NTNU – Trondheim
Norwegian University of
Science and Technology

Empirically Based Design Guidelines for Gaze Interaction in Windows 7

Børge Rødsjø
Håkon Raudsandmoen

Master of Science in Computer Science
Submission date: June 2012
Supervisor: Dag Svanæs, IDI

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

Abstract

The purpose of this study has been to test the use of gaze interaction in common everyday computer tasks, with the intent to suggest design guidelines for gaze interaction in Microsoft Windows 7. This has been done by organizing a user test with fifteen participants, using a self-made gaze interactive software called *Discovery* and a Tobii X60 eye tracker.

Five demo applications have been created within the *Discovery* software, all utilizing gaze interaction. They are customized to reflect five user test tasks; playing a video game, exploring a picture gallery, doing drag and drop operations, browsing a web page and interacting with different Microsoft Windows controls. The four types of controls tested are command buttons, links, check boxes and sliders. Both quantitative and qualitative data were gathered during the user test.

Through a discussion of the test results, we were able to suggest ten specific design guidelines for gaze interaction. These covers both the tested controls, drag and drop operations, automatic scrolling as well as the use of head gestures. Additional findings indicate that gaze interaction is more suitable for passive tasks such as reading with automatic scrolling, than for more physical tasks like doing drag and drop operations. To support gaze interaction, we found that current software will either require a major redesign or to be used in a combination with other interaction styles.

Eye tracking technology has improved over the last years, becoming increasingly affordable and accurate. Through this study we have seen that gaze interaction has much to offer everyday computing. By recommending fundamental design guidelines we hope to aid software developers and designers in the development of future gaze interactive systems.

Sammendrag

Hensikten med denne oppgaven har vært å teste bruken av blikkinteraksjon i kjente hverdagslige datamaskinoppgaver, med det formål å foreslå designretningslinjer for blikkinteraksjon in Microsoft Windows 7. Dette har blitt gjort ved å organisere en brukertest med femten deltakere, bruke en selvutviklet blikkinteraktiv programvare kalt *Discovery* og en Tobii X60 øyesporer.

Fem demoapplikasjoner har blitt laget i *Discovery*-programvaren, som alle nyttiggjør seg av blikkinteraksjon. De er spesiallaget for å gjenspeile fem brukertestoppgaver; spille et spill, utforske et bildegalleri, gjøre dra-og-slipp-operasjoner, lese en nettside, samt å interagere med forskjellige Microsoft Windows kontrollere. De fire kontrollerne som ble testet var knapper, lenker, avkrysningsbokser og glidebrytere. Både kvantitative og kvalitative data ble samlet inn under brukertesten.

En diskusjon av testresultatene gjorde det mulig å foreslå ti spesifikke designretningslinjer for blikkinteraksjon. Disse dekker både de testede kontrollerne, bruk av hodebevegelser, dra-og-slipp-operasjoner og automatisk scrolling. Ytterligere funn indikerer at blikkinteraksjon er bedre egnet for passive oppgaver slik som å lese med automatisk scrolling, enn for mer fysiske oppgaver slik som å gjøre dra og slipp operasjoner. For å støtte blikkinteraksjon ser vi at dagens programvare vil enten trenge store endringer i sitt design, eller bli brukt sammen med en annen interaksjonstype.

Øyesporingsteknologi har forbedret seg de siste årene, blitt rimeligere og mer presis. Gjennom denne oppgaven har vi sett at blikkinteraksjon har mye å tilby vanlig bruk av datamaskiner. Ved å forelså fundamentale designretningslinjer håper vi å kunne hjelpe programvareutviklere og designere i utviklingen av fremtidens blikkinteraktive systemer.

Preface

This report is a result of the work done the last semester of the Computer Science Master Study at the Department of Computer and Information Science (IDI) at the Norwegian University of Science and Technology in Trondheim. In addition to the report, the prototype source code uploaded in DAIM is a part of this master's thesis.

We would like to thank our supervisor Dag Svanæs, Professor at IDI, for his guidance throughout our work the spring of 2012. As an expert in the HCI field, he has been an important resource in our research. We would also like to thank Terje Røsand, Senior Engineer at IDI and the Norwegian EHR Research Center (NSEP), for technical assistance in the usability lab.

At last we would like to thank all system testers and user test participants that has been involved in this project, whose contributions were essential for our research.

Håkon Raudsandmoen and Børge Rødsjø

Trondheim, June 15, 2012

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter contains the motivation for this study as well as a statement of the research goal and research questions. In the end we present our research method and an outline of the report.

1.1 Motivation

Eye tracking is a technology that has been of interest for researchers since the late 1800s (Javal, 1879). At that time, invasive eye trackers were mounted on peoples' heads or eyes. Earlier research in the field was concerned with how the eyes operated when people read text and looked at objects. During the 1900s the technology improved, eye trackers became less invasive. When graphical user interfaces (GUI) became popular in the early eighties, research was done to find out how the GUI should look and how people would interacted with it. During this research, eye trackers were mainly used to capture where users looked at the monitor. Researchers would then sit down to watch and analyse the captured session in retrospect.

During the last decades, eye trackers have been used as an input device for human-computer interaction. This has opened up new possibilities for persons with disabilities, helping them to communicate and function in everyday life (Majaranta and Raiha, 2002). As the eye trackers have left the labs and are beginning to enter the public market we see an opportunity to utilize the eye tracker in everyday computer tasks.

To assure that the GUI is, and behaves, similar for the entire operating system, manufacturers often publishes *design guidelines*. These guidelines are intended to enabled designers and developers to create an uniform user experience across different applications. Large cooperation like Apple, Microsoft and Google publish design guidelines so that designers and developers all over the world can create a uniform look and feel of their third-party applications. Most end-users would appreciate a consistent GUI throughout the operating system. Consistency makes them more familiar with what they are used to see, and enables them to use already learned interaction techniques to control the applications.

Design guidelines are adapted to different interaction styles. For instance, the recommended size of a button for a Microsoft Windows operating system will vary depending on whether the interaction style is mouse, pen or touch. There are however no guidelines for gaze interactive applications.

The Norwegian EHR Reseach Center (NSEP, 2012) has a Tobii X60 eye tracker available, which has been used for usability testing both in-house, by master students, Ph.D candidates and in industry. The eye tracker is used together with Tobii Studio (Tobii, 2012d) that captures eye movements during a session and provides several statistics for further analyses. This equipment has traditionally been used to test the usability of screen based applications. By using the newly released software development kit (SDK) from Tobii, developers may take control over the gaze stream from the eye tracker and use it in their own applications. The SDK enables developers to create powerful software applications where users use their eyes as a control mechanism. This opens up many new opportunities to be explored.

During the fall of 2011, we started the creation of a software that utilizes this technology (Raudsandmoen and Rødsjø, 2011). The software, called *Discovery*, is optimized as a test bench for user testing of gaze interaction. The software consists of several demo applications that represent everyday computer tasks, and is ready for user testing.

1.2 Research Goal

By conducting a series of user tests with a representative selection of participants from the Norwegian population, and by using the *Discovery* software created by Raudsandmoen and Rødsjø (2011), this research should suggest a set of guidelines for designing Windows-based applications that utilize gaze interaction. The research should be conducted using the available equipment at the NSEP usability lab in Trondheim (NSEP, 2012), most importantly the Tobii X60 Eye Tracker (Tobii, 2012b) together with the Tobii SDK (Tobii, 2012c).

Our research goal is:

Suggest a number of empirically based design guidelines for gaze interaction in Windows 7.

We plan to achieve the research goal by answering the following research questions:

RQ1: *How do users assess gaze interaction for solving a set of common everyday computer tasks?*

RQ2: *How does gaze interaction perform when used to solve a set of common everyday computer tasks?*

RQ3: *Which design guidelines can be suggested for gaze interaction in Windows 7, based on performance and user assessments?*

1.3 Research Method

To achieve the stated research goal we intend to carry out a set of *user tests*. A related term is *usability testing*. The definitions of these terms varies and also interchanges. By user test we simply mean that a user is testing a product or a service, the purpose and content of the test is to be defined by the researcher. A usability test reveals usability problems of a product or service. This study will use the term *user test*, as our purpose with the testing is not do find usability flaws of a system, but rather to explore an interaction technology. Toftøy-Andersen and Wold (2011) presents the following characteristics of a user test:

1. simulates a real situation
2. contains concrete tasks
3. is a test where the user is being observed
4. is used to evaluate the usability of a system

For our purpose we will use the definition above, with a slight change of the first and last characteristic. Regarding the first characteristic, our user testing will not simulate a realistic situation since the prototype is not meant to act as a complete end-user application. The prototype used is especially designed for testing gaze interaction and acts as a test bench for that purpose. This will be described in detail in the prototype chapter (Chapter 3).

For the last characteristic; the user testing performed in this project is not done to evaluate the usability of a system, but rather to evaluate the usability of gaze interaction in common everyday tasks. Normally when doing system development with usability testing one follows an iterative cycle: one conducts a usability test of a system and discovers problems with it. The next step is to correct them and propose a redesign. Even though the redesign should correct the problems found, a new test is required, and so the cycle continues. This will however not be done in this case as our user testing purpose is not to improve the prototype, but to recommend a set of gaze interaction guidelines.

In the user test, the test team uses research methods such as observations, semi-structured interviews and so on. For more information about our research method, see Chapter 4.

1.4 Report Outline

Chapter 1
Chapter 1 holds the motivation for this project as well as our research goal and accompanying research questions. In the end there is a short description of our research method as well as this report outline.

Chapter 2
This chapter gives an introduction into eye tracking history and the eye tracking technology. The Microsoft Windows interaction guidelines are then described, followed by a general description of user testing.

Chapter 3
The prototype used in this project is described in Chapter 3. The chapter starts with a prototype introduction before explaining each of the prototype's demo applications in detail.

Chapter 4
This chapter explains our test method, including a presentation of the research method, participants, tasks, test facility, test equipment, data gathering methods, test measures and test procedure. In the end of the chapter there is a description of how we intended to suggest design guidelines based on our results.

Chapter 5
Chapter 5 holds all of our results. It is divided into subchapters according to which task it contains results from.

Chapter 6
In this chapter the results are analysed and discussed. It starts with the task-dependent results, then moves on to additional findings. In the end we include a method discussion and research limitations.

Chapter 7
The last chapter is devoted to our conclusion and further work.

Chapter 2

Background

This chapter describes the evolution of the eye tracking research. This by giving a brief overview of eye tracking history and a presentation of today's *state of the art* eye trackers.Furthermore, software design guidelines are explained, as well as an introduction of the Windows User Experience Interaction Guidelines. Finally, a general description of user testing is given.

2.1 Eye Tracking History

This chapter is dedicated to eye tracing history. It provides a brief overview of what has been done in the field of eye tracking since the first eye trackers were made available over a century ago. We present both what has been done using eye tracking for analysing interfaces (measuring usability) as well as an actual control medium for human-computer interaction. This chapter provides the information needed to understand the progress and development in the field, and what to expect from today's eye trackers.

2.1.1 A Century with Eye Tracking

Eye tracking technology has been around for over 100 years, and the technology has improved from invasive mechanical arrangements on the user's cornea into today's video cameras that capture the eye movements without influencing the user physically at all. The first documented research on eye tracking available is from Javal (1879), who used visual observations to track eye movements on users late in the 19th century. At the start of the 20th century the first non-invasive eye tracking techniques were developed, where the eye tracker responded to light reflections from the user's cornea (Dodge and Cline, 1901). The user's head had to be completely motionless and only horizontal eye movements were captured. However, through the insertion of a small white speck in the user's cornea, Judd et al. (1905) could record eye movements in two dimensions with motion picture recording.

Figure 2.1 and 2.2 shows early methods for measuring eye movements on users.

Time went by and in 1950 Paul Fitts and his colleagues (Fitts et al., 1950) conducted a study where they used eye trackers to investigate how the eye movement of a pilot was during instrument-landing approaches. This study has been considered as the first contribution to the use of eye tracking in *usability engineering* - the systematic study of users interacting with products to improve product design (Jacob and Karn, 2003).

In the 1950s and 1960s head mounted eye trackers were made and thus reduced the limitations of the head movements for the users. (Mackworth and Thomas, 1962). Through the 1970s a lot of research was done in the field of psychology and physiology in order to explore how the eye operated and what it could reveal about perceptual and cognitive processes. The eye tracking technology also developed further, and the research and progress done in the 1960s and 1970s made eye tracking useful as a real-time interaction style, instead of only using the gaze data retrospectively (Anliker, 1976). Not much work was done with respect to the use of eye tracking in

Figure 2.1: By using Electro-oculography (EOG) one could measure the potential differences between eye muscles of the users and thus deduce where the users were looking. (Kumar, 2007)

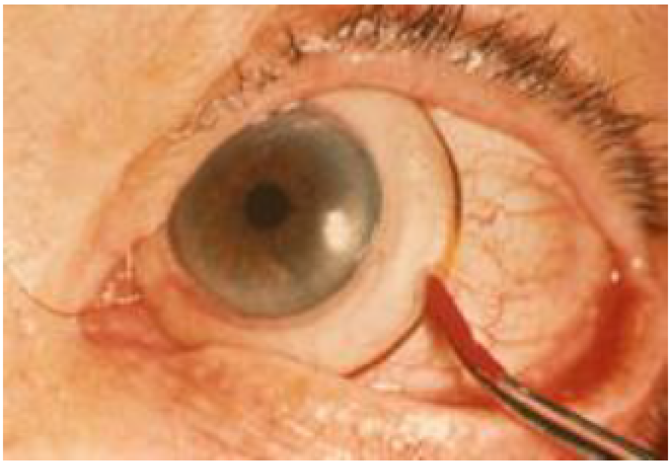


Figure 2.2: A scleral coil contact lens being inserted into a user's eye. When a user kept the head stationary inside a magnetic cage, the user's eye movements created changes in the magnetic field which could be used to measure them. (Kumar, 2007)

usability engineering. However, we have to remember that usability at the time was associated with computer command line entries, punched paper cards and tapes, and printed lines of alphanumeric output (Jacob and Karn, 2003).

During the 1980s eye tracking technology was used to investigate the usability of computer software, as personal computers and the Internet emerged and developed. As the eye tracking technology became increasingly relevant and applicable, researchers over the world started using eye tracking as an interaction style, both alone and in conjunction with more conventional modes of human-computer interaction. See (Bolt, 1981) (Bolt, 1982) (Levine, 1984) (Glenn et al., 1986) (Ware and Mikaelina, 1987).

2.1.2 Eye Tracking as an Interaction Style

Traditionally, eye tracking has been used retrospectively. This means that after an eye tracking session has been conducted, researchers would review and analyse the data in retrospect. During the last years eye trackers have gotten more accurate and easier to set-up, thus making them more available and relevant for a wider range of uses.

The use of the eye tracker as a control mechanism has mostly been studied for the possibilities for disabled users, who can use their eyes only for computer input. For instance Hutchinson et al. (1989) and Levine (1984) report work of which their primary focus was disabled users. Even though the use of the interfaces created for this purpose appears slow and challenging for non-disabled users, they are useful to their intended users (Jacob and Karn, 2003).

Another area in which eye movements have been used as an input device, is to make the illusion of a better graphical display. This has been done in flight simulators where Tong and Fisher (1984) created a display that gave a higher resolution where the user was gazing, and a lower resolution in the peripheral vision. This gives the user an illusion of always having a high resolution graphical display, while observers of the session may see that the resolution changes as the user moves his or her eyes. This technique changes the visual perception of the display, but does not however alter the human-computer dialogue.

Jacob and Karn (2003) discuss how to incorporate eye movements in the human-computer dialogue. The obvious solution would be to substitute the mouse with the eye tracker's *x,y* stream of gaze data. However, the eyes move very differently from the intentional way the hand moves a mouse. Jacob and Karn (2003) recommend the following considerations to be taken into account when comparing a mouse with eye movement:

- Eye movements are much faster than mouse movements

- "Operating" the eye requires no training or particular coordination for normal users; they simply look at an object. The control-to-display relationship for this device is already established in the brain.

- Eye movements give information about both where the user looks, and where the user has its attention
- It is difficult to move the eyes consciously and precisely at all times, while you always have full control with a mouse
- Eye movements are always *on*. Unlike the mouse which you can grab and release with your hand, you can not enable or disable your eyes in the same way
- Eye tracking lacks an analogue of the integral buttons most mice have. Using blinks to perform a click is a less than optimal solution.
- Eye tracking equipment is still far less stable and accurate than most manual input devices

A problem using the eyes as a control medium is for the interface to separate between when the users want to look at objects on the monitor, and when they want to interact with them. This is called the *Midas Touch* problem. It is the disambiguation of when the user is looking and when the user intends to perform an action (Kumar, 2007). The challenge in building a useful eye tracker interface is to avoid the Midas Touch problem. Moving the eyes is quite non-intentional, users do not intend much by moving their eyes, and far less than intentional actions with a keyboard or a mouse.

Provided that one overcomes the problems in building a good eye tracker interface, researchers have found that using an eye tracker interface is perceived as a highly responsive system, almost as though the system is executing the user's intentions before they are expressed (Jacob, 1991). This is the benefit one seeks from eye movement-based interaction.

2.2 Eye Tracking Technology

In the previous chapter we saw how the development and usage of eye tracking has progressed the last century. This chapter focuses on how far the technology has come today, and how it is currently being used. The first part addresses eye tracking *state of the art* and what to expect of today's eye trackers. In the last part the eye tracker that has been used during this study, the Tobii X60 eye tracker, is presented.

2.2.1 State of the Art

During the last two decades the eye tracking technology has left the lab and made its entrance into the public market. With the use of high resolution video cameras and infrared lights, the current eye trackers are not intrusive at all. The eye tracker is positioned next to (Figure 2.3a) or integrated in the monitor (Figure 2.3b). Consequently the users are free to move their head, as long as the eyes are in range of the eye tracker camera. Eye trackers may also be head mounted and thus provide total freedom of head movements.

(a) The Tobii X120 standalone eye tracker

(b) The Tobii T60XL integrated eye tracker

Figure 2.3: Two physically different eye trackers that suits different needs, however, the technology remains the same. (Tobii, 2012a)

This physical constraint on the user's head movement remains as one of the most significant barriers to solve, but for typical WIMP (windows, icons, menus and pointer) human-computer interfaces this constraint is not that disturbing (Jacob and Karn, 2003). Eye trackers may have both a single camera or multiple cameras, supported with infrared LEDs. The advantage of a single-camera set-up is reduced costs and size.

The infrared LEDs are not intrusive for the user, and creates general illumination and generate reflexes on the surface of the cornea and retina.

These corneal and retina reflexes are used to find the eyes in the camera image and determine the location of the centre of corneal curvature in space (Figure 2.4).

Figure 2.4: Infrared light creates reflexes on the user's cornea that the eye tracker camera can capture and thus find the eyes in the camera image

Most eye trackers have an accuracy of about 1 degree or less. This is achieved by using complex image processing algorithms to deduce where the gaze is. Eye trackers operates with different frame rates, but usually in the range of 50-300 Hz. This means that software that uses eye tracker data receives the eye position and/or the gaze 50-300 times a second.

Eye trackers collect information about several aspects concerning the user's eyes and gaze. The gaze stream may for instance constitute:

- the spatial position of both eyes (relative to the eye tracker)
- the pupil size for both eyes
- if the user gazes with one eye, the eye tracker can deduce if it is the left or right eye
- the gaze direction onto the plane where the eye tracker got calibrated
- and the gaze in 3D coordinates on the calibration plane

Eye trackers needs a set-up procedure before they can be used. If the eye tracker is not incorporated into a monitor, it needs to know where it is positioned relative to the monitor, as well as the size of the monitor. When a user wants to use the eye tracker a calibration procedure is needed. The procedure often involves simple objects that the user have to gaze at as they are presented at different locations on the monitor. With this procedure the eye tracker learns how the user's eyes operates, and can thus provide highly accurate eye tracking of the user's eyes. See Figure 2.5 for a picture of the calibration process with the standalone Tobii X60 eye tracker at the usability lab at NSEP in Trondheim.

Figure 2.5: User calibrating with Tobii X60 eye tracker at the usability lab at NSEP in Trondheim, Norway

2.2.2 The Tobii X60 Eye Tracker

The Tobii X60 eye tracker (Figure 2.6) is a standalone eye tracker that can be positioned on flat surfaces.

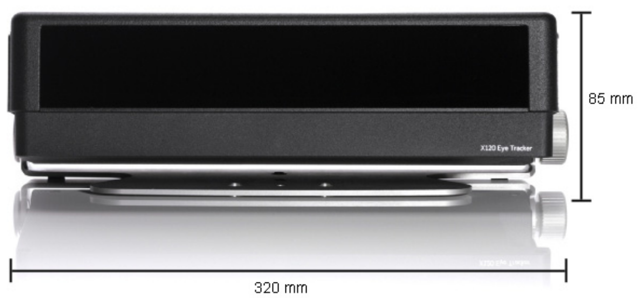


Figure 2.6: Picture of Tobii X120, the X60 version looks just the same. (Tobii, 2012a)

It works with most standard monitors, as well as with projectors. This eye tracker can also be used to track eye movements when investigating real world objects like newspapers, books, shopping shelves and televisions. As the name implies, it operates with a frame rate of 60 Hz. Similarly the X60's big brother, the X120, operates at 120 Hz. Tobii also has equipment enabling these eye trackers to do eye tracking on mobile devices, and a monitor mount to easily facilitate different monitor set-ups.

The Tobii X60 has an eye position and gaze accuracy of 0.5 degrees. Its freedom of head movement is limited to a spatial cube with dimensions 44 x 22 x 30 cm (17 x 9 x 12 inches) (width x height x depth). This allows the user some free head movements, and should not limit the user much when using typical desktop applications (like mentioned in Chapter 2.2.1). The weight is 3 kg (7 lbs) which makes it easy to move around and set-up in different environments.

The accurate and precise data leads to reliable research results, and the robust eye tracking capability allows work with a wide cross-section of the population. It creates a distraction-free test environment to ensure natural behaviour of subjects and valid research data.

In conjunction with the released Software Development Kit (SDK), programmers may easily implement their own programs that utilize the data stream from the eye tracker. The currently released SDK (version 3.0 RC 1) (Tobii, 2012c) supports programming languages like .NET (C#), Python 2.6, C++, Objective C (Cocoa) and several operating systems like Windows, Ubuntu and Mac OS X. The SDK gives developers access to real 3D coordinates from the eye tracker, real time access to the head movement box, and the possibility to both set and retrieve settings for the calibration plane. This openness with a free SDK allows programmers to be innovative, creative and to utilize the technology in a good way.

2.3 Windows User Experience Interaction Guidelines

In order to provide guidelines for gaze interactive programs for Windows (as stated in Chapter 1.2) we need to know what a guideline document contains and what guidelines really are. This chapter describes what is meant by a *guideline* and a *control*, and what the Windows User Experience Interaction Guidelines are about. Furthermore, the chapter sets the focus on the interaction part of the guidelines, and describes the touch interaction style to illustrate what content interaction guidelines usually consist of.

2.3.1 What is a Guideline?

> *A guideline is a statement by which to determine a course of action. A guideline aims to streamline particular processes according to a set routine or sound practice. By definition, following a guideline is never mandatory.*

(United States Deptartment Of Veterans Affairs, 2011)

In the case of user experience interaction guidelines, guidelines are seen as software development documents which offer software developers a set of recommendations. These recommendations often have the goal of creating a consistent experience across an environment. They often describe how the visual design should be, and how user input and interaction techniques work. There are similar guidelines as the Windows User Experience Interaction Guidelines for other operating systems as well: Apple Mac OS X has the *Mac OS X Human Interface Guidelines* (Apple, 2012) while Google Android has the *Android Design* (Google, 2012).

2.3.2 What is a Windows Control?

When reading interaction guidelines for operating systems, the term *control*, is frequently used. Other terms are also used, such as *GUI elements* or *widgets*. Microsoft Corporation (2011a) defines a *control* as;

> *A control is a child window that an application uses in conjunction with another window to enable user interaction.*

Furthermore, the Microsoft Corporation (2011a) states that controls provides the user with:

- a way to type text, choose options, and initiate actions.
- a variety of services, such as letting the user choose commands, view status, and view and edit text.

Basically by *control* we refer to visual screen elements that provides a user with information or with which a user can interact. Typical examples would be; check boxes, buttons, sliders, links etc. Some of the standard Microsoft Windows controls are shown i Figure 2.7 below.

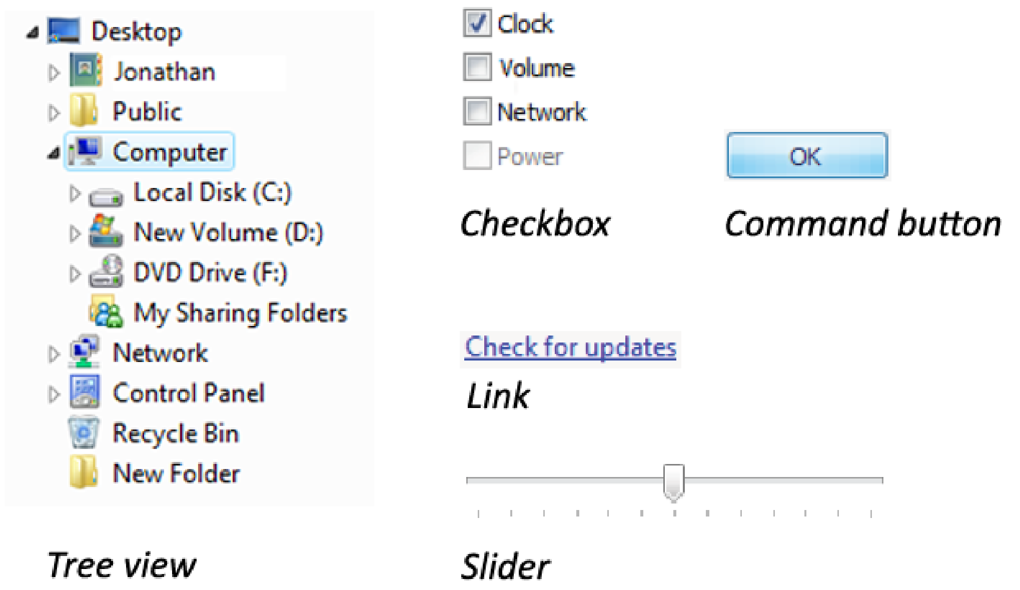


Figure 2.7: Some of the Microsoft Windows standard controls

2.3.3 The Windows User Experience Interaction Guidelines

Considering desktop operating systems, Microsoft Windows is the most commonly used operating system in the world (NetMarketShare, 2012). In 2011, Microsoft Windows had 93,06 percent of the market share. The Windows User Experience Interaction Guidelines (hereafter abbreviated as the *UX guidelines*) is a set of recommendations regarding how to design applications for the Microsoft Windows platform (Microsoft Corporation, 2011b). It consists of basic design principles, and provides guidelines for all the basic Windows controls and current interaction styles. Furthermore, it explains for instance how commands (menus, toolbars, ribbons), text (font, style etc.), windows and messages (errors, warnings etc.) should be designed, and discusses the aesthetic issues a designer needs to consider. The recommendations are concluded with guidelines for improving a variety of user experiences and guidelines for the various places within the Microsoft Windows environment.

The UX guidelines states that the official goals for Windows 7 and Windows Vista are to:

- *Establish a high quality and consistency baseline for all Windows-based applications.*
- *Answer developers specific user experience questions.*
- *Make developers job easier.*

(Microsoft Corporation, 2011b)

2.3.4 Interaction Styles

As mentioned in the previous chapter, the UX guidelines (Microsoft Corporation, 2011b) provides among other things recommendations for different *interaction styles*. Please note that what Microsoft refers to as an interaction style is often called an interaction technique or method by others. In this report we will use the same terms as Microsoft Corporation (2011b). The UX guidelines covers topics like control usage, size, layout and spacing for different interaction styles. Guidelines are also given in important areas such as affordance and accessibility. The UX guidelines show that the design should vary depending on the chosen interaction style. For instance, the recommended control size for mouse interaction will in general be too small for touch interaction. Furthermore, a multi-touch command like zooming with touch interaction will be hard to do with a mouse. The overall message is that applications must be designed and adapted to the interaction style they should support. The styles covered in UX guidelines are:

- Keyboard
- Mouse and pointers
- Touch
- Pen

The different interaction styles may also be combined. The most common example is the use of mouse interaction together with keyboard interaction. This is often refereed to as *mixed interaction* or *multimodal interaction*. By combining different interaction styles, one may exploit the strength of one interaction style where another is weak. For example, when organize your files, mouse interaction will perhaps be a good choice. However, if you goal in addition is to rename some of the files, mouse interaction may not be the most effective solution. By combining it with a keyboard, one may exploit the strength of keyboard interaction (text input) while keeping the strength of mouse interaction (interaction with objects).

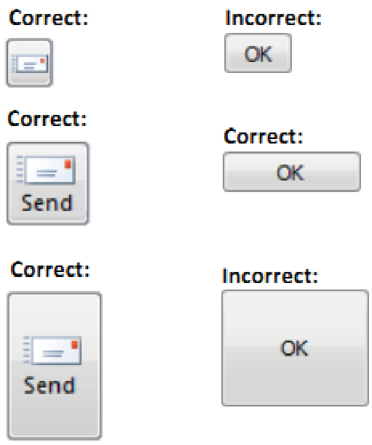
As seen from the list of interaction styles above, there is currently no official Windows guidelines for gaze interaction. This work aims to suggest such guidelines for a set of Windows controls and operations. The chapters on interactions styles should in that relation be examined more closely. As the touch interaction style has some similarities with gaze interaction, the touch interaction style has been selected as an example, and will be described more closely below.

Touch Interaction Style

According to Microsoft Corporation (2011b),

> *Touch refers to the way Windows lets you interact directly with a computer using a finger. Compared to using a mouse, keyboard, or pen, touch is often much more natural, engaging, and convenient.*

The touch interaction chapter deals with among other things guidelines on how Windows controls should be adapted to a touch interaction style. They should for instance be sized in a manner that fit the large surface area of a fingertip, recommended size is 40 x 40 pixels. The recommended general minimum size is 23 x 23 pixels (versus 16 x 16 px for mouse interaction). Note that these size recommendations are general, and will vary somewhat between specific controls. The recommendations for width and height ratio also varies. For instance, a command button meant to be used with mouse interaction is recommended to be at least 75 x 23 pixels. If enlarged, it is strictly recommended to keep its current width to height ratio. For touch interaction the recommended size for command buttons is minimum 40 x 40 pixels, but there is no recommendation to keep it squared when resized. Figure 2.8 comes from the UX guidelines (Microsoft Corporation, 2011b) and shows an example of command button resizing for both mouse and touch interaction.



(a) Touch resizing (b) Mouse resizing

Figure 2.8: Example from the UX guidelines (Microsoft Corporation, 2011b) showing how touch and mouse interactive command buttons respectively should be resized

Touch interaction is quick, but not particularly precise, thus the spacing between controls should be big enough to allow users to tap outside their intended target, while still hitting it. Preferably there should be at least 5 pixels between each control. It is also pointed out that the layout should be structured in such a way that controls are placed closely to where they are most likely to be used. Displayed in Figure 2.9, is illustrating in relation to how spacing vary when using mouse or touch interaction i Microsoft Windows.

Each interaction style has different advantages and disadvantages, and the guidelines should help utilize this. The UX guidelines do so by describing the different areas where different interaction styles have their strengths or weaknesses. A strength of the touch interaction style is for instance the use of multi-touch gestures. Multi-touch refers the surface's ability to recognize the presence of two or more points of contact with the surface. By moving multiple fingers in different ways, you are able perform different actions, such as the zoom and rotate gestures illustrated below (Figure 2.10).

Multiple simultaneously inputs like these are not supported by for example a mouse. The well know *mouse-over*, or *hover* function, in the mouse interaction style will on the other hand, not be supported by the touch interaction. This is considered a weakness as other actions must be made to not loose the additional information normally provided by the hover func-

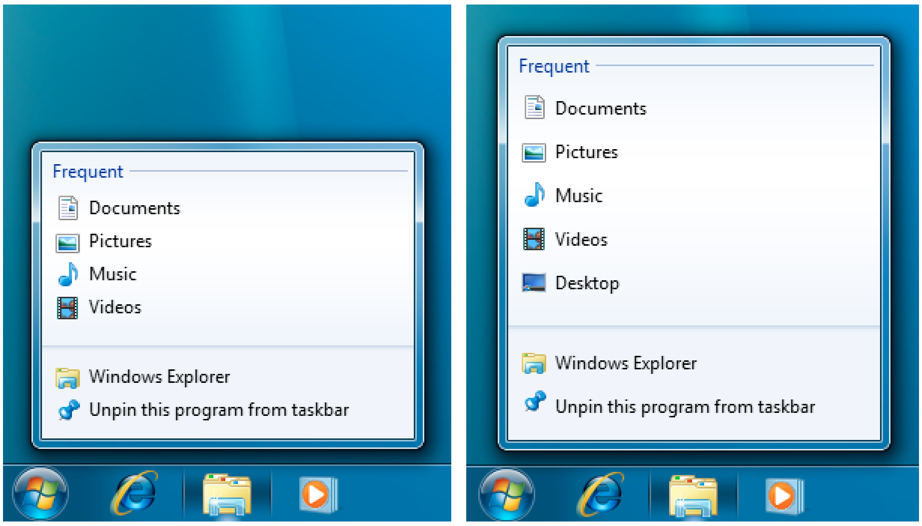
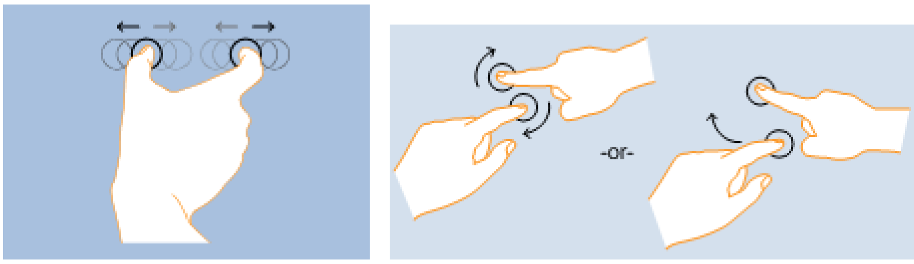


Figure 2.9: Example where the Windows 7 taskbar Jump Lists are more spacious when displayed using touch interaction style.Mouse input to the left, touch input to the right. (Microsoft Corporation, 2011b)



(a) The zoom gesture (b) The rotate gesture

Figure 2.10: Touch interaction gestures. (Microsoft Corporation, 2011b)

tion. In a similar way as with touch input, guidelines on eye tracker input should provide information about limitations and opportunities compared to inputs like mouse and touch.

