University of Ulm | 89081 Ulm | Germany

Faculty of Engineering, Computer Science and Psychology
Institute of Databases and Information Systems

Answering a Questionnaire Using Eyetracking

Master's Thesis at Ulm University

Presented by:
Laura Irlinger
laura.irlinger@uni-ulm.de

Reviewer:
Prof. Dr. Manfred Reichert
Dr. Winfried Schlee

Supervisor:
Marc Schickler

2016

Satz: PDF-LaTeX

Abstract

The beginning of eye tracking research lies far back in the past. Since eye tracking costs decreased over the past years, the usage of an eye tracker for everyday matters, like the interaction with a personal device, becomes more and more attractive.
In the present work, the realization of interacting with a computer interface with only the help of an eye tracker is illustrated. The conducted study examines the acceptance and usability of such a system.
Therefore, three different interaction methods have been implemented. In a study, the participants had to complete a questionnaire with those interaction methods using a Windows application and a low-cost eye tracking device.
All in all, the study results imply that the number of negative aspects about this system outweigh the positive ones. The biggest issue was the restriction of mobility during the usage of the tracking device. In addition, the usage of the system turned out to be rather exhausting for the eyes. Generally speaking, among the three implemented interaction methods, the interaction method that combines gaze with a second input modality (a keyboard) scored best in terms of interaction speed and usefulness considering the completion of a questionnaire.

Gratitude

First, I would like to express my sincere gratitude to my thesis supervisor, Marc Schickler from Ulm University for offering his continuous advice alongside the development of this thesis. Furthermore, I would like to acknowledge Professor Doctor Manfred Reichert and Doctor Winfried Schlee from Ulm University for reviewing this thesis. I am also thankful for the support of various people during my study, conducted for this thesis. Conclusively, I would like to thank my family and friends for their love and support during this thesis.

Contents

1 Introduction

Eye tracking has inspired people some time now. The eye tracking history starts as far back as in the 1800s where the first studies about eye movements have been conducted [WT05]. Since then, eye tracking became a major area of research. Investigating the usability of computer interface and enabling eye tracking as a computer input device became popular in the past years [JK03]. With the decreasing costs of eye trackers, the usage of these devices becomes more and more attractive. Now, some eye tracking devices are even affordable for the average human which makes it more and more socially acceptable. Simple user tracking (often head tracking) is already integrated in some devices of everyday usage. For example, some smartphones are able to keep the screen (brightness) on if the user is still looking at the phone or videos can be paused by looking away from the screen. The technical capabilities of modern smartphones [Gei+14; Sch+15a] could even play a part in contributing to use eye tracking day-to-day.

Presumably, devices with implanted eye tracking hardware and integrated eye tracking software will become common practice in the future.

1.1 Motivation for Eye Tracking

Today, many possibilities already exist where the usability of websites and web applications, among others, can be assessed with the help of eye tracking. Therefore, the user's eyes are tracked and a heatmap or other visualizations can be produced. So, a company is able to get insights into their systems. More precisely, eye tracking does not only help a company to investigate where a participant is looking at but also what they were doing respectively why they were doing so. Consequently, conclusions about the positioning of elements can be drawn, for example, in order to detect if the user noticed an element. Furthermore, differences between various user groups can be identified, such as variations between novice and experienced users.

It should not be forgotten that eye tracking can simplify the life of disabled persons like revealed in [Rud]. It allows people to control PCs and tablets among other devices with just their eyes. So, a loss of motor skills or rehabilitative disabilities does not hinder people from interacting with their devices anymore. This is a huge step towards the independency of the aforementioned user group(s). Possible tasks that can be performed with just the eyes are, among others, surfing the Web, sending E-Mails and text messages, play computer games and potentially interact with any computer program [Rud]. For example, the *Eyegaze Edge Talker*, introduced by LC Technolgies [Teca], an eye-operated communication system, helps people to communicate with their environment. The user types a sentence or chooses a pre-programmed one and the program generates speech.

Eye tracking can also be adjusted to specific user groups. Groups of persons who need their hands to perform a task could benefit from an eye tracking enabled system. Military pilots are one potential user group for this scenario. They need their hands to coordinate a military aircraft. In doing so, tasks could be performed with the help of eye tracking. Consequently, the hands remain free for other tasks at all time.

When it comes to the average user, most never interacted with an eye tracker by now. So, using just the eyes for input is rather new for most people. For them, eye tracking can also enrich the interaction with PCs and other devices.
Generally speaking, eye movements are fast. Accordingly, eye-based input could be performed faster than with any other currently utilized input device. By utilizing the same organ for perception as for input, the detour, that is a dedicated input device, can be circumvented. Thus, potentially improving interaction speed and rendering the experience more intuitive could be achieved.
Since eye movements are natural, the user can benefit from this and no further learning or practice is necessary. Of course, depending on the implemented interaction method, further learning could still be required.
Moreover, the user keeps their hands and arms busy when using a computer. With eye tracking, the arms and hands remain free, so a benefit could be the reduced stress for arm and hand muscles. Furthermore, the hands would be still free to do something else with them.
Generally speaking, more effective computer interaction methods could be developed when using gaze (in combination with other input modalities) as an input mechanism.

1.2 Thesis Objective

The purpose of this thesis is to investigate the acceptance of eye tracking as an interaction method. In so doing, the participants need to complete a predefined task. Here, the completion of a questionnaire came into mind. A questionnaire is well known by the average user. Moreover, a questionnaire can contain a lot of different input elements. So, not only the general usage but also the usage when it comes to different subtasks (for example, selecting a radiobutton and moving a slider) can be evaluated. Generally speaking, the user shall be able to complete a questionnaire with the help of their eyes. Therefore, three different interaction methods will be implemented and a user study will be conducted, at the end. The acceptance of the system in total shall be investigated. Furthermore, the different interaction methods shall be compared and advantages as well as disadvantages shall be identified. Since the probands will complete a question-

naire when working with the eye tracker, another research question is whether such a system is especially useful to perform the task of completing a questionnaire.
All in all, the acceptance of such systems and the willingness to integrate such interaction methods into our daily lives will be investigated.

1.3 Thesis Structure

This thesis consists of five main chapters.
Chapter 2 summarizes some of the related work. Here, so far investigated gaze-based interaction methods are highlighted. More precisely, 'gaze-only' interaction methods followed by examples where gaze is combined with a second modality are given. Furthermore, possibilities to give feedback for gaze-based interactions are shown.
An overview of eye tracking is given in chapter 3. The history of eye tracking as well as the different eye tracking categories are introduced. A choice of current eye tracking devices (wearable and remote devices) is given. Example eye tracking applications are given, as well. Furthermore, the attention is drawn to the limitations of gaze-based tracking.
The focus of chapter 4 is the background of the implementation and the implementation itself. Here, the concept for the implementation as well as the choice of the eye tracker, that will be used for the implementation and the study, are represented. The chosen eye tracker is presented in greater depth and the implementation used for the study is explained in detail.
The preparation of the study and the study conduction itself is introduced in chapter 5.
The study analysis is illustrated in chapter 6. Next to general information about the study participants, the overall user experience is investigated. Here, not only subjective data from a questionnaire has been regarded but also the objective data from the generated logfiles.
Finally, a conclusion and potential future work are given in chapter 7. Here, the most relevant topics are summarized and the results are outlined once more, followed by potential future prospects.
An overview about the thesis structure is given in figure 1.1.

Figure 1.1: An overview about the thesis structure.

2

Related Work

With the decreasing costs of eye trackers, gaze-based interaction with a User Interface (UI) becomes more and more attractive. Majaranta and Bulling [MB14] described that, different from perceiving the interface with the eyes and controlling it with the help of another input device (usually a mouse), gaze-based interaction uses the same modality for the perception and the control of an interface. So the system needs to differentiate between casual viewing and the intentional control. Jacob [Jac90] introduced the *Midas Touch* problem where the viewed objects are 'directly' selected (for further information see section 3.5). For this a lot of different possible interaction methods for the selection of an object have been introduced and evaluated over the years, for example *Dwell Time*, blinking, combinations with a key press or speech. In the following, these methods will be examined more precisely.

2.1 Gaze-Based Interaction Methods

When it comes to gaze-based interaction methods a distinction can be made between the 'gaze-only' interaction methods and the ones that combine gaze and a second modality to support the interaction.

2.1.1 Gaze-Only Interaction Methods

In gaze-only interaction methods the user is able to manipulate the interface with their eyes. Hence, the user is able to not only select but also trigger actions only with their gaze. An often used gaze-only interaction method is the so called *Dwell Time*. Gaze gestures, blinking and *Context Switching* are possible methods, as well.

Dwell Time

A widely used technique is to use *Dwell Time* to select an object. Therefore, the user has to fixate the object for a sufficient time to select it. *Dwell Time* is often used to implement gaze-based writing. On the one hand, long *Dwell Times* reduce false selections. On the other hand, a user wants to be able to use an interface quickly. Once an object is located on the screen the pointer is already there. With a mouse a user investigates the interface first and starts to move the mouse afterwards. So, long *Dwell Times* reduce the effectiveness of gaze-based interactions (the user waits until they are able to interact with a target). It can be hard for a user to achieve long *Dwell Times* because of possible interruptions (like blinking). Furthermore, it can cause eye fatigues. That is why one design issue when using *Dwell Time* is to choose a suitable time duration [Jac90]. Normally the *Dwell Time* lays between 600 ms and 1000 ms [TM14]. Shorter *Dwell Times* can be used for advanced users. Majaranta et al. [MAR04] studied the impact of feedback when it comes to eye-typing with short *Dwell Times*. The participants on their study have already participated in an earlier study before. So, they were not novice users anymore. They found out that 300 ms is long enough to react.
An individually adjustable *Dwell Time* can be implemented to suit the needs of every user.

ERICA, introduced by Lankford [Lan00], is a system that allows a user to perform various mouse clicking actions like dragging, clicking and double clicking as well as eye typing (which is not taken into consideration here) with the help of *Dwell Time*. The mouse cursor moves with the gaze. If the user fixates a point on the screen, a red rectangle appears that starts to collapse. Half way through the collapsing state the rectangle will become a blue circle. Dragging is now possible if the user looks away. When the user keeps the fixation up till the circle completely collapses, the system clicks once. Afterwards a green rectangle appears. After another fixation duration, the system double clicks. All collapse states can be seen in figure 2.1.

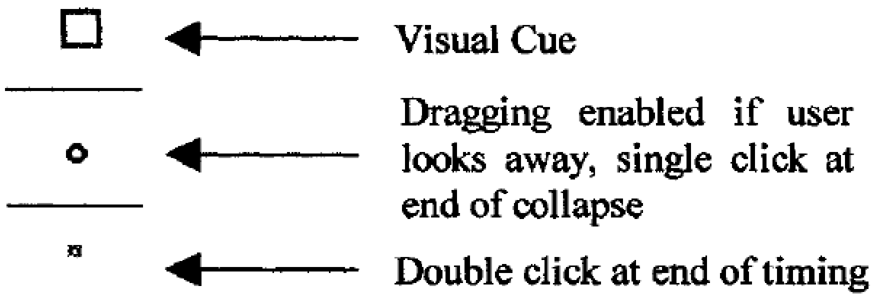


Figure 2.1: Three different gaze clicking collapse stages in *ERICA* [Lan00].

With this approach all commonly used mouse actions can be realized. Furthermore, it is possible to just activate some actions and disable others (i.e. disable double clicking). To reduce the waiting for some mouse actions (like double clicking) caused by the *Dwell Time*, another approach had been introduced. A menu (which can be seen in figure 2.2) with all possible mouse actions appeared as soon as the red rectangle reached a collapsed state. Now, the user simply fixates the action they like to perform [Lan00].
De Luca et al. [DDLS07] tested *Dwell Time* and gaze gestures (see paragraph Gaze Gestures) for the usage of mobile devices. Therefore, they implemented a phone book application. The user was able to scroll through their phone book entries and chose one person they like to call. *Dwell Time* was set to 2000 ms. Some participants of the taken study complained about feeling stressed. They were forced to look away to prevent selecting entries they only wanted to observe. On the contrary *Dwell Time* based interaction is much more intuitive than gaze gestures.

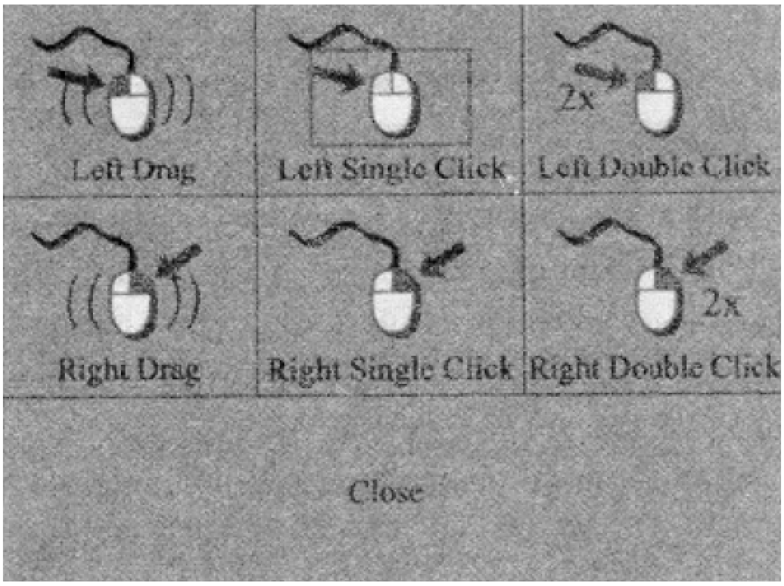


Figure 2.2: The click menu that appears after fixating on a spot in *ERICA* [Lan00].

Eye-S introduced by Porta and Turina [PT08] is a system to enable eye input to the computer, especially gaze-based typing. It uses *Dwell Time* as well. Therefore, nine different so-called hotspots are located on the screen (3x3 grid). The user has to fixate between two and four different hotspots to type a letter (see figure 2.3). In the undertaken study the *Dwell Time* was 400 ms. If a user has to fixate four different hotspots to form a letter, the *Dwell Time* for all four combined was 1600 ms. So the issue that *Dwell Time* is 'slow' shows again. The sequences to type letters were given, so the user had to learn them first. It is also possible to use *Eye-S* for general commands. Unlike alternative approaches no extra keyboard is needed, the screen remains free (hotspots can also be invisible).

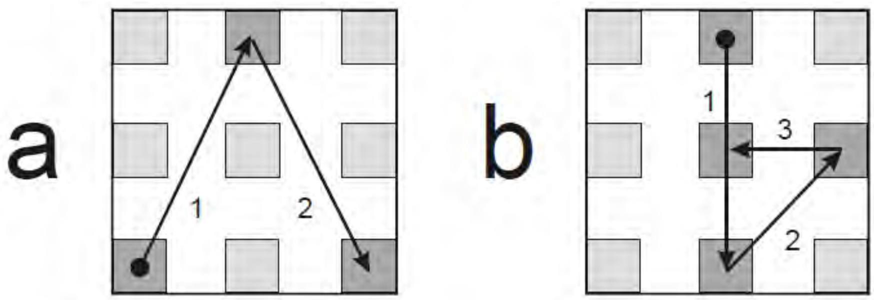


Figure 2.3: Possible eye sequences for typing letters in *Eye-S* [PT08].

Gaze Gestures

Gaze gestures are patterns of eye movements that trigger actions. That is why one of the major challenges with gaze gestures is the need to learn the given gaze gestures first. If the gestures are complex, it is not a natural behavior for the eyes to perform but if they are too simple, false unintended inputs could occur. Furthermore, complex gaze gestures can be time consuming. The difference compared to *Dwell Time* is that these sequences depend on relative coordinates, not absolute ones. This is especially useful if no calibration is desired. It is not important where the pattern is recognized, only that it is recognized.

Heikkilä and Räihä [HR12] introduced a drawing application that works with eye-based interaction. Therefore, gaze gestures had been used as well as blinking. Eight gaze gestures have been implemented to make an object move, comprising the cardinal and ordinal directions (N, NE, E, SE and so forth). In this way, the gestures are easily remembered. As soon as the user's gaze is on a movable object and the move gesture can be started, the cursor's color changes from black to orange. A small crosshair supports the user. The gesture ends the moment the user's gaze remains outside of the screen area for at least 200 ms. A stop gesture is implemented to stop the movement by blinking (for more information see paragraph Blinking below). Although, simple gestures have been used here, a study showed that learning has a considerable effect on gesture performance.

De Luca et al. [DDLS07] used gaze patterns to interact with a mobile phone. Therefore, a prototype had been implemented where a user can open and close the web browser. The mobile phone edges and corners supported the user as help lines and help points with the gesture execution. To not trigger false positives, the sequences have to differ from the normal eye movements when using a smartphone. There is no specific feedback during the execution of the gaze gesture but the opening or closing of the browser. They also implemented a phone book application which uses *Dwell Time* (for an explanation of *Dwell Time* refer to 2.1.1) as an interaction method. The results of the user study showed that a good threshold for the upper time limit for each (gesture) stroke would be 1000 ms. Compared to *Dwell Time*, users noted that gestures are not performed as naturally as *Dwell Time*. Moreover, most users preferred *Dwell Time* over gaze gestures.

They also complained that it could be hard to remember especially complex gestures. One advantage is that false positives are nearly impossible if the gesture is complex enough. In this way the *Midas Touch* Problem (for further information see the introduction in chapter 2) can be avoided.

Blinking

Blinking is natural behavior for a user. So, no new techniques have to be learned. But this is also the big issue with blinking. A person blinks every now and then, so the system has to differentiate between reflexive blinking and blinking on purpose.

Like described in 2.1.1 above, Heikkilä and Räihä [HR12] introduced a drawing application with gestures and 'closing of the eyes'. To stop the movement of an object caused by gestures, the user has to close his eyes for 390 ms. As soon as the tracker loses the gaze point a timer starts. The coordinates of the user's gaze that the tracker lastly detected are always saved. When the time is over a feedback sound is played to let the user know that the stop gesture is done. All in all this approach worked well in this setting. However, closing the eyes is not the same as blinking, because normally a person blinks really fast and does not have to close their eyes for a specific time. Moreover, the gaze interaction takes longer.

Lankford [Lan00] also considered blinking as a possible methodology to realize *ERICA* (see paragraph Dwell Time above). Besides the issue described above, he also complains about the difficulty to detect blinking.

Selection Region and Context Switching

Another method to select an element with only the gaze is the use of so called selection regions. Therefore, the element that shall be chosen has to be looked at to trigger the appearance of the selection region (i.e. a colored border, a dot or similar). The advantage of this approach is that there is no performance limitations based on long time durations like in *Dwell Time* or other.

Patidar et al. [PRS14] introduced *Quickpie*, an interface for gaze-based eye typing. *Quickpie* consists of four different regions: pie, characters, safe and selection region. The structure of *Quickpie* can be seen in figure 2.4.

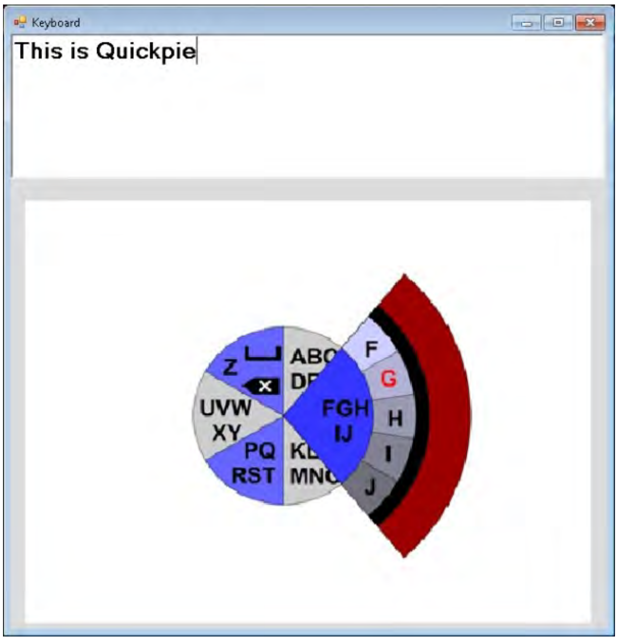


Figure 2.4: The interface of *Quickpie* (see [PRS14]) with its four regions: pie, characters, safe and selection region.

.

The pie contains all letters from the alphabet divided into six slices. When a user looks at one slice it is highlighted and it expands to show the characters region. In this region the characters are arranged outside of the pie. As soon as one character has been looked at, the text color becomes red. The safe region is black colored and is located the outer edge of the characters region. It is used to minimize errors caused by jittery movements (for further information see section 3.5) by the eye. The last region is the selection region which is located the outer edge of the safe region. This region is red and turns green as soon as the gaze enters the region. The highlighted character has been selected [PRS14].

Similar to this is the so called *Context Switching* paradigm [TM14]. Here, elements are grouped within areas (contexts) and these areas are separated by a bridge. The user fixates a key (i.e. for 150 ms) until it receives focus. Afterwards, the user crosses the bridge to activate a marker and saccades[1] back to a key in the interface. So, a two-step gesture has to be performed. This approach is used by the so called *Meta-Keys*

[1]Saccades are sudden, ballistic and nearly instantaneous eye movements. [Jac90]. For further information see paragraph Characteristics of Eye Movements

introduced by Tula and Morimoto [TM14]. A *Context Switching* paradigm is used. An example of this concept is illustrated in figure 2.5. The two-step gesture combined with the bridge reduced the number of false positives, avoids the *Midas Touch* problem described in paragraph Midas Touch as well as reduces the effect of eye noise. A user study revealed that *Meta-Keys* are easy to learn. Unlike with other approaches, there exist a temporal threshold because of the fixation time (150 ms) at the beginning.

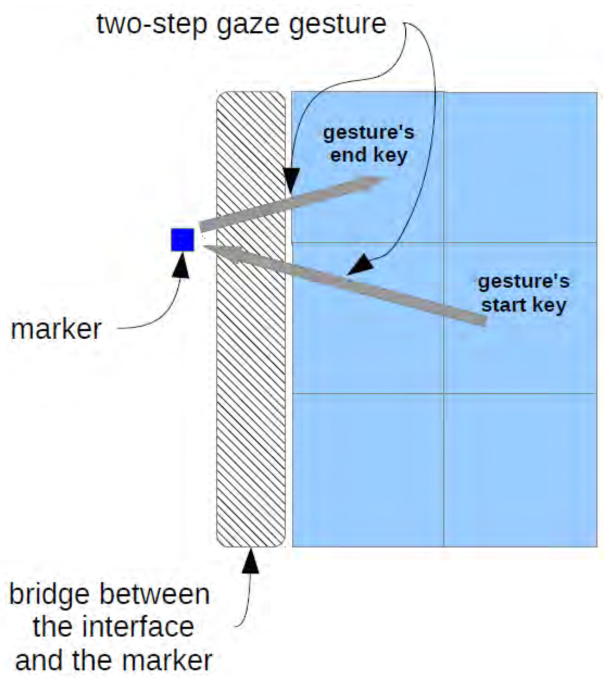
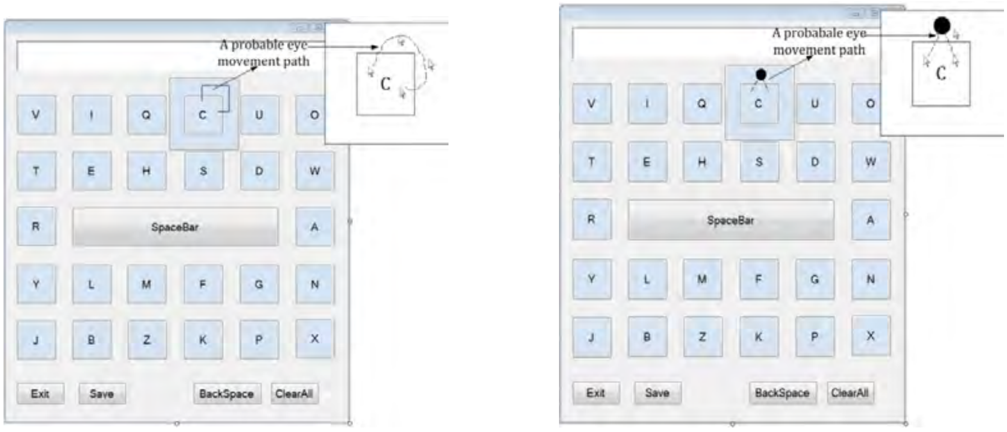


Figure 2.5: *Meta-Key* Concept: Selecting a key by focussing it, crossing the bridge to focus on a marker and saccade back to the interface by crossing the bridge again [TM14; PT08].

Chakraborty et al. [CSS14] introduced a similar concept for eye-typing. They implemented two methods to type without any *Dwell Times*. The first approach to select a character is to look at it followed by looking at the now appeared outside area and finally looking back at the character area (inside-outside-inside). This is shown in figure 2.6 (a). The second method is similar to the first one. As soon as looking at a letter, a black circle appears at the top of it in the outer area. Looking at this circle, followed by looking back at the key, selects the character. This method is shown in figure 2.6 (b). After both methods, the selected character's text color becomes red. Maintaining an experimental setup showed that participants like *Dwell*-free eye typing methods because it is easy to use and it is less fatiguing. All in all, users preferred method 2 (where the circle is displayed) even if users had problems with this method at the beginning. Also text entry rate and learning rate of method 2 were higher.



(a) Selecting a character by looking at the key, looking at the outside region and looking back at the key.

(b) Selecting a character by looking at said character, followed by looking at an appearing circle and finally looking back at the key.

Figure 2.6: Two methods to select a character by an eye movement path [CSS14].

2.1.2 Gaze Combined with a Second Modality

In this case the user's gaze is insufficient to manipulate the interface on its own. Hence, the gaze is combined with a second modality. The gaze is used to select an object and another modality triggers an action. Here the combination with a key press is prevalent. But the combination with touch, speech or other devices is common, as well.

Key Press

Users are familiar with the use of the keyboard. So, another approach is to combine gaze and key presses. With this the system can be used much faster as for example with *Dwell Time*. The user simply looks at the point on the screen where they want to click on and presses a key on the keyboard. Although this is a fast method, this approach is not available to some disabled individuals. To single click with the help of a key press has

been considered in *ERICA* (see paragraph Dwell Time above) as well [Lan00]. Looking on a point on the screen and clicking a key once performs a single mouse click while double clicking the key twice executes a double mouse click. Holding the key down allows dragging. In this case this approach had not been developed further because disabled individuals cannot use this method.
Kumar et al. [KPW07] introduced *EyePoint*, a system that combines gaze and key presses to point and select . Therefore, a look-press-look-release action had been used. The user looks at a target on the screen and presses a so called hotkey. A magnified view of the region appears. After fixating the target again the user releases the hotkey. This procedure can be seen in figure 2.7.

Figure 2.7: Pointing and selecting in *EyePoint* [KPW07] with a look-press-look-release action and the help of hotkeys.

Single click, double click, right click, mouse over or start click-and-drag exist as hotkeys. Click-and-drag is a two-step interaction where two hotkeys are used. Kumar et al. [KPW07] found that the performance of *EyePoint* compared to a common mouse usage is similar while error rates depend highly on the users and the calibration. Participants found that the gaze-based techniques are fatiguing over time. All in all, the users preferred *EyePoint* over the mouse because it is more natural.
A widespread problem with the combination of gaze and keyboard is the failure to synchronize gaze and triggers [Kum+08]. The so called *Late-Trigger error* arises when the user presses the key but the gaze has already moved to the next target. This problem only occurs when there are more targets on the screen. The opposite is called *Early-Trigger error*. This can happen due to many reasons: on the one hand there happen to be technical issues (like a sensor lag caused by the eye tracker), on the other hand the user looks at a target and presses the triggers before said target receives

focus. Consequently, error rates by combining gaze and keyboard have been very high in previous research.

Touch

With touch enabled interfaces a further possibility to interact is given. Normally, the user directly touches on the object they wish to manipulate. This can be annoying if the user's hand obstructs the view. Pfeuffer et al. [Pfe+14] introduced four different applications where the user is able to make a selection with the help of his gaze, but manipulates with (multi-)touch. Therefore, the user targets an object and performs a touch on the screen on any arbitrary position. In consequence, the user is able to see the entire object they manipulate and their view is not impaired by their hand(s). Furthermore, in some applications the user was still able to use direct touch (not gaze-touch) to manipulate objects. Therefore, the user's gaze and touch coordinates are compared. If the user touches on an object they also looks at, direct-touch is possible. All in all, gaze-touch showed benefits such as reachability, speed and no occlusion. Furthermore, the user profits from the similarities to direct-touch which reduces learning effort.

Voice

Eye tracking technology is especially attractive to people who are handicapped and can not use their hands or users who perform hands-busy tasks. So, another approach is to combine gaze and voice recognition. In this way the *Midas Touch* problem (for more information see paragraph Midas Touch) can be prevented and the interaction with the system remains hands-free. Furthermore, over the past years, voice recognition systems improved greatly [Hat+96]. Presumably, the combination of speech recognition and eye tracking simplifies the interaction process (compared to speech recognition) since the pointing does not need to be described verbally. Hatfield et al. [Hat+96] introduced the *Eye/Voice Mission Planning Interface* (EVMPI) for pilots and operators of advanced military systems. In this scenario, eye-tracking technology and voice recognition is used on cockpit displays. A user interacts with the system by looking at an item of the user interface and issuing verbal commands. One issue described is the aviation mission planning problem. For the prototype a tactical air strike scenario was chosen to serve as the basis. A pilot has to manipulate the various multi-function displays (with about sixty to one-hundred functions in a typical high performance combat aircraft). So, by replacing the interaction method only with voice recognition, the pilot would have to memorize the order of all commands. While the manual workload can be decreased the cognitive (memory) load should remain unchanged. Visual (i.e. object color change) and audio events (synthesized speech and non-speech audio as for instance button click sounds) will be used as system feedback.

Other Devices

Gaze is especially useful if the UI, someone likes to interact with, is too far away to simply reach it. Thus, normally used input devices like a mouse or a keyboard are not necessarily available. Stellmach and Dachselt [SD12] interact with a distant display with the help of gaze and an additional small touch-enabled device. Smartphones are commonplace and easy to use so the user does not have to rethink. One design goal here was to enable one-handed interaction, the user should be able to interact with the display with only his thumb on the touch-device. The device is subdivided into three areas: one to select a target (on the bottom of the device), one to deactivate the selection (at the top of the device) and the touch area in the middle of the device to interact with the selected target. In order to do so, a simple tap, a sliding gesture and dragging is implemented as well as rotation and tilting of the device. To improve the usability the so called *MAGIC touch* and *MAGIC tab* were introduced. With *MAGIC touch* the cursor can be moved according to the touch position (for example to select small and closely positioned targets). When it comes to *MAGIC tab*, spatially close items can be discretely cycled through for example with a continuous sliding gesture or by tilting the device. All in all, the eye tracking inaccuracies can be compensated with the help of these *MAGIC* techniques. In addition, a decent level of usability and a high overall performance was detected.

