UNIVERSITY of DERBY

WEB-BASED ARTIFICIAL INTELLIGENCE GAMES

MEDITERRANEAN MC

Student Name: **Alexanter Baso**
Student ID: **100533316**
Supervisor: **Sotirios Chronopoulos**

To all those people who supported me

and they believed in me ...

and to all who believe

that science is the knowledge of the few

in the service of many ...

Acknowledgements

"Above all else, I might want to thank God, who blessed me, invigorated me and gave me opportunities to complete this dissertation. I would like to thank my professors and every person who assumed a part in my academic accomplishments. I want to express my sincere appreciation to my supervisor, Professor Sotirios Chronopoulos, who convincingly guided and urged me to be proficient and professional. I would like to thank all those who with their daily support, patience, and positive thinking, contributed to the fulfillment of the goals I had set. The biggest "thank you" to my loved ones, to my parents, who accepted all my choices and provided me with support all this time, without which none of what I have achieved so far would be a reality. Also, I would like to thank the University of Derby and Mediterranean College for giving me the opportunity to obtain a degree."

Abstract

People are evolving along with technology. Since computers were just computers that was given commands line by line, now one computer can do everything. From writing texts, codes, editing videos and images, playing games to even thinking. Artificial Intelligence has penetrated deep into our lives because it is used almost everywhere. It is used in online shopping, in healthcare, in smart homes where it helps people make their lives easier, in cybersecurity but also in self-driving cars, such as Tesla. It is also used in games where the opponent (which is the computer) is programmed to work with static or dynamic Artificial Intelligence algorithm. My project is based on the Artificial Intelligence used in games, thus creating a suite of 2D games suite that were popular a few years ago which are Pong, Soccer Car, Snake and Tic Tac Toe. The form of this dissertation is divided into 8 Chapters and begins with Chapter 1 in which an introduction is made to technology and artificial intelligence so that the reader does not have difficulty in understanding the text later. Also, in this chapter the logic and the purpose of the project are pointed out. In Chapter 2, will be referred an explanation of what the Web application is, how it works and what are its advantages. Next, in Chapter 3 will list the requirements of the project, the MoSCoW prioritization method I used, and the languages used to implement it. In Chapter 4 will discuss in detail these languages and the program I used for writing these languages. Also, the Software Development Life Cycle, the Waterfall Methodology will be analyzed in detail along with its advantages and disadvantages. In Chapter 5 will be analyzed and reported in detail all the diagrams that helped me in implementation of the project. Then, in Chapter 6 will be analyzed some of the most important pieces of the code for each game and file separately. In Chapter 7 there will be a presentation of the game through the pictures. Finally, in Chapter 8, a summary will be made, and some conclusions will be drawn.

Keywords: Pong Game, Soccer Car Game, Snake Game, Tic Tac Toe Game, Artificial Intelligence, HTML Code - CSS Code - JavaScript Code, UML Diagrams, Waterfall Methodology

Table of Contents

Tables

Table of Code Examples

Table of Pictures

Table of Diagrams

Table of Presentation Pictures

Table of Code Pictures

1) Introduction

1.1) History of Technology

At these times where we are and live is the age of technology. Technology is evolving very fast and so all people need to get acquainted with it. The word technology has been around for many years and its significance has been constantly changing for two hundred years. Technology is the amount of procedures, abilities, and techniques utilized in the creation of services, products or in the achievement of scientific investigation goals. This term became known in the twentieth century regarding the Second Industrial Revolution. By the 1930s, the term technology alluded to the investigation of the industrial arts. The application of technology in our years is done more in systems such as machines, which get an input where they change it according to the use of the system and then produce a result. These are called technological systems. So, in 1940 computers began to appear, which consisted of valves and relays. These machines, that is, computers, brought a new and very important era for humans because through them they could achieve many things. A branch called "Information Technology" has now been added to the term technology. Information technology (IT) is the utilization of computers to recover, send, store, and control data. Computers are machines that can be programmed which means that people could use these machines to achieve their goals or make their lives easier. Computer programming is the way toward building and planning a computer program to achieve a particular result or to run a particular task. Programming includes tasks some of which are: creating algorithms in a chosen programming language and analysis. The program source code can be written in one or more programming languages that are understandable to developers. The motivation behind writing computer programs is to create a series of commands that will automate the task on a computer for solving a given issue. A programming language is a proper language including a bunch of instructions that produce different sorts of output and are utilized in computer programming to create algorithms. Some of the most used programming languages, markup languages and style sheet languages are Java, Python, PHP, JavaScript, Html and CSS. Due to these programming languages and the increase in the power of computers over time, computer games began to be created that were played through it. In these games, as the years went by, was incorporated the Artificial Intelligence which maximized the user experience.

1.2) History of Artificial Intelligence

Artificial intelligence (AI) is the intelligence that machines have. That is, it is the intelligence that man has (to think and decide) but without feeling joy, sadness and more. Is the capacity of a computer program to memorize and think and is one of the developing innovations which tries to mimic human thinking. As machines become progressively competent, assignments considered to require "intelligence" are included to the meaning of AI. Artificial Intelligence field was born at a laboratory at Dartmouth College in 1956 by John McCarthy to recognize the field from cybernetics and break the impact of the cyberneticist Norbert Wiener. A lot of data and AI procedures have become a fundamental piece of the technology industry, assisting with solve of many issues in computer science. The artificial intelligence market has reached one billion dollars by 1985. In the last part of the 1990s and mid twenty-first century, AI started to be utilized for medical and clinical diagnosis, data extraction and in many video games.

1.3) Related Work and Critical Analysis

With the passage of time, electronic games "gain" more users because they become more realistic and thus have more interest. Some of these games also incorporate Artificial Intelligence where they arrive the user experience to the highest level. Old games like my project games have been left in the lurch and no company has dealt with them to add Artificial Intelligence and upgrade them with modern technology. But many developers, on their own, try to develop and upgrade them and thus help each other as a result they make something very modern. I am one of them, and the project is a game suite in which Artificial Intelligence, Start Menu, Pause Menu, and others have been integrated. So, I made a suite of four games where the user can play against the computer, with the help of Artificial Intelligence. The code lines of the whole project that is a web-based game suite are 2650 and there is an explanation of the whole source code where some parts I will mention below.

Game Names			
Pong	**Soccer Car**	**Snake**	**Tic Tac Toe**

Table 1: Games of game suite.

1.4) Rational of the Project

As mentioned earlier, many games have enhanced the user experience. However, most of them either focus on the graphics and ignore the importance of these games or have never played them.

Therefore, a suite of games that have the same front-end design has been created. The purpose of creating this suite is that anyone who wants to play games such as the above that have Artificial Intelligence technologies and are merged into one project can use this suite, without having to search for each individual game to play (which are very few on the internet that support Artificial Intelligence). Users will be able to play any game against the computer at any time, automatically pause the game when changing tabs, and be automatically transferred when the game closes, in the home menu.

1.5) Purpose of the Project

The purpose of the project is to create and develop a suite of games that will run on the web, in which the user will be able to play against the computer.

The goals of the project are the following:

- Games must run on different computers and run on all browsers.
- Games must run fast in browsers.
- Games must be small in size so that it does not take up much space in hard drive.
- Games must have the same front-end design.
- Games must pause when the user change tab.

2) Web Application

2.1) What is a Web Application?

A web application is an application that is installed on a server and runs only from the browser, unlike native applications which run locally on the computer consuming space, memory, and system resources. Thus, the user interacts with the application only if he has an active network connection. (What Is a Web Application? How It Works, Benefits and Examples, 2021) A web application is different from a regular website. The websites that we constantly come across when we browse on the internet make the user surf the internet for many hours. Also, websites only offer information to the user, which has been manually written by other users. Web applications are more advanced than websites in terms of functionality and are more complex to implement because they use complex web-based technologies. Web applications use files written in a standard language like JavaScript, HTML and CSS, which are upheld by an assortment of browsers. Web applications can be considered as a particular variation of server-side and client-side software where the customer software is downloaded to the customer computer when visiting a web page, utilizing standard strategies like HTTP. Web browsers help the user to surf between web pages and web applications thus acting as a client for the customer. Thus, unlike websites, web applications are tools that do a specific task and do not just offer information.

2.2) How a Web Application work?

As mentioned above, the most common languages in which a web application is written and programmed are the JavaScript, Html and CSS. These are languages that the browser understands so that the program can run. Some of the web applications also need the help of the server to process data (This is achieved using the PHP language), which are called dynamic web applications, and some others do not need the server, so they are static. The game suite requires only one browser and an internet connection to run it.

2.3) Advantages of a Web Application

- Web applications are applications that run through the browser and are independent of the operating system of the computer. They can run on all computers that have a compatible browser.
- All users who use a web application use the same version, because the web application is always updated automatically from the central server, so there are no problems with their versions as in native applications.
- Web applications are installed on a web server so that they are not installed on the local computer and do not take up space on the hard disk.
- They reduce the "stealing" of programs (hackers cannot crack or hack them) because most of them work only with a subscription.

- For a client to run a web application does not need to have a high-performance computer, making it appropriate for the user.
- Web applications help developers and the cost of businesses. Businesses do not need to hire many programmers who know different languages to develop the program on different operating systems. Thus, programmers in this regard do not have much difficulty in implementing the code and are not responsible for making multiple versions for each operating system.

3) Requirements

The requirements of the application were determined according to the functionality of the application, divided into three parts, the functional, the non-functional and MoSCoW method.

3.1) Functional Requirements

- The game suite must be easy to use.
- The game suite must be run on all platforms and browsers that support latest HTML, CSS, and JavaScript Languages.
- All games in the suite must have the same front-end design.
- Games must be played by one player against the computer.
- The size of the games must be small.
- The game suite must have a main menu in which the player selects the game he wants to play.
- At each tab change, the game must be automatically paused.

3.2) Non-Functional Requirements

- The project must be uploaded to the internet and the user must enter a specific domain to enter the web application and start playing.
- Games could have the option "Player versus Player" and "Artificial Intelligence versus Artificial Intelligence". So, the player could choose if he wants to play against the computer, against another player or put the AI to play with itself without his intervention.
- The games will not be compatible with mobile devices but only with computers.

3.3) MoSCoW method

The MoSCoW analysis is a prioritization procedure utilized in administration, business investigation, project management, and programming improvement to arrive at a typical comprehension with customers on the significance they put on the delivery of every prerequisite. It is otherwise called MoSCoW method or MoSCoW prioritization. The term MoSCoW itself is an abbreviation gotten from the main letter of every one of four classes with which I will prioritize the project: M(Must have)oS(Should have)C(Could have)oW(Won't have). The Os do not mean anything and have been added to the word to make it pronounceable.

Classes	***Features***
MUST	• The game suite must be easy to use. • The game suite must be run on all platforms and browsers that support latest HTML, CSS, and JavaScript Languages. • All games in the suite must have the same front-end design. (FPS games must have Pause Option) • Games must be played by one player against the computer. • The size of the games must be small. • The game suite must have a main menu in which the player selects the game he wants to play. • At each tab change, the game must be automatically paused.
SHOULD	• The project must be uploaded to the internet and the user must enter a specific domain to enter the web application and start playing.
COULD	• Games could have the option "Player versus Player" and "Artificial Intelligence versus Artificial Intelligence". So, the player could choose if he wants to play against the computer, against another player or put the AI to play with itself without his intervention.
WON'T HAVE	• The games will not be compatible with mobile devices but only with computers.

Table 2: MoSCoW Method

3.4) Languages used

This project was made to work in a web environment, so I used for developing, the programming language JavaScript, Markup Language HTML and Style Sheet Language CSS.

Back-End Development	Front-End Development
JAVASCRIPT	HTML
	CSS
	JAVASCRIPT

Table 3: Languages used in project.

Below will be analyzed the languages in detail.

4) Analysis

4.1) HTML Markup Language

In my project I used HTML to structure the whole project. Through it I created divs with classes or ids and buttons, I put a google font in the project, I integrated some libraries, and I connected the CSS file with the project.

HTML was created by physician Tim Berners-Lee, who worked at CERN. HTML (initialization of HyperText Markup Language) is the primary markup language for web pages, and its elements are the basic building blocks of web pages or web applications. HTML is written in the form of HTML elements consisting of tags, which are enclosed in "greater than" and "less than" (for example <body>) symbols, within the content of the web page. HTML tags usually work in pairs (for example <title> and </title>), with the first being called the start tag and the second being the closing tag (or in other cases the opening tag and the closing tag respectively). Between tags, web designers can place text, tables, images, etc. The purpose of a web browser is to read HTML documents and compose them into pages that can be read or listened to. The browser does not display HTML tags but uses them to present the content of the page. HTML elements are used to build all its websites. HTML allows images and other objects to be embedded within the page and can be used to display interactive forms. Provides methods for creating structured documents (i.e., documents consisting of the content they convey and the content formatting code) defining structurally important elements for the text, such as headers, lists, paragraphs, links and more. Command scripts can also be embedded in languages such as JavaScript, which influence the behavior of HTML web pages and make them from static to interactive. Web browsers can also refer to CSS formatting styles to define the appearance and layout of text and other material. (Berners-Lee, T., Fielding, R., Frystyk, H.,1996) The World Wide Web Consortium (W3C) organization, which creates and maintains HTML and CSS templates, encourages the use of CSS instead of various HTML elements for content presentation purposes. An example HTML code is referred below.

```
<html>
        <head>
                <title> Web-Based Artificial Intelligence Games </title>
        </head>
<body>
</body></html>
```

Code Example 1: HTML code Sample

4.2) CSS Style Sheet Language

In my project I used CSS to structure the appearance of the whole project. Through this, I configured the divs with the classes or the ids and the buttons, that I created in the HTML file, in such a way that the front-end of the project is more beautiful.

The CSS language was created in 1994 by Hakon Wium Lie when he was working with Tim Berners-Lee at CERN, who created the HTML Markup Language. CSS (Cascading Style Sheets) is a computer language that belongs to the category of style sheet languages used to control the appearance of a document written in a markup language. (CSS, 2021) It is used to control the appearance of a document written in HTML and XHTML languages, i.e., to control the appearance of a web page and a website in general. The CSS Language that changes web page colors, alignment, backgrounds and gives more options than the Markup Html Language. So, in order to create a beautifully designed website with all the above and many other functions, this language is necessary.

4.2.1) Advantages of CSS compared to HTML

- Much more flexibility. CSS made formatting possible which was impossible or very difficult with classical HTML.
- Give better maintenance of websites. The appearance of an entire site can be controlled by a single external CSS file. Thus, any change in the style of the website can be made with a single change in this file, instead of editing multiple points on each page on the site.
- Smaller file size, as each formatting rule is written only once and not at each point applied.
- Search engines do not "confuse" between content and its formatting, but have access to plain content, so it is much easier to record and index it.
- Faster pages. When we use an external CSS file, the browser saves it in the cache the first time it loads a page of our site or web application, so it does not need to download it again every time the user enters our site or web application.

An example CSS code is referred below.

```
.main (This is a class that is created in a HTML file)

{

        background-color: red;

        height: 100%;

        width: 100%;

        position: relative;

        overflow: hidden;

        border-radius: 10px;

}
```

Code Example 2: CSS code Sample

4.3) JavaScript Programming Language

In my project I used JavaScript to create the functionality of the game. It is the language in which every game is programmed and works properly. With this the AI was created and edited, the player's side and in a few cases, it was used for front-end design.

JavaScript is a programming language which helps in the design of the front-end design of a website or in the implementation of a back-end server. It was originally part of the implementation of web browsers, so that client-side scripts can communicate with the user, exchange data asynchronously and dynamically change the content of the document being displayed. JavaScript is a prototype-based scripting language, it is dynamic, with weak types, and it has functions as first-class objects. Its syntax is influenced by C. (JavaScript, 2021) JavaScript copies many names from Java, but generally these two languages are not related and have very different semantics. The basic JavaScript design principles come from the Self and Scheme programming languages. It is a language based on different programming paradigms (multi-paradigm), supporting object-oriented, imperative, and declarative programming paradigms. JavaScript is also used in applications outside of web pages - such examples are PDF documents, site-specific browsers, and desktop widgets. Newer JavaScript virtual machines and development frameworks (such as Node.js, Angular.js and Angular.js) have also made JavaScript more popular for server-side web application development. The JavaScript programming language was created by Brendan Eich of Netscape company under the name Mocha. Later, Mocha was renamed to LiveScript, and eventually to JavaScript, mainly because its development was more influenced by the Java programming language. LiveScript was the official name of the language when it was first released in beta with the Web browser, Netscape Navigator version 2.0 in September 1995. The original name of JavaScript is LiveScript and in 1995 Sun Microsystems officially renamed it "JavaScript". This was done when the Netscape browser was upgraded to version 2.0B3. JavaScript has been very successful as a client-side language for executing code on web pages and on web applications and has been included in various Web browsers. An example JavaScript code is referred below.

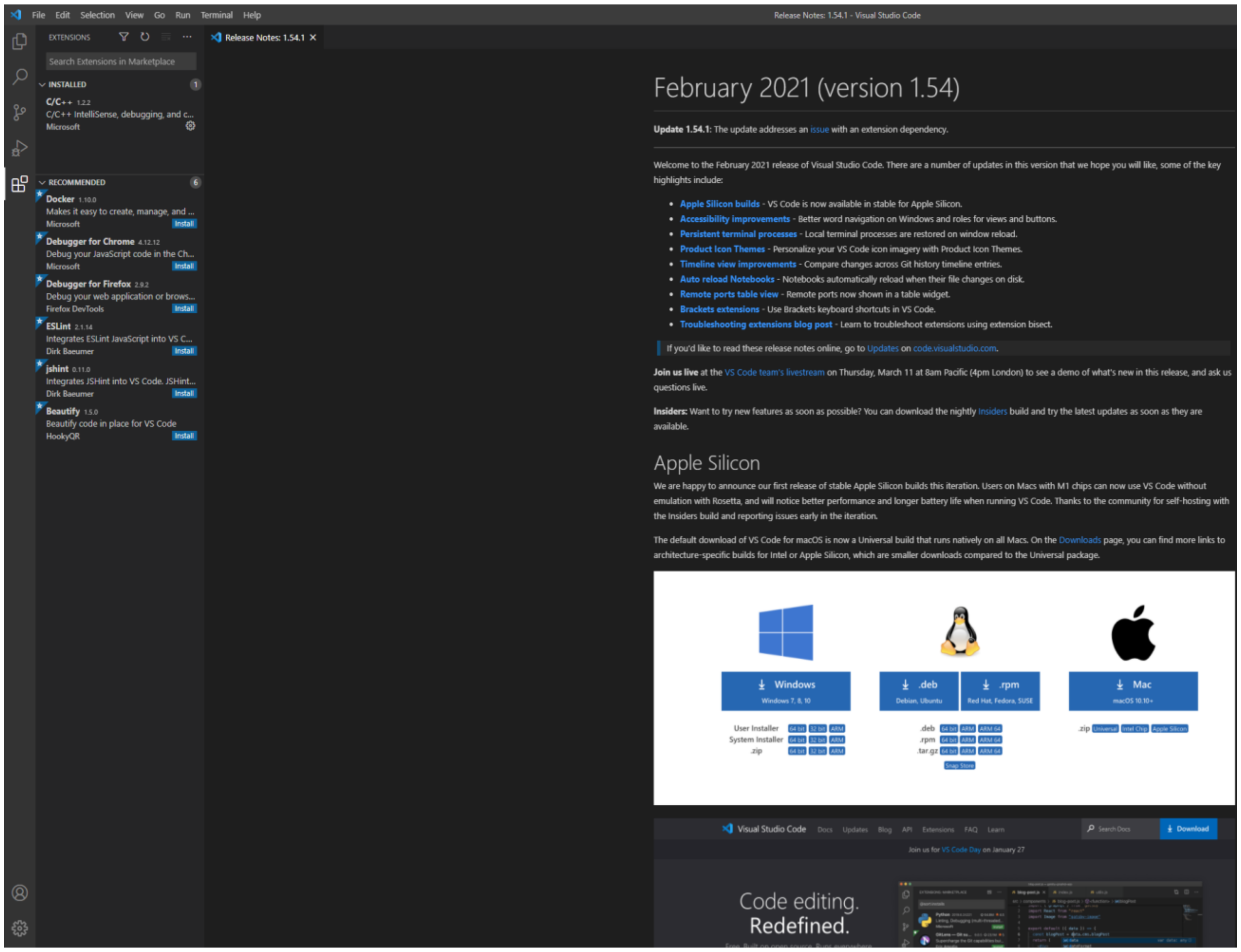
```
const firstInteger = prompt('Enter the first integer: ');
const secondInteger = prompt('Enter the second integer: ');
const resultOfFirstInteger = firstInteger % 10;
const resultOfSecondInteger = secondInteger % 10;
if(resultOfFirstInteger == resultOfSecondInteger)
{
   console.log(`In ${firstInteger} and ${secondInteger} last digit is the same.`);
}
else
{
   console.log(`In ${firstInteger} and ${secondInteger} last digit isn’t the same.`);
}
```

Code Example 3: JavaScript code Sample

4.4) Visual Studio Code

This project it was created entirely with the use of the below source-code editor.

Visual Studio Code was first reported on April 29, 2015 by Microsoft. A preview version has been created then. On November 18, 2015, Visual Studio Code was licensed by MIT and uploaded the source code accessible to GitHub. Visual Studio Code is a program for writing and editing code. It is one of the efforts of Microsoft to participate in the field of Open Source. (microsoft/vscode, 2015) It is based on Atom and is its own version of a very good text editor. The platforms that support it are Linux, Windows and macOS. Editor highlights incorporate help for debugging, code refactoring, syntax highlighting and more. Users can change the look of the program and add add-ons for a better experience. There are many languages supported by Visual Studio Code, such as Java, C/C++, HTML, Python, Go, CSS, JavaScript etc. The languages I used were all supported by this source-code editor, i.e., the Style Sheet Language CSS, the Markup Language HTML and Programming Language JavaScript.



C/C++ IntelliSense, debugging, and c...

Docker — Makes it easy to create, manage, and ... — Microsoft

Debugger for Chrome — Debug your JavaScript code in the Ch... — Microsoft

Debugger for Firefox — Debug your web application or brows... — Firefox DevTools

ESLint — Integrates ESLint JavaScript into VS C... — Dirk Baeumer

jshint — Integrates JSHint into VS Code. JSHint... — Dirk Baeumer

Beautify — Beautify code in place for VS Code — HookyQR

February 2021 (version 1.54)

Update 1.54.1: The update addresses an issue with an extension dependency.

Welcome to the February 2021 release of Visual Studio Code. There are a number of updates in this version that we hope you will like, some of the key highlights include:

- **Apple Silicon builds** - VS Code is now available in stable for Apple Silicon.
- **Accessibility improvements** - Better word navigation on Windows and roles for views and buttons.
- **Persistent terminal processes** - Local terminal processes are restored on window reload.
- **Product Icon Themes** - Personalize your VS Code icon imagery with Product Icon Themes.
- **Timeline view improvements** - Compare changes across Git history timeline entries.
- **Auto reload Notebooks** - Notebooks automatically reload when their file changes on disk.
- **Remote ports table view** - Remote ports now shown in a table widget.
- **Brackets extensions** - Use Brackets keyboard shortcuts in VS Code.
- **Troubleshooting extensions blog post** - Learn to troubleshoot extensions using extension bisect.

If you'd like to read these release notes online, go to Updates on code.visualstudio.com.

Join us live at the VS Code team's livestream on Thursday, March 11 at 8am Pacific (4pm London) to see a demo of what's new in this release, and ask us questions live.

Insiders: Want to try new features as soon as possible? You can download the nightly Insiders build and try the latest updates as soon as they are available.

Apple Silicon

We are happy to announce our first release of stable Apple Silicon builds this iteration. Users on Macs with M1 chips can now use VS Code without emulation with Rosetta, and will notice better performance and longer battery life when running VS Code. Thanks to the community for self-hosting with the Insiders build and reporting issues early in the iteration.

The default download of VS Code for macOS is now a Universal build that runs natively on all Macs. On the Downloads page, you can find more links to architecture-specific builds for Intel or Apple Silicon, which are smaller downloads compared to the Universal package.

Picture 1: Visual Studio Code Open-Source Editor

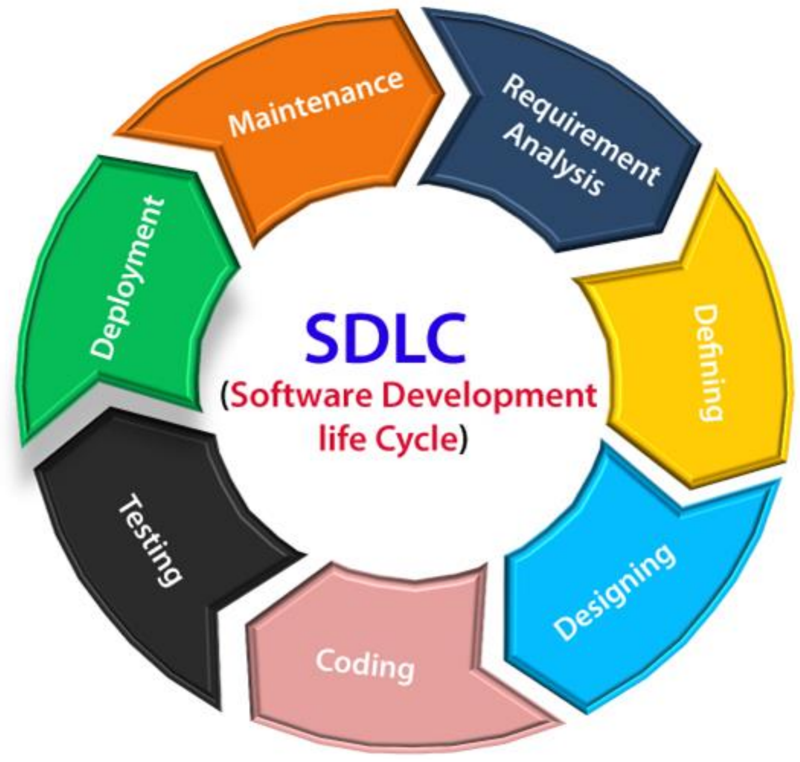
4.5) Software Engineering Process

In this project I have followed the Software Development Life Cycle. But what is the Life Cycle and what are its stages? The answers to these questions will be answered below and in Chapter 5 we will see in diagrams the implementation of the project for some of the stages.

