

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

2017-2018 COMP4801 Final Year Project Plan

Group members: Lau Chun Lam Frederick (3035123851)

Tang Wang Hin Kenneth (3035123459)

Project supervisor: Prof. Anthony Tam

Agenda

1. Background
2. Objective
3. Scope
4. Methodology
5. Possible Risks, challenges and mitigation
6. Overall Schedule and Milestones
7. Task Division timeline

1. Background

Brain Computer Interface (BCI) is becoming more common nowadays. BCI realizes a real-time interaction between the brain and the device, such as wheelchair or computer. User's thought of control is conveyed merely by brain signals without including those being generated by muscular activities or peripheral nerves.¹ Along with the advancement and gradually mature technology of using EEG headset to capture and analyze the human brainwave, many developers start to engage into the development of BCI in order to help the users to experience a more convenient control of computer, especially for the disabled/paralyzed. Since this field of technology is still in developing stage with great potential, a lot of investors are willing to invest in the researches in such field. Once there is demand in the market and money for the research, it encourages the development of various relevant products, including the hardwares and softwares, while BCI is one of the products within consideration. And multiple consumer products have been launched in recent years, it is no longer a product that is only available in laboratories. The technology is here but now the question is how mutual it is and if it is mutual enough to apply in real world applications. Ori Ossmy, Ofir Tam, Rami Puzi, Lior Rokach, Ohad Inbar and Yuval Elovici from Software & Information Systems Engineering Department in Ben-Gurion University had developed MindDesktop, a general purpose brain computer interface², to answer this question. MindDesktop is a BCI that allows the user to control the computer by using the Windows Operating System. This ignites the initial idea of this project. Notice that BCI like MindDesktop can provide such a convenient way of using the computer. Our team then wishes to create a similar system with some improvements base on the idea of MindDesktop.

¹ Wolpaw, J.R. , *Encyclopedia of Neuroscience*. New York State Department of Health and State University of New York1 ed , Vol . 1.2009.

² <https://arxiv.org/ftp/arxiv/papers/1705/1705.07490.pdf>

Commonly, BCI includes the EEG(electroencephalography) headset to interpret the brain signals into commands being input to the device and an software being installed in the device to execute this command to control the device. The implementation of BCI does not only limited to the control of computer, it could stretch to different electronic device, such as robot. However, computer undoubtedly one of the most frequently used devices and closely connected with our daily life. In addition, there had been multiple researches about the BCI implementation on computer, the relevant sources, SDK or API that can speed up our development can be found easier. So, our implementation of BCI will be on computer control and our team believes that the majority of people can be benefited from such product. Since the hardware part is not our concern within the scope, this project will focus on the software part and the headset will be bought from third party.

2. Objective

Although there are many BCIs came out in the recent years, not many of them could provide a user-friendly way of control. Under our observation, some of the BCI user interface are too complicated, user may need to call a substantial amount of unnecessary commands to finish a task, which may induce a heavy load to the brain.

Also, the number of types of command also affect the convenience or efficiency of the BCI system. Here, command means thought pattern that can be recognized by the headset being used in the system. If too many types of command are required, users need to frequently switch between different thinking patterns. It raises the difficulty of user training for accurately using all commands and also decrease the ease of control. Moreover, when the number of commands grow, user must remember all commands to successfully control the computer.

In light of these, this project aims to produce a user-friendly BCI system that allow user to control the computer barely by their thought with minimum amount of commands. This project is based on MindDekstop and aims to create a similar application but with a faster selection algorithm and user-friendly GUI. For instance, the virtual keyboard will have a hierarchy that is related to the user habit. The most frequently used group of keystrokes will place in the area allowing rapid selection. The objective of this project is that user will be able to finish a task with less time and easier control comparing to the MindDesktop. As Microsoft Windows is the most popular Operation System(OS) currently, Windows will be the target OS of this project.

3. Features

This project emphasize on producing a BCI system similar to the MindDesktop but with considerable improvement including but not limited to faster element selection algorithm and a more user-friendly GUI for the user to control their computer via the BCI. This BCI:

- Allow user to control computers running Microsoft Windows 10
- Is EEG-based
- Only accepts brain signals not produced by muscular movement or peripheral nerves as the input commands
- Help user to finish common tasks on computer barely with mind control, such as browsing on web and sending emails
- Avoid the dependency of hardware headset by creating a communication layer between the Software Development Kit(SDK) of the headset and the application, which allows feasibility of using other different headsets
- Use C and Python as the programming languages for the development
- Provides 2-3 commands for control

4. Methodology

From the preliminary research, Emotiv Insight has been found out to be most suitable BCI device for this particular project. It is the most affordable model that can detect EEG. There are other brands and models in the markets that claim that they can as well detect EEG but from reviews and reports, those devices are arguably practical for this project at all.

The application that will be developed in this project is going to be separated into two, and optionally three, main components. There will be a few advantages by doing so. Firstly, if there is a need for implementing a same purpose system (i.e. using very few inputs to control an Operating System) for another device, only additional code need to be added in this component to fit the interface, all other parts of the application will not need to be modified and will be worked straight away. Also, similar to the above fact, the testing of the application can simply change the input to some controlled inputs, such as keyboard or mouse, to ensure the application is running as expected before using the BCI device.

The first component will be a service application that use the SDK provided by Emotiv. It mainly will handle the mental command input from the BCI device and also will act as a server that will push the event in a predefined interface. By using this method, this component is providing an abstract layer between the input device and the actual application for the system. This component will be referred as **Connector** in the following content.

The second component will be the main application that will have the GUI for user to interact with the target Operation System. This component will be referred as **Main Application** in the following content.

