NTNU
Norwegian University of
Science and Technology

Gaze Interaction for Handheld Multi-touch Devices

An Explorative Study

Lars Emil Fjermeros
Shimin Sun

Master of Science in Computer Science
Submission date: June 2016
Supervisor: Dag Svanæs, IDI
Co-supervisor: Terje Røsand, IDI

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

Preface

This study is a master thesis conducted in the last semester of the master's degree program in Computer Science at Norwegian University of Technology and Science (NTNU) for the Department of Computer and Information Science (IDI). The specialization of the thesis is software.

We want to give a big thank you to our supervisor, Professor Dag Svanæs at NTNU. All guidance, feedback and ideas during the whole semester has been much appreciated. Also a thank you to Senior Engineer Terje Røsand who has been responsible for setting up the hardware and given technical support.
We would like to thank all participants who attended the usability tests, for spending their valuable time and give input, helping us to achieve our goal.

Lars Emil Fjermeros and Shi Min Sun
Trondheim, June 2016

Sammendrag

Denne masteroppgaven er skrevet som et siste ledd i studieprogrammet Datateknologi på NTNU. I oppgaven blir det utforsket konsepter rundt blikkinteraksjon, og hvordan dette kan være med å erstatte berøringsinteraksjon på håndholdte multi-touch enheter. Som resultat har det blitt utviklet 3 moduler som benytter blikket som interaksjonsmetode. Disse modulene har gjennomgått en brukertesting som består av to iterasjoner. Modulene vil bli sammenlignet med lignende applikasjoner som bruker interaksjoner for berøring, mus og tastatur.

Forskningsprosessen begynte med å analysere tidligere studier rundt interaksjonsteknikker for blikk og berøring. Ved å se nærmere på eksisterende guider for bruk av berøringsteknikker, blir en samling av grunnleggende bevegelser for berøring identifisert. Styrker og svakheter ved blikkinteraksjon blir undersøkt fra tidligere litteratur. Disse funnene danner ett forskningsgrunnlag som kan brukes til å opprette et utvalg av basis funksjonaliteter for interaksjon med blikket. 3 prototyper har blitt utviklet for å teste ut blikkinteraksjon, og disse vil være selve grunnlaget for en større brukbarhetstest.

I et forsøk på å dele inn testen i to iterasjoner, vil den første bestå av 9 tester som igjen er delt inn i 3 grupper for hver interaksjonsmetode – blikk, berøring, must og tastatur. Hver av disse gruppene vil bestå av en egenutviklet prototype, med to andre applikasjoner som enten benytter berøring eller tastatur og mus som interaksjon. Totalt 11 deltakere deltok i den første iterasjonen av brukbarhetstesten.

I den første iterasjonen har effektivitet blitt brukt som et mål og kriterie for brukervennlighet, hvor tidsbruk har vært en essensiell faktor. Det var forskjellige resultater basert på prototypene, og de vises å være forårsaket av begrensninger med erfaring, personlige faktorer, hardware og software. Effektiviteten har vist seg å være lik for de forskjellige prototypene, hvor resultatene indikerer i noen få tilfeller at blikkinteraksjon er den beste interaksjonsmetoden. Blikkinteraksjon har vist seg å være intuitiv og lett å lære, mens tilfredsheten ble målt til å være lav i alle tilfellene.

I den andre iterasjonen ble en fjerde modul inkludert, denne kombinerer de 3 forrige prototypene. Tilbakemeldinger fra den første iterasjonen ble benyttet for å forbedre opplevelsen. En gruppe på 4 deltakere ble satt opp som testpersonen, hvor modulen ble sammenlignet med resultatene fra den forrige testen, og resultatene viser seg for det meste å være positive.

Blikkinteraksjon har ett stort potensiale ut ifra brukervennlighetstesten, men på grunn av ulike faktorer er det vanskelig å sammenligne det med de andre interaksjonene på dette stadiet. Problemene som dukket opp kan derimot bli fikset med ny og forbedret hardware og dedikert utvikling av software.

Abstract

This thesis explores how promising gaze interaction can replace touch interaction on handheld multi-touch devices. A set of three modules have been developed that use gaze as interaction method. These modules are run through a usability test consisting of two iterations. The modules will be compared against similar applications that use interactions for touch, keyboard and mouse.

The research process begins by analyzing previous research on interaction techniques with gaze and touch interaction. By studying the *iOS Human Interaction Guidelines*, a collection of basic gestures for touch are identified. Strengths and weaknesses of gaze interactions are examined from previous literature. These findings create the research foundation, and has formed a selection of functionalities for gaze interaction. In the attempt to cover these gaze gestures, three prototypes has been developed that will be the basis for the usability test.

A first iteration of the usability test includes 9 test cases divided into 3 groups for each interaction method - gaze, touch, keyboard and mouse. Each group will consist of a self developed prototype, with two other applications using interaction with touch or keyboard and mouse. In total 11 individuals participated in this part of the usability test.

Efficiency has been used as a measurement for usability for the first iteration, where time commitment has been an essential factor. There were different results based on the prototypes, and they appear to be caused by limitations in user experience, personal factors, hardware and software. Effectiveness has shown to be similar between the prototypes, where the results indicate from a few cases that gaze interaction is the superior interaction method. Gaze interaction has shown to be intuitive and easy to learn, although user satisfaction has been low in all test cases.

With the first iteration completed, a second iteration includes a fourth module which combines the 3 previous prototypes. Feedback from the first iteration has been used to add improvements. A group of 4 participants tested this last module, where it was also compared with the previous developed prototypes. Mostly positive responses were received.

To conclude, gaze interaction has a great potential, but due to many factors it was difficult to justify it as superior over existing interactions from the usability test. It is unlikely for gaze interaction to replace interaction with touch, keyboard and mouse at the current state. However, issues can and will be fixed with new and excelling hardware running on dedicated software.

Table of Contents

List of Tables

List of Figures

Abbreviations

IT	-	Information Technology
CIF	-	Common Industrial Format
SQuaRE	-	Software product Quality Requirement and Evaluation
VR	-	Virtual Reality
IEC	-	International Electrotechnical Commission
ISO	-	International Organization for Standardization
NTNU	-	Norwegian University of Technology and Science
IME	-	Information Technology, Mathematics and Electrical Engineering
IDI	-	Department of Computer and Information Science
GTA	-	Grand Theft Auto
MDITIM	-	Minimum Device Independent Text Input Method

Chapter 1

Introduction

1.1 Motivation

Javal (1879) [17] was the first person who made a breakthrough in the field of eye tracking. His research focused on eye movement, specifically how people read books. He found that the eyes do not perform a smooth sweeping gaze movement along the text, which was assumed to be true at this time. Instead, Javal found that the eye will gaze at a word for a brief moment, this is known as *fixation*. Afterwards the gaze will jump a short distance to the fixation point. This phenomenon is called *saccade*, and the discovery raised many questions, like what words does the gaze fixate on? For how long? This marked the beginning for eye tracking as scientific study. In the early 20th century most of the research was focused on reading, such as a book named *The Psychology and Pedagogy of Reading* by Huey [14], but in later years eye tracking shifted focus to other field of studies, such as physiology by Keith Rayner [25].

Eye tracking involvement in interface design for digital computers were first used in the 1980s. The idea behind using eye tracking technology in interaction design is to record and display how the gaze moves when interacting with a computer. It is desirable to place important interactions where the user often gaze in order to avoid searching for them on the screen.

Gaze interaction research can be tracked back to the late 1980s with Ware and Mikaelian [36] being one of the first records found. Although most of researches proves gaze interaction have a great potential, the tremendous costs of eye tracking devices and lack of industrial support has made it a slow process to make it beneficial for everyday use.

The rapid technological development over the last 30 years has brought exciting new prospects for eye tracking interfaces. With the development of new and excelling hardware, eye tracking might be able to revolutionize human computer interactions once again, for the second time. Instead of improving traditional interaction methods, such as keyboard, mouse and touch interaction, gaze interaction itself could be the new modern approach for interaction with computers. Methods of interaction can be drastically changed in the future, and navigation using mouse and keyboard may not be as usual as it is today. In an ideal world, interaction using gaze will undoubtedly increase the efficiency by a noticeable amount. Lately multiple companies have started to produce eye tracking devices that are affordable, mobile, and easy to use. Also, there are already dedicated software that have built-in support for gaze interaction, notably a game called *The Division* by Ubisoft [34].

Even with the current development of gaze interactions, there are still limitations for assisting interactions with touch, mouse and keyboard. In this project, we want to take a closer look at operating multi-touch handheld devices using gaze interactions. If this is possible, it could help complete major tasks both faster and simpler.

The ideal target group for gaze interaction is most people who use any kind of computer devices, which in today's world could basically be everyone. The first group to adapt a device that only has gaze interaction, might be user group in workplaces where hands are not available. Imagine a car mechanic who has his hand soaked with oil, instead of using his dirty hands and break the device, he could use his eyes to control the device instead. Another early adaption might be those who are handicapped and can not use their hands due to muscle or other problems. An example of this is Professor Stephen Hawking, who has his own personal modified interaction method that is made specifically for him. Of course, not every handicapped person can afford such expensive setup, therefore gaze interaction might just be the prominent solution.

1.2 Research Goals

The goal for this thesis is to analyse how promising gaze interaction can replace touch interaction in multi-touch handheld devices, such as a smart phones or tablets. In order to achieve good results a set of research questions has been formed, where each of them will correspond to a specific goal that we want to achieve in certain sections of this report. By gathering and combining the answers from these research questions, a conclusion should be possible to highlight a result, and reveal if gaze interaction potentially could be good enough, and ready to replace general touch interaction methods.

1.2.1 Research Questions

1. ***Based on available literature, what functionalities does gaze interaction need in order to replace current interactions techniques on multi-touch handheld devices?*** The goal is to replace the current interaction method on multi-touch handheld devices with gaze instead, it is important to examine the basic functionalities gaze interactions need to fulfill. To achieve this we will analyze the functionalities touch interaction have, and try to replace each one of them with gaze instead. The result of this phase is to create a basic functionality which we will use to operate applications using gaze interaction.

2. ***Using basic functionality sets, what are the most promising use situations that will cover these functionalities?***
   Due to our constraint of time and the vast number of existing software categories, a decisive factor is to find software that would be greatly improved by including gaze interaction functionality. Since it is rather safe to assume our own prototype implementations will not be perfect, choosing a prominent type of software that can take advantage of gaze interactions will provide a better overall result. Therefore, the second goal for the literature study will be to find promising candidates that can be implemented with gaze functionality. We want to construct our prototypes in time, to be good enough for testing with other users.

3. ***What are the major technical challenges implementing these gaze interactions with current eye tracking technology?***
   An implementation of the software will most likely provide several challenges, simply because the current technology is not made with gaze interaction in mind. Today's devices are made with one interaction in mind from scratch, making it hard to implement new interaction methods on top. If these challenges are too difficult to overcome and require a specific platform, it would be much more difficult for the industry to adapt. On the other hand, if it is not problematic, it would be much less of a gamble for the industry to introduce such technology in the future. We hope to find a suitable answer for this question during our implementation, and if possible come up with a solution and best approach.

4. ***How do users assess the usability of gaze interactions for the use situations mentioned in Research Question 1, compared to similar interactions using touch, mouse and keyboard?***
   After finishing the implementation, all the modules will be run through a usability test. See Chapter 4 for the structure of the tests given. By testing the modules that uses gaze as interaction method and comparing it to touch, it is possible to find how competitive gaze interaction is compared to the others in terms of usability. In addition to touch interaction we also want to use keyboard and mouse as a control group.

5. ***What role might gaze interaction play in the coming development of computers and multi-touch handheld devices?***
   The last research question takes into account how gaze interaction is currently in the industry, and combines it with our result. We want to predict how gaze interaction will be used in the coming years.

1.3 Research Methods

This section will give an overall description of the research methods used during this project. We will use the process suggested by Briony J Oates [22], the model is displayed in Figure 1.1, and the red areas are research methods that will be used in this thesis.

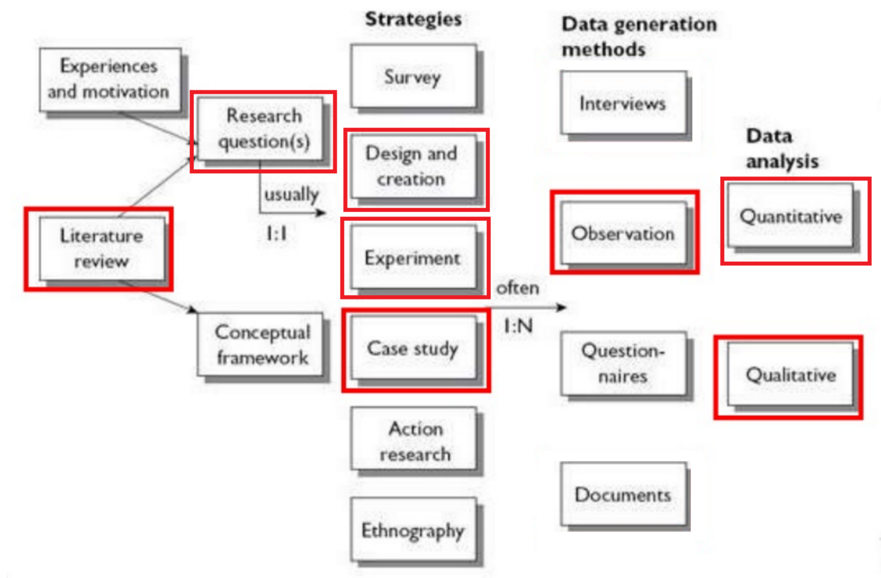


Figure 1.1: The research process used in this thesis using the model by Briony J Oates

1.3.1 Literature Review

A literature review often comes in two separated parts as described by Briony J Oates [22]. First part is where the researcher explores available literature to find a suitable research idea, and read about other researches that already have been made associated to a given topic. An outcome for the first part should combine the research ideas and give a deeper understanding around the research topics to form a set of research problems.
Second part is to gather and present evidence that assures the research as something new. Similar to a legal case where a lawyer will refer to previous cases, and why the current case is identical or different from the previous one.

1.3.2 Design and Creation

The design and creation part is a practice focused on to creating something. From an information technology (IT) perspective, it would often be a new product called an *artifact*. An artifact can come in four different types. First one being *construct*, which is about coming up with a new concept or vocabulary used in an IT-related domain. Second type is *models*, where a set of constructs are put together to present a situation in order to get a better understanding and solution of the development. Third type is a method that gives guidance on how to produce the models, and then goes step by step on how to use them to solve the problem. The last type is called *instantiations*, an instantiation is a working system that demonstrates a construct, model, guidance or theory in a working computer-based system as defined by Briony J Oates [22].

1.3.3 Experiment

In academic research, an experiment is a way to investigate cause and effect relationships. The goal for this experiment is to prove or disprove any links between the factors in play during the experiment and the outcome of it. An example of an experiment could be a research starting out with a research topic, which evolves to a testable theory. The research will then form a statement based on the said theory known as the *hypothesis*. Finally an experiment is designed to either prove the hypothesis, or disprove it. The actual content of the experiment can vary greatly from the process, duration, participant and location. This is related to the topic of research and the goal of the experiment.

There are multiple experimental designs, one of them is called Static Group Comparison [22]. This means that the participants are divided into two groups where one of the groups are using a modified version, while the others are using the original. This can be used as a control group.

1.3.4 Observation

Observation [22] is not only about sighting, but includes all the senses; hearing, smelling, touching and tasting. The reason why observation is used as a data generation method is to find out what the participants did, rather than what they *said* they did during a test. Observations can vary greatly from different situations, in one case the observation might have a small group of particular interests, where in another case the focus might be broaden. In some cases of observations the duration may exceed a few minutes, while others can span over several years. It can be located in a lab controlled experiment, but places like workplaces or public areas are also common. It is the flexibility of observation and the possibility to be applied to almost any research that makes it substantial as a research method.

1.3.5 Card Ranking

Card ranking is a qualitative research method that was first introduced by psychiatrist William Stephenson (1935). In this test method a set of cards containing different items will be given, and the tester will be prompted to sort them in the order depending on instructions. For instance, each of the cards could contain a statement, and then the instruction for the tester could be to sort them in order as they are ranked. The method will give a better understanding of relationships between different items from the testers view. Instead of using a vague description from interviews, card ranking will give precise and measurable answers, which can be a great advantage.

1.3.6 Quantitative Data Analysis

Quantitative data is rather easy to analyze and transforms from data to findings because the result from quantitative research is often in the form of numbers or models. Data is primarily generated by experiments or a survey, but other data generation methods can also work. The quantitative data comes in four types [22]. The first type is the *nominal data*,

this is when the data is not in the form of numbers, but rather in categories. An example for such data is gender, which is either male or female.
Second type is called *ordinal data*, this kind of data is in the form of numbers, however this data will be in an order, and can less likely be performed with any arithmetic operations. For example a set of students can have their experience of driving cars ranked from 1 to 5. It is known that ranking 4 signifies better than rank 2, but not what difference or what the interval between each rank is. This is decided by the person who did the ranking.
Third type of quantitative data is the *interval data*, this is where the difference or interval between two measure points are always consistent. For instance, the year range from 1992 to 1996 is the same as 2004 to 2008.
Last type is *ratio data*, which differs from interval data because it can contain values with zero. Because the value zero exists it is possible to say a person who is 24 years old is two times older as someone who is twelve. Consequently, it is not possible to state that year 2000 is twice as many years as year 1000, since there are many more years before year 0.

1.4 Outline

This thesis are mostly structured in two parts, the first part consist of chapter 2 to 3 which primarily focus on identifying the set of functionalities for gaze interaction, and the modules we have developed to showcase them. The second part consists of chapter 4 to 6 which is focused on the usability test.

Chapter 2 Chapter 2 is about the literature study performed at the beginning of the study. In this chapter the focus is on finding the basic functionality for gaze interaction and potential use situations that is suitable to it with. It begins with historical events and the current state of eye tracking, then continues to point out limitations eye tracking technology have for gaze interaction, both in terms of hardware and software. Later on, the chapter showcases the analysis of basic functionality sets for touch devices, and a design for gaze interaction with this set in mind is produced. Finally, three modules are suggested for covering the theory behind these discoveries.

Chapter 3 Chapter 3 focus on the development of the modules, with the aim to find any challenges facing implementation of gaze interaction. In this chapter all three modules will be explained into depth in multiple aspects, including functionality, development cycle, result, potential problems and challenges faced during development.

Chapter 4 Chapter 4 begins the second part of the thesis, which primarily focus on the usability test where our modules were compared against corresponding applications using touch, keyboard and mouse instead. In Chapter 4 the main goal is to explain how the experiment was set up, the objective of the experiment, characteristics of the participants, context of use for different test cases and our own hypothesis.

Chapter 5 Chapter 5 presents the result of our usability test, the result is sorted into three usability metrics, namely efficiency, effectiveness and user satisfaction. In addition, another section will focus on heatmaps provided by our software recording tools, that shows the gaze movement of participants during the test. The concept of heatmaps are explained in Section 2.7.1.

Chapter 6 Chapter 6 is the chapter where the results from Chapter 5 are discussed, with our theory of why some patterns came to be. The analysis is sorted into individual modules, where each module is sorted into the usability metrics efficiency, effectiveness and user satisfaction. This chapter ends with a list of factors that we believe are the reason why gaze interaction performed the way it did in the usability test, also what can be done to improve it.

Chapter 2

Literature Study

2.1 The Eye

Eye sight is one of the five senses of human body, and it is used to provide us with information in the form of images. Normally our eyes gathers information by movement and then fixate on a certain object. The movement phase between two fixation points is called *saccade*, and an example for saccade is illustrated in Figure 2.2. Saccade can be categorized in 4 formats from the intended goal, the first being *visual saccade*.

A visual saccade is when a certain visual stimulus have occurred, and the sight has moved towards this stimulus. Second being *antisaccade*, which is when the fixation has moved away from where it was previously. Third format is *memory guided saccade*, and takes place when the saccade is moving towards a certain point without any visual stimulus. The last saccade is known as the *sequence of predictive saccade*, this is when the sight is kept on a certain moving object [26].

Saccades are used to keep certain objects in focus on the fovea shown in Figure 2.1, or in the case of antisaccade, to not keep it in focus. Thus by being able to know where the saccade movement is headed, it is possible to predict the object of interest, as found by Velichkovsky and Hansen[35]. When human eyes are fixating they are actually not standing still, in reality they are moving extremely fast around where the focus point is located.

2.2 History of Eye Tracking

Research related to eye tracking have been conducted for over a century. With the earliest documentation being a French researcher named Javal [17]. Javal conducted his research by observing eye movement during reading, this was accomplished by using afterimages and compare their displacement to real images. Using this technique, he was able to con-

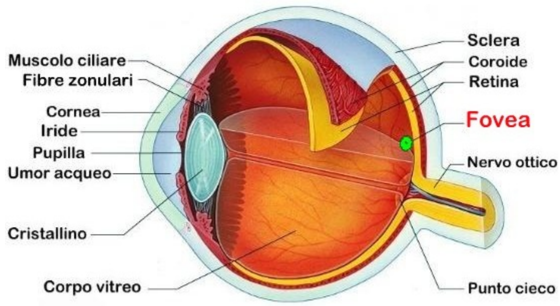


Figure 2.1: An illustration of the human eye

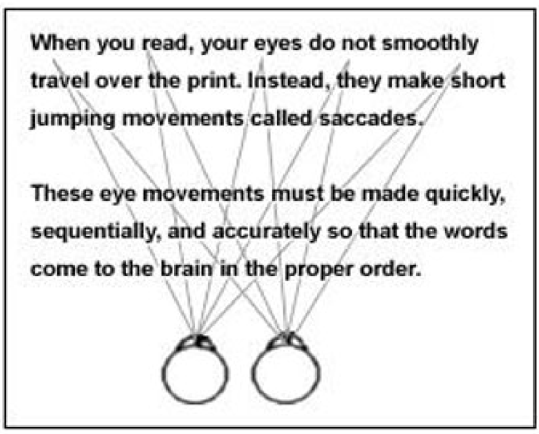


Figure 2.2: An example of saccade during reading

Figure 2.3: A set of different eye tracker devices over the last 80 years

clude that our eyes glide horizontally with no vertical deviation during reading. He also noticed the rapid movement between eye fixation, and it is widely believed that the term *saccade* was first introduced here as a reference to rapid eye movements. In the coming years the scientist Huey[14] continued the work from Javal and published *The Psychology and Pedagogy of Reading*, during which he also created the first ever eye tracker in 1908.

