

University of Nevada, Reno

**Integration of Assistive Technologies into 3D Simulations:
Exploratory Studies**

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in
Computer Science and Engineering

by
Angela Chan

Dr. Sergiu Dascalu/Thesis Advisor

December, 2016



THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

ANGELA CHAN

entitled

**Integration of Assistive Technologies into 3D Simulations:
Exploratory Studies**

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

Sergiu Dascalu, Ph.D., Advisor

Frederick C. Harris, Jr., Ph.D., Committee Member

Janet Usinger, Ph.D., Graduate School Representative

David W. Zeh, Ph.D., Dean, Graduate School

December, 2016

Abstract

Virtual worlds and environments have many purposes, ranging from games to scientific research. However, universal accessibility features in such virtual environments are limited. As the impairment prevalence rate increases yearly, so does the research interests in the field of assistive technologies. This work introduces research in assistive technologies and presents three software developments that explore the integration of assistive technologies within virtual environments, with a strong focus on Brain-Computer Interfaces. An accessible gaming system, a hands-free navigation software system, and a Brain-Computer Interaction plugin have been developed to study the capabilities of accessibility features within virtual 3D environments. Details of the specification, design, and implementation of these software applications are presented in the thesis. Observations and preliminary results as well as directions of future work are also included.

Acknowledgements

I would like to express my deepest appreciation for my advisor Dr. Sergiu Dascalu for the continuous support, encouragement, and guidance he has shown me. I would also like to thank Dr. Fred Harris and Dr. Janet Usinger for serving on my committee. I would like to thank all my friends and colleagues who have helped with my research, in particular Alex Gamino, Sam Tregillus, Rui Wu, Chase Carthen, Vinh Le, and Nolan Burfield. Lastly, I would like to thank my yoga instructors for keeping me mentally and emotionally stable during stressful times.

This material is based upon work supported by the National Science Foundation under grant number IIA-1329469.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Table of Contents

1	INTRODUCTION	1
2	AN OVERVIEW OF ASSISTIVE TECHNOLOGIES	4
	2.2 Landscape of Assistive Technologies.....	4
	2.3 Related Research and Development Efforts	5
	2.4 Major Institutional Laboratories/Organizations	7
	2.5 Assistive Product Design.....	7
	2.4 Applications.....	10
	2.5 Other Resources.....	11
	2.6 Discussion.....	12
	2.7 Summary.....	13
3	BRAIN COMPUTER INTERFACES	14
	3.1 Introduction.....	14
	3.2 Proposed Classification	15
	3.3 BCI Development	19
	3.4 BCI Applications	23
	3.5 Discussion.....	24
	3.6 Summary.....	25
4	RELATED WORK	27
	4.1 Introduction.....	27
	4.2 Visual Impairments.....	27
	4.3 Motor Impairments	36
	4.4 Issues/Limitations	38
	4.5 Summary.....	39
5	STUDY 1: SIM-ASSIST	41
	5.1 Introduction.....	41

5.2	High-level and Medium-level Design	41
5.3	Detailed Design	48
	Workflow Diagram.....	48
5.4	Prototype Walkthrough.....	50
5.5	Results.....	52
5.6	Summary.....	53
6	STUDY 2: GOOGLE STREET VIEW APPLICATION	54
6.1	Introduction.....	54
6.2	Application Overview.....	56
6.3	System Design	59
6.4	Software Testing.....	65
6.5	Preliminary Results.....	68
6.6	Summary.....	71
7	STUDY 3: BCI INTEGRATION WITH WATERSHED	72
7.1	Overview.....	72
7.2	BCI Plugin Integration Design	72
7.3	Use Case Modeling.....	76
7.4	System Walkthrough	80
7.4	Preliminary Results.....	83
7.5	Summary.....	85
8	CONCLUSIONS AND FUTURE WORK	86
	REFERENCES	90

List of Figures

1.1 Percentage of Individuals Living with Disabilities within U.S.	1
1.2 Age Distribution of Individuals with Disabilities in U.S.	2
2.1 The relationships between mainstream, universal design, assistive, and medical products	8
2.2 Iterative Multi-Stage Approach to Assistive Product Design	11
3.1 Abstract Taxonomy of BCI development	16
3.2 Generic model of BCI development cycle	20
4.1 Number of keywords used in a query search by blind sighted users	32
4.2 Difficulty choosing the right keywords in a query by blind and sighted users	32
4.3 Number of result pages visited in a search by blind and sighted users	33
4.4 Utilization of the refining search function by blind and sighted users	33
4.5 Various types of difficulties encountered by blind and sighted users	34
4.6 Accessibility of useful information in a search by blind and sighted users	34
5.1 System-level diagram of the Sim-Assist system	43
5.2 Program units for Sim-Assist system	44
5.3 Description for PlayGame program unit	44
5.4 Descriptions for ToggleDetection program unit	45
5.5 Descriptions for InterpretFacialGestures program unit	45
5.6 Descriptions for InterpretBrainWaves program unit	45
5.7 Descriptions for ControlMousePointer program unit	46
5.8 Descriptions for CheckForCustomVoice program unit	46
5.9 High-level activity workflow diagram of Sim-Assist	48
5.10 Activity Diagram for “Facial Expression Command”	48
5.11 Activity Diagram for “BCI-Controlled Mouse Pointer”	50
5.12 Sim-Assist welcome window	50
5.13 Player information window	51
5.14 Snapshot of Air Hockey game	52
5.15 Player using the Emotiv Epoc’s BCI controls to interact with the 3D Air Hockey game	53
6.1 Illustration of navigation through Movie Maps Interface	54

6.2 Control mappings for navigation within web application	57
6.3 Screenshot of Emotiv’s training software	58
6.4 Activity diagram of BCI controller within web application	59
6.5 Street View Application Program Units.....	60
6.6 Description for WebApplication program unit	61
6.7 Description for ToggleDetection program unit	62
6.8 Description for CognitiveActionCheck program unit	62
6.9 Description for ExpressiveActionCheck program unit	63
6.10 Description for InterpretCognitive program unit	63
6.11 Description for InterpretExpressive program unit	64
6.12 Description for TrainCommands program unit	64
6.13 Description for ManageCustomCommands program unit	65
6.14 Snapshot of Emotiv Control Panel’s Contact Quality window	66
6.15 Illustration of Emotiv’s training software	67
6.16 Snapshot of Street View application’s web interface	68
6.17 Software testing questionnaire	69
7.1 High-level diagram of BCI plugin’s integration with the Virtual Watershed Client	73
7.2 Workflow diagram of BCI plugin interaction with Virtual Watershed Client.....	74
7.3 Activity diagram for BCI-Controller.....	75
7.4 Activity diagram for head-mouse controller	76
7.5 Use case diagram of user interaction with Virtual Watershed client through BCI plugin..	77
7.6 Use Case Descriptions for Login, LoadProfile, TrainManually, and ZoomIn.....	78
7.7 Use Case Descriptions for ZoomOut, MoveCamera, Navigate, and ControlMouse	79
7.8 Snapshot of Emotiv Control Panel’s training session	80
7.9 Virtual Watershed Client Main Menu (with BCI integration)	81
7.10 Login Menu of Virtual Watershed Client (with BCI Integration)	81
7.11 Snapshot of Virtual Watershed Catalog Search	82
7.12 Snapshot of Virtual Watershed Client in action.....	83

List of Tables

5.1 Functional requirements for Sim-Assist	42
--	----

1 INTRODUCTION

Virtualization of a world or environment can serve many purposes in applications such as gaming, data visualization, and general social interaction. It is estimated that one billion people have experience with such virtualization, and over a quarter of a billion people actively participate in these applications regularly [Spence, 2008]. On the other hand, there are currently very few virtual applications that are accessible to people with impairments. Furthermore, the prevalence rate in people with impairments is increasing every year, thus raising the interests in assistive technologies by many researchers.

According to the *2014 Disability Statistics Annual Report*, the average percentage of disability in the U.S. population in 2013 was 12.7%; this computes to approximately 41 million people in the U.S. with some disability [Institute on Disability, 2014]. Figure 1.1 illustrates the various percentages of individuals with disabilities throughout the states. It is apparent that in 2013, Utah possessed the lowest percentage of the population of those with disabilities. The highest percentage belonged to West Virginia. Additionally, higher percentages were found in: the southern states; states around lower Mississippi; Main; Oregon; and New Mexico.

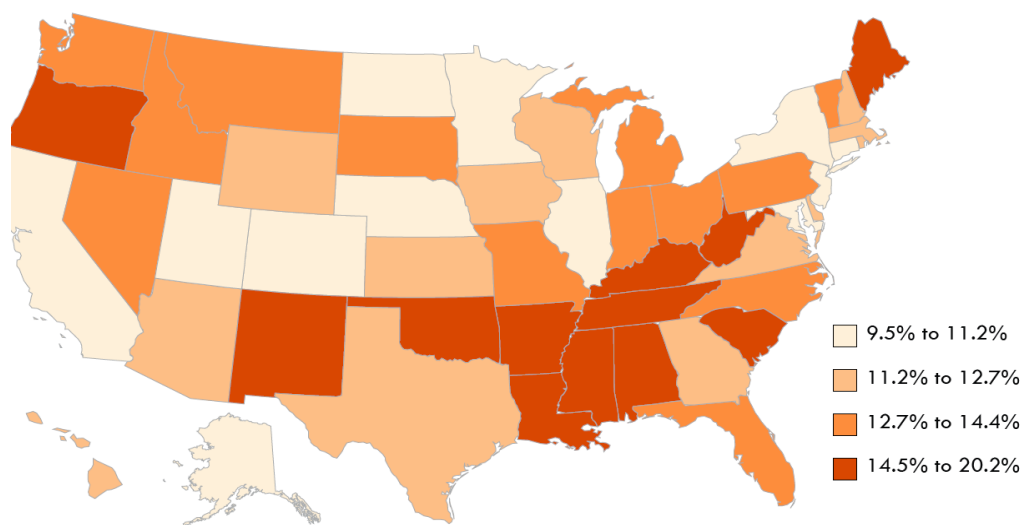


Figure 1.1: Percentage of Individuals Living with Disabilities within U.S. [Institute on Disability, 2014]

Individuals who had some disability were primarily at least 18 years old. The age distribution of people with disabilities in the U.S. population can be shown in Figure 1.2. Out of the individuals with disabilities, nearly 52% were between the ages 18 and 63; approximately 40% were at least 65 years old; roughly 7% were children and teenagers between 5 and 17; and 0.4% were under the age of 5 [Institute on Disability, 2014].

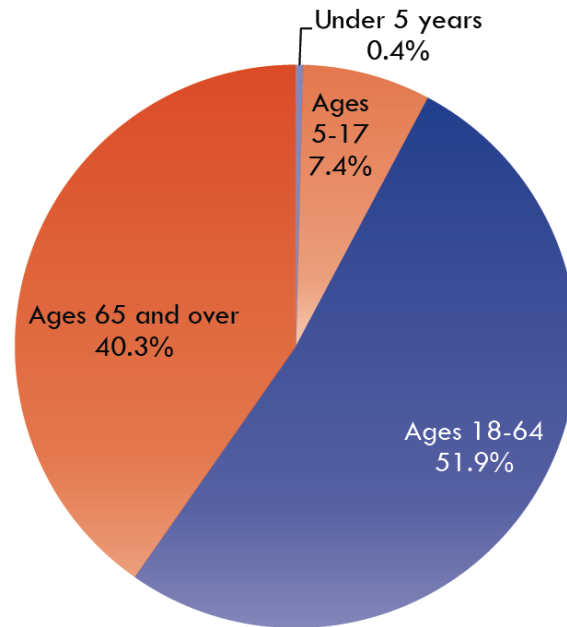


Figure 1.2: Age Distribution of Individuals with Disabilities [Institute on Disability, 2014]

As stated before, nearly 41 million people have some type of disability, and the majority of them are adults. This implies that a large number of people with disabilities will seek employment or are already employed. Luckily, the 1990 Americans with Disabilities Act (ADA) grants equal rights and opportunities to people with disabilities [FCTD, 2015]. Assistive technologies (AT) are created with such people in mind.

Integrating software-based assistive technology into virtual environments is the main motivation for this work. We developed three different software components that utilize assistive technologies, with a strong emphasis in Brain-Computer Interfaces, to interact with three separate virtual environments. The first system is Sim-Assist, which allows people with or without disabilities to play the three-dimensional virtual game of Air Hockey using speech and Brain-Computer interfacing. The second system is a Google

Street View navigation software system, which allows the user to navigate through any street using their thoughts and facial expressions. The third development is a Brain-Computer Interaction plugin that is used within the WC-WAVE's Virtual Watershed Platform, which allows users to interface the Virtual Watershed Client using head movements and expression recognition. As we explored the methods, results, and observed shortcomings of the first two systems, we gained insight that contributed to our final development, which is the integration of BCI technologies with the Virtual Watershed Client.

The remainder of this thesis is structured as follows. Chapter 2 presents an overview of assistive technologies. Chapter 3 contains the background information on brain computer interfaces, which is a subset of assistive technologies. Chapter 4 highlights some of the existing systems and applications. Chapter 5 features a study of integrating assistive technologies into a dynamic virtual environment. Chapter 6 presents an exploration of integrating brain computer interface technologies into a virtual world. Chapter 7 investigates the capabilities and results of enhancing accessibility in the Virtual Watershed Platform. Finally, Chapter 8 presents the conclusions of the thesis and outlines directions of future work.

2 AN OVERVIEW OF ASSISTIVE TECHNOLOGIES

This chapter outlines the background on Assistive Technologies. The related research and development efforts of various Assistive Technologies are presented in conjunction with some information on the current major laboratories and organizations working in this field. This chapter then provides some overview on assistive product design, alongside some of the main applications that resulted from it.

2.2 Landscape of Assistive Technologies

Assistive technologies (ATs) can be defined as any type of technology used to improve, enhance, or maintain the functional capabilities of individuals with disabilities. Such technology may include hearing aids, wheelchairs, electronic devices, or educational software. This chapter focuses strictly on software-based assistive technologies (SBAT).

Software-based assistive technologies are products that offer people with disabilities better accessibility to computers. For those who are unable to see, hear, speak, or lack motor skills, assistive technologies can alleviate their lives a little more. Some of its distinguishing categories include brain-computer interfacing, gesture recognition, and speech recognition. Brain-computer interfacing will mainly target users who lack motor skills, but can also be used by users who are unable to see, hear, and speak; gesture recognition will target users who are unable to see, hear, or speak; and speech recognition will target users who are unable to see and possibly those who lack motor skills.

One of the main challenges that lies in developing software for ATs is the collaboration between developers and disability specialists. There is often a disconnect, making the usability of the end product poor for the users. Another challenge is that the complex solutions are not necessarily the best. The more tools that are embedded in a program, the more complex the system becomes, making it more difficult for users to learn.

There also seems to be a large spectrum in the research and development of assistive technologies because there are various types of disabilities. This ranges from aiding students learn in the academic environment (applies to those with learning disabilities, blindness, and deafness) to assisting the elderly. Research in software-based

assistive technologies is important as society grows increasingly more high-tech. As computers are meant to improve the quality of our lives, they are used daily. Therefore, it makes sense to use computers as one of the tools, if not the main tool, for assisting people with disabilities. In order to keep up with the advancements in everyday technology, research into the software utilized for assistive technologies becomes important. The most distinguishing characters of software-based assistive technologies are the abilities to convert text from the computer to auditory speech and vice versa, and respond to speech commands, gesture commands, and certain brain signals. These substantial advancements within this area would benefit people who have a disability such as: auditory; hearing; speech, or motor impairments.

2.3 Related Research and Development Efforts

Assistance for people who are care-dependent has gain importance as a research field because it could potentially improve the quality of their lives as well as their care-givers. Such people may be dependent due to motor, sensory, or cognitive impairments. Due to such impairments, performing daily life activities such as feeding, hygiene, personal care, etc. may be difficult. Gomez et al presented and evaluated a system that shows adaptive manuals for daily life activities for people with disabilities [Gomez et al, 2013]. This proposed system is based on mobile devices and QR codes, and will help with the rehabilitation process of patients with acquired brain injury. Additionally, Donoghue et al also aimed to create a system for people with motor disabilities [Donoghue et al, 2007]. Specifically, they developed a human application of neural interface systems (NICs) for people with paralysis. However, the NIC is in its early stages of development. The pilot trial results provided evidence that a neuron-based control system is feasible. Thus, in this system, signals can be detected, decoded, and used for real time operation of computer software, ATs, and other devices.

According to the 2012 National Health Interview Survey, 20.6 million American adults reported experiencing vision loss [American Foundation for the Blind, 2012]. Individuals who experience vision loss can be referred to those who claim that they have difficulty seeing, even while wearing glasses or contact lenses, and people who are blind or unable to see at all. Blindness and vision impairment also increases rapidly as people

grow older, particularly after the age of 75 [Narasimhan et al, 2009]. At this age, people struggle to learn Braille and become accustomed to visual impairment without dependence on others. With this issue as motivation, Narasimhan et al (2009) developed a system to provide the visually impaired an independent shopping experience. This system allows the visually impaired shopper to find the correct aisles and shelves that contain the desired product. The system can also distinguish between different products of the same type (e.g., Marinara sauce vs. Alfredo sauce) once the general shelf location of the desired product has been identified. This system can also be used for accomplishing other daily tasks such as cooking.

Although there are different levels of disability, assistive technologies should not solely target severe disabilities. According to Edyburn, little attention has been dedicated to the assistive technology needs of students with mild disabilities. Mild disabilities may consist of learning disabilities, behavioral disorders, and mental retardation [Edyburn, 2006]. Some tools such as *iPing*, *Co:Writer*, and *WebMath* have been developed for these targeted users, but there is little evidence suggesting students with mild disabilities have access to such technologies [Edyburn, 2006]. In addition, due to the lack of performance measurement tools for assessing users' progress in the academic field, it is difficult to justify claims that the technology will enhance performance.

There have also been efforts to develop assistive technology for the elderly. So far, assistive technology targeted towards those who are older has yet to improve the quality of their lives. To target this issue, Bright and Coventry (2013) suggested potential design strategies to maximize a product's usability and usefulness for older adults. By doing so, their main goal was to show the importance of considering both psychological and socio-emotional design requirements when creating such technology.

2.4 Major Institutional Laboratories/Organizations

In this section, we name some of the major laboratories and organizations throughout the United States that focus on the research and development of Assistive Technologies.

The Assistive Technologies Research Center (ATRC), at Wright State University, is an organization that focuses on the research and development of Assistive Technologies [Center of Assistive Research Technologies, 2010]. It is partnered with organizations (located in Miami Valley and Ohio) involved with similar interests in ATs. The center aims to conduct basic and applied research for ATs for people with disabilities and the elderly.

EnableMart is currently the worldwide leading organization in distributing assistive technology for those with “blindness, low vision, hearing loss, learning disorders, and other disabilities” [School Health Corporation, 2016]. EnableMart has been supporting the community of people with disabilities for over a decade by providing accessibility products such as augmentative devices, Braille displays, and alternative input devices.

The Open Source Assistive Technology Software (OATS) Project is a pilot project that aims to investigate the demands and interests for open source software within assistive technology research [The OATS Project Consortium, 2014]. It was claimed to be the first open source repository that is explicitly dedicated to assistive technology.

The Georgia Project for Assistive Technology (GPAT) (2016) supports school systems to provide assistive devices and services to students with disabilities. GPAT aims to increase student’s access to the appropriate assistive technology devices as well as enhance educator knowledge of assistive technology. Doing so will improve student achievements, productivity, independence, and inclusion within society.

2.5 Assistive Product Design

Assistive products are considered to be a branch within the category of assistive technology, and can be divided into subcategories such as: mainstream, universal design, assistive, rehabilitation, and medical products [Hersh, 2016]. Mainstream products refer to those that are designed for the general public, and are available for purchase from any standard retail store. However, they are often designed without consideration of the needs

of people who are disabled. Universal design products aim to make products accessible and useable by a wide range of users. This disregards factors such as types of disability, age, culture, etc. Such products use a “design-for-all” approach. Assistive products are those designed specifically for people who are disabled, in addition to the elderly. Rehabilitation products are those designed for people with disabilities, the elderly, or people who had been severely ill or injured. Rehabilitation products allow these people to restore the functions that they have lost. Medical products are designed to support health care practices and promote healing in people who are ill.

The relationships among the different subcategories are illustrated in Figure 2.1, based on [Hersh, 2016]. In the Venn diagram, the intersections between the ovals (“design for all”, “assistive products”, “medical products”, and “rehabilitation products”) indicate the overlap between the different concepts. Additionally, the intersections between these ovals and the “mainstream products” circle indicate the extent mainstream products possess, with one or more of these features.

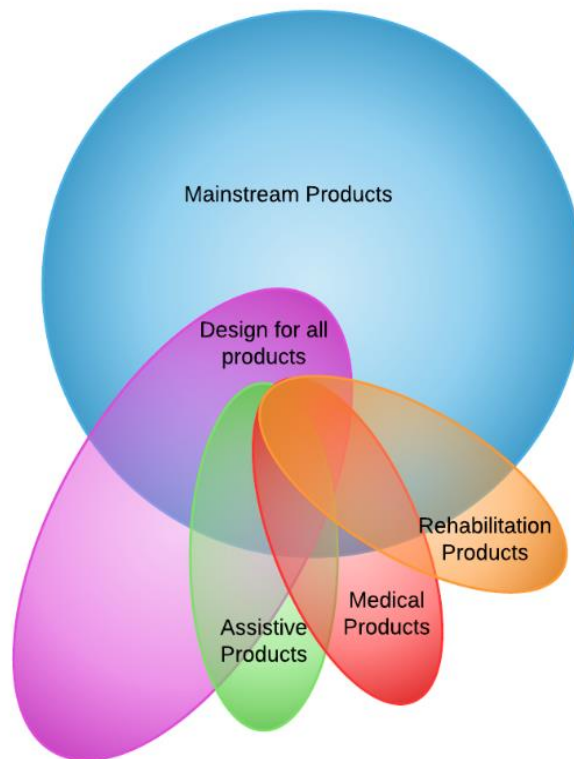


Figure 2.1: The relationships between mainstream, universal design, assistive, and medical products.

Assistive products can be divided into those used to “make existing mainstream products accessible” and those “required to overcome the barriers due to partially or totally inaccessible infrastructures and environments” [Hersh, 2016]. Although there may always be a need for some assistive device, the application of design for all principles to the design of mainstream products will remove the need for some of them. For example, screen readers are required to make computers accessible and usable for the visually impaired. However, many standard computer setups already include the screen reading functionality. With additional improvements in functionality and performance, these screen readers will eventually remove the need for an assistive device that functions explicitly as a screen reader.

Designing assistive products involves a multi-stage iterative process [Hersh, 2016]. This process includes frequent backtracking to previous stages, which may cause overlap between stages. For systems that involve human-computer interaction or software, the best practice is considered to be a user-centered design approach. It is especially important for assistive products to use such an approach because the end users may have specific requirements and designers may not be aware of such requirements. This method has shown to produce more usable systems and save time in large products [Bradley and Dunlop, 2005].

A general approach towards a user-centered design can be broken down into four main steps, as depicted in Figure 2.2, based on [Hersh, 2016]. The first step involves learning about the users and the tasks they wish to carry out. The second step consists of using the current knowledge of the users to contribute to the design. In the third step, users are repeatedly presented with a prototype for evaluation. In the final step, any issues are identified and resolved in the redesign. This is then repeated as often as needed until a final prototype is produced.

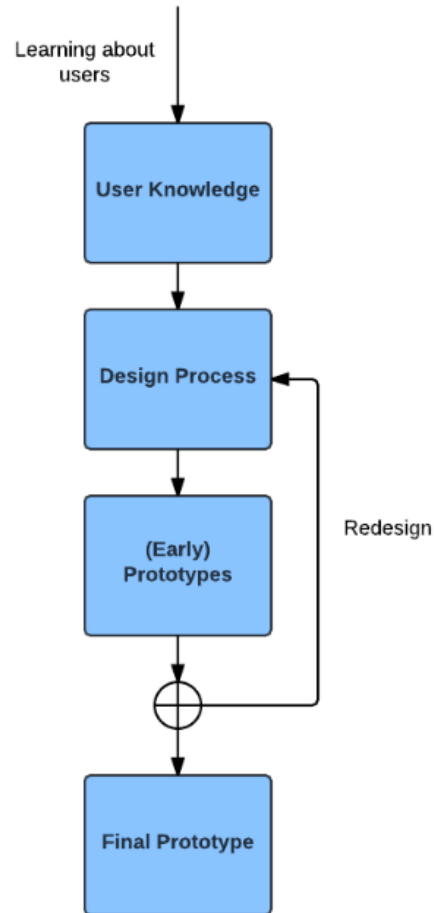


Figure 2.2: Iterative Multi-Stage Approach to Assistive Product Design

2.4 Applications

Many applications of SBATs are currently available for commercial release. However, there is not one application that covers all of the various disabilities (visual, speech, auditory, and motor impairments). Since different types of disabilities can be quite demanding in requirements, applications are mainly focused on one or two specific categories of disability.

