University of Hong Kong
COMP4801 - Final Year Project
Interim Report

Enhancement to MindDesktop: Improvement to a General Purpose
Brain Computer Interface

Project supervisor: Prof. Anthony Tam

Group members: Lau Chun Lam Frederick (3035123851)
               Tang Wang Hin Kenneth (3035123459)

Date of submission: 21st January 2018
Abstract

A proposed enhancement to a current Brain Computer Interface (BCI) system, MindDesktop, has been made. Reducing the number of input commands from three to two enables lower end Brain Computer Interface devices to serve the same purpose and increases the accuracy of reading inputs. For that purpose, the hierarchy of command for the system requires a redesign. The redesign also considers user input behavior that in turn reduces the number of commands needs to be made. Also, a different algorithm for the division of sections of the mouse selection from the implementation in MindDesktop will be used. Rather than using the method by divide the screen into four equal sections, a smarter algorithm will analyze the screen objects and will try to improve the performance of the selection process. The system will be divided into three components. The Connector component will act as an interface between the hardware device and the application. Training component will act as user profile collection application. Finally the Main component will provide all the features that mentioned above. In the preliminary stage, a research on different BCI has been conducted, Emotiv Insight has been chosen for the project and was tested by the team member. The research paper and demo video of MindDesktop has been reviewed for knowing more about how MindDesktop works.

Currently, the Connector component has been completed and tested. Graphical User Interface (GUI) framework for the Main Application has been developed. Navigation controls for the system via keyboard has been implemented with the propagation of keyboard commands for the command hierarchy tree but there is a problem on the propagation of a character and further investigation is required to resolve the problem. Making the mouse selection functional will be the immediate next task. An investigation on OpenBCI devices are conducting to consider to use them as alternative choice for the BCI devices due to the unreliable detection of EEG signal by using Emotiv Insight. The feasibility of using deep neural network with the raw signal data from OpenBCI is also being researched. If the OpenBCI is chosen to be used, the original optional Training component will become essential for the project since it strongly affects the result of machine learning.
Acknowledgement

We would like to thank our project leading professor, Dr. Anthony Tam for supporting us, through the development of the whole system. Dr. Tam has been giving us technical support and advices and also discuss with us to solve some problems that we encountered until this stage of development. Also, we would like to thank our project’s second examiner, Prof. C.L. Wang, for giving us advice on the focus of the remaining development of this BCI System to raise the value of this project.
Table of Contents

Abstract .............................................................................................................. I
Acknowledgement .............................................................................................. II
Table of Contents ................................................................................................ III - IV
List of Figures ..................................................................................................... V
Abbreviations ..................................................................................................... VI
Definitions ......................................................................................................... VII

1.0 Introduction .................................................................................................. 1 - 2
  1.1 BCI Background ......................................................................................... 1
  1.2 MindDesktop Introduction .......................................................................... 2
  1.3 Project Introduction .................................................................................. 2

2.0 Objective ...................................................................................................... 3
  2.1 Improving Efficiency of MindDesktop ....................................................... 3
  2.2 Enabling Possibility of Using Lower End Models ........................................ 3

3.0 Methodology ................................................................................................ 4 - 9
  3.1 Software Architecture of the System .......................................................... 4 - 9
    3.1.1 Connector Component ....................................................................... 5 - 6
    3.1.2 Main Component ............................................................................. 6
    3.1.3 Training Component ....................................................................... 7
  3.2 Reducing Number of Input Commands ...................................................... 7 - 8
  3.3 Redesign of Command Hierarchy for Better Efficiency .............................. 8 - 9
  3.4 Smarter Division Algorithm for Mouse Selection ...................................... 9

4.0 Project Status ............................................................................................... 10 - 18
  4.1 Preliminary Research on BCI Devices ....................................................... 10 - 11
  4.2 Development of Connector Component .................................................... 12 - 13
    4.2.1 Implementation of Connector Component ........................................ 12
    4.2.2 Testing of Connector Component .................................................... 13
  4.3 Proposed Command Hierarchy Tree ........................................................... 14
  4.4 Development of Main Component .............................................................. 15 - 18
    4.4.1 Client Receiver for Commands from Connector Component ............ 16
    4.4.2 Implementation of GUI Framework ................................................. 16 - 17
    4.4.3 Propagation of Keyboard Commands ............................................. 18
  4.5 Limitations of Hardware ............................................................................ 19
    4.5.1 Problems in Emotiv Insight .............................................................. 19
    4.5.2 Alternative BCI Devices ................................................................. 19
5.0 Future Plan .............................................................................................................. 20 - 22
5.1 Regulation of hardware adoption and relevant changes ................................. 20
5.2 Investigation on OpenVibe ................................................................................. 21
5.3 Investigation on Deep Neural Network ............................................................ 21
5.4 Modified Algorithm for Mouse Selection ....................................................... 21
5.5 Implementing Special Commands for Specific Applications ....................... 21 - 22
5.6 Plan for Training Component ......................................................................... 22
6.0 Project Schedule ............................................................................................... 23
7.0 Conclusion ......................................................................................................... 24

