Saarland University

Masters Thesis

Empirical Analysis of Selection Based Text Entry in Virtual Reality

Author:
Pascal Ziegler

Advisors:
Marco Speicher
Anna Feit
Supervisor:
Prof. Dr. Antonio Krüger
Reviewer:
Prof. Dr. Jürgen Steimle

A thesis submitted in fulfillment of the requirements for the degree of Master of Science Media Informatics

SAARLAND UNIVERSITY

June 2017

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis

Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken,________________________ ________________________________

(Datum/Date) (Unterschrift/Signature)

Declaration of Authorship

I, Pascal Ziegler, declare that this thesis titled, “Empirical Analysis of Selection Based Text Entry in Virtual Reality” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ________________________________

Date: ________________________________

"The good news is that virtual reality is here. The bad news is that something is still missing." [1]

Mychilo Cline

SAARLAND UNIVERSITY

Abstract

Faculty of Media Informatics
DFKI Saarbrücken

Master of Science Media Informatics

Selective Text Input in VR

by Pascal Ziegler

This thesis provides a list and classification of existing text entry methods for VR to this date and ones that could potentially be used in VR. It spans the design space for text entry in VR including non-evaluated variables. After that six QWERTY based text entry methods (Pointing, Head Gaze, Freehand, Pen Based, Touch Pad and Discrete Touch Pad) are implemented for the HTC Vive. They were presented to 14 users in a primary informal study followed by the main empirical study to explore their usability and user experience against criteria regarding text entry and VR: Words per Minute (WPM), Keystrokes per Character (KSPC), error rate, immersion, motion sickness, mental and physical demand (NASA task load index), as well as user experience.
Pointing with a tracked hand held controller turned out to be the "best" method out of these regarding nearly all tested measures. Those measures where Pointing was not the "best" method showed no significant differences between the methods. It also turned out that the method had no significant effect on immersion or motion sickness. Based on the findings of the study open questions of the design space are answered, older design rules are updated and possible future work is proposed.
This work allows developers of VR applications to choose an appropriate text entry method depending on the available hardware and kind of application. This is achieved through justified design guidelines.

Acknowledgements

I would like to thank the staff of DFKI Saarbrücken for their time and effort to make it possible for me to experiment in their laboratory;
Of course Marco Speicher and Anna Feit, my advisors for their support, wise advice, groundwork and patience, that went far beyond their obligation as well as my reviewers Prof. Dr. Antonio Krüger and Prof. Dr. Jürgen Steimle;
Jonathan Dawo for his good english;
And last but not least Alyona Morozova, Nauman Mughal, Dilip Rajkumar, Maike Maas, Katharina Feit, Dominik Scholtes, Julian Koch, Christoph Rosenberg, Achim Koch, Christoph Kreutzer, Nico Fries, Jill Els, Robin Wark, Tobias Weber, Michael Groß, Helena Rudolph, Lakhan Rathi, Kevin Klesen and David Mrochen;

Contents

Chapter 1

Introduction

"Symbolic communication - the use of abstract symbols to represent objects, ideas, concepts, quantities, and the like- is one of the crowning achievements of human civilization. It allows us to provide and obtain information precisely and concisely; it allows information to be persistent; it provides for methods of thought not possible without symbols. Imagine what it would be like to live in a world without langauge or mathematics. Symbolic communication pervades everything we do.
It is strange, then, that symbolic communication in most Virtual Reality applications neither present nor accept symbolic information; rather, they present a purely geometric and visual world for the user to perceive. Of the 3D applications that do include symbolic information, most provide symbolic output only. That is text, numbers, or speech are embedded in the environment. This might take the form of labels on virtual buttons, a legend on a map, a numeric display of the user's coordinates, or audio help, to name a few examples. Symbolic input, however, is rarely present."[2]
This is taken out of the book "3D User Interfaces: Theory and Practice" ([2]) by Bowman et al. from 2004. More than twelve years ago. Still not much has changed yet. There exist a few text entry methods for commercial products today but they are not heavily used in applications. Bowman et al. claim that a lack of symbolic input in VR interfaces does not mean that the task is unimportant for 3D applications, but rather that usable and efficient techniques for this task are difficult to design and implement, which has caused developers to largely avoid the issue [2].
With this work we want to lower this difficulty by giving developers rules and aides with to choose specific text input methods for specific circumstances and to clarify what is to be considered during development and implementation.

Over time and throughout the three waves of virtual reality there have been many attempts to define the term ”virtual reality” and to separate it from related fields. It dates back to the German philosopher Immanuel Kant [3], although its use did not involve technology. Its modern use was popularized by Jaron Lanier in the 1980s[4]. This thesis will follow the definition of S. M. LaValle for various reasons: He provides definitions for the terms ”Awareness” the contrary to ”Presence” or ”Immersion”. His definition also separates the term ”Virtual Reality” (henceforth VR) from ”Augmented Reality” (AR) which ”refers to systems in which most of the visual stimuli are propagated directly through glass or cameras to the eyes, and some additional structures appear to be superimposed onto the user's world” [4]. This difference is especially important for this thesis, because it introduces the restriction that the user can not see his or her body (especially his or her hands) directly.

1.1 Why Text Entry in VR

Bowman et al. [2] provide a set of scenarios for the use of text entry in VR: Design annotation (for example 3D artists or architects), filename entry, labeling, precise object manipulation, parameter setting, communication between users and markup [2]. Another scenario that can be added would be coding so developers don't need to frequently switch between the virtual and real world causing increased cybersickness.
Symbolic input techniques for 3D UIs or VR will all be necessarily different from traditional techniques (keyboards) because of the inherent differences between 3D (non-desktop) and 2D UIs. Simply put, traditional keyboards don't work in non-desktop 3D UIs because users are often standing, users may physically move about, there is usually no surface on which to place a keyboard and it may be difficult or impossible to see a keyboard when the user's vision is occluded (e.g., in an HMD) [2].
An aid to the designer is the wealth of information that already exists on symbolic input in nontraditional computing environments [2]. Designers of wearable computers, palmtop computers, PDAs, cell phones, Augmented Reality devices, etc. have had to tackle these issues [2]. Although the usability of some of the symbolic input techniques for these devices is questionable, these domains are an important source of ideas for 3D UI symbolic input [2].

There exist some studies that compare text entry methods for VR ([5], [6]), but they only compare them regarding performance measures like Words per Minute and error rate. They are missing important factors for VR and text entry like motion sickness and immersion as well as user experience. Plus the third VR wave has brought many new commercial products and input hardware that need to be reconsidered. The problem stated above, that text entry in VR is probably avoided mainly because it is so hard to design, could be addressed by giving clear design rules that are adapted to the modern VR world. This work fills this gap by spanning the design space of VR text entry, revising old design rules and giving new ones.

1.2 Research Question

The most straight forward question that can be asked now is:

”Is there a single best text entry method for VR?”

Reality shows that this question may be too simple. Looking closely at the most *common* text entry method for desktop computing, the QWERTY keyboard, one will find that it is not at all the *best* for all circumstances. There are methods that deliver a significantly better expert performance, more ergonomic layouts, etc. So maybe a better question to ask is:

”Which text input methods are feasible for VR and for which circumstances should a given method be chosen?”

Justified design rules are required to answer this question in a clear and scientific way.

1.3 Significance of the Study

This thesis provides a list and classification of existing text entry methods for VR to date and ones that could potentially be used in VR and spans the design space for text entry in VR including non-evaluated variables. Based on this design space six QWERTY based text entry methods are developed and implemented for HTC Vive: Pointing with hand held controllers, Head Gaze using the orientation of the Head Mounted Display, Freehand typing with fingers on the virtual keyboard, Pen Based typing on the virtual keyboard, (Discrete) Touch Pad using the touch pad of the HTC Vive's hand held controllers. The methods were presented to 14 users in a primary informal study followed by the main empirical study to explore their usability and user experience against criteria regarding text entry and VR: Words per Minute (WPM), Keystrokes per Character (KSPC), error rate, immersion, motion sickness, mental and physical demand (NASA task load index), as well as user experience. Conducting an informal study before the

main study was done at the advice of Bowman et al. in chapter 11 of "3D User Interfaces: Theory and Practice" ([2]) to avoid making substantial mistakes during.
Pointing with a tracked hand held controller turned out to be the "best" method out of the tested ones regarding nearly all tested measures: 16.7 WPM, 1.16 KSPC, 0.9% error rate, 1.57 SUS count, 0.14 MSAQ value, 31 NASA task load value, 1.37 UEQ value. Those measures where Pointing was not the "best" method showed no significant differences between the methods. It also turned out that the method had no significant effect on immersion or motion sickness. Based on the findings of the study open questions of the design space are answered, older design rules are updated and possible future work is proposed.

The three contributions of this thesis are:

- A design space for text entry in VR.
- Empirical comparison of QWERTY based selective text entry methods in VR regarding performance, user experience, immersion, motion sickness and mental and physical demand.
- Design guidelines for developers of VR text entry applications.

1.4 Outline

The following chapter will list and classify VR input and output devices as well as (possible) VR text input methods. Based on this related work a design space for text entry in VR is created from which six input methods are derived. Their implementation is then explained. The chapter "Method" describes how those methods were evaluated. Results are given in the next chapter, followed by their discussion and a look towards possible future work. The thesis finishes with a conclusion. Appended are the consent form and the questionnaires used for the evaluation. The bibliography can be found at the very end.

Chapter 2

Related Work

2.1 Virtual Reality Design

With the third VR boom many commercial VR systems went on sale in 2016. They differ a lot in complexity, quality and price. A representational sample is described in detail below.

2.1.1 Output Devices

Today's VR output devices change and improve very quickly and iteratively based on the feedback of the growing community.

2.1.1.1 Mobile VR

Mobile VR experiences build upon the fact that mobile devices today have a sufficient processing power to deliver a solid VR experience. The products often including lenses but omitting a processor or connection to an external one. Users can just put in their own phone and start right away. Head tracking is realized with the sensors included in the phone and relies heavily upon their quality. This can result in drifting over time. To handle this problem some experiences offer a possibility to recenter the view of the user in VR. The main advantage of mobile VR is the low price. You can get a "Google Cardboard" ([7]) for 20€ in 2017. "Gear VR" ([8]) by Oculus is one of the most successful products.

2.1.1.2 Front Facing VR

One of the main contributers to VR in the last years must be Oculus. The company has been developing and releasing their system "Oculus Rift" ([9]) step by step starting with the Developer Kit 1. Up until now it has been aimed at front facing VR experiences. The consumer version of 2016 comes with a head mounted display which is tracked by a tracking camera usually located on the desk of the user. It supports 3D sound via its own Headphones, but users can use their own too. The corresponding input devices "Oculus Touch" were officially released at the end of 2016 and are also tracked by the camera.

Another example of this kind of VR is "Playstation VR" [10] that comes with a tracked Playstation Controller. Another product worth mentioning is "Fove" [11], which is not yet available at the point of release of this thesis. It is the first consumer version of a VR headset that includes eye tracking. It is believed that all VR headsets will include eye tracking in the future because one can reduce computing costs by rendering in lower quality outside of the fovea.

The main disadvantage of front facing experiences is that turning your head 360 degrees is not possible or results in jitter or even tracking loss which in turn causes a break in immersion.

2.1.1.3 Room Scale VR

Room scale VR includes full six-degrees-of-freedom tracking of the users movements in the bounds of a certain room. The most successful commercial product delivering this kind of experience to this date is the "HTC Vive" [12] (see figure 2.1). The HTC Vive comes with a head worn display that is plugged into a computer via cables. It has its own power supply. HTC states that Vive will go wireless in 2017 with an additional gadget. Other gadgets like controller free trackers and third party additions like eye tracking are planned for the future. The HMD and hand held wireless controllers are tracked by a minimum of two base stations installed at opposite edges of the room. As tracking is done optically nothing should block these stations visually. Setting up the device involves drawing a border around your action space with the hand held controllers that will be shown in VR as a grid wall if you come too near to it to avoid bumping into real world objects or walls that have no representation in the VR application. The HMD has a front camera included to support some kind of AR experience such as helping the user find his or her keyboard.

FIGURE 2.1: The HTC vive. REPRINTED FROM theverge.com [13]

Throughout the experiments for this thesis HTC Vive is used as the hardware. The reason for this is that all interaction methods possible with the other available products can be simulated with the HTC Vive in some way but not the other way around.

2.1.2 Input Devices

In contrast to VR output devices that all somehow consist of a tracked head mounted display and differ mostly in complexity and portability, VR Input Devices come in great variety. They are specialized and designed with the task for which they will be used in mind.

2.1.2.1 Mobile Devices

Using mobile smart devices, such as smart phones and tablets as input devices has become more and more popular as most users have one with them in every situation and are familiar with its use and metaphors. Of course one could use them in VR too, as they can provide a uniform way of interaction, independent of the actual setting of the VR environment [14]. Additionally, their use does scale well to the number of users in multiuser environments [14].
"Mobile Devices for Virtual Reality Interaction. A Survey of Techniques and Metaphors" ([14]) by Jens Bauer and Achim Ebert presents the current State-of-the-Art in using

smart devices as input devices and assesses their applicability in the field of Virtual Reality [14]. The autors propose that ”text could be entered through the keyboard on the mobile device” or the DOF of a mobile device could be combined with selection methods like QuikWrite which is described in detail in chapter 2.2.2.

2.1.2.2 Hand held controllers

”Playstation VR” [10] comes with a tracked Playstation Controller, ”Google Daydream” ([15]), Google’s mobile VR system, with a small hand held controller to interact with the virtual world.
Oculus Touch is a hand held controller containing two buttons, a joystick for the thumb and a touchpad for the index finger (See figure 2.2). This simple design still allows complex gesture recognition like pointing (with the users index finger) and grabbing.

FIGURE 2.2: The Oculus Rift. REPRINTED FROM fortune.com [16]

Figure 2.3 shows the hand held controllers of the HTC Vive in detail. Just like the Vive HMD they are tracked in full six-degrees-of-freedom. They include a trigger button for the index finger, two buttons at the side for squeezing interactions, as well as two buttons and a touchpad for the thumb. The orientation of the thumb on the touchpad can easily be visualized in VR as a small dot for example. The touchpad itself can be pressed.

FIGURE 2.3: The HTC vive Controllers. REPRINTED FROM computerbild.de [17]

Throughout the experiments for this thesis HTC Vive hand held controllers are used as the hardware (except for freehand techniques). The reason for this is that all interaction methods possible with the other available products can be simulated with the HTC Vive in some way but not the other way around.

2.1.2.3 Freehand

There will certainly be other room scale VR systems than HTC Vive in the future. One promising product is called "Project Alloy" by Intel ([18]) which includes the possibility to use your hands or everyday objects as input without the need for additional hardware. With "Leap Motion" ([19]) it is already possible to track the users hands in VR by installing the small device at the front of a HMD (which was done for the methods used in the experiments for this thesis). The tracking, via infrared light, is then only possible if the user holds his or her hands right in front of his or her face. Figure 2.4 shows how this can look.

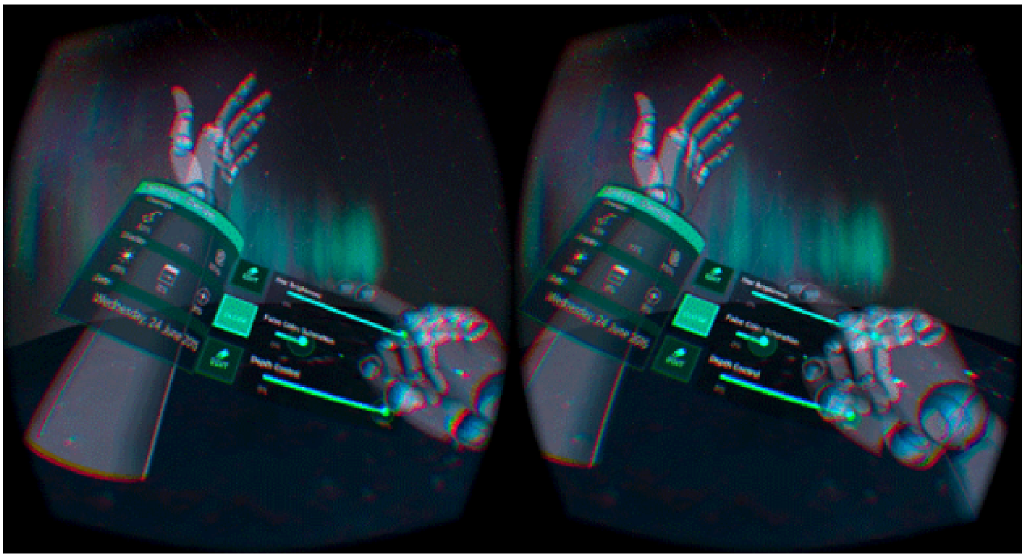


FIGURE 2.4: Leap Motion Planetarium. REPRINTED FROM developer.leapmotion.com [20]

Some older non commercial VR systems used gloves as freehand input devices. Modern versions of this are Manus VR [21] or Hi5 [22].

2.1.2.4 Other input devices

"Gear VR" [8] by Oculus is one of the most successful mobile VR products. Input can be done via a 2D touch pad at the side of the device.
"Fove" [11], which is not available yet at the point of release of this thesis, is the first consumer version of a VR headset that includes eye tracking which could be used for user input too.

2.1.3 Feedback in VR

One part of interaction design is feedback. Acoustic and visual feedback in VR is easily delivered. But especially haptic feedback can improve the usability and user experience of an text input method drastically. VR specific problems are how to handle body movements through solid objects, touching or grabbing.
With UltraHaptics ([23]) Carter et al. introduce a system designed to provide multi-point haptic feedback above an interactive surface. UltraHaptics employs focused ultrasound to project discrete points of haptic feedback through the display and directly on to users' unadorned hands [23]. The system is capable of creating multiple localised points of feedback in mid-air [23]. The authors also showed that users are able to distinguish

between different vibration frequencies of non-contact points with training [23]. For freehand text input (more on that later) this method could help bring about an almost direct translation of the traditional QWERTY keyboard to VR. Of course this technique is far from mass produced but still it is worth considering for future applications.
To compromise for the absence of this technology we used a concept called "Pseudo Haptics" [24]: In the contribution "Usability of Optically Simulated Haptic Feedback" [25] by Hermes et al. a method is described to optically simulate haptic feedback without resorting to mechanical force feedback devices. This method exploits the domination of the visual over the haptic modality [25]. The perception of haptic feedback, usually generated by force feedback devices, was simulated by tiny displacements on the cursor position relative to the intended force [25]. The authors state that OSHF outperforms mechanically simulated haptic feedback and normal feedback, especially in the case of small targets [25]. For VR this means that it should be possible to simulate virtual knobs, buttons and keyboards to achieve a high standard of usability.

2.2 Selective Text Input Methods for VR

The following work is about state of the art text input systems for VR and methods that are not specifically developed for but could potentially be transferred to and evaluated in VR. There is little scientific work to find in this area, nevertheless most commercial VR systems come with several text input possibilities. These are listed and described in detail below. We focus on selective text entry methods because as stated before the use case for text input in VR are short entries. Thus, good learnability is key, as well as avoiding frustration through detection problems.
Some methods can make use of the popular QWERTY layout (see figure 2.5), some not. The methods evaluated in this thesis all make use of it. Jan Noyes reviewed the origins of the QWERTY keyboard, and other sequential keyboards which have been developed since 1909 pretty well ([26]), and is recommended as further reading.

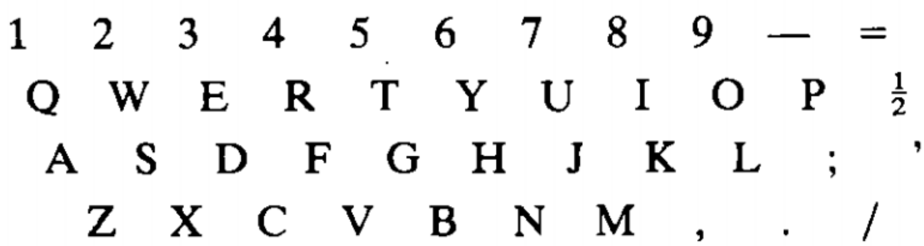


FIGURE 2.5: The standart QWERTY layout. REPRINTED FROM [26]

2.2.1 Head and Eye Gaze

Head and eye gaze can be seen as very much related in design approaches for VR. A design that works for one should easily be transferred to the other, with potentially different performance. This is important as most modern VR systems don't include eye tracking but it is believed that future systems are going to do so for reasons relating to computing power. Rendering only the part in detail that the user looks at can help make 3D applications a lot faster. Neither head nor eye gaze are powerful in selection tasks compared to, for example, the mouse [27]. So most work in this area regarding text input is done with disability or motorimpairment in mind, like the next paper "A Gaze-Controlled Interface to Virtual Reality Applications for Motor- and Speech-Impaired Users" ([28]) for example:
This project by Al-Mubaid et al. aims to overcome the access barriers to virtual worlds for motor- and speech-impaired users by building a gaze-controlled interface for Second Life that will enable them to interact with the virtual world by just moving their eyes [28]. The authors have conducted a study to assess (1) the facilitation of gaze-controlled text input using word prediction technique to speed up chatstyle text input in virtual worlds, (2) the influence of screen layout on the efficiency of text input, (3) the effect of the maximum number of suggested words on typing efficiency, and (4) the performance of non-disabled vs. motorimpaired users [28]. The project uses a QWERTY layout and word prediction methods to improve text input rates. The authors evaluated 0.17 "Delete Operations per Letter Typed" and 2350 "MS taken per Letter Typed" for healthy subjects. This roughly results in 12 seconds per word which again roughly results in 5 word per minute (henceforth WPM). A similar approach was used for the experiments for this study, but using head instead of eye gaze. It is important to notice that the term "virtual reality" used in the title of this paper does not align with the definition of LaValle ([4]). Al-Mubaid et al. rather use it as an umbrella term for virtual worlds like "Second Life" [29] or "World of Warcraft" [30]. Nevertheless the circumstances in VR are very similar. The "disability" here is that the user can not directly see his or her hands.
All head or eye gaze selection based methods could benefit from a VR context as it is easier to personalize size and position of the user interface (here the QWERTY keyboard) to be as ergonomic as possible. Using a simple monitor instead is a restriction for overall size.

2.2.2 Pen and Tablet Based Text Input

Pen and tabled based interaction requires two hands. One that holds the tablet and one that interacts on it with the pen. Alternatively the user may put the tablet on a desk or fix surface or the like. For the experiments of this study we chose a floating keyboard that could be interacted with via two pens.

