RotateEntry: Controller-rolling-style Text Entry for Three Degrees of Freedom Virtual Reality Devices

by

Ziming Li

Project submitted in partial fulfillment of the requirements for the degree of Master of Science in Human Computer Interaction

Rochester Institute of Technology

B. Thomas Golisano College
of
Computing and Information Sciences

Department of Information Sciences and Technologies

2020-12-03

RotateEntry: Controller-rolling-style Text Entry for Three Degrees of Freedom Virtual Reality Devices

Ziming Li

***Abstract*—Raycasting is the mainstream solution for text entry on virtual reality devices. For a keystroke, a raycasting-style text input method requires a user to aim at a particular key on the virtual keyboard in space by holding up the controller. In this way, the user might feel muscle fatigue after long-time use. In this work, we propose RotateEntry, a controller-rolling-style method for text entry on three degrees of freedom virtual reality devices. To move the key-selecting cursor in two dimensions on a QWERTY layout virtual keyboard, we developed three variants of RotateEntry: Rotate Column Rotate, Rotate Key, and Rotate Column Point. We conducted a comparative empirical evaluation of the four text input methods, including three proposed controller-rolling-style text input methods and the standard raycasting-style one. Text entry performance, accuracy, workload, usability, and user experience were tested and evaluated. Due to the COVID-19 situation, our study was conducted remotely. The impact of using online formats on VR research had also been assessed. After evaluating with 5 participants, we identified that Rotate Key had a higher text entry rate, outstanding overall user experience, and excellent overall workload performance among the three variants of RotateEntry. However, no evidence had been investigated to support the hypothesis that RotateEntry had better performance and experience compared to Raycasting.**

***Keywords*—Virtual reality, text entry method, three degrees of freedom, controller, QWERTY keyboard layout**

Table of Content

I. INTRODUCTION

Three degrees of freedom (3DoF) virtual reality (VR) devices are VR controllers which can detect rotational movements to interact with virtual content presented in a head-mounted display. Many recent commercial VR devices such as the Oculus Go, Google Daydream have featured 3DoF controllers with additional inputs such as buttons. Compared to a 6DoF VR device, which can track positional and rotational movements in a 3-dimensional space, a 3DoF VR device has a lower demand for spatial movements. A 3DoF device can primarily support activities with less movement, such as web browsing, watching videos, and VR socialization. Among these scenarios, the text input technique plays an essential role.

The controller-based raycasting keyboard is a popular built-in text entry solution for 3DoF VR devices. While using this type of text input method, the user can aim the virtual ray, which is cast by the controller, at a particular key on the virtual keyboard, and enter the focused character by pressing a button on the VR controller (Boletsis and Kongsvik 2019). The raycasting-style text input method provides an intuitive and precise way for text entry on VR devices. However, while using the raycasting input, the user has to hold a controller, keep raising his/her arm in use, and frequently move the arm in space to aim at a key. After long term use, this using posture can cause arm muscle fatigue (Grubert et al. 2018), resulting in reducing text input performance, accuracy, and user experience. Also, the aim-and-shoot style interaction implemented by the raycasting input would be challenging for those who cannot keep their arms in mid-air for a long time.

This paper proposes RotateEntry, a “controller-rolling-style” text entry method for 3DoF VR devices. RotateEntry moves a key-selecting cursor through the virtual keyboard using the controller’s rolling angle and relative pitching angle. In this way, it frees the VR user from the need to enter text with a fixed posture, and instead, the user can put his/her hand holding the controller in any spatial position. Hence, it could be a potentially more efficient and effort-saving way for VR text entry than the raycasting solution.

This study focused on the interaction comparison between RotateEntry and the standard raycasting method. Since the typical raycasting-style text input technique uses a QWERTY keyboard layout (Dube and Arif 2019), RotateEntry implemented the same keyboard layout, aiming to eliminate the potential effect of the keyboard layout. We developed three interaction methods named Rotate Column Rotate, Rotate Key, and Rotate Column Point. They were using the RotateEntry concepts that look at how to move the cursor across the standard virtual QWERTY keyboard. Next, we evaluated these three methods and the traditional raycasting method in a comparative empirical study. The knowledge obtained from this study might help us identify a proper interaction for RotateEntry and provide us insights to improve it further.

Due to the COVID-19 pandemic, we transformed this evaluation, previously designed as a face-to-face study, into a remote one. In this case, we also investigated the impact of using online formats on VR research.

II. Related Work

Existing VR text entry techniques can mainly be classified into three types: (a) game-controller-based text input methods, which entering text by using two thumbsticks on the gamepad; (b) VR-HMD-based (Head Mounted Display) text input methods, which entering text based on user's head motions; and (c) VR-controller-based text input methods, which entering text by using VR controllers.

In Pizzatext, a game-controller-based text input method, a customized keyboard layout was implemented (Yu et al. 2018). This text entry technique divided a circular layout into several slices, while each slice includes a certain number of letters. Yu's team has also offered three different keyboard layouts for their method. The words per minute (WPM) and total error rate (total ER) of each proposed layout were tested. Therefore, the best performance keyboard layout for their proposed text entry method could be identified by analyzing the data.

RingText was a dwell-free VR-HMD-based text entry method (Xu et al. 2019). In RingText, a circular keyboard layout with a "go-and-hit" character selection interaction was implemented. Xu's team has conducted three studies for the RingText evaluation. The first study was a comparative evaluation of 12 types of RingText keyboard layout designs, which helped them determine a proper layout design for RingText. The second one was a within-subject comparative experiment, which helped identify the text-entry rate and accuracy of RingText compared with those of the other four hands-free text input techniques. The last one was a 4-day study, which measured the trend of the novice and expert users' typing performance after long-term practice.

A potential solution for VR-HMD-based text input methods is SliceType (Benligiray, Topal, and Akinlar 2018), a gaze typing method involving a customized circular keyboard layout. The work from the Benligiray's team applied the eye-tracking technique to text entry. Their text input method's virtual keyboard can dynamically allocate a larger space for the target character key, which can be more comfortable for an eye-tracking cursor to focus on and assist in typing text more efficiently.

In terms of the VR-controller-based text entry techniques, Boletsis and Kongsvik's work evaluated four text entry techniques built for dual-controllers VR devices. The four evaluated techniques all used a QWERTY keyboard layout but with different keystroke methods. Boletsis and Kongsvik have conducted a within-subject comparative empirical evaluation among those four controller-based input methods. The text entry rate and accuracy were tested using the scale of words-per-minutes and total error rate. In addition to the text entry performance, the usability and the user experience of those text-entry techniques were also investigated using the System Usability Scale (SUS) questionnaire (Brooke 2013) and the Game Experience Questionnaire (GEQ) (IJsselsteijn, De Kort, and Poels 2013). In their evaluation, both the auto-completion and auto-correction functionalities had been disabled to eliminate potential effects.

The prior studies discussed above have proposed a series of VR text input methods based on various text entry interactions and virtual keyboard layouts. They also presented a few excellent research methods and metrics for a VR text input technique evaluation. However, while a user's perception of spatial presence in the VR display is minimal (Seibert and Shafer 2017), there is still a lack of study on controller-based VR text input methods less reliant on the perception of spatial presence.

III. EVALUATED TEXT INPUT METHODS FOR 3DOF VR DEVICES

Four different controller-based text input methods for 3DoF-single-controller VR devices were developed for the evaluation in this study, which was Rotate Key (RK), Rotate Column Point (RCP), Rotate Column Rotate (RCR), and Raycasting (C). The first three text input methods were implemented using the proposed RotateEntry interaction, and the last one was implemented using the standard raycasting-style interaction. Since this study focused on evaluating and comparing the text entry interactions, all of the aforementioned text entry methods shared the same QWERTY keyboard layout, which was similar to the Oculus Go built-in keyboard one.

All four text input methods evaluated in this study were developed on the Unity game engine. The evaluation was a demo application (see Fig. 1) running on an Oculus Go VR headset with an Oculus Go 3DoF controller. In the user interface of the application, there was a server connection status indicator on the left side of the keyboard. The indicator aimed to inform the user of the current network connection. On the right side of the keyboard, an instruction panel was shown. It was used to instruct the user on using the current activated text input method. The phrase panel at the top of the interface displayed the phrase that needed to be entered currently. The user was required to use the current activated text input method to enter characters in the input box below the phrase panel.

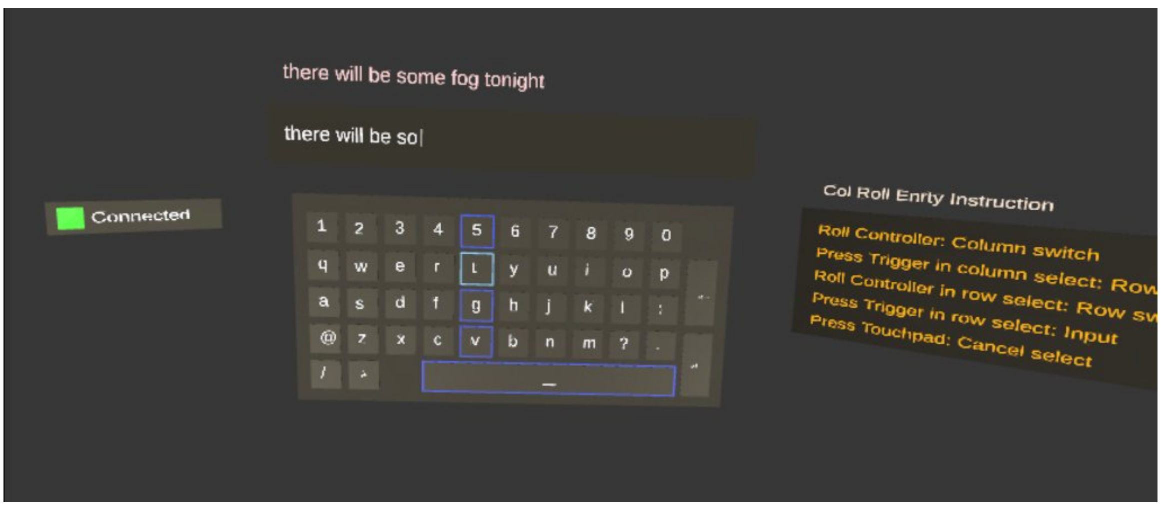

Fig. 1. A screenshot of the evaluated demo's user interface

A. RotateEntry Concept

As stated above, the implementation of RotateEntry was to capture the controller's rotational input of a particular axis. The orientation of RotateEntry's key-selecting cursor's movement was detected based on the offset between the controller's current effective rotational input and the base rotation (pitching/rolling) angle value. The way the base pitching angle captured was various base on each interaction method. In contrast, the base rolling angle was always equal to the initial rolling angle of the controller.

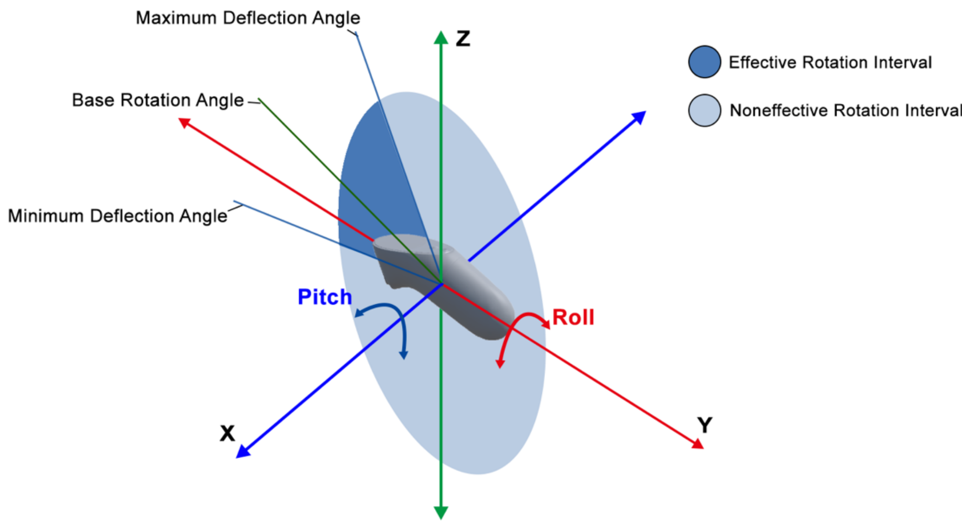

Fig. 2. The concepts of "roll", "pitch", "effective rotation (pitching) interval", "base rotation (pitching) angle", "maximum deflection angle", and "minimum deflection angle". (The controller's base rotation [rolling] angle and effective rotation [rolling] interval is not shown.)

The controller's effective rotation (pitching/rolling) interval was specified based on the base rotation (pitching/rolling) angle (See Fig. 2). In the design stage, we found that people's wrists were less accustomed to pitching downwards than raising upwards. Hence, in our case, the lower limit of the effective pitching interval was set closer to the base pitching angle than the higher limit of the effective pitching interval (See Fig. 3). On the other hand, the lower limit and the higher limit of the effective rolling interval were set symmetrically on both sides of the base rolling angle (See Fig. 4).

