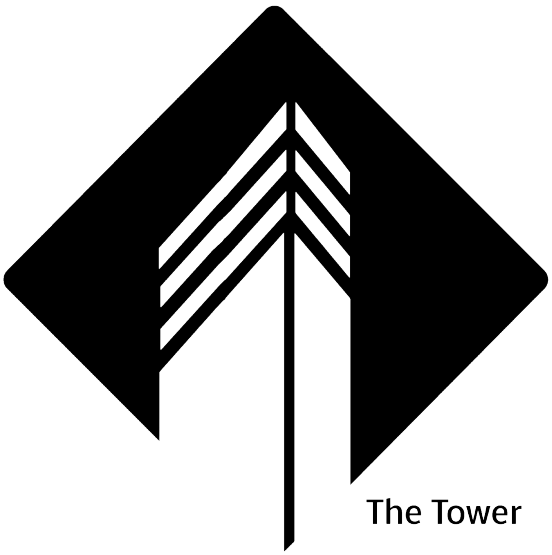
EPITA

Report

The Tower



Wet Water

Guillaume BAILLY

Thien Phuoc LE NGOC

Quoc Duong NGUYEN

Johan PARDO

supervised by

Christophe Boullay

Contents

1 Book of Specification

1.1 Introduction

This book of specifications will present our computer project for 2018 at EPITA. The main goal of this second semester is to provide a working game that will allow the best experience possible. This project will last 6 months during our second semester and will be an important part of it. The teams had to be made before the 6th of December 2017 and so our team Wet Water was made, it consists of Guillaume BAILLY, Thien Phuoc LE NGOC, Quoc Duong NGUYEN and Johan PARDO.

The Tower will be a puzzle game, in which the player has to complete a level in a limited amount of time. The player completing the level will be able to have basic game mechanics, such as running and jumping but he will also gain some abilities to be able to overcome some obstacles. In the multiplayer mode, both players will control their characters through a third person view but one of them will have at his disposal some traps or some kind of objects to prevent the other player from reaching the end of the level.

We want to make the game, visually speaking, simple yet appealing to the eye. We will make 3D models and animations if needed, using the software Blender. We will be using Unity as our game engine in order to develop our game, so the scripts will be written in C#.

The Tower will be inspired from the games we play and that we enjoy, like *Portal*. Indeed, our game will be set in a futuristic world and our levels will be test chambers, that the player has to complete. The game will be challenging but not too difficult and more importantly, will be fun to play.

In this book of specifications we will be describing you the making of the project but also all the tools at our disposition to achieve our goal. We will also be talking about the distribution of the tasks in our group as well as the cost of the project.

1.2 Origin and type of project

Straight after the presentations from the second year students, we could see the difference between the best and the worst projects. Puzzle games were really well made and attracted our attention straight away. They were visually simple and we believed we could do the same with our game, simple and yet really fun to play. Luckily, we agreed to make a puzzle game as we all liked the same game the most, a character that moved and had to arrive to his goal dodging cannonballs, jumping onto objects, opening traps and doing many others actions. So we decided to do our own research and look at other games which we had heard about to see what was doable and important in those kinds of games. As we looked we found a lot of games we had heard about or even played ourselves, it allowed us to see the requirements to make a good puzzle game. Seeing that we already had a lot of knowledge on this type of games we decided to fully commit to this idea and started thinking about what we would actually do in our game and which aspects of it would make it original.

Our game is inspired by mixtures of bits and pieces from games we enjoy playing, as well as books we read and series we watched. We decided to name our game The Tower, which comes from the anime *Sword Art Online*, in which a group of people has to complete all the levels of the tower in order to beat the game.
Unlike the game *Portal*, we will not be using portals as the main feature, but multiple abilities that the player can use to beat the levels.

1.2.1 Story

The game revolves around a scientist living in a mysterious Tower hidden from civilization. You are a journalist who received a special invitation to discover the facility. The scientist living in the tower is rumored to be illegally testing drugs

on innocent people. Your career is in danger so you want to be featured on the headlines. You want to find out if the rumors are true. The aim of this game is to go through all the floors of the Tower in order to escape and possibly meet the scientist and get the interview that will change your career forever.

1.2.2 Gameplay

Every level is filled with many obstacles or an AI that will prevent you from reaching the end of the level. In order to beat the level, you must get to the portal that will teleport you to the next one. You will be able to unlock powers by playing the first stages of game in order to overcome these obstacles such as turrets, traps, and deadly water. As you go up the tower, the difficulty will increase.
There will also be a 2-players mode where the second player will have to prevent the other player from completing the level by setting up traps or by throwing objects from above. Each player gets a turn for each role. The winner will be the player with the best time.

1.2.3 Ideas for powers

In our game, powers can be either beneficial or harmful and the player will have to experience them in order to know its effects. This is the list of powers that we may or may not implement in the game.

- Swiftness: Allows the player to run faster and on any liquid, allowing him to get better times at levels.
- Jump Boost: Gives the ability to jump higher to evade holes, jump on top of an unreachable place and do many tasks that would not be possible to overcome before acquiring the power.
- Mini: Makes the player much smaller and given the circumstances, being small can help to get through obstacles.

- Telekinesis: Allows the player to move objects.
- Cloak: Player becomes invisible to the AI evading a lot of danger.
- Clone: Allows the player to keep its body after death and creating a new one allowing the player to use its old body as weight to press buttons, to walk on in case the floor is dangerous or even to activate traps.
- Heavy Weight: The player will become as heavy as any kind of metal and will be unable to jump very high.
- Drunk: The player's controls get inverted and his vision becomes blurry.
- Head Explosion: The player's head explodes.

Powers will be able to overlap. For instance, jump boost and swiftness will allow the player to go much further than usual, evading huge obstacles. Each ability will have a limited amount of use which can be recharged over time.

1.3 Object of study

1.3.1 Goals and interests of the project

Goals

The main goal of this project, according to the Computer Project File, is to know how to develop a program from ground zero using multiple softwares. But this is the tip of the iceberg because when we develop a game or a software in a group, you cannot just write code without letting others understand your code. The main goal of this project is to combine our knowledge in various subjects depending on our strengths and weaknesses. Most importantly, it will teach us how to communicate with each other which will prepare us to the active world where most projects are done collectively.

Interests

Everybody in our group spends a lot of time playing games from MOBA to FPS (not during exams weeks), but one genre of game that we all enjoy is puzzle games. With our imagination, we will try to create the best game in this domain according to us, and even if there are imperfections, we will have learned a lot of things about game development.

1.3.2 Members

Guillaume Bailly: I come from the S-SVT section of my high school. I had not done much programing before coming to EPITA but the subject really intrigued me as my father loved programming as a child. I really want lo learn the most I can through this project and am ready to work hard to reach that goal. I have worked on different team based projects such as the TPE in Terminale as well as many other year long projects in middle school in Australia, in which teamwork is

really encouraged. Each project ended well, there were ups and downs but the final result has always been what we were looking for. I think that with a motivated group and good ideas, our team can lead to a very good project.

