Improving Assistive Technologies Using EEG Headsets

David Ivaşcu[a] and Isabela Drămnesc[b]

Department of Computer Science, West University of Timişoara, Romania

Keywords: EEG Headset, P300, Unity, Whisper, Speech-to-Text.

Abstract: Brain computer interfaces (BCI) have gained increasing attention in recent years due to the improved affordability and usability of electroencephalogram headsets (EEG). These headsets paired with the right software make computers usable without a physical input such as the traditional mouse and keyboard, creating new opportunities for users with motor impairments. In this paper, we present the design and development of an assistive application that employs an EEG headset (Unicorn Hybrid Black) as the main control interface for user interaction. The system integrates a launcher style interface that contains multiple accessible functions, allowing users to interact with software environments exclusively through EEG–based commands. This work aims to advance digital accessibility and promote independence for people who cannot rely on conventional input devices. By outlining a practical approach for integrating EEG headsets into everyday computer use, this paper contributes to the ongoing development of assistive technologies.

1 INTRODUCTION

Electroencephalography (EEG) is a method used to record the electrical activity of the brain using electrodes placed on the scalp. This method has been used for decades in the medical field to detect certain brain conditions by analyzing the electrical activity graph. But this technology is not relevant only in the medical field, the brainwave data can be used for brain computer interfaces (BCI), essentially using the headsets an input method for computer based systems such as: robotic limbs, smart home systems, graphical user interfaces. BCI's are often used for assistive systems meant for people with motor disabilities by replacing the traditional mouse and keyboard input with the EEG headset input, thus removing the need for physical interaction with the computer.

This technology still has not been widely adopted for assistive systems due to its experimental nature and sometimes complex setup processes. More vigorous testing and optimizations should be done on these headsets and the algorithms running on them to ensure that we can get the best performance and accuracy out of the existing hardware.

The aim of this paper is to highlight the potential of EEG based BCI systems in improving assistive technologies and to experiment with the existing hardware and EEG headsets by creating a Unity interface with multiple features which is controlled using the g.tec Unicorn Hybrid Black EEG headset.

[a] https://orcid.org/0009-0003-8777-7989
[b] https://orcid.org/0000-0003-4686-2864

2 RELATED WORK

EEG Powered Robotic Devices for Mobility Aid. Recent studies have demonstrated that EEG headset can enable BCI for controlling robotic systems, ranging from wheelchairs to exoskeletons. (Swee, Sim Kok et al., 2016) describe in detail the process of building an EEG wheelchair. For the EEG headset, Emotiv EPOC was chosen due to its good signal capturing capabilities and the SDK that handles data acquisition and processing. After the wheelchair was fully built, it was tested and the results presented by the authors were favorable with good accuracy. On the other hand, there were no safety features implemented.

(Tang et al., 2018) present a more advanced EEG controlled wheelchair with multiple safety features included. It is equipped with multiple sensors to ensure a reliable way of mapping the environment. The wheelchair uses a you only look once algorithm to select a target, which is the destination, then through a P300 paradigm BCI, the user can confirm the destination. The authors have conducted tests on this system by simulating real life scenarios. The tests concluded

Ivaşcu, D. and Drămnesc, I.
Improving Assistive Technologies Using EEG Headsets.
DOI: 10.5220/0013788400003982

In *Proceedings of the 22nd International Conference on Informatics in Control, Automation and Robotics (ICINCO 2025) - Volume 1*, pages 362-368
ISBN: 978-989-758-770-2; ISSN: 2184-2809

that the smart wheelchair works as intended and EEG powered BCIs have potential if implemented properly.

Because safety is a serious matter when talking about autonomous mobility devices, it is important to implement robust safety layers to ensure that the system does not have many points of failure. This topic was addressed in (Tariq et al., 2018), where the authors have reviewed many related work and concluded that, even if EEG based BCI controlled robotic systems can achieve good results in the accuracy department, the safety features still lack development and the systems are not yet deployable without supervision. The general idea behind EEG controlled mobility aid robotic devices is that they present potential with proper software implementation, but need thorough testing and development for better safety protocols in order to be deployed autonomously.