With "Virtual Notepad" Poupyrev et al. present Virtual Notepad, a collection of interface tools that allows the user to take notes, annotate documents and input text using a pen, while still immersed in VR [31]. Using a spatially-tracked, pressure-sensitive graphics tablet, pen and handwriting recognition software, Virtual Notepad explores handwriting as a modality for interaction in VR [31]. The authors report details of the Virtual Notepad interface and interaction techniques, discuss implementation and design issues, report the results of initial evaluation and overview possible applications of virtual handwriting [31]. The paper shows that one can interact with the pen and tablet in VR just in the way one would do outside of VR. So nearly all metaphors and interaction techniques stay the same. This results in the fact that one can adapt scientific work on pen and tablets to this special VR context. Text entry and error rates should adapt as well. Unfortunately this paper does not cover them. Bowman et al enhanced the idea of this paper with the following method: Virtual keys in the standard QWERTY layout are displayed on the surface of the virtual tablet, and users type a letter by touching it with a stylus then pressing the stylus button [5]. Compared in VR to other text entry techniques this method was relatively fast (49.68 cpm (roughly 10 WPM)), had the fewest errors, and was reported to be natural and easy to learn, but it also produced high levels of arm strain [5]. García et al. [6] found around 38 CPM (roughly 8 WPM) with an error rate of around 8%. They also tested two other pen based methods: disk keyboard and handwriting. Disk writing performed better regarding error rate (2%) but not regarding entry rate (around 21 CPM (roughly 2 WPM)). This method follows the Quikwriting method which is described below. Handwriting performed worse than QWERTY. Frees et al. [32] proposed a pen based method for VR (CTD) containing 9 connectable dots that can form characters. They compared their approach to a QWERTY based method and found that the QWERTY based method performed better concerning error rate and WPM.

The methods described next could benefit from a VR context due to the possibility to fixate the "tablet" anywhere in the virtual environment, in the most ergonomic position and size, reducing the physical demand.

Masui for example proposes a fast text input method for pen-based computers, where text is not composed by entering characters one by one, but by selecting words from a menu of candidates created by filtering the dictionary and predicting from context [33].

Similar methods are used in nearly all text entry systems nowadays to increase performance and usability. The method can make use of the QUERTY layout and achieved an average text input speed of about 40 chars/min for the Japanese language. It's hard to estimate the WPM for the Latin Alphabet or English language out of this but it may be around 10-15 WMP (Japanese uses a syllable based notation).

2.2.3 Pointing

Pointing methods can make use of the QWERTY layout. Due to the fact that many commercially successful VR products use pointing as a general input, text input via pointing has become the most common VR text input method. In fact the concept of pointing with your hands or fingers has some parallels to doing so with your head or eyes. One advantage here is, that you may use both hands, something Steam VR unfortunately makes no use of.

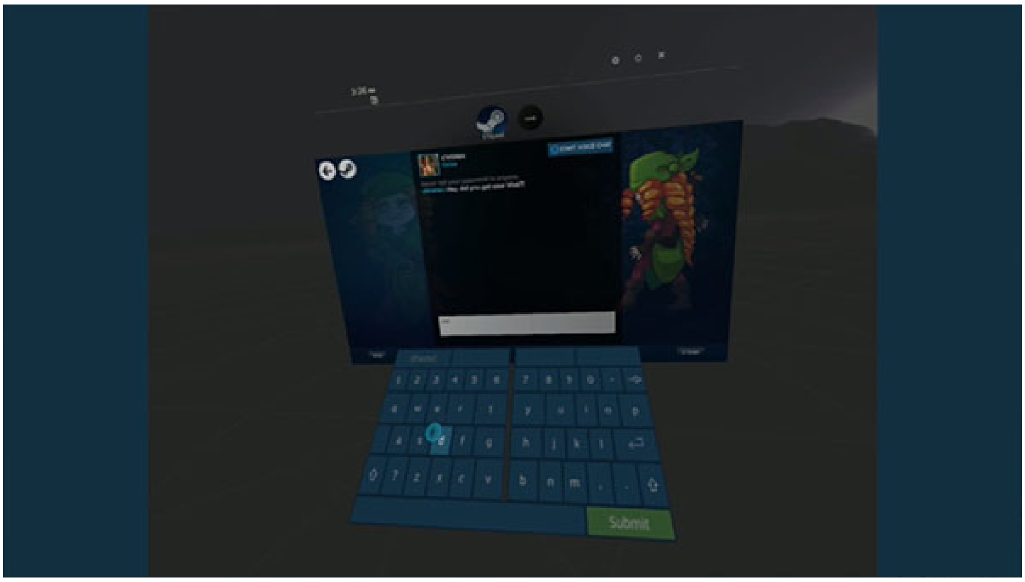


FIGURE 2.6: The Steam VR Keyboard. REPRINTED FROM valvetime.net [34]

Steam VR ([35]) is a pseudo operating system developed by Valve where the user can manage and navigate through his or her VR applications. He or she can even control her desktop in VR. Input is done via pointing with handheld controllers to control the cursor. Selection is done by pressing the trigger button. This way the user can select a textfield to make the Steam VR keyboard appear which follows the traditional QWERTY design (see figure 2.6). It is stuck in place in front and slightly below of the users head and delivers optical feedback of the ray between input device and selected key, as well as the selected key itself. An acoustic feedback is provided at the change of selection as well as

at confirming it (actual typing). As mentioned above, unfortunately you can only point with one controller. This method was evaluated during the experiments of this study, but users could use two controllers.
Daydream Keyboard ([15]) is another on-screen virtual reality keyboard for text entry in Daydream applications. Just like the Steam VR Keyboard it is stuck in front of the users head. It is separated into four logical parts: There is a numblock on the left hand side with the numbers from 1 to 0 as well as a plus and a minus symbol. In the middle you can find the QWERTY design along with some special characters. The right block contains three functional keys for deleting, the enter key and one for the keyboard options. The typed text is displayed above it all in a separate field. Input is done via pointing with the single (again just one) hand controller, as can be seen in figure 2.7.

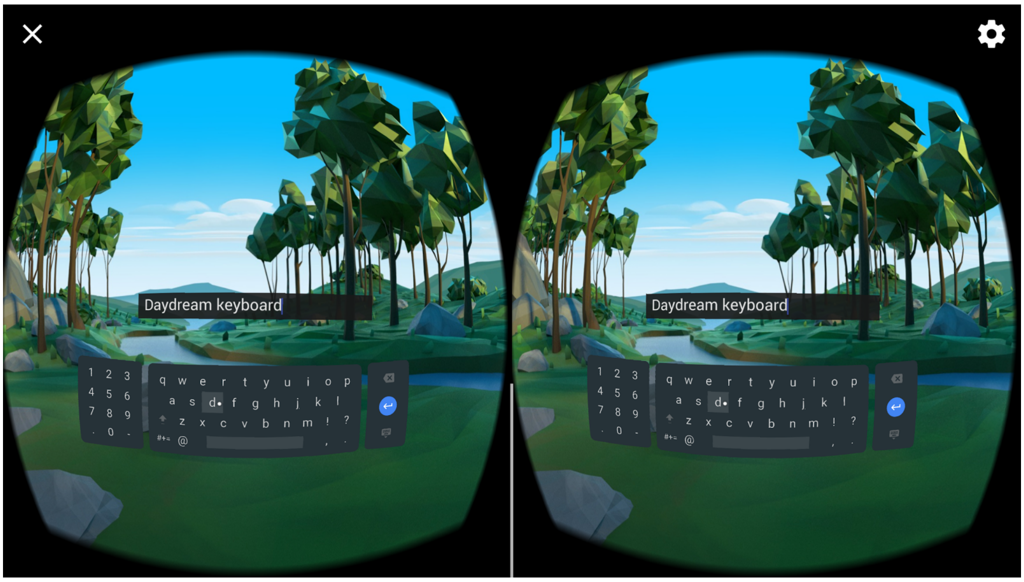


FIGURE 2.7: The Google Daydream Keyboard. REPRINTED FROM vr.google.com [15]

The paper "Mid-Air Text Input Techniques for Very Large Wall Displays" by Booth et al. discusses the potential of mid-air interaction techniques for text input on very large wall displays, and introduces two factors, distance-dependence and visibility-dependence, which are useful for segmenting the design space of mid-air techniques [36]. The authors suggest that distance-independent techniques may be best for use with very large wall displays [36]. The average input speed in WPM was 18.9 for their QWERTY method. Distance had a significant impact on performance: In the second experiment focusing on distance-dependence it was 14.5 wpm at 9 feet and 10.3 wpm at 18 feet [36]. The authors even found that the performance of other pointing techniques are distance-dependent too, even those that were designed not to be.
Hornbaek et al. found an average of 13.2 WPM with their similar method "Projected QWERTY" which shows a standard QWERTY keyboard layout on the display, with a

dot cursor that can be controlled by moving the hand [37]. A character is produced by moving the hand to a key and tapping [37]. The location of the user's hand is projected onto the display plane [37]. Users rated this method highest in terms of satisfaction and it resulted in the least physical movement compared to none QWERTY methods [37]. For the VR context this implies that there may be a "perfect" position and scale for the displayed keyboard for each user. Trying out distance-independent designs will probably not increase the usability and performance significantly. Nevertheless all pointing methods could greatly benefit from a VR context. Distant dependent pointing could be supported by visualizing the pointing ray giving the user more stability through this visual feedback. As size and position are not restricted in VR the problems of distance dependencies could be reduced too.

2.2.4 Freehand Typing

Much work has already been done on freehand text input that could benefit from a VR context by delivering pseudo haptic feedback (see above).
Some VR applications make use of hand tracking techniques, for example using the Leap Motion ([19]). This is what was done for the experiments in this thesis too. Haptic feedback and feedback in general is an issue in this case. Most common hand tracking systems like the Leap Motion ([19]) are not very precise and make conservative 10 finger input hard to realize in VR. Nevertheless Feit et al. found that self-taught typists can achieve performance levels comparable with touch typists, even when using fewer fingers [38]. The exposed 3 predictors of high performance: 1) unambiguous mapping (a letter is consistently pressed by the same finger), 2) active preparation of upcoming keystrokes, and 3) minimal global hand motion [38].
In the paper "ATK: Enabling Ten-Finger Freehand Typing in Air Based on 3D Hand Tracking Data" however, Gao et al. present ATK, an interaction technique that enables freehand ten-finger typing in the air based on 3D hand tracking data [39] using Leap Motion ([19]) (See figure 2.8)). The authors proposed a probabilistic tap detection algorithm [39]. Participants typed 23.0 WPM with an uncorrected word-level error rate of 0.3% in the first block, and later achieved 29.2 WPM in the last block without sacrificing accuracy [39].

FIGURE 2.8: ATK. REPRINTED FROM [39]

1998 Anthony Natoli already registered a patent on a freehand "Virtual reality keyboard system and method" ([40]) which expired in 2002. The VR keyboard system and method receive a VR glove position, generate a corresponding key code from the VR glove position using a predetermined mapping, and send the key code to an application program as a key input corresponding to a keyboard and/or keypad entrapment of data and/or a command [40]. The system and method also generate a display representing the key input based on the VR glove position [40]. There are other QWERTY based approaches for gloved based text entry like KITTY for VR [41] where the three key rows are mapped to three locations of the thumbs. DigiTap [42] works in a similar way without using the QWERTY layout. Bowman et al.'s pinch keyboard for VR [5] achieved around 30 cpm (roughly 6 WPM) [5], [6] with an error rate of 10% [6]. the authors claim that "subjects found the pinch keyboard technique natural and easy to learn" [6].
Senseboard ([43]) is a technology invented by Dr. Lars Asplund. It developed into "project Virtual Keyboard" (see figure 2.9) . The technology measures the position of the fingertip relative to the hand [43]. This actually makes it possible to have two degrees of freedom per finger (actually three or even four) [43].

FIGURE 2.9: The Senseboard. REPRINTED FROM [43]

Gamberini et al. derived design principles for socially acceptable, yet versatile, interaction techniques for smart glasses [44] which can partially be transferred to VR:

- Isolating sensing technology from the glasses.
- Using relative pointing for adapting to various postures.
- Designing small movements for subtle interaction.
- Aiming for intuitive gestures.
- Enhancing the tangibility.

This lead to the following design decisions for their haptic glove based text entry system:

- The keys are grouped in three, in which only one of the groups will be highlighted according to hand orientation. By bending one of the fingers (thumb, index, or middle finger), the user can type the corresponding character in the highlighted group. (See figure 2.10)
- Use QWERTY keyboard.
- Fixed key group composition: The composition of the key groups is fixed so that panning the hand horizontally results in shifting three keys at once. This way, the precision required in the interface is reduced.
- Additional margin in between key groups to increase robustness.
- Tactile feedback at changing the key group as well as activating a key.

[44]

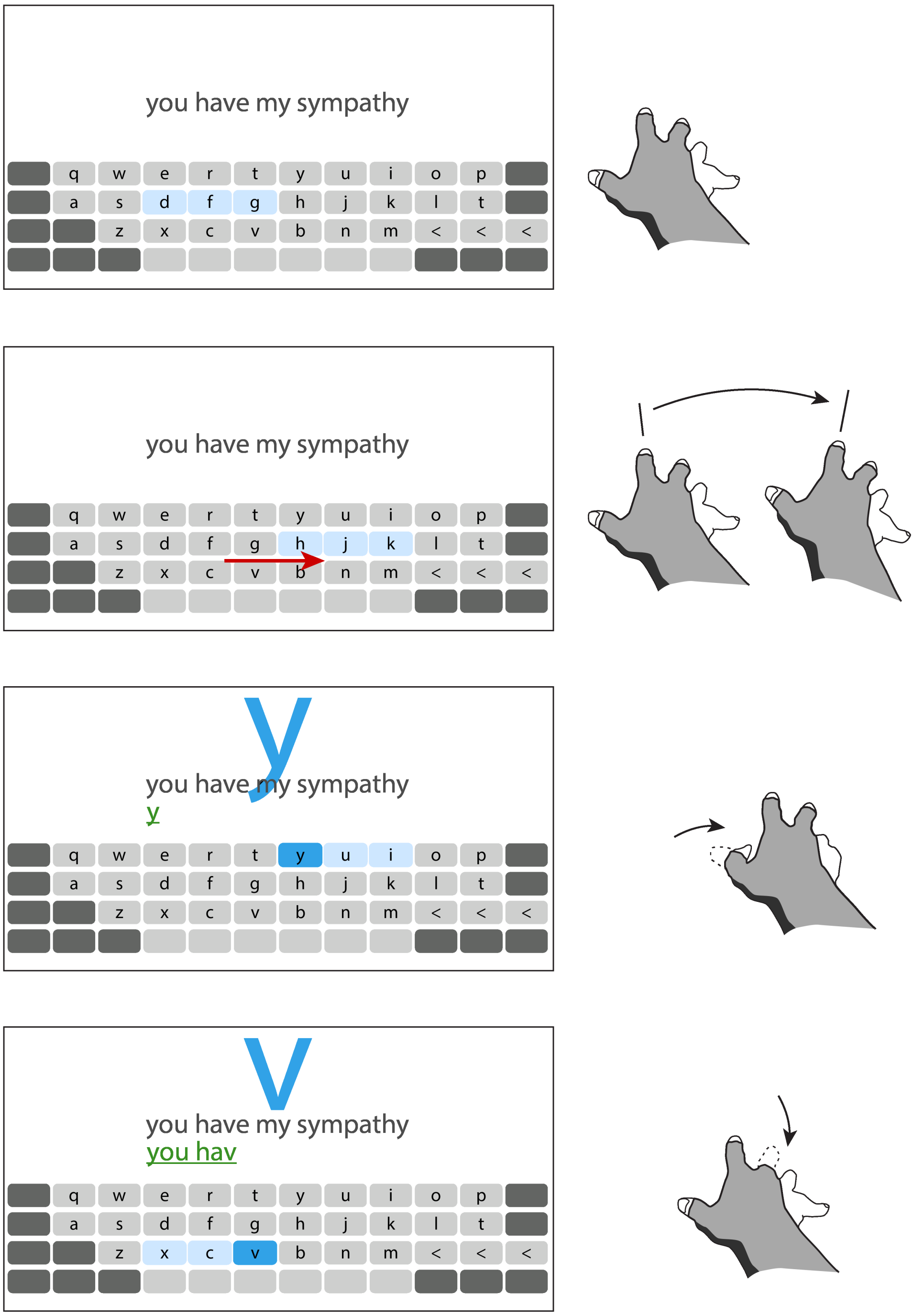


FIGURE 2.10: Text entry scenario. In a) and b) a key group is highlighted according to current hand orientation. A preset phrase appears for the evaluation. In c) and d) bending an individual finger would insert a corresponding key in the highlighted key group. REPRINTED FROM [44]

2.2.5 Display Separated Text Input

Here "display separated" means that the selection of keys on a keyboard is controlled indirectly. Chording keyboards are a special case.
In Steam VR (see Figure 2.6) you can not just input text per pointing technique but also by a game controller like method (also evaluated during the experiments in this thesis). You can touch the touch pad of one of the controllers which imediately switches the text input mode from pointing to this display separated method. Each controller of the HTC vive ([12]) has a touch pad to control the cursor belonging to one half of the keyboard. The keyboard stays the same but now you can control the selected key by moving your finger on the touchpad. The users left thumb controls the left side of the keyboard, the right thumb the right side. Visual, haptic and acoustic feedback is provided. The position of the thumbs on the touchpad is mapped directly to the absolute position of the selected key on the keyboard. This means this method differs from the one used in most traditional gaming consoles like the XBox ([45]) or Playstation ([46]), where you control the relative position of the selected key with a joystick. Here the user can control the relative position of the cursor via joysticks for his or her thumbs. Newer versions of both can even attach small keyboards which can be controlled via the user's thumbs.
Many proposed text entry method for VR use mobile phones or similar mobile devices for display separated input like VirtualPhonepad [47], or VirtualQWERTY [48]. García et al. compared their mobile phone keyboard for VR with other VR text entry methods. They found text entry rates of about 65 CPM (roughly 13 WPM) and error rates of about 3% [6].
In the non-VR-specific paper "A Pervasive Keyboard - Separating Input from Display" Magerkurth and Stenzel present a text input method for public displays and palmtop computers that separates the input from the display of the edited text [49]. While the public display shows both the edited text and a character generation interface, palmtop computers are used for input in a blind way, i.e. without the need to look at the palmtop's small screen [49]. With the QWERTY layout users reached between 15 and 30 WPM [49].

2.3 Non-Selective Text Input Methods

With "Quikwriting: Continuous Stylus-based Text Entry" Kevin Perlin presents a "heads-up" shorthand for entering text on a stylus-based computer very rapidly [50]. The user works with a very simple stylized alphabet, in which each character represents one character on the standard typewriter keyboard [50]. The writing area is centered around wherever the user first puts down the stylus [50]. This writing area is divided into a number of zones arranged around a central resting zone [50]. In the implementation of figure 2.11, the zones are arranged in a 3x3 grid, numbered 1 through 9, where zone 5 is the central resting zone. To form a character, the user drags the stylus from the central resting zone out to one of the eight outer zones (1,2,3,4,6,7,8,9), then optionally to a second outer zone, and finally back to the resting zone [50]. Gestures are chosen so that frequent characters can be entered very rapidly [50]. For example, to draw Space, 'e', 't', 'a', 'o', or 'n', the user moves the stylus out of the resting zone and then immediately back again [50]. To form other characters, the user moves the stylus from the resting zone first into one zone, and then into a second zone, before moving the stylus back into the resting zone [50]. Once writing begins, the stylus need never be lifted [50]. Furthermore, the user need never stop moving the stylus [50].

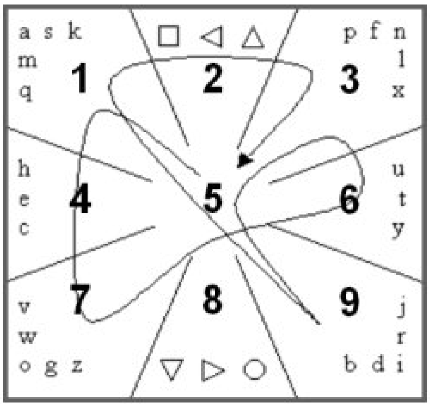


Figure 2.11: Quizkwriting input of "QUIK". REPRINTED FROM[51]

The good thing about this method is, that it is not bound to pen and tablet based input, but can be transferred to head or eye gaze (see [52]), or even to direct or indirect pointing techniques. This makes it a potentially good fit for VR as the method itself is independent from the input device.

2.3.1 Chording

The following work examines chording keyboards, text input methods using a combination of keys (chord) to express a certain letter. The concept works one handed and can be expanded to two hands. Even though high WPM rates can be achieved the learning curve is pretty high. One large drawback of this method is the lack or complexity of visual feedback; Something one could handle in VR by giving context sensitive visual feedback on the keys. So chording keyboard approaches could also benefit from a VR context. Anyways, chording keyboards can be used eyes free too because it delivers haptic feedback with distinct and simple-to-use finger-to-button mapping. It is obvious that chording keyboards can't easily follow the QWERTY layout.

One approach to use chording in VR is via gloves. Chords can be created by pressing the fingers against any surface, other fingers or even other body parts. One such glove is described by Rosenberg et al. as simple "Chording Glove" and is specifically designed as a text input device for wearable computers and virtual environments [53]. The keys of a chord keyboard are mounted on the fingers of a glove [53]. Shift buttons placed on the index finger enable the glove to enter the full ASCII character set [53]. After an average of 80 min of a tutorial, ten subjects reached a continuous text input speed of 8.9+1.4 WPM, and after 10 1-hr sessions, they achieved 16.8+2.5 WPM [53]. The authors made quite an effort to keep the chord keymap as intuitive and usable as possible. The chord can for example resemble the character(see figure 2.12), common sequences of chords are easier or one chord can be based on another one [53].

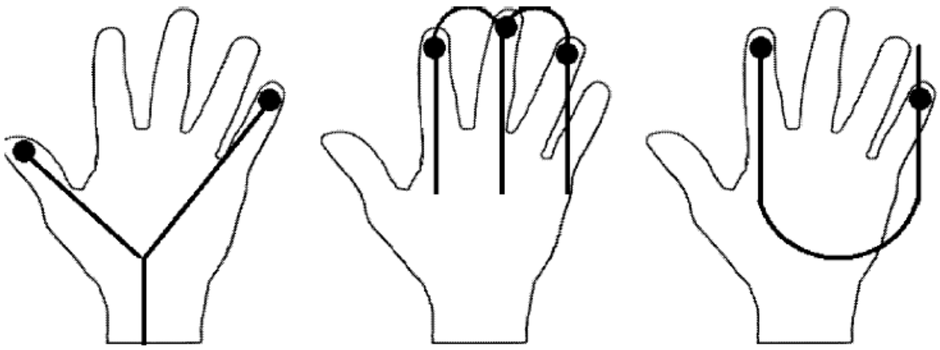


FIGURE 2.12: Chords for the letters Y, M, and U are some of the chords that resemble the letters they make. REPRINTED FROM [53]

This method can be used in VR. Still one bigger disadvantage of gloves is that users may not want to share them with others due to hygienic issues and that they are not easily put on.

