USE OF BRAIN COMPUTER INTERFACES IN MAIN CONTROL OF VIDEO GAMES AND IN DYNAMIC DIFFICULTY ADJUSTMENT

(BEYIN BILGISAYAR ARAYÜZLERININ VIDEO OYUNLARININ ANA KONTROLÜNDE VE DINAMIK ZORLUK AYARLAMASINDA KULLANIMI)

by

Elifsu FİLİZ, B.Sc.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

Supervisor: Assist. Prof. Dr. R. Burak Arslan

Mar 2021

This is to certify that the thesis entitled

USE OF BRAIN COMPUTER INTERFACES IN MAIN CONTROL OF VIDEO GAMES AND IN DYNAMIC DIFFICULTY ADJUSTMENT

prepared by **Elifsu FİLİZ** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering** at the **Galatasaray University** is approved by the

Examining Committee:

Assist. Prof. Dr. Reis Burak Arslan (Supervisor)
Department of Computer Engineering
Galatasaray University -------------------------

Assoc. Prof. Dr. Özlem Durmaz İncel
Department of Computer Engineering
Galatasaray University -------------------------

Assoc. Prof. Dr. Tuğba Dalyan
Department of Computer Engineering
İstanbul Bilgi University -------------------------

Date: -------------------------

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Reis Burak Arslan, my friends Özge Günaydın and Müjde Akdeniz, and my family for their support; and my husband Tahir Tekin Filiz for his incredible help in every possible way.

Mar 2021
Elifsu FİLİZ

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii

TABLE OF CONTENTS . . . iv

LIST OF FIGURES . . . vii

LIST OF TABLES . . . ix

ABSTRACT . . . xi

RÉSUMÉ . . . xii

ÖZET . . . xiii

LIST OF ABBREVIATIONS . . . xiv

1 INTRODUCTION . . . 1

1.1 Applications . . . 1

1.2 Challenges . . . 2

2 LITERATURE REVIEW . . . 4

2.1 Categorization of BCIs . . . 4

2.1.1 Dependent and Independent BCI . . . 4

2.1.2 Invasive and Non-invasive BCI . . . 4

2.1.3 Synchronous and Asynchronous (Self-Paced) BCI . . . 4

2.1.4 BCIs in terms of Brain Activity . . . 4

2.2 Brain waves categorized by their frequency bands . . . 5

2.3 Control Signal Modalities . . . 6

2.3.1 Evoked Signals . . . 6

2.3.2 Spontaneous Signals . . . 7

2.3.3 Hybrid Signals . . . 7

2.4 BCI as a Control Mechanism for Video Games . . . 8

2.5 BCI as Main Control in Videogames 9
2.6 Arranging Game Difficulty 13
2.7 Speller 16
3 MATERIALS AND METHODS 19
3.1 Emotiv EPOC 19
3.1.1 Electrode Placement System : 10 - 20 System 19
3.1.2 Electrode Placement System : 10 - 10 System 20
3.1.3 Emokit Driver 21
3.2 Unity Game Engine 22
3.3 Methods 22
3.3.1 Preprocessing Methods 22
3.3.1.1 Mean Normalization 22
3.3.1.2 Filtering 23
3.3.2 Feature Extraction Methods 23
3.3.2.1 Periodogram 24
3.3.2.2 Discrete Wavelet Transform 24
3.3.2.3 Entropy 24
3.3.2.4 Zero and Mean Crossings 25
3.3.2.5 Statistical Features For Biosignals 25
3.3.3 Classification Methods 26
3.3.3.1 Canonical Correlation Analysis 26
3.3.3.2 Multilayer Perceptron 28
3.3.3.3 Gradient Boosting 29
3.4 Implemented Games 30
3.4.1 Connect 4 30
3.4.2 Guitar Hero 32
4 RESULTS AND DISCUSSION 34
4.1 Experiments 34

4.1.1 Comparison of Different Colored Stimuli 34
4.1.2 Selection of Best 5 Stimulus Frequencies 35
4.1.2.1 Highest Accuracies 36
4.1.2.2 Minimum Elapsed Time 37
4.1.2.3 Choosing Best Frequencies 37
4.1.3 Comparing Specially Constructed Turkish Speller and Randomly Generated Speller 38
4.2 Tests on Implemented Games 42
4.2.1 Connect 4 42
4.2.2 Guitar Hero 43
5 CONCLUSION 50
REFERENCES 53
BIOGRAPHICAL SKETCH 54

LIST OF FIGURES

Figure 2.1 Brain waves visualization with different frequency bands (Siuly, 2016). 5

Figure 2.2 Mental states of a player according to game difficulty and player skill level (Nijholt, 2009). 8

Figure 2.3 Affective states (Russell 1980). 14

Figure 2.4 Three stages of the speller (Cecotti 2010). 17

Figure 2.5 Speller in rhombus layout (Valbuena et al. 2008). 18

Figure 3.1 Emotiv EPOC headset. 19

Figure 3.2 Emotiv EPOC electrode locations. 19

Figure 3.3 The 10 - 20 system electrode positioning. 20

Figure 3.4 The 10 - 10 system electrode positioning 21

Figure 3.5 Basic filter types. 23

Figure 3.6 CCA usage in determination of the target stimulus. 27

Figure 3.7 Connect 4 game board. 30

Figure 3.8 BCI based Connect 4 game board. 31

Figure 3.9 Signal processing loop. 32

Figure 3.10Guitar Hero game. 33

Figure 4.1 PSD plots of O1 channel with different colored stimuli. 34

Figure 4.2 Comparison of frequencies in terms of elapsed time. 35

Figure 4.3 Comparison of frequencies in terms of accuracy. 36

Figure 4.4 Specially constructed Turkish speller. 39

Figure 4.5 Randomly generated speller. 39

Figure 4.6 Changes in game difficulty when the game has an easy start. 47

Figure 4.7 Changes in game difficulty when the game has a medium start. . . . 48

Figure 4.8 Changes in game difficulty when the game has a difficult start. . . . 49

LIST OF TABLES

Table 2.1 Publications on BCI based video games 13

Table 2.2 Publications on EEG based DDA 16

Table 3.1 Letters of electrode placements and corresponding brain lobes in 10 - 20 system. 20

Table 3.2 Letter codes of extra electrodes in 10 - 10 system. 21

Table 3.3 Parameter values for MLP classifier. 29

Table 3.4 Parameter values for gradient boosting classifier. 30

Table 4.1 5 best frequencies in terms of accuracy and their elapsed time for each subject and session 36

Table 4.2 5 best frequencies in terms of elapsed time and their accuracy for each subject and session 37

Table 4.3 Number of times frequencies were ranked in top 5 38

Table 4.4 The actual elapsed time typing the phrases using the speller with Turkish layout versus the optimal time (seconds) 40

Table 4.5 The actual elapsed time typing the phrases using the speller with random layout versus the optimal time (seconds) 40

Table 4.6 The actual elapsed time typing the letters using the speller with Turkish layout versus the optimal time (seconds) 41

Table 4.7 The actual elapsed time typing the letters using the speller with random layout versus the optimal time (seconds) 41

Table 4.8 The ratio of correctly classified stimulus frequencies to all classifications. 42

Table 4.9 The performance of 3-class classifier on training, validation and test sets. 44

Table 4.10 The performance of arousal level classifier on training, validation and test sets. 44

Table 4.11 The performance of valence level classifier on training, validation and test sets. 45

Table 4.12 Test results of 3-class classifiers. 45

Table 4.13 Test results of valence and arousal classifiers. 46

ABSTRACT

Brain computer interface is a communication pathway between humans and computers. The acquired EEG signals are translated into a command using signal processing, machine learning and statistical methods. Brain computer interfaces are generally designed for patients with severe disabilities such as motor impairments, but there are also many applications for healthy users. In this study, two BCI based video games were developed. In one of the games, the game is fully controlled by making use of steady state visual evoked potentials. The average accuracy of the controls of this game was found to be 0.87. In the second game, even though the game is controlled by keyboard, the affective states of the player are predicted and the game difficulty is arranged accordingly. The valence and arousal predictions were tested in an offline analysis, which yielded 0.72 and 0.85 accuracy for valence and arousal levels respectively. The implemented dynamic difficulty adjustment system making use of the valence and arousal levels predicted in real time performed well enough to change the game difficulty to an enjoyable level.

Keywords : BCI, SSVEP, affective state recognition, video games, EEG

RÉSUMÉ

L'interface cerveau-ordinateur est une voie de communication entre les humains et les ordinateurs. Les signaux EEG acquis sont traduits en une commande en utilisant le traitement du signal, l'apprentissage automatique et des méthodes statistiques. Les interfaces cerveau-ordinateur sont généralement conçues pour les patients souffrant de handicaps sévères tels que des déficiences motrices, mais il existe également de nombreuses applications pour les utilisateurs en bonne santé. Dans cette étude, deux jeux vidéo basés sur BCI ont été développés. Dans l'un des jeux, le jeu est entièrement contrôlé en utilisant des potentiels évoqués visuels en régime permanent. La précision moyenne des commandes de ce jeu s'est avérée être de 0,87. Dans le deuxième jeu, même si le jeu est contrôlé par le clavier, les états affectifs du joueur sont prédits et la difficulté du jeu est arrangée en conséquence. Les prédictions de valence et d'excitation ont été testées dans une analyse hors ligne, qui a donné une précision de 0,72 et 0,85 pour les niveaux de valence et d'excitation respectivement. Le système d'ajustement dynamique de la difficulté implémenté utilisant les niveaux de valence et d'excitation prédits en temps réel a fonctionné suffisamment bien pour changer la difficulté du jeu à un niveau agréable.

Mots Clés : IOC, PÉVRP, reconnaissance de l'état affectif, jeux vidéo, EEG

ÖZET

Beyin bilgisayar arayüzü, insanlar ve bilgisayarlar arasındaki bir iletişim yoludur. Toplanan EEG sinyalleri; sinyal işleme, makine öğrenimi ve istatistiksel yöntemler kullanılarak bir komuta dönüştürülür. Beyin bilgisayar arayüzleri genellikle engelli hastalar için tasarlanmıştır, ancak sağlıklı kullanıcılar için de birçok uygulama vardır. Bu çalışmada, BCI tabanlı iki video oyunu geliştirilmiştir. Oyunlardan birinde oyun kontrolünün tamamı durgun hal görsel uyarılmış potansiyellerden yararlanılarak gerçekleştirilmiştir. Oyun kontrollerinin ortalama doğruluğu 0,87 olarak bulunmuştur. İkinci oyunda oyun klavye ile kontrol edilmektedir, ancak oyuncunun duygusal durumu tahmin edilip oyunun zorluğu buna göre düzenlenmiştir. Yapılan testlerde valans ve uyarılma seviyeleri, sırasıyla 0,72 ve 0,85 doğruluk oranıyla tahmin edilmiştir. Gerçek zamanlı valans ve uyarılma seviye tahminleri kullanan dinamik zorluk ayarlaması sisteminin oyunun zorluğunu eğlenceli seviyelere taşıyacak kadar iyi çalıştığı gözlemlenmiştir.

Anahtar Kelimeler : BBA, DHGUP, duygusal durum tanıma, video oyunları, EEG

LIST OF ABBREVIATIONS

BCI	Brain computer interface
CCA	Canonical correlation analysis
DDA	Dynamic difficulty adjustment
DWT	Discrete wavelet transform
ECG	Electrocardiography
EEG	Electroencephalography
FES	Functional electrical stimulation
FLDA	Fisher's linear discriminant analysis
GSR	Galvanic skin response
HV	High valence
HVHA	High valence high arousal
HVLA	High valence low arousal
ITR	Information transfer rate
LVHA	Low valence high arousal
LVLA	Lov valence low arousal
MLP	Multilayer perceptron
NZC	Number of zero crossings
PCA	Principle component analysis
SMR	Sensorimotor rhythm
SNR	Signal to noise ratio
SSVEP	Steady state visual evoked potential

1 INTRODUCTION

Brain Computer Interface is a communication channel between humans and computers. It is composed of hardware for acquiring electrical signals from brain, and software for interpreting these signals and converting them into commands.

BCIs are generally designed for patients with severe disabilities such as motor impairments. For instance, BCI based wheelchairs, robotic arms, spellers can be used by such patients to improve their quality of life.

BCIs can also be used by healthy people. There are many applications such as improving learning, controlling smart environments, and playing video games. In this thesis, two BCI based games are implemented for healthy users. In one of these games, BCI is used as the main control mechanism for the game, in the other one BCI detects the affective state of the player, and change the game difficulty accordingly.

In the following sections, applications of BCI, commonly faced challenges, and general information about BCI modalities are introduced.

1.1 Applications

BCIs are used in many different fields such as medicine, neuroergonomics and smart environment conrol, neuromarketing and advertisement, education and self-regulation, security and authentication, and finally games and entertainment. Abdulkader et al. (2015) explains how BCIs are used in these areas as follows.

In medicine, BCIs are used in detection, diagnosis and prevention of certain illnesses, as well as rehabilitation and restoration. For instance, BCIs can detect and diagnose tumors, brain and sleep disorders. They can prevent smoking, alcoholism and motion sickness. Patients who had a brain stroke or have psychological disorders can be rehabilitated. Disabled people can use BCI based robots to interact with their environment.

BCIs are also integrated to smart environments to offer further safety, luxury or control of the environment. BCIs can detect the cognitive state and improve workspace conditions accordingly, or can detect mental fatigue and warn user to take a break.

Commercial and political advertisements can be assessed by examining neural activity using BCIs. Also, estimation of TV advertisements memorization can be used to evaluate the advertisements.

Neurofeedback approach is used in education to examine the clearness of studied information. For self-regulation, BCIs are used in improving cognitive therapeutic approaches, such as learning to regulate anxiety and depression.

Security systems are generally based on knowledge, object, or biometrics. These systems tend to be vulnerable, for instance simple insecure password, shoulder surfing, theft crime, and cancelable biometrics are several drawbacks for vulnerable security systems. Biosignals or cognitive biometrics are a solution to this problem.

In entertainment and video game fields, players can use non-medical BCI devices to directly control a game element or walk around in a virtual environment. BCIs can detect the affective state of the player and change the game difficulty accordingly. In some serious games, players can learn to control their emotions.

1.2 Challenges

The challenges of BCIs are common in almost all fields they're used in. For instance, the training process is time consuming, application of gel to acquire more robust signals from scalp is uncomfortable, EEG signals are non-stationary and noisy, there are relatively less training sets to train a robust classification model. Some challenges for BCI based video games are elaborated below (Nijholt, 2008 ; Lotte, 2011).

There are many limitations for BCI video games to be commercially off-the-shelf. For instance, most of the BCI games are proof-of-concept. They are tested in laboratory conditions. EEG is very prone to noise, even eye blinking or muscle movement causes EEG data to be corrupted. In laboratory conditions, subjects' movements are often limited, but in real gaming environment players will often move or blink. Even in the laboratory conditions BCI performance (in terms of information transfer rate and latency) is already low, and in real gaming environment it is expected to get lower.