The third component is an optional component, which will only be implemented if there is extra time for the project. This component will be a specific interface for user to train up the computer to recognize their brainwave. Emotiv Xavier has already provided an interface for user to train for the data but a more sophisticated version can be made if the time allows. This component will be referred as **Training Application** in the following content.

The server and the main logic of the Connector will be written in Python. As for using the SDK from Emotive requires to use C as the programming language to call the functions, a wrapper library can be written to speed up the later development. Websockets for Python 3 will be used as the protocol to communicate between the Connector and other components with the balance of communication overhead and ease of development.

Main Application will be written in Python as well. As for Windows Operating System, the Windows API - UI Automation will be used for inspecting the interface that the user currently looking at for making a smarter selection algorithm and ultimately decrease the number of times of recursion and improve the performance of the application. Example of what properties that the API can provide is accessible through "inspect.exe" from Microsoft Windows. The target application will only require two inputs from user, next and confirm. A proposed command tree hierarchy is shown below:

Menu

- Keyboard
 - Letters
 - Groups of letters
 - Back
 - Numbers
 - Groups of numbers
 - Back
 - Symbols
 - all punctuation symbols
 - all arithmetic related symbols
 - all shift numbers in American keyboard layout
 - Others
 - Back
- Mouse
 - Select One
 - Select Two
 - Select Three
 - Select Four
 - Cancel
- Special Command
 - Application Dependent
- Exit

By limiting to two inputs only, users will be able to train their profile in a much shorter period of time and therefore use the system more easily. Letters, numbers and symbols should be separated into smaller groups at the end for faster selection. For the exception of keyboard and mouse, every time when the user finish select a command, it should return to the top of the tree. If the user select exit on the top hierarchy, user can always open the menu back by any commands. For the case of keyboard, when user type a letter, the command hierarchy should return to under letters as user will likely type another letter for the next command. For typing numbers, it will be similar to letters. Typing other keys on keyboard will return to the hierarchy of keyboard for the same reason. Mouse will also implement a similar concept to speed up selection. To overcome the problem of there will be no back command as there is only two commands available, a back button or cancel button will be placed on such hierarchy.

For making the Training Application, Python will also be used for using the same wrapper library provided by the Emotiv SDK from the Connector. A more intuitive interface shall be made to allow user know that they are training the specific two commands for the system.

5. Possible Risks, challenges and mitigation

Risk/Challenge	Mitigation	Priority*
-Unable to call the commands accurately due to poor interpretation of brain signals by the brainwave monitoring device	<ul style="list-style-type: none"> - Have constant training, maybe 3-4 hours each week -All groupmates get the chance to train, do not put the responsibility on one person -Use keyboard/mouse input for testing before having acceptable training result 	4
-May not find the materials that is conform to the requirements of original Design, such as suitable libraries	<ul style="list-style-type: none"> -Try to find alternatives -Write an api by our own 	5
-User may not be able to use this BCI if they do not have the specific brainwave monitoring device	<ul style="list-style-type: none"> - Avoid the dependency of hardware headset by creating a communication layer between the Software Development Kit(SDK) of the headset and the application, left a room for future development to support other brainwave monitoring devices 	2

*A priority from 1 to 5. A higher number means a higher priority to solve the problem.

6. Overall Schedule and Milestones

Phase	Work	Milestone	Time
Inception	<ul style="list-style-type: none"> -Research for relevant materials eg. literature, sdk, api, brainwave monitoring devices, previous BCI researches, which platform is the most appropriate for the implementation of BCI in this Project -Setup the project webpage 	<ul style="list-style-type: none"> -Decide basic framework for the BCI system -Decide what programming languages to use -Decide which EEG headset will be used in this project -Plan the schedule of this Project -Decide the implementation platform for the BCI -Collect a list of sdk, api or library that are useful in the project 	1st Sep - 30th Sep
Elaboration	<ul style="list-style-type: none"> -Build the connector -Start to build the main application -Maintain records for the progress of project -Training of using the headset -Maintain the webpage of project -Preparing for the presentation of elaboration phase -Finish the interim report 	<ul style="list-style-type: none"> -Get the solid framework for the implementation -Obtain a workable connector -Training of using headset achieve an acceptable result, which means correctly making commands according to our thought 	1st Oct- 20th Jan
Construction	<ul style="list-style-type: none"> -Figure out which part can be improved, and repeatedly undergo a process of improvement and implementation -Build the main application and merge with connector -Test and debug -Training of using the headset -Maintain the webpage of project -Preparing for the final presentation of this FYP -Finish the final report of this project 	<ul style="list-style-type: none"> -Obtain a bug-free and fully tested version of BCI system -Training of using headset achieve an high accuracy result -Finish preparing everything for the final presentation 	21st Jan- 15th Apr

7. Task Division Timeline

Task	Subtask	Expected time required	Expected time to finish
Part A. Build connector : A server application that use the SDK provided by Emotiv	1. Design the content of the abstract layer provided by the connector	3 weeks	20th Oct
	2. Setup the server for the connector, implementation of this server application	4 weeks	23th Nov
	3. Final Testing	1 week	30th Nov
Part B. Build the Main Application running on the computer	1. Figure out the selection algorithm being used by the application	5 weeks	5th Jan
	2. Design and build the GUI of application	4 weeks	30 Jan
	3. Build the remaining part of the application	5 weeks	27th Feb
	4. Final Testing	1 week	5th Mar
Part C. Testing and Debug for the whole system		3 weeks	30th Mar
Extra Part(Optional) Build the training application		If time available	If time available