BCSH - Project

Evandro Gomez Quintino

2960774

Submitted in partial fulfillment for the degree of

B.Sc. in Computing Science

Griffith College Dublin
06, 2021

Under the supervision of Joe Roe

Disclaimer

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the Degree of Bachelor of Science in Computing at Griffith College Dublin, is entirely my own work and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

6/14/2021

X Evandro Gomez Quintino

Signed by: Evandro

Figure 1. 1 - The Lord of the Rings - Tactics

I dedicate this project to my wife, who has always supported my dream to become a game designer and who has been beside me during all the difficulties along the way. And to myself, that beside all the odds like age or starting a new career in a different country, I managed to succeed all the barriers in order to become something that I only dreamed when I was younger.

The theme of this project was based on the "*The Lord of the Rings*" trilogy and turn based games that I have played during my life, including also chess, one of the first board games that I had learned how to play.

I hope that everyone playing this game can enjoy it and hopefully in a future soon, we will have more games made by myself in the marketing. Thanks again!

Evandro Gomez Quintino

Contents

List of Figures

List of Tables

List of Code Snippets

Chapter 1. Game Design

The following Chapter will introduce to my game industry research and cover game design ideas that I had when starting the project. Doesn't reflect all the results achieved by the end of the same.

Game Industry

Before we dig into the project details and design, let's have a quick overlook in the game industry.

Couple of years ago, actually a long time ago, I told my father that I would like to work in the game industry. It was 2006 and I didn't have any idea how to do it or even where to start, but motived by my personal hobby, I told myself that would be a great idea to work closely with something that I always loved.

My father, like anyone who grew up back in the 60s, had this image that games were made for children and the profits from that were minimal. In fact, he was actually right in one aspect, the game industry by that time was small and limited by the technology so for me would be really difficulty to give my first steps in this career.

Fortunately, we are now in 2021 and technology has grown in different ways that we couldn't even imagine back in 2006 and so has the game industry. As per this article in the GameIndustry.biz (Batchelor, 2020), the game industry has registered in 2020 a market worth of $174.9bn , which comparing against other cultural industries like cinema which made $100bn back in 2019 (Escandon, 2020) before the Covid-19 events, it's an expressive difference and allow us to predict what the industry reserves for the next years.

Also motivated by the f2p (free to play) game genre, which basically only requires the device and an internet connection, this genre added more profit to the industry by allowing players to do in-game purchases or microtransactions. This way, instead of receiving profit from game sales, the studios are now constantly adding more content in-game at a low cost, which guarantees their economy. As this article from Forbes suggests (Gardner, 2020) , the microtransactions will allow the game

industry to rise to 30% for the next three years, even with a declining purchase revenue of games, but motivated also by the Covid-19.

The following image allow us to visualize the whole picture of the game industry in the last 50 years in different platforms and genres, showing that it's just a matter of time for the same to become even greater and that new technologies like Cloud and VR, which have been recently added to the market, will soon bring more profits also to the industry.

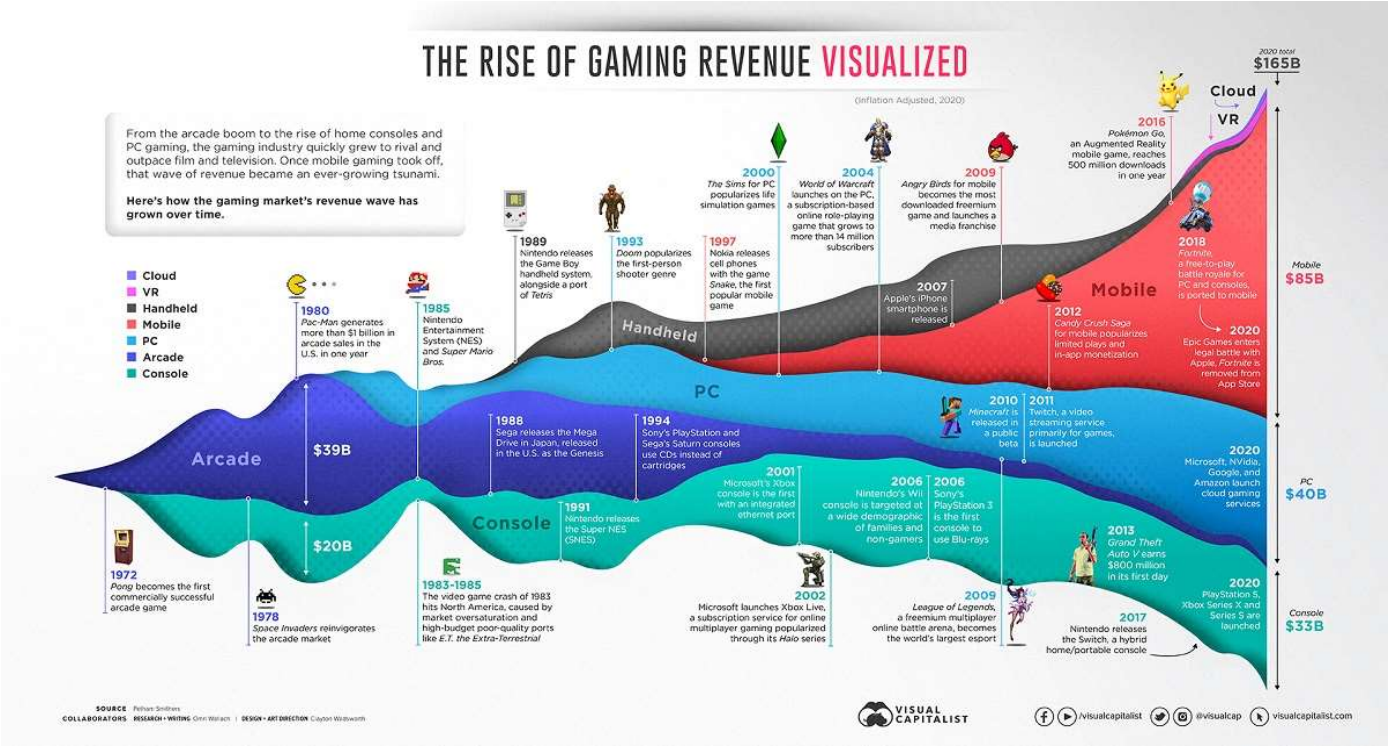

Figure 1. 2- The Game Industry performance in the last 50 years

Based on these values and projecting a blight future in the industry, it's impossible as a gamer and programmer, not be interested in joining the game industry in the future. And that's why choosing a game as my final project for the Computer Science degree was the best decision that I have made.

The Theme

My game *Lord of the Rings: Tactics* is based on the famous fantasy story of the "***Lord of the Rings***" written by J.R.R. Tolkien in 11 November 1954 (Tolkien, 1968). In this fantasy, an artefact called the "The Ring of Power", must be destroyed due it's evil power to corrupt and bend other races to the rings master will: Sauron.

With Sauron's death, his spirit was still trapped in the ring and only the same could be used to bring his master back to live. In order to avoid that and bring the peace back again to the Middle Earth, where the fantasy takes place, a group of heroes from different races team up so they can bring the ring back to Mordor, the only place where it can be destroyed.

Among the heroes, we have Frodo Baggins a hobbit from the Shire, who is the only one capable to carry the "Ring of Power" and resist Sauron's will. Also, we have three hobbits from Shire: Merry, Pippin, Sam; the dwarf Gimli; the elf Legolas and three humans: Boromir, Aragorn and Gandalf (actually Gandalf is from a different non-human race, but let's keep this introduction short).

The "Lord of the Rings" fantasy isn't just a great story, filled with great characters but it's also a huge universe filled with massive details and content. The lore, races, the languages created by Tolkien (like the elvish language), locations, side stories, etc. Everything created from scratch and rich in details…

That's most of the reason why "Lord of the Rings" has been a success even years later its creation and off course not only by the content itself, but also the quantity of other fantasies that were influenced by the same like: Harry Potter, Dungeons and Dragons, boardgames, Narnia, etc.

My own experience with "LOTR"

Most of the reason why I have decided to make a career in games development comes from "LOTR". My first experience with "LOTR" came from the first movie in 2001 released by Warner which by that time, at the age of thirteen, I was extremely excited and shocked by the end of the movie.

I was so anxious and full of joy with the fantasy and the characters, that I started to heavily play RPG games like Tibia, Ragnarök, Magic: The Gathering (card game), Final Fantasy and others titles. The following sequence of movies, which concluded the "LOTR" plot, just added more and more love for this series.

Later on, when I was nineteen, I decided to read the books which only added more details to my own version of the story. And finally, after managing to move outside Brazil like Frodo did when he left The Shire, I am trying to create today my own history into the games industry and find my way to Mordor.

Game Genre

The game genre of *Lord of the Rings: Tactics* is mainly a boardgame, with elements from tactic's RPGs and the traditional battle ship. The idea here was to create my own version of the "LOTR" where all the main characters were able to reach Mordor together, but they have to face a last time the Nazgul on their way to the Mount of Doom (where the ring must be thrown).

On this version, the heroes start at the top left corner where the Black Gate should be located (entrance to Mordor) and the Nazgul are located in the bottom left corner where Minas Morgul is located (the Nazgul Castle).