Quoc Duong Nguyen: I am the project leader. I come from the S section of my high school, where I chose the SI option. I had the opportunity to work on two group projects, both with Thien Phuoc, including one in which I could learned about Arduino, a microcontroller that allows people to connect and manage hardware, and the bare minimum of the C language that was required to work on Arduino projects. I had not had any other programming projects but look forward to learn a lot about game design and programming, and improve in that regard. Our group is motivated and it will certainly help in the process of making the game.

Thien Phuoc LE NGOC : I come from a S-SI section in my high school that gave me the opportunity to participate in a project where I had to learn the basis of the C language to help Quoc Duong in the project. Thanks to this project I was able to use and learn some programming language. Outside of that project I did not really use that language but still remember what I learned. I really am looking forward to this project because I have always been interested in video games. With this project I want to discover and learn about game programming. Even if I am still a complete rookie in computer science I hope to learn and improve a lot through this project. The four members of our group are very motivated and even if it has only been 3 weeks since the creation of our group, we are already determined to make a good game out of these 6 months.

Johan PARDO: Last year I was at Leonard De Vinci high school in Antibes where I was in the SI section with advanced courses in Math. Before then, I had already started programming on various supports, such as Python with the Raspberry Pi for Physical Programming project, HTML, CSS and JavaScript in Seconde where I updated our school (FIS of Hong Kong) website panoramic slide from Java to JavaScript because of all the securities problem related to this language. During this year I also helped fix a security issue on an early version of AGORA website (Alliance GÃ©nÃ©rations Orientation RÃ©seau AEFE). And finally in C on Arduino during last year's project (TPE), we developed shoes that could produce energy by walking.

1.4 State of the art

1.4.1 Puzzle games history

Puzzle video games originates from puzzles throughout human history. Physics puzzle games are not something new. A physics game is a type of puzzle video

game wherein the player must use the game physics to complete each puzzle. Physics games use realistic physics to make games more challenging. Popular games of this genre include The Incredible Machine, Portal and Tetris. The latter has revolutionized and popularized the genre which was inspired by a traditional puzzle game, Pentomino, in which the player has to arrange falling blocks to end up with lines without any gaps.

1.4.2 Main inspirations

Gameplay

- *Portal series* is a series of first-person puzzle-platform video games centered on a woman forced to undergo a series of tests within the Aperture Science. Each test involves using the "portal gun", that creates a human-sized wormhole-like connection between nearly any two flat surfaces. The player-character or objects in the game world may move through portals without losing any momentum. This allows complex moves to be used to cross wide gaps or perform other feats to reach the exit for each test chamber. A number of other mechanics, such as lasers, light bridges, tractor funnels, and turrets, exist to aid or hinder the player's goal to reach the exit.

- *BattleBlock Theater* is a comedy platform game. Controls are simple, with the game largely consisting of running, jumping and punching. Levels are made up of various types of blocks, such as collapsing blocks, sticky walls, bouncy volcanic rocks and deadly spikes, and also feature hazards such as water and deadly creatures. The main mode in the game is Adventure Mode, which can be played solo or co-operatively with a second player. In this co-operative mode players are able to work together to overcome obstacles, which includes throwing them across gaps, using their heads as platforms and helping each other up ledges, although

there is just as much opportunity of partner sabotage.

- *BioShock Infinite* is a first-person shooter video game. In this game we play as Booker and have the ability to acquire different powers that are gained by finding items throughout the game.

Setting

- *Tower of God* is a korean webtoon. This webtoon centers around a boy called Twenty-Fifth Bam, who has spent most of his life trapped beneath a mysterious Tower, with only his close friend, Rachel. When Rachel enters the Tower, Bam manages to open a door into it as well, and faces challenges at each floor of this tower as he tries to find his closest and only friend.

- *Sword Art Online/SAO* is a Japanese light novel series and was adapted into an anime. The series takes place in the near future and focuses on various virtual reality MMORPG worlds. In this light novel there is something similar to a tower called "Aincrad", Aincrad is an iron-and-stone made floating castle that consists of a hundred floors stacking straight upwards, meaning that each floor diameter is a little smaller than the previous one. Only one stairway links each floor to another, and the stairways exist beyond a boss room of each Labyrinth.

Voice Over

- *The Stanley Parable* is fiction video game. The game is presented to the player from the first-person perspective. The player can move around and interact with certain elements of the environment, such as pressing buttons or opening doors, but has no other controls. The story is primarily presented to the player via the voiceover of the game narrator.

-*Getting Over It* is a platformer video game. It revolves around a man wielding a hammer to grip objects and move himself with. The game is accompanied by voice-over commentary. The commentary provides quotations relating to disappointment and perseverance when significant progress is lost by the player.

1.5 Part of the project

We decided that everyone must try at least every aspect of the development of this game in order to not get stuck on one job and to be able to help each other and focus on different aspects according to its difficulty.

1.5.1 Task

Taches	**Guillaume**	**Thien Phuoc**	**Quoc Duong**	**Johan**
3D Modeling	-	S	-	**M**
Animation	-	S	-	**M**
Effects	**M**	-	S	-
Level design	S	**M**		-
Level build	**M**	S	-	-
Sound design	-	S	**M**	-
Game design	-	S	**M**	-
Website & Hosting	S	-	-	**M**
Artificial Intelligence	-	-	**M**	S
Multi-player Mode	**M**	-	-	S
Network	-	**M**	S	-

M : Main S : Support

3D Modeling: This part consists in developing objects to be implemented in the game, ranging from the entire map to small details.

Animation: Animation is everything that is related to the cut-scenes such as how the character behaves when moving, falling, running or any kind of movements.

: Special effects will give the atmosphere of the game, and therefore make it more visually appealing. For that part will be using things like water effects, to dust particles.

Level Design: Level Design is all about creating a possible level and making its blueprints.

Level Build: It is the part in which the level designed previously is implemented in the game.

Sound Design: Sound design consists in the choice of musics, the voice acting and other sound effects of the game which will make the game much more engaging.

Game Design And Game-play: As the name suggests this is all related to how the game will behave according to the various inputs of the player.

Website: The website will be the link between people, potential buyers, in which they are aware of the game progress, updates, future goals and all important news.

Artificial Intelligence: Te Artificial Intelligence will have a significant impact in our game because it will decide when to action traps, change the sound and will hence have an important role in this game realism in term of "smoothness".

Multi-player: This step is all about grouping the techniques allowing the game to execute for two or more players.

Network: Since this game will have multi-player we need to have a network to manage all the collisions of objects and the player's interactions.