Many VR devices come with hand held controllers containing many buttons though. They could be used for simulating or even designed towards resembling the Twiddler (See figure 2.13), a mobile one–handed chording keyboard with a keypad similar to a mobile phone. It has twelve keys arranged in a grid of three columns and four rows on the front of the device. Each row of keys is operated by one of the user's four fingers. Additionally, the Twiddler has several modifier buttons such as "Alt" on the top operated by the user's thumb. Users hold the device in the palm of their hand like a cup with the keys facing away from their bodies All five fingers on a hand can be used to type. Each letter of the alphabet can be typed on the Twiddler by pressing one or two keys concurrently. The Twiddler also has the feature of multi–character chords (MCCs). For instance, the keyboard has chords for some frequent words and letter sets such as 'and', 'the', and 'ing'. Users can also define their own MCCs. This has positive implications for the number of keystrokes per character (KSPC) needed to type. [54]

FIGURE 2.13: The Twiddler III. REPRINTED FROM theregister.co.uk [55]

An experienced user of the Twiddler averages speeds of 60 words per minute with letter–by–letter typing of standard test phrases [54]. This fast typing rate coupled with the Twiddler's 3x4 button design, similar to that of a standard mobile telephone, makes it a potential alternative for text entry for VR. Drew et al. ([54]) present a longitudinal study of novice users' learning rates on the Twiddler. The authors found that users initially have a faster average typing rate with multi–tap; however, after four sessions the difference becomes negligible, and by the eighth session participants type faster with chording on the Twiddler [54]. Furthermore, after 20 sessions typing rates for the Twiddler are still increasing [54]. The mean entry rates for session one of the evaluation were 4.3 WPM [54]. As sessions continued, the means improved and reached 26.2 WPM by session 20 [54]. Bowman et al. implemented a VR text entry method for the Twiddler2. They provide a visual aid in the HMD that shows the user the layout of the keys on the device, so that even novice users can determine which keys to press [5]. After comparing it in VR to other text entry methods the authors stated that the chord keyboard should not be the device of choice for text input in VR [5] as it had the worst performance (21 CPM (roughly 5 WPM)). García et al. [6] confirmed these results with 16 CPM (roughly 3 WPM) with an error rate of around 17%. One reason for that is, the high learning curve of cording keyboards. As text entry in VR is usually very short, this method was skipped for the experiments in this study.

2.3.2 Text Entry at HMD

Some Head Mounted Displays can be interacted with via touch input at the side of the device. The successful mobile VR glasses "Gear VR" ([8]) for example use a 2D touch pad at the side of the device for input. The Augmented Reality Glasses by Google ([56]) use a one dimensional one.

In the paper "One-Dimensional Handwriting: Inputting Letters and Words on Smart Glasses" ([57]) Li et al. present 1D Handwriting, a unistroke gesture technique enabling text entry on a one-dimensional interface. The challenge here is to map two-dimensional handwriting to a reduced one-dimensional space, while achieving a balance between memorability and performance efficiency [57]. The authors derive a set of ambiguous two-length unistroke gestures, each mapping to 1-4 letters. To input words, they design a Bayesian algorithm that takes into account the probability of gestures and the language model [57]. To input letters, they design a pause gesture allowing users to switch into letter selection mode seamlessly [57]. With extensive training, text entry rate can reach 19.6 WPM and users' subjective feedback indicates 1D Handwriting is easy to learn and efficient to use.[57]. This text input method of Li et al. could without further ado be transferred to all devices with a similar touch pad.

As some touch pads are even two dimensional the gestures can be designed to be even more intuitive. A lot of research is done on this area: Unistrokes ([58]) was designed for fast input, Graffiti ([59]) to resemble handwritten Roman, Edgewrite ([60]) for high accuracy and motion stability.

One touch pad based selective text entry method was evaluated during the experiment in this thesis which can be transferred to such a two dimensional touch pad too.

2.3.3 Gesture Based Text Input

Another promising approach for freehand text input in VR are word gesture keyboards which enable fast text entry by letting users draw the shape of a word on the input surface. Such keyboards have been used extensively for touch devices [61]. With Vulture (Figure 2.14) Hornbaek et al. present a word-gesture keyboard for mid-air operation [61]. Vulture adapts touch based word-gesture algorithms to work in midair, projects user's movement onto the display, and uses pinch as a word delimiter [61]. A first 10-session study suggests text-entry rates of 20.6 WPM and finds hand-movement speed to be the primary predictor of WPM [61]. A second study shows that with training on a few phrases, participants do 28.1 WPM [61]. Participant's recall of trained gestures in mid-air was low, suggesting that visual feedback is important but also limits performance [61].

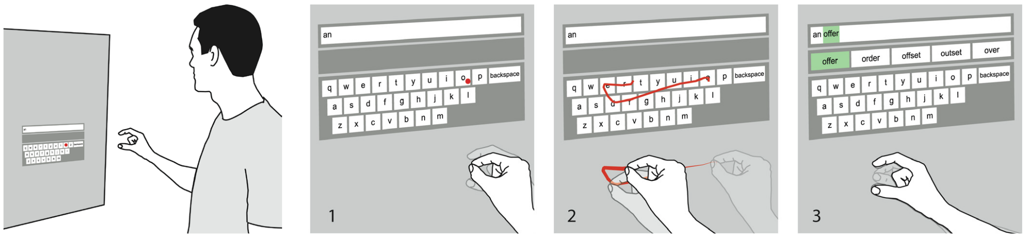


FIGURE 2.14: Text entry using word-gestures in mid-air: By moving the hand, the user places the cursor over the first letter of the word and (1) makes a pinch gesture with thumb and index finger, (2) then traces the word in the air—the trace is shown on the screen. (3) Upon releasing the pinch, the five words that best match the gesture are proposed; the top match is pre-selected. REPRINTED FROM [61]

One way to minimize distance-dependence is to project movement orthogonally onto the display. Orthogonal projection limits the user's reach, but maintains a constant control-display ratio across distances [61].

Another Midair method for text input is to move away from keyboard representations or abstractions towards gestures:

With "AirStroke", Bowman et al. explore the opportunity of bringing unistroke text entry to freehand gesture interfaces ([62]). Using Graffiti's alphabet, AirStroke takes advantage of the richer input capabilities of two-handed freehand gestures by providing combined mode selection and character entry with one hand, as well as word completion with the other hand [62]. A longitudinal study suggests that AirStroke has competitive speed and accuracy to unistroke methods based on stylus input (11 WPM) [62].

In view of the widely used Leap Motion ([19]) sensor it makes sense to have a look at what can be done with it in VR. In the paper "Investigating the Dexterity of Multi-Finger Input for Mid-Air Text Entry" Feit et al. investigate an emerging input method enabled by progress in hand tracking: input by free motion of fingers [63]. Their findings provide indices quantifying the individuation of single fingers [63]. They apply their findings to text entry by computational optimization of multi-finger gestures in mid-air and define a novel objective function that considers performance, anatomical factors, and learnability [63]. First investigations of one optimization case show entry rates of 22 WPM (Figure 2.15) [63].

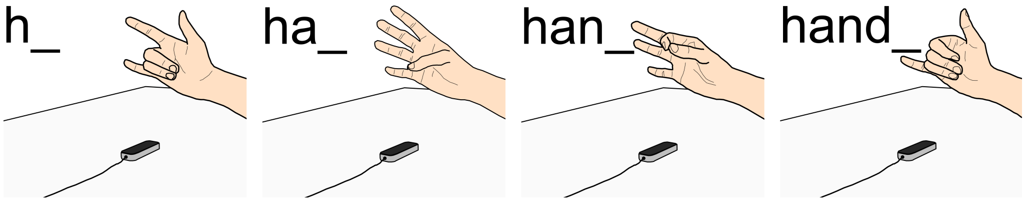


Figure 2.15: An example of the word 'hand' being typed using one of the author's automatically obtained designs. REPRINTED FROM [63]

In VR this could potentially be realized very quickly as current technology supports this approach ([19]).
Gesture based text entry methods have a lot of drawbacks such as a high learning curve and usually only achieve around 10 WPM.

2.3.4 Speech Input

Even though speech input has some serious drawbacks it certainly is worth looking into. Especially for VR applications text input per speech could be very immersive and is the first thing many people think of when asked how they would like to input text in VR. The problems begin with privacy issues and extend beyond recognition issues. One big drawback is, that speech is not persistent. Especially the social aspect that is often neglected when talking about VR. The problem here is, that the user wearing a head mounted display can not easily see the reaction of his or her surroundings. Plus using VR glasses in a crowded place may soon become reality too. The user may be concerned that his or her movements or what he or she says are interpreted different than intended or even ridiculous [44].
Schneiderman ([64]) states that problem solving and speaking are done in the same brain region so speaking consumes precious cognitive resources. Bowman et al. suggest to not assume that speech will always be the best technique for text entry in 3D user interfaces as it can lead to error-correction problems, user frustration, and user self-consciousness [2]. Compared in VR to other text input methods the (Wizard of OZ) speech technique of Bowman et al. [5] was the fastest (66 cpm (roughly 13 WPM)), but it also produced more errors than other methods, and was found to be tedious by many of the subjects. Either way performance can be an issue too. SpeeG for example is a multimodal speech- and body gesture-based text input system targeting media centres, set-top boxes and game consoles [65]. An evaluation of the SpeeG prototype has revealed that low error rates for a text input speed of about six words per minute can be achieved after a minimal learning phase [65]. The authors use Sphinx ([66]), an open source framework for speech recognition. It is just one of many as speech recognition itself is a huge area of investigation.

Chapter 3

Empirical Analysis of Existing Text Entry Methods for VR

3.1 Classification

Based on the research listed in the previous chapter classifications for VR input devices as well as (possible) text input methods are given in the following. For a classification of input devices see table 3.1. Please note that devices with zero buttons but a touch pad can simulate buttons on it. For a classification of (possible) VR text input methods see table 3.2. Please note the assumed WPM for each method are just rough assumptions and can not be compared as they come from different experiment setups.

Device	**Min DOF**	**Touch**	**Haptic Feedback**	**Extended Hardware**	**Physical Buttons**
Mobile Devices	6	✓	✓		0
Hand Held Controllers	6		✓		1-many
HMD	3			✗	1-very few
Freehand	6	✗	✗	✓	0

Table 3.1: Classification of VR Input Devices.

Method	**Needed Hands**	**Haptic Feedback**	**QWERTY Possible**	**Assumed WPM**	**Notes**
At HMD	1	✓		10-20 [57], [58], [59], [60]	This technique assumes buttons or touch pad at HMD
Chording	1-2	✓	✗	3-26 [54], [5], [6]	The WPM may improve with training
Head and Eye Gaze	0	✗	✓	5-8 [28], [52]	
Pen Based	1-2		✓	8-10 [5], [6]	Haptic feedback can be provided when a physical tablet is used
Pointing	1-2	✓	✓	13-19 [36], [37]	
Display Seperated	1-2	✓	✓	13-30 [49], [6]	
Freehand Typing	1-2		✓	6-23 [39], [5], [6]	
Gesture Based	1-2	✗		11-23 [61], [63], [62]	
Speech	0	✗	✗	6-13 [65], [6]	

TABLE 3.2: Classification of (possible) VR text input methods

3.2 Empirical Comparison

Most of the presented related work present a single (VR) text input technique. Few of them conducted performance or user experience studies and two already compared some VR input techniques.

Bowman et al. [5] compared a pinch keyboard, a pen and tablet keyboard, a chord keyboard, and speech. The basic concept of the pinch keyboard is that a simple pinch between a thumb and finger on the same hand represents a key press by that finger. More precisely, with the left (tracked) hand over the central row (the reference one), the contact between the little finger and the thumb is interpreted as pressing the ‘a’-key on a standard QWERTY keyboard, between the index and the thumb as the ‘f’-key and so on [6].

Their experiments showed that none of these techniques is clearly the best for text input in VR [5]. The speech technique was the fastest, but it also produced more errors than the pen and tablet keyboard, and was found to be tedious by many of the subjects

[5]. The pen and tablet keyboard was relatively fast, had the fewest errors, and was reported to be natural and easy to learn, but it also produced high levels of arm strain [5]. Subjects found the Pinch Keyboard technique natural and easy to learn, but its performance was sub-par due to some unresolved usability issues [5]. The authors also stated that the chord keyboard should not be the device of choice for text input in VR [5].

The authors posit a few general guidelines for the use of text input techniques in actual VR applications:

- If both hands need to be free to do other tasks in the VE, then only speech and the Pinch Keyboard (of the techniques they tested) need be considered [5].
- If at least one hand must be free, the pen and tablet technique may still be used assuming that the stylus can be used for the other interaction tasks [5].
- If user satisfaction and engagement is an important factor, then the pen and tablet keyboard and the Pinch Keyboard should be considered [5].

Bowman et al.'s work was proven and extended by García et al. [6]. They implemented and compared six text input methods in VR:

- Mobile phone keyboard: 9 keys with each key mapped to several characters, no virtual representation in VR
- Chord keyboard: Twiddler2, no virtual representation in VR
- Pinch keyboard: described above, user's hands are represented with yellow cubes
- Pen and tablet techniques: pen-based QWERTY keyboard, pen-based disk keyboard (a Quikwriting [50] technique) and handwritten character recognition

The authors tested 10 subjects without VR experience. Results can be seen in figure 3.1. Note that the CPM value is roughly five times higher than WPM.

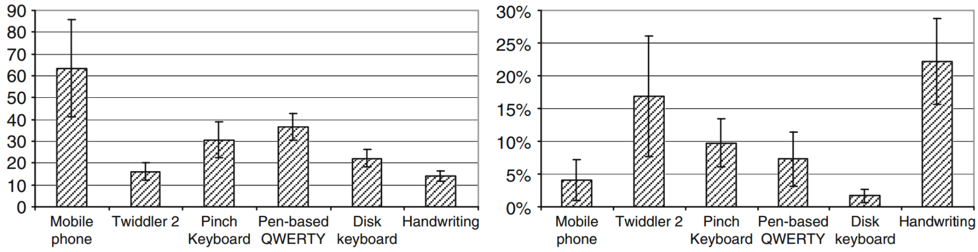


FIGURE 3.1: Characters per minute (left) and typing errors (right) for each technique. REPRINTED FROM [6]

Out of their results the authors developed a guidance tool for text input techniques in VR (see figure 3.2).

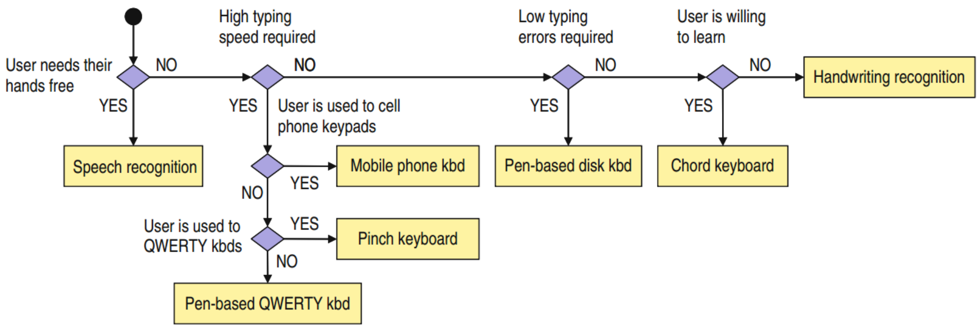


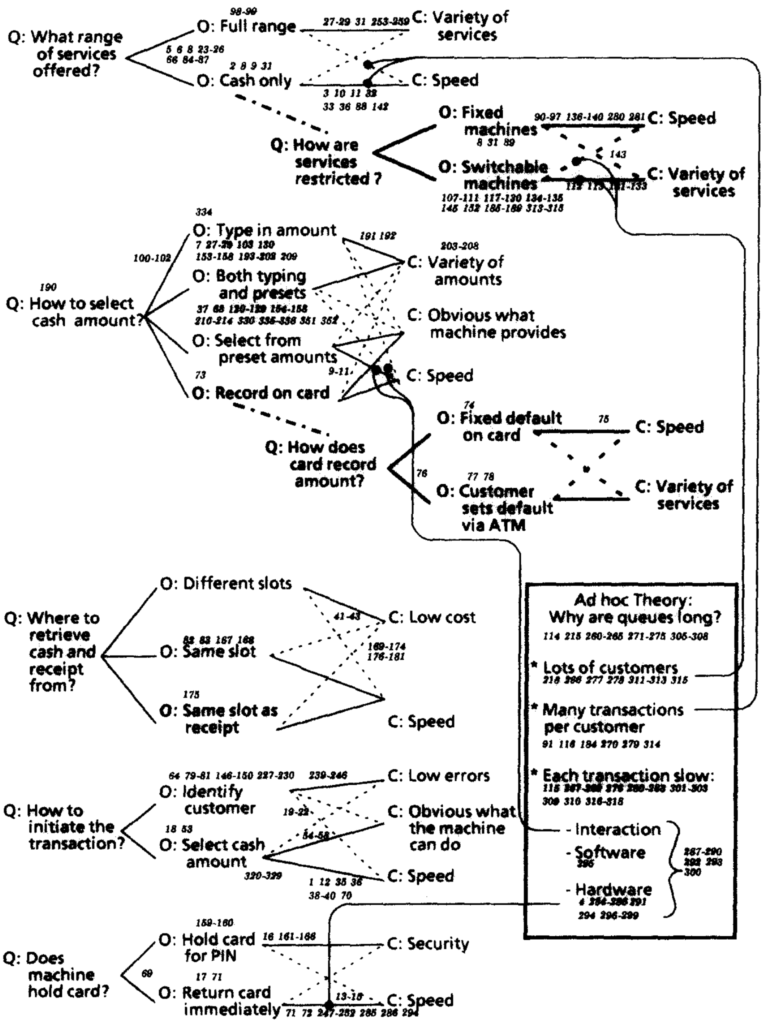
FIGURE 3.2: VE design guidance tool for text input techniques by García et al. REPRINTED FROM [6]

Chapter 4

Design Space of VR Text Input

In this chapter the design space for text input in VR is examined. Questions regarding the design are introduced and possible answers are classified according to criteria. Out of that 6 selective text entry methods for VR are derived from this and are considered in relation to this design space.
Design rationale is important because text input in VR needs to be understood by a wide variety of people who have to deal with it. This variety of people ranges from those who design and build it to those who sell and service it to those who actually use it. What is important for many of these people is not just the specific text input method itself but its other possibilities. For example, a designer decides between different possible ways to shape the keyboard layout; a developer wants to change the method to respond to a new need without disturbing the integrity of the method; a user wonders why this method is different from some other familiar one. An important way to understand a text input method for VR is to compare it to how it might otherwise be. [67]
To define the design space of VR Text Input the design space analysis methods of MacLean et al. [67] are used. They distinguished criteria (ways of assessing designs), questions (about what a design should do), and options (answers to questions) as three key components for mapping a design space. Figure 4.1 shows an example of such a design space proposed by the authors. For the sake of clarity we will split the visualisation of our own design space into one part per question.

Figure 8. **QOC diagram representing the structure of the discussion of the ATM1 design session. This diagram is based on the a priori analysis diagrammed in Figure 5. The bold items are new. The italic numbers represent the assertion numbers in the protocol.**



226

Figure 4.1: A Questions-Options-Criteria example of a design space([67])

4.1 Criteria

In the following section the main criteria for VR text input design are presented based on HCI principles like usability and user experience.

4.1.1 Effectiveness (WPM)

Words per minute (WPM) is perhaps the most widely reported empirical measure of text entry performance. Since about 1905, it has been common practice to regard a "word" as 5 characters, including spaces [51]. Importantly, the WPM measure does not consider the number of keystrokes or gestures made during entry, but only the length of the resulting transcribed string and how long it takes to produce it [51]. Thus, the formula for computing WPM is:

$$\frac{|T| - 1}{S} \times 60 \times \frac{1}{5} \tag{4.1}$$

T is the final transcribed string entered by the subject, and |T| is the length of this string. T may contain letters, numbers, punctuation, spaces, and other printable characters, but not backspaces. Thus, T does not capture the text entry process, but only the text entry result. The S term is seconds, measured from the entry of the first character to the entry of the last, including backspaces. The "60" is seconds per minute and the "1/5" is words per character.

[51]

4.1.2 Error Rate (Robustness, Efficiency)

One way to quantify errors during text entry is to use the keystrokes per character (KSPC) performance measure. This measure is a simple ratio of the number of entered characters (including backspaces) |IS| to the final number of characters in the transcribed string |T| . The formula for calculating KSPC is:

$$\frac{|IS|}{|T|} \tag{4.2}$$

[51]

Another way is the minimum string distance that can be calculated with the Damerau-Levenshtein-distance [68]. This distance is the minimum number of single-character edits (insertions, deletions, substitutions, flips) required to change one word into the other.

4.1.3 Intuitiveness/Walk up Performance

The learning curve represents a performance measure over time to show how fast a method can be learned and how well it can be remembered. The walk up performance represents the beginning of this curve and is especially important for input methods aimed at a broad public. It can be estimated by measuring initial Effectiveness and Error Rate.

4.1.4 Expert Performance

Expert performance means the later parts of the learning curve. It represents the maximum performance that can be achieved with a certain method. It can be estimated by measuring Effectiveness and Error Rate over many sessions.

4.1.5 Physical Ease

Some (possible) text input methods require the user to have his or her arms lifted for a certain amount of time or to move a lot. This can result in so called "gorilla arms" and is considered as negative. Other methods lack ergonomics. Physical ease describes the contrary to such physical demands. It can be measured with questionnaires.
Bowman et al. suggest not to neglect user comfort in 3D user interfaces by avoiding heavy devices and those that force the user to hold the hand in a single posture for long periods of time [2].

4.1.6 Immersion/Presence/Awareness (utility)

Presence (a shortened version of the term "telepresence") is "a psychological state or subjective perception in which even though part or all of an individual's current experience is generated by and/or filtered through human made technology, part or all of the individual's perception fails to accurately acknowledge the role of the technology in the experience. Except in the most extreme cases, the individual can indicate correctly that he or she is using the technology, but at some level and to some degree, his or her perceptions overlook that knowledge and objects, events, entities, and environments are perceived as if the technology was not involved in the experience." [69]

Lombard and Ditton ([70]) identified six different explications of presence that have been used in the literature: Presence as

- Social Richness: The extent to which the medium is perceived as sociable, warm, sensitive, or personal when it is used to interact with other people
- Realism: The extent to which a medium can seem perceptual and/or socially realistic
- Transportation: The sensations of "you are there," "it is here," and/or "we are together"
- Immersion: The extent to which the senses are engaged by the mediated environment
- Social Actor: Within medium, the extent to which the user responds socially to a representation of a person through a medium
- Medium as Social Actor: The extent to which the medium itself is perceived as a social actor

However, presence as discussed in literature related to immersive VR can most often be characterized by the concept of presence as transportation: people are usually considered "present" in an immersive VR when they report a sensation of being in the virtual world ("you are there") [69]. The term co-presence or social presence is often reserved for the sense of being together in a virtual world ("we are together") [69].
Regarding text entry another point of view becomes interesting: Zahorik and Jenison ([71]) state that successfully supported actions in an environment will lead one to perceive oneself as existing in that environment, to a sense of presence. An action is said to be successfully supported when the result of that action is considered lawful: responses from the environment must be similar to those in the real-world environment in which our perceptual system evolved [69].
Presence can be measured with questionnaires. Whether it can contribute to better task performance is controversial [69].

4.1.7 Cyber/Motion Sickness stability

Unlike simpler media such as radio or television, VR has the power to overwhelm the senses and the brain, leading to fatigue or sickness [4]. This phenomenon has been studied under the heading simulator sickness for decades [4]. Sometimes the discomfort is due to problems in the VR hardware and low-level software; however, in most cases, it is caused by a careless VR developer who misunderstands or disregards the side effects of the experience on the user [4]. In many cases, fatigue arises because the brain appears to work harder to integrate the unusual stimuli being presented to the senses [4]. In some cases, inconsistencies with prior expectations, and outputs from other senses, even lead to dizziness and nausea [4]. Cyber/Motion sickness can be measured with questionnaires.

