Analysis and Comparison of Text-Input Methods in Virtual Reality

Bachelor Thesis

Natural Science Faculty of the University of Basel
Department of Mathematics and Computer Science
Databases and Information Systems
https://dbis.dmi.unibas.ch/

Examiner: Prof. Dr. Heiko Schuldt
Supervisor: Florian Spiess, MSc.

Nakarin Srijumrat
nakarin.srijumrat@unibas.ch
2018-054-882

June 27, 2022

Abstract

Text input is one of the most common forms of Human-Computer Interaction and is essential for many tasks including interface operation and communication. While the conventional keyboard has established itself as the standard for text input for physical and touch-screen based devices, the search for a fast and reliable text input system for Virtual Reality based applications remains an unsolved problem.

This thesis aims to contribute to existing research by the means of implementing and evaluating three input methods as open-source software. While two designs are inspired by the conventional keyboard, the introduction of a new input method, the Analog Stick Keyboard, is an attempt at innovation in a young research field. The evaluation in the form of a study is conducted using current research standards overarching not only VR-based, but the entire field of text entry research. The results showed big differences in terms of speed, error rate and usability between all input methods and can serve as a basis for further, long-term research.

Acknowledgments

First and foremost, I want to greatly express my gratitude towards Florian Spiess, whose support and invaluable insight shaped the work in the thesis second to none. His expertise and communication was a driving factor for the realization of the results.

My thanks go to Prof. Dr. Heiko Schuldt for not only being a great educator that encourages and offers irreplaceable support for all his students, but also for giving me the opportunity to write my bachelor thesis in the Databases and Information Systems group.

I also thank all members of the DBIS group who run and maintain the VR lab at the facilities, without whom all the research going on wouldn't be possible.

I thank all my work colleagues at the IT, in particular Sebastian and Philippe for their support, mentorship and enabling the most fun and rewarding workplace a student can dream of. The three years (and counting!) have been nothing short of amazing, and giving me the chance to apply all the things I've learned during my time as a student assistant is something I value very deeply.

Finally, I want to thank my friends and family, especially Elena, for their unrivaled support and love not only during the thesis, but throughout the entirety of my studies.

Table of Contents

1 Introduction

The rise of Virtual Reality (VR) in recent years is undeniable and not without reason - it enables experiences for the user which would not be possible with conventional computer setups. The computer-generated simulation of a space, which is made to be experienced using devices such as headsets with integrated screens and controllers, offers unparalleled immersion thanks to innovative technologies. The interest in VR came along with the golden age of computer science research in the 70s and 80s, but has dwindled during the period of the 2000s. Since the introduction of the Oculus Rift in 2010 (and the deployment of its development kit in 2013), VR technology has found itself in a renaissance, leading to a surge in technological innovation and increased interest in research [18].

Uses of Virtual Reality include vocational and military training, simulations, entertainment centers, healthcare, software front-end interfaces and leisure activities [4, 14, 18, 19].

The number of owners of consumer-ready virtual reality systems such as the HTC Vive, Oculus Quest 2 and the more budget-friendly Samsung Gear VR are rapidly increasing and growth does not seem to slow down. The steady rise of VR device ownership reflects in the monthly published Steam users report, where a peak of around 2.94 million monthly connected devices has been reached in February 2022 with an exponential trend $R^2 = 0.928$ [10]. Currently, the VR device market is dominated by Meta (67.3% market share, January 2022), the manufacturer of the highly popular Oculus Quest 2[1].

[1] Steam Hardware Survey: https://store.steampowered.com/hwsurvey

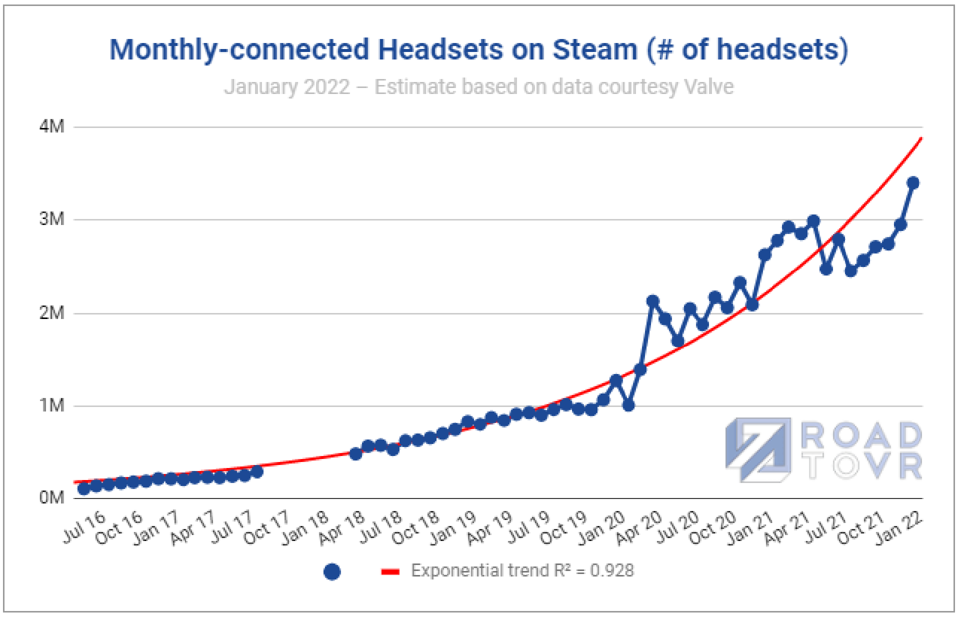

Figure 1.1: Number of headsets monthly connected to Steam [10]

1.1 Motivation

While the hardware aspects of Virtual Reality are quickly improving and leaps in technological advances have been made, the software landscape of virtual reality still has room for substantial improvement. This includes the area of text-entry, one of the most common forms of human-computer interaction essential for many tasks involving an interface. Currently, keyboards for VR modelled after conventional desktop counterparts exist, but suffer from lack of effectiveness. This had led to a discrepancy between reliability, speed and accuracy of text input between the physical and virtual space. Furthermore, many of the current proposed methods are not publicly available or have been developed for proprietary hardware [5].

This thesis aims to contribute to existing research by implementing and evaluating two established and one original input method in Virtual Reality. By comparing various methods, we can gain insight on key properties of input methods important for use in Virtual Reality. In addition, by making the keyboards publicly available as open-source software, the global community can collaborate and contribute to the results of the thesis. Taking into consideration current research standard metrics and evaluation methodologies of text entry methods, we can make assumptions comparable to past research.

2
Background

This section presents relevant technologies and previous work and helps to set the context for the thesis at hand.

2.1 Established VR text input methods

With VR devices reaching a consumer-ready state, various input methods have established themselves as a standard for the time being. Despite the methods not being standardized, developers use the same paradigm of a conventional physical computer keyboard.

2.1.1 Raycast Keyboard

Currently, the most popular way of text input in VR applications is the “aim and shoot” style keyboard, where a projected keyboard is visible and a ray from the controller is cast. The confirmation of a keystroke is made using a button or trigger of the controller [11].