1.5.2 Task Advancement through Time

Task	Defense 1	Defense 2	Defense 3
3D Modeling	50%	90%	100%
Animation	50%	90%	100%
Effects	0%	60%	100%
Level design	30%	70%	100%
Level build	20%	70%	100%
Sound design	0%	80%	100%
Game design	40%	90%	100%
Website & Hosting	25%	50%	100%
Artificial Intelligence	0%	60%	100%
Multi-player Mode	30%	70%	100%
Network	30%	70%	100%

1.6 Functional

1.6.1 Objectives

In order to carry out our project, we planned to fulfill some requirements that seem essential to the success of the game:

1. Menu that allows the player to choose between single and multi-player mode
2. GUI (Graphical User Interface) that shows the items that the player can use as well as subtitles for the voice over
3. Different powers that can either be benefice or harmful to the player
4. Animations for the player's movements and abilities

5. Network to allow the multi-player
6. AI that manages traps, turrets, obstacles, spawns and checkpoints
7. Level design that will define the difficulty and the pace of each level. It will also be adapted according to the different abilities
8. Scoreboard which will show the top scores on our website
9. Voice Over will set the story through the scientist's voice that will be voiced by one of our members

1.6.2 Further development area

- Bonuses might be given to the player as he advances through the game. Depending on his ability to find easter eggs or not, the player will gain new functionalities in game. These functionalities will not be major but will allow the player to have more freedom as he will gain the ability to change small in-game features, such as:

1. The language, which could be changed to French
2. Achievements, that you can unlock by completing different challenges

- A tutorial might be implemented to showcase the different aspects of the game (controls, abilities, obstacles).

1.7 Technological and methodological

This Project will be using multiple softwares:

1. Unity, which according to Wikipedia "is a cross-platform game engine developed by Unity Technologies, which is primarily used to develop both three-dimensional and two-dimensional video games and simulations for computers, consoles, and mobile devices." This software will be the bedrock of our

game because it greatly simplifies the use of particle and effect with all its libraries at our disposition.

2. Blender is a open source 3D creation software. With its large and active community this software will be easy to learn thanks to multiple online tutorial. Its complexity will allow us to make any figures that we want from big objects like maps, to small details, for example the hands of our characters.

3. LaTeX is a document preparation system which will be used to write our Book of Specifications and all of our project reports. We will also use Overleaf, we needed an online software which would enable us to write on our common LaTeX files without the need to be together.

4. Apache is a software which will be used to simplify the development and maintenance of HTTP servers and so will be of great help for Website Hosting.

We will keep this part up to date according to the various needs that we will encounter throughout our project.

1.8 Operational

1.8.1 Expenses

Because this project's main goal is not to make money but to learn how game development works, there is no cost related to marketing. The softwares that we are going to use are either open-source or free for tinkering so there are no expenses in this area. As for the website one of our member has a server and a domain available. The only cost of this project will be the game box, the guide and the CD.

1.8.2 Market

This game, because of its fairly simple design but highly intellectual game-play, will attract mostly young adults which, according to statistics, represent 27% of all sales in the gaming industry in 2017 in the United States.

2 Book Of Specifications Advancement

Task	**Defense 1**	**Defense 2**	**Defense 3**
3D Modeling	40%	80%	100%
Animation	40%	80%	100%
Effects	0%	60%	100%
Level design	30%	70%	100%
Level build	20%	70%	100%
Sound design	0%	70%	100%
Game design	30%	80%	100%
Website & Hosting	25%	50%	100%
Artificial Intelligence	0%	50%	100%
Multi-player Mode	30%	70%	100%
Network	30%	70%	100%

3 State By Parts

3.1 3D Modeling

For this second part I remade every models to add their articulations (knees, elbows) to have a player that look more alive than before.

Nero:

Nero, our main character, is composed of two parts, the body and the head. For the body it is just composed of a one cylinder with two semicircles on each side and finally two arms and two legs which are made of smaller main bodies but now they have articulation. For the head we had the idea to have something which looks like a falling water drop because Nero means water in Greek.

Fotia:

Fotia, the nemesis, have the same body and head (in red) as Nero except that he has a jet-pack. For modeling the jet-pack, I used a cube as the frame and a cylinder at the top to have a smooth top and two cones at the bottom for his reactor. I did the same thing as Nero with his arms and legs and his articulation.

Drone:

It is just a simple drone which is inspired by Doctor Who's creature called the Toclafane. It looks like a ball with spikes coming out from the bottom.

Power Up:

For this item, we first wanted to do something unique for each power but we thought that if the player could distinguish good ones from the bad ones, he would never choose the bad ones. For the design we went with something classic, a simple crate.

The Scientific:

The scientist has the same body as Nero except that he has hair inspired from Doctor Emmet Brown in Back to the future. That's why he has this type of hair. Another stereotype of Doctor that we add is glasses, So we put 2 white cylinder to mimics them.

3.2 Animation

Nero:

This character moves by walking so I made some animations required for this environment, so walking and jumping. But also the dying animation. Because he is made of water, the animation behave the same way as a snowman melting very

quickly. But after the first defense I realized that when the character is not moving he looks like a statue so I did an Idle animation as well.

Fotia:

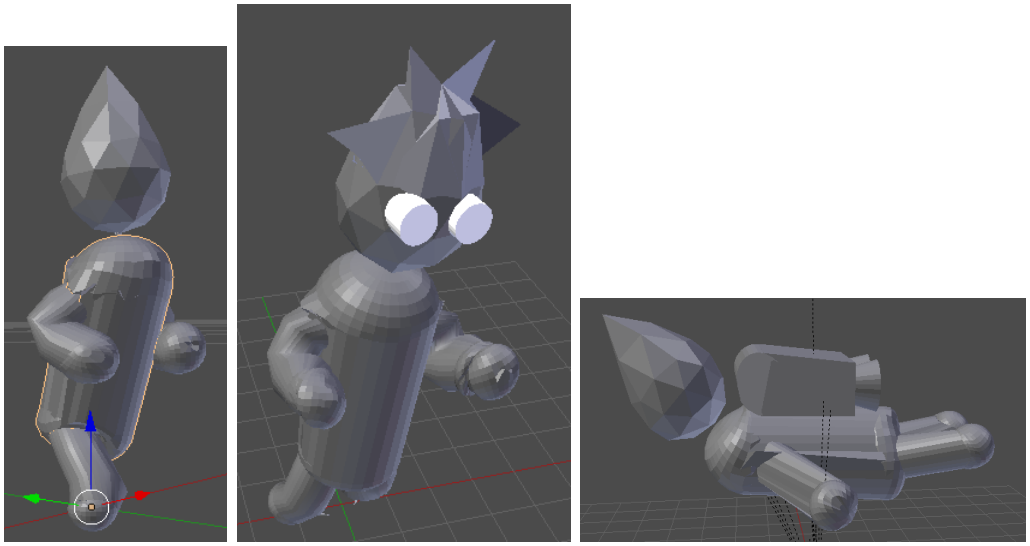
Fotia being the opposite of Nero, is made of fire so his head is burning the same way as Ghost Rider, and his jet-pack also has an animation, the engines. But they are not yet implemented in the game. But I did a flying animation that is implemented in the game. I also added a first step towards AI by making him shoot towards Nero but for now it is just cube but in the end it will be fireballs but we will comeback in details in the Artificial Intelligence's part.