Software Development Life Cycle is a process taken after by IT businesses to plan, implement and test IT products. SDLC process is taken after entirely to guarantee quality products are conveyed to the customers and shoppers inside the arranged budget and time estimates (Shobika, 2016). SDLC has seven stages whereas each stage has it possess deliverables which serve as the input for the following stage.

The seven stages of Software Development Life Cycle are:

1. Requirements Analysis
2. Defining
3. Designing
4. Coding
5. Testing
6. Deployment
7. Maintenance



Picture 2: Software Development Life Cycle

- Requirements Analysis
  - Sets up the components for building the system, including hardware and software requirements, and other fundamental components. Business prerequisites and specifications are gathered amid this stage. Sets up the expectations for software functionality and recognizes which from system requirements the software influences. In requirements analysis deciding the interaction required with other applications and databases. (Shobika, 2016)
- Defining
  - When the requirements are accumulated, it will be analyzed by a group of people to form beyond any doubt the plausibility of consolidating software development with the requirements accumulated. (Papataxiarhis et al., 2010)
- Designing
  - In this stage is deciding the software framework of a system to meet the particular prerequisites. This stage determines the major components and the interaction of those, but it does not determine the structure of each component.
- Coding
  - The main of this phase is development of the product. Whole plan will be broken into modules and programmers will work in separate teams. Coding standards and rules are taken while creating the application.
- Testing
  - It is decided if the software meets the desired necessities and finds any blunders display within the code.
- Deployment
  - When the application is successfully tested, it can be released to the clients for their utilization. (Shobika, 2016)
- Maintenance
  - Addresses issues and upgrade requests after the release of the software. (Papataxiarhis et al., 2010)

4.6) Development Methodology

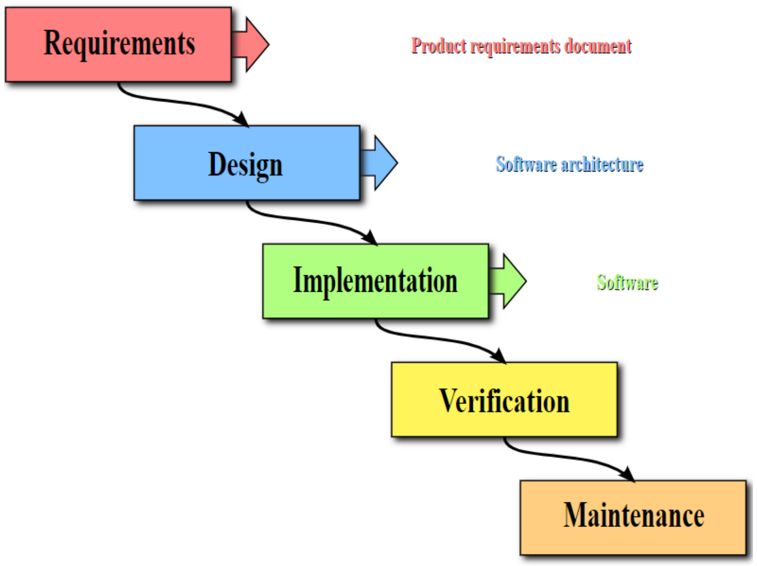
The methodology that I used to create this project is Waterfall. The waterfall methodology was developed by Royce in 1970. Waterfall methodology is the classical methodology of computer software engineering. This methodology is one of the most used methodologies and is broadly utilized in government projects and in numerous major IT businesses. This methodology emphasizes planning in early stages, and it ensures design faults before they create. It was the first methodology to be created and widely accepted, while still remaining particularly popular for small to medium sized applications as it contributes to the successful construction of reliable products in a short period of time. In this methodology the different phases are separated and followed serially. Each phase produces intermediate products which are used by the subsequent phases and culminates in a process of validation or verification of the products produced, with the aim of eliminating any errors. This methodology has its advantages but also its disadvantages.

4.5.1) Waterfall Methodology Advantages

- Strengthens habits: define-before-design and design-before-code.
- Works well on small or medium projects.
- The oldest and most mature method of software development.
- Easily understood and accepted by those involved in the development process.
- Often compatible with customer specifications.
- The steps of the waterfall model are the building blocks of the other methods.
- The model helps to divide the work between salespeople, managers, analysts, and developers.

4.5.2) Waterfall Methodology Disadvantages

- Program is conveyed late in project, delays revelation of genuine errors.
- The linearity of the model is rarely found in real works.
- System analysis and requirements identification can rarely be completed at the beginning of a project.
- It takes a long time from the start of the project until the first deliverable version of the system. In the meantime, only documentation is delivered.
- Customers and users are slow to get a functional picture of the system.
- The development team is slow to get a tangible version of the system. It is easy to disappoint an effort that for a long time does not seem to pay off.
- The model easily leads to a complete separation of the roles of developers, analysts, salespeople, and managers, with potential negative consequences.



Picture 3: Waterfall Methodology

5) Software Implementation

5.1) UML Diagram

I also used the UML Diagram to have an idea of how the whole program should work. Unified Modeling Language (UML) is the standard language for designing and mapping templates in software engineering. Used to graphically display, identify, construct, and document the components of a software system. (Rumbaugh, Jacobson and Booch, 1998) It can be used in various stages of development, from requirements analysis to the control of an integrated system. It consists of a set of pre-agreed terms, symbols and diagrams that allow:

- Mapping the limits of a system and its basic functions, using "use-cases" and "actors".
- Creating patterns for the behavior of objects with "state diagrams".
- The representation of a static structure of a system using "class diagrams".

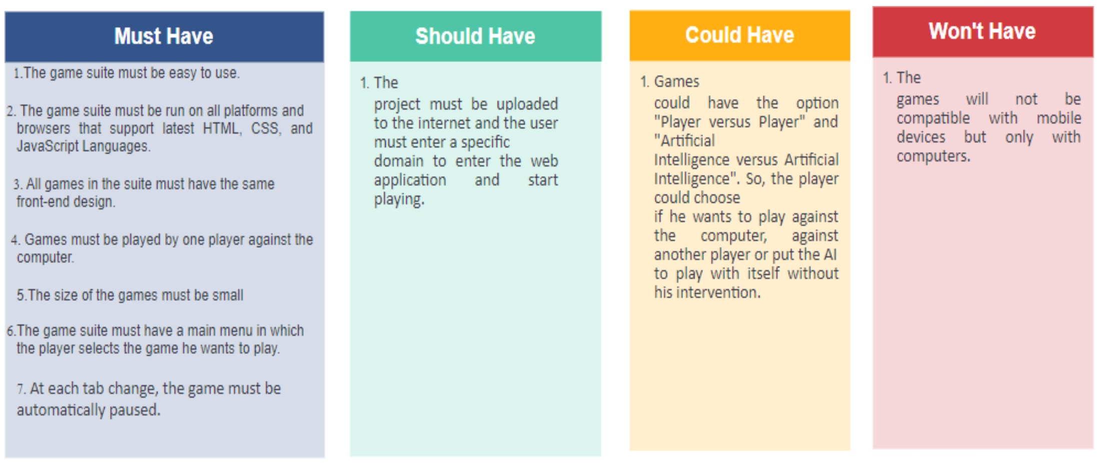
Using this diagram, I was able to achieve three goals which are the following:

- To visualize the whole project visually so that I have an overall picture of the project.
- Configure the entire structure of the project.
- I was able to rely on it to build the project.

5.2) Requirements Analysis

5.2.1) MoSCoW Diagram

For the requirements analysis I used the Moscow diagram to know what needs to be completed and implemented, what I could put next if I continued this project and what did not need to be implemented. Moscow has been analyzed in detail above.



Must Have	Should Have	Could Have	Won't Have
1.The game suite must be easy to use. 2. The game suite must be run on all platforms and browsers that support latest HTML, CSS, and JavaScript Languages. 3. All games in the suite must have the same front-end design. 4. Games must be played by one player against the computer. 5.The size of the games must be small 6.The game suite must have a main menu in which the player selects the game he wants to play. 7. At each tab change, the game must be automatically paused.	1. The project must be uploaded to the internet and the user must enter a specific domain to enter the web application and start playing.	1. Games could have the option "Player versus Player" and "Artificial Intelligence versus Artificial Intelligence". So, the player could choose if he wants to play against the computer, against another player or put the AI to play with itself without his intervention.	1. The games will not be compatible with mobile devices but only with computers.

Diagram 1: MoSCoW Diagram

5.2.2) Use Case Diagram

From the types of UML Diagrams, for my project I chose five of the diagrams, one of which is the UML Use Case Diagram. An UML Use Case Diagram is the essential type of requirements in software implementation for a non-construction yet software. Use Case indicate the normal conduct, i.e., what will be done in the project, and not the specific strategy for getting it starting the developing. A basic idea of Use Case is that it encourages us plan a system from the end client's point of view. It is a compelling strategy for communicating machine conduct in the client's terms by determining all remotely obvious machine conduct.

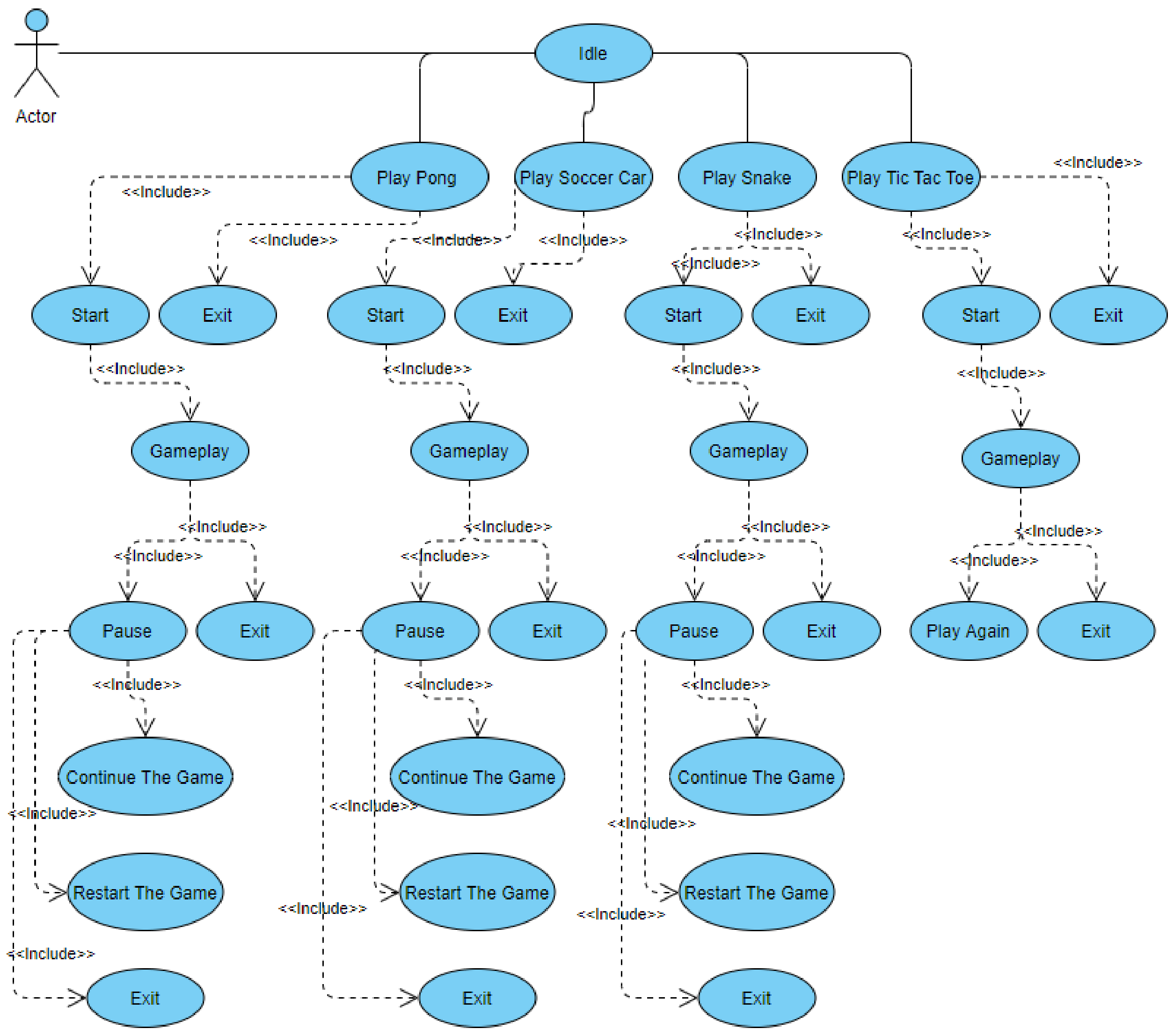


Diagram 2: Use Case Diagram

5.2.2.1) Detailed Explanation of the above UML Use Case Diagram

In the above diagram "Actor" is the user who uses the game suite. So, the first page which I have named Idle appears to the user and it is the selection of games. There the user can choose one of the games available, namely Pong, Soccer Car, Snake and Tic Tac Toe.

Thus, when the user chose to play one of the three fps games, which are Pong, Soccer Car and Snake, he is taken to the page where the selected game has been implemented. The first option that appears to the user is to either start the game or exit the game. When the user chooses to exit the game then he is automatically transferred to Idle, i.e., to the selection of games where the user can select a game. When the user chooses to start playing the game starts normally. Once the game starts, because it is a fps game, i.e., there is movement, at the bottom there is a pause option and an exit option. When the user select pause then the game "freezes" and a menu appear with options "Continue the game" where the game continues normally, the option "Restart the game "where the game starts from the beginning, and finally the" Exit "option where it is automatically transferred to Idle.

Then, when the user selects Tic Tac Toe to play, which is not a fps game, he is taken to the page where the game has been implemented. The first option that appears to the user is to either start the game or exit the game. When the user chooses to exit the game then he is automatically transferred to Idle. When the user chooses to start playing the game starts normally. Once the game starts, because it is not a fps game, i.e., there is no continuous movement, at the bottom there is no pause and exit option. Thus, since the game is a case of seconds, the user can exit the game or play again as soon as he wins, loses, or draws against the Artificial Intelligence.

5.3) Software Design

For the Software Design I used the State Machine and Class Diagram.

5.3.1) State Machine Diagram

UML State Machine, otherwise called UML Statechart, commonly are utilized to depict conduct for an object. The idea of State Machine Diagram is tied in with getting sorted out the way an application, website and more. UML State Machine is a variation of Harel statechart, extended and adjusted out by UML. An object reacts distinctively to a similar occasion which depends on the situation in which it is. Works to such an extent that an element or every one of its sub-elements is consistently in precisely one of various potential states and where there are all around characterized restrictive changes between these states. State Machine Diagrams are typically applied to objects however can be applied to any component that has conduct to different substances, for example, actors, subsystems and use cases.

The advantages of State Machine Diagram are the following:

- The object lifecycle, from implementation to deliver, can be demonstrated. This helps the comprehension of the members about the demonstrated item.
- The allowed conditions of an item and the occasions set off by state transitions can be portrayed. This makes the conduct of the item noticeable and fathomable.
- State Machine Diagram can be utilized effectively and much of the time, particularly since even unpracticed perusers can undoubtedly comprehend them with a little practice.

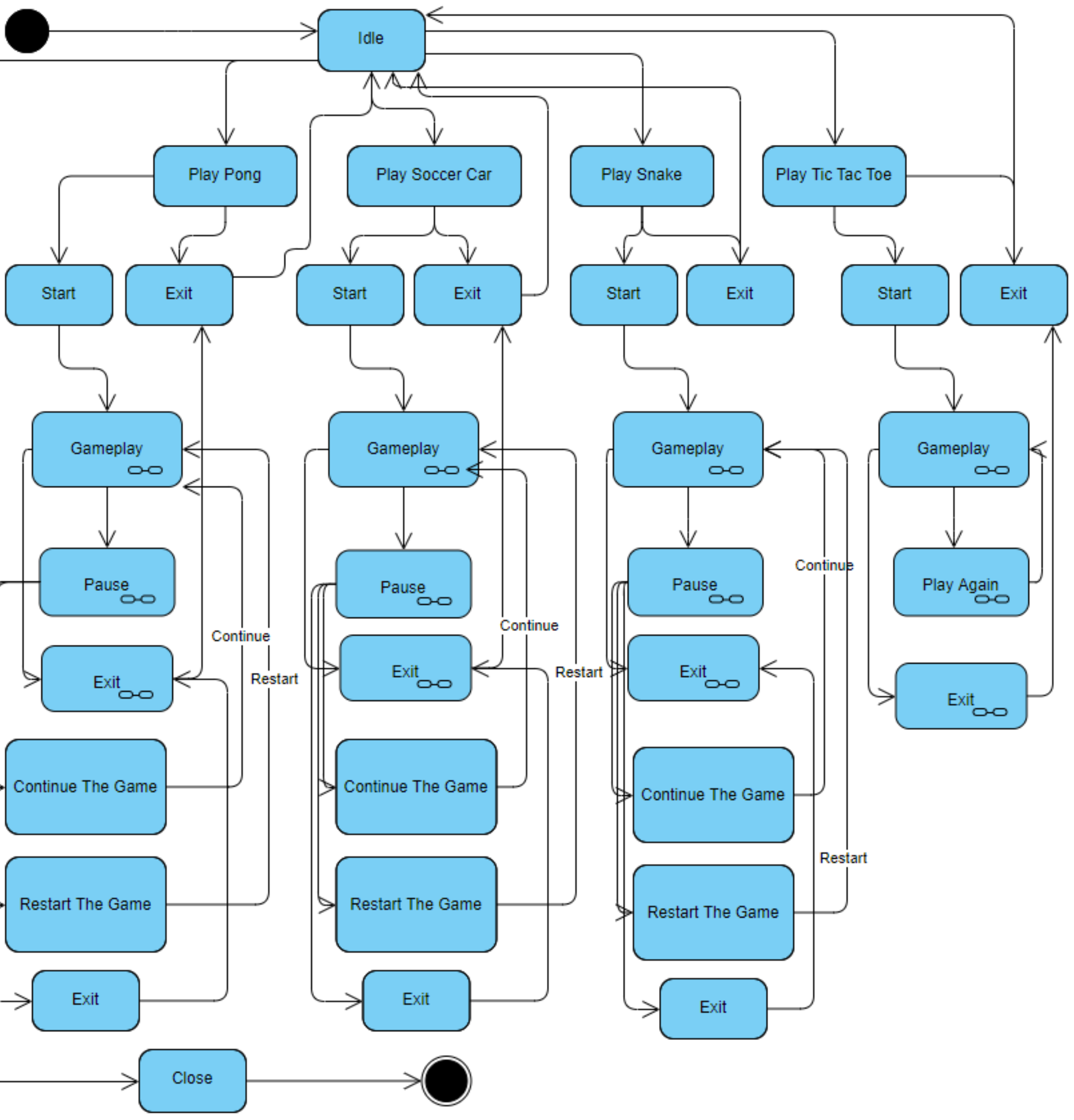


Diagram 3: State Machine Diagram

5.3.1.1) Detailed Explanation of the above State Machine Diagram

In my project I used the State Machine Diagram to have an idea of how the program will behave in each state change. As we can see in the diagram above, the black dot without the outline is the start of the program. The first state is the "Idle". Here are all the available games that the user can play. As we see, these are Pong, Soccer Car, Snake and Tic Tac Toe. The user can start playing or close the window. When the user chooses to close the window then the games suite closes (this is represented by the black dot with the outline in the bottom of the diagram).

When the user selects one of the three fps games which are Pong, Soccer Car and Snake, the state he was, changes and is transferred to the beginning of the game seeing two states, Start and Exit, where the first option starts the game and the second exit the game. When the user presses Exit, then the first state returns, i.e., it returns to idle. When the user presses Start, then the state changes, this state is the game that starts, and the player can play. Within the state there are two other states which are Pause and Exit. When the user presses Exit the result will be the same as the first Exit. When the user presses Pause, the state of the game changes, the game stops and three states are displayed, "Continue the game", "Restart the game" and Exit. When the user presses Exit the result will be the same as the first Exit. When the user clicks "Continue the game" the state of the game changes and returns to the state that was in the Gameplay. The same happens with "Restart the game" with the difference that everything is returned to its default values.

When the user selects the game, which is not a fps game which is Tic Tac Toe, the state it was in (i.e., in idle) changes and is transferred to the beginning of the game seeing two states, Start and Exit, where the first one starts the game and the second exit the game. When the user presses Exit, then the first state returns, i.e., it returns to idle. When the user presses Start, then the state changes, this state is the game that starts, and the player can play. When the game is over, that is, when the user wins, loses or draws against Artificial Intelligence, then two new states appear where are Play Again and Exit. Exit transports the player to the first state while Play Again transports it to the game, i.e., to the Gameplay with the values being restored to their original state.

5.3.2) Class Diagram

In programming, a Class Diagram is a kind of static construction that depicts the design of a program by showing the program classes, their traits, methods, and the connections among objects. It is utilized for general theoretical displaying of the design of the application, and for point by point demonstrating the classes that are translated to source code. In the Class Diagram, the entities (the classes) capture the interaction between them and all their elements. The Class Diagram is the principal procedure of object-oriented programming or object-oriented paradigm. The goal of Class Diagram is to demonstrate the static perspective on an application. Class charts are the charts which can be straightforwardly planned with object-oriented languages and are generally utilized at the hour of construction. UML charts like sequence chart can just give the stream of the application, but Class diagram is somewhat unique.

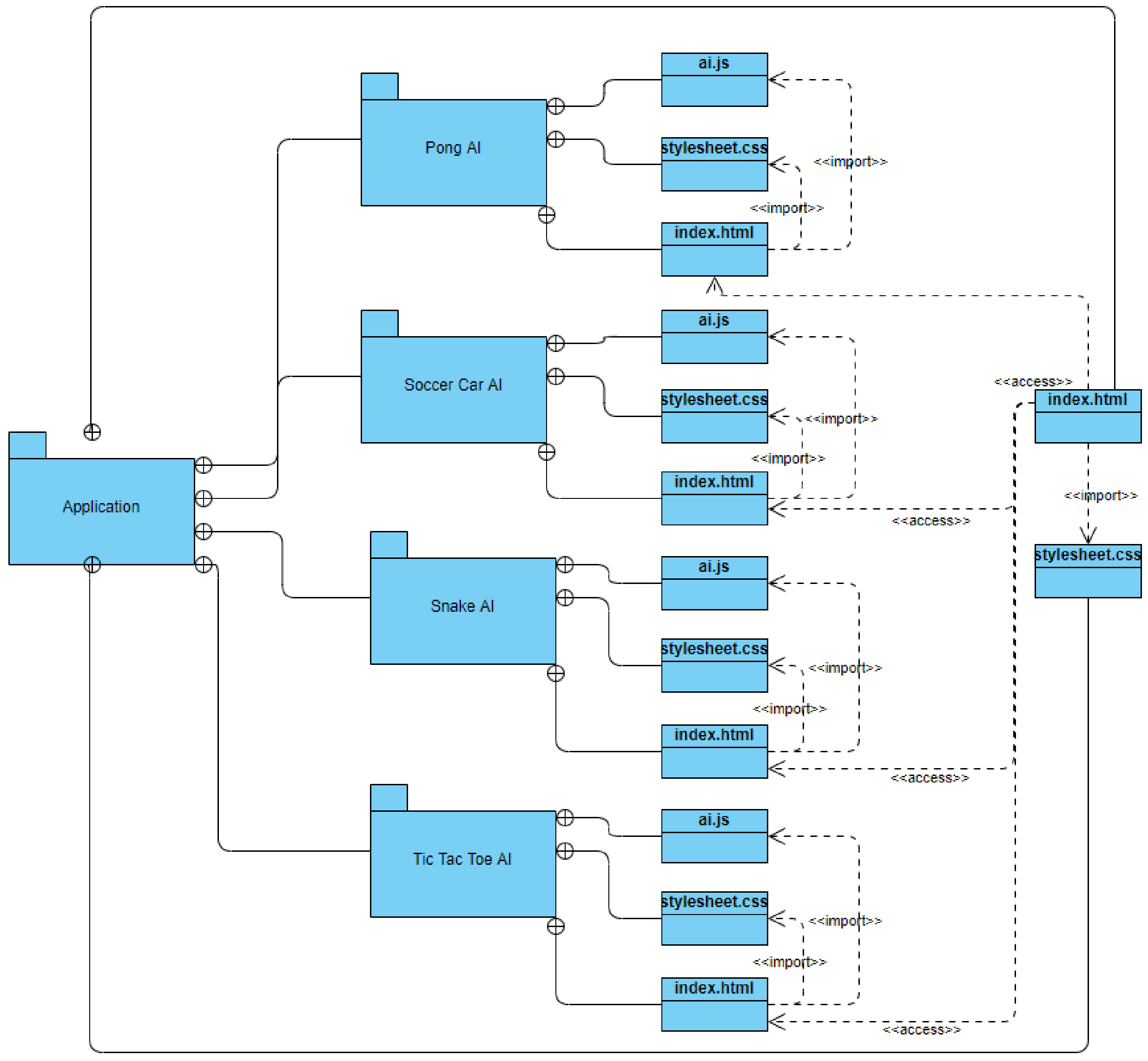


Diagram 4: Class Diagram

The reason for the class chart can be summed up as:

- Investigation and plan of the static perspective on an application.
- Depict duties of a program.
- Very important for diagrams like deployment.

The advantages of Class Diagram are the following:

- Class charts give you a feeling of direction. They give a good understanding into the construction of the program.
- They offer a brisk overview of the collaboration occurring among the distinctive components just as their properties and connections.
- Class charts are straightforward and quick. With the correct programming they are simple to make. They are the underpinning for making programs.

5.3.2.1) Detailed Explanation of the above Class Diagram

In my project, I did not use the class diagram in the classic way, because not all games are written with object-oriented paradigm. The class diagram I made helped me to structure the project into folders and files so that I have an idea of how to program the files and how to connect them. As we see above, the "Application" folder contains all the files of the project. This folder contains four folders and two files. The first folder contained in the main folder is Pong AI and contains three files which are the ai.js file, the stylesheet.css file and the index.html file. The last file is the main file where it connects to the other two files and so the game can start.

