



Departamento de Ciências e Tecnologias de Informação

## VEL - An online games platform

Carlos Filipe Paiva da Cunha Macieira Antunes

Projecto submetido para a unidade curricular de Introdução à Investigação em Engenharia no  
âmbito do Mestrado em Engenharia Informática

Coordenador(a) da disciplina:

Doutor Joaquim Lourenço dos Santos Esmeraldo, Professor auxiliar,  
ISCTE - Lisbon University Institute

Co-Orientador:

Doutor Sérgio Miguel Carneiro Moro, Professor auxiliar,  
ISCTE - Lisbon University Institute

Dezembro, 2019



University Institute of Lisbon

Department of Information Science and Technology

## VEL - An online games platform

Carlos Filipe Paiva da Cunha Macieira Antunes

A Dissertation presented in partial fulfillment of the  
Requirements for the Degree of

**Master in Computer Engineering**

### **Supervisor**

Joaquim Lourenço dos Santos Esmerado, Assistant Professor,

PhD ISCTE-IUL

### **Co-Supervisor**

Sérgio Miguel Carneiro Moro, Assistant Professor,

PhD ISCTE-IUL

December 2019

# Table of contents

Title	Page
1. Introduction	1
1.1 Video games evolution	4
1.2 Motivation	5
1.3 Context / Scope	7
1.4 Investigation questions	8
1.5 Objectives	10
1.6 Investigation Methods	12
1.7 Document structure	15
2. Theoretical Background and related Work	16
2.1 Game Development	16
2.2 Game engines	17
2.3 Virtual Reality	19
2.4 Augmented Reality	20
2.5 Select and justify the development technology used	21
2.6 Game Scrum	23
2.7 Gamification	24
2.8 N-Tier Architecture	26
2.9 Web services	30
2.10 Web development	34
2.11 Churn predictions	35

2.12 Future predictions	36
2.13 Online gaming platforms	37
2.14 Computer games	39
3. Research Gap	46
4. Panning	49
5. Expected results, conclusions and future work	50
6. Bibliographic references	51

“I love those who can smile in trouble, who can gather strength from distress, and grow brave by reflection. 'Tis the business of little minds to shrink, but they whose heart is firm, and whose conscience approves their conduct, will pursue their principles unto death.”

Leonardo da Vinci

## Acknowledgments

I would like to express my appreciation to the professors Joaquim Lourenço dos Santos Esmeraldo and Sérgio Miguel Carneiro Moro for giving me the opportunity to develop this system and also for their orientations in order to adjust to the necessary requirements and objectives and also I would like to show my gratitude to the continuous support given.

My gratitude to ISCTE which has given me the chance to develop my skills and also for everything I've learned in so many areas during this course of informatics engineering.

My appreciation to IST, Nova SBE and EST IPS universities where I have studied and helped me in my personal and professional life.

Also a mention of honor to some companies in which I have worked as a software engineer like Microsoft, BNP Paribas, Pfizer, Xerox, Airbus, Proximus, Infosys, Siemens among others which was a privilege to work for and that will always be present in my heart.

Finally, I would like also to give my gratitude to my family.

## **Abstract**

Video games have evolved over time, their complexity and variety has been increasing. The first computer game appeared in 1952, and today the market value of computer games is more than 120 billion dollars. New technologies have been used like mixed reality and virtual reality, in an area where there is no limit to the imagination. Online games emerged in the 70s and have been growing ever since. These type of games have a web development component, a game development component, a visual and 3D programming component, as well as an artificial intelligence component, among others.

This study aims to develop an online system that allows users to manage their profiles interacting with different types of games available on the platform. The system allows the implementation of several technologies on a single platform, taking into account other research projects in this area, as well as a business analysis. This project will investigate how certain users interact with different types of games.

The dissertation aims to present a new solution compared to the current platforms. It provides a credit system as well as the visualization of all historical data, which can be used for certain purposes such as withdrawal forecasts, mentor the development because it informs which games and features are less used and has an innovative aspect because it aims to implement virtual reality and augmented reality games online. It will also make a statistical and predictive analysis with a small error rate providing the churn for each player.

## **Resumo**

Os video jogos têm vindo a evoluir ao longo dos tempos, a sua complexidade e variedade tem vindo a aumentar. O primeiro jogo para computador surge em 1952, e actualmente o valor de mercado dos jogos de computadores é superior a 120 biliões de dólares. Novas tecnologias têm vindo a ser utilizadas como a realidade mista e a realidade virtual, numa área onde não há limite para a imaginação, os jogos on line surgiram na década de 70 e têm vindo a crescer. Possuem uma componente de desenvolvimento web, uma componente de desenvolvimento do jogo em si, uma vertente de programação visual e 3D, assim como uma componente de inteligência artificial entre outras.

Este estudo tem como objetivo desenvolver um sistema on-line que permita aos utilizadores gerir os respectivos perfis interagindo com diversos tipos de jogos disponíveis na plataforma. O sistema permite a implementação de diversas tecnologias em uma única plataforma, levando em consideração outros projetos de pesquisa nesta área, bem como uma análise empresarial. Este projeto investigará como certos utilizadores interagem com diferentes tipos de jogos.

A dissertação tem como objectivo apresentar uma solução nova comparativamente com as plataformas actuais visa como carácter diferenciador fornecer a possibilidade de um sistema de créditos assim como a visualização de todo o histórico podendo ser utilizados para determinados fins como previsões de desistência, ajuda o desenvolvimento pois informa quais os jogos e funcionalidades com menos utilização e como aspecto inovador visa implementar jogos de realidade virtual e realidade aumentada on line.

# 1. Introduction

The game industry has been increasing the financial value over the last years, currently the game industry is valued in more than 120 billion dollars, around the year 2000 the sector of mobile games has increased its value seriously and in the last 5 years a new sector namely the Virtual reality has also started to occupy a share of the market.

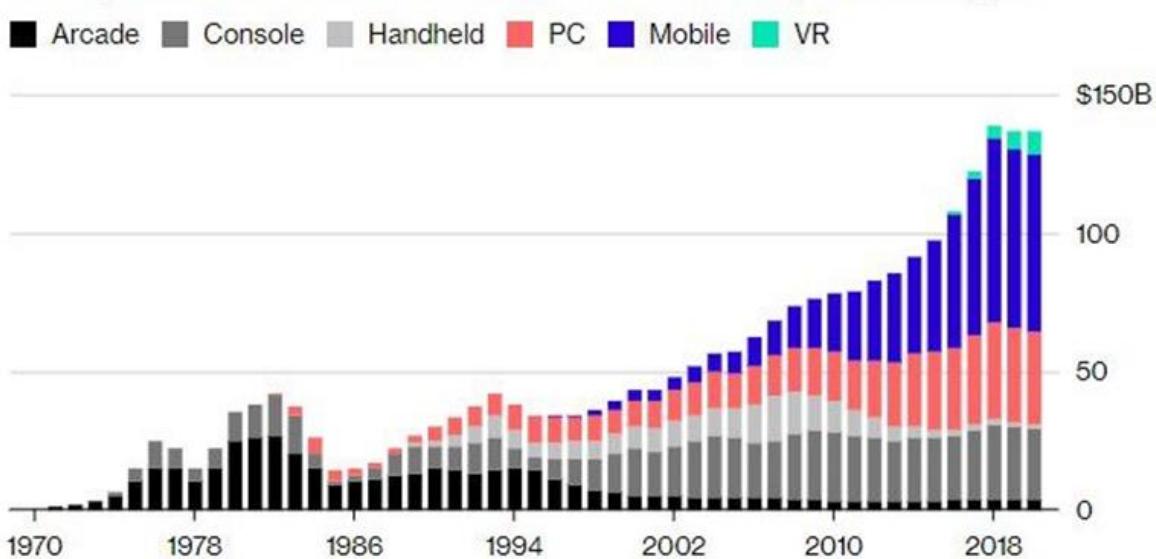


Figure 1 – Evolution of the financial value of the game industry

Source: Pelham Smithers

Based on [Jiajun 2018] there is an increasing trend of using computer games and also a higher variety of type of games taking in consideration the ratings by users and data sets of product reviews in Amazon it concludes that the gaming industry and the number of users is increasing.

Around 2006 in the industry of entertainment, video games started to become more profitable than the films and music industry, these last two industries maintain the same values over time and the gaming industry had an exponential growth as shown on figure 2.

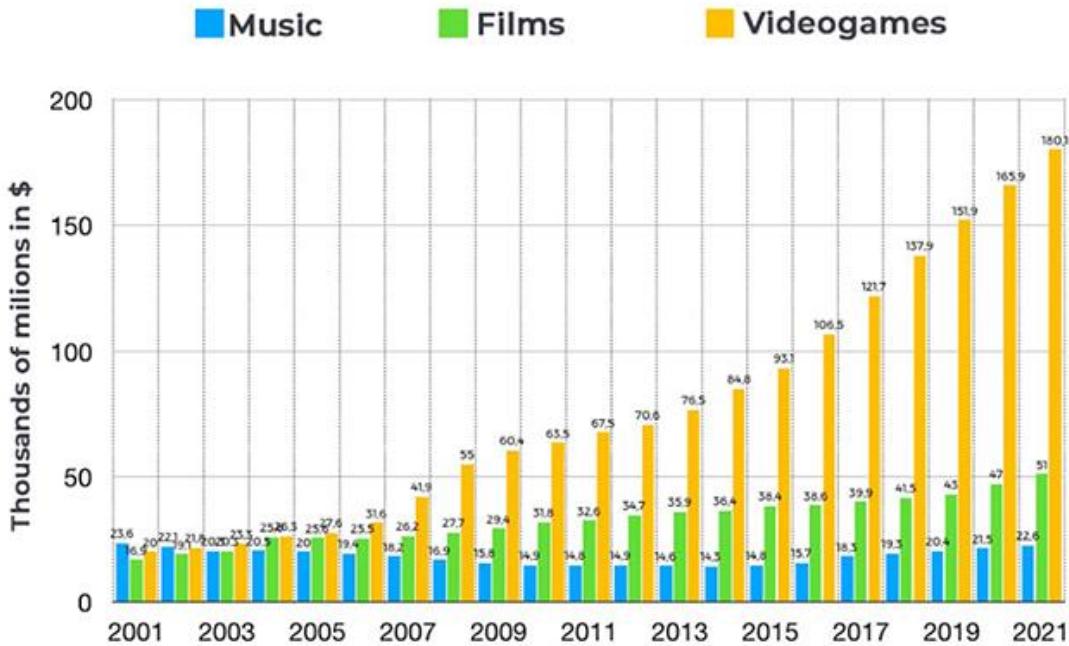


Figure 2 – Comparison of the game industry with the music and films industry

Source: Ethical Game design

In Portugal around 2008 more than 2.5 million video games were sold, surpassing the sale of films in DVD format for the first time. During this same year, in the United States, it was sold more than 268 million units. The game Grand Theft Auto IV sold more than 6 million units in just one week. One of the catalysts of this industry is the phenomenon of social networks, where publishers like Zynga and Playfish distribute and promote many of their games and that is why they have increased the popularity and use of their titles exponentially. EA (Electronic Arts) bought Playfish. Google has not yet revealed its plans for this industry, but it has already shown signs that it is attentive and preparing moves, with the registration of the patent “Web-Based System for Generation of Interactive Games Based on Digital Videos” [Fromme 2012].

Concerning the online gaming industry Zynga currently has a Market Capital of 5.77B and Miniclip a company that started in 2001 with an investment of 40 000 dollars has now reached the market value of 1 billion.

Valve Corporation that created the Steam platform that allows to share games has a current equity value of 2.5 billion,

The browser PC games industry only has 2% of the total market value of the games industry corresponding to a total amount of 3.5 billion dollars.

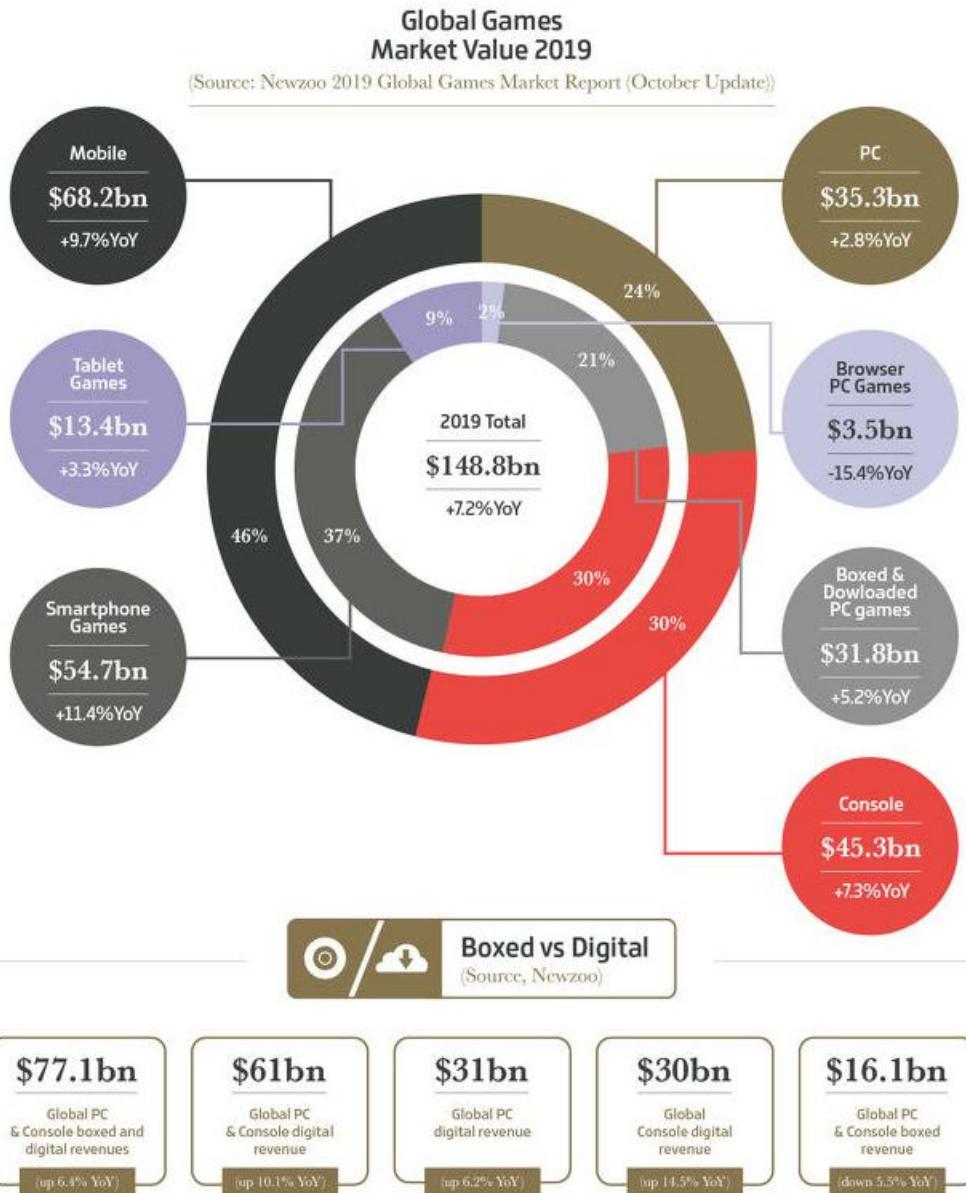


Figure 3 – Market value of the game industry per type of games

Source: [gamesindustry.biz](http://gamesindustry.biz)

According to [Johns 2006] the creation of networks of the video game industry leads to globalization, it takes in consideration the several stakeholders and the differences between the software and hardware development in the gaming industry.

## 1.1 Video games evolution

The research of [Chang 2007] shows the different hardwares used during their correspondent timeframes like the first Atari console, Nintendo, Sega Master System, Mega Drive, Saturn, PlayStation, Xbox among others, also provides other information like the frequencies used the CPU memory, etc...

Video games have evolved over time, their complexity and variety has been increasing. New technologies have been used like mixed reality and virtual reality in an area where there is no limit to imagination. Online games emerged in the 80's and have been growing ever since.

The first computer game appears in 1952 by A.S. Douglas, on the university computer, it was a version of the known game tic-tac-toe. In 1955 the U.S military created a game that simulated a war between the USSR and the USA.

At 1964 was created the BASIC programming language, which allows to create computer games. During the decade of the 70's appeared the first arcade; the first home console was created in 1972 and was called Magnavox Odyssey in this same year Atari develops the game Pong.

Over the decade of the 80's were developed famous games like Pac-Man, Donkey Kong, Tetris, SimCity, Sonic among others... During this decade Nintendo launches the 8-bit gaming console and the Gameboy. New 2D techniques were used based on visual patterns that allowed better visual games. The technique of the screen following the player's movement was introduced. Commodore introduces the Amiga computer and Sinclair research starts selling the ZX Spectrum.

The decade of the 90's was marked by Nintendo and Sega selling different types of gaming consoles. Some famous games of this decade are Doom and Warcraft. It was introduced the CD-ROMs, and the 2.5D technology started being using.

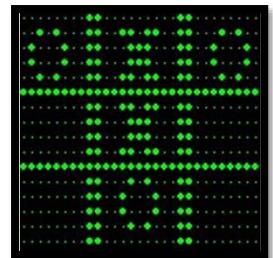


Figure 4 – The first computer game

The 2000s are a time period where it was introduced the 3D technology in many games like first person and third person shooters [Xian 2018]. Some of the famous games were Halo, Counter Strike, Hitman, and Grand Theft Auto among others...



Figure 5 – Virtual reality games



Figure 6 – Augmented reality games

According with [Dominic 2009], first person shooters appeared during the 90`s, being “Doom” one of the first of these type of games, the author also divides games in categories like action, driving, role play among others. It makes an analysis of shooting games and mentions some games like “Quake”, “Duke Nukem”, among others considering differences among them like the type of weapons, the maps, the mechanics and other factors.

Concerning the VR, Morton Heilig developed Sensorama in 1962 which was considered one of the first VR systems, although only recently VR started being commercialized in the computer game industry. The AR might be considered to be born on 1968 by Ivan Sutherland that created the first head mounted display, but only recently this technology has been used for computer games.

## 1.2 Motivation

The theme is concerning computer games and adjustment to new technologies like augmented reality and virtual reality. It is based on resolving the problem of how web games are currently developed and how they can be improved. This project is based on research to provide a more flexible web gaming platform, it will use new types of artificial intelligence in order to make the game more appealing to user and also has a statistical component that allows to predict which user is going to stop using the system and when. It also provides information to the developers like which game has more users, which game was played during a higher amount of time, among other useful information.

[Johnson 2001] mentions that the AI determines the success or failure of a computer development project it shows several techniques like extensive AI, which is going to be used in

this dissertation in order to make the game more appealing. This project aims to have applications using AR and VR which will be accessible on line which is something innovative. Research was done in order to make the games the more appealing as possible to users having in attention the graphic quality. This project has a strong component of 3d and visual development.

Another way to see the problem is to maximize the use experience of gaming platforms. This dissertation looks for innovation like inserting new technologies on line as VR and AR, after comparison with the current online gaming platforms. A statistical analysis and the ability of users in seeing all the historical activities and the implementation of a credit system allows to distinguish the project from the existent ones. It also provides information to the developers about user activities, by knowing the churn point they are able to know which game or which part of the project requires more programming efforts. The analysis of the user activities provides data for research and statistical analysis. It aims to build a platform that is not just based on playing games it will also have a profile management module where users can store personal data, like pictures, game reviews and access to statistics of the games played like time spent, chat conversations, among others. The project will have a credit system which allows to change credits among users; more credits will allow the access to new weapons, vehicles and new levels...

Based on [Morgan 2009], games have problems like sharing content among them, lag of online games, problems in scalability and execution. The online games are a billion dollar industry, with not so much graphic quality like a standalone game and difficulties to maintain the user's interest for long periods of time. This dissertation offers an approach to find a solution for these described problems.

The current web gaming platforms do not have an advanced profile system, a credit system and access to historical data, which could improve the user experience, and the graphic quality in some of the available games is not high. A lot of research was made trying to find augmented reality games and virtual games that functioned on line, but we were not able to find; currently the available software for these technologies has to be downloaded and installed locally on the computer or mobile device; this thesis aims to find a solution where these types of technologies can be accessible just by clicking a specific link.

### **1.3. Context / Scope**

This research is based on optimizing the games in order to be extremely easy to use for any kind of user. It is based on a complex interaction of different technologies, it uses several programming languages and several platforms like PHP, C#, Unity, Unreal Engine, C++, Visual Studio, MySQL among others. It also has APIs that call web services for some of the functions implemented. The system is based on a 3 tier architecture, having a front end, a back end and a data layer in order to make it more versatile and flexible; the code is structured and in some cases reused.

[Ampatzoglou 2010] basis the research approach on questions by developing review protocols, validating review protocols, identifying relevant research, selecting studies, checking study quality, extracting the required data and synthesizing the data obtained. This methodology was followed on this dissertation.

The scope of this project is based on the development of several games that simulates several events like car games, battles, plane control among other functionalities. Users are able to change the camera and to configure their game profiler. The project also offers the ability to check which games were played and when, among other historical data.

The project tries to simulate the behavior of racing cars, fighting scenarios, driving different types of vehicles, etc...

There is a strong component of visual programming and the main objective is to offer fun for the person that is going to use it. It is under the scope of the MSc of computer science lectured in the University of ISCTE and it aims to the areas of visual programming, game development, web development, database development and other areas like artificial intelligence, API development. One of the objectives is to find the most efficient way possible to aggregate several games according to criterias based on user decisions. The system was developed in a very organized manner, several parts of the HTML code are stored separately and can be added by using just one line of code on the main page. The system was conceived in order to add games just by adding one line of code.

Due to the complexity of the project the usage of PHP facilitates the implementation of the backend. Although it is a web platform where several games can be played, it can be adjusted to other areas like education and mathematics. It is adjustable to several operating systems and does not require a very powerful processor. As long as the computer has a browser and internet connectivity it can be played. It has several functionalities, like the option to crouch, pause, change camera, launch grenades, change weapon, generate health when not being hit, fire shotguns and pistols, run, etc... It has the functionalities of choosing levels, weapons and vehicles. It is free and easily accessible and it has a statistical calculation of the number of times the games were played and by who.

## 1.4. Investigation Questions

This research aims to optimize the development of online games and looks to make the games more appealing to users. It also developed a user profile system where data like images can be stored and a credit system allowing to change points among different users; there is also a component based on predicting when a player is going to quit (churn) and adjustment of the artificial intelligence to the current status of the player. Current online platforms in some cases oblige to install third party components and in general do not have a credit system or a profile where users can store data about their performance.

This project has a strong visual and 3D development component, and it uses many technologies like databases and web Services in order to provide an innovative web gaming platform.

After analyzing the current on line game platforms questions appeared like: Why the need to install 3<sup>rd</sup> party components? Why the graphics quality is not high and not very realistic in some cases? Why can't we play virtual reality and augmented reality games on line? Why current web game platforms do not have a credit system and in some cases only exists inside the game? How to minimize the latency? How to make the games and the on line platform the easier as possible in terms of usability? Why users can't see the games they have played and other information like the time spent? Why in some games the AI does not change according with the player status/level? How to make the on line platform the most appealing possible to users?

Based on [Navarrete 2013] tests were made with users with some similarities with the type of questionnaire that is going to be used in this dissertation. It is important to know how to

maximize the user satisfaction and what causes people to use a certain system. This project has a strong managerial component where the user performance is analyzed and the code is structured in order to facilitate the work of the programmers; because it is possible to see which games are more played, which games users quit in short periods of time, how many users per game, when they played, this information can be used in project management allowing the developers to know and prioritize their work, for example by knowing that there is a game that is played for short periods of time, that game has to be improved/changed in order to be more appealing to end users.

With these questions we can conclude that developing an online platform with these characteristics can be a successful project because it has new things and also different characteristics. The current online games can be better developed and users nowadays can play just by using a web browser. Games can be optimized in performance terms and are easier to play than in the past.

It also has a component of statistical management that is important for users and for developers. The use of new technologies can improve significantly the current on line gaming platforms.

Research idea	General Focus research questions
Game performance	How much is the lag time of the games? How can we make the games faster? How can the online games not be so dependent of the internet speed connection? How to make games adjust to the type of level of the user? How to allow access to technologies like augmented reality and virtual reality via web?
Credit system that allows access to new features like levels, weapons, etc...	How to make the credit system work for several games and not just one game? How to allow the credit change between users? Which types of accesses should be provided depending on the number of credits?
Ease of use	How to make games accessible without installing anything on the computer? How to make games available just by clicking one button? How to make the games change the difficulty level according with the status of the player?
Knowledge of historical data	How knowing what I have played and how much time I have spent in game can improve my use of a gaming platform? How can we

	predict when users will quit using the web platform? Which statistical data should be obtained to make accurate predictions?
The use of a profile system	What effect it has on users to be able to store game information? Which information should be stored in a web platform? What other functionalities should be offered in order to make it more appealing?
The use of specific hardware	How easy is it for users to be able to play the games that require specific hardware like augmented reality games and virtual reality games? Which characteristics these hardware units require? How to incorporate these specialized softwares on line?
Prioritization of tasks	How can programmers know which games they should improve first? Which games have less need for technical changes? Which games are more profitable? Which characteristics should be added to the games and to the platform?

Table 1 – Research ideas

Some other questions are:

- Is the user profile needed?
- Are AR and VR online a new thing?
- How can online gaming platforms be better?
- How can this technology be used in other areas?
- How can a web system be more adjusted to the user needs?
- Will it be able to provide users a better experience?

## 1.5 Objectives

This project aims to build an online platform for video games, and also to develop games in virtual reality and augmented reality.

The platform is available at:

<https://velsystem.000webhostapp.com/Vel/>

The problem currently detected is the visual quality of the current games available online. Current online gaming platforms do not have a user profile where images, videos and other information like game reviews can be stored and also do not provide statistical data related to time played, what games were played, how many times, etc... It also does not have a non-paid credit system where users can have access to more weapons, or cars, or levels depending on the amount of credits they have and also the possibility of users sharing credits among them.

This project aims to bring together several components, a high graphic quality and the implementation of artificial intelligence algorithms to define the behavior of enemies / competitors.

[Lindley 2003] mentions that game design can be improved, this dissertation will try to improve the graphic quality of on line games. This project also intends to facilitate the work of the developers because they will have access to information like which game has more users or which game was played for a longer period of time allowing them to know, which games or projects they should work and prioritize the development tasks. Statistical prediction will be used to know when a certain user is going to churn (quit using the web site based on statistical analysis).

Similar to other online websites, it aims to provide games of different areas such as shooting games, racing games, simulators, etc... One of the differentiating aspects is the profile system, a bit analogous to the one Facebook uses and a statistical analysis system somewhat similar with the one used by Google Analytics.

As an innovative aspect the project will have a virtual reality and augmented reality component available online which as per my knowledge does not exist yet, or it is not available... Concerning virtual reality you will need to use special glasses and for the augmented reality it will be required a camera with a screen to view the components generated by the computer. This dissertation has a differentiation component and also a research component.

## **1.6 Investigation Methods**

For this project it was used the deductive and the inductive approaches, it was developed a framework from the investigation made and tested in order to achieve the objectives proposed; from the data obtained it was possible to test theories and compare the information that was analyzed. It is concluded that the online gaming platforms still need improvements in very aspects like the graphic quality, the latency, among many other factors... This investigation allowed to provide insights and strategy techniques to achieve the objectives (results) desired.

There are several parameters on this research like looking for aspects that make online game platforms better than the current ones, looking for innovative things like online virtual reality and augmented reality and other aspects like the credit system and the profile module.

Several ideas were obtained via brainstorming and the reading of several articles and books related with this theme. The scope is based in optimizing the current processes and also making easier for programmers to change the platform based on the n-tier architecture, the MVC and other aspects that allow reuse and less code implementation. It were also obtained ideas for further research.

Key words were generated and several topics were researched like web development techniques, artificial intelligence processes, statistical analysis and predictions. This research was based in Design Science and Behavioral Science by making predictions about how users will interact with the platform and creation of new modules like the credit system.

The dissertation follows the method used in [Schoenau-Fog 2011] where an iterative game development method is used; Based on predicting the Churn point it is possible to define research methodologies for new and current games. It facilitates the management of the efforts done by the programmer by knowing for example which type of games have more participants. The data analysis allows to make predictions and find better solutions to optimize the platform.

Concerning the Design science method it was developed an online platform and tested with users with several metrics like level of graphics, functionalities, etc...

The programming languages that were used in this project were C#, C++, PHP, SQL, JavaScript, HTML, CSS, ActionScript, Unreal Script among others.... It was used APIs and the

MVC architecture and many tools like several versions of Visual Studio, Eclipse, MonoDevelop, 3d Max, Maya, Blender, CryEngine, Unreal Engine, UDK, several versions of Unity among others...

Several models were presented to achieve the final project and solutions to the current development of the online gaming platform. Many processes have been followed like the use of UML, requirement definitions, architecture of the code, development and design of the database, development in several engines, web development, analysis of existing related work, research on several aspects like latency, selection of the IDEs and game engines to use in order to solve the presented problems.

Many mathematical formulas were used to predict churn. Statistical analysis were done concerning the use of the web platform. Tests were performed with users to know usability metrics like satisfaction, difficulty among others... The dissertation has a component based on predictions and analysis of choice. Algorithms were developed to adjust the AI level and also to analyze the data in the database provided by the application users.

The project can be found at: <https://velsystem.000webhostapp.com/Vel/login.php>

The dissertation follows a strategy of identifying the needs. The system is available on line which facilitates the tests with users in order to perform an evaluation of the data obtained to improve the development process as described before on this document. It aims to improve the quality of online gaming.

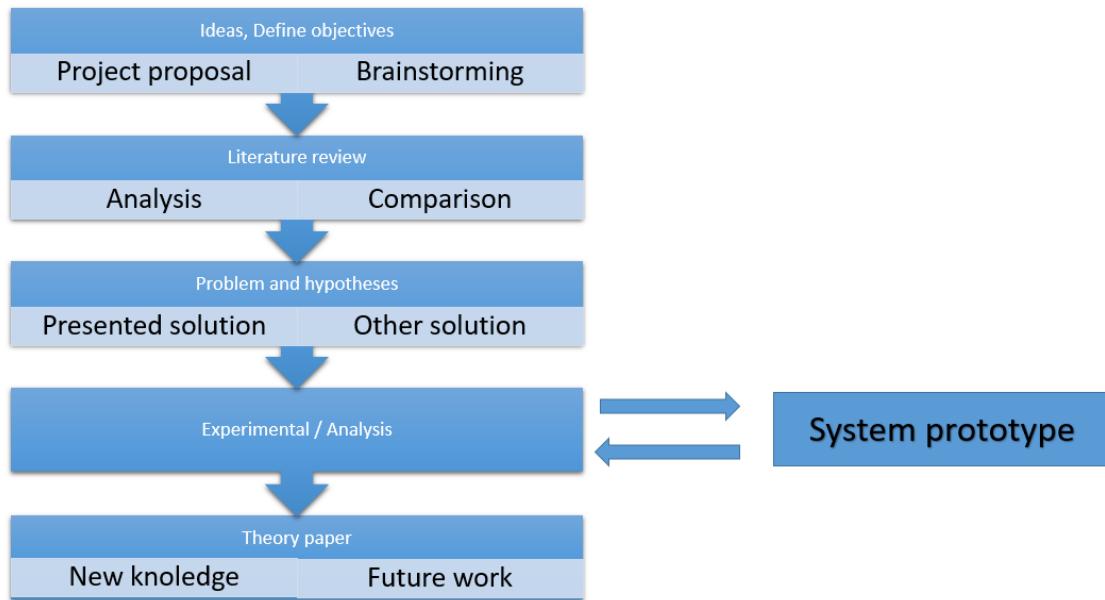


Figure 7 – Scientific research methodology

The methodology used on this dissertation is based initially on the definition of an idea, in this case it was taken in consideration aspects that could differentiate the web platform from the existing ones and also new ideas like implementing AR and VR online which can be a competitive advantage.