The following topics will cover more details about each reference for my game genre.

Chess

The core mechanic for my game comes from the classic Chess game, where the thirty-two pieces must fight to capture/defend the king, which is the most valuable piece in the game. Also, each piece has its own set of movements and players do their moves once per turn.

In my game the logic is similar because Frodo is the most important piece and once captured, the game ends. Also, each piece has its own set of movements, but the players are allowed to do two actions per turn (not with the same piece).

Another slight difference is that if the player controlling the Ring Society, manages to bring Frodo to the Mount of Doom, then the game ends. Which allowed me to keep the game close to the original lore, where the heroes were more concerned in destroying the ring and not just fighting the enemies.

Final Fantasy – Tactics

Another game that I have constantly played during my childhood was *Final Fantasy – Tactics* for Playstation 1. The game was my first experience with what would become later on known as "Tactical RPG" genre, a type of game where we have role-play characteristics like character creation, equipment, skills and lore but combined with a turn system and a grid map for movement / actions.

The genre became popular among some players with titles like *XCOM, The Banner Saga, Fallout: Tactics, Divinity, Pathfinder* and many others (especially

Japanese tittles). But for me it's a happy memory about my long hours playing and gathering the best team/strategy with each save.

Figure 1. 3 - Final Fantasy Tactics

The original idea, for the LOTR -Tactics, was to have something close to it where each character would have at least one unique skill based on the series. Allowing this way, the player to be able to move or spend his action with skills once per turn.

Battleship

Another boardgame that I used to play with my family during my childhood was the *Battleship*, where players have to add their ships in a board with coordinates for "X" (width) and "Y" (height). This way, without each player visualizing their opponents board, the player must guess a coordinate that will probably hit a ship.

A game with an easy system to be implemented and that could be played by anyone without a properly boardgame (just paper and pen). That's why became so popular in the past and even in the school I used to play with my colleagues.

The reason why I would like to implement this system into my game is because the same would allow my game to be close to the experience in the "LOTR" series, where the enemies were tracking and trying to find Frodo. While the heroes would try to avoid that, but at the same time, without a clue where the enemy would hit next.

However, I would remove the coordinates necessity and only keep the idea of hiding the pieces, so the players wouldn't have a clue about the opponent's position on the board.

2D Game Project

Due my lack of experience modelling 3D models or use programs such Blender for it, I was afraid to begin a 3D game project and don't finish it on time for the due date. For that reason, I decided to base my project in a 2D game pixel.

This way, I would also not just create something that remind me my childhood playing classic games for Super Nintendo, but also add some nostalgic and simple visual (which my skills could achieve).

But the original idea was to create a 3D game where the camera would be positioned in an isometric way. The pieces would be modelled like chess pieces, but characters from the "LOTR" and the map would have more details and unreachable areas (like the *Final Fantasy – Tactics*).

After some brainstorm and some concern with my time available for the project, I decided to move with the idea of a simple 2D game. This way the camera was positioned on the top so the player could have the whole vision of the board.

Characters

Figure 1. 4 - Frodo image

Frodo

Among all the pieces available, Frodo is the most important and must be kept alive during the game. As in "LOTR", his journey is to travel to *Mount Doom* in order to destroy the ring, so his fighting skills shouldn't be great if compared with the other pieces.

Also, due the Ring of Power, his movement is limited and requires some extra turns to allow the player to move him to Mount Doom. A similar characteristic with the *King* on Chess.

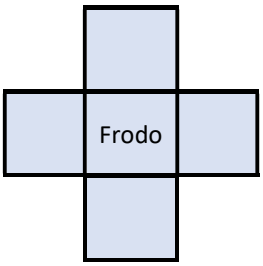

Figure 1. 5 - Frodo movement

If the skills were implemented, because Frodo is a hobbit and his ability to hide is high, the opponent wouldn't be able to visualize Frodo in the battlefield. The same would have to explore the map and reach a close distance from him to be able to visualize. That would allow the player playing as *Heroes* to create different strategies of how to move Frodo to his final destination.

Sam, Pippin and Merry

Figure 1. 6 Sam, Pippin and Merry

The other hobbits in the story are Sam, Pippin and Merry and they have great importance in Frodo's journey to keep him alive and protected. In my game, they can be used to lure the opponent to a different direction when trying to capture Frodo. If the skills were implemented, they would have also the ability to hide from the opponent in far distances. Requiring the same to reach a close distance from them.

Their movements are the same as per Frodo.

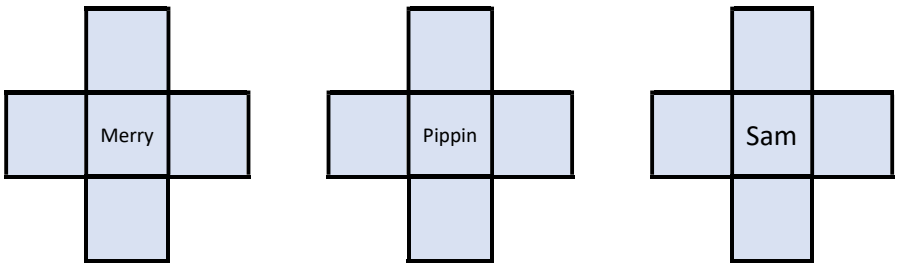

Figure 1. 7 - Hobbits movement

Gandalf

The most powerful character in story is also a great piece in my game. Can cover great distances and in all directions, allowing the player to use him to easily destroy the *Nazgul* pieces.

Originally Gandalf would have two lives, so once killed would appear again in the boardgame after a determined number of turns. But due the Project Timeline I had to change it.

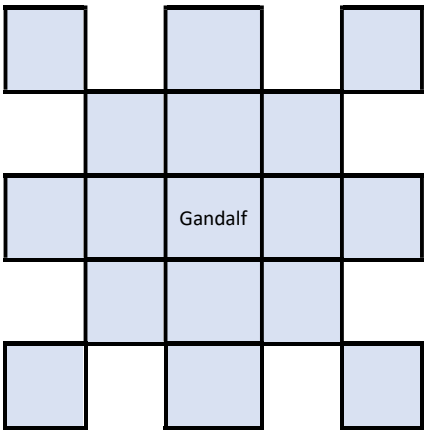

Figure 1. 8 - Gandalf Movement

Aragorn

Because Aragorn is a skilful warrior and he is really important in the story, his move set is great and allows the same to be able to defend the others from the Nazgul. The same can cover great distances and move in any direction.

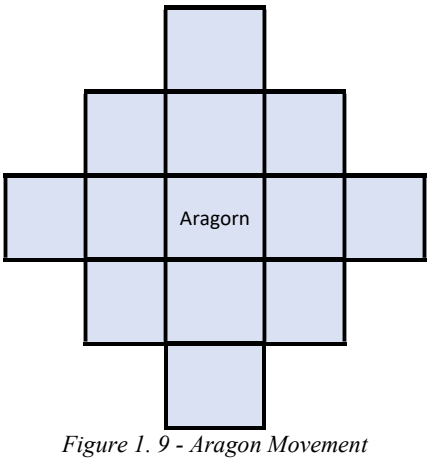

Figure 1. 9 - Aragon Movement

Boromir and Gimli

Both Boromir and Gimli have the same movement set which reflect their close combat.

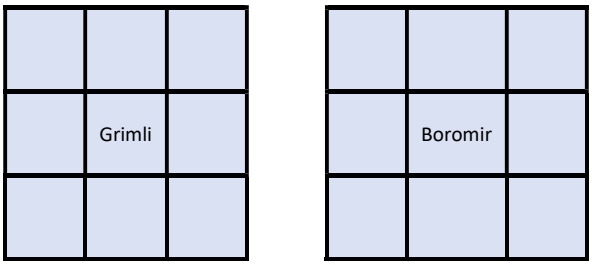

Figure 1. 10 - Boromir and Gimli movement

Legolas

Legolas has his movement based on the range of his bow. If the skills were implemented in the game, the same would be able to hit the enemy without having to actually move from his position, which would simulate properly a range weapon.

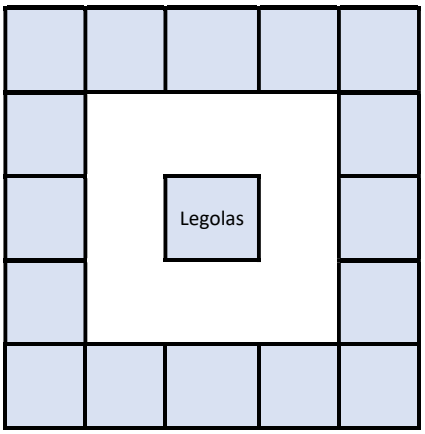

Figure 1. 11 - Legolas movement

Nazgul

The idea with Nazguls' movement was to balance their movement in some way that wouldn't affect the balance of the game, but at the same time would

represent some threat to the opponent. So, because they are spectres following the master's order, I decided to implement a movement that they have to move in straight lines but can reach great distances.

