*Bachelor Thesis BTH LOGO*

*Computer Science*

*June 2013*

Using your smartphone as a game controller

to your PC

Marcus Löwegren

Rikard Johansson

School of Computing

BTH - Blekinge Institute of Technology

Address: 371 79 Karlskrona, Sweden

Abstract

Many people in the world today own a smartphone. Smartphones of today usually have an advanced array of inputs in forms of tilting, touching and speaking, and outputs in forms of visual representation on the screen, vibration of the smartphone and speakers for sound. They also usually have different kinds of connectivity in forms of WLAN, Bluetooth and USB. Despite this we are still not seeing a lot of interaction between computers and smartphones, especially within games. We believe that the high presence of smartphones amongst people combined with the advanced inputs and outputs of the smartphone and the connectivity possibilities makes the smartphone a very viable option to be used as a game controller for the PC. We experimented with this developing the underlying architecture for the smartphone to communicate with the PC. We also developed three different games that we let users test to see if the smartphone’s inputs are good enough to make it suitable for such purpose. We also attempted to find out if doing this made the gaming experience better, or in other words increased the enjoyment, of a PC game. The phone was suitable to be used as a game controller for the first two types of game, as our results shows in both terms of enjoyment and in terms of quality of the technical features of the phone such as accuracy and response time. The third game (First Person Shooter) however was better suited to be used with a keyboard and mouse, which our results shows in terms of both reduced enjoyment and complaints about response time of the phone.

Keywords

Game Controller, Natural Mapping, PC Game, Phone-Computer Interaction, Smartphone Game, Human-Computer Interaction, HCI.

Abstrakt

Många människor äger idag en smarttelefon. Dagens smarttelefoner har oftast en mängd avancerade intag av data i form av lutning, pekskärm och mikrofon, och ut data i form av visuell representation på skrämen, vibration av telefonen samt högtalare för ljud. De har oftast också olika sorters anslutningar i form av WLAN, Bluetooth och USB. Trots detta så ser vi fortfarande inte särskillt mycket interaktion mellan datorer och smarttelefoner, speciellt inom spel. Vi tror att den höga förekomsten av smarttelefoner hos människor kombinerat med de avancerade in och ut data av smarttelefonen och anslutningsmöjligheterna gör smarttelefonen en bra kandidat till att användas som spelkontroll för datorer. Vi experimenterade med detta genom att utveckla den underliggande arkitekturen för kommunikationen mellan smarttelefonen och datorn. Vi utvecklade även tre olika spel som vi lät användare testa för att se om smarttelefonens intag av data är tillräckligt bra för att göra den användbar för ett sådant ändamål. Vi försökte även ta reda på om detta gjorde spelupplevelsen bättre, eller med andra ord om det var roligare att spela med hjälp av smarttelefonen. Smarttelefonen var lämplig för att användas som spelkontroll i de två första typen av spel som vi testade med, vilken kan ses i vårt resultat både när det gäller spelgäldje och kvaliten på de tekniska funktionerna av telefonen, som precision och responstid. Det tredje spelet (Första Persons Skjutare) var däremot bättre lämpat att spela med tangentbord och mus, vilket vårt resultat visar i form av minskad spelglädje och klagomål om responstiden med telefonen.

Nyckelord

Spelkontroller, Naturisk Mappning, PC Spel, Telefon-Dator Interaktion, Smarttelefon Spel, Människa-Dator Interaktion.

Marcus Löwegren

Email: lowegrenmarcus@gmail.com

Rikard Johansson

Email: rillee.johansson@gmail.com

Supervisor: Hans Tap

ISSN-Number:

Table of Contents

1 Introduction

1.1 Purpose and Goal

1.2 Research question and hypothesis

1.3 Limitations

2 Background

2.1 Related Works

2.2 Different types of Natural Mappings

2.2.1 Direct Natural Mapping

2.2.2 Kinesic Natural Mapping

2.2.3 Incomplete Tangible Natural Mapping

2.2.4 Realistic Tangible Natural Mapping

2.3 Vibration as an output

3 Game Implementation

3.1 Starchaser

3.2 Labyrinth

3.3 Sniper

3.4 Game Technology

3.4.1 Starchaser Game Technology

3.4.2 Labyrinth Game Technology

3.4.3 Sniper Game Technology

4 Method

4.1 Participants

4.2 Procedures

4.3 Measurements

5 Results

5.1 Technological Results

6 Discussion

6.1 Technological Discussion

6.2 Implementation Discussion

6.3 Starchaser

6.4 Labyrinth

6.5 Sniper

7 Conclusions

8 Future work

9 References

Appendix A

1 Introduction

This paper assumes that the reader have basic knowledge of computers and also basic knowledge about smartphones and its features. From this point and forward we will refer to smartphones as phones.

1.1 Purpose and Goal

Our goal with this project was to find out if it is possible to make a PC-game more entertaining and fun by having the user control a PC-game with a phone instead of the traditional keyboard and mouse, and if the phone is suitable for such a task. We experimented with three different kinds of games in different genres and compared using keyboard and mouse to the phone as game controllers to find out if the phone’s inputs are good enough, and if it is fun to use. We also wanted to try this because we believe there is a future in developing PC games that can connect many phones at the same time to allow multiplayer games with friends. So if friends comes over and they have smartphones, they can just connect them and start playing without having to buy additional controllers and bringing them. This research is a first step towards that where we began exploring the possibilities of this.

1.2 Research question and hypothesis

The questions we will attempt to answer are: Can the gaming experience be improved by having the user play with a phone as a controller instead of keyboard and mouse on a computer? Can a modern phone inputs and features provide good enough accuracy and response time to be used in computer games? Which types of games are they suitable for? We will not be able to test all different kinds of genres so this will only be an indication on what games could be good.

1.3 Limitations

We limited ourselves to single-player games using only one phone with android operating system. Due to time constraints, we also made limitations to develop and test three different kinds of games. We also limited what kinds of people we used when testing our games because of the limited amount of time.