2.4 User Testing

With the definition given in Chapter 1.3, user testing is an empirical research method. It being empirical means that we gain knowledge by observations or experience. User testing is all about observing users during a controlled experiment. It is recommended to have several people to just observe the user during the test, and to take notes of what is happening *on stage*. The stage does in this case represent the place where the user test is conducted. It could be any convenient location, such as a lab, a workplace, a private home or similar. On stage one normally has one person leading the test, the test leader, any observers, and the test participant. Below follows other typical ingredients in user test. (Toftøy-Andersen and Wold, 2011):

- A system to be tested
- A set of concrete tasks for the participants to solve
- Success criteria for each task
- Observation sheets
- A guide or plan to follow
- Questions or topics to discuss during or after the test
- Recording devices

2.4.1 A Typical User Test

The scope of a user test may vary. It can be anything from a small 5-10 minutes test over the lunch table for a hand-held device, to involving over 100 participants testing an online booking system. Regardless of the scope of the user testing, one should make sure that both the system and the test procedure are ready for testing. Before one starts user testing with participants, it is recommended to perform one or several pilot tests in order to make sure that the test has the desired quality one wants. In a pilot test one invites a colleague, friend or similar to play the role as a user, and the test team carries out a full user test with the volunteer. This is an effective way to reveal flaws in the test procedure, test equipment, tasks etc.

When using user testing as a research method one expects it to at least contain the three parts; introduction, task solving and wrap up.

In the introduction the test leader welcomes the person and informs the user about the product under test and about the test method. Depending on the size and scope of the test, this phase can be as short as only one phrase: *Welcome Tom! We would like your opinion about this system, please try it out and tell us your opinion*, or it can involve a longer explanation of the test and the system. In the next phase, the task solving, the participant is given some tasks to solve while the observer(s) takes notes. An important aspect is that one should not test the persons ability to solve the tasks, but rather how well the system reacts on the user interaction, and how easy it is to operate (measuring usability). When the user has finished all tasks, either with success or failure, the user is often free to share his or her thoughts about the system. The test leader and observers can also initiate a discussion and ask about particular findings that was observed during the task solving. Furthermore, the test team may have a small survey, interview or similar for the participant to answer. After this the test session is finished.

The empirical evidence from user tests may be analysed quantitatively or qualitatively. With recording devices that captures the screen and the user, with both video and sound, one can find interesting and useful quantitative information such as: *How many times did the user get confused? How many errors per minute did the participant have? How many keyboard short-cuts did the participant use?* Through the use of an eye tracker that captures users' gaze one can for instance examine if specific areas on the screen is of particular interest or not.

By analysing the qualitative data from a user test, for instance users' answers, comments, body languages and such, one may find results that contradicts the quantitative data. For instance; a user may solve all tasks with success and within the acceptable time limit, but the video recordings of the user shows that the user was very uncertain and confused while using the system.

2.4.2 Usability Metrics and Usability Test Reporting

So, what does user tests actually measures. Different usability metrics can be set for different types of tests. The most basic measures are for instance; success rate, the time a task requires, the error rate and users' subjective satisfaction (Nielsen, 2012). The International Organization for Standardization (ISO) has also defined some usability metrics. ISO 9241-11 (1998) defines how to measure usability in terms of user performance and satisfaction. The following usability measures should thus be taken into account when specifying or evaluating usability of a visual display terminal:

- **effectiveness:** Accuracy and completeness with which users achieve specified goals.

- **efficiency:** Resources expended in relation to the accuracy and completeness with which users achieve goals
- **satisfaction:** Freedom from discomfort, and positive attitudes towards the use of the product.

National Institute of Standards and Technology (NIST) Industry USability Reporting (IUSR) has created a Common Industry Format (CIF) for Usability Test Reports (ISO/IEC 25062, 2006). The intended readers are *"usability professionals in customer organizations who are evaluating both the technical merit of usability tests and the usability of the products"*, and *"other technical professionals and managers who are using the test results to make business decisions."* (ISO/IEC 25062, 2006). The CIF for Usability Test Reports is intended for use by usability professionals to report the results of summative usability testing. By using a standardized way of reporting usability tests, one reduces the training time for usability staff since individuals only need to learn to use one form regardless of where they work. It also enhances the potential for increased communication between vendors and purchasing organizations since readers of CIF-compliant reports will share a common language and common expectations. The main content of a CIF-compliant report should have the following:

- Title page
- Executive summary
- Introduction
  - Full product description
  - Test objectives
- Method
  - Participants
  - Context of product use in the test
    - Tasks
    - Test facility
    - Participant's computing environment
    - Display devices
    - Audio devices

    * Input devices
    * Test administration tools
  - Experimental design
    * Procedure
    * Participant general instructions
    * Participant task instructions
  - Usability metrics
- Results
  - Data analysis
  - Presentation of the results
- Appendices

Chapter 3

Prototype

The following chapter describes the prototype developed and used in this study. It starts with an introduction and a general description of the software, before explaining each of the its demo applications in detail.

3.1 Prototype Introduction

The prototype used in this project is called *Discovery version 2.0*. The first version of this software, version 1.0, was developed during our pre-project *"Exploring the Tobii X60 Eye Tracker as Input Device"* the fall of 2011. Version 2.0 is the result of several iterations with further development and testing. This improved version is tailored to fit this project's testing purposes. It consists of several Windows demo applications that make use of gaze interaction. Each of these applications act as a testing ground for a common everyday computer task (described in 3.4 - 3.8). The decision to use the Microsoft Windows environment was done based on the fact that it is the most commonly used operating system in the world (NetMarketShare, 2012).

The implemented demo applications is customized for five everyday computing tasks. These are:

- Playing a video game
- Exploring a picture gallery
- Doing drag and drop operations
- Browsing a web page
- Interacting with different controls

The last of the tasks mentioned above, holds four different types of controls. These are:

- Command buttons
- Links
- Check boxes
- Sliders

These controls are commonly used in standard Windows operating systems, office applications and online forms. Some of these controls have been used since their "birth" at Xerox PARC (Xerox Palo Alto Research Center) with the Xerox Alto and Xerox Star computers as late back as in 1973 (PARC, 2012). During almost 40 years these controls have been foundation stones in many graphical user interfaces, and were therefore chosen to be tested with gaze interaction.

Discovery version 2.0 has been developed for the Microsoft Windows platform with .NET C#, in the Integrated Development Environment(IDE)

Visual Studio 2010 Premium. Its structure and functionality will be described in the following chapters. For further details, see the source code attached to the delivery of this project.

3.2 Discovery Version 2.0

The entry point of the Discovery software is the main menu, which serves as a framework for all demo applications. It introduces important basic elements and functionalities inherited by the other applications. In addition to provide access to the demo applications, it also provides access to a calibration application as well as a mean for users to exit Discovery.

Figure 3.1 shows a screenshot of Discovery. The application window is split into two parts with the use of a Windows *split container*. The top split panel is used as a top menu, while opened demo applications are shown in the lower split panel. The top menu houses large buttons that are clickable with both mouse and gaze interaction. To the far right there is an exit button, that displays a prompt ensuring that the user really intended to quit the application. This as recommended in the Windows User Interaction Experience Guidelines (Microsoft Corporation, 2011b). In the far left corner of the top menu there is an information button that displays information about the application, shortcuts (explained in Chapter 3.2.4) and contact information (see Figure 3.5).

Being an entry point, the main menu is an important interface for catching the user's attention. The use of a simplistic interface with much space between elements, informative and colourful icons, large buttons and interesting text, have been important during the design of the main menu. It sets a colour theme and design layout that is consistent throughout the whole Discovery application. This has been done by creating a base class with a defined layout, and letting all other demo applications inherit the design from that base class. The Discovery software always runs in full screen mode. This is to utilize the eye tracking abilities in the best possible way, and to get more space in the application for GUI elements. Another advantage of running the application in full screen mode is that it will prevent the user from being distracted by other non-gaze interactive windows in the background.

3.2.1 Design

As recommended in the UX guidelines (see Chapter 2.3.4), we chose to customize the main menu design according to the type of interaction it should

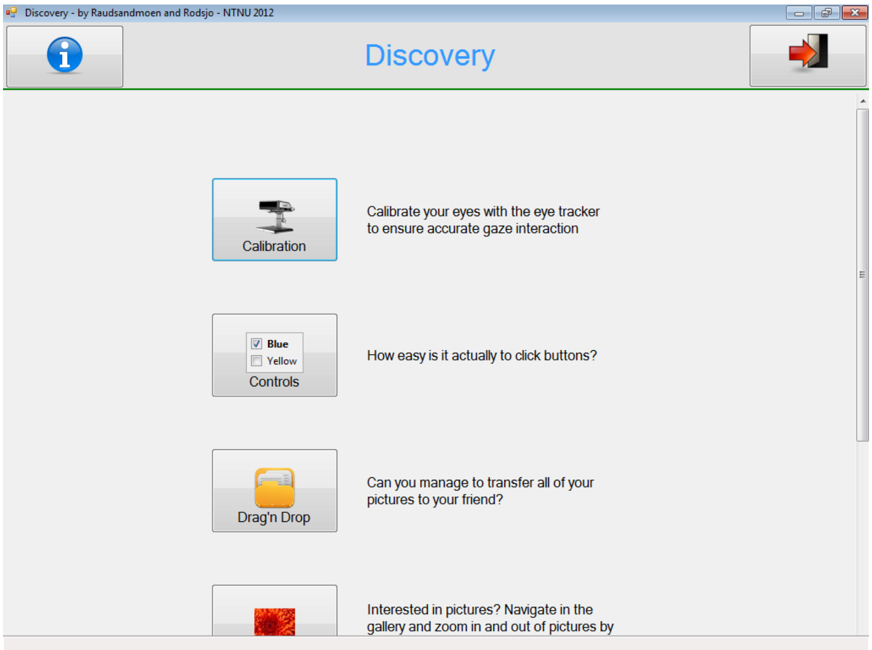


Figure 3.1: The Main Menu in Discovery v2.0

support. The main menu design does not contain a traditional Windows GUI layout with default font and button sizes. Instead it has large buttons with colourful icons and a larger sized font. Discovery is meant to be a gaze interaction application and thus it should also look like one. The interface should give the user a feeling of: *I can control this interface with my eyes.*

Several design proposals have been tried out before we ended up with the current one. First, a tab-based layout was implemented. It consisted of a Windows tab control, showing one demo application for each tab. With the default tab sizes being too small, the intended tabs proved challenging to hit with gaze interaction. Since an increase in tab size would put a limit to how many demo application we could support, a new design was tried out. It was a non-textual grid design with much of the same colours and elements as the current one. It consisted of several command buttons ordered in a grid. Each button had a picture to display its function. However, after adding textual explanation of each demo application, we ended up with the current design. It consists of large pictured menu buttons with associated text descriptions of each demo application, ordered in a centred one column list. It also facilitates scrolling, and is thus easy to expand with new demo applications (scalable). Figure 3.2 shows the main menu scrolled down. The gaze based scrolling is also implemented in *The Browser Application*, and will be explained in detail in Chapter 3.7.1.

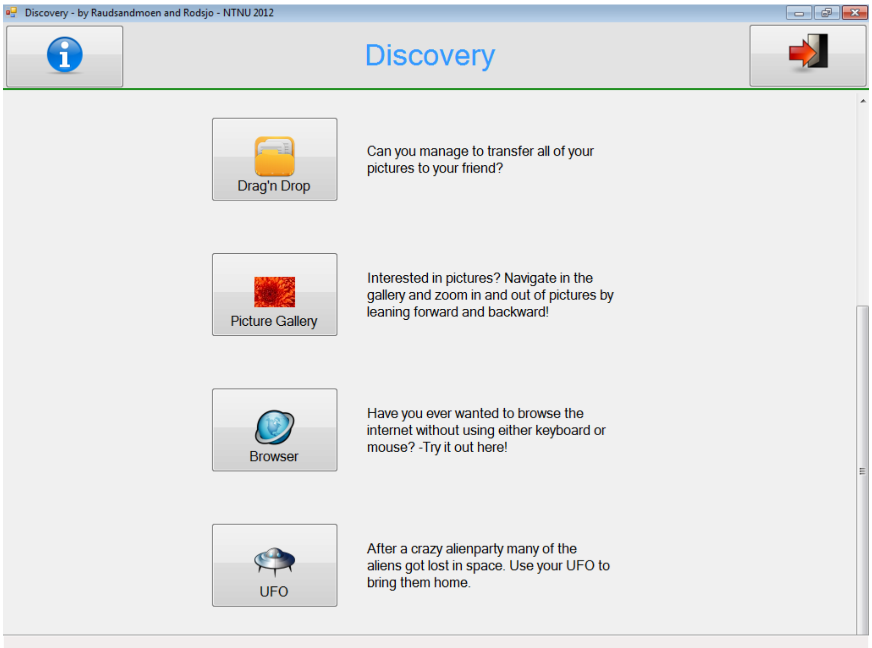
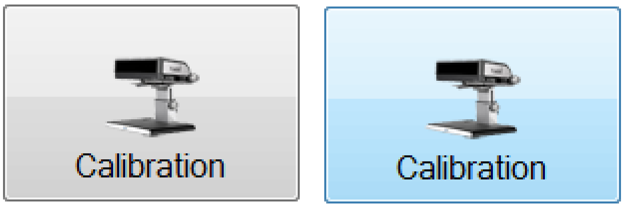


Figure 3.2: The Main Menu scrolled down

3.2.2 Interaction

Microsoft Corporation (2011b) states: *Never require users to click an object to determine if it is clickable*. The default command buttons provided from Windows are designed in order to achieve the above statement. That is why all buttons in our applications have a traditional Windows design. The menu buttons react upon the gaze just as normal windows buttons react upon the mouse; with a *hover* effect. When the user's gaze is passing over a button, the button displays a hover effect by changing its background colour into two shades of blue (see Figure 3.3). This behaviour adds click affordance to the buttons, and is a well-established interaction feedback from Microsoft Windows (Microsoft Corporation, 2011b).

As mentioned in Chapter 2.1.2, the eyes have no direct substitute for clicking as a mouse have. The most commonly used alternatives is to use dwell time, blink or a multimodal interaction style such as the gaze-keyboard combination. Using blinks to activate a click has proved to be a sub-optimal solution, due to its non-intentional nature (Lankford, 2000) (Jacob and Karn, 2003).



(a) Normal button (b) Button with hover effect

Figure 3.3: Button interaction. When a user is looking at a button, a hover effect is displayed

Using gaze in combination with a keyboard command is a solution that facilitates speed. While using an interface, the user presses a single key on the keyboard to simulate a left mouse button click. With this interaction style the user does not have to wait for the dwell limit to be reached to generate a click, but can instantly press the keyboard when looking at the desired target. Pressing the key twice quickly performs a double click; and holding the key down allows dragging of objects (Lankford, 2000).

We chose to implement button clicks by using dwell time. This means that a button click is simulated when the user has stared at a button for a certain time (until the dwell limit has been reached). We chose to implement button clicks this way based on the findings from Jacob (1990). Jacob finds that using a short dwell time, to generate object selection gives excellent results provided that the consequences of choosing the wrong object is small. If the response from the object selection is small, like displaying extra information about the object, then the user can quickly re-select the correct object.

For object selections that gives a larger response, like opening a new window, a longer dwell time is needed. Here Jacob (1990) observe that a longer dwell time than 3/4 seconds was in no case useful. This due to the fact that people do not normally fixate on one spot for that long, and also to avoid users getting the feeling of a non-responsive or crashed system.

Please note that the dwell times used during this study are presented in Chapter 4.3 - Tasks.

3.2.3 Eye Validity Feedback

In the top menu an eye validity feedback is implemented, ensuring that the user always know the status of the eye tracker. This is done by colouring the bottom line in the top menu black if the eye tracker is not connected, red if the eye tracker does not find the user's eyes, or green if the user's eyes are currently being tracked. This eye validity bar is inherited by all Discovery demo applications.

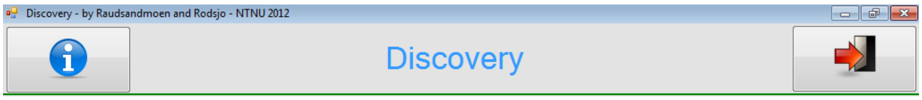


Figure 3.4: Eye validity feedback in Discovery v.2.0

3.2.4 Shortcuts

Also implemented for all demo applications are shortcuts for adjusting dwell times and enable/disable labels and feedback effects. Since Discovery is to be used as a testing tool for gaze interaction it should provide an easy way to adjust different testing parameters. This way, testers may customize their tests without having to reprogram Discovery. A label showing the current dwell time can be enabled by pressing the keyboard shortcut *Ctrl+D*. By pressing *Ctrl +* or *Ctrl -* the dwell time is adjusted up or down respectively in intervals of 250 milliseconds. To enable or disable the gaze marker (explained in Chapter 3.3.3) press *Ctrl+G*. To enable or disable hover effect (as explained in Chapter 3.2.2) press *Ctrl+H*. A summary of all implemented shortcuts is available through the main menu information button (see Figure 3.5).

3.2.5 Calibration

In order to ensure accurate eye tracking, a calibration is required. A calibration application is therefore a *must-have* when it comes to gaze interaction software. After visiting the main menu, the next place to go for a new user is the calibration application. An accurate calibration is beneficial in all the other demo applications. The Calibration Application can be accessed with both mouse and gaze interaction, and is described in detail in the next chapter.

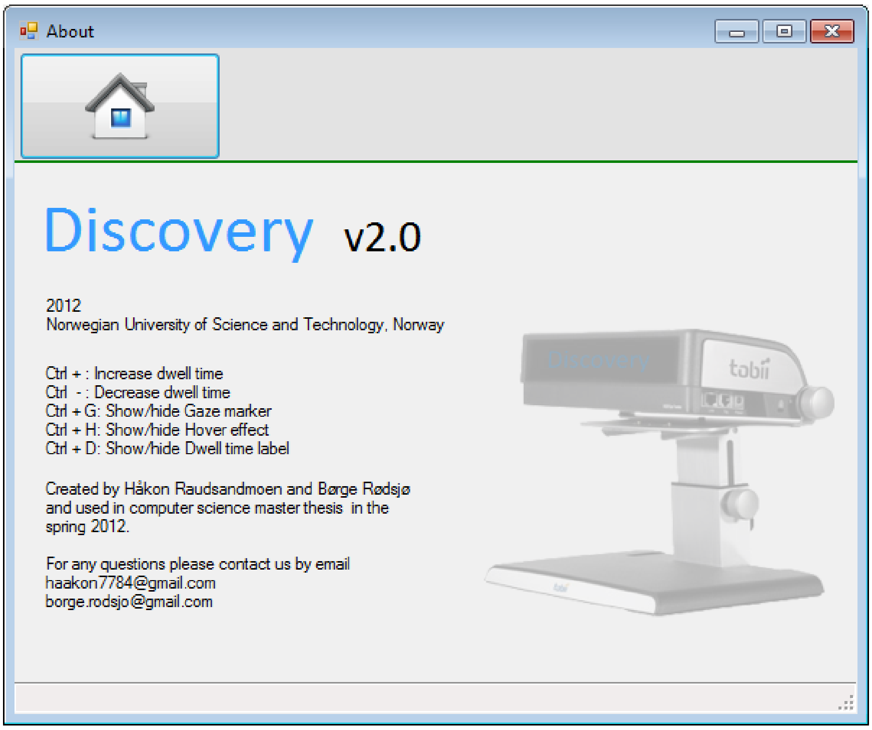


Figure 3.5: The Main Menu Information Window

3.3 The Calibration Application

The calibration application consists of a text field, a graphical representation of the user's eye position and several buttons (Figure 3.6). The text field is used to give the user relevant information and instructions on how to use the calibration application. The graphical representation of the user's eye position, called the *track status*, helps to position oneself in front of the eye tracker. The track status also contains a rectangle which is coloured the same way as the eye validity line in the top menu, as described in Chapter 3.2.3; red if the eye tracker camera is not able to find the eyes and green if the eye tracker camera finds both.

The buttons are grouped together based on what function they provide; run, save, load and view calibration are grouped together at the left hand side of the screen, while the two auxiliary functions show/hide gaze marker and calibration test are grouped together at the right hand side of the screen. To avoid unintentional interaction due to a bad calibration, these buttons only accepts mouse interaction.

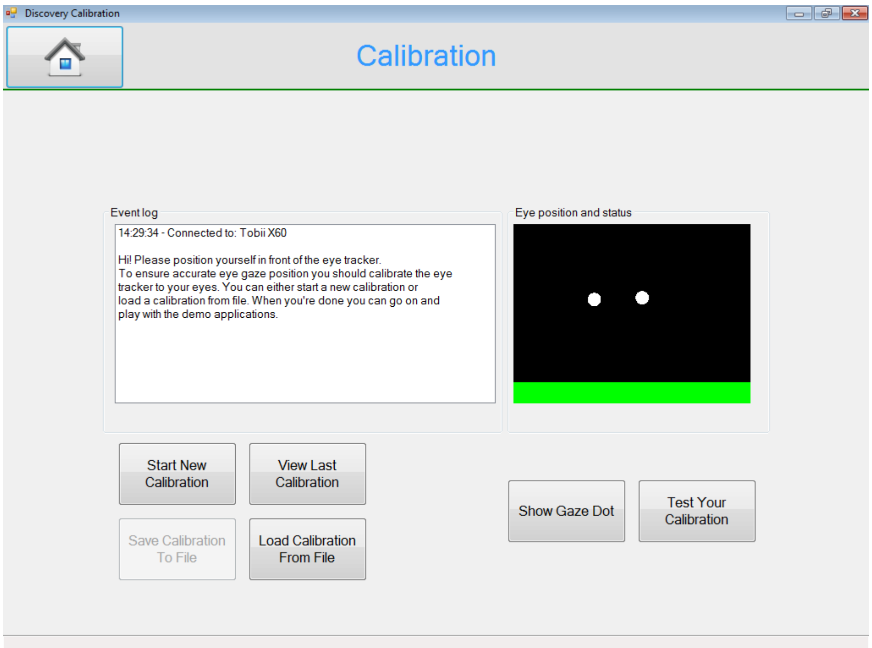


Figure 3.6: The Calibration Application

3.3.1 The Track Status

The track status helps users to position themselves in front of the eye tracker. The bounds of the track status corresponds to the bounds of the *head box* that the eye tracker is able to track. The Tobii X60 Eye Tracker allows the user to move freely inside a spatial cube with dimensions 44 x 22 x 30 cm (17 x 9 x 12 inches) (width x height x depth). Users also need to be seated within 50 - 80 cm from the eye tracker. Before users run a calibration it is recommended that they try to find a natural seating and to position their eyes in the middle of the head box. With the track status this can easily be achieved by following the graphical eye's location in the track status while adjusting the chair and seating.

3.3.2 Calibration Procedure

Current commercial eye trackers are not intrusive and requires little effort to set-up. However, there is still a need to run a calibration for each user to guarantee precise eye tracking. This calibration procedure is inspired by a sample calibration procedure from the Tobii SDK 3.0 RC 1. The user is looking at circles appearing at different positions on the screen. An offset is calculated between where the user actually looks and where their registered gaze is. It then gets sent, stored and applied in the eye tracker.

If the eye tracker is to be used by the same user in several eye tracking sessions, it is convenient to save a good calibration to file. In this way, for each of the following sessions, one can just load the stored calibration file instead of having to do a new calibration procedure each time. *Discovery* makes it easy to save and load calibration files to hard disk, with the single click of a button.

The eye tracker can only store one calibration at a time. In order to show which calibration this is, the calibration application interface has a button to view the currently active calibration. This is a graphical presentation of how the eye tracker tracked the left and right eye during the calibration procedure (Figure 3.7). A good calibration exists if all calibration points (circles) are coloured with red for the left eye and green for the right eye. If some calibration points do not contain any colour, we recommend the user to run a new calibration.

3.3.3 Gaze Marker

Just as the mouse cursor is an on-screen representation of where the user moves the mouse, we have implemented a gaze marker to visually represent

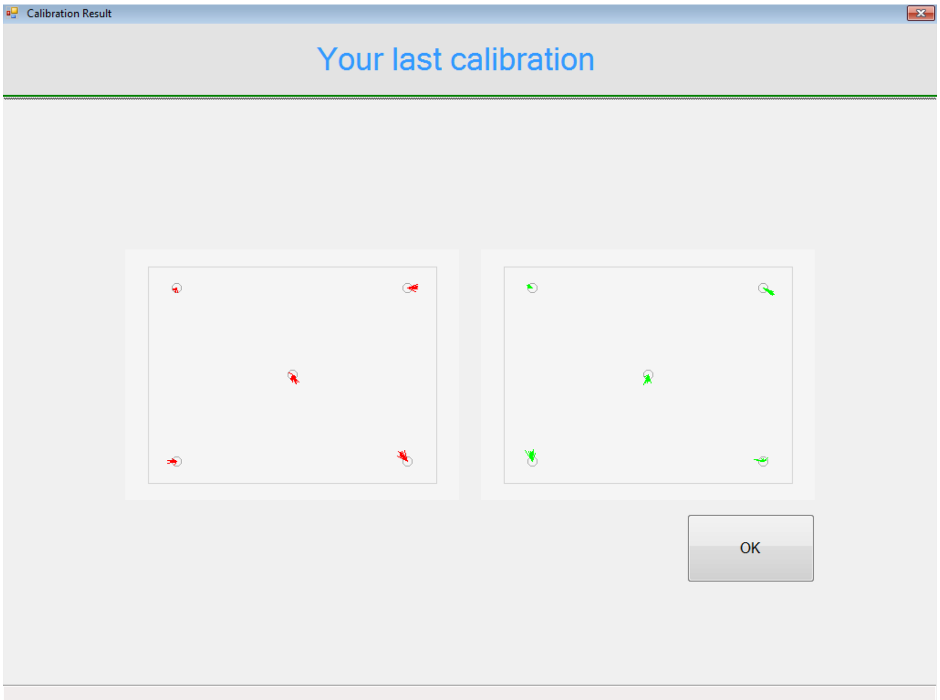


Figure 3.7: Viewing the currently applied calibration

where a user is looking. The gaze marker is a small 8 x 8 pixel square which is 30 % transparent. If a calibration goes wrong, or if a user is positioned too far away from the middle of the spatial head box, a person's actual gaze point will differ from the gaze point drawn at the screen. The gaze marker can therefore be of great help in order to understand for instance why users have a hard time hitting buttons. In addition to using the button found in the Calibration Application, the gaze marker may also be turned on and off at any time using the keyboard shortcut *Ctrl+G*. The gaze marker as a feedback method is tested in *The Drag and Drop Application* in Chapter 3.6.

3.3.4 Calibration Test

To test the calibration accuracy it is an application that tracks a user's gaze area when focusing on a given point (Figure 3.8) implemented. Even when calmly focusing on a small point, the human eye is not motionless. It moves rapidly around the point, constantly exploring. These small rapid eye movements, called *microsaccades*, varies in width, and are challenging to track for the eye tracker if the calibration is off or the user's eyes are far from the centre of the head box. If the user‘s gaze jumps too much back and forth when trying to stare at an object, it becomes hard to hit. In the calibration test the user stares at a small point for a certain amount of time, while an implemented algorithm calculates the average diameter of the eye gaze variation. In addition to presenting the numeric value, a circle with the recorded diameter is then drawn around the point. By continuously staring at the point, more circles are calculated and added (see Figure 3.8)

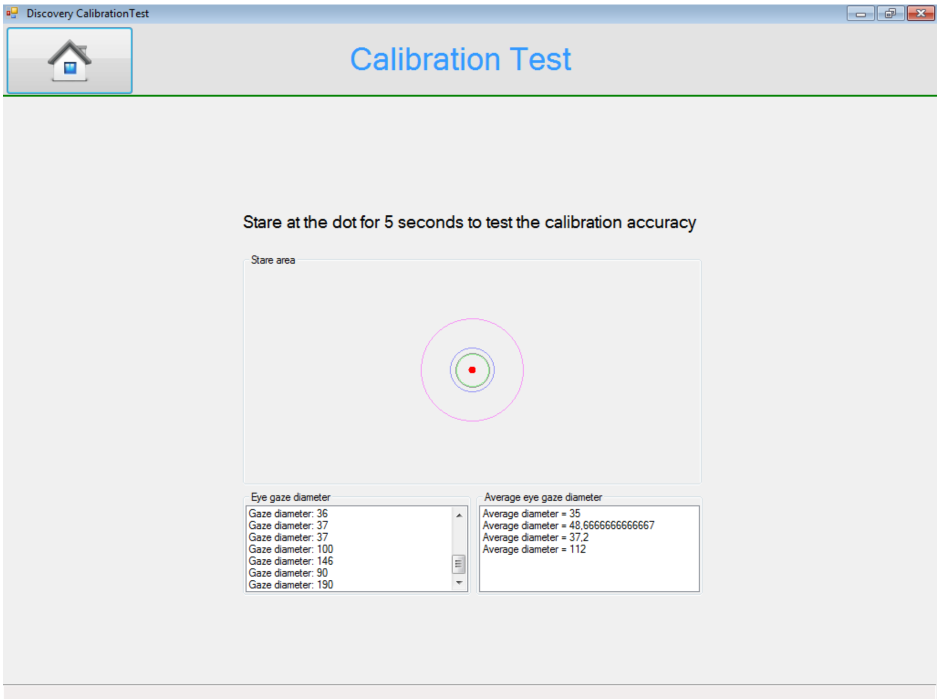


Figure 3.8: Showing the calibration test functionality for testing accuracy. In the figure displayed a user has gazed at the point for a while, so several circles have been calculated and drawn.

The calibration test application can be used to test different smoothing-algorithms on the raw eye gaze input stream. Basic eye movements may be divided into two types: *fixations* and *saccades*. A fixation occurs when the gaze rests steadily on a single point. A saccade is a fast movement of the eye between two fixations. However, even fixations are not stable and the eye jitters during fixations due to drift, tremor and involuntary micro-saccades. This gaze jitter, together with the limited accuracy of eye trackers, results in a noisy gaze signal (Kumar et al., 2008). Hence there is a need to even out this noise to effectively use eye gaze as interaction device. Different algorithms can be used (Kumar et al., 2008), we are currently using a lowpass filter on the data stream to avoid the largest peaks in user's gaze, and even out the discrete gaze events. When implementing different algorithms that affects the gaze events, the calibration test would be a valuable tool for testing the different results.

3.4 The UFO Application

The UFO Application is a video game which utilizes users' eye movements as input. Users are given the opportunity to control a traditional flying saucer, or UFO, using eyes only. The UFO follows the shortest path (flies in a straight line), to where the user is looking. The user's challenge to avoid that any dangerous objects is caught in between the UFO and the point of the gaze. The goal of the game is to navigate through different galaxies (levels), picking up lost aliens and to bring them back to their home planet safely. The galaxies are home to dangerous suns that you need to avoid, and as you proceed increase the degree of difficulty.

If you fly too close to a sun, the UFO will explode, and you have to start the game all over again. Figure 3.9 shows the GUI for The UFO Application. A user is playing the second galaxy of the game, indicated by two galaxy icons in the top menu. In the same menu it can be seen that the user has currently already picked up four aliens, and is heading for the fifth.

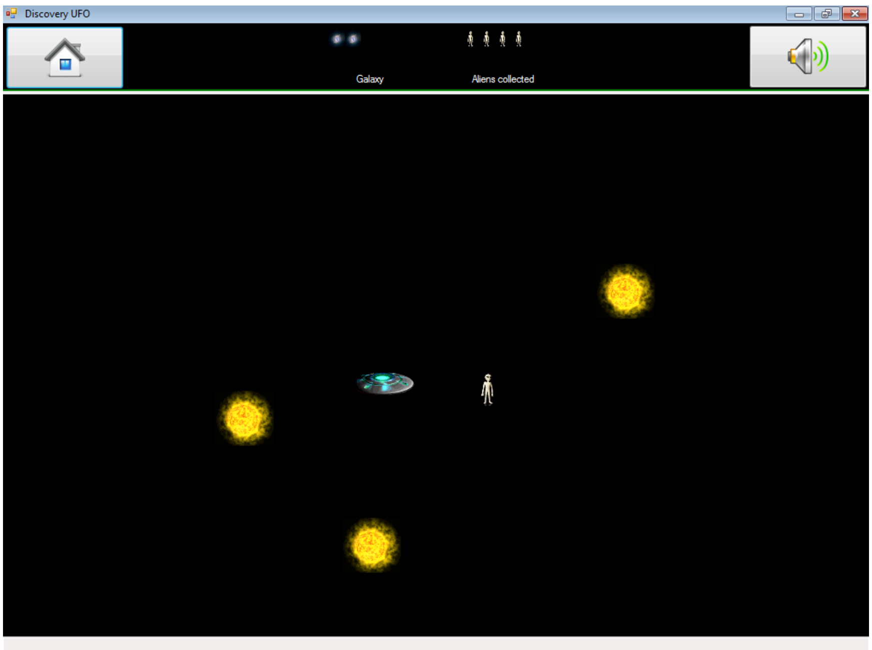


Figure 3.9: A user playing the second level of the UFO video game.

3.4.1 Graphics and Logic

The game features basic 2D graphics with applied animations. All game objects are created using the Microsoft .NET Windows Forms labels and picture boxes. A self-made build-in collision detector checks for collisions between the picture boxes, and triggers the appropriate logic.

The labels are changed and displayed when needed to give information to the user. A timer function has been created to make the labels do a count down before each level. This is done to make the user feel ready before the game starts. Five levels are implemented, each with a different (but fixed) number and types of suns.

Figure 3.10 shows The UFO Application welcome screen, including the top menu. The welcome screen provides the user with instructions on how to play the game, and describes the basic GUI elements of the first level. A hover effect on the right button in the top menu indicates that this is where the user is currently looking.

