HEILBRONN UNIVERSITY
OF APPLIED SCIENCES

SUBJECT (SPO1)

Enhancing User Experience and Productivity in Virtual Reality (VR) Programming Environments

KHEIREDDINE AZZEZ*

June 29, 2024

Submitted to
Prof. Dr.-Ing. Gerrit Meixner & Prof. Dr. Andrii Matviienko

*215021, kazzez@stud.hs-heilbronn.de

Contents

List of Abbreviations

AHAP Apple Haptic Programming

AI Artificial Intelligence

AMOLED Active Matrix Organic Light Emitting Diode

CC Continuous Cursor

CP Control Pointing

DC Discrete Cursor

DoF Degrees of Freedom

ER Error Rate

FFB Force Feedback

FH Free Hand

GPU Graphics processing unit

HCI Human-Computer Interaction

HMD Head-Mounted Display

HP Head Pointing

LCD Liquid Crystal Display

ML Machine Learning

OLED Organic Light Emitting Diode

SDK Software Development Kit

TAM Technology Acceptance Model

UI User Interface

VR Virtual Reality

WPM Words Per Minute

XR Extended Reality

List of Figures

List of Tables

Management Summary

This thesis investigates the enhancement of user experience and productivity within VR programming environments, focusing particularly on the integration of advanced Force Feedback (FFB) and optimized virtual keyboard design. As VR technologies evolve, there is a growing need to create more intuitive and immersive interaction systems that mimic the real-world experience as closely as possible.

The study explores various methodologies for improving VR interfaces, particularly through the implementation of dynamic and adaptive FFB systems that provide users with realistic tactile responses. This approach aims to increase the naturalness of interactions in VR, making them feel more intuitive and less fatiguing, which is critical for productivity in programming and other detailed-oriented tasks.

A significant portion of the research is dedicated to the development of a virtual keyboard optimized for use in VR. This includes experimenting with different layouts and feedback mechanisms to determine the most efficient and comfortable configurations for long-term use. By simulating realistic keyboard feedback and improving the ergonomic design of virtual input devices, the thesis aims to reduce the cognitive load and physical discomfort typically associated with virtual typing.

The findings suggest that enhancing FFB and optimizing keyboard ergonomics can significantly boost both the user experience and productivity within VR programming environments. The study provides a comprehensive set of guidelines for developers and designers to create more user-friendly VR systems, potentially influencing future developments in VR technology.

1 Introduction

In the rapidly evolving landscape of VR, spearheaded by industry behemoths such as Meta [10], Microsoft [15], and Apple [63], who are investing billions in immersive technologies, we are witnessing a transformative shift. This shift transcends the physical constraints of our world, opening up a realm of limitless possibilities. The advancements in Graphics processing unit (GPU) [8] and computing capabilities, along with powerful gaming engines, have brought us to a crucial crossroads on the path to a fully immersive virtual existence. The level of progress we have reached in this field was beyond imagination just a few decades ago.

In our understanding of reality, sensation plays a foundational role as it involves receiving and interpreting information from our surroundings through our sensory systems. Humans and their environments are inherently designed with biological interfaces that facilitate direct interaction, allowing sensations to be translated seamlessly without the need for mediators. This intrinsic connection between individuals and their environments forms the basis of our perceptual experience, where each sensory input is integrated and acted upon holistically [23].

In the realm of VR, this direct translation of sensation presents both a challenge and an opportunity. VR seeks to replicate this seamless interface in a digital context, aiming to create environments where users can interact with virtual objects as intuitively as they would with physical ones. The fidelity of these interactions in VR hinges on the ability to accurately simulate and elicit natural human responses to virtual stimuli, making the study of human sensory and perception systems critical to advancing VR technologies [76, 68, 38, 12, 27].

1.1 Motivation

Given this groundbreaking shift, this new paradigm presents a meta-problem: how do we translate the complex language of human sensation and perception [23] into the virtual domain? The interface between humans and computers, especially in immersive environments such as VR, poses significant challenges due to the need to accurately replicate sensory inputs so that they mimic the natural interactions one would expect in the real world [65, 67]. One area grappling with this challenge is software development, where text entry remains a fundamental skill for coding and debugging. The keyboard, a historical mainstay in programming, continues to be an indispensable tool. Despite explorations into voice recognition and gesture-based interfaces, efficient typing is still closely tied to coding proficiency [16]. Yet, the challenge becomes even more pronounced in VR, where the tactile and precise nature of traditional keyboards is disrupted, and each Head-Mounted Display (HMD) supplier

offers isolated solutions without standardization [87].

In VR, not only is the physical interaction with keyboards transformed but the perceptual alignment of visual, tactile, and kinesthetic cues is also critical for user proficiency and comfort. The lack of a standard approach complicates the development of universal solutions that could benefit all users across different systems. Moreover, the quest for efficient text entry in VR is further complicated by the diversity of user backgrounds and ergonomic needs, which require adaptable and flexible input methods [12]. This makes it essential for VR platforms to evolve beyond mere replication of the physical tools and to innovate towards more intuitive and integrated input methods that cater to the needs of diverse user populations.

1.2 Research Goals

This literature review aims to tackle the specific issue of enhancing text entry in VR, a fundamental challenge that bridges human-computer interaction and virtual environment design. Given the rapid typing speeds and extended daily keyboard use by programmers, accuracy issues in existing virtual reality tracking systems are a significant concern. The review methodically delves into existing research, identifying gaps in current methodologies and synthesizing insights that could shape the future trajectory of work in enhancing user experience and productivity within the virtual reality programming sphere. The goal extends beyond merely replicating the physical keyboard in a virtual space. Instead, it seeks to leverage the unique properties of VR—such as spatial interaction, multimodal integration, and context-aware interfaces—to create a more efficient and intuitive typing experience.

Innovations in VR text entry are poised to transform how users interact with software in immersive environments, facilitating smoother and more natural interactions that could significantly boost coding and debugging efficiency. By integrating insights from fields such as ergonomics, cognitive psychology, and artificial intelligence, this research aims to develop a set of design principles and technical guidelines that will inform the development of next-generation VR keyboards. These enhanced interfaces are expected not only to improve the speed and accuracy of text entry but also to reduce the cognitive load on users, thereby making VR more accessible and appealing for a broader range of professional and recreational applications.

By addressing the challenges of text entry in VR, this work will contribute to the overarching goal of making virtual environments as functional and user-friendly as their physical counterparts, thereby paving the way for more widespread adoption and innovative uses of VR technology.

2 Background

Before delving into the nuanced aspects of virtual reality text entry methods, we offer a comprehensive overview by scrutinizing diverse studies and articles. Employing three key questions—"Why is it being implemented," "How is it being executed," and "What is being utilized"—our objective is to underscore the shared intersections and innovative approaches within these studies. Drawing on the Technology Acceptance Model (TAM) as developed by Davis [18], we apply a theoretical lens to examine these aspects, focusing on the perceived usefulness and ease of use of VR text entry systems. This approach not only helps us understand the factors driving the adoption and practical execution of these methods but also assists in identifying the specific tools and interfaces currently in use. Furthermore, by considering the references and nomenclature utilized, we aim to leverage them as a foundational basis for more advanced research on this subject. This section will conclude by identifying the unexplored areas and the knowledge gaps in the examined studies, offering a structured pathway for future investigations.

In the context of VR, text entry methods are pivotal for enhancing user interaction and functionality. There are three primary techniques commonly employed: The use of physical devices such as controllers, gamepads, joysticks, and keyboards [20] with or without the pass-through technique, which allows users to see their physical keyboard in the virtual environment. This integration significantly improves typing speed and accuracy [22]. Body

Figure 2.1: VR Controller (Meta Quest 3)

Figure 2.2: Gamepad (PS Gamepads)

Figure 2.3: Joystick (HTC Vive)

Figure 2.4: Keyboard (Logitech)

Gestures, including movements of the head, hands (including fingers), and even blinking, provide alternative interaction methods. For instance, BlinkType and NeckType leverage users' eye blinks and neck forward and backward movements to select letters , thus offering hands-free interaction[42].

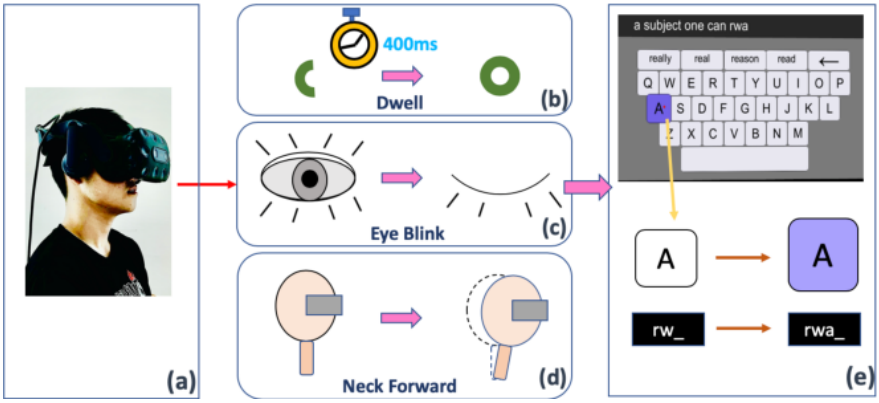


Figure 2.5: An illustration of the text entry with three hands-free techniques in VR [42].

The realm of haptics also plays a crucial role, with self-referenced haptics focusing on the user's tactile sensations on their own body [26]. Imagine a virtual button placed on the hand of a virtual player; this connection between the user's haptic feedback and their physical presence introduces a new layer of immersion and interaction in VR environments.

Moreover, the Speech-to-Text technique, which converts spoken language into written text, offers a viable alternative when manual entry is impractical or undesired [9]. The development and refinement of VR text entry techniques have been guided by the need for methods that are both efficient and minimally disruptive, as traditional input devices often do not translate well into VR settings [24].

The integration of these text entry methods into VR applications has opened new avenues for research and development. Studies have shown that the efficiency of text entry systems inVR can significantly affect user satisfaction and performance [46]. As VR technology continues to evolve, the exploration of more intuitive and natural text entry methods will be crucial. This involves not only technological innovation but also a deep understanding of human-computer interaction principles and user experience design [83].

Future research should focus on addressing the limitations of currentVR text entry systems, such as the cognitive load imposed on users and the physical discomfort that may arise from prolonged use. Exploring adaptive systems that can adjust based on the user's input speed and accuracy could provide breakthroughs in usability [28]. Additionally, cross-disciplinary studies involving ergonomics, linguistics, and machine learning could lead to the development of more advanced systems that better understand user intent and provide more personalized and context-aware interactions [88].

In conclusion, while significant strides have been made in VR text entry technologies, considerable gaps remain in the literature, particularly in terms of comprehensive usability

studies and long-term user engagement. Addressing these gaps will not only advance our understanding of VR interfaces but also enhance the overall user experience in virtual environments.

2.1 Comfortability for virtual typing

In terms of these techniques, long-term comfort is a critical factor when typing on a keyboard within the virtual world. Studies such as those by [20] and [87] have highlighted the ergonomic challenges associated with VR text entry methods. For instance, head pointing (HP) techniques, where a ray extends from the main camera to facilitate selection, can lead to neck discomfort and dizziness [21]. Similarly, Free Hand (FH) and Control Pointing (CP) techniques, while fostering innovation in interaction methods, impose significant physical workloads and can lead to 'gorilla arm fatigue', a term denoting the fatigue resulting from prolonged elevation and use of arms [7].

Moreover, techniques such as Discrete Cursor (DC) and Continuous Cursor (CC) introduce their challenges; both methods may cause texting thumb pain due to repetitive movement [22]. An alternative approach involving thumb-to-finger pinch gestures, which offers high spatiotemporal perception and has been reported to increase user satisfaction compared to traditional VR keyboards, was explored in [87]. This method exemplifies how nuanced gesture control can mitigate some of the physical discomfort associated with more traditional methods.

A collaborative study conducted by researchers from the University of Cambridge and the University of Toronto, in partnership with Meta-Reality Labs [41], explored user experiences with different text entry setups in virtual environments. The study found that typing on a virtual keyboard positioned on a physical table led to higher comfort ratings compared to typing in mid-air with either ten or two fingers, showcasing varied comfort levels dependent on the interface used [21]. These findings underscore the importance of adaptive interfaces that can dynamically adjust to the user's ergonomic needs and environmental contexts, potentially through the integration of AI-driven systems that learn from user behavior to optimize interface layout and interaction modalities [85].

In addressing the future of VR text entry, it is crucial to consider the integration of multimodal feedback systems that combine tactile, visual, and auditory cues to enhance the overall user experience. Such systems could alleviate some of the cognitive and physical strains identified in current technologies by providing more intuitive and natural interaction paradigms [26]. Additionally, advancing haptic technologies could offer more refined and realistic feedback, further enhancing the sense of presence and reducing the disconnect be-

tween user actions and system responses [60].

Ultimately, the evolution of text entry in virtual reality is contingent upon a multidisciplinary approach that embraces advances in ergonomics, human-computer interaction, and artificial intelligence. By continually refining these systems, we can move closer to creating VR environments that are not only effective in terms of task completion but also in promoting user well-being and satisfaction.

2.2 Keyboard's Modernity for virtual typing

Secondly, the modernity of the keyboard in virtual environments is a crucial factor for long-term use. During the study [87], researchers assessed the position and size of the virtual keyboard as essential determinants of usability, emphasizing the keyboard's spatial orientation, including considerations such as left or right-side placement and the optimal distance from the user. The Free Hand (FH) method was noted for its novelty, whereas the Control Pointing (CP) method was found to be most attractive, followed by FH [46].

The type of keyboard consistently used across these studies was Qwerty. However, the shape of the keyboard also impacts usability; two or three-dimensional configurations depend on the application's requirements. For instance, in the study [21], a 2D keyboard was selected because it was fixed to a wooden table and featured keys in a quadratic form, or involved a specific gesture like thumb-to-finger pinch [87]. Conversely, another study [7] preferred a 3D keyboard because it floats within the virtual environment, offering a more immersive experience [88].

