České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačů

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Nikita Sokolsky**

Studijní program: Softwarové technologie a management
Obor: Softwarové inženýrství

Název tématu: **Tvorba her pro systém biofeedbackové terapie**

Pokyny pro vypracování:

Cílem práce je tvorba her, které budou součástí většího softwarového projektu vyvíjeného v rámci skupiny Biodat. Jedná se o komplexní nástroj pro administraci biofeedbackové terapie. Student se seznámí se současnou verzí projektů a nastuduje způsob fungování systému. Následně provede návrh vlastních her, které budou přidány ke stávajícím. Bakalářská práce bude obsahovat popis návrhu, realizace a testování těchto her.

Seznam odborné literatury:

Saeid Sanei, J.A. Chambers. EEG Signal Processing. Wiley-Blackwell (an imprint of John Wiley & Sons Ltd)

Xavier J. Caro, Earl F. Winter. EEG Biofeedback Treatment Improves Certain Attention and Somatic

Symptoms in Fibromyalgia : A Pilot Study. Science + Business Media, LLC 2011. Published online : 9 June 2011.

Vedoucí: Mgr. Michal Vavrečka, Ph.D.

Platnost zadání: do konce letního semestru 2014/2015

L.S.

doc. Ing. Filip Železný, Ph.D.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 27. 2. 2014

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science and Engineering

BACHELOR THESIS

Games for biofeedback therapy system

Nikita Sokolsky

Thesis supervisor: Vavrečka Michal Mgr., Ph.D.

Study program: Software Technology and Management
Field of study: Software Engineering
Prague, 2014

Acknowledgements

I would like to thank:

- My supervisor Michal Vavrečka for his ideas and feedback
- My colleague Radek Procházka for technical assistance with the project
- My friend Mikhail Gudyrin for his help in the testing phase

Statement

I hereby state, that I have completed this thesis on my own and I have properly specified all the information sources used, according to the Guideline about observance of the ethical principles concerning university theses.

In Prague, 23rd May, 2014 Nikita Sokolsky

Abstract

This bachelor thesis is concerned with the creation of biofeedback games for therapy purposes. The games are built on top of an existing software project developed for the purposes of Neurofeedback created by BioDat Group. The games present clear visual and audio stimuli to the user. This thesis describes the theoretical research behind neurofeedback therapy as well as a description of the created games and their implementation.

Abstrakt

Daná bakalařská práce zabýva se tvořbou zpetnovázebních her zá učely terapie. Hry jsou součastí existujicího neurofeedbackového software vytvořeneho skupinou BioDat. Cílem her bylo dosahnout zřetelneho vizualního a zvukoveho feedbacku na základe mozkových vln. Práce obsahuje rešerše teoretické strány neurofeedbacku, popís vytvořených her a jejích implementace.

Contents

Chapter 1
Introduction

Computers play an ever increasing role in the medical field: from 3-dimensional MRI machines to gyroscopic sensors. Psychological therapy is no exception to this rule – while many therapists still rely on traditional methods such as psychoanalysis and medication, an increasing number of treatments arise utilizing computers and various sensors such as blood pressure, heart rhythm and EEG data.

Biofeedback is one of the methods used in the medical field for treatment. It measures the basic indicators of a person's body — blood pressure, heart rate, skin temperature, brain waves and so on to give the patient a chance to gain better control of the way their body works.

This thesis will be mainly focused on Neurofeedback — a relatively modern technique making use of the latest advances in EEG capturing, analysis and storage. The biofeedback software I was working on also includes a centralized server which could be used for comparison of results across many patients worldwide.

Neurofeedback can be used to treat a variety of diseases: from ADHD to insomnia. At the moment, the most researched application of Neurofeedback is for the treatment of ADHD. Meta-analysis by *Arns, M. et al.* has shown that Neurofeedback therapy produces clinically meaningful results and thus further research and development can be expected in the future. More information on the theory behind EEG, Neurofeedback and their application will be presented in the first half of the thesis.

My work was part of an effort by BioDat group to create an open-source software system which can be used for treatment of patients and storage of treatment results for further analysis. Two games were created in Java for the project: a racing game and an airplane shooting game. Detailed information on the planning and writing of the game software will be presented in the second half of the thesis.

The games are to be used by young patients being treated for ADHD who are expected to have a better experience throughout their treatment if the visual content was engaging enough, thus the importance of producing a wide variety of games for the patient to choose from. One of the biggest problems faced during the development process is the lack of conventional controls — the only input available to the software is the person's EEG data. Thus an effort was made to make the gameplay as interesting as possible within the limitations of the project's framework.

Chapter 2
Brain anatomy and the central nervous system

Higher brain functions such as decision making and actions are associated with an area of the brain called the cortex. The typical cortex (see fig. 2.1) is composed of four sections called lobes: the parietal lobe, the frontal lobe, the occipital lobe, and temporal lobe. The lobes' are associated with the following brain functions:

- Temporal lobe: memory, emotion, object recognition
- Frontal lobe: reasoning, planning, attention, working memory
- Parietal lobe: language comprehension, movement, orientation, perception of stimuli
- Occipital lobe: visual recognition, spatial analysis, visual attention

The cerebrum is divided into two symmetrical hemispheres, known as the left and right hemisphere. Individuals have a variable level of lateralization, the most evident effect of which is right-/left-handedness. The two hemispheres are connected by the *corpus callosum*.

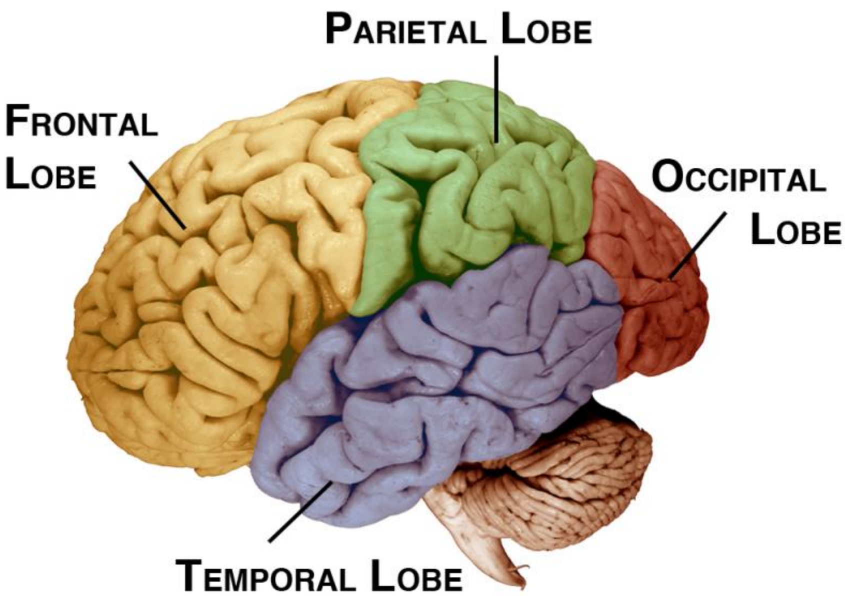

Fig 2.1: structure of cortex [11]

2.1 Neurons

The brain's most basic element is the neuron (see fig. 2.2). Neurons form a complex, interconnected system which expands across the brain and forms its structures. There are approximately 10^{10} neurons, each interconnected to other neurons through 10^4 synapses. A neuron is composed from: dendrites, soma and axon. The dendrites are the input elements of the neuron: once the inputs from the other neurons exceed a critical value, it sends out a weak electrical impulse through the axon. The axon is a long projection of the neuron, which splits into thousands of branches connected to other neurons. This forms the system used for transmitting signals throughout the brain.

The axon of the neuron is covered by a dielectric material called the *myelin sheath.* The myelin sheath helps signals propagate by helping prevent the electrical signal from leaving the axon.

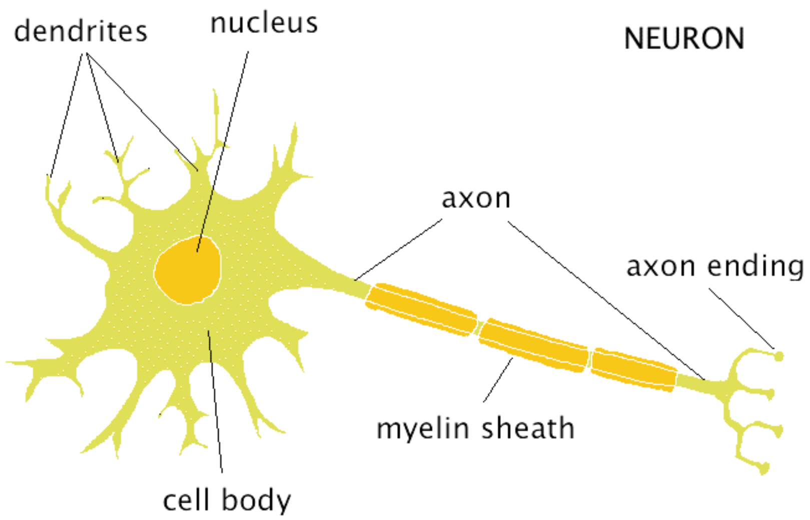

Fig 2.2: Neuron structure [12]

2.2 Brainwaves

Signals measured from the brain activity can be grouped according to their frequency and amplitude. They can be used to diagnose the activity of various regions of the brain as well as detect abnormalities in the brain functions. Normally we are unable to measure the activity of any single neuron; rather, we get the summary activities from thousands of neurons scattered under the sensors. Brainwaves are generally classified into four categories according to their bandwidths: alpha, beta, delta, and theta (see Figure 2.3).

Delta waves have the lowest frequency (0.1-4 Hz). In adults they manifest during deep sleep. In infants they are the dominant brain wave.

Theta waves (4-7 Hz) are exhibited while the person is in a state of drowsiness or sleep. They also occur as the result of an emotional response to stressful events.

Alpha waves (8-12 Hz) are most prominent in awake individuals who maintain their eyes closed and in a state of focus.

Beta waves (13-30 Hz) occur when the brain is actively thinking and processing information.

Waves with a higher frequency (30+ Hz) are called gamma waves.

