



MUHAMMAD JUNAID ASLAM

**01-134121-060**

SIKANDAR HAYAT

**01-134121-088**

# **Brain Computer Interfaced Game Controller**

**Bachelor of Science in Computer Science**

Supervisor: Dr. Imran Siddiqi (Associate Professor)

Department of Computer Science  
Bahria University, Islamabad

October 2015



# Certificate

We accept the work contained in the report titled "Brain Computer Interfaced Game Controller", written by Mr. Muhammad Junaid Aslam AND Sikandar Hayat as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by . . . :

Supervisor: Dr. Imran Siddiqi (Associate Professor)

---

Internal Examiner: Name of the Internal Examiner

---

External Examiner: Name of the External Examiner

---

Project Coordinator: Dr. Arif ur Rahman (Assistant Professor)

---

Head of the Department: Dr. Fazal Wahab

---

November 30<sup>th</sup>, 2015



# Abstract

It has been known since long that the neuron cells in the brain produce quantifiable and measurable electrical signals. Electroencephalography (EEG) refers to the measurement and recording of these electrical signals by using sensors exhibited over the scalp. EEG technology has been widely used in the medical field and is being researched to provide assistance to the handicapped individuals allowing them control over devices through brain signals. EEG signals and measurements have also been used as input to control computers. The Brain-Computer Interface (BCI) systems find applications in prosthetic devices, military and defense applications, gaming, virtual reality, robotic control and many more. Despite significant research on exploiting brain signals for these applications, such systems face a number of challenges including weak signal, delays between thoughts, problems in extracting discriminant features, correct action selection, correct action execution, similarity between signals generated by discriminant thoughts and the high cost of associated hardware. This project is aimed at developing a BCI system to control a game, a small step in the emerging field of Neuro-Gaming. We have employed the EEG device called Emotiv EPOC which is a head-mounted hardware that senses EEG signals which can be acquired through the accompanying API. The Emotiv API is interfaced with a game that utilizes on different types of actions including gestures, moods and thoughts. Using action and expression based controls, the developed system successfully provides game control through brain signals.



# Acknowledgments

We would like to thank the Computer Science Department, Bahria University, Islamabad for providing us with the budgetary support for the Emotiv EPOC Headset and Research SDK. Moreover, we broaden our appreciation towards the Emotiv Team who have built up a great device for anybody hoping to get into EEG exploration and who have addressed our questions on their forums effectively. At long last, we are obliged to thank our supervisor, Dr. Imran Siddiqi, who came up with the idea of exploring the field of Brain-Computer Interfacing and we are glad that he put confidence in us to undertake this project. His support, guidance and advice throughout the project is highly appreciated.

MUHAMMAD JUNAID ASLAM AND SIKANDAR HAYAT  
Bahria University Islamabad, Pakistan

December 2015





# Contents

<b>Abstract</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Project Background/Overview? . . . . .	1
1.2 Problem Description . . . . .	2
1.3 Objective . . . . .	2
1.4 Project Scope . . . . .	2
1.5 Organization of Document . . . . .	2
<b>2 Literature Review</b>	<b>5</b>
2.1 Related Work . . . . .	5
2.2 Methods and Techniques for BCI Design . . . . .	6
2.2.1 BCI System Design . . . . .	6
2.2.2 Training of subject . . . . .	7
2.2.3 Brain Signal Modulation . . . . .	7
2.2.4 Signal Acquisition . . . . .	8
2.2.5 Signal Processing unit . . . . .	9
2.2.6 Classification . . . . .	10
<b>3 Requirement Specifications</b>	<b>11</b>
3.1 Existing Systems . . . . .	11
3.2 Proposed System . . . . .	11
3.3 Requirement Specifications . . . . .	12
3.3.1 Interface Requirements . . . . .	12
3.3.2 Software and Development Environment Requirements . . . . .	13
3.4 Use Cases . . . . .	13
<b>4 Design</b>	<b>21</b>
4.1 System Architecture . . . . .	21
4.2 Design Constraints . . . . .	22
4.2.1 Software Requirements . . . . .	22
4.2.2 Development Environment Requirements . . . . .	22
4.2.3 Programming Language . . . . .	22
4.3 Design Methodology . . . . .	22
4.4 High Level Design . . . . .	23
4.5 Low Level Design . . . . .	29
4.6 GUI Design . . . . .	29

4.7	External Interfaces . . . . .	29
<b>5</b>	<b>System Implementation</b>	<b>37</b>
5.1	System Architecture . . . . .	37
5.2	Tools and Technology Used . . . . .	39
5.3	Development Environment/Languages Used . . . . .	39
5.4	Processing Logic/Algorithms . . . . .	39
<b>6</b>	<b>System Testing and Evaluation</b>	<b>41</b>
6.1	Graphical User Interface Testing . . . . .	41
6.2	Software Performance Testing . . . . .	42
6.2.1	Unit Testing . . . . .	42
6.2.2	Integration Testing . . . . .	43
6.2.3	Compatibility Testing . . . . .	43
6.3	Test Cases . . . . .	43
6.4	Exception Handling . . . . .	47
<b>7</b>	<b>Conclusion and Perspectives</b>	<b>49</b>
	<b>References</b>	<b>51</b>

# List of Figures

2.1	The basic work flow of mobile interface with a BCI system . . . . .	7
2.2	General steps in training a BCI system . . . . .	8
2.3	The Sensors placement on Human scalp . . . . .	9
3.1	System Use case Diagram . . . . .	14
3.2	Use Case: Controlling Game Application using Expressive Actions . . . . .	15
3.3	Controlling Game Application using Expressive Actions . . . . .	16
3.4	Use Case: Training User on Emotiv EPOC Control Panel's Cognitive Suite . . . . .	17
3.5	Training User on Emotiv EPOC Control Panel's Cognitive Suite . . . . .	18
3.6	Use Case: Controlling Game Application using Cognitive Actions . . . . .	19
3.7	Controlling Game Application using Cognitive Actions . . . . .	20
4.1	Basic System Architecture . . . . .	21
4.2	Design Methodology . . . . .	22
4.3	Context Diagram . . . . .	23
4.4	Conceptual Design . . . . .	24
4.5	Package Diagram . . . . .	24
4.6	Collaboration Diagram . . . . .	25
4.7	Sequence Diagram . . . . .	26
4.8	Deployment Diagram . . . . .	27
4.9	Module Diagram . . . . .	28
4.10	File & Directory Hierarchies . . . . .	28
4.11	UML Classs Diagram 1 . . . . .	30
4.12	UML CLass Diagram 2 . . . . .	31
4.13	UML CLass Diagram 3 . . . . .	32
4.14	GUI Design and Graphic Controls used . . . . .	32
4.15	Emotiv EPOC Control Panel . . . . .	33
4.16	Emotiv EPOC Expressive Suite . . . . .	33
4.17	Emotiv EPOC Affective Suite . . . . .	34
4.18	Emotiv EPOC Cognitive Suite . . . . .	34
4.19	Emotiv EPOC Test Bench . . . . .	35
4.20	Emotiv EPOC C# SDK . . . . .	36
5.1	C-Sharp Game Architecture . . . . .	37
5.2	Emotiv Component Architecture . . . . .	38
5.3	Integration of the Game and Emotiv EPOC components . . . . .	38
5.4	Whole System Architecture . . . . .	39

6.1	Graphical User Interface of the Game . . . . .	41
6.2	Debugging Configuration Manager . . . . .	43
6.3	Work Flow Steps . . . . .	44
6.4	Emotiv EPOC Cognitiv Suite Training Panel . . . . .	45
6.5	Exception 1 . . . . .	47
6.6	Exception 2 . . . . .	47

# List of Tables

2.1	Attributes of typical BCI systems . . . . .	9
6.1	Cognitive Actions training values . . . . .	45
6.2	Expressive Actions Test Table . . . . .	46
6.3	Hits and Misses using Expressive Actions . . . . .	46
6.4	Hits and Misses using Cognitive Action . . . . .	46



# Acronyms and Abbreviations

BCI	Brain Computer Interfacing
EEG	Electroencephalographic
SSVEP	Steady State Visually Evoked Potential





# Chapter 1

## Introduction

### 1.1 Project Background/Overview?

This project focuses on the emerging technology of Brain Computer Interfacing (BCI) and its usage in gaming. Brain Computer Interfacing is any form of technology where a machine is connected and operated by the activity of human brain. BCI examination has been motivated for a considerable length of time by the desire to give incapacitated individuals a new way of communication, hence enabling them interact with the outside world. This technology empowers those individuals who want to make interpretation of an aim into an activity despite the inability to perform physical activities [1]. Non-invasive BCI is any BCI technology which does not include physically implanting an outside gadget into the human brain. Electroencephalography (EEG) is a surely understood BCI system in which through setting anodes on the scalp it is conceivable to record the summed electrical action of the cortex. EEG-based BCI's are basically utilized and well known through different methods, in light of the fact that it has the greatest information exchange rate, up to 10-25 bits/min [2]. The BCI group perceives the requirement for frameworks that make BCI more easy to use especially by the physically impaired individuals.

BCI finds applications in a wide variety of applications ranging from device control to lie detectors. Among these, a very interesting and attractive problem is NeuroGaming. NeuroGaming is a new field of gaming which utilizes non-invasive BCI keeping in mind the end goal to enhance gameplay so clients can collaborate with a console without operating a conventional controller. NeuroGaming programming utilizes a player's brains waves/signals, heart rate, gestures and even feelings to finish a particular task [3].

## 1.2 Problem Description

This project is aimed at developing a game controller that works on the principle of Brain-Computer Interfacing using Electroencephalography technology. We are integrating the BCI technology with the gaming field. By taking BCI to the level of excitement, the inspiration for making user friendly, quicker, less expensive and open accessible frameworks will be entirely unexpected and will have a much higher need. Having the capacity to control a computer game with one's own brain represents an attractive application. A major issue however is that due to the emerging and uncommon nature of the hardware, third-party developers are hesitant in developing games for it. There are a few games developed or sponsored by the manufacturers of C-BCI technology [4]. The idea of using entertainment as a reason to spread BCI technology can indeed be motivating. The project is of great importance to people with physical disabilities, paralysis and gamers etc. Although the project demonstrates application of BCI to game control, the developed interface is quite generic and can be extended to many other applications as well, for instance, controlling of a wheel chair through brain, controlling of robot through brain and so forth.

## 1.3 Objective

The proposed project aims to develop a gaming controller using Brain-Computer Interfacing that will be used to control a single player game.

## 1.4 Project Scope

In modern day game controllers, we come across a large number of buttons intended for multiple functionalities. Our project focuses on functions related to control the movement in the game. The movement in game will be controlled through expressive as well as cognitive signals. Action/expression based signals are generated by brain as a result of some physical action of body like blink of eye, movement of head or a smile etc. Cognitive signals, which are generally very weak, are generated when brain focuses on a sequence of thoughts, e.g. 'push', 'pull', 'move-up' etc.

## 1.5 Organization of Document

This report is organized as follows. In the next chapter, we present an overview of the literature and applications related to BCI. Chapter 3 presents requirement specifications followed by system design in Chapter 4. Chapter 5 provides the implementation details

while Chapter 6 summarizes the evaluations carried out to test the system. Finally, we conclude the document in Chapter 7.



## Chapter 2

# Literature Review

This chapter presents an overview of some notable applications developed using the BCI technology. Methods and techniques for design of BCI systems are also discussed.

### 2.1 Related Work

Among well-known BCI applications, Philips et al. [4] employ EEG technology for controlling a wheelchair. The work mainly targeted physically hindered people and utilized EEG to distinguish thought examples and characterize planned activities. Moreover the final decision of motion was made by watching the environment surrounding the subject for a better avoidance of obstacles and obstruction. A similar application to control wheel chair by brain signals is presented in [5].

Similar to the control of wheel chairs, significant work has been carried out on utilizing the EEG technology to control prosthetic arms for the disabled individuals. Palankar et al. [6] presented a system that controls a '9-DOF control arm' utilizing the EEG technology. In [7], the authors utilized 'Emotiv EPOC' EEG headset for controlling a robotic arm by making the use of facial expressions. The authors concluded that while the EPOC headsets are very effective in capturing the brain EEG signals, the expressive expressions are, at times, very difficult to discriminate [1].

In addition to control of hardware, control of games and interaction with the virtual environment through brain signals has also been explored. In [2], utilization of minimal effort EEG gadgets for versatile gaming applications is proposed. The developers utilized Emotiv EPOC headset for the control of a virtual environment and recommended that characterization of proposed action of the users should be improved. Vliet et al. [8] explored

the utilization of buyer evaluation hardware utilizing SSVEP collaboration for gaming and presumed that for this application customer evaluation hardware demonstrates great results.

BCI frameworks for mobile applications have also been developed. In [6], the authors proposed utilization of remote EEG gadgets for measuring brain actions and showing the outputs using cell phones. A method where emotions are identified to help speech recognition software in cell phones was proposed. In another study, Campbell et al. [9] proposed the utilization of EEG headset for connection with a cell phone. Vourvopoulos and Liarokapis proposed the utilization of EEG gadgets for control of versatile robots in [3] and [10]. The system relies on Emotiv EPOC headset and Neurosky Mindset to control a Lego NXT versatile robot. The control of robot is carried out by means of a combination of thoughts and facial expressions.

In another system [11], the authors demonstrate the use of BCI in controlling mobile games. The developed BCI system first gets brain signals, removes the noise from raw signals and extracts a set of features. The features are then classified and used as input to control the game. Three types of responses have been considered in this work. These include P300 Event Related Potential (ERP), Steady State Visually Evoked Potential (SSVEP) and motor control. The system was evaluated on three healthy people (two females and one male) between ages of 20 to 23. The subjects had no previous experience with Brain Computer Interfacing. Each subject was given a training session and presentation about the gaming field. Figure 2.1 shows the work flow of the developed framework. The user is asked to put his attention and focus on the smart phone screen and the screen is connected with the EEG Headset [11]. Sensors are placed on the scalp of user to acquire neuro signals from brain which thus result in selection of the right fitting neuron in charge of the operation or the task user wants to perform.

## 2.2 Methods and Techniques for BCI Design

After having presented few known applications utilizing BCI, we present the general design and techniques of BCI systems in the following.

### 2.2.1 BCI System Design

The life of debilitated people can be improved by giving them the possibility of controlling different devices (wheel chairs, radio, TV, computer etc.) without physical movement. Common frameworks like Wivik and Talk Assist can be integrated with the BCI control to further enhance the effectiveness and usability of these systems [12]. The present BCI structure (BCI2000) contains four crucial segments that are source piece, pre-preparing

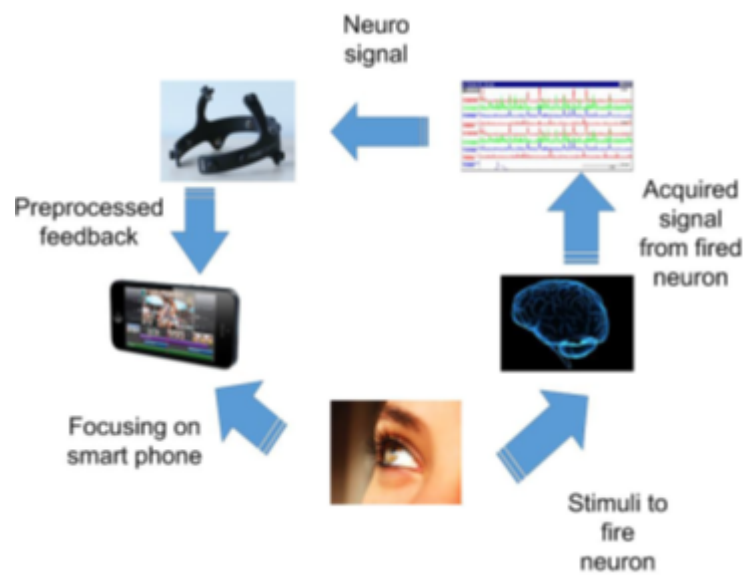


Figure 2.1: The basic work flow of mobile interface with a BCI system

square, application piece and application interface square [13]. The communication between these components is the trade of data using ‘TCP/IP’ based system framework. The ‘BCI2000’ model has a couple highlights; for instance, tradable, versatile, adaptable and real-time response.

