Multimodal Interface Study of Voice and Modified Gestures on Immersion and Effectiveness

ISABELLA BARNARD*, JOHN MAKSUTA*, and JADEN MASCARENHAS*, Colorado State University, USA

An introduction to Human Computer Interfaces and Virtual Reality. Related works are discussed and provide some context for the topic. Our methodology is then discussed and provided a detail the study we are conducting.

CCS Concepts: • **Human-centered computing → Gestural input**; **Pointing devices**; **Sound-based input / output**; **Empirical studies in HCI**; **Virtual reality**.

Additional Key Words and Phrases: Multi-modal

ACM Reference Format:
Isabella Barnard, John Maksuta, and Jaden Mascarenhas. 2018. Multimodal Interface Study of Voice and Modified Gestures on Immersion and Effectiveness. 1, 1 (May 2018), 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Human-Computer Interfaces and Virtual Reality are great tools for bringing users into environments that are often easier to produce than building the real thing. In VR we can create simulated environments that feel very immersive for users. In order to create an immersive environment, users must be able to interact with the device and the virtual world. The way a user interacts with a device is called a modality, and we will be evaluating a multi-modal interface where the user will be using two different interfaces to perform certain tasks.

For our project, we are going to create a 'spell casting' game. The focal idea for our project is to use speech as our main form of modality and to test various modalities against this to determine which is the most comfortable, easily utilized, and immersive for the user. We will be testing whether users prefer hand gestures, controllers, or the Logtitech MX Ink Combo pen, in addition to speech, to cast spells. We will be analyzing the user's ease-of-use, and we will gather information regarding the response time and other quantitative data from our test group. Our goal is to be able to determine which modality should be used in conjunction with speech to create the optimal experience of the game and fine-motor VR experiences in general.

2 Related Work

In relation to our project, we have reviewed many related research studies. Some of these works are in relation to drawing in the virtual space. A benefit of using VR for drawing and creative expression is that it is not as restrictive as real-life drawing. The canvases are infinite, and the user can create almost anything they want. In the study, it is discussed that Vr creates a space in which natural laws do not need to be followed. [6]. This can be

*all authors contributed equally to this research.

Authors' Contact Information: Isabella Barnard, isabelba@colostate.edu; John Maksuta, jmaksuta@colostate.edu; Jaden Mascarenhas, c835223467@colostate.edu, Colorado State University, Fort Collins, Colorado, USA.

ACM XXXX-XXXX/2018/5-ART
https://doi.org/XXXXXXX.XXXXXXX

extremely appealing for those who want to create an environment in VR that is not practical in real life. Our work extends this idea because we aim to create an experience that defies natural laws, such as casting spells.

Since we utilize drawing in virtual space, some of our research focuses on the issues that may arise while drawing in virtual space. Many studies have been conducted related to some of the issues users may have while drawing in a VR space. In this study, the researchers addressed the fact that 2D projections often allow the user to communicate easily, while 3D space is more difficult to utilize. The issue arises in response to a lack of understanding related to depth perception in a 3D space. It is discussed that accuracy is difficult to maintain while drawing in the 3D space, since 2D drawings are typically less complex. [19]. Building on this study, we have tried to reduce these issues by focusing on accurate strokes. In our gesture recognition, we attempt to remedy any issues the user may have while drawing in the virtual space.

In addition to depth perception, another issue that may arise is that structured sketches are also difficult to complete in VR because of low precision. Due to a lack of accurate expression, many users may opt out of using VR sketching tools. A visual guidance tool may improve precision and depth perception. A guidance tool may use continuous stroke recognition and guide lines to improve the user experience. The study concluded that the use of guide tools helped improve structural accuracy[9]. Unlike prior work, our goal was to use multi-modal interaction in order to improve the user's experience by improving accuracy through multi-modal input, in conjunction with speech recognition.