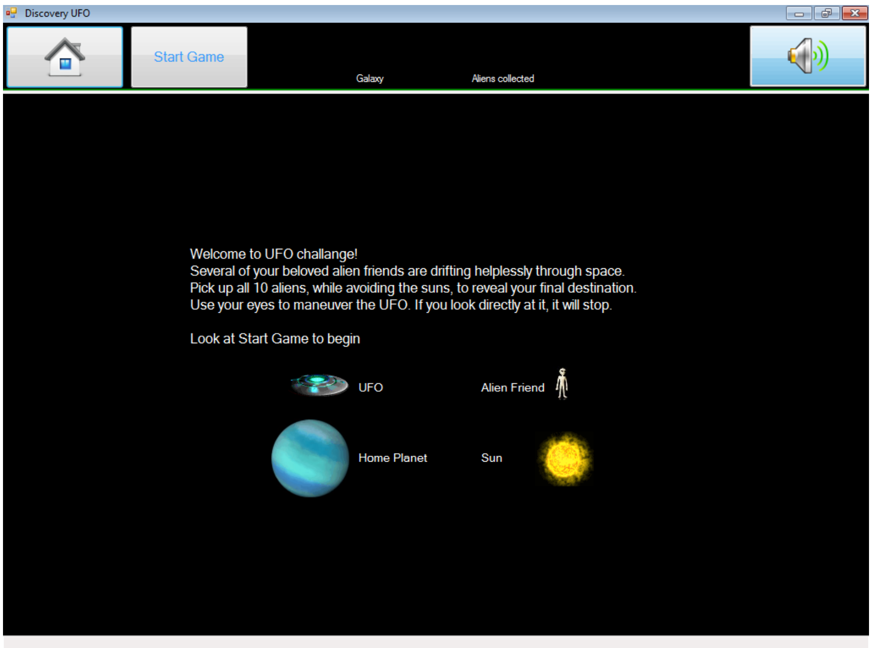
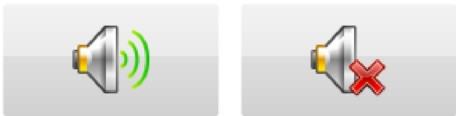


Figure 3.10: The UFO Application welcome screen provides instructions about how to play the game. The user is currently looking at the mute button, causing a hover effect

For more details about this application's graphics or logic, please see the source code attached the delivery of this project.

3.4.2 Sound

The game features basic sound effects, when the UFO is picking up aliens or crashing into a sun and exploding. All sounds are implemented with the Microsoft .NET framework Windows Forms MediaPlayer class. A gaze-controlled button in the top menu allows the user to turn the sound effects on and off. The button uses dwell time for click interaction and changes its icon when pressed (shown in Figure 3.11a and 3.11b).



(a) Sound enabled (b) Sound disabled

Figure 3.11: The UFO Application sound effect button.

3.4.3 Gaze Rest Area and Pause Function

In addition to holding all buttons, the application's top menu works as one giant pause button. As soon as the user's gaze leaves the main game window, the game pauses (but not the animations). This way a user may always take a look at the progression and interact with the buttons without the UFO in the meanwhile crashing into a sun. Furthermore, it works as a gaze rest area. If the user is tired or needs a little break, one may either look to the top menu, or just look away from the screen. The game will then pause.

3.5 The Picture Gallery Application

This application provides a picture gallery designed for gaze interaction. It utilizes both gaze input and head movements. It shows one picture at a time with the associated name and allows you to browse back and forth throughout the pictures by looking at two gaze controlled command buttons. One navigating to the next and one to the previous picture. Both buttons are 90 px in width and 280 px in height. A screen shot showing the application can be seen in Figure 3.12 below.

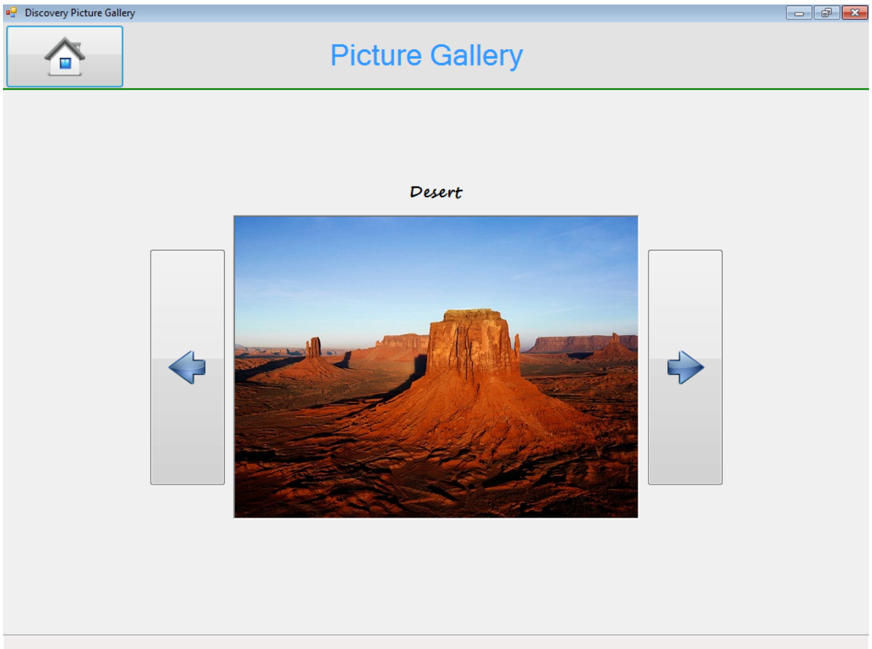


Figure 3.12: The Picture Gallery Application.

The first time users enter The Picture Gallery Application an *instruction window* is displayed. It informs the user of the implemented zoom feature, which allows users to zoom in and out of pictures by leaning towards and away from the computer screen. Figure 3.13 shows a screen shot of this instruction window.

The inspiration for this functionality comes from what one does in real life when wanting to enlarge and shrink objects of interest. When wanting to inspect an object that is too distant, one moves towards the object thus bringing it closer. Similarly, if one stand in front of a large poster one may sometimes need to step back to see the whole picture. These natural movements is what have been tried captured with the zoom functionality

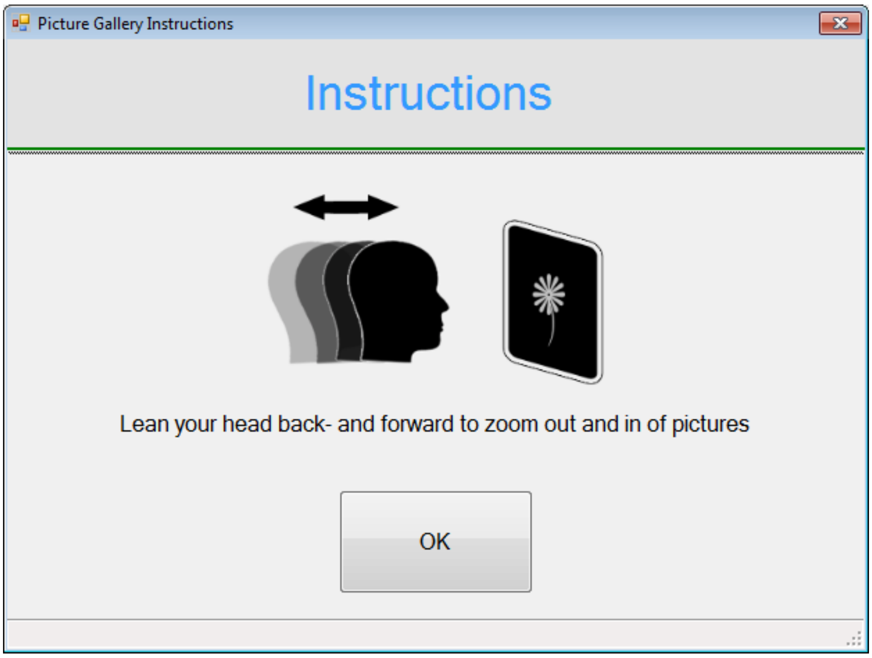


Figure 3.13: The Picture Gallery instruction window.

in Discovery. By leaning forward the picture is enlarged, while leaning backwards makes the picture smaller. This *lean* gesture has been made possible by using the eye tracker's eye 3D position data to continuously check the distance to users' eyes. To the left in Figure 3.14 we see a user looking at a picture of a Koala. To the right the same user leans forward and initiates the zoom function.

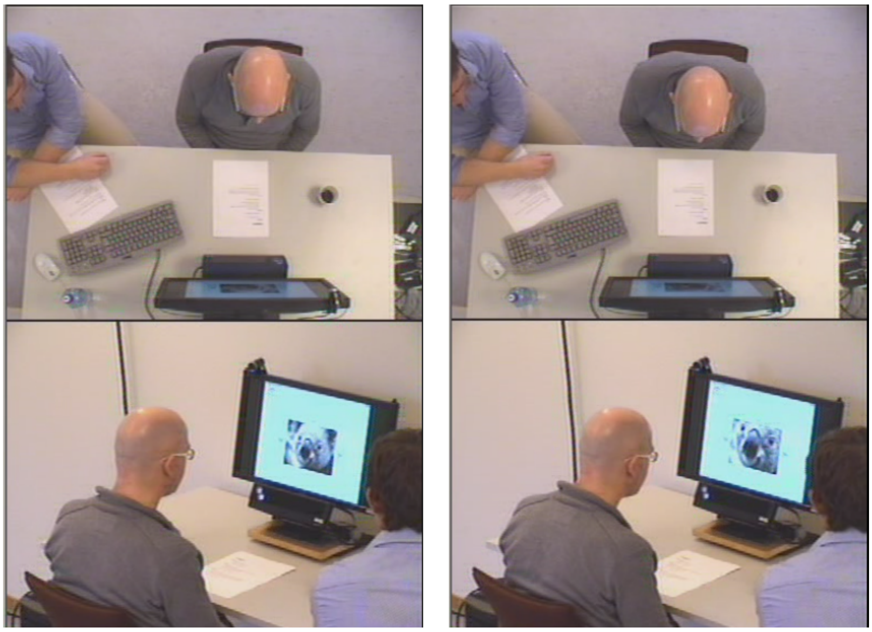


Figure 3.14: A user looking at picture of a Koala in The Picture Gallery Application (to the left), and by leaning forward he zooms in on the nose of the Koala (to the right).

3.6 The Drag and Drop Application

This application provides a basis for performing the *drag and drop operation* with gaze interaction. The drag and drop operation is well known, and widely used in both mouse and touch interfaces. It can be described as the action of selecting a virtual object by grabbing it, dragging it to a different location, and release it. The Drag and Drop Application enables users to pick-up, move and release files using eyes only. The surroundings is set in a typical Windows explorer window, with two file trees. The trees consists of a root node (folder), with several movable child nodes (pictures).

3.6.1 Feedback

Figure 3.15 shows the application. The small grey dot in the middle is a gaze marker. It visualizes actions, and provides feedback to the user on what is going on (explained in Chapter 3.3.3).

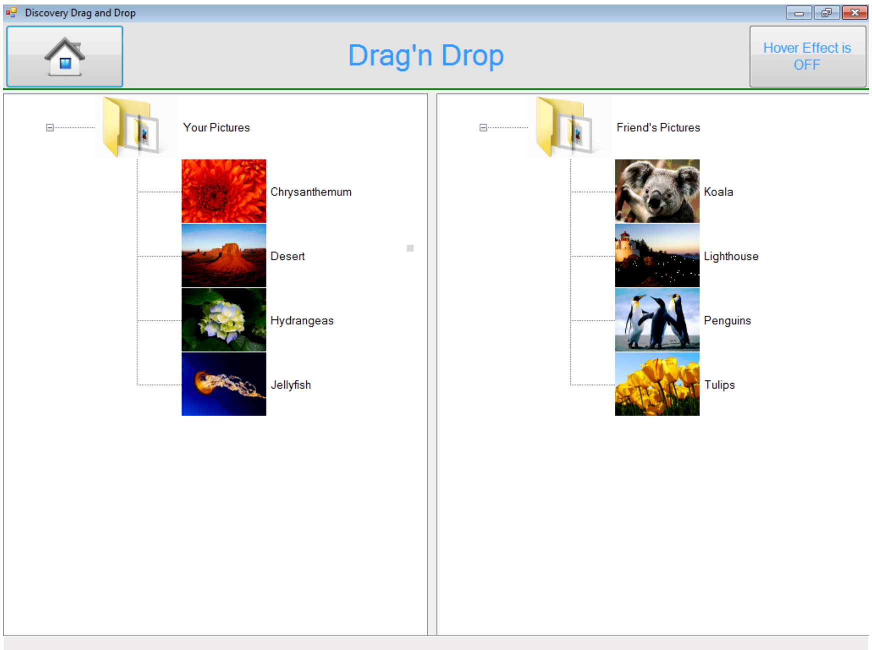


Figure 3.15: The Drag and Drop Application interface with file trees and the gaze marker.

As an alternative to the *gaze marker feedback* a *hover effect* is implemented. When a user is looking at a control and the hover effect is enabled, the background colour of the control will change. The hover effect varies in size and colour depending on the size and type of control the user is looking at. Figure 3.16 shows the hover effect feedback applied to a picture.

A gaze controlled command button in the top right corner of the application gives users the opportunity to switch between these two feedback mechanisms.

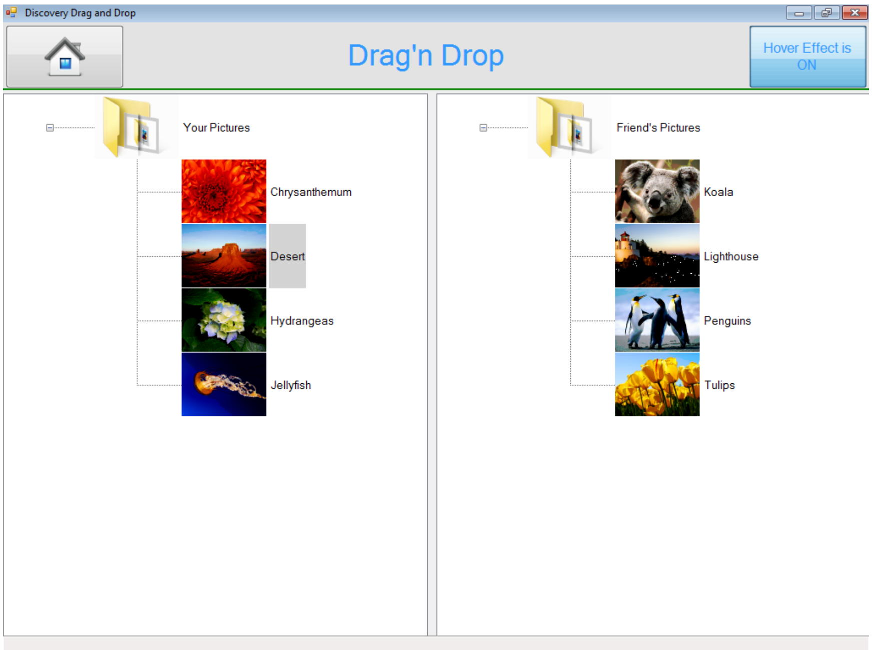


Figure 3.16: The Drag and Drop Application interface with the hover effect enabled. The user is here gazing at a picture, and the picture's text background is changed to grey to visualize that it is the selected picture.

3.6.2 Pick-up

Independent of the feedback method enabled, the implemented pick-up functionality enables the user to *grab* an item by staring at it for a given time interval (dwell time). When grabbed, a transparent copy of that item is attached to the gaze marker, while the original item remains in the tree (as illustrated in Figure 3.17).

This is intended to give the user a visual feedback of *dragging*, while no actual relocation is done before the item is released. If an item already is picked up, it can be replaced by staring at another item from the same tree. This replacement feature is meant to provide *forgiveness* in the application. Forgiveness is the ability to easily reverse or correct an undesired action,

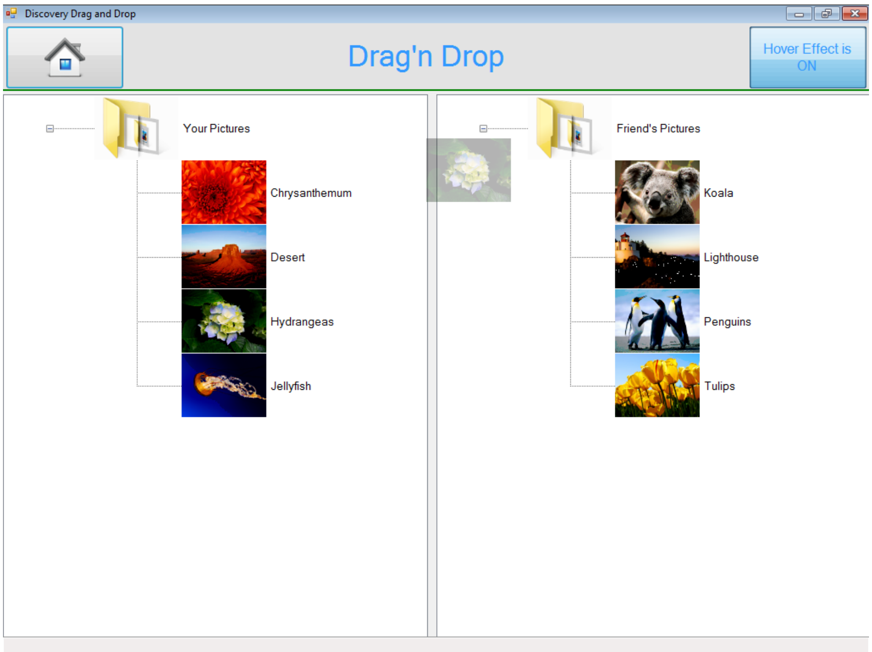


Figure 3.17: A user performing drag and drop with gaze interaction. The *Hydrangeas* picture is currently being *dragged* towards a folder named *Friend's Pictures*.

and as stated in the Microsoft Corporation (2011b), it is very important. If a user misses the element wanted to pick up, it is very easy to swap this element with another.

3.6.3 Move and Release

As soon as an item is picked-up, it follows the gaze movements until it is either replaced or released. When the user has picked up and is *dragging* an item, it can be released by staring at a valid destination for a given time interval (dwell time). Valid destinations in this case being a destination tree (root or child nodes) or a menu button. If released over a root node (folder) it will be added to the end of its child nodes, but if released over a child node it will be added directly below that particular child node. In both cases the item is moved from its origin tree to the destination tree. However, if the release action is executed over a menu button, the move action is cancelled.

3.7 The Browser Application

Surfing the Internet and reading on-screen text has become a common task in our everyday computing. When reading on-screen and navigating either up and down or side to side is what we call *scrolling*. Scrolling is normally activated by using either keyboard, mouse, mouse wheel or touch. The Browser Application provides a possibility to test gaze interaction in conjunction with screen based reading and scrolling. It consists of a gaze controlled web browser, and as the other applications; a top menu. In addition to the home button, the main menu, the top menu is designed to hold web browser navigation buttons. The navigation buttons give users the opportunity to go back- and forward in navigation history as well as navigating to other web pages.

3.7.1 Active Scrolling

To provide the user with the possibility to actively scroll within a web page, several predefined invisible scroll areas has been implemented. The scroll areas at the bottom and upper part of the browser enables scrolling up and down respectively when looked upon. If the web page is wider than the window displayed in, the application also supports scrolling horizontally. If both horizontal and vertical scrolling is enabled, the user can scroll diagonally by looking in the corners of the scrollable areas (see Figure 3.18).

If a user is continuously gazing at a scroll area, the scroll speed increases. This provides a faster way for a user to reach the desired target. A lot of web surfing consists of scanning through web pages to find desired information. Such scanning is traditionally done by scrolling the mouse wheel with different speeds as the user wants to go faster or slower when searching. With gaze interaction and The Browser Application this is done automatically. The implemented solution increases the scroll speed linearly as the user continues to look in a scroll area. This will be described further in Chapter 4.3.4.

3.7.2 Automatic Scrolling

In addition to support active scrolling, an area with slow scroll speed to support passive (automatic) scrolling is implemented. This area is called a *read area* and is positioned just above the bottom scroll area (see Figure 3.19). When a user's gaze touches this area of the screen a slow downward scroll is initiated which continuously scrolls the page while the user reads. The goal is to provide a even scrolling, that does not distract the user

Figure 3.18: The predefined scroll areas for scrolling up, down, sideways and diagonally.

while reading. The text is automatically scrolled, so that the user may read continuously without interruption and having to scroll actively. By fine tuning the scroll speed the application should scroll one line down in the same time taken for a user to read it. If the user looks away from the scrolling area the scrolling immediately stops.

Figure 3.19: The predefined read area for triggering automatic scrolling while reading

3.8 The Controls Application

The Controls Application is all about Windows controls (defined in Chapter 2.3.2). It is divided into four screens which each are customized for testing a specific control. The screens contain multiple instances of the control with different variations, all accepting gaze interaction. One can cycle through the different control screens by looking at two gaze controlled command buttons at the bottom of each screen. Similar to the command buttons in the picture gallery, there is one for navigating from one to the next as well as one for navigating back to the previous screen (see Figure 3.20). The controls covered in this application are:

- Command Buttons
- Links
- Check Boxes
- Sliders

The following subchapters will describe the four screens dedicated to these controls.

3.8.1 Command Buttons

In the command buttons screen, six command buttons with different sizes and placing are implemented (see Figure 3.20). All buttons react upon gaze input. If a user is staring at a button, a hover effect applies (similar to the one described in Chapter 3.6.1). If the user continues to stare, a click is generated after a given dwell time. The click triggers a change in the button's background colour, for immediate user feedback. The background colour cycles between red and green for each generated click (an example of this is shown in Chapter 4.3.5). The implemented button dimensions are posted in Table 3.1. As seen from Table 3.1, Button 5 is created according to the Microsoft Windows design standards (Microsoft Corporation (2011b)). That is, the recommended button size for Windows Vista and Windows 7.

The buttons are placed such that every button has equal distance to the centre of the screen. This to prevent any possible inaccuracies from the eye tracker, for instance drift or gaze offset. Remote video based eye trackers have higher accuracy in the centre of the view field of the camera. Thus the tracking is most accurate at the centre of the screen (Beinhauer, 2006). With this button placement each button should be equally easy to hit even though if drift or gaze offset have appeared.

Command Button	**Height**	**Width**
Button 1	80px	150px
Button 2	65px	125px
Button 3	50px	100px
Button 4	35px	75px
Button 5 (MS standard)	23px	75px
Button 6	19px	50px

Table 3.1: Command Button sizing

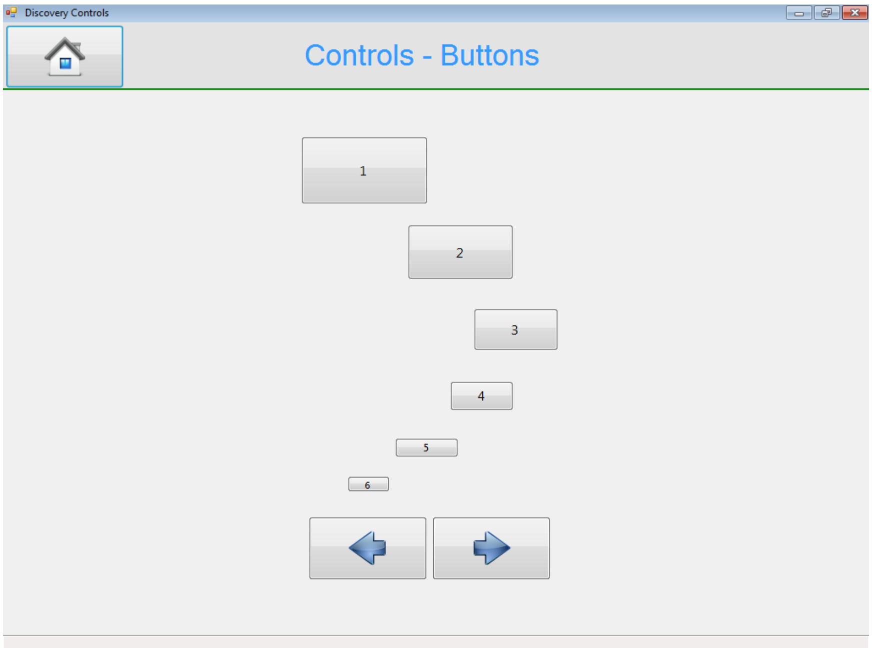


Figure 3.20: Screen shot showing the Buttons screen in The Controls Application.

3.8.2 Links

In the Links screen, a set of gaze interactive links are implemented (see Figure 3.21). A stare longer than the dwell time generates a click which opens a new window. This new window will normally be a web browser, but is in this case and for testing purposes just an information window. It simply informs the user that the link was successfully clicked, and asks them to click *OK* to continue. By staring at the gaze controlled command button named *OK*, the user returns to the Links screen. The links differ in font sizes to support link size testing. The font sizes implemented can be seen in Table 3.2. The *yr.no* link is created following to the Microsoft Windows

Link	Size
storm.no	21pt
google.no	17pt
ntnu.no	13pt
yr.no (MS standard)	9 pt

Table 3.2: Font sizes for different links.

font size recommendation of 9 pt (Microsoft Corporation (2011b)). All links uses the recommended Microsoft Windows link font type *Segoe UI* (Microsoft Corporation, 2011b). As seen from Figure 3.21, the links are presented in two rows. This is done for the testing of tooltips in links. For most websites it is common practice to display a tooltip (a small text box) when hovering over a link with the mouse. To test how this will affect the gaze interaction style, one of the rows makes use of such tooltips, the other just normal underlining. That is, when a user is looking at a link, the link text gets underlined thus providing an immediate user feedback.

3.8.3 Check Boxes

Sixteen gaze controlled check boxes have been implemented in the Check Box screen (shown in Figure 3.22). Both the boxes themselves and their associated texts accepts gaze input. This is done by defining a hit area that entails the whole check box control (including the text). The inside of the hit area accepts input, the outside do not. If a user stare at a hit area for a given time interval, the check box becomes selected (checked).

The check boxes are grouped together in four groups according to their design properties. Each group has been given a topic to suit creation of understandable user tasks. Instead of having abstract texts like *Option 1*, *Option 2*, *Option 3* etc., the check boxes labels were given common understandable terms such as *Cars*, *Shapes*, *Seasons* and *Colours*. The differences in design properties concerns spacing and sizing (height and width of a check box's hit area). There are exists two types of spacing, actual and visual. Actual spacing is the actual vertical distance between two check boxes' hit areas - the space that do not accept any input. The visual spacing is the visual vertical distance between to check boxes as experienced by the user. Figure 3.23 illustrates these two kinds of spacing.

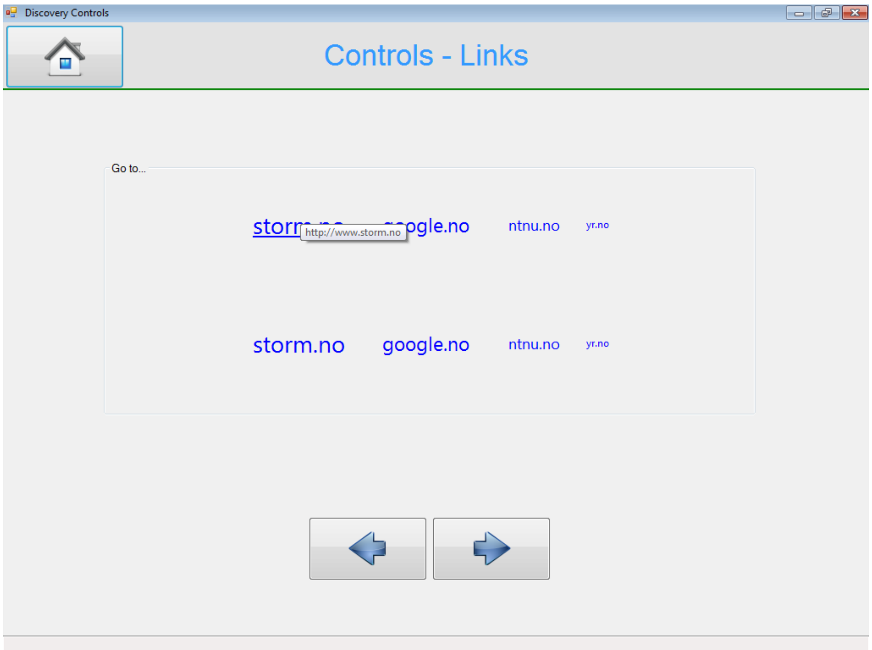


Figure 3.21: Screen shot showing the Links screen in The Controls Application. A user is currently looking at the storm.no link, thus displaying a tooltip. The bottom row of links does not show these tooltips when hovering.

Please note that the testing focuses on the actual spacing, and therefore may be referred to as just *spacing*.

For testing purposes each group is made similar to another group with the exception of one different design property. For example, *Cars* and *Shapes* have the same actual spacing, but different sizing. This enables testing of the sizing property's influence. Similar do *Seasons* and *Colours* have the same sizing but different spacing, for testing of the spacing property. All group design properties are displayed below, in Table 3.3. The *Colours* group is constructed according to Microsoft's check box recommendations for Windows Vista and Windows 7 (Microsoft Corporation (2011b)).

3.8.4 Sliders

The last screen in The Controls Application is the Sliders screen. It holds four gaze controlled sliders. When looking at a slider, it will follow ones horizontal gaze movements. However, all the sliders have different properties and will act differently.

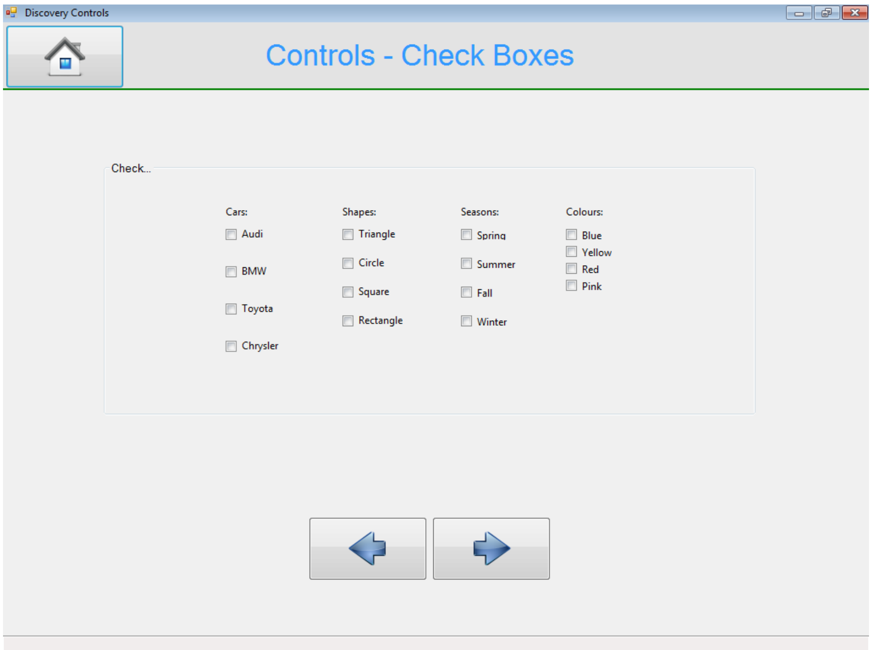


Figure 3.22: Screen shot showing the Check boxes screen in The Controls Application.

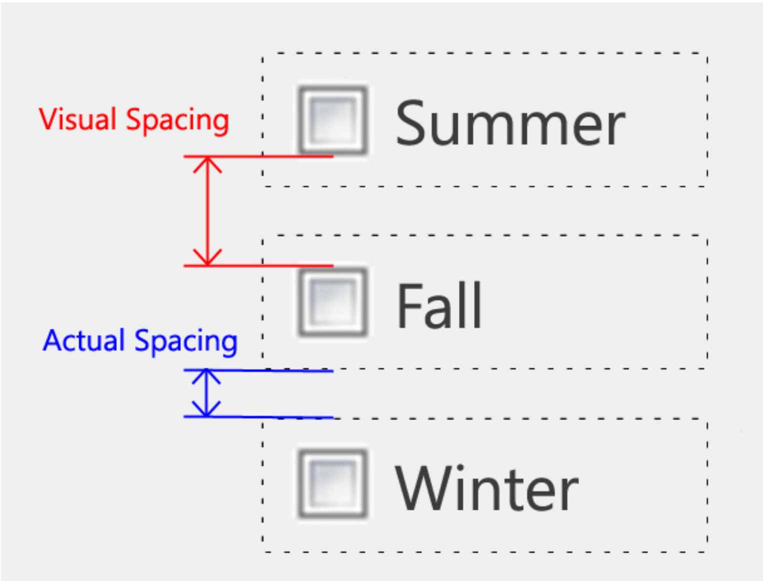


Figure 3.23: Showing the spacing properties of a check box group.

The most important properties are height, labelling and delayed func-

Group	Height	Width	Actual spacing
Cars	40px	83px	4 px
Shapes	30px	83px	4 px
Seasons	17px	83px	17px
Colours (MS standard)	17px	83px	3 px

Table 3.3: Check Boxes group properties

tionality. The height represents how high the slider's hit area is, as with the check boxes. The labelling deals with the text representing a sliders value. It could be either single og multi. *Single* imply that one single value label is centred below the slider, which dynamically updates itself when the slider moves from one value to another. *Multi* imply that multiple static value labels are used, one for each tick mark.

The delay functionality makes the slider wait a predefined time interval before moving (dwell time). When the time is elapsed, it moves one tick and waits another interval. This will prevent a slider from instantly jumping to a value when briefly looked upon.

Which properties that are implemented with each of the sliders is presented in Table 3.4. To support testing, each of the sliders have been constructed similar to another with one alternating property. Slider 1 and 2 are the same apart from different labelling, 2 and 3 the same except different delay functionality and 3 and 4 the same apart from different height. Also note that Slider 1 - *Year* is constructed according to Microsoft's slider recommendations for Windows Vista and Windows 7 (Microsoft Corporation (2011b)). The others are variations constructed for slider testing purposes. As with the check boxes, the sliders have been given namex to ensure un-

Slider	Name	Height	Labelling	Delay
Slider 1	Year (MS standard)	45px	Single	No
Slider 2	Month	45px	Multi	No
Slider 3	Date	45px	Multi	Delayed
Slider 4	Weekday	75px	Multi	Delayed

Table 3.4: Slider properties

derstandable user tasks. They include *Year*, *Month*, *Date* and *Weekday* (see Figure 3.24).

Figure 3.24: Screen shot showing the Sliders screen in The Controls Application.

Chapter 4

Method

This chapter describes the test method of this study. It starts by explaining how we planned to answer our research questions. It then continues with a description of the user test participants, the tasks given, the test facility and equipment, the data gathering methods and test measures used as well as our test procedure. Finally, it is explained how we intend to suggest design guidelines based on the results.

4.1 Method Introduction

As explained in Chapter 1.2, the following research questions should be answered in order to achieve our research goal.

RQ1: *How do users assess gaze interaction for solving a set of common everyday computer tasks?*

RQ2: *How does gaze interaction perform when used to solve a set of common everyday computer tasks?*

RQ3: *Which design guidelines can be suggested for gaze interaction in Windows 7, based on performance and user assessments?*

In order to answer the first and second research question, we will conduct a user test. As user testing allows for both qualitative and quantitative data gathering (explained in Chapter 2.4), it is an appropriate method when wanting to both measure performance and gather user assessments. The process of collecting user assessments will be a series of *post-task semi-structured interviews*. This is done by asking a set of pre-defined questions after each task, and let the participants answer freely. By recording and writing down the answers the desired data material is acquired. By registering the completion rate of the different tasks and subtasks, it will be possible to calculate how gaze interaction performs in each situation.

