Designing a Paint Application for Eye Tracking Control Methods

BIRMINGHAM CITY
University

Ciaran O'Dwyer - S15111608

Supervisor: Dr Chris Creed

M.S. Advanced Computer Science

CMP7200 Individual Masters Project 2018/19

Abstract

With eye-tracking becoming a reliable method of control for computers, more applications need to accommodate the method of control. One major part of controlling applications is the ability to scroll. To improve scrolling functionality, two iterations were developed that used different methods to scroll a paint application list. Various timed tasks were developed to test scrolling control and speed to compare the two iterations. A/B testing showed that while users were able to perform tasks faster with a click-based approach, users found the hover version to be more intuitive. Both versions would require more development and testing; however, both have the potential to be used as scrolling methods.

Acknowledgements

I would like to personally thank my advisor Chris Creed for all the advice and help offered throughout the project. I would also like to thank Ellen Stone for all the helpful advice given during this project. I would also like to thank the Birmingham Lions for all the good times had during this project.

Table of Contents

Abstract **2**

Acknowledgements **3**

Table of Contents **4**

Table of Figures **6**

1.0 - Introduction **8**
- 1.1 - Introduction 8
- 1.2 - Background 8
- 1.3 - Aims 8
- 1.4 - Objectives 9
- 1.5 - Section Summary 9

2.0 - Literature Review **10**
- 2.1 - Introduction 10
- 2.2 - Disability Research 10
- 2.3 - Eye Tracking Technology - Functionality and Accessibility 11
- 2.4 - Using Eye Tracking for Application Navigation - Related Projects 12
- 2.5 - Conclusion 14

3.0 - Interface Research and Design **15**
- 3.1 - Introduction 15
- 3.2 - Research Method 15
- 3.3 - Interface Design 16
  - 3.3.1 - Interface Design Introduction 16
  - 3.3.2 - Clicking Iteration 17
  - 3.3.3 - Hover Iteration 18
- 3.4 - Section Summary 20

4.0 - Experiment Methodology **21**
- 4.1 - Introduction 21
- 4.2 - Method of Testing 21
- 4.3 - Testing Technology and Resources 22
- 4.4 - Method of Gathering Data 23

5.0 - Results **25**
- 5.1 - Time Data 25
- 5.2 - Observational Data 28
- 5.3 - User Qualitative Feedback 29
- 5.4 - System Usability Scale (SUS) 30

6.0 - Discussion **32**

7.0 - Conclusion **36**

8.0 - References **37**

9.0 - Appendices **40**

Table of Figures

Figure 1: Table showing the population percentage of adult Americans who suffer from a form of cognitive disability. (Houtenville et al. 2016) 10

Figure 2: Diagram of an eye and how the reflection is obtained. (Poole et al. 2006) 11

Figure 3: A sketched diagram of the process of the “Discrete Scrolling” method. The scrolling will trigger once the user’s eye is below a set threshold. (Kumar et al. 2007) 15

Figure 4: Example of the standard Photoshop interface. 16

Figure 5: The paint layer application, designed in a similar style to painting and photo-editing applications. 17

Figure 6: Example of the clicking variation of the layer system. This iteration uses large buttons that are activated by clicking to scroll the application. 18

Figure 7: Example of the hover variation of the clicking system. 19

Figure 8: The code used to achieve the scrolling effect in the hover iteration. 19

Figure 9: The “Gaze Trace” orb that the Tobii software provides for navigation assistance. 22

Figure 10: An example of the standard setup used for the project. 22

Figure 11: Diagram demonstrating the times gathered in task one. 25

Figure 12: Diagram demonstrating the times gathered in task two. 26

Figure 13: Diagram demonstrating the times gathered in task three. 26

Figure 14: Diagram demonstrating the times gathered in task four. 27

Figure 15: Diagram demonstrating the times gathered in task five. 27

Figure 16: Diagram demonstrating the times gathered for task six. 28

Figure 17: Data obtained from the click survey demonstrating the usability feedback. 29

Figure 18: Data obtained from the click survey demonstrating the feedback on the strains experienced when using the system. 29

Figure 19: Data obtained from the hover survey demonstrating the feedback received on the usability of the iteration. 30

Figure 20: Data obtained from the hover survey demonstrating the feedback received on the discomforts experienced when using the iteration. 30

Figure 21: Diagram demonstrating the mean SUS score ratings corresponding to seven adjective ratings. (Bangor et al. 2009) 31

Figure 22: Diagram demonstrating the comparison of adjective ratings, acceptability scores and grading scores relative to SUS. (Bangor et al. 2009) 31

Figure 23. Diagram showing the changes in average times between both tasks. 40

Figure 24. Sample of qualitative data gathered from the clicking survey. 40

Figure 25. Sample of qualitative data gathered from the clicking survey. 40

Figure 26. Sample of qualitative data obtained from clicking survey asking about any problems experienced. 41

Figure 27. Qualitative data from clicking survey asking participants what recommendations they could offer to the project. 41

Figure 28. Sample of clicking qualitative data asking participants to provide detail on any strains experienced. 41

Figure 29. The SUS scale survey used as part of this project (Bangor et al., 2009) 42

Figure 30. Sample of qualitative data gathered from the hover survey. 42

Figure 31. Sample of qualitative data gathered from the hover survey. 42

Figure 33. Sample of hover qualitative data asking participants to provide detail on any strains experienced. 43

Figure 34. Sample of qualitative data obtained from hover survey asking about any problems experienced. 43

Figure 35. Qualitative data from hover survey asking participants what recommendations they could offer to the project. 43

1.0 - Introduction

This section will introduce the subject of eye-tracking control and the current methods being used for navigation. This section will then go on to give a brief background of the current eye-tracking technology available to control computer aspects such as the mouse. This section will then introduce the aims and objectives of the project.

1.1 - Introduction

In the last decade, Eye tracking has become an increasingly popular method of computer navigation (Pavani et al., 2019; Shakir et al., 2019; Zhou et al., 2019), with a heavy focus on using eye-tracking as a method of controlling the mouse. Eye-tracking is also readily available to consumers, with various companies providing products such as “Tobii” and “Gaze Point”, allowing for the purchase of infra-red based eye trackers. A major aspect of gaze-based cursor control is scrolling methods (Kumar et al, 2007). This project focuses on the creation of scrolling techniques, utilizing and testing two separate approaches.

1.2 - Background

Eye-tracking systems have a large amount of potential with accessibility in the field of computer control and navigation. Studies such as “Application of eye tracking…” (Liang et al., 2019) and “GazeButton” (Rivu et al., 2019) demonstrate the desire to develop new, inventive methods of controlling and adapting user interfaces required to navigate applications via the use of conventional eye trackers. Demonstrated within these projects are the major issues faced when designing for eye tracker cursor control. These main issues highlighted are “UI simplicity” and gaze area. To create UI’s that are compatible with eye-tracking control methods, a large “selection space” for buttons is required. A minimalist design is also imperative, to ensure that users select the correct input. These principals, in turn, can be applied with the control of scrolling functionality. Farokhian (2019) investigated scrolling in human-computer interaction while using eye-tracking and found that during its testing of scrolling functionality with eye trackers, the scrolling was described as problematic, lacking compatibility with eye-tracking approaches and being a major cause of control frustrations. This forms the basis of the project: designing and testing out scrolling methods designed for eye trackers.

1.3 - Aims

The main aim of this project is to design and test various methods of scrolling via the use of eye trackers. These scrolling techniques will be applied to a paint application for the context of the application. Using an Agile development methodology, two scrolling techniques will be prototyped and applied to a paint layer system. By utilising an A/B testing methodology, the two iterations of the painting application will be tested with a balanced weight to compare and contrast the usability.

1.4 - Objectives

The objectives of the project are as follows:
Objective one is to research and design various methods of scrolling. Two methods of scrolling will be created.
Objective two is to create a simplistic paint application that can utilise and incorporate the developed methods of scrolling into a layering system. This layer system must include the basic functionality of paint applications.
Objective three is to design a series of tests using the A/B testing methodology to measure the usability of the system. These tests will involve multiple measuring practices such as the System Usability Scale (SUS) and timed tasks.
Objective four is to review the testing data to compare and contrast the iterations to conclude which is more usable and evaluate the future potential of eye-tracking based scrolling.

1.5 - Section Summary

This section discusses the opportunities for new methods of control to be introduced within the field of eye-tracking. This section also highlights the availability to develop methods of control centred around scrolling. This section then highlights the aims and objectives of the project, showing the method of testing and measurement that will be used to measure the success of the scrolling systems.

2.0 - Literature Review

2.1 - Introduction

To understand the purpose of the research being conducted, previous research will be discussed. The main aspects of this literature review that will be analysed are: disability statistics, to understand the target audience of the research; eye-tracking technology and how it functions, to contextualise how eye trackers work; a review of current methods of eye-tracking computer navigation, to understand how the conducted research fits into the field of eye-tracking studies.

2.2 - Disability Research

One of the largest purposes of using eye-tracking as a method of navigation is so that those with various disabilities have the ability and opportunity to interact with technology. The "Annual Compendium of Disability Statistics" (Houtenville et al., 2016) documents that approximately 24.2% of the United States of America population is affected with some form of a cognitive disability, as shown in **figure 1**. With the large potential range of disabilities, many of which could affect the user's ability to navigate a computer, a unified method of computer control would be highly beneficial. It should be noted however that Houtenville et al., (2016)'s research only covers disability statistics from America and from 2016, which only covers a section of the audience that may utilise eye-tracking. Eye-tracking can serve as a method of control, (Su et al., 2006) due to it only requiring eye movement and nothing else, which provides users with conditions such as "cerebral palsy" an alternative and more comfortable method for computer navigation.