Despite these issues, many VR applications have been created for artistic expression. These applications include drawing, modeling, and animation applications. By using auto-correction, many applications are able to tidy up lines, allowing the user to have a more controlled experience. In order to do this, the user needed access to canvas manipulation. The goal of the study was to allow for users to freehand their sketches while maintaining precision and control. In the study, the completion time and the precision of the drawing were the main aspects tested. Participants were asked to draw circles, straight lines, bows, tildes, and eclipses. In the study, it was concluded that moving the canvas contributed to improve both smoothness and precision, which had the greatest impact on lowering completion time[18]. However, this study did not use the drawing aspects in conjunction with any other form of input. Our work extends these findings by utilizing canvas mechanics to improve the user experience by adding more precision and control in the virtual environment.

In terms of multi-modal interaction, a study was done regarding how combining hand gestures and drawing may lead to more accurate sketches. The researchers conducted a study comparing the use of two hands to tablet-pen interactions while sketching in VR. They used gesture capture to use one hand for a canvas while the other created the drawings with one finger. The researchers faced challenges related to the limited region, fingertip sensitivity, uneven canvases, and instability. Using a canvas for a hand caused the limited region. It was difficult to track the fingertips because the gloves relied on joints bending. The uneven canvas was caused by the hand's uneven surface. Instability was caused because our hands are not perfectly stationary. Despite these setbacks, the researchers allowed the canvas to move using a pinching motion. They added a step for calibration to calculate the fingertip distance. To fix the uneven surface, they created sub-regions on the hand. Lastly, they used algorithms to smooth the strokes[8]. Our work extends this idea by utilizing multi-modal interaction as well. However, unlike prior work, we are using speech in combination with our drawing inputs.

In addition to drawing in the VR space, another aspect of our project is the recognition of user input. For symbol recognition in VR spell-casting games, a study was done regarding line-based gesture recognition. In this study the researchers used a recognition program to convert the inputs into a list. This process was reliant on a database, which helped recognize the shapes. They tested this game mechanic with their research group, and found that mixed results arose from the spell-casting. The process of learning to cast the spells was easy, while the complexity of the symbols seemed to create issues. This difficulty that their users encountered was likely due to a lack of demonstration, which they stated in their findings. A proposed solution for this problem in their study was to increase the number of recognized symbols and methods for symbol recognition[10]. Our work builds on these findings through the use of an in-game spell book that we created. We aimed to reduce any complexity issues by providing the users a set of spells they can utilize to improve user confidence.

In our project, we will use gestures as a form of input for the spells. This is common in human-computer interaction. In a study done by Lin Jiang, Xiaoyang, and Lijun Wang, it is discussed that information is conveyed through gestures and that to recognize these gestures, you must utilize pattern recognition [7]. In this study, it is discussed that adding gesture recognition to VR has greatly improved interaction between consumers and the environments they are interacting with. In our project, we will utilize this idea to improve the ability to convey the user's intentions with their gestures, in order to cast the spells.

Gestures may also be used to enhance user experience with the game. Studies have been performed that examined the use of gestures in a One- or Two-Handed modality, using task based comparisons of the user interfaces[16]. In their study, they wanted to find out more about using VR as a tool and were analyzing two hand-tracking modalities, these being One- and Two-Handed[16].

In their analysis they evaluated the how well suited the gestures were to each interface [16]. This is similar to the approach that we are taking in evaluation of multi-modal interactions, although our scope is more limited. This study concluded that the One-Handed user interface was preferred over the Two-Handed interface except for the group who tested the One-Handed interface first. Their study concluded also that One-Handed interfaces were faster to complete tasks on average, and that right handed gestures were preferred, although the vast majority of participants were right-handed [16]. Our work extends these findings by focusing on certain quantitative metrics and task completion time. Unlike prior work, we are not testing whether one handed interactions are preferred over two handed interactions.

Input methods are often evaluated with many different human computer interactive technologies. These inputs are changing over time, and that VR may be used when it comes to challenging the traditional WIMP (Window, icon, menu, point-and-click devices) inputs [3]. For the head-worn displays, such as the Oculus Quest, the input methods most explored were freehand interaction, followed by head-based input, while speech interaction that was hardware based occurred frequently, but was considered less [14]. Building on previous studies, our study will evaluate a variety of inputs, each used as multi-modal input with a speech interaction. The multi-modal input combinations most frequently evaluated, in a 2023 review of interaction techniques, were head and hardware controller, followed by hand and speech, and hand with head[14]. Our project will incorporate hand and speech multi-modal interaction techniques.

