Motion Controllers for Game Consoles

MATTIAS LANDERHOLM

KTH Computer Science and Communication

Master of Science Thesis
Stockholm, Sweden 2011
Motion Controllers for Game Consoles

MATTIAS LANDERHOLM

Master’s Thesis in Media Technology (30 ECTS credits)
at the School of Media Technology
Royal Institute of Technology year 2011
Supervisor at CSC was Trille Fellstenius
Examiner was Nils Enlund

TRITA-CSC-E 2011:022
ISRN-KTH/CSC/E--11/022--SE
ISSN-1653-5715

Royal Institute of Technology
School of Computer Science and Communication
KTH CSC
SE-100 44 Stockholm, Sweden
URL: www.kth.se/csc
Motion Controllers for Game Consoles

Abstract

The idea of this thesis is to explore the possibilities of the new video game devices PlayStation Move and Microsoft’s Kinect, further on the goal was to design and verify an interaction technique for each device, suitable for playing First-Person Shooter (FPS) games.

During early prototyping it showed that it was not possible to turn the Kinect into an effective FPS game controller; because of this the focus of the thesis was shifted to the PlayStation Move exclusively. An interaction technique taking into account how FPS games traditionally have been played was designed. This technique differs from what is here referred to as the point-and-shoot technique which is often used in conjunction with motion controllers.

Performing tests on the implementation of the PlayStation Move on a total of fourteen persons showed that the suggested interaction technique seemed to have high performance and acceptance among the participants of the test. However more testing would be necessary to support the result.

Rörelsekontroller för spelkonsoler

Sammanfattning

Idén med detta examensarbete är att undersöka möjligheterna för PlayStation Move och Microsoft’s Kinect. Mer specifikt går det ut på att, för dem båda, designa och verifiera en interaktionsteknik som kan användas för spel av typen förstapersonskjutare (FPS).

Tidigt i prototypfasen visade det sig att Kinect inte skulle uppnå kraven för en effektiv FPS-kontroll. Detta gjorde att fokus helt och hållet flyttades över till PlayStation Move. En interaktionsteknik som påminner om hur man traditionellt sett har spelat FPS:er utvecklades. Tekniken skiljer sig från det som i rapporten kallas för point-and-shoot, vilket är en teknik som ofta används tillsammans med rörelsebaserade spelkontroller.

Tester på den implementerade interaktionstekniken, med totalt fjorton personer, visade att tekniken hade förhållandevis hög effektivitet och acceptans bland deltagarna. För att styrka detta vore det dock nödvändigt med ytterligare tester.
Acknowledgements

I would like to thank all the people at DICE who have helped me out during the writing of this thesis, foremost a big thanks to my supervisors Torbjörn Söderman and Håkan Kihlström for their support and guidance.

Stockholm, 2011
Mattias Landerholm
# Table of Contents

1 Introduction .................................................................................................................. 1
  1.1 Background to the Thesis ......................................................................................... 1
    1.1.1 EA DICE ........................................................................................................ 1
    1.1.2 New Technology ............................................................................................ 1
    1.1.3 The FPS Genre .............................................................................................. 1
  1.2 Objectives and Aim .................................................................................................. 1
    1.2.1 Problem Definition ......................................................................................... 2
    1.2.2 Additional Questions ....................................................................................... 2
    1.2.3 Delimitation ..................................................................................................... 2
  1.3 Structure of the Thesis ............................................................................................. 2

2 Background and Theory ................................................................................................. 3
  2.1 Related Work ........................................................................................................... 3
    2.1.1 Engagement ..................................................................................................... 3
    2.1.2 Movement ........................................................................................................ 3
    2.1.3 Taxonomies ..................................................................................................... 4
    2.1.4 3D User Interfaces ......................................................................................... 4
  2.2 PlayStation Move .................................................................................................... 4
  2.3 Isotonic Device ........................................................................................................ 5
  2.4 Microsoft Kinect ...................................................................................................... 6
  2.5 First-Person Shooter Games ................................................................................... 6
  2.6 Movement in Games ............................................................................................... 7
  2.7 Guidelines for 3D User Interfaces ......................................................................... 8
  2.8 Performance ............................................................................................................ 8
  2.9 Ergonomics ............................................................................................................. 9

3 Interaction Design ......................................................................................................... 10
  3.1 Decisions ................................................................................................................ 10
  3.2 Prototypes .............................................................................................................. 11
    3.2.1 PlayStation Move ......................................................................................... 11
    3.2.2 Kinect ........................................................................................................... 12
  3.3 Implementation ...................................................................................................... 13
    3.3.1 PlayStation Move ......................................................................................... 13

4 Test Method .................................................................................................................. 18
  4.1 Test Level ............................................................................................................... 18
  4.2 Playtest .................................................................................................................... 19

5 Results .......................................................................................................................... 21
Table of Figures

Figure 1: Sony PlayStation Move controller ................................................................. 4
Figure 2: Sony PlayStation Move controller (alternate angle) ........................................ 4
Figure 3: Sony PlayStation Move Navigation Controller .................................................. 5
Figure 4: Sony PlayStation Move Navigation Controller (alternate angle) ..................... 5
Figure 5: Microsoft Kinect .............................................................................................. 6
Figure 6: Ideal input curve .............................................................................................. 11
Figure 7: Dead zone ......................................................................................................... 11
Figure 8: Kinect prototype, left: top view, right: side view ................................................. 12
Figure 9: Squared input .................................................................................................... 14
Figure 10: Normalized dead zone ..................................................................................... 14
Figure 11: Ramp dead zone ............................................................................................... 15
Figure 12: Gestures ........................................................................................................... 16
Figure 13: Test Level – First target .................................................................................. 18
Figure 14: Diagram showing the average completion times per participant ...................... 21
Figure 15: Diagram showing the overall average of completion times ............................... 22
Figure 16: Mouse and keyboard experience in FPS games .............................................. 23
Figure 17: Gamepad experience in FPS games .................................................................. 23
Figure 18: PlayStation Move experience (games in general) ........................................... 23
Figure 19: Which button did you use to fire? .................................................................. 23
Figure 20: How did you like the control scheme in general? ........................................... 23
Figure 21: For jumping, crouching, meleeing; did you prefer to use: buttons or gestures? .. 23
Figure 22: How often did you use the possibility of lowering the aiming speed by only partially depressing the trigger? .............................................................. 24
Figure 23: Acquiring targets ............................................................................................ 24
Figure 24: Aiming at moving targets ............................................................................... 24
Figure 25: Aiming while rapidly firing ............................................................................. 24
Figure 26: “I would rather play FPS with the Move than a gamepad.” ............................... 24
Figure 27: “I would rather play FPS with the Move than a mouse and keyboard.” ............. 24
Figure 28: “I felt frustrated playing with the Move controller.” ....................................... 25
Figure 29: “I feel that I have improved my skill with the Move from when I first started.” .... 25
Figure 30: How tiresome was it to use the Move controller? ........................................... 25
Figure 31: How much pain or strain in muscles, tendons etc., did you experience? .......... 25
Figure 32: Test Level – Level start ............................................................................... 38
Figure 33: Test Level – Level end .................................................................................. 38
Figure 34: Test Level – Top view ................................................................................... 39
Figure 35: Test Level – Top view, textured ...................................................................... 39
Figure 36: Sony PlayStation Dual Shock 3 (top view) ..................................................... 40
Figure 37: Microsoft Xbox 360 Controller (top view) ...................................................... 40
Figure 38: Sony PlayStation Dual Shock 3 (front view) ................................................... 40
Figure 39: Microsoft Xbox 360 Controller (front view) .................................................. 40
1 Introduction

In this, the first chapter, the background for the thesis will be presented. It will introduce you to the technology discussed and also the prerequisites under which this thesis was written.

1.1 Background to the Thesis

1.1.1 EA DICE
The thesis was done at EA Digital Illusions CE AB (DICE). The company is situated in Stockholm and has around 270 employees, making it the largest game company in Sweden. DICE is most well-renowned for the Battlefield franchise but has also developed games like Mirrors Edge and the multiplayer part of the controversial Medal of Honor. DICE is currently developing the next generation of the Frostbite game engine which will part of the highly anticipated Battlefield 3 game.

1.1.2 New Technology
During the fall of 2010, both Sony and Microsoft introduced new interaction devices to their respective game consoles, the PlayStation Move for Sony’s PlayStation 3 and Kinect for the Xbox 360. These are both devices that can be categorized as motion based input devices. Unlike using a gamepad for interacting with games, which relies on joysticks and buttons, these new devices are based on movement. The first one of them, the PlayStation Move, developed by Sony, is similar to the Nintendo Wii Remote but uses different technology. The Move is used in conjunction with the PlayStation Eye camera to locate the controller in space. The other controller, competing with the Move, is the Kinect. The Kinect is not a hand held control but a device featuring a camera, infra-red sensor and a microphone array which enables you to interact with your Microsoft Xbox 360 using both movement and vocal commands.