Table 2.5 Employment—Civilians with Cognitive Disabilities Ages 18 to 64 Years Living in the Community for the United States and States: 2014

State	Total	Employed	
		Count	% [1]
U.S.	8,669,210	2,097,518	24.2
AL	176,341	29,134	16.5
AK	15,987	4,102	25.7
AZ	174,720	39,458	22.6
AR	104,470	20,495	19.6
CA	864,659	188,992	21.9
CO	122,940	36,199	29.4
CT	87,395	27,611	31.6
DE	25,881	6,341	24.5
DC	19,101	3,650	19.1
FL	504,843	101,020	20.0
GA	271,699	51,423	18.9
HI	30,340	9,265	30.5
ID	44,854	12,060	26.9
IL	262,919	67,024	25.5
IN	193,158	47,338	24.5
IA	73,926	26,391	35.7
KS	83,801	25,321	30.2

State	Total	Employed	
		Count	% [1]
KY	186,671	37,421	20.0
LA	158,671	36,387	22.9
ME	58,627	13,206	22.5
MD	129,300	37,622	29.1
MA	193,931	51,295	26.5
MI	349,485	76,018	21.8
MN	137,876	49,616	36.0
MS	110,466	21,871	19.8
MO	213,461	51,736	24.2
MT	28,447	8,134	28.6
NE	42,798	15,114	35.3
NV	73,830	22,941	31.1
NH	35,335	9,950	28.2
NJ	165,563	44,025	26.6
NM	72,025	13,560	18.8
NY	441,131	104,916	23.8
NC	296,333	64,130	21.6
ND	14,179	6,047	42.6

State	Total	Employed	
		Count	% [1]
OH	380,948	97,910	25.7
OK	124,275	31,582	25.4
OR	145,487	39,798	27.4
PA	395,976	96,423	24.4
RI	38,786	10,730	27.7
SC	156,050	32,446	20.8
SD	22,803	9,500	41.7
TN	244,136	48,017	19.7
TX	635,726	166,750	26.2
UT	61,546	20,373	33.1
VT	23,539	6,113	26.0
VA	200,478	54,162	27.0
WA	219,260	56,779	25.9
WV	84,854	13,279	15.6
WI	155,487	48,912	31.5
WY	14,696	4,931	33.6
PR	180,624	25,180	13.9

Figure 1. Table showing the population percentage of adult Americans who suffer from a form of cognitive disability. (Houtenville et al., 2016)

With the large range of disabilities that exist, eye-tracking can be applied to a large variety of these to serve as control systems. Adjoudadi et al. (2004) address this with, looking at systems that utilise eye-tracking to assist as a control method for an interface. This interface is utilised by participants who suffer from severe motor neurone disease. In this study, the benefits of these methods of control are highlighted, stating that minimal movement except for the eyes that are required to operate the system. The study also expresses the desire of creating a usable experience for users of these eye-tracking studies, with the study focusing on methods to stop and counter jittering effects when using eye-tracking. These jittering effects will need to be accounted for when designing how precise the actions the participants can perform.

2.3 - Eye Tracking Technology - Functionality and Accessibility

Commercial eye trackers function via an “infra-red” corneal reflection method (Poole et al., 2006). This works by reflecting infra-red light into the corneal of the user, which is then reflected towards the tracker. The vector of reflection is calculated by the tracker to generate a “point of regard”. Using this data, it is possible to map a mouse cursor onto the location which the user is looking, meaning that they can control mouse movements with their eyes. **Figure 2** gives a brief diagram of how infra-red functionality operates with the eyes.

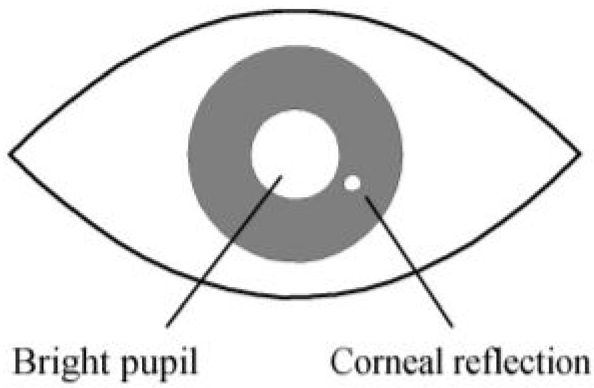


Figure 2. Diagram of an eye and how the reflection is obtained. (Poole et al., 2006)

Utilising eye-tracking as a control method to assist with controlling applications has been incorporated into various studies, as the practice has remained consistently popular. Corno et al., (2002) cover this issue, using a webcam to function as an eye-tracker to control what the user selected. Corno et al., (2002) expresses the desire to use eye-tracking technology that can easily be obtained, stating that precise equipment provides better functionality but at a high price. With the modern developments made in the eye-tracking industry, high-quality eye-trackers are now easily obtainable. When developing the scrolling systems, it’s imperative to use an eye-tracker with a fast response rate that is also affordable.

Utilising eye-tracking technology for mouse cursor control does bring with it challenges of accessibility. One of the biggest issues that must be considered when creating eye-tracking technology is eye-strain. Bruneau(2002) highlights these problems, noting that eye strain can be caused during eye-tracking tasks that involve a large workload. Backtracking must also be taken into account when controlling applications via eye-tracking, as large, constant movements should be avoided to prevent eye strain.

Another issue that must be considered when designing for eye tracking is the presentation of information. If users are unable to understand the information that they are shown immediately, it will result in the information being ignored (Manhartsberger et al., 2005). This leads on to the issues of "lack of affordance", meaning users will miss vital cues unless it sticks out visually.

One other aspect to be considered is the "Midas touch" problem (Jacob. 1995; Velichkovsky et al. 1997; Vrzakova et al., 2013). This problem is caused by the eye tracker being unable to distinguish between general actions of observation and actual inputs. This leads to incorrect or unintentional inputs leading to a frustrating experience. However, with some design considerations, these problems can be avoided. The most successful solution to the issue is to implement a delay in eye-tracking inputs. Recommended delay of input is approximately 450-500 ms to prevent accidental input and allow the user enough time to register the effect of their actions. The delay can be reduced depending on the complexity of the actions being performed. One other solution to the issue is to implement selection functionality to a different input device such as an "action button" so users have a larger degree of control over what is selected.

From the information discussed, it's imperative to understand that designing applications for eye trackers and general mouse usage, while fundamentally the same, require a degree of simplistic design and consideration to ensure intended usage and comfort when using the application.

2.4 - Using Eye Tracking for Application Navigation - Related Projects

This section will focus on analysing and researching studies similar to the research topic at hand, to understand what design and development considerations must be taken when designing the scroll system.

A study conducted by Sibert et al. (2000) focused on the creation of two different interaction techniques, comparing and contrasting using an A/B style testing format. This study does highlight several potential issues that can be faced with eye-tracking. The study highlights that users eye would suffer from "fatigue" if they were operating and performing tasks with the eye tracker for too long. Considering this, it's imperative to design testing scenarios to include breaks and ensure that the user is comfortable throughout testing. The study also notes the impact of the quality of the eye tracker. Due to the age of the study, affordable and effective eye-tracking equipment was generally unavailable at the time, generating an "obstacle" for this type of study. Taking advantage of fast, affordable eye-tracking equipment is mandatory to produce effective and reliable results.

“GazeButton: Enhancing Buttons with Eye Gaze Interactions” (Rivu et al., 2019) demonstrates aspects of UI design that must be considered when designing buttons for eye-tracking interaction. The study involves designing buttons that serve as “short-cuts” for various functions. When controlling how buttons are interacted with, the study suggests multiple methods. The first approach involves a mix of using gaze to control selection and touch screen to confirm the choices that the user has selected. The second approach utilises a “dwell” approach, where interactions are confirmed if the user has maintained eye contact with their selection for a short time. The authors of the study concluded that combining gaze and other forms of input can be highly effective, creating an effective solution to the “Midas touch” problem that affects eye-tracking research.

“PeyeDF: an Eye-Tracking Application for Reading and Self-Indexing Research” (Filetti et al ., 2019) demonstrates an application that utilises only eye tracking as a navigation tool for viewing PDFs. To achieve the function of scrolling the PDF, the application detects if the user’s gaze has looked below a certain threshold. Once the threshold has been passed, the page will scroll from one block of text to the next. This scrolling effect uses no clicking or input actions to initiate the scrolling action. This functionality of scrolling serves as an inspiration for the hover style of scrolling that was implemented within the hover iteration.

“Gaze and Mouse Coordination in Everyday Work” (Liebling et al., 2014) provides further evidence for the ease of access of modern eye-tracking equipment and its recently acquired ability to be implemented into modern computer systems with no specialised hardware requirements. This study also provides evidence for the idea of combining both gaze and click interaction to combat the Midas touch problem. Previous research highlighted within the study found that using just gaze and dwell for all interactions caused significant problems and incorrect inputs. Furthermore, the study also demonstrates the potential inaccuracies that can be experienced when operating an eye tracker for precise tasks. This suggests that objects designed for gaze interaction must be designed around these inaccurate inputs.

To further expand upon the Midas touch problem, Majaranta et al. (2002) provide more detail on some of the issues caused when attempting to navigate applications via eye-tracking. When discussing the issues faced when designing typing systems that are controlled by eye-tracking, the problem of involuntary eye movement is addressed. Majaranta et al. (2002) states that eye movement is split into conscious and unconscious movement, meaning that a user may be focusing on an object on screen but their attention could be focused elsewhere. This then causes a disconnect in the actions of the user which may then lead to involuntary inputs and other problems. The Midas touch problem will need to be heavily considered when designing the scrolling system, as the system must be designed so that involuntary or accidental movements or inputs don’t dramatically alter the application or confuse the participant.

2.5 - Conclusion

To conclude, the research presented shows that while there is lots of potential and reason for the development of eye-tracking application solutions, there are several pitfalls that can be fallen into which can negatively alter the experience for the user. It's important to consider that the developed scroll system must be simple and intuitive to understand. It must also include some form of controlling how and when input should be registered and how much impact the input should have on the system.

3.0 - Interface Research and Design

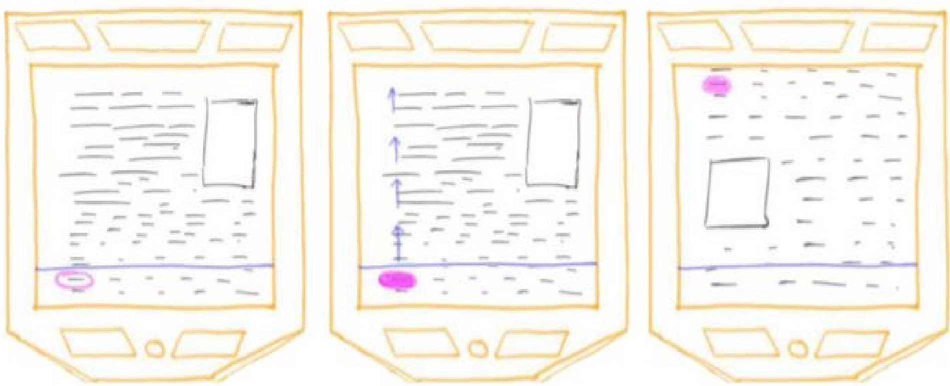
3.1 - Introduction

This section will detail the design ideas and process behind the creation of the scroll system. It will include the research question being addressed and the reasoning behind it. This section will also demonstrate the design of the two scrolling iterations being used.

3.2 - Research Method

To formulate the question, “Experimental Human Computer Interaction…” (Purchase. 2012) was used as a reference guide. The research question that will be addressed is: ‘Is a clicking or a hover approach more effective for navigating lists via Eye Tracking?’. To test this idea, two different scrolling methodologies will be created, using ideas formed from the related work research.

