UNIVERSITY OF BERGEN

Master thesis in Information Science

A Comparative Analysis of the Relationship between Gaming Controllers and Game Mechanics

Lene Nævdal

2013

Acknowledgements

My advisor, Dr. Viktor Kaptelinin, for his support and guidance throughout the project

My fellow students and lecturers at the Department of Information Science and Media Studies

The researchers at the Exertion Games Lab at RMIT University

My friends and family, for all their help and support

Vibeke Valkner, for her advice and discussions

My better half, Amilcar dos Santos, for all his encouragements and patience

Contents

Figures

Images

Tables

Abstract

In this thesis I have presented the issue of not exploiting the possibilities different technologies offer us in video games. The research questions were "how can input-technologies affect the enjoyment of video games?", "do different input-technologies work better for different types of game mechanics?" and "how can game developers best utilize input-technologies in video games?" In order to answer this I have conducted two case studies using case study research; a multiple-case study and a single-case study. Using a theoretical framework consisting of theories from HCI and game studies, I analyzed four video games in the multiple-case study and used the findings to propose a conceptual model called TechMech. The model was then used in the development of a prototype game, where the development process was used in a single-case study. While I have not found any definitive answers as to how developers can best utilize different input-technologies in their games, my findings suggest that games that support a high degree of p-actions, natural mapping and GameFlow elements were more enjoyable, while games that had a lower degree of p-actions and natural mapping, and which did not follow the criteria for GameFlow were less enjoyable. I present a conceptual model and some practices that I believe can be helpful for developers and researchers in the field. However further research is required; both on the topic in general, and in order to validate the TechMech model.

1 Introduction

You probably know someone who plays video games on a regular basis, or perhaps you do so yourself. Today most people have a gaming device readily available in their pocket through their smart phones. However video games as we know them were first introduced to the general public in the 1970's – 1980's, and the first game was developed already in the late 1940's. But this was only accessible to the lucky few who were either studying or working for major universities or companies that could afford computers. With the development in technology since then, gaming devices are now common property, and the devices are getting even more advanced technologically.

Up until 2006 the technology that was used for playing video games had basically stayed the same, with only minor differences such as a variety of button layouts. But in 2006 came the release of the gaming console Nintendo Wii, which used motion sensing in the controller. With the release of the Wiimote, the players could now use physical movement as a way of controlling the game, and soon Sony followed up with the PlayStation Move. At the same time Microsoft came up with the Kinect camera, which took us into the realm of motion sensing without the use of a physical controller. Today we also use our mobiles and tablets to play games using a touch interface. Since then more new and exciting technologies have continued to surface. Some examples of this are the Wii U, the Emotiv EPOC headset, Oculus Rift, Leap Motion etc. How has game developers responded to all this new and exciting technology? Has there been a revolution in how we play video games as many have foretold? The short answer is; no.

McCallum (2013) talked a bit about new technology in video games and on issues in the way we utilize them in his conference talk "Game Technology of the Future".

> *"Most of the reviews on the Wii U say: "there's nothing in the games that require that second screen.". … [This is] because the designers are doing what they did to the wiimote. When the wiimote came out, they treated the movement as a button. … [But] if you don't understand the nature of the new technology, and you just kind of manually make it fit into your current model, then you won't use the value of the new technology."*

The issues that McCallum mentions in the quote above are the same issues that I want to address in this thesis. There are several games that bring us new and enjoyable playing experiences, but with others we are baffled by their choice to put some new and fancy technology into their games without giving it any proper thought. Or as McCallum (2013) puts it, games where it looks like the developers just wanted to attach the Kinect sticker on the box.

Due to the increasing popularity with motion-sensing devices in gaming, researchers have started to take an interest into the technology and how it relates to the player's enjoyment. And when even more advanced technology starts being used by the general public, it is important that we understand how it can be used in an entertaining and enjoyable way. My hope is that the findings from this thesis can help enlighten the subject.

In this thesis I will try to address this problem by proposing a conceptual model that game developers can use when developing new games where they are trying to figure out what technology to use, and/or how to use it. In order to do this I conduct a multiple-case study where I look at four different video games that utilize different technologies. I analyze why the control schema[1] worked or didn't work using existing games and human-computer interaction research. Based on the theoretical framework and the results of the case study I will try to divide different types of input technology into categories of technology, and do the same thing with different types of game mechanics. I will then do a comparative analysis of the input-technologies and the game mechanics which will result in a conceptual model that can be used in future development

[1] A control schema is the setup of gaming controllers and their corresponding game mechanics

process or game analysis. In the end I conduct a single-case study, where the model is used in an actual development process of a prototype game that utilizes several input-technologies.

2 Research questions

In the section below I have outlined a tentative hypothesis that I will use as the base of my research questions and thesis:

“The technology used for a game should contribute to support the gaming experience. I propose that different input-technologies work better for different types of game mechanics, and by taking this into account one can utilize new technology in a way that will enhance the gaming experience for the players.”

To present what potential I believe there could be in using games developed specifically for the use of different types of input-technology, I have written a scenario, such as is used in interaction design (Rogers et al., 2011), which shows a possible gaming situation. This scenario gives an idea of how a game using multiple input-technologies could be, and helps to contextualize the topic I want to address in my research questions.

Jack is sitting in the living room playing his newly acquired game ‘Dragons Tale’, a game which has received much attention in gaming reviews, because of its use of the new EEG-headset that came with the new Xbox console from Microsoft. But what has been the cause of this attention is not only the headset, but also how it cooperates with the standard Xbox controller, and the Kinect camera, giving the game a whole new dimension. Jack has been playing his game for a couple of days now, and is starting to get the hang of it. He is not playing alone, but with some friends from school who have joined his game over Xbox-Live. They have just achieved the title as knight, after learning to use their powers at the Academy, and they are now doing one of their first quests. After just finishing killing some trolls, they are now getting ready for the boss. Jack is playing the role of the party’s healer. While they are waiting for the boss to appear, Ben is telling a story from school, while Jack asks Lisa if she can enhance his armor with a spell. The microphone in the Kinect camera makes it easy to communicate with each other.

When the boss appears, Jack fights his natural urge to concentrate, in order to embrace “the source”. If he fails to embrace it, he cannot perform his spells, and then he cannot do his part as a healer. This is a technique he has practiced from his time in the academy. Jack is pressing the buttons on his Xbox controller in order to heal his team members, but suddenly the boss goes after Jack’s character, instead of the tank. Since he cannot endure many hits, Jack hurries and throws his hands up in the air, a move which is caught by the Kinect camera, and causes a shield to close around Jacks character. Unfortunately the shock of being attacked has made Jack lose “the source”, and he has to relax soon to be able to heal the other players, who are now rapidly losing health.

The research questions I have formulated are based on the tentative hypothesis described above.

Research Questions:

1. **How can input-technologies affect the enjoyment of video games?**
2. **Do different input-technologies work better for different types of game mechanics?**
3. **How can game developers best utilize input-technologies in video games?**

In order for me to answer these questions, I will only be focusing on the relationship between input-technologies and game mechanics. I acknowledge that the use of input-technologies is not necessarily decisive for the success of a video game, and that there are several just as important aspects in games that are crucial to the success of video games. However, I believe that taking a closer look at smaller portion of the issue will in turn help us to look at the big picture, which is why I am focusing as narrowly in this thesis as I am.

I will in the end of the thesis propose a conceptual model, which is a comparative analysis between different types of game mechanics and input-technologies. Along with the conceptual model I also propose some practices that will hopefully be of help to developers as well.

From a human-computer interaction perspective, I am interested in theories of distributed cognition, embodied cognition and embodiment theories, which I believe can support my tentative hypothesis. These theories will be presented later in the thesis, along with other models and relevant research, such as the GameFlow model, Natural Mapping and systems development. This research is important as it gives the foundation this thesis is built on.

2.1 Motivation

What motivated me to research this particular topic were at first my own observations. When Kinect and PlayStation Move were released, I bought both of them, even though I already had the Nintendo Wii, because I was imagining so many possibilities on how the technology could be in different games. Unfortunately I also quickly realized that there weren't many games that seemed to really exploit the technology's possibilities. Of course there were some games that I really enjoyed playing, such as *Fruit Ninja Kinect*[2], *Kinect Adventures*[3] and PlayStations *Sports*

[2] STUDIOS, H. 2011b. Fruit Ninja Kinect. Xbox 360.

[3] STUDIO, G. S. 2010a. Kinect Adventures! Xbox 360: Microsoft Game Studios.

Champions[4]. And this made me start thinking about why those games were so much more enjoyable than for instance *Steel Battalion: Heavy Armor (Software, 2012)*. Journalists and game critics show that I am not the only person who is excited about the technology, but is more skeptical about the games developed for it.

> *"The Kinect as hardware is great, but there's plenty of room for software engineers and UI designers to improve."* (Miller, 2010a).
>
> *"(…) the success is going to be largely dependent upon the games that support it."* (Miller, 2010b).

There will always be some games that are successful, while others are not, and games which are developed for new technology are no different. However those games could almost be seen as pioneers, trying to figure out how to make the most out of the player's experience with these new controllers. Hopefully the conceptual model I will develop in this thesis can help future developers with figuring it out, and other researchers in studying these games.

2.2 Contributions and goals

My goals for this thesis is to propose a conceptual model which can be used by developers and researchers in order to better utilize different types of technology in gaming, in a way that enhances the gaming experience in new and exciting ways. Hopefully the model can also be of help in other areas in human-computer interaction and games studies as well, where new technology is continually being incorporated into our everyday devices.

2.3 Research process

In order to try and answer the research questions, I will be using a research process that can be divided into three main parts. The first part is to study existing video games and use the findings from that study together with the theoretical framework. The next part contains the main contribution, and is where I propose a conceptual model, based on the data found in the first part and the theoretical framework. For the last part, I try to bring the conceptual model into an environment similar to a real-world development process, by conducting an application description case study.

Hopefully this research process will let me 1) gather information, 2) use that information to develop a conceptual model, and 3) test my model. In the end the thesis will contain findings from this entire process.

---

[4] STUDIO, S. D. 2010c. Sports Champions. PlayStation 3: Sony Computer Entertainment.

3 Literature review

Research literature

In this section I will first give an account of the development of gaming controllers up until today. After that I present literature and theories that are relevant for my research questions. I will talk about human-computer interaction, where I specifically look at theories of embodiment, distributed cognition, pragmatist aesthetics and brain-computer interface (BCI). I will then present the field of games studies, where I will focus on embodiment in video games, natural mapping, the GameFlow model, game mechanics and control devices. The above theories and models are important for the multiple-case study and for the development of the conceptual model. Another field that is of interest in this thesis is systems development, where I will focus on agile methodology and game development, which is important for the single-case study. I will also look at some existing research that is similar to my own, and discuss them in relation to this thesis. In the end of this section I will present the theoretical framework used for my research.

3.1 History of gaming controllers

In the following sections I will briefly go through the history of gaming controllers (1up, 2005). There are some controllers I will not mention here, because they never became standards. I will also briefly present some of the new technologies that are expected to be used in future gaming, as well as briefly introduce a game development engine that is affecting the scale of game development for different input-technologies.

The first controller was made for the first video game, namely Spacewar, where they used four buttons to control thrust, rotation and firing. The next controller was made for Pong and consisted of a joystick. This controller became very popular, and was used by several manufacturers of gaming hardware. In 1982 Nintendo invented what is now known as the D-pad (directional pad), which is four buttons used to control the directions left, right, up and down. This continued as the standard for several years, where the difference in controllers was mainly adding more buttons for other actions, such as having four action buttons and shoulder buttons (1up, 2005).

In 1995 Nintendo introduced the joystick again, but this time it was controlled using the thumb, and not the entire hand such as the older joystick. This quickly became the new standard, and Nintendo announced that they would in addition add a Rumble Pak that would make the controllers rumble when certain functions were performed, providing haptic feedback for the user. In 1997 Sony developed the Dual Shock for their PlayStation 2, which had two joysticks, and two rumble motors, in addition to four action buttons and shoulder buttons. This setup is the standard used today, called the dual-axis controller (Juul, 2009). In addition there has been released several mimetic controllers, such as the Wii remote, PlayStation Move and the Kinect camera.

In the development for computer games controllers, the most noteworthy is that with the comings of 3D, people started using both arrow-keys and the mouse. Unfortunately this was not very ergonomic, and to solve this issue they started using the W, A, S and D keys as a replacement for the arrow-keys (WikiaGaming, 2010).

In November 2012 Nintendo released their latest game console, called the Wii U. The gaming controller is called the Wii U GamePad and features an accelerometer, gyroscope, camera, dual analog sticks, a built-in sensor strip and a built-in microphone. In addition it supports near field communication, which among other things can be used for figurines that can interact with the console and wireless credit card payments with compatible cards. The console is also compatible with the Wii Remote Plus, Nunchuk and Wii Balance Board, as well as a Wii U Pro Controller for more traditional controls (Nintendo, 2013).

The Wii U is the first console in what is often referred to as the eight generation of video game consoles, and both Sony and Microsoft are working on new releases of their own, where Microsoft have revealed that the new generation Xbox will include the Kinect camera embedded in the system (Xbox, 2013). We can see that the new generation consoles are already including different types of technology.

3.2 Human-computer interaction

In this thesis I will use theories and methods from the field of HCI to try and support the tentative hypothesis, and to help develop the conceptual model.

The field of **human-computer interaction** (HCI) is a multidisciplinary field, with researchers coming from backgrounds such as informatics, cognitive science, psychology, sociology, design etc. The goal is to create, evaluate and implement interactive products, to give a good user experience, which involves the study, planning and design of the interaction between people and computers (Carroll, 2009). In the next sections I will present theories that are of interest to this thesis.

3.2.1 Theories of embodiment

In theories of embodiment one looks at how the mind, body and world affects each other through our actions and thoughts. These theories are derived from a particular thread of philosophical thoughts from the late nineteen century called **phenomenology**. There are four philosophers who each had their own phenomenological positions, Husserl, Schutz, Heidegger and Merleu-Ponty. For this thesis, I will mainly focus on the views of Heidegger and Merleu-Ponty. For Heidegger, embodied action was essential to our mode of being and to the ways in which we encountered the world, and Merleu-Ponty emphasized the critical role of the body in mediating between internal and external experience (Dourish, 2004). A definition which Dourish (2004 , p.126) proposes is that "*Embodied Interaction is the creation, manipulation, and sharing of meaning through engaged interaction with artifacts".*

Here I will also mention the **cyborg theory** proposed by Haraway (1984) which talks about how the tools we use becomes a part of our bodies as prosthetics, effectively making us cyborgs. This way of thinking can easily be compared to theories of embodiment, and how our body (including tools we use, such as a blind man's cane) affects how we interact with the world. Although I believe that Haraway has several good points, I have chosen to exclude her theories, because I believe it will overlap with the embodiment theories.

3.2.2 Distributed cognition

As a cognitive theory, distributed cognition seeks to understand the organization of cognitive systems. However, it differs from other cognitive theories, in that it also considers interaction between people with resources and materials in the environment (Hollan et al., 2000). There are two theoretical principles in distributed cognition. The first concerns the boundaries of the unit of analysis for cognition. Distributed cognition looks for cognitive processes, wherever they may occur, on the basis of the functional relationships of elements that participate together in the process. The second principle concerns the range of mechanisms that may be assumed to participate in cognitive processes (Hollan et al., 2000).

Embodied cognition

A tenet of the distributed cognition approach is that cognition is embodied. This view bases the research on the thought that it is not an incidental matter that we have bodies locking us into relations with our immediate environments. The human body and the material world then take on central, rather than peripheral roles. For the design of work environments, this means that work materials from time to time become elements of the cognitive system itself, such as a blind person's cane is a central part of the way that person perceives and interacts with the world (Hollan et al., 2000). For my research this will be relevant because the technology we use can also become a central part of the way we perceive and interact with the world, for instance how the Google glasses are believed to be incorporated into our everyday lives. When the technology becomes simple and intuitive, we might just start using the technology in a way that simply seems natural for us.

3.2.3 Pragmatist aesthetics

This view in HCI looks at how one should design interactive systems as a result of focusing on aesthetics. In this view aesthetics does not only mean something which looks beautiful, but also about aesthetic experiences, and how to design systems which support this. An example of an aesthetic experience in video games can be found in the horror genre, with games such as Silent Hill, where the game mechanics forces the player to move slowly and listen for oncoming threats. They create an atmosphere of dread, which in turn results in a specific kind of experience (Deen, 2011).

3.2.4 Brain-computer interface

A brain-computer interface (BCI) is when we use electrophysiological measures of brain function in a way to provide a new non-muscular channel for interacting with computers (Wolpaw et al., 2002). The use of BCI has mainly been to offer people with disabilities a way to communicate with other people, and a way to interact with and control technology. There are two classes of BCI's, dependent and independent. A dependent BCI uses activity in pathways which is needed to generate brain activity, like using peripheral nerves and muscles to focus on a specific place. Independent BCI on the other hand does not depend on the brain's normal output pathways, and uses only direct EEG signals (Wolpaw et al., 2002).

This field of research is important for my thesis, because I am going to use the neuroheadset Emotiv EPOC as one way of controlling the game. Therefore I need to understand how the technology works, if I am to be able to use the technology in a way that will feel natural for the users.

3.3 Game studies

The field of game studies is a multidisciplinary field, bringing in researchers from many different research fields, in addition to game developers. This field is important for the thesis because it researches different aspects of video games, such as structure, player experience and development. It is a rather new field of research, but it is growing rapidly. Because there is such diversity in researchers, coming from fields like sociology and film studies, there have evolved several different ways that people research games. Some focuses on the culture around the games, some on the players and others on the formal structure of the games (Egenfeldt-Nielsen et al., 2008). In this thesis I will mainly focus on the formal structure of the game, what mechanics will correlate best with which controllers. But I will also look at the players, and how the technology in the input controllers can affect their player experience.

3.3.1 Embodiment in video games

Theories of embodiment are already a part of game studies, and are used to research the player's immersion in games. Gregersen and Grodal (2009) writes that:

> "… interacting with video games may lead to a sense of extended embodiment and sense of agency that lies somewhere between the two poles of schema and image – it is an embodied awareness in the moment of action, a kind of body image in action – where one experiences both agency and ownership of virtual entities." (Gregersen and Grodal, 2009, p.67).

When we look at different controllers used in video games, we can see that the degree of embodiment differs from controller to controller. The standard gamepad has minimal player actions (p-actions), such as pushing a thumbstick forward, which will make the avatar move forward. Other controllers such as the Wii remote will have more p-actions, where you flick the controller, which in turns flicks a virtual tennis racket. But since the controller reacts to movements instead of body acts, it is possible to use it with smaller P-actions than the system setup offers (Gregersen and Grodal, 2009), as well as players using gestural excess, which is when the player uses movement beyond what is necessary (Freeman et al., 2012). The Kinect camera on the other hand reacts to body acts, and will therefore have even more p-actions. We can then say that p-actions directly correspond to embodiment in games.

3.3.2 Natural mapping

Natural mapping formulates and discusses four possible types of mapping between gaming controllers and gaming mechanics, and the likely relationship of each to mental models and gaming experiences. These mapping types are not orthogonal and may overlap with one another. They are also not the only types of mapping but rather fall along the continuum from completely arbitrary to completely natural. The four types (in order of naturalness) are: (1)

directional natural mapping, (2) kinesic natural mapping, (3) incomplete tangible natural mapping, and (4) realistic tangible natural mapping (Skalski et al., 2011).

Directional natural mapping is the most basic manner in which controllers can be naturally mapped. This happens by producing a correspondence between the directions used to interact via a control device and the results in the world or on a screen. Kinesic natural mapping is a type of natural mapping which involves body movements that correspond to real-life actions without having a realistic and tangible controller. Incomplete tangible natural mapping involves giving players something that partially simulates the 'feel' of an object on the screen or in the game environment. And lastly realistic tangible natural mapping adds a realistic, tangible element to provide the highest level of natural mapping relative to the other three (Skalski et al., 2011).

In the research conducted by Skalski et al. (2011) they found that players who were using a controller offering tangible mapping reported more perceived controller naturalness than those who played using a variety of directionally mapped controllers.

3.3.3 GameFlow

The GameFlow method is derived from the theory of flow in psychology (Sweetser and Wyeth, 2005). Nakamura and Csikszentmihalyi (2002) explains flow as a mental state of being fully absorbed in an activity. The conditions of flow includes perceived challenges, or opportunities for action, that stretch existing skills, making the challenges appropriate to one's capacities, and clear proximal goals and immediate feedback about the progress being made.

The GameFlow method is used to measure player experience (PX). PX is derived from the term **user experience** (UX) from the field of HCI and the term is today mostly used as an umbrella term for focusing on aspects beyond usability. The current ISO[5] definition focuses on a person's perception and the responses resulting from the use or anticipated use of a product, system or service. In game research the term has not been used as much as it has in HCI, and the way it has been evaluated there has been from using a variety of concepts such as immersion, fun, presence, involvement, engagement, flow, play and playability and social play (Bernhaupt, 2010).

In this thesis I am using the GameFlow method as a way to analyze PX in existing games and to help predict it in the conceptual model. The **GameFlow** method identifies eight criteria for player enjoyment, namely *Concentration, Challenge, Player Skills, Control, Clear Goals, Feedback,*

[5] International Standards Organization, ISO 9241-210:2010

Immersion and *Social Interaction* (Sweetser and Wyeth, 2005). Below I will briefly describe what is meant by each criterion.