The second folder contained in the main folder is Soccer Car AI and contains three files which are the ai.js file, which is the implementation file of this game, the stylesheet.css file and the index.html file such as in the first game. The index.html file is the main file of the game where it connects to the other two files and so the game can start.

The third and fourth folders in the main folder are Snake AI and Tic Tac Toe AI respectively and each folder as above contains three files which is the ai.js file, which is the implementation file of each game respectively, the stylesheet.css file and the index.html file. The index.html file is the main file of each game, where it is linked to the other two files of each folder and thus the corresponding game can start.

The files in the main "Application" folder are the index.html and stylesheet.css. Stylesheet.css is linked to index.html and index.html is linked to each index.html of each folder to start the game.

5.3.3) Flow Diagram

A Flow Diagram is a sort of diagram that addresses a work process or workflow. A Flow Diagram can likewise be characterized as a diagrammatic portrayal of a program, a way to deal with settling a task. Also, shows the means as boxes of different sorts, and the connection of these elements is done with arrows. This portrayal represents an answer to a given issue. Flow Diagrams are utilized in planning or analyzing of an application in different fields. A Flow Diagram is depicted as "cross-functional" when the diagram is separated into various vertical parts, to portray the hierarchy and the control of various units. An element showing up in a specific part is inside the check of that unit. A cross-functional Flow Diagram permits the creator to accurately find the liability regarding playing out an activity or settling on a choice, and to show the liability of each unit for various pieces of a process. Flow Diagram is utilized in planning and illustration straightforward projects or applications. Like different kinds of charts, they help see what is happening and help comprehend a process, and maybe likewise find more defects inside the process. There are various sorts of Flow Diagrams: each sort has its own arrangement of boxes and elements. The two most basic sorts of boxes in a Flow Diagram are:

- A step that ordinarily called process and is indicated as a rectangular box.
- A decision typically signified as a rhombus with its two corresponding options.

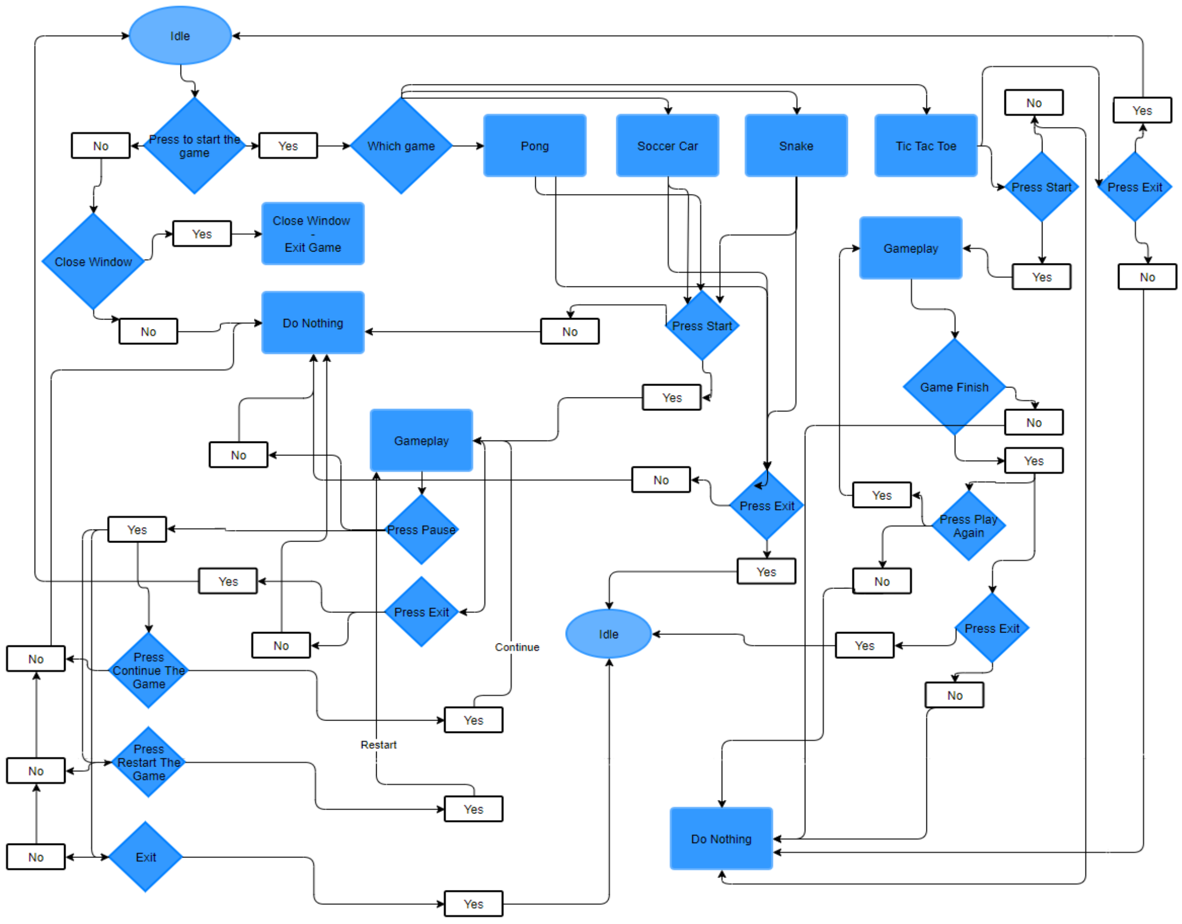


Diagram 5: Flow Diagram

5.3.3.1) Detailed Explanation of the above Flow Diagram

In my project I used the Flow Diagram to have an idea of how the program will flow. As we can see in the diagram above, the idle is the start of the program. Here are all the available games that the user can play. As we see, these games are Pong, Soccer Car and Snake, which are fps games and Tic Tac Toe which is not a fps game. The user can start playing or close the window. For each option, a decision must be made.

The first option we encounter is if the user pressed the game start button to start. If this is not true, then the option is whether the user has closed the window. If the user chooses to close the window then the games suite closes, otherwise nothing is done. If it is true that the user pressed the game start button, the option arises as to which game he chose. If the user chose to play one of the three fps games, then the homepage of the game is displayed which has two options if the user pressed Start and if he pressed Exit. If the user selected Exit then he is automatically transferred to Idle, if not then nothing is done. If the user selected Start then he is transferred to Gameplay, i.e., the game starts, if he did not select Start, then nothing happens. In Gameplay there are two other options which are whether the user pressed Pause or pressed Exit. If the user selected Exit then it is automatically transferred to Idle, if not then nothing is done. If the user did not select Pause then nothing happens, otherwise if he selected it then a menu with three other options appears. These are if the user clicked "Continue the game”, if he clicked "Restart the game" and if he clicked Exit. If the user selected Exit then it is automatically transferred to Idle, if not then nothing is done. If the user selected "Continue the game", then he is transferred to the game and continues to play normally, otherwise if he did not select it nothing happens. The same happens with "Restart the game" with the difference that everything is returned to its default values.

If the user chose the game Tic Tac Toe, which is not a fps game, then the main page of the game appears which has two options, if the user pressed Start and if he pressed Exit. If the user selected Exit then it is automatically transferred to Idle, if not then nothing is done. If the user selected Start then it is transferred to Gameplay, i.e., the game starts, if he did not select Start, then nothing happens. The next option is whether the game is over. If the game is not over, nothing happens, but if the game is over, then there are two more options, which are whether the user clicked Play Again or if he clicked Exit. If the user selected Exit then it is automatically transferred to Idle, if not then nothing is done. If the user selected Play Again then it is automatically transferred to Gameplay, i.e., it starts playing from the beginning with the default values, if the user did not click Play Again, then nothing happens.

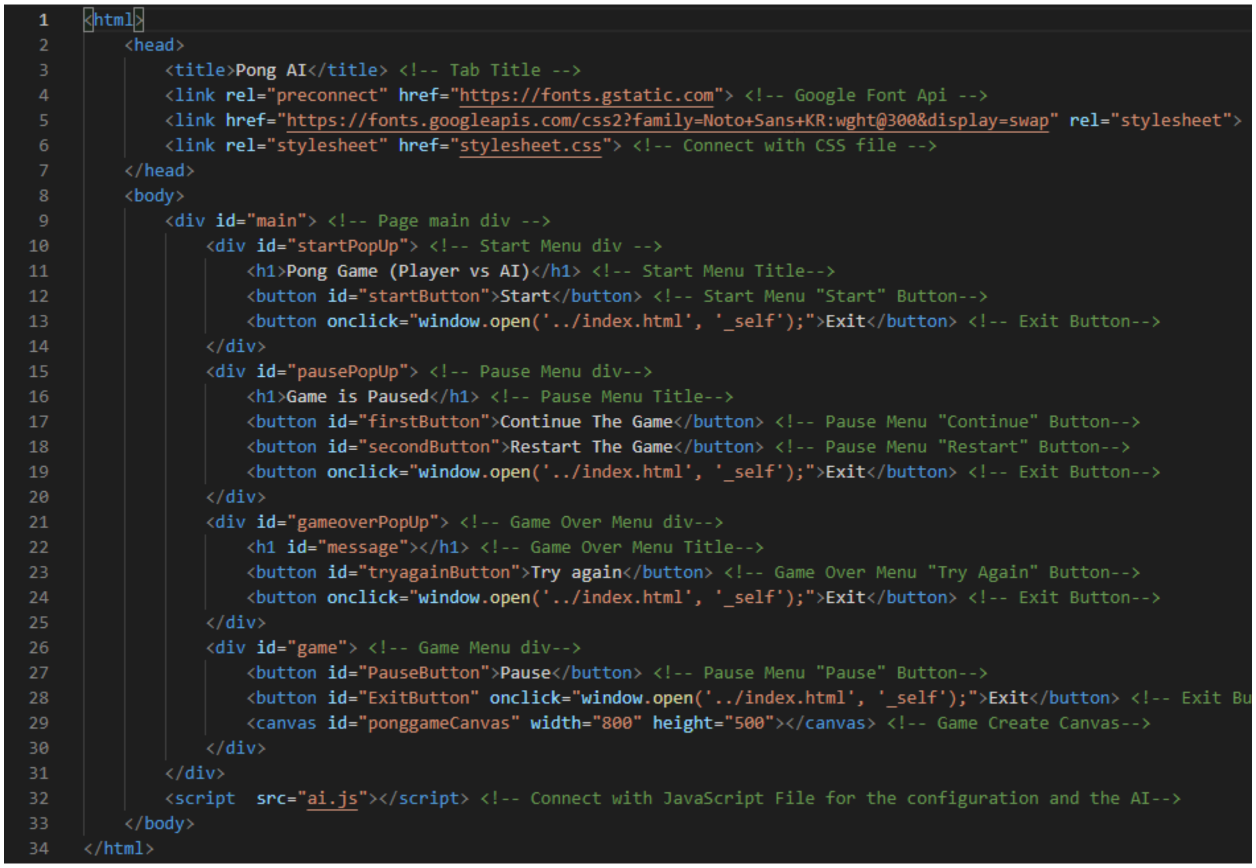
6) Coding

The program consists of a total of 2650 lines of code which are divided into nine files. Each game consists of 3 files, which are index.html, stylesheet.css and ai.js. The main menu of the game suite consists of two files which are stylesheet.css and index.html, which is linked to the index.html of each game. Below we will explain some important code pieces from each game.

6.1) Pong Game (Player versus Artificial Intelligence)

The purpose of this game is for the player to win against the Artificial Intelligence. So, in order to win the player must reach first the five points. If he does not reach them first, then Artificial Intelligence wins the game. The player can play this game many times.

6.1.1) File: index.html

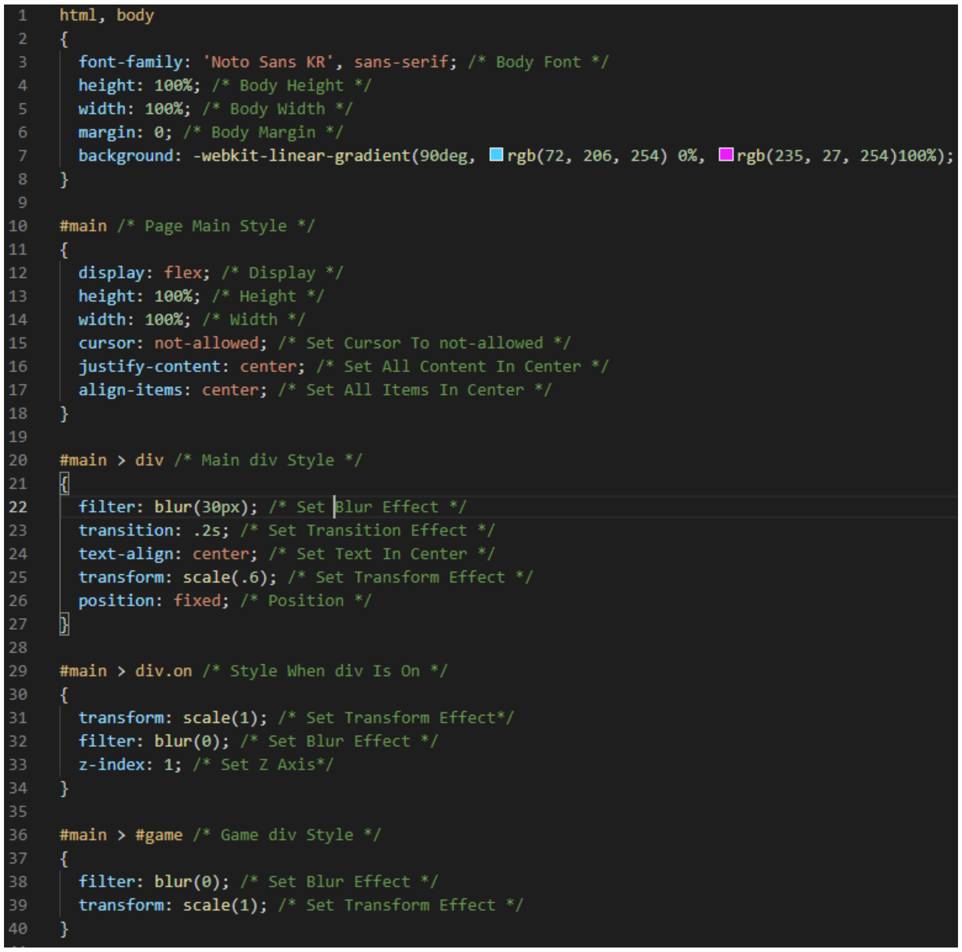


```html
<html>
    <head>
        <title>Pong AI</title> <!-- Tab Title -->
        <link rel="preconnect" href="https://fonts.gstatic.com"> <!-- Google Font Api -->
        <link href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@300&display=swap" rel="stylesheet">
        <link rel="stylesheet" href="stylesheet.css"> <!-- Connect with CSS file -->
    </head>
    <body>
        <div id="main"> <!-- Page main div -->
            <div id="startPopUp"> <!-- Start Menu div -->
                <h1>Pong Game (Player vs AI)</h1> <!-- Start Menu Title-->
                <button id="startButton">Start</button> <!-- Start Menu "Start" Button-->
                <button onclick="window.open('../index.html', '_self');">Exit</button> <!-- Exit Button-->
            </div>
            <div id="pausePopUp"> <!-- Pause Menu div-->
                <h1>Game is Paused</h1> <!-- Pause Menu Title-->
                <button id="firstButton">Continue The Game</button> <!-- Pause Menu "Continue" Button-->
                <button id="secondButton">Restart The Game</button> <!-- Pause Menu "Restart" Button-->
                <button onclick="window.open('../index.html', '_self');">Exit</button> <!-- Exit Button-->
            </div>
            <div id="gameoverPopUp"> <!-- Game Over Menu div-->
                <h1 id="message"></h1> <!-- Game Over Menu Title-->
                <button id="tryagainButton">Try again</button> <!-- Game Over Menu "Try Again" Button-->
                <button onclick="window.open('../index.html', '_self');">Exit</button> <!-- Exit Button-->
            </div>
            <div id="game"> <!-- Game Menu div-->
                <button id="PauseButton">Pause</button> <!-- Pause Menu "Pause" Button-->
                <button id="ExitButton" onclick="window.open('../index.html', '_self');">Exit</button> <!-- Exit Bu
                <canvas id="ponggameCanvas" width="800" height="500"></canvas> <!-- Game Create Canvas-->
            </div>
        </div>
        <script  src="ai.js"></script> <!-- Connect with JavaScript File for the configuration and the AI-->
    </body>
</html>
```

Code Picture 1: Pong – Index.html File

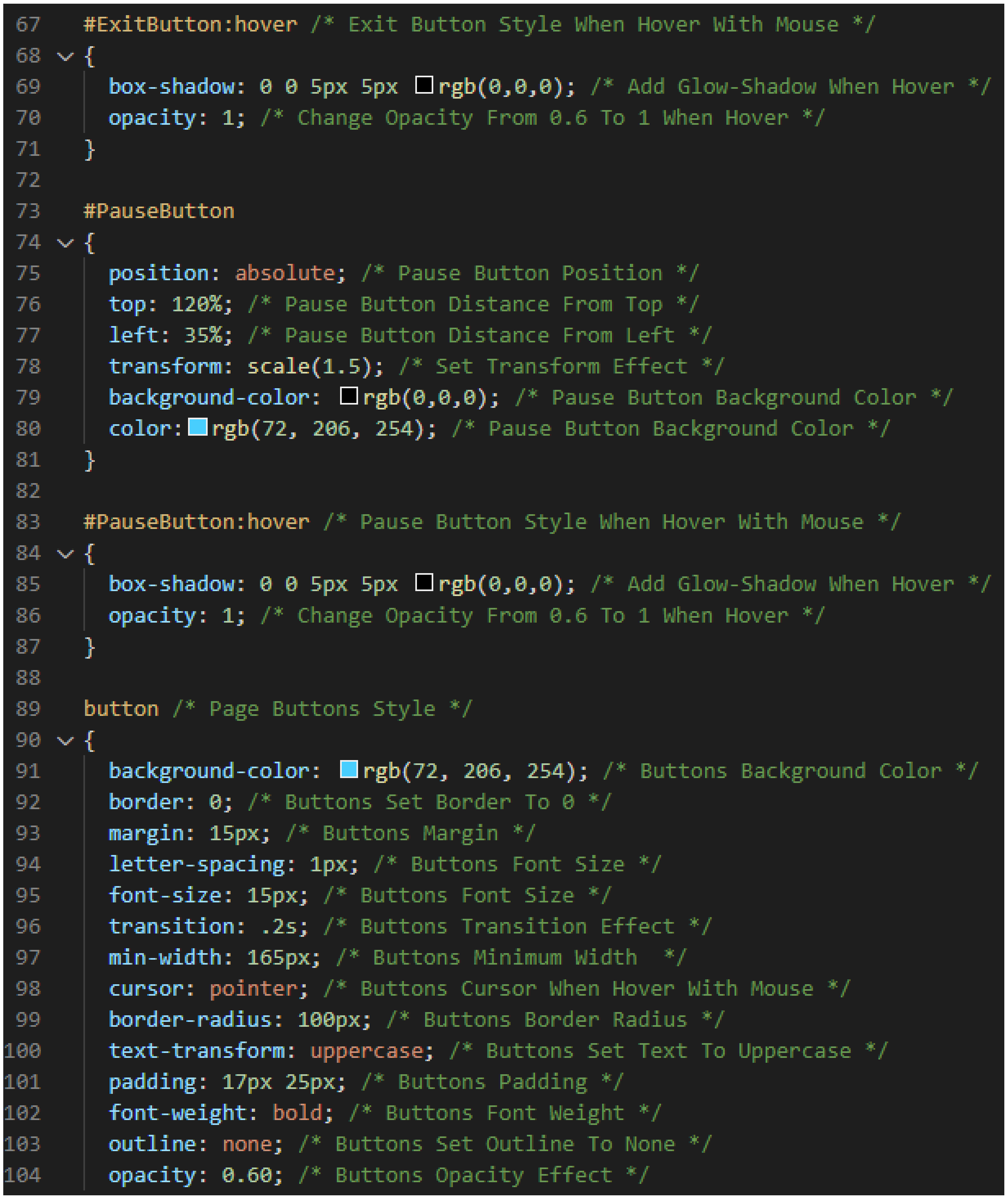
Each html file starts with the <html> tag and closes again with the </html>. Inside these tags is written the code with the Markup Language HTML. Then between the tag <head> and </head> are written all the metadata that the application will use. In my code I have integrated a google font which is free, and I have linked to this the stylesheet.css file that contains the game's appearance. The main code of the file is written between the <body> and </body> tags. So, the game is divided into five divs that the "id" makes them unique. In the main div are integrated all the other divs, i.e., the startPopUp, which contains the Start and Exit buttons, the pausePopUp, which contains the Continue The Game, Restart The Game buttons and the Exit, the gameoverPopUp, which contains the message that appears when the game ends and the two buttons Try again and Exit. Finally, the main div incorporates the div game in which is the game and the two buttons that are the Pause and Exit buttons. Any change to the main div will change the rest of the divs. At the bottom of the <body> before closing this tag, there is the <script> and </script> tag in which the developer can write a JavaScript code. In my case, because the file would be too large, the JavaScript code was written to a separate file and through this tag the index.html file was linked to the ai.js file.

6.1.2) File: stylesheet.css



```css
html, body
{
  font-family: 'Noto Sans KR', sans-serif; /* Body Font */
  height: 100%; /* Body Height */
  width: 100%; /* Body Width */
  margin: 0; /* Body Margin */
  background: -webkit-linear-gradient(90deg, rgb(72, 206, 254) 0%, rgb(235, 27, 254)100%);
}

#main /* Page Main Style */
{
  display: flex; /* Display */
  height: 100%; /* Height */
  width: 100%; /* Width */
  cursor: not-allowed; /* Set Cursor To not-allowed */
  justify-content: center; /* Set All Content In Center */
  align-items: center; /* Set All Items In Center */
}

#main > div /* Main div Style */
{
  filter: blur(30px); /* Set Blur Effect */
  transition: .2s; /* Set Transition Effect */
  text-align: center; /* Set Text In Center */
  transform: scale(.6); /* Set Transform Effect */
  position: fixed; /* Position */
}

#main > div.on /* Style When div Is On */
{
  transform: scale(1); /* Set Transform Effect*/
  filter: blur(0); /* Set Blur Effect */
  z-index: 1; /* Set Z Axis*/
}

#main > #game /* Game div Style */
{
  filter: blur(0); /* Set Blur Effect */
  transform: scale(1); /* Set Transform Effect */
}
```

Code Picture 2: Pong – stylesheet.css File 1 (Part 1)



```css
#ExitButton:hover /* Exit Button Style When Hover With Mouse */
{
  box-shadow: 0 0 5px 5px rgb(0,0,0); /* Add Glow-Shadow When Hover */
  opacity: 1; /* Change Opacity From 0.6 To 1 When Hover */
}

#PauseButton
{
  position: absolute; /* Pause Button Position */
  top: 120%; /* Pause Button Distance From Top */
  left: 35%; /* Pause Button Distance From Left */
  transform: scale(1.5); /* Set Transform Effect */
  background-color: rgb(0,0,0); /* Pause Button Background Color */
  color: rgb(72, 206, 254); /* Pause Button Background Color */
}

#PauseButton:hover /* Pause Button Style When Hover With Mouse */
{
  box-shadow: 0 0 5px 5px rgb(0,0,0); /* Add Glow-Shadow When Hover */
  opacity: 1; /* Change Opacity From 0.6 To 1 When Hover */
}

button /* Page Buttons Style */
{
  background-color: rgb(72, 206, 254); /* Buttons Background Color */
  border: 0; /* Buttons Set Border To 0 */
  margin: 15px; /* Buttons Margin */
  letter-spacing: 1px; /* Buttons Font Size */
  font-size: 15px; /* Buttons Font Size */
  transition: .2s; /* Buttons Transition Effect */
  min-width: 165px; /* Buttons Minimum Width  */
  cursor: pointer; /* Buttons Cursor When Hover With Mouse */
  border-radius: 100px; /* Buttons Border Radius */
  text-transform: uppercase; /* Buttons Set Text To Uppercase */
  padding: 17px 25px; /* Buttons Padding */
  font-weight: bold; /* Buttons Font Weight */
  outline: none; /* Buttons Set Outline To None */
  opacity: 0.60; /* Buttons Opacity Effect */
```

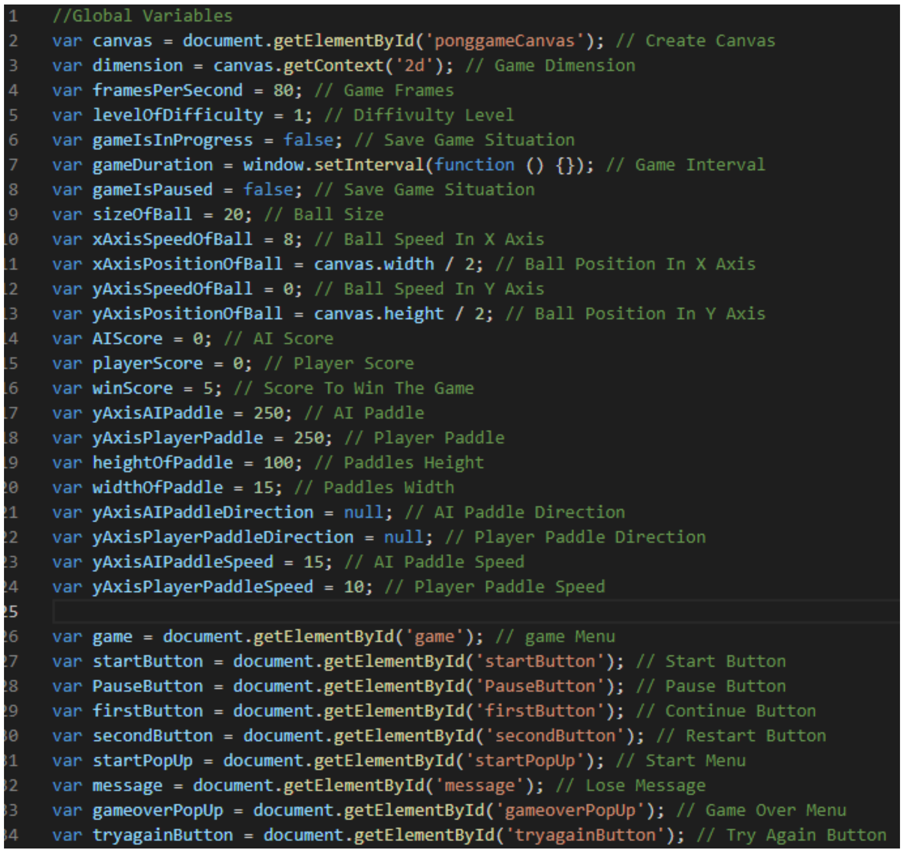
Code Picture 3: Pong – stylesheet.css File (Part 2)

The above code pictures are a part of the stylesheet.css file, where the appearance of the game is define. In this file the language with which we have to "program" is the Style Sheet Language CSS. This file is linked to the index.html file.

