

THE UNIVERSITY OF MANCHESTER

CogniDriver

Author:

Maria - Daniela Florescu

Supervisor:

Toby L. J. Howard

2015

Abstract

CogniDriver

Author: Maria - Daniela Florescu

The aim of the project is to provide an overview of the development of CogniDriver, a 3D mind-controlled car driving game which uses the Emotiv EPOC interface to control the car movement. The headset also allows the activation of various game effects such as raining when the frustration levels are too high or changing the camera view when the player does a left wink.

The Unity game engine has been used while developing the game in order to gain access to an advanced physics engine and to a multi-platform deployment tool.

The results of user testing on 13 participants have shown that although it is more difficult to control the game while in Cognitiv mode, the game appears more challenging and more interactive than a normal Keyboard mode play.

In the conclusions a short overview of the current deficiencies are presented as well as potential for further development.

Supervisor: Toby L. J. Howard

Acknowledgements

I would firstly like to thank my supervisor, Toby Howard, who has provided great advice and support throughout the project.

I would also like to thank my family for their continuous support during my studies.

Finally, a big thank you to everyone who has helped with testing this project and whose feedback was greatly appreciated.

Contents

1	Introduction	7
1.1	Applications	7
1.2	Aims and objectives	7
1.3	Report outline	8
2	Background	9
2.1	Development tools	9
2.1.1	Unity	9
2.1.2	Easy Roads 3D	10
2.1.3	Photoshop	10
2.2	Electroencephalography (EEG)	10
3	Introducing the Emotiv EPOC	12
3.1	SDK description	12
3.1.1	Caring for the headset	16
3.1.2	Headset issues	17
3.1.3	EmoComposer	17
3.1.4	EmoKey	18
3.2	Training	18
3.3	Current Applications	19
4	Design	20
4.1	Overview	20
4.1.1	Project workflow	20
4.1.2	Requirements	21
4.1.3	Separation of concerns	22
4.2	Design Patterns	22
5	Implementation	24
5.1	Interfacing with Emotiv	24
5.1.1	EdkDll	24
5.1.2	EmoAffectiv	24
5.1.3	EmoCognitiv	25
5.1.4	EmoEngine	25
5.1.5	EmoEngineInst	25
5.1.6	EmoExpressiv	25

5.1.7	EmoGyroData	26
5.1.8	EmoProfileManagement	26
5.1.9	EmoState	27
5.1.10	EmoUserManagement	27
5.1.11	LabelDraw	27
5.2	Scene description	27
5.2.1	CarChoiceScene	27
5.2.2	HelpScene	29
5.2.3	MainMenu	29
5.2.4	MainScene	29
5.2.5	TrainingScene	30
5.3	Notes	30
5.3.1	Gyro mode	31
6	Testing and evaluation	32
6.1	Observations	33
6.2	NASA TLX results	33
6.3	Analysis of user questionnaire answers	34
6.4	Good points about the game	34
6.5	Bad points about the game	34
6.6	What should be improved and what has been implemented?	35
7	Conclusions	37
7.1	Personal development	37
7.2	Summary	37
	Bibliography	39
A	CogniDriver - User Study	42
B	CogniDriver - 3D Object Sources	45

List of Figures

2.1	A composition of alpha, theta and delta brainwaves. Credit: [Gall, 1992]	10
3.1	Sensor positioning for the Emotiv EPOC. Credit: Emotiv	13
3.2	AF3 and AF4 channel response to eye blink. Credit: [Adelson, 2011]	14
3.3	Expressiv suite in the Emotiv EPOC Control Panel	15
3.4	Affectiv suite in the Emotiv EPOC Control Panel	15
3.5	Mouse emulator in the Emotiv EPOC Control Panel	16
3.6	From left to right: clean inside of golden plate; clean outside of golden plate; oxidised inside of golden plate; oxidised outside of golden plate.	17
3.7	Cognitiv suite training profile	19
4.1	Use case diagram for CogniDriver.	21
4.2	Domain class diagram for CogniDriver. The instance variables have been omitted for simplification.	23
5.1	Scene interaction diagram from the moment the application starts loading.	28
5.2	Car choice scene	28
5.3	Main menu scene	29
5.4	Main scene	30
5.5	Training scene	31
6.1	Trained user profile during user testing	33
6.2	A print screen of the message which will occur in case the player tries to follow a shortcut to the finish line	35

List of Tables

2.1	Information about brainwaves types [Ning-Han Liu and Hsu, 2013]. Image Credit: Hugo Gamboa	11
3.1	Left: Abbreviations for sensor positioning; Right: Sensor contact quality and colour codes	12

Chapter 1

Introduction

1.1 Applications

Using BCIs (Brain Computer Interfaces) to control an application is of great use to persons with motor disabilities whose use of keyboard or mouse devices might not be convenient. In addition, BCIs provide a great way to exercise brain functions and improve concentration. [Choon Guan Lim, 2012] shows how the use of a BCI platform may improve the attention and concentration of ADHD patients. [A. Ferreira, 2006] show a way of remote-controlling an industrial manipulator through the use of an EEG system. [Xiaorong Gao, 2003] describes the use of a SSVEP¹ type system with 48 LEDs to match key-presses of a television remote-controller. [Fabien Lotte, 2008] use a BCI to study the performances and preferences of 21 participants while playing a ‘Star Wars’ themed game. For more examples of real uses of brain computer interfaces, see section 3.3.

1.2 Aims and objectives

The aim of this project was to learn about the pros and cons of BCIs, how these interact with the computer and reach a conclusion about how they could be used in the future. Because the Emotiv EPOC Control Panel (chapter 3) allows only up to 4 actions to be recognised at any one time, the simplest choice of a game was that of a car driving one since one can also observe the car moving in the dictated direction which can ease the activation of an action through the visual feedback.

The key objectives of the project were to:

- Allow the car to move in each one of the four directions: forward, reverse, left, right through keyboard use;
- Reach access to the interpreted data of the developer’s SDK;

¹SSVEP (Steady State Visually Evoked Potential) is a signal which is the brain’s natural response to visual stimulation at specific frequencies

- Allow players to train new profiles inside the game;
- Provide an array of cars and colours that the user can select from.

1.3 Report outline

The report begins with describing the used tools and provides some information about the way the brain works in Chapter 2. Next, Chapter 3 describes how the Emotiv EPOC works and how it is organised. In this chapter, the report also contains an overview of some other applications developed with this technology.

Chapter 4 focuses on the description of the design of this project, while Chapter 5 oversees the implementation details. Chapter 6 describes the performed testing and appendix A shows the user questionnaire used during the user trials. Finally, Chapter 7 contains the conclusions of this project. Appendix B provides the sources of the external 3D objects used in the project.

Chapter 2

Background

This chapter aims to provide a general overview of the tools used while building CogniDriver. It also describes some basic concepts about how the brain works which would be useful to know before diving into the description of the headset.

2.1 Development tools

2.1.1 Unity

Unity is a game engine which allows the user to import objects easily and have them automatically updated if the source file changes. One may also create primitive shapes such as spheres and cubes in order to build simple objects to which textures can be applied. All objects support affine transformations and Unity provides a good physics engine. It also allow for scene design and collision detection between different components. The final product can be deployed on a variety of platforms.

We should also highlight some of the reasons why Unity was chosen in favour of other game engines such as UnReal. Unity allows file import by a simple drag-and-drop and it also auto-updates the objects if the source files are changed using another program (e.g. Paint, Photoshop). The advantages of UnReal over Unity would be better graphics (mainly lighting) and the fact that it can fracture meshes. In terms of scripting, Unity allows development in JavaScript, C# and Boo, while UDK only allows UnrealScript. For CogniDriver, C# has been the scripting language choice, because it allows Object Oriented Programming (OOP), it is scalable for larger projects and bugs are easier to track down.

Unity has been used to create games such as Assassin's Creed Identity [Unity, 2015] and Battlestar Galactica Online[Wikipedia, 2015].

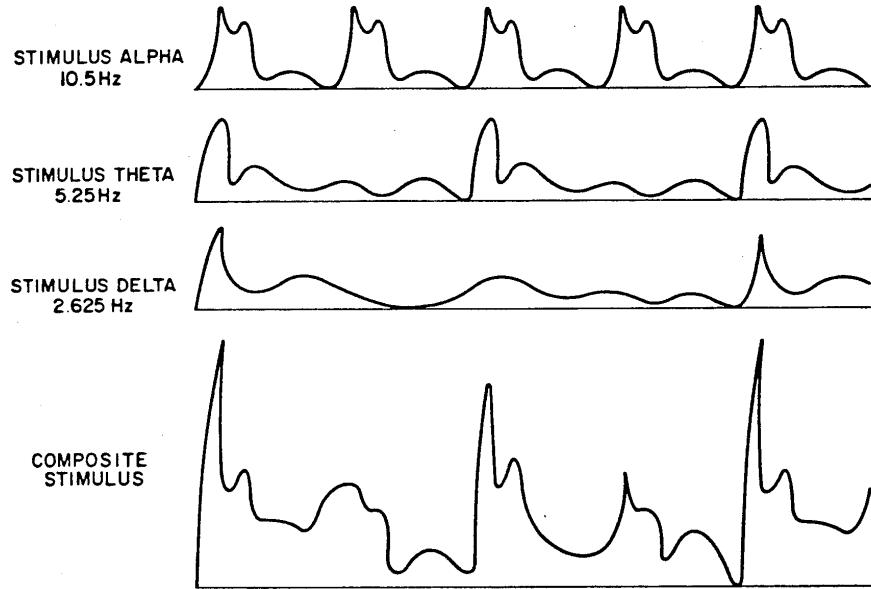


Figure 2.1: A composition of alpha, theta and delta brainwaves. Credit: [Gall, 1992]

2.1.2 Easy Roads 3D

This plugin is available in the Unity asset store. The free version has been used for this project. It allows placing markers through which the road should go, and given a specified width, it approximates corners and builds the rest of the road into a single layer.