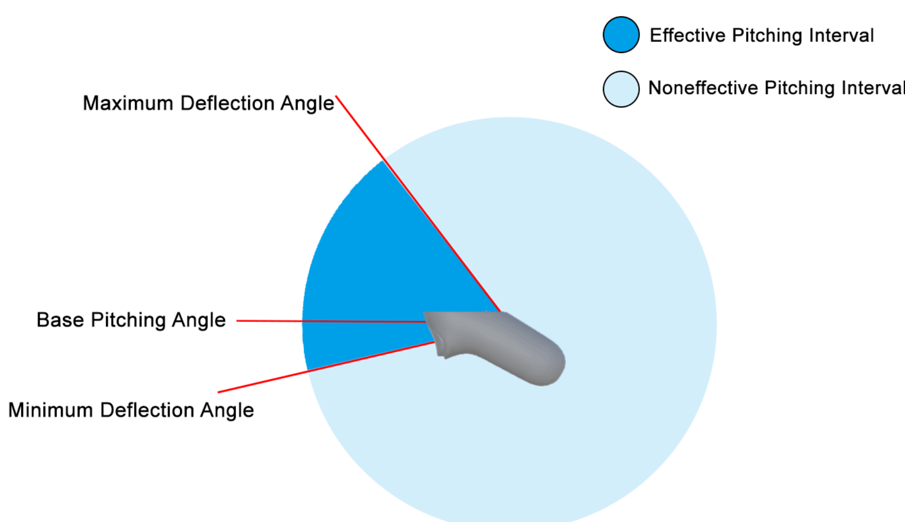

Fig. 3. The distribution of effective pitching interval, noneffective pitching interval, base pitching angle, minimum deflection (pitching) angle (the lower limit of the effective pitching interval), and maximum deflection (pitching) angle (the higher limit of the effective pitching interval)

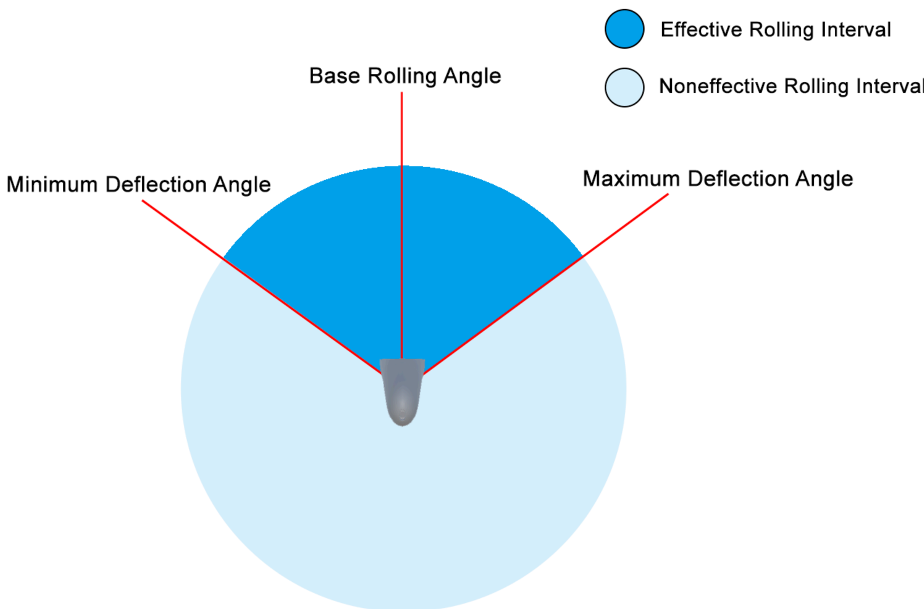

Fig. 4. The distribution of effective rolling interval, noneffective rolling interval, base rolling angle, minimum deflection (rolling) angle (the lower limit of the effective rolling interval), and maximum deflection (rolling) angle (the higher limit of the effective rolling interval)

By dividing the effective rotation interval by the number of rows or the number of columns of the QWERTY keyboard layout, a specific rotation interval was allocated to each key. The key whose allocated rotation interval included the controller's current effective rotation angle would be highlighted. For the current effective rotation angle's value, if the controller's current rotation angle was greater than the maximum deflection angle of the pre-defined effective rotation interval, the maximum deflection angle would be accepted. Similarly, if the controller's current rotation angle was less than the minimum deflection angle, the latter would be accepted.

B. RK: "Rotate Key" Interaction Method

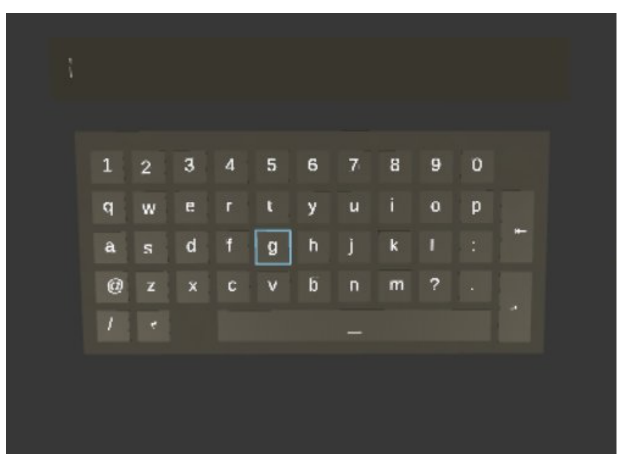

Fig. 5. RK: rolling the controller to move the cursor (light blue outline) left and right while pitching the controller to move the cursor up and down

RK (see Fig. 5) used a cursor for key selection on the virtual keyboard. The user rolled the VR controller to move the cursor horizontally while pitching the controller was to move the cursor vertically. The horizontal movement of the cursor was based on the controller's effective rolling angle. The cursor's vertical movement was detected based on the controller's current effective pitching angle. Its base pitching angle was captured on the input method's activation.

Once the user's posture changed, he/she could press a particular key on the controller to reset the base pitching angle to the controller's current pitching angle. This allowed the user to use the interaction method in any posture.

C. RCP: "Rotate Column Point" Interaction Method

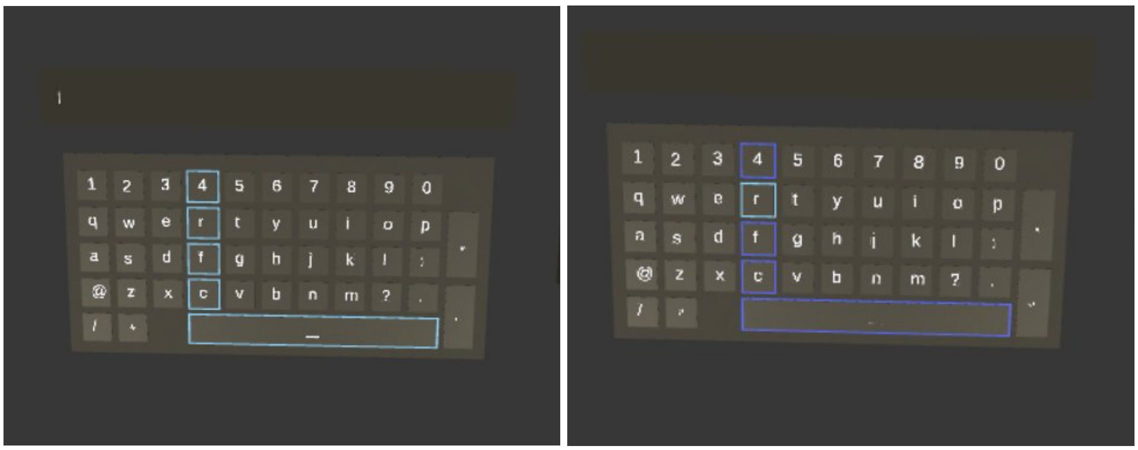

Fig. 6. RCP: (left) rolling the controller to switch the highlighted column (light blue outline), press the trigger button on the controller to lock the highlighted column; (right) Once a column is locked, pitching the controller to move the cursor (light blue outline) up and down within the locked column (dark blue outline)

RCP (see Fig. 6) separated the interaction in RK into two steps. The first step was column selection. In this step, a column of keys on the virtual keyboard would be highlighted at a time. The user could switch the highlighted column by rolling the controller. Like Rotate Key, the column-selecting cursor's movement was based on the controller's effective rolling angle. The user could press the trigger button on the controller to lock the highlighted column, which would lead the user to the second step – row selection. In this step, the user could move a key-selecting cursor vertically within the locked column by pitching the controller. The cursor's movement was based on the controller's current effective pitching angle. Its base pitching angle was captured on the row selection mode's activation. The user could press a particular key on the controller to quit the row selection mode.

D. RCR: "Rotate Column Rotate" Interaction Method

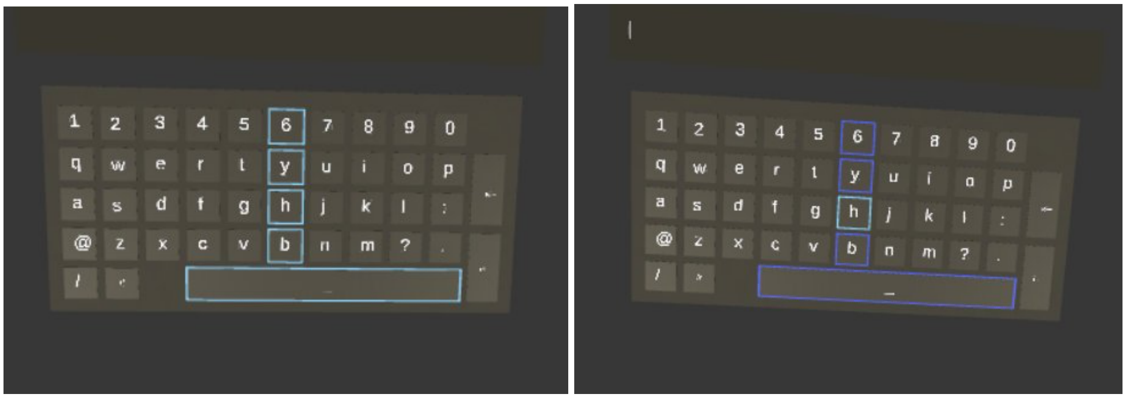

Fig. 7. RCR: (left) rolling the controller to switch the highlighted column (light blue outline), press the trigger button on the controller to lock the highlighted column; (right) Once a column is locked, rolling the controller to move the cursor (light blue outline) up and down within the locked column (dark blue outline)

RCR (see Fig. 7) also used a two-step interaction. Its interaction was very similar to the RCP. The significant difference between them was that the row selection of the RCR was roll-based, instead of pitch-based. In RCR's row selection mode, the user moved the key-selecting cursor within the locked column by rolling the controller. In RCR, both the horizontal movement and vertical movement of its keyboard cursor was based on the controller's effective rolling angle.

E. C: "Raycasting" Interaction Method

Fig. 8. Raycasting: moving the controller in space and casting the virtual ray to a particular key to highlight it

Raycasting (C) (see Fig. 8) was a mainstream VR text entry solution that implemented an aim-and-shoot style for the virtual keyboard interaction. For a keystroke, the user could cast the virtual ray, emitted from the top end of the controller, to a particular key on the keyboard.

When a particular key was highlighted, the final confirmation in these four text input methods to select the highlighted character would be pressing the controller's trigger button.

To eliminate potential deviation caused by other functionalities, neither auto-completion nor auto-correction functionalities were used for the VR text input techniques tested in this study. Moreover, no audible or haptic feedback was implemented for the keystrokes.

IV. EVALUATION STUDY

A comparative empirical evaluation was conducted to evaluate the text-entry rate, text-entry accuracy, system usability, user workload, and user experience of the four aforementioned VR text input methods.

A. *Study Design*

In this study, the within-subjects design was used with one independent variable (text entry method) consisting of four levels (experimental conditions): the three RotateEntry text input methods (RK, RCP, RCR) and the "raycasting-style" text input method (C). As the dependent variables, text-entry rate, text-entry accuracy, workload, usability, and user experience of the four VR text input methods were tested.

The text-entry rate of the study was measured based on the words-per-minute (WPM) metric presented by Wobbrock (Wobbrock 2007). The formula for computing words-per-minute was:

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

In the above formula, *T* was the final transcribed phrase inputted by the subject, and |*T*| was the length of the phrase. *S* denoted the time in seconds measured from the entry of the first character to the entry of the last one. In this case, the entry of the first character was not timed, hence "|*T*|-*1*" was used. The "*60"* denoted the number of seconds in each minute, and the "*5*" denoted characters per word (Wobbrock 2007).