An analysis and discussion of the user test results will be carried out to answer the third research question. The suggested design guidelines for gaze interaction should be based upon the evidences from this process. They will be based on both the user assessments, and the gaze interaction performance. This process will be more explicitly explained in Chapter 4.9.

Based on the recommendations described in Chapter 2.4, two pilot tests as well as several small technical tests will be conducted before one start the actual user test. The purpose of these tests will be to discover flaws in the test procedure, equipment and prototype, and will not be documented with the results from the actual user test.

The user testing in this study diverts somewhat from a standard usability test (see Chapter 1.3). It will however be provided enough information regarding the method for other testers to be able to replicate the procedure. This will be assured by using ISO/IEC 25062 (2006) as a check list for the method documentation. It includes; participants, context of product use in the test, experimental design and usability metrics. See Chapter 2.4 for more information about what should be included in a usability test report.

The upcoming chapters will describe our research method in detail.

4.2 Participants

The following three user groups were created for this user testing,

1. Youths (aged 10-19)
2. Young adults (aged 20-29)
3. Adults (aged 30-65)

In all, fifteen participants were recruited, five for each user group. With seven of the participants being males and eight being females, an equal sex distribution was acquired.

It was emphasized that the participant selection should represent a wide range of everyday computer users. All participants were therefore required to have some basic experience with the use of computers and internet. Through a pre-test survey (Appendix D), their level of experience was mapped. From this survey it was found that all of the participants both had access to computer and internet at home. This in contrast to the Norwegian national average, where according to Statistisk Sentralbyrå (2012), 91 % have a computer at home, and 92 % of these have access to the internet. Further, it was found that the participants, in average, are somewhat more experienced than the national average. This without considering the difference in access to a computer and the internet mentioned before.

Figure 4.1 illustrates what the participants use their computer for (blue line) compared to the national average (red line). The average difference between these results are 19 %.

Likewise Figure 4.2 illustrates what the participants use internet for (blue line) compared to the national average users (red line). The average difference between these results are 12 %.

As a conclusion, the user test participants are all familiar with the use of computers and the internet. Despite varying experience levels, it must be taken into consideration that they in average are *more* experienced with computer and internet usage than the national average.

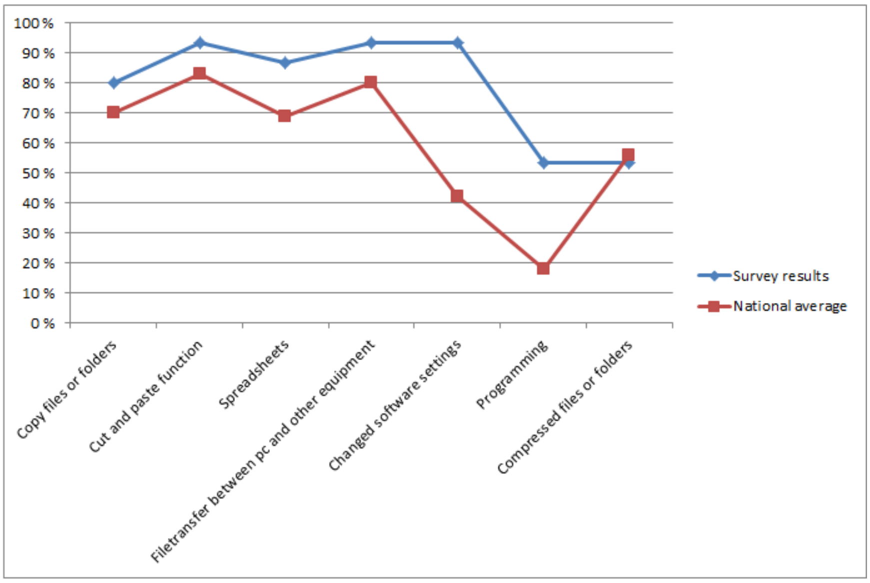


Figure 4.1: Participants average computer usage (blue), compared to the national average (red).

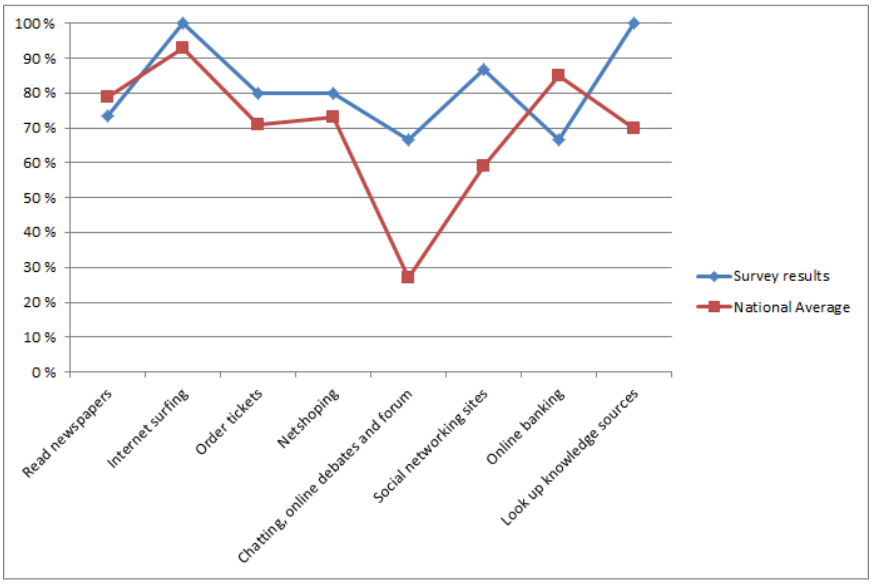


Figure 4.2: Participants average internet usages (blue) compared to the national average (red).

4.3 Tasks

The participants were asked to solve five main tasks during the user test. Each task was adapted to a different part of the Discovery software (as described in Chapter 3.1), and was to be solved using a certain demo application. These applications were made available through the main menu in Discovery. The dwell time for all menu buttons (described in Chapter 3.2.2) was set to 1,5 seconds.

This section describes each of the user test tasks with their purposes, details and connection to the Discovery software. The results from each task are presented in Chapters 5.1 - 5.5. For a representation of the actual task sheet that was given to the participants, please see Appendix E. Note that all tasks have been translated from Norwegian to English.

4.3.1 Task 1 - Playing a Video Game

a) Open the game UFO
b) Play through the first level
Go back to the main menu

Task 1 was created as a warm-up exercise for the participants. This being their first time using gaze interaction, the task's purpose was to get them accustomed to controlling the computer with their eyes. For this The UFO Application (se Chapter 3.4) was chosen.

Starting from the main menu the participant had to open up the game and play through the first level of the game. The task was estimated to take approximately two minutes, and the success criteria for completing the task is to play through the first level of the game.

4.3.2 Task 2 - Exploring a Picture Gallery

a) Open Picture Gallery
b) Go through the pictures until you find a picture of a koala
c) Zoom in on the nose of the koala
d) Find the picture of the penguins
e) Find out what the penguins are talking about
Go back to the main menu

In Task 2 the participants are asked to explore a picture gallery through The Picture Gallery Application. This implies to navigate to specific pictures and perform zoom operations to solve the tasks. Its purpose is to test gaze picture navigation and zooming with the use of head gestures. In Figure 4.3 below we see a participant that have solved subtask *e)* (found out what the penguins are talking about).

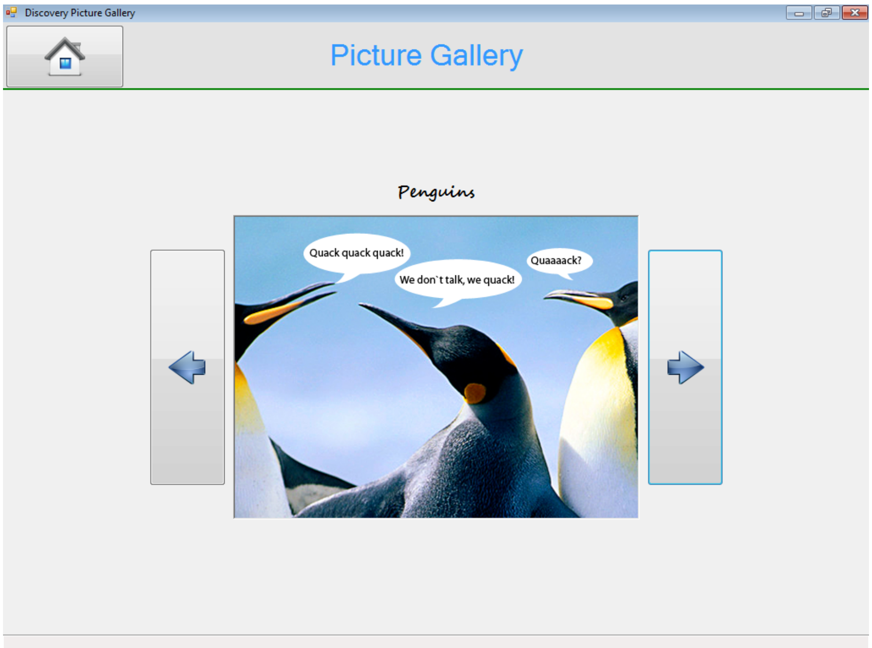


Figure 4.3: Participant solving Task 2e) - Find out what the penguins are talking about.

When starting the application, a pop-up window with instructions appear. These instructions explain how you can zoom in and out of pictures by moving your head for- and backwards towards the screen. The main focus of the task is to allow testing of the use of these head movements, and it was estimated to take approximately three and a half minute. The dwell time for the navigation buttons was set to 750 ms and the success criteria for completing the task is to successfully do all subtasks.

4.3.3 Task 3 - Doing Drag and Drop Operations

a) Open Drag'n Drop

b) Move the picture Desert over to the right side among the other pictures
c) Move the picture Koala over to the left side among the other pictures
d) Turn on Hover Effect and move Desert back to the left side
e) Move the picture Koala back to the right side
Go back to the main menu

Task 3 is about doing gaze drag and drop operations. In addition to testing the pick up, move and release operation, it tests what kind of feedback the participants prefer. A button in Discovery‘s drag and drop application can be toggled between *gaze marker* feedback and *hover effect* feedback (see Chapter 3.6 for details). Figure 4.4 shows a participant moving the picture *Desert* (subtask *d)*) back to the left side with the hover effect enabled.

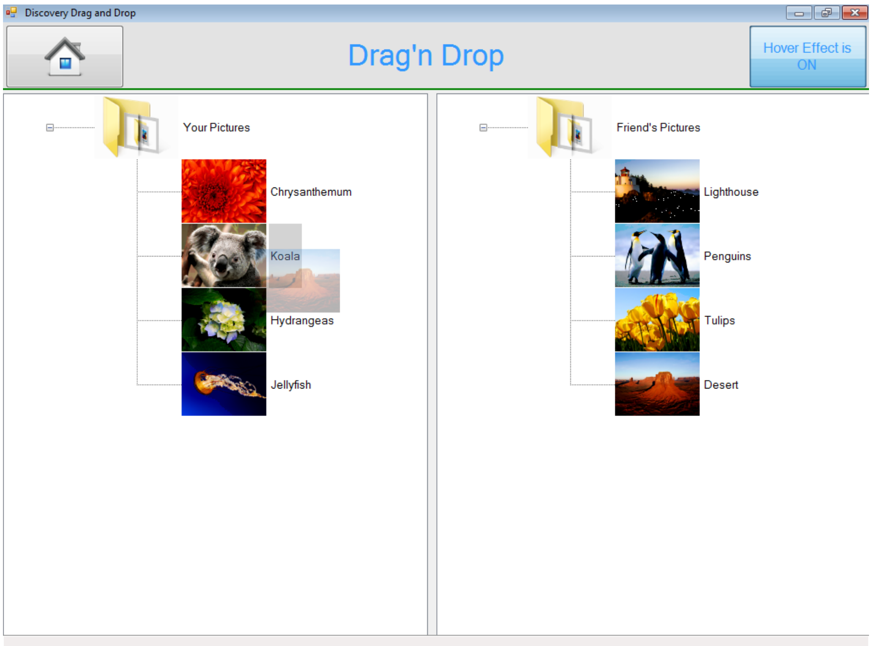


Figure 4.4: Participant solving Task 3 - Doing drag and drop operations.

The participants are given time to do the same operations with both of these feedback methods, and are then asked to give their opinion about which they preferred. In total, three minutes were estimated for this task and both the pick up time and the release time were set to 2,5 seconds. The success criteria for this task is to drag and drop at least two pictures to the opposite side, and use both gaze marker and hover effect feedback.

4.3.4 Task 4 - Browsing a Web Page

a) Open Browser
b) Read the first paragraph about Viking Age to Middle Age
c) Scroll down to Democratic constitution in 1814 and read the first paragraph
d) Scroll up to Decadence and Time of Danish Rule, and read the second paragraph about the 400-year's night

Task 4 uses Discovery's Browser Application with the web history navigation buttons and web page buttons disabled. The buttons are disabled to prevent unnecessary distractions for the participants during the testing. Additionally, a customized website has been created for this purpose. It displays an article about Norwegian history. The length of each paragraph is adjusted to best suit the different subtasks. Pictures are inserted to measure how they affect the user experience.

The two main purposes of Task 4 are to test active as well as passive scrolling (described in Chapter 3.7). The height of the scrolling areas are 15 % of the height of the web browser. With a screen resolution of 1024 x 768 pixels, and a web browser height at 728 pixels, the active scroll area becomes 109 pixels high. The start scroll speed is set to be 240 pixels each second (4 pixels 60 times a second), and if the user continues to scroll the speed is increased by 1 pixel each 500 ms. Maximum scroll speed is set to be 10 pixels 60 times a second, or in other words; 600 pixels each second. Using a monitor with 800 x 600 resolution, one may scroll one page per a second when the maximum scroll speed is reached.

When it comes to the read area, the height is set to 10 % of the browser's height, and as wide as the web page's breadth. With a screen resolution of 1024 x 768 pixels, and a web browser height at 728 pixels, this read scroll area becomes 72 pixels high. The automatic scroll speed is set to be 20 pixels down every second (1 pixel each 50 millisecond).

In Figure 4.5 a participant solving subtask *b)*, using the automatic scrolling functionality, is displayed. The subtasks of this task were constructed in such a way that the participants had to scroll both up and down between several reading sessions. The participants are encouraged to try reading both inside themselves and out loud, to test the self-adjusting scrolling speed. The test leader should also explain the active scrolling function in advance of the task.

The task was estimated to require approximately five and a half minutes, considerably longer than the previous tasks. This in order to make

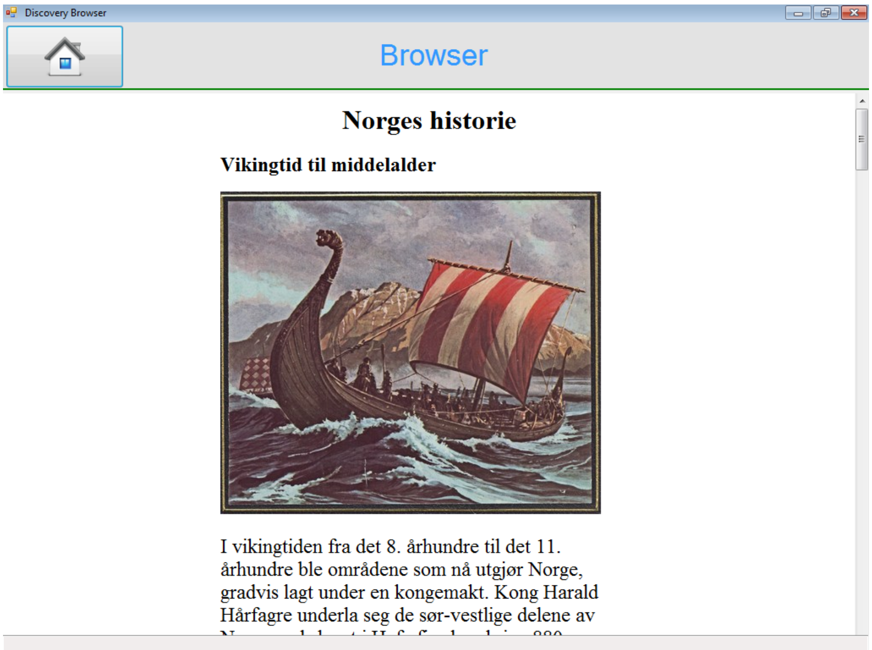


Figure 4.5: Participant solving Task 4 - Browsing a Web Page.

it possible to detect any changes in the participants opinions as they got accustomed to using the gaze operated scrolling mechanisms. The success criteria for the task was to read at least two paragraphs with the automatic scrolling, and actively scroll between them.

4.3.5 Task 5 - Interaction With Different Controls

a) Open Controls

b) Press the buttons until they become green

c) Press Next

d) Press the links, start with the largest one on the top row, storm.no

e) Press Next

f) Start from left and select as many of the check boxes as possible

g) Press Next

h) Set today's date by using the sliders

Task 5 deals with the use of different controls through the Discovery Controls application. It tests command buttons, links, check boxes and sliders. As mentioned in Chapter 3.8, some controls are designed with Microsoft Windows recommendations (Microsoft Corporation, 2011a) while others are designed as alternatives with different size, spacing, hit areas etc. The purpose with this task is to reveal how such controls should be designed, to be usable with gaze interaction. Subtask b) is created to test different command button sizes. The command button dwell time is set to 1 second, and to colour the button green, the participants will need to click it twice. The extra click is added to eliminate false positive errors (”lucky shots”), thus providing more reliable results. Figure 4.6 shows a participant clicking command buttons with varying sizes. This participant has managed to click the top three command buttons twice, and the fourth command button once.

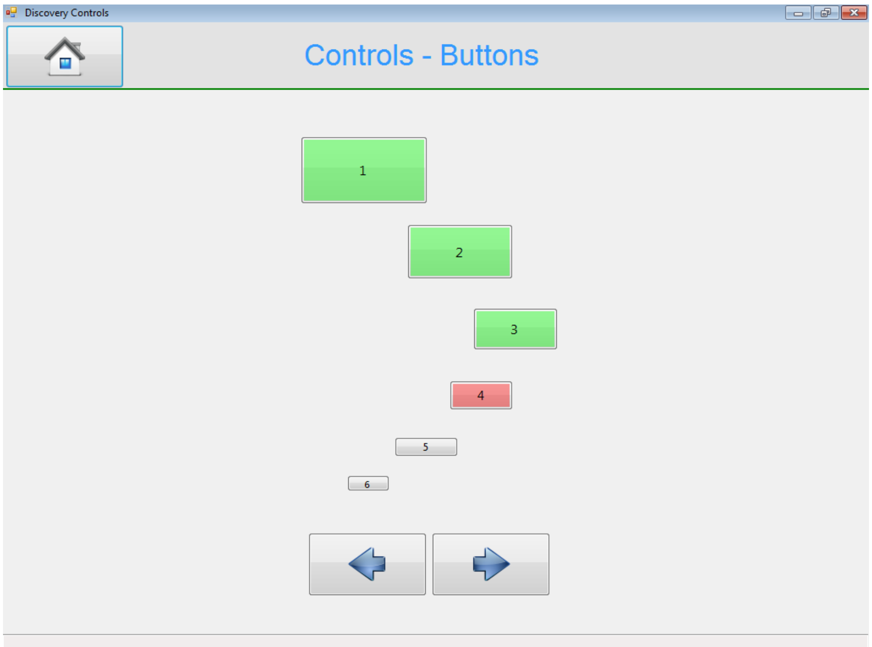


Figure 4.6: Participant solving Task 5 - Interaction with different controls.

In subtask d), the participant is presented with two rows of links with dwell time of 1 second. In addition to testing different font sizes, the task tests if tooltips are appropriate with gaze interaction. Consequently one row has tooltip feedback while the other just normal underlining. In subtask f), check box hit area and spacing are tested. The four columns each contain check boxes with different hit areas and spacing. By having four check boxes with the same properties, strokes of good luck are detected, and the possibility of false positive errors thus reduced. All check boxes have 500 ms dwell time. The final subtask (subtask h)), lets the participants try four sliders with different design properties. It tests the slider height, labelling and delay functionality (see Chapter 3.8 for more details). For the sliders with delay functionality, a 500 ms dwell time is used.

Five minutes was estimated for this task due to its many subtasks. In the subtasks containing comparisons, the participants were given some extra time to make up their mind of what they preferred.

The success criteria for these tasks are as follows:

- **Command Buttons** - Clicking a command button is registered as a success if the background colour of the button has turned green.
- **Links** - Clicking a link is registered as a success if its information window appears.
- **Check Boxes** - Managing to select check boxes in a column is registered as a success if two of the check boxes have been selected.
- **Sliders** - Using a slider is registered as a success if the participant is able to set it to a desired value.

4.4 Test Facility

The testing was conducted at the usability lab at NSEP (Norwegian EHR Research Center) in Trondheim. This usability lab was established in 2004 and has recognition both nationally and internationally (NSEP, 2012). It is located at the St. Olavs Hospital in Trondheim.

The lab is divided into several rooms, and the seating and environment may be arranged in multiple ways. The lab is well equipped for high quality recording of both participants and computer interfaces, and everything can be controlled from the observation room.

Our testing only required a small set up. We needed access to the observation room as well as a small testing environment in the usability lab. We had two tables put together with four chairs that was used during introduction and post-discussion, while the actual testing was conducted in front of the eye tracker on a separate table (see Figure 4.9). Two cameras were mounted in the ceiling to capture participants reactions and movements during the test. One camera recorded the participant and test leader from behind (*Camera 2* in Figure 4.9), while the other camera (*Camera 1* in Figure 4.9) was mounted in the ceiling above the participant, to be able to record the zoom gesture needed to solve Task 2 (see Chapter 4.3.2).

The test leader was present in the lab together with the participant, while the observer recorded the session from the observation room. The observation room has an 52 inches LCD monitor displaying the computer interface of the participant's computer, and the participant's gaze. This was achieved using LiveViewer from Tobii Studio (Tobii, 2012d) and extending the participant's computer desktop in Windows. Microphones in the ceiling and on the top of the participant's computer screen gave the observer the ability to hear what the participant and test leader said. See Figure 4.7 and 4.8 for illustrations.

The next chapter contains more information about the test equipment used during the tests.

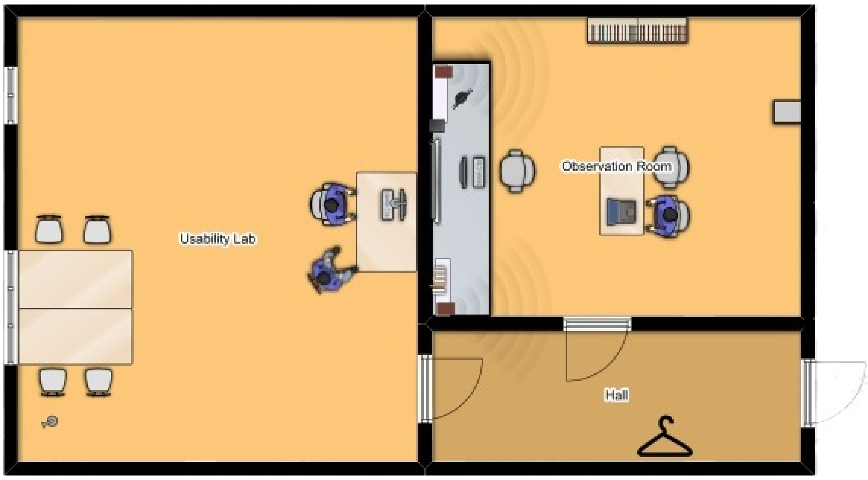


Figure 4.7: Overview of the test facility

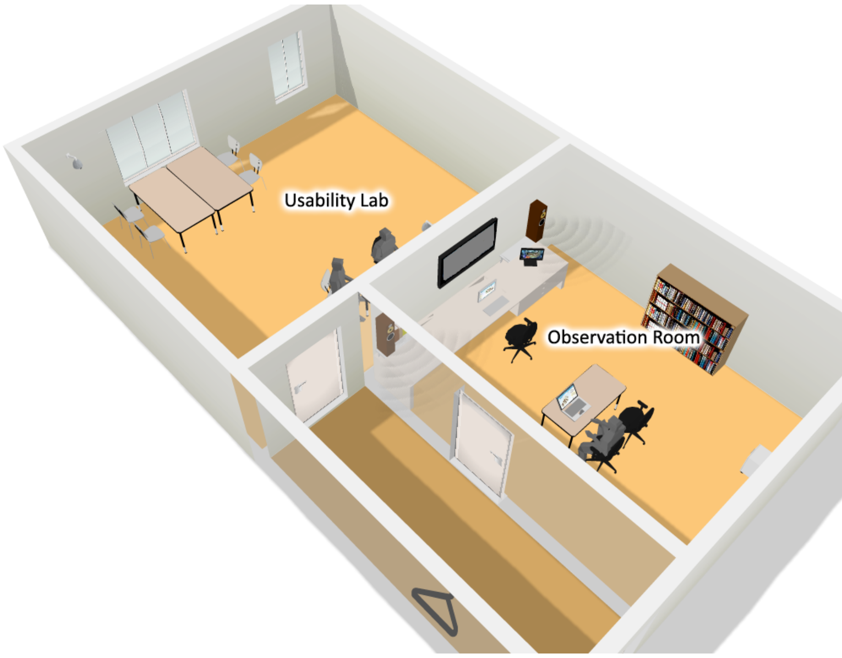


Figure 4.8: Overview of the test facility

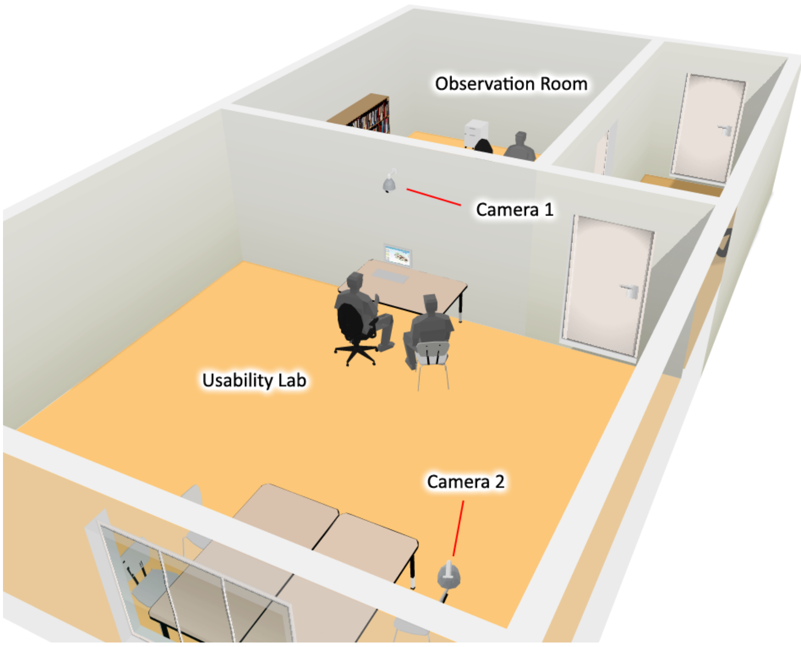


Figure 4.9: Illustration of participant and test leader in front of participant's computer with the eye tracker

4.5 Test Equipment

The following equipment was used in this study:

- A desktop computer running Microsoft Windows 7, with keyboard and mouse
- Tobii Studio software
- 23” monitor with screen resolution set to 1024 x 768 pixels
- Tobii X60 Eye Tracker
- Video cameras
- Microphones

In addition to this, there were equipment at the observation room such as speakers, recording devices, a large monitor etc. As this equipment does not affect the testing procedure from the viewpoint of the participant, the observation equipment is not further described.

The computer used during the testing was a well equipped desktop computer with a Intel Core i5 2.80 GHz CPU and 10 GB of RAM. A powerful graphic card was also in place, as well as sufficient hard disk place. Installed on the computer was Microsoft Windows 7 Professional 64-bit version together with a set of standard office applications. To record the user test sessions, Tobii Studio version 2.3.2.0 was used. By extending the computer desktop in Windows to a tv in the observation room, and by running Tobii Live Viewer on that screen, the observer could always pay attention to where the participant was looking and what he or she was doing, in addition to recording this screen from Tobii Studio. The screen recordings were used during the analysis. Figure 4.10 shows a screen shot of the Tobii Live Viewer during Task 3, with the participants face also being recorded for analyses purposes.

As shown in Figure 4.9 there were two cameras installed in the usability lab. *Camera 1* was mounted above the test participant to capture the head gestures needed to solve Task 2 while *Camera 2* was recording the test leader and the participants from behind. The video stream from these cameras were put together with a EureSys Picolo framegrabber, and ran through a Extron MVP 104GX video processor before it was sendt to the MultiCam Studio software. Through the microphones, audio was recorded as well. Thus we could both see test leader’s and user’s reactions and hear their comments and discussions during the test. Figure 4.11 shows the

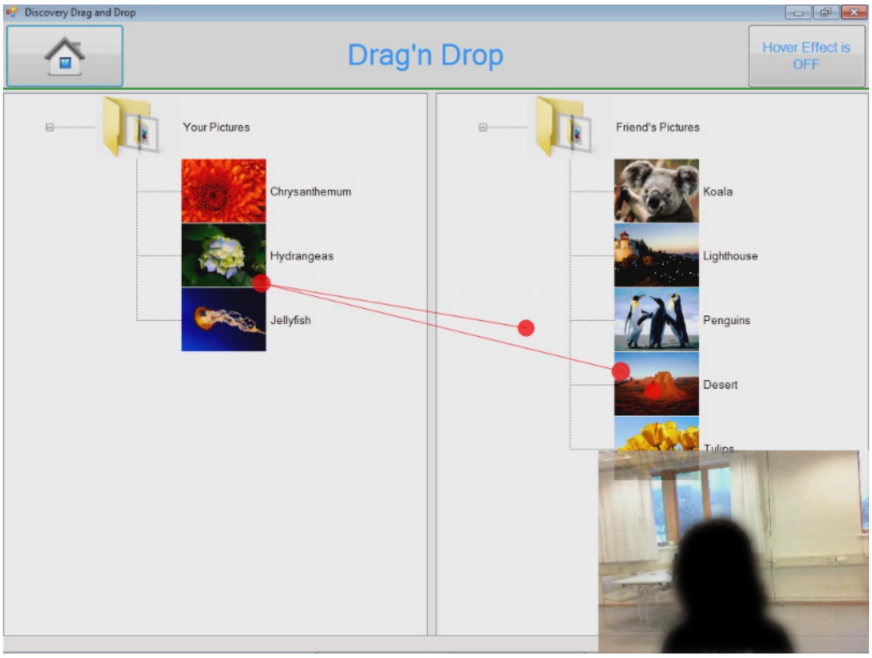


Figure 4.10: The captured video stream from Tobii Live Viewer showing the gaze of the participant in addition to a small video of the user in the lower right corner.

combined video stream for the two cameras. The video at the top is from Camera 1 while the video below is from Camera 2.

Technical specification for the eye tracker used, has been presented in the background chapter (Chapter 2.2.2). Users were seated in front of this eye tracker, and encouraged to place themselves comfortably and to adjust the chair so they centred the spatial cube of the eye tracker. This to optimize the eye tracking accuracy. An illustration of this spatial cube can be seen in Figure 4.12.

A Dell Ultrasharp U2311H monitor was used during the user testing. It was 23", had a widescreen 16:9 aspect ratio with a maximum resolution of 1920 x 1080 pixels (width x height). During early testing and development of the Discovery software we found the eye tracking inaccurate when running width this resolution and in 16:9 format. We therefore tested with different resolutions and formats and experienced that the best results were achieved using a standard resolution of 1024 x 768 format in 4:3 format. Online resources confirms that the most common screen resolution used in web surfing is 1024 x 768 (OneStat.com, 2012). Thus for the user testing, a 19 inches simulated screen with a resolution of 1024 x 768 was used. The DPI

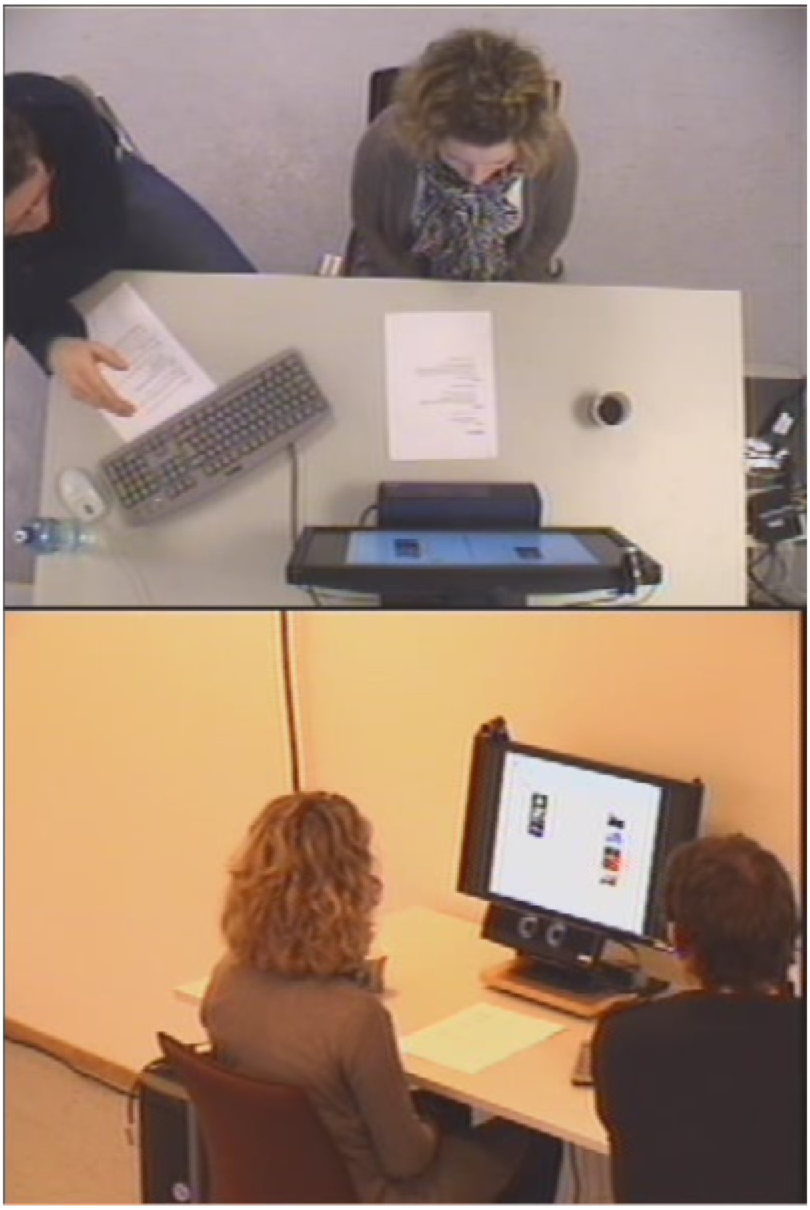


Figure 4.11: The captured video stream from Camera 1 (at the top) and Camera 2 (at the bottom).