2.1.3 Photoshop

A trial version has been used to create some materials and textures such as: skid marks, rain, smoke texture, speedometer, arrows, colour choice. None of these are difficult to create, but I like Photoshop because it is easy to use by both novice and professional users. In addition, the layers allow the user to create many compositions and modify the image at will.

2.2 Electroencephalography (EEG)

The neuron is the core component of the nervous system. More neurons can connect together to form synapses and when one of these cells is excited, an electrical signal is generated. The electrical signals can be amplified and the resulting waveforms, known as brainwaves, can thus be obtained. They can then be used to monitor the brain activity of a person in order to detect abnormalities. In table 2.1, a representation of each type of most common brainwave types is shown, together with the normal activities in which these occur most often.

¹The body or a region of the body is deprived of adequate oxygen supply

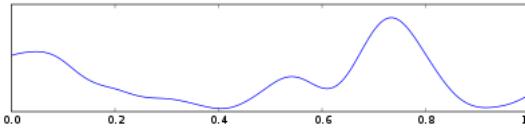
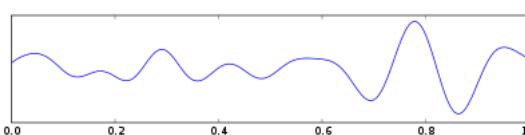
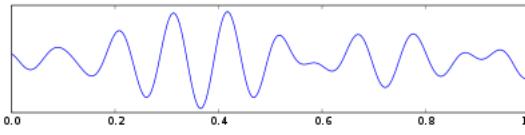
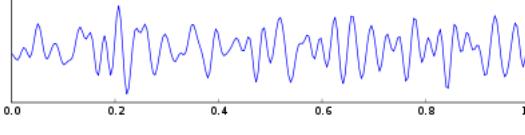
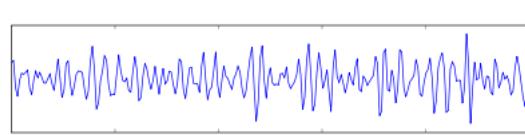
Name	Frequency (Hz)	Occurs in	Image Representation
Delta	<4	deep sleep , unconscious, deep anaesthesia, hypoxia ¹	
Theta	4 - 7	stress, unconscious, deep relaxation	
Alpha	8 - 15	relaxation	
Beta	16 - 31	consciousness, alertness, thinking, sensory stimulation	
Gamma	32 +	selective attention, human cognition and perceptive activities	

Table 2.1: Information about brainwaves types [Ning-Han Liu and Hsu, 2013]. Image Credit: Hugo Gamboa

Please note that almost always the EEG will display a composition of a set of types at any point in time. One example can be seen in the figure 2.1. Brain Computer Interfaces (BCIs) are a channel of communication between the electroencephalograph (EEG) and the computer.

Chapter 3

Introducing the Emotiv EPOC

3.1 SDK description

The Emotiv EPOC headset is a portable EEG device which consists of 14 sensors that transmit raw EEG data (see figure 3.1 for their positioning) to the computer and 2 reference sensors (CMS - Common Mode Sense and DRL - Driven Right Leg). The CMS sensor is the point on the scalp against which everything else is measured. The DRL sensor provides a feedback signal to cancel common mode noise in the electronics [Emotiv, 2015a, BioSemi, 2015]. See table 3.1 for the meaning of the abbreviations of sensor positions. The raw data gets interpreted and the user gains access to all of the information in one of the following 4 suites: Expressiv, Affectiv, Cognitiv, Mouse Emulator.

Abbreviation	Meaning	Colour code	Meaning
A	Ante	Black	No signal
F	Frontal	Red	Very poor signal
C	Central	Orange	Poor signal
P	Parietal	Yellow	Fair signal
T	Temporal	Green	Good signal
O	Occipital		

Table 3.1: Left: Abbreviations for sensor positioning; Right: Sensor contact quality and colour codes

We have been using the developer's SDK, so it is worth highlighting we did not have any access to the raw EEG data.

The battery lasts about 10 to 12 hours. The charging is done via a USB cable.

According to [Adelson, 2011], the data sampling rate is 128Hz. [emo, 2014b] states that there may be up to 2 seconds delay from when the data is sent until a visual feedback is observed.

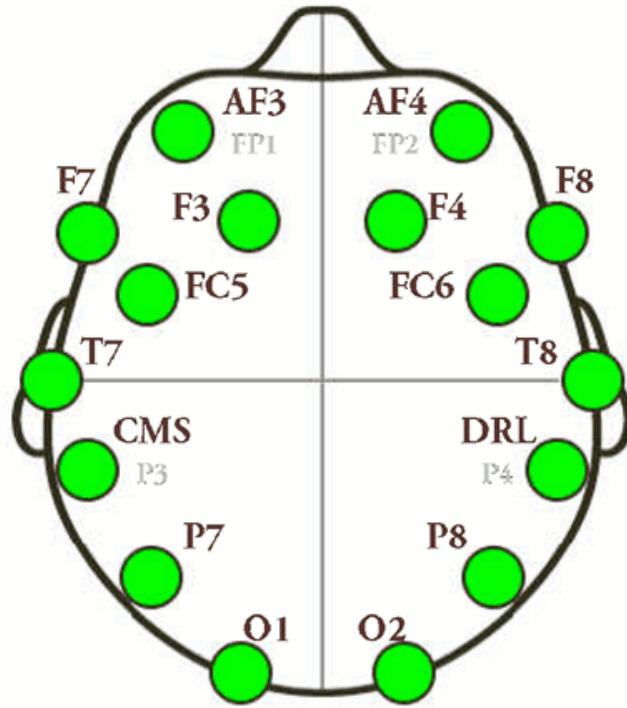


Figure 3.1: Sensor positioning for the Emotiv EPOC. Credit: Emotiv

ExpressivTMSuite

Detects 12 facial expressions:

- | | |
|---------------|------------------|
| 1. Blink | 7. Laugh |
| 2. Left wink | 8. Smirk left |
| 3. Right wink | 9. Smirk right |
| 4. Look left | 10. Raise brow |
| 5. Look right | 11. Furrow brow |
| 6. Smile | 12. Clench teeth |

One thing to note is that if the sensor contact quality is not perfect, blinks and winks might be mistaken; same applies to laughs and smiles. In fact, an ideal signal is obtained when all sensors display green (see table 3.1). It is acceptable to have some displaying yellow but anything less than yellow, means data might be unreliable or unavailable. Better sensor contact quality is obtained when the player lets their hair loose.

The API usually gives back data as a boolean, but for smile, clench and look left/right, a float between 0 and 1 can also be obtained which represents the extent of that facial expression.

Figure 3.2 shows an image of how the EEG looks like in channels AF3 and AF4 after a

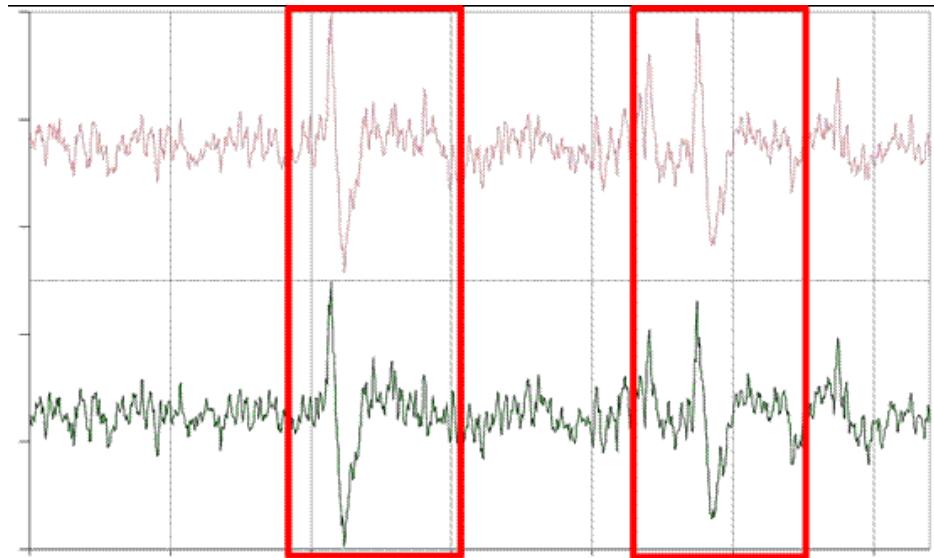


Figure 3.2: AF3 and AF4 channel response to eye blink. Credit: [Adelson, 2011]

blink. See figure 3.3 for a screenshot of the suite.