Drone:

The only animation that I had for the drone is that his spikes are just going forth and back from his body. And now it is implemented in the game. He is hovering over the main character

The Scientific:

This character is only there to explain the story to the player so he doesn't need to move so for the animation there is only a "talking animation".



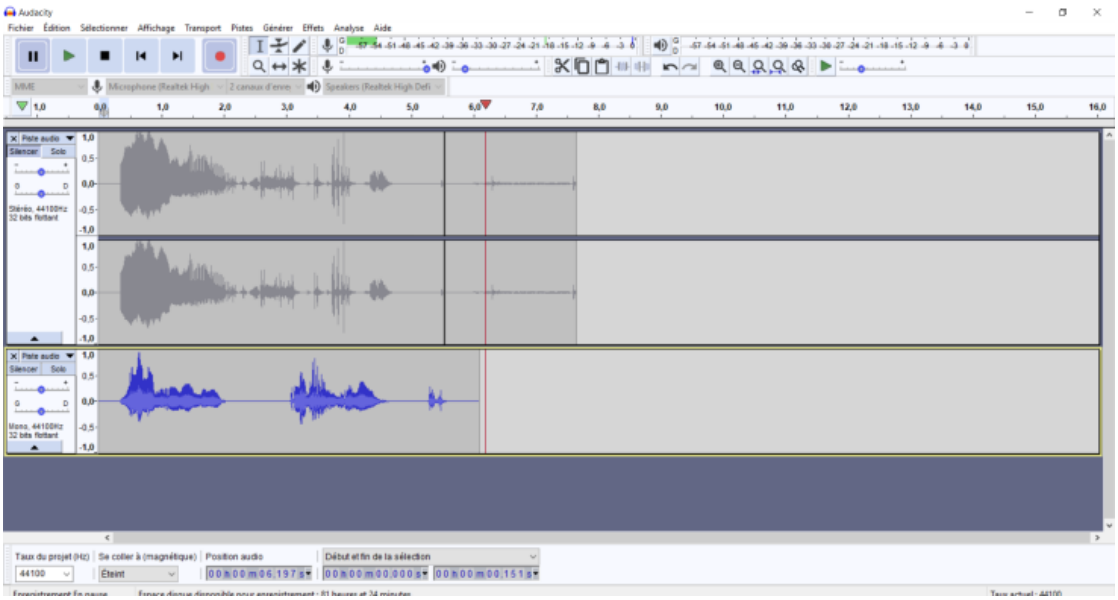
3.3 Effects

In the game we just added some simple effects. The effects we added are the ones for the bullets and fireballs when these touch a player or when they collide with anything else.

3.4 Sound design

3.4.1 Voice Acting

We took time to record our voices to tell the story of our game and recorded a lot of lines the scientist would say when death occurred to our character. These sound effects had for goal to make the game interact with the player by taunting him mainly as if the scientist was actually talking to the player. We used Audacity to record what we needed, it was very useful as it allows you to make a lot of recordings in just one file. A introduction to the game was recorded as well as the end where you meet the mad scientist in person. This was done to really provide the player with a story and to immerse him in the lore.



3.4.2 Music

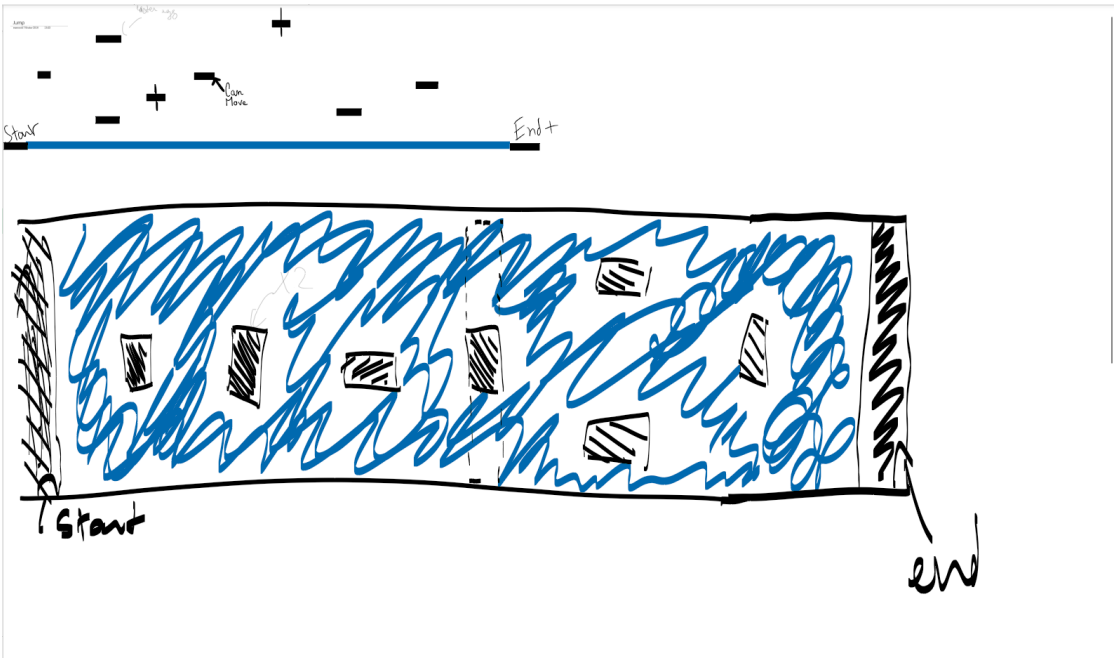
We decided to implement some music in the background to give the player a more enjoyable experience. These music help to set the atmosphere while the player goes through the levels.

Choosing the music for our game was quite difficult because it had to be energetic and fun and without being to overwhelming. It had to fit the pace of the game.

3.5 Level Design

3.5.1 The levels

The level design is not the most difficult part of the project. At first designing the levels was done mostly by drawing the maps in 2D first and then deep drawing them to be easy to understand and to make it on Unity. We decided to draw the maps on an online notepad. Thanks to that tool, sharing the maps was a lot easier than drawing them on paper and exchanging it. We were able to draw it directly on our computer without losing it as we would with papers.