1.1.3 The FPS Genre
One of the most demanding types of games is the First-Person Shooter (FPS). It is demanding both on the player and the interaction devices used. These types of games are often very fast paced and require quick response. The most dedicated gamers’ choice has always been to play with mouse and keyboard, most likely on a PC to achieve the precision needed (Cummings, 2007). The same games are often also played on consoles with the use of gamepads. The gamepad may give you a little bit more relaxed style of gaming but these players are still very competitive.

1.2 Objectives and Aim
The purpose of this thesis is to explain how motion controllers can be used for interaction in first-person shooter video games. This report will answer the question of how suitable it is to use motion controllers for this particular type of game. Is it possible to use the new technology and still maintain the feeling of playing a FPS game? The idea is to compare the motion controllers to a gamepad.
1.2.1 Problem Definition
The big question is what interaction techniques should be used for the two types of devices; this is not clear since the hardware is new. The first FPS games which are coming out choose to stick with an interaction technique seen before on the Wii. This technique which can be labeled Point-and-Shoot, builds on the idea of aiming on the screen wherever you want to shoot. Why this interaction technique might not be the best will be discussed later in the report. Another difficulty is that from a game developer perspective, one wants to give the player the same general feeling of the game regardless of the player is using a motion controller or a gamepad.

1.2.2 Additional Questions
The two following questions are also central in designing and verifying interaction techniques for these new motion based controllers.

- Is it possible to use the new controllers without losing the feeling of playing an actual first-person shooter game?
- Is it possible to use the motion controllers in a more effective way, compared to gamepads?

1.2.3 Delimitation
- FPS is the only game type that will be considered in this thesis.
- There will be no complete comparison between all common game input devices. Thus this will exclude both Nintendo and Sixense controls.
- The thesis will focus more on using the motions based input controls in an efficient way, than the actual gameplay experience in terms of immersion or flow.

1.3 Structure of the Thesis
Following this section there will be a chapter covering the background and some theory related to motion based controllers; various topics will be addressed, from the devices themselves to guidelines and ergonomics. In the Interaction Design chapter some initial ideas and decisions will be presented, it will show what was learned from the prototype phase and finish by describing the implementation itself. Next, the Test Method chapter will explain the methods for evaluating the implementation. After this the sections Results, Discussion and Conclusion will follow. At the end of the report (chapter 9) there is a short glossary explaining a few words used in the thesis.
2 Background and Theory

In this chapter the theoretical background to the area of motion controllers is presented. The following content is what was identified as the most important during the literature study, which was performed in the beginning of the project.

2.1 Related Work

Since there is not much written explicitly about the PlayStation Move or the Kinect, the related work consists of research of a more fundamental nature. Examples of this kind of research are studies of other similar devices such as Wii Remote, 3D user interfaces and also taxonomies of various controllers.

2.1.1 Engagement

There has been much written about how players experience computer games. For example the term Immersion is used to describe how deeply the player is engaged in the game. Jennett et al. discusses Immersion in the terms of being “in the game” (2009) and finds some contradiction with the classic view if immersion grounded by Brown and Cairns. In similarity to how immersion is used in video games the term presence is widely used in virtual environment (VE) systems.

Kien (2009) also discusses gaming experience with the adding of the concept of Flow. What is maybe most interesting is how the inclusion of movement in a game increases the players engagement. Why that is the case is not explained here. The experiment made by Bianchi-Berthouze et al. suggest even though the players in the research were rather few it seems that movement increases engagement even when the movement itself is not task specific. (Bianchi-Berthouze et al., 2007)

2.1.2 Movement

Motion input in games is far from new, but have existed in both the realms of virtual reality and console games. With the origin of the Nintendo Wii the motion based input world has experienced a slight revolution, bringing games to new types of players. Three examples of related motion based input applications are displayed below.

Kratz et al. (2007) created a game with 3D gesture recognition called Wiizard, obviously using the Wii Remote as input device. The recognition algorithm was based upon hidden markov models and achieved a very high recognition rate.

Using three cameras Schlattmann et al., in 2009, implemented a bare hand tracking system for use in computer games. The system was used for three different games: a racing game, a flight simulator and a first-person shooter game. For the FPS game they tried both a one- and two-handed control scheme. It showed that the one handed control was more intuitive.

A different approach to the FPS genre was taken by Slomp et al. (2007). They developed what they called a point-and-shoot technique for 3D immersive virtual environments. They used a combination of head mounted display (HMD), a data glove and a 3D tracker. The players participating in the evaluation of it, said it was entertaining, but it was not as efficient as using a mouse.
2.1.3 Taxonomies
Cummings (2007) describes the evolution game controllers, from very early computer games up until the motion controller, here represented by the Wii Remote. According to Cummings, motion controllers might just be the future of gaming if the Wii is successful.

To aid the development of active games, Stach et al. (2009) does a walkthrough of available hardware and classifies different kinds of input into six different categories: gesture, stance, point, power, continuous control and tap. This allows for studying the input decoupled from the hardware, which it is often very closely bound too, for a better understanding of the interaction taking place.

2.1.4 3D User Interfaces
A major work in the area is the 3D User Interfaces: Theory and Practice by Bowman et al. (2004), laying down a solid foundation while covering history, in- and output devices and techniques. The book also includes strategies for developing 3D user interfaces and the evaluation of them.

Wingrave et al. (2010) gives a guide on how to use the Wii Remote as a general 3D user interface control input device. There is a big cost benefit from using commodity hardware, instead of specialized and more costly alternatives.

2.2 PlayStation Move
The Sony PlayStation Move is a wireless motion controlled input system to be used with the Sony PlayStation 3 console. It is developed to give a realistic experience and full control of how the games are played, yet it is supposed to be easy to use (Sony 2010a). With this input system, Sony is competing even closer to the highly successful Nintendo Wii console. The Sony Move is geared towards all sorts of gamers, from casual gamers to more advanced gamers. The console will feature games for the PlayStation Move in a wide range of different genres. (Sony 2010b)

To achieve the high precision needed for the PlayStation Move, the control system consists of several different tracking components. The most obvious being the illuminated sphere on top of the PlayStation Move controller, which is tracked by the Sony PlayStation Eye camera. When several Move controllers are active at the same time, all of the spheres will be assigned different colors, so that the system is able to tell the controllers apart. If only one Move controller is used, it will when calibrating the device, choose a color that is clearly visible to the camera (Gamingbits, 2010). It is also possible to use the colors for a more esthetic purpose such as coloring the sphere red whenever enemies are close.

It is not enough to only use a camera for the tracking device; the PlayStation Move controller system also consists of an accelerometer, a gyro and a magnetometer to provide the system with all the data necessary to track the position and orientation of the device. (ifixit, 2010)

The PlayStation Move is a device that has six degrees of freedom (DOF). You get all these degrees by summing up the three dimensional axis, X, Y and Z and the rotation around each of them. As an example a normal mouse input device has only two degrees of freedom, the

Figure 1: Sony PlayStation Move controller

Figure 2: Sony PlayStation Move controller (alternate angle)
possibility to change values in the X and Y dimension. Six degrees of freedom makes these kinds of devices more complex than a traditional two dimensional device and allows for more complex movement.

To complement the PlayStation Move controller Sony have made available the PlayStation Move Navigation controller. The Navigation controller resembles the left half of the Dual Shock controller but is designed to be more comfortable to hold with only one hand. The Navigation controller can for example be used in games which require character navigation. (Sony, 2011)

![Figure 3: Sony PlayStation Move Navigation controller](image)

![Figure 4: Sony PlayStation Move Navigation controller (alternate angle)](image)

The first shooter game to be released (in the fourth quarter of 2010) for the PlayStation Move is The Shoot. It is a game for one or two players, using the Move controller to point and shoot at the screen. The game is a so called “on rails” shooter which means that the players don’t need to navigate themselves. The avatar instead travels in a predetermined way, leaving it up to the player to focus on the aiming.

Socom 4, a PlayStation 3 exclusive game, is announced to have support for both the Dual Shock 3 and Move. (1UP, 2010) Even though technically being a third person shooter, this game is similar to a regular FPS game. It allows you to freely move around using the PlayStation Move Navigator controller (or one half of the Dual Shock 3 controller). You turn the virtual camera by aiming to the side of the screen.