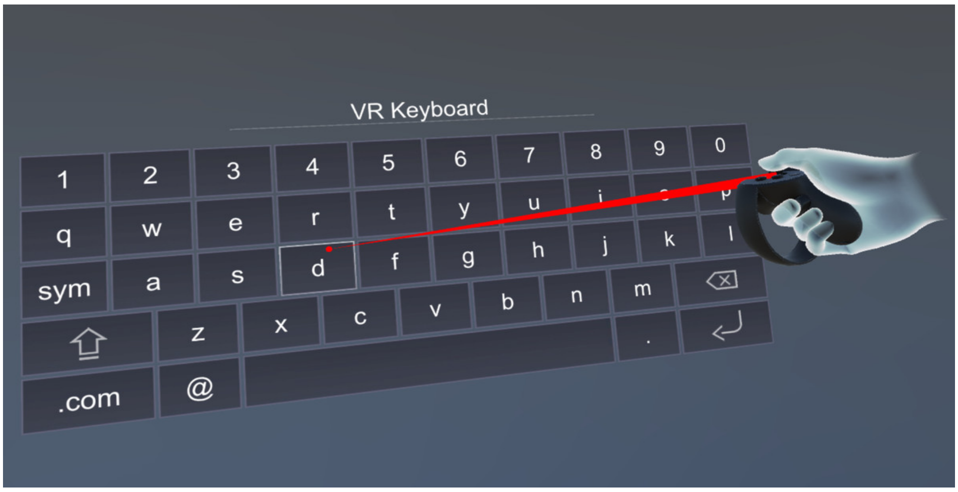

Figure 2.1: Raycast keyboard [11]

2.1.2 Drumstick keyboard

The drum-like VR keyboard (further referred to as Drumstick Keyboard) is a VR text input interface presented as a prototype from Google Daydream Labs in 2016 [7]. By using the left- and right-hand controllers as metaphors for drumsticks, the downward movement of the sticks "press" the keys of a virtual keyboard. Ever since its introduction, the efficiency and adaptability has been a widely discussed topic in the research field of Human-Computer Interaction in the context of VR. Newer research has indicated high usability and an altogether positive user experience of the keyboard, but warned of a possible learning curve and arm fatigue after extended usage [9].

Figure 2.2: Drumstick keyboard as initially presented in Daydream Labs

2.2 Steam Daisywheel keyboard

The Daisywheel keyboard was introduced together with Steam Big Picture in 2012, a controller-optimized user interface for the Steam client, a games and application manager for desktop and laptop computers. This input method splits the analog stick angles into eight equal areas and maps the four buttons on the right-hand side to a letter depending on the analog stick position. Selection is done using the analog stick and input confirmation is performed using one of the four corresponding buttons.

The Daisywheel keyboard has served as an inspiration for the Analog Stick keyboard to be introduced later in the thesis. It sparked interest for the idea by being an input method where its operation can be done without looking at the interface once it's memorized.

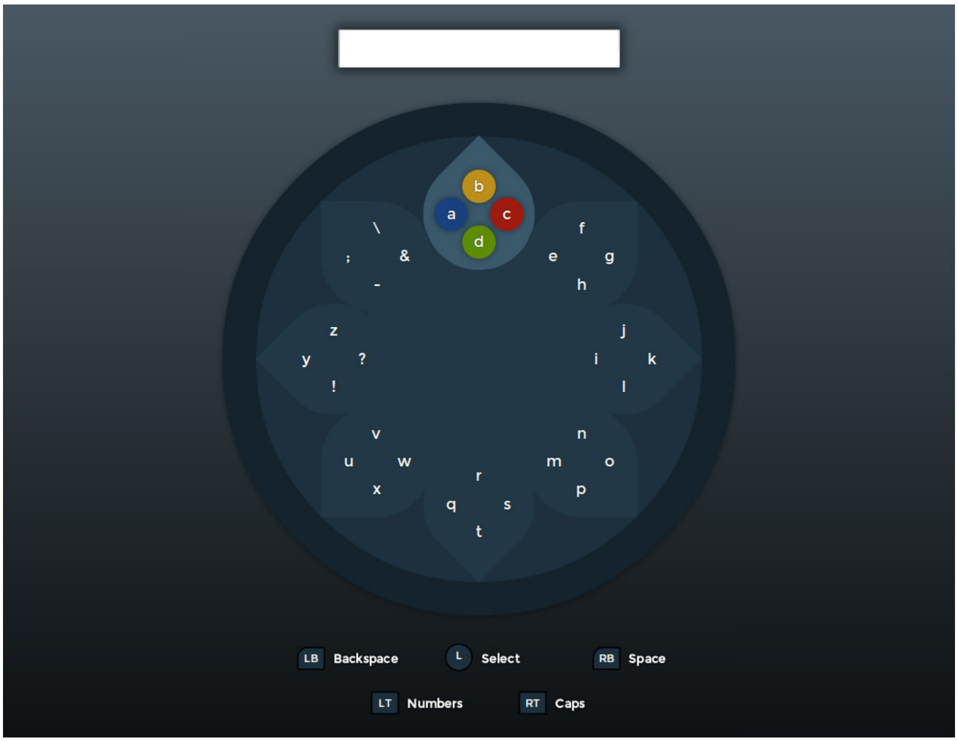

Figure 2.3: Daisywheel Keyboard by Valve[2]

2.3 vitrivr-VR

vitrivr[3] is an open-source multimedia retrieval system developed and maintained by the DBIS group of the department of mathematics and computer science at University of Basel. Focusing primarily on video, the system also supports querying for audio, images, 3D files and other structured data using a cineast retrieval engine and Cottontail DB, both also developed and maintained by the DBIS group [8, 16, 17].

The VR interface for vitrivr (vitrivr-VR[4]) has been developed using the Unity[5] engine, a 2D and 3D environment for the creation of multimedia experiences including, but not limited to, video games. Communication between VR hardware and software is handled by OpenXR. This is made possible thanks to the package management system employed by the Unity engine. The input methods implemented for the scope of the thesis were tested in the vitrivr ecosystem.

[2] Reverse engineered by GitHub user likethemammal: https://likethemammal.github.io/daisywheeljs/
[3] https://vitrivr.org/
[4] https://github.com/vitrivr/vitrivr-vr
[5] https://unity.com/

3
Implemented and Examined Input Methods

This chapter gives an overview of the three input methods that were developed and compared to each other. The aim is to explain the conceptual challenges and ideas that went into the realization of the products.

3.1 Virtual Keyboard

The Virtual Keyboard is a replication of a common keyboard found on the vast majority of computer setups. It features keys adhering to the US ANSI (a.k.a. QWERTY) layout and supports input of numbers and letters as well as symbols using the Shift-Key. Furthermore, navigation is realized using the arrow keys. Special inputs include Tabulator, Return and Backspace. The Caps Lock feature is implemented and L/R arrow can be used to navigate within a text field. With modularity in mind, the inputs are purely based on the key label and thus, keyboard layouts other than the US layout can be implemented by simply replacing the labels with the desired letters and symbols.

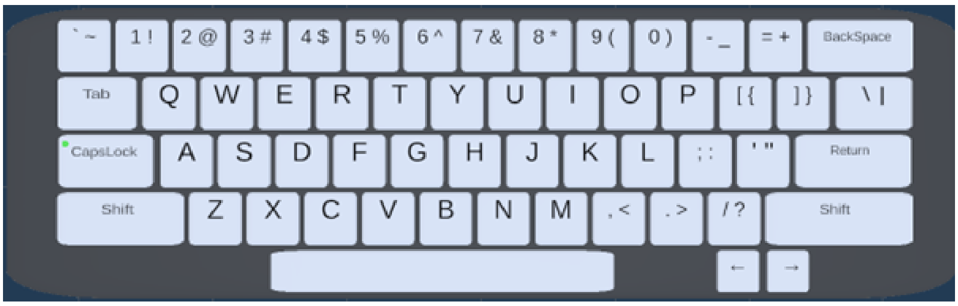

Figure 3.1: Virtual Keyboard with US Layout

3.1.1 Feedback

Audiovisual feedback is the key to a pleasant user experience of a system. In the case of the virtual keyboard, visual feedback is provided by reclining pressed keys into the keyboard body and painting the pressed key in a darker shade. Audio feedback is provided in the form of a click sound played whenever a keystroke is detected.

3.1.2 Typographical Error Prevention

A typographical error, or short ‘typo’ is referred to an unintentional input caused by accidental keystrokes or even misbehavior of the underlying system. The implementation takes several measures to undermine the frequency of typos caused by accidental keystrokes.

3.1.2.1 Disabling Key Rollover

Rollover is the ability of a keyboard to process simultaneous keystrokes. This functionality is desirable for operating a computer as it allows for commands which don’t have text input as a result. However, the ability to press more than two keys simultaneously is disabled if none of the keys is the Shift key. A keystroke “locks” all other keys until it is released in order to prevent inputs of the neighboring keys in case of lacking pointer precision.