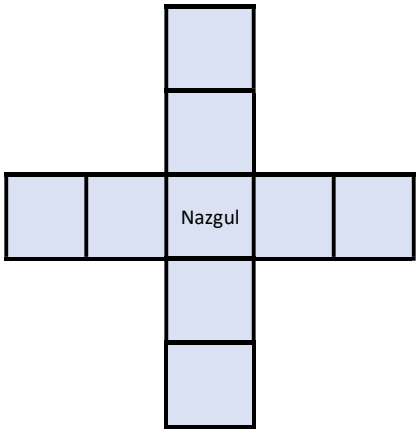

Figure 1. 12 - Nazgul Movement

Witch-King

Representing the bigger threat against the *Heroes* and to balance the game, the Witch-King has the same movement as per Gandalf.

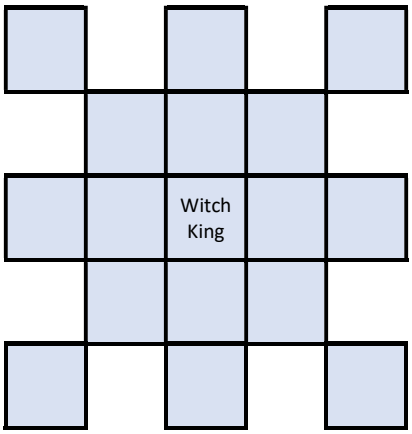

Figure 1. 13 - Witch King movement

Map

The map was designed to be similar to the representation of Mordor in the books which the same is surrounded by mountains. Having in the top left the Black Gate, which is one of the main access to Mordor and where our heroes in the game will start, while in the south we have Minas Morgul, the Nazgul's fortress.

Next, we have the “Mount Doom” which is the vulcan where the *“Ring of Power”* was forged. And beside it, the dark lord fortress Barad-dûr or Dark Tower, which is described to be higher than the mountains and has in the top of it the Flame Eye.

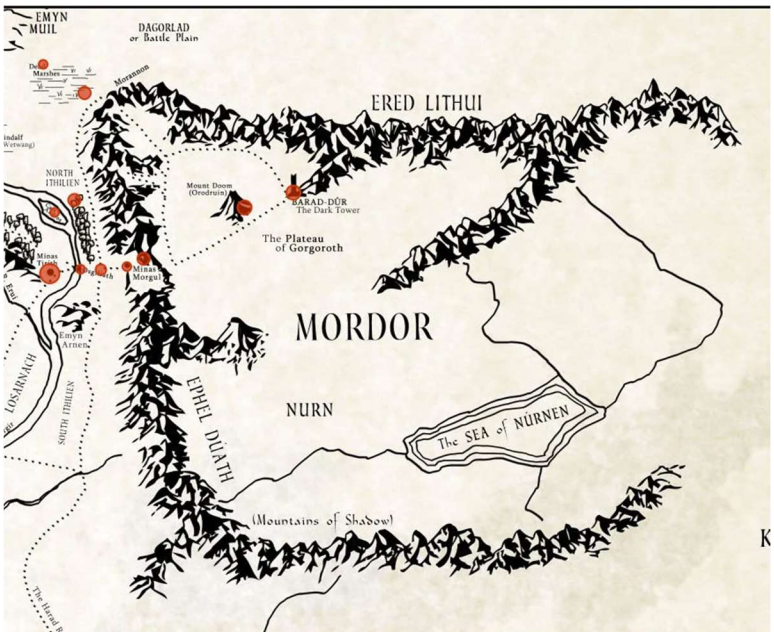

Figure 1. 14 - Mordor map as originally described in the Lord of the Rings books

Based on that description I decided to create something similar using tile maps, which would represent this scenario of destruction, dust, lava and rock. Also, the respawn position for the pieces, the “*Mount Doom*” and the “*The Flame Eye*” were positioned similar as the original map described.

The image below shows the map of the game with the pieces in their respective starting positions.

Figure 1. 15 - The map of Mordor in my game, capturing just the Black Gate region, Eye Flame and the Mount of Doom.

Display Resolution

Due my decision for a simple design using pixels and tile maps, I choose to set the resolution to be 800x500. This way, I managed to keep a good resolution and some quality for the image. Also, because the map isn't big, didn't require extra resolution to show all the details.

Main Menu

The first step to stablish my Main Menu was to find the best image that would reflect the dark atmosphere that is present in the "*Lord of the Rings*". Different from the "*The Hobbit, or There and Back Again*", which is a children's fantasy and was the first success for J.R.R Tolkien, "LOTR" is a fantasy with dark moments where evil and good forces fight each other.

Based on that, I was looking for a dark theme that would bring the feeling of "fighting" against dark forces. Also, I needed something more related to Mordor and not the movies.

Figure 1. 16 - My game main menu

Finally, I found the perfect image, which was not just dark enough but had also the Mount Doom and the Barad-dûr with the Flame Eye. Which soon gave me the idea to add particle effects for the smoke, lava and fire effects coming from them.

The text font for the menu was also based on the font used for the movie's title. And for the font colour I initially thought about something red/dark, but soon I realized that was just too much red for the Main Menu so I changed the font colour to be yellow/gold which reminds the ring itself.

The menu options were designed to be simple and divided into:

- Play
  - Which will allow player to choose between
    - Local – Two players sharing the same screen.
    - Multiplayer – Players playing each one his version of the game.
- Game Rules
  - Which will contain a quick explanation of the Game Rules.
- Quit
  - Close the Game

And a sound button located in the corner left, which will allow the player to turn on/off the background music. Which is our next topic.

Background Music

Before jumping into the game's music, it's important to stablish why I decided to pick up this song. If J.R.R. Tolkien was alive to watch the movie "*Lord of the Rings*" released in 2001, I believe that he would be not just proud of the quality that the movie reflected based on his books but also, he would fell in love with the music itself.

The music was composed, orchestrated, conducted and produced by Howard Shore, a Canadian composer and conductor, whose won three Academy Awards with his work in this movie and it's memorable. After twenty years, watching this movie still brings me chills to my skin, especially because of the music.

The music captures the essence of the story, the drama, the darkest moments and off course, the triumph moments. The orchestra adds also some feeling of knowledge/old which combines perfectly with the atmosphere in the movie and the drama which the characters are involved.

Finally, wouldn't be possible to create a game based on "*LOTR*" without adding the music from the movie in the game. Which again, combined with other elements like the Main Menu background image, just brings the feeling of urgency and fear of fighting Sauron's dark forces.

Game Design Flow

The following diagram represents the game design core idea that my game would follow, where *Player 1* would have to move *Frodo* to *Mount Doom* in order to win the game and use the heroes in the process to defend his path.

While *Player 2*, commanding the Nazgul forces, would have to capture Frodo in the process. And with the aid of *The Eye*, the player would be able to check specific spots in the map in order to locate Frodo and his enemies.

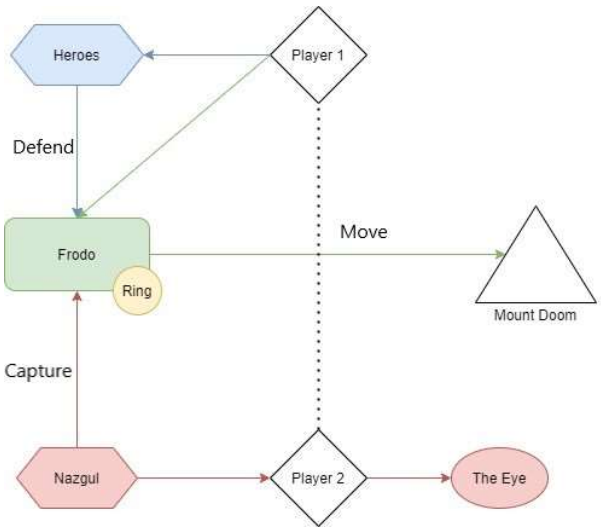

Figure 1. 17 - Game Design flow

Chapter 2. Gameplay

This chapter will cover some mechanics designed for the game.

The turn base system

The game is based in a turn base system, which comprehends into players executing a pre-determined number of actions each turn and at the end, the next player's turn will follow till the game ends.

It's a pretty common type of game mechanic and has been in the game industry for a long time. The only difference it's because for my game, the player will have two actions to perform before his turns ends.

The two actions were added in order to speed up the game and the pieces movements and to allow the Heroes, which contain pieces with a single square movement like Frodo, to be able to move it and also a second piece for attack/defence position.

After the conclusion of each player's turn, a message pops up followed by the Nazgul's image or Heroes' image showing that their respective turn has started.

Hidden Movement Mechanic

Following the game design inspiration from *Battleships* classic boardgame, the hidden movement game mechanic was added to bring the feeling of the unexpected

and a strategy mindset to the game. Allowing players to build this way different strategies in how to move Frodo to his destination or how to avoid it.

The mechanic consists of basically allowing the pieces to visualize at least three squares around them. Which the same would be reduced in the case of a piece trying to visualize a hobbit (not implemented), reducing this way the range to two squares around the same.

This strategy not just benefits the player who controls *The Fellowship*, but adds a small advantage also if well used. The player could move the hobbits in different directions and force the enemy to hunt them and once found, the same could be lure to a different direction of the map while Frodo is sneaking in a different direction.

Another concept important of the hidden mechanic is to balance the game, once *The Nazgul* have better movement distance and can easily cover different regions of the map. While "*The Fellowship"* have at least five different pieces with a limited movement set, which not just reduce the capacity to move around the map, but also increases the combat difficulty.