2.2 Feedback for Gaze-Based Interactions

Majaranta et al. [MAR04] investigate the effects of feedback on eye typing with *Dwell Time* in a study. Therefore the *Dwell Time* was set to 450 ms which is a rather short duration. Three different feedback modes were investigated: speech and two kinds of visual feedback (see figure 2.8). In the speech mode the chosen letter is spoken and no visual feedback is given. In the first visual mode ('1-Level Visual') the key will turn red as soon as it is selected while in the second visual mode ('2-Level Visual') the key turns red on selection as well but the key is also highlighted after 150 ms (when it is focused).

Feedback mode	While focused	When selected
Speech	none h	letter spoken h
1-Level Visual	none h	red background h
2-Level Visual	highlight h	red background h

Figure 2.8: Three different feedback modes for forming a letter: letter spoken, red background, red background combined with highlighting while focused (see [MAR04]).

All in all, the 1-Level Visual feedback produced the best results. The measured performance for 2-Level Visual feedback was good but it was found confusing. Speech feedback takes longer than for example a short visual feedback. Though, a user should not have to wait until (speech) feedback is finished. Most participants argued that the (very short) red flash does not seem to be enough and they would prefer an additional simple 'click' sound. Adding a non-speech auditory feedback was assessed as an effective way to improve visual feedback. Visual feedback was very important for many participants. All in all, it is essential to offer sharp and clear feedback [MAR04].

3

Eye Tracking Overview

The investigation of eye movements can be dated far back to the 1800s. The first eye tracking devices followed at the end of the 1800s [WT05]. Further details about eye tracking history is given in section 3.1.

According to [Duc07] and [Sun12], there exist two different techniques to measure eye movements: the ones that determine the position of the eye relative to the head and the ones that ascertain the orientation of the eye in space, also known as Point of Regard (POR). Furthermore, measuring methods are divided into four broad categories: *electro-oculography (EOG)*, *scleral contact lens/search coil*, *photo-oculography (POG) or video-oculography (VOG)* and *video-based combined pupil and corneal reflection*. These methods are explained in more detail in section 3.2.

Hereinafter, a choice of current eye tracking devices is introduced in section 3.3. These devices can be either remote (not attached to the user) or wearable (for example head mounted or glasses).

Eye tracking applications can broadly be divided in two main categories: On the one

hand diagnostic applications and on the other hand interactive applications. In diagnostic scenarios the eye's movements or POR is merely 'investigated' while the interactive usage of eye tracking comprises the user to be able to interact with an application respectively get a response [Duc07]. A more detailed overview of eye tracking applications is given in 3.4.
Conclusively, different limitations of (especially video-based) eye tracking is summarized and briefly explained in section 3.5. More precisely, the accuracy and latency of eye tracking devices and the so called *Midas Touch* problem is closer considered as well as problems caused by characteristics of eye movements and limitations caused by the user.

3.1 Eye Tracking History

The origin in eye tracking lays in the studies how eyes move and how they are positioned after a movement. In the 1800s a number of scientists were concerned with the nature of the human eye and eye movements. Investigations of Jan Evangelista Purkinje about reflections on the eyeball (for further information see section 3.2) play a part in eye tracking as do the findings of Louis Emile Javal. In 1879 Javal noticed that the eyes do not sweep smoothly during reading but pause on some words. So, a series of stops (fixations) and short rapid movements (saccades) (for further details on eye characteristics see paragraph Characteristics of Eye Movements) occur [WT05].

The first eye trackers were often invasive like the ones built by Delabarre and Huey at the end of the 1800s/the beginning of the 1900s. An eye cup has been used with a mounted lever. This was placed directly on the eye. When the eye position changes a pen moves over a rotating smoked drum. The tracker was able to measure eye movements during reading. Figure 3.1 shows the eye tracker by Huey and an example result [Gom+07].

A lot of other devices followed quickly. In the late 1800s Orschansky used an eye-cup with an attached mirror. So, he was able to record reflections. At the beginning of the 1900s Dodge and Cline developed the first non-invasive eye tracker and thus played a

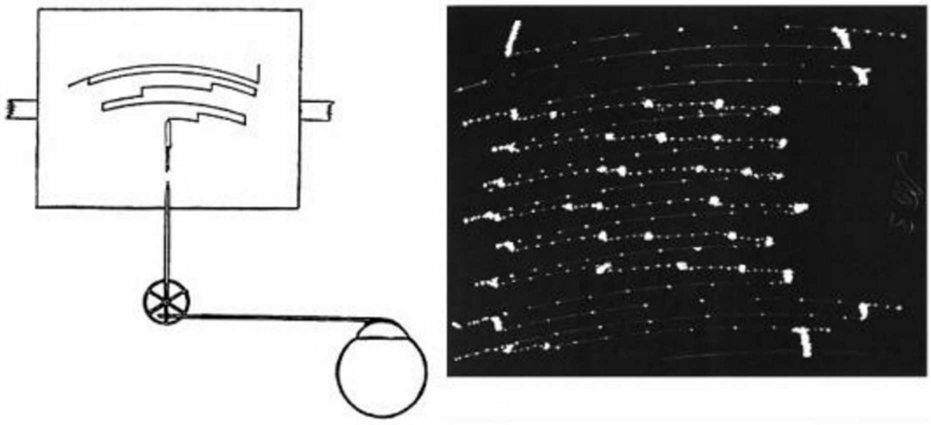


Figure 3.1: The tracker that is recording eye movements during reading is displayed on the left. An eye cup with a mounted lever was placed on the eyeball. As soon as the eye moves, a pen moves over a rotating smoked drum. In this way eye movements during reading were able to be measured. On the right, an example result is pictured. Here, six lines have been read (followed by two free movements) [Gom+07].

key role in the eye movement research. The device was based on photography and light reflection recordings from the cornea of the eye itself. So, it was much more comfortable for participants. The eye tracker is shown in figure 3.2 [LGE11].

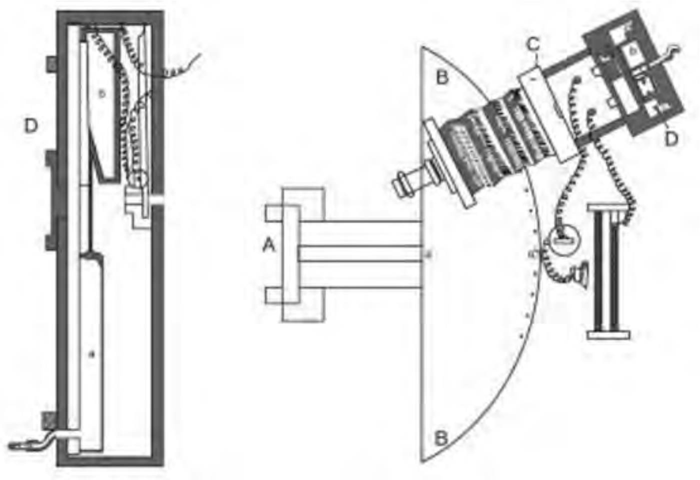


Figure 3.2: The eye tracker developed by Dodge and Cline to record eye movements [LGE11].

Subsequently, Dodge and Cline developed the so called *Dodge Photochronograph* that is displayed in figure 3.3. A photographic plate was used to record light reflected from the cornea. The photochronograph played an important role in the development of such technologies and inspired many upcoming eye tracking concepts [WT05]. It is deemed to be "the primary ancestor of the current video-based corneal reflection eye tracking systems" [Maj+12].

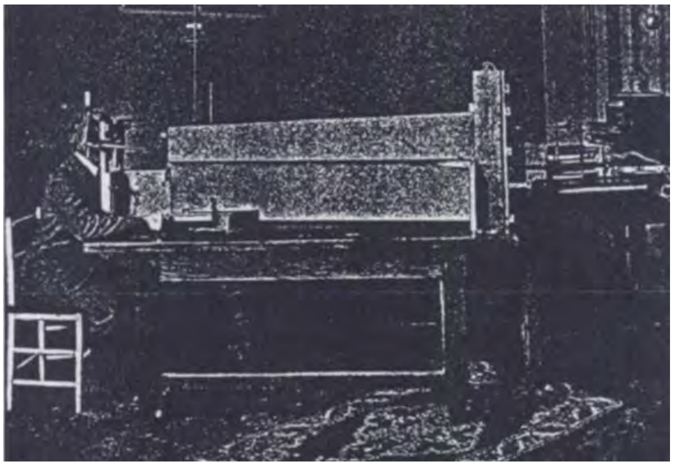


Figure 3.3: Light reflections from the cornea are recorded by the *Dodge Photochronograph* [WT05].

In 1935, Guy Buswell published his studies in eye movements in *How people look at pictures*. Therefore, he conducted a study with 200 participants and recorded their eye movement records during viewing multiple pictures. For the first time not eye movements during the investigation of text or simple geometrical patterns but complex pictures were systematically explored. Thus, this study became a key study in eye movement research and helped to understand eye movements [LGE11]. Bunswell's eye tracker can be seen in figure 3.4. Reflections from the eye's surface were recorded to measure the participant's eye movements. For this purpose, light in combination with lenses and mirrors were used. The camera was positioned next to the user. In this way the participants' field of vision was not covered. Moreover, head movements had been recorded to correct the eye position records [WT05].

In the mid-1900s, Yarbus, a Russian psychologist, published (in Russian) a pioneered and highly cited book. He carried out several studies to investigate eye movement

Figure 3.4: With Buswell's eye tracker, reflections from the eye's surface were recorded to measure the participant's eye movements. A series of lenses and mirrors were used. In this way the user's view was not covered [WT05].

characteristics. Therefore, different rubber suction caps were developed. During the study a cap was attached to the eye. Users need to be trained to prevent injury on the eye or damage on the device since the cornea was anesthetized and the lids were taped apart. The apparatus that is used to record the light reflections and examples of caps used for studies are shown in figure 3.5 [WT05]. Moreover, the eyes should not move. With this device, eye movements were studied while participants viewed pictures. He discovered a correlation between the task and the resulting eye movements [Maj+12].

In 1963, Robinson developed the first search coil eye tracker. Here, a contact lens with embedded wires is used (for further information see section 3.2). Its ancestors were introduced at the end of the nineteenth century. They were able to register eye movements but are not able to monitor their magnitude [WT05].

In the 1970s, eye movement research and eye tracking thrived once more [JK03]. Following in the 1980s, the use of personal computers was spreading out, so human-computer interaction issues came into focus. That is why researchers began to investigate how eye tracking could be used and integrated in this field. In this way, questions about user's

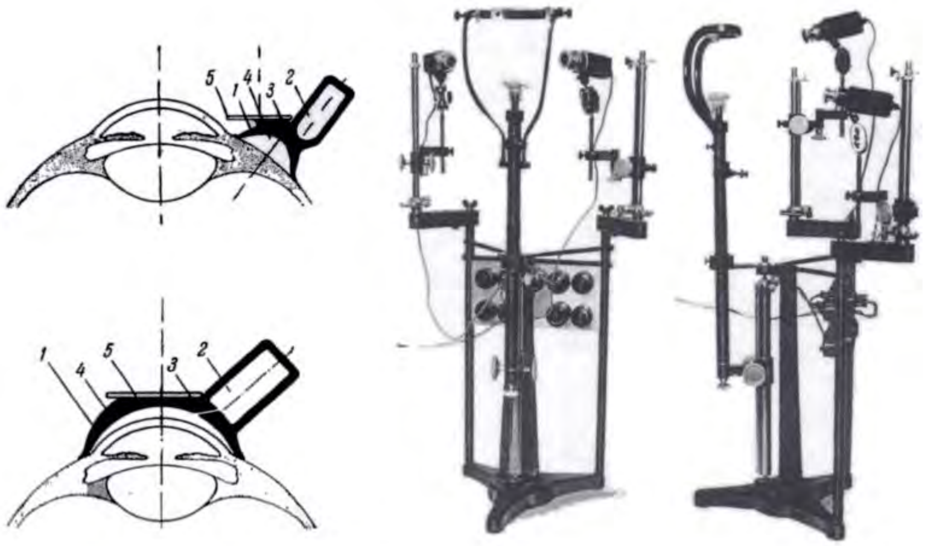


Figure 3.5: The left picture shows two example eye-cups. Yarbus developed a range of eye-cups with mirrors attached to it. With the help of light reflections eye movements can be recorded. On the right the apparatus that records the eye movements and illuminates the eye is displayed [WT05].

search habits, like searching for commands in computer menus, can be investigated. Moreover, eye tracking in real time became possible in the 1980s [JK03].

From that time onwards, eye tracking research in human-computer interaction increased. Consequently, investigating the usability of computer interfaces and enabling eye tracking as a computer input device moved into focus [JK03].

Today, eye trackers can be divided into four different categories which are discussed in the following section.

3.2 Eye Tracking Methods

There exist two different techniques to measure eye movements: by determining the position of the eye relative to the head and by ascertaining the orientation of the eye in space which is also known as POR. The most widely used method for the second approach is the video-based corneal reflection eye-tracker [Duc07; Sun12].

Normally, a calibration is needed to detect the POR. Therefore, usually a grid appears on the screen and a user has to fixate/follow a point/points. Afterwards, the system is capable of determining the POR (with a small variance).
Overall eye movement measuring methods can be divided in four broad categories: *electro-oculography (EOG)*, *scleral contact lens/search coil*, *photo-oculography (POG) or video-oculography (VOG)* and *video-based combined pupil and corneal reflection* [Duc07; Sun12].

The electronic method *electro-oculography* records electric potential differences of the skin. Skin electrodes are located around the eyes. Since there is a negative charge on the retina and a positive one on the cornea, the electric field changes when the eyeball moves. A DC or AC amplifier amplifies the potential [MF06]. The concept is represented in figure 3.6. However, serious noise issues caused by blinking, small ripple and other issues occur and the DC amplifier causes drifting of the *EOG* signal [Yag10].

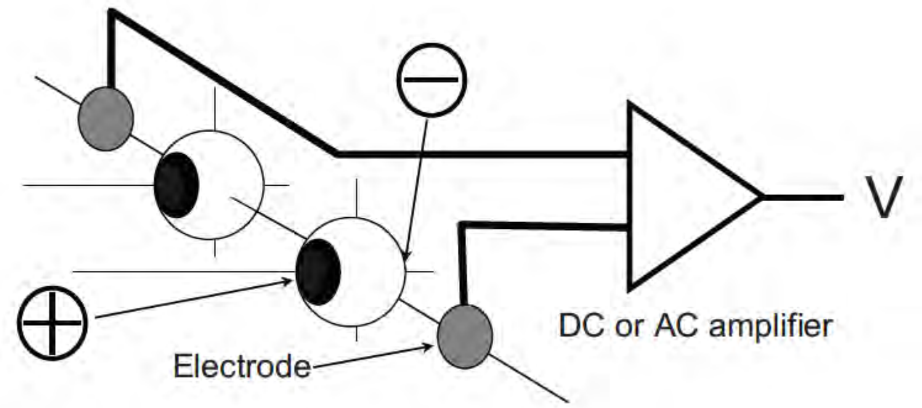


Figure 3.6: *Electro-Oculography* concept: Electrodes located around the eyes and connected to a DC or AC amplifier [Yag10].

According to [Duc07; Sun12], this method is more often used for relative eye movement measurements rather than absolute ones. It can only be used for POR measurements if the head position is tracked as well. *EOG* was widely used during the mid-1970s. Advantages of this method are that contact lenses and spectacle wearer can use this method and it is not very expensive [Duc07; Sun12]. Besides, nothing impedes the view. One disadvantage is the need of partly prominent wearable apparatuses. Figure 3.7

shows *EOG* electrodes integrated in a headphone [MF06].

Figure 3.7: *Electro-Oculography*: Electrodes integrated in headphones to measure relative eye movements [MF06].

For scleral contact lens/search coil, a (modified) contact lens is used. Different mechanical or optical reference objects are attached to the lens. The most popular ones are reflecting phosphors, line diagrams and wire coils (see figure 3.8). With the help of an

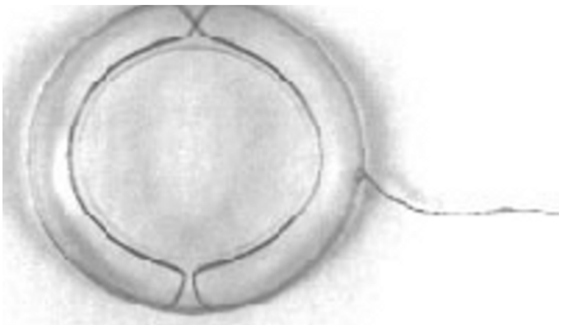


Figure 3.8: Sample contact lens with an embedded search coil. Eye movements can then be tracked with the help of an electromagnetic field frame [Duc07].

electromagnetic field frame eye movements can be tracked. It generally provides very accurate if not the most accurate measurements though this method is one of the more invasive ones because of the need to wear the contact lens [Duc07].

In *photo-oculography or video-oculography*, still or moving images (video) of one or both eyes are taken. Distinguishable features of the eyes are measured, for example the shape of the pupil or the position of the limbus. Both methods often do not provide the POR. *Video-oculography* normally works with infrared as light source [Duc07]. Figure 3.9 shows a head-mounted infrared *video-oculography* system.

Figure 3.9: The *3D VOG Video-Oculography System* by SensoMotoric Instruments is a head-mounted infrared *video-oculography* system for horizontal, vertical and torsional eye movements [Insa].

Video-based combined pupil/corneal reflection systems can be used to provide the user's POR. One possibility is to fixate the head. Another alternative is to measure multiple ocular features in order to differentiate the eye and head movements. This can be done with the help of the so called *corneal reflection*. A light source, usually infrared, forms a reflection on the eye. When measuring this reflection relative to the location of the pupil center, head movements can be detected. While the difference changes when only the eye moves, it stays relatively stable when the head slightly moves. An example of a corneal reflection is shown in figure 3.10 [Duc07]. Corneal reflections are also known as *Purkinje Reflections*. Based on the construction of the eye, there exist four different *Purkinje Reflections* which can be seen in figure 3.11. In video-oculography the first

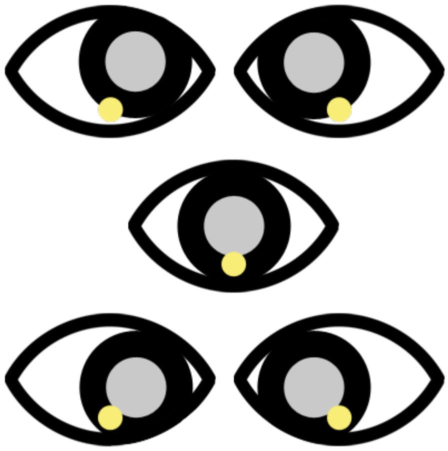


Figure 3.10: The difference of the corneal reflection (yellow) and the (center of) the pupil (gray) can be used to detect head movements.

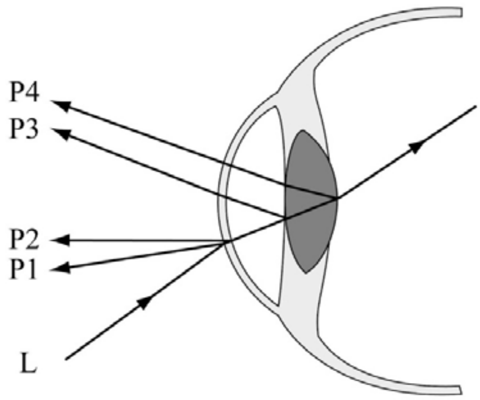


Figure 3.11: Based on the construction of the eye there exist four different so called *Purkinje Reflections*. The first image is normally used for video-oculography [Pop+12].

one is the one which is normally used. With the help of a calibration, the system is now capable of detecting the user's POR [Duc07; Pop+12].

3.3 Choice of Current Eye Tracking Devices

There exist a broad variety of eye tracking devices.Easily said, they can be either remote (not attached to the user) or wearable (for example glasses or other head mounted devices). An extensive list of different eye tracking systems, divided into different categories, is given by the *Communication by Gaze Interaction (COGAIN)* project[1]. In the following, a few eye tracking systems will be introduced, sorted by its system types: remote or wearable.

Wearable Eye Trackers

Arrington Research's[2] *SceneCamera* systems are video-based wearable devices. An example is depicted in figure 3.12. They are lightweight (the lightest weighs less than

Figure 3.12: The *SceneCamera* by Arrington Reasearch is a video-based, wearable device. Accuracies between 0.25 and 1.0 degrees are possible with this system [Resa].

25 g) and accuracies between 0.25 and 1.0 degrees are possible. The system provides monocular and binocular options and uses infrared (with a dark pupil). Here, the user is able to decide whether the tracking uses pupil measurements, the so called corneal reflection (for further information see 3.2 - corneal reflections) or a combination of both. To determine an accurate position of gaze, the system needs to be calibrated. The data can be stored, so a user only needs to calibrate once. Head movements are unlimited.

[1] `http://wiki.cogain.org/index.php/Eye_Trackers`
[2] `http://www.arringtonresearch.com/`

Arrington Research offers other devices for example remote/desk mounted ones and respective software to integrate into a device, as well [Resb].

The 60 Hz *Eye Tracking Glasses 2 Wireless (SMI ETG 2w)* introduced by SensoMotoric Instruments[3] is a wearable, lightweight (47 g), video-based eye tracking system. It supports real-time data access and can be controlled with a wireless connection. This way, among other things, it is possible to perform calibration and observe live gaze traces from a computer or tablet. It provides binocular tracking and works with contact lenses as well as most spectacles. A calibration is necessary to achieve a 0.5 degrees accuracy [Insb]. The glasses can be seen in figure 3.13.

Figure 3.13: The *Eye Tracking Glasses 2 Wireless (SMI ETG 2w)* introduced by SensoMotoric Instruments are wearable, lightweight, video-based eye tracking glasses. The tracker is able to achieve a 0.5 degrees accuracy [Insc].

The *EyeLink II* of SR Research[4] is a head mounted, infrared, video-based eye tracker. It is depicted in figure 3.14 [Ltd]. It provides the fastest data rate as well as the highest resolution of all head mounted, video-based eye trackers at the moment. Three miniature cameras are attached to a headband and no mirrors are used. The tracker is able to track both eyes (binocular eye tracking) with 250 or 500 Hz sampling rate and does so with an accuracy of 0.5 degrees. A calibration is obligatory to use the system. For tracking purposes, the system either uses the dark pupil or the white pupil technology

[3] http://www.eyetracking-glasses.com/
[4] http://www.eyelinkinfo.com/index.html

Figure 3.14: *EyeLink II*, introduced by SR Research, is an infrared video-based eye tracker. It works with an accuracy of 0.5 degrees [Ltd].

(with 250 Hz in combination with the corneal reflection). Additionally, a scene camera can be attached [Ltd].

The *Tobii Pro Glasses 2* introduced by Tobii Technology[5] is a wearable eye tracking system, especially used for real-world research. The natural, ultra-lightweight eye tracker with a weight of only 45 g is video-based and uses four cameras and a wide-angle HD scene camera. Data is captured with 50 or 100 Hz. The glasses have features like a persistent calibration and a minimum data loss during 'extreme' eye movements. The device tracks both eyes (binocular) and uses the dark pupil technology in combination with corneal reflection. Gyroscope and accelerometer sensors are included in the device [TAf]. The glasses can be seen in figure 3.15.

Remote Eye Trackers

EyeTech Digital Systems[6] distributes several eye tracking hardware and software. The video-based tracker*TM5 mini* described in [Sysa], is especially promoted to be a good

[5] http://www.tobii.com/
[6] http://www.eyetechds.com/

Figure 3.15: Tobii Technology developed the eye tracking glasses *Tobii Pro Glasses 2*. They are video-based and primarily used for real-world research [TAg].

addition to a speech device. Its dimensions are 29 x 3 x 2.5 cm and it weighs 0.23 kg. To detect the user's gaze, a calibration is necessary. Single or binocular tracking is used. Two infrared lights are integrated, so this dark pupil technology enables real-time eye tracking. Minor head motions can be detected and a 0.5 degree accuracy is possible. While the *TM5 mini* only works properly within a range of 40-75 cm [Sysa], the *VT2 XL* for example works within a range of up to 3 meters. The *VT2 XL* can be seen in figure 3.16 [Sysb]. Windows or Android are supported. There is also the possibility to just

Figure 3.16: The *VT2 XL*, developed by EyeTech Digital Systems, is a video-based tracker that works within a range of up to 3 meters [Sysb].

purchase software like, for instance, *QuickLINK*. Here, raw eye data can be streamed into an application [Sysc].

LC Technologies[7] offers several remote eye trackers. One of them is the *EyeFollower*, which is shown in figure 3.17 [Tecb]. Of all remote eye trackers in the world, this is (at

Figure 3.17: The *EyeFollower* is one of several remote eye trackers developed by LC Technologies. The typical average bias error is less than 0.4 degrees [Tecb].

the moment) the one tracker that is the most flexible in terms of head movements. The typical average bias error is less than 0.4 degrees and the sampling rate is 120 Hz. It is working on 90% of all users. Additionally, it tolerates most eyeglasses and contact lenses. It tracks both eyes with two separate cameras, is video-based and works with the bright pupil method. A special feature of this device is the fact that a monitor is already attached [Tecb; Tecc].

Tobii Technology[8] (see *Tobii Pro Glasses 2* in paragraph Wearable Eye Trackers, as well) also developed the so called *Tobii Pro X3-120*. The tracker is shown in figure 3.18 [TAe]. The device weighs 118 g and is an infrared- and video-based system. It works with binocular tracking and uses dark or bright pupil methods in combination with corneal reflection to track the user's gaze. The distance between the user and the tracker should be between 50 - 90 cm. The gaze accuracy is up to 0.4 degrees with a sampling rate of

[7] `http://www.eyegaze.com/`
[8] `http://www.tobii.com/`

Figure 3.18: The *Tobii Pro X3-120*, developed by Tobii Technology, is an infrared, video-based eye tracking system and achieves a gaze accuracy up to 0.4 degrees. Head movements are possible without falsifying the gaze data [TAe].

120 Hz. Head movements are possible without falsifying the gaze data[TAa].
Another device released by Tobii is the *Tobii Rex* eye tracker [TAb]. The distance between the tracker and the user should be between 40 and 90 cm, while for optimal tracking performance the distance should come to 65 cm. A so called trackbox is displayed. As long as the user's eyes remain in this box, the head can be moved. This tracker is positioned under the screen and works similar to the *Tobii Pro X3-120*. Here, different Software Development Kits (SDKs) with different features are available, one for Windows, Linux and Android as well as another one for Windows only [TAb].

Finally, the *EyeTribe*[9] is introduced. This low-cost eye tracker can be used by most persons that wear glasses and by all persons who wear normal soft contact lenses. The video-based tracker uses binocular gaze data and near-infrared illumination. The eye tracker with the dimensions 20 x 1.9 x 1.9 cm is currently the smallest one in the world and weighs 70 g. The tracker can be seen in figure 3.19 [Eyeh]. The accuracy for gaze measurement averages between 0.5 and 1.0 degrees after the user performs a calibration. The distance between user and tracker should maintain the range of 45 - 75 cm, whereas the distance for best results is 60 cm. The sampling rate can be set to 30 or 60 Hz [Eyeh].

[9] https://theeyetribe.com/

Figure 3.19: The *EyeTribe* tracker is a low-cost, video-based tracker. The accuracy for gaze measurement averages between 0.5 and 1.0 degrees [Eyeh].

3.4 Applications of Eye Tracking

Eye tracking applications can broadly be divided in two main categories: On the one hand *diagnostic applications* and on the other hand *interactive applications* [Duc07].
In times past, eye trackers have usually been used to study the human eye and, as a consequence, eye movements. After scientists investigated the human eye in general, especially understanding eye movements during reading has become a wide research area. In the early 1980s, with the spread of personal computers, human-computer interaction became more important (further information is given in section 3.1).
Today, there are a multitude of research areas in which eye trackers are used, such as for the purpose of marketing/advertising, usability studies, human factors and ergonomics [Duc07].

When the eye tracker gives objective and quantitative information related to the user's "visual and (overt) attentional processes" [Duc07], one talks of *diagnostic applications*. At this, eye movements are only recorded but do not cause visual effects for the user. This includes the assessment of web pages which is in wide use. Furthermore, the structure of the site or solely the placement of single objects, like for example advertisement banners, can be tested. For this purpose, the number of fixations (on a banner or another element) may be statistically evaluated [Duc07].

By comparison, a response or interaction is expected from an *interactive application* when it is used. These interactive applications can further be divided into two subcategories: *selective* and *gaze-contingent* applications. A user's gaze controls an interface with his gaze in selective applications, while in gaze-contingent applications the system is able to adapt its behavior when the user's visual attention changes [Maj+12].

Interactive applications can be a particularly important communication tool for disabled people who are not able to use a computer otherwise. For example, eye typing systems, that are numbered among selective interfaces, have been the first interactive applications. These were often slow or error-prone [Maj+12]. Today, eye typing speed is comparatively high and a wide variety of systems for this purpose exist (for example *EYE-S* or *ERICA*, mentioned in 2.1.1).

In greater depth, **gaze-contingent** systems can be divided into screen-based and model-based displays. Both methods are usually used to minimize bandwidth. For this purpose, screen-based displays manipulate image/pixel information, while model-based ones alter geometric objects. Both are, for instance, widely used in real-time Virtual Reality (VR) applications [Duc07]. An example for a screen-based display is given in figure 3.20, whereas a model-based display example can be seen in figure 3.21 [Duc07].

Figure 3.20: The image shows an example of a screen-based system. The images consists of a high resolution section at the viewer's gaze and low resolution parts at the remaining parts [DCM04].

The complete hierarchy of eye tracking applications is shown in figure 3.22.

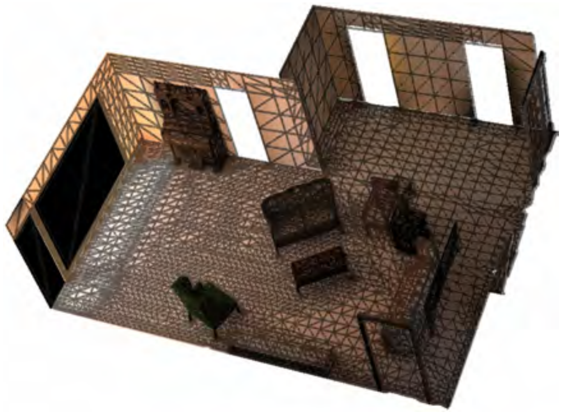


Figure 3.21: The image shows an example of a model-based system. The user is focusing the left room side. Consequently, the left section of the room is rendered with a higher level of spatial detail (triangles are more fine-grained) [DCM04].