4.1.8 Fun, etc. (user Experience /subjective qualities)

Subjective criteria are hard to measure empirically but still important for design. If a method requires for example frequent correction of input it could increase frustration and decrease fun. User experience can be measured with questionnaires.

4.1.9 (Haptic) Feedback

Haptic feedback could be considered as question too, but is put as criteria here for one important reason: We consider it not as choice to implement haptic feedback if possible; In contrast to acoustic or visual feedback which could be distracting in some circumstances. Haptic feedback should in no case decrease one of the constraints mentioned in this chapter.
Bowman et al. state that haptic feedback is an important component of keyboard use in 3D User Interfaces and suggest to use keyboards with physical buttons if practical [2]. If using virtual keyboards, one should place the virtual keys on a physical surface [2].

4.1.10 Adaptiveness

Due to the existence of many different VR devices with many different native input possibilities, adaptiveness is an important factor. A text entry method that works for more devices is more adaptive than one that works for less.

4.1.11 Tracking Robustness

Some methods (like freehand) require better tracking than is yet available in current commercial products. The more robust a method is to such tracking issues, the better.

4.2 Questions and Possible Answers

This section shows the design space graph with questions that arise when thinking about text input design in VR, answers to these questions and how they probably influence certain criteria of those above in detail. The graph is visualized for each question separately. Some of the relations between answers and criteria are assumptions by the author and thus need to be evaluated (represented by red lines), some can be justified by scientific work done in the fields of text input and VR (represented by black lines) which will be done in this section too.

4.2.1 Use QWERTY layout?

As described in "The QWERTY keyboard: a review" by Jan Noyes [26] the QWERTY has been the standard keyboard of the last century, and is today "the de facto standard layout for communications and computer interface keyboards". Most users are familiar with its layout. This supports the assumption that using a QWERTY layout has a positive impact on intuitiveness and thus the walk up performance. Drew et al. ([54]) (and many other more [51]) have shown that the QWERTY layout is not at all superior for expert performance. The main argument for QWERTY has always been adaptiveness as man is a creature of habit. Not restricting a method to the QWERTY design will normally result in a lower walk up performance. However this is only entirely obvious for standard keyboard input. Maybe the new VR context could motivate users to learn new or better designs.
Bowman et al suggest to use the QWERTY layout in 3D user interfaces if symbolic input will be infrequent or if most users will be novices [2].
Figure 4.2 summarizes the question.

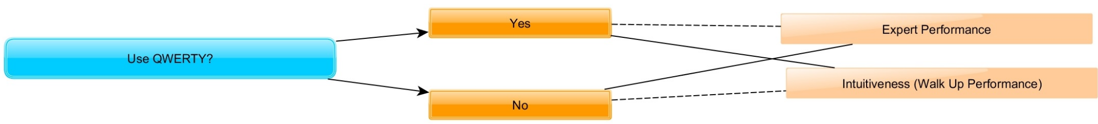


FIGURE 4.2: The Design Space visualisation for the question "Use QWERTY layout?". Questions are blue, answers orange and criteria rose. Straight lines between an answer and a criteria mean positive relation, broken lines mean negative relation. Red lines mean that this connection is an assumption by the author.

4.2.2 Which Input Device?

As mentioned before VR devices differ greatly in design and functionality. Some come with no input possibility at all, some with six DOF input devices, some with hand tracking and some with something in between. So the choice of input device for the text input method matters.

Some of the input methods require the user to lift his or her arms or hands for a certain up to a long time. These include freehand techniques or those based on the VR Headset like the touch pad for Gear VR. To use no extra input device at all like for example at selective methods by head or eye gaze could potentially increase physical ease. For other input devices it depends strongly on the method. As the traditional keyboard is used to type pretty ergonomicly for long times it has a positive impact on the factor physical ease.

Regarding intuitiveness the traditional keyboard is superior as people already know how to use it. Other unfamiliar input devices need to be understood and learned before they can be used efficiently.

We already know that handheld devices like the Twiddler hold the potential for superior expert performance. We've seen in the related work part that eye and head gaze as well as touchpad based interaction results in low overall performance.

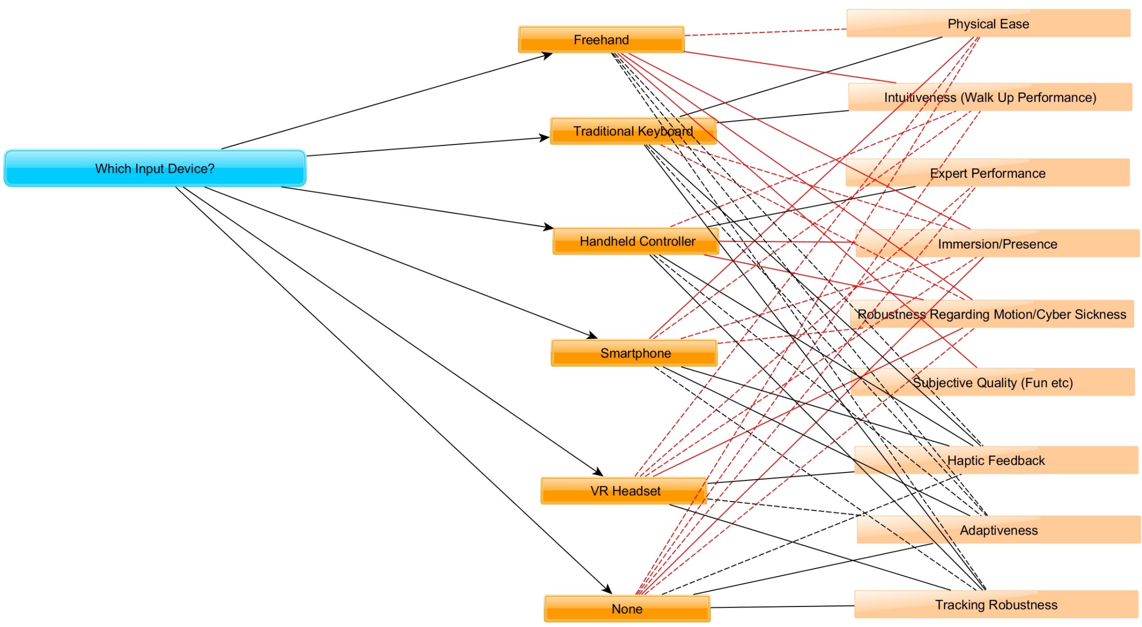


FIGURE 4.3: The Design Space visualisation for the questions "Which Input Device?". Questions are blue, answers orange and criteria rose. Straight lines between an answer and a criteria mean positive relation, broken lines mean negative relation. Red lines mean that this connection is an assumption by the author.

The most adaptive input device is the smartphone as it is completely detached from the VR device just the same as having no input device at all. Freehand and Headset based recognition could be included to all VR devices without much effort but some applications require other input forms.
Freehand as well as smartphone input both have the problem of low tracking accuracy which is a technical problem that my be solved in the future.
Figure 4.3 summarizes the question.

4.2.3 Selection or Recognition Based Approach?

Text input methods fall in one of these two categories each with its own advantages and drawbacks. Direct speech input for example is a recognition based approach in contrast to a QWERTY keyboard where the user selects keys to press.
As recognition based approaches rely heavily upon intelligent code they create another point of failure other than the user themselves. User input can be interpreted incorrectly or not at all which in most cases will result in a higher error rate compared to selection based techniques. However with effort it is possible to compete with selection based approaches as for example Chau et al showed ([2]), especially at expert performance. The walk up performance depends greatly on the learnability of the method. Gestures must be learned which will result in a lower intuitiveness.
Figure 4.4 summarizes the question.

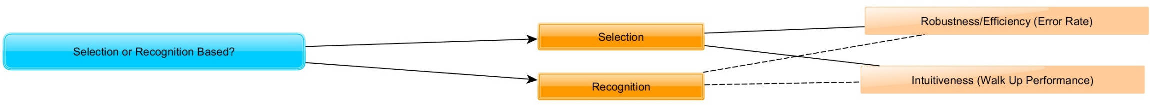


FIGURE 4.4: The Design Space visualisation for the questions "Selection or Recognition Based Approach?". Questions are blue, answers orange and criteria rose. Straight lines between an answer and a criteria mean positive relation, broken lines mean negative relation. Red lines mean that this connection is an assumption by the author.

4.2.4 2D or 3D

A user interface in a controlled digital environment like VR can make use of all three spatial dimensions or be reduced to less. It is questionable if all of the dimensions should be used for text input, a task that has successfully been a 2 dimensional one for decades. So we know a two dimensional text entry method will have a positive impact on Intuitiveness. Each dimension added introduces more complexity, but could potentially increase Immersion.

Two characteristics of input devices that are key in manipulation tasks are first, the number of control dimensions (how many DOF the device can control), and second, the integration of the control dimensions (how many DOF can be controlled simultaneously with a single movement) [2]. For example, a mouse allows for 2-DOF integrated control, magnetic trackers allow simultaneous control of both 3D position and orientation (i.e., 6-DOF integrated control) [2]. Typical game controllers, on the other hand, provide at least 4 DOF, but the control is separated - 2DOF allocated to each of two joy-sticks, where each has to be controlled separately [2].
The devices that are usually best for 3D manipulation are multiple DOF devices with integrated control of all input dimensions [2]. Integrated control allows the user to control the 3D interface using natural, well-coordinated movements, similar to real-world manipulation, which also results in better user performance [2]. But even when input devices like these are available it is (as stated above) questionable if all of the dimensions should be used for text input, a task that has successfully been a 2 dimensional one for decades.
Bowman et al. state that reducing the number of degrees of control in 3D user interfaces is especially crucial when high-precision interaction is needed [2] just like text entry requires. Figure 4.2 summarizes the question and its dependencies.

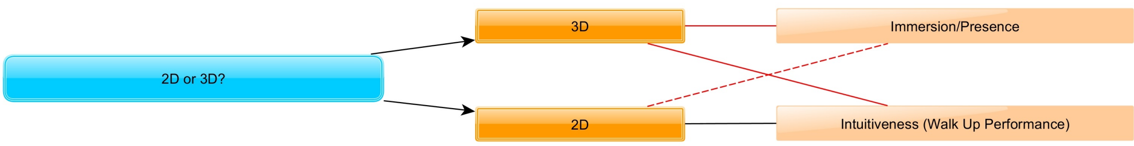


FIGURE 4.5: The Design Space visualisation for the question ”2D or 3D?”. Questions are blue, answers orange and criteria rose. Straight lines between an answer and a criteria mean positive relation, broken lines mean negative relation. Red lines mean that this connection is an assumption by the author.

4.2.5 Typing in relation to what?

Is the keyboard (or input representation) fixed and relative to the user or his or her surrounding? Steam VR ([34]) for example uses a keyboard fixed to the virtual world. So with a room tracking VR system the user can walk or punch through the keyboard representation. As no empirical study on VR specific text entry method covered this question yet it is hard to estimate the impact of this question on the criteria. Still it is fair to estimate that typing in relation to the head of the user could induce cyber sickness or at least be annoying as these GUI interaction elements are introduced as a

separate layer on top of the 3D world [2].
Figure 4.6 summarizes the question.

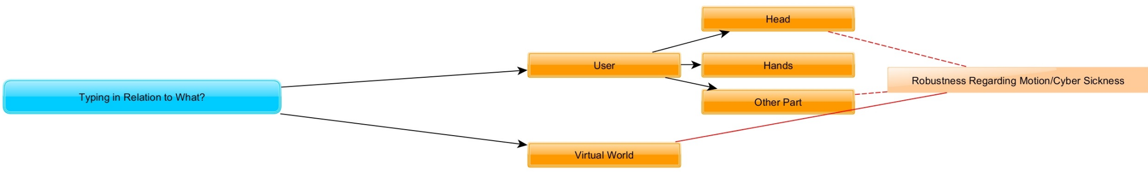


FIGURE 4.6: The Design Space visualisation for the questions "Typing in relation to what?". Questions are blue, answers orange and criteria rose. Straight lines between an answer and a criteria mean positive relation, broken lines mean negative relation. Red lines mean that this connection is an assumption by the author.

4.2.6 Position and size of text input representation and its customizability

The size of the input representation (like for example the QWERTY keyboard) matters especially for distant dependent text input methods but also for ergonomic reasons. Current text input methods for VR don't allow for position or size changes of the UI of the text input method.
Bachynskyi et al. ([72]) showed that these questions have no simple answer. They identified several clusters in input space for pointing gestures. Two of these clusters seem to fit the purpose of text entry (the clusters are evaluated with respect to the right arm):
"Cluster 2 covers short and middle-length movements in the lower right and central parts of the space in all directions and some long vertical movements in the middle part of the space. It exhibits better than average performance and optimal energy expenditure, which makes it suitable for the majority of interfaces that need long-term interaction.
Cluster 11 covers short and medium mostly vertical movements in the right part of the space. It can be used for medium-term interaction or for alternation between muscle loads." [72]
These findings advise to split the input space for right and left hand (if possible) and make the input representation fit the lower and peripheral input space. For one handed interaction this means that customizability of position (at least restricted) is necessary as the dominant hand differs from person to person.

Bachynskyi et al. ([72]) themselves also advise cluster 2 for a virtual keyboard (see figure 4.7).

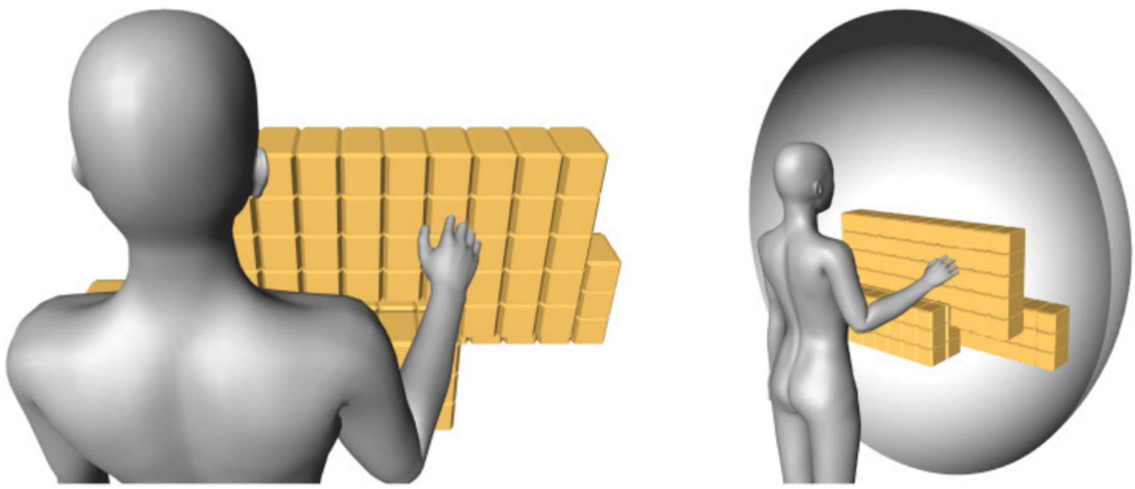


FIGURE 4.7: Keyboard position advised by Bachynskyi et al: The optimal input region for a virtual keyboard lies within Cluster 2. This cluster has good accuracy for aiming movements in a space that is large enough to host a virtual keyboard. Furthermore, the ergonomics in this cluster are very good: It has the lowest total normalized muscle activation. REPRINTED FROM [72]

Naturally, a larger input representation requires a less precise tracking technique but may afford quicker turning of the users head resulting in increased motion or cyber sickness. Immersion is driven by interactivity as stated before. This means the absence of customizability will reduce the immersed feeling of the user as he or she can not manipulate the virtual world like expected.
Figure 4.8 summarizes the question.

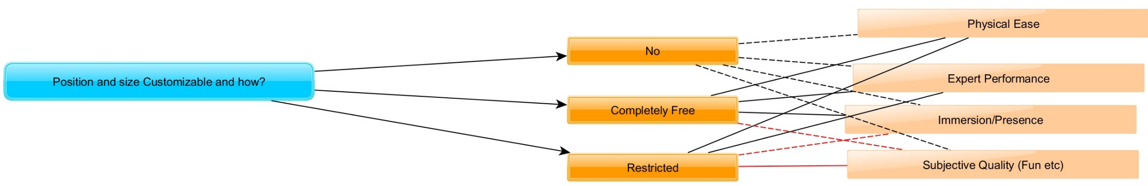


FIGURE 4.8: The Design Space visualisation for position and size of text input representation and its customizability. Questions are blue, answers orange and criteria rose. Straight lines between an answer and a criteria mean positive relation, broken lines mean negative relation. Red lines mean that this connection is an assumption by the author.

4.2.7 Visual Feedback?

Visual feedback like cursors, ray casting lines or pseudohaptics may be distracting or supportive. As visual feedback can explain new methods it may increase intuitiveness. Lesser visual feedback will result in more "trial and error"-like learning and decrease intuitiveness.

Figure 4.9 summarizes the question.

FIGURE 4.9: The Design Space visualisation for Visual Feedback. Questions are blue, answers orange and criteria rose. Straight lines between an answer and a criteria mean positive relation, broken lines mean negative relation. Red lines mean that this connection is an assumption by the author.

4.2.8 Use Word Completion Techniques?

It is controversial if word completion techniques make all text input methods perform better. They require the user to look at and choose from a list of proposed words (if it is not autocorrect, which has drawbacks too). The faster the text entry method the less benefit is gained from work completion. Different studies like [73] show no performance benefit. Still, as text entry methods in VR are currently comparatively slow the possibility of using word completion techniques should have a positive impact on Effectiveness and Error Rate; At least if the design does not force the user to use them. Figure 4.10 summarizes the question.

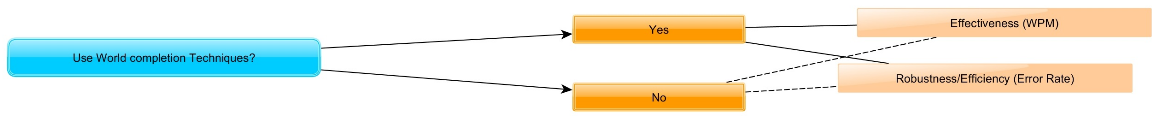


FIGURE 4.10: The Design Space visualisation for the question "Use Word Completion Techniques?". Questions are blue, answers orange and criteria rose. Straight lines between an answer and a criteria mean positive relation, broken lines mean negative relation. Red lines mean that this connection is an assumption by the author.

4.3 Derived Text Input Methods

To test for the unknown answers and how they affect the factors, some selective text input methods for VR were derived. In the following they are classified regarding the questions. Position and size are potentially customizable for all, visual feedback is provided and word completion techniques can be used with all methods. Especially the question "Which input device?" is captured by these methods, the question with the most red lines in the design space.

- Freehand Typing: The user is typing in relation to the virtual world as the keyboard is fixed in the air. This technique makes use of all three dimensions while using the QWERTY layout (pressing virtual buttons). One could, however, argue that it is still a 2D method.
- Pen Based: The user is typing in relation to the virtual world as the keyboard is fixed in the air. The relation can be changed to typing in relation to the users hand when he or she picks up the keyboard with one hand. In contrast to the freehand method the keys are pressed with a virtual representation of a hand held controller. This technique makes use of all three dimensions while using the QWERTY layout (pressing virtual buttons). One could, however, argue that it is still a 2D method.
- Pointing: This technique is using the QWERTY layout (pointing on a virtual keyboard) (2D). The user is typing in relation to the virtual world as the keyboard is fixed in the air.
- Touch Pad: This technique is using the QWERTY layout (controlling cursors on a virtual keyboard) (2D). The user is typing in relation to the virtual world as the keyboard is fixed in the air. The touch pad of a smartphone or a hand held controller can be used. Instead of a touch pad the joystick of a console controller can be used.
- Head Gaze: This technique is using the QWERTY layout (looking at keys on a virtual keyboard) (2D). The user is typing in relation to the virtual world as the keyboard is fixed in the air. The HMD itself is the input device.

We focus on selective text input methods because we think that walk up performance and intuitiveness is one of the most important factors for text entry in VR. This is due to the small amounts of text that need to be entered. In the design space above it is stated that selective text input methods may score better regarding this factor. A non selective text input method would be Chording for example. A modification of the

Twiddler with situational specific visual feedback would be a method of this kind and a potential good fit for expert performance in future VR applications that may include longer text entry.

Chapter 5

Implementation

The tested methods were implemented in C# with Unity3D 5.4.0 ([74]). All (except one) contained a 3D floating QWERTY Keyboard with four red dots at the edges to manipulate position, rotation and scale. When the user holds one dot (depending on the interaction method) the virtual keyboard becomes a child of the users hands or particular controller's representation in VR. He or she can then manipulate the position and the rotation of the keyboard until letting go. Grabbing another dot with the second hand or controller while holding a first one gives the opportunity to scale the keyboard. Rescaling and repositioning was not possible during the main study to ensure consistent data. Above the representation of the keyboard, a text field is shown where the entered text appears. Above that a stimulus (a sentence to copy) is presented. This stimulus is chosen randomly out of a corpus of short phrases. For the studies the corpus of Vertanen and Kristensson ([75]) was used.
The current text entry method is shown in the distance behind the keyboard. Except all that and a skybox showing a sunset, the virtual environment is empty. This follows related approaches ([5]).
Different acoustic feedback in form of short clicks is provided at the change of key selection and the activation of a key.
The Unity Assets for Leap Motion Orion Beta plugin[1] was used for Leap Motion VR support.
Eight text input methods were implemented overall:

- Chording: This non selective text input method was inspired by the promising chording keyboard "Twiddler" ([54]). To produce chords and corresponding key press events, both Vive controllers were used. Specifically, the following buttons were used on each controller: Trigger button, grip button and the depressible touch

[1] https://developer.leapmotion.com/unity

pad. The Touch pad is split into two buttons (up and down) to make sure that all 26 letters of the english alphabet plus some extra chords can be covered by the possible combination of two buttons. If that was not the case only 6*5=30 chords would have been possible. When none of the buttons are pressed, each is annotated with the letters that can be produced with this key in combination with another one. If the user presses a key, the visual feedback at the other buttons reduce to the one letter that is produced by pressing both: a chord of two keys. This kind of situationally adaptive visual feedback is not possible with the Twiddler ([54]). As this thesis concentrates on selective text entry the approach was abandoned before the first study was conducted.

- XBox Controller: As most gaming consoles use a selective text entry method via a controller this method was implemented. As all interaction types with this type of controller could easily be transferred to the Vive's controllers this approach was abandoned after the first study and replaced by two touch pad based methods:

- Touchpad: Two implemented methods made explicit use of the Vive controllers touch pads. The keyboard is split into two parts, left and right. The left part can be accessed with a cursor controlled by the left controller, the right part with one controlled by the right controller. To separate these two parts visually the keys of the left part are a little darker. One method allows control over the absolute position of the cursors on the keyboard by touching the pad with the thumb. The relative position of the thumb on the pad is mapped to the relative position of the cursor on its half of the keyboard. By pressing the pad the selected key is activated. This method is inspired by Steam VR ([35]). The other method requires the user to press the pad to move the cursor one key in the selected direction. Pressing the upper quarter of the pad results in the shifting of the cursor one key up and so on. The selected key is activated by pressing the Trigger button.

- Pointing: This method is inspired by Steam VR ([35]) and other VR text entry systems. When the user points a Vive controller towards the virtual keyboard a red cursor, with which to select keys, can be controlled. There is one cursor for each of the two controllers. To assist the user with distance dependency problems a gray ray between the cursor and the controller is shown as soon as the user points the controller towards the keyboard. By pressing the Trigger button the selected key is activated.

