

NEUROGAZE IN VIRTUAL REALITY: ASSESSING AN EEG AND EYE TRACKING INTERFACE  
AGAINST TRADITIONAL VIRTUAL REALITY INPUT DEVICES

by

WANYEA BARBEL

B.S, Computer Science, University of Central Florida, 2024

A thesis submitted in partial fulfillment of the requirements  
for the degree of Master of Science  
in the Department of Computer Science  
in the College of Engineering and Computer Science  
at the University of Central Florida  
Orlando, Florida

Spring Term  
2024

Major Professor: Joseph J. LaViola

© 2024 Wanyea Barbel

## **ABSTRACT**

Presenting NeuroGaze, a novel Virtual Reality (VR) interface that integrates Electroencephalogram (EEG) and eye-tracking technologies to enhance user interaction within Virtual Environments (VEs). Diverging from traditional VR input devices, NeuroGaze allows users to select and manipulate objects in a VE through gaze direction and cognitive intent, captured via EEG signals. The research assesses NeuroGaze's performance against conventional 3D User Interfaces (3DUIs) such as VR controllers and eye gaze combined with hand gestures. The experiment, conducted with 25 participants, evaluates task completion time, accuracy, and cognitive load through the NASA-TLX survey. Results indicate that while NeuroGaze presents a learning curve, evidenced by longer average task durations, it potentially offers a more intuitive and precise selection method, as suggested by its lower error rate compared to eye gaze with hand gestures. This study highlights the viability of incorporating biometric inputs for more natural and accessible VR interactions. Future work aims to explore a multimodal EEG-Functional near infrared spectroscopy (fNIRS) approach, further develop machine learning models for EEG signal classification, and extend system capabilities to dynamic objects selection, highlighting the progressive direction for the use of Brain Computer Interfaces (BCI) in virtual environments.

Dedicated to all my family, friends and colleagues who never stopped believing in me.

## **ACKNOWLEDGMENTS**

Thank you to my thesis advisor committee – Dr. Joseph LaViola, Dr. Ryan McMahan, and Dr. Mohsen Rakhshan. Thank you to my research assistant, Kyle Coutray, for assistance with the design and implementation of the NeuroGaze system. Additionally, thanks to all the members of UCF’S Interactive Systems and User Experience (ISUE) Lab whose unconditional support was vital to the completion of this work.

Finally, thank you to my loving family and friends for always pushing me to achieve more than I ever could alone.

## TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGMENTS .....	v
TABLE OF CONTENTS .....	vi
LIST OF FIGURES .....	ix
LIST OF TABLES .....	x
LIST OF ABBREVIATIONS.....	xi
CHAPTER ONE: INTRODUCTION .....	1
1.1 Motivation.....	1
1.2 VR & 3D User Interface Interaction Techniques .....	1
1.2.1 NeuroGaze – Eye Tracking & Electroencephalogram .....	1
1.2.2 Eye Gaze & Hand Tracking .....	2
1.2.3 VR Controllers .....	2
CHAPTER TWO: LITERATURE REVIEW.....	3
2.1 Input Modalities in Virtual Reality.....	3
2.1.1 VR Controllers .....	3
2.1.2 Eye Gaze .....	5
2.1.3 Brain Computer Interfaces .....	7
CHAPTER THREE: SOFTWARE & SYSTEM DESIGN – NEUROGAZE.....	11
3.1 System Overview .....	11
3.2 EEG Hardware .....	12
3.2.1 Emotiv Insight II.....	13
3.2.2 Muse II.....	14
3.2.3 Emotiv EPOC X.....	17

3.3 EEG Software .....	19
3.3.1 EmotivPRO .....	19
3.3.2 EmotivBCI .....	20
3.3.2.1 Training Data Collection & Mental Commands .....	22
3.3.3 Cortex API .....	24
3.4 NeuroGaze Integration .....	24
3.4.1 Unity Game Engine .....	26
3.4.1.1 Scenes .....	26
3.4.1.2 VR Player Rig .....	27
3.4.1.2 Eye Interactable .....	29
3.4.1.2 Mental Commands Manager .....	31
3.4.1.3 Interactable Manager .....	33
3.4.1.4 Assessment Manager .....	34
CHAPTER FOUR: USER STUDY .....	35
4.1 Experimental Design .....	35
4.1.1 Procedure .....	35
4.2 Results .....	39
4.3 Discussion .....	40
CHAPTER FIVE: FUTURE WORK .....	42
CHAPTER SIX: CONCLUSION .....	44
APPENDIX A: SAMPLE SURVEYS .....	45
A.1 Qualtrics Demographic Survey .....	45
A.2 Qualtrics NASA-TLX Survey .....	47
A.3 Qualtrics Post Evaluation Survey .....	49
LIST OF REFERENCES .....	51





## LIST OF FIGURES

<i>Figure 1: Electrode locations of International 10-20 system for EEG recording.....</i>	<i>8</i>
<i>Figure 2: Emotiv Insight II (left) and electrode placement (right) .....</i>	<i>13</i>
<i>Figure 3: Emotiv Insight II polymer sensors: three pronged (left), one pronged (right) .....</i>	<i>14</i>
<i>Figure 4: (a) Muse II EEG Headband with sensor locations, (b) electrode locations in International 10-20 system.....</i>	<i>15</i>
<i>Figure 5: Mind Monitor visual of EEG data to be exported, (b) inference performed on Edge Impulse model trained on EEG data from Mind Monitor.....</i>	<i>16</i>
<i>Figure 6: (a) Emotiv EPOC X EEG headset, (b) International 10-20 electrode placement for Emotiv EPOC X.....</i>	<i>17</i>
<i>Figure 7: EmotivPRO data stream playback from Emotiv EPOC X.....</i>	<i>20</i>
<i>Figure 8: EmotivBCI Training Profile Menu, (a) Brain Space Diagram, (b) connected headset and EEG quality, (c) user training profile name, (d) training profile mental commands.....</i>	<i>22</i>
<i>Figure 9: Example Unity Scene - NeuroGaze .....</i>	<i>26</i>
<i>Figure 10: Evaluation Scenes – VR Controllers (left), Eye Gaze combined with Hand Gestures (right)...</i>	<i>27</i>
<i>Figure 11: EyeTrackingRay C# script source code for NeuroGaze Input Device.....</i>	<i>28</i>
<i>Figure 12: EyeInteractable C# source code for NeuroGaze Input Device .....</i>	<i>30</i>
<i>Figure 13: Mental Commands C# script - Connecting to headset and subscribing to data stream .....</i>	<i>31</i>
<i>Figure 14: Mental Commands C# script – Fetching current mental command via Cortex API.....</i>	<i>32</i>
<i>Figure 15: Interactable Manager C# script – Changing cube colors when assessment has started.....</i>	<i>33</i>

## LIST OF TABLES

<i>Table 1: Demographics Survey Results.....</i>	40
<i>Table 2: Evaluation results for each selection technique.....</i>	40

## **LIST OF ABBREVIATIONS**

1. 3DUI - Three-Dimensional User Interface: A user interface that involves interaction with three-dimensional space, allowing users to manipulate 3D objects or navigate through 3D environments using various input devices.
2. AR - Augmented Reality: A technology that overlays digital content (such as images, sounds, text) on the real world, enhancing the user's perception of their surroundings through devices like smartphones, tablets, or AR glasses.
3. BCI - Brain-Computer Interface: A direct communication pathway between a brain and an external device, enabling control of the device using brain signals, often measured via EEG for non-invasive BCIs or implanted electrodes for invasive BCIs.
4. EEG - Electroencephalography: A method to record electrical activity of the brain using electrodes placed along the scalp. EEG measures voltage fluctuations resulting from ionic current flows within the neurons of the brain, widely used in neuroscience, cognitive psychology, and BCI applications.
5. FNIRS - Functional Near-Infrared Spectroscopy: A non-invasive imaging technique that measures brain activity by detecting changes in blood oxygenation and blood volume within the cortex, using near-infrared light.
6. HCI - Human-Computer Interaction: A multidisciplinary field of study focusing on the design of computer technology, and particularly the interaction between humans (the users) and computers. HCI studies how to design, evaluate, and implement interactive computing systems for human use.

7. MR - Mixed Reality: A blend of physical and digital worlds, creating new environments where physical and digital objects co-exist and interact in real-time. MR is often seen as a spectrum that encompasses both augmented reality (AR) and virtual reality (VR).
8. POC - Proof of Concept: A demonstration, typically in the early stages of development, to verify certain concepts or theories can be achieved in development. In technology, a POC tests a prototype of an idea to validate its feasibility, functionality, or potential for further development.
9. VE - Virtual Environment: A digital space generated by computer technology, simulating a real or imagined environment. VEs are interactive, allowing users to navigate and manipulate virtual objects or perform tasks. They are commonly used in VR systems.
10. VR - Virtual Reality: A simulated experience that can be like or completely different from the real world. VR technology creates a fully immersive environment, often engaging sight and sound, and increasingly touch, through specialized hardware like VR headsets, gloves, and treadmills.

# **CHAPTER ONE: INTRODUCTION**

## **1.1 Motivation**

For as long as I can remember, I have been obsessed with the idea of living in a virtual world. A world like the one we currently live in but enhanced in a way that only a truly digital world could. As a mixed reality engineer (MRE), building experience in virtual reality (VR) is the closest I've come to living in this "New World." However, no matter how immersive or present I feel within these virtual worlds, the methods in which I interact with these worlds have always made me feel disconnected from the environment. We don't interact with our current environment by clicking buttons to grab a cup so why should interacting with a virtual environment (VE) be any different?

The discovery of brain computer interfaces (BCI) led me to wonder what applications the technology might have in VEs. Selection within VEs is where I felt the most disconnected with the environment due to traditional 3D user interfaces (3DUI) such as VR controllers. Prior research has been done to show that electroencephalogram (EEG) recordings from the brain and eye gaze can be used to interact with computers [15]. Other research utilizing VR investigate how EEG and eye gaze can be used to evaluate engagement [3] or how eye movement artifacts found within EEG readings during VR simulations can be used to build artifact correction methods [21] Research involving BCI in VR use VR as a medium to simulate environments not easily replicable in the non-VEs in order to record EEG data but do not attempt to segment and classify artifacts in EEG streams with the purpose of selection in VR.

## **1.2 VR & 3D User Interface Interaction Techniques**

### **1.2.1 NeuroGaze – Eye Tracking & Electroencephalogram**

NeuroGaze is a novel 3DUI interaction technique we developed that utilizes infrared (IR) cameras built into a VR head mounted display (HMD) to track where a user is looking in a VE.

NeuroGaze also uses EEG to determine when a user would like to select an object once they have both looked at the object and thought about selecting it. This technique aims to provide users with a more intuitive experience when selecting objects in VR while continuing to maintain the level of reliability users would expect when using traditional interaction methods (e.g. VR controllers).

### **1.2.2 Eye Gaze & Hand Tracking**

Eye gaze and hand tracking is a selection method that uses eye tracking for users to “hover” over objects they would like to select and hand tracking to select or interact with objects. The “trigger” for these hand tracking actions is usually done utilizing gestures. Gestures use specific hand movement and positions to specify a command in a VE [16]. Specific to our evaluation, participants used a pinch gesture to select objects within the VE. This technique is used by the Apple Vision Pro for all selection within the AR environment.

### **1.2.3 VR Controllers**

VR controllers are handheld devices that allow users to interact with VE in an intuitive way. They are designed to be used as selection devices enabling users to navigate through menus, select items and interact with objects within a VE. For selection, participants will use a point and click interaction technique where a ray will be cast from the end of the VR controllers that participants can use to determine where they can interact with the VE and the controller triggers are used to selection objects. VR controllers allow users to have precise control over selection and manipulation tasks but can be physically tiring when used for a longer duration [17].

## CHAPTER TWO: LITERATURE REVIEW

### 2.1 Input Modalities in Virtual Reality

#### 2.1.1 VR Controllers

Early VR systems relied on basic input devices such as keyboards, mice, or joysticks, which did not provide a natural or intuitive way to interact with VEs. The development of dedicated VR controllers marked a significant advancement, offering users a more immersive and hands-on VR experience. Point-and-click, like a mouse and keyboard in traditional computer interfaces, is a widely used interaction technique in VR. Studies have investigated the efficiency and accuracy of point-and-click interactions using VR controllers, highlighting their effectiveness for selection and manipulating objects in 3D space. Over the years, research has been done expanding on these devices in new and interesting ways with the goal of building upon the initial VR controllers.

An example of this can be seen in Whitmire et al [23] who developed a unique handheld controller, denoted the *Haptic Revolver*, utilizing an actuated wheel. As participants spin a haptic wheel on the controller to move around the VE, the controller provides users with tactical feedback that can be interchanged with different wheels. This allows users to experience a variety of accurate haptic information depending on the VR scene. Although haptics is outside of the scope of this Thesis and the NeuroGaze system does not include haptic feedback, it is important to recognize the use of haptics to provide users with a more intuitive and immersive experience.