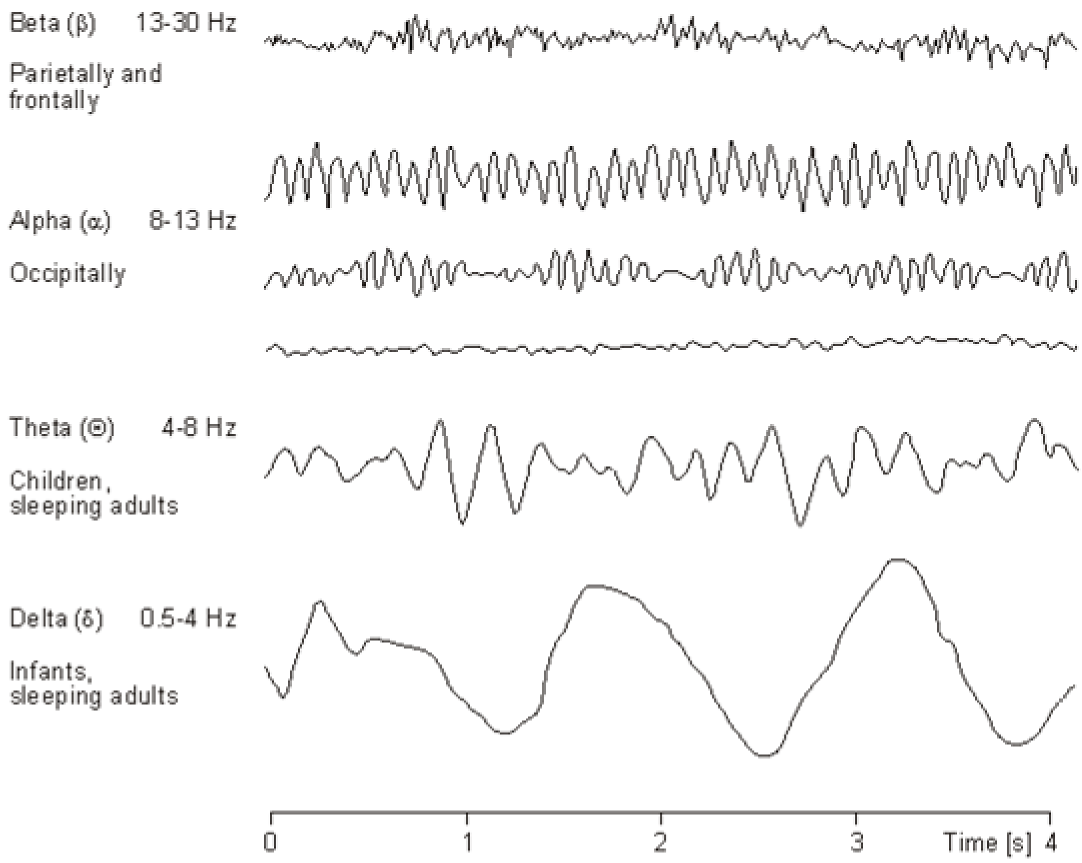

Fig 2.3: Brainwave frequencies [13]

Chapter 3

EEG

Electroencephalography (EEG) is the process of recording the electrical activity of the brain. [10] EEG is a non-invasive medical procedure which monitors the behavior of the central nervous system. It works by placing electrodes around the patient's scalp which then record the electric potentials of low amplitude, ranging from 10 μV to 100 μV. EEG is unable to record the activity of individual neurons as their signal is extremely weak. Instead, EEG detects the summed current of thousands of neurons beneath the electrodes.

3.1 History of EEG

The electrical activity of the brain was first recorded in the 19^{th} century by Richard Caton who has conducted experiments on rabbits and horses. He also noted that the signal strength is affected by lack of sleep, anesthesia and disappears upon death. [5]

Human EEG was first recorded in 1924 by German physiologist Hans Berger. He also invented the device used for recording of EEG activity, called the *electroencephalogram*. He was also the first to describe alpha and beta waves.

In the following years recordings of patients with brain tumors and epilepsy were produced thanks to improved instruments. The adoption of computer technology significantly accelerated the rate of adoption of EEG technology thanks to new devices allowing for the recording and the analysis of EEG data. Today it is commonly used in the medical field, along with other neuroimaging techniques such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT).

3.2 EEG Recording

Conventional EEG is a non-invasive technique which works by placing electrodes on the patient's head. It is thus differentiated from *Electrocorticography* (ECoG) which requires the placement of electrodes directly on the surface of the brain.

Electrodes are commonly placed on the head according to the "10/20" system (see fig. 3.1), first proposed by *H.H. Jasper* in 1958. The primary purpose of the 10/20 system is to provide a reproducible method for placing a relatively small number (typically 21) of EEG electrodes over different studies. [6] The name of the system comes from the fact that the distances between electrodes are measured in either 10% or

20% of the front-back or left-right radius of the scalp. Most studies use a special cap to simplify the application of EEG electrodes.

In the “10/20” system, electrodes are numbered with a letter and a number according to their location on the head: [7]

- **A** = Ear lobe
- **C** = Central
- **F** = Frontal
- **Fp** = Frontal polar
- **O** = Occipital

Odd numbers are used for the left hemisphere, even numbers are used for the right hemisphere.

Bipolar or unipolar electrodes are used for measurements. Bipolar measurements are connected sequentially through two closely located electrodes, which allows for greater accuracy. Unipolar electrodes are connected through a referential electrode (which remains inactive).

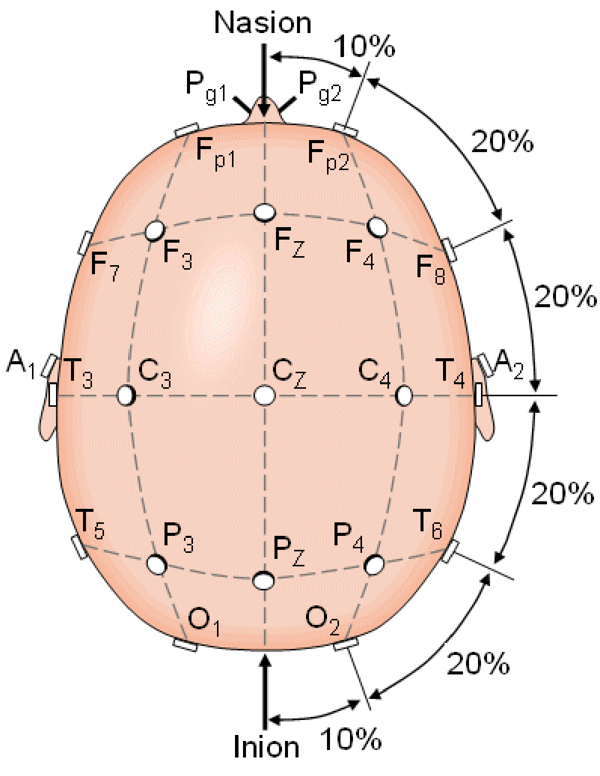

Fig. 3.1: “10/20” electrode placement [14]

Chapter 4
Biofeedback

Biofeedback is a technique which measures the bodily functions and enables the individual to change their psychological activity in order to improve their health and performance. It includes the measurement of blood pressure, brain waves (EEG), heart rate, muscle tension, skin conductance, etc. The measured signal undergoes processing and analysis — either by a health professional or a computer, and then presented to the individual in a graphical or otherwise informative form.

Biofeedback first evolved in the 1960s and 70s to become a medically accepted method of treatment of a variety of diseases such as urinary incontinence, migraines, high blood pressure and many other conditions. Computerized biofeedback devices open new horizons for biofeedback as they allow for easy and portable measurement of bodily indicators, as well as more advanced signal processing techniques.

4.1 EEG Biofeedback

EEG biofeedback (also known as Neurofeedback) is a type of biofeedback where the individual's EEG signal is being measured [1]. The individual can then use the provided feedback in order to adjust their brain activity and improve performance of the chosen brain waves. This allows the patient to acquire better control of their concentration and activity patterns, which can be useful for the treatment of conditions such as ADHD.

A variety of methods can be used to inform the user of the state of their brain waves: audio stimuli, visual stimuli, interactive games, animations and film. Despite the ease of computerizing the Neurofeedback process, it is important for a trained professional to be present during the sessions. While no harm has been shown to elicit from self-training utilizing EEG biofeedback, it is nevertheless crucial for a third-party to be present during the training for maximum effectiveness.

Medical applications of Neurofeedback can be divided into three categories:

- Childhood problems: nocturnal enuresis, insomnia, ADHD. Millions of children are diagnosed with ADHD each year and there are doubts about the safety of medicaments use in young children. Neurofeedback can serve as an alternative in that age group [8]
- Personality disorders: substance abuse, depression, anxiety [4]
- Degenerative disorders: Alzheimer's, dementia, fibromyalgia, epilepsy [2] [3]

- Cognitive performance: enhancement of attention spans and self-control in otherwise healthy individuals [4]

4.2 History of EEG Biofeedback

EEG Biofeedback dates back to the 1960's when researchers J. Kamiya and B. Sterman began to explore phenomena associated with EEG. J. Kamiya from the University of Chicago was attempting to elicit "alpha waves" in patients by providing them with feedback on their own EEG performance. He has found the process to be relaxing for his patients. This is considered the first Neurofeedback training in history.

At the same time B. Sterman from the University of California has been researching brain activity throughout sleep in cats. He has found that cats enter a more relaxed state when their brain waves correspond to a certain level. After adding biofeedback in the form of treats while the cats were in a relaxed state, he has found them to become more relaxed in general.

B. Sterman has later been assigned with the task of minimizing the psychological effects of astronauts exposed to a certain rocket fuel. The effects included hallucinations, nausea, severe epileptic seizures, and eventually death. Using the cats from the previous biofeedback experiment he has found them to be more resistant to seizures than normal cats. This was the first study demonstrating the efficacy of Neurofeedback in actual patients.

B. Sterman's experiments have been the been the basis of the work by J.F. Lubar on the use of EEG biofeedback for the treatment of ADHD in children during the 1980's. ADHD treatments remain the most highly recognized application of EEG biofeedback.

4.3 Devices used for EEG Biofeedback

EEG devices can have a range of electrodes — the standard configuration includes 19 or 21 electrodes, however many more can be used depending on the requirements of the experiment. The application of the electrodes is an unpleasant procedure which requires the application of either gel or sea water to the subject's scalp, necessary to increase the conductivity of the skin. Each electrode is then connected to an amplifier and eventually routed to a computer. These kits (see fig. 4.1) can cost upwards of thousands of dollars, require significant space for storage and application, and often take considerable time to apply and remove.