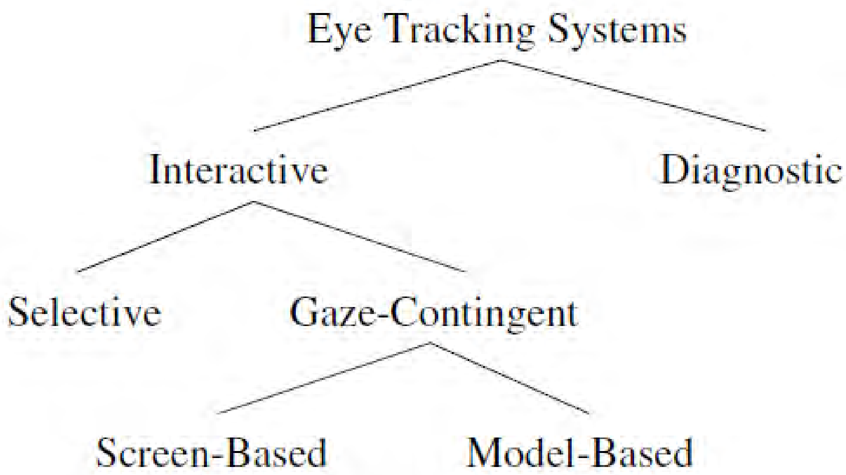


Figure 3.22: Hierarchy of eye tracking systems [Duc07].

3.5 Limitations of Gaze-Based Tracking

In the following, different limitations of (especially video-based) eye tracking are summarized and briefly explained.

Accuracy

Eye trackers do not have a 100% accuracy. Within video-based eye trackers, a variance of around 0,5 - 1 degrees is commonly accepted. Some eye trackers provide better or worse results. For example, the *SceneCamera* introduced by Arrington Research (see

3.3), is able to achieve an accuracy of 0.25 degrees [Resb]. However, it is not shown whether such results are only possible in a perfect environment.
The *EyeTribe* (see 3.3) provides an accuracy between 0.5 and 1.0 degrees, whereas the distance between tracker and user for best results is 60 cm [Eyeh]. This means, when having a 24 inches computer monitor with a display resolution of 1920x1080 px and a distance between the user and the tracker of 60 cm, the accuracy of the tracker is limited to ~19-38 px.

Accuracy must not be confused with precision where an eye tracker is able to reliably reproduce a measurement [Hol+11]. The difference is represented in figure 3.23.

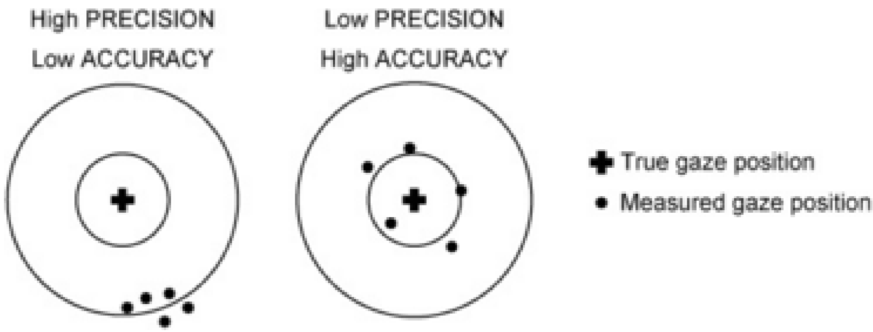


Figure 3.23: The difference between 'accuracy' and 'precision'. While the left image shows a system with high precision but low accuracy the right one displays a tracker with high accuracy but low precision [Hol+11]

.

Midas Touch

Jacob [Jac90] introduced the *Midas Touch* problem where the viewed objects are 'directly' selected. In this scenario, it is not possible to simply investigate the UI without constantly activating commands. Wherever the user looks at, they trigger a command. To avoid the *Midas Touch* problem, the user should be able to explore the interface without triggering actions and they should be able to interact with the interface on demand. Hence, a lot of different possible interaction methods for the selection of an object had been introduced and evaluated over the years. Some are listed in section 2.1.

Eye Tracker Latency

Gaze data is usually processed with a certain latency of the eye tracker. This latency depicts the time between capturing an image by the sensor and outputting gaze data to the network or an eye tracking application. More precisely the combination of camera exposure, transfer, calculation and delays in the system (network etc.) results in this value [TAc].

This delay differs from eye tracker to eye tracker. The *EyeTribe* (see 3.3 for further information) has a latency smaller than 20 ms (at 60Hz) [Eyeh]. Faster devices are for example the *Tobii Pro x3-120* described in 3.3 with a latency of 12 ms (at 120Hz) [TAa].

Characteristics of Eye Movements

The human vision can be divided into two parts: a very high resolution area called foveal vision and the remaining portion of the vision called peripheral vision [NP10]. The foveal vision only covers about two degrees. So, a person always has to combine multiple observations to get a sharp mental image. However, eyes do not move smoothly to achieve this but move in spurts and rests. The person itself is not aware of this because it happens so fast [NP10].

Resting on one spot is better known as a *fixation*. The eye rests on something typically around 200-600 ms. Between *fixations*, *saccades* occur. *Saccades* are sudden, ballistic and nearly instantaneous eye movements [Jac90]. *Saccades* only last between one-hundredth and one-tenth of a second and a person is effectively 'blind' during it. During a *fixation*, when the eye is holding still, a person is aware of the things they are looking at [NP10].

Even during *fixations* the eye is not steady but makes small jittery motions that are generally covering less than one degree [Jac90]. These motions are also called *oculomotor noise* and can be divided into *microsaccades*, *drifts* and *tremors*. While *tremors* are small movements, *drifts* are slow ones where the eyes move away from the center of fixation. *Microsaccades* take the eye quickly back to the center of fixation with a duration of 10-30 ms. In comparison, a drift lasts for 200 - 1000 ms [Hol+11].

The user is not aware of these jittery motions, so the interface should be 'ignoring' these movements as well. Smooth eye motions only occur in response to a moving object [Jac90].

Limitations Caused by the User

Conclusively, the eye tracker never knows for sure whether a user saw something conscientiously or why users are looking at something. Furthermore, not every person is able to use eye trackers. Some devices are not able to specify a user's gaze when they are wearing glasses or contact lenses. Besides, eye diseases like strabismus can cause problems, as well. Even very small eyes/pupils can cause problems with a few devices.

4 Implementation

The purpose of this work is to investigate the acceptance of eye trackers as an input device. The participants need to complete a predefined task to be able to evaluate the usage of the device afterwards. For this purpose, the completion of a questionnaire with the help of the eye tracker came into mind. Since a questionnaire can contain a lot of different elements (slider, text, radiobuttons and so on), a wide variety of different subtasks can be evaluated. Moreover, a questionnaire is well-known by the average human.

In section 4.1, the overall concept of the approach is outlined. Possible eye trackers are investigated and one was chosen for the purpose of this work in section 4.2. Following, an overview about the chosen tracker is given in section 4.3. The purpose of a calibration is further explained in section 4.4. Finally, the different steps of the study implementation itself are described more precisely in section 4.5.

4.1 Concept

First of all, the general idea is introduced in subsection 4.1.1. Several possibilities to interact with the UI shall be implemented. The choice which interaction methods will be implemented is specified in section 4.1.2. Conclusively, an overview of the UI design is given in subsection 4.1.3.

4.1.1 Idea

As already mentioned, the purpose of this work is to investigate the usage of eye trackers as an input device. More precisely, it shall be possible to complete a questionnaire with the help of an eye tracker, since the work with questionnaires is an important research area at the Institute of Databases and Information Systems at Ulm University, at the moment (like mentioned in [Sch+13] and [Sch+15c]). Furthermore, a lot of different subtasks can be evaluated since a questionnaire may contain a wide variety of elements like checkboxes or just plain text (a textfield) where the user has to perform different actions to interact with those elements.

A Windows application (Windows Presentation Foundation (WPF)[1]) will be implemented for this purpose. The first intent to implement a Windows Phone Application has been rejected due to problems with the integration of given eye tracker examples. The changed .NET environment for Windows applications caused problems in combination with the eye tracking examples integration.

After starting the application, a calibration has to be fulfilled by the user. Otherwise, the eye tracker can not be used. After successfully finishing the calibration process, the user will be forwarded to the following page where they are able to choose one of three different interaction methods (for further information see section 4.1.2). Every interaction method is briefly explained. Then, the user is able to fill out a questionnaire that shows up. The questionnaire has to include the most commonly used elements. After further investigations, these seem to be radiobuttons, checkboxes, buttons, textboxes and sliders. These elements are shown in figure 4.1. To reduce learning effects (in the following study), the elements' order is different in all three interaction methods. As soon

[1] "a graphical subsystem for rendering user interfaces in Windows-based applications by Microsoft "[Wpf]

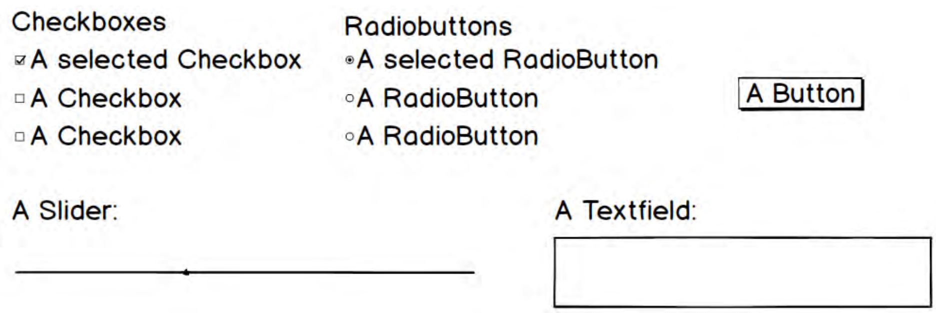


Figure 4.1: Sample of checkboxes (top left), radiobuttons (top middle), a button (top right), a slider (bottom left) as well as a textfield (bottom right).

as the user finished a questionnaire, they can return to the selection page to choose another interaction method.

4.1.2 Interaction Methods Choice

In the related work (see chapter 2) several gaze-based interaction methods, which had been used in other works, are summarized. These interaction methods can be divided into gaze-only methods and those where the gaze will be combined with a second input modality. In gaze-only methods, both pointing and target selecting will be done with nothing else than the eyes. Compared to this, the gaze can also be combined with a second input modality. So, pointing is achieved with the eyes while the selection of targets will be achieved with another input device like a keyboard. This work investigates both interaction methods. Thus, at least one interaction method must be allocated from each technique. All in all, not more than three techniques have been chosen due temporal restrictions in the study. In the following, possible methods are briefly introduced and three methods have been chosen.

Gaze-Only Interaction Methods

It is possible to interact with a Graphical User Interface (GUI) with gaze data only. Since *Dwell Time* (for further explanations see paragraph Dwell Time) is the most widely used technique, it appears natural to include this method.
As a second method, blinking came into mind. Blinking is natural behavior for a user and does not need to be learned. After an initial implementation, problems with blinking

became apparent. A person blinks every now and then, so the system has to differentiate between reflexive blinking and blinking on purpose. Several people are not able to wink, so winking did not seam to be an appropriate solution either. Repeated blinking or longer eye closures came into mind. Since this is still error-proneness compared to other methods, this approach was not pursued any further. In related work this method was not often considered.
With gaze gestures (introduced in section 2.1.1) that are complex enough, false unintended inputs can be avoided. This, however, means that the user has to learn these patterns first and the gestures can be time consuming. Compared to *Dwell Time*, no calibration is needed since the sequences depend on relative coordinates, not absolute ones. In this work, a calibration is necessary considering that the other implemented methods need an absolute point of gaze. Additionally, the need to learn the patterns first is too time-consuming for this purpose.
Another, not that well-known approach is to use selection regions or *Context Switching*. A selection region (i.e. a colored border, a dot or similar) appears as soon as an element that shall be chosen is looked at. Similar to this, the user has to switch areas (with their gaze point) to perform *Context Switching*. So, a two-step gesture has to be performed. Further explanation is given in paragraph Selection Region and Context Switching. There does not exist much data/information about this approach, since it is not often used. Compared to this, *Dwell Time*, the first chosen method, is a widely used technique. That is why *Context Switching* had been chosen, a rarely used interaction method.

Interaction Methods Combining Gaze with a Second Modality

Compared to gaze-only interaction methods, gaze can be used combined with a second input modality. Examples are the combination of the user's gaze and touch (further explanation is given in paragraph Touch) as well as the user's gaze and voice recognition (additional information is given in paragraph Voice). Touch can be used to manipulate elements without covering them with the hand. Voice recognition can be simplified since there is no need to describe an element's position verbally. All in all, both approaches seem promising. Due to missing hardware these approaches were not further pursued. Moreover, other devices can be used, such as the small touch-enabled device introduced in paragraph Other Devices. This is especially useful to interact with a distant display. As a distant display is not used in this work, there is no need for such an additional device. An often used technique is the combination of gaze data and key presses. Here, only a common keyboard is needed (for more details see paragraph Key Press). Computer users are used to work with a keyboard, consequently no practice is needed. Furthermore, this method is fast compared to many others. That is why this method is the third one implemented and included into the study of this work.

To sum it up, *Dwell Time*, selection region/*Context Switching* and gaze data combined with a keyboard will be implemented.

4.1.3 User Interface Design

The overall design of the GUI is illustrated in this section. To give an overview of the application, different mockups have been created, since mockups are used to specify the rough structure of the application [Sch+15b].

Like mentioned before, a calibration is needed to use the eye tracker as an input device. That is why the first page that shows up as soon as the application starts, is the *Calibration page*. Here, a trackbox is prominently shown in the middle of the site. The user's eyes are visualized in the trackbox and a colored circle appears in the middle of the box. It indicates how the user is positioned relative to the tracker. So, the color indicates whether the user can theoretically be tracked by the system. While the calibration process can be started when the trackbox is colored green, the user should relocate their position if the circle's color is yellow or red. The different stages are also illustrated in figure 4.2.

The calibration starts as soon as the *Calibrate* button is clicked. After the calibration process, the results are shown textually below the trackbox. If the calibration process was successful, the user can continue to the *selection page*. Otherwise the forward button will remain disabled. A mockup is shown in figure 4.3.

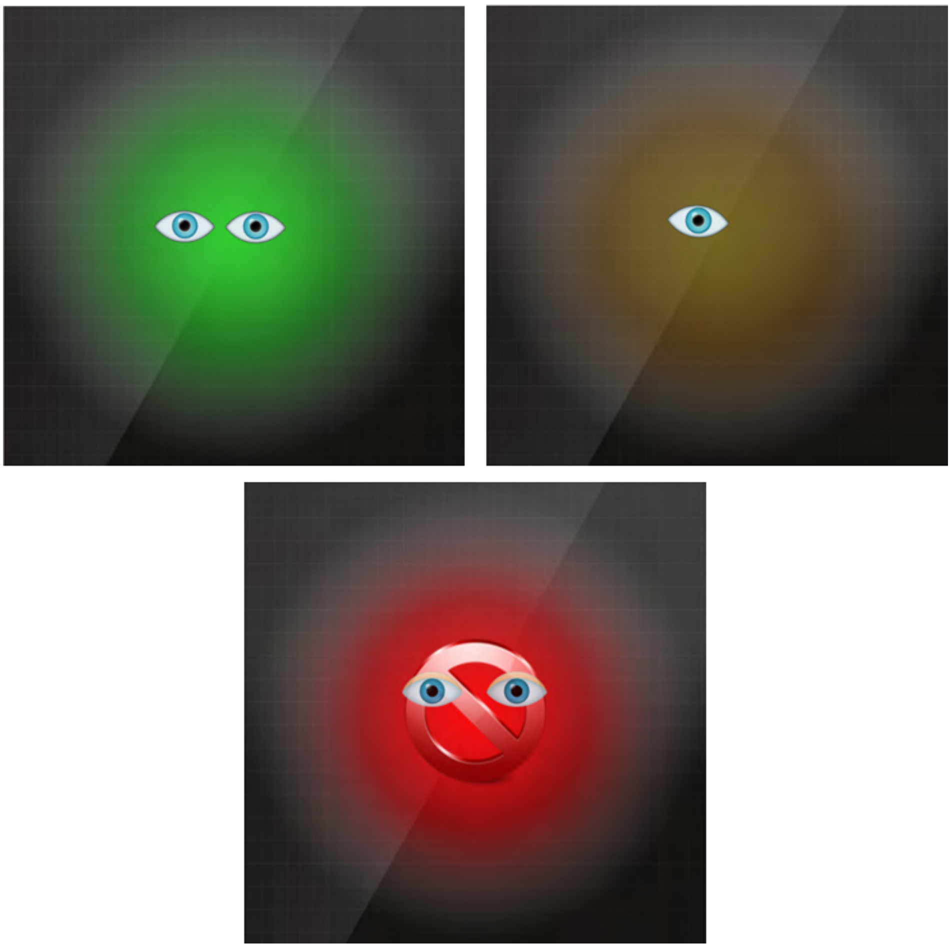


Figure 4.2: This trackbox is used to help the user locate in front of the eye tracking device. When the trackbox is colored green, the user is positioned correctly. Orange indicates that the eyes are not detected good enough by the tracker. When the trackbox is colored red, the user's eyes aren't detected at all. The tracker can only be used correctly when the trackbox is colored green (based on [Eyec]).

On the *Selection page* a list with the three possible interaction methods is displayed: *Context Switching*, *Dwell Time* and *Key Press* (for further information about the choice of interaction methods see subsection 4.1.2). As soon as clicking on one method, an explanation screen shows up. The user chooses one method by clicking the *Choose this method* button. Figure 4.4 depicts the mockup for this page.

According to the selection, a specific questionnaire shows up. Here, all elements are the same, no matter what interaction method had been chosen: a textfield, checkboxes,

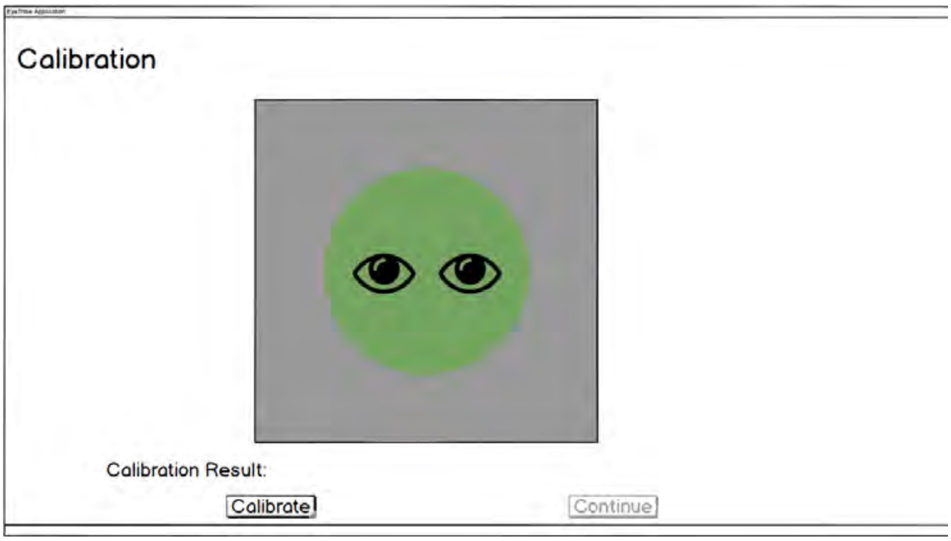


Figure 4.3: The *Calibration page* displays a trackbox to indicate the user's position relative to the tracker. The calibration starts after clicking the *Calibrate* button. As soon as the calibration has been successful, the user can forward to the next page.

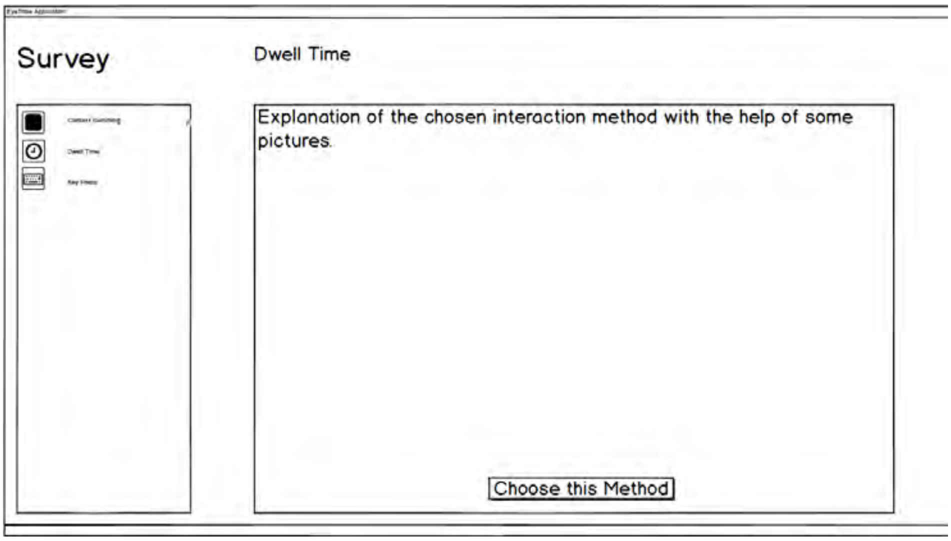


Figure 4.4: All three interaction methods are shown on the *Selection page*. As soon as choosing one method, an explanation is shown and the desired method can be chosen.

radiobuttons, a slider and a button. However, the elements' order differs. The elements, a proband should choose, are colored green. All remaining elements are colored red. This is important since the user should not think about 'the correct answer' but try to finish the questionnaire as fast as possible. Moreover, the trackbox mentioned before is shown in the upper right corner. This shall help the user to check whether they are still positioned correctly in the tracker's range. As additional help to interact with the questionnaire superiorly, the user's gaze is visualized with a small light blue dot. This shall not only help the user but also indicate whether the tracker is working properly since the user's gaze is rarely measured wrong, even with a successful calibration (for example when the user looks at the upper part of the screen but the gaze measurement indicates that they are looking at the lower part of the screen). This page is shown on the mockup in figure 4.5 with a blue dot to indicate the user's gaze position.

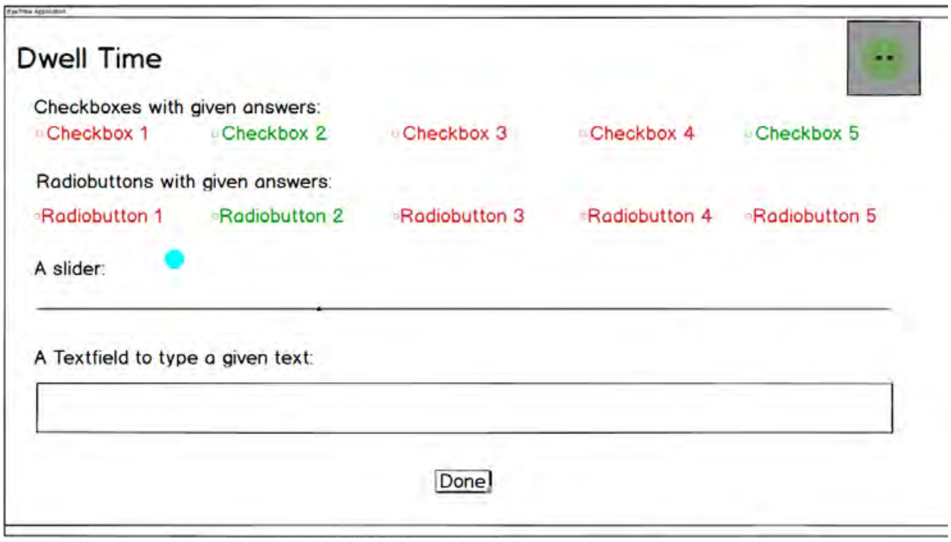


Figure 4.5: The order of the questionnaires' elements differ depending on the chosen interaction method. A small trackbox is shown in the upper right corner to support the user.

As soon as the user is done with the questionnaire, they can click the *Done* button and will be forwarded back to the calibration page. Re-calibration prior to executing each of the three interaction methods immediately is useful since the gaze measurements and, consequently, the study results shall not be affected by old and potentially obsolete calibration results (perhaps the user repositioned themselves in front of the tracker so the tracking results are not as valid as in the beginning). That is why the user has to recalibrate every time they start a new interaction method.

4.2 Eye Tracker Selection

A small subset of the multitude of existing eye tracking devices was already introduced in section 3.3. For this work, only low-cost devices were in line for the realization of the concept. Furthermore, only remote eye trackers (devices not attached to the user) came into consideration. They are often not as invasive as wearable alternatives. Today, video-based tracking systems are common, hence, a video-based tracker was chosen. The Tobii eye trackers and the *EyeTribe* had been shortlisted.

Tobii eye trackers are often used in similar studies. One possible tracking device is the Tobii Rex. The device can be used with two different SDKs: on the one hand, the Tobii Gaze Software Development Kit (SDK) and on the other hand, the Tobii EyeX SDK. The Gaze SDK is available for Windows, Linux and Android. A lot of configuration, calibration and data processing has to be done by the developer. That is why, according to Tobii Technology AB, this SDK should, in general, only be used if someone has control of the complete system, like for example in an embedded environment. In summary, this SDK is an "access library for Tobii eye trackers" [TAb]. Since the purpose of this work is not to optimize gaze data but to study the acceptance of eye trackers as input devices, the SDK was ineligible. By comparison, the EyeX SDK is an "OS extension for interaction using Tobii eye trackers" [TAb]. Here, a user works with the EyeX Engine Application Programming Interface (API), with which it is easier to start since it undertakes the calibration, screen setup and so on [TAb]. The EyeX SDK is available for C/C++ programming and as .Net version to be integrated into WPF and WinForms applications [TAd].
This SDK is only available for Windows [TAb]. Since the first purpose of the work has been to realize an Android application for questionnaires (for further details on the concept see section 4.1), this SDK could not be used with the originally concept.

The *EyeTribe* is a tracker developed by a danish startup company[2] [Eyeh]. This tracker, with a price of only 99$, is a low-cost tracker that comes with a full SDK. It is specified as a developer device since it does not contain any applications. The SDK is available for Windows, OS X and Android. Minimum system requirements are an USB 3.0 port

[2]The Eye Tribe Aps

and a strong processor (for example a core i5/i7 processor). C++, C# and Java are the programming languages that are compatible with the SDK. The SDK will only work with a few Android devices, like for example the Odroid-XU3 (Android 4.4.4) [Eyeh]. For this reason, this SDK was not usable for the development of an Android application.
That is why the originally planned concept had been changed and the application was then planned to be implemented for Windows.

Both, the *EyeTribe* as well as Tobii eye trackers could be used for the purpose of this work. Tobii eye trackers are frequently used in eye tracking studies. Whereas, eye tracking studies working with the *EyeTribe* are currently, compared to the Tobii systems, rare to find since the device is rather new. As a result, the *EyeTribe* was chosen to investigate an eye tracker as an input device for questionnaires.

4.3 Tracker Overview

To work with the *EyeTribe* tracker, a server is provided. This server has to run, otherwise the eye tracking won't work. It is responsible for device initializations, frameworks and so on and communicates with the SDK over an Open API. To get gaze data, client applications can communicate with the server with the help of this Open API. Hereby, JavaScript Object Notation (JSON) messages are exchanged asynchronously via Transmission Control Protocol (TCP) sockets. An overview of the architecture is illustrated in figure 4.6 [Eyee]. The SDK is available for C# , Java, C++ and Objective-C. Theoretically, every programming language that supports parsing JSON and opening TCP sockets is capable of connecting and interfacing with the tracker server. In summary, the SDK is a language-specific wrapper [Eyee]. With the SDK, developers can easily get started since the SDK undertakes the communication to the *EyeTribe* server through the API [Eyed]. It exchanges JSON formatted messages. The request message consists of three attributes, the query category, the actual request of the message and values (a JSON object or a JSON array of parameters). The query can be categorized in SDK tracker state and information related requests, calibration related requests and signaling heartbeats. The response message consists of a category (same as request category),

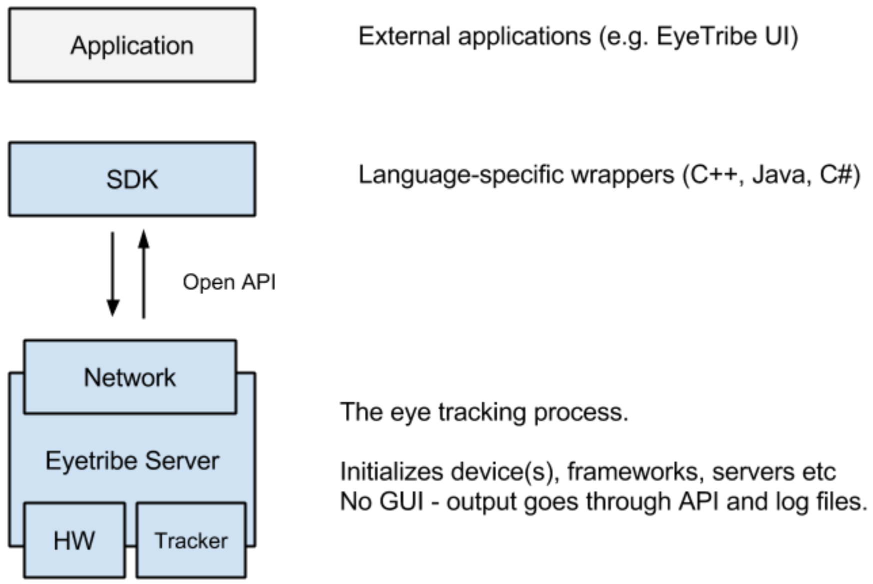


Figure 4.6: The *EyeTribe* Architecture [Eyee].

the type of request, a status code and values containing a JSON object of return values. An example of how to check whether the server is running and the system is calibrated is given in listing 4.1 [Eyea].

Listing 4.1: This JSON request is used to check whether the server is running and the system is calibrated [Eyea].

```
{
"category": "tracker",
"request": "get",
"values": [ "push", "iscalibrated" ]
}
```

A response for this request could be listing 4.2. For this work, the C# SDK was used, so the external application needs to be implemented. The SDK is .NET 3.5 compliant, so a windows application was implemented [Eyed].

Listing 4.2: On success the server responses (to the request message showed above) with this JSON message [Eyea].

```
{
  "category": "tracker",
  "request": "get",
  "statuscode" : "200"
  "values": {
    "push" : "true",
    "iscalibrated" : "true"
  }
}
```

4.4 Calibration

Like mentioned before, the use of the eye tracker requires a calibration. This is necessary since every person has different eye characteristics that have to be mapped to the display. Thereby, a circular target is shown in the middle of the screen. The user has to follow this target for approximately 20 seconds. Nine different calibration locations, a 3x3 grid, will be shown one-by-one on the screen [Eyeb]. Figure 4.7 shows the calibration pattern for a 3x3 grid.