References ............................................................................................................... i
**List of Figures**

Fig. 1 An illustration on what is a BCI device[11]

Fig. 2 MindDesktop Input device state hierarchy[5]

Fig. 3 MindDesktop Pointing device UI[5]

Fig. 4 Emotiv Epoc+[6]

Fig. 5 Emotiv Insight[7]

Fig. 6 Simple software architecture

Fig. 7 Structure of the Connector

Fig. 8 Alternative schedule of the Connector

Fig. 9 Structure of the Main Component

Fig. 10 Back button as an option in hierarchy of command tree

Fig. 11 Diagram that is illustrating the command hierarchy[]

Fig. 12 The selection area for each division may be different, based on items that are selectable in such area

Fig. 13 OpenBCI GangLion Board[]

Fig. 14 Ultracortex “Mark IV” EEG Headset[]

Fig. 15 NueroSky - MindWave Mobile EEG Headset[]

Fig. 16 Muse: The Brain Sensing Headband[]

Fig. 17 Screenshot of the running Connector Component with debug message of broadcasting command to clients

Fig. 18 Screenshot of the server and the testing client working as expected

Fig. 19 Proposed command hierarchy tree for the system

Fig. 20 A demo application to illustrate the effect of using glfw and OpenGL for translucent effect

Fig. 21 Hierarchy of GUI Framework Components

Fig. 22 Screenshot of GUI

Fig. 23 Screenshot of GUI PointingDevice
Abbreviations

API: Application Programming Interface
BCI: Brain Computer Interface
EEG: Electroencephalography
GUI: Graphical User Interface
OS: Operating System
SDK: Software Development Toolkit
UI: User Interface
USD - United States Dollars
**Definitions**

Emotiv - A commercial company that sell BCI devices in consumer level

Emotiv Epoc - The flagship, high end BCI device sold by Emotiv

Emotiv Insight - A lower end BCI device sold by Emotiv

MindDesktop - The software application that this project based on. A BCI system for a general purpose system

OpenBCI - A commercial company that sell BCI device components and relevant products in consumer level
1. Introduction

1.1 BCI Background

Using human’s thought to control the devices has once been a scene that only appeared on the screen in the cinema. Nowadays, it is no longer an imagination. Since the start of first Brain Computer Interface (BCI) research in 1970s which is granted by the National Science Foundation in US [1], the subsequent researches in such field has contributed a lot to the society. BCI realizes a real time communication between brain and the controlling device. With the aid of Electroencephalography (EEG)-based BCI, user’s thought of control can be conveyed merely by brain signals without including those being generated by muscular activities or peripheral nerves. [2] User can operate devices to finish some task merely with thought such as turning forward direction of wheelchair, typing in the computer and controlling robotic arm[3][4][5], which is especially helpful to the paralyzed or disabled people. Figure 1 illustrates BCI for controlling a laptop.

![Fig. 1 An illustration on what is a BCI device][11]

Unlike the invasive BCIs, the non-invasive BCIs are much safer to use because it does not implant the brain signal monitoring device into the user’s skull. Among the non-invasive interfaces, EEG is the most studied one due to its portable and low-cost setup features. In recent years, several manufacturers such as Emotiv and OpenBCI have produced some headsets or components for monitoring human brainwave in the way of EEG, which give awesome performance and further push the EEG-based BCI development. Many developers or researchers have used these headsets to carry out their BCI development projects.
1.2 MindDesktop Introduction

MindDesktop, a general purpose EEG-based BCI developed by students from Software & Information Systems Engineering Department in Ben-Gurion University, has drawn our team’s attention because it gives a clean and easy control of any Windows-based system via a pointing device and virtual keyboard [5]. MindDesktop used the Emotiv EPOC headset as the brainwave monitoring device, which is a 14-channel wireless EEG headset. It allows 3 user actions to achieve the full manipulation, including zoom in, zoom out and scroll to navigate through the hierarchy. (Figure 2). The pointing device (Figure 3) divides the screen into 4 equal sections and each time user choose one section, that section is subdivided into 4 equal sections and finally focus on a single element on the desktop. Once confirm, it eventually executes the corresponding action according to that element.