3.1.2.2 Expanding the Key Hitbox

Before a key is pressed, the trigger hitbox matches the 3D mesh of the corresponding key. After triggering a keystroke, the hitbox expands and the key is not reset until the pointer used to trigger the key leaves the new hitbox. This expansion of the hitbox further prevents unwanted additional keystrokes of the same key. If the hitbox were not to change, the key could also be pressed repeatedly by accident, as the key moves into the keyboard body upon triggering and retracts, triggering another keystroke if the pointer did not move in the meanwhile.

3.2 Drumstick Keyboard

The Drumstick keyboard is similar to the Virtual Keyboard, but differs in the way the keystrokes can be triggered. Round drums are hit using sticks with a sphere attached to one end and each hit represents hitting the key on a common keyboard. It was first shown at Google’s Daydream Labs and has since held a relevancy in Virtual Reality text input research.

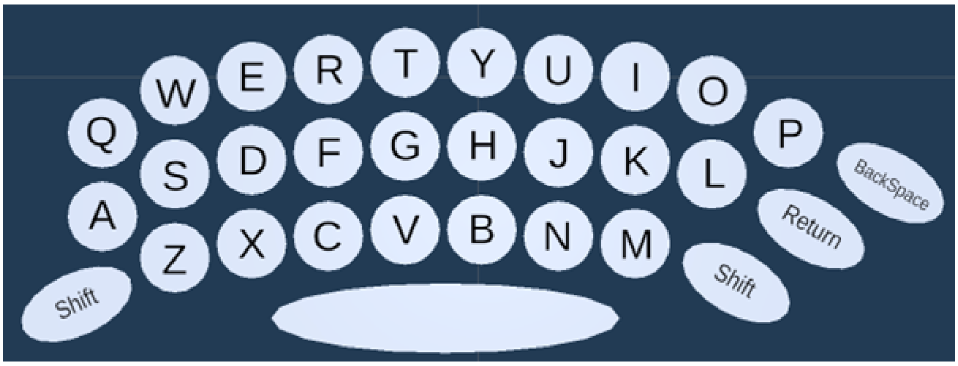

Figure 3.2: Drumstick Keyboard keybed

3.2.1 Feedback

A keystroke again provides visual and audio feedback for the user. The audio feedback is identical to the virtual keyboard, whereas the visual feedback is provided in a distinct manner by increasing the radius of the 3D model of the drum in addition to shading the model darker. This creates a similar sensation of stimulation to hitting a real physical drum membrane.

3.2.2 Positioning and arrangement of drums

The keys are arranged in concentric circles to mimic a drum kit, as it can be more intuitive. Using sticks to hit keys translates the origin of the movement from the arm to the wrists, which makes round movements more natural to perform rather than straight movements.

3.2.3 Rectangle hitbox vs. Cylinder model (implementation-specific)

The hitbox for a drum representing a key is a rectangle and thus, does not match the 3D model for the following two reasons: On one hand, the Unity engine does not support cylindrical-shaped colliders due to the complexity of calculating collisions with a cylinder. On the other hand, using box colliders does not influence usability and allows us to re-use part of the code used for the virtual keyboard, reducing redundancy.

3.3 Analog Stick Keyboard

The analog stick keyboard features two planes with four disks each attached to the left and right hand. By moving the analog stick of a controller into a quadrant, the corresponding keystroke will be triggered. By default, the disk in the middle is considered active and the user can input one of the four letters shown. By holding one or both triggers, the user is able to input keys labeled on the left, right and above the default disk. Compared to the virtual keyboard and the drumstick keyboard, the analog stick keyboard is an original input method and thus, particularly interesting for the research presented in this thesis. It also considered the only one of the keyboards that can be used blindly (i.e. without looking at the keys themselves while using).

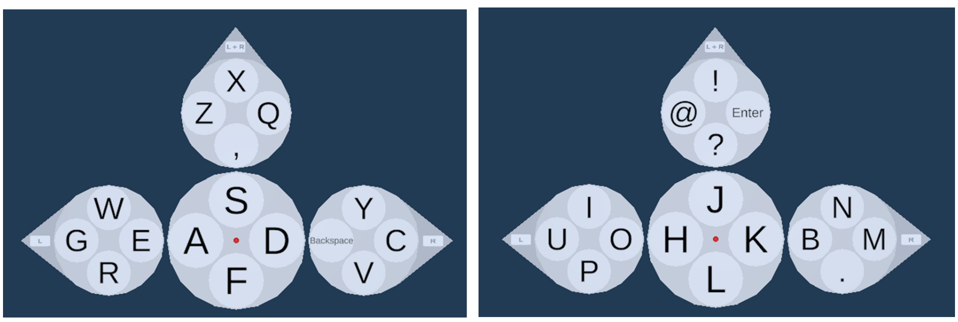

Figure 3.3: Analog stick keyboard with custom layout

Feedback Audiovisual feedback is provided similarly to the preceding input methods, namely a click sound upon triggering a keystroke as well as a visual cue that the input has been triggered. To enhance the user experience, a cursor is showing the current position of the analog stick on the active disk to inform the user about the current stick position.

3.3.1 Layout

The layout of the key labels is derived from the standard ANSI-US layout with minor changes to accommodate access to backspace and the space bar. This system supports 32 letters that can be chosen as possible input letters. After assigning every letter in the alphabet to a button, we are left with six spaces for system inputs and symbols. The system does not support capitalization as-is. The positioning of the disks is made to see the whole layout at all times in order to be able to plan the next movements in advance.

3.3.2 Keystroke detection

Keystrokes are triggered when the magnitude of the analog stick vector exceeds a certain threshold. Depending on the quadrant the analog stick is moved into, one of the four projected symbols will be input. Before a new keystroke can be triggered after a registered input, the analog stick must return to its origin. This prevents accidental inputs if the stick is extended near a quadrant's edge.

3.4 Form

The input methods are developed and deployed as a Unity package that can be imported into any project. Asset files and scripts are stored on a GitHub repository[6].

This method ensures modularity and allows easy modification or extension of the system in the future.

[6] https://github.com/N4karin/vr-open-text-input

4 Evaluation

In text-entry research, several methods are used to determine the performance and usability of a given input method. In this chapter, we examine the research plan for the evaluation of the input methods.

4.1 Key Metrics

We differentiate between performance and usability metrics. The former includes the text entry rate and the error rate, while the usability is determined via a questionnaire.

4.1.1 Performance metrics

The Words per Minute (WPM) metric is the most common measure of text entry performance. It does not consider the correctness, but rather the number of symbols in the input stream and the time taken to write the text. It is calculated as follows:

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

T denotes the length of the phrase and S the amount of time in seconds for the user to input the requested phrase from the first keystroke to the last. The factor of $\frac{1}{5}$ originates from the average word length in English. This equation also considers spaces in a text. The “-1” in the numerator of the first term originates from the fact that the time to the first keystroke is not considered [2].

Compared to measuring the text entry rate, measuring the error rate faces more difficulties as there are various metrics that are considered valid despite minor flaws. We have to decide whether the system allows for mistakes in the copied text and if yes, how to consider the mistakes. For this thesis, we use the Erroneous Keystroke Error Rate (EKR ER) which forces the testers to correct all mistakes before proceeding in the experiment. It measures the ratio of the total number of keystrokes compared to the final text and is defined as follows:

$$EKS\ ER = \frac{IF}{T} \times 100 \tag{4.2}$$

Like before, T denotes the length of the phrase. IF stands for Incorrect Fixed and contains the length of all keystrokes in the input stream, which do **not** appear in the final text result [2].

It is important to remember that while EKS ER shows a good picture of the participants' accuracy, by forcing all participants to fix all typing mistakes, the WPM metric is affected negatively.

4.1.2 Usability metrics

The usability of a given input method will be determined using the System Usability Scale [6]. The participants will be asked a set of ten questions about the user-friendliness for each of the input methods. As a consequence, a score from 0 to 100 can be derived from the answers and put into perspective by comparing the resulting scores. Something to note about the SUS is that it considers the system as-is and does not take into account the learning curve or future potential usability of a system.

