# A Survey and Implementation of Fast-Input Typing Methods in Virtual Reality


Ryan Kim* Computer Science, Cornell Tech

Katherine Donahoe† Computer Science, Cornell Tech

Prithvi Sriram‡ Computer Science, Cornell Tech


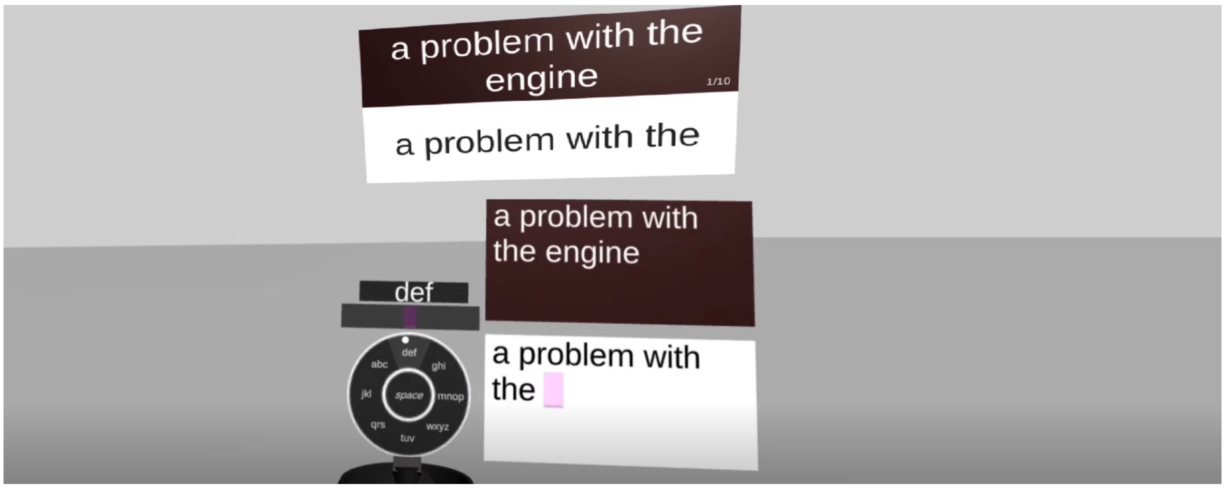


Figure 1: VRKeyboard operating in the testing suite


## Abstract

This work demonstrates VRKeyboard - a joystick and button-based text input method in Virtual Reality that resembles the semantic appearance and functionality of older T9 phone keyboards. This work also presents a modified version, Cheat VRKeyboard, which attempts to further improve the user experience by adding a shortcut “cheat” layer. These come from a collective goal to design a VR-oriented keyboard that is intuitive in nature and is functionally adequate for VR users’ general needs. Through experimentation with users, VRKeyboard’s two designs were evaluated both qualitatively and quantitatively to observe if they align with user expectations of functionality while maintaining a consistent performance across multiple users. Key results indicated that on average, users typed 8.49 WPM for the VRKeyboard, and 5.44 WPM for the VR “Cheat” keyboard, with experienced users typing 13.3 WPM for the VRKeyboard. Users made fewer errors with the VR Cheat keyboard, however felt it was more physically and mentally demanding than VRKeyboard.



**Index Terms:** Human-centered computing—Human computer interaction—Interaction paradigms—Virtual reality; Human-centered computing—Interaction techniques—Text input—


## 1 Introduction

With the rise in popularity of VR and Head-Mounted Displays, or HMD’s, the question of how to afford text input in VR inevitably arises as well. This is but one of many core issues surrounding consumer-level VR experiences, and research into possible solutions has extended from different input modalities such as soft keyboards with haptic feedback [15], word-gesture keyboards [8], and gaze-based keyboards [10] to heuristic evaluations such as concerns surrounding cognitive overload and fatigue.

In order to address this issue in an intuitive way, this study proposes that a soft keyboard design which aligns with users’ preconceived biases towards text input and conforms to the form factors of contemporary, popular VR controllers can also be a viable text input method. Both the Oculus Quest and HTC Vive, with their large consumer bases, have established themselves as the gold standard for consumer-level VR. Innovating within this ecosystem provides for a quick and easy way of reaching thousands of people. Thus, this study aspires to find an effective text-input mechanism with regards to the Quest and Vive as our main development environment. The answer to this inquiry also depends on how natural a mapping between button inputs and control factors feels to users.