EEG Controlled Robotic Hands. Robotic hands were created to research the possibility of restoring movement or studying the possibility of motor cortex EEG data acquisition to restore hand movement. (Kline and Desai, 2014) present the process of EEG data acquisition with the purpose to be used on a robotic hand. When collecting the data with the headset, the participants were shown a series of left and right arrow, having to raise the hand that matches the arrow direction. After acquisition was realized, the data was filtered and processed, then tested on a robotic hand. A similar approach was used in (Kasim et al., 2017), where the authors describe the developing process of a real time EEG controlled hand. The hand has two functions: open and closed. Each state was mapped to a specific action. If the user of the headset looks to the right, the hand opens and if the user smiles the hand closes. Even if the complexity of the robotic hand movement is limited, it is possible to map certain brain signals to actions.

Robotic hands have also been tested as assistive devices for individuals recovering from stroke related impairments. In these cases, damage to certain brain areas may limit the reliability of the signals captured by the EEG headset. (Fok et al., 2011) propose a specialized algorithm designed to identify alternative regions from the brain that display motor activity. By using the headset with the algorithm, they successfully enabled control of a robotic hand orthosis.

One of the limitations presented in all the cited works is the sensitivity of the EEG capturing device to artifacts. Physical movement of headset user can greatly affect the quality of the signal, as well as electromagnetic interference from other devices. Despite these limitations, EEG controlled robotic hands show potential for the future.

EEG Controlled Applications. A wide range of EEG based applications have been developed to demonstrate the potential of BCIs in real world scenarios. A number of studies have experimented with EEG controlled applications on different platforms. These applications are usually designed to help individuals with physical impairments by enabling them to interact with digital devices without the need of traditional input methods such as keyboard and mice. For example, (Rușanu et al., 2020) developed an EEG controlled chat application. The application connects a laptop instance and a smartphone instance. The laptop instance is meant to be used by the headset wearer and it features prefabricated messages that can be sent to the smartphone by selecting it using blinks. From the smartphone, messages are sent normally using a keyboard.

(Mugler et al., 2010) have created a web browser that is controllable using EEG BCI. The browser uses the P300 paradigm, so the user has to focus of flashing object in order to control the browser or spell words. The user interface was created to be minimal and intuitive, in order to facilitate the use with an EEG headset. The authors have tested the accuracy of the EEG headset on healthy volunteers as well as individuals with amyotrophic lateral screlosis (ALS). The recorded accuracy was about 90 percent for healthy individuals and 73 percent for the ALS patients.

A more advanced application presented in (He et al., 2017) integrates a speller, a web browser, an e-mail client and a file explorer. The application is controlled by a hybrid BCI that combines both electroencephalographic (EEG) and electrooculographic (EOG) signals. EEG is used for horizontal movement of the cursor and EOG is used for selecting items or moving the cursor on a vertical axis. Using these methods, the users are able to navigate through all the included applications. EEG based BCIs have shown significant promise as an alternative input method for computer interaction, especially when integrated with other assistive technologies and applications developed with the aim to bypass certain limitations of this emerging technology.

Compared to the system described by (He et al., 2017), our application offers a more refined and user friendly interface that improves the overall user experience. Additionally, our text input method is based on Whisper, an AI-based speech-to-text model, making text entry faster and more accurate.

3 METHODOLOGY

3.1 The Headset and Setup

The EEG headset used is the g.tec Unicorn Hybrid Black headset. It has eight channels for the electrodes, the placement is optimized for motor imagery and for P300 paradigm, representing the right choice for BCI implementation in Unity. The electrodes are hybrid, meaning that they can be used with or without gel. Using gel on an electrode facilitates conductivity, so it provides a cleaner more reliable signal while having the downside of long setup and cleaning time. Using dry electrodes, without conductive gel, results in lower accuracy but requires significantly less setup time. This trade-off can be beneficial in contexts where precision is not critical, such as preliminary functionality testing with the headset.