In order to get better accuracy there are two steps taken that gamers would probably like to avoid ; one is applying gel to head before EEG cap is worn, the other is calibrating the BCI system which takes about 5 to 20 minutes. But as the BCI systems are getting more advanced, even though with a little accuracy loss, these steps can be skipped.

For roughly 20% of BCI users, the signals cannot be interpreted with sufficient accuracy to form some kind of control. This means 20% of gamers will not be able to use BCI, therefore adoption of BCI might be a problem in the gaming community.

BCIs cannot grant rich game controls. In fact they are very limited, BCIs can recognize only a small number of mental states, such as distinguishing different motor imagery, or flickering objects with different frequencies.

Gamers would not like to wear wired and uncomfortable EEG equipment. But with recent advancements, light and wireless EEG caps are developed. In near future, these sensors can be integrated with some wearable devices such as headphones, microphones or glasses ; or clothing such as hats, caps or berets.

EEG signals are noisy and non-stationary with low spatial resolution, has low information transfer rate and high latency. These characteristics prevent BCIs to be the main control mechanism for complex video games. But they can be used as an additional control channel together with traditional controls such as mouse or keyboard.

This thesis is organized as follows. In Chapter 2, categorization of BCIs and BCI modalities are briefly explained, and related work on EEG based video games and dynamic difficulty adjustment is given. In Chapter 3, the hardware and software used in this study is introduced. Used algorithms are explained, and implemented games are introduced. In Chapter 4, conducted experiments in order to find best SSVEP stimulus color and frequency, and compare the letter layouts of two SSVEP based spellers are explained and the results are given. Then the tests for evaluating the SSVEP based game's control accuracy and implemented dynamic difficulty adjustment system's effectiveness are explained and results are given. In Chapter 5, a summary of this study is given, comparison of this study with other related studies is given, and finally what can be done to improve this study is explained.

2 LITERATURE REVIEW

2.1 Categorization of BCIs

There are many ways to categorize BCIs such as dependent or independent BCI; invasive or non-invasive BCI; synchronous or asynchronous (self-paced) BCI; and categorization with respect to brain activity. All of these categories are explained below (Abdulkader, 2015).

2.1.1 Dependent and Independent BCI

Dependent BCIs require a motor control while independent BCIs do not require any motor control, therefore independent BCIs are beneficial for severely disabled patients.

2.1.2 Invasive and Non-invasive BCI

Invasive BCIs require microelectrodes implanted in brain. While the acquired signal has a high quality, this type is generally not preferred, especially by healthy subjects. In non-invasive BCIs, the signals are acquired using a device like a cap. Even though the acquired signals are low in quality, this type of signal acquisition is more preferable.

2.1.3 Synchronous and Asynchronous (Self-Paced) BCI

In synchronous BCIs, the user is sometimes forced to interact with the system in order to acquire brain signals. In asynchronous BCIs, the user interacts with the BCI whenever he/she wants and the system will react to his/her mental activities.

2.1.4 BCIs in terms of Brain Activity

BCIs can be categorized as active, reactive and passive BCIs.

In active BCI, the user's intention is to interact with the BCI, therefore the user voluntarily generates brain signals. Motor Imagery falls under this category.

In reactive BCI, like active BCI, the user's intention is to interact with the BCI, but unlike active BCI, the user exposes himself/herself to an external stimulus to generate the brain signals. SSVEP can be an example for reactive BCI.

In passive BCI, the user doesn't have any intention to interact with the BCI, instead the BCI watches the user and reacts accordingly. As an example, BCI system can monitor the attention/concentration level of the user.

2.2 Brain waves categorized by their frequency bands

Brain waves can be categorized into 5 frequency bands. Delta bands (0.5 - 4 Hz) occur when one is in deep sleep state. Theta bands (4 - 8 Hz) are related to meditational state, or, when one's body is asleep but mind is awake. Alpha bands (8 - 13 Hz) occur when one is relaxed or when their eyes are closed. Beta bands (13 - 30 Hz) can be observed when one is paying attention to something. Finally, gamma bands ($>$ 30 Hz) occur when one is making a decision. Brain waves can be used to detect learning disabilities and difficulties maintaining conscious awareness (Siuly, 2016).

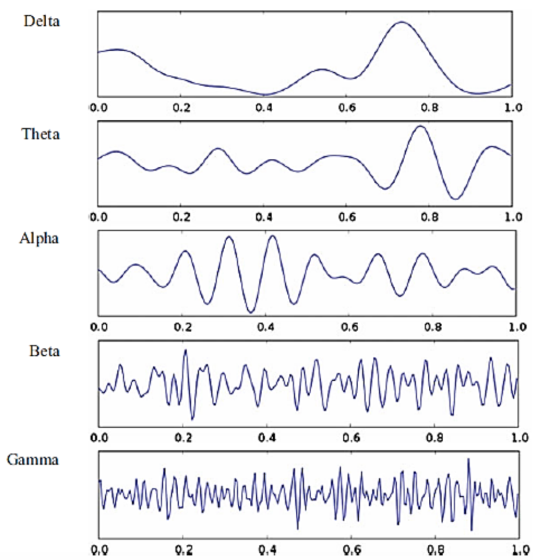


Figure 2.1: Brain waves visualization with different frequency bands (Siuly, 2016).

2.3 Control Signal Modalities

In order to control a computer using brain signals, the user requires to be stimulated by some internal or external stimuli, or be in a specific state of mind such as concentrated or relaxed. The most popular modalities used in BCI systems are categorized and explained as follows(Abdulkader, 2015).

— Evoked signals : P300, Steady State Visual Evoked Potentials

— Spontaneous signals : Sensorimotor rhythms, Non-motor cognitive tasks

— Hybrid signals

2.3.1 Evoked Signals

Evoked signals are generated by brain when the user is exposed to some specific external stimuli. The generated signals may show differences depending on appearance and behavior of the stimulus. The most commonly used evoked signals are P300 and SSVEP.

P300 wave occurs when the user is exposed to an infrequent external stimulus, thus surprising the user. Therefore, the signal is generally generated using an oddball paradigm. The user is exposed to a sequence of ordinary stimuli, and randomly and infrequently extraordinary stimuli.

The advantages of P300 over other signal modalities are, it requires no training and has relatively high information transfer rate (20-25 bits/min). The disadvantage is, it requires user to pay attention to the external stimulus permanently, therefore it is tiring for the user. P300 wave is suitable for selection tasks rather than continuous control. Congedo et al. (2011), Maby et al. (2012) and Finke et al. (2009) used P300 waves to select a target game object.

SSVEP is elicited when the user is exposed to a regularly flickering object. When the user focuses on the flickering object, SSVEP occurs in the visual cortex with the same frequency as the frequency of stimulus. The frequency of stimulus is generally chosen between 6-30 Hz. The advantages of SSVEP are, it requires very short training (around 2 minutes), has very high information transfer rate (60-100 bits/min), and has high signal to noise ratio (SNR). The disadvantages are, since only a limited number of frequencies are useful, number of controls are also limited, and as P300, the user's permanent attention to the stimulus is required. Like P300, SSVEP is also suitable for selection tasks. Vliet et al. (2012) used only one flickering object to make a selection between several choices. Choices are highlighted sequentially, when the user concentrates on the flickering object, the highlighted option is chosen. Gurkok et al. (2013) and Lalor et al. (2005) used flickering objects with different frequencies for each option. Aside from visual stimuli, steady state evoked potentials can be generated with tactile or audial stimuli.

SSVEP has important advantages over other BCI modalities. It has high signal to noise ratio, high information transfer rate, and requires little training.

2.3.2 Spontaneous Signals

Spontaneous signals are generated by the user voluntarily. They don't require any external stimuli. The most commonly used spontaneous signals are sensorimotor rhythms and non-motor cognitive tasks.

Sensorimotor rhythms are related to motor actions such as moving an arm or leg. By imagining movement of a limb, user's motor imagery states can be controlled. The intention of limb movement can be detected from the μ band in the motor cortex of the brain. The ITR of sensorimotor rhythms can vary from 3 to 35 bits/min. The advantage of this kind of signals is, they don't require an external stimulus, user generates them voluntarily, but the disadvantage is that it requires a long training. The sensorimotor rhythms are useful for continuous control, such as navigation. Holz et al. (2013) and Pour et al. (2008) used motor imagery to move a game object to left or right. Tangermann et al. (2009) used motor imagery to choose between the objects on the left or on the right.

Non-motor cognitive tasks, such as mathematical computation, music imagining, visual counting, etc. can be used to control a BCI.

2.3.3 Hybrid Signals

Hybrid signals can be composed of multiple brain signals, other physiological signals such as EMG, or other sensors such as eye trackers. Hybrid signals can be used to gain a more complex interaction, enhance the reliability of the BCI, or to avoid disadvantages of component signals.

2.4 BCI as a Control Mechanism for Video Games

The majority of the BCI applications for healthy users consist of video games. There are many reasons why video games make up this majority, explained by Allison et al. (2007) and Nijholt (2008). First of all, most of the gamers easily adopt new technology. Some of the gamers already wear headgear. Second, they know that in order to master something, a game or a control mechanism, gamers have to train a lot. Gamers are very competitive, they will work hard for the advantage provided by BCI. Third, some of the gamers do not hesitate to spend money on gaming peripherals. And finally, gaming companies can earn huge profits by introducing a new game element such as BCI, which is a powerful motivation to invest in research and development of BCIs.

There are many ways to integrate BCIs into video games, as explained by Nijholt et al. (2009). While playing a game, the player can feel bored, relaxed, worried, anxious or aroused depending on the game's difficulty level and player's skills. BCIs can be used to monitor and detect the mental states of the player and arrange the difficulty level of the game accordingly.

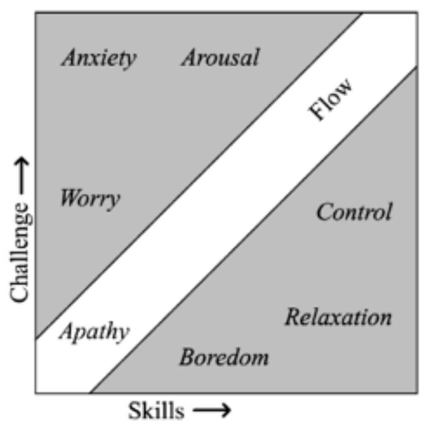


Figure 2.2: Mental states of a player according to game difficulty and player skill level (Nijholt, 2009).

BCIs can be used to force the player to make a selection by exposing the player to visual or auditory stimuli, and this can be achieved in a natural way in the game environment.

The player can consciously evoke particular brain signals by imagining limb movement or making mathematical calculations in order to control a game element.

The player can consciously control his or her state of mind such as being relaxed or concentrated to control a game element.

BCIs can be used for auxiliary control in games. For example, BCIs can be used to give optional commands instead of critical commands. These optional commands can make the gameplay easier and more enjoyable, but they are not a necessity to complete the game.

BCIs can also become handy as an extra control modality, especially in "induced" or "situational disability" moments, which means the moments when one wants have more skills and communication channels than traditional control mechanisms such as mouse or keyboard, to control the game environment. For example, instead of forcing player to hold multiple keys, BCI can be used for specific commands.

2.5 BCI as Main Control in Videogames

There are many proposed videogames from researchers that employ BCI as the main control mechanism. Examples from this kind of video games are given below.

Holz et al. (2013) conducted a study that aims to evaluate the usability of SMR-BCI controlled gaming prototype. There were 4 subjects, A B C and D, who had motor impairments. 2 of the subjects, C and D, were not able to speak and had no reliable eye movement. The other two were less impaired. They were able to use a keyboard to communicate and use a joystick to control a wheelchair. B couldn't speak and A used a voice amplifier to speak. The game, Connect-Four, is a two-player game. In the game, coins are placed in rows and columns, the objective is to connect four coins before the opponent. The players can select a row by imagining moving their right hand to move the cursor to right, or left hand to move the cursor to left. To place the coin, they can imagine moving their feet. The classifications of left hand, right hand or feet movement, which is used to controlling the video game were done using LDA. The accuracy varied between four subjects. A had 65%, B had 60%, C had around 40 - 50 % and D had 77% accuracy. The ITR also varied between subjects. For A, ITR (bits/min) was between 0.17-1.15 , 0.06-0.56 for B, 0.11-0.32 for C and 0.05-1.44 for D. The maximum ITR that 100% accuracy could be obtained was 4 bits per min. An interview was made after playing the game. The most suggested improvement was making system set-up (including the headset, training for the game etc.) easier.

Liarokapis et al. (2014) compared two EEG based BCI devices through a serious game that aims to teach history to children between age 11-14. The game illustrates the urban development of Rome from 1000 B.C. to 550 A.D. The player can control the navigation of the avatar with EEG signals. There are two headsets used to capture the EEG signals. One is NeuroSky Mindset device. NeuroSky Mindset uses one dry sensor. It doesn't require any calibration. It can extract the levels of attention and meditation of the user. Attention level is measured with the user's mental focus, while meditation level is measured with user's relaxed or stressed state of mind. The other headset, Emotiv, uses 14 wet sensors, and is capable of obtaining EEG signals as well as facial expressions. It requires training for classifiers. Both headsets are capable of fully controlling the navigation of the avatar. Players use their cognitive functions such as changing their attention and meditation levels to control the avatar. If they're using Emotiv, they also use their facial expressions. For example the players must blink their eyes to make the avatar turn left if they're using Emotiv, and they have to defocus their attention if they're using NeuroSky. 62 tests were performed. 31 subjects played the game using the Emotiv headset, and the NeuroSky headset separately. Then the subjects were asked to complete a questionnaire. Generally, playing the game with both headsets was found very interesting and fun. But there were also dissatisfactions. For example while making a turn, sometimes the avatar didn't do what the player wanted. Also sometimes 3 - 5 seconds lags occurred. It has also been stated that at first, adapting to the headsets and learning to use their cognitive functions were challenging. Moreover some participants stated that they began to feel tired after 30 minutes of playing.

Finke et al. (2009) developed a game which consists of a checkerboard styled game board with 18x28 fields, 12 trees and a character. The trees are randomly placed on the board. The player must move the character from tree to tree until every tree is visited. One turn of the game is as follows : All trees are highlighted consecutively. There are 120 milliseconds between each highlight. The target tree is highlighted in red, while the non-target trees are highlighted in yellow. The player focuses on the target tree, and elicits P300. When the turn ends, the classifier tries to identify the target. If a successful identification is made, the character moves to the target tree and another turn starts. The classification is based on supervised learning. There are 500 positive epochs (target stimulus) and 500 negative epochs (non-target stimulus) in training set. FLDA was used for classification. PCA was applied to reduce to dimension from 2050 to 137 - 146 with 99.9% variance. For online classification, the PCA matrix was precomputed and stored. The game was evaluated with 9 male and 2 female subjects of age between 24 and 39. 10 electrodes were applied to the scalp. Each subject completed training session and four game session consecutively. In offline classification evaluation 84.9% accuracy is obtained. In online gaming, 65.9% accuracy was obtained in single trial classification.