Microsoft Windows' *Narrator* is a screen reader that is included in the Windows operating system, and possesses text-to-speech capabilities and describes events (e.g., error messages) [Microsoft, 2016]. This allows the user to use the PC without a display, making it suitable for people who are visually impaired. *Narrator* is available free of charge and in various languages. Another software for people who are unable to see is *Dragon*. *Dragon*

is a cross platform speech recognition software that types as the user talks [Nuance, 2016]. Dragon also possesses the ability to perform various speech commands (e.g., format text, update social media, create presentations, send emails, and more) to boost productivity and save time.

iCommunicator is a software that aids people who lack the ability to hear [iCommunicator, 2010]. It has speech to text, speech/text to video sign language, and speech/text to computer generated voice capabilities. *iCommunicator* is currently being used in: academic environments; the government; healthcare; and corporate settings across the United States and Canada. *Dragon*, as mentioned before, may also be useful for those whose hearing is deteriorating. This software transcribes words as a person is speaking. However, the level of accuracy in transcriptions depends on the amount of noise in the environment.

For people with severe motor skills, controlling a wheelchair with their minds would be most beneficial. Despite their poor signal-to-noise ratio, recent experiments have shown for the first time that electroencephalogram (EEG) is sufficient for humans to continuously control a mobile robot similar to a wheelchair. Such an application would be suitable for people with severe motor disabilities. In Galán et al's paper, experimental results have shown that subjects can quickly master their EEG-based Brain Computer Interface (BCI) to control a wheelchair [Galan et al, 2008]. They can also autonomously operate the BCI over a long period of time, which demonstrated it's the system's ability to allow continuous mental control for the user.

2.5 Other Resources

Additional resources that may help contribute to software-based assistive-technology research include articles, websites, and videos.

Websites

Microsoft Windows is compatible with various assistive technologies (e.g., screen readers, magnifiers, and specialty hardware for impaired users). The Microsoft Accessibility website provides a list of specialty software and hardware to help aid users who are impaired [Microsoft, 2015]. Another notable website is Neurogadget.com.

Neurogadget.com is one of the leading news sources for brain computer interfaces [Neurogadget, 2016]. This website offers information on trending BCI, neuro-technology, and brain controlled applications. Although their main focus is on BCI, some of their posts mention BCI's integration with other disability areas for assistive technology. One useful article that can be found on this site is titled, "Blind Man Reads Braille Patterns with His Eyes, Using Implanted Retinal Prosthesis" [Andrew, 2012]. Retinal prostheses are devices that partially restore the vision of those who are blind. The prosthesis used, as mentioned in the article, is a 6x10 electrode array that had been implanted on a patient's retina [Andrew, 2012]. This implant was connected to a visor that combines a camera, an onboard computer, and an inductive coil for wireless data transfer to the electrodes. This article mentions that the patient was able to accurately recognize 89% of the words that had been flashing at him, letter by letter [Andrew, 2012].

Videos

The video, "What is Assistive Technology (A.T.)", describes in detail what assistive technology is and the main areas (access, communication, and entertainment) in which they can help in [RSLSteeper Assistive Technology, 2013]. Each scenario is illustrated per area, in how assistive technology can come into play. This video will deem useful for developers who are starting to develop ATs. Another video, "Apps for the iPad and Assistive Technology" shows a demonstration of how new apps on the iPad can replace similar existing assistive technology devices [INDATA Project, 2011]. The "Franklin Bill Reader", "Garmin" navigation system, and the "Color I.D." devices were shown to have an equivalent app available on the iPad. This video shows a window of opportunity for developers who are unable to afford expensive hardware for development. Additionally, apps available on the iPad are more cost-friendly for future consumers as well.

2.6 Discussion

There seems to be minimal research and development for those who are speech and hearing impaired. This is probably due to other devices and software that are already available such as speech-to-text software, hearing aids, and cochlear implants. Further research and development in software-based assistive technologies can be expected to

continue its main focuses on people with motor and visual impairments. Development for SBATs can be expected to continue its integration with mobile devices, and eventually targeting smartphone devices. Currently, almost every American owns a smartphone device. By transforming assistive technology software as an app for a smartphone, it will become one less device the disabled person will need to depend on.

In the future, we expect there will be an increase in the need for assistive devices in the form of Braille keyboards and displays for people who are visually impaired. Including such devices in a standard PC setup would increase its complexity and price, which would be unfair to both people with and without disabilities. We also expect an increase in eye tracking devices for people with motor disabilities. The launch of Google Glass has already opened a window of opportunity for other developers. Therefore, eye tracking features may be available in the near future for many mobile devices, thus further assisting people who are unable to move.

2.7 Summary

In this chapter, we provided an overview on the landscape of Assistive Technologies. This chapter highlighted some of the major development efforts in Assistive Technologies and included some information on the major labs and organizations that dedicate their research towards this field. This chapter also presented some details on assistive product design, alongside some of the main applications that resulted from it.

3 BRAIN COMPUTER INTERFACES

This chapter is based on our previously published conference paper [Chan et al, 2015] and poster [Chan and Dascalu, 2015], which are based on our prior work. This chapter aims to provide an overview of the research and development of Brain Computer Interfaces (BCIs). We propose and describe our taxonomy of BCI systems, breaking them down into three main categories: independent and dependent, invasive and noninvasive, and endogenous and exogenous. Finally, we explore the development of BCI systems and examine existing applications that were created as a result of BCI research.

3.1 Introduction

Advances in Human Computer Interaction (HCI) continue to play important roles in society. One growing development in HCI is the concept of a direct Brain Computer Interface. The goal of BCI is to improve the quality of one's life, and its full potential has been yet to be explored. Much of the existing research mainly has focused on people with severe motor disabilities, but BCIs also have potential in the domain of immersive video games, communication, and robotics. However, many challenges arise in the development of such systems. The main factors that affect the performance of a BCI system include the type of brain signals used as data, data acquisition methods, the algorithms that are used to translate the obtained data, the hardware that facilitates user control, the type of feedback the user receives when executing commands, and the characteristics of the users themselves. Therefore, future improvements in BCI systems require structured, well-controlled studies that evaluate and compare alternatives and combinations of signals, various data acquisition methods, alternative translation algorithms, and various control applications offered to different users.

Most current BCI systems do not readily support such structured research and development [Shain et al, 2003]. Though many researchers have tried to accomplish this (e.g., [Guger et al, 2001] and [Kennedy and Bakay, 1998]), up to now BCI research has only offered previews of the lackluster capabilities of current BCI systems. So far, BCI systems are restricted to utilize one type of brain signal at a given time, and specific data

acquisition, monitoring, and feature translations methods [Wolpaw et al, 2002]. These systems are still unable to fully unleash the potential capabilities of BCI, which might be achieved by utilizing and comparing various brain signals and processing methods. By demonstrating the capabilities of this yet underrated form of interaction through innovation, a more immersive means to operate this technology could be achieved. This is important because through immersion, not only can BCI boost the standard of living, society's unyielding mindset towards operating such new technology may also improve. Therefore, further progress in BCI research and development is needed.

3.2 Proposed Classification

Brain Computer Interface development is traditionally divided into several categories: independent or dependent, invasive or noninvasive, and exogenous or endogenous. Figure 3.1 illustrates our suggested taxonomy on BCI development [Chan et al, 2015], presenting the various types of current BCI that fit into their respective categories.

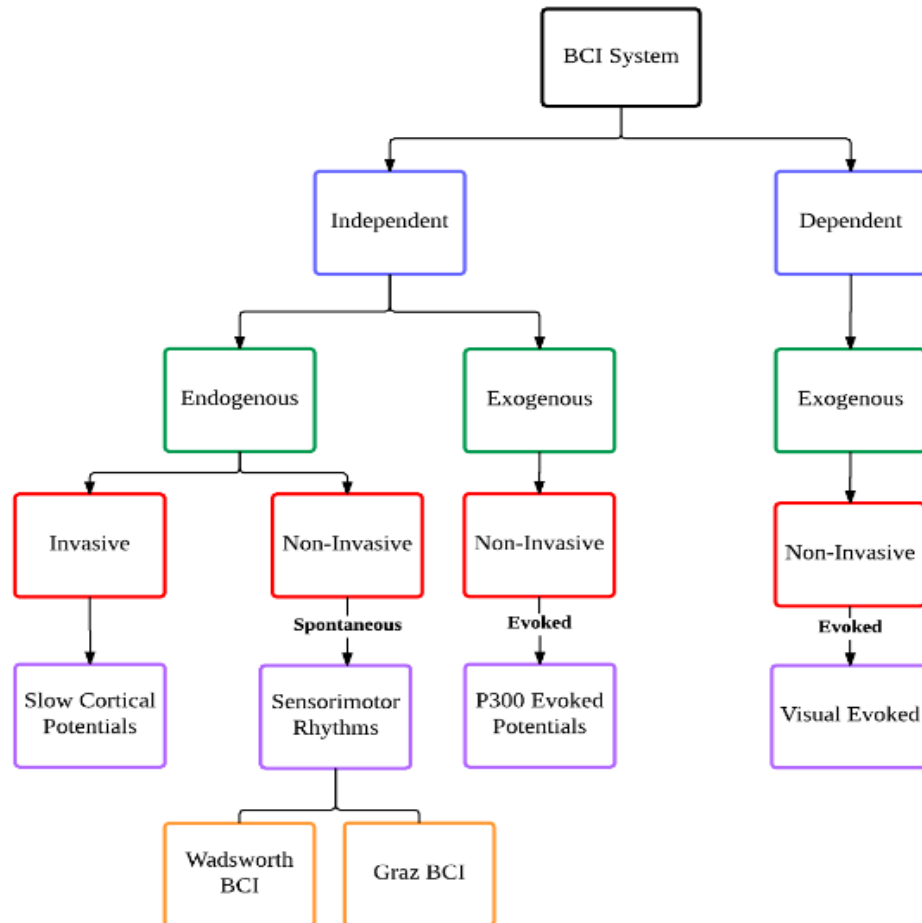


Figure 3.1: Abstract Taxonomy of BCI development [Chan et al, 2015]

Independent vs. Dependent

Independent and dependent BCI systems are distinguished by how reliant the system is on additional types of activity in order to function. An independent BCI does not depend on the brain's pathways (i.e., peripheral nerves or muscles); activity in such brain output is unnecessary to create the brain activity (e.g. EEG) required to execute a certain command [Wolpaw, 2002]. For example, a user can choose a specific letter by solely thinking about it. In one BCI system, the user is presented a matrix of sequentially flashing letters [Smith and Donchin, 1970] and [Farewell and Donchin, 1998]. As the letters flash, the user produces a P300 potential, allowing the user to select the currently lit letter. It should be noted that the generation of the appropriate signals depends on the user's intent instead of the direction of the user's gaze. Instead of relying on such peripheral nerves,

independent BCIs provide the brain with a new communication path, which could be more useful and theoretically interesting than dependent BCIs. In addition, independent BCIs may be useful for those with severe neuromuscular disabilities.

In contrast to independent, dependent BCIs use the activity in the brain's normal output pathways to generate the brain activity (e.g. EEG) required for the system to function. Similar to the previously mentioned system for independent BCI, Sutter described a dependent BCI system that also presents the user with a matrix of sequentially flashing letters [Sutter, 1992]. However, with this system, the user was able to select a specific letter by looking at it. As the user looks directly at the flashing letters, his/her visual evoked potential (VEP) was recorded over the visual cortex for each flashing letter; the largest recorded potential determined the user's selected letter. In this system, the EEG component of the brain is used to carry out its task, but the signal is generated through the direction of the user's gaze.

Invasive vs Noninvasive

Invasive and noninvasive BCI systems are distinguished by their method of data extraction. Invasive BCI requires implanting foreign materials into the subject's body. Such materials may include large electrode setups or chemical molecules. However, in order for invasive research to be safe enough for humans, it first requires animal research. It is no surprise why invasive BCI methods are avoided, due to social pressures to stop funding of such research. Invasive BCIs operate by monitoring single-neuron activity within the subject's brain [Krusienski et al, 2011]. While such systems have an improved spatial resolution and might deliver control signals with numerous degrees of freedom, BCIs that depend on electrodes within the subject's cortex face considerable difficulties in attaining and sustaining unwavering long-term recordings. However, due to encapsulation, it is likely that the signals within the invading electrodes will eventually degrade [Leuthardt et al, 2004]. In addition, small changes in the locations of the electrodes can move the recording sites away from areas that are easily recorded. These issues are crucial obstacles that currently prohibit their clinical use in humans.

Invasive BCI can also be performed through electro-corticography (ECoG), also known as intracranial EEG (IEEG), which is the practice of placing electrodes directly on

the brain to record electrical activity. Due to the low signal-to-noise ratio of EEG signals, ECoG is usually an alternative method of extracting data from brain activity. Electrical signals must also be conducted through the subject's skull when using EEG, and since bone has a low electrical conductivity, ECoG would contain a much higher spatial resolution. In addition, ECoG is expected to be safer and have a greater stability in the long-term, compared to the previously mentioned approach. This is due to the subdural electrode arrays that are used to record ECOG, which takes away the need for electrode that penetrate into the cortex [Leuthardt et al, 2004].

On the other hand, non-invasive procedures do not require any kind of implantation and the subject gets to interface with the machine through wearable devices. Most non-invasive BCI systems use electroencephalogram (EEG) signals (i.e., the electrical brain activity recorded from electrodes placed on the scalp). Non-invasive BCI is comparably more convenient, safe, and inexpensive. However, as mentioned before, this method results in low spatial resolutions. In addition, artifacts such as electromyographic (EMG) signals may obscure readings, and non-invasive BCI systems usually require extensive training for users [Li and Zhang, 2012].

To further classify BCI systems, non-invasive BCIs can be classified as “evoked” or “spontaneous”. An evoked BCI depends heavily on evoked potentials, which reflects the immediate automatic responses of the brain to some external stimuli. In principle, it is easy to detect evoked potentials through scalp electrodes. The P300 and steady-state visual potential (SSVEP) are the most commonly explored in BCI research. In addition, Slow Cortical Potentials (SCP) are also sometimes used in evoked BCI systems. The necessity of external stimulation does, however, restrict the applicability of evoked potentials to a limited range of tasks.

In contrast, spontaneous BCI systems allow the user to carry out cognitive processes freely [Neuper and Kilmesch, 2006] because it eliminates the need for external stimulation. This allows for the user to interact with the system in a more natural manner through intent. Such a method is especially beneficial when controlling robotic devices. Some signals spontaneous BCI may depend on are event related de/synchronization (ERD/ERS) and Steady State Evoked Potentials (SSEP).

Exogenous vs Endogenous

A BCI system is classified as exogenous or endogenous depending on the nature of the recorded signal. Exogenous BCI systems depend on neuron activity evoked by external stimuli. Such stimuli include VEPs or auditory evoked potentials. Exogenous BCI systems do not require intensive training since it is easy to setup their control signals (SSVEPs and P300) [Grimann et al, 2007]. In addition, the signal controls can be detected with a single EEG channel, capable of an information rate of up to 60 bits/min [Grimann et al, 2007].

In contrast, endogenous systems do not rely on an external stimulus; they are based mainly on brain rhythms and other potentials. The users have to learn the skill of producing specific patterns, which are decoded by the system. Training the users using neuro-feedback usually does this. The length of the training varies by subject as well as the experimental strategy and training environment. The strategy chosen for the experiment determines how the user learns and what they must do to produce the required brain activity patterns. Grimann et al (2007) describe two approaches for endogenous systems: Operant conditioning and performance of specific mental tasks.

BCI systems that use the operant conditioning strategy train using feedback. The user must rely on the feedback to learn to produce the intended brain activity. This is a similar strategy proposed in calibration- free robotics [Grizou et al, 2014]. In contrast, motor imagery is the most common mental task used to produce brain patterns that can be reliably produced and distinguished. Motor imagery is activated through the imagination of movements of limbs. The users perform such mental tasks without physically executing the corresponding movement. Doing so produces desynchronization (ERD) and event-related synchronization (ERS) [Neuper and Kilmesch, 2006].

3.3 BCI Development

In this section, we look first at the typical development framework for BCIs, then briefly survey modern BCI systems. Related issues and limitations are also examined.

3.3.1 Development Framework

At initial stages of development, all BCI systems follow a similar framework, as shown in Figure 3.2.

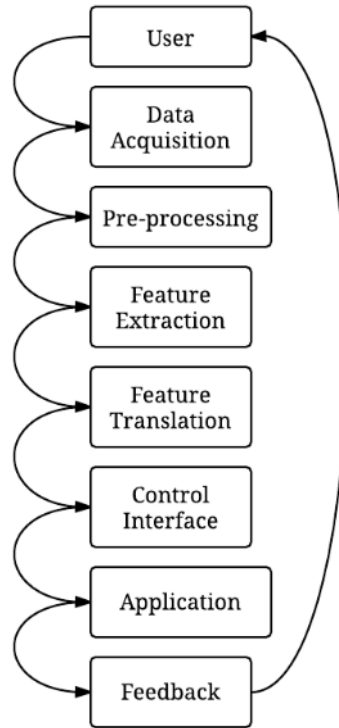


Figure 3.2: Generic model of BCI development cycle

First, the user is either monitored for specific mental states or brain activity, or the user intentionally executes a mental task by modifying his/her brain state [Miatliuk et al, 2014]. Raw brain signals resulting from the execution or mental state monitoring are then acquired by the EEG device, which get sent to the preprocessing stage. Here, the effects of artifacts and noise will be reduced to improve the signal-to-noise-ratio. Features from the preprocessed data are then extracted, informing the system on what it is supposed to detect. These features become translated into labels with logical meaning, which become inputs to the control interface. These inputs become semantic controls for the application or device, which will provide feedback, shown through the output. The feedback gets delivered back to the user, and the cycle restarts.

3.3.2 Modern BCI Systems

Modern BCI systems fall into five groups: visual evoked potentials (VEPs); slow cortical potentials; P300 evoked potentials; mu and beta rhythms (sensorimotor rhythms); and neuronal action potentials [Wolpaw et al, 2002]. They are distinguished as such based on the type of brain signals they use. Slow cortical potential interfaces rely on VEPs, which classify them to be dependent BCIs. On the other hand, the other four groups are classified to be independent BCIs. The following section discusses systems that have been developed using these five groups

Visual evoked potentials

Visual evoked potentials refer to electrical potentials that are evoked by brief visual stimuli. These potentials are recorded from the visual cortex and their waveforms are extracted from the EEG. VEPs are mainly used to measure the visual pathways from the eye to the brain's visual cortex. Middendorf et al (2000) presented a method for using VEPs to determine the direction of the user's gaze. The user faces a screen displaying several virtual buttons that flash at different rates. Once the user directs his/her gaze at a button, the system determines the frequency of the photic driving response over the user's visual cortex. When the frequency matches that of a displayed button, the system concludes that the user wishes to select it.

Slow cortical potentials

Slow cortical potentials (SCPs) are slow voltage shifts within an EEG that can last from one to several seconds. Slow cortical potentials correspond to the changes in the level of cortical activity. Positive SCPs are related to the decreased activity in neurons, whereas negative SCPs are associated with neuronal activity [Birbaumer et al, 1990]. These signals can be self-regulated by any type of user to control external devices through BCI. Shifts in SCP can be used to move a cursor and select targets presented on a computer screen. People can be trained to generate these shifts using a thought-translation device. This device shows visual-auditory marks so that the user can learn how to shift the SCP levels [Hinterberger et al, 2004].

P300 evoked potentials

P300 evoked potentials are the peaks found in an EEG due to infrequent visual, auditory, or somatosensory stimuli. The use of P300-based BCI systems does not require any training. However, its performance may be reduced since the user eventually adapts to the infrequent stimulus, causing the P300 amplitude to decrease [Ravden and Polich, 1999]. Typical applications of P300-based BCI systems consist of matrix of symbols in which selection of such symbols depends on the user's gaze.

Mu and beta rhythms

Mu and beta rhythms combine to create sensorimotor rhythms, which are oscillations in the brain activity localized in the mu and beta bands respectively. Sensorimotor rhythms are associated with motor imagery without any movement [Pfurtscheller et al, 1997]. Sensorimotor rhythms help design endogenous BCIs, which operate based on the user's intent. Extensive user training is vital because people tend to struggle with motor imagery. Imagining visual images of the corresponding real movements is insufficient for a BCI system. This is because sensorimotor rhythm patterns are dissimilar to motor imagery. Therefore, training should emphasize kinesthetic experiences rather than visual representations of the movements.

One well-known system, presented by Graz, uses sensorimotor rhythms as control signals [Pfurtscheller et al, 2003]. The Graz BCI system is based on ERD and ERS of sensorimotor rhythms. The user participates in an initial session to select a motor imagery paradigm. In each series of timed trials, the user imagines an action while EEG is submitted for feature extraction. After interpreting the user's motor imagery into an output, it is presented back to the user in the form of online feedback. In contrast, with the Wadsworth BCI system, people learn to control a cursor in one or two dimensions to target on a computer screen [Wolpaw et al, 2000]. During the initial sessions, most users employ motor imagery to control the cursor. However, as they continue to train, imagery becomes less important and users are able to move the cursor from pure muscle memory.

Neuronal action potentials

In BCIs that rely on action potentials, cone electrodes are inserted into the motor cortex to detect the single cortical neuron-induced potentials [Kennedy and Bakay, 1998].

So far, only one user has been able to control neuronal firing rates and use this ability to move a cursor to select items on a computer screen. By demonstrating this control in people who are nearly completely paralyzed, this initial data propose that cortical neurons can support an independent system [Wolpaw et al, 2002]. Although recurring illness and medication effects limited training, the results have been encouraging.

3.3.3 Issues and Limitations

Several issues obscure the further development and widespread application of brainwave technology. The first issue is the information transfer rate. The maximum information rates of current BCI devices are offered at 5-25 bits/min [Cheng et al, 2002]. Increasing this rate would be potentially useful in developing applications where the users need to interact with their environment in a timely fashion. Another issue lies in the training time for users to become familiar with the system. In spontaneous BCI systems, user training is unnecessary, but evoked BCIs often require extensive training. However, one of the main challenges of a spontaneous BCI system is the non-stationary nature of the EEG signals, which is apparent in the differences between the training and test data sets [Grizou et al, 2014]. In addition, changes in the user's brain processes (e.g., due to distractions, fatigue, etc.) during online operation may affect the system's performance as well

Intuitively, the less invasive the technique, the more likely it can be used in a wide range of applications. Implanted electrodes provide stability of location, freedom from artifacts, and much higher signal-to-noise ratio (SNR). But one difficulty in such a system is how to determine the locations and the number of the electrodes. Another difficulty is how to keep the system stable over long periods.

3.4 BCI Applications

Robotics

Within the last ten years, researchers have been able to successfully perform invasive procedures that allow primates to control machines with their brains. Carmena et al (2003) demonstrated a primate's ability to learn to control a robot arm through brain-machine interfaces. In addition, Serruya et al (2002) suggested that neural based control of movement might be suitable for humans through multi-electrode array implant. This

theory was based on a monkey's success in moving a computer cursor to any new position in its workspace.

For humans, non-invasive methods based on EEG signals are preferable because of ethical concerns and medical risks. Despite their poor signal-to-noise ratio, recent experiments have shown for the first time that EEG is sufficient for humans to continuously control a mobile robot similar to a wheelchair. In Galán et al's paper, experimental results have shown that subjects can quickly master their EEG-based BCI to control a wheelchair [Galán et al, 2008]. They can also autonomously operate the BCI over a long period of time, which demonstrated it is the system's ability to allow continuous mental control for the user.