### 2.2.2 Training of subject

A major component of any BCI framework is the training of system through the subject who is going to use the system. This training includes simple tasks such as target, selection and navigation etc. [12, 14]. This allows system to learn the specific brain signals associated with an individual and perform the desired tasks accordingly. These signals can then be processed to extract features which can be employed to train a classifier for a particular application. The major steps involved in a BCI system are illustrated in Figure 2.2.

### 2.2.3 Brain Signal Modulation

The boosts in v-BCI and a-BCI frameworks are the audio and video signals. These raw-Brain signals are affected by user’s scalp skin. These raw-Brain signals could be the result of a mild sensation or an observation etc. Sensations are results of those signals which are evoked by any change in external stimuli(also known as Sensitivity to things). These ‘evoked potentials’ (EVP) are produced by a sensation which could be the result of audio-video related brain signals [13, 15]. Recognition is the brain movement when client distinguishes, organizes, and interprets the tangible data which is going ahead at cognizant level which permits the client to sense the encompassing and environment. Additionally, the procedure of perception sensation contains learning, consideration, thinking and decision

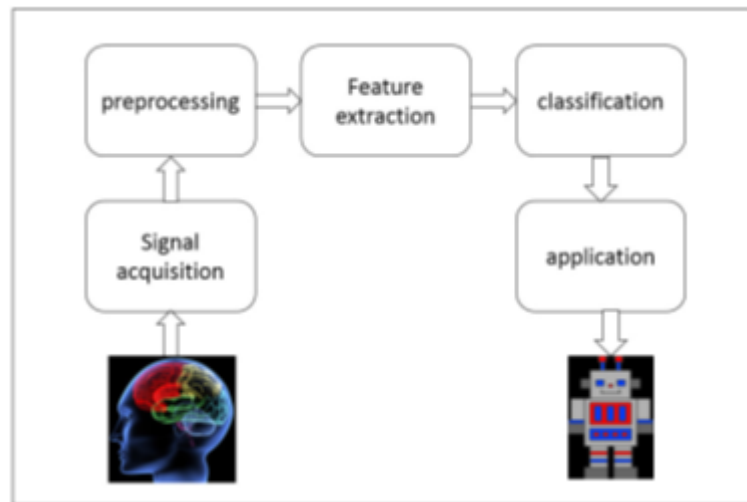


Figure 2.2: General steps in training a BCI system

making [12]. The mind signals which are utilized as a part of the a-BCI and v-BCI frameworks include the following.

- Brain signals created by outside stimulus by utilizing feature signal [16] and audio signal [12, 13, 15].
- Internal Modulated Brain Signals, for example, reaction to mental assignments, sudden reaction induced, semantic handling and feelings modulated mind signals [12].

The ‘VEP’ and ‘SSVEP’ signs are the reactions of brain to video and audio-related brain signals, having the most adequate frequency that can be recorded from back of the head area over the scalp. The subtypes of ‘VEPs’ includes in v-BCIs are as follows.

- ‘Transient VEP (TVEP)’ which happens in low-rate low frequency (raw) brain signal condition at frequency less than 2Hz.
- ‘Steady state Visually Evoked Potential(SSVEP)’ which happens at high-frequency brain signal conditions at frequency greater than 6Hz
- ‘Motion VEP’ which refers to the movement evoked potential.
- ‘Code-regulated VEP’, the arbitrary signals giving a successive production of potential in mind [11].

#### 2.2.4 Signal Acquisition

The brain signal is produced in the digital form by a signal acquisition unit [13]. A number of devices developed by various hardware vendors have been employed in different studies. All of these devices require appropriate positioning of EEG electrodes over scalp. The



subject has to wear an EEG device that measures the brain signals from all sensors on the scalp as demonstrated in Figure 2.3 [15, 17]. The EEG signals on every electrodes alludes right ear signals as ground terminal. The EEG electrodes are produced using a composition of Ag & AgCl and with a copper covering over electrodes. These sensors are placed at multiple positions over scalp. i.e. center, sides, forehead and behind ears etc. The sensors need to be properly moisturized to capture the signals. The number of sensors vary from device to device and larger the number of sensors better and detailed the acquired signals are. These sensors fire as a result of an expressive or cognitive action of brain and the resulting signals can be captured through an API in the application program.

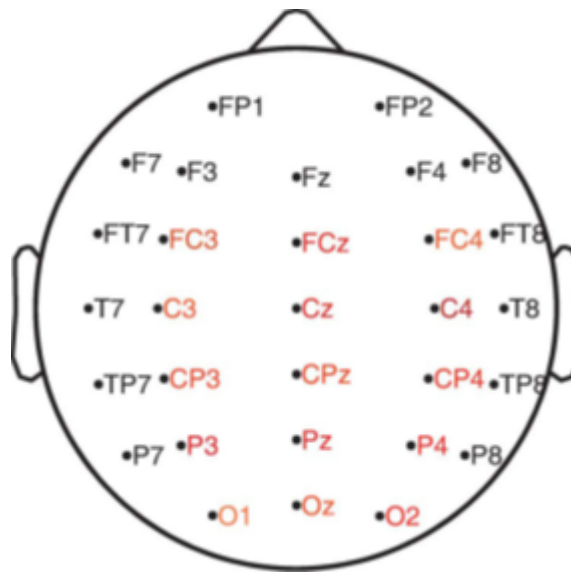


Figure 2.3: The Sensors placement on Human scalp

Table 2.1 summarizes different parameters of a typical BCI system along with the common values of these attributes.

Table 2.1: Attributes of typical BCI systems

Signals:	EEG
Number of channels:	2 channels / multi-channel
Display for stimuli:	LCD/CRT LED/Flash light
Number of targets:	Single/multiple
Classifiers:	ANN/SVM
Software:	MATLAB C++
Operation dependency:	Dependent/Independent

### 2.2.5 Signal Processing unit

The processing unit converts the EEG signal into signs that can control the output gadget connected with the BCI framework. This change can be sorted into two stages, for example,

highlighting the extraction of signals and highlighting the implementation of them [13]. Various systems can be used for removing the component vectors which include filtering of frequency, sampling, determination of channel, scaling of channels, spatial filtering and post-processing. The values of parameters for preparing the system calculations are independently chosen for different sets of data.

### **2.2.6 Classification**

Classification is carried out to attribute the received brain signal into one of the predefined classes. Classification can be carried out on raw signal or by first extracting features from the signal the later one being more effective. Combination of different features, evaluation of classifiers on individual and combined features and combination of different classifiers has been investigated to improve the classification rates. Once a given signal is classified, it can be used to implement any of the controls (moving a vehicle or wheel chair, controlling a robot, a certain move in a game etc. [18, 13, 14]).

## Chapter 3

# Requirement Specifications

This chapter first briefly introduces the existing systems followed by an overview of the proposed system. We then describe in detail the requirement specifications of the system.

### 3.1 Existing Systems

Brain computer interfacing has been attracting research attention for many years now. NASA has been utilizing brain computer interfacing in order to train their pilots to be more ready, attentive and mindful during a flight (NASA patent, dated 1994). Along with Sony, NASA utilized this technology to build up a specific play station controller that would respond to mind signals. Right brain patterns/signals were remunerated with a more responsive controller and the other way around.

In [19], a study was carried out where a group of kids with ADHD was dealt with utilizing brain computer interfacing. One half of a group was exposed to traditional brain computer interfacing and the other half utilized the play station computer game. The group playing the computer game enjoyed more and their guardians observed more change in kids as compared to those of the other group.

A number of other interesting applications developed using BCI technology to control wheel chairs, computers, robots and playing games etc. have been outlined in Chapter 2 of this document.

### 3.2 Proposed System

The proposed system aims to develop a gaming controller using Emotiv EPOC (Brain-Computer Interfaced device) and its integration with a C# game. The data is gathered from

the brain of user and issued as input in the gaming software. Data are in the form of signals which are produced from emotions, gestures and movements. The head-mounted Emotiv EPOC transmits the brain signals which are captured by the accompanying API. These signals are then interpreted to control the game.

### **3.3 Requirement Specifications**

We now formally present the requirement specifications of our system in the following sections.

#### **3.3.1 Interface Requirements**

Our system relies on brain computer interfacing and requires an appropriate hardware for reading and interpreting the EEG brain signals. Some commonly used devices for this purpose include the following.

- Emotiv EPOC
- NeuroSky Mindset
- Neural Impulse Actuator

Among these, we employ the Emotiv EPOC headset. The key features of this headset are as follows.

- 14 electrodes to collect data from brain
- Community support
- SDK available in multiple languages

The major drawback of the Emotiv EPOC headset is that it is relatively expensive when compared to other similar devices in the market.

NeuroSky Mindset is another known device for BCI. It is relatively cheaper and a number of example projects using this device are freely available. However, the sensors of this device cannot capture as many emotions as Emotiv EPOC and only a single electrode is available for data collection. Neural Impulse Actuator is the cheapest of the three. However, the SDK for this device has not yet been released making it impossible to be incorporated in custom developed projects.

### 3.3.2 Software and Development Environment Requirements

The following software/development environments are required for this project.

- A computer with Microsoft Windows operating system
- External dependencies in from of Emotiv EPOC DLL libraries
- Microsoft Visual Studio 2010
- C# SDK for Emotiv EPOC
- Emotiv EPOC control panel

## 3.4 Use Cases

The key uses cases of the system are presented in the following.

1. System Use Case Diagram: The system Use case Diagram is given in Figure 3.1 and each use case is distinctly discussed later.
2. Use Case 1: This first use case is controlling the game application using expressive actions as illustrated in Figure 3.2. The detailed description of the use case is presented in Figure 3.3.
3. Use Case 2: The second use case involves training the user on Emotiv EPOC Control Panel's cognitive suite. The generated training files are then used to control the game through cognitive actions. The corresponding use case diagram is presented in Figure 3.4 while Figure 3.5 provides a descriptive overview of the use case.
4. Use Case 3: The third use case involves controlling the game application using cognitive actions. The user is asked to focus thoughts on a particular action and the generated brain signals are used to control the game. This use case makes use of the training files generated by the training use case. The respective use case diagram is presented in Figure 3.6 while the accompanying details are presented in Figure 3.7.

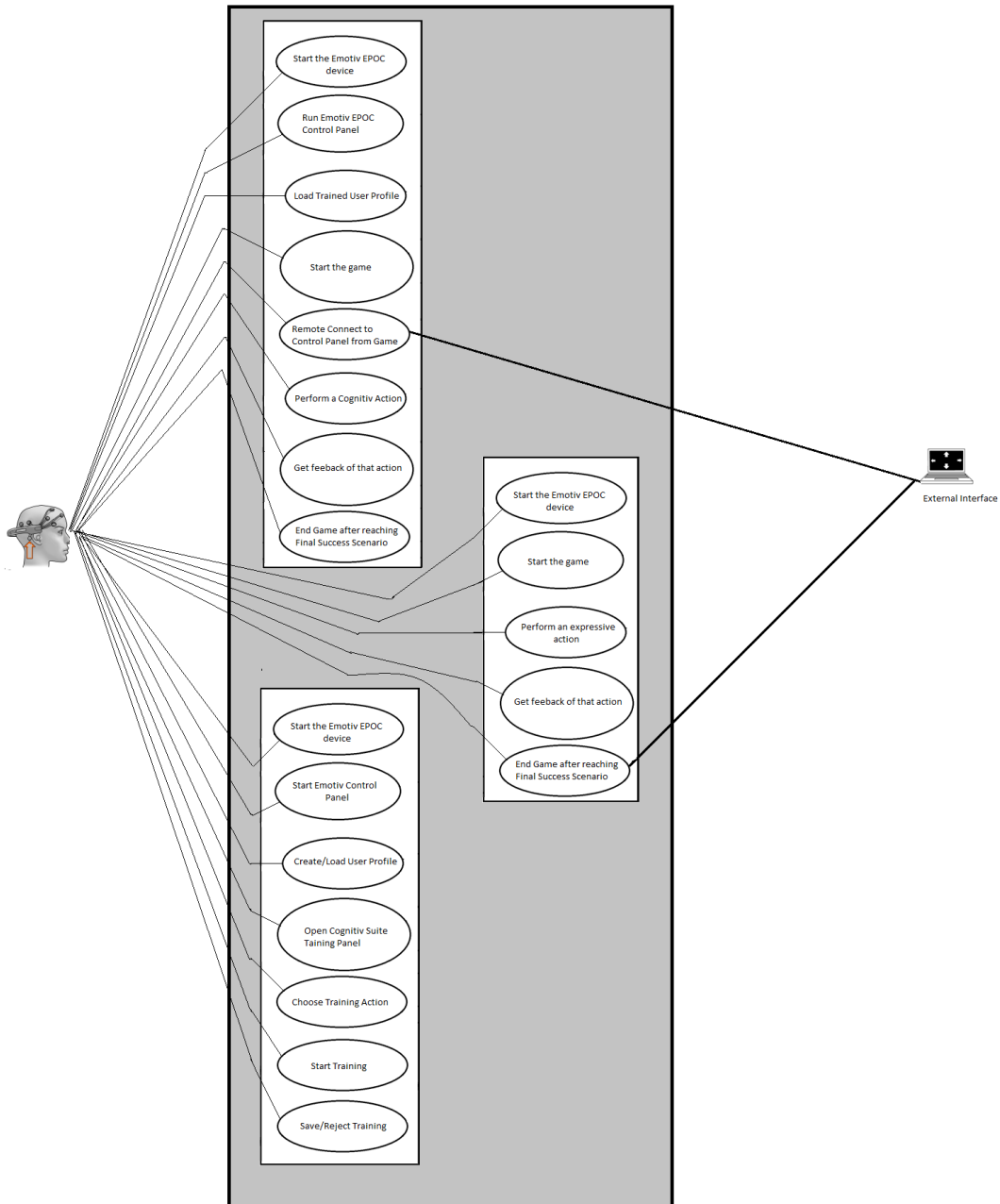


Figure 3.1: System Use case Diagram

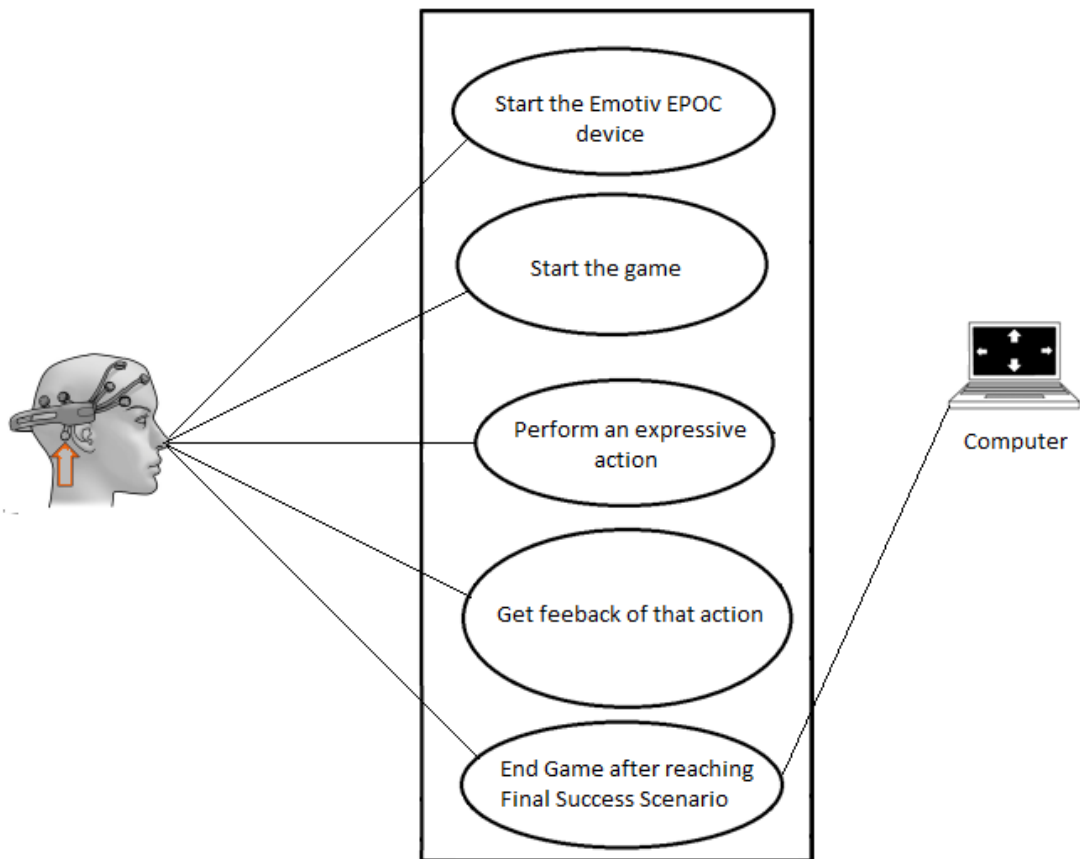


Figure 3.2: Use Case: Controlling Game Application using Expressive Actions

<b>Title</b>	Controlling Game Application using Expressive Actions	
<b>Version No.</b>	1.0	
<b>Actors</b>	Users	
<b>Description</b>	This use case represents interaction of user with computer game using expressive actions (winking, smiling, looking etc.).	
<b>Trigger</b>	This use case will be triggered once the game is on.	
<b>Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1.	Doing an expressive action as input for slider to move.
	2.	Reaching the highest score possible.
<b>Alternate Flows</b>	<b>Step</b>	<b>Action</b>
	1.	The device and game has started.
	2.	The user will make a move by giving input from brain in the form of expressive actions.
	3.	The signal/action will be processed using EEG Device SDK.
	4.	The game acts upon that particular move.
	5.	The game progresses after reaching success scenarios.
	6.	Game ends after reaching the final success scenario.
<b>Special Requirements</b>	a) User should be stress free. b) The device sensors should be properly moistened.	
<b>Assumptions</b>	No.	
<b>Pre-conditions</b>	The user should be wearing the Brain Computer Interfacing EEG Device.	
<b>Post-conditions</b>	The EEG device should be connected to PC	
<b>User interface</b>	C# Game	
<b>Issues</b>	The device sensors get dry after sometime and should be re-moistened to ensure effective reception of brain signals.	