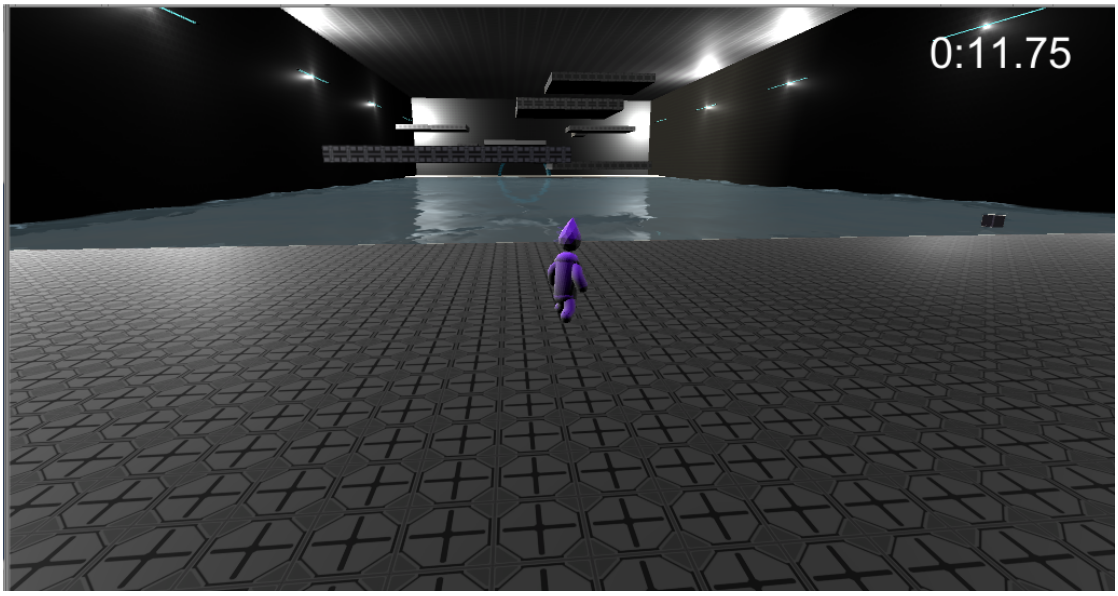
The main thing about level designing is to think about how we will make them look like. As you know, this game is a platform game so we have to be creative about the where to place the obstacles so that the player will have to use the right power at the right time. In addition we can also make obstacles that will make the player obliged to mix some powers to overcome the new obstacles and discovering some new gameplay.

We also decided that during the project the maps would be built in an increasing difficulty kind of way indeed, first of only tutorials where made and levels were what followed theses tutorials. But difficulty doesn't always give you a sense of evolution so as in most puzzle games like ours, the player finds new objects or powers as he progresses through the game.

3.5.2 Decoration

Then comes the easiest part of level designing, decorating the levels. When it comes to decorating the levels, we only need some props to make the maps more filled. Levels without any props would feel too empty and less enjoyable when we

play those maps. Every single prop that is integrated in the map has a role to play. For example, the lights are not there just for the map to be filled but they have to lit the map and to fit the overall atmosphere of the game. Meaning that they have to be a little intense but in a short range for the map to be a little dark.



Another example would be the screen at the beginning and the end of a level. they are not turned off but actually broken since our game is resolved around a tower where a mad scientist test drugs. As in portal where the screens give information about the actual level, in our game they are supposed to be broken.

3.6 Level Build

Level build is an essential part of a game, and it takes a lot of time to say the least. Even more so when you have to learn how to operate on Unity. To begin with this part of the project, I watched a few tutorials showing how to handle Unity and build levels using very useful tools such as ProBuilder, which is very straightforward to handle. I had a few issues as the coordinates of objects, which kept changing for no reason at first but after reinstalling Unity it all worked perfectly.

I learned to use the coordinates even more as the project advanced and started building bigger maps. As we had decided in the level design, the player would work his way trough the levels and discover new in-game content which will push him to keep on playing. First of we decided to make a tutorial for each power available in-game in single player and a tutorial in multi-player mod. Each tutorial had to present the boosts available in game but also debuffs: Heavy Weight or Drunk. Then we also decided to add a few maps featuring turrets, all our powers and movable platforms. So with the help of the drawings made in the level designing part of the game we made each map, they were made in accordance to a specific boost, which is assigned to a pickup box which is floating in the air:

Jump: This tutorial is based around the jumping boost and is pretty standard, the user has to jump from platform to platform to get to the arch at the end of the map present in every single map. But he has to be careful since falling in the water under the platforms mean death.

Speed: Based on the speed boost this tutorial is just like a very simple small maze which would take a while to finish without the speed boost. The pathways are long and in straight lines which is perfect to show the speed increase power as it will be very useful for players to get better times in the games at a later stage.

Mini: This map is only achievable if you are small enough. It uses small paths that can only be entered if the character is very small.

Telekinesis: It enables the player to move certain objects. In this map, the player will have to move objects using telekinesis to jump over walls or activate a door for instance. This power will always be active and the player will not be required to take any power cube's to gain it.

Walk on water: An uncommon power in our game, it allows the player to walk on water, the latter being deadly for our character, it is truly lifesaving in some situations. The map is very simple the only way to cross to the other side is by walking on water by taking the power, otherwise only death awaits.

Level 1: This map takes powers to a whole new level as it combines every single power presented in the tutorials as well as a debuff. This allows the character to have a huge and overpowered power that will allow him to achieve things that would seem impossible.

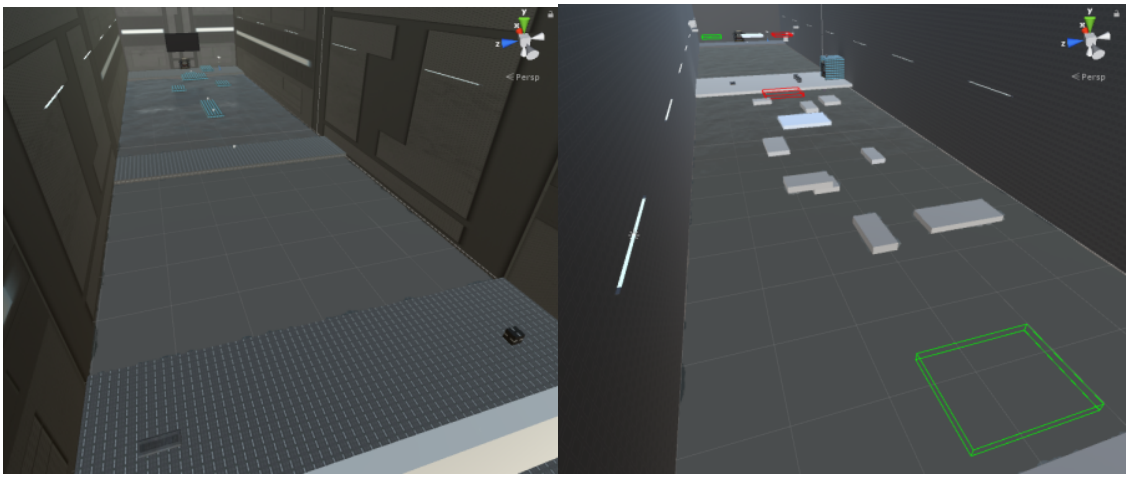
Level 2: Turrets are new to the game and are also the first enemy our character encounters, the player has to care for bullets while playing this map which makes it so much more dangerous.