Connecting the headset requires a proprietary application named Unicorn Suite Hybrid Black (g.tec, 2019) which is compatible with Windows 10 and 11. The headset connects to the application through Bluetooth using the provided wireless dongle. For the best connection it is recommended to temporarily disable the drivers of the integrated Bluetooth adapter before inserting into the computer the dongle provided by g.tec. Use with a generic Bluetooth adapter is possible, but not recommended due to the low accuracy. After connecting you can install the Unity SDK package from the main page. More detailed installation instructions can be found at (g.tec, 2024).

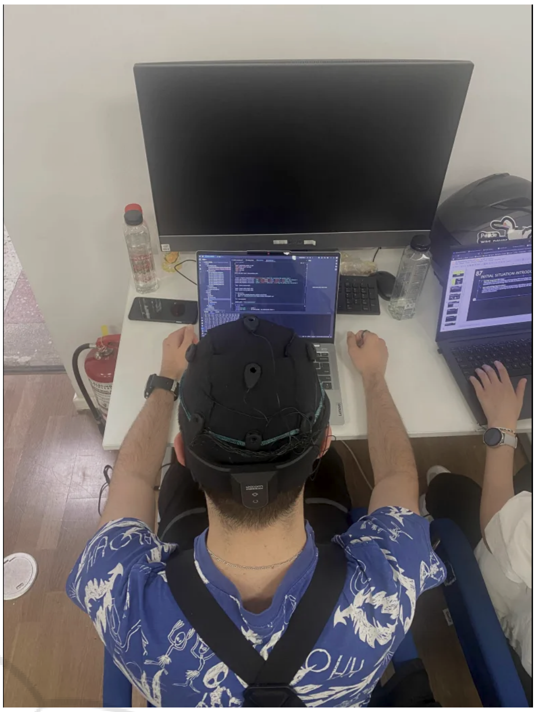

Figure 1: g.tec Unicorn Hybrid Black headset.

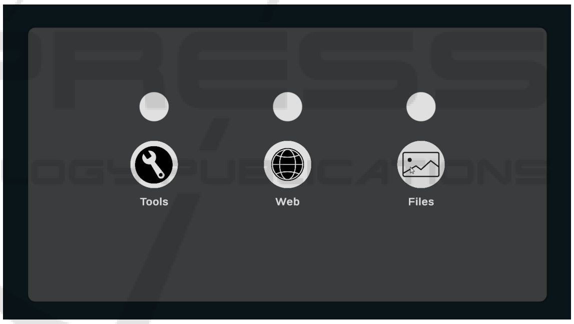

Figure 2: Application Page.

3.2 The Application

The application developed for this paper is a launcher style interface built around the g.tec Unicorn Hybrid Black EEG headset, see Figure 1. This interface contains multiple features such as: a calculator, a web browser with speech-to-text capabilities, an AI chatbot interface and a file viewer. All these features are usable with the help of the EEG headset. Each button from the application is paired with a white circle that starts flashing when the selection process begins. A button can be selected by focusing on it's respective flashing element. Figure 2 illustrates the application page with the function icons and the white circles which are the flashing elements.

The application was built using Unity 2022.3.60f1 (Technologies, 2005) with a 2D project setup. Each screen was created using a canvas set to 1920x1080 resolution. Switching screens is done by changing the coordinates of the camera. This multiple canvas approach was used in favor of switching scenes to ensure a stable integration with the Unicorn Hybrid Hybrid Black EEG headset. This design choice ensures data persistence because switching scenes causes problems with data continuity.