The first non-intrusive eye tracking technology was developed in the early 20th century by Dodge and Cline [6]. This technique involved the use of mirrors to observe the movement of the human eye's cornea. Before the usage of mirrors began, the used technique consisted of an ivory cup with a bristle on it attached to the user's cornea. In the Figure 2.3 the first eye tracker from top left is an eye tracker used in 1937 related to a psychology research.

The first ever recorded case of using eye tracking related to usability engineering [24] was done by Fitts, Jones and Milton [9], where they used eye tracking to find the pilot's eye movement in an aircraft during an instrumental landing.

In the 50s and 60s devices with head mounted eye tracking [12] was introduced. The first device for military and research usage is similar to the eye tracker shown in Figure 2.3. These new devices made it possible for users to move their head freely during tests, and therefore leading to better research and testing options. During the 70s the eye tracking technology was mostly used in the field of psychology and physiology, with usage in finding relations among eye movement, and perceptual or cognitive processes. In this period computer interactions were not a big field for eye tracking, this was because at this time computer interactions were mostly text based, or through punched paper cards and tapes. In the 1980s multiple market groups started using eye tracking for research. They used it to measure the effect of which ads were seen on physical objects in magazines to determine which objects that were noticed [8]. Previously techniques, such as *voice stress analysis* and *galvanic skin stress*, were used to evaluate effectiveness of ads (both are a form of lie detection). At that time eye tracking was mainly used to help researchers in better understanding of how our eyes and mind cooperate to distinguish images, problems, and literature [25].

In the late 80s and early 90s eye tracking began to be incorporated into print and screen designs. *EURO RSCG*, being one of the largest marketing and advertisement companies in the world, began to use eye tracking technology to study and analyze information from the World Wide Web [29]. Until this point it was assumed for most web designers to design web pages similar to traditional print and magazines, results from research helped to change the general design of web pages to what we have today. In the same time period multiple researchers started to experiment with gaze interactions [36], and this has been an ongoing study ever since [16]. A goal for this research was to test the viability of gaze interactions. Extensive experiments from previous researches shows that comparisons of gaze interactions are more efficient in certain tasks, such as selection of objects and navigation on a map. However, gaze interaction also revealed to have many flaws, such as sending commands back to the computer is seen as a diificult task.

Since 2001, and today, companies like Tobii Technology have been developing eye tracking devices to allow disabled users to interact using their eyes [27], one of their products is shown in Figure 2.3. Today eye tracking devices have become smaller and mobile, some of them are implemented as glasses such as the Tobii Pro Glasses model, also shown in Figure 2.3. With the rise of smart phones and tablets, eye tracking also becomes mobile. It is already possible to buy affordable personal eye tracking kits for tablets from *TheEyeTribe* company [11], although it is still in development, the current design is shown in Figure 2.3. Demonstrations already lets the user control and interact with tablets using basic functions with their eyes. There are some research also being done with gaze interaction on multi-touch devices, such as the work by Pfeuffer, Alexander, Chong and Gellersen in 2014 [23] where gaze interaction was implemented on an Microsoft Surface device as supplement to touch interaction.

The beginning of 2016 marked the first commercial computer with an eye tracking device integrated. This computer is called *GT72S 6QE DOMINATOR PRO G TOBII* manufactured by MSI [21], the device is shown in Figure 2.3. It incorporates gaze interactions on a multitude of software, but most notably on the game *The Division* by Ubisoft [34], where the gaze interaction works like an extended mouse.

2.3 Current State

Eye tracking systems today are designed to find where the user is looking by analysing images of the user's eyes with one or more cameras. The camera captures images in combination with infrared lights. By using infrared lights on the eye a reflection will be produced on the cornea, known as *corneal reflection*, or *glint*. In combination with po-

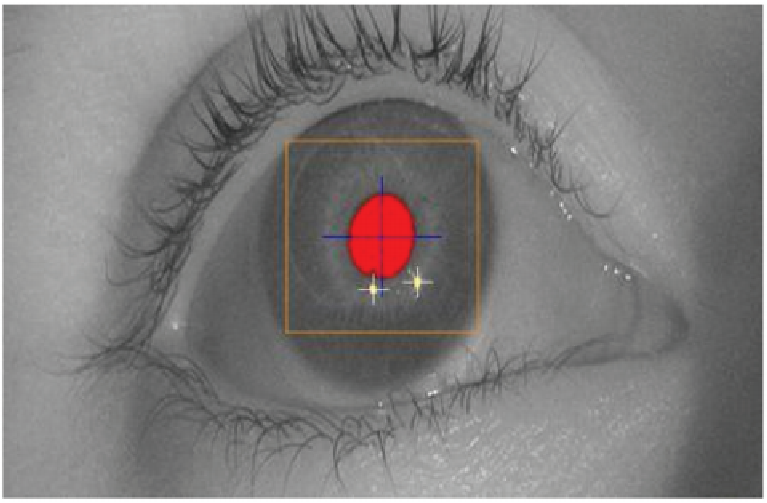


Figure 2.4: Finding corneal reflection and iris position using infrared lights

sition of the iris it is possible to estimate current position of the gaze. Infrared lights are invisible for the human eye, and therefore will not distract the user. Figure 2.4 shows how a single image looks like during eye tracking, the red circle is the position of the iris and the two small white stars are the corneal reflection. The image is taken from an open source ITU eye tracker [28].

Normally eye tracking devices come in two different types, either remote or head mounted. The remote device is often placed at a certain distance away from the user, often below the screen where the user is looking at. In Figure 2.5, the user is sitting a certain distance away from the eye tracker and the screen, with the eye tracking device placed below the screen. This specific setup is designed to make sure the eye tracking device does not obscure any vision for the user. In a head mounted solution, the eye tracking device is incorporated into a helmet or pair of glasses. One of the eye trackers in Figure 2.3 displays Tobii's latest and leading eye tracking technology per date - *Tobii Pro Glasses 2*.

A futuristic ambition for gaze interaction studies is to replace touch, mouse and other pointing interactions, with gaze point (where the user is fixating) instead. Multiple researches that have been conducted in the field, such as the one by Ware and Mikaelian [36], suggest that gaze interaction is as effective, if not better, at navigating around the screen. The downside of replacing mouse with gaze point is that our eyesight lack the capability to interact similarly to mouse clicks. Several ways to interact have been tested, including blinking, dwell time and gaze gestures.

Blinking in its nature is unreliable, because sometimes it can not be controlled by our selves. It also put a lot of stress to keep blinking all the time, and the eye tracker will lose contact with the gaze for a short moment every time the user blinks. Dwell time is a concept implemented with a timer, typically as 1 second, and if the user keep his sight on the

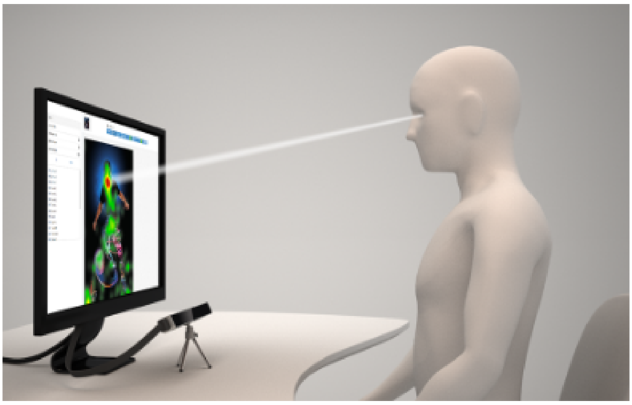


Figure 2.5: An illustration of how eye tracking devices could be set up

same position for more than one second it sends a click signal. This implementation might also be unreliable, because if the gaze is located at the same point for a short time, without the intention to click, it could easily create confusion. Dwell time has to be higher enough to avoid misunderstandings, but too high time limits will constrict the user's efficiency.

With the approach of using gaze gestures the eye movement needs to be predefined with different functions. For instance, by drawing a circle with the saccadic movement could be counted as a click. This path is better suited to perform certain tasks, such as writing passwords using eye gaze, or play the next song in a play list [5]. Because of the complexity of performing gaze gestures, it is safe to assume a user will not mistakenly perform actions without the intention, but because of the complexity it might take users multiple attempts in performing correct handling of gestures.

The use of gaze pointing helps elderly, handicapped and others where hands are not available to use computers, tablets, smart phones or other devices. Although the pointing with gaze is faster than traditional mouse and touch interaction, the human eye is never motionless or perfectly still, which makes it hard to maintain high precision. There is also the problem with click events using gaze interaction, normally the technique used is by dwell time, which might result in unintended clicks. Distinguishing what user want to interact with and look at is known as the *Midas Touch problem*. In short, Midas Touch Problem is about King Midas who could turn everything into gold with his touch, the problem was that it even affected the food and later his own daughter. This is similar with eye tracking that have problems distinguishing click events with normal movements. One of the solutions to Midas Touch Problem is to add a "clutch" operation, a button, level or a pedal that can be used to signal and determine if an interaction is intended. A good "clutch" operation should be quick to operate, it should not reduce any performance, and should not disturb users gaze pattern.

2.4 Gaze Interaction in Complex Tasks

In Section 2.3 we mentioned that gaze interaction is not a perfect interaction method for input events. It is difficult to integrate certain complex functions that are expected of today's electronic devices, such as input for writing and handling shortcuts. This section will explain multiple ways to solve these problems with gaze interaction, and the effectiveness of these solutions.

2.4.1 Writing

In any modern electronic device that is not entirely designed for a single task, it is necessary to have input for writing in one way or another. There are mainly two types of writing systems, they are called *direct typing* and *multi-tap* by Bee and André [3]. Direct typing is when a keyboard will be shown for the user to type into. This is the main typing system for both touch device and computers today. Before smart phones were on the market, the central typing system for mobile devices were multi-tap. This is when a single button corresponds to multiple letters. By clicking the button multiple times the letters will be cycled through.

To write with gaze interaction with either of the mentioned typing methods, it would most likely be implemented with dwell time, explained in Section 2.3. With dwell time the writing would occur by holding the gaze on a keyboard, or any other type of input button. The problem is the huge lack of efficiency. Assume the dwell time is set to half a second, meaning in order to write the word *hello*, a total of at least 2.5 seconds is needed. Time taken for movement between the letters is not accounted for in this scenario. In contrast, any experienced user will be able to use touch or keyboard to write these five letters in far less time.

It clearly stands out that writing with an eye tracking device does not correspond with ordinary writing solutions. To compensate these issues, gaze gestures were introduced for writing using gaze interactions. A first solution was presented by Isokoski [15] where he utilized a system called *Minimum Device Independent Text Input Method* (MDITIM). Originally this method was designed to be used with trackpads, trackball, mouse, and anything else that can be moved in a pre designated 2D area. It was reasonable straightforward to adapt it for gaze interaction, as the gaze will be used to move in a given area. The only thing needed to be replaced was a button which is switched with dwell time. MDITIM encodes all 26 letters in the alphabet into directions, where each letter is defined with a given set of directions. Three to four directions is needed for each representation of a letter.

For example, to let a user write the letter B, he would first need to move his gaze towards the south section of the 2D area, then continue east, and finally move towards west as shown in figure 2.6. The system needs to recognize interruption between letters, and this has shown to be problematic in the long run.

In the report by Bee and André [3], a new writing system called *Quick Writing* was implemented specifically for gaze interaction. Quick Writing keeps all letters in a center,

Figure 2.6: An example of MDITIM for writing letter B

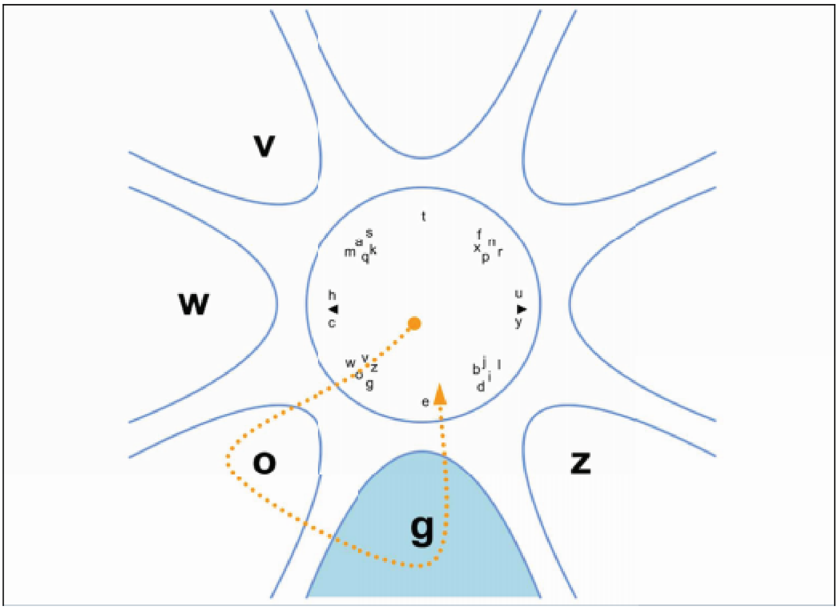


Figure 2.7: An example of using Quick Writing to write letter G

and divides them into sections surrounded by a circle. If a user for instance wants to write the letter *g*, he would first gaze in the middle of the inner circle, and maneuver out cross the section that have the letter *g* in it, then move to the outer section containing this letter, and finally move into the inner circle again to verify the gesture. This task is illustrated in Figure 2.7. It might seem a bit complex, but the report showcases this method as having vastly improved effectiveness compared to dwell time. This method has no clicking involved, which makes the chance of typos less likely. A problem with use of this method is training. While most users are experienced in direct typing or multi-tap, there are few who have tried this particular system. Therefore, the final result is unfortunately still far behind traditional type systems.

2.4.2 Shortcut

Multiple modern electronic devices have shortcuts included to increase the user experience, and of course for speeding up certain tasks for more experienced users. This is especially true for complex applications and softwares where there are numerous of set-

Figure 2.8: An example of a pattern where the path is repeating right to left and then back three times

tings and functions that will affect the outcome.

Shortcuts in term of gaze interaction is generally through the use of gaze gestures. An example would be to create a gesture for a given shape that will trigger a specific shortcut action. Figure 2.8 could have been a shortcut that required the user to look right to left, and repeat three times in a row that would trigger the shortcut when completed.

An experiment was conducted by Drewes and Schmidt [7], where they had a shortcut used with confirmation of a task. The shortcut required the user to gaze through the four corners of a window clockwise. The average result was 1905 milliseconds with a standard deviation of 600 milliseconds. They did not mention how many attempts users averagely needed in order to complete the task. Considering it took about two seconds to complete it, there is a high chance that several attempts were needed.

2.5 iOS Human Interaction Guidelines

Today's development of gaze interactions on multi-touch devices are still in an early stage where new possibilities are explored, hence there are no standard implementations of gaze interaction to be found. If the ultimate goal is to find a way to replace touch interaction with gaze, it is important for gaze interactions to have at least the same set of basic functions as those available. In order to find the basic functionalities for gaze we chose to analyze base interactions for touch devices using the iOS Human Interaction Guidelines by Apple [1]. By analyzing the basic functions for touch, we are aiming at narrowing it into a few core functionalities that could be implemented into software prototypes, and then replace it with gaze interactions.

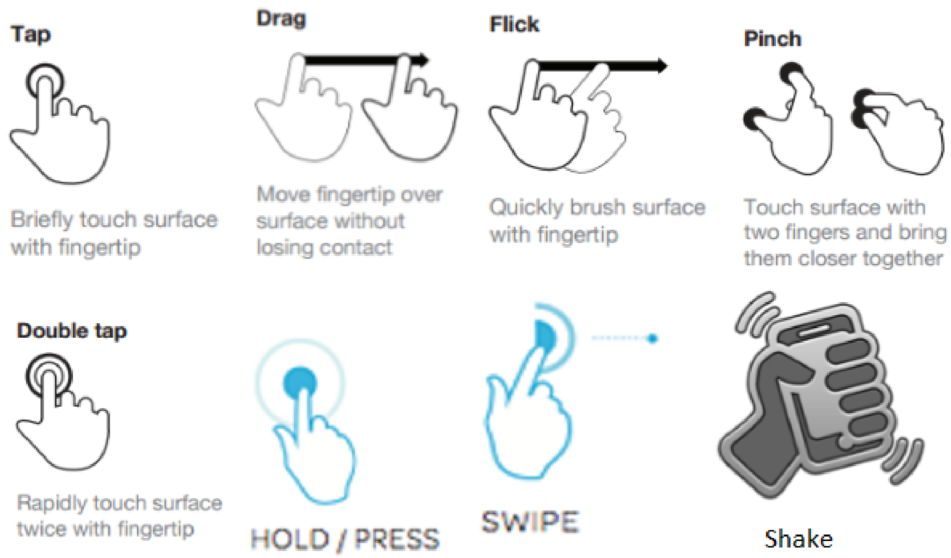
2.5.1 Advantages

There are several reasons why touch is better suited to be replaced by gaze interactions, instead of mouse and keyboard interaction. First and foremost is the simplicity in touch interaction techniques. With iOS Human Interaction Guidelines there are 8 basic interaction gestures, implying that if there is a way to make these 8 gestures with gaze instead, while keeping the same quality, then gazing could potentially replace touch gestures. All of the 8 touch gestures are presented below in Section 2.5.2.

Secondarily, another reason is the tolerance of precision. Touch interaction, in contrast to mouse and keyboard, often has to deal with precision due to the small size of the screen. Considering the same problem also haunts gaze interaction it could make the transition easier.

Furthermore, if touch gestures could be replaced with gaze gestures, it could also potentially compensate for keyboard and mouse. Touch is an interaction which already performs the same tasks as keyboard and mouse, and in many cases also better.

2.5.2 Touch Gestures



1. **Tap:** The first and probably most used touch gesture is tap. Tapping is generally used to select or control an item on the screen. The item varies from a link, to a picture, or an app. With distinct items the response is also different. The idea behind tapping is similar to mouse clicks, namely to interact with items, and this will also be important for gaze interaction

2. **Drag:** Second touch gesture is drag, which is often used to scroll through a page or navigate around on a map with panning.

3. **Flick:** Flick is generally a faster type of dragging, and will result in a faster pan or scroll response from the touch device.

4. **Pinch:** Pinch is used together with images and maps to zoom in and out. Pinching closer will zoom in, while doing the opposite with pulling the fingers away, zooms out.

5. **Double Tap:** Double tap is often used in the same manner as pinch. Double tapping will zoom in, and zooms out when the zoom limit is reached. Compared to pinch, double tap should be used between a few set of zoom levels.

6. **Hold and Press:** By holding or pressing over a certain point of text, a magnifying glass will show up to select a section. This interaction is especially useful when editing texts because of the lack of precision for touch devices. For instance, it is hard to pinpoint and select a specific letter that is a typo, or to add something between two previous words.

7. **Swipe:** Swipe is similar to drag and flick, and is in essence operated equally, but the difference is that swiping is activated at the end of the screen, followed by flicking towards the other side. Swiping is normally related to open sidebars, and other hidden elements.

8. **Shake:** The last of basic touch gestures is shake. Shaking in iOS has a standard response for undo- or redoing actions. In contrast to other interaction which is on the device screen, shake requires the user to physically wave the whole device back and forth.

2.6 Hardware Tools

Tobii X60 is the eye tracking device used in this research project, although the device is fully capable of reaching the research goals, it does include limitations that might affect the tests.

Firstly, the eye tracker have a freedom of head movement of 44x22x30 cm [32]. Meaning that once the eye tracker finds the user's eyes it will be centred, and the user can move in a room of the given measurements. Notice that it is in a 3D space, therefore in addition to where the user is looking at the screen, the eye tracker will also know how far the user's head is away from the screen.

Secondly, the eye tracker has a data collection rate of 60 Hz. If the recording is lost or delayed it could cause issues. Assume we implement a dwell time approach, because of the loss or delay, the interaction might not respond as intended.

Thirdly, a connection between the eye tracker and the computer is through a LAN connection using an ethernet cable, which has a delay of 15 ms.

Finally, with the need of finding and tracking the user's eyes and process the data, an additional time frame of about 15 ms is required. These reasons put together might produce some noticeable delays that makes it difficult to solve entirely within the software.

There is also a problem with the size of the screen. We did not find the exact screen limits for the *X60*, but during a meeting with a representative from Tobii in 2015, we were told that the *X120* device is capable of handling screen dimensions up to 17 inches, and because X60 is inferior to X120 we can assume it is also designed to handle screens smaller than 17 inches. This does not mean it is not working with larger screens, but it is more prone to failure calculation of exact gaze movements from the user. In our setup the screen

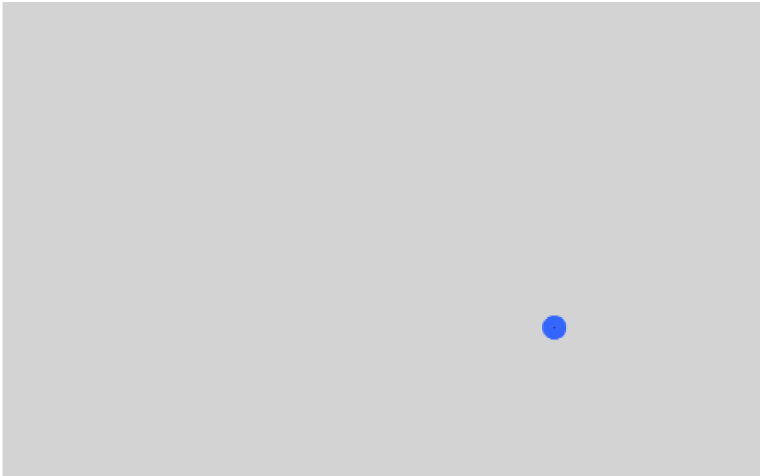


Figure 2.9: How a single moment during calibration process looks like

used is *HP LP2465* [13], which has a size of 24 inches, which is considerably bigger than the suggested 17 inches, and it might result in calculation errors during operation.

2.7 Software Tools

Different softwares were needed to handle certain functionalities for various reasons. In this section there will be an explanation for the tools used, how they were used and why.

2.7.1 Tobii Studio

Tobii Studio was used because it has a good calibration tool for the eye tracker, it also has the capabilities to record user gaze movements during testing it can illustrate the movements in a timeline afterwards. The program outputs several types of data that can be used to generate convenient graphical representations, such as *heatmaps* explained later in this section.