Figure 3.3: Controlling Game Application using Expressive Actions



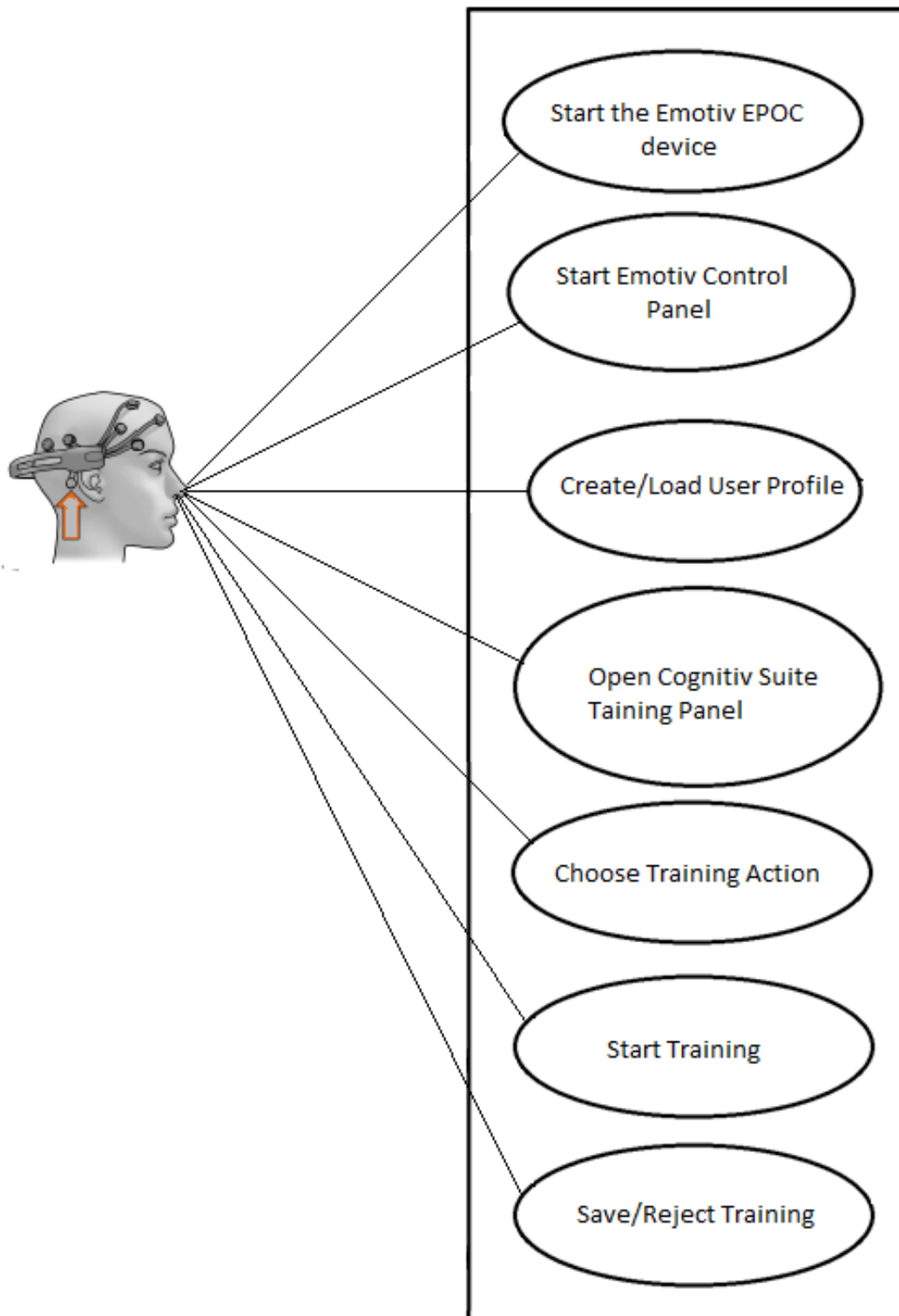


Figure 3.4: Use Case: Training User on Emotiv Epoc Control Panel's Cognitive Suite

<b>Title</b>	Training User on Emotiv Epoc Control Panel's Cognitive Suite.	
<b>Version No.</b>	1.0	
<b>Actors</b>	Users	
<b>Description</b>	This use case represents training user on the Cognitive Suite of Emotiv Epoc Control Panel.	
<b>Trigger</b>	This use case will be triggered before running the game and once the user chooses training action and presses 'Start' button in training panel.	
<b>Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1.	Open the Cognitive Suite of Emotiv Epoc Control Panel.
	2.	Choose training action from Training panel in the Cognitive Suite.
	3	Start training using the 'Start' button.
<b>Alternate Flows</b>	<b>Step</b>	<b>Action</b>
	1.	The user profile is loaded.
	2.	The Emotiv Epoc Control Panel is active.
	3.	The user chooses Cognitive Suite tab in Control Panel.
	4.	The User chooses 'Training action'.
	5.	The User starts training.
	6.	The user Ends training.
	7.	The user trains for 8 seconds.
	8.	The user saves/reject training.
<b>Special Requirements</b>	<ul style="list-style-type: none"> <li>a) The training should only be saved if the signals strength was appropriate.</li> <li>b) User should be stress free.</li> <li>c) The device sensors should be properly moistened.</li> </ul>	
<b>Assumptions</b>	No.	
<b>Pre-conditions</b>	The user should be wearing the Brain Computer Interfacing EEG Device.	
<b>Post-conditions</b>	The EEG device should be connected to PC	
<b>User interface</b>	Emotiv EPOC Control Panel	
<b>Issues</b>	Training may have to be repeated many times to get better results. The device sensors get dry after sometime and should be re-moistened to ensure effective reception of brain signals.	

Figure 3.5: Training User on Emotiv Epoc Control Panel's Cognitive Suite

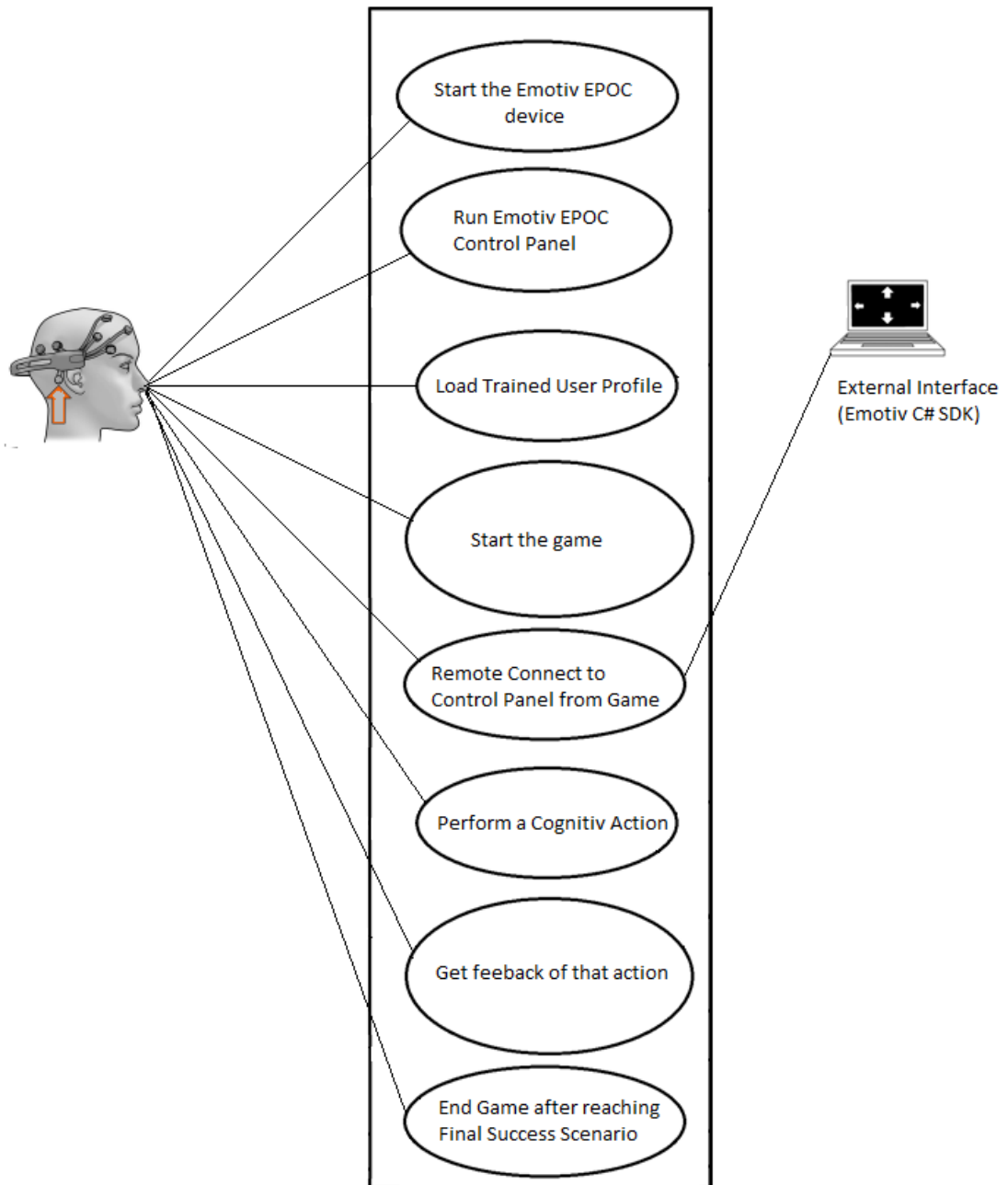


Figure 3.6: Use Case: Controlling Game Application using Cognitive Actions

<b>Title</b>	Controlling Game Application using Cognitive Actions	
<b>Version No.</b>	1.0	
<b>Actors</b>	Users	
<b>Description</b>	This use case corresponds to the interaction of user with computer game using Cognitive actions (thinking pushing, pulling, moving etc.).	
<b>Trigger</b>	This use case will be triggered once the game is on.	
<b>Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1.	Using the trained cognitive actions as input to game
	2.	Reaching the highest score possible.
<b>Alternate Flows</b>	<b>Step</b>	<b>Action</b>
	1.	The device and game has started.
	2.	Connect the device to Control Panel.
	3.	The user will make a move by giving input from brain in the form of cognitive actions.
	4.	The signal/action will be processed using EEG Device SDK in accordance with the training data.
	5.	The game acts upon that particular move.
	6.	The game progresses on after reaching success scenarios.
	7.	Game ends after reaching the final success scenario.
<b>Special Requirements</b>	<ul style="list-style-type: none"> <li>a) User should be stress free.</li> <li>b) The training should have been done correctly.</li> <li>c) The device sensors should be properly moistened.</li> </ul>	
<b>Assumptions</b>	No.	
<b>Pre-conditions</b>	The user should be wearing the Brain Computer Interfacing EEG Device. The EEG Device control panel should be connected with Game. The user should have done training on Cognitive Suite.	
<b>Post-conditions</b>	The EEG device should be connected to PC	
<b>User interface</b>	C# Game	
<b>Issues</b>	Improper training data may lead to delays in processing inputs. The device sensors get dry after sometime and should be re-moistened to ensure effective reception of brain signals.	

Figure 3.7: Controlling Game Application using Cognitive Actions

# Chapter 4

## Design

### 4.1 System Architecture

In our system, brain signals represent the input to the application. These signals are acquired from user using an EEG device. The signals are then processed using the external interfaces and they are used to control the game. The overall system architecture of the application is illustrated in Figure 4.1.

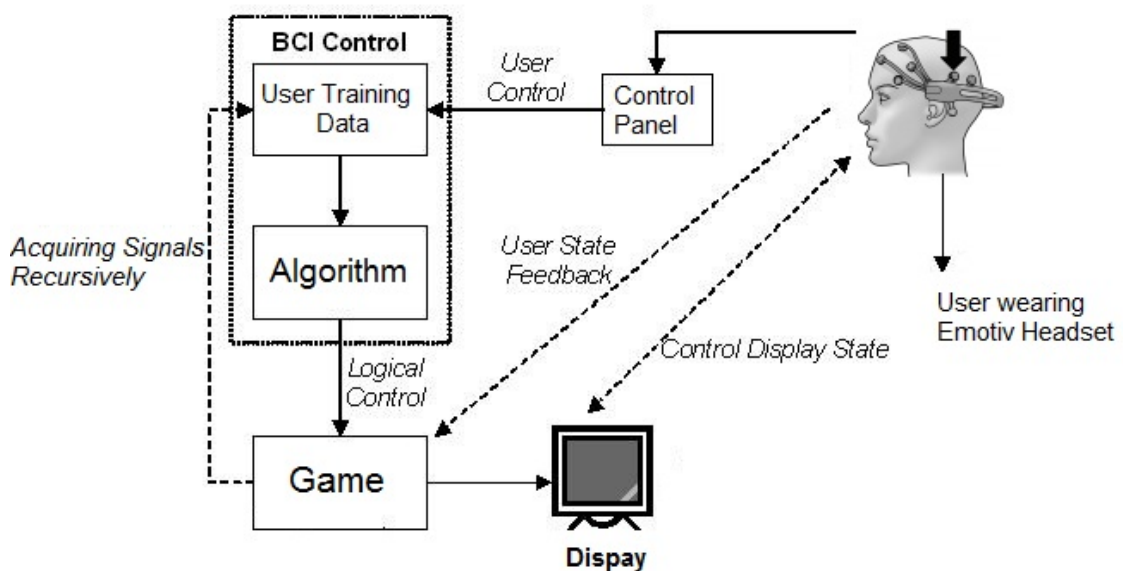


Figure 4.1: Basic System Architecture

## 4.2 Design Constraints

### 4.2.1 Software Requirements

- A computer with Windows Operating system
- Emotiv EPOC Software

### 4.2.2 Development Environment Requirements

- Microsoft Visual Studio
- Software Development Kit of Emotiv EPOC
- Test Bench of Emotiv EPOC

### 4.2.3 Programming Language

- C# Programming Language

## 4.3 Design Methodology

The design methodology of the system is summarized in Figure 4.2 while Figure 4.3 shows the context diagram of the system.