As we can see, this language is not a programming language but works with specific commands. At the beginning, in Code Picture 2, is determined the height, width, font and background that our page will have. The "#" symbol identifies the "id" tha is declared in index.html. So, by writing "#main" we modify the div that have given the id "main" in the index.html file. Also, here is determined the height and width that this div will have, the alignment of the items and how the cursor will look. Below, is defined the appearance of all the divs contained in the div main. Thus, is determined the position of the div, the alignment of the text, the filter and some animations. Then when one of these divs is activated then their appearance changes with the commands we observe in the picture.

In Code Picture 3, we see that is defined the appearance of the buttons in the game. At the beginning we notice that is defined the ExitButton when we hover with the mouse. So, as soon as the user hovers with the mouse then the button appears clearly with some shadow from behind. The #PauseButton determines how the button will look like, i.e., is defined the position, the background of the button, the color of the font and an animation. At the end of the picture we specify the appearance of all the buttons that do not have an id and are in the game. In this are defined more things in relation to the other buttons. That is, is defined the background color of the buttons, animations, cursor, opacity, letter spacing, border radius etc.

6.1.3) File: ai.js

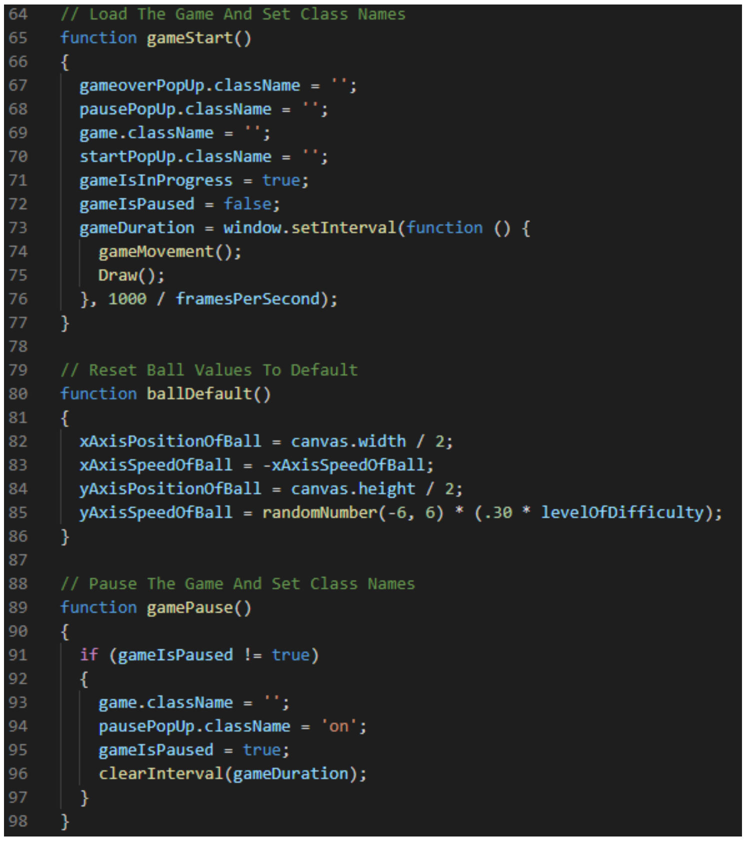


```
//Global Variables
var canvas = document.getElementById('ponggameCanvas'); // Create Canvas
var dimension = canvas.getContext('2d'); // Game Dimension
var framesPerSecond = 80; // Game Frames
var levelOfDifficulty = 1; // Diffivulty Level
var gameIsInProgress = false; // Save Game Situation
var gameDuration = window.setInterval(function () {}); // Game Interval
var gameIsPaused = false; // Save Game Situation
var sizeOfBall = 20; // Ball Size
var xAxisSpeedOfBall = 8; // Ball Speed In X Axis
var xAxisPositionOfBall = canvas.width / 2; // Ball Position In X Axis
var yAxisSpeedOfBall = 0; // Ball Speed In Y Axis
var yAxisPositionOfBall = canvas.height / 2; // Ball Position In Y Axis
var AIScore = 0; // AI Score
var playerScore = 0; // Player Score
var winScore = 5; // Score To Win The Game
var yAxisAIPaddle = 250; // AI Paddle
var yAxisPlayerPaddle = 250; // Player Paddle
var heightOfPaddle = 100; // Paddles Height
var widthOfPaddle = 15; // Paddles Width
var yAxisAIPaddleDirection = null; // AI Paddle Direction
var yAxisPlayerPaddleDirection = null; // Player Paddle Direction
var yAxisAIPaddleSpeed = 15; // AI Paddle Speed
var yAxisPlayerPaddleSpeed = 10; // Player Paddle Speed

var game = document.getElementById('game'); // game Menu
var startButton = document.getElementById('startButton'); // Start Button
var PauseButton = document.getElementById('PauseButton'); // Pause Button
var firstButton = document.getElementById('firstButton'); // Continue Button
var secondButton = document.getElementById('secondButton'); // Restart Button
var startPopUp = document.getElementById('startPopUp'); // Start Menu
var message = document.getElementById('message'); // Lose Message
var gameoverPopUp = document.getElementById('gameoverPopUp'); // Game Over Menu
var tryagainButton = document.getElementById('tryagainButton'); // Try Again Button
```

Code Picture 4: Pong – ai.js (Part 1)

In this file, the game is implemented. In Code Picture 4 we see that all the variables are defined. Some of these variables are framePerSecond, because it is a fps game, AIScore where is the AI score, yAxisPaddleDirection where it stores the paddle direction on the y axis, sizeOfBall where is the size of the ball, heightOfPaddle and widthOfPaddle where is the height and width of the Paddle. All the names of the variables are written in such a way that only from the name the programmer knows what it is and what this variable does.

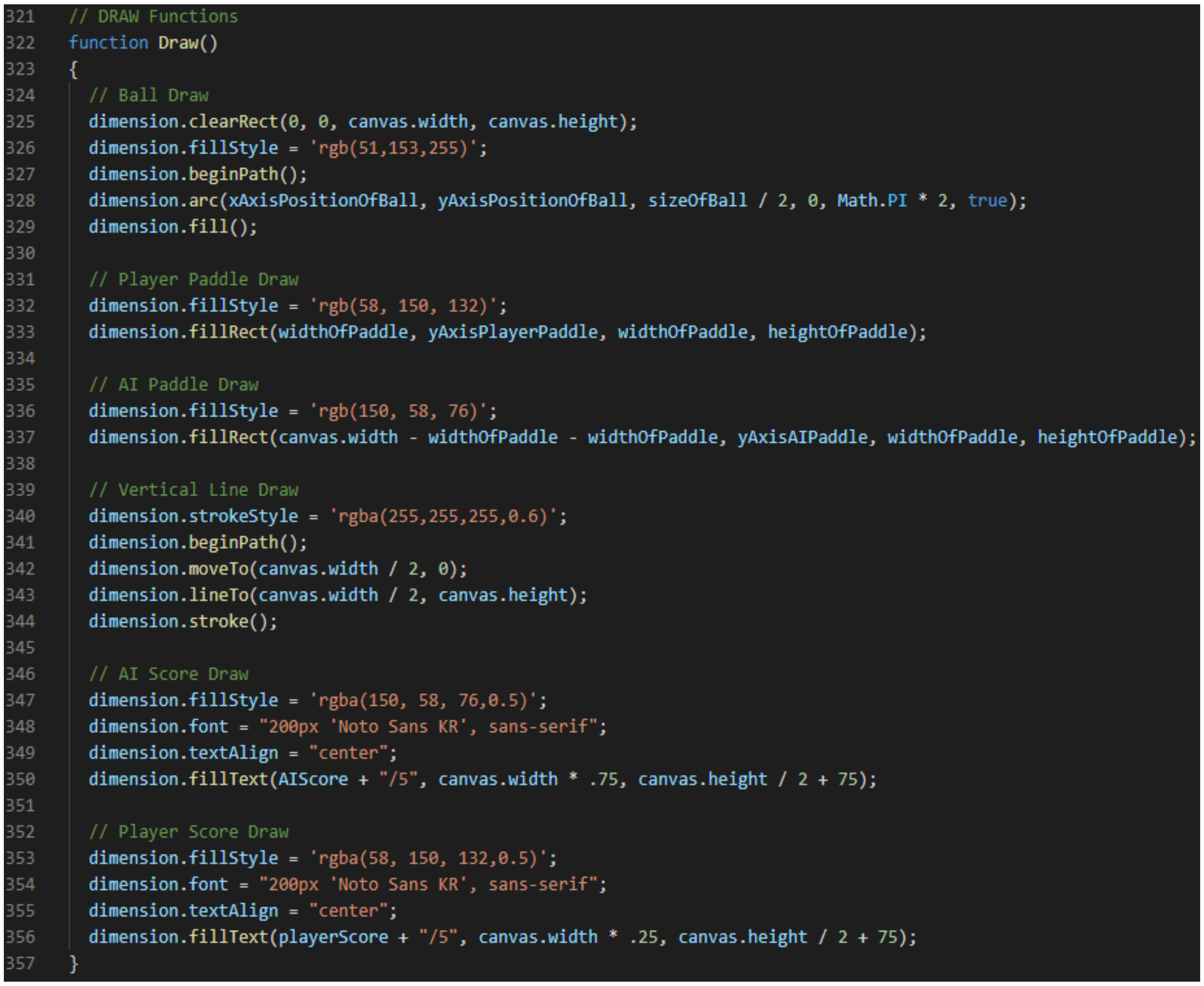
Then the elements that have id are connected with this file and are saved so they can be programmed. Some of them are startButton, which is the game start button and is stored in the variable startButton, PauseButton, which is the game pause button and is stored in the variable PauseButton, the message, which is the message that appears when game ends and is saved in the variable message.



```
// Load The Game And Set Class Names
function gameStart()
{
  gameoverPopUp.className = '';
  pausePopUp.className = '';
  game.className = '';
  startPopUp.className = '';
  gameIsInProgress = true;
  gameIsPaused = false;
  gameDuration = window.setInterval(function () {
    gameMovement();
    Draw();
  }, 1000 / framesPerSecond);
}

// Reset Ball Values To Default
function ballDefault()
{
  xAxisPositionOfBall = canvas.width / 2;
  xAxisSpeedOfBall = -xAxisSpeedOfBall;
  yAxisPositionOfBall = canvas.height / 2;
  yAxisSpeedOfBall = randomNumber(-6, 6) * (.30 * levelOfDifficulty);
}

// Pause The Game And Set Class Names
function gamePause()
{
  if (gameIsPaused != true)
  {
    game.className = '';
    pausePopUp.className = 'on';
    gameIsPaused = true;
    clearInterval(gameDuration);
  }
}
```

Code Picture 5: Pong – ai.js (Part 2)

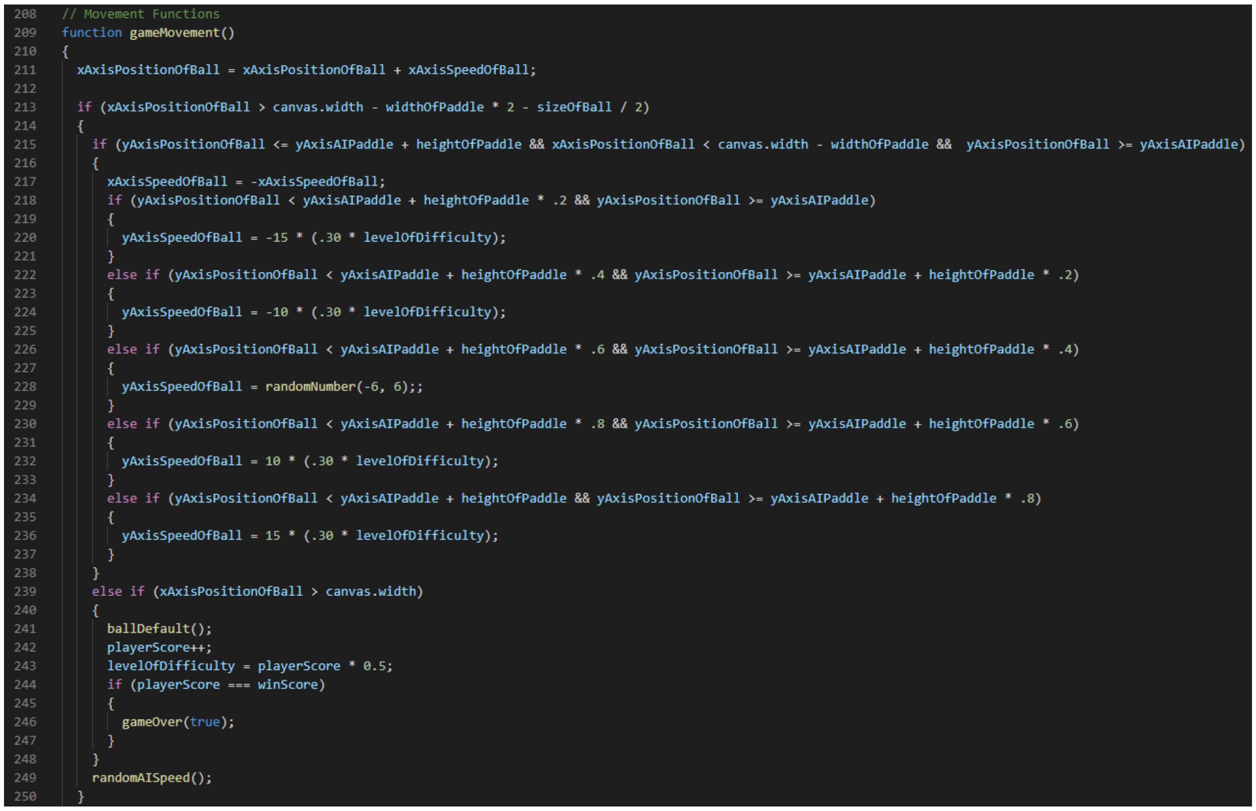
In the Code Picture 5 we see some functions which are connected to the buttons of the game. The first function is the function that is connected to the start button and the third to the pause button. In the second function all the variables of the ball are reset to its default values (this is useful when the game is restarted). In the first and the second functions is defined the class names of some variables and initialize some other variables that are useful for starting the game like in the second function.



```
// DRAW Functions
function Draw()
{
  // Ball Draw
  dimension.clearRect(0, 0, canvas.width, canvas.height);
  dimension.fillStyle = 'rgb(51,153,255)';
  dimension.beginPath();
  dimension.arc(xAxisPositionOfBall, yAxisPositionOfBall, sizeOfBall / 2, 0, Math.PI * 2, true);
  dimension.fill();

  // Player Paddle Draw
  dimension.fillStyle = 'rgb(58, 150, 132)';
  dimension.fillRect(widthOfPaddle, yAxisPlayerPaddle, widthOfPaddle, heightOfPaddle);

  // AI Paddle Draw
  dimension.fillStyle = 'rgb(150, 58, 76)';
  dimension.fillRect(canvas.width - widthOfPaddle - widthOfPaddle, yAxisAIPaddle, widthOfPaddle, heightOfPaddle);

  // Vertical Line Draw
  dimension.strokeStyle = 'rgba(255,255,255,0.6)';
  dimension.beginPath();
  dimension.moveTo(canvas.width / 2, 0);
  dimension.lineTo(canvas.width / 2, canvas.height);
  dimension.stroke();

  // AI Score Draw
  dimension.fillStyle = 'rgba(150, 58, 76,0.5)';
  dimension.font = "200px 'Noto Sans KR', sans-serif";
  dimension.textAlign = "center";
  dimension.fillText(AIScore + "/5", canvas.width * .75, canvas.height / 2 + 75);

  // Player Score Draw
  dimension.fillStyle = 'rgba(58, 150, 132,0.5)';
  dimension.font = "200px 'Noto Sans KR', sans-serif";
  dimension.textAlign = "center";
  dimension.fillText(playerScore + "/5", canvas.width * .25, canvas.height / 2 + 75);
}
```

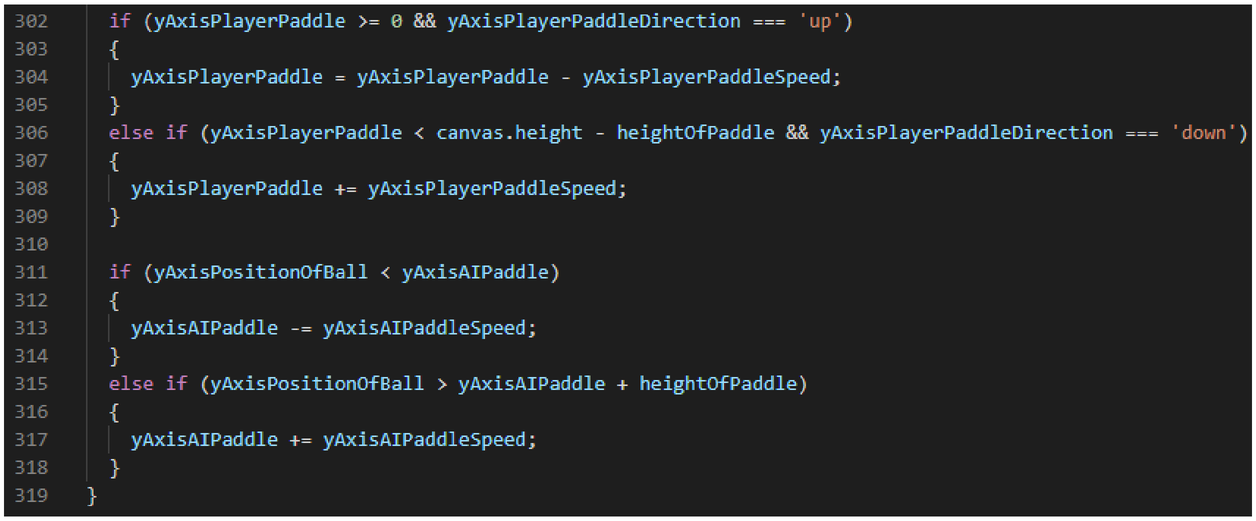
Code Picture 6: Pong – ai.js (Part 3)

In the Code Picture 6 we see a function without which the game could not run. This function is Draw() and is very useful because it displays the objects on the screen. In the beginning is drawed the ball where is made circular, is defined its color and is appointed its position. Next, the Player Paddle is created where the color and position are defined. Also, the AI Paddle is created where the color and position are defined. Below, the vertical line is created that divides the field into two parts where the color of the line is defined and its design is done with moveTo and lineTo commands. The next draw is the draw of the score that appears on the right side, ie on the side of Artificial Intelligence. This defines the color, the font, the font size, the alignment and the position of the score on the canvas. Finally, the player's score is created where the color, the font, the font size, the alignment and the position of the score on the canvas are defined.



```javascript
// Movement Functions
function gameMovement()
{
  xAxisPositionOfBall = xAxisPositionOfBall + xAxisSpeedOfBall;

  if (xAxisPositionOfBall > canvas.width - widthOfPaddle * 2 - sizeOfBall / 2)
  {
    if (yAxisPositionOfBall <= yAxisAIPaddle + heightOfPaddle && xAxisPositionOfBall < canvas.width - widthOfPaddle &&  yAxisPositionOfBall >= yAxisAIPaddle)
    {
      xAxisSpeedOfBall = -xAxisSpeedOfBall;
      if (yAxisPositionOfBall < yAxisAIPaddle + heightOfPaddle * .2 && yAxisPositionOfBall >= yAxisAIPaddle)
      {
        yAxisSpeedOfBall = -15 * (.30 * levelOfDifficulty);
      }
      else if (yAxisPositionOfBall < yAxisAIPaddle + heightOfPaddle * .4 && yAxisPositionOfBall >= yAxisAIPaddle + heightOfPaddle * .2)
      {
        yAxisSpeedOfBall = -10 * (.30 * levelOfDifficulty);
      }
      else if (yAxisPositionOfBall < yAxisAIPaddle + heightOfPaddle * .6 && yAxisPositionOfBall >= yAxisAIPaddle + heightOfPaddle * .4)
      {
        yAxisSpeedOfBall = randomNumber(-6, 6);;
      }
      else if (yAxisPositionOfBall < yAxisAIPaddle + heightOfPaddle * .8 && yAxisPositionOfBall >= yAxisAIPaddle + heightOfPaddle * .6)
      {
        yAxisSpeedOfBall = 10 * (.30 * levelOfDifficulty);
      }
      else if (yAxisPositionOfBall < yAxisAIPaddle + heightOfPaddle && yAxisPositionOfBall >= yAxisAIPaddle + heightOfPaddle * .8)
      {
        yAxisSpeedOfBall = 15 * (.30 * levelOfDifficulty);
      }
    }
    else if (xAxisPositionOfBall > canvas.width)
    {
      ballDefault();
      playerScore++;
      levelOfDifficulty = playerScore * 0.5;
      if (playerScore === winScore)
      {
        gameOver(true);
      }
    }
    randomAISpeed();
  }
```

Code Picture 7: Pong – ai.js (Part 4)



```javascript
    if (yAxisPlayerPaddle >= 0 && yAxisPlayerPaddleDirection === 'up')
    {
      yAxisPlayerPaddle = yAxisPlayerPaddle - yAxisPlayerPaddleSpeed;
    }
    else if (yAxisPlayerPaddle < canvas.height - heightOfPaddle && yAxisPlayerPaddleDirection === 'down')
    {
      yAxisPlayerPaddle += yAxisPlayerPaddleSpeed;
    }

    if (yAxisPositionOfBall < yAxisAIPaddle)
    {
      yAxisAIPaddle -= yAxisAIPaddleSpeed;
    }
    else if (yAxisPositionOfBall > yAxisAIPaddle + heightOfPaddle)
    {
      yAxisAIPaddle += yAxisAIPaddleSpeed;
    }
  }
```

Code Picture 8: Pong – ai.js (Part 5)

In the Code Picture 7 and 8 we see a part of the function without which the game does not make sense. This part is the part of implementing Artificial Intelligence. This part is the most important in the game and the most complicated. Below will be explained the part of the function that is in the code pictures above.

In the Code Picture 7 the position of the ball is determined and it changes according to its speed, so that it knows the program when the scores of both sides increase. In the beginning we see that if the position of the ball (the value) is greater than the value of the width of the canvas minus the width of the Paddle by two minus the size of the ball by two, then we continue to the command located on line 215. If this command is valid then the ball bounces back with the same speed. If the command on line 218 is valid then when the ball hits the top of the AI Paddle then the speed and direction of the ball changes (upwards) and the ball returns to the player (if it was 0 then the ball would turn vertically). Then if the command on line 222 is valid then when the ball hits before the middle of the AI Paddle then the speed and direction of the ball changes (upwards) and the ball returns to the player (if it was 0 then the ball would turn vertically). Also, if the command on line 226 is valid then when the ball hits the middle of the AI Paddle then the speed and direction of the ball changes with a random value from -6 to 6 and the ball returns to the player (if it was 0 then the ball would turn vertically). Next, if the command on line 230 is valid then when the ball hits just after the middle of the AI Paddle then the speed and direction of the ball (downwards) changes and the ball turns to the player (if it was 0 then the ball would turn vertically). However, if the command on line 234 is valid then when the ball hits the bottom of the AI Paddle then the speed and direction of the ball (down) changes and the ball returns to the player (if it was 0 then the ball would turn vertically). Finally, if the position of the ball is greater than the width of the canvas then the player scores (one point is added to the player). If the player's score is equal to five then he is the winner of the game.

In the Code Picture 8, the movement of the two paddles, ie the player and the AI, is implemented. In the first "If" the direction of the player's paddle is implemented. If it is up then the paddle moves upwards, while in "else if" the exact opposite happens and so it moves downwards. The second "If" implements the AI paddle direction. If the position of the ball is less than the value of the AIPaddle on the Y axis, then the paddle moves upwards, while in "else if" the exact opposite happens and thus moves downwards.

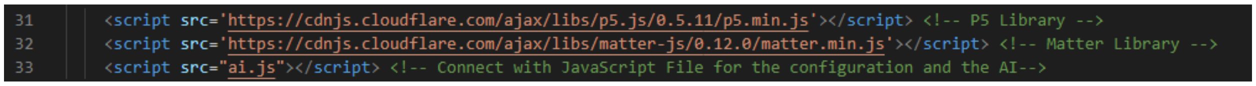
6.2) Soccer Car Game (Player versus Artificial Intelligence)

The purpose of this game is for the player to win against the Artificial Intelligence. So, in order to win the player must reach first the five points. If he does not reach them first, then Artificial Intelligence wins the game. The player can play this game many times.
Because all games that are fps (i.e., Pong, Soccer Car and Snake) have the same structure in the index.html file and in the stylesheet.css file, it does not need to be analyzed again below. Some minor differences will be analyzed below, which do not spoil the front-end design.

6.2.1) File: index.html

The code of this file is the same as the Pong code and does not need to be further analyzed. The only difference with the pong code is that in the <script></script> tag, in addition to the integration of the main JavaScript file where the game is implemented, there is also the integration of some libraries that are necessary for the game. These libraries are p5.min.js and matter.min.js.

See Index File Of Pong for more information.