Variation one will utilise a conventional clicking approach, using large buttons that will scroll the list set distances depending on the button pressed. The second condition for the test will be a smooth-scrolling hover approach. This scrolling method, similar to the “Discrete Scrolling” method (Kumar et al., 2007) uses targets that are positioned at the top and the bottom of the page. When the mouse hovers over these targets, the page will begin to scroll in the direction of the target. **Figure 3** demonstrates an example of the process in practice when being used to navigate a web page.



A. The user’s gaze position when the eye gaze drops below the scrolling threshold.

B. If the user’s gaze stay below the threshold for the duration of a micro-dwell (~150-200ms) the system issues a Page Down command, which results in the GazeMarker being drawn.

C. The motion of the GazeMarker directs the user’s gaze up to the top of the page keeping it positioned where the user was reading. The GazeMarker slowly fades away over a couple of seconds.

Figure 3. A sketched diagram of the process of the “Discrete Scrolling” method. The scrolling will trigger once the user’s eye is below a set threshold (Kumar et al., 2007).

The independent variables that are being taken into consideration for this study are the methods of vertical scrolling, the speed of the scroll and the input required to activate the scroll. The population that will be participating in testing for the scrolling methods will vary in age between eighteen and seventy. All the participants will have varying levels of eyesight ability and computer literacy. This choice of population is due to the generalisability of the scrolling methods. The intended use of the scrolling method is to apply it to various applications that use small sorting lists to group and control content. With this, using adults as the target audience for the project reflects the demographics of people who would use the system under the context of a working environment. Participants must be able to use the eye-tracker provided to participate. If the participants suffer from any eye conditions that affect the eye-trackers abilities, the participant will not be able to perform the testing tasks. To contextualise this idea even further, the scrolling system designs will be applied under the context of a paint layer system.

3.3 - Interface Design

3.3.1 - Interface Design Introduction

This section will highlight the two scrolling interfaces that have been designed. It will detail the main differences and the rationale behind the design choices of the systems. This layer system will be contained within a simplified painting application. The developed paint system was modelled after the application “Photoshop”. This was due to the high popularity and common design of “Photoshop”. **Figure 4** provides an example of a typical interface found within the application.

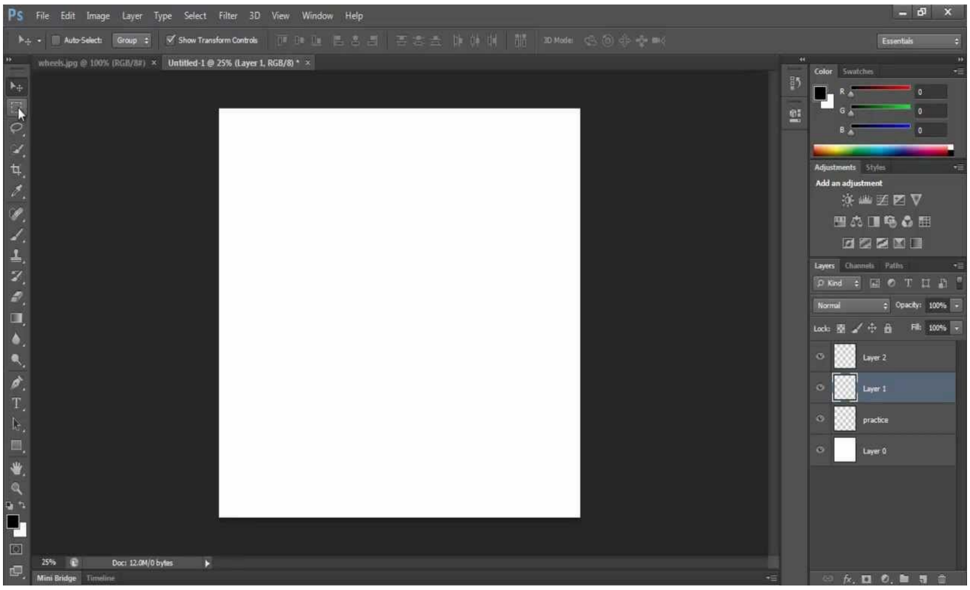


Figure 4. Example of the standard Photoshop interface.

However, the design must be considerably different due to the fact that an eye tracker will be used for navigation, as opposed to a conventional keyboard and mouse. This eye-tracking adapted, simplified design is shown in **figure 5**, where the paint application consists of a bordered page, containing a canvas and some simple paint tools. The application also includes a start button and covered section used for testing the layer system. When the start button is clicked, the grey box that is covering where the scrolling system is located on the page will disappear. The layer system that is currently selected for the paint application will then be revealed. This is so that participants will not be able to look at the layer system and scrolling method and attempt to figure out how the system functions before testing begins. When the start button is clicked, a timer is also started in the background of the application. This timer will count the number of seconds it takes for the participant to complete the task that they have been set. Once the participant completes the task, the space bar on the keyboard is programmed to stop and output the time the participant took to complete the task. This stops the functionality of the application when used to prevent the application from being used further until the testing conditions are satisfactory again. Testing will be discussed in greater detail in section four.

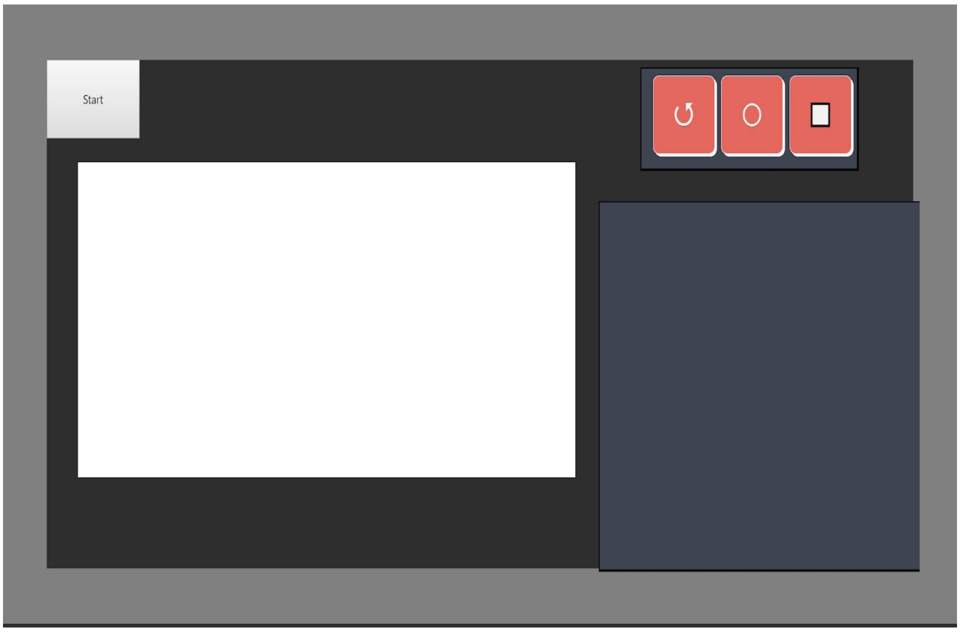


Figure 5. The paint layer application, designed in a similar style to painting and photo-editing applications.

3.3.2 - Clicking Iteration

The first iteration that will be discussed is the click variation of the paint system. **Figure 6** shows the clicking design. The layer system uses a simple list that is split into three tabs. A tab for selecting multiple layers, the layer number and the name. There are sixty layers listed to recreate a paint layer system being used mid-project. In this iteration, the scrolling is performed by clicking the large buttons to the right of the list. Each of the buttons is “100x100” in pixel size. The amount of arrows on the buttons indicates their functionality. The buttons that only contain one arrow move the list either up or down by three layers. The buttons that contain two arrows move the list either up or down by twenty-five layers. These buttons are activated by clicking them with the mouse. There is also an add and remove the layer button shown above the list of layers. These were included to provide further detail and recreate a standard layer system that would be found in most conventional painting applications, like “Photoshop”.

Figure 6. Example of the clicking variation of the layer system. This iteration uses large buttons that are activated by clicking to scroll the application.

3.3.3 - Hover Iteration

The second variation that will be discussed is the hover variation of the paint layer system. **Figure 7** shows the hover design. The hover variant functions similarly to the previous variant discussed, using the same list design and approach. However, the most notable and important difference is the method of scrolling. Instead, the hover variation uses two large, transparent bars at the top and the bottom of the layer list. Each of the bars is “360x70” pixels. These bars, when interacted with, smoothly scroll the list up or down depending on which bar is interacted with. The hover scroll achieves this by repeating a scroll function, scrolling the list by fifty pixels every time the function is called. Once the mouse cursor is removed from the scroll box, the interval function is reset to register the next scrolling command. The code used to achieve this scrolling functionality is demonstrated in **figure 8**. The list slowly scrolls to prevent users from losing their place in case they accidentally interact with the hover bars. Both designs also use a drag and drop system for rearranging layers. This is controlled by clicking and holding down on the desired layer and dragging and dropping the layer to the intended position.

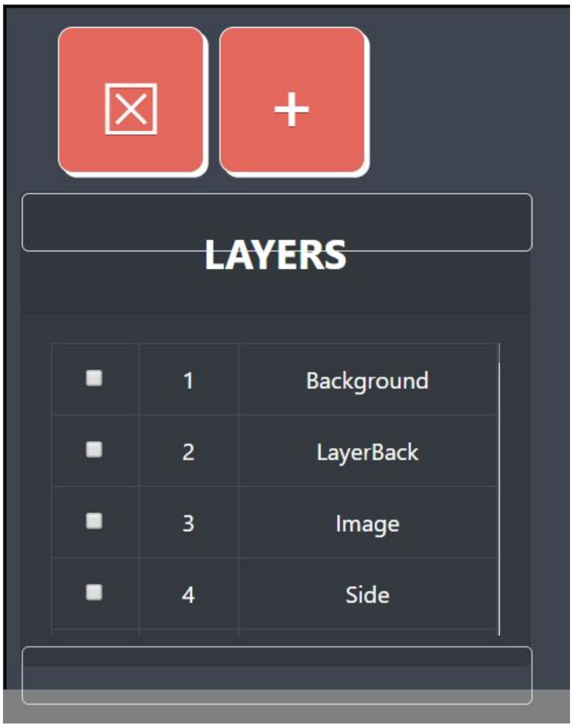


Figure 7. Example of the hover variation of the clicking system. This design uses white outlined boxes to serve as the method of input for scrolling, which is activated upon being hovered over by the mouse.

```
function hoverOneUp(){
    var elmnt = document.getElementById("content");
    elmnt.scrollTop -= 50;
}

function hoverOneDown(){
    var elmnt = document.getElementById("content");
    elmnt.scrollTop += 50;
}

function scrollOneUpHover(){
    s = setInterval(hoverOneUp, 70);
}

function scrollTenUpHover(){
    s = setInterval(hoverTenUp, 70);
}

function scrollOneDownHover(){
    s = setInterval(hoverOneDown, 70);
}

function scrollTenDownHover(){
    s = setInterval(hoverTenDown, 70);
}

function clearScroll(){
    clearInterval(s);

}
```

Figure 8. The code used to achieve the scrolling effect in the hover iteration. The code functions by finding the list and running the scrolling code at constant intervals. Once the mouse cursor has left the hover scrolling box, the clearScroll() function is called to stop the interval calling the scrolling function, stopping the scrolling.