Concentration
Games should require concentration and the player should be able to concentrate on the game.

Challenge
Games should be sufficiently challenging and match the player's skill level.

Player Skills
Games must support player skill development and mastery.

Control
Players should feel a sense of control over their actions in the game.

Clear Goals
Games should provide players with clear goals at appropriate times.

Feedback
Players must receive appropriate feedback at appropriate times.

Immersion
Players should experience deep but effortless involvement in the game.

Social Interaction
Games should support and create opportunities for social interaction.

3.3.4 Game mechanics

In games studies there have been multiple attempts to come up with a definition of what game mechanics are. For the purpose of this thesis I will be using the definition proposed by Sicart (2008), which defines game mechanics as methods invoked by agents for interacting with the game world. Where other definitions includes both rules and mechanics, this definition states that were game mechanics are concerned with the actual interaction with the game state, rules provide the possibility space where that interaction is possible. This definition allows the study of the systemic structure of games in terms of actions afforded to agents to overcome challenges, but also the analysis of how actions are mapped onto input devices and how mechanics can be used to create specific emotional experiences in players (Sicart, 2008).

Game mechanics can further be divided into three different types; *core mechanics, primary mechanics* and *secondary mechanics.* Core mechanics can be defined as the game mechanics (repeatedly) used by agents to achieve a systemically rewarded end-game state. Primary mechanics can be understood as core mechanics that can be directly applied to solving challenges that lead to the desired end state. Secondary mechanics, on the other hand, are core mechanics that ease the player's interaction with the game towards reaching the end state. Secondary mechanics are either available occasionally or require their combination with a primary mechanic in order to be functional (Sicart, 2008).

Because of the nature of this research, I found this definition to be very useful, as it divides mechanisms and rules. In his paper Sicart (2008) mentions that a possible use of his definition, is as a formal tool for describing and modifying mechanics in a coherent and comprehensive way, by understanding the relations between the different methods, its properties, and how those are mapped onto the control interface. This corresponds well with what I am trying to do with the conceptual model.

3.3.5 Gaming controllers

For the purpose of this thesis, gaming controllers refer to control devices that can be used to control games. In games studies there has been a great deal of research on different types of control devices, but this mostly focuses on specific gaming controllers or specific technologies. Freeman et al. (2012) writes in their article *"There has been a great deal of research on controlling user interfaces with handheld motion input devices [2, 15, 40] tangibles [4, 22], touch [34], in-air gestures [1, 8, 14], and the whole body [5, 10, 13, 20, 21, 24, 25, 33, 36, 41]."*, which shows that there a lot of previous research on gaming controllers. But at the same time there doesn't seem to be much research into the categorization and definition of different types of gaming controllers, but rather using regular terms such as Freeman et al. (2012) uses in their description of different gaming controllers.

In this research I will not look at specific gaming controllers, but rather focus on defining and categorizing different input-technologies that can be used in gaming controllers.

3.4 Systems development

Because I am going to do some development it is important to understand more about systems development, and the methods used there. For the single-case study the goal is to test the conceptual model in a near real-life environment, by developing a prototype game. The focus of the case study is therefore on the development process, and not on the finished product. To ensure that the development process is similar to that the conceptual model can be used for, I am following methods and practices from mainly two fields; agile methodology and game development.

The reason why I chose agile methodology was mainly because it is a well-used and recognized development methodology that focuses on quick results over comprehensive documentation, which worked well with my time constraints. In addition I have prior experience with agile development, and felt that it was better for the thesis if I used practices which were familiar to me, unlike game development. While I did not have any practical experience with game development, I did have theoretical knowledge of the field, and have researched and incorporated game development elements alongside agile practices.

3.4.1 Agile methodology

Agile software development is a development methodology based on iterative and incremental development. There are several agile methods that developers can choose to follow, such as Scrum, Kanban or eXtreme Programming (XP) (Fowler and Highsmith, 2001). What is often common between agile methods are that they promote development, teamwork, motivation and quick adaptation. Some of the practices used in these methods include pair-programming, code ownership and short planning meetings (Abrahamsson et al., 2002).

The most important aspects of agile methodology for this thesis are the iterative development process, producing code quickly and the possibility to adapt changes throughout the development.

3.4.2 Game development

Video games have high demands of performance, expansive content, quick and efficient release cycles, and immersive content. Because of this the video game industry has been shifting from the traditional waterfall development methodology to using more agile methods, such as Scrum, that allow these demands to be fulfilled in a cost and time efficient manner. Due to the natures of iterative agile development, and the presence of unique demands and requirements in video games, the development industry must apply unique and innovative solutions to each step of the software process. One of the differences between regular software development and video game development, is the game design document (Koepke et al., in press).

A game design document should describe several things. It should define the game by articulating what the game is as clearly as possible. It should define the core gameplay, often done by describing the main view, player activity and the user interface. It should describe the contextual gameplay, such as shell menus, tutorial mechanics and multiplayer mechanics. Also it should include a talk story which describes the world backstory, character backgrounds, the levels and missions. Lastly it should include the game assets, such as design of the 3D models, voices, motion capture, sound effects and music (Koepke et al., in press).

In my development process the main contribution from game development is the game design document. While I did not develop a fully-fledged game design document, I used aspects from it and made a game concept which defines the game, a world backstory and game assets. I excluded the contextual gameplay and I used the scrum wall and user stories to define the core gameplay.

3.5 Similar research

In this section I will present existing research similar to my own. I have looked at an article called *Biofeedback Game Design* by Nacke et al. (2011), where the similarity lies in researching how different sensors can be used to control and augment different game mechanics. Another article that is of interest for this work is *Measuring the Impact of Game Controllers on Player Experience in FPS Games* by (Gerling et al., 2011) where they examine the impact controllers had on a first-person shooter game (FPS). I have also looked at a conference talk made by McCallum (2013) called *Game Technology of the Future* which discusses different technology used in games, and how developers should take this into consideration when developing games. Lastly is an article called *Videogame Control Device Impact on the Play Experience* (McEwan et al., 2012), where the authors look at how different controllers can have an impact on the player experience.

3.5.1 Biofeedback Game Design

The paper *Biofeedback Game Design* by Nacke et al. (2011) proposes a classification of direct and indirect physiological sensor input, used to augment traditional game control. The authors had two main questions they wanted to answer; how would users respond when physiological sensors were used to augment, instead of replace, and what types of sensors would work best for which in-game tasks? To answer this they conducted a mixed-methods study, and their results show that their participants preferred direct physiological control in games (Nacke et al., 2011).

The main difference between this research and my own is that I propose a general conceptual model for control schemas, for different input-technologies and game mechanics, whereas they looked at a specific controller schema, and used their findings to generalize about gaming controllers and game mechanics. While we both develop a game, their game was used to gather data from users, while I am developing a game to test my conceptual model.

The research done by Nacke et al. (2011) is relevant to this thesis, because they are studying the impact biofeedback as a gaming controller can have on the player experience, and what game mechanics are suitable for the specific controllers. Because I am looking more generally at all input-technologies for games, their findings regarding biofeedback are important for me in developing my conceptual model.

3.5.2 Measuring the Impact of Game Controllers on Player Experience in FPS Games

(Gerling et al., 2011) conducted a study where they examined the impact controllers had on the FPS game *Battlefield: Bad Company 2*[6], comparing the game when played with keyboard and mouse and the PlayStation 3 controller DualShock 3. They compared the player experience to game usability issues, where their results show that switching to a new platform would cause more usability issues. They also found that the players would see themselves as more challenged, but that the players reported they had an equally positive overall experience as they did on platforms they were comfortable with.

This research is helpful for my thesis because we are both trying to measure the impact of gaming controllers, but it differs in that this research uses a well-established and popular game, with the use of standard controllers, resulting in findings specific for that environment. My research on the other hand looks at different input-technologies and different game mechanics in general. But even though the research process is different, their findings are still highly relevant for my work.

[6] CE, E. D. I. 2010. Battlefield: Bad Company 2. Xbox 360, PlayStation 3: Electronic Arts.

3.5.3 Game Technology of the Future

In a talk at a game developer conference McCallum (2013) introduces the technology that we can expect to see in 2013 and its importance for the video game industry. The talk includes some guesses at future technologies and it discusses how video game developers will need to change in order to prepare for the large changes that are happening in the game development industry over the next 3-5 years.

This talk is helpful for my thesis because McCallum is discussing the same issues I have found, which eventually prompted me to write this thesis. He does not give any specific instructions on how the industry can work to avoid this issue in this talk, but he raises several interesting issues and viewpoints, such as possible play experiences and how we need to actually understand the technology in order to use it properly.

3.5.4 Videogame Control Device Impact on the Play Experience

McEwan et al. (2012) conducted a study that sought to clarify the influence the control interface has on the play experience. In the article they categorized three commercial control devices using an existing typology, most notably natural mapping (as presented in section 3.2). Then they used a within-group experimental design aimed at measuring differences in play experience across 64 participants. Their findings showed that people's positive response to the play experience seemed to be related to the degree of natural mapping of the control device, and not to their performance or capability with that device. In short, people seemed to respond positively to more naturally mapped interfaces even when their performance was worse with them, much the same result as (Gerling et al., 2011) found.

For this thesis their research is relevant because McEwan et al. (2012) are trying to do the same thing as I am, to study control devices and how they can have an impact on the player experience. Their findings on how natural mapping increases the player experience, even though they performed worse with it, is information that is very interesting when developing the conceptual model.

Even though we are both studying the impact control devices have on the player experience, the research differs in the way we approach the topic. They are taking a more quantitative approach (similar to Nacke et al. (2011) and Gerling et al. (2011)), and are relying solely on existing typology. I on the other hand am taking a qualitative approach, and while I am looking at existing typologies, my intent is to use it to propose a different model which includes game mechanics.

3.6 Theoretical framework

Firstly I want to explain where I am placing this research, which is in Human-Computer Interaction (HCI) as well as Game Studies. Because my background as a researcher stems mostly from HCI, this highly affects the theories and research methods that make the foundation of this thesis. From my background in Game Studies, and from my background as a gamer, I have knowledge about the development of games, gaming history and commercial aspects of the field, which I have also been using in this thesis. It is this mix of knowledge that I have used to reach the conclusions. In the theoretical framework I will incorporate the research mentioned in the previous sections, and explain how it supports the tentative hypothesis and the conceptual model I am proposing.

Heidegger (Dourish, 2004) talks about embodied action, and how our body affects the way in which we see and experience the world. I believe that the same is true for the technology we use when playing video games. The technology is what we use to control and experience a video game, and therefore that technology is important to take into consideration when designing a video game. The same way a blind man navigates differs from the way a man who can see navigates; I believe that a player using one type of gaming controller will make the experience differ from another player's experience when using another type of controller. When developing the conceptual model, one way I have taken into account how different technologies can support a high degree of embodiment, was to consider the p-actions the technology have, and what mechanics would be most suitable for the different degrees of p-actions. Similarly I have considered how some input technologies support natural mapping more than others, and taken this into account as well.

Today we have multiple technologies that are at our disposal when playing a video game, and if we choose to use several of them at the same time, we need to think differently about the gaming experience than we do when there is only one controller. Even if the player only has one physical gaming controller to manipulate, that single controller can incorporate multiple input-technologies, such as the Wiimote and PlayStation Move does, and this is something I needed to take into consideration when developing the conceptual model. In order to do this, I have looked at distributed cognition, especially embodied cognition, and how those theories fit in with the conceptual model. This prompted my decision to look at the different technologies used to control video games, instead of my initial idea of looking at physical controllers. By looking at the cognitive system, and recognize that while the physical artifacts play an important role in defining the cognitive system, this does not mean that a cognitive system with only one controller is necessarily a simple system, but that it can in fact be complicated because there are many different technologies incorporated into it at the same time. I believe that when developing games, we should think about the controllers as components in a cognitive system, and that the technologies used need to support each other in such a way that the player can simply focus on playing the game, instead of focusing on which controller should be used when.

In the next paragraph I will talk about how this research and the similar research mentioned in section 3.2 combined gives a better understanding of how technologies affect games, and why this is important.

The talk *Technical Development in Games* (McCallum, 2013) is reaching out to both game developers and future game developers, bringing the topic to the ones who actually need to take this into consideration in their work. Hopefully this thesis will further give developers a tool to use in order to address the issue, and which can be used by anyone who wishes to be more attentative to what technology they currently are, and could, be using. But in order to figure out more on whether or not the conceptual model is helpful, we need more specific research on the mechanics and the technology, and how they fit together. And this is what the authors of *Biofeedback Game Design* (Nacke et al., 2011) are already doing; cross referencing the different components and researching which controller schema the player's preferred. Their research differs from mine because they are focusing on how technology can be used to augment, instead of focusing on what properties each technology used in gaming has, and how they can best be utilized.

We then have the article *Measuring the Impact of Game Controllers on Player Experience in FPS* (Gerling et al., 2011), who is comparing the same game but used with different controllers, and how that affects the player experience. Although this research is on a perhaps smaller scale when it comes to the differences in the technology used, it is still important to know how we are affected by even smaller differences between controllers. But also it gives us an idea of how important it is that the game is enjoyable, because as they found, the PX did not change when using the different controllers, meaning that the game in itself is extremely important, and a fun game could perhaps outweigh bad controllers.

In the article *Videogame Control Device Impact on the Play Experience* (McEwan et al., 2012), the authors are doing research similar to *Measuring the Impact of Game Controllers on Player Experience in FPS* (Gerling et al., 2011), but with a bigger difference between the controllers. What is interesting to notice is that the findings from both studies say the same; namely that the

PX was not necessarily worse when performance was worse. This could mean not only that the game in itself is extremely important, but also that gaming controllers with a high degree of natural mappings can increase PX even though the difficulty increases to some degree.

4 Methodology and research strategy

4.1 Research strategies

I have mainly been using one research strategy for this thesis, namely qualitative. In the sections below I will explain briefly the research strategy chosen for this work, and why that strategy was chosen over quantitative.

4.1.1 Qualitative research

I chose to use a qualitative research strategy, because I needed to gather data which could be used to answer "why" and "how" questions. Also the field in which I have been researching is very new, which made exploratory and qualitative research more appropriate.

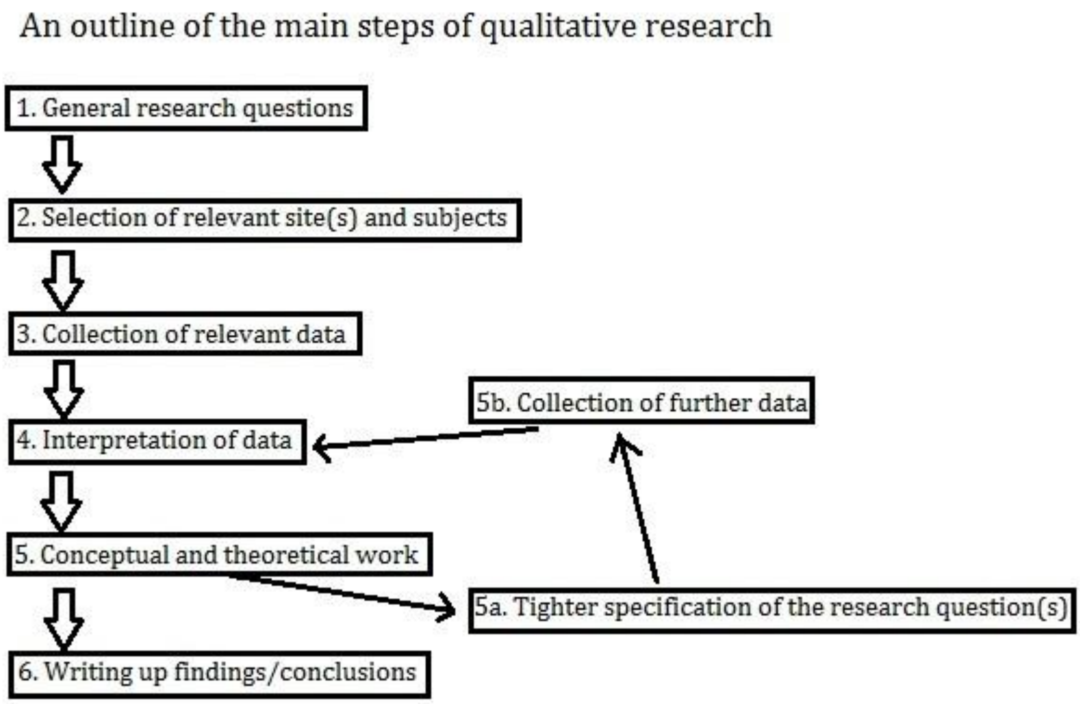


Figure 1: Main steps of Qualitative Research (Bryman, 2008, p. 370)

The research process is organized similar to the main steps for qualitative research outlined in **Error! Reference source not found.**. At first I begin with presenting the research questions and the literature review, which corresponds with steps 1 and 2. After that I collected data from video games in a multiple-case study, and did an analysis on the data, which corresponds to steps 3 and 4. Following that I used the analysis from the multiple-case study, in addition to theories presented in the literature review, in order to develop a conceptual model. This corresponds to step 5 and 5a. After that I used the conceptual model in a single-case study, which corresponds to step 5b. I then did an analysis of the collected data, which corresponds with step 4. In the end I have a discussion on the findings, which corresponds with step 4 and 5, as well as a conclusion on findings which corresponds with step 6. By following the practices from case studies, and qualitative research, the hope was that this would provide the best process in order to gather findings that could answer the research questions.

Some of the criticisms for qualitative research are that it is too subjective, it is difficult to replicate, that there are problems of generalization and a lack of transparency (Bryman, 2008). Despite of these criticisms I still believe it will be the best approach for this research. With regards to the general criticisms towards qualitative research, such as it being too subjective, I think that because my research is exploratory, it is crucial that I don't start out the data gathering by excluding some areas in order to get a more detailed view at another. When it comes to the research being difficult to replicate, I agree that this could be a problem, but my goal with this thesis is not to discover a definite truth about something, but rather to explore possibilities and record my findings so that future research can build on and evaluate it. Regarding the criticism against qualitative strategies about generalization, I agree that it is difficult to make generalizations on populations based on the few cases I have been researching, but I believe it is first crucial to explore the field and come up with a possible solution, which could later on be studied further using quantitative research. And for the criticism regarding a lack of transparency, this is something I will try to address when collecting my data.

Where qualitative research is often associated with collection of words, quantitative research is often associated with the collection of numerical data. It is also more related to the natural science approach than qualitative research (Bryman, 2008). Although I could have used this research strategy for this thesis, I chose to use qualitative research instead because of the subjective nature of the data I hope to collect. If my research questions were more focused on the specifics of using new technology in games, I would probably have chosen to use a quantitative strategy, but because I am interested in figuring out a possible and general model, I believe that subjective information will be more applicable for my research at this point. I do however believe that the conceptual model which I propose, should be further tested using quantitative research, but that is not within the scope of this thesis.

4.2 Research method

4.2.1 Methodologies

When deciding what methodology I would be using for this research, I considered using systems development research, but I found that case study research would be more suitable as exploratory research into the field, in order to get a general understanding of the relationship between input-technology and gaming controllers. Although I could arguably have used systems development methodology, I think this would have been a better approach if I instead was mostly interested in the actual development process and findings based on observation of test-subjects in controlled experiments (Nunamaker JR et al., 1991).

4.2.2 Case study research

For this thesis I chose to apply case study research. Case studies are a social research method that have been used in a variety of disciplines over the years, such as psychology, sociology, political science, anthropology, social work, business, education, nursing and community planning. It is the preferred research method for answering "how" or "why" questions, or for research where the investigator has little control over events. It is also the preferred strategy when the focus is on a contemporary phenomenon with some real-life context (Yin, 2008). Because case studies focus on single cases, critics of the method believe that findings cannot be used to establish reliability or generality, and that the findings are biased due to the intense exposure the researcher has to the case. Yet researchers show that they get successful results by using the method (Soy, 1997).

Case studies can be either single-case or multiple-case, and they can have an embedded or holistic design, see Figure 2. Yin (2008) explains that a case can be an individual, a decision, a program etc., and that a single-case study focuses for instance on an individual, while a multiple-case study would focus on multiple individuals. An embedded or a holistic design relates to how many units of analysis there are. The holistic design has one unit of analysis, and is often used for studying the entirety of a phenomenon, while an embedded design has more than one unit of analysis and looks more closely at specific phenomena. At the same time the cases must be carefully chosen to either give a literal replication or a theoretical replication. A literal replication is when cases are predicted to have the same results, and theoretical replication is when cases are predicted to have contradictory results (Yin, 2008).