Calibration

Precision is a major issue concerning eye tracking in general, especially with the Tobii X60 device. It has problems with screens larger than 17 inches, and it is essential for us to have the best calibration we can get. Hence the use of Tobii Studio, where adjustments to the size of the screen is possible. This makes calibration more accurate, and effectively shrinking the screen to a size X60 can properly handle. The calibration process begins by a blue dot moving around the screen as shown in Figure 2.9, during this phase the user is supposed to focus as much as possible on the blue dot. When finished, a resulting window showcasing how good the calibration is will be displayed. A good result will look as shown in Figure 2.10, where the green lines are inside the circle, while a bad result will

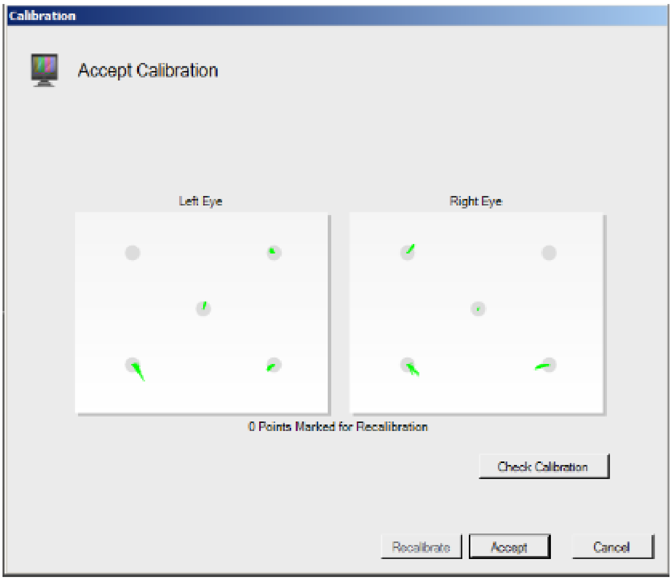


Figure 2.10: A result for good calibration

appear as Figure 2.11. Bad calibration will result in green lines all over the place. Also the calibration implementation in Tobii Studio is solid, and it allows testing on the actual calibration, which lets users test their calibrations against fixed points shown in Figure 2.12. The grey circle is where the software thinks the user is looking at when calibrated.

Recording

Tobii Studio records all gaze motions during a test, these can be reviewed for analysing. An important feature is displaying the recording as a heatmap, gaze plot or gaze opacity. These type of maps can be used to find patterns from the usability testing.

A heatmap is shown in Figure 2.13, and it illustrates where the user has gazed at, and highlights areas with different colors depending on how long the gaze has been located at a specific point.

Gaze plot is a recording of user's gaze in the same order as user performed them. This helps to understand a user's movement during the recording. An example of this is shown in Figure 2.14.

Gaze opacity is an illustration of what a user saw during his movement as shown in Figure 2.15. The black areas on the figure does not mean the user has not gazed here, but they represent areas without eye focus, and it can be assumed that the user has not noticed all details in these areas. Due to peripheral vision the user probably have a general idea of what is placed there. The area that is highlighted is where the user is looking at that

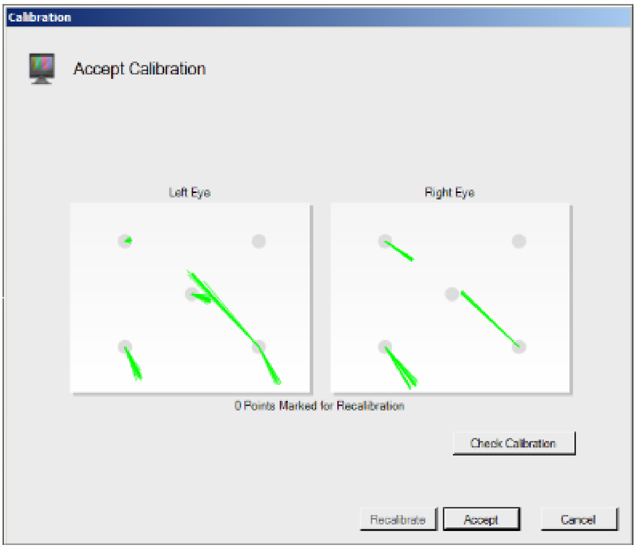


Figure 2.11: A result for bad calibration

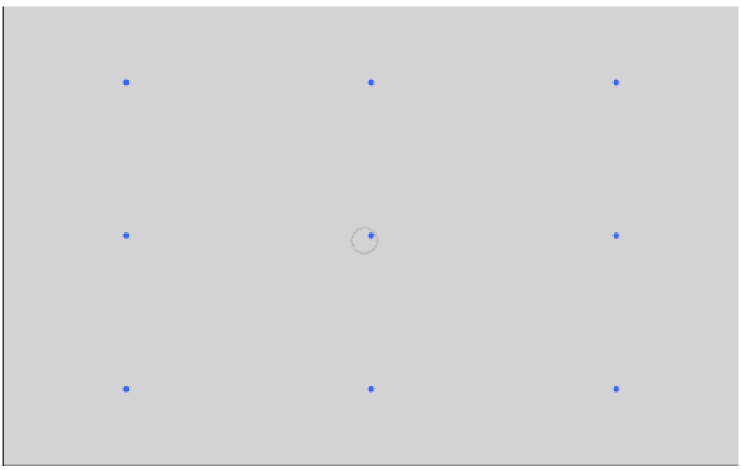


Figure 2.12: How it looks when a user tries to check their calibration

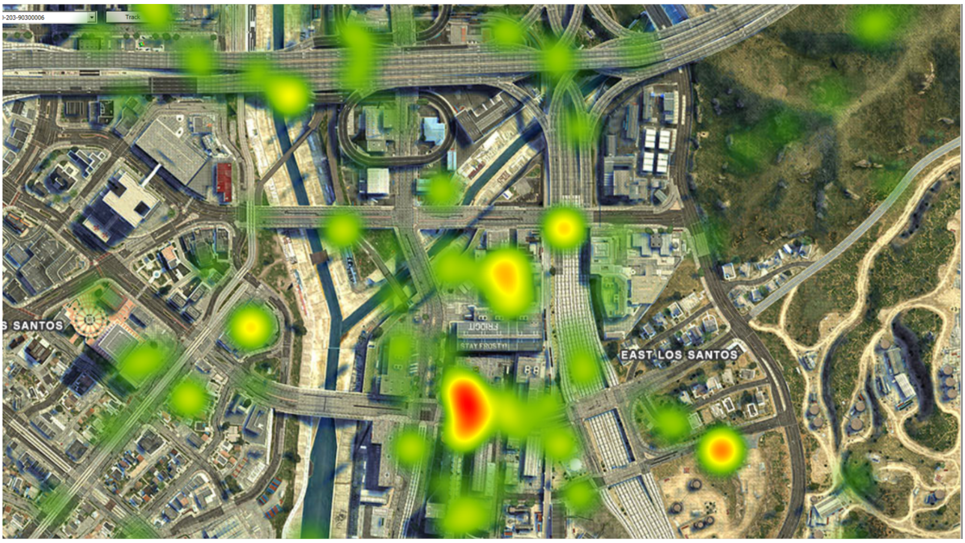


Figure 2.13: A heatmap produced by Tobii Studio

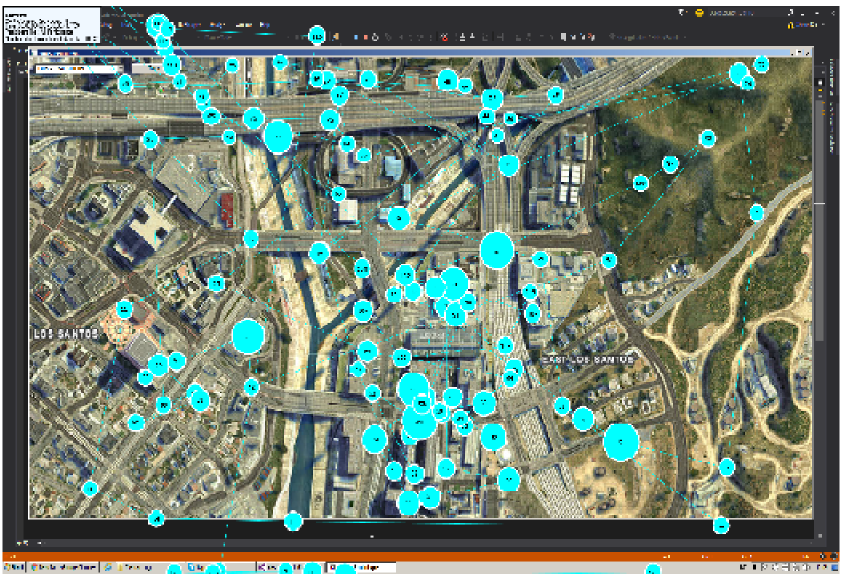


Figure 2.14: A gaze plot produced by Tobii Studio

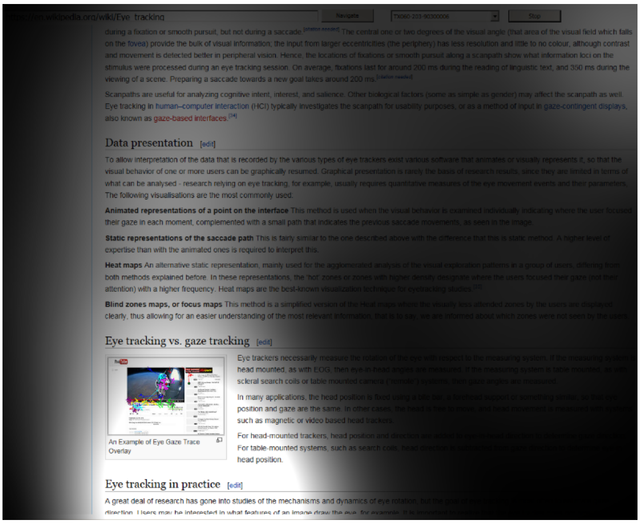


Figure 2.15: A gaze opacity map produced by Tobii Studio

specific time during recording.

2.8 Evaluation

In order to compare strengths and weaknesses between different interaction methods, a usability test was conducted. The test was set up using *Common Industrial Format* (CIF) [4] for Software product Quality Requirement and Evaluation (SQuaRE), also known as ISO 25062, which is developed by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) as a standard way to conduct and present usability test results. The goal of this standard is to set a format for measuring usability by comparing severeal factors. This can be used by usability professionals within supplier organizations to generate related tests and reports for the customer organizations in a CIF report. The customers can use these reports to confirm the quality in a software. In Chapter 5 there will be a CIF report used in connection of what has been tested, together with a resulting report.

In this section only the terms and definitions will be mentioned, since there are scant points of showing the actual report format, considering it will be used in more details later.

1. **Terms and definitions**
   There are multiple terms that is used in a CIF report, in this section the most important ones will be explained.

2. **Usability**

The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction, in a specified context of use. Note that this definition is from ISO/IEC 9241 which is different from Quality in Use definition from ISO/IEC 9126. Quality in Use applies other quality characteristics, such as understandability, learnability, operability, attractiveness and compliance. These characteristics are important for research with gaze interaction, because most of potential users has no experience with it. It is therefore important that they will understand and learn to use it swiftly. Also, the fact that eye tracking devices are prone to dust and need to be carried around, might play a factor with the operability.

3. **Usability Testing**
   A usability test refers to a test that evaluates a service or product by having the potential users as participants to test out the service or product in question.

4. **Effectiveness**
   The accuracy and completeness with which users achieve specified goals.

5. **Efficiency**
   Resources expended in relation to the accuracy and completeness with which users achieve goals.

6. **Satisfaction**
   Freedom from discomfort, and positive attitude towards the use of the product.

7. **Context of use**
   Users, tasks, equipment (hardware, software and materials), and physical and social environments in which a product is used.

8. **User**
   Person who interacts with the product.

9. **User group**
   Subset of intended users who are differentiated from other intended users by factors such as age, culture or expertise that are likely to influence usability.

10. **Goal**
    Intended outcome of user interaction with a product.

11. **Task**
    Activities required to achieve a goal.

12. **Assisting technologies**
    Hardware or software that is added to or incorporated within a system that increases accessibility for an individual.

13. **Assist**
    Tester intervention in the form of direct procedural help provided by the test administrator to the test participant in order to allow the test to continue when the participant could not complete the tasks on their own.

2.9 Summary

The objective for literature study was to find the basic functionalities gaze interaction needed to fulfill in order to replace the already exisisting touch interaction on multi-touch devices, and then create a few use situations that will cover the mentioned functionalities. In Section 2.5 it was mentioned a list of basic touch gestures, by analyzing the functions for each of the gestures we came up with a set of functions that gaze interaction needs in order to replace touch. This includes zooming, panning, scrolling of content and a replacement of tap and flick. With these functions it is possible to move freely on the screen, and interact with different objects. A reason behind why certain function such as double tap, swipe or shake was not included is because these are extensions of tap and flick respectively. Shake on the other hand, is more physically based and there are no direct replacement for it using gaze.

There will be in total three different use situations. The first use situation will be map navigation, because it will demonstrate gaze interactions ability to zoom, pan and at same time utilize the strength of gaze interaction, namely navigation as mentioned in Section 2.3. Second use situation involves an image gallery, the reason we chose this specific use situation is because on touch devices image galleries are often associated with flick or swipe, while on mouse and keyboard devices the general operation to switch is using the arrow keys. A simple image gallery covers many of the possible ways to interact with graphical elements. Lastly, the third use situation will use web browsing as a foundation. In this use situation the scrolling functionality will be tested and the tapping for interaction with elements on a web page. The reason we chose web browsing is mainly due to the popularity of such activity on electronic devices today.

Although it would be nice to try out complex tasks, there are studies that proves gaze interaction lacking in situations with such situations, as mentioned in Section 2.4, thus we decided to not proceed with modules for writing and handling shortcuts.

Chapter 3

Prototype

Using the insight gathered from the literature study, the prototype has been combined into four core functions.

1. Ability to move freely on the screen
2. Ability to select or interact with different objects
3. Ability to zoom in and out
4. Ability to pan and scroll

To achieve these functions we have decided to build three modules that will form our prototypes. Each module will be presented and explained. A module number four is also presented, and this is a combination of the three modules as one functional system.

Design and Create In this thesis an instantiation, mentioned in Section 1.3, is created that have gaze interaction implemented as the interaction technique. The instantiation is used to demonstrate a hypothesis of gaze theory replacing touch, keyboard and mouse.

3.1 Module 1 - Pan and Zoom

The first module for our prototype is called the pan and zoom module. Panning is known as the action of dragging, and is needed when the user can only see a minor section of an interface, typically a map. In order to see the rest of the map the user can drag it around using the pan action. Panning changes the view, but it will not change the scale on the map itself. If a user want to change the scale of the map they will need to use the zoom action. In modern touch devices, pan action is often related to drag, while zoom is related to pinch, both of which have been mentioned before in Section 2.5.

3.1.1 Purpose

For this module the purpose is to simulate a standard map interface, similar as the Google Maps (`https://www.google.no/maps`) or Bing Maps (`http://www.bing.com/maps/`) interfaces. Traditionally map interaction is handled with touch or keyboard and mouse, but our module will utilize gaze interaction instead. The interaction that is performed in this scenario is also used in many other software that are not associated with maps. If we can make a good gaze interaction solution to this, it will open doors for many other types of applications to follow.

3.1.2 Panning

In order to implement an gaze interaction prototype with panning functionality, we started off by developing an interface with a mouse as the source of input. It was easier to work with mouse to begin with rather than touch, because there is no need for an emulator. Besides, the pan gesture is similar for both touch and mouse, therefore we chose to use mouse as input at the initial stage. Subsequently, the mouse interaction was replaced with gaze interaction.

After a few tests a couple of issues appeared. By using mouse with panning there are actually two movements being handled, the user will first click somewhere on the map as point A. Next, the user will click and drag the mouse cursor in a direction. When the mouse button is released, a new position of the mouse cursor will be saved as point B. Multiple small vectors will be calculated during the movement between point A and B, then the map will then move with these specific movements. The problem with gaze interaction is that it is unnatural to create a point B, as users will expect the map to move towards wherever they are looking at.

Prototype 1

Our first working prototype used the eye tracker to keep two consistent points at all time. To connect with the eye tracking device, users need to select any Tobii eye tracking device connected to the computer in the selection box, and click *track* as shown in Figure 3.1 marked by the red circle. Assuming a user then looks somewhere on the screen, the first recorded point will be point A. If the user then gazes 200 pixels or more away from point A, point B will be created, and a vector will be made between A and B to move the map in this direction. An issue with this implementation is that the animation movement will start after point B has been set. When the user moves his eyes for more than 200 pixels away from B, the animation between A and B will be interrupted, and the new animation will start. When users move frequently the map will just move back and forth, creating a shaky and freezing look. It also had unknown bugs that sometimes freezes the panning movement.

Prototype 2

Our second prototype, which is also the final for this module, threw away the idea of recording two points and then calculate the vector between them. Instead the current gaze

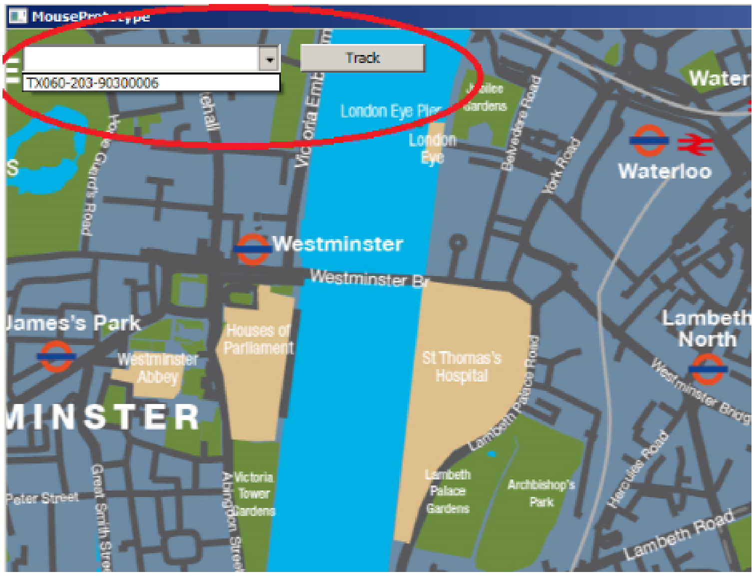


Figure 3.1: How to connect with the eye tracker device

point on the screen will be used. In this prototype a vector will constantly be calculated between where the user is looking at and the center of the screen. Afterwards the animation will start and drag the map towards where the user is looking. Although the logic behind this solution should have been straightforward, we still tried to solve the problem by thinking what is natural for traditional interaction such as touch and mouse, and not gaze interaction as a whole.

Several improvements were made to Prototype 2 as the project progressed. When the user is looking in the center, or close to the center of the map, the movement is turned off to avoid cases where the map is constantly moving due to fast eye movement. Panning speed will increase the further a user is away from the center of the map. Our final prototype is displayed in Figure 3.2, where the section marked in red is the section where movement is turned off.

3.1.3 Zooming

To begin with, we imported an implementation for a mouse wheel similar to what we did with panning, and later replaced it with a gaze interaction feature. It is problematic with zooming using gaze interaction to replace the mouse wheel, because of the lack to emitting information from our eyes. After a brief brainstorming for finding an easy and effective approach on handling zoom events, the final solution ended up with the user's head position as input. As mentioned in Section 2.6, the eye tracker is capable of measuring the distance of a user's eyes from the screen.

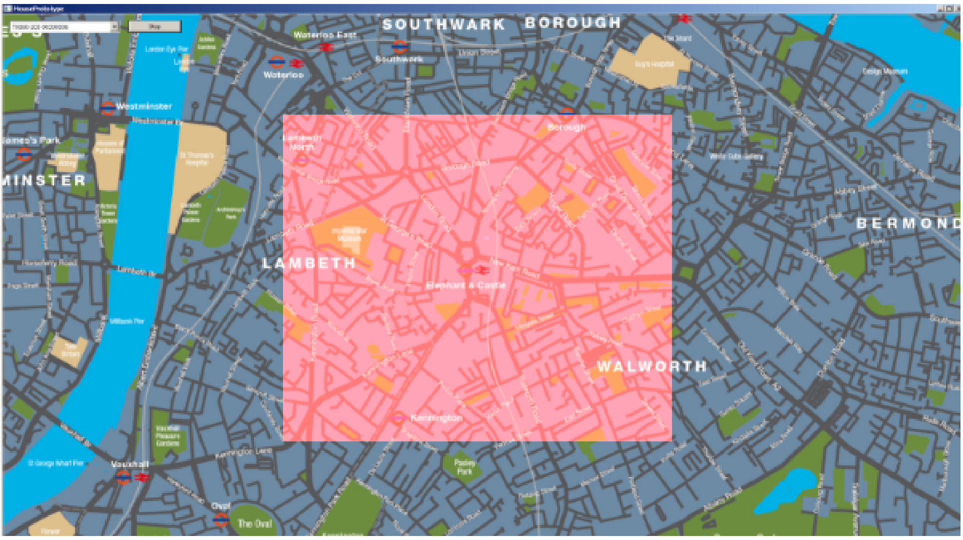


Figure 3.2: The approximately red section where the pan action will not occure

Prototype 1

Prototype 1 recorded the user's head position when the software first started. When the user moves his head a certain number of centimeters backwards (this distance is adjustable), the map will start zooming out, and the opposite goes for leaning forward. This requires users to keep their head position still, or at least inside the given centimeters area when they do not want to zoom in or out. Obviously this is a huge drawback for long term use and is seen as poor usability wise because the zooming events will fire off at unexpected occasions.

Prototype 2

Second edition of the zooming feature became the final version. It utilises an action button to solve the problem with head position described in Prototype 1. When the action button is pressed the current head position will be recorded as position X, and any movement while holding down this button will be recorded. If the movement is more than a certain threshold from position X, the map or image will zoom in or out depending on the direction the user is moving. During our testing in development it was clear that the threshold could be set to as low as one centimeter, and still have a neglectable error rate.

We did an improvement with the zooming effect by taking the coordinates on the screen, where the user is looking at, and always position the window frame in the centre of this position. At last, we tested with our supervisor where he suggested to disable pan mode while zooming is activated to avoid sudden conflicts due to head movements.