After the ideas are defined the next step is to do research and read what other authors did and drive conclusions as well as annotate all the aspects and take in consideration the processes used. For this dissertation it was taken in consideration several books, scientific articles and what other companies are currently doing.

The objective is to find a solution for the problem, like in example how can a software development project succeed in the real market? How can the current on line games be optimized? What can a game platform offer that the current ones do not have? Among other questions...

There are several stages like development, design and tests, for which is given a prototype in order to adjust the development to what the user really wants or needs. The data obtained is analyzed and follows a continuous integration methodology, implementing the new inputs in the solution.

The final step will be writing the results and conclusions obtained, present future work and also what could be done better.

## **1.7 Document Structure**

This paper has an initial section where it is given an introduction showing the game industry and its value as well as some of the players of the market, it also shows the problem and the given solutions. In this section it is given also the research questions and methods that are going to be used in this dissertation.

The second part of this dissertation mentions the articles and research that was made in this thematic. It has an analysis of the technologies and architecture used like the n-tier, the type of engines, the APIs among other processes and methodologies required to implement the web platform.

On the third section it is going to be mentioned details of the project as well as the design used and technical details like the database, backend, frontend and data layer that allow to provide several functionalities and to store data, it will also mention some of the techniques used for developing the games.

The forth chapter is based on the tests and questionnaires that are going to be done, as well as a demonstration of the application and the findings. Reveals also what was achieved, what can be improved, analysis of the results and future predictions.

The last part of this work will mention the conclusions and future work.

## **2. Theoretical Background and related Work**

### **2.1 Game Development**

The technology of computer games has been developing significantly over the last years. This project aims to allow the access to augmented reality and virtual reality on line [Aukstakalnis 2016]. It also has other objectives like implementing a credit system and a user profile, where the people that are going to use this platform can access their data and upload information, among other functionalities.

Based on [Blow 2004] nowadays several factors have to be taken in consideration like project size and complexity and at the same time new technologies are facilitating the process of development like the use of 3<sup>rd</sup> party components, code provided by engines, the possibility of using existing projects and that some engines make it easier to export to different platforms.

In order to make it more appealing the artificial intelligence adapts to the current status of the player.

Computer development projects face several issues like project size and complexity and difficulties on the engineering level. Games became more complex and requires more knowledge not only at a programming level.

Many games are made in C++, some take a long time to compile, the game architecture should take in consideration the minimization of dependencies. Another process that can be used is to implement a third party tool to distribute compiles among many machines.

This project has the problem that the time it takes to load a game depends on several factors like the internet connection, which for some games with high level of graphics quality might take some time to load.

The tests also have some complexity. Unit testing was used dividing the application in modules.

## 2.2 Game Engines

Game engines include modules of simulation code, like input, output (3D rendering, sound) among others... It should be taken in consideration the hardware requirements, the multiplayer support, the graphic quality, among other factors...

According with the work of [Dieter Fritsch 2009] many research was done in creating map textures and differentiating the indoor environments form the outdoor environments. This project aims to create realistic scenarios on the games.

The CryEngine has many features like the flow graph, integration with 3D Max, modelling tool that allows rigging, and a high graphic quality module that allows games to look more realistic. This engine also allows to develop multi thread and the use of several programming languages like LUA, ActionScript, C++, C# and has a world editor of very high capacity. It also allows to change the type of illumination and offers a high variety of shaders, has a module that allows to create animations, allows to place sounds for certain regions in the map, provides the option for multiplayers, uses several types of image compression, offers many libraries and also provides assets like the GameSDK that facilitates the use of this engine [Busby 2010].

Unreal Engine uses DirectX9, has a store where several objects can be downloaded; belongs to Epic Games, has a world editor, has the blueprints that allow to manage many game components like the artificial intelligence without being hard coded, as a very high graphic quality, allows to export to several platforms like mobile, has an advanced lightning system, many famous games were built with this engine. It has a vast amount of texture treatments, and allows character and animation creation, it can be combined with other softwares like Maya which allows the import of several type of files, it is an update of the previous version the UDK. It supports C++ and C#, and has an integration with Visual Studio 2005.

Unity engine allows the integration of several type of files and has an asset store with many products, allows to create files, offers an IDE called MonoDevelop, has a great compiler and export system which is compatible with many platforms like iOS, PlayStation, Xbox, Linux among many others...

There are many shaders in Unity, it has an interface that is more accessible to new users and it has a console that mentions the error and bugs that occur, it does not allow to compile or run the program if there is an error, previous versions supported JavaScript but the latest versions only support C#.



Figure 8 – Project game: By the night... with unreal engine



Figure 9 – IDE: Unreal Engine 4



Figure 10 – Project game: Hell are the others with Unity



Figure 11 – IDE: Unity Engine 2019



Figure 12 – Project game: Huma with CryEngine

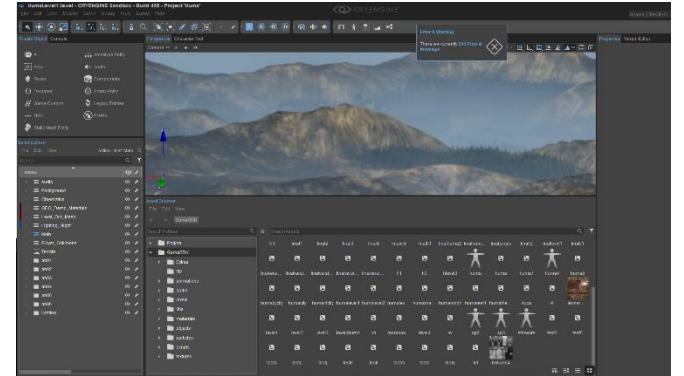


Figure 13 – IDE: CryEngine 5.6.5

## 2.3 Virtual Reality

Allows to provide a visual experiment, using specialized hardware like the VR glasses (Head mounted displays), data gloves or special installations known as CAVEs.

This type of development requires the use of specialized libraries and add-ons in order to treat the images accordingly so it can be seen on the display.

It takes in consideration the player position and orientation. For this project it will be used Vuforia that allows to recognize objects and images, it also has the function of recognizing text, it allows to create a 3D map it can be plugged in with Unity. Unreal Engine and unity support virtual reality.

According with [Zheng and Gibson 1998] there are 3 factors that any VR application should have, that is response to the user actions, the 3D graphics should be in real time, and there should be a sense of immersion.

The virtual reality over a high speed network can be a research tool because it can use big data sets, advanced graphics, real time processing among other possibilities.

The process of data visualization can be described as a sequence of fundamental processing steps [Haber & McNabb, 1990]

Simulation: the data sensing and measurement are used as an input

Data selection and filtering: The data obtained has to be analyzed and selected.

Visualization mapping: This data has to be transformed to primitives as well as their properties like colour and size.

Rendering: The primitives are rendered as images that will appear on the screen.

The virtual reality is an interface between man and the machine that simulates virtual images; it uses 3D dimensions generated by computer and can use several hardwares to interact like the VR glasses, it also provides a sensation of immersion.

## 2.4 Augmented Reality

Concerning related work like [Van Krevelen 2010] that is based on automatic calibration, human tracking, use of several displays and also a global positioning system. This project takes in consideration several UIs and allows to obtain different types of mixed reality, it also takes in consideration limitations like the human factor.

It is based on a mix of virtual elements in the real world, it uses the real environment and adds computer generated images.

Many games have been developed using this technology like Pokémon Go. Contrarily to virtual reality that places the human in a virtual world, it uses the real physics.

Engines like Unity and Unreal Engine allow to add these virtual elements, this project will use Unity and Vuforia to achieve this objective.

This paper aims to provide augmented reality games in an on line platform.

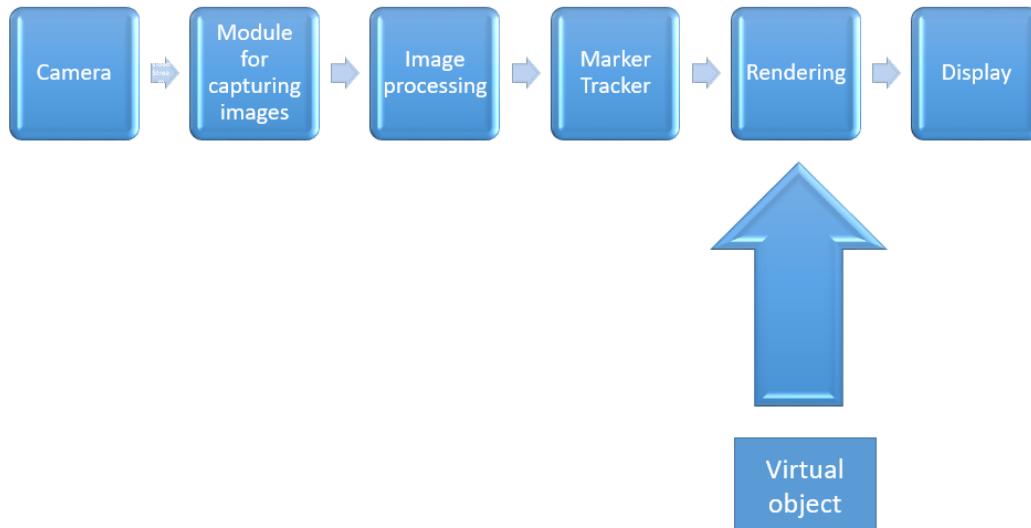


Figure 14 – Virtual object rendering

After analyzing several papers it was noticed that not many mention a web component on augmented reality projects and also the artificial intelligence in these kind of projects is not much advanced.

This project aims to research an improvement for augmented reality games. The web platform allows users to choose which game they prefer, where several types of game are displayed some are based on virtual reality and augmented reality.

Concerning related work [Van Krevelen 2010] basis on automatic calibration and human tracking.

## **2.5 Select and justify the development technology used**

This project is based on developing futuristic games and simulators, the technology used was Unity and Unreal Engine due to the nature of some requirements the programming languages used were C#, PHP, SQL, JavaScript among others...

Based on [Prof Manjula R 2016] it was identified several problems when producing a game like:

- Planning and crunch time (a term used for periods of extreme work overload, occurring usually closer to deadlines).
- Management problems.
- Difficulty to get an effective communication.
- Difficulty in determining certain elements like the fun of the game.
- Project estimation.
- Pre-Production phase.
- Prototyping.
- Getting tired during the project.
- Polish.
- Advertising.
- Quality standards.
- Tests.

It also mentions the need to optimize the tools and the pipeline in order to make the game profitable. On this work there is a detailed description of the pre-production, production and test stages.

Many of the new games available are based in unreal engine, but this project was made in order to look the most realistic as possible, strategy followed by many of the bigger companies in the game development area like Ubisoft and Electronic Arts.

The technology used was 3DS Max for modelling and rigging, the major part of the games were made in Unity, it was also used OpenShot and other movie editors.

It was decided to use mainly C# combined with C++, although developers vary on the opinion of which technology to be used.

Due to high graphic quality and processing effort, the games work fine on computers with fast internet connections.

The Unity version used was the 2018 since the 2019 does not support WebGL so well. Some of the games have a serious GPU consumption. There are several decisions needed when defining the development of an application, several factors must be taken into account, when choosing which technology should be appropriate to the operating system where it will be installed, as a general rule there should be at least two environments, a production environment and a test environment., preferably the databases should also be on different servers, another fact is licensing, like the use of .NET based technologies or databases like SQL Server (these have annual licenses). Before proceeding with the development the existing environment should be analyzed, many measures should be taken into account such as free server space, processing capacity, which web server is used. For example if one server is using IIS, it is not recommended to use Java applications.

Application architecture is another fact to keep in mind, there is currently a generalization of the MVC (Model, View, and Controller) architectural standard such as .NET MVC or Java Spring MVC.

Applications may be web based or standalone depending on the project requirements, via web requires the use of various components such as an application server, use of front end technologies and the interface to be compatible in a browser which requires that pages are created in formats like .HTML or .JSP or .ASPX among others so that it can be accessed by a browser respecting the protocols being used in network terms, like the Hyper Text Transfer Protocol.

The database requires in most cases the use of SQL queries among other possibilities, in order to be able to design a database it is first necessary to have a knowledge of the application functionalities in order to be designed in agreement with the project requirements. The objectives

and business needs should be clear so that information can be stored in a structurally efficient manner.

In most cases, a system can be viewed as a combination of hardware and software.

The system development life cycle can be applied to both technical and non-technical systems. It is a conceptual model that goes from an initial feasibility study to the maintenance of the application after it has been implemented, in this cycle there are various development phases and is widely used in project management.

## 2.6 Game Scrum

According with [Godoy 2010] game scrum is based on project Management,

Dividing tasks and other agile practices are adopted like XP to be more focused on the engineering side. The objective is to find a mix of Scrum and XP which allows to combine juniors with seniors members of the team and motivate the team to participate on the interactions and meetings of the scrum although the distribution of work in game development is not equally distributed during its iterations.

On the pre-production are defined the game objectives, the client requirements, the “fun factor”, the choice of programming language and platforms and it will guide the production phase. It is also common to develop prototypes and to find the ideal game concept and design on this stage.

After it should be defined the game design document which is a guide and has the project scope, the development and testing of the game. This document must have a comprehensive description of the game in all aspects in order to make the development easier. The game design document describes, objects, items, maps and characters, mentioning what they do, what's their purpose how they interact and their behavior in the game. It is a document based on continuous integration and it will become the product backlog at a later stage.

On the production stage the scope has already been defined and the game design document should be translated in a product backlog. Artistic tasks might have many specialists. In example the modeler gives his work to the animator. Possible use of Kanban and lean. On scrum, the work must be done on the end of the sprint. And the time of a task should be limited.

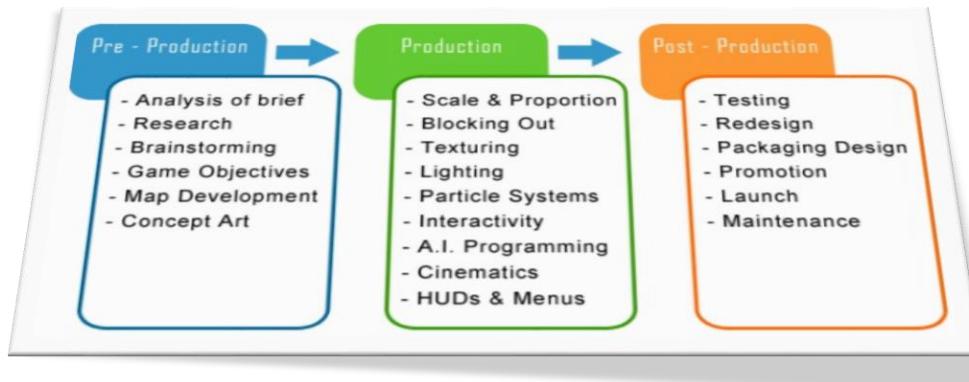


Figure 15 – Game scrum

On the post production stage gameplay is used as tests which should find the strengths and weaknesses of the development process. On this stage are defined estimations/predictions of game adjustments and improvements and 3 Aspects should be taken in consideration:

The good aspects: tools, strategies used.

The bad aspects: find bugs and problems; list of key issues.

The ugly: a list of critical issues that absolutely must be corrected.

On this stage should be measured the metrics in addition to the experiences: publisher and developer. And also the number of senior and junior developers, time of development and release date.

As conclusions due to costs it is important to know if the game will be profitable or not. An iterative methodology allows to have features ready sooner. An agile process lets you focus on what really matters.

Approaches of the Game-Scrum:

- Artistic content: Use of iterations.
- Project scope: Use of brainstorming, prototyping and the game design document
- Project management: using Scrum to develop
- Team organization: Prioritize, define members, skills among other factors...

## 2.7 Gamification

According with [Sebastian Deterding 2011], gamification is the application of game-design elements and game principles in non-game contexts. It can also be defined as a set of activities and processes to solve problems by using or applying the characteristics of game elements.

Gamification commonly employs game design elements to improve user engagement, organizational productivity, flow, learning, crowdsourcing, employee recruitment and evaluation, ease of use, usefulness of systems, physical exercise, traffic violations, voter apathy and more. A

collection of research on gamification shows that the majority of studies on gamification find it has positive effects on individuals. However, individual and contextual differences exist.

The gamification concept: Simulation based, task oriented gamification course, which makes it interactive and engrossing.

Examples like to map the game scenario to the context of risk management and the incremental learning was provided at each stage of the game as the learners took the challenges and overcame them.

Examples like incorporating a real work environment (visually), an element of challenge (bonuses and bombs), rewards for success (caps, badges), and learning through activities including elements of surprise and delight.

Examples like provide the learners the choice to seek support while performing the assigned task like in a real life scenario mapping to actual human behavior in such situations. This ensured a true simulated environment to encourage application of knowledge through performance.

Positive aspects:

- Diverse research on the heuristics
- Design patterns and dynamics of games
- Game-based systems.

Negative aspects:

- Integration with application contexts
- High CPU and GPU required
- Connection with the database

Use of video game elements to improve user experience and user engagement in non-game services and applications.

Areas where it is currently being used:

- Productivity
- Finance
- Health
- Sustainability

- News
- User-generated content
- Tutorials

Use of game processes like:

- points
- badges
- levels
- and leader boards

As a conclusion Gamification provides a better user experience, it allows the use of gamification processes like levels, scoreboard, etc... It can add quality to the applications. It can be used in several sectors. Allows users to better understand the program.

## 2.8 N-Tier Architecture

Based on [Channu Kambalyal 2010] the N-tier architectures have advantages in terms of development speed, scalability, performance and availability. These multi-layered architectures allow to standardize applications and facilitate their flexibility.

N represents the number of layers with 3 tier being the 3 layer model.

They are client-server architectures in which functions such as presentation, application processing and data management are physically separated. When separating an application into layers, developers have the option to change or add a specific layer instead of redoing the entire application. It provides a model by which programmers can create flexible and reusable applications.

The 3 tier architecture may or may not be followed in other software design patterns such as MVC (Model View Controller) which is common in .Net and Java in application development.

A three-tier architecture is typically composed of a presentation layer, an application layer, and a data storage layer and can be run on a separate processor.

The presentation layer is where application layout is displayed, corresponds to the highest level of the application where users can access directly from a web page or the operating system GUI (in so-called standalone applications). The main function of this layer is that the tasks and results are easy for the user to understand, includes the interfaces and what the user sees, thus communicating with the other layers of the network.

It is common for applications to have a front end that communicates with the page html, jsp, .aspx, .php, etc... Which helped with technologies like JavaScript, angular, thimeleaf, etc ... can give applications functionalities in the frontend and through a API send information via web services, where it is common for web applications to use the http post, get, put, and delete commands. The front end communicates with another application or part application commonly named backend which in turn sends to another application or part of the application named data layer.

The format of the uploaded data can be tested with programs like postman, where it is displayed generally in xml and json format depending on what is being used.

The application layer acts as a control of where business logic, rules, algorithms and other features are located, and is in an intermediate position, moving and processing data between the two adjusting layers. There are programs that facilitate the development in the second layer, as well as libraries, methods, classes, etc... And also facilitate communication between layers. This layer makes the coordination and control of the application, processes the commands, makes logical decisions, evaluates and performs calculations. It also controls the functionality of an application by performing detailed processing.

The Data Layer is the lowest layer and has a strong relationship with the database. The information is passed on for processing and then returned to the user. Information is saved and retrieved (backups, restores, etc...) from the database or file systems. Generally companies use relational databases, it includes data persistence mechanisms (database servers, file sharing etc...) and provides Application Programming Interface (API) to the application layer, which provides data management methods for the data stored in the database. Databases may also be non-relational NoSQL that are not limited to the use of tables and relationships and have different data structures (e.g. MongoDB).

It has better performance than a thin client approach and is simpler to manage than a thin client approach, improves reusability and scalability, as more requests or information are added, more servers can be added, reduced network traffic offers maintenance and flexibility. Some disadvantages are sometimes encountered in testing due to lack of testing tools, cost effectiveness and server availability is more critical.

According with [Norman G. Depledge 1995] a three-tier, object-oriented client-server architecture that includes an open systems transaction processing monitor can provide a basis for connecting users and customers to existing enterprise transaction processing systems.

Distributed architectures are based on availability, fidelity and transparency. A distributed system uses network-wide features where these resources transmit messages between them. These can be SOA (Service oriented architecture) or peer to peer applications.

Data or applications or part of applications may be stored on multiple computers or on computers (servers) spread across the network. A database can be distributed, or an application can use multiple databases.

Formerly objects were passed less consistently, distributed architectures provide greater flexibility and reuse.

Distributed architectures are flexible as they do not require technical expertise to access database reports.

By allocating resources in different parts of the network, they allow you to improve performance rather than having the one machine restriction.

Load balancing processes also allow you to distribute the workload between servers and how much they participate in the process, not being limited if one or more servers fail(s).

Allows authorized users to access database systems in different geographic locations. It is more flexible to users in terms of availability as they can access the same information more reliably.

In a distributed database the performance of the machines is maximized, since in some cases the databases are parallelized or replicated and there is load balancing between servers, there is also autonomy in which only data departments can access certain types of data. Systems can be removed or added to the database without affecting other systems.

Solutions like MVC because this architecture has flexibility allows the possibility to access different databases and applications can be changed without affecting other applications or processes being used by the same database.

They are also economically flexible and can cost a computer network less than other undistributed processes.

Distributed architectures in development make it easy for developers the maintenance, one example is that the application interface can be easily changed without affecting other parts of the application or the relation with the database.

There is also ease of changing drivers as well as processes for accessing data without impacting other areas.

There is an advantage in terms of development speed, performance scalability and availability.

Parts of the application can be developed without affecting others by allowing teams to focus on certain areas, like front end and back end.

One advantage in scalability / flexibility terms is that if there are too many requests in a given resource we can escalate the application server where that resource is allocated without affecting the others.

Using many computers and the existence of production and testing environments, among other factors, make it more malleable to make changes and also makes them more efficient and flexible to use, allows to minimize system failures by not having so much dependency on fewer resources.

The use of web services and APIs also make things more flexible as they allow to send messages between different applications without making things dependent.

The data is usually replicated so that it is in multiple locations so that when a site fails, (e.g. the server goes below the threshold) the data is still accessible in this type of architectures.

New sites can be added at different geographic points or removed that will not affect what they do elsewhere.

Many organizations have delegations at various geographic points, which means that the database is distributed at these locations.

Data distribution should also be taken into account because users in one place can access data elsewhere, usually accessing the nearest one.

Performance increases if data is placed near the most sought after locations and the use of parallel or distributed databases increases database access speed than having remote access to a central database.

Increased availability as database management systems are designed to function even if one fails. In case of failure it will be redirected to another in a different location.

One component in the business layer is easily accessed by many components in the presentation layer or even by different presentation layers.

Works well for big data scenarios that require high performance and complicated business rules. Separating the application into layers makes it easy to implement or reuse components.

Authorized users have access to data in the most different locations. Distributed architectures may support different types of databases like Oracle, MySQL, and SQL Server among others...

## 2.9 Web services

Web Services and Service Oriented Computing reduce the costs of building new systems for a variety of reasons. They allow communication between applications or part of the application so they can use existing applications and services without having to develop the application from scratch.

According with [Apostolos Zarras 2004] the web services are analyzed based on then UML and the Markov models.

In the development process, the use of Web Services allows the application to be divided into different layers, such as Front End, Back End and Data Layer, which has many advantages because you do not need to change the entire application when a bug or problem arises. Facilitates the development as it allows the bug fix to be treated only locally and does not affect the rest of the application.

This development methodology is based on the reuse of components obtained by the network. They do not have interoperability issues (compatibility between operating systems), as well as incompatibility between the development languages used and are not dependent on the characteristics of client computers, so they are cheaper and easier to incorporate.

Service Oriented Computing deals with modules and components that can be reused or integrated through interfaces that may have a specific purpose.

Web Services are a software module designed to perform a certain set of tasks. You use an address such as a URL on the World Wide Web to give access to your service. They can be seen as the way to do web integration using open standards such as XML, SOAP, and WSDL among others through an Internet protocol. A Web Service is a way for two machines to communicate over a network, and an API is a set of definitions and protocols that allows one application to communicate with another application.

XML or JSON among others are used to tag data, SOAP is a messaging protocol used to exchange structured information, WSDL is primarily for describing available services and UDDI is for listing which services are available. Mainly used as a mean of communication. These can be searchable over the network and can also be called properly, when called, Web Services can provide functionality to the customer who calls this Web service.

Web Services use standard web protocols (HTTP or HTTPS) to interoperate, communicate, and exchange data messages, which have a standard visible in Extensible Markup Language (XML) or JavaScript Object Notation (JSON), among others over the Internet. They can be tested using specific programs like Postman, SoapUI Pro, among others. Web Services use open standards that any part of the hardware or software program can access. Web services can be written once and used many times or in different ways. The open standards used by Web Services enable to connect processes more easily and reliably. Web services also allow to bring different systems together, avoiding the need to build new systems from scratch.

API is a very broad term. It is usually like a piece of code that communicates with another. In web development, the API usually refers to the way we retrieve information from an online service. The API documentation provides a list of URLs, query parameters, and other information on how to make an API request and informs what type of response will be given for each query.

Representational State Transfer (REST) is an architectural pattern based on a set of rules / standards / guidelines on how to create a web API. Since there are several ways to do this, having an agreed system of structuring an API saves time in making decisions as well as saves time in understanding how to use it.

RESTful refers to the implementation of Web Services that use the REST architecture, which can use the HTTP, GET, POST, PUT, and DELETE methods.

To understand service-oriented architecture (SOA), we must start with a clear understanding of the term service. A service is a function that is well-defined, independent and does not depend on the context or state of other services. Web services technology is the most common connection technology in service-oriented architectures.

The availability and effective utilization of these new features and capabilities requires the restructuring of many existing applications. SOA comes at a time when there was a pressing need for architectural conversion, to seize opportunities and adapt in the new age, best suited for building dynamic solutions such as e-business solutions.

User interaction - being able to provide a unique and interactive user experience.

Process integration - manage applications and services.

Information integration - manages and moves data.

Build to Integrate - creates and implements new applications and services.

It enable incremental deployments and asset migration which is rentable.

Numerous integration projects have failed due to their complexity, cost, and impractical implementation schedules. Including a development environment that will be built around a standard component, better promotes the reuse of modules and systems and also that the assets are migrated to the structure and implementation of new technologies. Today's businesses must be able to adapt flexibly and quickly to market needs, but even small process changes can involve a lot of work in various information systems. To be competitive, maintenance efforts must be reduced, but information systems must continue to evolve.

According with [Yousef Baghdadi 2012] there are difficulties matching the SOA with the business requirements.

SOA enables the transition from a silo-based system to a service-oriented system. It facilitates flexible coupling, the abstraction of the correspondent logic, flexibility, reuse and discovery. SOA is a type of architecture in which applications use services available on the network via protocols.

This architectural approach is particularly applicable when there are multiple applications running on various technologies and platforms that need to communicate with each other like the presented project. In this way, it is possible to mix and compare services to perform transactions with little programming effort. SOA is a logical way to design a software system to provide services to end users or other services distributed over a network through disposable interfaces, this architecture also defines an interaction between software agents as message exchanges between service requesters (users) and service providers. Clients are software agents that request the execution of a service. Vendors are software agents that provide the service.

SOA uses services available on the network and on the web. Allows reuse and are built based on services. They can allow reuse of existing applications without interoperability or development language issues.

It is important to note that services:

- are independent and perform predetermined tasks.

- can be dynamically discovered.

Composite services can be built from the aggregation of other services. Web services allow internal and external systems integration and can be used as tools to build new information systems or improve existing ones, or add functionality to existing ones. They have universal standards making them cheaper and easier to use than adopting licensed software.

Web services may interconnect with other web services. They provide cost savings because they can share and communicate data without influence from the operating system, programming language used by applications or computer features, and enable business to business relationships.

There is a very common use by different companies to communicate via Web Services where data from one company is sent or received to another company. This implementation can be done in several ways that facilitate the process of data communication between computers of different companies.

Web services allow you to exchange information and can build web applications with open standards and enable communication. They have no code costs and allow the interface between applications.

In this process format tags (in XML, JSON...) are used where the characteristics of objects are generally described, allowing to manipulate and interpret data automatically and perform operations without human intervention. Thus allowing the exchange of data.

The browser or software must follow a set of rules programmed to use and display the data. Data is carried over protocols.

In a service-oriented architecture there are several services communicating with each other in an orderly manner to create functional software. Some business features are executed through the calls and receipts of these services. This methodology if it is web can be used internationally.

Developers can use these services in other applications. Many companies like Microsoft offer tools, such as Visual Studio, that let you build Web Services. Another IDE also used to create Web Services is Eclipse.

## **2.10 Web development**

In order to develop via web it is common to use databases and APIs, the browser has to be able to open the page and process the information.

According with [Jakus 2010] if using HTML it will have to combine with CSS to make the page layout and esthetical aspects among JavaScript for functionalities and Angular if being used to execute the commands.

Nowadays web applications use Web services to communicate with parts of the application or other applications. On the HTML code there are several inputs or tags that allow to use the forms that were used in this project. For this project was used WebGL in order to place the games online.

The application was divided in 3 tiers, and the data layer makes the interface with the database. According with [Norhaidah A. Haris 2019] in her study pointed out that PHP is the programming language more used which is the one this project is based. It mentions the several advantages of the MVC architecture like security.

The process of comparing frameworks is not easy and several criterias should be taken in account like debugging, how it connects with the database among many other factors.

Many technologies can be used in web development like .NET java Spring MVC, on the frontend it can be used Angular. It is important to choose a technology, .NET has several particularities.

Concerning web development it is important to configure the servers according with the type of application that is being developed.

