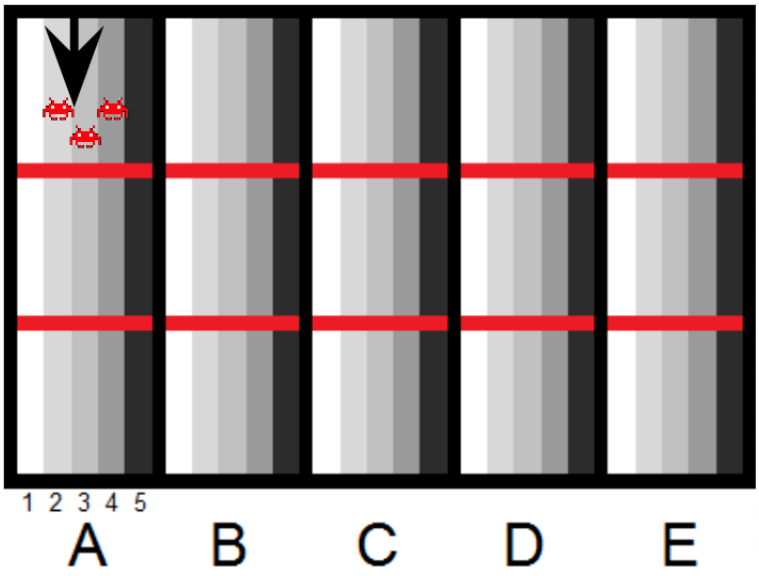


FIGURE 12.1: Rhythm-based Space Invaders concept

Although rhythm-based and capable of using all the keys of the Seaboard, this concept does not take advantage of the Seaboard's capabilities. In regards to the five dimensions of touch explained in Chapter 8, this concept currently uses only strike. Some changes would have to be made for this concept to be worth playing with the Seaboard.

12.2 Magic Rhythms

In *Magic Rhythms*, which is inspired by the combat system of *Magicka*, the player controls a wizard who is attempting to protect his tower from attacking enemies. This concept features a rhythm-based king-of-the-hill play style, where the player character is stationary in the middle of the screen on top of a tower. Enemies enter from all sides of the screen, and the player must keep them at bay until they have enough magic power to teleport themselves and the tower away. The game is 2D and viewed top-down.

Although the player is stationary, they can still rotate. By gliding their finger right or left across one of the Seaboard's ribbons, the player can rotate clockwise or counter-clockwise. A subset of keywaves each correspond to an element, which can be combined similarly to *Magicka*, but the player is restricted to selecting one element per beat. One group of four keywaves are used to attack, where each key modifies if the spell is cast as a projectile, beam, cone, or circular area of effect. Spells must be cast on the beat and cannot be cast at the same time as an element is added to the

spell. If a beat is missed, any stored elements in the spell are lost. Rotating is not rhythm-based, because then it would have to happen in increments instead of being fluid. The power of attacks can be varied by sliding up on one of the keywaves used for attack. More powerful attacks can increase the radius of the projectile, width of the beam, length of cone, or diameter of area of effect. Powerful attacks consume mana, which is partially replenished per beat.

There are various types of enemies, some of which are resistant to certain elements, but can be vulnerable to others. If an enemy reaches the tower, they will begin attacking it. Should the tower be destroyed before the player can teleport, the game is lost.

Chapter 13

SeaRhythm

The game I have decided to create is *SeaRhythm*. It is based mostly on a combination of *Rhythm Invaders* and *Magic Rhythms* from Sections 12.1 and 12.2. It also adds some of the most promising mechanics from the games in Chapter 11 into a concept that will be both enjoyable and able to take advantage of the Seaboard's capabilities. In this chapter I describe *SeaRhythm* as a conceptual (near-)complete game. It does not take into account anything that has been cut due to time constraints or for other reasons.

In *SeaRhythm*, the player controls a merchant ship that is traveling across the ocean from port to port. Along the way they encounter pirate ships who will attack them and try to steal the player's wares. In order to get to harbor safely, the attacking pirates must be eliminated. The player's ship is not equipped with any weapons, but that does not matter because they have a powerful wizard at their disposal. The wizard can be commanded to combine different elements to rain down destruction upon the player's enemies and secure their safe passage to port. Should the player's ship be destroyed by the pirates, the game is over, and they must start from the beginning again.

SeaRhythm is a rhythm-based top-down scrolling shooter game. A song plays for each level and provides the beat. If the player is not defeated before the song ends, that level is complete. The player is limited to moving between different lanes like in *Rhythm Invaders*, except that there are no positions within each lane, and the ribbons are used to move horizontally instead of the keywaves. This frees up a large number of keys so that they can be used for other actions.

Combat is similar to that of *Magic Rhythms*, but with some key changes. An attack requires three parts: an attack modifier and two elements. The attack modifier decides if the attack is a projectile, line, cone, or an area of effect, while the two elements decide what the attack consists of. For instance, fire and earth would create a lava attack. With a projectile modifier a ball of molten rock would be launched from the player's ship. Also, to increase the power of attacks, players instead need to hold an

attack key down and release it a number of beats later. Using slide on an attack key, i.e. dragging a finger quickly from the bottom to the top of a keywave, increases the velocity of the attack. Players can simply strike a combination of three attack keys to perform an attack, without the need for additional modifiers. Movement and attacks must be performed on the beat, and cannot be performed at the same time.

Enemy ships take on attributes from both *Rhythm Invaders* and *Magic Rhythms*. Some ships are armored and tough, while others sacrifice armor for speed. Some could even be resistant to various types of magic, and vulnerable to others. Enemy ships appear from the top of the screen and move down towards the bottom. Enemy movement and attacks also occur on the beat and they will attempt to shoot their cannons at the player. If an enemy ship is sunk the player will receive a reward, but if the enemy manages to reach the bottom of the screen it will disappear off-screen and the player will receive no reward for it.

Difficulty can be adjusted by how many and what types of enemies there are, as well as how often enemies appear. It can also be adjusted by using a songs with higher or lower beats per minute. Game modes which impose restrictions upon the player, such as reducing the number of elements they have available, could also be introduced to increase difficulty.

The player earns treasure from destroying enemy ships and from completing a level. By holding the beat for an extended amount of time, the player gains a score multiplier, which increases the amount of treasure the player is awarded for destroying pirate ships. The multiplier is lost if the player misses the beat, or if they are harmed by hostile attacks. Treasure can be spent between levels while in harbor to purchase upgrades for the player's ship, new abilities, or usable items. The total amount of treasure that a player gathers is considered their game score, and will be included on a leaderboard when the game is over.

Chapter 14

Game Design

This chapter will connect the game concept in Chapter 13 to the frameworks presented in Chapter 9. By basing the design of the game on these principles, the game should be more enjoyable. Not all of the principles are included, but I have attempted to include as many as seems fitting while still remaining feasible.

So as not to overwhelm players who are playing the game for the first time, not all features of the game are immediately available. Consequently, several things that are described in this chapter are not implemented in the game since it is merely a prototype. For a full experience of the concept that has been presented, the game would need to undergo further development. However, enough parts of the game have been implemented so as to make the first level of the game playable.