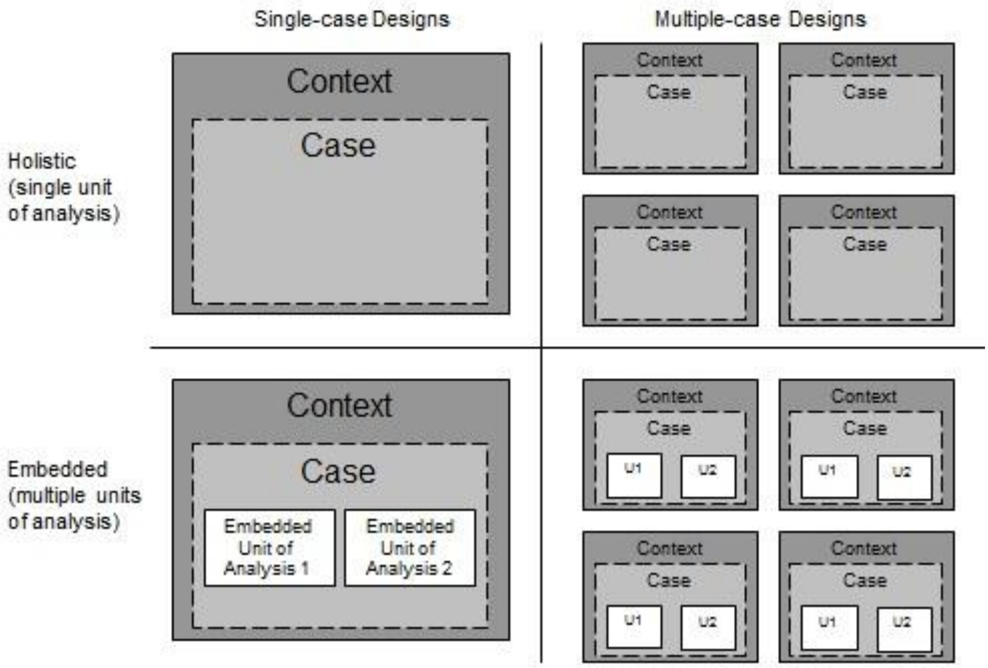


Figure 2: Basic Types of Designs for Case Studies (Yin, 2008, p.46)

The reason why I chose this research method is because research on input-technology and gaming controllers are still at an early stage. By using an exploratory research strategy, I can gather data by looking closely at multiple sources of evidence and triangulate the data to gather findings. When looking at the research questions, they are typical "why" and "how" questions, which makes case study a suitable research method to apply (Yin, 2008).

I decided to do a multiple-case study with an embedded design to gather findings that could be used when working on the description and theory building of this research. The cases I chose to study are four video games, and the units of analysis are the control schema and associated game reviews. The cases are chosen to give a theoretical replication.

In addition to the multiple-case study I also wanted to do a holistic single-case study for the outset of my theory generation, and for late theory testing. I therefore decided to develop a prototype game using the conceptual model, and use that development process as the case used in the single-case study. This type of case study is described by Benbasat et al. (1987) as an application description, which is a type of case study used by practitioners that describe the author's experiences in implementing a particular application.

4.3 Data collection methods

In this section I will introduce the methods for gathering data for this thesis. I will be using three of the data collection methods which Yin (2008) proposes; documentation, direct observations and physical artifacts.

4.3.1 Documentation

In order to find information which could be useful for me throughout my work with this thesis, I have continually been searching for articles and reviews on the web, as well as making use of the libraries at my availability. By searching the internet and libraries I have gotten relevant information, presented in the literature review and in the appendix, from research fields, official gaming Medias as well as from development fields. .

4.3.2 Direct observations

I have been in contact with different communities as a way to discuss and learn from other people in order to gain as much knowledge about the specific technologies as possible. I have also used observations which I have made during the development of the prototype game.

Stack-overflow

This is a community for helping people online with programming issues. I have been using this community for help with the development of my game.

Unity3D

There is a big community attached to the Unity3D software, especially the Unity forum. I have been using this forum for help with using the software.

Exertion Games Lab at RMIT University

The Exertion Games Lab focuses on the intersection between Games, Technology and the Physical. I have been a visiting researcher at the Exertion Games Lab while writing my thesis, which has let me discuss relevant research and research projects, as well as having a space to work.

Of particular help from this community has been the help I have gotten from people with experience with the Kinect technology and the Emotiv EPOC technology.

Single-case study: Folktale

In order to gather information on how the use of the conceptual model could work in an actual development process, I have used the observations I made from the development of the prototype game *Folktale*.

4.3.3 Physical artifacts

I have used data gathered from examining and using technologies used for gaming today in new ways, as well as video games. The physical artifacts I have used in this thesis are the controllers Emotiv EPOC, the Kinect and the PlayStation Move, in addition to the video games *Bioshock: Infinite, Kung Fu Rider, The Elder Scrolls V: Skyrim* and *Steel Battalion: Heavy Armor*.

4.4 Multiple-case design

For the multiple-case study for this thesis, I am using an embedded design. I have studied four video games as cases, in order to find possible answers to the research questions "how can input-technologies affect the enjoyment of video games?" and "do different input-technologies work better for different types of game mechanics?". When selecting the cases for this study I used the website Metacritic (2001), which ranks video games based on both professional reviews and user reviews, in order to decide which games to analyze. I wanted to find cases that would result in a theoretical replication by triangulating the data, and use the findings to develop the conceptual model. I therefore looked specifically for cases that had the highest and lowest Metacritic score, on the assumption that their corresponded with the level of enjoyment they offered, and where the use of input-technology could arguably play an important role in their score. This refers back to the research questions on how input-technologies can affect the enjoyment of video games, and if different input-technologies work better for different types of game mechanics.

The games I chose were the highest ranked and lowest ranked PlayStation Move games, which are *Bioshock: Infinite*[7] and *Kung Fu Rider*[8] in addition to the highest ranked Kinect game, *The Elder Scrolls V: Skyrim*[9]. My initial plan was to also use the Kinect game with the lowest score,

---

[7] GAMES, I. 2013. BioShock: Infinite. PlayStation 3: 2K Games.

[8] STUDIO, J. 2010b. Kung Fu Rider. PlayStation 3: Sony Computer Entertainment.

[9] STUDIOS, B. G. 2011a. The Elder Scrolls V: Skyrim. Xbox 360: Bethesda Softworks.

which was *Self-Defence Training Camp*[10], but in the reviews I found that the reason it had received such poor scores were mainly because of the concept, and not on the gaming controller. So instead I decided to study the game *Steel Battalion: Heavy Armor*[11], which has the 8th lowest rating on Metacritic, and which I knew from before had received bad reviews based on how difficult it was to play with Kinect.

For each case the units of analysis are game reviews and control schemas. The websites used to find the reviews are; The Escapist (2005), Gamespot (1996), IGN (1996), Joystiq (2004), Push Square (2009) and Game Informer (1996). Some of the games have fewer reviews than others. This is because I decided I would rather prioritize that the reviews came from similar sources, than to have the same number of reviews per game.

I chose games with Kinect and PlayStation Move for several reasons; because the controllers are industry standards, because the controllers are known to me, and because they use different technologies.

When it comes to the multiple-case design there are some limitations I want to address as well. Some of the limitations include the low number of cases, the limited available documentation and the narrow focus I employed. While I looked only at the relation between the controllers and the game mechanics, the games are most likely successful and unsuccessful due to several components. This could mean that the popular games are successful or unsuccessful despite their controller configurations, in which case I would base my findings on wrong data. In order to counteract this I have looked at different reviews and searched for their opinions regarding the controller configuration, with the thought that if the reviewers have noticed the controller configuration, then it probably has something to do with the degree of success.

When it comes to the limitations regarding the number of cases, I decided to use only a small number because of the time frame of the thesis, especially since I also needed to take into account the time needed for the single-case study. I do however believe that the conceptual model should also be compared to several other video games as well to determine its validity, and to further customize the model based on those findings.

When it comes to the limited documentation, I believe that because I have a narrow focus on the games, I do not need vast amounts of documentation on other aspects of the games. If I were to

---

[10] UBISOFT 2011. Self-Defense Training Camp. *My Coach.* Xbox 360: Ubisoft.

[11] SOFTWARE, F. 2012. Steel Battalion: Heavy Armor. Xbox 360: Capcom.

look more closely at the entire game, then I believe that it would be good to look at games that have been researched more thoroughly, where there are more documentation to look at.

4.5 Single-case design

For the single-case study I have used a holistic design, where the case study is the prototype game I develop, called *Folktale*, and the unit of analysis is the development process. This kind of case study is what Benbasat et al. (1987) describes as an application description. They are written by practitioners and detail the author's experiences implementing a particular application and conclude with a list of "dos" and "don'ts". The reason why I chose to do an application description instead of for instance doing a quantitative study of the game's controller configurations was because I wanted to implement the conceptual model into the process, and see how it could work in a near real-life environment. Because the goal behind this single-case study was to gather data about the implementation of the conceptual model, this also excluded the possibility of choosing another case to study.

I realize that my results from this case study will be biased as I have the role of both researcher and developer, but I believe that my findings are still relevant as I am not attempting to use the gathered data to prove that my conceptual model is accurate. Rather I am attempting to uncover possible issues that may arise when using the model in practice, instead of only in theory, and present this as "dos" and "don'ts" for when applying the conceptual model.

The game will not be fully developed as a playable game, for a couple of reasons. One is that I do not have the competence needed in order to do the graphical design, the level design and the audio, which is needed in order to make the game enjoyable. I believe that the lack of these crucial elements could also affect the feedback I got from any players, for instance if they found the game boring or ugly they might be more negative towards the controller schema as well. Another reason is that I believe the scope of this thesis is too small to perform such research with the time and resources it would need. I do however believe that such research would be beneficial and that it is necessary. I hope this thesis will give a good basis for possible future experiments on this, and on other general research on this subject.

4.5.1 Practices

In this section I will present the practices used for the development of the game prototype in this thesis. I did not use a specific development methodology, because I was the single developer for this product, which meant that several practices would not be applicable to this project, such as for instance stand-up meetings in Scrum. Although there are some methodologies for single developers, I believe that I will get a better result by instead applying different practices from the agile methodologies, Extreme Programming (XP) and Scrum. The reason for this is because I have prior experience with these methods. I have also used some practices from game development and interaction design.

Spike

Because I was using technologies that were new to me, I needed to set off some time for learning, which I did by using them in practice. This is often called an initial spike (Shore and Warden, 2007). By setting aside some time in the beginning to focus only on learning how to use the different technologies, I was able to more precisely predict how my development process would be, and what I would be able to do, and how to do it.

User stories

In order to document the criteria for the game, I used story cards. This is a way to describe each activity which should be possible to do in the system (Martin, 2003). This practice is used in both XP and Scrum, and is normally used as a way for the customer and developers to describe the system. Therefore it does not contain technical specifications, only short descriptions of an activity.

Product backlog

I used a product backlog to organize the story cards. This is a practice often used in Scrum, where the point is to keep track of all tasks which need to be done, and the priority of each story (Abrahamsson et al., 2002). The product backlog also gave a rough overview of dependencies between the story cards.

Short cycles

One of the most characteristic practices in agile development is that it is done in iterations (Martin, 2003). I worked with short iterations consisting of two weeks. After two cycles with development, the next iteration would be dedicated to data gathering, followed by two further development cycles.

Scrum wall

The story cards were organized on a scrum wall (Abrahamsson et al., 2002), which were divided into an overall process plan in addition to an iteration plan. The iteration plan showed which phase each story card was in; development, testing or finished, while the process plan was dedicated to showing statistical information such as a progress graph.

Prototyping

The game in itself was only a prototype, which meant that I could exclude some aspects of the game that would have been necessary for a fully developed game. An example of such a practice that I did not need to include was contextual gameplay.

Concept statement

A concept statement is a very short description of the concept of the game (Ryan, 1999). This is the same as defining the game in a game design document (Koepke et al., in press).

Gameworld story

This is a short story which introduces the player to the game universe (Koepke et al., in press). I chose not to include character backgrounds, the levels and missions because I was only developing a prototype of a game, and therefore did not need it.

Game assets

While this should include design of the 3D models, voices, motion capture, sound effects and music (Koepke et al., in press), I instead opted to use a concept art poster which showed the aesthetic look and feel the game should have in the end (Ryan, 1999).

4.5.2 Tools

For the development process I used a variety of tools, which I will present in this section.

- **Game Development Tool: Unity3D**

  This is a graphical game development tool used by both smaller and bigger companies. It is made specifically for developing games in a 3D environment, and for reuse of game elements. It is a free development tool, and there are downloadable plugins such as for developing for the Kinect and Emotiv EPOC (Unity3D, 2012).

- **Emotiv EPOC development kit**

  I used the Emotiv EPOC headset with a developer's kit, which I used in order to program the headset (Emotiv, 2012a).

- **Kinect camera & Zigfu ZDK**

I also bought the Kinect camera, but that came with a development kit which I could not use with the Unity3D software. Instead I used the development kit Zigfu ZDK.

- **Adobe Photoshop CS6**

  In some small cases I needed some graphical elements for the game, I used Photoshop CS6. I also used this program for some quick sketches on the concept art poster.

4.6 Samples and validity

4.6.1 Samples

The main samples for this research will be the video games used in the multiple-case study and reviews of them. As mentioned in section 4.4, the reason why I chose these samples was because of their success degree, and that I hoped by selecting a couple of unique cases I could also compare them and look at the differences and similarities between cases. Another sample used for this thesis is the prototype game I developed for the single-case study; *Folktale*. By using the development process from *Folktale* as a sample, I wanted to gather data that first becomes evident when a concept is tried in a near real-life environment.

I believed that the sum of these samples would provide data that would help in the construction of the conceptual model, as well as give me practical experience that could perhaps result in developing practices that could accompany the model and give insights into how the conceptual model may work in an actual development process.

4.6.2 Validity

In this thesis I will need to ensure that my research is well constructed to ensure construct validity, internal validity, external validity and reliability (Soy, 1997). Construct validity is something many case studies have issues with, because of potential investigator subjectivity (Tellis, 1997), and my research is no different. This is especially a problem with the single-case study, but I believe that because my findings from that case will not be used in forming my conceptual model, but rather as a presentation of my experiences and a summary of what I have learned from the development process, this limits the validity problem somewhat. Although there are some issues with construct validity in the multiple-case study as well, I have hopefully overcome them to some degree by looking at the opinions of professional game reviewers in addition to my own observations, which is what Yin (2008) proposes as a remedy; using multiple sources of evidence.

Internal validity is usually a problem with "inference" in causal cases, which the multiple-case design can arguably be categorized as, and can be dealt with using pattern matching (Tellis, 1997). To counteract this in my study I am using models that show the controller configuration of games and in the analysis I will look for patterns in those configurations.

When it comes to external validity, the issues mostly relate to single-case studies, as they might not be generalizable beyond the immediate case, but this again is mostly directed at the statistical generalization and not the analytical one that is often emphasized in case studies (Tellis, 1997). Although there could arguably be a problem with generalization for games based solely on the controller configuration, I still believe that my findings can be used to generalize on other games controller configuration, if not on the entire game, and that this still is valid and important findings.

Lastly there is the issue with reliability, which refers to the stability, accuracy and precision of measurement (Soy, 1997). In order to increase the reliability of this thesis, I have included the documentation that is used in both my case studies in the appendix, and I have tried to give as good a recount on the practices and methods used in my development process in the single-case.

I do admit that there are still some issues with validity of this thesis, which could have been improved if I had documented different data from the video games and organized them in a case study protocol (Tellis, 1997). This is something I would have changed if I were to do this again. I would also have focused more on the multiple-case study, and abandoned the single-case study, even though I learned a lot from it. That way I could have focused more on the multiple-case study, chosen more cases and had a more in-depth study than I have today and hopefully ended up with a stronger foundation for the conceptual model, which had a higher degree of validity. Doing the single-case study could then have been a part of further research regarding how and if the conceptual model was useful.

4.7 Analysis

In the following sections I will present the units of analysis for the case studies and the data collection that I have used.

4.7.1 Units of analysis

The unit of analysis is what the study focuses on (Yin, 2008). For the multiple-case design this is the control schema and game reviews for each game, and for the single-case design it is the development process of the game. By focusing on the control schema, I hoped to gather data that could help answer the research questions "how can input-technologies affect the enjoyment of video games?" and "do different input-technologies work better for different types of game mechanics?", by comparing video games with different degrees of enjoyment, which was predicted to give contradictory results. I also hoped that by focusing on the development process of *Folktale* that the experiences I gained as a developer could help me as a researcher by testing the conceptual model in a near real-life environment.

4.7.2 Data collection

The data collection used for the multiple-case study consists of a control schema for each case, four in total, in addition to video game reviews for each case, and the physical artifacts Kinect and PlayStation Move. The data collection for the single-case study consists of my own notes and accounts of the development process, as well as the programming code of the game *Folktale*, and the physical artifacts Kinect and Emotiv EPOC.

4.7.3 Issues

As mentioned before I agree that there are some issues with the research method that could have been done differently. I believe the main issue lies with the single-case design, because of the bias I as a researcher have when looking at my own work as a developer. However, I would like to point out that my findings from the single-case design is not used to formulate the theoretical work of developing the conceptual model, but rather to give an account of my experiences as a developer when using the conceptual model in a near real-life environment. While I believe the single-case study has given me a valuable insight into the topic, if I were to conduct this research over again, I would probably have chosen not to do the single-case study, and rather put all my focus on doing a more in depth multiple-case study instead. This would also have helped with another issue I have found from this research, which is that the multiple-case study would give more weight if it had included more cases.

5 Multiple-case study: video games

This section will first present the four cases studied, and then give a brief recollection of the data collections before presenting the analysis and findings from this case study. The findings have been used in order to gain knowledge on what game mechanics and input-technologies are suitable for each other. This is based on the GameFlow model and p-actions as well as natural mapping. I have also taken into consideration the Metacritic score on each game.

5.1 Cases

The four cases I have looked at are the following; *Bioshock: Infinite*, *Kung Fu Rider*, *The Elder Scrolls V: Skyrim* and *Steel Battalion: Heavy Armor*. Each case is presented with a descriptive text, a controller schema and a short summary of accompanying game reviews.

5.1.1 BioShock: Infinite

Published by 2K Games and developed by Irrational Games; *BioShock: Infinite* is the third game in the series *BioShock*. It was released 26th of March 2013, and it is a first-person shooter game (Infinite, 2013). Although it was released for PC and Xbox 360 in addition to the PlayStation 3, I have focused only on the PlayStation 3 version with the Move controller.

The game is set in 1912, where the player takes the role of former Pinkerton agent Booker DeWitt, sent to the flying city of Columbia on a rescue mission. The woman he is sent to rescue is Elizabeth who have been imprisoned since childhood. During their daring escape, Booker and Elizabeth form a powerful bond -- one that lets Booker augment his own abilities with her world-altering control over the environment. Together, they fight from high-speed Sky-Lines, in the streets and houses of Columbia, on giant zeppelins, and in the clouds, all while learning to harness an expanding arsenal of weapons and abilities (Bioshock Infinite, 2013).

Figure 3: Control schema for BioShock: Infinite

The controllers used to play this game are the PlayStation Move controller along with the associated navigation controller. The player navigates the game using the joystick on the navigation controller while physically aiming with the Move, which also controls the camera angle. Shooting is done by pressing the button on the back of the Move controller. While most of the other actions done in the game are controlled by pressing different buttons, the player uses the Move controller to punch an attacker by thrusting the controller forward.

IGN Podcast

Although this is not specifically a review, I would like to include this podcast because of the explanation *BioShock: Infinite* director Ken Levine gives as to how they supported the Move controller in the game. He talks about how some gamers love Move, and how other gamers are afraid of anything other than traditional controllers. Their solution then was to make a game that could easily be played both with the traditional DualShock 3 controller, as well as the Move. But he also says that there are some added benefits on the PlayStation 3 version that Move supports, and that this makes the game allover better. (IGN, 2013)

PushSquare

The review on PushSquare is titled “Is BioShock: Infinite Any Good with the PlayStation Move?” to which their conclusion is “Quite simply, the answer’s yes”. Here the reviewer was already accustomed to playing the game with the DualShock 3. He says that it took some time getting used to the Move controller, however, because the in-game button prompts adapt to whichever control device you are using, the adjustment feels as seamless as possible(Barker, 2013).

There are several control presets available; Casual, Standard, and Expert. Each setting adapts the dead zone and turning speed slightly, as well as the level of auto aim. Although the reviewer writes that regardless of which option you choose, manually lining up your shots with the Move feels very satisfying. There is also a custom option, which allows you to tweak the game's dead zone, sensitivity, turning speed, and auto aim to your own personal preferences(Barker, 2013).

The reviewer writes that the only real downside he can find with playing the game with the Move controller is that the game doesn't make more use of the motion controls. He then proposes that simple gestures such as pulling levers could have easily taken advantage of the peripheral's technology instead of using regular buttons. He would also have liked the device's tracking orb to give feedback on whichever Vigor he was using, by changing color. His conclusion is that the implementation of Move is unspectacular yet perfectly functional (Barker, 2013).

5.1.2 Kung Fu Rider

This game was developed by SCE Japan Studio and published by Sony Computer Entertainment in 2010 as a PlayStation Move exclusive game. The player can choose to play as Toby, a private investigator, or Karin, Toby's secretary. The goal of the game is to escape from the Triad in Hong Kong, on an office chair (PlayStation, 2010).