The Move implementation in Killzone 3, an actual FPS game, appears to function in a similar way as Socom 4 does. These two games have an expected release date in the first quarter of 2011.

### 2.3 Isotonic Device

The PlayStation Move is also an isotonic device, meaning it moves freely without any resistance, in difference to an isometric device which you need to apply force to control it. A mouse device is also isotonic, even though there is friction, there is no force pushing it back to the origin. For a more thorough explanation of isotonic and isometric and also examples of different devices see Zhai (1995). When using a device there is a question of how to translate the physical movement into virtual movement on the screen (i.e. in a game or some other 3D user interface). Zhai experimented with the two most common approaches: position control and rate control. Position control means mapping the input from the device to the cursor in a way directly corresponding to the position of the device (like using a mouse to control the cursor on a standard desktop computer) whereas rate control means controlling the cursor speed depending on how much you offset the device (imagine the cursor on your computer moving faster in correlation to how far you move your mouse from an imaginary center). The experiments by Zhai show that isotonic devices together with position control and isometric devices coupled with rate control surpasses the other two possible combinations. Since the PlayStation Move is an isotonic device this suggests it would be preferable to use position control. One downside with that combination is that it might be more prone to cause fatigue when it is used for a long time. (Zhai, 1995)
2.4 Microsoft Kinect

In contrast to the PlayStation Move, Microsoft’s controller system only consists of a camera, a depth sensor and a built in microphone array (Kotaku, 2010). This is a new take on motion based input not seen before in commercial game input systems. This separates the Kinect from both Nintendo Wii and PlayStation Move, since these two systems use handheld devices. The idea now is to use your whole body – your arms, legs, feet and hands et cetera – to control games. Without any buttons to learn, or device to control, it is supposed to be more intuitive to play games. Using natural movement also decreases the time learning how to play the game. (Xbox.com, 2010)

![Figure 5: Microsoft Kinect](image)

The system is capable of tracking full body movement as well as recognizing gestures, telling different people apart and responding to voice.

The Kinect seems very popular amongst the developers of dance- and fitness games. As several of these kinds of games have been announced, i.e. Dance Central, Dance Dance Evolution, EA Sports Active 2.0 and Zumba Fitness. Other games coming to the Kinect is the Kinect Sports and Child of Eden. The last one mentioned is a kind of on rails shooter that aims for a multisensory experience bringing movement, music and graphics together (Ubisoft, 2010).

2.5 First-Person Shooter Games

The game Wolfenstein, from id Software, is often acclaimed to be the first true member of the genre that is first-person shooter (Bryce and Rutter, 2002). The game was released in 1992. There were earlier games using the same perspective but they only used graphics consisting of points and lines. Wolfenstein was the first to situate the player in a room where at least the walls had some sort of texture. The game allowed you to shoot enemies and travel thru doors to different rooms. This laid the ground for several games following in the genre. You could walk back and forward, turning left and right using the arrow keys and fire your gun with another key on the keyboard. There was no possibility to aim high or low, but that changed in id Software’s new game Doom, which came out one year later. Doom had a big impact, in both being a high controversy game (as were Wolfenstein) and making the FPS genre popular among people. With the invention of mouse look, the game Quake, also by id Software and released in 1996, set the standard for how FPS games are played even to this day. (Cummings, 2007; Wikipedia 2010b; Wikipedia 2010c; Wikipedia 2010d)

The basics for a modern FPS game is to place the virtual camera in the same place as your avatar’s eyes and let the player control the view with the mouse (or gamepad). Usually a crosshair is rendered in the exact middle of the screen allowing the player to aim by moving the camera. This means that if you want to move in a direction you are not looking at, you have to strafe or move backwards.

FPS games are today played on both computers and gaming consoles, with either a mouse-keyboard combination or a gamepad (mouse and keyboards being more common for computer.
Background and Theory

With the mouse, you change the angles of the camera up/down and left/right; this is known as mouse look. On the gamepad, this is handled by one of the joysticks (usually the right one if you are right-handed). Walking used to be done by using the arrow keys but nowadays it is much more common to use the W, A, S, and D keys on the keyboard, allowing for a better reach to nearby buttons (Cummings, 2007). On the gamepad, you simply use the other joystick. Firing the weapon, which is a central part of any FPS (per definition), is done by the first mouse button or a trigger button (or equivalent) on the gamepad.

2.6 Movement in Games

Motion in games is experiencing a big boom with the success of the Nintendo Wii and the trend of natural input devices such as microphones and guitars in games like Band Hero and Rock Band. The goal for using motion input in games or 3D user interfaces is often to be able to interact in a natural way even if this is not always the case. The benefits of using natural movement as input are that it is easier to learn, gives you a more emerging feeling, and there are also potential health benefits which many new games will take advantage of. As already stated, dance, fitness, and sports game is very common among games which features motion-based input. LaViola writes that the reason for the emerging of motion-based input in commodity game consoles depend on three things: the consoles competing with arcade games (arcade games had motion-based input), game controllers were getting more and more complex, and the next step to improve gameplay was to use motion controllers and the development in hardware (sensors) has finally come far enough (LaViola and Marks, 2010).

There are two different approaches when implementing motion interaction in games: motion recognition and motion tracking (Chakravorty et al., 2010). Recognizing movement is to figure out what the player is doing whereas motion tracking is to follow the user’s movement and for example map them to player input. One example of motion recognition is the gesture recognition in the Wizards game by Kratz et al. (2007). Another example but on motion tracking is the game Tumble where you can build towers of blocks by directly controlling them with six degrees of freedom. It may be necessary for some games to combine these two methods.

A difficulty with the motion tracking method is to infer when a motion starts and when it stops. Two techniques can be applied, either player-driven or game-driven segmentation. The player-driven technique means the player tells the game when the motion starts or stops for example by holding down a button. This is an easy way to do it but it might not be appropriate for some games. The other way is to have the game figure out for itself when this happens, which is probably harder to implement. One way of simplifying the problem is to tell the user when to start the motion.

(Chakravorty et al., 2010)

When using motion tracking and realizing we cannot reach far enough, for example it is not possible to move a cursor to a specific place in a 3D user interface, then there are two ways to solve the problem. You can give up the one-to-one linear mapping and replace it with a polynomial mapping or similar. The other way is to use a clutch: a button that you can press (or stop pressing) to declutch and move your input device to a more comfortable position, then disengage the clutch to continue the motion (Bowman, 2004). The latter approach is similar to how one lifts a mouse device and moving it to prevent running out of space on the desktop. It is important to understand that clutching is really an unwanted motion as it takes time and focus from the actual task.
2.7 Guidelines for 3D User Interfaces

Some work has been done to develop guidelines to simplify when designing 3D user interfaces. Many of these guidelines might not be possible to translate directly to game applications but are certainly worth considering.

Teather and Sturzlinger suggest eight guidelines for “designers of games and virtual environments for developing intuitive 3D manipulation techniques”. They will be listed here and very briefly explained.

1. Avoid floating object. Since objects seldom float in the real world they should not do this in a virtual world.
2. Object should not interpenetrate each other. Using the same argument as above such things that do not occur in the real world should preferably not occur in a virtual world since users may find it confusing.
3. Support relative positioning of objects by bringing them in contact with one another. A natural way of positioning object is to slide them on a surface.
4. Only visible object can be manipulated. If the object is not in view the user probably do not want to modify it.
5. The most important cues for judging 3D position in real scenes are perspective and occlusion.
6. Avoid technical computer graphics techniques such as “handles” and “3 orthogonal views”. More intuitive techniques for manipulation should be used and a perspective view more efficient for a novice user.
7. In general, 3DOF or 6DOF input devices provide less precision than 2DOF input devices. Holding a device freely in front of you will induce more jitter than using a 2D input device such as a mouse.
8. Use the entire area of visual overlap of the moving object with the static background scene when deciding the position of the object.

(Teather and Stuerzlinger, 2007)

LaViola have compiled guidelines on how to use the Wii Remote for universal 3D interaction. Although general and for the Wii Remote they might be important for other applications as well (LaViola and Marks, 2010). The guidelines are divided into: selection, manipulation, travel, system control and symbolic input. This classification is the one from Bowman but leaving out the area of way finding as it is more of a cognitive process than related to the input system (Bowman et al., 2004).

2.8 Performance

Several sources have come to the conclusion that a 2D device outperforms the more complex devices in three dimensional position techniques (Teather and Stuerzlinger, 2007). Especially the normal mouse device which performs very well in games compared to other devices. This may, to some degree, be because of the mouse’s commodity status. Most people have used a mouse even if they do not play games. Isokoski and Martin (2007) performed a test comparing the Xbox 360 Gamepad with a mouse-keyboard combination. It showed that when aiming with the mouse the player was twice as effective compared when using the 360 Gamepad. This seemed to be true even when putting the players to more training.