Level 3: Moving in a hallway isn't always that fun so this level was made differently compared to all the previous maps, the player will have to climb up to reach his objective. The jumps in this map are much harder than in all the previous maps and only players who have mastered their jumping skills will be able to achieve it without difficulty.

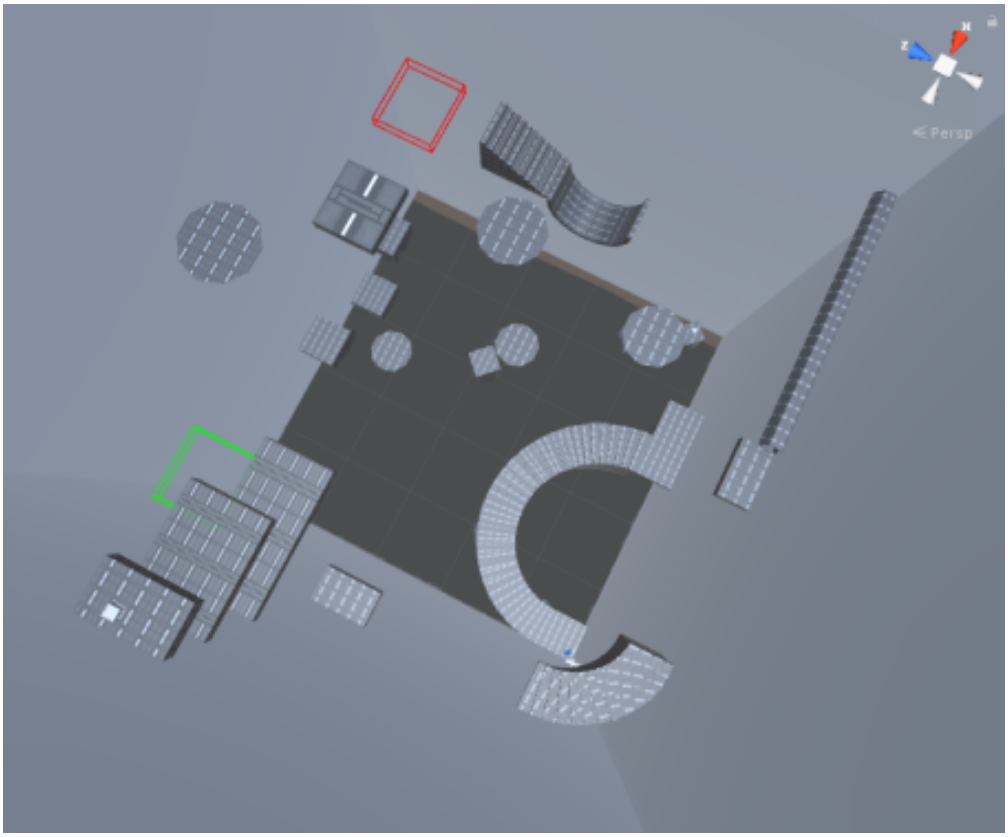
Level 4: In this level the player has to face a big number of turrets, the player will have to get to the other side of the map without getting hit by any bullet. He is required to move fast as bullets will not slow down. The player will need to have good dodging skills to finish this level as with more turrets come a lot more danger.

Level 5: For this final level, the player will be trying to survive from a number of lasers. Indeed there is a new trap: the laser will make a lot of appearances in this final fight for survival, you will need to get to the other side of the map while facing every single powers and traps found during the game.

Map 1, Map 2:



Map 3:



3.7 Game Design

Powers :

- Speed: This power allows the player to move faster throughout the map. It simply needed an increase of the force in the direction in which the character is walking towards.

- Jump: This power allows the player to jump higher throughout the map. The Unity physics engine handled the gravity of every object in the scene which simplifies all the work, so this power is as simple as the Speed power. It required an increase of the vertical force put on the character.

- Telekinesis: This power is as simple as the other powers. It consists of having the ability to pick up a certain cub, the black cube, which is the only object that can activate doors with pressure plates.

- Mini: This power reduces the size of the player, allowing him to access different parts of the map that the player couldn't reach with its normal size.

- Walk on water: This power allows the player to walk on deadly water. The water object needed to know if the player acquired the power beforehand.

Moving platforms have been added to make the map more challenging to play. These are implemented in the Jump map.

Player's movement:
For this part, I assigned a movement for every input. For example, if the W key, the player moves forward, S moves backward, and so on. That part was handled through Unity's input manager. One of the main difficulties of this part was to have a movement that didn't seem too fast or too slow so the player doesn't get bored.

Camera:
This part might be one of the most complex part, as most of the time spent on it was debugging. The camera consists of doing the same thing as the player's movement except for the mouse but because the camera doesn't have the standard Cartesian coordinates but relative coordinates so I needed to create a function that calculate the proper coordinates from the relatives one. The first camera implemented was very easy to implement, it was fixed to one position relative to the

player. The difficult part was to make the movement independent to the camera which means that the character moved with the keyboard and the camera moved with the mouse, as well as making the character model face one way without the camera moving. There are also boundaries to the camera so that it doesn't flip and invert the controls.

Re-spawn:

First of all, I implemented a health system to the player, you can set a certain amount of health to a player, as well as set the damage output of an object easily. As of now, the player only has one health and the deadly objects gives on damage. We will be able to modify these values very easily when there is more complex obstacles or enemies.

When the player touches deadly objects, such as water, the player object is destroyed and is re-spawned to the original position. There is a short black transition when the character goes from the dying position to the original position. Afterwards, a flashing animation gives a visual feedback which gives the player invincibility for a short moment after spawning. The only object that can affect our player so far is the water object, so this re-spawn system is only on maps where there is water objects.

UI:

At first, we made a simple menu allowing the player to choose either play, options which as yet to be implemented with settings such as volume and quit which exited the game. Inside play we wanted the player to be able to choose between single player and multi-player mode in which their is a tutorial button linked to the available scenes and a level button which is empty for now. We then added a back button so that the player could go back on his choices. This at first took quite

a while to make as we had to get used to the way Unity worked but fortunately Unity UI management makes it simpler and neater.

We also decided to make an in-game pause menu. It was similar to the making the main menu so it wasn't really hard to get used to. The scene is frozen in the background when you pause the game.

For the in-game UI, I added icons for each power. When you take a certain power pickup, the icon will pop up on the top left. There is also a timer on the top right which counts how long the player takes to get through the map. It stops when the player gets to the zone at the end of the level and invites the player either to quit or get to the menu. The timer also freezes when the pause menu is brought up.

Finally, I added in the main menu more settings, such as the resolution, the quality which goes from medium to ultra, and a toggle for the full-screen mode. As our game should run on most computers in its state, it is by default on ultra.