- Head Gaze: With this method the user can control the position of a cursor on the virtual keyboard by looking at, or rather pointing his or her head in the direction of it. The selected key can be activated by pressing the Trigger button of the HTC Vive controller.

- Pen Based: With this method the user can use the HTC Vive controllers as virtual pens. A small red dot is shown at the bottom of the controllers with which the user can press down the virtual keys on the keyboard. In contrast to the methods stated before this one requires some sort of "physical" manipulation. They keys really need to be pushed inside the keyboard to activate them.
- Freehand: This method resembles the pen based method but instead of using the digital representation of a controller to depress the buttons, a digital representation of the users hands and fingers is used. This representation is created with the help of the data provided by the Leap Motion ([19]) attached to the HMD. This implies that the user needs to look at his or her hands while typing.

Besides the methods themselves, a logging script can be switched on, which measures and saves the users motion of head and hands, as well as position, rotation and scale of the virtual keyboard and key press events. All this data is saved in two files: One that measures per frame and one per key press event. In the end this data will be used to calculate the values for WPM, error rate and KSPC automatically for each entered phrase and saved in a third file.

Chapter 6

Method

Two studies were conducted: A smaller pilot study and a bigger main study. This chapter describes the methods used to evaluate the considerations of the last chapters based on the main study.

6.1 Participants

Before the main study a smaller pilot study was conducted with four participants (one female and three male) with a mean age of 25.25. 15 participants (four female, eleven male) took part in the main study of which one had to abort because she did not feel well. The remaining were aged between 22 and 29 (M=25.14, SD=1.93). 85.7% preferred the German QWERTY keyboard layout, the rest 14.3% the US QWERTY layout. VR experience reached from 1 (low) to 5 (high) (M=2.14, SD=1.36). Novice users (VR experience < 3) made 64.3%, experienced users (VR experience $>= 3$) made 35.7%. It is normally distributed. All of them stated being able to read and copy english sentences (M=4.79, SD=0.78 with experience reaching from 1 (beginner) to 5 (native speaker)). 28.6% had a visual impairment like wearing glasses, but no participant was color blind. They were compensated with the possibility to play around in VR for half an hour before the session (In the first study participants could do it after the session). To compensate the participants before the session began made sure that they had a certain background in VR and were not too overwhelmed by the experience to be responsive. The same short Steam VR ([35]) tutorial was shown to all of the participants.

6.2 Apparatus

The experiment setup was nearly the same for both studies. The experiment was a within-subjects design where each subject tests each of six of the implemented methods: Pointing, Pen Based, Head Gaze, Freehand, Touch Pad and Discrete Touch Pad. Out of the tracked data the mean WPM, KSPC and error rates were computed for each participant and method. Together with the results for each question of the questionnaires that were answered per method, per participant one single CSV file was created. Both studies were conducted in the same tracked laboratory space. They differed in the tested methods.

6.2.1 Hardware and Setup

A room tracked HTC Vive system ([12]) was used for all tested methods. The tracking basis stations were installed about 2.5 meters above the ground and at opposite edges of the tracked space of about four x four meters (See figure 6.1).

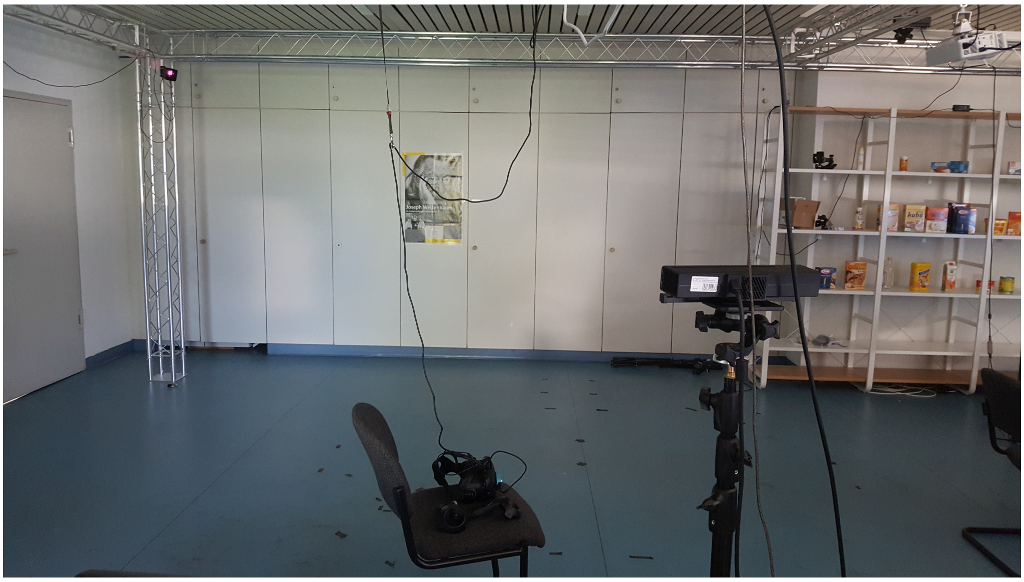


FIGURE 6.1: The Setup

For acoustic feedback the participants wore closed headphones (AKG Y55). The freehand interaction was realized with the hand tracking system Leap Motion ([19]), an infrared camera system that tracks the hands of the user via image recognition. It can be installed at the HMD so the hands can be tracked when held in front of the face. One method of the first study required a wireless controller. As the operating system used was Windows, a wireless XBox controller ([45]) was chosen. For the other methods the HTC Vive wireless hand held controllers were used. During the experiment the participants where filmed with a video camera.

6.3 Procedure and Task

Each participant was tested alone to ensure there were no interruptions and tested each method, which follows the setup of García et al [6]. Upon arriving the participants where offered sweets and cookies and asked to turn off their mobile phone. After that, the following was explained: the duration of experiment (ca 90 min), the rights of the participant, that breaks are possible at any time between the methods, the purpose of the experiment, the task, and the data that is recorded. The following was read to each participant:
"The purpose of the experiment is to understand how to best enter text in Virtual Reality. You will have to enter several short phrases and random words that are presented to you in VR. Please type as fast and as accurately as possible. We will record what you type, and how you move your hands and head while entering text. Plus we will film you."
After that the participant was asked to fill in an informed consent sheet (see appendix A). Then the participant was asked to put on the HMD and headphones and asked to stand up and confirm that he or she can see his or her environment in VR and everything feels correct. If not, adjustments were made until the participant felt well and comfortable.

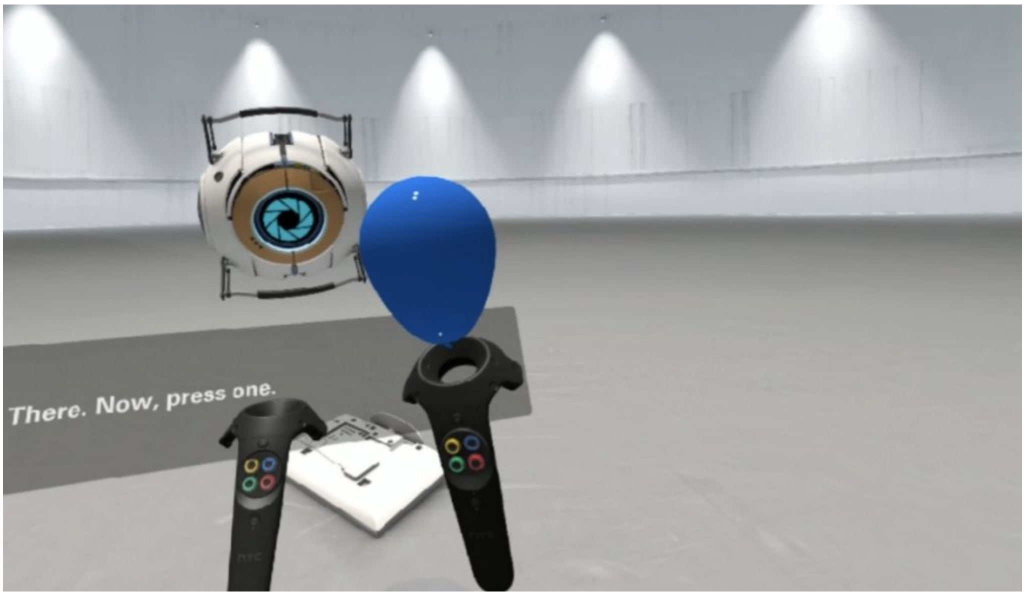


FIGURE 6.2: The SteamVR [35] tutorial

In the main study a short VR tutorial of about 10 minutes was was then presented to the participant. The Steam VR tutorial explains the bounding box that appears when the user is about to leave the tracked area to avoid him or her bumping into objects in the real world, the controllers and their buttons and how to use them (see figure 6.2).
After the tutorial the participant had the opportunity to explore the steam VR play store for VR games and experiences they could be interested in for 20 more minutes and was then asked to pull off the controllers, headphones, and HMD. The duration, breaks and task were then explained like this:
"There are 6 text input methods and you have to input 5 sentences with each. Please take off the head mounted display after each method to fill in a questionnaire about your experience.
Your task it to enter the text that is displayed above the virtual keyboard. Simply try to type as well as you can. If a sentence is shown read through it first and try to memorize it. Then type it from memory. At the end you should type enter to get to the next sentence."
After that the camera was turned on and 6 methods were tested after each other. In both studies Pointing, Headgaze, Freehand, Penbased and the Touchpad version in which the absolute position of the cursors can be controlled was tested. In the first study the XBox controller method was tested and replaced by the discrete Touchpad method in the main study. The procedure for each method were a following:

- Start method
- Explain it to the participant
- Put on the HMD and headphones
- Let user try out method until it was fully understood
- Start tracking
- Instruct: "You may begin to type. Remember to first memorize the sentence and then type as well as you can."
- At the end: Ask participant to take off the HMD and Headphones to fill in the questionnaires about the experience: SUS Questionnaire (See Appendix B [76]), Motion Sickness Assessment Questionnaire (See Appendix C [77]), User Experience Questionnaire (See Appendix D [78]), Nasa Task Load Index[1] ([79])
- Ask participant for remarks

[1]http://www.keithv.com/software/nasatlx/nasatlx.html

After all six methods were tested the user was asked to fill in a general questionnaire (See Appendix E) and the camera was switched off. In the first study the participant now had the opportunity to play around in VR as compensation (in the main study that was done before the experiment).
All the data captured for this participant was collected and saved in one place after the trial.

Chapter 7

Results

This chapter deals with the results of the studies in detail. Table 7.1 shows an overview of the results.

In the following we will look at the results in detail for each measure. The dataset will also be split into two groups and each test will be conducted again for each: Novice users (VR experience < 3) (64.3%) and experienced users (VR experience $>= 3$)(35.7%). VR experience reached from 1 (low) to 5 (high) with a mean of 2.14. It is normally distributed. We did this to check whether there are differences between these groups, which could be important for design.

Measure	Free-hand	Head Gaze	Pen Based	Poin-ting	Touch Pad	Discrete Touch Pad	Signifi-cance
WPM	9.35 (2.29)	10.83 (2.12)	13.89 (2.78)	16.74 (3.03)	8.79 (1.83)	5.43 (0.92)	0.000 (F=44)
KSPC	1.308 (0.182)	1.169 (0.125)	1.173 (0.100)	1.166 (0.228)	1.123 (0.079)	5.171 (1.514)	0.000
Error Rate	2.7% (3.2%)	0.9% (1.4%)	1.2% (1.9%)	0.9% (1.2%)	1.5% (1.7%)	3.5% (3.3%)	0.050
SUS Count	1.786 (1.528)	1.429 (1.342)	1.929 (1.940)	1.571 (1.423)	1.286 (1.684)	1.214 (1.762)	0.646
SUS Mean	4.321 (1.259)	4.381 (0.701)	4.559 (1.165)	4.596 (0.666)	4.488 (1.049)	4.011 (1.217)	0.703 (F=0.6)
MSAQ Over-all	0.160 (0.056)	0.200 (0.136)	0.140 (0.037)	0.139 (0.025)	0.153 (0.048)	0.145 (0.035)	0.801
MSAQ Gas-trointestinal	0.129 (0.030)	0.155 (0.073)	0.123 (0.026)	0.123 (0.021)	0.127 (0.028)	0.125 (0.032)	0.731
MSAQ Cen-tral	0.162 (0.093)	0.198 (0.156)	0.141 (0.053)	0.138 (0.044)	0.148 (0.053)	0.135 (0.040)	0.873
MSAQ Peri-fal	0.127 (0.035)	0.190 (0.099)	0.151 (0.073)	0.159 (0.076)	0.138 (0.049)	0.146 (0.064)	0.502
MSAQ So-pite Related	0.212 (0.095)	0.254 (0.225)	0.147 (0.044)	0.143 (0.031)	0.198 (0.200)	0.179 (0.078)	0.324
NASA Task Load Index	50.14 (19.60)	40.26 (21.89)	40.93 (19.58)	31.00 (18.34)	42.10 (20.02)	51.52 (18.52)	0.083
Mental Demand	36.07 (23.87)	33.93 (28.09)	27.14 (20.44)	29.29 (24.24)	35.00 (27.10)	41.43 (26.63)	0.697
Physical De-mand	42.86 (26.73)	41.43 (26.99)	47.14 (27.37)	25.36 (20.61)	22.86 (17.40)	22.50 (18.99)	0.013
Temporal Demand	38.93 (20.86)	42.86 (23.02)	42.50 (23.92)	41.07 (30.46)	36.07 (22.97)	45.00 (24.26)	0.927
Performance	44.64 (29.12)	32.86 (24.32)	32.14 (17.29)	23.21 (17.72)	45.00 (23.37)	53.93 (29.56)	0.030
Effort	57.50 (26.87)	37.14 (27.30)	43.93 (25.05)	25.36 (18.76)	42.86 (24.62)	54.29 (23.28)	0.017
Frustration	56.79 (30.92)	41.07 (34,32)	31.07 (28.02)	24.64 (23.57)	47.50 (28.06)	63.57 (26.42)	0.005
Attrac- tive-ness	0.048 (1.791)	0.667 (1.712)	0.893 (1.329)	1.619 (0.888)	0.000 (1.228)	-1.107 (1.382)	0.000
Perspicuity	0.643 (0.758)	1.554 (0.511)	1.286 (0.678)	1.625 (0.517)	0.464 (0.837)	0.911 (0.725)	0.000
Efficiency	0.018 (1.439)	1.071 (1.501)	0.804 (1.066)	1.304 (1.101)	0.143 (1.188)	-0.893 (1.184)	0.000
Dependa-bilty	-0.036 (1.704)	0.893 (0.745)	1.018 (1.215)	1.339 (0.738)	0.321 (0.787)	0.214 (0.924)	0.006
Stimulation	0.768 (1.543)	0.929 (1.399)	0.875 (1.259)	1.429 (1.021)	0.018 (1.325)	-1.411 (1.103)	0.000
Novelty	1.161 (1.385)	1.107 (1.247)	0.696 (1.305)	0.893 (1.281)	0.357 (1.018)	-1.161 (0.824)	0.000
UEQ Overall	0.434 (1.282)	1.037 (0.929)	0.929 (0.912)	1.368 (0.679)	0.217 (0.865)	-0.574 (0.748)	0.000

TABLE 7.1: Overview of results. The values for the methods are the averages for the measure for the method. The values in braces are the standard deviation. If the last cell in a row is green it means that the differences between the methods were significant with a large effect (effect size >0.5); green yellow means a medium (effect size >0.3), yellow a minor (effect size >0.1) effect.

7.1 Words Per Minute (WPM)

The words per minute (WPM) value per participant per method was calculated as the mean WPM of the five entered sentences.
We found a significant difference between the input methods for WPM.
As shown in figure 7.1 average WPM ranged between 5.43 (Discrete Touch Pad) and 16.74 (Pointing).

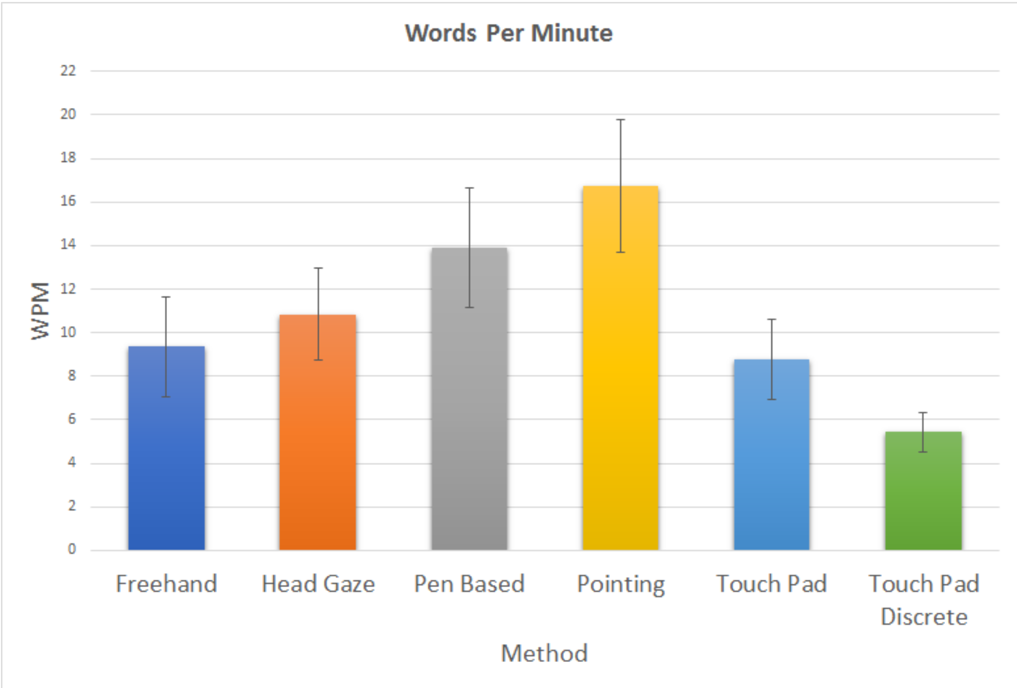


FIGURE 7.1: Overview of means of WPM. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

1-way ANOVA showed a significance between the differences (F = 43.787, p=0.000) with an effect size of 0.737, which is a large effect. Bonferroni corrected pairwise comparisons showed significant differences between all methods but Freehand and Head Gaze, Freehand and Touch Pad, Head Gaze and Touch Pad. This means Freehand, Touch Pad and Head Gaze can be seen as one group with relation to WPM.
1-way ANOVA for novice users showed a significance between the differences (F = 28.612, p=0.000) with an effect size of 0.749, which is a large effect. Bonferroni corrected pairwise comparisons showed significant differences between all methods but Freehand and Head Gaze, Freehand and Touchpad, Head Gaze and Pen Based, Head Gaze and Touch Pad, Pen Based and Pointing as well as Touch Pad and Discrete Touch Pad.
As the data was not normally distributed for expert VR users, Kruskall Wallis was conducted which showed a significance between the differences (p=0.000) with an effect size of 0.793, which is a large effect. Bonferroni corrected pairwise comparisons showed

significant differences between Discrete Touch Pad and Pen Based as well as Discrete Touch Pad and Pointing.

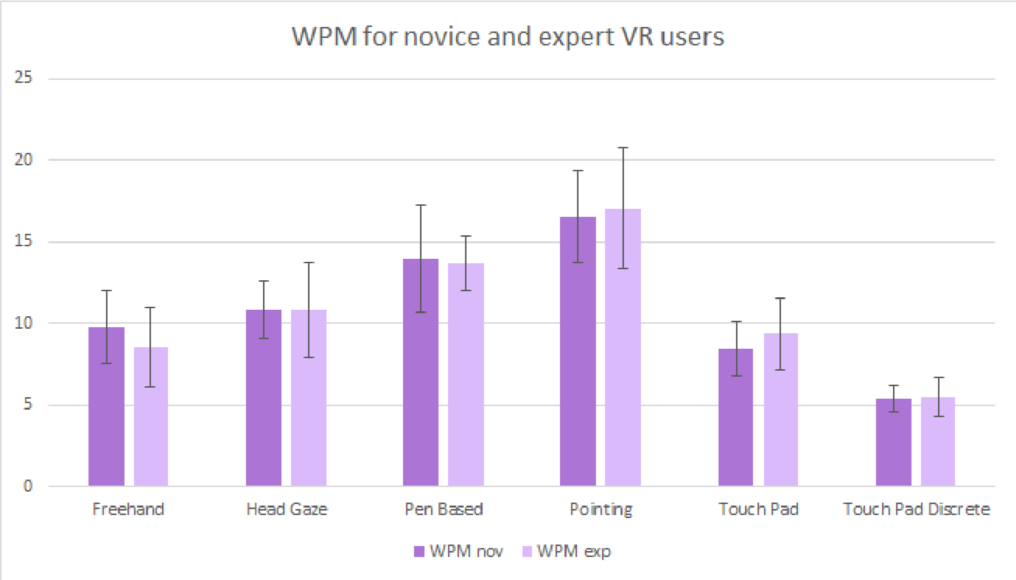


FIGURE 7.2: WPM for novice and expert VR users. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

2-Way Anova showed no significant difference regarding VR experience (see figure 7.2).

7.2 Keystrokes Per Character (KSPC)

The Keystrokes Per Character (KSPC) value per participant per method was calculated as the mean WPM of the five entered sentences.
We found a significant difference between the input methods for KSPC.
As shown in figure 7.3 average KSPC ranged between 1.123 (Touch Pad) and 5.171 (Discrete Touch Pad).
As Head Gaze, Pointing and Touch Pad were not normally distributed we conducted a Kruskal Wallis test ($p = 0.000$, effect size: 0.503, which is a large effect). Bonferroni corrected pairwise comparisons showed significant differences between Discrete Touch Pad and the other methods. The other methods showed no pairwise significant difference. This means that Discrete Touch Pad is the single worst method regarding KSPC.