Text-entry accuracy was measured based on the total error rate (total ER) metric presented by Soukoreff and MacKenzie (Soukoreff and MacKenzie 2003). The formula for computing total error rate was:

$$Total\ Error\ Rate = \frac{INF + IF}{C + INF + IF} \times 100\%$$

In the above formula, *INF* was the number of the *incorrect and not fixed* keystrokes, *IF* denoted the number of the *incorrect but fixed* keystrokes, and *C* denoted the number of the *correct* keystrokes (Soukoreff and MacKenzie 2003).

Besides, we used the following to test the user's subject experiences. A 10-item System Usability Scale (SUS) questionnaire (See Appendix III) was used to evaluate the subjective system usability of the text input techniques (Brooke 2013). To measure the user experience of the text input techniques, nine dimensions ("Competence", "Immersion", "Flow", "Tension", "Challenge", "Negative Affect", "Positive Affect", "Returning to Reality", and "Tiredness") were selected from the Game Experience Questionnaire (GEQ) In-game version and GEQ Post-game module following the GEQ scoring guidelines (IJsselsteijn, De Kort, and Poels 2013) (See Appendix III). And six dimensions ("Mental Demand", "Physical Demand", "Temporal Demand", "Performance", "Effort", and "Frustration") were selected from the NASA Task Load Index (NASA TLX) to assess a participant's task workload (Hart 2006) (See Appendix III).

As the main task, the participant was required to enter a phrase in a VR environment using the text entry method determined by the condition. Each condition presented five phrases selected from the phrase set used in Boletsis and Kongsvik's study. The selected phrases are shown in Appendix II. Each participant was presented with a randomized order of the conditions, and the phrases were presented in a randomized order for each condition. Therefore, each participant completed a total of 20 tasks (4x5).

B. Remote Study

In our original plan, the evaluation study would be held face-to-face. All the eligible participants would be randomly recruited. In the proposed research, an Oculus Go device with the evaluated demo installed and a 3DoF Oculus Go controller would be shared among all the participants. And during the proposed experiment session, the research moderator would help set up tasks on the test device, download the testing records of each condition from the test device to the computer, and hand out the materials, such as the Informed Consents and questionnaires, to the participants.

However, due to the COVID-19 situation, the study was switched to an online format and was conducted remotely. Therefore, there would be no physical contact between the research moderators and the subjects. For the remote testing control, input monitoring, and data recording, a password protected Node.js server was developed and deployed. A MongoDB database was implemented on the server-side to record the testing data. Socket.io technique was used to provide low latency network communication between the VR end and the server-side. In this case, the text input and the calculated WPM and total ER result emitted from the VR end would be captured by the server near real-time.

Furthermore, a dashboard web admin interface was developed using the HTML5 technique (see Fig. 9). Password was required to access the web page. Through the dashboard web page, the research moderator could set up the current tested text input method and the current phrase on the VR end, modify the server-side's settings and see the VR-end input data visualization. These additional developments helped the experimenter monitor the study progress efficiently and intervene in case of potential issues.

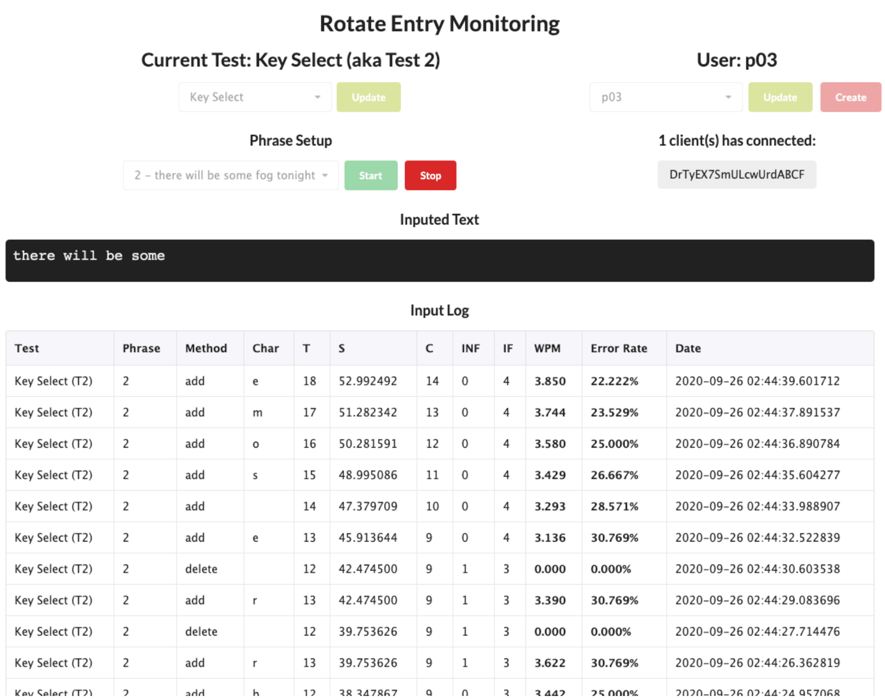

Fig. 9. Web admin interface

Since the evaluated demo application was developed for the Oculus Go VR device, only the subject who has an Oculus Go device would be recruited.

After the informed consent form was signed, an application package of the evaluated demo was built and sent to the participant. A unique four-digit identification code, which was used to identify the associated participant, was assigned to each application when it was built. The participant was asked to install the software on his/her Oculus Go device.

The experimental session was conducted on Zoom.us video conferencing platform. For the participant, a PC with a webcam, an Oculus Go device, and a mobile phone paired with the Oculus Go device were required. Before the study session, the participant was told to connect the devices mentioned above to the wireless Internet and join the specific Zoom.us meeting session from his or her PC and mobile phone. In the meantime, the moderator would register and activate the subject's identification code on the server-side (see Fig. 10). There can be only one activated identification code on the server-side at a time. And only the application corresponding to the activated identification code can connect to the server. In this way, the moderator can ensure that only the specified subject's VR device can communicate with the server during the study session.

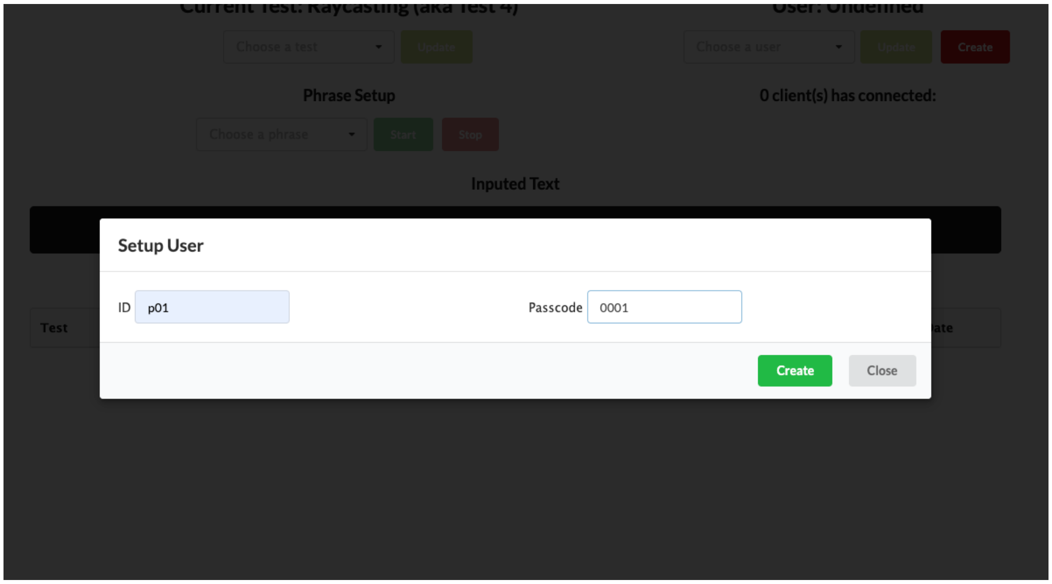

Fig. 10. Assigning a user ID with a 4-digit identification code (passcode) on the web admin interface

During the experimental session, the participant was suggested to turn on his or her webcam to help the research moderator identify whether the test device had been set up correctly. The participant was also told to cast the VR device's screen to his or her mobile phone using the Oculus mobile app's built-in functionality and share the mobile's screen to the Zoom.us meeting session. In this way, the research moderator could help the participant get familiar with the evaluated application and identify whether the test device could communicate with the server. Both video call and VR screen sharing were only used at the beginning of the study session. When the initialization check was completed, the participant can feel free to turn off the webcam or stop VR screen sharing to save the test device's power and the network bandwidth. However, the subject's Oculus Go device was required to stay connecting to the server during the entire study session. And the Zoom.us audio call was also essential.

Instead of using paper copies, all the consents and questionnaires used in this study had been converted to digital ones through the Smallpdf PDF e-sign platform or the Google Forms online survey platform. During the study session, the materials' e-copies were exported as links and were distributed via email or Zoom.us chatroom. Once the participant had filled out the distributed digital form, he or she could send the form back to the research moderator by clicking on the "send" button on the web interface.

See Appendix IV for more details of the remote study design.

C. Participants

Due to the COVID-19 situation, the size of the participant group was limited. The study procedures and protocols were approved by the university's Internal Review Board. A group of 5 people was recruited. The participants were selected through word-of-mouth based on their accessibility to an Oculus Go device. Among the 5 participants in the study, 2 were males, and 3 were females. Their ages ranged from 21 to 28 (Mean = 25.2, SD = 2.77). One of them used VR devices frequently, three of them rarely used VR devices, and one of them never used a VR device before. Those four participants who had used VR devices had only used 3DoF ones (Oculus Go or Gear VR), and they mainly used the devices for entertainment proposes (playing games, watching movies).

D. Procedure

Once recruited, the participants were provided with the Informed Consent form via email. Once a participant provided their consent, the study software was emailed to the participants with detailed installation instructions. The participant was free to install the software before or during the experimental session.

After filling out a demographic questionnaire during the study session (See Appendix I), each participant was asked to complete four experimental conditions. Before each task in the condition, a participant had 5 minutes of practice time to get familiar with the tested text input technique for the upcoming task. Then, in each condition, the participant was asked to enter five preselected phrases as fast and accurate as possible using one of the four VR text input methods. Each phrase was shown to the participant at a time and kept displaying on the user interface until the participant completed it. After completing each condition, the participant was told to fill in the SUS questionnaire, the GEQ questionnaire, and the NASA TLX questionnaire. There was a 5-minute break after each condition. The same procedure was used within the remaining VR text input methods. The participant was told to use the dominant hand to hold the controller and not switch the hand in use during the whole testing session. Both the order of the VR text input methods and the phrases' order were randomly organized for each participant. The character input from the VR controller, the WPM data, and the total ER data was monitored and recorded from the server-side during the test. After the participant completes all the tasks, a semi-structured interview was conducted to collect the participant's comments for each of the four evaluated VR text input methods. The experiment took approximately 90 minutes per participant to complete.

After the study session, the participant was provided with a digital copy of the payment form. The participant was told to fill out the form with his/her preferred way of electric cash transfer and his/her payment account and send it to the researcher via email. Once the researcher received the email, a $15 compensation would be sent to the participant based on the payment information he/she provided.

V. RESULT

A. Text Entry Rate

Table 1 and Figure 11 show the words-per-minutes of the four evaluated text input methods (higher is better). For each task in the study, the data of the first attempt was discarded. The results further analyzed the data using repeated measures ANOVA. A significant main effect was found on the Text Entry Method ($F[3, 12] = 73.769$, $p < 0.001$). Post-hoc comparisons showed significant differences between RCP and RK ($p = 0.005$), RCP and Raycasting ($p < 0.001$), RK and RCR ($p = 0.006$), RK and Raycasting ($p < 0.001$), and RCR and Raycasting ($p < 0.001$). No significant difference was revealed between RCP and RCR ($p = 0.772$).