PHP appeared in 1995 by Rasmus Lerdorf and can be used inside the HTML pages, it has been used in large scale especially on the backend side,

## 2.11 Churn predictions

The system will record all the user's activity on the web platform, below one of the tables of the database mentioning user activity:

- Opções		<input type="button" value="T"/>	<input type="button" value="id"/>	<input type="button" value="username"/>	<input type="button" value="password"/>	<input type="button" value="created_at"/>
<input type="checkbox"/>  Edita	<input type="checkbox"/>  Copiar	<input type="checkbox"/>  Apagar	1	kuca	Portugal79	2019-08-18 04:30:22
<input type="checkbox"/>  Edita	<input type="checkbox"/>  Copiar	<input type="checkbox"/>  Apagar	2	test	\$2y\$10\$k3/z7eVm3U5bq6rf6VWr5.FiDI2ZFYk0x1CVdAEKv7a...	2019-08-18 04:33:32
<input type="checkbox"/>  Edita	<input type="checkbox"/>  Copiar	<input type="checkbox"/>  Apagar	3	test2	\$2y\$10\$4//yUYJUgNxFzAheYrBHnONbc8pW99kt6C/t/kNi/9U...	2019-08-18 16:59:14
<input type="checkbox"/>  Edita	<input type="checkbox"/>  Copiar	<input type="checkbox"/>  Apagar	4	123	\$2y\$10\$6gwxg7o01FS/gpmdlgl7LbepoWbb/GICfZmEe4ZaXvp...	2019-08-18 18:43:45
<input type="checkbox"/>  Edita	<input type="checkbox"/>  Copiar	<input type="checkbox"/>  Apagar	5	test9	\$2y\$10\$/mGjRvbSdQpoUonji3fzi.TjUoOCF46S/Ri.ohUIMFN...	2019-08-18 19:36:35
<input type="checkbox"/>  Edita	<input type="checkbox"/>  Copiar	<input type="checkbox"/>  Apagar	6	test14	\$2y\$10\$cysvE5nuiAJ0Jb6UlJSdCOTonvIE/er9UXhCqV08PBf...	2019-08-18 20:48:01
<input type="checkbox"/>  Edita	<input type="checkbox"/>  Copiar	<input type="checkbox"/>  Apagar	7	test19	\$2y\$10\$VP52QaKnmF7lgSQKEIx82.35SJIAjgMES74XRNoxe2l...	2019-08-18 20:51:03
<input type="checkbox"/>  Edita	<input type="checkbox"/>  Copiar	<input type="checkbox"/>  Apagar	8	scmoro	\$2y\$10\$HsHbUcmBTj322D8KQwqx9eHodBophBWSld/TsI3UTu2...	2019-08-20 11:33:32
<input type="checkbox"/>  Edita	<input type="checkbox"/>  Copiar	<input type="checkbox"/>  Apagar	9	test99	\$2y\$10\$xwaBQ0VyjEhlH0oQ1ujXz.8oljzbumb.JpojRSu/A6o...	2019-09-01 01:40:32

Figure 16 – User table of the database

This data will allow to do predictions.

According with [Emiliano G. Castro 2015] Wavelets are basic functions, building blocks to represent signals in time and frequency simultaneously.

Alfréd Haar developed the first discrete wavelet transform:

$$\psi(x) = \begin{cases} 1, & \text{for } 0 \leq x < 0.5 \\ -1, & \text{for } 0.5 \leq x < 1 \\ 0, & \text{for other values of } x \end{cases} \quad (6)$$

$$\phi(x) = \begin{cases} 1, & \text{for } 0 \leq x < 1 \\ 0, & \text{for other values of } x. \end{cases}$$

$\psi(x)$  Represents the Haar mother and  $\phi(x)$  represents the Haar father.

The wavelet power spectrum is calculated by:

$$\text{WPS}[0] = (s_{(\log_2 n),1,1})^2$$

$$\text{WPS}[i] = \sum_{k=1}^{2^{i-1}} (d_{(\log_2 n)-i+1,1,k})^2 \quad (7)$$

Where it ranges between 1 and  $\log 2 (n)$ . Therefore, the number of components of the WPS from an  $n$  elements data array is  $\log 2 (n) + 1$ .

$$t_0 \xrightarrow{\Delta t_P} t_1 \xrightarrow{\Delta t_F} t_2 \quad (8)$$

$t_0$  is the login time,  $t_1$  is the current time and  $t_2$  the predicted end of session

The profit is given by:

$$\begin{aligned} \Pi^v = N\alpha & [\beta\gamma(CLV - c - \delta) + \beta(1 - \gamma)(-c) \\ & + (1 - \beta)(-c - \delta)] - A. \end{aligned} \quad (9)$$

- 1)  $N$ : the total number of players;
- $\alpha$  2) : the fraction of players targeted for the retention campaign;
- $\beta$  3) : the fraction of the targeted players who would churn;
- $\delta$  4) : cost of the campaign's retention incentive (per player);
- $\gamma$  5) : the fraction of the targeted players who would churn that decided to keep playing the game because of the campaign's incentive (success rate);
- 6)  $c$  : cost for contacting the player to make the offer (the incentive);
- 7)  $CLV$ : a customer lifetime value (i.e., the value received by the game publisher if the player is retained);
- 8)  $A$ : fixed administrative costs of the retention campaign. The profit generated from running a single retention campaign.

## 2.12 Future predictions

This project aims the development of a web platform which allows users to play different types of games. It will allow to check what users have done when using this website. It has a strong component of web development, game development and database development. It was used several APIs and it also uses visual programming.

This system allows to check which games were played and when, it is working online so it is accessible to any user that has a computer with a browser and connection to the internet. It has a user profile management and a game management option.

Several games were implemented like in example: first person shooters, adventure games, third person shooters, flight simulators, car race, car fights, and tanks among other. The main objective is to make the system the most simple possible for users and avoid crashes and bugs.

On the front end several technologies were used like HTML, CSS, Javascript that can be used as a template for future projects, most of the project was hand coded, the project aims to have a visual close to reality.

The majority of the games were made based on C# and C++, several engines and technologies were used like Unity and Unreal Engine, it tries to be the most captive possible in visual terms and the easy to play.

Tests are going to be made to users so we can make an analysis on which type of games people prefer in general, which are easier to play, how they interact, which are the strong and weak points according to the user point of view. An initial mockup was presented in order to show what is going to be implemented and allowing a general view of the global project.

It will be used techniques to find the player's churn [Emiliano 2015], predict the number of hours played and also statistical analysis of which games users prefer, which day time is more used in order to provide developers the information. Predictions will be made using statistical methods and comparisons with the values observed. Datasets will be developed that allow to analyze the information of the logs. The survival time for each player will be analyzed

## **2.13 On line gaming platforms**

According with [Mark Claypool 2006] it is measured the latency and for which actions the latency is higher in computer games. Games are classified in different categories and the details are analyzed.

Based on [A. Shaikh 2006] it is presented a logical platform architecture, how data is stored, the use of application server for a service platform for online games, it is showed several

steps of the architecture as well as measures, a game deployment service that this project also uses and the use of auxiliary service platforms.

Currently there are many online game platforms like Steam, Miniclip and Kongregate.

Steam offers the possibility to download an application which allows to play games inside the computer and have better graphic quality and speed and it is owned by Valve Corporation.

Miniclip allows to play games just with the browser some require to install the flash add in or the unity add in.

The online gaming platform developed for this project allows to play games without installing anything on the computer, it only requires a browser and an internet connection.



Figure 17 – Steam

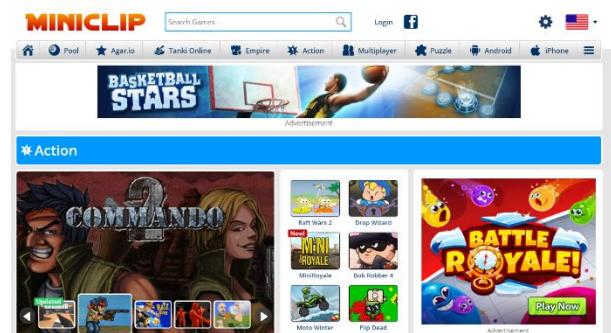


Figure 18 – Miniclip

Paco Games developed an online platform with extensive use of Unity, it provides games of several types and the graphic quality is high in some cases, the games are free to play but there are lots of advertisements which are the main profit of this platform, in order to play any of the available games the player has to watch an advertisement for a few seconds.

Zynga has games like poker which is so available on Facebook also among other games that can be played via browser and others have to be downloaded for mobile devices or to the computer in order to be played. The company that holds this platform is Valve Games with a total equity of 2.5 billion dollars, the company was formed by former Microsoft employees and made games like Half-life and Counter-Strike.

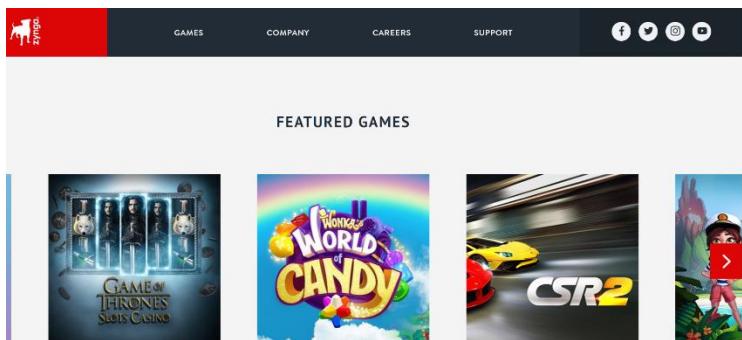


Figure 19 – Zynga

William Hill is a bookmaker company that has an on line betting platform that allows to place bets in different games, in 2018 had a revenue of £1,621.3 million, it offers the possibility to play games like live casino among others.

Epic games the company that developed Unreal Engine and many games like Fortnite and Unreal tournament, in 2018 was estimated to be worth 4.5 billion dollars. Some games have to be downloaded and installed in the computer or PlayStation or mobile device and once installed gives the opportunity to play on line with other players.

Ubisoft and Electronic Arts are also know companies in the gaming industry, these companies developed some of the most famous games in the world like FIFA or Assassins Creed among many others, Ubisoft has a total assets of 2.8 billion and Electronic Arts had a revenue of 15 billion in 2018.

Activision Blizzard has a total assets of 17.8 billion and Take-Two Interactive Software, Inc a total asset of 4.2 billion.

Companies like Lego, Warner Bros and Gameloft have also made strong investments in the game development industry.

## 2.14 Computer games

Many of the games presented on the platform are first person shooters, in this dissertation will be demonstrated the game ARMA due to the high graphic quality, developed by Bohemia Interactive a Czech company that developed their own game engine and nowadays is adapted for

virtual reality. As per the picture below ARMA III is a very realistic game when compared with previous first person shooters / third person shooter games.



Figure 20 – ARMA III



Figure 22 – ARMA IV

[Wright 2002] mentions that some first person shooter games are so advanced that you enter in a virtual society, full of rules, objectives and many other factors and the most important thing is to have fun.

Another game that should also be highlighted due to its realism is F1 2019, developed by the English company Codemasters. On the picture below a view of the game.



Figure 22 – F1 2019

Article	Reference	Goal	Method	Findings
Games and Big Data: A Scalable Multi-Dimensional Churn Prediction Model	Paul Bertens, Anna Guitart and Africa Perianez	Predict the market values via estimating when a player quits which level, hours played	Use of Cox prediction model to estimate the time a player quits. There is a playtime and a level model.	To help developers know when a player quits, so they can improve their games. The dataset should be stored in a database.
Game Data Mining Competition on Churn Prediction and Survival Analysis using Commercial Game Log Data	EunJo Lee, Yoonjae Jang, etc...	Analyze the logs of 10 000 players. Predict the churn point	Use of datasets and data mining techniques. Use of deep neural networks	Development of a framework for game data mining using the logs of current companies. The data is always changing on online games, the test servers and the test data sets have to be more accurate.
General Video Game AI: a Multi-Track Framework for Evaluating Agents, Games and Content Generation Algorithms	Diego Perez-Liebana, Ahmed Khalifa, Julian Togelius	Comparison between different general AI game frameworks and identify approaches for AI and procedural content generation	Use of several methods like the monte carlo tree search based on Markov decision process, evolutionary algorithms and hybrids	Allows to evaluate the performance of video game playing agents and for testing algorithms to create game content. In the future might help game designers to display content.
StarCraft Bots and Competitions	David Churchill, Mike Preuss, Florian Richoux	Development of bots to play StarCraft	Several techniques were used for each bot, some of the several examples are use of 3 layered abstraction and divide and conquer,	Analysis of the performance of different AI techniques, definition of strategies. Not able to develop a total winning bot.

development of specific algorithms					
Motion Gaming AI using Time Series Forecasting and Dynamic Difficulty Adjustment	Guillaume Lample, Devendra Singh Chaplot	AI that predicts the human action via time series forecasting	Methodologies used by determining the position of the player. Use of UBI1 algorithms	Balance the player's health and the AI Health	the game taking in consideration the AI Health
Improving Exercise Balance and Enjoyment					
Churn Prediction in Online Games Using Players' Login Records: A Frequency Analysis Approach	Emiliano G. Castro and Marcos S. G. Tsuzuki	With the login data being able to do predictive analysis	Use of discrete wavelet transform, algorithms and time base frequency	Helps to target the players during the considered period	the target the considered period
Visualization using game engines	Dieter Fritsch, Martin Kada	Make realistic scenarios with game engines	Use of different engines and comparison	Still is hard to get existing spatial data into a game.	
Eliminating Distance in Scientific Computing: an Experiment in Televisualization	Robert B. Haber, David A. McNabb, Robert A. Ellis	Visualization of the results in scientific computing	Use of distributed computers	There are still obstacles in the use of supercomputers	
Virtual reality	Zheng, JM; Chan, KW; Gibson, I	Bring to virtual reality response to user actions, real-time 3-D graphics and a sense of immersion.	Use of perceptual model.	challenges hardware, software, human factors and delivering virtual reality over high-speed network.	concern
A Survey of Augmented Reality Technologies, D.W.F. van Krevelen and R. Poelman		Technologies to enhance the visual perception	Use of several type of displays, use of a global positioning	Developers still have to improve their work taking in consideration	

Applications and Limitations			system, use of limitations like the different UIs and 3D human factor pointing.
Game Development. Harder than you think	Jonathan Blow	Minimize difficulties of developing a game like too many algorithms available, other risk factors like project size	Use of tools, special workflows, use of third party components and multi-platform development. Games are difficult to develop but the appearance of new technologies facilitates the work of the developers and make it more appealing
Software Engineering Challenges in Game Development	Prof Manjula, Chakradhar Raju M , Sai Chand M	Describe the several steps of the game development process, like game play, concept, feature list, story, target audience, requirements and schedule, staff & budget estimates.	Divide the work in several stages in time like pre-production, production, testing among the choice of technology. Mentions that most games are in C++ and C and also the use of API, OpenGL and Direct X. The management should adjust to the game development process, the tools and pipeline should be optimized in order to make the game more efficient.
Game-Scrum: An Approach to Agile Game Development and Gamification: Using Game Design Elements in Non-Gaming Contexts.	André Godoy Ellen F. Barbosa	Adjust the game development to agile methodologies	Definition of several stages, adjust agile methodologies to the game development process. An iterative methodology allows to have features ready soon and thus discover and work the “fun” of the game is easier,
Gamification: Using Game Design Elements in Non-Gaming Contexts	Sebastian Deterring, Miguel Sicart, Lennart Nacke	Use gaming methodologies and objects in other areas. Apply game	Use of game concepts like maps for applications of time management, Provides a better user experience and can be applied in several sectors.

			methodologies in use of the game other contexts that are not only game based.	menus, use of scoreboards in management activities among other methods and strategies...
3-Tier Architecture	Channu Kambalyal	Achieve flexibility, increase performance, provide more security	Comparison between different types of architectures showing advantages and disadvantages for each	Comparison of different technologies using different types of architectures.
An Open, Distributable, Three-tier Client-Server Architecture with Transaction Semantics	Norman G. Depledge, William A. Turner, Alexander Woog	A distributable, three-tier client-server architecture for heterogeneous, multivendor environments	Mix of technologies and products in a 3 tier architecture to facilitate the process of developing new application	A three-tier, object-oriented client-server architecture that includes an open systems transaction processing monitor can provide a basis for connecting users and customers to existing enterprise transaction processing systems
A methodology for web services-based SOA realization	Youcef Baghdadi	Reduce the complexity of SOA implementation	Use of a methodology of web services	The use of architectures with several levels might not be coherent

					with the business objectives.
Model-Driven Dependability Analysis of Web Services	Apostolos Zarras, Panos Vassiliadis, Valérie Issarny	Develop a methodology for the analysis of web services. Investigate the dependability analysis of composite services	Extend the UML and also use of the Markov models to perform analysis	In the future develop a quality analysis Web service that actually performs this task for the designers.	
A new model for the selection of web development frameworks:	Khaoula Benmoussa, Majida Laaziri, Samira Khoulji	Find the PHP Framework more adjusted to the type of objectives wanted	Comparison of the frameworks	Frameworks were measured via different criterias	
PHP Usability in Web Application Development	Norhaidah A. Haris, Nurdatillah Hasim	Help developers in choosing the framework	Several metrics and scores attributed	Several platforms have differences that might affect the development.	
New technologies for web development	regia Jakus, Matija Jekovec, Sašo Tomažič	Development adjustments and processes to the new web technologies	Adjust to HTML5	Use of new features like WebSocket protocol.	
On Latency and Player Actions in Online Games	Mark Claypool, Kajal Claypool	Check where there is latency on online games, like for which player actions, etc...	Speed measurement for different actions, use of models for analysis of the data obtained.	Online games that use the player controller have more latency than games that use the Omnipresent model.	

On demand platform for online games	A. Shaikh, S. Sahu, M.-C. Rosu, M. Shea, D. Saha	Develop of a prototype of a service platform for games that are available online.	Definition of several parts of the online game platform, server connection and management, use of design principles	In the future to add enhancements to the service platform.
-------------------------------------	--	---	---	--

Table 18 – Article references

### 3. Research Gap

This project has a credit system on an online gaming platform which currently exists inside some games but not in an online platform and also provides statistical information about user performance and a profile management module, where user can upload information like photos, game reviews, among other possibilities.

These characteristics are very important because it will make the platform more attractive to users, they are able to check what they have been doing and can store information, which has some similarities with the profile management that Facebook uses.

Another innovative aspect of this project is that it allows to play virtual reality games and augmented reality games online, which is something new and allows to provide more happiness to users as they can play new type of games just by clicking in one link. It aims to be the very simple in usability terms so it can reach a higher number of users.

These new technologies are very important because most likely will be highly used in the future, concerning the augmented reality it can combine the real world with virtual computer generated objects, giving a higher sense of reality to gaming. It also has an algorithm that changes the AI behavior according to the current status of the player, in case the user has difficulties the AI will pass automatically to an easier level so it can

This project aims the development of several types of games and a user system which allows to manage their preferences and see which games were played and when. It has several functionalities, implements simulators, car games, different types of AIs among other components.

It has:

Database to store the users information
Simulators of vehicles
Action Games
Management of user's activities
Login system
Web based
APIs for CRUD and visualization of data
User profile management
Adventure games
Video and animation
Credit System
Augmented Reality
Virtual Reality
Predictions
Statistical analysis

Other characteristics important to refer are:

- Different types of Artificial intelligences
- Starting Interface / Dashboard
- Game selection
- Several cameras and several points of view
- It has the option to drive different vehicles
- Car AI, human AI, among others...

- ✚ Possibility of seeing what the user has done
- ✚ HUD, target, Health, Weapon change, shoot
- ✚ Etc...

The system has a menu interface with a video. It has a dashboard and a user profile menu. The system should have the functionalities of shooting, changing weapon, vehicle race simulation, run and crouch.

The HUD should display current and maximum levels of health and other relevant information.

It should have the functionalities of choosing different scenarios.

It has several types of games, statistics and user modules...

The opponents should have an intelligent system of following points on the scenario that are not visible, it has AI path, motion and the AI characters have different behaviors.

It should have the functionality of changing view, by moving the camera behind the player (third person), and a second one that is the first person camera.

It has the functionality of restarting the level every time the player dies.

The functionality of returning to the main menu.

The functionality of showing a congratulations panel when the objectives are fulfilled.

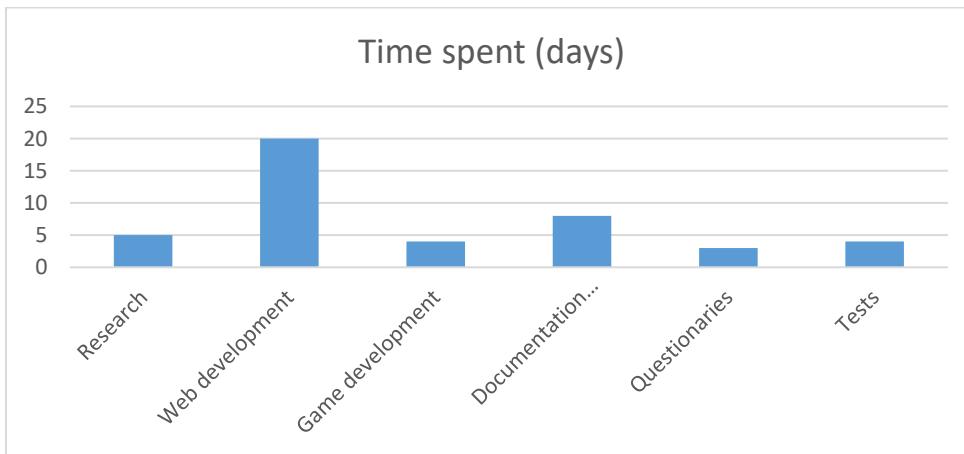
Should be user friendly.

## 4. Planning

The research will be time consuming, other activities like analyzing on line companies will have a high time consumption and are expected to be concluded during February.

The development activities will follow and are expected to take around 20 days. The development activities are divided in types, API development, web development, game development and database development.

The expectation is that when the system is complete to perform questionnaires and tests with around 20 users.



Scheduling made with Microsoft Project:

Task Mode	Task Name	Duration	Start	Finish	Predecessors	2020	02 Feb '20	09 Feb '20	16 Feb '20	23 Feb '20	01 Mar '20	08 Mar '20													
						T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	
Normal	Research	4 days	Sat 01-02-20	Wed 05-02-20																					
Normal	Documentation elaboration	5 days	Thu 06-02-20	Wed 12-02-20																					
Normal	Web programming	10 days	Sun 16-02-20	Thu 27-02-20																					
Normal	Database development	5 days	Fri 10-01-20	Thu 16-01-20																					
Normal	Game development	6 days	Sat 22-02-20	Fri 28-02-20																					
Normal	Questionaries	3 days	Wed 01-04-20	Fri 03-04-20																					
Normal	Tests	4 days	Fri 01-05-20	Wed 06-05-20																					

The project is expected to be implemented around May and the documentation is expected to be delivered around the month of May also.

## **5. Expected results, conclusions and future work**

The expected results are to simulate futuristic game scenarios close to the reality. For different time frames the AI will have the information of where the player is and the best process to fulfill an efficient attack. The levels have several AI areas. Variables like speed, health and ammo will be taken in account.

The system will measure the number of AI elements and will also mention how AI should proceed, there will be correlations between variables and statistics values for different distributions that will be generated for the different scenarios. It will measure the player performance for each level and will look for the optimization of time. It will present several displays and graphs for the different agents...

The game industry has been developing in several areas, and new technologies should be used in order to stay competitive. Many companies are investing in game technology like Google.

User experience can be optimized in web gaming platforms and future predictions can provide estimations, but will never provide the exact value.

In the future it is expected an increment use of AR and VR, future mixes of hardware and software to provide a better experience. There are many competitors and games graphic quality will increase.

As a conclusion, the online gaming platforms can be improved, there are several functionalities that can be added to the current systems that allow improvement in several terms like usability, graphic quality, statistical analysis, among other factors. It is expected to develop a platform with several functionalities that allow users to have more functions like a profile where they can store images, game data, a credit system which allows credit sharing with other users, the higher the number of credits the higher the access to new levels

## 6. Bibliographic references

- AmpatzoglouIo, Apostolos (2010).Software engineering research for computer games: A systematic review in: *Information and Software Technology*
- Arsenault, Dominic (2009) Video Game Genre, Evolution and Innovation in *Eludamos. Journal for Computer Game Culture*. 2009; 3 (2), p. 149-176
- Aukstakalnis, Steve (2016), *Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR* (2<sup>nd</sup> ed.). Washington, Addison-Wesley Professional.
- Bethke, E. (2003). *Game development and production*. Wordware Publishing. isbn: 1556229518
- Billinghurst, M., Clark, A., & Lee, G. (2015). A Survey of Augmented Reality. *Foundations and Trends® in Human-Computer Interaction*. 8(2-3): 73-272.  
Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE Multimed.*, vol. 19, no. 2, pp.4–10, 2012.
- Busby, Jason & Parrish, Zak (2010). Mastering Unreal Technology, Volume II: Advanced Level Design Concepts with Unreal Engine 3. Sams Publishing.
- Cao, J., Xia, T., Li, J., Zhang, Y., & Tang, S. (2009). *A density-based method for adaptive LDA model selection*. *Neurocomputing*, 72(7), 1775-1781  
<https://doi.org/10.1016/j.neucom.2008.06.011>
- Chang, KyuSik (2007). *Video Game Console Audio: Evolution and Future Trends*. 156-756
- Chesñevar, C. I., Maguitman, A. G., & Simari, G. R. (2006). Argument-based critics and recommenders: A qualitative perspective on user support systems. *Data & Knowledge Engineering*, 59(2), 293–319. doi:10.1016/j.datark.2005.09.005
- Claudia, M. & tom DieckTimothy Jung (2019), *Augmented Reality and Virtual Reality: The Power of AR and VR for Business* (1<sup>st</sup> ed.). Springer US.
- Craig A. Lindley (2003) Game Taxonomies: *A High Level Framework for Game Analysis and Design*. Gamasutra.

- Eric Klopfer and Kurt Squire. Environmental detectives the development of an augmented reality platform for environmental simulations. *Educational Technology Research and Development* 56:203–228, 2008. 10.1007/s11423-007-9037-6.
- Fay, J. J. (1993). *Encyclopedia of Security Management - Techniques and Technology*. (1st Edition), Butterworth-Heinemann
- Fromme, Johannes & Unger, Alexander (2012). Computer Games and New Media Cultures: A Handbook of Digital Games Studies. Springer
- Hawkes, R. (2011). *Foundation HTML5 Canvas: For Games and Entertainment*. Friends of ED.
- Johns, Jennifer (2006). Video games production networks: value capture, power relations and embeddedness in *Journal of Economic Geography*, Volume 6, Issue 2. 151–180. DOI: <https://doi.org/10.1093/jeg/lbi001>
- Johnson, Daniel (2001). Computer Games with Intelligence
- Laudon, C.L., Laudon, J.P. (2018). *Management Information Systems - Managing the Digital Firm*. (15th Edition), Pearson Education.
- Lim, S. and Reeves, B. (2010). Computer agents versus avatars: Responses to interactive game characters controlled by a computer or other player. *International Journal of Human-Computer Studies*, 68(1):23 – 4
- M. Zyda et al., (2005) From Viz-Sim to VR to Games: How We Built a Hit Game-Based Simulation. In *Organizational Simulation: From Modeling & Simulation to Games & Entertainment*, W.B. Rouse and K.R. Boff, eds., Wiley Press, 2005, pp. 553-590.
- M. Zyda and D. Bennett (2002), *The Last Teacher*, Visions, US Dept. of Commerce.
- Morgan, Graham (2009), *Challenges of Online Game Development: A Review*. Doi: <https://doi.org/10.1177/1046878109340295>

- Navarrete, Cesar C. (2013), Creative thinking in digital game design and development: A case study in *Computers & Education*
- Porter, M. E., & Heppelmann, J. E. 2017. Why Every Organization Needs an Augmented Reality Strategy. *Harvard Business Review*, December: 46-58.
- Ronald T Azuma. (2016). The most important challenge facing augmented reality in *PRESENCE: Teleoperators and Virtual Environments* 25, 3 (2016), 234–238.
- Rozanski, N., Woods, E. (2005). *Software Systems Architecture - Working with Stakeholders Using Viewpoints and Perspectives*. 4th Edition, Pearson Education
- Rhee, M. Y. (2003). *Internet Security - Cryptographic Principles, Algorithms and Protocols* (1st Edition), John Wiley & Sons Ltd
- Schoenau-Fog, Henrik (2011). Evaluation of Continuation Desire as an Iterative Game Development Method
- Sundar, S. S. 2007. Social psychology of interactivity in human–website interaction. In A Joinson (Ed.), *Oxford handbook of Internet Psychology* (p. 89–102). Oxford, UK: Oxford University Press.
- Tipton, H. F., Nozaki M. K. (2016). *Information Security Management Handbook*. 6th Edition, CRC Press
- UX Planet. (2017). Mobile UX Design: Key Principles – UX Planet. [online] Available at: [https://uxplanet.org/mobile-ux-design-key-principles-dee1a632f9e6`](https://uxplanet.org/mobile-ux-design-key-principles-dee1a632f9e6)
- Xian, Jiajun & Yang, Dang (2018). *The evolution of people's interest in video games*. University of Electronic Science and Technology of China
- Wright, Talmadge (2002). *Creative Player Actions in FPS Online Video Games*
- Yannakakis G.N., Togelius J. (2018) Modeling Players. In: *Artificial Intelligence and Games*. Springer, Cham, 203-250, doi: 10.1007/978-3-319-63519-4\_5
- Young, R. M. and Riedl, M. (2003). Towards an architecture for intelligent control of narrative in interactive virtual worlds. In *Proceedings of the 8th international conference on intelligent user interfaces*, pp 104–132. ACM.