The number of tasks performed for evaluation for the head-worn displays varied from rarely one or four, and the most two or three tasks. Only one study in the review evaluated voice and hand gestures for a task that interacted with a virtual character[14]. Unlike prior work, we will not be interacting with a virtual character but rather a target in which we can cast our spells on. Besides hand gestures, we will also be using speech input for our game.

Speech input methods had many interesting findings in the 2023 review. Although, it was found to be less natural for single user applications and more appropriate for collaborative environments with multiple users. In educational applications, voice input was found to aided in learning and memory retention[14]. Adding speech inputs has also shown to be advantageous in terms of task completions. When compared against the traditional mouse WIMP pointing device, speech allowed users to complete tasks far more easily [13].

AI may be used in conjunction with speech input for better annotation results. In a study conducted by Shanshan Yang and Ding Liu, it is stated that artificial intelligence based annotation had many advantages[20]. By using AI based annotation for VR speech corpus, the researchers determined that the annotation method they tested had higher efficiency as well as accuracy in comparison to traditional annotation[20]. Our speech recognition also uses a highly accurate transcription of what the user is saying in order to cast spells, so we can accurately decipher which spell the user is trying to cast.

Another study by Barry A. Po, et al. studied a hypothesis regarding two visual systems, which provides information about the interactions between perception and movement. This study similar to ours explored the multi-modal interfaces of speech and gesture, however, in their study, they were concerned with target selection and made several predictions which their study experiments were conducted to test. The results of their study demonstrated how things such as visual cues can impact speech and gesture interaction in regards to target selection[2]. Our work builds on this study by using the same multi-modal interfaces for interaction in the virtual environment. However, their use was for target selection, whereas ours is utilized for task completion.

Voice and gesture interaction has also been used in the teaching scenarios, and this was explored in a study by Ke Fang and Jing Wang[4]. They used voice recognition and gestures to complete a series of tasks including teleportation, adjusting lighting levels, and entering numbers in a simulated keypad. This was compared against completing the tasks with controllers only. In the experiment, tasks 1 and 2 were longer and task 3 was shorter using voice and gesture versus controller. Although it was not entirely conclusive due to the scope whether voice and gesture are better inputs than controller, it demonstrated an enhanced interactive experience in regards to virtual teaching [4]. We build off this study by testing the controller and the hand gestures in terms of which form of input is preferred by users. [17] [15] [1] [11]

3 Methodology

To evaluate our different modalities, we developed a Unity-written virtual reality program to be played on the Oculus Quest 3. Our study evaluates three different modalities: controllers, hands, and the pen device. The Control variable in this study is the speech input which will remain constant and be present in each test. The independent variables are the second modalities.

Each task will require the user to "cast a spell" of some type and either target an object or have it automatically target the object in the game. The spell will be considered successfully cast when the user successfully completes each modality task within a certain time frame, such as 5 seconds. The speech input is considered successful when the user says an acceptable phrase, such as "cast fireball" or "cast charm", etc. The independent variable, input modalities, will be considered successful when the user draws a certain symbol path using their gesture input. Rather than relying on static hand gestures, the user will be performing semaphoric gestures by making a movement with their hand in the 3D space. The hand's path will be recorded and manipulated to derive a 2D path of coordinates, which forms a unique symbol. These unique symbols drawn by the user will be compared against a 'spell book' set of symbols that will identify a particular spell. If the user completes these two tasks nearly simultaneously, the task is considered successful. Successful tasks will then display a notification to the user, in the form of a 3D object spawning, along with a sound. The failed tasks will result in a different notification to indicate the failure. The number of successes and failures will be recorded, as well as the time it takes to complete each task. After completion of each independent variable modality, the user will be presented with the end of the game and complete a survey that is distributed to them by the researcher.