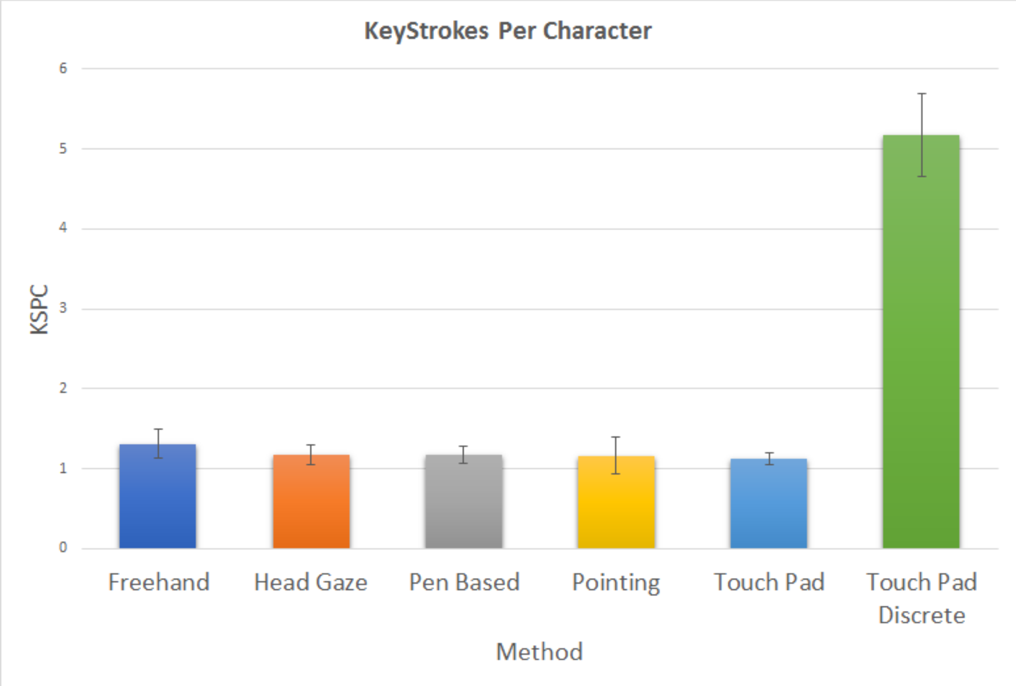


FIGURE 7.3: Overview of means of KSPC. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

As the data was not normally distributed for VR novices, Kruskall Wallis was conducted which showed a significance between the differences (p=0.000) with an effect size of 0.488, which is a medium effect. Bonferroni corrected pairwise comparisons showed significant differences between Discrete Touch Pad and all the other methods except Freehand (p = 0.55). As the data was not normally distributed for expert VR users, Kruskall Wallis was conducted which showed a significance between the differences (p=0.002) with an effect size of 0.644, which is a large effect. Bonferroni corrected pairwise comparisons showed significant differences between Discrete Touch Pad and the methods Touch Pad, Head Gaze and Pointing.

2-Way-Anova showed no significant difference regarding VR experience (see figure 7.4).

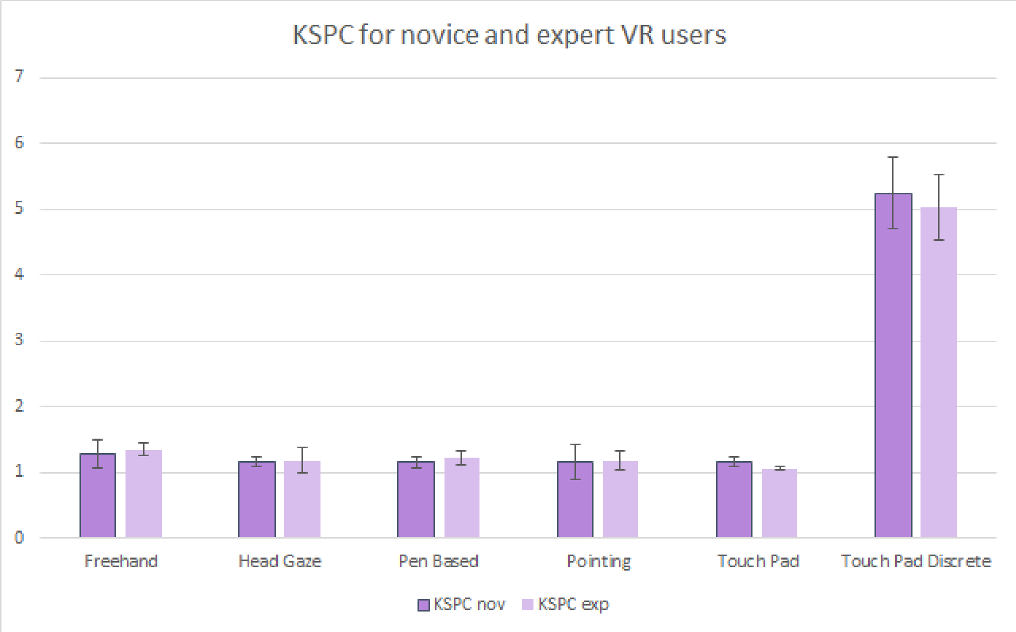


FIGURE 7.4: KSPC for novice and expert VR users. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

7.3 Error Rate (Minimum String Distance)

The error rate (minimum string distance) value per participant per method was calculated as the mean WPM of the five entered sentences.
We found no significant difference between the input methods for error rate.
As shown in figure 7.5 average error rate ranged between 0.009 (Pointing) and 0.035 (Discrete Touch Pad).
As the data was not normally distributed we conducted a Kruskal Wallis test ($p = 0.050$, effect size: 0.133, which is a minor effect). Bonferroni corrected pairwise comparisons showed no significant differences, even for a significance level of 0.6. More observations would be needed to confirm an effect.

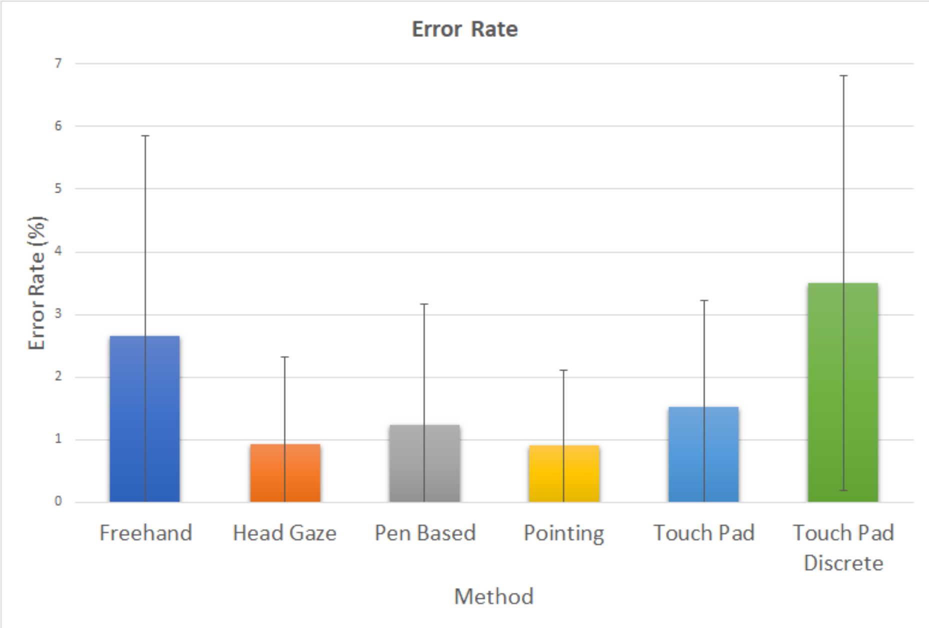


FIGURE 7.5: Overview of means of error rate. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

As the data was not normally distributed for novice VR users, Kruskall Wallis was conducted which showed no significance between the differences (p=0.178).
As the data was not normally distributed for expert VR users, Kruskall Wallis was conducted which showed no significance between the differences (p=0.219).
The data shows that expert VR users made more mistakes than novices (see figure 7.6), still 2-Way-Anova showed no significant difference regarding VR experience.

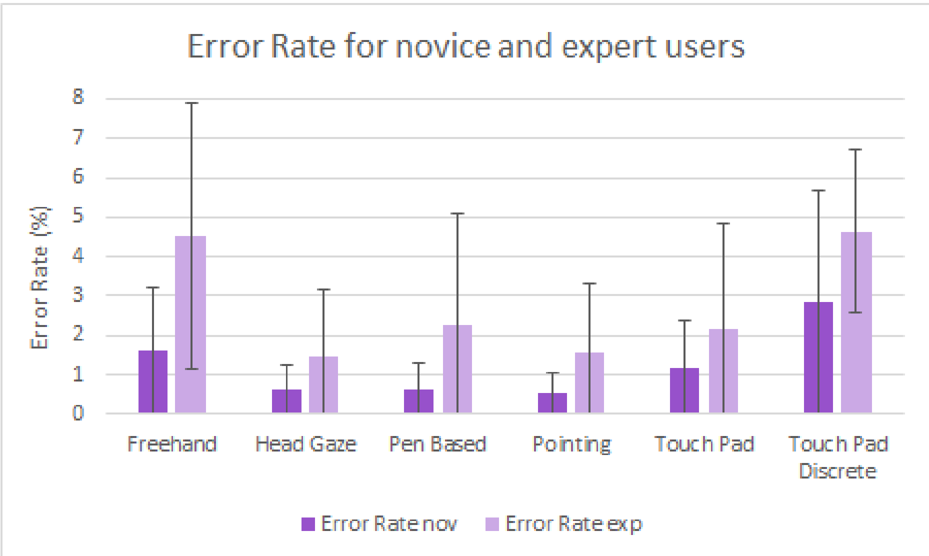


FIGURE 7.6: Error Rate for novice and expert VR users. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

7.4 Slater Usoh Steed (SUS) Questionnaire

This questionnaire gave six values between one and seven for each question per participant and method. The "SUS Count" and the "SUS Mean" were calculated using these 6 values. SUS Count is the count of '6' or '7' scores amongst the six questions, the SUS Mean is the mean of the values.
We found no significant difference between the input methods for SUS Immersion, for neither mean nor for count.
As shown in figure 7.7 average SUS count ranged between 1.21 (Discrete Touch Pad) and 1.92 (Pen Based) and average SUS mean ranged between 4.01 (Discrete Touch Pad) and 4.60 (Pointing).

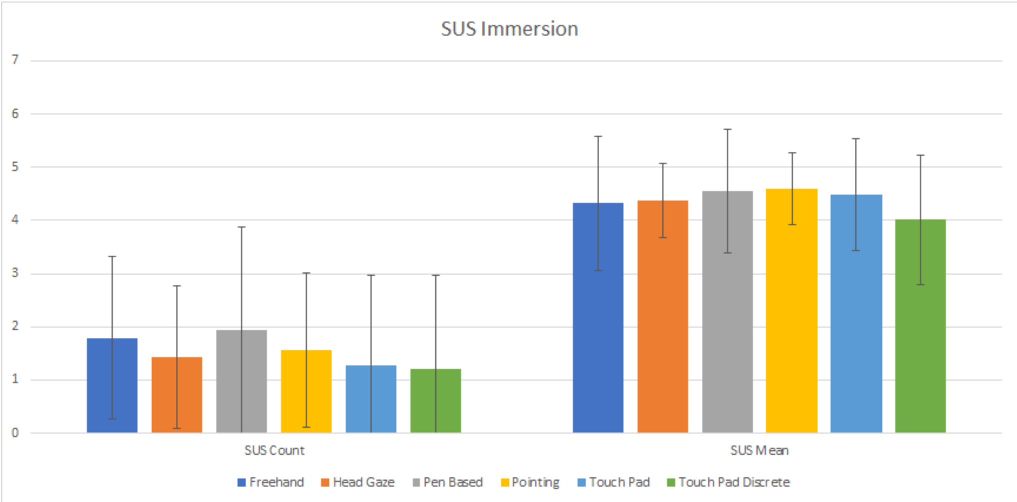


FIGURE 7.7: Overview of means of SUS Immersion. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

1-way ANOVA showed no significance between the differences for mean (F = 0.596, p=0.703) with an effect size of 0.037. Count was not normally distributed. Kruskal Wallis showed no significance between the differences for count (p=0.646) with an effect size of 0.040. This means that the text entry method had no effect on immersion.
As the data was not normally distributed for novice VR users, Kruskall Wallis was conducted which showed no significance between the differences (p=0.439 for mean, p=0.495 for count).
As the data was not normally distributed for expert VR users, Kruskall Wallis was conducted which showed no significance between the differences (p=0.713 for mean, p=0.627 for count).

2-Way-Anova showed that the means show no significant difference regarding VR experience (see figure 7.8).

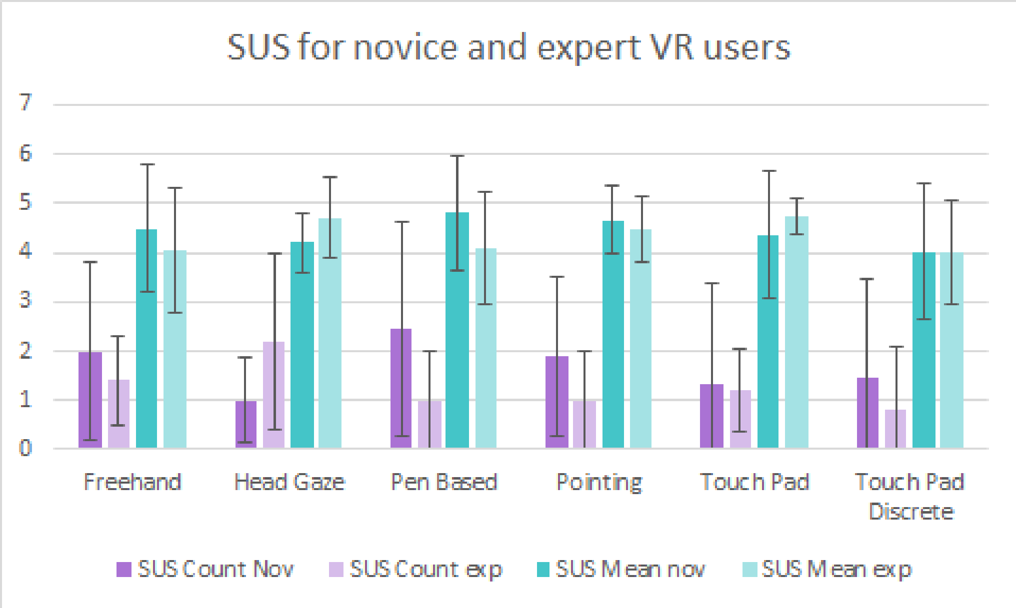
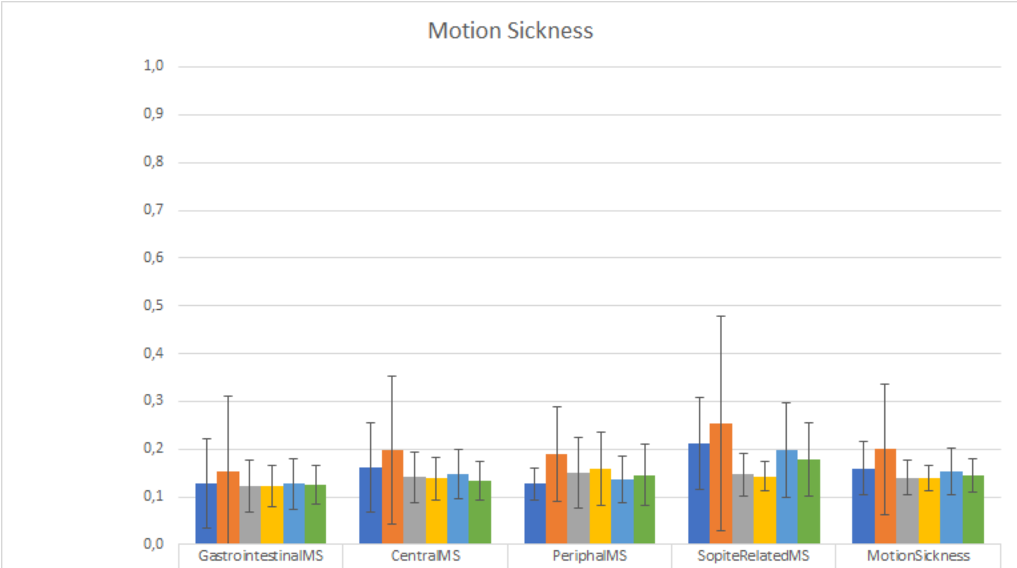


FIGURE 7.8: SUS Immersion for novice and expert VR users. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

7.5 Motion Sickness Assessment Questionnaire (MSAQ)

This questionnaire gave 16 values between one and nine for each question for each participant and method. Out of these 16 values the overall motion sickness score is obtained by calculating the percentage of total points scored: (sum of points from all items/144) x 100. Subscale scores are obtained by calculating the percent of points scored within each factor: (sum of gastrointestinal items/36) x 100; (sum of central items/45) x 100; (sum of peripheral items/27) x 100; (sum of sopite-related items/36) x 100.
Gastrointestinal items are: "I felt sick to my stomach", "I felt nauseated", "I felt queasy" and "I felt as if I may vomit".
Central items are: "I felt faint-like", "I felt dizzy", "I felt lightheaded", "I felt like I was spinning" and "I felt disoriented".
Peripheral items are: "I felt sweaty", "I felt hot/warm" and "I felt clammy/cold sweat".
Sopite-related items are: "I felt tired/fatigued", "I felt annoyed/irritated", "I felt drowsy" and "I felt uneasy".
We found no significant difference between the input methods for motion sickness

As shown in figure 7.9 average motion sickness ranged between 0.139 (Pointing) and 0.200 (Head Gaze).



	GastrointestinalMS	CentralMS	PeriphalMS	SopiteRelatedMS	MotionSickness
Freehand	0,129	0,162	0,127	0,212	0,160
Head Gaze	0,155	0,198	0,190	0,254	0,200
Pen Based	0,123	0,141	0,151	0,147	0,140
Pointing	0,123	0,138	0,159	0,143	0,139
Touch Pad	0,127	0,148	0,138	0,198	0,153
Touch Pad Discrete	0,125	0,135	0,146	0,179	0,145

FIGURE 7.9: Overview of means of MSAQ. Values range from 0 (no) to 1 (highest) motion sickness. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

As the data was not normally distributed we conducted the Kruskal Wallis test. It showed no significance between the differences for overall value (p = 0.801, effect size = 0.010) neither for the factors Gastrointestinal (p = 0.731, effect size = 0.034), Central (p = 0.873, effect size = 0.025), Peripheral (p = 0.502, effect size = 0.006) or Sopite Related (p = 0.324, effect size = 0.004). This means that the text input method has no significant effect on motion sickness. Either way it is noticeable that for each factor Head Gaze has the highest mean value and gastrointestinal factors have the most effect.
As the data was not normally distributed for novice VR users, Kruskall Wallis was conducted which showed no significance between the differences for overall value (p = 0.973) neither for the factors Gastrointestinal (p = 0.751), Central (p = 0.966), Peripheral (p = 0.965) or Sopite Related (p = 0.679).
As the data was not normally distributed for expert VR users, Kruskall Wallis was conducted which showed no significance between the differences for overall value (p = 0.690) neither for the factors Gastrointestinal (p = 0.686), Central (p = 0.753), Peripheral (p = 0.317) or Sopite Related (p = 0.382)).

2-Way-Anova showed that the means show no significant difference regarding VR experience (see figure 7.8).

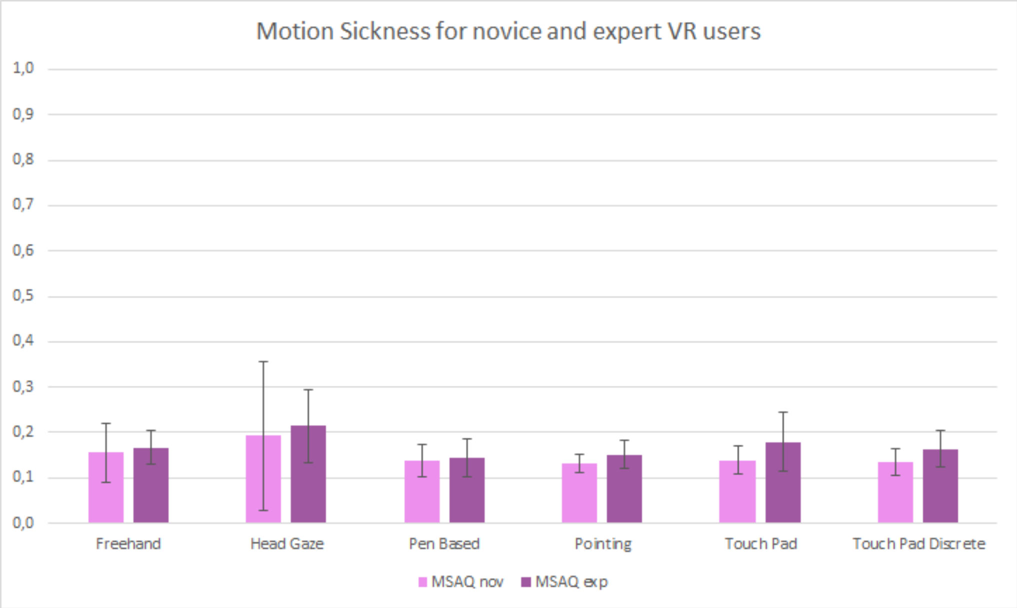


FIGURE 7.10: MSAQ for novice and expert VR users. Values range from 0 (no) to 1 (highest) motion sickness. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

7.6 User Experience Questionnaire (UEQ)

The UEQ contains 6 scales with 26 items overall. The Attractiveness scale has 6 items, all other scales have 4 items. The Figure 7.11 shows the assumed scale structure of the UEQ and the English items per scale. The range of the scales is between -3 (horribly bad) and +3 (extremely good).

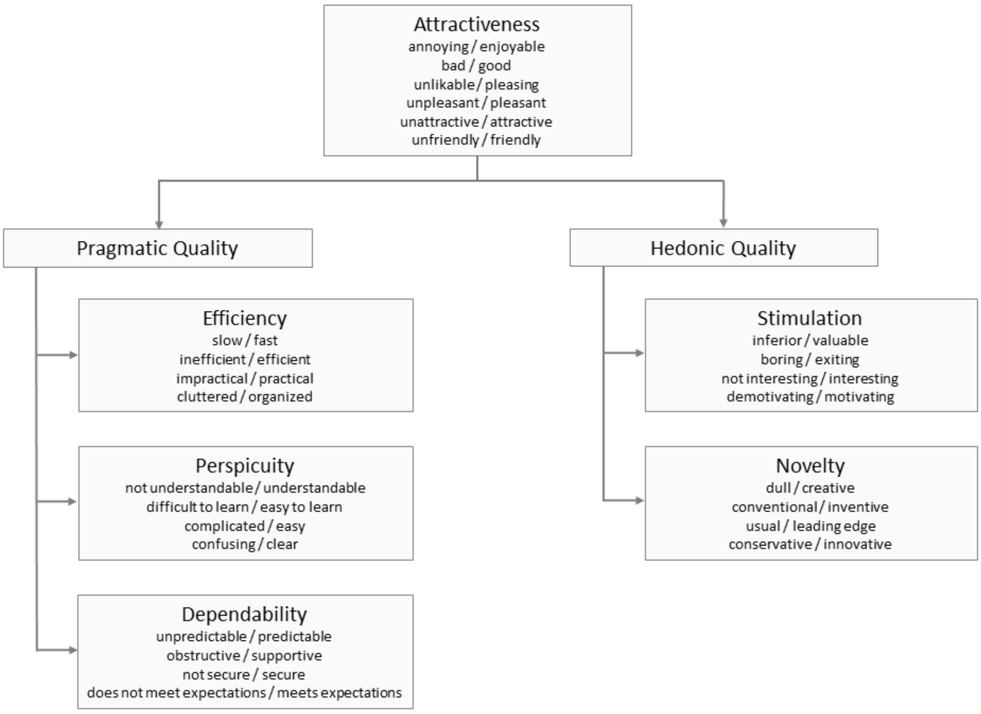
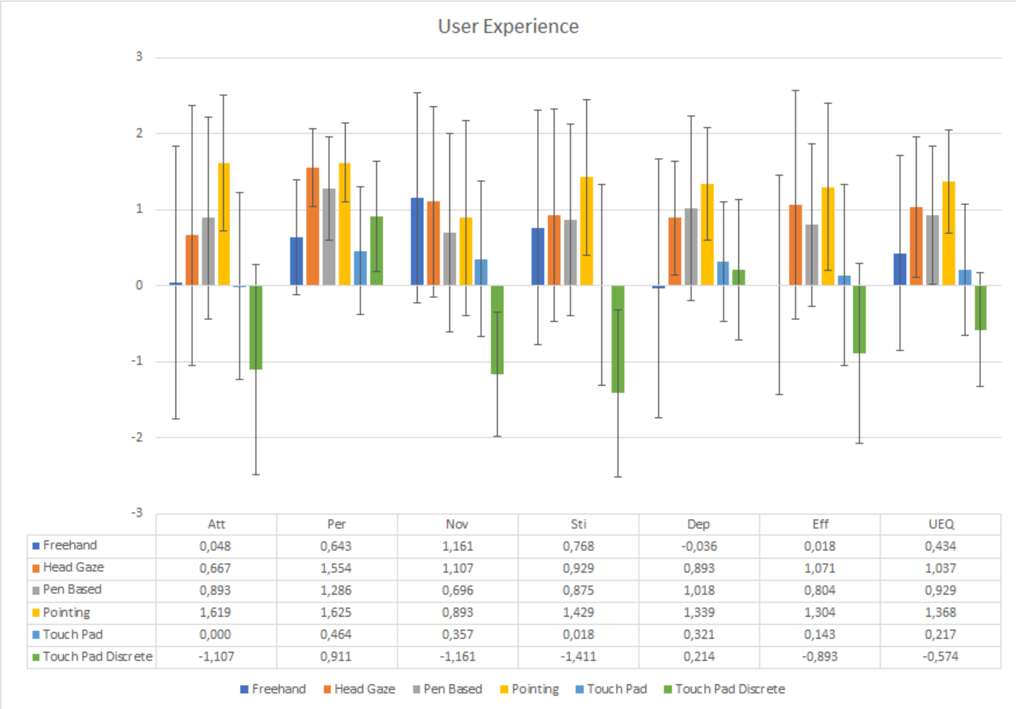


FIGURE 7.11: Assumed scale structure of the UEQ. [78]

We found a significant difference between the input methods for user experience
As shown in figure 7.12 average User experience ranged between -0.574 (Discrete Touch Pad) and 1.368 (Pointing).