In a paper written by Millán et al, two human subjects learned to drive a robot between rooms in a house-like environment by mental control only [Millán et al, 2003]. Furthermore, mental control was only marginally worse than manual control on the same task. It was also shown that a device was able to operate through BCI-based control without any prior calibrations [Grizou et al, 2014]. This device was able to learn an intelligent behavior solely through feedback received after performing each action.

Entertainment

Recent developed BCI applications have focused on those who do not have any disabilities. One of such applications incorporates BCI with gaming. Researchers at University College Dublin and MediaLabEurope have created a BCI-based virtual reality game, *MindBalance* [Lalor et al, 2005]. The character is balancing on a tightrope, and the goal of this game is to maintain its balance using only EEG.

One of the more popular novel devices in present-day is the cat ears headband, manufactured by the company *NeuroSky* [NeuroSky, 2016]. Popular among cosplayers, these cat ears move in correspondence with the user's mood. Though not very useful, this headband is used mainly as an accessory to a costume.

3.5 Discussion

We believe that non-invasive BCI systems will become more predominant due to ethical and safety reasons. Unfortunately, the frustrations that arise due to the non-invasive approach will continue to remain and hinder a rapid progression in the field. However, this

is not to say that BCI research for fully capable and robust systems will eventually reach a dead end. Some issues may be resolved if BCI systems are not so heavily dependent on specific brain signals. So far, existing BCI systems rely on methods that require intensive user calibration. This opens up many opportunities for false detection within input brain signals.

We have shown that BCIs can play a significant role in AI research due to the varied and promising potential it possesses. Judging from its current integration with AI, we can expect a BCI research to continue in this direction. Within the last ten years, the progress of BCI development was revealed through its integration with robotics. It was shown that a device was able to operate through BCI-based control without any prior calibrations [Grizou et al, 2014]. This device was able to learn an intelligent behavior solely through feedback received after performing each action. As shown by Millán et al, two human subjects learned to drive the robot between rooms in a house-like environment by mental control only [Millán et al, 2003]. These examples demonstrate how easily the two fields complement each other.

To further exploit the potentials of BCI, machine learning methods can be applied to BCIs as well [Muller et al, 2004]. A successful BCI system, capable to learn, possesses the ability to classify various features obtained from the user's brain activity and performs an action as a result. Classification, essentially, is what allows for the 'mind control magic' to occur in BCIs. Unfortunately, classification methods in general still have not been perfected. Furthermore, classification becomes difficult in BCIs because one must distinguish brain activity intended for control from other types of activity.

3.6 Summary

In this chapter, we have provided a general overview of research and development for BCI systems. This chapter focused on the software end of BCIs, providing a general sense of the various ways one can approach BCI development. The three categories of BCIs were discussed, in addition to their relationships to modern BCI systems. This chapter also

presented the issues and limitations that BCI development currently faces as well as several existing applications developed.

4 RELATED WORK

This chapter outlines and highlights some of the major contributions in assistive technologies for those with visual and motor impairments. Issues and limitations of the presented existing works are also further explored.

4.1 Introduction

As mentioned in Chapter 2, assistive technologies were developed in order to improve the quality of disabled people's lives. However, it appears that research and development efforts have not yet advanced to a stage that allows people with disabilities to interact with 3D environments. Instead, there are existing user interfaces (UIs) that allow people with disabilities to interact with 2D environments in such a way that is more user-friendly and accessible for them. Through the developments of these existing works, we may see user interfaces for 3D environments in the future. This chapter explores these mentioned works and the methods that were used in such interfaces.

4.2 Visual Impairments

There are various degrees of visual impairment, also described as "low vision" [Low Vision Aids and Technology, 2016]. Variations of low vision may include: a diminished acuity; the loss of one's field of vision; one's sensitivity to light; and distorted vision or loss of contrast. Currently, a large selection of assistive technologies is made commercially available to the general public such as screen readers and magnifiers, braille keyboards, and text-to-speech software. However, these generic products only solve the most basic problems that one with low vision must encounter. Researchers are now exploring alternative methods that can further expand the limited capabilities of these people.

Auditory Interfaces

People who are blind depend heavily on their sense of hearing. Previous research in spatial audio revealed that it has potential as a new medium for creating nonvisual interfaces [Burgess, 1992]. Researchers Weinberg and Hellstrom (2000) used spatial audio

to create an auditory interface that utilizes direct manipulation for graphical user interfaces (GUIs) to become more accessible for the blind. Although direct manipulation improves the average user's usability and learnability of a system, it is not available in modern screen readers that most blind users depend on [Weinberg and Hellstrom, 2000].

Direct manipulation is based on three main properties [Hutchins et al, 1985]:

1. Objects are depicted in a continuous manner—Users do not need to browse and look for the objects.
2. Physical actions or labeled button presses—Users do not need to type in a command on the keyboard or select from a list of actions.
3. Rapid reversible actions whose impact on the target object is immediately visible—Users can immediately see results of an operation, and they can be easily reversible.

Previous research in this type of interaction for blind users revealed systems that include using tactile devices with audio ([Mynatt and Weber, 1994] and [Petrie et al, 1995]) and a system that uses 3D audio and a data glove [Savidis et al, 1996]. However, in these systems, only a subset of the interface objects is presented to the user, and the user has to browse the auditory space in order to get an overview of the object. Therefore, the user is interacting directly with interface objects instead of using direct manipulation, according to the definition mentioned earlier. Another system that uses auditory interfaces is a system developed by Pitt and Edwards (1991). In this system, the cursor becomes a “virtual microphone” that is used for selecting items from a menu [Pitt and Edwards, 1991].

In the paper written by Weinberg and Hellstrom (2000), they aim to answer two research questions:

1. Is it possible for a UI to utilize auditory direct manipulation?
2. Is this method of interaction interesting, or should different forms for interaction with an auditory interface be investigated?

In order to address the questions, the authors implemented an auditory form of the game “Towers of Hanoi” [Weinberg and Hellstrom, 2000]. Each disc had a corresponding sound that was different in pitch and timbre. The smaller the disc, the higher the pitch. The sounds also had slightly different tunes with respect to each other to further distinguish the

size of the disc. The tower on which the disc is determined by the various levels of stereo panning and amplitude envelopes. Furthermore, the vertical position of the disc is distinguished by the length of the sound. In this system, the mouse is similar to the one used by Pitts and Edwards. Users can move the cursor to each tower based on differences in volume. Results from testing this system revealed that auditory direct manipulation is possible and possesses the potential of being significant for blind computer users [Weinberg and Hellstrom, 2000].

Search Engines

In today's world, people conveniently gather information primarily through the Internet. It provides answers to the simplest or most complex of questions in an instant. However, braille displays and speech-to-text software are insufficient for navigation through webpages, and information that is coded within the layout of web pages is undetected by some screen readers [Donker et al 2002]. Some approaches to increase the usability of search engines for the visually impaired include using auditory interfaces or redesigning the structure of the search engine itself, as shown by the works of Donker and Andronico, respectively.

There are three main obstacles which cause complications for blind users to interact with Graphical User Interfaces (GUIs): text-to-speech systems are unable to read the screen output because it is represented as a pixel map in the graphics memory; blind users cannot effectively use pointing devices without any form of feedback; and textual descriptions of recognizable icons may be long, inaccurate, and complex [Boyd, 1991]. Donker even suggested a fourth obstacle—the layout “barrier” of web pages [Donker et al 2002].

Donker focused mainly on breaking the obstacles blind users face when browsing through graphically dense web pages [Donker et al 2002]. They designed a system aimed for better identification of the graphical elements as well as navigation through webpages using a “three-dimensional auditory interaction realm” (AIR) [Donker et al 2002]. Each web page component is represented by a distinct hearcon (an audio based icon) in the AIR, which allows the user to localize the source of the sound [Donker et al 2002]. The users can use the pointing device to interact with the hearcons, similar to how a sighted user interacts with the GUI using a mouse. A “torch” metaphor is used to implement how a user

is able to listen for hearcons if there are too many hearcons displayed on a webpage [Donker et al 2002]. Once a hearcon is selected, a function is executed, or its name is read by the screen reader. Although Donker's concept of the AIR client seems intriguing and promising, it demonstrated poor performance in the usability testing. Results revealed that the test subjects processed tasks less effectively and efficiently with the system than their currently used screen readers. In addition, the subjects took a longer amount of time and identified fewer relevant objects on the webpages [Donker et al 2002]. The subjects were also unable to identify the structure of the webpage's layout.

In order to design a search engine that caters to the needs of the blind, Andronico et al first conducted a study that assessed the accessibility and usability issues that arise while interacting with a screen reader [Andronico et al, 2006]. To test for accessibility, web accessibility tools such as Bobby and Torquemada were used [Cogan, 2015]. Results have shown that out of seven different search tools (i.e., Google, Altavista, Yahoo, Excite, HotBot, Vivisimo, and Kartoo), only Google proved to satisfy the minimum accessibility degree [Andronico et al, 2006]. In contrast, frequent errors were discovered in the other search engines with the same defects (e.g., inability to easily read table content due to table layout, and inconsistency of the same error throughout a page).

A four-part questionnaire¹ was used to investigate usability issues. In part one, results suggested that it was difficult for the blind users to use an interface that was crowded with elements. Results from part two revealed that 60% of users always used the same search engine, but 79% of users tried performing queries with different search products [Andronico et al, 2006]. This suggests that users have attempted to find better user interfaces. Furthermore, 58% often utilize search engines, while 33% only use them when needed [Andronico et al, 2006].

Part three of the questionnaire asked how users utilize search engines. For query searches, only 92% of sighted users and 69% of blind users specified more than one keyword [Andronico et al, 2006], as shown in Figure 4.1. In addition, Figure 4.2 revealed

¹ Fifty-two people participated in the questionnaire. Out of the participants, 25% were visually impaired and 75% were sighted; 33% were female and 67% were male; and the age range was between 20-60+ years old [Andronico et al, 2006].

that 67% of sighted users and 38% of blind users did not find it difficult to choose the right keywords [Andronico et al, 2006]. Once the search results were displayed, only 15% of blind users needed to explore more than two result pages, whereas 80% of these users accessed only the first two [Andronico et al, 2006], shown in Figure 4.3. One explanation for such a great difference in percentage is that the blind need more time to visit each result page. This makes it more difficult to browse a large number of pages. Furthermore, only 23% of blind use the refining search function [Andronico et al, 2006], shown in Figure 4.4.

The final part of the questionnaire aimed to identify the greatest difficulties for users. Results presented that blind users not only struggled to choose the right keywords, they also had difficulty reading results and accessing interfaces, as shown in Figure 4.5 and Figure 4.6 [Andronico et al, 2006]. Only 38% of blind users were able to sometimes find useful information and 8% would almost never find useful information [Andronico et al, 2006].

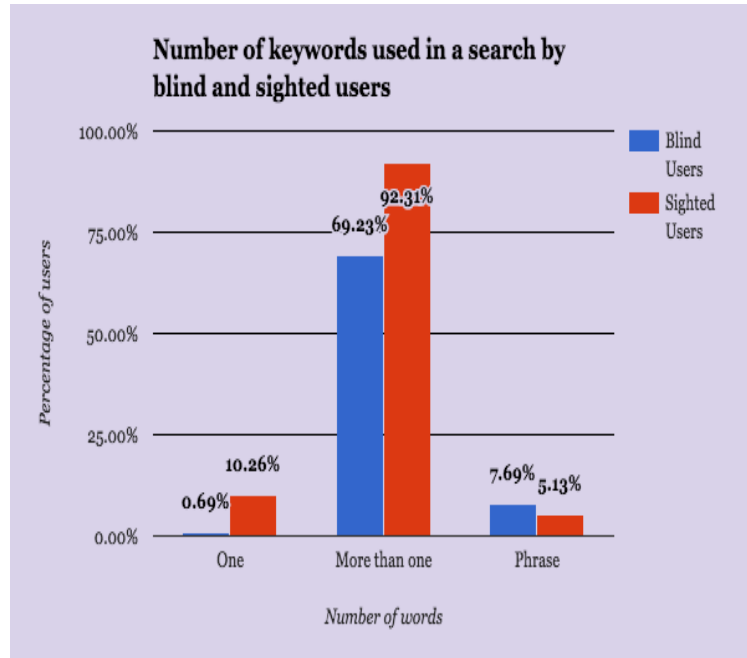


Figure 4.1: Number of keywords used in a query search by blind and sighted users

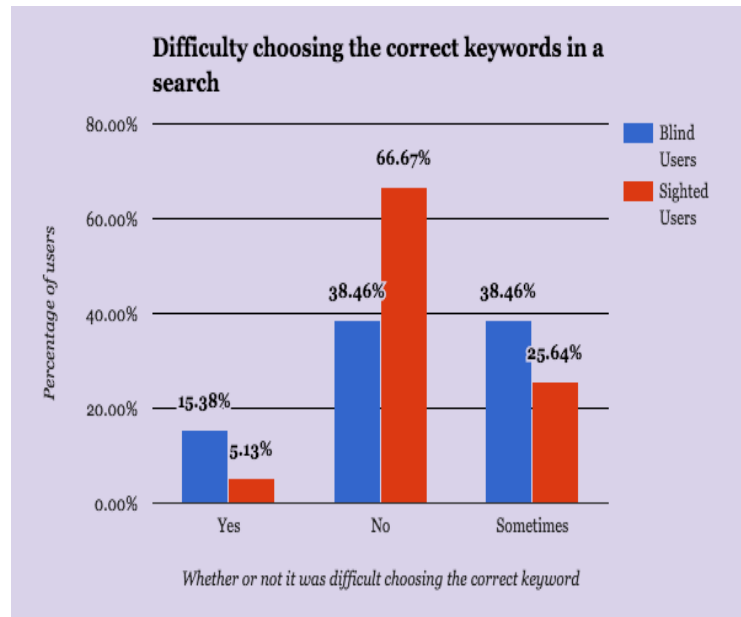


Figure 4.2: Difficulty choosing the right keywords in a query by blind and sighted users

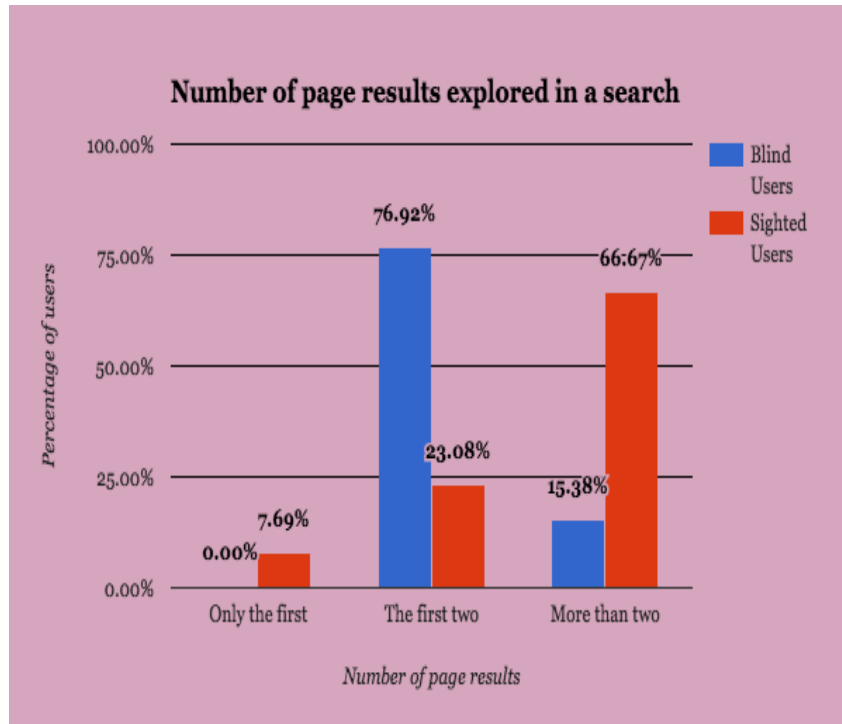


Figure 4.3: Number of result pages visited in a search by blind and sighted users

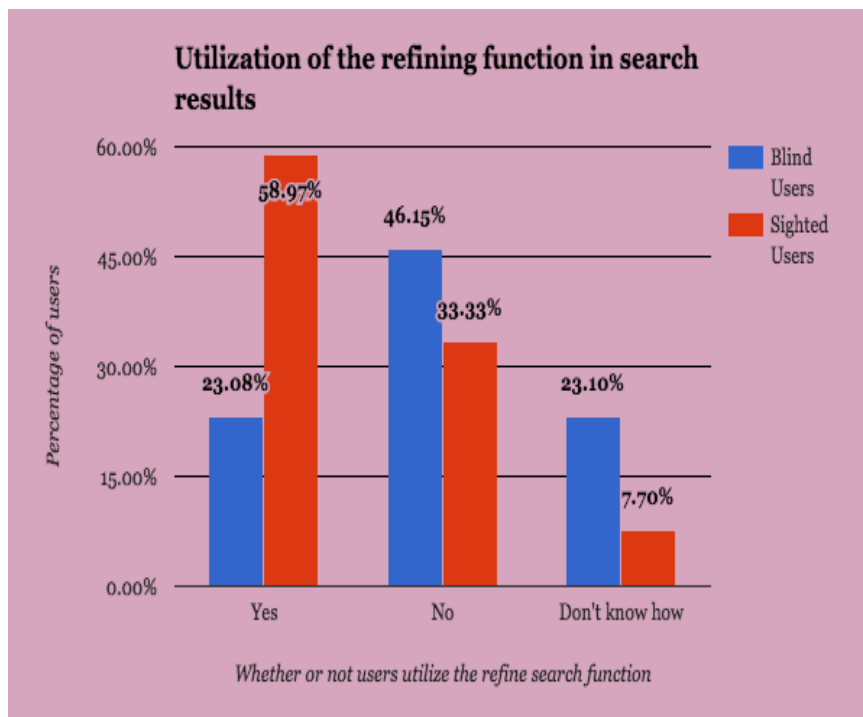


Figure 4.4: Utilization of the refining search function by blind and sighted users

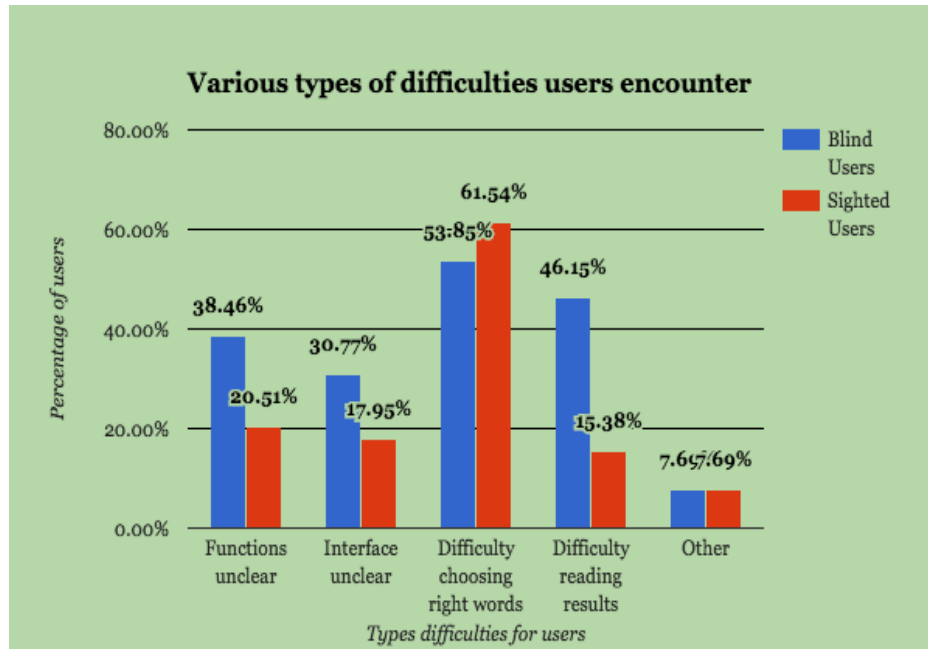


Figure 4.5: Various types of difficulties encountered by blind and sighted users

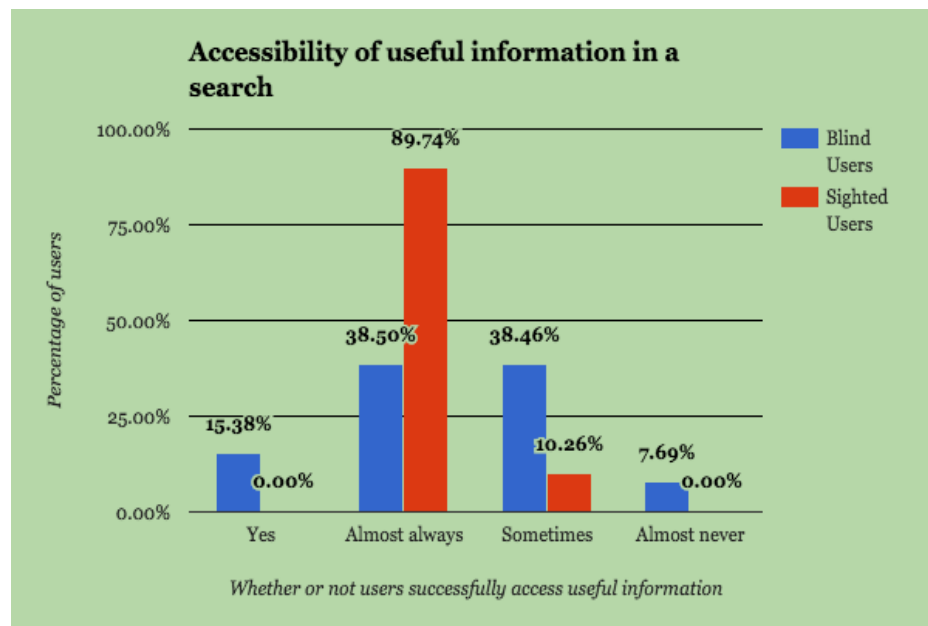


Figure 4.6: Accessibility of useful information in a search by blind and sighted users

By obtaining these data, the researchers were able to formulate a set of guidelines for designing search engine interfaces for users with visual impairments [Andronico et al, 2006]:

1. Place edit fields and search options in an easily accessible location.
2. Search results should be highlighted.
3. Arrange the list of results with their summary.
4. Separate sponsored links from other results.
5. Links that lead to result pages should be placed at the end of the list (not before). Help or navigation links would also be useful for moving around the page.
6. Levels of important should be assigned to various elements for fast navigation through a page.
7. Different notification sounds should be used.
8. Aural style sheets may be used for creating more usable and accessible web contents.

In this study, the proposed guidelines were applied to the Google UI as a specific case. The authors reengineered the code in the Google UI while applying their guidelines, but maintained the integrity of the Google layout. The successes and limitations from testing the modified interface had not yet been evaluated, but will be one of the authors' future works. The authors wish to demonstrate that a user with visual impairments can execute fewer steps to accomplish a specific search task using their interface compared to the original [Andronico, 2006].

Custom Interfaces

One of the main reasons why people with disabilities find it difficult to use computers is the disconnect between the users' individual needs and the designers' assumptions [Gajos et al, 2007]. Most graphical user interfaces are designed to only cater to the general public. However, there are some cases in which the interface can be easily rendered to suit one's unique capabilities. For users with restricted vision, enlarging the fonts and visual cues appropriately may suffice. Similarly, for users who partially lack

motor control, it may be easier to interact with interfaces that incorporate widgets with large targets so that dragging is unnecessary. In cases where the use of ATs is required, a specific GUI design may be desired. For instance, it might be more user-friendly to design a hierarchical structured interface for navigation through UIs that utilize screen readers.

When designing a GUI, it is important to focus on the users' needs. However, it is unreasonable to design a specific GUI that handles each individual's unique preferences. A paper written by Gajos et al proposes generating a custom user interface which would better suit users, especially for users with visual and motor impairments [Gajos et al, 2007]. The authors present a system called SUPPLE++ that automatically generates UIs that caters specifically to either an individual's motor or vision capabilities, or both. One of their major contributions included a method to predict the capabilities for each user by automatically selecting features of a custom regression model. This compensates for the fact that Fitts' law, a human movement model, often does not adequately describe the performance of people with disabilities [Pino et al, 2003]. Another contribution was an efficient optimization-based algorithm for automatically generating GUIs adapted to users' motor capabilities. The authors also presented a modification to their system that allows UIs to be manually adjusted to users' vision capabilities. Their preliminary results indicated that their system allowed one user to complete tasks, which she could not perform using a standard interface. Compared to other users, the SUPPLE++ allowed the user to save time by 20% [Gajos et al, 2007].