To successfully complete the task, we use a Task Recognizer that coordinates the events of two sub-components: the Voice Recognizer and Gesture Recognizer. When both recognizers detect a successful cast, the task recognizer calls the OnSuccess event. There is a time recorded when a single success is logged, and if the timeout expires, the task recognizer calls the OnFailure event, and the values of the task recognizer and each sub-component are all reset for the next attempt.

The voice input is utilizing the Meta Voice SDK, which uses the "Wit.ai Natural Language Understanding (NLU) service" [5]. We have trained the service with three vocal components or phrases for spells, "fireball", "smoke", and "charm". Each spell can be cast by saying the name of the spell or a phrase such as "cast fireball". The SDK in unity then receives data objects from the service and logs when a spell phrase has been successfully recognized and which spell it is.

The gesture input is managed by collecting a Vector3 coordinate from an input interface, which is implemented by each input method separately. There is an implementation of the interface for each input type: controller, hands, and pen. The raw Vector3 coordinates are stored in a list and then projected on the screen as Vector2 screen coordinates in the domain and range [-1.0, 1.0]. The Vector2 screen coordinates use the same domain and range as previously defined spell library gestures. These library entries are the gestures used to successfully cast a certain spell. These library entries are also displayed to the user in a spell book that is displayed when they pinch their left thumb and index fingers. When coordinates are captured, they are converted and searched in the spell library for a match. We then utilize the Directed Hausdorff distance comparison method to recognize the shape during the search. "The directed Hausdorff distance function:

$$h(A, B) = max_{a \in A} min_{b \in B}(d(a, b))$$

Where a and b are points of sets A and B respectively, and d(a, b) is any metric between these points," and this

returns a max distance per shape in the library [12] [21]. The smallest maximum distance score will be considered the best score, which will be returned. There is a NULL shape in the library which is a Vector2 list with one point at coordinate (0, 0).

The results of the qualitative and quantitative data stored will be analyzed and compared to answer the following questions, and each modality will be ranked for each question.

- `Ease of use`: Which modality in conjunction with speech was the easiest (which took the least amount of time)?
- `Preference`: Which modality did the participant prefer?
- `Effectiveness`: Which modality was the most effective?

3.1 Participants

Voluntary subjects will be recruited and information will be provided to obtain informed consent from the participants in the study. Subjects will be given a randomized order of the three variations of modalities, Speech and Controllers, Speech and Hands, and Speech and Pen device. We will utilize latin squares to randomly assign the order of the tests for each participant.

3.2 Hardware

The hardware devices are the Quest 3 headset, the included pair of controllers, and the Logitech MX Ink Combo Pen. The Quest 3 headset uses a series of cameras to determine the position of hands when using the hand inputs, rather than the controller. It uses the head mounted display to display the virtual environment to the user. The display is capable of using its cameras to display an Augmented Reality mode called pass-through, where the cameras display their output in real-time to the head-mounted display through which the user is viewing. This creates the perception that one is seeing their real-world surroundings while also allowing the placement of virtual objects in the environment. For our purposes, we will be using the full virtual reality display without the pass-through feature, but it is worth noting that the multi-modal inputs of voice and gesture could have many use cases in Augmented Reality as well.

3.3 Prototype Description

Our prototype is written in Unity version 6000.0.38f1. We are using OpenXR and XR Hands, with the Meta Voice SDK, and the packages for the Logitech pen support. Our application loads a model for the environment that appears like a rural medieval landscape. At the beginning of the application, the user is prompted to enter a two-

or three-digit Latin square code, which is a numeric code in our application, where 1 is the controllers input, 2 is the hands input, and 3 is the pen input. After entering the latin square code, the game makes a list of levels and their input, then starts the game. The user is presented in front of a training dummy and given 10 minutes per level to complete 5 spell castings. They may cast one of 3 spells: fireball, smoke, and charm. The user then says a phrase or word and makes the corresponding gesture, for instance "cast fireball" or just "fireball" and makes a forward slash gesture shape. To activate voice, the user uses the primary button on the left controller or pen, or makes the "shaka" hand shape in hand inputs. The "shaka" shape is where the pinky and thumb are extended and the other fingers are folded over, is often seen in surfing communities for "hang loose". Successful attempts are marked on the screen with a green check mark, and failures are marked with a red x. Once five successes are logged or ten minutes elapses, the level ends and the next level begins. Each level displays the latin square code, the current Latin square value, and the input method, as well as the successes and failures. When every level is completed, a game-over screen is displayed, and the trial ends. At the end of each level, a report file is generated, which logs the start and end times of the level and the statistics for each success and failure. The files are then used to collect the data and evaluate them.