VR Text Input Methods	*WPM Mean (SD)*	*WPM Range*
RCP	2.89 (0.74)	2.30 – 4.15
RK	5.51 (1.48)	3.06 – 6.88
RCR	3.08 (1.10)	1.83 – 4.76
Raycasting (C)	11.35 (0.56)	10.70 – 12.03

Table.1. Words-per-minutes (WPM) performance of the evaluated text input methods

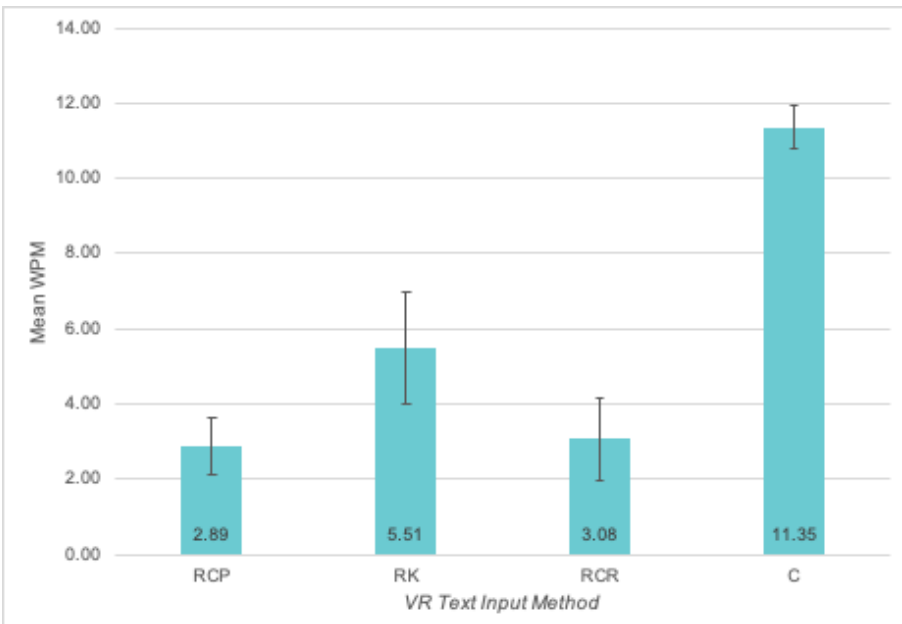

Fig. 11. Mean words-per-minutes (WPM) of the evaluated text input methods (higher is better). Error bars indicate standard deviation

B. Text Entry Accuracy

Table 2 and Figure 12 show the total error rate of the evaluated text input methods (lower is better). For each task in the study, the data of the first attempt was discarded. The results further analyzed the data using repeated measures ANOVA, which revealed a significant main effect ($F[3, 12] = 3.844$, $p = 0.039$).

VR Text Input Methods	*Total ER Mean (SD)*	*Total ER Range*
RCP	8.35% (4.48%)	4.23% - 16.01%
RK	8.99% (4.66%)	4.09% - 14.15%
RCR	7.76% (4.68%)	0.81% - 13.45%
Raycasting (C)	2.69% (2.19%)	0.00% - 5.12%

Table.2. Total Error Rate (total ER) of the evaluated text input methods

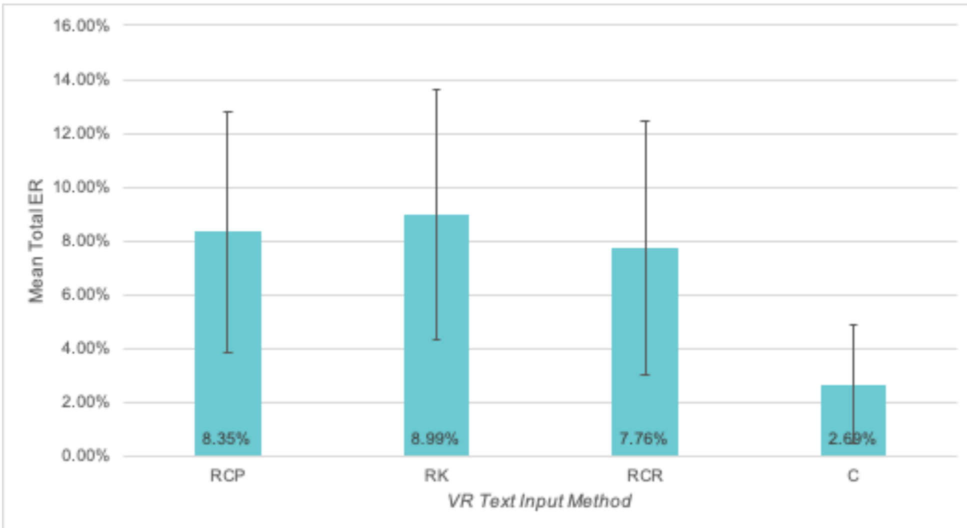

Fig. 12. Mean total Error Rate (total ER) of the evaluated text input methods (lower is better). Error bars indicate standard deviation

C. System Usability

Table 3 and Figure 13 show the SUS scores and the SUS ratings of the evaluated text input methods (higher is better). The SUS rating is obtained based on a 7-point adjective scale from Bangor and his team's work (Bangor, Kortum, and Miller 2009).

VR Text Input Methods	*SUS Mean (SD)*	*SUS Range*	*SUS Rating*
RCP	60 (30.10)	10 – 87.5	OK
RK	61 (10.55)	45 – 72.5	OK
RCR	64.5 (25.46)	32.5 – 100	OK
Raycasting (C)	87.5 (16.01)	70 – 100	Excellent

Table.3. The SUS scores and ratings of the evaluated text input methods

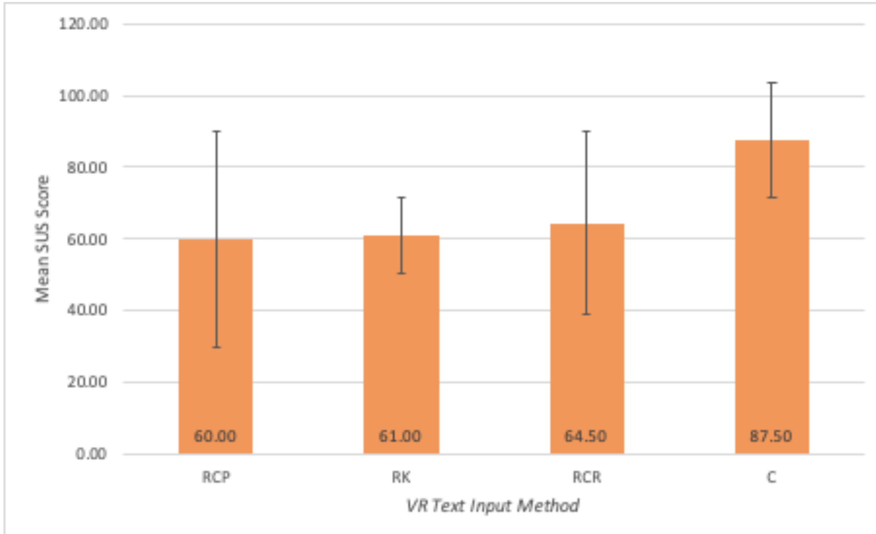

Fig. 13. The SUS scores of the evaluated text input methods. Error bars indicate standard deviation (higher is better)

D. User Experience

Table 4 and Figure 14 show the GEQ scores of the four evaluated text input methods across the nine dimensions. Each item value ranges from 0 ("not at all") to 4 ("extremely"). For the scores from the "Competence", "Immersion", "Flow", "Positive Affect" dimensions, the higher is better. For the scores from the "Tension", "Challenge", "Negative Affect", "Returning to Readlity", and "Tiredness" dimensions, the lower is better.

The Friedman Test indicates that there are no statistically significant differences in all the GEQ dimensions, except Tension ($X^2[3] = 8.333$, $p = 0.040$) and Negative Affect ($X^2[3] = 8.455$, $p = 0.037$).

GEQ Dimensions	*GEQ Mean (SD)*				*X^2 (p-value)*
	RCP	*RK*	*RCR*	*Raycasting*	
Competence	2.5 (1.27)	2.9 (0.96)	3.2 (0.76)	3.4 (0.65)	7.000 (0.072)
Immersion	2.4 (1.67)	3 (0.71)	2.2 (1.30)	3.2 (1.10)	1.923 (0.589)
Flow	2.6 (0.89)	3 (0.94)	2.9 (1.14)	2.1 (0.89)	4.674 (0.197)
Tension	2 (1.27)	1.7 (1.40)	2 (1.58)	0.5 (1.12)	8.333 (0.040)
Challenge	2.6 (0.96)	2.5 (1.06)	2.4 (0.65)	1 (1.00)	7.625 (0.054)
Negative Affect	2.1 (1.43)	1.5 (1.37)	2 (1.37)	0.4 (0.89)	8.455 (0.037)
Positive Affect	2.6 (1.56)	3.1 (0.42)	2.7 (0.67)	3.3 (0.76)	3.000 (0.392)
Returning to Reality	1.1 (0.74)	0.8 (1.10)	0.7 (0.76)	0.4 (0.55)	6.818 (0.078)
Tiredness	2 (1.58)	1.2 (0.84)	1.7 (1.48)	0.4 (0.89)	7.769 (0.051)

Table.4. GEQ scores across the 9 GEQ dimensions of the evaluated text input methods

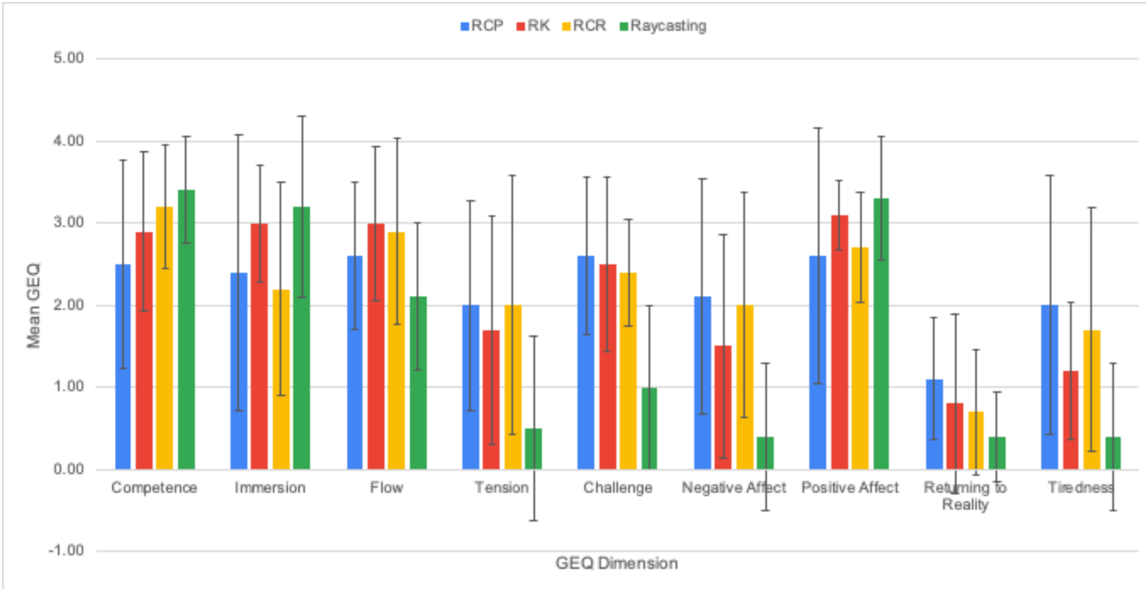

Fig. 14. GEQ scores of the evaluated text input methods (For the scores from the "Competence", "Immersion", "Flow", "Positive Affect" dimensions, the higher is better. For the scores from the "Tension", "Challenge", "Negative Affect", "Returning to Readlity", and "Tiredness" dimensions, the lower is better.). Error bars indicate standard deviation

E. Workload

Table 5 and Figure 15 show the scores of the four evaluated text input methods across the 6 NASA TLX dimensions. The value of each item ranges from 1 ("very low") to 7 ("very high"). In this case, for the scores from all dimensions, except for "Performance", the lower is better. For the scores from the "Performance" dimension, the higher is better.

The Friedman Test indicates that there are no statistically significant differences in all NASA TLX dimensions, except "Physical Demand", which $X^2(3) = 8.500$, $p = 0.037$.

NASA TLX Dimensions	*NASA TLX Mean (SD)*				*X^2 (p-value)*
	RCP	*RK*	*RCR*	*Raycasting*	
Mental Demand	5 (1.87)	3.6 (2.19)	4.8 (1.64)	2.4 (1.52)	6.391 (0.094)
Physical Demand	5.6 (2.07)	5 (1.87)	5.4 (1.95)	2.4 (1.95)	8.500 (0.037)
Temporal Demand	3.8 (2.17)	4.6 (1.52)	3.8 (1.79)	3 (1.87)	1.884 (0.597)
Performance	5.6 (1.14)	6.4 (0.89)	6 (1.41)	6.6 (0.89)	5.625 (0.131)
Effort	5 (1.87)	4.8 (1.64)	5.4 (1.52)	2.8 (2.17)	4.256 (0.235)
Frustration	4.6 (1.95)	3.8 (2.39)	4 (2.00)	1.8 (1.79)	7.786 (0.051)

Table.5. Scores across the 6 NASA TLX dimensions of the evaluated text input methods (for the scores from all dimensions, except for "Performance", the lower is better. For the scores from the "Performance" dimension, the higher is better)

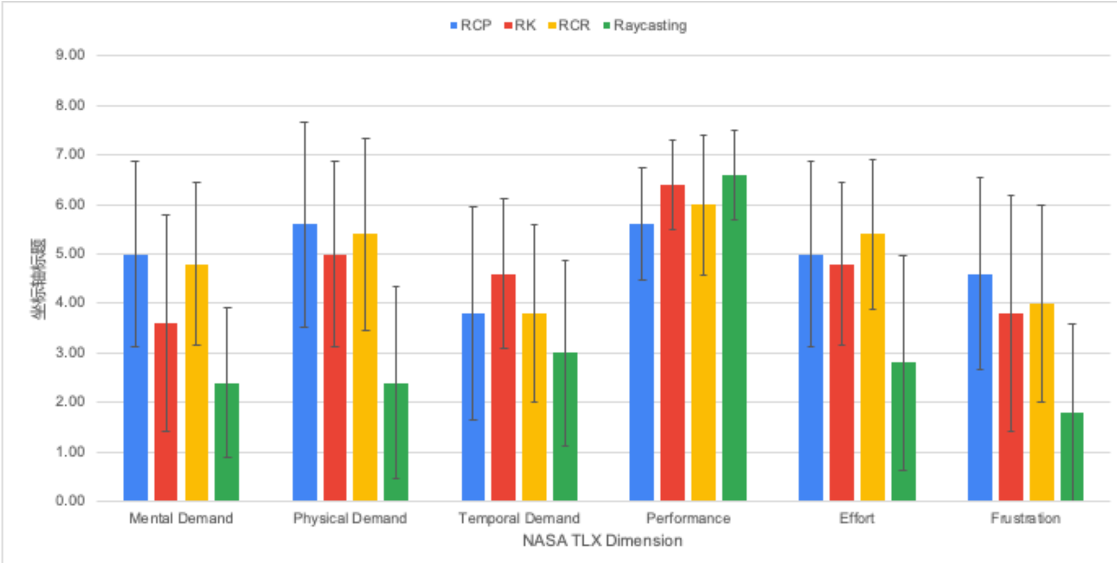

Fig. 15. NASA TLX scores of the evaluated text input methods. Error bars indicate standard deviation

F. Interview Comment

Table 6 shows the comments collected from the semi-structured interview sessions with the 5 participants. The comments under each condition are ranked according to the number of participants who had mentioned it. Each comment has been labeled as positive ("P"), negative ("N"), or neutral ("-"), based on its outcome.