Moreover, the study [60] introduced an innovative design called FLIK, where the keyboard keys are arranged in the form of a sphere, deviating from the traditional layout. This spherical arrangement potentially enhances the ergonomic comfort and reduces the cognitive load by aligning more naturally with human hand movements [75]. Additionally, another study [41] pushed the boundaries even further by transforming the keyboard into a flower-shaped configuration while maintaining the order of the QWERTY keys, aiming to combine aesthetic appeal with functional ergonomics [83].

Each of these innovations reflects ongoing efforts to optimize text entry in virtual environments, highlighting the importance of adaptable interfaces that cater to diverse user needs and preferences [9, 28]. As virtual reality technology continues to evolve, the exploration of alternative keyboard layouts and their impacts on user performance and satisfaction will remain a vital area of research [66, 55].

The continuous advancement in VR technology requires a reevaluation of traditional design principles to include virtual ergonomics and the physiological effects on users, advocating for designs that prevent fatigue and increase user satisfaction [14, 35].

2.3 Keyboard's Accuracy for virtual typing

When it comes to accuracy, the initial metrics that come into play before pressing a key on the virtual keyboard are key detection and the prevention of typing errors. In a pivotal study [7], characters are registered when the finger reaches specific distances, precisely 3 cm away from the center of the key (4.5 cm for the Pinch keyboard, as the radius of a sphere was 1.5 cm), and the virtual hands were made semi-transparent to strategically avoid potential collisions. This study proposes a novel technique involving the use of finger-to-thumb as a double-factor confirmation for character entry, enhancing accuracy by confirming intent before final character selection.

Furthermore, the virtual keyboard automatically deactivates all other keys when a collision occurs between the target key and the cursor of the index finger, positioned at the index fingertips to enhance precision during the entry process. This approach is rooted in findings from ergonomic research which suggests that minimizing extraneous movements and providing tactile feedback can significantly reduce entry errors and increase typing speed [47, 84].

The integration of spatial and gestural controls as seen in this study aligns with the principles outlined in [55], which advocate for natural user interfaces that conform to human motor capabilities and cognitive functions. Additionally, studies such as [88] and [9] highlight how leveraging deep learning for predictive text input and error correction algorithms can further refine the accuracy of virtual keyboards by anticipating user input and adjusting the interface dynamically.

The use of semi-transparent virtual hands and strategic cursor placement also draws upon principles from visual perception psychology, as discussed in [82], to reduce visual clutter and enhance the user's focus on target elements, thereby reducing cognitive load and potential input errors. This multidisciplinary approach to interface design, combining HCI, ergonomics, and psychology, is pivotal in developing more intuitive and user-friendly virtual environments [35, 14].

2.4 Haptic Feedback and Visual Feedback in Virtual Environment

In the realm of feedback within virtual environments, three distinct types are typically identified: haptic, visual, and auditory. In specific studies [60, 87], haptic feedback primarily involves vibrotactile sensations generated through controllers. This modality not only aids in conveying spatial relationships more effectively than visual and audio feedback alone but also enhances the user's sense of presence by simulating physical interactions, making these relationships more tangible [26].

While haptic feedback plays a crucial role, visual feedback is also emphasized due to the limited options available for haptic enhancements. In a novel approach, a pseudo-haptic keyboard is introduced with three key features: Protrusion, Hit Effect, and Penetration Blocking [7]. Additionally, a unique pinch keyboard design with bubble-shaped keys is implemented to further enhance user guidance, where the nearest key to the target key is highlighted with a distinct color. Participants in these studies expressed a preference for the Pseudo-haptic and Pinch keyboards, reporting stronger haptic sensation and embodiment compared to a Normal keyboard [88].

Furthermore, another study [60] employed a technique called "Big Key," an algorithm that dynamically adjusts virtual keyboard key sizes based on the likelihood of the next letter in a word. After the initial keystroke, BigKey searches for possible words and scales key sizes according to the frequency of subsequent characters. This approach demonstrated superior performance and high user satisfaction [46]. Conversely, Word Disambiguation, another visual aid technique used in the same study, suggests likely words based on their frequencies as keys are entered, presenting the top three ranked options for efficient word completion. However, this method led to increased frustration and higher mental demand among users [9].

Each of these feedback mechanisms, whether haptic or visual, plays a pivotal role in shaping the usability and user experience within virtual environments. The integration of advanced feedback techniques, guided by ergonomic and cognitive principles, is crucial for developing interfaces that are both intuitive and effective [35, 55].

2.5 Performance for Virtual Typing

The performance of virtual typing is not a standalone factor; it is intricately linked to various elements, including the technique employed, the type of feedback (haptic or visual), and the specific use case. Commonly used metrics to measure this performance include Words Per Minute (WPM), Error Rate (ER), Depth of Pressing, Press Duration, and Hand Movements.

Words Per Minute (WPM) is typically defined as the average number of words entered by a user in one minute, with a word usually considered as five characters, including spaces and punctuation:

$$\text{WPM} = \left(\frac{|T| - 1}{5s} \right) \times 60 \tag{2.1}$$

The Error Rate (ER) is the percentage of characters entered incorrectly by the user. ER is calculated by dividing the number of errors by the total number of characters in the text, where P and T denote the presented and transcribed text, and MSD is calculated using Levenshtein's algorithm [43]:

$$\text{ER}(\%) = \frac{100 \times MSD(p,T)}{\max(|p|,|T|)} \tag{2.2}$$

Press Duration is the average time that the user's fingers stay on a key after pressing it on the virtual keyboard. This metric can significantly affect the user's feedback, confidence, and satisfaction with VR [20]. Additionally, the Depth of Pressing is the average distance that the user's fingers travel in the depth dimension when pressing a key on the virtual keyboard. Depth of pressing can influence the user's fatigue, comfort, and immersion in VR [20].

These metrics have been determined through two distinct methods: one involves analyzing the logs generated from keyboard typing, while the other employs a third-party service for calculation. For example, the system in one study [60] records the raw input stream. Upon completion, the testbed logs relevant details in a file. In another study [7], all text entries were conducted through a third-party web application.

2.6 Unveiling Omissions

2.6.1 Immersion and Realism

Initial studies in the field of VR often focused narrowly on haptic feedback limited primarily to simple vibrations, which primarily affected the hand without offering a full spectrum of tactile sensations. This limited approach often led to incomplete sensory experiences, underscoring the need for a more comprehensive exploration of tactile feedback in VR. Recent advancements have emphasized the importance of simulating nuanced sensations such as those experienced when grasping or releasing objects. For instance, a study [5] discusses the development of a device that enhances directional sensations in VR through mechanical force concentrations using motor rotations. This technology significantly improves the realism of

physical interactions, thereby elevating the user's sense of presence and immersion [47, 75].

Additionally, the study "Accelerating Skill Acquisition of Two-Handed Drumming using Pneumatic Artificial Muscles" [17] exemplifies the use of advanced haptic systems employing pneumatic artificial muscles to enhance the recall of complex patterns, demonstrating the potential of sophisticated haptic feedback to improve learning and performance in VR applications.

Moreover, while many studies have focused on distinct text entry methods, there is a notable gap in considering the integration of multimodal inputs, which combine hand gestures with voice commands or elements of auditory and tactile feedback. Such integration is crucial for creating more engaging, convincing, and user-friendly VR simulations [83]. The omission of these synergies could restrict the naturalness and intuitiveness of VR text input experiences. Recent research has begun to address these limitations by employing technologies like Word Disambiguation algorithms, which assist in text entry by predicting likely word completions [9].

Furthermore, while significant attention has been paid to the design of virtual keyboards, there remains a lack of focus on enhancing the virtual hand model to improve interaction during typing tasks. Most studies have maintained a static approach to keyboard orientation in virtual environments, neglecting the potential benefits of a dynamic setup that allows users to navigate and interact more freely within their virtual space. This static approach has often overlooked the significance of environmental textures and other sensory details that could enhance the typing experience and overall interaction quality [73].

The study "How to Get There When You Are There Already? Defining Presence in Virtual Reality and the Importance of Perceived Realism" [6] further explores the critical role of perceived realism in VR. It highlights how users inevitably compare virtual objects with their real-world counterparts and assess their congruence, affecting their overall experience and satisfaction with VR technologies [50].

2.6.2 Consistency

Consistency in haptic feedback, particularly within virtual environments, has traditionally been limited to static haptic effects—haptic signals that do not change during rendering. This static approach often results in a decrease in user engagement because the repetitive and unchanging nature of the effects fails to mimic the dynamic and varied tactile feedback experienced in real-world interactions [26]. Static haptic feedback can quickly become predictable, reducing the sense of immersion and realism that is crucial for effective virtual

environments.

Challenges of Static Haptic Feedback

Static haptic feedback systems do not account for changes in user actions or environmental factors, which can lead to a disconnection between the user and the virtual experience. For instance, when interacting with different textures or surfaces in a virtual world, static feedback would fail to provide the nuanced differences users would expect in a real-world scenario [46]. This lack of responsiveness can diminish the perceived quality of the VR system and could potentially lead to user dissatisfaction.

Dynamic Haptic Feedback for Enhanced Engagement

To address these limitations, more recent studies advocate for dynamic haptic feedback, which adjusts the intensity, pattern, and type of feedback based on the user's interactions and the context within the virtual environment [47]. For example, varying the feedback when a user touches different virtual objects—such as changing from a smooth to a textured sensation when moving from a glass surface to sandpaper—can significantly enhance the realism and user engagement.

Implementations and Technological Advancements

Several implementations of dynamic haptic feedback have been explored in advanced VR systems. Technologies such as haptic suits and gloves equipped with fluid-filled bladders or air pockets that adjust based on the virtual environment have been developed to provide more realistic tactile sensations [75]. For example, the Teslasuit integrates full-body haptic feedback that can simulate different environmental impacts, from the brush of wind to the impact of a virtual ball.

Practical applications of dynamic haptic feedback have been particularly noteworthy in fields such as medical training and remote operations, where the feel of different tissues or materials is crucial. A study involving surgical simulations found that trainees who used VR platforms with dynamic haptic feedback performed better and reported a higher level of confidence in their skills compared to those training with static feedback systems [88].

The future of haptic feedback in VR is likely to involve a greater integration of AI and machine learning technologies to further refine the responsiveness and adaptability of feedback systems. These advancements could lead to personalized haptic experiences, where the system adjusts not only to general user behavior but also to individual preferences and responses, potentially transforming how users interact with virtual environments [83].

In conclusion, while static haptic feedback has been a staple in early VR applications, the shift towards dynamic, responsive feedback systems represents a significant advancement in the field. By increasing the variability and realism of haptic experiences, VR systems can achieve higher levels of user satisfaction and engagement, paving the way for more sophisticated and immersive virtual environments.

2.6.3 Skill Transfer and User Experience

Prior research in virtual environments has largely focused on gathering performance data and subjective user feedback, yet often overlooked is the importance of descriptive effects in enhancing user experience. These effects, akin to the MIDI system for musical instruments, represent a method of encoding haptic feedback in a signal-independent manner, which involves detailing only the notes and their 'descriptive' attributes. For example, Apple's AHAP effect files, part of the CoreHaptics framework, utilize a JSON-based language for describing haptic effects [33]. This method allows developers to create expressive and varied haptic sensations without needing a direct motor signal, facilitating richer user interaction.

Moreover, while metrics like entry speed and error rates are commonly assessed, there is a notable gap in the exploration of other performance indicators that could provide a more holistic view of user experiences. These metrics include the learning curve associated with different text entry methods and the adaptability of users to varying techniques over time. These aspects are critical in understanding the long-term effectiveness and user satisfaction with VR interfaces. For example, the accuracy of typing, which is crucial for assessing the effectiveness of text entry methods, can be quantified in several ways:

$$\text{Accuracy in words: } \text{Accuracy} = \left(100\% - \frac{\text{Words with errors} \times 100\%}{\text{Total number of Words}}\right) \tag{2.3}$$

$$\text{Accuracy in characters: } \text{Accuracy} = \left(100\% - \frac{\text{Characters with errors} \times 100\%}{\text{Total number of Characters}}\right) \tag{2.4}$$

$$\text{Accuracy in keystrokes: } \text{Accuracy} = \left(100\% - \frac{\text{Incorrect keystrokes} \times 100\%}{\text{Total number of Keystrokes}}\right) \tag{2.5}$$

Case Studies and Practical Applications

Examining case studies where these metrics have been applied reveals their utility in enhancing VR training simulations, educational tools, and interactive media. For instance, a VR surgical training program that utilized adaptive feedback based on user performance metrics showed that trainees achieved significantly higher accuracy and reduced error rates [44].

Another example includes a VR language learning application that measured the adaptability of users to different instructional techniques over time, providing valuable insights into how VR can be optimized for faster learning curves and better retention rates [62].

Looking ahead, integrating a broader array of performance metrics into VR studies will enhance our understanding and development of user-centric VR systems. Future research should also consider the potential of emerging technologies such as artificial intelligence and machine learning to dynamically adapt VR environments to user behaviors, thereby improving the personalization and effectiveness of virtual training environments [70].

In conclusion, while significant strides have been made in capturing performance and subjective data in VR, a comprehensive approach that includes descriptive effects and a wider range of performance metrics is essential for the next generation of VR applications. Such an approach will not only refine current systems but also pave the way for more sophisticated and immersive user experiences.

2.7 Summary

This chapter provides a comprehensive overview of the varied VR text entry methods and their significance in enhancing user interaction and functionality. By employing three key questions—"Why is it being implemented," "How is it being executed," and "What is being utilized"—we have explored the shared intersections and innovative approaches within existing studies. The chapter introduces the primary text entry techniques in VR, including the use of physical devices, body gestures, and the Speech-to-Text technique. Each method's role in facilitating effective communication within virtual settings is discussed, highlighting their contributions to reducing cognitive load and increasing typing efficiency.

Additionally, the chapter identifies unexplored areas and knowledge gaps in the existing literature, particularly focusing on ergonomic challenges, the integration of adaptive systems, and the need for more comprehensive usability studies. The significance of a multidisciplinary approach involving ergonomics, linguistics, and machine learning is emphasized to foster the development of text entry systems that are both user-friendly and highly functional.

In summary, while significant advancements have been made in VR text entry technologies, there remains a substantial scope for research, particularly in enhancing the user experience and addressing the limitations of current systems. Future research directions include the exploration of adaptive interfaces and multimodal feedback systems that cater to the

ergonomic needs of users and enhance their overall interaction with the virtual environment.