AffectivTMSuite

Returns a float between 0 and 1 which describes the power of one of the following emotions:

- Engagement/Boredom
- Frustration
- Meditation
- Excitement short term
- Excitement long term

I have found that these emotions represent my mood quite well. One thing to note is that boredom is not the kind of boredom one feels when they have nothing to do, but it is rather the opposite of engagement, according to Emotiv. See figure 3.4 for a screenshot of the suite.

The Emotiv User Manual [emo, 2014b] (page 31) provides information on the activities and brainwave types that may trigger each emotion.

CognitivTMSuite

This is probably the most exciting feature of the Emotiv EPOC. Here, one can train 13 actions: push, pull, left, right, lift, drop, 6 rotations (one in each direction for the X, Y and Z axis) and disappear. The disappear action is more abstract because while the first

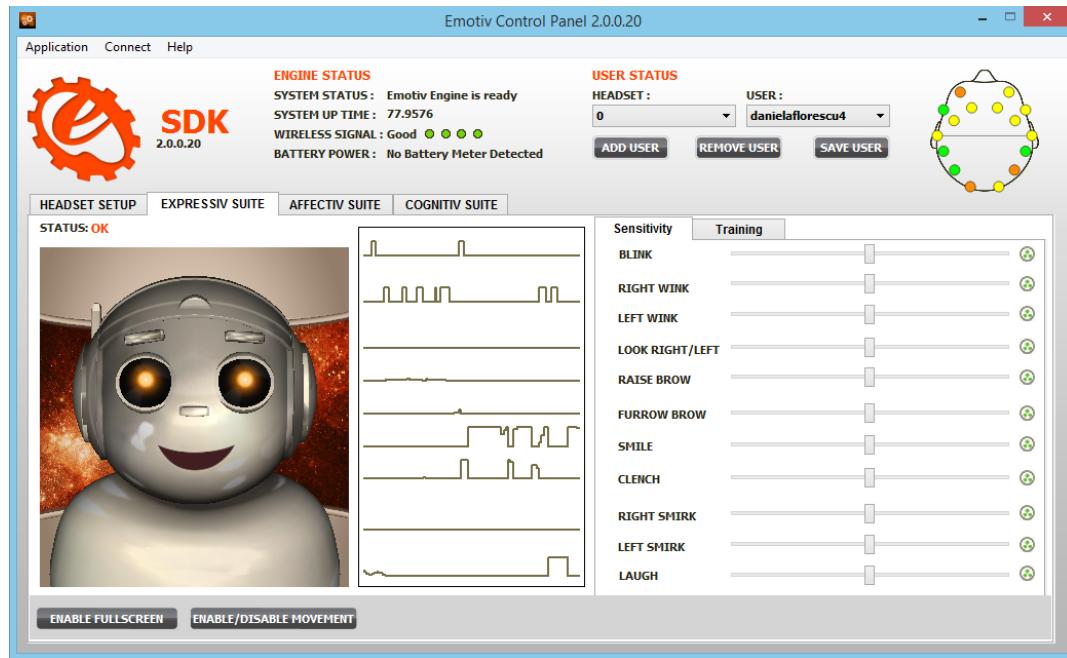


Figure 3.3: Expressiv suite in the Emotiv EPOC Control Panel

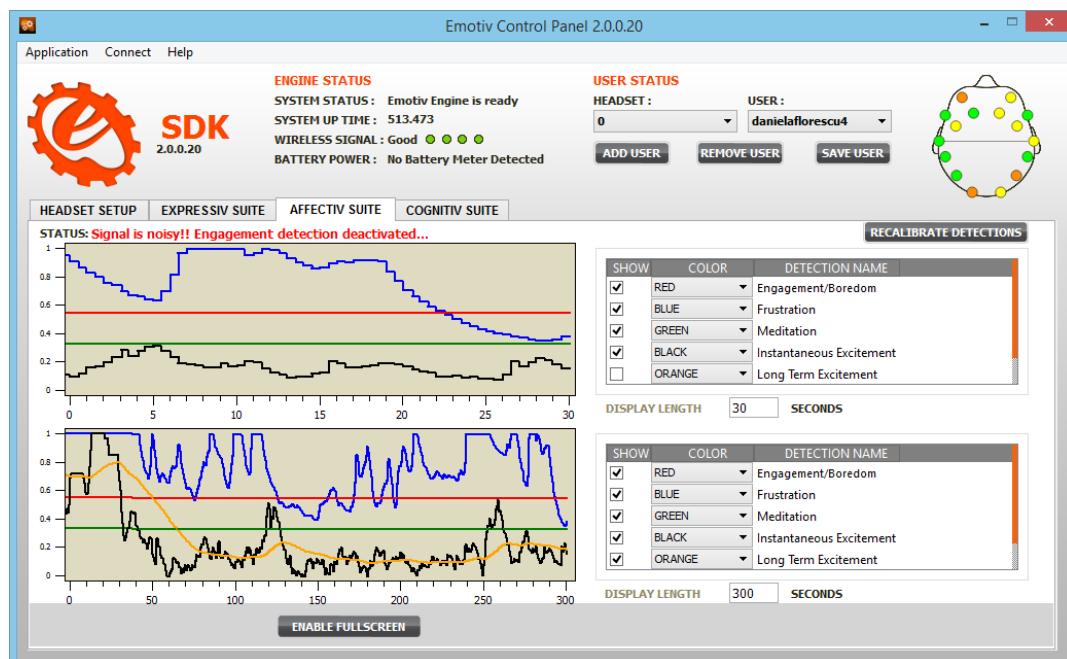


Figure 3.4: Affectiv suite in the Emotiv EPOC Control Panel

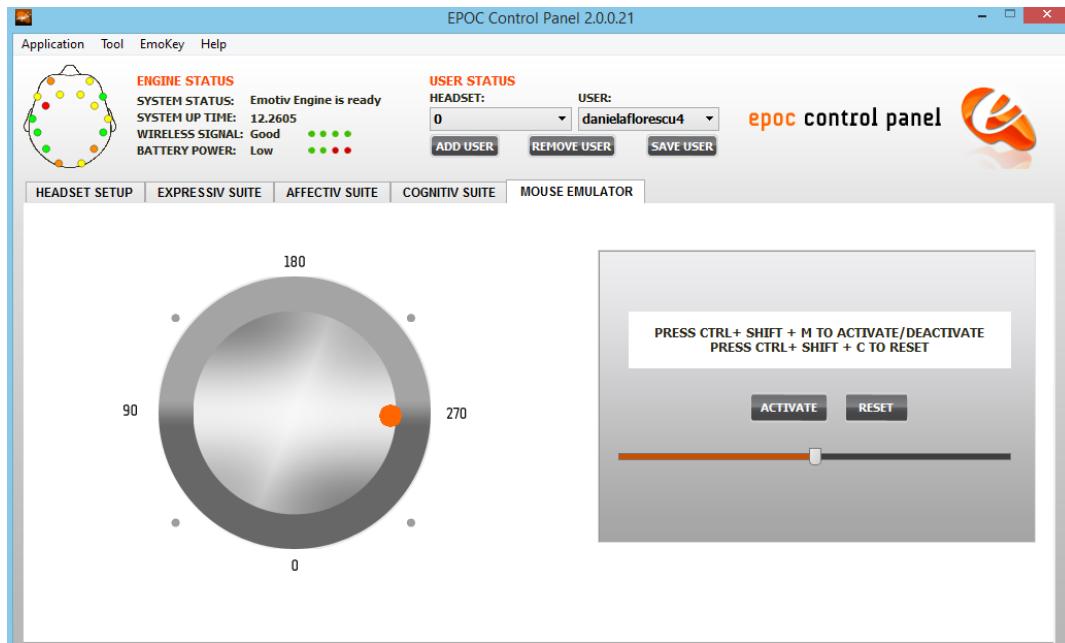


Figure 3.5: Mouse emulator in the Emotiv EPOC Control Panel

12 are real life actions, an object cannot vanish into thin air, in reality, through whatever methods. The user has to train each action for a period of 8 seconds. This should be done multiple times until an as high as possible skill level is obtained. The more you get used to training, the easier it will become. I have achieved close to 100% on all 4 actions: push, pull, left, right. At one point, I was able to achieve 100% on the push action in less than 5 trainings. The SDK can only detect up to 4 actions at any one time, so one has to choose wisely which actions would be best for their application. See figure 3.7 for a screenshot of the suite.