4.3 Motor Impairments

Most of the interactions between the user and the GUI rely on selection operations [Barreto et al, 2000]. Actions such as inputting data can also be achieved by selecting various letters through on-screen keyboards. This type of selection operation is executed by pointing the cursor at a desired area of the screen, and clicking to indicate the selection of the item the cursor is pointing to.

For people with motor disabilities there are several devices that can be used as an alternative for the mouse. For example, the "Tonguepoint" [Salem and Zhai, 1997] as the name suggests, is a device in which the user can operate the cursor using his/her tongue.

Results for users who have tested this system revealed that they were able to achieve a performance level that was only 5-50% slower than users who used a standard pointing device.

Another alternative device is the “Headmouse”, which is a pointing device that moves the cursor based on head movements [Origin Instruments Corporation, 2016]. This is similar to Emotiv’s Neuro Mousecontrol application that allows the user to control the cursor using head movements while he/she is wearing the Epoc headset [Emotiv, 2016]. However, whereas the Neuro Mousecontrol application depends on cognitive action to trigger left and right mouse button operations, the Headmouse device achieves these actions when the user dwells over a particular key on an on-screen keyboard for a set period of time, or by using a remote switch. There are also more complex approaches for computer interface functionality such as eye-tracking systems. These systems only depend on the user’s ability to control the gaze of his/her eye. The cursor follows the eye’s line of gaze, and the clicking operation is executed whenever the cursor remains in a location for a certain amount of time, which is similar to the Headmouse device.

Another complex approach is the Brain-Computer Interface that uses EEG signals, as further described in Chapter 3. However, Andrew Junker developed a device called Cyberlink that allows the user to control the cursor using both EMG (produced by muscle movements from the head and neck) and EEG signals (produced by thoughts) [Junker et al, 2001]. Unfortunately, this method of interaction hinders the overall control process of the cursor because intensive concentration is required, which may be difficult for users.

Barreto et al proposed a system that only utilizes EMG signals for cursor control [Barreto et al, 2000]. In their approach, cursor movement depended on the natural and voluntary movements of the user’s face such as clenching different sides of the jaw (left, right, and both) and eyebrow movement (up and down). These types of movements are usually controllable for people who are impaired from the neck down. The system was tested by six subjects that participated in 20 trials. In each trial, subjects were asked to click on the Stop button by moving the cursor from one corner on the computer screen to the center, where the button is located. Results revealed that the subjects required an average of 16 seconds to complete their desired tasks [Barreto et al, 2000]. Though this time is

notably greater compared to the amount of time it would take for an unimpaired subject to accomplish the same task using a mouse, the response time would still be usable with most standard GUIs. In addition, this proposed interface may potentially be more affordable and portable than other complex interfaces like the eye-tracking system.

4.4 Issues/Limitations

Visual Impairments

In the case of auditory interfaces, one issue is its scalability as the complexity increases. In the Towers of Hanoi game, only five discs were used and users seemed to perform relatively well. However, Towers of Hanoi had rule restriction that were able to offer some guidance for the users. In a desktop interface, in which there will inevitably be a larger number of objects, will blind users benefit from auditory direct manipulation? As demonstrated by Donker's system, as the amount of hearcons increased in the interface, the user was not able to efficiently and effectively process their tasks.

Another issue in auditory interfaces is representing the selection state in a GUI. The selection state for when users execute selection operations such as highlighting text or selecting objects to drag and drop needs to be represented. In the game, Towers of Hanoi, users were not able to recognize whether or not they were actually moving the disc to the next tower instead of just hovering the cursor over each tower. Furthermore, if large numbers of interaction objects are present, the user may be limited to his/her memory. To forget the location of an object would be time consuming for the user to constantly check each item present one by one for the correct object. Another issue is using various sounds could cause the user to forget the meaning of each sound during the beginning stages of using the system. This means that auditory interfaces may be more difficult to learn for users.

One major concern that arises in Andronico et al's modified Google UI is that it had not been tested. Without any results, we cannot evaluate the proposed guidelines for designing a user interface for people with visual impairments. Furthermore, this user interface depended on code for the Google search engine. It was mentioned that other existing search engines were less than qualified for reengineering, which means that future

designers may be restricted to using Google UIs if they do not wish to develop their own search engine from scratch.

Motor Impairments

For UIs that cater specifically to people with motor impairments, there may be some possible disadvantages depending on the category of motor disability. For instance, a user with cerebral palsy may not be able to operate the Tonguepoint device due to lack of motor abilities in the tongue. Similarly, a user with spinal fusion may not be able to move his/her head, thus unable to fully operate any of the mentioned BCI systems. However, for those who are capable of operating a BCI system, one limitation is the reaction speed of the cursor. In addition, it takes a considerable amount of concentration to train and execute commands. If the user becomes distracted while training the system, it will be difficult to accurately perform a desired task. Furthermore, it is difficult for one to measure the effectiveness of a BCI system against a mouse when executing the same tasks. This is because Fitts' law is unable to describe the performance of people with disabilities [Pino et al, 2003].

Two major disadvantages of eye tracking systems are that they are significantly expensive and require a considerable amount of focus to control. If the user loses focus and unintentionally stares into one location on the UI, he/she may accidentally click the cursor. Another limitation is that the calibration of the device can be easily lost if the user changes position with respect to the screen [Barreto et al, 2000]. Furthermore, gazing at a location that is out of the camera's field of vision will hinder the system's operation.

4.5 Summary

As technology continues to advance, its usability starts to grow in complexity. Equal accessibility for people with disabilities then becomes compromised. This chapter explored the various user interfaces that allow people with visual and motor impairments to keep up with today's technology-dependent society. For people with low vision, the concept of an auditory interface was introduced. However, further research is required to determine whether or not this form of interaction can hold its own against the standard direct manipulation on a GUI. This chapter also introduced a set of guidelines for

improving search engines for the visually impaired were also proposed. However, there were no test results that proved the validity and effectiveness of this criteria. A system was also introduced in this Chapter that could automatically generate a UI that could adapt to one's visual and motor capabilities. For users with motor impairments, various systems were briefly mentioned. However, the most promising systems were the eye tracking system and the BCI system.

5 STUDY 1: SIM-ASSIST

This chapter introduces and explores our system, Sim-Assist. The design of our system is outlined in detail. This chapter also presents a walkthrough of the prototype of Sim-Assist and provides some results. The content in this chapter is based on our previously published paper at ITNG-2016 [Chan et al, 2016], which received the best student paper award at the conference.

5.1 Introduction

Recently, assistance for people who are care-dependent has gained importance as a research field because it could potentially improve the quality of their lives as well as the lives of their caretakers. Gomez et al presented and evaluated a system that presents interactive and personal manuals for daily life activities for people with disabilities [Gomez, 2013]. Additionally, Donoghue et al (2007) also aimed to create a system for people with motor disabilities. Specifically, they developed a human application of neural interface systems (NICs) for people with paralysis. Blindness and vision impairment also increases rapidly as people grow older, particularly after the age of 75 [American Foundation for the Blind, 2015]. Narasimhan et al developed a system to provide visually impaired people with an independent shopping experience [Narasimhan et al, 2003].

Software-based assistive technology (SBAT) research is the main motivation for this chapter. Our system, Sim-Assist, aims to integrate SBATs into a 3D simulation. The main goal of Sim-Assist is to allow people with disabilities to play the 3D virtual game Air Hockey. In the near future, we plan to use Sim-Assist as a foundation to integrate assistive technologies (AT) into 3D scientific simulations.

5.2 High-level and Medium-level Design

Our system uses a 3D version of the Air Hockey game, implemented by Alex Gamino. Sim-Assist is innovative because it allows people to play an engaging game such as Air Hockey with reduced effort. In addition, there are no existing commercially available video games that incorporate brain-computer interface (BCI) technology.

The main functionalities of Sim-Assist include: the ability to move the Air Hockey paddle using BCI capabilities; a notification for each new score; and the ability to input player information, restart, and pause and resume the game using speech. The implemented functional requirements of the Sim-Assist system are shown in Table 5.1. The intended users of this system are people with visual and motor impairments, or both. However, people without disabilities may also use Sim-Assist to experience a new form of gaming (BCI-based).

Sim-Assist was developed in C++ on a Microsoft Windows platform, using Microsoft Visual Studio 2012. To play the Air Hockey game, Sim-Assist requires the Emotiv EPOC headset for all BCI-related user inputs [Emotiv, 2016]. To record the user's voice for speech commands and speech-to-text capabilities, the Microsoft Kinect peripheral was used, with a dependency on the Microsoft Speech Platform.

Table 5.1. Functional requirements for Sim-Assist

Functional Requirements
<p>Sim-Assist shall:</p> <ul style="list-style-type: none"> ● Display a 3D simulation game of Air Hockey. ● Allow the user to control the mouse using head movements. ● Grant the user the ability to control the mouse using various thoughts. ● Permit the user to use voice commands to pause/un-pause the game using speech. ● Notify when user scores a goal. ● Enable the user to toggle voice detection. ● Allow the user to input player information using speech. ● Present the option to select number of players. ● Provide the capability to quit the game using speech. ● Enable the user to enter/exit full screen mode using speech. ● Authorize the user to assign facial expressions specific to mouse control. ● Permit the user to assign thoughts to specific mouse control. ● Accept multiple users to play at one time.

The Sim-Assist system, from a high-level perspective, consists of different layers in its architecture, as shown in Figure 5.1.

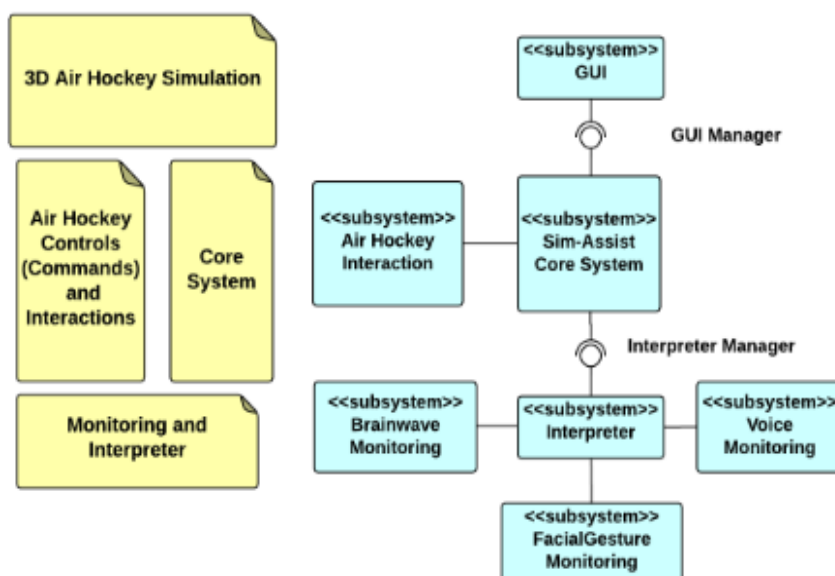


Figure 5.1: System-level diagram of the Sim-Assist system

The lowest layer is the monitoring and interpreter, where Sim-Assist detects speech, facial and head gestures, and brainwaves. This is accomplished by the brainwave, facial gesture, and voice monitoring subsystems. The interpreter subsystem then takes these possible commands and determines if they are actual commands. If so, they are sent to the Sim-Assist core system. This system is responsible for all the major decisions within the system, including holding registered commands, as well as managing GUI information. The top layer of Sim-Assist is the Air Hockey game, which consists of the GUI subsystem that communicates with the Sim-Assist core system.

Program Units

This section presents a diagram of the structure and organization of the Sim-Assist software, as shown in Figure 5.2. Each module, function, and procedure is separated into program units, some of which are further described in the following section.

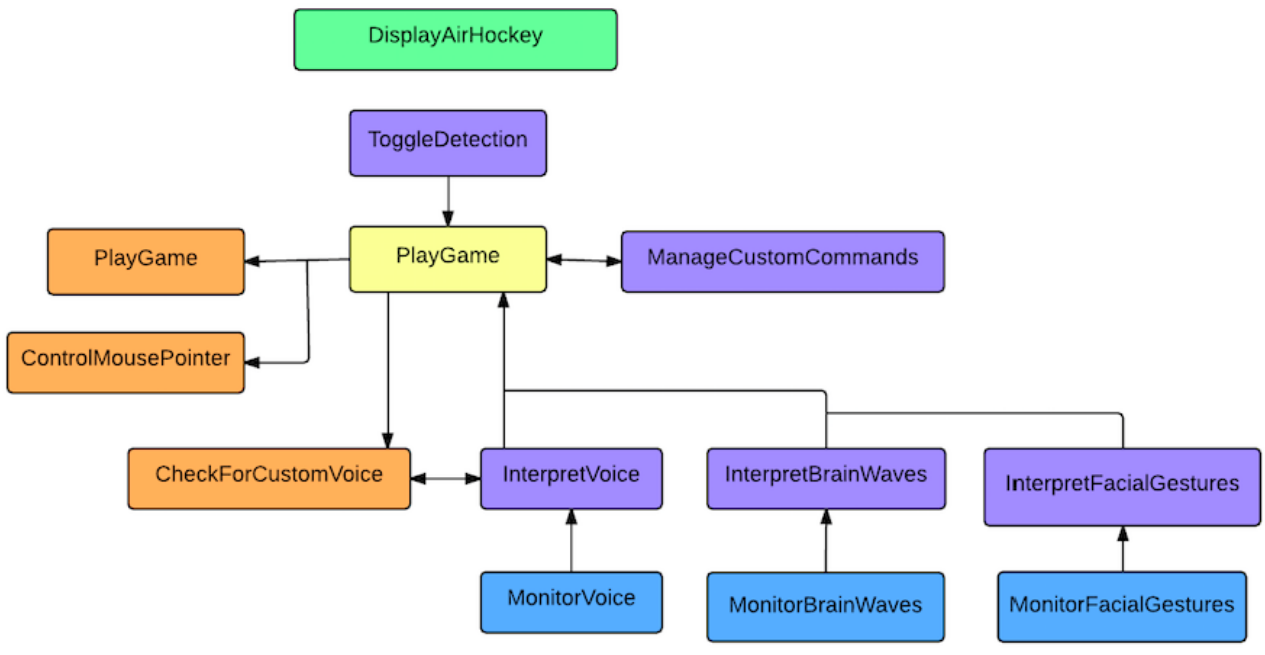


Figure 5.2: Program units for Sim-Assist system

Figure 5.3 - Figure 5.8 presents the descriptions and further details of the main program units that are significant to the Sim-Assist software. For each program unit, an overview of its functionality is provided, followed by the higher-level unit it belongs to, its input and output, and other program units called by this unit.

Name	PlayGame
Subsystem	Air Hockey Interaction
Input	-Game to be launched -Voice command/other user input in GUI
Output	Air Hockey Game
Other Program Units Called	-MonitorVoice/Interpret Voice -MonitorBrainWaves/Interpret Brainwaves -MonitorFacialGestures/FacialGestures
Exceptions and Interrupts	-Game may not exist on computer -Cannot recognize brain signals or voice
Description	Starts the game. The player will be prompted to start the game by clicking a button or executing a voice command.

Figure 5.3: Description for PlayGame program unit

Name	Toggle Detection
Subsystem	GUI
Input	User specifies detection on/off
Output	Boolean: Execute/Ignore commands
Other Program Units Called	None
Exceptions and Interrupts	None
Description	If the user specifies for no detection, a Boolean is used to tell Sim-Assist to ignore voice and facial/brainwave commands. This module is called again if user wishes to turn detection back on.

Figure 5.4: Description for ToggleDetection program unit

Name	InterpretFacialGestures
Subsystem	Interpreter
Input	Monitor data from MonitorFacialGestures
Output	Facial gesture type data to interpreter
Other Program Units Called	MonitorFacialGestures
Exceptions and Interrupts	None
Description	Interprets facial gesture data from MonitorFacialGestures. If a facial gesture is made, the type of gesture is sent to the interpreter for action execution.

Figure 5.5: Description for InterpretFacialGestures program unit

Name	InterpretBrainWaves
Subsystem	Interpreter
Input	Monitor data from MonitorBrainWaves
Output	Brain wave type data to interpreter
Other Program Units Called	MonitorBrainWaves
Exceptions and Interrupts	None
Description	Interprets brain wave data from MonitorBrainWaves. If a recognized thought is made, the type of thought is sent to the interpreter for action execution.

Figure 5.6: Description for InterpretBrainWaves program unit

Name	ControlMousePointer
Subsystem	Sim-Assist Interaction
Input	Brainwave/Facial expression commands from Interpreter
Output	None
Other Program Units Called	None
Exceptions and Interrupts	Epoc headset might not be able to recognize user's thoughts/facial expressions
Description	ControlMousePointer accepts input from the Interpreter to control the mouse pointer as well as its buttons.

Figure 5.7: Description for ControlMousePointer program unit

Name	CheckForCustomVoice
Subsystem	Interpreter
Input	-Voice command -Custom voice commands list
Output	Boolean: Voice command action found
Other Program Units Called	None
Exceptions and Interrupts	None
Description	CheckForCustomVoice is called when InterpretVoice encounters a probable voice command that is not a hardcoded command.

Figure 5.8: Description for CheckForCustomVoice program unit

Data Design

This section offers the details of the data structures used by Sim-Assist.

Static Array of Preset Facial Gestures

- One-dimensional static array of gesture information
- Array holds information on pre-programmed gestures for detection
 - Facial gesture information
 - Possible head movement information
- Gestures programmed approved by Sim-Assist developer
- Users cannot enter new gestures for detection (static array)

Static Array of Preset BrainWave commands

- One-dimensional static array of brain wave information
- Array holds information on pre-programmed brain wave commands for detection
 - User's brain wave information
- BCI commands programmed approved by Sim-Assist developer
- Users cannot enter new gestures for detection (static array)

Static Array of Built-in Voice Commands

- One-dimensional static array of voice commands information
- Array holds information on pre-programmed gestures for detection
 - Phase to be spoken
 - Command executed by phrase
 - "Confidence Level" that must be reached to execute command
- Built-in voice commands approved by Sim-Assist developer.

Dynamically Allocated List for Custom Voice Commands

- Dynamic list that holds custom voice commands
- Information for user-specified voice commands
 - Custom phrase entered by user
 - Command to be executed by user (open web page, launch program, etc.)

Dynamically Allocated List for Custom Gesture Execution

- Dynamic list that holds custom commands from preset gestures
- Information for user-specified gesture commands
 - Preset gesture to begin executing
 - Command to be executed

5.3 Detailed Design

This section presents the workflow of the Sim-Assist system, followed by activity diagrams that offer a glimpse into the logic behind two of Sim-Assist’s functionality.

Workflow Diagram

Figure 5.9 illustrates a high-level depiction of the workflow diagram in Sim-Assist.

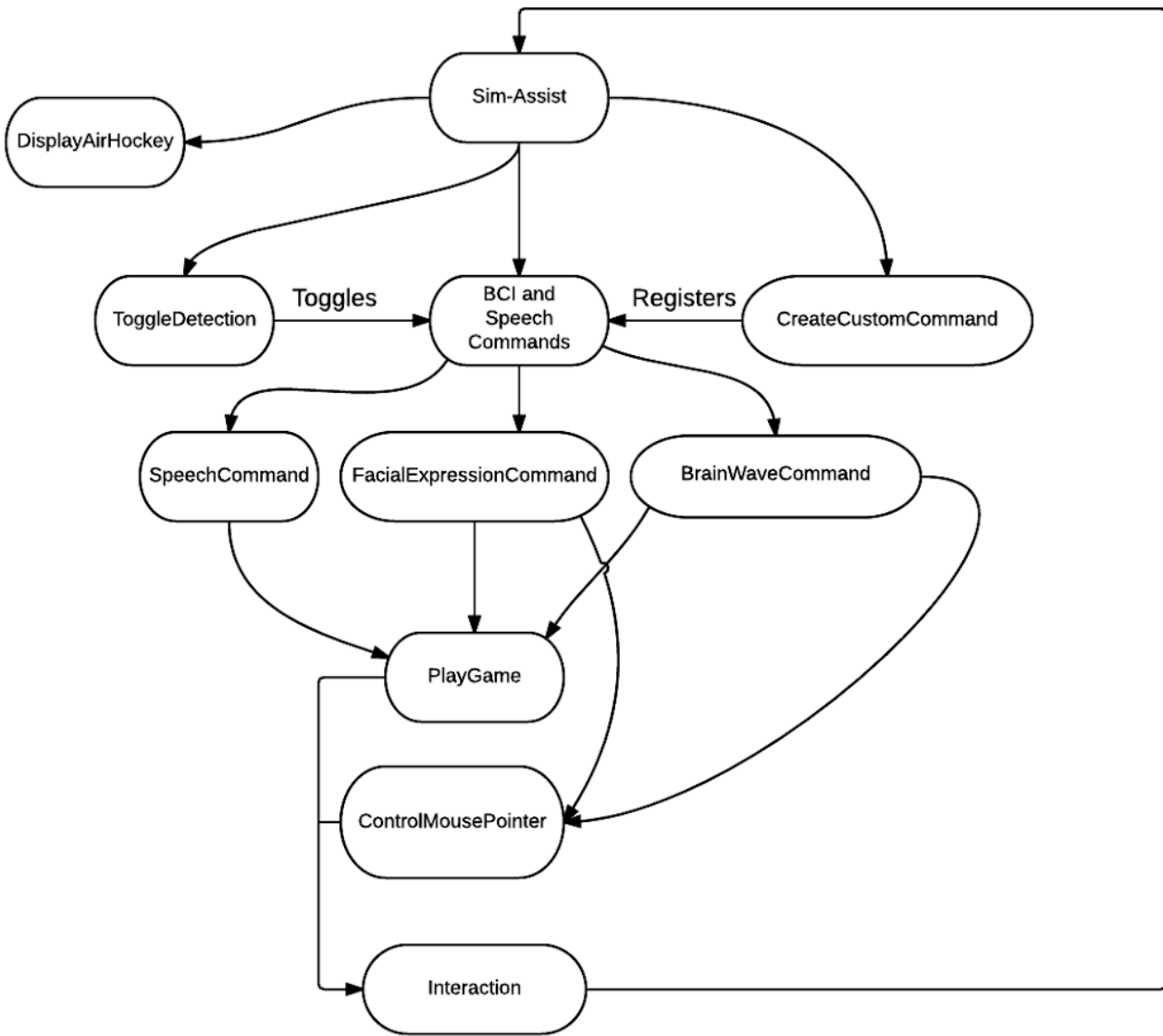


Figure 5.9: High-level activity workflow diagram of Sim-Assist

Activity Diagrams

Figure 5.10 presents the activity diagram describing the detection, interpretation, and execution of the facial expression command. Facial expressions are one of the most important features in Sim-Assist as it simulates mouse clicks, launching the game paddle within Air Hockey.