- Zheng, J.M. & Chan, K.W. & Gibson, I. Virtual reality (1998), *Virtual reality*, DOI: 10.1109/45.666641

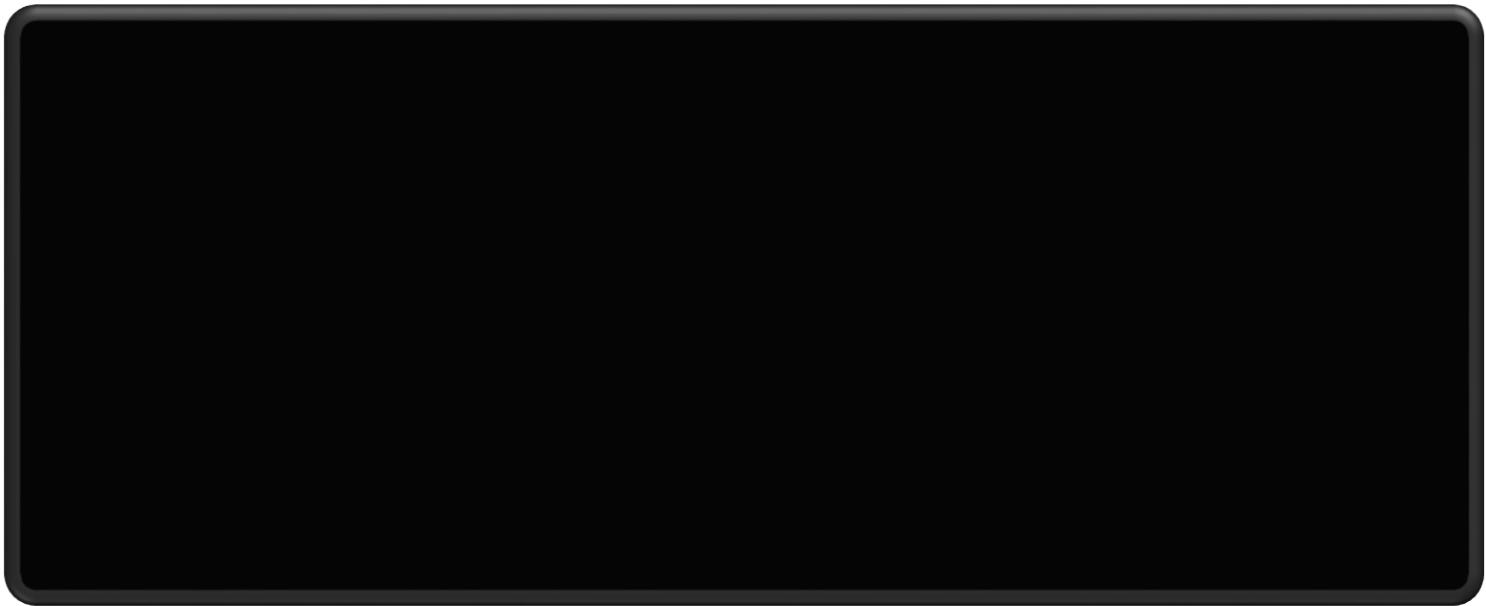
## References

- Blizzard Entertainment. Starcraft, 2010: <http://us.blizzard.com/en-us/games/sc> [03/01/2020]
- Electronic Arts <https://www.ea.com/pt-pt> [04/01/2020]
- Identifying Requirements, The Right Way  
<https://www.modernanalyst.com/Community/CommunityBlog/tabid/182/ID/3757/Identifying-requirements-the-right-way.aspx> [21/12/2019]
- Manifesto for Agile Software Development: <https://agilemanifesto.org/iso/ptpt manifesto.html> [21/12/2019]
- Miniclip: <https://miniclip.com/games/pt/> [10/01/2020]
- Paco Games <https://www.pacogames.com/> [05/01/2020]
- Requirements. Why is it important?  
<https://steelkiwi.com/blog/requirements-why-it-important/> [21/12/2019]
- Spelling, terminology and markup.  
<http://gcc.gnu.org/codingconventions.html#Spelling> [21/12/2019]
- Steam <https://store.steampowered.com/> [14/01/2020]
- Subway Map to Agile Practices.  
<https://www.agilealliance.org/agile101/subway-map-to-agile-practices/> [21/12/2019]
- Top 4 software development methodologies <https://www.synopsys.com/blogs/software-security/top-4-software-developmentmethodologies/> [21/12/2019]
- What are Web Services?  
[https://www.tutorialspoint.com/webservices/what\\_are\\_web\\_services.htm](https://www.tutorialspoint.com/webservices/what_are_web_services.htm) [21/12/2019]
- What is Service-Oriented Architecture (SOA)? <https://www.visual-paradigm.com/guide/development/what-is-service-orientedarchitecture/> [21/12/2019]

## 11. Attachments A - Games Portfolio



Gameplay:



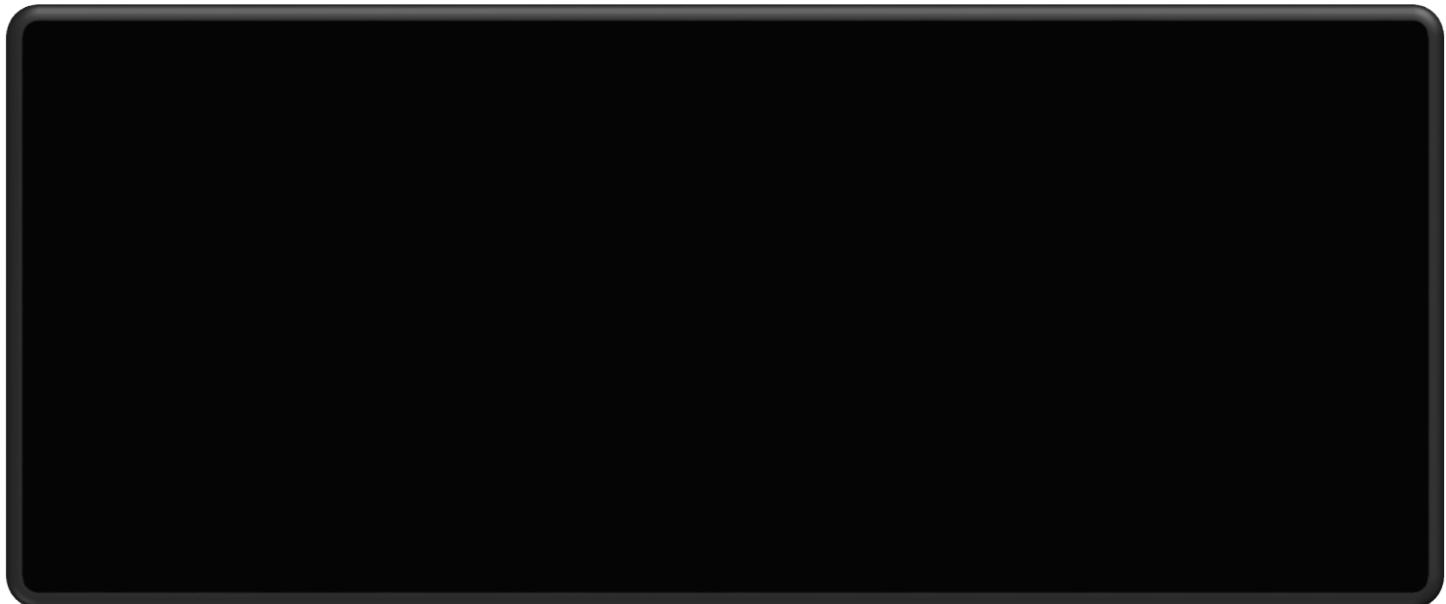
Programming languages used: C++ / Engines / IDE's: Visual Studio 2005 and Unreal Engine

4.22.3 / Type: First Person Shooter / Third Person Shooter / Requirements: Web Browser

Database: SQL Server



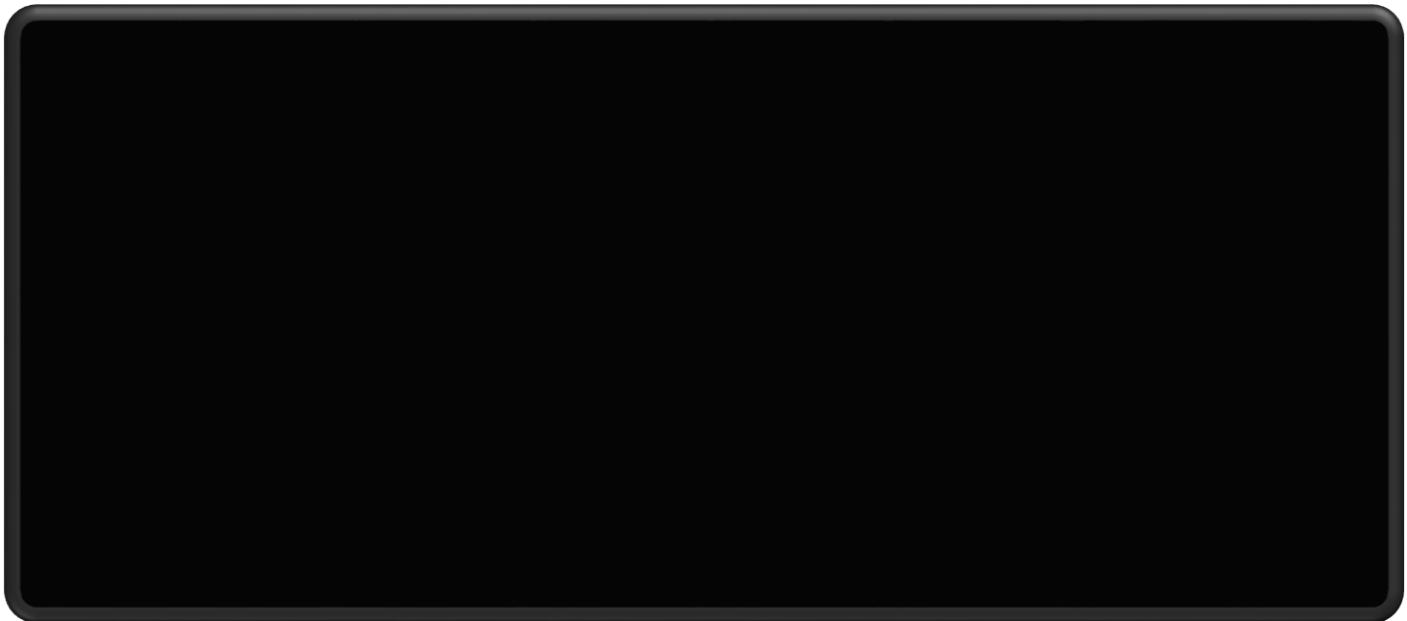
Gameplay:



Programming languages used: C# /Engines / IDE's: Unity 2018, MonoDevelop, Visual Studio  
2019 /Type: First Person Shooter / Third Person Shooter / Requirements: Web Browser



Gameplay:

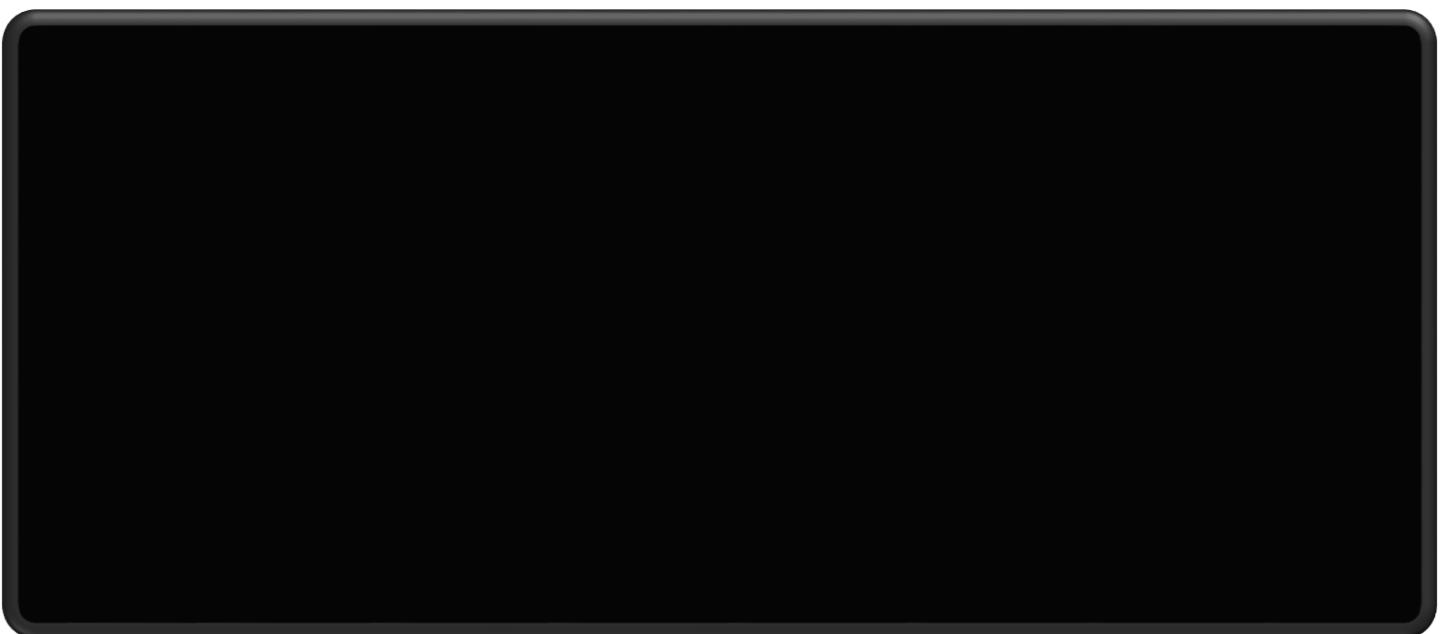


Programming languages used: C# / Engines / IDE's: Unity 2019, MonoDevelop, Visual Studio  
2019 / Type: First Person Shooter / Third Person Shooter / Requirements: Web Browser



*By the night...*  
VEL

Gameplay:



Programming languages used: C++ / Engines / IDE's: Visual Studio 2005 and Unreal Engine  
4.22.3 / Type: First Person Shooter / Third Person Shooter / Requirements: Web Browser



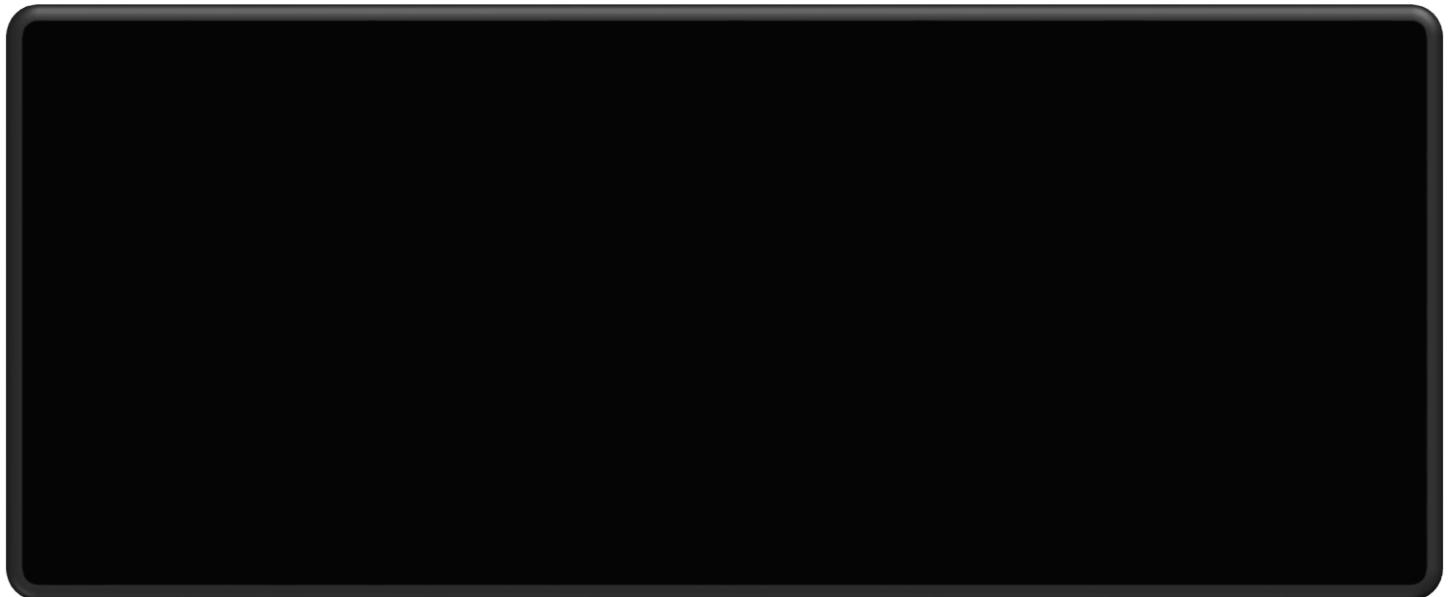
Gameplay:



Programming languages used: C# /Engines / IDE's: Unity 2018, MonoDevelop, Visual Studio  
2019 /Type: First Person Shooter / Third Person Shooter / Requirements: Web Browser



Gameplay:



Programming languages used: C# /Engines / IDE's: Unity 2018, MonoDevelop, Visual Studio  
2019 /Type: Car racing / Requirements: Web Browser

# HUMA

# HOWA

Gameplay:



Programming languages used: C++, LUA, ActionScript, and C #, API /Engines / IDE's:

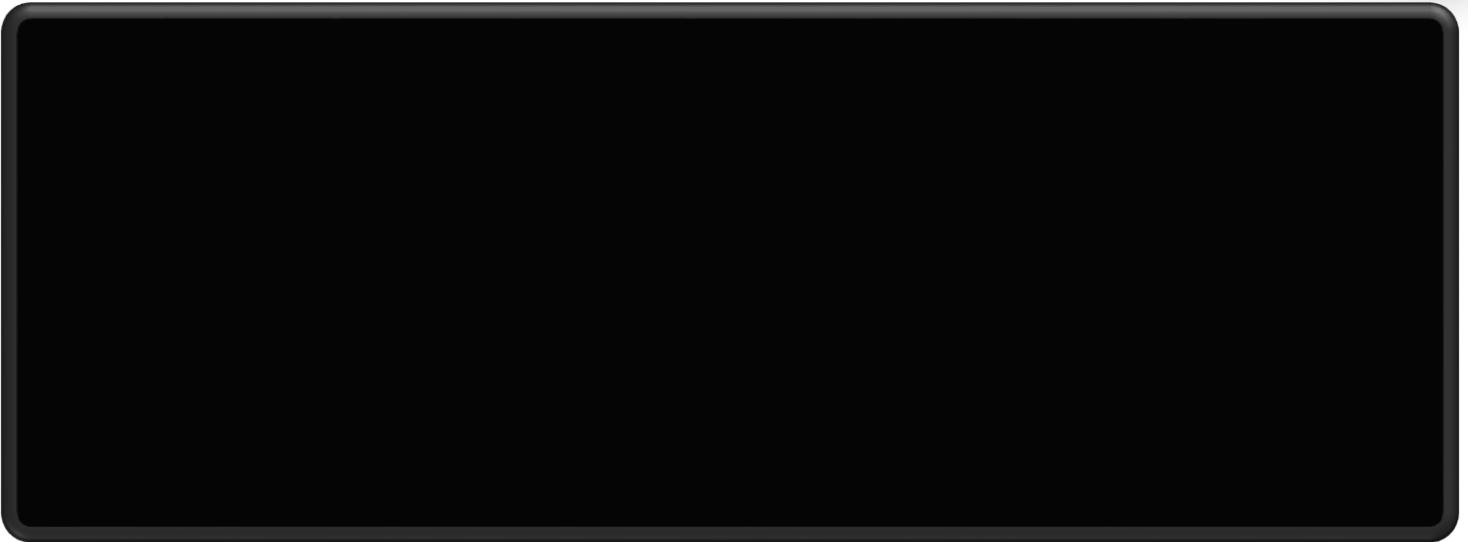
Cryengine, 3D Max, Visual Studio 2019, After effects / Type: Car racing / Requirements: Web

Browser





Gameplay:

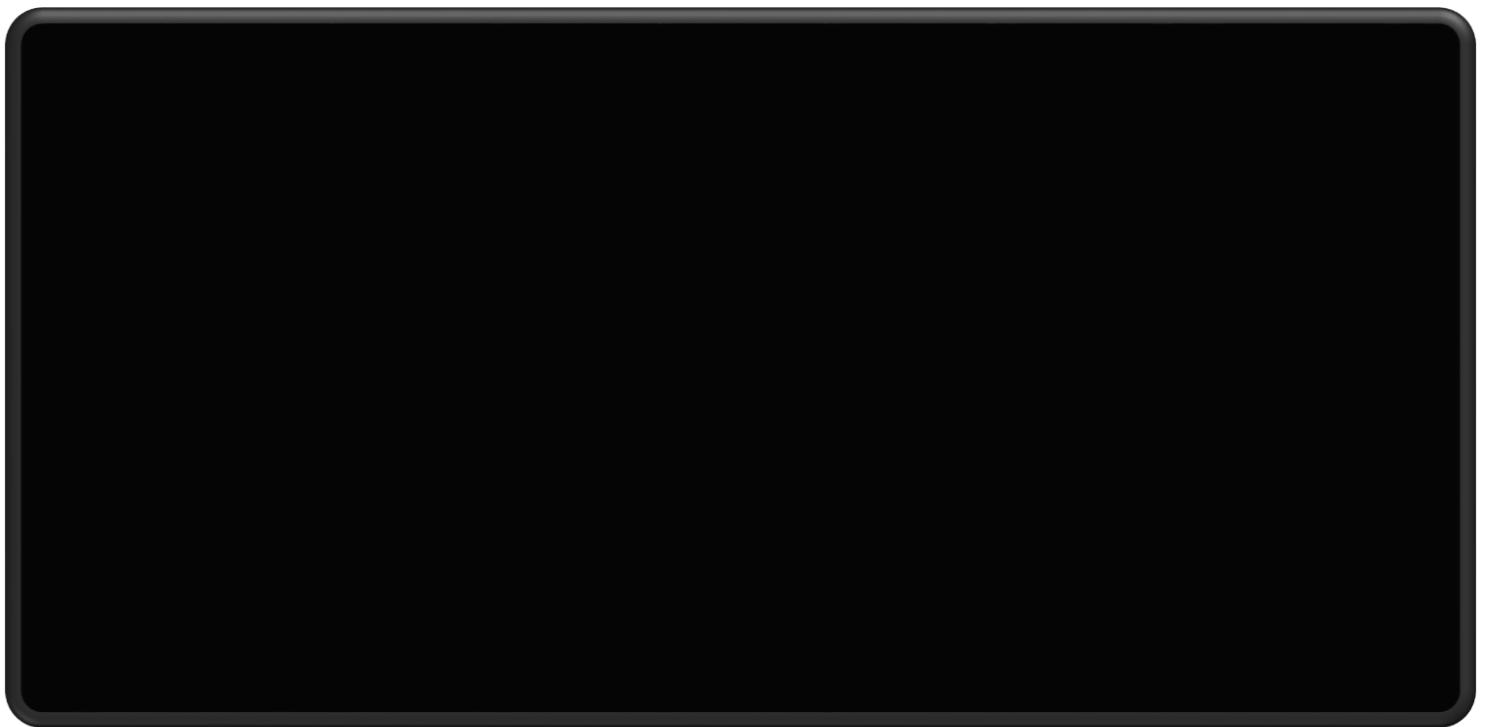


Programming languages used: C++, LUA, ActionScript, C#, API /Engines / IDE's: Cryengine, 3D Max, Visual Studio 2019, After effects / Type: First Person Shooter / Third Person Shooter  
/Requirements: Web Browser

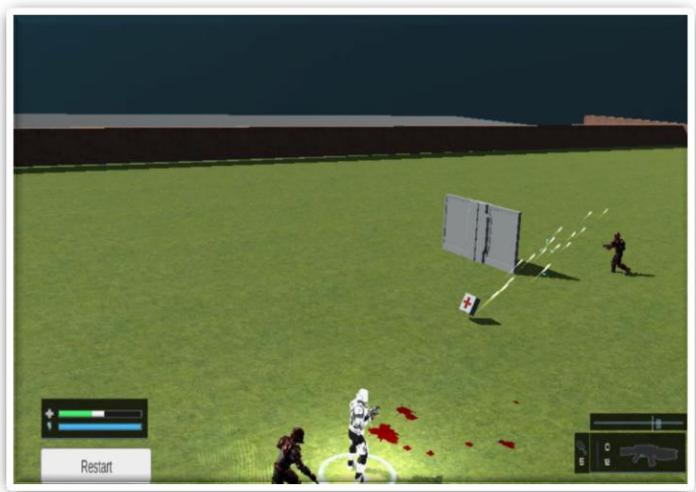
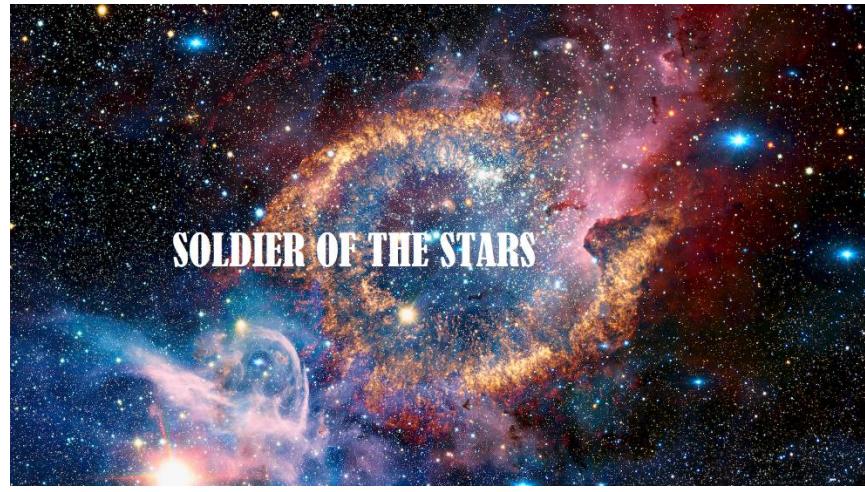
# AND THEN THERE WERE NONE



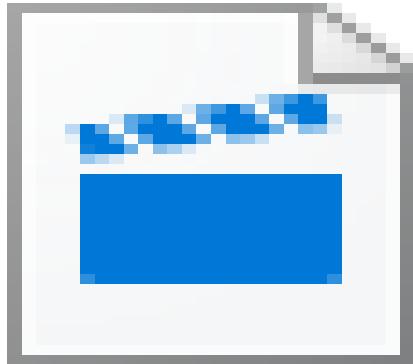
Gameplay:



Programming languages used: C# /Engines / IDE's: Unity 2018, MonoDevelop, Visual Studio  
2019 / Type: First Person Shooter / Third Person Shooter / Requirements: Web Browser



Gameplay:



# Star.mp4

Programming languages used: C# /Engines / IDE's: Unity 2018, MonoDevelop, Visual Studio  
2019 /Type: First Person Shooter / Third Person Shooter / Requirements: Web Browser



# Porradas.mp4

Programming languages used: C#

Engines / IDE's: Unity 2018, MonoDevelop, Visual Studio 2019

Type: Action / Fighter

# DROID WARS



Gameplay:



Programming languages used: UnrealScript programming language

Engines / IDE's: UDK, Flash

Type: First Person Shooter / Third Person Shooter / Requirements: Web Browser

# B A T T L E   T A N K S



Programming languages used: C++, LUA, ActionScript, C#, API

Engines / IDE's: Cryengine, 3D Max, Visual Studio 2019, After effects

Type: Car racing / Requirements: Web Browser

## **B- User Questionnaire Guide**

### **Introduction and objectives**

---

First we would like to show your gratitude for the participation in these tests.

As mentioned previously this test is under the scope of the Master thesis of Informatics engineering, in which the objective is to create an applications that is useful and easy to use.

We have the responsibility of creating an application called VEL (Velocity Engineering Light) that is based on online gaming.

This session is divided in 3 parts, the first in answering a questionnaire where some personal data will be asked, this will be used for an analysis and comparison between different uses of the application, the second part is the core of the test where it will be asked to use the application freely, and the third part a questionnaire about the evaluation and how you feel concerning the web portal.

### **Task**

---

You know that now is possible to play online without any costs, please use the application freely to play games.

## C – Right grant form

### **Consentment**

---

I, \_\_\_\_\_, carrier of the Passport n°  
\_\_\_\_\_ authorize the audio and video recording during the test session of the  
application VEL for studying purposes.

\_\_\_\_\_, \_\_\_ of December 2019

---

(Signature)

## **D – Pre-Test Questionnaire**

### **Initial questionnaire**

---

**Sex:** \_\_\_\_ (M or F)

**Age:** \_\_\_\_

**Professional situation:** \_\_\_\_\_

**District of residence:** \_\_\_\_\_

**Nationality:** \_\_\_\_\_

## **E - Questionnaire after test**

### **Final questionnaire**

---

For the following questionnaire answer each affirmation with the value that more adjusts to your experience. It will be used a scale from 1 to 5 where 1 means “I totally disagree and 5 means “I totally agree”.

1. The application VEL is useful. \_\_\_\_
2. The application functioning is very complex. \_\_\_\_
3. The application is easy to use. \_\_\_\_
4. I need the help of someone with technical knowledge to use the application. \_\_\_\_
5. The several functions of the application are well integrated. \_\_\_\_
6. The applications shows a lot of inconsistency. \_\_\_\_
7. The users will learn quickly to control the application. \_\_\_\_
8. The application is very confusing. \_\_\_\_
9. I am satisfied with the application. \_\_\_\_
10. It was necessary to learn many new things in order to use the application. \_\_\_\_

## F- Requirements Identification

Requirements ID	Type	Priority	Requirements description
FR1	Functional, System	High	The system should allow an interface for user to play game, define variables and parameters.
FR2	Functional	Low	The system should register the number of players per level
FR3	Functional System	Medium	The system should register the health level, current weapon, ammo, and speed among other variables.
FR4	Functional System	High	The system should allow to control several type of characters and vehicles to move around the scenario.
FR5	Functional	High	The system should register the time when a game started and the player data.
FR6	Functional	Low	The system should have the possibility to change from first person view to third person view.
FR7	Functional	High	The system should be able to upload photos and other information's to the correspondent user profile.
FR8	Functional	Medium	The system should allow a credit system allowed to be changed with other users.
FR9	Functional	High	The system should have an option to choose different type of games.
FR10	Functional	High	The system should allow to play against different characters and vehicles (AI).
FR11	Functional	High	The system should simulate realistic values.
FR12	Functional	High	The system should register the value of the variables.
FR13	Functional	High	The system should allow to run, jump and crouch.
FR14	Functional	High	The system AI should attack or compete with the player.
FR15	Functional	Medium	The system should have more than one type of enemies.
FR16	Functional	High	The system should provide predictions.
FR17	Functional	High	The system should have a credit system.

<b>FR18</b>	Functional	High	The system should have buttons to allow the player to move left, right, down or up, among others...
<b>FR19</b>	Functional	High	The system should allow the player to rotate according to the mouse.
<b>FR20</b>	Functional	Medium	The system should show current health and ammo for the different weapon.
<b>FR21</b>	Functional	High	The system should allow to play multiple games.
<b>FR22</b>	Functional	High	The system should allow an interface for user to play game, define variables and parameters.
<b>FR23</b>	Hardware	High	The system should work with a browser and one computer
<b>FR24</b>	Hardware	High	For virtual reality the user should have the respective hardware, in example VR glasses

Table 4 – Functional Requirements

#### Environmental requirements

<b>SERVER</b>		<b>Category</b>
<b>ID</b>	<b>Description</b>	
RQ1	Fast Processor	Hardware
RQ2	More than 2 GB DDR2 of RAM	Hardware
RQ5	Long Storage Unit	Hardware
	Windows XP	Software
	SQL Server	Software

Table 5 – Environmental requirements for the server

<b>User</b>		<b>Category</b>
<b>ID</b>	<b>Description</b>	
RQ1	At least a processor Pentium 1.2 megahertz (MHz)	Hardware
RQ2	More than 128 megabytes (MB) of RAM	Hardware
RQ3	At least 2 GB of hard disk availability	Hardware
RQ4	Widows [XP   2000   98]	Software
RQ5	[Internet Explorer 11   Mozilla Firefox 4.0 ]	Software
RQ6	Internet Connection	Telecommunications

Table 6 – Environmental requirements for the client computer

Development		
ID	Description	Category
RQ1	Visual Studio 2005, Eclipse, Unity208, Unity 2019, Unity 5, Unreal Engine, Cry engine, Visual Studio 2019	Framework
RQ2	MYSQL	Programming Language
RQ5	3DMax, Maya, Blender	Model of Development

Table 7 – Development requirements

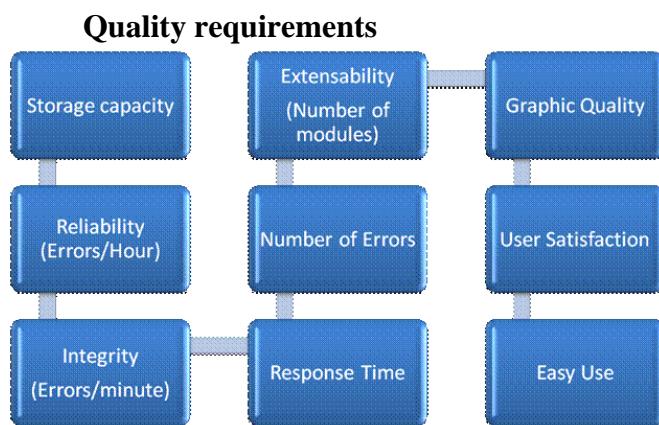


Figure 9 – Quality requirements

### Specification of the quality requirements

QR1	
Description	Storage capacity
Metrics	TB
Test	Maximum number of users allowed
Worse case	0.5
Planned	1
Category	Storage

Table 8 – QR (Quality Requirement) 1

QR2	
Description	Scalability
Metrics	Accesses/minute
Test	Tests of charge
Worse case	1
Planned	50
Category	Performance

Table 9 – QR 2

QR3	
Description	Fidelity
Metrics	Errs/hour
Test	Charge test
Worse case	1
Planned	0
Category	Performance

Table 10 –QR 3

QR4	
Description	Integrity
Metrics	Errs/minute
Test	Charge test
Worse case	0,007
Planned	0,0001
Category	Performance

Table11 – QR 4

QR5	
Description	Response time
Metrics	Transactions/milliseconds
Test	Tests of response time
Worse case	2
Planned	10
Category	Performance

Table 12 – QR 5

QR6	
Description	Stability
Metrics	Errs
Test	Intensive usage by qualified and specialized users concerning application data.
Worse case	1
Planned	0
Category	Performance

Table 13 – QR 6

QR7	
Description	Extensibility
Metrics	Number of additional modules
Test	Test the integration of new models
Worse case	5
Planned	NA
Category	Support

Table 14 – QR 7

QR8	
Description	User satisfaction
Metrics	% of satisfied users
Test	Questionnaires were made
Worse case	65%
Planned	95%
Category	Satisfaction

Table 15 – QR 8

QR9	
Description	Ease of use
Metrics	Tempo gasto sem progresso (em segundos)
Test	Utilização intensiva por parte de utilizadores "leigos"
Worse case	7
Planned	3
Category	Usability

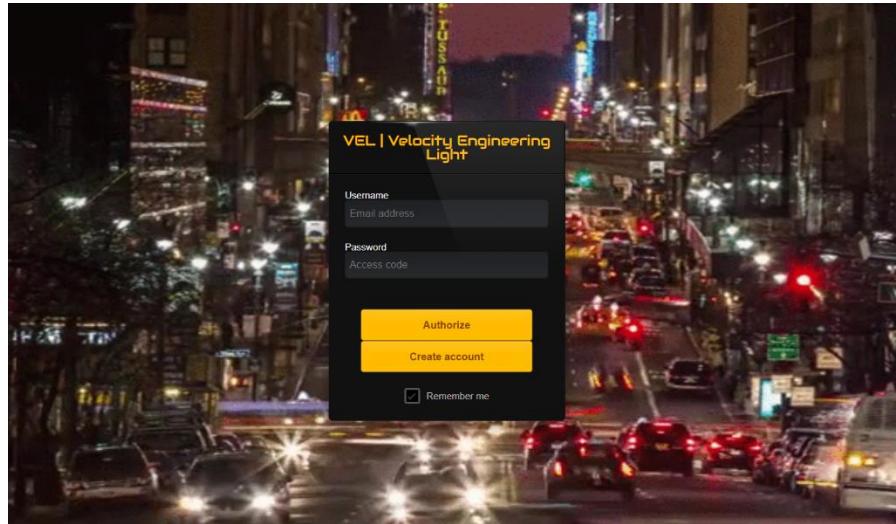
Table 16 – QR 9

QR10	
Description	Aesthetic appreciation
Metrics	Users/day
Test	Performance of questionnaires
Worse case	65%
Planned	95%
Category	Satisfaction

Table 17 – QR 10

## G - Login system

The system has a login functionality implemented which allows to know which games the users played, and they are also able to edit their profiles.