Informal observations into the affordances of the Oculus Quest and Vive’s controllers suggest that a key selection method that involved the joystick would be most intuitive. A symbolic attribution of multiple characters to one input motion as well as a key layout that could afford such attributions therefore are required for a keyboard of this design quality.

These insights form the foundation for the design of VRKeyboard. VRKeyboard relies on a combination of symbolic attribution and a circular T9-based keyboard layout as its main design feature. The rest of this paper describes the design and functionality of VRKeyboard, which takes advantage of the affordances offered by the Oculus Quest and utilizes a keyboard layout and interaction metaphors that aligns closely with that of T9 keyboards. This paper also outlines the methodology and findings of a user study that evaluates the performance of both VRKeyboard and an alternative


*e-mail: rk494@cornell.edu
†e-mail: kd459@cornell.edu
‡e-mail: ps976@cornell.edu

design to VRKeyboard coined as Cheat VRKeyboard. At the end, this paper suggests how to improve the design of VRKeyboard and future possible extrapolations of the conclusions found from user testing.

## 2 Related Work

Before the advent of VR as a field of study, researchers were invested in alternative input methods that differed from the then-popular QWERTY layout. Chord Keyboards [12] offered a keyboard layout wherein only a small number of buttons were ergonomically designed to fit the grip of one's hand. This keyboard relied on symbolic attribution of multiple characters to one button, reducing the total number of buttons in the interface and allowing for one-handed text input. While the different input scheme required significant training time, the concept of a one-handed input method appealed to many and sometimes outweighed the cost of training time.

A later empirical comparison between a VR soft keyboard and miniature keyboards attached to various parts of the body suggested that a miniature keyboard strapped to the forearm of the non-dominant hand offered a better experience than purely virtual keyboards [14]. At the time of this paper's release, VR was not at the adequate level of polish that contemporary, consumer-level VR experiences offer. Several years after that, an empirical comparison between Pinch type, Pen-Tablet, Chord, and Speech-to-text input metaphors revealed that neither of the four metaphors are better than the other in terms of both performance and user comfort [1]. In particular, the Chord keyboard was ousted as the least viable option due to the lowest Character per Minute rate (CPM), as well as the highest number of errors, though this can be attributed to the high cost of training necessary to use the Chord keyboard.

Since then, various researchers have designed and proposed unique solutions to the problem of text input in VR. Word Gesture Keyboards (WGK's), such as Markussen's Vulture [8] and Yu et al.'s head-based GestureType [16] are notably common, yet many suffer from fatigue due to muscle stress over a significant period of time. Stylus-based methods such as Cirrin [7] are also common, though Bowman's empirical study from 2002 has shown that stylus-based methods suffer from severe hand fatigue over other modalities [1]. A common trend in gesture-based keyboards therefore is fatigue, and therefore an intuitive design would have to remove the potential for fatigue as much as possible.

An investigation by Ren and O'Neill on dual-circle key layouts for gesture-based text input showed the potential for a circular key layout, as a circular arrangement reduced fatigue during key selection due to all keys being equidistant from one center [11]. The idea of a circular keyboard is easily translatable to a joystick-based implementation, and the idea of circular keyboards has also been addressed by other researchers as well. Most notable cases include Cirrin [7], which features a trained typing rate of 20 WPM, Quikwriting [2], which featured a joystick-based circular key implementation that reached up to 13 WPM, and most recently HiPad's 6-key circular layout [3], which reached an average trained WPM of approximately 13.57. Mankoff and Abowd in particular revealed that any circular-based keyboard implementation would cause a narrowing of the user's vision to only the circumference of the circle. This means that any periphery items such as the textbox itself or other buttons outside the circumference would be largely ignored, which can be extrapolated to say that periphery buttons and displays should be situated around the circumference of the circular layout in order to increase their exposure to the user.