Mouse Emulator (Gyro)

This data actually comes from a gyroscope fitted in the headset. The original version aims to emulate the mouse movement by changing the head position. Once the head turns to left or right and remains in that position, the centre of the circle becomes the new head position. However, having access to the Unity 3D plugins, we were able to modify this suite so that the centre of the circle was not recalculated if the head stayed to the left or to the right because we wanted to use the gyro as yet another way to control the car movement. See figure 3.5 for a screenshot of the suite.

3.1.1 Caring for the headset

The 16 sensors need to be properly hydrated before starting to use the Emotiv EPOC. The hydration is done with contact lenses saline solution which has the purpose of enabling a better transmission of the brainwaves while also sanitising the felt pads. For the first use



Figure 3.6: From left to right: clean inside of golden plate; clean outside of golden plate; oxidised inside of golden plate; oxidised outside of golden plate.

of the sensors, one may need to apply 15-20 proper hydrations before good and reliable sensor contact quality is achieved.

When the user does not expect to use the headset for a period of more than 3-4 days, it is recommended to remove the felt parts from the sensors. Otherwise, the process of oxidation happens much more quickly. See figure 3.6 for images of clean sensors and oxidised sensors. The green residue can be cleaned with a cotton swab dipped into a solution of 1/2 water 1/2 white vinegar. If cleaning the interior of the golden plates, take care not to remove the slightly white polymer paste. I have left the felt parts in the sensors when I went on holiday for about 3 weeks. When I returned, the headset was losing contact quality very quickly and after about 15 minutes, it had to be rehydrated. It looks like salt columns may also form in the felt parts. My problem though, was that the golden plates oxidised a lot and after cleaning them as described above we achieved full green sensor contact quality which is the best you can wish for. We firstly cleaned the outside of the golden plates but that did not help too much. Then, we moved to cleaning the inside as well and that did the job for me, although on the Emotiv forums this is not something they would recommend. We were amazed to see the great contact quality we could achieve and it did not have any negative effects on the sensors as they are still in perfect shape 4 months after the cleaning.

3.1.2 Headset issues

One of the main problems encountered is that the wireless connection to the computer is not reliable and it can drop out of the blue. The EPOC+ is now on the market and it seems it might come with a solution for this.

3.1.3 EmoComposer

EmoComposer is a tool which sends simulations of EmoEngine events to an application. This made it easy to test the game without connecting the headset. This was particularly useful in more tricky to test situations which would have meant spending a long time trying to achieve an action by pure concentration.

3.1.4 EmoKey

EmoKey allows mapping of keys to all the interpretations described in this chapter. In this way, one can use the headset to send smiley faces when smiling or laughing for example. One may also use it to paint. These are two of the things that we have tried. It is supposed to be more useful for playing games using the headset. This has been tried in [Reinhold Scherer, 2012] for the *World of Warcraft* game.

3.2 Training

Because training is a difficult part of getting started with the headset, this section is only trying to provide some pointers as to what might work.

First of all, while training the user should try to keep as still as possible and concentrate on the action being trained. For example, if one is trying to train the push action, the thing which works most of the times is to visualise how the training cube is being pushed to the back of the virtual room.

Other people find it easy to visualise a flow of energy in the direction of the trained action.

Repeating words into one's mind does not work. We have seen it working when people say words out loud but the same pace should be kept.

Doing hand gestures works sometimes as [Emotiv, 2015b] seems to suggest. We have seen one user trying to do this. It worked for a while but then it stopped. Same happened to me.

One thing we have not tried is shifting concentration on the left or right hand side of the body as it is suggested in [Emotiv, 2015b].

Imagining colours or objects for different actions, does not seem to work.

The problem with training is that there is no 'recipe' one can follow. Rather, the user has to try to come up with something that works for them.

With regards to the training success, in figure 3.7 you can see one of my achievements. [Emotiv, 2015b] suggests the fastest learners seem to be young children who believe they can do anything, and very relaxed elderly. In the given example, the father of the poster, aged 82, was able to train reliably 4 actions in less than 5 minutes. For some notes on how the user testing training worked, read chapter 6.

Some people might encounter difficulties in training the headset. According to [Carmen Vidaurre, 2009], between 15% and 30% of the people trying to use a BCI, will not be able to do so. This condition is called 'BCI illiteracy' and it better analysed by [Minkyu Ahn, 2013]. Their study suggests these people have high theta and low alpha waves present across different mental states such as non task related, resting before motor imagery and motor imagery. [Carmen Vidaurre, 2009] have tried to come with a solution to this by using a subject-optimised classifier.

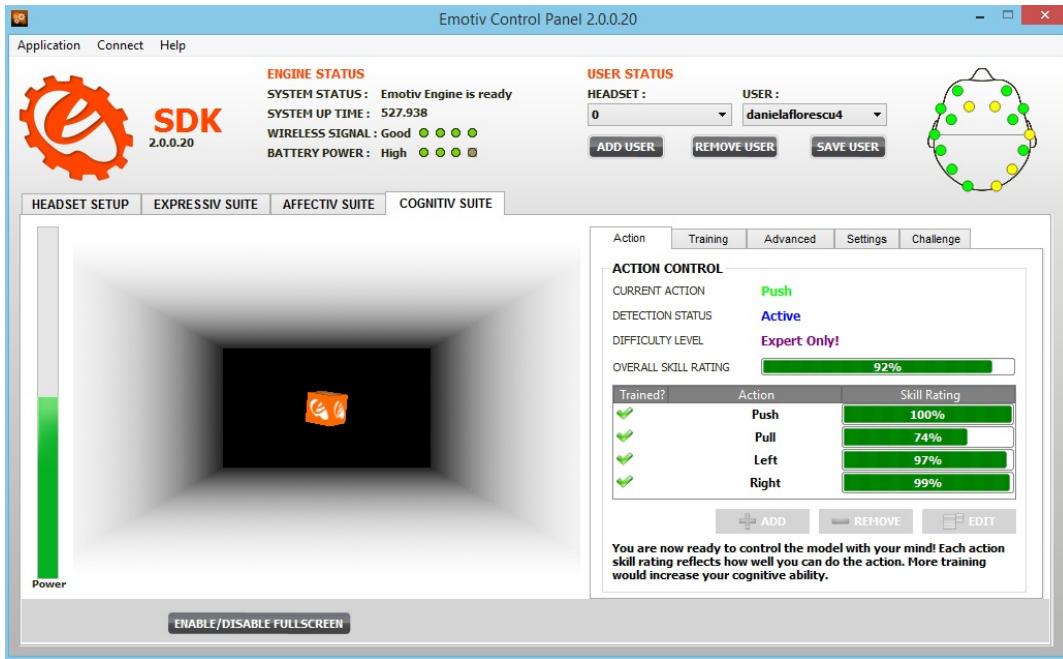


Figure 3.7: Cognitiv suite training profile

3.3 Current Applications

In this section, we are going to have a quick go-through some of the existent applications developed with the help of the Emotiv EPOC.

- NeuroPhone [Andrew T. Campbell and Raizada, 2010] is an application which uses the P300¹ signal and allows the user to scroll through photos of 6 contacts at a time. Once the person the user is looking for is found, the application will dial that person's number.
- [Sam Fok, 2011] have developed a system which allows the control of a hand prosthetic which opens and closes a patient's hand.
- [Francesco Carrino, 2012] shows it is difficult to successfully control applications in a highly error sensitive context, such as a wheelchair.
- Mindtunes [Smirnoff, 2013] is a project in which DJ Fresh has used multiple headsets in order to produce club quality music from the raw EEG data.
- EmoLens [Inc., 2010] tags a user's Flickr photos with emotions and learns to show similar pictures in the future as well as allowing picture search based on the tagged emotion.

¹The P300 is a detectable peak in activity that appears approximately 300ms after some infrequent stimuli [Lang, 2012]

Chapter 4

Design

This chapter describes the design decisions taken in this project and how these work towards achieving the aims described in chapter 1.

4.1 Overview

As a methodology used during the project, the Agile Unified Process has been followed. This decision was taken because the process is iterative and it allows the developer to be flexible if circumstances change. Agile UP also focuses on producing working code over comprehensive documentation.

For splitting the tasks, we have been using <http://trello.com>, a free project management tool which allows creation of virtual boards and cards for each task, making it a virtual Kanban system.

For producing the UML diagrams, we have used <http://www.gliffy.com/>, an online tool which allows the user to create 5 diagrams for free.

4.1.1 Project workflow

The project has consisted of 4 phases:

- **Phase 1** consisted of initial project documentation, background reading and an attempt to build a 3D car model to be used during the game. The sides of the car model were realistic enough, but I could not figure out how to build the front and the back, so I have used 3 free 3D models that I have found online. See Appendix B for the sources of these objects.
- During **phase 2**, I have learnt more about Unity by following the [Studio, 2014] tutorial. After that, [Tutorials, 2014] was a good starting point to how one can create a car game in Unity. Some more advanced car physics ideas were inspired from [Monster, 2015].

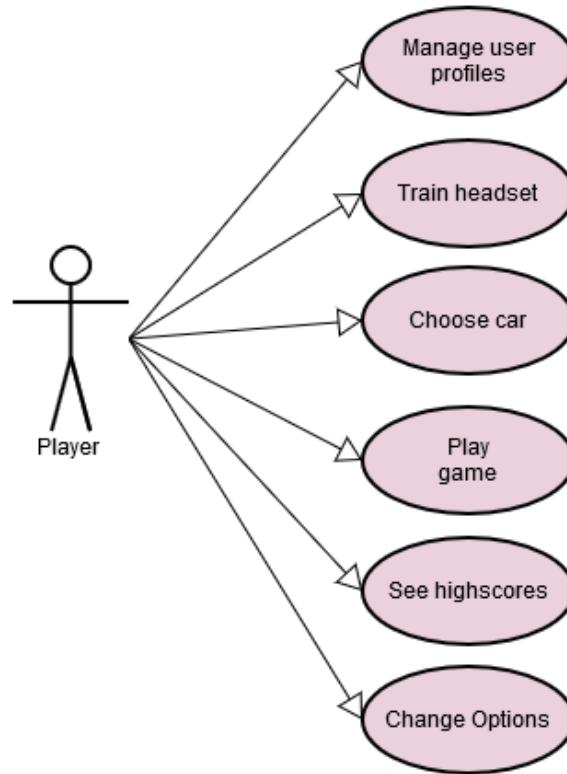


Figure 4.1: Use case diagram for CogniDriver.

- **Phase 3** has seen the start of implementing the functionality from the developer's API of the Emotiv EPOC. The most important part was getting the training working inside the game. See figure 4.1 for the use case diagram.
- In **Phase 4** the main concern was improving the user experience, getting some user testing done and allowing the player to choose between a variety of car models and colours.

4.1.2 Requirements

Functional requirements

In the final game, the player should be able to:

- Create/Delete/Change a player profile;
- Train the headset for neutral, push, pull, left, right states;
- Allow the user to accept/reject/reset a training session;

- Display the updated skill level after each training session;
- Provide visual feedback as to how well the training is going by using an animated object;
- Alert the user in the training scene if a training profile is not complete;
- Display the top 10 highest scores for Keyboard mode play and Cognitiv mode play;
- In the Options tab, allow the user to change sound volume, sound effects volume and whether the game will be played in fullscreen;
- Allow the user to change between Keyboard, Cognitiv and Gyro play modes;
- Provide instructions on how to train and play the game;
- Pause/Exit the game;
- Allow the selection of a car model and colour;
- Race yourself during the game play.

Non-functional requirements

- The game should be portable (i.e. playable on multiple operating systems);
- Once loaded, there should be no long waiting periods of time.

4.1.3 Separation of concerns

The Model-view-controller architecture has been used with the purpose of separating the concerns such that the controller (the script) is manipulating the model (car or environment) which then updates the view (the user interface).

4.2 Design Patterns

Abstract Factory has been used to create cars from the abstract Car class. AudiR8 and FerrariCalifornia are thus inheriting from Car. See figure 4.2 for the UML class diagram.

Strategy is used to decide which car control options to follow depending on the game play mode: Keyboard, Cognitiv or Gyro. It is also used in the car model choice.

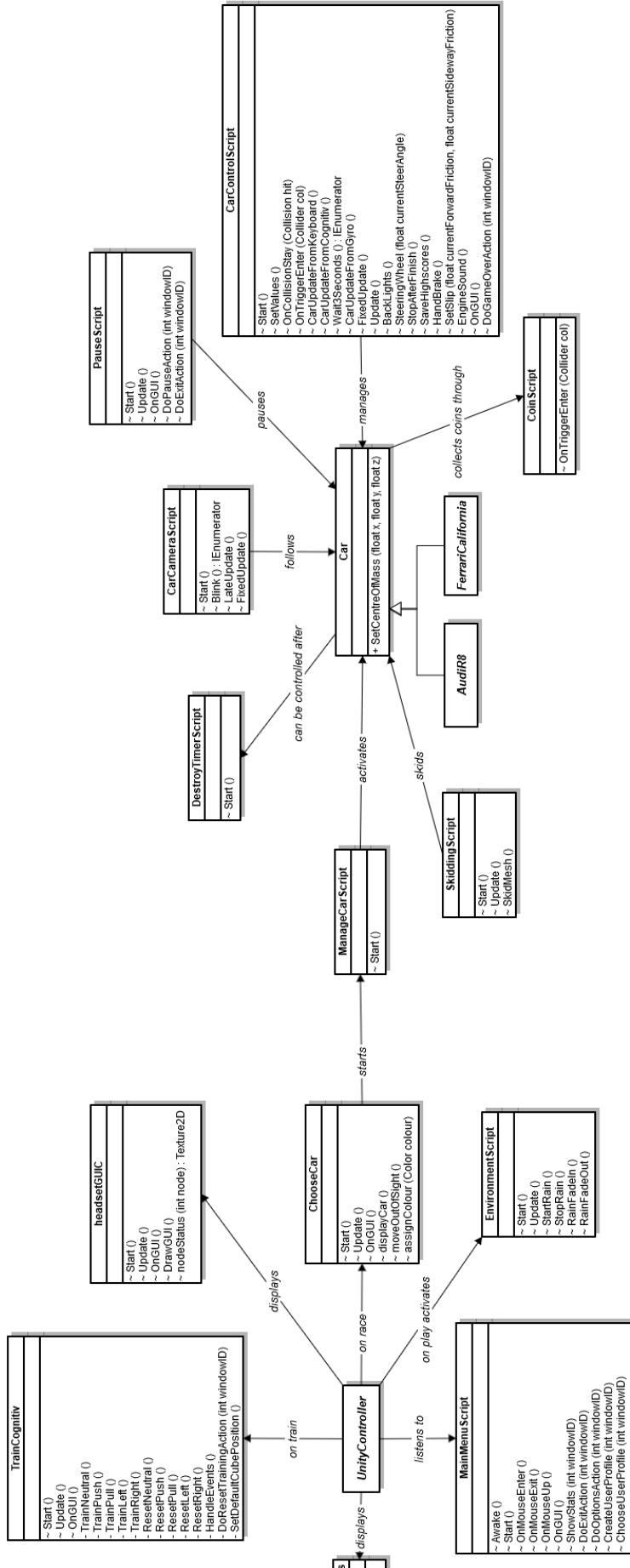


Figure 4.2: Domain class diagram for CogniDriver. The instance variables have been omitted for simplification.

Chapter 5

Implementation

This chapter is going through some of the implementation details during the project development phase of the project.

5.1 Interfacing with Emotiv

The Unity 3D plugin package from Emotiv comes in a suite of 11 C# files which we will describe in a bit more detail. In order to get access to this data, an `EPOCManager` object was created. This object contains all the plugins in the package apart from `EdkDll`, `EmoState`, `EmoProfileManagement` and `LabelDraw`. In the following subsections, we are going to provide some very short code snippets since these might prove useful in the future for someone wishing to develop a game using Unity and Emotiv.

5.1.1 EdkDll

Contains the implementation of the Emotiv API.

5.1.2 EmoAffectiv

This class gives access to the player emotions described in section 3.1 through calls that directly access one of the public instance variables. For example,

```
EmoAffectiv.frustrationScore
```

returns a value representing how frustrated the user is.

In CogniDriver, frustration is used to get the rain started once the score is higher than 50% and the excitement is lower than 50%.

5.1.3 EmoCognitiv

It is a class containing methods to start or reset the training. It automatically handles events which start, reset, erase, declare successful or failed training. We have added accessor methods to obtain the current action, its power and the trained actions. We have also added a GUI method, `DoTrainingCompleteAction(int windowID)` which pops up the window asking the user whether the training should be accepted or rejected and it handles the response appropriately. Before starting the training though, the 4 actions (push, pull, left, right) have to be activated through a call such as:

```
EmoCognitiv.EnableCognitivAction(EmoCognitiv.cognitivActionList[6],  
true);
```

where 6 represents the index of *right* action. At the end, the list of 4 actions has to be set through a call to

```
EmoCognitiv.EnableCognitivActionsList();
```

The skill for the *push* action is obtained as follows:

```
Single pushSkill = EmoEngine.Instance.CognitivGetActionSkillRating(  
(uint)EmoUserManagement.currentUser,  
EmoCognitiv.cognitivActionList[1]);
```

5.1.4 EmoEngine

'Is the logical abstraction of the functionality that Emotiv provides in edk.dll.' [emo, 2014a]

5.1.5 EmoEngineInst

It is used most often to change the connection method from live data to simulated data (through the use of EmoComposer (subsection 3.1.3)).

5.1.6 EmoExpressiv

The EmoExpressiv class is useful in order to get access to the player's facial expressions through simple calls to the public instance variables, such as the following which checks whether the player did a left wink:

```
EmoExpressiv.isLeftWink
```

In CogniDriver, left wink is used to change the camera view. Clenching teeth will action the car's handbrake.

5.1.7 EmoGyroData

Gives access to the head position in relation to the circle observed in figure. One may also obtain the x and y coordinates of the gyro. However, we could not find the maximum and minimum values, nor are there 2 online references which give the same numbers.