Fig.2 MindDesktop Input device state hierarchy           Fig.3 MindDesktop Pointing device UI

1.3 Project Introduction

This project is based on the idea of MindDesktop and try to find a solution that can enhance the user experience and reduce the pricing threshold for consumers by using lower ends model of BCI devices.

In this report, the objective of the project will be discussed in Chapter 2, which is about improving the efficiency of using MindDesktop and makes lower end BCI devices able to control the system. Follow by the methodology in Chapter 3 where the software architecture of the new system will be explored. In order to achieve the objective, three different ideas will be explained regarding how they could improve the original system and how would they be implemented. Hardware limitations on the BCI will also be discussed as an obstacle of the project. Follow by a report of the current project status in Chapter 4 and will end with the future plan in Chapter 5 for the coming months.
2. Objective

Although MindDesktop provides a solution for using BCI in a General Purpose System, which opens a door for those who are disabled/paralyzed who wish to make use of the device for doing daily operation on a computer, there are certain areas that can be further improved to enhance the user experience and lower the threshold to use the technology in terms of budgetary.

2.1 Improving efficiency of MindDesktop

To improve the efficiency of MindDesktop, our team focus on two major areas which can have effective improvements.

First, our team recognize that there are quite a substantial amount of redundant steps by using MindDesktop to type text. These redundant steps can be eliminated by a better design of command tree, which means reconstruction of input device state hierarchy shown in figure 2. By doing this, the time for user to complete a task can be greatly shortened.

Second, the division of section for mouse selection in MindDesktop can be modified with a smarter algorithm rather than divide it equally into four sections. A faster algorithm for division will be introduced to enable selection with less steps. This will reduce the number of recursions and therefore increases the efficiency.

2.2 Enabling possibility of using lower end models

Currently, the target device model of MindDesktop is Emotiv Epoc+, which costs $799 USD[6] to date. This price tag is considered relatively high when there are models from the same company, Emotiv Insight, costs $299 USD[7] to date. In this project, Emotiv Insight will be tested and see if it is capable of controlling the system built here. By enabling the possibility of using lower end models of BCI, it will allow more end users to afford and enjoy the benefit that the system brings.
3. Methodology

In order for the system to achieve what was stated in the Objective section, there are four main focuses that will help to meet the goal. To start with, an overview of the software architecture of the system will be discussed that will explore the benefits of using a modular approach design. It will then follow by a discussion of how reducing number of input commands can help to meet the objectives and compare the level of reduction with the effect that those can bring. A redesign of the command hierarchy in MindDesktop for better efficiency by reducing the amount of commands that users would need to make for a same particular task. Finally, a study on how could a smarter division algorithm for mouse selection to be implemented will be conducted.

3.1 Software Architecture of the System

A modular approach will be used in the design of the architecture in this system rather than packing all items into a single bulky application. By using this approach, the BCI application can be divided into numbers of smaller components, each responsible for a core function or collection of related functions. It allows the BCI system to have a cleaner design, effective division of tasks for system development and easier management of programming code. Most importantly, it greatly increases the flexibility of the application in the case that one of the components can be replaced with a new one or being modified without affecting other components. As a result, modification to the application can be done with less work.

The application will be separated into three main components, namely Connector Component, Main Component and Training Component. All components will be written with the programming language - Python, for its ease of development and a large base of available libraries.

Fig. 6 simple software architecture
3.1.1 Connector Component

Connector Component will be a server application that connects the hardware, receives the signals from headset and broadcast it with a standard interface. The design should not limit to be compatible with a specific hardware. In this particular circumstances, Connector should use the Software Development Kit (SDK) provided by Emotiv and call the implemented server application functions for broadcasting. Furthermore, input from the keyboard is also going to be implemented for a controlled input source for testing the system. Websocket protocol will be used for inter-component communications for its balance between simplicity of implementation and the overhead it incurs.

Figure 7 illustrates the structure of the Connector Component. All the training data will be stored in a user profile specific to the user. After receiving the signals from the headset, the SDK provided by Emotive will then process the signals to fire an event which tells the current mental command and an index telling the possibility of such command is current wanted by the user. Here, this index is described as confidence level. After that, a piece of code that acts as analyzer will then determine whether or not to broadcast such message to the Main Component.