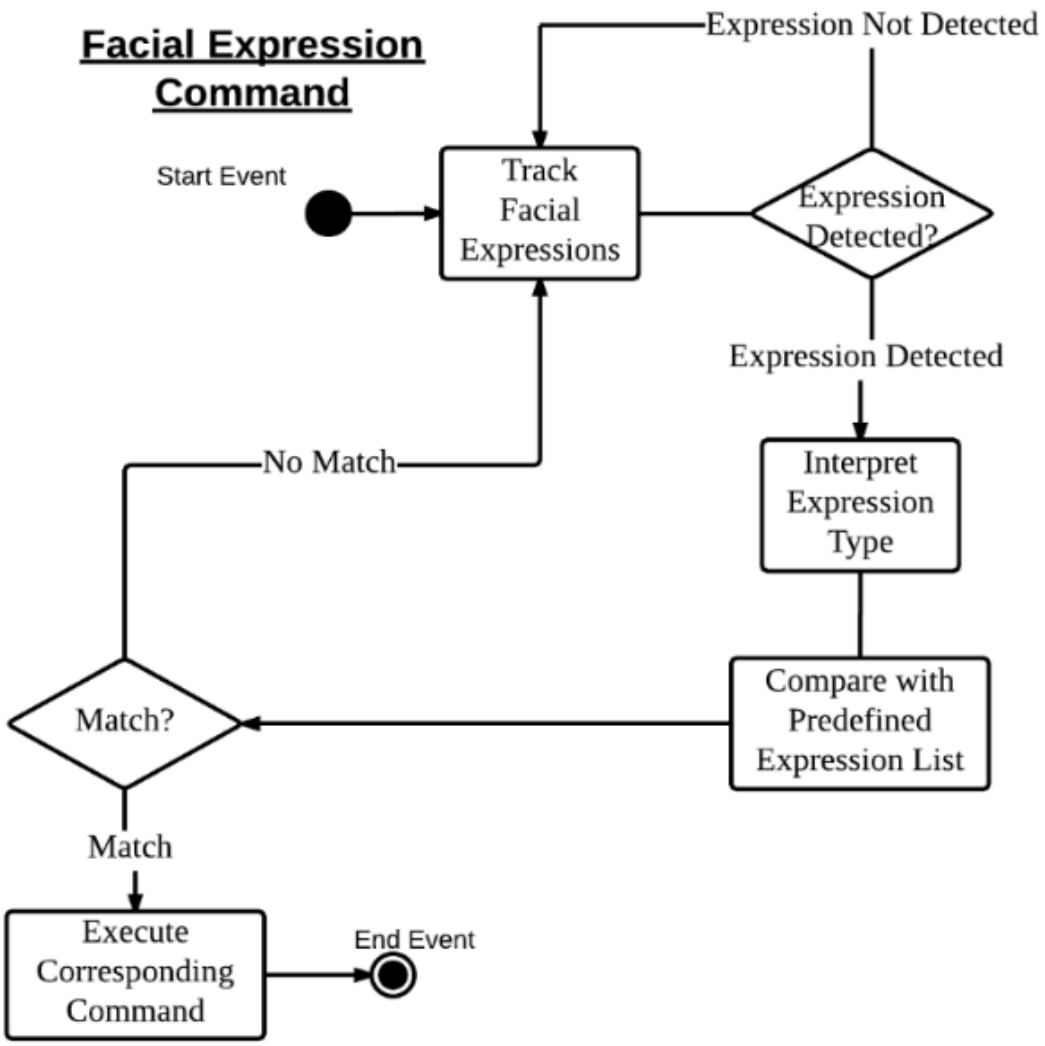


Figure 5.10: Activity Diagram for “Facial Expression Command”

Figure 5.11 describes the detection, interpretation, and command execution of BCI commands used in Sim-Assist. The BCI controls are mapped to the mouse cursor, which is the main method to interact with the system.

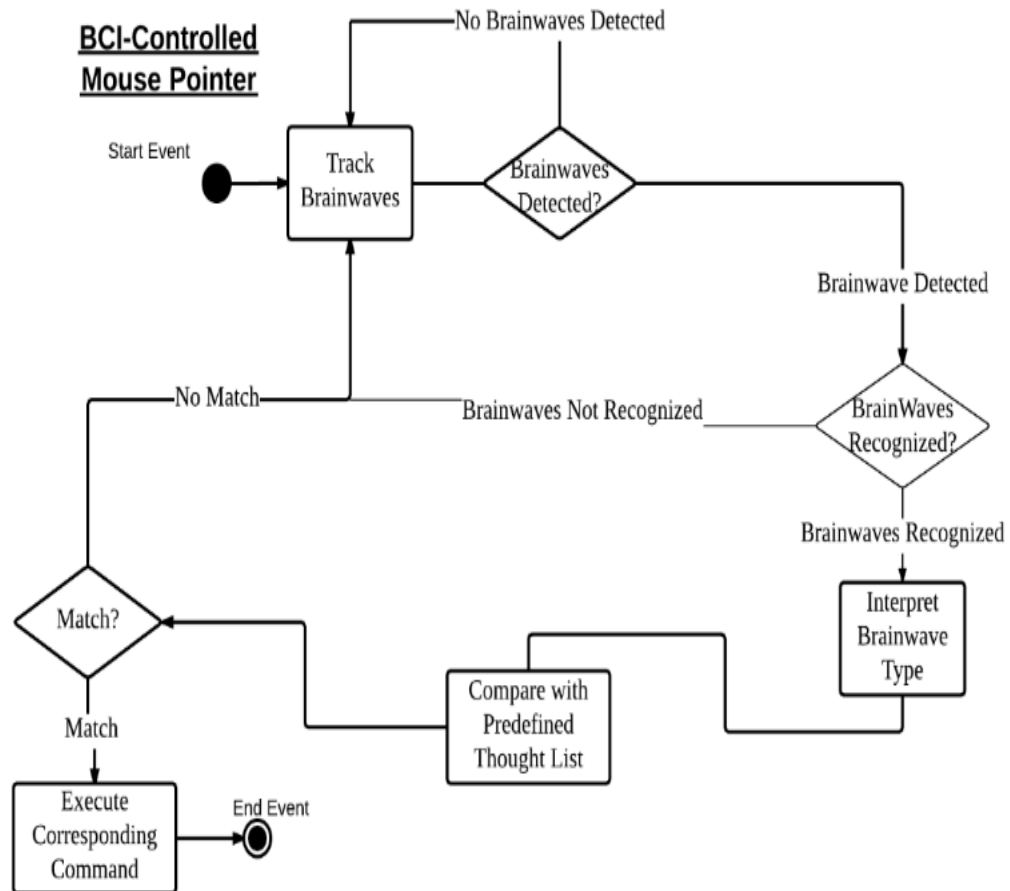


Figure 5.11: Activity Diagram for “BCI-Controlled Mouse Pointer”

5.4 Prototype Walkthrough

This section presents a brief walkthrough of a prototype of Sim-Assist. Upon startup, Sim-Assist will greet the user with a welcome window, as shown in Figure 5.12.



Figure 5.12: Sim-Assist welcome window

The user must enter his or her information prior to playing Air Hockey, as shown in Figure 5.13. The system will prompt the user to enter his or her name verbally. A database of approximately 200 popular names (both genders) in the U.S. is used to help the system recognize the user's name. The user also has the option to select how many players will participate in the game (one or two).

Once player information is submitted, the user must click on the "BCI" button from the welcome window. Once the training session is completed, the user can map trained commands to the mouse control. This will allow the user to move the mouse, along with playing the game, using his or her thoughts.



Figure 5.13: Player information window. Here, Sim-Assist will assign a default avatar (koala) to the player.

Finally, the user can launch the game by clicking on the "Let's Start!" button. The user is presented with an Air Hockey table, along with a scoreboard, two paddles, and a puck. The user's verbally submitted player information will also be displayed onto the game board, as shown in Figure 5.14.



Figure 5.14: Snapshot of the Air Hockey game

It is important to note that our settings were that Player 1 could control the paddle using only his or her thoughts and facial expressions, while Player 2 can control the paddle only with keyboard presses. Either player could also: enter or exit full screen mode, as well as pause, resume, or restart the game using speech commands.

5.5 Results

One limitation of our system was that Player 1 could use only BCI controls, while Player 2 could use only keyboard controls. Upon system testing, we realized that the Air Hockey game is too difficult to play with BCI controls alone. While playing the game, Player 1 was under too much stress and had difficulties executing certain thoughts accurately. Since Player 2 could control his or her paddle using only keyboard presses, this served as an unfair advantage. Consequently, in our future versions, we plan to fix the current player settings of the game.

One significant constraint in our system was that in order for Player 1 to use the Epoc headset, he or she needed to have little or no hair, as shown in Figure 5.15. Otherwise, the sensors on the headset were unable to detect any signals.

Because the Epoc was used, an advantage of SimAssist is that there are many different types of thoughts the user can use to map to a specific mouse control. To compare, systems that use NeuroSky's "MindWave" headset will operate only if the user is in a

focused or relaxed state [NeuroSky, 2016]. Using this headset may be too restrictive for developers to create a robust system.



Figure 5.15: Player using the Emotiv Epoc’s BCI controls to interact with the 3D Air Hockey game.

5.6 Summary

This chapter introduced our system, Sim-Assist, which allows people with disabilities to interface with a three-dimensional simulation game of Air Hockey. Our system utilizes brain-computer interfacing, voice commands, and speech-to-text capabilities, which targets people with visual and/or motor impairments. Sim-Assist serves as a foundation to integrate assistive technologies into a larger variety of 3D simulations.

6 STUDY 2: GOOGLE STREET VIEW APPLICATION

This chapter introduces and explores the integration of BCI technologies with navigation applications. For this study, we designed a BCI web application that is for the Google Street View™ mapping service, which is outlined in detail. This chapter also presents some software testing of our prototype along with results. This chapter's contents will be used in a conference paper, currently in preparation [Chan et al, 2017a].

6.1 Introduction

In May 2007, Google launched the Google Street View feature of Google Maps™ mapping service [Vincent, 2007]. The main goal of Google Street View is to provide people with an interface that can display images of any location at the street level in such a way that enables navigation between images without having to compromise the map context. Other related projects have been found that pre-dates the introduction of the Street View feature. The earliest related project we discovered was launched in the late 1970s with a system called, “Movie Maps” [Lippman, 1980]. Movie Maps allowed users to take virtual tours of the city of Aspen, Colorado. This system's data-capture utilized a gyroscope stabilizer and four cameras that were attached on top of a car. For every ten feet driven by the car, the mounted cameras were triggered to take a picture of the scene. The panoramic imagery taken from the cameras were stitched together and organized into numerous scenes, each encompassing approximately a city block, and stored on a laser disk. A user could virtually navigate the streets of Aspen using the interface (Figure 6.1).



Figure 6.1: Illustration of navigation through Movie Maps Interface [Lippman, 1980]

Following the Movie Maps system, several other related systems emerged. Companies such as Yahoo, Microsoft, and Amazon have also launched geospatial viewing tools that are linked to text-based searches. Similar to Google, they are free, fast, easy to use, and abundant in global data. However, unlike Google, Yahoo's Yahoo Maps, Microsoft's MSN Virtual Earth, and Amazon's A9 do not offer a free three-dimensional global desktop map viewer. These tools all offer contextual map data from various providers, such as NAVTEQ and TeleAtlas, and sometimes in combination [Lippman, 1980]. Google and MSN also offer aerial imagery with or without annotation, and Amazon offers what it calls "BlockView" images of the landscape (typically building frontages) on both sides of the selected road segment. In addition to combining Internet search results with graphic maps, each site has its own extras, such as Yahoo Maps' real-time traffic mapping option showing vehicle flow at roadway sensor locations. Functionally, the online tools are simple viewers designed to plot points on contextual maps. Their technical innovation is in the design of their interfaces, which behave more like desktop applications than online tools. Pans —accomplished by clicking and dragging a map — reveal new map graphics seemingly without a browser refresh.

A unique feature of Street View is the merging of street-level imagery with maps in a natural manner [Vincent, 2007]. In addition, Street View also ties the imagery to the map in a way that makes navigation easier from the map or imagery itself. This is achieved through the "pegman" Street View avatar. This avatar rotates and moves along with the map even when navigating from the panoramic image itself. Due to Street View's seamless merge of panoramic images with their corresponding locations on the map, we use Street View as our target for our nearly hands-free navigation application.

As reported in Chapter 4, voice recognition software is the main solution to web surfing and GPS navigation for people who possess a motor or visual disability. However, voice recognition is unable to cater to their limited capabilities in an application that integrates both web searching and GPS navigation. When using an application such as Google Maps, direct manipulation from the user is needed. Although the user can simply input location searches using voice, it is difficult to interact with the map itself without

using the cursor. We propose an alternative method by which people with motor impairments can interact with such a direct manipulation-dependent application.

6.2 Application Overview

Our web application mainly consists of a side-by-side view of Google Maps and Google Street View. With this application, users can navigate through Google Maps or Google Street View in a virtually hands-free manner. The map displayed on the web application is defaulted to Boston, Massachusetts. The user can search for a specific location using the search toolbox on the GUI by keyboard. With the exception of expressive actions (i.e., any form of facial expressions), the user must undergo a training session in order to navigate through Google Street View through his/her cognitive actions (i.e., the user's thoughts).

Training Session

As shown in Figure 6.2, the user can train our application to recognize up to four distinct navigation controls using cognitive actions. Although expressive actions do not need to be trained, users must go through this training session to map four separate camera controls to a different expressive action. We note that the list of cognitive actions shown in Figure 6.2 does not reflect the complete list of built-in cognitive actions in from which the user can select. We composed a list of basic cognitive actions which the older version of Epoc will also recognize. This allows for some flexibility in our BCI headset. Our list of cognitive actions is ordered according to the level of difficulty for a user to execute.

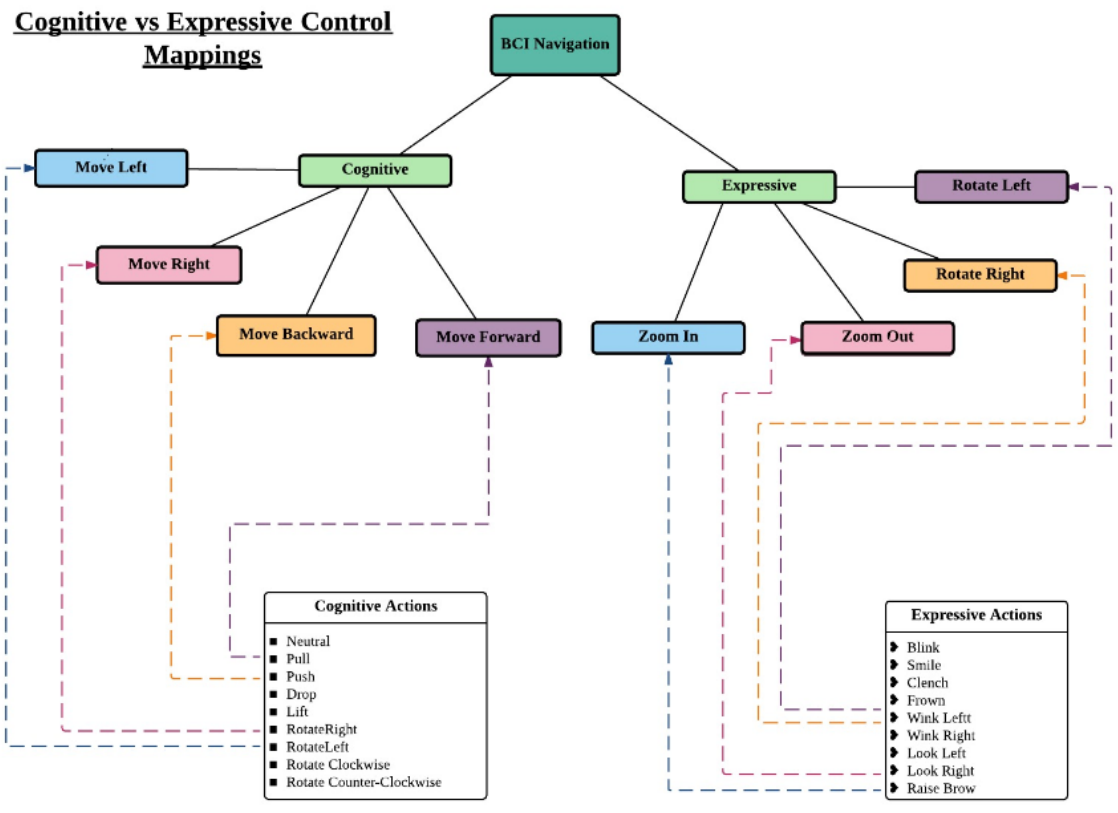


Figure 6.2: Control mappings for navigations within web applications

The difficulty levels in executing each cognitive action increases for every newly trained action (See Figure 6.3). Although we strongly advise our users to map cognitive actions to navigation controls that are intuitive (e.g., RotateLeft → move left), the desired cognitive action may be too high in difficulty for a novice BCI user, thus affecting performance. We will demonstrate some differences in performance later in this chapter.

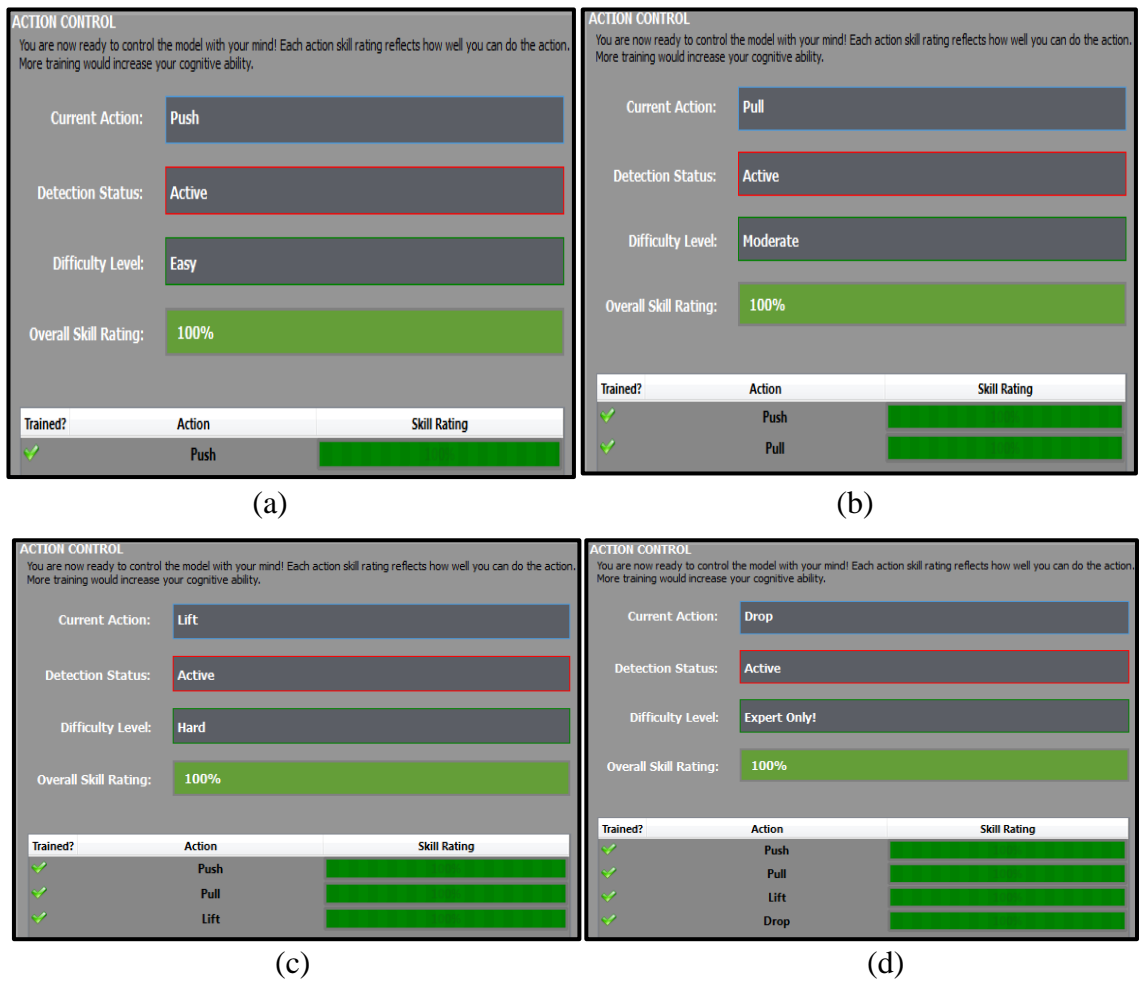


Figure 6.3: Screenshot of Emotiv's training software [Emotiv, 2016]. Figure 6.3 (a)-(d) depict an increase in difficulty levels for each trained cognitive action.

Users are also limited to train four cognitive actions at a time. This is a built-in restriction with the Epoc headset. For this reason, we must also utilize expressive actions as our user input to expand user capabilities within our application. To train a cognitive action, the user must hold that action for eight consecutive seconds. In other words, the user must constantly think the specific thought until the trial is complete.

6.3 System Design

Figure 6.4 presents an activity diagram of our BCI controller within our Street View application. The user's brainwaves are monitored, interpreted, and compared to a predefined action list. If the input brain signal matches with the action list, the corresponding command is executed.

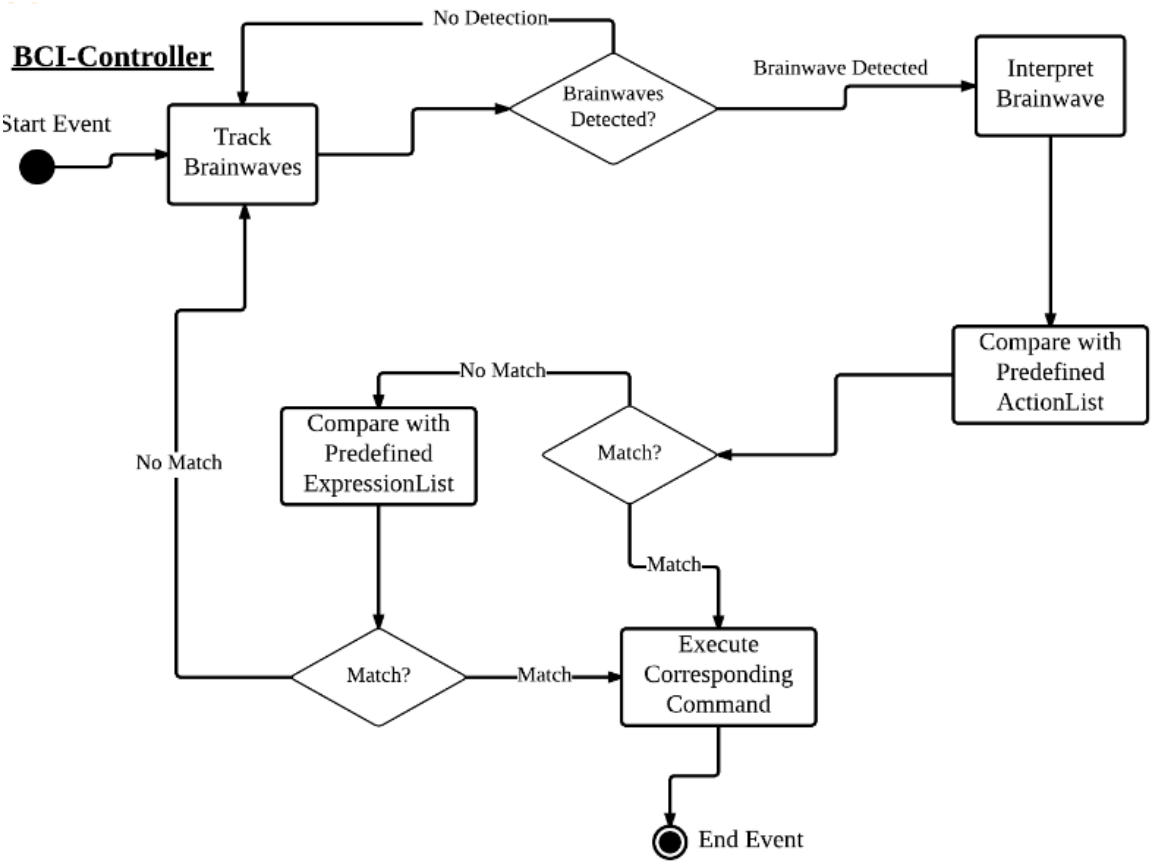


Figure 6.4: Activity diagram of BCI controller within web application

Figure 6.5 presents the structure and organization of our system. Each module, function, and procedure is separated into program units.