3 Haptics Implementation

Building on the fundamental insights provided by "XR Haptics: Implementation and Design Guidelines," this study aims to thoroughly integrate and expand upon the principles of haptic feedback in VR environments. By bridging the gap between theory and practice, we seek to optimize user interaction through sophisticated haptic designs that engage users more deeply by mirroring real-world sensations during typing on the keyboard.

3.1 Role of Haptics

The role of haptics in VR environments is multifaceted, extending beyond simple replication of real-world dynamics to enhancing the virtual experience through strategic feedback modifications. Haptics play a crucial role in reinforcing user actions, providing essential cues that transform the virtual experience from a visual-auditory one into a fully immersive sensory encounter.

3.1.1 Realism

According to "XR Haptics: Implementation Design Guidelines," effective haptic design must prioritize realism to ensure that virtual objects behave as users expect based on their real-world experiences. For the virtual keyboard, this involves simulating not only the typical mechanical response of keys but also incorporating variations such as the tactile feel of different materials or the response from a key with wear over time. Implementing nuanced haptic textures and resistance changes can significantly enhance the perception of realism, making the virtual environment more believable and engaging.

3.1.2 Enhanced Immersion through Adaptive Feedback

Adaptive haptic feedback, which adjusts in real-time according to user interactions and environmental conditions, is essential for maintaining immersion. This adaptive approach allows the haptic system to introduce feedback that complements changes in the visual and auditory narrative, thus maintaining a consistent and unified user experience. For example, altering haptic feedback based on the virtual environment's ambient conditions—such as changing surface textures or interacting with moving objects—can make the virtual world feel alive and responsive.

3.1.3 Skill Transfer and Advanced User Experience

Haptic feedback is instrumental in skill transfer from the real world to the VR environment. By accurately mimicking the tactile feedback associated with different tasks, users can apply their real-world skills within virtual settings without the typical learning curve associated with new interfaces. For instance, a virtual piano keyboard that replicates the haptic feedback of actual piano keys can aid musicians in performing without looking at the keys, transferring their skills directly into the VR domain.

3.1.4 Consistency and Customization

Consistency in haptic feedback helps in reducing the cognitive load on users as they navigate through various virtual scenarios. However, customization plays a pivotal role in catering to diverse user preferences and needs, as highlighted in the "XR Haptics: Implementation and Design Guidelines." By allowing users to adjust the intensity, type, and duration of haptic feedback, the system can accommodate personal sensitivity differences, enhancing comfort and usability.

3.2 Haptic Feedback

After establishing the role of haptic feedback in enhancing virtual interactions, it is crucial to identify the specific types of haptics required for the experiment. As introduced earlier, there are two main categories of haptic feedback: kinesthetic and tactile.

3.2.1 Kinesthetic and Tactile Feedback

Kinesthetic feedback, often referred to as FFB, imparts sensations of force and resistance. This type of feedback divides into two categories: passive and active (or resistive) force feedback. Passive feedback typically functions as a brake, limiting finger or body motion, while active feedback applies direct forces to the user, enhancing the realism of interactions such as lifting or pushing objects within a VR environment [26].

Conversely, tactile feedback involves sensations directly related to touch, such as vibrations similar to those experienced on a smartphone or contact-spatial feedback, which provides sensations through surface contact without physical interaction. In VR environments, tactile feedback is crucial for simulating the realistic click of a keyboard button or the texture of different surfaces.

3.2.2 Application in Virtual Keyboards

In the context of a virtual keyboard, both kinesthetic and tactile feedback play pivotal roles. For tactile interactions, simple non-spatial feedback is often sufficient due to the uniform texture of keyboard keys. However, vibrotactile feedback is essential for delivering nuanced sensations that enhance the user's engagement and perception of reality. For actions that involve more dynamic interaction, such as lifting a virtual keyboard, resistive kinesthetic feedback is utilized to simulate the force and weight of the object.

3.2.3 Referential Haptic Feedback

Building on the concept of Referential Haptic Feedback introduced in Section 2.6, our design phase aims to seamlessly integrate this advanced haptic technology into our virtual hand model. This simulation will focus primarily on the user's hand during the typing process, diverging from traditional methods that simulate multiple aspects simultaneously. By concentrating solely on the hand, we strive to create a more realistic and immersive typing experience, enabling users to interact with the virtual keyboard swiftly and intuitively.

3.2.4 Pseudo-Haptics Feedback

Pseudo-haptics involves the integration of visual effects to simulate physical interactions. In our VR environment, dynamic changes occur in the shape and color of the keyboard's keys during typing, accompanied by visual alterations in the keyboard handlers when grasped by the user. Additionally, changes in fingertip coloration help prevent visual penetration, enhancing the realism of interactions.

3.2.5 Allocentric and Egocentric Haptics

The haptic feedback in our VR environment operates within a three-dimensional framework and is categorized into two distinct types: Allocentric and Egocentric. Allocentric haptics depict spatial relationships among objects within the environment, enhancing the user's external spatial awareness. Egocentric haptics, conversely, illustrate the spatial relationships between the user and immediate objects, such as the virtual keyboard, ensuring that feedback is directly related to the user's interactions.

3.2.6 Multimodal Integration

The integration of multiple sensory modalities is crucial for creating compelling and immersive simulations. Our approach uses the "reinforce" method, where haptic feedback is synchronized with auditory and visual elements to enrich the overall experience. This multimodal integration reinforces the tactile sensations with corresponding audio-visual cues,

enhancing the user's engagement and satisfaction while interacting with the virtual keyboard.

3.3 Summary

This chapter on Haptics Implementation has focused on enhancing the realism and immersion of VR environments through sophisticated haptic feedback. Building on principles from the "XR Haptics: Implementation and Design Guidelines," we discussed the role of haptics in creating a fully immersive sensory experience, emphasizing realism, adaptive feedback, and user customization.

Specific types of haptic feedback, including kinesthetic and tactile feedback, were identified as critical for simulating realistic interactions within VR environments, particularly in applications like virtual keyboards. The chapter also introduced innovative haptic technologies such as Referential and Pseudo-Haptics Feedback, and discussed the importance of multimodal integration in reinforcing user engagement and satisfaction. Overall, the chapter aims to bridge the gap between theoretical insights and practical applications in haptic design to optimize user interaction in virtual settings.

4 Hardware and Software Selection

Building upon insights from the previous chapters and the integral role of haptic feedback in enhancing user experience, this section delineates the specific hardware components and software solutions capable of delivering sophisticated haptic feedback for keyboard typing in virtual environments.

4.1 Head-mounted Devices

In our investigation, the Meta Quest 3 is selected as the primary head-mounted display (HMD) due to its superior features when compared to its predecessors and competitors. This decision follows a comprehensive analysis of its capabilities in contrast to other devices such as the HTC Vive, Oculus Rift, Valve Index, PlayStation VR2, Pico 4, HTC Vive Focus 3, HTC Elite XR, Varjo XR-3, Samsung Odyssey+, and HP Reverb G2.

Device	**Display**	**Resolution (per eye)**	**Refresh Rate**	**Tracking**
Meta Quest 3	LCD	2064x2208	120Hz	6DoF inside-out (4 cameras, depth sensor)
HTC Vive	OLED	1080x1200	90Hz	6DoF outside-in (base stations)
Oculus Rift	OLED	1080x1200	90Hz	6DoF outside-in (sensors)
Valve Index	LCD	1440x1600	Up to 144Hz	6DoF outside-in (2 base stations)
PlayStation VR2	OLED	2000x2040	120Hz	6DoF inside-out (4 cameras)
Pico 4	LCD	2160x2160	90Hz	6DoF inside-out (4 cameras)
HTC Vive Focus 3	LCD	2448x2448	90Hz	6DoF inside-out (4 cameras)
HTC Elite XR	OLED	1600x1600	90Hz	6DoF inside-out
Varjo XR-3	Micro-OLED	1920x1920 (focus area) 2880x2720 (peripheral area)	90Hz	6DoF inside-out (integrated cameras)
Samsung Odyssey+	AMOLED	1440x1600	90Hz	6DoF inside-out
HP Reverb G2	LCD	2160x2160	90Hz	6DoF inside-out (4 cameras)

Table 4.1: Comparison of Head-mounted Devices

Device	**Controllers**	**Connectivity**	**Additional Features**
Meta Quest 3	2 6DoF Touch Plus	WiFi 6E, Bluetooth, USB-C	Hand tracking, WiFi streaming [48]
HTC Vive	2 6DoF Vive	USB, HDMI	Expandable with wireless adapter [32]
Oculus Rift	2 6DoF Touch (1st gen)	USB, HDMI	Oculus Store integration [56]
Valve Index	2 advanced 6DoF (finger tracking)	DisplayPort 1.2, USB 3.0	Adjustable IPD, Comfortable design [80]
PlayStation VR2	2 6DoF PS VR2 Sense	USB-C to PS5	Adaptive triggers, haptic feedback [72]
Pico 4	2 6DoF (capacitive touch)	WiFi 6, Bluetooth, USB-C	Lightweight design [58]
HTC Vive Focus 3	2 6DoF	WiFi 6, Bluetooth	Business applications [31]
HTC Elite XR	Advanced (haptic feedback)	WiFi 6, Bluetooth	High comfort, VR and AR capabilities [30]
Varjo XR-3	Professional-grade	Ethernet, USB-C, DisplayPort	High-fidelity focus area, Mixed Reality [81]
Samsung Odyssey+	2 6DoF	WiFi, Bluetooth	Anti-SDE (Screen Door Effect) technology [61]
HP Reverb G2	2 6DoF	WiFi, Bluetooth	High resolution, Adjustable lenses [29]

Table 4.2: Additional Features of Head-mounted Devices

Given these specifications, Meta Quest 3 is chosen for its cutting-edge display quality, superior tracking, and enhanced connectivity, making it the most suitable option for our experiments.

4.2 Haptic Sensations and Devices

The role of haptic devices in virtual reality is to bridge the gap between the real and virtual worlds by transmitting tactile sensations. A company named SynTouch has developed a product known as the SynTouch-Toccare Haptics Measurement System, capable of quantifying 15 haptic dimensions of any material, thus replicating human touch perception. This technology is pivotal in providing a nuanced understanding of material textures and interactions within the virtual environment.

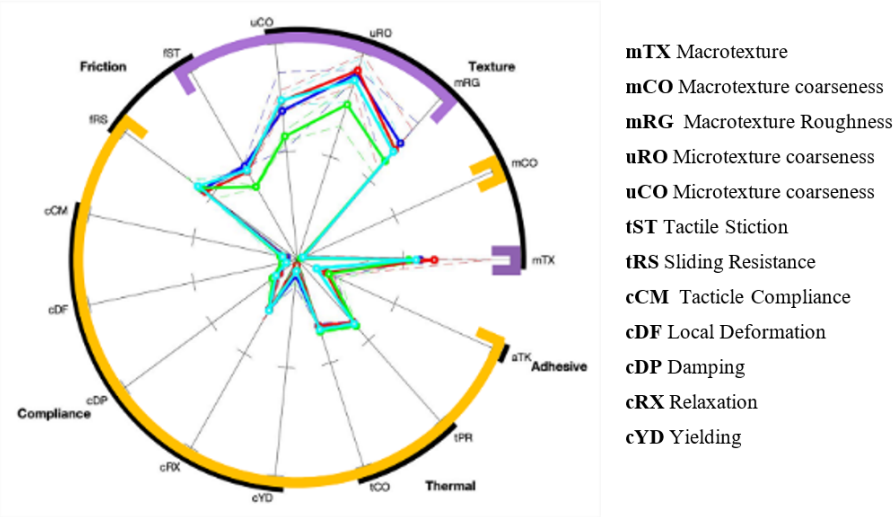


Figure 4.1: SynTouch's 15-Dimensional Sensations [13]

4.2.1 Advanced Haptic Gloves

For the experimental setup, we propose using the SenseGlove Nova, renowned for its advanced haptic feedback capabilities and innovative design. The SenseGlove Nova offers a range of features that enhance the realism and immersion of virtual reality interactions:

- **Force Feedback:** Each glove is equipped with high-precision brakes that deliver up to 20N of force on each finger. This feature enables users to feel the weight and resistance of virtual objects, ranging from heavy tools to delicate items, providing a more authentic interaction experience [1].
- **Vibrotactile Feedback:** The SenseGlove Nova incorporates vibrotactile actuators that generate realistic sensations such as button clicks, vibrations, and impacts. This

Figure 4.2: The Sense Glove Nova [1]

feedback significantly enhances the user's sense of touch and engagement with virtual environments [77].

- **Sensor-Based Tracking:** The gloves utilize multiple high-accuracy sensors to capture the movements of the thumb and fingers. This precise tracking is crucial for accurate hand representation in virtual reality, ensuring that user actions are mirrored accurately within the virtual space [40].

The SenseGlove Nova's combination of force feedback, vibrotactile feedback, and precise tracking makes it an ideal choice for applications requiring detailed and nuanced hand interactions. Its advanced technology supports a wide range of virtual reality applications, from training simulations to interactive gaming, by providing users with a tactile experience that closely mimics real-world interactions.

The use of SenseGlove Nova in our experimental setup is expected to significantly enhance the user experience by providing realistic and responsive haptic feedback. This technology not only improves the immersion but also aids in tasks requiring fine motor skills and detailed manipulation of virtual objects, making it a valuable tool for our virtual reality system [69].

4.3 Virtual Environment Development with Unity

Unity will be utilized for developing the virtual environment in which the virtual keyboard will be used. The advantages of using Unity include:

- **Real-Time Rendering:** Unity's powerful real-time rendering capabilities ensure smooth and immersive visuals, essential for creating a realistic virtual environment.
- **Cross-Platform Support:** Unity supports multiple platforms, allowing the virtual environment to be accessible on various devices, from VR headsets to mobile devices and PCs.

- **Extensive Asset Store:** Unity's Asset Store provides a vast array of pre-made assets, scripts, and plugins that can accelerate development and add rich features to the environment.
- **Robust Developer Community:** A large and active community of developers means extensive support, tutorials, and forums for problem-solving and learning.
- **Seamless Integration with Other Tools:** Unity can easily integrate with other development tools and software, facilitating a cohesive and efficient development process.