	Interview Comment	*Count*
	RCP	
N	The cursor was sensitive.	3
N	Column locking was complex.	2
N	It was complex. Because it had two movements involving the rotation and then you had to do up and down also.	1
P	It had a good pace.	1
N	It was a bit tiring.	1
P	The typing was getting very easier because I could select the columns and then rows, instead of just going to select all the alphabet and searching for everything.	1
P	It has two distinct interaction patterns.	1
N	it was a little difficult for me to start. When I did not know it was rotating the wrist left and right.	1
	RK	
N	When I have to move right left, it is fine, but if I had to select the rows by going up and down, that is difficult.	2
N	It was a bit tiring.	2
N	It is kind of confusing. It was like moving on its own. I felt like I did not have control, but actually, I did.	2

P	It was easy to use. It needs little effort. I just have to tweak and move up and down.	2
N	The cursor was too sensitive for me.	1
P	It was intuitive.	1
N	When my cursor was moved beyond the delete button, I was lost because I could not see where my cursor was.	1
RCR		
N	I had to rotate around to go down to the bottom row. It was difficult.	2
N	The column locking makes it complex to use.	2
N	The cursor is sensitive.	2
P	Since its control was rotation only, I felt confident to do it.	1
-	Some letters (e, t) were very easy to locate, while some were hard (space, delete).	1
P	I could just get my wrist on the table stiff and then move.	1
Raycasting		
P	It was intuitive. I did not have to use multiple clicks. I just had to point directly to what I want.	4
N	It was effortful to use. I had to point at the keyboard, and then to the exact key.	1
P	It was comfortable to use.	1
P	I knew where exactly a key was at the keyboard, so I was effortless to type quickly.	1

**P: Positive comment, N: Negative comment, -: Neutral comment*

Table.6. Comments collected from the 5 participants in the interview sessions

VI. DISCUSSION

A. Text Entry Method Study

The study result indicates that Rotate Key (RK) has the highest text entry rate among the three variants of RotateEntry, with 5.51 WPM. There are no significant differences in the text entry rate between Rotate Column Point (RCP) and Rotate Column Rotate (RCR), with 2.89 WPM and 3.08 WPM respectively. The three variants' differences might be attributed to RK using a one-step control while the other two use two-step ones. For the two using the two-step control, there would be extra time consumption in switching between the two modes of moving the cursor horizontally and vertically, which may result in a decrease in input rate. However, in terms of text entry performance, Raycasting outperforms the other three evaluated text input methods, with 11.35 WPM. 80% of the participants indicates that selecting a character using Raycasting is intuitive, which just needs to aim at a particular key on the virtual keyboard. While 40% of participants indicate the "column locking" mechanism on RCP and RCR is complex, and 40% of participants point out that moving the cursor on RK is difficult. Only 20% of the participants suggest that the interaction on RK is intuitive.

As for the accuracy of text entry, the total error rate of RCR is 7.76%, which is the lowest compared to RCP (8.35%) and RK (8.99%). Raycasting yields the lowest total error rate (2.69%) among the evaluated text input methods. As indicated by some of the participants (RCP: 60%, RK: 20%, RCR: 40%), the key-selecting cursors on all of the RotateEntry text input methods are sensitive, which may result in the high total error rate of RCR, RCP, and RK. This issue may be due to the cursor control algorithm of RotateEntry lacked the noise-reducing process for the input signal.

In terms of the subjective system usability in the evaluation, Raycasting yields the best SUS rating ("Excellent"), while the ratings of the three RotateEntry text input methods are all "OK". Among the three RotateEntry text input methods, RCR has a slightly higher mean SUS score (64.5), compared to RCP (60) and RK (61). It is worth mentioning that RCP and RCR have significantly higher standard deviations in their SUS scores (RCP: 30.10, RCR: 25.46), compared to RK (10.55) and Raycasting (16.01). It indicates that, compared to RK and Raycasting, the subjective usability of RCP and RCR varies significantly among the 5 participants.

When it comes to the user experience, Raycasting shows the best GEQ performance among the four text input methods on all dimensions except Flow, which supports its high SUS rating ("Excellent") and the high proportion (80%) of user comments which consider it is intuitive. Among the three RotateEntry text input methods, RK has the highest GEQ performance on 6 GEQ dimensions, including Immersion, Flow, Tension, Negative Affect, Positive Affective, and Tiredness. What is more, on the Flow dimension, RK's performance surpasses all the other three evaluated text input methods, which may explain the participant comments that indicate it is easy to use (40%) and intuitive (20%). RCR and RCP have similar GEQ performance. However, RCR mildly outperforms all of the GEQ dimensions except Immersion and Tension, compared to the latter. It may support the higher mean SUS score of RCR (64.5) than RCP (60).

Raycasting has superior workload performance. Its scores are the best on all six NASA TLX dimensions among the evaluated text input methods. Moreover, it has outstanding performance on Physical Demand (2.4), Effort (2.8), and Frustration (1.8) compared to the other three techniques.

The above evidence may support its remarkable performance on the GEQ dimension of Tension, Challenge, and Tiredness. Among the three RotateEntry text input methods, RK marks the best scores on all NASA TLX dimensions, except Temporal Demand (4.6). On the Temporal Demand dimension, RK is also the worst performer among the evaluated text input methods. The workload performance of RCP and RCR is similar. Nevertheless, RCR has slightly higher performance on all dimensions compared to RCP, except Performance and Frustration.

Overall, in this evaluation study, Raycasting has achieved the better performance. It surpassed the three RotateEntry text input methods on all dimensions, except GEQ Flow. However, it should be noted that four out of our five participants had frequent experience with VR and using the standard Raycasting method. In this first study, we compared Raycasting only as a baseline. Nevertheless, 1 of the 5 participants (20%) indicates that *"Raycasting was effortful to use. I had to point at the keyboard, and then to the exact key"*, while in terms of RK, one of the variants of RotateEntry, the participant indicates that *"RK needs little effort. I just have to tweak and move up and down"*. It may provide evidence that RotateEntry has potential advantages on the accessibility aspect.

Among the three RotateEntry text input methods, RK yields a high text entry rate, outstanding overall performance on the GEQ dimensions, and the NASA TLX dimensions. However, meanwhile, it has the lowest text entry accuracy and a relatively low SUS score. Some participants (40%) argue that its cursor control is confusing, and some of the participants (40%) also find the cursor control is difficult when it comes to row selecting. Moreover, it may have a usability issue that the cursor would become invisible when it moves to a blank space as stated by one participant.

RCP and RCR have similar performance. However, compared to the former one, RCR exhibits higher text entry rate and accuracy, a higher SUS score, and better overall performance on both of the GEQ dimensions and the NASA TLX dimensions. 40% of the participants indicate that the column locking mechanism on these text input techniques is problematic. Other than that, the comments on their controls are various. Some participants may prefer RCP's control *("it has two distinct interaction patterns"*), while some may prefer RCR ones (*"since its control was rotation only, I felt confident to do it*").

B. Remote Study

The evaluated application package and a digital copy of installation instructions were usually emailed to the participants two days in advance to offer sufficient preparation time. The instructions were step-by-step and with marked screenshots on each step, aiming to provide a user-friendly experience on the app installation. However, an Oculus developer account was required for the installation. The participant who was a non-Oculus developer might face issues with registering the account. In the circumstances, the participant was told to feel free to contact the research moderator for help via email. The moderator would provide timely feedback based on the situation. Besides, subjects can also choose to install the app during testing. In this case, the subject would be informed to schedule with the moderator and enter the test session for at least half an hour in advance to ensure that the experiment could be completed on time.

With the WebSocket connection used by the Socket.io, the input data from the VR side was collected and monitored near real-time. Therefore, potential issues on the VR side could be identified in time. Since the WPM and total ER were calculated on the VR side, it eliminated the impact of the potential network delay on the accuracy of the testing results. However, the network latency and unstable network connection may cause the server-side to fail to receive some input data from the VR side during the experimental session. In our case, we informed every participant to stop performing the task as soon as the network connection indicator on the left side of the interface (see Fig. 1) was displaying “disconnected”, which might reduce the possible impact (wasted time, increased fatigue, etc.).

Observing the participant’s actions was a significant problem for this remote study. In the experimental session, we used Zoom.us video conferencing platform to communicate with the participants. However, considering multiple devices (VR devices, mobile phones, PCs, etc.) required real-time network communication during the test, video calls that last the entire session were not encouraged, aiming to save network bandwidth. What is more, it was difficult for the research moderator to observe the subjects’ hand movements due to the webcam’s limited shooting angle. To prevent the test results from being disturbed, it was prohibited to require the subjects to use postures that were easier to be observed (for example, put the hand holding the controller on the table). Therefore, it was almost impossible to learn directly whether the subject was using the controller in an anticipated way. We could understand the user’s situation in three ways: verbally inquiring the participant, monitoring the participant’s input data, and watching the participant’s VR screen sharing. In our case, before the study session, the moderator would instruct the subject on using the controller through the video call. Every participant was informed to keep using his/her dominant hand for the entire test to eliminate possible bias. After the experiment session, the semi-structured interview also allowed us to learn more participants’ usage posture details.

The test environment setting of the subject could also be an issue. Before the study session, we informed the subjects that they needed to perform the tasks in a quiet environment and keep their phones muted. However, it was inevitable that the participant might get disturbed by the surroundings. We occasionally found that people who were not related to the experiment were walking and talking in the background during the test. Since mobile phones and computers were used during the study session, subjects would sometimes be disturbed by notifications that appeared on the screen. Besides, while the VR screen sharing was conducted through the phone, the screen sharing would often be blocked by calls or phone display turning off. And the screen sharing would be interrupted because of the unstable connection between the VR device and the mobile phone. When screen sharing interruption occurred, we would tell the participants to continue with the task. Since we collected the user input data via the communication between the VR end and server-side, the screen sharing interruption would not affect the testing process.

C. Study Limitations

Due to the COVID-19 situation, the evaluation only had 5 participants recruited. Although the general difference among the evaluated text input techniques has been investigated, the small sample group may result in reducing the potential to reflect the statistically significant effects on some of the dimensions (Button et al. 2013).

As stated above, the experiment duration for each participant was around 90 minutes. As a result of the tight schedule, each participant only had approximately 5-minute practice time to get familiar with the text input method evaluated in the following condition. Since the tested device's built-in text input technique was raycasting-style, the participant may be potentially an expert user on Raycasting. At the same time, he/she may be a novice user on the three proposed RotateEntry text input techniques. It may bias the performance and experience result of the evaluated text input techniques.

D. Future Work

Firstly, as indicated from the discussion above, a noise reduction algorithm for the input signal would be embedded for the three proposed RotateEntry text input techniques to improve their text-entry accuracy, usability, and user experience performance.

Secondly, unlike Raycasting, the three proposed RotateEntry text input techniques make no demand on users to aim at a key so that future study may focus on the effect of the keyboard size and the keyboard position of RotateEntry. Furthermore, instead of the standard QWERTY keyboard layout implemented in this study, a customized keyboard layout potentially more suitable for RotateEntry would also be investigated.

Finally, for the future experiment design, a larger sample size group and a long-term experiment session would be implemented.