To evaluate the effectiveness of the g.tec Unicorn Hybrid Black EEG headset integration, a basic calculator app with simple buttons and controls was created, as shown in Figure 3. It uses a script to execute simple mathematical operations. Each button has a flashing object bound to it in order to ensure headset compatibility. Due to the large number of buttons, it is not possible to have all buttons flashing at the same time. To select a button, it is necessary to first select the column of that button, then select the button. This limitation is due to the SDK, which allows a maximum of fifteen flashing objects and the calculator has seventeen buttons. A dynamic button approach which would disable the elements that do not make sense

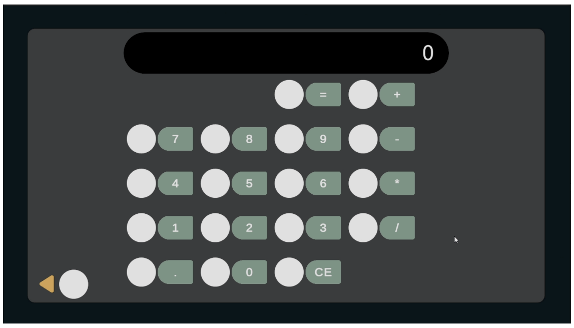

Figure 3: Calculator Page.

to be selected was considered, but even with such an approach, at some point the number of selectable buttons would be greater than the supported value.

The file viewer page consists of a vertical scroll view that is populated with buttons representing the files from the default photos folder as illustrated in Figure 4.

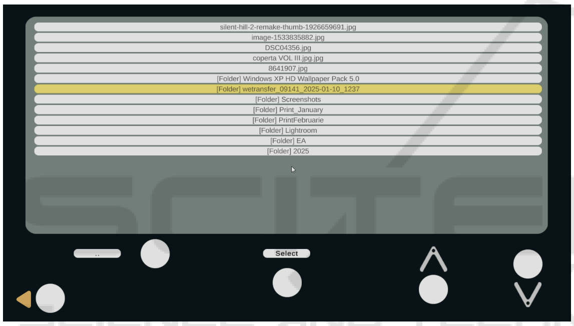

Figure 4: FileViewer.

This layout allows users to visually browse available content. Navigation is realized with the help of directional buttons (up and down) and a select button. The first highlighted item is always in the middle, making the distance from the starting point to the first file and the last file the same. Selecting a file of supported type (currently .png, .jpeg, .jpg) opens it in a full screen view as shown in Figure 5.

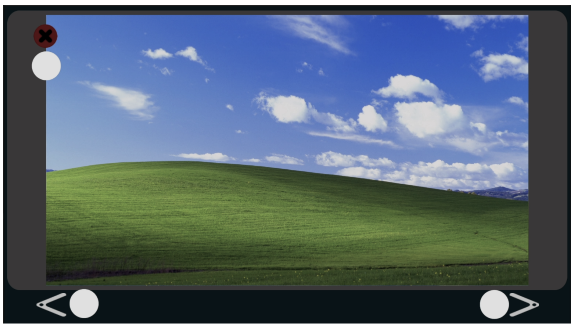

Figure 5: Fullscreen View.

In this mode, directional navigation buttons are available, allowing sequential browsing through the images. Closing this view takes the user back into folder view. The system allows navigation into subdirectories, enabling hierarchical file structure exploration. Going back to the parent folder is facilitated by a up one folder button.

The functionality part of the file viewer was implemented using two main scripts: SimpleFileBrowser and UINavigationSimulator. The file loading is done using SimpleFileBrowser script by accessing the directory specified by the user. If no directory is provided, it defaults to the user's image folder. Prior to populating the scroll view with file buttons, all previously generated file buttons are removed from the FileBrowser scroll view, to prevent duplication and ensure no residual items are left. Displaying a supported file type in fullscreen works by calling Unity's getTexture request to dynamically render the image. The UINavigationSimulator manages highlighting and navigation in the scroll view, simulating up, down or enter keystrokes.