3.4 Procedure

Subjects will be assigned to complete tasks in an order determined by latin squares to randomize the order of tests. There are three tasks the subject will complete: voice and controllers, voice and hands, and voice and pen. Each multi-modal pair will involve the user saying a command phrase and performing a specific gesture within a given time frame. When both tasks are successful and the time is within a certain limit, the task will be completed successfully. The number of attempts, and completion time will be recorded for analysis.

Each level has its own report and data points recorded. The Latin Square Code and current input method is recorded for the level. When a level starts, the start time is recorded, and when a level ends, the end time is recorded.

Tasks are recorded individually, with each having a recognized voice task and time stamp, recognized gesture task and time stamp, recognized task and time stamp, and whether the task was a success or failure. When the recognized voice and gesture tasks match, the recognized task is a match and the task is successful. When there is no match, or there is only one input (i.e. time expires) the task is a failure. Successes and failures are both logged and reported when the level is ended.

In the effort to collect qualitative data we will also have a survey that will either be distributed after the subject completes the tasks, or it will be displayed as a software user interface for the user to complete at the end of the tasks. The data will elicit the opinions of users on a scale of various data points such as, ease of use, immersion, realism, and preferences. We will also collect some demographics information such as age and prior experience with XR technology.

3.5 Design

The experiment is a one-way within-subjects design. Where each participant performs under each of the 3 input variables: controllers, hands, and pen.

Some issues were encountered which prevented more participants from performing the trials, so only a limited set was produced.

The total number of trials was 6 (=2 participants x 3 input methods x 1 session x 1 trial/session).

3.6 Reproducibility

To allow for reproducibility of our study, all code was written in Unity version 6000.0.38f1. All of our scripts are written in C#. We are using the XR Interaction Toolkit for VR compatibility. We are using the Logitech Assets to utilize the Logitech MX Ink pen, and we used the Meta Quest headset for testing.

In order to replicate the study please see our GitHub page here:

$https://github.com/csu-hci-projects/CS465_{M}ascarenhasJ_{B}arnardI_{M}aksutaJ$

You can clone the repository, open the project in Unity (version 6000.0.38f1), and play on the Meta Quest headset.

4 Results

4.1 Evaluation

Task completion time is calculated and evaluated for each task. Each individual task attempt is recorded with a start time as the earliest recognized time between the voice or gesture, and the latest recognized time between the voice or gesture. This also includes null results for failures. Average completion time for the Controller was 12 seconds and the Hands average was 24 seconds. Hands completion time was twice as long as the controller. The standard deviation of the controller is 7 seconds while the hands is 16 seconds.

Standard Devation is calculated as: s = $\sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}}$ where x_i is the individual data point, $\bar{x}$ is the mean, and n is the total number of data points.

In the task completion, only one error was recorded, with a voice phrase fireball and a null gesture. This failure was recorded using the hands input source.

The level duration time is calculated and evaluated for each task. It is calculated by using the difference between the end time and start time of the level. The hands was found to take longer time for overall completion. Corresponding to the task completion time, the hands takes much longer compared to the controllers.

Qualitative data evaluated the ease of use of the Speech, Controllers, Hand, and Pen inputs, as well as the level of Immersion and the level of Enjoyment. Each point was in a scale from 1 to 10. We found that the ease of use was easy with hands only being slightly less easy. The immersion was highly immersive, however the enjoyment was also slightly less enjoyable.