Another curious point to add is because my decision to use pixel art and a small resolution, lead to a scenario where is difficulty to differentiate characters like Merry, Pippin and Sam. However, such negative aspect can be enhanced by the hidden mechanic, once the player controlling *The Nazgul* might not recognize exactly if he found or not Frodo. Which is again another particular event that happens in the books, when Merry and Pippin are captured by orcs but they are not sure if the same are the correct hobbit that they are looking for.

The Eye Mechanic

Another game mechanic that I was anxious to add in the game is the Eye mechanic, which is directed linked to the Hidden Movement mechanic. Having both mechanics present in my game, would enhance the same and be so close to the lore as possible.

The Eye Mechanic would give another balance in the game, by allowing *The Nazgul* to spend one action per turn and be able to visualize five squares (in a cross format) anywhere in the map. With a cooldown of three turns, *The Nazgul* would be

able to identify possible enemies coming close to *Mount Doom* or even locating Frodo's last position.

Such game aspect would give a feeling of terror and again, would improve the strategy aspect of the game in such way that both players would have to take their actions well planned, otherwise they could lead their own game to an early end.

The Mount Doom Indication

Another simple game mechanic added to the game was the *Mount Doom* "arrow" indication. Instead of adding a single tile map different from the others, stablishing this way the destination where Frodo should go, I decided to add a ring animation that would point out the correct square.

This way, by selecting Frodo's character, the ring is displayed over the destination where he needs to go. The same is turned off once another piece is selected or when clicking a movement tile.

The ring animation consists of a small animated ring, similar to the *The Ring of Power*, which moves in the 'Y' direction up and down while Frodo is selected. Such mechanic is simple, but adds more design to the "LOTR" universe.

Pieces Movements

In "LOTR Tactics", each piece has its own set of tiles for movement and the same reflect characteristics of the characters in the story. This way, characters with some limitation for movement like hobbits due their size, will have a proportional number of tiles while powerful characters like Gandalf or Witch King, will present a greater set of tiles.

Boardgames for example like "Checkers / Draughts" are dated of 3000 BCE and had already the concept of pieces moving accordingly to tiles and the limitation of each piece. With game development moving to the digital world, such mechanic was well adapted and implemented in the games early on.

For that reason, I have decided to add such mechanic to my game, so I could exploit its simplest structure, but adding different layers based on the characters from "Lord of the Rings".

Chapter 3. Development

This chapter will cover the stages that I had when implementing my project and the strategies and technologies that I decided to use.

Paper and Pen

First step of my project was decided in many pieces of paper in format A4 and a lot of hours drawing and thinking about some mechanics that I would like to developer. This phase helped me a lot to establish a mind map of what I was desiring to achieve in terms of gameplay and design, also was a quick introduction of what would become my game.

After some drafts, I finally managed to recreate a boardgame and using fake pieces I started to "play" what would become my digital game. First deciding how many pieces for each side; then their starting position on the map; what would be the goal for the Nazgul; what would be the goal for the Heroes; what would be winning/losing condition for both sides and finally some extra details to enhance the main ideas.

Using a Chess board, I applied my game rules and played a couple of times. Each interaction allowed me to have better ideas, remove some unnecessary rules and improve this way the overall idea about the game.

Once the core of the game was decided, I started to upload all my notes and draws to the computer, where I would organize everything and stablish some milestones for the project.

Setup the Project

An important step in the project that truly helped me to keep track of my tasks and organize myself before and during the execution of the project was to setup the milestones and activities in different places. The following technologies were used in order to improve my results.

Trello

It's important to state that Trello had a great contribution for the overall project in terms of organization, timeline and agile process. Using Trello, I was able

to organize my ideas into categories, which I would use as milestones for the project along the year.

The project was divided into:

- LOTR 1.0
    - First stage is where I setup the base structure for the game on Unity: boardgame, GameObjects, Pieces, Grid, Game Rules, Movement for the pieces, Turn system, etc.
- LOTR 2.0
    - Second stage I added minor features and details like: Mount Doom, basic UI, The Flame Eye, improve the movement set, etc.
- LOTR 3.0
    - Here I planned to add: main menu, fire/lava animation, sounds, multiplayer system, etc.
- LOTR 4.0
    - And finally, the last stage should be dedicated to: testing, prepare game documentation, fix major bugs and create the demo.

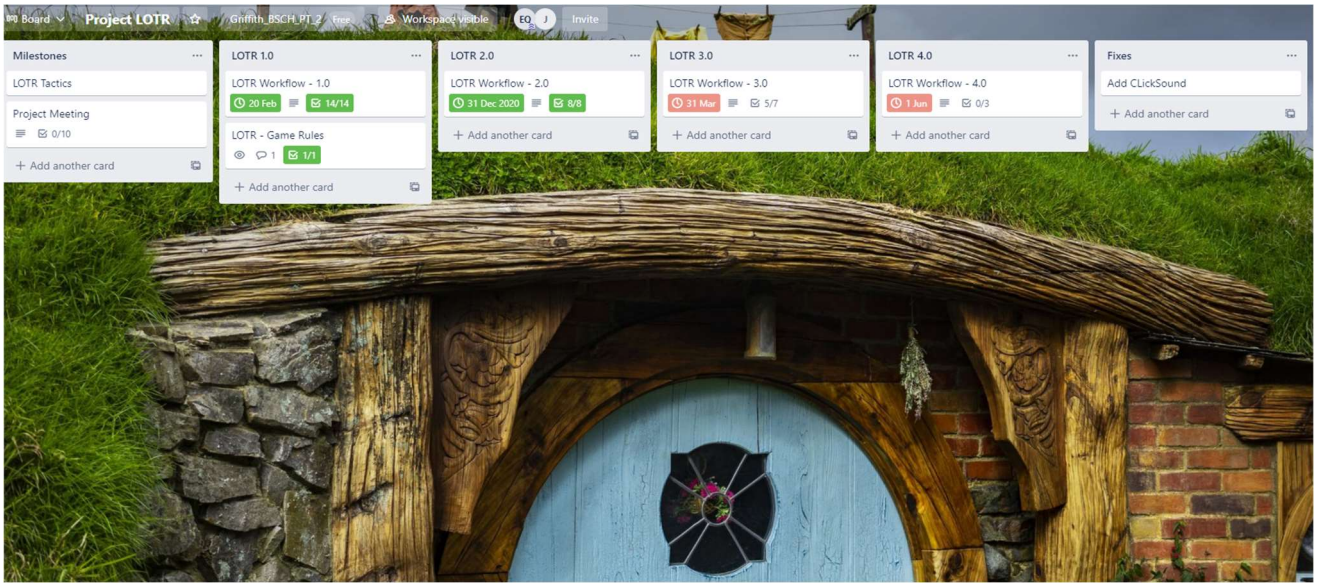

Figure 3. 1 - Trello project board

This simple structure helped me a lot to organize myself every day before starting working on the project and it's something that I have implemented for my

own life and workday. Preparing myself for the task before starting it, allows me to visualize possible solutions and problems along the way and quickly adapt myself to it.

Timeline

The following timetable was designed based on the original milestones, back in October 2020 where I setup the Trello and stablish some due dates for each phase of the project. Unfortunately, I couldn't follow exactly the due date as expected once the tasks included here are just the main and during the project minor tasks were added which is one of the reasons for the delay in the conclusion of each phase.

Phases	Main Tasks	Due Date	Conclusion Date
LOTR 1.0	Fill up the Project Form	20/02/2021	15/01/2021
	Game Rules		
	Design the board game		
	Set the camera		
	Set the light		
	Set the Grid		
	Set the pieces		
	Create the Board Game		
	Pieces selection		
	Pieces Movement		
	Pieces combat		
	Movement Tiles selection		
	Adding Players 1 and 2		
	Turn system		
LOTR 2.0	Pieces Sprite	30/03/2021	20/02/2021
	Board tiles sprite		
	Mount Doom should be added		
	The Eye should be added		
	Pieces starting position		
	UI message for each turn		
LOTR 3.0	Local mode is playable	30/04/2021	30/05/2021
	Main Menu		
	Flames animation		
	Multiplayer system		
	Sound effects		
LOTR 4.0	Demo	30/05/2021	14/06/2021
	Documentation		
	Testing		

Table 3. 1 - Timeline table for my game

In the Chapter 5 – Conclusion I will discuss more details and different approaches that I should have done to be able to follow the timeline above and to manage to implement all the features that I wanted.

Microsoft To Do

In order to keep track of my daily activities and plan myself during the hours working on the project, I added each daily task to my Microsoft To Do application. The software is free and part of the Windows 10 package.

Before using it, I usually would expend hours executing something for the project, but the simple action of adding daily activities before working on them helped me to keep track of the time wasted. This way, I would select the quickest ones first and then move to the ones that I know that would take more time to conclude.

GitHub

I have to confess that I don't have much experience using GitHub but since the beginning I would like to have one in order to keep a repository for my project. For this reason, I have created one (LOTR – Tactics) but I still have to work on it in order to improve.

Technologies Used

The following topics will cover technologies used specific for the development of my project, not just for the planning phase.

Unity - Game Engine

One of the biggest steps that we have to do before creating our own game is to decide which game engine to use. There are many different game engines available in the marketing, for different proposes, project sizes, necessity, lacking of coding or even using specific programming languages.