Alongside this stipulation with circular key layouts, Lee et al. realized with their air-gesture text input interface HIBEY [5] that tactile feedback reduces the potential for error due to the natural feeling of pressing onto a surface with fingers. This means that a keyboard interface that takes advantage of surface-based tactile feedback would be advantageous for error reduction. Furthermore, [4] realized that any newcomers unfamiliar to a virtual reality-based keyboard system would prefer indicators in the VR world that hinted at their hands' current orientation [4]. VR Keyboards therefore will benefit if they allow for some indication of the user's hand or finger position with respect to buttons and keys.

## 3 Keyboard Design

VRKeyboard consists of two keyboard variations, both of which are intended to make the user's onboarding experience feel intuitive. Both designs are adapted from the 3x4 numeric telephone keypads present in the first generation of personal mobile phones, as seen in Figure 2, and take cues from lessons learned from past related work.


Figure 2: An example of the 3x4 numeric telephone keypad

### 3.1 Design 1: VRKeyboard

The first design, as seen in Figure 3, is a direct adaptation of the 3x4 numeric keypad. The 26 letters of the alphabet are divided into 8 keys mapped to the eight dominant directions of the joystick. Each of the eight possible directions of the joystick therefore allows the user to choose a subset of 3-4 letters, and the user can rotate through and select characters in each character group by clicking a face button on the Quest's controller multiple times.


Figure 3: The default design of VRKeyboard. Note that the white circle indicates the joystick's current orientation and the black textbox above the keyboard that corresponds with the key the joystick is currently highlighting

### 3.2 Design 2: Cheat VRKeyboard

In an attempt to improve the speed of typing and reduce the number of clicks to select a desired letter, a significant modification has been made to the VRKeyboard. Building on top of the first design, the second design has an inner (direct access) ring of letters referred to as the "Cheat" layer. The second design is shown in Figure 4. These letters are among the most frequently used letters in the english alphabet [9]. This Cheat VRKeyboard provides the users an opportunity to directly select certain keys from the alphabet instead of clicking through a subset of letters.


Figure 4: The Cheat VRKeyboard. Note the Cheat layer unique to this version of VRKeyboard.

### 3.3 Position of Virtual Textbox

As indicated by prior research [7], a textbox located separately from a circular key layout will increase the chances of the textbox being ignored, and directing one's gaze between the keyboard and textbox will lead to eye and muscle fatigue. To compensate, a second virtual textbox is placed right beside the circumference of the keyboard that matches the content of the current textbox the user is typing into. By keeping the keyboard right beside the keyboard, VRKeyboard largely reduces the movement of the head from the textbox to the keyboard.

### 3.4 Functionality

VRKeyboard requires the use of both controllers, with the non-dominant hand representing the non-dominant controller and the dominant hand representing the dominant controller. The keyboard and colocated textbox are attached to the non-dominant controller, as seen in Figure 5. For a right-handed person, they will use their left, non-dominant hand's joystick to select the keys of the circular keyboard. Each key is symbolically attributed to a subset of 3-4 characters, and an easily accessible face button or trigger ( "A" on the Oculus Quest's controller) is used to rotate through the characters among the chosen character's subset. For instance, the "Up" direction on the non-dominant hand's joystick is used to choose the character subset "def", and pressing button "A" once selects "d", twice selects "e", or three times selects "f". Spaces are added either by clicking the non-dominant hand's joystick or by pressing the "A" button when the non-dominant hand's joystick is hovering


Figure 5: The colocated textbox, alongside another textbox useful for displaying instructions or questions.

over "Space". For error correction such as backspace, a second face button ("B" on the Oculus Quest's controller) is mapped to the keyboard's delete function. With regards to the Cheat VRKeyboard, the cheat layer is togglable via capacitive hold of the non-dominant hand's index trigger. Selection of the keys attached to the cheat layer is the same as for the default layer. Figure 6 further details the controller functionality, and Table 1 and 2 explain the use case of each of the options as well as marks potential future functions.


Figure 6: A visualization of the control scheme of VRKeyboards, both non-cheat and cheat.

## 4 Experiment Setup

### 4.1 System Setup

We have used the Oculus quest to test participants. These quest devices belonged to either the authors of the paper or were borrowed from the Mixed Reality Collaboratory at Cornell Tech.