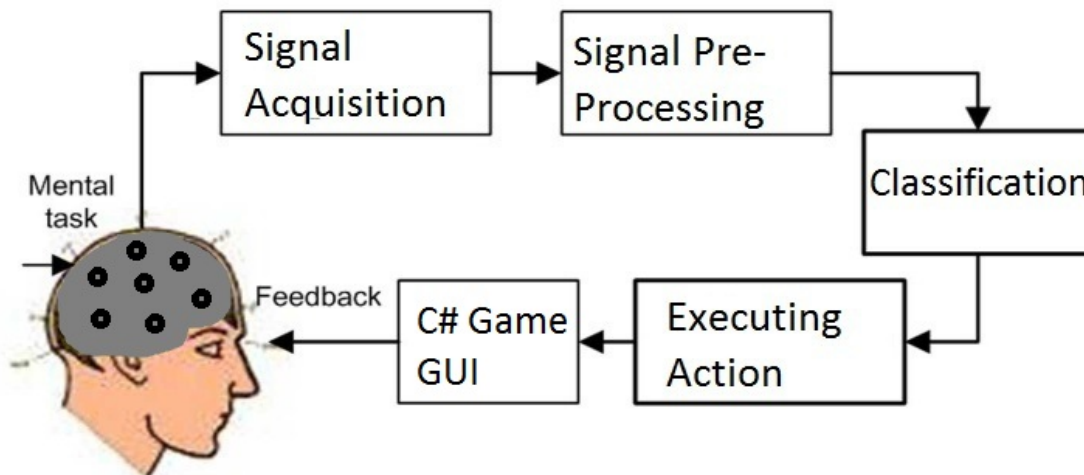


Figure 4.2: Design Methodology

Our EEG-BCI system design relies on the interfacing between C# game, Emotiv C# SDK and Emotiv EPOC Control Panel Expressive and Cognitive Suites. The expressive actions are signals which have higher frequency as compared to the other signals. While playing a game, if user performs an expressive action, the output is displayed at run time.

For cognitive thoughts, the user has to train the system first and based on that training data the user will control the game later. The user has freedom of saving, erasing and rejecting the training. Multiple trainings guarantee more effective learning and consequently resulting in better gameplay experience.

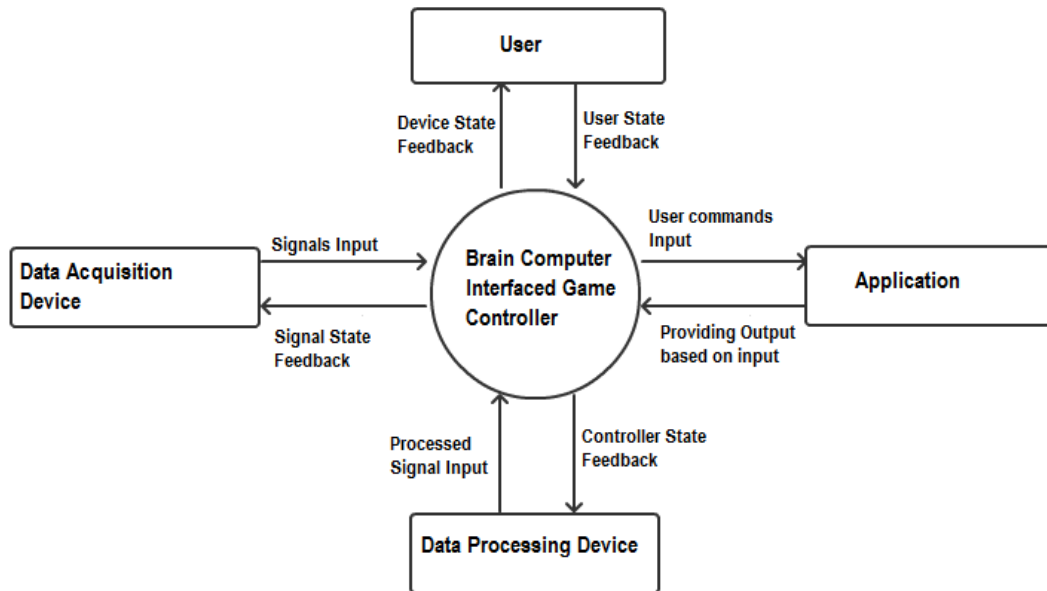


Figure 4.3: Context Diagram

EEG-BCIs can be named either synchronous or offbeat. The PC drives Synchronous frameworks by giving the client a prompt to perform a certain mental activity and after that recording the client's EEG signals in a fixed time-window. Asynchronous systems are driven by the client and work by latency and persistency, checking the client's EEG information and endeavoring to group it on the fly. Synchronous BCI systems are far less demanding to build and have truly been the essential method for working BCI systems [20]. There are four units in the context diagram (Figure 4.3). These include Data Acquisition Device which is the Emotiv EPOC, the Data Processing Device which is a computer using Emotiv EPOC C# SDK and the User controls the C# Application.

## 4.4 High Level Design

### 1. Conceptual Design

Most BCI frameworks work using the EEG data collected from the EEG device and then extracting features and using them to achieve an EEG based goal. For our system, we are collecting data using the Emotiv Epoc- EEG device. We then train our user onto some actions and save training data, then further utilize that training data in our application/game. The training translation algorithm is used to connect

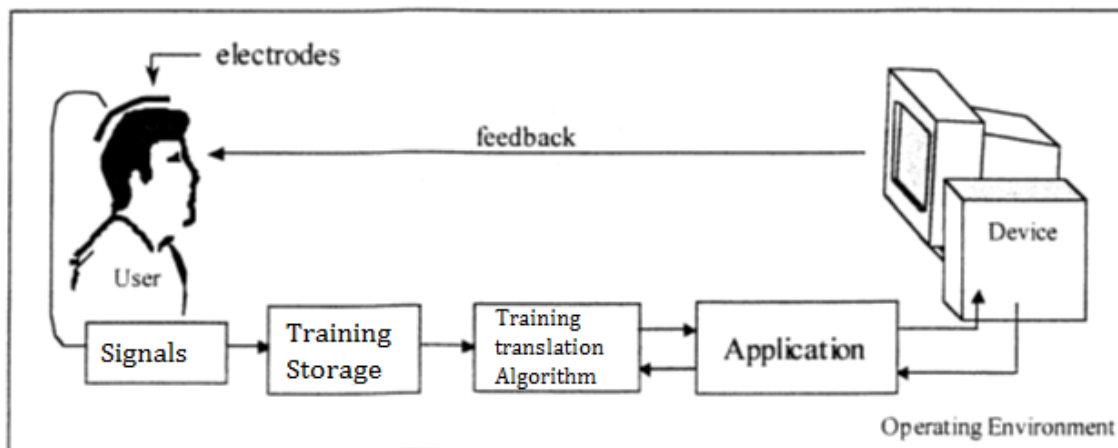


Figure 4.4: Conceptual Design

the application and the training data. The user gets a continuous feedback of the actions result. Figure 4.4 shows the conceptual design of the project.

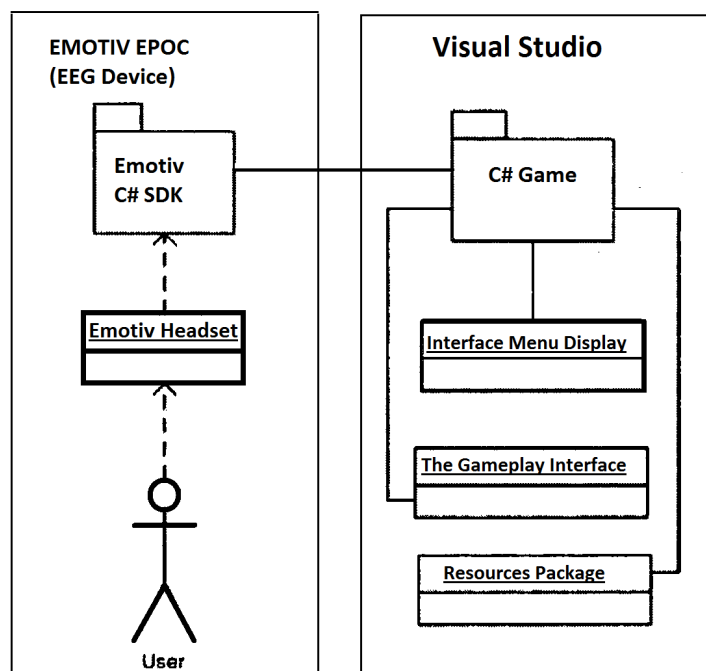


Figure 4.5: Package Diagram

The package diagram specifies the main packages and sub-packages of our system (Figure 4.5). Emotiv EPOC comes with a C-Sharp SDK package which has further classes in it. The Visual Studio package shows the C-Sharp game package further divided into Resource package (containing icons & pictures), Gameplay interfaces



containing two classes for game codes and References package including external/internal dependency files. The Emotiv Package is interfaced to Visual Package using the C-Sharp SDK and game.

2. Process Design: The process design provides a generic view of run-time integration of Emotiv EPOC and game using C-Sharp SDK. The process design is summarized using the interaction diagrams which comprise collaboration and sequence diagrams.

The collaboration diagram (Figure 4.6) of our system illustrates the running processes, starting with the Emotiv EPOC Control Panel. The system is trained using the Affective and Congitiv Suites. The training data is then accessed using Emotiv EPOC and is used as input to the game. Form-1, on execution, initializes its components. On the basis of processed inputs the player on Form-1 moves accordingly. The SettingsForm provides user the options to customize the layout of the form. The Timer control event handlers are the main components of the game and are connected with the Form-1, Emotiv EPOC event handlers and the pictures. The movement in game is based on time and the brain event handler is called recursively using the timer.

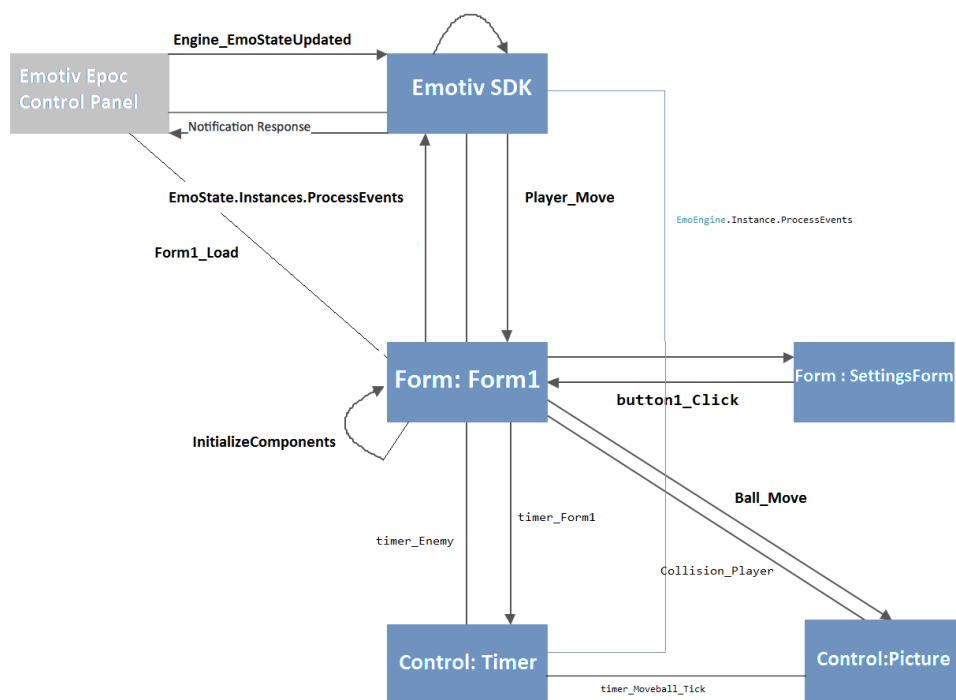


Figure 4.6: Collaboration Diagram

The sequence diagram (Figure 4.7) shows the underlying structure and sequence in which the program executes. Once the Form-1 is executed, it is loaded using `Form1_Load()` event handler, initializes its predefined components using `InitializeComponents()` and then calls the timer events `timer\MoveBall\_tick()`, `timer1\_tick()`, `timer\_Move\_tick()` and `timer\_sec\_tick()`.

Every timer event has `EmoEngine.Instance.ProcessEvents()` variable whose values gets overwritten every second. After every second the `engine\_EmoStatusUpdated` event handler is called in which the required brain state is acquired and used as input in Form-1.

The SettingsForm after loading initializes its predefined components using `InitializeComponents()` and at button click it calls the `button1_Click()` event handler, using the global variables it sets object properties values in Form-1. The `ApplySettings` function then uses those values and changes them accordingly.

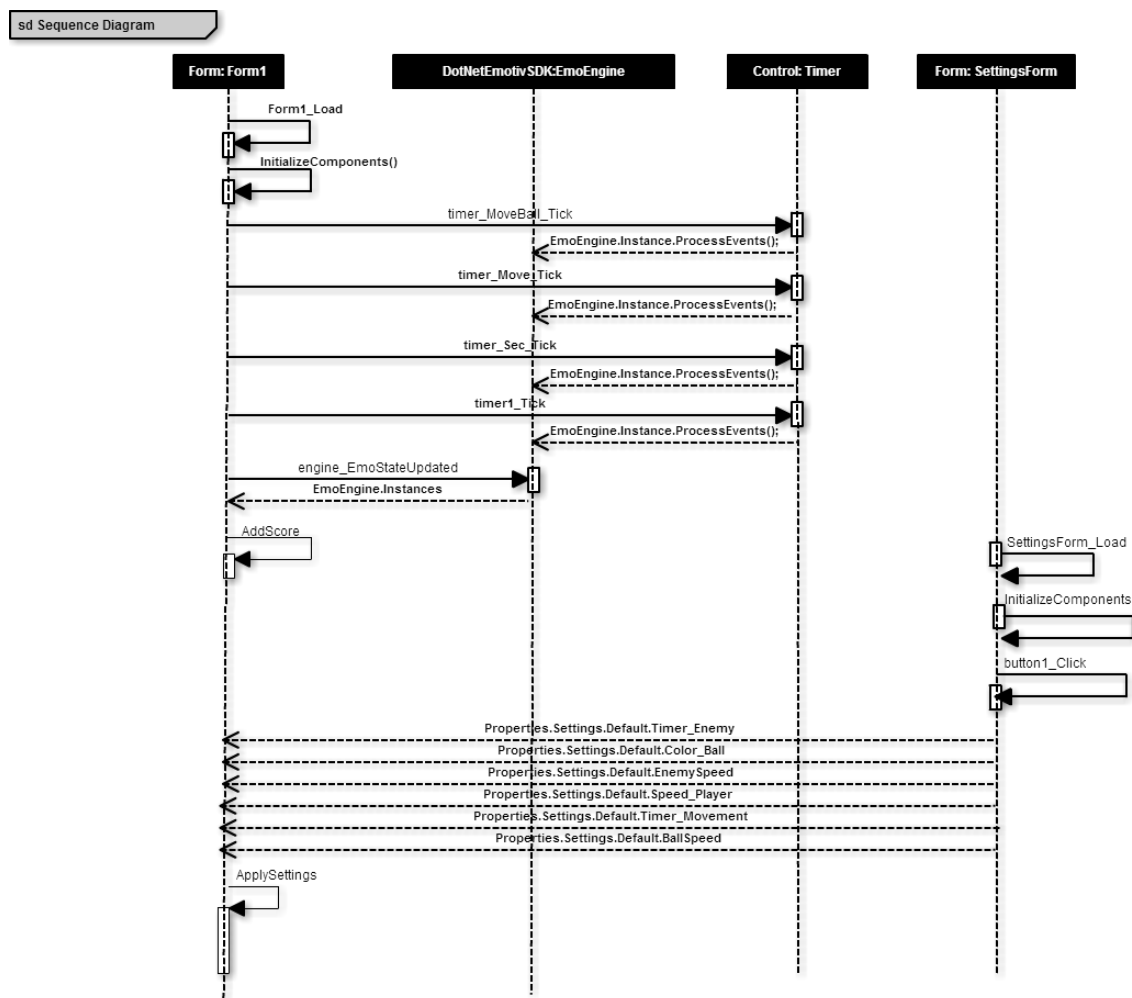


Figure 4.7: Sequence Diagram

- Physical Design: The physical design contains the physical objects of this project which are the EEG Device, its SDK and the C-Sharp game. The deployment diagram of our system (Figure 4.8) shows that the user provides the brain signals using the EEG device Emotiv EPOC which is connected to a Computer (PC). The PC has an Emotiv SDK and a C-Sharp game. The C-Sharp application and EEG device are connected to SDK, which serves as a bridge.