(Dots Per Inch) for this resolution and screen was 67.

As full screen applications automatically become resized at different screen resolutions, one should note that software experience may vary when using different screen resolutions. When using eye tracking this becomes even more important, as the physical size of GUI components on the screen

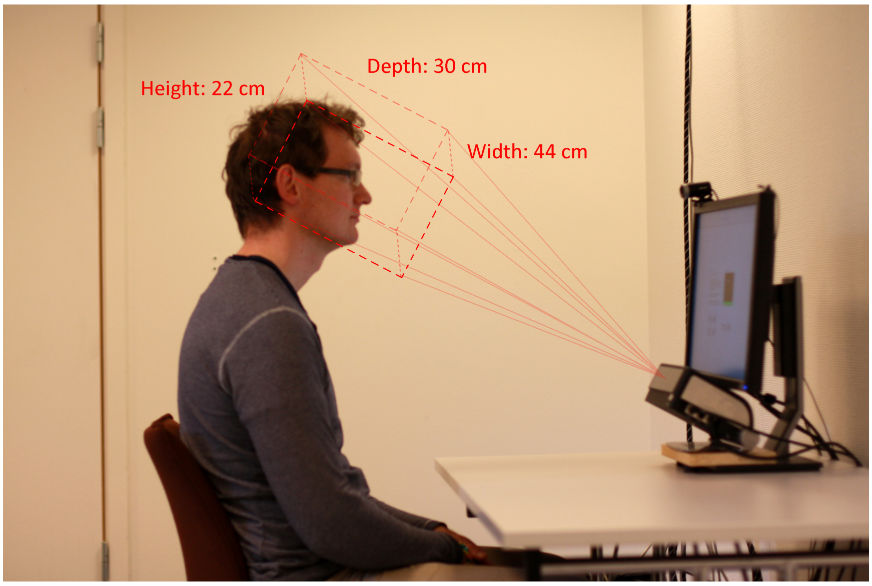


Figure 4.12: A user in front of the test set-up at the usability lab at NSEP in Trondheim. Eye tracking accuracy improves as the user's eyes are centred in the spatial cube.

changes in different screen resolutions. Setting a high screen resolution (for instance 1920 x 1080) makes GUI components small and harder to hit with gaze, using a smaller resolution (for instance 1024 x 768) makes them easier to hit. When running Discovery in 1024 x 768 resolution, and with a monitor DPI of 67, a main menu button of size 150 x 100 pixels becomes physically 57 x 38 mm. See Figure 4.13 for an illustration. With a screen resolution of 1280 x 960, the same button becomes 45 x 30 mm, and is thus reduced by over 20 %. This influences the user experience as it becomes harder to hit smaller buttons.

As it is standard practice to use pixels when talking about sizes in computer software, and as Microsoft Corporation (2011b) also gives guidelines in pixels, we will use pixels in our recommendations as well. However, one can calculate the physical sizes in mm by using Equation 4.1.

$$mm = 0.38 * px \tag{4.1}$$

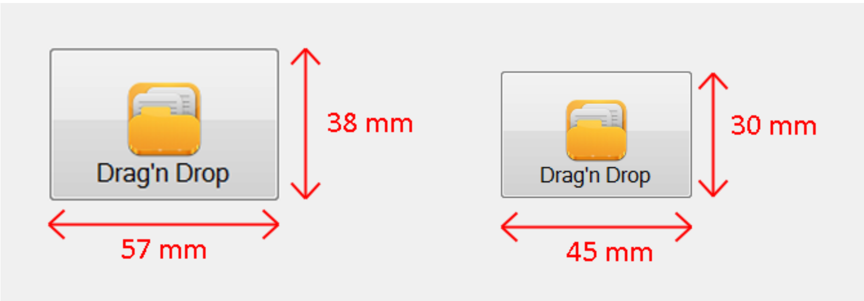


Figure 4.13: Physical size of a button in Discovery when viewed in 1024x768 (left) and 1280x960 (right). Both buttons have the same size in pixels: 150x100 px

4.6 Data Gathering Methods

The following data gathering methods were used in this study:

- Survey
- Questions
- Observations
- Screen, audio and video recordings

4.6.1 Survey

A two page survey was created in order to gather background information about each participant. The survey was short and specific and asked for personal information and the participants' past experience with internet and computers. The survey was customized in order to make them comparable with the national averages. The questions asked were short, understandable across different user groups, and without using a too technical vocabulary.

The survey is attached in Appendix D (originally in Norwegian, but translated into English).

4.6.2 Post-Task Semi-Structured Interview

The participants were asked a set of pre-defined questions after each user test task. The questions were tailored to the specific task and carefully prepared in advance. The test leader would often ask follow-up questions depending on the participant's answer. All interviews were recorded, while the observer wrote down relevant observations and conversations.

Each participant was in total asked 15 pre-defined questions. With fifteen participants this gave us 225 answers.

See Appendix C for the questions used in the post-task semi-structured interview.

4.6.3 Observations

Using an observation form, the observer took notes during the user test session. The observation form was created from scratch. It contained standard information such as date, time, user group and observer name. In addition it held pre-defined spaces for each of the user tasks and questions. For each task the observer took notes regarding the participant's reactions and comments, as well as the answers to the post-task semi-structured interview. The notes taken were mostly centred around the use of gaze interaction and not around Discovery's functionality.

See Appendix F for the observer form.

4.6.4 Screen, Audio and Video Recordings

Both screen, audio and video were recorded for each session. The screen recordings were done with Tobii Studio (Tobii, 2012d), which records the user's gaze in addition to a picture-in-picture of the participant. An example of this screen recording can be seen in Figure 4.10. This screen recording also captured audio.

Furthermore, video recordings from the other two cameras in the usability lab (see Chapter 4.4) were made. These recordings captured participant gestures and reactions during the test. Audio was enabled for these captures as well. See Figure 4.11

Both the Tobii Studio recordings and the camera recordings were useful during the analysis phase. If there was any doubt concerning the observation notes, one could examine either the screen recording or the video recording and figure out what had occurred. The screen recordings were also helpful to analyse specific events for each participant.

4.7 Test Measures

As seen in Chapter 2.4, user testing can be used to measure different metrics. With the definitions given in ISO 9241-11 (1998), this project aims to measure user effectiveness and satisfaction.

Effectiveness will be measured by task completion rate, and is something that the observer records during each user test session. Each user task has a success criteria that needs to be achieved before the task is marked as solved.

User satisfaction will be measured by qualitatively analyse post-task semi-structured interviews (explained in 4.6.2).

Due to time and scope limitations of this research, *efficiency* will not be measured. It would however have been interesting to measure a parameter like time to task completion in further research.

4.8 Test Procedure

All testing was scheduled to be done in one week. 5 days, 15 participants and approximately 45 minutes for each test session. The test team strived for consistency for all user tests, in order to make sure that all participants had a common ground when tested. This was done by using *test leader guide* (Appendix C). The test leader guide is a bullet list that the test leader follows throughout the user test session. Most important is the introduction to each participant, such that everyone are equally prepared before starting the test. The test team switched roles between tests, according to the test plan in Appendix B.

Below follows a description of a user test session with our set-up.

4.8.1 Welcome and Introduction

Before the user test, the test team prepared the test facility, equipment as well as the necessary documents. When the participant arrived the test leader and observer met the participant in the lab entrance and started of with a friendly chat about the weather, the location or similar. The participant was then briefly shown the observation room. The participant was, however, not given details about the observation room. The feeling of being observed might frighten the participant and may thus influence the results. The test leader and participant then entered the usability lab and sat down by the main table (see Figure 4.8), while the observer took place in the observation room (see Figure 4.9).

The test leader adjusted the conversation and vocabulary with respect to the test person. Consequently, the conversations differed between a teenager and a grown up, as well as between participants with little or much technology experience (Toftøy-Andersen and Wold, 2011).

The participant was offered coffee, cookies and snacks at the table while the test leader went through the introduction. Initially, the test leader informed the participant about the project and the upcoming test. How detailed information each participant was given depended on their past experience (if any) with user testing. Thereafter the test leader asked the person if he/she has been to a user test before. For more detailed information, see the Test Leader Guide attached in Appendix C. Then the participant was asked to read and fill out a standard informed consent form (see Appendix H), as well as a survey (see Appendix D).

While the participant filled out the informed consent form and the survey the test leader had the opportunity to look through the test leader guide to ensure that everything had been explained. When the participant was ready to start the test, the test leader and the participant moved over to the table with the eye tracker, while the observer started the recording.

4.8.2 Tasks and Questions

Before opening Discovery and starting the user tasks, the participant adjusted the chair in order to achieve an optimal calibration. The test leader then explained the calibration procedure and the limitations of head movements. The calibration was performed using Tobii Studio (Tobii, 2012d) with 5 calibration points. If the user's first calibration was inadequate, the calibration process was repeated. The test leader took note of how the participant was seated during the calibration. This in order to have a reference point later in the session. If the participant had a hard time interacting with Discovery, the test leader could ask the participant to adjust his or her seating towards the original position.

The test leader then explained the interaction possibilities in Discovery, before handing out the task sheet and started the test (see Chapter 4.3 for the tasks).

As this user test diverts somewhat from standard usability test, where the aim is to test usability, the test leader was allowed to explain, answer and be more active during the test. The test leader could therefore explain limitations and clarify tasks when needed. For instance in Task 3 (see Chapter 4.3.3) there was a need to explain the limitations in the application; the software did not allow for dropping pictures on white areas, or to pick up and drop pictures in the same tree etc.

For each task the participants were asked pre-defined questions (see Chapter 4.6), while the observer was taking notes. For more information about the observer role; see Chapter 4.6.

4.8.3 Wrap Up

When all tasks were completed and all questions has been asked and answered, it was time to wrap up the test through a post-discussion. The test leader and participants returned to the main table while the observers ended the recordings and joined them. The test leader started with an open question: *What do you think about this technology?*, and the participants answered thus setting the terms for the further conversation. A few follow-up questions were asked based upon what the participants answers. This post-discussion was not recorded nor taken into consideration as a result from the user test. It acted just as a friendly wrap up conversation. Finally, the test leader ended the session by awarding the participant with two cinema tickets.

4.9 From Test Results to Guidelines

As explained in Chapter 4.1, the goal is to suggest a set of design guidelines for gaze interaction based upon the user test results. To enhance these results, binomial proportion confidence intervals were applied and discussed. An explanation of the binomial proportion confidence interval has been prepared and can be found in Appendix A.

Both the result and discussion chapters are organized in connection to the corresponding user tasks (see explanation in Chapter 5 and 6). This in accordance to the ISO/IEC 25062 (2006) recommendations.

The suggested guidelines originate from one clear, or several strong user task result(s). Each result was both analysed by itself and in comparison to others to discover if it provided an adequate basis for a guideline (see the discussion chapter, Chapter 6, for more details). It is however important to note that our proposals are just a first step towards complete design guidelines for gaze interaction. Extensive testing with several methods and a larger sample sizes is required before reaching final guidelines.

The following chapter holds the test results, which act as a foundation for our recommendations.

Chapter 5

Results

This chapter lists the results from the user test. It is divided into subchapters, each containing the results from one of the tasks described in Chapter 4.3. In addition to 14,8 hours of recorded sound and video material, 38 pages of written observer notes were taken during the user tests. The results listed are a summery of this material, and will be reflected upon and discussed further in Chapter 6. Please note that all participant quotes have been translated from Norwegian to English

5.1 Task 1 - Playing a Video Game

As described in Chapter 4.3.1, the purpose of this task was to get the participants accustomed to control the computer using gaze interaction. Even though no specific results were expected from this task, a couple of observations are worth mentioning.

Firstly, a process took place among the participants. They tried to *gaze* in different ways to best control the game. For some it was natural to move their head in the direction that they were gazing. As a consequence of this, the eye tracker did not fully register the participants' eye movements, making the UFO difficult to control. By analysing the video material, one can see several participants moving their heads a lot in the start of the game, and then gradually starting to understand that it is enough to just use their gaze. One can also see that most participants tilts their head forward a little and squint their eyes in the start of the task. Some returned to a more natural position after a couple of minutes, but others remained in this concentrated position for the remaining tasks. The test leader encouraged the mentioned participants to relax and find a comfortable position to improve the eye tracking accuracy.

Secondly, by analysing the video material and participants' comments, we discovered that the majority of them found it enjoyable to interact using their gaze. It was registered that the game appealed especially to the youths. However, two users commented that it felt a bit unusual and unnatural to actually control something with the eyes.

All fifteen participants completed this task. Below follows a set of participant quotes:

- ”It was a bit weird, but fun”
- ”Wow, how do you control the UFO speed?”
- ”Very amusing!”

5.2 Task 2 - Exploring a Picture Gallery

All fifteen participants completed this task (Figure 5.1), both navigating between pictures and using head gestures for zooming. By using Formula A.5 with a sample size of 15 persons, we know with 95 % certainty that above 94,96 % of the population will manage to solve such tasks. No problems were encountered in the navigation part of the task. All participants navigated swift and easily through the pictures using the 90 x 280 pixels (width x height) command buttons. Everyone also managed to zoom in on pictures using the *lean* gesture. The majority of users found the gesture natural to do when wanting to zoom a picture of interest.

Among the positive statements the participants found this functionality natural and intuitive, quoting;

- ”Actually it was pretty intuitive, because one usually leans forward to take a closer look. It was very intuitive. This is what I would do if it was a sheet of paper.”
- ”I wish I had this at home”
- ”Cool, seems natural”
- ”Okay, do I zoom like this?” [the user moves his head forward and backward] ”That was clever”
- ”Fantastic”

Fourteen out of fifteen participants gave such positive statements, the remaining person had a more negative experience with the use of head gestures for zooming. The person felt it unnatural to use this type of head gesture and said among others: *”It was a bit weird in my opinion. Usually when looking at pictures it is not this...* [user thinks] *...It is not this motion one is used to use. It seems a bit unnatural.”* And the person continues: *”This would not be the first I would think about, that one should move the head.”*

Adding up the positive and negative statements we get the following results for using head gestures for zooming pictures:

- Positive Statements: **93,3 %**
- Negative Statements: **6,7 %**

In this task we also observed a learning curve for the participants. With the video material collected from the camera placed directly above the participant (Camera 1 in Figure 4.9), it was easy to see that participants’ head gesture movements varied during the task. Most started with a lot of movement and then gradually adapted their head movements to fit each of the picture gallery's zoom steps (zoom steps described in Chapter 3.5).

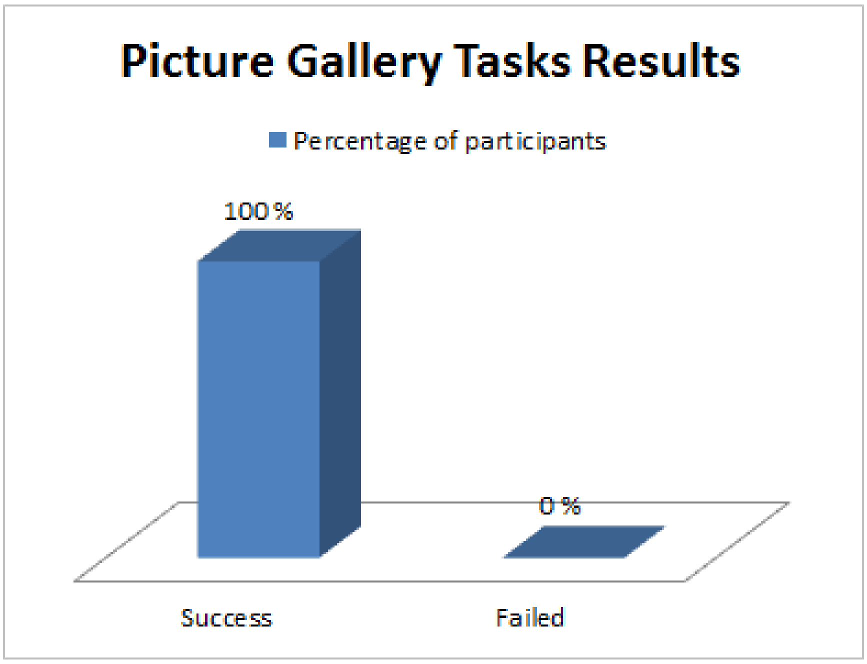


Figure 5.1: Graph showing the completion rate of participants for the Picture Gallery tasks

5.3 Task 3 - Doing Drag and Drop Operations

From the Drag and Drop task described in Chapter 4.3.3, we experienced that most participants easily picked up and dragged the pictures, but had problems dropping them. When asked about the overall experience of the task, ten out of fifteen participants (67 %), commented that it was difficult to release a picked-up picture. This without being asked specificity about the pick-up or release functionality. Some said they did not understand why releasing was so hard, where others commented that the release time was too slow. From an observer's perspective, one saw a tendency of user impatience. When releasing the picture they stared correctly at a valid destination for approximately 1,5 seconds, then moved their gaze a little to see if it would help. After their gaze settled at the new location, they waited another 1,5 seconds before they moved their gaze again (and so it continued). The release time was set to 2,5 seconds. The times they moved their gaze, the picture often went outside the hit area causing the release time to reset. This was also the case for participants with poor calibration. When gazing at a valid release destination the gaze briefly jumped outside the hit area and the release time was reset.

Using Formula A.5 (from Appendix A) we find the 95 % confidence interval for the ones who commented that it was difficult to release a picked-up picture to be: 50,8 % < p < 82,53 %.

The Drag and Drop task allowed participants to do exactly the same operations first with a gaze marker and then with a hover effect. After having tried both, they were asked which one they preferred. Figure 5.2 shows the distribution of the preferred interaction feedback. Again using Formula A.5 (explained in Appendix A) we find the 95 % confidence intervals for the population proportions displayed in Table 5.1. The participants who

Interaction Feedback	**Sample Proportion**	**Confidence Interval** $(\alpha = 0,05)$
Hover Effect	60 %	45,13 % < p < 74,87 %
Gaze Marker	33 %	25,40 % < p < 41,27 %
Undecided	7 %	5,81 % < p < 7,53 %

Table 5.1: Confidence intervals for the population proportion (p) for preferred gaze interaction feedback in a Drag and Drop operation

preferred the gaze marker feedback, described the hover effect as jagged and that it stole the attention away from the picture. They stated:

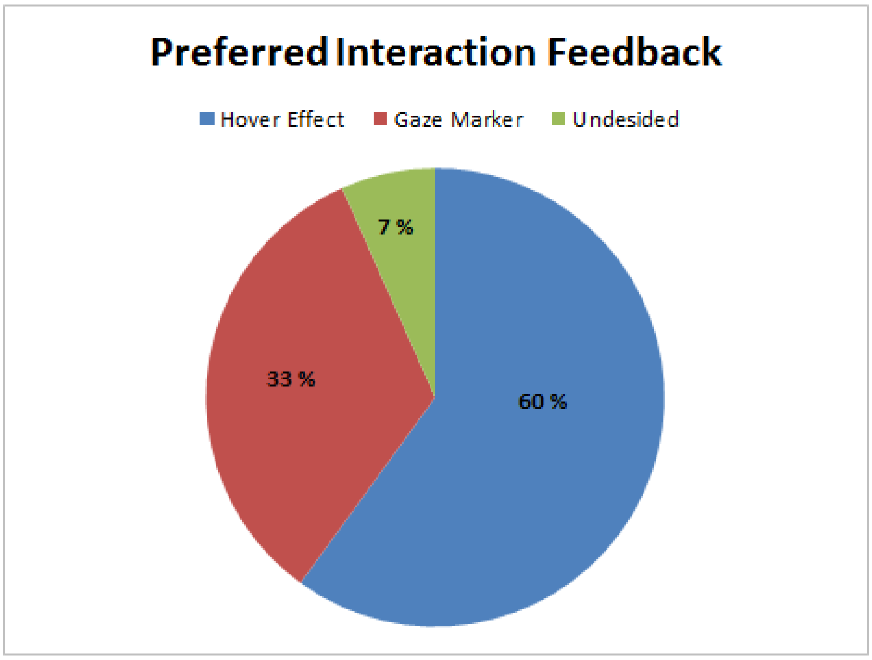


Figure 5.2: Graph showing the distribution of which interaction feedback the user test participants preferred.

- ”I like the gaze marker the most because then I always know where I am”
- ”I would like the gaze marker to be a bit larger, but I still prefer it over the hover effect”
- ”I lost focus on the picture when hover effect was enabled”

On the other hand, the participants who preferred the hover effect found it more natural and easier to use compared to the gaze marker. They described the gaze marker as too small, annoying and distracting. Some quotes are:

- ”This was much easier with the hover effect”
- ”I prefer the hover effect, partly because of the gaze marker shaking”
- ”I am used to the hover effect highlighting from using the mouse”

In the end of the task, when asked if they would prefer using gaze interaction for organizing their pictures at home, most would not (as shown in Figure 5.3). They found gaze interaction unnatural and slow, and said among other things:

- ”I would rather use a mouse, because it is much faster”
- ”Even if available, I would not have used this at home”
- ”I think this would be much easier with a mouse, at least if many pictures are to be moved”

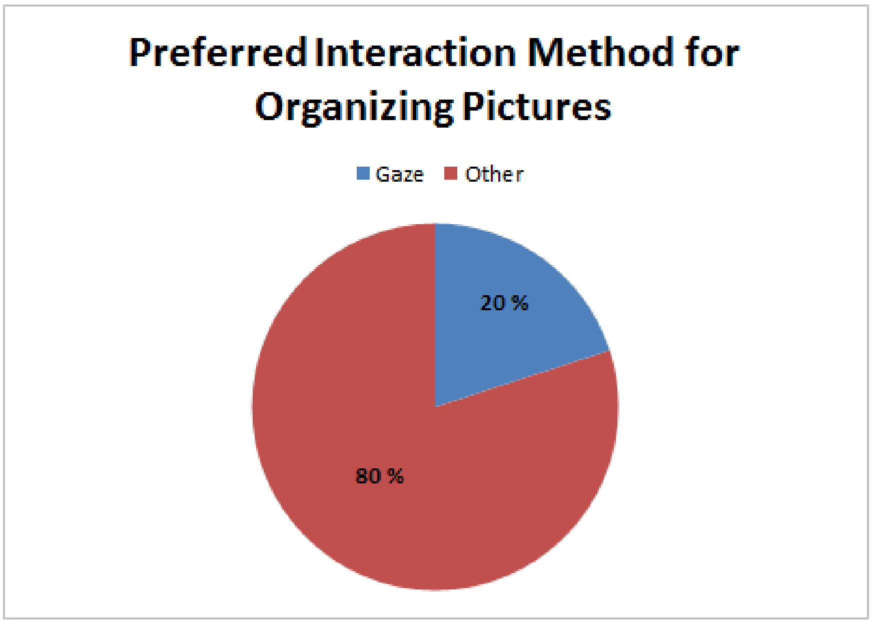


Figure 5.3: Showing the number of participants who would prefer using gaze interaction in preference to other methods when organizing pictures.

5.4 Task 4 - Browsing a Web Page

As described in Chapter 4.3.4, the two main purposes of Task 4 was to test active as well as passive scrolling. In both cases a change in the participants opinions throughout the task was observed. Some first impressions were characterized by a feeling of little control, and several participants described the experience as confusing, unnatural and disturbing. However, after about a minute letting them get accustomed to the functionality, most of these participants changed their mind. Below some of one participant's statements throughout the task are listed. Statements after reading the first paragraph (subtask *b)*):

- ”It would be more natural if the text did not move while I was reading”
- ”The text moving is a disturbing element for me”

after reading the second paragraph (subtask *c)*):

- ”This became much better during this segment”
- ”I think it have much do to with habituation”
- ”I was no longer disturbed by the moving”

Statements after reading the third paragraph (subtask *d)*):

- ”Now it is actually comfortable [...] I can relax while I am using it”

5.4.1 Reading

Disregarding the habituation process mentioned above, all fifteen participants were positive towards the automatic scrolling functionality. They felt comfortable using it, and all stated they would definitely use this functionality if it was available on their personal computer (see Figure 5.4). Some participants were clearly aware of the functionality, excitingly testing its limits and features. Others enjoyed the experience by calmly reading as in a book or newspaper. Below follows some participant quotes:

- ”I really like that it slows down when I am reading difficult words”
- ”Very comfortable in contrast to using the mouse”
- ”Fantastic! Dice Throw 6”
- ”Amazingly comfortable! The screen moves exactly in line with my eyes”

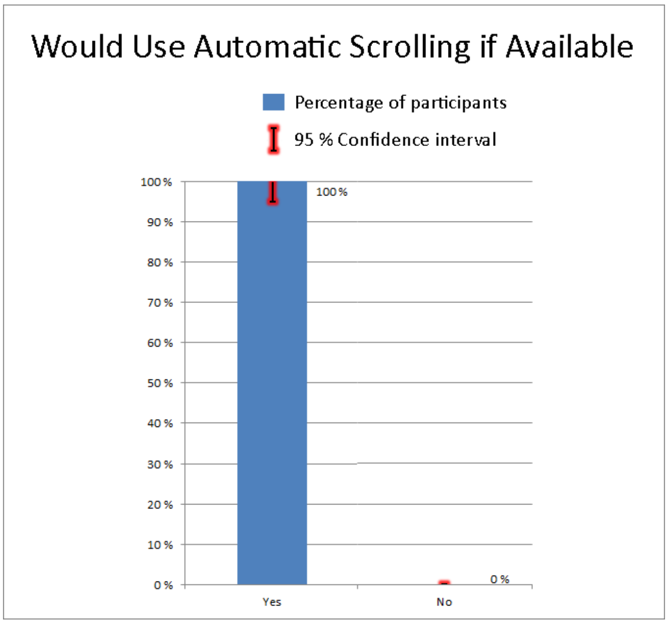


Figure 5.4: Graph showing how many participants would use automatic scrolling if available on their personal computer

Two of the participants found it somewhat stressful to read with automatic scrolling. When asked if the reading experience was either stressful or relaxing one participant answered:

> *I would say stressful, but [...] I noticed that I got used to it only in the two, three, four minutes we were doing this. It felt a bit like... Like what they have on the television, a Teleprompter. It felt like that. But, as I say, one gets used to it, and then it works better.*

In spite of its ambiguity, this answer was characterized as ”stressful”. When asked to rate this experience as natural or unnatural the participant said:

> *Hmmm... I would say it is unnatural. But it is a bit like... The first time one got a touch telephone, it was also a bit like: ”Wow!” And then it took 10 minutes and it was all fine. So I think one gets really fast used to it. It was actually really enjoyable to not have to press anything.*

In spite of its ambiguity, this answer was characterized as ”unnatural”.

The participants were also asked to comment on the scrolling speed. 87 % of the participants found the automatic scrolling speed appropriate.

When asked if they felt they were forced to read faster than usual because of the automatic scrolling, the majority of the participants responded no. They said they felt the scolling speed adjusted itself according to their reading speed and that it was comfortable. The remaining 13 % found the scrolling speed too fast and somewhat stressful to keep up with. No one felt the scrolling speed was too slow while reading. Table 5.2 summarizes the above findings.

Reading Experience	**Sample Proportion**	**Confidence Interval** ($\alpha = 0,05$)
Would use automatic scrolling at home	100 %	94,96 % < p < 100 %
Scrolling speed appropriate	87 %	71,92 % < p < 100 %
Scrolling speed too fast	13 %	11,06 % < p < 15,60 %

Table 5.2: Confidence intervals for the population proportion (p) for different opinions about the reading experience with automatic scrolling

5.4.2 Active Scrolling

87 % of the participant felt it was easy to scroll actively. The remaining 13 % found it uneven and difficult to control. From an observers viewpoint, it was evident that when scrolling actively, the participants unintentionally looked at and were following pictures, words or headlines out of the scrolling area. This caused the scrolling function to stop and reset the scrolling speed. This is reflected in the participant statements. Most of the participants got accustomed to the scrolling functionality even though it was difficult to control. The few who managed to scroll without interruptions said they defocused their gaze to not be affected by the rolling text. Some participant quotes follows below:

- ”The text distract me, so I just defocus my eyes”
- ”It is a bit weird to not look at the text while scrolling”
- ”I don‘t like it, it is continuously interrupted”
- ”It is easier to scroll down than up, but both are too slow”

Characteristics

After completing this task the participants were asked to characterize the experience as either:

- Stressful or relaxing
- Easy or difficult
- Natural or unnatural

Their answers varied depending on the different scrolling functions. For instance, some found it relaxing to read with automatic scrolling, but stressful to scroll actively. For each characteristic, the results are therefore split into separate charts showing the results for both automatic scrolling and active scrolling respectively (Figure 5.5).

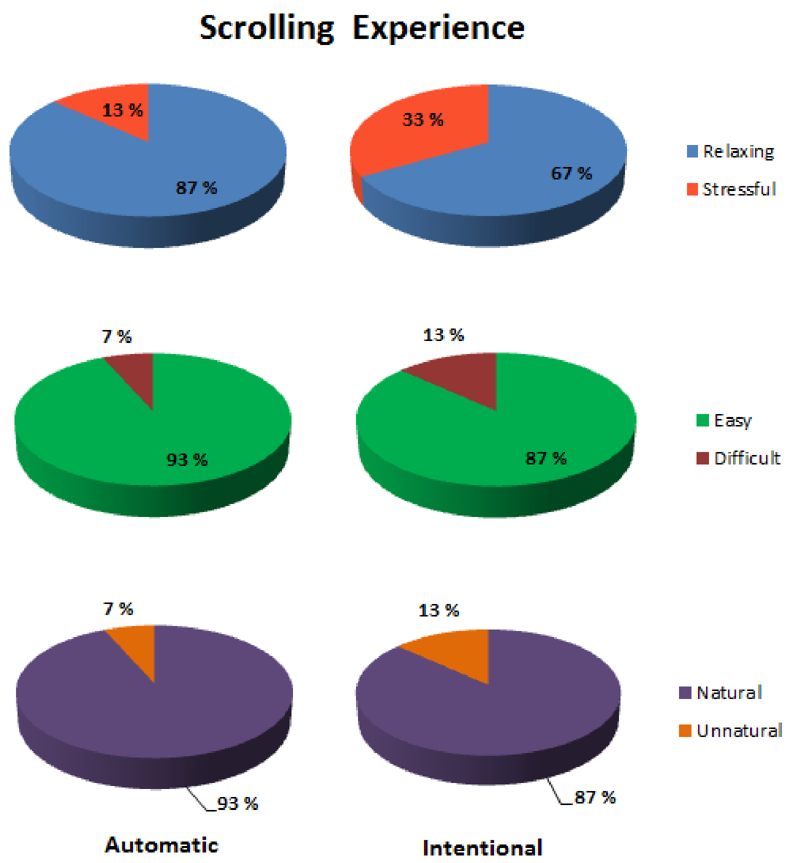


Figure 5.5: Graph showing how the participants characterized the scrolling experience in Task 4.

5.5 Task 5 - Interaction With Different Controls

This last task, as described in Chapter 4.3.5, focused on Windows Controls. It was divided into subtasks, each utilizing a different part of the Discovery's Controls application for testing a specific control. The results are represented in figures 5.6 to 5.11.

Command Buttons

Figure 5.6 shows the percentage of participants that managed to click the different buttons in subtask b). Each button had to be clicked twice to be registered. Button sizes are displayed below the button names, and the Windows logo marks the pillar representing the Microsoft Windows size recommendations. When participants did not manage to hit a button, it was mainly due to poor calibration or gaze jitter. From the video material it was observed that these sources of error were mainly vertical, thus only affected by the height of the command buttons. The width however, was never a problem.

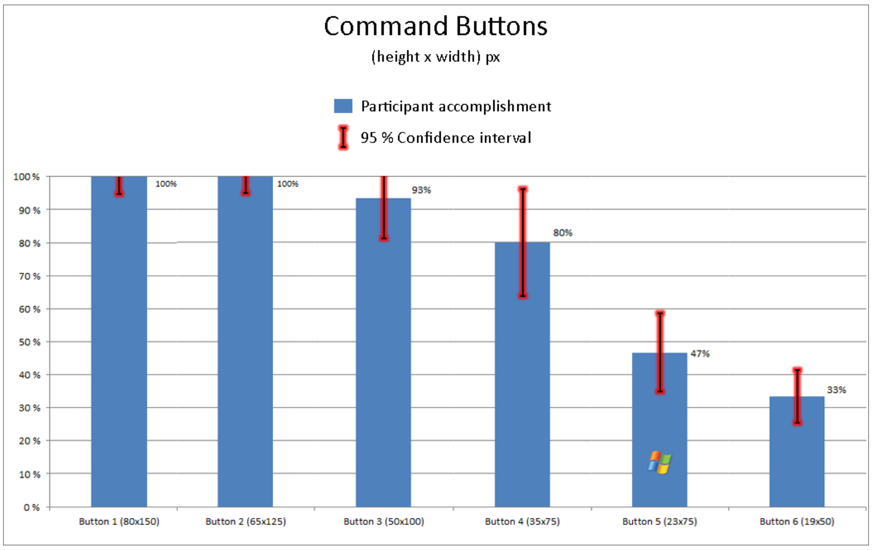


Figure 5.6: Results from subtask 5b), showing which buttons the participants were able to hit

Table 5.3 shows the computed confidence intervals for the command button results.

Button	**Sample Proportion**	**Confidence Interval ($\alpha = 0,05$)**
Button 1 (80 x 150 pixels)	100 %	94,96 % < p < 100 %
Button 2 (65 x 125 pixels)	100 %	94,96 % < p < 100 %
Button 3 (50 x 100 pixels)	93 %	81,28 % < p < 100 %
Button 4 (35 x 75 pixels)	80 %	63,81 % < p < 96,19 %
Button 5 (23 x 75 pixels)	47 %	34,88 % < p < 58,45 %
Button 6 (19 x 50 pixels)	33 %	25,40 % < p < 41,27 %

Table 5.3: Confidence intervals for the population proportion (p) for accomplishing pressing buttons with varying sizes. Button 5 is the Windows recommended button size.

Links

Figure 5.7 shows some of the results from subtask d), clicking links. Each pillar represent how many participants that managed to click a specific link. Below the pillars are the link number and font size displayed. The pillar with the Microsoft Windows logo represent the link with the Microsoft Windows font size recommendation. Figure 5.8 shows how many participants preferred link tooltips. A confidence interval has been added to the figure from 94,96 % to 100 %. As seen from the figure, all participants disliked the tooltips. Their arguments were mainly that the tooltips were distracting, annoying or unnecessary. Some arguments are quoted below:

- ”I don‘t like the ones with bubbles, I unwillingly move my eyes to read them”
- ”I prefer the ones on the lower line, the bubbles are annoying”
- ”The box appearing is very annoying”

Table 5.4 shows the computed confidence intervals for the link results. The pixel and millimetre calculations are based on a 19 inches screen running 1024x768 resolution, which gives a DPI of 67 (see Chapter 4.5 for more information regarding screen resolution and sizes).