2 Background

This project falls within the areas of Human-Computer interaction, User Interface, Human Interface Device and an area we will call Phone-Computer Interaction. The problems within this area lies around making controllers and input devices with good usability (that feels intuitive, is effective and is easy to use and learn) (Löwgren, 1998), and we investigated this problem specifically for games. There are a lot of different controllers being used as inputs for games, ranging from the standard computer-input, keyboard and mouse, to new and innovative controllers such as the Xbox Kinect (Microsoft, 2013a), Playstation Move (Sony , 2013a), and Wii-controller (Nintendo, 2013a), with the standard console-like controllers to Playstation (Sony , 2013b) and Xbox (Microsoft, 2013b) for example being somewhere in between. We experimented with finding possibilities with using a phone as a controller, since a phone is something many people own today, making it a convenient game controller. Our vision is a future with games where you can go over to a friend’s place, just carrying your phone in your pocket like always, and being able to conveniently connect it to any PC and play multiplayer games with your friends. Phones of today usually have several different inputs that measures elevation and acceleration of the phone as well as a touch-screen and microphone (Joselli and Clua, 2009). We experimented using some of these inputs to find out if they are good enough to make a game-controller of the phone.

2.1 Related Works

There are a couple of previous studies that are related to our work. Some examples are “Implementing Mobile Phone As a Multi-Purpose Controller using 3D Sensor Technology” (Kiran and Patel, 2010), “Using Mobile Phones to Control Desktop Multiplayer Games” (Malfatti et al., 2010) and “Mapping the road to fun: Natural video game controllers, presence, and game enjoyment” (Skalski et al., 2011). These three studies falls into the same area as our study. The first only goes into the technological parts while the technological parts are only a means to us. They do not explore the enjoyment of using a phone as a controller. The second and third one on the other hand are closely related to what we wanted to do. The differences between the second one and our work is that they are using older phones in their experiments that does not have the advanced features that we used. They write that using the accelerometer in phones for games is the next step for future work. In the third one they are comparing the enjoyment of a more natural controller, just like us, but it differs in the way that they used a Wii-controller and a playstation-controller, while we used a phone and a keyboard and mouse.

2.2 Different types of Natural Mappings

Natural mapping in the context of video games is typically thought of as how closely

actions represented in a game match the actions used to bring about that change in a real environment (Tamborini and Bowman, 2010). There are different kinds of Natural Mapping for game controllers, Direct Natural Mapping, Kinesic Natural Mapping, Incomplete Tangible Natural Mapping and Realistic Tangible Natural Mapping.

2.2.1 Direct Natural Mapping

Direct Natural Mapping(Fig. 1) is the most normal way a controller can be mapped. An example from “The Design of Everyday Things” (Norman, 1988) of Direct Natural Mapping is a stove with four burners that are arranged in a two-by-two order. Most stoves would have the controllers for the burners lined up in a straight row while some stoves puts them in a two-by-two layout that correspond to the same two-by-two layout that the actual stove has. By using the two-by-two layout you make it more intuitive for the user to control the four burners separately. A more game related example could be a keyboard . If you implement so when you push the “A” key your character moves to the right, the player would probably be troubled by how the controls worked and it would not be direct mapped. If you make it direct mapped so that when you push the “A” key your character moves to the left the user would faster understand how the controls work. Tamborini and Skalski (Tamborini and Skalski, 2006) argue that more naturally mapped gaming controllers should allow players to quickly access mental models of real-world behavior, if they exist for the player, thereby providing more accurate information about how to interact with the game. If a controller is good mapped the player might even know how to use the controller without even testing it.



*Example of Direct Natural Mapping. Fig 1.*

2.2.2 Kinesic Natural Mapping

Kinesic Natural Mapping(Fig. 2) is when you use body movement that both looks and feels like the movements that you would do in real life to control the environment but you are not using any actual physical controller. An example of this mapping is Microsoft Kinect (Microsoft, 2013a). There are plenty of games that uses this device to capture the motions that the players does in front of it and uses them to control various game elements. By using this technique you let the player use his/her own physical experience and mental modes to control the games.



*Example of Kinesic Natural Mapping. Fig 2.*

2.2.3 Incomplete Tangible Natural Mapping

Incomplete Tangible Natural Mapping(Fig. 3) means letting the user have a controller in the hand that represent the item that the user is going to control. The controller does not need to have the same shape or the weight but it has to be controlled the same way as the item on the screen or in the game environment. In a popular game called Wii Sports (Nintendo, 2013b) there is a golf mini game that uses a Nintendo Wii controller (Nintendo , 2013a) to simulate the control of the golf club , it is not the same shape or weight as a golf club but you are using the same motions as if you would swing a golf club in real life. The wrong shape and weight is what makes it incomplete mapping and not realistic mapping.



*Example of Incomplete Tangible Natural Mapping. Fig 3.*

2.2.4 Realistic Tangible Natural Mapping

Realistic Tangible Natural Mapping(Fig. 4) is the last and most realistic natural mapping that provides the highest amount of natural mapping of the four types. Many arcade games utilize this kind of mapping as you can only play one type of games on them. For example most arcade car racing games use realistic and tangible controllers such as steering wheel, a stick and pedals to control the game, this sort of games often have a special chair to make the driving experience even more realistic. The controls let the player easily access the mental modes that he/she would use when he/she drives a normal car.



*Example of Realistic Tangible Natural Mapping. Fig 4.*

2.3 Vibration as an output

When playing PC games with a mouse and keyboard we usually use two of our five senses to take in information from the games, sight and hearing. Game-controllers for consoles such as the Playstation 3 (Sony, 2013c) often adds another sense, touch, also called the haptic feedback, in form of vibration of the controller. The vibration can either be used to enhance virtual reality by for example emulating the recoil in a weapon when shooting, or by simply giving feedback to the player of something happening in the game. When playing a game the players vision is usually focused on the main task of the game (El-Nasr and Yan, 2006). Shifting the gaze to look at user interface objects such as texts telling you your score is distracting and cognitively demanding. Giving the player haptic feedback is advantageous in this kind of environment and lets the player focus their sight on the main tasks of the game. Humans receive visual and haptic information simultaneously and the nervous system is able to process them in parallel (Goldstein, 2009). Humans also recognizes feedback in the form of touch faster than in the form of sight (Burdea, 2000). Vibration is a very common feature of phones that normally tells the person when there is an incoming call or when he/she receives a text message. Because of this, implementing vibration when using the phone as a game controller is quite intuitive and simple way to tell the player that something important has happened. Even though the vibrators in phones and game controllers normally are too weak to convey actual forces (Badshah et al., 2012), they can still be used to enhance gameplay in the form of giving the player simple feedback. The length of the vibration is important because the player must feel the vibration but the vibration should not distract or trouble the player, the optimal length of the vibration is between 50 ms and 200 ms (Kaaresoja and Linjama, 2005).