Lalor et al. (2005) developed the MindBalance game, in which, the character is on a rope and the player tries to balance the character using EEG signals. There are two checkerboard images and the character is in between these images. As the character starts to lean on one side, the player can focus on the image in the opposite direction to balance the character again. Before the game was developed, some preliminary analyses were performed. It was found that black and white checkerboard pattern produces more SSVEP signals compared to a yellow circle flickering on a black background. 17 Hz and 20 Hz are identified as the two bilateral checkerboard frequencies that elicit the most SSVEP. Two feature extraction methods were tested. The data were divided into segments of 1 and 2 seconds. The first method is to raise these segments to 4 seconds using zero-padding, take the fast Fourier transform and square it. The second method is to calculate the autocorrelation function for each segment and then take the FFT. It was found that the second method is more resilient to noise. For the classification, LDA was used. The accuracy was around 70%. During the online gameplay, one machine is used for data acquisition and signal processing, and another computer was used for 3D graphics rendering. The off-balance animation of the character lasts 3 seconds, so that the player can focus on the checkerboard on the other side, and elicit the SSVEP signals. These signals are then used for decision making. There were 48 games played, 8 trials per game. The peak accuracy was 89.5%.

Pour et al. (2008) developed a wireless BCI game for mobile phones. In the game named BreakOut, subjects control a bar to bounce a ball. To move the bar to left or right, subjects were asked to imagine moving their left/right hand. The recorded data is sent to a PC with Bluetooth. After feature extraction and translation, the commands to move the bar to left or right is sent to mobile device using WI-FI. The game was evaluated with four male subjects, that were between 20-40 years old. The average accuracy is 63.9%. The highest accuracy is 100%.

Vliet et al. (2012) compared two different headsets, one is a research lab with expensive equipment, the other one is a consumer grade headset (Emotiv EPOC). The game is a tower defense game, the goal is to protect the tower from waves of enemies by building different structured and placing them on the map. When an enemy reaches the tower, the game is over. When the game starts, the user selects the places to build their defense structures. They can also undo an action. After they are done with building their structures, they can start the game. The user loses control until all enemies are dead, or an enemy reaches to tower. All controls are done using BCI. A visual stimulus is used flickering at a fixed frequency. The selections are highlighted one by one, and when a user wants to choose a selection, when it is highlighted they must look at the stimulus. Even though the accuracy obtained when using the research headset is higher compared to the accuracy obtained when EPOC headset is used, EPOC is also sufficient enough for the game.

Ren et al. (2020) investigated the effect of Functional Electrical Stimulation (FES) on the lower limbs. They've developed a BCI that uses VR for visual guidance to the subjects, and FES to stimulate the subjects' lower limbs before the imagination of the limbs, so they experience the muscles' contraction and gather their attention on the stimulated limb. In the experiment, the subjects are shown the ball kicking movement in the VR, and asked to imagine the same movement when the see the visual cue. They used the sliding window method to increment the number of samples, the sub-frequency band division and common spatial pattern methods for feature extraction, and SVM for classification. The 12 subjects were divided into two groups. One group completed the experiment without FES, and the other completed it with FES. Then the group that didn't use FES repeated the experiment with FES, and the other group completed the experiment without it. In both of the groups, the subjects performed better using FES. The average accuracy of all subjects is %84.83 with FES, and %78.10 without FES.

Liu et al.(2018) used an SSVEP-BCI system to control a toy car. They used four flickering LEDs with different frequencies (18, 20, 22 and 24 Hz) for four different directions. For estimation the spectrum of EEG signals, multiple signal classification (MUSIC) method was used. Then, for each frequency, they've generated reference sine signals and calculated the correlation coefficient of EEG signal and its corresponding reference signal for each frequency. These coefficients formed the feature set which is used in Support Vector Machine classifier. The accuracy has reached %94 in the 3 second timescale.

The summary of the literature on BCI based video games can be found in Table 2.1.

Table 2.1: Publications on BCI based video games

Year	Authors	Modality	Accuracy	Classifier
2020	Ren et al.	SMR	78.10% - 84.83%	SVM
2018	Liu et al.	SSVEP	94%	SVM
2014	Liarokapis et al.	Attention level	Questionnaire	Emotiv CP
2013	Holz et al.	SMR	47% - 77%	LDA
2012	van Vliet et al.	SSVEP	83.7% - 88.2%	SLIC
2009	Finke et al.	P300	65.9% - 84.9%	LDA
2008	Pour et al.	SMR	63.90%	BCI2000
2005	Lalor et al.	SSVEP	76.7 - 89.5%	LDA

2.6 Arranging Game Difficulty

There are many metrics for measuring the player experience. Some of them are fun, engagement, immersion, presence, involvement, incorporation, enjoyment and flow (Ang, 2017). Players feel frustrated and anxious when the game is too difficult, or they feel bored when the game is too easy for them. Csikszentmihalyi (1997) stated that players go through a flow channel when they don't feel boredom or frustration illustrated in Figure 2.2.

Even though a game has different difficulty settings, the player may not know how to match their skills to the difficulty level, or if the game is a multiplayer game it is important to match players with similar skills. Therefore a system that dynamically changes the game difficulty according to the player's skill is needed.

Zohaib (2018) defines dynamic difficulty adjustment as "a technique of automatic real-time adjustment of scenarios, parameters, and behaviors in video games, which follows the player's skill and keeps them from boredom or frustration". In many studies it was shown that games with DDA are more likely to keep players in the flow compared to the games that don't involve DDA (Ang, 2017 ; Zohaib, 2018 ; Stein et al., 2018).

There are many ways to apply DDA to a game. One of the most simple way is based on heuristics. According to the game score of the player such as life point, kill count, passed time to complete a task etc. the difficulty can be increased or decreased. Probabilistic based and AI based DDAs are also found to be increasing the engagement in the games.

DDA systems using affective states of the players are also very popular. Biosignals such as ECG, GSR or EEG can be used to recognize the affective state of the player. There are two ways to measure a person's affective state. One is categorizing the affective state such as happy, sad, angry, or relaxed.The other way is to determine the affective state in two main dimensions ; the unpleasant-pleasant, also called valence and activation-deactivation, also called arousal. When players have low valence and high arousal they may be frustrated, and when they have low valence and low arousal they may be bored.

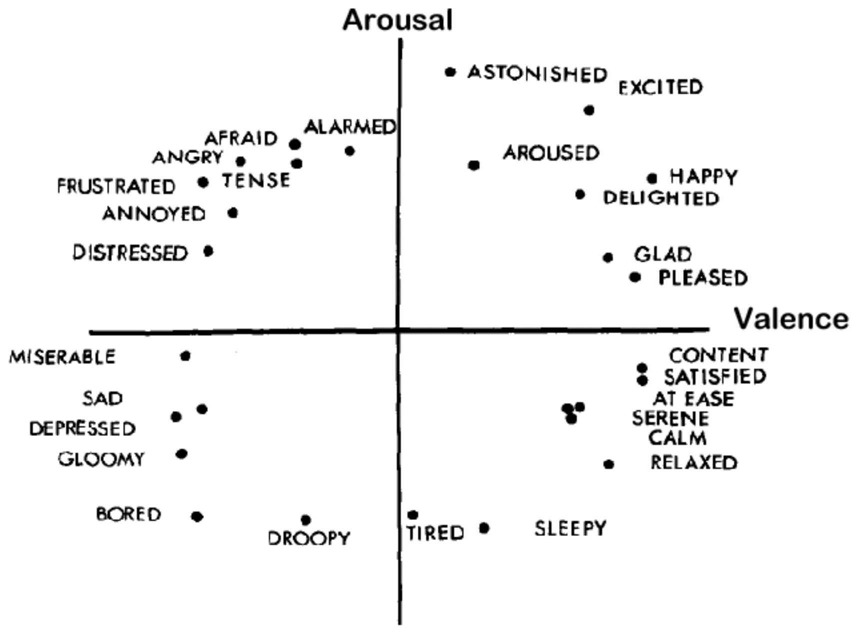


Figure 2.3: Affective states (Russell 1980).

Szegletes et al. (2014) implemented a biofeedback framework for DDA in serious games. Serious games are a specific group of games that intend to teach something to the players while they're playing the game. They've used ECG, EEG and eye tracking sensors to categorize the affective state of the players as fatigued, bored, engaged or frustrated. They've combined the result of the affective state categorization with game score to determine if the game difficulty should be changed.

Giakoumis et al. (2011) designed a biosignal based system that automatically detects boredom in video games using ECG and GSR signals. They've induced boredom naturally by making players play an easy, repetitive labyrinth game. They've compared conventional biosignal features such as mean and standard deviation of heart rate variability and interbeat interval, mean level of galvanic skin response with moment based biosignal features, i.e variations of Legendre moments and Krawtchouk moments. They've monitored these features for each subject during the experiment and examined the values of these signals when the state of boredom is naturally induced. The subjects filled a questionnaire after each trial about their affective state. They've found that using only conventional features, they've recognized boredom state with %85.19 accuracy. Using only moment based features, the accuracy was %83.07. When both conventional and moment based features are used together, the accuracy increased to %92.25.

Park et al. (2014) used EEG signals to recognize players' emotion to keep the player engaged in the game. They analyzed the asymmetry among left and right brain hemispheres to recognize players' emotion. They designed a game with three difficulty settings. When the user feels bored, the game difficulty is increased, and when the user feels anxious the game difficulty is decreased. They concluded that EEG signals can be used for dynamic game difficulty control system and can be a useful tool for game design.

Stein et al. (2018) used Emotiv Affective Suit for their dynamic difficulty adjustment system, to maximize the short term excitement in players. To test their system, they used a multiplayer, open source, third person shooter game called Boot Camp. They've designed four game modes, two for stronger players to make the game more difficult, and two for weaker players to make the game easier. They've also designed a heuristic DDA system based on kill count to compare to the EEG based DDA. 24 subjects completed the game with EEG based DDA, heuristic based DDA, and without any DDA. Subjects reported that the game with EEG-based DDA was the most interesting and challenging one compared to others.

Mikami et al. (2017) developed a level design system based on player's attention and game performance. They've used a 2D platform game similar to Super Mario Bros. The attention level of player is gathered from attention value of eSense software using EEG data. The player performance is based on number of deaths or hits by an enemy. As the player gets more experienced with the game, the difficulty of the game is increased by increasing the number of enemies and other game elements.

Reuderink et al. (2009) investigated how frustration affects the EEG signals during gameplay. The game is very similar to Pacman. In this study BCI is not a way of controlling the character, the player controls the character using keyboard. However

EEG signals are collected by BioSemi ActiveTwo EEG system to analyze the effects of frustration on the signals. To frustrate the user, both user input and visual output is manipulated. 5% of the key presses are missed, and the screen freezes for two to five frames at 25 frames per second with a probability of 5%. One session consists of three blocks. Each block is a two minute of gameplay. Two of these blocks are not manipulated but one of them causes frustration. After each block, the user state is evaluated by Self Assessment Manikin, in terms of valence, arousal and dominance. The ERD data for normal and frustration conditions are compared, and it is found that frustration has a visible affect on EEG signals, therefore it could deteriorate BCI performance.

The summary of the literature on EEG based DDA systems can be found in Table 2.2.

Table 2.2: Publications on EEG based DDA

Year	Authors	DDA input
2018	Stein et al.	Short term excitement level
2017	Mikami et al.	Attention level
2014	Park et al.	Valence and arousal levels
2009	Liu et al.	Anxiety level

2.7 Speller

BCI is a great communication channel for patients with motor and speech impairment. These patients can communicate with the environment and people by typing sentences using a BCI based speller, expressing themselves. Some of developed BCI based spellers are given below.

Cecotti (2010) developed an SSVEP based speller that doesn't require any calibration. In the speller, one can type 27 characters, the 26 Latin characters [A...Z] and "_" for separating the words, delete the last character and undo the last selected command. Three of five visual stimuli are used for selecting characters to type, while the remaining two stimuli are used for delete and undo commands. Characters are recursively divided into three groups until only one character is left to be selected. A linear combination of the signals measured from different electrodes is calculated using PCA to reduce the noise and enhance the information in the signal. To improve the robustness of the selection, four different frequencies are also considered in the decision-making process. To generate a command, the frequency power of each class, nine in total, is normalized into a probability by a softmax function and the frequency with the highest probability is selected. 11 subjects completed the experiment and it was found that the average accuracy is 92.25%, the information transfer rate is 37.62 bits per minute, and the average speed is 5.51 letters per minute.

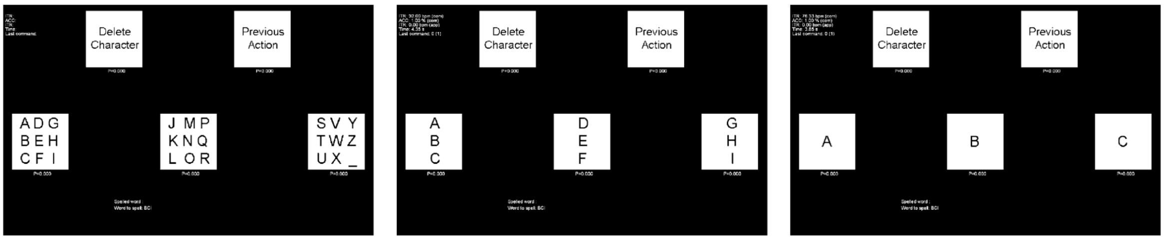


Figure 2.4: Three stages of the speller (Cecotti 2010).

A different SSVEP based speller was implemented by Valbuena et al. (2008) in which 32 characters including clear and delete commands were presented in the Rhombus layout. Friman et al. (2007) stated that the Rhombus layout was constructed by taking the occurrence frequency of the letters in the English language into consideration. It was designed so that commonly used letters were closer to the center. Also, it was assumed that paying attention to the same stimulus for longer was more comfortable than switching the attention between different stimuli, therefore commonly used letters were placed so that the user could select them by looking at the same stimulus. Also in Rhombus Layout, it is easier to correct a mistake. For example one can easily change the cursor's direction if it moves in the wrong direction, therefore correcting the mistake, even before typing a letter. Five visual stimuli were used ; four of them for navigating the cursor left, right, up or down ; and the fifth one is for selecting the character which the cursor is located on. After a character is selected, the cursor moves to the center of the speller. To test this speller, 106 subjects aged between 18 and 79 were asked to type 5 phrases. It was stated that 35% of the subjects couldn't even spell the shortest phrase which is "BCI". The accuracy was calculated only for the subjects who spelled the desired phrase correctly or with some errors and the average accuracy was calculated as 92%.