For the speech to text functionality, the whisper.unity package (Macoron, 2023) was used. It is a Unity3D binding for whisper.cpp (ggml.org, 2022), making OpenAI's (OpenAI, 2022) automatic speech recognition model usable in Unity. This is a model that runs locally on the user's machine with no internet connection required to function. To set up this model in the project, an empty object named Whisper was created then loaded with the WhisperManager script. From this script one can select which model to run the speech-to-text and tweak other parameters from it such as: language, translation, toggle GPU usage and change advanced model settings such as the sampling method. The model chosen for this project is ggml-tiny. It is the default model from the package, but it provides a good performance. GPU compute is turned on to ensure fast processing and the sampling strategy is greedy sampling. For microphone access, it was necessary to create an empty object that was loaded with the MicrophoneRecord script that is provided in the package. This script captures audio input from the user's microphone that is meant to be given later to the model. A MicrophoneManagerScript is used to take the output from MicrophoneRecord and give it to the Whisper model to process.

The AI chatbot functionality, see Figure 6, was implemented using the Groq API which was chosen for the straightforward code integration, favorable rate limits and diverse AI model selection. The connection to the API servers was realized using a script named AiChatScript which manages the communication with the Groq servers by connecting to the appropriate endpoint. It also handles the transmission

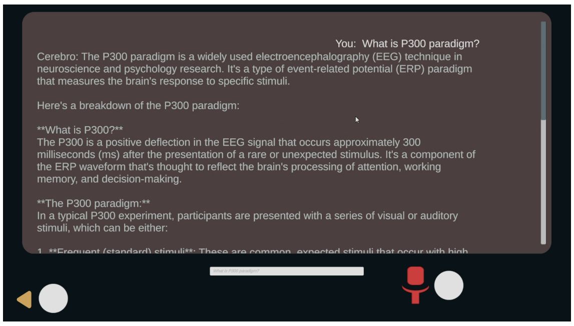

Figure 6: AI chatbot.

of user prompts and it processes responses from the server which are then sent to ChatUI script. This script manages the UI part from the AiChat screen canvas by taking the user inputs and server responses from AiChatScript and displaying them in an organized scroll view. The chosen AI model was llama-3.3-70b-versatile. It was selected because it offers low latency, ensuring fast responses and high performance. The used user prompt input method is the same AI speech-to-text model used for the browser search feature.

Unity does not have native support for web browsing. In order to overcome this limitation, we integrated the UnityWebBrowser package developed by Voltstro.com (Voltstro-Studios, 2023). This package embeds a Chromium based browser within the Unity environment which supports JavaScript injection, allowing programmatic control of web elements. The interaction with the browser (see Figure 7) can be done through traditional input methods like a mouse and keyboard or with the flashing elements that are controlled by JavaScript driven commands.

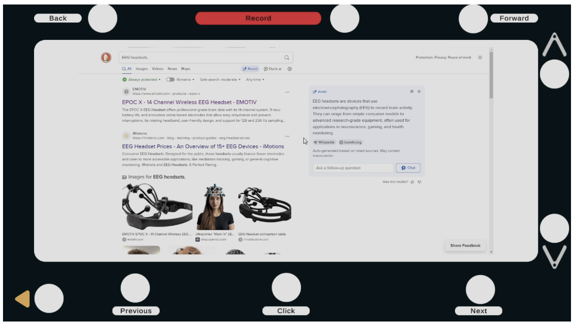

Figure 7: Browser Page.

Each interactive button from the interface controls the browser by injecting specific JavaScript functions, enabling actions such as: scrolling, highlighting clickable elements, simulating click events on the highlighted item and navigating back and forward through the pages. Text input for search is facilitated using an AI based speech-to-text model, offering hands free interaction and supporting accessibility.