3 Game Implementation

We have developed our own games and app that we used in the testing, We felt that doing so would give us complete freedom in changing parts of the program to tailor the system for its purpose. We developed four games but only used three games for the testing. The fourth game was a helicopter game that we ended up removing from the testbed due to that it was going to be too complex to implement the phone as a controller and we felt that it would require too much time for the testers to understand it. The three games that we tested were Starchaser, Labyrinth and Sniper. All three games have a time limit of two minutes.

3.1 Starchaser

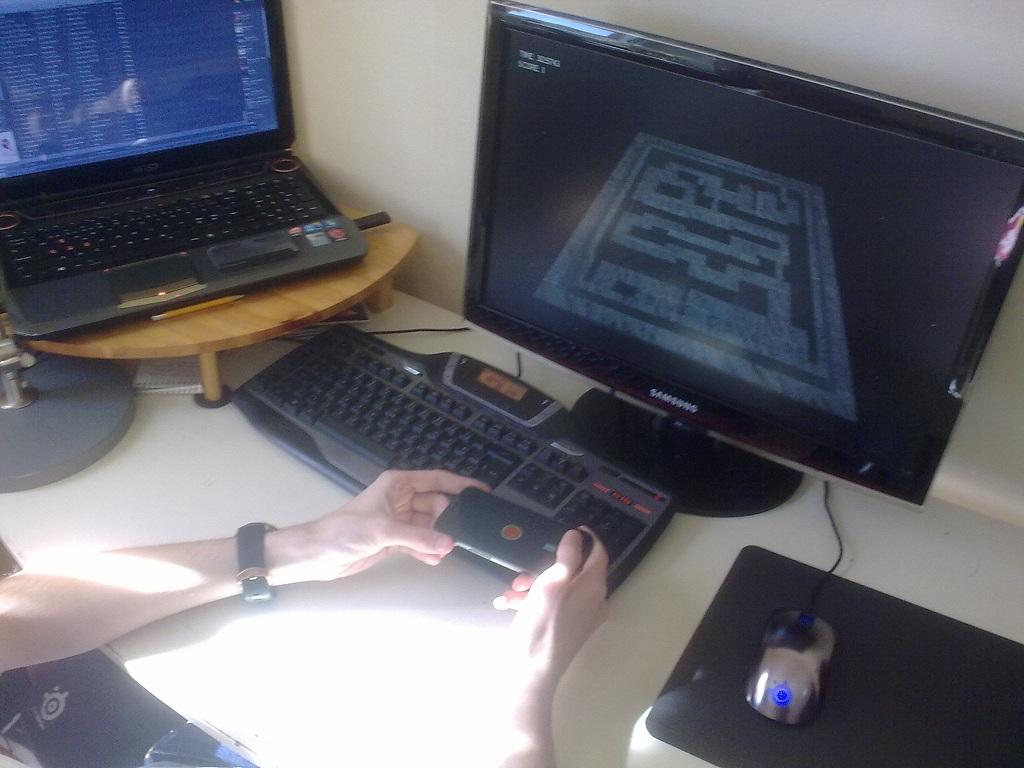
Starchaser (Fig. 5) is a racing game in space. The mission is to collect as many stars as possible before the time runs out. You gain one score for each star that you collect, and when you collect a star the phone gives a short vibration as feedback, as well as a star being shown briefly on the computer screen. What is interesting here is that the natural mappings of the game does not fit the mapping of a keyboard and mouse. When controlling with the keyboard you press “W” or “S” to pitch the spaceship up and down, you can press “A” and “D” to roll the spaceship. To change the speed of the spaceship you can press “+” and “-” on the numpad. To play with the phone you hold it in a steering wheel position and then you can pitch and roll the phone to do the same with the spaceship. If you want to increase or decrease the speed you touch the screen. If you touch at the top of the screen you accelerate and if you touch the bottom you decelerate.