Fahmi et al. [7] investigated the overall user experience of VIVE VR controllers when compared to a Leap Motion Controller (LMC) and Senso Gloves. LMCs use an infrared monochromatic camera that periodically takes pictures [7]. An observation area is set and when a user's hand enters this area, the hand gestures are recorded, and an action is performed in VR. Senso Gloves are wireless devices that use

Inertia Measurement Unit (IMU) sensors to observe hand and finger movement. This device also includes haptic feed in the form of vibrators in the fingertips, back of hand and wrists [19]. When a user experience study was conducted, high Likert scale values for the VIVE VR controllers suggested that participants felt that VR controllers were very interactive, and the controls were easy to learn and use. When ideating on how to use EEG effectively as a part of the NeuroGaze input device, the first challenge was how to leverage a user's previous experience with input devices such as VR controllers. Although NeuroGaze does not use VR controllers, the system should still feel intuitive and easy to use. Studies like Fahmi et al. showed me that if I was able to make NeuroGaze as highly interactive, intuitive, and satisfying to use as VR controllers then participants would report a positive user experience. User experience when compared to VR controllers is only half of the battle, the input system should also perform tasks successfully and effectively. Luong et al. [18] evaluates the interaction performance of VR controllers when compared to hand controllers. Both input devices utilize the virtual pointer metaphor which alleviates a major shortcoming of both input devices: interacting with objects that are out of reach [2]. This study showed that for the interactions, participants preferred VR controllers with ray cast because of the precision they could get. Additionally, these participants performed better when evaluated on both speed and accuracy than they did while using the virtual hand. It was decided that NeuroGaze would use some type of ray casting as the interaction metaphor of choice due to its speed and intuitiveness to use. However, a common complaint with using the ray pointing with VR controllers for longer durations is physical strain. In research evaluating this input device and metaphor, participants start to hold the controller closer to their body to reduce fatigue [2]. For this reason, it was determined that the NeuroGaze system would not be using VR controllers despite its intuitiveness to use. Looking deeper into how prolonged use of input devices can cause fatigue, I investigated input devices that were compatible with the ray casting metaphor [2] but did not induce as much physical effort as VR controllers



over longer durations of time. The next steps in developing the NeuroGaze system pointed towards eye tracking as a possible input device for VEs.

### 2.1.2 Eye Gaze

Eye gaze uses eye tracking technology to measure a user's gaze direction, or where the person is looking, and the movement of the users' eyes. This input method is usually achieved with infrared cameras that illuminate the eye by refracting light off the cornea and the retina to create distinct patterns, called glints. Software like RemoteEye developed by Hosp et al. [8] uses glint detection algorithms to calculate the gaze point and achieve high speed eye tracking capabilities. When applied to VR as an input device in VEs, an entire genre of interaction technique presents itself. Novel techniques like the ones evaluated by Piumsomboon et al. [20], show us the versatility of the eye gaze input device. In this evaluation, three selection techniques based on eye gaze were evaluated. The first, *Duo-Reticles*, uses one eye-gaze and one inertial reticle to perform selection. *Radial Pursuit* specializes in selecting cluttered objects with smooth pursuit. *Nod and Roll* is a hand gesture-based interaction based on the vestibulo-ocular reflex. These techniques in combination highlighted the four primary natural types of eye movements [23] that should be studied when constructing an eye gaze-based technique for selection. These include (1) *saccade*, a quick eye movement with a fixed end target, demonstrated by Duo-Reticles, (2) *smooth pursuit*, a smooth eye movement towards a moving target, demonstrated by Radial Pursuit, (3) vestibulo-ocular reflex (VOR), an automatic eye movement that counters head movement when a user's gaze is fixed on a target, demonstrated by Nod and Roll and finally (4) vergence, when our eyes converge or diverge to look at targets of varying distance. Participants were tasked with finding a matching picture as quickly and accurately as possible. Participants were then asked to answer a 7-point Likert scale usability questionnaire that asked participants to rank the input devices with statements like “It felt natural the use” and “I felt satisfied using it.” The results showed that the Duo-Reticle was highly favored even if the results did not yield a high difference compared to the other input devices. When considering eye gaze

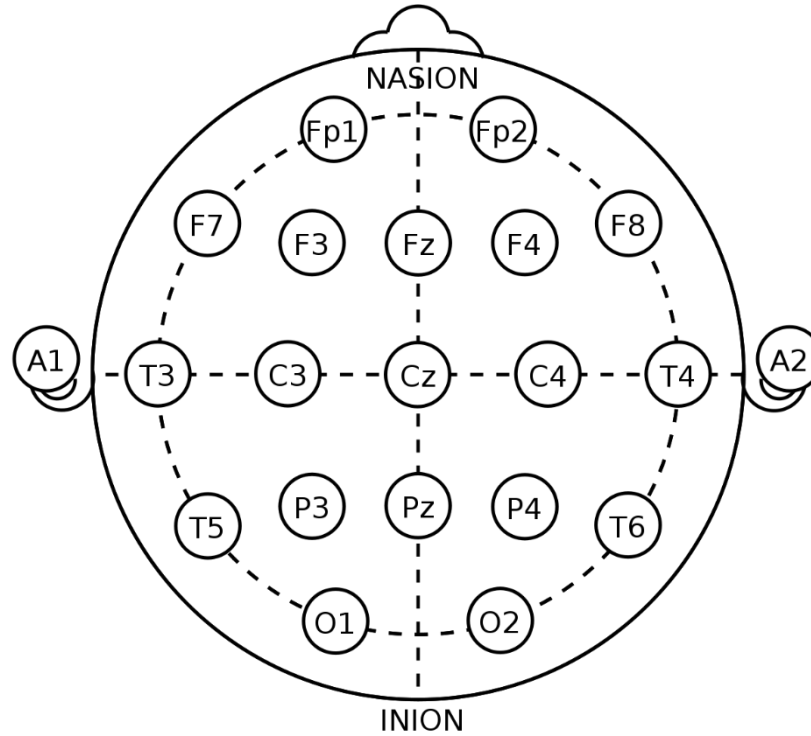
and a part of the NeuroGaze input device, we needed to decide not only what technique to use with this input device but also which would be best for our experimental design. At an early stage in our experimental design, we wanted to focus on moving targets but as we learned more about EEG, we realized that limiting kinematic activity, especially head movement, would be greatly beneficial to the success of a BCI. The evaluation of input methods that take advantage of saccade [9], such as Duo-Reticles, showed us how NeuroGaze could take advantage of the eyes natural ability to rapidly update to new stationary targets. With this in mind, we were curious how quickly the eye could update to a new stationary target when compared to existing fast input methods like VR controllers.

Francisco et al. [17] evaluated VR controller accuracy when compared to an eye gaze technique using previously mentioned ray casting metaphor to perform aiming selection tasks [2]. To evaluate how demanding the input devices were, a raw NASA-TLX survey was distributed to participants. A system usability scale (SUS) survey was used to Participants were asked to complete an eye tracking calibration test before using their eye gaze to look at a target of interest and use VR controllers to select the object. The same evaluation was performed with VR controllers with a point and click technique. The evaluation results showed that the eye gaze input method performed on par with VR controllers. The survey results showed participants felt that the eye gaze input method produced a lower Physical Demand and Effort than VR controllers. At this point in our literature review we were able to determine that eye tracking is not only as performant as VR controllers, but also produces less physical demand and effort than VR controllers when selecting fixed target. Under varying fields of view (FOV), eye gaze even performed over head-gaze based selection [17, 23]. Based on our research, eye tracking seemed to be the most precise and intuitive way to allow users in a VE to show intent for selecting an item. With BCIs still in our minds, we turned towards research in this area and the best techniques, hardware, and strategies for integration with eye gaze given the limitations and scope of this Thesis.

### **2.1.3 Brain Computer Interfaces**

Brain computer interfaces harness neural signals that are usually translated into commands that can control some computer system. The area of study has broadened itself in the last 15 years and its applications have increased in range including technologies like robotics, Internet of Things (IoT) , and VR [24]. In recent years, invasive neural interfaces have seen human testing, such as intracortical recordings. This process involves implanting neural interfaces composed of biocompatible materials with the goal of curing existing limitations of a participant or enhancing them [14]. While it is important to be cognizant of invasive interfaces, this scope of this Thesis is strictly focused on signal acquisition using non-invasive methods (e.g. EEG).

Electroencephalogram or EEG is a non-invasive recording of electrical activity along the scalp. Specifically, EEG is the measure of the voltage fluctuations resulting from ionic current flows within the neurons in the brain. [12]. EEG data has been used in a variety of uses cases such as medical diagnostics for neurological disorders [10], cognitive and behavior and behavior research and BCIs. With a multitude of areas of research using EEGs, standards must be practiced when placing electrodes on the scalp or it becomes difficult to compare EEG data across evaluations. The collection of EEG data is dependent on the electrode system chosen.



*Figure 1: Electrode locations of International 10-20 system for EEG recording*

The 10-20 electrode system, or International 10-20 system, is an internally recognized method to describe and apply the location on scalp electrodes in the context of an EEG test or experiment [12]. In *Figure 1*, we can observe a top-down view of the head where the Nasion, the depressed area between the eyes, is the front of the head, and the Inion, the crest of the back of the skull, is the back of the head. The “10” and “20” refers to the 10% or 20% of spacing between each adjacent electrode, starting from ear to ear. The starting letter is used to identify a lobe or area of the brain: central (C), occipital (O), parietal (P), temporal (T), and pre-frontal (Fp). The following number represents the side of the brain where even numbers (2,4,6,8) represent the electrodes on the right side of the brain and odd numbers (1,3,5,7) represent the left side of the brain. With standardized electrode placement, researchers build upon previous works with little variances in electrode placement on a participant's scalp. It should be noted that other systems exist such as 10/5, 10/10 exist are not internationally recognized nor do they have noticeable performance increase when evaluating quality of the EEG data [10]. The 10-20 system is the

current standard when collecting EEG data and NeuroGaze is compliant with this standard. Until other electrode placement standards are established or new positioning systems [11] are created and tested through roughly, we do not plan to diverge from this standard.

Larsen et al. [15] recently published a paper evaluating selection in VR using a similar input method as NeuroGaze. The paper showcased a proof of concept (POC) called *SSVEP-Speller* which uses eye trackers to make a preselection of a subsection of a virtual 3D keyboard to select a letter. EEG is used to measure the temporal difference between the time it takes to completely close the eyes during a blink. Electrodes were placed over the occipital region at O1, Oz, O2, P3, P7, Pz, P8 and P4 to record EEG data of the visual stimuli. To collect data for their temporal blink classifier, participants were instructed to wear the EEG headset while holding a VR headset to their head and blink naturally with both eyes at about a rate of 60 beats per minute (BPM). During evaluation, participants were tasked with looking at letters they wanted to select while varying levels of flickering occurred, turning their screen completely black, to simulate a blinking sensation. The results showed that these researchers were successfully able classify intervals of blinking to minimize jittering that occurs during blinking during selection with eye gaze. This paper educated us on the effects of noise in an EEG data stream because of blinking. While NeuroGaze does not utilize a pretrained machine learning model for classifying noise caused by blinking, this paper solidified our decision to find a software that would handle this filtering for us. In initial pilots of the NeuroGaze system we did experience issues with jittering due to blinking due to this being handled for us in the Meta Quest Pro HMD. Future work, independent of the Meta Quest Pro HMD, could benefit from software like Blink [1], designed to manage noise potentially interfering with eye tracking data.

Putze et al. [22] presents research closest to our latest version of the NeuroGaze input device. Interestingly, their selection device was designed for use not in immersive or 3D environments typical of VR, but rather in 2D or screen-based environments. This distinction is crucial, as it emphasizes the different challenges and interaction dynamics present in non-immersive versus immersive settings. Their

system, denoted *EEG+GAZE*, consists of Tobii X60 eye tracker with a sampler frequency of 60Hz and a BrainProducts 32 electrode EEG cap. Participants had 29 electrodes positioned respective of the 10-20 system at the following locations: Fp1, Fz, F3, F7, FT9, FC5, FC1, C3, T7, CP5, CP1, P3, Pz, P7, O1, Oz, O2, P4, CP6, CP2, Cz, C4, T8, FT10, FC6, FC2, F4, F8, FP2 and reference electrode positioned at Fz. EEG+GAZE used gaze to detect when a user is likely and unlikely to select an object. This is achieved by monitoring when a user is slowly moving their eyes towards an area of interest versus quickly scanning the screen. During the window where a user's gaze seems like they want to interact with an object, the EEG headset records their temporal window. Data to train the classifier for the eye gaze and EEG, participants were tasked with following key on a monitor with their eyes and in the moments where the objects speed increased and slowed down, the data coming from these two devices created a window of interest. The evaluation consisted of 10-fold cross validation for each participant individually and averaged the result. The data shows that their classifier produced a lower  $F_{0.2} - Score$  the lower the tolerance measured in number of window and milliseconds. This suggests that their EYE+GAZE systems classifier had very limited false positive and thus more accurate classification for event-based evaluation scenes [25].