3.8 Website and Hosting

3.8.1 Website

For the website itself, following the research of various style of templates, we chose to go with one that has a minimalist design but most importantly, only has one page. One page means that you only need to load the page one time and you have all the informations about the game and so it does not stress the server, that is only a micro-computer.

For now here is the templates that we chose of our website

Templates:

-About The Game :

Gives the description of our game.

Gives the lore of our game.

-The Team:

List of the members with their contacts and their roles.

-Advancement of our game:

Gives the percentages of the progress of each of the part in the game.

-Download:

Section where you can download an executable to launch the game.

-Contact:

All of the contacts needed are in this section. From here you will be able to contact us if any problem arise.

3.8.2 Hosting

After some research on how to host a website, we opted to use a Raspberry Pi as our server. First because we had no expenses in this project and because of his small form factor it can easily be installed in a matter of seconds and it is really easy to update our website compared to third-party server.
Then for the software I used apache HTTP server, an open-source HTTP server. I also added a git repository so that when we needed to update the website we could first do it on our own computer then we could push it. After we would pull on the raspberry pi via ssh and then move the Website folder in the Apache folder. So basically we can update our website anywhere
Finally to have a proper website we needed to have a proper URL so all we did was get this address wetwater.ml for free and we linked it to the server IP. The last

step was to port-forward the server so we went to the box setting and activated the DMZ setting. You can take a look on our website on http://wetwater.ml/main/

3.9 Artificial Intelligence

3.9.1 Fotia

The first one is Fotia which is hovering over Nero in a predefined area which is relative to the Nero so when he is moving Fotia is also moving to stay in this space. Then for his movement a just assigned that he must be above him of 5 (unit in Unity) but he can go anywhere in the X and Z axis by adding randomly generated coordinates within range of the predefined area that we have said before.
The second one is the firing Intelligence. Every 5 seconds Fotia is "firing" a sphere, it spawns around Fotia and land in a area close to Nero. Then we applied some gravity to the fireball, it falls and can touch the player but with the speed at which the sphere go, the player can easily dodge it .And after 5 seconds the fireball disappears. Since it is fired by Fotia we don't have to worry about the position of the bullets. Since it is in a range around Fotia it will always land near our main character.
After that we made some modifications to that fireball, we added a script that allows it to hurt the player and kill him if the player collide with the sphere. As long as the fireball touches something it will disappear but if it touches the player then he dies.

3.9.2 Turrets

Along with Fotia, we made an automatic turret. A fairly simple turret that fires bullets when a GameObject with the tag player enters the range predefined for the turret. We can also modify freely the fire rate of the turret, these bullets will always follow, or seek the player and try to kill him. But after some time (around

3 seconds), the bullets disappear or destroy themselves.

As you can see in this image the turret, on the left of the image, is shooting at you (blue character) a red bullet and the red cubes that you see on the right of the player are some fragments of the bullets. In the game theses are considered as particles.

3.10 Multi-player Mode

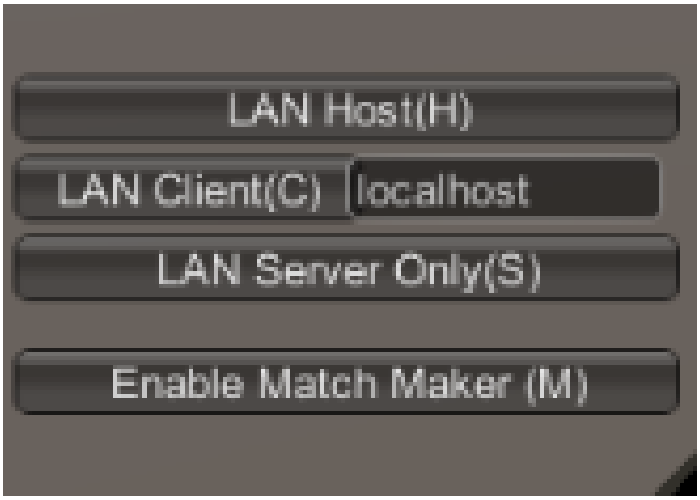
3.10.1 Unity Network

At first Photon Engine seem like a good idea since in the beginning if the year some presentations of game were saying that using it would be easy for us to implement the multi-player mode. But when I actually compared both Photon Engine and Unity Network it seemed like using the built-in network system was easier than the Photon one. Even if the Unity network seemed easier to use we did give Photon Engine a try. A choice had to be made between using the Network already implemented in Unity or use the Photon Engine network that lot of tutorial makers used for making the network for their games. At the end, Unity Network was the best choice because for the network, as our game only uses 2 players so an already implemented module was a lot easier to use than an external one. As

it was said earlier, a lot of tutorial are made for Photon Engine but we managed to find some for Unity Network that helped a lot.

3.10.2 Implementation of the multi-player mode

It was fairly easy to do the multi-player part because Unity is giving us a lot of components that we can use to make the network inside our game. For example, we already have a little GUI for the network that allows us to connect/create a LAN host where you can play in solo.



The second functionality allows you to join the local host that is the first thing you will be able to do but in that space you can enter the wireless LAN IP address of the host and you will be able to connect to the same game as the host and play with him. The separation of the control of 2 player is easily done by a simple script where you create an array with the component of the behaviors of a player (for example the player controller, the camera following the player) and disable them if you are not the local player meaning that all the functionalities linked to a player will be disabled if you did not connect the player with your computer.

3.10.3 Gameplay

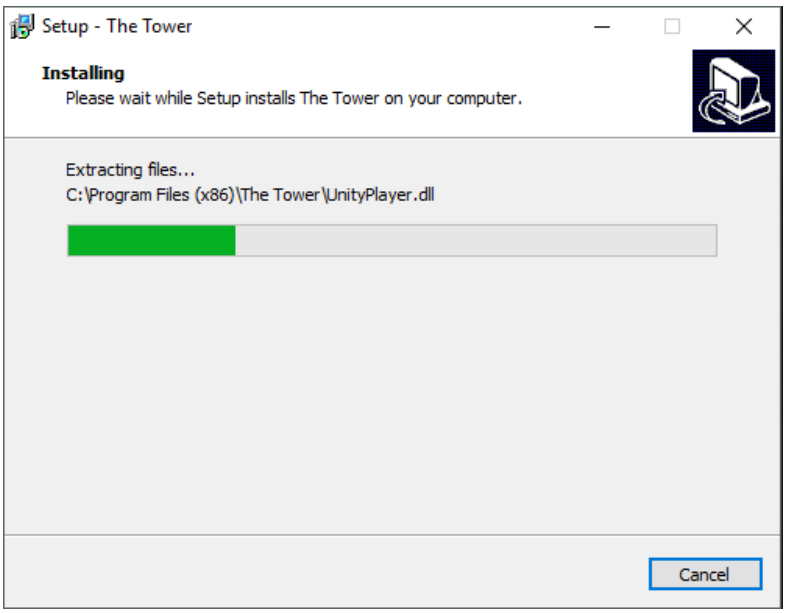
In our game, the multi-player is a new kind of gameplay. Compared to the normal levels, in here, an aspect of competition arises The multi-player mode can be played with 2 players or more. The goal of this mode is for the player to be the first to finish a level. All the participants will play the same character, since the goal is to be the first to finish, we can fire bullets to kill the other player and try to beat the level before the other one. But another difficulty arise. You can only shoot when you are not doing anything else, and as it was said earlier, Fotia will fire some other bullets from above and will try to kill you. You have to be able to shoot fast enough to be able to dodge the bullets fired by Fotia. All of players will have Fotia firing things to them.