4.2 Quantitative Data Summary

	Start Time:	End Time:	Completed Tasks Count:	Duration (s):
Subject 1				
Controller	5/3/25 19:38	5/3/25 19:39	5	57.072704
Hands	5/3/25 19:39	5/3/25 19:42	5	158.88037
Subject 2				
Controller	5/3/25 19:52	5/3/25 19:53	5	57.292635
Hands	5/3/25 19:54	5/3/25 19:55	5	71.894384

4.3 Quantitative Data

	Time:	Phrase:	Phrase Time:	Gesture:	Gesture Time:	Success:	Failure:
Subject 1							
Input Method: Controller							
	7:38:30 PM	fireball	7:38:44 PM	Fireball	7:38:44 PM	True	False
	7:38:44 PM	fireball	7:38:48 PM	Fireball	7:38:48 PM	True	False
	7:38:48 PM	charm	7:38:55 PM	Fireball	7:38:55 PM	False	True
	7:38:55 PM	fireball	7:39:00 PM	Fireball	7:39:00 PM	True	False
	7:39:00 PM	fireball	7:39:19 PM	Fireball	7:39:19 PM	True	False
	7:39:19 PM	fireball	7:39:26 PM	Fireball	7:39:26 PM	True	False
Input Method: Hands							
	7:39:27 PM	fireball	7:40:22 PM	Fireball	7:40:22 PM	True	False
	7:40:22 PM	fireball	7:40:40 PM	Fireball	7:40:40 PM	True	False
	7:40:40 PM	fireball	7:41:28 PM	Fireball	7:41:28 PM	True	False
	7:41:28 PM	fireball	7:41:47 PM	Fireball	7:41:47 PM	True	False
	7:41:47 PM	fireball	7:41:55 PM	null	7:41:55 PM	False	True
	7:41:55 PM	fireball	7:42:05 PM	Fireball	7:42:05 PM	True	False
Subject 2							
Input Method: Controller							
	7:52:17 PM	fireball	7:52:41 PM	Fireball	7:52:41 PM	True	False
	7:52:41 PM	fireball	7:52:46 PM	Fireball	7:52:46 PM	True	False
	7:52:46 PM	fireball	7:52:52 PM	Fireball	7:52:52 PM	True	False
	7:52:52 PM	fireball	7:53:04 PM	Fireball	7:53:04 PM	True	False
	7:53:04 PM	fireball	7:53:14 PM	Fireball	7:53:14 PM	True	False
Input Method: Hands							
	7:54:07 PM	fireball	7:54:27 PM	Fireball	7:54:27 PM	True	False
	7:54:27 PM	fireball	7:54:43 PM	Fireball	7:54:43 PM	True	False
	7:54:43 PM	fireball	7:55:01 PM	Fireball	7:55:01 PM	True	False
	7:55:01 PM	fireball	7:55:10 PM	Fireball	7:55:10 PM	True	False
	7:55:10 PM	fireball	7:55:18 PM	Fireball	7:55:18 PM	True	False

4.4 Qualitative Data Summary

	Ease of Speech	Ease of Controllers	Ease of Hand	Ease of Pen	Immersion	Enjoyable
	10	10	9	N/A	10	10
	10	10	10	N/A	10	7
AVG:	10	10	9.5	N/A	10	8.5

5 Conclusion

In the experiment, these data suggest the average completion time for the controller is less than hands, and it has a smaller deviation. All levels were cleared successfully, and all five tasks were completed in less than 3 minutes, given the limit of 10 minutes. There was only one attempted task failure.

Some observations were collected by observation of the subjects and through statements made by participants during and after the trial. A participant stated that the user has to clear the hand gesture after each voice command. They also stated that they had to do the voice and then the gesture, although the code does not require this at all.

For a successful task to be completed a matching voice and gesture has to be completed within a certain time frame. The "shaka" hand gesture was also difficult for some subjects than others. Prior experience may play a factor in completion time, and the controller's voice activation of a button press may be easier to use than the "shaka" gesture. The "shaka" hand shape may need to be substituted for a less difficult hand shape.

In the trials our participant pool size was not large enough to perform a statistical analysis with any utility for science. So the trials were performed as a small test as a pilot for a larger study that can be performed outside of a course setting. Because it is a course, and not a real study that we are intending to publish, the sample size is very small. For a trial to have any real meaning a sample size of greater than 20 participants must be utilized, because otherwise the statistical data is flawed.