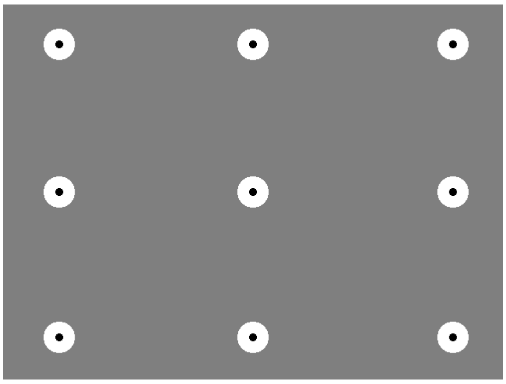


Figure 4.7: The calibration screen with nine targets, results in a 3x3 pattern [Eyeg].

During the calibration, the user's head must remain steady to achieve a good gaze tracking accuracy. After the calibration process, the user has to keep his position up and may not move the tracker. Otherwise a recalibration might become necessary. Afterwards a rating (see [Eyef]) is given:

- **perfect**: The calibration result is the best one that can be achieved. The accuracy is < 0.5 degrees.

- **good**: The accuracy is < 0.7 degrees and the calibration result is good enough for the purpose of eye tracking.
- **moderate**: The calibration process was acceptable but a recalibration is recommended since the accuracy is < 1 degrees.
- **poor**: this result is not good enough for eye tracking. A recalibration is necessary.

4.5 Study Implementation

For the upcoming study, a Windows application was implemented. Further detail of the implementation is given in this section. First, the overall program structure is briefly explained in paragraph Program Structure. Following, the correlation to the *EyeTribe* server is described in paragraph Server. Subsequently, the needed implementation to actually obtain gaze data (see paragraph Obtaining Gaze Data), and the procedure to calibrate the system is explicated (see paragraph Calibration). Finally, the algorithms to realize the different interaction methods *Dwell Time*, *Context Switching* and *Key Press* are introduced and explained in paragraph Interaction Methods Implementation.

Program Structure

The Windows application is implemented in Microsoft Visual Studio. The .NET Framework 3.5 is the used target framework due to EyeTribe SDK dependencies. When the application starts, a window is created within which a frame is used to integrate pages. First of all, it is checked whether the *EyeTribe* server is running in the background. Further information is given in the following paragraph Server. The first visible page for the user is the calibration page. Here, the user has to calibrate to make use of the eye tracker in the first place. After a successful calibration, the selection page can be opened where all interaction methods are explained and one can be chosen. Then, a questionnaire is shown that can be completed with the chosen interaction method. After finishing the questionnaire, the user will be redirected to the calibration page and the procedure starts over.

Server

To work with the *EyeTribe* tracker, the *EyeTribe* server has to run in the background. This can be done manually. Here, also a programmatic solution was implemented to avoid errors during the usage of the eye tracker. In so doing, it was checked whether the *EyeTribe* process was running and starts the server if necessary.
After starting the server, it can be configured manually to some degree by specified command line arguments or the change of the *EyeTribe* configuration file. Otherwise, default values are used to initialize the system. Parameters that could be changed are the socket port, the possibility to use a remote connection, the number of devices (only one can be used as tracking device) and the frame rate. The default frame rate is 30 frames per second (fps) [Eyee]. This had not been changed to 60 fps (which is available as well) since 30 fps are slower, but the allowed tracking box is larger. Consequently, larger head movements are possible. Moreover, the server was not configured at all since the default values are well suited for most users [Eyee].

Obtaining Gaze Data

To work with the eye tracker, the C# SDK [Eyed], introduced in section 4.3, was used. With the so called *IGazeListener*, the client can be connected and the class can be registered for events. An example is given in listings 4.3. When the *GazeManager* is activated (see listings 4.3 line 6), the API version and the client mode (push data continuously to the application or pull to fetch data by request) are set [Eyeg]. The *IGazeListener* interface has to be implemented by all classes that shall receive gaze data. The implemented method *OnGazeUpdate* contains, among other data, the on-screen gaze position and the size of the pupil. Only the gaze position is of interest here. To actually output gaze data, the eye tracker server must be calibrated [Eyeg]. The calibration is examined more closely in the following paragraph. At the end, the client has to be disconnected, like shown in listings 4.4 [Eyeg].

Calibration

A calibration needs to be performed to get gaze data. For this purpose, the user has to follow a circular target that is displayed on the screen. Nine positions (a 3x3 grid) have to

Listing 4.3: The client connection is acitvated to receive gaze data [Eyeg]. Gaze data is shown in the *OnGazeUpdate* method.

```
public class GazePoint : IGazeListener
{
  public GazePoint()
  {
    // Connect client
    GazeManager.Instance.Activate(GazeManager.ApiVersion.VERSION_1_0, GazeManager.
        ClientMode.Push);

    // Register this class for events
    GazeManager.Instance.AddGazeListener(this);
  }

  public void OnGazeUpdate(GazeData gazeData)
  {
    double gX = gazeData.SmoothedCoordinates.X;
    double gY = gazeData.SmoothedCoordinates.Y;

    // More Code
  }
}
```

Listing 4.4: Disconnection of the client [Eyeg]. No gaze data will be received, anymore.

```
  // Disconnect client
  GazeManager.Instance.Deactivate();
```

be sampled. At each point the circle remains for around one second where the system collects samples. When the calibration process proceeded successfully, on-screen x and y coordinates are produced. Further information about the calibration process is also given in section 4.4. To implement the calibration, the dynamic link library (dll) *TETWinControls.Calibration*, from the C# samples section on Github[3] can be used. A code example is given under listings 4.5.

As soon as the *Calibrate* button is clicked, the *CalibrationRunner* handles the entire process. The calibration will be performed in a new window that closes after completion. The method *calRunner_OnResult* can be used to investigate the calibration result [Eyeg]. In this work, this section is used to investigate whether the user can proceed or whether they have to redo the calibration process (in case the result had not been good enough to provide reliable eye tracking data).

[3] `https://github.com/EyeTribe/tet-csharp-samples`

Listing 4.5: Integration of the Calibration Process with the help of the *CalibrationRunner*. To examine the results, the method *calRunner_OnResult* can be used [Eyeg].

```
using TETControls.Calibration;

public class MyApplication
{
  private void ButtonCalibrateClicked()
  {
    CalibrationRunner calRunner = new CalibrationRunner();
    calRunner.OnResult += calRunner_OnResult;
    calRunner.Start();
  }

  private void calRunner_OnResult(object sender, CalibrationRunnerEventArgs e)
  {
    switch (e.Result)
    { //do something with the results }
  }
}
```

Gaze Coordinates

After finishing the calibration the system provides the user's gaze coordinates. Here, a point on the screen is defined where the user is currently looking at. At this, the x-position is defined to be left-oriented in a 2D coordinate system while the y-position is defined to be top-oriented. This is visualized in figure 4.8 [Eyee]. Thereof, both raw and smoothed coordinates exist [Eyee]. In this work, smoothed coordinates have been used to improve tracking results. In so doing, the gaze point remains more steady.

Trackbox

As mentioned earlier in section 4.1.3, the trackbox is used to position the user correctly in front of the tracker. In other words, the user's position is illustrated relative to the sensor. Therefor, the *TETControls.Trackbox* is used. The *TrackingStatusGrid* control is used as visual component [Eyeg]. Furthermore, a new *TrackBoxStatus* is added to the grid. The trackbox is displayed in figure 4.9.

Interaction Methods Implementation

Context Switching, *Dwell Time* and *Key Press* have been chosen as interaction methods (like explained in section 4.1.2).

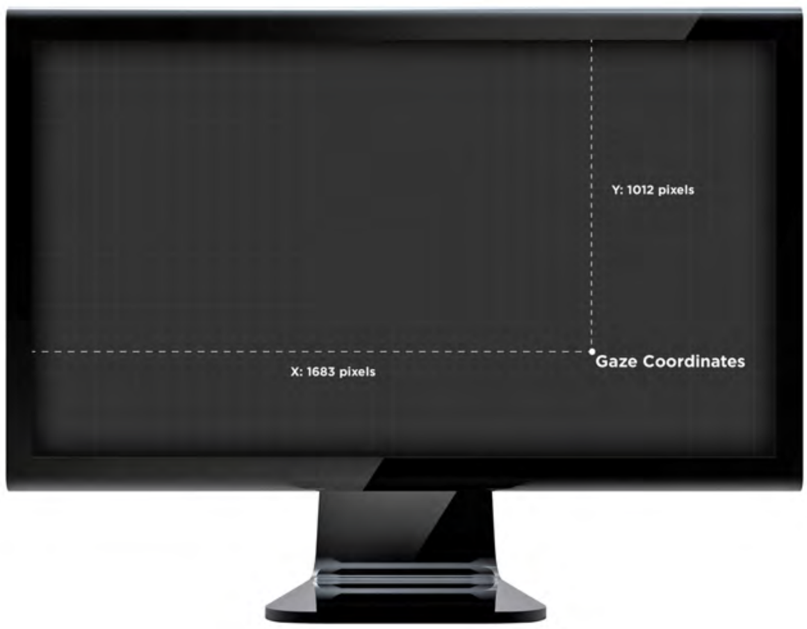


Figure 4.8: Gaze coordinates in pixels. The x-position is defined to be left-oriented in a 2D coordinate system while the y-position is defined to be top-oriented [Eyee].

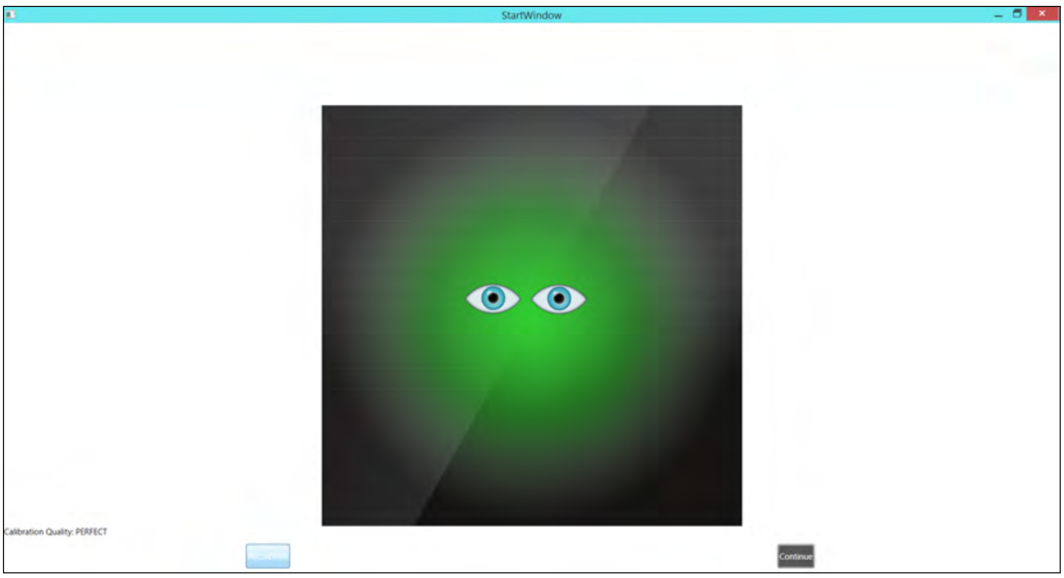


Figure 4.9: When starting the application, a trackbox is shown. The user is able to positiom themself in front of the tracker more easily because of the visual feedback (color of the trackbox) (trackbox based on [Eyec]).

Although the smoothed, not the raw gaze coordinates have been used in the implementation, it became noticeable that it can still be hard to perform interactions where the user has to fixate an element for a specific time. This issue might be the result of potentially unsteady detection of the eye's position. So, if a point at the border of an target has been fixated, this can cause problems. That is why the focus of an element does not change immediately in all three interaction methods. A duration of 100 ms has to pass until the focus changes. Whether an element has focus is visually indicated by a border appearing around the target. This can be seen in figure 4.10.

Figure 4.10: The focus of a control element is visualized with a colored border (here, a dark green one).

Loosing the focus is conform to the same principle in the *Dwell Time* and the *Context Switching* scenario. A user has to look away from a control element (to another section on the screen where no control elements are located at) for at least 100 ms to lose focus on the control element. That is different to the *Key Press*. Here, the fixation remains unchanged even if the user fixates a part of the screen with no control elements, or if their point of regard is registered to be outside of the screen. This is important since the viewer's glance could wander to the keyboard minimally or completely. The user should be able to look at the keyboard (for example to find a key) without losing focus (especially when typing in the textbox).

To detect which control element is fixated, each control element is assigned to a defined rectangle. If the point of gaze intersects a control element's rectangle, the user is looking at this specific element. An example screen with marked rectangle regions is illustrated in figure 4.11 and 4.12.

In *Dwell Time* and *Context Switching*, typing text is performed with nothing else than the user's gaze. As soon as a textbox is focused, an on-screen keyboard appears. This keyboard contains all characters from A-Z (in qwerty layout). Moreover, space, comma and period are available. Lower case is not available, since only the overall interaction concept is of interest in this scenario. All characters are capitalized. Back is performed with the *Back* button while the keyboard can be closed with the *ESC* button. All keys

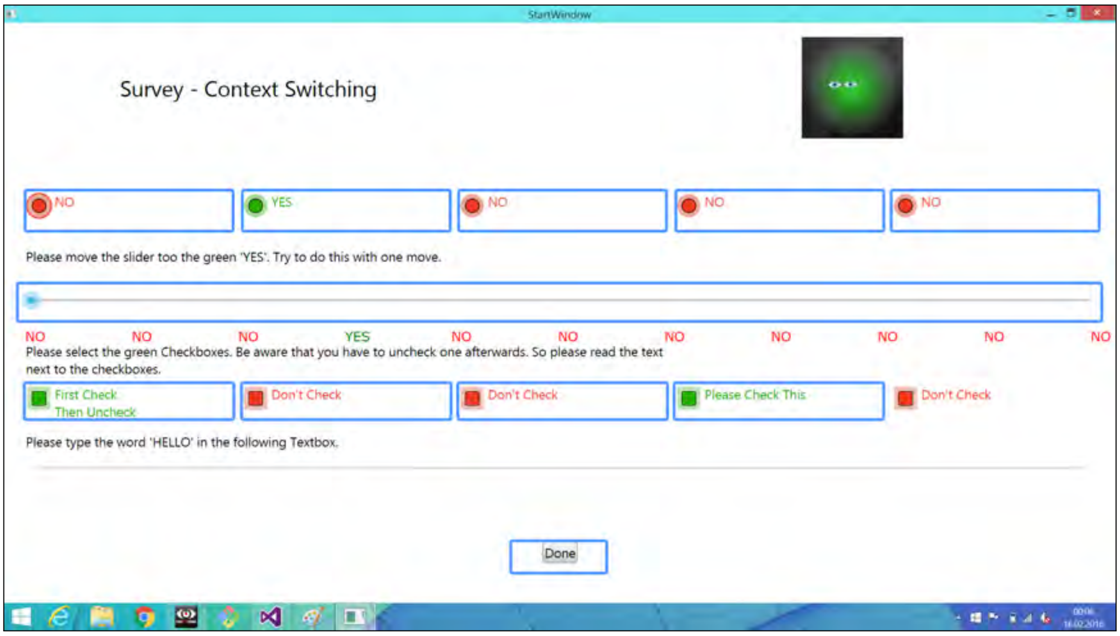


Figure 4.11: Visualization of the rectangles that are used to calculate whether the user is focusing an element. Here, all elements besides the elements on the on-screen keyboard are shown.

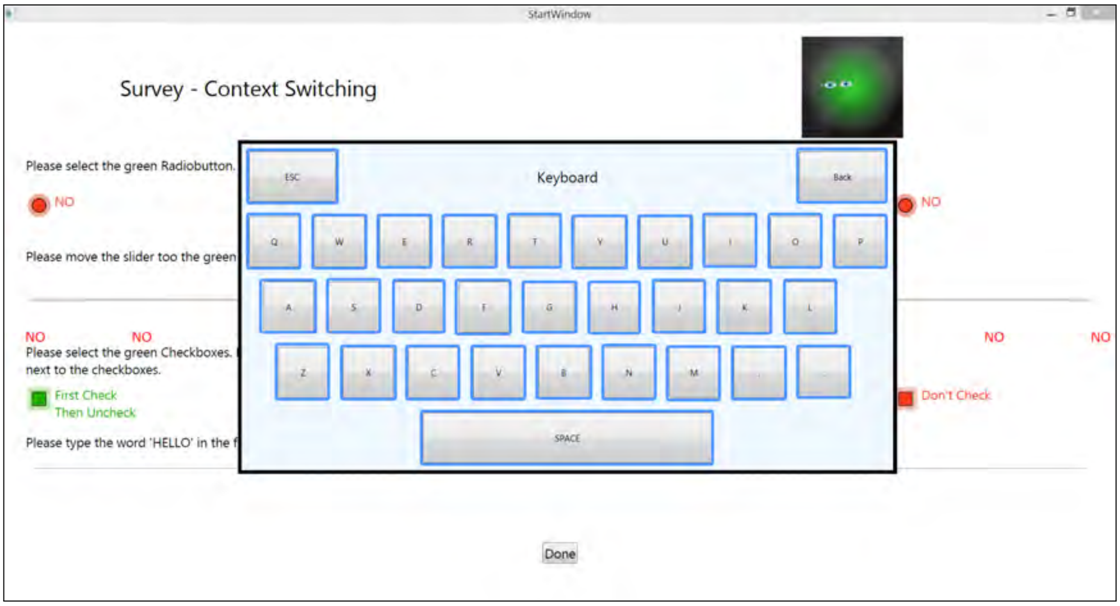


Figure 4.12: Visualization of the rectangles that are used to calculate whether the user is focusing an element. Here, all elements on the on-screen keyboard are shown.

can be used identically to the other controls in *Dwell Time* and *Context Switching*. In *Key Press*, the physical keyboard is used to type. An example is shown in figure 4.13.

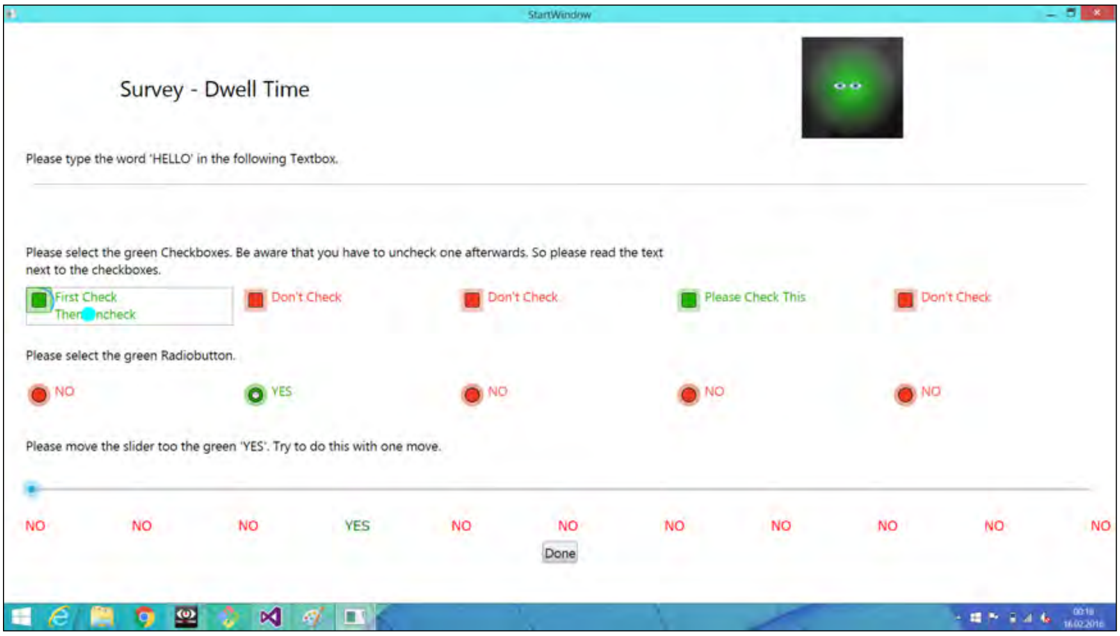


Figure 4.13: The questionnaire is shown, in this specific case, *Dwell Time* would be used. The order of the elements varies per interaction method.

Dwell Time

A duration of 750 ms was chosen for the *Dwell Time*. According to Tula and Morimoto [TM14] a useful duration for *Dwell Time* lays between 600 ms and 1000 ms. Shorter time durations can be used for advanced users. It was assumed that the study would be carried out by novice users as well, so the *Dwell Time* should be at least 600 ms. Since it can be challenging for users to fixate one specific area for a long time, the *Dwell Time* should not be too long. Besides, the overall needed interaction time increases with long time durations (if it is seen separately from wrong inputs). A pseudocode to realize *Dwell Time* is depicted in algorithm 1.

The *Dwell Time* is visualized via the animation of a blue circle that is closing clockwise. This can be seen in figure 4.14.

Algorithm 1 *Dwell Time* algorithm - on the example of a TextBox

Require:

$gazePoint$	// *user's point of view*
$controlElementRectangle$	// *control elements area and location*
timer $focusTimer$	// *time duration until the control element gets focus*
timer $dwellDoneTimer$	// *checks whether given dwell time is done*
timer $animationTimer$	// *to animate the dwell time*
$controlElementFocus$	// *control element that has been looked at lastly*
$controlElement$	// *control element that has focus*

procedure dwellTime ($gazePoint, controlElementRectangle, focusTimer, dwellDoneTimer, animationTimer, controlElementFocus, controlElement$)

```
if eyes are on screen then
    if controlEelementRectangle contains gazePoint then
        if control element has no focus then
            /* focus timer started on different control element */
            if focusTimer is enabled AND controlElementFocus is not current control element then
                stop focusTimer
                stop dwellDoneTimer
                stop animationTimer
                hide animation
            end if
            /* focus timer is stopped without finishing focus time */
            if focusTimer is not enabled AND focus duration is not finished then
                /* element the user looks at */
                controlElementFocus ← current control element
                start focusTimer
            end if
            /* focus time duration is finished correctly */
            if focus duration is finished AND controlElementFocus is current control element then
                give current control element focus
                start dwellDoneTimer
                start animationTimer
                /* element where the dwell time started */
                controlElement ← current control element
                start animation
            /* focus time duration is finished but on the wrong control element */
            else if focus duration is finished AND controlElementFocus is not current control element then
                stop focusTimer
            end if
```

Algorithm 1 *Dwell Time algorithm* - on the example of a TextBox (continued)

```
29:         else                                // element has focus
30:             /* focus was set in another control element */
31:             if controlElementFocus is not current control element then
32:                 remove focus from control element
33:                 stop dwellDoneTimer
34:                 stop animationTimer
35:                 hide animation
36:             end if
37:             /* focus duration is done and control element is correct */
38:             if dwell time duration os finished AND controlElement is current control
                element then
39:                 stop dwellDoneTimer
40:                 stop animationTimer
41:                 hide animation
42:                 /* open keyboard since the control element is a TextBox */
43:                 open keyboard
44:             /* dwell time duration is finished but on the wrong control element */
45:             else if focus duration is finished AND controlElementFocus is not current
                control element then
46:                 remove focus from control element
47:                 stop dwellDoneTimer
48:                 stop animationTimer
49:                 hide animation
50:             end if
51:         end if
52:     else
53:         /* focusTimer duration checking analog to lines 4-17 above */
54:         /* followed by: */
55:         if focus duration is finished AND controlElementFocus is 'else' then
56:             remove focus from every control element
57:             stop dwellDoneTimer
58:             stop animationTimer
59:             hide animation
60:             /* if a dwell timer starts –> it was not on a control element */
61:             controlElement ← 'else'
62:         /* focus time duration is finished but on another control element */
63:         else if focus duration is finished AND controlElementFocus is not 'else' then
64:             stop focusTimer
65:             remove focus from every control element
66:         end if
67:     end if
68: end if
end procedure
```

Figure 4.14: The concept of the *Dwell Time*. As soon as an element received focus, a blue circle indicates the 750 ms time duration. Here, the checkbox will be unchecked when the circle is completely shown.

Context Switching

In *Context Switching*, a blue rectangle appears as soon as the user fixates a control element (after this element receives focus). Normally all rectangles appear above the control elements or keyboard keys. Due to space problems on the on-screen keyboard, it can also show up next to a key. To select an element a two-step gesture has to be performed: from the control element that should be selected, to the appearing blue rectangle back to the control element. The rectangle appears for 3000 ms. This time is much higher compared to the *Dwell Time* since it seemed to be difficult to select the appearing rectangle quickly without accidentally selecting another control element. The same time span was used as a limit to go back to the control element and focus it again. Compared to the *Dwell Time* algorithm introduced in 4.5, an additional timer was used to achieve the two-step gesture. Apart from that, the algorithm is implemented analog. The concept of *Context Switching* is illustrated in figure 4.15. When the time passes and the

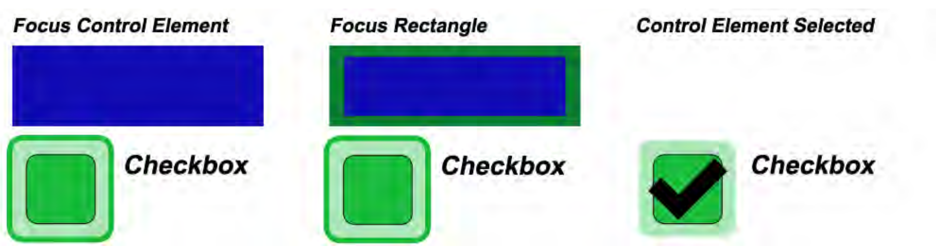


Figure 4.15: The concept of *Context Switching*. A blue rectangle appears as soon as a control element receives focus. The fixation of the rectangle is indicated by a green border. After fixating the control element (within another 3000 ms) the checkbox will be checked.

rectangle or the control element was not fixated, the rectangle simply disappears and no selection will be triggered. The same happens when focusing another control element (a new rectangle appears above this element and the first shown rectangle disappears).

Key Press

The implementation of the *Key Press* is the simplest one. Focusing an element behaves analog to the other two approaches. So, the user has to fixate an element for 100 ms until it receives focus. The difference, as mentioned before, is that the element keeps its focus until another element is focused. That way, the gaze can be lowered to the physical keyboard while pressing a key/typing. Otherwise, the user would be obliged to always fixate the target while typing. To select an element as soon as it receives focus, the space key on the physical keyboard has to be pressed. A de-selection (for example unchecking a checkbox) works analogously. Typing is realized by simply typing on the physical keyboard as soon as the textbox is focused.

5 Study

The purpose of the study was to investigate the assessment of eye tracking as an input device. Moreover, the overall acceptance of the implemented interaction methods was researched. Additionally, it shall be tested whether it makes sense to use eye tracking as input device for questionnaires.
First of all, the preparation of the study is introduced in section 5.1. Following, the study conduct is explained in more detail in section 5.2.

5.1 Study Preparation

The study took place at the University of Ulm. In the past, some research about eye tracking already took place here. Thus, it is expected that novice and experienced eye tracking users participate at the study. Three different input methods with eye tracking have been implemented. So, each participant can test one method or all three methods. Since it is a fundamental research question which method would be preferred by the users, it seems logical that all three input methodologies are tested from each participant. In order to prevent learning effects during the study, the order of the input methods differed each round. For example, one participant starts with *Dwell Time*, followed by *Context Switching* and finally *Key Press*, while another one starts with *Key Press* continues with *Dwell Time* and finishes with *Context Switching*. Consequently, learning effects can be prevented. Additionally, the questionnaire elements are ordered differently in each input method. The study took place over a period of three days in November 2015. Twenty to forty minutes have been foreseen for one study round. Most participants needed more trials to come by a successful calibration result, hence, the assessed time was fully used and some probands even needed longer than initially thought. The study took place in a room of the university. Since the eye tracker needs to be located below the screen, working with a notebook's display was not an option. An additional second display (24 inches) and a peripheral keyboard were connected to the laptop. In doing so, it was possible to position the eye tracker below the screen and even adjust the screen height according to the specific needs of the participant. The study is only undertaken with one subject at a time. This is necessary since the disturbances should be kept to a minimum. The test set-up can be seen in figure 5.1.

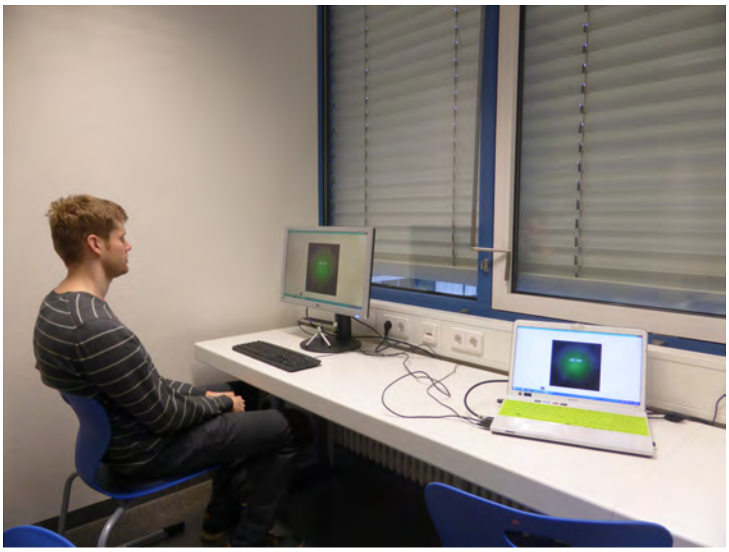


Figure 5.1: Here, the study set-up can be seen. A participant is currently conducting the study.

5.1.1 Choice of the Study Technique

Providing questionnaires, requesting interviews and recording videos is only part of the available study methods. In order to chose one, the different advantages and disadvantages of several techniques have been examined.

One advantage of a questionnaire is that no support by the study observer is given. Thus, the possibility to save time is given (multiple probands are able to complete a questionnaire at the same time). This was not needed here due to the fact that only one participant is participating in the study at a given time. Compared to an interview, another advantage of a questionnaire is the fact that the study observer does not accidentally influence the proband [RP08]. A disadvantage is the limitation of coverage. Questions are limited and can not be extended if needed. In an interview, single aspects can be extended or added easily. Furthermore, dependencies can be simply conceived. All in all, the questioning is controllable at any time. However, the time exposure such an interview requires is disadvantageous [RP08]. For these matters, an interview or an combination of a questionnaire with an interview was excluded. In particular, the simple, unintended manipulation of the participant by the interviewer was the main reason to dismiss an interview as a possible study technique. A questionnaire will be used to get enough information from the probands. Here, open and closed questions can be used. A combination of both is possible as well. When it comes to open questions, the questionee can phrase a text to the given question or statement. On the other hand, given answers are already formulated when it comes to closed questions. This is especially helpful if exact information shall be retrieved [RP08]. In this study, a combination of both, open and closed questions was used to cover own estimations and exact data.

