Tunisian republic
Ministry of Higher Education and Scientific Research University of Sousse

Higher Institute of Applied Science and Technology
of Sousse

IT Department

END OF STUDY INTERNSHIP REPORT

With a view to obtaining the Fundamental License degree
in: computer science
Option: computer science

Multiplayer FPS Game

Developed by: **Khalil Allah Dimassi**

Tasnim Kassebi

Supervised by:

Maha Khemaja — **Higher Institute of Applied Science and technology of Sousse**

Rami Rouini — **Pipeline Studio**

University Year: 2020/2021

Subject Code: LF-SI21-020

Acknowledgements

We take this opportunity to express our profound gratitude and deep regards to our internship supervisor Mr Rami Rouini for his exemplary guidance, monitoring and constant encouragement throughout the course of this internship.

We would like to express our heartfelt gratitude to Dr. Maha Khemaja, our professor, for her kind support and inspiration which has immensely strengthened our confidence during our internship program.

Finally, we would like to thank the administrative body for their contribution to the success of our graduate studies.

Table of Contents

Acknowledgements ii
Table of Contents iii
Figures List vii
Tables List viii
Project framework 10
1. Introduction 10
2. Company presentation 10
3. Methodology 11
3.1. The disciplines adopted 11
3.2. Software engineering adopted 11
3.3. Project presentation 12
3.3.1. Objective of the project 12
3.4. Video games domain 12
3.4.1. Types of video games 12
- Strategy games 12
- Sports Games 13
- Adventure games 13
- First-person shooting games 14
- MOBA games 14
3.4.2. Game modes 15
- Solo 15
- Multiplayer 15
3.4.3. conclusion 15
4. Concept and originality of our Game 15
5. chronogram 16
Similar games 17
1. Introduction 17
2. Counter-Strike: Global Offensive 18
2.1. Game concept 18
2.2. Game Play analysis 18
3. Apex Legends 19
3.1. Game concept 20
3.2. Game Play analysis 20

4. League of Legends 21
4.1. Game concept 21
4.2. Game Play analysis 22
5. Inspiration syntheses 22
6. Conclusion 23
Requirements specifications 24
1. introduction 24
2. Requirement analysis 24
2.1.1. What the player should be able to do 24
2.2. Functional requirements: Era's End 24
- What the Game provides 24
2.3. Functional requirements: The website 25
- What the Server provides 25
- What the Website provides 25
2.4. Non-functional requirements: Era's End 25
2.4.1. Graphic constraints 25
2.4.2. Ergonomic constraints 25
2.4.3. Technical constraints 25
2.4.4. Security constraints 26
2.4.5. Modularity constraints: Sectional Modularity 26
2.5. Non-functional requirements: Website 26
2.5.1. Graphic constraints 26
2.5.2. Ergonomic constraints 26
2.5.3. Technical constraints 26
2.5.4. Security constraints 26
2.5.5. Modularity constraints 27
3. Use case Diagrams 27
4. Product Backlog 30
5. Sprint 1: Website (registration, authentication, home page, profile) 32
5.1. Sprint backlog 32
5.2. "Sign up" use case diagram 32
5.3. Textual description: "sign up" 33
6. Sprint 2: Database creation 34
6.1. Sprint backlog 34
6.2. Diagram 35

7. Sprint 3: Game User Interface 35
7.1. Sprint backlog 35
7.2. Diagram 36
7.3. Textual description: “Login” 36
8. Sprint 4: Networking 38
8.1. Sprint backlog 38
8.2. Diagram 38
8.3. Textual description 39
9. Sprint 5: Design & Assets preparation 39
9.1. Sprint backlog 39
10. Sprint 6: Gameplay development 40
10.1. Sprint backlog 40
10.2. Diagram 40
Design 41
1. Introduction 41
2. Graphic design 41
2.1. Synopsis 41
2.2. Scenario 41
2.3. Navigation scheme 41
2.4. Graphic charter 42
2.5. Game’s logo 44
3. Technical design 45
Realization 48
1. Introduction 48
2. Working environment 48
2.1. Hardware requirements 48
2.2. Software workspace 49
2.2.1. Game 49
- Game engine: unity 49
- Script: Assembly-C# 49
2.2.2. Website 49
- Framework: Laravel 7 49
- Scripts: Php, HTML 5, CSS 3, Js 50
2.2.3. Assets 50
- Ui design: photoshop 50

- 3D modelling: Blender 51
- 3D animation: Autodesk Maya 2020 51

2.2.4. Collaboration workplace: GitHub 51

3. Sprint 1 realization: Website 52

3.1.1. Website 52

4. Sprint 2 realization: Database 53

5. Sprint 3 realization: Game UI 53

5.1. UI Interface 53

- champion's selection : 55
- player's information : 55

6. Sprint 4 realization: Networking 57

6.1. introduction 57

6.2. benefits 57

6.3. Most notable features 58

6.4. Development 58

6.4.1. Networking 58

7. Sprint 5 realization: Design & Assets preparation 59

7.1. Modelling 59

7.1.1. Map modelling 59

- Buildings 60
- Nature: 61
- Props: 62
- Champions modelling: 63
- Items modelling: 63

7.2. Animation 64

7.3. Integration 64

7.4. sound design 65

8. Sprint 6 realization: Gameplay Scenario 66

Conclusion and perspective 74

1. Perspectives 74

2. Conclusion 74

References 75

3. Netography 75

4. Bibliography 75

Figures List

Figure 1 PIPELINE logo 10
Figure 2 Scrum process representation 11
Figure 3 World of Warcraft - screenshot 12
Figure 4 Need for speed - screenshot 13
Figure 5 Assassin's creed - Screenshot 13
Figure 6 Call of duty - screenshot 14
Figure 7 League of Legends - screenshot 14
Figure 8 Gantt chart diagram 16
Figure 9 Apex Legends logo 17
Figure 10 CS:GO logo 17
Figure 11 League of Legends logo 17
Figure 12 Isometric Battle Arena 22
Figure 13 MVC Process Diagram 27
Figure 14 Game global use case diagram 28
Figure 15 Website Global use case diagram 29
Figure 16 "Sign up" use case diagram 32
Figure 17 "Sign up" sequence diagram 34
Figure 18 Entity-Relationship Diagram 35
Figure 19 Game UI "login" use case diagram 36
Figure 20 GUI "login" sequence diagram 37
Figure 21 "match making" use case diagram 38
Figure 22 in game use case diagram 40
Figure 23 Navigation Scheme 42
Figure 24 Account background 42
Figure 25 Login Form in-game 43
Figure 26 Buttons Style in profile 43
Figure 27 Champion Select Area 43
Figure 28 Account info Display 44
Figure 29 "Fun Game" font 44
Figure 30 Game Logo final design 44
Figure 31 Website & database class diagram 45
Figure 32 Match Class diagram 46
Figure 33 Package diagram 47
Figure 34 Game UI class diagram 47
Figure 35 C# logo 49
Figure 36 Unity Logo 49
Figure 37 HTML5 & CSS3 logo 49
Figure 38 Laravel logo 49
Figure 39 JavaScript Logo 50
Figure 40 Php Logo 50
Figure 41 Blender Logo 51
Figure 42 Photoshop 2021 Logo 51
Figure 43 Autodesk Maya Logo 51
Figure 44 GitHub Logo 51
Figure 45 Website Sign up page 52

Figure 46 Website Home Page 52
Figure 47 Website Login page 53
Figure 48 in-game login form: final design 54
Figure 49 in-game Start menu: final design 54
Figure 50 Ranks Logo designs 55
Figure 51 Leader Board: final design 56
Figure 52 Game Rules Explanation 56
Figure 53 Match Room interface: final design 57
Figure 54 Networking architectures 58
Figure 55 Arena Buildings samples 60
Figure 56 Arena's Nature samples 61
Figure 57 Arena's Props samples 62
Figure 58 Champions 3D Models 63
Figure 59 first item 3D model 63
Figure 60 third item 3D model 64
Figure 61 second item 3D model 64
Figure 62 Unity's Model import settings 65
Figure 63 Unity's Model rigging system 65
Figure 64 Game Scenario Activity diagram 66
Figure 65 Era's End Game Insight Activity diagram 67
Figure 66 in-game Team Leader interface 68
Figure 67 Main menu Interface 68
Figure 68 Potion item interaction (10 seconds shield) 69
Figure 69 Interaction with an enemy player 70
Figure 70 Book interaction (reads enemy status) 71
Figure 71 Coins interaction (add bonus points) 72
Figure 72 Results Declaration & Game's over 73

Tables List

Table 1 CS:GO Technical sheet 18
Table 2 Apex Legends technical sheet 19
Table 3 League of Legends Technical sheet 21
Table 4 Produced Backlog 31
Table 5 Sprint 1 Backlog 32
Table 6 "Sign up" textual description 33
Table 7 Sprint 2 backlog 35
Table 8 backlog of the third sprint 35
Table 9 "login" textual description 37
Table 10 backlog of the fourth sprint 38
Table 11 "start match" textual description 39
Table 12 backlog of the fifth sprint 39
Table 13 backlog of the sixth sprint 40
Table 14 System requirements 48

Project framework

1. Introduction

The following project was carried out as part of an end-of-studies internship in order to obtain a “Fundamental License in Computer Science” diploma. This internship lasts four months, and was carried out in the company "Pipeline studio".

The goal of this project was to design and produce a 3D video game. This report details the journey taken throughout the project, going from game analysis similar until the realization part.

2. Company presentation

Figure 1 PIPELINE logo

Pipeline studio (figure 1) provides virtual services that includes mobile games and mini-games, personalized video games, 3D modelling, VR and AR products, 3D films, trailers and announcements, 2D animations as well as training sessions in video game development and 2D / 3D design.

3. Methodology

3.1. The disciplines adopted

We linked our objectives by introducing into our work the subjects studied in class that would be relevant and useful in the realization of the project:

- **Object Oriented Design and Software Engineering** for the life cycle and design phases of the project.
- **Object-Oriented Programming** for the different programming steps in the project
- **Design and Graphic Design** for the creation of 3D objects as well as for the creation of the game interfaces.

3.2. Software engineering adopted

The speed of application development and flexibility in relation to functional needs are essential to be competitive in the production of large-scale software quality, especially when we have time constraints linked to often fluctuating specifications.

There is no doubt that the Scrum method is the most famous and commonly used agile method in the current ecosystem. It is considered a structured and quickly deployable project management method.

The Scrum method provides the possibility to control the progress of the project in an **iterative** form (more commonly referred to as a "**Sprint**"). The sprint lasts 1 to 4 weeks on average and is based on a specific goal:

We therefore chose UML for the design and scrum as an agile methodology.

- **UML**: is a graphic modelling language based on pictograms designed to provide a standardized method for visualizing the design of different views (static / dynamic) of a system.
- **Scrum**: Scrum is an agile method of responding to changing needs while productively delivering quality products.

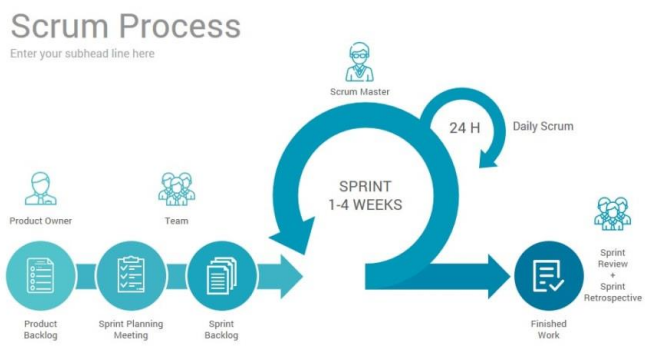

Figure 2 Scrum process representation

3.3. Project presentation

3.3.1. Objective of the project

The main goal of our project is to design and produce a 3D video game capable of capturing the attention of gamers aged 10 or over. As well as:

- Integrate into professional life and acquire new skills (script writing, design, Art Game, etc.).
- learn about the video game industry.
- become familiar with agile methods and use them as a framework for the management of our project.

3.4. Video games domain

The field of video games is booming, and is experiencing a meteoric rise. Many small development studios are stealing large market shares from the giants of the field. This clearly shows the potential that exists and the place that remains to be conquered in a field which only considers the originality of the idea and the quality of the realization.

3.4.1. Types of video games

The lack of consensus on the standardization of the criteria that define video games sometimes results in an inconsistency in their classifications. However, they can be classified in a more or less subjective manner. We are going to name some genres of the most played video games.

- Strategy games

Strategy games are divided into two types, real-time strategy games and turn-based strategy games. We are interested in the first type, which is the most played in the world of video games (). This kind of game requires skilful thinking, caution and a sense of planning in order to achieve victory.

Figure 3 World of Warcraft - screenshot

- Sports Games

Sports video games encompass many disciplines such as basketball, tennis, football, etc. They allow the player to play as an athlete or as a team of athletes. The objectives of a sports game are to represent as closely as possible the reality of the chosen sport and to take up the principles of reality without its constraints, aiming rather at the spectacular side of the discipline.

Figure 4 Need for speed - screenshot

- Adventure games

Adventure games are characterized by a focus on research and exploration, puzzle solving, rather than reflexes and strategies. The main objective is to tell a story like a movie or a comic book.

Figure 5 Assassin's creed - Screenshot

- First-person shooting games

It is an often FPS action game subgenre for "First-Person-Shooter". This type of video game is based on combat in first person vision, that is, the player sees the action through the eyes of the protagonist.

Figure 6 Call of duty - screenshot

- MOBA games

It stands for Multiplayer Online Battle Arena (MOBA). It is sometimes referred to as an action real-time strategy (ARTS) game. Essentially, the game is played between two teams of several players who choose a character from a roster of playable characters to level up and fight on a map with specific goals.

Figure 7 League of Legends - screenshot

3.4.2. Game modes

In the context of video games, game modes are the different ways the user can choose to play. We can classify them into two distinct categories.

- Solo

In the field of video games, the term single player game refers to a game mode involving a single player during the game and only involves a single player during the game, the latter playing generally against itself or against the artificial intelligence managed by the game.

- Multiplayer

The multiplayer mode allows several people to participate together and simultaneously in the same game. This functionality can be achieved either by sharing the same hardware or in network play mode.

3.4.3. conclusion

For the creation of our game, and after doing an analysis of the different types of video games, we chose to combine between the FPS, strategy, adventure and MOBA types.

4. Concept and originality of our Game

"Era's end" is a video game designed to entertain and amuse gamers. Besides the points' distribution by the leader gives the privilege to play with a strategy and always get an unexpected ending. Not to mention the non-violent fighting style (melee) and the items on the map that gives both teams the opportunity to turn tables at any time.

5. chronogram

A Gantt chart is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity.
To summarize (figure 7), a Gantt chart shows what has to be done (the activities) and when (the schedule).

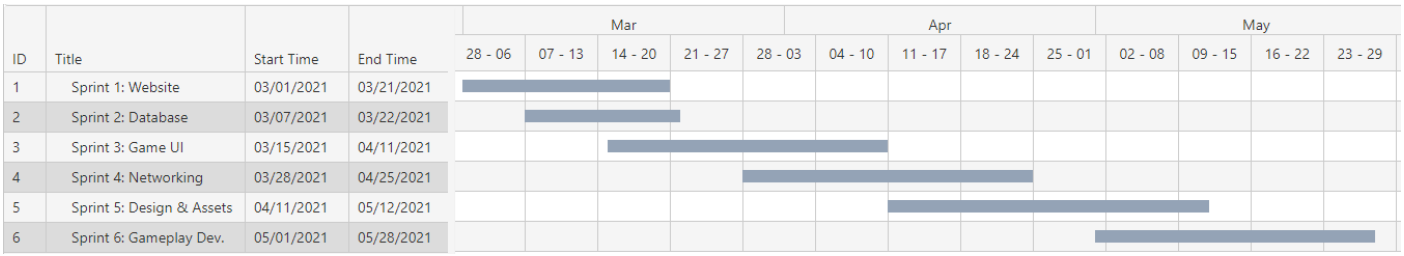

Figure 8 Gantt chart diagram

Similar games

1. Introduction

We cannot develop our first 3D first-person game without looking back and studying the similar ones.
In addition, in order not to get lost in this vast field, we must first limit ourselves and set a goal. The choice fell on world-famous games that have been on the market for a decade at least and others that are quite new.

The first analysis is based on one of the iconic games of the genre: ***Counter-Strike: Global Offensive*** *(figure 10)*.
Then, the next paragraph is devoted to the analysis of ***Apex Legends*** *(figure 9).* A relatively new shooter game with new gameplay mechanics.
To conclude, this chapter will include an analysis of the game ***League of Legends'*** *(figure 11)* networking system.

Figure 11 League of Legends logo

Figure 10 CS:GO logo

Figure 9 Apex Legends logo

2. Counter-Strike: Global Offensive

Technical sheet	
Description	***Counter-Strike: Global Offensive*** (***CS:GO***) is a multiplayer first-person shooter.
Version	Windows, Linux and Mac OS // PlayStation 3 // Xbox 360
Publisher	Valve
Developer	Valve & Hidden Path Entertainment
Engine	Source
Genre	first-person shooter
Mode	Local multiplayer / Online multiplayer
Target players	+12-year olds
Release	August 21, 2012

Table 1 CS:GO Technical sheet

2.1. Game concept

CS:GO is the fourth game in the Counter-Strike series. Developed for more than two years, "Global Offensive" was released first on PC, PS and Xbox almost a decade ago yet Valve will still update the game regularly, including smaller balance patches and larger content additions.