Numerous identical messages will be generated out at a short period of time if every events turns to messages and send to the Main Component. In this situation, user may want to execute the command once but the command is executed many times. As a result, alternative idle and active schedule is introduced. (Figure 8) In active state, after receiving three consecutive identical events with confidence level higher than the threshold, the analyzer generates a message containing the command and pass to the server section. Then the Connector is idled for one second before returning to active state again. This cycle carries on during the execution of the program.
3.1.2 Main Component
Main Component is the main application that is based on MindDesktop. A similar Graphical User Interface (GUI) will be implemented and a similar concept of mouse selection will be used with modifications details that are explained in the following sections. This component will connect to the server that Connector Component is hosted and listen for the input commands fired by the server.

Figure 9 illustrates the structure of Main Component. Inside the component, the code can be divided into two main functions. The GUI Manager is responsible for the GUI rendering through interacting with the coordinator, where the Coordinator is responsible for the coordination of GUI Manager, processing of incoming messages from Connector and the interaction with the Windows OS.
3.1.3 Training Component

Training Component is an application for users to train the user profile for Emotiv device. Currently, there is a software named Xavier that can be used for the training but it is not sophisticated for the use case and some users may find it hard to configure. Therefore, a specific application will be made for this particular purpose if there is enough time after completing other features in the project.

3.2 Reducing Number of Input Commands

One of the main problem of using lower end models BCI devices is that the accuracy of detecting EEG signals will be significantly decreased compared to expensive models due to the fact that the number of sensors in the cheaper devices are much less than those in high-end devices. Another factor that will greatly affect the accuracy of the detections is the number of commands that are registered in the user profile. Base on this, a proposed solution is to reduce the number of inputs that will be detected by the system. In MindDesktop, there are three input commands, therefore the solution will be reducing the input commands to two or one.

From preliminary exploration, according to the demo video of MindDesktop, provided by the group of members from Ben-Gurion University, the three command inputs in MindDesktop is suggested to be “Next”, “Confirm” and “Back”. The minimum input to interact with a system is two, which would be “Next” and “Confirm” while the command “Back” can be implemented by adding as an option in each level of hierarchy of the command tree, as shown in the following figure 10.

![Figure 10: Back button as an option in hierarchy of command tree](image-url)
The “Back” command can also be eliminated with an implementation of “after choosing the lowest level button, the pointer of the command tree will return to the root or appropriate level in the hierarchy”. In fact, the proposed design for this system will contain a mix and match of both methods to try maximize the efficiency.

The number of command, can be reduced to one with the command “Next” to be executed in a fixed time interval but this will slow down the selection of commands and especially when the user chose the wrong command. This essentially goes against the original objective of increasing the efficiency of the system. Therefore, the minimum number of command for efficient control is two.

3.3 Redesign of Command Hierarchy for Better Efficiency

A redesign of the command hierarchy can potentially increase the efficiency of the system a lot when consider the end user behaviors. The grouping of commands and how the command tree reset can greatly reduce the number of times that a user would need to input in order to do the same task. In particular, the command hierarchy will be designed as follow in figure 11:

![Command Hierarchy Diagram](image)

Fig. 11 Diagram that is illustrating the command hierarchy
In fact, the basic layout of the command hierarchy is similar to MindDesktop. The main difference is that in the design of this system, for every letters typed in the system, the level of the command hierarchy will return to the state of selecting letters. This implementation assumes that users will likely want to type another letter after a letter and the next letter will not in the same group of the previous letter. In that case, with this implementation, users will go through less redundant commands and essentially increase the efficiency. Similar concepts are applied to Numbers and Symbols. Each one though, will require a back option or otherwise an infinite loop will be created.

3.4 Smarter Division Algorithm for Mouse Selection

MindDesktop’s mechanism for selecting items with mouse is using a recursive selection with division of screen into four equal parts. In this system, a similar approach will be done except that the division of screen will not be equal. Main component will analyze the User Interface (UI) on screen and will try to divide the screen into portion that has equal amounts of selectable items with the information provided. The aim of this approach is to reduce the number of recursions during the selection. In the preliminary study, an Application Program Interface (API) provided by Microsoft Windows named UIAutomation will be able to provide such capability.

Fig. 12 The selection area for each division may be different, based on items that are selectable in such area
4. Project Status

To date, there are totally five sections that will be discussed in this chapter. It starts with a report on the preliminary research on different BCI devices and the choice of the device used in this project. The details implementation of Connector Component will be go through with the testing of the completed version of it. A proposed command hierarchy tree that will be used in the Main Component will be shown and explained next. After there will be a detailed report on the development of the Main Component. Lastly, the chapter will end with a discussion about the limitations of hardware that became a significant obstacle in this particular project and alternative BCI devices that are in consideration for replacement.