Another aspect of this is the use of natural movement in games. Research shows that non-natural movement can have a much higher efficiency (McMahan et al., 2010). The research was done with the Mario Kart game to the Wii console and different interaction devices and techniques were used. The Wii Remote was used both as a standalone motion controller and in place in a plastic wheel. The Wii Classic controller and GameCube controller was also used.
The Classic controller and the GameCube controller performed better than the natural movement controllers.

Even though more complex game controls are often outperformed by simpler ones, natural or higher degree of dimension controllers are considered more fun for example in the experiment mentioned above (McMahan et al., 2010) and in work by Teather and Stuerzlinger (2007).

2.9 Ergonomics

When developing techniques for interaction devices, the question of ergonomics must be thought about to avoid injuries and unnecessary fatigue. This is especially important when dealing with devices that are held freely in front of you, or when dealing with full body movement. It is important to avoid unnatural motion that causes strain. As Bowman et al. (2004) writes strain can lead to stress injuries and therefore devices of this kind should be lightweight and require little training. Schlattmann et al. (2009) stresses the importance of that the basic movements of the interaction technique are comfortable, especially flexing the wrists excessively can cause problems. In his experiments on controlling a FPS game with bare hands, he found out that using two hands instead of just one causes more fatigue.

When designing for these sorts of applications one must also consider the imperfections of the human body. For example involuntary shaking when trying to do exact motions, like holding your hand in front of you keeping it dead still. Hand tremor can be reduced by using some sort of support, according to Steurzlinger and Wingrave (2010). They use the example of a sculptor that can support his hands on the object itself which he is working on. The precision of the interaction technique will also depend on the design of the device. This is because the design will determine how the device is held and thus which muscles in your body you depend on controlling it. Controlling a device only using large muscle groups will cause less precision than for example manipulating it with the fingers (Zhai, 1995).
3 Interaction Design

This chapter focuses on the design of the interaction technique, from initial ideas to the implementation. The goal of the chapter is to motivate and explain the decisions made in the design phase.

3.1 Decisions

Choosing a fundamental interaction technique for this project to build upon, means considering both position control and rate control presented in section 2.3 and also the point-and-shoot technique briefly mentioned in the Problem Definition 1.2.1. The choice between position- and rate control is simple for both PlayStation Move and the Kinect, as it is very hard to reposition the device (or hand) to an imagined origin with great precision, when trying to stop the motion. This is something that also was confirmed while doing prototyping. With the point-and-shoot I simply refer to the technique which many Wii games use and not specifically the interaction technique developed by Slomp et al. briefly mentioned before. The point-and-shoot technique is used in such games as The Conduit and Modern Warfare: Reflex. In addition new games for the PlayStation 3 like Socom 4 and Killzone 3 will use the same technique1 (YouTube, 2011a; YouTube, 2011b). The interaction technique is very intuitive because of its simplicity. Having a device with a trigger button, the affordance from using it as a gun, aiming at the point you want to hit, is powerful. This works for on rails shooter games (as long as the system is calibrated) where the camera view is handled by the game itself. However when using the point-and-shoot technique in a game with freely moveable camera its biggest flaw is obvious: the cumbersomeness of actually rotating the camera in the desired direction. This is most times implemented by aiming to the sides of the screen. When you do, the camera starts to move in that direction, the further to the side you aim, the faster the camera will move (in a rate control manner). This is not an ideal situation as it takes focus of the game and makes it hard to turn around in an accurate way. There is also a possibility of rotating the camera by mistake, when trying to shoot at a target that is slightly off center, which can be very frustrating.

The point-and-shoot technique also differs in an important way compared to a normal FPS game. To help the player aiming at the screen, there is usually a crosshair rendered to the point which you currently are aiming at. As a contrast to this, in normal FPS games the crosshair is always fixed in the center of the camera, allowing for a quick way to turn around, since the action of looking around or turning, is the same as aiming. Having the crosshair fixed in the center of the camera has for long been fundamental, and it strongly contributes to the feeling of a FPS game.

As we learned in the Background chapter earlier: having fun while playing games is not necessarily the same thing as being effective. Using the point-and-shoot technique when playing on rails shooters is fun because of its intuitiveness and high accessibility: almost anyone can play it - young or old, experienced gamer or not. When applying the point-and-shoot technique on a FPS game however people might still find it fun but because of the nature of FPS games the technique will not be very effective. Because of the theses factors I have chosen to go forward with the position control technique.

Since the design of the interaction technique is made in the context of only one screen, there is an obvious problem when using position control: if you try to turn around far enough you will not be facing the screen anymore. This would not be a problem if you had a special room where

1 The two games mentioned are not released while writing this thesis. The information is taken from videos showing the unfinished games.
the game could be projected on all six sides. Since this is not very common, the solution to the problem is instead to use a clutch of some sort, allowing for the repositioning of the device to a more comfortable position. The concept of using a clutch will be discussed in following section 3.2 Prototypes, as well as in section 3.3 Implementation.

### 3.2 Prototypes

To learn about the two types of motion controllers a few prototypes were constructed. These were built to explore the possibilities of the controllers and get a feel for what could work in a real FPS game.

#### 3.2.1 PlayStation Move

The prototype application for the PlayStation Move was built upon an existing demo. The demo itself is not important but it provided a free flight camera and a sense of direction. The gamepad input in the demo was replaced by the input from the PlayStation Move. The prototype could switch between using raw values from the device internal sensors, and using the interpreted values available when using the PlayStation Move together with the PlayStation Eye. The camera was controlled by rotating the device left/right and up/down, thus affecting the yaw and pitch respectively, as long as a specific button (clutch) was pressed. Moving the camera forward/backward and strafing was handled with the joystick of either the left side of the gamepad, or the Move Navigation controller.

One of the first things that were noticed, while running the prototype on the PlayStation 3, was that the gyro in the Move seemed to have a very high accuracy, consequently it seemed promising that the Move could be used as a game device, beyond the point-and-shoot technique. Controlling the free-flight camera when the application was running with a frame refresh rate of 60 hertz, gave a smooth and fluid experience. However at this point, the input was affected by the drifting of the gyro: even though holding the device completely still the input wandered off in some direction. When using the Move together with the camera, the problem with drift is eliminated. In addition to this, the camera motion felt less fluent but more exact, however this was a subtle difference. The act of calibrating the PlayStation Move did not have a major effect on input performance; this is probably due to the fact that the positional data was not used in the prototype, only velocity data was.

Since the precision of the PlayStation Move is high, the hand tremor was propagated thru and noticeable when trying to hold the camera still or doing fine movement. This is represented in Figure 6 below, showing the input x from the device directly mapped to the output f(x), which is the same as movement on screen. Even with the crude method of ignoring small changes in the input, seen in Figure 7, a much more comfortable camera control was achieved.

![Figure 6: Ideal input curve](image)

![Figure 7: Dead zone](image)
To be able to do full rotation a clutch is necessary, however it is not obvious how to design and implement this. The first question is: what button do I use? And the second: do I press it or release it? The PlayStation Move control has two buttons that are suitable; actually one of them is the trigger and the other one is the big Move button on top of the device. There are also four small buttons around the Move button which could be used, but I would argue that in comparison to the small buttons, it is much more intuitive to use the Move button because of its position and size. Furthermore the question of pressing or depressing the clutch was tried out. The results will be discussed in section 3.3 Implementation later.

Using the PlayStation Move Navigation controller instead of a Dual Shock 3 for moving around is pretty much the same, except that the Navigation controller is easier to get a good grip of with only one hand. When using the Dual Shock 3 it is probably preferred to support it on your lap. Personally I find the d-pad on the Dual Shock 3 easier to reach then the d-pad on the Navigation controller.

### 3.2.2 Kinect

The prototype for the Microsoft Kinect also derived from a preexisting demo. This demo however already featured a FPS camera and a room environment. The demo allowed for something close to a FPS style control but only allowed for rotation in one axis (steering left/right). The demo was modified to use a different FPS interaction technique. The new technique was based upon moving the camera with the left hand and shooting with the right. When the left hand was moved far enough in front (or to the side) of you the steering was engaged, directly controlling the camera. When wanting to rotate the camera to the left you simple moved your arm/hand to the left et cetera. The clutching of the movement was achieved by retracting your hand towards your body, and to declutch you simply extended your hand again. To fire the virtual weapon a punching movement was used. For moving the camera two different techniques were tried: stepping and leaning. Here stepping means that you physically move (take a step) in the direction you want the camera to move. The problem here is when you want to stop it is hard to find your original position. At least this is true if you want to do exact movement and don’t want move several meters in the physical world. The leaning technique do not suffer this problem as much, since after leaning in any direction it is natural to return to an upright position. The problem with leaning is that the accuracy is low due to the limited range of motion, or simply put: if you lean to much you will fall over. Mixing the two techniques in a way that forward/backward movement was handled with stepping, and strafing with leaning side to side seemed to be a reasonable solution.