3.4 - Section Summary

To summarise this section, two different scrolling methods have been designed and created to test out the potential of new scrolling methods. These methods are then integrated into a simplistic paint application to recreate the layout and functionality of a standard paint application. When implementing these scrolling methods, the type of eye tracker and A/B testing technology was considered when controlling and testing out these scrolling methods. These will be discussed in greater detail in the experiment methodology section.

4.0 - Experiment Methodology

4.1 - Introduction

This section will discuss the testing methodology of the project and the various resources that were used and implemented to achieve the desired method of control and data measurement. This will include the type of testing conducted, the tests designed for the project, the eye tracker used for controlling the system and the A/B testing technology used for gathering data.

4.2 - Method of Testing

As the purpose of the testing was to compare two different iterations of scrolling systems, the decided methodology for this testing was an A/B testing format (Speicher et al., 2014; Bruun et al, 2009). This format of testing functions creates an equal “weight” of participants for each of the developed versions to gain first impressions of both versions. However, due to a large number of participants required, a “within-participants” approach is adopted as well (Lazar et al., 2017; Purchase., 2012). This approach functions by having participants use all versions of the scrolling system. This study design controls for any participant differences that may affect performance in each iteration, which may skew the data. However, the order of which scrolling system they use and the tasks they perform will be randomised to prevent bias caused by familiarity

To gather data, six different tasks were devised that would incorporate the scrolling system, requiring the participant to scroll various ranges. Tasks one to three would have the participants scroll short to long distances. Tasks four and five would then have participants utilise the layer drag and drop system, asking participants to drag layers short and medium distances. The final tasks would then combine everything that the participants had performed over the previous tests. The participants would be required to scroll down a far distance and then move a layer upwards. The participants would have to add this layer manually by clicking the “+” button located at the top of the layer list. Once these tasks were completed, the participants filled in a survey to evaluate their experience with the scrolling systems. Both versions would be tested in the same session with the participant. Participants would perform the testing by starting the test themselves when prompted to ensure that they were adequately prepared to perform the tasks. This was achieved by using the start button and list hiding function as demonstrated in figure five.

4.3 - Testing Technology and Resources

One of the most important pieces of technology with this study was the choice of the eye tracker. For this study, the “Tobii 4C” eye tracker was chosen for control over the location of the mouse. This is due to its commercial availability, along with various other features, such as a comfortable operating distance of 95cm, to ensure that mouse tracking was quick and responsive. One major advantage which this Tobii eye-tracker offered was its inbuilt functionality that would assist with control and navigation. This technology, which is part of Tobii’s eye-tracking software, is the “Gaze Point” application. This software allowed for the mouse to be automatically connected to gaze data that was inputted from the eye-tracker. The only requirement for participants was a simple calibration exercise to check the functionality of the eye tracker and to ensure that the eye tracker was calibrated. Another major advantage which Tobii’s software offered was the “Gaze Trace” function. When activated, a semi-transparent orb will follow your gaze to give the user an idea of where they are currently looking. An example of the “Gaze Trace” function being used is demonstrated in **figure 9**.

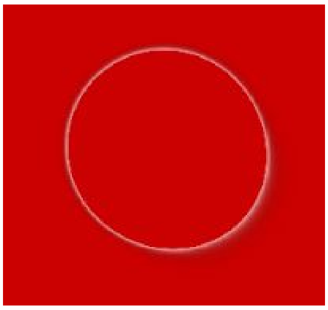


Figure 9. The “Gaze Trace” orb that the Tobii software provides for navigation assistance. When used in conjunction with “Gaze Point”, the mouse cursor will be located directly in the middle of the circle.

For the hardware used to run the eye tracker, an ASUS laptop supporting an “Intel Core i5-8300H CPU @ 2.30GHz” and “8GB” of Ram was used for the entirety of the project. This gave the eye tracker an optimal environment to run. Another hardware addition made to this setup was the inclusion of a mount for the eye tracker. Using this mount, the eye tracker could be set up in a consistent position to provide reliable results. An example of this set up in use is shown in **figure 10**.

Figure 10. An example of the standard setup used for the project. The eye tracker is secured by a magnetic mount provided with the eye tracker to ensure that it remains in the same position.

For clicking functionality, a standard mouse was used but had been altered so that only the left click was functional, so participants would not use the mouse to control the cursor location. As for the creation of the scrolling system, this was designed using a combination of web development techniques, coded in HTML, CSS and Javascript to design and control the different methods of scrolling and other functions included. To control the design of the scrolling systems and separate them for testing purposes, Google Optimise was implemented into the project. This was achieved by uploading a version of the developed web page onto an online server to provide a web link. The Google Optimise script was then included in the code of the paint system for it to be used within the Google Optimise development environment. Implementing Google Optimise into the project allowed for the design to be split into two versions, using Google's design tools to hide and delete the scrolling method that was not being used in the various testing versions. Once the changes were made, the changes would be applied to the web link, meaning that the web page would then run one of the two different scroll methods to control the page and ensure separation of the methods of scrolling.

4.4 - Method of Gathering Data

To evaluate the success of the different methods of scrolling and their usability, methods of measurement would need to be incorporated to evaluate. For the measurement of task performance, the main measurement system used was time taken to complete the task, serving as quantitative data. Participants were briefed on the task and asked to click a button to start the task. Once they completed the task, the participant would hit another button to declare they were finished, which would then output the time. Time to complete the task (Wang et al., 2019) was implemented due to the efficiency and reliability of the data it provides, reflecting real scenarios where users of a system would aim to complete a task as fast as possible.

The second form of measuring usability and the experience of participants in testing was recorded using qualitative feedback. After participants had finished all the tasks for the iteration they were testing, they would fill out a survey. In this survey, participants were asked questions asking them to place a numerical value between one and ten to evaluate different aspects of the iteration. These questions included general efficiency, enjoyment of using the system, strains experienced and any other problems encountered. Participants were then asked to justify the answers that they gave to the numerical questions. Qualitative data allows users to express precisely what they like and dislike about the system, creating feedback that can be more effectively utilised (Rose et al., 2005).

The third method of measuring the usability of the scrolling system was the System Usability Scale (SUS). The SUS scale operates by asking the participant ten questions about their experience with the system, asking questions ranging from comfort to understanding. The question answers are then used to produce a score. The score produced would then be calculated to produce an average score and then measured on the SUS grading scale(Brooke. 1996). The SUS scale was implemented into testing as part of the survey which participants had to fill out at the end of testing.

To obtain the data from the participants, a survey was designed using “Google Forms”. This service allowed for surveys to be designed which have the ability to store a variety of different data types. Utilising this service, a survey was designed that contained all of the methods of measurement required to measure the usability of the scrolling system. Two separate surveys were designed for each of the scrolling systems.

One other method of gathering data that was implemented into this experiment was observational data. While testing was being conducted, the researcher who was overseeing testing would note anything they witnessed that they believed would be of note. This data gathering data allows the researcher to note various interactions or experiences that the participant has performed without them realising and contextualise them into effective feedback and system improvement (Baber et al., 1996).

5.0 - Results

Twenty-eight participants were gathered for testing overall. However, due to problems with eye calibration with the eye tracker, only twenty-six of the participant's data could be used. These problems mainly spawned from the shape and condition of the eyes of the participants. Furthermore, some individual task data were excluded due to inconsistencies in the data that participants provided. These exclusion criteria will be discussed in greater detail in the Time Data section. This section will demonstrate the data obtained from each of the measurement methods performed in the testing process.

5.1 - Time Data

Time data was gathered from measuring the completion time of each of the conducted tasks. Shapiro-Wilk normality tests were run on all variables. The data gathered was non-normally distributed meaning that non-parametric tests were conducted as opposed to dependent samples t-tests. This test run was the Wilcoxon Signed-Rank Test. When measuring the times of the participants, if a participant were to complete a task with only one iteration or was unable to complete the task with both of the iterations, their time data was excluded from the test.

In task one, involving short-distance movement, greater task efficiency leaned towards the click iteration, with fifteen participants performing the task faster. However the click iteration revealed no significant difference between the iterations ($F = 199.0$, $p = 0.063$). It should be noted, however, that the *p* value is only 0.013 away from being significant. **Figure 11** demonstrates a box and whisker diagram providing further details to the gathered times for this task.

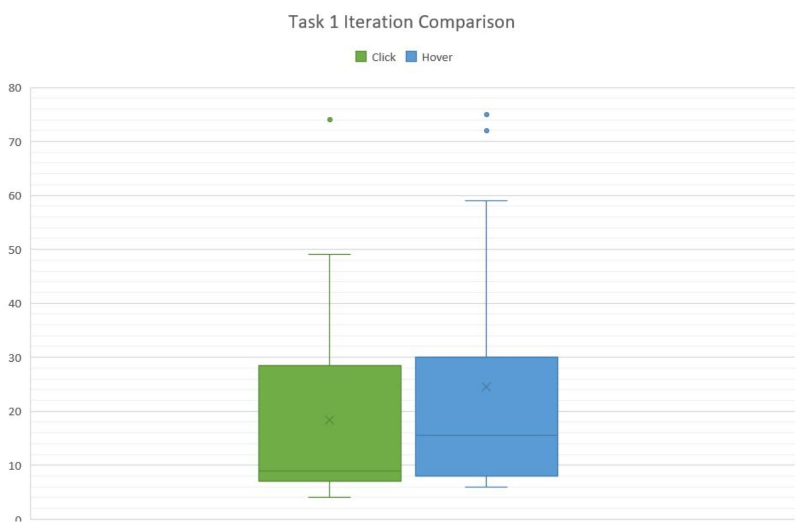


Figure 11. Diagram demonstrating the times gathered in task one. Click was faster with it's lower median than the hover iteration. This task had high means due to the number of outliers. The task was close to being significant, with both iterations providing similar interquartile range (IQR) data.

Task two involved moving moderate distances. Two participants were excluded from this test due to being unable to finish the task (N = 24). Fifteen participants performed faster using the click iteration. The Wilcoxon Signed-Rank Test revealed that there was once again no significant difference (F = 158.0, p = 0.819) between the iterations. **Figure 12** demonstrates the gathered data in greater detail.