The question arises as to whether a questionnaire/s would provide sufficient data. There was a risk that only subjective information will be covered. That is why the users' interactions with the eye tracker have been logged. More precisely, the users' calibration results and all of their clicks have been logged with a timestamp. This data can be used to get objective information about the participants' actions. Here, a video recording could be an option as well. Since a log seems to be adequate in this case and the data is easier to evaluate, simple logging was chosen. The logfile listed all calibration results and gaze-based clicks that were performed by the user, marked with a timestamp. Since the order of clicks a participant should perform is given, faulty input can be detected. The time needed to perform a gaze based click can be determined, as well. An advantage of recording the actions of a proband on video is that they can be asked later to explain their actions. Since the time was limited, this was not an option in this scenario.

5.1.2 Study Materials

For this study, several materials were needed. First of all an informed consent was needed. Following, the creation of questionnaires was necessary to evaluate the subjective experiences of the participants later. A paper-based questionnaire was used even if this has several drawbacks like the possibility of errors when copying the data to electronically analyze it [Sch+14]. In this way, a following participant can start to work with the eye tracker even if the previous proband is still answering the questionnaire. Furthermore, a logfile was created for each proband to capture additional objective information.

Informed Consent
Each participant needs to sign an informed consent form that provides more details about the study requirements, confidentiality and so on. More precisely, the research topic, the procedure of the study and more were explained in detail. Moreover, the possible benefit of the study was further declared. Conclusively, the confidentiality of the data was explained. Here, the data will be processed in a pseudonymized form. This is necessary since it is possible to draw conclusions to a specific person with the given time slot of this person.

Questionnaires
In addition, a demographic questionnaire was created. Here, general questions relevant for this study have been inquired. The age, sex and study major/profession were asked. The question about eye diseases and whether the participants wear glasses/contact lenses was especially relevant for the work with an eye tracker. Furthermore, questions

about earlier eye tracking usage were posed. The complete questionnaire can be seen in appendix 1.
Finally, a questionnaire about the usage of this eye tracker and the different implemented input methods was created. This questionnaire can be seen in appendix 2. Most questions make use of checkboxes since the users were able to check more than one answer. Besides, asking closed questions simplifies the subsequent study evaluation. Additionally, an *other* field for own thoughts was included. First of all, questions about the usage of the eye tracker in general had been asked. For example, what a participant likes/dislikes about the eye tracker/eye tracking experience and the calibration process have been asked. Furthermore, detailed information about the given input methods have been posed. What the participants liked/disliked about the specific methods has been investigated as well as the evaluation of the questionnaire's elements. Also, input specific questions like the duration of the *Dwell Time* have been investigated. At the end, each user had the possibility to give additional negative, positive or neutral comments. The different documents were tied to each other by the help of a participant number that each participant received.

Data Logging

Besides subjective information, data about the duration of different interactions are of interest. That is why, a logfile was created for each study participant. At the beginning, the participant needed to perform a calibration. The calibration results were integrated in the logfile. Consequently, it can be determined how many calibration iterations were necessary to achieve a valid calibration result. The duration of the different interactions and the element selection play an important part, as well. With this data, the three interaction methods *Dwell Time*, *Context Switching* and *Key Press* can not only be investigated upon a subject evaluation by the user, but also upon facts. The overall time that is needed to perform the task (complete a questionnaire) with each interaction method can be compared. The time needed to perform an activation/selection of different elements (for example checking a checkbox) can also be calculated with the help of the logfile. As soon as the participant selects an input method, the questionnaire opens and a timestamp is written in the logfile. As soon as the user clicks an element, the element description and the occurring timestamp is recorded. The time difference between the elements' timestamps represent the time the user needed to fulfill the click on this element. This is possible since each subject had the possibility to practice the completion of the questionnaire earlier and all elements that have to be selected were marked (when the participant has to check the second checkbox, it is highlighted green, the others are highlighted red, like mentioned in subsection 4.1.3). Furthermore, the order of the input methods is different for each participant. Since the participants may not be disturbed during the completion of the questionnaire and they were asked to complete it as fast as possible, time dependencies are negligible. Additionally, the amount of incorrect entries and the associated loss of time can be calculated.

5.2 Study Conduct

Ahead of the study, each participant choses one time slot. Like mentioned before, twenty to forty minutes have been foreseen for one study round. Since most probands needed more trials to come by a successful calibration result, they were busy for 35 minutes at an average to complete the work with the eye tracker. Due to the fact that they had to complete the questionnaires as well, some needed longer than originally thought. At the beginning, each participant had to read and sign the informed consent to attend the study. After that, they completed the demographic questionnaire. These two documents have been completed by the participants previous to the work with the eye tracker. Following, the proband is seated in front of the eye tracker and the display. As soon as the participant is sitting comfortably, the eye tracker, the seat and the display were adjusted correctly. The study was explained in more detail, even if it has been described in the informed consent as well, so that each proband had for sure the same level of knowledge. Moreover, tips for the calibration and the overall tracker usage were given: the head should remain as steady as possible as soon as the system is calibrated; The user should not blink (too much) during the calibration process. Subsequently, the participant started with the first calibration round. Afterwards, an interaction technique was chosen and specific information for this technique was given, written and verbally. Now, the participant began to use their gaze to complete a questionnaire. After a training round of approximately one minute, the calibration was repeated. The interaction input was used again to gain relevant information for the study results. This procedure followed the same pattern for the remaining two input techniques. During the test iteration, tips were given, if necessary (for example what a user could do, if they are not able to click a specific element). In the iteration round, relevant for the study results, no further tips were given. Questions were allowed during the study at any time. After finishing all three input methods, the proband had to complete the second questionnaire. Here, subjective evaluations about, for example, the eye tracker in general and the specific input methods were inquired.

6 Study Analysis

This section illustrates the analysis of the undertaken study results. First, general information about the study participants is given in section 6.1. Subsequently, the overall tracking experience is classified in section 6.2. Here, among other research questions, the calibration process and the suitability for daily use of such an eye tracking system has been evaluated. The comparison of the three different interaction methods is the focus of section 6.3. The characteristics of each interaction method (for example the size of the rectangles used in *Context Switching*) have been evaluated in section 6.4. The purpose of completing a questionnaire with this system has been evaluated in section 6.5. Concluding, a summary of the study results is given in section 6.6.

6.1 General Information

All in all, eighteen probands participated in this study. For three of them, (parts of) the tracking did not work properly. One participant was not able to fulfill the calibration with a satisfying result since they blinked too often. One participant slightly squints, so the tracker was not able to discern their eyes correctly (this was realized since the illustration of the user's eyes were flickering all the time). The third proband was, as well, not able to complete the calibration process successfully. Even with an intense observation of the user's actions, the reason was not found. The user was wearing contact lenses, so this could be the reason. Furthermore, one participant was late, so time was too short to complete the study. That is why this participant was left out. For this reason, just the study results of fourteen participants were taken into account in the evaluation. Unless otherwise stated, only the fourteen remaining participants are considered in the following study analysis.

The participants were between 23 and 34 years of age. Nine of them were studying Computer Science or a comparable subject, while four were working in the field of Computer Science. Only one person was studying Economics, a non-related subject. Five probands already worked with an eye tracker before (mostly a Tobii product), some of them often, others only once. Nine participants never worked with one before. Most of the probands were male (twelve out of fourteen) due to the fact that a lot of participants work/study in a computer related field in which women are often under-represented.

6.2 Overall Experience

In the second questionnaire, which was completed at the end of the study, the participants were able to subjectively rate among others the usage of the eye tracker and the interaction methods. First of all, the participants evaluated the eye tracking usage in general. The majority assessed the general tracking usage as intuitive (nine out of fourteen probands). Even more (eleven out of fourteen users) found that the handling of the eye tracker can be learned quickly. Besides, two participants liked the short interaction method compared to the mouse as input device and one appreciated the good overall tracking (for example a good recognition of the gaze point). All aspects are represented in diagram 6.1.

Nevertheless, the negative aspects outweigh the positive ones, since the amount of negative ratings was higher. Nearly all participants, thirteen out of fourteen, perceived

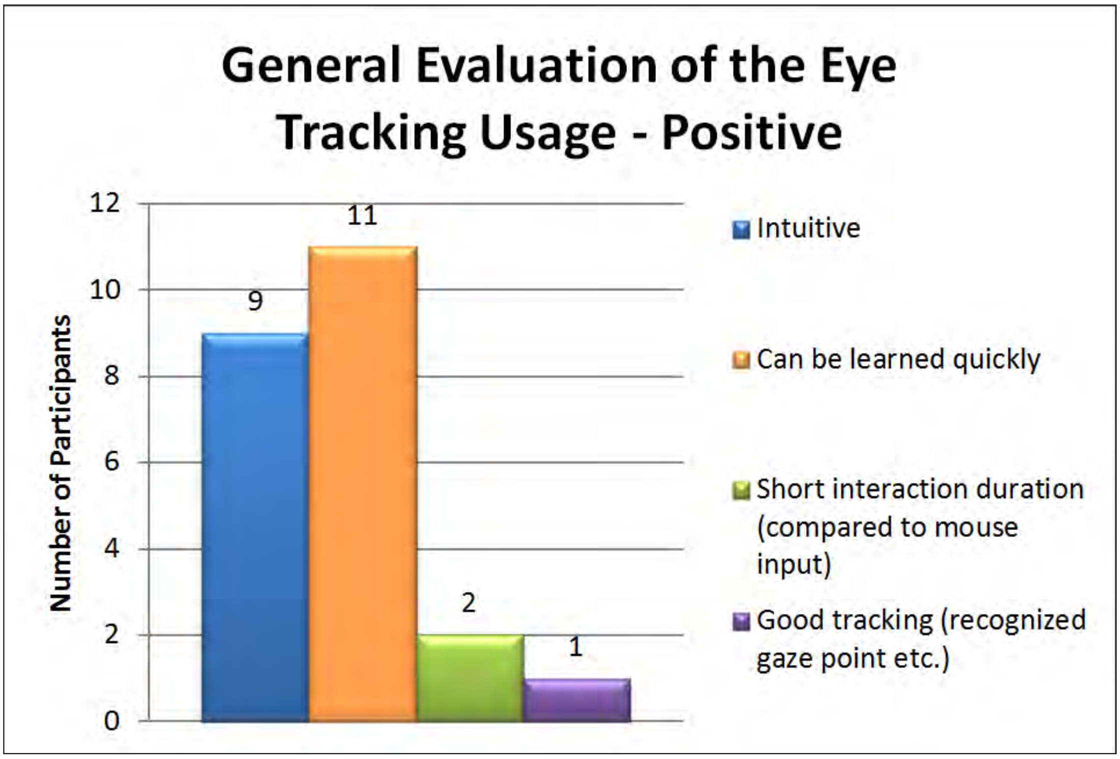


Figure 6.1: The probands were able to rate the usage of the eye tracker. Positive aspects, like for example that the usage can be learned quickly, are displayed here.

the restriction of mobility, like keeping the head still, as a disturbing factor. Nine out of fourteen participants were thinking that the usage is tiring or exhausting for the eyes, while five probands especially disliked the calibration process. Six participants felt that the tracking was not good enough. Too many faulty inputs and a long interaction duration compared to mouse input was criticized by three participants each. Nobody found faults with the point of learnability. The rating is illustrated in graph 6.2.

It can be summarized that the participants were not to satisfied with the eye tracking usage. This is further illustrated by the fact that the majority of probands would not use the system (without changes) for private purpose. The participants were asked to decide whether they would use the system with one of the three interaction methods for private purpose. This is shown in figure 6.3. Nobody would use the system with *Context Switching*. Five would use the system after further changes and eleven would not use the system with *Context Switching* as input possibility at all. At least one proband would use the system when it is utilized with *Dwell Time*. However, eight participants would

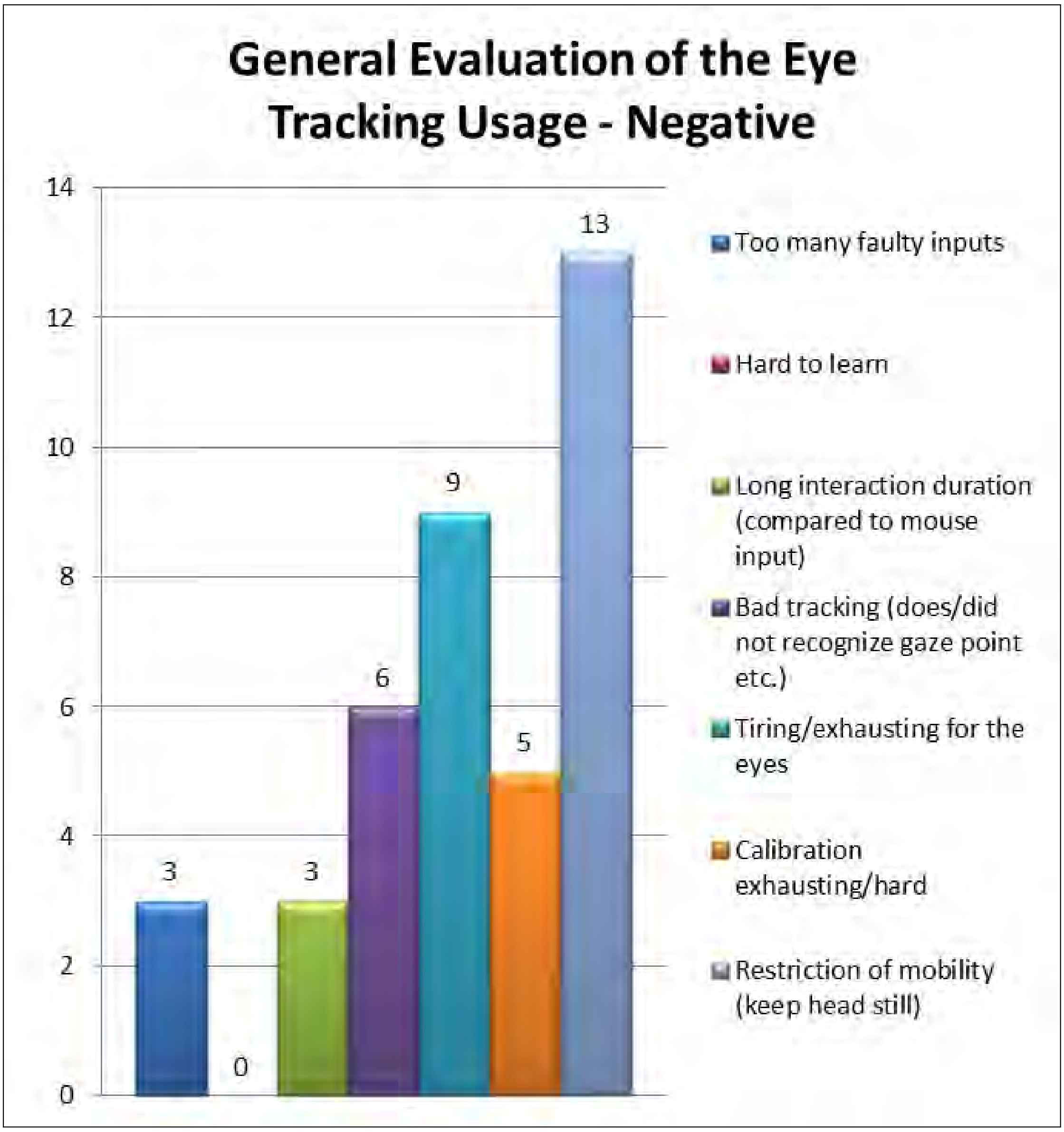


Figure 6.2: Negative aspects of the eye tracking usage as perceived by the probands. Most participants disliked the restriction of mobility.

not use such a system, while five would use it with further modifications. *Key Press* was the best performer. Here, six users would employ such a system for private purpose and three would use it after further changes. Though, there are still five probands who would not use such a system. The following charts clearly show that the amount of reasons why such a system would not be used is much higher than the reasons why they would

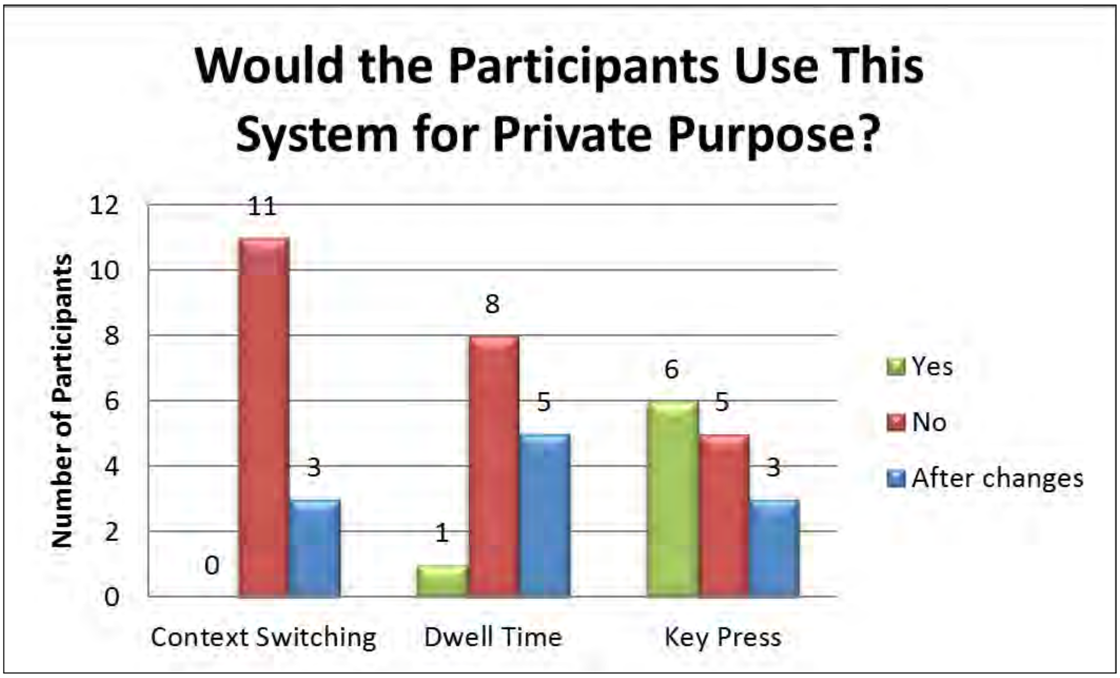


Figure 6.3: The participants were able to decide whether they would use such a system for private purpose. The results are shown in this diagram.

use such a system. Figure 6.4 displays that there is no positive aspect about using *Context Switching*. *Dwell Time* was positively evaluated as fast and easy by one person each. *Key Press* was assessed best. Three persons assessed this input possibility as fast and one user evaluated this input method as intuitive. Moreover, one person each appreciated the fact that no prior learning is necessary and the combination of gaze and keyboard operates in conjunction. In contrast, however, the negative assessments outweighed the positive ones. Especially the little freedom of movements, the interaction duration and the overall exhaustion and circuity bothered the users. Here, too, *Key Press* received the best marks with only five bad aspects which were each only named once. Seven issues have been named in view of *Dwell Time* as interaction techniques. All in all, *Context Switching* has been assessed worst. Especially the duration of the input as well as the fact that this method was rated as exhausting and too cumbersome, has been criticized. All negative results can be seen in graph 6.5.

As described above, some participants would use the system with one of the three input methods after further changes. Here, the most often demanded change, for all

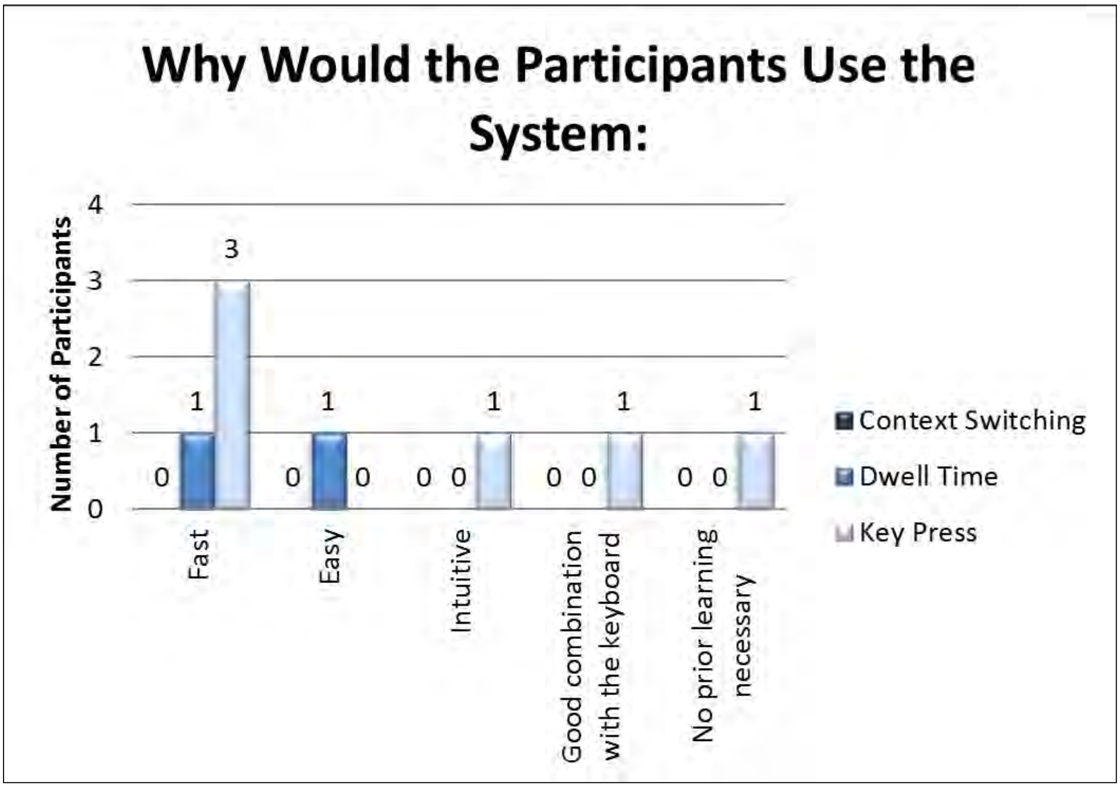


Figure 6.4: This figure illustrates why the participants would use the system. The results are shown per interaction method.

three interaction methods, is to increase the freedom of movement. While for *Key Press* only two different changes have been demanded, six changes have been requested for *Context Switching* and even seven different ones have been called for *Dwell Time*. An overview is given in figure 6.6.

While in *Context Switching* the second most requested change that has been asked for was a shorter interaction duration (claimed by two probands). In *Dwell Time* the tracking should be more precisely according to two participants. The remaining changes were all requested by only one participant per interaction method.

The overall calibration process evaluation also reflects the rather bad than good results for general eye tracking usage (depicted in diagram 6.7). While nine out of fourteen users classified the calibration process as medium exhausting, five users even rated it as exhausting. Nobody felt that the calibration process was not exhausting. At this, a crucial factor could be that the participants needed an average of 1.74 tries until the

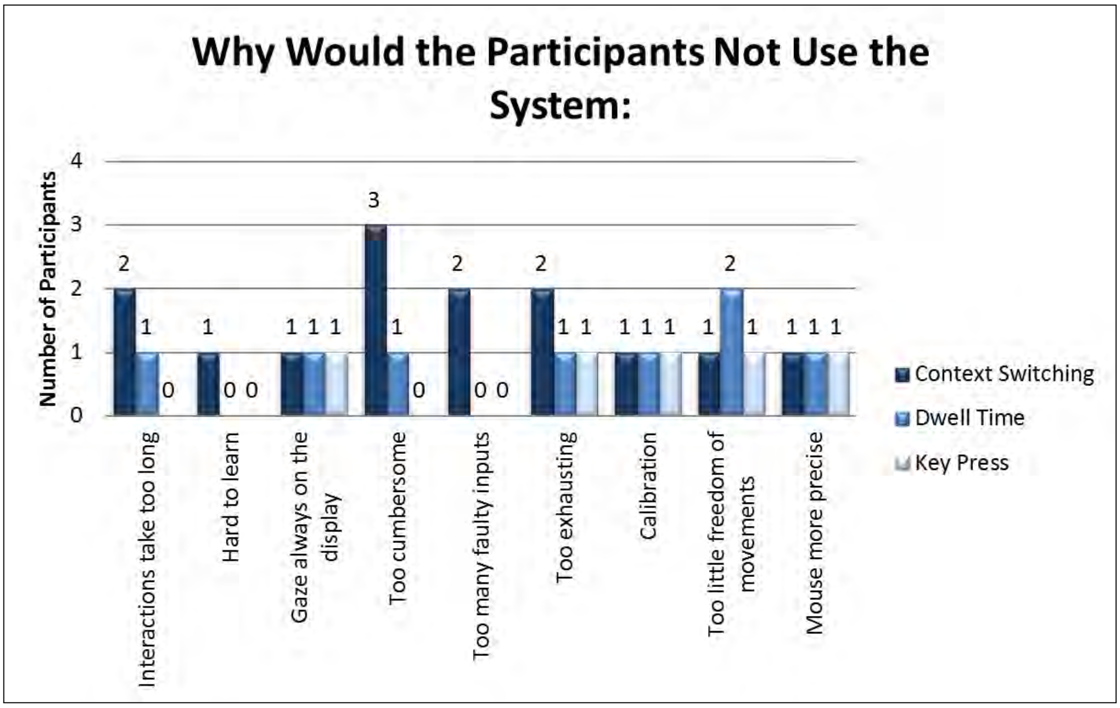


Figure 6.5: Illustration of the facts why the participants would not use the system. The results are shown per interaction method.

calibration process succeeded. So, this brings together a great number of tires, since the calibration had to be repeated for each interaction method at least twice (for the testing and the actual study relevant iteration).

Furthermore, the questionnaire asked the participants to classify the size of the different elements (textfield, character (key) buttons during the text input, checkboxes and radiobuttons, slider and the *Done* button). This is illustrated in diagram 6.8. Here, nearly all participants (in a range of nine to twelve probands out of fourteen) ranked all components' sizes as "exactly right". Only ten out of fourteen participants ranked the *Done* button as "too small". This is surprising, since the button's "clickable" region and the characters on the on-screen keyboard are of comparable size. The regions, where an element receives focus, are marked in figure 4.11 and 4.12 in paragraph 4.5. So, it seems that the location of the elements played an important role as well. The *Done* button was located at the bottom of the screen while the characters were located in the middle of the screen.

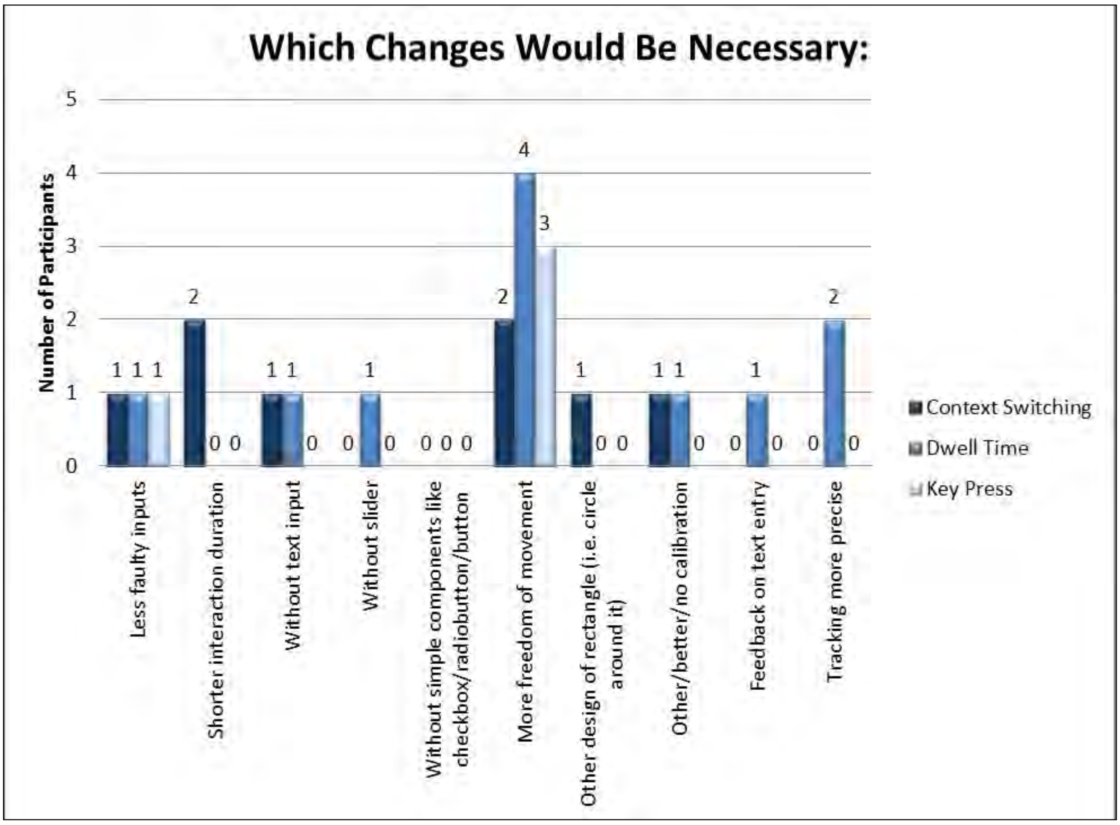


Figure 6.6: Overview about the changes that would be necessary before the participants would use the system. All changes are grouped per interaction method.

All in all, it can be said that the negative aspects outweigh the positive ones. Especially *Context Switching* scored low. It was rated worst by the participants and no positive aspects were mentioned here. *Dwell Time* performed better. Still, most participants would not use the system (without changes) with this interaction method. *Key Press* was rated best. Here, several positive aspects were mentioned and nearly half of the probands would use the system with this interaction method without further changes. But even here, some negative aspects were mentioned. An important part plays the calibration process. Since nobody rated this procedure as "not exhausting" but nine rated it as medium exhausting and five ranked it as exhausting, this might be a reason for the overall demotion.

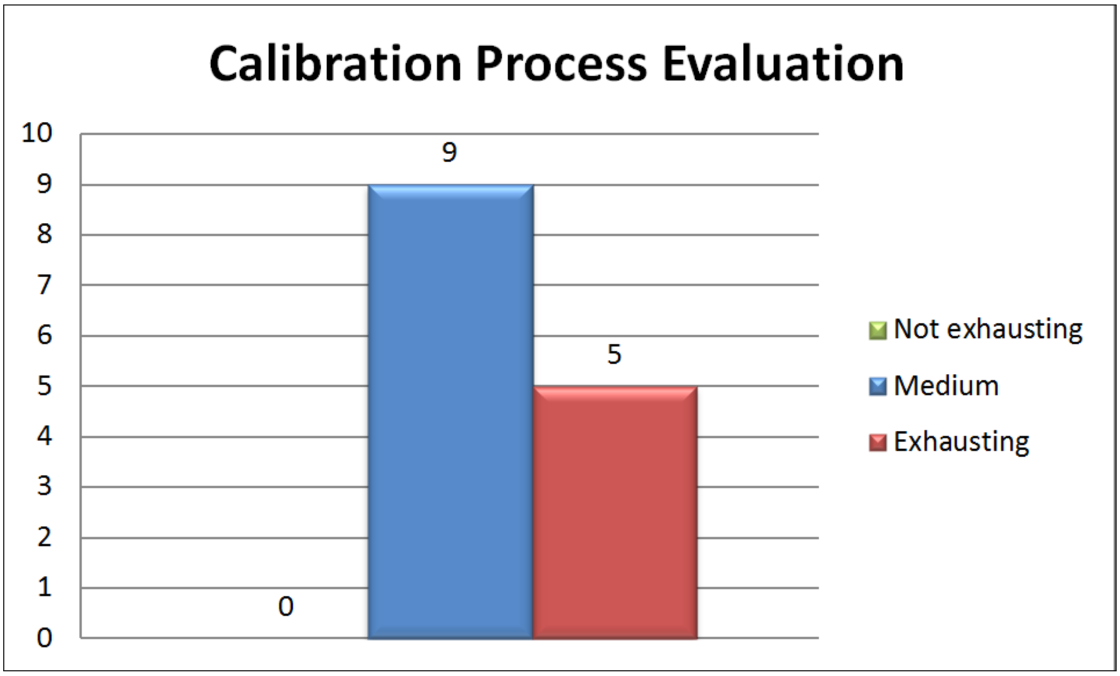


Figure 6.7: Illustration of the calibration process evaluation.