2.2. Game Play analysis

Like previous games in the "Counter Strike" series, "Global Offense" is a goal-based multiplayer first-person shooter game. The two opposing teams of terrorists and counter-terrorists compete in a game mode to repeatedly accomplish goals, such as ensuring locations where bombs can be placed or dropped, and rescuing or capturing hostages.

At the end of each short round, players will be rewarded with in-game currency based on individual and team performance so that they can be spent on other weapons or tools in subsequent rounds. Winning games

usually reward more money than losing them, and completing map-based goals (including killing enemies) will provide additional cash rewards.

The game uses different goal-based game modes to set the terrorist and counter-terrorist teams against each other. The most common game mode involves terrorists planting bombs, while counter-terrorists try to stop the bombs.

There are eight official game modes, all of which have unique characteristics specific to that mode. The game also has a matching support function, allowing players to play on a dedicated Valve server.

3. Apex Legends

Technical sheet	
Description	**Apex Legends** is a free-to-play hero battle royal game
Version	Windows // PlayStation 4 // Xbox One// Nintendo Switch // Android // iOS
Publisher	Electronic Arts
Developer	Respawn Entertainment
Engine	Source
Genre	first-person shooter // Battle royal
Mode	Online Multiplayer
Target players	+16-year olds
Release	February 4, 2019 for Windows, PS4 & Xbox One March 9, 2021 for Nintendo Switch E/A 2022 for Android & iOS

Table 2 Apex Legends technical sheet

3.1. Game concept

The game is in the same science fiction world as Respawn Entertainment's "Titanfall" and "Titanfall 2".

Apex's release in 2019 was surprising, because until then it had been assumed that Respawn Entertainment was working on the third instalment for the studio's previous main game, the "Titan Fall" series. Apex Legends was well received by critics, praising its gameplay, progress system and the integration of various genre elements. Some people think it is a strong competitor to other Royal Royale games.

3.2. Game Play analysis

Each match generally features twenty teams of three-player squads. Before the match, each player on the squad selects one of the 16 playable characters (as of Season 8). Each character in the squad has a unique design, personality, and abilities that provide different playstyles to the team. All teams are then placed on an aircraft that passes over the game map and the team's leader (randomly selected) chooses when and where to jump.

Once on the ground, the squad can scavenge for weapons, armour, and other equipment that is scattered around buildings, or in crates randomly distributed around the map.

Apex includes a nonverbal communication "ping system" which allows players to use their game controller to communicate to their squad certain directions, weapon locations, enemies, and suggested strategies which was the newest addition to this game genre and it was so great that most other games implemented the system shortly after its appearance in Apex.

Over time, the game's safe zone will reduce in size around a randomly selected point on the map; players outside the safe zone take damage and may die if they do not reach the safe zone in time. This also confines squads to smaller spaces to force encounters. The last squad with any members left alive is crowned the "Apex Champions" of that match.

4. League of Legends

Technical sheet	
Description	**League of Legends** is a free-to-play highly competitive global game with more than 200 million active players.
Version	Windows, OS X
Publisher	Riot Games Garena
Developer	Riot Games
Engine	Riot's proprietary game engine (core: C++)
Genre	Multiplayer online battle arena
Mode	Online Multiplayer
Target players	+13-year olds
Release	October 27, 2009 for Windows March 1, 2013 for OS X

Table 3 League of Legends Technical sheet

4.1. Game concept

League of legends is often considered to be the largest e-sports in the world, with an international competitive scene consisting of 12 leagues. The domestic league reached its peak in the annual League of Legends World Championship. The 2019 champion has more than 100 million independent viewers, and the highest number of simultaneous viewers is 44 million. Live streaming sites such as Twitch, YouTube, Bilibili, and cable sports channel ESPN are all broadcasting domestic and international events.

4.2. Game Play analysis

League of Legends is a Multiplayer Online Battle Arena (MOBA) game in which the player controls a character ("champion") with a set of unique abilities from an isometric perspective (figure 12).

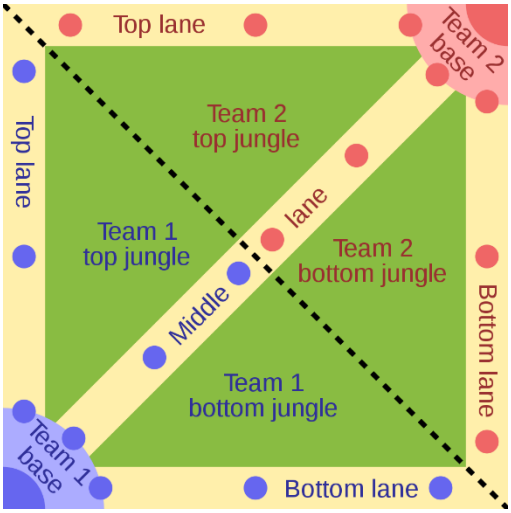

Figure 12 Isometric Battle Arena

As of April 2021, 155 champions can participate in the competition. In the course of the game, the champion gains experience points (XP) by killing enemies, thereby gaining levels. Items also increase the strength of the champion and are purchased with gold. Over time, players will passively accumulate gold coins and actively make money by defeating the opposing team's minions, champions or defensive structures. In the main game mode (Summoner's Rift), items are purchased through the shop menu, and the shop menu is only available to the player when the player's champion is in the team base. Each match is discrete; levels and items will not be transferred from one game to another.

5. Inspiration syntheses

As gamers that have played these games for a long time, programmers that enjoyed tackling problems and creating our own small things: these traits is what inspired our team to have a game as our project and these games is where we found our most inspirations.

- We chose the perspective to be “first-person” (like Apex or CS:GO) because of how immersive it tends to get. Having the player see the world of the game from the same angle as the character he’s controlling feels like actually being there and doing what the character does.

- Using a simple locomotion pack as eight directional movement and a sprint, it helps the players learn the game very quickly and leaves room for the dedicated ones to fully master it.
- Adding small side objectives like collecting hidden items, avoiding traps and exploring the large map, a technique used in games like Apex and League helps tremendously to keep the players' experience fresh and non-repetitive in each match.
- Our game depends much on strategic team plays thus having the ability to communicate with each other in-game using a "voice" or "text" chat is very important.
- All major competitive games have very strong restrictions on the player age because of how gruesome the gameplay tends to be and so we thought, let us remove this restriction by simply removing the "killing" part of the game so any child can hop in and enjoy the game without any disturbing scenes.

6. Conclusion

So many games are already out in the market dominating the gaming scene and make the competition barely possible specially for indie game developers such as our team. But even so, the gamers community is ever growing and hungry for new releases thus giving new games a chance to pop into popularity if it satisfies the masses.

Requirements specifications

1. introduction

Specifying requirements is an important and difficult step. It is used to define the contract aspect. It establishes the guidelines and constraints that applications must follow.

2. Requirement analysis

To get a better understanding of the game, we need to specify the needs of its main actors. Era's End has one primary actor which is the Player, and one secondary actor that the game interacts with is the database's Manager.

But first and foremost, the Player needs a way to download the game and also, a way to create his account: and therefore, like every other online multiplayer game, Era's End needs its own website.

2.1.1. What the player should be able to do

- create an account on the website.
- download the game from the website.
- access the account information from the website.
- login from the game.
- start/join matches.

2.2. Functional requirements: Era's End

- What the Game provides

- A profile page containing the account's details and match history.
- A champ select area where the players can prepare their character before the match.
- A leader board containing the top 10 players on the server depending on their personal level.
- A start match button so the player can start playing.
- A character that the player controls and moves around the map.
- An actual map where the player can move around and interacts with.
- A Game objective(s) that when achieved the match ends.

2.3. Functional requirements: The website

- What the Server provides

- robust and secure database containing players accounts and data.
- being up and running 24/7 with no significant delay or package loss.
- stable matchmaking system that does not disconnect in match.

- What the Website provides

- authentication: register and login.
- profile page with accounts data and match history.
- a “Download the Game” button on the home page.

2.4. Non-functional requirements: Era’s End

To make the game attractive, it is essential to define the graphic, ergonomic, technical, security and modularity constraints.

2.4.1. Graphic constraints

Quality and graphics are the main image of the game. Indeed, the first thing players notice is the graphics. The balance of shapes and harmony of colours should first attract the eyes. This is why every game must abide by its own graphics charter.

2.4.2. Ergonomic constraints

Before anything else, the game UI must be simple, understandable and above all readable. In addition, the game must provide visual consistency and quick access to information. It is important not to forget to limit the overloading of the menus.

2.4.3. Technical constraints

Programming in an object-oriented language is very important because it helps to organize and therefore the project is easier to manage. It is reusable and its method is more intuitive.

Also, the game must be cross-platform in order to benefit a large number of users, considering the capacity of the various machines on the market.

Then, it is essential to use the Lightmapping system for scene lighting, because it can present static light and very detailed shadow shapes.

2.4.4. Security constraints

The registered user's information is fully secure in a private database online. Also, the "sign in" and "sign up" methods use the PHP password hashing algorithm to apply a thicker protection layer on the player's account.

Furthermore, the users can utilise the official website to manage their accounts: change the email or the password, whenever they feel threatened.

2.4.5. Modularity constraints: Sectional Modularity

In order to avoid the risk of errors and to facilitate the programming of the game, it is necessary to write the game code in several independent modules.

Sectional Modularity is a form of modularity adopted in mass customization. Segmented modularization allows a set of components selected from a set of component types to be configured in any way, as long as these components are connected at their interfaces. Each component can have one, two or more interfaces to allow sequential addition of components. Lego bricks can show the most basic form of modularity. A seemingly unlimited number of items can be constructed from a set of Legos.

2.5. Non-functional requirements: Website

2.5.1. Graphic constraints

Because the website is the first place the players need to visit to start their journey, it is necessary for it to be welcoming and intuitive: using high spirited colours and attracting images/animations.

2.5.2. Ergonomic constraints

The players would need to register before anything else and so, the GUI and forms have to be easily accessed, well organised and fully comprehensible.

2.5.3. Technical constraints

After consulting our industrial and university supervisors, we agreed to use the Laravel framework because of how easy it was to learn and how beginner friendly it was. It helped us move forward very swiftly and keep our work well organised all the time.

2.5.4. Security constraints

The Laravel framework provides excellent user's data security using the built in Auth System. Registration requires a powerful password, a working email and a verification mail to fully activate the player's account before he can launch the game. The Auth System provides password encryption security as well by hashing the input before saving them in the database tables.

2.5.5. Modularity constraints

Model View Controller, which is popular as MVC (figure 13), is a widely used model currently used for Web development purposes. It is popular in various major programming languages today, such as Java, Python, Ruby, PHP, but the MVC model is widely used for web application development purposes on the ASP.net platform.

3. Use case Diagrams

The Use Case diagram defines the relationship between the system

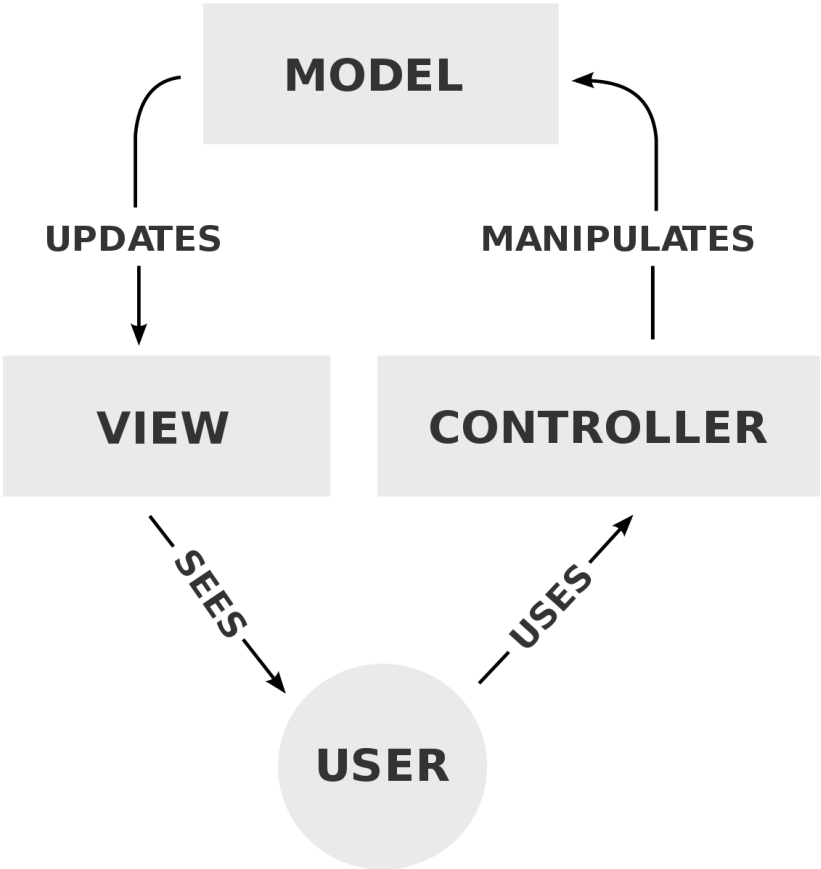

Figure 13 MVC Process Diagram

and its actors. Indeed, it gives a global view of the behaviour of the system.