4.2 Participants

14 Participants with a technical expertise range from novice to highly advanced, aged 21 to 55 (average 28), took part in the experiment. While not everyone is familiar with VR technology, all participants were given time to get used to the virtual space if required. All the participants were proficient in the English language. 3 of the 14 participants were female, the remainder consists of males. As the participation was on a purely voluntarily basis, no compensation was granted for taking part in the research.

4.3 Used Devices and Location

All experiment sessions were conducted on the Vive Cosmos using the Vive Cosmos controllers using Hardware capable of driving VR devices. The tracking method was inside-out, meaning that no lighthouse tracking was used for the experiment. Furthermore, all sessions were held at the local research facilities of the computer science department at the University of Basel.

4.4 Procedure

Participants will be asked to copy a random phrase taken from the MacKenzie phrase set N ($N = 5$ for virtual and drumstick keyboard, $N = 3$ for analog stick keyboard) times. For every successful input of a copied phrase, the system calculates the key metrics and displays a new phrase. After the successful input of the n-th phrase, the experiment terminates and the average of the metrics for every phrase are displayed for the participant and research conductor to see.

The MacKenzie phrase was chosen due to it being an established standard in the text-input field, as it represents the average English phrases found in the language. It also makes the research comparable to other publications in the same research field [12].

The sequence of the experiment is as follows:

1. The participant will be told that they will use the controller to operate different input methods to copy text displayed above a text field. After that, there will be a questionnaire to fill out.

2. The VR headset will be setup for the participant to ensure comfortable wear and adjust pupil distance settings for best experience.

3. If necessary, the participant will be given opportunity to accommodate to the VR space if they are unfamiliar with the technology.

4. The participant will use the Virtual Keyboard to complete the experiment. The participant is free to practice and test the implementation to get familiar with the input method. The research conductor takes note of the WPM and EKR ER result.

5. The participant will use the Drum Stick to complete the experiment. The participant is free to practice and test the implementation to get familiar with the input method. The research conductor takes note of the WPM and EKR ER results.

6. The participant will use the Analog Stick Keyboard to complete the experiment. The participant is free to practice and test the implementation to get familiar with the input method. The research conductor takes note of the WPM and EKR ER results.

7. The participant is given a questionnaire with the SUS questions for each keyboard and fills them out.

8. The experiment session is concluded.

5 Results

The following sections present the analysis and discussion of the data collected from the experiment and questionnaire.

5.1 Performance

Measuring the average words per minute for the various input methods as described in 4.1.1, we can compare the input methods in terms of text entry speed. Furthermore, we can group the results by expertise to draw further conclusions. In our case, we distinguished between three levels of VR experience and general technical proficiency (Expert n = 5, Intermediate n = 5, Novice n = 4).

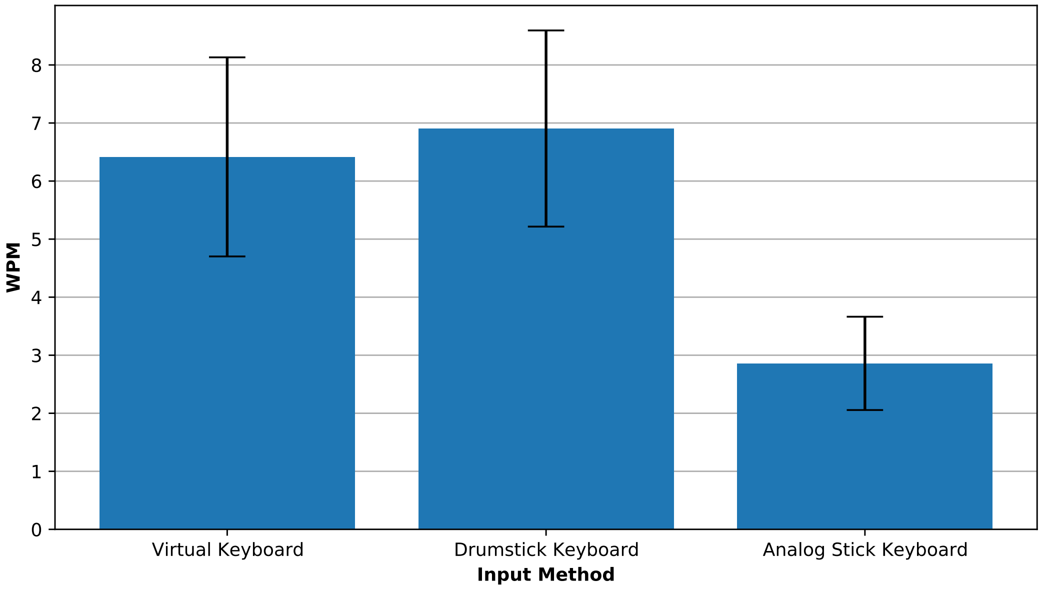

Figure 5.1: Words per minute per input method

5.1 shows that participants performed the best when using the Drumstick keyboard with an average of 6.91 WPM (SD = 1.69), while the Analog Stick keyboard was the slowest to use with an average of 2.80 WPM (SD = 0.82). The virtual keyboard performs slightly worse than the Drumstick Keyboard with an average WPM value of 6.42 (SD = 1.71).

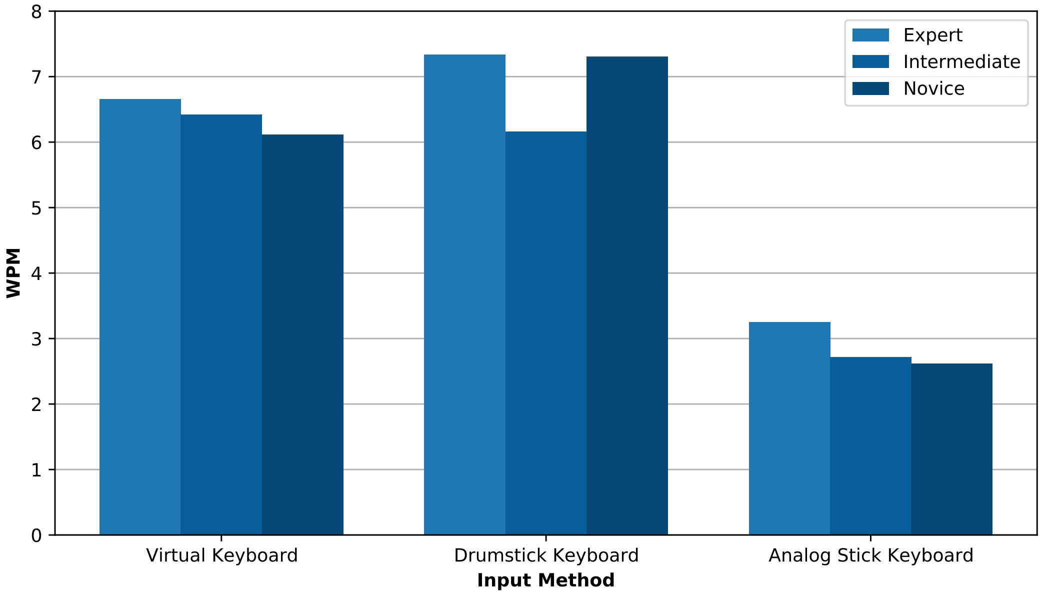

Figure 5.2: Words per minute per input method, grouped by expertise

Looking at the grouped data from 5.2, experts perform better than their intermediate and novice peers with the virtual keyboard and Analog Stick keyboard, which is expected. However, expert and novice users greatly outperform intermediate users with the Drumstick Keyboard. One of the possible reasons could include the intuition of experts and increased focus for novice users, whereas intermediate users could be overestimating their ability and perform worse by focusing to a lesser extent. Because the sample size is rather low, we cannot draw apparent conclusions.

Nevertheless, the speed of all presented input methods is hugely below the expected text entry speed when using a physical computer (40 WPM for the general population, up to 120 WPM for professional typists, not considering unintentional mistakes) [3]. This could mean that the gap when it comes to text entry speed between the physical world and virtual reality could remain significant using the methods implemented in this thesis.

Unfortunately, the scope of the thesis doesn't include the long-term usage and learning curve of the input methods, which could be interesting for the Analog Stick keyboard in particular.