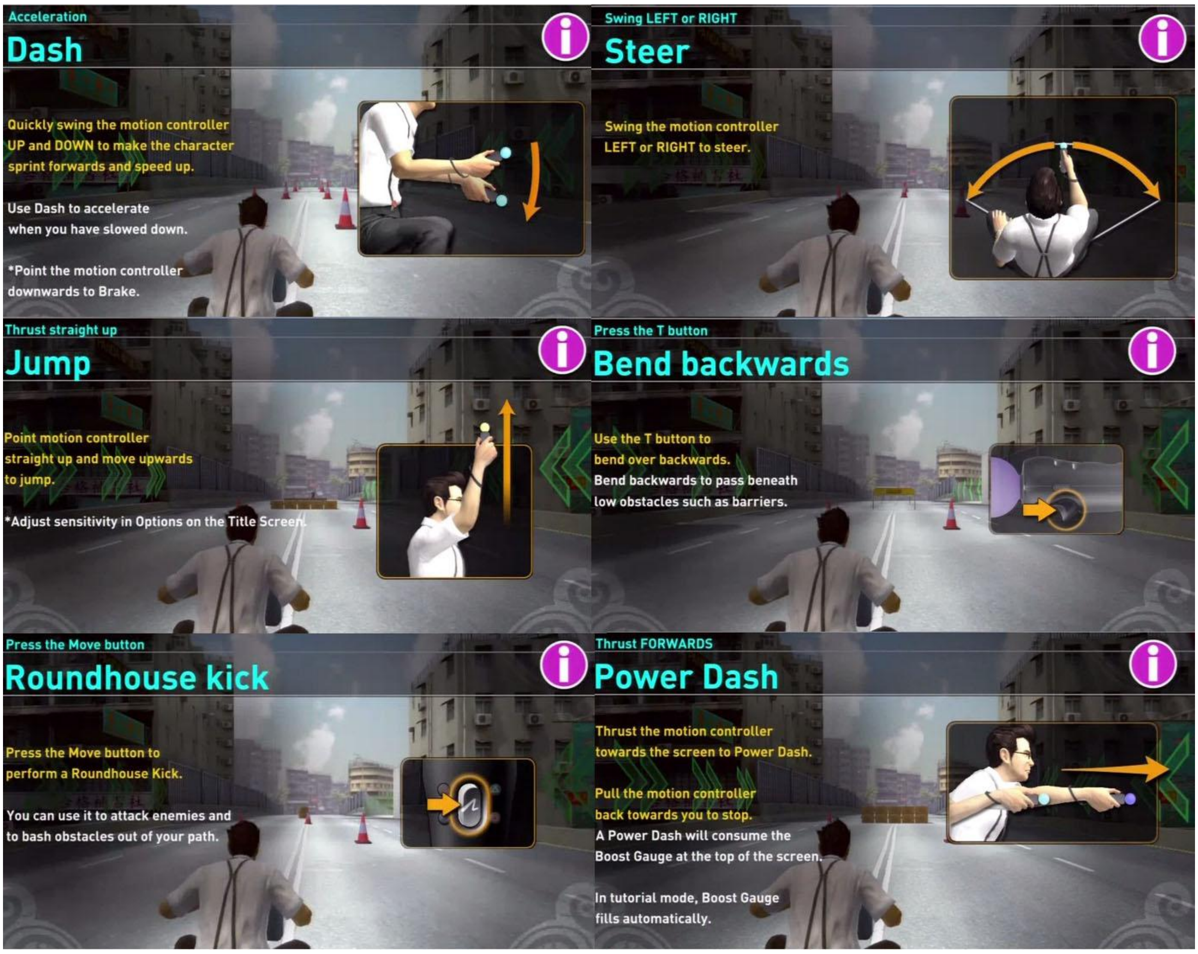


Figure 4: Control schema for Kung Fu Rider

The game is played using only the Move controller. By moving the controller up and down the player accelerates, and by swinging the controllers sideway the player steers. In order to jump the player has to flick the controller upwards, and to duck the player has to press a button. Another button is used to kick obstacles, and if the player needs to power dash (move forward quickly) the player has to thrust the controller forward.

Joystiq

For the reviewer at Joystiq *Kung Fu Rider* is a lovely game with a really neat concept, but the controls just about ruin it. He agrees that tilting the Move controller left or right to turn makes sense, the same does flicking up in order to jump and thrusting forward to accelerate. But besides that, it just gets weird and complicated. For him it was hard enough just to remember what everything does, and he says that nothing in the game feels precise or reactive enough, even on the most sensitive settings. His conclusion is that *Kung Fu Rider* is a pricey and poor execution of a cool concept (Nelson, 2010).

IGN

The reviewer at IGN states that his experience playing *Kung Fu Rider* was abysmal. He even points out that it is not because he dislikes the PlayStation Move, it is because of the game itself.

One of the things he had problems with was the accelerate mechanic, where the player needs to constantly flick the controller up and down. Not only did he found it tiring, but it directly conflicts with the jumping mechanic, which is just a more forceful flick of the controller upwards. He continues to say that the poor control schema applies to almost every aspect of the Kung Fu Rider controls. Turning was loose at best and ducking under objects was very inconsistent (Clements, 2010).

He does point out that watching Toby and Karin tumble through the air was definitely hilarious, but that it was not enough to warrant playing "this otherwise frustrating mess of poor motion controls" (Clements, 2010).

He concludes that *Kung Fu Rider* is not a good game. The controls were terribly unresponsive, the game punishes players unfairly and the level design was repetitive. While playing *Kung Fu Rider*, the PlayStation Eye takes random pictures of the player at key moments in the mission. The reviewer found it quite telling that in all those photographs, he looked really angry (Clements, 2010).

5.1.3 The Elder Scrolls V: Skyrim

The Elder Scrolls is the fifth game in the video games series *The Elder Scrolls* developed by Bethesda Game Studio and published by Bethesda Softworks. It is an action role-playing game and released on November 11th, 2011 for PC, PlayStation 3 and Xbox 360 (Wikipedia, 2010). The game originally did not come with Kinect support; this was instead released as a title update on 1st of May 2012.

The main story in *Skyrim* revolves around the player character's efforts to defeat Alduin, a Dragon who is prophesied to destroy the world. The game is set two hundred years after *Oblivion* (the previous game in the series), and it takes place in the fictional province of Skyrim, upon the continent of Tamriel, and the planet of Nirn. *The Elder Scrolls* series features an open world gameplay which returns in *Skyrim*; where the player can explore the land at will and ignore or postpone the main quest indefinitely (Wikipedia, 2010).

Figure 5: Control schema for The Elder Scrolls V: Skyrim

With the Kinect support the developers did not want to change the way the game was played, so they did not incorporate physical actions into the gameplay. Instead they wanted to enhance the experience of the game, which they did by adding voice commands (Gilbert, 2012). The game features over 200 voice commands meant to make the player more effective, and to enhance the player experience. Some of the voice commands are used for the interface of the game, such as "quicksave", "quickmap" and "quickitems weapons". By letting the user perform actions, such as saving the game with a simple command, the player becomes more effective, and by using commands to navigate in the map by saying "quickmap" and "where am I", the player saves a lot of time, especially since it can be a bit slow using the joystick to navigate on the map. The player can also use voice commands to communicate with an ally in the game by for instance saying "ally wait".

In addition to using voice commands to control the interface, *Skyrim* has incorporated voice commands into combat situations as well. In the game the player can learn a magical ability called Dragon Shouts, which lets the player attack by shouting. The player can learn multiple shouts which have a name both in English and in Dragon language. With Kinect support the player can use voice commands to perform shouts, so that instead of pushing a button to shout, they can now instead say "unrelenting force" or "fus ro dah" which will trigger the attack.

Joystiq

The review begin by saying that the Kinect update for *Skyrim* on Xbox 360 goes more for function than flash, and the reviewer writes that after spending some time issuing voice commands, he can unequivocally say it's much-welcomed functionality(Hinkle, 2012).

The first addition he addresses is the ability to yell *Skyrim*'s "dragon shouts" both in English and Dragonspeak. While he says it is neat at first, and that being able to control the intensity of the shout by speaking as many words of the shout as you'd like is handy, but that novelty wears thin pretty quickly. Ultimately it becomes a utilitarian function.

While he mentions that Kinect's voice recognition capabilities aren't the best – even Bethesda's Pete Hines had a bit of trouble having his voice issues recognized. But for the most part he thinks the technology is sound.

In his conclusion he says that *Skyrim* benefits greatly from its new Kinect support. Being able to have different weapons for different situations and switch between them with a simple phrase is a great benefit. The same with having the ability to switch between shouts and spells on the fly, simply by uttering a few words. Most importantly, he says it's an update that values your time and helps you waste less of it (Hinkle, 2012).

IGN

The review at IGN is organized in sections based on what they "love", "like" and what was "meh" (Hopper, 2012).

What the reviewer loved about the Kinect update for *Skyrim* was how easy it was to organize. He writes that getting through menus was super easy and felt natural, and he believes this is likely going to be the most used and well-received of *Skyrim's* Kinect feature set (Hopper, 2012).

The reviewer writes that he liked the ally commands, which makes it easy and intuitive to interact with your ally. Although he points out that he feels it is a tad impersonal having to refer to your ally as "ally" instead of a name. One of the things he liked really well with the ally commands, were being able to initiate trades with them, and using the inventory organization options, which he found was a really handy way of going about it.

Another thing the reviewer liked was the Dragon Shouts, which he finds very entertaining. Although he points out that since Dragon is not the Kinect's native language, you have to really enunciate your Shouts in order for them to be registered correctly.

What the reviewer was not as impressed with was how cumbersome it was to assign spells and equipment through hotkeys. You have to favorite them, then assign them, then equip them to get them to register properly, which was rather a mouthful. The reviewer predicts that many gamers will likely just default to the standard control schema instead for that.

In conclusion the reviewer writes that as a player you likely won't use most of what's being implemented. However, there is still plenty to excited about in order to extend the Skyrim experience further (Hopper, 2012).

5.1.4 Steel Battalion: Heavy Armor

Steel Battalion: Heavy Armor was developed by From Software and published by Capcom. The game was developed for Xbox 360 and it is a sequel to *Steel Battalion* and *Steel Battalion: Line of Sight*. While the previous games in the series used a large controller with two joysticks and over 40 buttons, this game uses a combination of Kinect and the standard Xbox 360 controller (Capcom, 2012).

The game takes place in 2082 in a world without computers, because of a microbe that started destroying all microprocessors in the world. It opens with the United States military running a landing operation on an occupied New York using massive robots known as Vertical Tanks (VT) as the primary attack force (Capcom, 2012).

Figure 6: Steel Battalion: Heavy Armor tutorial

The game is played sitting down, using the standard controller to navigate in the game, as well as a couple of other actions, such as shooting. Along with the story, the player uses Kinect to perform certain aspects, such as shake another soldier's hand or to catch an apple that is thrown to the player. The player use mimetic movements to perform these story actions.

Figure 7: Steel Battalion: Heavy Armor tutorial

The majority of the game is situated in a VT the player is controlling along with three other characters. In order to speak to each one, the player uses his arm to swipe sideways in the direction he wants to turn. Inside the VT, there are a number of levers and buttons for the player to control, all of which are controlled by physical movements that imitates what you want to do.

Figure 8: Steel Battalion: Heavy Armor tutorial

For instance if the player wants to pull on the lower right corner, the player needs to put his arm out to the right at the appropriate height and pull back. The player is controlling the navigation of the tank by using the standard controller, while shooting with the VT's weapons by pushing a button. In order to see properly, the character has to lean forward to look through a glass, which the player does by putting both his arms forward. When the player wants to lean back he does the same movement. The player can also use a periscope view, which is done by raising his arm and pulling on a periscope. In order to zoom out of the periscope view, the player raises his arm again.

While in combat, there are multiple incidents that can occur, such as the cockpit filling up with smoke. If that happens the player needs to ventilate the cockpit, which is done by extending the right arm, grab a control panel and yank it back before yanking a handle back and downwards. Sometimes the player needs to look out for enemies from afar, which is done by looking out of a roof hatch. In order to do this the player stands up from their seat, and if they need to see even further the player raises his hand to his forehead, which brings up binoculars for the character.

Because the control schema for *Steel Battalion: Heavy Armor* is so extensive, I have only included a couple illustrative images in Figure 6, Figure 7 and Figure 8. The entire control schema is included in the data collection section in the appendix Image A 8: Control schema for Steel Battalion: Heavy Armor.

Gamespot

The review at Gamespot starts with describing the concept and the positive experience he had with playing *Heavy Armor*.

> *"Steel Battalion and the Kinect want you to feel like you're in the cockpit of your VT, so you reach up to pull down the periscope, stretch your arm forward to press dashboard buttons, extend out to the side to pull out the map, and so on. This is a cool concept, and the moments when you feel a strong connection to your cockpit are rewarding. More often, however, the link between you and the action is tenuous at best." (Watters, 2012).*

As can be seen in the quote above, the reviewer does like the concept of the game and he did have some rewarding moments. He further writes that these moments give you glimpse of what the game could have been, but unfortunately there are too many flaws in the game for it to be truly enjoyable. He also writes that the flaws are of the insidious kind, the kind that make you think you have a chance to master them, but you don't (Watters, 2012).

IGN

The reviewer writes that while the original *Steel Battalion* was an impeccable, hardcore giant-robot sim that gained instant infamy for its amazing immersion-boosting control system, *Heavy Armor* is just plain hard. Though he things that it too will probably live on in "Xbox lore" but unfortunately for all the wrong reasons. He writes that the awful, inaccurate Kinect controls only make the bad situations worse, leaving Heavy Armor as a great idea fatally crippled by its own technical and design failings (McCaffrey, 2012).

He hypothesizes that the reason why *Heavy Armor* might be Kinect's most spectacular failure, is because it is so ambitious. He further writes that the game has tried something genuinely new, by building a hardcore game around Kinect, but also fully committing to do so in an immersive way. And because he genuinely wants the game to be fun and functional, it becomes even more disappointing(McCaffrey, 2012).

He goes on to say that *Heavy Armor* deserves a lot of credit for trying to craft a fresh videogame experience, and that while he believes it to be one of the best ideas on Xbox that year, it ultimately became one of the system's worst games (McCaffrey, 2012).

The Escapist

"*No one wants to play a broken game, but it's worse if you can catch glimpses of a brilliant concept buried beneath all the muck and filth*" (Tito, 2012).

The reviewer at the Escapist writes about the gameplay concepts, but continues to explain that unfortunately those concepts don't work in practice. Raising your hand to grab the periscope works only half the time, otherwise you mistakenly zoom to the viewport or ineffectually wave your character's hands on the screen (Tito, 2012).

While the reviewer thought that using Kinect to grab panels and pull them towards you was a great way to place more controls at your virtual fingertips, but this also means that you have to convince your Kinect to complete not one but two - and sometimes more - actions in sequence to get anything done. And that reduces your chances of success to "nil" (Tito, 2012).

One of the things the reviewer wonders about is why the designers didn't map at least some of the controls on the Xbox controller, since you already use it to steer and fire. "*Would it have killed them to use Y to toggle between armor-piercing rounds and normal shells?*" (Tito, 2012).

Game Informer

"*Steel Battalion: Heavy Armor is as fun as fumbling around for your keys in the dark while being pelted with billiard balls" (Cork, 2012).*

For the reviewer at Game Informer, playing *Heavy Armor* left him feeling frustrated and completely drained. He calls it a showcase for the worst that Kinect has to offer, where nearly every gesture is either ignored or misinterpreted – often with game-ending repercussions. Although he does admit that it is a neat concept (Cork, 2012).

The reviewer gives a harsh judgment over the game, writing that it is a technical failure, and that it has the worst motion controls he has ever experienced, on Kinect or anywhere else. He writes that he, like any reasonable person, expects a baseline degree of functionality when he plays a game. Just because a game uses Kinect, players shouldn't dilute their expectations. Nobody would rally around a game that registers a button press 25 percent of the time, and you shouldn't do so here (Cork, 2012).

Joystiq

The review starts by describing the reviewers experience when playing *Heavy Armor*.

> *"As my team and I regroup, a second shell rocks the cabin once more, shattering the viewport glass and filling the cockpit with smoke. The panel that houses the ventilator control is tucked away to the right. I pull it forward, lock it into place, reach for the ventilator chain and ... accidentally flick on the headlights. On my second attempt, instead of pulling the ventilator chain, I put the entire control panel away again. I pull the panel out again, reach for the ventilator chain and put the control panel away again. I pull the panel out a third time and reach very, very carefully for the ventilator chain ... at which point my entire crew dies of smoke inhalation"*(Mitchell, 2012).

The reviewer writes that the technology behind the Kinect is capable of enabling incredible fantasies, and that it has allowed him to interact with virtual worlds in ways he never thought possible. What *Steel Battalion: Heavy Armor* has taught him is that, should that technology fail just *once*, the entire fantasy comes crashing down in an instant (Mitchell, 2012).

When everything in the game was running smoothly, the reviewer found the deliberate actions to be deeply rewarding, but the second that an action fails to execute as intended, the entire experience broke down. He writes that Kinect really does enhance the experience of the game, enabling the fantasy of piloting a very real walking tank and delivering a thrill of satisfaction with every confirmed kill. But when the tech fails, for him it failed in such a fundamental way that is was impossible for him to ignore (Mitchell, 2012).

5.2 Data collections

The data collection for this case study includes video games, video game reviews and control schema. The video games were chosen based on their score from Metacritic, except for *Steel Battalion: Heavy Armor* as mentioned in section 4.4 Multiple-case design. The video game reviews I chose from specific gaming websites, in order to get a similar quality on the reviews between the games. The control schema were made from videos on Youtube or found on gaming websites.

Video games

- BioShock: Infinite
- Kung Fu Rider
- The Elder Scrolls V: Skyrim
- Steel Battalion: Heavy Armor

Game reviews

- IGN
  - BioShock Infinite
  - Kung Fu Rider
  - The Elder Scrolls V: Skyrim
  - Steel Battalion: Heavy Armor
- Joystiq
  - Kung Fu Rider
  - The Elder Scrolls V: Skyrim
  - Steel Battalion: Heavy Armor
- The Escapist

- Steel Battalion: Heavy Armor

- Gamespot
  - Steel Battalion: Heavy Armor
- Game Informer
  - Steel Battalion: Heavy Armor
- PushSquare
  - BioShock: Infinite

Control schema

- BioShock: Infinite (Decayedmatter, 2013)

  Appendix: Image A 1: Control schema for BioShock: Infinite

- Kung Fu Rider (Defendini, 2010)

  Appendix: Image A 2: Control schema for Kung Fu Rider

- The Elder Scrolls V: Skyrim (Bradley, 2012)

  Appendix; Image A 3: Control schema for The Elder Scrolls V: Skyrim

  Image A 4: Control schema for The Elder Scrolls V: Skyrim

  Image A 5: Control schema for The Elder Scrolls V: Skyrim

  Image A 6: Control schema for The Elder Scrolls V: Skyrim

  Image A 7: Control schema for The Elder Scrolls V: Skyrim

- Steel Battalion: Heavy Armor (VIDEOS, 2012)

  Appendix: Image A 8: Control schema for Steel Battalion: Heavy Armor

5.3 Analysis

First I will look at each of the video games, and analyze them using the criteria from the GameFlow model; *concentration, challenge, player skill, control, clear goals, feedback, immersion,* and *social interaction* (Sweetser and Wyeth, 2005). I will also analyze them according to their level of p-actions (Gregersen and Grodal, 2009) and to what degree the control schema supports the four degrees of natural mapping; *directional natural mapping, kinesic natural mapping, incomplete tangible natural mapping*, and *realistic tangible natural mapping* (Skalski et al., 2011).

BioShock: Infinite

If we look at the game *Bioshock: Infinite*, we can see that it has a simple and intuitive control schema. It uses a joystick to navigate in the game, which supports *directional natural mapping* and it uses standard button configurations for a PlayStation game. This supports *player skill*, and makes the game easy to learn.

The game uses the Move controller in a way that mimics a gun, letting the player hold and aim with an object, and shoot by pressing a button with the left index finger. This is basically the same thing children do when pretending to be a cowboy, and is familiar to most people. Using the controller in this way supports *incomplete tangible natural mapping,* and gives it a high degree of p-actions, both of which raises the player's feel of embodiment while playing.

It also supports *player skills* because as players we do not need to learn how to point and shoot*, control* because the controller gives the player a high sense of control over their characters movement and interaction and *immersion* because it makes the player viscerally involved in the game, in addition to making the player less aware of their surroundings.

The reviewer concludes that the game was very enjoyable to play with the Move control schema, and that it felt satisfying to line up shots with the Move (Barker, 2013). This concurs with the statements in the previous paragraphs, that the reviewer felt immersed and had a positive experience based on it supporting *incomplete tangible natural mapping* and having a high degree of p-actions.

Director of *BioShock: Infinite* Ken Levine (IGN, 2013) stated that they wanted the game to be playable for all gamers by making a control schema that fit with both DualShock 3 and PlayStation Move. This they have achieved, but at the same time they could have taken further advantage of the possibilities Move offered. As it is now the game is perfectly functional, but as the reviewer at Joystiq stated; it isn't spectacular. The reviewer proposes that they could also have used the Move's motion-sensing technology for simple gestures such as pulling levers (Barker, 2013). If the developers had done this, they would have further increased the degree of p-actions, and given the game more *incomplete tangible natural mapping*.

Kung Fu Rider

Unlike *Bioshock: Infinite*, *Kung Fu Rider* has a control schema that is difficult to use, and difficult to remember. In order to play the game the player has to go through an extensive tutorial which explains how to control all the mechanics. This supports the flow element *player skills*, because the tutorial has been made to feel like an actual level. Unfortunately at the same time it contradicts the same element, because the control schema is still difficult to remember. This is because many of the actions on the screen do not resemble the actions you do with the controller; for instance we normally do not move our arms up and down to walk forward. This does not support natural mapping, nor does it give a high degree of p-actions. Both the reviewers at Joystiq and IGN pointed out that the control schema was not easy to neither use nor remember.