VII. CONCLUSION

In this study, we proposed RotateEntry, a text-entry interaction that moved the key-selecting cursor on a virtual keyboard by rolling a 3DoF VR controller. Based on the concept of RotateEntry, three controller-rolling-style text input methods were developed: Rotate Key, Rotate Column Point and Rotate Column Rotate. A comparative empirical evaluation with 5 participants was conducted to test and evaluate the text-entry rate, text-entry accuracy, system usability, user experience, and workload of the three RotateEntry text input methods and a standard Raycasting text input method. By analyzing the evaluation results, we identified that Rotate Key had an outstanding text input speed, higher overall user experience scores, and excellent overall workload performance, compared to the other two RotateEntry text input techniques. However, no evidence was identified to support that RotateEntry had better performance and experience compared to Raycasting.

VIII. ACKNOWLEDGMENTS

We would like to thank all the subjects in this study for their voluntary participation and feedback, which help us identify the usability issues on the evaluated text input methods.

IX. REFERENCES

Bangor, Aaron, Philip T. Kortum, and J. T. Miller. "Determining what individual SUS scores mean: adding an adjective rating scale." *Journal of Usability Studies archive* 4 (2009): 114-123.

Benligiray, Burak, Cihan Topal, and Cuneyt Akinlar. "SliceType: Fast Gaze Typing with a Merging Keyboard." *Journal on Multimodal User Interfaces* 13, no. 4 (2018): 321–34. https://doi.org/10.1007/s12193-018-0285-z.

Boletsis, Costas, and Stian Kongsvik. "Controller-Based Text-Input Techniques for Virtual Reality: An Empirical Comparison." *International Journal of Virtual Reality* 19, no. 3 (2019). https://doi.org/10.20870/ijvr.2019.19.3.2917.

Boletsis, Costas, and Stian Kongsvik. "Text Input in Virtual Reality: A Preliminary Evaluation of the Drum-Like VR Keyboard." *Technologies* 7, no. 2 (2019): 31. https://doi.org/10.3390/technologies7020031.

Brooke, J. "SUS: a retrospective." *Journal of Usability Studies archive* 8 (2013): 29-40.

Button, Katherine S., John P. A. Ioannidis, Claire Mokrysz, Brian A. Nosek, Jonathan Flint, Emma S. J. Robinson, and Marcus R. Munafò. "Power Failure: Why Small Sample Size Undermines The Reliability Of Neuroscience". *Nature Reviews Neuroscience* 14 (5): 365-376. doi:10.1038/nrn3475.

Dube, Tafadzwa Joseph, and Ahmed Sabbir Arif. "Text Entry in Virtual Reality: A Comprehensive Review of the Literature." *Human-Computer Interaction. Recognition and Interaction Technologies Lecture Notes in Computer Science*, 2019, 419–37. https://doi.org/10.1007/978-3-030-22643-5_33.

Grubert, Jens, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. "Text Entry in Immersive Head-Mounted Display-Based Virtual Reality Using Standard Keyboards." *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2018. https://doi.org/10.1109/vr.2018.8446059.

Hart, Sandra G. "Nasa-Task Load Index (NASA-TLX); 20 Years Later." *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50, no. 9 (2006): 904–8. https://doi.org/10.1177/154193120605000909.

IJsselsteijn, W., de Yaw Yvonne Kort and K. Poels. "The Game Experience Questionnaire." (2013).

Seibert, Jonmichael, and Daniel M. Shafer. "Control Mapping in Virtual Reality: Effects on Spatial Presence and Controller Naturalness." *Virtual Reality* 22, no. 1 (2017): 79–88. https://doi.org/10.1007/s10055-017-0316-1.

Soukoreff, R. William, and I. Scott Mackenzie. "Metrics for Text Entry Research." *Proceedings of the conference on Human factors in computing systems - CHI '03*, 2003. https://doi.org/10.1145/642611.642632.

Wobbrock, Jacob O. “Measures of Text Entry Performance.” *Text Entry Systems*, 2007, 47–74. https://doi.org/10.1016/b978-012373591-1/50003-6.

Xu, Wenge, Hai-Ning Liang, Yuxuan Zhao, Tianyu Zhang, Difeng Yu, and Diego Monteiro. “RingText: Dwell-Free and Hands-Free Text Entry for Mobile Head-Mounted Displays Using Head Motions.” *IEEE Transactions on Visualization and Computer Graphics* 25, no. 5 (2019): 1991–2001. https://doi.org/10.1109/tvcg.2019.2898736.

Yu, Difeng, Kaixuan Fan, Heng Zhang, Diego Monteiro, Wenge Xu, and Hai-Ning Liang. “PizzaText: Text Entry for Virtual Reality Systems Using Dual Thumbsticks.” *IEEE Transactions on Visualization and Computer Graphics* 24, no. 11 (2018): 2927–35. https://doi.org/10.1109/tvcg.2018.2868581.

X. APPENDIX I: PRE-SESSION QUESTIONNAIRE

1. Your assigned number * (Ask the investigator)
2. Gender *
   a) Female
   b) Male
   c) Prefer not to say
   d) Other:
3. Age *
4. Do you wear glasses? *
   a) Yes
   b) No
   c) Prefer not to say
   d) Other:
5. Frequency of VR use *
   a) Never
   b) Rarely
   c) Frequently
   d) Everyday
6. Have you ever used a VR headset? *
   a) Yes *(Skip to question 7)*
   b) No *(Skip to the end of the questionnaire)*
7. What type of VR headsets have you used? * (Can you tell its name? [e.g. Oculus Go, HTC Vive, etc.])
8. How long have you used a VR headset? *
9. What would you mainly use a VR headset for? *
10. Have you used any text entry methods on the VR headset? *
    a) Yes *(Skip to question 11)*
    b) No *(Skip to question 15)*
11. What is/are the text entry method(s) you have used on a VR headset? * (e.g. Built-in text entry method/Ray-casting etc. If you cannot tell its name, try to describe it.)
12. What kinds of scenarios did you use the text entry method for? * (e.g. Social [texting], Searching, etc.)
13. Among the text entry methods you used, do you prefer one? What is it?
14. Can you share your experience of text entry on a VR headset? *(Skip to the end of the questionnaire)*
15. Can you tell us the reason why you don't use text entry methods on the VR headset?

** Required*

XI. Appendix II: Phrases used in the study

Phrase	*Length (Character)*
my preferred treat is chocolate	31
question that must be answered	30
there will be some fog tonight	30
physics and chemistry are hard	30
we are subjects and must obey	29

XII. APPENDIX III: POST-TASK QUESTIONNAIRE

1. What is your assigned number? * *(Ask the investigator)*
2. What is the current task number? * *(Ask the investigator)*
3. Mental Demand * *(How mentally demanding was the task?)*

	1	2	3	4	5	6	7	
Very Low	◯	◯	◯	◯	◯	◯	◯	Very High

4. Physical Demand * *(How physically demanding was the task?)*

	1	2	3	4	5	6	7	
Very Low	◯	◯	◯	◯	◯	◯	◯	Very High

5. Temporal Demand * *(How hurried or rushed was the pace of the task?)*

	1	2	3	4	5	6	7	
Very Low	◯	◯	◯	◯	◯	◯	◯	Very High

6. Performance * *(How successful were you in accomplishing what you were asked to do?)*

	1	2	3	4	5	6	7	
Very Low	◯	◯	◯	◯	◯	◯	◯	Very High

7. Effort * *(How hard did you have to work to accomplish your level of performance?)*

	1	2	3	4	5	6	7	
Very Low	◯	◯	◯	◯	◯	◯	◯	Very High

8. Frustration * *(How insecure, discouraged, irritated, stressed, and annoyed were you?)*

	1	2	3	4	5	6	7	
Very Low	◯	◯	◯	◯	◯	◯	◯	Very High

9. I felt successful *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

10. I felt bored *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

11. I found it impressive *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

12. I forgot everything around me *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

13. I felt frustrated *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

14. I found it tiresome *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

15. I felt skillful *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

16. I felt completely absorbed *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

17. I felt content *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

18. I felt challenged *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

19. I had to put a lot of effort into it *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

20. I felt good *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

21. I found it hard to get back to reality *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

22. I felt disoriented *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

23. I felt exhausted *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

24. I felt weary *

	0	1	2	3	4	
Not at all	◯	◯	◯	◯	◯	Extremely

25. I think that I would like to use this system frequently *

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

26. I found the system unnecessarily complex *

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

27. I thought the system was easy to use *

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

28. I think that I would need the support of a technical person to be able to use this system *

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

29. I found the various functions in this system were well-integrated *

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

30. I thought there was too much inconsistency in this system *

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

31. I would imagine that most people would learn to use this system very quickly *

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

32. I found the system very cumbersome to use *

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

33. I felt very confident using the system *

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

34. I needed to learn a lot of things before I could get going with this system *

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

** Required*

Note:

NASA TLX Questionnaire: Question 3 – 8

GEQ questionnaire: Question 9 – 24

SUS questionnaire: Question 25 – 34

XIII. APPENDIX IV: A CASE STUDY OF ADAPTING A VIRTUAL REALITY USER STUDY TO AN ONLINE FORMAT DURING COVID-19

(Please find the document on the next page.)

A Case Study of Adapting a Virtual Reality User Study to an Online Format During COVID-19

ZIMING LI, School of Information Rochester Institute of Technology, USA
ROSHAN PEIRIS*, School of Information Rochester Institute of Technology, USA

The recent COVID-19 pandemic resulted in many challenges or complete cessation of conducting typical face-to-face user studies to avoid any physical contact between researchers, participants, and devices. Due to this, especially virtual reality (VR) research studies have faced many limitations recently. To adapt to this situation, we attempted to convert our entire user study for a VR based text entry method to an online format as a case study. During this process, we identified few key challenges that may arise in relation to the test environment setup, data collection, participant observations, experimental environment control and participant recruitment. As such, in this paper, we report our methods of adaption, experiences, challenges, and potential solutions that we identified during this process.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: online study, virtual reality, COVID-19, head-mounted display, WebSocket, text input method

ACM Reference Format:
Ziming Li and Roshan Peiris. 2018. A Case Study of Adapting a Virtual Reality User Study to an Online Format During COVID-19. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The COVID-19 pandemic has presented several challenges in the academic sector due to the requirement to limit face-to-face interactions and remote working conditions [1, 5, 12]. This situation is especially challenging for human-computer interaction research due to the difficulty or inability to conduct face-to-face user studies [11] that are especially important in research areas such as wearable computing and tangible interfaces. Virtual reality (VR) research has faced a similar plight due to the wearable nature of the head-mounted displays that require contact between the device and the user. Similarly, the COVID-19 pandemic created several challenges to our study on a new virtual reality (VR) text input method. Here too, the main challenge was that as per the new guidelines issued by our Institution's Review board[1], we could not conduct face-to-face experiments as we had planned mainly due to the inability to use the same VR equipment that contained the study software setup and the inability to use our controlled laboratory environment in order to limit physical contact. Hence, we developed an online system that allowed us to switch the entire study to an online format. Thus, in this paper, we explore our research and its adaption to the online format for user studies

[1] https://www.rit.edu/research/hsro/

as a case study. Our case study reports our experience, especially the challenges we faced while adapting to present circumstances to conduct an online evaluation study with VR head-mounted displays (VR HMDs).

Our study investigated the VR text input method called "RotateEntry". It was based on a "controller-rolling-style" text entry interaction for three degrees of freedom (3DoF) VR HMDs. Based on the RotateEntry concept, we developed three interaction methods named Rotate Column Rotate, Rotate Key, and Rotate Column Point, aiming to look at how to move the cursor across the standard virtual QWERTY keyboard. Here, we intended to evaluate these three interaction methods and the traditional raycasting method in a comparative empirical study. As the dependent variables, text-entry rate, text-entry accuracy, workload, usability, and user experience of the four VR text input methods were tested. Also, participants' comments on the evaluated text input methods were collected by questionnaires and brief discussions.

However, as we could not conduct any face-to-face studies due to the COVID-19 pandemic, we shifted our entire study procedure to an online format. Previous research had explored conducting Presence and Immersive Experiences based user studies for VR systems using consumer VR equipment [14] and, recent articles discussed potential best practices for such efforts, especially during the COVID-19 pandemic [15]. However, we were presented with additional challenges due to the sudden closure of our institution premises and the nature of our study that involved the use of the VR controllers. Thus, this paper will mainly share our experience in conducting online research with VR HMDs, including the challenges we encountered during the study sessions and our proposed solutions for these challenges.

2 IMPLEMENTATION OF ROTATEENTRY

The application of RotateEntry was developed on the Unity game engine[2] and was deployed on the Oculus Go[3] 3DoF VR HMD. It used a QWERTY keyboard layout (see Figure 1) to reduce users' learning-curve since the same keyboard layout was implemented on the default interaction method for most VR HMDs [6].