Looking at the error rate presented by 5.3, we can make further assumptions about the performance of the text entry methods.

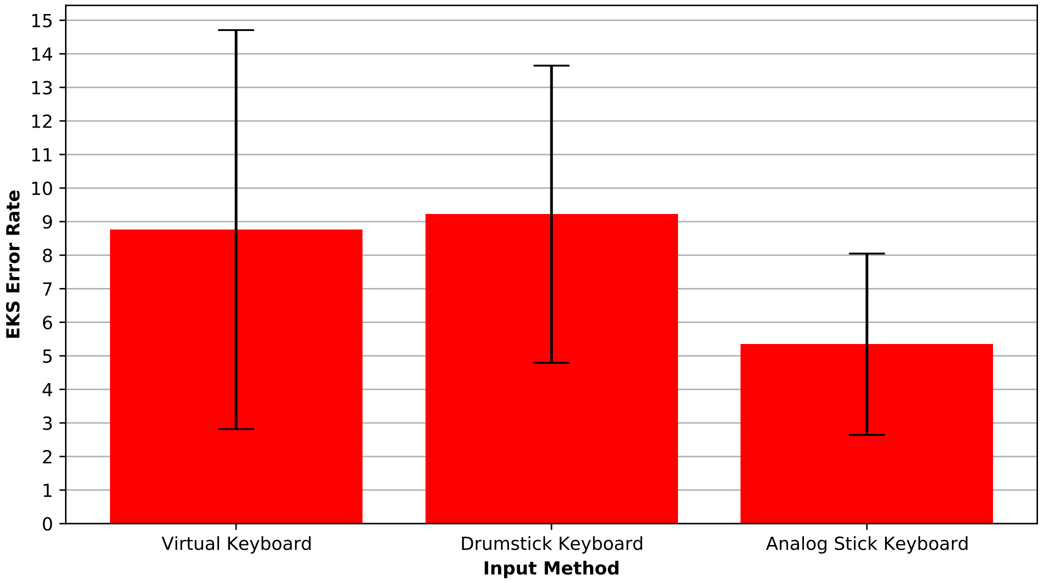

Figure 5.3: EKS Error Rate per input method

The Drumstick keyboard interestingly has the highest error rate with 9.22% (SD = 4.42%) while being the input method with the highest average WPM among the compared input systems. The smaller standard deviation compared to the Virtual Keyboard (Average ER = 8.76%, SD = 5.95%) suggests a more consistent input experience. On average, the Analog Stick keyboard is the input method where the participants were the least prone to errors with an error rate of 5.35%, but a relatively high SD of 2.7%.

The discrepancy of the error rate of the Analog Stick keyboard compared to the two alternatives could be contributed to by the cognitive challenge this particular method provides. As every input has to be precise and requires a lot of planning (given the layout is not memorized yet), each keystroke tends to bear a strong intent. Interesting to note is that there was one participant who was able to type all requested sentences using the Analog Stick keyboard without ever inputting an erroneous keystroke (2.72 WPM, almost matching the average of 2.80 WPM).

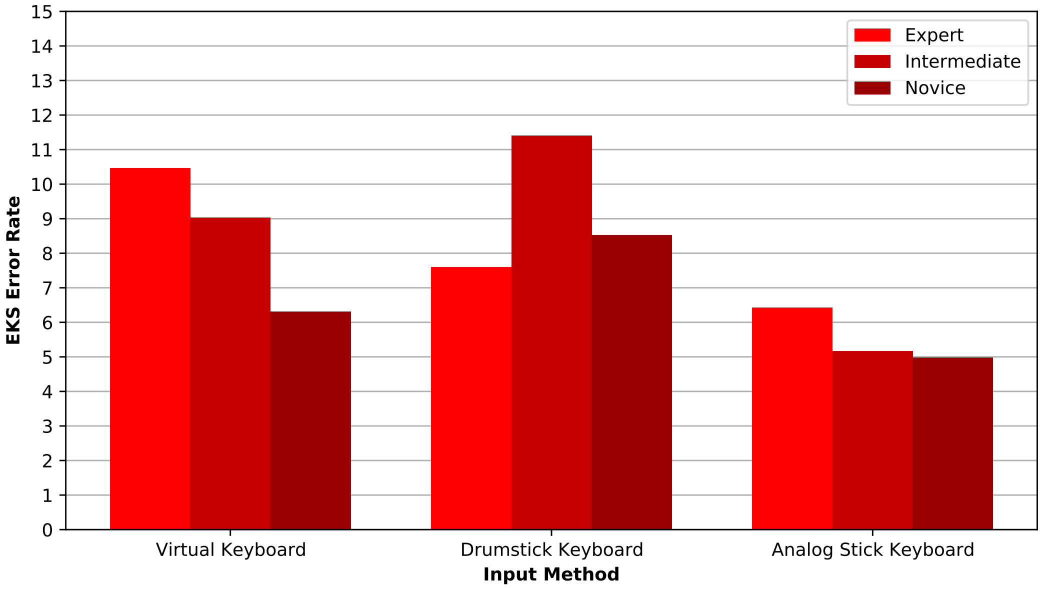

Figure 5.4: EKS Error Rate per input method, grouped by expertise

Grouping the error rate performance by expertise in 5.4 reveals another interesting pattern for the Virtual and Analog Stick keyboard, where experts tend to make more errors than their intermediate and novice counterparts. Again, this can be contributed to a false sense of security for the virtual keyboard and the technicality behind VR technology. The error rate is inversely correlated to the text entry speed, which was expected because correcting mistakes slows down the input flow. However, experts are inputting faulty letters to a lesser extent, the more sophisticated and "out of the box" the input method is, which could be linked to technical proficiency and the consequential adaptability.

It is important to keep in mind that the calculation of error rates is more complex than text entry speed, as different metrics highlight different aspects and look at errors from different perspectives (as mentioned in 4.1.1).

To achieve a frame of reference, we can compare the error rates to other physical input methods of Arif's research in 2009. All methods are performing worse than the common physical keyboard (1.8%) and similar or better compared to a 12-Key keyboard as found on older mobile phones (9.10%) [2].

Input Method	Participants	Error Metric	**Error Rate ↓**	WPM
Drumstick Keyboard	14	EKS ER	9.22	6.91
12-Key Multi-tap	55	EKS ER	9.10	9.94
Virtual Keyboard	14	EKS ER	8.76	6.42
Analog Stick Keyboard	10	EKS ER	5.35	2.80
Physical Keyboard	25	ER*	1.80	75.84

Table 5.1: Error rate comparison with select physical input methods

* $ER = \frac{INF}{|T|} \times 100\%$, where INF (Incorrect Not Fixed) is the number of unnoticed errors (incorrect characters) in the transcribed text [2].

5.2 Usability

In the following, we examine the average SUS score of each of the input methods, deriving their usability. A high score is considered positive with the points ranging from 0 to 100.