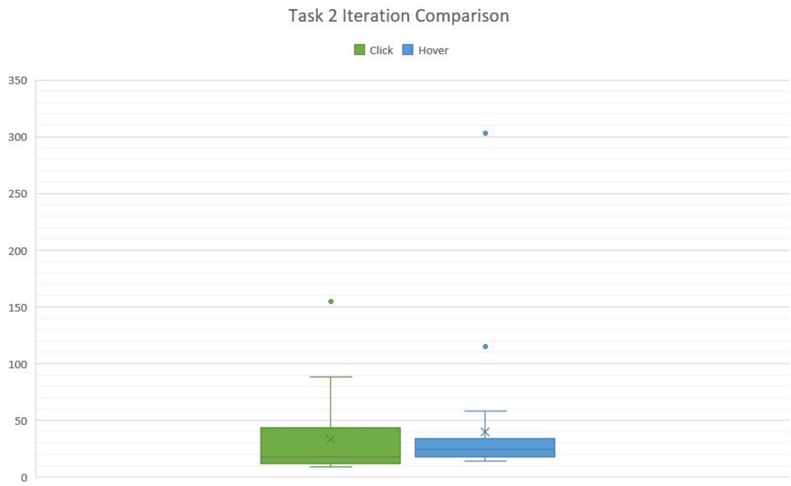


Figure 12. Diagram demonstrating the times gathered in task two. In this task, click was faster as shown by the lower median. However, the clicking task had a greater range of times. The IQR of the task for the clicking is also significantly larger than the hover iteration.

Task three involved long-distance movement. Two participants were excluded due to being unable to complete the task (N = 24). Task three showed that a large majority performed faster in this task using the click iteration, with nineteen participants performing faster. The Wilcoxon Signed-Rank Test also showed that there was a significant difference between the entries (F = 246.5, p = 0.006), where participants performed the task faster using the click iteration. **Figure 13** demonstrates the time gathered for this task.

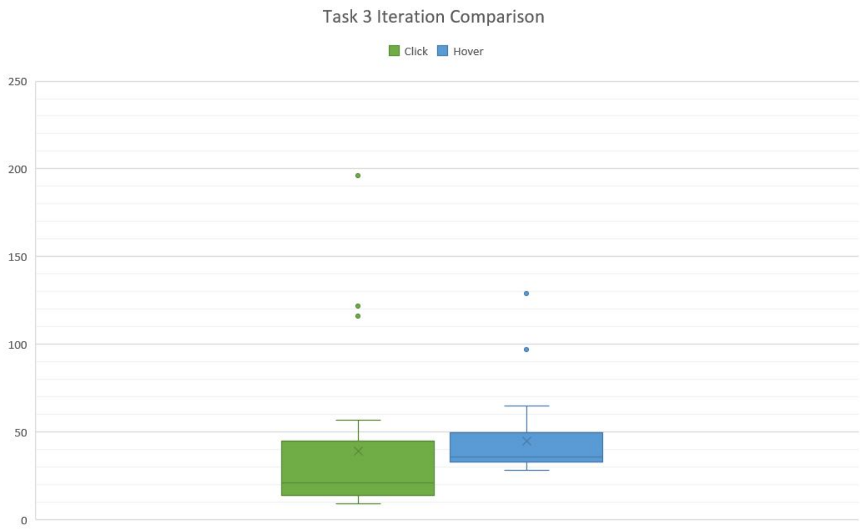


Figure 13. Diagram demonstrating the times gathered in task three. In this task, the clicking was evidenced to be significantly faster than the hover, with the median value for the click iteration being below the lowest value of the hover iteration. The smaller IQR on the hover iteration also shows that participants were consistently slower on this task.

Tasks four, five and six provided non-symmetrical data, therefore a Sign test was conducted instead. This involved measuring the median difference to establish differences in data distribution between the iterations. These entries do not use times that were tied however, dropping the total entries below twenty-five. Because of this, an exact sign test was used. Once again, if a participant was unable to complete the task for one of the iterations or was unable to complete the task for both iterations, then the data would be removed from the results of the task.

Task four involved scrolling and dragging layers short distances, and three participants were excluded (N = 23). For this task, hover was the fastest when completing the given task, with thirteen participants performing faster. Two data entries were tied, therefore they were excluded. However, the sign test revealed that there were no significant differences in distribution around the median values between iterations (Z = -0.876, p = 0.383). **Figure 14** demonstrates all the times gathered for task four.

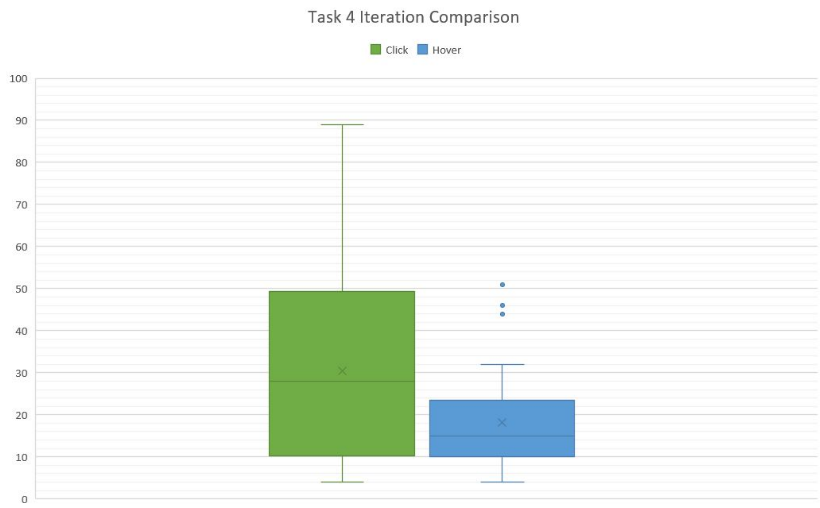


Figure 14. Diagram demonstrating the times gathered in task four. In this task, the hover iteration appeared to be faster, with a lower median than the click iteration. It can be noted as well that in this task, the click iteration has a significantly larger range than the hover iteration.

Task five involved scrolling and dragging layers moderate distance. Three participants were unable to complete the task, excluding their entries (N = 23). One entry was also tied meaning the number of entries used for calculating the median was twenty-two. The hover iteration proved faster in this task, with fourteen participants performing faster. Sign Testing showed that there were no significant differences between groups (Z = -1.066, p = 0.286). **Figure 15** demonstrates the times that were obtained during this task.

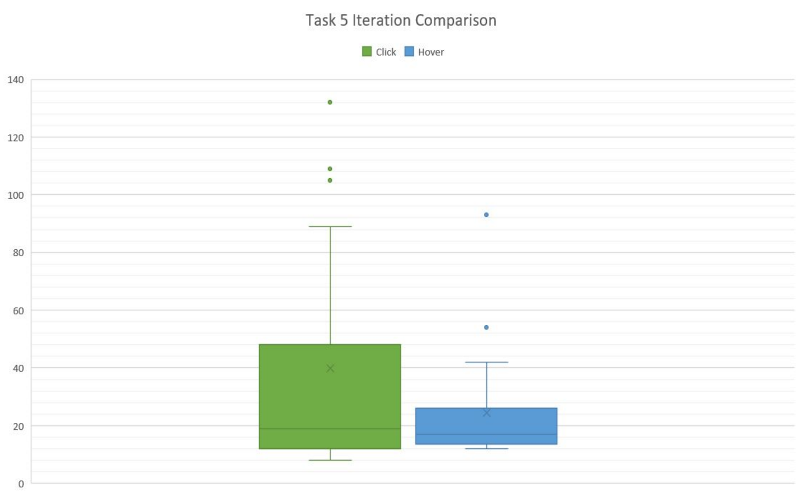


Figure 15. Diagram demonstrating the times gathered in task five. In this task, the hover iteration appeared slightly faster than the clicking iteration. Once again, the clicking iteration had a significantly larger range than the hover iteration.

Task six involved scrolling and dragging layers long distances. One participant was unable to finish the task (N = 25). One entry was also tied reducing the usable entries to twenty-four. Clicking was revealed to be the faster iteration in this task, with nineteen participants performing faster. A sign test showed a significant difference between the iterations ($Z = 2.654$, $p = 0.004$). **Figure 16** demonstrates the times gathered for the task.

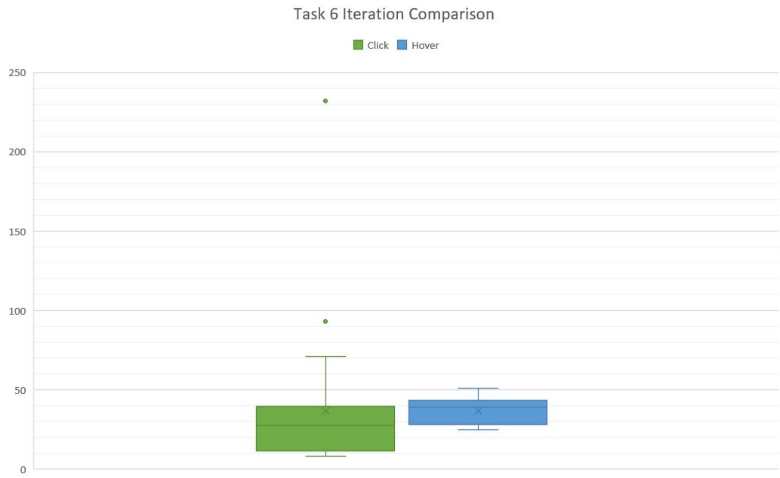


Figure 16. Diagram demonstrating the times gathered for task six. In this task, clicking was shown to be significantly faster, as shown by the lower median, and the lower range of the data Clicking once again also demonstrates a larger range of times.

5.2 - Observational Data

While participants were being tested, observations were noted about actions taken while they operated the system. One major observation noted during testing was the impact of jittering eyes on the control of the system. Participants that exhibited this jittery eye issue took longer to perform all of the tasks. Another problem that was observed was found during the clicking iteration tasks. Participants that would move the mouse while trying to operate the clicking buttons would accidentally drag the buttons. This would make the click not register and confuse the participant as to why their input was not registering. One major issue that was observed with one participant was the general lack of understanding of how the eye tracker functioned with cursor control. The participant would try to move their head to control the mouse. Because of this, a few participants were unable to complete a majority of the tasks. However, most of the participants proved to quickly understand the system after they had performed their first task with the iteration they were using but commented that using just their eyes to control where the mouse went to be an unconventional experience.

5.3 - User Qualitative Feedback

Participants were asked a series of questions to evaluate their experience with the system. Over 65.3% of the participants for the click iteration rated a six or above. Reasons for these answers range from its simplicity, intuitiveness and smooth scrolling. The clicking drag issue addressed in the observations was also addressed in the feedback, along with issues with the peripheral vision. 15.4% of participants also reported experiencing eye strain when using the system. Participants recommended that to improve the clicking iteration, better accuracy and adjustable scrolling speed would need to be implemented. **Figure 17** and **figure 18** demonstrate the feedback on the usability and the comfort of the experience.