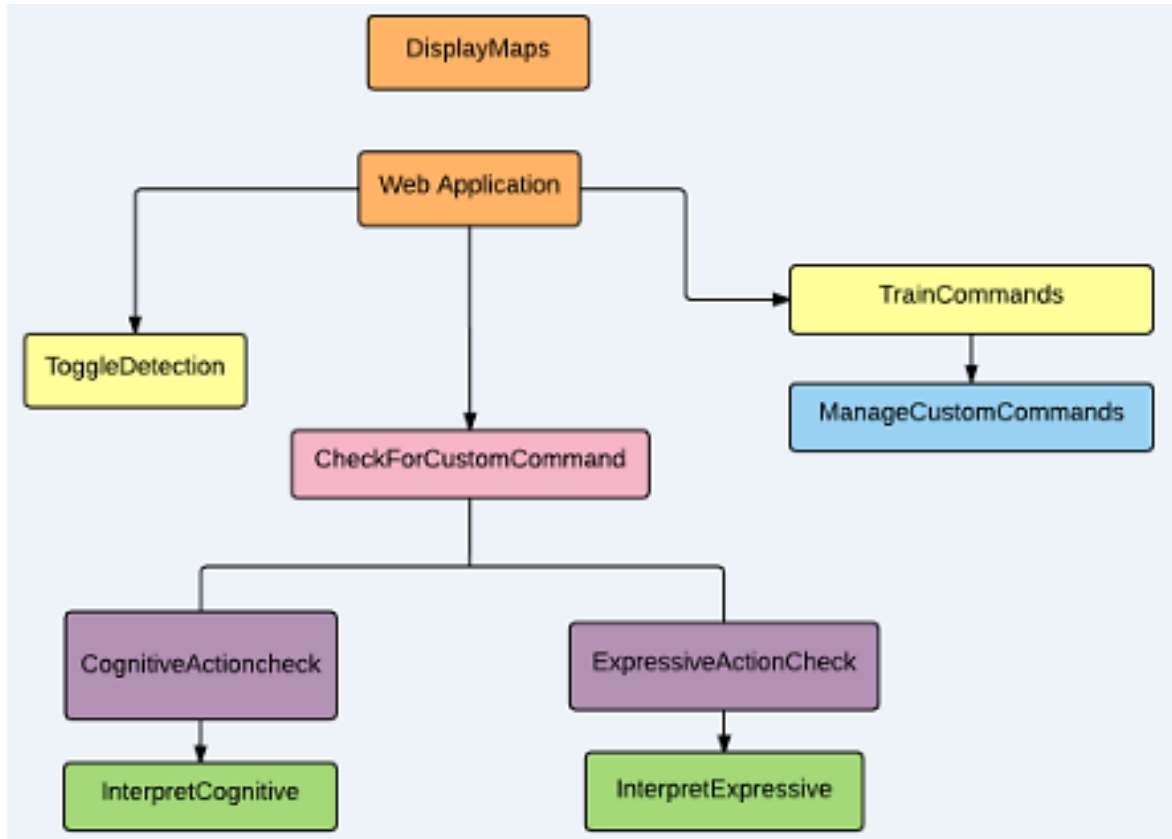


Figure 6.5: Street View Application

Figure 6.6 – Figure 6.13 presents the descriptions and further details of the main program units that are significant to our Street View application. For each program unit, an overview of its functionality is provided, followed by the higher-level unit to which it belongs, its input and output, and other program units called by this unit.

Name	WebApplication
Subsystem	Google Street View Interaction
Input	<ul style="list-style-type: none"> -Cognitive Action (Trained) -Expressive Action (Trained) -Voice Command -Keyboard Search Input in GUI
Output	-Rendered display of Google Maps and Google Street View
Other Program Units Called	<ul style="list-style-type: none"> -Display Maps -TrainCommands -ManageCustomCommand -CheckForCustomCommand
Exceptions and Interrupts	-Cannot recognize voice or cognitive/expressive actions
Description	<p>Starts and executes main application. The user can either manually insert search via keyboard or via voice commands. Navigation in Street View is performed via cognitive and expressive actions. Any position displacement in Street View will be reflected in Google Maps as well.</p>

Figure 6.6: Description for WebApplication program unit

Name	ToggleDetection
Subsystem	GUI
Input	-User specifies detection on/off for either voice or BCI
Output	-Boolean: Execute/Ignore commands
Other Program Units Called	None
Exceptions and Interrupts	None
Description	If the user specifies no detection, a Boolean will be used to tell the web application to ignore voice and BCI commands. This module will be recalled if user wishes to turn detection back on.

Figure 6.7: Description for ToggleDetection program unit

Name	CognitiveActionCheck
Subsystem	Interpreter
Input	EEG data from CheckCustomCommand
Output	-Boolean: Cognitive action found/not found -Cognitive data for cognitive action interpreter if found.
Other Program Units Called	-InterpretCognitive -CheckCustomCommand
Exceptions and Interrupts	None
Description	Recognizes EEG data as a cognitive action. If recognized, data will be sent to InterpretCognitive for further interpretation.

Figure 6.8: Description for CognitiveActionCheck program unit

Name	ExpressiveActionCheck
Subsystem	Interpreter
Input	EEG data from CheckCustomCommand
Output	-Boolean: Expressive action found/not found -Expressive data for expressive action interpreter if found.
Other Program Units Called	-InterpretExpressive -CheckCustomCommand
Exceptions and Interrupts	None
Description	Recognizes EEG data as an affective/expressive action. If recognized, data will be sent to InterpretExpressive

Figure 6.9: Description for ExpressiveActionCheck program unit

Name	InterpretCognitive
Subsystem	Interpreter
Input	-EEG data from CognitiveActionCheck -Custom command list
Output	-Boolean: Cognitive action recognized/not recognized -Execution of custom command if found
Other Program Units Called	-CognitiveActionCheck -ManageCustomCommand
Exceptions and Interrupts	None
Description	Interprets EEG data as a cognitive action. If recognized, command mapped to cognitive action will be executed.

Figure 6.10: Description for InterpretCognitive program unit

Name	InterpretExpressive
Subsystem	Interpreter
Input	-EEG data from ExpressiveActionCheck -Custom command list
Output	-Boolean: Expressive action recognized/not recognized -Execution of custom command if found
Other Program Units Called	-ExpressiveActionCheck -ManageCustomCommand
Exceptions and Interrupts	None
Description	Interprets EEG data as an expressive/affective action. If recognized as a custom command, command mapped to expressive action will be executed.

Figure 6.11: Description for InterpretExpressive program unit

Name	TrainCommands
Subsystem	Core System
Input	EEG data for training
Output	List of custom commands
Other Program Units Called	ManageCustomCommand
Exceptions and Interrupts	-Headset must be connected. -Voice commands do not need to be trained.
Description	User can train up to four cognitive actions and map them to custom commands. User can also map expressive actions to four additional commands. Trained and mapped commands will be sent to the ManageCustomCommand module.

Figure 6.12: Description for TrainCommands program unit

Name	ManageCustomCommands
Subsystem	Core system
Input	-Trained custom commands from TrainCommands module
Output	Custom command list
Other Program Units Called	TrainCommands
Exceptions and Interrupts	None
Description	Keeps a record of trained commands and its control mappings. This record will be used by the interpreter to recognize any incoming cognitive and expressive actions as valid input.

Figure 6.13: Description for ManageCustomCommands program unit

6.4 Software Testing

For our software testing session, we informally asked several colleagues in the computer science department to provide feedback as software developers. We plan for usability testing in the future, which will be performed in part for a paper soon to be submitted to an international conference [Chan, 2017a].

Apparatus

For our test, we used an Emotiv Epoc+ controller, which was connected to the PC. The PC had a 2.33 GHz CPU, with 8gb RAM, running 64 bit Windows 7. In addition, the appropriated shared run-time libraries (such as EDK.dll) were installed using Emotiv's SDK installer. Our web site was created in ASP.Net with Microsoft Visual Studio 2012. Our testing facility, a quiet office with minimal distractions, remained constant.

Software

For this test, we used Emotiv's control panel software to ensure good contact quality of the sensors in our Epoc+ controller (Figure 6.14). The software was left open to monitor the contact quality of each sensor throughout use of the application. Expressive states were also monitored on the Expressiv tab of the control panel. This section of the software indicated whether any of the users were making any facial movements. Testers

were reminded to keep their face relaxed during training and during some test trials that cognitive action needed to be executed.

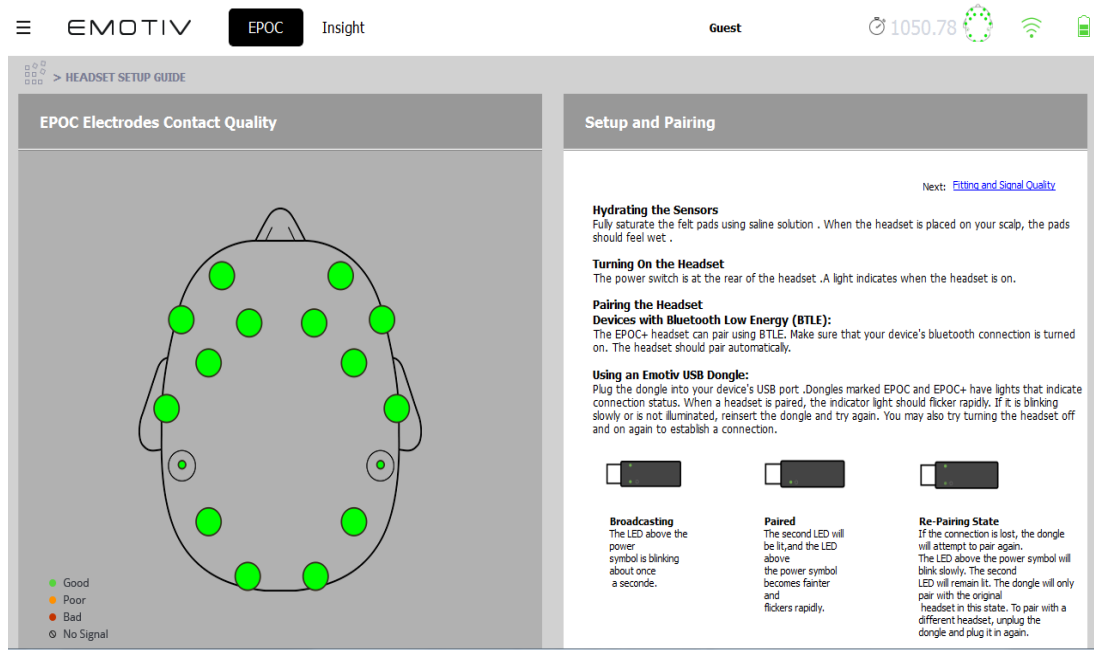


Figure 6.14: Snapshot of Emotiv's Control Panel's Contact Quality window [Emotiv, 2016]

To control the experiment and log our data, a program was written in C# that uses the EmoEngine through the proprietary API. Emotiv's EmoEngine is composed of a logical abstraction of the functionality that Emotiv provides in the run-time library. This engine communicates with the EPOC headset, receives preprocessed EEG data, manages settings, performs post-processing, and translates the detection results in a data structure called an EmoState. After each cognitive action has been trained, our program can read EmoStates from the EmoEngine. This consists of the current action being thought about and its power. The power of a state refers to the system's confidence that the user's thinking is consistent with what they had trained. Each state is both recorded and sent to our Street View application.

Testing Session

In the training phase, the system presents the users with a cube (Figure 6.15). All new training sessions start by collecting neutral data. Each training trial for a cognitive action lasts for 8 seconds, which is enforced by the software. Once the headset was trained,

the peers who helped with testing the software were presented with our Google Street View Application and asked to explore the streets through cognitive actions and facial expressions.

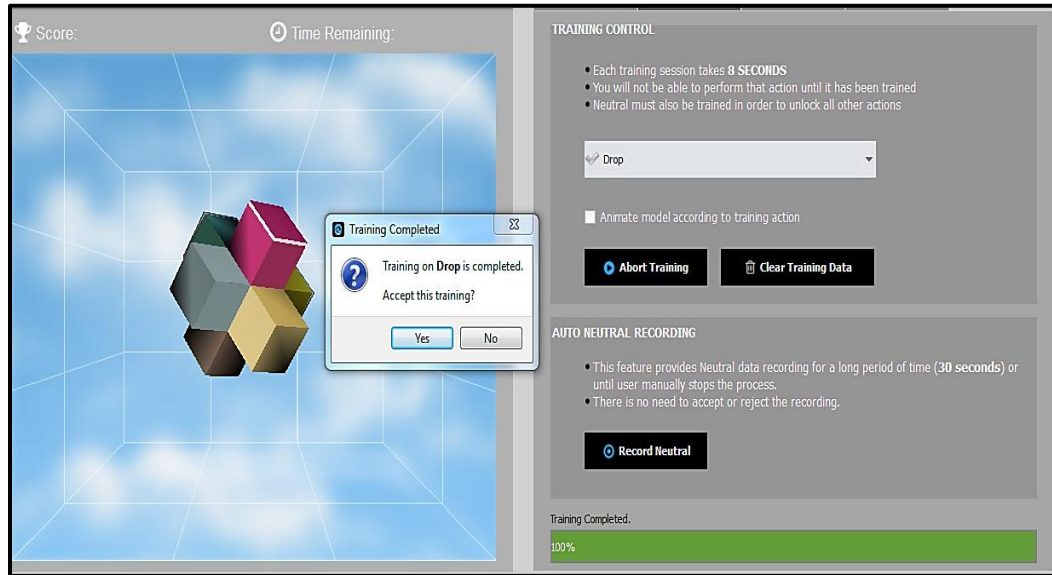
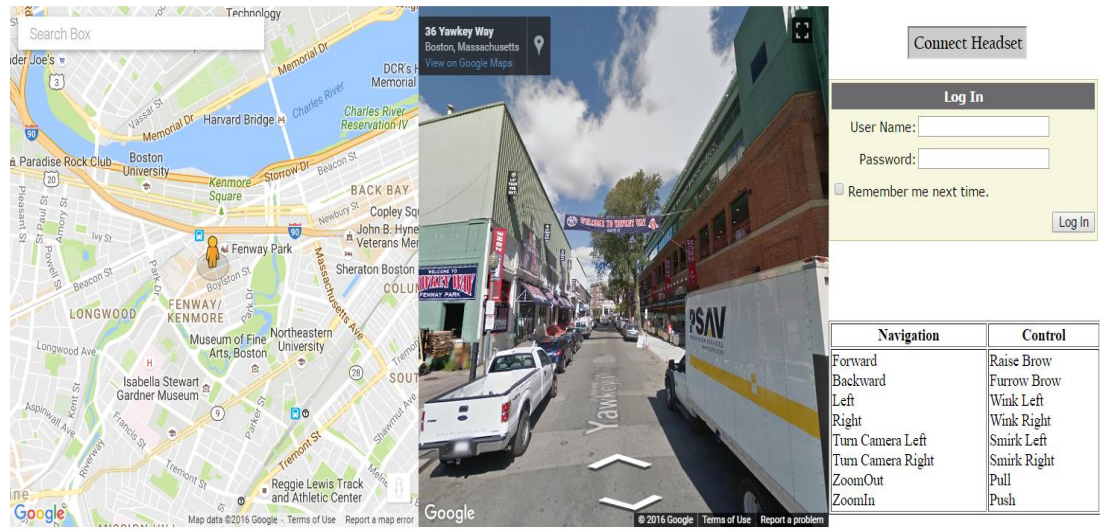


Figure 6.15: Illustration of Emotiv's training software [Emotiv, 2016]

Figure 6.16 illustrates a screenshot of our BCI application's interface. The defaulted location within is 36 Yawkey Way in Boston, Massachusetts.



Steps:

- 1) Click "Connect Headset".
- 2) Train headset.
- 3) Upload training profile to cloud.
- 4) Log in.

Figure 6.16: Snapshot of our application's web interface

6.5 Preliminary Results

The questionnaire we used to elicit feedback from our users after the software testing session is shown in Figure 6.17. This section reveals the major feedback we received in addition to any significant observations we made during the testing session.

Please note this is not a usability test, rather a software testing session in which we wish to elicit feedback from software developers

1. Did the software work as expected? Please comment to what extent [specification].
2. What features do you think are important for this software application? [specification]
3. What features are less important for application testing against its specification?
4. Related to 2 and 3 above, what features would you recommend to either add and remove to/from this application?
5. Did you find any software deficiencies? [e.g., bugs or lack of responsivity]
6. Do you have any technology recommendations for future versions? [e.g., architecture solutions, programming languages, development tools]
7. Do you have any other recommendation to improve the development process and the quality of this software application?
8. What is your overall impression about the purpose of this software? Please comment.

Figure 6.9: Software testing questionnaire

Feedback

Only some of our peers reported in their responses to the first question of the questionnaire that the software worked as expected. The software worked to the extent that they were able to successfully navigate through the streets of Google Street View. Features such as zooming out, moving backwards, and turning left and right behaved as expected. Furthermore, other responses revealed that the software could have been more responsive towards various actions. In particular, the software was less responsive to the “zoom in” and “move forward” commands.

For response to the second question, our feedback indicated that it was important for the software to take into account facial expressions as well as their thoughts for navigation. Facial expression detection alleviates further mental strain when executing commands, which is a significant feature.

In response to the third question, our users reported that all the features within this software application are important. However, the least important feature is zooming in and zooming out of the street view and map.

Responses to the fourth question revealed that an important feature to add to the software would be an improvement in accuracy. Winking commands affected the ability to zoom in and zoom out properly in.

One major software deficiency our users reported as a response to the fifth question was the over-responsiveness towards actions executed on the dominant side of the face and under responsive towards the non-dominant side of the face. Our feedback also revealed that moving forward and backward, which are executed through raising and lowering eyebrows, conflicted with zooming in and zooming out of the street, which are controlled by cognitive actions.

Our users reported in their response to the sixth question that they recommend utilizing a different BCI headset, one that has better accuracy in its detection. Another recommendation was utilizing some form of eye tracking within the application.

As a response to the seventh question, in order to improve the quality of the software, our users recommended to remove the winking feature, or at least alter the detection threshold to not affect the other directional movements. One of our users suggested modifying the software to detect which specific directional arrow the user is thinking of to improve upon navigation through Street View. Furthermore, our feedback responses also recommended further testing the responsiveness in types of commands to determine its optimal threshold level for detection. Incorporating other BCI apparatus and applying a noise filter for EEG detection were additional recommendations we received as well.

Our responses to the eighth question indicated that this software has potential to serve as an accessibility feature within a navigation system. Our feedback also revealed that this software application may also be useful for people without motor abilities, such as interacting with Street View while it is displayed on a large television screen.

6.6 Summary

For applications that integrate GPS navigation and web searching such as Google Maps and Google Street View, speech commands may not be enough to explore a desired location. For Street View, direct manipulation from the mouse plays an important role in allowing users navigate through various streets and cities. We replaced mouse input with brain input in our Street View web application. Through cognitive or affective/expressive controls, the user is able to explore through the streets and alleyways of any location in the world.

7 STUDY 3: BCI INTEGRATION WITH WATERSHED

This chapter introduces and explores the integration of BCI technologies within WC-WAVE's Virtual Watershed Platform. We developed a managed Unity plugin as an extension of the Virtual Watershed Client, which provides BCI accessibility. The design of our plugin is outlined in detail. This chapter also presents a walkthrough of our prototype along with some feedback we received software testing. This chapter's contents will be used in a paper soon to be submitted to an international conference [Chan et al, 2017b].

7.1 Overview

The Virtual Watershed Client is a 3D analytical tool that provides data scientists and researchers a way to visualize model data [Carthen et al, 2015]. Since the Virtual Watershed Client is built using the Unity game engine, we generated Unity-specific scripts to demonstrate the functionalities of our plugin. However, our plugin may also be used in applications built outside of Unity. Part of the development in this plugin is influenced by the obtained results and feedback, in addition to the successful methods, of the previous two systems. Similar to our web application for Google Street View, the user may interact with and navigate through the watershed using expressive actions and cognitive actions. The user will also have a hands-free control over the mouse cursor, which will be driven by the user's head movements. Similar to Sim-Assist and our Google Street View application, users must undergo a training session in order to execute any cognitive controls. Users may also load any pre-existing training profile by signing into their Emotiv cloud account.

7.2 BCI Plugin Integration Design

From a high-level, our plugin integration with the Virtual Watershed Platform consists of different layers in its architecture. The lowest layer is the monitoring and interpreter, where our plugin takes care of detecting new cognitive and expressive states, and keeps track of gyroscope data within the Epoc+ headset. This is accomplished by the three monitoring subsystems. The interpreter subsystem takes in data from the headset and

determines if they are valid commands/actions. If the input is valid, then the data are sent to the higher level, which is the plugin core. The core is responsible for all the events within our BCI extension. The top layer is the Virtual Watershed Client, which is what our plugin will interact with. Figure 7.1 provides an expanded view of our layers.

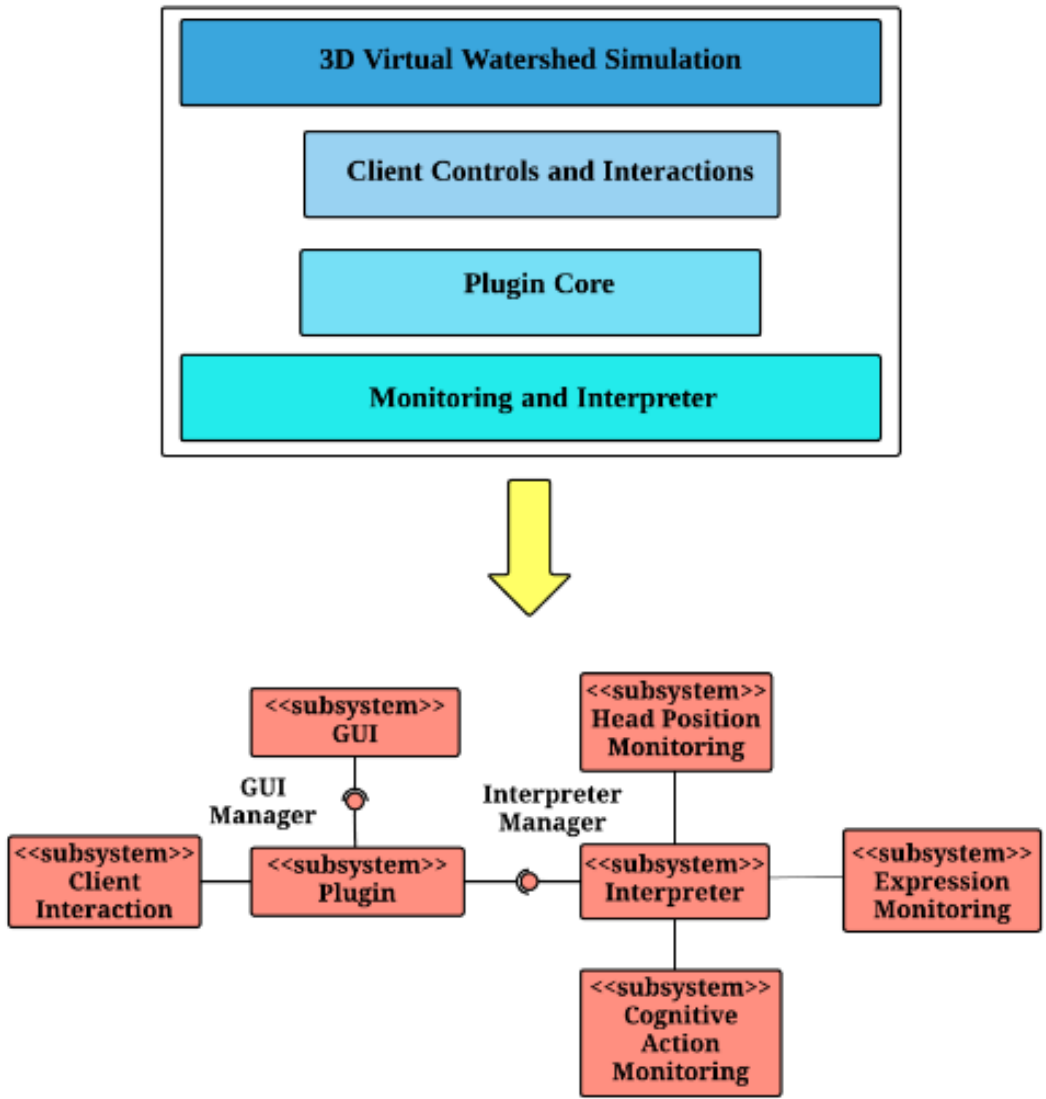


Figure 7.1: High-level diagram of BCI Plugin's integration with the Virtual Watershed Client

Figure 7.2 illustrates how our BCI plugin interacts with the Virtual Watershed Client through a workflow diagram.