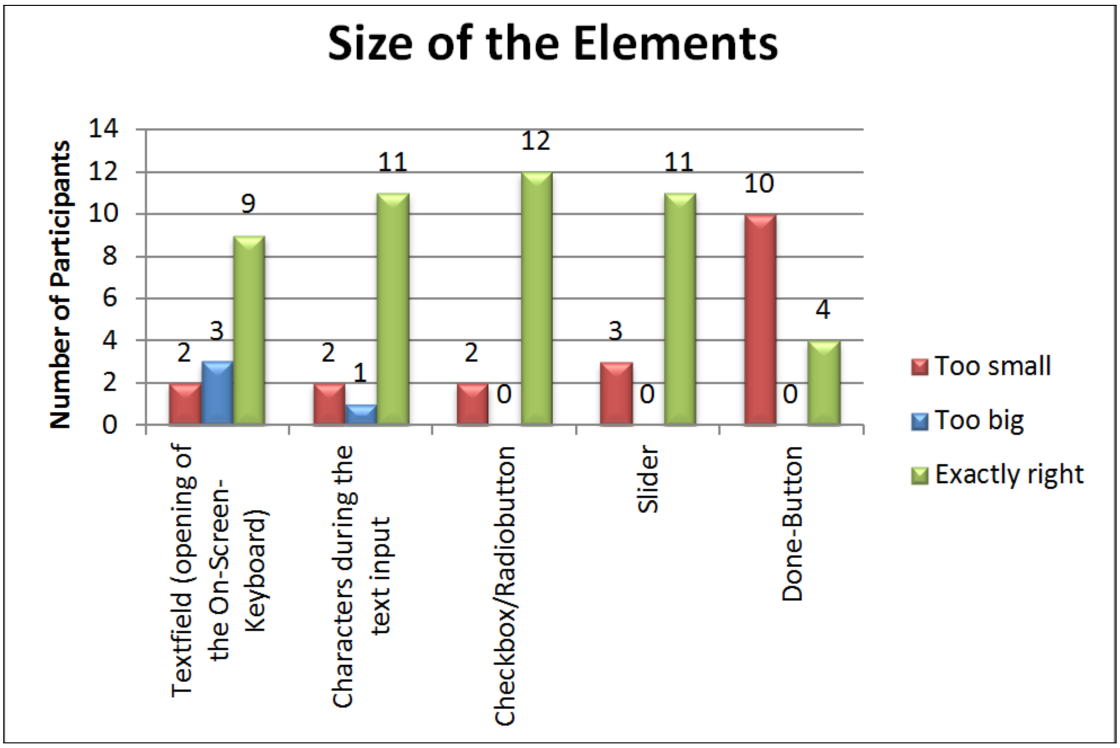


Figure 6.8: Evaluation of the questionnaire's elements' size: too small, too big, exactly right.

6.3 Comparison of the Three Interaction Methods

All in all, the participants mostly preferred the input method *Key Press*. Eleven out of fourteen probands lean towards this method, while multiple entries have been allowed. One subject rated all three methods as equally. Two participants would prefer *Context Switching* or *Dwell Time*, one participant has no preference. This is summarized in figure 6.9.

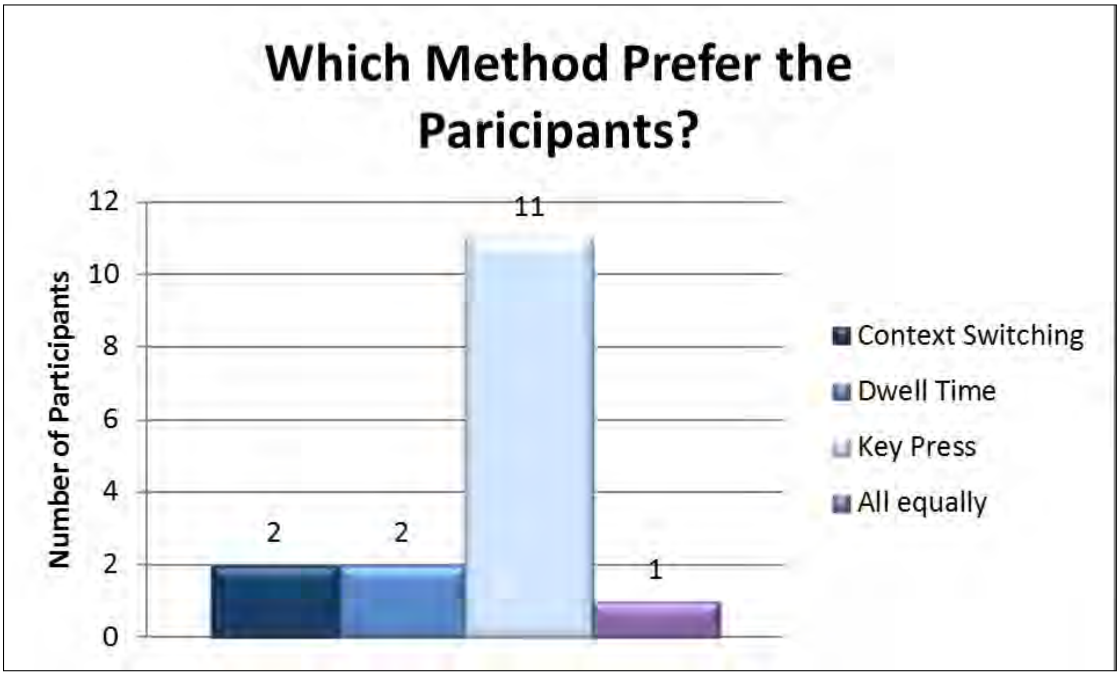


Figure 6.9: This chart shows which interaction method, *Context Switching*, *Dwell Time* or *Key Press*, would be preferred by the participants.

In general, most of the users have been rating the interaction technique *Key Press* as intuitive (thirteen out of fourteen), as quickly learnable (twelve out of fourteen) and nine out of fourteen have appreciated the short interaction duration (compared to the other two input methods *Context Switching* and *Dwell Time*). Furthermore, one participant each liked the simple usage and the fact that less errors occurred during input, compared to the other two interaction methods. *Dwell Time* was receiving several positive ratings, as well. Here, ten out of fourteen users felt that the method is intuitive while eleven participants thought that it could be learned quickly. But, compared against *Key Press*, only four probands out of fourteen felt that *Dwell Time* offers a short interaction method compared to the other two possible input techniques. Moreover, nobody gave *Dwell Time* another positive aspect. Nevertheless, *Dwell Time* receives better results than *Context Switching*. Here, half of the participants thought that the method can be learned quickly. Only three out of fourteen probands each felt that *Context Switching* can be rated as intuitive and offers a short interaction technique, compared against the other two input methods. All positive ratings are represented in diagram 6.10.

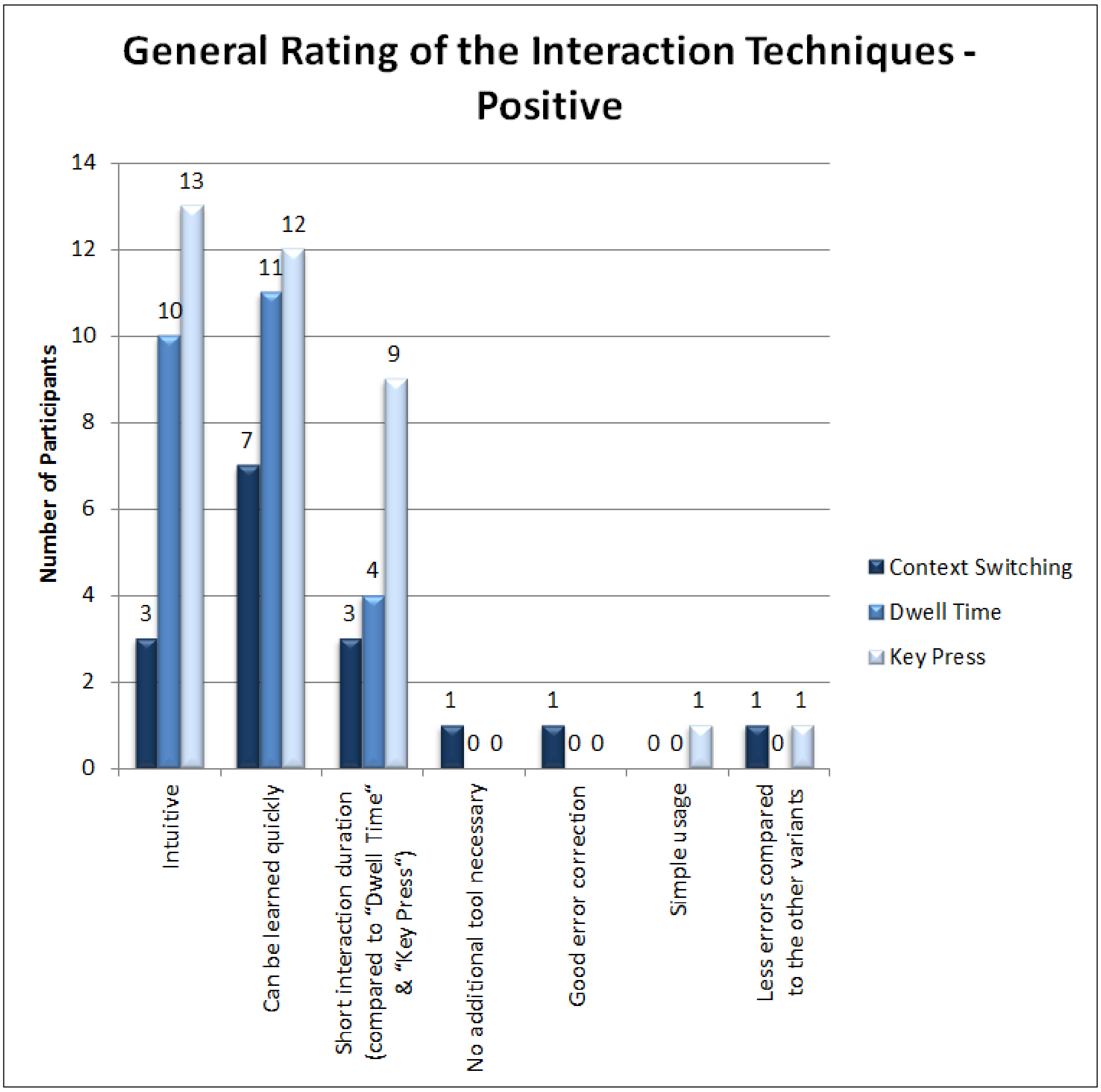


Figure 6.10: Illustration of the general positive rating of the interaction techniques (*Context Switching*, *Dwell Time* and *Key Press*).

The participants had the possibility to assess the three methods negatively, as well. Two participants disliked the fact that an additional tool is needed for *Key Press*. It is worth mentioning that this was the only negative fact given about this interaction method in general. Both, *Context Switching* and *Dwell Time* were evaluated negative. Three probands out of fourteen each felt that too many faulty inputs have been produced with the two interaction methods. In *Context Switching* eight probands out of fourteen thought that the interaction duration was too long compared to the other possible interaction techniques, while six out of fourteen users evaluated the interaction duration for *Dwell Time* as too long. Five users assessed the interaction method as tiring/exhausting for *Dwell Time* whereas two felt this way while using the method *Context Switching*. The results are completely shown in figure 6.11.

So, this emphasizes the fact that by far most participants would prefer *Key Press* as interaction technique. This method has been rated best in the field with the highest positive rating and the lowest negative one.

Additionally, the usage of the questionnaire's components has been evaluated as highly, medium or poorly usable. The components have been summarized in text input, slider and simple components (checkbox, radiobutton and button). In this case, *Key Press* performed best, as well. All three components have been rated as highly usable by nearly all users. Twelve out of fourteen participants each assessed slider and text input as highly usable, whereas eleven probands felt the same way about simple components. Only one participant rated the usability of text input as poor, while the remaining participants rated the components' usability as medium. The usability of the simple components in *Dwell Time* have been assessed as high by ten users and have been evaluated as medium by four probands. Moreover, half the participants evaluated the slider as highly usable while the other half thought that the usability can be rated as medium in *Dwell Time*. Only the text input was rated badly by four participants. Half of all probands felt that the usability of text input can be rated as medium, while only three users rated the usability as high. *Context Switching* has been rated the worst. Nine out of fourteen probands thought that text input has a poor usability in this scenario. Three participants thought so about

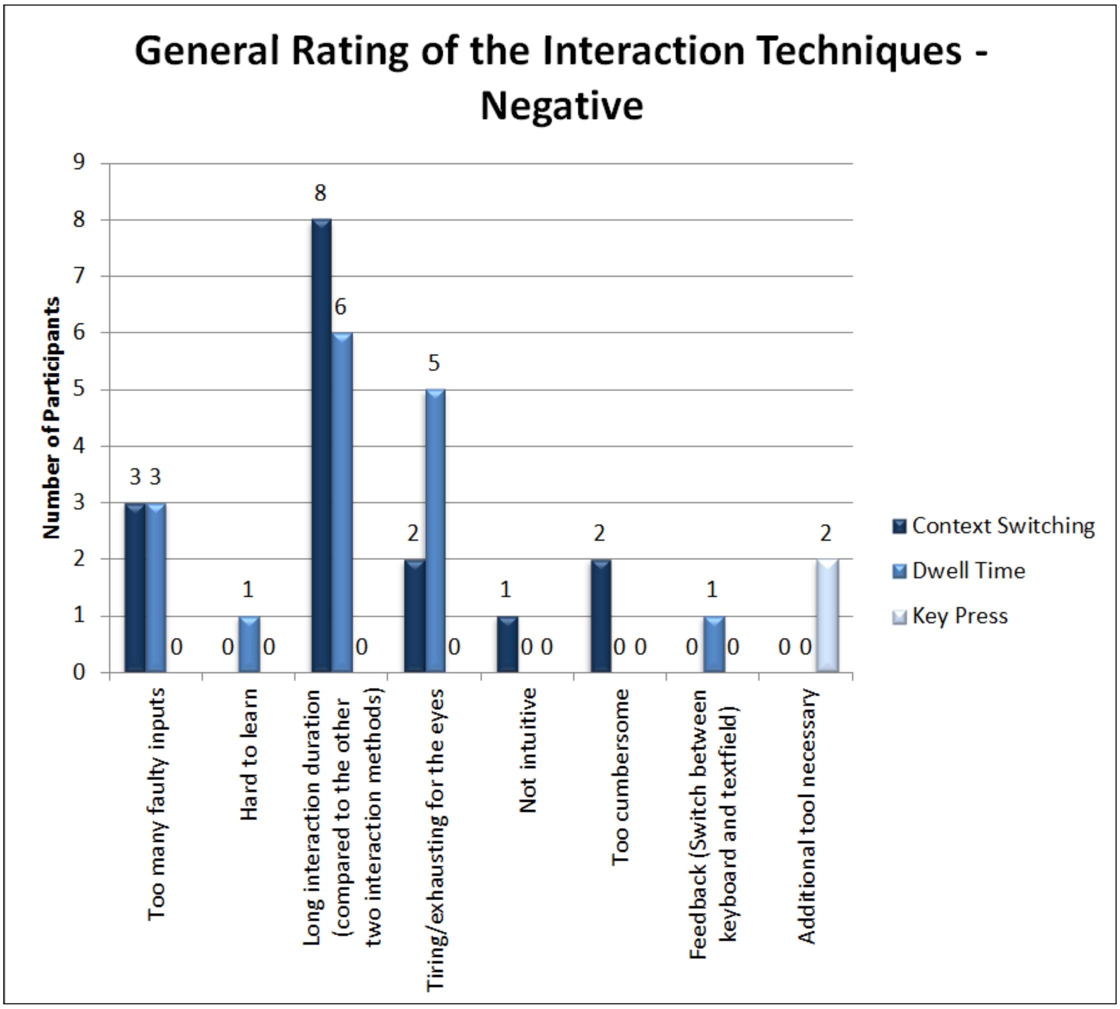


Figure 6.11: Figure that shows the general negative rating of the interaction techniques (*Context Switching*, *Dwell Time* and *Key Press*).

slider input and two felt that way about simple components. Furthermore, the slider's usability has been rated as medium by ten participants and as high by one proband. The text input usability was assessed as high by one user as well, and it has been rated as medium by four participants. The simple components did the best with a high rating by seven users and a medium rating by five probands. All results are summarized in figure 6.12.

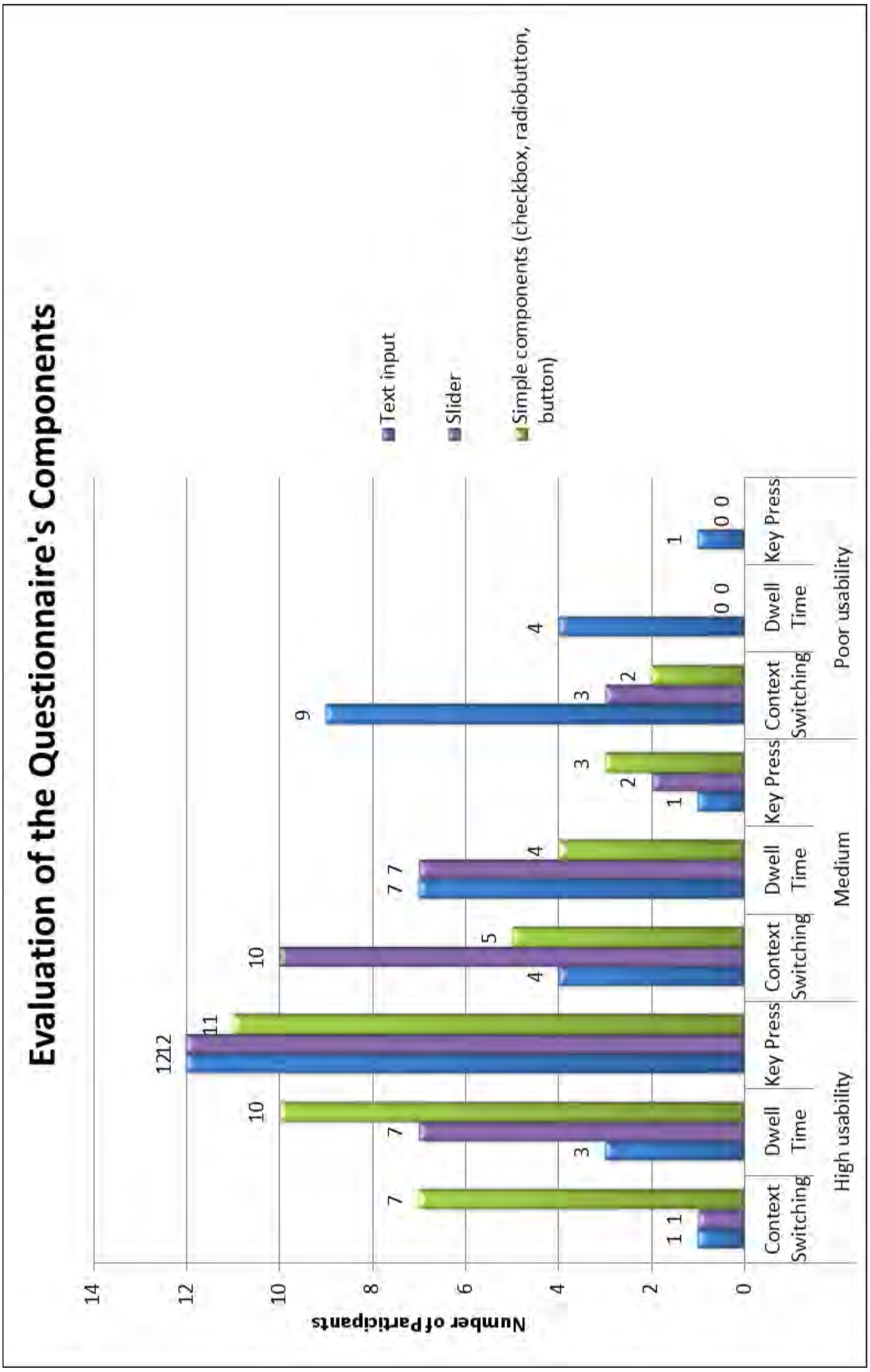


Figure 6.12: Evaluation of the questionnaire's components. The components are grouped in text input, slider, and simple components like checkboxes or buttons.

All in all it can be said that *Key Press* has been assessed as the best input method (see section 6.2 as well). It has always been rated best in terms of general usage and the usability of the particular components.

This subjective feedback is also backed by the data obtained from/by the logging. Here, the needed time to complete the questionnaire has been logged. The most time was needed with *Context Switching* to complete the questionnaire (104 seconds). For the completion of the questionnaire with *Dwell Time*, the used time was high, as well, compared against *Key Press*. 79 seconds were needed, while for *Key Press* only 22 seconds were required to complete the questionnaire. This is summarized in diagram 6.13.

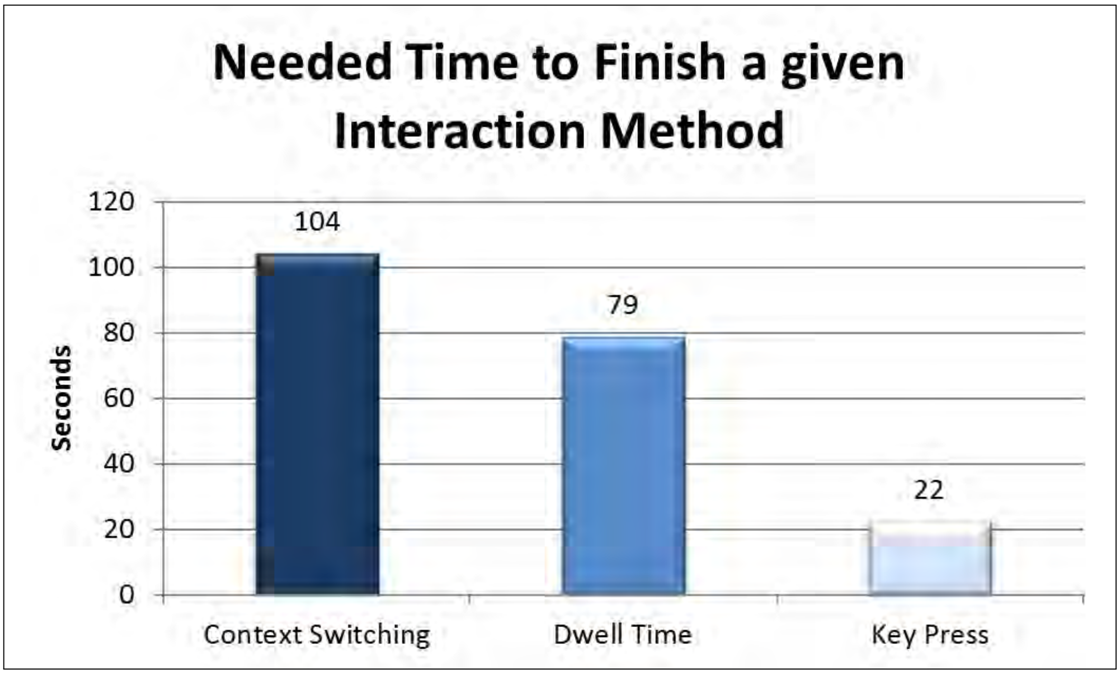


Figure 6.13: This figure displays the needed time to finish a given interaction method.

Furthermore, it can be seen that *Key Press* was almost always the fastest method. It took between two and five seconds to fulfill a task. Compared against *Context Switching*, the needed time ranged between four and 62 seconds in average. The needed time to fulfill the tasks with *Dwell Time* ranged between three and 42 seconds.

In greater detail, the slider movement with *Dwell time* was performed in three seconds while it took five seconds to accomplish the movement with *Key Press*. With *Context Switching* the movement of the slider to the correct position has been performed in seven seconds. So, all three interaction methods are suitable to fulfill the task in a reasonable amount of time. Besides, this is the only task where *Key Press* has not achieved the fastest results. When it comes to text input, it required the most time of all with *Context Switching* (62 seconds in average). The time needed for text input with *Dwell Time* was with 42 seconds high, but 20 seconds less compared to *Context Switching*. Here, *Key Press* performed best, which is not surprising considering the fact that a physical keyboard is used. The complete information is shown in figure 6.14.

Faulty inputs can also be determined with the help of the logfiles. The most faulty inputs have been produced with the interaction technique *Dwell Time*. Here, each participants produced 3.5 faulty inputs, with a range between zero at the lowest and twelve at the highest. In *Context Switching* the average is much smaller with only 0.9 faulty inputs per participant with a range between zero and three faulty inputs. The best performing input method was *Key Press* with an average of 0.4 faulty inputs per participant. Only two participants entered faulty clicks (in the range between one and five). The exact details can be seen in figure 6.15.

Once again it has been demonstrated that *Key Press* turned out best. It was by far the most chosen interaction method when the participants would have to choose one. Moreover, it was rated best of all three interaction methods as well. Only two probands found a negative aspect about this input method: the additionally needed tool. Furthermore, *Key Press* did best when it comes to the overall questionnaire's components rating. Here, only one user rated one component as poorly usable. The short needed time to fulfill a task or to fulfill the complete questionnaire with *Key Press* reflects the good overall rating, as well. The evaluation results of *Context Switching* and *Dwell Time* are comparable. If a comparison must be made, *Dwell Time* can be rated slightly better.

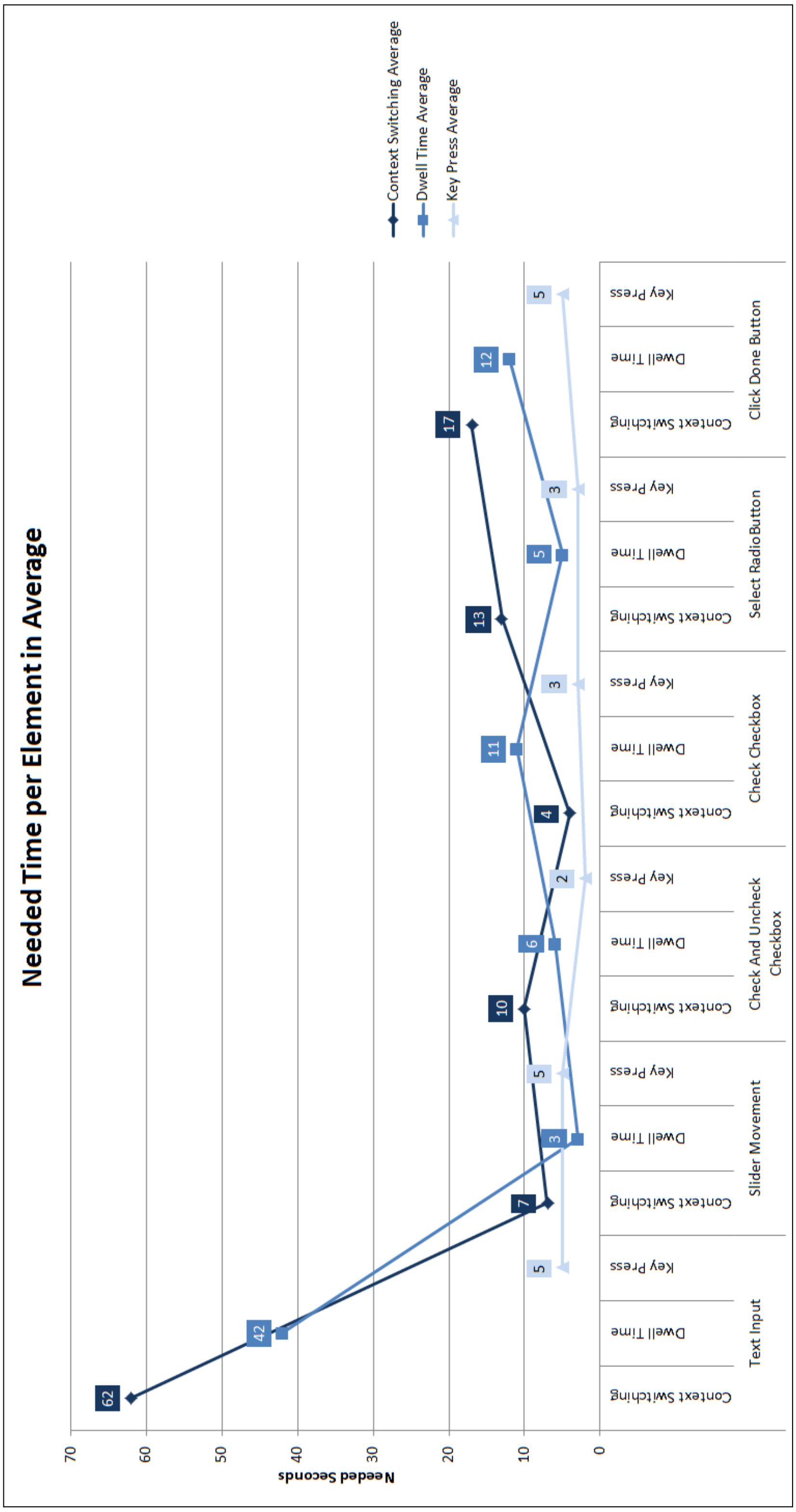


Figure 6.14: An average of seconds needed for each element type is shown. The results are grouped by the interaction methods.