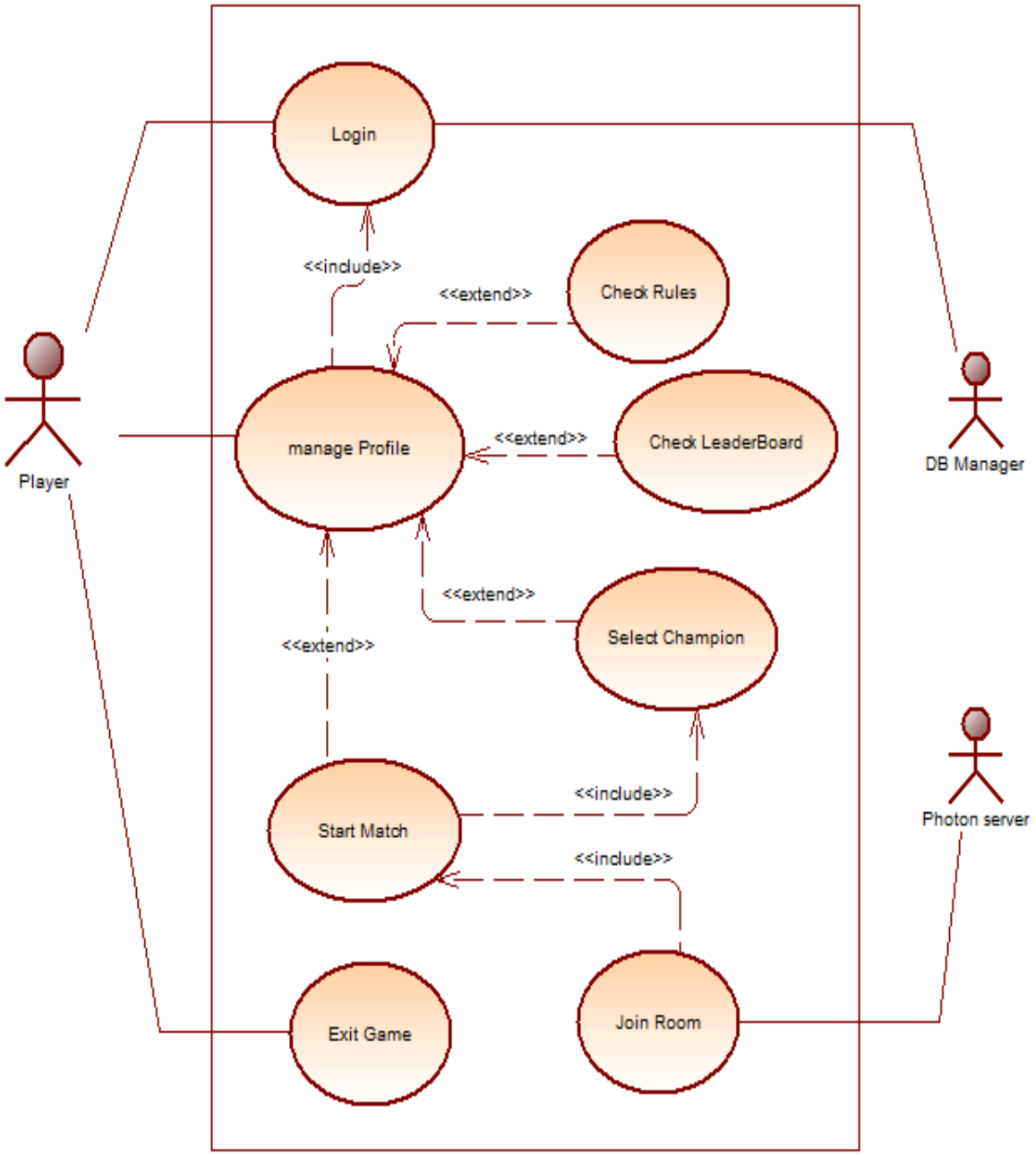

Figure 14 Game global use case diagram

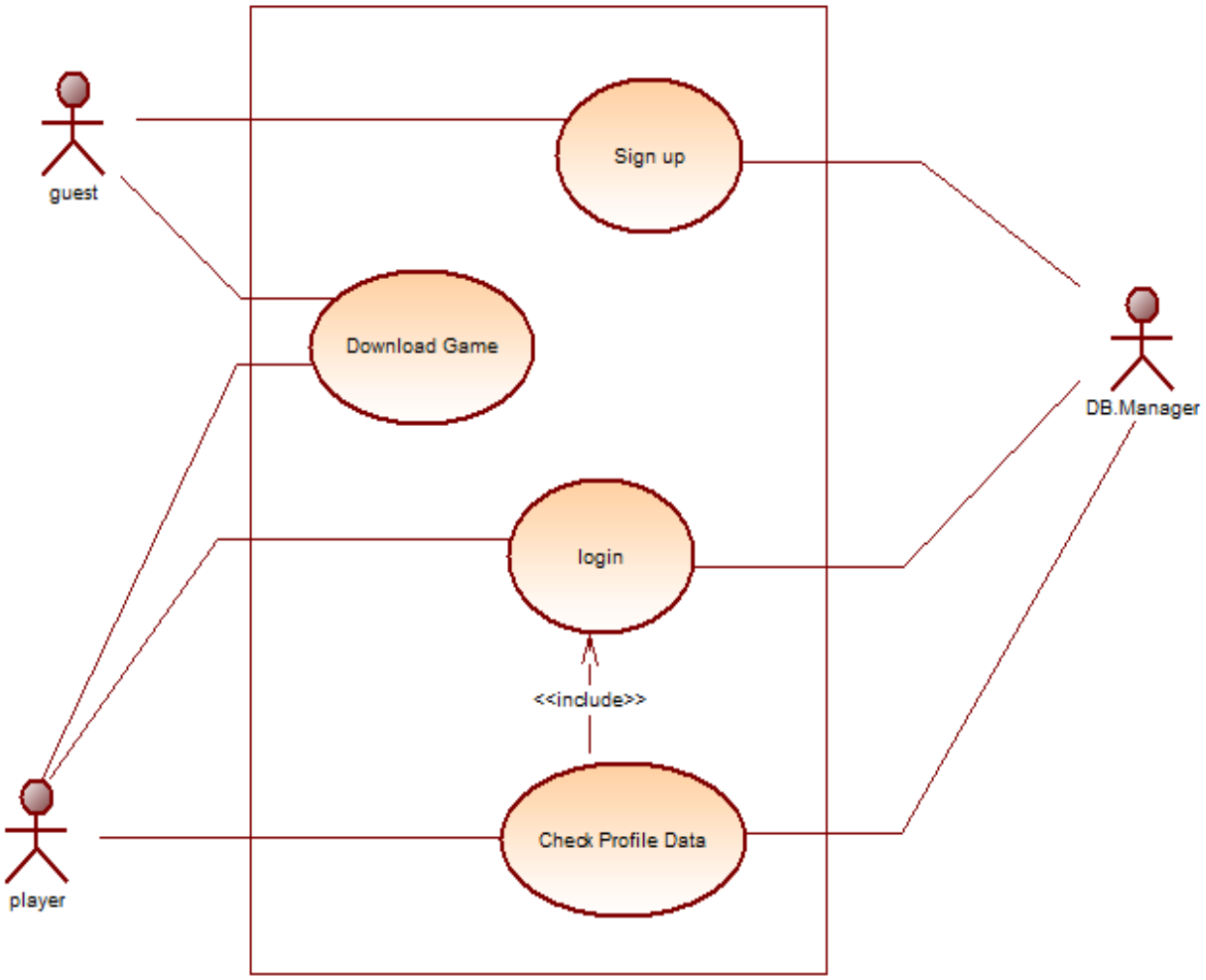

Figure 15 Website Global use case diagram

4. Product Backlog

The product backlog: is intended to expose a prioritized list of features to be developed. The elements of the "User Story" product backlog which presents the functionality expected by the customer. Usually presented as follows: as A, I wish B, in order to.

- **Priority**: The MoSCoW method helps the team know what to do first, what needs to be done later, and what not to do. MoSCoW has the meaning:
  - **M - Must**: These are really the critical points, if these features are not met, the success of the project will suffer and lead to its failure.
  - **S -Should**: These features are important but not urgent.
  - **C - Could**: These features can be achieved as long as it has no effect on other tasks.
  - **W - Won't**: These features are not part of the project, but they are included in the optional part.
- **Story point**: This approach is used to predict the work to be done to develop and deliver its User Stories. To decline the amount of effort required, we use the Fibonacci sequence (1, 2, 3, 5, 8, 13, 21...)

Title	As	I want to	In order to	Priority	Story points
Sign up	Guest	Input personal information	Create an account	MUST	3
login	Player	Input login details	• Get access to website profile • Get my games history • Check my rank and level	SHOUD	3
Download Era's End	Guest Player	Download the game	• Install Era's end and get to play	MUST	5
Database	Player	Save my progress Save my account	• Play online • Evaluate my skills over time	MUST	5
Game login	Player	Input login details in game UI	• Get access to the game by my unique account	MUST	5
Profile management	Player	• Check my progress in game • Check my rank in the leader board • Check the game rules • Select a champion • Start a match	• Keep a record of my skills growth • Keep notes about the other players skill level • Learn about the games more • Pick a character to play as. • Play a match with other players	MUST	8

Lobby	Player	See which team I'm on and which players I'm playing with	• Learn beforehand my spawn location on the map. • Predict the possible outcomes of the match	MUST	2
Characters	Player	Choose the best-looking character	• Relates to the character as much as I want • Get more depth into the game	SHOULD	3
Arena	Player	Explore the map	• Enjoy the ambient feeling of the arena for the entire match	SHOULD	5
Sounds	Player	Have a full experience	• Not being disturbed by weird voices whilst focusing on gameplay	SHOULD	2
Animation	Player	Enjoy playing certain characters	• Not being disturbed by weird character movements in games • Not worrying about my character breaking in game.	SHOULD	8
Server	Player	have a stable connection in the game	• Enjoy the gameplay without interruption • Not worrying about losing due to disconnection	MUST	3

Table 4 Produced Backlog

5. Sprint 1: Website (registration, authentication, home page, profile)

5.1. Sprint backlog

The Sprint Backlog contains a list of tasks identified by the Scrum team that should be completed before the end of the sprint. “Table 4” represents the backlog of the first sprint.

ID	User Story
1	Install and configure Laravel, Composer & GitHub repository
3	Create the main views: home view, login view, sign up view & profile view.
4	Create the authentication and security algorithms.
5	Create website theme and implement the new design.
6	Add the download button in the home page to download the game.

Table 5 Sprint 1 Backlog

5.2. “Sign up” use case diagram

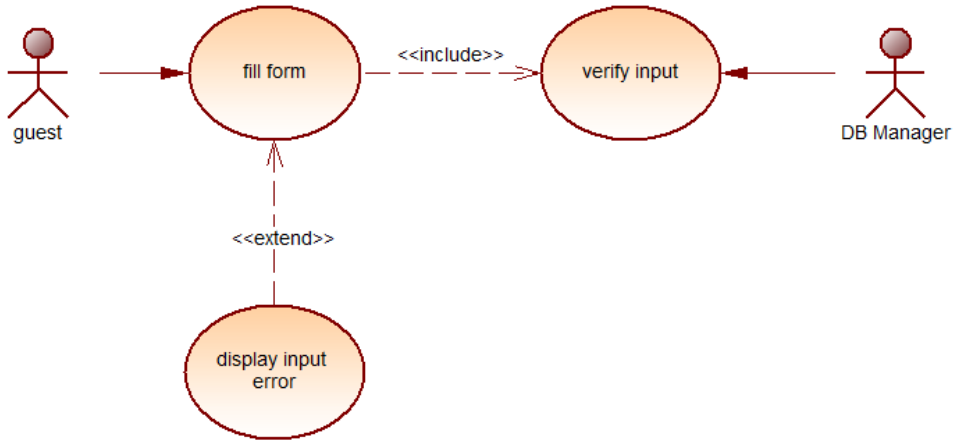

Figure 16 "Sign up" use case diagram

5.3. Textual description: “sign up”

it is very important to establish a text description of the use case. It makes it possible to clarify the progress of functions and describe the chronological sequence of actions that must be performed. “Table 5” describes the “sign up” use case followed by “figure 17” the sin up sequence diagram.

Title	Sign up
Goal	Create a new account
Summary	new users need to create an account to play the game later.
Actors	Guests (unregistered users) // DB manager
Preconditions	○ The user must have internet access. ○ The user must have an email and have access to its inbox
Postconditions	○ The user’s new account is created and automatically granted access to the website’s features.
Nominal chaining	1. The user fills all the required fields accordingly. 2. The system verifies the inputs. 3. The user gets a confirmation email sent to their inbox. 4. The user verifies the email sent. 5. The user is redirected to the homepage where he can download the game or check/edit his profile details.
Alternative sequence N ° 2	1. The system detects the error in the provided data and deduces its type. 2. The webpage delivers a clear error message and guides the user to fixing the error(s).

Table 6 "Sign up" textual description

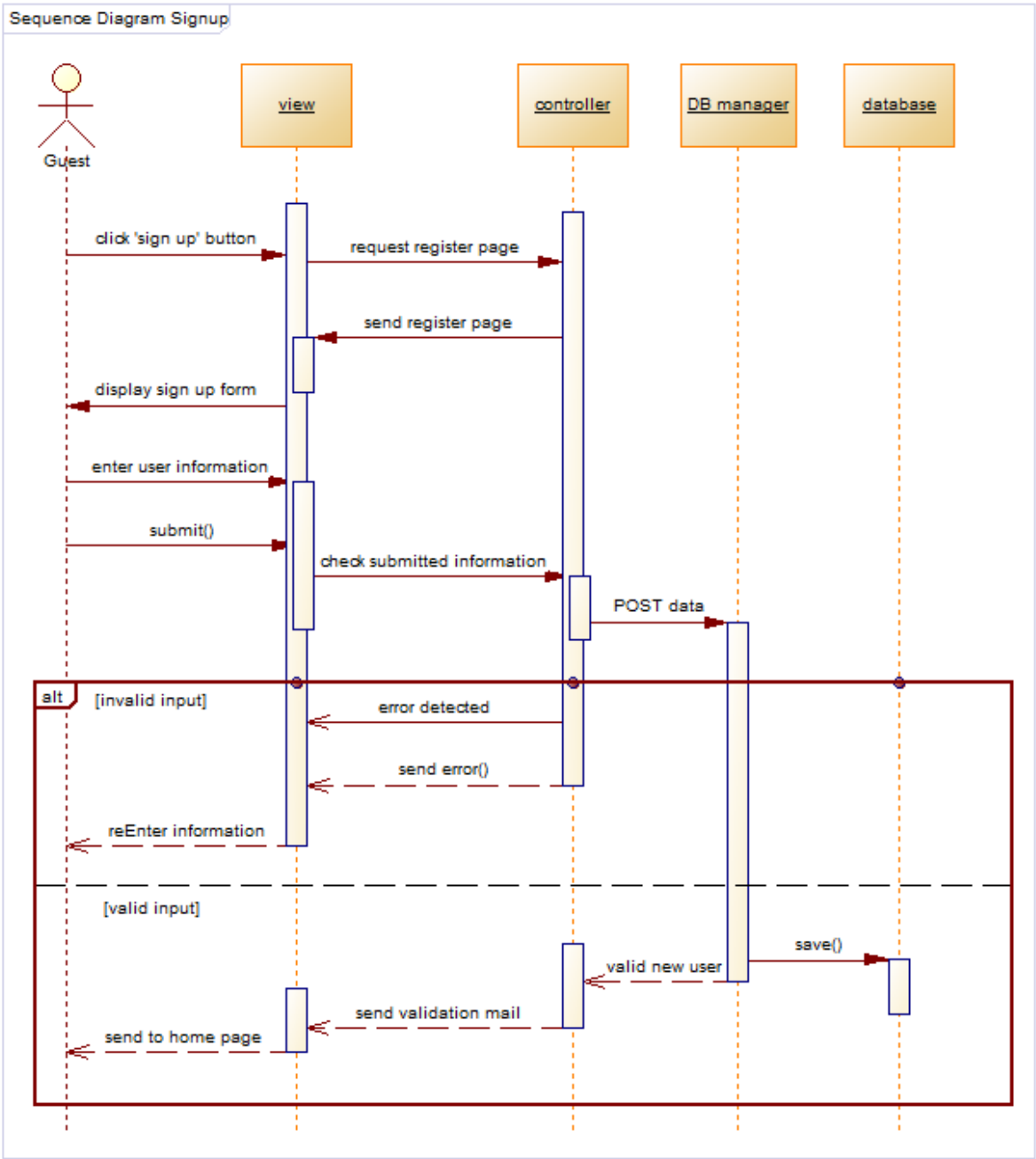

Figure 17 "Sign up" sequence diagram

6. Sprint 2: Database creation

6.1. Sprint backlog

"Table 6" represents the backlog of the second sprint: database creation.

ID	User Story
1	Study the functional requirements of the game data
3	Create the main table “users” for accounts creation
4	Create the “profiles” table for account information holding
5	Create the “games” table for match history recording

Table 7 Sprint 2 backlog

6.2. Diagram

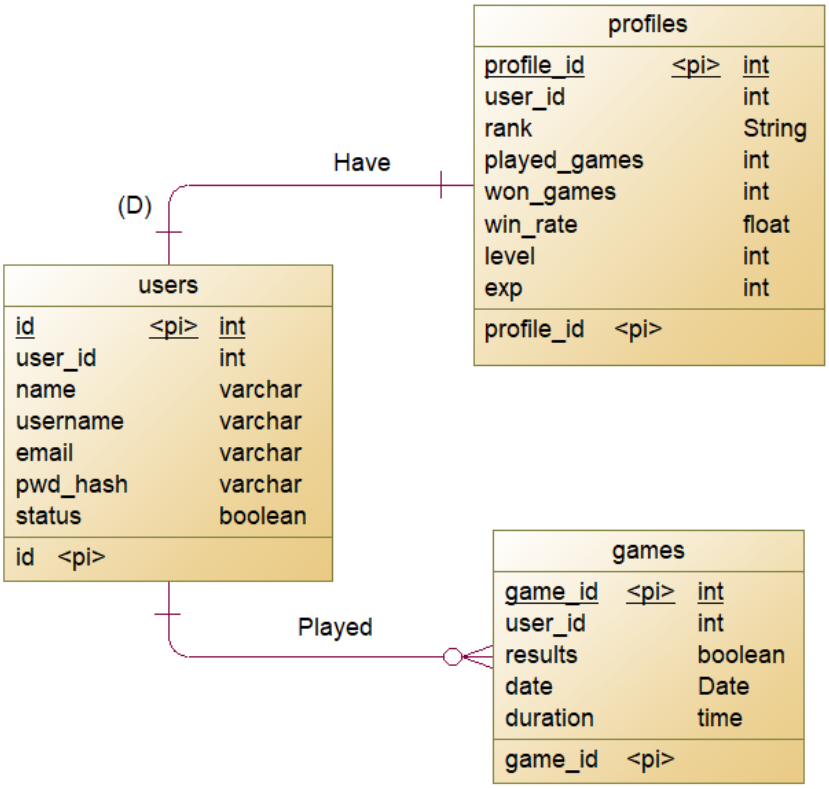

Figure 18 Entity-Relationship Diagram

7. Sprint 3: Game User Interface

7.1. Sprint backlog

ID	User Story
1	Study the functional requirements of the game user interface (GUI)
3	Create the canvases for the main interfaces (login form, main menu)
4	Create and implement the login algorithm
5	Make the database manager and link it to the login form
6	Implement the main functions of the main menu scene (check profile info, check leader board, check game rules, select champion and start match)

Table 8 backlog of the third sprint

7.2. Diagram

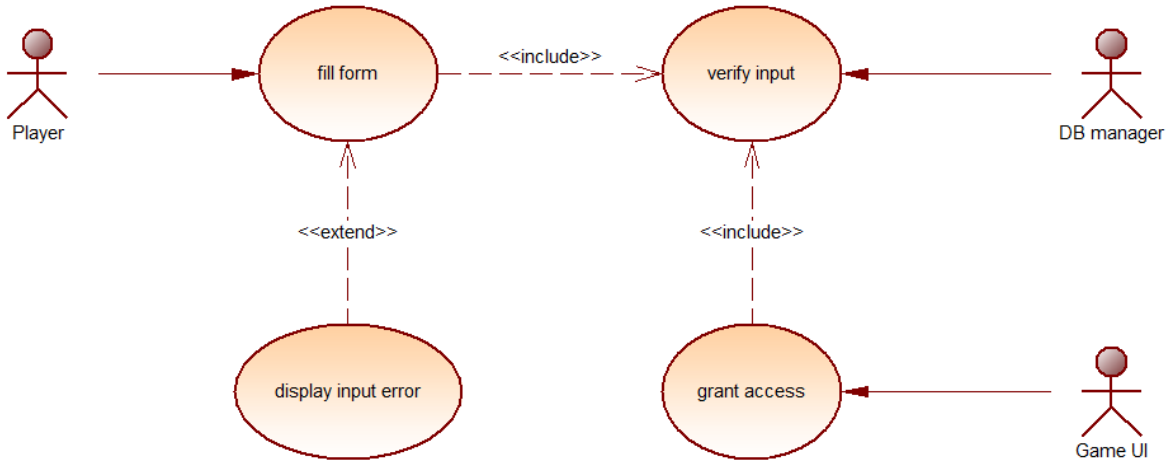

Figure 19 Game UI "login" use case diagram

7.3. Textual description: "Login"

Title	Login
Goal	Get access to Era's End
Summary	The player must identify himself with his Username and Password
Actors	Player
Preconditions	o The player must have access to the internet. o The player must already have registered an account on the website.
Postconditions	o The player signed in and gained access to the game.
Nominal chaining	**1.** The player enters their username and password in the provided fields. **2.** The system verifies the authentication of the player's username and password. **3.** The player is redirected to the interface which contains the main menu.
Alternative sequence N ° 2 Either the username or the password is incorrect or account logged in from another device	**1.** The system detects the error in the provided data and deduces its type. **2.** The game displays an error popup to the player with information. **3.** The player is sent back to the login form to try again.