For my project, a simple 2D game based on pixels, most of the game engines would work fine, however I decided to use the Unity to build my game for a couple of reasons.

Firstly, Unity is used in the game development module in Griffith College, which would allow me to gain more experience while having classes and doing my

project. Also, I would be able to bring some questions regarding the engine to class related with my own project.

Another point would be my previous experience studying and practicing with this software along the last couple of years. This project was my first time to get my hands in a big project and manage to produce a playable game from it but by having such previous experience, gave me confidence in the engine and the results that I would be able to accomplish.

The technical points of choosing Unity over such big game engine software like Unreal would be the wide community that Unity has; the programming language which is C#, while Unreal uses C++ and for gamming purposes, C# has a better performance ; Unreal is well-known by its graphic qualities but I believe such potential would be wasted in my project and finally, Unity has a huge store with many mods that I could eventually use to add mechanics, characters, etc.

Finally, a personal reason that I could point out as one of the motives for me to consider Unity as my game engine is because in the last couple of years the same has been responsible for big games to be released, especially indie games, like: *Cuphead, Escape from Tarkov, Hearthstone* and *Hollow Knight* are some of the examples.

So based on these points, I believe that choosing Unity was the best decision made and the one that would suit my timeline available for learning/implementing the game engine as per the project's tasks. Other game engines would require some time for adaptation and learning that could eventually affect my project in the future.

Unity - Hierarchy

If there is something that I have learned during my experience working with Unity it's the importance of the hierarchy and the organization associated with that. Similar to the code itself, which requires some structure and pattern in order to be better understood, the hierarchy may be enhanced by this practice also.

As we can see below, both Hierarchies are from the two scenes present in my game. By using a simple technique to add empty Game Objects and naming them accordingly with the category that I want to represent, I can divide each component in a specific category.

Even for a reader, without any experience with my code and the project itself, it's easy to identify what each GameObject is referring to. Such process not just

helped me to improve my organization but can I also avoid small mistakes, unnecessary content in the hierarchy and improves my understand about how the game is working.

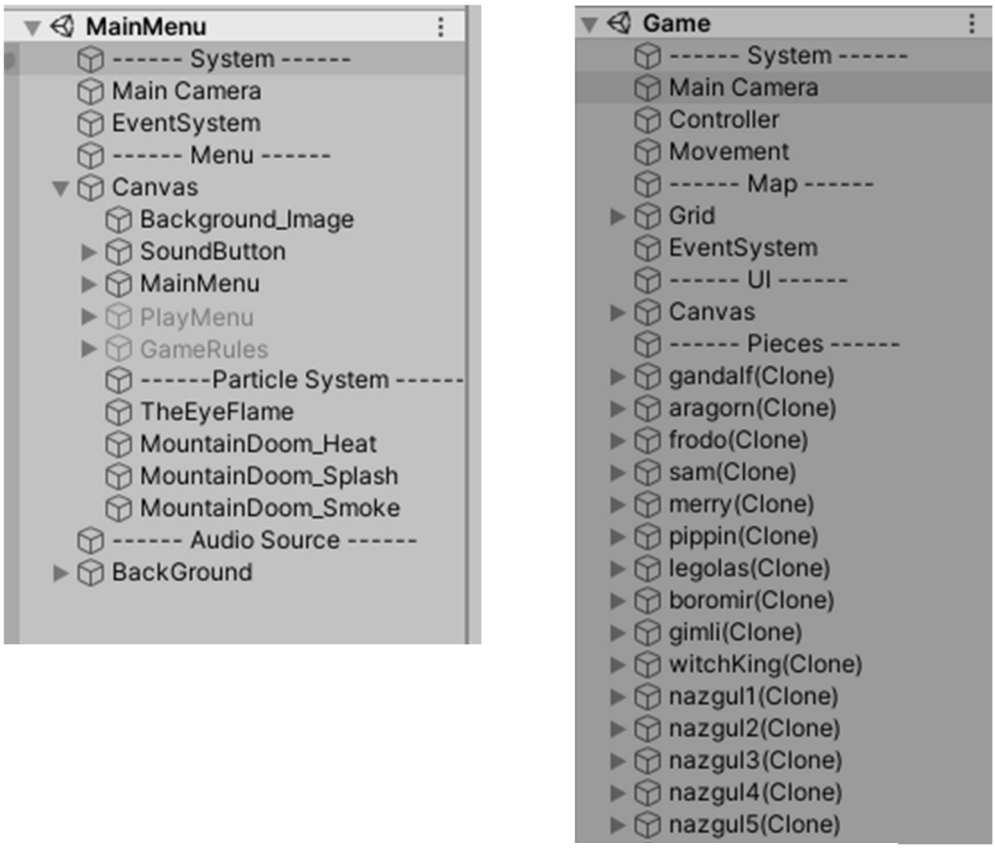

Figure 3. 2 - Unity hierarchy structure that I have setup

Unity – Project Folder Structure

Another key point really necessary to keep the project organized and structured is how the folder is setup. In order to keep the project clear in such way that anyone accessing the same for the first time can be able to locate a component, I had to organize it into different levels.

The folder structure was divided into:

- Fonts
  - Adding all font styles and Text Mesh Pro components
- Audio
  - All the audios resources: music and click button sound event.
- Animation
  - All the animations present in the game.
- Photon

    - Package used by Unity to allow us to create and run games in multiplayer.
- Prefabs
    - Where all the prefabs for my game objects would be stored
- Scenes
    - A scene in Unity would be like a different layer in the game, such as main menu; different level in a puzzle game; a cinematic used during the game, etc. So, for my project, I used this folder to store the scenes for *Main Menu* and the *Game* itself.
- Scripts
    - Where all the C# codes were saved for my project.
- Tiles
    - The tile maps and images used to create them. Here is where the boardgame components to create *Mordor* were saved.

Multiplayer System – Photon vs Mirror

In order to implement the multiplayer system into the game I decided to research for a technology that would be more suitable for my project. While searching, I came across two particular technologies: Photon and Mirror.

Mirror is a well-known networking API that allow us to connect players with each other in a high scale and has been even used in MMORPG projects, connecting hundreds of users at the same time. The asset can be easily download from Unity Asset Store and added to the project, without the necessity to code a separate network script for it.

While seems to be nice and has a great showcase available, Mirror looks to be more suitable for large projects where not just connecting high number of users is necessary, but also the network security behind it must be great against possible cheaters.

For that reason, I decided to follow the project with Photon instead of Mirror, because Photon has an easy and friendly API design that allow us to quick understand how it works and also can be easily setup once you learn the tool. Also, the first step which is to setup users and room was really easy to do and has a lot of tutorials explaining it including their own website (Photon, 2021).

Photon

As previous mentioned, Photon is a Network API that allow us to implement the multiplayer system. Includes also services like cross-platform, chatting system,

Realtime cloud, connection direct from client to server without relying in WebSocket and has free access for games up to 20 players.

Also, it's easy to understand and overwritten the Pun (Photon Unity Networking) functions, which allow us to quick create and implement new logics for the creation and connectivity of our users.

By using functions containing the header "[PunRPC]", which I will cover later one, we can stablish how the method will be executed during the game. If the same will be reproduced only in the master client, local, all clients connected into the game, if the same will be buffered so anyone joining later can also receive the previous messages, etc.

Once I understood these concepts and managed to create my own functions, everything started to work as expected and I was able to at least create a working multiplayer system where two players can interact with each other in a turn base system.

Photon - Room Setup

As mentioned, the room setup was so easy to implement that I couldn't believe that connecting a multiplayer system would be require just a few steps (at least the connection). First step was to create an *App. ID* on Photon's webpage, which I would use such ID to link in my code when creating the room.

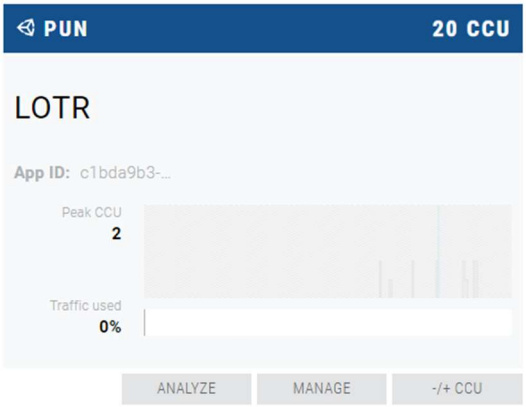

Figure 3. 3 - Registering our App. ID on Photon

Once the project was created and I got my *App. ID,* I just had to import Photon from the Unity Asset Store and pass over the App. ID into the popup that shows up after downloading.