	Att	Per	Nov	Sti	Dep	Eff	UEQ
Freehand	0,048	0,643	1,161	0,768	-0,036	0,018	0,434
Head Gaze	0,667	1,554	1,107	0,929	0,893	1,071	1,037
Pen Based	0,893	1,286	0,696	0,875	1,018	0,804	0,929
Pointing	1,619	1,625	0,893	1,429	1,339	1,304	1,368
Touch Pad	0,000	0,464	0,357	0,018	0,321	0,143	0,217
Touch Pad Discrete	-1,107	0,911	-1,161	-1,411	0,214	-0,893	-0,574

FIGURE 7.12: Overview of means of UEQ. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

As data was not normally distributed, Kruskal Wallis was conducted. It showed significance between the differences of the overall UEQ value (p = 0.000) as well as over all the sub values (p = 0.000 except for Dependability (p = 0.006)). Effect sizes were 0.325 for the overall value which is a medium effect, 0.277 for Attractiveness which is a minor effect, 0.367 for Perspicuity which is a medium effect, 0.315 for Novelty which is a medium effect, 0.326 for Stimulation which is a medium effect, 0.196 for Dependability which is a minor effect and 0.276 for Efficiency which is a minor effect. ANOVAS of all the sub values showed significances too. Bonferroni corrected pairwise comparisons of Effectiveness showed significant differences between Discrete Touch Pad and Head Gaze as well as Discrete Touch Pad and Pointing. For Perspicuity there was a pairwise significance between Freehand and Head Gaze, Freehand and Pointing, Head Gaze and Touch Pad, Pen Based and Touch Pad, Pointing and Touch Pad. For Dependability there was a pairwise significance between Freehand and Pointing as well as Discrete Touch Pad and Pointing. For Stimulation there was a pairwise significance between Discrete Touch Pad and all the other methods except for Touch Pad. Same for Novelty. For Attractiveness there was a pairwise significance between Discrete Touch Pad and Pointing as well as Discrete Touch Pad and Pen Based. For User Experience overall there was a significant difference between Discrete Touch Pad and Pen Based, Discrete Touch Pad and Freehand, Discrete Touch Pad and Pointing as well as Touch Pad and Pointing.

UEQ gives a classification of the resulting means based on the evaluation of many other (not restricted to VR) products. This classification is shown in figure 7.13

FIGURE 7.13: UEQ Classification

As the data was not normally distributed for novice VR users, Kruskall Wallis was conducted which showed significance between the differences of the overall UEQ value ($p = 0.001$) as well as over all the sub values Efficiency ($p = 0.002$), Perspicuity ($p = 0.000$), Dependability ($p = 0.022$), Stimulation ($p = 0.001$), Novelty ($p = 0.001$) and Attractiveness ($p = 0.002$).
As the data was not normally distributed for expert VR users, Kruskall Wallis was conducted which neither showed significance between the differences of the overall UEQ value ($p = 0.391$) nor over any the sub values Efficiency ($p = 0.470$), Perspicuity ($p = 0.757$), Dependability ($p = 0.632$), Stimulation ($p = 0.462$), Novelty ($p = 0.224$) and Attractiveness ($p = 0.329$).

The means show that novices rated some methods better than experts. (see figure 7.14). Still 2-Way-Anova showed that the means show no significant difference regarding VR experience (see figure 7.8).

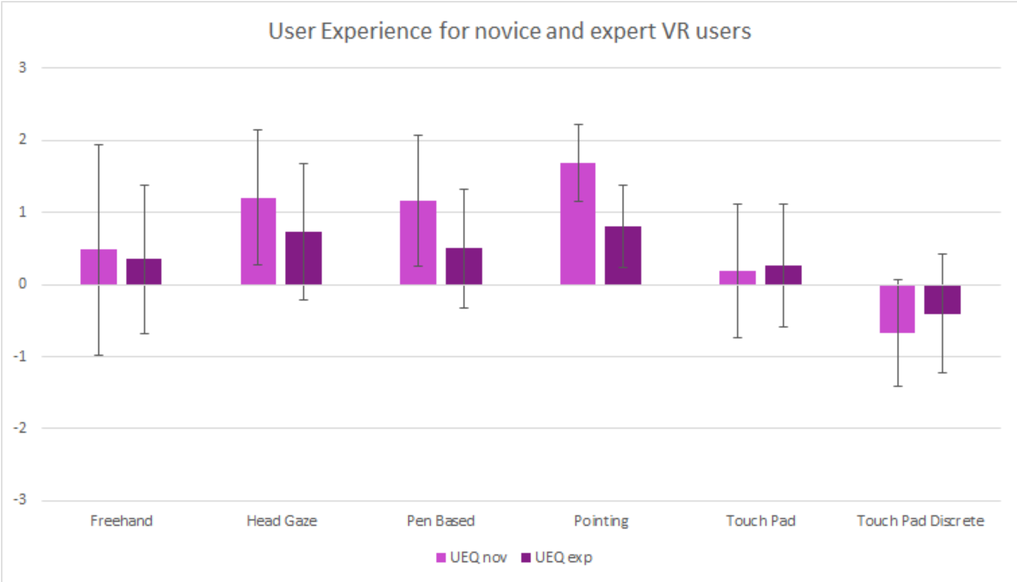
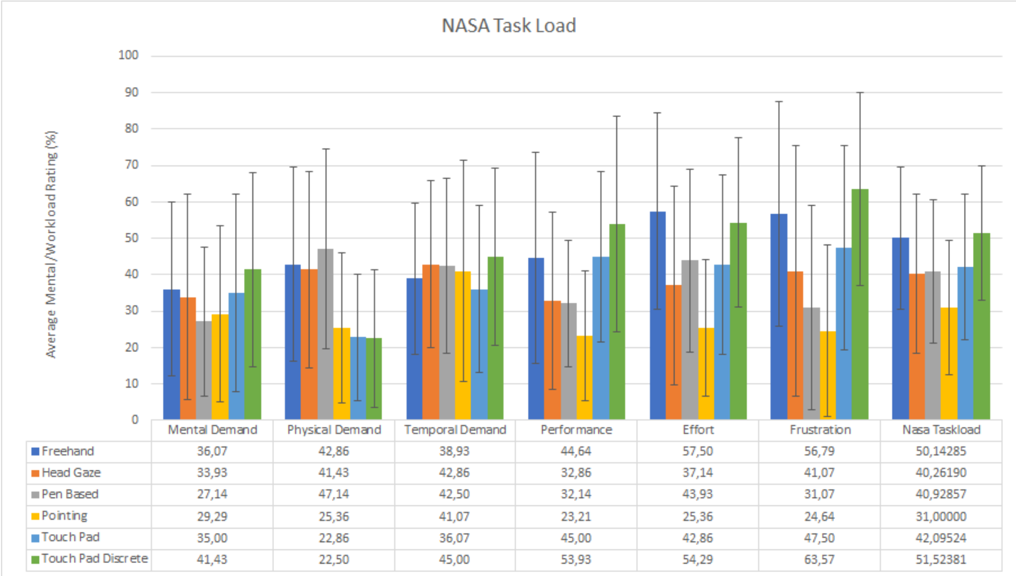


FIGURE 7.14: UEQ for novice and expert VR users. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

7.7 NASA Task Load Index

The NASA Task Load Index gives six sub scales with value and rating: Mental demand, physical demand, temporal demand, performance, effort and frustration. The values multiplied with their rating added up give the overall NASA Task Load Index.
We found significant differences between the input methods for NASA task load index. As shown in figure 7.15 average task load ranged between 31.00 (Pointing) and 51.52 (Discrete Touch Pad).



	Mental Demand	Physical Demand	Temporal Demand	Performance	Effort	Frustration	Nasa Taskload
Freehand	36,07	42,86	38,93	44,64	57,50	56,79	50,14285
Head Gaze	33,93	41,43	42,86	32,86	37,14	41,07	40,26190
Pen Based	27,14	47,14	42,50	32,14	43,93	31,07	40,92857
Pointing	29,29	25,36	41,07	23,21	25,36	24,64	31,00000
Touch Pad	35,00	22,86	36,07	45,00	42,86	47,50	42,09524
Touch Pad Discrete	41,43	22,50	45,00	53,93	54,29	63,57	51,52381

FIGURE 7.15: Overview of means of NASA Task Load Index. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

As some of the data is not normally distributed, Kurskal Wallis was conducted too and showed no significance for the overall value (p = 0.083, effect size = 0.117) but for physical demand (p = 0.013, effect size = 0.174 which is a minor effect), performance (p = 0.030, effect size = 0.149 which is a minor effect), effort (p = 0.017, effect size = 0.116 which is a minor effect) and frustration (p = 0.005, effect size = 0.201 which is a minor effect). Either way it is noticeable that pointing has the lowest overall task load which can be seen as a surprising result, as this method requires the lifting of the users arms. For this reason and for the reason that the NASA task load value itself is calculated by rating the intermediate results it is worth taking a closer look at them. Bonferroni corrected pairwise comparisons showed significant differences between Pointing and Discrete Touch Pad for performance, between Pointing and Freehand for effort as well as Pointing and Discrete Touch Pad for frustration.
As the data was not normally distributed for novice VR users, Kruskall Wallis was conducted which showed significance between the differences of the overall value (p = 0.029), for physical demand (p = 0.002), performance (p = 0.018), effort (p = 0.019) and frustration (p = 0.007).
As the data was not normally distributed for expert VR users, Kruskall Wallis was conducted which neither showed significance between the differences of the overall value (p = 0.890) nor for the subvalues.

2-Way-Anova showed that novices rated the methods significantly less physical demanding, but not depending on the method. (see figure 7.16).

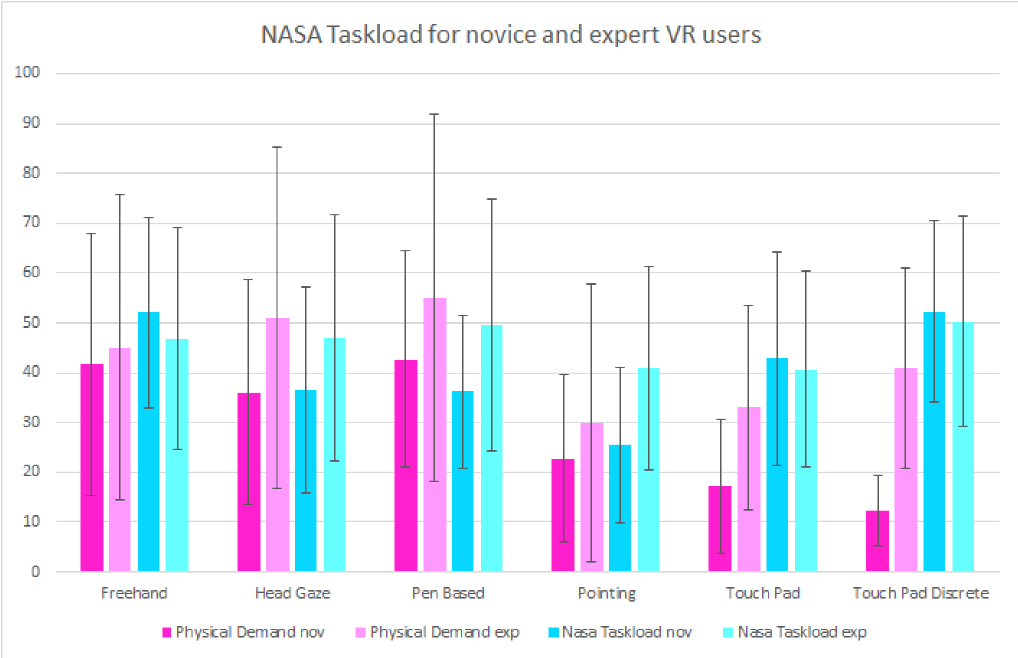


FIGURE 7.16: NASA Task Load Index for novice and expert VR users. Error bars represent the standard deviation. (The interval is twice the standard deviation.)

7.8 Observations

Each participant could remark upon his or her experience. These remarks were notated in addition to annotations concerning the way the participant interacted with the text input methods. Things like the way he or she held his or her arms, the controllers, how and how much he or she moved, etc. These annotations can be reproduced with the video recordings.

7.8.1 Body Posture

Most participants hold the controllers close to their bodies and their elbows in a right angle for all methods but the ones that required direct manipulation of the virtual keyboard. One user however let his hands hang during the touch pad methods. Another one "shot from the hip" during Pointing. Two participants used just one hand during Freehand and two just one controller during Pen Based. For Pen Based there were three ways of holding and using the controller: like a drumstick, a pen (equally distributed) and like a wand (just one participant). The difference between the drumstick and the wand is the way of interacting. While the drumstick users hit the buttons like using a stick the wand users ”stabbed” them.

7.8.2 Technical Difficulties

Recognition problems of the Leap Motion resulted in extreme difficulties with nearly all participants using Freehand. During the first informal study the application crashed a few times because the HTC Vive’s tracking cameras were installed too far apart. For the main study, this issue was corrected and all other methods besides Freehand showed no technical difficulties.

7.8.3 Audio Visual Feedback

One participant stated that the acoustic feedback was not helpful but annoying. Some other ones were only aware of it after asked if the headphones are working. Another one stated after the experiment that he felt claustrophobic as the virtual room around him was all black even though it was not. Two participants remarked that the floating keyboard was a slight immersion break for them. They would have preferred to have it installed on a virtual representation of a desk for example.

Chapter 8

Discussion and Future Work

This chapter discusses the results of the studies in detail.
Results showed that the VR experience, as expected, had no significant effect on most measures. Either way novice users rated all methods significantly less physical demanding than expert users, maybe because the overwhelming new feeling of VR made them forget the "hard" parts about it more easily. Still, the method had no impact on that. Experts average error rate was twice as high as the error rate of novices for all methods. Still WPM were pretty much the same. This means experts performed worse. This could be because more experienced VR users expect a user interface to run seamlessly and don't tolerate as many difficulties.
It is noticeable that no method was rated better than "good" at the UEQ with Discrete Touch Pad being rated worse than the classification captured. What is important to consider here is that this classification is based on the evaluation of many other (not restricted to VR) products and user interfaces, including web, mobile and desktop interfaces. These interfaces and products are highly investigated in detail. It may be that VR applications are rated worse in general. Looking at figure 7.13, it is clear to see, that the only considerable methods are Pointing, Head Gaze and Pen Based. As Pen Based uses the same input device as Pointing and performed worse in all other measures it can be dropped (Even though Pen Based performed a lot better WPM wise than other studies found ([5], [6]), probably because users could use two pens on a floating keyboard). All other methods were rated so poorly that we would only consider using them if Pointing or Head Gaze is not possible. As a result, the following design rules were developed.

8.1 Design Guidelines

The first thing to point out is that out of the methods tested, Pointing was the single best one regarding all tested measures. There were some where Pointing did not have the best means but differences to the other methods were not significant in those cases. It is important to add that in contrast to other pointing based text entry systems for VR, a two handed approach was used which obviously worked well. Interesting too, is that Pointing was rated very low regarding physical demand even though the user needs to lift his or her arms up to type. This may be due to the fact that the experiment was not very long and the infamous "gorilla arms" could not become a problem. The other tracked controller based method, Pen Based scored worse for all measures. Combined with the fact that means of WPM for Pointing are similar to other QWERTY based pointing approaches ([36], [37]) this brings us to the following design recommendation:

If a tracked controller is available use a pointing based approach.

Another important observation is, that Head Gaze came off better performance wise than the (discrete) Touch Pad method. Head Gaze resulted in slightly (not significantly) higher motion sickness than all other methods but was nevertheless rated better in user experience than the touch pad based methods. Only if physical demand is key Head Gaze is not superior to the touch pad based methods. This is important as many commercial VR systems use a game controller with text input methods similar to the touch based methods. Using smartphones as input hardware was proposed too, in literature [47]. As alternative to that, Head Gaze should be considered:

Head Gaze should be considered before Game controller, smartphone or touch pad based methods if physical demand is not the most important factor.

When it comes to longer text entry physical demand can become an important factor. One should definitely keep in mind that head gaze performs worse for that factor.
The VR design guidance tool for text input techniques by García et al. [6] can therefore be updated: If the first choice "User needs their hands free" is answered with yes, Head Gaze should definitely be considered besides speech recognition, as it has major drawbacks (see chapter 2.3.4). Performances are competitive too (6-13 WPM for speech input [65], [6], 11 WPM for Head Gaze). WPM were also slightly better than other head gaze based non VR text entry studies found ([28], [52]), which means that this method benefits from the VR context. Of course a comparison regarding immersion needs to be done in order to say more
It is interesting to see that the method had no significant effect on neither immersion nor on motion sickness. This is important as these two factors are often considered the

most important ones when it comes to VR and affords the designer the freedom to focus more on performance and other user experience aspects:

The text entry method has no impact on immersion or motion sickness.

This does not mean that the design of the text input method itself has no impact on immersion or motion sickness. An important remark of two participants was that they would have preferred a virtual representation of a desk instead of a floating keyboard, because that broke the immersion for them.

8.2 Design Space

Let's take a look at the design space of chapter 3 again. Some red lines for the question "Which input device?" can turn black.
The factor physical ease is represented through the measure "Physical Demand" of the NASA Task Load Index. It shows that Freehand indeed has a negative effect on physical ease. As stated above touchpads at the side of an HMD should not be used for text entry. Nevertheless the means of physical demand were lower for touch pad based methods than for Head Gaze. However with these methods the user did not have to lift his or her arms up, which he or she would need to do if the touch pad was located on the HMD. Head Gaze (the only method tested realizable without an input device) however, showed higher physical demand than other methods. The positive effect on physical ease that one would have assumed must thus be switched to a negative effect. Participants stated that this method resulted in neck strain which was the reason why the rated it so low on physical ease. Maybe eye gaze could deliver other results. It is important to see that the touch pad based methods achieved the best score regarding physical demand which implies that smartphone based techniques can have positive impact on physical ease.
The factor intuitiveness (walk up performance) is represented through the performance values WPM, KSPC and error rate. As the experiment tested five sentences for each method per participant, the means should give good values for this factor. To check the expert performance more sessions would be needed. Nevertheless use cases for text entry in VR do not (yet) go beyond short phrases. Our experiment could not confirm the assumption that Freehand has a positive effect on walk up performance. WPM were relatively low (although not the lowest), error rates were relatively (though not significantly) high. This is definitely due to the fact that the hardware used could not deliver a satisfying experience. Compared to the mistakes made during the act of typing the mean WPM were pretty high. With increased quality of hand recognition performance values could go up even higher than pen based. Nevertheless at this moment freehand

methods have no positive impact on walk up performance. The negative impact of using no extra input device on walk up performance needs to be dropped too, as Head Gaze had average WPM and KSPC means and together with Pointing the lowest error rate, which actually makes it superior to other methods the also use an input device. An important thing to add here is that, in fact, during our experiment, the user had to select a character by pressing a button on the HTC Vive controller. This way an input device was used. To avoid that, techniques like button time selection or wiping will need to be used, which will lower the means of the performance values. Pointing was the single best method regarding performance measures. So hand held controllers can have a positive impact on walk up performance. The Touch Pad based methods had the lowest WPM values, and pretty poor error rates. Maybe different methods for smartphones could deliver better walk up performance, but for now we will stick with our assumption of a slight negative impact.

As stated above the factors immersion and motion sickness need to be removed from the question, as the method had no impact on immersion or motion sickness regarding text entry.

Subjective quality is represented through user experience. It is clear to see that Freehand has no positive but rather, a negative impact on user experience, even though participants liked the novelty of the method. This implies that an increased quality of hand recognition could result in better user experience values. Besides that we can say that using Hand held controllers (Pointing or Pen Based) can result in good user experience and using touch pad based (smartphone) techniques in poor user experience.

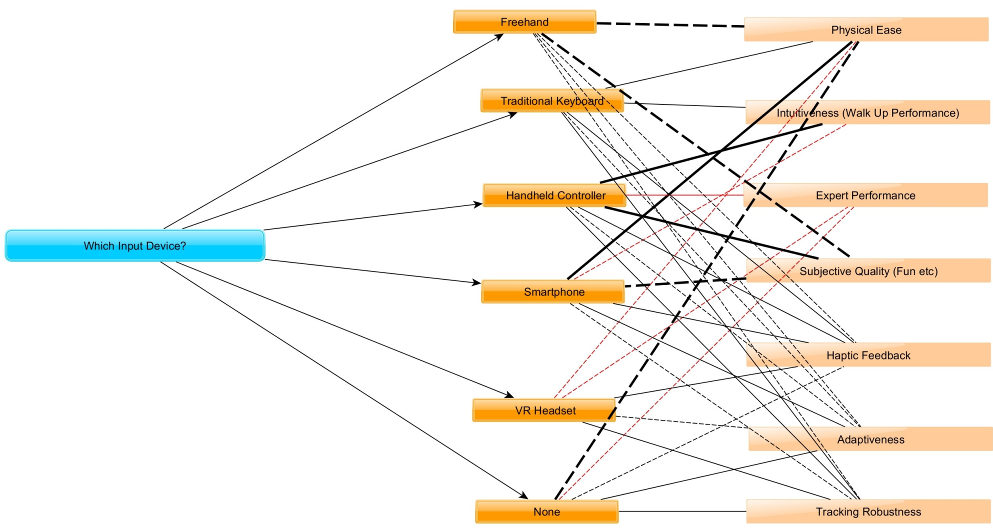


FIGURE 8.1: The updated design space for the question "Which input device?"