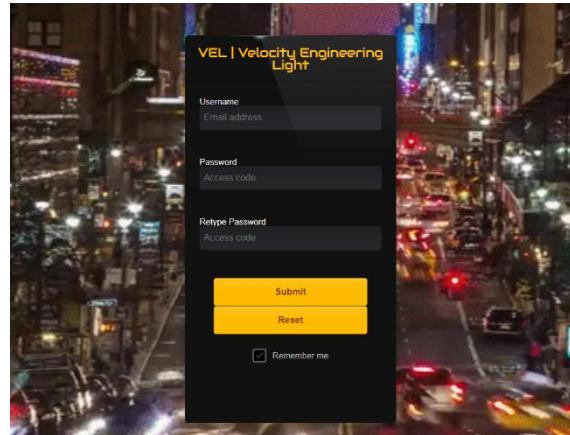
When an account is created, it will be inserted the values on the database, allowing to see when users created their accounts and which games they have played.

Opções								
		id	username	password	created_at			
<input type="checkbox"/>		<a href="#">Copiar</a>		Apagar	1	kuca	Portugal79	2019-08-18 04:30:22
<input type="checkbox"/>		<a href="#">Copiar</a>		Apagar	2	test	\$2y\$10\$k3/z7eVm3U5bq6rf6VWr5.FiDI2ZFYk0x1CVdAEKv7a...	2019-08-18 04:33:32
<input type="checkbox"/>		<a href="#">Copiar</a>		Apagar	3	test2	\$2y\$10\$4//yUYJUgNxFzAheYrBHnONbc8pW99kt6C/t/kNi/9U...	2019-08-18 16:59:14
<input type="checkbox"/>		<a href="#">Copiar</a>		Apagar	4	123	\$2y\$10\$6gwxg7o01FS/gpmldg7LbepoWbb/GICEfZmEe4ZaXvp...	2019-08-18 18:43:45
<input type="checkbox"/>		<a href="#">Copiar</a>		Apagar	5	test9	\$2y\$10\$/mGjRvbSdQpoUonji3fzi.TjUoOCF46S/Ri.ohUMFN...	2019-08-18 19:36:35
<input type="checkbox"/>		<a href="#">Copiar</a>		Apagar	6	test14	\$2y\$10\$cysvE5nuiAJ0Jb6UIJSdCOTonvIE/er9UXhCqV08PBf...	2019-08-18 20:48:01
<input type="checkbox"/>		<a href="#">Copiar</a>		Apagar	7	test19	\$2y\$10\$VP52QaKnmF7lgSQKEIx82.35SJiAjgMES74XRNoxe2l...	2019-08-18 20:51:03
<input type="checkbox"/>		<a href="#">Copiar</a>		Apagar	8	scomoro	\$2y\$10\$HsHbUcmBTj322D8KQwqx9eHodBophBWSId/Tsl3UTu2...	2019-08-20 11:33:32
<input type="checkbox"/>		<a href="#">Copiar</a>		Apagar	9	test99	\$2y\$10\$xwaBQ0VyjEhlH0oQ1ujXz.8oljzbumb.JpojRSu/A6o...	2019-09-01 01:40:32

As per the database table showed it is possible to observe that the passwords are encrypted in the database table, this allows to provide security, it also appears on the last column, the date the login is occurred. This information will allow to measure the time played, when the user logged in the system which days, the will allow to place this information on the user profile, so contrarily to other on line gaming platforms users will be able to see which games they played and for how long as well as other information concerning their performance.

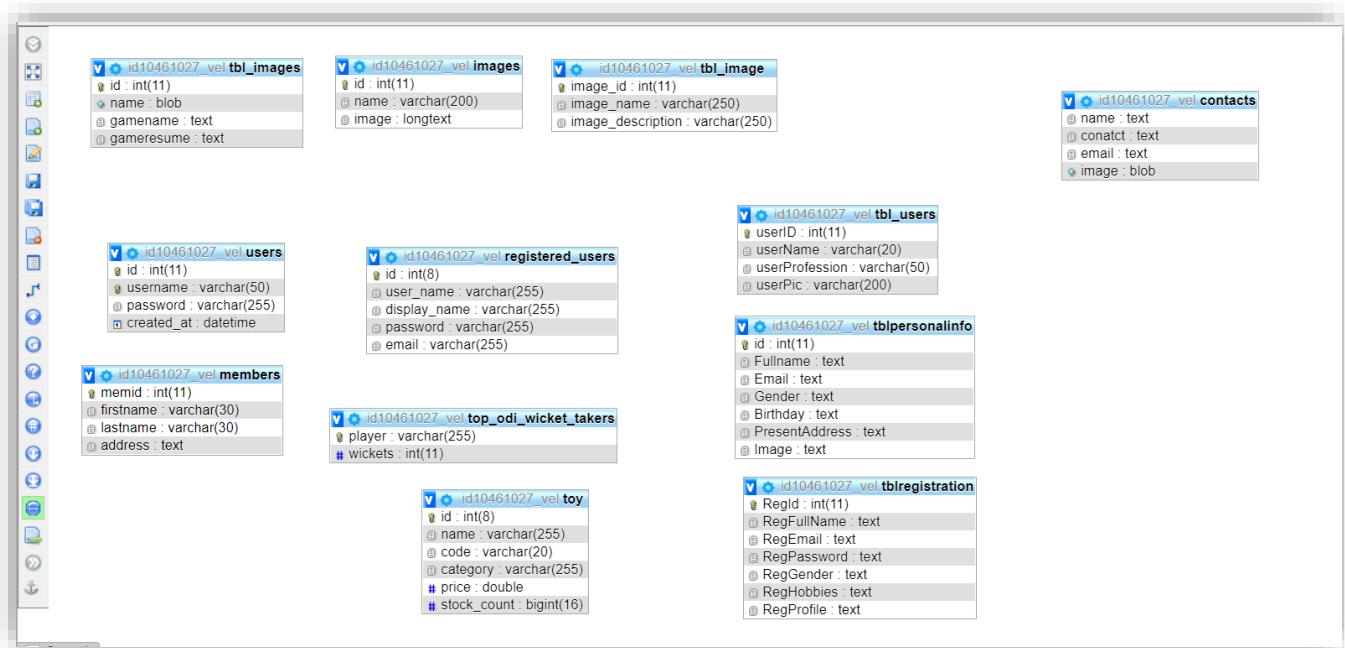
Each user has an amount of credits that can be shared with other users, these credits allow access to new levels, opens and skills, to make the platform more interesting to play.

To create an account it is asked the username and the password 2 times in order to increase security:



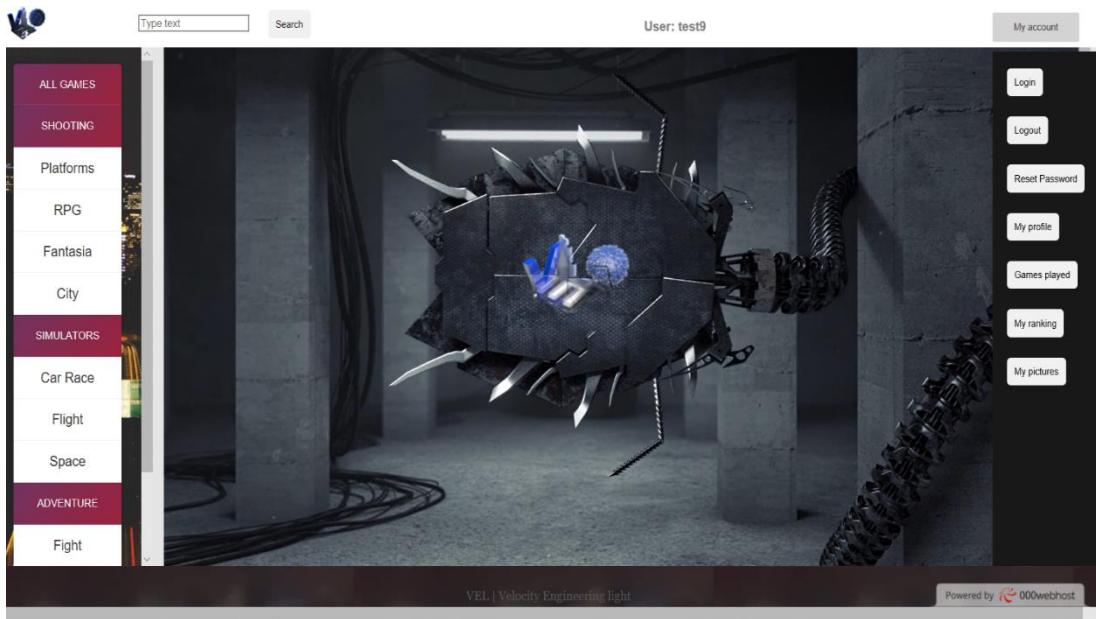
# H - Database

Below a view of the initial database in MySQL:

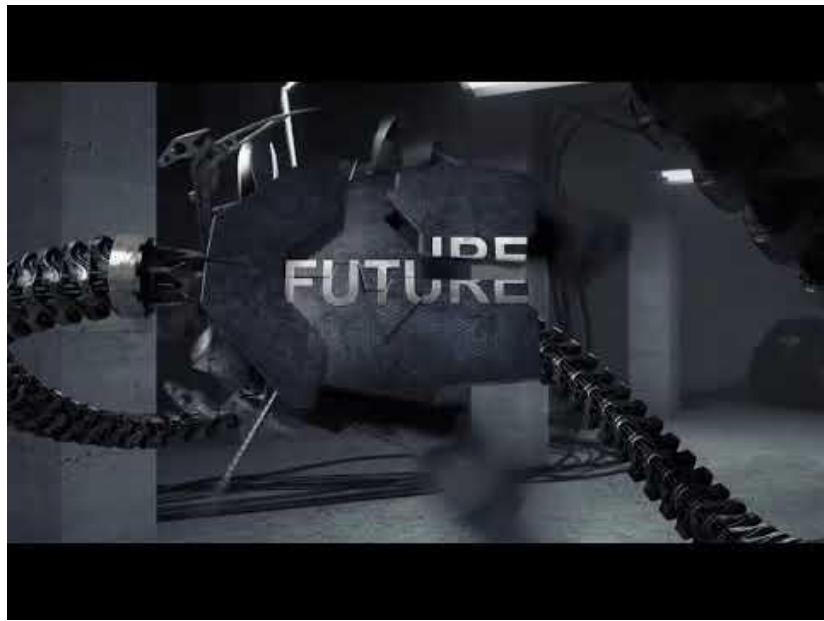


# I - Web platform layout

This is an initial demonstration of the layout of the platform.



Below a demonstration of the background video:



## **J - System restrictions**

This theme was chosen due to the technical requirements of making a game it uses different engines mainly unity and unreal engine and also uses 3DS Max and Adobe Flash, the application runs on Windows x86.

During the development of this project several sprints were performed where it was defined the objectives of the project.

Nowadays the tendency of playing computer games with high graphic capacity is high, there are several cases of success like flappy birds, angry birds, Minecraft, and the games from electronic arts, game loft and warner bros, among many other cases.

Due to the timeline restriction, the project has a limited number of games and characters, one of the scenarios is a city and occupies too much space and processing capacity.

The city level should run ok on higher capacity computers, although one of the objectives of the project is that the application works fine for different types of computers and browsers.

When pressing some buttons the processor takes some time to execute for example loading levels, because the levels have lots of information that delay the execution, although this delay time will not occur in computers with higher processor capacity and fast internet connectivity.

The games use new strategies and responses.

## K - Methodology/Development

The problem lies in developing an online platform that supports high-quality gaming as well as augmented reality and virtual reality, additionally utilizing the necessary components such as VR oculus or an AR display camera.

Several difficulties were faced, such as free server space, browser memory not allowing large games to run, visual bugs that required a lot of removal effort, incompatibility of some versions of Unity with the web platform unit, Unreal Engine often fails to cook the web game and problems finding testers.

The development part of the game is very time consuming and has many steps to follow until the final implementation on the web server, several backups had to be made during the course of development.

It has third person shooter games and first person shooter games that interact with users, the games have objectives, missions and there is an instruction manual to provide the general guidelines for the game to be played, as in the example of which keys to use.

This project has several features, users can manage their profile and make changes to the data, and also there is a site activity report and an overview of player performance.

The amount of turnover and the number of hours each player takes to play each game will be determined.

According with [Paul Bertens and Anna Guitart 2017] it is possible to have close predictions based on time frames of when a player is going to quit (churn).

A coefficient will be used for each day, due to the tendency of changes in turnover after a certain moment, this coefficient  $W_i$  is determined by:

$$W_i = \frac{1}{d(t_q, t_i)} \quad (1)$$

$t_q$  : Date of q (query) point  
 $t_i$  : Last date + 1  
 $d(t_q, t_i)$  : Distance between two dates

This formula will be used to determine the churn point (when the user gives up).

But it is based on the differences between the first played date and all the others till the last, among other factors such as average playing time, to predict when a specific user will stop accessing the gaming platform...

For artificial intelligence will be used the formula described below:

X<sub>j</sub> is the current player's life amount and Y<sub>i</sub> is the current enemy player's life amount.

If  $(X_j - Y_i) > 50$  then AI will advance to a more aggressive state, the enemy player will have better health and more speed.

If  $0 < (X_j - Y_i) \leq 50$  then the AI does not change.

If  $(X_j - Y_i) < 0$  then AI loses life and becomes slower.

The goal is that the enemy's performance is consistent with the level of the user, so as to make the game more appealing to the user, if his health is very low and the enemy's health is very high, the enemy will have easier to play characteristics., health and speed are reduced, if the player is at a very high health level compared to the enemy's health, the enemy's health will increase, it assumes that the player is very skilled and can play at a more advance level. The goal is to make the game as appealing as possible to the player, because if it is extremely easy or extremely difficult it may lead to the player giving up.

X<sub>j</sub> is the amount of player's life, being that j represents the current time and Y<sub>i</sub> the amount of life of the enemy where j corresponds to the several enemies that the level have.

With regard to augmented reality the area obtained by the camera will be calculated, in order to obtain the collisions with the objective that the system will place the computer generated enemies properly on the screen.

In the virtual reality part it will take into account the placement of the hands in order to be able to pick up objects like weapons.

The web system aims to implement several features that online platforms do not have to provide statistical data on the use of each user based on the login date, how long they played, what games they played and also offer a profile management module where images can be saved, game reviews inserted, check credits, etc...

For this purpose a combination of PHP and JavaScript will be used to implement these features, in analogy with e-commerce sites will also give the user the ability to create items to which they can associate information.

The aim is to create a credit system, in which each player has certain associated number of credits and can give or receive credits to another player, the more credits the player has, higher is the access to more weapons, or more cars, or more levels.

In order to create a social component, the availability of a chat also facilitates the process of credit exchange, the number of credits is visible on the player profile.

In order to do a better evaluation, during this experience, it will be taken notes of the following information, considered as metrics: time spent playing each game, total time spent in the application, number of mouse clicks in the application (not counting the ones during the game), number of times that returned to a previous page, number of errors, number of time to recover from Errors and number of functions (including games) used.

## L - Evaluation

When all games are online and the statistical method is working it will be made tests to 21 users, without mentioning their names as per the table below (with auto generated data), initially it will be obtained the demographic data like age, sex profession, district and nationality, it aims to have both sexes in equal proportions and also aims to some users that have more technical difficulties in order to know which problems they face and how to make the system more accessible.

User Number	Sex	District	Professional status	Age	Nationality
1	M	Caldas da Rainha	Operador fábrica	56	Portuguesa
2	F	Lisboa	Estudante	23	Portuguesa
3	F	Lisboa	Empregada	39	Portuguesa
4	F	Lisboa	Desempregada	50	Portuguesa
5	M	Lisboa	Estudante	23	Portuguesa
6	F	Lisboa	Bibliotecária	44	Portuguesa
7	M	Évora	Desempregado	35	Portuguesa
8	F	Caldas da Rainha	Empresária	35	Portuguesa
9	F	Lisboa	Médica	45	Portuguesa
10	F	Lisboa	Estudante	18	Portuguesa
11	F	Lisboa	Técnica Saúde	27	Portuguesa
12	M	Lisboa	Estudante	18	Portuguesa
13	M	Évora	Estudante	19	Portuguesa
14	M	Caldas da Rainha	Estudante	22	Portuguesa
15	M	Évora	Repcionista	46	Portuguesa
16	F	Lisboa	Engenheira Mecânica	32	Portuguesa
17	M	Lisboa	Reformado	83	Portuguesa
18	M	Lisboa	Operador Call Center	28	Portuguesa
19	F	Lisboa	Operador Caixa	31	Portuguesa
20	M	Évora	Operador Loja	29	Portuguesa
21	M	Évora	Reformado	68	Portuguesa

Table 1 – Demographic data



Figure 4 – Distribution by sex

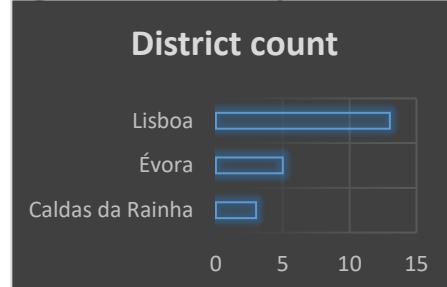


Figure 5 – Distribution by District

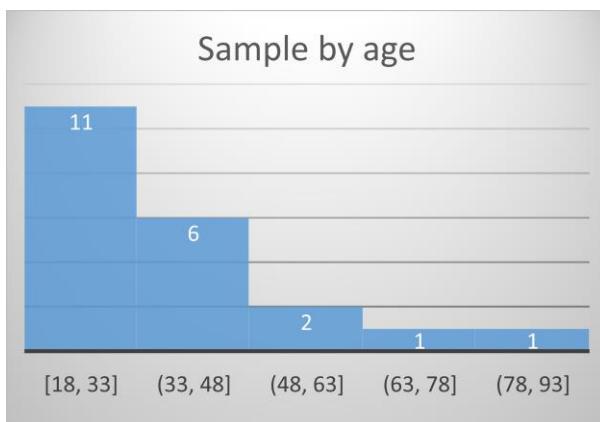


Figure 6 – Distribution by age

Of all the population that could use this application, the tests are carried out in this process of twenty one (21) possible users. The present work is found in three parts. First of all, a small description of the experience, where you can obtain the general information and information about a demonstration and evaluation in itself. In the second part, the users will execute a series of tasks related to the application being studied and on the third part will answer a quick qualitative questionnaire.

These tests aim to answer the following questions:

How much time do users take to perform a task? What kind of errors they do and which is the more common? What functionalities do they use? What types of users have the most difficulties? How many times one user did went back? When they commit fails, how much time and what they do to recover? What are the most used functions?

In terms of equipment used it requires a computer with a browser and internet connection, this web platform has been tested for Internet explorer, Microsoft Edge, Google Chrome and Mozilla Firefox.

The moment users log to the application all their activities will be stored in the database.

The main usability measures are: time spent on the site, time spent playing (efficiency), the number of errors (security) and the final questionnaire with various metrics (satisfaction).

Pilot Testes will be performed before the official tests to adjust the plan to reality, to follow the protocol to check the guide that is provided to users mentioning the details of the test and how they should proceed and also a consentement form in order to comply with the ethics. An initial form is provided with the demographic details and after the tests a form will be given asking about the satisfaction.

These pilot test also allow to improve the method of data analysis in order to check if the metrics are adjusted to real experiences.

Concerning the tests there will be preparations before the user's arrival, with the documents to fill, with text explaining how to use the online gaming platform, with the ethical norms, the consentment form, the questionnaires and a computer with the browser opened in the correspondent link in a close environment eliminating the possibility of interruptions.

At the beginning there will be a formal explanation of what is expected as well as the ethical norms in order to leave the user in a comfortable state, it will be asked to talk during the test, no help can be provided and notes are going to be taken as well as it will be counted the number of clicks and chronometer the time taken using the application, the time taken for each game and the time taken to recover from errors.

In the end the data obtained is stored specially the ones concerning the performance metrics, it is asked to fill the questionnaire after test, opinions will be asked and it will be made a general balance, keeping the correspondent documents to each user identified by a number. Finally, gratitudes are given for the participation and it is written a report stating what happened, which observations were made, the commentaries given, which problems were verified and mention the results obtained. The evaluation report should contain objectives, a description of what is being tested, the environment, the tasks, the characteristics of the participant, the methodology, the metrics and an analysis of the data obtained.

Concerning the data analysis for each metrics it will be given the descriptive statistics as shown in figure 6.

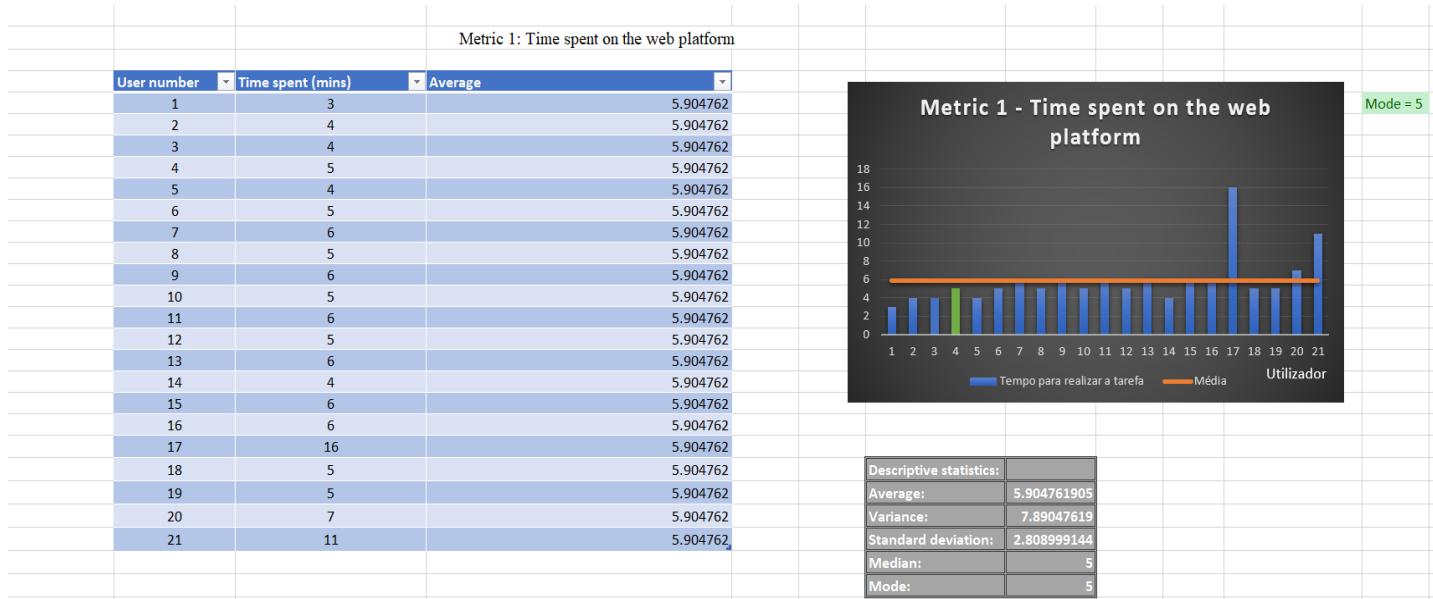


Figure 7 – Analysis of the metric: Time spent on the web platform

On the example above it will be calculated the average, variance, standard deviation, median and mode. Taking in consideration a sample of 21 users it will be used 7 metrics as dependent variables that are:

Time spent playing each game; total time spent in the application; number of mouse clicks in the application (not counting the ones during the game); number of times that returned to a previous page; number of errors; time to recover from errors; number of functions (including games) used.

The questionnaire after test based on the Liker scale will also be analyzed. For each metric will be provided a table as the one in table 2.

Descriptive statistics:	
Average:	5.904761905
Variance:	7.89047619
Standard deviation:	2.808999144
Median:	5
Mode:	5

Table 2 – Descriptive statistics of the metric: Time spent on the web platform.

The average will be compared with the pretended value and the standard deviation will mention how far is the distribution from the average it allows to check if the objectives are obtained. A comparison will be made between the different metrics, and development adjustments will be performed based on the results obtained.

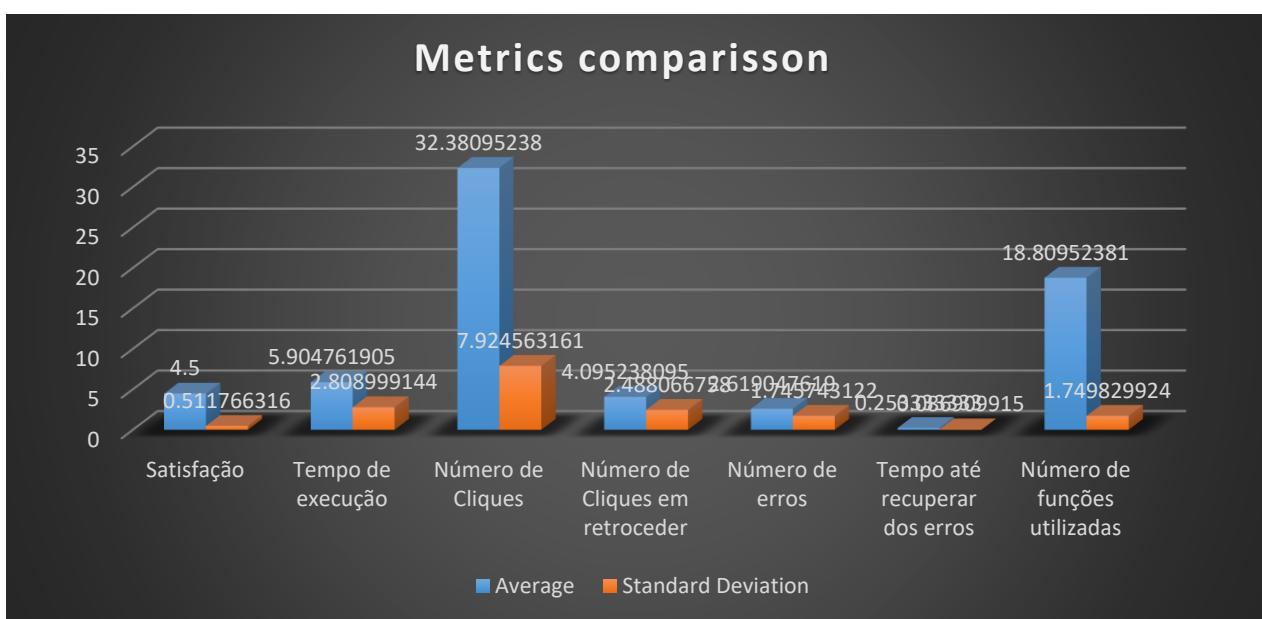


Figure 8 – Average and standard deviation for all metrics

On this analysis it will also be used the confidence intervals and chi square. For metrics like time spent it will be provided a minimum time in this case 20 minutes and the confidence intervals will be calculated with a trust rating of 95% based on the variance and the value of the number of users of the sample in this case 21, which means n=20. We will calculate the average of the standard deviation:

$$s_{em} = \sqrt{S^2 / N} \quad (2)$$

Based on the table for p=0,05 t=3.182

$$X_{\min} = \bar{X} - (t_{p,gl} \times s_{em}) \quad (3)$$

$$X_{\max} = \bar{X} + (t_{p,gl} \times s_{em}) \quad (4)$$

If the confidence interval is above the 20 minutes restriction it considered that this metric was successful.

Concerning games comparison it will be used the Chi-Square, where it will take in consideration the number of times a game was chosen. In this case n= number of games – 1, the formula of the Chi-Square is:

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (5)$$

where:

c = degrees of freedom

O = observed value(s)

E = expected value(s)

Based on the table 3:

Game / Option	Frequency expected (number of users / number of games)	Frequenc y observed	Differenc e	Square of the difference	/ Frequency expected
<b>Game 1</b>	10	5	-5	25	2.5
<b>Game 2</b>	10	15	5	25	2.5
...		9	-1	1	0.1

Table 3 – Calculation of the Qui-Square

For this calculation it will be considered a margin of security of 95%, after checking the value on the table (depends on n) it will give x, if the Chi-Square calculated is < x then we can assume with a 95% of certainty that a certain game is preferred to others.

The application will allow users to check their activity and provide graphics mentioning which games they played and the correspondent amount of time.

# M – Code for the game Crazy Cars

Some of the several classes used.