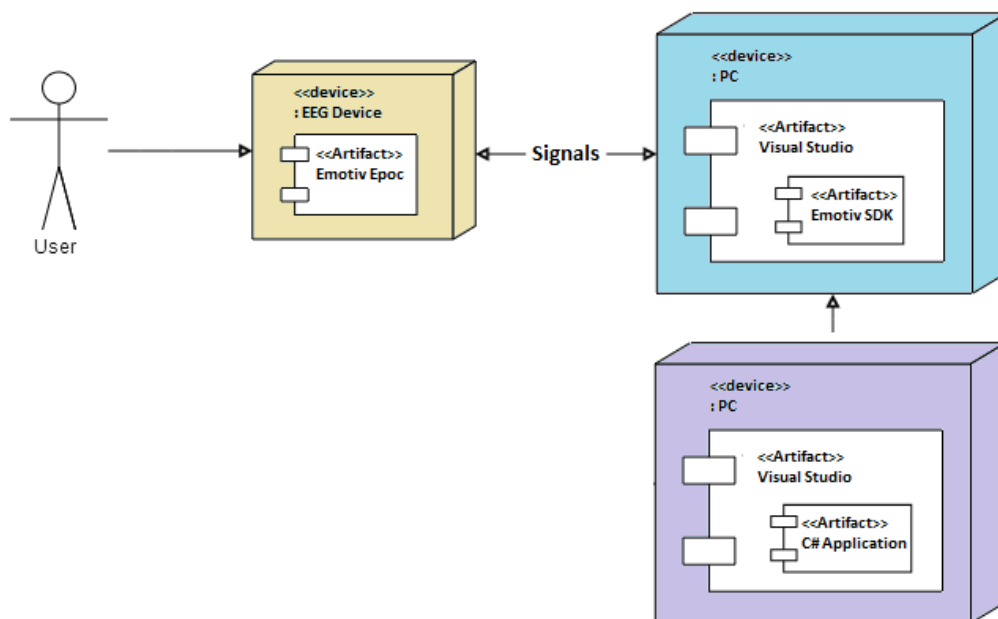


Figure 4.8: Deployment Diagram

#### 4. Module Design:

The module design illustrating the different modules of the system is presented in Figure 4.9 while the file and directory structure of the project is shown in Figure 4.10. The project comprises 'properties', 'bin', 'debug' and 'obj' folders. It contains four '.cs' source files. The external dependencies are also added as 'dll' files. The external Emotiv related files include `edk.dll`, `EdkUtils.dll` and `edk_utils.dll`. There are two C-Sharp Forms and their designer files on which the game objects are placed. The 'debug' folder contains the `DotNetEmotivSDK.dll` file and `DotNetEmotivDonglDriver.dll` file. The 'properties' and 'references' folder contain the pre-defined files which are added when the C-Sharp Form applications are created in default mode.

- Security Design: Security desing is not applicable Not Applicable to our project.

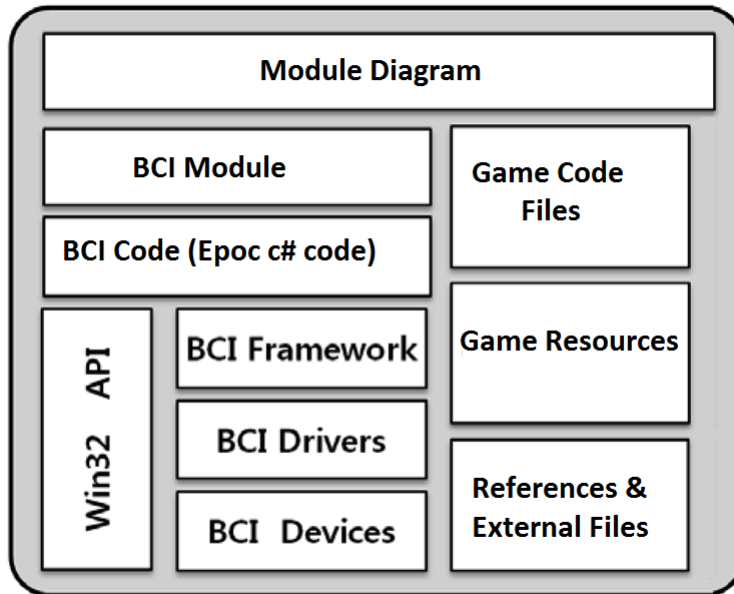


Figure 4.9: Module Diagram

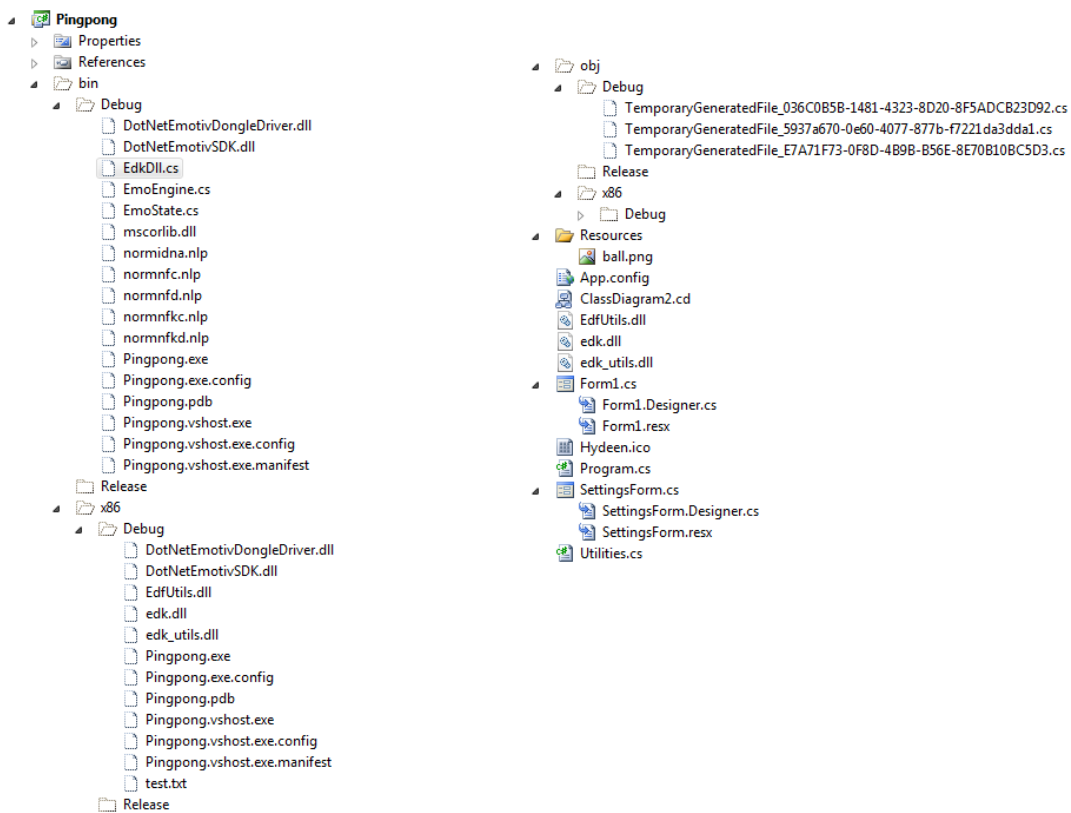


Figure 4.10: File & Directory Hierarchies

## 4.5 Low Level Design

A number of classes have been designed for the project. The UML class diagram of the system (Figure 4.11, Figure 4.12 and Figure 4.13) summarize these classes.

## 4.6 GUI Design

The GUI design of the game is shown in Figure 4.14. The game has a PictureBox controls acting as paddles for both player and enemy. There is a tool strip Menu which has Picture Boxes showing lives left of both player and enemy. The Ball is also a Picture Box. All of these objects are enclosed in a Panel. The Labels are used to show the time and other messages to user.

## 4.7 External Interfaces

As already mentioned, the project is developed using an EEG device hence it requires appropriate SDKs/APIs. The external interfaces that will be used in this project include the following.

- Emotiv EPOC Control Panel
- Emotiv EPOC Test Bench
- Emotiv EPOC C-Sharp SDK

### **Emotiv EPOC Control Panel:**

The Emotiv Control Panel as illustrated in Figure 4.15 is a graphical-client interface which works as a passage to utilizing the EPOC headset. It administers uniting with the headset, preprocessing and grouping the info signs, offering input to the client, and permits the client to make a profile and prepare considerations and activities. The control panel incorporates the Expressive, Affective, and Cognitive suites and also a Mouse Emulator which permits the client to control the mouse by moving their head and using the headset gyroscope.

Emotiv EPOC has the following three different detection suites.

1. **Expressive Suite:** The Expressive suite (Figure 4.16) is utilized for distinguishing client's outward appearances. It can recognize the flicker, right/left blink, right/left eye turns, raise of brow, wrinkle temples, grin, hold, right/left smile and giggle. These expressions are recognized by the EEG sensors getting signs to facial muscles,

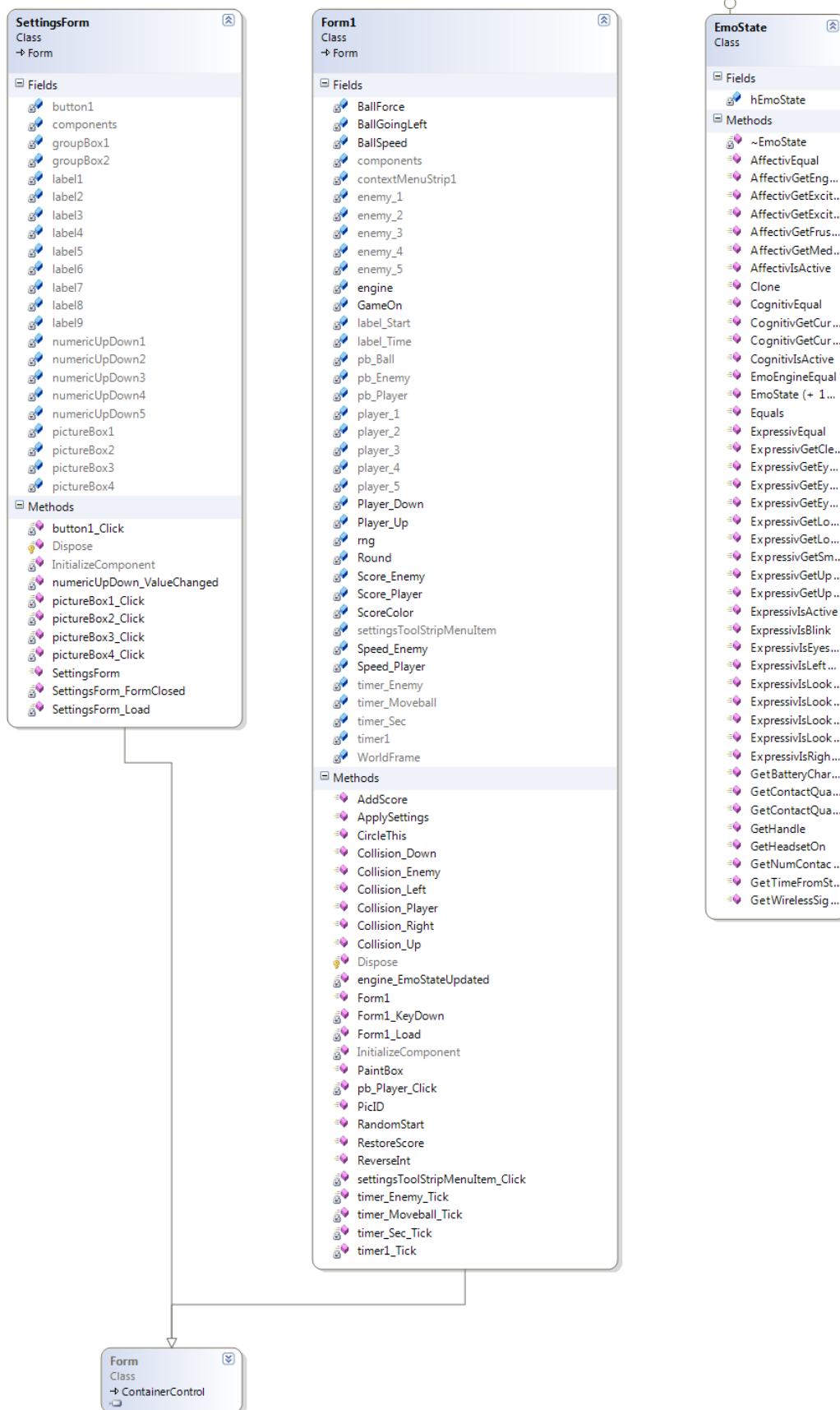


Figure 4.11: UML Class Diagram 1

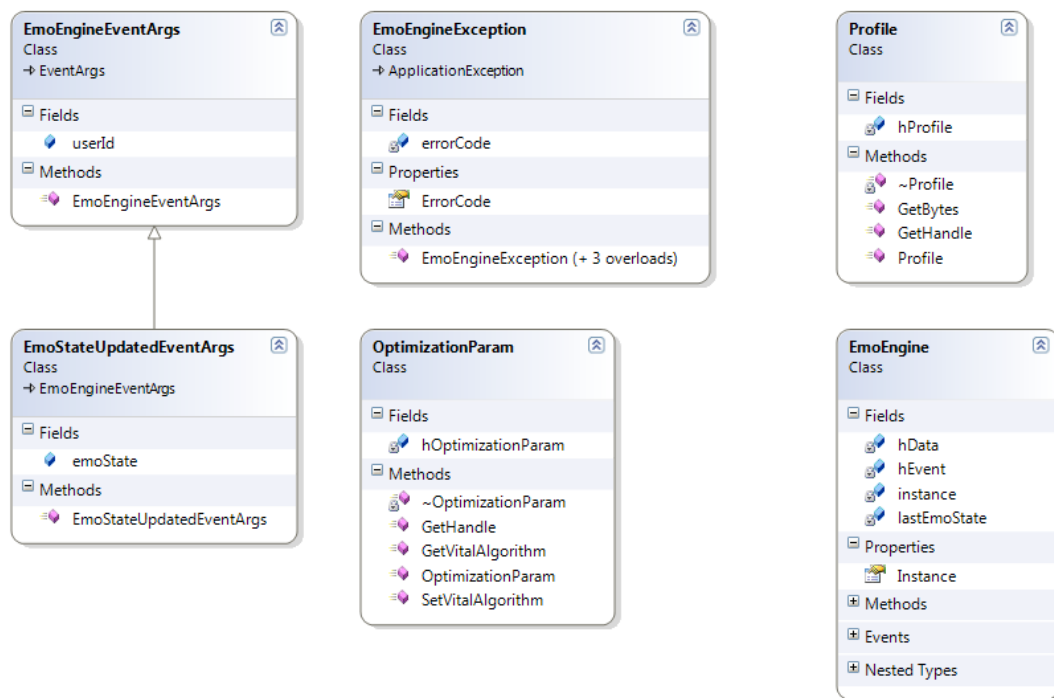


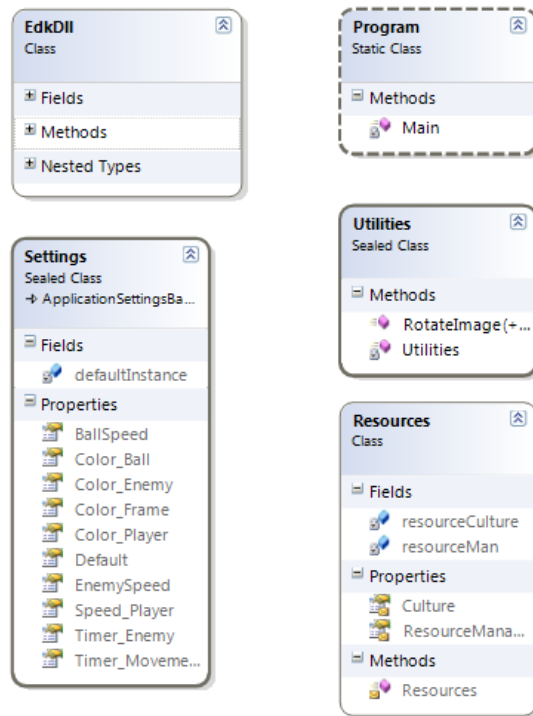
Figure 4.12: UML Class Diagram 2

instead of by perusing brainwaves. Dissimilar to perusing mental action, these location are quick verb(10ms).