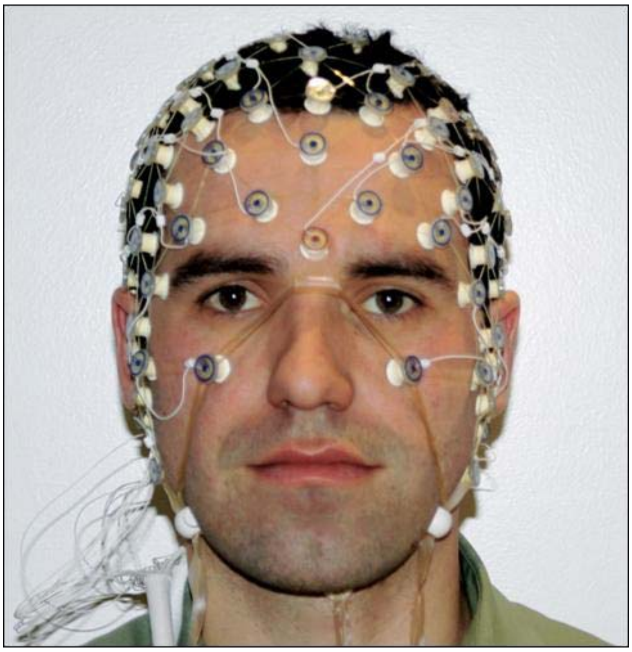

Fig 4.1: EEG cap [15]

Much of the previous EEG research has been focused on persons with heavy disabilities, such as those in a hospital setting, but the technology continues to become more streamlined, facilitating adoption among the potential user-base of therapists, physicians, psychologists and other interested parties. More recently a range of inexpensive devices have become available, originally targeted at the videogames market as a novelty I/O device. As of 2014, headsets are available from an Australian company called Emotiv, containing 14 sensors, which is comparable to traditional EEG caps. Another alternative comes from American company NeuroSky – with just one sensor.

4.3.1 Emotiv EPOC Neuroheadset

The EPOC was originally marketed as a gaming device. The Neuroheadset is inexpensive compared to most medical EEG caps, priced at 299$ for retail consumers. It has 14 sensors with accordance to the international 10-20 System. The device has an internal sampling rate of 2048Hz and the external output of EEG data is at 128 Hz. The processed EEG signal is transferred using a wireless USB dongle. It is much faster to setup when compared to medical EEG caps and requires a saline solution instead of a special gel, which is much easier to clean up and more pleasant for the patient. [9]

The Emotiv EPOC headset has been used as the base device for the BioDat group's project and was the reference device through which testing was carried out.

Chapter 5

Project Analysis

The assignment required the implementation of two visually distinct games which would accompany the existing six games previously developed by FEL CTU student Tomas Pojkar. The games were to be used in the "Hub" project currently developed by the BioDat team (see fig. 5.1)

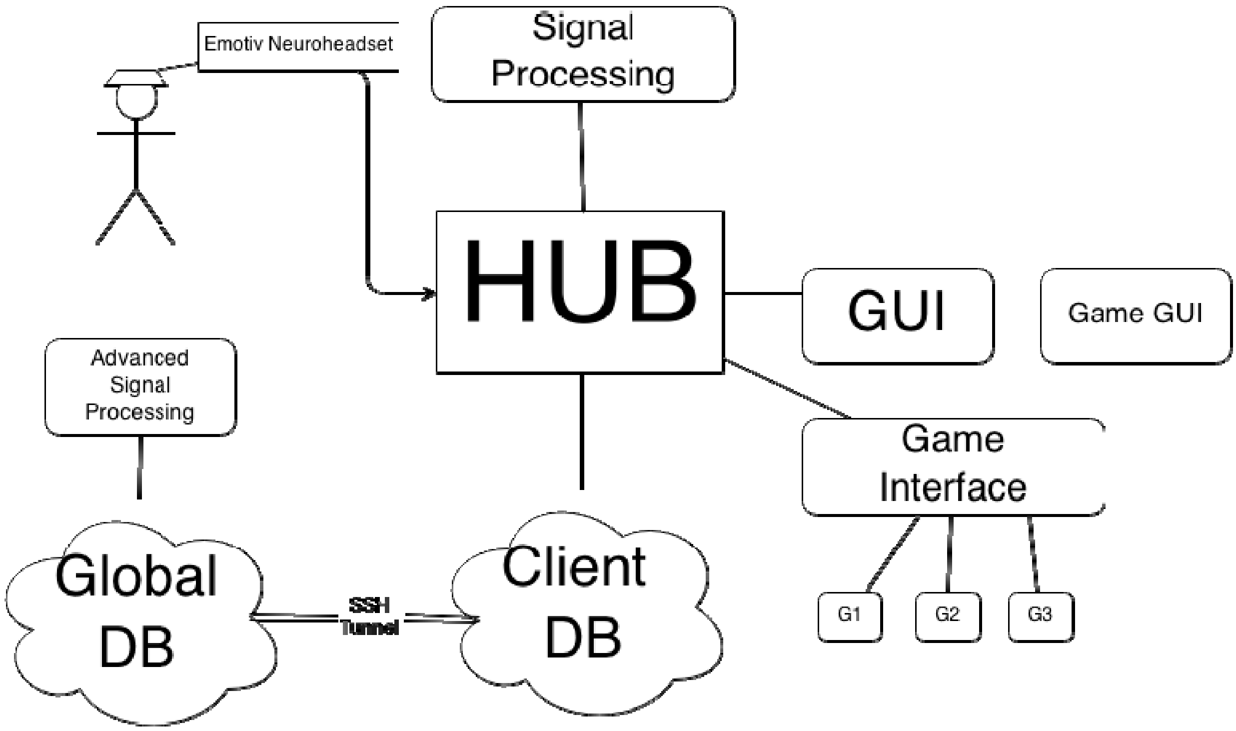

Fig 5.1: Global project structure

5.1 Project environment

Since the assignment mandated that the implemented games are to be a part of the previously developed project, it was decided that it would be unwise to steer from the existing project specifications. Thus, both games were implemented in Java using the Netbeans development environment. This makes further interoperability, testing and revision easier than if conflicting standards were chosen, as well as making the project seamlessly integrated.

As both games were to be 2-dimensional, Java 2D API was chosen as the graphical library. While better alternatives exist for Java projects, such as LWJGL (Lightweight Java Game Library) or JOGL (Java OpenGL), it was mandated that Java 2D is chosen since all of the existing games are implemented using either the Java 2D or the Java 3D API. The biggest benefit of the configuration is complete project portability, as both Java, Java 2D and Java 3D are cross-platform libraries.

5.1.1 Previously implemented games

The following games were previously implemented (see fig. 5.2):

- Bacterium war: players control a set of bacteria fighting against computer AI
- Airplane flight: players try to collect bonus points from the air
- Balloon flight: balloon alternative to the previous game
- Scene improvement: an initially black-and-white image turns colorful as the player's brain waves improve
- Pyramid puzzle: a puzzle resembling a pyramid is being assembled (or disassembled) depending on how well the player is performing
- Glass puzzle: same as above, but using a glass model
- Target shooter: player tries to hit the given target as close to the center as possible
- Can shooter: player tries to hit a set of cans
- Wall builder: a wall is being assembled, with the speed depending on the player's brain waves.

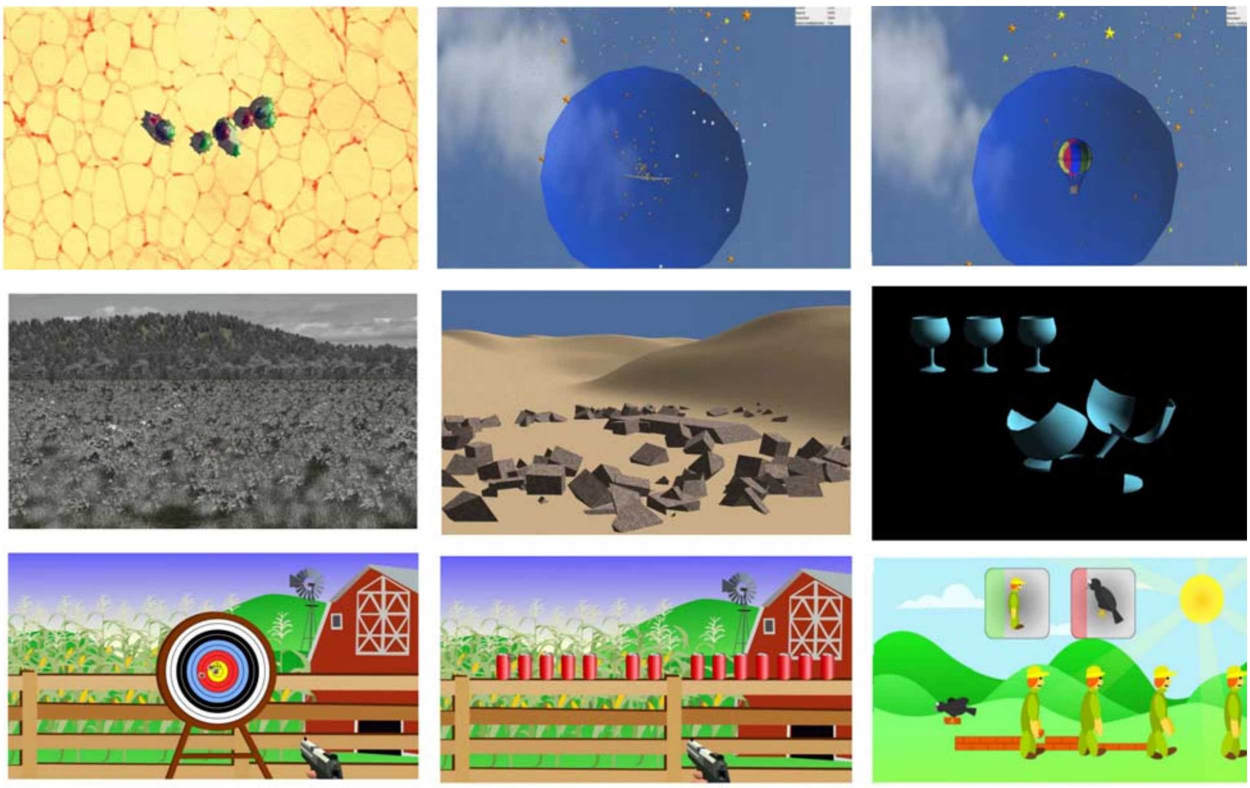

Fig 5.2: Previously implemented games: bacterium war, airplane flight, balloon flight, scene improvement, pyramid puzzle, glass puzzle, target shooter, can shooter, wall builder [16]

It was important to maintain originality, thus an attempt was made to make sure that the paradigm behind the newly implemented games differed significantly from what was previously implemented.