As previously mentioned, the control schema is also difficult to use. We usually don't move our arms up and down, while at the same time moving our arms sideways, which makes it unnatural and difficult to both accelerate and steer. This can lower the player's sense of control over the characters movements, which contradicts the flow element *control*. The same applies for flicking the controller upwards to jump, because we are already doing that same movement, only not as extreme, in order to move forward. This particular problem was also pointed out by the reviewer at IGN, who had problems with the game misinterpreting his actions. This leads to detrimental errors in the game which also contradicts *control*.

Another thing that makes the controllers difficult to learn, is that while jumping is flicking the controller up, flicking the controller down does not make the character duck, as would be natural to conclude. Instead the player needs to press a button to do that. This could also be said about using a button to kick, when the movement in the game is similar to slashing the controller from one side to the other. However I do not believe that implementing these solutions would have improved the game, because it would only increase the difficulty even more. It does however make the decision to use physical movements or buttons seem somewhat random. This ensures that the game has a low degree of p-actions and that there is very little natural mapping. The exception is flicking the controller up to jump, thrusting the controller forwards to quickly accelerate and steering by moving the controller left or right. These three actions give the game a low degree of p-actions, and supports *directional natural mapping*. Unfortunately these low degrees of immersion are sabotaged by the other actions in the game.

The Elder Scrolls V: Skyrim

One thing that separates *The Elder Scrolls V: Skyrim* from the other games studied in this case is that the game was not initially developed to support Kinect; instead this functionality was added later on. The developers therefore had to think about the game as it already existed, and figure out which mechanics would benefit from Kinect support, and which would not. When it was announced that *Skyrim* was going to get the Kinect update, Pete Hines, the VP of marketing and PR, said that replacing any combat with physical actions in front of a Kinect sensor wouldn't make a lot of sense. "Doing combat and swinging your sword and all that stuff without a controller would dramatically change the game. We're trying to enhance the experience as we designed it with some additional functionality" (Gilbert, 2012). I believe that this decision was a very smart decision to make, seeing as many different functionalities would be affected if they decided to change the controller configurations in an already established game, that has already been developed with a certain control schema in mind.

The decision to include voice-commands with Kinect support made the game more effective, and added welcomed functionality according to the reviewers. Because it made it easier for the player to navigate the game's interface, this supported the flow element *player skills*. And by using certain command words, they greatly lessened the possibility of the player making detrimental errors in the game.

It also supported the flow element *immersion*, because the player becomes emotionally and viscerally involved in the game, when they can actually Shout out commands and "talk" to the Interface, for example by asking "Where am I?", and get the answer shown to you on the map. Also, using the voice commands to control your ally, supports *social interaction*. However as the reviewer at IGN points out, the immersive feeling was somewhat lessened by having to address the ally as "Ally" instead of using their name. The game also increased its p-actions when the player can shout out and their character Dragon Shouts the same, which leads to the game having an increased sense of embodiment. This also gives the highest degree of natural mapping; *realistic tangible natural mapping*.

Steel Battalion: Heavy Armor

The last video game, *Steel Battalion: Heavy Armor*, tried to do something which could potentially be very interesting when they decided to use a combination of the standard Xbox 360 controller and the Kinect, but unfortunately the idea was better than the experience. This is something that concurs with the several reviewers who stated that they really liked the concept, and wanted the game to be enjoyable.

First I want to talk a bit about what works well in this game, which is the controller combination. By using the standard controller to navigate in the game, the developers supported the flow elements *player skills* and *control*. It also gave the game *directional natural mapping*. Anyone who has used a standard Xbox 360 controller before, can jump right into the game and play it, and it lets the player feel a sense of control over the movement and interactions of their character. This is also true for the use of physical movements for the simple actions the player does in the beginning of the game; such as shaking another characters hand, and catching an apple. We already know how to shake someone's hand, and how to catch something that is thrown to us. Using the Kinect for these actions also gives the game a higher degree of p-actions, compared to using the standard controller for the same actions, which leads to a greater sense of embodiment. It also gives the game *kinesic natural mapping*. The reviewers also reported that they did have some moments of great enjoyment and feelings of immersion.

Even though the game has a couple of good features, there are too many bad features that outweigh them. When we take into account that this game is part of a series that earlier utilized a controller developed specifically for the game, see Image 1, I think this can help explain what went bad with the development. In *Heavy Armor* the player is controlling a VT, just as the player did in the previous games in the series, so the developers had to "translate" the control schema. When the old controller had over 40 buttons, while the Kinect have no buttons, problems arose.

Image 1: Controller developed for Steel Battalion.

When you have several buttons to press, the physical action does not change much, but the controller has no problem figuring out what you are trying to do, because it only cares about what button was pressed. Kinect on the other hand has to interpret what you are physically doing in order to figure out your intent, and when the physical movements for different actions are very similar this can lead to the game misinterpreting your intent. And this again causes detrimental errors while playing, which contradicts the flow element *control*. And when the player experiences results in errors on most of the actions they are trying to perform, it destroys the player's sense of control. In addition the player's sense of embodiment is lowered, because the game no longer has any p-actions due to the characters actions are no longer corresponding with your own. This also ruins any natural mapping in the game.

When the result of the game misinterpreting your actions can often lead to the player's character dying, this disrupts the flow element *immersion* because the players experience isn't effortless, rather it requires a lot of effort, and it might still result in the player loosing. This problem was reported by all reviewers, and was the main problem they had with the game.

5.4 Findings

Two of the research questions in this thesis are "do different input-technologies work better for different types of game mechanics?" and "how can input-technologies affect the success/enjoyment of video games?". After studying and analyzing the video games in section 5.1 Cases, the findings from this case study mainly suggest that if video games support a high degree of natural mapping and p-actions, they lead to a higher sense of embodiment. However, incorporating a high degree of natural mapping and p-actions should not happen in spite of GameFlow elements, especially *control* and *player skills*. I suggest that controllers which support a high degree of embodiment seem to be able to give a better PX, as long as they follow the criteria of GameFlow, in particular *control* and *player skills*. This suggests that there is a difference between input-technologies, and that they work better for different types of game mechanics. And because the controllers with the highest degree of embodiment also seem to be the controllers that can disrupt those GameFlow elements the easiest, developers need to find the correct balance between *immersion* and *control*.

What is common between all games, were that the reviewers enjoyed the concepts, unfortunately that was not enough to make the game enjoyable. This was especially the case with *Steel Battalion: Heavy Armor*. As the reviewers pointed out, it was a game they genuinely wanted to be good, and when it worked they reported a high degree of immersion and enjoyment. This supports the findings from McEwan et al. (2012), where they found that games with a high degree of natural mapping gave good PX. Although both McEwan et al. (2012) and Gerling et al. (2011) found that poorer performance from the players did not affect their PX, this was with games that did not have the same degree of issues and faults as does *Steel Battalion: Heavy Armor*. From their findings we can assume that if *Steel Battalion: Heavy Armor* had only had smaller problems with the controls, their game would still have offered the players a high degree of immersion, and perhaps their problems would have been forgiven. But this was not the case. As the reviewers pointed out, the errors were of such a degree that their feeling of immersion and control was stripped away, and this is what in the end caused the game to be so unsuccessful as it is.

When we compare the two successful games in this study; *BioShock: Infninite* and *The Elder Scrolls V: Skyrim*, we can see that one thing they had in common was that they went for functionality and did not try to reinvent the wheel so to speak. The reviewers for both games mentioned this, the reviewer for *Bioshock: Infinite* even pointing out features he would have liked to see. However, what was incorporated into the game worked exactly as it should, and it gave the games a higher sense of embodiment, which in turn led to the games being slightly enhanced from what they would have been without the controllers. We can also see that this was the intent from the statements from Ken Levine and Pete Hines, they both wanted to add something, but not reinvent the game.

While I think this case study has given an insight into whether or not different input-technologies work better for different types of game mechanics and how input-technologies affect the success/enjoyment of video games, I still do not have an answer to the last of the research questions; how can game developers best utilize input-technologies in video games? In the next section I have used the findings from this case study and tried to come up with a possible answer.

6 Comparative analysis of controllers and mechanics

6.1 Process

In order to try and answer how game developers can best utilize input-technologies in video games, I will first go over the theories from the theoretical framework, and how they can be of help in constructing a conceptual model. I will then do the same with the findings from the multiple-case study. In the end I present a comparative analysis and a conceptual model.

6.2 Theories and findings

In order to construct the conceptual model, I will take into account theories of embodiment and distributed cognition; more specifically I will use the GameFlow model and look at natural mapping and p-actions in much the same way as I did in the multiple-case study.

The findings from the multiple-case study lean towards there being a correlation between the degree of embodiment the controllers support, showed by how much they support natural mapping and p-actions, and how enjoyable the game is. It was also found that the most enjoyable games supported the criteria for GameFlow. Therefore I will take this into consideration when crafting the conceptual model.

At the same time the findings from the model also suggests that a game with high degree of embodiment can also fail miserably if it is difficult to use or if it destroys the player's sense of control. The balance should then lie in trying to gain the sense of embodiment, but only when it is easy and suitable for the game mechanics, and follows the criteria for GameFlow.

When developing the conceptual model, I will take into account the findings mentioned in the paragraphs above, and ensure that the combination of technology and mechanics should support p-actions, natural mapping and GameFlow elements. Taking specific care that the GameFlow elements *control* and *player skills* should not be hampered. This is especially important to take into consideration when looking at technologies that are not represented in the multiple-case study, such as artifact, gesture, biological and photographic. In order to better clarify the grounds for those technologies, I will in some cases use examples of other games that use the corresponding technology.

6.3 Comparative analysis

In order to construct the conceptual model based on gaming controllers and game mechanics, I first had to define and categorize the two. First I look at different types of input-technology that are usable in games today, and divide them into proposed input-technology types. I then look at different types of core game mechanics that can be used in games, and divide them into different core game mechanic categories.

6.3.1 Input-technologies

I first looked at the different control devices that are associated with the today's console generation; the dual-axis controller, Kinect, Wiimote and PlayStation Move. I also looked at computer mouse and keyboard which is also used extensively in gaming. By focusing on the technologies they incorporate instead of the actual control devices, I came up with these categories; abstract, physical and audio.

I then looked at some other control devices that can also be used for gaming. These included the Emotiv EPOC headset, muscle sensors, sweat sensors, leap motion, steering wheel, tablets and mobile phones. After looking at these devices I came up with some additional categories; biological, artifact, virtualization and gestures.

If we look at the quote from Freeman et al. (2012) in section 3.3.5 Gaming controllers, we can see that he mentions different controllers and divides them as following; handheld motion input devices, tangibles, touch, in-air gestures, and the whole body. Those terms match closely with the technology categories I propose, where handheld motion input devices and tangibles are artifacts, touch and in-air gestures are gestures and the whole body is physical. Although I focus on the technology and do not divide based on the actual controller, and I have also included the standard controllers as abstract, in addition to biological controllers and virtualization.

In the end I came up with 7 different input technology categories, and will give a description of these below.

1. Abstract – responds to standardized objects with most often abstracted actions
2. Artifact – responds to artifacts made especially for mimetic behavior
3. Physical – responds to physical input
4. Gestures – responds to finger and hand gestures
5. Biological – responds to biological input
6. Audio – responds to input of sound
7. Virtualization – responds to real-life object input

Abstract: This category includes controllers such as the dual-axis, as well as mouse and keyboard. What characterizes this technology is that the control devices have been standardized and are therefore used for virtually all games. The other thing that characterizes them are the abstracted actions, by which I mean that the act of e.g. swinging a sword or picking a flower have been abstracted into an action where the player presses the "X" button or the left mouse-button. Although this will almost always be abstracted, sometimes they are not. Examples of this can be with the old text-based adventures, where the player types in words by using the keyboard, or if a player has to press a button in the game where it says "X", and he does so by pressing the "X" button on his controller.

Artifact: This category includes controllers such as steering wheels and pedals, joysticks and the guitars used in guitar hero. What characterizes this technology is that the controllers are made to resemble regular objects in a way that lets the player mimic the action done on the screen. While the player may not be able to play a guitar or pilot an airplane, he can pretend to do so by using the guitar controller from guitar hero or the joystick in a flight simulator program.

Physical: This category includes controllers such as the Kinect camera, the Wiimote and PlayStation Move. What characterizes this technology is that the controllers let the player's body control the actions in the game.

> **Gestures:** This is a subcategory of physical technology , and the reason why I included it is because although these controllers are used with physical input, using our hands gives us more options and is both quicker and often easier than using our entire body. This again means that we can do more detailed and diverse actions when using gesturing, than with full-body actions. The controllers typically found in this category are touch screens or smaller motion sensing devices such as the Leap Motion.

Biological: This category includes controllers such as EEG-headsets and muscle-sensors. What characterizes this technology is that the player's own biological input, such as level of concentration or whether or not the player's body is flexing or relaxed, is controlling the game.

Audio: This category includes controllers with microphones such as headsets or Kinect. What characterizes this technology is that they are controlled by using sound, such as talking, shouting or whistling.

Virtualization: This category includes controllers such as the Kinect, the PlayStation EyeToy camera and the Near Field Communication (NFC) technology in Wii U. The characteristics of controllers with this technology are that they use the camera or another technology, so the input is images of the player and its surroundings, or scanned items.

While some controllers are strictly in only one category, some controllers can belong to several. One example of such a controller is the Wii remote (see Figure 9). Because the player will be using its body by e.g. swinging the remote, it belongs to the physical category, but for some games the controller belongs to the abstract category, because it has a standardized setup of buttons you have to use. You also have the possibility to attach items to the controller, which will transform the controller from a mere tool into an artifact, like e.g. turning it into a tennis racket or a golf club.

Figure 9: Wii Remote for different controller genres

6.3.2 Game mechanics

As mentioned in section 3.3.4 Game mechanics I am using the definition of Sicart (2008), which states that mechanics are methods invoked by agents for interacting with the game world. He also distinguishes between core, primary and secondary mechanics. For the purpose of the conceptual model I will only focus on the core mechanics, because it is the most basic type of mechanic, and is therefore more suitable to be compared directly to input controller. I could also have included primary mechanics, but I think those mechanics may be too broad to effectively be used in this model. For instance in *Steel Battalion: Heavy Armor* one primary mechanic is to control a VT, but this is done using multiple technologies.

Because the conceptual model should be usable for all types of games, I wanted the different categories to be as broad as possible. After looking at the games in the multiple-case study as well as different games I have played over the years, I ended up with 5 possible categories for game mechanics.

1. Action
2. Navigational
3. Interface
4. Communication
5. Affective

Action: when the player performs actions in the game. These actions are deliberate and will often be associated with the overall goal of the game, such as hitting a ball in order to win a tennis match, fire a gun in order to kill the zombie, or pick a flower in order to sell it at an auction house.

Navigational: when the player navigates around the game world, such as walking forwards, backwards, jumping, crouching etc. This applies both to game cameras as well as game avatars.

Interface: when the player interacts with the game's user interface, such as menus, settings or dialogue boxes.

Communication: when the player communicates with other players in the game, or when their avatar communicates with other in-game characters.

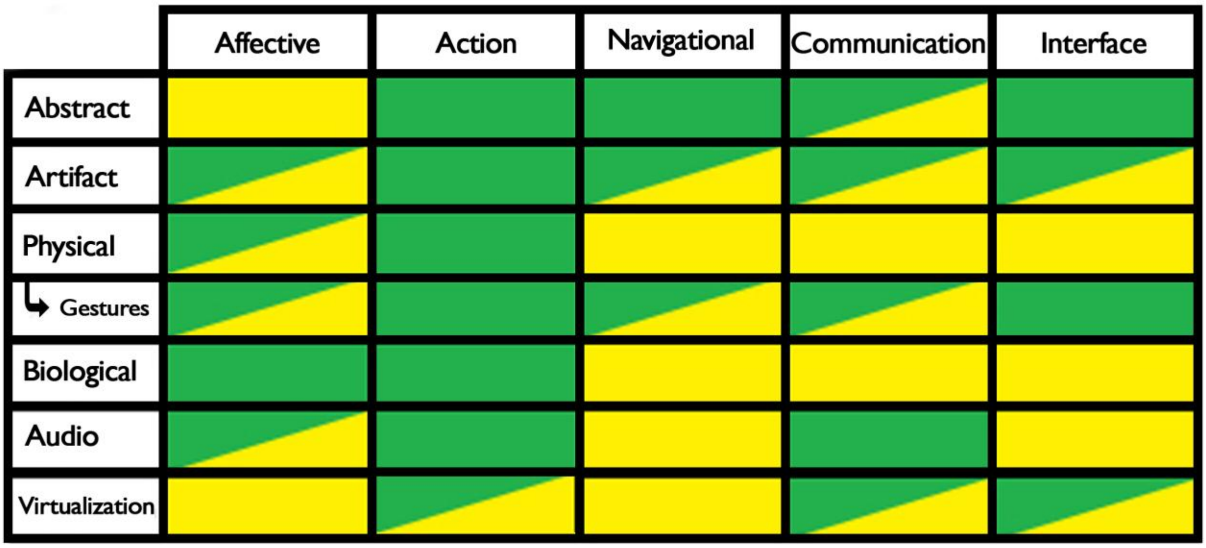
Affective: when something affects the game state. This can for instance be if your stress level is affecting how difficult the game is, or if sweat sensors are controlling the in-game lighting etc.

While these game mechanic categories may be too simple to be used as a formal categorization of core mechanics, I believe that they are descriptive and simple enough to be used for the purpose of this thesis.

6.3.3 Conceptual model: input TECHnology and game MECHanics (TechMech)

In this comparative analysis I look at how some game mechanics will work with different game controllers, and explain why they are compatible or not. In Table 1 I present which mechanics I propose will work well with which controllers, where the green is most suitable and yellow is less suitable. Where there are both colors it means that the controller can be either, based on the specific controller.

Table 1: Comparison of game mechanics and game controllers: TechMech model



6.4 Analysis

In this section I will explain each controller technology against the different game mechanics, and give an explanation for my decisions based on the theories and findings I mentioned in section 6.2 Theories and findings.

Abstract: Because the abstract controller is made to be standardized and efficient, it is the most versatile controller, and therefore the one which is most suitable to most of the game mechanics. The one mechanic it differs in is the affective. Because all the input it provides is from a deliberate act, it is difficult for this controller to affect the game in an unintentional way. Also it depends on the specific controller how suitable it is for communication, for instance is the keyboard perfect for communicating, but using the dual-axis controller is much slower and is therefore not as suitable for instance in-game. This type of controller has a low degree of p-actions, where it supports *directional natural mapping*, so it does not support embodiment as much as other technologies, but instead it is a very effective controller that is well suited for controlling multiple mechanics. This supports the flow element *control* by giving the player a high sense of control over their characters. If we look at how the Xbox controller was used in *Steel Battalion: Heavy Armor*, we can see that none of the reviewers mentioned any problems with navigating the VT, but at the same time they did not mention the controller giving them the enjoyable immersive moments either. One reviewer pointed out that he could not understand why they did not map anything on to the Xbox controller, suggesting that he thought the game would have been easier to control if they had.

Artifact: Because these controllers will often vary according to what game it is used for, it is difficult to place these according to game mechanics. For some games the controller might be very useful for e.g. affective mechanics by including a gyro sensor or for navigational mechanics by replicating a steering wheel. Because these controllers will vary in such a large degree from each other, the same does their suitability for the different controllers. But what they have in common is that because they are usually made specifically for each type of game, they are therefore also very suitable for the actions they do in those games. These controllers have a high degree of p-actions, because it mimics the activity on screen with similar motions and appearance. They have the highest degree of natural mapping because they give *realistic tangible natural mapping*. These mimetic behaviors support the flow element *immersion* because the players become less aware of their surroundings, and it supports feeling viscerally involved in the game. However this is assuming that the artifact controller is used for the game it has been developed for.

Physical: Because physical controllers rely on the player's body and movements, they are also dependent on the player's physical health, such as stamina, in addition to the player environment. They can therefore be a lot more tiring than e.g. the abstract controller, which can in turn disrupt the flow element *control* and *player skills*. This often makes them less suitable for game mechanics such as navigation, communication or interface.

If we look at the review from IGN on *Kung Fu Rider*, we can see that the reviewer complained about getting tired from continually having to flick the controller up and down in order to move forward.

On the other hand physical controllers give most p-actions, which support a high degree of embodiment. They also support natural mapping by giving *incomplete tangible natural mapping* such as with the Wiimote and PlayStation Move, or they can give *kinesic natural mapping* such as with Kinect. An example of a game that has done very well using the physical controller Kinect (and with gesturing on smartphones) is the award winning *Fruit Ninja Kinect* developed by Halfbrick Studios (Halfbrick, 2013). Here the developers have made a game which uses the Kinect controller in a simple and easy, but very immersive way. The game uses one action mechanic, slicing in the air, which are controlled by the player physically slicing the air with their arms and legs, which actually supports *realistic tangible natural mapping*. In addition to supporting a high degree of embodiment, physical controllers support *player skills* because they are easy to learn and use, since we are already familiar with our own body. They can also support *control* because the player has control not only of the interactions of the characters, but they can also to a very high degree control the character's movement. In addition these controllers support *immersion* much as the artifact controllers, because the player becomes less aware of their surroundings and feel viscerally involved in the game.