2. **Affective Suite:** The Affective suite (Figure 4.17) is utilized for reading so as to check subject's present enthusiastic state. It can identify mental states like engagement/weariness, disappointment, contemplation, momentary and long term excitement.
3. **Cognitive Suite:** The Cognitive suite (Figure 4.18) is utilized for identification of cognitive thoughts by perusing subject's brainwaves. It has the capability of identifying thirteen diverse thought designs including push, draw, lift, drop, move right/left, pivot right/left, turn clockwise/counter clockwise, turn forward/opposite and vanish. The recognition of thoughts is the most challenging and complex operation due to very minor signal changes in different thoughts.

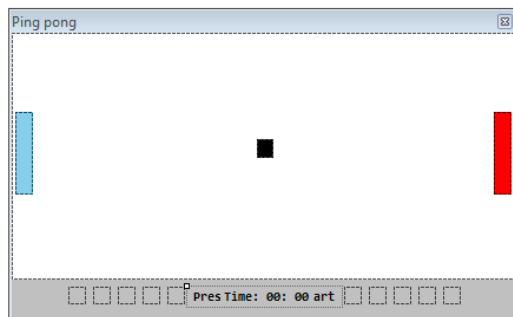
### Emotiv EPOC Test Bench:

The Emotiv TestBench interface (Figure 4.19) provides an ongoing showcase of the Emotiv headset signals stream. It permits the client to see the EEG contact quality and real information rolling in from every sensor. In addition, the gyroscope information, wireless packet data, and battery life is also shown. Besides, it can show a Fast Fourier Transform verb(*FFT*) of any approaching channel and can show the Delta, Theta, Alpha, and Beta



[!hbp]

Figure 4.13: UML Class Diagram 3



**Controls Used:**

- contextMenuStrip1 System.Windows.Forms.ContextMenuStrip
- contextMenuStrip1 System.Windows.Forms.ContextMenuStrip
- enemy\_1 System.Windows.Forms.PictureBox
- enemy\_2 System.Windows.Forms.PictureBox
- enemy\_3 System.Windows.Forms.PictureBox
- enemy\_4 System.Windows.Forms.PictureBox
- enemy\_5 System.Windows.Forms.PictureBox
- Form1 System.Windows.Forms.Form
- label\_Start System.Windows.Forms.Label
- label\_Time System.Windows.Forms.Label
- pb\_Ball System.Windows.Forms.PictureBox
- pb\_Enemy System.Windows.Forms.PictureBox
- pb\_Player System.Windows.Forms.PictureBox
- player\_1 System.Windows.Forms.PictureBox
- player\_2 System.Windows.Forms.PictureBox
- player\_3 System.Windows.Forms.PictureBox
- player\_4 System.Windows.Forms.PictureBox
- player\_5 System.Windows.Forms.PictureBox
- settingsToolStripMenuItem System.Windows.Forms.ToolStripItemMenuItem
- timer\_Enemy System.Windows.Forms.Timer
- timer\_Moveball System.Windows.Forms.Timer
- timer\_Sec System.Windows.Forms.Timer
- timer1 System.Windows.Forms.Timer
- WorldFrame System.Windows.Forms.Panel

Figure 4.14: GUI Design and Graphic Controls used

groups and a client characterized custom band. At last, the TestBench can record, recover, and playback information in European Data Format verb(.edf) and can change over spared EDF information to Comma-isolated Value position verb(.csv) in an Excel file.



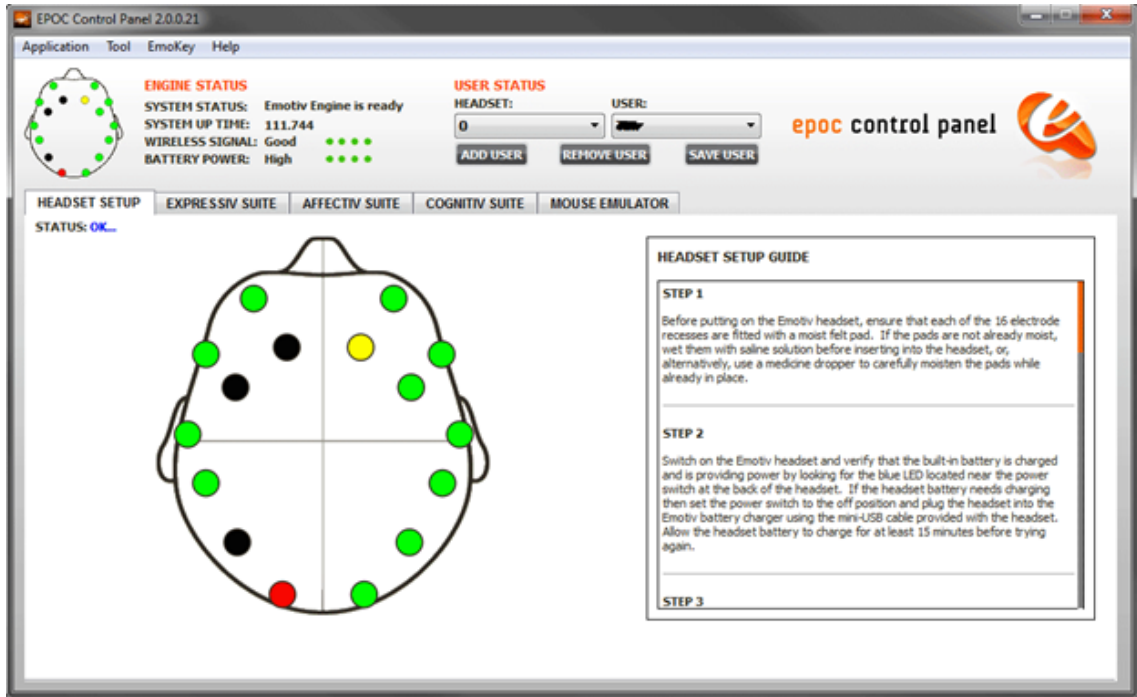


Figure 4.15: Emotiv EPOC Control Panel

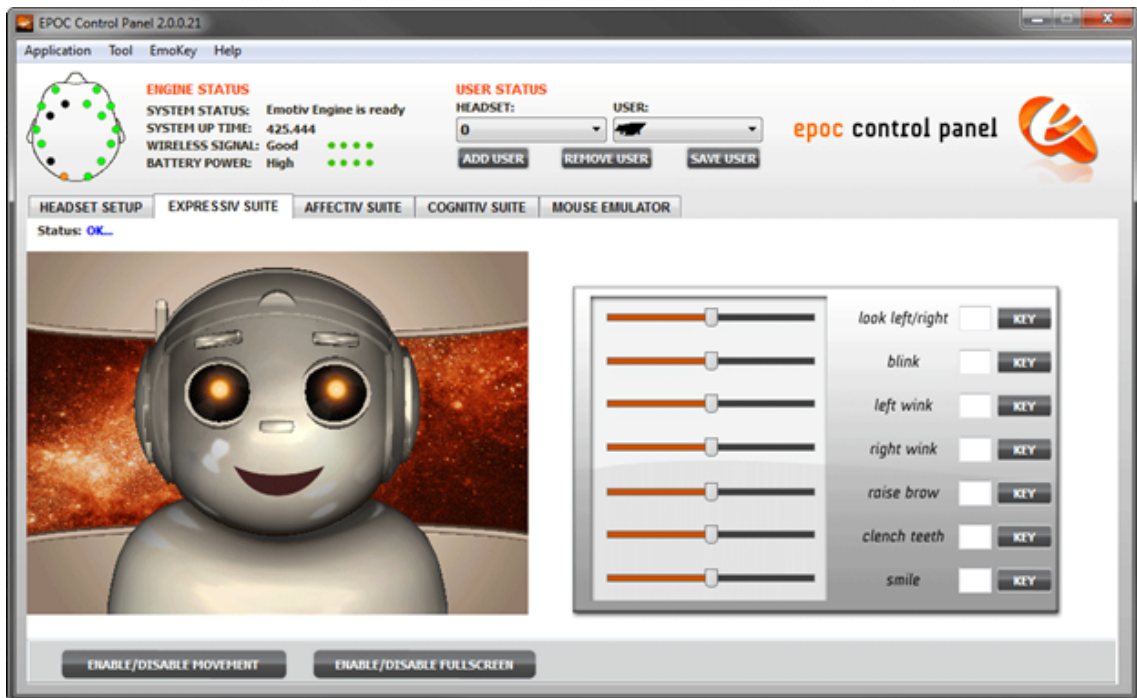


Figure 4.16: Emotiv EPOC Expressive Suite



Figure 4.17: Emotiv EPOC Affective Suite

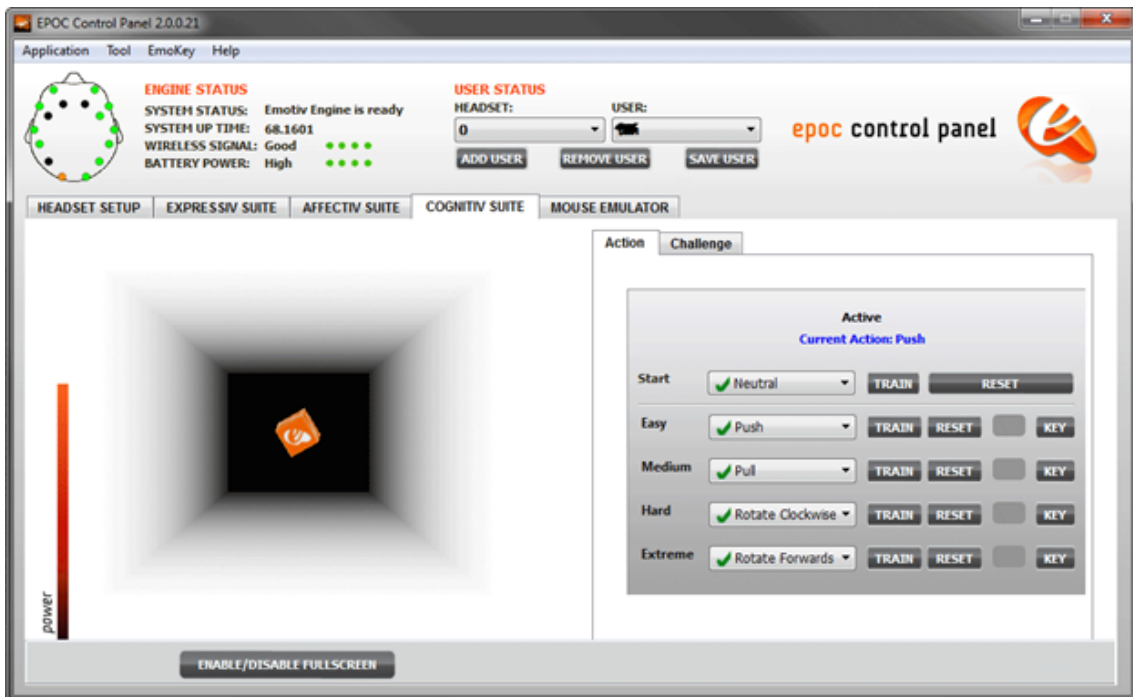


Figure 4.18: Emotiv EPOC Cognitive Suite

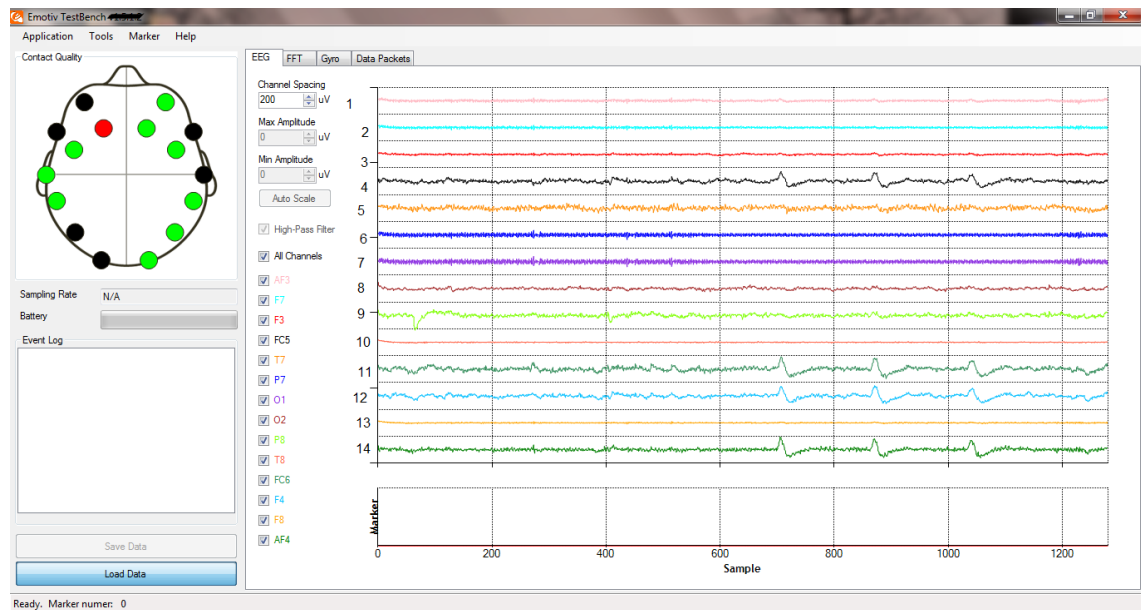


Figure 4.19: Emotiv EPOC Test Bench

### Emotiv EPOC C-Sharp SDK:

The Emotiv API (Figure 4.20) is uncovered as an ANSI C interface actualized in two Windows DLLs: *verbedk.dll* and *verbedk\_utils.dll*. The center of the Emotiv SDK is the EmoEngine, which is an intelligent abstract system that ‘speaks with the Emotiv headset, gets preprocessed EEG signals and gyroscope information, oversees client particular or application-particular settings, performs post-handling, and deciphers the Emotiv signals results into a simple to-utilize structure called an EmoState’. Every EmoState speaks to the present information from the headset including ‘facial, emotional, and subjective state’ and, contains sensor estimations for every contact. Using the Emotiv API comprises associating with the EmoEngine, detecting, distinguishing and discovering new EmoStates, and calling code significant to the new EmoState [19].