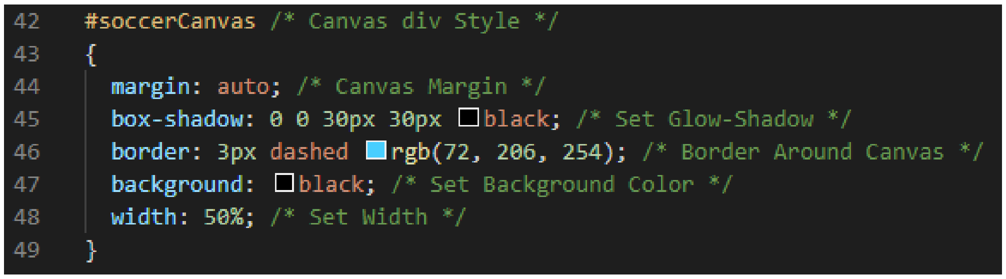
```html
    <script src='https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.5.11/p5.min.js'></script> <!-- P5 Library -->
    <script src='https://cdnjs.cloudflare.com/ajax/libs/matter-js/0.12.0/matter.min.js'></script> <!-- Matter Library -->
    <script src="ai.js"></script> <!-- Connect with JavaScript File for the configuration and the AI-->
```

Code Picture 9: The addition of the code in index.html of Soccer Car game.

6.2.2) File: stylesheet.css

The code of this file is the same as the Pong code and does not need to be further analyzed. The only difference with the pong code is that in the name of canvas is changed from "#main > #game > #ponggameCanvas" to #soccerCanvas.

See Stylesheet File Of Pong for more information.

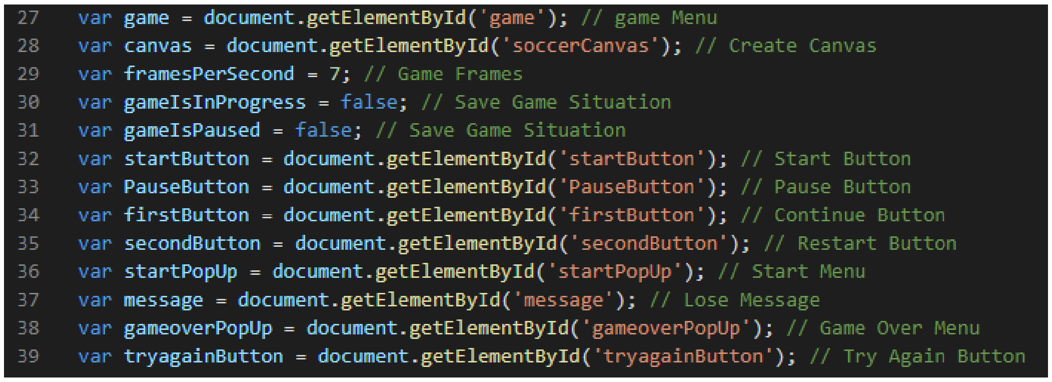


```css
#soccerCanvas /* Canvas div Style */
{
  margin: auto; /* Canvas Margin */
  box-shadow: 0 0 30px 30px black; /* Set Glow-Shadow */
  border: 3px dashed rgb(72, 206, 254); /* Border Around Canvas */
  background: black; /* Set Background Color */
  width: 50%; /* Set Width */
}
```

Code Picture 10: The rename from "#main > #game > #ponggameCanvas" to #soccerCanvas.

6.2.3) File: ai.js

In this file, the game is implemented. Below, is analyzed a part of the file ai.js.

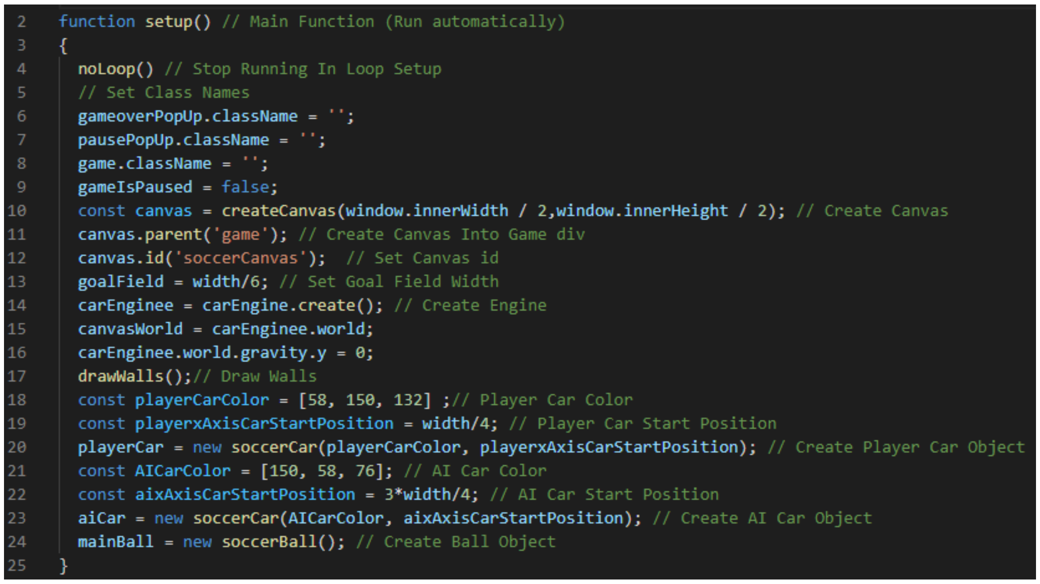


```javascript
var game = document.getElementById('game'); // game Menu
var canvas = document.getElementById('soccerCanvas'); // Create Canvas
var framesPerSecond = 7; // Game Frames
var gameIsInProgress = false; // Save Game Situation
var gameIsPaused = false; // Save Game Situation
var startButton = document.getElementById('startButton'); // Start Button
var PauseButton = document.getElementById('PauseButton'); // Pause Button
var firstButton = document.getElementById('firstButton'); // Continue Button
var secondButton = document.getElementById('secondButton'); // Restart Button
var startPopUp = document.getElementById('startPopUp'); // Start Menu
var message = document.getElementById('message'); // Lose Message
var gameoverPopUp = document.getElementById('gameoverPopUp'); // Game Over Menu
var tryagainButton = document.getElementById('tryagainButton'); // Try Again Button
```

Code Picture 11: Soccer Car – ai.js (Part 1)

In this file, the game is implemented. In Code Picture 11 we see that all the variables are defined. Some of these variables are framePerSecond, because it is a fps game, gameIsInProgress, where the value is stored whether the game is in progress or not, gameIsInPaused, where the value is stored whether the game is in paused or not.

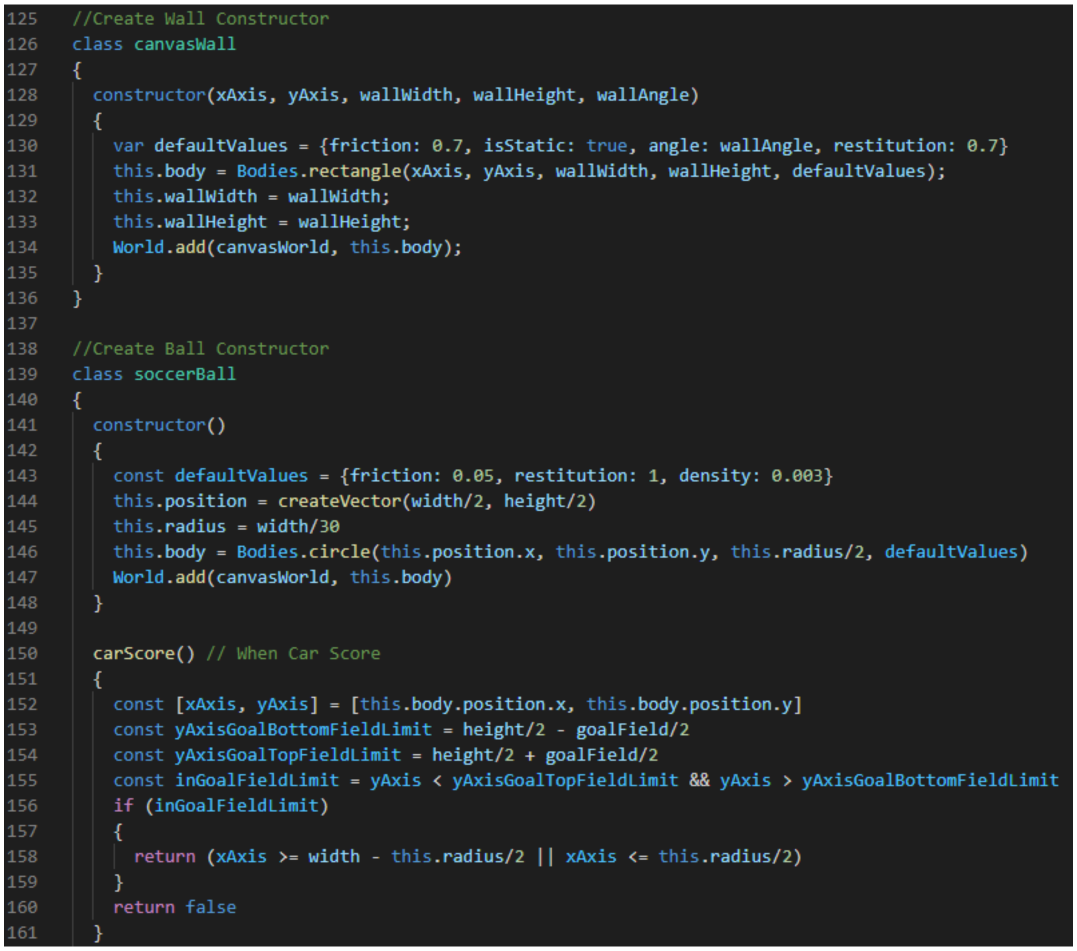
Then the elements that have id are connected with this file and are saved so they can be programmed. Some of them are startButton, which is the game start button and is stored in the variable startButton, PauseButton, which is the game pause button and is stored in the variable PauseButton, the message, which is the message that appears when game ends and is saved in the variable message.



```javascript
function setup() // Main Function (Run automatically)
{
  noLoop() // Stop Running In Loop Setup
  // Set Class Names
  gameoverPopUp.className = '';
  pausePopUp.className = '';
  game.className = '';
  gameIsPaused = false;
  const canvas = createCanvas(window.innerWidth / 2,window.innerHeight / 2); // Create Canvas
  canvas.parent('game'); // Create Canvas Into Game div
  canvas.id('soccerCanvas');  // Set Canvas id
  goalField = width/6; // Set Goal Field Width
  carEnginee = carEngine.create(); // Create Engine
  canvasWorld = carEnginee.world;
  carEnginee.world.gravity.y = 0;
  drawWalls();// Draw Walls
  const playerCarColor = [58, 150, 132] ;// Player Car Color
  const playerxAxisCarStartPosition = width/4; // Player Car Start Position
  playerCar = new soccerCar(playerCarColor, playerxAxisCarStartPosition); // Create Player Car Object
  const AICarColor = [150, 58, 76]; // AI Car Color
  const aixAxisCarStartPosition = 3*width/4; // AI Car Start Position
  aiCar = new soccerCar(AICarColor, aixAxisCarStartPosition); // Create AI Car Object
  mainBall = new soccerBall(); // Create Ball Object
}
```

Code Picture 12: Soccer Car – ai.js (Part 2)

The function of Code Picture 12 is the function which runs automatically as soon as the game starts. At the beginning, the names of the classes of variables are specified, which store the values from the index.html file. In this function the whole game is tuned, that is, the gravity is determined, the Engine, which come from the libraries that were integrated, the goal field is defined, the objects of the cars are created, the color of the cars is defined, the walls are created and so on.

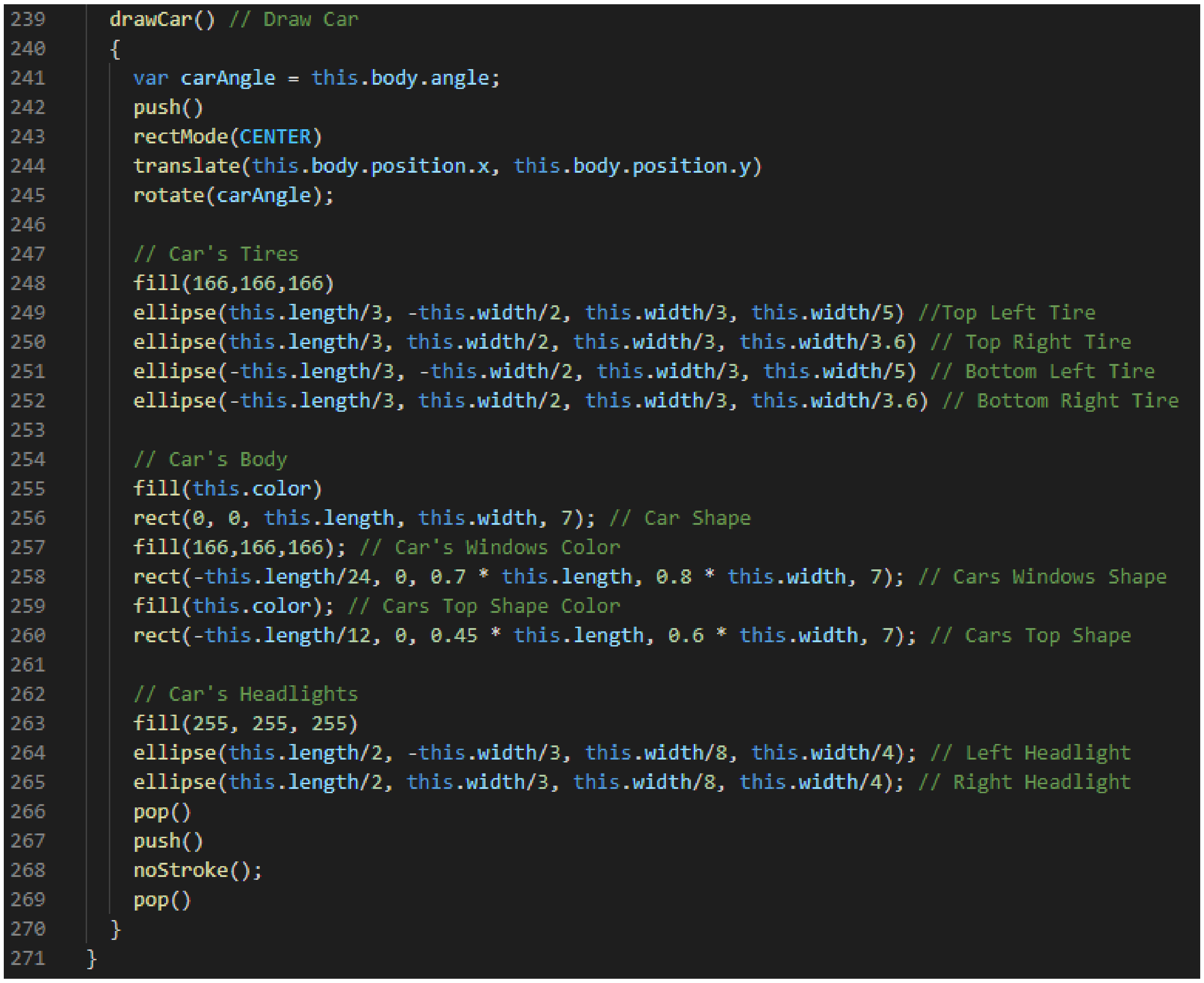


```
//Create Wall Constructor
class canvasWall
{
  constructor(xAxis, yAxis, wallWidth, wallHeight, wallAngle)
  {
    var defaultValues = {friction: 0.7, isStatic: true, angle: wallAngle, restitution: 0.7}
    this.body = Bodies.rectangle(xAxis, yAxis, wallWidth, wallHeight, defaultValues);
    this.wallWidth = wallWidth;
    this.wallHeight = wallHeight;
    World.add(canvasWorld, this.body);
  }
}

//Create Ball Constructor
class soccerBall
{
  constructor()
  {
    const defaultValues = {friction: 0.05, restitution: 1, density: 0.003}
    this.position = createVector(width/2, height/2)
    this.radius = width/30
    this.body = Bodies.circle(this.position.x, this.position.y, this.radius/2, defaultValues)
    World.add(canvasWorld, this.body)
  }

  carScore() // When Car Score
  {
    const [xAxis, yAxis] = [this.body.position.x, this.body.position.y]
    const yAxisGoalBottomFieldLimit = height/2 - goalField/2
    const yAxisGoalTopFieldLimit = height/2 + goalField/2
    const inGoalFieldLimit = yAxis < yAxisGoalTopFieldLimit && yAxis > yAxisGoalBottomFieldLimit
    if (inGoalFieldLimit)
    {
      return (xAxis >= width - this.radius/2 || xAxis <= this.radius/2)
    }
    return false
  }
```

Code Picture 13: Soccer Car – ai.js (Part 3)

This game uses object-oriented programming and follows the object-oriented paradigm. At the beginning of Code Picture 13 is the implementation of the constructor of the walls and then the implementation of the constructor of the soccer ball. Thus, in the first class, in the constructor, is defined the default values that the walls will have. Next, is defined their shape with the help of the Matter library and define their width and height.

In the following class, the constructor determines as above the default values of the ball, how circular it will be, its position and shape. The carScore (), is the function that checks if the ball has passed the field limits which is specified from the variables "yAxisTopFieldlLimit" and "yAxisTopFieldlLimit". If the ball is between these values then there is a "goal", if not then the ball bounces back.



```
  drawCar() // Draw Car
  {
    var carAngle = this.body.angle;
    push()
    rectMode(CENTER)
    translate(this.body.position.x, this.body.position.y)
    rotate(carAngle);

    // Car's Tires
    fill(166,166,166)
    ellipse(this.length/3, -this.width/2, this.width/3, this.width/5) //Top Left Tire
    ellipse(this.length/3, this.width/2, this.width/3, this.width/3.6) // Top Right Tire
    ellipse(-this.length/3, -this.width/2, this.width/3, this.width/5) // Bottom Left Tire
    ellipse(-this.length/3, this.width/2, this.width/3, this.width/3.6) // Bottom Right Tire

    // Car's Body
    fill(this.color)
    rect(0, 0, this.length, this.width, 7); // Car Shape
    fill(166,166,166); // Car's Windows Color
    rect(-this.length/24, 0, 0.7 * this.length, 0.8 * this.width, 7); // Cars Windows Shape
    fill(this.color); // Cars Top Shape Color
    rect(-this.length/12, 0, 0.45 * this.length, 0.6 * this.width, 7); // Cars Top Shape

    // Car's Headlights
    fill(255, 255, 255)
    ellipse(this.length/2, -this.width/3, this.width/8, this.width/4); // Left Headlight
    ellipse(this.length/2, this.width/3, this.width/8, this.width/4); // Right Headlight
    pop()
    push()
    noStroke();
    pop()
  }
}
```

Code Picture 14: Soccer Car – ai.js (Part 4)

In Code Picture 14, we see the drawCar() function, which forms the car. In the beginning the variable is created which will store the degrees that the car will turn. The car rectangles are made in the center and the car rotate according to the variable, which is changed. Also, in the beginning the tires of the car are created with the ellipse commands where it create circles. The car lights are fitted in the same way. Finally, the body of the car is created with the help of the rect command which makes squares. The fill() command fills the area where it is called with color.

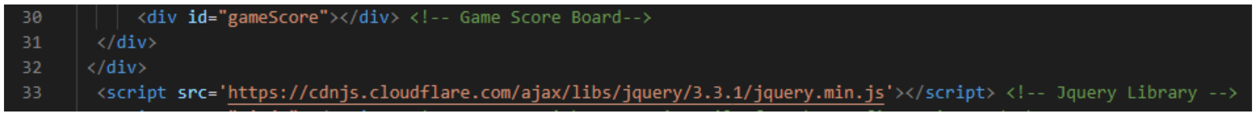
6.3) Snake Game (Player versus Artificial Intelligence)

The purpose of this game is for the player to win against the Artificial Intelligence. So, in order to win the player must eat as much food as possible and grow up as much as possible, not to hit the walls and not to hit himself or the Artificial Intelligence snake. If he does not eat so much food, hit the walls and hit himself or Artificial Intelligence snake, then Artificial Intelligence wins the game. The player can play this game many times.
Because all games that are fps (i.e., Pong, Soccer Car and Snake) have the same structure in the index.html file and in the stylesheet.css file, it does not need to be analyzed again below. Some minor differences will be analyzed below, which do not spoil the front-end design.

6.3.1) File: index.html

The code of this file is the same as the Pong code and does not need to be further analyzed. The only differences with the pong code are that in the <script></script> tag, in addition to the integration of the main JavaScript file where the game is implemented, there is also the integration of a library that are necessary for the game. This library is jquery.min.js. The second difference is that another div is added which is useful for the game scoreboard. This addition was made because the score was not incorporated into the canvas, because the game is not divided into two parts.

See Index File Of Pong for more information.

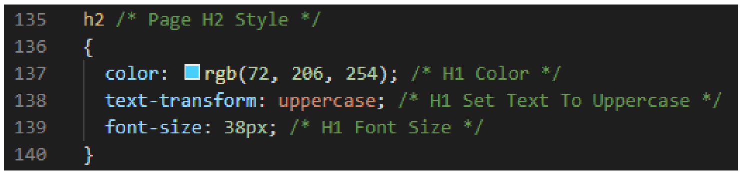


```
      <div id="gameScore"></div> <!-- Game Score Board-->
    </div>
  </div>
  <script src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js'></script> <!-- Jquery Library -->
```

Code Picture 15: Addition of the div and library.

6.3.2) File: stylesheet.css

The code of this file is the same as the Pong code and does not need to be further analyzed. The only differences with the pong code are that in the name of canvas is changed from #ponggameCanvas to #gameCanvas and the appearance addition of the h2 header.



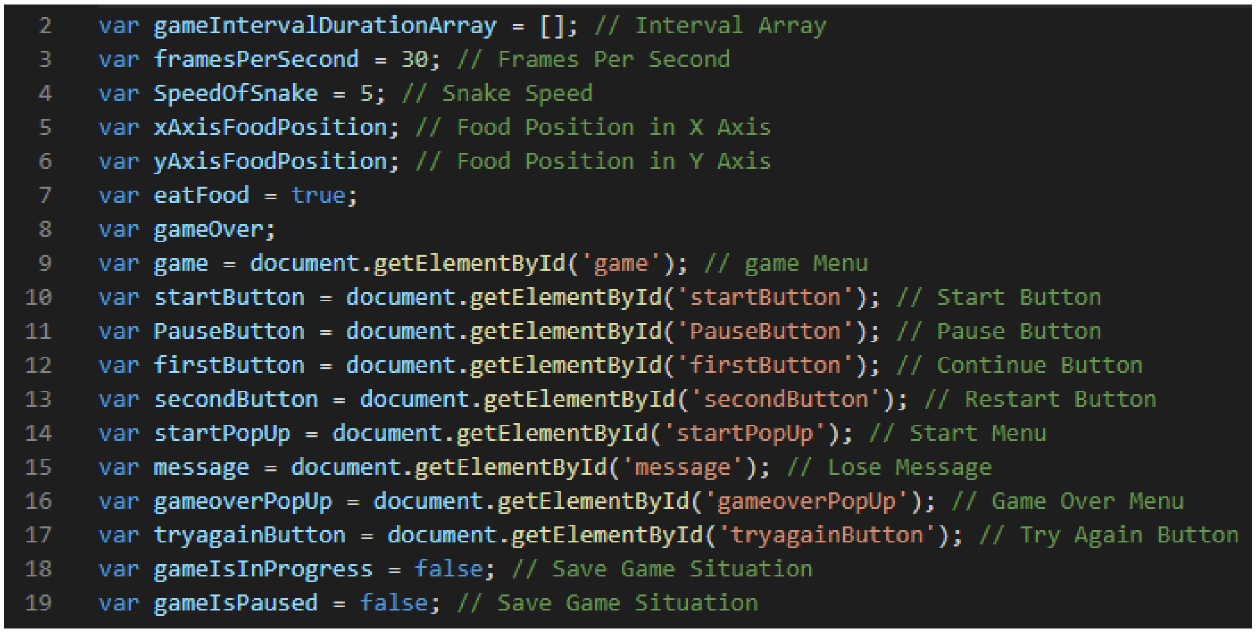
```
h2 /* Page H2 Style */
{
  color: rgb(72, 206, 254); /* H1 Color */
  text-transform: uppercase; /* H1 Set Text To Uppercase */
  font-size: 38px; /* H1 Font Size */
}
```

Code Picture 16: Addition of the h2 header appearance.

See Stylesheet File Of Pong for more information.