5.2 Gameplay requirements

Besides the need to maintain originality, it was important to make sure that the chosen game format was suitable for the patients to enjoy. The crucial game parameters are described in table 5.1.

Game genre	Example	Controls involved	Tolerance for random input	Suitable for small children
Artillery game	Worms	Arrow keys, weapon keys	Low	No
Ball and paddle	Arkanoid	Left-right arrow keys	Low	Yes
Brawler	Street fighter	Arrow keys, fight keys	High	No
Construction simulator	Sim City	Mouse	Low	Yes
First-person shooter	Doom	Arrow keys, weapon keys	Low	No
Maze game	Pac-Man	Arrow keys	Medium	Yes
Pinball	Microsoft Pinball	Left-right arrow keys	Medium	Yes
Platform game	Super Mario	Arrow keys	Low	Yes
Point-and-click adventure	Myst	Mouse	Low	Yes
Real-time strategy	Warcraft	Mouse	Low	Yes
Role-playing game	Final Fantasy	Mouse	Low	No
Top-down shooter	Grand Theft Auto	Arrow keys, weapon keys	Low	No

Table 5.1: Game genres and their suitability for EEG biofeedback

Since not all game genres are appropriate for Neurofeedback and some game genres were already implemented, it was decided to focus on vehicle simulators: an airplane simulator and a car race simulator. Both games are easily understood by children of all ages, controlled by a limited set of keys and produce instant feedback on the player's overall performance.

5.2.1 Gameplay controls

The games are controlled (see fig. 5.3) by at maximum of two user inputs at a time: alpha waves (demonstrating the user's level of relaxation) and beta waves (demonstrating the user's level of concentration). Both inputs are subjects to random variations due to inaccuracies in the measuring software as well as a lack of complete of control over the inputs on behalf of the users.

Fig 5.3: User participating in a Neurofeedback therapy session [17]

It is important for the gameplay controls to have a profoundly visible effect on the game session as the person must obtain immediate feedback on the state of his brain. At the same time, the controls should have a familiar effect to that commonly found in other games of the genre so that the person will have an understanding of what they are trying to control. For example, in a shooter game alpha waves could control the rate of fire and the beta waves would control the rate of movement.

5.2.2 Gameplay feedback

Biofeedback requires a clearly set model of feedback being available to the user at all times which allows them to estimate their own estimate. Feedback in games can be presented in the following forms:

- Visual stimuli: progress bars, achievements, level progress, color indicators, game points, change of brightness
- Audio stimuli: change of sound within the game depending on the user's success, warnings about unacceptable brain wave status
- Vibration stimuli: feedback through an accessory such as a steering wheel

The developed games made extensive use of visual and audio stimuli for a constant feedback loop.

Chapter 6

Project implementation

The game implementation works on top of the existing project structure (see fig. 5.1) and are executed in the following way (see fig. 6.1):

1) The therapist sets the duration and complexity of the game
2) The therapist launches the games
3) The game is constantly updated with the user's EEG channel values and visualizes them within the gaming process.

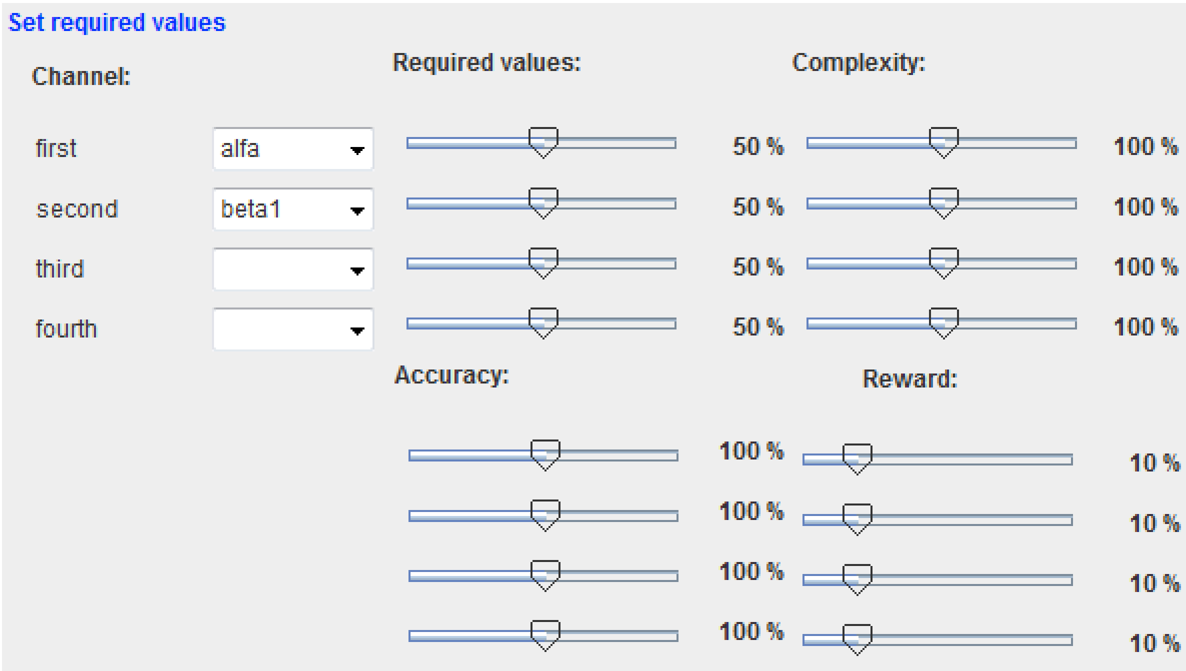

Fig 6.1: Game setup screen example

The software supports control from up to 4 distinct channels, however only 2 are used in the implemented games. The rest of the therapy setting values have the following meaning in the game:

- **Required values:** the ideal the therapist would like to see in the patient (R)
- **Complexity:** tolerance for user mistakes (C)
- **Accuracy:** interval in which the user's input is considered acceptable (A)
- **Reward:** level of positive reinforcement for maintaining the necessary brain waves level

The maximum and minimum acceptable brain wave values (B) is thus calculated according to the following formula:

$$B = R \pm C \times A \div 100$$

6.1 Car Race Game

The Car Race game is a classical top-down racing simulator. It presents the user with a circle track and 4 opponents controlled by the computer (see fig. 6.2).

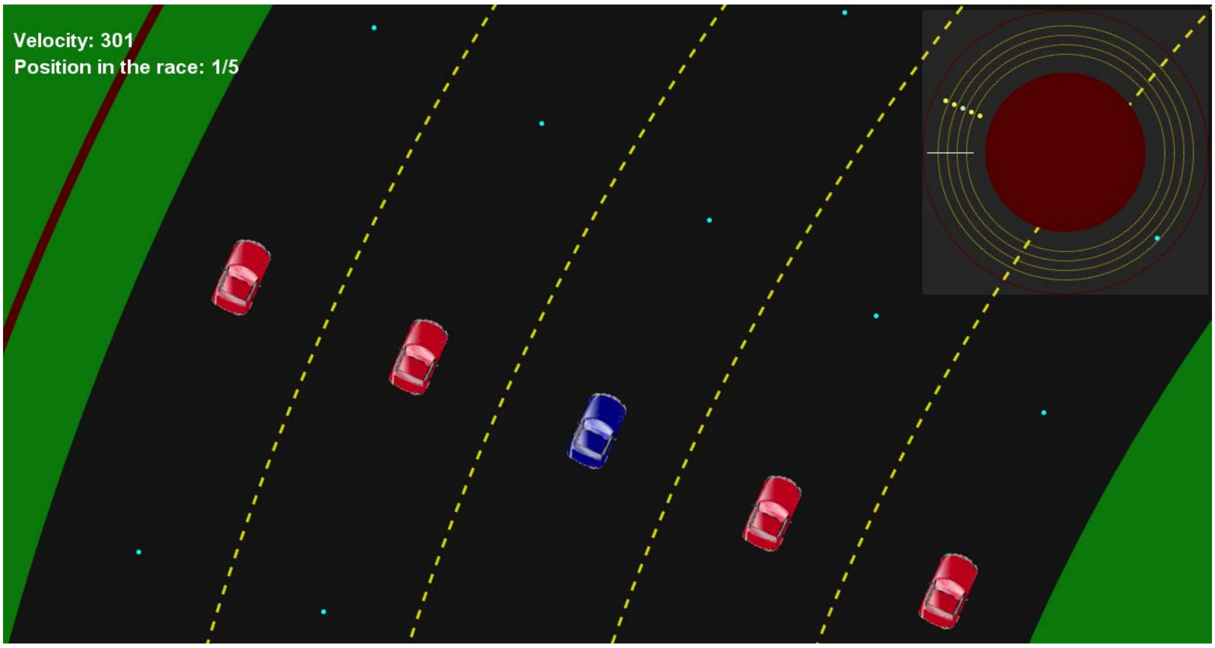

Fig 6.2: Car Race game interface

The race track (see fig. 6.3) is dynamically generated based on the computer's screen size. The computer's cars drive in a fully automated fashion, simulating external players. The race track contains distinct roads (marked with yellow) and waypoints (marked with blue), according to which both the player and the computer cars navigate the game. Entering another player's road or the grass causes a speed penalty.