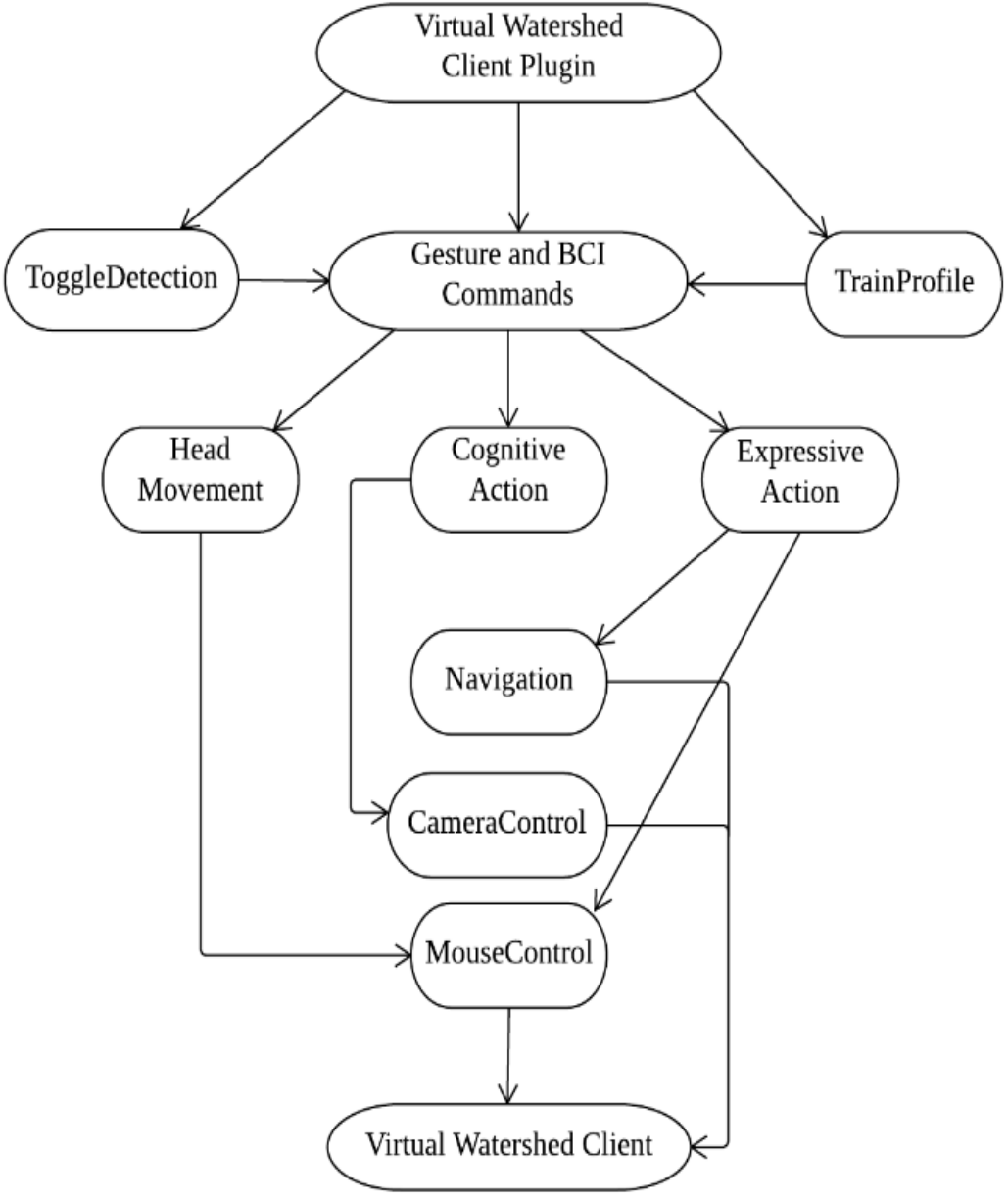


Figure 7.2: Workflow diagram of BCI plugin interaction with Virtual Watershed Client

Figure 7.3 illustrates the behavior of our BCI controller, which acquires the user’s brain signals and determines if the user is executing a cognitive or affective command. Since the headset always detects EEG data, we set our threshold for cognitive command detection to 0.4. Consequently, the threshold for expressive commands is 0.2. This minimizes any false detection or unintentional cognitive or expressive action.

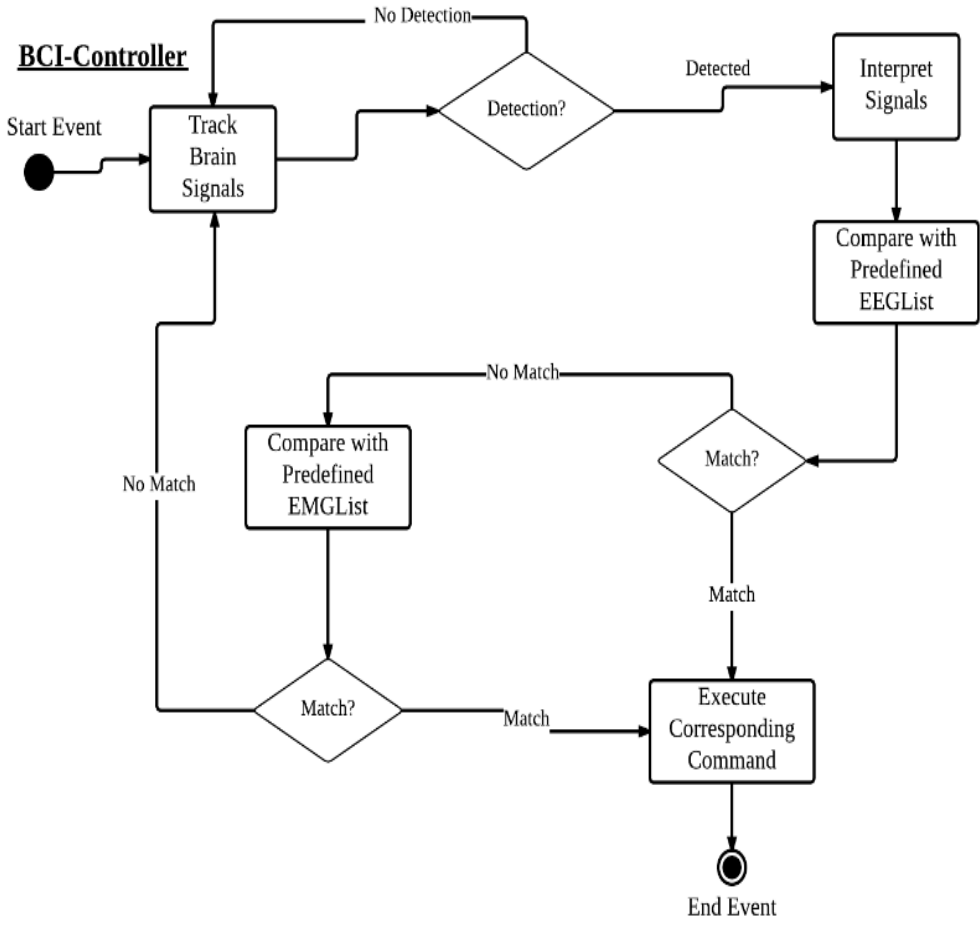


Figure 7.3: Activity diagram for BCI-Controller

Figure 7.4 describes the behavior of our mouse controller, which retrieves gyroscope data from the headset. This data will be used to position the mouse cursor on the screen. The controller also listens for an expressive action (“left blink”) for any mouse clicks that have been executed by the user.

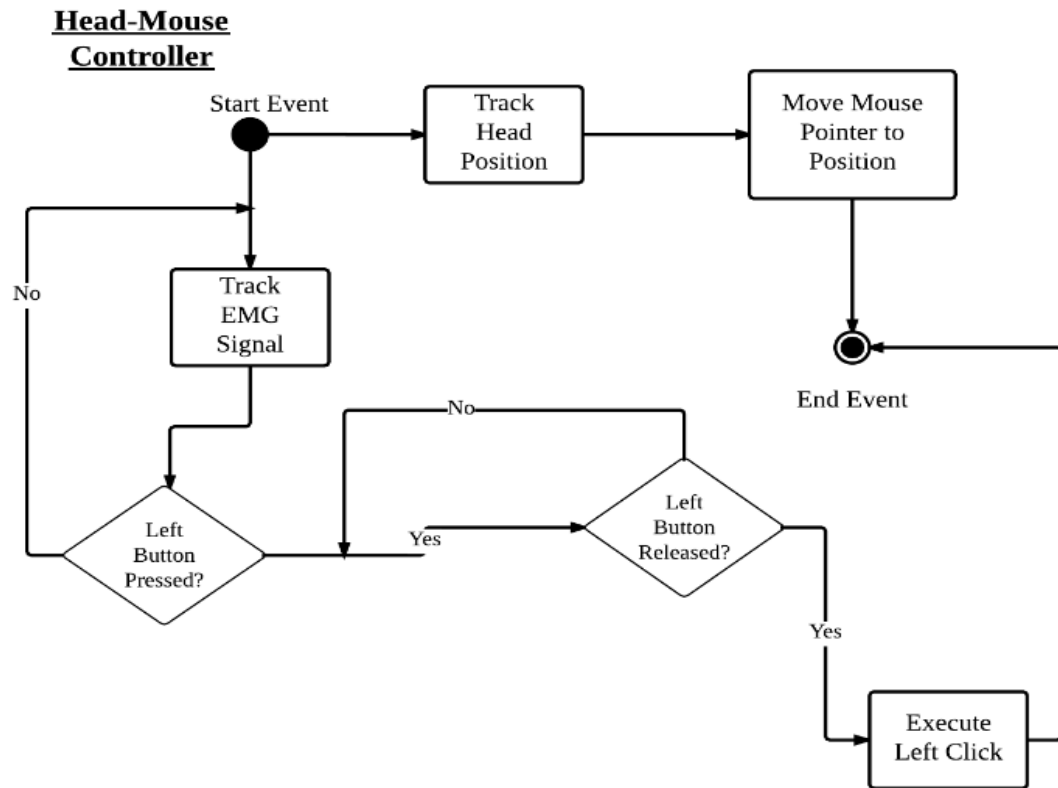


Figure 7.4: Activity diagram for head-mouse controller

7.3 Use Case Modeling

For better understanding of how the user will use the BCI plugin to interact with the Virtual Watershed Client, this section describes use cases for when the Virtual Watershed Client is executed. The use case model is shown in Figure 7.5.

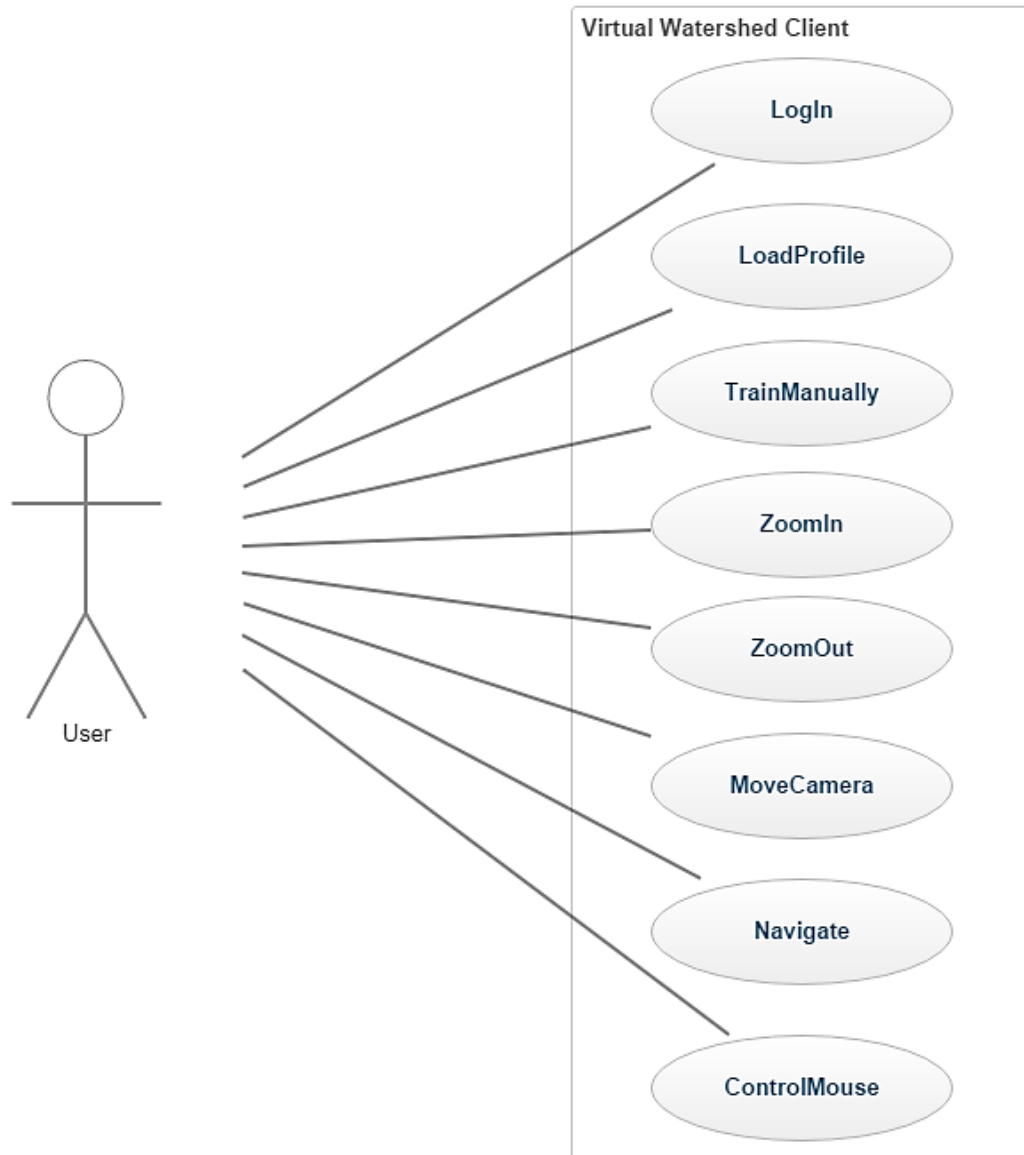


Figure 7.5: Use case diagram of user interaction with the Virtual Watershed Platform through the BCI plugin

Further detail in each use case for our interaction with the Virtual Watershed Platform through the BCI plugin is shown in Figures 7.6 and 7.7.

Use Case Descriptions		
UC01	LogIn	The user is presented with a login menu. The user must enter his/her Emotiv account information in order for the plugin to retrieve the user's training profile.
UC02	LoadProfile	The user's training profile is loaded from the user's Emotiv Cloud profile. Only one profile can be loaded at a time.
UC03	TrainManually	If the user does not have an Emotiv profile, the user may train cognitive actions manually. The user clicks a train button and must hold that cognitive action for 8 seconds. The user will then be notified when the training is finished.
UC04	ZoomIn	The user may use cognitive actions to zoom into the Virtual Watershed Client. The defaulted cognitive action to zoom in is the "Pull" action.

Figure 7.6: Use Case Descriptions for LogIn, LoadProfile, TrainManually, and ZoomIn

Use Case Descriptions		
UC05	ZoomOut	The user may use cognitive actions to zoom out. The default cognitive action to zoom in is the “Pull” action.
UC06	MoveCamera	The user may use facial expressions to move the camera left and right. The default expressive actions are “look left” and “look right”.
UC07	Navigate	The user may use facial expressions to navigate through the Virtual Watershed Client. The default expressive actions are “raise eyebrows” (forward), “lower eyebrows” (backward), “smirk left” (left), and “smirk right” (right).
UC08	ControlMouse	The user may use head movements to move the mouse cursor. Only left clicking is available, which is activate by “blink left”

Figure 7.7: Use Case Descriptions for ZoomOut, MoveCamera, Navigate, and ControlMouse

7.4 System Walkthrough

Similar to the systems presented in Chapters 5 and 6, we must first train the headset to detect the correct cognitive actions using Emotiv's control panel software (Figure 7.8).

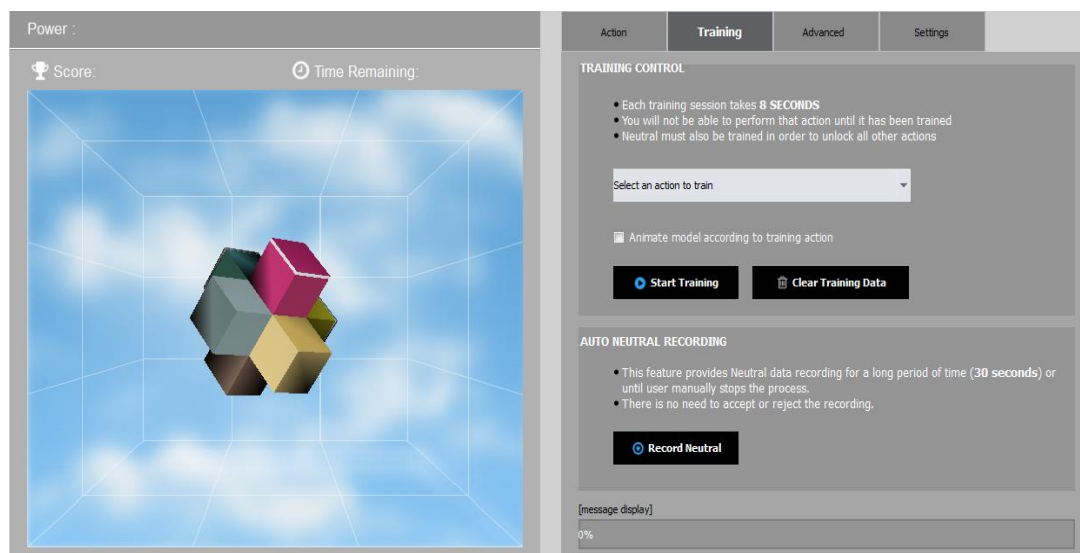


Figure 7.8: Snapshot of Emotiv Control Panel's training session [Emotiv, 2016]

Users begin all training sessions by recording neutral data. This will provide a base signal for easier classification of other cognitive actions. Following the neutral state, users will train “Push” and “Pull”. This will take several attempts in order for the software to acquire a sufficient number of trained signatures for accurate detection.

Once training is complete, the user will launch the Virtual Watershed Client executable. Users are initially presented a main menu (Figure 7.9) that will allow users to connect the headset to the program, log in to retrieve their training profile, and start interacting with the Virtual Watershed Client.



Figure 7.9: Virtual Watershed Client Main Menu (with BCI integration)

In the login menu, users can log into their Emotiv Cloud profiles (Figure 7.10). If they do not have an account, they can manually train cognitive actions without the software. However, there will not be any visual aid to help with training.

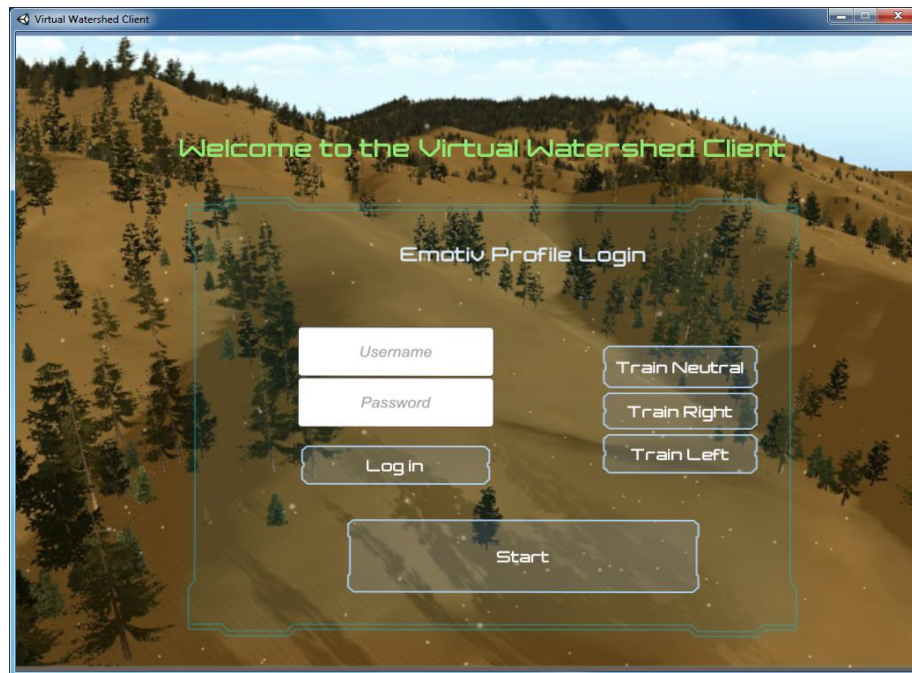


Figure 7.10: Login Menu of Virtual Watershed Client (with BCI Integration)

After the user's training profile is loaded, the user may proceed to the Virtual Watershed Search Catalog (Figure 7.11). We note that in this walkthrough, the Dry Creek watershed was used. In the search catalog, users can select a dataset to be loaded from file or downloaded from the server. The dataset will be used to generate the simulation, with which the user can interact.



Figure 7.11: Snapshot of Virtual Watershed Catalog Search

Once the simulation is loaded, the user can select points of interest using our head-mouse controller, use thoughts and facial expressions to move the camera, and navigate through our BCI controller (Figure 7.12)

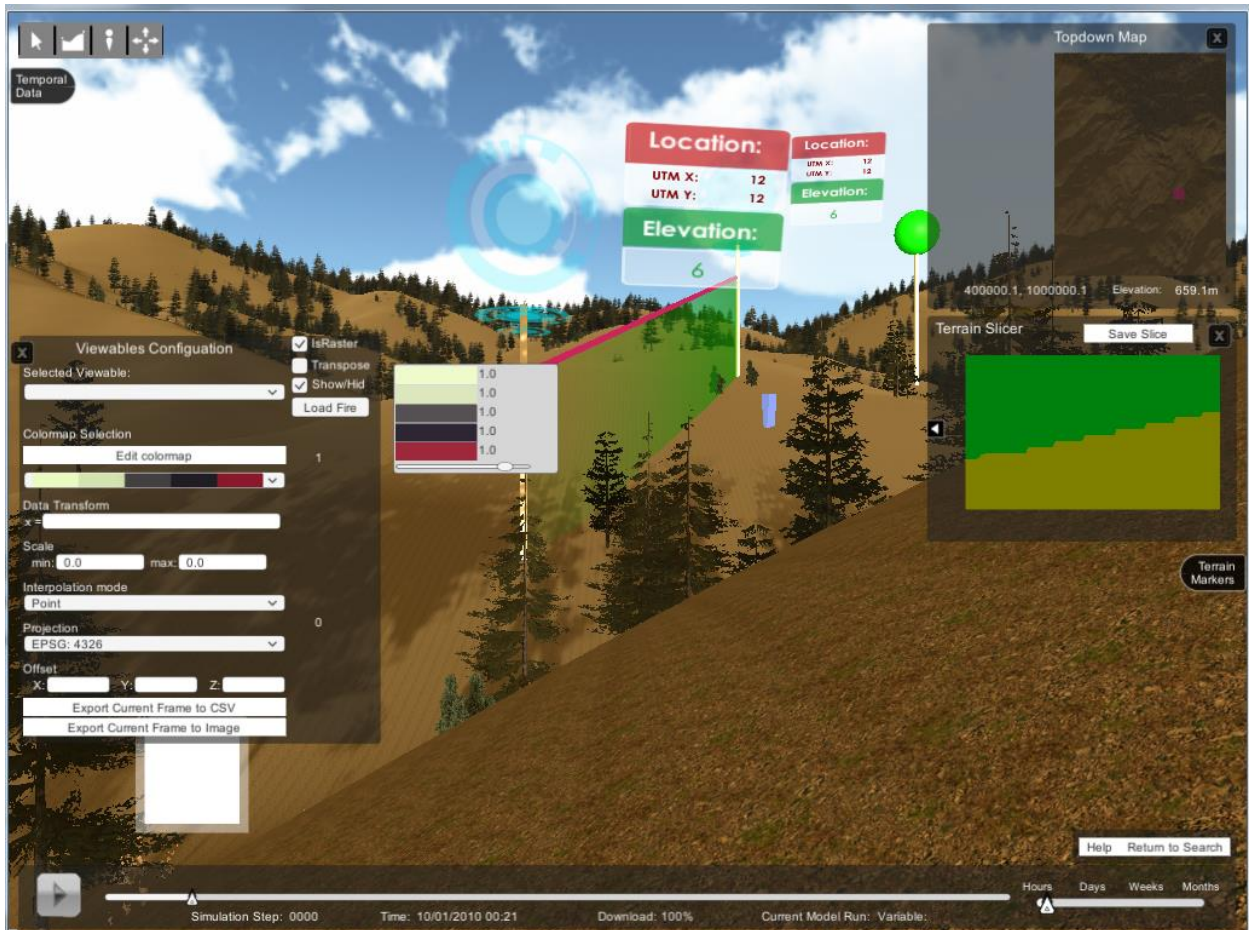


Figure 7.12: Snapshot of Virtual Watershed Client in action

7.4 Preliminary Results

We informally tested our BCI-Watershed integration with the help from a group of colleagues in the Computer Science Department at the University of Nevada, Reno. The questionnaire we used to elicit feedback from our peers after the software testing session was the same as the one shown in Figure 6.9. This section reveals the major feedback we received from the questionnaire in addition to significant observations we made during the software testing session.