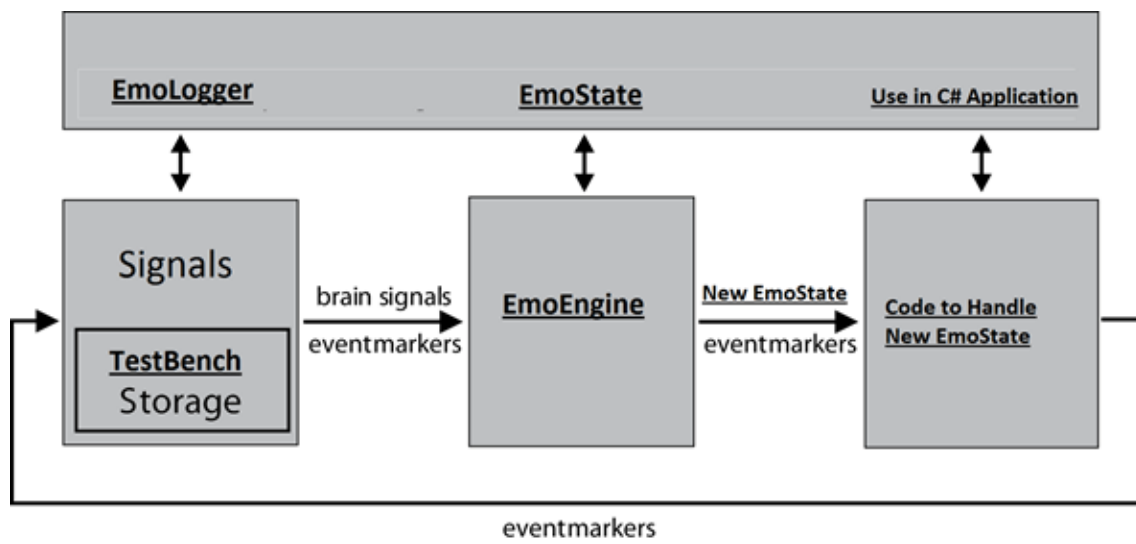


Figure 4.20: Emotiv EPOC C# SDK

## Chapter 5

# System Implementation

This chapter presents in detail the implementation details of the system in the following sections.

### 5.1 System Architecture

As mentioned earlier, the project relies on interfacing between two different systems. In this section we will discuss the components of our systems individually and the communication between them.

The first Component of our system is a C# game, the game architecture being illustrated in Figure 5.1. The game consists of two forms and the external dependencies.

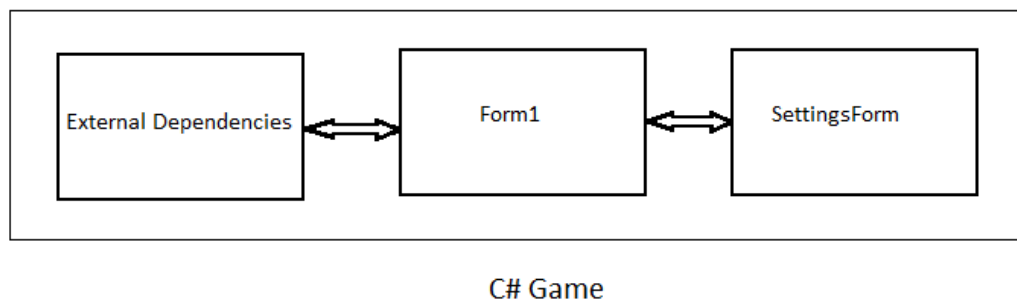


Figure 5.1: C-Sharp Game Architecture

The Emotiv EPOC contains sub components which are Emotiv Control Panel and SDK (Figure 5.2). Emotiv Control panel has three different suites/modes i.e. Affective, Expressive and Cognitive. The Emotiv C# SDK contains several classes like EmoState,

EmoEngine, EmoLogger. Through these classes, we can work on all functions/features which are present in Emotiv Control Panel.

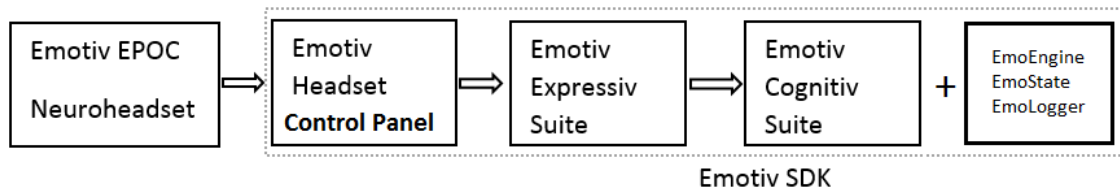


Figure 5.2: Emotiv Component Architecture

The communication of Emotiv EPOC and our C# game using Emotiv C# SDK is illustrated in Figure 5.3.

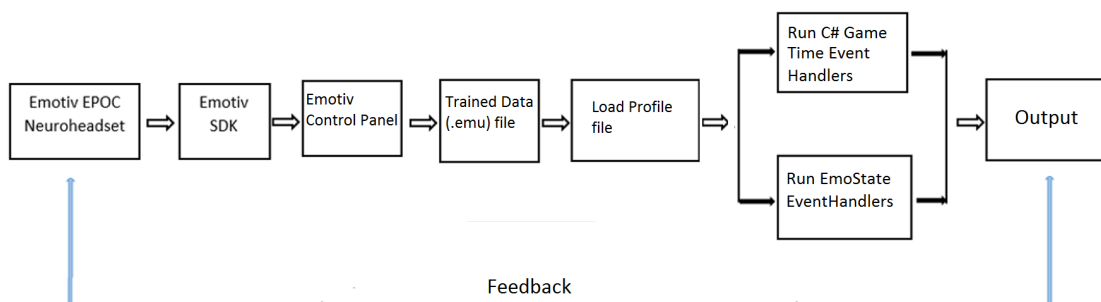


Figure 5.3: Integration of the Game and Emotiv EPOC components

The over all system architecture is divided into three parts listed below and illustrated in Figure 5.4.

- Signal Acquisition Unit
- Data Signal Processing Unit
- Application

In the Data/Signal acquisition block we create/load user profile and create connection with SDK/Control Panel. We then acquire brain signals from the user. User is provided with a proper GUI which shows all three suites and depicts the training start/end/success in a separate tab. The data processing algorithm section represents the part that involves training and later usage of the input signals in C# game as stimuli with the support of external references, files and libraries. The application block contains a C# game with GUI and is connected to Emotiv EPOC data acquisition block as neural data is the source of input for the game.

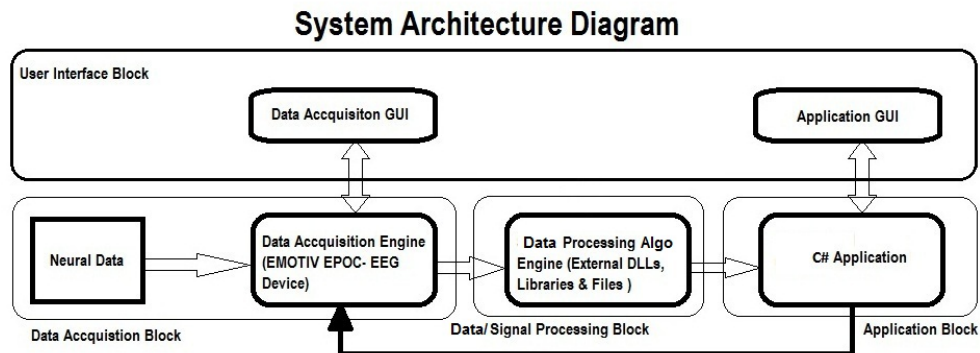


Figure 5.4: Whole System Architecture

## 5.2 Tools and Technology Used

The following tools and technologies have been employed in our project.

- Microsoft Visual Studio
- Emotiv EPOC Headset
- Emotiv Control Panel
- Emotiv C# SDK

## 5.3 Development Environment/Languages Used

C# has been used as the development language while Microsoft Visual Studio as the development environment.

## 5.4 Processing Logic/Algorithms

The basic logic employed in our system is training a subject on the cognitive suite of the Emotiv EPOC Control Panel. User is asked to concentrate thoughts on a particular action like 'push' or 'pull' etc. for a certain duration. Signals captured during this duration are stored in a training file for the particular subject. A subject may train for many actions the common ones include thinking 'Push', 'Pull', 'Left', 'Right', 'Rotate' and 'Move' etc. During the game play, the training file of the particular user is loaded in the program. The brain signals acquired during game play are matched against the training file to find a match between a stored thought and current thoughts. Once the signal during game play is matched with one of the pre-stored thought signals, the action corresponding to the

respective thought is executed in the game.

In addition to the cognitive suite, we also employ the expressive suite for game control. Unlike cognitive control, the expressive control is relatively simple and does not require repeated trainings on a subject. The signals corresponding to the expressive actions (winking, looking Right, looking left etc.) are more prominent and detectable and are used to control the game.

We used a number of Emotiv Event Handlers for controlling the game with expressive actions or cognitive thoughts. These Timer Event Handlers are synced with C# Game's timer controls to recursively have the input.



## Chapter 6

# System Testing and Evaluation

This chapter presents the evaluations we carried out to test the functionality of the developed system. Different evaluations are described in the following sections.

### 6.1 Graphical User Interface Testing

This section describes the GUI testing of the C# game module of the system (Figure 6.1). For testing, we employed a sample of three different users, who were required to use the interface and provide their feedback.

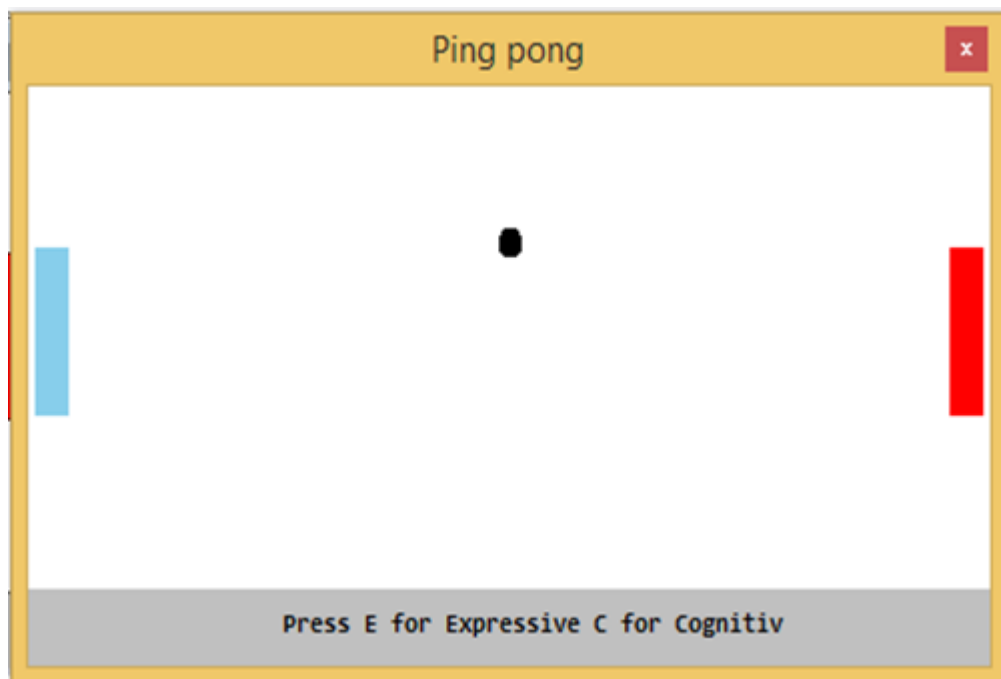


Figure 6.1: Graphical User Interface of the Game

The key points in the feedback provided by the users are listed in the following.

- The color scheme of UI controls is not appealing.
- Instead of counting and showing the remaining lives of the players in numeric values, progress bars should have been used.

These points were taken into account and the revised interface of the game comprised appealing colors. Likewise, labels were used to show the status of the game.

## 6.2 Software Performance Testing

The performance of the system was evaluated on a number of individuals. The findings of only a limited number of these are incorporated to guarantee productive examinations and discussions. The testing was carried out by means of the following.

- Unit Testing
- Integration Testing
- Compatibility Testing

Each of these is discussed in the following sections.

### 6.2.1 Unit Testing

Since the system is based on interfacing of an EEG device and a C# game, we have to first test the EEG device and then the game. The EEG device, Emotiv EPOC contains Control Panel and a C# SDK. The following was observed during testing.

- The Emotiv EPOC Control Panel starts, load and creates profiles flawlessly. Furthermore, the Affective, Expressive and Cognitive Suites also start and work without any errors.
- The C# SDK has exception handling and proper error messages incorporated once an error occurs. The C# SDK is well written and documented we did not any error while using it multiple times.
- The C# game was also tested and the readings were recorded for multiple users. The details of this testing are provided later in the chapter.

### 6.2.2 Integration Testing

In integration testing we faced few problems. The external libraries `verb'edk.dll'` and `verb'edkutil.dll'` are 32 bit system files while the developed game contains 64-bit system files. When executing, the system throws an exception (Figure 6.2) and the game does not run. We, therefore, had to change the Debug Configuration of the system to x86 Active solution platform from Debug Configuration Manager of Visual Studio 2010.

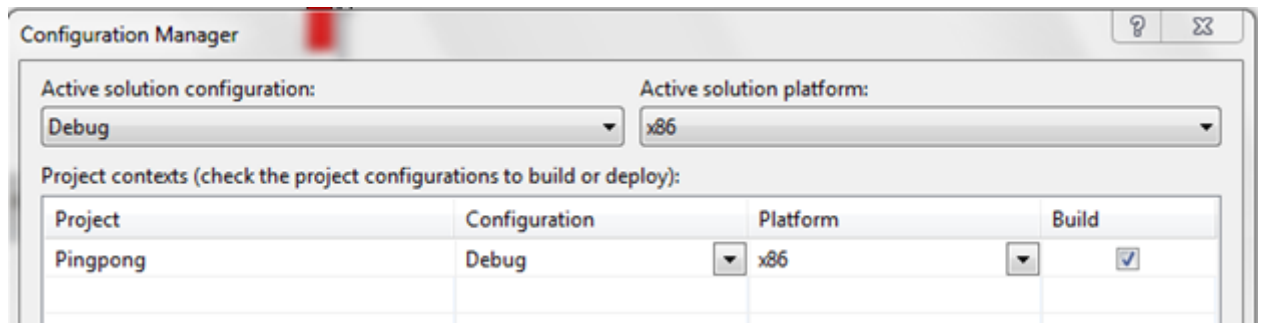


Figure 6.2: Debugging Configuration Manager

### 6.2.3 Compatibility Testing

The Emotiv EPOC SDK has been written in C# while our developed game also employs C# as the development platform. Consequently, we did not face any compatibility issues with the two modules.

## 6.3 Test Cases

To design test cases, we first consider the work flow of our system as illustrated in Figure 6.3. The user chooses among Expressive/Cognitive Suites to execute the desired action. If the player paddle misses the ball the score of the opponent increases by 1 and user life gets wasted. The final score is recorded when the user or opponent has 5 lives wasted.

We have used three test cases corresponding to the three use cases we presented in Chapter 3.

1. **Test Case 1: Training user on Emotiv EPOC Control Panel Cognitive Suite** In this test case, we train subjects on Emotiv EPOC Control Panel Cognitive Suite. Multiple users have been trained on the suite in order to get test cases readings. It has to be ensured that each user is relaxed before the start of the training. The following steps are carried out during training.

- A new profile is created for every new user.