The main game screen is initiated with the following code:

```
public CarRace(boolean run) {
        this.setTitle("Car race");

        _board = new Board(run);
        getContentPane().add(_board, BorderLayout.CENTER);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pack();
        setExtendedState(JFrame.MAXIMIZED_BOTH);
        setVisible(true);
        _board.setSize(this.getWidth(), this.getHeight());
        });
}
```

6.1.1 Game controls and penalization

The game takes the following input from the player:

- **Alpha waves** control the player car's speed
- **Beta waves** control the accuracy of the player's driving

Both status indicators are easy for the player to see when observing the main player's car (see fig. 6.3 and fig. 6.4). Computer-controlled cars are also subject to random steering and speed problems, to make the game more challenging for the player. Since the nature of EEG data and performance is not under direct control of the player, the game rapidly returns the player to a winning position (see fig. 6.5) once the brain wave indicators return to those prescribed by the therapist in the main program window.

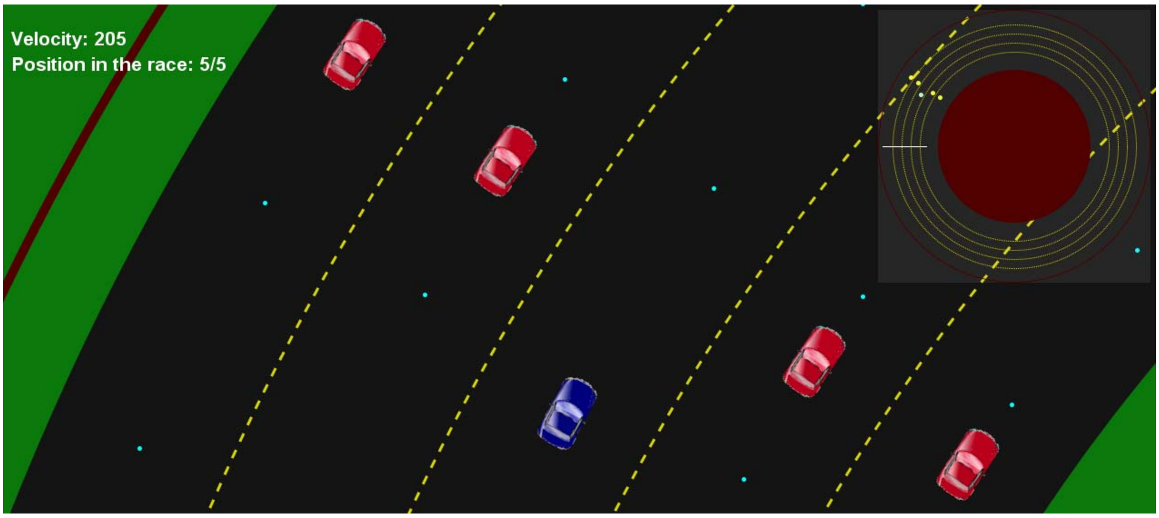

Fig. 6.3: Situation where player's alpha wave is below normal (speed problems)

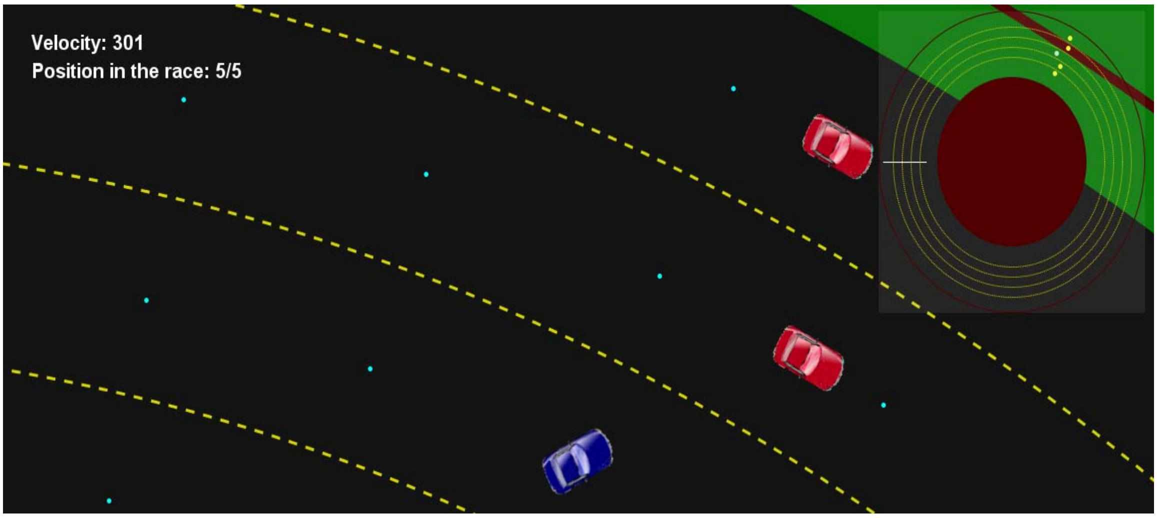

Fig. 6.4: Situation where player's beta wave is below normal (accuracy problems)

The player is also given audio indication of his level of success – a car sound is being constantly played out in the background, with the alpha and beta activity regulating its amplitude and pitch. The player is also given an indication of their speed and race.

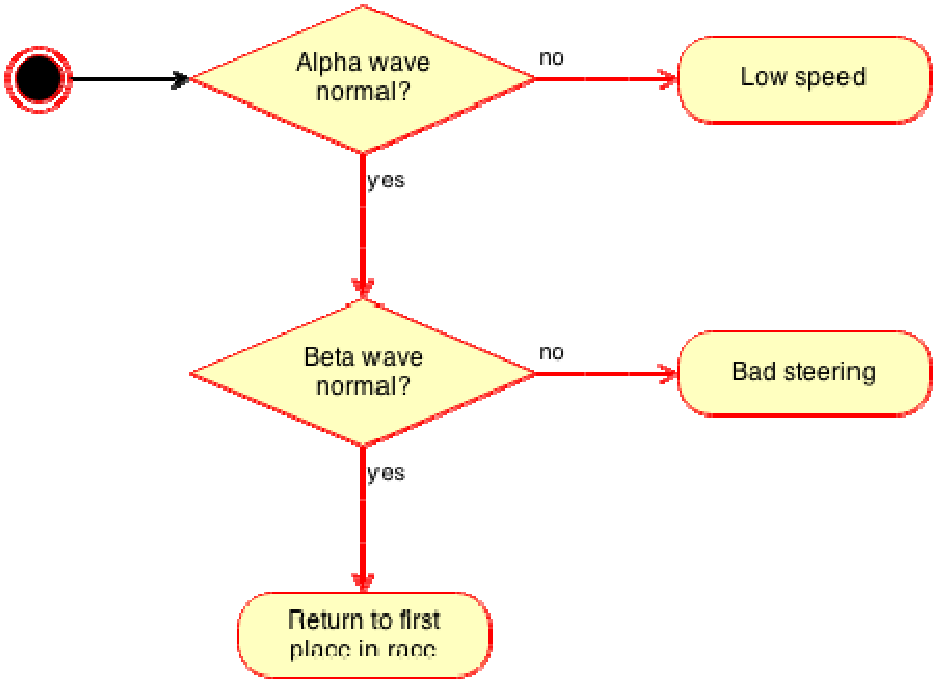

Fig. 6.5: State diagram of race conditions

The biofeedback conditions loop is implemented with the following code:

```
// Velocity deviation
float velocityDeviationOld = _velocityDeviation;
_velocityDeviation = Math.max(0, Math.min(1, velocityDeviation));

if (_velocityDeviation > 0) {
        _constantVelocity = _originalVelocity * (1 - Math.min(1, _velocityDeviation));
} else {
        _constantVelocity = _originalVelocity;
}
_velocity = _constantVelocity;
// Angle deviation
float angleDeviationOld = _angleDeviation;
_angleDeviation = Math.max(0, Math.min(1, angleDeviation));
// Recover to first place
_recoverLosses = (((velocityDeviationOld > 0) || (angleDeviationOld > 0)) &&
(_velocityDeviation <= 0) && (_angleDeviation <= 0));
_finishedRecovery = false;
```

6.1.2 Graphics Implementation

The game makes extensive use of the Java2D API capabilities. Since loading the entire map into the video memory would be an extremely encumbering operation (total memory requirements would be over 1GB of video memory), it repeatedly calculates the visible map area (see fig. 6.6) and projects only the areas of the map which are visible to the user.

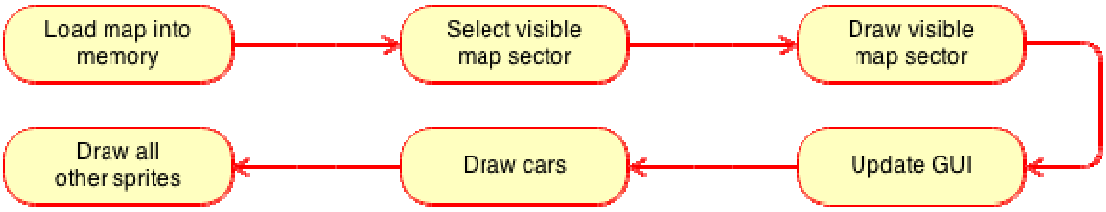

Fig 6.6: Graphics generation diagram

The main graphics loop is implemented with the following code:

```
private void drawXY(Graphics2D g, int carNumber, int x, int width) {
        g.setPaint(_game.getBackColor());
        g.fillRect(x, 0, width, Controller.instance.getHeight());
        <...>
        BufferedImage mapImage = _game.getMapImage();
        <...>
        // Draw map according to car rotation angle
        _affineTransform.rotate(_game.getCarDirection(carNumber),
                posxf + Car.getWidth() / 2,
                posyf + Car.getHeight());
        g.drawImage(mapImage.getSubimage(imx, imy, width, Controller.instance.getHeight()),
        x, 0, null);
        // Draw cars
        AffineTransform af;
        for (int i = 0; i < _game.getNumberOfCars(); i++) {
            <...>
                g.drawImage(_game.getCarImage(i), af, null);
        }
        // Draw mini map
        BufferedImage mapMiniImage = _game.getMapMiniImage();
        <...>
        g.setPaint (new Color(255, 255, 255, 20));
        g.fillRect (mapMiniImageX, mapMiniImageY, mapMiniImageWidth,
        mapMiniImageHeight);
```

```
        g.drawImage(mapMiniImage, mapMiniImageX, mapMiniImageY,   mapMiniImageWidth,
        mapMiniImageHeight, null);

        for (int i = 0; i < _game.getNumberOfCars(); i++) {
                <...>
                g.fillOval(carCircleX, carCircleY, 5, 5);
        }
}
```

Since the user is not guaranteed to see all of the opponents' cars at the same time due to varying car speeds, it is necessary to indicate the player's position on the global map. This is achieved through the use of a mini-map and player position indicator (see fig. 6.7).