Feedback

In response to the first question in the questionnaire, our peers indicated that the software only partially worked as expected to the extent that they were able to interact with the Virtual Watershed Client with their facial expressions and head movements, but there

were some issues executing commands. Moving the cursor did not behave as expected; cursor control seemed to be over- responsive or too sensitive. However, the software was able to easily detect winking and furrowing brow actions, which allowed users to execute mouse clicks and move backwards. The feedback also noted that there was a slight lag for moving left and right within the watershed.

The majority of the responses to the second question revealed that the most important feature of this software is cursor control. Other important features include the ability to load the user's training profile from the cloud and manually training the user if he/she does not have an Emotiv account.

In response to the third question, our users indicated that one of the least important features was zooming in and zooming out within the environment. Our feedback also revealed that the manually training feature is theoretically important, but less important within this software.

Responses to the fourth question revealed that one important feature our users would have liked to see was some sort of display of the commands/controls for the software. Our users would have also liked a "drag" feature in the mouse control.

For the fifth question response, one major software deficiency our users reported was the over-responsiveness towards the cursor control and under responsiveness towards moving left and right.

Our users recommended, in response to the sixth question, incorporating additional BCI apparatus within our application.

In response to the seventh question, our users recommended lessening the sensitivity in the headset for cursor control and increasing the detection threshold for moving left and right. It was also recommended to change the type of cognitive action input or allowing users to change the cognitive controls themselves. Other recommendations to our software included using additional unrelated features as input, and including a feature where users could test each interface control during the testing phase.

Responses to the eighth question revealed that our application possesses great potential to allow users with motor impairments to interact with the Virtual Watershed

Client. Furthermore, additional responses indicated that the application may also be useful if the interface was to be presented on a large display.

7.5 Summary

This chapter showcased our third and final software development, the Virtual Watershed accessibility plugin. To demonstrate the capabilities of our plugin, we generated scripts in the Virtual Watershed Client's Unity project. As a result, the user was able to load their training profiles from the cloud, navigate through the simulation using facial expressions, control the mouse using head and facial movements, and move the camera through cognitive thoughts. Feedback revealed that improvements must be made in order for a more natural interaction with the simulation.

8 CONCLUSIONS AND FUTURE WORK

This work implemented three related software components to further explore the integration capabilities of assistive technologies within a virtual environment.

The first system is Sim-Assist, which allows for a hands-free way for people with or without disabilities to interact with a three-dimensional simulation game of Air Hockey using speech and cognitive thoughts. Sim-Assist demonstrates that brain control can be used to interact with objects in 3D. However, this control is better for slower, less reactive environments where sensitivity and reaction is not as important such as 3D environments that show data. We also observed that any input reliant on cognitive control is heavily demanding on the user's mind, which affects overall performance of the software.

The second software component is a web application compatible with Google Street View, which allows users to navigate through the streets of any city using their thoughts and facial expressions. Our application reveals that BCI control can be used to navigate through a static 3D environment. This application also justifies that a BCI system cannot solely depend on cognitive control as it causes additional mental strain on the user. Therefore, a mix of cognitive and expressive controls should be used in a BCI application. However, one must take into consideration that some facial expressions may affect cognitive detection.

The third development is the Virtual Watershed Plugin, which adds an accessibility feature within the Virtual Watershed Platform. This plugin enables BCI capabilities for users to interface with the Virtual Watershed Client using head movements and cognitive and expressive actions. The developments of the first two applications acted as a gradual prelude to our third software component, as our BCI plugin applies some of their concepts and improves upon their shortcomings. From the observations obtained from Sim-Assist, we integrated our plugin within a static environment such as a 3D simulation of a watershed. From the observations obtained from our web application for Google Street View, we altered the software to mainly accept expressive control as input instead. In order to further the user's interactions within the Virtual Watershed Client, we added a mouse control feature that moves the cursor using head movements. We observed from the Google

Street View application that the two types of controls (cognitive and expressive) conflicted with each other. For our integration with the Virtual Watershed's interface, moving the head to control the cursor would have affected cognitive commands, but not expressive. Therefore, we mapped cognitive commands to the more flexible controls such as zooming in and zooming out, which would not have impacted the interface if accidentally executed.

Main contributions of the work presented in this thesis include: a survey of the relatively new domain of Brain Computer Interfaces and exploration of its major related research and development efforts; an established classification taxonomy of various types of Brain Computer Interfaces; studies on three increasingly more complex systems in BCI to expand assistive technology research; a demonstration of how assistive technologies can or cannot be integrated within static and dynamic environments; and a formulation of a usability study outline for future BCI research, which will be submitted to an international conference.

The remainder of this chapter discusses the main areas where more work can be done.

Future Development and Testing

For future work in Sim-Assist, we may add extra functionality and features into the system. There could be an AI opponent in the Air Hockey game so that single players will have more fun playing alone. Currently it is difficult to execute commands fast enough to maintain level with Air Hockey's fast, dynamic nature. Therefore, we may include some expressive functionality since expressions take less time to perform. We also strongly consider using a different EEG headset since the newest Emotiv EPOC model, EPOC+, we will have more accurate EEG readings.

For the BCI plugin used in the Virtual Watershed Client, we plan to implement a more elegant method to interact with the panels on the interface. A display of the controls and commands or interaction may also be included, depending on whether it will add any additional clutter on the interface. In addition, we plan to remove cognitive control entirely from the software, including the zooming in and zooming out feature, which seemed

unnecessary in this application. Three separate methods of different inputs may be too overwhelming for users. Furthermore, both head and facial movements would definitely affect cognitive detection, whereas cognitive detection may or may not induce unintended facial or head movements. We also would like to have an option to adjust sensitivity levels for cursor control. In addition, customizable mappings to controls may be implemented in the future.

For our Web application with Google Street View, we plan to include a hands-free method of inputting location searches. We also would like to have a better understanding of the optimal detection threshold levels that can be used across the majority of users. In the future, we may include an option for users to change the different types of BCI mappings to Street View controls. We also plan to conduct a user study for our Web application with Google Street View based on observations from the software testing from Chapter 6. Results obtained from this study will be used in a conference paper in preparation.

In the future, our testing procedure will significantly be more elaborate than that from our software testing session. All new training sessions for each participant will start by collecting neutral data. For training the neutral state, participants will be informed to relax and keep a clear mind if possible. Participants must look at the cube presented to them without the intention to move it. Before training any other cognitive actions, participants will be given the advice to imagine performing a physical action on the cube. This is essential since we observed our testers in the software testing struggled with cognitive training. Participants will be given some time before each new trial to ensure they had a concrete thought for that action. These thoughts will need to be consistent throughout the whole trial. Within and between trials, participants will be instructed not to move or tense any facial or neck muscles. Any movements by other parts of the body would also affect results. This is also another key component to successful training. From our software testing, we observed our testers moved often. All participants will be required to practice moving the cube after each trial. Doing so would allow for the subject to understand which actions need further training.

The testing phase will consist of an equal number of trials for cognitive and affective/expressive control. Subjects will be asked to perform an adequate number trials of each navigation control (right, left, forwards, and backwards) in a randomized order. The number of successes and failures for each navigation control will be counted in order to determine the overall success rate for each participant.

We expect our results to reveal that affective/expressive control is significantly easier than cognitive. This is because cognitive tasks require training, which can be mentally demanding on the user. We also expect the lack of experience or BCI illiteracy in our subjects may also have an effect on their performance. This is because users with a substantial amount of EEG experience may have more focus when controlling their thoughts. They may also have acquired knowledge about which techniques work best for their own training or cognitive execution, allowing for better performance in BCI applications.

REFERENCES

- American Foundation for the Blind (2012). Facts and figures on adults with vision loss. <http://www.afb.org/info/blindness-statistics/adults/facts-and-figures/235>. Accessed December 1, 2016.
- Andrew (2012). Blind man reads braille patterns with eyes, using implanted retinal prosthesis. <http://neurogadget.com/2012/11/30/blind-man-reads-braille-patterns-with-his-eyes-using-implanted-retinal-prosthesis/6292>. Accessed December 1, 2016.
- Andronico, P., Buzzi, M., Castillo, C., and B. Leporini (2006). Improving search engines for blind users: A Case Study. *Universal Access in the Information Society*, pp. 23-40, 2006.
- Barreto, A., Scargle, S, and M. Adjouadi (2000). A practical EMG-based human-computer interface for users with motor disabilities. *Journal of Rehabilitation Research and Development*, pp. 53-64.
- Birbaumer, N., Elbert, T., Canavan, A.G., and B. Rockstroh (1990). Slow potentials of the cerebral cortex and behavior. *Physiological Reviews*, pp. 1-41.
- Bradley, N. and M. Dunlop (2005). Toward a multidisciplinary model of context to support context-aware computing. *Human-Computer Interaction* 20 (4): 403-446, 2005.
- Bright, A. K. and L. Coventry (2013). Assistive Technology for Older Adults: Psychological and Socio-Emotional Design Requirements. The 6th Intl. Conference on Pervasive Technologies Related to Assistive Environments (PETRA-2013), ACM, pp. 9-12.
- Burgess, D. Techniques for low cost spatial audio (1992). In *Proceedings of the 5th annual ACM symposium on User Interface software and Technology*, pp. 53-59.
- Carmena, J.M., Lebedev, M.A., Crist, R.E., O’Doherty, J.E., Santucci, D.M., Dimitrov, D.F., Patil, P.G., Henriquez, C.S., and M.A.L., Nicolelis (2003). Learning to control a brainmachine interface for reaching and grasping by primates. *PLoS Biology*, vol. 1, pp. 193–208.

- Carthen, C., Rushton, T., Johnson, C., Hesson, A., Nielson, D., Worrel, B., Anderson, J., Lew, R., Wood, R., Delparte, M., Johansen, M., Dascalu, S., and F.C. Harris, Jr. (2015). Design of a Virtual Watershed Client for the WC-WAVE Project. *Proceedings of the 2015 International Conference on Collaboration Technologies and Systems (CTS-2015)*, pp. 90-96.
- Center of Assistive Research Technologies (2010). Assistive Technologies Research Center (CART). <http://cecs.wright.edu/atrc/>. Accessed December 1, 2016.
- Chan, A., Quiroz, J., Dascalu, S., and F. C. Harris (2015). An Overview of Brain Computer Interfaces. *Proceedings of CATA-2015*, Honolulu, HI, pp. 327-334.
- Chan, A. and S. Dascalu (2015), Brain Computer Interfaces: Overview and Directions of Work. *Poster*. UbiComp-2015 Broadening Participation Workshop, Osaka, Japan.
- Chan, A., Gamino, A., Harris, F.C., Jr., and S. Dascalu (2016). Integration of Assistive Technologies into 3D Simulations: An Exploratory Study. *Proceedings of the 2016 International Conference on Information Technology: New Generations*, Las Vegas, NV, April 2016. In Shahram Latifi (ed.), *Advances in Intelligent Systems and Computing*, Springer International, vol. 448, pp. 425-437.
- Chan, A., Dascalu, S., and F.C. Harris, Jr. (2017a). Enhancing the Virtual Watershed Platform with Brain Computer Interfaces. To be submitted in Spring 2017 to the International Conference on Collaborative Technologies and Systems (CTS-2017), location TBD.
- Chan, A., Harris, F.C., Jr., and S. Dascalu (2017b). Using Brain Computer Interface Technology in Connection with Google Street View. To be submitted in December 2016 to the International Conference on Computers and Their Applications (CATA-2017), Honolulu, Hawaii, March 2017.
- Cheng, M., Xiaorong G., Shangkai G., and D. Xu (2002). Design and implementation of a brain-computer interface with high transfer rates. *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 10, pp. 1181-1186.
- Cogan, D. (2015). Bobby Approved: Practical Web Accessibility Tool Icon. <http://www.coggan.com/bobby-approved.html>. Accessed December 1, 2016.

- Donker, H., Klante, Palle, and P. Gorny (2002). The design of auditory user interfaces for blind users. *Proceedings of the second Nordic conference on Human-Computer Interaction*.
- Donoghue, J.P., Nurmikko, A., Black, M., and L.R. Hochberg (2007). Assistive technology and robotic control using motor cortex ensemble-based neural interface systems in humans with tetraplegia. *The Journal of Physiology*, pp. 603-611, 2007.
- Edyburn, Dave L. (2006). Assistive technology and mild disabilities. *Special Education Technology Practice*, pp. 18-28, 2006.
- Emotiv (2016). <http://emotiv.com>. Accessed December 1, 2016.
- Farewell, L.A. and E. Donchin (1998). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology* 70 (6): 510-523.
- FCTD (2015). Assistive Technology 101. <http://www.fctd.info/factsheet/at101>. Accessed December 1, 2016.
- Galan, F., Marnix, N., Elleen, L., Pierre, W., Ferrez G.V., Johan, P., and J.R. Millan (2008). A brain-actuated wheelchair: asynchronous and non-invasive brain-computer interfaces for continuous control of robots. *Clinical Neurophysiology*, pp. 2159-2169, 2008.
- Gajos, K., Wobbrock, J., and D. Weld (2014). Automatically Generating User Interfaces Adapted to Users' Motor and Vision Capabilities. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, pp. 231-240, ACM, Georgia Department of Education. 2014.
- Georgia Project for Assistive Technology (2016). <http://www.gpat.org/>. Accessed December 1, 2016.
- Gomez, J., Montoro, German, Haya, P. A., Alaman, X., Alves S., and M. Martinez (2013). Adaptive manuals as assistive technology to support and train people with acquired brain injury in their daily life activities. *Personal and Ubiquitous Computing* 17 (6): 1117-1126.
- Graimann, B., Brendan, A., and A. Graser (2007). New Applications for Non-invasive Brain-Computer Interfaces and the Need for Engaging Training Environments.

- Brain-Computer Interfaces and Games Workshop at the International Conference on Advances in Computer Entertainment Technology, pp. 25-28.
- Grizou, J., Iturrate, I., Montesano, L., and P.Y. Oudeyer (2014). Calibration-Free BCI Based Control. Twenty-Eighth AAAI Conference on Artificial Intelligence.
- Guger, C., Schlogl, A., Neuper, C., Walterspacher, D., Strein T., and G. Pfurtscheller (2001). Rapid prototyping of an EEG-based brain-computer interface (BCI). *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 9: 49-58.
- Hersh, M. (2016). The Design and Evaluation of Assistive Technology Products and Devices Part 1: Design International. International Encyclopedia of Rehabilitation. <http://cirrie.buffalo.edu/encyclopedia/en/article/309/>. Accessed December 1, 2016.
- Hinterberger, T., Schmidt, S., Neumann, N., Mellinger, J., Blankertz, B., Curio, G., and N. Birbaumer (2004). Brain-computer communication and slow cortical potentials. *IEEE Transactions on Biomedical Engineering* 51: 1011-1018.
- Hutchins, E.L., Hollan, J.D., and D.A. Norman (1985). Direct Manipulation Interfaces. In *User Centered System Design*, D. A. Norman and S. W. Draper, Eds. Hillsdale, NJ, Lawrence Erlbaum Associates Inc, pp. 87-124, 1985.
- iCommunicator (2010). Company Profile. <http://www.icommunicator.com/aboutus.shtml>. Accessed December 1, 2016.
- INDATA Project (2011). Apps for the iPad and Assistive Technology. <https://www.youtube.com/watch?v=3CXZa4FkIpA>. Accessed December 1, 2016.
- Institute on Disability (2014). Annual Disability Statistics Compendium. <http://disabilitycompendium.org/>. Accessed December 1, 2016.
- Kennedy, P. R. and R. A. Bakay (1998). Restoration of neural output from a paralyzed patient by a direct brain connection. *Neuroreport* 9 (8): 1707-1711.
- Krusienski, D.J., Grosse-Wentriup, M., Galan, F., Coyle, D., Miller, K.J., Forney, E., and C.W. Anderson (2011). Critical issues in state-of-the-art brain-computer interface signal processing. *Journal of Neural Engineering* 8 (2).
- Junker, A. M., Sudkamp, T., Eachus, T., Mikov, T., Wegner, J., Edmister, E., Livick, S., Heiman-Patterson, T., and M. Goren (2001). Handsfree computer access for the severely disabled.

- http://www.wright.edu/lwd/documents/computeraccess_mateo.pdf. Accessed December 1, 2016.
- Lalor, E.C., Kelly, S.P., Burke, R. Smith, R., Reilly, R.B., and G. McDarby (2005). SteadyState VEP-based brain computer interface control in an immersive 3-D Gaming Environment, *EURASIP J. Applied Signal processing*, pp. 3156-3164.
- Lippman, A (1980). Movie maps: an application of the optical videodisc to computer graphics, *ACM SIGGRAPH Computer Graphics*, vol. 14, no. 3, pp. 32-42.
- Leuthardt, E.C., Schalk, G, Wolpaw, J.R., Ojemann, J.G., and D.W. Moran (2004). A brain-computer interface using electrocorticographic signals in humans. *Journal of Neural Engineering* 1 (2): 63-71.
- Li, J. and L. Zhang (2012) Active training paradigm for motor imagery BCI. *Experimental Brain Research* 219 (2): 245-254.
- Low Vision Aids and Technology (2016). Macular Degeneration Foundation. http://www.mdfoundation.com.au/resources/1/MDF_LowVisionAids.pdf. Accessed December 1, 2016.
- Miatliuk, K., Nawrocka, A., and K. Holewa (2014). Conceptual Design of BCI in the Formal Basis of Hierarchical System. In *Proceedings of the 15th International Carpathian Control Conference (ICCC)*, pp. 336-341.
- Microsoft (2015). Assistive Technology Products. <http://www.microsoft.com/enable/at/>. Accessed December 1, 2016.
- Microsoft (2016). Hear text read aloud with Narrator. <https://support.microsoft.com/en-us/help/17173/windows-10-hear-text-read-aloud>. Accessed December 1, 2016.
- Middendorf, M., McMillan, G., Calhoun, G., K.S Jones (2014). Brain-computer interfaces based on steady-state visual evoked response. *IEEE Transactions on Rehabilitation Engineering* 8: 211-213.
- Millán, J.R., Renkens, R., Mourino, J., and W. Gerstner (2003). Non-invasive Brain-actuated Control of a Mobile Robot. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pp. 1121-1126.
- Müller, K.R., Krauledat, M., Dornhege, G., Curio, G., and B. Blankertz (2004). Machine learning techniques for brain-computer interfaces, *Biomedical Engineering*.

- Mynatt, E.D. and G. Weber (1994). Nonvisual Presentation of Graphical User Interfaces: Contrasting Two Approaches. In *Proceedings of the Human Factors in Computing Systems Conference, CHI'94*, ACM Press, pp. 166-172.
- Narasimhan, P., Gandhi, R., and D. Rossi (2009). Smartphone-Based Assistive Technologies for the Blind. *International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, pp. 223-232.
- Neuper, C. and W. Kilmesch (2006). *Event-related dynamics of brain oscillations*. Elsevier.
- Neurogadget (2016). <http://neurogadget.com/>. Accessed December 1, 2016.
- NeuroSky (2016). <http://neurosky.com>. Accessed December 1, 2016.
- Nuance (2016). Dragon Speech Recognition Software. <http://www.nuance.com/dragon/index.html>. Accessed December 1, 2016.
- Origin Instruments Corporation (2016). Wireless Head Controlled Mouse. <http://www.orin.com/access/headmouse/>. Accessed December 1, 2016.
- Petrie, H., Morley, S., and G. Weber (1995). Tactile-based Direct Manipulation in GUIs for Blind Users, in *Conference Companion on Human Factors in Computing Systems, CHI'95*, New York: ACM, pp. 428-429.
- Pino, A., Kalogeros, E., Salemis, E., and G. Kouroupetroglou (2003). Brain Computer Interface Cursor Measures for Motion-impaired and Able-bodied Users. In *Proceedings of the 10th International Conference on Human-Computer Interaction*, vol. 4, pp. 1462- 1466.
- Pitt, I. J. and A.D.N. Edwards (1991). Navigating the Interface by Sound for Blind Users. In *Proceedings of the HCI'91 Conference on People and Computers VI*, p. 373-383.
- Pfurtscheller, G., Neuper, C., Muller, G.R., Obermaier, B., Krausz, G., Schlogl, A., Scherer, R., Graimann, C., Keinrath, D., Skliris, M., Wortz, G., and C. Schrank (2003). Graz- BCI: State of the art and clinical applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, pp.177-180.

- Pfurtscheller, G., Neuper, C., Flotzinger, D., and M. Pregenzer (1997). EEG-based discrimination between imagination of right and left hand movement. *Electroencephalography & Clinical Neurophysiology*, pp. 642-651.
- Ravden, D. and J. Polich (1999). On P300 measurement stability: Habituation, intra-trial block variation, and ultradian rhythms. *Biological Psychology* 51: 59-76.
- RSLSteeper Assistive Technolog (2013). What Is Assistive Technology. <https://www.youtube.com/watch?v=SIIm2MuJUCTE>. Accessed December 1, 2016.
- Savidis, A., Stephanidis, C., Korte, A, Crispian, K., and K. Fellbaum (1996). A Generic Direct-Manipulation 3DAuditory Environment for Hierarchical Navigation in Non-visual Interaction, in *Proceedings of Assets '96*, New York: ACM, pp. 117-123.
- Shain, W., Spataro, L., Dilgen, J., Haverstick, K., Retterer, S., Isaacson, M., Saltzman, M., and J.N. Turner (2003). Controlling cellular reactive responses around neural prosthetic devices using peripheral and local intervention strategies. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 2 pp.186-188.
- Salem, C. and S. Zhai (1997). An Isometric Tongue Pointing Device. In *Proceedings of the ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI '97)*, pp. 538-539.
- School Health Corporation (2016). Enablemart--Assistive Technology. <https://www.enablemart.com/>. Accessed December 1, 2016.
- Serruya, M.D., Hatsopoulos, N.G., Paninski, L., Fellows, M.R. and J. Donoghue (2002). Instant neural control of a movement signal. *Nature*, pp. 141-142.
- Shain, W., Spetaro, L., Dilgen, J., Haverstick, K., Retterer, S., Isaacson, M., Saltzman, M., and J.N. Turner (2003). Controlling cellular reactive responses around neural prosthetic devices using peripheral and local intervention strategies. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, pp. 186-188.
- Smith, D.B. and E. Donchin (1970). The contingent negative variation and the late positive wave of the average evoked potential. *Electroencephalography and Clinical Neurophysiology*, pp. 201-203.

- Spence, J. (2008). Demographics of Virtual Worlds. (Journal For Virtual Worlds Research) <https://journals.tdl.org/jvwr/index.php/jvwr/article/view/360/272>. Accessed December 1, 2016.
- Sutter, E.E. (1992). The brain response interface: communication through visually induced electrical brain responses. *Journal of Microcomputer Applications* 15: 31-45.
- The OATS Project Consortium. OATS (2014). <http://www.oatsoft.org/Info/Project>. Accessed December 1, 2016.
- Vincent, L. (2007), Taking online maps down to street level, *Computer* 40 (12): 118-120.
- Weinberg, F. and S.O. Hellstrom (2000). The Quest for Auditory Direct Manipulation: the Sonified Towers of Hanoi. In *Proceedings of the 3rd International Conference on Disability, Virtual Reality and Associated Technologies (ICDVRAT 2000)*, pp. 75-81.
- Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., and T.M. Vaughan (2002). Brain-computer interfaces for communication and control. *Clinical Neurophysiology* 113 (6): 767-791.
- Wolpaw, J.R., McFarland, D.J., T.M. Vaughan (2000). Brain- computer interface research at the Wadsworth Center. *IEEE Transactions on Rehabilitation Engineering*, pp. 222- 226.