*A person playing Starchaser. Fig 5.*

3.2 Labyrinth

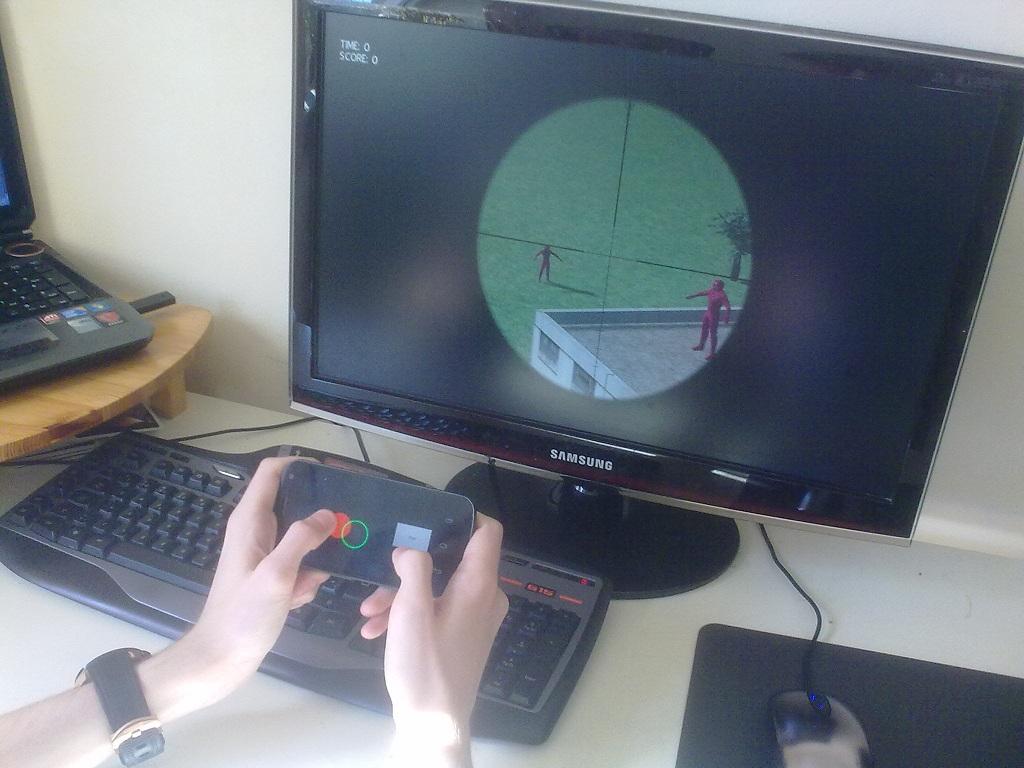
Labyrinth (Fig. 6) is a puzzle game based on an old game first introduced 1946 by BRIO (BRIO, 2012). The goal is to get a ball through the labyrinth without falling down the holes in the gamefield. To move the ball you need to tilt the labyrinth in different directions, making the ball roll. There are a variation of games that let you tilt the labyrinth in different ways. We have fourteen checkpoint spread over the map to be able to calculate score. You gain score by moving the ball to these checkpoints. When you reach a checkpoint the phone gives off a short vibration and a star is shown briefly on the computer screen to make you aware of this. When the game ends you will get the score equivalent to the last checkpoint that you reached (if you finish the game you get a score of fourteen and your time is recorded). We chose this game because it is a game where the phone have a near perfect mapping to real life. When playing with the keyboard you use the WASD keys to tilt and to reset the ball to the second last checkpoint you press the “R” button. If you playing with the phone you hold it horizontally with the display facing up and the tilt it in different direction like you would in real life. If you want to reset the ball you press a “Reset” button on the touch screen.



*A person playing Labyrinth. Fig 6.*

3.3 Sniper

Sniper (Fig. 7) is a first person shooter where you play as a sniper on top of a rooftop. Your mission is to shoot down all the targets. You get one point for each target that you hit. Every time you shoot the phone gives off a short vibration to simulate recoil of the gun, and when you hit a target it gives off a longer vibration and briefly shows a star on the computer screen to make the player aware of this. Usually FPS games are played with the keyboard and mouse, so we chose this game to see if how the phone performs on grounds where the keyboard and mouse usually is the prefered option. In this game when played with a keyboard and mouse, the mouse is used to aim on the targets and the left mouse button is used to shot, and right mouse button to use the scope. To move around on the roof you press “A” (left) and “D” (right) on your keyboard. When you are playing with the phone you use the virtual joystick on the phone which is located on the left side of the touchscreen to aim and you press the “Shoot” button which is located on the right side of the touchscreen to shot. If you want to move to the left or right you tilt the phone in the direction you want to move. To scope in you tilt the phone away from you briefly.



*A person playing Sniper. Fig 7.*

3.4 Game Technology

All our games were developed in C++ using Visual Studio 2012 and a DirectX 11 Graphics Engine. The app was developed for Android (Samsung Nexus Galaxy/X) using Java and Eclipse. The PC program is fully controllable with the app. You can move backwards and forward in the menus and you can also close the program. When you start the app on the phone it scans the network that it is connected by via WLAN with subnet 255.255.255.0 and port 10 000 for a connection and connects to the first ip that runs a server that it finds. When connected to the PC it checks if there is any game started, if a game is already running it starts the corresponding activity on the app, if there is not any game running it shows the select-game menu and it is ready to use. We have also implemented vibration in the phone to be able to give feedback to the player that something noteable has happened. All vibrations are between 50 ms and 200 ms in length, this because for the player to notice the vibration has to be above 50 ms and more than 200 ms is registered as irritating.

3.4.1 Starchaser Game Technology

The Starchaser game has 10 predetermined positions where stars can spawn and it simply loops through them to find a new position for the next star. We decided to implement it like this so that the results could not be affected by randomness of how the stars spawn. We think that because it being in space, and because there is no clear up and down, learning the pattern of the way the stars spawns will require a lot more time with the game than what the testers gets, so learning the map should not be a factor in our tests. The rotation of the spaceship is limited to 1 radian per second around both axises and speed changes are limited to 10% per second. We also have a minimum speed of 10% to prevent people from standing still and turning freely. The phone input uses an algorithm to trim the values to better fit the game. Because the input being a normalized vector we divide both the axises used for rolling and pitching by the third axis. This ensures that when rolling and pitching at the same time you do not have to turn the phone more in each direction. The values of each axises are also scaling exponentially between 0.0 and 0.33, and linearly between 0.33 and 1.0. This makes the sensitivity of the phone inputs lower when it is almost aligned for moving straight forward and it is needed to remove flickering back and forth when trying to hold the ship in a steady course.

3.4.2 Labyrinth Game Technology

The Labyrinth game has 14 predetermined positions that are score-positions. When you are within 2 length units of distance from a score position you get a score worth the value of the place that position is. For example moving straight to score-position number 7 means that your score gets set to 7. Only score-positions higher than your current score are tested. For this game we implemented realistic physics using force and direction of the ball. Every update it checks the entire platform’s triangles for collision against itself and if collision is found it checks the normal of that triangle and adds force to the ball in that direction. Gravity is constantly pulling the ball downwards. We added a limitation to how much the platform could rotate to prevent cheating with quick turning. This limitation is 0.39 radians per second of rotation when using the keyboard, and a maximum limitation of 0.5 radians per second of rotation when using the phone. The reason that the phone has a higher limitation is that the keyboard is digital in its inputs, making it impossible to move the platform slowly. Having a 0.5 radians per second rotation limit on the keyboard resulted in a too high sensitivity, making small movements very hard. The phone’s input values are divided by 4 as well to reduce the sensitivity of the phone, for the same reason as the keyboard, meaning that when rotating the phone 0.8 radians in real life, the platform only gets rotated 0.2 radians in game.