![Figure 8: Kinect prototype, left: top view, right: side view.](image)
The small sphere in the center represents the shoulder joint. It is connected to the elbow joint which then connects to the hand. Around the shoulder there is a cylinder, extending your hand outside the cylinder engaged the aiming.

The interaction technique used in the prototype more or less employs the whole body and it is clear that you have to be standing up to be able to move the camera. Using large body movement is as previously stated not as precise as using smaller muscle groups. This and the fact that there is really no way to support your hands/arms make the need for smoothing evident. If one instead directly controls the camera with the unsmoothed values the camera will be twitching a lot, similar to how in the Move prototype when there was no input threshold (dead zone). However in the case with Kinect it is harder to get a good compromise; using too much smoothing will make the camera control imprecise.

One of the biggest disadvantages with using the Kinect for FPS games today is that the tracking becomes more unstable in some poses, for example when the hands are very close to each other and in front of the body, which in the technique used in the prototype, happened fairly often. In addition to this, using the prototype more than a few minutes can be extremely fatiguing. The complexity seemed much higher when controlling two axes instead of just one (left/right).

### 3.3 Implementation

This section describes the design decisions that were implemented in the Frostbite 2 game engine. Support for the PlayStation Move was implemented, building on the camera control used in the prototype. It was decided that the Kinect would not be implemented into the game engine because of the hardship of using the Kinect system as a direct FPS controller. However there is a section later in the Discussion chapter (6.3) mentioning alternative approaches using Kinect in a FPS game.

#### 3.3.1 PlayStation Move

Coming from the prototype the confidence was high that the camera control was suited to be used in a game. One big difference is that the camera is now attached to a soldier, hence to the ground, and not flying free. Since the PlayStation Move controller is now implemented in a real game engine it is possible to map all different FPS specific actions to the control, such as aim, fire, jump et cetera. As the engine is currently is running at 30 Hz frame rate which is half of what the prototype ran; there was a concern this would affect the precision of the Move in a negative way. However this did not cause any problems, in addition doing the sampling twice a frame instead of only once did not improve the feeling of the controller.
3.3.1.1 Aiming

The control of the yaw and pitch of the camera, the aim, needs to be done in such a way that it is not affected by hand tremor. In the prototype this was done by ignoring small changes in the input. Another way to do it would be to use filtering, to make the current input sample depend on earlier sampling; however this makes the input less direct and responsive, this was therefore only briefly tested. Instead the focus was to remove the small changes in an effective way.

One way of doing this is to use squared input curves (Figure 9), making small values even smaller. This is great in the sense that it creates a seamless border between no input and very little input. On the downside this works like acceleration on the device, which makes the camera turn a different number of degrees depending on the how fast the device is rotated. In a way this feels inconsistent.

To achieve a more consistent feeling, the solution from the prototype was brought back. This solution is similar to a dead zone that is normally used on a gamepad to account for the fact that when releasing the joysticks on the gamepad they never return to the exact same place. This will stop the gamepad to output offset values even though the joysticks are not touched. This works best when the dead zone is normalized. This means that the output starts at zero exactly on the border to the dead zone and not some arbitrary value. This is represented in Figure 10. When normalizing the dead zone of the Move controller it gave a much smoother experience, but still, very small input values are cropped, which can be annoying when trying to perform very precise movement. One advantage is that the cut-off point for the dead zone is very easy to tweak for individual players, since it only relies on one parameter.
As previously mentioned when experimenting with the input, it was evident that squaring the input function gave a very good result for small movements, eliminating hand tremor efficiently and still preserving accuracy. However, as stated, it makes the control feel inconsistent and it is hard to find a good compromise when considering both small and larger movements of the control. Either the small movement results in too low aiming speed, or the large movement results in too high speed. The idea then was to take the best of both worlds: the accuracy in the small movements from the squared function and the consistency of a linear function whenever there is a larger movement. The two functions are joined together where the derivate of the squared function coincides with the linear function, as showed in Figure 11. This idea of using a ramped dead zone showed to be a stable solution, improving the accuracy in small movements compared to only using a linear input function. The downside is that it is a bit more complex to tweak for different settings in aiming speed. It is important that the player cannot feel where the transition from a squared function to a linear function is taking place; if that is possible it will again feel inconsistent. This is mostly a risk when using very high aiming speed setting, which means the linear curve, will be very steep. If this is a problem one could use a higher order polynomial function for the ramp part.

![Figure 11: Ramped dead zone](image)

### 3.3.1.2 Clutching

Choosing between the Move button and trigger to use for clutching is not easy because there are pros and cons with using both of them. Using the Move button as a clutch leaves the trigger for firing the weapon, which is highly intuitive and therefore desirable. However firing is a state that is either on or off in most games, and using the trigger, which is range sensitive, is a bit of a waste. Instead using the trigger when aiming, allow you to make smaller camera movement when lightly pressing the trigger and normal camera movement when the trigger is fully depressed. Further on this allows for pressing other buttons on the Move controller when aiming, which can be practical for example jumping or melee attacking. It might also be more fatiguing to press the Move button to aim in comparison to simply holding down the trigger. A downside is that furiously pressing the Move button to fire in the heat of the moment may induce more recoil: accidently shaking the controller when pressing down and releasing the button. This could propagate thru to the aiming, making it harder to be precise. If this would show to be a real problem it could of course be smoothed out via filtering. Due to the intuitiveness of using the trigger for shooting, in this implementation it is up to the player to choose which is preferred.

### 3.3.1.3 Buttons and Gestures

One of the biggest questions is how to actually assign an action to each button on the controller. Since the PlayStation Move is a motion based input device of course there are possibilities to assign the actions to movements of the controller, and perhaps with assigning different buttons to press, to start recognizing other gestures the possibilities are even greater. Since one of the aims for this thesis is to focus on using the Move in an effective way, this means the most
important thing is that the camera control or aim is exact and nothing interferes with it. This means using buttons when applicable instead of motion because of the fact that even a simple gesture can be performed in error while the press of a button is always easier to perform correctly. One could however argue that in the heat of the moment an intuitive gesture could be easier to perform instead of trying to remember which button to press. There is also a possibility to use redundancy in the sense that a gesture and a button could do the same thing, for example jumping could be achieved by pulling the device upwards or pressing the X button. This gives the player the opportunity to choose whichever is most comfortable, depending on the situation or preference. The downside is that if the available actions are great it might not be possible to do this. If it is not and an action only can be performed by a gesture it is essential that this gesture is easy to perform and that it is not performed by mistake. For this implementation seven different gestures were made available. They were: quick movement up, down, left, right and forward, wrist flick left and right (rotation of the device quickly) as seen in Figure 12.

![Figure 12: Gestures](image)

### 3.3.1.4 Control Schemes

When designing the control scheme it is important to consider various things. Five different areas were identified as important and worked as a foundation for the implementation. The areas are: use of gestures, available actions, familiarity, logic and comfort.

The first of the two are pretty clear after reading the previous section on buttons and gestures. Which gestures are available and which of them will work without interfering with the basic handling. The number of actions is a big deal, trying to add an extra controllable action into an existing control scheme may compromise the whole layout and make you need to rethink all of your decisions. In addition to this a higher action count will result in a more complex and less intuitive control scheme and in worst case more frustrating gameplay.

The control layouts in the FPS genre are often quite similar between games. At least some actions are handled the same when comparing different games. This means a player will have no problem transitioning to a new game, since he or she only has to remember a few different actions and how they are laid out on the control. This means it is probably not a good idea to rework the entire idea of how a FPS game is played, unless the game is unique and it conforms well to the other areas mentioned in this section.

The area of logic goes hand in hand with the above. A layout that appears logical to the player will be easier to remember for the user since it allows the player to make a mental model of the control. If the player feels he or she understands why an action was mapped to a specific button or gesture it is a big advantage.

Comfort and ergonomics is also very important. Such actions that are performed frequently or for a long time must be easy to perform, as discussed previously in the Background chapter. For example pressing a button with your thumb for a long time may cause discomfort. It is also important that you can perform many actions at the same time, for example using the circle
button to interact with something at the same time you are able to fire using the Move button, which probably can be learned without much difficulty, however if another button was used for interact you would not be able to do both actions at the same time.