14.1 GameFlow

The very nature of a rhythm-based game like *SeaRhythm* requires concentration. The player needs to not only hold the beat, but also be capable of interpreting and reacting to the actions of enemies. The workload increases as the player progresses through the game, since enemies become more numerous and harder to defeat. The game provides appropriate challenge to the player through variable difficulty. As the game goes on, new challenges are steadily introduced, such as new types of enemies, that the player needs to figure out how to deal with. It is also crucial that the controls of the game are responsive since it is a rhythm-based game. In order to allow the player to feel a greater sense of control, a short grace period should be added before and after the beat. Rewards are given to the player when they defeat enemies in the form of treasure. The treasure can be spent while in port to purchase upgrades, new skills, and items.

Because of the fantasy-like setting of *SeaRhythm*, a narrative to introduce the game world to the player seems appropriate. After the player has been given context for

the world, they should be provided with a tutorial which could be structured as such: (1) The player is introduced to the beat and asked to move the ship a few times while holding the beat. (2) The player is introduced to the magic elements and are asked to create a few different spells and cast them in different ways. (3) An enemy ship appears and the player is given a brief explanation of how enemy ships work, before being instructed to destroy it. It would be preferential if the tutorial was narrated by an in-game character, rather than just being presented to the player through text boxes.

With many different things going on at the same time, it is important that the player receives feedback from the game. Visual and audio indicators can be used to give the player an overview of the following: when the player or enemy attacks or are hit; the beat; where the lanes are; ship hull integrity; amount of treasure; and progress through the level. Also, the player should easily be able to see what elements have been combined so far.

Figure 14.1 shows a screenshot from *SeaRhythm* close to the beginning of the game where the player has just launched a fireball. The two enemy pirate ships are clearly distinct from the player's ship and they move in opposite directions. The player's current gold (score) is in the bottom left corner, while the player's health is in the bottom right corner. Between those is the beat indicator which is a circle that fills in synchronization with the beat. This provides the player with a visual indicator of when to perform an action to the beat so that they do not need to rely solely on the music of the game.

14.2 Challenge, Fantasy, and Curiosity

At the beginning of *SeaRhythm*, the player is presented with the main goal of the game, which is to travel from port to port, defeating pirates on the way. The player is not certain to win, especially as they progress further and the game becomes harder. Smaller goals can be achieved in each level, such as keeping the beat for a certain amount of time.

SeaRhythm has a soundtrack fitting the game world playing while the player progresses through a level, which can provoke curiosity. Additionally, as mentioned in Section 14.1, there are many possibilities for audio and visual effects, such as the sound of an explosion hitting a ship or visual spells effects.

Because *SeaRhythm* is a rhythm-based game, the game music is important. The music should be appropriate to the fantasy of the game, which in this case would be something that would fit into a pirate movie. Another consideration is the speed of the music. For a benchmark, I have considered *Crypt of the NecroDancer*'s music where most of the song are in the range of 120 to 150 BPM (beats per minute) [40].

FIGURE 14.1: *SeaRhythm*

Based on this information I have used a song called Blackmoor Tides as the primary game music [1]. The song fits the fantasy of a battle at sea, with a heavy drum beat, chanting, and an appropriate rhythm. I edited the song slightly to reduce the BPM from 135 to 115 to make *SeaRhythm* more in line with the easier songs of *Crypt of the NecroDancer*.

14.3 Natural Mapping in Game Controllers

Since *SeaRhythm* is rhythm-based, and the Seaboard is not a traditional controller, I have used directional natural mapping for the controls, as described in Section 9.3. I have attempted to make it easier to understand the controls of the game by using the ribbons to change lanes, and by grouping similar actions together on the keywaves. The mapping of the Seaboard in Figure 14.2 is explained below.

[1] Blackmoor Tides is created by Matthew Pablo. Any music used in *SeaRhythm* is purely for educational purposes. *SeaRhythm* will not be published for commercial purposes or be made available to the public.

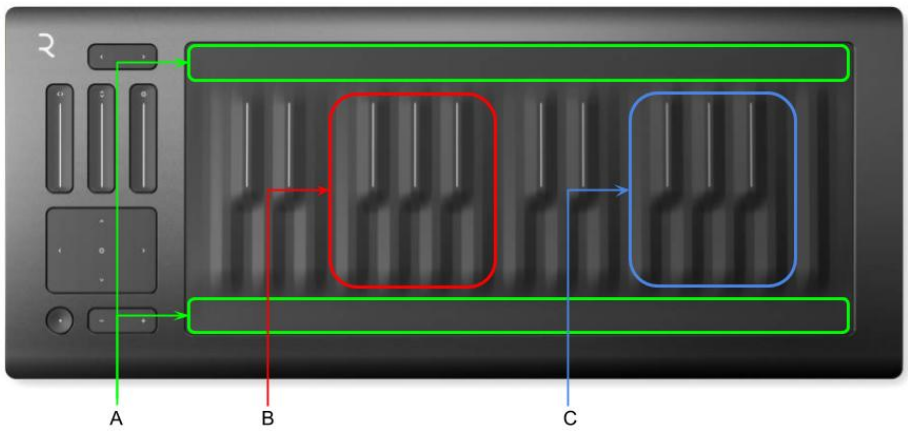


FIGURE 14.2: *SeaRhythm* controls on the Seaboard

A **Movement**: Swipe left or right on the ribbons to move to the lane to the left or right, respectively.

B **Attack modifiers**: From left to right (ignoring the upper keys): projectile, line, cone, area of effect.

C **Elements**: From left to right (ignoring the upper keys): fire, earth, water, air.

Instead of attempting to find a use for all 25 keys and both ribbons, I focused on using dimensions of touch and a smaller amount of keys. I did not want players to feel overwhelmed when play testing the game. Although there are many games with more complicated actions, the unfamiliarity with a piece of hardware such as the Seaboard must be factored in. Therefore I decided on using only the ribbons and two groups, where each group consists of keys that performs similar actions to the other keys of the same group. By using two groups slightly apart, players can rest one hand on each of the groups, requiring very little movement to reach any relevant key. This position also allows the player to easily operate the bottom ribbon using either thumb. Once the player has become comfortable with the basic controls of the game, later levels could introduce extra mechanics using the remaining currently unmapped keys, but that is outside the scope of this prototype.

As described in Chapter 13, the player needs to hit three keys in order to attack: one attack modifier key and two element keys. These can be hit in any succession to perform an attack, or even at the same time if the player does not want to wait at least three beats to attack. Hitting several keys at the same time allows more knowledgeable players to defeat enemies faster, thus earning more rewards. Players can also sacrifice speed in exchange for a stronger attack. By holding the attack keys down for an additional two beats, the attack will deal double damage. Conversely, players can sacrifice damage in order to make an attack faster. By sliding their finger

quickly from the bottom of a key towards the top of it, the attack will move twice as fast but inflict reduced damage. A fast attack consists only of one element and will always be a projectile. In other words, it requires only one key to make an attack compared to the regular combination of three keys. A fast attack cannot be a strong attack, or vice versa.

Each attack modifier changes the nature of the player's attack. A projectile has the longest range, but travels only in a straight line from the front of the player's ship and stops once it hits another object. A line attack also only travels in a straight line in the same manner as that of a projectile, except it pierces through other objects rather than stop once it hits them. However, the line attack does have shorter range than the projectile. A cone attack works the same as a line attack, but it also hits the lanes adjacent to the player and has shorter range. An area of attack effect is a circular attack centered on the player, hitting the lanes adjacent to them in addition to directly in front of and behind them.