6.3.3) File: ai.js

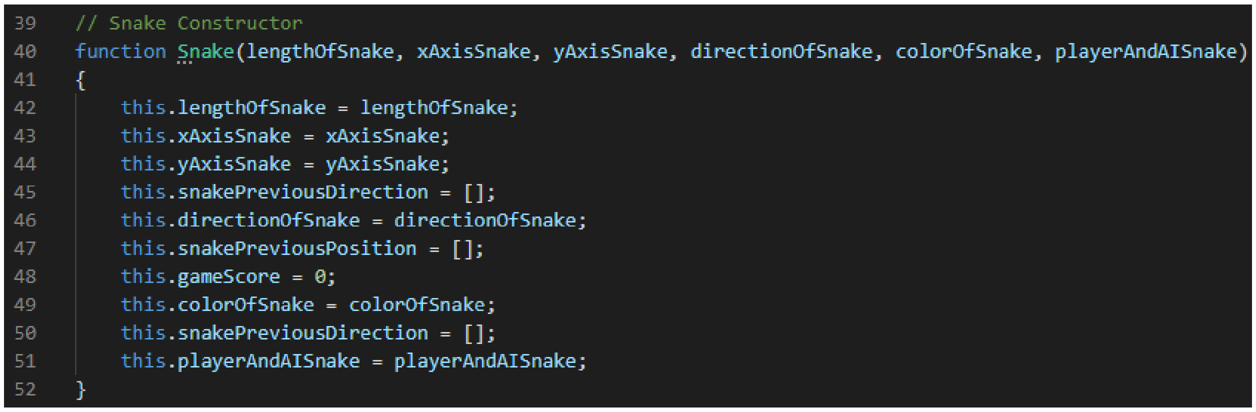
In this file, the game is implemented. Below, is analyzed a part of the file ai.js.



```javascript
var gameIntervalDurationArray = []; // Interval Array
var framesPerSecond = 30; // Frames Per Second
var SpeedOfSnake = 5; // Snake Speed
var xAxisFoodPosition; // Food Position in X Axis
var yAxisFoodPosition; // Food Position in Y Axis
var eatFood = true;
var gameOver;
var game = document.getElementById('game'); // game Menu
var startButton = document.getElementById('startButton'); // Start Button
var PauseButton = document.getElementById('PauseButton'); // Pause Button
var firstButton = document.getElementById('firstButton'); // Continue Button
var secondButton = document.getElementById('secondButton'); // Restart Button
var startPopUp = document.getElementById('startPopUp'); // Start Menu
var message = document.getElementById('message'); // Lose Message
var gameoverPopUp = document.getElementById('gameoverPopUp'); // Game Over Menu
var tryagainButton = document.getElementById('tryagainButton'); // Try Again Button
var gameIsInProgress = false; // Save Game Situation
var gameIsPaused = false; // Save Game Situation
```

Code Picture 17: Snake – ai.js (Part 1)

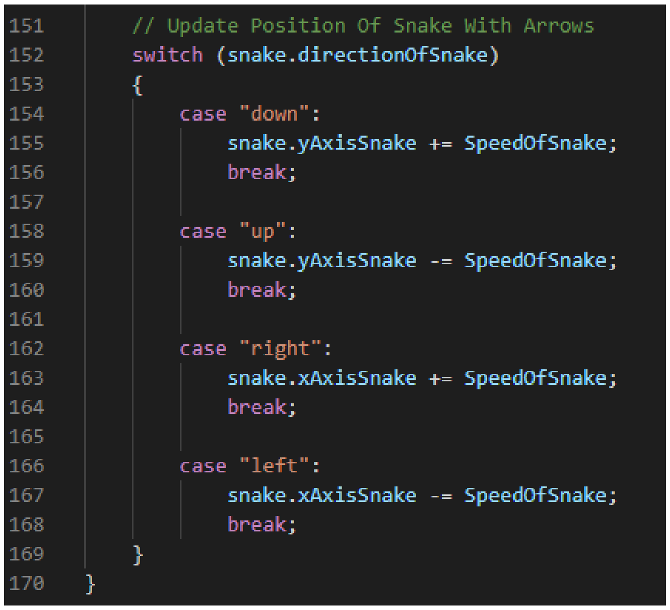
In Code Picture 17 we see that all the variables are defined. Some of these variables are framePerSecond, because it is a fps game, speed of the Snake, the food position in X and Y Axis and so on. All the names of the variables are written in such a way that only from the name the programmer knows what it is and what this variable does. Then the elements that have id are connected with this file and are saved so they can be programmed. Some of them are startButton, which is the game start button and is stored in the variable startButton, PauseButton, which is the game pause button and is stored in the variable PauseButton, the message, which is the message that appears when game ends and is saved in the variable message.



```javascript
// Snake Constructor
function Snake(lengthOfSnake, xAxisSnake, yAxisSnake, directionOfSnake, colorOfSnake, playerAndAISnake)
{
    this.lengthOfSnake = lengthOfSnake;
    this.xAxisSnake = xAxisSnake;
    this.yAxisSnake = yAxisSnake;
    this.snakePreviousDirection = [];
    this.directionOfSnake = directionOfSnake;
    this.snakePreviousPosition = [];
    this.gameScore = 0;
    this.colorOfSnake = colorOfSnake;
    this.snakePreviousDirection = [];
    this.playerAndAISnake = playerAndAISnake;
}
```

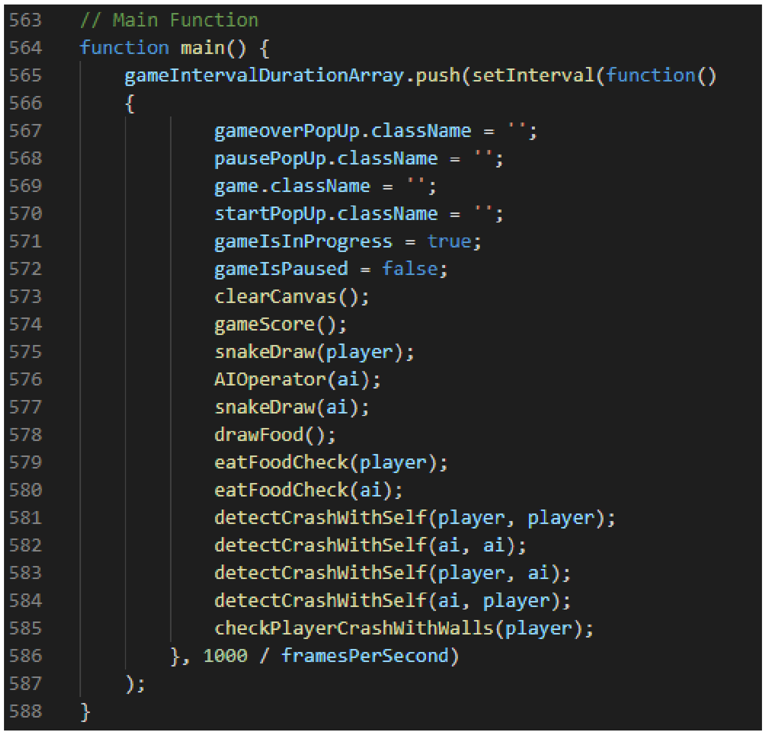
Code Picture 18: Snake – ai.js (Part 2)

At the beginning of Code Picture 18 is the implementation of the Snake function. This function creates the Snakes and sets some values. Thus, this function determines the size of the snake, its direction, the score of the game, its color, the winner, the previous positions of the snake and so on.



```
        // Update Position Of Snake With Arrows
        switch (snake.directionOfSnake)
        {
            case "down":
                snake.yAxisSnake += SpeedOfSnake;
                break;

            case "up":
                snake.yAxisSnake -= SpeedOfSnake;
                break;

            case "right":
                snake.xAxisSnake += SpeedOfSnake;
                break;

            case "left":
                snake.xAxisSnake -= SpeedOfSnake;
                break;
        }
    }
```

Code Picture 19: Snake – ai.js (Part 3)

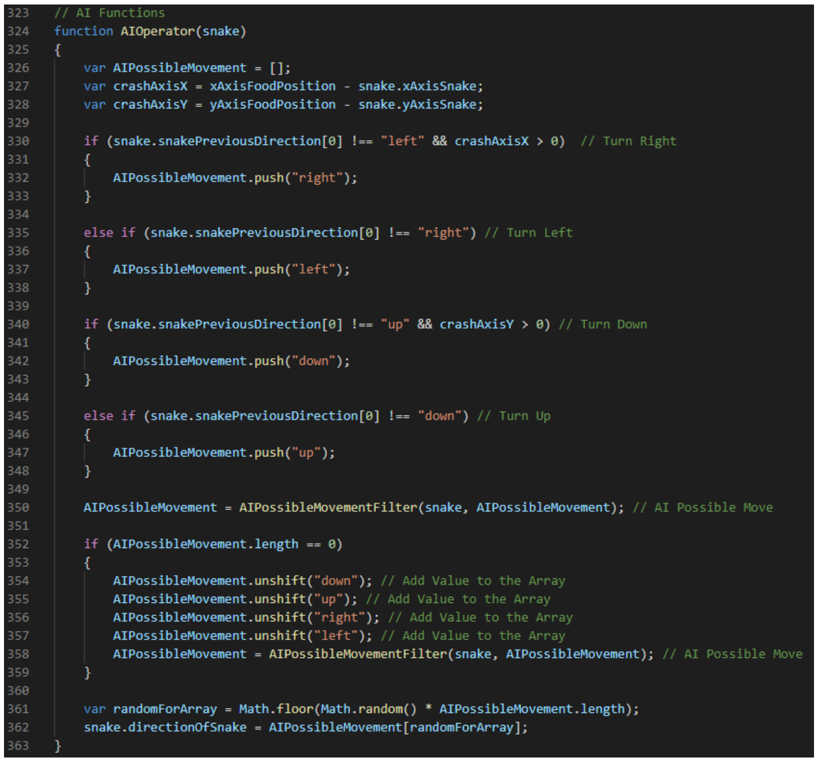


```
// Main Function
function main() {
    gameIntervalDurationArray.push(setInterval(function()
    {
            gameoverPopUp.className = '';
            pausePopUp.className = '';
            game.className = '';
            startPopUp.className = '';
            gameIsInProgress = true;
            gameIsPaused = false;
            clearCanvas();
            gameScore();
            snakeDraw(player);
            AIOperator(ai);
            snakeDraw(ai);
            drawFood();
            eatFoodCheck(player);
            eatFoodCheck(ai);
            detectCrashWithSelf(player, player);
            detectCrashWithSelf(ai, ai);
            detectCrashWithSelf(player, ai);
            detectCrashWithSelf(ai, player);
            checkPlayerCrashWithWalls(player);
        }, 1000 / framesPerSecond)
    );
}
```

Code Picture 20: Snake – ai.js (Part 4)

In Code Picture 19 we see the commands by which the player changes the direction of the snake. So pressing the "down arrow" then the value of the Y axis increases and the snake turns down. Pressing the "up arrow" then the value of the Y axis decreases and the snake turns upwards. Pressing the "right arrow" then increases the value of the X-axis and the snake turns to the right. Pressing the "left arrow" then the value of the X axis decreases and the snake turns to the left.

In Code Picture 20 we see the main function which is the function that starts the game. This includes all the variables and functions that are needed for the gameplay. The class names of the variables that store the ids elements of the file index.html are specified. It is called the ClearCanvas() function where it clears the canvas, the gameScore() function where is the scoreboard, the snakeDraw() where the snakes form (color etc and appear on the screen), the drawFood() where the food appears on the canvas, the AIOperator where it states the directions for the movement of Artificial Intelligence Snake and other functions.



```
// AI Functions
function AIOperator(snake)
{
    var AIPossibleMovement = [];
    var crashAxisX = xAxisFoodPosition - snake.xAxisSnake;
    var crashAxisY = yAxisFoodPosition - snake.yAxisSnake;

    if (snake.snakePreviousDirection[0] !== "left" && crashAxisX > 0)  // Turn Right
    {
        AIPossibleMovement.push("right");
    }

    else if (snake.snakePreviousDirection[0] !== "right") // Turn Left
    {
        AIPossibleMovement.push("left");
    }

    if (snake.snakePreviousDirection[0] !== "up" && crashAxisY > 0) // Turn Down
    {
        AIPossibleMovement.push("down");
    }

    else if (snake.snakePreviousDirection[0] !== "down") // Turn Up
    {
        AIPossibleMovement.push("up");
    }

    AIPossibleMovement = AIPossibleMovementFilter(snake, AIPossibleMovement); // AI Possible Move

    if (AIPossibleMovement.length == 0)
    {
        AIPossibleMovement.unshift("down"); // Add Value to the Array
        AIPossibleMovement.unshift("up"); // Add Value to the Array
        AIPossibleMovement.unshift("right"); // Add Value to the Array
        AIPossibleMovement.unshift("left"); // Add Value to the Array
        AIPossibleMovement = AIPossibleMovementFilter(snake, AIPossibleMovement); // AI Possible Move
    }

    var randomForArray = Math.floor(Math.random() * AIPossibleMovement.length);
    snake.directionOfSnake = AIPossibleMovement[randomForArray];
}
```

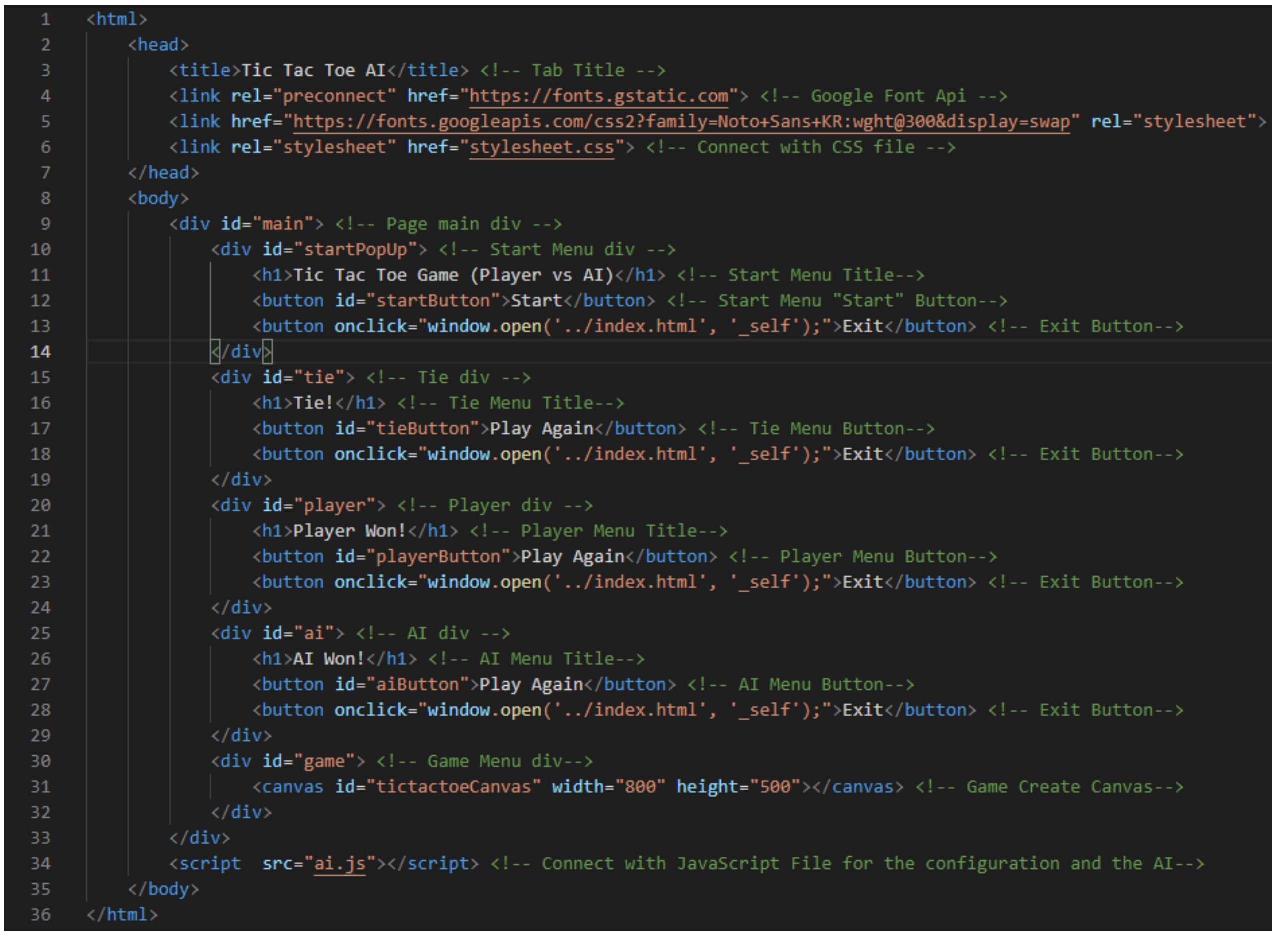
Code Picture 21: Snake – ai.js (Part 5)

In Code Picture 21 we see the changes in motion made of the Artificial Intelligence Snake. If the previous direction of Artificial Intelligence Snake was not "left" and the crash on the X axis is greater than 0 then the value "right" is stored in the AIPossibleMovement table. Below, if the previous direction of Artificial Intelligence Snake was not "right" then in the AIPossibleMovement table the value "left" is stored. If the previous direction of Artificial Intelligence Snake was not "up" and the crash on the Y axis is greater than 0 then the value "down" is stored in the AIPossibleMovement table. If the previous Artificial Intelligence Snake direction was not "down" then the value "up" is stored in the AIPossibleMovement table. Finally, the AIPossibleMovementFilter function is called and controls the values of AIPossibleMovement, ie if with the values given by AIPossibleMovement the Artificial Intelligence Snake crash or not, and thus returns the best move to avoid crash.

6.4) Tic Tac Toe Game (Player versus Artificial Intelligence)

The purpose of this game is for the player to win against the Artificial Intelligence. So, in order to win the player must achieve three "X" in the horizontal axis or vertical axis or diagonal axis. If he does not achieve three "X" in the horizontal axis or vertical axis or diagonal axis,then Artificial Intelligence wins the game or the game is tied. The player can play this game many times. Because this game is not an fps game, some things that were useful for the previous three games which are fps games were not integrated.

6.4.1) File: index.html

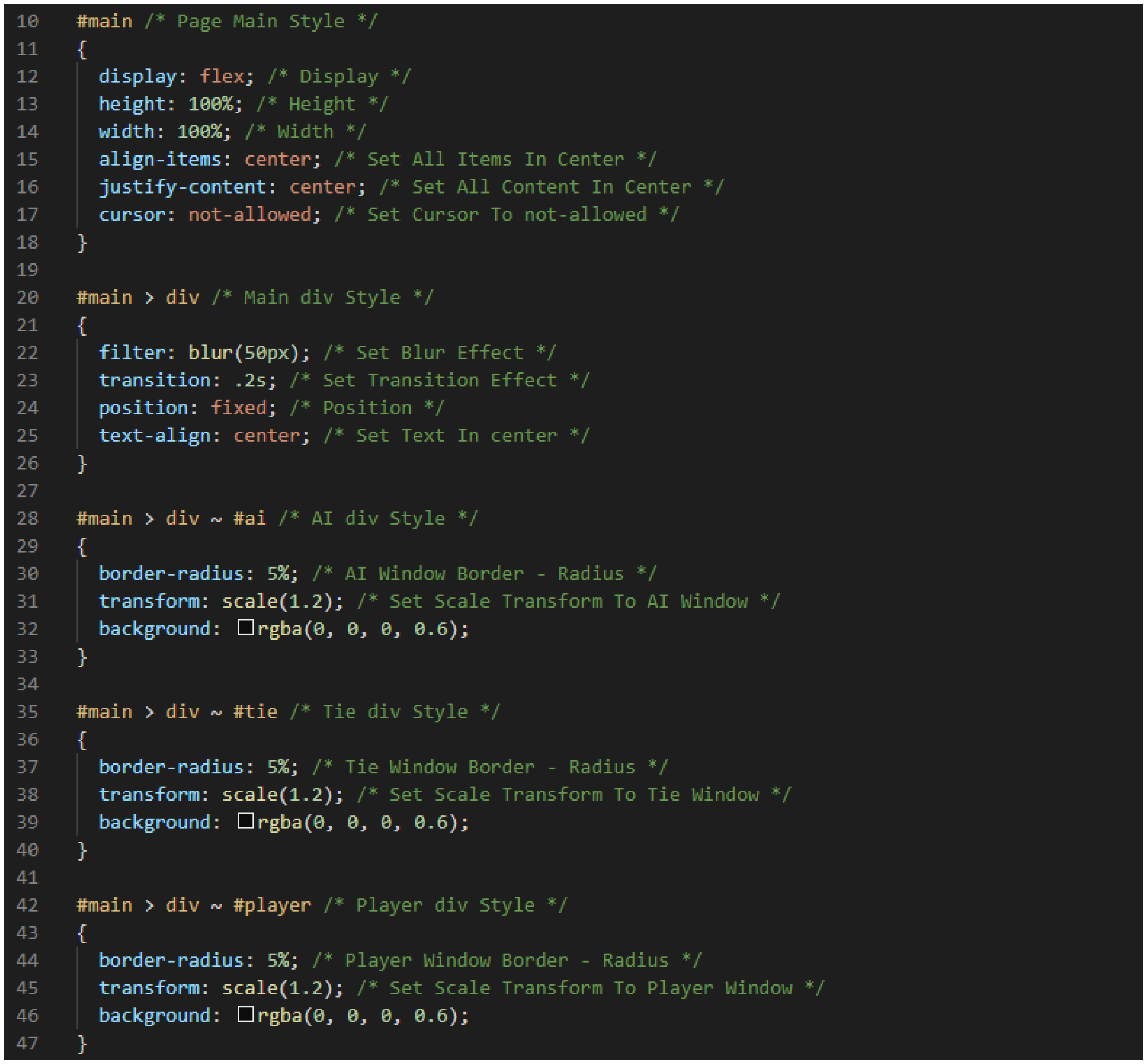


```html
<html>
    <head>
        <title>Tic Tac Toe AI</title> <!-- Tab Title -->
        <link rel="preconnect" href="https://fonts.gstatic.com"> <!-- Google Font Api -->
        <link href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@300&display=swap" rel="stylesheet">
        <link rel="stylesheet" href="stylesheet.css"> <!-- Connect with CSS file -->
    </head>
    <body>
        <div id="main"> <!-- Page main div -->
            <div id="startPopUp"> <!-- Start Menu div -->
                <h1>Tic Tac Toe Game (Player vs AI)</h1> <!-- Start Menu Title-->
                <button id="startButton">Start</button> <!-- Start Menu "Start" Button-->
                <button onclick="window.open('../index.html', '_self');">Exit</button> <!-- Exit Button-->
            </div>
            <div id="tie"> <!-- Tie div -->
                <h1>Tie!</h1> <!-- Tie Menu Title-->
                <button id="tieButton">Play Again</button> <!-- Tie Menu Button-->
                <button onclick="window.open('../index.html', '_self');">Exit</button> <!-- Exit Button-->
            </div>
            <div id="player"> <!-- Player div -->
                <h1>Player Won!</h1> <!-- Player Menu Title-->
                <button id="playerButton">Play Again</button> <!-- Player Menu Button-->
                <button onclick="window.open('../index.html', '_self');">Exit</button> <!-- Exit Button-->
            </div>
            <div id="ai"> <!-- AI div -->
                <h1>AI Won!</h1> <!-- AI Menu Title-->
                <button id="aiButton">Play Again</button> <!-- AI Menu Button-->
                <button onclick="window.open('../index.html', '_self');">Exit</button> <!-- Exit Button-->
            </div>
            <div id="game"> <!-- Game Menu div-->
                <canvas id="tictactoeCanvas" width="800" height="500"></canvas> <!-- Game Create Canvas-->
            </div>
        </div>
        <script  src="ai.js"></script> <!-- Connect with JavaScript File for the configuration and the AI-->
    </body>
</html>
```

Code Picture 22: Tic Tac Toe - Index.html File

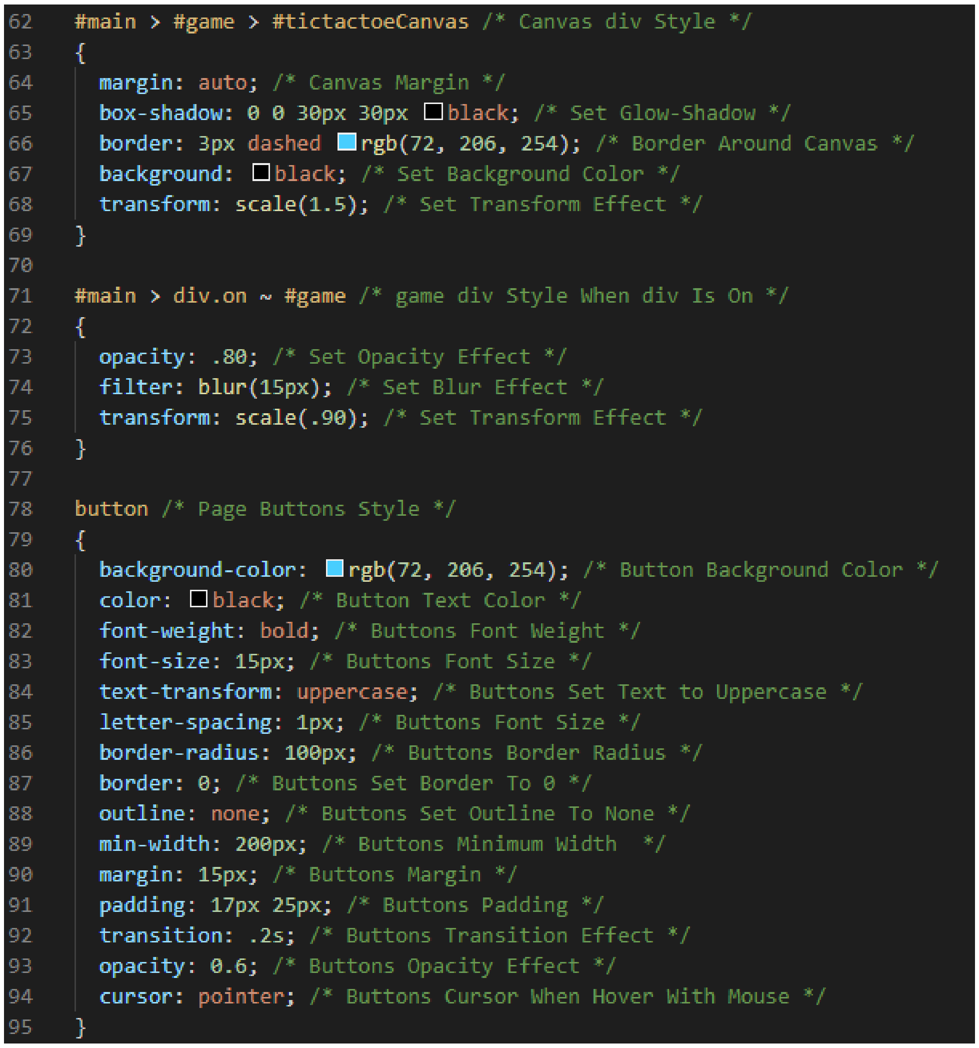
In my code I have integrated a google font which is free, and I have linked to this the stylesheet.css file that contains the game's appearance. The main code of the file is written between the <body> and </body> tags. So, the game is divided into six divs that the "id" makes them unique. In the main div are integrated all the other divs, i.e., the startPopUp, which contains the Start and Exit buttons, the "tie", which contains the pop-up that appears when the game ends (the game is tied) and the two buttons Play Again and Exit, the "player", which contains the pop-up that appears when the game ends (the player is the winner) and the two buttons Play Again and Exit, the "ai", which contains the pop-up that appears when the game ends (the Artificial Intelligence is the winner) and the two buttons Play Again and Exit. Finally, the div with id "game" contained in the game is integrated. Any change to the main div will change the rest of the divs. At the bottom of the <body> before closing this tag, there is the <script></script> tag in which the developer can write a JavaScript code. In my case, because the file would be too large, the JavaScript code was written to a separate file and through this tag the index.html file was linked to the ai.js file.