### 4.2 Participants

A total of 5 subjects (4 male, 1 female) were recruited through family members of the authors. In all cases, care was taken to make sure participants were made familiar with the Oculus environment. The devices used were hygienic and cleaned at the beginning of each experiment. Participants were aged 18-28 (M=22, SD=3.0). The experiments took an average of 35 minutes to complete for each participant.

| Non-Dominant Controller | Functions | Notes |
|---|---|---|
| 8-Directional Trackpad or Joystick | Character Selection | Keys must be accessible with symbolic attribution instead of mapping one character to one key |
| Button #1 (Face Button) | (none) | Face buttons must require pulling the thumb away from character selection - hence why this is not attributed to a function as of yet. |
| Button #2 (Face Button or Index Trigger) | (none) | Depending on the controller type, either the controller has a secondary face button or a dual-level trigger. |
| Button #3 (Index Trigger) | Shortcut / Cheat Capacitive Toggle | Chosen for the shortcut toggle due to easy access (index finger) without impeding character selection (thumbstick) |

Table 1: **Non-Dominant Controller Functions**

| Dominant Controller | Functions | Notes |
|---|---|---|
| 8-Directional Trackpad or Joystick | (none) | Can be attributed to later functions (ex. Sentence navigation) |
| Button #1 (Face Button) | Backspace (Press & Hold) | Triggers are more likely to implicate actions that move forward in nature (ex. shooting something, selecting something) - any a backspace button therefore must be attributed to a button that does not implicate such a nature of interaction |
| Button #2 (Face Button or Index Trigger) | Character Typing (Press & Hold) | If attributed to a button, button must be more accessible than the backspace button. If attributed to a trigger, it must be definite press, not capacitive. |
| Button #3 (Index Trigger) | (none) | Can be attributed to future functions (ex. sentence navigation) |

Table 2: **Non-Dominant Controller Functions**

### 4.3 Phrase Sets

In creating a phrase set, the goal is to use phrases that are moderate in length, easy to remember, and representative of the target language. The phrase sets were chosen from [6]. These collections of phrases - amounting to 500 sets - were particularly published for such evaluations. From these 500 sets, 6 phrase sets are chosen for our experiments. In each experiment, 2 of these phrases were randomly chosen for each user for the testing sessions. This was done in order to reduce cognitive bias and familiarity with the phrases. Table 3 shows the count of the number of words and characters for each phrase set. The average number of words is 50.33 (ranging 46-54) and the average number of characters is 250.33.

| Phrase set | words | characters |
|---|---|---|
| 1 | 51 | 267 |
| 2 | 54 | 291 |
| 3 | 53 | 269 |
| 4 | 47 | 215 |
| 5 | 51 | 237 |
| 6 | 46 | 223 |
| **Average Across Sets** | **50.33** | **250.33** |

Table 3: **Phrase Sets Statistics**

## 5 Experimentation

The experimentation sequence begins with users typing with the VRKeyboard and later on the Cheat VRKeyboard. To get acquainted with VRKeyboard (as well as the Cheat VRKeyboard), each participant was given 5 minutes to get accustomed prior to their test session. Following the practice sessions, each test session required the users to type ten phrases. An example of a phrase in a testing session is located in Figure 7. Thus, there are 4 different sessions in all:

- Practice Session 1 - users gets 5 minutes to get accustomed to the VRKeyboard
- Test Session 1 - users type ten phrases with the VRKeyboard
- Practice Session 2 - users gets 5 minutes to get accustomed to the Cheat VRKeyboard
- Test Session 2 - users type ten phrases with the Cheat VRKeyboard

The test sessions are timed to calculate the Words Per Minute (WPM) and Characters Per Minute (CPM). The sessions are closely moderated to keep track of the user's errors. The quantitative measures thus calculated are Words Per Minute (WPM), Characters Per Minute (CPM), and Error Rate.

Post the sessions, the users were asked to fill a questionnaire adapted from the standardly used presence questionnaire (using the 4-point Likert scale) as well as the NASA task load index (using the 7-point Likert scale).

It is to be pointed out that the users are not compelled to use the cheat layer in test session 2. This is particularly enforced to ensure that we measure the degree of use of the cheat layer and thus measure the participant's natural choices.