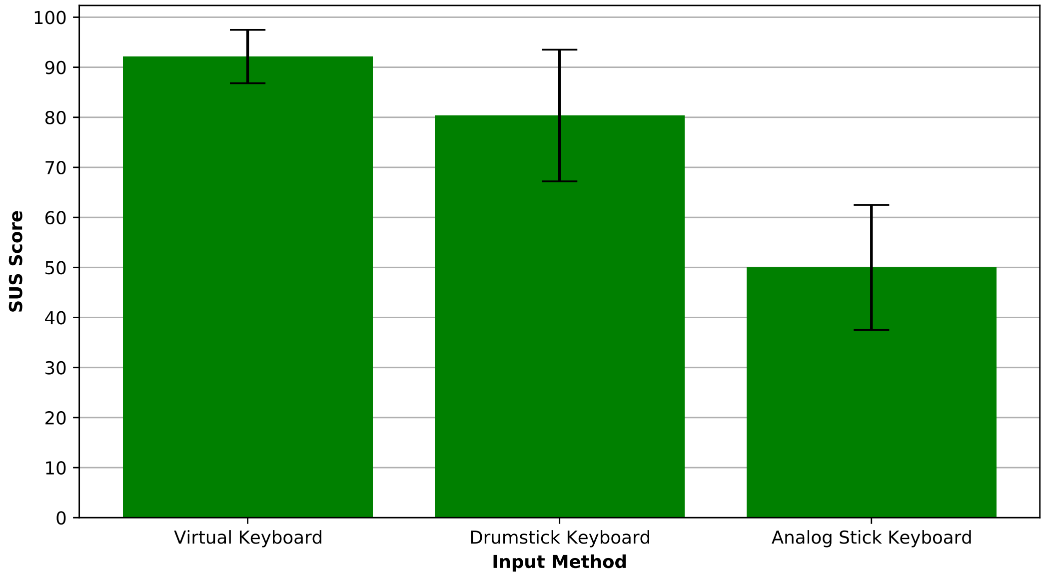

Figure 5.5: SUS Score per input method

5.5 shows that the Virtual Keyboard achieved a very high average score of 92.14 (SD = 5.33), outclassing the Drumstick Keyboard, which achieved a score of 80.36 (SD = 13.16). The Analog Stick keyboard has been ranked as the least pleasant to use with a SUS score of 50 (SD = 12.5). These results imply that despite the higher typing speed of the Drumstick Keyboard, participants considered the Virtual Keyboard more pleasant to use. This could be explained by the factor or familiarity of the Virtual Keyboard, as the use of drumsticks to operate a keyboard is rarely seen as intuitive without any explanation.

Furthermore, the Analog Stick keyboard requires a lot of initial explanation, which likely contributed to the low score as well as the need to get accustomed to the layout of the keys. Many participants pointed out the learning curve and the need to get to know the layout before feeling they could use the system with confidence. However, a few individuals stated that this particular input method would be their method of choice given more familiarity (such as the memorization of the layout in order to use the keyboard blindly) and more visual and functional polish.

For the Virtual Keyboard and the Drumstick Keyboard, many participants wished for a shortcut button for (back-)space inputs that don't require movement akin to a letter input.

In addition, both keyboards seemed to be too sensitive on space bar keystrokes, which would result in unwanted double-spaces in the text. While this was a common mention, it is hard to find the correct spacing required for the reset position of the keys, as every user has a different typing movement “style”. Another problem with the same origin is the input of double letters, resulting in unreliable and unwanted behavior from a user’s perspective. It is likely that the gripes mentioned above lead to a lower score.

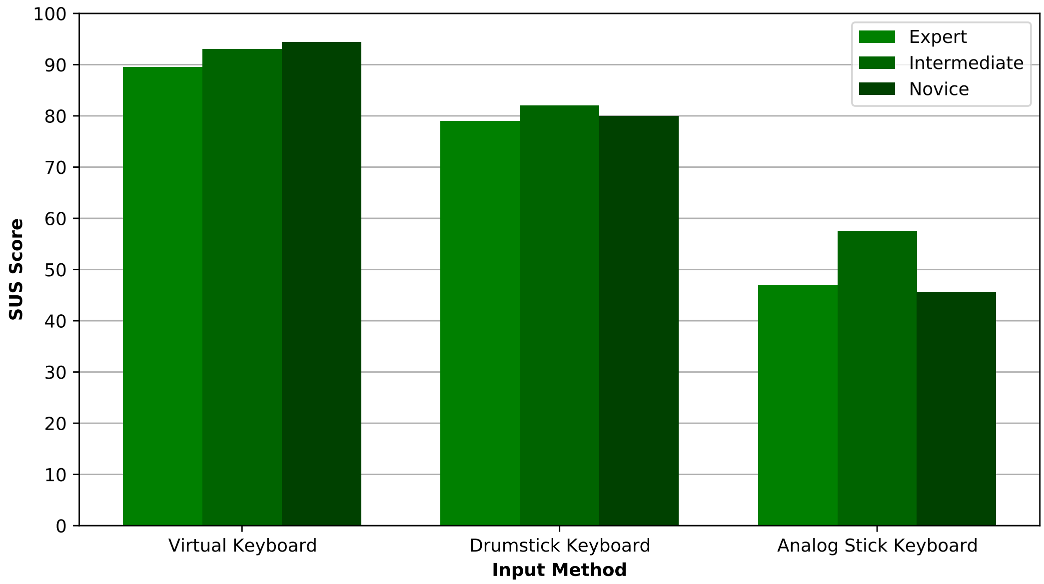

Figure 5.6: SUS Score per input method, grouped by expertise

Grouping SUS scoring by expertise (5.6) reveals that experts tend to judge more strict and intermediate users more graciously when it comes to usability. Dealing with technology on a daily basis can raise expectations, which could explain the lower scores from the expert group. In general, the scoring is in the same range for all expertise groups, with the intermediate participants having a stronger affinity to the Analog Stick keyboard compared to experts and novices.

We can narrow the scope even further and examine the scoring for the individual questions that make up the System Usability Scale and compare the average score per question for each input method.

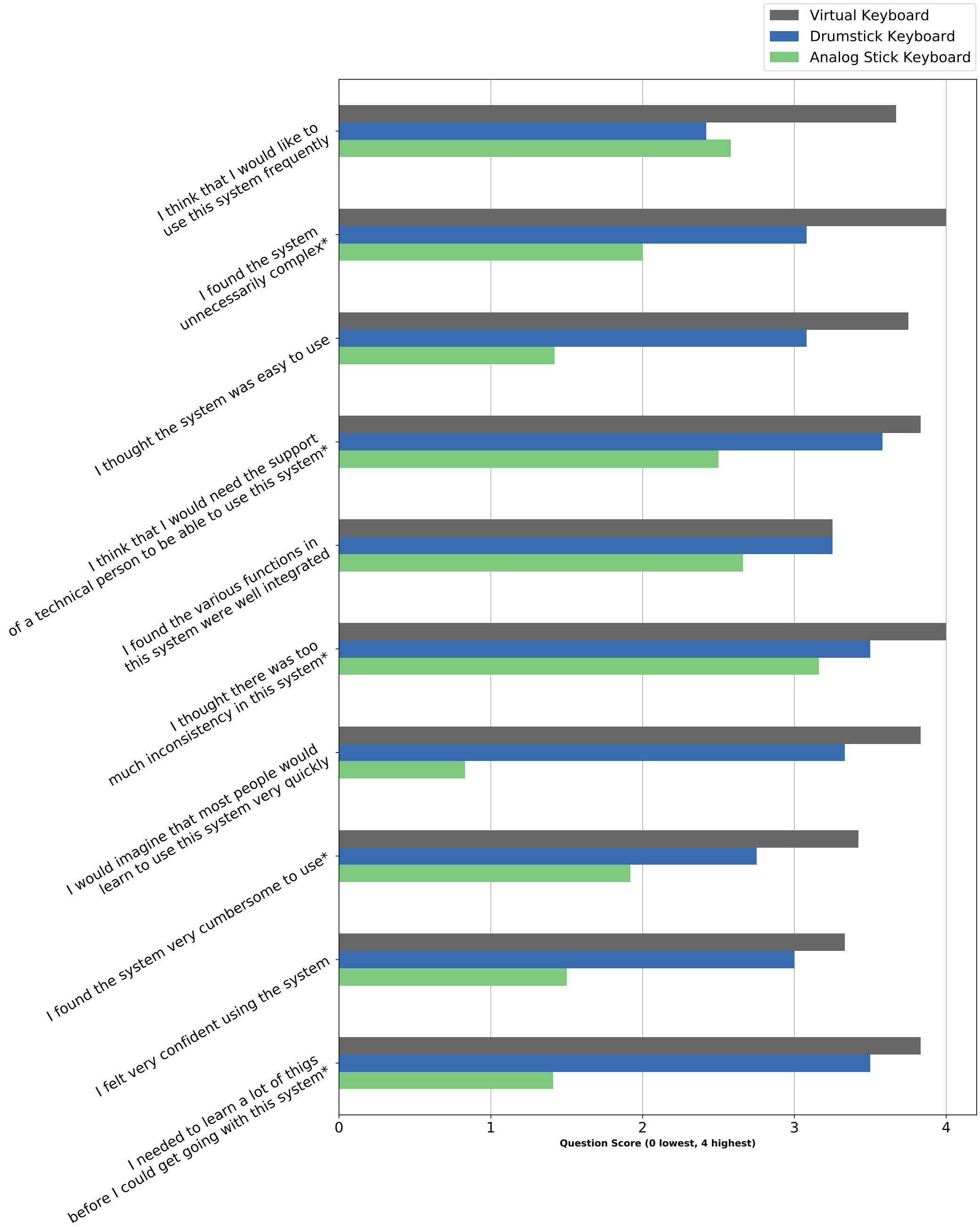

Figure 5.7: SUS Score grouped by individual question, a * indicates that disagreement to the statement leads to a high score

In 5.7, there was no individual SUS question where the Virtual Keyboard is bested by any of the other keyboards, meaning that the Virtual Keyboard is the preferred input method of choice for every sub-metric the SUS employs. Interestingly, participants agree that they would rather use the Analog Stick keyboard more frequently compared to the Drumstick Keyboard, despite the text entry speed and usability score being considerably lower.