4.1 Preliminary Research on BCI Devices

Research on which BCI device will be the most suitable for the project has been conducted before the start of the development. Emotiv Epoc is the model that had been used for the project MindDesktop. As one of the objectives is to enable the possibility of using lower end models, the price of Emotiv Epoc has been used as reference. In other words, only commercial devices that are similar or cost less than Emotiv Epoc has been compared.

The devices that has been researched are Emotiv Insight, Muse, OpenBCI Ganglion Board with Ultracortex “Mark IV” EEG Headset (Unassembled) and NeuroSky - MindWave Mobile: Brainwave Starter Kit. As of 2017, pricing of these four devices are $299 USD[7], $249.99 USD[8], $649.98 USD[9][13] and $99 USD[10] respectively. Emotiv Insight has 5 EEG sensors with 2 reference sensors, 7 sensors in total.[7] Muse has 4 EEG sensors with 3 reference sensors, 7 sensors in total.[8][12] OpenBCI Ganglion Board has 4 channels that is equivalent to have capability of 4 EEG input sensors.[9] NeuroSky - MindWave Mobile EEG Headset has only 1 channel that is equivalent to have capability of 1 EEG input sensor.[14] From these four devices, the BCI device from NeuroSky despite being the most inexpensive one, it only capable of having 1 EEG input sensor. While Emotiv Insight being second most expensive but at a cost of half from OpenBCI Ganglion Board with Ultracortex “Mark IV” EEG Headset (Unassembled) and have the most number of EEG sensors.
Base on the price and number of sensors, Emotiv Insight and Muse were aggregated out for the choice of the project. By comparing the SDK of Emotiv Insight and Muse, it was decided that Emotiv Insight is better suit the project as it has an event called MentalCommand that allows users to register their brainwave signal to the computer first and a callback event will be fired when the computer detects the similar signal. Muse though, does not have any thing that offers similar function.
4.2.1 Implementation of Connector Component

Connector Component is implemented as a server application. Websocket library in Python has been used for making a server implementing with the websocket protocol. The particular library package for websocket can be installed via pip with the command “pip install websockets”. A broadcast function has been implemented in the server module and any message can send to all clients that are connected to the server. An abstract class represent the input device for sending the commands has been written in order to support different devices.

Currently, Emotiv Insight and keyboard are the input devices that supported by the system, while the keyboard input is purposed for controlled test to make sure the Main Component behaves correctly. Dictionary object has been used for easy selection of the input device by simply passing the key of the desire device to the function. API from the SDK provided by Emotiv is used in order to connect the device to the server (the Connector).

There were some difficulties in implementing the controlled keyboard input source; a few libraries and methods had been tried and none was able to allow the application to grab the keyboard input key while the application window is out of focus until the library named “keyboard” from the Github project “bopreh/keyboard” has been used. This library package can also be installed via pip with the command “pip install keyboard”. Having the ability to grab input globally is required as the application that takes keyboard input cannot be guaranteed that it would be the foreground application at all time. Keyboard as the input device is essential to the system as it is acted as a controlled input source for testing the main application.

![Fig. 17 Screenshot of the running Connector Component with debug message of broadcasting command to clients](image-url)
4.2.2 Testing of Connector Component

For confirming that the Connector Component is working as expected, a light-weight websocket client testing script has been written to see if the commands sending from the Connector Component can be correctly received.

The testing script simply will connect the websocket server running in the Connector Component and wait for message to be sent from the server and print the received text to the screen. The following figure is the screenshot of the server and the test client running together:

Fig. 18 Screenshot of the server and the testing client working as expected
4.3 Proposed Command Hierarchy Tree

To make MindDesktop to be more efficient, a considerate command hierarchy tree is made to try to minimize the number of inputs that the user need to execute for doing an action. The root of the command tree is the state when the GUI is hidden from the screen. The next hierarchy is when the input commands, “Keyboard”, “Mouse”, “Special” and “Exit” will be displayed on screen. In the “Keyboard” section, the input will separated into space bar and three main groups, named “Letters”, “Numbers” and “Symbols”. While user is entering “Letters”, the command hierarchy tree will stay in the level of “Letters” until they choose the back command. This allows user to enter words with much less number of times of input. Similar concept is applied to numbers. While input such as symbols are less likely to have consecutive input, the user will be directed to one upper level after finish the input.