Figure 3.3: Swipe module with indication of the swiping areas in red

3.2 Module 2 - Swipe

Swiping or flicking is two of the basic gestures for touch devices, but essentially they performed the same gestures, it simply depends on where the touch point began initially, if it is on the edge of the device it is considered as swiping, otherwise it is flick. In this thesis we have decided to combine both of them and implement a single module representing both, and the gesture performed will be called swipe.

For touch device swipe is mostly used for navigating through different interfaces and is also a great way of switching images in a gallery. With swiping the user needs to swipe a certain amount on the screen to let the action be completed. There is a bit complicated process when transforming this gesture to gaze interaction, and we have discussed primarily two implementations for handling it with gaze interaction.

First, the simplest solution is to use dwell time, as mentioned in Section 2.3, an example of this requires a field on each side of the module window that is used for activation of the gesture. Each field will correspond to either a swipe event to the right or left. When the gaze hits one of these fields, a countdown should be used to verify if a swipe event should be triggered.

Second solution takes the approach of letting the users virtually drag with their gaze across the screen to activate the swipe event. In this solution the operation becomes more like a gaze gesture rather than a simple navigation. From Section 2.3, gaze gestures are prone to errors, considering how often swipe operation is used in an everyday usage, it would be undesirable to implement a complex solution.

Although both solutions had their drawbacks, we did not find any factor suggesting significant improvements for operational speed, intuitiveness and learnability using a complex solution. The first solution mentioned has been selected for this module, additionally it contains an animation effect to approximate a touch gesture.

Another topic that was discussed during the development is the size of the swipe area. There are two main solutions. The first one is to implement the area to cover the entire

Figure 3.4: Swipe module with indication of the swipe area in orange

height over the image as shown in Figure 3.3 as the red boxes. This implementation is inspired by applications that uses touch interaction, where the vertical position does not matter, and the whole horizontal direction of the physical swipe movement triggers it. An advantage is that users can easier perform the action, and the calibration error will take less affect. A disadvantage is that the user could look on the upper left corner for 700 ms, and it would swipe, which might not be the intention. Also, there are no visual indications to let the user know when the areas are activated, which could hurt the intuitiveness of the module. This feature was later changed to be around the arrow button instead, as shown in Figure 3.4 with orange boxes. Notice how the box is somewhat bigger than the actual button, this is an attempt to off set potential calibration errors.

Final version is shown in the Figure 3.4 where the current image is viwed in the center with two arrow signs at each side. The arrows give a hint that they can be interacted with. In this case the users should understand if the environment is for gaze interaction. Arrows suggest that an event will happen, thus making the interaction intuitive to understand. In order to operate the arrows, the user is required to gaze at them for 700 ms, at which point it would either swipe right or left, depending on the side the user was looking at. This action can then be repeated forever as long as users want, and if it hits the last image in the group it would simply start over at the beginning.

3.3 Module 3 - Scroll

Module 3 targets readability with the operation of scrolling. Anything from reading e-books, documents, to browsing the Web requires it. Scroll is similar to panning, however our goal with Module 1 was to see how users experience navigation operations on maps with gaze interaction. In Module 3 the interaction is targeted towards enhancing the readability by enabling the system to naturally scroll as the user reads.

Figure 3.5: The final prototype for Module 3 also known as scroll module

3.3.1 Implementation

The scroll prototype features a browser, shown in Figure 3.5 that incorporated gaze interaction as the scrolling mechanism, and moves the mouse around in the window itself. As the user clicks the track button, the system starts to track where the user is looking at, and it will scroll accordingly. The first test version had a stable scroll speed at all time, and whether it scrolls up or down depends on where the user is looking at compared to the center of the window. This is similar to what we had for pan mode in Module 1 in Section 3.1. A known issue that was discovered was related to the scrolling speed. If it is too high it will often scroll past where the user is looking at, and he will have to scroll back. This will create a back and forth movement for focusing on the next section where the user is reading. With a slower scroll speed it will be cumbersome to skip to a certain point, or head back to a certain point. Instead of doing the implementation as done in Module 1 with panning, where the speed is determined by the distance from the center of map, we created zones in the browser window as shown in Figure 3.6.

Each zone occupies a portion of the window and they all have a different set of scrolling speeds set when activated. Zone A is in the middle, and has scroll speed set to zero, meaning it stands still. As soon as the user gets to Zone B the text will smoothly scroll about 30 pixels or 2 lines of texts in the given direction. This scrolling distance is approximately 1-2 lines of text, and this will move a small portion of text inside Zone A which is unread text. If the user wants to scroll fast to a certain point they will have to move to Zone C, which have a higher speed. We call this technique *static speed*, in contrast to Module 1 where the speed would change depending on distance. By comparing feedback on both

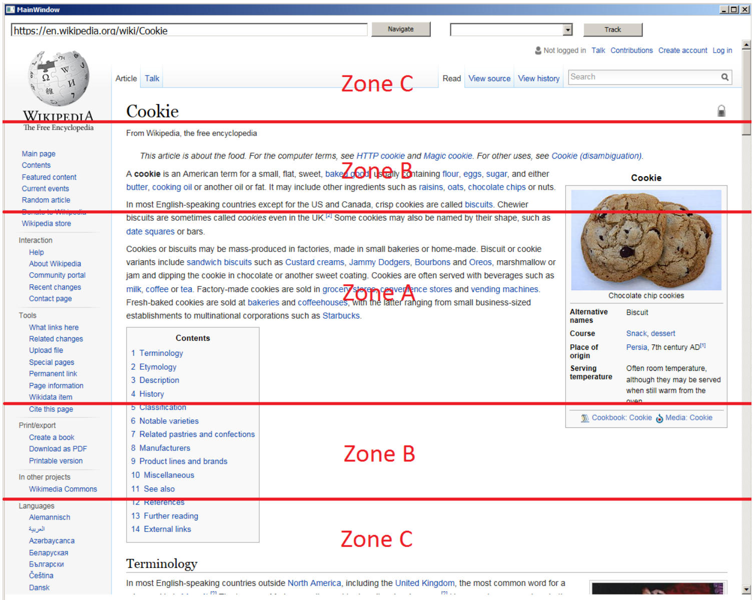


Figure 3.6: The approximately zone implementation of scroll module

techniques, we might identify the optimal speed technique for gaze interaction.

As mentioned earlier in Section 3.1 we inserted a simple mouse functionality. In the case of scrolling the plan is to move the cursor with gaze while hiding it, and make it able to click using an action button to simulate a web browsing experience.

However, there were two issues that made it difficult to implement. First is precision. It was problematic to get nearby the clickable elements on the web page, even with suitable calibration in Tobii Studio. By approaching a perfect calibration profile, after many attempts, we were able to get the cursor to hit in some occasions. Second problem is the mouse cursor itself. Because many web pages have their own style sheets (CSS files) that interacts with mouse animations, like hover effects, it is bothersome to rewrite styles that hides the cursor. To cope with this issue it is possible to use inline styling that overrides the webpage's design. This will fix most of the basic elements like links, but there could be several more complex elements (components) on a webpage that use individual style sheets that are bound from the main styles. On more simple pages, such as Wikipedia, we were able to accomplish the desired outcome. By consulting with our supervisor, he suggested to focus on the scrolling part with reading, which is why this module does not have clicking in our test cases.

3.4 Further Development

This section will explain what should be improved for the different cases in the future.

3.4.1 Module 1 - Pan and Zoom:

Maps were imported as large images, but an ideal implementation would be similar to Google Maps, where the map is dynamic with different details on each zoom level. Time constraints and porting issues are reasons for not making this as a priority. It appears that most official plugins for connecting map services, like Google Maps and Bing Maps, are not offering full access for dealing with other input interactions than the traditional touch, mouse and keyboard.

Other features that could be considered are the navigation speed for panning, and the rate of zoom levels. Currently these values are set from numbers found after a decent amount of testing, but a speed value that is comfortable for us might not be for others. It may be useful to add parameter settings that can set these values subsequently, but still have a default set of values that are comfortable for most user groups.

3.4.2 Module 2 - Swipe

There are multiple things that could have been improved on the swipe module, the first thing is the position of the arrows. Currently the arrows are placed on top of the images as Figure 3.4 shows, this could create situations where the user swiped without an intention. A better way would perhaps be to move the arrows outside of the image frame, having a section in black on both side of the image, and position the swiping area outside the image frame instead.

Another possibility is removal of the dwell time implementation all together, and instead use the action button from the module for pan and zoom. When the action button is clicked the software will calculate where the gaze is currently at, and depending on which side, the image will swipe towards this direction. There are multiple advantages with a solution of this kind, most important of all is the freedom of gaze movements, where the user can use the eye's freely and swipe without gazing in a specific section. Another advantage will be the low amount of unintentional swipe actions, due to the fact that an action button has to be pressed. To counterbalance bad calibrations, we suggest when the action button is pressed, an illustration displaying current gaze point on the screen as a blue dot should show up. A line should appear in the middle of the screen, where the user can move the dot into either side to confirm swiping in a specific direction.

3.4.3 Module 3 - Scroll

An important extension for the scroll module that should be merged is a proper functionality for making elements clickable. One technique to consider is a delayed movement, meaning that if the cursor hits a clickable element it will stay there unless the eye moves a substantial distance. However, there will be a problem with this implementation when multiple links are grouped close together. How should a user switch from one link to another?

Another approach is to zoom in on the element, which is the standard method on most touch devices. Often users will zoom in to click a link on touch devices because the screen size is small, or the precision of the thumb is not as accurate as a cursor. Gaze interaction can have the same solution by using zooming to click on a desired element, although this will require more time from the user.

One more improvement is to target the interface further towards apps and web pages. Currently most web pages have a maximum length, and in the bottom it might be placed a next page button or something similar. Using gaze interaction it would be beneficial to have endless pages to keep scrolling through. Most applications for touch devices has this feature, where it is possible to scroll forever as long as there is more elements to load.

We had a final idea on how to make it easier to hit a clickable element, and it has been called *delayed movement*. The concept is about letting the gaze hit a clickable element, and it would stay there until the gaze has moved a substantial distance. Another approach is to check if there are any clickable elements nearby. When this is true, the gaze position should point or highlight it. Both techniques should improve user experience by covering up calibration errors, but it requires further testing.

3.5 Module 4 - Combining the Previous Modules

After the completion of implementing and testing Modules 1 - 3, and completed the first phase of the usability test, we noticed that we were ahead of time. Therefore, by using the feedback from the usability test, a fourth module was proposed to be implemented. Module 4 will merge all previous three modules, improve them, and combine them to a new implementation.

3.5.1 Combination of Modules

The module will simulate a Google Maps interface, shown in Figure 3.7. Left side contains a window with information, and includes an image gallery showing the location, weather and some facts about the current location. The right section will display the actual map for navigation. If the user selects something specific on the map, the left side will be updated with relevant data.

This combined module will utilize our previous modules 2 and 3 to simulate the left side, where the swipe module will be the image gallery and the scroll part will show a text. While the right side will be replaced by the pan and zoom module. Figure 3.8 shows the complete module, where the scroll is situated on the lower left part of the screen, swipe is on the upper left side of the screen, and pan and zoom is located at right side.

In addition to the module combinations, a new functionality was added. In order to simulate the interaction users have on Google Maps, it is needed to click a certain point on the map in order to update the content on the left side. This module also had something similar, we designed ours as a zone experience. The map on the right side is separated

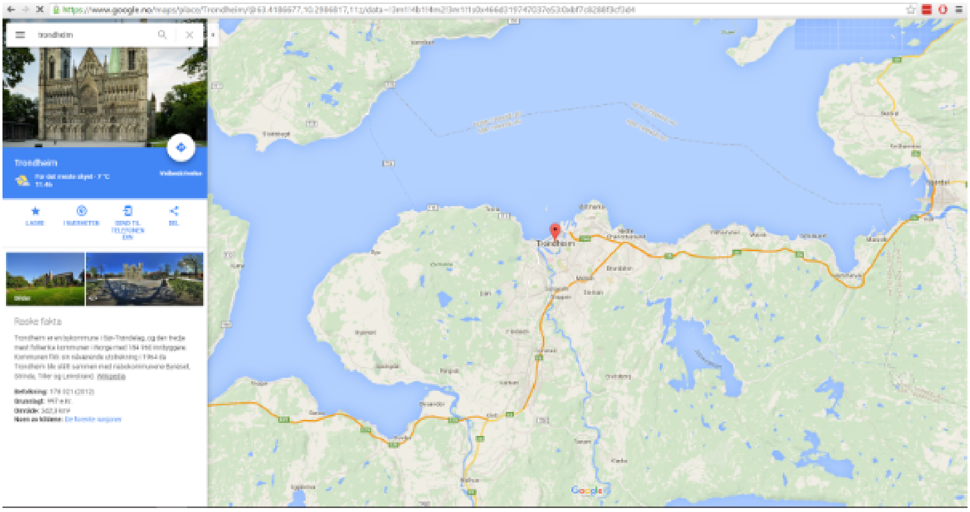


Figure 3.7: An example of Google Maps of Trondheim

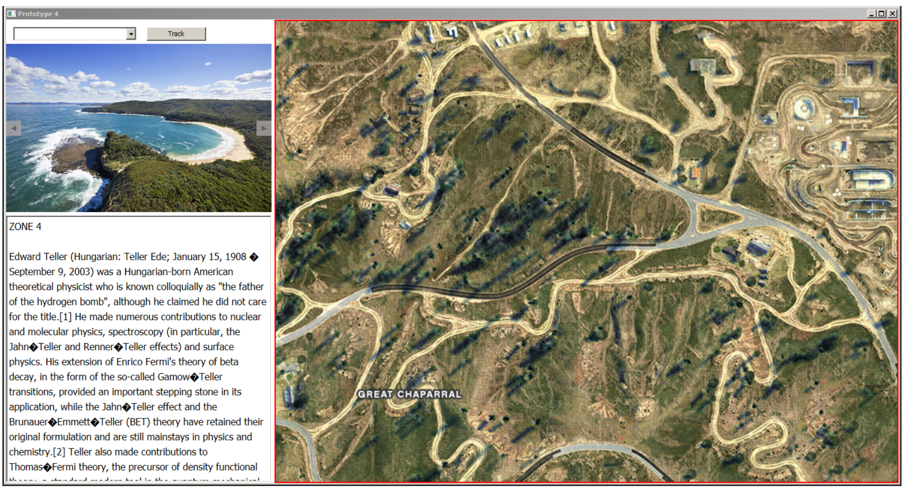


Figure 3.8: The final prototype of Module 4

into 9 zones. When the user is within a specific zone, the elements on the left side will be updated.

3.5.2 New Improvements

During the first iteration of usability testing it was noted the panning action is constantly active, which resulted in undesirable movements that were both confusing and annoying. In this specific module there are 3 window frames with dynamic content, this will make the interface a total chaos if each window frame react simultaneous from gaze gestures. There was a need to get control of this issue, and not let the problem prevail. Therefore, an action for pausing inactive windows had to be added.

There are two ways we could have implemented a pause action. One is to implement an extra action button, in which case it would be necessary with an extra button to let the module function properly. There is already a button for operating the zoom action within the map window, and to implement pausing, a second button is needed. Adding buttons might not seem to be such a bad decision, but it is more contradictory to our initial plan which is to have the eye interaction doing most of the work. Another approach is to keep using the current button, but having a twist on it. For example, a double click could trigger a pause action, which was tried with the first implementation. However, it proved to be misconceptions between the two events. In the end, it was settled with using two buttons instead.

There was a need to separate the window frames in a way to only let a single one be activated at all times. An inconspicuous method for this is to let the pause action handle and switch active frames. All views will be paused expect from the current view that the user is looking at. Another way is to only pause the map module at the right side. This will leave us to deal with precision between the scrolling and swiping module, but at the same time let us test the pause action. In addition, it is possible to make a concept we called *window focus*, which is a combination with the pause button to make a rotating focus solution on each window frame. The focus is initiated on the main module with a map, this module will be highlighted shown in Figure 3.8. Notice the red frame around the map, that means the map is in focus, and is the only window that can be interacted with gaze. When the pause button is clicked, the focus is passed on to the next module, and now only the image gallery with swiping is activated. Furthermore, another click will activate the scrolling module.

We ended up on selecting the mode with window focus. Two buttons were already implemented, and it is difficult to get precision between several active frames correctly, due to the vast difference in calibration. A small area between the two window frames on the left side would be necessary if the focus mode is not used. Combinations of a small area and poor calibration would mean unintended movements, which results in confusion. On the other side, by enlarging this area it would be unnecessary waste of space for virtually no advantages. Consequently, the better approach is to use focus mode until near perfect calibration is achievable.

An important phenomenon noticed during the tests is associated with the button placement. During the first iteration of usability testing, the action button was located as a left click on the mouse button. This created the illusion for a few participants that moving the mouse had some effect to the module, which created a puzzling effect. To avoid this we moved the action button onto the keyboard.

A potentially better adjustment for scroll speed was added for the final module. It is now dynamic, meaning the speed goes incrementally faster, while previously the scrolling factor was a static value for each scrolling section. Generally, further away from the window center will increase the scrolling speed in a comfortable pace. This gives the user control, instead of a rampant increase from one section to another.

3.5.3 Problems Encountered

One of the main problems during the whole implementation for this module is the screen size. Gaze interaction does not have flawless precision yet, and therefore it is favorable to make interactive elements as big as possible to decrease chances for users missing them. In an ideal situation interactive elements should be big enough to physically be able to gaze into the center of it, and this point should be recorded by the eye tracking device at all times.

Previous 3 modules had separated windows, but being merged into a single one, extra space was needed in between to offset bad calibration. This results in a sacrifice for the size of interactive elements, such as the reduced size for interaction with swiping and scrolling. However, sacrificing these areas will require a higher level of calibration. Ideally the area size should be big enough to cover as much as possible, while not overlapping each other. If the area is too small means some users may not be able to interact with the software as intended. If the areas starts to overlap it could evoke situations where users are trying to swipe, but instead the map module is still moving which will cause confusions.

3.6 Challenges

A challenge that was experienced is the space in front of the eye tracker where it will be able to find a user's gaze. As specified in Section 2.6, the space for movement is in an area of 44x22x30 cm, which can be described as free head movement from 40 cm to 70 cm in front of the eye tracking device. In practice, 30 cm is a quite short range and considering the zooming function from Module 1 requires users to move their head towards or away from the eye tracker, it makes it easy to lose connection. In a probable event where the eye tracker has disconnected due to an out of range exception, the interfaces will stop and wait until the connection is restored back into range. In a real life situation this will restrict the user's movement when using these interaction techniques, which is not desirable.

Second challenge comes from the C# programing language. It exists minor documentation and support for gaze interaction from the language itself. Resulting in beneficent functionalities that are not working as intended with eye tracking. This includes finding elements a user is looking at, which would been straightforward with use of mouse as input. Computers might not be the best platform to implement gaze interactions at the current state, but with more development and support from companies like Tobii, it could be improved in the future.

Third and greatest challenge includes calibration, we had problems getting the position on screen to center where the physical gaze was located. Especially on swiping, where the swiping action will not trigger due to calculation of the gaze being mispositioned a few pixels from where the user is actually gazing at. The same problem caused us to await adding clickable functionality in Module - 3 Scroll, because it is challenging to get the calibration on exact points.

3.7 Summary

With implementation of the prototype, it can be formalized an answer to Research Question 3:

What are the major technical challenges implementing these gaze interactions with current eye tracking technology?

In Section 3.6 - Challenges, we observe some issues arising from the hardware, or a combination of hardware and software. These problems could be fixed by improving the eye tracking device, with better support and algorithms for calculating calibration. Although facing the challenges made it difficult, all modules were created to be functional and ready for testing. Boiled down, there are no direct problems with gaze interaction that makes it unusable or futile to implement. Nevertheless, there are still potential improvements to be made that could drastically improve gaze interactions.

Chapter 4

Test Design

Our test method consist of two iterations. First iteration will include Modules 1-3 from Chapter 3, while the second one involves the combination of the previous modules, known as Module 4. The results from the first iteration will be used to improve on the second iteration.

4.1 Test Objective

The main objective for our tests is to answer Research Question 4:

How do users assess the usability of gaze interactions for the use situations mentioned in Research Question 1, compared to similar interactions using touch, keyboard and mouse?

Our solution is to have three parallel tests for each module, where each test will have different interaction methods, in our case gaze interaction, touch interaction, and keyboard and mouse interaction. This section will explain the test process step by step, and go into depth on how this can answer Research Question 4.

4.1.1 Experiment

For testing out the modules an experiment was proposed, as described in Section 1.3.3. This experiment uses the *Static Group Comparison* method [22], but instead of having two groups, this testing will consist of three interaction techniques. For example a participant might be given the task to navigate through three sets of images, one with gaze, another one with touch and the last one with keyboard and mouse. This will correspond as one test case. By having participants using gaze, touch, keyboard and mouse interaction, we can compare the result between them and use touch, keyboard and mouse as the control groups.

4.1.2 Observation

During the test one of us will have the role as observer, where the main task is to note down any difficulties, errors, assists and comments participants had during the test. A camera will record the participant's movement and conversation during the experiment, which can be used for analysis at a later stage. The eye tracker device will output relevant data, and we can use this to generate convenient graphs, as explained in Section 2.7.1. All data generated from observation will be separated into different interaction groups, which can be analysed to find any patterns and similarities among gaze and the other two interaction methods.

4.1.3 Card Ranking

After the participants have finished a module, they will be given 3 cards for ranking, namely eye tracking, touch, and keyboard and mouse. The participants will be tasked to sort the cards in the order they felt most comfortable with. They are encouraged to briefly give a reason for their ratings.

4.1.4 Test process

This section will explain step by step the process each participant will be taken through during the test. A more detailed version of the test plan can be found in Appendix B.

- **Step 1 Introduction**
  When a new participant arrives we begin the process by introducing him or her to our work, the goal of the project, a consent statement, lab setup, the hardware and software that will be used, and an introduction to how the technology works. The consent statement can be found in Appendix A.

- **Step 2 Participant Information**
  Second step is to gather information about the participant, this includes relevant experience, occupation, age and gender. This step will be explained in more detail later in Section 4.2.2.

- **Step 3 Testing**
  Testing begins, the participant will be starting with one of the three modules, where each module will be separated into three interaction methods. Assuming the participant is going through Module 2 - Swipe, the participant might first start with the gaze interaction, this means the participant will be able to swipe through multiple images, and perform certain set of tasks. When the participant is finished with gaze interaction, a new test will start, but this time on a touch device. At the end there will be a similar task on a computer using keyboard and mouse.