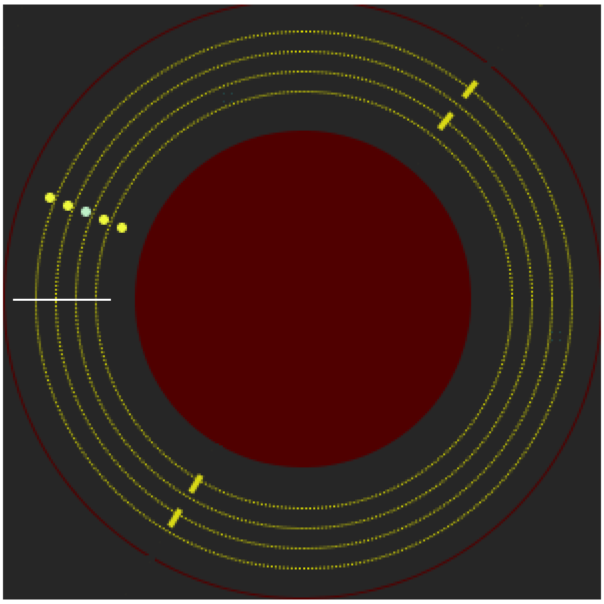

Fig. 6.7: Mini-map with player indicators

Java's internal garbage collector doesn't manage the memory resources well enough, so we have to manually request the GC() function once per second. Overall the performance level is acceptable and suitable for old PCs (see chapter on Testing).

6.1.3 Race track generation

Initially I tested out generation of a so-called infinite race track (see fig. 6.8). The infinite race track works by projecting a fixed map projection on the screen, with additional sprites symbolizing game bonuses, opponents, pits, and so on. Unfortunately this approach was unacceptable for a racing game since it appeared relatively boring from the biofeedback user's perspective.

Instead, I've decided to generate an actual 2D map of the race track (see fig. 6.9). This approach brings additional flexibility as it allows for the generation of different map sizes for different screen sizes and potentially allows for complicated race tracks to be created. For this project, I've opted to generate a fixed circle track with 5 roads: one for the player's car and four for his opponents.

Fig. 6.8: Example of an infinite race track [18]

The generator potentially allows for an unlimited number of car racetracks to be generated and the curve generator can be used to create shapes more complicated than a circle. If the project ever becomes a multiplayer one, it can potentially be used to create smaller multiplayer tracks for human players.

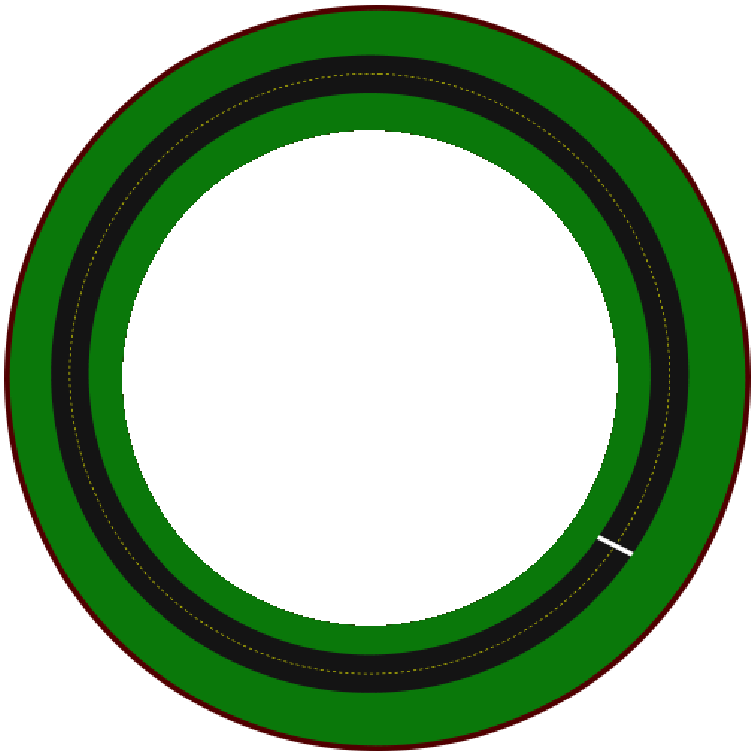

Fig. 6.9: Example of a race track generated by the generator

The main race track generation code is as follows:

```
//Tracks
g2.setPaint(new Color(20, 20, 20));
g2.setStroke(new BasicStroke(TRACKS_WIDTH));
g2.drawOval(FINAL_TRACKS_X, FINAL_TRACKS_Y, img.getWidth() -
(FINAL_TRACKS_X * 2), img.getHeight() - (FINAL_TRACKS_Y * 2));

//Tracks borders
g2.setPaint(new Color(214, 214, 3));
float dash1[] = { mini ? 1.0f : 10.0f };
BasicStroke dashed = new BasicStroke(FINAL_TRACK_BORDER_WIDTH,
BasicStroke.CAP_BUTT, BasicStroke.JOIN_MITER, 10.0f, dash1, 0.0f);
g2.setStroke(dashed);
<...>
//Draw waypoints
g2.setPaint(new Color(0, 255, 255));
int[][][] wayPoints = getWayPoints();
for (int t = 0; t < wayPoints.length; t++) {
        for (int p = 0; p < wayPoints[t].length; p++) {
                g2.fillOval(wayPoints[t][p][0], wayPoints[t][p][1], 5, 5);
        }
}
```

6.1.4 Player behavior implementation

Practically speaking the code controls both the player and the computer car navigation since the player is unable to provide any meaningful input beyond a fixed variable showing his current brain wave indicators. While many potential technologies exist for complete control over computer software through the use of EEG and other biological sensors, it would be impractical to attempt to use them during biofeedback therapy since the patient is supposed to concentrate on his own condition rather than attempt to steer whatever is shown on the screen.

The car navigation is implemented by placing waypoints throughout the map (visible as blue dots) and then calculating the angle to which the car must rotate to get to the next destination waypoint. The following formula is used to calculate the car rotation angle (where the *yd* is the y-axis distance and *xd* is the x-axis distance):

$$Angle = a\tan(yd, xd) \bmod 360$$

The game uses Cartesian coordinates throughout the application which makes it easy to use the angle from the formula to turn the car until the right angle is reached.

6.2 Air Shooter Game

The second implemented game was a fighter plane simulation (see fig. 6.10). It presents the user with a desert-like environment where the player is in control of an airplane. The airplane's goal is to shoot down his enemies, as well as avoid being shot by his opponents.

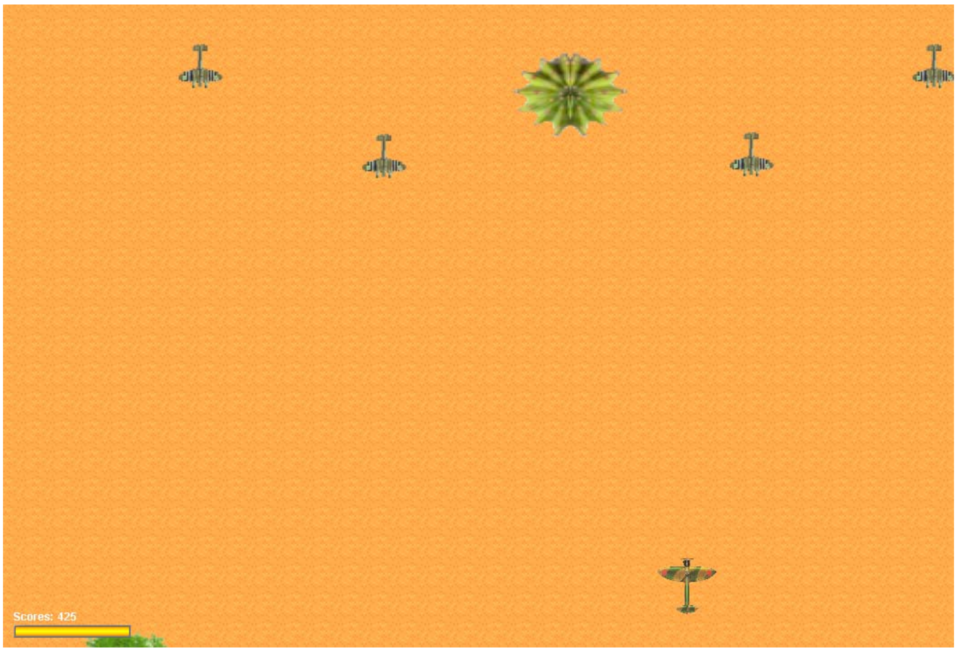

Fig. 6.10: Air Shooter Game interface

The player's airplane meets a variety of incoming enemies which shoot a variety of different projectiles. The airplane has a speed indicator as well as points indicator. Points are accumulated through shooting down the enemy's airplanes as well as lost when the airplane is destroyed. The player has an infinite number of lives — the only time limitation is set by the therapist. The main game window is initialized with the following code:

```
_board = new Board();
_board.init(auto);
addWindowListener(new WindowAdapter() {});
getContentPane().add(BorderLayout.CENTER, _board);
pack();
setVisible(true);
setExtendedState(JFrame.MAXIMIZED_BOTH);
_board.setSize(this.getWidth(), this.getHeight());
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
if (run) {
        _board.start();
}
```

6.2.1 Game controls and penalization

The game takes the following input from the player:

- **Alpha waves** control the airplane's speed
- **Beta waves** control the number of bullets shot by the enemy airplanes

When the player's brain waves are within the limits prescribed by the therapist, the game lets the player's plane shoot down the enemies without problems, therefore the player receives visual biofeedback on the state of his brain waves through his success in the game (see fig. 6.11). The game also shows two other indicators of the player's success: a health indicator and a points indicator (see fig. 6.12), as well as provides an audio feedback in the form of a background airplane engine's noise, the type of which depends on the airplane health, similar to other games in the airplane genre.