For making a demo application to work, it is decided that the mouse input will start with four evenly distributed areas, the method mentioned before in Chapter 3 will be implemented in a latter stage. Special commands are specific for different application. Some complicated commands can potentially script as a macro command. This can hugely increase the speed of user to do an action.

Fig. 19 Proposed command hierarchy tree for the system
4.4 Development of Main Component

As the window of the application will require transparent around the elements and translucent during the mouse selection, the common GUI libraries in python such as the built in library in python, PyQt, Tkinter and wxPython will not able to perform the described effect. An alternative idea is to make a screenshot of the current screen when user using the mouse selection and overlay the GUI on the screenshot but that will produce undesired effect when there are elements on the screen is not static. Such as video will become paused and will be partially skipped when the user has made the mouse selection.

After conducting research on web, there is a working method that can make the desired effect by using OpenGL directly. In fact a library for OpenGL named glfw is used in order to make the effect. Though, the wrapper library of glfw that can be installed with pip by “pip install glfw”, in Python does not support the crucial constant APLHA_BIT_MASK for the function windowHint that enables the transparent background and translucent effect. Luckily, only small modifications need to be made in the library in order to make it works. The python library glfw does not support OpenGL commands directly, another library named pyOpenGL will need to be installed. It can also be installed with pip by the command “pip install PyOpenGL PyOpenGL_accelerate”

The python library glfw does not work directly after installation, it requires the glfw shared library to be installed in the system. As suggested by the package information page of glfw Python wrapper library, the shared library can be put into any directory of the search path in the operation system.[17] Glfw shared library can be compiled with the source code in the official website, similar to the wrapper library, the constant ALPHA_BIT_MASK is not included in the standard package and this is added for the requirement in this project.

A demo application on the described effect has been made to show that this method is working as shown below as Figure 20:

Fig. 20 A demo application to illustrate the effect of using glfw and OpenGL for translucent effect
4.4.1 Client Receiver for Commands from Connector Component
The Main Component implements the Websocket protocol to acts as a client which waits for the messages sending from the Connector Component. By setting a specific port number, the Connector and the Main components communicate through the intranet. Once receiving a “next” or “confirm” message, it calls the corresponding GUI Framework function and also interacts with the OS to do the task.

4.4.2 Implementation of GUI Framework
A self-defined GUI Framework is designed specifically for the BCI system. It is because some features that are essential to this project are not provided by the existing libraries, such as the transparency of the window. Moreover, it enables full control of the GUI design and flexibility of implementation.

The GUI Framework is completely built from the OpenGL library functions, which gives fast graphical computation. It provides functions to create different GUI components. Starting from the minimal rendering components including Panel and Label, composite components are built from these basic components in a hierarchical structure(Figure 21).

![Fig. 21 Hierarchy of GUI Framework Components](image)

Each Component in the hierarchy is a class with its own properties and functions.
Each ButtonsGroup renders elements in the same level in Figure 19. GUI of the Main Components consists of a tree of ButtonsGroup. The PointingDevice is used for emulating the mouse selection, which cover the selected area with four areas with four different semi-transparent colors, each area is actually a Panel in the under layer.

Every components contains editable attributes for customization. By initializing the components with different attributes, various styles of GUI can be built. These attributes include background color, font-color, border-color, border thickness, border radius, opacity, width, height, xy coordinates, hover color, focus color… etc. This design style is similar to the HTML style.
Other than providing editable attributes, the high level components such as ButtonGroups also allows interaction with the users. For instance, when the Main Component receive the next command, it will call the next function of ButtonsGroup, so that the focusing button will become the immediately next button. Screenshots of GUI is shown in Figure 22 and Figure 23:

Fig. 22 Screenshot of GUI buttons

Fig 23. Screenshot of GUI PointingDevice

The focusing button or focusing area will be in different color to distinguish from the normal ones. In this figure, the focusing color is red and the normal color is blue. These colors are modifiable.
4.4.3 Propagation of Keyboard Commands
Currently, the propagation of keyboard commands is simply implemented with the functions provided by the module “keyboard”. After testing, all of the keyboard commands are working as expected with the exception of the character “?”. Both functions “write” and “send” have been tested but unfortunately neither of them can input the character “?” . It is suspected that there may be some problems of the implementation of the library module; further investigation and fixes will be done in the future.
4.5 Limitations of Hardware

Emotiv Insight being a lower end model in the series with a more affordable pricing has its own drawback. The team had a hard time testing the device as the recognition of mental command was not performing very well and was exponentially difficult when the number of mental commands recorded increased. Therefore, there is a suggestion of reducing the number of mental commands to be recorded to improve the successful rate that user would be able to control the application. Furthermore, from numerous of testing, some users may find it easier to think a less abstract scene to correlate with a specific command. Training time for Emotiv Insight varies from users to users but for most of those candidates participated in this project, take a substantially long period of time before completing a training that is able to control the system.