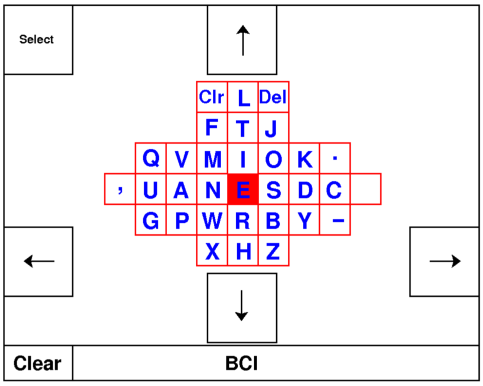


Figure 2.5: Speller in rhombus layout (Valbuena et al. 2008).

3 MATERIALS AND METHODS

3.1 Emotiv EPOC

Emotiv EPOC is a headset that captures EEG signals from scalp. It is noninvasive and it doesn't require any gel to be applied, instead it uses wet sensors. It has 14 channels (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4) and two references, Common Mode Sense (CMS) / Driven Right Leg (DRL) at P3/P4 according to 10-10 electrode placement system explained below. It is wireless, it connects to the computer via Bluetooth using a USB dongle. Emotiv EPOC is one of the most popular headsets used in BCI research.

Figure 3.1: Emotiv EPOC headset.

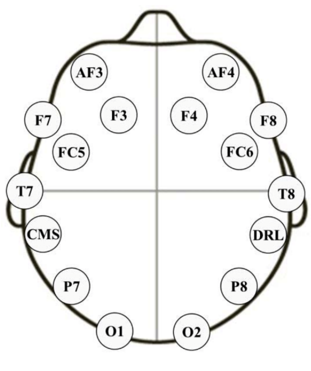


Figure 3.2: Emotiv EPOC electrode locations.

3.1.1 Electrode Placement System : 10 - 20 System

10 - 20 system is a method for describing the locations of sensors. Each sensor location is composed of at least one letter and one number. Letters describe the brain lobe,

while numbers describe which hemisphere of brain the sensor is located. If the numbers are even, then the sensors must be placed on the right hemisphere, if the numbers are odd, the sensors must be placed on the left hemisphere. Z (zero) means the electrodes must be placed on the midline.

Table 3.1: Letters of electrode placements and corresponding brain lobes in 10 - 20 system.

Letter	Location
Fp	Pre-frontal Lobe
F	Frontal Lobe
T	Temporal Lobe
C	Central
P	Parietal Lobe
O	Occipital Lobe
A	Ear

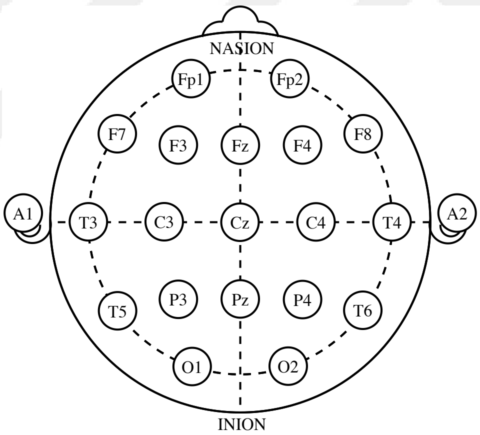


Figure 3.3: The 10 - 20 system electrode positioning.

3.1.2 Electrode Placement System : 10 - 10 System

If more electrodes are used for a signal with higher resolution, the extra electrodes are placed in middle of the electrodes in 10 - 20 system. This new system is called 10 - 10 system.

In 10 - 10 system, electrodes' locations are also shown with letters and numbers. Like in 10 - 20 system, even numbers correspond to right hemisphere and odd numbers correspond to left hemisphere. T3, T4, T5 and T6 electrode positions' names in 10 - 20 system are changed to T7, T8, P7 and P8 respectively. The extra electrodes have new letter codes as shown below.

Table 3.2: Letter codes of extra electrodes in 10 - 10 system.

Letter	**Location**
AF	between Fp and F
FC	between F and C
FT	between F and T
CP	between C and P
TP	between T and P
PO	between P and O

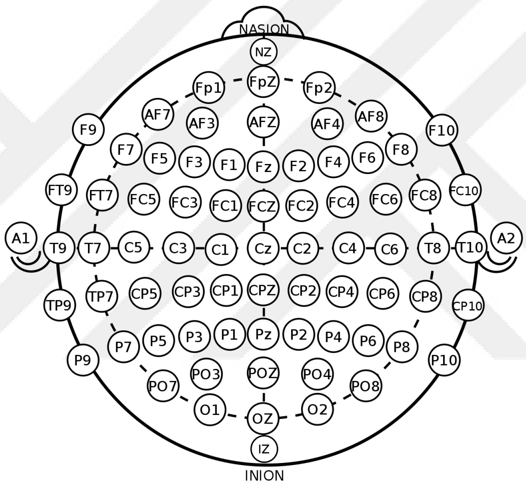


Figure 3.4: The 10 - 10 system electrode positioning

3.1.3 Emokit Driver

To control a video game with brain signals, we need to collect raw data from the Emotiv EPOC headset, process and map them to some control mechanism in the game. Even though Emotiv provides a number of software to use the headset, it does not give us the raw EEG data. Therefore a driver named Emokit[1] is used to obtain the raw EEG data. Emokit is an open source driver, written by Cody Brocious and Kyle Machulis and is compatible with Emotiv EPOC and EPOC+. It works on Windows, OSX and Linux.

1. https ://github.com/openyou/emokit

3.2 Unity Game Engine

Unity is one of the most popular game engines that is mainly used for developing video games in 2D and 3D. Its scripting languages are C#, Boo (a Python-like language) and Unity Script (a JavaScript-like language), but around 80% of the scripts are written in C#. Unity is also used in film making, automotive and transportation, gambling, virtual and augmented reality, and architecture engineering and construction. Unity supports building games for 27 different platforms, including Windows, Mac, Linux, iOS, Android, Playstation, Xbox, Wii, Oculus Rift, and Google Cardboard.

3.3 Methods

3.3.1 Preprocessing Methods

Data preprocessing methods are applied to raw EEG data to remove noise and to prepare data for feature extraction. EEG equipment and its connections to the scalp, normal electrical activity of the subject as heart, eye blinking, eyes movement and muscles in general cause noise. If this noise is reduced, better results can be obtained. The preprocessing methods used in our experiments are explained in the following sections.

3.3.1.1 Mean Normalization

Some feature extraction methods such as higher order crossings require the data to have zero mean. Therefore, before feature extraction, data can be normalized by subtracting the mean from it. The mathematical formula of used normalization function is below.

$$x' = \frac{x - average(x)}{max(x) - min(x)} \tag{3.1}$$

where x is the original signal and x' is the normalized signal.

3.3.1.2 Filtering

Filtering is applied when a set of frequencies below, above or between certain thresholds is wanted to be removed. The reason for this may be noise removal, for example, in order to reduce the effect of the voltage from the galvanic skin response across the head, a filter can be applied to attenuate frequencies below 5 Hz. There are mainly four types of filters

1. Low-pass filter : all frequencies below a defined frequency are passed and all frequencies above this limit are rejected.
2. Band-pass filter : all frequencies between defined lower and upper frequency limits are passed.
3. High-pass filter : all frequencies above a defined frequency limit are passed and all below are rejected.
4. Band-stop (Notch) filter : all frequencies between a defined lower and upper frequency limit are rejected.

Figure 3.5 presents these filter types.

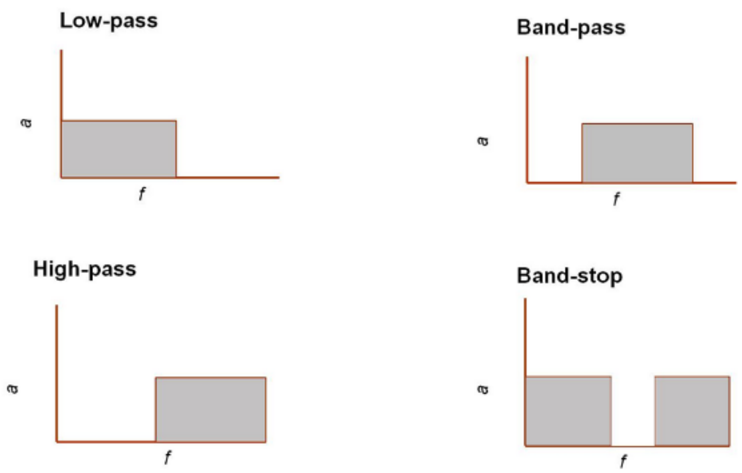


Figure 3.5: Basic filter types.

3.3.2 Feature Extraction Methods

Some information can be hidden in the signal, and revealing these information can greatly increase the classification accuracy. To recognize the affective state of the player, higher order crossing features are used.

3.3.2.1 Periodogram

Periodogram is used to estimate the spectral density of a signal. It is calculated as follows.

$$PSD(f) = \frac{|FFT(W_t \times X_t, f)|^2}{L} \times \frac{1}{\Delta f} \tag{3.2}$$

where X_t is the signal, L is the number of samples, W_t is the applied window function and Δf is the frequency interval.

3.3.2.2 Discrete Wavelet Transform

Wavelet transform represents the signal in both the time and frequency domain. Discrete wavelet transform is a wavelet transform where the wavelets are discretely sampled. The degree of correlation between the signal and the wavelet function of it at different instances of time is represented by coefficients of DWT (Yohanes 2012). The DWT coefficients can be calculated as follows :

$$C_{x(t)}(l, n) = \int_{-\infty}^{\infty} x(t)\Psi_{l,n} dt \tag{3.3}$$

$$\Psi_{l,n}(t) = 2^{-(l+1)}\Psi(2^{-(l+1)}(t - 2 - ln)) \tag{3.4}$$

where $x(t)$ is the signal, l and n are the scale and translation variables of the wavelet function, respectively.

3.3.2.3 Entropy

Entropy is a measure of disorder in a data set. For a given data set, one can calculate the entropy as follows.

$$S = -\sum_{k=1}^{n} p_k \times \log_2(p_k) \tag{3.5}$$

where p_k is the probability of observation k to appear in the data set, and n is the number of distinct observations.

3.3.2.4 Zero and Mean Crossings

Physiological signals such as EMG, ECG or EEG oscillates differently under different conditions. By examining the characteristics of the oscillations, one can find useful information. Therefore the characteristics of the oscillations can be extracted as features for classification purposes.

The oscillation around 0 can be expressed as zero crossings and the oscillation around mean can be expressed as mean crossings.

First, the backwards difference operation must be applied to the signal Z.

$$\nabla Z(k) \equiv Z(k) - Z(k-1) \tag{3.6}$$

To find the number of zero crossings, the time series X is constructed as

$$X(k) = \begin{cases} 1 & \text{if } \nabla Z(k) \geq 0, \\ 0 & \text{if } \nabla Z(k) < 0 \end{cases} \tag{3.7}$$

Then the feature can be extracted by counting the sign changes in X

$$NZC = \sum_{k=2}^{N} [X(k) - X(k-1))]^2 \tag{3.8}$$

To find the number of mean crossings, the same procedures can be applied to the mean normalized signal.

3.3.2.5 Statistical Features For Biosignals

According to Healey and Picard (1998), statistical properties such as mean or standard deviation of biosignals can extract valuable information for classification. In our experiment, we extracted the following features.

The mean of the signal is calculated as follows :

$$\mu_X = \frac{1}{N}\sum_{n=1}^{N} X(n) \tag{3.9}$$

the variance of the signal is calculated as follows :

$$\sigma_X^2 = \frac{1}{N}\sum_{n=1}^{N}(X(n) - \mu_X) \tag{3.10}$$

the standard deviation of the signal is calculated as follows :

$$\sigma_X = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(X(n) - \mu_X)} \tag{3.11}$$

the k^th percentile of the signal is n^{th} value of the sorted signal where n is calculated as follows :

$$n = \frac{k}{100}(N + 1) \tag{3.12}$$

the root mean square of the signal is calculated as follows :

$$RMS = \sqrt{\frac{\sum_{n=1}^{N} X(n)^2}{N}} \tag{3.13}$$

where N is the length of the signal, and $X(n)$ is the n^{th} observation.

3.3.3 Classification Methods

3.3.3.1 Canonical Correlation Analysis

CCA can be used to determine the target SSVEP stimulus by using multi-channel EEG data. CCA works as follows : Let X and Y be two multidimensional variable, and $x = X^T W_x$ and $y = Y^T W_y$ be the linear combinations of these two variables. CCA finds the weight vectors W_x and W_y which maximize the correlation between x and y by solving the equation 3.14.

$$\max_{W_x,W_y} \rho(x,y) = \frac{E[W_x^T XY^T W_y]}{\sqrt{E[W_x^T XX^T W_x]E[W_y^T YY^T W_y]}} \quad (3.14)$$

In equation 3.14 $\max_{W_x,W_y} \rho(x,y)$ yields the maximum correlation coefficient. In SS-VEP based BCIs, CCA is used to find the target stimulus in the following manner : For each possible target stimulus, a reference signal is generated using equation 3.15.

$$Y_f = \begin{pmatrix} sin(2\pi ft) \\ cos(2\pi ft) \\ \vdots \\ sin(2\pi N_h ft) \\ cos(2\pi N_h ft)) \end{pmatrix} \quad (3.15)$$

In equation 3.15 N_h is the number of harmonics, in our experiment N_h is equal to 3. As shown in Figure 3.6, the correlation coefficient between each Y_f and EEG data was calculated using CCA and the stimulus that yields the maximum correlation coefficient was selected as the target stimulus.

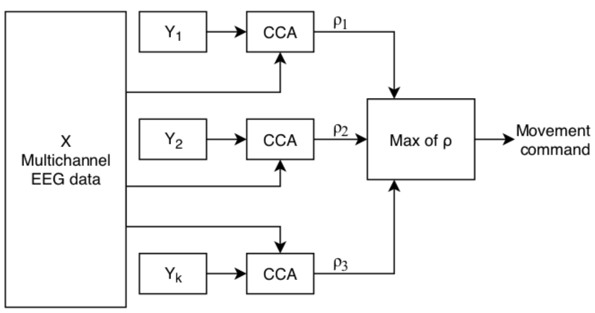


Figure 3.6: CCA usage in determination of the target stimulus.

3.3.3.2 Multilayer Perceptron

MLP is a supervised classification algorithm and is a class of feedforward artificial neural network. It consists of at least three layers of nodes ; input layer, hidden layer and output layer. Number of hidden layers can vary. All nodes in hidden and output layers are neurons with nonlinear activation functions. Each node in a layer connects to every node of the following layer with a weight. These weights are changed using backpropagation according to the amount of error, which is calculated using the output and expected result.

The degree of error of the n^{th} data point in the output node j can be represented by

$$e_j(n) = d_j(n) - y_j(n) \tag{3.16}$$

where $d_j(n)$ is the actual class and $y_j(n)$ is the predicted class. Then the weights are adjusted so that the errors are minimized. Total error is calculated using the following equation

$$\mathcal{E}(n) = \frac{1}{2}\sum_j e_j^2(n) \tag{3.17}$$

The change in each node is calculated using gradient descent algorithm

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n) \tag{3.18}$$

where y_i is the output of the previous neuron and η is the learning rate.