Unfortunately, due to last minute changes in the equipment distribution, we were not able to include the Logitech MX pen in the trials. Voice activation was not working due to an unknown bug that has to be fixed in the code. It is unknown why the button press is not being registered in the XR Input Actions. The positional data was being captured, but the button press was not.

Based on all the data and the experiment, it is inconclusive from a statistical point of view, but initial findings suggest that controller input is easier to use than hands inputs when performing a simple gesture, in conjunction with speech.

5.1 Limitations

Some limitations in our project are due to the fact that we built our project in Unity ourselves, and so issues and bugs may arise during testing if the player deviates from the instructions. We also used the Meta Quest headsets, so compatibility with other headsets cannot be guaranteed. Our gesture and hand recognition relies on hand shape and can cause issues to arise if the person testing our game has a hand shape that is not compatible with our tracking system. A limitation with our speech recognition is that it was trained by us as the creators of the game, and it only recognizes the English language. This voice recognition may also not account for those with accents, and can be quite fickle if the words are not pronounced clearly.

5.2 Future Work

In the future, to combat these limitations, researchers may be able to increase the number of devices our game is compatible with, which will make our study more replicable. Development to fix the code to support the Logitech MX pen must be completed. Researchers may also be able to discover and patch any possible bugs that may arise during testing, as a wider variety of players is more likely to discover potential issues. In addition to this, gesture and hand recognition may also be able to be adjusted to fit a larger community of test subjects, ensuring more reliability for all players. The hand shape used for voice activation should be changed to an easier but distinct shape to allow easier activation. A voice activation that does not require a button press or hand shape would be best. Supporting other languages and accents to be acknowledged and used in our recognition system, which is the Meta Voice SDK.

Initially, the project was envisioned as a game, however as time progressed, development tasks required to make a functional game were not able to be completed. This made the project a task-oriented application within a virtual environment. We still would like it to be a game, but for that to happen more code is required were needed. We also wanted an in-game instructor, but that was not completed or added to the final scene. We have a world level large enough to accommodate movement, but no movement code was developed. Our visual elements were not completed with high-quality assets for the spell effects either. Currently, the application is not fun, so it is not really a game. When participants are finished, it feels like they are doing work, not having fun.

To run an actual scientific trial, the participant pool must be large enough to have statistical significance. Future work would include a larger trial selection.

References

[1] Rahul Arora, Rubaiat Habib Kazi, Danny M. Kaufman, Wilmot Li, and Karan Singh. 2019. MagicalHands: Mid-Air Hand Gestures for Animating in VR. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. 463–477. doi:10.1145/3332165.3347942

[2] Brian D Fisher Barry A. PO and Kellogg S Booth. 2005. A Two Visual Systems Approach to Understanding Voice and Gestural Interaction: Lanquage, Speech and Gesture for VR. *Virtual reality: the journal of the Virtual Reality Society* 8, 4 (2005), 231–241. doi:10.1007/s10055-005-0156-2

[3] A. Van Dam. 2000. Beyond WIMP. *IEEE computer graphics and applications* 20, 1 (2000), 50–51. doi:10.1109/38.814559

[4] Ke Fang and Jing Wang. 2024. Interactive Design with Gesture and Voice Recognition in Virtual Teaching Environments. *IEEE access* 12 (2024). doi:10.1109/ACCESS.2023.3348846

[5] Meta Horizon. 2025. Voice SDK Overview. https://developers.meta.com/horizon/documentation/unity/voice-sdk-overview/. Accessed: 2025-04-20.