4.5.1 Problems in Emotiv Insight

The Emotiv Insight has given an undesirable performance in terms of the signal stability and the physical contact with the EEG sensors. One of the sensor even gave no signals all the time during our test. For the remaining sensors, the signal quality are below expectation. The signal quality is fluctuating during the tests and a long period of time is required (about 20-30 minutes) is required to get relatively stable signals.

However, the Emotiv Insight device is somehow not working suddenly. It seems that the device cannot detect the brainwave signals and the signals being broadcast by the device become extremely unstable. The team can neither use the device to undergo mental command training nor perform single mental command control. In conclusion, it is assumed to be broken and is no longer feasible to be used for testing in this project.

As a result, after considering the undesirable performance of Emotiv Insight in the previous tests and the accident that the device is somehow becomes not workable, an alternative BCI device will be purchased in order to continue the projects. Investigation of other suitable devices will be discussed in the next section.

4.5.2 Alternative BCI Devices

After comparison between various BCI device in terms of their reliability, performance and allowability of raw signals accessing, the conclusion is choosing between the two OpenBCI models, Ganglion and Cyton. Ganglion is 4-channel while Cyton is 8-channel, the number of channels represents the number of sensors for brainwave detection. According to the online users’ reviews, both models have good performance in detecting brainwave signals. Though, number of channels can affects the sample quantity of collected signals per time unit so as affecting the accuracy of signal processing. Ganglion cause 300 USD less than Cyton but it has a little problem in EEG detection since there is attenuation for certain range of wave frequency. The struggle is getting a better tradeoff between performance and the price. Decision will be confirmed soon.
5. Future Plan

Although there were some problems encountered such as the GUI for the Main Component and the hardware limitations, most of the tasks are meeting their time schedule that was stated in the project plan. There are more tasks that will need to be completed in the near future of the project. Purchasing another suitable BCI device will be the immediate next task. Also, new schedule will be planned since the project is device dependant. Changing of device results in modification of the BCI system structure. After that, an implementation for the modified algorithm for mouse selection will be coded. Lastly, the chapter will end with the plan for Training Component that becomes most important part of the BCI system. Before modification of system structure, the training component is the least important and it will be explained why it becomes the most important one in the following sections.

5.1 Regulation of hardware adoption and relevant changes

As mentioned in the above limitation of hardware section, the Emotiv Insight device becomes not workable suddenly and the project will be continued with another BCI device. After some investigation on various devices and analysis of the users’ review on those devices, the team will choose between OpenBCI Ganglion model and OpenBCI Cyton Model. First of all, OpenBCI models has a relatively better reputation among the users and the performance is unquestionable. Second, it allows user to retrieve raw signals from the device without charging extra payment, so user can use these data for post-processing on their own. In contrast, Emotiv requires user to subscribe an extra scheme with extra payment in order to get the raw signals from the headset. This is a critical feature for making the decision because the team is aiming to build the entire Training Component on self-effort. Originally, the Training Component is integrated with the Emotiv SDK to use the training functions provided by the SDK. All the data processing behind those functions are unknown, the liability of the SDK is skeptical. Also, the theory behind the analysis of incoming brainwave signals to certain mental command is hidden from the user, user has nothing to do with the signal processing. It raises the team’s concern on the accuracy of the processed data output from the SDK.

As a result, the possibility of building a framework for mental command training and brainwave signals processing is now being investigated. In order to enables the training function to produce usable data for recognizing the mental commands, investigation on Deep Neural Network is conducted. Also, a supplementary application called OpenViBE is found, which can be very helpful for fast build-up of the framework. The investigation of OpenViBE and Deep Neural Network are discussed in the following sections.
5.2 Investigation on OpenViBE

OpenViBE is an open-source software platform dedicated to designing, testing and using BCIs[18]. It supports scripting by Python and provides many useful functions for Network IO, Data Visualization, Data generation, Signal processing, Machine learning… etc. By using this software platform, the framework for mental command training and processing can be speeded up.

The software supports most of the currently popular BCI hardwares, including the OpenBCI Cyton. Although OpenBCI Ganglion is not in the list of supported hardware, an extra layer of communication protocol called LabStreamingLayer can be used to retrieve signals from the Ganglion model, which means OpenViBE is applicable to both the two OpenBCI models under the team’s consideration.