Description of HMI requirements HMI = human machine interaction	o Two input fields for ID & password. o Two buttons for submitting input data & cancelling the operation (quitting the game). o A popup with a comprehensible message identifying the authentication problem.

Table 9 "login" textual description

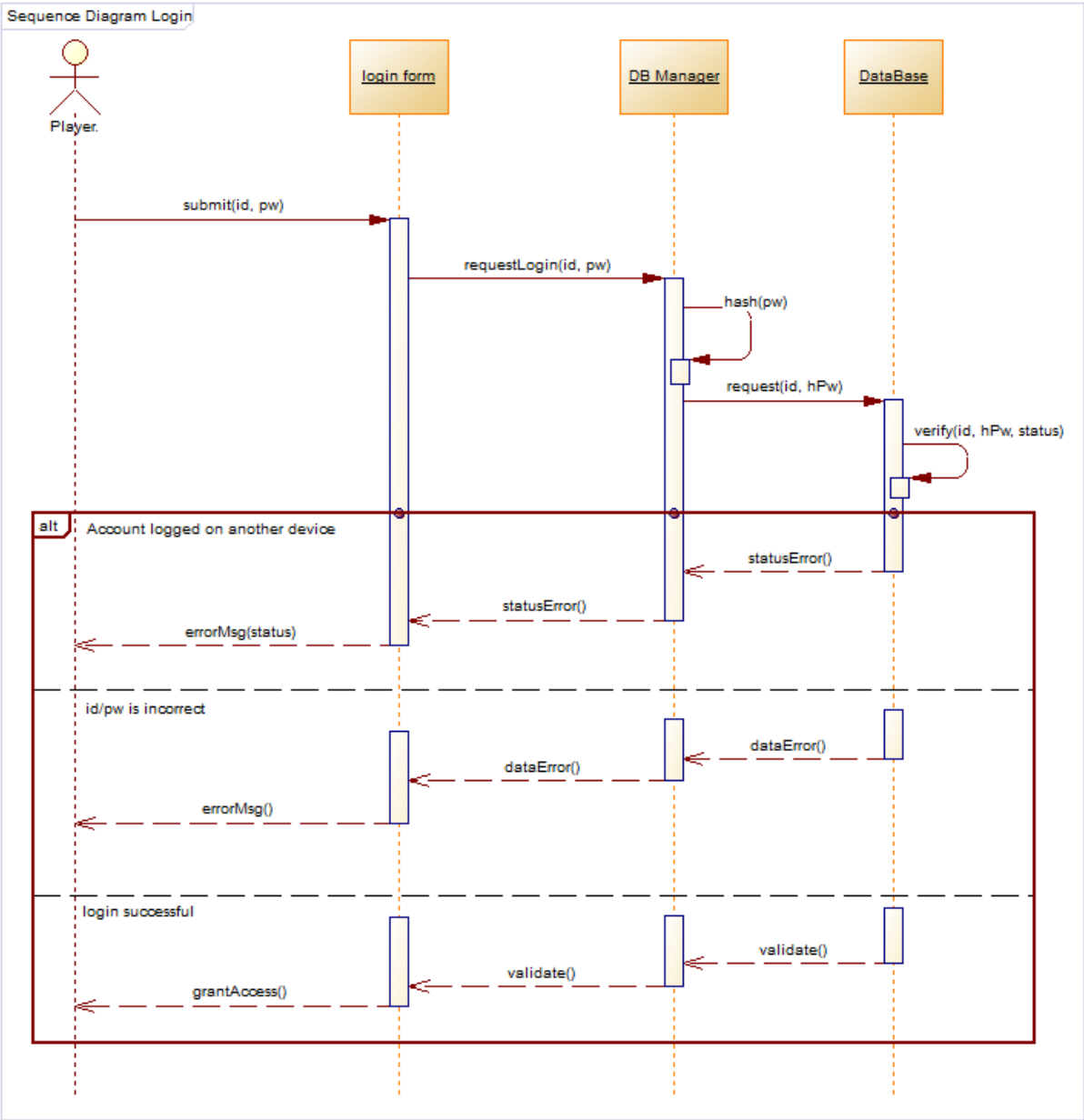

Figure 20 GUI "login" sequence diagram

8. Sprint 4: Networking

8.1. Sprint backlog

ID	User Story
1	Study the available networking algorithms for unity
3	Picking “Photon Unity Networking 2” as the main networking service
4	Studying the best networking architectures for our game needs
5	Implementing the “client-server” architecture in the game
6	Creating the match making algorithms

Table 10 backlog of the fourth sprint

8.2. Diagram

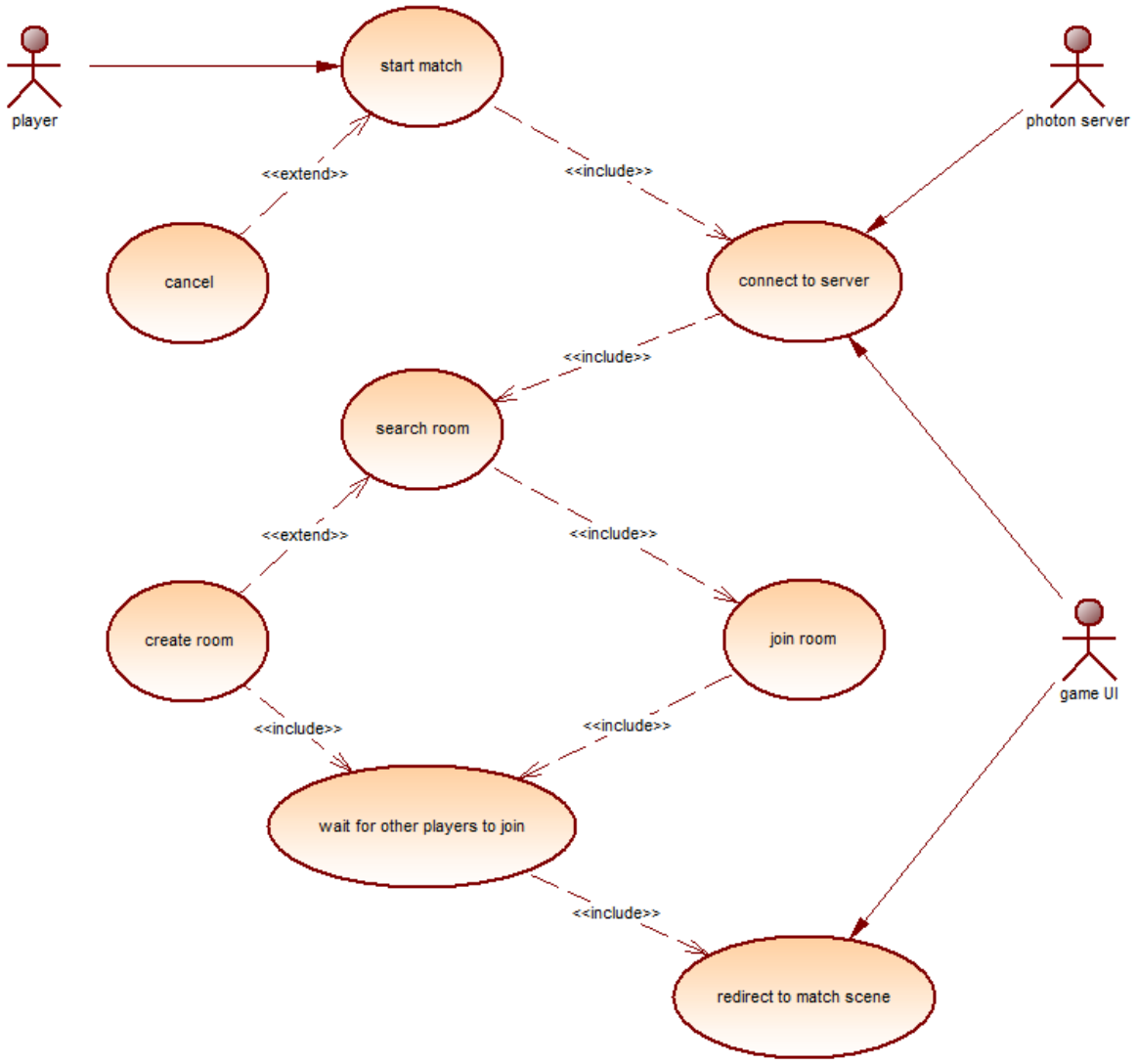

Figure 21 "match making" use case diagram

8.3. Textual description

Title	Start match
Goal	start the gameplay
Summary	the game starts looking for open rooms, if there is none then create one and wait for additional players
Actors	Player
Preconditions	○ The player must have access to the game. ○ The minimum number of players must be online.
Postconditions	○ The player joins a team and spawns in the Arena.
Alternative sequence N ° 2	**1.** The player waits for the game to connect to the main servers as indicated on the UI. **2.** The player can pick a character to play with as well as a cosmetic skin for that character. **3.** Once the “Launch” button is revealed the player can start the matchmaking. **4.** The game sends a request to the server for the “open rooms” list. **5.** The game joins a random open room and assigns the player to a team. **6.** Once the room completely fills up or the timer is up, the players are transported into the arena to start the match.
Description of HMI requirements	○ The “Launch” button is greyed out until the game is connected to the server. ○ Six buttons with a display to indicate which character the player picked for the next match. ○ A “cancel” button if the player wants to stop looking for a match.

Table 11 "start match" textual description

9. Sprint 5: Design & Assets preparation

9.1. Sprint backlog

ID	User Story
1	Decide on a game theme
3	Choose the colour palette according to the theme
4	Create GUI assets + Ranks symbols
5	Implement character models online + animations
6	Implements the arena into the project
7	Modify the raw assets to better suit the game theme

Table 12 backlog of the fifth sprint

10. Sprint 6: Gameplay development

10.1. Sprint backlog

ID	User Story
1	Study the game rules
3	Decide on game mechanics and algorithms
4	Apply the game scenario and game insight
5	Implement character models online + animations
6	Implements the arena into the project
7	Modify the raw assets to better suit the game theme

Table 13 backlog of the sixth sprint

10.2. Diagram

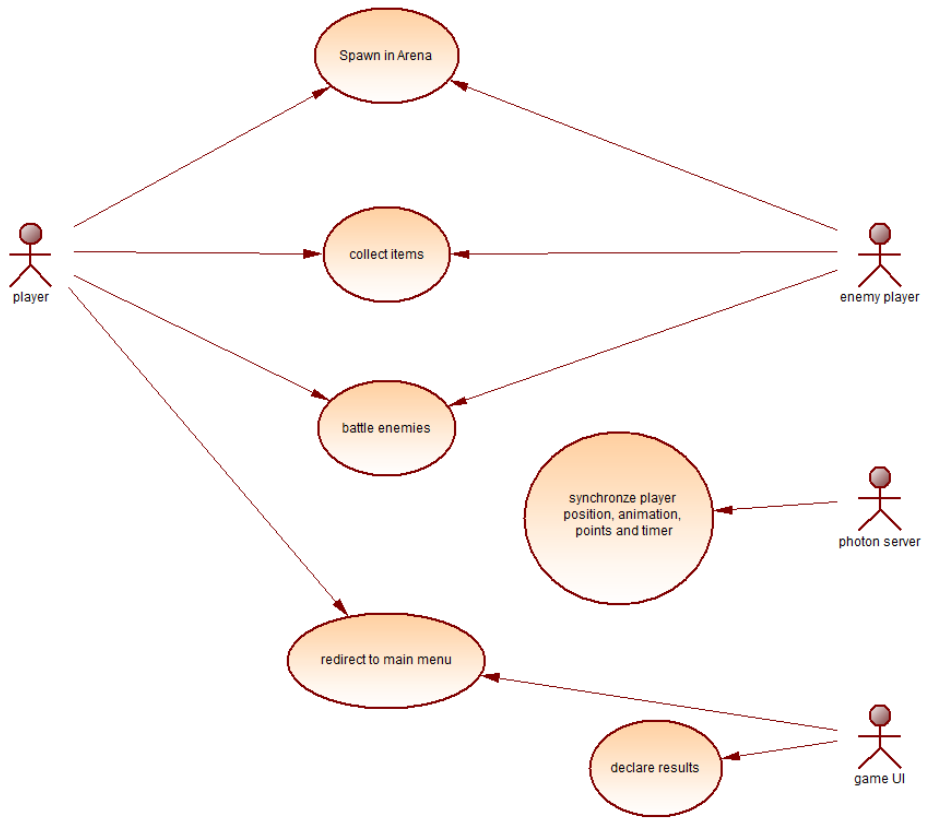

Figure 22 in game use case diagram

Design

1. Introduction

Design is a crucial step in a project's cycle as it describes its objectives visual and conceptual. This part will allow the development of a detailed project and a solid framework to facilitate its execution.

2. Graphic design

2.1. Synopsis

Goal: The first goal of a video game is leisure, then comes enhanced interaction, fantasy fulfilment, and stress relief. Era's end is a game in which each player will be able to participate in matches.

Target customer: Our choice of not using violence in the game removes all age restrictions.

Context of use: It's a game available on computers since the performance of PCs is much higher than that of consoles. They can even be increased further unlike the console.

2.2. Scenario

Objective: Objective of the game Management is simple: Just pick an avatar, wait in the lobby for other players. Once the game starts the first player who gets in the lobby will be selected as the team leader. The leader will divide the points between all the team's members. Then explore the map, search for items and accumulate points either by finding them in the map or by stealing them from the enemy's team to thread the victories as there can be losses. Winning matches allows players to accumulate points to be able to level up. In addition, the player should garner a lot of success to Rank up.

Characters: Each player has to pick one avatar before starting the game.

Place: The game takes place in a foggy environment depicting flooded rural area in a post-apocalyptic/horror style.

2.3. Navigation scheme

The choice was directly based on linear navigation, where the user has no other choice than to browse the application in the established order.

This kind of navigation is very exploited in video games.

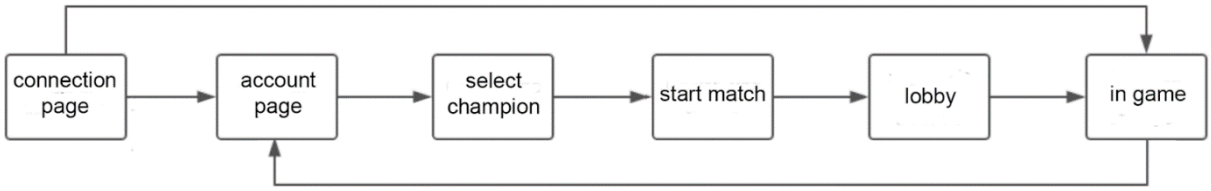

Figure 23 Navigation Scheme

2.4. Graphic charter

First of all, you have to specify the graphic charter because it is the basis of the game.

The main colours chosen for the game are Crimson, Midnight, blue and Turquoise. The crimson refers to this fairly bright pinkish shade of red which carries strong symbolism for both power and danger but also happiness and celebration. The midnight blue is linked with power, strength and intelligence. And for the turquoise it symbolizes compassion, clarity and communication.

The first plan contains the game's interfaces; the login interface and the account's interface. The login's background should be attractive and gives a slight idea of the game's elements such as the champions. For the account's interface the background must be simple and dark so that the buttons would be clearer and the interaction would be easier. That's why the first background contains three champions in the middle, a neon lightning in the back and the game's logo on top. As for the second background it contains a reflected gradient using the midnight blue and plum (a dark red purple) with a light and low opacity checked.

Figure 24 Account background

After several tests, the chosen fonts are strict and simple because they correspond with the intended target through the game.