Check Boxes

Results from subtask 5f) are presented in Figure 5.9. Each pillar shows how many percent of the participants who managed to select the check boxes from a specific column. Under each pillar are the column numbers and the check box design properties (width x height x spacing). Width and

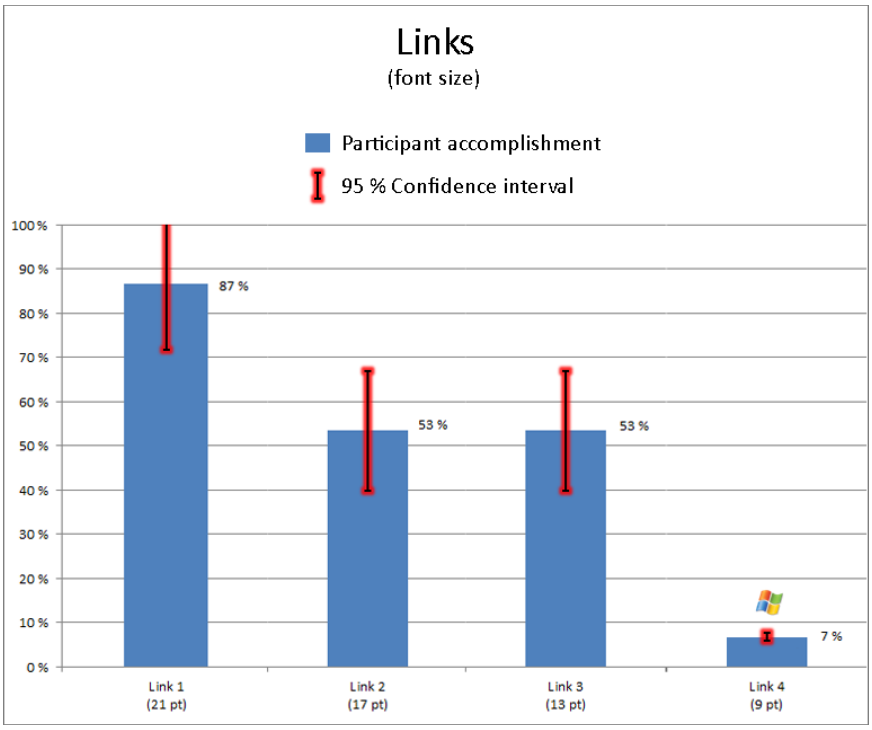


Figure 5.7: Results from subtask 5d), showing which links the participants were able to hit with gaze. 1 pt = 0.3527 mm. (see Table 3.2 in Chapter 3.8.2 regarding font sizes)

Link	**Size (height)**	**Sample Proportion**	**Confidence Interval ($\alpha = 0,05$)**
Link 1	21 pt, 20 px, 7 mm	87 %	71,92 % < p < 100 %
Link 2	17 pt, 16 px, 6 mm	53 %	39,86 % < p < 66,81 %
Link 3	13 pt, 12 px, 5 mm	53 %	39,86 % < p < 66,81 %
Link 4	9 pt, 8 px, 3 mm	7 %	5,81 % < p < 7,53 %

Table 5.4: Confidence intervals for the population proportion (p) for accomplishing clicking links with varying sizes. Link 4 is the Windows recommended link size.

height represents the size of the check box hit area, and spacing represents the space between the check boxes hit areas. The participants had to select (check) two or more check boxes in a column for the subtask to be registered as accomplished. In this figure, the Windows logo marks the pillar representing the check boxes designed according to Microsoft Windows recommendations for sizing and spacing. Table 5.5 shows the computed

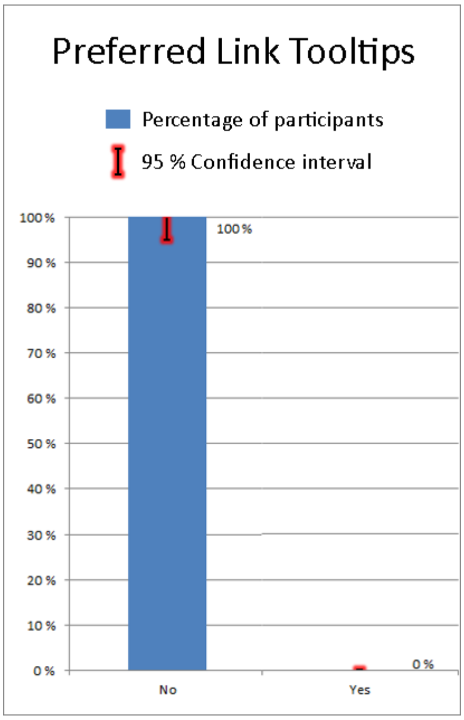


Figure 5.8: Results from subtask 5d), showing how many participants preferred link tooltips

confidence intervals for the check box results.

Check box column	**Sample Proportion**	**Confidence Interval** ($\alpha = 0,05$)
Check box column 1 (83x40x4)	100 %	94,96 % < p < 100 %
Check box column 2 (83x30x4)	80 %	63,81 % < p < 96,19 %
Check box column 3 (83x17x17)	27 %	20,93 % < p < 33,07 %
Check box column 4 (83x17x3)	33 %	25,15 % < p < 40,85 %

Table 5.5: Confidence intervals for the population proportion (p) for accomplishing selecting check boxes with varying sizes and spacing (width x height x spacing in pixels). Check box column 4 represents the Windows recommended check box size and spacing

As mentioned in Chapter 3.8.3, it would be interesting to compare the results from different columns to measure of the impact of sizing and spacing variations. In Figure 5.10 the results from column 1 is compared to those of column 2, and similarly column 3 to column 4. They illustrates how

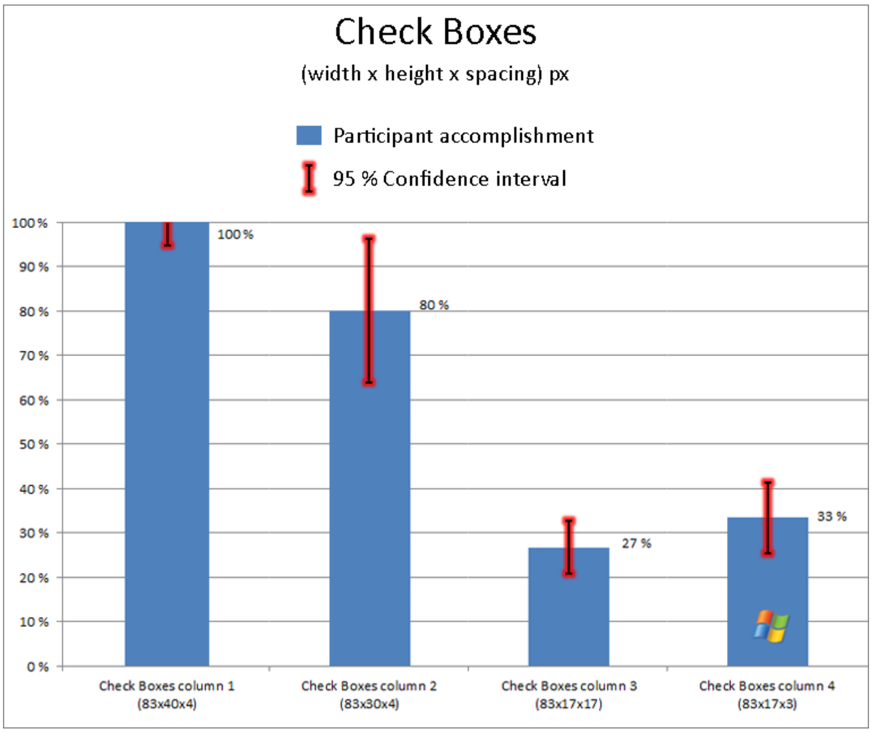


Figure 5.9: Results from subtask 5f), showing which columns of check boxes the participants were able to hit

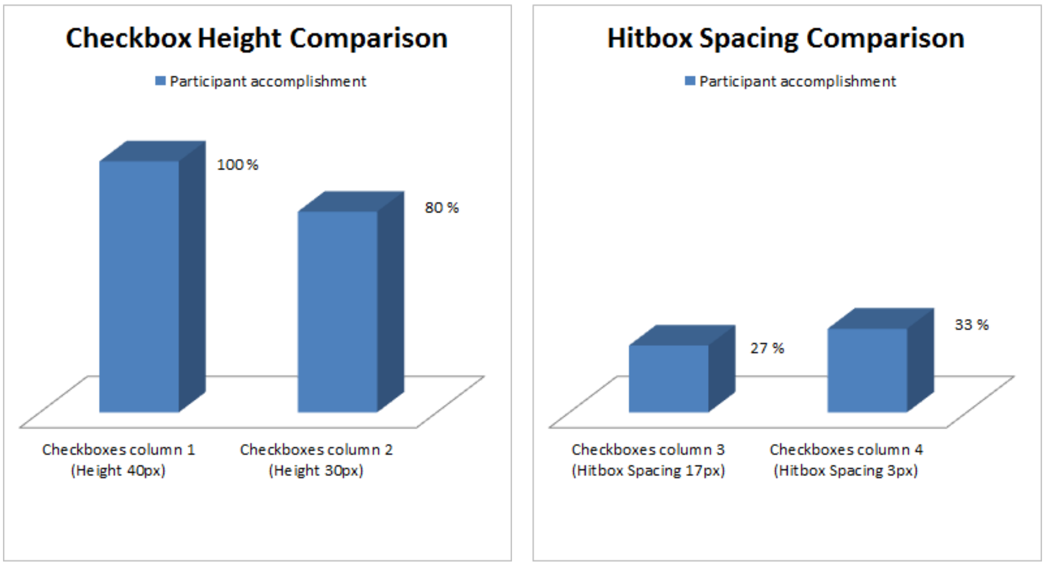
changing the height and spacing of the check boxes‘ hit areas affects the rate of completion.

Sliders

The last part of Task 5 tested slider design. The participants were asked to set today's date using four sliders with different design. All participants accomplished the subtask (see Figure 5.11), but had different opinions about which one they preferred.

Only one out of fifteen participants preferred Slider 1. It was chosen for it responsiveness, something that was commented as both positive and negative by the participants. A negative experience registered was that it was difficult to control. Its single value label led to unintentional movement of the slider. The slider naturally jumped towards the label every time the participant checked the current value (gazed at the label). Below follows some participant quotes from using Slider 1:

- "I feel I can not check whether it is set correctly or not, because then



(a) Height Comparison (b) Spacing Comparison

Figure 5.10: Comparison of the height and spacing impact on task completion for check boxes.

I may change it again"

- "Wow, this one moves as soon as you glimpse at it"
- "This was sensitive and good"

The multi labelled Slider 2 was the most popular with 8 of 15 votes. The recorded positive statements were mainly about its responsiveness and precision. They felt it was easy to control and release at the desired values. The participants that did not like Slider number 2 found it to sensitive, similar to Slider 1, and wanted a way to lock its value after it was set. Some participant quotes using Slider 2 were:

- "This one was so much better than the first one"
- "I like this one the best even though it leads to more text"
- "I wish it was a way to make it non-editable when I don‘t want to change it"

Slider 3 and 4 received respectively 2 and 4 of the 15 votes. They were both complemented for their delay functionality and criticized for being too slow or unresponsive. The latter was especially related to Slider 3. From an observers viewpoint one could see that some participants' gaze sometimes

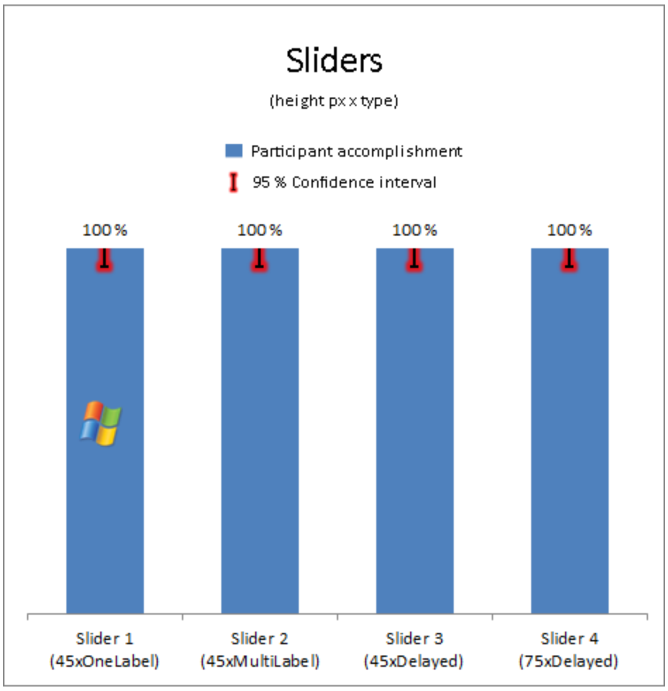


Figure 5.11: Results from subtask 5f), showing which columns of check boxes the participants were able to hit. The confidence intervals with $\alpha = 0,05$ are 94,96 % <p< 100 %.

fell outside the hit area of Slider 1, 2 and 3, but not with Slider 4. This led to a positive attitude towards Slider 4 for the affected participants. The delay functionality in Slider 3 and 4 provided for some participants the missing locking functionality. They felt in control and that is was comfortable to use and release. The delay functionality resulted in a somewhat slower speed, which several participants found too slow and annoying. Below are some participant quotes from using Slider 3 and 4:

- Regarding Slider 3: *"I am looking at the numbers, but nothing is happening!"*
- Regarding Slider 3: *"This one is too slow, makes me want to drag it with my hand instead"*
- Regarding Slider 4: *"This one is very comfortable, I don't know why"*
- Regarding Slider 4: *"There is not much difference between these last twos"*

The distribution of which slider the participants preferred are shown in Figure 5.12. After each slider number are the associated height and type displayed. Height meaning the slider's hit area hight in pixels, and type being one of the types specified in Chapter 3.8. The Windows logo marks the slice representing the slider designed according to the Microsoft Windows guidelines ((Microsoft Corporation, 2011b)).

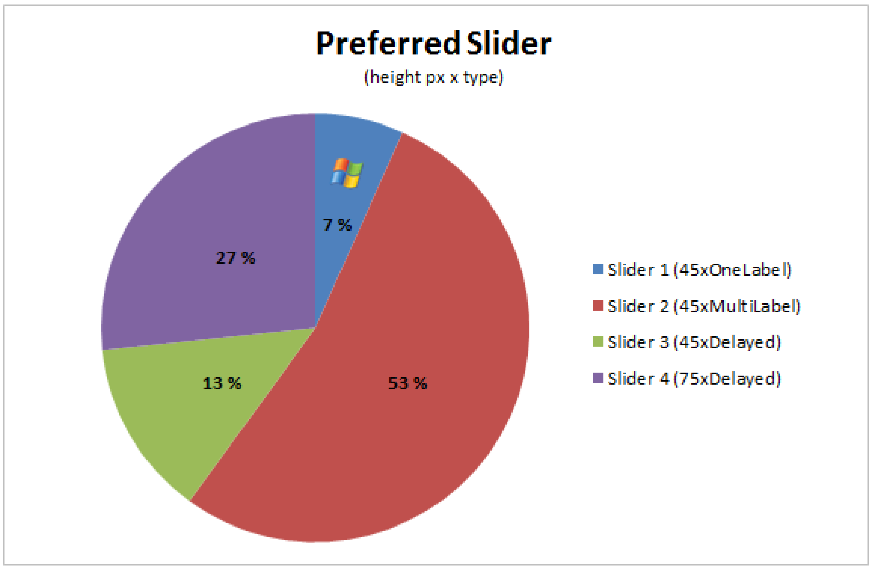


Figure 5.12: Results from subtask 5h), showing which slider the participants preferred

These results and associated confidence intervals are presented in Table 5.6.

Preferred Slider	**Sample Proportion**	**Confidence Interval** ($\alpha = 0,05$)
Slider 1 (45xOneLabel)	7 %	6,10 % < p < 7,90 %
Slider 2 (45xMultiLabel)	53 %	39,61 % < p < 66,39 %
Slider 3 (45xDelayed)	13 %	10,79 % < p < 15,21 %
Slider 4 (75xDelayed)	27 %	20,93 % < p < 33,07 %

Table 5.6: Confidence intervals for the population proportion (p) for the preferred slider.

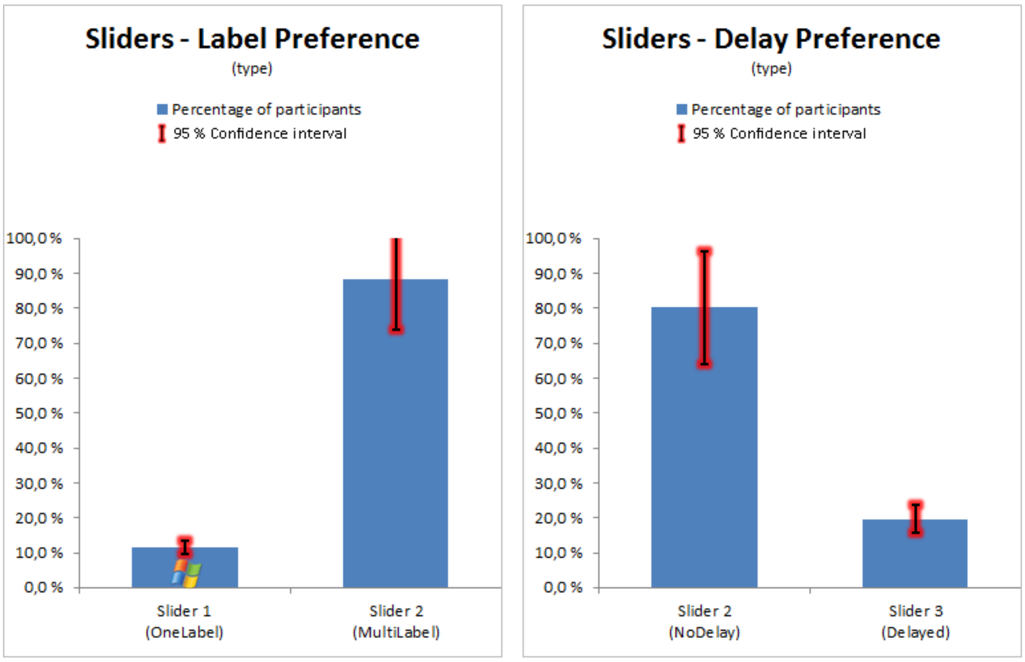
However, to be able to measure how labelling, delay and the slider hit area hight affects the task completion, some recalculation is needed. This is done for three different cases, which are:

- **Label Preference**
  Sliders 1 and 2 are identically designed, with the exception of labelling

type. Therefore, to see the influence of the different types of labelling, the results from these sliders were recalculated and compared. Isolating Slider 1 and 2 and adding up their results, one get 60 %. Then we calculate the shares of 7 and 53 respectively in relation to 60. The new results of 11,7 % and 88,3 % with associated confidence intervals are given in Table 5.7, and illustrated in Figure 5.13a.

- **Delay Preference**
  To see the delay preference the results from the sliders with and without delay separately are added up. For Slider 1 and 2 we get 60 % and for 3 and 4 we get 40 %. The results with associated confidence intervals are given in Table 5.7 and illustrated in Figure 5.13b.

- **Height Preference**
  The hit area height is the only design difference between Slider 3 and 4. By recalculating as described under *Label Preference* we get new results of 32.5 % and 67.5 % respectively. These results with associated confidence intervals are presented in Table 5.7 and illustrated in Figure 5.14



(a) Label Preference (b) Delay Preference

Figure 5.13: Label and delay comparison of sliders.

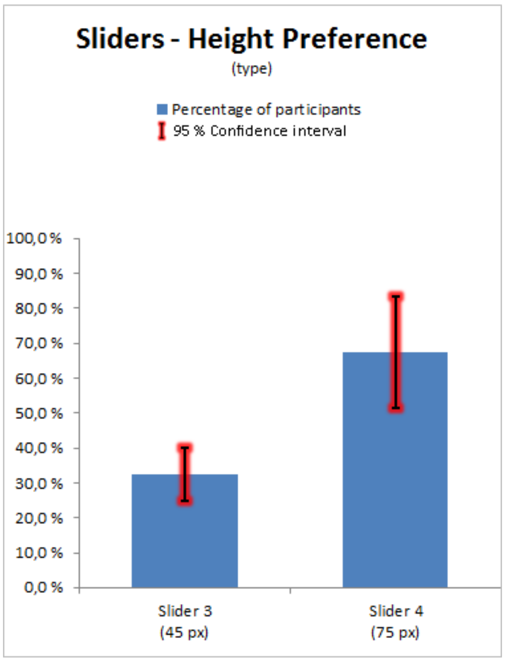


Figure 5.14: Height comparison of sliders.

Sliders (type)	**Sample Proportion**	**Confidence Interval ($\alpha = 0,05$)**
Slider 1 (OneLabel)	11,7 %	9,79 % < p < 13,61 %
Slider 2 (MultiLabel)	88,3 %	73,91 % < p < 100 %
Slider 2 (NoDelay)	80,3 %	64,14 % < p < 96,46 %
Slider 3 (Delayed)	19,7 %	15,73 % < p < 23,67 %
Slider 3 (45 px)	32,5 %	24,80 % < p < 40,20 %
Slider 4 (75 px)	67,5 %	51,50 % < p < 83,50 %

Table 5.7: Confidence intervals for the population proportion (p) for the preferred slider, recalculated for comparison.

Age and Sex Influence

Table 5.8 shows how the results from Task 5 were distributed across different user groups and sexes. This opens for a study of what influence sex and age have on such results. As seen from the table, all numbers are given in percentage of how many in each group that were able to interact with a specific control.

Control name	**Youths**	**Young adults**	**Adults**	**Men**	**Women**
Command Button 1	100%	100%	100%	100%	100%
Command Button 2	100%	100%	100%	100%	100%
Command Button 3	100%	80%	100%	85,7%	100%
Command Button 4	80%	80%	80%	71,4%	87,5%
Command Button 5	40%	60%	40%	28,6%	62,5%
Command Button 6	40%	20%	40%	28,6%	37,5%
Link 1	60%	100%	100%	100%	100%
Link 2	20%	60%	80%	85,7%	25%
Link 3	20%	60%	80%	85,7%	25%
Link 4	0%	20%	0%	14,3%	0 %
Check Boxes Column 1	100%	100%	100%	100%	100%
Check Boxes Column 2	60%	80%	100%	100%	62,5%
Check Boxes Column 3	20%	20%	40%	42,9%	12,5%
Check Boxes Column 4	0%	20%	80%	42,9%	25%
Slider 1	100%	100%	100%	100%	100%
Slider 2	100%	100%	100%	100%	100%
Slider 3	100%	100%	100%	100%	100%
Slider 4	100%	100%	100%	100%	100%

Table 5.8: The numbers show how many percent of the participants that managed to interact with the controls in Task 5. They are divided into user groups and sexes.

Chapter 6

Discussion

In this chapter the results and method used are discussed. The first part of the chapter is divided into subchapters according to the task results presented in Chapter 5. The last part are devoted to the discussion of task independent results and our method. Research limitations are also presented.

6.1 Task 1 - Playing a Video Game

Task 1 was intended as a warm-up exercise for the users to get accustomed with gaze interaction. See a description of the demo application in Chapter 3.4. The task is described in Chapter 4.3.1 and Chapter 5.1 presents the results.

As can be seen from the result chapter, Chapter 5.1, it was observed a habituation process among the users as they got accustomed to the technology. We observed that the habituation process decreases as the knowledge about the technology increases. When new technology is presented to users, some challenge the technology, testing its limitations. Others have more ”respect” for the technology, and is more conservative. They merely follow the path in front them, carefully using only what is necessary to complete their task. These different approaches should be taken in consideration when designing gaze interactive programs. Even though we are not to prepare a guideline based on this observation, we recommend arranging for both a steep and a slack learning curve when developing gaze interaction software.

A few of the participants commented that they felt it was unnatural to control the UFO and press buttons with their eyes (see Chapter 5.1). This is natural. The eyes are normally used as a passive medium to receive visual impressions, not to interact with the surrounding world. To do so in this task, was probably something few have tried before.

There are not enough findings in this task to conclude with any guidelines. More extensive testing will be required to give guidelines for development of gaze interactive games. The task was used as a warm-up exercise, and it was found that the majority of participants enjoyed it as well as their knowledge around the eye tracking technology and its limitations seemed to increase. We argue they were better prepared for the following tasks due to this exercise, and recommend the use of warm-up exercises in future gaze interaction testing.

6.2 Task 2 - Exploring a Picture Gallery

In Task 2 the participants were asked to browse a picture gallery. After opening The Picture Gallery Application (see Chapter 3.5 for implementation details), they were supposed to navigate between pictures and use the zoom functionality by leaning their head back and forth. There were five subtasks to be carried out. They consisted of locating pictures in particular and zooming in on them.

As shown in Figure 5.1 from Chapter 5.2 , all of the participants accomplished all subtasks. By using a 95 % confidence interval, we get a lower limit of approximately 95 %. This imply that at least 95 % of the Norwegian population will be able to operate a picture gallery such as the one implemented in the Discovery software. This result seems promising for the future of gaze controlled picture galleries.

The picture gallery used in Task 2 was tailored specifically for gaze interaction. The large buttons were supposed to make the navigation easy for the participants. From the results in Chapter 5.2 it is made clear that the size of 90 x 280 pixels is enough for all participants to be able to hit the buttons. However, more testing is required to give precise recommendations concerning picture gallery button sizes.

By leaning towards the screen the participants managed to trigger the zoom function, discovered what the penguins were talking about (see Task 5e, Chapter 4.3.2). The zoom function was assumed to be a natural way of enlarging and shrinking pictures of interest, just as one in the real world moves objects closer to investigate, or more distant for an overview (see explanation in Chapter 3.5). The testing results confirms this, as the majority of the users felt it was a natural thing to do.

Furthermore, the results show that one participant had a negative experience using the head gestures. He felt it was unnatural and unusual to interact this way. One could speculate if a longer habituation process would affect his opinion. Would it feel more natural after longitudinal use? Further testing would help determine this.

The remaining fourteen out of fifteen participants had a positive attitude towards the picture gallery and its functions. The participants' statements indicate that using head gestures as a way of enlarging and shrinking pictures is something they found intuitive and natural.

Our findings support earlier research within this topic. The *Lean and Zoom*-system from Harrison and Dey (2008) works in a similar fashion as ours. By using a camera that tracks the distance between two spots on the user's face (eyes are good candidates for such spots), the system calculates the distance to the screen as the user leans forward and the distance between the two spots increases. The system magnifies the content around the mouse cursor, and panning is done by moving the mouse. Results from one of their user studies indicates that people find the technique natural and intuitive. Nine out of the ten participants indicated that they believed the technique would increase their performance.

According to our results in Chapter 5.2, most participants found the use of head gestures in order to zoom a positive experience. They described it as natural and intuitive. Based upon these results we recommend the following:

G1: *For gaze interactive picture galleries, the head gesture* lean *can be used as an intuitive and natural way of zooming in and out of pictures.*

6.3 Task 3 - Doing Drag and Drop Operations

In Task 3, the participants performed gaze drag and drop operations with different feedback methods. In Chapter 5.3, the results were presented. In this chapter these results will be discussed with the intent to give design guidelines for gaze interactive systems. Please note that all figures that are referenced to in this chapter are located in result Chapter 5.3.

The two different feedback methods tested were *gaze marker* and *hover effect*. As seen from Figure 5.2, the participants showed a preference toward the hover effect (60 %) compared to the gaze marker (33 %). This preference is however not strong enough for us to draw any conclusions. As seen from Table 5.1, the confidence intervals are widespread and inconclusive. The preference towards the hover effect could reach as high as approximately 75 %, but also be as low as 45 %. Based on this, it is difficult to give any precise recommendation regarding what feedback method to use. All participant managed to use both, but disagreed on which one was the best to use. The participants statements revealed several weaknesses with both the gaze marker and the hover effect. By tending to these weaknesses the numbers might change. Based on the participant's statement, a larger gaze marker, a less conspicuous hover effect and a better filter (less shaking) for both methods would be preferable. We recommend further research in this area.

Without being a part of either the task or the test leader's questions, as many as 67 % commented negatively on the release time, while finding the pick-up time appropriate. This number is interesting, and could even be higher if all participants had been asked to comment on pick-up and release times. With 67 % struggling due to a high release time, it was found by using a 95 % confidence interval that minimum 50,8 % of the population would always want a lower release time than pick up time. The remaining 33 % of the participants did not comment negatively on the release time but neither positively towards it. Since everyone of those who commented on the release time wanted it to be shorter, we can use this data to provide a guideline. That is, nobody wanted a higher release time, or commented that the release time was appropriate. 67 % wanted it to be lower, the remaining 33 % did not mention it at all. We argue that lowering the release time would reduce both what in the results (Chapter 5.3) is referred to as user impatience and help users with poor calibration. A lower release time will help users with poor calibration since if the gaze is prone to jitter, a lower release time would increase the chance for the gaze hitting the desired spot.

Even though more testing will be required to give a specific pick up or release time value, we can conclude that the release time should always be lower than the pick up time.

G2: *When designing gaze interactive drag and drop operations, use a shorter release time than pick-up time.*

As many as 80 % of the participants answered that even if available they would *not* use gaze interaction to organise their pictures. They argued mainly that it was both slower and more cumbersome than ”just using a mouse”. Let us discuss some of the causes for these arguments. First, do training play a role in this case? As revealed in Chapter 4.2, the participants were somewhat more experienced than the national average in the use of computers. This would imply that they have some experience with doing drag and drop operations using mouse interaction. Would the results have been any different if one tested using participants with no or equally as much mouse experience as with gaze interaction experience? Secondly, are the pick-up and release times responsible for the operation to be perceived as slow? As described in Chapter 2.1.2, eye movements are much faster than mouse movements, implying that the gaze movement itself (excluding the pick-up and release times) would be faster with gaze interaction. How would a lowering of these values affect the user experience? Thirdly, the drag and drop operation is a simulated realistic physical operation. One is asked to literary ”grab, move and drop” something. Would such a physical operation perhaps be something that the average man would find more connected to a mouse operation (using the hand), than a gaze operation (using the eyes)? To answer all these questions, more testing is needed. The discussion regarding which interaction techniques are the most suited for gaze interaction will be continued in Chapter 6.6.2.

6.4 Task 4 - Browsing a Web Page

In Task 4, the participants were asked to actively scroll as well as read a text in Discovery's browser application. The task consisted of four subtasks, in which the participant should read several paragraphs with automatic scrolling as well as actively scrolling between them. See Chapter 3.7 for a description of The Browser Application, Chapter 4.3.4 for a description of the task and Chapter 5.4 for the task results. This chapter presents a discussion of these results. The chapter is divided into three parts; first is a discussion of the habituation process, secondly a discussion of the automatic scrolling while the last part addresses the active scrolling.

6.4.1 The Habituation Process

As seen in Chapter 5.4, all fifteen participants solved the task successfully and we saw a habituation process among everyone. During the reading of the first two paragraphs the majority of the participants felt uncomfortable and distracted by the automatic scrolling function. They noticed that the text was scrolling while they read, and could not entirely concentrate on the reading. See quotes in Chapter 5.4. However, after a short habituation process they found the passive scrolling natural and comfortable. We thus see that this is a functionality that users quickly adapts. Although it is an interesting observation, we do not have sufficient data to give any specific guideline regarding the habituation process. Instead we will recommend further testing and research.

The participants' active scrolling performance also improved as they got used to it. The first times they tried to actively scroll upwards and downwards, they felt the scrolling jagged and that it on several occasions stopped. This led to a feeling of the scrolling slowing down. A further discussion of this jagged experience is explained in Chapter 6.4.3. As they got more accustomed to the scrolling, they felt they had more control and that they understood its mode of operation. We see that there exists a need for understanding how the active scroll function actually works to fully utilize its potential. This effect was also noticed during the participants' first experiences with gaze interaction, as discussed in Chapter 6.1. As we do not have sufficient results regarding this effect we can not give any guidelines regarding the habituation process for the active scrolling. Also here we will instead recommend further testing and research.

6.4.2 Reading

As seen in Chapter 5.4, reading with a passive scrolling function can be regarded suitable for gaze interaction. Fourteen of fifteen participants found it both easy and natural. Also, thirteen participants found it to be a relaxing experience. Two of the participants felt it was a stressful reading experience, and one felt it both stressful and unnatural. Even though the quotes from this person were registered as stressful and unnatural, respectively, we clearly see an indication in the participant's answers that his opinion is not entirely negative towards the gaze-operated scrolling. He says that he would get used to it, and it actually was enjoyable to not have to press anything to initiate the scroll command. The remaining participants were positive, and some were really enthusiastic about the functionality.

Chapter 5.4 reports that thirteen of fifteen participants found the speed of the automatic scrolling to be convenient. Some said that they felt the scrolling speed was automatically adjusted to their reading speed, even though there is no such functionality implemented. By investigating the video material we see that the reason for this could be that when the participant was struggling to read a difficult word the scrolling function would stop due to the gaze leaving the invisible read area. When the participant continued reading, the gaze reentered the read area and the text continued to scroll (See Figure 6.1 for a video screen capture of this).

As early as in 1984, a project with name *Erica* was started at the University of Virginia. The Erica computer workstation was equipped with imaging hardware and software and tracked users' gaze. The system had several gaze interaction applications, such as leisure games, a word processor, and a text reading application. Even though the system indicated promising results during the development phase, no extensive testing has been reported (Hutchinson et al., 1989). A more recent research was done by Kumar (2007) who investigated gaze operated scrolling techniques. In his dissertation Kumar presents three different scrolling techniques. We will here discuss two of them in light of our own findings. The first technique was called *Eye-in-the-middle*. By using this technique the scrolling measured the user's reading speed while dynamically adjusting the rate of the scrolling to keep the user's gaze in the middle third if the screen. This technique is similar to ours, however the read area was put higher on the screen, and it was larger. No test results were reported for this method though. An illustration of this scrolling technique can be seen to the left in Figure 6.2. The second technique, called *Smooth scrolling with gaze-repositioning*, acted differently. When a user's gaze dropped below a start threshold, the document started to scroll up slightly faster than the user's reading speed.