3.3.1.5 Finished Design

The finished design consists of several different control schemes, depending on the type and number of actions available. The default control scheme will be listed and discussed below. The rest of the control schemes will be available in the appendix.

<table>
<thead>
<tr>
<th>Action</th>
<th>Button/Gesture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aim</td>
<td>L1</td>
</tr>
<tr>
<td>Fire</td>
<td>MOVE</td>
</tr>
<tr>
<td>Grenade</td>
<td>L2</td>
</tr>
<tr>
<td>Melee</td>
<td>TRIANGLE/Gesture: Forward</td>
</tr>
<tr>
<td>Jump</td>
<td>CROSS/Gesture: Up</td>
</tr>
<tr>
<td>Reload</td>
<td>SQUARE/Gesture: Rotate CW/CCW</td>
</tr>
<tr>
<td>Interact</td>
<td>CIRCLE</td>
</tr>
<tr>
<td>Sprint</td>
<td>L3</td>
</tr>
<tr>
<td>Crouch</td>
<td>CROSS (Hold)/Gesture: Down</td>
</tr>
<tr>
<td>Scoreboard</td>
<td>SELECT</td>
</tr>
<tr>
<td>Menu</td>
<td>START</td>
</tr>
</tbody>
</table>

Notes: The d-pad has been omitted as it may be used just as on a regular gamepad. The buttons L1 and L2 in Table 1 are found either the Dual Shock 3 or the Move Navigation control and not on the PlayStation Move itself. Calibration is handled under the menu and not directly on the select button as in some Move featured games. The implementation also features force feedback. The Move and trigger button are optionally interchangeable, which means if you prefer to use the trigger for firing instead, you can use the Move button for clutching.

The control scheme has been designed with the five different areas mentioned before in mind. A great deal of the assignments should be familiar to anyone playing FPS games on PlayStation 3. For example the aim, jump, reload and sprint all are where they are expected. One thing that differs from the Dual Shock 3 gamepad is that there is no R3 button, for that reason the crouch has been assigned to holding down the cross button briefly. Clicking it fast instead results in jump. The crouch action can also be accomplished via gesturing.
4 Test Method

This chapter will explain the methods used for verifying the implementation discussed in the last chapter. Two different tests were used: one quantitative method and one that were more qualitative in its approach.

4.1 Test Level

As a first way to verify the implementation, a test game level was built. The level was in the form of a normal playable level but instead of shooting regular enemies you were supposed to go around on a path shooting at circular targets. These targets appeared in front of the player as he got close enough, spawning one by one or in groups. In total, there were thirty three targets which disappeared as soon as the player hit them. The time was measured per lap, but the timer was only running when there were targets active. Whenever a target or a group of targets spawned the timer started and it was stopped when the targets were destroyed. When a lap was finished the time was logged and the timer was reset. This was to focus more on the aiming, instead of running thru the level as fast as possible. There was an idea about using several timers on each lap to record more detailed data on specific areas in the level. This could have been interesting since the level had some different types of challenges. However this was discarded due to the increase in complexity displaying the data and also the increase in time needed to build the actual level.

Figure 13: Test level – First target

The targets always appeared in the same order and spawned in on the same position every time. This made the level static in a way and introduced the factor of learning. As the player played the level several times he might remember where some of the thirty three targets appeared. The most important here were to have to player moving and shooting at the same time as he would in a real game. It would have been simple to have the player standing around and shooting at targets randomly spawning around him, but that would have taken away a big part of the feeling of playing a FPS game. One could also imagine a level were targets randomly appeared in front of the player as he moved along the path, but this would probably made the level feel unnatural
since very few AI enemies and even fewer human controlled enemies appear randomly in such fashion. A much better approach then might have been to use alternate spawn points for every target, which would allow the targets to spawn in different but equally difficult places. This would however also have increased the complexity because of the number of targets. In addition to this, it is important to remember that playing this kind if test level is not like playing a normal game. This is because of the rules are fundamentally different. Setting aside the learning process it is however a reliable way to measure time. And since the level was built both for PC and PlayStation 3 it is possible to compare aiming times between mouse/keyboard, gamepad and PlayStation Move.

Eight test subjects participated in playing the level. They were all male and very familiar with playing FPS games. Almost all of them were experienced in playing FPS games with a gamepad. None of them had tried out this PlayStation Move implementation before. One of them was left handed. The subject all played with the same automatic assault rifle with unlimited ammunition. This was to avoid the need of reloading the weapon while targets where active.

The participants were orientated on what would happen, such as targets appearing along the way. It was said that the level being quite short were to be played five times on each controller configuration and also that the participant had a minute or two to accustom the self to each controller before starting the level. After this they were clearly instructed to: follow the path, destroy targets as soon as they appear and to focus more on aiming and shooting than running very fast thru the level.

All of the players started out with the mouse and keyboard combination on PC. Half of the subjects continued with the gamepad and finished with the PlayStation Move. The other half did the opposite and played the level with the PlayStation Move before the gamepad. This was to reduce the effect of learning the level in the average result. This of course means that the mouse/keyboard combination is not really comparable to the other two, but works more like a reference.

### 4.2 Playtest

The other method for testing the implementation was to run a multiplayer game playtest. This made it possible to get a feeling about how the Move-implementation worked in a real game environment.

The level selected for the playtest was under development and therefore not in its finished state. It had an urban setting and was relatively small which were appropriate for testing the PlayStation Move with only a few players. The game mode was set to Conquest, which basically means you are supposed to conquer some areas and defend them when the enemy attacks. The teams were chosen at random and also changed during the game. However the game mode and team/player points are not important here. What matters is the participants’ experience of the PlayStation Move controller.

The idea was to have eight people playing at the same time, four with gamepad and four with PlayStation Move for two times twenty minutes with a small pause in the middle to shift the equipment around. For practical reasons it was not possible to gather all players at one place, instead they were sped out in an office environment. The participants in this test was, as in the test level, very well experienced in playing FPS games with both mouse/keyboard and gamepad. Two of them had also tried the Move-implementation before.

The participants were instructed via email about the basics in setting up the camera, starting the game, calibrating the device and how to play with the PlayStation Move controller. The players were encouraged to try the setting which switched the aiming and firing buttons (Move-button and trigger).
After the playtest all participants were given a questionnaire to fill in. The full questionnaire can be found in the appendix. Most of the questions in the survey were to be answered by circling a number on a scale. The scale ranged from one to seven, giving the participant an option to circle the number four if he did not feel strongly about the question in any way. On a few questions there were also a possibility of commenting in own words. The aim of the questionnaire was to find out how high level of acceptance the players had for using the Move controller. The questionnaire had questions sorted under the following areas: player background, control scheme, efficiency, experience and physical.

The participants were instructed via email that they should circle a number in the scale to answer and that they could be anonymous if they wanted. The idea by having them filling in their name was to be able to correct possible mistakes afterwards. This was important due to the low number of participants.
5 Results

In this chapter the results will be presented. The data from the two tests were converted into graphs to make them easier to read. The results will be discussed in the next chapter.

5.1 Test Level

Only one of all the laps made in the test level was not finished. This and the fact that is it possible to make a mistake that gives you a bad time motivated the decision of taking the three best times out of five lap times and taking the average of them. This will also be discussed later.

The participants did five laps in the test level per controller configuration; hence they each did fifteen laps in total. The notation of the controller configurations in the diagram above are PC (mouse and keyboard, Gamepad and Move (PlayStation Move and PlayStation Move Navigation controller). An average value was calculated from the three best times per controller configuration and player. In Figure 14 these average times are presented.

In the diagram above the PC-times are quite even. There is a bigger variation in the gamepad and PlayStation Move times. One can also note that it is possible to achieve a low time even when playing with the Move.
Figure 15: Diagram showing the overall average of completion times

Figure 15 shows the same data as in Figure 14 but in a form that is an average of all the players. This means that the three bars in the diagram shows the average completion times for the level for each controller configuration.

The diagram above speaks for itself, however to interpret the diagram one should be aware about the background of the results. This will be discussed in the next chapter.
5.2 Playtest

The resulting graphs from each question in the questionnaire are presented below. As can be seen from in the following figures, only six people participated in the test. The results are in the form of histograms. The horizontal scale below all of them is the same scale as the participants used to answer the questions. The histograms show the distribution of the participants’ answers, to all of the questions that were in the questionnaire. The full questionnaire can be found in the appendix.

Figure 16: Mouse and keyboard experience in FPS games

Figure 17: Gamepad experience in FPS games

Figure 18: PlayStation Move experience (games in general)

Figure 19: Which button did you use to fire?