  A typical test set is illustrated in Table 4.1. All interactions will use three image packs that rotates for each participant, meaning Participant 1 might use image pack A for gaze interaction, while the next participant will use either image pack B or C

Figure 4.1: A participant during testing

for gaze interaction. Figure 4.1 shows a participant in the middle of a test.

Table 4.1: One set of tasks during the test

Task 1	Swipe through all the images
Task 2	Find the yellow train
Task 3	How many pictures contains water?

- **Step 4 Card Ranking**
  When participants has completed all interactions for a module, they will be handed 3 cards with one interaction on each one. Figure 4.2 shows the card that was used as the touch interaction. The participant will then be asked to order the cards by what they were most comfortable with. In this step we will try our best to guide the participant to explain directly regarding usability measurement, what they felt was the main difference between using gaze interaction compared to the other two interactions, any advantage or disadvantage they could think of, and if there are any potential improvements.

- **Step 5 Repeat for Next Module**
  Participants will then go through the rest of the modules and perform same procedures. Step 5 will be repeated until all three modules are finished. In total there are 9 individual tests, using three interaction methods on three modules.

4.1.5 Purpose

Main purpose of these tests are to compare all interaction methods with similar tasks. It might seem pointless to request the user to perform touch, keyboard and mouse considering how much usage a normal person have with these interactions on a daily basis. However, this way the user will be able to compare them with a fresh mind instead of

Touch

Figure 4.2: The touch card from card ranking

recalling something they did earlier.

Other reasons for using this method is because certain usability parameters are easier to measure, where the two most important ones are *efficiency* and *effectiveness*. Effectiveness as explained in Section 2.8, is the accuracy and completeness of users achieving their goals, while efficiency is a resource expended to achieve the goal. In this experiment effectiveness will be about how easy the user completed the task, how many assists they needed for completing it, and number of errors. This will also reflect on other usability parameters such as learnability, since all participant have little or no experience of gaze interaction while having high experience on devices with touch, keyboard and mouse. How fast a participant will be able to learn to control and use gaze interaction effectively might determine if they can finish the tasks with the same effectiveness as the other interactions. Efficiency, on the other hand, is time based in this research, and it is the easiest factor to measure for this project. By measuring time spent completing tasks for all three interaction methods, it can be used to compare efficiency for gaze interaction against touch, keyboard and mouse.

4.1.6 Test Location

The test location is set to the lab where we usually work, the user will be required to sit in front of the computer where the eye tracking device is connected, with two of us sitting right beside of him. One of us will be responsible to guide the user through the test modules, which includes giving hints and asking questions during the test phase. The other one will have the role as observer, and take notes of the observations and answers questions the participant has.

4.1.7 Test Material

In the map module there are three maps used: Manhattan, London and Los Santos from the game Grand Theft Auto 5 (GTA5) [31]. The swipe module contains image packages and includes collections of nature pictures. In the module for scrolling a web browser was used. Participants were asked to read Wikipedia articles that covers *cookies*, *Katana* (types of Japanese swords) and *eye tracking*.

4.1.8 Test Measurement

For each interaction method within the modules, the user will be measured on different types of data sets. First one is time used on each individual task, second data set will be the errors that participants made during the test and any number of assists they might needed in order to complete the test. There will be recorded data of eye movement, and a measurement of graphs mentioned in Section 2.7 will be used. Finally a satisfaction measurement will also be used, which is the card process at the end of each module, where participants order the interactions they were most pleasant with. The document framework used for data recording can be found in Appendix C.

4.2 Test Setup

In this section we will present an in depth explanation of the different aspects of this test, using the CIF approach, described in Section 2.8.

4.2.1 Participants

The total number of participants for this usability test was 11. Potential users for this technology include anyone using computer and touch devices, which means basically everyone in today's modern world. We tried our best to find varied groups of test participants, and all our participants can be grouped into two groups. The first being young adults from the age of 22-30, 7 of the participants were in this group, and one of them were female. The other group is mid age people around 40 years old, all of them being males.

4.2.2 Characteristics

The younger group had the key characteristic of being familiar with computer and touch devices, some of them even had experience with eye tracking in the past. Overall they had a high degree of familiarity with the other interaction methods. Traditionally the younger group is also more receptive to new technology. The mid age group consists of faculty members found at the Department of Computer and Information Science (IDI). They have been working at NTNU for 10 or more years, and they all have extensive knowledge of computers and technology.

This group of participants does not directly reflect all the ideal users for our product, nor does it present the whole intended user group. Among the older generation, the participant group is possibly the ones with most technology expertise and highest capability to

Table 4.2: Table of participants information that was taken at the beginning of the tests

Test ID:	
Name:	
Gender:	
Education:	
Computer experience:	
Task experience:	
Eye tracking experience:	

adapt to new technologies quickly. While the younger group is mostly consisting of computer scientists who understand how software and interactions works. We believe these two groups will give good feedback on how to improve the module in an early stage.

Before beginning the tests, each participant will be asked a series of question regarding their characteristics, these questions are shown in Table 4.2. Computer and task experiences will reflect how familiar they are with the tasks during the test, which should result in a shorter time required or less assists required. Since nearly all of our participants have high computer and task experience, it was a factor that could not be quantified. However, eye tracking experience is something that have some effect on the results, considering participant with higher eye tracking experience should be able to adapt to gaze interaction faster, thus complete the tasks more quickly.

One important group that will help our case is handicapped people, because they may not be able to use mouse, keyboard and touch devices easily. Which means they could be the user group that benefits most from use of gaze interaction. Unfortunately we did not have the resource to recruit any potential participants from this group, and it was prioritized to develop the prototype modules.

4.2.3 Context of Use

Some of the tasks require a computer that is connected by keyboard and mouse, while the eye tracking device is also connected to a computer. Because the tasks for every case was supposed to be completed in the same room, participants will have to sit in front of a computer during the whole test, creating a context of use that is similar to an office job.

4.2.4 Context of Use for Touch Devices

Normally a touch device is something reasonably small, like a phone or a tablet. This results in great mobility, giving the user opportunity to use it everywhere. In an ideal situation the expected context of use for touch devices will be anywhere, but we could not illustrate this aspect for touch devices in our evaluation context due to the restriction of location.

4.2.5 Context of Use for Keyboard and Mouse

Keyboard and mouse is normally connected to a computer, although there exist hybrid devices that are more common today, such as the Microsoft Surface [18]. This device is a computer with a thin keyboard, and has a touch display. It has the possibility to disconnect the keyboard at which point it basically functions like a tablet. In this test setup the keyboard and mouse will be connected to a traditional desktop, simulating a situation at the office, or home where a less mobile device will be used. Our evaluation context matches exactly what the expected context of use will be.

4.2.6 Context of use for Gaze Interaction

Eye tracking being something that is not a stand alone device will always require a tablet or a computer to be available. Today's eye tracking devices are small enough to fit on a tablet, therefore the expected context of use will vary from device to device. The ideal situation for gaze interaction will be when hands are available, since both touch and keyboard require hands exclusively to be able to operate. Consider these situations: Working on car with dirty hands, wearing thick gloves on a construction site, a doctor in the middle of a surgery, or handicapped people who are unable to use their hands. These are all candidates for gaze interaction. We did not simulate situations like these in our evaluation context, since these situations will mean that gaze interaction is the only alternative.

We want to test how gaze interaction can replace touch, keyboard and mouse. Meaning that the context of use for our experiment will be a situation where all three interaction methods are available to the participant.

4.3 Computing Environment

In order to recreate these tests again, all used components will be listed in this section. This will include the main computer used, display, recording devices such as audio, visual and input devices the participants used. The eye tracker used is described in Section 2.6.

4.3.1 Working Device

The computer used for this test is from Supermicro Computer Inc Model 5036T-T [2], it has been used for a long time, the exact time can not be determined, but it is noticeable slower compared to modern computers. This computer was used both for testing cases involving gaze interaction and test cases for keyboard and mouse.
For the touch cases a Microsoft Surface Pro 4 [19] was used, it is both a computer and a tablet, known as a *hybrid* device. During the test it was used as tablet only.

4.3.2 Display Device

The screen we used is the model HP LP2465 [13], which has a size of 24 inches. We did not utilize the whole screen during our test, this is mainly due to the eye tracker lacking precision in the corners of the screen, explained previously in Section 2.6. The screen is used both for interaction with keyboard and eye tracking, although the whole screen was used during testing with keyboard and mouse, whereas 75 percent of the screen during gaze interaction. Surface Pro comes with a 12.3 inch screen, which will be used during the touch cases.

4.3.3 Recording Device

There are two other recording devices, excluding the eye tracker, used during the usability testing. Firstly an audio recording microphone, which records all conversations between us and the participants during the tests. Unfortunately there were no model on the microphone, but any audio recording device would suffice for a rerun of this test. Video recording was accomplished with the software Microsoft LifeCam Cinema [20].

4.4 Hypothesis

There are some important factors that will alter the whole result dramatically. Firstly, the implementation of the gaze interaction. There are no interaction guidelines for gaze interaction in the same fashion as the *iOS Human Interaction Guidelines*. Therefore all the interaction methods are designed by our selves and with minimal testing. Depending on how solid these interaction designs are, they will have a varied effect on the results.

Secondly, the hardware restriction. How accurate the eye tracker can pinpoint where the participant is looking will affect the results in distinct grades of severity for the different modules. For instance the module including pan and zoom will definitely be less affected by bad precision than the swipe module. In addition, this factor can also vary from user to user.

Due to these factors mentioned we believe gaze interaction should perform differently for all the modules, but for each individual module all three interactions should have equal or similar effectiveness with the tasks reading, panning and zooming. It is assumed gaze interaction will lack efficiency compared to touch, keyboard and mouse, and it is expected that participants will perform all the tasks correctly, but with lower usability scores and more time used over all compared to the other interactions.

The card order method mentioned in Section 1.3.5 should result in gaze interaction being noticeably less favorable among participants. This is caused by a multitude of reasons, but we expect experience to be one of the main factors, combined with implementation, flaws and interface design.

Chapter 5

Results

This section will showcase the data gathered during the usability testing for modules 1-3. The data gathered is an important part for answering Research Question 4 from Section 1.2.1:

How do users assess the usability of gaze interactions for the use situations mentioned in Research Question 1, compared to similar interactions using touch, mouse and keyboard?

If a comparison between the input interactions is conducted there is be a need to identify the key data, in our case we will use the usability metrics *efficiency*, *effectiveness* and *satisfaction*.

5.1 Qualitative Data Analysis

Qualitative data comes from recordings during the experiment, comments by the participants and heatmaps produced by participants gaze data. During the whole experiment the participants were encouraged to compare gaze interactions to touch, keyboard and mouse interaction. Since all participants were familiar with touch, keyboard and mouse they were used as a control group in the analysis. Much of the same was done with heatmaps, these were compared side by side against other heatmaps from keyboard and mouse. It is possible to locate patterns using this technique.

5.2 Quantitative Data Analysis

By using a combination of experiments and questionnaires we were able to generate multiple sets of quantitative data, including user information data and time taken during the experiment. There are two sets of important data, the first being completion time for tasks, this data contains the type *ratio data*. Second one comes from card ranking, described

Table 5.1: Average user information

Average age	32.09
Number of male/female	10 male participants and 1 female participant
Education of each participant	5 Computer Science, 2 Teacher education 2 engineers, 1 IT and 1 Electronic engineer
Occupation	7 students and 4 working for IDI
Average computer experience	3
Average task experience	3
Average gaze interaction experience	1.55

in Section 1.3.5, that resulted in *ordinal data*. Both of these quantitative data types are mentioned in Section 1.3.6.

5.3 Participant Data

Using the user information gathered during the test a table of user information was made, the total information is shown in Table 5.1. Note that experience levels are graded from 1 to 3, where 3 is the highest experience.

5.4 Efficiency

In order to find a good estimate for efficiency we will use the data related to time intervals gathered from the test, and organize it with the interactions to see if gaze interaction is any slower than touch or keyboard and mouse interactions. Data includes the average time participants spend on each task, and the standard deviation between all the participants. Considering there were three particular user test packages, it would also be important to separate the time by each of them, since they vary considerably.

5.4.1 Pan and Zoom

In the module for pan and zoom the participants are presented with three maps, where the first one is from the game GTA5, second being London and the last one is Manhattan. For each map there will be 2-3 tasks the participants need to complete, they will do this for all maps using interactions with touch, gaze, keyboard and mouse. During the test each task will be deliberated with the execution time, and this will be presented as measurement of efficiency. By comparing each of the maps, sorted by interaction methods, it is possible to estimate the interaction methods efficiency ratios.

Map of GTA 5

Figure 5.1 displays a graph of the map from GTA 5, displaying average time used for interaction methods with the tasks. Gaze interaction (referred as *ET* for eye tracking in the graphs) is the interaction method that have the longest completion time, at 14 seconds

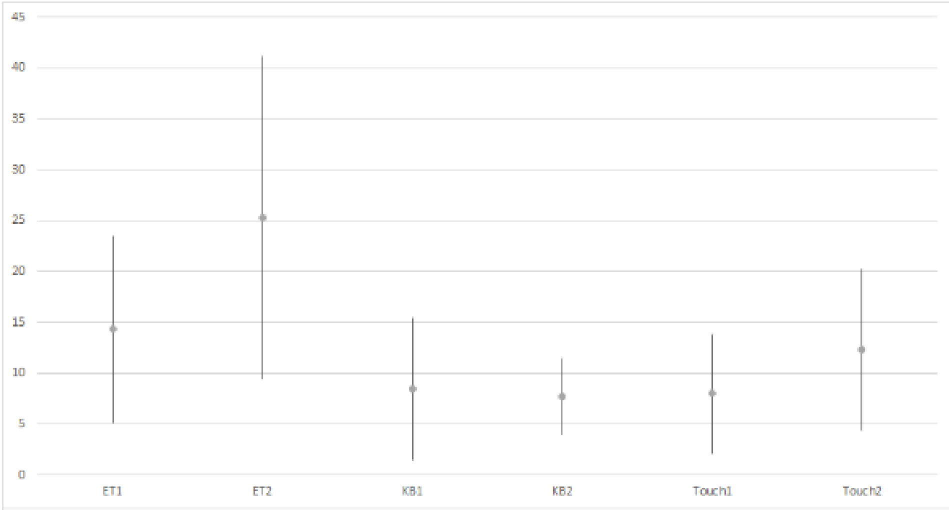


Figure 5.1: Average time used for Task 1 and 2 for the map of GTA, including standard deviation.

for Task 1, and 25 seconds for Task 2. Touch and keyboard (assigned as *KB* in the graphs) are about equal at 8 seconds, and touch on Task 2 being 12 seconds. It is important to be aware of the huge standard deviations. These are high because some participants were able to finish their tasks at half the speed of someone else. This phenomenon is especially noticeable in gaze interaction where the fastest participant finished Task 1 within 6 seconds while a passive participant used 22 seconds. Due to this immense dispersion of time used to finish tasks, the standard deviation difference are as much as 4 times between lowest deviation to the highest, and at all cases the standard deviation is close to the average value itself.

Map of London

Figure 5.2 shows the result from the map of London, this result is far beyond what we were expecting, where touch, which is a well-known interaction method, had some unusual large spikes time-wise compared to the others, and the standard deviation is quite enormous. Gaze interaction performed really well on this map, with Task 1 having an average execution time of 12 seconds, which is the shortest of all the interactions for this specific map. However, just like the previous map, the standard deviation is still quite high for all three interactions.

Map of Manhattan

The last map is of Manhattan, and the data shown in Figure 5.3 do not have any superior interaction methods in both tasks, but keyboard interaction does have a significant better performance on Task 2. It also has the second most time spent to complete Task 1. Again, the standard deviation is huge, which is a shared trait among all the maps.

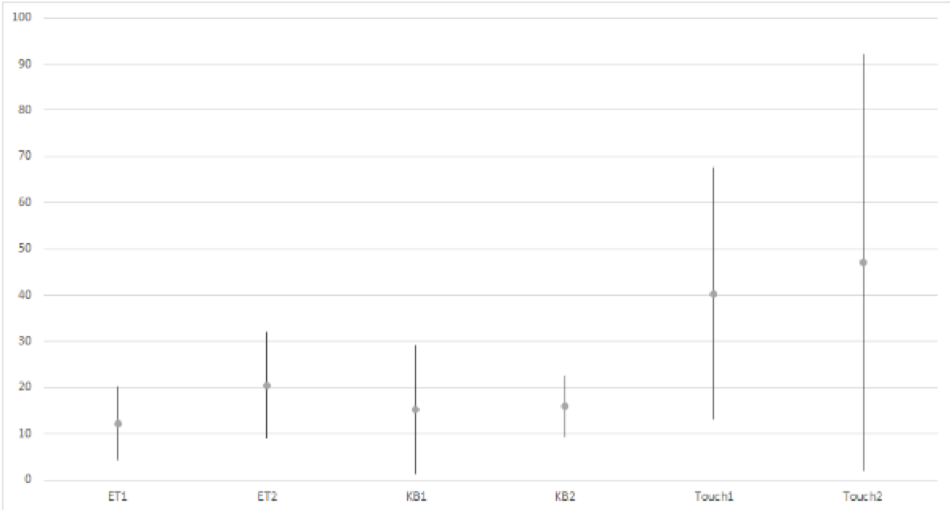


Figure 5.2: Average time used for Task 1 and 2 for map of London, including standard deviation.

Table 5.2: Time used for tasks in the module with pan and zoom, sorted by interaction.

Interaction method	Gaze interaction	Keyboard & mouse	Touch
Time used (GTA)	32.00 seconds	17.00 seconds	20.75 seconds
Time used (London)	32.75 seconds	31.25 seconds	87.33 seconds
Time used (Manhattan)	39.33 seconds	16.25 seconds	20.33 seconds
Time used (Total)	104.03 seconds	64.50 seconds	148.31 seconds

Table 5.2 showcases average number of seconds from the tasks in each map added together. This table shows that in all cases gaze interaction comes up short comparing to keyboard and mouse. In the map of GTA the time used is almost doubled, while in Manhattan the time difference is even twice as big. In total, the time difference between keyboard and gaze interaction is 38 percent. Gaze interaction did better than touch interaction in terms of total time used, mainly because of the huge spike in the map of London. Nonetheless, if we only compare the other two maps the resulting difference is 44 percent in favor of touch, meaning that touch is preferable compared to gaze interaction in these two specific cases.

5.4.2 Swipe

Second test case is swipe, the participants are required to perform a set of tasks on three image packages with interaction methods. During the test all completion times were recorded to illustrate as an estimate of how effective certain interactions are compared to the others.

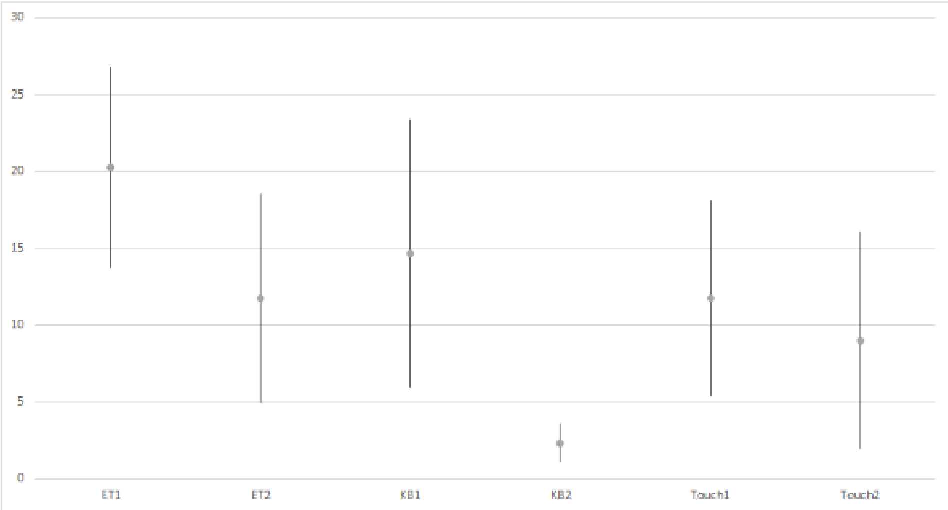


Figure 5.3: Average time used for task 1 and 2 for map of Manhattan, including standard deviation.

Table 5.3: Time used for swipe tasks, sorted into each interaction.

Interaction method	Gaze interaction	Keyboard & mouse	Touch
Time used (Packages A)	41.34 seconds	20.00 seconds	17.50 seconds
Time used (Packages B)	38.40 seconds	15.50 seconds	20.00 seconds
Time used (Packages C)	29.00 seconds	15.50 seconds	20.75 seconds
Time used (Total)	108.74 seconds	47.50 seconds	58.25 seconds

Figure 5.4 is the result of image package A, which is very similar to package C as shown in Figure 5.6. Both figures demonstrates that gaze interaction generally have a higher average time required to finish a task. Another characteristic is the standard deviation for gaze is much higher than keyboard or touch interaction. The same characteristics can be found both in package A and package C.

However, these characteristics do not hold for package B, especially in the touch tasks. Figure 5.5 shows an extremely low value for touch in Task 1, and also a low standard deviation, followed up by a huge spike in Task 2, which has the highest standard deviation.

Table 5.3 sums the average times for both tasks sorted into each interaction, it shows that gaze interaction overall have a worse result distinguished to keyboard and touch. The table showcase that gaze interaction in most cases has twice as high values, and is even higher than double when comparing against keyboard and mouse.

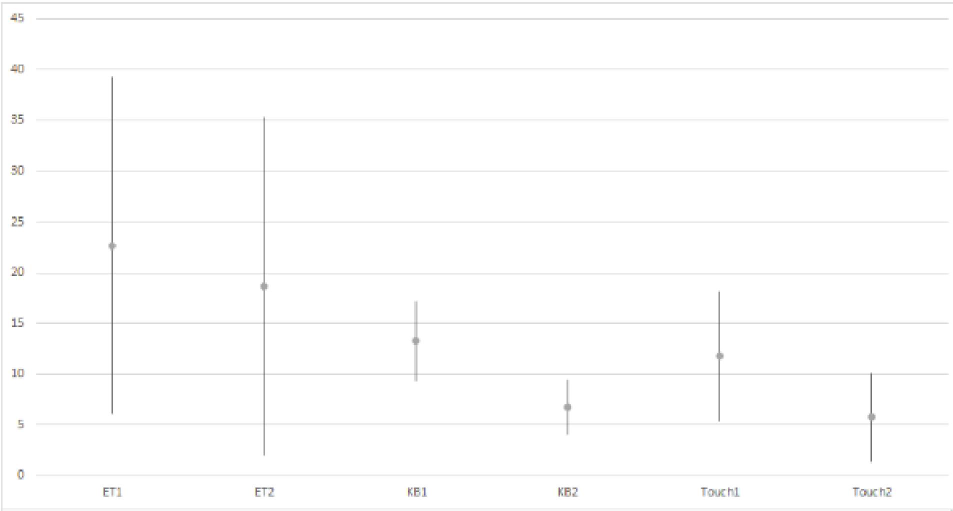


Figure 5.4: The result of swipe for image package A.