## 6 Initial Results

### 6.1 Typing Results

Quantitative assessments were derived from users typing a set of ten phrases. Using VRKeyboard, users typed 8.49 WPM and 35.68 CPM on average, making on average 2.21 errors per minute. Experienced typists typed 13.3 WPM and 71.67 CPM.


Figure 7: Example of a user typing a phrase in the testing suite

Using Cheat VRKeyboard, users typed 6.55 WPM and 25.56 CPM on average, making on average 0.99 errors per minute. All users except one typed both fewer words and fewer characters with Cheat VRKeyboard than with VRKeyboard. The anomaly user typed 10.12 WPM with Cheat VRKeyboard, but only 7.91 with the VRKeyboard. Users made less errors with Cheat VRKeyboard, likely because they were typing slower. More detailed results can be found in Table 4.

### 6.2 Questionnaire Results

Overall, users rated both keyboards well on the presence questionnaire. Users felt in control of the keyboard. They felt that there was no noticeable delay when typing, and that the keyboard was responsive to their typing. The keyboard layout did not affect users' sense of presence. Overall, users scored equally high in feeling accomplished for both keyboards. More detailed results can be viewed in Tables 5 and 6.

When interviewed after the test period, users felt that the Cheat Keyboard suffered in performance due to the additional step of holding the index finger trigger to toggle the cheat layer. Furthermore, users have to initiate a recall step to remember which characters are on the cheat layer despite the cheat layer already depicting the characters visually. These additional steps add cognitive load on the user's typing capabilities and therefore slow down the typing rate with the Cheat Keyboard. Some users reported that since they already knew the default layer's key mappings, they sometimes gravitated towards using the default layer and ignored the cheat layer. Although users deemed Cheat VRKeyboard to be both more mentally and physically demanding, on average they felt they didn't work as hard to achieve their level of performance and thought that the Cheat VRKeyboard was faster despite the quantitative data revealing slower WPM rates. Participants also felt less rushed when accomplishing typing tasks, and on average felt calmer when completing tasks.

| Questions | Cheat | Regular |
|---|---|---|
| How mentally demanding was the task? | 6.4 | 5.4 |
| How physically demanding was the task? | 3.6 | 2.2 |
| How hurried or rushed was the pace of the task? | 5.2 | 6.4 |
| How successful were you in accomplishing what you were asked to do? | 6.8 | 7.2 |
| How hard did you have to work to accomplish your level of performance? | 4.8 | 5.8 |
| How insecure, discouraged, irritated, stressed, and annoyed were you? | 3.6 | 4.6 |
| How much delay did you experience between your actions and expected outcomes? | 6.6 | 7 |

Table 5: **NASA task load**

Across both the default VRKeyboard and Cheat VRKeyboard, users were initially confused about the way the characters were ordered and symbolically attributed to the eight keys on the default VRKeyboard layout, but this confusion drastically disappeared and reportedly felt intuitive to users after several minutes of practicing with the keyboards. Additional feedback from users mentioned a lack of capitalization features, numbers, and punctuation as subtle distractions during the typing experience. Furthermore, users who were not used to VR had trouble identifying buttons by name and had to verbally ask what the keys were on each controller. Due to this confusion, users sometimes spent time trying to identify which buttons they had to press during the testing phase, elongating their time spent with the test and thereby reducing their WPM.

## 7 Discussion

### 7.1 Limitations

Several key limitations in the implementation of VRKeyboard and Cheat VRKeyboard are readily apparent after user testing and must be mentioned. Despite the Cheat VRKeyboard seeing fewer errors on average than the default VRKeyboard, users still felt cognitively overloaded by the process of selecting keys on the cheat layer. One explanation for this behavior is the additional step of toggling the cheat layer via index trigger hold on the controller, which interferes with the core user loop of typing.