Figure 20: How did you like the control scheme in general?

Figure 21: For jumping, crouching, meleeing; did you prefer to use: buttons or gestures? (Leave blank if you used the trigger for firing instead)
Results

Figure 22: How often did you use the possibility of lowering the aiming speed by only partially depressing the trigger?

Figure 23: Acquiring targets

Figure 24: Aiming at moving targets

Figure 25: Aiming while rapidly firing

Figure 26: “I would rather play FPS with the Move than a gamepad.”

Figure 27: “I would rather play FPS with the Move than a mouse and keyboard.”
5.2.1 Comments

In the questionnaire there was room for commenting in own words on four of the questions. The comments pretty much reflect what can be seen in the diagrams above. For example several of the participants expressed that they were not very happy about how the gestures worked. The general opinion was that the gestures should have been better implemented; not as many; or simply removed. Another thing evident in the comments was the need for proper support when holding the PlayStation Move controller. The opinions were that the level of fatigue was dependent of how you sat while playing; resting the hand and controller against the leg worked fine; and it would not have worked for a long time without supporting the arm.
6 Discussion

This chapter contains a discussion about the results found from the test level and the playtest. It is then finished by a short discussion about the PlayStation Move in general and also some notes about alternative ways for using Kinect in a FPS context.

6.1 About the Results

6.1.1 Test Level

The results from the test level might not look very impressive, with the PlayStation Move average times roughly nine seconds higher than the gamepad counterpart. It is important to take into consideration that all of the participants were skilled in using the gamepad but had no experience whatsoever in using the PlayStation Move in a FPS game. It is with this in mind one should interpret the diagrams. It seems reasonable that at least some of the participants in the test would be more effective with the PlayStation Move than the gamepad, given some practice. For example look especially at Player 5, 6 and 7 in Figure 14, whom all have average times with the PlayStation Move that is close to their gamepad times. Player 5 even shows that it is possible to achieve a very low average time with the Move.

There were some problems due to the equipment and also with the test level itself. First of all when using the PlayStation Move it is of course important that the sphere on the controller at all times is visible by the camera. This had not been a problem until the actual testing, when the screen and PlayStation Eye camera was moved to facilitate the test and avoid the participants hitting something when using the controller. The camera was placed below the screen instead of on top of it to be sure that the sphere was visible at all time. This seemed to have the effect of making the tracking worse, since the distance from the sphere to the camera decreased and became too small in the test setup. It was not possible or desirable to sit much further away from the screen during the testing. This resulted a few times in the need for recalibration of the device in the middle of the level.

The problem with the level was that a couple of times a target was passed by unnoticed. This happened when the player came very close to some target, destroying it and then the next target spawned beside the player but out of view. This was probably due to the test subject running closer than necessary to the targets before shooting at it.

These two problems, with the camera and the level, should not have such a big impact on the result in the diagrams since the values are averages of the three best lap times.

6.1.2 Playtest

By having the participants rating their own experience of FPS games it might be possible to see consequences in how they answered the rest of the questionnaire. One idea was that highly experienced FPS gamers, playing both on PC and PlayStation, having little experience with the PlayStation Move would be less inclined to accept the Move controller as an suitable input device for FPS games. It looked like this could have been true with four people; three highly experienced gamers being skeptical about the PlayStation Move and one person, rating himself as less experienced, being positive about the Move controller. However two persons, rating themselves as the highest level of experience in gaming FPS games with both mouse/keyboard and gamepad, were very positive about using the PlayStation Move.
From the result is looks like using the trigger for shooting was preferred in comparison to using the Move-button. It seemed however that not everyone actually tried both settings. This of course reflects itself in Figure 22. Why this was not tried will be addressed soon.

The words “control scheme” in question 2.2 (Figure 20) may mean different things to different people. This is evident after reading the comments. Most of the participants commented on the whole experience of aiming, others on the gestures or only the buttons.

It is clear in Figure 21 that the gamers preferred to use buttons instead of gestures when interacting with the game. This is also apparent when reading the comments about the gestures. It might have to do with the fact that pushing a button is easier to always get right in contrast to a gesture, even how simple it might be. One person said he would not know when the gesture “kicked in”. This could be helped by improving the recognition of the gestures and also providing better feedback to the player.

The diagrams showing the perceived efficiency (Figure 23, Figure 24, Figure 25) look slightly positive. Many of the participants think that aiming is relatively easy and aiming at moving targets is a bit harder. To aim while firing rapidly is even harder. This is probably due to the amount of virtual recoil in the game which is hard to compensate for. It is also very clear that number of people answering the questionnaire was too few. Further on one should remember that the answers are not a comparison to gamepad, but the participant’s own experience.

Looking at the acceptance level of the participants in Figure 26 and Figure 27 is interesting. With the level of combined experience in PC gameplay of the participants and the effectiveness of using the mouse and keyboard witnessed in the test level, it is almost surprising to see anything but disagreement in Figure 27. The acceptance level comparing the PlayStation Move against the gamepad (Figure 26) is a bit higher. Again it is very clear here that the numbers of responses is far too low.

The rest of the result is much as expected with the exception of some participants thinking that the use of the controller was quite fatiguing and also discomforting in some cases. But then again considering that the PlayStation Move is an isotonic device and used with position control it might not be that surprising.

There is reason to think that a practical implication had some effect on the results for the playtest. The fact that it was not possible to gather all participants in one room to perform the playtest and instructing all of them at the same time, made it a bit harder. Instead the participants were spread out in an office environment and instructed via email. It would have beneficial to help everyone to get started. Changing the option to switch the trigger and Move-button as discussed before was not obvious and therefore probably soon forgotten even though explicitly mentioned in the instructions sent out.

### 6.2 PlayStation Move

The PlayStation Move offers six degrees of freedom and low latency, this and because of its precision and numerous buttons, it is a versatile device not only suitable for games using natural movements but also other genres. The implementation for this thesis does not fully take advantage of all of the Move possibilities, such as the position coordinates made available. This is due to the fact the interaction technique depends on the rotation of the device and not absolute position.

The biggest advantage of this implementation compared to a gamepad or using the point-and-shoot technique in a FPS game is that it makes it possible to turn around much faster, which is beneficial when an enemy is coming from the side or behind.

Using the PlayStation Move with a mouse analogy is interesting, using a clutch instead of lifting the mouse from the surface. The benefits with using a mouse is that you have constant support for your arm all the time, in contrast to the Move which you have to hold more or less freely in the air. This also means that you do not need to sit by a desk, instead it is possible to have a
more laid back gaming experience. The precision of the PlayStation Move cannot compete with the mouse and keyboard in a FPS game, but it makes a great alternative to the gamepad.

6.3 Kinect

With the prototype of Kinect as a FPS controller in mind, it did not feel encouraging continuing implementing it into Frostbite. As stated earlier, the main reason for this was the lack of precision when using full body movement as input, and also the technical limitations. Since time is always a factor it was decided to shift the focus in the favor of PlayStation Move. This does not necessarily mean that Kinect does not have any place in the FPS genre of games; on the contrary I will here present two possible alternatives of using the Kinect in such a game.

The first alternative is to relax the genre a bit and think more about the exploration in the game. The technique used in the prototype or similar techniques could be used for navigation, but it would be preferable to only control the camera in one axis (yaw). Fighting of enemies could be done in a way that focused more on reaction time or performing different gestures, instead of precise aiming.

The other alternative is to use the Kinect as a complement to enrich the FPS experience. The idea is to give the player a higher sense of immersion thru body movement. Since the player might prefer to sit down while playing the game as he or she is used to, this sort of involvement would focus on such things as arm and hand gestures, leaning and face tracking. One interesting feature would be able to issue commands using your voice in a FPS style game.

Using the Kinect as a direct FPS controller is problematic. People will probably not play FPS games with the Kinect until the genre itself becomes redefined or using such alternatives as the two listed above. This is only a question of being innovative and trying new ideas. The game Child of Eden might be a good start.
7 Conclusion

This chapter contains the conclusions drawn from the previous chapter and also recommendations for the future.

7.1 Kinect

Initially one of the ideas for this thesis was to be able to compare the Kinect and the PlayStation Move against each other, in for example the test level that was built, as seen in chapter 4. But since the prototype showed that the Kinect were unsuitable for this kind of gameplay, it would not have been a serious test, and therefore one can conclude that the Kinect fails to meet the demands of a high speed FPS controller.

7.2 Conclusions of the Discussion

It is hard to make any strong conclusions based on the work here due to the low number of test participants. However some trends are visible.