3.4.3 Sniper Game Technology

The Sniper game has 10 predetermined spawn locations for targets. Each target will spawn at its location and 5 of them will start moving back and forth between a waypoint. When a target is hit it will get removed, and it will respawn after 5 seconds at its spawn location. When shooting we have implemented a recoil that lasts for 1.1 seconds and a delay between shots for 1 second to increase realism and increase the difficulty of the game. Previous rotations of the phone gets stored and then compared to the current rotation to determine whether the phone has been rotated enough to trigger the effect of moving left and right and scoping in. This to prevent the phone from having a static default position, and the user can then hold the phone at whatever rotation he/she wants to. The emulated joystick on the phones touch screen uses an algorithm to trim its values. The values gets sent as a two-dimensional vector containing the distance from the center of the joystick that the user is touching the screen in X and Y coordinates. It uses a circular limit of how far you can move the joystick of 150 length units in any direction. The algorithm trims these values by first clamping them to 0 if they are too small to create a deadzone where the user can touch the phone without the aim moving. After that the value is exponentially increased to make the sensitivity non-linear to allow for small fine movements as well as large fast turns.

4 Method

To be able to answer our questions we had to do some tests and collect some data. The data was collected by letting ten people try out the three games that we developed. We let the participants play each of the games in a random order and starting with different input devices each time to prevent learning to interfere with the results. After they had played the games the data was collected by asking them predetermined questions (*Table 1*) about their experience with the games and the input devices. The questions were devised by focusing on getting answers that we could use to answer our research questions.

4.1 Participants

The persons that participated in the tests consisted of friends of ours that wanted to be in the experiment uncompelled. The age range was between 19 and 27 years old. We gathered background information from the participants consisting of the amount of time they usually play video games, what platform they prefer to play video games on, and how important getting a good score in games is to them in terms of having fun. We tried to get participants with a wide range when it comes to gaming experience so that the experience would not be a deciding factor. The participants are ranging from playing games 0-5 hours per week to 15 hours or more per week. We used this background information to draw conclusions about our results.

4.2 Procedures

All tests were done using a Steelseries Sensei mouse, a Logitech G15 keyboard, a Razer Destructor 2 mousepad and a Samsung Nexus Galaxy/X phone. The test persons that came to the experiment room were asked to sit in a chair in front of the screen. The device that the person was going to use first was predetermined and was the same for all the three games, but not every person got to start with the same device. Half of the participants started with the phone as an input device, and half with the keyboard and mouse. They were shown the instructions for the games and the input devices and made sure they read them. We also told them in which general direction they should hold the phone. Some of the participants was allowed play a training round with each controller to see how experience would factor into either input device’s results. The score was recorded during the training round to be able to compare it to their score during the real round to see how much increase in score either control would get. The others were not allowed to try out the controller beforehand but instead immediately started the testing after the participants had read the instructions. They played each game for two minutes each regardless of the result in the game. When they had played with one device they were asked to switch device and play the same game again, and when they had played with both devices they switched to a new game. When they had played all the games we interviewed them and asked them the questions that we had prepared (*Table 1*). By interviewing them we could better assure that they understood all the questions and we could ask follow up questions if we wanted to.

4.3 Measurements

The results from the experiment were measured by looking at our notes from the impressions we got when the participants played and reading the answers given in the interview (*Table 1*).

**Table 1.** *Questions asked in the interview*

Questions

General:

**Gender?**

**How old are you?**

**How many hours do you play video game each week?**

**On what device do you play games the most?**

Questions for each game (StarChaser, Labyrinth and Sniper):

**It was more fun to play the game with the mobile phone than the keyboard and mouse.**

**It was harder to play with the mobile phone than the keyboard and mouse.**

**The input-delay of the phone was good enough to use for this game.**

**The precision/accuracy on the phone was good enough for this game.**

Overall:

**Did you notice the vibration feedback?**

**The vibration in the mobile phone increased the gaming experience.**

**What device was the most fun to play on?**

**What game was the most fun to play?**

**It is important to get a good score for a game to be fun.**

5 Results

The result that we got from the questionnaires that the participants answered after performing our tests were processed and formatted so that we could easier extract information from them. We condensed our results to tables showing average, median, max and min values where applicable. All our result-values are between 1 and 7, with 1 being strongly disagreeing and 7 being strongly agreeing, and 4 being neutral. We chose a 7-point scale because we wanted an uneven number so there would be a neutral number, and with enough precision for different amounts of agreement / disagreement but still with a low enough number to not confuse the testers.

**It was more fun to play the game with the mobile phone than the keyboard and mouse.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | StarChaser | Labyrinth | Sniper |
| Average | 5 | 5,5 | 2,7 |
| Median | 6 | 6 | 2,5 |
| Max | 7 | 7 | 6 |
| Min | 2 | 1 | 1 |
| Answered 5 or higher | 60% | 80% | 20% |

**It was harder to play the game with the mobile phone than the keyboard and mouse.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | StarChaser | Labyrinth | Sniper |
| Average | 5,2 | 3,4 | 6,3 |
| Median | 5,5 | 3 | 7 |
| Max | 7 | 6 | 7 |
| Min | 1 | 1 | 4 |

**The input-delay of the phone was good enough to use for this game.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | StarChaser | Labyrinth | Sniper |
| Average | 6,5 | 5,4 | 4,4 |
| Median | 6,5 | 5 | 4 |
| Max | 7 | 7 | 7 |
| Min | 6 | 2 | 2 |

**The precision/accuracy on the phone was good enough for this game.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | StarChaser | Labyrinth | Sniper |
| Average | 5,9 | 5,9 | 5,6 |
| Median | 6 | 6,5 | 5,5 |
| Max | 7 | 7 | 7 |
| Min | 4 | 4 | 3 |

**Did you notice the vibration feedback?**

7 out of our 10 participants noticed the feedback.