## Car AI Controller

Class for controlling the car and interface buttons.

```
using System;  
using UnityEngine;
```

```
namespace UnityStandardAssets.Vehicles.Car
```

```
{  
    internal enum CarDriveType  
    {  
        FrontWheelDrive,  
        RearWheelDrive,  
        FourWheelDrive  
    }
```

```
    internal enum SpeedType  
    {  
        MPH,  
        KPH  
    }
```

```
    public class CarController : MonoBehaviour  
    {  
        [SerializeField] private CarDriveType m_CarDriveType = CarDriveType.FourWheelDrive;  
        [SerializeField] private WheelCollider[] m_WheelColliders = new WheelCollider[4];  
        [SerializeField] private GameObject[] m_WheelMeshes = new GameObject[4];  
        [SerializeField] private WheelEffects[] m_WheelEffects = new WheelEffects[4];
```

```

[SerializeField] private Vector3 m_CentreOfMassOffset;
[SerializeField] private float m_MaximumSteerAngle;
[Range(0, 1)] [SerializeField] private float m_SteerHelper; // 0 is raw physics , 1 the car will
grip in the direction it is facing
[Range(0, 1)] [SerializeField] private float m_TractionControl; // 0 is no traction control, 1 is
full interference
[SerializeField] private float m_FullTorqueOverAllWheels;
[SerializeField] private float m_ReverseTorque;
[SerializeField] private float m_MaxHandbrakeTorque;
[SerializeField] private float m_Downforce = 100f;
[SerializeField] private SpeedType m_SpeedType;
[SerializeField] private float m_Topspeed = 200;
[SerializeField] private static int NoOfGears = 5;
[SerializeField] private float m_RevRangeBoundary = 1f;
[SerializeField] private float m_SlipLimit;
[SerializeField] private float m_BrakeTorque;

private Quaternion[] m_WheelMeshLocalRotations;
private Vector3 m_Prevpos, m_Pos;
private float m_SteerAngle;
private int m_GearNum;
private float m_GearFactor;
private float m_OldRotation;
private float m_CurrentTorque;
private Rigidbody m_Rigidbody;
private const float k_ReversingThreshold = 0.01f;

public bool Skidding { get; private set; }
public float BrakeInput { get; private set; }

```

```

public float CurrentSteerAngle{ get { return m_SteerAngle; } }

public float CurrentSpeed{ get { return m_Rigidbody.velocity.magnitude*2.23693629f; } }

public float MaxSpeed{get { return m_Topspeed; }}

public float Revs { get; private set; }

public float AccelInput { get; private set; }

// Use this for initialization

private void Start()

{

    m_WheelMeshLocalRotations = new Quaternion[4];

    for (int i = 0; i < 4; i++)

    {

        m_WheelMeshLocalRotations[i] = m_WheelMeshes[i].transform.localRotation;

    }

    m_WheelColliders[0].attachedRigidbody.centerOfMass = m_CentreOfMassOffset;

    m_MaxHandbrakeTorque = float.MaxValue;

    m_Rigidbody = GetComponent<Rigidbody>();

    m_CurrentTorque = m_FullTorqueOverAllWheels - (m_TractionControl*m_FullTorqueOverAllWheels);

}

private void GearChanging()

{

    float f = Mathf.Abs(CurrentSpeed/MaxSpeed);

    float upgearlimit = (1/(float) NoOfGears)*(m_GearNum + 1);

    float downgearlimit = (1/(float) NoOfGears)*m_GearNum;

```

```

if (m_GearNum > 0 && f < downgearlimit)
{
    m_GearNum--;
}

if (f > upgearlimit && (m_GearNum < (NoOfGears - 1)))
{
    m_GearNum++;
}

// simple function to add a curved bias towards 1 for a value in the 0-1 range
private static float CurveFactor(float factor)
{
    return 1 - (1 - factor)*(1 - factor);
}

// unclamped version of Lerp, to allow value to exceed the from-to range
private static float ULerp(float from, float to, float value)
{
    return (1.0f - value)*from + value*to;
}

private void CalculateGearFactor()

```

```

{
    float f = (1/(float) NoOfGears);

    // gear factor is a normalised representation of the current speed within the current gear's
    range of speeds.

    // We smooth towards the 'target' gear factor, so that revs don't instantly snap up or down
    when changing gear.

    var targetGearFactor = Mathf.InverseLerp(f*m_GearNum, f*(m_GearNum + 1),
    Mathf.Abs(CurrentSpeed/MaxSpeed));

    m_GearFactor = Mathf.Lerp(m_GearFactor, targetGearFactor, Time.deltaTime*5f);

}

```

```

private void CalculateRevs()
{
    // calculate engine revs (for display / sound)
    // (this is done in retrospect - revs are not used in force/power calculations)

    CalculateGearFactor();

    var gearNumFactor = m_GearNum/(float) NoOfGears;

    var revsRangeMin = ULerp(0f, m_RevRangeBoundary, CurveFactor(gearNumFactor));
    var revsRangeMax = ULerp(m_RevRangeBoundary, 1f, gearNumFactor);
    Revs = ULerp(revsRangeMin, revsRangeMax, m_GearFactor);

}

```

```

public void Move(float steering, float accel, float footbrake, float handbrake)
{
    for (int i = 0; i < 4; i++)
    {
        Quaternion quat;

```

```

Vector3 position;

m_WheelColliders[i].GetWorldPose(out position, out quat);

m_WheelMeshes[i].transform.position = position;

m_WheelMeshes[i].transform.rotation = quat;

}

//clamp input values

steering = Mathf.Clamp(steering, -1, 1);

AccelInput = accel = Mathf.Clamp(accel, 0, 1);

BrakeInput = footbrake = -1*Mathf.Clamp(footbrake, -1, 0);

handbrake = Mathf.Clamp(handbrake, 0, 1);

//Set the steer on the front wheels.

//Assuming that wheels 0 and 1 are the front wheels.

m_SteerAngle = steering*m_MaximumSteerAngle;

m_WheelColliders[0].steerAngle = m_SteerAngle;

m_WheelColliders[1].steerAngle = m_SteerAngle;

SteerHelper();

ApplyDrive(accel, footbrake);

CapSpeed();

//Set the handbrake.

//Assuming that wheels 2 and 3 are the rear wheels.

if (handbrake > 0f)

{

    var hbTorque = handbrake*m_MaxHandbrakeTorque;

    m_WheelColliders[2].brakeTorque = hbTorque;
}

```

```

m_WheelColliders[3].brakeTorque = hbTorque;
}

CalculateRevs();
GearChanging();

AddDownForce();
CheckForWheelSpin();
TractionControl();
}

private void CapSpeed()
{
    float speed = m_Rigidbody.velocity.magnitude;
    switch (m_SpeedType)
    {
        case SpeedType MPH:
            speed *= 2.23693629f;
            if (speed > m_Topspeed)
                m_Rigidbody.velocity = (m_Topspeed / 2.23693629f) * m_Rigidbody.velocity.normalized;
            break;

        case SpeedType KPH:
            speed *= 3.6f;
            if (speed > m_Topspeed)

```

```

        m_Rigidbody.velocity = (m_Topspeed/3.6f) * m_Rigidbody.velocity.normalized;
        break;
    }
}

private void ApplyDrive(float accel, float footbrake)
{
    float thrustTorque;
    switch (m_CarDriveType)
    {
        case CarDriveType.FourWheelDrive:
            thrustTorque = accel * (m_CurrentTorque / 4f);
            for (int i = 0; i < 4; i++)
            {
                m_WheelColliders[i].motorTorque = thrustTorque;
            }
            break;

        case CarDriveType.FrontWheelDrive:
            thrustTorque = accel * (m_CurrentTorque / 2f);
            m_WheelColliders[0].motorTorque = m_WheelColliders[1].motorTorque = thrustTorque;
            break;

        case CarDriveType.RearWheelDrive:
            thrustTorque = accel * (m_CurrentTorque / 2f);
    }
}

```

```

    m_WheelColliders[2].motorTorque = m_WheelColliders[3].motorTorque =
thrustTorque;
    break;

}

for (int i = 0; i < 4; i++)
{
    if (CurrentSpeed > 5 && Vector3.Angle(transform.forward, m_Rigidbody.velocity) <
50f)
    {
        m_WheelColliders[i].brakeTorque = m_BrakeTorque*footbrake;
    }
    else if (footbrake > 0)
    {
        m_WheelColliders[i].brakeTorque = 0f;
        m_WheelColliders[i].motorTorque = -m_ReverseTorque*footbrake;
    }
}
}

```

```

private void SteerHelper()
{
    for (int i = 0; i < 4; i++)
    {
        WheelHit wheelhit;
        m_WheelColliders[i].GetGroundHit(out wheelhit);
        if (wheelhit.normal == Vector3.zero)

```

```

        return; // wheels arent on the ground so dont realign the rigidbody velocity
    }

    // this if is needed to avoid gimbal lock problems that will make the car suddenly shift
    direction

    if (Mathf.Abs(m_OldRotation - transform.eulerAngles.y) < 10f)
    {
        var turnadjust = (transform.eulerAngles.y - m_OldRotation) * m_SteerHelper;
        Quaternion velRotation = Quaternion.AngleAxis(turnadjust, Vector3.up);
        m_Rigidbody.velocity = velRotation * m_Rigidbody.velocity;
    }

    m_OldRotation = transform.eulerAngles.y;
}

// this is used to add more grip in relation to speed
private void AddDownForce()
{
    m_WheelColliders[0].attachedRigidbody.AddForce(-transform.up*m_Downforce*
    m_WheelColliders[0].attachedRigidbody.velocity.magnitude);
}

// checks if the wheels are spinning and is so does three things
// 1) emits particles
// 2) plays tire skidding sounds
// 3) leaves skidmarks on the ground
// these effects are controlled through the WheelEffects class

```

```

private void CheckForWheelSpin()
{
    // loop through all wheels
    for (int i = 0; i < 4; i++)
    {
        WheelHit wheelHit;
        m_WheelColliders[i].GetGroundHit(out wheelHit);

        // is the tire slipping above the given threshold
        if ((Mathf.Abs(wheelHit.forwardSlip) >= m_SlipLimit ||
            Mathf.Abs(wheelHit.sidewaysSlip) >= m_SlipLimit))
        {
            m_WheelEffects[i].EmitTyreSmoke();

            // avoiding all four tires screeching at the same time
            // if they do it can lead to some strange audio artefacts
            if (!AnySkidSoundPlaying())
            {
                m_WheelEffects[i].PlayAudio();
            }
            continue;
        }

        // if it wasnt slipping stop all the audio
        if (m_WheelEffects[i].PlayingAudio)
        {
            m_WheelEffects[i].StopAudio();
        }
    }
}

```

```

        m_WheelEffects[i].EndSkidTrail();

    }

}

// crude traction control that reduces the power to wheel if the car is wheel spinning too much
private void TractionControl()
{
    WheelHit wheelHit;
    switch (m_CarDriveType)
    {
        case CarDriveType.FourWheelDrive:
            // loop through all wheels
            for (int i = 0; i < 4; i++)
            {
                m_WheelColliders[i].GetGroundHit(out wheelHit);

                AdjustTorque(wheelHit.forwardSlip);
            }
            break;

        case CarDriveType.RearWheelDrive:
            m_WheelColliders[2].GetGroundHit(out wheelHit);
            AdjustTorque(wheelHit.forwardSlip);

            m_WheelColliders[3].GetGroundHit(out wheelHit);
            AdjustTorque(wheelHit.forwardSlip);
            break;
    }
}

```

```

case CarDriveType.FrontWheelDrive:
    m_WheelColliders[0].GetGroundHit(out wheelHit);
    AdjustTorque(wheelHit.forwardSlip);

    m_WheelColliders[1].GetGroundHit(out wheelHit);
    AdjustTorque(wheelHit.forwardSlip);
    break;
}

}

private void AdjustTorque(float forwardSlip)
{
    if (forwardSlip >= m_SlipLimit && m_CurrentTorque >= 0)
    {
        m_CurrentTorque -= 10 * m_TractionControl;
    }
    else
    {
        m_CurrentTorque += 10 * m_TractionControl;
        if (m_CurrentTorque > m_FullTorqueOverAllWheels)
        {
            m_CurrentTorque = m_FullTorqueOverAllWheels;
        }
    }
}

```

```

private bool AnySkidSoundPlaying()
{
    for (int i = 0; i < 4; i++)
    {
        if (m_WheelEffects[i].PlayingAudio)
        {
            return true;
        }
    }
    return false;
}
}

```

## Car AI part 2

Define AI acceleration, breaking, etc...

```

using System;
using UnityEngine;
using Random = UnityEngine.Random;

namespace UnityStandardAssets.Vehicles.Car
{
    [RequireComponent(typeof(CarController))]
    public class CarAIControl : MonoBehaviour
    {
        public enum BrakeCondition
        {

```

```

    NeverBrake,           // the car simply accelerates at full throttle all the time.

    TargetDirectionDifference, // the car will brake according to the upcoming change in
    direction of the target. Useful for route-based AI, slowing for corners.

    TargetDistance,        // the car will brake as it approaches its target, regardless of the
    target's direction. Useful if you want the car to

        // head for a stationary target and come to rest when it arrives there.

    }

    // This script provides input to the car controller in the same way that the user control script
    does.

    // As such, it is really 'driving' the car, with no special physics or animation tricks to make
    the car behave properly.

    // "wandering" is used to give the cars a more human, less robotic feel. They can waver
    slightly

    // in speed and direction while driving towards their target.

    [SerializeField] [Range(0, 1)] private float m_CautiousSpeedFactor = 0.05f;           //
    percentage of max speed to use when being maximally cautious

    [SerializeField] [Range(0, 180)] private float m_CautiousMaxAngle = 50f;             //
    angle of approaching corner to treat as warranting maximum caution

    [SerializeField] private float m_CautiousMaxDistance = 100f;                      //
    distance
    at which distance-based cautiousness begins

    [SerializeField] private float m_CautiousAngularVelocityFactor = 30f;           //
    how
    cautious the AI should be when considering its own current angular velocity (i.e. easing off
    acceleration if spinning!)

    [SerializeField] private float m_SteerSensitivity = 0.05f;                     //
    how
    sensitively the AI uses steering input to turn to the desired direction

    [SerializeField] private float m_AccelSensitivity = 0.04f;                   //
    How
    sensitively the AI uses the accelerator to reach the current desired speed

    [SerializeField] private float m_BrakeSensitivity = 1f;                      //
    How
    sensitively the AI uses the brake to reach the current desired speed

```

```

[SerializeField] private float m_LateralWanderDistance = 3f; // how far
the car will wander laterally towards its target

[SerializeField] private float m_LateralWanderSpeed = 0.1f; // how fast
the lateral wandering will fluctuate

[SerializeField] [Range(0, 1)] private float m_AccelWanderAmount = 0.1f; // how much the cars acceleration will wander

[SerializeField] private float m_AccelWanderSpeed = 0.1f; // how fast
the cars acceleration wandering will fluctuate

[SerializeField] private BrakeCondition m_BrakeCondition =
BrakeCondition.TargetDistance; // what should the AI consider when accelerating/braking?

[SerializeField] private bool m_Driving; // whether the AI is
currently actively driving or stopped.

[SerializeField] private Transform m_Target; // 'target' the
target object to aim for.

[SerializeField] private bool m_StopWhenTargetReached; // should
we stop driving when we reach the target?

[SerializeField] private float m_ReachTargetThreshold = 2; // proximity
to target to consider we 'reached' it, and stop driving.

private float m_RandomPerlin; // A random value for the car to base its wander on
(so that AI cars don't all wander in the same pattern)

private CarController m_CarController; // Reference to actual car controller we are
controlling

private float m_AvoidOtherCarTime; // time until which to avoid the car we recently
collided with

private float m_AvoidOtherCarSlowdown; // how much to slow down due to colliding
with another car, whilst avoiding

private float m_AvoidPathOffset; // direction (-1 or 1) in which to offset path to avoid
other car, whilst avoiding

private Rigidbody m_Rigidbody;

private void Awake()

```

```

{

    // get the car controller reference
    m_CarController = GetComponent<CarController>();

    // give the random perlin a random value
    m_RandomPerlin = Random.value*100;

    m_Rigidbody = GetComponent<Rigidbody>();

}

private void FixedUpdate()
{
    if (m_Target == null || !m_Driving)
    {
        // Car should not be moving,
        // use handbrake to stop
        m_CarController.Move(0, 0, -1f, 1f);

    }
    else
    {
        Vector3 fwd = transform.forward;
        if (m_Rigidbody.velocity.magnitude > m_CarController.MaxSpeed*0.1f)
        {
            fwd = m_Rigidbody.velocity;
        }

        float desiredSpeed = m_CarController.MaxSpeed;

```

```

// now it's time to decide if we should be slowing down...

switch (m_BrakeCondition)
{
    case BrakeCondition.TargetDirectionDifference:
        {
            // the car will brake according to the upcoming change in direction of the target.
            Useful for route-based AI, slowing for corners.

            // check out the angle of our target compared to the current direction of the car
            float approachingCornerAngle = Vector3.Angle(m_Target.forward, fwd);

            // also consider the current amount we're turning, multiplied up and then
            compared in the same way as an upcoming corner angle
            float spinningAngle =
m_Rigidbody.angularVelocity.magnitude*m_CautiousAngularVelocityFactor;

            // if it's different to our current angle, we need to be cautious (i.e. slow down) a
            certain amount
            float cautiousnessRequired = Mathf.InverseLerp(0, m_CautiousMaxAngle,
                Mathf.Max(spinningAngle,
                    approachingCornerAngle));

            desiredSpeed = Mathf.Lerp(m_CarController.MaxSpeed,
m_CarController.MaxSpeed*m_CautiousSpeedFactor,
                cautiousnessRequired);

            break;
        }

    case BrakeCondition.TargetDistance:
        {

```

```

    // the car will brake as it approaches its target, regardless of the target's
    direction. Useful if you want the car to

        // head for a stationary target and come to rest when it arrives there.

        // check out the distance to target
        Vector3 delta = m_Target.position - transform.position;
        float distanceCautiousFactor = Mathf.InverseLerp(m_CautiousMaxDistance, 0,
delta.magnitude);

        // also consider the current amount we're turning, multiplied up and then
        compared in the same way as an upcoming corner angle
        float spinningAngle =
m_Rigidbody.angularVelocity.magnitude*m_CautiousAngularVelocityFactor;

        // if it's different to our current angle, we need to be cautious (i.e. slow down) a
        certain amount
        float cautiousnessRequired = Mathf.Max(
            Mathf.InverseLerp(0, m_CautiousMaxAngle, spinningAngle),
            distanceCautiousFactor);

        desiredSpeed = Mathf.Lerp(m_CarController.MaxSpeed,
m_CarController.MaxSpeed*m_CautiousSpeedFactor,
            cautiousnessRequired);

        break;
    }

    case BrakeCondition.NeverBrake:
        break;
}

// Evasive action due to collision with other cars:

```

```

// our target position starts off as the 'real' target position
Vector3 offsetTargetPos = m_Target.position;

// if are we currently taking evasive action to prevent being stuck against another car:
if (Time.time < m_AvoidOtherCarTime)
{
    // slow down if necessary (if we were behind the other car when collision occured)
    desiredSpeed *= m_AvoidOtherCarSlowdown;

    // and veer towards the side of our path-to-target that is away from the other car
    offsetTargetPos += m_Target.right*m_AvoidPathOffset;
}

else
{
    // no need for evasive action, we can just wander across the path-to-target in a
    random way,
    // which can help prevent AI from seeming too uniform and robotic in their driving
    offsetTargetPos += m_Target.right*
        (Mathf.PerlinNoise(Time.time*m_LateralWanderSpeed,
    m_RandomPerlin)*2 - 1)*
        m_LateralWanderDistance;
}

// use different sensitivity depending on whether accelerating or braking:
float accelBrakeSensitivity = (desiredSpeed < m_CarController.CurrentSpeed)
    ? m_BrakeSensitivity
    : m_AccelSensitivity;

// decide the actual amount of accel/brake input to achieve desired speed.

```

```

        float accel = Mathf.Clamp((desiredSpeed -
m_CarController.CurrentSpeed)*accelBrakeSensitivity, -1, 1);

        // add acceleration 'wander', which also prevents AI from seeming too uniform and
robotic in their driving
        // i.e. increasing the accel wander amount can introduce jostling and bumps between
AI cars in a race
        accel *= (1 - m_AccelWanderAmount) +
(Mathf.PerlinNoise(Time.time*m_AccelWanderSpeed,
m_RandomPerlin)*m_AccelWanderAmount);

        // calculate the local-relative position of the target, to steer towards
Vector3 localTarget = transform.InverseTransformPoint(offsetTargetPos);

        // work out the local angle towards the target
float targetAngle = Mathf.Atan2(localTarget.x, localTarget.z)*Mathf.Rad2Deg;

        // get the amount of steering needed to aim the car towards the target
float steer = Mathf.Clamp(targetAngle*m_SteerSensitivity, -1,
1)*Mathf.Sign(m_CarController.CurrentSpeed);

        // feed input to the car controller.
m_CarController.Move(steer, accel, accel, 0f);

        // if appropriate, stop driving when we're close enough to the target.
if (m_StopWhenTargetReached && localTarget.magnitude <
m_ReachTargetThreshold)
{
    m_Driving = false;
}

```

```

        }

    }

private void OnCollisionStay(Collision col)
{
    // detect collision against other cars, so that we can take evasive action
    if (col.rigidbody != null)
    {
        var otherAI = col.rigidbody.GetComponent<CarAIControl>();
        if (otherAI != null)
        {
            // we'll take evasive action for 1 second
            m_AvoidOtherCarTime = Time.time + 1;

            // but who's in front?...
            if (Vector3.Angle(transform.forward, otherAI.transform.position -
transform.position) < 90)
            {
                // the other ai is in front, so it is only good manners that we ought to brake...
                m_AvoidOtherCarSlowdown = 0.5f;
            }
            else
            {
                // we're in front! ain't slowing down for anybody...
                m_AvoidOtherCarSlowdown = 1;
            }
        }
    }
}

```

```

// both cars should take evasive action by driving along an offset from the path
centre,
    // away from the other car
    var otherCarLocalDelta =
transform.InverseTransformPoint(otherAI.transform.position);
    float otherCarAngle = Mathf.Atan2(otherCarLocalDelta.x, otherCarLocalDelta.z);
    m_AvoidPathOffset = m_LateralWanderDistance*-Mathf.Sign(otherCarAngle);
}
}
}

```

```

public void SetTarget(Transform target)
{
    m_Target = target;
    m_Driving = true;
}
}

```

### **Car Player Controller**

```

using UnityEngine;
using System.Collections;

```

```

public enum DriveType
{

```

```

        RWD,
        FWD,
        AWD
    };

[System.Serializable]
public class WC
{
    public WheelCollider wheelFL;
    public WheelCollider wheelFR;
    public WheelCollider wheelRL;
    public WheelCollider wheelRR;
}

[System.Serializable]
public class WT
{
    public Transform wheelFL;
    public Transform wheelFR;
    public Transform wheelRL;
    public Transform wheelRR;
}

[RequireComponent(typeof(AudioSource))]/needed audiosource
[RequireComponent(typeof(Rigidbody))]/needed Rigid body
public class CarControlCS : MonoBehaviour {

    public WC wheels;
    public WT tires;
    public WheelCollider[] extraWheels;
    public Transform[] extraWheelObjects;
}

```

```

public DriveType DriveTrain = DriveType.RWD;
public Vector3 centerOfGravity;//car's center of mass offset
public GUITexture gasPedal;
public GUITexture brakePedal;
public GUITexture leftPedal;
public GUITexture rightPedal;
public GUITexture buttonA;
public GUITexture speed;
public float maxTorque = 1000f;//car's acceleration value
public float maxReverseSpeed = 50f;//top speed for the reverse gear
public float handBrakeTorque = 500f;//hand brake value
public float maxSteer = 25f;//max steer angle
public bool mobileInput = false;//do you want this to be a mobile game?
public float[] GearRatio;//determines how many gears the car has, and at what speed the
car shifts to the appropriate gear
private int throttleInput;//read only
private int steerInput;//read only
private bool reversing;//read only
private float currentSpeed;//read only
public float maxSpeed = 150f;//how fast the vehicle can go
private int gear;//current gear
Vector3 localCurrentSpeed;
public float x = Input.acceleration.x;
private Vector3 dir;

// Use this for initialization
void Start () {
    throttleInput = 1;
}

```

```

//find all the GUITextures from the scene and assign them
gasPedal = GameObject.Find("GasPedal").GetComponent<GUITexture>();
brakePedal = GameObject.Find("BrakePedal").GetComponent<GUITexture>();
leftPedal = GameObject.Find("LeftPedal").GetComponent<GUITexture>();
rightPedal = GameObject.Find("RightPedal").GetComponent<GUITexture>();

buttonA = GameObject.Find("ButtonA").GetComponent<GUITexture>();
speed = GameObject.Find("Speed").GetComponent<GUITexture>();

//Alter the center of mass for stability on your car
GetComponent<Rigidbody>().centerOfMass = centerOfGravity;
}

// Update is called once per frame
void FixedUpdate () {

    if (GetComponent<Rigidbody>().centerOfMass != centerOfGravity)
        GetComponent<Rigidbody>().centerOfMass = centerOfGravity;

    AlignWheels ();
    GUIButtonControl();
    DriveMobile ();
    Drive ();
    EngineAudio ();

    currentSpeed = GetComponent<Rigidbody>().velocity.magnitude *
2.23693629f;//convert currentspeed into MPH

    localCurrentSpeed = transform.InverseTransformDirection
(GetComponent<Rigidbody> ().velocity);
}

```

```
//if (currentSpeed > maxSpeed || (localCurrentSpeed.z*2.23693629f) < -  
maxReverseSpeed){
```

```
}
```

```
void AlignWheels()
```

```
{
```

```
//align the wheel objs to their colliders
```

```
Quaternion quat;
```

```
Vector3 pos;
```

```
wheels.wheelFL.GetWorldPose(out pos,out quat);
```

```
tires.wheelFL.position = pos;
```

```
tires.wheelFL.rotation = quat;
```

```
wheels.wheelFR.GetWorldPose(out pos,out quat);
```

```
tires.wheelFR.position = pos;
```

```
tires.wheelFR.rotation = quat;
```

```
wheels.wheelRL.GetWorldPose(out pos,out quat);
```

```
tires.wheelRL.position = pos;
```

```
tires.wheelRL.rotation = quat;
```

```
wheels.wheelRR.GetWorldPose(out pos,out quat);
```

```
tires.wheelRR.position = pos;
```

```
tires.wheelRR.rotation = quat;
```

```
for (int i = 0; i < extraWheels.Length; i++)
```

```
{
```

```

        for (int k = 0; k < extraWheelObjects.Length; k++) {

            Quaternion quater;
            Vector3 vec3;

            extraWheels [i].GetWorldPose (out vec3, out quater);
            extraWheelObjects [k].position = vec3;
            extraWheelObjects [k].rotation = quater;

        }

    }

```

```

void GUIButtonControl()
{
    //simple function that disables/enables GUI buttons when we need and dont need
them.

    if (mobileInput)
    {
        gasPedal.gameObject.SetActive(true);
        leftPedal.gameObject.SetActive(true);
        rightPedal.gameObject.SetActive(true);
        brakePedal.gameObject.SetActive(true);

        buttonA.gameObject.SetActive(true);
    }

    else{

```

```

        gasPedal.gameObject.SetActive(false);
        leftPedal.gameObject.SetActive(false);
        rightPedal.gameObject.SetActive(false);
        brakePedal.gameObject.SetActive(false);

        buttonA.gameObject.SetActive(false);
    }

}

void DriveMobile()
{
    if (!mobileInput)
        return;
    //dont call this function if the mobileinput box is not checked in the editor
    float gasMultiplier = 0f;

    if (!reversing) {
        if (currentSpeed < maxSpeed)
            gasMultiplier = 1f;
        else
            gasMultiplier = 0f;
    } else {
        if (currentSpeed < maxReverseSpeed)
            gasMultiplier = 1f;
        else
            gasMultiplier = 0f;
    }
    foreach (Touch touch in Input.touches) {

```

```

//if the gas button is pressed down, speed up the car.

if (touch.phase == TouchPhase.Stationary &&
gasPedal.HitTest(touch.position))

{

    throttleInput = 1;

}

//when the gas button is released, slow the car down

else if (touch.phase == TouchPhase.Ended &&
gasPedal.HitTest(touch.position))

{

    throttleInput = 0;

}

//now the same thing for the brakes

if (touch.phase == TouchPhase.Stationary &&
brakePedal.HitTest(touch.position))

{

    throttleInput = -1;

}

//stop braking once you put your finger off the brake pedal

else if (touch.phase == TouchPhase.Ended &&
brakePedal.HitTest(touch.position))

{

    throttleInput = 0;

}

//now the left steering column...

if (touch.phase == TouchPhase.Stationary &&
leftPedal.HitTest(touch.position))

{

    //turn the front left wheels according to input direction

```

```

        steerInput = -1;

    }

    //and stop the steering once you take your finger off the turn button

    else if (touch.phase == TouchPhase.Ended &&
leftPedal.HitTest(touch.position))

    {

        steerInput = 0;

    }

    //now the right steering column...

    if (touch.phase == TouchPhase.Stationary &&
rightPedal.HitTest(touch.position))

    {

        //turn the front left wheels according to input direction

        steerInput = 1;

    }

    //and stop the steering once you take your finger off the turn button

    else if (touch.phase == TouchPhase.Ended &&
rightPedal.HitTest(touch.position))

    {

        steerInput = 0;

    }

if (touch.phase == TouchPhase.Stationary && buttonA.HitTest(touch.position))

{

    //turn the front left wheels according to input direction

    dir = Vector3.zero;

```

```
dir.x = -Input.acceleration.x;
```

```
if (dir.x > 0)
```

```
{
```

```
steerInput = -1;
```

```
}
```

```
else if (dir.x < 0)
```

```
{
```

```
steerInput = 1;
```

```
}
```

```
}
```

```
/*
```

```
else
```

```
{
```

```

steerInput = 0;
}

/*
//now that we have our input values made, it's time to feed them to the car!
wheels.wheelFL.steerAngle = maxSteer * steerInput;

        wheels.wheelFR.steerAngle = maxSteer * steerInput;

        //~~~~~
if (DriveTrain == DriveType.RWD)
{
    wheels.wheelRL.motorTorque = maxTorque * throttleInput *
gasMultiplier;
    wheels.wheelRR.motorTorque = maxTorque * throttleInput *
gasMultiplier;

    if (localCurrentSpeed.z < -0.1f && wheels.wheelRL.rpm < 10)
{ //in local space, if the car is travelling in the direction of the -z axis, (or in reverse), reversing
will be true

        reversing = true;
    } else {
        reversing = false;
    }
}

if (DriveTrain == DriveType.FWD)
{
    wheels.wheelFL.motorTorque = maxTorque * throttleInput *
gasMultiplier;
    wheels.wheelFR.motorTorque = maxTorque * throttleInput *
gasMultiplier;
}

```

```

        if (localCurrentSpeed.z < -0.1f && wheels.wheelFL.rpm < 10)
{ //in local space, if the car is travelling in the direction of the -z axis, (or in reverse), reversing
will be true

    reversing = true;

} else {

    reversing = false;

}

if (DriveTrain == DriveType.AWD)

{

    wheels.wheelFL.motorTorque = maxTorque * throttleInput *

gasMultiplier;

    wheels.wheelFR.motorTorque = maxTorque * throttleInput *

gasMultiplier;

    wheels.wheelRL.motorTorque = maxTorque * throttleInput *

gasMultiplier;

    wheels.wheelRR.motorTorque = maxTorque * throttleInput *

gasMultiplier;

}

if (localCurrentSpeed.z < -0.1f && wheels.wheelRL.rpm < 10)
{ //in local space, if the car is travelling in the direction of the -z axis, (or in reverse), reversing
will be true

    reversing = true;

} else {

    reversing = false;

}

}