In this study, the implementation of MLP in Scikit-learn library version 0.24.1 is used (Pedregosa et al., 2015). The hidden layer size parameter is set to 500, rest of the parameters are set to their default values.

3.3.3.3 Gradient Boosting

Gradient boosting is a supervised classification algorithm that makes use of a loss function to generate an approximation $\hat{F}(x)$ to a function $F(x)$ in the form of weighted sum of functions $h_i(x)$ which are called base or weak learners.

This algorithm works in several steps. In any step m, an imperfect model F_m is improved by adding a new weak learner $h_m(x)$. Thus, in the next step F_{m+1} becomes

$$F_{m+1}(x) = F_m(x) + h_m(x) = y \tag{3.19}$$

Weak learners try to correct the errors from the previous step, therefore they are generated using the negative gradients of the mean squared error loss function with respect to $F(x)$.

$$L_{\text{MSE}} = \frac{1}{2}\left(y - F(x)\right)^2 \tag{3.20}$$

$$h_m(x) = -\frac{\partial L_{\text{MSE}}}{\partial F} = y - F(x) \tag{3.21}$$

In this study, the implementation of gradient boosting in Scikit-learn library version 0.24.1 is used (Pedregosa et al., 2015). Estimator number parameter is set to 500, rest of the parameters are set to their default values.

3.4 Implemented Games

Two games are implemented in Unity game engine. One of them is an SSVEP based Connect4 game, which uses BCI as the main control. The second game is a game similar to Guitar Hero. The main control in this game is keyboard, though BCI is used to dynamically change the game difficulty according to the affective state of the player.

3.4.1 Connect 4

Connect 4 is a two player, turn based game. There is a vertical board with 6 rows and 7 columns. Each player has 21 coin-like pieces, and each of their coins are in different color. The goal of the players is to make a horizontal, vertical or diagonal line using their coins.

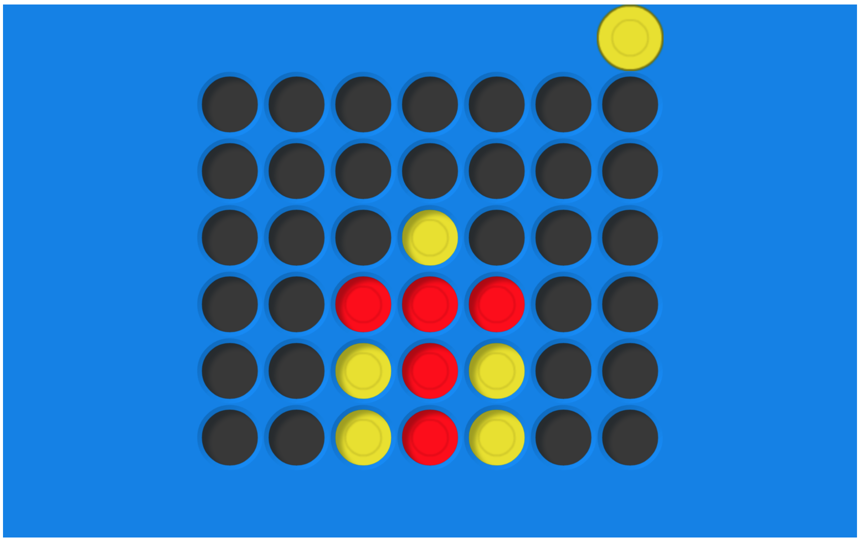


Figure 3.7: Connect 4 game board.

The original game where the control is established using a mouse is developed in Unity by KoalaMoala[2]. This implementation is a one player game, and the opponent is an AI based on Monte Carlo Tree Search algorithm. The player plays with yellow coins and the AI plays with red coins.

To use BCI as a main control unit, several modifications are done on this game. The background is changed to black to help the user elicit stronger SSVEPs. Mouse control is disabled. Three black-yellow SSVEP stimuli are added to top left, top right and bottom right corners of the game screen. These stimuli flicker at 5.455, 4.615 and 6 Hz respectively. The player's coin color is changed from yellow to green so it doesn't affect the SSVEPs elicited using black-yellow stimuli.

As the game starts, the player can enter their name using a BCI based speller, and then who gets to play the first turn is randomly selected, AI or the player. When it is the player's turn, the yellow coin stands on the top of the first row, fourth column. Then the player can choose to move their coin to left by gazing at the stimulus on the top left corner, or to right by gazing at the stimulus on the top right corner. When the coin stands on the selected column, the player can drop the coin by gazing at the right bottom corner. On the top left corner, the selected stimulus and the correlation coefficient is presented.

2. https ://github.com/KoalaMoala/Connect4

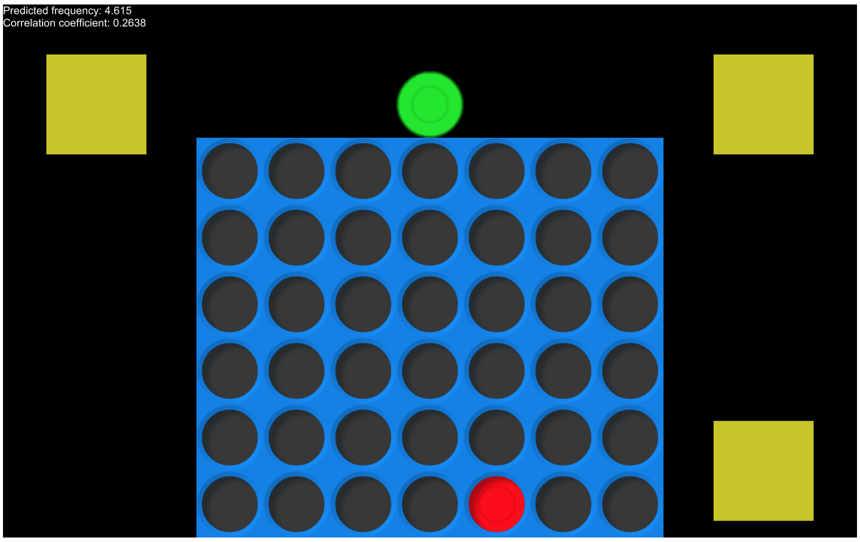


Figure 3.8: BCI based Connect 4 game board.

EEG signal is acquired from the Emotiv EPOC headset, and the raw signals are sent to Python script using Emokit driver. SSVEPs can be detected from P7, P8, O1 and O2 channels since they are located on the occipital lobe, therefore in this experiment only these four channels are used. To obtain clean data, each electrode is moisturized with a solution and every electrode is placed so that they are in contact with the scalp. Every three seconds, canonical correlation analysis is performed with the most recent three seconds data in Python script in order to predict the target stimulus and calculate the correlation coefficient. Emotiv EPOC has 128 Hz sampling rate, therefore each analysis has 384 samples as input. 3-second data is long enough for a correct classification, and short enough to enable real time gaming. After CCA is performed, the predicted stimulus frequency and the correlation coefficient are sent to the game via TCP. If the correlation coefficient is greater than or equal to 0.3, the corresponding command is performed.

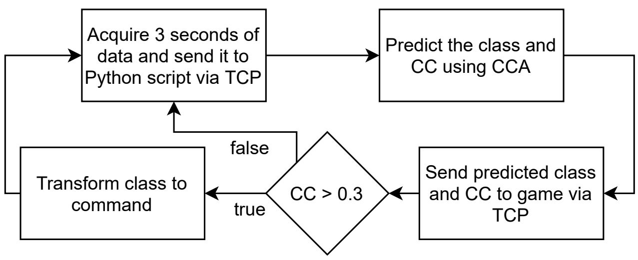


Figure 3.9: Signal processing loop.

3.4.2 Guitar Hero

Guitar Hero is a rhythm based game, where there are four activators on the bottom of the screen and many notes falling from top of the screen towards the activators. The activators are activated by pressing the A, S, D, and F keys on keyboard. The goal of the player is to activate the correct activators at the correct time to capture the falling notes. If the player fails to capture a note, they lose two points of the RockMeter, and if they successfully captures a note, they gain one point of the RockMeter. The initial point of RockMeter is 25, and the maximum point of RockMeter is 50. If the player fails to capture too many notes and the RockMeter point falls to 0, the player loses the game and the game ends. If the RockMeter is greater than 0 when the last note is hit, the player wins the game.

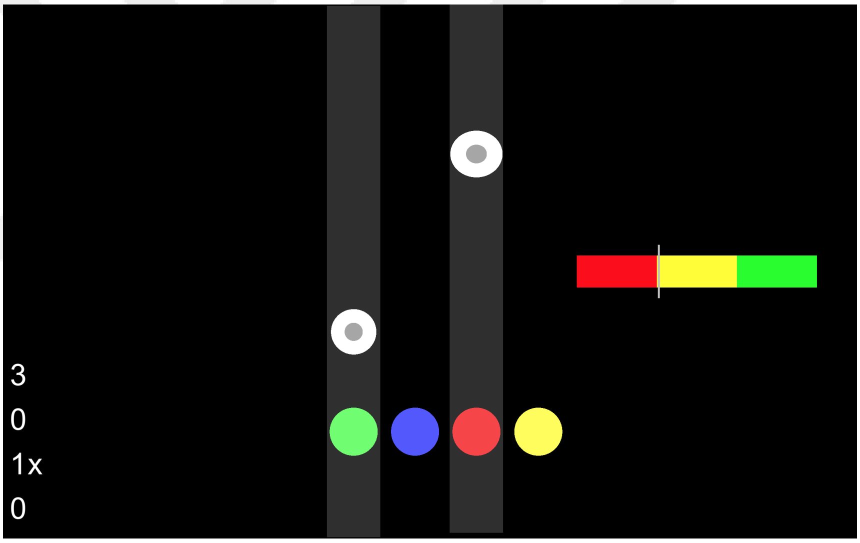


Figure 3.10: Guitar Hero game.

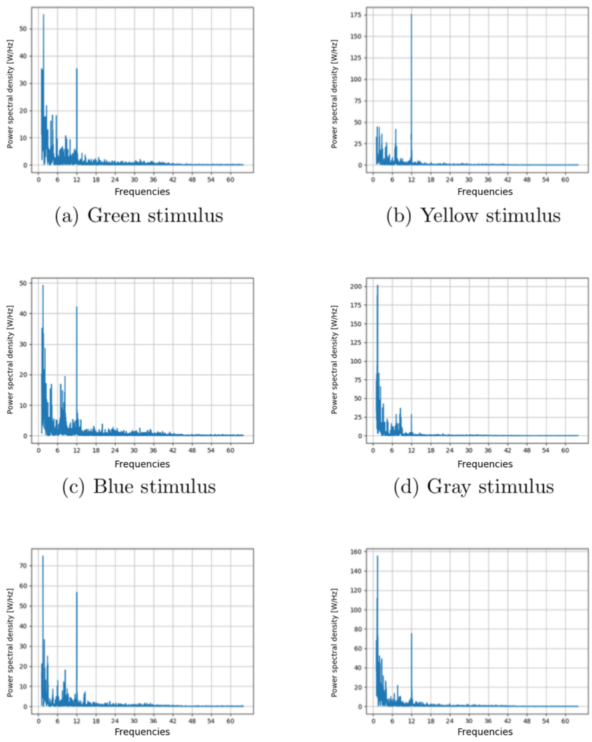
The falling speed of the notes determine the difficulty of the game. The goal of this study is to correctly classify the affective state of the player as high valence, low valence high arousal, and low valence low arousal, and arrange the difficulty of the game accordingly by incrementing or decrementing the speed of the falling notes. Every 10 seconds, the collected EEG data is examined and the valence and arousal level of the player is classified as low or high. If valence is high, it is interpreted as the player is having fun, and the game difficulty is not increased or decreased. If the valence is low, to determine if the player is stressed because the game is difficult, or if the player is bored because the game is too easy, the arousal level of the player is classified. If arousal level is low, the speed of notes is increased to make the game more difficult. If the arousal level is high, the speed of the notes is decreased to make the game easier.

4 RESULTS AND DISCUSSION

4.1 Experiments

4.1.1 Comparison of Different Colored Stimuli

In this experiment, the optimal color for SSVEP stimuli was studied. 12 Hz flickering stimuli were prepared in red, white, gray, yellow, blue and green. This experiment is a calibration process for BCI based Connect-4 game. Each player is expected to complete this step to find the best color for SSVEP stimulus. Since the Connect-4 game is tested on one player, in this experiment data was collected from one subject in one session. Power spectral density for O1 channel of each data was computed using periodogram method. In an ideal case, there should be peaks in the harmonics of the flickering frequency of the stimulus. The PSD plots of each colored stimuli are given below.



(a) Green stimulus

(b) Yellow stimulus

(c) Blue stimulus

(d) Gray stimulus

(e) White stimulus

(f) Red stimulus

Figure 4.1: PSD plots of O1 channel with different colored stimuli.

Since the stimulus frequency power is high and other frequencies' powers are low in yellow stimuli PSD, the stimulus color is chosen to be yellow.

4.1.2 Selection of Best 5 Stimulus Frequencies

This experiment is conducted to find the best 5 stimulus frequencies from 4.286, 4.615, 5, 5.455, 6, 6.667, 7.5, 8.571 and 10 Hz, in terms of accuracy and selection speed. The selection of these frequencies are explained as follows. An X Hz stimulus should flicker (change its color and return back to its previous color) X times in a second. Therefore it should change its color every $1/(2X)$ seconds. The monitor's refresh rate is 60 Hz, therefore the selected stimulus frequencies should be divisible by 60.

The selected stimuli were 300×300 pixel, yellow squares. Subjects were asked to gaze at one stimulus at a time. EEG data was collected and classified using CCA in real time until 200 target stimulus predictions which the correlation coefficient is greater than or equal to 0.3 were obtained. After obtaining 200 classifications, the experiment was repeated with a different frequency. 3 second data is used for classification and every 0.25 seconds a classification was done. In the best case scenario where every selection has a correlation coefficient greater than or equal to 0.3, this experiment takes 50 seconds. Since we select a frequency out of 9, a random classifier would give 0.11 accuracy.

Two subjects completed this experiment twice on two different days. Subjects started the experiment from the frequency 4.286 towards 10 in the first session, and from 10 towards 4.286 in the second session.

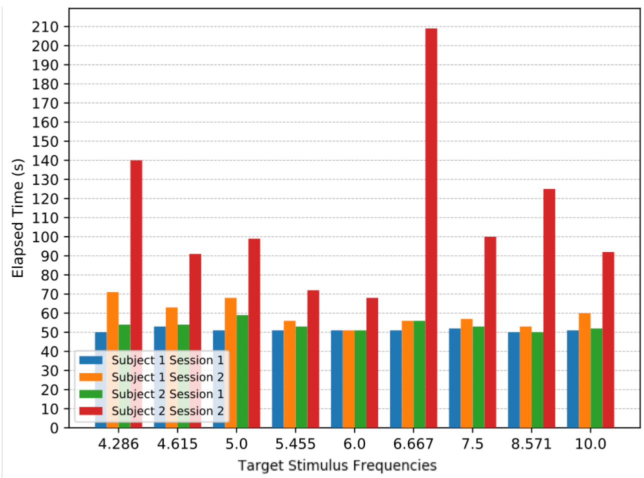


Figure 4.2: Comparison of frequencies in terms of elapsed time.