6.4.2) File: stylesheet.css



```css
#main /* Page Main Style */
{
  display: flex; /* Display */
  height: 100%; /* Height */
  width: 100%; /* Width */
  align-items: center; /* Set All Items In Center */
  justify-content: center; /* Set All Content In Center */
  cursor: not-allowed; /* Set Cursor To not-allowed */
}

#main > div /* Main div Style */
{
  filter: blur(50px); /* Set Blur Effect */
  transition: .2s; /* Set Transition Effect */
  position: fixed; /* Position */
  text-align: center; /* Set Text In center */
}

#main > div ~ #ai /* AI div Style */
{
  border-radius: 5%; /* AI Window Border - Radius */
  transform: scale(1.2); /* Set Scale Transform To AI Window */
  background: rgba(0, 0, 0, 0.6);
}

#main > div ~ #tie /* Tie div Style */
{
  border-radius: 5%; /* Tie Window Border - Radius */
  transform: scale(1.2); /* Set Scale Transform To Tie Window */
  background: rgba(0, 0, 0, 0.6);
}

#main > div ~ #player /* Player div Style */
{
  border-radius: 5%; /* Player Window Border - Radius */
  transform: scale(1.2); /* Set Scale Transform To Player Window */
  background: rgba(0, 0, 0, 0.6);
}
```

Code Picture 23: Tic Tac Toe – stylesheet.css File (Part 1)



```css
#main > #game > #tictactoeCanvas /* Canvas div Style */
{
  margin: auto; /* Canvas Margin */
  box-shadow: 0 0 30px 30px black; /* Set Glow-Shadow */
  border: 3px dashed rgb(72, 206, 254); /* Border Around Canvas */
  background: black; /* Set Background Color */
  transform: scale(1.5); /* Set Transform Effect */
}

#main > div.on ~ #game /* game div Style When div Is On */
{
  opacity: .80; /* Set Opacity Effect */
  filter: blur(15px); /* Set Blur Effect */
  transform: scale(.90); /* Set Transform Effect */
}

button /* Page Buttons Style */
{
  background-color: rgb(72, 206, 254); /* Button Background Color */
  color: black; /* Button Text Color */
  font-weight: bold; /* Buttons Font Weight */
  font-size: 15px; /* Buttons Font Size */
  text-transform: uppercase; /* Buttons Set Text to Uppercase */
  letter-spacing: 1px; /* Buttons Font Size */
  border-radius: 100px; /* Buttons Border Radius */
  border: 0; /* Buttons Set Border To 0 */
  outline: none; /* Buttons Set Outline To None */
  min-width: 200px; /* Buttons Minimum Width  */
  margin: 15px; /* Buttons Margin */
  padding: 17px 25px; /* Buttons Padding */
  transition: .2s; /* Buttons Transition Effect */
  opacity: 0.6; /* Buttons Opacity Effect */
  cursor: pointer; /* Buttons Cursor When Hover With Mouse */
}
```

Code Picture 24: Tic Tac Toe – stylesheet.css File (Part 2)

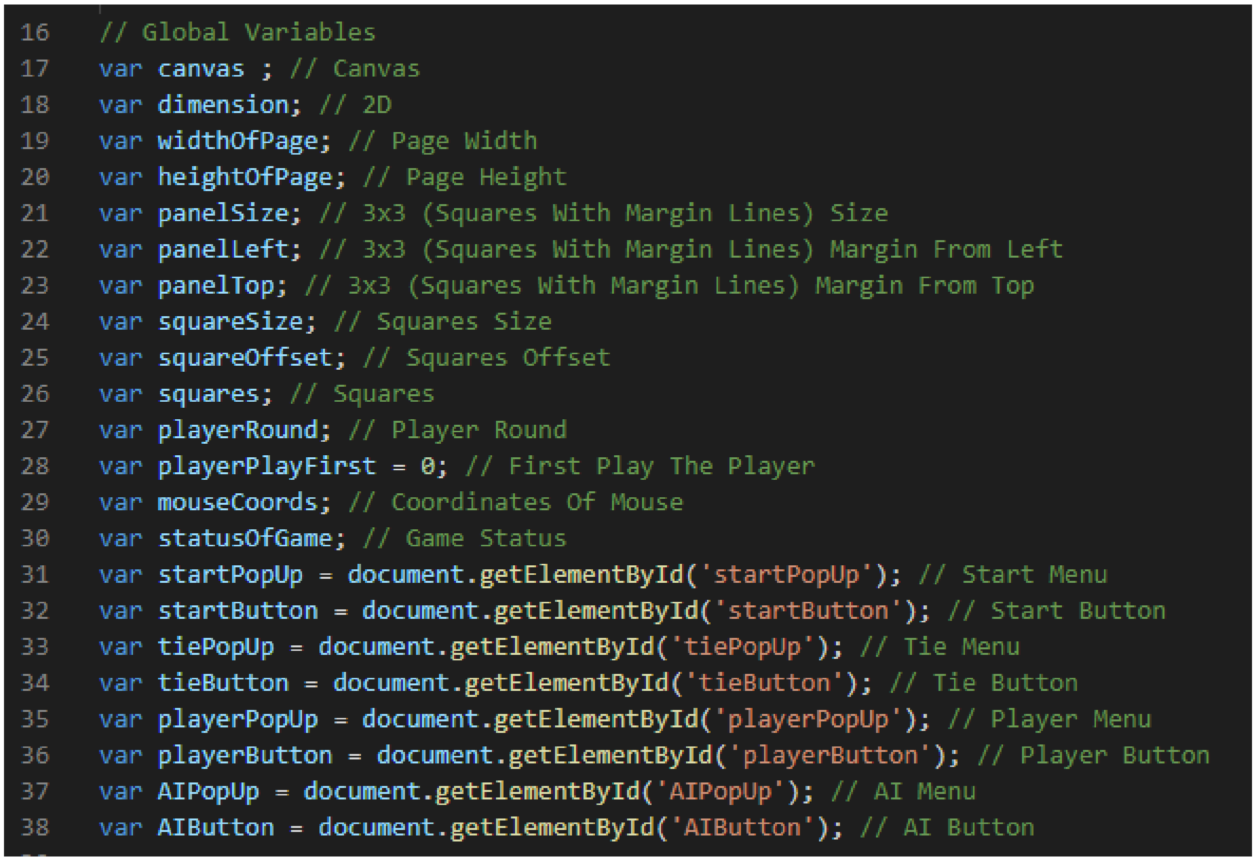
In the Code Pictures above (23 and 24) are a part of the stylesheet.css file of the Tic Tac Toe Game. This file is linked to the index.html file.

As we see in Code Picture 23, at the beginning the appearance of the div "main" is defined, which determines the height and width that this div will have, the alignment of the items and how the cursor will look. Below, is defined the appearance of all the divs contained in the div main. Thus, the position of the div, the alignment of the text, the filter and some animations are determined. The next three (i.e., #ai, #tie, #player) specify the menu that popups when the game ends. Here the background is defined, a transform animation and a border radius which rounds the popup at the edges.

In Code Picture 24, we see that the appearance of all the buttons that do not have an id and are in the game is determined. In this are defined more things in relation to the other buttons. That is, is defined the background color of the buttons, animations, cursor, opacity, letter spacing, border radius etc. In the beginning the appearance of the canvas is determined and then the appearance of the div "game" when its class name changes to "on".

6.4.3) File: ai.js

In this file, the game is implemented. Below, is analyzed a part of the file ai.js.

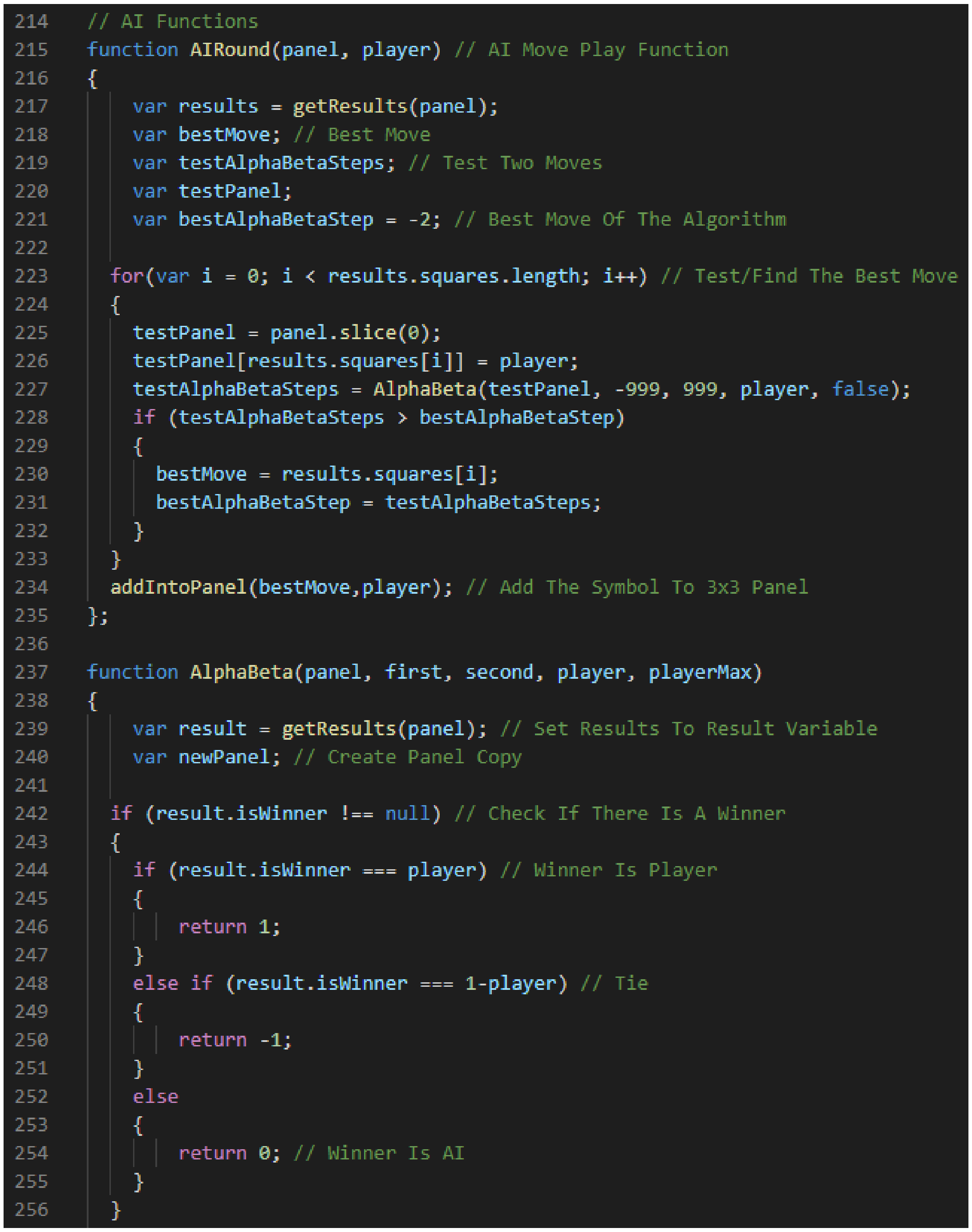


```
// Global Variables
var canvas ; // Canvas
var dimension; // 2D
var widthOfPage; // Page Width
var heightOfPage; // Page Height
var panelSize; // 3x3 (Squares With Margin Lines) Size
var panelLeft; // 3x3 (Squares With Margin Lines) Margin From Left
var panelTop; // 3x3 (Squares With Margin Lines) Margin From Top
var squareSize; // Squares Size
var squareOffset; // Squares Offset
var squares; // Squares
var playerRound; // Player Round
var playerPlayFirst = 0; // First Play The Player
var mouseCoords; // Coordinates Of Mouse
var statusOfGame; // Game Status
var startPopUp = document.getElementById('startPopUp'); // Start Menu
var startButton = document.getElementById('startButton'); // Start Button
var tiePopUp = document.getElementById('tiePopUp'); // Tie Menu
var tieButton = document.getElementById('tieButton'); // Tie Button
var playerPopUp = document.getElementById('playerPopUp'); // Player Menu
var playerButton = document.getElementById('playerButton'); // Player Button
var AIPopUp = document.getElementById('AIPopUp'); // AI Menu
var AIButton = document.getElementById('AIButton'); // AI Button
```

Code Picture 25: Tic Tac Toe – ai.js File (Part 1)

In Code Picture 25 we see that all the variables are defined. Some of these variables are the panelSize where is the size of the board, Squares Size where is the size of the squares, the statusOfGame which shows who won and more. All the names of the variables are written in such a way that only from the name the programmer knows what it is and what this variable does.

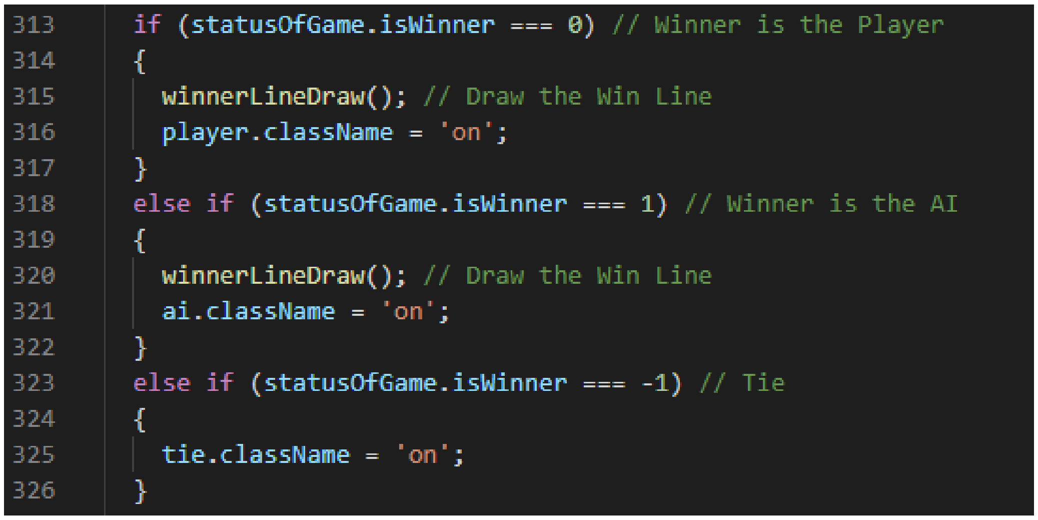
Then the elements that have id are connected with this file and are saved so they can be programmed. Some of them are startButton, which is the game start button and is stored in the variable startButton, AIButton, which is the Artificial Intelligence menu button when the game ends and is stored in the variable AIButton, the tiePopUp, which is the popup menu that appears when the game ends and is tied and is saved in the variable tiePopUp.



```
// AI Functions
function AIRound(panel, player) // AI Move Play Function
{
    var results = getResults(panel);
    var bestMove; // Best Move
    var testAlphaBetaSteps; // Test Two Moves
    var testPanel;
    var bestAlphaBetaStep = -2; // Best Move Of The Algorithm

  for(var i = 0; i < results.squares.length; i++) // Test/Find The Best Move
  {
    testPanel = panel.slice(0);
    testPanel[results.squares[i]] = player;
    testAlphaBetaSteps = AlphaBeta(testPanel, -999, 999, player, false);
    if (testAlphaBetaSteps > bestAlphaBetaStep)
    {
      bestMove = results.squares[i];
      bestAlphaBetaStep = testAlphaBetaSteps;
    }
  }
  addIntoPanel(bestMove,player); // Add The Symbol To 3x3 Panel
};

function AlphaBeta(panel, first, second, player, playerMax)
{
    var result = getResults(panel); // Set Results To Result Variable
    var newPanel; // Create Panel Copy

  if (result.isWinner !== null) // Check If There Is A Winner
  {
    if (result.isWinner === player) // Winner Is Player
    {
      return 1;
    }
    else if (result.isWinner === 1-player) // Tie
    {
      return -1;
    }
    else
    {
      return 0; // Winner Is AI
    }
  }
```

Code Picture 26: Tic Tac Toe – ai.js File (Part 2)

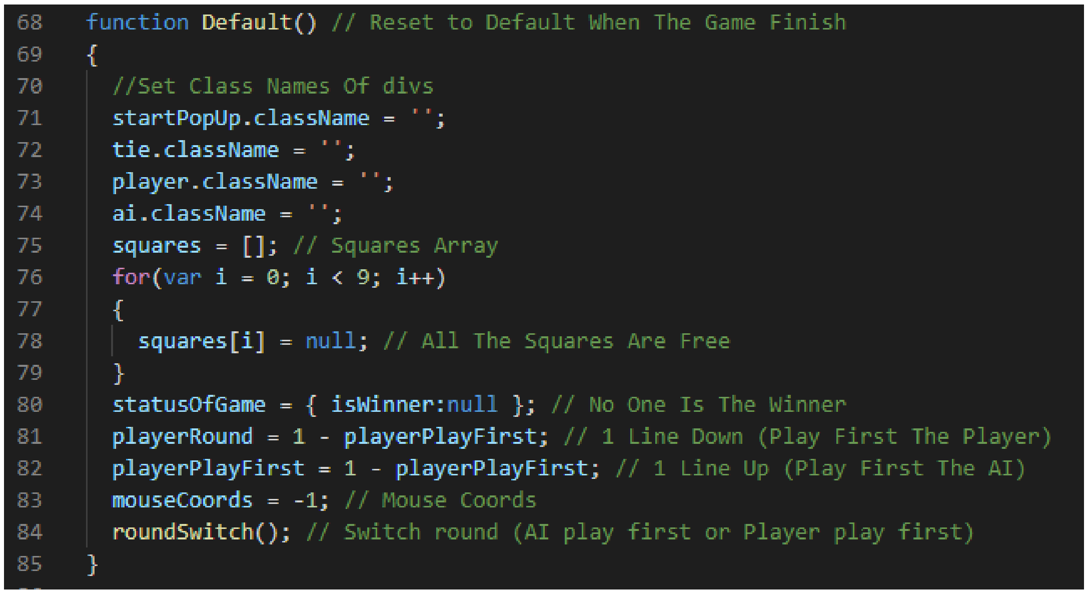
In Code Picture 26 we see some of the functions associated with the Artificial Intelligence of the game and specifically of the player's opponent. The AlphaBeta() function controls all the steps the player has taken and whether there is a winner in the game. The AIRound() function is the function where AI moves. So in this function a copy of the whole board is created, all possible movements for the positions are checked and the best move is selected and then the "O" symbol appears on the board.



```
if (statusOfGame.isWinner === 0) // Winner is the Player
{
  winnerLineDraw(); // Draw the Win Line
  player.className = 'on';
}
else if (statusOfGame.isWinner === 1) // Winner is the AI
{
  winnerLineDraw(); // Draw the Win Line
  ai.className = 'on';
}
else if (statusOfGame.isWinner === -1) // Tie
{
  tie.className = 'on';
}
```

Code Picture 27: Tic Tac Toe – ai.js File (Part 3)

In Code Picture 27 we see the commands with which the winner comes out. So if the player wins the game, the class name of the div with id "player" change to "on" and the winnerLineDraw() function is called where is the function with which the red line is created. If the AI wins the game, the class name of the div with id "ai" change to "on" and the winnerLineDraw() function is called. If the game draws, the class name of the div with id "tie" change to "on" and the winnerLineDraw() function is not called because there is no winner.



```
function Default() // Reset to Default When The Game Finish
{
  //Set Class Names Of divs
  startPopUp.className = '';
  tie.className = '';
  player.className = '';
  ai.className = '';
  squares = []; // Squares Array
  for(var i = 0; i < 9; i++)
  {
    squares[i] = null; // All The Squares Are Free
  }
  statusOfGame = { isWinner:null }; // No One Is The Winner
  playerRound = 1 - playerPlayFirst; // 1 Line Down (Play First The Player)
  playerPlayFirst = 1 - playerPlayFirst; // 1 Line Up (Play First The AI)
  mouseCoords = -1; // Mouse Coords
  roundSwitch(); // Switch round (AI play first or Player play first)
}
```

Code Picture 28: Tic Tac Toe – ai.js File (Part 4)

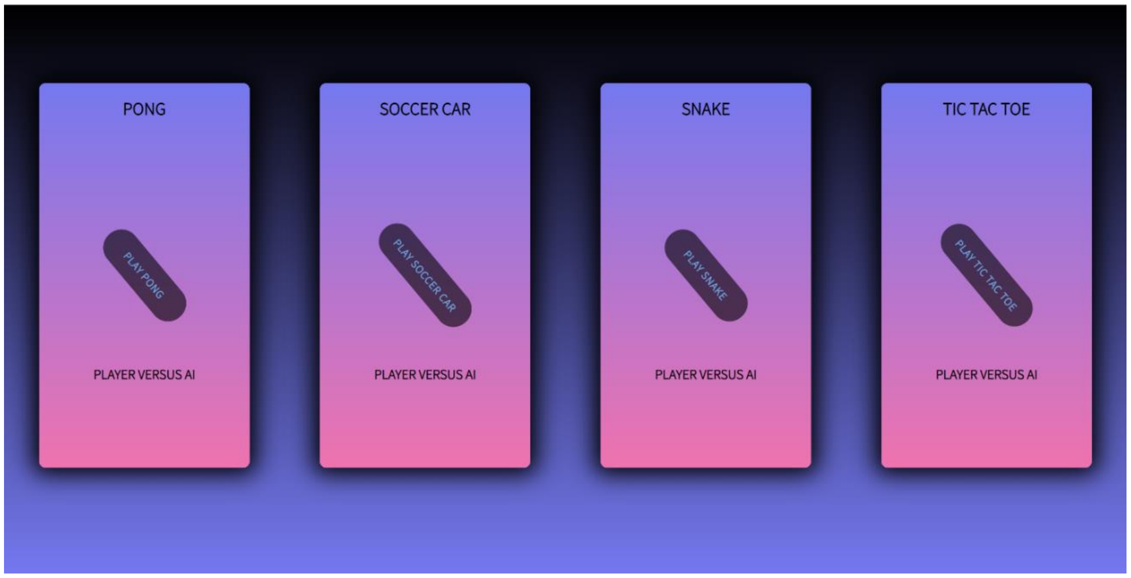
In Code Picture 28 we see the function that sets the default values of the variables. Some of these are statusOfGame where it states that no one is a winner, RoundSwitch () where it changes the order (who will play first) of the players each time the game starts from the beginning, sets the squares table, clear out the class names of the divs and more.

7) Presentation of Games Suite

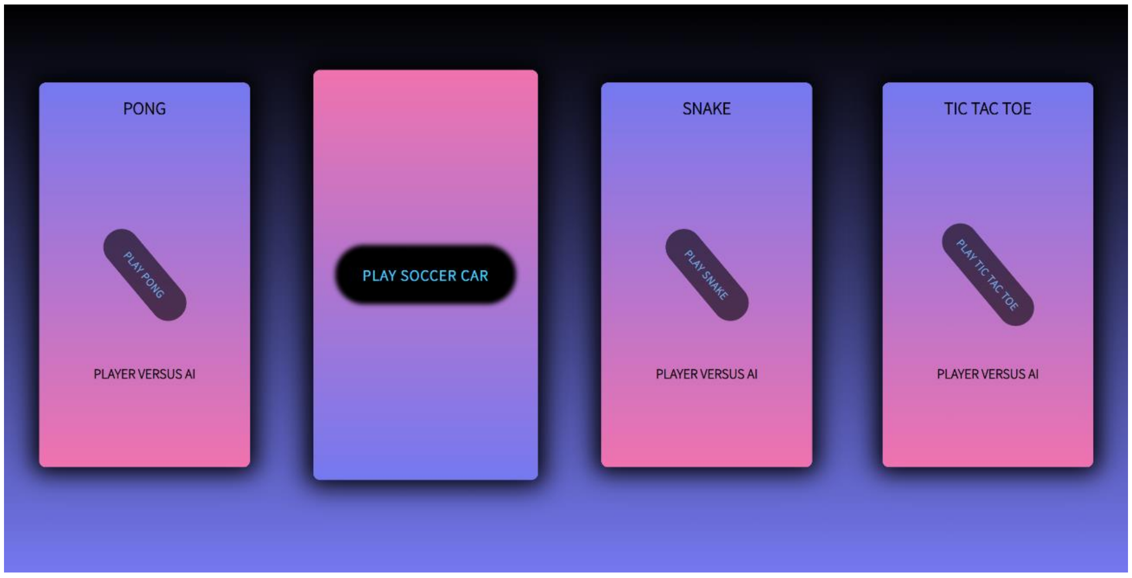
Below, the game will be presented with the help of .gif pictures.

7.1) Main Page

This is the main page and from here the selection of games is made and it is the first page that the player encounters.