Chapter 15

Technology

Since this project is centered on the Seaboard there are certain restrictions as to what technologies I can use. Because the Seaboard is not very widespread there is a limited amount of third party applications for it, and I have discovered none that have attempted to use the Seaboard as a game controller. On account of there being little technology available related to the Seaboard, I have based my choices in technology on what I could find.

15.1 ROLI Seaboard RISE 25

The Seaboard RISE 25 is the type of Seaboard used as a controller for this project. Other models include the larger RISE 49, and the GRAND. It is a digital keyboard with a pressure-sensitive surface created by ROLI, a music technology company. The Seaboard has been described in further detail in Chapters 8 and 10.

15.2 JUCE

JUCE is an open-source C++ API, with a large plug-in library and good audio functionality. It offers many different tools and classes that seem promising for the implementation of *SeaRhythm*. ROLI acquired JUCE in 2014 [41] and are the current developers for the platform. Given that they are the developers and that they have made software using JUCE, like the software synthesizer Equator [42], it seems like a good tool to incorporate into the project. JUCE has several tutorials on how to get started and its API provides easy access to many of the Seaboard's functions and will primarily be used to retrieve input events from the controller.

15.3 SFML 2.4.2

SFML 2.4.2 (Simple and Fast Multimedia Library) is an API written in C++ and can be used as a multimedia library to create games or multimedia software [43]. It has provided me with an easy way to handle the sort of simple 2D graphics that I will be using in *SeaRhythm*, in addition to an audio library and a game loop which will prove useful for handling the rhythm matching aspect of the game.

15.4 Microsoft Visual Studio 2017

Microsoft Visual Studio is an integrated development environment (IDE) used to develop software for Microsoft Windows. It features a high-level code editor, a debugger, and more. Visual Studio is compatible with C++ and is one of the recommended IDE's to use with JUCE. Additionally, SFML can be built inside it and it is a tool that I am familiar with. Using Visual Studio will likely be the most efficient for me, given both that it's a powerful IDE and my previous experience with it.

Chapter 16

Requirements

Throughout the development of SeaRhythm, I had a list of requirements that I was working to fulfill. Most of these requirements were identified prior to or during early development, while others were updated to reflect changes in the development plan. Tables 16.1 and 16.2 lists these requirements and briefly explain them. The requirements reflect only what I originally intended to implement for the prototype of the game. For a potential full release there would be many more requirements.

ID	Requirement description
FR1	The game screen shall show the player's ship, any active enemy ships, any projectiles, and the following head-up display elements: player ship health, current gold, and a music beat indicator
FR2	Game actions shall be updated to the beat of the game music
FR2.1	The game shall require the player to perform actions to the beat of the game music
FR2.2	The AI shall perform actions to the beat of the game music
FR3	The game shall allow the player to perform different attacks
FR3.1	The game shall allow the player to combine an attack modifier and two elements to perform a normal attack
FR3.2	The game shall allow the player to make a normal attack into a strong one
FR3.3	The game shall allow the player to make a fast attack
FR4	The game shall allow the player to move their ship from a lane to another existing, unoccupied, and adjacent lane
FR5	The game shall determine whether the AI performs an attack, or if it moves from a lane to another legal lane
FR6	If a ship is hit by an attack, its health shall be reduced
FR6.1	If an enemy ship's health is reduced to zero, it shall be removed from the game and the player's gold shall increase

ID	Requirement description
FR6.2	If the player's ship's health is reduced to zero, the game is over
FR7	When the game is over, the player's total accumulated gold shall be displayed
FR8	The game shall keep a running record of how many beats the player has hit in a row
FR9	The game shall use sound effects to indicate important events
FR10	After completing a level, the game shall allow the player to spend their gold to purchase upgrades for their ship
FR11	The game shall have multiple types of enemies

TABLE 16.1: Functional requirements

ID	Category	Requirement description
NFR1	Usability	It should not take longer than 5 minutes to learn the controls of the game
NFR2	Usability	The BPM of the game music should be appropriate
NFR3	Reliability	When playing the game for 15 minutes or less, the game should not crash

TABLE 16.2: Non-functional requirements

Chapter 17

Software Architecture

The software architecture of *SeaRhythm* is outlined within this chapter. An architectural overview of the game is shown in the form of a package diagram, while an overview of the game's classes is illustrated in the form of a class diagram.

17.1 Architecture Overview

SeaRhythm has been split into two primary layers, View and Modules, which are illustrated in the package diagram of Figure 17.1. The diagram provides an overview of the packages and how they communicate.

The *view layer* is in charge of what the player sees and hears by handling the game sound and sprites. The audio player reads and plays audio files while the game view loads the sprites that are to be shown on the screen. This is mostly handled by the API of SFML.

The *module layer* contains the game logic and the controller. The game logic handles all of the objects in the game and how they communicate, while the controller is used to detect player input.

17.2 Class Overview

Figure 17.2 is a class diagram of *SeaRhythm*. It shows the classes that have been developed and gives further insight into what the packages from Section 17.1 contain.

SoundPlayer and *MusicPlayer* are both built from SFML's audio library. *SoundPlayer* handles short sound events that occur, for instance, when two objects collide. These sounds are small enough that they can be kept buffered. The game music, which is

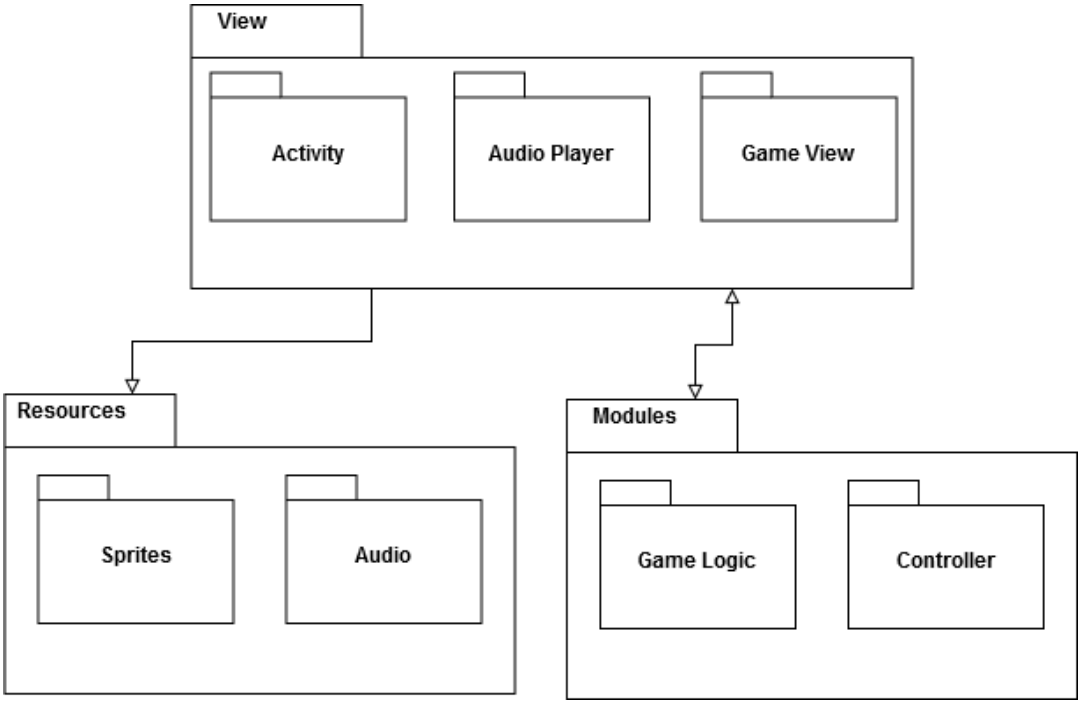


FIGURE 17.1: Package diagram

handled by *MusicPlayer*, is much larger and is instead streamed so the entire song does not have to be loaded into memory at once.

The classes in the *Game Logic* package encompasses the logic for updating the game. *SpriteObject*, *Player*, and *Enemy* contain the logic for player and enemy actions, while *EventListener* is used to listen for collisions between objects. *GameLoop* is the primary game loop.

ControllerEvent and *ControllerEventHandler* uses JUCE's API to detect input from the Seaboard and handle it. The events are sent to the *Game Logic* package to update the game according to the player's inputs.

MainGameView contains all of the visual elements of the game screen and is what the player sees while playing the game.*MenuView* is a simple screen shown before the game starts.

The most important class is *MainActivity*. It connects the whole application together by communicating with *MainGameView* and the classes in *Game Logic* in order to properly display the position and status of the sprites as the game goes on. It also connects *MusicPlayer* with *GameLoop* to keep the game music synchronized with the game loop.

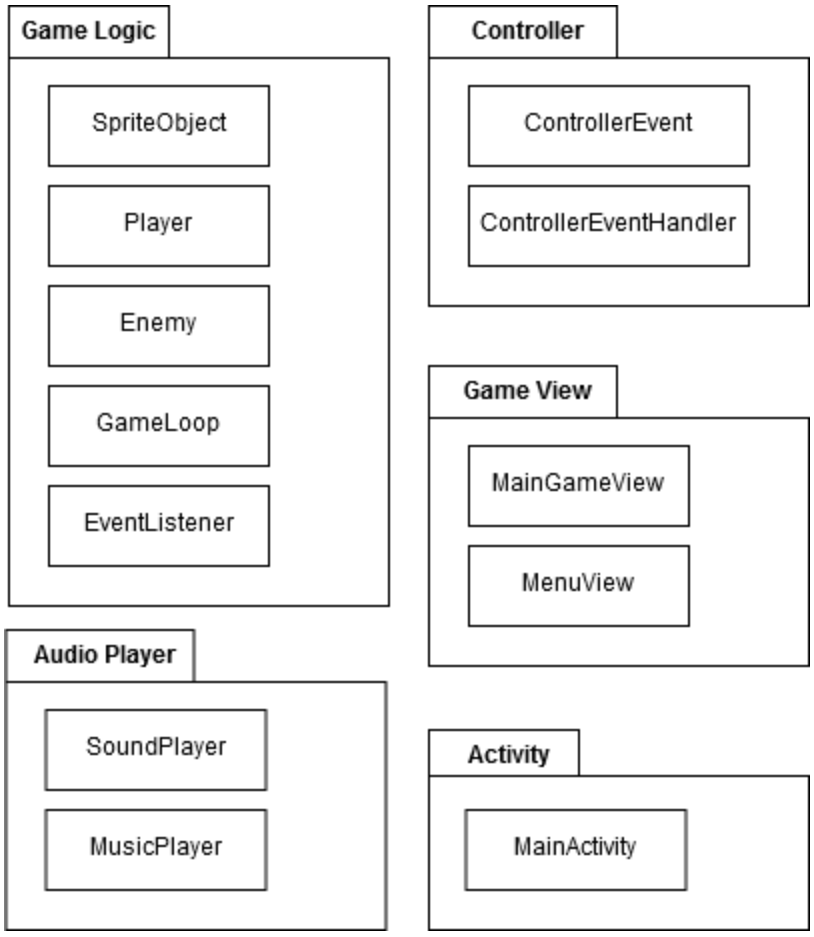


FIGURE 17.2: Class diagram

Chapter 18

Testing

This chapter describes the testing process during the development phase of *SeaRhythm*. I briefly present the testing approach before explaining the tests themselves and their criteria. Finally, an overview of the tests is included.

18.1 Testing Approach

During the implementation phase, I used the functional requirements from Chapter 16 as a form of milestone. They could be seen as a set of goals that could be designated as complete when the outcome of an action or event occurred as specified in the requirements. Therefore, I decided to use these as test criteria during the testing phase of the development.

The first course of action during the testing was to check whether or not the outcome of an event was as expected. However, just because a certain task can be completed in the intended way does not mean that it is bug free and cannot be broken. Therefore, the testing also involved attempting to break the game through misuse cases. This mostly included pressing a lot of keys on the Seaboard at the same time, since that is the only way the player can interact with the game, in an attempt to either crash the game or have it perform an unintended action. Only after passing the misuse cases to a satisfactory degree was a test deemed successful.

It was not possible to test the non-functional requirements during the development phase, except for NFR3 which was inherently a part of testing the functional requirements. NFR1 and NFR2 requires several players to test, and results can vary from user to user. These two non-functional requirements are discussed further in Part IV.

18.2 Test Results

This section describes the tests, test criteria, and validation of the requirements from Chapter 16. Since I am the sole person working on this project, all testing and quality assurance has been done by me. I used a spreadsheet to keep track of the status of the requirement validations which gave me an easy way to update the status of the tests and continue to the next step. Table 18.1 contains an overview of the tests and their outcomes.

ID	Description	Status	Comment
T1	The game screen shall show the player's ship, any active enemy ships, any projectiles, and the following head-up display elements: player ship health, current gold, and a music beat indicator	C	All elements show
T2	Game actions shall be updated to the beat of the game music	C	The game updates to the beat
T2.1	The game shall require the player to perform actions to the beat of the game music	C	Player input not during the beat is ignored
T2.2	The AI shall perform actions to the beat of the game music	C	Enemy ships are updated to the beat
T3	The game shall allow the player to perform different attacks	C	Three types of attack are available
T3.1	The game shall allow the player to combine an attack modifier and two elements to perform a normal attack	C	The player can make an attack by pressing an attack modifier key and a combination of any two element keys
T3.2	The game shall allow the player to make a normal attack into a strong one by holding down the keys for an additional two seconds	C	The attack becomes stronger by holding down the keys for a normal attack for an extra two beats
T3.3	The game shall allow the player to make a fast attack	C	The player can make a fast attack by swiping the key of an element from bottom to top during a single beat
T4	The game shall allow the player to move their ship from a lane to another existing, unoccupied, and adjacent lane	C	The player can perform legal moves by swiping left or right on the ribbons of the Seaboard

ID	Description	Status	Comment
T5	The game shall determine whether the AI performs an attack, or if it moves from a lane to another legal lane	C	The AI will semi-randomly attack or do a legal move
T6	If a ship is hit by an attack, its health shall be reduced	C	When a collision between an attack and a ship is detected, the ship's health goes decreases
T6.1	If an enemy ship's health is reduced to zero, it shall be removed from the game and the player's gold shall increase	C	The ship disappears from the screen and the player's gold increases by 10
T6.2	If the player's ship's health is reduced to zero, the game is over	C	The game stops and the player is informed that the game is over
T7	When the game is over, the player's total accumulated gold shall be displayed	C	The player is informed of how much gold they collected
T8	The game shall keep a running record of how many beats the player has hit in a row	NC	Not included since in the current state it would perform no service beyond showing the counter
T9	The game shall use sound effects to indicate important events	PC	Sound effects have been added to some events, like a projectile hitting a ship, while others have no sound at all
T10	After completing a level, the game shall allow the player to spend their gold to purchase upgrades for their ship	NC	Not implemented due to time constraints. It could be implemented in the future, but would require some modifications to ship and/or player class
T11	The game shall have multiple types of enemies	PC	Only one type of enemy has been added. Adding more would require some modifications to ship and/or enemy class

TABLE 18.1: Testing of the functional requirements. Legend: C (Complete), PC (Partially Complete), NC (Not Complete)

In the end, nearly all of the functional requirements were satisfied, but not all. Both T9 and T11 were partially completed, while T8 and T10 were not completed. T9 was not a very high priority and therefore only some of the more significant events play sound effects, such as a ship getting damaged. This was mostly due to the time it took to find certain sound effects. With many combinations of elements and other events, it was not worth finding unique sound effects for each and every one.

Like T9, T8 was not a high priority and was simply not implemented due to time constraints. Although a rather major part of the game, T10 had to be cut due to time constraints. It would have increased the complexity of the game and would require modifications to several classes and an interface the player could interact with. It was critical to have at least one type of enemy in the game, but multiple types of enemies were a low priority and T11 was then only partially completed.

Part IV
Results

This part details the execution of the user testing of *SeaRhythm* and its results. The results are a product of the user tests, the observations, and the survey. The results are simply presented here, They will be discussed in the next part of the report.

Chapter 19

Execution

This chapter contains a detailed description of how I planned the user testing of *SeaRhythm*. The process began with an information brief, followed by a user test while I observed, and ended with the player answering a survey after having played the game.

19.1 Information Brief

Before beginning the user test, each participant were given a short information brief on how the game works and how they play it. The brief was intended to work as a tutorial for first-time players, since there is no current in-game tutorial. First, the participants were introduced to the Seaboard and given a short description of how it works. Next, they were showed which keys are used to play *SeaRhythm* and what each of the keys do. This stage was over in the span of approximately two minutes.

19.2 User Test and Observation

The next step was for the participant to begin playing the game. While the participants were playing, they were being recorded, which they had been informed of beforehand. I was also acting as an observer throughout the user test. Each user test lasted about 15 minutes.

The participants were encouraged to speak their thoughts out loud while they were playing. Their thought processes might highlight sources of confusion or frustration that have been overlooked during the design process, and which should be improved upon. Also, if after playing the game, a participant happened to forget a thought that occurred to them while playing the game, I will still have a record of

them saying it. No specific actions were asked of the participants regarding gameplay, besides trying their best not to be beaten.

During the observation, I was studying how the participant interacted with the game. This included whether or not they had difficulty learning the controls, and their emotional state while playing. By looking for different reactions like joy or frustration, I can combine the results of the observation to more accurately gauge the level of game enjoyment.

19.3 Survey

After completing the user test, the participants needed to fill out a survey. They were asked to add a short description or explanation to each of their answers. The survey consisted of 14 questions, not including a field for any additional comments, and was designed so that the participants would answer in a qualitative manner. Most of the questions relate to game enjoyment, but they also aim to detect flaws in the design of the game. Table 19.1 lists the questions of the survey.

ID	Question
Q1	How was your experience playing the game with the Seaboard as a controller?
Q2	Did you experience the mapping of the controller as intuitive or natural?
Q3	What did you like best about the controller?
Q4	What did you like least about the controller?
Q5	Do you think this controller is innovative?
Q6	Do you think the game would be harder or easier to play using a traditional controller?
Q7	Would your experience of the game be better or worse if you had used a traditional controller?
Q8	What other kinds of games could benefit from this controller?
Q9	Did you enjoy the game?
Q10	Did the game challenge you? In what way did you feel challenged, if any?
Q11	Did you feel that you improved at the game as you played?
Q12	Was the objective of the game clear to you?
Q13	Was it clear that you were making progress in the game?
Q14	Did the game respond to your input as expected?

TABLE 19.1: Survey

Questions Q1-Q8 attempt to discern the player's opinion of the Seaboard as a controller and how its use affected their enjoyment of the game. Q1, Q6, and Q7 all pertain to how the player experienced the combination of controller and game. The latter two, in addition to Q8, also compares the controller to other controllers and types of games. Questions Q2-Q5 and Q9-Q14 are directly related to the criteria for game enjoyment that were presented in Chapter 9 about challenge, skill development, goals, control, and mapping in game controllers.

Chapter 20

Results

This chapter presents the results that were obtained from the observation, user test, and survey detailed in Chapter 19. First, a brief introduction is given about the participants before moving on to the results from the observation and survey.

20.1 Participants

14 students from NTNU volunteered to participate in the user testing of *SeaRhythm*. All of the participants completed the testing and answered the survey. Age and gender of the participants were not recorded, but with such a small sample size it is unlikely that any meaningful conclusions could have been drawn from that information. For the purposes of familiarity bias, it should be stated that I had no previous relation to any of the participants.

20.2 Results from Observation

It was immediately apparent that most participants had some difficulty in getting used to the controls of the game. Several commented that the Seaboard was unfamiliar, or that they hadn't used a game controller similar to this before. After a few attempts however, all of the players managed to move the ship and perform some attacks more or less reliably. Six of the players also stated that they liked the feel and texture of the Seaboard.

Of the three types of attack (normal, fast, and strong), the strong attack was the least used with only two participants using it somewhat regularly. The fast attack was quite popular, with ten of the participants using it quite often. One of them even went so far as to use it nearly exclusively, and with good results. Everybody used the normal attack.

Comparing the four attack modifiers (projectile, line, cone, and area of effect), the projectile was the most used. The cone attack was used often too since it could hit both adjacent lanes and the lane of the player and had a semi-decent range. No one used the area of effect with any regularity, and the line was used just slightly more often. The area of effect had a very limited use as enemy ships were rarely close enough to be hit by it, and the line attack was only superior to the projectile if there were two enemy ships in the same lane which rarely happened.

Interestingly, two players attempted to move their ship to an adjacent lane while there was an enemy ship in the way in order to ram into it, but the game did not consider this a legal move.

Although it seemed like no one outright disliked the game, there were also none that exhibited any sign of experiencing total flow as described in Chapter 9. There were some signs that there were elements of game enjoyment present. A few players stopped talking and tensed up in concentration when there were many enemies on the screen, and looked happy when they managed to survive. At one point or another while playing, most of the players were bobbing their heads along to the music, indicating that they were taking part in the fantasy of the game. Half of the players at some point during the gameplay commented that there were no enemies on the screen and nothing to do.

20.3 Results from Survey

Since the survey was of a qualitative nature the full results are not easily quantifiable. Although the individual reasons differ, some parallels, like how many participants enjoyed the game, can be drawn. Where applicable, the results have been quantified and recorded in Table 20.1. The questions of the survey are listed in chronological order below, along with a summary of the answers and interesting comments.

Q1 - *How was your experience playing the game with the Seaboard as a controller?*
The general consensus was that the participants had a good time playing the game. They liked trying out the Seaboard and thought it was an interesting piece of technology. There are improvements that could be made, but the players were aware that this was merely a simple prototype, and not a complete game.

Q2 - *Did you experience the mapping of the controller as intuitive or natural?*
Over half the players did not have much of an opinion here. Those who experienced the controller as not intuitive said that they would have had a difficult time figuring out the controls without help. The one player who responded that the controls felt intuitive wrote that he understood the thinking behind the controls after he had played for a bit.

Q3 - *What did you like best about the controller?*
The most common answer was that it was something new that the players had not tried before. One player responded that it was incredibly satisfying to use the fast attack as it felt as if he was flinging attacks at his enemies.

Q4 - *What did you like least about the controller?*
Several participants responded that there were so many keys that were not used or that the control was too large compared to how many actions the player could perform. Two others stated that the strong attack was clunky to use.

Q5 - *Do you think this controller is innovative?*
The players were unanimously agreed that the Seaboard was used in an innovative way. It was unexpected to play a game using such a control scheme.

Q6 - *Do you think the game would be harder or easier to play using a traditional controller?*
All of the players believed that the game would be easier to play with a traditional controller, in most part due to familiarity. One player did add "It would be easier with a normal controller, but doesn't that defeat the purpose?".

Q7 - *Would your experience of the game be better or worse if you had used a traditional controller?*
The responses to this question were quite mixed. Some stated that the experience would probably be worse because the controller was the most fun part of the game to them. Others responded that they would feel more in control of the game if they had access to a traditional controller.

Q8 - *What other kinds of games could benefit from this controller?*
This was the only question that some of the players left blank. One suggestion was to make a fighting game where you controlled the character through the touch surface of the Seaboard.

Q9 - *Did you enjoy the game?*
11 of the players responded that they enjoyed their time with the game. A recurring sentiment was that it was fun to try something new. Of the two that did not enjoy the game, one said that there was currently too little content for the game to be fun for them. Similarly, one of the players who said they enjoyed the game declared that they would not want to keep playing the game beyond the user test if it stays in its current state.

Q10 - *Did the game challenge you? In what way did you feel challenged, if any?*
Nearly all the participants stated that they felt challenged by the game. Some said they had to concentrate in order to not make a mistake.

Q11 - *Did you feel that you improved at the game as you played?*
All of the players thought they were better at the game at the end of the test than at the beginning. The majority thought that it was a bit hard at first, but they got the

hang of it after a few tries.

Q12 - *Was the objective of the game clear to you?*
Half of the players did not think that the objective was clear since it was never presented outside of the information brief. One player said "...it was clear that the other ships were enemies. They had pirate flags and a different color", but also said that he was uncertain at first if the goal of the game was to sink them.

Q13 - *Was it clear that you were making progress in the game?*
Most players did not get a sense that they were progressing. One player wrote "I know I should be progressing, but there is no indication that I am doing so", which seemed to be the general sentiment.

Q14 - *Did the game respond to your input as expected?*
Several players stated that they did not have any expectations of how the game would respond since both the game and controller were unfamiliar. Others said that as they learned the game, the behavior was consistent and therefore became expected.

ID	Question	Disagree	Neutral	Agree
Q2	Did you experience the mapping of the controller as intuitive or natural?	35.7%	57.1%	7.1%
Q5	Do you think this controller is innovative?	0%	0%	100%
Q10	Did you enjoy the game?	14.2%	7.1%	78.6%
Q11	Did the game challenge you?	0%	14.2%	85.7%
Q12	Did you feel that you improved at the game as you played?	0%	0%	100%
Q13	Was the objective of the game clear to you?	50%	21.4%	28.6%
Q14	Was it clear that you were making progress in the game?	64.3%	35.7%	7.1%
Q15	Did the game respond to your input as expected?	0%	57.1%	42.9%

TABLE 20.1: Survey results

At the end of the survey there was a text field where the players could leave a comment or a suggestion for improving the game. The suggestions included adding a shield "attack" that would protect the player's ship from enemies. Another was to use more of the screen by making the game side-scrolling, since computer screens are wider than they are tall. However, this would make the movement less intuitive as you would move up or down by swiping left or right on the ribbons. Currently you swipe left or right to move in the corresponding direction. Lastly, someone suggested that instead of always having to use an attack modifier, an attack could default to a projectile if a modifier is not selected. This would not only make attacking

easier, but also open up an easily accessible key binding.

Part V
Discussion

This part contains the discussion and evaluation of the project. First is an evaluation of the design, followed by an evaluation of the results, and ending with an evaluation of the project and its goal.

Chapter 21

Design Evaluation

SeaRhythm is the first game that I have designed and implemented from the ground up all by myself. When reviewing the controllers in Chapter 7, I realized there was little precedence for a controller like the Seaboard. Since it is relatively new, and not very widespread there has been little to no research or third-party applications regarding the Seaboard that I could use in this project. Consequently, it is likely that there are several design decisions that should have been done differently.

Some functionality that was originally intended to be part of the prototype unfortunately had to be left out due to lack of time. By adding a system allowing the player to upgrade their ship using gold would add a new level of complexity to the game. There are many different upgrades that would be possible to add, like new attacks, new ships, and improved weapons. Each upgrades could also provide a partial goal for players to work towards, adding to the fantasy and enjoyment of the game.

The two non-functional requirements from Chapter 16 that were not tested was NFR1 *It should not take longer than 5 minutes to learn the controls of the game* and NFR2 *Usability The BPM of the game music should be appropriate*. After having conducted the user tests I can say that NFR1 is satisfied. Although not all participants had mastered the controls after five minutes of gameplay, all of them were able to move the ship and perform attacks. NFR2 can also be considered satisfied as only one of the 14 players thought that the game was too fast.

Chapter 22

Evaluation of Results

This chapter discusses and analyzes the results from Part IV. The analysis is performed on the basis of the criteria for game enjoyment and flow from Chapter 9.

22.1 GameFlow, Challenge, Fantasy, and Curiosity

From the results it is clear that the majority of the test participants enjoyed playing *SeaRhythm*. 11 of the 14 players agreed when asked if they enjoyed the game. From Section 9.1 we know that a significant contribution of game enjoyment comes from flow, which was used to indicate how much enjoyment a player experienced. It is therefore safe to say that the players experienced some degree of flow while playing the game. This also correlates to my observations from Section 20.2 about how some players were concentrating hard while playing and with the fact that most of the players stated that they felt challenged by the game. It should also be noted that there were players who commented on downtime during gameplay. This means that the game did not always provide a high enough workload for the players, which results in a potential loss of enjoyment.

All of the players experienced an increase in their skill level while playing the game. This ties back to skill improvement in the GameFlow model and further contributes to game enjoyment.

A significant amount of the participants did not think the goal of the game was clear. From Section 9.2 we know that a simple game should have clear goals. In that respect, *SeaRhythm* has failed to provide the player with sufficient direction. This could be remedied by adding a simple message to the menu screen before the game begins. Players also found it difficult to gauge their progress in the game, in part due to there being no clear goal, but also because there was no on-screen element that showed the progress of the player.

There weren't many ways for the players to engage with the fantasy of the game.

Yet, during the user tests the players were observed bobbing their heads along to the music at some point during their playthrough. This indicates that the game music was a good choice for immersing the player further into the game.

22.2 Natural Mapping in Game Controllers

Not surprisingly, the participants had some difficulty getting used to the controls of the game due to them being unfamiliar. Familiarity is an advantage of computer keyboards and traditional game controllers because more and more people grow up using them from early childhood on a near-daily basis. Knowing that the players would be using a controller that was not familiar, I attempted to design the layout of the controls in such a way that it should be relatively easy for a new player to understand how it worked. Directional natural mapping was used where applicable, which was along the ribbons and for the fast attack. Considering that the players managed to learn the controls fairly quickly implies that I was successful in this regard, even though several players did not find the controls intuitive.

Additionally, many players liked using the Seaboard as a controller, though it seems some would prefer to use a smaller controller or a traditional controller. The Seaboard was brought up by a few players as being the most fun part of the game.

Chapter 23

Project Evaluation

This chapter will present the degree to which the research questions and research goal from Chapter 3 have been fulfilled based on the results from Part IV.

23.1 Fulfillment of Research Questions

After completing the specialization project, it was clear that the framework I should use to evaluate the Seaboard and *SeaRhythm* would be the criteria for flow and game enjoyment. The research questions were formulated so that I had a few clear objectives to work towards during the development, and later the user tests.

23.1.1 RQ1 - In which way did the use of the Seaboard as a controller influence the player's opinion of the game?

The objective of this research question is to determine a way for the Seaboard to positively affect the player's game enjoyment. Based on the research of Skalski *et al.* on the topic of natural mapping in game controllers presented in Section 9.3, I had some guidelines I could follow when deciding the layout of the controls for the game. During the concept design phase I determined that the most natural way I would be able to incorporate the Seaboard as a game controller would be through directional natural mapping.

The user tests and the survey revealed that the players were for the most part enjoying the game and the Seaboard. What I did not really consider as a factor for enjoyment, was that the players would like the Seaboard because it was new and different. In consideration of the research question, it can be considered that the Seaboard influenced the players' enjoyment of the game in a positive manner.

23.1.2 RQ2 - To which degree does *SeaRhythm* fulfill the criteria of game enjoyment?

It was important that *SeaRhythm* was an enjoyable game in order to evaluate the Seaboard's influence on enjoyment. Had the game not been enjoyable at all, it would be hard to conduct a proper evaluation of the Seaboard as a controller. Therefore I did my best to follow the frameworks for game enjoyment. As has already been discussed in Chapters 21 and 22, I was not entirely successful in this venture. However, the players found some enjoyment in the game meaning that *SeaRhythm* fulfills the criteria of game enjoyment to some degree, but not enough for players to experience proper flow.

23.1.3 RQ3 - Does the players' experiences of *SeaRhythm* support the design choices that were made to increase game enjoyment?

While some degree of game enjoyment was experienced by the players, the design of *SeaRhythm* still has plenty of room for improvement. Based on the answers and suggestions from the survey, I have realized that simple but important elements like clear goals are missing. This is an oversight on my part, and the game would need a minor overhaul to add these components that are essential for game enjoyment. In summary, the players experienced some game enjoyment, but the design was missing certain elements which, if present, would likely increase game enjoyment even more.

23.2 Fulfillment of Research Goal

The research goal of this project was to *develop a game with the Seaboard as the controller and investigate how the Seaboard affects game enjoyment*. From the goal I created the three research questions which were discussed above. From the results and discussion we can confirm that the game provides game enjoyment and that the Seaboard has a positive effect. The research questions are deemed fulfilled, to a greater or lesser extent, ergo the research goal is also deemed fulfilled.

Part VI
Conclusion and Future Work

This part contains my conclusion of the project based on the discussion from Part V. The report is wrapped up with a summary of work that should be done in the future to improve *SeaRhythm*.

Chapter 24

Conclusion

The purpose of this project was to create an enjoyable rhythm-based computer game which uses the Seaboard as a game controller in an innovative way. Many of the design choices are based upon a specialization project during which I studied the evolution of game controllers, and frameworks for game enjoyment and flow. This was done in order to gain a better understanding of what sort of design choices and features defines a fun and enjoyable game, and how the game controller might affect it.

Next, in order to design a concept fit for the Seaboard, I studied its advantages and limitations, which provided valuable knowledge in relation to the game concepts. Based on what I discovered about the Seaboard, I describe a few games that I used as inspiration and that had several game mechanics that I believed would suit the Seaboard. These mechanics turned into promising game concepts, which ultimately were combined to form the final game concept of *SeaRhythm*.

Lastly, this project includes the design, development, and testing of the prototype. The results from the testing are analyzed and discussed, and will indicate how enjoyable players experienced the game to be.

SeaRhythm illustrates that unconventional game controllers can be used in new and entertaining ways which was part of the objective for *research question 1*. The test results showed that the use of the Seaboard affected players' experiences positively.

Research question 2 asked how *SeaRhythm* fulfilled the criteria for game enjoyment. Although the game does not provide a proper flow experience, the results still support that it offers enjoyment to its players.

Research question 3 asked if the players' experience of the game correlate to the game's elements of achieving game enjoyment. In the end, the design was flawed, but there were still enough elements providing game enjoyment for the players to have a positive experience.

The project's goal was to develop and subsequently investigate the effect the Seaboard had on game enjoyment when it was used as the game controller. The experiences of the players who tested the game were recorded and examined. The conclusion of the project is that the Seaboard has had a positive effect on the players' experiences of *SeaRhythm*.

Chapter 25

Future Work

This final chapter contains suggestions for future work to be done on *SeaRhythm*. Based on the results of the user tests and the survey, the next step would likely be to go back and redesign parts of the game to better incorporate elements that affect game enjoyment. There are many features and improvements which can be added to the game, including the features that were cut from the implementation, the following of which are in the planning stage:

- Clear goals
- A way to measure the player's progress through a level
- Fix the game logic controlling the spawn rate of enemies so that there is less downtime
- More enemy types
- More levels
- More attacks
- Add a way for players to shield themselves from attacks
- Allow the player to select which attacks they have at their disposal
- Improved sound effects
- Improved graphics
- A tutorial
- A store where the player can spend gold to upgrade their ship
- Customizable ships
- Additional soundtracks

- Look into alternative control layouts

This list is quite comprehensive and it would take a lot of time to implement all of it. If the goal is to prepare the game for a commercial release, all of this would likely have to be implemented.

Because the testing was done on a prototype game, further testing should be conducted once the game has been extended. With the addition of an upgrade system, some improvements to the audio and graphics, and a few extra levels, a demo could be released to gauge interest for the game.

Bibliography

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

[2] Wikipedia. *Game controller*. https://en.wikipedia.org/wiki/Game_controller. Accessed 10-06-2017.

[3] Alastair H Cummings. "The evolution of game controllers and control schemes and their effect on their games". In: *The 17th Annual University of Southampton Multimedia Systems Conference*. Vol. 21. 2007.