```

```
}
```

```
void Drive()
{
    if (mobileInput)
        return;
    //dont call this function if mobile input is checked in the editor
    float gasMultiplier = 0f;

    if (!reversing) {
        if (currentSpeed < maxSpeed)
            gasMultiplier = 1f;
        else
            gasMultiplier = 0f;

    } else {
        if (currentSpeed < maxReverseSpeed)
            gasMultiplier = 1f;
        else
            gasMultiplier = 0f;
    }

    //the car will be 4 wheel drive or else it will be slow or feel a little sluggish
    //no matter how much you increase the max torque.
    if (DriveTrain == DriveType.RWD)
    {
        wheels.wheelRR.motorTorque = maxTorque * Input.GetAxis("Vertical")
        * gasMultiplier;
    }
}
```

```
wheels.wheelRL.motorTorque = maxTorque * Input.GetAxis("Vertical")
* gasMultiplier;
```

if (localCurrentSpeed.z < -0.1f && wheels.wheelRL.rpm < 10) { //in local space, if the car is travelling in the direction of the -z axis, (or in reverse), reversing will be true

```
reversing = true;
```

```
} else {
```

```
reversing = false;
```

```
}
```

```
}
```

```
if (DriveTrain == DriveType.FWD)
```

```
{
```

```
wheels.wheelFL.motorTorque = maxTorque * Input.GetAxis("Vertical") *
gasMultiplier;
```

```
wheels.wheelFR.motorTorque = maxTorque * Input.GetAxis("Vertical") *
gasMultiplier;
```

if (localCurrentSpeed.z < -0.1f && wheels.wheelFL.rpm < 10) { //in local space, if the car is travelling in the direction of the -z axis, (or in reverse), reversing will be true

```
reversing = true;
```

```
} else {
```

```
reversing = false;
```

```
}
```

```
}
```

```
if (DriveTrain == DriveType.AWD)
```

```
{
```

```
wheels.wheelFL.motorTorque = maxTorque * Input.GetAxis("Vertical") *
gasMultiplier;
```

```
wheels.wheelFR.motorTorque = maxTorque * Input.GetAxis("Vertical") *
gasMultiplier;
```

```

wheels.wheelRL.motorTorque = maxTorque * Input.GetAxis("Vertical")
* gasMultiplier;

wheels.wheelRR.motorTorque = maxTorque * Input.GetAxis("Vertical")
* gasMultiplier;

if (localCurrentSpeed.z < -0.1f && wheels.wheelRL.rpm < 10) {//in local
space, if the car is travelling in the direction of the -z axis, (or in reverse), reversing will be true

    reversing = true;

} else {

    reversing = false;

}

}

```

```

wheels.wheelFL.steerAngle = maxSteer * Input.GetAxis("Horizontal");
wheels.wheelFR.steerAngle = maxSteer * Input.GetAxis("Horizontal");
if (Input.GetButton("Jump"))//pressing space triggers the car's handbrake

{
    wheels.wheelFL.brakeTorque = handBrakeTorque;
    wheels.wheelFR.brakeTorque = handBrakeTorque;
    wheels.wheelRL.brakeTorque = handBrakeTorque;
    wheels.wheelRR.brakeTorque = handBrakeTorque;

}

else//letting go of space disables the handbrake

{
    wheels.wheelFL.brakeTorque = 0f;
    wheels.wheelFR.brakeTorque = 0f;
    wheels.wheelRL.brakeTorque = 0f;
    wheels.wheelRR.brakeTorque = 0f;
}

```

```

    }

}

void EngineAudio()
{
    //the function called to give the car basic audio, as well as some gear shifting
effects
    //it's prefered you use the engine sound included, but you can use your own if you
have one.

    //~~~~~[IMPORTANT]~~~~~

    //make sure your last gear value is higher than the max speed variable or else you
will
    //get unwanted errors!!

    //anyway, let's get started

    for (int i = 0; i < GearRatio.Length; i++) {
        if (GearRatio [i] > currentSpeed) {
            //break this value
            break;
        }

        float minGearValue = 0f;
        float maxGearValue = 0f;
        if (i == 0) {
            minGearValue = 0f;
        } else {
            minGearValue = GearRatio [i];
        }
    }
}

```

```

maxGearValue = GearRatio [i+1];

float pitch = ((currentSpeed - minGearValue) / (maxGearValue -
minGearValue)+0.3f * (gear+1));

GetComponent< AudioSource > ().pitch = pitch;

gear = i;

}

}

void OnGUI()
{
    //show the GUI for the speed and gear we are on.

    GUI.Box(new Rect(10,10,70,30),"KMH: " +
Mathf.Round(GetComponent< Rigidbody >().velocity.magnitude * 2.23693629f));

    if (!reversing)

        GUI.Box(new Rect(10,70,70,30),"Gear: " + (gear+1));

    if (reversing)//if the car is going backwards display the gear as R

        GUI.Box(new Rect(10,70,70,30),"Gear: R");

}

```

## AI way points

```
using System;
using System.Collections;
using UnityEngine;
#if UNITY_EDITOR
using UnityEditor;

#endif

namespace UnityStandardAssets.Utility
{
    public class WaypointCircuit : MonoBehaviour
    {
        public WaypointList waypointList = new WaypointList();
        [SerializeField] private bool smoothRoute = true;
        private int numPoints;
        private Vector3[] points;
        private float[] distances;

        public float editorVisualisationSubsteps = 100;
        public float Length { get; private set; }

        public Transform[] Waypoints
        {
            get { return waypointList.items; }
        }
    }
}
```

```

//this being here will save GC allocs

private int p0n;
private int p1n;
private int p2n;
private int p3n;

private float i;
private Vector3 P0;
private Vector3 P1;
private Vector3 P2;
private Vector3 P3;

// Use this for initialization

private void Awake()
{
    if (Waypoints.Length > 1)
    {
        CachePositionsAndDistances();
    }

    numPoints = Waypoints.Length;
}

public RoutePoint GetRoutePoint(float dist)
{
    // position and direction
    Vector3 p1 = GetRoutePosition(dist);
    Vector3 p2 = GetRoutePosition(dist + 0.1f);
}

```

```

Vector3 delta = p2 - p1;
return new RoutePoint(p1, delta.normalized);
}

public Vector3 GetRoutePosition(float dist)
{
    int point = 0;

    if (Length == 0)
    {
        Length = distances[distances.Length - 1];
    }

    dist = Mathf.Repeat(dist, Length);

    while (distances[point] < dist)
    {
        ++point;
    }

    // get nearest two points, ensuring points wrap-around start & end of circuit
    p1n = ((point - 1) + numPoints)%numPoints;
    p2n = point;

    // found point numbers, now find interpolation value between the two middle points
}

```

```

i = Mathf.InverseLerp(distances[p1n], distances[p2n], dist);

if (smoothRoute)
{
    // smooth catmull-rom calculation between the two relevant points

    // get indices for the surrounding 2 points, because
    // four points are required by the catmull-rom function
    p0n = ((point - 2) + numPoints)%numPoints;
    p3n = (point + 1)%numPoints;

    // 2nd point may have been the 'last' point - a dupe of the first,
    // (to give a value of max track distance instead of zero)
    // but now it must be wrapped back to zero if that was the case.
    p2n = p2n%numPoints;

    P0 = points[p0n];
    P1 = points[p1n];
    P2 = points[p2n];
    P3 = points[p3n];

    return CatmullRom(P0, P1, P2, P3, i);
}

else
{
    // simple linear lerp between the two points:
}

```

```

    p1n = ((point - 1) + numPoints)%numPoints;
    p2n = point;

    return Vector3.Lerp(points[p1n], points[p2n], i);
}

private Vector3 CatmullRom(Vector3 p0, Vector3 p1, Vector3 p2, Vector3 p3, float i)
{
    // comments are no use here... it's the catmull-rom equation.
    // Un-magic this, lord vector!
    return 0.5f*
        ((2*p1) + (-p0 + p2)*i + (2*p0 - 5*p1 + 4*p2 - p3)*i*i +
        (-p0 + 3*p1 - 3*p2 + p3)*i*i*i);
}

private void CachePositionsAndDistances()
{
    // transfer the position of each point and distances between points to arrays for
    // speed of lookup at runtime
    points = new Vector3[Waypoints.Length + 1];
    distances = new float[Waypoints.Length + 1];

    float accumulateDistance = 0;
    for (int i = 0; i < points.Length; ++i)
    {

```

```

var t1 = Waypoints[(i)%Waypoints.Length];
var t2 = Waypoints[(i + 1)%Waypoints.Length];
if (t1 != null && t2 != null)
{
    Vector3 p1 = t1.position;
    Vector3 p2 = t2.position;
    points[i] = Waypoints[i%Waypoints.Length].position;
    distances[i] = accumulateDistance;
    accumulateDistance += (p1 - p2).magnitude;
}
}

```

```
private void OnDrawGizmos()
```

```
{
    DrawGizmos(false);
}
```

```
private void OnDrawGizmosSelected()
```

```
{
    DrawGizmos(true);
}
```

```
private void DrawGizmos(bool selected)
```

```
{
```

```

waypointList.circuit = this;
if (Waypoints.Length > 1)
{
    numPoints = Waypoints.Length;

    CachePositionsAndDistances();
    Length = distances[distances.Length - 1];

    Gizmos.color = selected ? Color.yellow : new Color(1, 1, 0, 0.5f);

    Vector3 prev = Waypoints[0].position;
    if (smoothRoute)
    {
        for (float dist = 0; dist < Length; dist += Length/editorVisualisationSubsteps)
        {
            Vector3 next = GetRoutePosition(dist + 1);
            Gizmos.DrawLine(prev, next);
            prev = next;
        }
        Gizmos.DrawLine(prev, Waypoints[0].position);
    }
    else
    {
        for (int n = 0; n < Waypoints.Length; ++n)
        {
            Vector3 next = Waypoints[(n + 1)%Waypoints.Length].position;
            Gizmos.DrawLine(prev, next);
            prev = next;
        }
    }
}

```

```
        }

    }

}

[Serializable]
public class WaypointList
{
    public WaypointCircuit circuit;
    public Transform[] items = new Transform[0];
}

public struct RoutePoint
{
    public Vector3 position;
    public Vector3 direction;

    public RoutePoint(Vector3 position, Vector3 direction)
    {
        this.position = position;
        this.direction = direction;
    }
}

}

namespace UnityStandardAssets.Utility.Inspector
```

```

{
#if UNITY_EDITOR

[CustomPropertyDrawer(typeof(WaypointCircuit.WaypointList))]

public class WaypointListDrawer : PropertyDrawer
{

    private float lineHeight = 18;
    private float spacing = 4;

    public override void OnGUI(Rect position, SerializedProperty property, GUIContent label)
    {
        EditorGUI.BeginProperty(position, label, property);

        float x = position.x;
        float y = position.y;
        float inspectorWidth = position.width;

        // Draw label

        // Don't make child fields be indented
        var indent = EditorGUI.indentLevel;
        EditorGUI.indentLevel = 0;

        var items = property.FindPropertyRelative("items");
        var titles = new string[] {"Transform", "", "", ""};
        var props = new string[] {"transform", "^", "v", "-"};
        var widths = new float[] {.7f, .1f, .1f, .1f};
    }
}

```

```

float lineHeight = 18;
bool changedLength = false;
if (items.arraySize > 0)
{
    for (int i = -1; i < items.arraySize; ++i)
    {
        var item = items.GetArrayElementAtIndex(i);

        float rowX = x;
        for (int n = 0; n < props.Length; ++n)
        {
            float w = widths[n]*inspectorWidth;

            // Calculate rects
            Rect rect = new Rect(rowX, y, w, lineHeight);
            rowX += w;

            if (i == -1)
            {
                EditorGUI.LabelField(rect, titles[n]);
            }
            else
            {
                if (n == 0)
                {
                    EditorGUI.ObjectField(rect, item.objectReferenceValue, typeof (Transform),
true);
                }
                else

```

```

{
    if (GUI.Button(rect, props[n]))
    {
        switch (props[n])
        {
            case "-":
                items.DeleteArrayElementAtIndex(i);
                items.DeleteArrayElementAtIndex(i);
                changedLength = true;
                break;
            case "v":
                if (i > 0)
                {
                    items.MoveArrayElement(i, i + 1);
                }
                break;
            case "^":
                if (i < items.arraySize - 1)
                {
                    items.MoveArrayElement(i, i - 1);
                }
                break;
        }
    }
}

```

```

y += lineHeight + spacing;

if (changedLength)
{
    break;
}

}

else
{
    // add button

    var addButtonRect = new Rect((x + position.width) - widths[widths.Length - 1]*inspectorWidth, y,
        widths[widths.Length - 1]*inspectorWidth, lineHeight);

    if (GUI.Button(addButtonRect, "+"))

    {
        items.InsertArrayElementAtIndex(items.arraySize);
    }

    y += lineHeight + spacing;
}

// add all button

var addAllButtonRect = new Rect(x, y, inspectorWidth, lineHeight);

if (GUI.Button(addAllButtonRect, "Assign using all child objects"))
{
    var circuit = property.FindPropertyRelative("circuit").objectReferenceValue as
WaypointCircuit;

    var children = new Transform[circuit.transform.childCount];

    int n = 0;

```

```

foreach (Transform child in circuit.transform)
{
    children[n++] = child;
}

Array.Sort(children, new TransformNameComparer());
circuit.waypointList.items = new Transform[children.Length];
for (n = 0; n < children.Length; ++n)
{
    circuit.waypointList.items[n] = children[n];
}
y += lineHeight + spacing;

// rename all button
var renameButtonRect = new Rect(x, y, inspectorWidth, lineHeight);
if (GUI.Button(renameButtonRect, "Auto Rename numerically from this order"))
{
    var circuit = property.FindPropertyRelative("circuit").objectReferenceValue as
WaypointCircuit;
    int n = 0;
    foreach (Transform child in circuit.waypointList.items)
    {
        child.name = "Waypoint " + (n++).ToString("000");
    }
}
y += lineHeight + spacing;

// Set indent back to what it was
EditorGUI.indentLevel = indent;

```

```

        EditorGUI.EndProperty();

    }

public override float GetPropertyHeight(SerializedProperty property, GUIContent label)
{
    SerializedProperty items = property.FindPropertyRelative("items");
    float lineAndSpace = lineHeight + spacing;
    return 40 + (items.arraySize * lineAndSpace) + lineAndSpace;
}

// comparer for check distances in ray cast hits

public class TransformNameComparer : IComparer
{
    public int Compare(object x, object y)
    {
        return ((Transform) x).name.CompareTo(((Transform) y).name);
    }
}

#endif
}

```

### Menu buttons and change camera

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

```

```
using UnityEngine.SceneManagement;

public class Button1 : MonoBehaviour
{

    public Camera cam1;
    public Camera cam2;
    public Camera cam3;
    int cam = 0;

    // Start is called before the first frame update
    public void button1()
    {
        SceneManager.LoadScene("CCCity4");
    }

    public void button2()
    {
        SceneManager.LoadScene("CCMenu2");
    }

    public void button3()
    {

        if (cam1.enabled == true)
        {
            if (cam == 0)
            {
                cam1.enabled = false;
                cam2.enabled = true;
                cam3.enabled = false;
                cam = 1;
            }
        }
    }

    else if (cam2.enabled == true)
    {
        if (cam == 1)
        {
            cam1.enabled = false;
            cam2.enabled = false;
            cam3.enabled = true;
            cam = 2;
        }
    }
}
```

```

        }
    }

else if (cam3.enabled == true)
{
    if (cam == 2)
    {
        cam1.enabled = true;
        cam2.enabled = false;
        cam3.enabled = false;
        cam = 0;
    }
}
}

public void button4()
{
    SceneManager.LoadScene("CCbyNight");
}

```

```

public void button5()
{
    SceneManager.LoadScene("CCDesert");
}

```

```

public void button6()
{
    SceneManager.LoadScene("CCbyNightCar2");
}

```

```

public void button7()
{
    SceneManager.LoadScene("CCbyNightCar3");
}

```

```

public void button8()
{
    SceneManager.LoadScene("CCdesertCar2");
}

```

```

public void button9()
{
    SceneManager.LoadScene("CCDesertCar3");
}

```

```
 }  
 }
```

## Java Script – Car AI

```
var rearWheel1 : WheelCollider;  
var rearWheel2 : WheelCollider;  
var frontWheel1 : WheelCollider;  
var frontWheel2 : WheelCollider;  
  
var wheelFL : Transform;  
var wheelFR : Transform;  
//var wheelML : Transform;  
//var wheelMR : Transform;  
var wheelRL : Transform;  
var wheelRR : Transform;  
  
var steer_max = 20;  
var motor_max = 40;  
var brake_max = 100;  
var steerSpeed = 20;  
  
//private var steer = 0;  
private var forward = 0.2;  
private var back = 0;  
private var brakeRelease = false;  
private var motor = 0.2;  
private var brake = 0;  
private var reverse = false;  
private var speed = 0;  
  
var CollRevSec : float = 1;  
var CollFwdSec : float = 1.1;  
var CollRev1Steer : float = 0.28;  
var CollRev2Steer : float = -0.21;  
var CollFwdSteer : float = 0.15;  
var ObsSteer : float = 0.28;  
var colltime : float = 2;
```

```

var avoidcolltime : float = 2;
var steerouttime : float = 2;

var throttle : float = 1;

var waypoints : Transform[];
var waypointRadius : float = 3;
var damping : float = 0.1;
var loop : boolean = false;
//var speed : float = 2.0;
var faceHeading : boolean = true;

private var targetHeading : Vector3;
private var currentHeading : Vector3;
private var targetwaypoint : int;
//var xform : Transform;
private var useRigidbody : boolean;
private var rigidmember : Rigidbody;

private var steerMaxAngle = 70.0;

// var collisionObject : GameObject;
var avoidcollision : boolean = false;
var hitinfo : RaycastHit;
var steerout : boolean = false;

var angleDisplay : GUIText;
//var obstacDisplay : GUIText;
//var obsvehDisplay : GUIText;
//var collDisplay : GUIText;
//var kphDisplay : GUIText;
var steerxDisplay : GUIText;
var steeraDisplay : GUIText;
//var distxDisplay : GUIText;
//var throttleDisplay : GUIText;
//var dragDisplay : GUIText;

function Start() {

```

```

rigidbody.centerOfMass = Vector3(0, -0.05, 0);

brakeRelease = true;
starttime = Time.time;
}

function Update () {

//seek code

var localTarget = transform.InverseTransformPoint(waypoints[targetwaypoint].position);
var targetAngle = Mathf.Atan2(localTarget.x, localTarget.z) * Mathf.Rad2Deg;

var tarX = Mathf.Round(targetAngle);
var tarxabs = Mathf.Abs(tarX);
angleDisplay.text = "angle: " + tarX;

var steerx=targetAngle/180;

if(throttle == 1 !avoidcollision){
// steer = Mathf.Clamp(steerx, -steerMaxAngle/180, steerMaxAngle/180);
steer = steerx*2;
steer = Mathf.Clamp(steer, -steerMaxAngle/180, steerMaxAngle/180);
}
var steera = steer*180;
steerxDisplay.text = "steer: " + steer;
steeraDisplay.text = "angle: " + steera;

var targetDist = (waypoints[targetwaypoint].position - transform.position).magnitude;
var distx = Mathf.Round(targetDist);

//distxDisplay.text = "distance: " + distx;

//throttleDisplay.text = "throttle: " + throttle;
//dragDisplay.text = "drag: " + rigidbody.drag;
//bodyrend.renderer.material = mainMaterial;

//after reverse if towards target point
// if(avoidcollision){

}

```

```

// steer = -0.3;
// if((Time.time - avoidcoltime)%60f > CollFwdSec){
// avoidcollision = false;
// }
// }

//steering control
//rigidbody.drag = 0.15;
rigidbody.drag = 0.2;

//slow down if close
if(distx < 10) {
    rigidbody.drag += 0.5;
}
//slow down (based on angle) if angle big

if(taxabs > 20) {
    rigidbody.drag += taxabs*0.04;
}

if(Vector3.Distance(transform.position,waypoints[targetwaypoint].position)<=waypointRadius)
{
    targetwaypoint++;
    print(targetwaypoint);

    if(targetwaypoint>=waypoints.Length)
    {
        targetwaypoint = 0;
        // if(!loop)
        // enabled = false;
    }
}
targetHeading = waypoints[targetwaypoint].position - transform.position;

currentHeading = Vector3.Lerp(currentHeading,targetHeading,damping*Time.deltaTime);

motor = forward;
brake = back;
speed = rigidbody.velocity.sqrMagnitude;

//steer = Input.GetAxis("Horizontal"); //pc
//steer = -Input.acceleration.y; //mobile
//forward = Mathf.Clamp(Input.GetAxis("Vertical"), 0, 1);

```

```

//back = -1 * Mathf.Clamp(Input.GetAxis("Vertical"), -1, 0);
//var throttle = Input.acceleration.x;

forward = Mathf.Clamp(throttle, 0, 1);
back = -1 * Mathf.Clamp(throttle, -1, 0);

//if(speed == 0 forward == 0 back == 0) {
//brakeRelease = true;
//}
//if(speed == 0 brakeRelease) {
//if(back > 0) { reverse = true; }
//if(forward > 0) { reverse = false; }

if(throttle > 0) { reverse = false; }
if(throttle < 0) { reverse = true; }
//}

//if(reverse) {
//motor = -1 * back;
//brake = forward;
//} else {
//motor = forward;
//brake = back;
//}

//reverse
if(reverse){
rigidbody.drag = 0;
motor = -1 * back;
throttle = -1;
brake = forward;

//if(steer>0){
steer = CollRev1Steer;
//}else{
//steer = -CollRev1Steer;
//}

//print(steer);
if((Time.time - colltime)%60f > 2){
throttle = 1;
}

```

```

reverse = false;
avoidcollision = true;
avoidcoltime = Time.time;
}
else {
motor = forward;
brake = back;
}
}
//after reverse
if(avoidcollision){
steer = CollFwdSteer;
if((Time.time - avoidcoltime)%60f > 2){
avoidcollision = false;
reverse = false;
motor = forward;
brake = back;
}}
//if (brake > 0 ) { brakeRelease = false; };

rearWheel1.motorTorque = motor_max * motor;
rearWheel2.motorTorque = motor_max * motor;
rearWheel1.brakeTorque = brake_max * brake;
rearWheel2.brakeTorque = brake_max * brake;

//if ( steer == 0 frontWheel1.steerAngle != 0) {
//if (Mathf.Abs(frontWheel1.steerAngle) <= (steerSpeed * Time.deltaTime)) {
//frontWheel1.steerAngle = 0;
//} else if (frontWheel1.steerAngle > 0) {
//frontWheel1.steerAngle = frontWheel1.steerAngle - (steerSpeed * Time.deltaTime);
//} else {
//frontWheel1.steerAngle = frontWheel1.steerAngle + (steerSpeed * Time.deltaTime);
//}
//}
//} else {
//frontWheel1.steerAngle = frontWheel1.steerAngle + (steer * steerSpeed * Time.deltaTime);
//if (frontWheel1.steerAngle > steer_max) { frontWheel1.steerAngle = steer_max; }
//if (frontWheel1.steerAngle < -1 * steer_max) { frontWheel1.steerAngle = -1 * steer_max; }
//}

frontWheel1.steerAngle = steer;

frontWheel2.steerAngle = frontWheel1.steerAngle;
wheelFL.localEulerAngles.y = frontWheel1.steerAngle;

```

```

wheelFR.localEulerAngles.y = frontWheel2.steerAngle;

wheelFR.Rotate(frontWheel1.rpm * 6 * Time.deltaTime, 0, 0);
wheelFL.Rotate(frontWheel2.rpm * 6 * Time.deltaTime, 0, 0);
//wheelMR.Rotate(0, 0, rearWheel1.rpm * -6 * Time.deltaTime);
//wheelML.Rotate(0, 0, rearWheel2.rpm * -6 * Time.deltaTime);
wheelRR.Rotate(rearWheel1.rpm * 6 * Time.deltaTime, 0, 0);
wheelRL.Rotate(rearWheel2.rpm * 6 * Time.deltaTime, 0, 0);

}

function OnCollisionEnter (collision : Collision)
{
//collision detect

//if(collision.gameObject.tag == "Roadblock"){
//collDisplay.text = "collision: " + collision.gameObject.tag;
//print(collision.gameObject.tag);
//reverse
//back =-throttle;
throttle = -1;
reverse = true;
coltime = Time.time;
steer = CollRev1Steer;
//}

//if(collision.gameObject.tag == "Vehicle"){
//collDisplay.text = "collision: " + collision.gameObject.tag;
print(collision.gameObject.tag);
//reverse
// throttle = -1;
// coltime = Time.time;
//steer = CollRev2Steer;
//}

// print("collision");
// throttle = -1;
// coltime = Time.time;
//steer = CollRev1Steer;
}

```

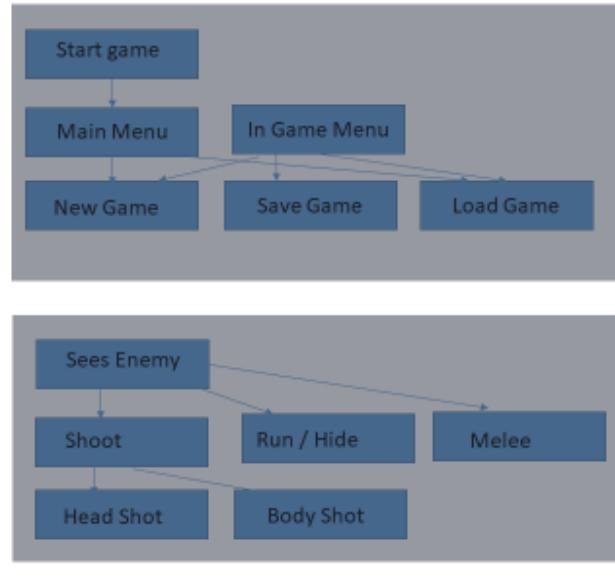
## N – Technical description of the game Huma

Decision Tree



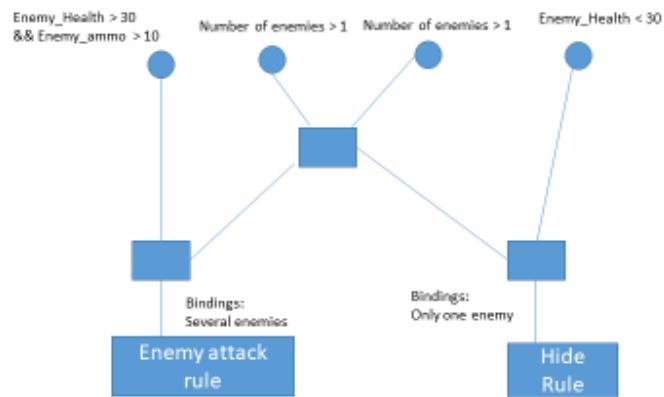
Proximity	Ammo	Life	State
Close	Low	Low	Hide
Far	High	High	Shoot
Close	Low	High	Melee
Close	High	Low	Hide
Far	Low	Low	Shoot
Far	Low	High	Shoot
Far	High	Low	Shoot
Far	Low	Low	Hide
Close	High	High	Melee

# Behavior Trees

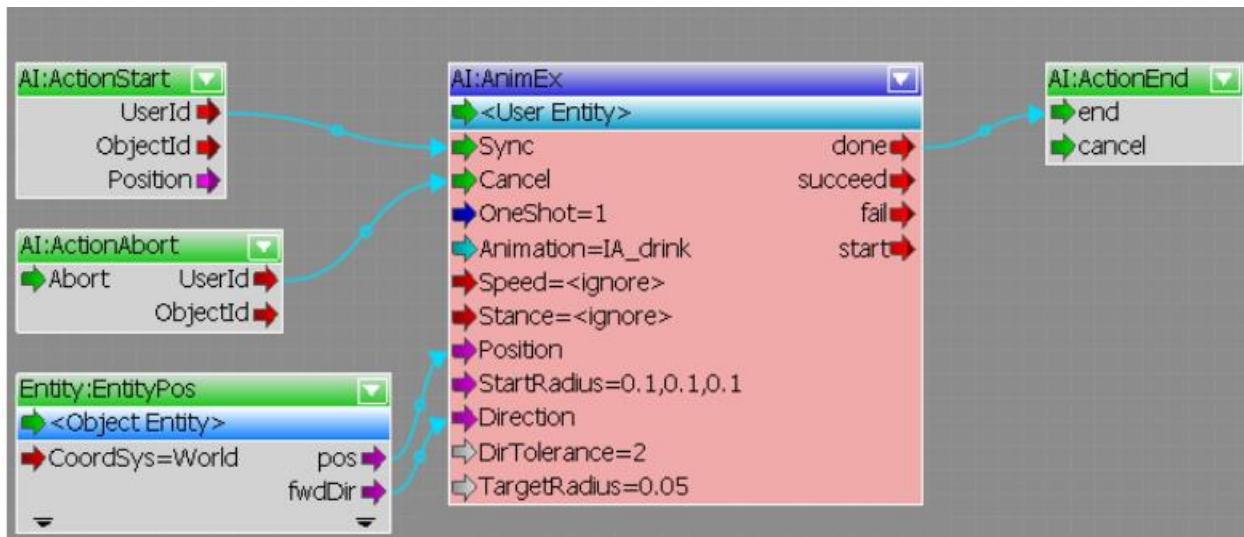


## RETE and System Rules

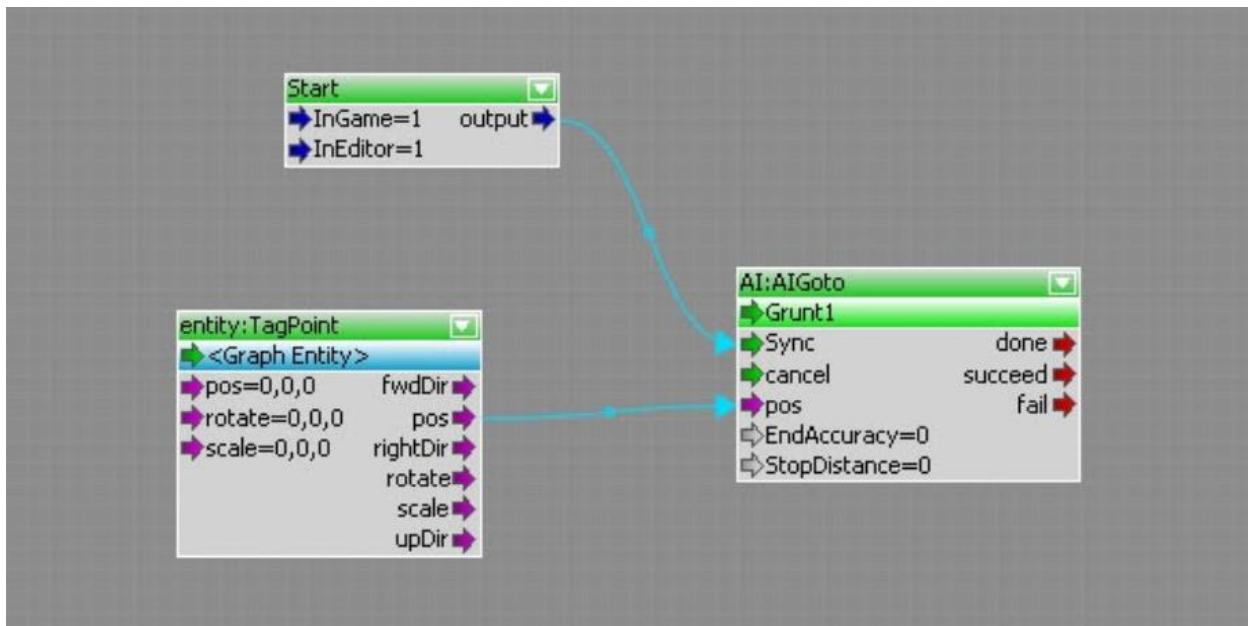
- Enemy Attack Rule: IF (player (health > 25) && (Distance to player > 14) && (ammo = null)) THEN shoot() || grenade();
- Hide Rule: IF (player (health < 25) && (ammo < 10) THEN hide();



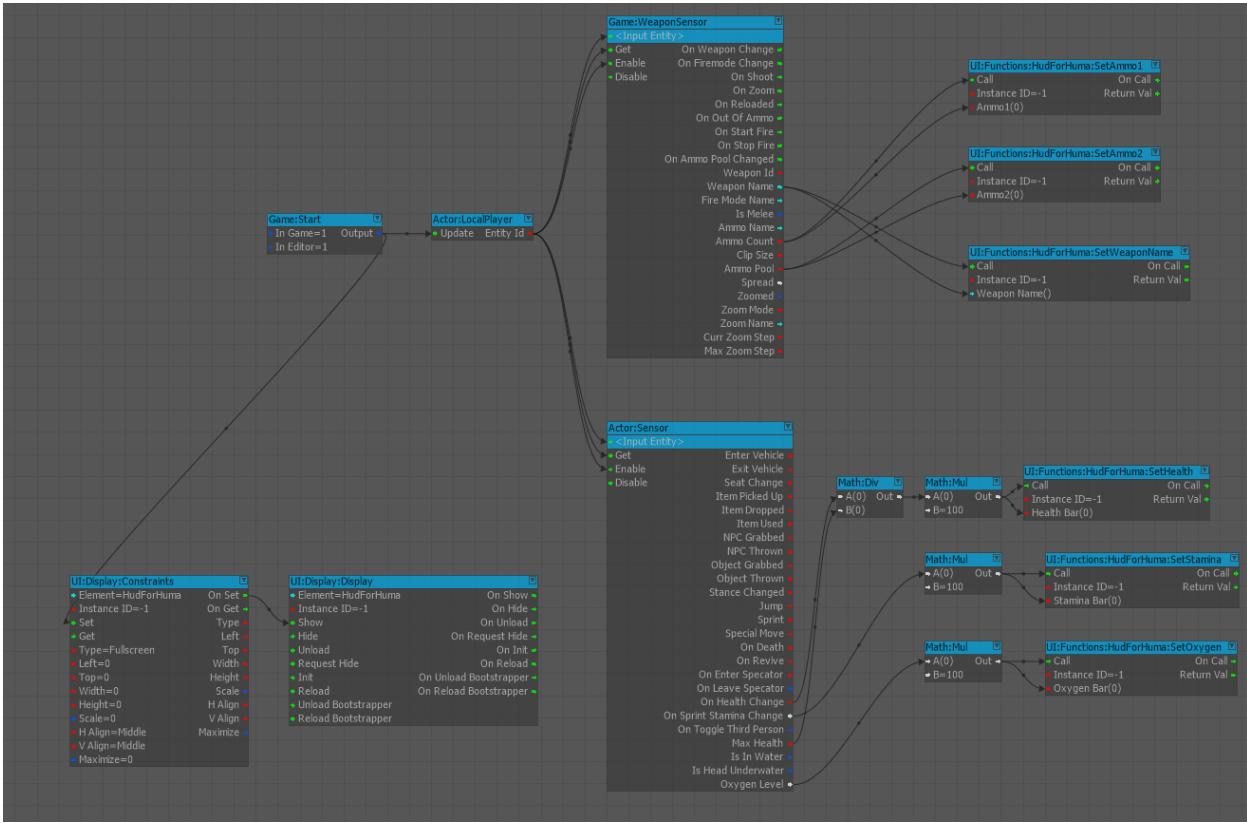
Add animation to AI via Flow Graph:



Move AI character to a point in space via Flow Graph:



## HUD via Flow Graph:



## API for the HUD:

```
<UIElements name="HudForHuma">

    <UIElement name="HudForHuma" render_lockless="0" is_Hud="1">
        <GFx file="HudForHuma.gfx" layer="0">
            <Constraints>
                <Align mode="fullscreen" valign="center" halign="center" scale="0" max="0" />
            </Constraints>
        </GFx>
        <functions>
            <function name="SetHealth" funcname="SetHealth">
                <param name="HealthBar" desc="" type="int" />
            </function>
        </functions>
    </UIElement>
</UIElements>
```

```

<function name="SetStamina" funcname="SetStamina">
    <param name="StaminaBar" desc="" type="int" />
</function>

<function name="SetOxygen" funcname="SetOxygen">
    <param name="OxygenBar" desc="" type="int" />
</function>

<function name="SetAmmo1" funcname="SetAmmo1">
    <param name="Ammo1" desc="" type="int" />
</function>

<function name="SetAmmo2" funcname="SetAmmo2">
    <param name="Ammo2" desc="" type="int" />
</function>

<function name="SetWeaponName" funcname="SetWeaponName">
    <param name="WeaponName" desc="" type="String" />
</function>

</functions>

</UIElement>

</UIElements>

```

Flash: Action Script code for the HUD:

```

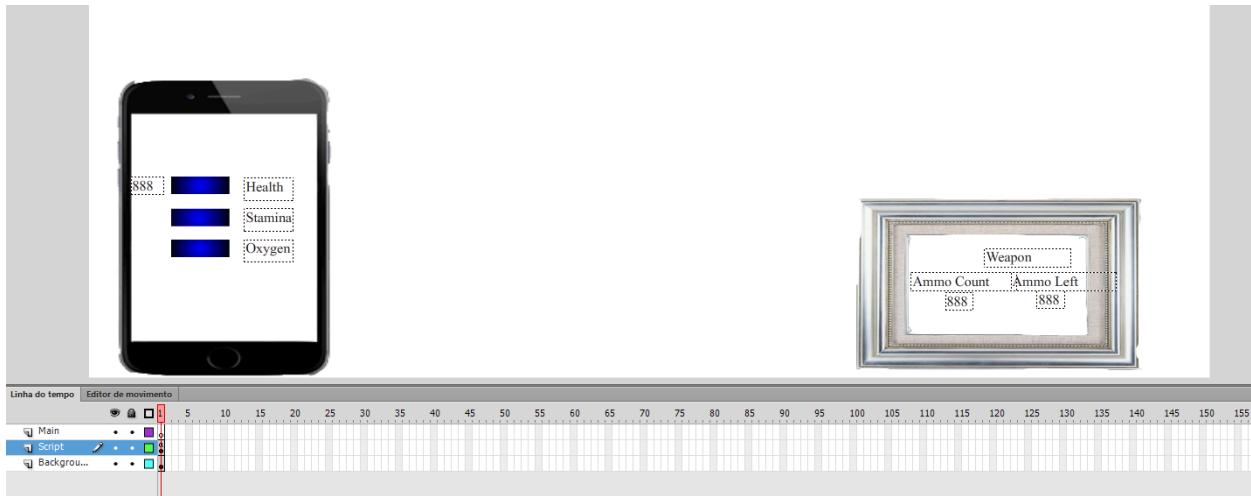
function SetHealth (Health)
{
    HealthBar._width = (Health)
    _root.HealthN.text = (Health)
}

function SetStamina (Stamina)

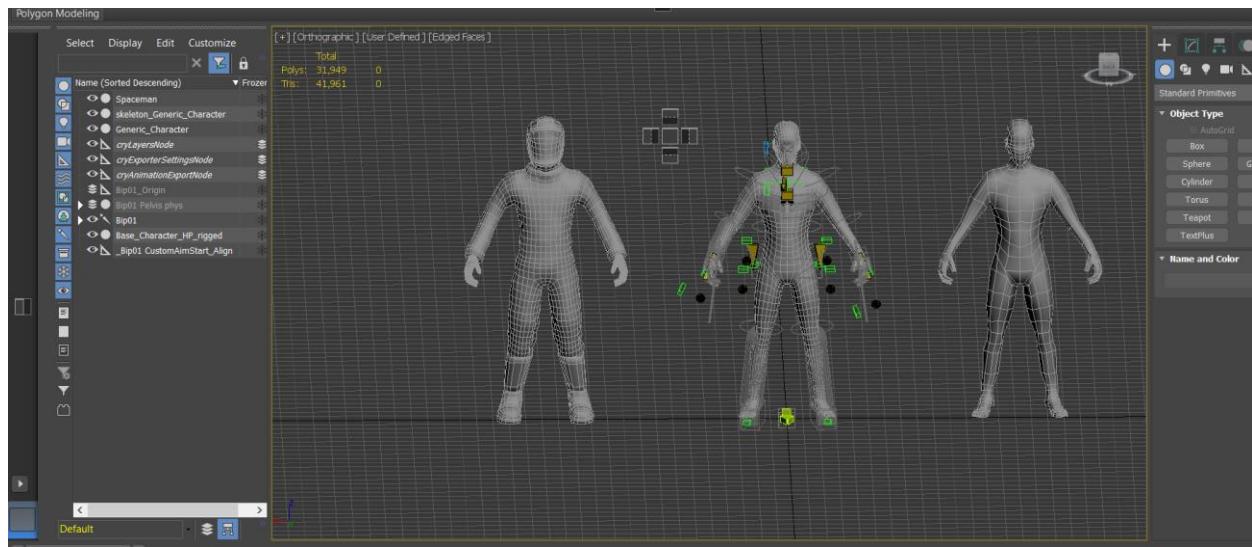
```

```
{  
StaminaBar._width = (Stamina)  
}  
  
function SetOxygen (Oxygen)  
{  
OxygenBar._width = (Oxygen)  
}  
  
function SetAmmo1 (Ammo1)  
{  
_root.Ammo1.text = (Ammo1)  
}  
  
function SetAmmo2 (Ammo2)  
{  
_root.Ammo2.text = (Ammo2)  
}  
  
function SetWeaponName (WeaponName)  
{  
_root.WeaponName.text = (WeaponName)  
}
```

## Flash – The HUD in designer view



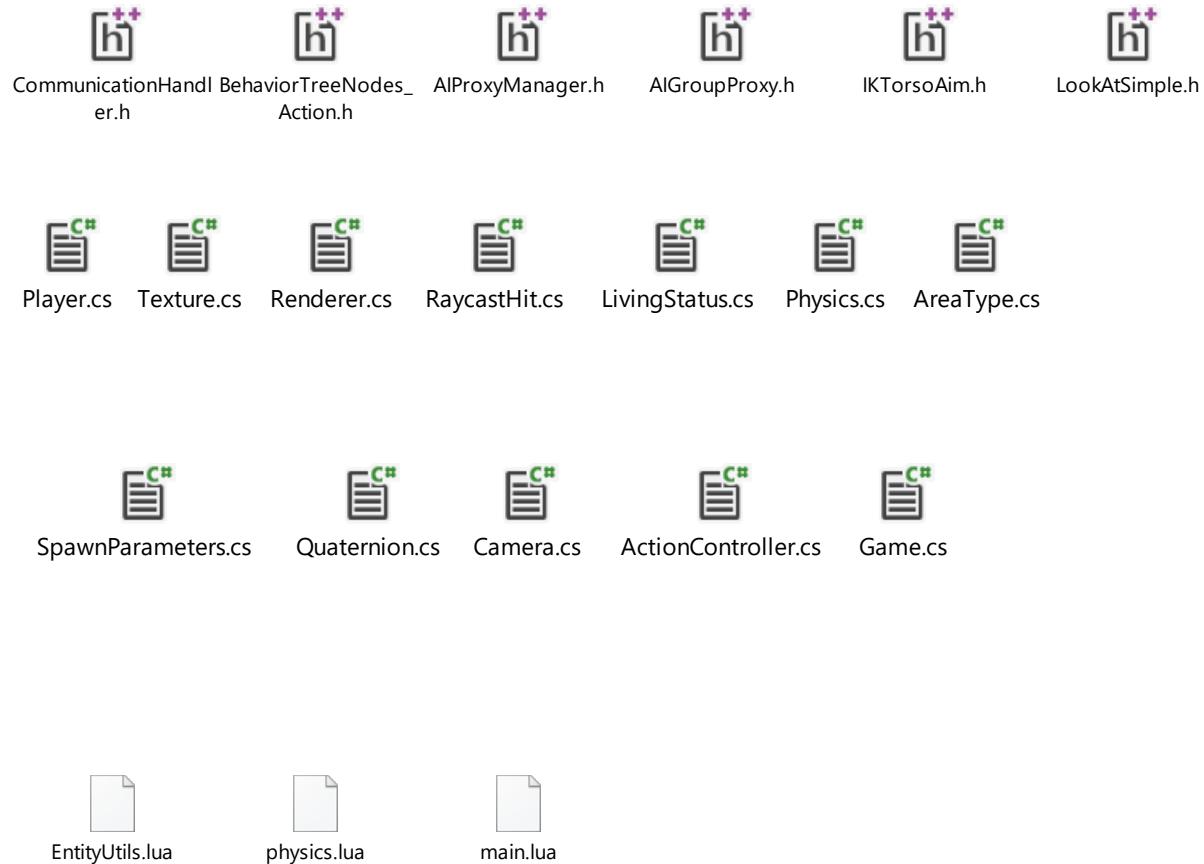
Default Skeleton and rig in 3D Max plus Cry Engine exporter:



Character rig on Cry Engine:



Some of the classes used with C#, C++ and LUA:



## O – AR Game



## AI Script

```
using UnityEngine;  
using System.Collections;  
  
public class enemyScript : MonoBehaviour {  
  
    // Use this for initialization  
  
    void Start () {  
  
        StartCoroutine ("Move");  
    }  
  
    // Update is called once per frame  
  
    void Update () {  
  
        transform.Translate(Vector3.forward * 3f * Time.deltaTime);  
    }  
  
    IEnumerator Move() {  
  
        while (true) {  
  
            yield return new WaitForSeconds (3.5f);  
  
            transform.eulerAngles += new Vector3 (0, 180f, 0);  
        }  
    }  
}
```

### **Camera Script:**

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class webCamScript : MonoBehaviour {

    public GameObject webCameraPlane;
    public Button fireButton;

    // Use this for initialization
    void Start () {

        if (Application.isMobilePlatform) {
            GameObject cameraParent = new GameObject ("camParent");
            cameraParent.transform.position = this.transform.position;
            this.transform.parent = cameraParent.transform;
            cameraParent.transform.Rotate (Vector3.right, 90);
        }
    }

    Input.gyro.enabled = true;
```

```

fireButton.onClick.AddListener (OnButtonDown);

WebCamTexture webCameraTexture = new WebCamTexture();
    webCameraPlane.GetComponent<MeshRenderer>().material.mainTexture
    webCameraTexture;
    webCameraTexture.Play();
}

void OnButtonDown(){

    GameObject bullet = Instantiate(Resources.Load("bullet", typeof(GameObject))) as
GameObject;
    Rigidbody rb = bullet.GetComponent<Rigidbody>();
    bullet.transform.rotation = Camera.main.transform.rotation;
    bullet.transform.position = Camera.main.transform.position;
    rb.AddForce(Camera.main.transform.forward * 500f);
    Destroy (bullet, 3);

    GetComponent< AudioSource > ().Play ();
}

// Update is called once per frame
void Update () {

```

```

Quaternion cameraRotation = new Quaternion (Input.gyro.attitude.x, Input.gyro.attitude.y, -
Input.gyro.attitude.z, -Input.gyro.attitude.w);

this.transform.localRotation = cameraRotation;

}

}

```

**Collision Script:**

```

using UnityEngine;

using System.Collections;

public class collisionScript : MonoBehaviour {

    // Use this for initialization

    void Start () {

    }

    // Update is called once per frame

    void Update () {

    }

    //for this to work both need colliders, one must have rigid body (spaceship) the other must
    have is trigger checked.

```

```

void OnTriggerEnter(Collider col)
{
    GameObject explosion = Instantiate(Resources.Load("FlareMobile", typeof(GameObject)))
as GameObject;
    explosion.transform.position = transform.position;
    Destroy(col.gameObject);
    Destroy(explosion, 2);

    if (GameObject.FindGameObjectsWithTag("Player").Length == 0){

        GameObject enemy = Instantiate(Resources.Load("enemy", typeof(GameObject))) as
GameObject;
        GameObject enemy1 = Instantiate(Resources.Load("enemy1", typeof(GameObject))) as
GameObject;
        GameObject enemy2 = Instantiate(Resources.Load("enemy2", typeof(GameObject))) as
GameObject;
        GameObject enemy3 = Instantiate(Resources.Load("enemy3", typeof(GameObject))) as
GameObject;

    }

    Destroy(gameObject);
}

}

```

## P – Web Platform code

### Index.php

```
<?php

// Initialize the session
session_start();

// Check if the user is logged in, if not then redirect him to login page
if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){

    header("location: login.php");

    exit;

}

?>

<?php

$con = new mysqli("localhost","id10385432_vel","Portugal79","id10385432_vel");

?>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>VEL | Velocity Engineering Light</title>

<link rel="shortcut icon" href="Images/vel.ico">
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<link rel="stylesheet" type="text/css" href="css/theme.css">

<style>

main {
    position: absolute;
    bottom: 50px;
    left: 220px;
    <!-- right: 200px; -->

    height:100%;
    width: 100%;

}

</style>

</head>
```

```
<body style=" overflow: auto;">

<?php include 'Components/header.html';?>

<main style="width:85%;height:86.5%;overflow:scroll;overflow-y:scroll;overflow-x:hidden;">

<div class="innertube" style= "overflow-y:scroll; overflow:visible;">

<video autoplay style="position: absolute; min-width: 100%; min-height: 100%; z-index:-1;" controls loop>

<source src="Images/games.mp4" type="video/mp4" >

</video>
```

```

</div>

<?php

$results_per_page = 3;

if (!isset($_GET['page'])) {

$page = 1;

} else {

$page = $_GET['page'];

}

// determine the sql LIMIT starting number for the results on the displaying page

$this_page_first_result = ($page-1)*$results_per_page;

?>

<center><h1 style = "Color: white">Current Games</h1></center>

<table style=" width: 30%; float: left">

<tr align="center">

<td>

Play

</td>

</tr>

```

```

<tr align="center" style=" height: 150; width:250; z-index:1;">
<td>
<a href = "Games/zombies.php">
<video autoplay height="150" width="250" controls loop muted>
<source src="Images/Zombies.mp4" type="video/mp4" >
</video>
</a>
</td>
</tr>

<tr align="center" style=" height: 150; width:250; z-index:1;">
<td>
<a href = "Games/startrooper.php">
<video autoplay height="150" width="250" controls loop muted>
<source src="Images/Star.mp4" type="video/mp4" >
</video>
</a>
</td>
</tr>

<tr align="center" style=" height: 150; width:250; z-index:1;" >
<td>
<a href = "Games/porradas.php">
<video autoplay height="150" width="250" controls loop muted>

```



```

<th></th>

</tr>

<?php

/* Fetch data from databse(select query) */

$res = $con->query("select * from tblregistration LIMIT {$this_page_first_result} ,
{$results_per_page} ") or die(mysqli_error($con));

while($row = mysqli_fetch_array($res))

{

?>

<tr>

<a href="https://velit.000webhostapp.com/Vel/Games/["RegFullName"].php">

<td align="center">">
height="150" width="250" alt="User image"> </image> </td>

<td align="center"></td>

<td align="center"><?php echo $row["RegFullName"]; ?></td>

<td align="center"></td>

<td align="center"><?php echo $row["RegEmail"]; ?></td>

<td align="center"></td>

<td align="center"><?php echo $row["RegGender"]; ?></td>

<td align="center"></td>

```

```

<td><a href="registration.php?isEdit=<?php echo $row["RegId"]; ?>"> </img> </a> </td>

<td> <a href=?delete=<?php echo $row["RegId"]; ?>"> </img></a></td>

<td></td>

</tr>

<?php

}

?>

</table>

<!--

<form method="post" action="">

<div id="div_pagination">

<input type="hidden" name="row" value=<?php echo $row; ?>>

<input type="hidden" name="allcount" value=<?php echo $allcount; ?>>

<input type="submit" class="button" name="but_prev" value="Previous">

<input type="submit" class="button" name="but_next" value="Next">

</div>

</form>

-->

<?php

// retrieve selected results from database and display them on page

$sql='SELECT * FROM tblregistration ';
```

```

$result = mysqli_query($con, $sql);

$number_of_results = mysqli_num_rows($result);

$number_of_pages = ceil($number_of_results/$results_per_page);

/*
while($row = mysqli_fetch_array($result)) {

    echo $row['RegId'] . ' ' . $row['RegFullName']. '<br>';

}

*/
// display the links to the pages

for ($page=1;$page<=$number_of_pages;$page++) {

    echo '<a href="index.php?page=' . $page . '">' . $page . '</a>';

}

?>

<p>
</p>
<p>
</p>

<form align="center" action="registration.php">
<input type="submit" value="Create Game" />
</form>

```

```

<p>
</p>

<p>
</p>

<?php

/* Delete code. When click on delete link this will perform.!*/
if(isset($_REQUEST["delete"]))
{
    $result=$con->query("select      RegProfile      from      tblregistration      where
RegId=".$_REQUEST["delete"]."") or die(mysql_error($con));

    while($row1=mysqli_fetch_array($result))
    {
        $image1=$row1["RegProfile"];

        unlink("images/".$image1);

        $con->query("delete from tblregistration where RegId=".$_REQUEST["delete"]."") or
die(mysql_error($con));

        echo          "<script>alert('Data
deleted
successfully..!');window.location='index.php';</script>";
    }
}
?>

```

</main>

<!--left menu-->

<?php include 'Components/left.html';?>

<!--right menu-->

<?php include 'Components/right.html';?>

<!--footer-->

<?php include 'Components/footer.html';?>

```
</body>

<script>

$(window).load(function() {

$('a.btn').click(function() {

$(this).toggleClass("active");

});

});

function openNav() {

document.getElementById("myNav").style.width = "100%";

}

function closeNav() {

document.getElementById("myNav").style.width = "0%";

}

function openNav2() {

document.getElementById("myNav2").style.width = "100%";



```

```
}
```

```
function closeNav2() {  
    document.getElementById("myNav2").style.width = "0%";  
}
```

```
function openNav3() {  
    document.getElementById("myNav3").style.width = "100%";  
}
```

```
function closeNav3() {  
    document.getElementById("myNav3").style.width = "0%";  
}
```

```
function openNav4() {  
    document.getElementById("myNav4").style.width = "100%";  
}
```

```
function closeNav4() {  
    document.getElementById("myNav4").style.width = "0%";  
}
```

```
function openNav5() {  
    document.getElementById("myNav5").style.width = "100%";
```

```
}
```

```
function closeNav5() {  
    document.getElementById("myNav5").style.width = "0%";  
}
```

```
function openNav6() {  
    document.getElementById("myNav6").style.width = "100%";  
}
```

```
function closeNav6() {  
    document.getElementById("myNav6").style.width = "0%";  
}
```

```
function openNav7() {  
    document.getElementById("myNav7").style.width = "100%";  
}
```

```
function closeNav7() {  
    document.getElementById("myNav7").style.width = "0%";  
}
```

```
function openNav8() {  
    document.getElementById("myNav8").style.width = "100%";
```

```
}
```

```
function closeNav8() {  
    document.getElementById("myNav8").style.width = "0%";  
}
```

```
function openNav9() {  
    document.getElementById("myNav9").style.width = "100%";  
}
```

```
function closeNav9() {  
    document.getElementById("myNav9").style.width = "0%";  
}
```

```
function openNav10() {  
    document.getElementById("myNav10").style.width = "100%";  
}
```

```
function closeNav10() {  
    document.getElementById("myNav10").style.width = "0%";  
}
```

```
function openNav11() {  
    document.getElementById("myNav11").style.width = "100%";
```

```
}
```

```
function closeNav11() {  
    document.getElementById("myNav11").style.width = "0%";  
}  
  
}
```

```
function openNav12() {  
    document.getElementById("myNav12").style.width = "100%";  
}  
  
}
```

```
function closeNav12() {  
    document.getElementById("myNav12").style.width = "0%";  
}  
  
}
```

```
function openNav13() {  
    document.getElementById("myNav13").style.width = "100%";  
}  
  
}
```

```
function closeNav13() {  
    document.getElementById("myNav13").style.width = "0%";  
}  
  
}
```

```
</script>
```

```
</html>
```

## Registration.php

```
<?php

$con = new mysqli("localhost","id10385432_vel","Portugal79","id10385432_vel"); /*

Connection code(Connect with database) */

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<style>

input[type=text],input[type=email],input[type=password]{

width:100% !important;

}

table{

border: 1px solid rgb(202,207,210);

}

form {

margin: 10% auto 0;

}

</style>

</head>

<body>

<?php
```

```

/* Start Update Code */

if(isset($_REQUEST["btnupdate"])) /* Update button click event.. */

{
    $image="";

    if($_FILES["profile"]["name"]=="")
    {
        $image=$_REQUEST["txtimg"];
    }

    else
    {
        unlink("images/".$_REQUEST["txtimg"]);

        move_uploaded_file($_FILES["profile"]["tmp_name"],"images/".$_FILES["profile"]["name"]);

        $image=$_FILES["profile"]["name"];
    }
}

$data = $con->query("update tblregistration set
RegFullName='".$REQUEST["txtfname"]."',RegEmail='".$REQUEST["txtuemail"]."',RegPas
sword='".$REQUEST["txtpassword"]."',RegGender='".$REQUEST["rdogender"]."',RegHobbi
es='".$chk."',RegProfile='".$image."' where RegId='".$REQUEST["txtid"]."');");
if($data == TRUE)
{

```

```

        echo             "<script>alert('Data           updated
successfully..!');window.location='index.php';</script>";
    }

else

{

    echo "<script>alert('Error while updating..!!')</script>";

}

}

if(isset($_REQUEST["btncancel"])) /* Cancel button click event.. */

{

    echo "<script>window.location='index.php';</script>";

}

if(isset($_REQUEST["btninsert"]))

{

    echo "<script>window.location='index.php';</script>";

}

/* End update code */

?>

```

```

<?php

/* Click on edit from display page check variable that you send from that page and fetch
data of that variable. */

if(isset($_REQUEST["isEdit"]))

{

```

```

$rs      =      $con->query("select      *      from      tblregistration      where
RegId=".$_REQUEST["isEdit"]."") or die(mysql_error($con));

while($row = mysqli_fetch_array($rs))

{

$ch1 = $ch2 = $ch3 = "";

$myArray = explode(',', $row["RegHobbies"]);

foreach($myArray as $chk)

{

if($chk == "Cricket")

{

$ch1 = 'checked';

}

if($chk == "Hockey")

{

$ch2 = 'checked';

}

if($chk == "Singing")

{

$ch3 = 'checked';

}

}

?>

```

```

<form method="post" enctype="multipart/form-data" >

<input type="hidden" name="txtimg" value="<?php echo $row["RegProfile"]; ?>">

<input type="hidden" name="txtid" value="<?php echo $row["RegId"]; ?>">

<center><h1>Edit Game Details</h1></center>

<table align="center" border="0">

<tr>

<td>Game Name:</td>

<td><input type="text" name="txtfname" value="<?php echo $row["RegFullName"]; ?>"></td>

</tr>

<tr>

<td>Description:</td>

<td><input type="text" name="txtuemail" value="<?php echo $row["RegEmail"]; ?>"></td>

</tr>

<tr>

<td>Link:</td>

<td><input type="text" name="txtpassword" value="<?php echo $row["RegPassword"]; ?>"></td>

</tr>

<tr>

<td>Details:</td>

<td><input type="text" name="rdogender" value="<?php echo $row["RegGender"]; ?>"></td>

<!--

```

```

<td>

<?php

/*
if($row["RegGender"] == "Male")

{

?>

<input type="radio" name="rdogender" value="Male"
checked="true">Male<input type="radio" name="rdogender" value="Female">Female

<?php

}if($row["RegGender"] == "Female")

{

?>

<input type="radio" name="rdogender" value="Male">Male<input
type="radio" name="rdogender" value="Female" checked="true">Female

<?php }

*/



?>

</td>

-->

</tr>

<!--

<tr>

<td>Evaluation:</td>

```

```

<td><input type="text" name="chkhobby" value="<?php
echo $row["chkhobby"]; ?>"></td>

</tr>
-->

<tr>
<td>Game Picture:</td>
<td><input type="file" name="profile"><br/>" alt="User Profile" height="100" width="100"></td>
</tr>
<tr>
<td colspan="2" align="center"><input type="submit" value="Update"
name="btnupdate"> <input type="submit" value="Cancel" name="btncancel"></td>
</tr>
</table>
</form>
<?php
}
}else{
?>

```

```

<form method="post" enctype="multipart/form-data" >
<center><h1>Insert Game Details</h1></center>
<table align="center" border="0">
<tr>

```

```

<td>Game Name:</td>
<td><input type="text" name="txtfname"></td>
</tr>

<tr>
<td>Description:</td>
<td><input type="text" name="txtuemail"></td>
</tr>

<tr>
<td>Link:</td>
<td><input type="text" name="txtpassword"></td>
</tr>

<tr>
<td>Details:</td>
<td><input type="text" name="rdogender"></td>
<!--
<td><input type="radio" name="rdogender" value="Male">Male<input
type="radio" name="rdogender" value="Female">Female</td>
-->
</tr>
<!--
<tr>
<td>Evaluation:</td>
<td><input type="text" name="chkhobby"></td>

```

```

<td><input type="checkbox" name="chkhobby[]" value="Cricket">Cricket
<input type="checkbox" name="chkhobby[]" value="Hockey">Hockey
<input type="checkbox" name="chkhobby[]" value="Singing">Singing</td>

</tr>
-->

<tr>
<td>Game Picture:</td>
<td><input type="file" name="profile"></td>
</tr>

<tr>
<td colspan="2" align="center"><input type="submit" value="Insert"
name="btninsert"></td>
</tr>
</table>
</form>
<?php } ?>

<?php
/* Insert Code Start */
if(isset($_POST["btninsert"])) /* Insert button click event */
{
    /* Move image into images folder which you have to create in
C:\xampp\htdocs\foldername\ */

```

```

move_uploaded_file($_FILES["profile"]["tmp_name"],"images/".$_FILES["profile"]["name"]);

$image=$_FILES["profile"]["name"];

/*
$checkbox1=$_POST['chkhobby'];

$chk="";
foreach($checkbox1 as $chk1)

{
if($chk == "")

{
$chk .= $chk1;

}

else{

$chk .= ",".$chk1;

}

}

*/



$res = $con->query("insert into
tblregistration(RegFullName,RegEmail,RegPassword,RegGender,RegHobbies,RegProfile)
values('".$_POST["txtfname"]."', '".$_POST["txtuemail"]."', '".$_POST["txtpassword"]."', '".$_PO
ST["rdogender"]."', '".$_POST["chkhobby"]."', '".$image."')") or die(mysqli_error($con));

if($res == TRUE)

{

```

```
echo "<script>alert('Data added successfully..!!')</script>";  
}  
  
else  
{  
    echo "<script>alert('Something getting wrong.Please try again..!')</script>";  
}  
}  
  
/* Insert Code End */  
  
?>  
</body>  
</html>
```