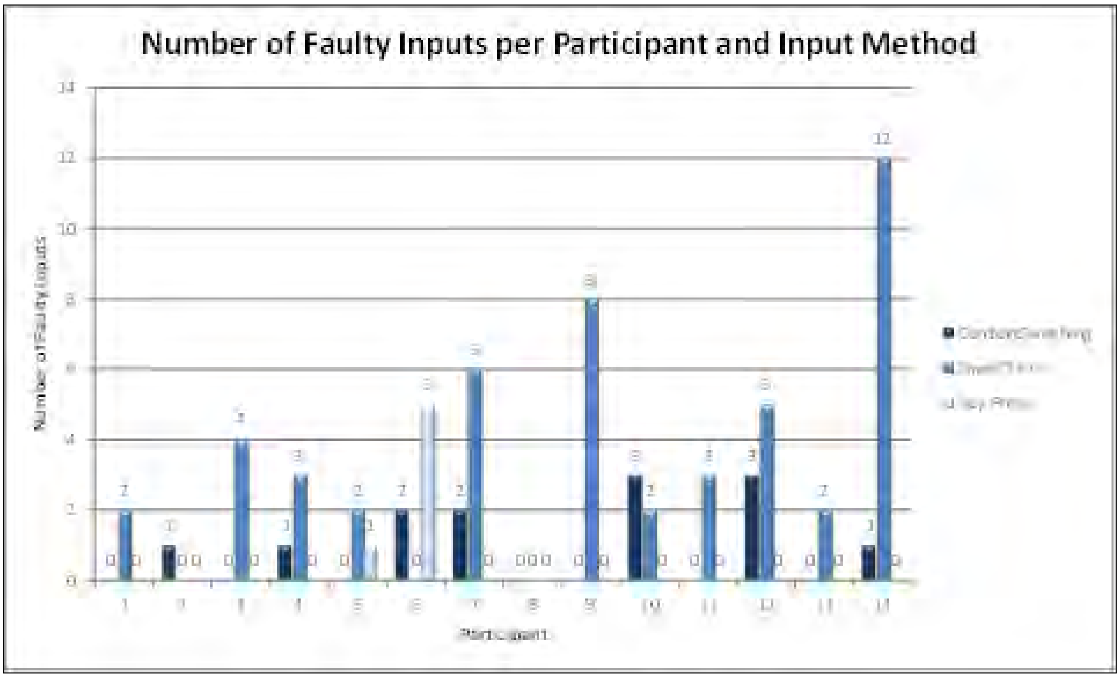


Figure 6.15: The numbers of faulty inputs per participant are shown here. The results are grouped by the input methods.

6.4 Specific Results per Interaction Method

Some specific questions per interaction method have been asked as well. These are evaluated in this section.

Dwell Time

When using the interaction method *Dwell Time*, an important part is played by the duration an element has to be focused to be chosen. Like mentioned in paragraph Dwell Time, the duration was estimated to be 750 ms. The participants were asked to assess the duration of the *Dwell Time*. Here, ten out of fourteen probands ranked the duration to be "exactly right". Two participants each ranked the duration as "too long" and "too short". Consequently, the time needed to fulfill the *Dwell Time* seems to be chosen right. The results are illustrated in figure 6.16.

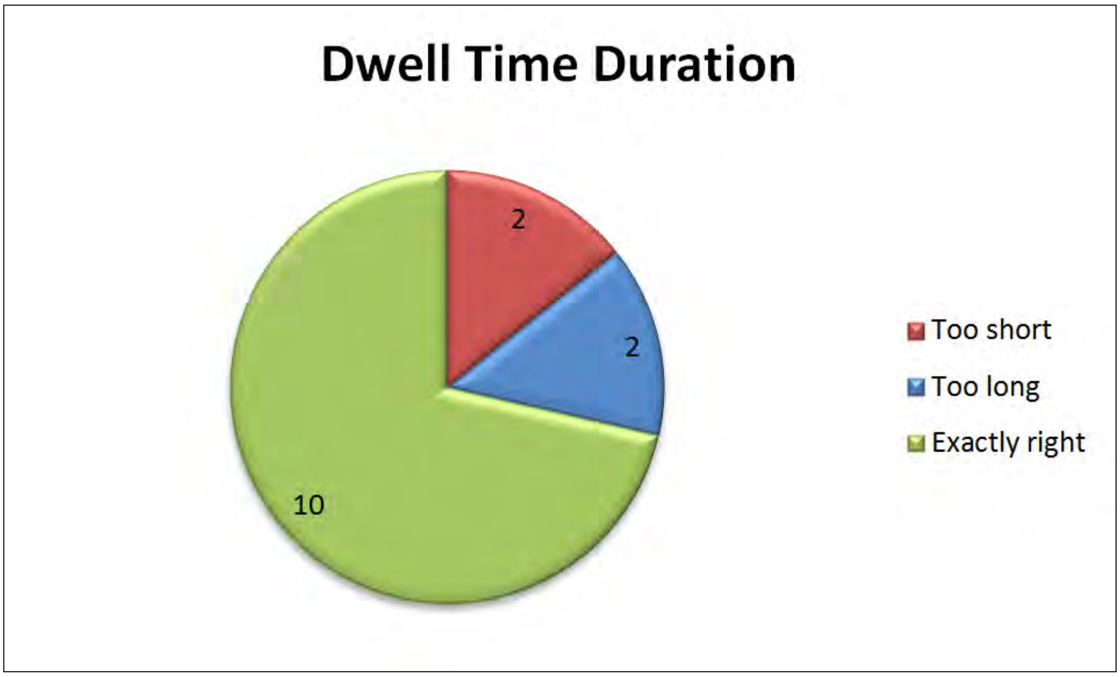


Figure 6.16: This diagram illustrates the rating of the *Dwell Time* duration. All fourteen participants had to rate the duration as "too short", "too long" or "exactly right".

Context Switching

In the interaction method *Context Switching*, the participant selects a component when they perform a two-step gesture (see paragraph Gaze-Only Interaction Methods for further information). As soon as an element receives focus, a blue rectangle appears above/next to the element. This rectangle has to be focused in a given time (3000 ms). Afterwards the element has to be focused again in the same time span (3000 ms as well). So, not only the duration of the time span but also the size of the appearing rectangle played an important role. The duration the blue rectangle is shown was rated as "exactly right" by eight out of fourteen users. Two participants felt that the duration was "too short", while four thought the duration was "too long". So, the duration could be a little bit shorter to please more probands. But, all in all, the duration the rectangle appears seems to be chosen correctly. A diagram displaying these result is shown in figure 6.17.

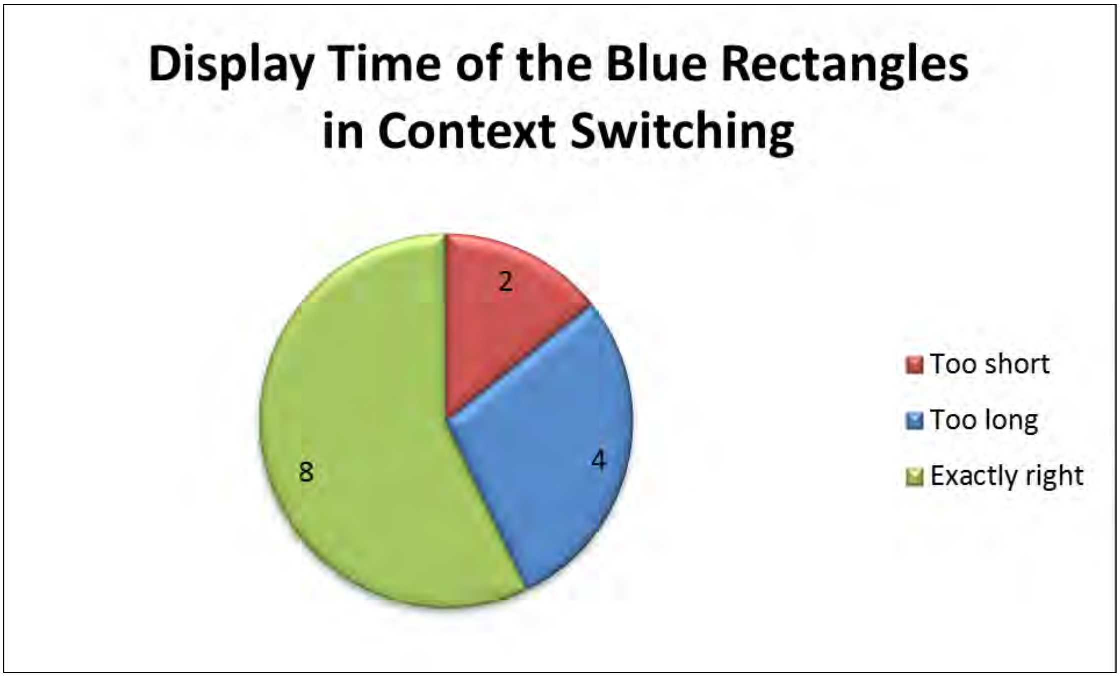


Figure 6.17: Duration of the display time of the blue rectangles which appear during the two-step gesture in *Context Switching*. All fourteen participants had to rate the duration as "too short", "too long" or "exactly right".

The size of the appearing rectangles has been evaluated as well. The participants were able to rank the size of each questionnaire component. All results can be seen in diagram 6.18. The rectangle's size appearing above the checkboxes/radiobuttons was ranked as "exactly right" by all participants. The rectangle appearing above (or besides) the textfield, the characters appearing during text input and the slider have been ranked as "exactly

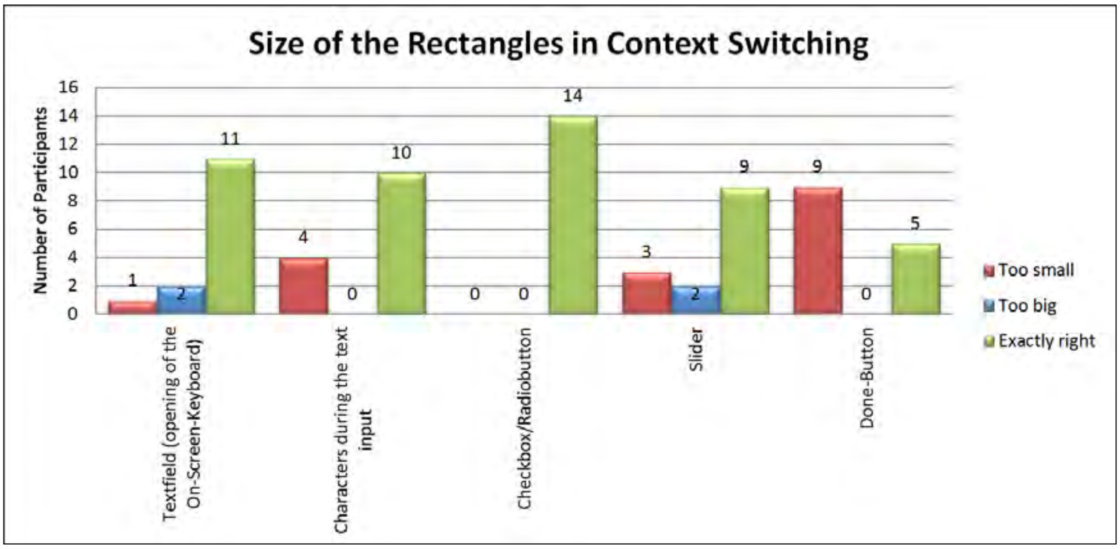


Figure 6.18: Size of the appearing rectangles during the two-step gesture in *Context Switching*. Since the size differs according to the element, the rating has been made per element group (textfield, characters, checkbox/radiobutton, slider and *Done* button).

right" in size by the most participants (between nine and eleven out of fourteen probands). Only the rectangle appearing above the *Done* button has been evaluated as "too small" by most users. Here, nine probands thought this way, while only five ranked the size as "exactly right". This is surprising, as mentioned in the evaluation of the overall size of the elements in section 6.2, since the rectangles appearing above the *Done* button and the ones appearing above/next to the characters during the text input are of comparable size.

All in all, it seems that the specific parameters of an interaction method were chosen adequately. The duration needed to select an element in *Dwell Time* was rated good, at large. The same applies to the appearance of the rectangles in *Context Switching*. Only the implementation of the *Done* button was rated poorly.

6.5 Usage of Eye Tracking to Complete a Questionnaire

A research question was whether an eye tracker can be used to complete a questionnaire, respectively whether it is useful. Half of the participants thought that it is not useful in that scenario. However, the system was evaluated as useful to complete a questionnaire by five probands while two users believed that this system could be useful for this purpose after further changes. This is depicted in figure 6.19.

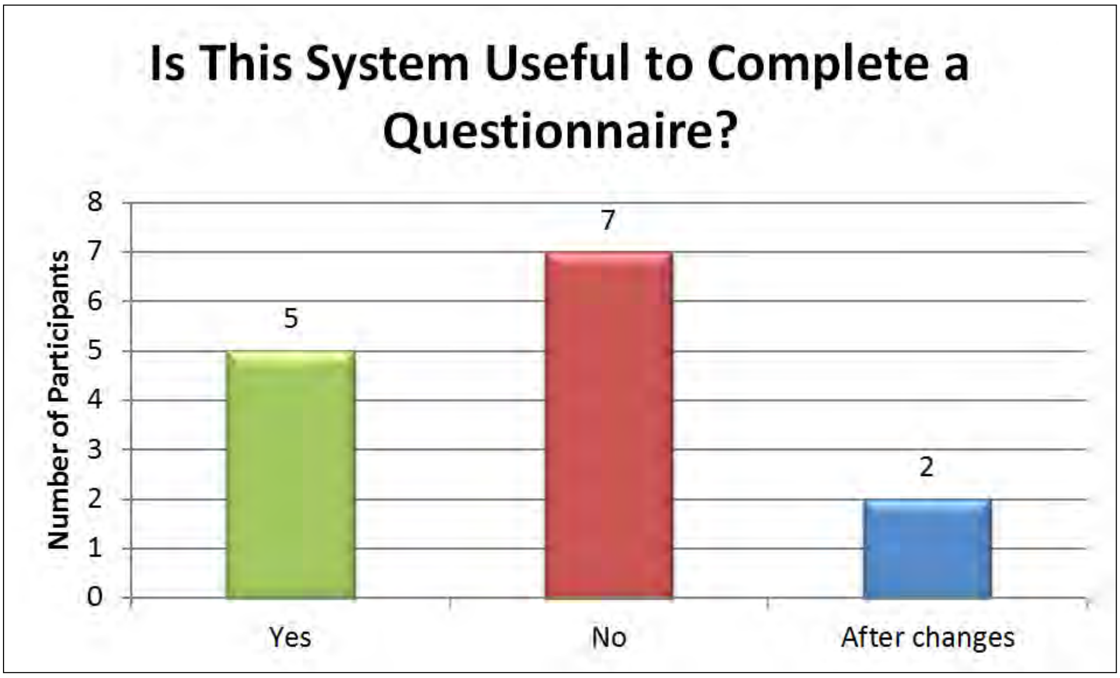


Figure 6.19: Evaluation whether this system is useful to complete a questionnaire. "No", "After changes" and "Yes" were possible answers.

The amount of reasons for a negative rating outweigh the amount of reasons for a positive result. Seven different aspects have been mentioned when it comes to reasons why participants would not use this system to complete a questionnaire. Here, three users rated the system as too exhausting. Moreover, reasons like the calibration process and the slow interaction procedure are among the most significant drawbacks/opinions/impressions that have been indicated/expressed/mentioned by the participants by one participant each. All reasons are displayed in figure 6.20.

With seven negative aspects compared to five positive ones, negative aspects have been mentioned more often than positive ones. Most of the positive aspects have been mentioned by one participant each. For instance the usage has been evaluated as fast and fun by one participant each. The fact that "plain widgets" (such as checkboxes

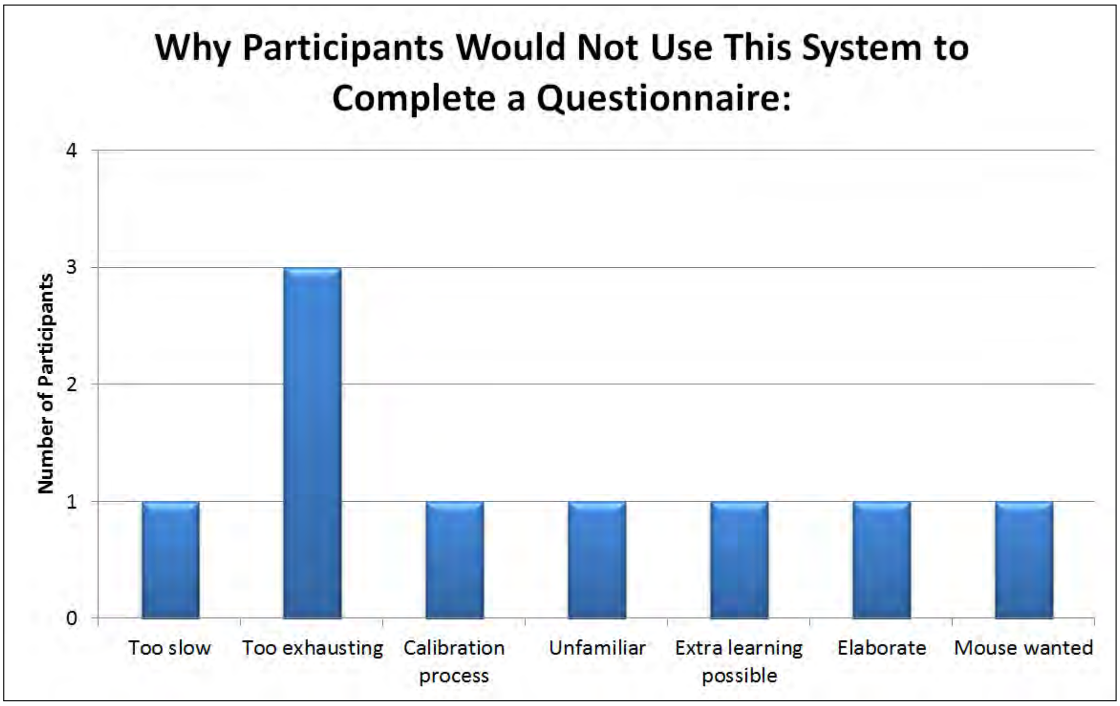


Figure 6.20: Reasons why participants would not use this system to complete a questionnaire. Multiple answers were allowed.

and buttons) are often used in a questionnaire and the good fixation of elements was positive for one user each. The fact that such a system is useful for disabled people was positively ranked by two users. All in all, multiple entries were allowed. An overview of the results is given in diagram 6.21.

Two participants would think that this system is useful after further changes. Here, four necessary modifications have been mentioned (all by one participant each). The interaction duration has to be shorter, the inputs may not be that faulty and the text input and the slider have to be removed completely. This is illustrated in diagram 6.22.

In consequence, the conclusion can be drawn that, after some modifications, such a system can be useful to complete a questionnaire. Depending on the implemented interaction method, complex elements have to be dismissed from the questionnaire (like for example the slider). Furthermore, the overall tracking experience has to be less exhausting, which can be achieved by granting more freedom of movement and a less

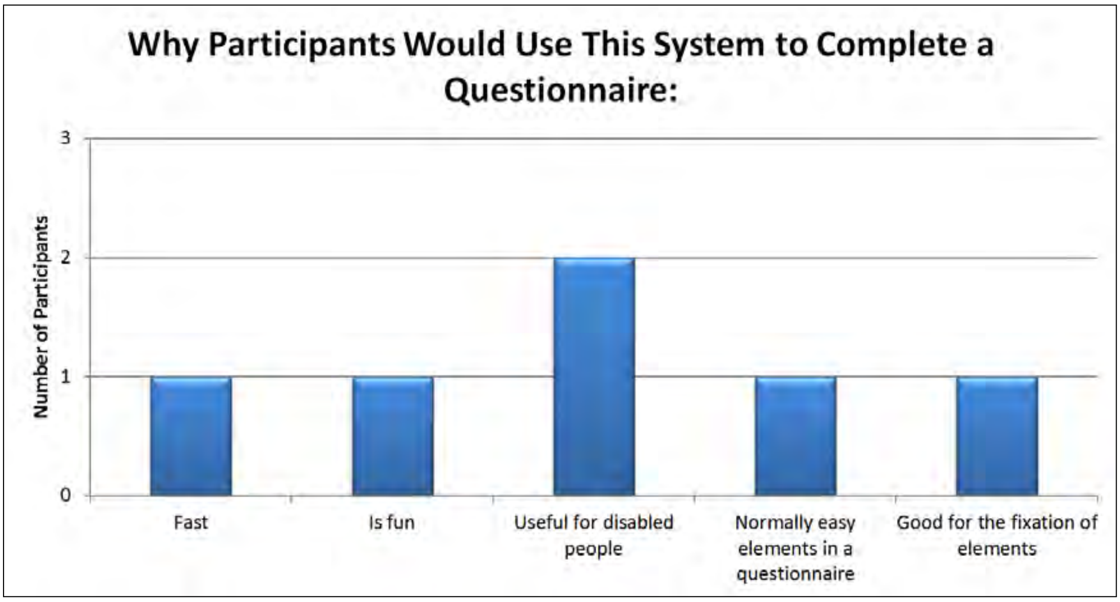


Figure 6.21: Reasons why participants would use this system to complete a questionnaire. Multiple answers were allowed.

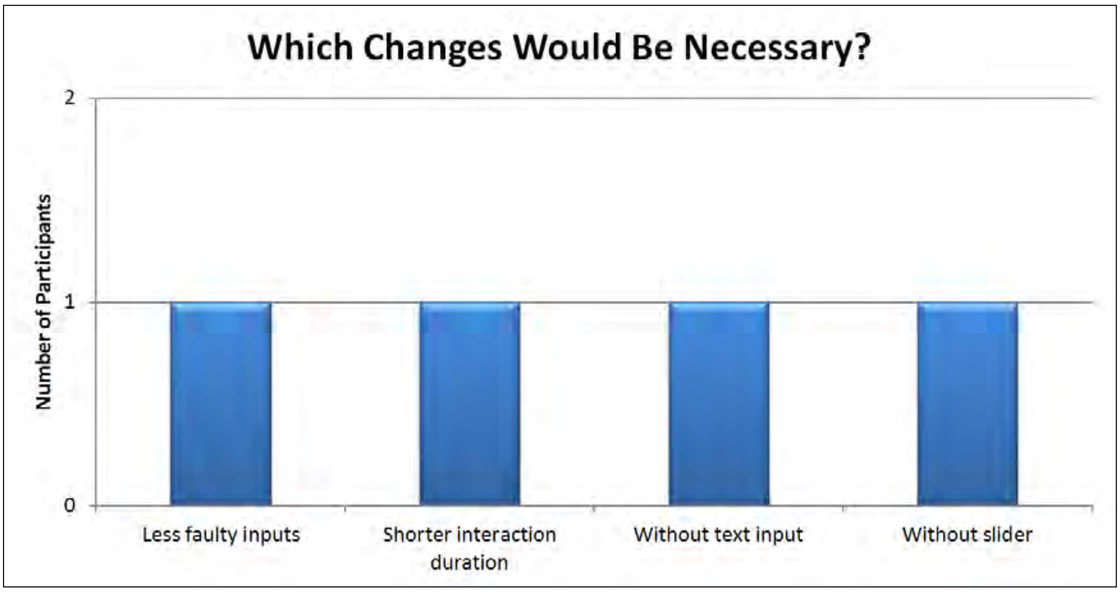


Figure 6.22: Illustration of the changes that would be necessary until the participants would use the system to complete a questionnaire. Multiple answers were allowed.

exhausting calibration process or none at all. In doing so, such a system might turn out to be a suitable alternative to conventional input modalities and especially useful for disabled people.

6.6 Evaluation of the Study Results

It can be assumed that most people would prefer *Key Press* as an interaction technique. This is not really surprising since the participants are accustomed to make use of the keyboard as an input device. So, the extra learning remains small. Furthermore, the *Midas Touch* problem (for further information see section 3.5) does not play a role here. Not both, the fixation and clicking, is done by the eyes but only the fixation. This is also reflected in the low amount of errors as determined by the study. Only two participants had errors at all while half of the users produced faulty inputs with *Context Switching* as an input method and even eleven out of fourteen probands had errors during the usage with *Dwell Time*. Moreover, the questionnaire has been completed fastest while using the input method *Key Press*. More than half of all participants would use the system (after further changes) with *Key Press*. Compared to that more than half of the participants would not use the system with *Dwell Time* and nearly all probands (eleven out of fourteen) would not use the system with *Context Switching*. Taken as a whole, *Dwell Time* was rated second best. Consequently, it is conceivable that, after changes such as reducing the possibility of faulty inputs, *Dwell Time* could be an appropriate interaction option that relies exclusively on the eyes.

Furthermore, this system can be useful to complete a questionnaire. Especially with *Key Press* (the most accepted interaction method) the system can be an enrichment for users. It seems that complex questionnaire elements, like the text input, should be dismissed when it comes to the other two, eye-based only input methods. This can be explained with the much higher interaction duration when the user wants to type something. While with *Key Press* the participants needed five seconds on average to type the word "HELLO", it took 42 seconds to do that while using *Dwell Time* and even 62 seconds while working with *Context Switching*.

All in all, the overall eye tracking experience should be improved. First, the calibration process has to be simplified/dismissed to reduce the exhaustion of the eyes. Long interaction durations should be avoided since this amplifies the exhaustion of the eyes, as well. The amount of faulty inputs should be reduced by a more permissive hardware and software implementation of eye trackers. One of the most criticized aspects was the restriction of mobility. The user should be able to move in front of the tracker as free as possible without affecting the interaction.

7 Conclusion and Future Work

The purpose of this thesis was to investigate the usability and acceptance of eye tracking as an input mechanism. So, after a conclusion of this work, summarized in 7.1, an outlook is given in 7.2.

7.1 Conclusion

Starting with chapter 2, an overview about related work addressing the usage of an eye tracker as an input device was given. The interaction methods can be divided into two categories, gaze-based methods and techniques where gaze is combined with a second modality. When using gaze-only interaction methods, the user is able to manipulate the interface with only their gaze. The considered methods were the so called *Dwell Time*, gaze gestures, blinking and *Context Switching*. Methods using gaze in combination with another input device include gaze combined with key press, with a touch input and with voice recognition as well as with other devices like a remote. The feedback a gaze enabled system could provide, is considered as well. The importance of feedback becomes evident when it comes to interaction methods using gaze.

An overview about eye tracking is given in chapter 3. To give a deeper insight into eye tracking, the history of eye tracking was summarized. The origin of eye tracking lays in the 1800s. At this time, scientists started to study eye movements. Following, a variety of devices that can be used to track the eyes have been investigated. Today, eye tracking can be divided into four different categories, *electro-oculography*, *scleral contact lens/search coil*, *photo-oculography (POG) and video-oculography (VOG)* and *video-based combined pupil and corneal reflection*. Each method has been explained in greater detail and examples have been given. Furthermore, a choice of current eye tracking devices have been given. Therefore, several trackers have been introduced. These devices can be divided into two main groups, those that will be worn (wearable eye tracker) and the remote trackers. Consequently, an overview of existing applications has been given. These applications can broadly be divided into two main categories. On the one hand, *diagnostic applications* provide information related to the user's "visual and (overt) attentional processes" (see [Duc07]). On the other hand, *interactive applications* are used to perform interactions like eye typing (typing with the help of the eyes). The limitations of gaze-based tracking has been investigated lastly. One problem occurring with eye trackers is accuracy, since a device is not completely accurate. Moreover, the *Midas Touch* problem occurs when a user only wants to fixate an element with the gaze but does not intend to select it. Since the tracker does not know whether a user just wants to investigate or whether they want to interact with an element, false inputs can occur. The eye tracker latency and limitations caused by the user (contact lenses can for example prevent the user from successfully using the eye tracker in the first place) are two more problems that might occur. The characteristics of eye movements can also interfere with the eye tracking gaze data, since the human eye never remains completely still.

In chapter 4, the implementation background and realization of the eye tracking system have been explained in more detail. Therefore, the overall concept of this implementation was introduced. Three interaction methods have been implemented. Here, the choice was made for two gaze-only interaction methods and one technique where the gaze in combination with a second input modality triggers selections. Both variants were chosen since the comparison of them could produce interesting results. *Dwell Time*, where an element needs to be fixated for a given period of time, seems to be the most widespread input method considering eye tracking based input. Therefore, *Dwell Time* was the first choice that was implemented. As second gaze-only interaction method, *Context Switching* was chosen, since this is a rather unknown and not thoroughly widespread variant. Here, a two-step gesture (from the element to a given point and back) has to be performed with the eyes to select an element. For gaze in combination with a second modality, a key press is used to select a fixated element. This method is called *Key Press* in the following. Since (nearly) all users worked with a keyboard before, no extra learning was required. Furthermore, the other combinations utilizing different input devices turned out to be impractical since further devices would have been necessary (for example a touch enabled display).

The user interface design was kept simple. A questionnaire, the users have to complete, is shown. For the study, the elements of the questionnaire which should have been selected by the participants were marked green, all others were marked red. By predetermining the answers in the questionnaire, the actual time a participant needed to fulfill a task with the given eye tracking method can be tracked more easily. The participants did not have to think about the answers, so all tasks could theoretically be done in the same time. Moreover, the user's position relative to the eye tracker is illustrated in a so called "trackbox". This trackbox is not only shown at the beginning but also during the study. If the position of a proband changes, they can see this information in the trackbox and can relocate in front of the tracker to be tracked correctly again. Additionally, a light blue dot indicates the user's gaze point, similar to a mouse pointer.

When it comes to the choice which tracker will be used, the different advantages and disadvantages of several tracking devices have been compared. In this work, a *video-oculography* eye tracking device has been used, the so called *EyeTribe*. This tracker has a server that is running in the background which is providing gaze data. Further on, a calibration is obligatory to produce relevant gaze data. During the calibration process,

different results from perfect to poor can be produced. The user was only able to start the study when they performed good enough to achieve a good or perfect result.
Finally, the study implementation was explained in further detail. Here, a .NET Framework 3.5 Windows application has been developed. With the known position of the user's gaze and the elements on the screen, it can be calculated whether a user fixates an element or not. To prevent losing focus during tracking when small jittery eye movements occur, the focus of an element only changes when the gaze has been registered on another element/screen area for at least 100 ms. When it comes to *Dwell Time*, the duration the element has to be fixated until it is selected is 750 ms. This duration has been chosen since this time span seemed to be right for both, novice and experienced users, according to the related work. Considering *Context Switching*, the user has to focus an element so that a rectangle appears above or next to this element. When the user fixates this rectangle and subsequently moves their point of gaze back to the initial element, all within 3000 ms each, the element was selected.

Chapter 5 deals with the study in general. Here, the study preparation is summarized. A study technique without interferences from the observer has been chosen. The users had to complete two questionnaires and a logfile has been filled with objective data while they were doing so. As a consequence, the subsequent study analysis was backed by both objective as well as subjective data.