Furthermore, character overlaps between the cheat layer and

| | | WPM | | CPM | | Character Errors Per minute | |
|---|---|---|---|---|---|---|---|
| | Typing Experience | Session 1 | Session 2 | Session 1 | Session 2 | Session 1 | Session 2 |
| User1 | Beginner | 5.79 | 2.78 | 30.31 | 12.72 | 2.04 | 1.30 |
| User2 | Beginner | 6.99 | 3.86 | 36.59 | 17.66 | 3.84 | 0.82 |
| User3 | Moderate | 7.91 | 10.12 | 40.15 | 46.29 | 0.75 | 0.86 |
| User4 | Moderate | 8.45 | 3.87 | 44.23 | 17.71 | 5.62 | 0.45 |
| User5 | Experienced | 13.3 | 6.55 | 71.67 | 29.96 | 10.34 | 4.04 |
| **Average** | | **8.49** | **5.44** | **35.68** | **25.56** | **2.21** | **0.99** |

Table 4: **WPM, CPM, and Error Results**

| Questions | Cheat | Regular |
|---|---|---|
| How much were you able to control events with the keyboard? | 7.8 | 8.4 |
| How much did the control devices interfere with the performance of assigned tasks or with other activities? | 6.8 | 7.2 |
| How much did the visual display quality interfere or distract you from performing assigned tasks or required activities? | 6.4 | 5.6 |
| How proficient in interacting with the virtual environment did you feel at the end of the experience? | 6.8 | 8 |
| How quickly did you adjust to the virtual environment experience? | 8.6 | 8.6 |
| To what degree did you feel confused or disoriented at the beginning of breaks or at the end of the experimental session? | 5.4 | 4.6 |
| How well could you examine objects from multiple viewpoints? | 9 | 7.8 |
| How closely were you able to examine objects? | 9.2 | 8.6 |
| Were you able to anticipate what would happen next in response to the actions that you performed? | 8.6 | 8.6 |
| How much did your experiences in the virtual environment seem consistent with your real-world experiences? | 7.2 | 7.4 |
| How inconsistent or disconnected was the information coming from your various senses? | 7.2 | 7.8 |
| How compelling was your sense of objects moving through space? | 7 | 7.8 |
| How aware were you of your display and control devices? | 8.6 | 9.4 |
| How aware were you of events occurring in the real world around you? | 5 | 5.8 |
| How well could you concentrate on the assigned tasks or required activities rather than on the mechanisms used to perform those tasks or activities? | 6.8 | 6.6 |
| How much delay did you experience between your actions and expected outcomes? | 9.2 | 9.8 |
| How much did the control mechanism of the keyboard distract you from the main task? | 6.8 | 6.2 |
| How much did the visual aspects of the keyboard involve you? | 9.2 | 8.8 |
| How natural did your interactions with the keyboard seem? | 8 | 8.2 |
| How responsive was the keyboard to actions that you initiated or performed? | 8.4 | 9.8 |
| Were you involved in the experimental task to the extent that you lost track of time? | 7 | 6.8 |

Table 6: **Presence Questionnare Results**

default layer (i.e. characters in the cheat layer are also available in the default layer) confuse the user and force them to focus on the cheat layer, distracting them from the typing task. Compounding this effect is the possibility that users were already trained in using the default VRKeyboard to the point that they naturally gravitate to using only the default keys and not the cheat layer keys. To fix these issues, the index toggle trigger would have to be remapped such that the functionality of the cheat layer does not interfere with the typing task at hand. Overlaps between the cheat layer and default layer can be avoided by repurposing the cheat layer with different characters such as punctuation or numbers, thereby reducing fixation on the cheat layer. To avoid the possibility of the learning effect of the default VRKeyboard affecting the output of the Cheat VRKeyboard, the order by which the keyboards are introduced must be randomized, alongside an increased number of test phrases to add an additional observation of the learning effect.

### 7.2 Future work

Future work could entail incorporating voice-to-text functionality. A promising 2016 study by Stanford indicated that a voice-to-text implementation could achieve up to three times WPM compared to a standard QWERTY keyboard [13], as well as a lower error rate, provided there is a high-functioning machine learning driven backend. In addition, to better optimize the existing keyboard implementations, auto-complete functionality could be added to increase typing speed. There are many open source APIs that could be of assistance to achieve auto-complete functionality. Both of these potential features will definitely increase the speed of text entry, however, were out of scope of this project, which sought to examine text entry via typing on a virtual keyboard only. In addition, the testing suite could be expanded further to gather more detailed data on error rates as well as the amount of times the cheat layer was accessed.