8.3 Limitations

A big limitation of the conducted study is that it only compared selective text entry methods. We left out speech recognition and chording keyboards, two potential methods for VR. Speech recognition could have had an impact on immersion, chording keyboards on expert performance, which was not tested at all, as only five sentences were tested with each method and participant. The claims made about Head Gaze as hands free method should be handled with care, as in the experiment selection was done via hand held controllers. As no word completion techniques were used, the methods intentionally had all pretty much the same layout and were not fully designed to be perfect, means may increase more for one method than for another once fully designed for a use case or commercial use resulting in greater or lesser effects.

8.4 Future Work

There are some implications in the design space that still remain unclear. For example the impact of the method on expert performance. Here chording keyboards like the Twiddler approach could deliver great results. Especially when designed and implemented well. The approaches that tried to bring the Twiddler to VR gave no visual feedback on the device itself, as it was not tracked. With situational specific visual feedback the learning curve and resulting frustration for beginners could be lowered and result in a much better overall experience then in the "real" world. This, of course would need to be evaluated too. Other detection based techniques could, of course benefit from the VR context. Thus a study comparing these techniques in VR and the "real" world in a standardized manner would be interesting. It would be interesting to see if the VR context as a new and exciting medium can motivate users to learn new input methods better than traditional devices. The question "Typing in relation to what?", head, hands or body needs to be answered too, as well as if a customizable position, rotation and scale has a positive impact on performance and usability. It would also be interesting to know a bit more about the "best" method, Pointing: Does having once controller instead of two have an impact on performance and usability? This is especially important as the most common pointing based text entry methods for VR only make use of maximum one tracked controller.

Besides text entry, text output is also an important topic.

As stated in the chapters above a classification for UEQ for VR applications could be helpful.

Chapter 9

Conclusion

The three contributions of this thesis are:

- A design space for text entry in VR
- Empirical comparison of QWERTY based selective text entry methods in VR regarding performance, user experience, immersion, motion sickness and mental and physical demand.
- Design guidelines for developers of VR text entry applications

With this work, designers of VR applications that want to add text entry now have a starting point by giving them rules and aides which text input method to chose under which circumstances and what is important to consider during development and implementation. All of this can be done with respect to modern VR technology. For scientists, the design space shows a direction towards what still needs to be scientifically examined. The empirical comparison of QWERTY based selective text entry methods in VR showed some surprising and novel results: Head Gaze had a negative impact on physical ease, especially compared to methods that required the hands to be lifted up like Pointing; The text entry method had no effect on immersion or motion sickness; Game controllers should be avoided if possible. Another surprising fact was that VR experts performed slightly worse than novices. It is now also scientifically proven that the most common VR text entry method, Pointing is the "best", at least out of the selective methods tested.

Text entry is an essential part of human computer interaction. Design annotation (for example 3D artists or architects), filename entry, labeling, precise object manipulation, parameter setting, communication between users and markup [2] are just a few applications for text entry in VR. Future VR systems may be designed to enable the user to stay immersed in VR for longer times, perhaps up to many hours. In that case, even longer text entry would be feasible. Thus, interaction in VR is remains difficult until efficient and well designed text entry methods are established. This thesis brings this goal a little closer to being achieved.

Appendix A

Consent Form

Consent Form for the study "Empirical Analysis of Selection Based Text Entry in Virtual Reality"

I agree to participate in the study conducted by Pascal Ziegler for Saarland University.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

I understand and agree that I am filmed and my gestures including head and arm position and rotation as well as entered text and given data on the questionnaires are recorded, saved and used for scientific purpose.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date:_________

Please print your name: ____

Please sign your name: ____

Thank you!

We appreciate your participation.

Figure A.1: Consent Form

Appendix B

SUS Questionnaire

SLATER-USOH-STEED QUESTIONNAIRE (SUS)

1. Please rate your *sense of being in the* virtual environment, on a scale of 1 to 7, where 7 represents your *normal experience of being in a place.*

2. To what extent were there times during the experience when the virtual environment was the reality for you?

3. When you think back to the experience, do you think of the virtual environment more as *images that you saw* or more as *somewhere that you visited*?

4. During the time of the experience, which was the strongest on the whole, your sense of being in the virtual environment or of being elsewhere?

5. Consider your memory of being in the virtual environment. How similar in terms of the *structure of the memory* is this to the structure of the memory of other *places* you have been today? By 'structure of the memory' consider things like the extent to which you have a visual memory of the virtual environment, whether that memory is in colour, the extent to which the memory seems vivid or realistic, its size, location in your imagination, the extent to which it is panoramic in your imagination, and other such *structural* elements.

6. During the time of your experience, did you often think to yourself that you were actually in the virtual environment?

FIGURE B.1: SUS Questionnaire [76]

Appendix C

Motion Sickness Questionnaire

8.4.2017 Selective Text Entry in Virtual Reality Method Questionnaire 2

Selective Text Entry in Virtual Reality Method Questionnaire 2

To be filled in per method.

* Erforderlich

1. **User ID** *

2. **Method** *
Markieren Sie nur ein Oval.
   - Head Gaze
   - Pen Based
   - Touch Pad
   - Pointing
   - Touch Pad Discrete
   - Freehand

3. **I felt sick to my stomach** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all										Severely

4. **I felt disoriented** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all										Severely

5. **I felt faint-like** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all										Severely

6. **I felt tired/fatigued** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all										Severely

https://docs.google.com/forms/d/1hdeN4E_G04kL9RqaCBwq4hFRDAy0mDPgk1H0liABuGQ/edit 1/3

FIGURE C.1: Motion Sickness Questionnaire [77]

8.4.2017 Selective Text Entry in Virtual Reality Method Questionnaire 2

7. **I felt annoyed/irritated** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	Severely

8. **I felt nauseated** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	Severely

9. **I felt sweaty** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	Severely

10. **I felt hot/warm** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	Severely

11. **I felt queasy** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	Severely

12. **I felt dizzy** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	Severely

13. **I felt lightheaded** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	Severely

https://docs.google.com/forms/d/1hdeN4E_G04kL9RqaCBwq4hFRDAy0mDPgk1H0liABuGQ/edit 2/3

FIGURE C.2: Motion Sickness Questionnaire [77]

8.4.2017 Selective Text Entry in Virtual Reality Method Questionnaire 2

14. **I felt like I was spinning** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Severely

15. **I felt drowsy** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Severely

16. **I felt as if I may vomit** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Severely

17. **I felt clammy/cold sweat** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Severely

18. **I felt uneasy** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	8	9	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Severely

Bereitgestellt von
Google Forms

https://docs.google.com/forms/d/1hdeN4E_G04kL9RqaCBwq4hFRDAy0mDPgk1H0liABuGQ/edit 3/3

FIGURE C.3: Motion Sickness Questionnaire [77]

Appendix D

User Experience Questionnaire

Please make your evaluation now.

For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can express your agreement with the attributes by ticking the circle that most closely reflects your impression.

Example:

attractive	○	⊗	○	○	○	○	○	unattractive

This response would mean that you rate the application as more attractive than unattractive.

Please decide spontaneously. Don’t think too long about your decision to make sure that you convey your original impression.

Sometimes you may not be completely sure about your agreement with a particular attribute or you may find that the attribute does not apply completely to the particular product. Nevertheless, please tick a circle in every line.

It is your personal opinion that counts. Please remember: there is no wrong or right answer!

FIGURE D.1: User Experience Questionnaire [78]

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7
unpredictable	○	○	○	○	○	○	○	predictable	8
fast	○	○	○	○	○	○	○	slow	9
inventive	○	○	○	○	○	○	○	conventional	10
obstructive	○	○	○	○	○	○	○	supportive	11
good	○	○	○	○	○	○	○	bad	12
complicated	○	○	○	○	○	○	○	easy	13
unlikable	○	○	○	○	○	○	○	pleasing	14
usual	○	○	○	○	○	○	○	leading edge	15
unpleasant	○	○	○	○	○	○	○	pleasant	16
secure	○	○	○	○	○	○	○	not secure	17
motivating	○	○	○	○	○	○	○	demotivating	18
meets expectations	○	○	○	○	○	○	○	does not meet expectations	19
inefficient	○	○	○	○	○	○	○	efficient	20
clear	○	○	○	○	○	○	○	confusing	21
impractical	○	○	○	○	○	○	○	practical	22
organized	○	○	○	○	○	○	○	cluttered	23
attractive	○	○	○	○	○	○	○	unattractive	24
friendly	○	○	○	○	○	○	○	unfriendly	25
conservative	○	○	○	○	○	○	○	innovative	26

FIGURE D.2: User Experience Questionnaire [78]

Appendix E

General Questionnaire

8.4.2017 Selective Text Entry in Virtual Reality General Information

Selective Text Entry in Virtual Reality General Information

* Erforderlich

1. **User ID** *

2. **Age** *

3. **Sex** *
   Markieren Sie nur ein Oval.
   - Male
   - Female
   - Sonstiges:

4. **Preferred keyboard layout** *
   e.g. QWERTY (us)

5. **Experience HTC Vive** *
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
Low						High

6. **Experience VR in general** *
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
First Usage						Developer

7. **Experience Head Gaze User Inferfaces** *
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
Low						High

https://docs.google.com/forms/d/1QmI26UfawMTcqXZCnEGRWkZ92-hkwXu0Sf7YSjFq4gU/edit 1/3

Figure E.1: General Questionnaire

8. **Experience Pen Based Text Entry** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	High

9. **Experience Pointing Based User Interfaces** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	High

10. **Experience Game Controller Based User Interfaces** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	High

11. **Experience Freehand User Interfaces/Leap Motion** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	
First Usage	⬭	⬭	⬭	⬭	⬭	Developer

12. **multi-finger input experience** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	High

13. **How often do you deal with digital text entry** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	
Monthly	⬭	⬭	⬭	⬭	⬭	Daily

14. **Visual Impairness** *
Markieren Sie nur ein Oval.

⬭ Yes
⬭ No

15. **Color Impairness** *
Markieren Sie nur ein Oval.

⬭ Yes
⬭ No

FIGURE E.2: General Questionnaire

16. **Susceptibility to sea or motion sickness** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	High

17. **English skills** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	
Beginner	⬭	⬭	⬭	⬭	⬭	Native speaker

Bereitgestellt von
Google Forms

Figure E.3: General Questionnaire

Bibliography

[1] Mychilo Cline. Virtual reality: a catalyst for social and economic change, 2009.

[2] Jacob O. Wobbrock, Duen Horng Chau, and Brad A. Myers. An alternative to push, press, and tap-tap-tap: Gesturing on an isometric joystick for mobile phone text entry. CHI '07 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Pages 667-676, May 2007.

[3] S. T. von Soemmerring and Immanuel Kant. Ueber das organ der seele. Koenigsberg, 1796.

[4] Steven M. LaValle. Virtual reality. Cambridge University Press, 2016.

[5] Doug Bowman, Christopher Rhoton, and Marcio Pinho. Text input techniques for immersive virtual environments: An empirical comparison. Proceedings of the Human Factors and Ergonomics Society, Volume: 46, issue: 26, pages: 2154-2158, September 2002.

[6] Gabriel González, José P. Molina, Arturo S. García, Diego Martínez, and Pascual González. Evaluation of text input techniques in immersive virtual environments. New Trends on Human–Computer Interaction, pp 109-118, February 2009.

[7] Google cardboard. https://vr.google.com/cardboard/, December 2016.

[8] Gear vr. https://www3.oculus.com/en-us/gear-vr/, December 2016.

[9] Oculus rift. https://www3.oculus.com/en-us/rift/, December 2016.

[10] Playstation vr. https://www.playstation.com/de-de/explore/playstation-vr/, December 2016.

[11] Fove. https://www.getfove.com/, December 2016.

[12] Htc vive. https://www.vive.com/de/, November 2016.

[13] Htc vive user picture. http://www.theverge.com/2016/6/7/11877294/htc-vive-vr-headset-should-you-buy-it, December 2016.

[14] Jens Bauer and Achim Ebert. Mobile devices for virtual reality interaction. a survey of techniques and metaphors. International Dagstuhl Seminar, Dagstuhl Castle, Germany, Revised Selected Papers, June 2013.

[15] Google daydream. https://vr.google.com/daydream/, December 2016.

[16] Oculus rift user picture. http://fortune.com/2016/01/04/facebook-delays-oculus-touch/, December 2016.

[17] Htc vive controller picture. http://www.computerbild.de/artikel/cb-Tests-PC-Hardware-HTC-Vive-Review-Test-Steam-Valve-11469625.html, December 2016.

[18] Project alloy. https://newsroom.intel.com/press-kits/project-alloy/, December 2016.

[19] Leap motion. https://www.leapmotion.com/, November 2016.

[20] Leap motion planetarium. https://developer.leapmotion.com/gallery/planetarium, November 2016.

[21] Manus vr. https://manus-vr.com/, April 2017.

[22] Hi5. https://hi5vrglove.com/, April 2017.

[23] Tom Carter, Sue Ann Seah, Benjamin Long, Bruce Drinkwater, and Sriram Subramanian. Ultrahaptics: Multi-point mid-air haptic feedback for touch surfaces. UIST '13 Proceedings of the 26th annual ACM symposium on User interface software and technology, October 2013.

[24] Andreas Pusch and Anatole Lécuyer. Pseudo-haptics: From the theoretical foundations to practical system design guidelines. ICMI '11 Proceedings of the 13th international conference on multimodal interfaces, November 2011.

[25] Koert van Mensvoort, Dik J. Hermes, and Maurice van Montfort. Usability of optically simulated haptic feedback. International Journal of Human-Computer Studies Volume 66, Issue 6, Pages 438–451, June 2008.

[26] Jan Noyes. The qwerty keyboard: a review. International Journal of Man-Machine Studies Volume 18, Issue 3, Pages 265-281, March 1983.

[27] S.K. Card, S.D. Mackinlay, and G.G. Robertson. The design space of input devices. Conference on Human Factors in Computing Systems, CHI 1990, Seattle, WA, USA, April 1-5, 1990,, 1990.

[28] Wei Ding, Ping Chen, Hisham Al-Mubaid, and Marc Pomplun. A gaze-controlled interface to virtual reality applications for motor- and speech-impaired users. University of Massachusetts Boston, University of Houston-Downtown, University of Houston-Clear Lake, 2009.

[29] Linden Lab. Second life. www.secondlife.com, 2016.

[30] Blizzard Entertainment. World of warcraft. https://worldofwarcraft.com/, 2016.

[31] Ivan Poupyrev, Numada Tomokazu, and Suzanne Weghorst. Virtual notepad: Handwriting in immersive vr. Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998, March 1998.

[32] Scott Frees, Rami Khouri, and G. Drew Kessler. Connecting the dots: Simple text input in immersive environments. Virtual Reality Conference, 2006.

[33] Toshiyuki Masui. An efficient text input method for pen-based computers. CHI '98 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, April 1998.

[34] Steam vr keyboard. http://www.valvetime.net/threads/using-unity-at-valve-vision-summit-2016.253857/, November 2016.

[35] Steam vr. http://store.steampowered.com/steamvr, December 2016.

[36] Garth Shoemaker, Leah Findlater, Jessica Q. Dawson, and Kellogg S. Booth. Mid-air text input techniques for very large wall displays. Proceedings of Graphics Interface 2009, Pages 231-238, May 2009.

[37] Anders Markussen, Mikkel Roenne Jakobsen, and Kasper Hornbaek. Selection-based mid-air text entry on large displays. IFIP Conference on Human-Computer Interaction INTERACT 2013: Human-Computer Interaction, pp 401-418, 2013.

[38] Anna Maria Feit, Daryl Weir, and Antti Oulasvirta. How we type: Movement strategies and performance in everyday typing. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16). ACM, New York, NY, USA, 4262-4273, 2016.

[39] Xin Yi, Chun Yu, Mingrui Zhang, Sida Gao, Ke Sun, and Yuanchun Shi. Atk: Enabling ten-finger freehand typing in air based on3d hand tracking data. UIST '15, Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology, Pages 539-548, November 2015.

[40] Anthony James Francis Natoli. Virtual reality keyboard system and method. Patent US6388657 B1, December 1998.

[41] Carsten Mehring, Falko Kuester, Kunal Deep Singh, and Michelle Chen. Kitty: Keyboard independent touch typing in vr. Virtual Reality. Proceedings. IEEE, 2004.

[42] Manuel Prätorius, Dimitar Valkov, Ulrich Burgbacher, and Klaus Hinrichs. Digitap: an eyes-free vr/ar symbolic input device. Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology,Pages 9-18, 2014.

[43] Senseboard. http://www.senseboard.com/, January 2017.

[44] Yi-Ta Hsieh, Antti Jylhae, Valeria Orso, Luciano Gamberini, and Giulio Jacucci. Designing a willing-to-use-in-public hand gestural interaction technique for smart glasses. CHI '16 Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, May 2016.

[45] Xbox controller. http://www.xbox.com/de-DE/xbox-one/accessories?controllers, November 2016.

[46] Playstation controller. https://www.playstation.com/de-de/explore/ps4/accessories/, November 2016.

[47] Jaewoo Ahn and Kyungha Min. Virtualphonepad: A text input tool for virtual environments. Advances in Artificial Reality and Tele-Existence. Lecture Notes in Computer Science, vol 4282. Springer, Berlin, Heidelberg, 2006.

[48] Jaewoo Ahn and Kyungha Min. Virtualqwerty: Textual communication in virtual reality. Advances in Visual Computing. ISVC. Lecture Notes in Computer Science, vol 4292. Springer, Berlin, Heidelberg, 2006.

[49] Carsten Magerkurth and Richard Stenzel. A pervasive keyboard - separating input from display. Pervasive Computing and Communications. Proceedings of the First IEEE International Conference, March 2003.

[50] Ken Perlin. Quikwriting: Continuous stylus-based text entry. UIST '98 Proceedings of the 11th annual ACM symposium on User interface software and technology, November 1998.

[51] I. Scott MacKenzie and Kumiko Tanaka-Ishii. Text entry systems. Morgan Kaufmann Publishers Inc. San Francisco, 2007.

[52] Nikolaus BeeElisabeth André. Writing with your eye: A dwell time free writing system adapted to the nature of human eye gaze. Perception in Multimodal Dialogue Systems. Lecture Notes in Computer Science, vol 5078. Springer, Berlin, Heidelberg, 2008.

[53] Robert Rosenberg and Mel Slater. The chording glove: A glove-based text input device. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 29, NO. 2, MAY 1999, May 1991.

[54] Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and E. W. Looney. Twiddler typing: One-handed chording text entry for mobile phones. College of Computing and GVU Center, Georgia Institute of Technology, Atlanta, USA, April 2004.

[55] Twiddler 3 picture. http://www.theregister.co.uk, November 2016.

[56] Google glass. https://www.google.com/glass/start/, March 2017.

[57] Chun Yu, Ke Sun, Mingyuan Zhong, Xincheng Li, Peijun Zhao, and Yuanchun Shi. One-dimensional handwriting: Inputting letters and words on smart glasses. CHI '16 Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, May 2016.

[58] David Goldberg and Cate Richardson. Touch-typing with a stylus. CHI '93 Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, May 1993.

[59] Steven J. Castellucci and I. Scott MacKenzie. Graffiti vs. unistrokes: An empirical comparison. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08), 305-308., 2008.

[60] Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel. Edgewrite: A stylus-based text entry method designed for high accuracy and stability of motion. UIST '03 Proceedings of the 16th annual ACM symposium on User interface software and technology, November 2003.

[61] Anders Markussen, Mikkel Roenne Jakobsen, and Kasper Hornbaek. Vulture: A mid-air word-gesture keyboard. CHI '14, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Pages 1073-1082, April 2014.

[62] Tao Ni, Doug A. Bowman, and Chris North. Airstroke: Bringing unistroke text entry to freehand gesture interfaces. CHI '11 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, May 2011.

[63] Srinath Sridhar, Anna Maria Feit, Christian Theobalt, and Antti Oulasvirta. Investigating the dexterity of multi-finger input for mid-air text entry. SIGCHI Conference on Human Factors in Computing Systems (CHI) 2015, Seoul, South Korea, 2015.

[64] Ben Shneiderman. the limits of speech recognition. Communications of the ACM CACM Volume 43 Issue 9, Pages 63-65, September 2000.

[65] Lode Hoste, Bruno Dumas, and Beat Signer. Speeg: A multimodal speech- and gesture-based text input solution. AVI '12 Proceedings of the International Working Conference on Advanced Visual Interfaces, May 2012.

[66] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. Sphinx-4: A flexible open source framework for speech recognition. Sun Microsystems, Inc. Mountain View, CA, USA, 2004.

[67] Allan MacLean, Richard M. Young, Victoria M.E. Bellotti, and Thomas P. Moran. Questions, options, and criteria: Elements of design space analysis. Human–Computer Interaction Journal Volume 6,Issue 3-4: Design Rationale, 1991.

[68] Fred J. Damerau. A technique for computer detection and correction of spelling errors. Communications of the ACM, Volume 7 Issue 3, Pages 171-176, March 1964.

[69] Martijn J. Schuemie, Peter van der Straaten, Merel Krijn, and Charles A.P.G. van der Mast. Research on presence in virtual reality: A survey. CyberPsychology and Behavior 4(2), pages 183-201, July 2004.

[70] Matthew Lombard and Theresa Ditton. At the heart of it all: The concept of presence. Journal of Computer-Mediated Communication, 3(2), September 1997.

[71] Pavel Zahorik and Rick L. Jenison. Presence as being-in-the-world. Journal Presence: Teleoperators and Virtual Environments archive Volume 7 Issue 1, Pages 78-89, February 1998.

[72] M. Bachynskyi, G. Palmas, A. Oulasvirta, and T. Weinkauf. Informing the design of novel input methods with muscle coactivation clustering. Transactions on Computer-Human Interactions ACM, 2015.

[73] Philip Quinn and Shumin Zhai. A cost-benefit study of text entry suggestion interaction. Proceedings of the CHI Conference on Human Factors in Computing Systems, Pages 83-88, May 2016.

[74] Unity3d. https://unity3d.com, April 2017.

[75] Keith Vertanen and Per Ola Kristensson. A versatile dataset for text entry evaluations based on genuine mobile emails, August 2011.

[76] Mel Slater, Martin Usoh, and Anthony Steed. Depth of presence in virtual environment. Presence, Vol. 3, No. 2, Pages: 130-144, 1994.

[77] Peter Gianaros, Eric Muth, J. Toby Mordkoff, Max Levine, and Robert Stern. A questionnaire for the assessment of the multiple dimensions of motion sickness. Aviat Space Environ Med, 72(2): 115–119, February 2001.

[78] Bettina Laugwitz, Theo Held, and Martin Schrepp. Construction and evaluation of a user experience questionnaire. Symposium of the Austrian HCI and Usability Engineering Group, USAB 2008: HCI and Usability for Education and Work pp 63-76, 2008.

[79] Sandra Hart and Lowell Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. Advances in Psychology, Volume 52, 1988, Pages 139-183, 1988.