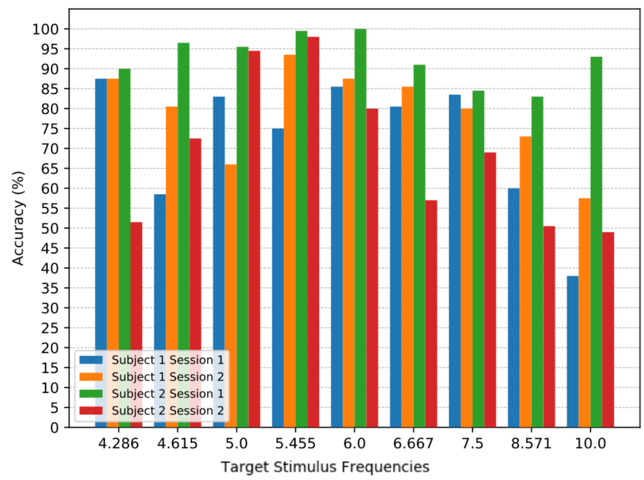


Figure 4.3: Comparison of frequencies in terms of accuracy.

4.1.2.1 Highest Accuracies

For each subject and session, 5 frequencies that have the highest accuracy are given in Table 4.1. Elapsed times are also given.

Table 4.1: 5 best frequencies in terms of accuracy and their elapsed time for each subject and session

Subject	Session	Ranking	Frequency	Accuracy	Elapsed Time (s)
1	1	1	4.286	87.50	50
1	1	2	6	85.50	51
1	1	3	7.5	83.50	52
1	1	4	5	83.00	51
1	1	5	5	83.00	51
1	2	1	5.455	93.50	56
1	2	2	4.286	87.5	71
1	2	3	6	87.50	51
1	2	4	6.667	85.50	56
1	2	5	4.615	80.50	63
2	1	1	6	100.00	51
2	1	2	5.455	99.50	53
2	1	3	4.615	96.50	54
2	1	4	5	95.50	59
2	1	5	10	93.00	52
2	2	1	5.455	98.00	72
2	2	2	5	94.50	99
2	2	3	6	80.00	68
2	2	4	4.615	72.50	91
2	2	5	7.5	69.00	100

4.1.2.2 Minimum Elapsed Time

For each subject and session, 5 frequencies that have the minimum elapsed time are given in Table 4.2. Accuracies are also given.

Table 4.2: 5 best frequencies in terms of elapsed time and their accuracy for each subject and session

Subject	Session	Ranking	Frequency	Accuracy	Elapsed Time (s)
1	1	1	4.286	87.50	50
1	1	2	8.571	60.00	50
1	1	3	6	85.50	51
1	1	4	5	83.00	51
1	1	5	6.667	80.50	51
1	2	1	6	87.50	51
1	2	2	8.571	73.00	53
1	2	3	5.455	93.50	56
1	2	4	6.667	85.50	56
1	2	5	7.5	80.00	57
2	1	1	6	83.00	50
2	1	2	5.455	100.00	51
2	1	3	4.615	93.00	52
2	1	4	5	99.50	53
2	1	5	10	84.50	53
2	2	1	6	80.00	68
2	2	2	5.455	98.00	72
2	2	3	4.615	72.50	91
2	2	4	10	49.00	92
2	2	5	5	94.50	99

4.1.2.3 Choosing Best Frequencies

In order to find the best 5 stimulus frequencies, the top 5 ranked frequencies in terms of accuracy and elapsed time were examined. In Table 4.3, how many times the frequencies were ranked top 5 are given in all sessions. If a stimulus frequency is in top 5 both in time comparison and accuracy comparison, it is listed in "Both" column.

Table 4.3: Number of times frequencies were ranked in top 5

Frequency	Accuracy Only	Elapsed Time Only	Both
4.286	1	0	1
4.615	2	0	1
5	2	0	1
5.455	0	1	3
6	0	0	4
6.667	0	0	2
7.5	2	2	0
8.571	0	3	0
10	0	1	1

Assuming ranking in top 5 in both comparisons is better than accuracy only and elapsed time only together, "accuracy only" and "elapsed time only" were awarded 2 points, and "both" was awarded 5 points. The best five frequencies are found to be 6 Hz (20 points), 5.455 Hz (17 points), 6.667 Hz (10 points), 5 Hz (9 points) and 4.615 Hz (9 points).

4.1.3 Comparing Specially Constructed Turkish Speller and Randomly Generated Speller

Two BCI spellers are constructed in rhombus layout. Both spellers include characters from A to Z, space, dot, delete and clear. For the first one, letter frequencies in Turkish are obtained from Serengil & Akın (2015). In the specially constructed speller, commonly used letters, such as A, E, İ, R, L are placed close to the center of the speller. Also, since looking at one stimulus for a longer period of time is more comfortable than changing the focus to a different stimulus, the layout is designed so that commonly used letters could be selected without changing the focus from one stimulus to another. In the randomly generated speller, every letter is placed to a random location.

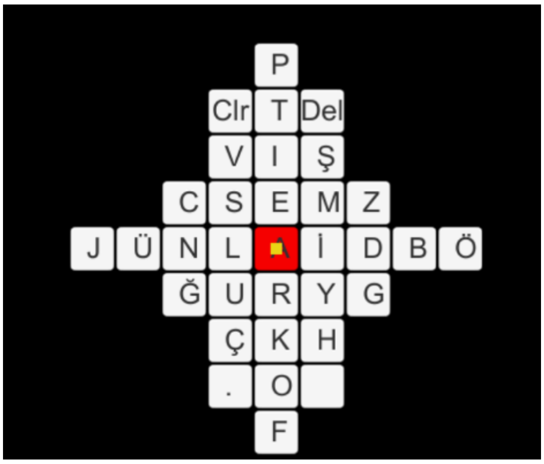


Figure 4.4: Specially constructed Turkish speller.

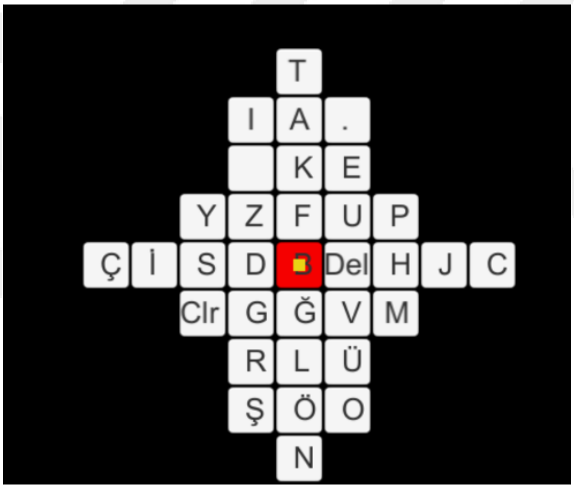


Figure 4.5: Randomly generated speller.

Both layouts are implemented in Unity. The cursor is moved down, left, up or right by looking at four yellow visual stimuli flickering at 6 Hz, 6.667 Hz, 8.571 Hz and 10 Hz respectively ; and when the cursor is on the desired letter, one can type the letter by looking at another stimuli flickering at 7.5 Hz. After making a selection, the cursor moves back to the center of the speller.

To test the performance of the spellers, 6 phrases ; BEN, EVET, HAYIR, MERHABA, KONUŞURUZ, SONRA GÖRÜŞMEK ÜZERE ; which translate to I, YES, NO, HELLO, WE WILL TALK, SEE YOU LATER, respectively, were spelled using both spellers and the elapsed time when typing these phrases are measured. These phrases are commonly used in Turkish, and the number of letters in these phrases vary. Subject was asked to spell the shortest phrase first, and longest phrase last. The tables below show the actual elapsed time as well as the optimal time for each phrase, they also show the difference between actual time and optimal time, divided by the number of selections needed to spell that phrase, to eliminate the effect of phrase length and letter placement on the difference. The optimal time is calculated as follows. For each letter in a phrase, number of selections to move the cursor onto the letter is calculated, and since one more selection must be done to actually type the letter, the number of selections is incremented by 1. Finally, since 3 second data is used for each classification, the calculated total is multiplied by 3. For instance, when calculating the optimal time for the phrase "BEN" in Turkish Layout, we first need to find the number of movements to move the cursor on letter B. The cursor needs to go right three times, and one more selection is necessary for typing the letter, so the total is 4 selections. After making a selection, the cursor moves back to the center. For the letter E, the cursor has to move up once, with typing selection, the total is 2. For letter N, the cursor has to move left twice, with typing selection the total becomes 3. Therefore for the whole phrase, the number of selections is 9. So the optimal time is 27 seconds. The results are as follows :

Table 4.4: The actual elapsed time typing the phrases using the speller with Turkish layout versus the optimal time (seconds)

Phrase	Actual Time	Optimal Time	Difference per Selection
BEN	45	27	2.00
EVET	42	36	0.50
HAYIR	81	39	3.23
MERHABA	60	51	0.69
KONUŞURUZ	141	87	1.86
SONRA GÖRÜŞMEK ÜZERE	288	192	1.50

Table 4.5: The actual elapsed time typing the phrases using the speller with random layout versus the optimal time (seconds)

Phrase	Actual Time	Optimal Time	Difference per Selection
BEN	60	30	3.00
EVET	66	48	1.13
HAYIR	142	60	4.10
MERHABA	180	72	4.50
KONUŞURUZ	162	102	1.76
SONRA GÖRÜŞMEK ÜZERE	553	237	4.00

Even in a best case scenario where each spelling takes optimal time, Turkish layout takes much less time compared to random layout. The difference per selection shows that in the experiment, typing using random layout takes much more time. This is probably due to the fact that, in Turkish layout one can find the letters easily because the user can guess where the letters are located, but in random layout, it takes more time trying to find the location of the desired letter, also user makes unintentional selections while looking for the letter, therefore needs to correct them by selecting delete command.

The following tables show the time takes to spell each character, as well as the optimal time and difference per selection.

Table 4.6: The actual elapsed time typing the letters using the speller with Turkish layout versus the optimal time (seconds)

Letter	BEN	EVET	HAYIR	MERHABA	KONUŞURUZ	SONRA GÖRÜŞMEK ÜZERE	Average	Optimal Time	Difference per Selection
A			3	3,3		3	3	3	0
B	27			12			19.5	12	7.5
E	9	6,6		6		6,12,6	7.29	6	1.29
G						33	33.00	12	21
H			12	19			15.5	12	3.5
I			15				15	9	6
K					9	12	10.5	9	1.5
M				12		18	15	9	6
N	9				9	12	10	9	1
O					12	15	13.5	12	1.5
Ö						15	15	15	0
R			6	6	6	6,9,9	7	6	1
S						9	9	9	0
Ş					15	12	13.5	12	1.5
T		15					15	12	3
U					9,15,12		12	9	3
Ü						18,12	15	12	3
V		15					15	9	6
Y			9				9	9	0
Z					12	12	12	12	0
DELETE			15,15		21,15	15,18	16.50	15	1.5
SPACE						15,15	15	15	0

Table 4.7: The actual elapsed time typing the letters using the speller with random layout versus the optimal time (seconds)

Letter	BEN	EVET	HAYIR	MERHABA	KONUŞURUZ	SONRA GÖRÜŞMEK ÜZERE	Average	Optimal Time	Difference per Selection
A			18	18,12		12	15	12	3
B	12			15			13.5	3	10.5
E	18	12,18		12		12,18,18,12	15	12	3
G						12	12	9	3
H			12	15			13.5	9	4.5
I			15				15	15	0
K					9	15	12	9	3
M				12		15	13.5	12	1.5
N	30				33	15	26	15	11
O					21	15	18	15	3
Ö						15	15	12	3
R			24	33	12	18,18,18	20.5	12	8.5
S						9	9	9	0
Ş					24	21	22.5	15	7.5
T		15					15	15	0
U					15,9,9		11	9	2
Ü						12,12,18	14	12	2
V		9					9	9	0
Y			18				18	12	6
Z					9	33,15	19	9	10
DELETE		9	21,12	6,6	6,6	6,9,6,6,6,9,9,12,6,15,6	8.67	6	2.67
SPACE						30,15,15	20	12	8

The mean of the differences per selection in Turkish layout is 3.10, while in random layout is 4.19. One can say that random layout is slower, and more error prone, because in Turkish layout delete was used 6 times, where in random layout it was used 16 times.

It seems like the place of a letter in a phrase doesn't affect the spell time. For instance, in Turkish layout, the first and third E took 6 seconds to spell, while the second E took 12 second in the phrase SONRA GÖRÜŞMEK ÜZERE. In random layout, first U took 15 seconds, while second and third seconds took 9 seconds in the phrase KONUŞURUZ.

4.2 Tests on Implemented Games

4.2.1 Connect 4

To assess the performance of the game, four sessions of one minute long game playing is conducted. In three sessions, the player is focused on one stimulus, different stimuli in each session. In the final session, the player is focused only on the board since the user needs to examine the board to select a column to drop their coin.

Since one session lasts one minute and CCA is performed every 3 seconds, about 20 analyses are performed in each session. Since the classification results whose correlation coefficient is greater than or equal to 0.3 are translated into a command, only these classifications are included in the assessment. In Table 4.8 the ratio of correctly

classified stimulus frequencies to all classifications are given for each stimulus. The commands for moving the coin to the left or right, or dropping the coin is achieved with an average of 0.87 accuracy.

Table 4.8: The ratio of correctly classified stimulus frequencies to all classifications.

Stimulus frequency	Accuracy
4.615 Hz	1
5.455 Hz	0.78
6 Hz	0.82

In an ideal game, when the user examines the board, all correlation coefficients must be less than 0.3, but even if the correlation coefficients are greater than 0.3, it is most important to not drop the coin. If the coin moves towards right or left, this action can be undone, but if the coin is dropped, it would be a mistake that cannot be undone. When the player examines the board, the ratio of number of classifications with a correlation coefficient less than 0.3 to all classifications is 0.48 and ratio of number of classifications with correlation coefficient greater than 0.3 that doesn't choose to "drop coin" command to all classifications with correlation coefficient greater than 0.3 is 0.95.

4.2.2 Guitar Hero

To correctly classify the valence and arousal levels of the player, three versions of the game is implemented. In one version the note speed is medium, which the player enjoys playing the game, therefore the collected data while playing this game was labeled as high valence. Another version has many notes and note speed is high, therefore the collected data is labeled as low valence and high arousal. Finally in the last version has a few notes and note speed is very low, therefore the collected data while playing this version is labeled as low valence and low arousal.