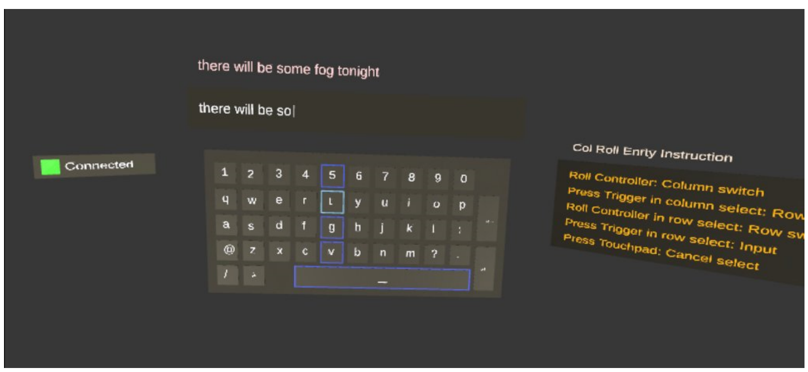

Fig. 1. A screenshot of the evaluated demo's user interface

The implementation of RotateEntry was to capture the controller's rotational input of a particular axis. By dividing the effective rotation interval (See Figure 2) by the number of rows or the number of columns of the QWERTY keyboard layout, a specific rotation interval was allocated to each key. The key whose given rotation interval included the controller's current effective rotation angle would be highlighted.

[2]https://unity.com
[3]https://www.oculus.com/go/

The demo application of RotateEntry provided three interaction methods with different ways of moving the cursor across the virtual keyboard: Rotate Key (RK), Rotate Column Point (RCP), and Rotate Column Rotate (RCR).

(1) RK used a cursor for key selection on the virtual keyboard. The user could roll the VR controller to move the cursor among columns. And they could pitch the controller to move the cursor among rows.
(2) RCP used a two-step interaction. Firstly, the user could roll the controller to move the cursor among columns. The user could press a key to select a column and then move the key-selecting cursor within the selected column by pitching the controller.
(3) RCR also used a two-step interaction, similar to RCP. The user could move the cursor among columns by rolling the controller. After pressing a key to select a column, the user could then move the key-selecting cursor within the chosen column by rolling the controller again.

Additionally, the demo application contained a standard raycasting-style interaction method, named Raycasting. Raycasting used an aim-and-shoot style for the key selecting interaction. The user could cast the virtual ray, emitted from the controller's top end, to a particular key on the virtual keyboard for a keystroke.

Finally, when a particular key was highlighted in any of the four interaction methods mentioned above, the user could press the controller's trigger button to select the focused character.

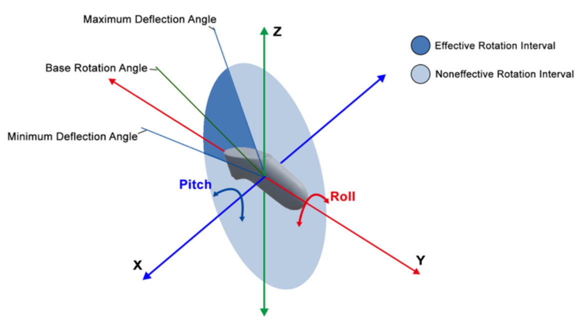

Fig. 2. The concepts of "roll", "pitch", "effective rotation (pitching) interval" for RotateEntry

3 PRELIMINARY STUDY: IDENTIFYING CHALLENGES

In our original plan, the evaluation study was to be held face-to-face. All the eligible participants would be randomly recruited. An Oculus Go HMD with the evaluation software installed would be shared among all the proposed research participants. During the proposed experiment session, the research moderator would help set up tasks on the test device, download the testing records of each condition from the test device to the computer, and hand out the materials, such as the Informed Consents and questionnaires, to the participants.

Our research was in preparation to start the user study phase when the COVID-19 pandemic resulted in the closure of institution premises for physical access (in March 2020). At that time, although the Institutional Review Board had approved our study plans and protocols, the same entity issued specific new guidelines to cease all on-going or planned face-to-face user studies. Therefore, upon further discussion, we decided to switch the entire research into an online format.

To identify the potential challenges we might face with an online VR study, a preliminary study was conducted remotely within the authors. This study aimed to simulate the online study situation and the communication between the participant and the research moderator. Both authors of this paper had access to personal Oculus Go devices. Thus, the lead author who developed the system, tested the system, procedure, and other aspects of the study with the second author, who acted as a participant. Before this preliminary study, a factory-reset was performed on the second author's device. This action was so that we could identify each step that would be required to install the RotateEntry system and conduct the study. A few possible challenges of running a remote study on VR HMDs were identified during the preliminary research.

Firstly, the authors only had Oculus Go devices in hand, which made developing the system for different devices challenging. In addition, due to the COVID-19 pandemic, the campus laboratory was closed. We could not get access to more Oculus Go devices or other alternative HMDs from our campus laboratory. Hence, only those who had access to an Oculus Go HMD would be recruited for our experiment. However, on June 23, 2020, one month before our evaluation started, Oculus officially announced that they would discontinue Oculus Go devices[4]. This act might reduce the number of people purchasing it recently. Therefore, since the Oculus Go was our only in-hand VR HMD, we failed to test our demo application on other HMDs, such as Oculus Quest and VIVE Focus, which also limited the recruitment's conditions.

Secondly, we identified that collecting data in the remote study would be a problem. The subjects' involvement in data collection should be reduced to eliminate the issues caused by accidental misuse. Therefore, the testing data should be collected remotely instead of storing it on the subject's VR device. However, while sending the testing data from the VR HMD to an online server, the security and privacy of the communication ought to be ensured. Moreover, to facilitate the moderator to monitor the experiment process, the communication was best to be in real-time.

Thirdly, it would be difficult to instruct the subject on installing the demo application remotely. Since our evaluated application had not been released to the Oculus' official application store, the subject had to install it after activating the developer mode of the HMD. However, for those who were not Oculus developers, the application installation process could be complicated.

Finally, a solution for monitoring the participant's actions on the HMD side was needed. At the beginning of the experimental session, the participant might need help in using the evaluated application. It would also be intuitive to know whether the participant had set up the application correctly by getting access to the screen of the participant's VR HMD. However, there was no direct way to share the VR HMD's screen remotely during the study session.

4 STUDY DESIGN FOR THE ONLINE STUDY

4.1 Adaptation to An Online Format

Based on the challenges and requirements identified as described in the previous section, we modified our system and procedures to facilitate a remote online study. In addition, we estimated that the study would take approximately 90-120 minutes, and the participant would be allowed to complete the testing while seated.

For the remote testing control, input monitoring, and data recording, a password protected Node.js[5] server application was developed and deployed on a private HTTPS server. A MongoDB[6] database was implemented on the server-side to record the testing data. Websocket protocol implemented by Socket.io[7] was used on both the HMD side and the server-side to establish low latency network communication [13]. In this case, the text input and the calculated WPM

[4]https://www.oculus.com/blog/an-update-on-the-evolution-of-the-oculus-platform-/
[5]https://nodejs.org/
[6]https://www.mongodb.com
[7]https://socket.io

(Words Per Minute) and total ER (Error Rate) result emitted from the HMD side would be captured by the server near real-time. Besides, the WebSocket Secure (WSS) connection was used via the "wss" URI scheme to provide secure network communication [7]. The database used in the study would be isolated, and no personal identities of the participants would be recorded to protect the subjects' privacy.

To identify and isolate individual client-server communication, we assigned a unique four-digit identification code to each client. The client identification code was embedded in the demo application package when it was built; in other words, each subject would receive a different software package. There would be only one activated identification code on the server-side at a time, and only the corresponding client can connect to the server. In this way, we can ensure that only the specified device can communicate with the server.

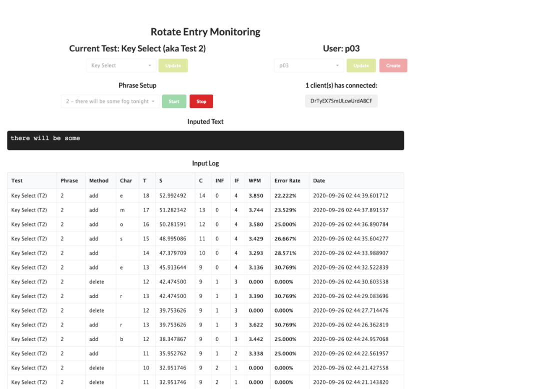

Fig. 3. Dashboard of the Web admin interface

Furthermore, a dashboard web admin interface was developed using the HTML5 technique (see Figure 3). Password was required to access the web page. The research moderator could set up the current tested text input method and the current phrase on the HMD side through the dashboard web page, modify the server-side's settings, see the HMD side input data visualization, and download copies of the testing results. These additional developments helped the experimenter monitor the study progress efficiently and intervene in case of potential issues.

4.2 Participant Recruitment

Since the evaluated demo application was developed for the Oculus Go HMD, only subjects who had access to an Oculus Go device would be recruited. Therefore for the scope of the case study, the participants were selected through word-of-mouth based on the institution's computer science department on their accessibility to an Oculus Go HMD. Due to the COVID-19 situation, we could not find enough participants with Oculus Go HMDs within a limited time, so the participant group size was limited. In the end, two main participants who owned Oculus Go HMDs were selected. These two main participants lived in two different three-room apartments. They were encouraged to invite their roommates to participate in our study and share their HMDs with their roommates. It was ensured that all roommates had been following the proper quarantine procedures since the issued guidelines. As a result, a group of 6 participants was recruited. One of them was invited to participate in our pilot study, while the rest were recruited as our formal test subjects.

4.3 Pilot Study Procedure

After the electronic informed consent form was signed, an application package of the evaluated demo was built and sent to the pilot study participant. The participant was asked to install the software on his/her Oculus Go device. Along with this, the participant was provided with detailed step-by-step instructions, including instructions that enable the developer mode and install the study software (the guidelines included as a Supplementary Material).

The study session consisted of the setup (if the participant hadn't pre-installed the study software) and conducting the formal user study that included testing the four text entry method conditions described in Section 2 in a randomized order. After each condition, the participant was provided with a link to the questionnaires to provide feedback and comments. Thus, our new procedure essentially remained similar to the previously planned face-to-face study procedure. Here, the main difference was the addition of a step to set up the software remotely, and the test being conducted entirely remotely online. This new online procedure was approved (as an amendment) by the Institutional Review Board.

4.4 Pilot Study Observations

After conducting a pilot test with one of the six recruited participants, some other issues were identified.

First, we found the VR HMD and the study encountered network latency problems, which caused the server to miss several test data sent from the HMD in the middle of the second task. For this, we identified that there were multiple devices (such as VR devices, mobile phones, and PCs) in the room connecting to the public (at home with multiple roommates) wireless network station at the same time, which might cause further network latency issues. In the meantime, the VR device's power drained before the pilot experiment ended, which required the participant to have the charger connected to the HMD to complete the study. As the main reason, we identified that it was the HMD screen sharing that caused the Oculus GO HMD battery to drain quickly.

Secondly, the participant unexpectedly switched the hand (controller using hand) in use while performing the task, but we did not know it until the study ended. Since our evaluation focused on text entry techniques, using different hands to complete a task might affect the evaluation result. However, due to the webcam's limited shooting angle, we failed to directly observe the participant's posture. Here, it must be noted that at the beginning of the study, the participant was instructed to keep using the same hand. However, upon later inquiry, we found that the participant had forgotten these instructions and had unknowingly switched the hands, possibly due to fatigue.

Thirdly, we occasionally found that other persons who were not involved in the experiment (roommates) walked in the background or talked with the participant (without knowing about the study being conducted), which might interfere with the subject while performing tasks. This situation was mainly due to the participant using the shared space, such as the house's living room, to participate in the study.

4.5 Refined Online Study Procedure

Based on the above issues revealed in the pilot study, we refined our study procedure design for the formal test.

After the participant was recruited in the research, he/she was provided with the Informed Consent form via email. Once a participant provided their consent, the study software was emailed to the participants with detailed installation instructions. The participant was free to install the software before or during the experimental session. Along with the email, there was also an invitation link to the study's video conferencing session.

Before the study session, the moderator would activate the subject's identification code on the server-side. The participant was informed that he/she needed to stay in a room alone and cannot talk to others, except the research moderator, during the entire session. After that, the participant was told to connect his/her Oculus Go HMD, mobile phone, and PC to the wireless network. The participant should then join the mobile phone and the PC to the Zoom.us[8] video conferencing session via the link attached in the invitation email. Next, the participant was told to open the evaluation software on the Oculus Go HMD, cast the HMD's screen to the phone via the Oculus app, and share the phone's screen in the video conferencing. The evaluation software would automatically connect to the server when it was opened. Finally, the participant was told to turn on his/her PC webcam and microphone.