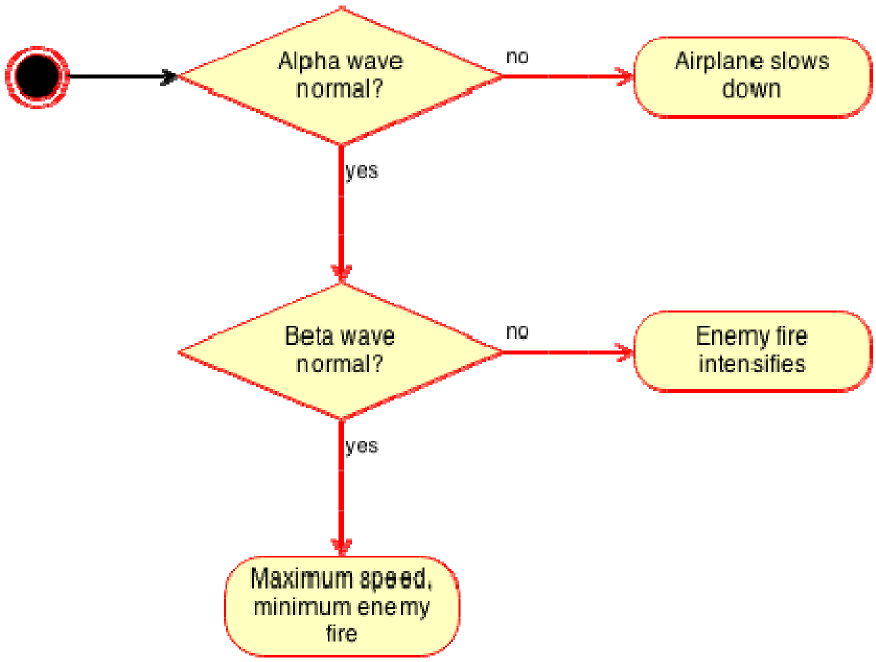

Fig. 6.11: State diagram of flying conditions

The state diagram above is implemented as following inside the software:

```
public void gameDataChanged() {
        //patient EEG data updated
        super.gameDataChanged();
        float planeSpeedDeviation = getDeviation(0); //alpha wave
        Controller.Instance().setAutoPlaneSpeedDeviation
        (planeSpeedDeviation);
        float enemyBulletsDeviation = getDeviation(1); //beta wave
        Controller.Instance().setAutoEnemyBulletsDeviation
        (enemyBulletsDeviation);
}
```

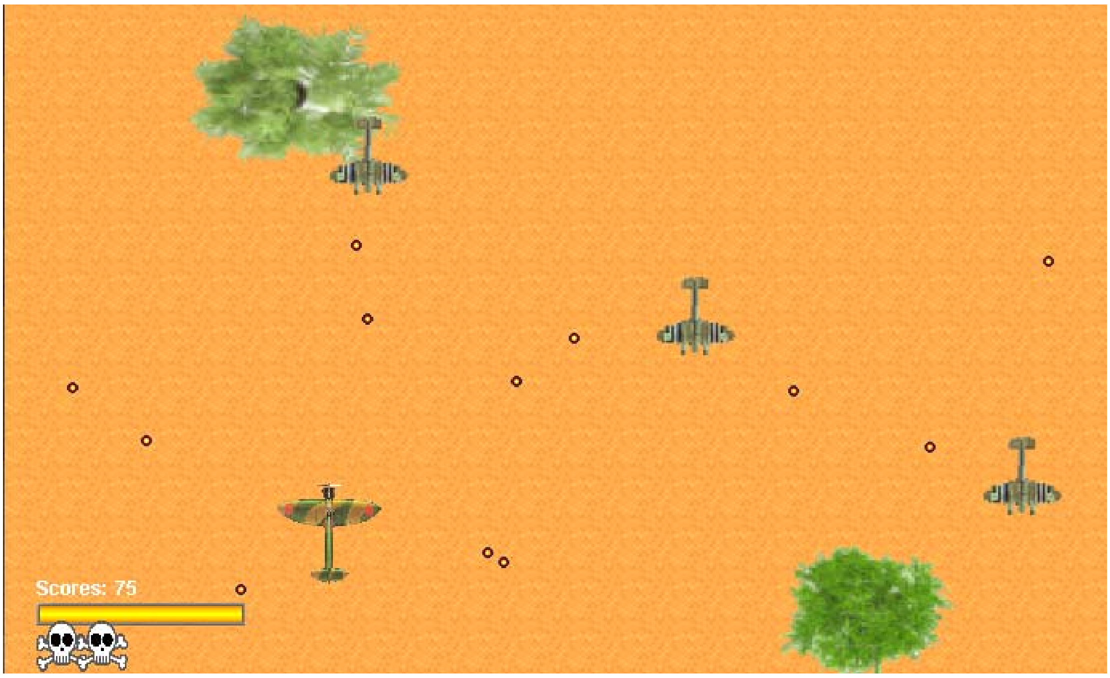

Fig. 6.12: Maximum enemy fire, with health, death and score indicators visible

6.2.2 Graphics Implementation

Unlike the previous car race game, the airplane shooter makes use of an infinite game field generation routine (see fig. 6.13). This works better because the game is based on the shooting aspect of the game rather than the map navigation and racing aspect.

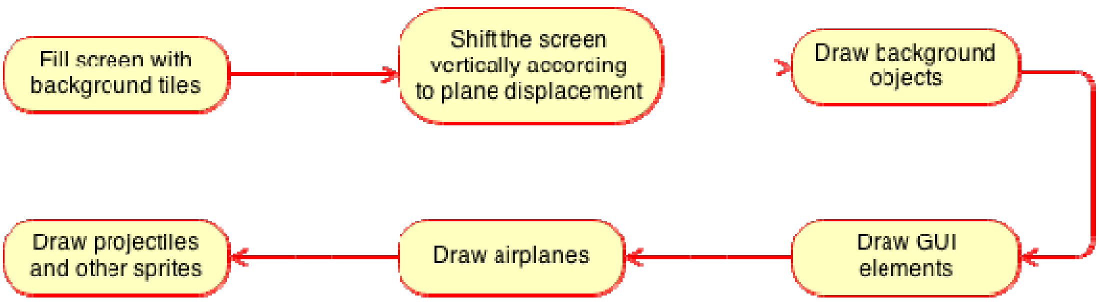

Fig. 6.13: Graphics generation cycle

This approach has the advantage of being faster to draw, since we don't have to load the entire map in the screen and can instead fill the screen with small tiles. Another difference is that instead of using fixed objects on the map, the aircraft game draws fixed background objects (such as bushes or trees) as sprites as well. It is possible to add different background sprites in the future to make the game look visually different each time it is run, therefore making it potentially more enjoyable to frequent players.

The graphics are being drawn on the screen with the following code:

```
//make sure the game is running full-screen
c.setWidth(w);
c.setHeight(h);
_speed = (int) (1 * c.getHeightCoefficient());
initSprites(w, h);
//calculate the plane speed based on the screen size
final int planeSpeed = (int) ((_auto ? 2 : 5) * c.getWidthCoefficient());
//draw
drawBackGroundWithTileImage(w, h, g2);
//draw Sprite number 1
_sprite1.update(w, h);
_sprite1.draw(g2, this);
//draw Sprite number 2
_sprite2.update(w, h);
_sprite2.draw(g2, this);
//draw Sprite number 3
_sprite3.update(w, h);
_sprite3.draw(g2, this);
//draw everything else on the screen
c.updateAndDraw(w, h, g2, this);
//draw plane
c.updatePlane(planeSpeed);
c.drawPlane(g2, this);
```

6.2.3 Player behavior implementation

Similar to the car race game, it is necessary to ensure that all the airplanes on the map are moving autonomously. The implementation was fairly straightforward since the game track is essentially an endless vertical map. The enemy planes are moving either side-to-side or top-down. The player airplane is only moving on the x-axis, constantly moving up at a fixed speed. There are several algorithms in place:

1) Enemy crash avoidance algorithm. Since the plane loses health from hitting enemy airplanes, we should attempt to avoid such contact.

```
private void preventAutoCollision() {
    if (Controller.Instance().hasAutoCollisionWithEnemy(_x, _y, _width, _height, _speed,
    _left, _right)) {
            _left  = false;
            _right = false;
            _autoDestinationEnemy = null;
    }
}
```

2) Enemy seeking algorithm. We want to make sure that our airplane fires into the enemy planes, rather than firing in empty space. To do that we must align our x-axis with the closest enemy plane.

```
private void updateAutoDestinationEnemy(int w) {
    <...>
    //get the closes enemy based on the player's coordinates
    if (_autoDestinationEnemy == null) {
            _autoDestinationEnemy = Controller.Instance().
            findAutoDestinationEnemy(_x, _y, _width, 10);
    }
    //reset the movement direction
    _left  = false;
    _right = false;
    //choose appropriate direction
    if (_x + (_width / 2) - 1 < _autoDestinationEnemy.getX()) {
            _right = true;
    }
    else if (_x + (_width / 2) - 1 > _autoDestinationEnemy.getRightX()) {
            _left  = true;
    }
}
```

3) Enemy shooting algorithm. Some of the enemy planes attempt to shoot directly at the player rather than shooting straight ahead.

```
//calculate distance to the plane location
int xDistance = c.getPlaneX() - _x;
int yDistance = c.getPlaneY() - _y;
//calculate the total Cartesian distance
double totalDistance = Math.sqrt(Math.pow(xDistance, 2) + Math.pow(yDistance, 2));
//calculate the bullet speed and direction
int speed = (int) Math.rint(3 * c.getWidthCoefficient());
_speedX = (int) (speed * (xDistance / totalDistance));
;       _speedY = (int) (speed * (yDistance / totalDistance));
```

Overall the game provides a smooth and consistent gaming experience, which would hopefully be engaging enough to the participating patients.

Chapter 7
Testing

Testing was carried out using the Emotiv EPOC Neuroheadset (see fig. 7.1), using recorded data from patients, as well as using a manual brain wave value simulator.

Fig 7.1: Emotiv EPOC Neuroheadset [19]

7.1 User testing

Two subjects carried out tests (see fig. 7.2) with actual physical Neuroheadsets. The games worked as expected and were rated positively by the participants. Unfortunately I was unable to carry out testing on patients suffering with ADHD, therefore both individuals were healthy adults. Clinical testing will be carried out in the future by the BioDat team.