6 Conclusion

Three text input methods for use in Virtual Reality space were implemented and compared in terms of performance and usability by means of a user study.

One of the input methods, namely the Analog Stick keyboard, is an original design where no prior implementation existed. It ranked below the Virtual Keyboard and Drumstick Keyboard in terms of speed and usability, but sparked interest among the participants for future use.

The best performing input method in terms of text entry speed and usability was the Virtual Keyboard. Furthermore, the amount of produced errors with the Drumstick Keyboard and the Virtual Keyboard was similarly high, while the least errors were produced with the Analog Stick Keyboard.

Overall, all the presented input methods lack the speed and precision that real-world physical keyboards are capable of, suggesting that there are still many innovations and improvements to come regarding VR text input.

7 Future Work

While the implemented input methods provide a solid foundation for text entry in Virtual Reality, there are many areas of possible improvements. Due to the modular nature of the implementation, modifications that contribute to the usability and feature richness are possible.

7.1 Improvements on the Keyboards

The keyboards offer many areas where user-friendliness and ease of use could be enhanced. The possible refinements can be grouped into conceptual ideas and ideas specific to the implementations present in the thesis.

7.1.1 Conceptual Improvements

In this part, we highlight ideas that could serve as a base for improvements for the input methods discussed before.

Haptic Feedback None of the provided input methods make use of the vibration function common in VR controllers to create haptic feedback. Haptic interfaces can greatly elevate the user experience of a system and improve productivity and efficiency [13]. It increases the interaction level between a user and the system by stimulating yet another sensory interface, adding a layer of immersion and thus, ease of use to the experience [1].

Improved Visual Feedback for Analog Stick Keyboard The visual feedback of the Analog Stick Keyboard is currently lacking in clarity and can be improved by for example indicating the active disk and keystrokes using colors. If a secondary disk is active, the remaining disks could contract into another to further indicate that a modificator is pressed. Furthermore, when disks of opposite hands collide, they clip into each other, which is visually unappealing. This could be improved by implementing a displacement system where the disks collide and move out of position upon contact.

More Symbols and Capital Letters Using Modificator Buttons The Drumstick Keyboard and Analog Stick Keyboard currently only supports input based on the label of the keys. However, introducing a key that replaces the labels with new symbols and letter could allow for a greater domain of the input possibilities. This would allow the Analog Stick Keyboard to input capitalized letters as well.

Alternative Layout for the Analog Stick Keyboard The layout of the Analog Stick Keyboard could be optimized according to the frequency of certain letters in the English language, instead of relying on the default US-ANSI layout. This would allow faster input of common letters, as it would be no longer required to press one or two modifiers in order to reach the desired letter.

7.1.2 Implementation-specific improvements

Here, we highlight areas that lack polish or are missing functionalities that could be added to the implementations of the input methods presented in this thesis.

Shortcut Buttons When common non-letter symbols such as space and backspace are desired, the user has to trigger the keystroke in the same way as all common alphanumeric letters. However, having the ability to insert a space or delete the letter preceding the cursor using a designated button on the controller was a much requested feature from the study participants. This feature would improve the text entry speed as the user can save time by directly accessing the shortcut instead of going through the motion of triggering the desired key.

Deadzone (Sensitivity) Adjustment An option to adjust parameters linked to resetting a pressed key (required distance of finger/drumstick) would allow customization for individual users, as a common remark was the keyboard behavior when trying to input the same letter twice or triggering the space bar. A similar feature could be introduced to the Analog Stick Keyboard that governs the required magnitude of stick displacement to be considered a keystroke.

Hold vs. Toggle Drumstick Participants were required to hold the drumsticks by keeping a grip button of the controller pressed, which lead to a decrease in usability as hand fatigue set in earlier. Having the option to toggle holding the drumstick or even introducing a fixed offset of the drumsticks could lead to a more satisfying user experience.

Predictive Text System Tying the input methods to a dictionary system that provides suggestions of possible words starting with the current input could improve the speed of the method considerably, more so with particularly slow performers such as the Analog Stick Keyboard. Important to note is that the prediction system would only work for languages with orthographic standards, which usually excludes slang and dialects.

7.2 Use of Alternative Error Rate Measurements

Using an error rate calculation which doesn't allow for an imperfect copy of a phrase could taint the quality of the results by restraining the observation to just one perspective. By utilizing varying methodologies to determine errors in text input, it is almost certainly always necessary to re-conduct the study in an at least similar environment in order to be truly conclusive. Allowing the user to commit phrase copies with mistakes can greatly influence text entry speed, as the WPM metric is independent of errors present in the text. Shortcomings of accuracy metrics in text input is a highly discussed topic in the field of HCI, with researchers constantly looking for new metrics that combine various factors [15].

7.3 Exploration of Other Input Methods

While the thesis encompasses the examination of three input methods, there are countless other possibilities to produce a solution that can be used to generate text, not constrained to using a controller. For instance, theses written parallel to this one study methods like free-hand writing or swipe-mechanism keyboards in VR. Speech-to-Text engines also present a possibility for text generation, regardless of reality plane.

As technology (related and unrelated to VR) advances, standards can change, and new methods can be adopted. It is the innovations of tomorrow that will decide how the landscape of VR text input will develop, and taking inspiration by the discoveries in related fields is not only crucial, but also an inspiring path where collaboration will yield results for generations to come.

Bibliography

[1] Faieza Abdul Aziz and Maryam Mousavi. A review of haptic feedback in virtual reality for manufacturing industry. *Journal of Mechanical Engineering*, 40:68, 09 2009. doi: 10.3329/jme.v40i1.3476.

[2] Ahmed Arif and Wolfgang Stuerzlinger. Analysis of text entry performance metrics. pages 100 – 105, 10 2009. doi: 10.1109/TIC-STH.2009.5444533.

[3] R.U. Ayres and K. Martinás. *On the Reappraisal of Microeconomics: Economic Growth and Change in a Material World*. Edward Elgar, 2005. ISBN 9781845422721. URL https://books.google.ch/books?id=ksxK7J95IF8C.

[4] Costas Boletsis. The new era of virtual reality locomotion: A systematic literature review of techniques and a proposed typology. *Multimodal Technologies and Interaction*, 1(4), 2017. ISSN 2414-4088. URL https://www.mdpi.com/2414-4088/1/4/24.

[5] Costas Boletsis and Stian Kongsvik. Controller-based text-input techniques for virtual reality: An empirical comparison. *International Journal of Virtual Reality*, 19(3):2–15, Oct. 2019. doi: 10.20870/IJVR.2019.19.3.2917. URL https://ijvr.eu/article/view/2917.

[6] John Brooke. *SUS – a quick and dirty usability scale*, pages 189–194. 01 1996.

[7] Andrey Doronichev. Daydream labs: Exploring and sharing vr's possibilities, May 2016. URL https://blog.google/products/google-ar-vr/daydream-labs-exploring-and-sharing-vrs/.