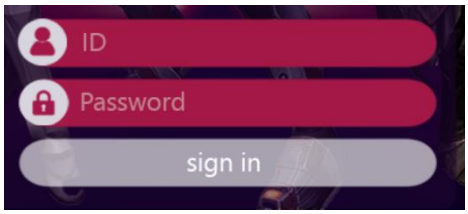

Figure 25 Login Form in-game

Figure 26 Buttons Style in profile

Figure 27 Champion Select Area

Figure 28 Account info Display

2.5. Game's logo

The best type of logo for a game is Wordmarks (or the logotype) that way the name will stick in customers' minds besides the name reflects the game's story.

Therefore, in order to make a logo, we must follow the game's graphic charter. This is why we chose dark red. As for the font, we chose "Fun Game".

Figure 29 "Fun Game" font

the game's interfaces are dark so the logo becomes invisible once you are included in them. For this, the idea was to change the colour.

Figure 30 Game Logo final design

3. Technical design

The class diagrams below (figure 31, 32 and 33) represents the system's behaviour as a set of interactive objects. And then the package diagram (figure 34) regroups the project into one united entity.

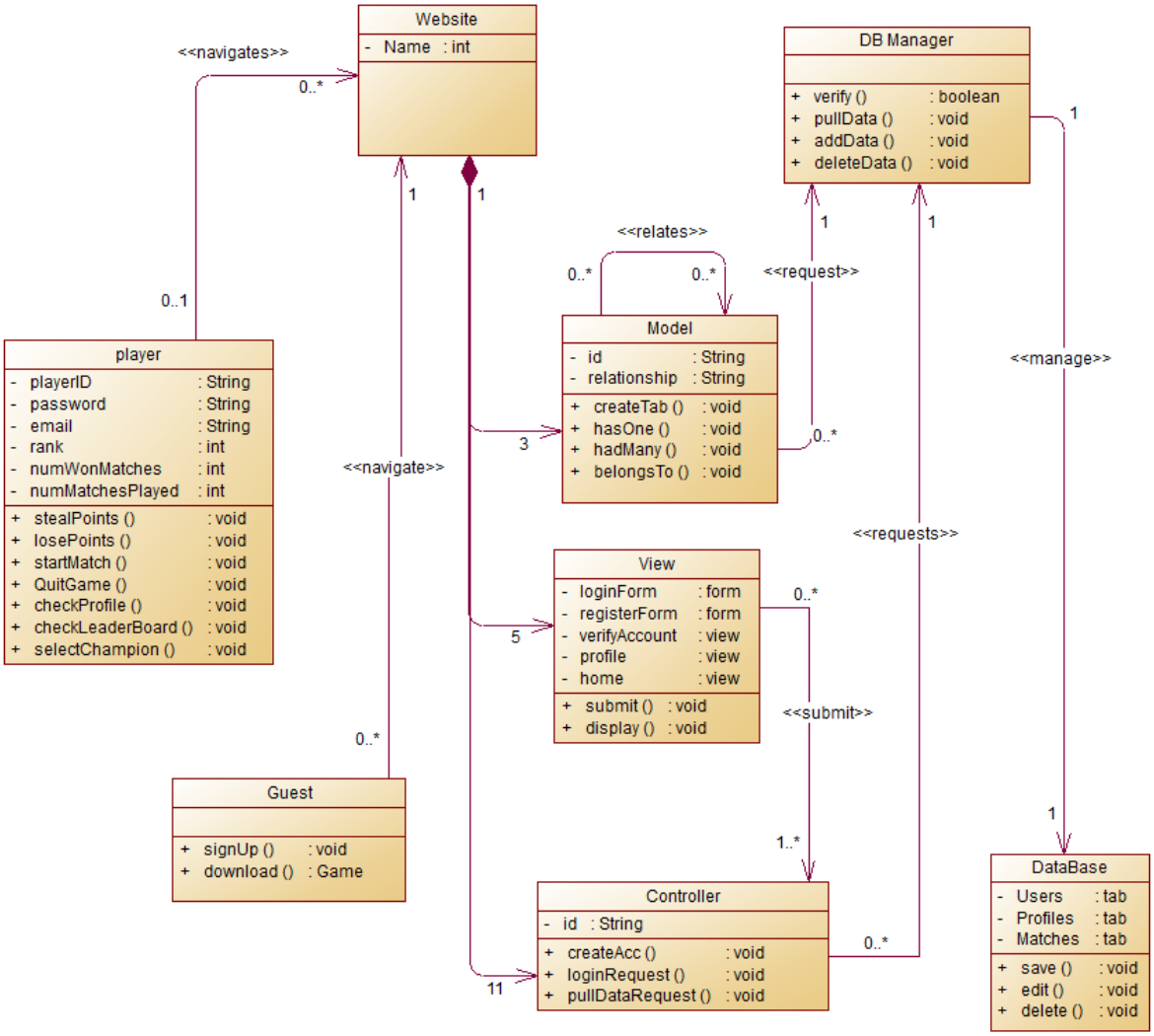

Figure 31 Website & database class diagram

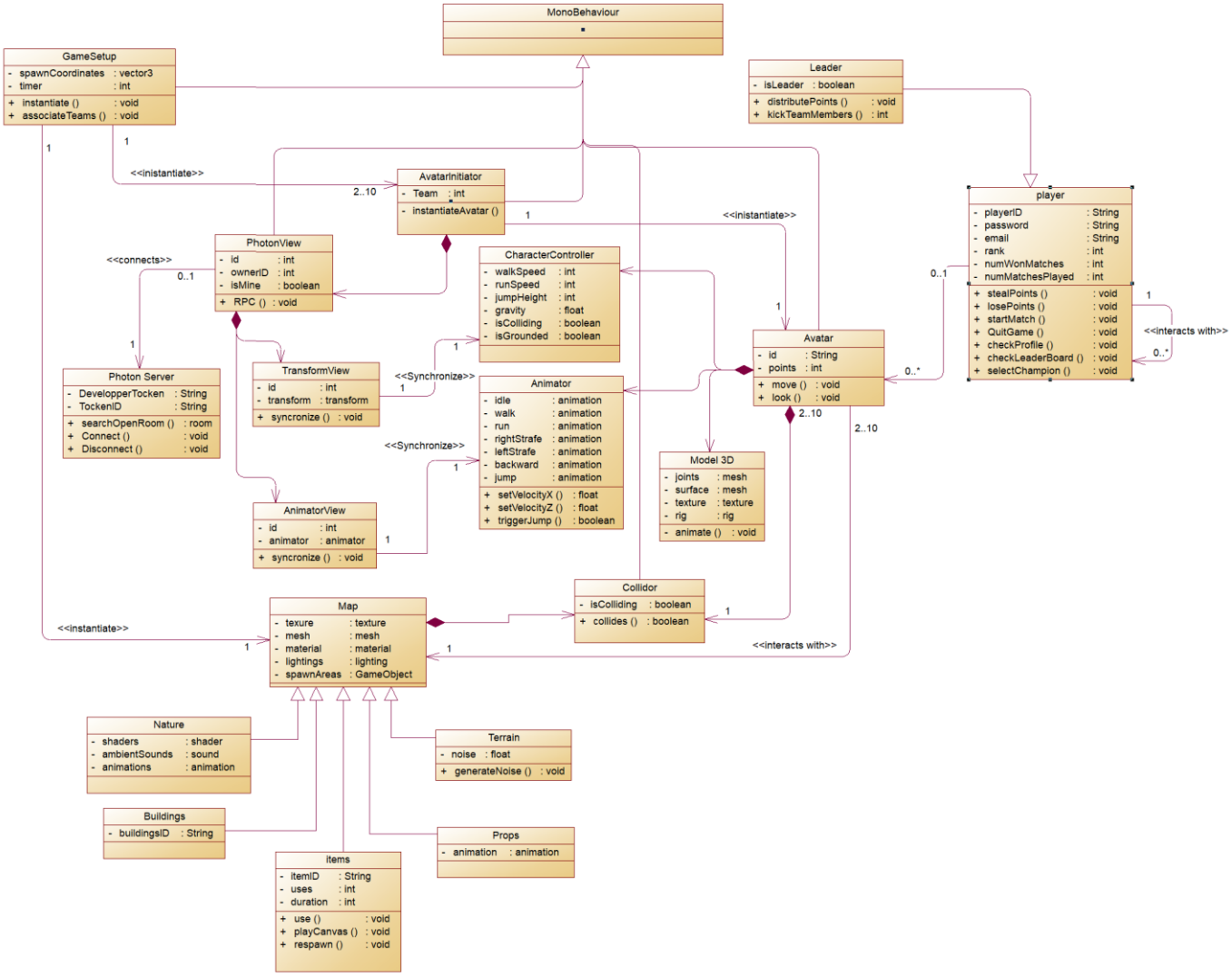

Figure 32 Match Class diagram

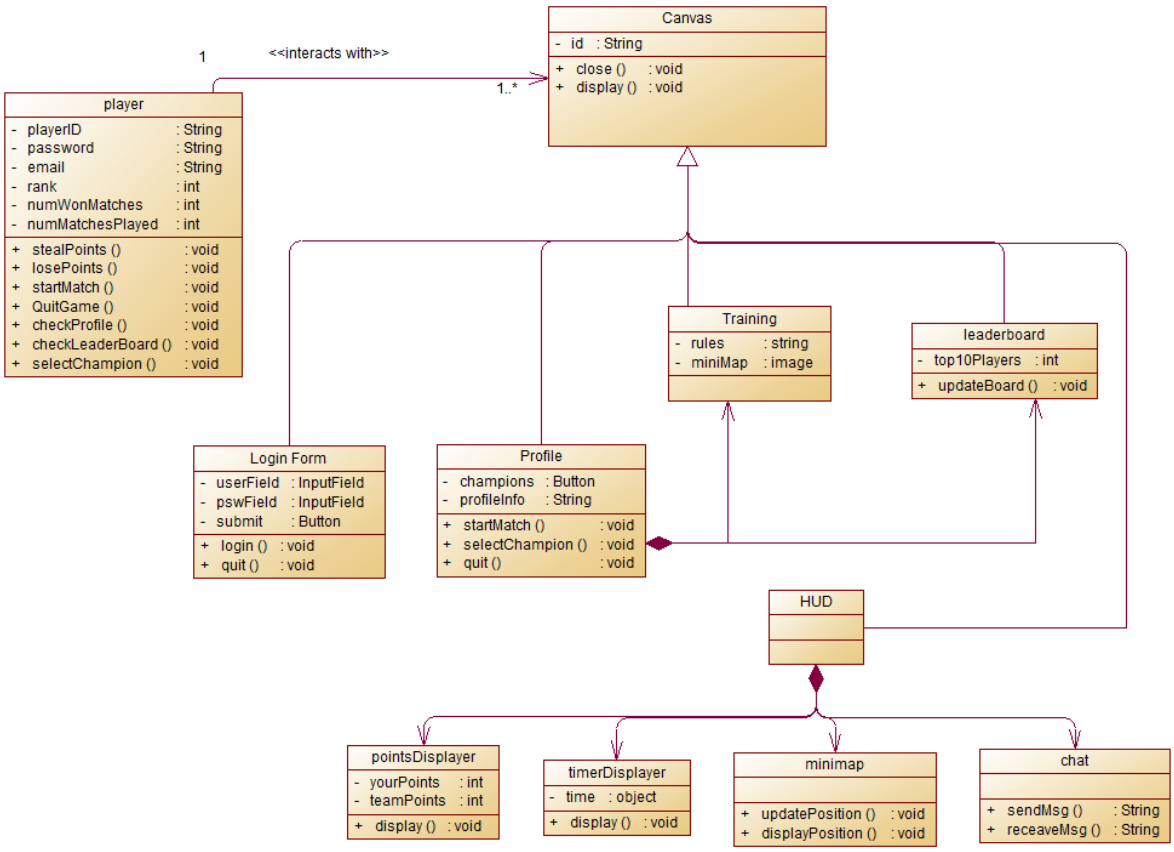

Figure 34 Game UI class diagram

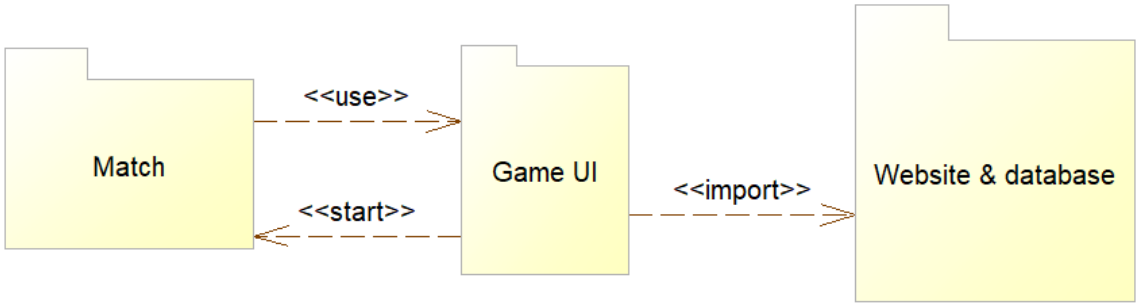

Figure 33 Package diagram

To conclude, following the various parts previously studied, such as the study of the existing one, specification of needs and design, the project is finally ready for the final step which is realization.

Realization

1. Introduction

Before embarking on the realization of the game, it is necessary to take the time to cut out the project in tasks in order to plan their execution and to define the resources to be mobilized.

This chapter lists three main parts: graphics, technique and sound design.

2. Working environment

2.1. Hardware requirements

Operating system	Windows	macOS	Linux
Operating system version	Windows 7 (SP1+) and Windows 10	High Sierra 10.13+	Ubuntu 20.04, Ubuntu 18.04, and CentOS 7
CPU	x86, x64 architecture with SSE2 instruction set support.	x64 architecture with SSE2	x64 architecture with SSE2 instruction set support.
Additional requirements	Hardware vendor officially supported drivers. For development: IL2CPP scripting backend requires Visual Studio 2015 with C++ Tools component or later and Windows 10 SDK.	Apple officially supported drivers. For development: IL2CPP scripting backend requires XCode. Targeting Apple Silicon with IL2CPP scripting backend requires macOS Catalina 10.15.4 and XCode 12.2 or newer.	Gnome desktop environment running on top of X11 windowing system Other configuration and user environment as provided stock with the supported distribution (such as Kernel or Compositor) Nvidia and AMD GPUs using Nvidia official proprietary graphics driver or AMD Mesa graphics driver.

Table 14 System requirements

2.2. Software workspace

2.2.1. Game

- Game engine: unity

The leading platform for creating interactive, real-time content. Unity provides a wide range of services and libraries that create opportunities to work on much more than just games like: Automotive, transportation & manufacturing. Films, 3D animations & cinematics. Architecture, engineering and Construction. Simulations, Virtual reality & augmented reality.

Figure 35 C# logo

Figure 36 Unity Logo

- Script: Assembly-C#

The assembly class Represents an assembly, which is a reusable, versionable, and self-describing building block of a common language runtime application. C# is the main and default scripting language the Unity engine recognises and works along.

2.2.2. Website

- Framework: Laravel 7

Also known as the “PHP Framework for Web Artisans”, is a free, open-source framework and intended for the development of web applications following the model–view–controller architectural pattern and based on Symfony.

Figure 38 Laravel logo

Figure 37 HTML5 & CSS3 logo

- Scripts: Php, HTML 5, CSS 3, Js

PHP is a general-purpose scripting language, especially suitable for Web development. The PHP reference implementation is now produced by the PHP Group.

HTML5 is a mark-up language used to build and present content on the World Wide Web. It is the fifth and last major HTML version recommended by the World Wide Web Consortium. The current specification is called the HTML Living Standard.

Cascading style sheets are a style sheet language used to describe the representation of documents written in a mark-up language (such as HTML). CSS is the cornerstone technology of the World Wide Web and goes hand-in-hand with HTML and JavaScript

JavaScript, usually abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, usually just-in-time compiled, and multi-paradigm. It has curly brace syntax & dynamic typing.

Figure 40 Php Logo

Figure 39 JavaScript Logo

2.2.3. Assets

- Ui design: photoshop

Adobe Photoshop is a raster graphics editor developed and released by Adobe Inc. for Windows and macOS. It was originally created in 1988 by Thomas and John Knoll. Since then, the software has not only become the industry standard for raster graphics editing, but also the industry standard for digital art as a whole.

- 3D modelling: Blender

The Blender Foundation is a non-profit organization responsible for the development of Blender, an open source 3D content-creation program.

Figure 42 Photoshop 2021 Logo

Figure 41 Blender Logo

- 3D animation: Autodesk Maya 2020

Maya 2020 includes new animation, rendering, effects, modelling and rigging improvements, including access to Arnold 6 for CPU and GPU rendering. Based on the powerful animation features introduced in Maya 2019, Maya 2020 adds more than 60 new updates that animators will notice in their daily work

2.2.4. Collaboration workplace: GitHub

GitHub, Inc. is an Internet hosting provider that uses Git for software development and version control. It provides Git's distributed version control and source code management functions, as well as its own features.

Figure 43 Autodesk Maya Logo

Figure 44 GitHub Logo

3. Sprint 1 realization: Website

3.1.1. Website

As we started our game development, our team found the answer to the question “Where are the player going to get our game ? where are they going to make their accounts?” and we decided to create our own website to provide these functionalities.