3.3 Integrating EEG Headset Control

The headset integration was done by using the Unicorn Hybrid Black Unity SDK. The BCI selection works by flashing items on the screen at different intervals, the user selects an item by looking at it. According to the P300 paradigm, when a person sees a meaningful stimulus, after about 300 milliseconds there is a significant peak in brain activity which is detected by the headset. Knowing when the brain activity peak happened it is possible to detect which item flash triggered it, thus knowing the item the user was looking at. This is a core functionality of Unicorn Hybrid Black speller SDK. The whole application is based around this SDK and implements flashing objects that are used for the EEG headset selection process.

The first step in using the headset is the calibration, after the user presses the start button, the page changes to the training or calibration page, where the user is prompted to select the headset intended for use and tweak frequency settings. After that, the user specific calibration starts. If necessary, retraining is possible until the desired accuracy is obtained. The calibration has to be done every time the user starts the application.

Because the flashing objects cannot be put in a canvas and have to be Worldspace objects, each screen which is essentially a canvas was assigned an empty object that acts as a container for the flashing objects intended to be used for that screen. These objects are assigned a transform, a sprite renderer and a ButtonTrigger script which binds the object to the button it is meant to activate. The flash object conversion happens in the FlashObjectManager script through the PopulateFlashObjects function which takes all the children objects of a container and creates a list of objects to be used as placeholders for the flash objects. Then using the newly created list, it populates the Application Object fields of the ERPFlashController script. Before each change, the objects from the previous screen are cleared from the placeholder list as well as the ApplicationObject list to ensure that only the objects from the current screen are in the list.

When the controller detects a selection, it returns the ClassID of the selected element, using this ClassID we can call the TriggerButton function from the chosen object which then calls the OnClick methods assigned to the linked button. The whole BCI implementation works like a module that adds a new input method for the application.

3.4 Final Touches

To assure a better reliability of the P300 paradigm detection, some items that were too close were repositioned so that they would not interfere with each other. Other parts of the UI have been modified to have a more modern look. Fixed some bugs related with the selection script, sometimes causing multiple simultaneous actions being run, thus making the UI unusable.

The full implementation code can be found at: `https://github.com/AiRobo-UVT/EEG-headsets-application`.

4 EXPERIMENTS AND RESULTS

After the application was completed, its functionality has been evaluated using the Unicorn Hybrid Black Headset across various scenarios.

First, the application was tested under suboptimal conditions. Meaning no gel was used and the environment was not clear of electromagnetic interference and distractions, but the provided Bluetooth dongle was used. In this scenario, the results were unsatisfactory, with the selection accuracy being very low, comparable with random item selection.

In the next testing stage, conductive gel was applied to the electrodes, which has significantly improved the accuracy and made the selection process usable, even though some occasional selection errors still occurred.

The highest selection accuracy was achieved when the headset was used with conductive gel in a controlled environment with minimal electromagnetic interference, selection errors were still present at times, but the occurrences were rare.

Another method used to improve accuracy involved extending the training duration and adjusting the flash object parameters, such as flash duration. In general, longer training sessions resulted in higher selection accuracy and reduced selection times.

It was observed that making a short sound in your mind every time the target object flashes contributed to improving the selection accuracy.

A demo of the application is here `https://youtu.be/_HrQQcw5aYk`.

5 CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

Electroencephalogram (EEG) based brain computer interfaces (BCI) represent a promising and increasingly relevant technology that is very important for the future of human computer interaction by enabling control without the need of physical input. Despite their potential, EEG headsets remain sensitive to electrical noise and artifacts generated by muscle activity. For optimal signal quality, users are advised to operate the headset in an environment with low interference and minimal distractions.

The integration of EEG BCI control into software applications is slowly becoming more accessible. This is demonstrated by platforms such as g.tec's Unicorn Hybrid Black Unity SDK, which offers robust functionality and integration of their EEG headsets into Unity. However, the closed source nature of the program and the licensing costs might limit its adoption.

The developed launcher application successfully includes EEG headset input capabilities alongside speech-to-text functionality. When using the launcher with a headset that has wet electrodes and the bundled dongle, the selection performance is satisfactory. All the included features can be used without the need of physical input.