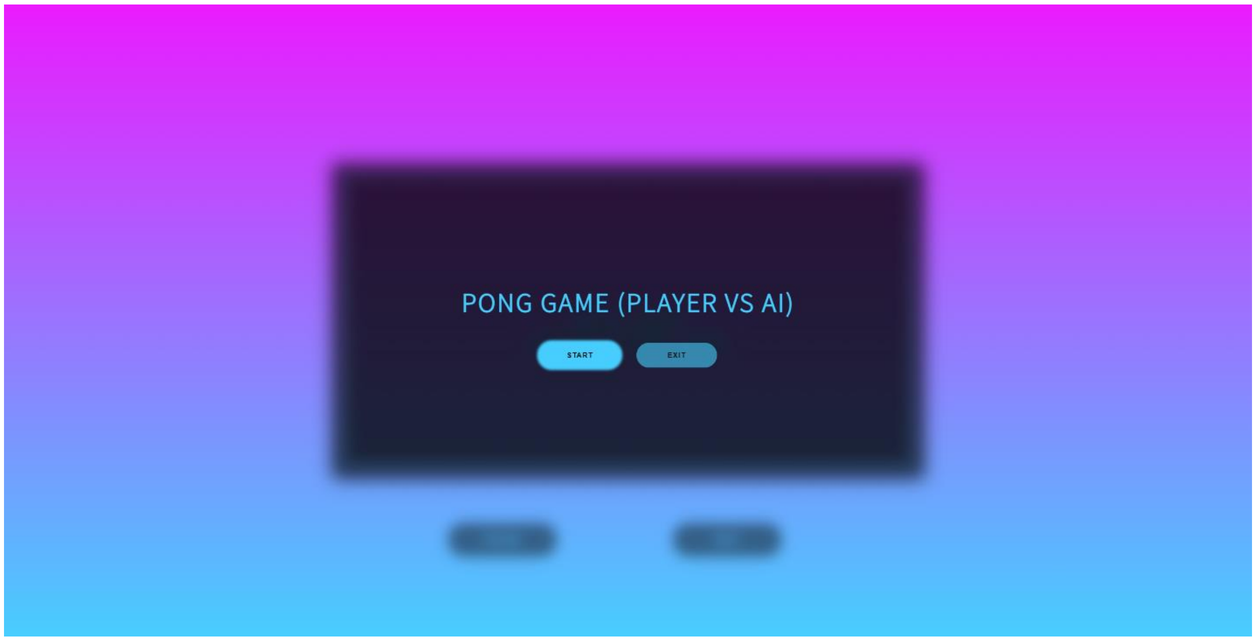
Presentation Picture 1: Games Suite Main Page



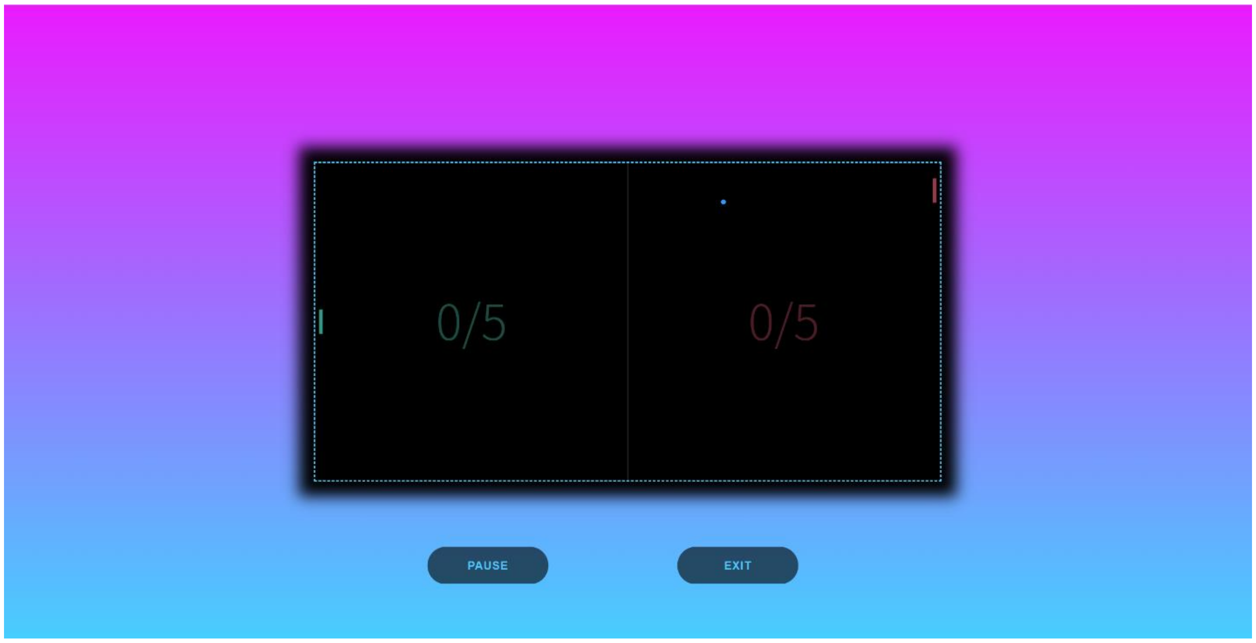
Presentation Picture 2: Games Suite Main Page

7.2) Pong Game

As soon as the user presses the "Play Pong" button then he encounters the following picture. Pressing Start starts the game, pressing Exit returns to the main page, that is, to the selection of games. Pressing Pause the game stops and a menu with three options is displayed.



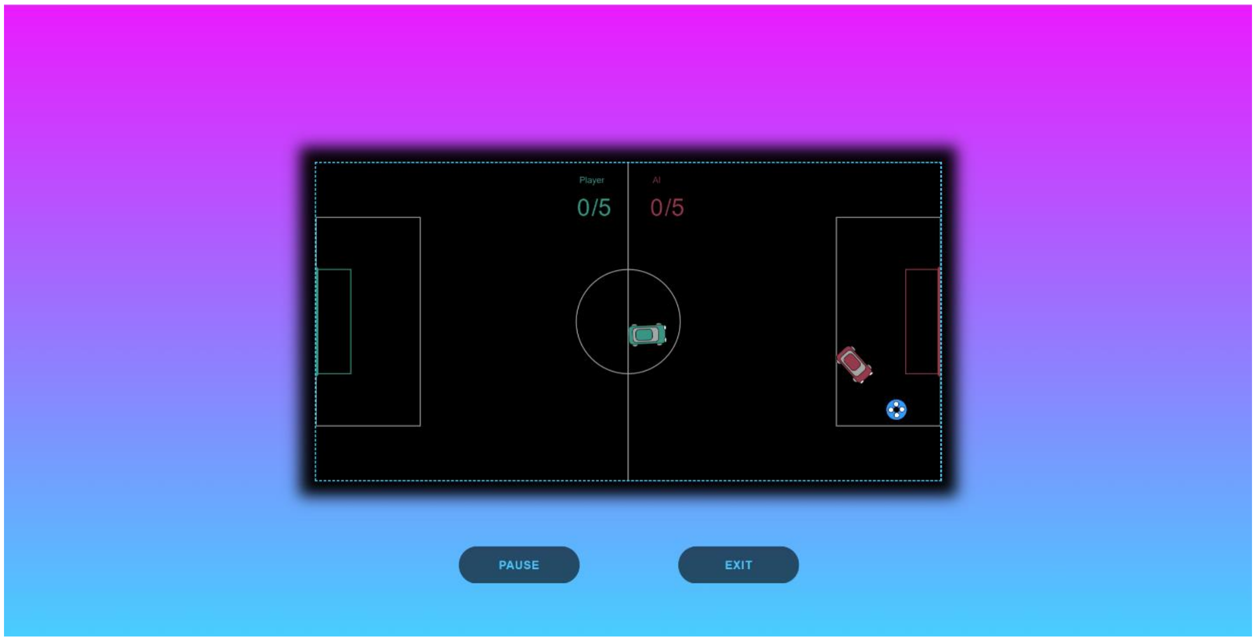
Presentation Picture 3: Pong Game



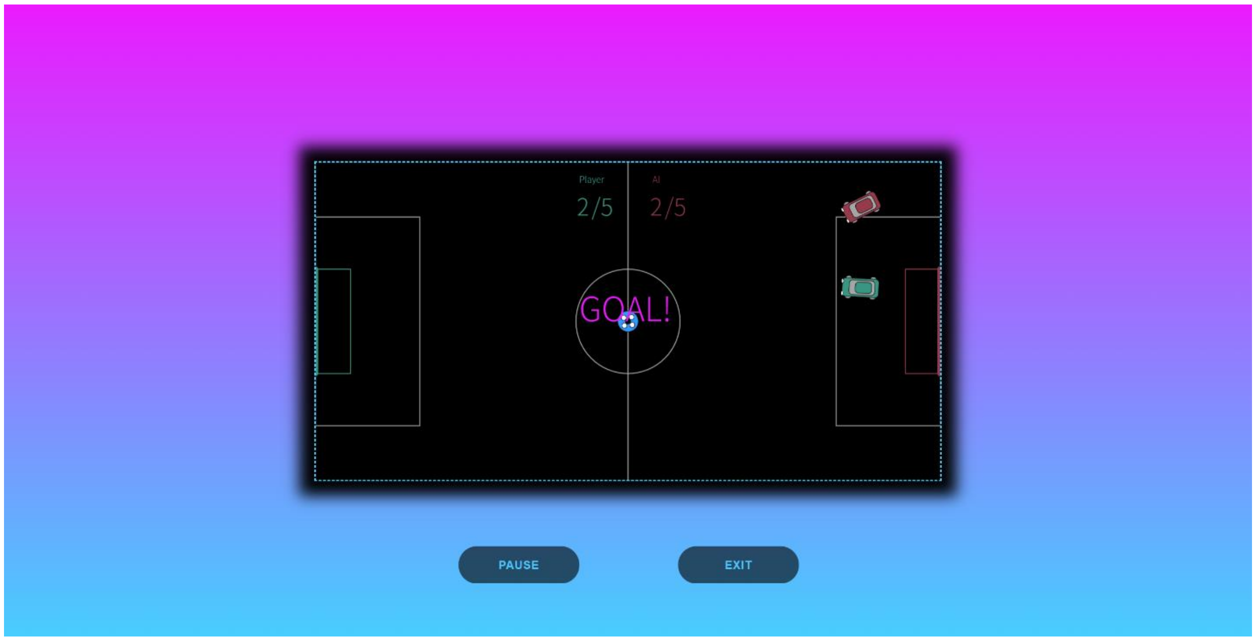
Presentation Picture 4: Pong Game

7.3) Soccer Car Game

As soon as the user presses the "Play Soccer Car" button then he encounters the following picture. Pressing Start starts the game, pressing Exit returns to the main page, that is, to the selection of games. Pressing Pause the game stops and a menu with three options is displayed.



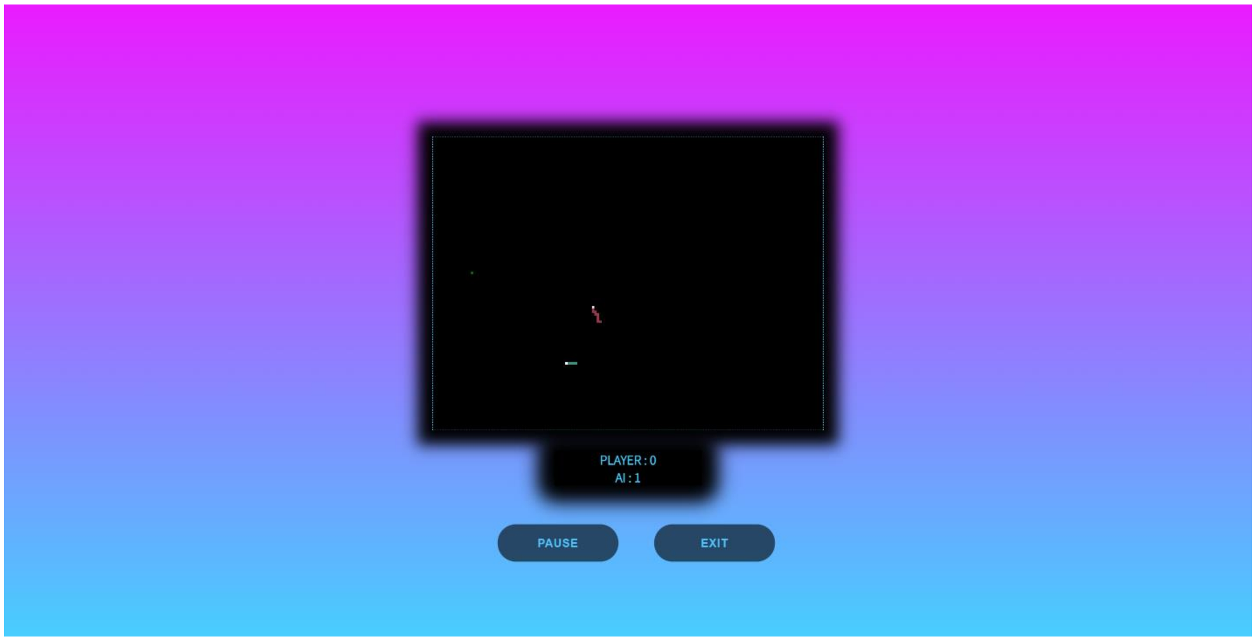
Presentation Picture 5: Soccer Car Game



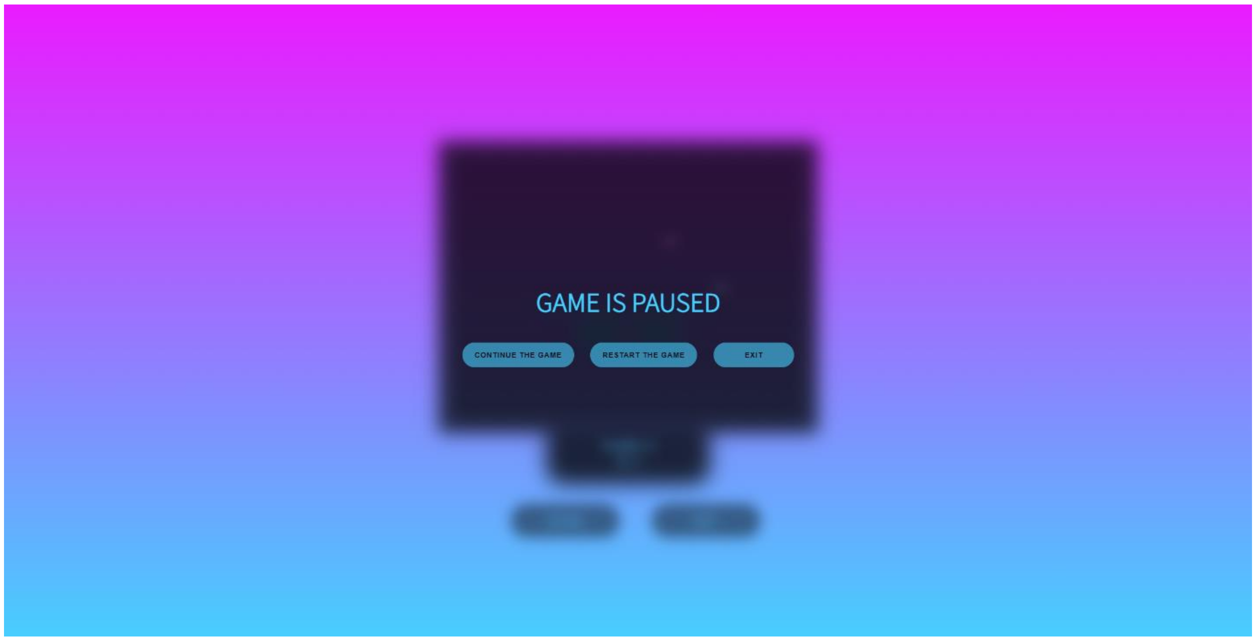
Presentation Picture 6: Soccer Car Game

7.4) Snake Game

As soon as the user presses the "Play Snake" button then he encounters the following picture. Pressing Start starts the game, pressing Exit returns to the main page, that is, to the selection of games. Pressing Pause the game stops and a menu with three options is displayed.



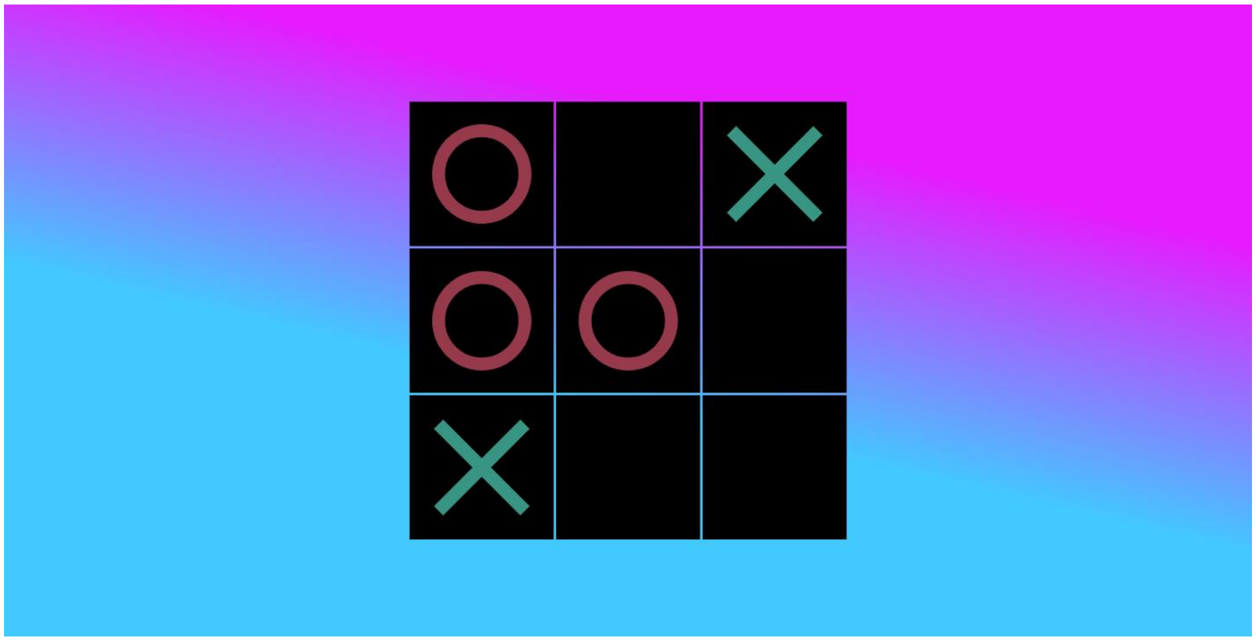
Presentation Picture 7: Snake Game



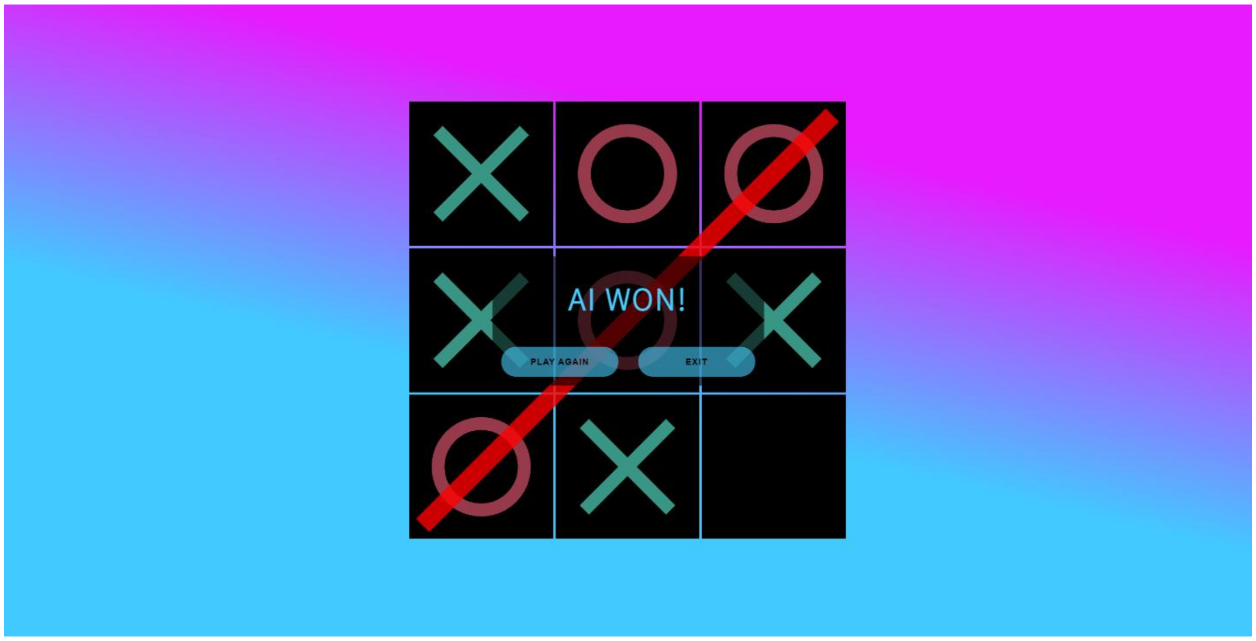
Presentation Picture 8: Snake Game

7.5) Tic Tac Toe Game

As soon as the user presses the "Play Tic Tac Toe" button then he encounters the following picture. Pressing Start starts the game, pressing Exit returns to the main page, that is, to the selection of games. When the game ends then the appropriate message window popup appears.



Presentation Picture 9: Tic Tac Toe Game



Presentation Picture 10: Tic Tac Toe Game

8) Conclusion

In Conclusion, technology has helped the people a lot and with-it people evolve more and more. The development of the internet in our age has zeroed the distances between people. The internet has now become a way of life and most people cannot live without it. Many businesses have been created because of it and many people live because of it. In recent years, the branch of Artificial Intelligence is growing and becoming more and more important in people's lives. Nowadays everything is smart from light bulbs, speakers to even whole houses operate only with one button. All this "Intelligence" brought the creation of smart electronic games. These electronic games are growing at a high rate and all companies are using Artificial Intelligence in them so that the user has a better experience. In the project we saw some games which are old, which when they were created only the player was playing without the opponent who is an Artificial Intelligence algorithm. It was chosen to create the project in web-based format because the user can log in from all platforms and play immediately (if I uploaded the suite on the internet). Thus, in the project is integrated the Artificial Intelligence and in particular static Artificial Intelligence. This means that the games were programmed statically, i.e., the Artificial Intelligence algorithm cannot make decisions on its own, but the commands and instructions are written by the developer and do not change. In the development of the code the diagrams and the methodology of the waterfall helped a lot. Each game was first made separately and then all four games were merged into one file.
Also, in the project I used the Software Development Life Cycle where through the diagrams and the organization of each stage the project was achieved.

9) References

- Howe, D., 2010. *information technology from FOLDOC*. [online] Web.archive.org. Available at: <https://web.archive.org/web/20130415234011/http://foldoc.org/Information+Technology> [Accessed 28 February 2021].
- Butler, J., 1997. *Information Technology History - Outline*. [online] Tcf.ua.edu. Available at: <https://tcf.ua.edu/AZ/ITHistoryOutline.htm> [Accessed 28 February 2021].
- Bebbington, S., 2014. *What is computer programming?.* [online] Tumblr. Available at: <https://yearofcodes.tumblr.com/what-is-programming> [Accessed 28 February 2021].
- En.wikipedia.org. 2021. *Programming language*. [online] Available at: <https://en.wikipedia.org/wiki/Programming_language> [Accessed 28 February 2021].
- Poole, D., Mackworth, A. and Goebel, R., 1998. *Computational Intelligence and Knowledge*. [online] Cs.ubc.ca. Available at: <https://www.cs.ubc.ca/~poole/ci/ch1.pdf> [Accessed 1 March 2021].
- Gellersen, H.-W., & Gaedke, M. (1999). *Object-oriented Web application development. IEEE Internet Computing, 3(1), 60–68.*
- Oxford, 2009. "IT", *A Dictionary of Physics*. Oxford University Press.
- Indeed.com. 2021. *What Is a Web Application? How It Works, Benefits and Examples*. [online] Available at: <https://www.indeed.com/career-advice/career-development/what-is-web-application> [Accessed 2 March 2021].
- Berners-Lee, T., Fielding, R. and Frystyk, H., 1996. *RFC 1945 - Hypertext Transfer Protocol -- HTTP/1.0*. [online] Tools.ietf.org. Available at: <https://tools.ietf.org/html/rfc1945> [Accessed 2 March 2021].
- En.wikipedia.org. 2021. *CSS*. [online] Available at: <https://en.wikipedia.org/wiki/CSS> [Accessed 3 March 2021].
- Kamvouti - Verou, M., n.d. [online] Pages.cs.aueb.gr. Available at: <http://pages.cs.aueb.gr/courses/epl131/files/CSS_notes.pdf> [Accessed 3 March 2021].
- En.wikipedia.org. 2021. *JavaScript*. [online] Available at: <https://en.wikipedia.org/wiki/JavaScript> [Accessed 4 March 2021].
- GitHub. 2015. *microsoft/vscode*. [online] Available at: <https://github.com/microsoft/vscode> [Accessed 4 March 2021].
- Papataxiarhis, V., Dimitrov, B., Goyal, V. and Nehinbe, J., 2010. *© IJCSI PUBLICATION 2010*. pp.96-100, [online] Citeseerx.ist.psu.edu. Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.402.8120> [Accessed 5 March 2021].
- Mihai, L., 2014. *Comparative study on software development methodologies*. [online] Dbjournal.ro. Available at: <https://dbjournal.ro/archive/17/17_4.pdf> [Accessed 5 March 2021].
- Www0.dmst.aueb.gr. n.d. *The Waterfall Methodology*. [online] Available at: <http://www0.dmst.aueb.gr/louridas/lectures/dais/process/ar01s04.html> [Accessed 6 March 2021].
- Shobika, 2016. *Software Development Life Cycle – SDLC – Deliverables*. [online] StuDocu. Available at: <https://www.studocu.com/row/document/university-of-technology-jamaica/software-

engineering/summaries/sdlc-deliverables-stages-of-software-life-cycle/1489591/view> [Accessed 6 March 2021].

- Balaji, S. and Sundararajan Murugaiyan, D., 2012. *WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC*. [online] Mediaweb.saintleo.edu. Available at: <https://mediaweb.saintleo.edu/Courses/COM430/M2Readings/WATEERFALLVs%20V-MODEL%20Vs%20AGILE%20A%20COMPARATIVE%20STUDY%20ON%20SDLC.pdf> [Accessed 7 March 2021].
- Rumbaugh, J., Jacobson, I. and Booch, G., 1998. *The Unified Modeling Language Reference Manual*. Addison Wesley Longman, Inc.

Pictures:

- Static.javatpoint.com. 2021. [online] Available at: <https://static.javatpoint.com/tutorial/software-engineering/images/software-engineering-software-development-life-cycle.png> [Accessed 5 March 2021].
- Upload.wikimedia.org. 2021. *Waterfall model*. [online] Available at: <https://upload.wikimedia.org/wikipedia/commons/e/e2/Waterfall_model.svg> [Accessed 6 March 2021].

Licenses: