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 the testing and whose feedback was greatly appreciated.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Applications

Using BCI (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 exercising brain functions and improving concentration.

1.2 Aims and objectives

The aim of this project was to learn about the ... and falls 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 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 you 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, back, left, right through keyboard use;
- Reach access to the interpreted data of the developers' SDK;
- 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 a quick look over 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 reviews the obtained results, while Chapter 7 described the performed testing. Appendix 1 shows the user questionnaire used during the user trials. Finally, Chapter 8 contains the conclusions of this 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:

- Game deployment on a variety of platforms;
- Easy object importing;
- Object manipulation;
- Scene design;
- Creation of primitive shapes (sphere, cube, etc.);
- Collision detection;
- Automatic object updates;
- Game physics.

I 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 you change the source files 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) and it is scalable for larger projects.

Unity has been used to create games such as Assassin's Creed Identity, Need for Speed World and Battlestar Galactica Online.

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, colours 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.

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.

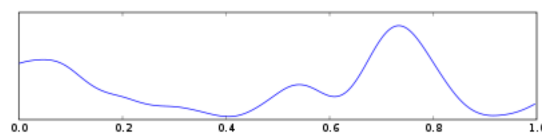
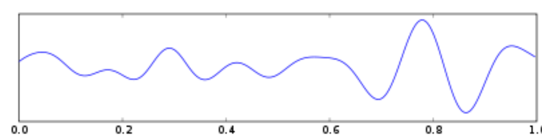
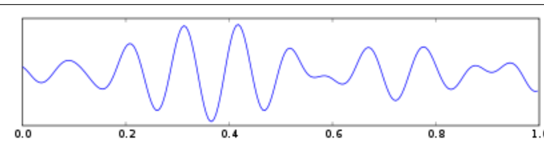
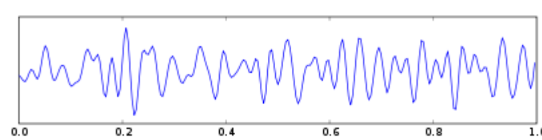
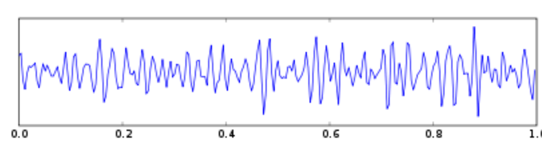
Name	**Frequency (Hz)**	**Occurs in**	**Image Representation**
Delta	<4	deep sleep , unconsciousness, deep anaesthesia, hypoxia	
Theta	4 - 7	stress, unconsciousness, deep relaxation	
Alpha	8 - 15	relaxation	
Beta	16 - 31	conciousness, alertness, thinking, sensory stimulation	
Gamma	32 +	selective attention,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

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

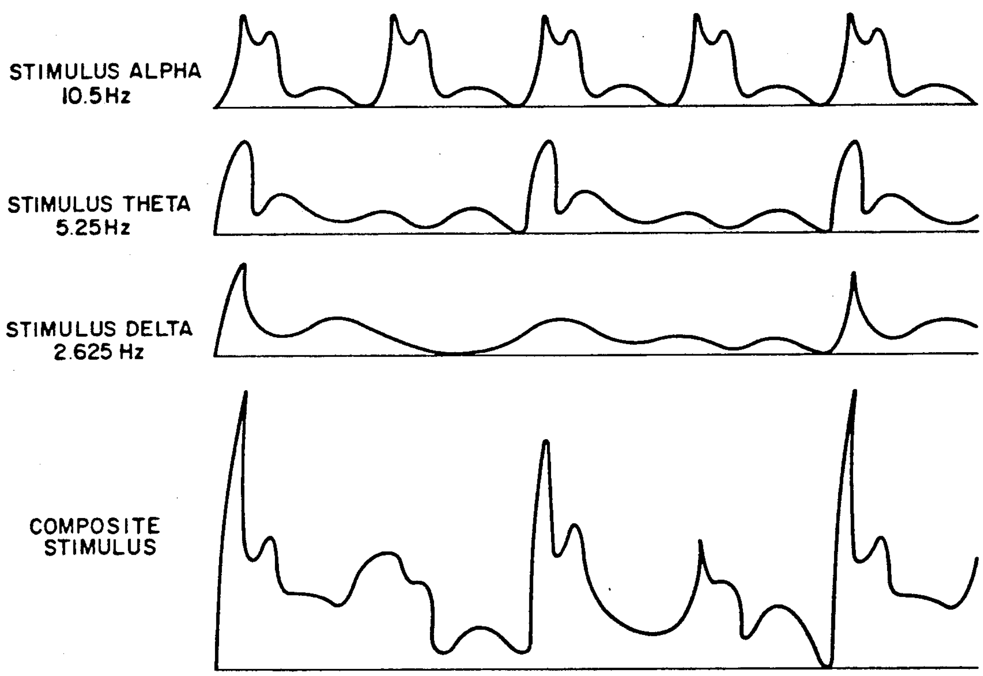


Figure 2.1: A composition of alpha, theta and delta brainwaves. Credit: [Gall, 1992]

Chapter 3

Introducing the Emotiv EPOC

3.1 SDK description

3.2 Training

Chapter 4

Design

4.1 Overview

4.2 Design Patterns

Chapter 5

Implementation

5.1 Interfacing with Emotiv

5.2 Scene description

5.3 Interesting aspects of development

Chapter 6

Results and analysis

Chapter 7

Testing and evaluation

7.1 User testing

Chapter 8

Conclusions

8.1 Achievements

8.2 Personal development

8.3 Further work

8.4 Summary

Bibliography

[Gall, 1992] Gall, J. (1992). Method and system for altering consciousness. US Patent 5,123,899.

[Ning-Han Liu and Hsu, 2013] Ning-Han Liu, C.-Y. C. and Hsu, H.-M. (2013). Improving driver alertness through music selection using a mobile eeg to detect brainwaves. *Sensors*.

Appendix A

Example of operation

An appendix is just like any other chapter, except that it comes after the appendix command in the master file.

One use of an appendix is to include an example of input to the system and the corresponding output.

One way to do this is to include, unformatted, an existing input file. You can do this using `\verbatiminput`. In this appendix we include a copy of the C file hello.c and its output file hello.out. If you use this facility you should make sure that the file which you input does not contain `TAB` characters, since LaTeX treats each `TAB` as a single space; you can use the Unix command `expand` (see manual page) to expand tabs into the appropriate number of spaces.

A.1 Example input and output

A.1.1 Input

(Actually, this isn't input, it's the source code, but it will do as an example)

A.1.2 Output

A.1.3 Another way to include code

You can also use the capabilities of the `listings` package to include sections of code, it does some keyword highlighting.