**The vibration in the mobile phone increased the gaming experience.**

|  |  |
| --- | --- |
|  | Overall |
| Average | 5,6 |
| Median | 5,5 |
| Max | 7 |
| Min | 4 |

**What device was the most fun to play on?**

8 out of our 10 participants answered that they thought the phone was the most fun device to play on.

**What game was the most fun to play?**

StarChaser: 4

Labyrinth: 4

Sniper: 2

**Participant score with keyboard and mouse.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | StarChaser | Labyrinth | Sniper |
| Average | 10.2 | 3.2 | 30.7 |
| Median | 10.5 | 2.5 | 32.5 |
| Max | 15 | 6 | 41 |
| Min | 6 | 1 | 15 |

**Participant score with phone.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | StarChaser | Labyrinth | Sniper |
| Average | 5.4 | 3.2 | 14 |
| Median | 5 | 2 | 14 |
| Max | 11 | 6 | 18 |
| Min | 3 | 1 | 9 |

**Participant score when allowed to play a test round**

Playing a test-round first increased the participants score by 24% on average when using the keyboard and mouse and by 34% on average when using the phone.

5.1 Technological Results

We achieved an average ping of 60 ms back and forth between the phone and the computer, with spikes upwards 200 ms happening rarely. The maximum frequency of network packets sent by the phone that we managed to send without getting input delay was 100 packets per second. When we exceeded this number the computer was unable to process and handle the messages in time.

6 Discussion

We believe that the method we used when we performed our tests was good and accurate. We would however had liked to have more people perform our tests in order to get a more weight behind our results. However the amount of people we had in our tests is enough for us to draw some conclusions. Overall the participants had a worse score when playing the games with the phone. This is to be expected because of how little experience they have of using a phone as a game controller. We believe that with enough practice the phone could be a better game controller in terms of score. An indication towards this can be seen when we let some participant play a test round with either controller first. On average the participants score when using the phone increased by 34% when they were allowed to play a test round first, compared to 24% with the keyboard and mouse. This could just very well be the results of the learning curve being harder on the keyboard due to the previous experience with it. Preferably we would have liked to have some of our participants test our games over the span of a couple of months to be able to deeper analyze the results of long-term use, but we were unable to do this.

Our expectations before the tests was that Starchaser was going to be the most fun game with the phone, followed by Labyrinth and on the last place Sniper. This was based on how fun and hard the games were in our opinion, and how natural it was to implement the phone controls for the games. Our expectations were almost correct because Starchaser and Labyrinth was chosen as the most fun game by four participants each and Sniper was chosen by two. However 80% agreed that Labyrinth was more fun to play on the phone, making it a suitable game to play with the phone. We think that the phone is more fun because it is a new and exciting way to interact with computer games. The phones mapping is also almost perfect to the Starchaser and Labyrinth game and this is probably another reason why it is so fun to play these games with the phone, and why Sniper is not.

We were unable to draw any conclusions about the age of the participant affecting the opinions and / or score when playing with the phone. When it comes to previous gaming experience it only showed in the Sniper game when played with mouse and keyboard. There was a strong correlation there between number of hours they spent playing video games per week and the score they got.

6.1 Technological Discussion

We believe that the features of the phone are good enough to be used for games. We were surprised with how many network packages with updates that the phone managed to send to the computer per second, more than the computer could handle when taking care of all the game logic as well. So we had to lower the amount of packages sent by the phone to 100 per second to reduce the input delay. With this update rate the computer could handle it and it was enough to keep the response time low. This frequency can be compared to for example a normal computer screens refresh-rate at 60 updates per second. When asked if the delay between computer and phone was good enough the average answer was above 4.0 on all the games. around 60 ms ping back and forth between the phone and computer results in around 30 ms delay between performing an action on the phone and the game getting the update of that action, which we think is well within acceptable values. The accelerometer had a precision of six decimals with values between 0 - 10, and the joystick gives a precision of four decimals and values between 0 - 150 (the joystick values can be changed to match a specific game easily). Both the accelerometer and the joystick had more than enough accuracy to use for the games that we developed. When our participants were asked about the accuracy in the games the average answer was above 5.0 on all games. The vibration of the phone was met with positive reactions. Seven out of our ten participants noticed that the phone vibrated during the tests. The average score when stated that the vibration made the gaming experience better was 5.6. This makes us draw the conclusion that vibration as a form of feedback is a good thing in games. One participant even said that he wanted more vibration integrated in the games because he liked it so much. This might not be a good idea though, since if the phone is vibrating too often a person could stop reacting to the feedback and ignore it or get confused on what feedback he/she is getting.

6.2 Implementation Discussion

When we did our experiments we noticed that the battery of the phone depleted fairly quickly. This could be a disadvantage to our implementation technique if you are for example going over to a friends house and you have not recharged your phone that day. The app is not developed with power usage kept in mind because it was not a problem for us since we could charge the phone between sessions. But we think that it is possible to optimize the power useage. The app that we developed is locking the screen so the display can not dim itself to consume less power. By letting the display dim itself when the screen is not in use by the player the battery will last longer. Another solution would be to optimize the communication between the app and the computer to only send data when noticing a significant change in the values. This would reduce the amount of packages that would be sent to the computer and therefore reduce power usage. One more easy optimization that we thought about doing is to lower the update rate of the accelerometer on the phone. We set this updaterate to 60 updates per second, and it can probably be lowered to 30 updates per second without noticeable increase of input delay. We decided not to implement this optimization though because we did not have time to test if and how it affected the games and the gameplay. The problem we had with the computer only being able to process around 100 network packets per second could most likely be optimized too by better multithreading the program. Having a dedicated thread taking care of network packets and filtering packets that contains non-important data and batching updates that contain similar data and then sending updates to another thread that takes care of the actual gameplay would probably be a better approach if a higher update rate of packets is desired. With several phones connected to a single computer for multiplayer this would most likely be needed.