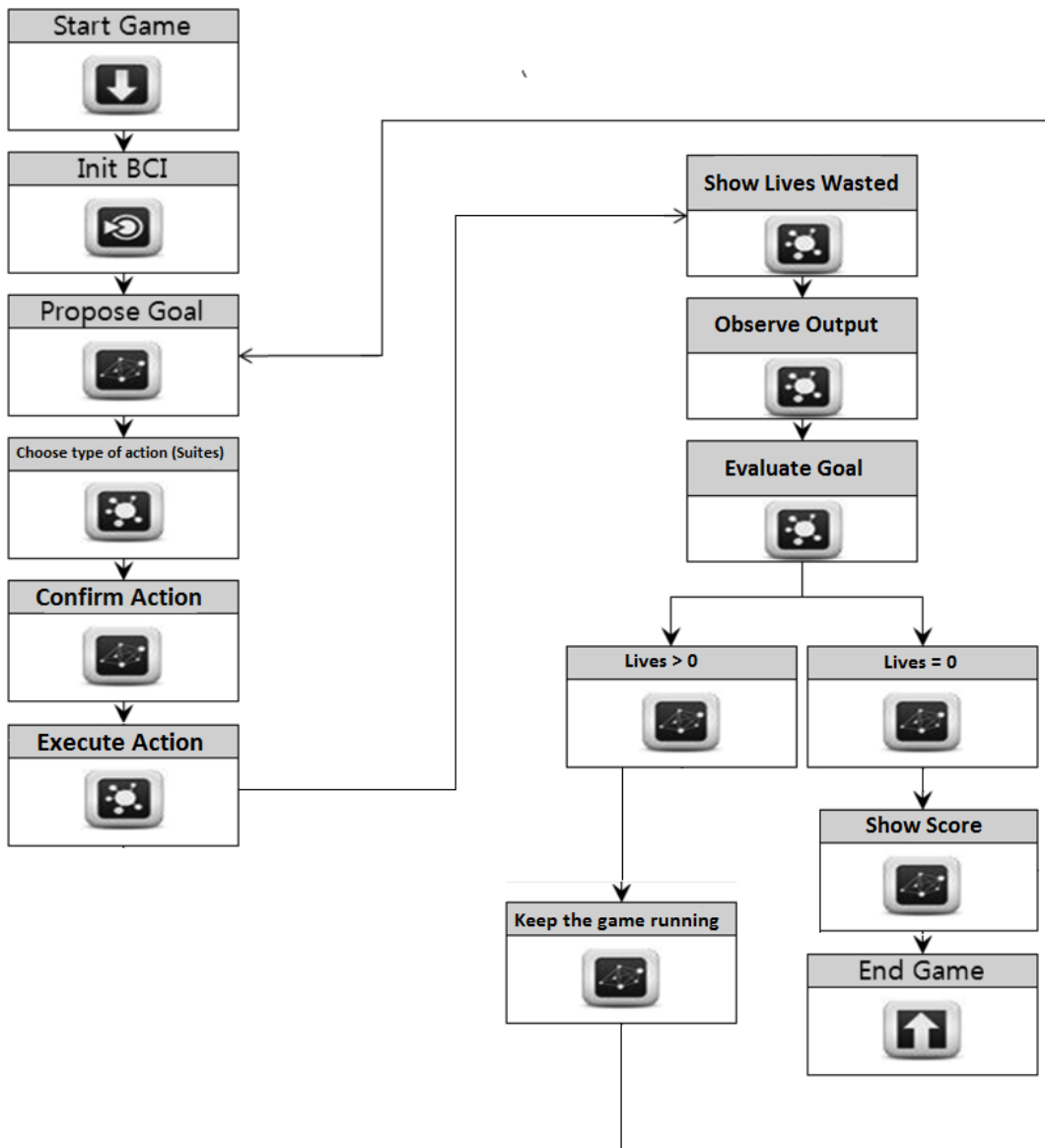


Figure 6.3: Work Flow Steps

- For an existing user, the previously generated profile is loaded and training is launched.
- The participant is asked to move a cube in a 3D space with brain signals (thoughts like push, pull, rotate etc.) in order to train the classifier.

In our system, we use for cognitive actions. These include thoughts of moving towards left, moving towards right, pulling towards user and pushing away from user. These actions produce different parameter values for different users. Emotiv EPOC Control Panel records the training action values in a float variable ‘Power’ on a scale of 0.0 to 1.0. The readings for three different users are summarized in Table 6.1 while a snapshot of the cognitive suite training panel is presented in Figure 6.4.

Table 6.1: Cognitive Actions training values

Actions	Push	left	Right
User 1	0.5	0.7	0.65
User 2	0.3	0.4	0.7
User 3	0.6	0.4	0.3

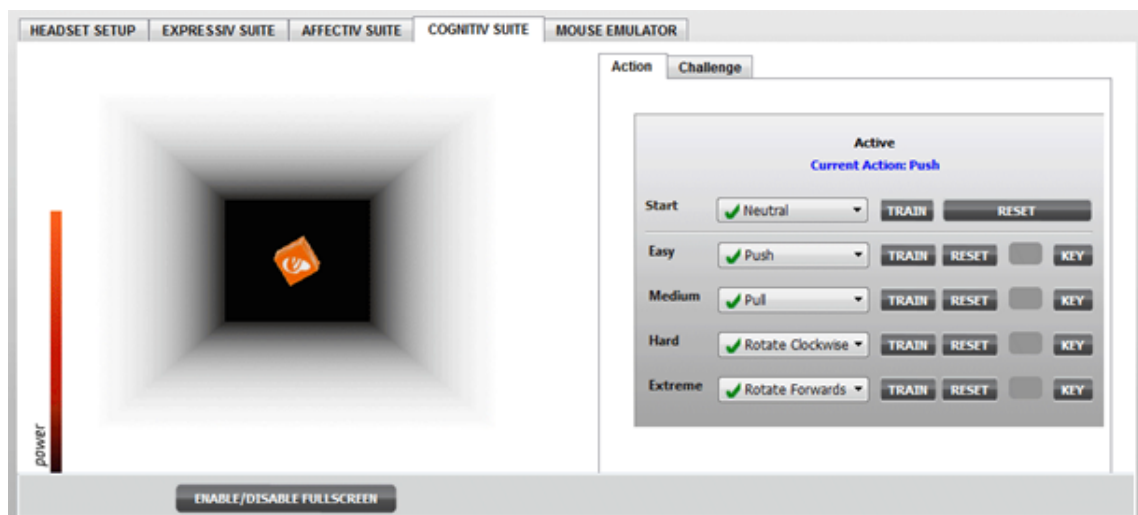


Figure 6.4: Emotiv EPOC Cognitiv Suite Training Panel

2. **Test Case 2: Controlling Game Application using Expressive Actions** In this test case, we evaluate the system performance using expressive actions as input to the game. The expressive actions result in prominent and noticeable deflections in brain signals and can be easily recognized when compared to the cognitive signals. The three expressive actions considered in our system include looking left, looking right and blink of eye. Users were asked to perform these actions and the corresponding parameter values were recorded (Table 6.2).

Table 6.2: Expressive Actions Test Table

Actions	Looking Left	Looking Right	Expressive Eye Blink
User 1	0.4	0.3	0.55
User 2	0.4	0.5	0.65
User 3	0.7	0.6	0.8

The user actions were then employed to control the game, the corresponding hits and misses being summarized in Table 6.3 which shows the number of hits and misses in 15 seconds of gameplay using expressive thoughts as input.

Table 6.3: Hits and Misses using Expressive Actions

Users	Hits	Misses
User 1	4	7
User 2	6	4

**3. Test Case 3: Controlling Game Application using Cognitive Actions** This test case evaluates the system performance using cognitive actions as input to the game. Cognitive actions do not produce as strong signals as those of the expressive actions. We first load the training data for a particular subject and ask the subject to focus on a sequence of thoughts. A lot of variation in parameter values of different subjects is observed. This is the main reason a separate training file is to be maintained for each user. The following steps are taken while using the cognitive actions for game control.

- User is asked to think those actions for which training is carried out.
- User should be patient as the cognitive action signal processing is slow.
- If the desired action does not execute within five seconds, we have to check all the mediating factors which include sensor placement, sensor dryness, user attention etc.

During game control testing, users were asked to control the game for 15 seconds using only cognitive thoughts as input. The hits and missed during 15 seconds of game play are summarized in Table tab:HitsAndMissesWRTTime2.

Table 6.4: Hits and Misses using Cognitive Action

Users	Hits	Misses
User 1	4	5
User 2	5	3

## 6.4 Exception Handling

During system testing, we came across few exceptions which are listed in the following.

1. **Exception 1:** This exceptions occurs before start of the game. It is thrown by system to signal failure to load the required dll file. The exception message says ‘Unable to load DLL *verbedk.dll*’ as illustrated in Figure 6.5.

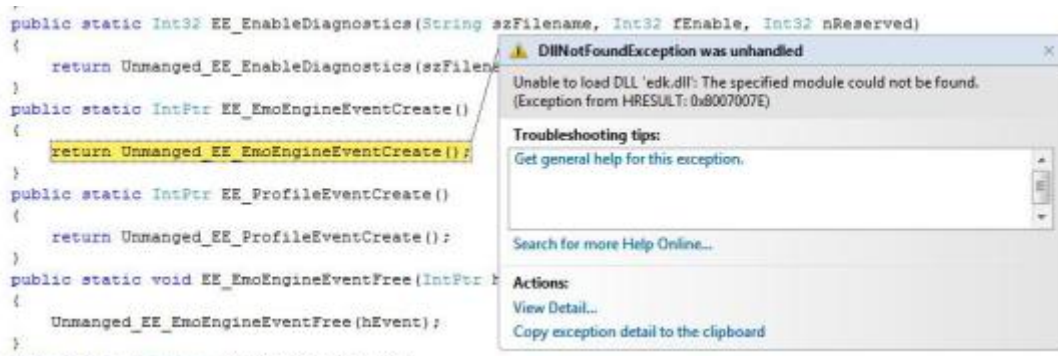


Figure 6.5: Exception 1

This exception was delat by placing the external libraries i.e. ‘*edk.dll*’ and ‘*edk\_Utils.dll*’ in the ‘Debug’ folder of the project directory.

2. **Exception 2:** The system throws an exception if we start a game without connecting it to the Emotiv EPOC device. The exception message says ‘EmoEngineException: Unable to establish connection with remote instance of EmoEngine’. We have handled this exception by proving user with appropriate message. In case of no connectivity, user is provided with the respective message as illustrated in Figure 6.6.

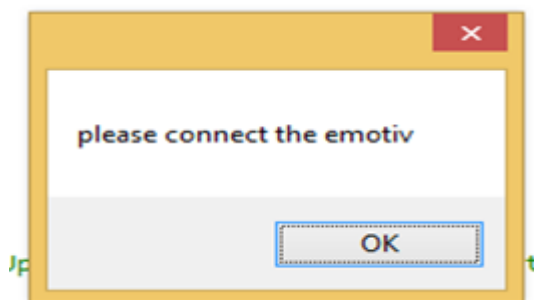


Figure 6.6: Exception 2





## Chapter 7

# Conclusion and Perspectives

We presented a Brain Computer Interfaced (BCI) system that employs game control as an application. Using brain signals to control robots, wheel chairs, vehicles and other devices has been a very attractive research area over the last two decades the challenging part being the availability of appropriate hardware to capture the weak signals from the brain. While the early BCI applications mostly focused on medical aspects to assist the impaired, neuro-gaming has emerged as an interesting application in the recent years and also makes the subject of our study. Having many options and features to control the game, we can read the mood changes of players and we can provide them a better game experience.

We employed the Emotiv EPOC headset to capture brain signals using the accompanying SDK. These signals were then used to control a game. Two kinds of EEG signals were considered in our study, expressive and cognitive. While the signals corresponding to expressive actions are strong, those of cognitive actions are relatively weak and requires training of subjects prior to control the game.

We considered a simple game to demonstrate that thought signals can be effectively employed to control a game or a device. The same idea can be extended to more complex games where a number of cognitive actions can be related with different functionalities. The Affective Suite can also be explored and can be used, for example, to change songs on a music player by reading the user stress/frustration/excitement level. Combining the signals from the affective, expressive and cognitive suites a large number of BCI applications can be developed in the emerging domain of neuro-gaming as well as elsewhere.



# References

- [1] Dietmar Dietrich, Roland Lang, Dietmar Bruckner, Georg Fodor, and Brit Müller. Limitations, possibilities and implications of brain-computer interfaces. In *Human System Interactions (HSI), 2010 3rd Conference on*, pages 722–726. IEEE, 2010. Cited on pp. 1 and 5.
- [2] Paul Coulton, Carlos Garcia Wylie, and Will Bamford. Brain interaction for mobile games. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, pages 37–44. ACM, 2011. Cited on pp. 1 and 5.
- [3] Athanasios Vourvopoulos and Fotis Liarokapis. Robot navigation using brain-computer interfaces. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 1785–1792. IEEE, 2012. Cited on pp. 1 and 6.
- [4] Johan Philips, José del R Millán, Gerolf Vanacker, Eileen Lew, Ferran Galán, Pierre W Ferrez, Hendrik Van Brussel, and Mamix Nuttin. Adaptive shared control of a brain-actuated simulated wheelchair. In *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*, pages 408–414. IEEE, 2007. Cited on pp. 2 and 5.
- [5] Jose Del R Millan, Ferran Galán, Dirk Vanhooydonck, Eileen Lew, Johan Philips, and Marnix Nuttin. Asynchronous non-invasive brain-actuated control of an intelligent wheelchair. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 3361–3364. IEEE, 2009. Cited on p. 5.
- [6] Mayur Palankar, Kathryn De Laurentis, and Rajiv Dubey. Using biological approaches for the control of a 9-dof wheelchair-mounted robotic arm system: Initial experiments. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 1704–1709. IEEE, 2009. Cited on pp. 5 and 6.
- [7] GN Ranky and S Adamovich. Analysis of a commercial eeg device for the control of a robot arm. In *Bioengineering Conference, Proceedings of the 2010 IEEE 36th Annual Northeast*, pages 1–2. IEEE, 2010. Cited on p. 5.
- [8] Marijn Van Vliet, Arne Robben, Nikolay Chumerin, Nikolay V Manyakov, Adrien Combaz, and Marc M Van Hulle. Designing a brain-computer interface controlled video-game using consumer grade eeg hardware. In *Biosignals and Biorobotics Conference (BRC), 2012 ISSNIP*, pages 1–6. IEEE, 2012. Cited on p. 5.

- [9] Andrew Campbell, Tanzeem Choudhury, Shaohan Hu, Hong Lu, Matthew K Mukerjee, Mashfiqui Rabbi, and Rajeev DS Raizada. Neurophone: brain-mobile phone interface using a wireless eeg headset. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, pages 3–8. ACM, 2010. Cited on p. 6.
- [10] Athanasios Vourvopoulos and Fotis Liarokapis. Brain-controlled nxt robot: Teleoperating a robot through brain electrical activity. In *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2011 Third International Conference on*, pages 140–143. IEEE, 2011. Cited on p. 6.
- [11] NR Raajan and GN Jayabhavani. A smart way to play using brain machine interface (bmi). In *Information Communication and Embedded Systems (ICICES), 2013 International Conference on*, pages 1130–1135. IEEE, 2013. Cited on p. 6.
- [12] Shangkai Gao, Yijun Wang, Xiaorong Gao, and Bo Hong. Visual and auditory brain–computer interfaces. *Biomedical Engineering, IEEE Transactions on*, 61(5):1436–1447, 2014. Cited on pp. 6, 7, and 8.
- [13] Gerwin Schalk, Dennis J McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R Wolpaw. Bci2000: a general-purpose brain-computer interface (bci) system. *Biomedical Engineering, IEEE Transactions on*, 51(6):1034–1043, 2004. Cited on pp. 7, 8, and 10.
- [14] Melody M Moore. Real-world applications for brain-computer interface technology. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 11(2):162–165, 2003. Cited on pp. 7 and 10.
- [15] Jinyi Long, Yuanqing Li, Hongtao Wang, Tianyou Yu, Jiahui Pan, and Feng Li. A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 20(5):720–729, 2012. Cited on pp. 7, 8, and 9.
- [16] Cecilia L Maeder, Claudia Sannelli, Stefan Haufe, and Benjamin Blankertz. Pre-stimulus sensorimotor rhythms influence brain–computer interface classification performance. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 20(5):653–662, 2012. Cited on p. 8.
- [17] Christoph Guger, Herbert Ramoser, and Gert Pfurtscheller. Real-time eeg analysis with subject-specific spatial patterns for a brain-computer interface (bci). *Rehabilitation Engineering, IEEE Transactions on*, 8(4):447–456, 2000. Cited on p. 9.
- [18] Paul S Hammon and Virginia R De Sa. Preprocessing and meta-classification for brain-computer interfaces. *Biomedical Engineering, IEEE Transactions on*, 54(3):518–525, 2007. Cited on p. 10.
- [19] Danny Plass-Oude Bos, Boris Reuderink, Bram van de Laar, Hayrettin Gürkök, Christian Mühl, Mannes Poel, Anton Nijholt, and Dirk Heylen. Brain-computer interfacing and games. pages 149–178, 2010. Cited on pp. 11 and 35.

- [20] Ernst Niedermeyer and FH Lopes da Silva. *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005. Cited on p. [23](#).