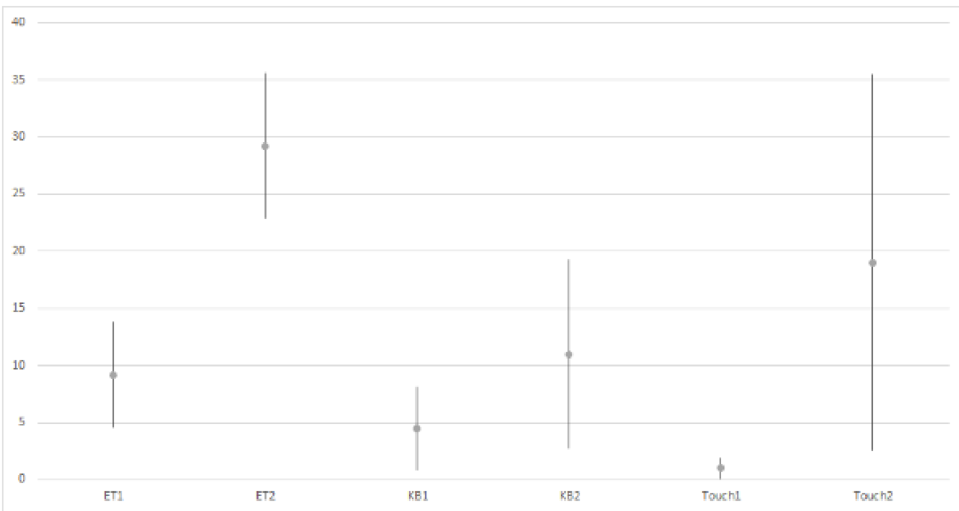


Figure 5.5: The result of swipe for image package B.

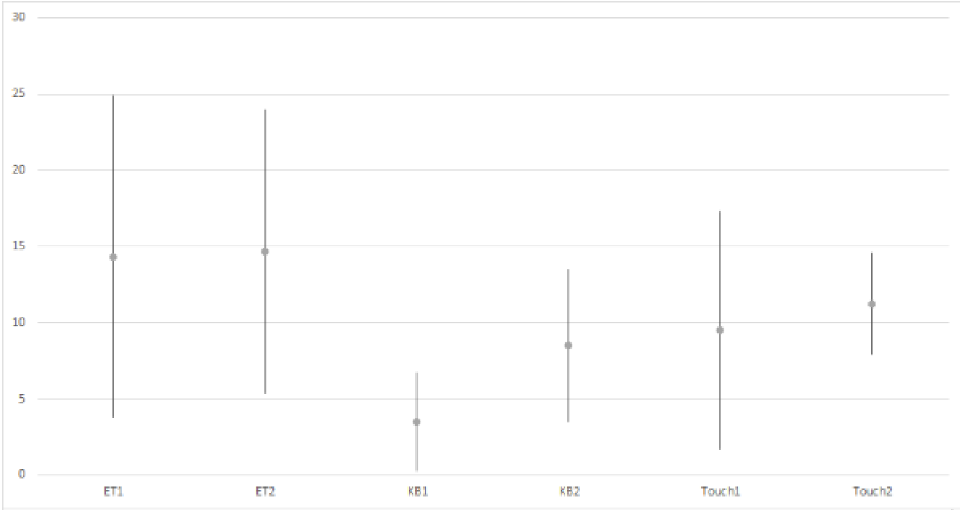


Figure 5.6: The result of swipe for image package C.

5.4.3 Scroll

The last test case involves scrolling through Wikipedia articles and reading a few selected text segments. This is to simulate a situation where a user is browsing a web page, and looks for a specific item, at the same time it gives a presentation of reading. Three articles has been preselected, with one interaction on each of them, and the participants will handle different interactions on each article.

Cookie Article

First test page chosen was an article about cookies (`https://en.wikipedia.org/wiki/Cookie`). The participants were required to scroll to the section about famous brands, where they were instructed to read the whole list of brands. There were 3 tasks in total and the end results are shown in Figure 5.7. A thing to note from this figure is that gaze interaction actually have a lower standard deviation compared to the other two interactions. The average time used on the tasks is also quite good, it even beats touch and keyboard on Task 3.

Eye Tracking and Katana

Articles about eye tracking (`https://en.wikipedia.org/wiki/Eye_tracking`) and Katana (`https://en.wikipedia.org/wiki/Katana`) are the remaining ones used. These are both similar to the previous result from Figure 5.7 explained above. Figure 5.8 shows the result of time on the eye tracking article, Task 1 and 2 (particularly ET1, ET2, KB1, KB2, Touch1 and Touch 2) had similar results on all three interactions, but

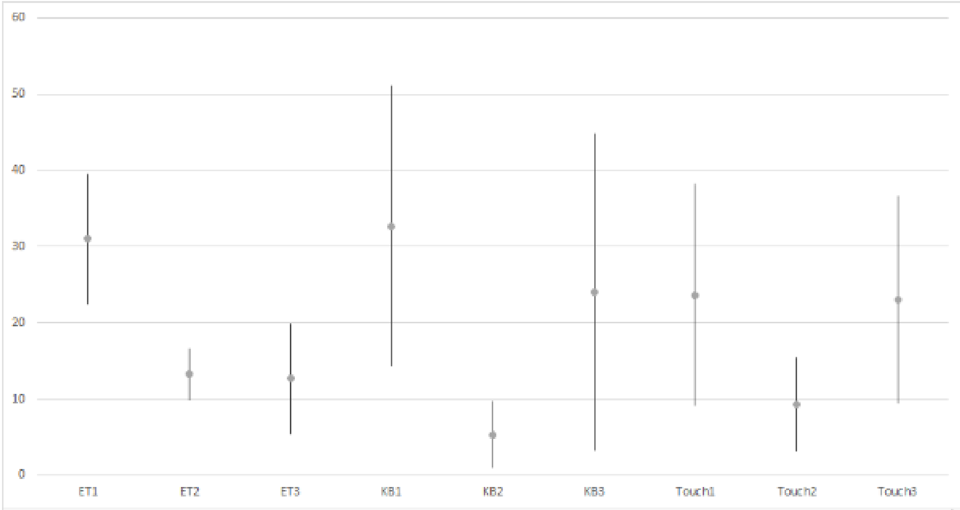


Figure 5.7: Results for the scroll module for cookies on Wikipedia

Table 5.4: Time used for all scroll tasks, sorted into each interaction

Interaction method	Gaze interaction	Keyboard and mouse	Touch
Time used (Cookie)	56.92 seconds	61.33 seconds	56.00 seconds
Time used (Eye tracking)	110.00 seconds	87.75 seconds	90.34 seconds
Time used (Katana)	66.17 seconds	56.50 seconds	40.50 seconds
Time used (Total)	233.09 seconds	210.58 seconds	186.84 seconds

Task 3 with gaze interaction has a clearly higher time consumption compared to the other two. There is also slightly similar results from the Katana article shown in Figure 5.9, where the average time for gaze interaction is close to the others. In both situation the standard deviation is much higher for gaze than the other two methods.

Table 5.4 shows the result of time added together from all tasks and sorted into interactions. In contrast to swipe, the pan and zoom module has similarities in gaze interaction, and occasionally even better than keyboard and touch.

5.5 Effectiveness

Effectiveness is measured by the amount of assists and failures during the tests, this should give a clear picture on using gaze as an interaction method, and it should be possible to interpret if it is harder to achieve goals compared to touch and keyboard. An assist will be counted if the user needs help in solving a task. For example, if the user have problems in finding map objectives during the pan and zoom module, and it required us to point at

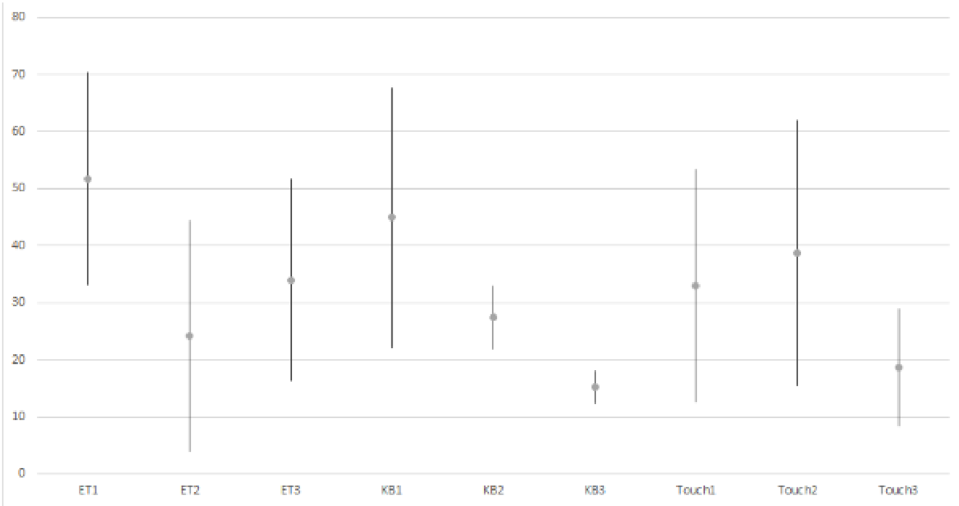


Figure 5.8: The result of scroll module on the Wikipedia page about eye tracking

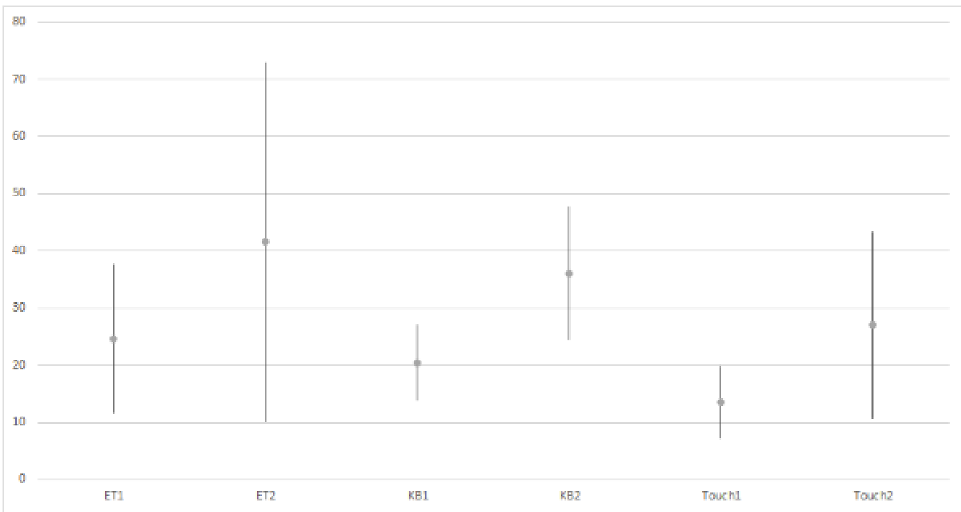


Figure 5.9: The result of scroll module on the Wikipedia page about Katana

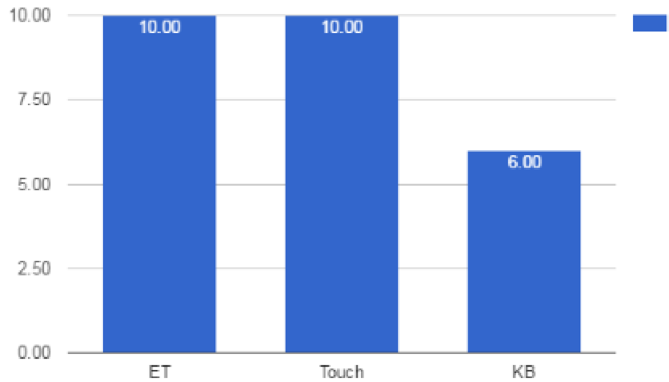


Figure 5.10: The result of effectiveness from the pan and zoom module.

the screen or give additional information to explain the location they were suppose to find, then it will be counted. Something that was unexpected is the low number of errors during the test, often there are just one or two participants who had errors in each module. There is no meaning behind displaying error results on their own, but at same time errors tends to be more severe than assists. Accordingly, errors are weighted as two assists, which makes it viable to have assists and errors as a combination into a single graph.

5.5.1 Pan and Zoom

The case that required the highest number of assist was pan and zoom, resulting in a total of 26 assists among 6 participants combined. This means that most participant who needed assists during the test had many of them, and 4 of the participants needed 5 or more. The final result numbers are shown in Figure 5.10, where gaze and touch interaction both required 10 assists, while the keyboard and mouse only had 6.

5.5.2 Swipe

Total number of assists during swipe is 9, shown in Figure 5.11, with the keyboard and mouse having the highest need for assists, with 4 assists. Gaze interaction needed 3 in total, and touch came in as the most effective with 2 assists required. An interesting point is that all 9 assists are divided between 5 participants, meaning that in average participants did not need a lot of assists in this case.

5.5.3 Scroll

Scroll module coming in lowest on number of assists required, by only needing 8 assists in total divided among 4 participants, where one participant in particular needed 4 as-

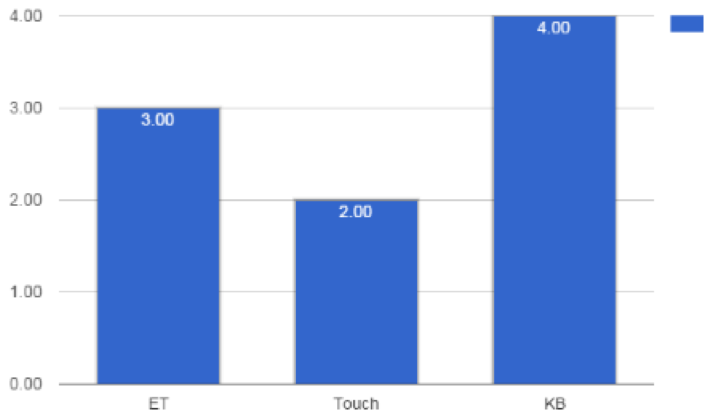


Figure 5.11: The result of effectiveness from the swipe module.

sists. Figure 5.12 shows that gaze interaction required 4 assists, while touch and keyboard needed only 2.

5.6 Satisfaction

The last data we measured during the tests were the response for the ordering of cards after each case. By forcing the participants to order the cards by their experience, it would give a measurable data in terms of *satisfaction*, making it easier to analyze. Instead of sorting by test cases, similar to what has been done in the sections about efficiency and effectiveness above, every piece of data will be sorted by interaction and which order they received. Figure 5.13 contains the average sequence of cards, during the test all participants ranked each interaction with a score of 1, 2 or 3. Giving a score of 1 is the interaction technique that the participant prefer the most, and a score of 3 is the least favored of the 3 interaction techniques.

5.6.1 Keyboard and Mouse

Keyboard and mouse was the interaction with the highest user satisfaction with an average score of 1.45 from 33 test cases, shown in the Figure 5.13. This means that keyboard and mouse was the best interaction method in about half the test cases, in only 3 out of 33 cases did participants rank keyboard and mouse last. The best keyboard and mouse results are placed in swiping, where all but 1 of the cases are ranked as second, and the rest are all ranked first. When asked why the participants preferred using keyboard and mouse over the others, most of them told that the speed and responsiveness is the reason they prefer using keyboard and mouse in an image gallery. This might be because of the software used for image gallery on most computers do not have an animation when swiping/switching

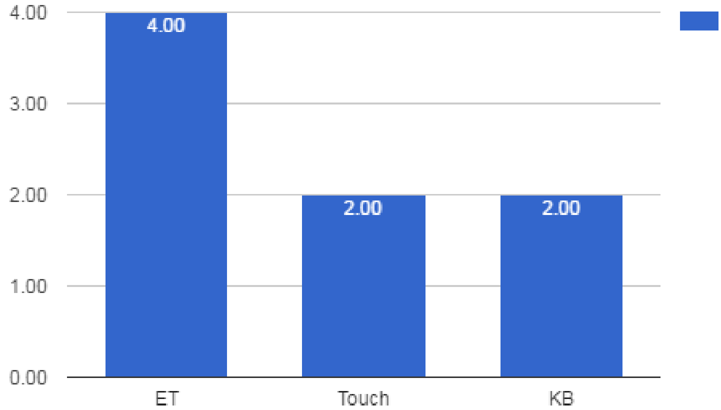


Figure 5.12: The result of effectiveness from the scroll module.

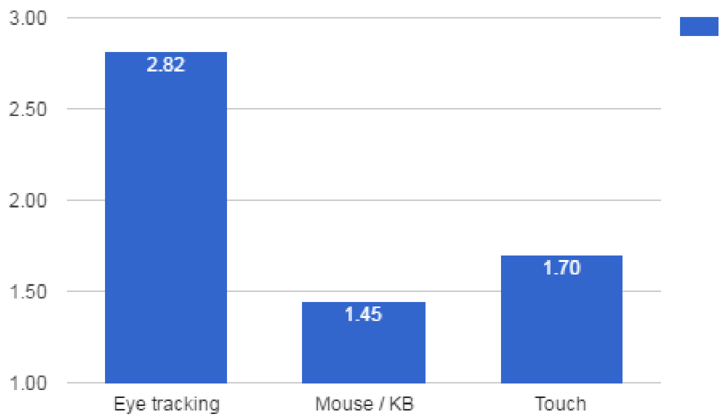


Figure 5.13: The average result from the user card rankings.

images, in contrast to both gaze and touch, where a short animation takes place. Keyboard and mouse stands less out from the other two tests, namely the modules with pan and zoom, and scroll, where the results are equal between touch, keyboard and mouse. All in all keyboard and mouse did great. A considerable advantage for interaction with mouse is the responsiveness, speed and precision in combination with the keyboard.

5.6.2 Touch

Touch interaction has been rated second in terms of user satisfaction with a average score of 1.7, shown in Figure 5.13. Although touch had an average score of 2.0 in the task with swiping, it fell behind keyboard and mouse in the other two tests. The highest advantages participants mentioned are easy to navigate, simplicity, mobile and better precision compared to gaze interaction.

5.6.3 Gaze Interaction

Gaze interaction is definitively the less favored interaction method amongst the participants. Having an average score of 2.82, meaning in most of the cases it was ranked last. Event though it might seem like gaze interaction is bad compared to the other two, most participants mentioned the lack of experience for this interaction might be the reason why they did not preferred it. Tiresome for the eyes, too slow, and bad precision are all problems that also was mentioned and need to be addressed.

5.7 Heatmaps

Using the software Tobii Studio it is possible to record gaze locations of the participants. With these recordings we can produce graphs that can show handy patterns. Heatmaps, as shown in Figure 5.14, are specifically valuable. A heatmap shows where the gazing appears most, and it does this by representing these areas with warm colors. Using this representation we can examine what and where the user is looking at, for how long and how they react when they are told to find certain elements on the screen, and where they are focusing.

All participants interacted with gaze, keyboard and mouse on the same computer, this made it possible to record gaze data for the keyboard and mouse modules too. Despite the lack of resources to set up eye tracking for the touch device, with two set of gaze recordings it is possible to discover and analyse patterns between interaction methods.

By considering keyboard and mouse as the freely interplay when performing the test tasks, for the rest of this thesis the concept of gaze movement when using keyboard and mouse is called *free movement*. Using the free movement we can now inspect the gaze

Figure 5.14: Heatmap for Pan and Zoom for participant P1 using gaze interaction

movements to see if they differ using gaze interaction. Although this result does not directly reflect any usability parameters, it can show if gaze interaction is natural to use and how good it has been implemented. A sufficient reason for believing gaze movement during keyboard and mouse is superior, is because participants do not need to focus with their gaze to control the movement, therefore this pattern must be the one participants are most comfortable with.

Good gaze interactions should not put restraints on users eye movement, but rather follow the free movement and perform the task at the same time. In a perfect implementation with gaze interaction it should consist of similar eye movement compared to the other interaction methods.

5.7.1 Pan and Zoom

Figures 5.14 and 5.15 feature heatmaps from gaze, keyboard and mouse for a single participant. Note that gaze interaction did not utilize the whole screen due to calibration issues as explained beforehand. It is unclear what consequences this would have for the final result, but the figures prove in both cases that the participant had his gaze somewhere in the middle of the screen most of the time, with a few exceptions, where there are red spots located outside the center in both cases. Gaze interaction tends to be consistently positioned in the middle compared to keyboard and mouse (Figure 5.14). Interaction with keyboard and mouse has registered multiple red areas outside the middle of the screen (Figure 5.15).

When analyzing the rest of the participants, similar patterns as mentioned above are found, where most of the eye focus is in the middle or nearby. Identical outcome is true for keyboard and mouse interaction, with all participants spending most of their time in the center, with minimal red spots outside of the borders. A few results with keyboard and

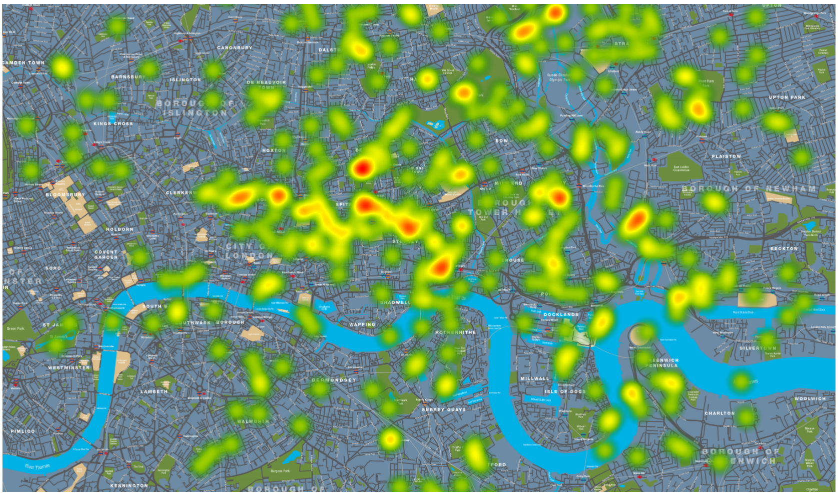


Figure 5.15: Heatmap of Pan and Zoom for participant P1 using keyboard and mouse

mouse have similar, or near identical heatmpas as gaze interaction. Figure 5.15 highlights a participant with longer time used to complete the tasks, and it clearly shows several more red spots. Another example is Figure 5.16 from participant P10. Here are the red spots also close to the center, but there are not as many because of a shorter time frame.