This literature review has shown the process of deciding the design of the NeuroGaze system by breaking apart this new input device's subcomponents. VR controllers presenting us with the ray casting interaction metaphor, eye gaze provided us with comparable saccade interaction methods, like Duo-Reticles, and evaluation with fixed points in a 3D environment for our evaluation of the system. Finally, an overview of the research done in BCI in 2D, and 3D environments shows us that it is possible to classify these variances in neuron voltage. Putting this all together, we get the NeuroGaze input device that aims to challenge how quickly, accurately, and intuitively a 3DUI input device can be in VR.

## **CHAPTER THREE: SOFTWARE & SYSTEM DESIGN – NEUROGAZE**

### **3.1 System Overview**

NeuroGaze is an input device proof of concept (POC) specifically developed for selection in a VE. The objective of NeuroGaze is to empower users with a selection device that allows them to interact in VR in a new way to improve accuracy and efficiency when compared to traditional input devices such as VR controllers. Our system uses eye trackers built into the Meta Quest Pro HMD to activate a “hover state” on objects in the scene they can interact with. When a user's eye gaze is broken by one of these interactable objects, the object slowly and continuously grows until it reaches its max size. Similarly, when a user's eye gaze moves off an interactable object, the object will shrink at a slow continuous speed until it is returned to its original scale. The users eye gaze interacts with objects in the scene using the ray cast interaction metaphor and a white line renderer to represent the midpoint between each of the users' eyes at any moment. NeuroGaze uses the Emotiv EPOC X EEG headset for the EEG data stream and the EmotivBCI program for creating training profiles that collect EEG data and handle EEG noise sanitization and EEG artifact classification. The Unity Game engine and Meta XR All in One SDK were used to complete the development of NeuroGaze, integrating the hardware components and software solutions to create a seamless and intuitive user experience within VE. This integration allows NeuroGaze to provide a hands-free, efficient, and precise method for interacting with and selecting objects in VR, showcasing a significant improvement over conventional input devices in terms of user engagement and interaction fidelity. The implementation of NeuroGaze represents a novel approach to VR interaction, leveraging the precision of eye tracking and the sophistication of EEG data analysis to facilitate a more natural and immersive experience. By utilizing the Emotiv EPOC X EEG headset and EmotivBCI software, NeuroGaze effectively minimizes noise and accurately classifies user intentions based on neural

activity, demonstrating the potential of combining multiple biometric inputs for enhanced control within a VE.

Throughout the development process, a user-centered design philosophy [14] was emphasized, ensuring that NeuroGaze not only advances the technological capabilities of VR interfaces but also addresses the practical needs and preferences of users. By focusing on ease of use, accessibility, and the reduction of physical exertion, NeuroGaze aims to make virtual interactions more enjoyable and less tiring, particularly for users who may find traditional input devices cumbersome or completely inaccessible.

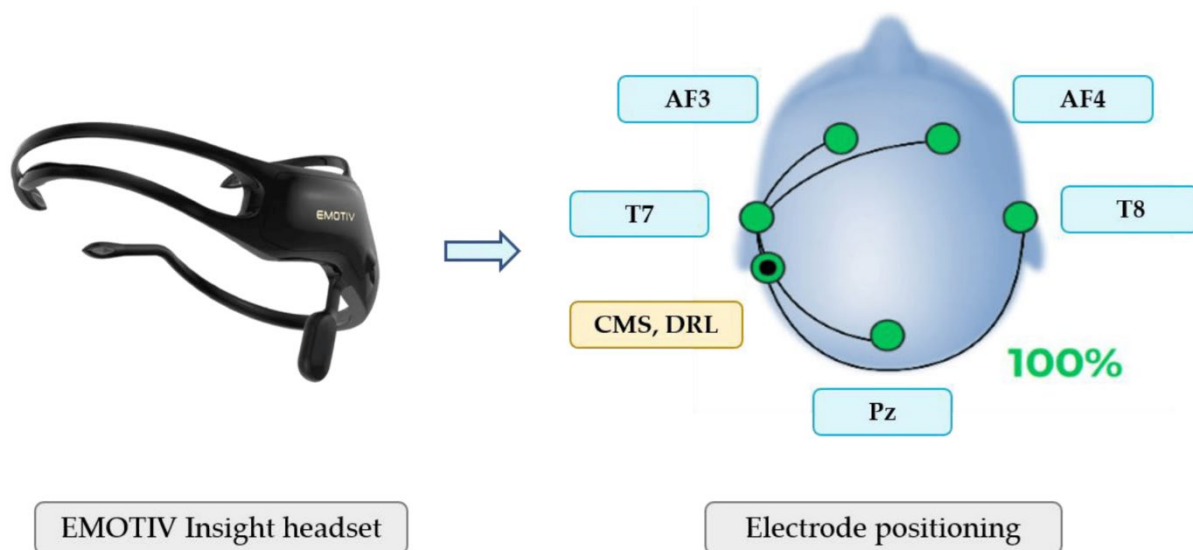
### **3.2 EEG Hardware**

When deciding which EEG headset to use for the NeuroGaze system, our immediate concern was ensuring we chose a headset that gave us a high-fidelity EEG stream that was budget efficient. We wanted to keep our budget in the \$1000 range or less due to this Thesis not being funded. Additionally, from an ergonomics perspective, we needed to ensure that the EEG headset could be worn underneath the Meta Quest Pro VR HMD. If possible, we also would prefer the electrodes to be in a different position than the position where the VR headset would be holding the user. Although we are limited by these two pieces of hardware not designed for one another, if we can reduce pressure points on the user's head, this would greatly increase comfort and reduce unnecessary artifacts from discomfort. Getting data off these headsets was also a factor we had to consider. How many channels do we need to access? How does the EEG data export and does the headset interface with any existing data collection, synthesis and machine learning software? All these unknowns needed to be investigated and the most “out of the box” functionality we could get for our hardware POC was ideal. We reviewed many different EEG headsets, but the following were the three we narrowed down for the final design for the NeuroGaze input device.



### 3.2.1 Emotiv Insight II

Emotiv is a pioneer in the field of neuroscience with its development of high-quality, accessible EEG technology. Their products are used across the world for research, neuroeducation, and brain computer applications, focusing on producing comprehensive brain monitoring solutions. Emotiv's commercial EEG headset was not only budget friendly but also would aid with tracking the quality of the EEG data we were collecting through the EmotivPRO and EmotivBCI programs. The Insight II model from Emotiv has 5 channels with 2 reference sensors, focusing on key areas of cognitive state monitoring. The polymer sensors are arranged according to the international 10-20 system and the location of the sensors are in the AF3, AF4, T7, T8 and Pz positions as seen in *Figure 2*.



*Figure 2: Emotiv Insight II (left) and electrode placement (right)*

*Source: Suárez, M. (2023)*

These sensor locations focus on the frontal and parietal areas which are important for cognitive and emotional insight. This headset has a sampling rate of up to 128HZ and connects to any computer that supports a Bluetooth 4.0 connection. We used this headset for a couple weeks, and while the contact quality was good, this headset was very uncomfortable to use for a longer period. This is partially

because the Meta Quest Pro HMD distributes its weight to the front of the headset and the AF3 and AF3 electrodes are in this same position. Additionally, these two nodes were required to be a three-pronged contactor (See *Figure 3*) which was very uncomfortable and sometimes even painful to wear for a prolonged period while wearing the Meta Quest Pro.



*Figure 3: Emotiv Insight II polymer sensors: three pronged (left), one pronged (right)*

For this reason, we removed this headset from our apparatus and looked for alternatives for the NeuroGaze input device.

### **3.2.2 Muse II**

Muse is another contributor to the neurotechnology space, offering devices that promote mental wellness through guided meditation and biofeedback. They offer the Muse II, a commercial EEG headband that is the more cost efficient of the three EEG devices we were interested in using. Additionally, this device is a hand band not a headset so the device did not have any nodes or prongs that could be uncomfortable to wear. The EEG headband consists of 5 EEG sensors along the forehead (3 reference sensors, 2 EEG sensors), one grounding sensor behind each ear and a heart monitor (See *Figure 4*). The Muse II is worn like a pair of glasses making it perfect when paired with the Meta Quest Pro HMD and although there are

sensors on the front of the headset in the forehead area, because these sensors are flat, they did not cause discomfort.

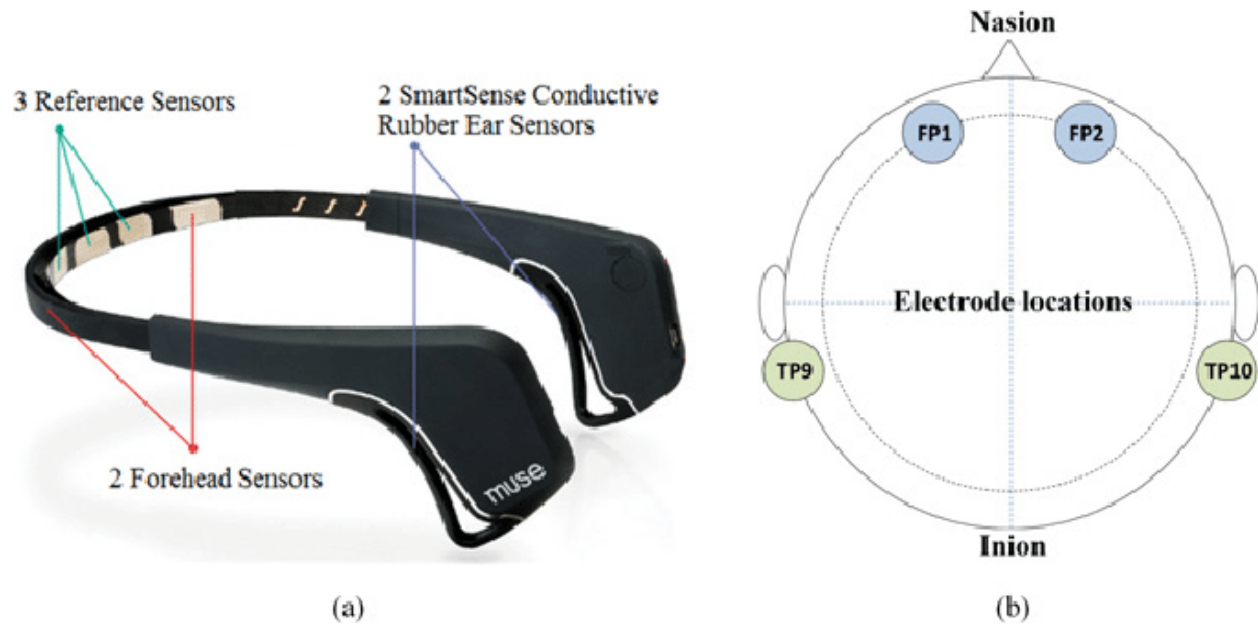


Figure 4: (a) Muse II EEG Headband with sensor locations, (b) electrode locations in International 10-20 system

Source: Liu, R. (2020)]

To pull the stream and visualize the EEG data from the headset and store it, we attempted to use the open-source application Mind Monitor (See Figure 5). To classify data, we used Edge Impulse to import the raw data recorded from Mind Monitor and construct a neural network classifier. We collected data on thinking right, left and neutral in 5 second increments per thought. The data is stored as one CSV per session that Edge Impulse can use for an 80/20 training-testing split. Once the model trained on our data, the model performed inference with our testing split with a highest accuracy of 40%. After testing this pipeline further, we concluded that the noise artifacts impeded within the EEG data were clouding the data too much to classify intent with any level of confidence.