The results of the study are illustrated in chapter 6. All in all it can be said that *Key Press* did best compared to the other two interaction methods. Here, nearly half of all participants would use the system with this interaction method without further changes. Approximately one out of five probands would use the system after further changes. However, roughly 35 percent would not use the introduced system at all. Furthermore, *Context Switching* scored especially low. It was ranked worst by the participants and no positive aspects have been recorded (for this interaction method). This is also reflected in the fact that nobody would use the system with *Context Switching* and only approximately one out of five probands would use the system after further changes with *Context Switching*. Although *Dwell Time* performed better, most probands would not

use the system (without changes) with this input method, either. This estimation was supported by the fact that nearly all of the probands would prefer *Key Press* as an input method, as well. Moreover, this technique was rated best of all three interaction methods and only one negative aspect was mentioned by two subjects.
This subjective estimation of the participants is emphasized by the findings evaluated from the logfiles. To complete the task with *Key Press*, 22 seconds where needed while 79 seconds were required to fulfill the task with *Dwell Time* and the completion even took 104 seconds with *Context Switching*. Additionally, the participants only produced 0.4 faulty inputs on average while using *Key Press* as an interaction method. It can be said that the subjects performed well with *Context Switching* considering errors. Less than one faulty input per proband (0.9 faulty inputs) was made. Compared to that, the number of faulty inputs produced while using *Dwell Time* was significantly higher with 3.5 faulty inputs.

When it comes to the question whether such a system is useful to complete a questionnaire, the subjects were able to rate the usage in general and the usage of the specific elements used in the questionnaire. The system was rated as not useful to complete a questionnaire by half of the subjects while five out of fourteen probands would use the system for this purpose. Especially with *Key Press*, the system can be an enrichment for users. Here, the required time for executing interactions lays between two and five seconds. Compared to that the other two input methods have been significantly slower when it comes to text input (42 seconds with *Dwell Time* and 62 seconds with *Context Switching*). It seems that especially complex questionnaire elements (like text input) should be dismissed when it comes to the completion of questionnaires with either *Dwell Time* or *Context Switching*.

All in all, the overall eye tracking experience needs to be improved to achieve better evaluation results. The exhaustion of the eyes seems to play a significant role when it comes to the poorer ratings by the participants. Especially the simplification or the complete exclusion of the calibration process could reduce the exhaustion of the eyes. Furthermore, (too) long interaction durations should be avoided.

7.2 Outlook

In the view of future work, the usability of the implemented methods should further be evaluated. Especially when it comes to the gaze-only input methods, further improvements seem necessary. Here, *Context Switching* seems to be too complex which expedites the exhaustion of the eyes. This method could be simplified if the appearing blue rectangles could be selected more easily. The surrounding elements could, for example, not be focusable. In so doing, the rectangle could be selected more easily since the gaze could also pass by other elements but the rectangle would not disappear. Furthermore, some elements are difficult to interact with when it comes to gaze-only input methods. The best example of this is the text input. Here, the purpose of several research works is to find good solutions for this problem, like for example *Quickpie* which was further explained in section 2.1.1. So, another method than just typing with the help of an on-screen keyboard should be taken into consideration. The disadvantage of *Key Press* is the need for a keyboard. Thus, this method is not hands-free. Here, other interaction methods could be implemented. The combination of voice recognition and gaze seems to be a potential candidate to achieve good evaluation results.
Furthermore, the calibration process can be exhausting for the eyes. A definitive field for future work is the simplification of the calibration process. This also comprises to dismiss the calibration process completely. In so doing, relative eye movements could be tracked. So, for example gaze gesture could be used as input method. Considering a limited number of interactions, a mapping can be created between said interactions and respective gaze gestures. If the user performs a gaze gesture, the system would be able to recognize this gesture and perform a specific interaction. In so doing, no calibration would be necessary since the system does not need to know the exact point of view. Nevertheless, an implementation of this approach for the purpose of the completion of a questionnaires could be difficult. What mapping would be realized if a user wants, for example, select a specific checkbox in a questionnaire. Another approach is to integrate the calibration but to hide it from the user. In so doing, the calibration is processed without the user's notice.
Moreover, more freedom of movement should be tolerated when interacting with eye trackers in the future. Since the research related to eye tracking is ongoing, there is

hope for improved hardware and more elaborate software solutions to achieve the goal of a precise but also permissive eye tracking experience.

List of Figures

Acronyms

API Application Programming Interface. 44, 45, 49

JSON JavaScript Object Notation. 45, 46

SDK Software Development Kit. 30, 43–46, 48, 49

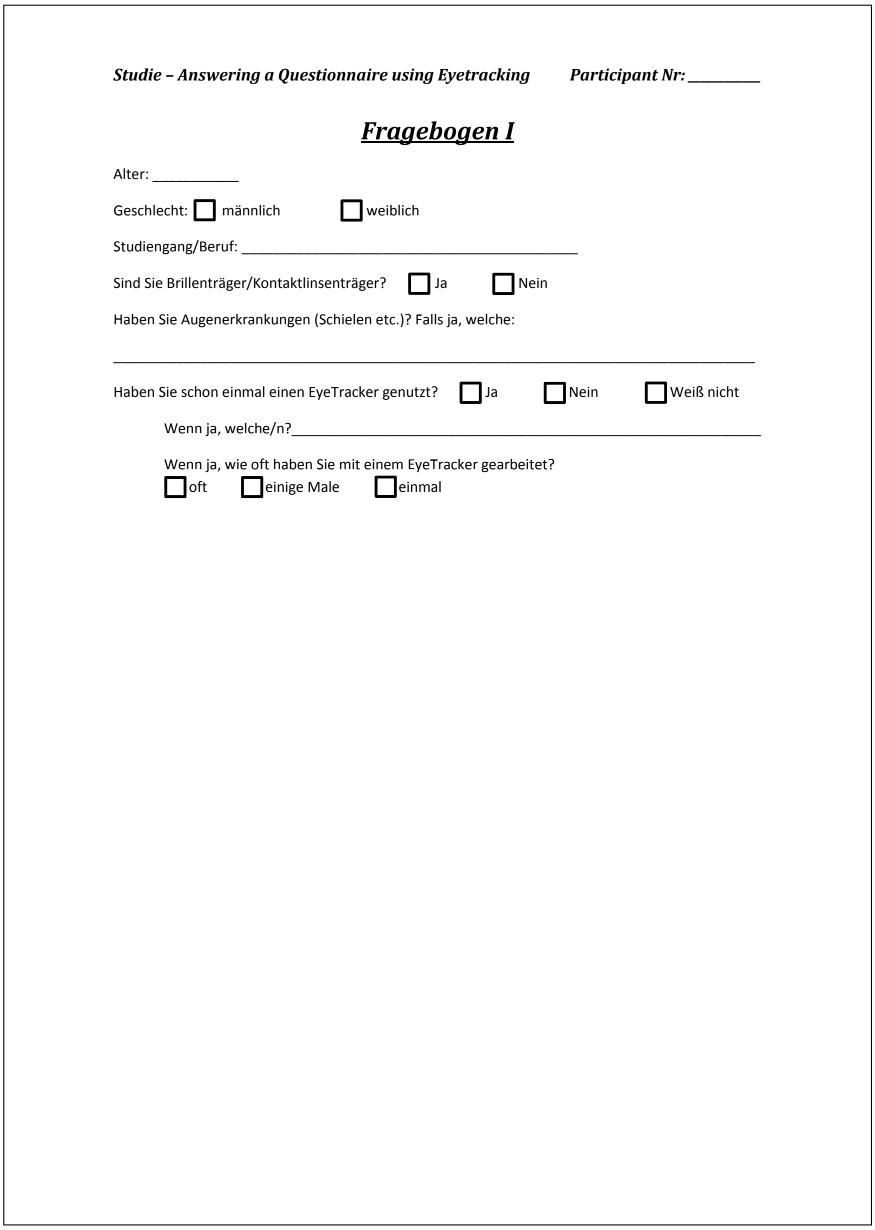
TCP Transmission Control Protocol. 45

UI User Interface. 1, 12, 34

WPF Windows Presentation Foundation. 38, 44

Appendix

1 Questionnaire I - Demographic Data



Studie – Answering a Questionnaire using Eyetracking ***Participant Nr:*** _________

Fragebogen I

Alter: ___________

Geschlecht: ☐ männlich ☐ weiblich

Studiengang/Beruf: ____

Sind Sie Brillenträger/Kontaktlinsenträger? ☐ Ja ☐ Nein

Haben Sie Augenerkrankungen (Schielen etc.)? Falls ja, welche:

____

Haben Sie schon einmal einen EyeTracker genutzt? ☐ Ja ☐ Nein ☐ Weiß nicht

Wenn ja, welche/n?____

Wenn ja, wie oft haben Sie mit einem EyeTracker gearbeitet?
☐ oft ☐ einige Male ☐ einmal

2 Questionnaire II - Study Results

Studie – Answering a Questionnaire using Eyetracking ***Participant Nr:*** ________

Fragebogen II

Was hat Ihnen an der Nutzung des Eyetrackers allgemein gefallen?

- ☐ Intuitiv
- ☐ Schnell erlernbar
- ☐ Kurze Interaktionsdauer (im Vgl. zur Mauseingabe)
- ☐ Gutes Tracking (Blickpunkt wird erkannt etc.)
- ☐ Sonstiges: ____

Was hat Ihnen an der Nutzung des Eyetrackers allgemein nicht gefallen?

- ☐ Zu viele fehlerhafte Eingaben
- ☐ Schwer zu erlernen
- ☐ Lange Interaktionsdauer (im Vgl. zur Mauseingabe)
- ☐ Schlechtes Tracking (Blickpunkt zu ungenau etc.)
- ☐ Ermüdend/anstrengend für die Augen
- ☐ Kalibrierung anstrengend/schwierig
- ☐ Bewegungseinschränkung (Kopf still halten)
- ☐ Sonstiges: ____

Als wie anstrengend empfanden Sie die Kalibrierung für die Augen?

☐ Nicht anstrengend ☐ Mittelmäßig ☐ Anstrengend

Was hat Ihnen an der Interaktionsvariante "Context Switching" gefallen?

- ☐ Intuitiv
- ☐ Schnell erlernbar
- ☐ Kurze Interaktionsdauer (im Vgl. zu anderen Varianten ("Dwell Time" & "Key Press"))
- ☐ Sonstiges: ____

1

Studie – Answering a Questionnaire using Eyetracking **Participant Nr: _________**

Was hat Ihnen an der Interaktionsvariante "Context Switching" nicht gefallen?

- ☐ Zu viele fehlerhafte Eingaben
- ☐ Schwer zu erlernen
- ☐ Lange Interaktionsdauer (im Vgl. zu anderen Varianten ("Dwell Time" & "Key Press"))
- ☐ Ermüdend/anstrengend für die Augen
- ☐ Sonstiges: ____

Was hat Ihnen an der Interaktionsvariante "Dwell Time" gefallen?

- ☐ Intuitiv
- ☐ Schnell erlernbar
- ☐ Kurze Interaktionsdauer (im Vgl. zu anderen Varianten ("Context Switching" & "Key Press"))
- ☐ Sonstiges: ____

Was hat Ihnen an der Interaktionsvariante "Dwell Time" nicht gefallen?

- ☐ Zu viele fehlerhafte Eingaben
- ☐ Schwer zu erlernen
- ☐ Lange Interaktionsdauer (im Vgl. zu anderen Varianten ("Context Switching" & "Key Press"))
- ☐ Ermüdend/anstrengend für die Augen
- ☐ Sonstiges: ____

Was hat Ihnen an der Interaktionsvariante "Key Press" gefallen?

- ☐ Intuitiv
- ☐ Schnell erlernbar
- ☐ Kurze Interaktionsdauer (im Vgl. zu anderen Varianten ("Dwell Time" & "Key Press"))
- ☐ Sonstiges: ____

2

Studie – Answering a Questionnaire using Eyetracking ***Participant Nr: ________***

Was hat Ihnen an der Interaktionsvariante "Key Press" nicht gefallen?

- ☐ Zu viele fehlerhafte Eingaben
- ☐ Schwer zu erlernen
- ☐ Lange Interaktionsdauer (im Vgl. zu anderen Varianten ("Dwell Time" & "Key Press"))
- ☐ Ermüdend/anstrengend für die Augen
- ☐ Sonstiges: ____
____

Welche der drei Varianten würden Sie bevorzugen?

☐ Context Switching ☐ Dwell Time ☐ Key Press ☐ Alle gleich

Würden Sie dieses System auch privat nutzen (mit "Context Switching")?

☐ Ja ☐ Nein ☐ Nach Änderungen

Begründen Sie: ____
____

Falls Sie 'Nach Änderungen' angekreuzt haben, was müsste geändert werden?

- ☐ Weniger fehlerhafte Eingaben
- ☐ Kürzere Interaktionsdauer
- ☐ Ohne Texteingabe
- ☐ Ohne Slider
- ☐ Ohne einfache Komponenten wie Checkbox/Radiobutton/Button
- ☐ Mit mehr Bewegungsfreiheit
- ☐ Sonstiges: ____
____

3

Studie – Answering a Questionnaire using Eyetracking ***Participant Nr:_________***

Würden Sie dieses System auch privat nutzen (mit "Dwell Time")?

☐ Ja ☐ Nein ☐ Nach Änderungen

Begründen Sie: ____

____

Falls Sie 'Nach Änderungen' angekreuzt haben, was müsste geändert werden?

☐ Weniger fehlerhafte Eingaben

☐ Kürzere Interaktionsdauer

☐ Ohne Texteingabe

☐ Ohne Slider

☐ Ohne einfache Komponenten wie Checkbox/Radiobutton/Button

☐ Mit mehr Bewegungsfreiheit

☐ Sonstiges: ____

____

Würden Sie dieses System auch privat nutzen (mit "Key Press")?

☐ Ja ☐ Nein ☐ Nach Änderungen

Begründen Sie: ____

____

Falls Sie 'Nach Änderungen' angekreuzt haben, was müsste geändert werden?

☐ Weniger fehlerhafte Eingaben

☐ Kürzere Interaktionsdauer

☐ Ohne Texteingabe

☐ Ohne Slider

☐ Ohne einfache Komponenten wie Checkbox/Radiobutton/Button

☐ Mit mehr Bewegungsfreiheit

☐ Sonstiges: ____

____

4

Studie – Answering a Questionnaire using Eyetracking ***Participant Nr:*** _________

Finden Sie ein solches System speziell für das Ausfüllen eines Fragebogens sinnvoll?

☐ Ja ☐ Nein ☐ Nach Änderungen

Begründen Sie: ____

____

Falls Sie ‚Nach Änderungen' angekreuzt haben, was müsste geändert werden?

- ☐ Weniger fehlerhafte Eingaben
- ☐ Kürzere Interaktionsdauer
- ☐ Ohne Texteingabe
- ☐ Ohne Slider
- ☐ Ohne einfache Komponenten wie Checkbox/Radiobutton/Button
- ☐ Sonstiges: ____

____

Wie würden Sie die einzelnen Komponenten des Fragebogens einschätzen (bei "Context Switching")?

	Gut nutzbar	Mittelmäßig	Schlecht nutzbar
Texteingabe	☐	☐	☐
Slider	☐	☐	☐
Einfache Komponenten (Checkbox, Radiobutton, Button)	☐	☐	☐

Wie würden Sie die einzelnen Komponenten des Fragebogens einschätzen (bei "Dwell Time")?

	Gut nutzbar	Mittelmäßig	Schlecht nutzbar
Texteingabe	☐	☐	☐
Slider	☐	☐	☐
Einfache Komponenten (Checkbox, Radiobutton, Button)	☐	☐	☐

5

Studie – Answering a Questionnaire using Eyetracking ***Participant Nr:*** ________

Wie würden Sie die einzelnen Komponenten des Fragebogens einschätzen (bei "Key Press")?

	Gut nutzbar	Mittelmäßig	Schlecht nutzbar
Texteingabe	☐	☐	☐
Slider	☐	☐	☐
Einfache Komponenten (Checkbox, Radiobutton, Button)	☐	☐	☐

Wie empfinden Sie die Größe der Elemente allgemein im Hinblick auf das EyeTracking?

	Zu klein	Genau richtig	Zu groß
Textfeld (öffnen der On-Screen Tastatur)	☐	☐	☐
Buchstaben bei der Texteingabe	☐	☐	☐
Checkbox/Radiobutton	☐	☐	☐
Slider	☐	☐	☐
Done-Button	☐	☐	☐

Wie empfinden Sie die Dauer der Dwell Time?

☐ Zu kurz ☐ Genau richtig ☐ Zu lang

Wie empfinden Sie die Dauer der Anzeige der blauen Rechtecke beim Context Switching?

☐ Zu kurz ☐ Genau richtig ☐ Zu lang

Wie empfinden Sie die Größe der Rechtecke beim Context Switching?

	Zu klein	Genau richtig	Zu groß
Textfeld (öffnen der On-Screen Tastatur)	☐	☐	☐
Buchstaben bei der Texteingabe	☐	☐	☐
Checkbox/Radiobutton	☐	☐	☐
Slider	☐	☐	☐
Done-Button	☐	☐	☐

6

Studie – Answering a Questionnaire using Eyetracking **Participant Nr:** _________

Weitere positive Anmerkungen:

____

____

____

Weitere negative Anmerkungen:

____

____

____

Sonstiges:

____

____

____

7

Bibliography

[Eyea] *API Reference*. URL: http://dev.theeyetribe.com/api/ (visited on 12/22/2015).

[Eyeb] *Basics*. URL: http://dev.theeyetribe.com/general/ (visited on 12/22/2015).

[Eyec] *C# samples for The EyeTribe Tracker*. URL: https://github.com/EyeTribe/tet-csharp-samples (visited on 12/22/2015).

[Eyed] *C# SDK for the EyeTribe Dev Kit*. URL: https://github.com/EyeTribe/tet-csharp-client (visited on 12/22/2015).

[CSS14] Tuhin Chakraborty, Sayan Sarcar, and Debasis Samanta. "Design and Evaluation of a Dwell-free Eye Typing Technique." In: *Proceedings of the Extended Abstracts of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*. CHI EA '14. Toronto, Ontario, Canada: ACM, 2014, pp. 1573–1578. ISBN: 978-1-4503-2474-8.

[Eyee] *Developers Guide*. URL: http://dev.theeyetribe.com/dev/ (visited on 12/22/2015).

[DDLS07] Heiko Drewes, Alexander De Luca, and Albrecht Schmidt. "Eye-gaze Interaction for Mobile Phones." In: *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology*. Mobility '07. Singapore: ACM, 2007, pp. 364–371. ISBN: 978-1-59593-819-0.

[Duc07] Andrew T. Duchowski. *Eye Tracking Methodology: Theory and Practice*. Springer London, 2007. ISBN: 9781846286087.

[DCM04] Andrew T. Duchowski, Nathan Cournia, and Hunter Murphy. “Gaze-Contingent Displays: Review and Current Trends.” In: Adaptive Display Conference 2004. 2004.

[Gei+14] Philip Geiger et al. “Location-based Mobile Augmented Reality Applications: Challenges, Examples, Lessons Learned.” In: *10th Int'l Conference on Web Information Systems and Technologies (WEBIST 2014), Special Session on Business Apps*. 2014, pp. 383–394.

[Eyef] *Getting Started*. URL: http://dev.theeyetribe.com/start/ (visited on 12/22/2015).

[Gom+07] Roger P.G. van Gompel et al. *Eye Movements: A Window on Mind and Brain*. Educational psychology. Elsevier Science, 2007. ISBN: 9780080449807.

[Hat+96] F. Hatfield et al. “Principles and guidelines for the design of eye/voice interaction dialogs.” In: *Human Interaction with Complex Systems, 1996. HICS '96. Proceedings., Third Annual Symposium on*. 1996, pp. 10–19.

[HR12] Henna Heikkilä and Kari-Jouko Räihä. “Simple Gaze Gestures and the Closure of the Eyes As an Interaction Technique.” In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. ETRA '12. Santa Barbara, California: ACM, 2012, pp. 147–154. ISBN: 978-1-4503-1221-9.

[Hol+11] Kenneth Holmqvist et al. *Eye Tracking: A Comprehensive Guide to Methods and Measures*. Oxford University Press, 2011. ISBN: 9780199697083.

[Insa] SensoMotoric Instruments. *3D Vog Video-Oculography System*. URL: http://www.smivision.com/en/gaze-and-eye-tracking-systems/products/3d-vog.html (visited on 12/09/2015).

[Insb] SensoMotoric Instruments. *SMI Eye Tracking Glasses 2 Wireless. Mobile eye tracking made easy, robust, efficient and versatile*. URL: http://www.eyetracking-glasses.com/fileadmin/user_upload/

documents / smi _ etg2w _ flyer _ naturalgaze . pdf (visited on 12/22/2015).

[Insc] SensoMotoric Instruments. *SMI Eye Tracking Glasses 2 Wireless - Easy, Robust and Efficient*. URL: http://www.eyetracking-glasses.com/products/eye-tracking-glasses-2-wireless/technology/ (visited on 12/22/2015).

[Jac90] Robert J. K. Jacob. "What You Look at is What You Get: Eye Movement-based Interaction Techniques." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '90. Seattle, Washington, USA: ACM, 1990, pp. 11–18. ISBN: 0-201-50932-6.

[JK03] Robert J.K. Jacob and Keith S. Karn. "Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises." In: *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*. Ed. by Radach Hyona and Deubel. Elsevier Science, 2003, pp. 573 –605. ISBN: 978-0-444-51020-4.

[KPW07] Manu Kumar, Andreas Paepcke, and Terry Winograd. "EyePoint: Practical Pointing and Selection Using Gaze and Keyboard." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA: ACM, 2007, pp. 421–430. ISBN: 978-1-59593-593-9.

[Kum+08] Manu Kumar et al. "Improving the Accuracy of Gaze Input for Interaction." In: *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*. ETRA '08. Savannah, Georgia: ACM, 2008, pp. 65–68. ISBN: 978-1-59593-982-1.

[Eyeg] *Language Tutorial - C#*. URL: http : / / dev . theeyetribe . com / csharp/ (visited on 12/22/2015).

[Lan00] Chris Lankford. "Effective Eye-gaze Input into Windows." In: *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*. ETRA '00. Palm Beach Gardens, Florida, USA: ACM, 2000, pp. 23–27. ISBN: 1-58113-280-8.

[LGE11] Simon P. Liversedge, Iain D. Gilchrist, and Stefan Everling. *The Oxford Handbook of Eye Movements*. Oxford Library of Psychology. OUP Oxford, 2011. ISBN: 9780199539789.

[Ltd] SR Research Ltd. *EyeLink II*. http://www.eyelinkinfo.com/eyelinkII.html and http://www.eyelinkinfo.com/pdf/elII_table.pdf. (Visited on 12/09/2015).

[MAR04] Päivi Majaranta, Anne Aula, and Kari-Jouko Räihä. "Effects of Feedback on Eye Typing with a Short Dwell Time." In: *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*. ETRA '04. San Antonio, Texas: ACM, 2004, pp. 139–146. ISBN: 1-58113-825-3.

[Maj+12] Pälvi Majaranta et al. *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies*. Premier reference source. Medical Information Science Reference, 2012. ISBN: 9781613500989.

[MB14] Päivi Majaranta and Andreas Bulling. "Eye Tracking and Eye-Based Human–Computer Interaction." English. In: *Advances in Physiological Computing*. Ed. by Stephen H. Fairclough and Kiel Gilleade. Human–Computer Interaction Series. Springer London, 2014, pp. 39–65. ISBN: 978-1-4471-6391-6.

[MF06] Hiroyuki Manabe and Masaaki Fukumoto. "Full-time Wearable Headphone-type Gaze Detector." In: *CHI '06 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '06. Montreal, Quebec, Canada: ACM, 2006, pp. 1073–1078. ISBN: 1-59593-298-4.

[NP10] Jakob Nielsen and Kara Pernice. *Eyetracking Web Usability*. Voices That Matter. New Riders, 2010. ISBN: 9780321498366.

[PRS14] Pawan Patidar, Himanshu Raghuvanshi, and Sayan Sarcar. "Quickpie: An Interface for Fast and Accurate Eye Gazed based Text Entry." In: *arXiv preprint arXiv:1407.7313* (2014).

[Pfe+14] Ken Pfeuffer et al. “Gaze-touch: Combining Gaze with Multi-touch for Interaction on the Same Surface.” In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST ’14. Honolulu, Hawaii, USA: ACM, 2014, pp. 509–518. ISBN: 978-1-4503-3069-5.

[Pop+12] Stanislav Popelka et al. “Advanced Map Optimalization Based on Eye-tracking.” In: *Cartography: A tool for spatial analysis*. 2012.

[PT08] Marco Porta and Matteo Turina. “Eye-S: A Full-screen Input Modality for Pure Eye-based Communication.” In: *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*. ETRA ’08. Savannah, Georgia: ACM, 2008, pp. 27–34. ISBN: 978-1-59593-982-1.

[Resa] Arrington Research. *SceneCamera Eye Tracking*. URL: http://www.arringtonresearch.com/scene.html (visited on 12/22/2015).

[Resb] Arrington Research. *SceneCamera Eye Tracking*. URL: http://www.arringtonresearch.com/scenetechinfo.html (visited on 12/22/2015).

[RP08] Prof. Wolfram E. Rossig and Prof. Dr. Joachim Prätsch. *Wissenschaftliche Arbeiten. Leitfaden für Haus- und Seminararbeiten, Bachelor- und Masterarbeiten, Diplom- und Magisterarbeiten, Dissertationen*. Ed. by BerlinDruck. 7th ed. Achim, 2008.

[Rud] Tara Rudnicki. *Eye-control Empowers People with Disabilities*. URL: http://www.abilities.com/community/assistive_eye-control.html (visited on 02/14/2016).

[Sch+15a] Marc Schickler et al. “An Engine Enabling Location-based Mobile Augmented Reality Applications.” In: *10th International Conference on Web Information Systems and Technologies (Revised Selected Papers)*. LNBIP 226. Springer, 2015, pp. 363–378.

[Sch+15b] Marc Schickler et al. *Entwicklung mobiler Apps: Konzepte, Anwendungsbausteine und Werkzeuge im Business und E-Health*. Springer-Verlag Berlin Heidelberg, 2015. ISBN: 9783642330575.

[Sch+15c] Johannes Schobel et al. "Process-Driven Data Collection with Smart Mobile Devices." In: *10th International Conference on Web Information Systems and Technologies (Revised Selected Papers)*. LNBIP 226. Springer, 2015, pp. 347–362.

[Sch+14] Johannes Schobel et al. "Towards Process-Driven Mobile Data Collection Applications: Requirements, Challenges, Lessons Learned." In: *10th Int'l Conference on Web Information Systems and Technologies (WEBIST 2014), Special Session on Business Apps*. 2014, pp. 371–382.

[Sch+13] Johannes Schobel et al. "Using Vital Sensors in Mobile Healthcare Business Applications: Challenges, Examples, Lessons Learned." In: *9th Int'l Conference on Web Information Systems and Technologies (WEBIST 2013), Special Session on Business Apps*. 2013, pp. 509–518.

[SD12] Sophie Stellmach and Raimund Dachselt. "Look & Touch: Gaze-supported Target Acquisition." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 2981–2990. ISBN: 978-1-4503-1015-4.

[Sun12] Veronica Sundstedt. *Gazing at Games: An Introduction to Eye Tracking Control*. Synthesis digital library of engineering and computer science. Morgan & Claypool, 2012. ISBN: 9781608456499.

[Sysa] EyeTech Digital Systems. URL: `http://www.eyetechds.com/tm5-mini-assistive-tech.html` (visited on 12/22/2015).

[Sysb] EyeTech Digital Systems. URL: `http://www.eyetechds.com/specialty-hardware.html` (visited on 12/22/2015).

[Sysc] EyeTech Digital Systems. *QuickLink. Stream the raw eye data into your own application*. URL: `http://www.eyetechds.com/quick-link-sdk.html` (visited on 12/22/2015).

[Teca] LC Technolgies. *Communicate with the world using the power of your eyes*. URL: `http://www.eyegaze.com/eye-tracking-assistive-technology-device/` (visited on 02/14/2016).

[Tecb] LC Technolgies. *Eye Tracking for Research and Analysis*. URL: http://www.eyegaze.com/eye-tracking-for-research-and-human-computer-interaction/ (visited on 12/22/2015).

[Tecc] LC Technolgies. *Eye Tracking for Research and Analysis*. URL: http://eyegaze.com/wp-content/uploads/LCT%20Product%20Brochure%202012.pdf (visited on 12/22/2015).

[TAa] authorTobii Technology AB. URL: http://www.tobiipro.com/siteassets/tobii-pro/product-descriptions/tobii-pro-x3-120-product-description.pdf (visited on 12/22/2015).

[TAb] authorTobii Technology AB. http://developer.tobii.com/differences-between-tobii-gaze-sdk-and-tobii-eyex-sdk/ and http://developer.tobii.com/rex-technical-specs-faq/. (Visited on 12/22/2015).

[TAc] authorTobii Technology AB. URL: http://www.tobiipro.com/learn-and-support/learn/how-do-tobii-eye-trackers-work/eye-tracker-timing-performance/ (visited on 12/22/2015).

[TAd] authorTobii Technology AB. *The Tobii EyeX SDK for C and C++*. URL: http://developer.tobii.com/eyex-sdk/c-cplusplus/ (visited on 12/22/2015).

[TAe] authorTobii Technology AB. *Tobii Pro X3-120*. URL: http://www.tobiipro.com/product-listing/tobii-pro-x3-120/ (visited on 12/22/2015).

[TAf] authorTobii Technology AB. *tobii pro/glasses 2*. URL: http://www.tobiipro.com/product-listing/tobii-pro-glasses-2/ (visited on 12/22/2015).

[TAg] authorTobii Technology AB. *tobii pro/glasses 2. The world's most innovative wearable eye tracking system for real-world research.* URL: http://www.tobiipro.com/siteassets/tobii-pro/brochures/tobii-pro-glasses-2-brochure.pdf (visited on 12/22/2015).

[Eyeh] *The EyeTribe Tracker*. URL: https://theeyetribe.com/products/ (visited on 12/22/2015).

[TM14] Antonio Diaz Tula and Carlos H. Morimoto. "Meta-keys: Extending the Functionality of Gaze-based Interaction." In: *Proceedings of the 13th Brazilian Symposium on Human Factors in Computing Systems*. IHC '14. Foz do Iguacu, Brazil: Sociedade Brasileira de Computação, 2014, pp. 285–292. ISBN: 978-85-7669-291-1.

[WT05] Nicholas J. Wade and Benjamin W. Tatler. *The Moving Tablet of the Eye: The Origins of Modern Eye Movement Research*. Oxford University Press, 2005. ISBN: 9780198566168, 9780198566175.

[Wpf] *Windows Presentation Foundation*. URL: https://en.wikipedia.org/wiki/Windows_Presentation_Foundation (visited on 01/13/2016).

[Yag10] Tohru Yagi. "Eye-gaze Interfaces Using Electro-oculography (EOG)." In: *Proceedings of the 2010 Workshop on Eye Gaze in Intelligent Human Machine Interaction*. EGIHMI '10. Hong Kong, China: ACM, 2010, pp. 28–32. ISBN: 978-1-60558-999-2.

Name: Laura Irlinger Matrikelnummer: 741748

Erklärung

Ich erkläre, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den 26.02.2016 ....

Laura Irlinger