5.7.2 Swipe

In contrast to the module for pan and zoom, where there were many similarities in the generated heatmaps between gazing and keyboard, the swipe module had disparities. Figure 5.17 contains the result from gaze interaction, and it shows that during swipe most of the gaze is close to the right arrow, which is where the interaction to swipe to next picture is located. There is also some focus on the left side arrow, but it is clear that going backwards is not as popular. In the middle the focus is on the images, and it appears that a relatively small area is needed to get an overview of the image. A logical reasoning behind this narrow and delimited area, is the smaller size of the window in comparison to the version for keyboard interaction. Figure 5.18 shows the same participant using keyboard and mouse instead (in this specific case the participant used the mouse exclusively for navigating). Note from the figure that the participant also had most of his attention towards the interaction part of the software, which in this case was the arrow key down in the middle. With gaze interaction the eye movement on the actual image is more spread, and with close to identical color values on left and right side of the heatmap.

A general pattern is kept through for all the participants. When using gaze interaction the eye movement is heavily confined to the right portion of the window, especially on the navigation arrow. With keyboard and mouse the eye movement is evidently more spread out and a reflection of the gaze pattern is formed on the sides.

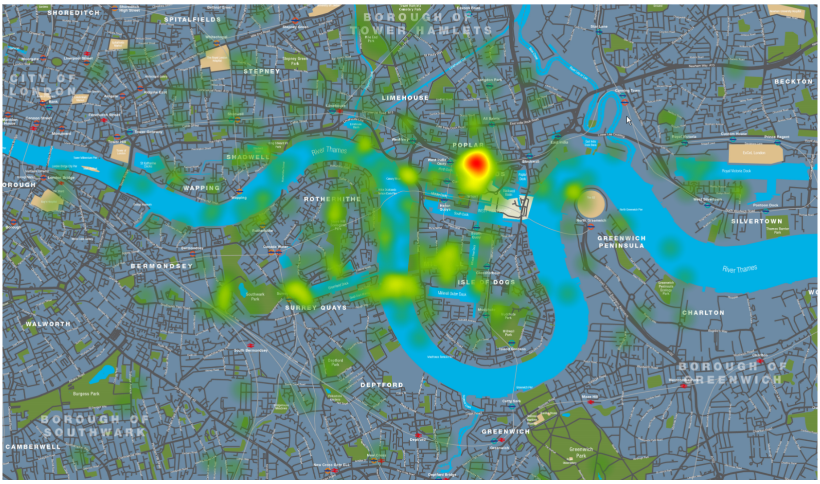


Figure 5.16: The heatmap of Pan and Zoom for participant P10 using mouse and keyboard

Figure 5.17: Heatmap of swiping for participant P10 using gaze interaction

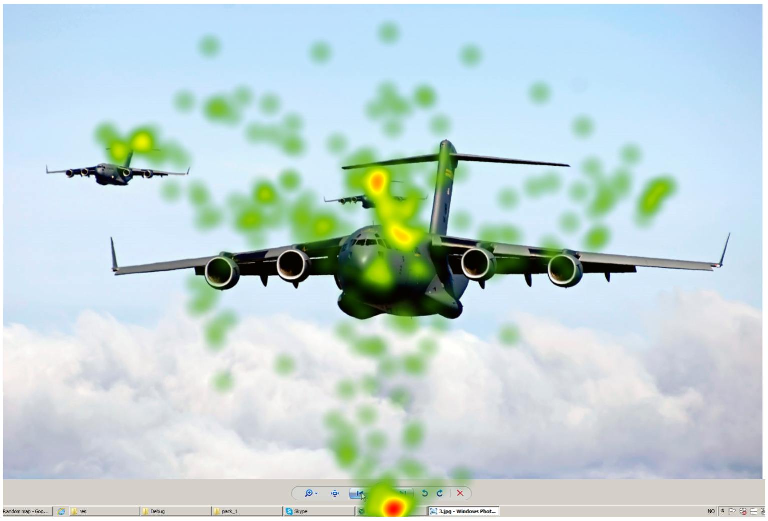


Figure 5.18: Heatmap of swiping for participant P10 using mouse and keyboard

5.7.3 Scroll

With the scroll module a smaller difference between keyboard and gaze is found. Figure 5.19 shows a heatmap for the scroll module using gaze interaction. First, notice the pattern with the colorful line of gaze recordings on the left side. It appears to coincide with the placement of Wikipedia's subsection titles for each of the articles. Second pattern is the horizontal lines on the right side, which occurs when the participants were asked to read certain parts of an article.

In Figure 5.20 the result from the same participant with keyboard and mouse is shown. Again, there are multiple eye movements placed densely confined to the left portion of the page, forming a straight line. The horizontal reading lines also remain. These patterns hold true in all of our cases, and participants form straight lines on the left no matter of scrolling interaction.

5.8 Additional Patterns

There are additional patterns among the data that is not directly linked to a specific case, but rather associated with a particular participant. These patterns varies from time the participant used to complete tasks, and the eye movement they had during tasks that might reflect different aspects for the module tested.

5.8.1 Alternating Completion Times

During testing we noticed certain participants used abnormally longer time on some tasks, regardless of interaction method. For example in the Pan and Zoom module, where a participant used more than 15 seconds on a single task, he or she will use on average more

Figure 5.19: The heatmap of scroll for participant 07 using gaze interaction

Figure 5.20: The heatmap of scroll for participant 07 using mouse and keyboard

than 15 seconds in 5 out of 6 tasks. On the other hand, participants with faster average completion times (performing in 15 seconds or lower) on two out of three interaction methods, shows that the last interaction will always be relatively quick finished.

There are also a couple of participants with extremely fast task completion times, where most tasks are completed in under 15 seconds, with a few exceptions where they either did not understand the task, or started off incorrectly. However, this pattern does not hold true in the module for swiping, where faster participants are rather random, and it can go either way. A few participants are fast on Task 1, but not on the next one using equal interaction method. Differences between a fast and slow completion times can be as much as 500% in extreme cases. Certain participants who were fast in the module for pan and zoom, also have the lowest average times with scrolling. They will often use only 10 seconds to complete tasks that took others 40 seconds or more.

Chapter 6

Discussion

This section will discuss and analyze the results gathered from the usability tests. It will go into aspects around usability for gaze interaction, what influences the issues had, factors that might have affected the tests, and which usability metrics are influenced most.

6.1 Data Sets

From the usability tests there were a total of 11 participants who attended. The participants are separated into three groups, and this makes each test case consisting of four sets of data. To make a reasonable presentation of every possible outcome from these tests, 11 users is a minimalist number, and the tests could ideally include up to twice as many.

An unexpected problem happened when one of the participant showed up when the touch device was not available at the lab, and coincidentally this participant was placed in the group of 3 participants. This accident can possibly have an effect for some test cases. From the map of London (pan and zoom), and image package B (swipe module) there will be only 2 sets of data with touch gesture, which might be the reason for Figure 5.2, where touch interaction has an enormous spike. An adjustment was made in the participants task order to let the scroll module have three sets of data.

Overall, the data would be more precise if some extra participants attended, especially for Figure 5.2, which seems like an outlier. Rest of the data sets provide sufficient information for how good gaze interaction is compared to the other interactions.

6.2 Pan and Zoom

6.2.1 Efficiency

Efficiency is a major factor for analyzing usability tests. How long time it takes to complete a task can say a lot when comparing against other test cases. With the module for pan and zoom, the efficiency rate was as expected, and in terms of average time spent the results for gaze interaction was worse compared to touch, keyboard and mouse interaction, as shown in Table 5.2. It can be noted that gaze interaction did achieve similar times compared to the other interactions in a few cases, such as the map of London where there is only a second in difference between gaze and keyboard interaction.

Factors

Experience for the different interaction methods appears as the main reason for the various time intervals. Most of the participants have limited experience with gaze interaction from before, and none have used it in similar situations. On the other hand, they are all familiar with map navigation on both touch devices, and computers with keyboard and mouse. This creates an enormous advantage for touch, keyboard and mouse, because the user knows the response for the actions they perform.

A phenomenon noticed is some participants move their entire body in order to zoom, as explained in Section 3.1.3. This is highly unpractical, because when the whole body is moving, they will get out of bounds from the eye tracker's range of effectiveness. When losing the connection it can result in unforeseen responses, and the navigation may not respond in the way they were expecting. Problems like this can be resolved by using gaze interaction for an extended period of time, which also can help to improve the efficiency greatly.

A second factor is the value of speed for panning in the pan and zoom module. There has not been used any specific guide for panning with the human eyes. The speed value has been set with what is the most pleasing from our own eyes. However, it is not static, but a value multiplied with the distance from center of the screen is used. This turns out to be a problem since some participants feel it as too slow, and that they could complete tasks faster by increasing it. Some participants also stated the value for being too high, and that they would lose tracking of moving objects. One of the participants reported feelings of nausea and being uncomfortable. This problem is not as significant for touch, keyboard and mouse as in both cases the participants could adjust the speed by their hand movements. Both of these interactions can be adjusted even more to fit the individual needs, and they have a solid foundation of testing behind. We believe the factor of speed value has affected the results in some ways that is difficult to quantify at this stage. With additional testing and the possibility to let users adjust the panning speed, this should not be a problem.

Lastly is the personal factor, as mentioned previously in Section 5.8.1. It seems like certain participants complete tasks faster than others. At first we thought they were simply faster to adapt and learn, but the fact that the speed of finishing tasks is kept no matter what interaction method, suggests it has nothing to do with gaze interaction. This factor did not affect the average results, but it resulted in large gaps for the standard deviation, which is not preferred.

Standard Deviation

For some participants the factors mentioned above in Section 6.2.1 might have affected them to perform slower, but there are participants who adjusted quickly. It seems like experience and speed did not inhibited them. This leads up to another aspect of the results, which is a high standard deviation. From all the tasks there was an average standard deviation of 7.9 seconds per task, and the average time per task is 15.61 seconds. Resulting in a optimized situation where a potential user is expected to spend 7.71 seconds per task, which is enough time to compete with the other two interaction methods.

Another factor to consider the is grade of luckiness, as we believe some participants solved the task faster simply because they were close to the target area when the test began. During all three interactions the participants were allowed to look around, and get familiar with the map. By starting at random positions it will naturally result in cases where some started nearby the target area, giving them a fast completion time, and longer for those having to pan further.

6.2.2 Effectiveness

An important usability metric mentioned in the preceding chapter is *effectiveness*. In an ideal world when navigating through a map looking for specific locations, there will be helping tools such as a search field or street view, but since no tools were provided for the test cases there were multiple assists and errors for all interactions. From the result shown in Figure 5.10, it can be nuanced that gaze interaction has an equal score as touch, with keyboard and mouse being the better of the three.

Factors

We believe the main factor that helped improving effectiveness is the free movement of the eyes being similar in all three interactions. This is highlighted in Figure 5.16 where gaze movement is comparable to the others, and it normally occurs when a user navigates a map with mouse and keyboard using their gaze movements where the mouse is positioned. A difference is with gaze interaction, where users will skip clicking and dragging with the mouse, because the system will do it automatically. Number of assists and errors were similar in both test cases. Errors typically includes moving in wrong direction, overlooked a location, or giving a wrong answer from a test.

6.3 Swipe

6.3.1 Efficiency

In 5 out of 6 tasks did gaze interaction use more time in average, the only exception being in package B for Task 2, which as explained before in Figure 5.5 might be an exception. Comparing the average time of gaze interactions to keyboard and mouse from the Table 5.3, gaze interaction required about twice as much time for completing the same task. If the exception from touch in Task 2 is ignored, it can be pointed that touch is superior compared to gaze interaction in terms of efficiency.

Factors

One of the key factors is again the experience for the same reason as mentioned above for Pan and Zoom module, and this will be a following trend through all modules. Because it is already mentioned and explained, it will not be commented again in this section, neither in the next module for scrolling.

Another factor is the time of delay for registering events. Due to the design of swipe, a user have to look at a zone for a certain period of time in order to confirm the action. Traditionally, eye interaction always had the problem about how to send out information as described in Section 2.3, and in our solution we chose the approach of using dwell time. This resulted in participants wasting time at the edges of the image, and if the participant goes out of this zone, he or she will have to restart the process. In contrast for keyboard and mouse, where the interaction reacts instantly, and it is possible to swipe through all the images in matter of seconds, while keeping a reasonable idea about what the image was presenting. Although touch devices require a finger movement, it is still faster than gaze interaction because the eye is available to analyze the image while the finger is handling the swiping.

We believe the main reason for this module lacking in efficiency is the combination of tools for hardware and software. In Section 2.6 it was mentioned that the hardware is not adapted to our screen size, and in Section 2.7 it is noted how important calibration is for gaze interaction. In our specific case, when the calibration is offset by a few millimeters it might be enough to fail the zone check multiple times. This happened in a few times during the tests, and many participants had to try swiping various times in order to accomplish it because of calibration failures. This phenomenon also had the side effect for some participant to feel uncomfortable, because they had to try and focus on a single point several times. It is easy to spot these reactions by looking at the heatmaps shown in figures 5.17 and 5.18 where the gaze movement using gaze interaction does not match the one using keyboard and mouse interaction. Keyboard and mouse is what is referred to as the free gaze movement to complete the task.

During the implementation we tried to imitate a similar look and feel as the photo galleries from smart phones, meaning an animation will start the moment dwell time is completed, unlike keyboard and mouse where the photo swaps instantly. This had some delays for our solution because the participants had to refocus after every swipe, while on a touch device they can look around freely on the image, while the finger does the rest.

Standard Deviation

The standard deviation for swipe is different from Pan and Zoom, where all three interactions had huge standard deviation, in this case it seems like touch, keyboard and mouse have a smaller standard deviation comparing to gaze interaction. We conclude the main reason for this is caused by the combination of software and hardware factors. For some participants the calibration was spot on, and swiping was easy to perform, while for others it was rather challenging. Again, this is confirmed by participants during the tests, where most stated that swiping was an easy task, but others had to try several times to succeed one single swipe action.

6.3.2 Effectiveness

Figure 5.11 shows that the effectiveness between gaze interaction, touch, keyboard and mouse is more or less the same. This is because the tasks are about finding a specific image. All participants have done similar tasks multiple times before, but the only difference is the kind of interaction method they use, and gaze interaction is very intuitive and easy to learn. Final results ended in a few errors that are caused by participants swiping too far. They did not see the image or gave the wrong answer.

Although few errors and assists were needed to complete the tasks, it is safe to assume some participants, namely those who had poor calibrations, probably had to perform certain actions multiple times for the eye tracker to spot it properly. Unfortunately we did not record these incidents, either because we were not able to record when the participants performed the tasks, or the eye tracker lost connection. Multiple factors might have caused the action to not be triggered, but we could not quantify each factor's influence. Overall, since the same problem did not exist in touch or keyboard and mouse, it is reasonable to state effectiveness from gaze interaction is worse than the other two interaction methods.

6.4 Scroll

6.4.1 Efficiency

Comparing to the other two prototypes, the scrolling module showed great efficiency. As shown in Table 5.4 gaze interaction generally had an average time similar to the others. The difference is about 10 percent comparing to keyboard and mouse, which is something that most will consider a good interaction. Note that scrolling is never placed in the tests where touch had an unusual spike, this is probably due to our adaption of generating more data sets.

Factors

An essential reason why efficiency is similar between the test cases might be caused by the tasks given. The tasks involved scrolling to a specific point and read a certain amount of text. This means that most of the time the participant is busy reading, and the reading speed is almost identical no matter what interaction method.

Another reason is the free movement of the user's eyes fit perfectly with gaze interaction. In normal situations the eyes will move towards the lower portion of the screen when reading, and with this implementation there is no need for any hand gestures. From a theoretical standpoint, gaze interaction could perform better than the other two interactions, but this was not the case, probably due to unfamiliar motions when reading, or the scrolling speed was not fast enough.

These circumstances leads to the scrolling speed as the main reason why gaze interaction is less efficient. The scrolling speed is a constant value, and is incremented when the gaze progress to a certain portion of the screen. With 300 pixels from the center, the value is doubled and this will escalate the longer gaze is positioned in the lowermost zone.

6.4.2 Effectiveness

There are few errors registered that relates to a specific interaction method. The tasks mostly involve scrolling to a specific position, where there are some text they need to read, all of which the participants have done before, and the effectiveness is higher because of this.

6.5 User Satisfaction

As Figure 5.13 shows, it is clear that in most situations participants will prefer using touch, keyboard and mouse instead of gaze interaction. In nearly every module was gaze interaction placed last, with a few situations as second, and a single test where it was placed first. This was expected on beforehand since gaze interaction is in an exploratory stage without the same qualities as the other test cases. It therefore resulted in worse user satisfaction.

6.5.1 Factors

The first factor that affected user satisfaction is the lack of personal adjustments. Values for scrolling speed was set to constant, and this may only fit a small group of people. This was proven when most participants either felt it as too slow or fast. If users can adjust the speed and other factors by themselves, or a profile is set when the user is calibrating, it would help improving user satisfaction.

Second factor is the design of the implementation, where gaze interaction should be different from the other two interactions in terms of functionality and user interface. This was not considered in detail before the development, and the implementation imitated already established software. This is one of the reasons why some of the modules are similar with existing applications using touch or keyboard. With relations to the user interface, improvements could be added to increase the aesthetics, which include better feedback and faster response times, but this was not prioritized since our aim is to primarily test the functionality. To conclude, an ideal gaze interaction module should be designed from scratch with gaze interaction in mind. This will include interaction methods implemented, esthetics, optimization, and genuine feedback.

6.6 Overall Assessment

6.6.1 Efficiency

Overall we observe gaze interaction to be a slower alternative to touch, keyboard and mouse, with variations from module to module. Scrolling seem to be the most efficient module using gaze interaction, while the modules for pan and zoom, and swipe are a little worse. The main reason for the decrease in efficiency tends to be a combination between the experience, software and personal adjustments. A specific factor for effectiveness is hard to quantify, but considering how many users consistently mentioning reasons for why they believe tasks were slower to complete, it has to be counted as a factor with greater impact.

With the calibration accuracy being drastically changing from participant to participant, there were varying test results. The problem was mostly prevalent in the module for swiping, because of the implementation whit dwell time.

Personal adjustments is a factor that is shared among all the modules. The more tests completed, the more clear it became; in order for gaze interaction to work with everyone, it should be possible to adjust the speed of scrolling, also panning and zooming should have improved synergy.

Gaze interaction can get on the same level compared to efficiency as touch, keyboard and mouse. From the results we can denote the average times for gaze interactions are low enough to be competitive, but considering this will be targeting a wide user group, consisting of all ages and grades of experiences, it will require more improvements.

6.6.2 Effectiveness

In contrast to efficiency, the effectiveness of gaze interaction is proven to be similar as touch, keyboard and mouse. The results were worsened for pan and zoom, but improved with scrolling. Swipe module is special in this case, since it has a gap with effectiveness. Some actions are not triggered when intended, and it is believed the same problem exists in a limited extent for the other modules as well.

We believe one of the factors for effectiveness is the simplicity using the interaction. This is in our case how fast a user can learn to use the interaction method. By contemplating if gaze interaction is easy enough to learn and use, it should yield similar results compared to touch, keyboard and mouse. From the observations we could clearly see tendencies of progress. It did not take many tries for the users until they figured how to use gaze interaction, although most of them were a bit clunky to begin with.

All three interactions have similar effectiveness when comparing number of assists and errors. It is worthwhile to mention the number of assists in module for pan and zoom is higher than the other modules. This module used maps for navigation purposes, and most of the participants were not familiar with them. Everyone had not been to Manhattan or London, and only some played GTA. Even if they knew all of the maps/locations, it still would require a lot to remember all names that were asked.

Another factor for effectiveness is the preciseness of actions that are recorded by the eye tracker. As explained in Section 6.3.2, although there were few errors and assists needed to complete, the participants might had to perform some actions during testing multiple times due to the actions not being registered. This is a problem that might be improved with better hardware and software. However, even with a perfect system the problem might still persist. The root of the problem lies in the human eye's lack of capability to send out correct information.

6.6.3 Heatmap

It is hard to appraise exactly what impact similarities between gaze movement when using gaze interaction, and free eye movement had on the usability aspect. We believe the similarity is a measurement on how well the implementation is conducted, as mentioned in Section 5.7. In our case, we noted that swiping overall was worse than panning and scrolling, particularly in efficiency. In addition to what is already mentioned in Section 6.3.1, a new implementation that have gaze movement similar to patterns from Figure 5.18 will have a better results, both in effectiveness and efficiency.

6.7 Module 4

Module 4 was implemented in time, and was meant as a second iteration for our usability testing. There was not enough time and participants to set up a full test, instead the three previous participants and a new participant attended for shorter test experience. The main goal in this case is to compare Module 4 against the preceding modules 1, 2 and 3 to see if there are any improvements or drawbacks by putting all of them into a single application.

All the test participants were given a set of tasks, these are specifically designed to guide the participant through the changes that have been made. In most of the cases they would comment to have notice the changes themselves, although they often could not point out exactly the difference. Both during and after the tests all participants were asked how they liked it and how it was compared to the first iteration of testing.

6.7.1 Results

There are no quantitative data produced by this result. There were no recording of task times, and the goal was to let the participant thoroughly experience the new improvements, rather than measuring the efficiency of using gaze interaction. Therefore, all data generated from this test is qualitative. It is gathered from of conversations and observations with the participants during and after the test.

First and foremost, the results shows that the system works, the property for focusing window frames was not complicated to use. The biggest improvement was in the scroll module, where almost all users experienced a noticeable improvement. This is probably due to the dynamical speed increase compared to the static speed increase, which was the case in Module 3. For the other modules the improvements were satisfying, but the user experience was not affected to the same extent.

Event though the data samples could be larger in order to make a conclusive statement on Module 4, it seems like the dynamic value of speed in scroll was well received, and proved to be a better implementation than a static value. A possible future improvement is to modify the panning module to also have dynamical panning speed, which increments by time instead of distance from center.