While physical controllers support several flow elements, developers need to take extra care, because they can also hamper the *control* element, which can also ruin the *immersion* element. An example of how physical controllers can destroy flow is from the game *Steel Battalion: Heavy Armor*. There Kinect is controlling a lot of similar actions, and the game often misinterprets what the player is trying to do. This leads to the player feeling that they do not have a sense of control over their characters or over the controller, and they are left being very aware of their own surroundings, which do not correspond to the surrounding of the character. This was perhaps especially true in the case of *Steel Battalion: Heavy Armor* where the previous games in the series had an artifact controller for the game.

Gesture: Although a lot of the principles which apply for physical controllers also apply for gesture controllers, there are some key differences. For instance we do not tire as quickly when using just our hands as we do when using our entire body, and we are quicker with our hands as well. This makes gesture controllers more suitable for navigational, communication and especially interface mechanics, than regular physical controllers. Gesture controllers would for instance work well with navigating maps and levels, and it could work well with communication by using sign language or military hand signals. Although when used for more typical communication or navigational mechanics, gesture controllers would eventually also suffer from the same issues as physical controllers. The one mechanic where gesture control differs most from physical control is with interface. Because of the use of touch devices in our society, there are already developed standards on how to use gesturing to interact with interfaces (most notably with touch screens, less so with for instance leap motion), so it would work well with this in games as well, which increases the flow elements *player skills* and *control*. Generally speaking the flow elements and p-actions are the same for gestures as with physical controllers.

Biological: Biological controllers have a similar dependency on the players as does the physical controllers; this is because they depend on the player's stamina. Because of this they too are less suitable for navigational, communication and interface mechanics, which requires more efficiency and repetitive actions than affective and action mechanics. They can also be more difficult to use, because we are not used to interacting with technology using for instance our mind, controlling our emotions or blinking. But these controllers are very suitable for affective mechanics, and can be used to give a more immersive experience. McCallum (2013) proposes using sensors to detect our stress level, and use this to customize the player experience by for instance reducing or raising the difficulty level. The game company Steam is apparently also looking at using biological controllers for affective mechanics in games (Gizmodo, 2013), so this might be something we will see a lot more of in the future.

Biological technology controllers have a low degree of p-actions, although this can arguably be higher when used for action mechanics. It can in some cases give *kinesic natural mapping* or *realistic tangible natural mapping*, for instance if they are supposed to bend a fork with their mind, however this will be very dependent on the specific game. When these controllers are used for action and affective mechanics, making the game read the player and adapt accordingly, they support *immersion* by making the player emotionally involved in the game. Affective mechanics with biological controllers would also support the element *feedback* by giving instant feedback on their emotions. Developers should however be aware that biological controllers can make the game more difficult to control and it could be too challenging for players which could ruin the *control* and *challenge* flow of the game.

Audio: Because audio controllers do not have that much variety in the kinds of input, they are not very suitable for navigational mechanics, which often requires several types of input, such as a directional pad or mouse. Also it takes longer to for instance say "forward" than it does to just press a button, and that time might make the difference whether a player escapes or dies. McCallum (2013) says "click the button and you win the game (…) people using the gestures always lose.", and this can be applied to audio controllers as well as gestures. In those cases the flow elements *control* and *player skills* is being hampered, because the player has less control over the game and their opponents, and the controller is unnecessarily difficult. But audio could be suitable for affective mechanics, such as interpreting the tone of the player's voice. It could also be very suitable for action mechanics, such as we see in *The Elder Scrolls V: Skyrim* when the player uses voice commands for Dragon Shouts. This arguably supports a high degree of p-action and gives *realistic tangible natural mapping* which could lead to a more immersive experience. But it does have an issue with being dependent on the player's environment, for instance it does not work well with screaming at the TV if you have a toddler sleeping in the next room.

When it comes to communication, audio controllers are very suitable for especially player-communication, often called voice-chat, but they could also be very suitable for character-communication. Because player-communication is interpreted by humans, it is easier to incorporate, but when the audio has to be interpreted by a machine this is harder to do. But we can see in *The Elder Scrolls V: Skyrim* how this can be done by using simple commands such as "ally follow", and how it is not only more efficient than typing, but it also supports a more immersive experience. Lastly it can also be very suitable for Interface mechanics, such as finding information about the player's equipment or to navigate on a map, which was clear the reviewers of *The Elder Scrolls V: Skyrim* enjoyed.

Audio controllers support the flow elements *player skills*, *control, immersion* and *social interaction*. The same applies as with physical controllers; players already know how to talk so the controller is easy to learn and to use, although a caveat is that the player often needs to be careful of their pronunciation, like the reviewer at IGN pointed out (Hopper, 2012). The controller can also make the player feel a sense of *control* over the character and environment, as well as the game shell, such as in Skyrim by saying "quicksave". The controller can also contribute to the player feeling emotionally and viscerally involved in the game, and it can help support communication both in-game and with other players.

Virtualization: This technology is perhaps the least versatile one, yet it can make a big impact on the immersion of a game, or the enjoyment of it. Because virtualization controllers are so limited in how and what input can be used, the technology is less suitable for affective, navigational and in some cases communication mechanics. When it comes to action mechanic, the controller will in most cases not be very suitable, but for instance in a game where the player needs to find certain items in the house, for instance a shoe or a spatula, the controller could be very suitable. In games where this controller has been used, it has been used with the interface, such as in *Harry Potter for Kinect*[12]. Here the player can take a picture of his or her face, which will then be transformed into the face of their character, see Image 2. So the virtualization controller is suitable for the interface controller in those cases, however it is again limited in its input, so it is less suitable for actually interacting with an interface. The same goes for using a virtualization controller for communication, because it is limited in its input it does not work well with communication, however if the controller has a camera, such as Kinect, and is paired with a voice controller, it can work very well.

[12] EUROCOM 2012. Harry Potter for Kinect. Xbox 360: Warner Bros. Interactive Entertainment.

Image 2: Harry Potter Kinect, face scan

Another controller that falls into this technology is the Wii U, which has NFC technology in the console. An example of how this can used is with the game *Disney Infinity*[13], where the player can buy actual figurines, and by scanning the item with the NFC gets the figurine virtualized into the game to be played with (Zaccari, 2013).

This technology has a low degree of p-actions, yet I argue that it can still lead to a high sense of embodiment, because it can literally incorporate your body and appearance, or that of your favorite figurine into the game. This same feature also supports the flow element *immersion*, because it makes the player emotionally involved in the game. It can also support the element *social interaction* because it can let the players interact while showing the other players their body language.

6.5 Findings

Going back to the research questions, I have in this section tried to address the question; how can game developers best utilize input-technologies in video games? While I am not able to find a definitive answer to this, I propose a solution that could hopefully help developers use different input-technologies to make a great game. By using the constructed conceptual model TechMech, developers should have something they could look to in order to create games that support a high degree of embodiment, but without it affecting the player's sense of control.

Although the TechMech model should be thoroughly tested and verified, I also wanted to test the model in a near real-life environment to see whether or not it could be of any particular help in a game development setting. In the next section I will present the single-case study where I did just that.

[13] SOFTWARE, A. 2013. Disney Infinity. Wii U: Disney Interactive Studios.

7 Single-case study: game development

In this section I will describe the details of the development process of the prototype game using the conceptual model TechMech proposed in section 6 Comparative analysis of controllers and mechanics. I will give a brief explanation for my choices regarding control devices used in the game, and I will present the data collection, in addition to some issues I encountered during development. In the end I will present the analysis and findings from this case.

7.1 Case: Folktale

The main parts of the process were an initial spike, a brainstorming session guided by the TechMech model, creating user stories, and in the end two week iterations of coding until the prototype was finished. I will give a more detailed description of each of the different parts in section 7.2.1 Development process.

The first thing I did after deciding that I was going to do an application description, was figuring out what types of input-technology the game should incorporate. I ended up with using the Kinect, Emotiv EPOC and a regular dual-axis controller. My hope was that by using a combination of these controllers, I would be able to develop a game that used several varieties of input-technology, and that this would make the possible benefits or disadvantages of the conceptual model more apparent. Because of this, the development process was shaped around the controllers from the beginning, starting with the brainstorming session.

7.2 Data collection

Because this case study is an application description, the data collection mainly consists of my own documentation from the development process, which I give an account of in the next section. In addition is the finished control schema for *Folktale* which is included at the end.

7.2.1 Development process

In this section I have documented the development process by giving an account of how I have used the practices mentioned in section 4.5.1 Practices.

Initial spike

During the initial spike, the first thing I did was to set up my technical environments; Unity3D, and the Emotiv Control Panel. I installed an SDK called Zigfu, which is used for Kinect programming (Zigfu, 2012). To get the Emotiv EPOC headset to work in Unity3D, I also bought and installed the Emotiv Epoc Unity3D Developer Support Pack (Emotiv, 2012b).

The other thing I did during the initial spike was to read and watch tutorials online. These were mostly on Unity and JavaScript/C#, but also on Emotiv EPOC and Kinect. Mostly I used tutorials offered by Walker Boys Studio (Walker Boys Studio, 2011) and Berg Zerg Arcade (Berg Zerg BergZergArcade, 2011). I also used the website Codecademy in order to learn JavaScript (Codecademy, 2012).

Game design

In the following paragraphs I will present the different practices used and components that construct the game design.

Brainstorming

The first thing I did in the design phase was to have a brainstorming session with the use of a whiteboard (see Image 3). Because I already knew what controllers my game would use, I divided the whiteboard into categories of each controller. At first I looked at the different controllers and which game mechanic categories that each controller could carry out. Then I added some possible suitable game mechanics. After that I looked at the different controllers, and tried finding specific game mechanics that might be incorporated into the game, such as *swinging a weapon* with the dual-axis and *shooting a fireball* with the Emotiv EPOC. During this phase, I at first tried thinking of different types of actions, and not focusing on a specific genre. However, I soon came up with a concept, and immediately got inspired and found several mechanics that would fit into it. The next thing was then to draw up an overview of the game that should be developed, and in a structured manner decide more specifically how the game mechanics in the game would be, and which controller would be performing them (see Image 4). At this point I also started thinking about the name of the game; and came up with *FolkTale*.

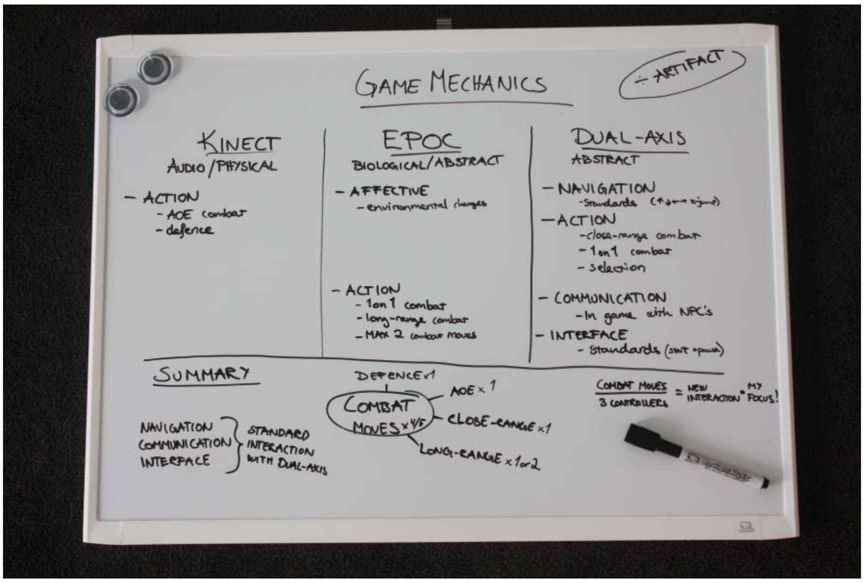


Image 3: Initial Brainstorming

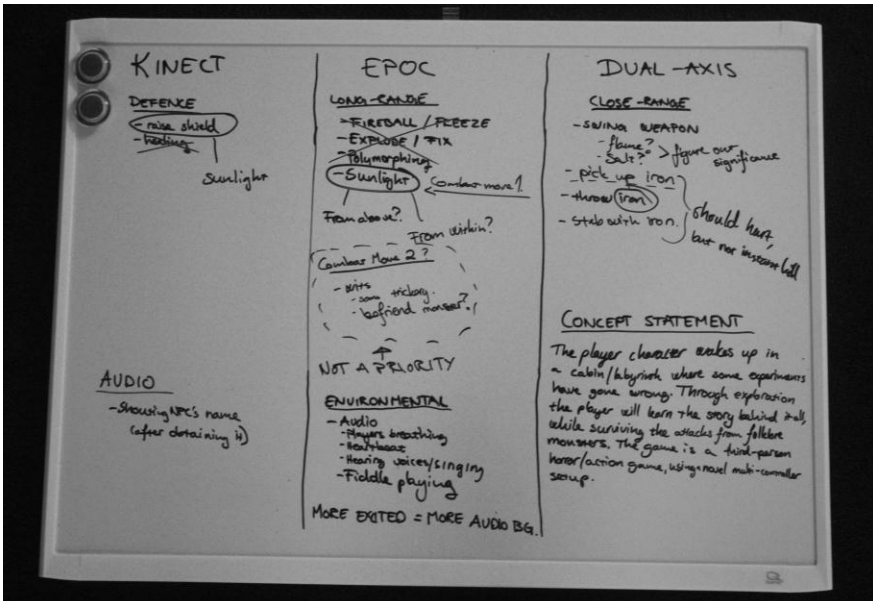


Image 4: Concept Brainstorming

User stories

The next part of the design phase, was to start writing user stories (see Image 5). I divided the user stories into categories which would be helpful in the development process, and identified the different categories by colored post-its. *Gameplay*, which included user stories that dealt with game mechanics and controllers. *Narrative*, which included aspects relating to the creative history of the game. *Aesthetic*, which included graphical elements and aesthetic experiences, and lastly *technical*, which included specific technical elements of the game.

Image 5: User Stories

Concept statement

In order to get a clear goal of what the game's concept was, I wrote a short concept statement, which would help in figuring out what would be the important aspects of the game, and get an understanding for the game.

> "The player character wakes up in a large cabin, where some experiments have obviously gone wrong. Through exploration, the player will learn the story behind it all, while surviving attacks from folklore monsters. The game is a third-person horror/action game, set in a fantasy setting, and it uses a novel controller schema".

World backstory

As proposed by Koepke et al. (in press) I wrote a world backstory as part of the simplified game design document.

> "Once upon a time, when the world was still in darkness, people used to tell stories. These stories were meant to entertain and scare, but the warnings they contained, would always be heeded. Don't go out at night alone, be a kind and clever person, and always have faith. As Man progressed, he became more and more knowledgeable, and eventually he gained control over light. But he forgot about the dangers lurking in the dark. The stories were replaced by other forms of entertainment, and the old warnings were

merely laughed at. Meanwhile, the creatures of the dark kept evolving as well, and while Man dreamed of controlling light, they dream of taming the sun. If they ever succeed, you better remember those old tales, because their warnings will be your only chance at survival. “

Game asset: Concept art poster

Initially I did some rough sketches in Photoshop to help visualize what kind of game it would be, but because I am not an artist myself, I decided to get a friend of mine to help and make a more proper concept art poster. I told him about the game, and showed him some inspirational pictures, and in result I got this poster.

Image 6: Concept Art Poster

Control schema

I also made a visual representation of the control schema for the game.

Figure 10: Control schema for Folktale

Implementation

Scrum wall

After the game design phase, I took the user stories made during the Game Design phase, and fit them to a Scrum Wall, and then wrote the rest of the user stories that was needed. Because I am the only user of this Scrum Wall, I decided to implement a bit of Kanban into the wall as well (during the development phase), and made a layout which I felt would best suit my needs.

The phases of the Scrum wall were as follows:

Backlog: all the user stories which are currently not being worked on.
Priority: the user stories which are next in line to be worked on.
Development:

> **Ongoing**: the user stories that are currently under development.
>
> **Finished**: the user stories that is functional.

Testing: the user stories that are being tested. If they are not functioning, they go back to ongoing, and if they are functioning, but not perfectly, they go back to finished until the next prototype.

Implementing: the user stories which are ready to be implemented into the final product/prototype.

Here are a couple example pictures of the scrum wall after an iteration.

Image 7: Example picture of Scrum wall after an iteration

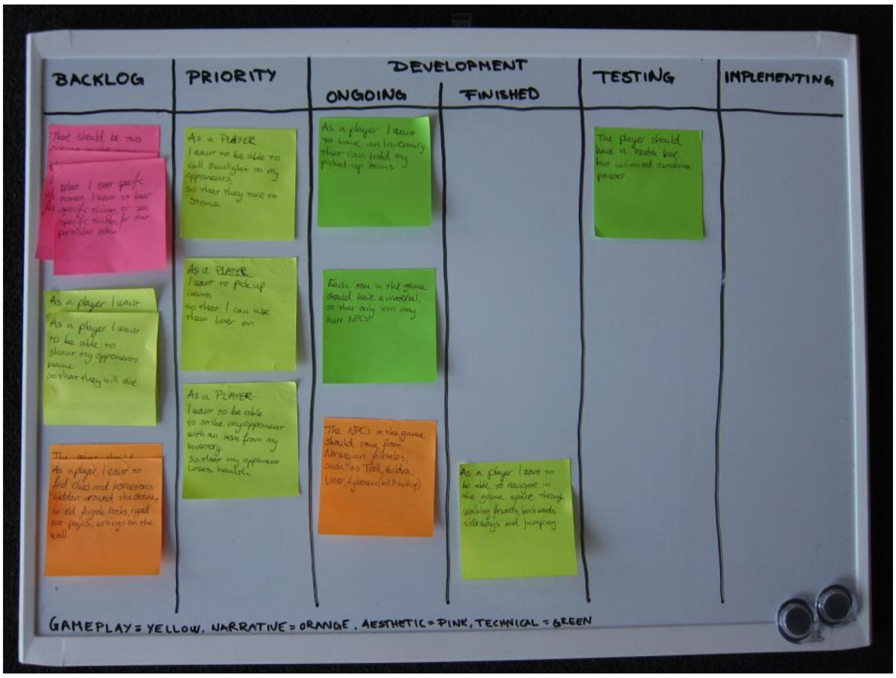


Image 8: Example picture of Scrum wall after an iteration

Coding

When writing the code for the game, I chose one of the story cards and started worked on that. When it came to writing the code in relation to the different technologies, what I did was to first make the code run with a simpler input, which was basically just to press a specific key. When the mechanic of the game matched what I wanted, first then did I start working on the code for the input from the correct technology.

Because this was just a prototype, I did not need to write any specific code for the input from the Emotiv EPOC headset. Instead I used a program which was included in the developer's package, called EmoKey, which let me map commands from the headset with the writing of a specific key or word. Although this would not be a suitable solution for a fully developed game, it was an adequate solution for this prototype.

7.3 Issues

In this section I will present some of the problems that occurred during the development process.

Kinect

In the beginning I had decided to use both the camera functionality and the audio functionality Kinect provides. Unfortunately what I didn't know was that the Zigfu ZDK did not support any of the audio functionality. And because of the way the code was written, I didn't find this out until after I had used a lot of hours developing the mechanics. Also I had envisioned a game that was highly dependent on this audio technology (searching for books to find the name of the trolls, which would instantly kill them when the player screamed into the Kinect), which meant that loosing this mechanic meant that a lot of the story and gameplay needed to be changed.

Emotiv EPOC

In the game the player was supposed to use his mind and body interchangeably in combat situations, but apparently the technology behind an EEG headset does not support the user being physically active while using it. When discussing the game with a fellow researcher at the RMIT Games Exertion Lab, who has prior experience with the technology, he explained that when a player makes big movements (such as raising an arm) the brain is receiving a lot of signals. This means that it won't be able to recognize when the user is trying to perform a cognitive command. For the game this meant that each time the player raised his arms to put up a shield, there would be approximately a 6 second cooldown before he could use the Emotiv EPOC to attack the Troll. Although it was still possible to use both body and mind, I as a developer had to include the cooldown period into the game in some way.

7.4 Analysis

When developing the prototype game *Folktale,* I found the conceptual model TechMech to be a lot of help in designing the gameplay. By having the model in front of me, it helped me remember and consider more closely what mechanics could be a good match for what controllers, especially in the initial brainstorming. But when the game was finished defined, I did not use the conceptual model any more in the development process. However, if I were to continue developing the game, it would have been unavoidable for me to not change the gameplay, due to the problems mentioned in the previous section. At that point I would likely have used the TechMech model again to figure out the best approach to changing the gameplay.

When I was designing the gameplay of *Folktale*, I had to decide which controller should be used for the action mechanic of hitting an enemy. According to the TechMech model I could use both an abstract controller and a physical controller. I could have just randomly decided which controller to use, but because I had already been reading about the theory of pragmatist aesthetics, I started thinking about the experience I wanted the player to have. And because I wanted to game to be a horror/action game, as seen in the concept statement, I decided that the abstract controller would be most suitable. This was because the player would be more effective with it, and if I had used Kinect it could have misinterpreted hitting another player with raising a shield. This in turn could make the player angry instead of feeling the sort of dread that horror games causes.