Using Unity ensures that our virtual environment will be engaging, versatile, and accessible, providing users with a seamless and immersive experience while interacting with the virtual keyboard.

4.4 Keyboard Design with Blender

For the design and modeling of the virtual keyboard, we will use Blender, a powerful and versatile 3D modeling software. Blender is chosen for several reasons:

- **Comprehensive Toolset:** Blender offers an extensive range of tools for modeling, sculpting, texturing, and rendering, making it an all-in-one solution for creating detailed and high-quality 3D models.
- **Open Source and Free:** Blender is open-source software, which means it is free to use. This allows us to allocate resources to other aspects of the project while still utilizing professional-grade software.
- **Community Support and Documentation:** Blender has a large, active community and extensive documentation, providing ample resources for troubleshooting and learning advanced techniques.
- **Integration Capabilities:** Blender supports various file formats and can be integrated with other software tools used in our project, ensuring a smooth workflow.

Using Blender allows us to create a highly detailed and customizable virtual keyboard, enhancing the user's interaction experience by providing a visually appealing and functional design.

4.5 Integration With Meta Quest 3

The integration of SenseGlove Nova with the Meta Quest 3 is facilitated through the SenseGlove Nova SDK [64] . Connections can be established via Bluetooth or by directly linking

the glove's controller transform object into the virtual environment setup. This integration allows for precise control over force feedback and tactile sensations, ensuring a rich and immersive user experience.

4.6 Assisting and Guiding

In addition to enhancing the surrounding area of the virtual keyboard to assist users during typing by providing directions and guidelines, we will integrate advanced language understanding features through a third-party service called ChatGPT. This integration aims to improve the structure and content of entered text by offering sentence and word suggestions [57, 34]. Users will be able to select the most suitable suggestion for their attention, thus elevating the overall typing experience. ChatGPT, trained on a diverse range of internet text, possesses advanced language understanding capabilities, enabling it to generate human-like text based on provided context [57]. This unique attribute makes it an ideal tool for suggesting contextually relevant and grammatically correct words or sentences, surpassing the capabilities of simple auto-correct systems. Unlike traditional auto-correct systems that focus solely on the last word typed, ChatGPT provides contextual suggestions by considering the overall context of the sentence [78, 79]. This approach results in more accurate and helpful suggestions, contributing to an improved user experience.

The integration of ChatGPT goes beyond correcting typos; it enriches the typing experience by allowing users to explore different ways to express their thoughts with the assistance of versatile suggestions. This not only enhances efficiency but also adds an element of enjoyment to the typing process [34]. Furthermore, ChatGPT's continuous learning capability as a machine learning model means that it can improve over time. The more it is utilized, the better it becomes at providing relevant suggestions, ensuring a dynamic and evolving tool for users. In terms of versatility, ChatGPT can be seamlessly adapted to various applications, making it a valuable tool in diverse scenarios beyond typing assistance. Its versatility extends its utility across different domains, showcasing its potential as a multifaceted solution [57].

4.7 Summary

Selecting appropriate hardware and software is crucial for achieving a high-fidelity virtual experience. The chosen devices support advanced delivery of haptic feedback and ensure a seamless, intuitive user experience. By integrating these technologies, we aim to create a virtual typing interface that closely mimics real-world interactions, thereby enhancing realism and effectiveness in the virtual environment.

5 Conception

Building upon the insights from previous chapters regarding the pivotal role of haptic feedback, this chapter transitions from theoretical discussions to the practical implementation phase of our study. We have explored the rationale for incorporating haptic feedback, its intended applications, and the specific tools planned for use. Now, we move forward to actualize these concepts, beginning with a general use case diagram to visually represent system interactions. Following this, we will examine the visual aspects of our 3D interface and conclude by elucidating the underlying mechanisms that support our proposed solutions. This comprehensive approach ensures that our transition into the implementation phase is well-founded, aiming to enhance user experience through detailed planning and thoughtful design.

5.1 General Use Case

This section introduces the General Use Case diagram, which serves as a crucial visual tool depicting the dynamic interactions between users and the core functionalities planned for our VR environment experiment. This diagram provides a comprehensive overview of the various scenarios and interactions that users may encounter, elucidating how they engage with the key features incorporated into the system.

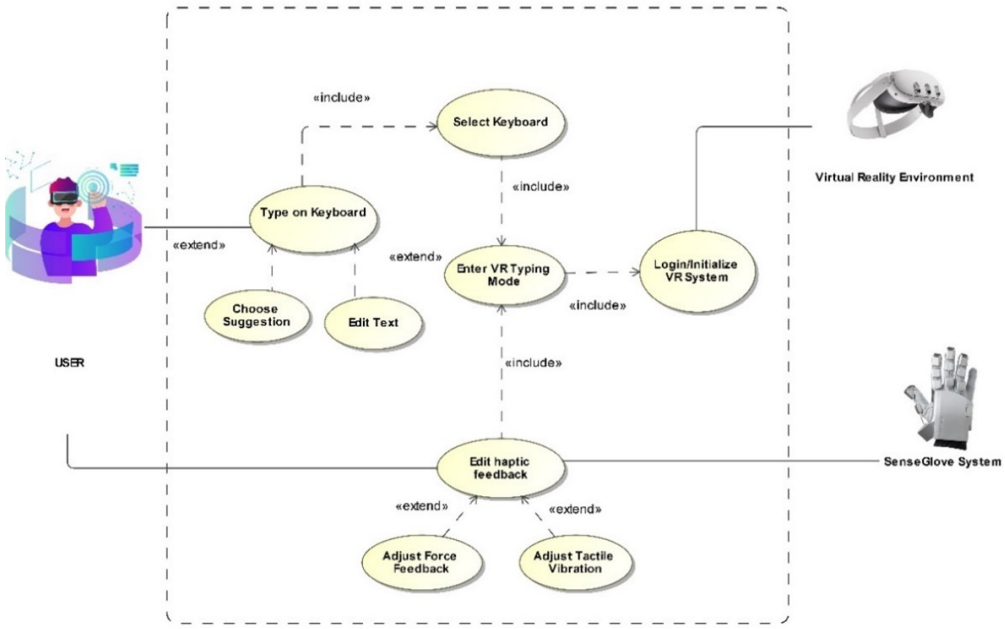


Figure 5.1: General Use Case Diagram for our experiment

Accompanying the diagram, Table 5.1 details the specific use cases, providing a structured description of each scenario within the VR environment. This table categorizes user interactions, from initiation through various interaction phases, to modifications in settings, highlighting the system's response to each action.

Use Case	**Description**
Login/Initialize VR System	To initiate the activity, the user is required to verify their identity, ensuring secure access to the virtual environment.
Enter VR Typing Mode	Following successful login, the user transitions seamlessly into the immersive virtual typing environment.
Select Keyboard	The user selects the keyboard layout and settings to start typing, choosing from various available configurations.
Type on Keyboard	Interaction begins as the user types on the virtual keyboard, engaging with digital content creation.
Choose Suggestions	Users can select from contextually relevant suggestions provided by the system while typing, enhancing typing efficiency and accuracy.
Edit Text	Users have the flexibility to edit text at any point, with options to add, delete, or rearrange content as needed.
Edit Haptic Feedback	Users can enter a mode to customize the haptic feedback settings according to their personal preference.
Adjust FFB	Users can fine-tune the intensity of FFB to match their desired level of tactile response.
Adjust Tactile Vibration	The intensity of tactile vibrations can be adjusted, allowing users to tailor the haptic experience to their comfort and liking.

Table 5.1: Descriptions of General Use Cases

Each use case outlines critical interactions within the system, ensuring that user needs and preferences are meticulously catered to, from system initiation to detailed personalization of the typing experience. This structured approach not only enhances user satisfaction but also fosters an intuitive and engaging interaction environment.

5.2 Network Diagram

Following the identification of the primary use cases, we now focus on examining the intercommunication among the key hardware components of our VR environment setup. The network diagram presented in Figure 5.2 offers a detailed visual representation of the interaction among these components, which include advanced haptic gloves, the Execution Environment, the local subnet, and the HMD. This diagram is essential for understanding how data flows within our system and how components are interconnected to support seamless user interactions.

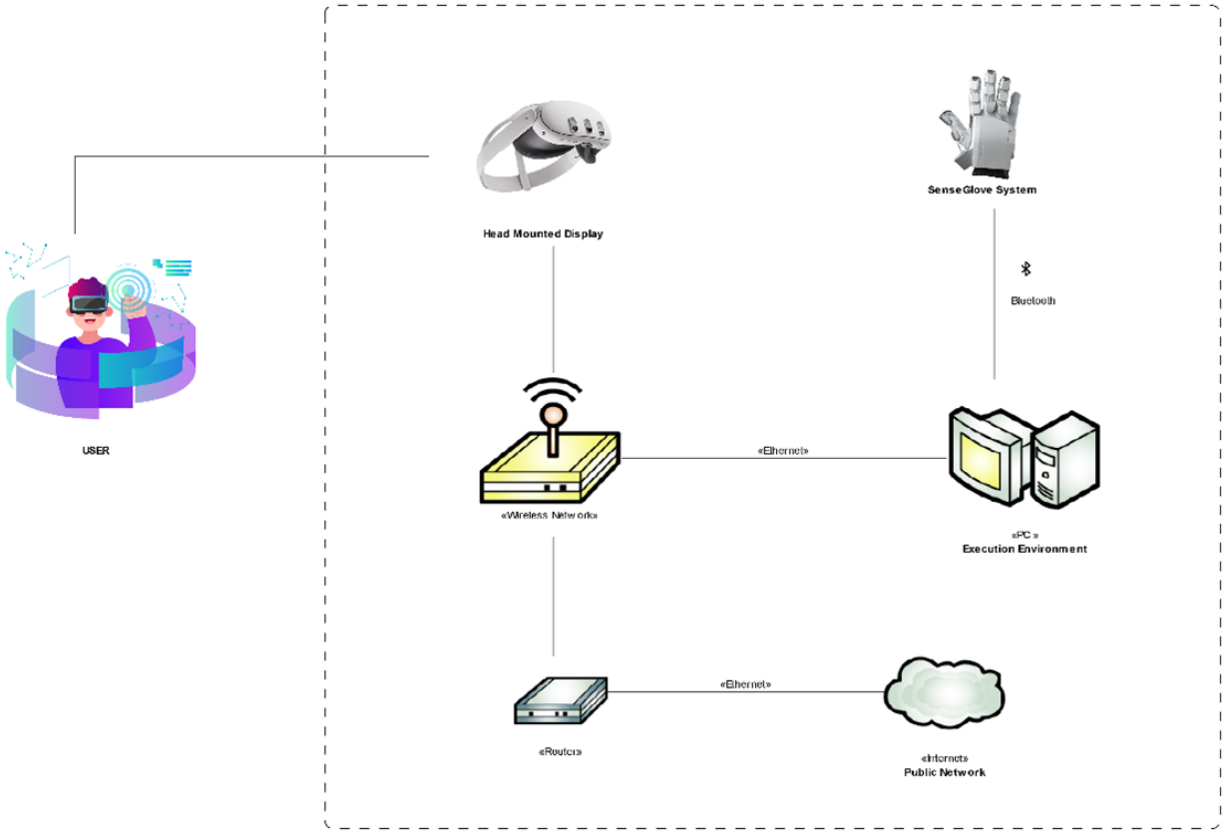


Figure 5.2: Network Diagram showing the interconnectivity of system components.

Hardware	Description
SenseGlove Nova	The SenseGlove Nova transmits sensory data to the Execution Environment via Bluetooth. It can also establish a direct connection to the HMD when required.
Wireless Network	Communication between the HMD and the Execution Environment is facilitated through Air Link, a high-performance wireless connection that supports real-time data exchange.
Execution Environment	All scenario executions occur locally within this environment, employing Bluetooth as the primary communication protocol to ensure rapid and secure data transfer.
Router	The router plays a pivotal role in managing communications between the HMD, Execution Environment, and the public internet, ensuring robust and reliable connectivity.
Head-Mounted Device (HMD)	The HMD communicates with the Execution Environment using the TCP protocol to facilitate seamless access and synchronization with the public internet.

Table 5.2: Descriptions of the Network Diagram Components

This network setup is designed to optimize the flow of information and control signals across different components of the system, ensuring that the VR environment operates smoothly and efficiently. The integration of these components via robust communication protocols and network connections is critical for delivering a responsive and immersive user experience, thereby enhancing both the realism and functionality of the virtual environment.

5.3 Virtual Keyboard's Conception

While smartphones have become the most used hardware in the market, surpassing other devices [39], they still rely on fundamental components common to many hardware devices. For instance, consider the conventional structure of a keyboard, which is an integral part of many devices, not just computers but also smartphones in the form of digital keyboards.

The keyboard comprises two primary components: the keys and the substrate. The substrate is a planar surface responsible for housing and supporting the keys. In the context of a smartphone, the 'keys' are the touch-responsive elements displayed on the screen, and the 'substrate' is the underlying software framework that captures and interprets key presses [74]. Given user familiarity with the established paradigm of smartphone keyboard architecture, the design of the keyboard will conform to this norm. Consequently, emphasis will primarily

be directed toward the clarification and optimization of the aforementioned two components.

Hence, our goal is to replicate the smartphone keyboard in a VR environment. The following sections delve into the intricacies of the keyboard design slated for use in the forthcoming immersive experience.

5.3.1 Conceptualization of the Board

To make a virtual keyboard look and act real, attention to detail is crucial. By replicating as many features as possible, the virtual version can seem more genuine. A study [86] investigated the impact of QWERTY and T9 [2] input methods on smartphone text input across various sizes (4.7 inches, 5 inches, and 5.5 inches). Findings indicate that QWERTY is more effective for two-thumb input, and T9 for one-thumb input, with two-thumb entry providing an overall superior user experience. Optimal performance was observed with QWERTY and T9 on a 5-inch smartphone and QWERTY on a 5.5-inch smartphone.