[6] Christian Jones Janice Tan, Lee Kannis-Dymand. 2023. Examining the potential of VR program Tilt Brush in reducing anxiety. *Virtual Reality* 27, 4 (Dec 2023), 3379–3391. doi:10.1007/s10055-022-00711-w

[7] Lin Jiang, Xiaoyang Yu, and Lijun Wang. 2020. A brief analysis of gesture recognition in VR. *SID Symposium Digest of Technical Papers* 51, S1 (2020), 190–195. doi:10.1002/sdtp.13787

[8] Ying Jiang, Congyi Zhang, Hongbo Fu, Alberto Cannavò, Fabrizio Lamberti, Henry Y. K. Lau, and Wenping Wang. 2021. HandPainter - 3D Sketching in VR with Hand-based Physical Proxy. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, 1–13. doi:10.1145/3411764.3445302

[9] Shuxia Wang Weiping He Jingjing Kang, Shouxia Wang. 2021. Fluid3DGuides: A Technique for Structured 3D Drawing in VR. *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology* (Dec 2021), 1–2. doi:10.1145/3489849.3489955

[10] M. Katzourin, D. Ignatoff, L. Quirk, J. J. LaViola, and O. C. Jenkins. 2006. Swordplay: Innovating Game Development through VR. *IEEE Computer Graphics and Applications* 26, 6 (2006), 15–19. doi:10.1109/mcg.2006.137

[11] Daniel Keefe, Robert Zeleznik, and David Laidlaw. 2007. Drawing on Air: Input Techniques for Controlled 3D Line Illustration. *IEEE Transactions on Visualization and Computer Graphics* 13, 5 (2007), 1067–1081. doi:10.1109/TVCG.2007.1060

[12] Manigandan T. Chitra D. Murali L. Kumar, K. S. 2020. Object recognition using Hausdorff distance for multimedia applications. *Multimedia Tools and Applications* 79(5–6) (2020), 4099–4114. doi:s11042-019-07774-z

[13] T. N. Richard P. Verhulst E Marounene Kefi, Hoang. 2018. An evaluation of multimodal interaction techniques for 3D layout constraint solver in a desktop-based virtual environment. *Virtual Reality: The Journal of the Virtual Reality Society* 22, 4 (2018), 339–351. doi:10.1007/s10055-018-0337-4

[14] Frutos-Pascual M. Creed C. Williams I Spittle, B. 2023. A Review of Interaction Techniques for Immersive Environments. *IEEE Transactions on Visualization and Computer Graphics* 29, 9 (2023), 3900–3921. doi:10.1109/TVCG.2022.3174805

[15] Shinjiro Sueda, Andrew Kaufman, and Dinesh K. Pai. 2008. Musculotendon Simulation for Hand Animation. In *SIGGRAPH '08: ACM SIGGRAPH 2008 Papers*. 1–8. doi:10.1145/1399504.136068

[16] Teijo Lehtonen Taneli Nyyssönen, Seppo Helle and Jouni Smed. 2024. A Comparison of One- and Two-Handed Gesture User Interfaces in Virtual Reality–A Task-Based Approach. *Multimodal technologies and interaction* 8, 2 (2024), 10–. doi:10.3390/mti8020010 web.

[17] Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2012. Gestures as Point Clouds: A $P Recognizer for User Interface Prototypes. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*. 273–280. doi:10.1145/2388676.238873

[18] Can Liu Mingming Fan Tianren Luo Zitao Liu Mi Tian Teng Han Feng Tian Xiaohui Tan, Zhenxuan He. 2024. WieldingCanvas: Interactive Sketch Canvases for Freehand Drawing in VR. *Proceedings of the CHI Conference on Human Factors in Computing Systems* (May 2024), 1–16. doi:10.1145/3613904.3642047

[19] Akshay Sharma Yotam Gingold Xue Yu, Stephen DiVerdi. 2021. ScaffoldSketch: Accurate Industrial Design Drawing in VR. *The 34th Annual ACM Symposium on User Interface Software and Technology* (Oct 2021), 372–384. doi:10.1145/3472749.3474756

[20] Shanshan Yang and Ding Liu. 2022. Automatic annotation method of VR speech corpus based on artificial intelligence. *International Journal of Speech Technology* 25, 2 (2022), 399–407. doi:10.1007/s10772-021-09952-7

[21] Xilin Yi and O. I. Camps. 1999. Line-based recognition using a multidimensional Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 9 (1999), 901–916. doi:10.1109/34.790430

A Online Resources

Our code repository is located on GitHub and is available here:

$https : //github.com/csu - hci - projects/CS465_{M}ascarenhasJ_{B}arnardI_{M}aksutaJ$