5.1.8 EmoProfileManagement

This class is useful to handle the player profiles. In CogniDriver, we have limited the number of player profiles to 10 because of the following reasons:

- The training for a saved profile can take a few MB;
- Unity does not contain a drop down list as a GUI element;
- Unity does not allow overflow of windows.

When the game starts, the list of user profiles (files saved with the .up extension) is loaded through a call to:

```
EmoProfileManagement.Instance.LoadProfilesFromFile();
```

A new profile is added by calling:

```
EmoProfileManagement.Instance.AddNewProfile(playerName);
```

An existing profile is deleted by calling

```
EmoProfileManagement.Instance.DeleteProfile(selectedPlayer);
```

Profile data is saved after training by a call to

```
EmoProfileManagement.Instance.SaveCurrentProfile();
```

followed by

```
EmoProfileManagement.Instance.SaveProfilesToFile();
```

On selection of a profile, this is set by calling

```
EmoProfileManagement.Instance.SetUserProfile(selectedPlayer);
```

5.1.9 EmoState

Represents the emotional status of the user at a given time [emo, 2014a]. This class is used by EmoAffectiv, EmoExpressiv and EmoCognitiv mainly. We did not need to call any functions directly from this class.

5.1.10 EmoUserManagement

Keeps track of the current user, the current number of user profiles and also fires events when a USB dongle is inserted or removed. I did not need to use this class directly apart from obtaining the current user by calling

```
EmoUserManagement.currentUser
```

5.1.11 LabelDraw

We did not need to use this class while developing CogniDriver.

5.2 Scene description

In Unity, a scene contains all the objects for a part of the game such as a level or a different screen. For CogniDriver, we have split the game in 6 scenes. All of the scenes, apart from the splash screen, will display the sensor contact quality, current action and the power of the current action. This can be observed in figure 5.3.

See figure 5.1 for an example of how the scenes interact with each other.

5.2.1 CarChoiceScene

The purpose of this scene is to allow the player to choose a car model and colour. At the moment, I have only used 2 car models. I had three of them in mind but one of the models did not allow me to modify the grouping of its components. Figure 5.2 shows how this looks like in the game.

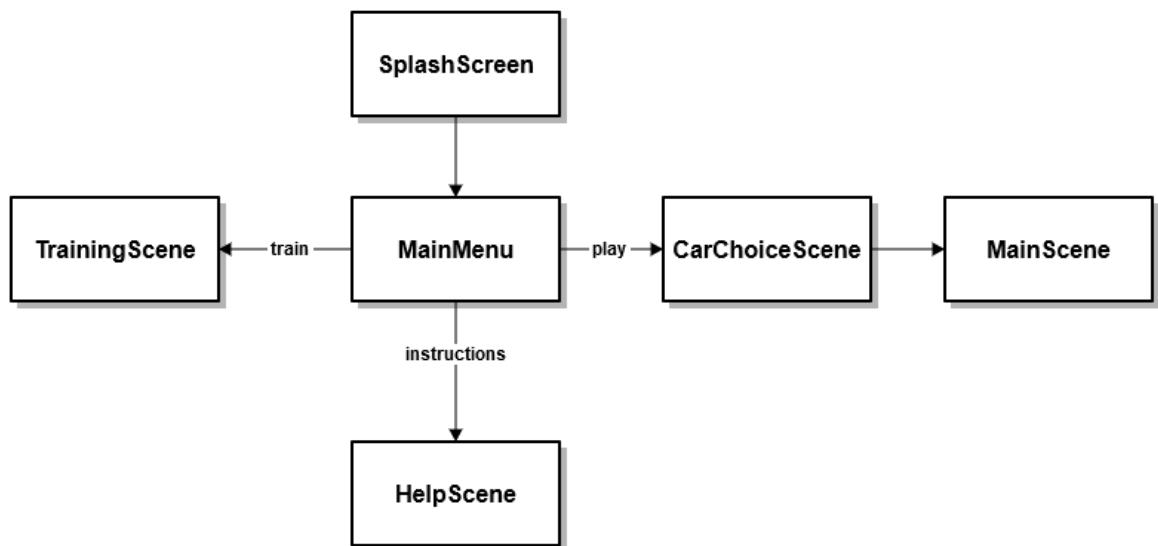


Figure 5.1: Scene interaction diagram from the moment the application starts loading.



Figure 5.2: Car choice scene



Figure 5.3: Main menu scene

5.2.2 HelpScene

Contains simple instructions describing how the game is to be played.

5.2.3 MainMenu

This scene allows the player to:

- manage the player profiles;
- start the race;
- train a player profile to be played in Cognitiv mode;
- see the highscores for Keyboard and Cognitiv mode in the *Statistics* tab;
- read instructions about training and playing the game;
- adjust the sound effects and music volume, fullscreen enabling and select the play mode;
- exit the game.

Figure 5.3 shows an example of how the main menu scene looks like.

5.2.4 MainScene

This is the proper game play where the player is controlling the car around the track. It can be observed in figure 5.4.



Figure 5.4: Main scene

5.2.5 TrainingScene

Its aim is to provide a place where the player can train the 5 actions: neutral (staying relaxed), push, pull, left, right. The training may also be reset for each action in turn. The user will receive visual feedback from the animated car on how well the training has been done. A screenshot can be observed in figure 5.5.

5.3 Notes

We took the decision of having the Cognitiv training done inside the game so that a new player would not need to open another application, for example the Emotiv Control Panel, in order to train the profile.

For passing variables between different scenes, Unity's PlayerPrefs module has been very useful. One example of its use is in saving the highscores. We will go in a bit more detail about what the saving highscores process is.

1. Get the current number of entries in the highscore list.
2. If it is >10, check if current entry is better than existing ones. If it is, record it.
3. A top 10 value is stored as:

```
key = topKeyPrefix;
value = String( playerName; elapsedTimeInSeconds; coins )
```

4. The entry is a fraction of the elapsed time in seconds and the number of collected coins.

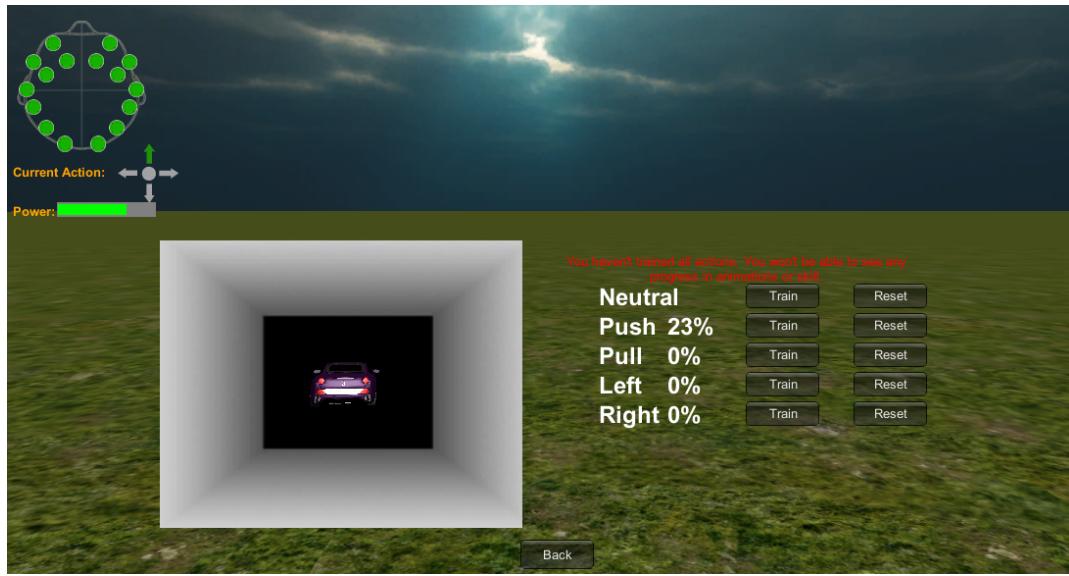


Figure 5.5: Training scene

5. Find the position where the new entry should be inserted.
6. Shuffle previous entries if necessary.
7. Finally, insert the new value and update the number of entries in the highscores table.

The user has access to some screen helpers during the time they are playing the game. These are:

- a speedometer which displays the current speed in both analog and digital mode;
- a square minimap which shows where the car is placed on the map when it is seen from above;
- the elapsed time since the start of the game;
- the number of coins collected.

5.3.1 Gyro mode

Playing the game while using the headset's gyroscope is doable. However, most of the times the user is getting eyes off the screen meaning they may not see where and how the car is moving.

Chapter 6

Testing and evaluation

The initial plan was to have a total of 10 participants to take part in a 2-hour testing process. However, some people dropped out while other people expressed interest, so the final number of participants was 13. All the participants were university students. The testing took place in week 4 of second semester, more precisely 17th to 20th of February.

At the time of user testing, the game was finished with the exception of the car model and colour choice.

In this chapter, we are going to discuss the results of the user testing. The used questionnaire can be found in appendix A.