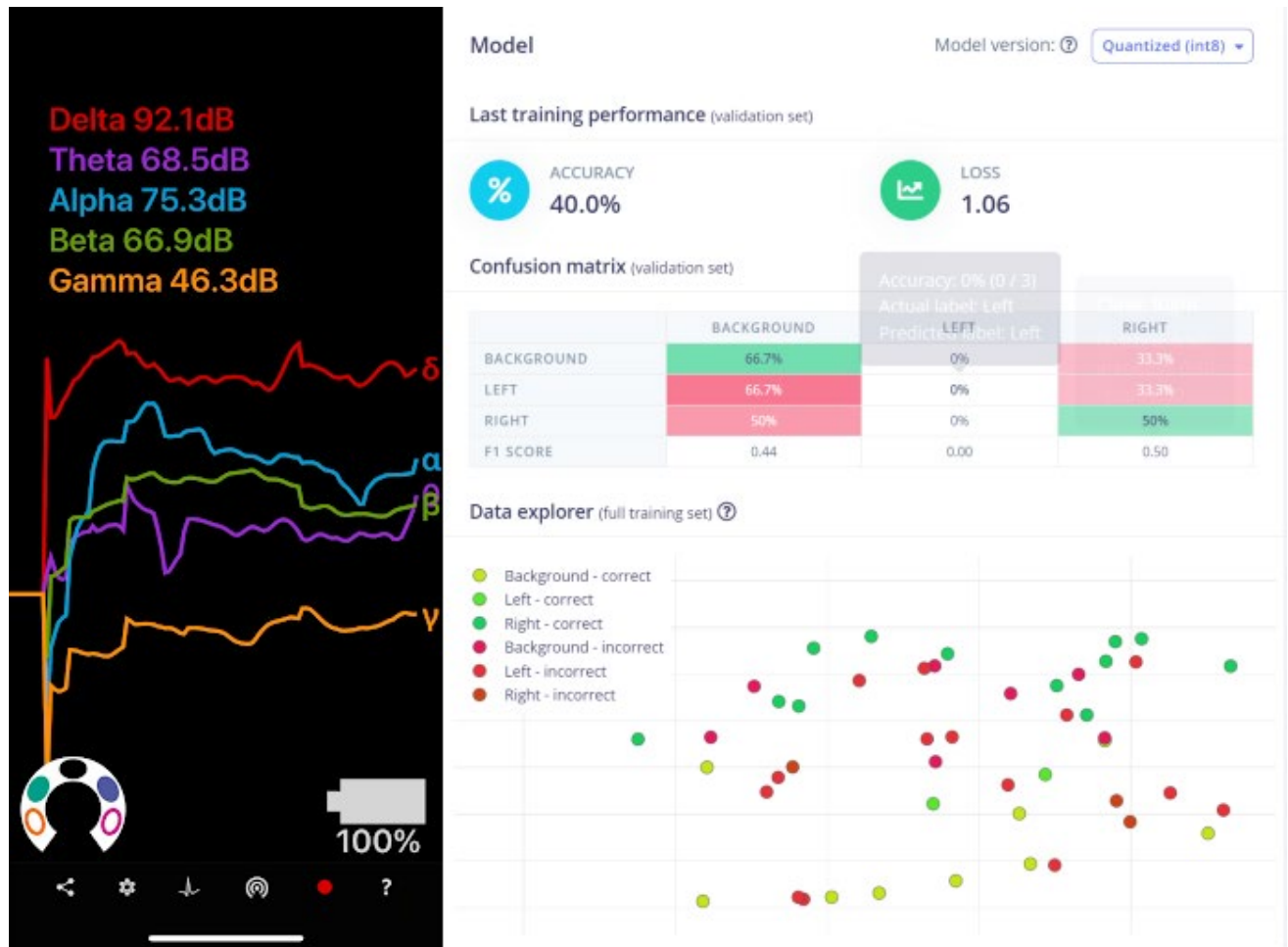


Figure 5: Mind Monitor visual of EEG data to be exported, (b) inference performed on Edge Impulse model trained on EEG data from Mind Monitor

The drawbacks of this headset are the limited channels we have access to for streaming EEG data, additionally the Muse SDK lacks any functionality for cleaning or classifying the data. If we selected this device, we would be required to collect ample data to train our own machine learning classifiers for identifying noise artifacts (e.g. eye blinks, random kinematic movements) and identifying when a user wants to interact with an object in the scene. Due to time restrictions and the POC nature of the NeuroGaze input device, we did not want to dedicate resources to machine learning classifiers and instead wanted to focus on finding the best configuration and techniques to use. For this reason, we chose not to

use this headset and looked towards less cost efficient Emotiv headsets that can handle our data processing pipeline.

### 3.2.3 Emotiv EPOC X

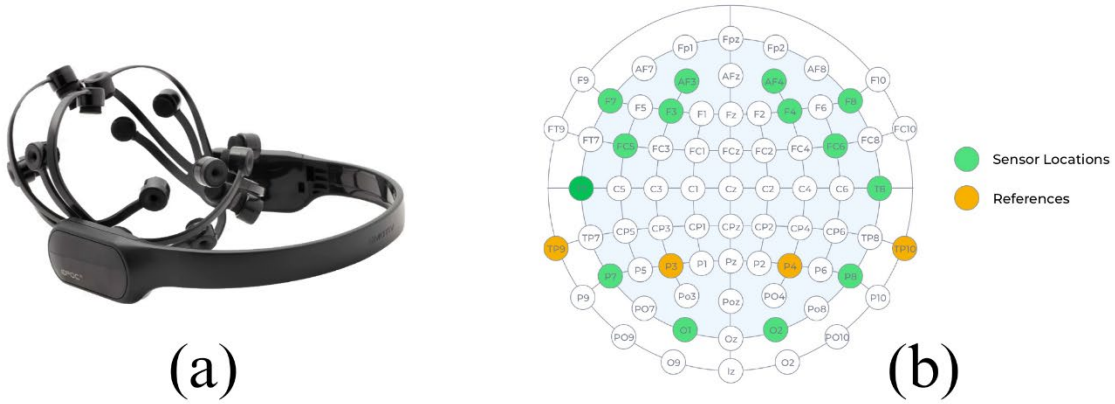


Figure 6: (a) Emotiv EPOC X EEG headset, (b) International 10-20 electrode placement for Emotiv EPOC X

When choosing the EEG headset for the NeuroGaze system, considering extended beyond technical specifications to consider factors such as user comfort, integration ease, and application relevance. The Emotiv EPOC X stands out for its comprehensive 10-20 system channel coverage (sensors: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4, references: TP9, P3, P4, TP10) and an extensive software ecosystem (See Figure 6b). Although this headset is the most cost inefficient of the three candidates, we decided it was well worth it. Despite its functionality and integration possibility while being a commercial headset, the location of its electrode connections makes it ideal for compliance with the ergonomics of the Meta Quest Pro VR HMD. Additionally, the nodes are made of felt and saline solution can be added through the refillable sensor hole located on each sensor (See Figure 6a). This design allows for maximum comfort, even at pressure points, and easy access to wetting dried out electrodes with saline solution even after the user has configured the NeuroGaze system. During evaluation participants are told to think of a motor command they would physically perform if they weren't stationary in a VE. With this

in mind, we wanted to understand how effective the Emotiv EPOC X would be with classifying motor imagery and how internal motor act rehearsed in working memory [5]. Fakhruzzaman et al. [6] used the Emotiv EPOC X to attempt to classify motor imagery using BCI scenarios. Participants were told to imagine the motor action they would prefer during the following testing scenarios:

1. Left Hand Movement
2. Right Foot Movement
3. Simultaneous Left Hand and Right-Hand Movement
4. Simultaneous Left-Hand Movement and Head Nodding
5. Simultaneous Right Foot and Left Foot Movement
6. Simultaneous Right Foot Movement and Head Nodding

When attempting to classify the EEG data that correlate to these motor imageries, the Scenario 1 test results had a 76.67% success rate, Scenario 2 test results showed a 91.67% success rate, Scenario 3 yielded a 28.33% success rate, Scenario 4 get 13.33% success rate, Scenario 5 showed 60% success rate and Scenario 6 success rate was 53.33% success rate. The main take aways from this study is that the Emotiv EPOC X could not handle complex motor imagery classification so we would need to limit participants to a single, consistent thought (e.g thinking of selecting objects with their dominant hand). Additionally, the paper commented that while the classification for these tasks is not amazing, for a consumer grade product, the EPOC X is sufficient for POCs but further investment into a research question, more less commercial hardware should be used. Given our limitations, it was decided that for this Thesis, the Emotiv EPOC X would be sufficient for our use case and based on the results, further financial investment would be made.

### 3.3 EEG Software

When deciding on which EEG software to use, we prioritize software that could integrate well with the Unity Game Engine. We needed the classification of user's intent to select an object to selection an object in the VE to have as little latency as possible. This is because eye tracking will be used for a hover state and if the classification is too delayed, the user could be looking at another object and accidentally select an object they didn't mean to. At this point, we had already decided that we were going to use the Emotiv EPOC X as our EEG device and Emotiv already has software built to stream, clean, and train the EEG data coming off their headsets. Using the EmotivBCI program, we were able to make "training profiles" to associate EEG data with a user and train the model on specific "mental commands." The Emotiv Launcher handled authorization of our Unity programs' access to these training profiles and all the data associated with it. With most of the software development efforts of building our own classification model alleviated, we could focus on integrating our Unity application with Emotiv's Cortex API.

#### 3.3.1 EmotivPRO

The EmotivPRO software handles the raw stream of EEG data coming from the EPOC X. This program also handles EEG preprocessing. All our raw EEG data is cleaned with a high pass filter using this program. All 14 channels on the headset can also be played back (See *Figure 7*). The raw stream is measured in microvolts converted directly from a 14-bit unsigned integer. The program has performance metric data that classifies and charts emotional states (e.g. Attention, Engagement, Excitement, Interest, Relaxation, Stress). We do not these performance metrics in the NeuroGaze system, but it could be interesting to investigate in future works. This program was critical in handling the initial stages of the software pipeline and allowed us to remove unwanted artifacts from our data like blinking and small kinematic movements Although this software does not give us direct access to training specific artifacts to

remove from our EEG data stream, the EmotivPRO software is great for POC type projects where little budget and short time are both major contributors.

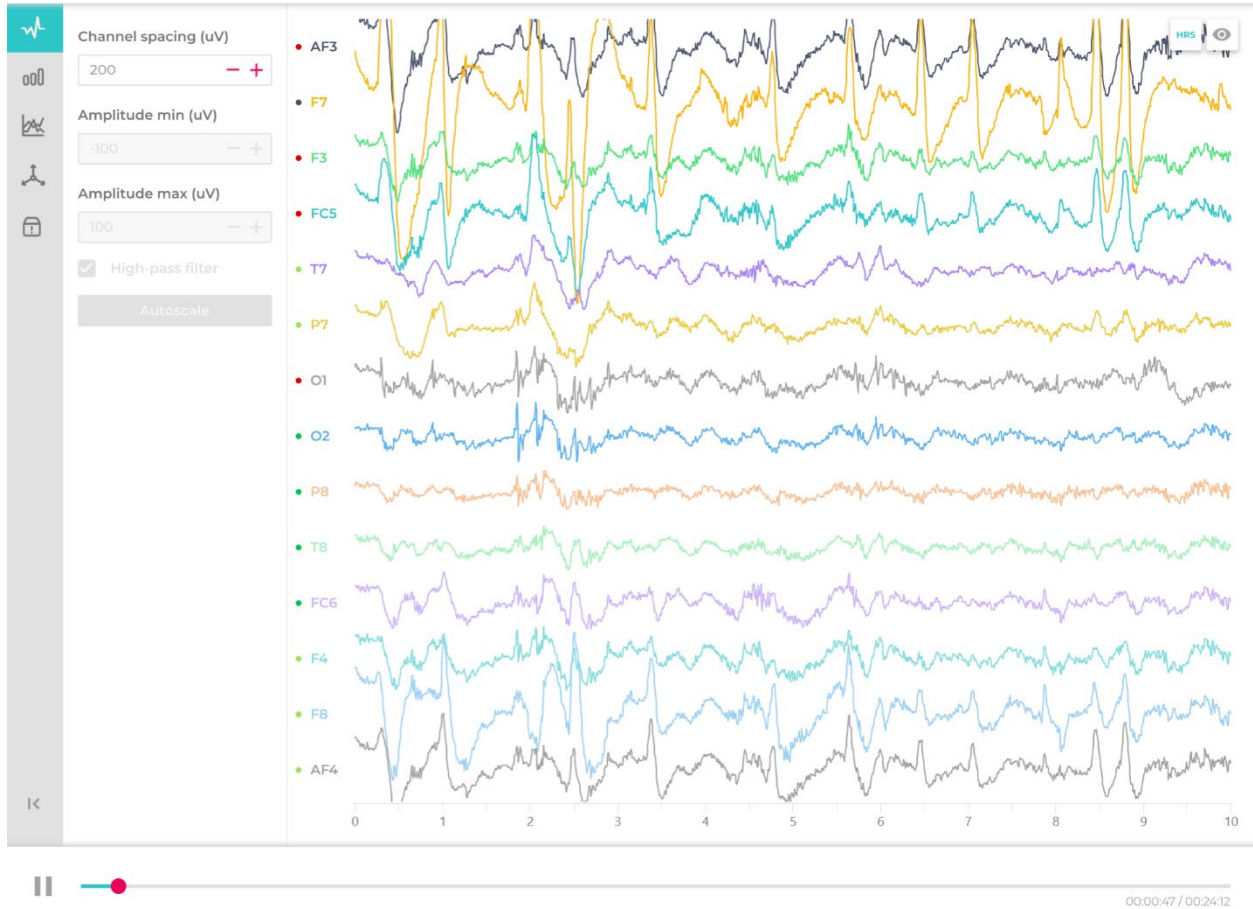


Figure 7: EmotivPRO data stream playback from Emotiv EPOC X

### 3.3.2 EmotivBCI

The EmotivBCI program is the main piece of software our Unity program interfaces with. This program handles training profiles for each of our participants with a unique identifier (See *Figure 8c*) that we can use to load any training profile into our Unity program. Before we begin training any of these profiles, we must check the contact and Overall EEG Quality currently being streamed from the headset. For each node, the contact and EEG Quality (EQ) is calculated individually and is ranked on the sensor map as follows: black means very bad or no contact or EEG signal is detected from this node, red means



the contact or EEG signal is bad, orange means the contact or EEG signal is average, green means the contact or EEG signal quality is good (*Figure 8b*). Contact quality is just the measurement of impedance of the channel for that node or channel. This measure and the EQ are not the same, however the EQ can only be as good or less than the contact quality so this value should be 100% all the time to ensure the highest quality EEG stream. EQ is a bit fickle and achieving a 100% EQ like contact quality is virtually impossible. This is because EQ is a machine learning trained algorithm that outputs a percentage based on the following metrics:

1. Contact Quality (CQ): impedance measure indicating the quality of the electrical signal passing through the sensors and reference (0 or black – very bad, 1 or red – bad, 2 or orange – average, 4 or green – good)
2. ML Signal Quality (SQ): a machine learning algorithm trained on high quality EEG recordings collective by the Emotiv Research team, Data is accessed over 2 seconds increments (0 or black – very bad, 1 or red – bad, 3 or light green – good, 4 or green – great).
3. Signal Magnitude Quality (SMQ): a measure of a single's amplitudes. Even if CQ and SQ are high, but the signal amplitude can still be low which would lead to an undetectable Fast Fourier Transform (FFT). (0 or black – very low, 1 or red – low, 4 or green – average).

The color corresponding to the minimum value out of these three metrics is display in the EmotivBCI EQ user interface for each mode (See *Figure 8a*). The Overall EQ is represented as a percentage (See *Figure 8b*) calculated by taking the minimum three EQ of all available nodes, summing them up and diving by the maximum EQ of the same three nodes (value of 4 for CQ, SQ, or SMQ) The Overall EQ can also be presented as an equation:

$$Overall\ EQ = \left( \frac{min1 + min2 + min3}{max\ possible\ score\ for\ 3\ channels} \right) \times 100 = \left( \frac{min1 + min2 + min3}{12} \right) \times 100$$

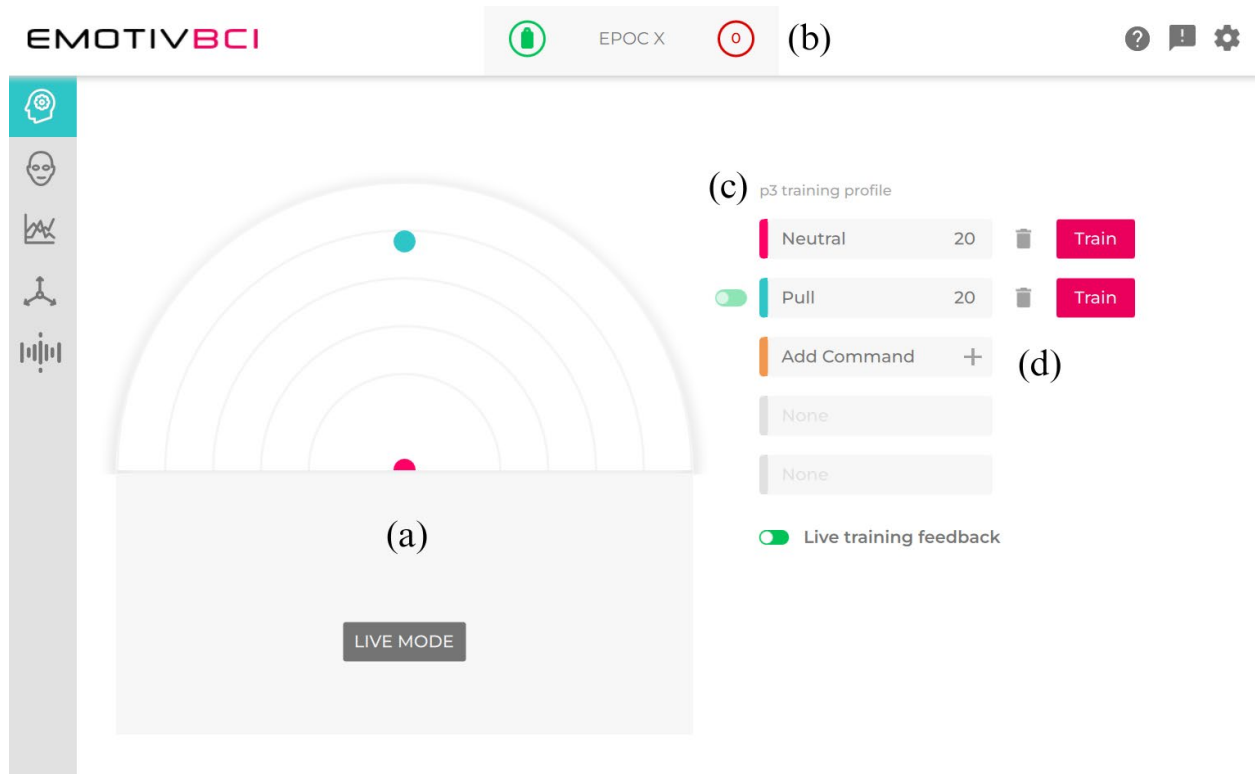


Figure 8: EmotivBCI Training Profile Menu, (a) Brain Space Diagram, (b) connected headset and EEG quality, (c) user training profile name, (d) training profile mental commands

### 3.3.2.1 Training Data Collection & Mental Commands

After confirming our contact quality and Overall EEG Quality is high, we begin training a profile on mental commands. A mental command is a recognized pattern in brain activity. When a mental command is triggered, you can trigger a response and they are at the core of the NeuroGaze system. Although these mental commands have a set name (*Neutral*, *Pull*, *Push*, *Lift*, *Drop*, etc.), if the same thought is consistently producing a recognizable pattern in brain activity, it does not matter if you're thinking of something different than what the mental command describes (e.g. thinking of making an item shrink when training the *pull* mental command). Before you can train other mental commands, you must start by training the *neutral* mental command. The *neutral* mental command is the baseline state that represents the brain activity when the mind is relaxed, at ease and not focusing on a particular thought.

When training a mental command, the goal is performing a thought that is both reproducible and separable from other mental commands. When training multiple mental commands, it becomes harder to distinguish between them. To alleviate this, Emotiv recommends linking each mental command to a single and unique kinematic modality. An example of this could be visually seeing an object move up during a lift command training, hearing an object move towards you during a *pull* command training and gripping your fists during a drop command training. Additionally, if you have disposition towards a single modality, picking a unique thought within that modality can also be very effective (e.g. a musician linking a mental command to thinking about what sounds different notes make). Mental command training sessions last eight seconds after the “Train” button is clicked in which a training screen will display showing a cube performing the mental command you chose and a timer telling you how much you have left in the session. After two training sessions, the EmotivBCI app provides feedback on how good the training was on a scale from 1-100 based on previous training sessions. After a session, you will see a number representing how good the session was with a goal of 75 out of 100. At this point, you can accept the training or reject it and try again (Figure #). During the NeuroGaze evaluation, only training sessions with a value of 75 or higher were accepted. Additionally, each mental command was trained for 20 high quality training sessions. Due to the nature of the evaluation, we only needed two mental commands: *neutral* and *pull*. The *neutral* state is represented by the user’s mind when it is calm and wondering, the user does not want to select an item during this mental state. The *pull* mental command is used to select items of interest (objects that have met their eye gaze) in the scene. Users were prompted to think about a particular action and thought of their choosing. Based on the research done by the Emotiv Research team, we believed it would be unwise to attempt to standardize these thoughts and instead leverage the unique experiences of each participant and let them choose a thought that felt most comfortable and familiar to them. With mental commands trained, we can now provide the Unity game engine with the users unique training profile to trigger events in the scene depending on the command sent to Unity via the Cortex API.

### 3.3.3 Cortex API

To access training profile and mental commands from the EmotivBCI software, we needed to use the Cortex API from Emotiv to integrate with Unity. An EmotivID and paid License is required create Cortex Apps. Cortex Apps are linked to your EmotivID and its how both you and Emotiv can keep track of projects that have access to your data. Once you have created a new Cortex App, you will be presented with a *Client ID* and *Secret ID* which you will need to provide to the EmotivSDK and approve the third-party application for EEG data to start flowing to Unity.

### 3.4 NeuroGaze Integration

At a high level, the NeuroGaze input device pipeline is as follows:

1. Users create a new training profile and train the *neutral* and *pull* commands.
2. Using an EmotivID, a new Cortex App is created, and a new Client ID and Secret ID are generated.
3. The Client ID, Secret ID, and CortexApp name and unique training profile name are set in the MentalCommands and Config C# scripts.
4. The requestAccess API is called from Unity to Cortex and the Unity application is accepted via the Emotiv Launcher.
5. Once access is granted, the Epoc X connects to the PC running Unity via Bluetooth.
6. The controlDevice API with the “refresh” command is called from Unity to start scanning for headsets every frame.
7. The queryHeadsets API is called to display available Emotiv headsets to connect to.
8. The controlDevice API with “connect” command specifies which headset from the queried list the Unity application wants to connect to.
9. The Authorize API is called to get a Cortex token.

10. The createSession API is called to open a new session for BCI streaming.
11. Call queryProfile API to get list of training profiles associated with Emotiv ID
12. Load the users training profile by calling LoadProfile with the users unique profile name.
13. Subscribe to data stream of Epoc X using SubscribeData.
14. Every frame, call GetMentalCommand from created instance of MentalCommands.
15. Every frame, check if eye gaze is intersecting with an selectable object and mental command is *pull*.
  - a. If both are true, select object.
  - b. If either is false, do nothing.

The NeuroGaze architecture heavily relies on the built in eye tracking feature of the Meta Quest Pro to pull out the position of each eye in World Space to calculate the center position. The Cortex API is also essential to this design as it gives the system a way to access the classified mental command from the user every frame based on the pretrained training profile in the EmotivBCI program.