Figure 3. 4 - Adding our App.ID on Unity

Once this step is done, I just had to add the following code into the *Connection* script that came with Photon.

```
// -----------------    Accessing Lobby    -----------------//

public override void OnJoinedLobby()
{
    // Accessing my Lobby
    PhotonNetwork.JoinRoom("LOTR");

}

// -----------------    Creating Room   -----------------////
public override void OnJoinRoomFailed(short returnCode, string message)
{
    if (returnCode == ErrorCode.GameDoesNotExist)
    {

        // Settings our room

        RoomOptions roomSettings = new RoomOptions { MaxPlayers = 2 };
        // Creating room
        PhotonNetwork.CreateRoom("LOTR", roomSettings, null);
        PhotonNetwork.LocalPlayer.NickName = "PLAYER1";
        PlayerName = PhotonNetwork.LocalPlayer.NickName;
        nazgulPlayer = PhotonNetwork.LocalPlayer;
    }
}
```

Code Snippet 3. 1 - Implementing the connection to Photon server

Here I am basically telling Photon to connect to my lobby “LOTR” which can contain multiple game rooms, where the player will actually play the game. After that, I just needed to stablish the room settings like for example the maximum number of players per room and differing local players from master client.

And that's it, the room was setup, the connection was working and I just had to move to the logic process to code the [PunRPC] functions so the players could communicate with each other.

Chapter 4. System Flow

This chapter will present the design and flow of my project. Also, I will add some logic behind my code.

Class Diagram

In order to better understand the project flow and how the coding was designed, the following class diagram represents all the classes present in my project

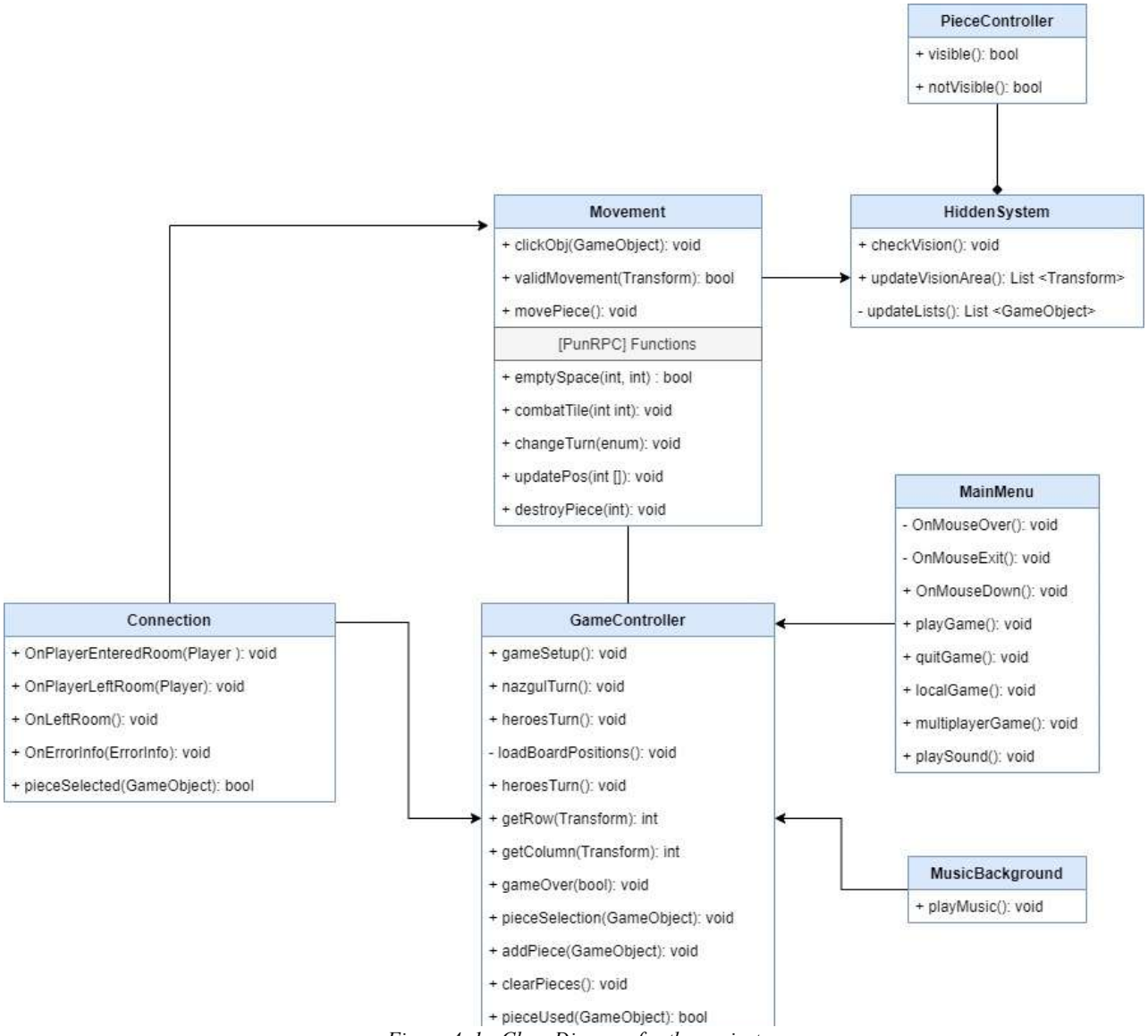

Figure 4. 1 - Class Diagram for the project

and, the relationship between them, their functions, parameters and return values. That helps us to visualize the system and to have an overview of each element.

Classes Overview

The classes were designed to be built around the *GameController* and *Movement*, where would contain the core mechanic in the game and the main functions in order to make the game run. Game Controller contains functions related with the setup of the game like preparing the turn phase for each player, creating the boardgame to store all the pieces, recording pieces already used in the turn, etc.

Movement class was designed to cover all the functions around the piece's movement in the boardgame, this way it contains functions like checking the piece clicked, checking the movement path for the same, executing the movement, checking the children related to the object in movement, etc. Also, we can find here some functions that will be called by the multiplayer mode, the PunRPC functions, which in general will repeat the same structure in the local game but adding some differences like how the same will be transmitted for each user or in which order.

Connection is a class build-in with Photon and was added in the project after importing the same from the Unity Store. It contains a series of functions that can be overridden so we can expand the same and add customs logics as per game requirement, but in my project, I have basically used the same for setup the rooms for multiplayer connection, distinguish the players connected and testing.

MainMenu contains all the functions related with the *Main Menu* scene, where the player will find the first screen before jumping into the game. Here we will have functions for start the local mode, multiplayer mode, some TextMesh Pro functions to animate the text when mouse hover, quitting the game and playing sounds when clicking the button.

MusicBackground was added just to be responsible to set On/Off the background music. I could add it to the, however I thought that adding a separate class would help me to keep my code clear.

HiddenSystem controls the game mechanic that hides enemy piece's movement from players and have the *PieceController* class dependent on it. *HiddenSystem* has functions like calculate the overall vision of the pieces, update the same based on the piece's movement and by using *PieceController* functions,

HiddenSystem can interact with the pieces and enable/disable their sprite which will "hide" them.

Sequence Diagram

The following sequence diagram shows the order of each Game Object. Here it's possible to see that the player will interact with the Main Menu, which will lead them to the Game.

Once playing, both players will play their actions in a turn system base and once the game is finished, both will have the opportunity to retry the game or end the application.

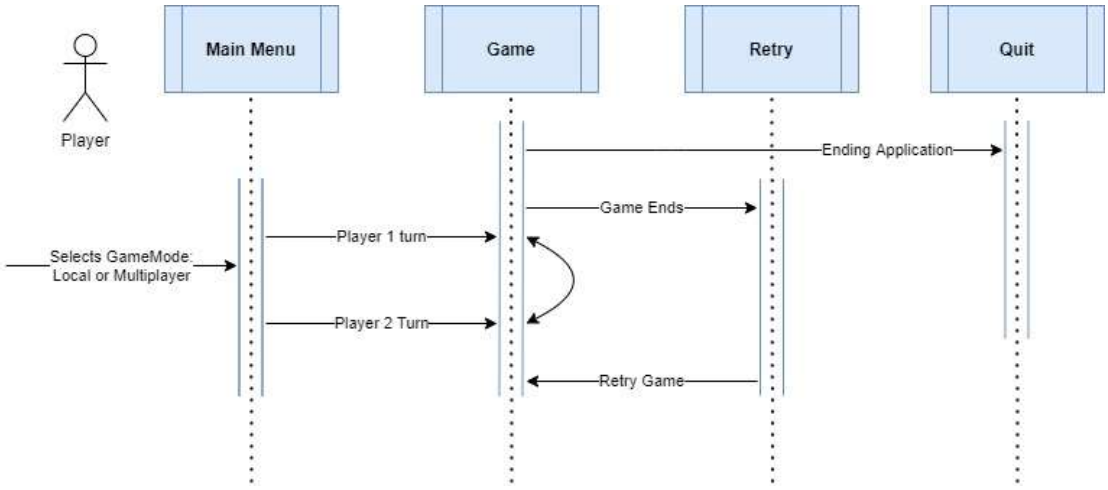

Figure 4. 2 - Sequence diagram for the project

Code Snippets

As mentioned before, this project was my first experience designing and creating a game from scratch, but also it was my first experience coding something on my own and having to decide how the logic would work. The following code snippets will cover some of the core logics that I decided to implement.

Some of the logics / codding I am not fully happy with the result or the implementation. I truly believe that there are different approaches that I could have taken in order to make it clearer and more professional, but that's something that I will state in the conclusion section.

Movement

If there is a logic that took me some time to improve was the Movement class. It's the core of the game and would require the pieces to move in a pre-determined set of movement, inside a limited range (the boardgame) and being able to identify enemies' pieces in the process.