[4] Dan Ackerman. *PC gaming beat consoles a long time ago, so now consoles are becoming more like gaming PCs*. CNET. https://www.cnet.com/news/pc-gaming-beat-consoles-a-long-time-ago-so-now-consoles-are-becoming-more-like-gaming-pcs/. Accessed 10-06-2017.

[5] Dan Hsu. *The Xbox One controller: Projectors, smells, and other stuff that didn't make it in*. VentureBeat. https://venturebeat.com/2013/11/18/the-xbox-one-controller-projectors-smells-and-other-stuff-that-didnt-make-it-in-part-1-exclusive/. Accessed 10-06-2017.

[6] Statista. *Value of the video game market in the United States from 2011 to 2020*. https://www.statista.com/statistics/246892/value-of-the-video-game-market-in-the-us/. Accessed 03-06-2017.

[7] Statista. *Total and segment revenue of the U.S. video game industry from March 2015 to March 2017*. https://www.statista.com/statistics/201073/revenue-of-the-us-video-game-industry-by-segment/. Accessed 03-06-2017.

[8] Mark Overmars. "A brief history of computer games". In: *Department of Information and Computing Sciences, Faculty of Science, Utrecht University* (2012).

[9] Steven L. Kent. *The Ultimate History of Video Games. Ch. 4 - And Then There Was Pong*. Three Rivers Press, 2010.

[10] Wikipedia. *DualShock*. https://en.wikipedia.org/wiki/DualShock. Accessed 10-05-2017.

[11] Wikipedia. *Nintendo DS*. https://en.wikipedia.org/wiki/Nintendo_DS. Accessed 12-06-2017.

[12] Wikipedia. *Wii U*. https://en.wikipedia.org/wiki/Wii_U. Accessed 12-06-2017.

[13] Wikipedia. *Nintendo Switch*. https://en.wikipedia.org/wiki/Nintendo_Switch. Accessed 17-06-2017.

[14] Statista. *Number of available apps in the iTunes App Store from 2008 to 2016*. https://www.statista.com/statistics/268251/number-of-apps-in-the-itunes-app-store-since-2008/. Accessed 17-06-2017.

[15] Imagine Media. "The Ultra 64 Joypad". In: *Next Generation* 14 (1996), pp. 38–39.

[16] Wikipedia. *Nintendo 64 controller*. https://en.wikipedia.org/wiki/Nintendo_64_controller. Accessed 10-05-2017.

[17] Statista. *Global unit sales of current generation video game consoles from 2008 to 2016*. https://www.statista.com/statistics/276768/global-unit-sales-of-video-game-consoles/. Accessed 10-06-2017.

[18] Wikipedia. *Xbox Controller*. https://en.wikipedia.org/wiki/Xbox_Controller. Accessed 19-06-2017.

[19] Emily Gera. *Xbox One controller was the result of 200 prototypes and 20 studies*. Polygon. https://www.polygon.com/2013/6/7/4405488/xbox-one-controller-was-the-result-of-200-prototypes-and-20-studies. Accessed 12-02-2018.

[20] Wikipedia. *Steam Controller*. https://en.wikipedia.org/wiki/Steam_Controller. Accessed 13-06-2017.

[21] Sherif Saed. *The Steam Controller has sold nearly 1 million units*. VG247. https://www.vg247.com/2016/10/13/the-steam-controller-has-sold-nearly-1-million-units/. Accessed 16-06-2017.

[22] Dan Isacsson. *A half million Steam Controllers in seven months is pretty terrible*. Digital Trends. https://www.digitaltrends.com/computing/steam-controller-machine-sales-2016/. Accessed 16-06-2017.

[23] Wikipedia. *Wii Remote*. https://en.wikipedia.org/wiki/Wii_Remote. Accessed 13-06-2017.

[24] Shabana Arif. *Sales of the PlayStation Camera and Move controllers are through the roof*. VG247. https://www.vg247.com/2016/03/18/sales-of-the-playstation-camera-and-move-controllers-are-through-the-roof/. Accessed 11-06-2017.

[25] Jon Porter. *Meet the minds behind Nintendo Switch's HD Rumble tech*. Techradar. http://www.techradar.com/news/meet-the-minds-behind-nintendo-switchs-hd-rumble-tech. Accessed 18-06-2017.

[26] Wikipedia. *Guitar hero*. https://en.wikipedia.org/wiki/Guitar_Hero. Accessed 19-06-2017.

[27] Jean-Baptiste Thiebaut, Sean Soraghan, and Lauren Ianni. "MULTI-DIMENSIONAL CONTROLLERS, EXPRESSIVITY, SOFTWARE: AN INTEGRATION PROBLEM". In: *Journées d'Informatique Musicale* (2013).

[28] ROLI. *Seaboard RISE 25*. https://eu.store.roli.com/products/seaboard-rise. Accessed 23-05-2017.

[29] James Vincent. *Feeling the music with Roli's squishy, pressure-sensitive keyboard*. The Verge. https://www.theverge.com/2015/9/23/9372961/roli-seaboard-rise-grand-hands-on. Accessed 09-05-2017.

[30] ROLI. *Seaboard RISE Creator Manual*. Version 1.1.2. Feb. 2016.
[31] Penelope Sweetser and Peta Wyeth. "GameFlow: a model for evaluating player enjoyment in games". In: *Computers in Entertainment (CIE)* 3.3 (2005).
[32] Thomas W Malone. "What makes things fun to learn? Heuristics for designing instructional computer games". In: *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*. ACM. 1980, pp. 162–169.
[33] Paul Skalski et al. "Mapping the road to fun: Natural video game controllers, presence, and game enjoyment". In: *New Media & Society* 13.2 (2011), pp. 224–242.
[34] Wikipedia. *Kinect Star Wars*. `https://en.wikipedia.org/wiki/Kinect_Star_Wars`. Accessed 13-06-2017.
[35] Robert Farrow and Ioanna Iacovides. "Gaming and the limits of digital embodiment". In: *Philosophy & Technology* 27.2 (2014), pp. 221–233.
[36] Anthony J Smith and Joanna J Bryson. "A logical approach to building dungeons: Answer set programming for hierarchical procedural content generation in roguelike games". In: *Proceedings of the 50th Anniversary Convention of the AISB*. 2014.
[37] Wikipedia. *Crypt of the NecroDancer*. `https://en.wikipedia.org/wiki/Crypt_of_the_NecroDancer`. Accessed 18-06-2017.
[38] Wikipedia. *Magicka*. `https://en.wikipedia.org/wiki/Magicka`. Accessed 19-06-2017.
[39] Wikipedia. *Space Invaders*. `https://en.wikipedia.org/wiki/Space_Invaders`. Accessed 18-06-2017.
[40] Crypt of the NecroDancer Wiki. *Music*. `http://crypt-of-the-necrodancer.wikia.com/wiki/Music`.
[41] JUCE. *History and Development*. `https://juce.com/history-and-development`. Accessed 01-02-2018.
[42] Ted Wood. *Building Equator with JUCE*. `https://www.juce.com/case-studies/equator`. Accessed 23-05-2017.
[43] SFML. *SFML FAQ*. `https://www.sfml-dev.org/faq.php`. Accessed 03-01-2018.
[44] Tore Bjørlo. "Music Game with the ROLI Seaboard - Innovation in Game Controllers". Technical report. Norwegian University of Science and Technology, Department of Computer Science, 2017.