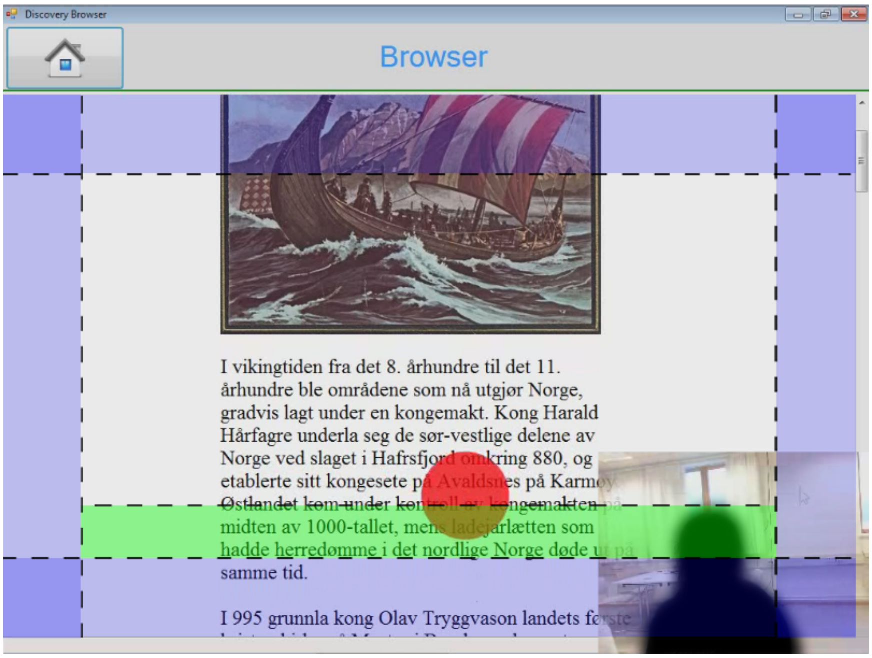


Figure 6.1: The scrolling function stops when a participant reads a difficult word, since the gaze falls above the read area. The large red circle indicates that the participant has looked at the same area for some time.

This forces the user to read increasingly higher on the screen, until the gaze reaches a stop threshold where the scrolling is stopped. Then the text is stationary until the gaze drops below the start threshold again. An illustration can be seen to the right in Figure 6.2.

Kumar tested this scrolling technique with ten participants in a two part study. The first group (group 1) was given no explanation regarding how the scrolling worked, while the other group (group 2) was given both explanations and time to test it in advance. By having the test subjects answer a 7-point Likert scale (subjects rating a statement with points from 1 to 7, where 1 means *disagree* and 7 means *agree*) the results showed that group 2 gave a higher score to the system than group 1. This result is similar to our observations of the active scrolling function. Both groups would like to use this approach to read paper/text on a website with a mean score of 4,3 (group 1) and 4,4 (group 2). Group 1 gave a mean score of 4,8 for the statement that they were able to read comfortably, while group 2 gave 4,9. These results are promising towards using automatic scrolling when reading text on-screen. Kumar says *"...subjects may be comfortable with reading moving text with practice."*, and further concludes his chapter with:

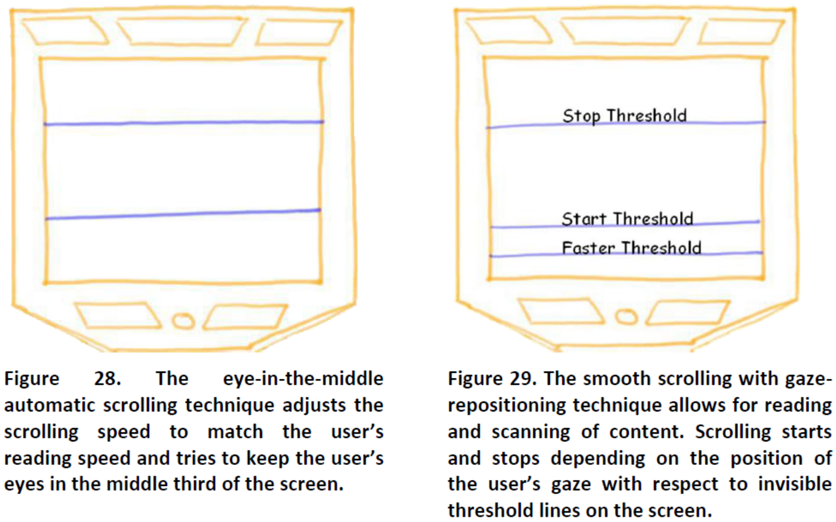


Figure 6.2: Two scrolling techniques by Manu Kumar. No results were reported for the technique to the left. (Kumar, 2007)

"...gaze-based scrolling techniques will increase in importance and provide users with a natural alternative to current approaches.".

Since 93 % of our participants found it easy and natural to read with automatic scrolling (see results in Chapter 5.4), and based upon the above findings, we hereby propose the following guideline for automatic scrolling:

G3: *In gaze interactive browsers, use automatic scrolling to provide a natural and easy reading experience.*

6.4.3 Active Scrolling

The majority of the participants found the active scrolling jagged and interruptive the first times they used it. After a short habituation process, the participants got more control over the scrolling as they understood how it worked. As seen from result Chapter 5.4, this resulted in that 87 % of the participants found active scrolling easy and natural, while 33 % of the participants found it stressful. The test leader briefly explained how the scrolling was implemented by saying that if the participant looked down on the document a downward scroll was initiated, and by looking at the top the document an upwards scroll was initiated. The problem arose when participants were told to scroll down to find a specific headline in the text, or a specific paragraph. To be able to scroll the text they needed to look at the invisible scroll area at the bottom or the top of the document, and to find the specific headline (or paragraph) they had to skim the text using their gaze. When their gaze left the scroll area to skim the text for the desired target, the scrolling stopped instantly and the scroll speed was reset to default speed. If the desired target was not found in the text, they had to continue scrolling to find it, and thus activated the scrolling again. This led to a jagged and interrupted scrolling experience.

In comparison to using a mouse, one does one thing with each ”device”. One can skim the text with one’s gaze while the hand uses the mouse or keyboard to scroll. Thus no problem occurs. We suggest further research regarding this issue.

The majority of the participants said they preferred the automatic scrolling over the active scrolling. Still, thirteen of fifteen found it easy and natural to scroll actively. From our point of view, it should be intuitive that when one looks down in a document, one would like to view some more text, and vice versa at the top. We thus consider the active scrolling as not affected by the Midas Touch problem (read more about the Midas Touch problem in the background chapter, Chapter 2.1.2).

As the majority of the participants felt the active scrolling technique as jagged and interrupting, we will not propose a specific guideline. However, as 87 % of the participants felt it was an easy and natural scrolling experience we recommend further research of this interaction technique.

6.5 Task 5 - Interaction With Different Controls

In Chapter 5.5 we presented our Task 5 findings. In this chapter they are discussed with the intent to give guidelines for designing gaze interactive controls for Windows. Please note that all figures that are referenced to in this chapter are located in result Chapter 5.5.

6.5.1 Command Buttons

The first part of Task 5 dealt with command button controls. Participants were asked to interact with (press) differently sized command buttons until they became green (see Chapter 4.3.5 for more task details). The command button results from the *Command Buttons* subchapter in Chapter 5.5, will now be examined focusing on the following three topics;

- Command Button height - Comparison of Button 4 and 5.
- Task completion rate - Comparison of Button 2 and 3.
- Give command button design guidelines.

An interesting comparison exists between the results of Button 4 and 5. Button 5 is constructed according to the Microsoft Windows standards. Button 4 is identical in every way except of being 12 pixels taller. As seen from Figure 5.6 there is a significant drop in the task completion between these two buttons. This does not only emphasize the importance of adequate height when designing command buttons, but also the deviation from the Microsoft Windows standard. This deviation and its consequences will be further discussed in Chapter 6.6.3.

As can be seen from the results in Figure 5.6, all participants managed to gaze interact with command buttons 1 and 2. Using a 95 % confidence interval the interval stretches from approximately 95 % to 100 %. A lower interval limit of 95 % in this case implies that with 95 % probability, 95 % of the population will manage to click these buttons, which we find as an acceptable result. Button 3 however, had a task completion rate of 93 % and a lower confidence interval limit of approximately 81 %. This meaning that in worst case, approximately one out of five will not be able to click this button. This confidence level is in our opinion too low.

When it comes to the actual design properties, height seemed to be what affected the results. This might be natural since the height was tested from 80 down to 19 pixels, while the width was tested from 150 to 50 pixels.

The height of Button 2 and 3 was respectively 65 and 50 pixels, thus our height design recommendation will be contained within this interval. To be able to converge in on a more accurate height value we recommend further research with more testing in this interval. For now we will recommend a design guideline on the basis of the highest number of the interval. The guideline will thus reflect command Button 2 in Discovery:

G4: *For gaze interactive command buttons, use a minimum width and heigth of 125x65 pixels.*

6.5.2 Links

Subtask d) of Task 5 encouraged the participants to interact with links. Links of different sizes and types were to be pressed using gaze interaction. The results presented in the *Links* subchapter of Chapter 5.5 shows in general that users had a hard time hitting the links. The link sizes tested were based on the Microsoft Windows standard of 9 pt, with an increase of 4 pts in four steps (13 pt, 17 pt and 21 pt). As seen from Figure 5.7, only 7 % managed to hit the 9 pt link (Link 4). Using a confidence interval, we can say with 95 % certainty that between 5,8 and 7,5 percent of the population will be able to hit this link. This is most certainly too low for any recommendation.

We see a connection between the font size and task completion rate, but even the most successful link, Link 1, only had an completion rate of 87 %. With a confidence interval lower limit of approximately 72 % we also find this too low for any recommendation. Based on these findings our recommendation will be to further test with even larger font sizes. Another discussion will be if links at all are suitable for gaze interaction, considering they will have to be considerably enlarged. That discussion will be covered in Chapter 6.6.2.

The two rows of links in subtask d) set the *tooltip feedback* up against ordinary *underline feedback*. Even as a popular gesture with mouse interaction, tooltips performed poorly with gaze interaction. Since stealing the attention of the participants when popping up, it was found distracting or annoying by all participants. We find the lower confidence interval limit of approximately 95 % as acceptable, and is therefore able to give the following simple guideline:

G5: *For gaze interactive links, do not use tooltips.*

6.5.3 Check Boxes

In subtask 5f), the participants were asked to select (check) gaze controlled check boxes. The boxes were arranged in columns with different design properties. From Figure 5.9 we see the overall participants' task completion. Column 1 and 2 both have check boxes with the same width and spacing, but a difference of 10 pixels in check boxes' hit area height. This difference is interesting since all participants managed to select the check boxes in column 1, while only 80 % managed selecting the ones in column 2 (Figure 5.10). Considering the lower confidence interval limits, we see that column 1 gives an acceptable limit of approximately 95 % in contrast to column 2 with a lower limit of approximately 64 %. Based on these results we find that gaze controlled check boxes should be designed with a minimum hit area height between 30 and 40 pixels. We will give our recommendation based on the highest number, but also recommend further research.

G6: *For gaze interactive check boxes, use a hit area with a minimum height of 40 pixels.*

Another interesting comparison is between column 3 and 4, as seen in Figure 5.10. They have the same height and width but different hit area spacing, and the results shows little difference in task completion. Only 27 % managed to select the check boxes with a massive spacing of 17 pixels in column 3, while 33 % accomplished selecting check boxes with the Microsoft Windows recommended spacing of 3 pixels in column 4. The small difference between these columns indicate that a new minimum spacing recommendation for gaze interaction is not necessary. It is a matter of fact that actually one more participant managed to select the check boxes where the spacing was smaller. Based on these results we recommend to keep the current Microsoft Windows standard of minimum 3 pixels actual spacing (7 pixels visual) also for gaze interactive check boxes.

G7: *For gaze interactive check boxes, use a hit area spacing of minimum 3 pixels.*

6.5.4 Sliders

In the last subtask, 5h), the participants were asked to set today's date using four sliders with different design properties. Even though all participants accomplished the task (see Figure 5.11), their opinions were divided. From Figure 5.12 it can be seen that 7 % (one participant) preferred Slider 1 while 53 % preferred Slider 2. Slider 1 was constructed according to the current

Microsoft Windows standard with one label centred under the slider and no delay. Slider 2 was constructed as Slider 1 apart from having multiple labels; one under each tick mark. Regarding the labels, the participants stated among others that the single value label of Slider 1 led to unintentional movements and that use of multiple labels in Slider 2 led to more text displayed on-screen. This considered, the comparison of the two sliders in Figure 5.13a indicated that 88,3 % would prefer Slider 2 over Slider 1. As seen from a 95 % confidence intervals in Table 5.7, in a worst-case scenario for Slider 2 (lower limit), 73,9 % will still prefer it over Slider 1. The upper confidence interval limit of 100 % indicates that in the best-case scenario, all will prefer Slider 2 over Slider 1, implying they will prefer using multiple labels over using a single label. Based on these results we will recommend using multiple labels when designing sliders for gaze interaction. We argue that this will lead to more gaze friendly sliders that are easier to control, in spite of the extra text required on-screen.

G8: *For gaze interactive sliders, display a value label under each tick mark.*

Slider 3 was constructed just as Slider 2 apart from having a delay functionality. As seen from the comparison in Figure 5.13b, the majority of participants preferred the non-delayed slider, Slider 2. The delayed slider provided a feeling of more control, but the trade-off with speed was not welcomed by the participants. The translated statement regarding Slider 3 illustrates this point: *"This one is too slow, makes me want to drag it with my hand instead."* It appears that the participants rather would make some errors in high speed and then go back and correct them, than using a slower speed and not making the errors in the first place. These results indicate that using a delay functionality in sliders are something to avoid. That is delays of 500 ms. By this number being so high, the speed reduction also became high. We will not give any specific guideline against using delay functionality in gaze operated sliders, but rather recommend testing of lower delays. This might tip the control-speed trade-off in an other direction, or at least confirm that the delay functionality actually is unsuitable in gaze interactive sliders.

In the comparison of Slider 3 and 4, hit area height was the only variable. As seen from Figure 5.14, approximately 68 % preferred Slider 4 with a height of 75 px over Slider 3 with a height of 45 px. It was observed that the larger hit area of Slider 4 naturally picked up some of the gazes that fell outside the hit areas of the other sliders. This led to a better experience for especially the participants with poor calibration. This considered, all participants managed to use all sliders. If we look at the cost of having a larger hit area, it would mainly be the hit area taking up more space and the slider being easier to hit unintentionally. The hit area taking up more space will not be a problem as long as the slider hit area is limited to the control itself. The tick marks and value labels are already taking up space in addition to the hit area. Figure 6.3 illustrates this. By extending the hit area over these elements, the complete on-screen control size would remain the same.

Figure 6.3: Showing the sliders with a default hit area (top) and with hit area extended downwards over the tick marks and value labels (bottom).

The slider being easier to hit unintentionally is a natural side effect of the slider being generally easier to hit. The spacing used around the control will play an important role in this matter. If the spacing is adequate, one's gaze should not interfere with the slider's hit area when operating other nearby elements. By extending the hit area as described above, filling out the control, it becomes similar to other controls such as command buttons, check boxes, links etc. They all have hit areas defining their control size. The spacing needed between these controls will therefore be the same. We recommend further research to give specific recommendations for spacing of gaze interactive controls. Based on the results, we give a split recommendation. Since all participant managed using all sliders, we find the 45 px hit area suitable to serve as a minimum height recommendation.

G9: *For gaze interactive sliders, use a minimum hit area height of 45 pixels.*

Based on the results of the Slider 3 and 4 comparison we find that a larger hit area is preferable, and through the discussion we found that it comes with little costs. Assuming that one uses default gaze interactive control spacing, one should extend the hit area downwards to cover tick marks and value labels.

G10: *For gaze interactive sliders, increase the hit area height downwards to cover tick marks and value labels.*

6.6 Other Discussion

This chapter presents a discussion of results that has been found independently of any specific task. It focuses on common trends and tendencies, discuss what gaze interaction is most suited for as well as how current software must be redesigned in order support gaze interaction in the future. Finally, we study if sex and age have been influencing the result.

6.6.1 Habituation Process

As described in the previous discussion chapters (Chapters 6.1 - 6.5) we observed a habituation process in all user tasks. The participants quickly adapted to the new interaction style.

Jacob and Karn in (Jacob and Karn, 2003) found that:

> *"Operating" the eye requires no training or particular coordination for normal users; they simply look at an object. The control-to-display relationship for this device is already established in the brain.*

Our observation confirms this statement. Even though the gaze technology was a new and unknown way of interaction for most of the participants, they quickly understood its mode of operation and completed their tasks. Even though the habituation process proved to be short, it was seen noticeable changes in the results before and after it. It was thus necessary to ensure accurate results. In future gaze interactive user testing, we recommend adjusting the time given in each task to compensate for a natural habituation process.

6.6.2 Gaze Interaction Suitability

From the testing of four different controls and four different user tasks we see differences in the perceived usability. Firstly, as can be seen in Chapter 5.2 and 6.2, the majority of the participants felt it was natural and intuitive to lean forward and backward to zoom in and out of pictures. This indicates that gaze interaction is suited for zooming. As have been discussed in Chapter 6.2 we believe this is related to the fact that the leaning mimics the natural movements one does when one wants bring an object closer or more distant. The zooming happens instantly. That is, as soon as the user moves.

Secondly, the same trend is seen when it comes to the automatic scrolling function. All users found the automatic scrolling function natural and easy.

This may be related to the minimal effort required. As soon as the gaze enters the reading area (see Figure 3.19 in Chapter 3.7.1) the text instantly starts to. The automatic scrolling is therefore considered suited for gaze interaction.

In contrast to the two tasks mentioned above is the gaze interactive drag and drop operation. The testing revealed that it was perceived as more cumbersome than the two other tasks. The observations showed that it was challenging to release the object at the desired target. In contrast to the two tasks above where the systems responded instantly, this did not and was perceived too slow. Furthermore, dragging and dropping objects mimics a physical operation of lifting something up, moving it to another location and dropping it. Such a physical operation is presumably more natural to do with a mouse (using hands) than with gaze (using eyes). Consequently, the testing indicates that drag and drop is not suited for gaze interaction.

Regarding the tested controls, the results show that none of the controls with the recommended design properties from the UX guidelines are suited for gaze interaction. In general they become too small for this interaction style. This discussion will be continued in the next chapter.

6.6.3 Future Gaze Interactive Software

As seen in the Task 5 results in Chapter 5.5 and the Task 5 discussion in Chapter 6.5, current Windows controls will need a redesign before supporting gaze interaction. This can be seen in comparison to other interaction styles such as the touch interaction style described in Chapter 2.3.4. The chapter explains how the UX guidelines recommends how Windows controls should be adapted to support touch interaction. It describes a redesign covering basic properties such as sizing, layout and spacing, but also ways of usage and the utilization of interaction strengths and weaknesses.

For gaze interaction, a similar redesign is needed. Let us take command buttons as an example. Our testing resulted in a minimum size recommendation of 125 x 65 pixels for gaze interactive command buttons. This in contrast to the touch interaction recommendation of 40 x 40 pixels and the mouse interaction recommendation of 75 x 23 pixels. Through the discussion in Chapter 6.5 we found that the height property was what affected the results. When comparing the height property of the size recommendations above, we can see an 73.9 % (23 - 40 px) increase in size when redesigning from mouse to touch interaction. From mouse to gaze interaction one can see that an increase of 182.6 % (23 - 65 px) is needed. That is, with the current technology. One should note that as the technology gets more accurate, the size recommendations may be reduced. Figure 6.4 shows an example of how a basic Windows 7 window could change when applying the different interaction recommendations for command buttons.

As discussed above, a redesign of current Windows controls is necessary for them to be able to support gaze interaction. If the goal had been to make Windows gaze interactive, a major redesign had been required, similar to what has been done to support touch interaction. Such redesign will probably take time and be both expensive and complicated, making it important to explore other approaches as well. In the discussion of gaze interaction suitability in Chapter 6.6.2 we argued that gaze interaction is more applicable for some controls and operations than others. Some of the most successful operations such as the gaze interactive automatic scrolling would be possible to apply to current software without any visual redesign. It would also be possible to use it in collaboration with other interaction style (multimodal interaction). By combining different interaction style, exploiting their different strengths and advantages, one may end up with a better solution than with any one specific interaction style. This will be more closely examined in the upcoming subchapter.

Multimodal Interaction

To avoid a complete redesign of current software when adding support for gaze interaction, multimodal interaction could be a solution. By using gaze interaction in combination with other interaction styles, less changes would be required. One could provide support for gaze interaction in some controls and operations, and provide support for different styles in others. We argue that it would not only require less changes, but could also result in a more efficient interaction style than any one single interaction style. By utilizing the strength of one interaction style where another is weak, one could enhance both the efficiency and user experience of current software. Let us look at some examples:

- Gaze-Mouse Combination
  A known challenge with gaze interaction is to click, such as one does with a mouse (see Chapter 2.3.4). Our results indicated that gaze interaction is less accurate (required larger controls) than mouse interaction(5.5). However, as described in Chapter 2.1.2, eye movements are much faster than mouse movements. We argue that by combining gaze and mouse interaction, one should be able to harvest the strengths of both interaction styles. The speed of the gaze interaction and the precious and click-functionality of the mouse in-

teraction. Another example of utilizing the strengths of both gaze and mouse interaction, would be to use them for different operations. For example, our research shows that gaze interaction is well suited for scrolling activities, but struggles when used to hit web links (Chapter 5.4 and 5.5). For a web browser, a solution will thus be to use gaze interaction for scrolling and mouse interaction for clicking links.

- Gaze-Keyboard Combination
  By using gaze interaction in combination with the keyboard interaction style, we could also utilize the speed and comfort of gaze interaction together with a way of performing clicks. In this case different clicks could be achieved by pressing different keyboard buttons. In addition, the keyboard interaction style offers a strength in text inputs. Using the same example as above, one could still scroll using gaze interaction, while to interact with links, one could look at them while pressing a keyboard button. By interaction with at a text input field one could use the keyboard to write.

There are of course many more examples, both in combination with the mouse and keyboard styles and with others. Based on our results and discussion, we see that multimodal interaction in theory could solve many of challenges gaze interaction faces today. However, the discussion about whether multimodal interaction is the future for gaze interactive software or not, falls outside the scope of this project. We would recommend both testing of gaze interaction in combination with other interaction styles and further research in this area.

6.6.4 Age and Sex Influence

A few differences between both the age separated user groups and the sexes. As seen from Table 5.8 in Chapter 5.5, the results from the young adults and the adults are fairly similar, and follow each other across the different controls. We see the same pattern with the results from the youths as well, but they distinguish themselves by having somewhat lower rate of completion with the link controls.

Between the men and women the pattern, despite different values, is the same. The women had a lower completion rate with the links and check boxes, but a higher rate with the command buttons. Since the results were not consequently lower when interacting with different controls, it is hard to interpret the results. It seems that sex and age do not have any influence on gaze interaction, however with so few as five participants in each user group, it would be indefensible to jump to any conclusions. Larger tests and further research will be required before reaching a conclusion.

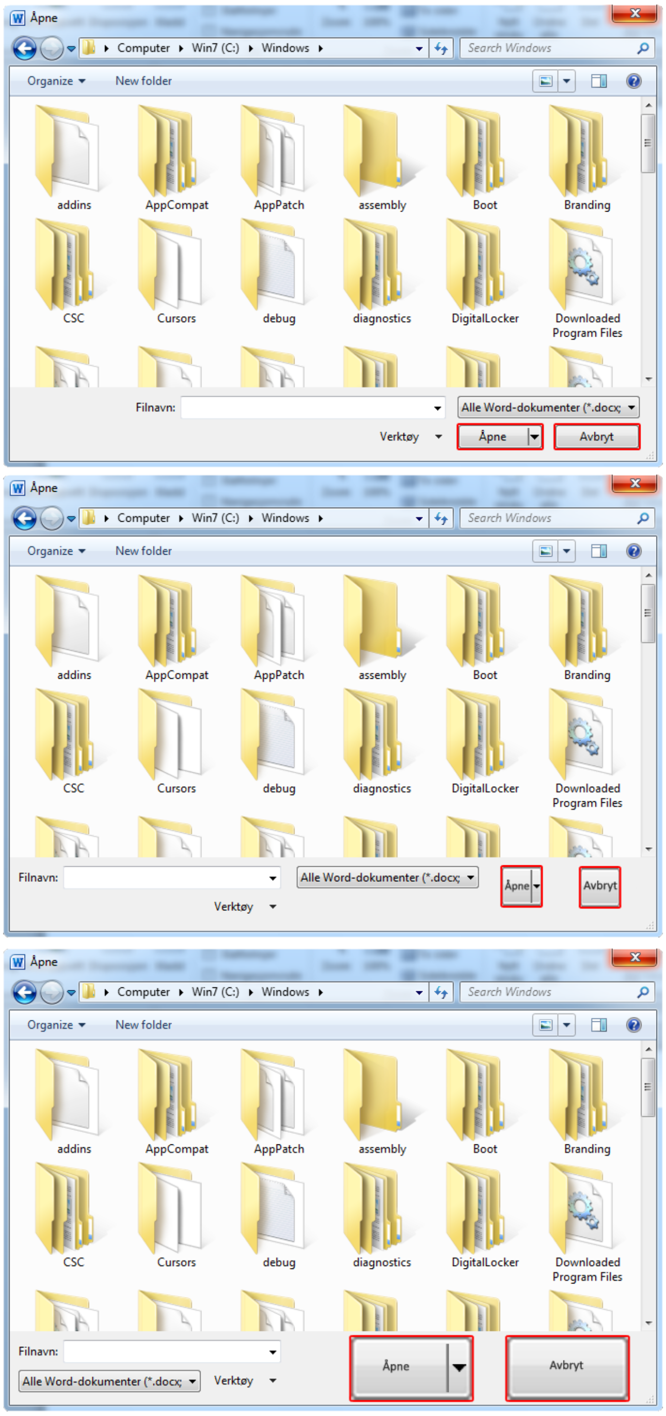


Figure 6.4: An example of how a basic Windows 7 window could change when applying mouse (at the top), touch (in the middle) and gaze interaction (at the bottom) recommendations to the two command buttons *Open* ("Åpne") and *Cancle* ("Avbryt").

6.7 Method Discussion

Our research method is thoroughly described in Chapter 4. This chapter discuss the validity and robustness of the method.

6.7.1 The User Testing

As stated in Chapter 4 we find user testing well suited as research method for this study. To be able to give guidelines for gaze interaction there is a need to do empirical research. Furthermore, quantitative data is preferable as one may want to achieve a certain user success rate for what is under test. For instance, one wants at least 95 % of all participants to be able to hit a button with their gaze. To be ensure this and to discover appropriate button size one needs to do empirical research that gathers quantitative data. User testing provides such data through observation of whether or not participants successfully clicks the button. As this study is limited in both time and resources, it was not possible to carry out a large usability study with many participants. A sample of fifteen participants was chosen for convenience. Even though the number of participants is few, we still observe clear results and tendencies among our test results. By using an appropriate statistical probability distribution these results are strengthen.

User testing provides the ability to collect qualitative data. The participants' comments, thoughts, reactions and discussions with the test leader has been taken into consideration while analysing the results. As mentioned in the discussions of the task results, these are important data to analyse. Direct quotations from participants generally gives research higher credibility, but care should be taken when analysing qualitative data in order not to alter its original meaning. In our opinion we have treated the collected qualitative data with such care.

During the test sessions there was a good tone and atmosphere between the test team and the participants. The test team consisted of young adults that strived to be social and forthcoming. We believe this made the participants feel comfortable thus sharing their true opinions about the system. However, user testing may create an artificial situation. This can be criticised as opinions may be influenced by the test situation, for instance by expressing what they believe the test team wants to hear rather than their real opinion.

The test team took turns being the test leader and the observer. This may have affected the user test results, as participants may have behaved differently around one test leader than the other. To accommodate for this effect, we carefully followed The Test Leader Guide (see Appendix C) when being test leader, and strived for consistency. We were aware that it is important that all participants have a common understanding before starting solving the tasks. In contrast, as a test leader one must also be flexible towards the different participants, and (in some sense) adjust one's behaviour to the situation. Although it is difficult to act similar in all situations during fifteen test sessions, we can not see any obvious correlation between this effect and our test results.

6.7.2 Participants

All participants were recruited from the Trondheim area. This weakens the basis of comparison when comparing with the Norwegian population. It was however recruited the same number of participants in each user group to maintain the basis of comparison.

The pre-test survey (see Chapter 4.2)showed that our participants were above average when it comes to computers and internet experience. Our participants experience with computers were 19 % above the national average and the experience with internet 12 % above. This may have influence on the results presented in this study.

6.7.3 Tasks

There was a progressive refinement of all tasks during several iterations. In our opinion the tasks were well written and easy to comprehend for participants at all ages. The tasks were tailored to the specific software they were meant to test, and they were concrete and short as recommended by usability experts and literature. There were observed no problems regarding the tasks during the test sessions, and none of the participants had any questions or remarks in relation to the tasks. Limitations regarding the tasks will be discussed in the Limitations chapter (Chapter 6.8).

6.7.4 The Test Facility

As described in Chapter 4.8 the participants were allowed into the observation room before the test. There were no thorough explanation about the observation room, only a quick statement like: *"This is our observation room where the observer observe what is happening during the test"*. This was done in order to mane the participant feel comfortable in the situation. There are varying practices in the industry regarding this. Some do not recommend showing the observation room to the participants as they may feel under surveillance and monitored, while others recommend to give the participants a brief introduction. In our opinion the participants were not shy or reserved due to the fact that they were being observed. A couple of the participants in the youths group were a bit shy, but we believe this had more to do with personality than the observation. In contrast, we also experienced very extrovert persons in the same user group.

The usability lab at NSEP in Trondheim may feel a bit cold, as it is located in a hospital and is mostly used for usability testing in the health sector. It does not reflect a typical private home environment. This may affected the results making the participants more reserved then normally. Sweets, fruit, coffee and similar were served to the participants in order to make a friendly atmosphere. To test how gaze interaction works in a typical user setting, one should preferably test it where it is supposed to be used. This was however not possible for us to do in this study.

6.8 Limitations

Several limitations exists in this study. This has to do with the limited amount of resources and time available, as well as practical considerations. Research results and provided gaze interaction guidelines has to be seen in light of these limitations. This chapter is devoted to the research limitations. Limitations regarding the research method (the user testing) has mainly been discussed in Chapter 6.7.

6.8.1 The Prototype

Only a subset of available Windows controls and operations were selected to be included in the prototype. The selection was a result of the work done in the pre-project (Raudsandmoen and Rødsjø, 2011). For a complete set of design guidelines for gaze interaction, all Windows controls and user operations must be covered.

The prototype is optimized for a specific screen resolution. Use of the prototype in other resolutions will require a recalculation of GUI element sizes.

Dwell-time is currently implemented as the only way to simulate the click of the mouse click. Implementing additional methods would enhance the prototype‘s testing capabilities.

The provided prototype does not reflect a standard Windows application, but has been created to act as a test bench for testing gaze interaction in controlled experiments. The prototype reduces the complexity found in ordinary computer programs, such as combinations of controls and procedures. This complexity has to be investigated in order to provide complete design guidelines.

6.8.2 The User Testing

A discussion regarding the research method was done in Chapter 6.7. This chapter explains additional limitations.

We have taken a scientific approach in the user testing. As the prototype is tailored to test specific elements and user operations, the tasks are also very concrete. The user was for instance told to click at certain buttons, to drag specific pictures from one side to another, and to read specific paragraphs. There were little to no freedom in what the user could do during the testing besides solving the tasks.

A known critique of user testing is that it only reveals users’ first impressions of a system, and does cover longitudinal use. This also applies in our user testing. Longitudinal use is important, but falls outside the scope of this project. We recommend further research on this topic.

Gaze interaction has been tested in isolation in this study. This means no multimodal interaction (interaction with several "devices" such as mouse, keyboard, gaze etc.) was tested. Although multimodal interaction fall outside the scope of this project, we encourage testing of gaze interaction in combination with other interaction styles in future research.

This study has measured users' effectiveness and satisfaction, but not efficiency. As recommended in the ISO 9241-11 (1998), this usability measure should also be taken into consideration. We recommend testing gaze interaction efficiency in further research.

6.8.3 Test Equipment

With an eye tracker accuracy of 0.5 degrees, and with a screen resolution of 1024x768 at an 19 inches screen, the accuracy is 12-19 px when the user is sitting respectively 50 to 80 cm from the screen. In practice this implies that the confidence interval for a target can have a circular spread of up to 38 pixels in diameter (Figure 6.5). This is based on the fact that if the user is looking at a point (1x1 pixel) target, the reading from the eye tracker may be off by up to 19 pixels in any direction. The results of this study needs to be seen in the light of this inaccuracy.

In addition to the inaccuracy of the eye tracker itself, a calibration is needed for each user. During the test procedure the user was not encouraged to start on the tasks before a sufficient calibration was achieved. Even though we strived for consistency among all users, we cannot guarantee that all users had similarly good calibrations. This may have affected the task completion rate and the user experience.

During the testing the users were required to keep their eyes in the middle of the eye tracker's spatial cube for the system to function optimal. Even small adjustments in the seating had impact on the accuracy, and may have influenced the test results.

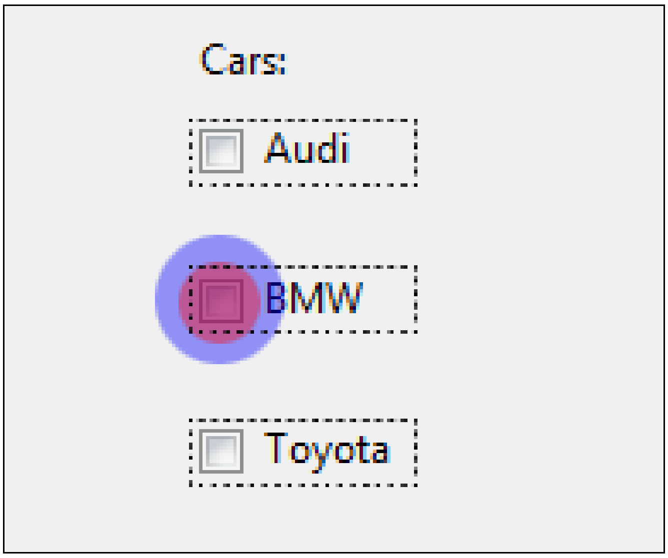


Figure 6.5: Eye tracker accuracy. The inner circle shows the accuracy at 50 cm viewing distance (24 px in diameter), while the outer circle shows for 80 cm viewing distance (38 px in diameter). Standard hit box height of a check box is 17 px, as shown by the dashed rectangle.

Chapter 7

Conclusion

This chapter concludes our findings in light of our research questions and research goal.

7.1 Conclusion

The purpose of this study has been to test the use of gaze interaction in common everyday computer tasks, with the intent to provide design guidelines for gaze interaction. This has been done by organizing a user test with fifteen participants, using a self-made gaze interactive software called *Discovery* and a Tobii X60 eye tracker.

Our research goal was to:

Suggest a number of empirically based design guidelines for gaze interaction in Windows 7.