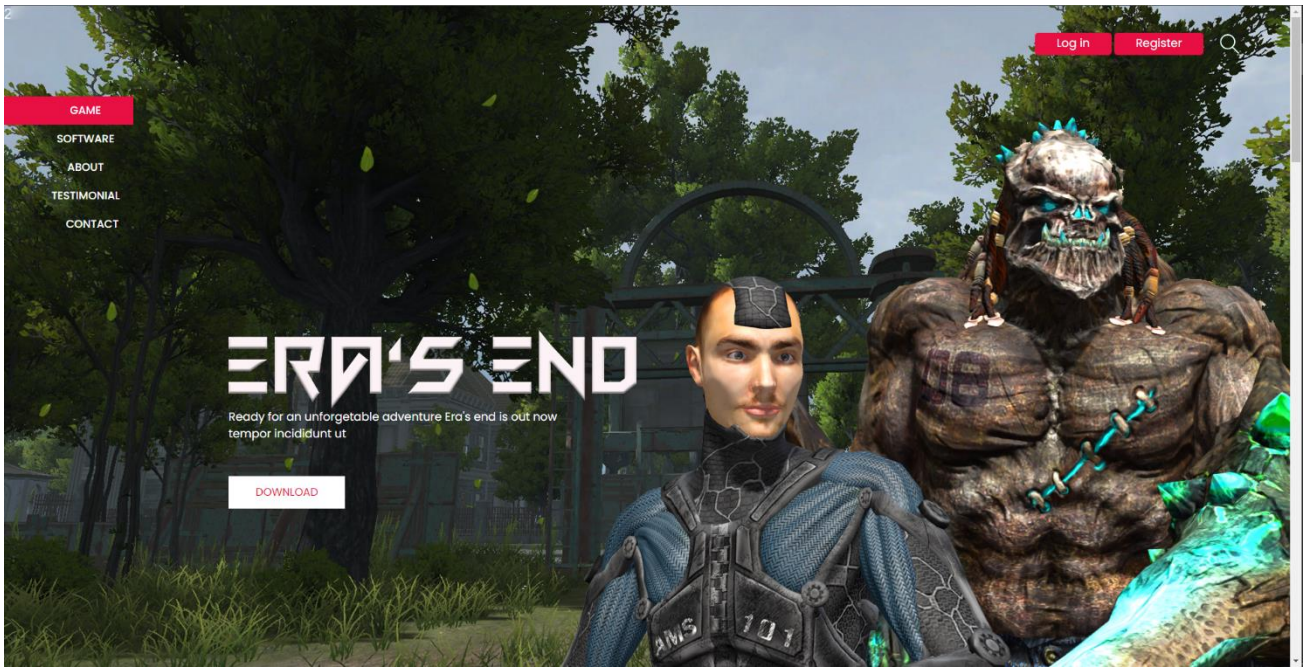

Figure 46 Website Home Page

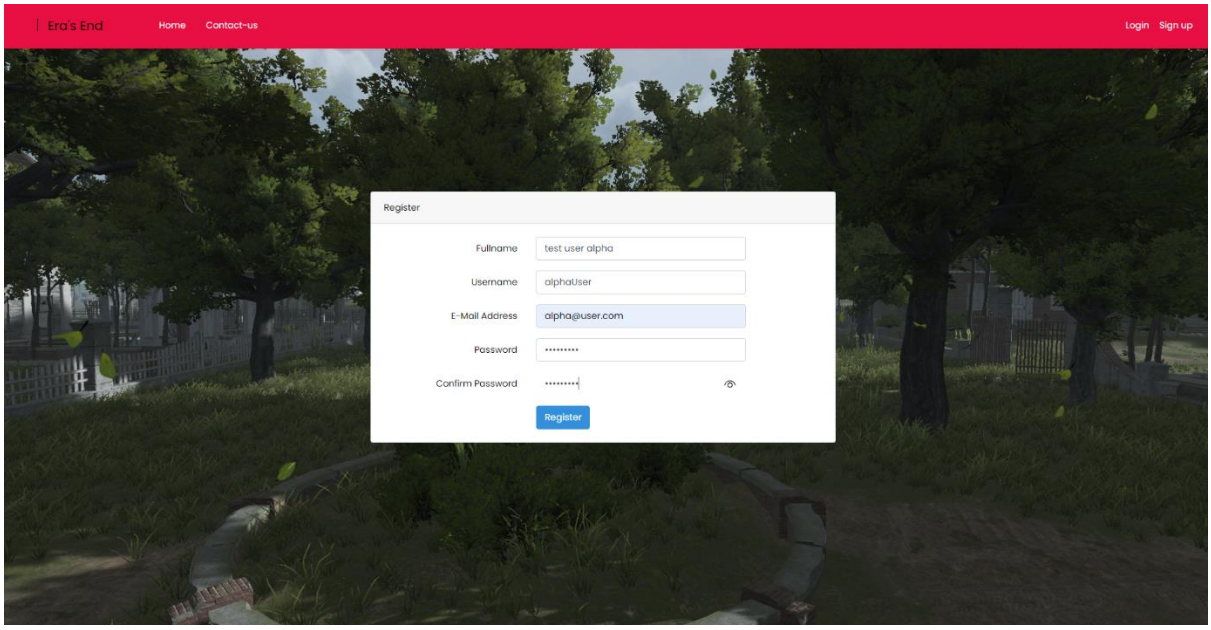

Figure 45 Website Sign up page

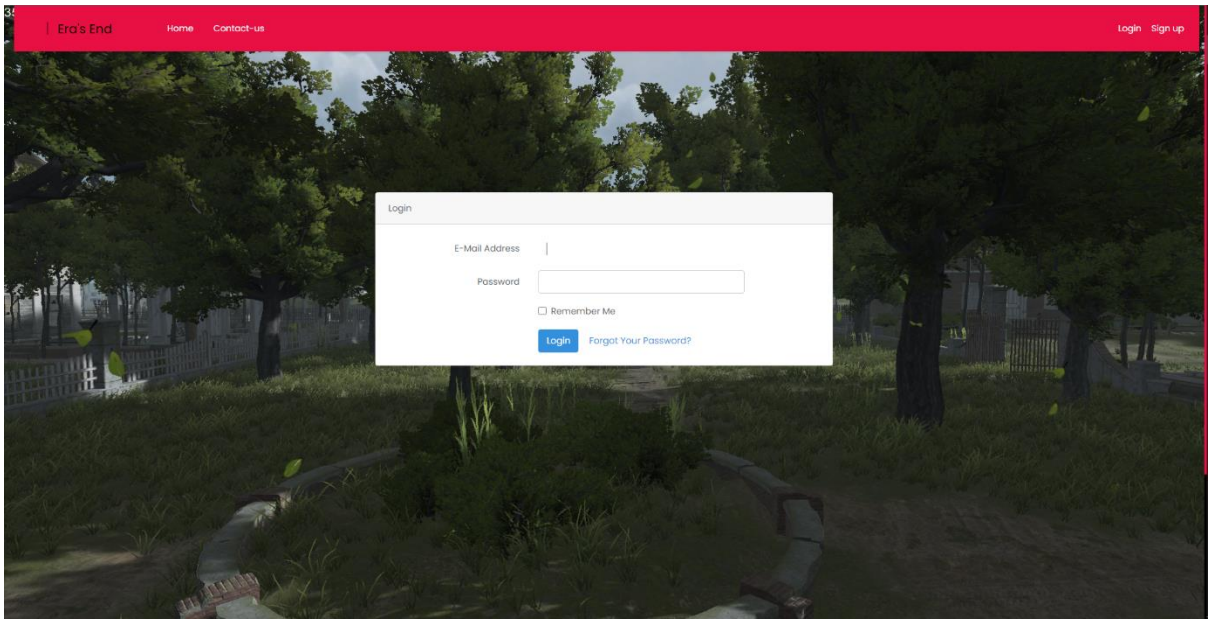

Figure 47 Website Login page

4. Sprint 2 realization: Database

A database is an organized collection of data, generally stored and accessed electronically from a computer system. These database providers online costs a great sum of money monthly which is why we resulted on making out own database locally for the time being as the game is still in its development phase but, our team would eventually switch to a more secure, powerful and stable provider as the game develops.

5. Sprint 3 realization: Game UI

5.1. UI Interface

Once decided on the details mentioned above. It was possible to start the creation of Game interfaces.
The graphics of these interfaces were made entirely with the Game engine: "Unity" using the "UI System" library and the designs were created with “Photoshop”.

For the login menu (figure 45), the idea was to keep the champions in foreground for a 3D effect. As for the account interface it was divided into three main parts; the champion’s selection, player’s information and the menu.

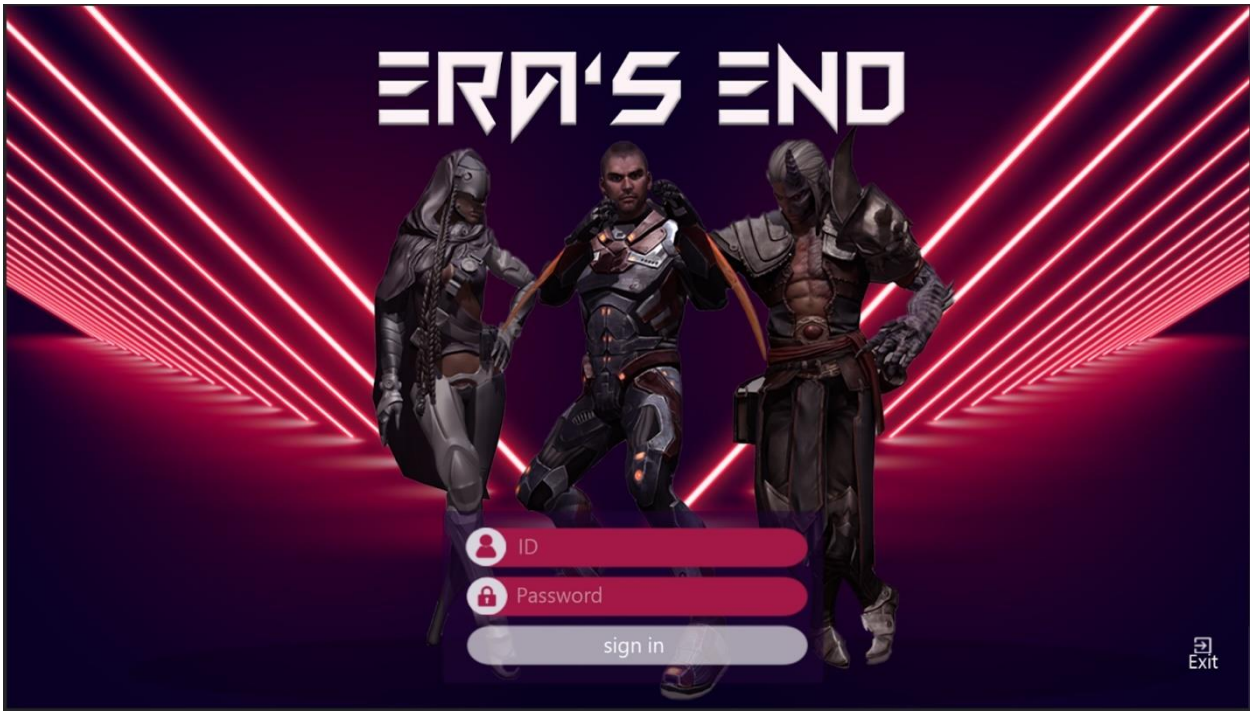

Figure 48 in-game login form: final design

The login interface also has a button to quit the game.

Once the login menu is passed, the user is in the account interface. This interface is divided into three main parts as mentioned

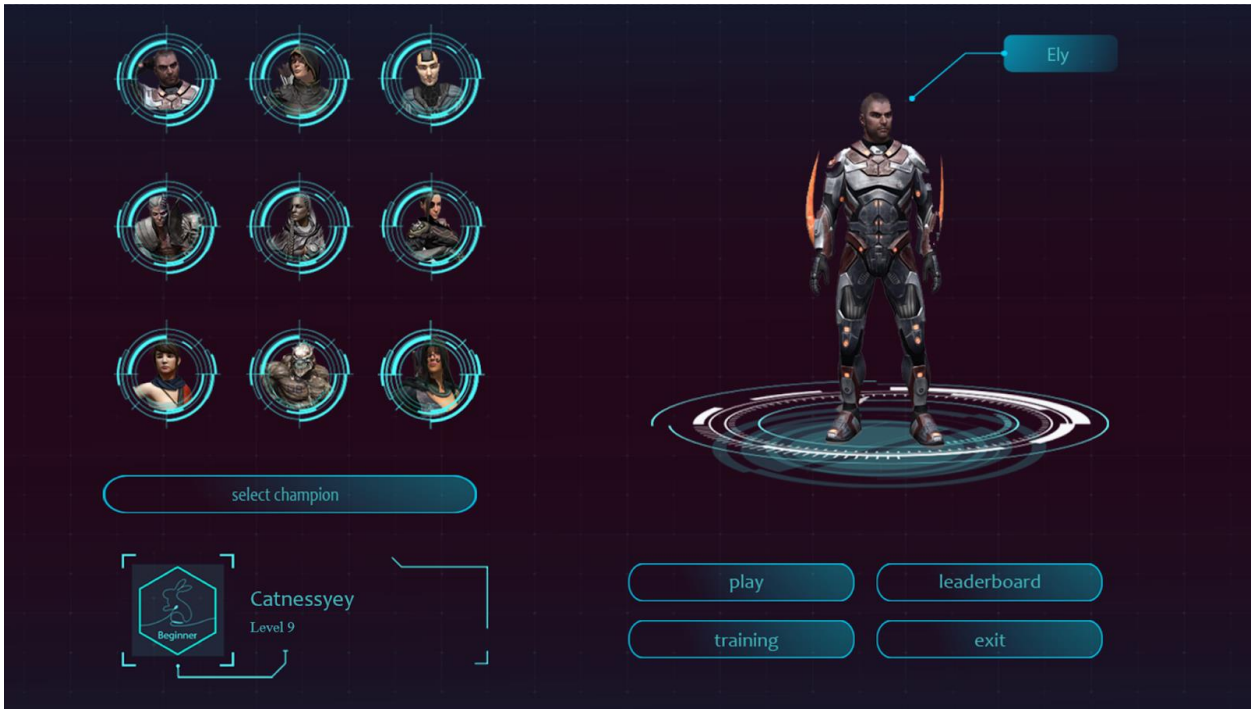

Figure 49 in-game Start menu: final design

- champion's selection :

All the 9 champions are presented in circular frames. Once the player clicks on any of them the champion will fully appear on the right. After picking the champion that he likes to play with, the player has to click on the "select champion" button.

- player's information :

in this part the player will find his Id, his level in numbers and his rank. There are 10 ranks which are related to the player's level.

level 1-10 : Beginner
level 11-25 : Apprentice
level 26-35 : Skilled
level 36-45 : Professional
level 46-55: Artisan
level 56-65 : Master
level 66-75 : Grandmaster
level 76-85 : Epic
level 86-100 : Legend
level 100+: Mythic

Figure 50 Ranks Logo designs

menu : In this part there are 4 buttons :

- Exit button
- Leader board when clicking on this button a board will pop up to show the 10 top players and the rank of the player himself.
- Training helps the player to know more about the game. It'll give him instructions on how to play and the rules of the game.
- Play button is active right after picking the champion, clicking on it will lead the player to the lobby where he'll wait for other players to start the match.

Figure 51 Leader Board: final design

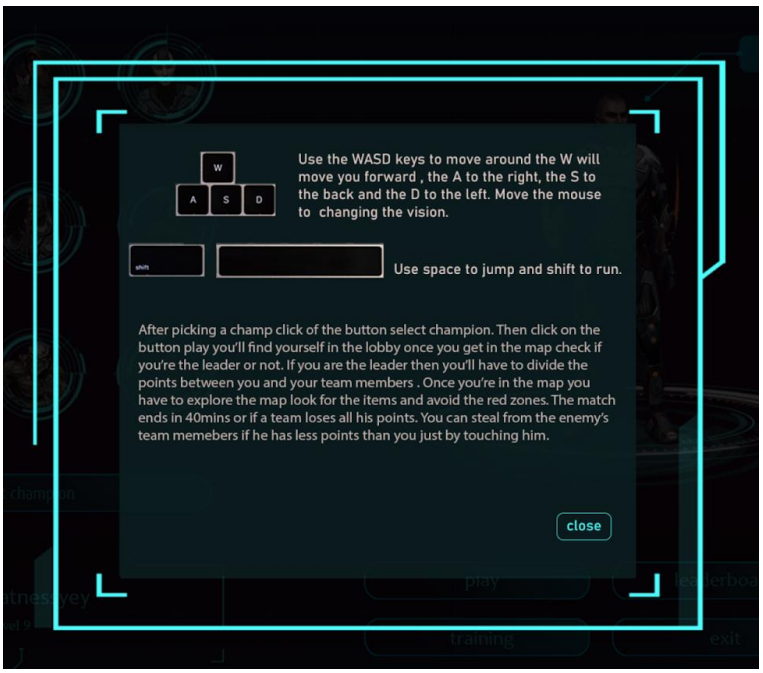

Figure 52 Game Rules Explanation

Figure 53 Match Room interface: final design

6. Sprint 4 realization: Networking

Server: Photon Unity Networking 2

6.1. introduction

Photon is a fast, streamlined and flexible real-time multiplayer game development framework. Photon consists of a server and multiple client SDKs for the main platform.