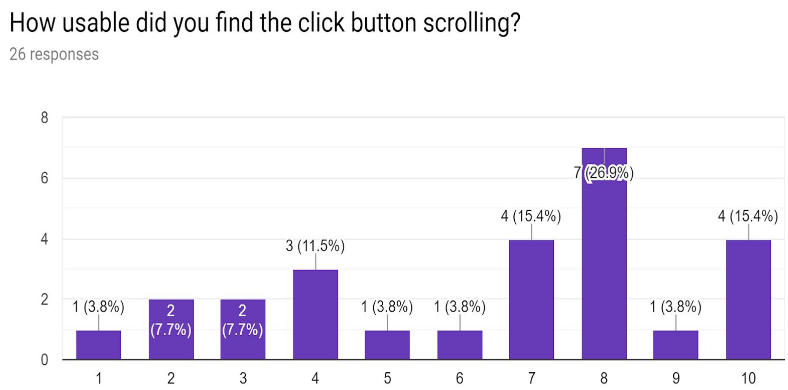


Figure 17. Data obtained from the click survey demonstrating the usability feedback. Usability feedback is generally mixed in this iteration.

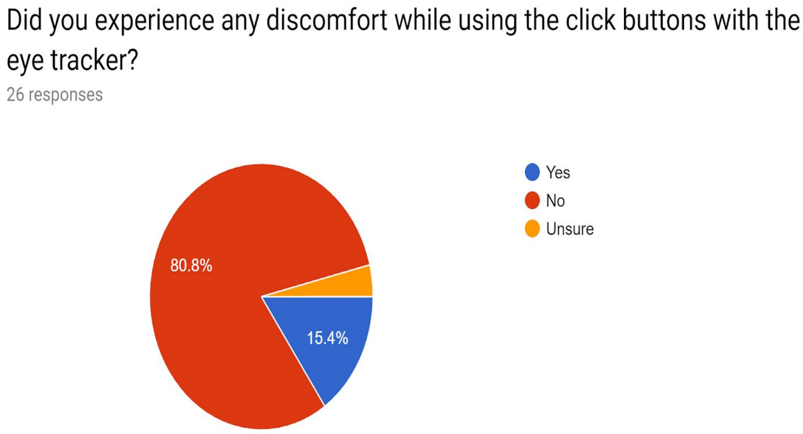


Figure 18. Data obtained from the click survey demonstrating the feedback on the strains experienced when using the system.

For the hover iteration, 84.6% of participants rated six or above for how usable they found the system. Reasons for these answers included the system being “easy to use”, “better peripheral vision” and less “effort” required to operate the system. Participants stated they found the system “accurate” and “easy to understand” but found they were sometimes unable to see the list properly. When asked about discomfort with the system, 26.9% of participants experienced some form of strain, with some participants quoting that the strain occurred when scrolling up. Participants recommended that for improving this iteration, more adjustable speed options would have to be incorporated along with better targeting. **Figure 19** and **figure 20** demonstrate the feedback gathered from the survey, showing positive feedback on usability and strain feedback.

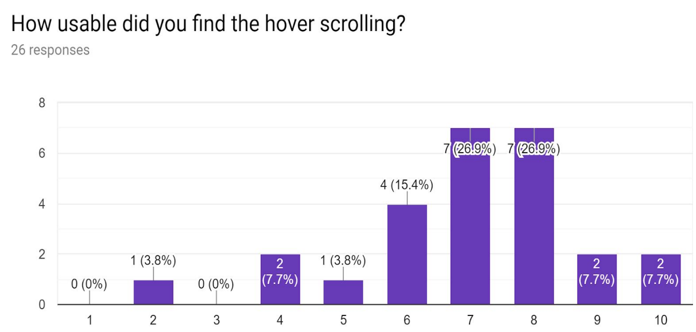


Figure 19. Data obtained from the hover survey demonstrating the feedback received on the usability of the iteration. As can be seen in the diagram, a large majority of the participants rated the usability highly.

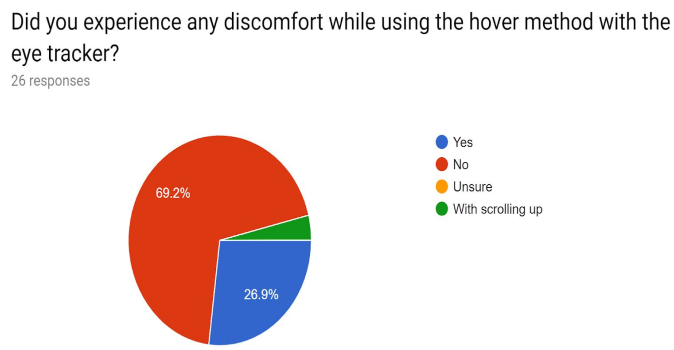


Figure 20. Data obtained from the hover survey demonstrating the feedback received on the discomforts experienced when using the iteration.

5.4 - System Usability Scale (SUS)

A SUS score was obtained from every participant for each iteration, producing a score between zero and one hundred. These scores would then be used to produce an average SUS score for each of the iterations. The SUS score consists of ten questions, each ranging from questions that ask if they experienced difficulty, required assistance and if they could understand and use the system by themselves. The scores that were obtained from the SUS scale could then be graded to determine the usability (Bangor et al., 2009). The click iteration provided a score of “70”, equating to a grade C on the SUS scale. The hover iteration provided a score of “74.5”, equating to a grade C on the SUS scale. **Figure 21** and **figure 22** demonstrate some of the methods of adjectivally classifying the SUS scores that have been generated.

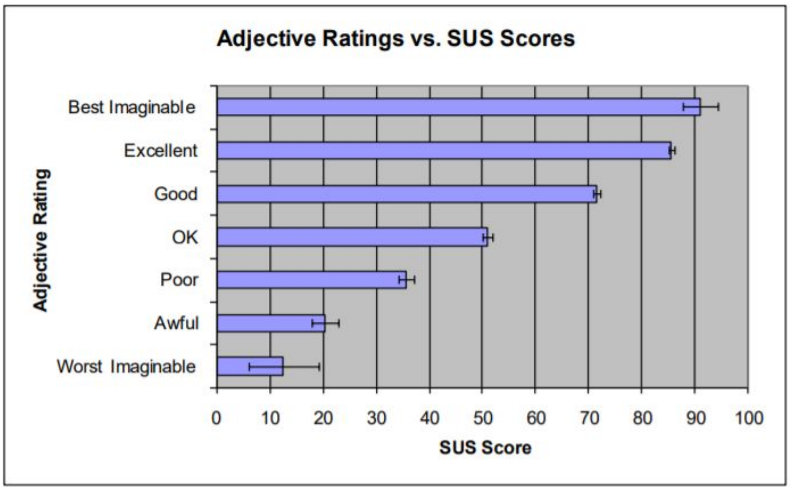


Figure 21. Diagram demonstrating the mean SUS score ratings corresponding to seven adjective ratings. (Bangor et al., 2009)

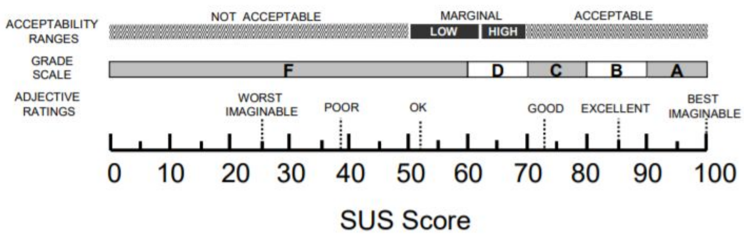


Figure 22. Diagram demonstrating the comparison of adjective ratings, acceptability scores and grading scores relative to SUS. (Bangor et al., 2009)

6.0 - Discussion

This study’s objective was to observe the usability of the two developed scrolling methods, compare and contrast to evaluate which may be the better approach. Reviewing the times gathered, it was shown that clicking was faster on scrolling tasks, with clicking being performed faster on the first three tasks and task six. The hover was performed faster by more participants on task four and five. The Wilcoxon Signed-Rank Test revealed though that only task three and six provided a significant difference. The remaining tasks proved to show no significant difference. The only other task that was close to showing a significant difference was task one, showing a significance level of “0.063”. From this it could be assumed that only a slightly greater population size would be required to reach significance; furthermore, the trend of faster times in the clicking iteration for tasks one to three, which require similar responses from the participant, makes this a more confident assumption. From this gathered time data, it can be agreed upon that the clicking iteration of the scrolling system proved to be the faster version of the scrolling system. However, one thing of note when reflecting upon the gathered data was the large range of times found in the click data. Goldberg et al. (2002) also suffered from the problem in their eye-tracking study, observing a significantly large variance in the times taken to perform the various tasks, with some tasks taking as long as five minutes. In tasks four and onwards, the range of data entries was significantly larger than the hover data entries. This time data suggests the idea that while clicking was the faster iteration, the times of completion were inconsistent. This is in comparison to the hover iteration, which while slower than the clicking iteration, had a significantly smaller range of times. This suggests the theory that the clicking iteration may have had some degree of chance when being used, with such a large divide of participants. With the number of tasks that provided data that was not significant, a greater number of participants would need to be included to provide significantly different results or more consistent data.

While the gathered time data leans towards the clicking iteration being the better iteration, the observation data suggests that the hover iteration was easier to use and understand. Several participants reported having issues with the clicking iteration, stating that their inability to use their peripheral vision hindered their experience. Another aspect which affected the quality of the clicking control was jittering eyes. When the participants moved their eyes and clicked at the same time, the action would not register, adding to task time and producing overall slower times. This may explain the larger variances in time data in the click iteration, as the issue of jittery eyes may have considerably slowed those who suffered from it. This issue could potentially be solved by developing a more robust clicking system that can compensate for this jittering eye problem.

The qualitative data gathered showed that participants preferred the hover iteration. When comparing how many participants found the system usable, 19.3% of participants rated six or more on the hover iteration compared to the clicking iteration. When asked to give a reason for the answers provided, participants stated that they preferred the hover iteration due to it being easier to understand and being able to see where they were in the list while also scrolling at the same time. While qualitative data may initially lean towards the hover iteration being the more favoured iteration out of the two, eye strain proved to be a problem. The Hover iteration caused 11.5% more participants to experience eye strain than the click iteration. When asking what produced the straining effect, participants stated that the constant motion of looking up was the main cause of the strain. Reflecting on the qualitative data gathered, while the hover iteration may have provided the more enjoyable and easier to use experience, the system causing strain to 26.9% of the participants involved cannot be ignored.

Analysing the SUS data generated from the two iterations also demonstrates interesting results. The hover iteration scored better than the click iteration, but only by 4.5 points - a marginal difference. Both versions obtained a score above fifty, making them both within an acceptable threshold of usability, both placing within the 'C' grade of usability. This means that while the hover iteration is slightly more usable, the difference is not considerable enough to state that one version of the scrolling system could be preferred over the other.