During the implementation there are some theories about shrinking the window will be a problem, but the overall reaction is positive, meaning that shrinking down some portion of the window played a tinier role than expected.

6.8 Summary

Result of this usability test is directly linked back to Research Question 4:

"How do users assess the usability of gaze interactions for the use situations, compared to similar interactions using touch, mouse and keyboard?"

By analyzing the data gathered from the tests, we were able to piece together an answer. The overall usability of gaze interaction can vary a lot from user to user, and depends on factors that tend to include experience: How experienced is the user with the task at hand, or how experienced is the user with gaze interaction in general. Other factors involve design of the software interaction. Firstly, make the interaction so that the gaze movement fits the natural movement for the task, this also helps to make the interaction intuitive, thus improving learnability. Secondly, there should be a way to adjust values for the user in control, this should result in improved user satisfaction. Calibration has shown to be an essential factor for the success rate of gaze interaction.

In most cases gaze interaction will be able to complete the task, but often slower compared to touch, mouse and keyboard, and the overall user satisfaction is worse, although the user satisfaction might have been caused by the design of the software itself. A more definitive answer could have been given with extensive testing and continued development of Module 4.

Chapter 7

Conclusion

7.1 Research Questions

With the aim to explore possibilities of gaze interaction to replace the current interaction methods on multi-touch handheld devices, a set of research questions were presented to correspond to a part of the thesis.

RQ1: Based on available literature, what functionalities does gaze interaction need in order to replace current interaction techniques on multi-touch handheld devices?

RQ2: Using basic functionality sets, what are the most promising use situations that will cover these functionalities?

Both of these research questions were answered by the literature study in Chapter 2. The goal of Research Question 1 is to identify what functionalities gaze interaction needs to fulfill in order to replace touch. To develop this set of functionalities we analysed the basic touch gestures for iOS, and used them to develop basic gestures for gaze interaction. The focus was on the four main interaction techniques - drag, pinch, tap and flick. With these gestures in mind, three use situations were suggested to be implemented.

The first one is a module for navigating a map where the gestures dragging and pinching will be covered. This use situation will highlight the navigational capabilities of gaze interaction, which we assumed to be a strength for this type of interaction.

Second module is an image gallery that can be both associated to flicking and tapping on multi-touch devices. In contrast to map navigation, the image gallery will instead showcase the weaknesses of gaze interaction, namely sending messages back to the computer using the gaze.

Third use situation was intended as web browsing, where dragging and tapping could be implemented. This use situation is focused on a everyday tasks that is vital on today's computers and touch devices. A combination of these use situations cover the basic functionalities required for gaze interaction.

RQ3: What are the major technical challenges implementing these gaze interactions with current eye tracking technology?

The third research question is answered in Chapter 3 covering the prototypes, where the main goal is to identify potential technical challenges when implementing gaze interaction. During the project it was discovered that successful implementations of gaze interaction faces three challenges.

The first challenge is the hardware, in our case the eye tracking device. Limitations on head space will put restraint on how long time gaze interactions can be applied consecutively until the user is exhausted. Another hardware challenge, mentioned earlier in Section 2.6, puts 17 inches as the optimal size for gaze interaction, whereas today's displays are often larger, therefore eye tracker devices should be able to support larger screens.

Second challenge comes from the software, the limited support from the C# language and lack of libraries made it harder to implement gaze interaction properly.

The last challenge is a combination of hardware and software, it is the part considering calibration. Especially when implementing the scroll module, it was difficult to get click actions to work because the calibration is frequently a few pixels off, which is why this function was not tested in later stages. Calibration might very well be the make or break factor for gaze interaction in the future, so far there are no perfect calibration services, but it should be improved with better hardware and techniques. In the future it will probably not be a problem.

RQ4: How do users assess the usability of gaze interactions for the use situations mentioned in Research Question 1, compared to similar interactions using touch, mouse and keyboard?

Research Question 4 assess the usability of the three modules we created by comparing it with similar software using touch, keyboard and mouse instead. An experiment was designed with a total of 11 participants, where each of them performed a 45 minutes test to evaluate the usability of gaze interaction. By collecting average times a participant used finishing tasks, a comparison of effectiveness was created, and compared interaction methods against each other.

Results express gaze interaction as less effective in modules containing swipe, pan and zoom, but performs similar as other interactions with scrolling. Other observations are the huge standard deviations in most of the cases, which leads us to believe that there are some factors that make it easier for certain participants to interact using gaze than others.

Experience, calibration and personal adjustment are the three main factors. With extensive usage, better applications, improved calibration, and the possibility to adjust settings for gaze actions, we believe the efficiency will improve greatly.

Effectiveness on the other hand, was similar between gaze and the other two interactions. If gaze interaction was worse than the others, it could suggest that gaze interaction, or the general implementation had some fundamental flaws. This can be reflected from the numbers of assists and errors, which mostly occurred when the participants did not understand the interaction, or the interaction performed differently from what they were expecting.

User satisfaction could be from a multitude of factors, and it can be connected with both functionality and design of the user interface. In this thesis we could not go into depth to identify which factors played the major role. However, results proposed the general feeling for gaze interaction is worse than the other methods with a noticeable margin. The metrics of usability in efficiency, effectiveness and user satisfaction show that gaze interaction from the implementations is still equivalent with touch, keyboard and mouse. It shows great potential and all the factors that might have affected gaze interaction could be fixed and enhanced.

RQ5: What role might gaze interaction play in the coming development of computers and multi-touch handheld devices?

Final research question is about how gaze interaction will be in the future on both computers and handheld devices. In order for gaze interaction to success, it would require eye tracking devices to be widely available. This can only be achieved by cheaper costs and a higher value of use, and it will require more research to solve how gaze interaction can be best utilized. There are no handheld devices today that have eye tracking integrated, but companies like Eye Tribe have started to sell functional and tiny eye tracking devices for this purpose. Another potential field for widely spreading of eye tracking could be with virtual reality (VR). Ealier this year, the long awaited Oculus Rift was released, it has the capability to be upgraded with eye tracking [30]. Other VR headsets have even gone further and integrated eye tracking in the product [10]. With the expansion of eye tracking devices, gaze interaction would be more relevant. Currently most gaze interaction is supplementary to mouse and keyboard, but as our results suggest, gaze interaction is competitive to touch, keyboard and mouse, and in the future a interface for gaze could positively be developed.

7.2 General Conclusion

There are both upsides and downsides for gaze interaction on multi-touch handheld devices. The good news is that in our usability tests we have proven that gaze interaction can be used as a side tool, and even replace general interactions in the best cases. In the worst cases it is not far behind neither. This is hugely dependent on the task, how the application is designed, and the hardware and software challenges concerning it. In an ideal implementation the gaze interaction could be better or equal to touch interaction, this will at least be true for simple tasks. Even if gaze interaction do not replace touch, it could still be a good supplement for it, allowing the user to manipulate multiple objects on the screen at the same time.

The bad news is the low amount of eye tracking devices for commercial use. Gaze interaction will have a difficult chance on evolving without enough initiative and investment from the industry, and since eye trackers are not widely available today, it weakens the potential. There are also problems regarding the size of eye trackers, promising calibration improvements, costs and software support. Assuming these factors will be fixed in the future, it will provide a opportunity to widespread the use of gaze interaction. If the development of virtual reality continues to grow, there will be good chances for gaze interaction becoming a notorious term.

7.3 Future Development

After all the tests and improvements with the implementation we were able to get a better idea of what gaze interaction can do and how users can utilize it properly. These ideas should result in a better experience. We did not have the resource, nor the time during the projects to apply these notions, but we feel they need to be documented for a future advancement.

7.3.1 Adjustments

First addition to consider is a personal adjustment that have been mentioned in the previous section. With personal adjustments, we mean preferences that adjust the system to fit each user.

This could have been implemented using various settings which can be accessed with an interface, here a panel will present values the user can edit to fit the software better. There should be a preview pane that visualise the updated settings when a value is changed. It could be a smaller window where the user could test the new options in effect, together with a confirmation dialog after the values have been entered. The user could have 10-15 seconds to test the new settings and then click accept, or the changes would be reverted.

Another way to implement personal adjustments will be do initiate a dialog automatically when the application using gaze interaction has started. It would be unintuitive in the long run for users to find the settings panel, and complete the changes. This is rather common in the gaming world today, where settings for brightness and controls can be adjusted upon startup. An example for such dialog is showed in Figure 7.1 where the user is asked to adjust the brightness to a certain level. The same could have been done for gaze interaction, for example with a circle moving with a static speed from left to right, and the user is tasked to set a value to *catch it* using the gaze. This way we can find a value that is tested and proven to work for each user.

The ultimate way is to let the system analyze the user's behaviour and then set adjustments accordingly. The system will register all gaze movements and adapt the the settings if the user is constantly changing his direction uncontrollably. This is an ideal situation, but is probably not as realistic and requires lots of work.

7.3.2 Extra Modules

We made three modules using gaze interaction, although the interaction from these are widely used in similar ways by other software, we believe they do not designate the whole

Figure 7.1: Brightness adjustments form the game Dark Souls 3

picture, and there will be certain tasks where gaze interaction fits best. We suggest to implement gaze interaction into different environments to perform tasks and observe the results. For example a module for navigating through a folder, or a module using gaze interaction to browse the Internet. It is hard to predict what kind of modules will perform better with gaze interaction or worse, but with testing it is possible to find tasks that gaze interaction are desirable with.

7.3.3 Improved Interactions

The designing of the interaction used in this thesis is a combination of what has been done in previous work and our own experiences. A future possibility might consist of designing new set of interactions that works in new ways. In which case we strongly suggest to make the new implementation use free gaze movements recorded as basis, and then design gaze interaction that approximate these gaze movements. We believe this is the best approach for improving effectiveness, efficiency and user satisfaction.

7.3.4 Secondary Supplement

From our results we realise that gaze interaction does not fit well as a standalone interaction method at current state, caused by the lack of possibilities to pass information. This is explained in swipe module in Section 6.3.1 where the efficiency of gaze interaction is worse than the other two interactions. This problem does not really matter when gaze interaction is used as navigation tool in a similar manner as in the modules with scroll or pan and zoom. When gaze interaction is used with navigation, it might even be faster than touch, keyboard and mouse, and frees up both hands for the user to do other tasks. This lead us to believe using gaze interaction as a standalone interaction for devices is undesirable, instead by combining it with other interaction methods it can harvest the best from both worlds.

There is a new notebook recently released to the market by MSI called *GT72S* [21] that have an eye tracker integrated, and it is primarily a gaming laptop. Although the eye tracker is small, it is just as powerful as the one used in this project. The notebook allows playing modern games using gaze interaction. A recent game using such interaction methods is *The Division* [34], where it uses gazing as a second mouse on the screen which can aim, and hover over objects. The gaze is also used to extend the vision. For instance, when the player looks in the edges, the camera will broaden this area. These functions are traditionally tasks performed by the mouse. With an eye tracker on a notebook there is an ocean of possibilities. Eye tracking on consumer products might not be a huge success yet, but with further improvements it could absolutely form a new era of interactions.

Another industry adaptation for eye tracking is the *Tobii EyeX* [33], that uses eye tracking to improve interaction for working with computers. It has handy functions for window recognition, and it can scroll in another window without having the mouse placed inside of it, gazing at it is enough. EyeX provides a feature called *Mouse Warp*, which will move the mouse pointer to where the user is gazing at. There are several more feature like this included with the EyeX software, and many of them can become beneficial, but it may take some time to get used to it.

A future project could involve using modern devices, such as the *MSI GT72S*, *Eye tribe tracker pro*, or *EyeX* to verify if gaze interaction combined with touch, keyboard and mouse does improve efficiency, effectiveness and user satisfaction.

Bibliography

[1] Apple. Interactivity and feedback. URL `https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/InteractivityInput.html#//apple_ref/doc/uid/TP40006556-CH55-SW1`. Updated: 2016-03-21.

[2] Avadirect. 5036t-t black mid-tower platform. URL `http://www.avadirect.com\/5036T-T-Black-Mid-Tower-Platform-LGA1366-Intel-X58\-DDR3-1333-24GB-6-SATA-RAID-5-HS-4-HDA-GbLAN-2-465W-Low\-Noise-PSU/Product/1851786`.

[3] N. Bee and E. André. Writing with your eye: A dwell time free writing system adapted to the nature of human eye gaze. In *Perception in Multimodal Dialogue Systems*, pages 111–122. Springer, 2008.

[4] I. E. Commission. iso/iec 25062. URL `https://www.iso.org/obp/ui/#iso:std:iso-iec:25062:ed-1:v2:en`.

[5] A. De Luca, R. Weiss, and H. Drewes. Evaluation of eye-gaze interaction methods for security enhanced pin-entry. In *Proceedings of the 19th australasian conference on computer-human interaction: Entertaining user interfaces*, pages 199–202. ACM, 2007.

[6] R. Dodge and T. S. Cline. The angle velocity of eye movements. *Psychological Review*, 8(2):145, 1901.

[7] H. Drewes and A. Schmidt. Interacting with the computer using gaze gestures. In *Human-Computer Interaction–INTERACT 2007*, pages 475–488. Springer, 2007.

[8] P. M. Fischer, J. W. Richards, E. J. Berman, and D. M. Krugman. Recall and eye tracking study of adolescents viewing tobacco advertisements. *JAMA*, 261(1):84–89, 1989.

[9] P. M. Fitts, R. E. Jones, and J. L. Milton. Eye movements of aircraft pilots during instrument-landing approaches. *Ergonomics: Psychological mechanisms and models in ergonomics*, 3:56, 1950.

[10] FOVE. Fove vr. URL http://www.getfove.com/.

[11] M. Guim. The eye tribe tracker lets you control your tablet with your eyes, January 2015. URL http://www.windowscentral.com/eye-tribe-tracker-hands-on.

[12] M. N. H. and T. E. L. Head-mounted eye-tracker camera. In *Head-mounted eye-tracker camera. Journal of the Optical Society of America*, pages 713–716. Optical Society of America, 1962.

[13] HP. Hp lp2465 24" lcd monitor - technical specification. URL http://h20564.www2.hp.com/hpsc/doc/public/display?docId=c00871848.

[14] E. B. Huey. *The psychology and pedagogy of reading*. The Macmillan Company, 1908.

[15] P. Isokoski. Text input methods for eye trackers using off-screen targets. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, pages 15–21. ACM, 2000.

[16] R. J. Jacob. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 11–18. ACM, 1990.

[17] E. Javal. Essai sur la physiologie de la lecture. In *Annales D\'Oculistique*, 1879.

[18] Microsoft. Surface, . URL http://www.microsoftstore.com/store/mseea/no_NO/cat/categoryID.66342300?gclid=CKmD2Lju9MsCFYSIGQodUOYDnw&gclsrc=ds&tduid=(08a545928920aba2a2e8f55796851dad)(193030)(1017310)(jkp_CKmD2Lju9MsCFYSIGQodUOYDnw)().

[19] Microsoft. Microsoft surface pro 4 - 128gb / intel core m3, . URL http://www.microsoftstore.com/store/msusa/en_US/pdp/Surface-Pro-4/productID.325711500.

[20] Microsoft. Lifecam cinema, . URL https://www.microsoft.com/hardware/en-gb/p/lifecam-cinema.

[21] MSI. Gt72s 6qe dominator pro g tobii, 2016. URL https://www.msi.com/Laptop/GT72S-6QE-DOMINATOR-PRO-G-tobii.html#hero-overview.

[22] B. J. Oates. Researching information systems and computing. pages 71–280. SAGE, 2006.

[23] K. Pfeuffer, J. Alexander, M. K. Chong, and H. Gellersen. Gaze-touch: combining gaze with multi-touch for interaction on the same surface. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 509–518. ACM, 2014.

[24] R. Radach, J. Hyona, and H. Deubel. *The mind's eye: Cognitive and applied aspects of eye movement research*. Elsevier, 2003.

[25] K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372, 1998.

[26] N. N. Rommelse, S. Van der Stigchel, and J. A. Sergeant. A review on eye movement studies in childhood and adolescent psychiatry. *Brain and cognition*, 68(3):391–414, 2008.

[27] T. Rudnicki. Eye-control empowers people with disabilities, June 2012. URL `http://www.abilities.com/community/assistive_eye-control.html`.

[28] J. San Agustin, H. Skovsgaard, E. Mollenbach, M. Barret, M. Tall, D. W. Hansen, and J. P. Hansen. Evaluation of a low-cost open-source gaze tracker. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, pages 77–80. ACM, 2010.

[29] S.-M. W. Service. In tracking net, the eyes have it. *Eugene Register Guard*, page 12, 1998.03.19.

[30] SMI. Eye tracking upgrade for the oculus rift dk2. URL `http://www.smivision.com/en/gaze-and-eye-tracking-systems/products/eye-tracking-hmd-upgrade.html`.

[31] R. star games. Grand theft auto 5. URL `http://www.rockstargames.com/V/`.

[32] Tobii. X-series, . URL `http://acuity-ets.com/products_x-series.html`.

[33] Tobii. Tobii eyex, . URL `http://www.tobii.com/xperience/`.

[34] Ubisoft. Tomclancy the division, 2016. URL `http://tomclancy-thedivision.ubi.com/game/en-ca/home/`.

[35] B. M. Velichkovsky and J. P. Hansen. New technological windows into mind: There is more in eyes and brains for human-computer interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 496–503. ACM, 1996.

[36] C. Ware and H. H. Mikaelian. An evaluation of an eye tracker as a device for computer input2. In *ACM SIGCHI Bulletin*, volume 17, pages 183–188. ACM, 1987.

Appendix

A Consent Statement

Consent Statement

Participant for the usability test

I hereby confirm that I have been given information about the research, possibility to ask questions and been given all knowledge that I'll require to conduct this test. I understand that this test is voluntary and I can deny the use of my data whenever I want without reason.

There will be recording of eye position both on the screen and the distance from the screen during the usability test. This is done to ensure possibility to review the data later if needed. This test will be made anonymous so that the tester's identity will not be exposed. This also include information that will be published will not be able to connect to any single person. Only people involved in the project itself will be able to view the data result.

The project group reserves any and all right to pictures from and during the usability test.

I give my consent to participate in this study.

Trondheim,______________

____
Signature

B TEST LEADER GUIDE

Test Leader Guide

This document was used during each usability test session to make sure all participants got the same information and was asked the same question.

Welcome and Introduction

- Welcome the participant and introduce us
    a. Who are we
    b. What technology are we working with
    c. The duration of the test approximate to 40 minutes
    d. Why the they was chosen for this test
- Showcase the device in the lab
    a. What they are gonna interact with
        i. The computer
        ii. The camera
        iii. The eye tracker device
        iv. Voice recording device
        v. The tablet
        vi. Keyboard and mouse
    b. Where they are gonna sit
    c. How far they are gonna sit
- Explain the concept of usability testing
    a. It is not you we are testing, it is the system
    b. You can always abort a task or the whole test if you feel to, and you do not need to explain why you want to abort
    c. The tasks are created such that you will struggle to finish some of them, and they are created that way with a purpose
    d. It is important to explain your experience during the test
    e. We can not do the task for you, however we can give you hints if necessary

Consent Statement

- We will make sure that you will be kept anonymous. Your name will not be connected to the results from this test, and if desired your result can be deleted from this test if necessary.

Explain Gaze Interaction

- The concept of gaze interaction, that you can operate application using where you look.

Any Questions Before We Begin?

Explain the Calibration Process

- Guide the participant through calibration process
- Check the calibration

Start Recording

- Start the recording of heatmaps
- Start the recording of video during testing

Start the First Module

- If the module will use gaze interaction:
    1. Explain the interaction for this specific module
    2. Make sure the participant at least tried all the interaction this module can offer once each, and form the basic understanding of the image, map or webpage they are on
    3. Give the participant some free time to test interaction to get familiar
    4. Ask if the participant are ready for tasks
    5. Begin the first task if participant is ready, otherwise back to point 3
    6. For each task start the timer

7. Repeat until all tasks are complete
8. Switch to next interaction method

- If the module will use touch interaction:
    1. Give the tablet to the participant
    2. Explain the interaction for this specific module if needed
    3. Make sure the participant knows how to operate the tablet, by asking them to get basic understanding of the image, map or the webpage they are on
    4. Ask if the participant are ready for tasks
    5. Begin the first task if participant is ready, otherwise back to point 3
    6. For each task start the timer
    7. Repeat until all tasks are complete
    8. Switch to next interaction method
- If the module will use keyboard and mouse interaction:
    1. Give the participant access to keyboard and mouse
    2. Explain the interaction for this specific module if needed
    3. Make sure the participant knows how to operate the computer, by asking them to get basic understanding of the image, map or the webpage they are on
    4. Ask if the participant are ready for tasks
    5. Begin the first task if participant is ready, otherwise back to point 3
    6. For each task start the timer
    7. Repeat until all tasks are complete
    8. Switch to next interaction method

Card Ranking

- After all three interaction methods for one single module have been completed
    - Present the cards
    - Ask the participant to order them in the order of preference

    - After the participant is finished, asked them why they ordered it in this specific order
    - What are the overall experience using gaze interaction?
    - What could have been made to make gaze interaction better for this module?
    - What are the advantage and disadvantage of gaze interaction compared to the other two interactions?
    - In what situation could you imagine yourself use gaze interaction?
- Continue to the next module

Test Closure

- If all modules are completed:
    - Do you think this technology is something that you would have used in the future?
    - Is there anything more you would like to add?
    - Thank you for your participation

C TEST NOTES

Tester Information:

Test ID:	
Name:	
Age:	
Gender:	
Education:	
Occupation:	
Computer Experience:	
Task Experience:	
Eyetracking Experience:	

Pan and zoom Module:

	Time used:	Number of failures (note task unfinished):	Number of assists:
Touch device:			
Mouse & keyboard			
Eyetracking			

Card order:____

Reasoning behind the card orders:

Eye tracking:	
Mouse/Keyboard:	
Touch:	

Swipe Module:

	Time used:	Number of failures (note task unfinished):	Number of assists:
Touch device:			
Mouse & keyboard			
Eyetracking			

Card order:____

Reasoning behind the card orders:

Eye tracking:	
Mouse/Keyboard:	
Touch:	

Scroll Module:

	Time used:	Number of failures (note task unfinished):	Number of assists:
Touch device:			
Mouse & keyboard			
Eyetracking			

Card order:____

Reasoning behind the card orders:

Eye tracking:	
Mouse/Keyboard:	
Touch:	