So, the following diagram will give a brief idea of how the logic was implemented.

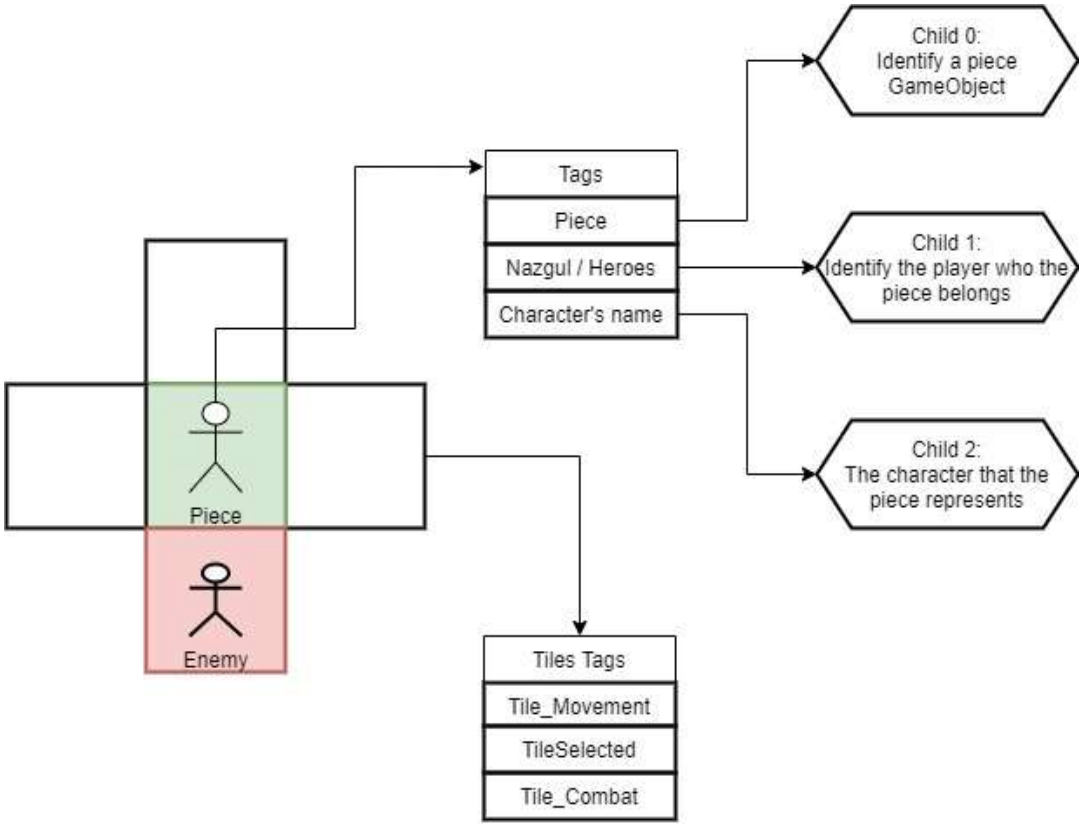

Figure 4. 3 - Movement diagram

As we can notice in the diagram above, the main logic around my game was done using tags in order to trigger the mechanics. Therefore, a piece contains multiple tags that we will use in the following code snipped to identity this Game Object and execute specific actions.

```
public void clickObj(GameObject obj)
    {


        objSelected = obj;

        …

        //Selecting pieces
        if (objSelected.tag == "Pieces" && game.pieceUsed(objSelected) == false)
        {

            …

                ////Displaying entrance to MountDoom
                if (objSelected.transform.GetChild(0).tag == "frodo")
                {

                    game.RingPointer.SetActive(true);

              game.RingPointer.GetComponent<Animator>().SetTrigger("Frodo_Selected");
                }
                else
                {
                    game.RingPointer.SetActive(false);
                }
            }

              …
        }



      ….
    }
```

Code Snippet 4. 1 - Selecting a Game Object and checking for the tag "Pieces"

As we can see in the code snippet above, once a player selects a Game Object it will trigger a method that will try to identify exactly what type of object is the player clicking. If the same contains the tag “Piece”, then a nested condition will be trigger and will execute different checks.

As we can see in the snippet, one of the checks was in relation to the piece’s character name “Frodo”, which if true, should trigger the animation ring over the Mount of Doom, else should turn it off.

The same strategy was applied to the movement tiles, where for each one of them were added as child to my piece object. Then, once the piece was clicked, the following method would check the tiles and execute specific conditions based on their tags.

```
public void clickObj(GameObject obj)
{
        …
        //Selecting pieces
        if (objSelected.tag == "Pieces" && game.pieceUsed(objSelected) == false)
                {

                    // Loop checking children and executign properly actions
                    foreach (Transform child in objSelected.transform)
                    {
                        //Showing up tile selected
                        if (child.tag == "TileSelected")
                        {
                            child.gameObject.SetActive(true);
                        }

                        //Showing up the tiles for movement
                        if (child.tag == "Movement")
                        {

                            foreach(Transform tile in child)
                            {
                                if (validMovement(tile))
                                {
                                    int childXPosition = game.getRow(tile);
                                    int childYPosition = game.getColumn(tile);

                                    if(emptySpace(childXPosition, childYPosition))
                                    {
                                        tile.gameObject.SetActive(true);
                                    }
                                }
                                else
                                {
                                    tile.gameObject.SetActive(false);
                                }
                            }
                        …
                        }
                    …
```

Code Snippet 4. 2 - Checking the Game Object selected and checking for the movement tiles tag

In relation to the movement tiles, once the piece is clicked, the code above would check each child in piece and:

- Activate the tile with tag"*TileSelected*", which is the tile where the piece is located (green tile).
- Access the child with tag "*Movement*"
    - Then loop through all the children of "Movement" and check if:
        - They are a valid movement inside the boardgame
        - And if they are empty
            - If not, then the child's tag changes to "*Tile_Combat*" (red tile)

The following code snippet contains the *validMovement*() and the *emptySpace*() methods:

```
//Checking if movement is inside the board
  public bool validMovement(Transform obj)
    {
        if (obj.transform.position.x >= -6.5f && obj.transform.position.x <= 6.5f &&
obj.transform.position.y >= -6.5f && obj.transform.position.y <= 6.5f)
        {
            return true;

        }
        else
        {
            return false;
        }
    }

       …

    //Checking if position is empty
    public bool emptySpace(int xPos, int yPos)
    {
        int x = xPos;
        int y = yPos;


        // GameMode Local
        if (game.positions[x, y] == null || game.positions[x, y] ==
GameObject.FindGameObjectWithTag("MountDoom"))
        {
            return true;
        }
        //Avoiding piece to move to The Eye tile
        else if (game.positions[x, y] == GameObject.FindGameObjectWithTag("Eye"))
        {
            return false;
        }
        else
        {

            combatTile(x, y);
            return false;
        }


    }
```

Code Snippet 4. 3 - Checking if tiles are valid (inside the board) and if the same are empty.

So, as we can see, the tags are heavily used here to trigger the right conditions and even noticing that some logics could be replaced and the number of tags reduced, I thought that using the multiple tags would enhance the logic and the reading of the code. For example: checking if Game Object is a “Piece”, I could replace the same for something like “equal to Nazgul OR Heroes” and remove the tag “Piece”, however reading the code as it is sounds clear the purpose of this condition so I decided to keep this way.

Multiplayer Position

Another method that requires some explanation is how the multiplayer system works to register the opponents movement and share our own movement is the *updatePos*(). Once the game is in multiplayer mode and the player moves his own piece, the same will trigger the updatePos() which basically is a integer array that is send to the opponent.

Such array contains five different information regarding the piece that we have just moved:

- [0] – The 'x' previous position of the piece
- [1] – The 'y' previous position of the piece
- [2] – The new 'x' position for the piece
- [3] – The new 'y' position for the piece
- [4] – The piece's ID number

When receiving these details, the opponent will update his own list of piece's positions in the boardgame. By getting the piece's ID number, the player can set the previous positions to 'null' and add the new piece's position. Again, the reason for the integer array is because we can't pass game objects into the Remote Procedure Calls ([PunRPC]) so the best solution that I found was to submit this way.

```
if (game.gameModeMultiplayer == true)
          {

              // Update Vision
              viewSystem.checkVision();

              // Simply storing pieces positions and GameObject ID
              int[] positionList = new int[5];
              positionList[0] = xPrevPosition;
              positionList[1] = yPrevPosition;
              positionList[2] = newXPos;
              positionList[3] = newYPos;
              positionList[4] = previousPiece.GetPhotonView().ViewID;


              // Updating new position for other players
              photonView.RPC("updatePos", RpcTarget.Others, positionList);

          }
…

    [PunRPC]
    public void updatePos(int [] pos)
    {

        // Set previous positions to null
        game.positions[pos[0], pos[1]] = null;
        // Assign Game Object to new position
        game.positions[pos[2], pos[3]] = PhotonView.Find(pos[4]).gameObject;

    }
…
```

Code Snippet 4. 4 - Updating pieces position in multiplayer game mode for all players

Chapter 5. Conclusion

This chapter will conclude the documentation and add my personal experiences with the project, difficulties/challenges, testing and solutions applied along the way.