6.3 Starchaser

Starchaser was harder to play on the phone, as can be seen in both the score and the average rating when the participants was asked about this, which was 5.2. Even if the game was harder 60% thought that it was more fun to play with the phone. We believe this is because the mapping between the phone and the computer is close to perfect to how it would be in real life if you were using, for example, a plane joystick (Fig 8). Starchaser was the best game when considering the performance of the features of the phone. When asked if the input-delay was good enough to use for this game the average score was 6.5 and the accuracy of the phone got an average score of 5.9. So the phone’s inputs are definitely good enough to use the phone as a controller when developing a game like Starchaser. We believe that the reason that the input delay got a very good rating is that when flying a spaceship one does not think that turning it will be instantaneous. When moving a big vehicle turning usually takes time, both in real life and in other games. In this case the input delay that is there might very well be conceived as added realism in terms of people expecting the spaceship to take time turning around. The speed change is also not updated directly, but the spaceship instead accelerates and decelerates over time, making the input delay unnoticeable in that regard.



*Pilot steering wheel controller. Fig 8.*

6.4 Labyrinth

Labyrinth was the most surprising of the three games. We did not anticipate the positive feedback that we got from this game. It seemed that this was the favorite game of the testers when played on the phone and a few even told us that we should continue developing games like this one and try to release them. The average score when asked if it was more fun to play this game with the phone was 5.5, and it was the highest score of the three games, also 80% of our participants answered that question with a 5 or above. This game was the only game where the participants felt that it was easier to play with a phone compared to a keyboard and mouse, with an average score of 3.4 when asked if the phone was harder. This game also showed that some people scored better using the phone than the keyboard and mouse, something we did not see any any of the other games. This is probably because the phone has a perfect mapping between the game and the phone and this is also the reason that makes it more fun. A mistake we made was making this game too hard. The highest score any participant got was 6 out of the possible 14, while a few only managed to get 1 score. We think if we made the maze easier the game would have been more enjoyable for those that performed badly.

6.5 Sniper

Sniper was not surprisingly the hardest and least fun game when played with the phone. This is to be expected because of how well established the mouse and keyboards are as controllers for First Person Shooter games. The average score when asked if the phone was more fun to play with in this game was 2.7, so almost everyone thought that it was more fun to play with the keyboard and mouse. It was also harder to play with the phone, with an average score of 6.3 when asked about it, making it the hardest of the 3 games. This is probably one of the reasons that reduces the enjoyment of the game. The emulated joystick on the left side of the touchscreen makes it a lot harder to play with the phone, it is because you can not feel how far you move because you do not have any physical resistance or representation of it. The accuracy does not seem to have any importance on how bad the phone performed in terms of enjoyment. When asked about that the accuracy was good enough for this game the average answer was 5.6, which is good. The input delay on the other hand did most likely have some importance to the low enjoyment when played with the phone. The average participant answered with a 4.4 but there was some scores that was below the middle.

7 Conclusion

The gaming experience can be improved by having the user play with a phone instead of a keyboard and mouse on a computer. Our results showed that 80% of our participants preferred playing with the phone over the keyboard, and even in the game where the phone scored worst, 20% of our participants still thought that it was more enjoyable playing with the phone. The inputs and features of a modern phone does provide good enough accuracy and response time to be used in computer games. In all our games that we tested with the phone the participants thought that the phone was adequate, with the StarChaser game and the Labyrinth game getting high ratings and the Sniper game getting just above acceptable values in these regards. The three genres that we used in the testing were a flight / space-racing game, a puzzle game and an First Person Shooter game. The phone was very suitable to be used as a game controller for the first two types of game, as our results shows in both terms of enjoyment and in terms of quality of the technical features of the phone such as accuracy and response time. The third game however was better suited to be used with a keyboard and mouse, which our results shows in terms of both enjoyment and complaints about response time of the phone.

8 Future Work

Phone-Computer interaction is an area with much left to be done, especially within games. Based on our results we definitely think that the phone very well can be a viable option as a game controller for the PC. We see the potential of future work to be done where the possibilities of multiplayer is examined. Since phones are so accessible we can see the potential of multiplayer games using several phones as game controllers to a single computer being very popular. Since the technique we used for the communication between the phone and computer in theory supports a very high amount (IPV4/6 would be the current limitation) simultaneous phones being connected to the computer, we could see the potential of people coming together to play computer games with each other in real life, without having to worry about buying additional controllers, or even bringing their controllers, since they would always have their phone with them anyway. Implementing classic board games such as trivial pursuit (Bellis, 2013) could also work very well with the phone as each player's controller. On a bigger scale using this technique at big events such as concerts and sport matches could also be a big potential. Every spectator could connect their phone to a WLAN at the event and then use them to interact with the event and other spectators. Another potential area for future work could be integrating the phones speakers to be used as an output of game-sounds. The speakers in todays phones are getting better and better and could potentially replace external speakers or headset that people normally use today for playing sounds in games.

9 References

Badshah, Akash; Gupta, Sidhant; Morris, Daniel; N Patel, Shwetak; Tan, Desney; (2012) GyroTab: A Handheld Device that Provides Reactive Torque Feedback, in Proceedings of the *SIGCHI Conference on Human Factors in Computing Systems*, ACM New York, USA 2012, pp. 3153-3156.

Bellis, Mary, ([2013](http://www.nintendo.com/games/detail/1OTtO06SP7M52gi5m8pD6CnahbW8CzxE)) The History of Trivial Pursuit http://inventors.about.com/library/inventors/bl\_trivia\_pursuit.htm[, April 2013, [Online], Available:](http://www.nintendo.com/games/detail/1OTtO06SP7M52gi5m8pD6CnahbW8CzxE) http://inventors.about.com/library/inventors/bl\_trivia\_pursuit.htm

BRIO, (2013) Labyrinth Game, http://www.brio.net/ToPlay/3\_years/Games/34000\_Labyrinth.aspx, last visited, April [2013](http://www.nintendo.com/games/detail/1OTtO06SP7M52gi5m8pD6CnahbW8CzxE), [Online], Available: http://www.brio.net/ToPlay/3\_years/Games/34000\_Labyrinth.aspx

Burdea, Grigore, (2000) Haptics issues in virtual environments, (eds) Rawlinson, Anne, Proceedings of *Computer Graphics International*. Proceedings, pp. 295-302.