### 7.3 Conclusion

Two VR-oriented keyboards were proposed, VRKeyboard and Cheat VRKeyboard, each designed to be intuitive to users as well as functionally adequate. Results derived from testing the keyboard with users revealed a clear difference in speed and comfort between the two. Results indicate that on average users typed 1.5 times faster on VRKeyboard, indicating that VRKeyboard is more functionally adequate for VR users. In addition, users felt VRKeyboard was less physically and mentally demanding than Cheat VRKeyboard, therefore more comfortable to use. Despite this, users made almost half as many errors with the VR Cheat keyboard compared to the VRKeyboard, and also generally claimed that it felt faster and they felt calmer using it. While there are clear differences in speed and mental/physical exertion levels, both keyboards were rated well for presence and task load. Both are well suited for users, and compare well against existing keyboard designs.

## References


[1] D. Bowman, C. Rhoton, and M. Pinho. Text input techniques for immersive virtual environments: An empirical comparison. *In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Baltimore, MD, USA*.

[2] P. Isokoski and R. Raisamo. Quikwriting as a multi-device text entry method. In *Proceedings of the Third Nordic Conference on Human-Computer Interaction*, NordiCHI '04, p. 105–108. Association for Computing Machinery, New York, NY, USA, 2004. doi: 10.1145/1028014.1028031

[3] H. Jiang. Hipad: Text entry for head-mounted displays using circular touchpad. *IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2020.

[4] P. Knierim, V. Schwind, A. M. Feit, F. Nieuwenhuizen, and N. Henze. Physical keyboards in virtual reality: Analysis of typing performance and effects of avatar hands. In *Proceedings of the 2018 CHI Conference*

*on Human Factors in Computing Systems*, CHI '18, p. 1–9. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3173919

[5] L. H. Lee, K. Yung Lam, Y. P. Yau, T. Braud, and P. Hui. Hibey: Hide the keyboard in augmented reality. In *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom*, pp. 1–10, 2019.

[6] I. S. MacKenzie and R. W. Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, p. 754–755. Association for Computing Machinery, New York, NY, USA, 2003. doi: 10.1145/765891.765971

[7] J. Mankoff and G. D. Abowd. Cirrin: A word-level unistroke keyboard for pen input. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, p. 213–214. Association for Computing Machinery, New York, NY, USA, 1998. doi: 10.1145/288392.288611

[8] A. Markussen, M. Jakobsen, and K. Hornbæk. Vulture: A mid-air word-gesture keyboard. *In Conference on Human Factors in Computing Systems - Proceedings*.

[9] C. D. of Mathematics. *English Letter Frequency*.

[10] V. Rajanna and J. Hansen. Gaze typing in virtual reality: Impact of keyboard design, selection method, and motion. *Eye Tracking Research and Applications Symposium (ETRA)*.

[11] G. Ren and E. O'Neill. Freehand gestural text entry for interactive tv. In *Proceedings of the 11th European Conference on Interactive TV and Video*, EuroITV '13, p. 121–130. Association for Computing Machinery, New York, NY, USA, 2013. doi: 10.1145/2465958.2465966

[12] N. Rochester, F. C. Bequaert, and E. M. Sharp. The chord keyboard. *Computer*, 11(12):57–63, Dec. 1978. doi: 10.1109/C-M.1978.218024

[13] S. Ruan, J. O. Wobbrock, K. Liou, A. Ng, and J. A. Landay. Comparing speech and keyboard text entry for short messages in two languages on touchscreen phones. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(4), Jan. 2018. doi: 10.1145/3161187

[14] B. Thomas, S. Tyerman, and K. Grimmer. Evaluation of text input mechanisms for wearable computers. *Virtual Real.*, 3(3):187–199, Sept. 1998. doi: 10.1007/BF01408561

[15] C. Wu, C. Hsu, and T. et. al. A virtual reality keyboard with realistic haptic feedback in a fully immersive virtual environment. *Virtual Reality*, 21.

[16] C. Yu, Y. Gu, Z. Yang, X. Yi, H. Luo, and Y. Shi. Tap, dwell or gesture? exploring head-based text entry techniques for hmds. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, p. 4479–4488. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3025453.3025964