To achieve this goal, three research questions were created:

RQ1: *How do users assess gaze interaction for solving a set of common everyday computer tasks?*

RQ2: *How does gaze interaction perform when used to solve a set of common everyday computer tasks?*

RQ3: *Which design guidelines can be suggested for gaze interaction in Windows 7, based on performance and user assessments?*

These questions have in our opinion all been answered through the results from the user test. RQ1 was answered by collecting user assessments in a series of *post-task semi-structured interviews*. These are gathered in Chapter 5. RQ2 was answered by registering the completion rate of the different tasks and subtasks. These are represented in the graphs of Chapter 5. Through the analysis and discussion of the user test results in Chapter 6, ten guidelines have been suggested thus answering RQ3. These are listed below.

The user test tasks covered were; playing a video game, exploring a picture gallery, doing drag and drop operations, browsing a web page and interacting with different Microsoft Windows controls. From a general perspective, we discovered that gaze interaction is more suitable for passive tasks such as reading with automatic scrolling, than for more physical tasks like doing drag and drop operations. To support gaze interaction, we found that current software will either require a major redesign or to be used in a combination with other interaction styles. From a more specific perspective, we suggested as mentioned above, ten design guidelines for gaze interaction.

These are:

G1: *For gaze interactive picture galleries, the head gesture* lean *can be used as a natural way of zooming in and out of pictures.*

G2: *When designing gaze interactive drag and drop operations, use a shorter release time than pick-up time.*

G3: *In gaze interactive browsers, use automatic scrolling to provide a natural and easy reading experience.*

G4: *For gaze interactive command buttons, use a minimum width and heigth of 125x65 pixels*

G5: *For gaze interactive links, do not use tooltips.*

G6: *For gaze interactive check boxes, use a hit area with a minimum height of 40 pixels.*

G7: *For gaze interactive check boxes, use a hit area spacing of minimum 3 pixels.*

G8: *For gaze interactive sliders, display a value label under each tick mark.*

G9: *For gaze interactive sliders, use a minimum hit area height of 45 pixels.*

G10: *For gaze interactive sliders, increase the hit area height downwards to cover tick marks and value labels.*

Note that our proposals are just a first step towards complete design guidelines for gaze interaction, thus they have to be seen in light of the limitations presented in Chapter 6.8. Also note that some of the guidelines recommends sizes in pixels. With the set-up used in this study one can calculate the physical size of these controls in millimetres by using a factor of 0.38; *mm = 0.38 * px.*

7.2 Further Work

During our analyses and discussions we have discovered interesting research topics that we recommend for further research. These includes among others:

Further testing of gaze interactive games (Chapter 6.1), and the use of head gestures (Chapter 6.2).

More extensive testing concerning different gaze feedback methods and pick-up and release times (Chapter 6.3), passive and active scrolling (Chapter 6.4), as well as control sizing, spacing and properties (Chapter 6.5).

We recommend also a closer study of the habituation process (Chapter 6.6.1), multimodal interaction (Chapter 6.6.3) and longitudinal use of gaze interaction (Chapter 6.8).

Bibliography

Anliker, J. (1976). *Eye Movements and Physological Processes*, chapter Eye movements: On-line measurement, analysis, and control, pages 185 – 199. Lawrence Erlbaum Associates.

Apple (2012). Mac os x human interface guidelines. `https://developer.apple.com/library/mac/#documentation/UserExperience/Conceptual/AppleHIGuidelines/Intro/Intro.html`, month = April.

Beinhauer, W. (2006). A widget library for gaze-based interaction elements. In *ETRA: Eye Tracking Research and Applications Symposium*, pages 53 – 53. ACM Press.

Bolt, R. A. (1981). Gaze-orchestrated dynamic windows. *Computer Graphics*, (15):109 – 119.

Bolt, R. A. (1982). Eyes at the interface. In *Proceedings of the ACM Human Factors in Computer Systems Conference*, pages 360 – 362.

Dodge, R. and Cline, T. S. (1901). The angle velocity of eye movements. *Psychological Review*, 8:145 – 147.

Fitts, P. M., Jones, R. E., and Milton, J. L. (1950). Eye movements of aircraft pilots during instrument-landing approaches. *Aeronautical Engineering Review*, 9(2):24 – 29.

Glenn, F., Iavecchia, H., Ross, L., Stokes, J., Weiland, W., Weiss, D., and Zaklad, A. (1986). Eye-voice-controlled interface. In *Proceedings of the 30th Annual Meeting of the Human Factors Society*, pages 322 – 326. Human Factors Society.

Google (2012). Android design. `http://developer.android.com/design/index.html`.

Harrison, C. and Dey, A. K. (2008). Lean and zoom: Proximity-aware user interface and content magnification. In *CHI 2008 Proceedings - I am here. Where are you?*, pages 507 – 510.

Hutchinson, T. E., White, K. P., Martin, W. N., Reichter, K. C., and Frey, L. A. (1989). Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems, Man and Cybernetics*, (19):1527 – 1534.

ISO 9241-11 (1998). Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability.

ISO/IEC 25062 (2006). Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Common Industry Format (CIF) for usability test reports.

Jacob, R. J. K. (1990). What you look at is what you get: Eye movement-based interaction techniques. In *CHI'90 Proceedings*, pages 11 – 18. ACM New York.

Jacob, R. J. K. (1991). The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Transactions on Information Systems*, 9:152 – 169.

Jacob, R. J. K. and Karn, K. S. (2003). *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, chapter Eye Tracking in Human-Computer Interaction and Usbility Research: Ready to Delvier the Promises. Elsevier Science BV.

Javal, E. (1879). Essai sur la physiologie de la lecture. *Annales d'Oculistique*.

Judd, C. H., McAllister, C. N., and Steel, W. M. (1905). *Psychological Review, Monograph Supplements*, volume 7, chapter General introduction to a series of studies of eye movements by means of kinetoscopic photographs, pages 1 – 16. The Review Publishing Company.

Kumar, M. (2007). *GAZE-ENHANCED USER INTERFACE DESIGN*. PhD thesis, Stanford University.

Kumar, M., Klingner, J., Puranik, R., Winograd, T., and Paepcke, A. (2008). Improving the accuracy of gaze input for interaction. In *ETRA*

2008 - Proceedings of the Eye Tracking Research and Application Symposium, pages 65 – 68.

Lankford, C. (2000). Effective eye-gaze input into windows. In *ETRA Eye Tracking Research and Applications Symposium. Palm Beach Gardens*, pages 23 – 27. ACM Press.

Levine, J. L. (1984). Performance of an eyetracking for office use. *Computational Biology and Medicine*, (14):77 – 89.

Mackworth, N. H. and Thomas, E. L. (1962). Head-mounted eye-tracker camera. *Journal of the Optical Society of America*, (52):713 – 716.

Majaranta, P. and Raiha, J. K. (2002). Twenty years of eye typing: Systems and design issues. In *Preceedings of Eye Tracking Research and Applications (ETRA2002)*, pages 15 – 22.

Microsoft Corporation (2011a). Windows controls. `http://msdn.microsoft.com/en-us/library/windows/desktop/bb773173(v=vs.85).aspx`.

Microsoft Corporation (2011b). Windows user experience interaction guidelines. `http://msdn.microsoft.com/en-us/library/windows/desktop/aa511258.aspx`.

NetMarketShare (2012). Desktop operating system market share. `http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=0&qptimeframe=Y&qpsp=2011`.

Nielsen, J. (2012). Usability metrics. `http://www.useit.com/alertbox/20010121.html`.

NSEP (2012). Norwegian ehr research center. `http://www.ntnu.no/nsep/brukbarhetslaboratorium`.

OneStat.com (2012). Wide screen resolutions extremely popular. almost one third of internet users is using a wide screen according to onestat.com. `http://www.onestat.com/html/press-release-wide-screen-resolutions-extremely-popular.html`.

PARC, X. (2012). Xerox parc history. `http://www.parc.com/about/`.

Raudsandmoen, H. and Rødsjø, B. (2011). Exploring the Tobii X60 Eye Tracker as Input Device.

Statistisk Sentralbyrå (2012). Ikt-bruk i husholdningene. http://www.ssb.no/ikthus/.

Tobii (2012a). Eye tracking research. http://www.tobii.com/en/eye-tracking-research/global/.

Tobii (2012b). Flexible eye tracking - tobii x60 and x120. http://www.tobii.com/en/eye-tracking-research/global/products/hardware/tobii-x60x120-eye-tracker/.

Tobii (2012c). Tobii software development kit. http://www.tobii.com/eye-tracking-research/global/products/software/tobii-software-development-kit/tobii-software-development-kit/.

Tobii (2012d). Tobii studio eye tracking software. http://www.tobii.com/en/eye-tracking-research/global/support-and-downloads/?product=787.

Toftøy-Andersen, E. and Wold, J. G. (2011). *Praktisk Brukertesting*, volume 1. Cappelen Damm Akademisk.

Tognazzini, B. (1992). *Tog on Interface*, chapter User Testing on the Cheap, pages 185 – 199. Addison-Wesley Publishing Company, Inc.

Tong, H. M. and Fisher, R. A. (1984). Progress report on an eye-slaved area-of-interest visual display. In *Report No. AFHRL-TR-84-36, Air Force Human Resources Laboratory, Brooks Air Force Base, Texas. Proceedings of IMAGE III Conference.*

United States Deptartment Of Veterans Affairs (2011). Glossary. http://www.va.gov/trm/TRMGlossaryPage.asp.

Ware, C. and Mikaelina, H. T. (1987). An evaluation of an eye tracker as a device for computer input. In *Proceedings of the ACM CHI+GI'87 Human Factors in Computing Systems Conference*, pages 183 – 188. ACM Press.

Appendix A

The Binomial Proportion Confidence Interval

When estimating a variable in a population we can use something in statistics called *a proportion*. When SSB (Statistisk Sentralbyrå, 2012) in Norway states that 59 % of the Norwegian population have used online social networks in the second quarter of 2011, the 59 % is a proportion. Proportion is a part of a whole. It can be expressed as a fraction, decimal, or percentage. Proportions can also represent probabilities. In this case, if we randomly pick a Norwegian person it is 59 % chance that he has used online social networks in the second quarter of 2011.

But how has SSB calculated this proportion? SSB has calculated it from a sample, preferably randomly picked Norwegians. What follows in this appendix is an explanation on how to estimate an either-or variable in a population from a small sample size, with a given level of assurance.

A.1 The Binomial Distribution

The *binomial proportion confidence interval* is a confidence interval for a proportion in a statistical population. Binomial means that the outcome from the test is either a success or failure. When doing a user test each user task is given a success criteria, and the task is regarded as either a success or failure depending on if the user met the success criteria or not. The criteria for using a binomial distribution are:

1. There are two outcomes

2. Constant probability

3. Fixed number of trials

4. The trials are independent of each other

Let us take an example. Fifteen participants are given a task which involve clicking buttons. The task criteria for success is that a specific button gets clicked. The task has two outcomes, the user successfully clicks the button or not (1). No participant has any advantage over the other, so the probability of a participant clicking the button is constant (2). The number of trials is set to 15 (3), and all trials are considered independent of each other since the participants are tested separately(4).

A.2 Proportion Notation

The following symbols are commonly used when talking about proportions:

- p = symbol for population proportion
- $\hat{p}$ = symbol for sample proportion

With these definitions:

$$\hat{p} = \frac{X}{n} \tag{A.1}$$

$$\hat{q} = \frac{n-X}{n} \tag{A.2}$$

or

$$\hat{q} = 1 - \hat{p} \tag{A.3}$$

X = number of sample units that possess the characteristics of interest and n = sample size.

A.3 The Binomial Proportion Confidence Interval

There are several ways to compute a confidence interval for a binomial proportion but the normal approximation interval is the most commonly used and it relies on approximating the binomial distribution with a normal distribution. This approximation is justified by the central limit theorem.

To be able to construct a confidence interval about a proportion one must use the maximum error of estimate, which is shown in Equation (A.4):

$$E = z_{1-\frac{\alpha}{2}} * \sqrt{\frac{\hat{p}\hat{q}}{n}} \tag{A.4}$$

Confidence interval about proportions must meet the requirement that np ≥ 5 and nq ≥ 5. To estimate a confidence interval for the true population proportion, one use the sample proportion minus the estimate error as a lower threshold, and sample proportion plus the estimate error as an upper threshold. This is illustrated in the following equation:

$$\hat{p} - z_{1-\frac{\alpha}{2}} * \sqrt{\frac{\hat{p}\hat{q}}{n}} < p < \hat{p} + z_{1-\frac{\alpha}{2}} * \sqrt{\frac{\hat{p}\hat{q}}{n}} \tag{A.5}$$

where α is the error percentile, and $z_{1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ percentile of a standard normal distribution. For example, for a 95 % confidence level, the error α is 5 %, so $1-\frac{\alpha}{2} = 0.975$ and by a look up in the standard normal table we find that $z_{1-\frac{\alpha}{2}} = 1.96$. By solving Equation A.5 with these values and a sample proportion n, we can be 95 % sure that the true population proportion lies in the found interval.

Appendix B

Test Plan

This is our test plan that was made during the planning of the user tests. Our plan is inspired by the test plan recommended by Toftøy-Andersen and Wold (2011). During both the planning and the execution of user tests this document was frequently used. The test plan is translated into English.

B.1 Test Plan

B.2 Purpose of the User Tests

The purpose of the user tests is to get the participants - the users - opinions regarding the use of their eyes to perform common everyday computer tasks. The user test will be created such that the test team after the test, may give guidelines on the use of gaze interaction in Windows 7.

B.3 System used in the Test

The system used in the test is *Discovery v2.0*. See Chapter 3.1 for more information about the software.

B.4 Functionality that will be Tested

Below a list of the functionalities to be tested:
(All functionality is tested with gaze interaction)

- Gaming
- Navigating a picture gallery
- Using head movements to zoom in and out of pictures
- Dragging and dropping pictures
- Scrolling text up and down
- Reading text while the page scrolls
- Clicking buttons
- Clicking links
- Using check boxes
- Using sliders

B.5 Participants

The participants are to be recruited among the common Norwegian population. The users should have some basic experience with the use of computers and internet, and aged between 13 to 65. Due to practical reasons the participants will be recruited in the Trondheim area in Norway

B.5.1 Selection Criteria

1. The user has basic experience with use of computers
2. The user has basic experience with use of internet

B.5.2 Demographic Criteria

1. 5 users aged between 13 and 19
2. 5 users aged between 20 and 29
3. 5 users aged between 30 and 65
4. Approximately half of the users should be female and the other half male

This demographic criteria gives us three user groups; *youths* (13 - 19), *young adults* (20 - 29) and *adults* (30 - 65). The test team is interested in any differences in the use of gaze interaction between the user groups.

User Group	Name	Mobile phone	Email
Youths	Youth 1	...	...
Youths	Youth 2	...	...
Youths	Youth 3	...	...
Youths	Youth 4	...	...
Youths	Youth 5	...	...
Young adults	Young adult 1	...	...
Young adults	Young adult 2	...	...
Young adults	Young adult 3	...	...
Young adults	Young adult 4	...	...
Young adults	Young adult 5	...	...
Adults	Adult 1	...	...
Adults	Adult 2	...	...
Adults	Adult 3	...	...
Adults	Adult 4	...	...
Adults	Adult 5	...	...

Table B.1: Participants

Table B.1 has an overview of all participants.

Due to participant anonymity the user information has been depersonalised

B.6 Test Location

The testing shall be conducted at the Usability Lab at the Norwegian EHR Research Center (NSEP) in Trondheim. Below is the street address and a map (Figure B.1) that shall be sent to all participants together with the invitation. The red arrow at the map shows the building's entrance.

Street address:
Medical Technology Research Centre
Faculty of Medicine
Olav Kyrres gate 9
7030 Trondheim

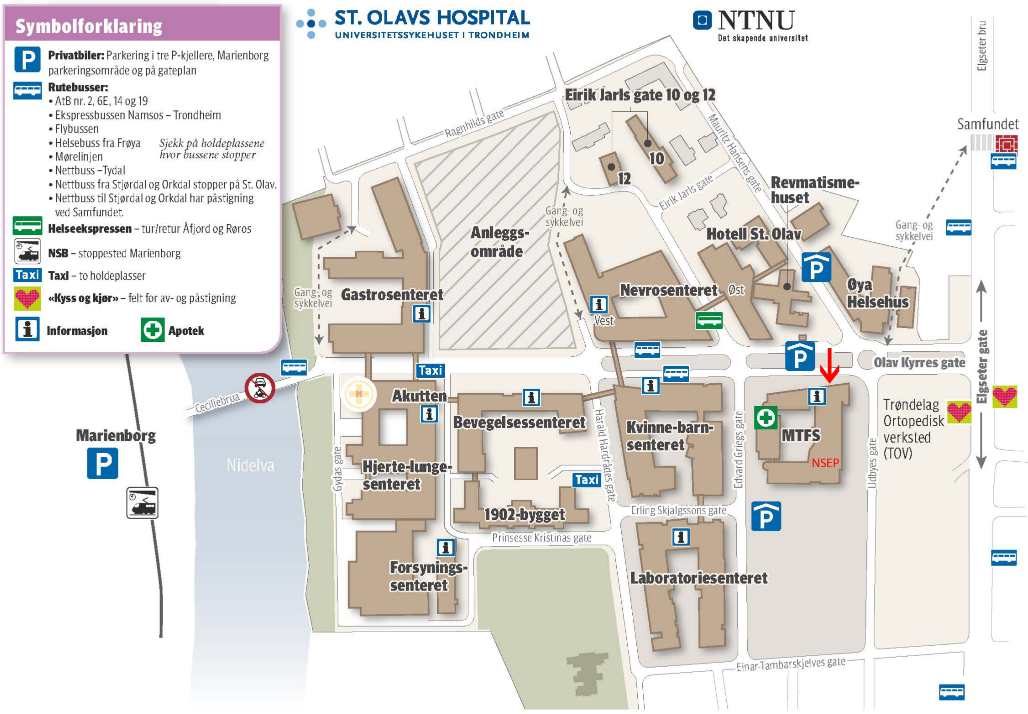


Figure B.1: Map for the test location

B.7 Test Environment

Below a list of the test equipment used in the test room. In addition to this there is equipment at the observer room such that the observer can record and observe the session.

- A PC with keyboard and mouse
- Microsoft Windows 7
- Discovery v2.0
- Tobii Studio
- 23" monitor with screen resolution 1024 x 768
- Tobii X60 Eye Tracker
- Video cameras
- Microphones

Date	Time	Name	Test Leader
20. Feb	09:00-10:30	Young adult 2	Børge
20. Feb	10:30-12:00	Adult 4	Håkon
20. Feb	13:00-14:30	Young adult 3	Børge
20. Feb	14:30-16:00	Youth 5	Håkon
21. Feb	09:00-10:30	Adult 1	Håkon
21. Feb	10:30-12:00	Youth 1 & Youth 2	Håkon
21. Feb	13:00-14:30		
21. Feb	14:30-16:00	Adult 5	Børge
22. Feb	09:00-10:30		
22. Feb	10:30-12:00		
22. Feb	13:00-14:30		
22. Feb	14:30-16:00		
23. Feb	09:00-10:30	Young adult 5	Håkon
23. Feb	10:30-12:00	Young adult 1	Børge
23. Feb	13:00-14:30	Adult 3	Børge
23. Feb	14:30-16:00		
24. Feb	09:00-10:30	Adult 2	Børge
24. Feb	10:30-12:00	Young adult 4	Håkon
24. Feb	13:00-14:30	Youth 3 & Youth 4	Børge
24. Feb	14:30-16:00		

Table B.2: Time Schedule (empty slots is free time)

B.8 User Tasks

See Appendix E

B.9 Time Schedule

Table B.2 is a schedule for the testing. All testing is done in one week.

Due to participant anonymity the user information has been depersonalised

B.10 Questions

In the introduction of the user tests the participants are asked if they have been to a user test before. If they have they are presumably more familiar with the concept *thinking aloud* and the test leader may thus talk about this more briefly. The users are also presented with a survey to get some participant background information (see Appendix D). After each task the users are asked some pre-defined questions, and the answers are written down by the observer. All these questions, those in the introduction and the task-specific questions can bee seen in Appendix C.

B.11 The Test Team

The test team consist of two persons and two roles. The test persons may exchange roles from test to test. The two persons are Håkon Raudsandmoen and Børge Rødsjø and the two roles in the test are described below.

B.11.1 Test Leader

The test leader is responsible to greet the participant and guide him or her through the test. The test leader use the Test Leader Guide as described in Appendix C.

B.11.2 Observer

The observer takes notes and records screen, video and sound during the user testing sessions.

B.12 Analysis

The analysis takes part after the user testing is completed, and is conducted by Håkon Raudsandmoen and Børge Rødsjø during the spring of 2012.

Appendix C

Test Leader Guide

This test leader guide is inspired by (Tognazzini, 1992). This document was used during each user test session to ensure that all the participants got the same information and were asked the same questions. This document was originally written in Norwegian, and is translated into English.

C.1 Test Leader Guide

C.2 Welcome and Introduction

- Welcome and introduce us
- At the observation room we have an observer
- Please serve your self with soda, coffee and snacks at any time
- Who are we and why have you been contacted
- Have you done user testing before?
- Explain what user testing is:
  - Explain the concept *thinking aloud*
  - It is not you that we are testing, it is the system
  - The tasks are created such that you will struggle to finish some of them, and they are created that way with a purpose
  - 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

    - I can not help you solving the tasks

- The results will be used to create guidelines so that it will be easier in the future to create computer programs that can be controlled with the eyes
- The test will last for approximately 45 minutes
- For your participation you will be rewarded with two tickets to the cinema

C.3 Informed Consent Form

- We will make sure that you will be kept anonymous. Your name will not be connected to the results from this test

C.4 Survey

- To get some background information about your computer and internet experience

C.5 Describe the Purpose With the Test

- To get results about how we should make computer programs easy to control with our eyes

Walk over to the eye tracker and the equipment

C.6 Describe the Equipment in the Room

- A PC, monitor and eye tracker
- We are recording the test with video cameras and microphones. This in order to be able to watch the session in retrospect.

C.7 Do You Have Any Questions Before We Start?

C.8 Explain the Calibration Procedure and Get Going

C.9 Give the Participant the Task Sheet

C.10 Task 2

- How did you feel about using head movements to control the picture gallery?

C.11 Task 3

- How did you feel about moving pictures with your eyes?
- If you were to organize your pictures on your home computer, would you prefer to use your eyes?
- What did you like the most; moving pictures with of without the hover effect?

C.12 Task 4

- How did you feel about scrolling with your eyes?
- What do you think about the scroll speed when scrolling up and down?
- How did you feel about reading with automatic scrolling of the page?
- What do you think about the scroll speed when reading?
- Would you describe this experience as:

  Stressful or relaxing?

  Easy or difficult?

  Natural or unnatural?
- If you could have used automatic scrolling on your home computer, would you have used it?

C.13 Task 5

- Do you prefer the links with or without tooltip?
- Which slider do you prefer the most?
- What did you like the most; moving pictures with of without the hover effect?

C.14 Test Closure

- Do you think this technology is something that you would have used in the future?
- Is there anything more you would like to add?
- As a reward for your participation I would like to give you two cinema tickets from Trondheim Cinema
- Thank you very much for your time! Your participation will hopefully affect coming computer programs!

Appendix D

Survey

The following pages contains the survey that was handed out to each test person. The survey collects key information about the users as well as their experience with computers and internet. The survey was originally written in Norwegian, but have been translated into English.

SURVEY

Empirically based guidelines for gaze interaction in Windows 7

Survey of technology experience

1) Age

☐ 10- 15 ☐ 16- 24 ☐ 25- 34 ☐ 35- 44 ☐ 45- 54
☐ 55- 64 ☐ 65- 74

2) Gender

☐ Female ☐ Male

3) Do you have a computer at home?

☐ Yes ☐ No

4) If yes, do you have access to the internet with this computer?

☐ Yes ☐ No

5) **Do you use the internet via a computer on any of the following places:** (set multiple marks if necessary)

☐ At work
☐ At school/university
☐ At the library
☐ At a café, restaurant or hotel with internet available
☐ At friends
☐ Other places: ______________________________

6) If you use the internet, what do you use it for?
(set multiple marks if necessary)

☐ Read newspapers
☐ Internet surfing
☐ Order tickets (eg. flight tickets, cinema tickets)
☐ Netshoping (eg. buy books, movies, music)
☐ Chatting, online debates and forum (talk with writing over the internet)
☐ Social networking sites (eg. Facebook, Twitter)
☐ Online banking (using bank services over the internet)
☐ Look up knowledge sources (eg. Wikipedia)

7) If you use PC, what of the following have you done?
(set multiple marks if necessary)

☐ Have copied files or folders
☐ Have used the cut and paste-function
☐ Have used spread sheets, eg. Excel
☐ Have transfered files between PC and other equipment
☐ Have changed configuration on software or PC
☐ Have used a programming language
☐ Have compressed files or folders

8) If you use internet, how often do you use it?

☐ Daily ☐ Weekly ☐ Monthly ☐ Never

Thank you so much for taking time to answer the survey!

Appendix E

User Tasks

This is the tasks the participants were given at the user tests. After each task the participants were asked some questions from the test leader. These questions are stated in Appendix B. The task sheet below was originally written in Norwegian, but has been translated into English.

E.1 Tasks

E.2 Task 1

a) Open the game *UFO*

b) Play through the first level

Go back to the main menu

E.3 Task 2

a) Open *Picture Gallery*

b) Go through the pictures until you find a picture of a koala

c) Zoom in on the nose of the koala

d) Find the picture of the penguins

e) Find out what the penguins are talking about

Go back to the main menu

E.4 Task 3

a) Open *Drag'n Drop*

b) Move the picture *Desert* over to the right side among the other pictures

c) Move the picture *Koala* over to the left side among the other pictures

d) Turn on *Hover Effect* and move *Desert* back to the left side

e) Move the picture *Koala* back to the right side

Go back to the main menu

E.5 Task 4

a) Open *Browser*

b) Read the first paragraph about *Viking Age to Middle Age*

c) Scroll down to *Democratic constitution in 1814* and read the first paragraph

d) Scroll up to *Decadence and Time of Danish Rule*, and read the second paragraph about the 400-year's night

Go back to the main menu

E.6 Task 5

a) Open *Controls*

b) Press the buttons until they become green (as shown on Figure E.1)

c) Press *Next*

d) Press the links, start with the largest one on the top row, storm.no

e) Press *Next*

f) Start from left and select as many of the check boxes as possible

g) Press *Next*

h) Set today's date by using the sliders

Go back to the main menu

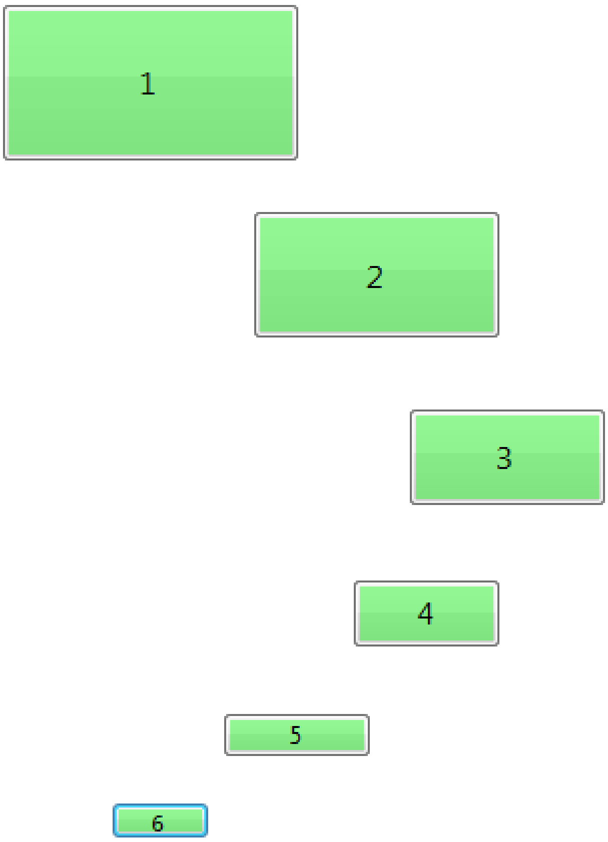


Figure E.1: Task 5b) Press the buttons until they become green

Appendix F

Observer Form

This observer form was used during the user testing. The observer sat in the observation room in the usability lab and took notes during each user testing session with both audio, video and live computer screen interface available. Most observations were written in short descriptive notes. Quantitative investigations were written with *OK* if the criteria of success was achieved for a specific task, and left blank if not achieved. The following document was originally written in Norwegian, but has been translated into English.

F.1 User test X

Date:
Hour:
User Group:
Observer:

F.2 General

General observations of the whole user test session. This could be calibration results, technical difficulties or errors, or other important notes or comments that does not fit in anywhere else

F.3 Task 1 (UFO)

General observations notes are written here

F.4 Task 2 (The Picture Gallery)

General observations notes are written here

Questions
How did you feel about using head movements to control the picture gallery?
User's answer is written here

F.5 Task 3 (Drag and Drop)

General observations notes are written here

Questions
How did you feel about moving pictures with your eyes?
User's answer is written here

If you were to organize your pictures on your home computer, would you prefer to use your eyes?
User's answer is written here

What did you like the most; moving pictures with of without the hover effect?
User's answer is written here

F.6 Task 4 (Browser)

General observations notes are written here

Questions
How did you feel about scrolling with your eyes?
User's answer is written here

What do you think about the scroll speed when scrolling up and down?
User's answer is written here

How did you feel about reading with automatic scrolling of the page?
User's answer is written here

What do you think about the scroll speed when reading?
User's answer is written here

Would you describe this experience as:
Stressful or relaxing?
Easy or difficult?
Natural or unnatural?

User's preferred answer is marked in bold font

If you could have used automatic scrolling on your home computer, would you have used it?
User's answer is written here

F.7 Task 5 (Controls)

OK *is stated for each control if the user achieves the success criteria for the specific control or task*

Button 1 -
Button 2 -
Button 3 -
Button 4 -
Button 5 -
Button 6 -

Link 1 (storm.no) -
Link 2 (google.no) -
Link 3 (ntnu.no) -
Link 4 (yr.no) -

Do you prefer the links with or without tooltip?
User's answer is written here

Check box column 1 -
Check box column 2 -
Check box column 3 -

Check box column 4 -

Slider 1 -
Slider 2 -
Slider 3 -
Slider 4 -

Which slider do you prefer the most?
User's answer is written here

Appendix G

Project Information Sheet

The following sheet was given to the parents of participants younger than 18 years old. The sheet explains the project background and purpose, what involves being a participant at the project and what happens about the information that the study collects. At the end of the sheet the parents give their consent about the participant's attendance in the project. The project information sheet is written in Norwegian.

Skriv til foresatte angående deltakelse i forskningsprosjektet:

Empirisk-baserte retningslinjer for blikkinteraksjon i Windows 7

Bakgrunn og hensikt
Øyesporingsutstyr gjør det mulig for en datamaskin å registrere en brukers øyebevegelser slik at den kan følge med på hvor brukeren har fokus på skjermen. Dette gjør det mulig å styre datamaskinen v.h.a. øynene. I forbindelse med vår masteroppgave på NTNU, forsker vi på hvordan det er å bruke slikt utstyr til å løse hverdagslige oppgaver på datamaskinen. Forskningen skal resultere i et forslag til retningslinjer for bruk av øyesporing i Windows 7.

Hva innebærer testen?
For å finne ut av hvordan mennesker synes det er å bruke øyne til å styre datamaskiner, ønsker vi å gjennomføre en test med et bredt utvalg testpersoner. Testen innebærer at testpersonen titter på en dataskjerm, mens et kamera fanger opp øyebevegelser. Dette gir testpersonen kontroll over datamaskinen, og muligheten til å styre et dataprogram kun ved hjelp av øyene. Vi vil observere og ta notater, samt gjøre lyd- og videoopptak under testen. Dette gjøres for at vi skal kunne vurdere resultatet av testen senere, for å forsikre oss om at utsagn og handlinger er blitt forstått på riktig måte. Selve testen vil finne sted ved Norsk Senter for Elektronisk Pasientjournal (NSEP), på St. Olavs Hospital.

Mulige fordeler og ulemper
Deltakelse i testen vil ikke innebære noe ubehag for testpersonen.
Deltakere vil motta to gavekort på Trondheim Kino som takk for hjelpen.

Hva skjer med informasjonen?
Informasjonen som registreres om deltakeren skal kun brukes slik som beskrevet i hensikten med testen. Videoene og lydopptakene vil bli slettet ved prosjektet avslutning, og prosjektslutt er satt til 31.06.12. All øvrig informasjon vil bli avidentifisert, lagret på data og behandlet konfidensielt. Dette betyr at alle opplysningene vil bli behandlet uten navn eller andre direkte gjenkjennende opplysninger.

Samtykke til deltakelse i studien

Jeg gir herved ____ tillatelse til å delta i forskningsprosjektet beskrevet ovenfor.

----------------   --

Dato   Navn foresatte (blokkbokstaver)

Underskrift foresatte

Appendix H

Informed Consent Form

The following form is the informed consent form that was used in the project. Every participant signed this sheet which informed them about the user test, their tasks, their ability to abort the test whenever they wanted, and how we kept the information anonymous. The informed consent form is in Norwegian.

Samtykkeerklæring

Jeg bekrefter herved at jeg ønsker å delta i brukertest av blikkinteraksjon. Testen er en del av masteroppgaven *«Empirisk-baserte retningslinjer for blikkinteraksjon i Windows 7»* av Håkon Raudsandmoen og Børge Rødsjø, utført ved Norges Teknisk- Naturvitenskaplige Universitet (NTNU) i Trondheim.

Testen innebærer bruk av øyesporingsutstyr som gjør opptak av mine øyebevegelser. Jeg vil bli bedt om å utføre spesifikke oppgaver som løses ved å se på dataskjermen, samt svare på enkle spørsmål.

Som testbruker har jeg rett til å avbryte testen når som helst uten begrunnelse. Jeg er anonym, og mine personalia og kontaktinformasjon skal ikke offentliggjøres eller brukes i annen sammenheng. Som kompensasjon for deltakelse mottar jeg to gavekort på Trondheim Kino.

Jeg samtykker i at jeg blir observert under brukertesten, og at det vil bli gjort opptak av lyd, bilde og skjerminteraksjon. Opptakene vil bli slettet ved prosjektet avslutning, og prosjektslutt er satt til 31.06.12. All øvrig informasjon vil bli avidentifisert, lagret på data og behandlet konfidensielt. Dette betyr at alle opplysningene vil bli behandlet uten navn eller andre direkte gjenkjennende opplysninger.

__________________________________ _____________________

Underskrift Dato

____

BLOKKBOKSTAVER