### 3.4.1 Unity Game Engine

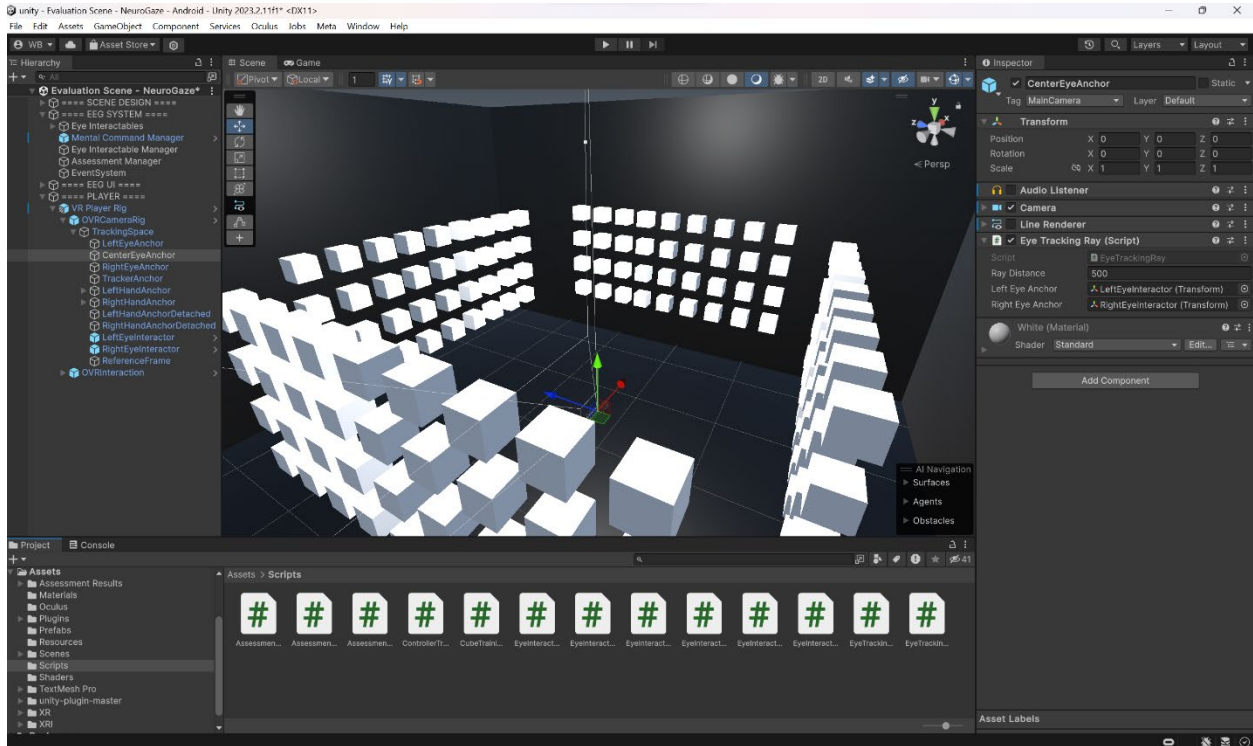


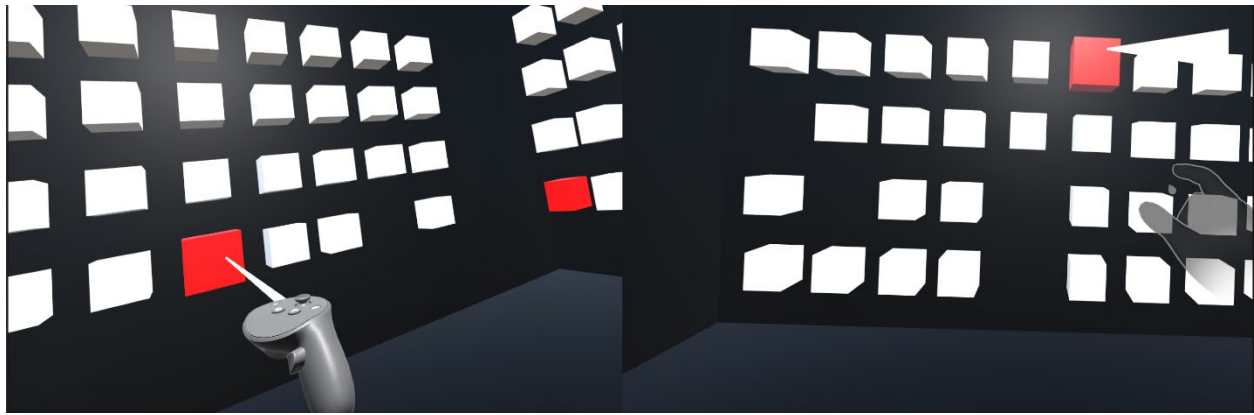
Figure 9: Example Unity Scene - NeuroGaze

The Unity Game Engine played a pivotal role in the development of the NeuroGaze system. Unity provided us with a platform to build our VR application on to test the NeuroGaze system in a VE. Unity's Game Object and Prefab components allowed us to configure structures to handle replicable logic for interaction and assessment management. The integration of NeuroGaze with Unity allowed us to use the Meta XR All in One SDK to create a VR player rig and interactable objects that the user can select while in VR. With the use of C# scripts, we were able to program logic for handling eye tracking, hand gestures, selection, and a gameplay loop.

#### 3.4.1.1 Scenes

Scenes in Unity are where your gameplay content lives. In the NeuroGaze Unity project, scenes were used to separate parts of the training and evaluation. We had five scenes: Training Scene – Neutral

Mental Command, Training Scene – Pull Mental Command, Evaluation – NeuroGaze, Evaluation – Eye Gaze combined with Hand Gesture, Evaluation – VR Controllers (See *Figure 10*) All of these scenes consisted of the black room, lit with point lights, to ensure there were no unnecessary artifacts produced in the EEG data from an overstimulating design. The *neutral* command training scene consist of a single wall with white cubes that the users can hover on using their eye gaze.



*Figure 10: Evaluation Scenes – VR Controllers (left), Eye Gaze combined with Hand Gestures (right)*

The *pull* command training scene is identical in appearance and functionality to the *neutral* command training scene, except it has a single red cube in the middle that acts as a focus point for users to train their brain activity for the *pull* command. The evaluation scene for all three input devices being evaluated are the exact same. The only notable difference in the scene is what represents the player's hand (VR controllers or synthetic hands) and where the ray cast originates from (VR Controller or midpoint between the user's eyes). Unity scenes provided us with a VE in which to test input devices against each other in an intuitive and useful way.

### **3.4.1.2 VR Player Rig**

Using the Meta XR All in One SDK imported into the Unity project, I was able to configure a VR rig that handles tracking and representing the user's eyes, hands and controllers in the VE. The VR rig consists of an *OVRCameraRig* game object that contains the main VR camera and *Tracking Space* game

object. This Tracking Space contains the *LeftEyeInteractor* and *RightEyeInteractor* game objects which represent the position of the user's eyes in the scene (World Space) using the *OVREyeGaze* C# script. The *OVREyeGaze* script works by requesting permission from the Meta Quest Pro HMD for its eye tracking data, then every frame eye tracking data is retrieved and processed using *GetEyeGazeState* and a predefined *ConfidenceThreshold*. The transform of the *LeftEyeInteractor* and *RightEyeInteractor* are changed to match the appropriate position and rotation users should expect their eye to be in the VE.

```
private void Update()
{
    Vector3 eyesCenter = (leftEyeAnchor.position + rightEyeAnchor.position) * 0.5f; // Average the left and right eye anchors to find the midpoint
    Vector3 forwardDirection = (leftEyeAnchor.forward + rightEyeAnchor.forward) * 0.5f; // Average the forward direction of both eyes
    Ray ray = new Ray(eyesCenter, forwardDirection);
    RaycastHit hit;
    bool isHit = Physics.Raycast(ray, out hit, rayDistance);

    // Check the users eye gaze has hit an object
    if (isHit)
    {
        EyeInteractable eyeInteractable = hit.collider.GetComponent<EyeInteractable>();

        // Check if the users eye gaze hit an item we can interact with (select)
        if (eyeInteractable)
        {
            if (lastEyeInteractable != eyeInteractable)
            {
                if (lastEyeInteractable != null)
                {
                    lastEyeInteractable.Hover(false); // If previous interactable object was in a hover state, set this state to false
                }
                lastEyeInteractable = eyeInteractable;
            }
            eyeInteractable.Hover(true); // Enlargen or set this cubes "hover state" to true to let users know that can select it
        }
        else if (lastEyeInteractable != null)
        {
            lastEyeInteractable.Hover(false); // If previous interactable object was in a hover state, set this state to false
            lastEyeInteractable = null;
        }
    }
    else
    {
        if (lastEyeInteractable != null)
        {
            lastEyeInteractable.Hover(false); // If previous interactable object was in a hover state, set this state to false
            lastEyeInteractable = null;
        }
    }

    // Update the LineRenderer to represent the ray deriving from the users eye
    lineRenderer.SetPosition(0, eyesCenter);
    lineRenderer.SetPosition(1, eyesCenter + forwardDirection * rayDistance);
}
```

Figure 11: *EyeTrackingRay* C# script source code for *NeuroGaze* Input Device

Using these eye interactors, I have written my own script called *EyeTrackingRay* (See Figure 11) that has its logic update every frame (updates at approximately 60 frames per second). This script calculates the midpoint between two eye interactors and the forward vector. Using this position vector, a



RayCast is projected from the midpoint of the user's eyes and a check is performed to see what the object the user looked at was. If it was in object the user can interact with, an *EyeInteractable*, slowly grows to a fixed scale to signify to users that it is in a "hover state." If the user makes eye contact with another *EyeInteractable* or non-*EyeInteractable*, the previous object the user looked it with return to its original scale and it will be out of its "hover state." A Line Renderer component is also updated every frame from the user's eyes to 500 meters in the forward vector direction to represent where the user is looking. The *EyeTrackingRay* script is used to trigger the hover state on *EyeInteractable* in the NeuroGaze and Eye Gaze combined with Hand Tracking input methods.

### 3.4.1.2 Eye Interactable

The Eye Interactable game objects are represented by cubes in the Unity scene. They are the only objects users can select in the scene. Each Eye Interactable game object in the scene has the *EyeInteractable* script (See *Figure 12*) attached to it which interfaces with the *EyeTrackingRay* and *MentalCommandsManager* scripts. These objects have two states: a hover state set to true, and a hover state set to false. When the objects state is set to true, its scale will slowly grow to a max of 1.28x the objects original scale (*minimum scale:  $0.18m^3$ , maximum scale:  $0.2304m^3$* ). Similarly, if the hover state is false, the object will slowly shrink from its current scale to its original scale. When an Eye Interactor's hover state is set to true and the mental command received from is *pull*, further interaction with this object is disabled and the object is shrunk and destroyed from the scene. The Eye Interactable object knows what mental command is currently being classified from the EEG data through a reference to the *MentalCommandsManager* Instance which has a public method *CurrentMentalCommand* that scripts in our Unity project can access. During the assessment of each input device, 12 of these Eye Interactables will turn red and users must select as them as quickly and accurately as possible.

```

private void Update()
{
    string mentalCommand = EyeInteractableManager.Instance.CurrentMentalCommand; // Reference to EyeInteractableManager Instance

    // Check hover state and current mental command
    if (isHovered && mentalCommand == "pull")
    {
        // Set flags to select and destroy cube, stop cube from being interacted with
        shouldShrink = true;
        isBeingPulled = true;
    }
    else
    {
        if (isBeingPulled)
        {
            transform.localScale = originalScale;
            isBeingPulled = false;
        }
    }

    // Check mental command and hover state triggers
    if (shouldShrink)
    {
        ShrinkAndDestroy();
    }
    else if (isHovered)
    {
        ScaleUp(); // Change interactable scale to show user they are looking at it
    }
    else
    {
        ScaleDown(); // Change interactable scale to show user they are not looking at it
    }
}

```

*Figure 12: EyeInteractable C# source code for NeuroGaze Input Device*

### 3.4.1.2 Mental Commands Manager

```
public class MentalCommands : MonoBehaviour
{
    EmotivUnityItf _emotivUnityItf = new EmotivUnityItf();
    bool mentalCmdRcvd = false;
    string mentalCommand;
    List<string> dataStreamList = new List<string>() { DataStreamName.MentalCommands, DataStreamName.SysEvents };
    string clientId = "1";
    string clientSecret = "1";
    string appName = "MSI3";
    public string profileName = "wanyeaPilot";
    public string headsetId = "EPOCHX-E50208B2";

    // Delegate for mental command changes
    public delegate void OnMentalCommandChanged(string newCommand);
    public event OnMentalCommandChanged MentalCommandChanged;

    @ Unity Message | 0 references | Wanyea Barbel, 13 days ago | 1 author, 1 change
    void Start()
    {
        _emotivUnityItf.Init(clientId, clientSecret, appName);
        _emotivUnityItf.Start();
        DataStreamManager.Instance.ScanHeadsets();
    }

    @ Unity Message | 0 references | Wanyea Barbel, 13 days ago | 1 author, 1 change
    private void OnGUI()
    {
        if (Event.current.Equals(Event.KeyboardEvent("1")))
        {
            try
            {
                _emotivUnityItf.CreateSessionWithHeadset(headsetId);
                _emotivUnityItf.LoadProfile(profileName);
                _emotivUnityItf.SubscribeData(dataStreamList);
                mentalCmdRcvd = true;
            } catch {
                UnityEngine.Debug.Log("Creating Session with: " + headsetId);
            }
        }
    }
}
```

Figure 13: Mental Commands C# script - Connecting to headset and subscribing to data stream

The core component to accessing the training profile and EEG data collected in the EmotivBCI program is the Mental Commands Manager. This game object is configured with the *MentalCommands* script (See Figure 13) which utilizes the Cortex API to retrieve EEG data from the Emotiv EPOCH X (See Figure 14). *MentalCommands* is configured with the App Name, Client ID and Client Secret generated when you create a new Cortex App. Additionally, the unique name of the training profile created in the EmotivBCI program, and the headset ID found on the EPOCH X are configured in this script. The NeuroGaze system set up so a researcher can start the Unity scene and click “1” on the keyboard to handle all of the Emotiv logic and start reading the EEG stream from the headset, On Start, a new instance of *EmotivUnityItf* is initialized with the user credentials. If the user’s profile has a valid Emotiv license and the Unity application has been approved to access Emotiv data via the Emotiv Launcher, the program will begin

scanning for Emotiv headsets. Then, if the *headsetId* matches a headset found during the scan, a connection to the headset will be established with the *CreateSessionWithHeadset* method. Then, the *LoadProfile* method will be called with the *profileName*. If a training profile associated with the *clientId* and *clientSecret* is found a EEG data stream will start and mental commands can be extracted from the EmotivBCI program. Accessing these mental commands is achieved by calling the *mentalCmdIs* method from the instance of *EmotivUnityIft* and a string representing this mental command will be returned (See Figure 14).

```
void Update()
{
    if (mentalCmdRcvd)
    {
        string currentCommand = _emotivUnityIft.mentalCmdIs();

        if (currentCommand != lastMentalCommand)
        {
            Debug.Log($"The current mental command is: {mentalCommand}");
            mentalCommand = currentCommand;
            lastMentalCommand = currentCommand;

            // Trigger the event
            MentalCommandChanged?.Invoke(mentalCommand);
        }
    }
}

1 reference | Wanyea Barbel, 13 days ago | 1 author, 1 change
public string GetMentalCommand()
{
    return mentalCommand;
}
```

Figure 14: Mental Commands C# script – Fetching current mental command via Cortex API

The Mental Commands Manager is referenced by each Eye Interactable every frame to check what the users current mental command is. By creating one instance of this object, we can send data about the EPOC X to any part of our Unity program without the need to configure an EEG stream per component.