When it comes to the problems encountered with both Kinect and Emotiv EPOC, I believe these problems could have been avoided if I had known more about the technologies beforehand. The problem with the Emotiv EPOC was easily solved by including a cooldown on the action “raise shield”, and by justifying it in the game’s narrative story - such as saying that the sun-energy used for shields was so vast that the character had to wait a couple of seconds before doing other sun-related attacks. But the problem I encountered with the Kinect would eventually have led to either changing the entire gameplay, or to change the code by using another SDK. In a bigger setting this would have cost the developers a lot of time, work and money. But it could have been avoided by consulting an expert on the technology about what was possible and how it could be done, already in the game design phase.

7.5 Findings

Although the problems I encountered did not have anything to do with the conceptual TechMech model, but rather my inexperience with the technologies, I still believe it is relevant for the research questions. The issues I experienced could just as well have happened to other development teams whom are starting out developing games with new technologies. To address these issues I came up with two practices that I would advocate that other game developers use together with the conceptual model. The practices I propose are the following:

Pragmatist aesthetics

If the TechMech model says that several technologies are equally suitable for a specific mechanic, think about the aesthetic experience you want the player to have, and decide which technology better supports that.

Technology experts

When developing games that utilize different technologies, you should have an expert on that specific technology with you in the development team, most importantly in the early design phase. This can be done by either hiring a temporary expert or by appointing someone in the team as the *expert, and let that person be tasked with learning as much as possible about the hardware and software of the technology.*

Hopefully these practices together with the conceptual model will help developers when developing games that utilize different technologies in their games. Because these practices are not specific towards gaming controllers, these practices might be helpful in dealing with both input-technology and output-technology.

8 Discussion

In this section I will discuss the findings from both case studies, and the conceptual model TechMech. I will discuss both positive and negative aspects with using the TechMech model when developing games that utilize new technology, and draw in examples from the case studies to justify my claims. I will also present some general observations made while working on this thesis.

8.1 Multiple-case study

While the number of cases in this study could have been increased, I was still able to get valuable data from the study. The findings mainly suggest that if video games support a high degree of natural mapping and p-actions, they lead to a higher sense of embodiment. However, incorporating a high degree of natural mapping and p-actions should not happen in spite of GameFlow elements, especially *control* and *player skills*. I arrived at this by comparing the different video games, and trying to figure out what differentiated the enjoyable games from the unenjoyable. What I found reoccurred, was that the higher degree of natural mapping and p-actions the games supported, the more the reviewers reported to have been immersed in the game. But if that was the only criteria for an enjoyable game, then there was no reason why *Steel Battalion: Heavy Armor* should not be more enjoyable. The fact that reviewers reported that they had experienced immersive and rewarding moments in the game, which is likely due to the high degree the game has of p-actions, and natural mapping, suggests that the game could easily have been a success. Therefore something else must have been the reason for the games being less enjoyable.

Another thing that differentiated the games, were that the unenjoyable games had aspects that ruined some of the GameFlow elements, most notably *control* and *player skills*. This has led me to believe that these aspects are what ruin the player's sense of flow while playing, and that this is enough to sabotage the possible high degree of embodiment the game offers. This supports previous research where the findings suggest that games which support Flow gives a good PX (Gerling et al., 2011).

During the multiple-case study I noticed that while both *The Elder Scrolls V: Skyrim* and *Steel Battalion: Heavy Armor* were games that belonged to a previously developed video game series, only *Skyrim* were successful in incorporating Kinect into the game universe. I believe that the reason for this is that the *Skyrim* developers did not try to force the Kinect to fit the game mechanics, instead they took a step back and figured out where and how the Kinect could instead enhance the gameplay, which was by only utilizing the audio technology. The *Heavy Armor* developers on the other hand, took the game mechanics from the previous game in the series, and forced the Kinect to control those mechanics, without realizing that the mechanics were too complicated for the Kinect to handle. If they had taken a step back, figured out how Kinect could enhance the game instead, or how the game mechanics could be changed so that they were more suitable for Kinect, and thereby assuring that the game supported the flow element *control*, they might have released a more enjoyable game.

Although I did not study games that incorporated other input-technologies, using existing models and research lets me make assumptions that the findings would also apply for other technologies as well. Although this is something that should be further researched and validated.

8.2 TechMech model

The conceptual model has been developed on the basis of the findings from the multiple-case study. This means that it tries to support a high degree of p-actions and natural mapping, as well as supporting flow by adhering to the GameFlow elements. In particular it tries to avoid issues with *control* since this was something that was a reoccurring issue throughout the analysis. This assumption is however built on a small number of cases, and is something that should be validated by further research based on testing the TechMech model itself, or to try and replicate the findings from the multiple-case study with more cases. However, the research done by McEwan et al. (2012) and Gerling et al. (2011) to some degree confirms the assumption by having similar findings.

One thing I would like to explain about the use of the conceptual model is how it should be interpreted. For instance, when the TechMech model says that a controller genre is less suitable with a game mechanic, this does not mean that it cannot be done at all. What it means is that the developer needs to take extra care when deciding if it should be done, and how it can be done in the best way. For instance the model says that physical controllers are less suitable for navigation, yet the PlayStation Move is used to navigate in *BioShock: Infinite*, and it works very well. The reason behind this is that the player still uses an abstract controller for the movements that require efficient use and different types of input, such as forward, right, left, backwards etc. The Move controller however works well with physically pointing to aim, something we are familiar with from playing cowboys and Indians when we were children. They also let the camera follow the player's aim, something which feels natural because we would normally look in the same direction as we were aiming. This way of navigating works very well with the first-person shooter genre, because it increases the p-actions, but still keeping the efficiency and speed we want from that genre.

In this thesis I have only been focusing on the input technologies in games, but there have also been a lot of advancements on output technologies which can be used in games, such as using special light bulbs in the living room which can be used as a way to enhance the player's experience. Although I have not included this in my thesis, I would like to mention that when this technology is included as well, then it will be even more important to think about all the technologies, both input and output, as a part of the game's cognitive system. On a side note, output technology could also probably gain a lot of value from taking into consideration the pragmatist aesthetics view.

In developing the conceptual model I was considering how pragmatist aesthetics could be taken into consideration. What I found was that this view was not very helpful in devising the conceptual model, but it was however very helpful when developing the prototype game. This lead to my conclusion that pragmatist aesthetics can be a helpful view to use alongside the conceptual model, in the way that it can help with choosing the technology that best suits the games aesthetics, if all the possible technologies are suitable for the mechanics.

8.3 Single-case study

By using the TechMech model when designing new gaming experiences, my hope is that the developers will think more closely on the effect the controllers have on the game, and take this into account when developing games. And not end up with ideas that are fun in theory, but not in practice. This is also the experience I made from developing *Folktale*, that the design decision were more structured, but at the same time I found that I became more creative by virtue of having some principles that had to be considered. But I also recognize that this is my subjective opinion, and that other developers might feel their creative process is being hindered by having to consider the principles from the conceptual model. If that is the case, a possible solution could be to use the model after the creative design process, and use the model as a reference to compare their game design with, and then reevaluate possible problems, if there are any.

Another issue I would like to point out when it comes to the findings from the single-case study, is that they assume the conceptual model is correct. But because the TechMech model is based on a multiple-case study with a limited number of cases, which in itself has not been validated, the findings from the single-case study may be inaccurate.

In section 8.2 I proposed some practices to use alongside the TechMech model; *pragmatist aesthetics* and *technology experts*. While these practices are my subjective opinion based on the experience I had with developing a game with different input-technologies, there is little research to back them up, and they are strictly the advices from a developer.

While I have learnt a lot from doing this single-case study, I believe that it should not have been included in this thesis. Instead it should have been a part of further research trying to validate the conceptual model, and been given the time and resources it needs. For the scope of this thesis, that is something I could not do. In further research on this topic I believe it would have been better to rather conduct a quantitative study instead of the single-case study. That way it would be possible to test each type of game mechanic with each type of input-technology and compare the results with the TechMech model. Even further research could have provided established game developer teams with the conceptual model, and conducted a less biased case study on the usage of the TechMech model in a real-life environment.

8.4 Research questions

The research questions for this thesis were:

1. **How can input-technologies affect the enjoyment of video games?**
2. **Do different input-technologies work better for different types of game mechanics?**
3. **How can game developers best utilize input-technologies in video games?**

In order to answer this, I first conducted a multiple-case study which provided a possible answer to the first two questions. Games that support a high degree of p-actions, natural mapping and GameFlow elements were suggested to be more enjoyable, while games that had a lower degree of p-actions and natural mapping, and which ruined the GameFlow were less enjoyable. This suggests that input-technologies can affect the enjoyment of video games. It further suggests that different input-technologies work better for different game mechanics, where the controller schema should aim to have a high degree of embodiment, but still ensure that it fulfills the GameFlow criteria, especially *control* and *player skills*.

In order to address the third research question, I came up with a possible solution in the form of a conceptual model called TechMech that compares game mechanics and input-technologies to each other based on the findings from the multiple-case study and theories from HCI and games studies.

While I have been able to find partial answers to the research questions, the findings should still be validated.

8.5 Research method

While I believe that the research method used for this thesis have provided me with much relevant and important data, I would not have chosen the same research method if I had to do it all over again. I now think that I would have divided this research into two different thesis's; one for the multiple-case study and proposing the conceptual model, and later on I would rather have done a qualitative study similar to that of Gerling et al. (2011), Nacke et al. (2011) and McEwan et al. (2012) to try and validate the conceptual model, instead of the single-case study. That way I would have been able to go more in-depth on both topics.

9 Conclusion

In the introduction I presented an issue in the game development field, an issue I am not alone in noticing. With games becoming more and more advanced and accessible, we as gamers will also have higher expectations to the games being as enjoyable as possible. And it is therefore crucial that game researchers and game developers not only understand the medium and the technology, but also how it affects the player experience.

In order to research this topic I presented a hypothesis on the issue, and came up with three research questions that I would try to answer.

1. **How can input-technologies affect the enjoyment of video games?**
2. **Do different input-technologies work better for different types of game mechanics?**
3. **How can game developers best utilize input-technologies in video games?**

While I have not been able to find definitive answers to my research questions, I have been able to find some possible pointers to the answers.

When it comes to how input-technologies can affect the enjoyment of video games, my findings suggest that when they support a high degree of p-actions, natural mapping and GameFlow elements they have a positive effect on the degree of enjoyment. For the games that are less enjoyable, my findings suggest that they may have a lower degree of p-actions and natural mapping, and may include elements that disrupt the GameFlow.

After working with this thesis I believe that there are multiple reasons why some games are more or less enjoyable as a consequence of their input-technology use. It might be that the developers do not understand the technology they are developing for. It might also be that they are trying to force the technology to fit their game mechanics, instead of finding better alternatives. In addition there are those games that are unenjoyable not because of the controllers they utilize, but for the "good old reasons" - the game is just not entertaining enough. However, I hope the findings from this thesis can shed light on part of the subject, and help with further research on this topic.

In the research questions I also wanted to figure out how game developers could best utilize different input-technologies in their games. While I have not found any definitive answer as to how we can avoid developing games that are not enjoyable due to their controller schema, I came up with a possible solution in the form of a conceptual model called TechMech that compares game mechanics and input-technologies to each other based on the findings from the multiple-case study. In addition to the conceptual model, I have presented some practices that could also be of help if applied in the development process. Although I believe the model can be helpful to developers, I also believe that it is necessary with more research, both on the conceptual model and on the issue in general. Hopefully the research I have done here can be of help in the industry, and for other researchers who are concerned with the same issues as I have explored in this thesis.

Other researchers have approached the issue with different viewpoints than what I have done. Mostly they have taken a more quantitative approach with testing specific controller schema or specific games, such as Nacke et al. (2011), Gerling et al. (2011), Freeman et al. (2012) and Skalski et al. (2011) does. I hope that the work presented in this thesis together with their research can shed light on the topic in a broader sense.

Further research on this topic is needed in order to help understand the effect technology and controllers have on the gaming experience. I believe the findings from this thesis should be validated, and propose two possible ways to do this. The first one is that more video games should be analyzed in the same way as in the multiple-case study to see if the findings can be attributed to a larger data collection. The other approach I would suggest is to systematically test the conceptual model using quantitative research, where one looks at each type of game mechanic with each type of controller in order to figure out what users would prefer. This way the TechMech model could be validated or adjusted according to newer findings.

My hopes for this thesis is that it can contribute to our understanding of video games and controllers, and that in the future we will continue to experience enjoyable and immersive video games.

10 Sources

Games cited

AVALANCHE SOFTWARE 2013. *Disney Infinity*. Wii U: Disney Interactive Studios.

BETHESDA GAME STUDIOS 2011. *The Elder Scrolls V: Skyrim*. Xbox 360: Bethesda Softworks.

EA DIGITAL ILLUSIONS CE 2010. *Battlefield: Bad Company 2*. Xbox 360, PlayStation 3: Electronic Arts.

EUROCOM 2012. *Harry Potter for Kinect*. Xbox 360: Warner Bros. Interactive Entertainment.

FROM SOFTWARE 2012. *Steel Battalion: Heavy Armor*. Xbox 360: Capcom.

GOOD SCIENCE STUDIO 2010. *Kinect Adventures!* Xbox 360: Microsoft Game Studios.

HALFBRICK STUDIOS 2011. *Fruit Ninja Kinect*. Xbox 360.

IRRATIONAL GAMES 2013. *BioShock: Infinite*. PlayStation 3: 2K Games.

JAPAN STUDIO 2010. *Kung Fu Rider.* PlayStation 3: Sony Computer Entertainment.

SAN DIEGO STUDIO 2010. *Sports Champions.* PlayStation 3: Sony Computer Entertainment.

UBISOFT 2011. *Self-Defense Training Camp*. *My Coach.* Xbox 360: Ubisoft.

Technology cited

EMOTIV SYSTEMS . *Emotiv EPOC.* Wireless Neuroheadset: Emotiv Systems.

MICROSOFT 2010. *Kinect*. Motion Sensing Input Device: Microsoft.

LEAP MOTION, INC 2010. *Leap Motion*. Motion Controller: Leap Motion, Inc.

OCULUS VR, INC 2012. *Oculus Rift*. Head-mounted display: Oculus VR, Inc.

SONY COMPUTER ENTERTAINMENT 2010. *PlayStation Move*. Motion Controller: Sony Computer Entertainment

NINTENDO 2006. *Wii.* Video Game Consol: Foxconn.

NINTENDO 2012. *Wii U*. Video Game Console: Foxconn.

MICROSOFT 2013. *Xbox One*. Video Game Console: Microsoft.

Literature references

1UP. 2005. *Controller's History Dynamite: A history of gaming input devices, from stolen switches to analog triggers.* [Online]. www.1up.com: 1up. Available: http://www.1up.com/features/controller-history [Accessed 20 March 2012].

ABRAHAMSSON, P., SALO, O., RONKAINEN, J. & WARSTA, J. 2002. Agile software development methods. *Relatório Técnico, Finlândia*.

BARKER, S. 2013. First Impressions: Is BioShock: Infinite Any Good with the PlayStation Move? Available: http://www.pushsquare.com/news/2013/03/first_impressions_is_bioshock_infinite_any_good_with_the_playstation_move [Accessed 09 April 2013].

BENBASAT, I., GOLDSTEIN, D. K. & MEAD, M. 1987. The case research strategy in studies of information systems. *MIS quarterly*, 369-386.

BERGZERGARCADE. 2011. *Hack & Slash RPG - A Unity3D Game Engine Tutorial* [Online]. www.bergzergarcade.com: Berg Zerg Arcade. Available: http://www.burgzergarcade.com/hack-slash-rpg-unity3d-game-engine-tutorial [Accessed 14 August 2012].

BERNHAUPT, R. 2010. User Experience Evaluation in Entertainment. *Evaluating User Experience in Games*, 3-7.

BRADLEY, L. 2012. *All of Skyrim's Kinect Voice Commands in One Place* [Online]. www.xbox360achievements.org: Xbox360Achievements. Available: http://www.xbox360achievements.org/news/news-11574-All-of-Skyrim-s-Kinect-Voice-Commands-in-One-Place.html [Accessed 23 May 2013].

BRYMAN, A. 2008. *Social research methods*, Oxford university press Oxford.

CAPCOM. 2012. *Steel Battalion: Heavy Armor* [Online]. www.capcom-europe.com: Capcom. Available: http://www.capcom-europe.com/games/Steel-Battalion-Heavy-Armor-Xbox-360 [Accessed 23 May 2013].

CARROLL, J. M. 2009. Human Computer Interaction (HCI). *In:* SOEGAARD, M. A. D., RIKKE FRIIS (ed.) *Encyclopedia of Human-Computer Interaction.* Aarhus, Denmark.

CLEMENTS, R. 2010. Kung Fu Rider Review: An exercise in frustration. Available: http://au.ign.com/articles/2010/09/01/kung-fu-rider-review [Accessed 06 February 2013].

CODECADEMY. 2012. *JavaScript* [Online]. www.codecademy.com: Codecademy. Available: http://www.codecademy.com/tracks/javascript [Accessed 15 August 2012].

CORK, J. 2012. Gameplay Goes AWOL In Mech-Combat Revival. *Steel Battalion: Heavy Armor* [Online]. Available: http://www.gameinformer.com/games/steel_battalion_heavy_armor/b/xbox360/archive/2012/06/19/gameplay-goes-awol-in-mech-combat-revival.aspx [Accessed 25 February 2013].

DECAYEDMATTER. 2013. *BioShock Infinite - Playstation 3 - In Depth PS Move Demonstration* [Online]. Youtube. Available: https://www.youtube.com/watch?v=zSGgdsnCTRE [Accessed 09 April 2013.

DEEN, P. D. 2011. Interactivity, Inhabitation and Pragmatist Aesthetics. *Game Studies,* 11.

DEFENDINI, G. 2010. *Kung Fu Rider - PSMove - Gameplay (PS3)* [Online]. Youtube. Available: https://www.youtube.com/watch?v=5NcuvcwcdsA [Accessed 06 February 2013].

DOURISH, P. 2004. *Where the action is: the foundations of embodied interaction*, The MIT Press.

EGENFELDT-NIELSEN, S., SMITH, J. H. & TOSCA, S. P. 2008. *Understanding video games: the essential introduction*, Taylor & Francis.

EMOTIV. 2012a. *Developer neuroheadset* [Online]. Available: http://www.emotiv.com/store/hardware/epoc-bci-eeg/developer-neuroheadset/ [Accessed 20 March 2012].

EMOTIV. 2012b. *Emotiv EPOC Unity3D™ Developer Support Pack* [Online]. Available: https://emotiv.com/store/apps/applications/131/4021 [Accessed 22 August 2012].

ESCAPIST, T. 2005. *The Escapist* [Online]. www.escapistmagazine.com: Alloy Digital, LLC. Available: http://www.escapistmagazine.com [Accessed 29 May 2013].

FOWLER, M. & HIGHSMITH, J. 2001. The agile manifesto. *Software Development,* 9**,** 28-35.

FREEMAN, D., HILLIGES, O., SELLEN, A., O'HARA, K., IZADI, S. & WOOD, K. The role of physical controllers in motion video gaming. Proceedings of the Designing Interactive Systems Conference, 2012. ACM, 701-710.

GAMESPOT. 1996. *Gamespot* [Online]. www.gamespot.com: CBS Interactive. Available: http://www.gamespot.com [Accessed 29 May 2013].

GERLING, K. M., KLAUSER, M. & NIESENHAUS, J. Measuring the impact of game controllers on player experience in FPS games. Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, 2011. ACM, 83-86.

GILBERT, B. 2012. Skyrim's voice-enabled Dragon Shouts arrive 'the week of April 23'. *The Elder Scrolls V: Skyrim* [Online]. Available: http://www.joystiq.com/2012/04/13/skyrims-voice-enabled-dragon-shouts-arrive-the-week-of-april-2/ [Accessed 23 May 2013].

GIZMODO. 2013. Valve Testing Sweat-Sensing Biometric Gaming Input in Left4Dead. Available: http://www.gizmodo.co.uk/2013/05/valve-testing-sweat-sensing-biometric-gaming-controls/ [Accessed 07 May 2013].

GREGERSEN, A. & GRODAL, T. 2009. Embodiment and interface. *The Video Game Theory Reader,* 2**,** 65-85.

HALFBRICK. 2013. *Fruit Ninja Kinect* [Online]. www.fruitninja.com: Halfbrick. Available: http://fruitninja.com/fruit-ninja-kinect/ [Accessed 29 May 2013].

HARAWAY, D. 1984. A manifesto for cyborgs: Science, technology, and socialist feminism in the 1980s. *The postmodern turn: New perspectives on social theory***,** 82-115.

HINKLE, D. 2012. Talking my way through Skyrim's free Kinect update. *The Elder Scrolls V: Skyrim* [Online]. Available: http://www.joystiq.com/2012/04/20/talking-my-way-through-skyrims-free-kinect-update/ [Accessed 23 May 2013].

HOLLAN, J., HUTCHINS, E. & KIRSH, D. 2000. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction (TOCHI),* 7**,** 174-196.

HOPPER, S. 2012. Kinecting with Skyrim: Dragon Shout your way to Dovahkiin glory. Available: http://au.ign.com/articles/2012/04/20/kinecting-with-skyrim [Accessed 23 May 2013].

IGN. 1996. *IGN* [Online]. www.ign.com: IGN Entertainment. Available: http://au.ign.com [Accessed 29 May 2013].

Why is PlayStation Move in BioShock Infinite? - Podcast Beyond Episode 282, 2013. Podcast video. Directed by IGN. www.ign.com.