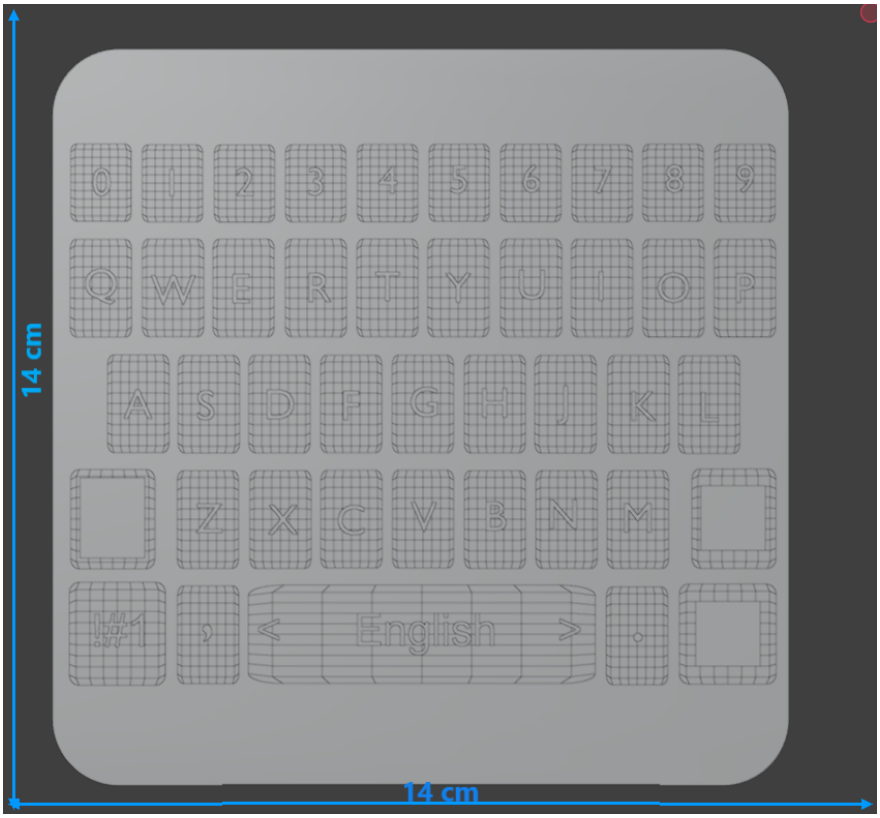


Figure 5.3: Schematic representation of the virtual keyboard layout used for testing QWERTY input methods, displayed in Blender viewport shading (Solid Display).

In addition to keyboard size, the most commonly used colors for smartphone keyboards are black, gray, and white. This is supported by the popularity of black as a phone color and the prevalence of these colors in top Android keyboards like Fleksy, Gboard, and SwiftKey [51]. These colors are likely chosen for their neutrality, which allows them to blend well with

various phone themes and user preferences.

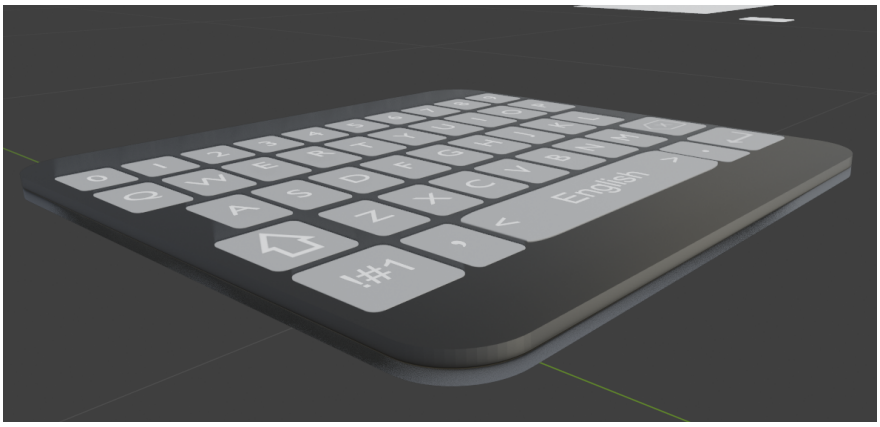


Figure 5.4: Schematic representation of the virtual keyboard layout used for testing QWERTY input methods, displayed in Blender viewport shading (Material Display).

To enhance the portability of our keyboard, integrating hand support is imperative, aiming to emulate the familiar grip of a smartphone. This augmentation contributes to a heightened sense of realism and immersion in VR experiences, allowing users to interact with the keyboard in a manner akin to physical settings, even while wearing gloves. Moreover, ensuring universal comfort necessitates careful consideration of users' hand dimensions. According to the National Aeronautics and Space Administration (NASA), the average hand length for males is 7.6 inches, with a breadth of 3.5 inches and a circumference of 8.6 inches. Correspondingly, for females, these dimensions are 6.8 inches, 3.1 inches, and 7.0 inches, respectively. Incorporating this data allows us to ensure that the thumb can comfortably reach the edge of the keyboard when the user enters the typing mode, ensuring a user-friendly and universally comfortable device [52, 54, 53].

5.3.2 Key Conception for the Keyboard

In our pursuit to optimize the VR keyboard and replicate the efficiency of physical typing, we propose a distinctive proportion mechanism aimed at minimizing typing errors. This mechanism operates by identifying the key with the largest contact area with the virtual finger. To streamline this process, we advocate for the division of each virtual key into multiple sections.

To implement this division, we suggest employing a grid-based method. Each virtual key's surface is subdivided into cells of equal size, forming a grid. The dimensions of this grid are determined by the average size of the virtual fingertip and the size of the virtual key. For instance, if both the virtual key and the average virtual fingertip measure 10 mm x 10 mm, we could opt for a 3x3 or 8x10 grid configuration.

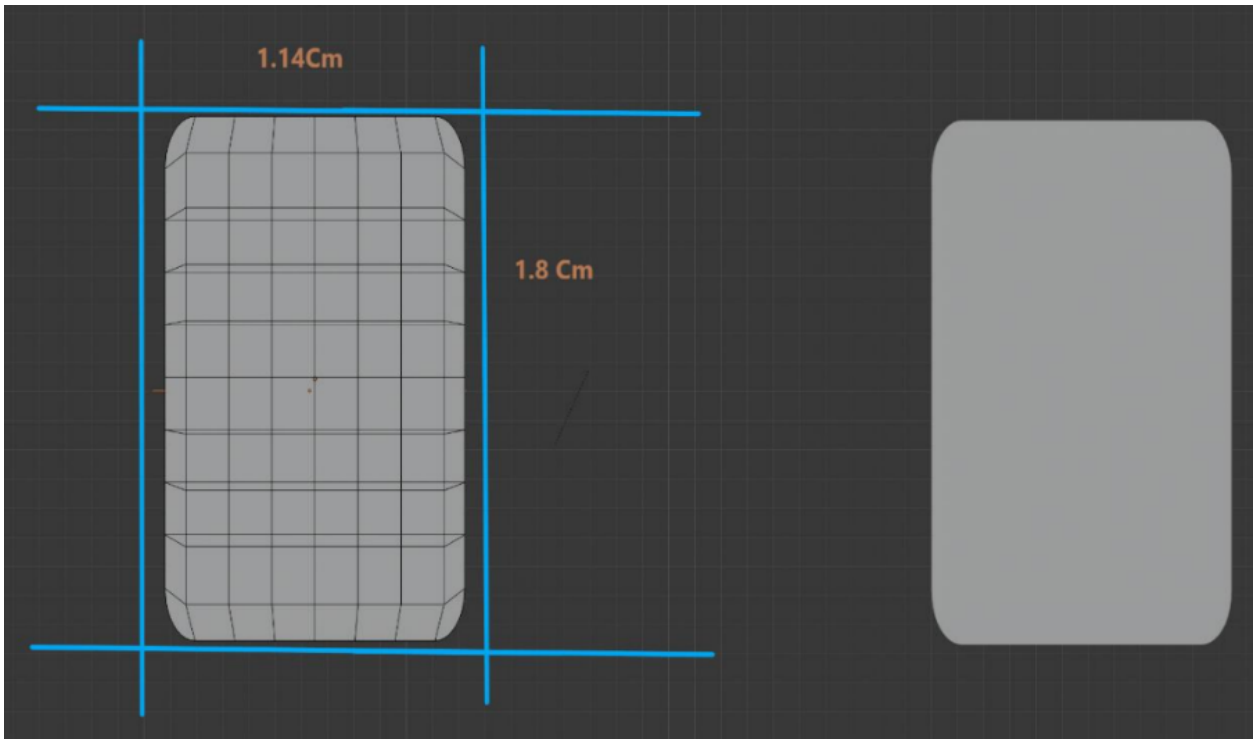


Figure 5.5: Schematic representation of the virtual key layout used for testing QW- ERTY input methods, displayed in Blender viewport shading (Solid Display)

The subsequent step involves scrutinizing each cell within the grid to assess the degree of contact with the virtual finger. This assessment is facilitated by incorporating a texture on the surface of the thumb finger of the virtual hand model, as provided by the SenseGlove Nova SDK.

To enhance the user experience, we propose integrating an animation feature. The shape of the virtual key during interaction will mimic the shape it takes when pressed on a smartphone. A pop-up animation will accompany this interaction, visually indicating the pressed key. When designing virtual keyboards, particularly the size of the keys, it's essential to balance ergonomics with functionality to enhance user experience and performance. Research indicates that key size can significantly affect typing speed, accuracy, and overall comfort. According to a study [37], key sizes on touch screen virtual keyboards influence productivity, usability, and typing biomechanics, suggesting that virtual keyboards with a key size less than 16 mm may slow typing speed and increase wrist extension, highlighting the importance of optimizing key size for user comfort and efficiency. Another study [4] proved that virtual keyboards with key sizes smaller than recommended (18 to 20 mm for notebook and desktop keyboards) can lead to slower typing speeds, higher muscle activity, and greater wrist extension, indicating potential discomfort and reduced productivity. Keyboards with 13x13 mm

keys, for example, resulted in a 15% slower typing speed and higher static shoulder muscle activity compared to larger key sizes.

For a VR smartphone keyboard, choosing the right texture for the keys is essential to enhance realism and user experience. The texture should provide visual cues that simulate the feel of a real keyboard. Research in human-computer interaction (HCI) [3] has shown that reducing glare and reflections on surfaces can improve visibility and reduce eye strain. A Dark texture on the VR keyboard keys can help achieve these goals, making it easier for users to see and interact with the keyboard.

5.4 Dynamic Virtual keyboard Interactions

5.4.1 Key Press Animation Feedback

For key press animation, the goal is to provide immediate feedback to the user, indicating that their input has been recognized. This type of animation falls under the category of micro-interactions. These animations are crucial in making interactions feel more tangible and responsive [25]. Therefore, a 0.2-second animation at 30 frames per second, resulting in 6 frames, would be consistent with these principles, offering a quick and responsive feedback loop that enhances the user experience without overwhelming the user with unnecessary motion or delay.

In our project, the scale values were adjusted during the animation. The X and Y components were each increased by approximately 11%, while the Z component was increased by approximately 4%. These adjustments were calculated to provide a noticeable yet subtle feedback effect, aligning with the principles of effective micro-interactions and ensuring a smooth, responsive user experience. We found that this scale percentage is the optimal size to avoid key collision during scaling, ensuring that the keys remain distinct and accessible during the animation.

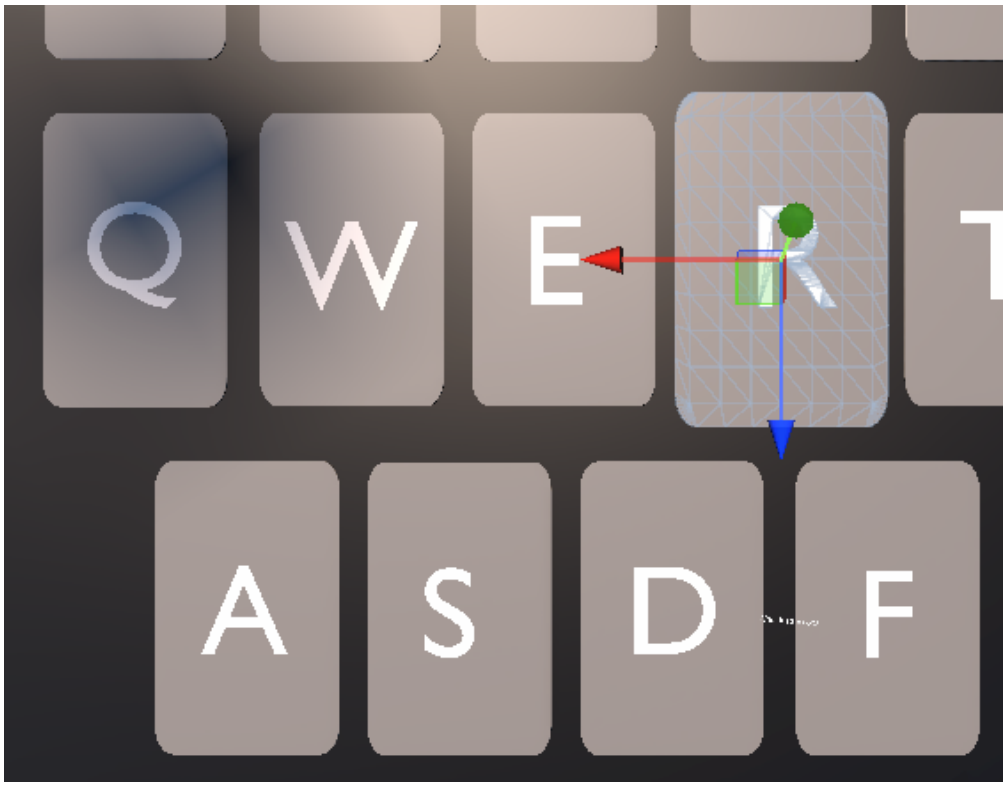


Figure 5.6: Key Appearance After Animation

5.4.2 Hover Effect Implementation

In virtual reality (VR) environments, providing immediate visual feedback to users significantly enhances their interaction experience, improving both accuracy and comfort. One critical aspect of this feedback mechanism is the hover effect, which visually highlights key parts as users move their fingers over them. This section explains the logic behind the hover effect implementation in the VR keyboard, including the dynamic color changes of the key parts.

Detection of Hovered Key Parts

The hover effect is implemented by continuously monitoring the position of the user's thumb in the VR environment. A virtual bounding box is defined around the thumb, and this box is used to detect any interaction with the virtual keyboard key parts.

Raycasting Technique

To determine which key parts are being hovered over, a technique called raycasting is used. This involves projecting a virtual box around the thumb's position and checking for intersections with the key parts on the virtual keyboard. The system identifies which key parts are within this box, indicating that they are being hovered over by the user's thumb.

Visual Feedback Mechanism

Once a key part is detected as being hovered over, the system provides immediate visual feedback by changing the appearance of the key part. This visual change primarily involves altering the color of the key part. The color transition is determined based on a predefined start color and end color, and the specific weight assigned to each key part. The weight represents the degree of interaction or importance of the key part, allowing for a smooth gradient transition between the start and end colors [19] .

Handling Multiple Key Parts

The system keeps track of the key parts that were previously hovered over. If the thumb moves away from a key part, the visual feedback for that key part is reset to its original state. This ensures that only the currently hovered key parts are visually highlighted, providing real-time feedback as the user moves their thumb across the virtual keyboard.

Interaction States

The hover effect is closely integrated with the overall interaction states of the virtual keyboard. When no key part is actively being pressed, the system focuses on detecting and highlighting the key parts being hovered over. If a key part is pressed, the hover detection is temporarily paused to prevent conflicts between hover and press feedback.

Dynamic Color Adjustment

The system dynamically adjusts the color of the key parts based on the user's actions. When a key part is hovered over, its color changes according to a calculated weight. The weight determines the blend between the start and end colors, using a linear interpolation technique to create a smooth transition. This provides immediate and clear visual feedback, enhancing the user's interaction experience. When the thumb moves away, the key part's color is reset to its original state, ensuring that the visual feedback accurately represents the current interaction state [11].

User Experience Enhancement

By providing immediate and clear visual feedback through the hover effect, including dynamic color changes, the system enhances the user's typing experience in the VR environment. Users can easily see which key parts they are about to press, reducing errors and increasing typing efficiency. This real-time feedback is crucial for creating a more intuitive and engaging virtual interaction [55].

5.4.3 Key Press Sound Feedback

In our project, we aimed to generate unique audio feedback for each key press on a virtual keyboard. The process involved modifying the pitch of a base click sound to create distinct auditory signals for each key, ensuring that each key press has a unique sound effect. This approach enhances the user experience by providing clear and immediate auditory feedback, aligning with the principles of effective micro-interactions. We utilized the `pydub` library for audio processing [59], which was responsible for loading, modifying, and exporting audio files.

1. Pitch Modification

Pitch modification refers to changing the frequency of an audio signal to alter its pitch without changing its duration [71]. This is a crucial technique in audio processing used to create variations in sound.

The core principle behind generating unique sounds for each key is pitch shifting. Pitch shifting involves changing the frequency of an audio signal without altering its duration. Mathematically, this is achieved by altering the sample rate of the audio signal. The relationship between the original frequency (f) and the modified frequency (f') when shifting the pitch by s semitones is given by:

$$f' = f \cdot 2^{\frac{s}{12}} \tag{5.1}$$

where:

- f is the original frequency,
- f' is the new frequency after pitch shifting,
- s is the number of semitones by which the pitch is shifted.

2. Sample Rate Adjustment

Sample rate adjustment is the process of changing the sampling rate of an audio signal to alter its pitch. By modifying the sample rate, the pitch of the sound can be raised or lowered while keeping the duration constant [36].

To implement this pitch shift, we adjust the sample rate of the audio signal. If the original sample rate is R, the new sample rate R' after shifting the pitch by s semitones can be calculated as:

$$R' = R \cdot 2^{\frac{s}{12}} \tag{5.2}$$

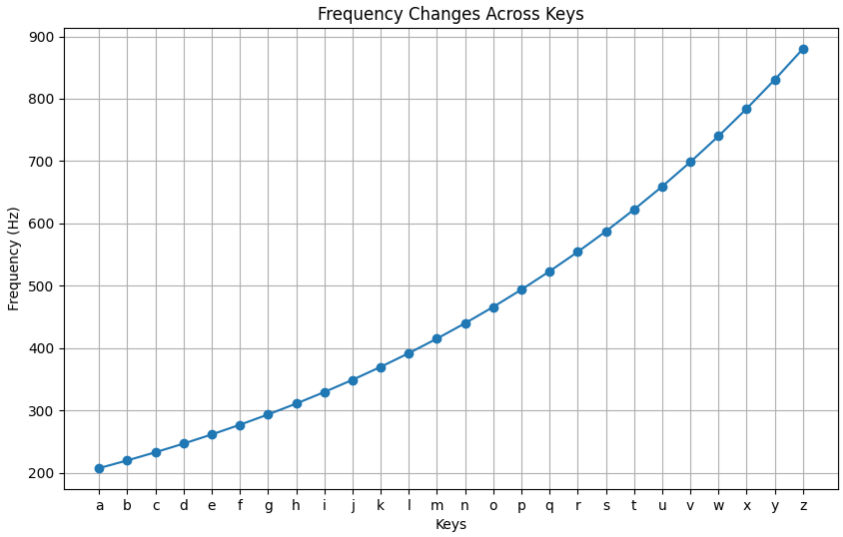


Figure 5.7: Frequency changes across keys from 'a' to 'z'.

By resampling the audio signal to this new sample rate and then adjusting it back to the original sample rate, we effectively change the pitch while maintaining the duration of the audio.

The sample rate adjustment is executed within the `change_pitch` function. Initially, we calculate the new frame rate (R') based on the desired pitch shift in semitones. We use the `_spawn` method from the `pydub` library to create a new `AudioSegment` with the modified frame rate, which changes the pitch of the sound. Finally, we reset the frame rate back to the original to ensure the duration remains unchanged, thereby achieving the desired pitch modification.

3. Application to Key Press Sounds

For our virtual keyboard, we needed a unique pitch for each letter key (from 'a' to 'z'). We assigned a specific number of semitones to each key based on its position in the alphabet. The assignment is defined as follows:

$$s_i = i - 13 \tag{5.3}$$

where:

- s_i is the pitch shift for the i-th letter,
- i is the index of the letter in the alphabet (0 for 'a', 25 for 'z').

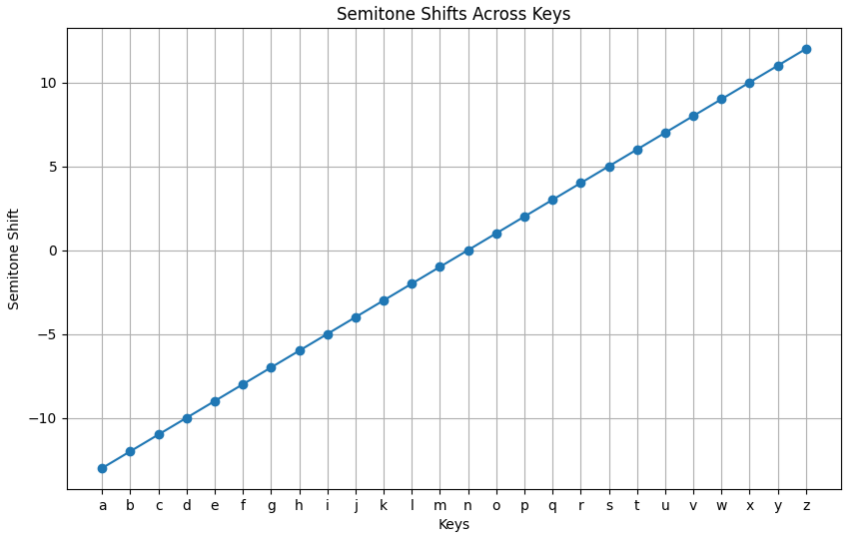


Figure 5.8: Semitone shifts across keys from 'a' to 'z'.

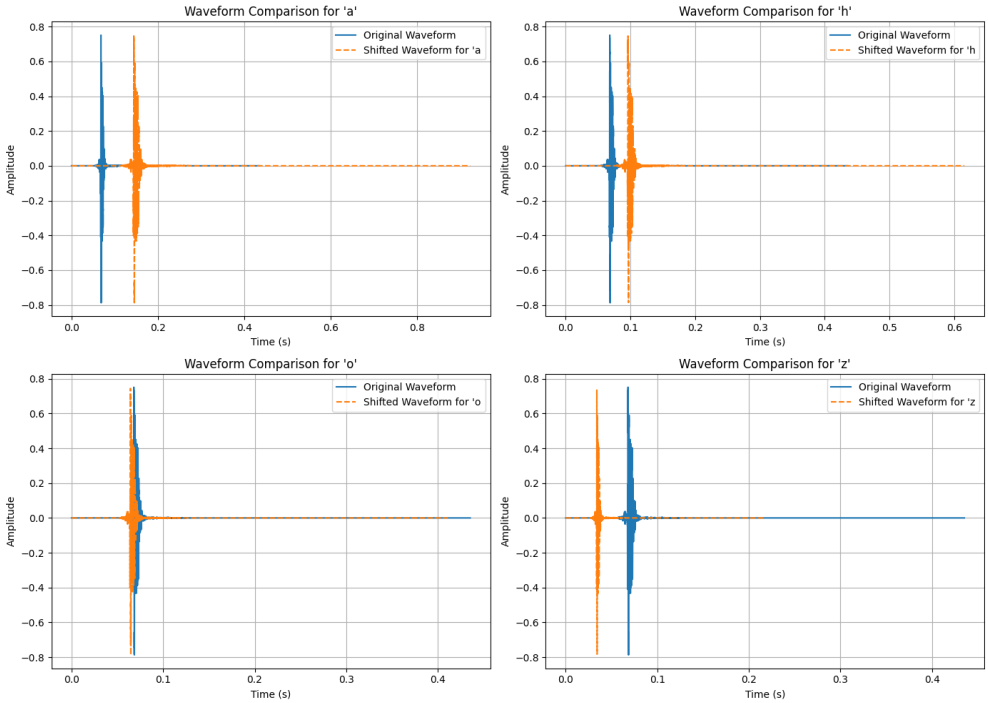


Figure 5.9: Waveform comparison of original and shifted sounds .

This results in a pitch shift range from -13 semitones for 'a' to +12 semitones for 'z'. By applying these mathematical principles of pitch shifting and sample rate adjustment, we generated distinct and responsive audio feedback for each key on a virtual keyboard. This

method enhances the user experience by providing immediate and clear auditory confirmation of each key press, aligning with the principles of effective micro-interactions.

5.4.4 Key Weight Calculation

In the virtual keyboard simulation developed , we quantify the tactile feedback and the accuracy of the desired typed keys through a calculated "weight" assigned to each key and its respective parts. This weight metric plays a crucial role in simulating dynamic responses typical of physical keyboard interactions, which is essential for enhancing the accuracy of typing within the virtual environment. The need for such a simulation arises from the inherent discrepancies in the data frequencies between the frame rate of the virtual environment and the tracking system used for modeling hand movements. By synchronizing these elements through calculated key weights, we ensure that the user's interactions with the virtual keyboard are both intuitive and responsive, closely mimicking real-world typing experiences.

1. Key Part Weight Calculation

The weight of each key part is determined based on its designated role and position within the key layout. For edge key parts, which are less frequently engaged during typing, a constant base weight is assigned, ensuring a uniform weight distribution for less significant key parts and is given by:

$$W_{\text{base}} = 0.1 \quad \text{(as an example value)} \tag{5.4}$$

For central and more interactive key parts, the weight is dynamically calculated based on the key part's Cartesian coordinates on the keyboard grid. This method emphasizes the importance of centrally-located key parts:

$$W(x,y) = \frac{1}{\sqrt{(x - x_{\text{center}})^2 + (y - y_{\text{center}})^2 + 1}} \tag{5.5}$$

where x_{center} and y_{center} represent the coordinates of the central key part of the keyboard, and the addition of 1 in the denominator prevents division by zero, ensuring that central key parts have the highest weights.

2. Overall Key Weight Calculation

The total weight of a key at any moment is the sum of the weights of all its collided key parts. This sum is calculated periodically to determine the overall weight of the key:

$$W_{\text{key}} = \sum_{i=1}^{n} W_i \tag{5.6}$$

where W_i is the weight of the i-th collided key part. This cumulative weight is then used to identify the active key, which is the key with the highest total weight among all keys engaged at that instance.

This approach models real-world typing by dynamically adjusting key sensitivity based on typical finger movements and interactions, thereby simulating an intuitive and realistic typing experience. The mathematical model ensures that the most probable key press is detected based on tactile feedback, which is crucial for providing accurate and responsive keyboard simulations.

3. Process Overview

1. **Collision Detection:** Each key part uses Unity's physics engine to detect collisions.
2. **Event Notification:** Collisions trigger events that notify the parent key of the key part's status.
3. **Register collided key:** The key notifies the keyboard upon activation and registers itself within the keyboard.
4. **Weight Calculation:** The keyboard periodically checks all activated keys, aggregating the weights from their active key parts to calculate their total weights. It then identifies the key with the highest total weight
5. **Key Press Detection:** A key is considered pressed if its aggregated weight is the highest among all keys.
6. **Key Typed:** The determination of a key press is finalized when all collisions have ceased, particularly as the user lifts their finger from the key.

6 Implementation of Typing Test Scenario and Metrics

6.1 Introduction

This chapter details the implementation of a comprehensive typing test scenario designed to evaluate user performance with different virtual keyboard types in a virtual reality (VR) environment. The scenario includes various elements such as user input handling, data collection, phrase management, error tracking, and database integration to provide a robust framework for testing and analysis.

6.2 Typing Test Scenario

6.2.1 Overview of the Typing Test Scenario

The typing test scenario aims to assess the typing performance of users using different keyboard configurations in a VR environment. Specifically, the test compares the MRTK (Mixed Reality Toolkit) Keyboard with a custom virtual keyboard designed for this study. The MRTK Keyboard, part of Microsoft's Mixed Reality Toolkit, provides a standardized virtual input method that includes features such as spatial awareness, robust input handling, and support for hand and eye tracking [49]. It is designed to facilitate ease of use and seamless interaction within virtual reality environments, making it an ideal choice for evaluating virtual text entry.

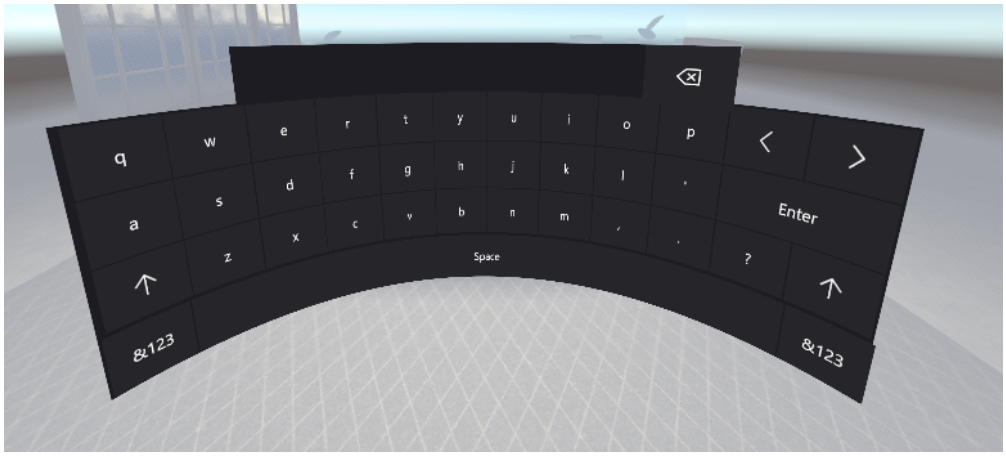


Figure 6.1: MRTK Keyboard interface showing a curved QWERTY layout designed for virtual reality environments inisde unity.

The custom virtual keyboard, on the other hand, was created using Blender 5.3 to incorporate enhanced haptic feedback mechanisms and ergonomic design tailored to improve user interaction and typing efficiency. This custom keyboard aims to address the limitations of existing virtual keyboards by providing a more intuitive and immersive typing experience through advanced haptic feedback and ergonomic considerations.

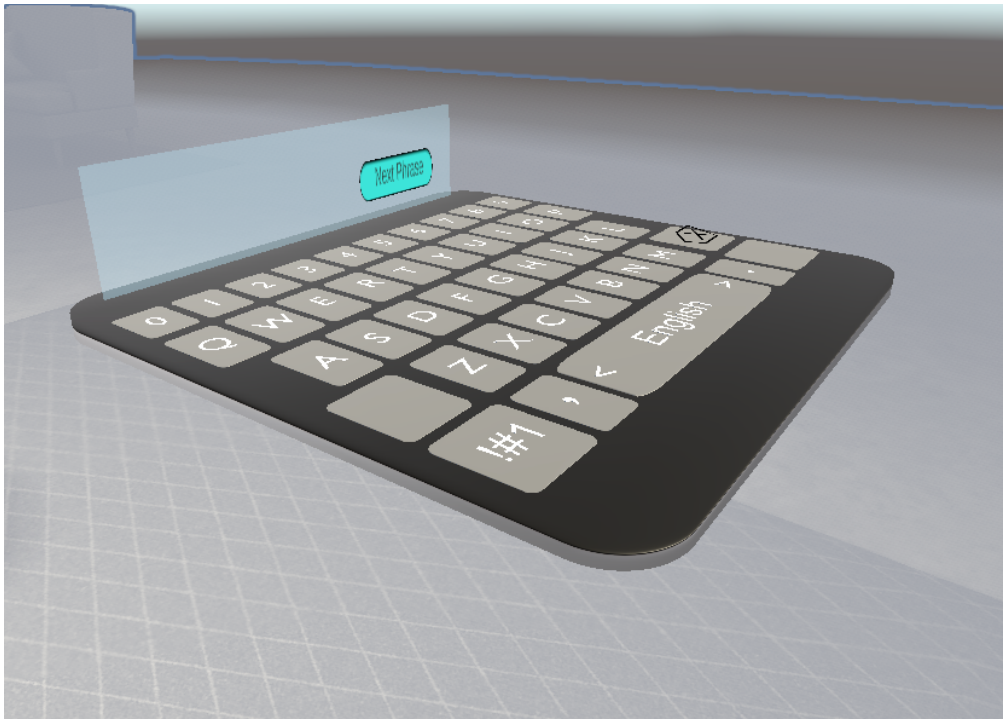


Figure 6.2: Custom virtual keyboard interface designed with enhanced haptic feedback and ergonomic features for improved typing efficiency in VR inside unity.

The system captures detailed user input data, records performance metrics, and manages the flow of the test to ensure a comprehensive evaluation. The phrases used in the test are sourced from Scott MacKenzie's phrase sets, which are widely recognized in the field of text entry research. MacKenzie's phrase sets are designed to provide a standardized basis for evaluating text entry methods, featuring a wide range of commonly used words and phrases that reflect typical typing scenarios. These phrase sets are carefully curated to balance frequency and representativeness, ensuring that the typing tasks are both realistic and challenging [45].

By using these standardized phrases, the study can reliably measure typing performance metrics such as Words Per Minute (WPM) and Error Rate (ER), providing valuable insights into the efficiency and accuracy of each keyboard configuration. The consistent use of MacKenzie's phrase sets allows for comparability with other studies in the field, facilitating a broader understanding of virtual text entry performance.

6.2.2 User Input Handling

The system continuously monitors the user's typing input in a designated text field, tracking each keystroke and identifying any mistakes made during the typing process. This allows for real-time data collection and analysis of user performance.

6.2.3 Data Collection

Detailed data about the typing session is collected, including the time taken, the number of keystrokes, and the accuracy of the typed text compared to the expected text. After the user finishes typing a phrase, the system records this data and saves it inside a third-party database.

6.2.4 Phrase Management

The test involves typing a series of phrases. The system loads a new phrase for the user to type and displays it on the screen. Once the user finishes typing a phrase, the system prepares the next phrase, ensuring a continuous flow of the test.

6.2.5 Error and Accuracy Tracking

The system not only counts the total keystrokes but also keeps track of incorrect keystrokes, helping to measure the user's typing accuracy. It calculates various metrics such as error rate, accuracy in characters, accuracy in words, and typing speed.

6.2.6 Interactive Elements

Interactive elements such as a button allow the user to proceed to the next phrase. This ensures that users have control over the transition between different phrases and keyboard types.

6.2.7 Data Storage

All the typing data collected during the test is stored in a database. This data includes details like the typed text, the expected text, the time taken, and various accuracy metrics. Storing this data allows for detailed post-test analysis.

6.2.8 Configuration and Setup

Before starting the test, the system loads configuration settings such as the types of keyboards to be tested and other parameters that guide the test process. The configuration settings include:

- Names of the keyboards to be tested.
- Number of phrases the user should type in both the training scene and the testing scene.
- Waiting duration between different keyboards of the test.
- Vibration and force feedback settings for the gloves.

6.2.9 Scene Management

The test scenario involves three main scenes: Training, Testing, and Waiting.

- **Training Scene:** The user practices typing to get familiar with the keyboard.
- **Testing Scene:** The user is tested on their typing performance.
- **Waiting Scene:** The user waits while the system transitions between different keyboard types.

6.2.10 Scene Occurrences

For each keyboard, the sequence of scenes is:

1. Training Scene
2. Testing Scene
3. Waiting Scene

If we have 2 keyboards to test, the sequence repeats for each keyboard. Therefore, for 2 keyboards, the scenes will appear as follows:

- Training Scene: 1 time per keyboard $\times$ 2 keyboards = 2 times
- Testing Scene: 1 time per keyboard $\times$ 2 keyboards = 2 times
- Waiting Scene: 1 time per 2 keyboards

6.2.11 Adjustable Settings in the Database

- **Vibration and Force Feedback:** The vibration and force feedback settings of the gloves used in the test can be adjusted via the database. This ensures that the feedback experienced by the user is tailored to the specific requirements of the test scenario.
- **Waiting Duration:** The duration in seconds of the waiting period between different types of keyboards of the test can be adjusted in the database.

- **Keyboard Names:** The names and types of keyboards to be tested can be updated in the database, making the system adaptable to different testing needs.
- **Number of Phrases:** The number of phrases that the user is required to type in both the training and testing scenes can be configured in the database, ensuring the test is appropriately challenging and comprehensive.

6.2.12 Detection of a Finished Phrase

The system detects that a phrase is finished when the user types the expected number of words of the expected phrase to type and the phrase ends with a period. Upon detection, the system:

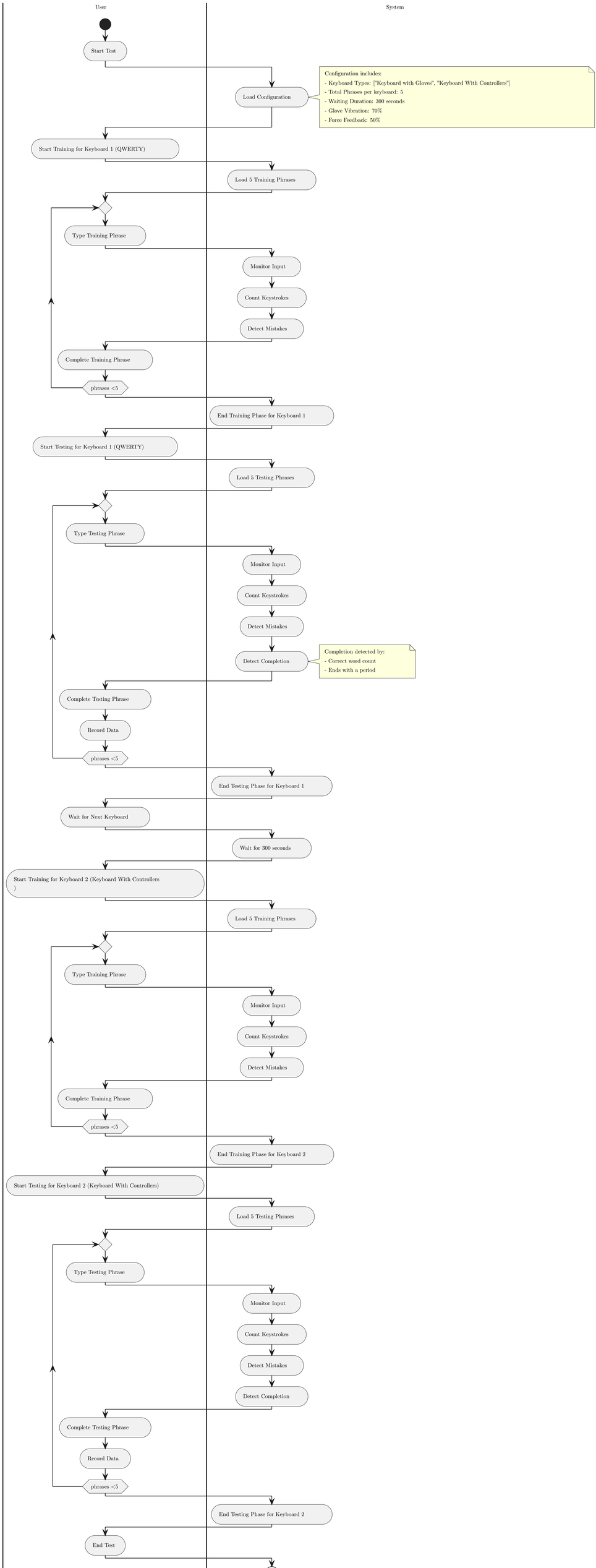
- Records the typing data for the completed phrase.
- Makes the button for proceeding to the next phrase interactable.

6.3 Workflow Summary

The workflow of the typing test scenario involves the following steps:

1. **Initialization:** The system sets up the test environment and prepares the first phrase.
2. **Typing Phase:** The user types the displayed phrase, and the system tracks keystrokes and errors in real-time.
3. **Detection of Completion:** The system detects when the user has finished typing the phrase by checking word count and punctuation.
4. **Data Recording:** Upon completion of each phrase, the system records the typing data.
5. **Phrase Transition:** The user moves to the next phrase by clicking a button.
6. **Scene Transition:** The system transitions between Training, Testing, and Waiting scenes as needed for each keyboard type.

Typing Test Scenario Workflow



6.4 Typing Performance Metrics and Database Registration

6.4.1 Metrics Overview

The performance of users is analyzed using several metrics. These metrics provide a comprehensive understanding of the user's typing efficiency and accuracy in the VR environment.

- **Error Rate:** This metric calculates the percentage of errors by computing the Levenshtein distance between the expected text and the typed text. The Levenshtein distance $d(s,t)$ is a measure of the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one word into the other. The formula used to calculate the error rate is:

$$\text{Error Rate} = \frac{d(s,t)}{\max(\text{Length of Expected Text}, \text{Length of Typed Text})} \times 100 \tag{6.1}$$

  where $d(s,t)$ is the Levenshtein distance between the strings s (expected text) and t (typed text).

- **Character-Level Accuracy:** This metric measures the accuracy at the character level by calculating the Levenshtein distance and then determining the percentage of characters typed correctly. The formula used is:

$$\text{Accuracy in Characters} = 100 - \left(\frac{d(s,t)}{\text{Length of Expected Text}} \times 100 \right) \tag{6.2}$$

  This metric highlights the proportion of correctly typed characters relative to the expected characters.

- **Word-Level Accuracy:** This metric assesses the accuracy at the word level. It compares each word in the expected text with the corresponding word in the typed text, counting the mismatches. The formula used is:

$$\text{Accuracy in Words} = 100 - \left(\frac{\text{Number of Words with Errors}}{\text{Number of Words in Expected Text}} \times 100 \right) \tag{6.3}$$

  This metric provides insight into how accurately the user can reproduce words as opposed to individual characters.

- **Keystroke-Level Accuracy:** This metric evaluates the accuracy based on the number of incorrect keystrokes out of the total keystrokes made. The formula used is:

$$\text{Accuracy in Keystrokes} = 100 - \left(\frac{\text{Incorrect Keystrokes}}{\text{Total Keystrokes}} \times 100 \right) \tag{6.4}$$

This metric helps in understanding the precision of each keystroke made by the user.

- **Typing Speed:** This metric calculates typing speed in words per minute (WPM). It assumes that an average word consists of five characters. The formula used is:

$$\text{Typing Speed} = \left(\frac{\text{Number of Characters Typed}}{5}\right) \div \left(\frac{\text{Time Taken in Seconds}}{60}\right) \quad (6.5)$$

  Typing speed is an important metric to gauge how quickly a user can type, which is crucial for evaluating productivity.

- **Keystrokes per Character:** This metric calculates the number of keystrokes per character typed. The formula used is:

$$\text{Keystrokes per Character} = \frac{\text{Keystroke Count}}{\max(1, \text{Character Count})} \quad (6.6)$$

  This metric indicates the efficiency of the user's typing, with a lower value signifying higher efficiency.

- **Levenshtein Distance:** The Levenshtein distance $d(s,t)$ between two strings s and t is defined as the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform s into t. The distance can be computed using a dynamic programming approach where $d(i,j)$ represents the distance between the first i characters of s and the first j characters of t. The recursive formula is:

$$d(i,j) = \begin{cases} i & \text{if } j = 0 \\ j & \text{if } i = 0 \\ \min \begin{cases} d(i-1,j)+1 \\ d(i,j-1)+1 \\ d(i-1,j-1)+(1-\text{match}(s_i,t_j)) \end{cases} & \text{otherwise} \end{cases} \quad (6.7)$$

  Here, $\text{match}(s_i,t_j)$ is 1 if the characters s_i and t_j are the same, and 0 otherwise. This metric provides a foundational measure of typing accuracy by quantifying how similar the typed text is to the expected text.

6.4.2 Database Registration and Storage

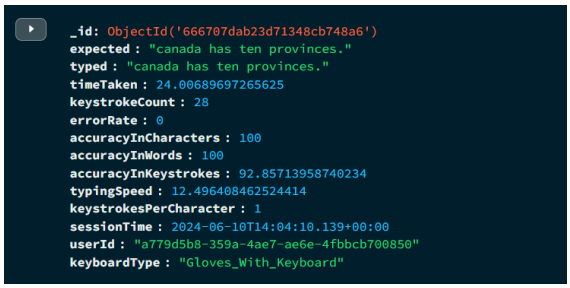
The application establishes a connection to the MongoDB database. This connection is initiated during the startup process, ensuring that the database connection is established when the application begins running. The necessary collections for storing user data and configuration settings are initialized at this point.

Inserting Typing Data

Typing performance data, encapsulated in a structured format, is inserted into the MongoDB collection. This data includes various metrics such as:

- **Expected text:** The predefined text that the user is expected to type during the test.
- **Typed text:** The actual text that the user types, which is recorded for analysis.
- **Time taken to type:** The total time (in seconds) that the user takes to complete typing the expected text.
- **Keystroke count:** The total number of keystrokes made by the user during the typing session.
- **Error rate:** The percentage of errors made by the user, calculated using the Levenshtein distance between the expected text and the typed text.
- **Character-level accuracy:** The accuracy of the typed text at the character level, indicating how many characters were typed correctly relative to the expected text.
- **Word-level accuracy:** The accuracy of the typed text at the word level, indicating how many words were typed correctly relative to the expected text.
- **Keystroke-level accuracy:** The accuracy based on the number of incorrect keystrokes out of the total keystrokes made.
- **Typing speed:** The typing speed measured in words per minute (WPM), assuming an average word consists of five characters.
- **Keystrokes per character:** The number of keystrokes made per character typed, indicating the efficiency of the user's typing.
- **Session time:** The total duration of the typing session, from start to finish.
- **User ID:** A unique identifier for the user, used to track individual performance data.
- **Keyboard type:** The type of keyboard used by the user during the typing session, which could influence performance.

To illustrate the data collected during the typing test, we include two examples of documents stored in MongoDB. The first example shows a case where the phrase is typed correctly, and the second example shows a case where the phrase contains errors.



(a) Correctly typed phrase

(b) Incorrectly typed phrase

Figure 6.3: Examples of typing data stored in MongoDB

Fetching Global Configuration

Global configuration settings are retrieved from the database. The application queries the configuration collection and extracts various settings related to gloves configuration, keyboards to test, session settings, and waiting duration. These configurations are then used to adjust the application's behavior based on the retrieved settings.

Inserting User Data

User data, encapsulated in a structured format, is inserted into the MongoDB collection. This data includes:

- User ID
- User name
- User age
- User sex

7 Conclusion

...

Bibliography

[1] Senseglove nova.

[2] T9 (predictive text).

[3] Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci, 11 2004.

[4] The effect of key size of touch screen virtual keyboards on productivity, usability, and typing biomechanics, 5 2014.

[5] Directional force feedback: Mechanical force concentration for immersive experience in virtual reality, 9 2019.

[6] How to get there when you are there already? defining presence in virtual reality and the importance of perceived realism, 5 2021.

[7] Pseudo-haptics and self-haptics for freehand mid-air text entry in vr, 10 2022.

[8] M. Alba. Gpus power the magic of vr for engineers, 9 2023.

[9] Melanie Baljko and Alvin Tam. Automatic detection of typing errors for large-scale log analysis of text entry. *Human-Computer Interaction*, 21:363–400, 2006.

[10] T. Bezmalinovic. This is how much meta is investing in vr, ar and horizon, 2022.

[11] R. A. Bolt. Put-that-there: Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, pages 262–270, 1980.

[12] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 3d user interfaces: Theory and practice. *Addison-Wesley*, 2007.

[13] Kent Bye. Quantifying touch on 15 dimensions with syntouch. https://voicesofvr.com/496-quantifying-touch-on-15-dimensions-with-syntouch/, 2016.

[14] Neil Charness. Aging and human performance. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 47(1):218–233, 2005.

[15] Microsoft Corporation. Annual report, 2022.

[16] Yngve Dahl and Kari Thorelli Sørli. Text entry in virtual reality: A comprehensive review of the literature. *Virtual Human Interaction*, 31(4):563–579, 2018.

[17] S. Das and T. Goto. Accelerating skill acquisition of two-handed drumming using pneumatic artificial muscles, n.d.

[18] Fred D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340, 1989.

[19] S. A. Douglas, A. E. Kirkpatrick, and I. S. MacKenzie. Testing pointing device performance and user assessment with the iso 9241, part 9 standard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 215–222, 1999.

[20] J. T. Dube and A. S. Arif. Text entry in virtual reality: A comprehensive review of the literature, 6 2019.

[21] J. J. Dudley et al. Performance envelopes of virtual keyboard text input strategies in virtual reality, 12 2019.

[22] A. Giovannelli et al. Exploring the impact of visual information on intermittent typing in virtual reality, 10 2022.

[23] E. B. Goldstein. *Sensation and Perception*. Cengage Learning, 2016.

[24] Jens Grubert, Eyal Ofek, and Michel Pahud. Text input in immersive head-mounted display-based virtual reality using standard keyboards. In *IEEE Conference on Virtual Reality and 3D User Interfaces*, pages 599–606, 2018.

[25] A. HANNAH. Ui animation: A complete guide for beginners, 8 2021.

[26] Vincent Hayward. Xr haptics, implementation & design guidelines, 2022.

[27] Carrie Heeter. *Being There: The Subjective Experience of Presence*. MIT Press, 1992.

[28] Juan David Hincapié-Ramos and Pourang Irani. A meta-analysis of the effects of haptic feedback in human-computer interaction. *International Journal of Human-Computer Studies*, 72(6):609–623, 2014.

[29] HP. Hp reverb g2 specifications. *HP Official Website*, 2023.

[30] HTC. Htc elite xr specifications. *HTC Official Website*, 2023.

[31] HTC. Htc vive focus 3 specifications. *HTC Official Website*, 2023.

[32] HTC. Htc vive specifications. *HTC Official Website*, 2023.

[33] Apple Inc. *Core Haptics*, 2023.

[34] iOS Hacker. Best ai-powered keyboard extensions for iphone and ipad. https://www.ioshacker.com, 2023.

[35] Julie A. Jacko. *Human Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*. CRC Press, 2009.

[36] Emily R. Jones. Advanced audio processing techniques. *IEEE Transactions on Audio, Speech, and Language Processing*, 11(4):512–519, 2003.

[37] L. S. A. O. T. M. B. P. J. Jeong Ho Kim. The effects of virtual keyboard key sizes on typing productivity and physical exposures, 9 2013.

[38] Jaron Lanier. Homuncular flexibility in virtual reality. *Journal of Consciousness Studies*, 13(7-8):5–24, 2006.

[39] F. Laricchia. Smartphones - statistics & facts, 10 2024.

[40] J. Lee and D. Kim. High-accuracy sensor-based tracking for hand movements in virtual reality. *IEEE Transactions on Haptics*, 13(1):45–56, 2020.

[41] Jiaye Leng et al. Efficient flower text entry in virtual reality, 9 2022.

[42] Jiaye Leng, L. W. X. L. X. S., and M. W. Exploration of hands-free text entry techniques for virtual reality. IEEE Xplore, November 2022.

[43] V. Levenshtein. Levenshtein distance: Minimum string distance, n.d.

[44] Meng Li and Yan Zhou. Enhancing surgical training through adaptive feedback in virtual reality. *Journal of Medical VR*, 5(1):10–25, 2023.

[45] Scott MacKenzie. Phrase sets for evaluating text entry techniques. https://www.yorku.ca/mack/RN-PhraseSet.html.

[46] Mark McGill and Stephen Brewster. Dovetail: Modular systems for building and evaluating interfaces for virtual environments. *International Journal of Human-Computer Studies*, 72(8-9):661–674, 2015.

[47] David McNeill. Gesture and environment. *Journal of Pragmatics*, 41(3):518–530, 2009.

[48] Meta. Meta quest 3 specifications. *Meta Official Website*, 2023.

[49] Microsoft. Mixed reality toolkit - mrtk.

[50] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, E77-D(12):1321–1329, 1994.

[51] J. E. Muelaner. 9 best keyboards for android in 2023, 1 2023.

[52] NASA. Anthropometry and biomechanics. https://msis.jsc.nasa.gov/sections/section03.htm.

[53] NASA. Extravehicular activity (eva). https://msis.jsc.nasa.gov/sections/section14.htm.

[54] NASA. Human performance capabilities. https://msis.jsc.nasa.gov/sections/section04.htm.

[55] Donald A. Norman. *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books, 2013.

[56] Oculus. Oculus rift specifications. *Oculus Official Website*, 2023.

[57] OpenAI. Chatgpt can now see, hear, and speak. https://www.openai.com, 2024.

[58] Pico. Pico 4 specifications. *Pico Official Website*, 2023.

[59] James Robert. pydub: Manipulate audio with a simple and easy high level interface. https://github.com/jiaaro/pydub, 2010–2021.

[60] E. S. and R. Teather. Text entry in virtual reality: Implementation of flik method and text entry testbed, 10 2020.

[61] Samsung. Samsung odyssey+ specifications. *Samsung Official Website*, 2023.

[62] Maria Santos and Hieu Nguyen. Optimizing vr environments for language learning: An empirical study. *Journal of Educational VR*, 4(2):134–150, 2021.

[63] E. J. Savitz. Apple's vr/ar headset could be a big revenue booster, 6 2023.

[64] SenseGlove. Senseglove sdk. https://github.com/Adjuvo/SenseGlove-Unity, 2024. Accessed: 2024-05-15.

[65] Thomas B. Sheridan. *Musings on Telepresence and Virtual Presence*. MIT Press, 1992.

[66] Ben Shneiderman and Catherine Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson, 2010.

[67] Mel Slater. Implicit learning through embodiment in immersive virtual reality. *Virtual Reality Psychology*, 22(1):47–58, 2018.

[68] Mel Slater and Maria V. Sanchez-Vives. *Real Virtuality: A Code of Ethical Conduct. Recommendations for Good Scientific Practice and the Consumers of VR-Technology*. Frontiers Media SA, 2020.

[69] A. Smith and T. Brown. A comprehensive review of advanced haptic gloves: Technology and applications. *Haptics and Virtual Reality*, 10(2):123–137, 2022.

[70] John Smith and Anil Kumar. Leveraging artificial intelligence to enhance user interaction in virtual reality. *Journal of Interactive VR*, 7(3):200–215, 2023.

[71] John G. Smith. Digital signal processing: Principles, algorithms, and applications. *Journal of Audio Engineering*, 47(6):453–461, 1999.

[72] Sony. Playstation vr2 specifications. *PlayStation Official Website*, 2023.

[73] Charles Spence and John Driver. *Multisensory Design: Handbook of Multisensory Integration*. MIT Press, 2008.

[74] D. K. Staff. All the parts of a mechanical keyboard explained, 12 2022.

[75] Kay M. Stanney and Kelly S. Hale. *Handbook of Virtual Environments: Design, Implementation, and Applications*. CRC Press, 2019.

[76] Jonathan Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of Communication*, 42(4):73–93, 1992.

[77] R. J. Stone and M. S. Walker. Vibrotactile feedback for enhanced virtual reality experiences. *Journal of Virtual Reality*, 25(4):214–227, 2021.

[78] Yellow Systems. Integrating chatgpt into your web and mobile apps [comprehensive guide]. https://www.yellow.systems, 2024.

[79] Online Tech Tips. How to add and use chatgpt with whatsapp. https://www.online-tech-tips.com, 2023.

[80] Valve. Valve index specifications. *Valve Official Website*, 2023.

[81] Varjo. Varjo xr-3 specifications. *Varjo Official Website*, 2023.

[82] Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 3 edition, 2012.

[83] Christopher D. Wickens and Justin G. Hollands. *Engineering Psychology and Human Performance*. Pearson Education, 2015.

[84] Daniel Wigdor and Dennis Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann, 2011.

[85] T. Xu. What is haptic feedback?, 6 2023.

[86] L. Ying, Y. Fangzhou, J. Ming, J. Z. Rui, and Y. Xuqun. Effect of gestures and smartphone sizes on user experience of text input methods, 11 2021.

[87] A. M. F. P. Z. and M. S. A. Kruger. Selection-based text entry in virtual reality, 4 2018.

[88] Peng Zhang, Xiangyuan Lan, Victor Chang, and Xin Liang. Deep learning in virtual reality. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2344–2358, 2017.