Upon comparing and contrasting the gathered results on both of the scrolling iterations developed, it can be concluded that none of the iterations performed significantly better than their counterpart. While it can be argued that the click iteration is the better iteration due to faster performance, it can also be argued that the hover iteration is the better iteration due to more enjoyable and understandable user experience. One potential solution to this problem caused is to perform more extensive testing on both the clicking and hover iterations of the scrolling system. With only twenty-six participants and six tests, only two of the tests provided data that was significantly different between conditions. If more participants took part in the testing, more data could be provided, and potentially more significance could be seen, especially in the tasks where the p-value was close to significance. Use of statistical software, such as 'G-Power', could provide greater insight into how many participants would be required to reach significance in the tasks, to a high power level.

When comparing the two different iterations, however, it is important to note that there are some limitations in both of their designs that impacted the reliability of the data that was provided. This main limitation with these iterations spawned from inconsistency and oversight of the design of the scrolling systems. When comparing the scrolling speeds of the iteration, the clicking iteration was given an unfair advantage compared to the hover iteration of the scrolling system. The clicking iteration was provided with two options for controlling the speed of the scrolling, compared to the hover iteration which was only given one option for the scrolling speed. This advantage may have lead to significantly faster times on the scrolling only tasks, which is demonstrated in the results. Kumar et al. (2007) experienced this problem when conducting testing, as participants experienced difficulty finding a universally agreed speed that the gaze-scrolling method should move the page. This presents an outstanding problem with the design of gaze-scrolling methods for research in this topic: more needs to be done to establish the best scrolling speed to suit all users. This may be solved by introducing two scrolling speeds, with faster or slower scrolling depending on how high or low the user looks within the hover bars. The hover iteration also had a distinct advantage over the clicking iteration as well, in reference to the moving layer system.

To move the layers in the system, a drag and drop system was implemented, which would function by holding down click on the layer that was desired to be moved and then dragging and dropping the layer to the intended position. This design was fairly intuitive within the hover iteration, as the list was already scrolled by the user looking up and down. But with the clicking iteration, this layer would be moved via this style as well, severely clashing with the design and functionality of clicking buttons to move the list. This flaw was demonstrated within the results of tasks 4, 5 and 6. These tasks had a significantly larger range of values compared to the hovering iteration, which produced a consistent range of results. Although this can be disputed, as task 6 produced results that showed that the clicking iteration was significantly faster than the hover iteration in a task involving moving layers. This could be contributed to the faster scrolling options that the clicking iteration offered compared to the hover iteration. These design flaws are aspects which will need to be heavily considered when developing and designing further iterations of these scrolling techniques. When developing further iterations, more measurement methods would need to be included to gain a greater understanding of where participants are looking during testing. “Eyetracking web usability” (Nielsen. and Pernice., 2009) suggests using heat-maps as a method of documenting where participants are looking and for how long. Understanding these factors may provide better data and understanding of how participants are using the system.

One other major limitation that needs to be addressed with the project is the lack of participants that had any major disabilities. This problem has been experienced in various other studies, such as the study by Kocejko et al. (2009) investigting camera mouse control, which, while providing positive data, only tested on able-bodied participants. When testing, due to ethical restrictions, only non-vulnerable participants were able to participate in testing, meaning that participants that may have truly benefited from the system were unable to participate with this system. This presents a problem of the system being developed and designed with only able-bodied people in mind for controlling it. To address this potential problem, the next stage of developing these scrolling techniques must include participants with a range of disabilities. These disabilities must impact the participant's ability to use a mouse or other piece of conventional computer hardware of software. Participants with disabilities may also perform better at the tasks, as participants may have used eye-trackers or other various unconventional navigation techniques to control computer systems before. Some participants may even be experienced in controlling applications using only eye-tracking, and may even have specialised equipment to use to negate problems such as accidental head movement.

A conclusion that can be derived from these results is that with both results providing similar yet positive outcomes, both can serve as their own, equally effective methods of scrolling, as opposed to one being the better of the scrolling methods. Reflecting upon the survey results, participants were asked if they could offer any feedback on the various scrolling iterations. Participants requested that for further developments of these projects, the systems should have better accuracy and options for controlling speed. With clicking, these improvements could be implemented by programming the buttons to accept clicks at all times instead of when the mouse is still. The third series of buttons could also be added onto the system to add another scrolling distance option but this risks cluttering the interface. As for improving the hover system, a bigger input box would be required, along with making the input boxes closer together to avoid the eye travelling as much distance. The input boxes could also scroll faster the higher the user looks, to enable the user to scroll large distances in a shorter amount of time. Another improvement to consider when reflecting upon the observational data that was gathered was improving the peripheral vision aspect of the clicking iteration of the scrolling system. When observing the clicking variation of the developed scrolling system, it can be seen that the clicking buttons are located on the right side of the layer list, while the layer numbers are located on the left side of the layer list. To solve this problem, the scrolling buttons could be moved to the left side of the layer list instead. This would allow participants to see both the clicking buttons and the numbers of the layers at the same time because of their close proximity. A potential problem that this solution presents, however, is that participants may find scrolling being located on the left offputting, as most conventional scrolling systems such as web browsers place their scrolling systems on the right side. Another solution for this problem would be to place the layer numbers on the right side instead so the scrolling isn’t affected.

7.0 - Conclusion

To conclude, the research and testing conducted shows that while accessibility options with computers have improved, finding a variety of options for users with accessibility issues is still a challenge. With the two scrolling systems developed, one of them can not be considered truly better than the other. While the clicking scrolling system proved to be a faster option, the hover scrolling system proved to be a more comfortable experience. With this information, it can be concluded that both iterations of the scrolling system hold some form of merit. With this considered, an ideal approach would be to improve and implement both iterations. If both iterations were improved and given new options such as controllable speeds and improved accuracy, both could be implemented as a scrolling system depending on the preference of the user. Programmes using layer scrolling systems would not lose significant usability by choosing to implement one system over the other. When designing further iterations of these scrolling systems, scrolling functionality should match to create a fairer testing environment. In turn, conducting more testing with a greater study population to achieve more significant results, as well as extending inclusion criteria to involve the intended user demographic, could also prove to be highly beneficial in seeing and understanding if these scrolling systems have a future in computer applications.

8.0 - References

Adjouadi, M., Sesin, A., Ayala, M. and Cabrerizo, M., 2004, July. Remote eye gaze tracking system as a computer interface for persons with severe motor disability. In International conference on computers for handicapped persons (pp. 761-769). Springer, Berlin, Heidelberg.

Albert, W. and Tullis, T., 2013. *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes.

Baber, C. and Stanton, N.A., 1996. Observation as a technique for usability evaluation. Usability Evaluation in Industry. London, Taylor & Francis, pp.85-94.

Bangor, A., Kortum, P. and Miller, J. (2009) 'Determining what individual SUS scores mean: Adding an adjective rating scale', *Journal of usability studies*, 4 (3), pp. 114-123.

Brooke, J. (1996) 'SUS-A quick and dirty usability scale',. *Usability evaluation in industry*, 189 (194), pp. 4-7.

Bruneau, D., Sasse, M.A. and McCarthy, J.D., 2002. The eyes never lie: the use of eyetracking data in HCI research. ACM.

Bruun, A., Gull, P., Hofmeister, L. and Stage, J., 2009, April. Let your users do the testing: a comparison of three remote asynchronous usability testing methods. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1619-1628). ACM.

Corno, F., Farinetti, L. and Signorile, I., 2002, August. A cost-effective solution for eye-gaze assistive technology. In Proceedings. IEEE International Conference on Multimedia and Expo (Vol. 2, pp. 433-436). IEEE.

Farokhian, S., 2019. Human-computer interaction using eye-gaze: Formation of user interface design guidelines from a cognitive science perspective.

Filetti, M., Tavakoli, H.R., Ravaja, N. and Jacucci, G., 2019. PeyeDF: an eye-tracking application for reading and self-indexing research. arXiv preprint arXiv:1904.12152.

Goldberg, J.H., Stimson, M.J., Lewenstein, M., Scott, N. and Wichansky, A.M., 2002, March. Eye tracking in web search tasks: design implications. In Proceedings of the 2002 symposium on Eye tracking research & applications (pp. 51-58). ACM.

Houtenville, A.J., Brucker, D.L. and Lauer, E.A., 2016. 2015 Annual Disability Statistics Compendium. *Institute on Disability, University of New Hampshire*.

Jacob, R.J. (1995) ‘Eye tracking in advanced interface design.’, *Virtual environments and advanced interface design*, pp. 258-288.

Kumar, M., Winograd, T., Winograd, T. and Paepcke, A., 2007, April. Gaze-enhanced scrolling techniques. In CHI'07 Extended Abstracts on Human Factors in Computing Systems (pp. 2531-2536). ACM.

Kocejko, T., Bujnowski, A. and Wtorek, J., 2009. Eye-mouse for disabled. In Human-computer systems interaction (pp. 109-122). Springer, Berlin, Heidelberg.

Lazar, J. ed., 2007. *Universal usability: Designing computer interfaces for diverse user populations*. John Wiley & Sons.

Lazar, J., Feng, J.H. and Hochheiser, H., 2017. *Research methods in human-computer interaction*. Morgan Kaufmann.

Liang, Y., Wang, W., Qu, J. and Yang, J., 2019, February. Application of Eye Tracking in Intelligent User Interface. In *Journal of Physics: Conference Series* (Vol. 1169, No. 1, p. 012040). IOP Publishing.

Liebling, D.J. and Dumais, S.T., 2014, September. Gaze and mouse coordination in everyday work. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (pp. 1141-1150). ACM.

Majaranta, P. and Räihä, K.J., 2002, March. Twenty years of eye typing: systems and design issues. In ETRA (Vol. 2, pp. 15-22).

Manhartsberger, M. and Zellhofer, N., 2005. Eye tracking in usability research: What users really see. In Usability Symposium (Vol. 198, No. 2, pp. 141-152).

Nielsen, J. and Pernice, K., 2010. *Eyetracking web usability*. New Riders.

Pavani, M.L., Prakash, A.B., Koushik, M.S., Amudha, J. and Jyotsna, C., 2019. Navigation Through Eye-Tracking for Human–Computer Interface. In *Information and Communication Technology for Intelligent Systems* (pp. 575-586). Springer, Singapore.

Platt, D., 2016. *The Joy of UX: User Experience and interactive design for developers*. Addison-Wesley Professional.

Poole, A. and Ball, L.J., 2006. Eye tracking in HCI and usability research. In *Encyclopedia of human computer interaction* (pp. 211-219). IGI Global.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. and Carey, T., 1994. *Human-computer interaction*. Addison-Wesley Longman Ltd.

Purchase, H.C., 2012. *Experimental human-computer interaction: a practical guide with visual examples*. Cambridge University Press.