[8] Ralph Gasser, Luca Rossetto, Silvan Heller, and Heiko Schuldt. Cottontail db: An open source database system for multimedia retrieval and analysis. 11 2020. doi: 10.1145/3394171.3414538.

[9] Jens Grubert, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. Text entry in immersive head-mounted display-based virtual reality using standard keyboards. *CoRR*, abs/1802.00626, 2018. URL http://arxiv.org/abs/1802.00626.

[10] Ben Lang. Vr usage on steam jumps to all time high - but why?, Jun 2022. URL https://uploadvr.com/steam-vr-usage-all-time-high-why/.

[11] Yongjae Lee and Gerard J. Kim. Vitty: Virtual touch typing interface with added finger buttons. In Stephanie Lackey and Jessie Chen, editors, *Virtual, Augmented and*

Mixed Reality, pages 111–119, Cham, 2017. Springer International Publishing. ISBN 978-3-319-57987-0.

[12] I. Mackenzie and R. Soukoreff. Phrase sets for evaluating text entry techniques. *Conference on Human Factors in Computing Systems - Proceedings*, 05 2003. doi: 10.1145/765891.765971.

[13] Grégoire Richard, Thomas Pietrzak, Ferran Argelaguet, Anatole Lécuyer, and Géry Casiez. Studying the role of haptic feedback on virtual embodiment in a drawing task. *Frontiers in Virtual Reality*, 1, 2021. ISSN 2673-4192. doi: 10.3389/frvir.2020.573167. URL https://www.frontiersin.org/article/10.3389/frvir.2020.573167.

[14] Giuseppe Riva and Brenda Wiederhold. The new dawn of virtual reality in health care: Medical simulation and experiential interface. *Annual Review of CyberTherapy and Telemedicine*, 13:3–6, 12 2015. doi: 10.3233/978-1-61499-595-1-3.

[15] André Rodrigues, Hugo Nicolau, Diogo Branco, Jay Rainey, David Verweij, Jan Smeddinck, Kyle Montague, André Santos, and Tiago Guerreiro. Investigating the tradeofs of everyday text-entry collection meth-ods. 05 2022. doi: 10.1145/3491102.3501908.

[16] Luca Rossetto, Ivan Giangreco, Silvan Heller, Claudiu Tănase, and Heiko Schuldt. Searching in video collections using sketches and sample images – the cineast system. In Qi Tian, Nicu Sebe, Guo-Jun Qi, Benoit Huet, Richang Hong, and Xueliang Liu, editors, *MultiMedia Modeling*, pages 336–341, Cham, 2016. Springer International Publishing. ISBN 978-3-319-27674-8.

[17] Luca Rossetto, Ivan Giangreco, Claudiu Tanase, and Heiko Schuldt. Vitrivr: A flexible retrieval stack supporting multiple query modes for searching in multimedia collections. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, page 1183–1186, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450336031. doi: 10.1145/2964284.2973797. URL https://doi.org/10.1145/2964284.2973797.

[18] Peter A. Smith. Using commercial virtual reality games to prototype serious games and applications. In Stephanie Lackey and Jessie Chen, editors, *Virtual, Augmented and Mixed Reality*, pages 359–368, Cham, 2017. Springer International Publishing. ISBN 978-3-319-57987-0.

[19] Diego Vergara, Manuel Pablo Rubio, and Miguel Lorenzo. On the design of virtual reality learning environments in engineering. *Multimodal Technologies and Interaction*, 1(2), 2017. ISSN 2414-4088. doi: 10.3390/mti1020011. URL https://www.mdpi.com/2414-4088/1/2/11.

A
Written Feedback

A.1 Virtual Keyboard

‘I had no VR experience, understandable without explanation’

‘Self-explanatory and easy to use.’

‘Difficult to use when hitting a button more than once.’

‘I misstyped [sic.] many keys, so a quick backspace key would be useful.’

‘- Double triggers when “pushing too far down” on the keys (fixable)
- Space/bcksp shortcut buttons?’

A.2 Drumstick Keyboard

‘A short introduction was enough. Sticks could disappear in the keys, which was troublesome. Struggled with the space bar.’

‘Not so easy to use when nervous/hands are shaking. Delay for space bar would be nice.’

‘Cool! But again, hitting a button twice in a row does not work reliably.’

‘Really fun to use! Takes a while to get the sticks aligned correctly. I liked pressing keys from bottom and top alternatively while typing. I kinda got into the flow.’

‘need to keep grip pressed’

‘no arrow keys :(no drum sound’

A.3 Analog Stick Keyboard

‘System needs to get used to, needs more training. Less intuitive to understand.’

‘If one doesn’t have a left-right weakness, this would be the best input method for me.’

‘I have the impression there is a learning curve. But for the first few attempts it’s very hard to use.’

‘Steep learning curve. I could imagine it to be useful once you get used to it, as you don’t need to watch the keyboard while typing, but as a beginner I had a lot to think about.’

‘R on left hand and L on right hand collide when holding hands together.’

‘Hard point to get: not super visually clear which dish is active...’

B

Raw Experiment Data

ID	Gender	Age	Expertise	Virtual Keyboard			Drumst. Keyboard			An. St. Keyboard		
				WPM	ER	SUS	WPM	ER	SUS	WPM	ER	SUS
1	Male	23	Expert	5	17	92.5	5	15	87.5	2.8	-	32.5
2	Male	23	Expert	6.83	5.44	80	7.26	7.59	57.5	3.85	-	35
3	Male	30	Interm.	8.83	10.96	95	6.28	15.6	67.5	3.72	3.71	62.5
4	Male	21	Interm.	3.9	15.29	90	4.73	8.89	87.5	-	-	-
5	Male	23	Expert	7.43	3.24	92.5	7.68	4.58	92.5	-	-	-
6	Male	28	Interm.	3.56	14.46	95	6.05	19.4	75	2.05	6.54	45
7	Male	22	Novice	5.39	4.32	92.5	4.51	4.94	65	2.3	4.89	57.5
8	Male	27	Expert	5.22	19.96	90	7.97	6.56	67.5	3.41	10.08	57.5
9	Male	26	Expert	8.8	6.66	92.5	8.77	4.25	90	2.93	2.78	62.5
10	Male	25	Novice	6.69	8.72	87.5	11.29	8.08	95	1.92	7.61	35
11	Male	35	Intermediate	8.23	4.43	85	6.87	5.8	97.5	1.14	4.32	50
12	Female	55	Novice	7.72	1.28	100	6.93	11.61	65	2.72	0	52.5
13	Female	22	Intermediate	7.57	0	100	6.85	7.3	82.5	3.95	6.09	72.5
14	Female	25	Novice	4.66	10.91	97.5	6.48	9.48	95	2.84	7.44	37.5
Average		28		6.42	8.76	92.14	6.91	9.22	80.36	2.8	5.35	50
Standard Dev.		8.4		1.71	5.95	5.33	1.69	4.43	13.16	0.82	2.7	12.5

University of Basel

Faculty of Science

Declaration on Scientific Integrity
(including a Declaration on Plagiarism and Fraud)
Translation from German original

Title of Thesis: Analysis and Comparison of Text-Input Methods in Virtual Reality

Name Assesor: Prof. Dr. Heiko Schuldt

Name Student: Nakarin Jerom Srijumrat

Matriculation No.: 2018-054-882

With my signature I declare that this submission is my own work and that I have fully acknowledged the assistance received in completing this work and that it contains no material that has not been formally acknowledged. I have mentioned all source materials used and have cited these in accordance with recognised scientific rules.

Place, Date: Basel, 27.06.2022 Student: ______

Will this work be published?

○ No

● Yes. With my signature I confirm that I agree to a publication of the work (print/digital) in the library, on the research database of the University of Basel and/or on the document server of the department. Likewise, I agree to the bibliographic reference in the catalog SLSP (Swiss Library Service Platform). (cross out as applicable)

Publication as of: Basel, 27.06.2022

Place, Date: Basel, 27.06.2022 Student: ______

Place, Date: ______ Assessor: ______

Please enclose a completed and signed copy of this declaration in your Bachelor's or Master's thesis .

August 2021