For the purposes of testing the projects, the participants were firstly asked to play the game in keyboard mode. Following this, they were asked to fill in a NASA TLX (Task Load Index) test so that we would have a measurement of how demanding this task was for the user. Next, the Emotiv EPOC headset was fit on the participant's head and the training process would begin. After feeling confident enough for the first action (push), the user could see it working in the actual game play. We insisted on properly training 2 actions: push - for acceleration - and left - for a left turn. The reason why left was necessary is because the track does a left turn on a slightly elevated terrain. Less than 5 users have probably had enough time to train the other 2 states (pull and right) as well. Again, after the 2 hours were almost up, the participant was asked to take another NASA TLX test, this time taking into account how demanding the process was for Cognitiv game play. Finally, the participant was asked to fill in the user study questionnaire in appendix A.

The following sections will describe the results of the user testing. In section 6.6, we will highlight what was changed in CogniDriver following the participants' answer, in the period of 2 weeks left before the project demonstrations period begun. Figure 6.1 shows one of the trained user profiles during testing.

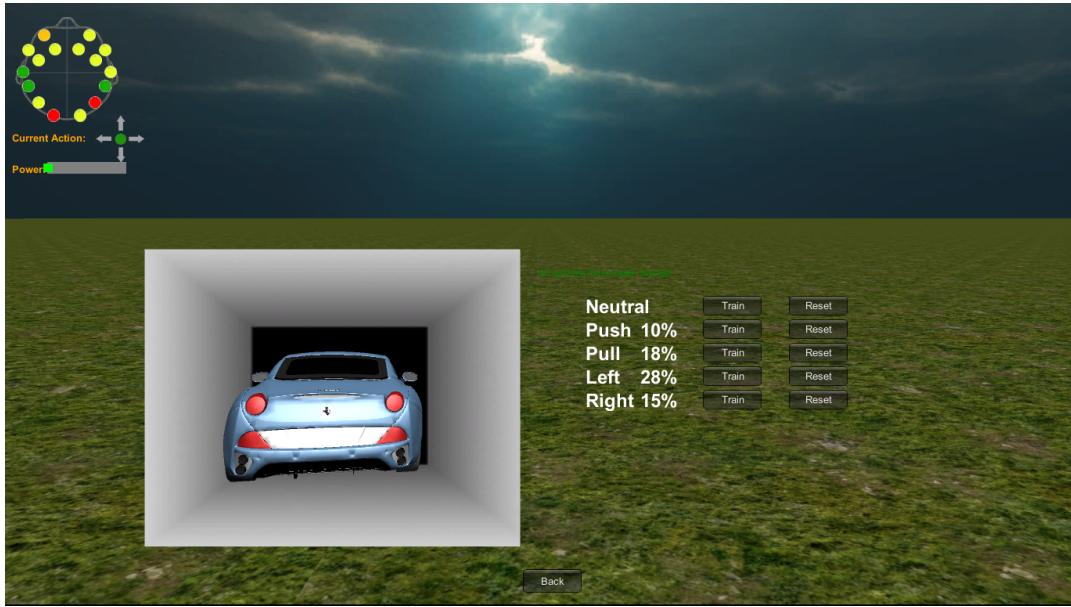


Figure 6.1: Trained user profile during user testing

6.1 Observations

Some of the participants displayed better skill in training the headset if the testing was done in the morning. This is compared to evenings, when some of the participants complained about being tired and not being able to concentrate properly.

The testing was done using a cube as the training object. In my opinion, this would have made training consistent with people used to the Emotiv EPOC Control Panel or other applications which require Cognitiv training.

It may be difficult to switch to a different mental state when training a new action. As a result, new training data for an action, might override data for a previous action. This happened to me a few times and it did happen with some of the participants as well.

When training or playing the game, if the car is being moved, the enthusiasm generated by that may make the player lose concentration.

6.2 NASA TLX results

The test was taken using the [Vertanen, 2010] online version developed by Keith Vertanen. It produces the analysis of the input data for each person at the end of the test.

For keyboard mode, the overall mean difficulty was 51.611% with a standard deviation of 17.494%.

For Cognitiv mode, the overall mean difficulty was 90.571% with a standard deviation of 13.964%.

This tells us the participants have found the game much easier to play in keyboard mode compared to Cognitiv mode which was very demanding.

6.3 Analysis of user questionnaire answers

The user questionnaire aimed to measure things such as the comfort of wearing the headset and how demanding the training and playing were, especially in Cognitiv mode.

The mean of the comfort of wearing the headset was 5.692 out of 10 with an approximately equal spread of scores. Most of the participants have started feeling the discomfort after 20-40 minutes of wearing the headset.

For the questions regarding the demand of the process, the participants' scores were quite unanimous. The training demand obtained a score of 4.23/10, the play demand obtained a score of 3.153/10 and the ease demand obtained 2.153/10. The similarity between the keyboard play and Cognitiv play was of 2.307/10 suggesting the two modes are not similar.

6.4 Good points about the game

The first point to make was that users found it easy to focus on the task because driving is a real life skill and it is easy to understand. Many times, participants made correlations between the driving game and the real life skill.

Raining when frustrated was considered a great feature by most.

The graphics were generally appreciated and feedback was received on the track as being sensible.

Some players enjoyed the funny game physics because there were opportunities for cheating. However, these opportunities have been reduced now and we have tried improving the physics so that not as much skidding would occur.

The training interface was considered clear and understandable.

6.5 Bad points about the game

It was hard to tell which way is the right direction of driving if the car skid and the player got disorientated.

The car accelerated too fast and some of the game physics were peculiar.

Some considered it would have been useful to see the brainwaves for each channel since this might have provided more feedback on how the training was going and how to concentrate on completing the task.



Figure 6.2: A print screen of the message which will occur in case the player tries to follow a shortcut to the finish line

The training process requires concentration and it can take a long time. Some people were disturbed by the amount of clicking going on in order to start training, accept or reject it and then start again. It also got the participant distracted and it was breaking the concentration level. Sometimes, I would do the clicking for the participant so I could catch the events when the object was really being moved. Complaints were received regarding the training as being too stressful, frustrating and not rewarding. The amount of clicking, although disliked by many, is not something which could be changed since the training process can vary in duration. It was suggested that an approach to this might be to automatically accept a training if it meets some set guidelines but then the participant is quite out of control of the process. In addition, it might make the movement more difficult to control.

Some invisible colliders¹ from around the mountains were slightly touching the road side. This has now been fixed.

The driver seat view was not always enjoyable, especially when the sensor contact quality was not perfect and the blink would be mistaken for a left wink which would then cause the camera view to change.

6.6 What should be improved and what has been implemented?

It has been suggested by more than a half of the participants that training would be easier if it was done with a car object rather than a cube as this made them more properly visualise the action. It also made it easier to activate an action while playing the game. This has been implemented.

Checkpoints were another suggestion so that players could be stopped from cheating and taking a short route to the finish line. This has been implemented with the help of invisible colliders which were being triggered when the car would pass through them. A total of 17 checkpoints have been inserted. At the finish line, if the number of checkpoints the player has been through is different than 17, the message in figure 6.2 will appear, giving the option to restart the game.

¹An invisible object that is used either as a barrier to stop another object from going through it or which is used as a trigger to detect whether another object has reached that point

The current action used to be displayed as text, but the suggestion was to have a system of arrows being highlighted for each type of movement. This has been implemented.

One suggestion was to add some coins to the road that the player could collect. This has been implemented and it solves the issue of the users not knowing which is the right direction of going. It also obliges the player to stick more to the centre of the road and it provided an extra way of deciding the highscores, besides the elapsed time.

There used to be a quite annoying reverse sound which was suggested to be removed by the majority of participants. It was seen as distracting when the player was trying to concentrate on the action. The motor sound though, did not seem as stressful.

There used to be no way of restarting the race from the keyboard. This is now done when the player is pressing the 'R' key. Also, using 'T' will make the car respawn at the last checkpoint it has been through just in case the player gets lost around the environment. It is not something that should happen too often, but it might occur.

The things which have not been implemented had to do with not having access to the raw EEG data or them simply not being feasible or important.

From a talk after the testing session was complete, it was suggested it would be good to give the car different speeds on grass and road. This has been implemented. The car accelerates about 2 times slower when it gets on the grass area.

Chapter 7

Conclusions

This chapter will describe what I have learnt, what I could have improved and some final thoughts about the project.

7.1 Personal development

Before starting this project, I have not been programming in C# nor did I have any experience in working with Unity. Every time I try a new programming language or technology, I get a confidence boost because I see I can handle the situation and the experience will be beneficial in my future career. What I wish I had done about Unity, was to check the default access modifiers, especially for methods. Also, the order of the things in the Update function might matter.

An application should be designed with software design patterns in mind. This will make it more secure and it will careful consideration on behalf of the developer.

Documentation is important since other people might need to continue your work or they might want to adjust certain parts in order to fit their purposes.

It is useful to keep a logbook of the work one is doing and to write down the problems they encounter. This will provide a good retrospective about the things they have learnt or they should improve for the next application.

It is important to use suitable names at all times.