Testing

While designing and coding the game I tested multiple times the game in different aspects, not just the mechanics but also the game design. I tried to make sure that the game was at least balanced for both players and that they could perform their actions without taking too much advantage over the opponent.

Also, when trying to implement the multiplayer system, I had to constantly test it in relation to the connection and if both players were able to send and receive actions. Which specially at this point took me some time due the constant crashing of builds due exceptions; code issues; mechanics not working and logics incorrect.

Finally, I managed to test all the core mechanics and logics and at last most of the game was playable.

Issues along the way

During the project I have faced different issues along the way. Most of them were simple/medium issues, related with my lack of experience programming games which I managed to solve after a given time.

Among them, one that took some time to solve for example was the boardgame mapping, where I wanted to create a GameObject array containing all the pieces and their respective positions. Everything packed as prefabs and instantiate them in a multidimensional array where my [0,0] would comprehend the centre of my map.

The issue started when instantiating them in the boardgame, once their GameObject position wasn't synchronized with the grid position. In order to solve that, after a couple of days, I managed to convert each GameObject position to a specific Grid Position and match this way the squares.

Off course, by the end of my project I realized based on a course in Game Development that I bought, that an easiest solution would be to change the grid's position for example and have this way a better logic.

All these issues are normal in a game project and could be solved after some time, however my biggest issue was related by adding the Multiplayer feature which affected not just the overall project's timeline but also resulted in a high demand of energy to solve it. The following topic will cover better it.

The Incorrect Strategy

Again, for my first big game project, I have the feeling that the project was well executed based on my lack of experience. However, I have to point out that some issues along the way were result of an incorrect strategy that I decided to follow in the beginning of the project.

In my Game Design phase I decided to stablish the core of the game that would be something really similar to a chess game but with "LOTR" theme. This way, some mechanics were not present in the project in the first phase and here is where the reason for my main issues with the multiplayer started.

After realizing that my project wasn't much complex and I still had some time to improve it and add more features, I started I second Game Design phase where I would implement more mechanics and features, including the multiplayer. This way, I would allow my game to be played in two different instances instead of just a local machine.

Well, after most of the core game has been created as a local game mode, by adding the multiplayer feature I just not compromised my previous logic but also, I had to redo most of the work already done. And even wasting days trying to implement it, without compromise too much my original logic, by the end I realized that would be more professional and clearer to have two different coding logics for each game mode, instead of trying to bending the local mode into the multiplayer.

So, after such experience has been achieved, I can clearly state now that my strategy should be since the beginning to decide if I wanted to add or not the multiplayer feature. If added into the project, then due the complexity of the remote procedure calls, would be better to start coding based in the multiplayer logic and once done, adding the local mode would be much easier.

Project Timeline

The project officially started during 2020 second semester. After ending the last activities for the third year, I decided to have a look in different projects that I would like to work on for my course conclusion. In October, when we had to submit a project planning describing our ideas, I had already for a couple of months the intention to developer a game based on "LOTR".

After having my project approved, I decided to breakdown the project into categories and milestones which I should accomplish along the next year, which I have described in the Chapter 3. Setup the Project – Trello.

The project progressed well till the start of the second semester for the 4th year (February). Once the classes begin again, the assignments started to become a problem for my timeline.

I lost track of the project and because I had already done much work during the last couple of months, I decided to focus on the assignments when I should actually have dedicated at least 10% of my available time to check my own code or do some small changes, but never fully stop the progress.

By the end of the assignments, I was obliged to take off two weeks from my work so I could speed up the project and put it back on the tracks again. Not just the project wasn't fully completed, but the documentation wasn't done yet.

The last couple of weeks, before the submission, have been toughs due my issues trying to finalize the project and its multiplayer game mode. Also, writing the documentation has been part of my daily project work, which I have managed to write a couple of pages every day in order to achieve the minimum requirements.

Features missing

Some of the features that I have mentioned along the documentation like the Eye mechanic that would allow the player who controls the Nazgul to spend one action to visualize a specific point in the map or the skills for each piece were not added.

That happened due my problem with the timeline for this project and because I didn't want to add partially or a not functional mechanic that could result in eventually my game crashing. For that reason, I decided to remove both mechanics

from my game for now but I intend to add them as soon as my college activities are done.

Such mechanics would add more charm to the game and transform the same into something that any Lord of the Rings fan would fell in love. But as mentioned, I am planning to finish these mechanics and improve my game as soon as possible.

Different approaches

During the execution of the project I realized that some strategies/decisions were not well implemented or didn't achieve what I was expecting. Here I am adding some key points that I would do differently when approaching a new project in the future.

Game Design

For my next personal project, I will dedicate as much as possible time and energy for the design. Not just planning the first concepts, but heavily stablishing the core mechanics, UI, features that I want to add, devices, window resolution, animations, etc.

The perspective that I got with this project is that "there is no waste of time, when the same is with the design phase" and that might not be fully right when thinking about specific projects and their timeline, but personally that's the feeling that I got with this project. I had so many issues due key points that I didn't stablish the concept in the design phase, that I deeply regret not dedicating more time with it.

For example, the 2D pixel art chosen for the game. Initially, I thought that I wouldn't be able to create a 3D game and model the pieces but after some research about it I realize that maybe I could have done it, not in professional way, but at least descent.

Another aspect that I should had explored even more was deciding the movement sets and the possibility to add skills for each piece. That would enhance the game even more and add a high level of design to it.

Finally, a strategy that I decided to use was to conclude some milestones of the game and as soon as new material was added, design or improve some extra features. I decided to do that because I was afraid that I wouldn't be able to create everything

that was originally designed, so by taking small steps while the project was progressing seemed to be a good strategy by that time.

Clearly a mistake because that resulted in some difficulty to track my time available and the necessary timeline as new material was being added after each milestone completed. For that reason, I will pay more attention in future project to avoid this type of mistake and again, dedicate more time with the design phase.

Project Timeline

I really don't believe that I have failed with my project timeline in the beginning of the project, however due the academic semester, I have indeed missed the opportunity to accomplish the expected results and I have failed to dedicate more hours to the project.

The lesson learned is that I should have divided assignments and the game into different projects and added an exclusively timeline for each one. Instead of completely stopping one, would be better to add small tasks for the game project for example, while dedicating more energy and hours to finish the assignments.

This way I believe that I would at least be able to visualize future timeline issues with my game project and would have some time to adjust myself in order to achieve the expected results. For my next experience, I will extremely pay attention to my timeline and stablish a better plan.

Project conclusion

This project helped me to stablish the next goals in my career as a Game Developer. As my final project for the college, the same allowed me to discover gaps in my knowledge; the skills that I need to improve; the ones that I am lacking off and new skills that I should explore in the future.

Like for example in terms of coding I still have a lot to improve and I am even considering to study in deep C# in order to get better results with my programming skills. Also, improving such skill, would allow me to get faster solutions when facing some issues along the project.

After all my efforts, time and energy spent with “LOTR – Tactics”, I am happy with the results achieved. Off course I wished to get better results with the project overall, especially in terms of game design, coding and mechanics. However, as my first experience in game development I am satisfied and I hope to improve my skills for the next project.

I will soon return to this project and finish some of the extra mechanics that are missing like the Eye Mechanic or the skills, also improve the quality of my code and try to achieve a better result so I can have the same in my personal portfolio. Also, I will implement properly the GitHub so the same can be presentable.

I can conclude that this game was a big opportunity and at the same time a challenge for myself. I had to evolve in different ways and at the same time adapt to my lack of knowledge or poor skills.

After all, I am happy with the results but I can see different areas and strategies that need to be improved in order to achieve my career in games design. But such experience allowed me to explore different situations that I wouldn’t imagine while studying about game design itself, so I am really gratefully for the opportunity.

To finish I would like to mention a phrase from The Lord of the Rings, which describes the feeling that I have right now after successfully designing my first game and the path that I want to follow in this career as a game designer:

“It’s a dangerous business, Frodo, going out your door. You step onto the road, and if you don’t keep your feet, there’s no knowing where you might be swept off to.”

J.R.R Tolkien, The Lord of the Rings.

Thanks,
Evandro

References

Batchelor, J. (2020, 12 22). *GamesIndustry.biz presents… The Year in Numbers 2020*. Retrieved from GamesIndustry.biz: https://www.gamesindustry.biz/articles/2020-12-21-gamesindustry-biz-presents-the-year-in-numbers-2020

Escandon, R. (2020, May 12). *The Film Industry Made A Record-Breaking $100 Billion Last Year*. Retrieved from Forbes: https://www.forbes.com/sites/rosaescandon/2020/03/12/the-film-industry-made-a-record-breaking-100-billion-last-year/?sh=2e33639f34cd

Gardner, M. (2020, September 19). *Report: Gaming Industry Value To Rise 30%– With Thanks To Microtransactions*. Retrieved from Forbes: https://www.forbes.com/sites/mattgardner1/2020/09/19/gaming-industry-value-200-billion-fortnite-microtransactions/?sh=45a342922bb4

Photon. (2021, June 6). Retrieved from Photon Engine: https://www.photonengine.com/pun

Tolkien, J. R. (1968). The Lord of The Rings. In J. R. Tolkien, *The Lord of The Rings - The Fellowship of the Ring.* Allen & Unwin.