Initially, one classifier with three classes is implemented. The classes were labeled as HV, LVLA and LVHA. After obtaining low classification accuracy on tests of 3-class classifier, two classifiers are trained, one for classifying the valence level as high or low, and another one is for classifying the arousal levels as high or low. The training and evaluation of both 3-class classifier and valence/arousal classifiers are same and explained below.

In multiple sessions, EEG data is collected using Emotiv EPOC headset while playing three versions of the game. The sessions are split into training, validation and test sets. Training set is used to train many classification algorithms, validation set is used to choose one of the classifiers, and test set is used to evaluate the performance of the classifier.

Since Emotiv EPOC's sampling rate is 128 samples per second, and we want to classify the valence and arousal levels every 10 seconds, trials are constructed from 1280 samples. For each channel in a trial, the wavelet coefficients are obtained using Symlet 5 wavelet. The following features are extracted from these coefficients. The entropy ; statistical features such as the 5^{th}, 25^{th}, 75^{th} and 95^{th} percentile, median, mean, standard deviation, variance and root mean square ; number of zero crossings and mean crossings[1].

After feature extraction, the following algorithms are tested using validation set ; support vector machines, 5 nearest neighbors, 3 nearest neighbors, linear discriminant analysis, random forest, logistic regression, Naive Bayes, stochastic gradient descent,

1. https ://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/

decision tree, quadratic discriminant analysis, gradient boosting with 100, 500, 1000 and 5000 estimators, MLP with 100, 500, 1000 and 5000 hidden layers. All of the classifications are made using Python programming language (version 3.7.9) and Scikit-learn library (version 0.24.1).

This process is done four times with different channel sets. In one training, only frontal lobe channels (AF3, AF4, F3 and F4 channels) are used. In others, only the channels on the left hemisphere, only the channels on the right hemisphere and all channels are used. Channel selection is done before feature extraction. The algorithm that yields the highest F1 score on validation set is chosen as the classifier.

For 3-class classifier, the selected classifier is linear discriminant analysis, and all channels are used. This model's performance on training, validation and test sets are given in Table 4.9.

Table 4.9: The performance of 3-class classifier on training, validation and test sets.

Dataset	Accuracy	F1 Score	Confusion Matrix
Training set	0.96	0.96	70 2 0 2 69 1 3 1 67
Validation set	0.71	0.70	15 1 1 5 9 2 3 3 12
Test set	0.43	0.43	2 8 0 5 4 1 0 2 6

For arousal, the selected classifier is gradient boosting with 500 estimators, and only the channels on the right hemisphere are used. This model's performance on training, validation and test sets are given in Table 4.10.

Table 4.10: The performance of arousal level classifier on training, validation and test sets.

Dataset	Accuracy	F1 Score	Confusion Matrix
Training set	1.0	1.0	103 0 0 103
Validation set	0.74	0.74	16 3 6 10
Test set	0.85	0.85	14 0 4 9

For valence, the selected classifier is MLP with 500 hidden layers, and all channels are used. This model's performance on training, validation and test sets are given in Table 4.11.

Table 4.11: The performance of valence level classifier on training, validation and test sets.

Dataset	Accuracy	F1 Score	Confusion Matrix
Training set	0.85	0.85	61 2 16 44
Validation set	0.76	0.76	12 3 4 10
Test set	0.72	0.72	18 12 3 21

The tests above are done after the data was collected. The following tests yield the results of real time classification. Every 10 seconds, the valence and arousal levels of the player are predicted through the game. These tests are done using all three versions of the game, 6 in total. The games' difficulty is not changed according to the results, because the aim of this experiment is to evaluate the classifiers' performances when they work in real time. Each test lasted 120 seconds. The results are given in Table 4.12 and Table 4.13.

Table 4.12: Test results of 3-class classifiers.

Test number	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Average accuracy
Expected class	LVLA	LVLA	LVHA	LVHA	HV	HV	
Number of LVLA predictions	0	0	0	0	3	0	
Number of LVHA predictions	11	4	12	12	8	12	
Number of HV predictions	1	8	0	0	1	0	
Accuracy	0	0	1	1	0.08	0	0.35

In the Table 4.12, one can see that majority of the predictions is LVHA, which is interpreted as the player finds the game difficult. Also, the average accuracy (0.35) is very close to a random classifier accuracy (0.33). Therefore, this classifier is not employed in the DDA system.

Table 4.13: Test results of valence and arousal classifiers.

Test number	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Average accuracy
Expected class	LVLA	LVLA	LVHA	LVHA	HV	HV	
Number of LVLA predictions	5	2	0	1	1	2	
Number of LVHA predictions	1	2	4	4	0	2	
Number of HVLA predictions	5	5	0	1	1	2	
Number of HVHA predictions	0	1	7	5	9	4	
Arousal accuracy	0.91	0.70	1.00	0.82			0.86
Valence accuracy	0.55	0.40	0.36	0.45	0.91	0.60	0.55
3-class accuracy (HV, LVHA or LVLA)	0.45	0.20	0.36	0.36	0.91	0.60	0.48

In the Table 4.13, the arousal accuracy shows the accuracy without taking valence into consideration. In this case, both HVHA and LVHA mean high arousal. Similarly both HVLA and LVLA mean low arousal. The average accuracy for arousal is found to be 0.86. High arousal is predicted more accurately (0.91) compared to low arousal (0.81).

The valence accuracy shows the accuracy without taking arousal into consideration. In this case, both HVHA and HVLA mean high valence. Similarly both LVHA and LVLA mean low valence. The average accuracy for valence is found to be 0.55. High valence is predicted more accurately (0.76) compared to low valence (0.44). The reason for this could be the fact that half of low valence dataset was constructed with the data collected when the user played the difficult version of the game, and the other half was constructed with the data collected when the user played the easy version of the game. So the information for low valence may not be sufficient to train a successful model.

If the player's valence level is classified as high, no difficulty change will be applied to the game, so HVHA and HVLA mean HV. Only if valence level is classified as low, the game decides to increase or decrease the game difficulty according to the arousal level. So, actually there are three classes, HV, LVHA and LVLA. Since this is the case, another metric, 3-class accuracy, was added. The average accuracy is found to be 0.48.

Finally, a game that changes its difficulty according to the valence and arousal levels predicted in real time is developed. The classifiers mentioned above are used as the valence and arousal classifiers in this game, which are MLP and gradient boosting classifiers respectively. For testing purposes, this game has three different starting game speed ; easy, medium and difficult. The game speed between 5 and 10 units per second is declared as an enjoyable experience by the player, therefore this interval is interpreted as a high valence zone, which is highlighted with green color in the graphs. The maximum speed is set to 20 units per second, and the minimum speed is set to 1 unit per second. When the game changes its difficulty after a LVLA or LVHA prediction, its speed is increased or decreased by one unit per second.

The graph below shows the game difficulty changes for the game that has an easy start. The starting speed for this game is set to 1 unit per second. When the game speed drops below 5 units per second, the player finds the game boring, so the expected valence and arousal levels are low. It is expected for game to increase its difficulty.

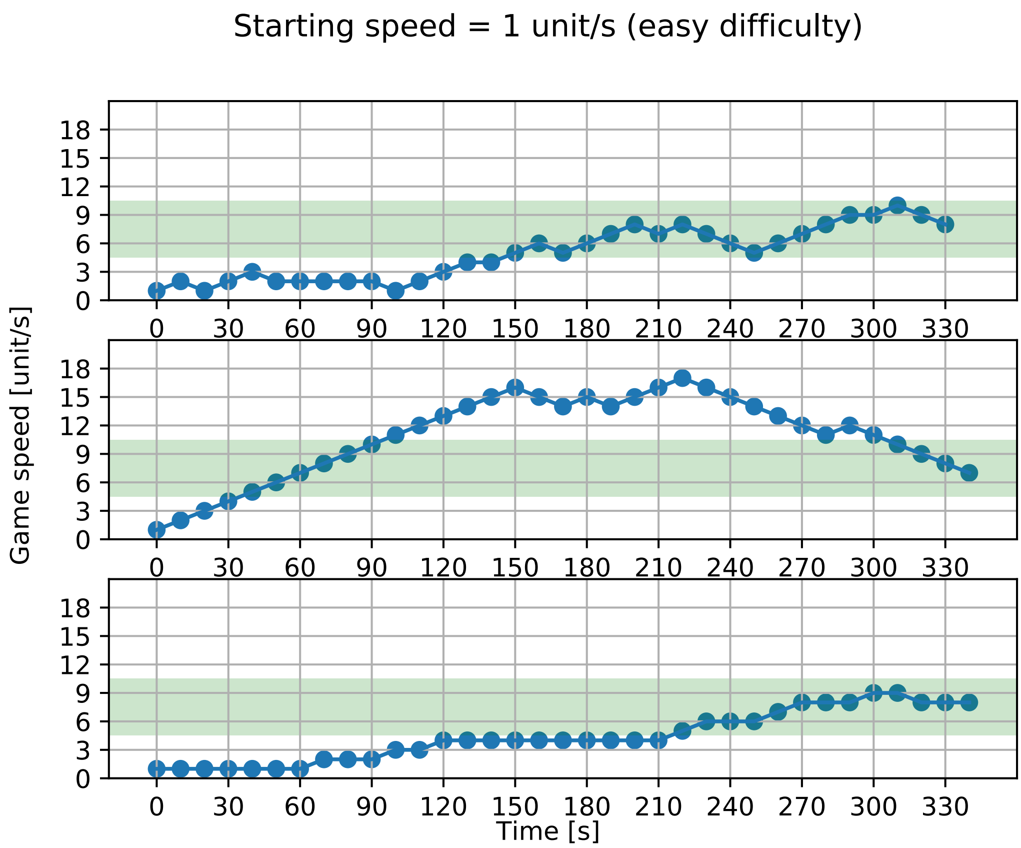


Figure 4.6: Changes in game difficulty when the game has an easy start.

In all three tests, the game successfully increased its difficulty and entered to the high valence zone. In the second test, even though its difficulty is increased too much, it then reduces its difficulty back to the high valence zone.

The graph below shows the game difficulty changes for the game that has a medium start. The starting speed for this game is set to 7 units per second. The expected valence level is high. It is expected for game to maintain its difficulty between 5 and 10 units per second.

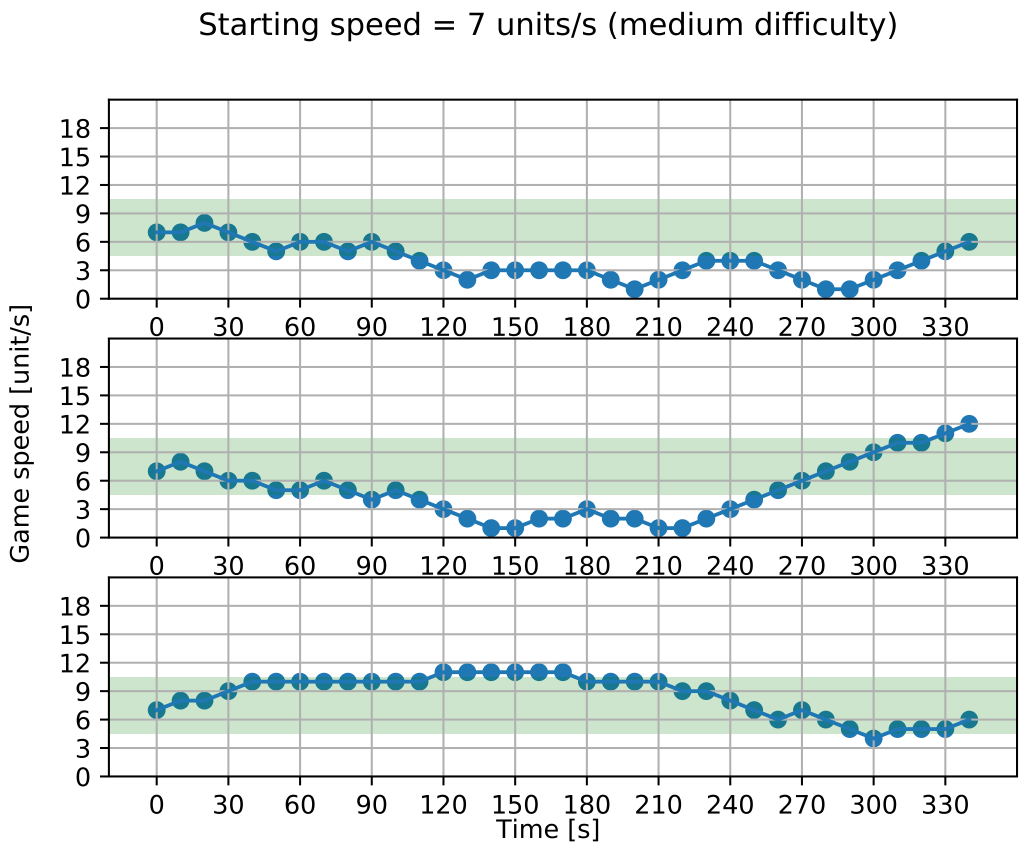


Figure 4.7: Changes in game difficulty when the game has a medium start.

In the third test, the game speed is in the high valence zone or its vicinity. In the first and second tests, after 100 seconds the difficulty is decreased to the point that it exits the high valence zone, but towards the end its difficulty is increased again.

The graph below shows the game difficulty changes for the game that has a difficult start. The starting speed for this game is set to 12 units per second. When the game speed exceeds 10, the player finds the game difficult, so the expected valence level is low and arousal level is high. It is expected for game to decrease its difficulty.

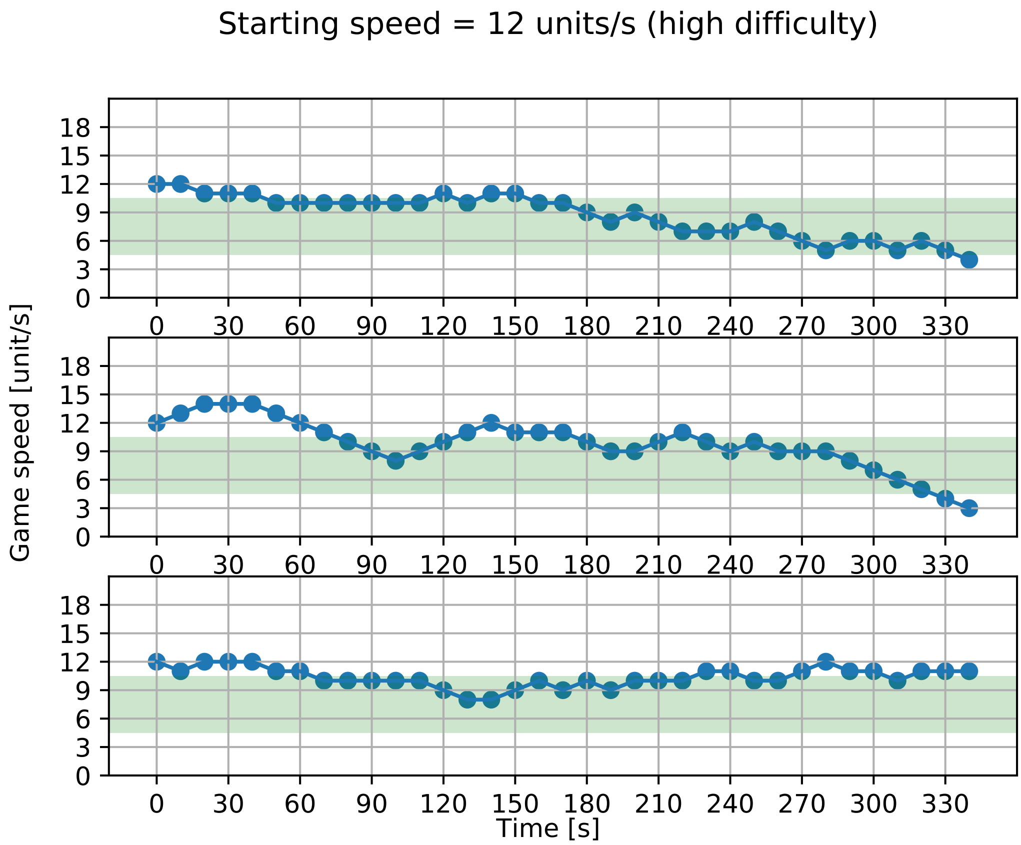


Figure 4.8: Changes in game difficulty when the game has a difficult start.

In all three tests, the game successfully decreased its difficulty and entered to the high valence zone. In the second test, even though its difficulty is increased in the beginning, it then reduces its difficulty back to the high valence zone. In the third test, the difficulty exits the high valence zone but its highest speed is limited to 12 units per second.

5 CONCLUSION

Even though brain computer interfaces are commonly designed for enhancing the life quality of patients with severe disabilities, they can also be used by healthy people in many fields. In this thesis, their use on video games are studied. It was found that BCIs can be used in video games in many ways, such as the main control, side controls, or emotion recognition for difficulty adjustment. Two BCI based video games were desired to be developed, one that uses BCI as the main control mechanism, and one that uses BCI to classify the valence and arousal levels of the player to dynamically change the game difficulty.

Before actually implementing these applications, many other experiments were conducted. For instance, to select the best visual stimuli for SSVEP based game, an experiment was conducted to select the stimulus color. It was found that yellow colored stimulus yields the best results. Another experiment was conducted to find the best frequencies for SSVEP based application, which revealed that the frequencies flickering at 6 Hz, 5.455 Hz , 6.667 Hz, 5 Hz and 4.615 Hz resulted in highest accuracies in minimum time.

Besides from the games, an SSVEP based speller was implemented. Four different visual stimuli were used to move the cursor around the speller to select the desired letters. To find the effect of the letter placement, two spellers were compared. In one of these spellers, the most commonly used letters in Turkish language were placed close to the cursor and to each other. In the other one, letters were placed randomly. It was found that subjects were able to type desired words and sentences in shorter time and with less errors using the speller where commonly used letters are placed closer.

Then two BCI based video games are implemented. One game is designed to be played with elicited SSVEPs. The other game's main control is keyboard, but as the player plays the game their affective state is predicted and the game difficulty is adjusted accordingly.

The first game, Connect 4, is originally a two-player board game, which the players take turns to drop their coins to a 6 row, 7 column board to make a vertical, horizontal or diagonal line composed of 4 coins. This game was originally planned to have seven visual stimuli, each of them placed on one column. The user was supposed to look at the desired column, and the coin would be dropped to the selected column. This implementation was realized, but the game controls were not accurate enough for an enjoyable game play experience. One reason for low accuracy is the changes made on the stimuli. First of all their size are shrank, secondly they are placed very close with other stimuli, which affects the SSVEPs elicited in brain. Finally, the classifier is expected to predict one class from 7 different classes, which yielded in low accuracy. Instead of using seven different stimuli for each column, the game controls are changed so that the player interacts with fewer stimuli. It is implemented with three selections, two for moving the coin to left or right, and one for dropping the coin. These selections are realized by classifying the SSVEPs of the player. To elicit three different SSVEPs, three visual stimuli flickering at 4.615 Hz, 5.455 Hz and 6 Hz are used. The average accuracy is found to be 0.87.

The second game, Guitar Hero is a music rhythm game in which the player tries to catch the falling notes with activators using the A,S,D,F keys on keyboard. The difficulty of the game is determined by the speed of falling notes. The aim of this study is to correctly classify the valence and arousal levels of the player to increase the difficulty when they're bored, decrease the difficulty when they're stressed or anxious, or not changing the difficulty if the player is having fun. Originally the valence and arousal levels were planned to be classified as low, medium and high, but having a medium valence and arousal level doesn't give a meaningful feedback to the game, so they are classified as low and high. The data set is constructed from scratch using the implemented game. For features, the DWT coefficients are calculated and many features are extracted from these coefficients, such as statistical features, number of zero and mean crossings. In the non-real time tests, the average accuracy of valence prediction is found to be 0.72, and the average accuracy of arousal prediction is 0.85. In the real time tests where the game difficulty is not changed, the average accuracy of valence predictions is found to be 0.55, the average accuracy of arousal levels is 0.86, and the three-class accuracy is 0.48. The arousal accuracy is satisfactory, but the valence accuracy is lower than expected. There are many reasons for such low accuracy. First of all, the data used for low valence training is composed of both LVLA and LVHA data, so the classifier doesn't perform as well it does in classifying high valence data, which is trained using only HV data. Secondly, there are many factors affecting the valence levels of the player besides from the game itself. For instance, while conducting the second test for the game where game difficulty is changed according to the classification results, the subject was going through a bad day, and the valence level was almost never classified as high. Similarly, while conducting the third test, the subject was going through a very good day, and valence level was classified as high many times. Another reason for such low accuracy is that, high valence data may not be entirely correctly labeled, that is, it is not guaranteed for subject to experience positive emotions during the acquisition of high valence data used in training set. Even though the real time test results are not very satisfactory, the results of tests where the game difficulty is adjusted dynamically show that the valence and arousal levels are predicted good enough to increase or decrease the game difficulty to an enjoyable speed.

To conclude, SSVEP based game was successfully controlled by the player. The player was able to perform desired commands both in short time and with high accuracy. The state-of-the-art studies show that SSVEP based BCIs yield between 70% - 96% accuracy (Zerafa et al., 2018). The implemented SSVEP based game is as successful as the state-of-the-art SSVEP studies. Even though, the online classification results of affective states of the player was not very successful, the performance of dynamic difficulty adjustment system is satisfactory. The latest research about recognizing the valence level of subjects yield between 61.17% and 87.44% accuracy and arousal level of subjects yield between 64.84% and 88.49% accuracy (Torres et al. 2020). The trained classifiers are able to predict the valence and arousal levels as accurately as the state-of-the-art research.

One main problem in this study is the lack of subjects. As future studies, the conducted experiments will be repeated with more subjects. Similarly, the data set used for training the DDA classifiers should be enhanced with variety of subjects' data. In order to verify the accuracy of the classifiers, further data analysis techniques, such as cross validation, are needed. These analyses will be conducted after gathering more data thus ensuring more reliable results. The implemented DDA system's performance can be tested on other games. Finally the implemented EEG based DDA system can be compared with other DDA systems, such as DDA based on heuristics, to evaluate its effect on the user experience.

REFERENCES

Abdulkader, S. N., Atia, A., & Mostafa, M. S. M. (2015). Brain computer interfacing: Applications and challenges., *Egyptian Informatics Journal* 16(2): 213-230.

Allison, B., Graimann, B., & Gräser, A. (2007). Why use a BCI if you are healthy. *In ACE Workshop-Brain-Computer Interfaces and Games* pp. 7-11.

Ang, D., & Mitchell, A. (2017). Comparing Effects of Dynamic Difficulty Adjustment Systems on Video Game Experience., *In Proceedings of the Annual Symposium on Computer-Human Interaction in Play* pp. 317-327.

Cecotti, H. (2010). A self-paced and calibration-less SSVEP-based brain–computer interface speller. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 18(2): 127-133.

Congedo, M., Goyat, M., Tarrin, N., Ionescu, G., Varnet, L., Rivet, B., ... & Jutten, C. (2011). " Brain Invaders": a prototype of an open-source P300-based video game working with the OpenViBE platform.

Csikszentmihalyi, M. (1997). Finding flow (Vol. 131). New York: Basic Books.

Finke, A., Lenhardt, A., & Ritter, H. (2009). The MindGame: a P300-based brain–computer interface game. *Neural Networks* 22(9): 1329-1333.

Friman, O., Luth, T., Volosyak, I., & Graser, A. (2007). Spelling with steady-state visual evoked potentials. *In 2007 3rd International IEEE/EMBS Conference on Neural Engineering* pp. 354-357.

Giakoumis, D., Tzovaras, D., Moustakas, K., & Hassapis, G. (2011). Automatic recognition of boredom in video games using novel biosignal moment-based features. *IEEE Transactions on Affective Computing* 2(3): 119-133.

Gürkök, H., Nijholt, A., Poel, M., & Obbink, M. (2013). Evaluating a multi-player brain–computer interface game: Challenge versus co-experience. *Entertainment Computing* 4(3): 195-203.

Holz, E. M., Höhne, J., Staiger-Sälzer, P., Tangermann, M., & Kübler, A. (2013). Brain–computer interface controlled gaming: Evaluation of usability by severely motor restricted end-users. *Artificial intelligence in medicine* 59(2): 111-120.

Lalor, E. C., Kelly, S. P., Finucane, C., Burke, R., Smith, R., Reilly, R. B., & Mcdarby, G. (2005). Steady-state VEP-based brain-computer interface control in an immersive 3D gaming environment. *EURASIP Journal on Advances in Signal Processing* 2005(19): 706906.

Liarokapis, F., Debattista, K., Vourvopoulos, A., Petridis, P., & Ene, A. (2014). Comparing interaction techniques for serious games through brain–computer interfaces: A user perception evaluation study. *Entertainment Computing* 5(4): 391-399.

Liu, C., Xie, S., Xie, X., Duan, X., Wang, W., & Obermayer, K. (2018). Design of a video feedback SSVEP-BCI system for car control based on improved MUSIC method. *In 2018 6th International Conference on Brain-Computer Interface (BCI)* pp. 1-4.

Lotte, F. (2011). Brain-computer interfaces for 3D games: hype or hope?. In *Proceedings of the 6th International Conference on Foundations of Digital Games* pp. 325-327.

Maby, E., Perrin, M., Bertrand, O., Sanchez, G., & Mattout, J. (2012). BCI could make old two-player games even more fun: a proof of concept with “Connect Four”. *Advances in Human-Computer Interaction*, *2012*.

Mikami, K., & Kondo, K. (2017, June). Adaptable Game Experience Based on Player's Performance and EEG. *In 2017 Nicograph International (NicoInt)* pp. 1-8.

Nijholt, A. (2008). BCI for games: A ‘state of the art’survey. In *International Conference on Entertainment Computing*, Springer, Berlin, Heidelberg, pp. 225-228

Nijholt, A., Bos, D. P. O., & Reuderink, B. (2009). Turning shortcomings into challenges: Brain–computer interfaces for games. *Entertainment computing* 1(2): 85-94.

Park, S., Sim, H., & Lee, W. (2014). Dynamic game difficulty control by using EEG-based emotion recognition. *Int J Control Autom*, 7: 267-272.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

Petrantonakis, P. C., & Hadjileontiadis, L. J. (2009). Emotion recognition from EEG using higher order crossings. *IEEE Transactions on information Technology in Biomedicine* 14(2): 186-197.

Pour, P. A., Gulrez, T., AlZoubi, O., Gargiulo, G., & Calvo, R. A. (2008). Brain-computer interface: Next generation thought controlled distributed video game development platform. In *2008 IEEE Symposium On Computational Intelligence and Games* pp. 251-257.

Reuderink, B., Nijholt, A., & Poel, M. (2009). Affective Pacman: A frustrating game for brain-computer interface experiments. *In International conference on intelligent technologies for interactive entertainment,* Springer, Berlin, Heidelberg, pp. 221-227.

Russell, J. A. (1980). A circumplex model of affect. *Journal of personality and social psychology* 39(6): 1161.

Siuly, S., Li, Y., & Zhang, Y. (2016). EEG signal analysis and classification. *IEEE Trans Neural Syst Rehabilit Eng*, 11: 141-4.

Stein, A., Yotam, Y., Puzis, R., Shani, G., & Taieb-Maimon, M. (2018). EEG-triggered dynamic difficulty adjustment for multiplayer games. *Entertainment computing*, 25: 14-25.

Szegletes, L., Köles, M., & Forstner, B. (2014). The design of a biofeedback framework for dynamic difficulty adjustment in games. In *2014 5th IEEE Conference on Cognitive Infocommunications (CogInfoCom)* pp. 295-299.

Tangermann, M. W., Krauledat, M., Grzeska, K., Sagebaum, M., Vidaurre, C., Blankertz, B., & Müller, K. R. (2008, December). Playing pinball with non-invasive BCI. In *Proceedings of the 21st international conference on neural information processing systems*, Curran Associates Inc. pp. 1641-1648

Torres P, E. P., Torres, E. A., Hernández-Álvarez, M., & Yoo, S. G. (2020). EEG-based BCI emotion recognition: A survey. *Sensors*, *20*(18), 5083.

Valbuena, D., Sugiarto, I., & Gräser, A. (2008). Spelling with the Bremen brain-computer interface and the integrated SSVEP stimulator.

van Vliet, M., Robben, A., Chumerin, N., Manyakov, N. V., Combaz, A., & Van Hulle, M. M. (2012). Designing a brain-computer interface controlled video-game using consumer grade EEG hardware. *In 2012 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC)* pp. 1-6.

Yohanes, R. E., Ser, W., & Huang, G. B. (2012). Discrete wavelet transform coefficients for emotion recognition from EEG signals. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society* pp. 2251-2254

Zerafa, R., Camilleri, T., Falzon, O., & Camilleri, K. P. (2018). To train or not to train? A survey on training of feature extraction methods for SSVEP-based BCIs. *Journal of Neural Engineering*, *15*(5), 051001.

Zohaib, M. (2018). Dynamic difficulty adjustment (DDA) in computer games: A review. *Advances in Human-Computer Interaction*.

BIOGRAPHICAL SKETCH

Elifsu Filiz was born in 1992. She has graduated from computer engineering department in İstanbul Bilgi University in 2015. Since 2016 she has been working in İstanbul Bilgi University as research assistant.

PUBLICATIONS

— Filiz, E., & Arslan, R. B. (2020). Design and Implementation of Steady State Visual Evoked Potential Based Brain Computer Interface Video Game. In *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)* pp. 335-338.

— Filiz, E., & Arslan, R. B. (2020). Durgun Hal Görsel Uyarılmış Potansiyel Tabanlı Beyin Bilgisayar Arayüzü Video Oyununun Tasarımı ve Gerçekleştirilmesi. In *2020 IEEE 28th Signal Processing and Communications Applications Conference (SIU)* pp. 1-4.