INFINITE, B. 2013. *The Game* [Online]. Available: http://www.bioshockinfinite.com/the-game/ [Accessed 09 April 2013].

INFORMER, G. 1996. *Game Informer* [Online]. www.gameinformer.com: GameStop Corp. . Available: http://www.gameinformer.com [Accessed 30 May 2013].

JOYSTIQ. 2004. *Joystiq* [Online]. www.joystiq.com: AOL. Available: http://www.joystiq.com [Accessed 30 May 2013].

JUUL, J. 2009. *A casual revolution: reinventing video games and their players*, The MIT Press.

KOEPKE, B., PELLETIER, B., ADAIR, D., JHAWAR, R., MACAULAY, I. & BIELECKI, T. in press. Agile Game Development. Unpublished paper, University of Calgary.

MARTIN, R. C. 2003. *Agile software development: principles, patterns, and practices*, Prentice Hall PTR.
MCCAFFREY, R. 2012. Steel Battalion: Heavy Armor Review. Available: http://au.ign.com/articles/2012/06/19/steel-battalion-heavy-armor-review [Accessed 25 May 2013].
MCCALLUM, S. 2013. Game Technology of the Future. *Evolution of Games, the Game industry, Game Technology and Gamers.* JoinGame.
MCEWAN, M., JOHNSON, D., WYETH, P. & BLACKLER, A. Videogame control device impact on the play experience. Proceedings of The 8th Australasian Conference on Interactive Entertainment: Playing the System, 2012. ACM, 18.
METACRITIC. 2001. *Metacritic* [Online]. www.metacritic.com: CBS Interactive. Available: http://www.metacritic.com/game [Accessed 05 February 2013].
MILLER, R. 2010a. Kinect for Xbox 360 review. Available: http://www.engadget.com/2010/11/04/kinect-for-xbox-360-review/ [Accessed 18 September 2013].
MILLER, R. 2010b. PlayStation Move review. Available: http://www.engadget.com/2010/09/01/playstation-move-review/ [Accessed 18 September 2013].
MITCHELL, R. 2012. Steel Battalion: Heavy Armor review: Grasping at nothing. Available: http://www.joystiq.com/2012/06/19/steel-battalion-heavy-armor-review/ [Accessed 25 February 2013].
NACKE, L. E., KALYN, M., LOUGH, C. & MANDRYK, R. L. Biofeedback game design: using direct and indirect physiological control to enhance game interaction. 2011. ACM, 103-112.
NAKAMURA, J. & CSIKSZENTMIHALYI, M. 2002. The concept of flow. *Handbook of positive psychology***,** 89-105.
NELSON, R. 2010. PlayStation Move review: Kung Fu Rider. Available: http://www.joystiq.com/2010/09/01/kung-fu-rider-review/ [Accessed 06 February 2013].
NINTENDO. 2013. *Wii U* [Online]. www.nintendo.com: Nintendo. Available: http://www.nintendo.com/wiiu/features/ [Accessed 29 May 2013].
NUNAMAKER JR, J. A. Y. F., CHEN, M. & PURDIN, T. D. M. 1991. Systems Development in Information Systems Research. *Journal of Management Information Systems,* 7.
PLAYSTATION. 2010. *Kung Fu Rider* [Online]. www.playstation.com: PlayStation. Available: http://uk.playstation.com/ps3/games/detail/item285333/Kung-Fu-Rider/ [Accessed 29 May 2013].
ROGERS, Y., SHARP, H. & PREECE, J. 2011. *Interaction design: beyond human-computer interaction*, Wiley Publishing.
RYAN, T. 1999. The Anatomy of a Design Document, Part 1: Documentation Guidelines for the Game Concept and Proposal. Available: http://www.gamasutra.com/view/feature/131791/the_anatomy_of_a_design_document_.php?page=2 [Accessed 25.09.2012].
SHORE, J. & WARDEN, S. 2007. *The art of agile development*, O'Reilly.
SICART, M. 2008. Defining game mechanics. *Game Studies,* 8.
SKALSKI, P., TAMBORINI, R., SHELTON, A., BUNCHER, M. & LINDMARK, P. 2011. Mapping the road to fun: Natural video game controllers, presence, and game enjoyment. *New Media & Society,* 13**,** 224-242.
SOY, S. K. 1997. The case study as a research method. *Unpublished paper, University of Texas at Austin*.
SQUARE, P. 2009. *Push Square* [Online]. www.pushsquare.com: Push Square. Available: http://www.pushsquare.com [Accessed 29 May 2013].

STUDIO, W. B. 2011. *Video Training and Tutorials* [Online]. www.vimeo.com: Walker Boys Studio. Available: http://vimeo.com/channels/151501 [Accessed 14 August 2012].
SWEETSER, P. & WYETH, P. 2005. GameFlow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE),* 3**,** 3-3.
TELLIS, W. 1997. Application of a case study methodology. *The qualitative report,* 3**,** 1-17.
TITO, G. 2012. Steel Battalion: Heavy Armor Review. Available: http://www.escapistmagazine.com/articles/view/editorials/reviews/9774-Steel-Battalion-Heavy-Armor-Review [Accessed 25 May 2013].
UNITY3D. 2012. *Unity3D* [Online]. Available: http://unity3d.com [Accessed 20 March 2012].
VIDEOS, M. 2012. *Steel Battalion Heavy Armor - Demo Tutorial Gameplay HD* [Online]. Youtube. Available: https://www.youtube.com/watch?v=O_KmU9eCA9U [Accessed 25 February 2013].
WATTERS, C. 2012. Steel Battalion: Heavy Armor Review. Available: http://au.gamespot.com/steel-battalion-heavy-armor/reviews/steel-battalion-heavy-armor-review-6383411/ [Accessed 25 February 2013].
WIKIAGAMING. 2010. *WASD* [Online]. Available: http://gaming.wikia.com/wiki/WASD [Accessed 10 April 2012].
WIKIPEDIA. 2010. *The Elder Scrolls V: Skyrim* [Online]. www.wikipedia.org. Available: http://en.wikipedia.org/wiki/Skyrim [Accessed 29 May 2013].
WOLPAW, J. R., BIRBAUMER, N., MCFARLAND, D. J., PFURTSCHELLER, G. & VAUGHAN, T. M. 2002. Brain-computer interfaces for communication and control. *Clinical neurophysiology,* 113**,** 767-791.
XBOX. 2013. *Xbox One* [Online]. www.xbox.com: Xbox. Available: http://www.xbox.com/en-AU/xboxone/what-it-is [Accessed 29 May 2013 2013].
YIN, R. K. 2008. *Case study research: Design and methods*, SAGE Publications, Incorporated.
ZACCARI, J. 2013. *Disney announces Disney Infinity for Wii U* [Online]. www.wiiudaily.com: Wii U Daily. Available: http://wiiudaily.com/2013/01/disney-infinity-announced/ [Accessed 17 January 2013].
ZIGFU. 2012. *Zigfu - Kinect Development in HTML, Unity3D and Flash:* [Online]. zigfu.com. Available: http://zigfu.com [Accessed 17 September 2012].

11 Appendix

Image A 1: Control schema for BioShock: Infinite

Image A 2: Control schema for Kung Fu Rider

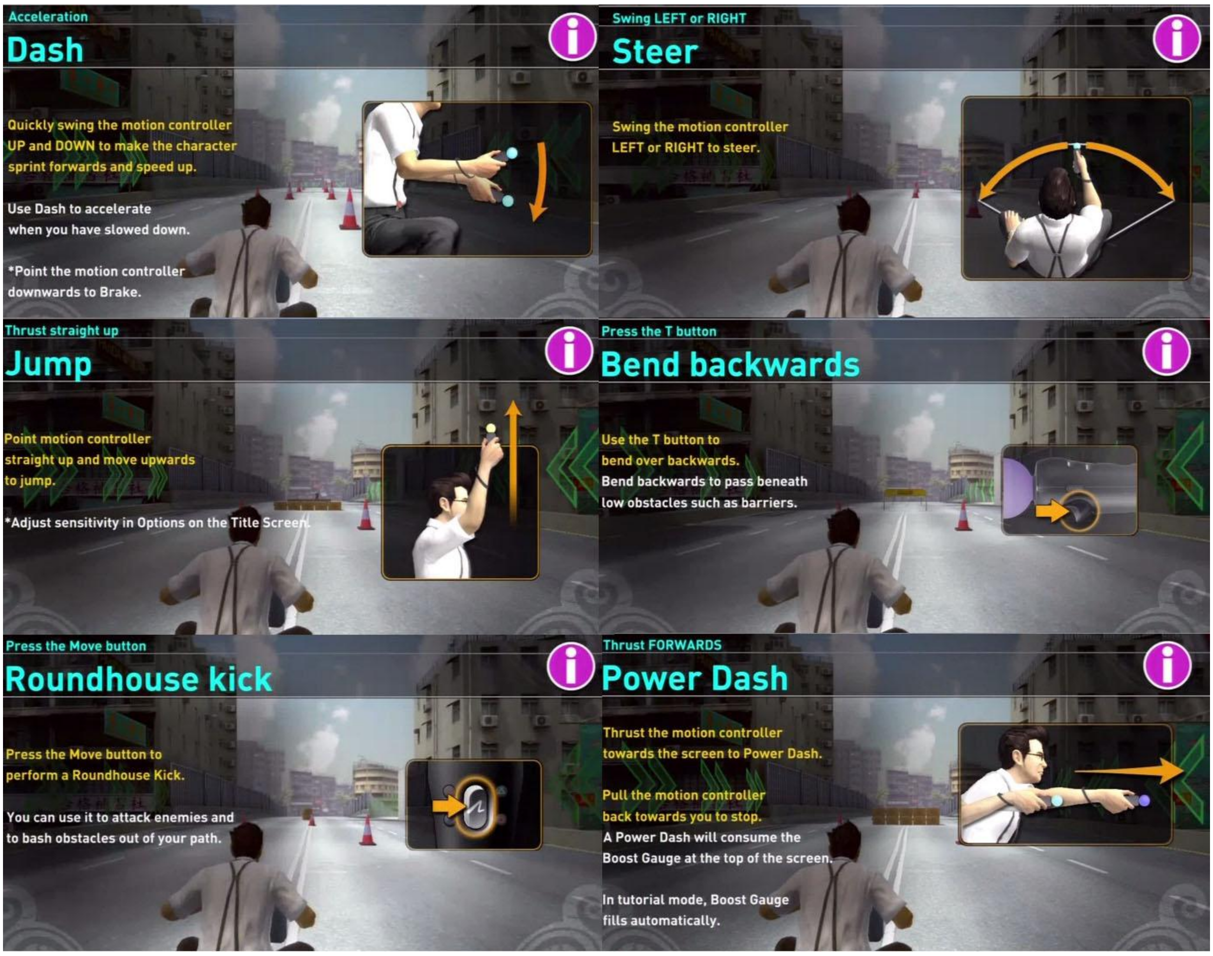


Image A 3: Control schema for The Elder Scrolls V: Skyrim

Image A 4: Control schema for The Elder Scrolls V: Skyrim

Image A 5: Control schema for The Elder Scrolls V: Skyrim

FAVORITES MENU & HOTKEY EQUIPPING

Only usable in the Favorites menu.

ASSIGN <hotkey command> sets the selected item to the spoken hotkey. <hotkey command> can be any of the following:

- HEALTH POTION
- MAGICKA POTION
- STAMINA POTION
- POISON
- SWORD
- MACE
- AXE
- DAGGER
- BOW
- SHIELD
- DUAL WIELD LEFT
- DUAL WIELD RIGHT
- SOUL TRAP
- GREATSWORD
- BATTLEAXE
- WARHAMMER
- FIRE SPELL
- FROST SPELL
- LIGHTNING SPELL
- WARD SPELL
- RITUAL SPELL
- BOUND WEAPON
- SUMMON SPELL
- ARMOR SPELL
- CALM SPELL
- FRENZY SPELL
- HEALING SPELL
- LIGHT

Only usable during main gameplay.

EQUIP <hotkey command>
Equips item in the default hand

EQUIP LEFT <hotkey command>
Equips item in the Left hand, if possible.

EQUIP RIGHT <hotkey command>
Equips item in the Right hand, if possible.

EQUIP DUAL <hotkey command>
Equips the item in both hands, if possible.

EQUIP SWORD AND SHIELD/ EQUIP MACE AND SHIELD/ EQUIP AXE AND SHIELD/ EQUIP DAGGER AND SHIELD
Equips the item assigned to the Sword / Mace / Axe / Dagger hotkey in the right hand, and the item assigned to the Shield hotkey in the left hand.

EQUIP DUAL WEAPONS
Equips the item assigned to the DualWieldLeft hotkey in the left hand, and the item assigned to the DualWieldRight hotkey in the left hand.

FOLLOWER COMMANDS

ALLY <action command>
Tells the ally to activate the thing you're looking at, as if you entered Command Mode and selected it.

FOLLOW / FOLLOW ME
Ally follows you

WAIT / WAIT HERE
Ally stays where they are

TRADE / ITEMS
Opens the Teammate Menu

OPEN
Ally opens the container/door you're looking at

STAND
Ally stands at the spot you're looking at

RETRIEVE
Ally picks up the item you're looking at

ATTACK
Ally attacks the NPC you're looking at

INTERACT / USE
Does a generic interact command with the ally. Same as if you entered Command mode and pressed Ⓐ. What the ally does is dependent on what you're looking at.

ITEMS MENU

After opening ITEMS, the following commands open the corresponding categories:

- FAVORITES
- ALL
- WEAPONS
- APPAREL / ARMOR
- POTIONS
- SCROLLS
- FOOD
- INGREDIENTS
- BOOKS
- KEYS
- MISCELLANEOUS

After opening any of the above sub-menu categories, you can use the following to sort your items.

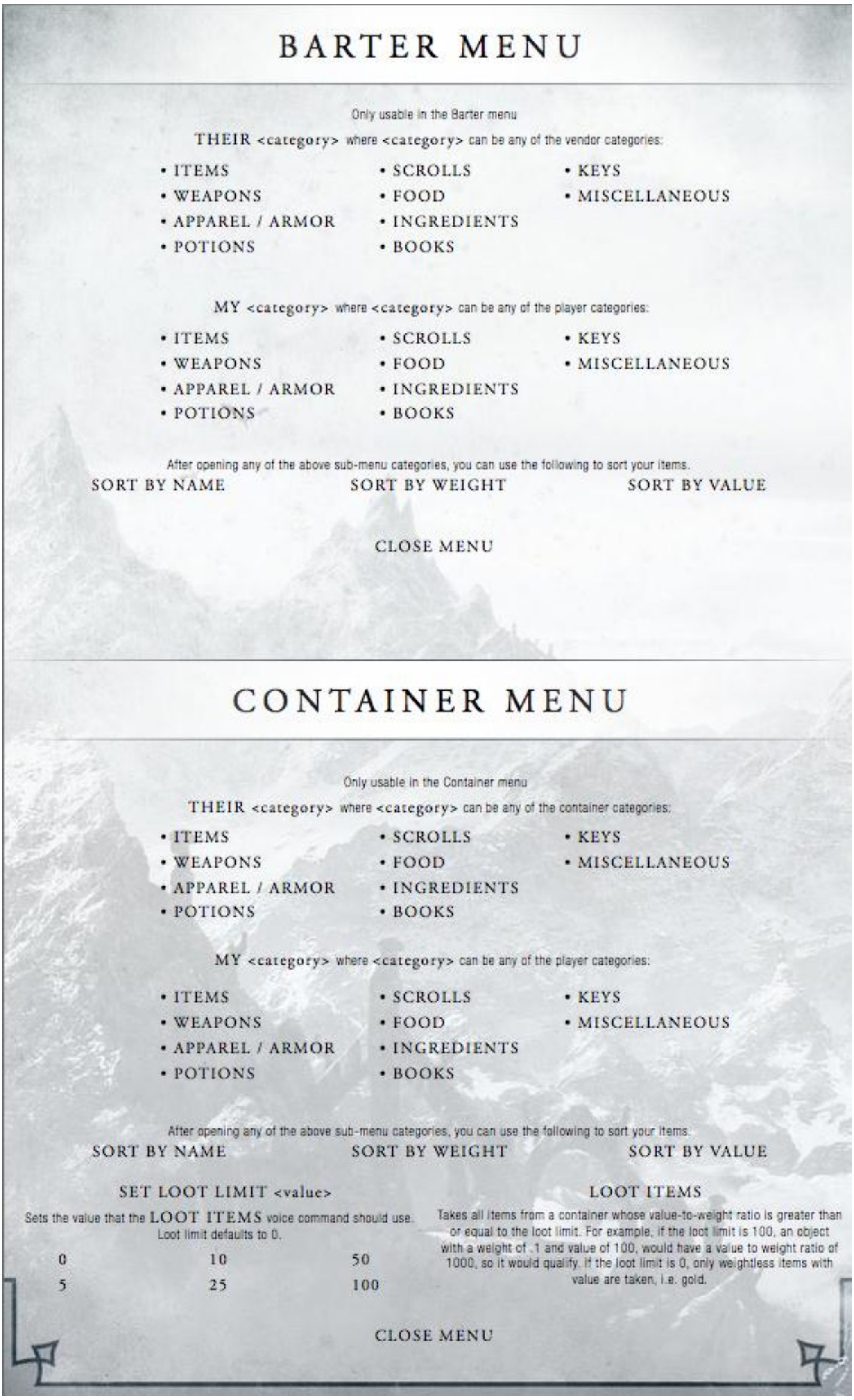
SORT BY NAME
Sorts the item list by name, increasing. If the list is already sorted by name increasingly, sorts it decreasingly.

SORT BY WEIGHT
Sorts the item list by weight, decreasing. If the list is already sorted by weight decreasingly, sorts it increasingly.

SORT BY VALUE
Sorts the item list by value, decreasing. If the list is already sorted by value decreasingly, sorts it increasingly.

CLOSE MENU
Closes the menu

Image A 6: Control schema for The Elder Scrolls V: Skyrim



BARTER MENU

Only usable in the Barter menu

THEIR <category> where <category> can be any of the vendor categories:

- ITEMS
- WEAPONS
- APPAREL / ARMOR
- POTIONS
- SCROLLS
- FOOD
- INGREDIENTS
- BOOKS
- KEYS
- MISCELLANEOUS

MY <category> where <category> can be any of the player categories:

- ITEMS
- WEAPONS
- APPAREL / ARMOR
- POTIONS
- SCROLLS
- FOOD
- INGREDIENTS
- BOOKS
- KEYS
- MISCELLANEOUS

After opening any of the above sub-menu categories, you can use the following to sort your items.

SORT BY NAME SORT BY WEIGHT SORT BY VALUE

CLOSE MENU

CONTAINER MENU

Only usable in the Container menu

THEIR <category> where <category> can be any of the container categories:

- ITEMS
- WEAPONS
- APPAREL / ARMOR
- POTIONS
- SCROLLS
- FOOD
- INGREDIENTS
- BOOKS
- KEYS
- MISCELLANEOUS

MY <category> where <category> can be any of the player categories:

- ITEMS
- WEAPONS
- APPAREL / ARMOR
- POTIONS
- SCROLLS
- FOOD
- INGREDIENTS
- BOOKS
- KEYS
- MISCELLANEOUS

After opening any of the above sub-menu categories, you can use the following to sort your items.

SORT BY NAME SORT BY WEIGHT SORT BY VALUE

SET LOOT LIMIT <value>

Sets the value that the LOOT ITEMS voice command should use. Loot limit defaults to 0.

0 10 50
5 25 100

LOOT ITEMS

Takes all items from a container whose value-to-weight ratio is greater than or equal to the loot limit. For example, if the loot limit is 100, an object with a weight of .1 and value of 100, would have a value to weight ratio of 1000, so it would qualify. If the loot limit is 0, only weightless items with value are taken, i.e. gold.

CLOSE MENU

Image A 7: Control schema for The Elder Scrolls V: Skyrim

MAGIC MENU

After opening MAGIC, the following opens the menu to the corresponding category:

FAVORITES
ALL

ALTERATION
ILLUSION
DESTRUCTION

CONJURATION
RESTORATION
SHOUTS

POWERS
ACTIVE EFFECTS / EFFECTS

CHARACTER MENU

After opening CHARACTER, the following opens the menu to the corresponding category:

ITEMS
MAGIC
SKILLS
MAP

MAP MENU

After opening MAP, the following centers the camera on the corresponding locations:

WINDHELM
FALKREATH
DAWNSTAR

MORTHAL
SOLITUDE
MARKARTH

RIFTEN
WHITERUN
WINTERHOLD

PLAYER / WHERE AM I
Centers camera on you.

WAYPOINT
Centers camera on marker set by you, if applicable.

QUEST MARKER / QUEST
Centers camera on active quest target. Saying it multiple times will cycle through targets.

SKILLS MENU

After opening SKILLS, use the following commands to center the camera on that skill's perk tree:

ONE HANDED
TWO HANDED
ARCHERY
BLOCK
SMITHING

HEAVY ARMOR
LIGHT ARMOR
PICKPOCKET
LOCKPICKING
SNEAK

ALCHEMY
SPEECH
ALTERATION
CONJURATION
DESTRUCTION

ILLUSION
RESTORATION
ENCHANTING

Image A 8: Control schema for Steel Battalion: Heavy Armor

Attachment 9: a .zip file containing the programming files for the prototype game *Folktale*.