The interaction technique proposed here might not be as intuitive as the point-and-shoot technique, and since it depends on a lot of buttons, as many fps-games do (clutching, aiming, and shooting), there is initially a relatively steep learning curve. When progressing however, it is very much likely to be more effective with this technique than with the point-and-shoot technique.

Competitive gamers have very high demands on the interaction with the game. It needs to be precise and never failing. It would therefore be wise to minimize the amount of possible gestures in a FPS game to a few core gestures that are not likely to be performed in error. If gestures are necessary it might be a good idea to have a dedicated button to activate the recognition of auxiliary gestures.

One of the hardest problems when dealing with the PlayStation Move in applications that require precise movement is hand tremor. This is independent of the interaction technique used. In a FPS game it is important that the input is responsive and exact or it will be frustrating to play, the dead zone approach was chosen instead of filtering because of this reason. It would be interesting to continue improving the ramped dead zone approach, possibly using it with a higher order polynomial function.

The suggested interaction technique is very plausible to use in a FPS game, especially one that targets the competitive console gamers. Because this technique follows the standard in how FPS games are played, the core experience of playing FPS games is still intact. Regarding the efficiency compared to a gamepad it will likely, with some training, have large advantages.
7.3 Changes in the Implementation

A few changes were made to the implementation after the test. First of all, a simple warning that showed up when the PlayStation Move controller was not tracked by the camera was added. The up and down gestures were disabled since they did not work to well. This of course results in a slight modification to the control scheme, which can be seen below.

<table>
<thead>
<tr>
<th>Action</th>
<th>Button/Gesture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aim</td>
<td>L1</td>
</tr>
<tr>
<td>Fire</td>
<td>MOVE</td>
</tr>
<tr>
<td>Grenade</td>
<td>L2</td>
</tr>
<tr>
<td>Melee</td>
<td>Gesture: Forward</td>
</tr>
<tr>
<td>Jump</td>
<td>CROSS</td>
</tr>
<tr>
<td>Reload</td>
<td>SQUARE/Gesture: Rotate CW/CCW</td>
</tr>
<tr>
<td>Switch main weapon</td>
<td>TRIANGLE</td>
</tr>
<tr>
<td>Interact</td>
<td>CIRCLE</td>
</tr>
<tr>
<td>Sprint</td>
<td>L3</td>
</tr>
<tr>
<td>Crouch</td>
<td>CROSS (Hold)</td>
</tr>
<tr>
<td>Scoreboard</td>
<td>SELECT</td>
</tr>
<tr>
<td>Menu</td>
<td>START</td>
</tr>
</tbody>
</table>

7.4 Recommendations

What would still be interesting to see is how well the PlayStation Move performs with a more experienced user. It should be tested with people from the target audience, people who mostly play FPS games on console rather than on the PC.

With the difference in characteristics between a gamepad and a Move controller it will be possible to see different tactics depending on the input device. Therefore, if this implementation is to be used in a game there is big chance that mixing gamepad players and players with the Move controller, in a multiplayer environment might be unfair. There is a few ways of dealing with this problem. First, you could make the PlayStation Move exclusive to single player mode. If it is desirable to have all console gamers play together one could implement the point-and-shoot for multiplayer. It would probably be necessary to have PlayStation Move-only multiplayer games for the suggested interaction technique, because of the possible advantage over gamepads. Implementing both the suggested technique and point-and-shoot would be preferred if gun peripherals is to be used since the suggested technique is not compatible with them.

Which interaction technique is used with the PlayStation Move will ultimately depend on the game. Will the game use the PlayStation Move to attract new players to the game, or is it implemented to please the most dedicated FPS gamers. It is all about the fun versus efficiency contradiction, but if one agrees that a more effective input makes for better gameplay, the suggested interaction technique might just be the one to use.
References


31


9 Glossary

D-pad, short for directional pad, refers to the group of directional buttons found on a gamepad.

FPS is an abbreviation for First-Person Shooter.

Melee attack refers to some sort of close combat attack, i.e. knifing or punching.

Pitch angle is the angle that is changed to be able to look up and down.

Strafing (to strafe) is the action of moving sideways in a game.

Yaw angle is the angle that is changed to be able to look left and right.
10 Appendix

10.1 Questionnaire

1 Background

*Rate your own level of experience (before this test) in the following fields.*

1.1 Mouse and keyboard experience in fps games:

- Very little experience
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- Very much experience

1.2 Gamepad experience in fps games:

- Very little experience
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- Very much experience

1.3 PlayStation Move experience (games in general):

- Very little experience
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- Very much experience

2 Controls

2.1 Which button did you use to fire: **Trigger** or **Move-button** (circle your answer)

2.2 How did you like the control scheme in general?

- Did not like it at all
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- Liked it a lot

Comments: ________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________

2.3 For jumping, crouching, meleeing; did you prefer to use: **buttons** or **gestures**. (circle your answer)

Motivate: ________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________

35
2.4 How often did you use the possibility of lowering the aiming speed by only partially depressing the trigger? (Leave blank if you used the trigger for firing instead)

Never used it --- 1 --- 2 --- 3 --- 4 --- 5 --- 6 --- 7 --- All the time

3 Efficiency

How did it feel to do the following?

3.1 Acquiring targets

Very hard --- 1 --- 2 --- 3 --- 4 --- 5 --- 6 --- 7 --- Very easy

3.2 Aiming at moving targets

Very hard --- 1 --- 2 --- 3 --- 4 --- 5 --- 6 --- 7 --- Very easy

3.3 Aiming while rapidly firing

Very hard --- 1 --- 2 --- 3 --- 4 --- 5 --- 6 --- 7 --- Very easy

4 Experience

Rate the following statements to how well they agree with your own opinions.

4.1 “I would rather play fps with the Move than a gamepad.”

Strongly disagree --- 1 --- 2 --- 3 --- 4 --- 5 --- 6 --- 7 --- Strongly agree

4.2 “I would rather play fps with the Move than a mouse and keyboard.”

Strongly disagree --- 1 --- 2 --- 3 --- 4 --- 5 --- 6 --- 7 --- Strongly agree

4.3 “I felt frustrated playing with the Move controller.”

Strongly disagree --- 1 --- 2 --- 3 --- 4 --- 5 --- 6 --- 7 --- Strongly agree

4.4 “I feel that I have improved my skill with the Move from when I first started.”

Strongly disagree --- 1 --- 2 --- 3 --- 4 --- 5 --- 6 --- 7 --- Strongly agree
5 Physical

5.1 How tiresome was it to use the Move controller?

Not tiresome at all  --- 1 --- 2 --- 3 --- 4 --- 5 --- 6 --- 7 ---  Very tiresome

Comments:____________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________

5.2 How much pain or strain in muscles, tendons etc., did you experience?

No pain at all  --- 1 --- 2 --- 3 --- 4 --- 5 --- 6 --- 7 ---  Very high level of discomfort

Comments:____________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
10.2 Screenshots

10.2.1 Test Level

![Figure 32: Test level – Level start](image)

![Figure 33: Test level – Level end](image)
Figure 34: Test level – Top view

Figure 35: Test level – Top view, textured
10.3 The Gamepads

Figure 36: Sony PlayStation Dual Shock 3 (top view)

Figure 37: Microsoft Xbox 360 controller (top view)

Figure 38: Sony PlayStation Dual Shock 3 (front view)

Figure 39: Microsoft Xbox 360 controller (front view)
## 10.4 Control Schemes for PlayStation Move

**Table 3: Control scheme – Weapon menu**

<table>
<thead>
<tr>
<th>Action</th>
<th>Button/Gesture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aim</td>
<td>L1</td>
</tr>
<tr>
<td>Fire</td>
<td>MOVE</td>
</tr>
<tr>
<td>Grenade</td>
<td>L2</td>
</tr>
<tr>
<td>Melee</td>
<td>Gesture: Forward</td>
</tr>
<tr>
<td>Jump</td>
<td>CROSS/Gesture: Up</td>
</tr>
<tr>
<td>Reload</td>
<td>SQUARE/Gesture: Rotate CW/CCW</td>
</tr>
<tr>
<td>Interact</td>
<td>CIRCLE</td>
</tr>
<tr>
<td>Sprint</td>
<td>L3</td>
</tr>
<tr>
<td>Crouch</td>
<td>CROSS (Hold)/Gesture: Down</td>
</tr>
<tr>
<td>Scoreboard</td>
<td>SELECT</td>
</tr>
<tr>
<td>Menu</td>
<td>START</td>
</tr>
<tr>
<td><strong>Weapon menu</strong></td>
<td>TRIANGLE (+ gestures)</td>
</tr>
</tbody>
</table>