In the near future, the team may try to use this software for the framework development. Studies on other useful tools will also continuously conducted to find out a rapid and efficient way to build the framework.

5.3 Investigation on Deep Neural Network

For the case of OpenBCI devices are chosen for the next device to be tested in the project and OpenVibe is not used for mental command training, deep neural network algorithms are considered to be used for such purpose. There are plenty of research and libraries of deep neural network to choose from and further investigation will require for choosing the most suitable one for the project.

5.4 Modified Algorithm for Mouse Selection

A smarter division algorithm that was suggested in section 3.4 will be implemented after the basic implementation of mouse selection works. In this part, the GUI of the division screen will be modified and more importantly, the logic for faster selection will be implemented. Microsoft UIAutomation API will be used for analyzing elements on screen.

The estimated completion date for implementing the keyboard functionality will be the end of January in 2018.

5.5 Implementing Special Commands for Specific Applications

Different applications may have some actions that are common but require multiple input commands or only easy to be accessed by mouse. This idea is from MindDesktop directly as it had implemented functions such as the action “Send” for an email application that is proofed to accelerate the efficiency of the system significantly.

The estimated completion date for implementing special commands for specific application will be 28th February 2018.
5.6 Plan for Training Component

If OpenBCI hardware is chosen as the device for further project development, Training Component will be an extremely important module of the project. As there is no mental command training software or SDK provided by the OpenBCI company, a mental command training and mental command recognizing system must be built independently. Unlike the preliminary design that Training Component is integrated with the Emotiv SDK, the new design will probably integrate with some unofficial tools such as the OpenViBE for rapid development. Investigation of signal processing algorithms and deep neural network are being conducted before the team can have a consolidate design of the training component. The inner structure of the training component is expected to be drawn out at the beginning of February in 2018 and finished roughly at the middle of March.
### 6. Project Schedule

<table>
<thead>
<tr>
<th>Component/ Major Task</th>
<th>Subtask</th>
<th>Status</th>
<th>Expected completion date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Component*</td>
<td>Integrate with Emotiv SDK</td>
<td>Obsolete</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Server Section</td>
<td>Finished</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Testing with keyboard</td>
<td>Finished</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Testing with headset</td>
<td>To be done</td>
<td>15 Mar, 2018</td>
</tr>
<tr>
<td>Main Component</td>
<td>GUI Framework</td>
<td>Finished</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Virtual Keyboard</td>
<td>Finished</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Interaction with OS</td>
<td>Finished</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Mouse Selection with smarter algorithm</td>
<td>In Progress</td>
<td>31 Jan, 2018</td>
</tr>
<tr>
<td></td>
<td>Element Inspection using Windows Automation</td>
<td>To be done</td>
<td>15 Feb, 2018</td>
</tr>
<tr>
<td>Training Component</td>
<td>To be confirmed</td>
<td>Under investigation</td>
<td>10 March 2018</td>
</tr>
<tr>
<td>Full Testing</td>
<td>Testing all components and fixing bugs found in the tests</td>
<td>To be done</td>
<td>5 Apr, 2018</td>
</tr>
</tbody>
</table>

*Modification of such module is required*
7. Conclusion

An exploration on what enhancement, mainly on the efficiency, can be made on an existing application that uses BCI devices to control a general purpose system (Microsoft Windows). Also, a study on if lower end models, which are more affordable, of BCI device can be used for replacement to enable a larger audience to use the system. In order to increase the efficiency, two approach of redesigns to the system has been made. Firstly, there is a restructure of the command hierarchy that aim to reduce the number of commands that user would need to issue for doing the same task. Secondly, a smarter algorithm for mouse selection that has the same aim for reducing the number of commands. To allow lower end models work properly in the system, the number of available input commands is reduced from three to two. This greatly decreases the difficulty in training of end users. The whole system are separated into three components, Connector Component, Main Component and Training Component. Currently, Connector Component has already been completed and tested. The development of Main Component is also almost finished. The major remaining task for the Main Component will be the element inspection of Windows.

Unfortunately, the Emotiv Insight headset is suddenly not working after the completion of Connector Component. As the BCI device is critically important to the project and Emotiv Insight does not give a pleasant performance during the early testing, the team decides to choose an alternative device possibly between the OpenBCI Ganglion model and OpenBCI Cyton Model. Although the overall architecture of the BCI system remains unchanged, the inner structure of the Connector and Training Component will be redesigned. Once being confirmed which BCI device will be used as alternative, an appropriate design will be figured as soon as possible.
References