Rivu, S., Abdrabou, Y., Mayer, T., Pfeuffer, K. and Alt, F., 2019, June. GazeButton: enhancing buttons with eye gaze interactions. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (p. 73). ACM.

Rose, A.F., Schnipper, J.L., Park, E.R., Poon, E.G., Li, Q. and Middleton, B.(2005) 'Using qualitative studies to improve the usability of an EMR.', *Journal of biomedical informatics*, 38 (1), pp. 51-60.

Shakir, M.S., Akhtar, A., Kulsum, H., Mukhtar, A. and Saleem, I., 2019, February. Enhancing Cursor Control Using Eye Movements. In *International Conference on Intelligent Human Systems Integration* (pp. 909-913). Springer, Cham.

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N. and Diakopoulos, N., 2016. *Designing the user interface: strategies for effective human-computer interaction*. Pearson.

Sibert, L.E. and Jacob, R.J., 2000, April. Evaluation of eye gaze interaction. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems (pp. 281-288). ACM.

Speicher, M., Both, A. and Gaedke, M., 2014, July. Ensuring web interface quality through usability-based split testing. In International conference on web engineering (pp. 93-110). Springer, Cham.

Su, M.C., Wang, K.C. and Chen, G.D. (2006) 'An eye tracking system and its application in aids for people with severe disabilities.', *Biomedical Engineering: Applications, Basis and Communications*, *18* (06), pp. 319-327.

Velichkovsky, B., Sprenger, A. and Unema, P., 1997. Towards gaze-mediated interaction: Collecting solutions of the "Midas touch problem". In Human-Computer Interaction INTERACT'97 (pp. 509-516). Springer, Boston, MA.

Vrzakova, H. and Bednarik, R., 2013, April. That's not norma (n/l): a detailed analysis of midas touch in gaze-based problem-solving. In CHI'13 Extended Abstracts on Human Factors in Computing Systems (pp. 85-90). ACM.

Wang, J., Antonenko, P., Celepkolu, M., Jimenez, Y., Fieldman, E. and Fieldman, A. (2019) 'Exploring Relationships Between Eye Tracking and Traditional Usability Testing Data.', *International Journal of Human–Computer Interaction*, 35 (6), pp. 483-494.

Zhou, T., Zhou, Y., Wang, D., Zhao, Y., Liu, H., Zhang, C., Shi, J., Ma, C., Wang, X., Wang, X. and Niu, J., 2019, February. The Influence of Target Layout on Dragging Performance Based on Eye-Control Technique. In *International Conference on Intelligent Human Systems Integration* (pp. 265-270). Springer, Cham.

9.0 - Appendices

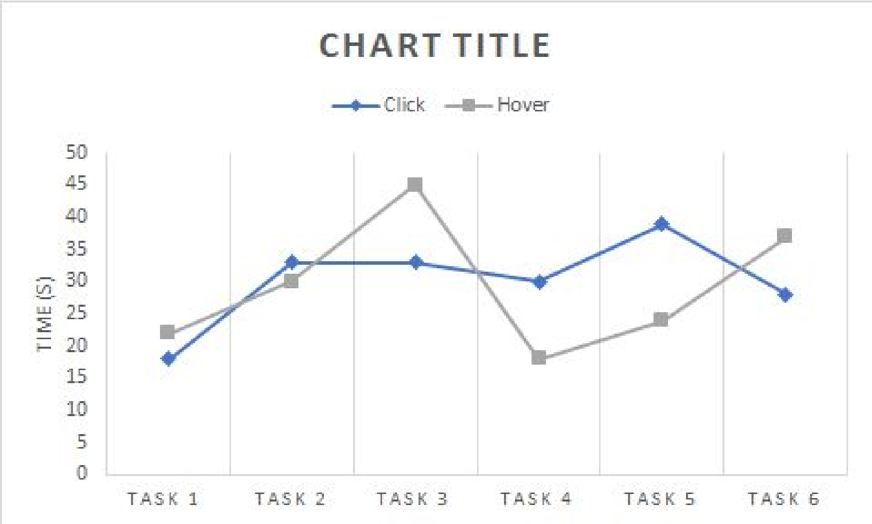


Figure 23. Diagram showing the changes in average times between both tasks.

What is your reason for your previous answer?

26 responses

It took a few tries to get used to, specifically for getting the accuracy right, but after a few tries it worked very well

I got the hang of it very quickly. Using the arrows seemed easier than hovering on the bar used in previous test.

Couldn't always move the move with eye recognition

Easier with clicking arrows

Little difficult to scroll smoothly

really difficult as have to keep checking the numbers which moves the mouse - very confusing.

not computer literate

difficult to control

easy to use

As before it's intuitive and straightforward.

It was easy and simple to use

Figure 24. Sample of qualitative data gathered from the clicking survey. Participants were asked to justify their answers when asked about how usable they found the clicking iteration.

What aspects of the click button scrolling did you like/dislike?

26 responses

I liked the accuracy of the scrolling to track my eye movement, although the accuracy could be improved when looking at small tick boxes etc.

I lost control of it meaning one task took a long time.

sometimes difficult to get eye recognition to work

Easier to use

Having to move from scroll button with eyes to check where you are

I liked very little - having to retrain your brain IS PRETTY DIFFICULT AND IT REALLY DIDN'T TRACK where my eyes were very well.

challenge

could not control between scrolling/clicking

I found it was easy to use

I liked it for the same reasons as above.

it was very simple and fun to use, sometime can be a little quick

Figure 25. Sample of qualitative data gathered from the clicking survey. Participants were asked what aspects of the clicking iteration they liked and disliked.

Did you experience any bugs or glitches while using the click buttons? If so, please describe the encountered problems.

17 responses

- No
- no
- I dropped one of the layers and lost it for a while causing a long pause.
- nil
- yes, would not let me scroll up many times even when the bubble was hovering in the correct space.
- Sometimes the mouse didn't follow the eye tracker bubble
- yes, a block sign would sometimes come up and i wasn't able to click on anything.
- n/a
- Sometimes the mouse click wouldn't respond even though it was held over the target button.
- Difficulty clicking at one point which meant I couldn't scroll up and was stuck
- clicks on buttons didn't always register
- sometimes would unintentionally drag instead of pressing button

Figure 26. Sample of qualitative data obtained from clicking survey asking about any problems experienced.

What recommendations could you give for this project going forward?

26 responses

- n/a
- na
- Potentially have a trial task first to get used to the idea of clicking the mouse and using the scrolling tracker
- Perhaps allow the user to adjust the scroll speed.
- better hit rate with eye recognition
- This system was easier to use so I guess more integrating mouse and eye tracker
- If scrolling and dropping smoother would be very easy to use
- Needs more improvement
- none
- make system easier/provide training
- It's difficult to see what layer you're on while you're looking at the scroll buttons to click on them, meaning you need to keep darting your eyes back and forth to check the layers to see where you are relative to where you want to be. I must admit however, I'm not sure of any way this could be remedied.

Figure 27. Qualitative data from clicking survey asking participants what recommendations they could offer to the project.

If you experienced discomfort with the click method, please describe what was affected.

7 responses

- n/a
- Feeling like i had to keep my head in one position
- eyestrain
- mild eye strain
- eyes watering

Figure 28. Sample of clicking qualitative data asking participants to provide detail on any strains experienced.

	Strongly Disagree				Strongly Agree
1. I think that I would like to use this product frequently.	1	2	3	4	5
2. I found the product unnecessarily complex.	1	2	3	4	5
3. I thought the product was easy to use.	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this product.	1	2	3	4	5
5. I found the various functions in the product were well integrated.	1	2	3	4	5
6. I thought there was too much inconsistency in this product.	1	2	3	4	5
7. I imagine that most people would learn to use this product very quickly.	1	2	3	4	5
8. I found the product very awkward to use.	1	2	3	4	5
9. I felt very confident using the product.	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this product.	1	2	3	4	5

Figure 29. The SUS scale survey used as part of this project (Bangor et al., 2009)

What is the reason for your previous answer?

26 responses

It was fairly easy to use but took some time to scroll to the higher numbers

Seemed to be a variation in the speed of scrolling between some tests.

Easier to use

I had difficulty scrolling upwards

Drop part of drag and drop difficult

It was usable but still much harder than just using a normal mouse

better with practice

initially difficult to get used to but once used to it became easier

easy to use but sometimes scrolls to quick

Everything was fairly accurate and responsive.

It was very simple to use

I thought it was simple to use

Figure 30. Sample of qualitative data gathered from the hover survey. Participants were asked to justify their answers when asked about how usable they found the hover iteration.

What aspects of the hover scrolling did you like/dislike?

26 responses

It looks simple and easy to use

Was very easy to learn and master the idea of how to control the scrolling.

Scrolled easier

I liked scrolling downwards and callibrating but disliked scrolling up

Dropping scrolling movement is a bit to quick

quick but sometimes slightly out of control

novelty

just strang to get used to using keyboard/mouse

easy to use

It feels straightforward and responsive, it's great for its ease of use.

i liked the fact it was easy to use, the dragging was a little quick

I liked how it does what it says it does

Figure 31. Sample of qualitative data gathered from the hover survey. Participants were asked what aspects of the hover iteration they liked and disliked.

If you experienced discomfort with the hover method, please describe what was affected.

11 responses

n/a

Eye strain as I felt they kept rolling back into my head as I tried to manipulate the cursor upwards

nil

bluured vision as i had to keep my eyes fixed to the bottom

Only slightly dry eyes when scrolling far as the eye tracker bubble moved when I blinked and interrupted scrolling

felt eye strain

eye strain from trying to be accurate

Eyes hurt

Felt my eyes were straining a bit

eyes watering

Figure 33. Sample of hover qualitative data asking participants to provide detail on any strains experienced.

Did you experience any bugs or glitches while using the hover method? If so, please describe the encountered problems.

17 responses

no

No

n/a

Some tests scroll at different speeds. The fastest was difficult to control as I kept overshooting my target.

Nope

nil

Not as easy as scrolling with mouse as eye movement less easy to control precisely.

slight error in the first try at calibration

none

Figure 34. Sample of qualitative data obtained from hover survey asking about any problems experienced.

What recommendations could you give for this project going forward?

26 responses

n/a

na

Make an option to scroll through the lists faster as well as the other scrolling options

Standardise the scrolling speed. Perhaps make the scroll bar a little larger.

Difficult to say

Good project - maybe alter scrolling upwards or maybe a verbal tutorial ?

If the scrolling up and down was more smooth would be very easy to use

eye scrolling was interesting but I still think some work needs to be done on comfort of the user

none

training

Perhaps the scrolling would benefit from having the up and down buttons slightly further apart from where the layers actually are. This may prevent a user from accidentally scrolling when they are attempting to look at a layer at the bottom of the current view-able list.

Figure 35. Qualitative data from hover survey asking participants what recommendations they could offer to the project.