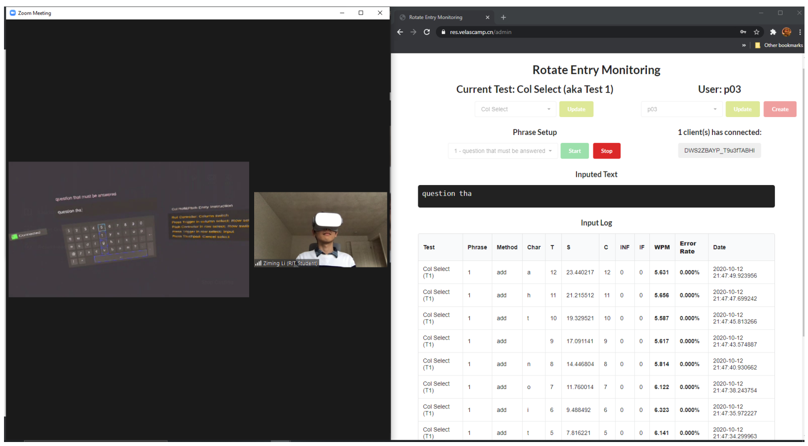

Fig. 4. A screenshot of the researcher's view of the online study

A screenshot of the researcher's view of the online study is shown in Figure 4. The webcam and the HMD screen sharing were only required at the beginning of the study session. When the initialization check was completed, the participant can feel free to turn off the webcam or stop HMD screen sharing to save the test device's power and the network bandwidth. However, the subject's Oculus Go device was required to stay connecting to the internet during the entire experiment. And the participant was required to keep his/her microphone on for the whole study session.

After filling out a demographic questionnaire, each participant was asked to complete four experimental conditions. In each condition, the participant was asked to enter the pre-selected phrases based on Boletsis and Kongsvik's work using one of the four VR text input methods mentioned above. At the beginning of each condition, the participant was also informed to keep using the dominant hand to hold the controller and complete the given tasks. The character input from the VR HMD, the WPM data, and the total ER data was monitored and recorded on the server-side during the test.

After completing each condition, the testing result was automatically recorded to the server-side database. In the meantime, a copy of the result was saved to the moderator's computer as a backup. Next, the participant was told to fill out three digital copies of the testing materials, including a SUS questionnaire [3, 4], a GEQ questionnaire [10], and a NASA TLX questionnaire [8, 9]. There was a 5-minute break after each condition. The same procedure was used within the remaining VR text input methods.

[8]https://zoom.us

After the participant completed all the tasks, a semi-structured interview was conducted via audio conferencing to collect the participant's comments for each of the four evaluated VR text input methods. The experiment took approximately 90 minutes per participant to complete.

After the study session, the participant was provided with a digital copy of the payment form. The participant was told to fill out the form with his/her preferred way of electric cash transfer and his/her payment account and send it to the researcher via email. Once the researcher received the email, a $15 compensation would be sent to the participant based on the payment information he/she provided.

5 OBSERVATIONS AND DISCUSSIONS

This section will focus only on the observation of conducting our VR study in a completely online format from a case study perspective. Here, we will not discuss the quantitative or qualitative results of our RotateEntry research as the scope of this work focuses on understanding the challenges and potential solutions of conducting a VR study online.

5.1 Test Environment Setup

The evaluated application package and a digital copy of installation instructions were emailed to the participants two days in advance to offer sufficient preparation time. The instructions were step-by-step and with marked screenshots on each step, aiming to provide a user-friendly app installation experience.

However, an Oculus developer account was required for the installation. Three participants who were non-Oculus developers faced issues with registering the account. In these circumstances, the participants were told to feel free to contact the research moderator for help via email. The moderator would provide timely feedback based on the situation. Besides, participants could also choose to install the app during the session. In this case, the participants were informed to schedule additional time with the moderator and enter the test session for at least half an hour in advance to ensure that the experiment could be completed on time.

5.2 Data Collection

With the Socket.io technique, a WebSocket connection was established between the HMD side and the server-side. It allowed the input data from the HMD side to be collected and monitored near-real-time [13]. Therefore, potential issues on the HMD side could be identified immediately. Also, since the WPM and total ER were calculated on the HMD side, it eliminated the impact of the potential network delay on the accuracy of the testing results.

However, the network latency and unstable network connection may cause the server-side to fail to receive some input data from the participants' VR devices during the experimental session. In our formal study, we informed every participant to stop performing the task as soon as the network connection indicator on the left side of the interface (see Fig. 1) displayed "disconnected", which might reduce the possible impact on the research, such as wasted time, and increased fatigue.

5.3 Observing the Participant

Observing the participant's actions was a significant problem for this remote study. In the experimental session, we used Zoom.us video conferencing platform to communicate with the participants. However, turning on the webcam for the entire study session was made optional, aiming to save network bandwidth. In this case, we could not observe the participant through the webcam if the participant chose to turn it off due to network issues. Additionally, it was difficult for the research moderator to observe the subjects' hand movements due to the webcam's limited shooting angle.

Therefore, it was almost impossible to learn directly whether the subject was using the controller in an anticipated way. Besides, such restrictions could affect the outcome of the study, depending on the research. For example, in RotateEntry, restricted movements may affect the participant using the system. Our instructions to restrict the postures were limited only to the beginning of each condition to address this aspect. In this way, the moderator could ensure that the participants would use proper postures.

We attempted to address this during the study session (to understand the user's postures and situations) in three ways: verbally inquiring the participant, monitoring the participant's input data, and watching the participant's HMD screen sharing. In our case, before the study session, the moderator would instruct the subject on using the controller through video conferencing. Every participant was informed to keep using his/her dominant hand for the entire test to eliminate possible bias. After the experiment session, the semi-structured interview also allowed us to learn more participants' posture details.

5.4 Experimental Environment Control

Typically, similar user studies are done in controlled lab environments to limit external disturbances. Therefore, during this remote study, the participant's test environment setting could also be an issue. Before the study session, we informed the subjects that they needed to perform the tasks alone in a quiet environment and keep their phones muted. However, due to various circumstances, we found that three of the six participants conducted the study in the common spaces of their homes where other roommates lived. Therefore, it was inevitable that the participant might get disturbed by the surroundings.

Additionally, since mobile phones and computers were used during the study session, subjects would sometimes be disturbed by notifications that appeared on the screen. Besides, while the HMD screen sharing was conducted through the phone, the screen sharing would often be blocked by calls or phone display turning off. And the screen sharing would be interrupted because of the unstable connection between the VR device and the mobile phone. When screen sharing interruption occurred, the moderator would instruct the participants to continue with the task. Since we collected the user input data via the communication between the HMD side and the server-side, the screen sharing interruption only minimally affected the testing process.

5.5 Participant Recruitment

Before the COVID-19 pandemic, we ignored developing the RotateEntry system compatible with different HMDs. This situation was mainly because we could use the same device for various participants in the planned face-to-face studies. Therefore, our research was focused only on the Oculus Go HMD. However, this limited the potential participant pool that we could recruit.

5.6 Recommendations for Remote VR Studies

Through these experiences and observations, we present a few recommendations to consider when designing remote VR studies.

(1) As for a privacy-protected, secure, and real-time data collection process, a password-protected admin web page and server, and a WebSocket communication with Transport Layer Security (TLS) enabled are recommended. A unique identification code could be assigned for each client to isolate the communication. Also, minimal personal identities of participants should be recorded on the server-side.

(2) Depending on the research and where possible, aim to develop the VR software for different models of VR HMDs. This act could increase potential participation.
(3) Clear instructions on the study application installing and uninstalling are essential. The instructions may also contain what the subjects should and should not do during the test. When preparing the instructions, it is best to assume that the participant does not know about setting up a VR device. It is also recommended to test the instructions on a factory-reset device to ensure all steps are covered.
(4) It could be common to encounter unexpected issues in a remote study. Therefore researchers should reserve as much time as possible for each study session.
(5) Remind the participants to have the HMD's battery and controller's battery recharged before commencing the study (preferably charge levels should be above 90%). If there were alternative ways to monitor data from HMD, the HMD screen sharing could be turned off in later tasks to save power.
(6) The number of devices connected to the network could be reduced to save network bandwidths. For example, the webcam and the HMD screen sharing could be turned off in later tasks.
(7) In a remote study, the conditions currently being tested should be set by the moderator.
(8) At the beginning of the session, inform the participant to turn on the webcam to check the testing environment setup.
(9) When the study session starts, there should not be any people unrelated to the experiment together with the participant in the room. The participant should mute his/her mobile phone and turn off all the notifications.

5.7 Advantages of Remote VR Studies

During the study session, we found there were a few advantages of conducting a remote study with VR HMDs.

Firstly, we found that these circumstances allowed us to conduct the studies in a more real-world (or in-the-wild) setting. For example, we observed that three participants performed the testing in their rooms with no interference from other people around them. In such situations, their performance in the experiment might lead to more relaxed and natural usage than the experiments conducted in the laboratory. Therefore, the data collected in remote studies might be closer to the one measured in the real environment.

Secondly, when recruiting subjects, the researcher did not have to consider whether the subjects were local, which was unlike the participant recruitment for a face-to-face study. In this case, the composition of the subjects could be enriched. In addition, although we only approached the computer science students from the authors' institution, this would allow the research to reach a much wider audience across the globe if advertised on forums and social networks. Such has been recommended in Steed et al.'s work.

6 CONCLUSION

This paper discusses using a privacy-protected, secure, and real-time data collection system to adapt our evaluation on VR text input methods with HMDs from a face-to-face format to an online one. Additionally, we share our experience in designing study procedures for conducting a VR remote study. Also, we describe the challenges we encountered during the remote research, our proposed solutions, and the advantages of using an online format in a VR study.

REFERENCES

[1] Nic Beech and Frederik Anseel. 2020. COVID-19 and Its Impact on Management Research and Education: Threats, Opportunities and a Manifesto. *British Journal of Management* 31, 3 (July 2020), 447—449. https://europepmc.org/articles/PMC7361241

[2] Costas Boletsis and Stian Kongsvik. 2019. Controller-based Text-input Techniques for Virtual Reality: An Empirical Comparison. *International Journal of Virtual Reality* 19, 3 (Oct. 2019), 2–15. https://doi.org/10.20870/IJVR.2019.19.3.2917

[3] J. Brooke. 1996. SUS: A 'Quick and Dirty' Usability Scale.

[4] John Brooke. 2013. SUS: A Retrospective. *J. Usability Studies* 8, 2 (Feb. 2013), 29–40.

[5] David DeMatthews, David Knight, Pedro Reyes, Amber Benedict, and Rebecca Callahan. 2020. From the Field: Education Research During a Pandemic. *Educational Researcher* 49, 6 (2020), 398–402. https://doi.org/10.3102/0013189X20938761 arXiv:https://doi.org/10.3102/0013189X20938761

[6] T. Dube and A. Arif. 2019. Text Entry in Virtual Reality: A Comprehensive Review of the Literature. In *HCI*.

[7] Ian Fette and Alexey Melnikov. 2011. The websocket protocol.

[8] S. Hart and L. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in psychology* 52 (1988), 139–183.

[9] Sandra G. Hart. 2006. Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50, 9 (2006), 904–908. https://doi.org/10.1177/154193120605000909 arXiv:https://doi.org/10.1177/154193120605000909

[10] W.A. IJsselsteijn, Y.A.W. de Kort, and K. Poels. 2013. *The Game Experience Questionnaire*. Technische Universiteit Eindhoven.

[11] Natalia Kucirkova, Cecilie Evertsen-Stanghelle, Ingunn Studsrød, Ida Bruheim Jensen, and Ingunn Størksen. 2020. Lessons for child-computer interaction studies following the research challenges during the Covid-19 pandemic. *International Journal of Child-Computer Interaction* (2020), 100203. https://doi.org/10.1016/j.ijcci.2020.100203

[12] M. Bishr Omary, Jeetendra Eswaraka, S. David Kimball, Prabhas V. Moghe, Reynold A. Panettieri, and Kathleen W. Scotto. 2020. The COVID-19 pandemic and research shutdown: staying safe and productive. https://www.jci.org/articles/view/138646

[13] V. Pimentel and B. G. Nickerson. 2012. Communicating and Displaying Real-Time Data with WebSocket. *IEEE Internet Computing* 16, 4 (2012), 45–53.

[14] A. Steed, S. Frlston, M. M. Lopez, J. Drummond, Y. Pan, and D. Swapp. 2016. An 'In the Wild' Experiment on Presence and Embodiment using Consumer Virtual Reality Equipment. *IEEE Transactions on Visualization and Computer Graphics* 22, 4 (2016), 1406–1414.

[15] Anthony Steed, Francisco R Ortega, Adam S Williams, Ernst Kruijff, Wolfgang Stuerzlinger, Anil Ufuk Batmaz, Andrea Stevenson Won, Evan Suma Rosenberg, Adalberto L Simeone, and Aleshia Hayes. 2020. Evaluating immersive experiences during Covid-19 and beyond. *Interactions* 27, 4 (2020), 62–67.