For now, EEG BCI input functions more reliably when paired with other non physical input methods such as speech-to-text, eye trackers and motion trackers. As it was discovered when setting up the BCI features of the application, a higher number of flashing objects could introduce problems in the interface by decreasing the accuracy or even making use impossible. It is not yet feasible to use EEG input to navigate applications with dense elements.

5.2 Future Work

Future work will focus on developing a more intuitive user interface and implementing smart objects that activate when the system predicts user intent.

Furthermore, an open machine learning–based EEG data processing framework should be designed for flashing item selection, enabling more granular control.

Further experiments with the current setup are also required to obtain more reliable results and a more accurate assessment of system performance.

ACKNOWLEDGMENTS

This work is co-funded by the European Union through the Erasmus+ project *AiRobo: Artificial Intelligence-based Robotics*, 2023-1-RO01-KA220-HED-000152418.

REFERENCES

Fok, S., Schwartz, R., Wronkiewicz, M., Holmes, C., Zhang, J., Somers, T., Bundy, D., and Leuthardt, E. (2011). An EEG-based brain computer interface for rehabilitation and restoration of hand control following stroke using ipsilateral cortical physiology. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6277–6280.

ggml.org (2022). Whisper.cpp. https://github.com/ggml-org/whisper.cpp.

g.tec, u. (2019). Unicorn suite hybrid black. https://github.com/unicorn-bi/Unicorn-Suite-Hybrid-Black.

g.tec, u. (2024). Unicorn hybrid black unity interface. https://github.com/unicorn-bi/Unicorn-Hybrid-Black-Unity-Interface.

He, S., Gu, Z., and Li, Y. (2017). A hybrid BCI web browser based on EEG and EOG signals. volume 2017, pages 1006–1009.

Kasim, M. A. A., Low, C. Y., Ayub, M. A., Zakaria, N. A. C., Salleh, M. H. M., Johar, K., and Hamli, H. (2017). User-Friendly LabVIEW GUI for Prosthetic Hand Control Using Emotiv EEG Headset. *Procedia Computer Science*, 105:276–281. 2016 IEEE International Symposium on Robotics and Intelligent Sensors, IRIS 2016, 17-20 December 2016, Tokyo, Japan.

Kline, A. and Desai, J. (2014). SIMULINK® based robotic hand control using Emotiv™ EEG headset. In *2014 40th Annual Northeast Bioengineering Conference (NEBEC)*, pages 1–2.

Macoron (2023). whisper.unity. https://github.com/Macoron/whisper.unity.

Mugler, E. M., Ruf, C. A., Halder, S., Bensch, M., and Kubler, A. (2010). Design and Implementation of a P300-Based Brain-Computer Interface for Controlling an Internet Browser. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(6):599–609.

OpenAI (2022). Whisper. https://github.com/openai/whisper.

Rușanu, O. A., Cristea, L., and Luculescu, M. C. (2020). LabVIEW and Android BCI Chat App Controlled By Voluntary Eye-Blinks Using NeuroSky Mindwave Mobile EEG Headset. In *2020 International Conference on e-Health and Bioengineering (EHB)*, pages 1–4.

Swee, Sim Kok, Teck Kiang, Kho Desmond, and You, Lim Zheng (2016). EEG Controlled Wheelchair. *MATEC Web of Conferences*, 51:02011.

Tang, J., Liu, Y., Hu, D., and Zhou, Z. (2018). Towards BCI-actuated smart wheelchair system. *BioMedical Engineering OnLine*, 17(1):111.

Tariq, M., Trivailo, P. M., and Simic, M. (2018). EEG-Based BCI Control Schemes for Lower-Limb Assistive-Robots. *Frontiers in Human Neuroscience*, Volume 12 - 2018.

Technologies, U. (2005). Unity. https://unity.com/.

Voltstro-Studios (2023). Unitywebbrowser. https://github.com/Voltstro-Studios/UnityWebBrowser.