6.2. benefits

Photon Unity Network (PUN) is the basis of our advanced Unity-specific solutions: pairing, easy-to-use call-backs, components for synchronizing Game Objects, remote procedure calls (RPC) and similar functions provide a good start. In addition, it also provides a solid and extended API for more advanced control.

The complete source code is provided, so you can extend this package to support any type of multiplayer game you come up with.

The package is compatible with the hosted Photon Cloud service, which can run the Photon server for you. The setup window will register you in less than a minute (free of charge).

6.3. Most notable features

- Easy to use API
- Lots of demos and an extensive PUN Basics Tutorial
- Server available as hosted service (free for development) or as "On Premise"
- Load-balanced! Scales across servers (with no extra effort)
- Outstanding performance of the Photon Server
- Dedicated servers. No NAT punch-through needed
- Offline mode: reuse your multiplayer code in single player game modes

6.4. Development

6.4.1. Networking

There are mainly two possible network architectures: peer-to-peer and client-server. In the peer-to-peer architecture, data is exchanged between any pair of connected players while in the client-server architecture, data is only exchanged between players and the server.

We chose the latter as it provides a more stable and predictable service for the players and the photon servers helps greatly in guaranteeing these benefits.

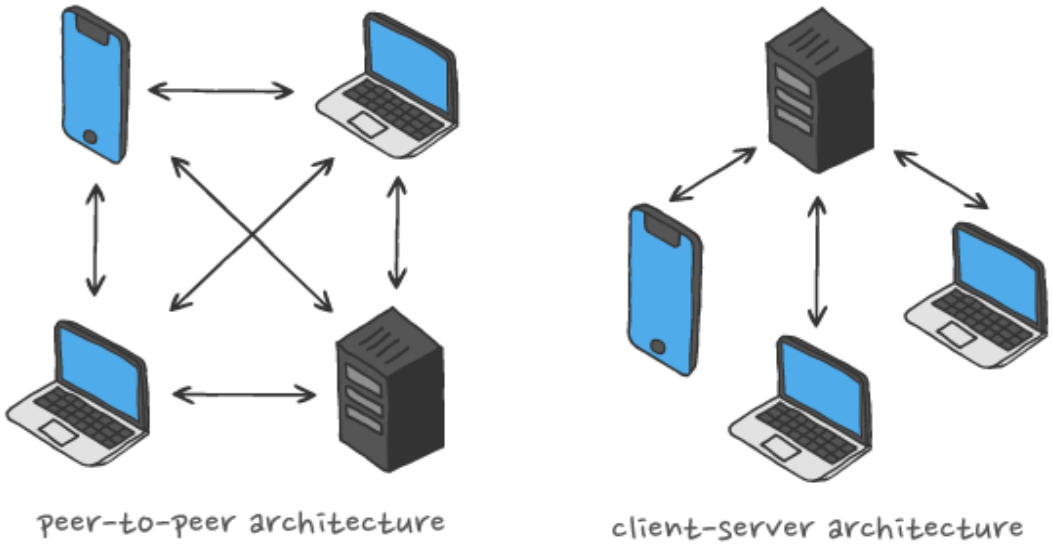

Figure 54 Networking architectures

7. Sprint 5 realization: Design & Assets preparation

7.1. Modelling

One of the most important parts of the game that attracts the player is the game graphics (the map and the champions). Unfortunately creating superb assets will take at least 6 months and since we had only three months we had to search for some online assets that fit the theme of our game.

7.1.1. Map modelling

We downloaded all the map's assets from the Unity store. Our map will consist of three categories:

- Buildings

villas, cabins, barns, lighthouse, guardhouse, greenhouse, churches and structures.

Figure 55 Arena Buildings samples

- Nature:

trees, grass, rocks and bushes.

Figure 56 Arena's Nature samples

- Props:

beds, cabins, chairs, lamps, tables, sofas, vases, paints, boats, clocks, cars and gravestones.

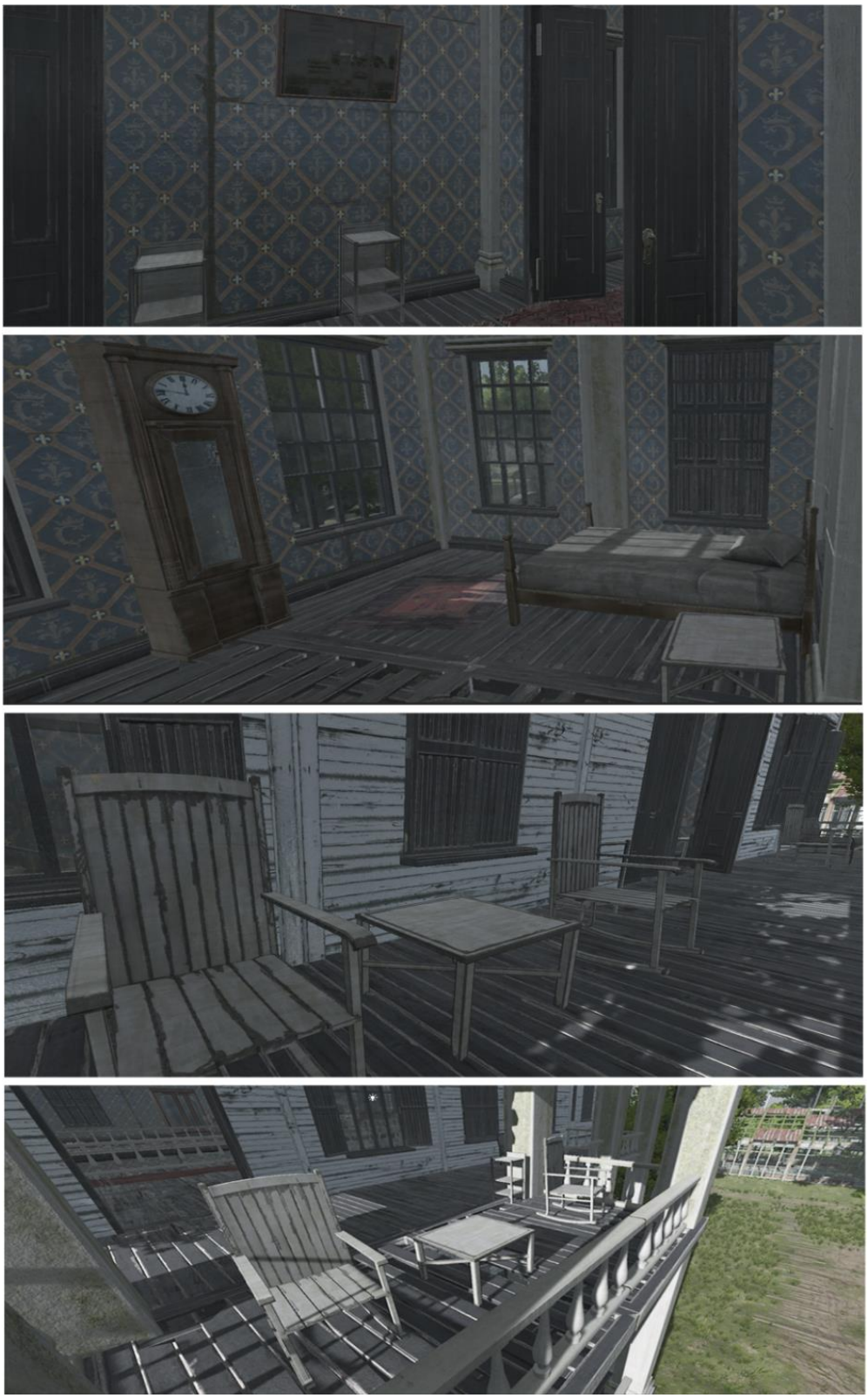

Figure 57 Arena's Props samples

- Champions modelling:

For now, we have only 9 champions available. The champions have various looks so that each one can have his own story. We have 5 female champions and 4 male champions. They're all downloaded from Maximo.

Figure 58 Champions 3D Models

- Items modelling:

We have 3 items that will be randomly located in the map. Acquiring these items as a secondary objective provides beneficial bonuses such as: **extra points**, **shield** and **vision.**

The vision will be presented as a book that contains all the player's points at that moment, the shield will be presented as a potion and finally the extra points it will be presented as coins.

Figure 59 first item 3D model

Figure 61 second item 3D model

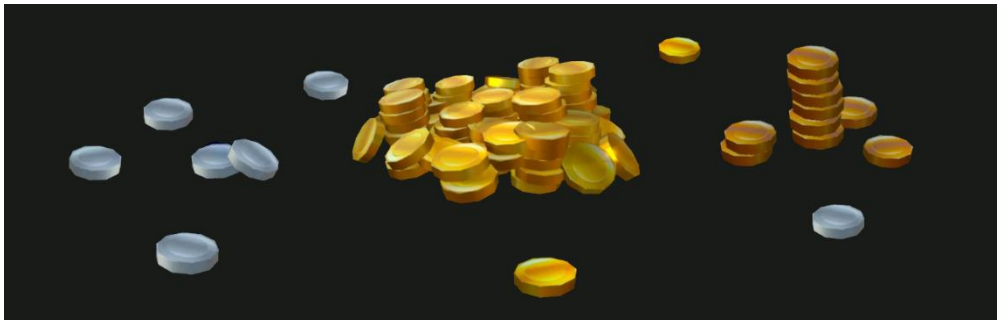

Figure 60 third item 3D model

7.2. Animation

The last step, in the realization of a 3D model, consists in “boning” and “rigging” the character, that is to say creating an animation skeleton placed in the modulization and adding controllers on this skeleton to animate it like a virtual puppet. In the Era’s end game, a champion requires 18 animations such as: **Walking, Running, Crouching, Jumping, left strafe, right strafe …** etc**.**

7.3. Integration

To be able to integrate 3D models into "Unity", it is necessary to convert them to "FBX19" format.
This integration takes place in several stages:

- Import 3D models into "Unity"
- Import textures from 3D models.
- Import animations from “Mixamo”
- Select "Humanoid" as the type of animation in the "Rig" section

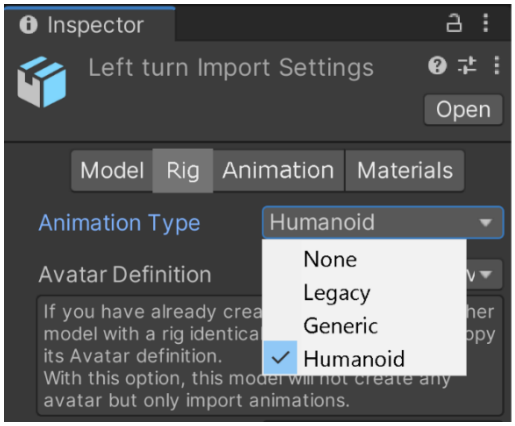

Figure 62 Unity's Model import settings

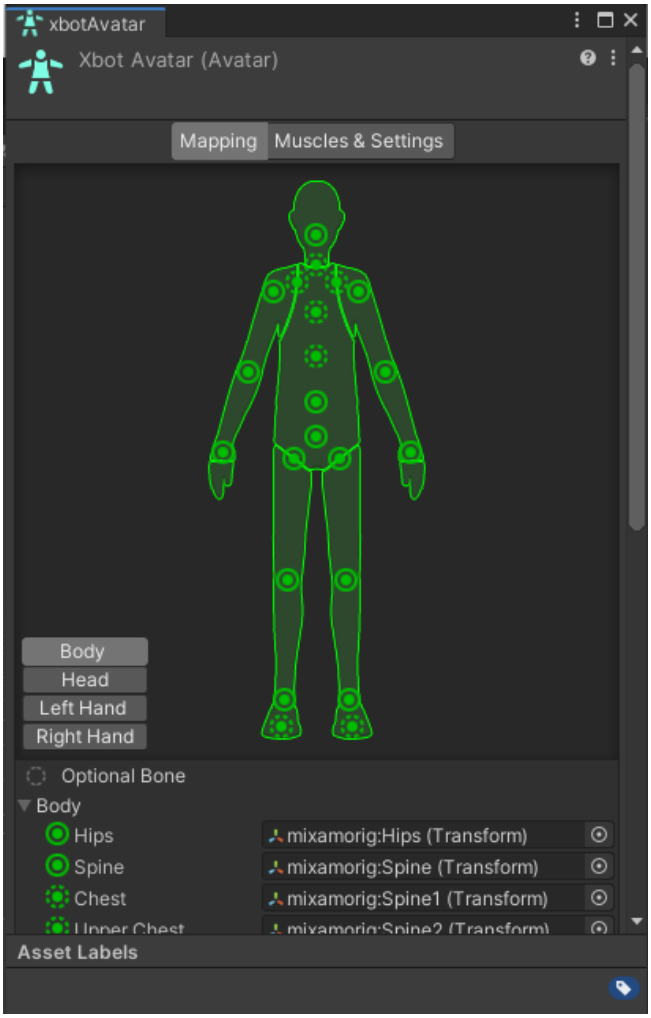

Figure 63 Unity's Model rigging system

this procedure allows Unity to consider the character models as “Humanoids” which is necessary as we are dealing with only humanoid characters and a lot of animations for these characters.

7.4. sound design

Music is the art of compiling sounds, it reinforces the visual feeling in the player, thus it allows him to dive into the world of the game. All the sounds used in this project are of the “mp3” type. The Era's end game includes background music and sound effects.

The choice of background music for the lobby fell on "Twisted Treeline" because it combines darkness, calmness and mystery that we have found adequate with this game. For the sound effects there are leaf rustle sound, water sound and wind sound effect.

8. Sprint 6 realization: Gameplay Scenario

As in all games in history, the gameplay is most important and it is what leaves the biggest impression on the new players and old veterans alike. And due to our team's inexperience in the field, we decided to take the matter slowly to not ruin the game for the players. The Gameplay was simple and clear, the rules are little: if you're caught against an opponent with more points, you lose points. If you find a special item on the map you get a bonus depending on the item. If you drain all the enemy team's points it is your team's victory. That or the time limit is reached and your team have the most points.