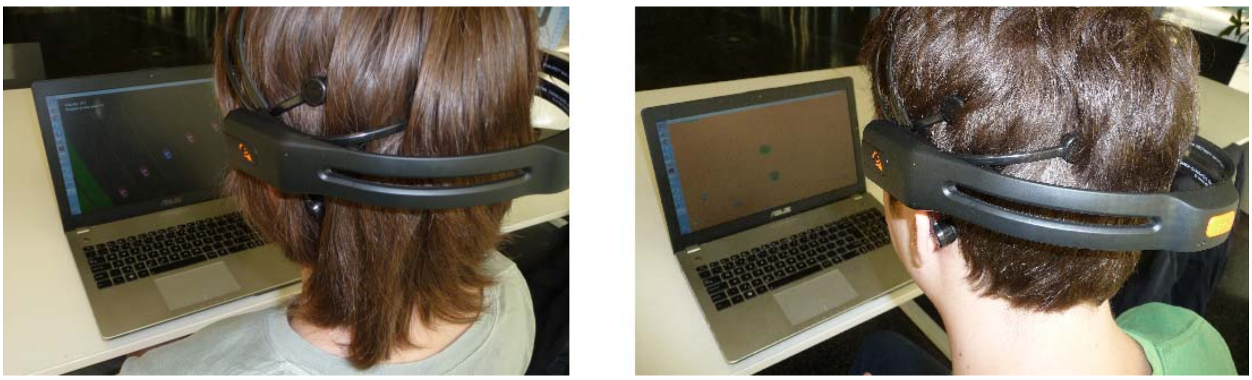

Fig 7.2: User testing

7.2 Hardware testing

Testing was carried out with the following machines:

ID	PC type	Model	OS	CPU	RAM	GPU
PC1	Laptop	ASUS N56VM	Windows 7	Intel Core i7	6 GB	Nvidia GT 630M
PC2	Laptop	ASUS UX31E	Windows 7	Intel Core i7	4 GB	Intel HD Graphics 3000
PC3	Laptop	Lenovo X220	Ubuntu 14	Intel Core i5	4 GB	Intel HD Graphics 3000
PC4	Desktop	N/A	Windows 8	Intel Core i7	8 GB	Integrated

Table 7.1: Testing hardware

The game was fully functional on all 4 PCs. The main performance indicators were then recorded:

- Rendering speed (in Frames Per Second – minimum, maximum, average)
- Average CPU load
- Average memory consumption

The testing results are presented for Car Race Game (see table 7.2) and Airplane Shooter Game (see table 7.3).

ID	FPS (MIN)	FPS (MAX)	FPS (AVG)	CPU (AVG)	RAM (AVG)
1	31	50	35	16%	200 MB
2	19	44	32	19%	225 MB
3	10	46	27	25%	340 MB
4	35	51	38	12%	220 MB

Table 7.2: Testing results for Car Race Game

ID	FPS (MIN)	FPS (MAX)	FPS (AVG)	CPU (AVG)	RAM (AVG)
1	44	59	40	8%	205 MB
2	30	50	38	10%	250 MB
3	15	44	30	16%	240 MB
4	48	65	45	7%	220 MB

Table 7.3: Testing results for Airplane Shooter Game

Chapter 8
Conclusions

The main project goals were implemented successfully and are fully functional. The project will hopefully prove useful to real-world patients through future work by BioDat team.

8.1 Thesis results and timeline

The following tasks and assignments were successfully completed throughout the duration of the thesis project:

- I successfully familiarized myself with previous work by BioDat team
- Errors and project roadblocks, such as preexisting bugs were fixed and the project was prepared for my future work
- I studied extensive information about biofeedback, Neurofeedback, software applications of biofeedback, ADHD and its treatments
- Two games were planned out with my supervisor with accordance to the project needs and actual requirements
- The Car Race game was implemented and tested
- The Air Shooter game was implemented and tested
- User testing and hardware was carried out, confirming positive results

8.2 Potential improvements

While the implemented games are fully functional, considerable improvements can be made, such as improved graphics, additional sounds, in-game bonuses and achievements, multiplayer interaction with other patients as well as global players across the Internet, etc. Since the main reason for the creation of the games was the replayability factor and interactivity factor, it would be beneficial to reach the standard of modern PC games, which would allow for better engagement with child patients.

Additionally, extra games could be implemented in the future: I believe that there are still many suitable game genres, including games oriented towards adult patients such as chess or checkers. Since patients' tastes may vary considerably, providing a bigger choice would potentially improve the therapeutic results

Finally, the games could potentially be used by healthy home users wishing to utilize the biofeedback technique. The success of devices such as the Emotiv EPOC shows great potentials for such applications.

Bibliography

[1] BUDZYNSKI, Thomas. *Introduction to quantitative EEG and neurofeedback: advanced theory and applications*. 2nd ed. Amsterdam: Academic Press/Elsevier, 2009, xxii, 502 p. ISBN 978-012-3745-347.

[2] CARO, Xavier J. a Earl F. WINTER. EEG Biofeedback Treatment Improves Certain Attention and Somatic Symptoms in Fibromyalgia: A Pilot Study. Applied *Psychophysiology and Biofeedback*. 2011, vol. 36, issue 3, s. 193-200. DOI: 10.1007/s10484-011-9159-9.
Available from: http://link.springer.com/10.1007/s10484-011-9159-9

[3] KOTCHOUBEY, Boris, Simone BUSCH, Ute STREHL a Niels BIRBAUMER. *Applied Psychophysiology and Biofeedback*. vol. 24, issue 4, s. 213-233. DOI: 10.1023/A:1022226412991. Available from: http://link.springer.com/10.1023/A:1022226412991

[4] CRITCHLEY, H. D., Simone BUSCH, Ute STREHL a Niels BIRBAUMER. Brain activity during biofeedback relaxation: A functional neuroimaging investigation. *Brain*. vol. 124, issue 5, s. 1003-1012. DOI: 10.1093/brain/124.5.1003. Available from: http://www.brain.oupjournals.org/cgi/doi/10.1093/brain/124.5.1003

[5] COLLURA, Thomas. History and evolution of electroencephalographic instruments and techniques. Available from: http://www.brainmaster.com/tfc/index_files/Publications/535-027%20theory%20and%20evolution%20of%20electroencephalographic.pdf

[6] JURCAK, Valer, Daisuke TSUZUKI and Ippeita DAN. 10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems. *NeuroImage*. 2007, vol. 34, issue 4, s. 1600-1611. DOI: 10.1016/j.neuroimage.2006.09.024. Available from: http://linkinghub.elsevier.com/retrieve/pii/S1053811906009724

[7] HERWIG, Uwe, Peyman SATRAPI and Carlos SCHÖNFELDT-LECUONA. Using the International 10-20 EEG System for Positioning of Transcranial Magnetic Stimulation: Their validity as relative head-surface-based positioning systems. *Brain Topography*. 2003, vol. 16, issue 2, s. 95-99. DOI: 10.1023/B:BRAT.0000006333.93597.9d. Available from: http://link.springer.com/10.1023/B:BRAT.0000006333.93597.9d

[8] BIRBAUMER, Niels, Ander RAMOS MURGUIALDAY, Cornelia WEBER a Pedro MONTOYA. Chapter 8 Neurofeedback and Brain–Computer Interface. s. 107. DOI: 10.1016/S0074-7742(09)86008-X. Available from: http://linkinghub.elsevier.com/retrieve/pii/S007477420986008X

[9] REYES, Janet and Sabri TOSUNOGLU. An Overview of Brain-Computer Interface Technology Applications in Robotics. Available from: http://www.eng.fiu.edu/mme/robotics/elib/FCRAR2011-An%20Overview%20of%20Brain-Computer%20Interface-PAPER-JR-ST-FIU.pdf

[10] NIEDERMEYER, Ernst and F LOPES DA SILVA. *Electroencephalography: basic principles, clinical applications, and related fields*. 5th ed. Philadelphia: Lippincott Williams, c2005, xiii, 1309 p. ISBN 07-817-5126-8.

[11] CHERRY, Kendra. The Anatomy of the Brain [online]. Available from: http://psychology.about.com/od/biopsychology/ss/brainstructure.htm

[12] HASLACH, Henry. Contribution of Brain Extracellular Fluid in Mild Damage of Brain Tissue From External Insults [online]. Available from: http://www.cecd.umd.edu/projects/contribution-of-brain.html

[13] THOMPSON, James. Electrical and Chemical Activity in the Brain [online]. Available from: http://bit.ly/1n3cgeQ

[14] AMERICAN ACADEMY OF SLEEP MEDICINE. Technical *Specifications for Electroencephalogram (EEG)* [online]. Available from: http://www.aasmnet.org/scoringmanual/v2.0.2/html/index.html?ATechnicalSpecificationsforElect.html

[15] BYU MICHAEL LARSON LAB. What is EEG? [online]. Available from: https://cogneuro.byu.edu/Pages/FAQ.aspx

[16] POJKAR, Tomas. *Tvorba zpetnovazebnıch hernıch vizualizacı pro neurofeedback system*. 2013. Available from: https://dip.felk.cvut.cz/browse/pdfcache/pojkatom_2013dipl.pdf

[17] VAVRECKA, Michal. *Neurofeedback Prezentacni Video*. [online]. https://www.youtube.com/watch?v=QORqHH-YSd4

[18] Need For Extreme. [online]. Available from: http://www.thegameszone.org/sportsgames/need_for_extreme/need_for_extreme.html

[19] EMOTIV. Emotiv EPOC [online]. Available from: http://www.emotiv.com/epoc/

Appendix A
List of abbreviations

FPS	Frames per second
PC	Personal Computer
EEG	Elektroencefalogram
ADHD	Attention Deficit Hyperactivity Disorder
CPU	Central Processing Unit
JDK	Java Development Kit
RAM	Random Access Memory

Appendix B

CD contents

The attached CD contains the following files:

- /session/ - files required to start the simulation
- / EEG2DGames.AirplaneShooter/ - Airplane Shooter Game
- / EEG2DGames.CarRace/ - Car Race Game
- /text/ - contains the text documents pertaining to this thesis
- /contents.txt – list of files