### 3.4.1.3 Interactable Manager

The Interactable Manager checks the state of each Eye Interactable in the scene and reports their status to the Assessment Manager. Additionally, this component is responsible for the assessment handoff to the Assessment Manager. Each Eye Interactable is a child of one of 4 parent game objects: *leftWall*, *rightWall*, *frontWall* or *backWall*. When the Interactable Manager starts the assessment, 4 of the child game objects on each wall will randomly turn red (See *Figure 15*).

```
IEnumerator ChangeRandomCubesColor(GameObject parentObject)
{
    if (parentObject == null)
    {
        Debug.LogError("Parent object is not assigned!");
        yield break;
    }

    // Get all child objects from parent
    Transform[] children = parentObject.GetComponentsInChildren<Transform>();

    // Shuffle the array
    System.Random rng = new System.Random();
    int n = children.Length;
    while (n > 1)
    {
        n--;
        int k = rng.Next(n + 1);
        Transform value = children[k];
        children[k] = children[n];
        children[n] = value;
    }

    // Change the color of 3 cubes per wall to red
    int cubesColored = 0;
    foreach (Transform child in children)
    {
        if (child != parentObject.transform && cubesColored < targetsPerWall)
        {
            Renderer renderer = child.GetComponent<Renderer>();
            if (renderer != null)
            {
                renderer.material.color = Color.red;
                child.gameObject.tag = "RedCube"; // Tag the cube as RedCube
                cubesColored++;
            }
        }
    }
}
```

Figure 15: Interactable Manager C# script – Changing cube colors when assessment has started

At this moment, the Assessment Manager will begin monitoring the Eye Interactables for how many red cubes have been deleted, how many white cubes have been deleted and how many complete misses have been made.

#### **3.4.1.4 Assessment Manager**

The Assessment Manager oversees all the Eye Interactables when the Unity scene is in assessment mode. During this mode, 12 Eye Interactables will turn red as a cue to inform which objects, they should be selecting. The Assessment Manager keeps track of how many times the user attempted to select an object and missed, how many times they selected the incorrect object and how long it took for the user to successfully select all the correct objects in the scene. These values are stored until the assessment is over and then they are exported as a CSV. A unique participant id is set by the evaluator and this id is included in the exported CSV file. This logic is applied to all three select devices to give evaluators an easy and intuitive way to run the same assessment on any input device.

## CHAPTER FOUR: USER STUDY

### 4.1 Experimental Design

#### 4.1.1 Procedure

The study is a 1 (task complexity; within subject) x 3 (available 3D user interfaces; between subject) mixed design study in which participants will be evaluated on all three 3DUI techniques in virtual reality (NeuroGaze: eye gaze for highlighting, EEG for intent to interact with object. VR controller: ray from controller to highlight, right trigger button to select. Eye Gaze combined with Hand Tracking: eye gaze for highlighting, pinch gesture to interact with object). Users will have to select all the red cubes out of the white ones in the scene where task performance and accuracy will be evaluated by determining the time taken to select all the red cubes and how many were selected correctly. The studies took place on the UCF main campus in the same room with the Principal Investigator (PI) is consisted of the following procedure:

1. The participant will first complete a demographics survey recording age, gender, familiarity to using VR systems, familiarity playing video games, listing any video games they play, listing any VR video games they play and if the participant is allergic to any of ingredients found in our saline solution.
2. We will provide an overview of the electroencephalogram (EEG) headset the participant will be wearing (the Emotiv EPOC X) and the VR headset (the Meta Quest Pro) and demonstrate to participants how to put both headsets on.
3. The participant will put the EEG headset on, and we will assist moving the nodes around to ensure the best contact quality and comfort for participants. Saline solution will be applied to each node on the EEG headset to help establish a higher contact quality between the EEG headset

and the participant's scalp. Some participants may be asked to put their hair in a higher position to accommodate for EEG node placement.

- a. At this time, the EmotivBCI software used to collect user EEG data should be running and connected to the Emotiv EPOC X EEG headset.
  - b. The EmotivBCI program displays the contact quality of each node on the participants head and this is the time to make sure every node has a high contact quality.
  - c. More saline solution may be applied to a node on the EEG if the EmotivBCI program is showing that the contact quality of that node is low.
4. A new-unused silk headband will be put around the participants head to keep the EEG headset in place and provide more comfort to the participant.
5. We will put the VR headset on the participants head and help the participant adjust it for their comfort.
6. We will calibrate the eye trackers in the VR headset by asking the participant to look at a target while it is moving in VR.
7. We will load the training game in VR and explain what participant will need to do.
  - a. At this time, the EmotivBCI software should be ready to begin collecting data.
8. The participant will perform tasks in VR within the training game that we will use to gather their EEG data.
  - a. The training game within the VE consists of a black room with a singular red cube in the middle of the room.
  - b. Participants are told to look around the room over 8 seconds intervals. These 8 second intervals are repeated 10 times (80 seconds in total).



- c. Over these 8 second intervals, the PI will start the EEG data collection within the EmotivBCI program for the neutral state. This is the baseline state for the participants EEG data.
  - d. Participants are then told to look at the red cube in the black room and over 8 second intervals, attempt to shrink the cube by imagining themselves shrinking it. This 8 second interval is repeated 10 times (80 seconds in total).
  - e. During these 8 second intervals, the PI will start the EEG data collection within the EmotivBCI program for the “shrink” state. Additionally, the PI will force the red cube to shrink over this 8 second interval to give the illusion that the participant is shrinking the cube for the EEG training data to be as accurate as possible. This is the data used to classify when the participant wants to shrink an object.
9. Once the training sessions are over, we will load the evaluation game in VR and participants will be told what to do.
- a. The evaluation game puts participants in the same black box within the VE as they were in during the training game. Now, instead of one red cube, each wall has a 4x9 array of white cubes (36 per wall, 144 in total).
  - b. When participants look at the cubes, they will grow to a set size to let the participant know they are interacting with the cube.
  - c. When the participant is ready, the PI will start the evaluation game and start the link between the evaluation game and the EmotivBCI program.
  - d. Once the evaluation game has started, one cube on each wall will be randomly chosen to turn red (4 red cubes in total).
  - e. Participants will need to look at these red cubes and imagine they are shrinking it. Once these conditions have been met, the red cube in the game will disappear.

- f. Participants are tasked with making all the red cubes in the game disappear as quickly and accurately as they can to complete the evaluation game.
  - g. This evaluation game is run for three rounds.
- 10. We will remove the EEG headset, the silk headband, and VR headset from the participant's head.
- 11. Participants will complete one Qualtrics NASA-TLX survey on a laptop that gives feedback on the NeuroGaze interaction technique. After completion of this survey, this will conclude the NeuroGaze evaluation section of the experiment.
- 12. We will put the VR headset back on the participant and help adjust the headset for their comfort. We will load the evaluation game in VR and the participant will perform a task using their eyes and hands.
  - a. Like the NeuroGaze evaluation, participants are placed in the same black box with the same 4x9 array of cubes that change size when the users look at them.
  - b. When the PI starts the assessment, 4 cubes will randomly turn red and participants must as quickly and accurately as possible, look at a red cube and perform a "pinch gesture" with their hands to make all the red cubes disappear.
  - c. This evaluation game is run for three rounds.
- 13. We will remove the VR headset from the participants head and participants will complete one Qualtrics NASA-TLX survey on a laptop that gives feedback on the Eye Tracking combined with Hand Tracking interaction technique. After completion of this survey, this concludes the Eye Tracking combined with Hand Tracking evaluation section of the experiment.
- 14. We will put the VR headset back on the participant and help adjust the headset for their comfort. We will put the participant in the evaluation game in VR and the participant will perform a task using VR controllers.
  - a. Participants are placed in the same black box with the same 4x9 array of cubes.

- b. These cubes will change size when the user points their controller at them.
  - c. When the PI starts the assessment, 4 cubes will randomly turn red and participants must as quickly and accurately as possible, point their controller at a red cube and click the trigger to make all the red cubes disappear.
  - d. This evaluation game is run for three rounds.
15. We will remove the VR headset from the participants head and gather the VR controllers from the participant. Participants will complete one Qualtrics NASA-TLX survey on a laptop that gives feedback on the VR controller interaction technique. After completion of this survey, this concludes the VR controller evaluation section of the experiment.
  16. The VR headset will be removed, and we will gather the VR controllers.
  17. Participants will complete one Post Evaluation survey giving their feedback about the entire experiment and all three interaction techniques evaluated.
  18. Participants will be paid \$15 in cash and are free to leave.
  19. After participants leave, the EEG headset will be sanitized with saline on each of the nodes, the VR headset will be cleaned with sanitized wipes, and the headband will be thrown away.

## **4.2 Results**

Each participant went through the same procedure to remove any unnecessary bias from the study. The was conducted with 25 participants aged from 18 to 35 with 10 participants who identified as female and 15 participants who identified as male. Participants were asked to complete a demographics survey that detailed their age, gender identity and familiarity with VR, AR, and video games. The results can be found in Table 1. This table lists the answers question: “Please rate your level of expertise with each of the following (1 means no experience, 5 means very experienced).” The number in each row represents how many participants marked the answer.

*Table 1: Demographics Survey Results*

	1	2	3	4	5
Virtual Reality	4	3	8	6	4
Augmented Reality	8	6	6	3	2
Video Games	2	0	4	7	12

As stated in the procedure, participants were tasked with selecting all the red cubes in the scene as efficiently as possible. Missing a selectable object or selecting the wrong object amassed to the same error count and this was recorded. The average duration was calculated by the Assessment Manager and the value represents how long each user took to select every red cube in the scene. The results of how each selection technique did after this evaluation can be found in Table 2.

*Table 2: Evaluation results for each selection technique*

Selection Technique	Average Durations (s)	Average Error Rate
NeuroGaze	39.26	0.135
Eye Gaze combined with Hand Gesture	15.44	0.172
VR Controllers	10.78	0.122

After each selection method, participants were asked to fill out a NASA-TLX to evaluate cognitive load and a Post Evaluation survey.

### 4.3 Discussion

The evaluation of NeuroGaze against traditional VR input devices and eye gaze combined with hand gesture reveals significant insights into user interaction within virtual environments (VEs).

NeuroGaze's higher average duration indicates a learning curve and adjustment period to the novel interaction method, reflecting its innovative integration of EEG and eye tracking for object selection.

However, the lower error rate compared to eye gaze with hand gesture highlights NeuroGaze's potential

for precise selection once users acclimate. VR controllers, with the shortest average duration and lowest error rate, remain the most familiar and intuitive for users, likely due to their widespread use and physical feedback. However, the physical exertion and potential accessibility issues for users with mobility impairments underscore the importance of developing alternative interfaces like NeuroGaze. The cognitive load assessed through NASA-TLX scores emphasizes the balance between ease of use and the cognitive effort required to master new technologies. As NeuroGaze matures, optimizing its interface to reduce cognitive load will be key to enhancing user experience and acceptance.