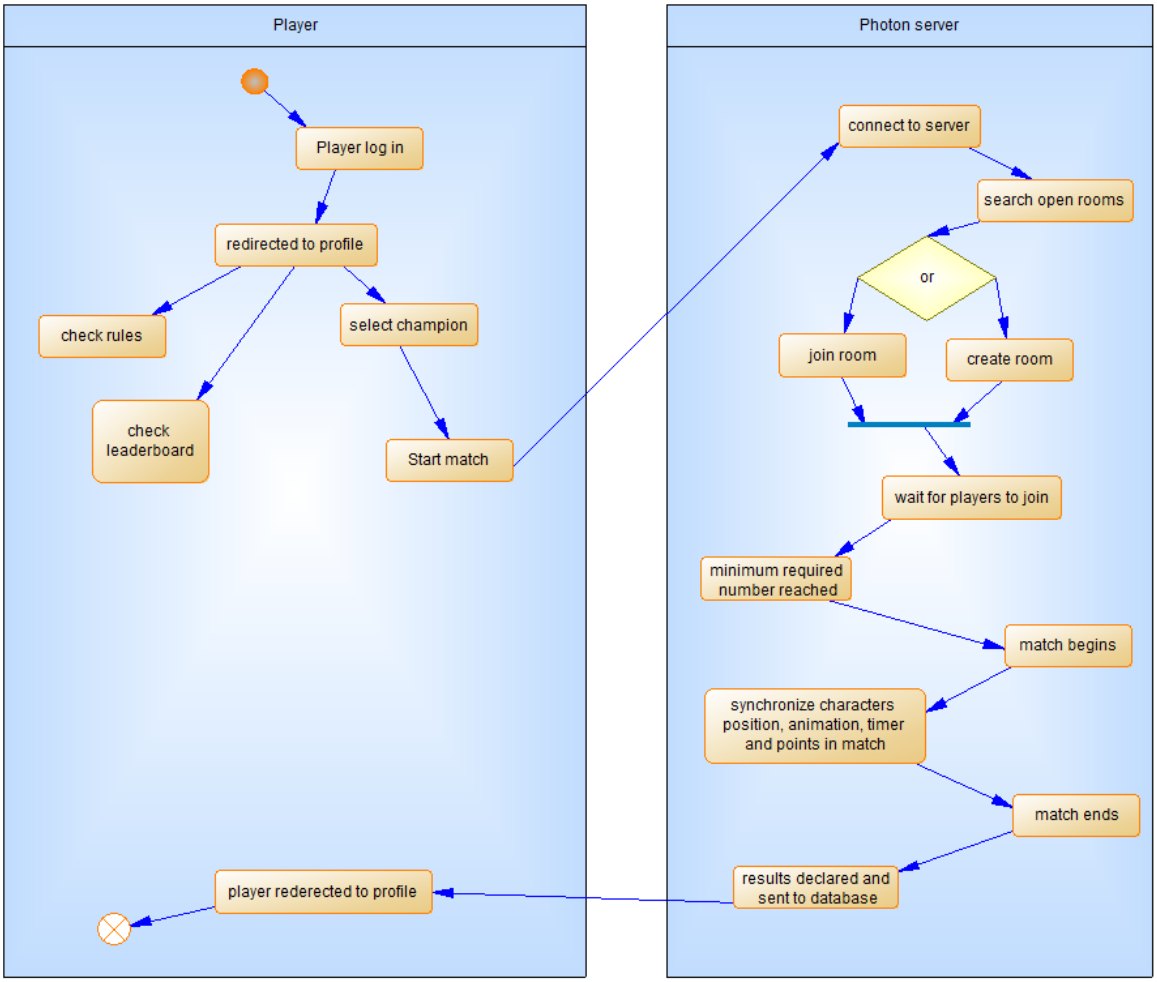

Figure 64 Game Scenario Activity diagram

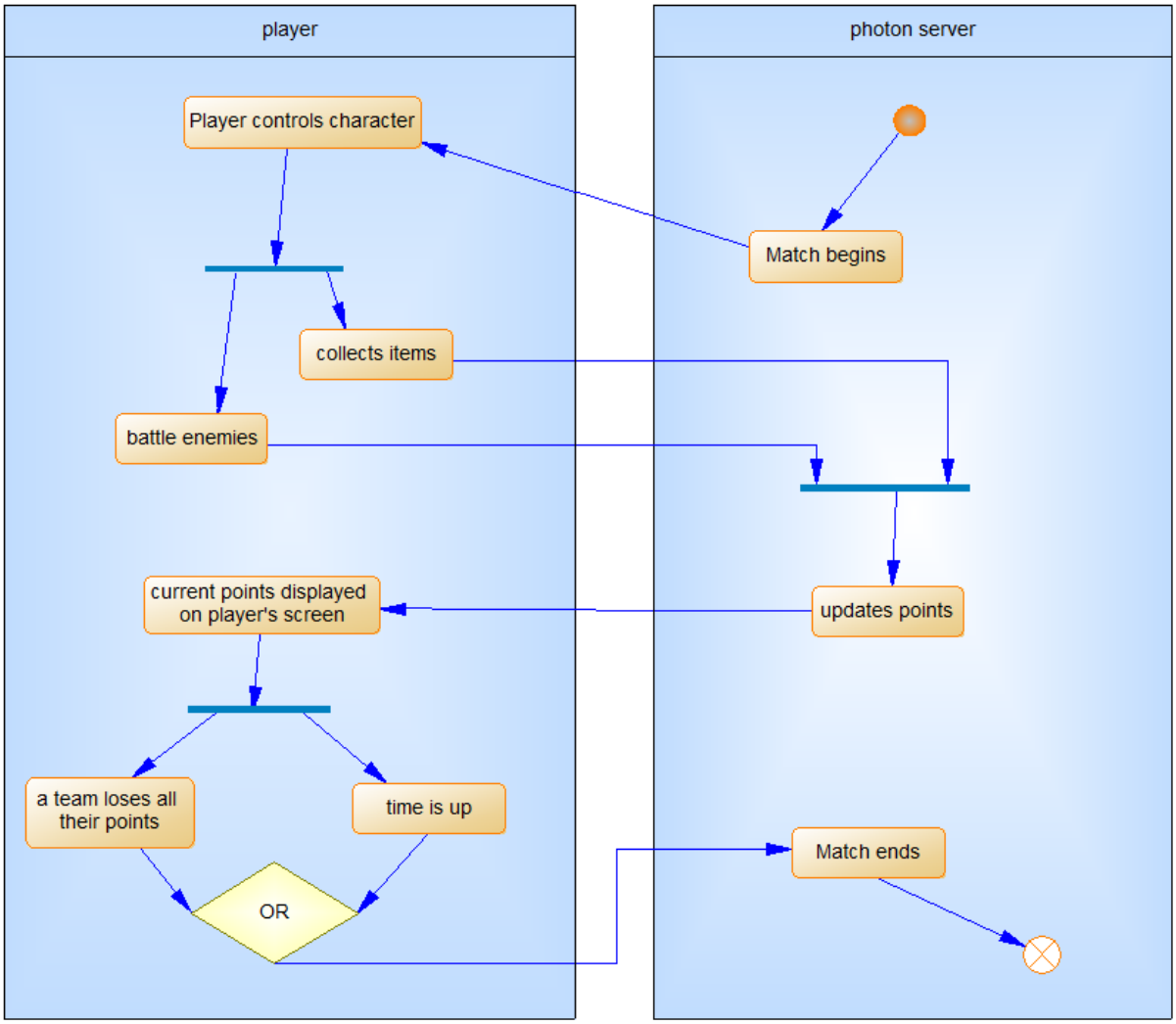

Figure 65 Era's End Game Insight Activity diagram

And we can conclude the game play development with samples and screenshots from the actual game as follows.

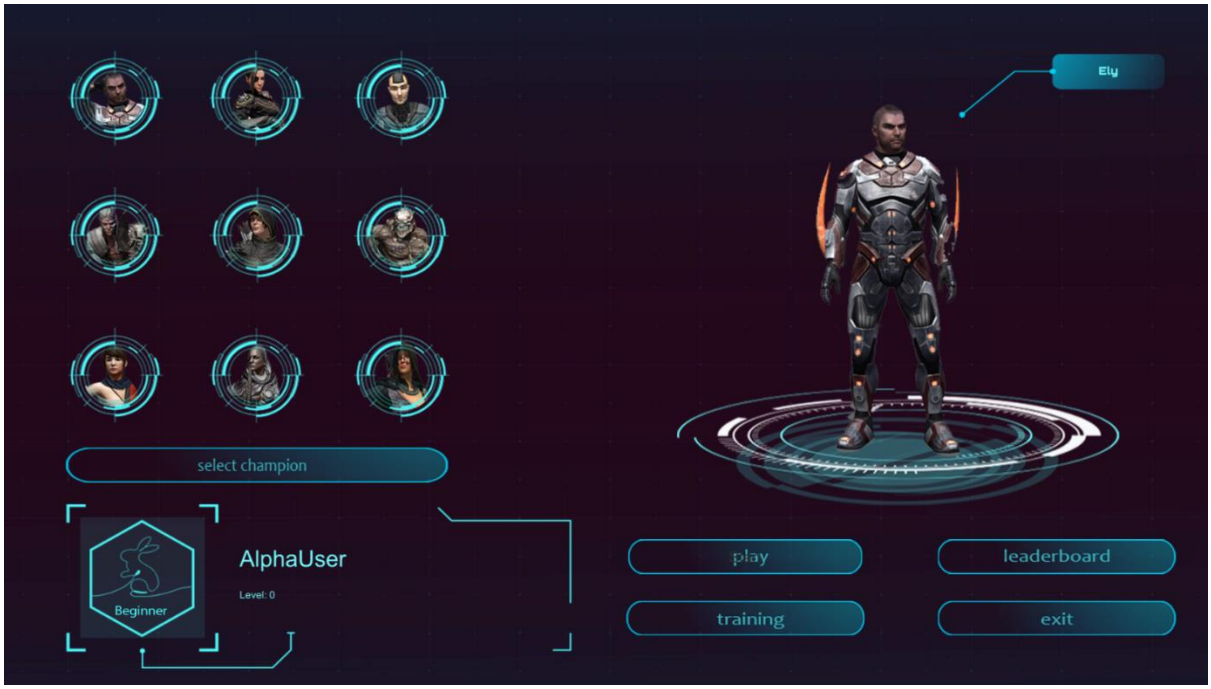

Figure 67 Main menu Interface

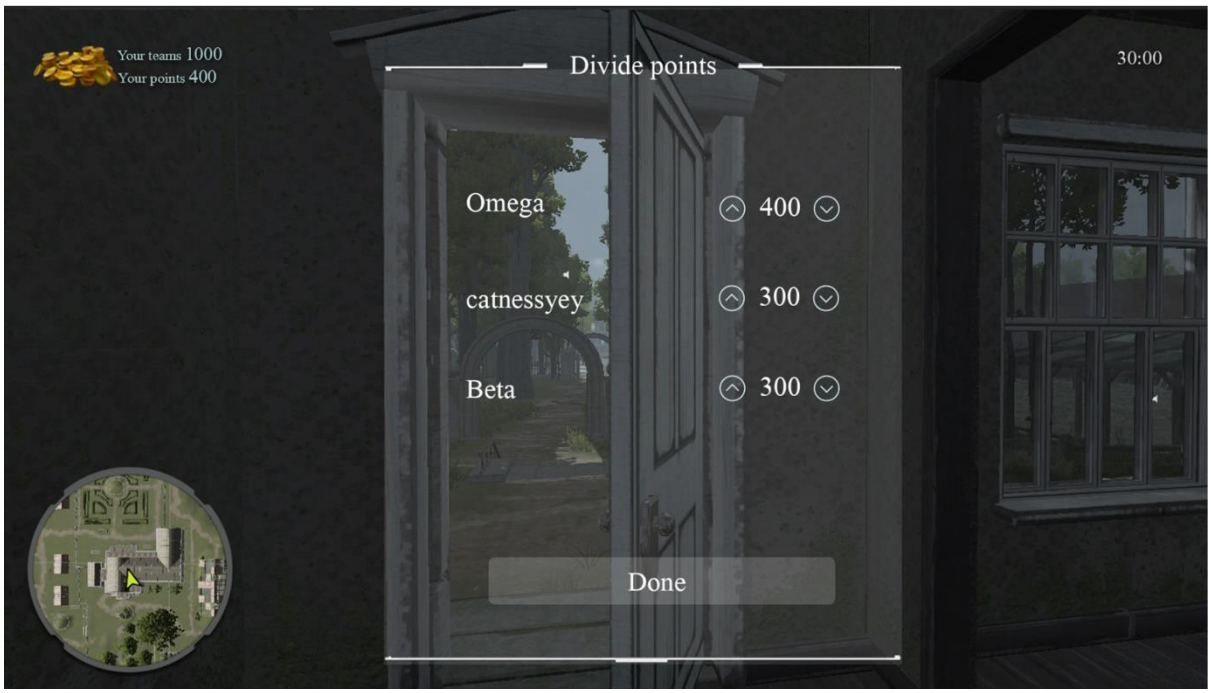

Figure 66 in-game Team Leader interface

Figure 68 Potion item interaction (10 seconds shield)

Figure 69 Interaction with an enemy player

Figure 70 Book interaction (reads enemy status)

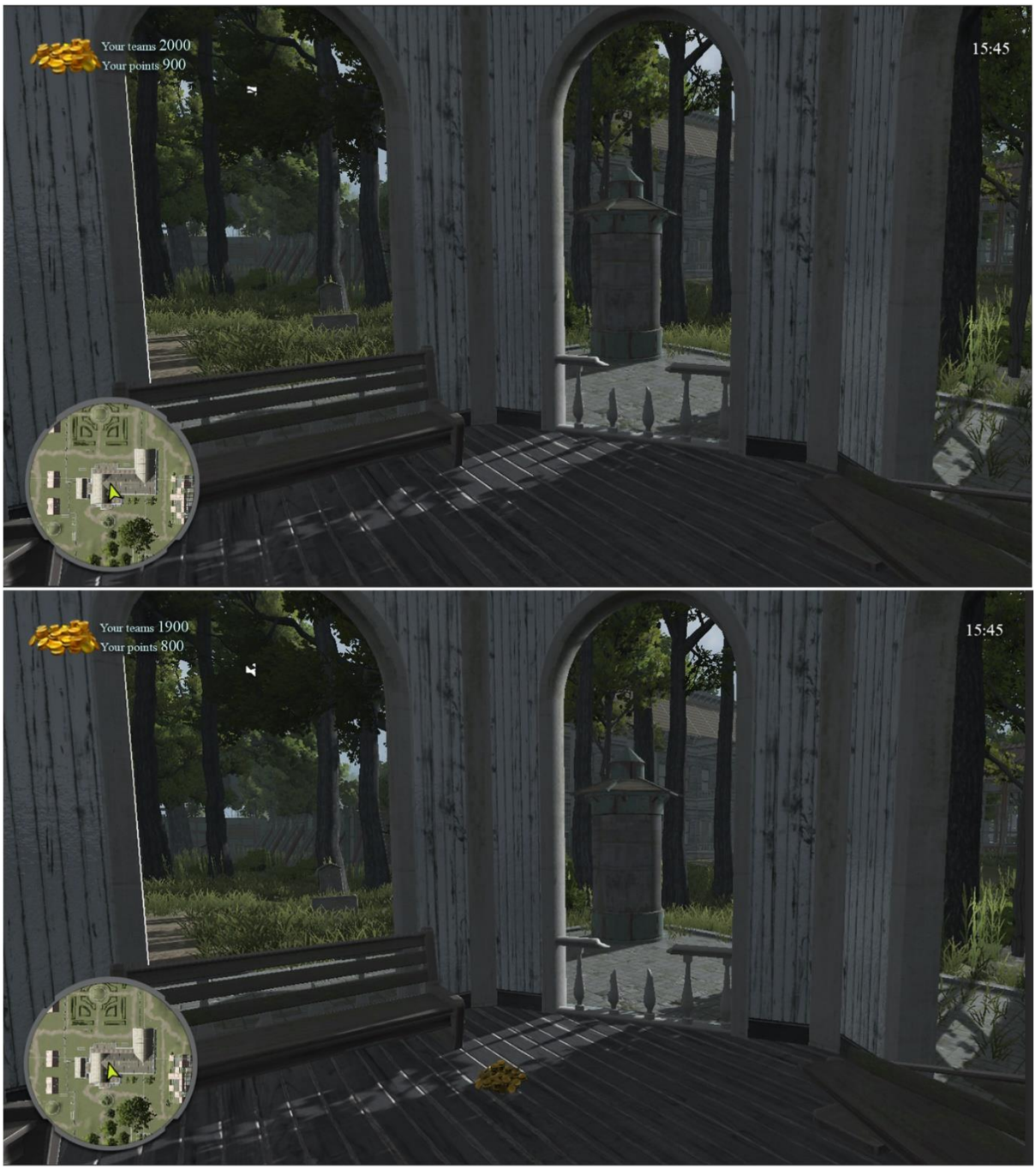

Figure 71 Coins interaction (add bonus points)

Figure 72 Results Declaration & Game's over

Conclusion and perspective

•

9. Conclusion

This project was an important experience during which we were able to develop several skills and learn new techniques. Managing a video game project is not an easy task because it requires a lot of checking, research and testing to achieve an end result of acceptable quality.

Completing this project as part of a company was the other aspect of professional initiation and preparation for managing company expectations and user needs.

the next phase of developing Era's End further is the closed Alpha Phase: where limited samples of the game is distributed around a number of trusted players to get more feedbacks from different vision.

Era's End have very little that it misses before the distribution point and our team is proud of what it has achieved so far and that is why it is taking it beyond the educational limit.

1. Perspectives

This work has reinforced our motivation to work in the field of video games. It also instilled our spirit of sharing and group.
Thus, we wish to add:

- In-game chat with team mates + voice chat
- More game modes
- More Arenas
- Champion's special abilities
- Background stories for the champions making them more "alive"

References

2. Netography

[1] Unity Group. (2005, June 1). Unity documentation. Unity.Com. https://docs.unity3d.com/Manual/index.html

[2] Photon Engine. (2021, March 19). PUN 2. https://doc.photonengine.com/en-us/pun/current/getting-started/pun-intro

[3] Mixamo. (2015, June 1). Adobe Mixamo. https://www.mixamo.com/

[4] Laravel - The PHP Framework for Web Artisans. (2020, November 25). Laravel 5. https://laravel.com/docs/8.x/installation

[5] PHP: Documentation. (2020, June 3). Php Stable 8. https://www.php.net/docs.php

[6] DevDocs — HTML documentation. (2008, January 22). HTML 5 Docs. https://devdocs.io/html/

[7] JSDoc. (2020, June 1). JSDocs. https://jsdoc.app/about-getting-started.html

[8] SDK Docs - Valve Developer Community. (2021, May 21). Valve SDKs. https://developer.valvesoftware.com/wiki/SDK_Docs

[9] Respawn Entertainment. (2019, February 4). Respawn Entertainment SDKs. https://www.respawn.com/studio

[10] About Riot Forge. (n.d.). Riot Forge. Retrieved June 7, 2021, from https://riotforgegames.com/

[11] V. (n.d.). The Many Different Types of Video Games & Their Subgenres. ID Tech. Retrieved June 7, 2021, from https://www.idtech.com/blog/different-types-of-video-game-genres

[12] Wikipedia contributors. (2021a, June 6). Single-player video game. Wikipedia. https://en.wikipedia.org/wiki/Single-player_video_game

3. Bibliography

- Kremers, R. (2009). *Level Design: Concept, theory, and practice*. CRC Press.

INSTITUT SUPÉRIEUR DES SCIENCES APPLIQUÉES
ET DE TECHNOLOGIE DE SOUSSE