El-Nasr, Magy Seif; Yan, Su, (2006) Visual Attention in 3D Video Games in Proceedings of *Advances in computer entertainment technology*, ACE '06 New York, USA 2006, Article No.: 22.

Goldstein, Bruce, (2009) *Sensation & Perception, 8th ed.* Wadsworth, Belmont,

Joselli, Mark; Clua, Esteban; (2009) grmobile: A framework for touch and accelerometer gesture recognition for mobile games, in Proceedings of the *VIII Brazilian Symposium on Games and Digital Entertainment*, ser. SBGAMES ’09, Washington, DC, USA: IEEE Computer Society, 2009, pp.141–150, last visited, May 2013, [Online]. Available: http://dx.doi.org/10.1109/SBGAMES.2009.24

Kaaresoja, Topi; Linjama, Jukka, (2005) Perception of Short Tactile Pulses Generated by a Vibration Motor in a Mobile Phone. In: *WHC 2005. First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 471–472.

Kiran, Bhagiya; Patel, Sankita, (2010) Implementing mobile phone as a multi-purpose controller using 3D sensor technology, In *2010 3rd IEEE International Conference on Computer Science and Information Technology*, ISBN 9781424455379, Volume 2, pp. 584 - 588.

Löwgren, Jonas; Stolterman, Erik, (1998) *Design av informationsteknik – materialet utan egenskaper*, Studentlitteratur, Lund.

Malfatti, Silvano Maneck; Ferreira dos Santos, Fernando; Rodrigues dos Santos, Selan, (2010) Using Mobile Phones to Control Desktop Multiplayer Games In *Brazilian Symposium on Games and Digital Entertainment*, ISSN 2159-6654, ISBN 9781612843919, pp. 230 - 238.

Microsoft, (2013a) Introducing kinect for Xbox 360, http://www.xbox.com/en-US/kinect, last visited, April 2013. [Online]. Available: <http://www> .xbox.com/en-US/kinect

Microsoft, (2013b) Xbox 360 Controller, http://www.xbox.com/en-US/xbox360/accessories/controllers/Xbox360Controller, last visited, April 2013. [Online]. Available: http://www.xbox.com/en-US/xbox360/accessories/controllers/Xbox360Controller

Nintendo, (2013a) Controls for Wii, http://www.nintendo.com/wii/what-is-wii/#/controls, last visited, April 2013. [Online]. Available: http://www.nintendo.com/wii/what-is-wii/#/controls

Nintendo, (2013b) Wii Sports http://www.nintendo.com/games/detail/1OTtO06SP7M52gi5m8pD6CnahbW8CzxE, April 2013 [Online], Available: http://www.nintendo.com/games/detail/1OTtO06SP7M52gi5m8pD6CnahbW8CzxE

Norman, Donald A; (1988) *The Design of Everyday Things*, Basic Books, New York.

Skalski, Paul; Tamborini, Ron; Shelton, Ashleigh; Buncher, Michael; Lindmark, Pete; (2011) Mapping the road to fun: Natural video game controllers, presence, and game enjoyment, In *New Media Society 13:224*, originally published online before print 12 October 2010, DOI: 10.1177/1461444810370949, The online version of this article can be found at: http://nms.sagepub.com/content/13/2/224

Sony, (2013a) This Is How I Move™, http://us.playstation.com/ps3/playstation-move/, last visited, April 2013. [Online]. Available: http://us.playstation.com/ps3/playstation-move/

Sony, (2013b) DUALSHOCK®3 Wireless Controller, http://us.playstation.com/ps3/accessories/dualshock-3-wireless-controller-ps3.html, last visited, April 2013. [Online]. Available: http://us.playstation.com/ps3/accessories/dualshock-3-wireless-controller-ps3.html

Sony, (2013c) Playstation 3, http://se.playstation.com/ps3/, Last visited, May 2013. [Online]. Available: http://se.playstation.com/ps3/

Tamborini Ron, Skalski Paul; (2006) The role of presence in the experience of electronic games. In: Vorderer, Peter; Bryant, Jennings; (eds), *Playing Video Games: Motives, Responses, and Consequences*, Mahwah, NJ: Lawrence Erlbaum, pp. 225–240.

Tamborini, Ron; Bowman, Nicholas David; (2010) Presence in video games. In: Bracken, Cheryl Campanella; Skalski, Paul; (eds) *Immersed in Media: Telepresence in Everyday Life*, Routledge, New York, pp. 87–109.

Appendix A

These are the instructions that were given to the player before they started the test.

General:

You are going to test three different games with both a keyboard and mouse and a phone to control the game. Every game has a time limit of two minutes and your goal is to get as high score as possible.

Starchaser:

Starchaser is a game where you are going to fly a spaceship that is going to fly around and pick up stars in space. There is an arrow in the top of the screen that shows where the next star that you are going to collect is located.

Keyboard:

You steer the ship using the WASD keys and you can press and hold “+” and “-” on the numpad to increase or decrease the speed.

Phone:

Hold the phone like a steering wheel with the back of the phone away from you. You steer by tilting the phone in different directions. You can use the touchscreen to increase or decrease the speed. Your speed is set dependent on where you pressed on the screen, if you touch in the middle your speed is going to be set to 50%.

Labyrinth:

Labyrinth is a lot like the normal labyrinth game where you tilt the game with wheels on the side of the game. Your goal is to take the ball from the top right corner to the bottom left corner. You get score while moving toward the goal.

Keyboard:

You steer with the WASD keys on the keyboard, if you fall down you can press R to come up again.

Phone:

Hold the back of the phone downwards and the platform will follow the movements of the phone. If you fall down you can press a reset button on the screen to get up again.

Sniper

Sniper is a First person shooting game where your goal is to shoot down all the targets. You stand on a rooftop with a sniper and you move sideways along the edge of the roof.

Keyboard and mouse:

You aim by moving the mouse. You can move left and right by pressing the A and D key. You can use the scope by pressing the right mouse button and you shoot by pressing the left mouse button.

Phone:

You aim with the joystick on the left side of the touchscreen. You move left and right by tilting the phone that way. You scope in by tilting the phone forward a brief moment and you shoot by pressing the Shoot button on the right side of the screen.