## CHAPTER FIVE: FUTURE WORK

This Thesis provided our research team with a proof of concept to build upon in future interactions. The NeuroGaze system has shown that it is possible to accurately select objects in a VE with eye gaze and EEG data using motor imagery. This was a learning experience and as such we will be taking the lessons learned to iterate on the NueroGaze design. We plan to try new EEG headsets as the EPOC X is a commercial EEG headset and a medical grade headset would give us access to more channels and higher quality EEG data for classification. Additionally, users expressed general discomfort when wearing the NeuroGaze system for a prolonged period. To mitigate this in the future, we would try an EEG cap which might reduce the amount of pressure put on the front of the user's head since EEG caps usually have more evenly distributed channels. It might be interesting to develop our own EEG headset using the OpenBCI electrode nodes and firmware. This would allow us to strategically choose which channels we would want to access and design the headset in a way that's more compatible with the Meta Quest Pro HMD. If funding allows, we would like to research the use of functional near-infrared spectroscopy (fNIRS) in place of EEG. This is because studies suggest that EEG is better for classification of emotional state but fNIRS has better performance as classification motor imagery. We are also interested in potentially spending the time to train our own machine learning denoising and classification machine learning algorithm. The ability to personalize this model to our needs would allow us classify specific artifacts we want to filter out such as repeated kinematic movement. This would remove physical restrictions we had to put in place for the NueroGaze system and would allow users to move around in the Unity scene without the fear of unnecessary noise in the EEG data. Evaluation of a users ability to track moving objects with eye gaze and select using this version of NeuroGaze could also bring about some interesting research into this systems ability utilize the natural eye movement of *saccade* to *smooth pursuit*. Overall, we are very interested in continuing to research the applications of BCI interfacing with VEs and 3DUIs.



## **CHAPTER SIX: CONCLUSION**

This thesis presented NeuroGaze, a novel VR interaction system that combines eye tracking with EEG data to facilitate object selection in VEs. The system's design, development, and evaluation underscore the potential of integrating biometric inputs to create more intuitive and accessible user interfaces.

Despite facing challenges such as user adaptation and cognitive load, NeuroGaze demonstrates a promising direction for future VR interaction technologies. By pushing the boundaries of traditional VR input devices, NeuroGaze not only offers an alternative for users with physical disabilities but also enriches the interaction paradigm within VEs. As VR continues to evolve, technologies like NeuroGaze will play a pivotal role in shaping immersive experiences that are both inclusive and engaging.

NeuroGaze represents a significant step forward in the quest for seamless human-computer interaction within virtual worlds. Its development and study contribute valuable insights to the field of VR research, laying the groundwork for future innovations that will further dissolve the barriers between the human mind and interactions in virtual environments.



## APPENDIX A: SAMPLE SURVEYS

### A.1 Qualtrics Demographic Survey

#### NeuroGaze VR Demographics Survey

Welcome! This is the Demographics Survey for the Study on Assessing EEG and Eye Tracking Interfaces Against Conventional VR Selection Methods. Click the arrow to begin.

Q1 Participant Number

---

Q2 Age

---

Q3 Gender Identity

---

Q4 Please rate your level of expertise with each of the following (1 means no experience, 5 means very experienced)

	1 (1)	2 (2)	3 (3)	4 (4)	5 (5)
Virtual Reality (1)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Augmented Reality (2)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Video Games (3)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Q5 List any virtual reality games you have played:

---

---

---

---

---

---

Q6 Are you allergic to any of the following ingredients found in the saline solution () we will be using?

Ingredients: Sodium Citrate, Sodium Chloride, Boric Acid, Sorbitol, Aminomethyl Propanol, Disodium EDTA, Two Wetting Agents (Tetronic 1304 and HydraGlyde Moisture Matrix [EOBO-41- polyoxyethylenepolyoxybutylene]) with Polyquad (Polyquaternium-1) 0.001% and Aldox (Myristamidopropyl Dimethylamine) 0.0006% Preservatives. HydraGlyde Moisture Matrix is a Proprietary Multi-functional Block Copolymer that is Primarily Designed for Wetting and Lubricating Silicone Hydrogel Lenses.

☐ Yes (1)

☐ No (2)

---

Q7 Do you have any previous history of simulator sickness?

☐ Yes (1)

☐ No (2)

End of Block: Default Question Block

---

## A.2 Qualtrics NASA-TLX Survey

### NeuroGaze VR NASA TLX

---

Start of Block: Default Question Block

Welcome! This is the NASA TLX Survey for the Study on Assessing EEG and Eye Tracking Interfaces Against Conventional VR Selection Methods. Click the arrow to begin.

---

Participant Number

---

Technique being Evaluated

- ☐ NeuroGaze: EEG + Eye Tracking
  - ☐ Eye Tracking + Hand Tracking
  - ☐ VR Controllers
-

Please answer each prompt based on your experience:

	1 (Very Low)	2	3	4	5	6	7 (Very High)
How mentally demanding was the task?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
How physically demanding was the task?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
How hurried or rushed was the pace of the task?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
How successful were you in accomplishing what you were asked to do?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
How hard did you have to work to accomplish your level of performance?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
How insecure, discouraged, irritated, stressed, and annoyed were you?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

End of Block: Default Question Block

A.3 Qualtrics Post Evaluation Survey

NeuroGaze Post Evaluation Survey

Start of Block: Default Question Block

Welcome! This is the Post Evaluation Survey for the Study on Assessing EEG and Eye Tracking Interfaces Against Conventional VR Selection Methods. Click the arrow to begin.

Participant Number

Did you feel like you were making the cube shrink during the training scene?  
(Include any additional comments under your answer choice)<

- ☐ Yes
- ☐ No

How intuitive was each technique to to use? (Include any additional comments the under technique)

	1 (least intuitive)	2	3	4	5 (most intuitive)
NeuroGaze (EEG combined with Eye Tracking)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eye Tracking combined with Hand Tracking	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
VR Controllers	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

---

What would you change about any of the interaction techniques or the experiment? If nothing, answer "Nothing"

---

---

Which interaction technique did you prefer overall?  
(Include any additional comments under your answer choice)

☐ NeuroGaze (EEG combined with Eye Tracking)

---

☐ Eye Tracking combined with Hand Tracking

---

☐ VR Controllers 

---

End of Block: Default Question Block

---

## LIST OF REFERENCES

- [1] Agarwal, M., & Sivakumar, R. (2019). Blink: A Fully Automated Unsupervised Algorithm for Eye-Blink Detection in EEG Signals. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (pp. 1113-1121). Monticello, IL, USA: IEEE.  
<https://doi.org/10.1109/ALLERTON.2019.8919795>
- [2] Argelaguet, F., & Andujar, C. (2013). A Survey of 3D Object Selection Techniques for Virtual Environments. *Computers & Graphics*, 37(3), 121-136. <https://doi.org/10.1016/j.cag.2012.12.003>
- [3] Baceviciute, S., Lucas, G., Terkildsen, T., & Makransky, G. (2022). Investigating the redundancy principle in immersive virtual reality environments: An eye-tracking and EEG study. *Journal of Computer Assisted Learning*, 38(1), 120-136. <https://doi.org/10.1111/jcal.12595>
- [4] Blattgerste, J., Renner, P., & Pfeiffer, T. (2018). Advantages of eye-gaze over head-gaze-based selection in virtual and augmented reality under varying field of views. In *Proceedings of the Workshop on Communication by Gaze Interaction*. <https://doi.org/10.1145/3206343.3206349>
- [5] Decety, J. (1996). The neurophysiological basis of motor imagery. *Behavioural Brain Research*, 77(1-2), 45-52. [https://doi.org/10.1016/0166-4328\(95\)00225-1](https://doi.org/10.1016/0166-4328(95)00225-1)
- [6] Fakhruzzaman, M. N., Riksakomara, E., & Suryotrisongko, H. (2015). EEG Wave Identification in Human Brain with Emotiv EPOC for Motor Imagery. *Procedia Computer Science*, 72, 269–276. <https://doi.org/10.1016/j.procs.2015.12.140>
- [7] Fahmi, F., Tanjung, K., Nainggolan, F., Siregar, B., Mubarakah, N., & Zarlis, M. (2020). Comparison study of user experience between virtual reality controllers leap motion controllers and senso glove for anatomy learning systems in a virtual reality environment. *IOP Conference Series:*

- Materials Science and Engineering*, 851(1), 012024. <https://doi.org/10.1088/1757-899X/851/1/012024>
- [8] Hosp, B., Eivazi, S., Maurer, M., Fuhl, W., Geisler, D., & Kasneci, E. (2020). RemoteEye: An open-source high-speed remote eye tracker. *Behavior Research Methods*, 52, 1387–1401. <https://doi.org/10.3758/s13428-019-01305-2>
- [9] Imaoka, Y., Flury, A., & de Bruin, E. D. (2020). Assessing Saccadic Eye Movements With Head-Mounted Display Virtual Reality Technology. *Frontiers in Psychiatry*, 11, 572938. <https://doi.org/10.3389/fpsy.2020.572938>
- [10] Jurcak, V., Tsuzuki, D., & Dan, I. (2007). 10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems. *NeuroImage*, 34(4), 1600–1611. <https://doi.org/10.1016/j.neuroimage.2006.09.024>
- [11] Khan, G. M. (2015). A new electrode placement method for obtaining 12-lead ECGs. *Open Heart*, 2(1), e000226. <https://doi.org/10.1136/openhrt-2014-000226>
- [12] Khazi, M., Kumar, A., & J, V. M. (2012). Analysis of EEG using 10:20 electrode system. *International Journal of Innovative Research in Science, Engineering and Technology*, 1(2).
- [13] Khundam, C., Vorachart, V., Preeyawongsakul, P., Hosap, W., & Noël, F. (2021). A Comparative Study of Interaction Time and Usability of Using Controllers and Hand Tracking in Virtual Reality Training. *Informatics*, 8(3), 60. <https://doi.org/10.3390/informatics8030060>
- [14] LaViola Jr., J. J., Kruijff, E., Bowman, D. A., McMahan, R. P., & Poupyrev, I. (2017). *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional.
- [15] Larsen, O. F. P., Tresselt, W. G., Lorenz, E. A., Holt, T., Sandstrak, G., Hansen, T. I., Su, X., & Holt, A. (2024). A method for synchronized use of EEG and eye tracking in fully immersive VR. *Frontiers in Human Neuroscience*, 18, 1347974. <https://doi.org/10.3389/fnhum.2024.1347974>



- [16] Lin, W., Du, L., Harris-Adamson, C., Barr, A., & Rempel, D. (2017). Design of hand gestures for manipulating objects in virtual reality. In M. Kurosu (Ed.), *Human-Computer Interaction. Theories, Methods, and Human Issues: 19th International Conference, HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings, Part I* (pp. 584–592). Springer International Publishing. [https://doi.org/10.1007/978-3-319-58071-5\\_44](https://doi.org/10.1007/978-3-319-58071-5_44)
- [17] Lopez Luro, F., & Sundstedt, V. (2019). A comparative study of eye tracking and hand controller for aiming tasks in virtual reality. In *ETRA '19: Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (Article No. 68, pp. 1–9). ACM. <https://doi.org/10.1145/3317956.3318153>
- [18] Luong, T., Cheng, Y. F., Mobus, M., Fender, A., & Holz, C. (2023). Controllers or Bare Hands? A Controlled Evaluation of Input Techniques on Interaction Performance and Exertion in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics*, 29(11), 4633-4642.
- [19] Perret, J., & Vander Poorten, E. B. (2018). Touching Virtual Reality: A Review of Haptic Gloves. Conference Paper, June 2018. Haption GmbH, Aachen, Germany; Department of Mechanical Engineering, KU Leuven, Belgium. Retrieved from <https://www.researchgate.net/publication/324562855>
- [20] Piumsomboon, T., Lee, G. A., Lindeman, R. W., & Billinghamurst, M. (2017). Exploring natural eye-gaze-based interaction for immersive virtual reality. In *Proceedings of the 2017 IEEE Symposium on 3D User Interfaces (3DUI)* (pp. 1-10). IEEE. <https://doi.org/10.1109/3DUI.2017.7893315>
- [21] Plöchl, M., Ossandón, J. P., & König, P. (2012). Combining EEG and eye tracking: Identification, characterization, and correction of eye movement artifacts in electroencephalographic data. *Frontiers in Human Neuroscience*, 6, Article 278. <https://doi.org/10.3389/fnhum.2012.00278>

- [22] Putze, F., Popp, J., Hild, J., Beyerer, J., & Schultz, T. (2016). Intervention-Free Selection using EEG and Eye Tracking. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction (ICMI '16)*, Tokyo, Japan. ACM. <https://doi.org/10.1145/2993148.2993199>
- [23] Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3), 372–422. <https://doi.org/10.1037/0033-2909.124.3.372>
- [23] Whitmire, E. (2018). High-Fidelity Interaction for Virtual and Augmented Reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces* (pp. 796-798). Reutlingen, Germany: IEEE. <https://doi.org/10.1109/VR.2018.8446520>
- [24] Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., & Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6), 767–791. [https://doi.org/10.1016/S1388-2457\(02\)00057-3](https://doi.org/10.1016/S1388-2457(02)00057-3)
- [25] Zander, T. O., Gärtner, M., Kothe, C., & Vilimek, R. (2011). Combining Eye Gaze Input With a Brain–Computer Interface for Touchless Human–Computer Interaction. *International Journal of Human-Computer Interaction*, 27(1), 38-51. <https://doi.org/10.1080/10447318.2011.535752>