7.2 Summary

Nowadays, we have exercised using the keyboard for long periods of time, so it seems easier to control a game in this mode, but if we put more practice into playing with an EEG and properly training the Cognitiv suite, we could achieve same results with much less of an effort. [Shinji Nishimoto, 2011] shows it is possible to use an fMRI (functional

magnetic resonance imaging) to see quite similar pictures with what the participant is seeing while watching a movie. Given this, in a relatively short period of time, brain devices may be able to read one's mind.

My personal feeling about this project is that it has helped me observe how design decisions affect the development of an application. In addition, the more time passes from a bad decision, the longer it will take to fix it. I am happy to think that my next project will benefit from the lessons learnt here and I also hope the views I have provided in this report will be helpful to other people with similar circumstances.

At the time of this writing, the full application source code and executable binary are available at <https://github.com/florescu/CogniDriver>.

Bibliography

- [emo, 2014a] (2014a). *Emotiv Software Development Kit User Manual for Release 2.0.0.20.* Emotiv.
- [emo, 2014b] (2014b). *EPOC User Manual.* Emotiv.
- [A. Ferreira, 2006] A. Ferreira, T. F. Bastos-Filho, M. Sarcinelli-Filho. F. A. Cheein. J. Postigo. R. Carelli. (2006). Teleoperation of an industrial manipulator through a tcp/ip channel using eeg signals. *2006 IEEE International Symposium on Industrial Electronics*, 4:3066 – 3071. doi: 10.1109/ISIE.2006.296105.
- [Adelson, 2011] Adelson, M. (2011). Experimenter epoch - an experimentation and mind-reading application for the emotiv epoch. http://compmem.princeton.edu/experimenter/ExperimenterReport.html#_Toc290642075.
- [Andrew T. Campbell and Raizada, 2010] Andrew T. Campbell, Tanzeem Choudhury, S. Hu. H. Lu. M. K. Mukerjee. M. Rabbi. and Raizada, R. D. S. (2010). Neurophone: Brain-mobile phone interface using a wireless eeg headset. *MobiHeld/ACM*.
- [BioSemi, 2015] BioSemi (2015). Reference sensors. <http://www.biosemi.com/faq/cms&drl.htm>.
- [Carmen Vidaurre, 2009] Carmen Vidaurre, B. B. (2009). Towards a cure for bci illiteracy. *BMC Neuroscience*. doi:10.1186/1471-2202-10-S1-P85.
- [Choon Guan Lim, 2012] Choon Guan Lim, Tih Shih Lee, C. Guan. D. S. S. Fung. Y. Zhao. S. S. W. Teng. H. Zhang. K. R. R. Krishnan. (2012). A brain-computer interface based attention training program for treating attention deficit hyperactivity disorder. *PLoS ONE* 7(10): e46692. doi: 10.1371/journal.pone.0046692.
- [Emotiv, 2015a] Emotiv (2015a). Emotiv reference sensors. <https://www.emotiv.com/forum/forum4/topic3409/messages/>.
- [Emotiv, 2015b] Emotiv (2015b). Emotiv training. http://emotiv.com/forum/messages/forum4/topic114/message392/?phrase_id=464885#message392.
- [Fabien Lotte, 2008] Fabien Lotte, Yann Renard, A. Lécuyer. (2008). Self-paced brain-computer interaction with virtual worlds: A quantitative and qualitative study ‘out of the lab’. *4th international Brain Computer Interface Workshop and Training Course*. <https://hal.inria.fr/inria-00304340>.
- [Francesco Carrino, 2012] Francesco Carrino, Joel Dumoulin, E. Mugellini. O. A.

- Khaled. R. Ingold. (2012). A self-paced bci system to control an electric wheelchair: evaluation of a commercial, low-cost eeg device. *Biosignals and Biorobotics Conference (BRC), 2012 ISSNIP*. doi: 10.1109/BRC.2012.6222185.
- [Gall, 1992] Gall, J. (1992). Method and system for altering consciousness. <http://www.google.com/patents/US5123899>. US Patent 5,123,899.
- [Inc., 2010] Inc., A. T. (2010). Emolens. https://www.youtube.com/watch?v=E9_XZlHoSp0.
- [Lang, 2012] Lang, M. (2012). Investigating the emotiv epoc for cognitive control in limited training time.
- [Minkyu Ahn, 2013] Minkyu Ahn, Hohyun Cho, S. Ahn. S. C. Jun. (2013). High theta and low alpha powers may be indicative of bci-illiteracy in motor imagery. *PLoS ONE* 8(11): e80886. doi:10.1371/journal.pone.0080886.
- [Monster, 2015] Monster, M. (2015). Car physics for games. <http://www.asawicki.info/Mirror/Car%20Physics%20for%20Games/Car%20Physics%20for%20Games.html>.
- [Murray, 2014] Murray, J. W. (2014). *C# Game Programming Cookbook for Unity 3D*. CRC Press.
- [Ning-Han Liu and Hsu, 2013] Ning-Han Liu, C.-Y. Chiang. and Hsu, H.-M. (2013). Improving driver alertness through music selection using a mobile eeg to detect brainwaves. *Sensors*. <http://www.fermentas.com/techinfo/nucleicacids/maplambda.htm>.
- [Reinhold Scherer, 2012] Reinhold Scherer, Markus Prll, B. Allison. . G. R. Mller-Putz. (2012). New input modalities for modern game design and virtual embodiment. *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*. doi: 10.1109/VR.2012.6180932.
- [Sam Fok, 2011] Sam Fok, Raphael Schwartz, M. Wronkiewicz. C. Holmes. J. Zhang. T. Somers. D. Bundy. E. Leuthardt. (2011). An eeg-based brain computer interface for rehabilitation and restoration of hand control following stroke using ipsilateral cortical physiology. *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. doi: 10.1109/IEMBS.2011.6091549.
- [Shinji Nishimoto, 2011] Shinji Nishimoto, An T. Vu, T. Naselaris. Y. Benjamini. B. Yu. J. L. Gallant. (2011). Reconstructing visual experiences from brain activity evoked by natural movies. *Current Biology*, (21):1641–1646. doi: <http://dx.doi.org/10.1016/j.cub.2011.08.031>.
- [Smirnoff, 2013] Smirnoff (2013). Mindtunes. <https://www.youtube.com/watch?v=PgfxKZiSCDQ>.
- [Studio, 2014] Studio, W. B. (2014). Unity tutorial. http://walkerboystudio.com/html/unity_course_start_here_free_.html.
- [Tutorials, 2014] Tutorials, F. (2014). Unity car game tutorial. <https://www.youtube.com/watch?v=c5Snsi68xzE&list=PL67XFC3MYQ6K0PXSad15xFrhxNL76r4Te>.

- [Unity, 2015] Unity (2015). Unity game list. <https://unity3d.com/gallery/made-with-unity/game-list/profile>.
- [Vertanen, 2010] Vertanen, K. (2010). Nasa tlx test. <http://www.keithv.com/software/nasatlx/nasatlx.html>.
- [Wikipedia, 2015] Wikipedia (2015). Battlestar galactica online. http://en.wikipedia.org/wiki/Battlestar_Galactica_Online.
- [Xiaorong Gao, 2003] Xiaorong Gao, Dingfeng Xu, M. Cheng. S. Gao. (2003). A bci-based environmental controller for the motion-disabled. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):137 – 140. doi: 10.1109/TNSRE.2003.814449.

Appendix A

CogniDriver - User Study

1. On a scale of 1 to 10, how comfortable was the headset on your head?

Extremely disturbing	1	2	3	4	5	6	7	8	9	10	Extremely comfortable
-------------------------	---	---	---	---	---	---	---	---	---	----	--------------------------

2. Have you encountered any discomfort in wearing the headset?

3. On a scale of 1 to 10, how demanding did you find the training process?

Extremely demanding	1	2	3	4	5	6	7	8	9	10	Extremely easy
------------------------	---	---	---	---	---	---	---	---	---	----	-------------------

4. On a scale of 1 to 10, how demanding did you find the Cognitiv game play?

Extremely demanding	1	2	3	4	5	6	7	8	9	10	Extremely easy
------------------------	---	---	---	---	---	---	---	---	---	----	-------------------

5. On a scale of 1 to 10, how easy was it for you to control the game when using the Cognitiv mode?



6. On a scale of 1 to 10, how similar do you find controlling the game in keyboard mode versus Cognitiv mode?



7. What did you like about the game as a whole?

8. What did you not like about the game as a whole?

9. What improvements would you like to see in the game as a whole?

Appendix B

CogniDriver - 3D Object Sources

Below is a list of all the 3D objects that have been imported in the game:

Image	Source
	http://tf3dm.com/3d-model/audi-r8-69023.html
	http://archive3d.net/?a=download&id=8c488f86
	http://www.mediafire.com/download/cbdb5neezcva198/California.zip
	http://www.archibaseplanet.com/download/dc2c3978.html
	http://tf3dm.com/3d-model/house-10483.html
	http://tf3dm.com/3d-model/old-farm-house-91130.html
	http://archive3d.net/?a=download&id=6d06cd33