Death and Re-spawn:

If a player gets hit by anything that can kill him, he will respawn at the beginning of the level on the coordinates (0, 0, 0). In the multi-player mode we implemented another health system because a new element was added to the player, the health bar. Even if the player only have health for now we can still modify it easily and change it value. Since all players only have health, just getting hit by anything that can do damage, they will be re-spawned. As for the damage given we just have to use a function of the health script of the player on the script of the object we want to give an amount of damage and set it value. Just like the health of the player the amount of damage given by an object can be set easily depending on what kind of enemy or object it is.

3.11 Installation

We used Inno Setup, an open source software used for making installers for Windows programs. The installer of our game is as simple as it gets. You have the possibility of choosing your game directory as well as choosing whether you want a desktop shortcut or not. When the installation finishes, you also have the option to launch the game immediately.



3.12 Manual

We decided to make a complete manual of our game so that the player would not be thrown in new horizons without any guidelines. The manual is divided in four parts:

- Story: tells the story of the game

- Installing and starting up: describes each procedure to install or uninstall the game if required.

- Navigating in the menu: allows the user to learn how to navigate in the menu

- Beginning of the game: teaches the player how to start the game, the menu's functionalities, the assigned keys and even the powers available in game.

4 Personal Experience

4.1 Guillaume BAILLY

I for one loved making the project, it was a unique experience that allowed me to develop a lot of new skills as well as my teamwork. Having never used unity, this experience was new to me and taught me a great deal. While I was not in charge of the biggest codding part of the game I learned a lot of new things that could be done like how you could make powers that can activate on a player. I was the person in charge of making new maps in relation to the powers/traps made, I really enjoy adapting a map difficulty, it was really fun. Through this I learned about a lot of map building tools available on Unity and I learned to use them much faster than before, starting from using shortcuts I really improved my designing skills as I also used coordinates a lot more. In the first defense, I had a really hard time building the maps as the coordinates changed for some reason, I still have not found out why it did that at the time but this pushed me to move the objects manually which looked very bad as there was small gaps in between each walls. I even rebuilt the maps that had this problem as it was very obvious to the eye. The first time I made each map took me a lot of hours but as I redid those maps I found out that the time required to do them was divided by two which shows the huge advancement that I made in map building. Moreover, in the

first defense I had so much issues with GitHub, it was horrible. For some reason some of the texture files downloaded on the Unity asset store where more than 100 MB so GitHub didn't allow me to upload them and every platform which had those textures on disappeared or just lost their color. I had to repair a map at least 3 times because of this. Also someone reverted a change and the main menu I made vanished. We tried getting it back up but could only take back the main frame of the menu as the text and the background had become blank. I think the most important thing is that I now know how GitHub works, well more or less and everyone checks what they push a lot more. Finally, I made my debut as a voice actor which was really weird for me, listening to your own voice can be really weird but I loved to embrace our game's story. Overall I had a fun time doing this project with the best team I could have ever asked for, it taught me so many things that it has become one of the highlights of my first year in EPITA.

4.2 Thien Phuoc LE NGOC

I had to do some level designing at first so by doing this it helped me understand how hard it is to design new maps. Not only we have to be careful to where to put our obstacles but also all the element that have to appear in the game have to be present in the sketch for it to be as understandable as possible for the map builder. Being a map builder is a really difficult task since he had to understand every drawing that I made. Thanks to this project I came to realize how important it is to be really careful and focused while designing the maps. Since I helped for the level build, I have a better overview on how to deal with coordinates now, while for the first defense I was always moving the objects little by little to be able to place them where I wanted them to be. Also as I had to decorate the maps I did play even more with the coordinates. I had to be even more precise about the placement of the props in the maps since they are usually small.

Even if the AI we currently have are fairly simple it helped me to understand how some object can have a range where it identifies whether a player is present or not and how to deal with it. Also I helped in doing the AI of Fotia so I understands how we can create GameObject with some intervals and making deadly for the player, and then if they do not touch the player then for any collision destroying them.

The multi-player mode was the most difficult part for me. At first I had to choose between Unity Network and photon engine since the second year students seem to like photon. But at the end the most convenient one was Unity Network. First of all it was already implemented in Unity, no need to add any extension to our game, secondly, Unity Network gives us a lot functionalities that we could use for our game. Anything that had to do with the communication between the client and the server was done by Unity. Even if it is already implemented into Unity we still have to learn how it interact between each other. It is not because Unity do it all for us that we can think it is the end. If not I would not be able to implement all the accessories that goes with the players like Fotia (the Artificial Intelligence) or the bullet that the player can fire by himself. be cause of those elements, we have to send by ourselves the information to the server and tell him to send it back to all the other players that are present in the server. Not only that but we also have to deal with the death of the players. It is not as simple as in the single player mode since we cannot just destroy the object and spawn it back wherever we want. Since the object "player" is linked to a certain computer in this case i just had to put back the player to beginning the point I set as spawning point.

In the multi-player part I also tried to make an UI / Menu for the connection. At first I just needed to do a custom Network Manager that looks like the implemented one but with more function to be able to connect, load and disconnect the players in the server. It did not seem to work so I decided to keep the implemented UI.

But thanks to this failure I have come to know how to do Menus easily. I think that making menu is quite easy but takes a lot of times since we have to connect a lot of things together like button with certain function that will trigger when the player click on that button.

As for the choice of music, no music that I chose were implemented for the game, but I think that finding music for the game is actually quite relaxing. Finding the music is really difficult since it has to fit the overall atmosphere oh the game. But thanks to that I actually discovered really good music that I could add to my personal play-list.

For GitHub now I am really confident in using it, I understand GitHub Desktop much better than I used to. Managing files to upload and share to the teammates now is really easy even if sometimes we still encounters minor problems it is still a really good platform for sharing our project. I think the only problem on Github Desktop that we really have to be careful about, is to not Push or Pull at the same time as another teammate.Now that I'm used to it, i have to say that GitHub in general is a very powerful tool. Since it helps to share a lot of files that we would not be able to manage if it was on our own. The sole problem is that I think GitHub Desktop should be use for personal project. The fact that wee cannot push and pull at anytime can be a annoying. We did not want to use Unity Collab because we were already into GitHub but it could have been a good idea to use it.

4.3 Quoc Duong NGUYEN

As part of my project, I had to learn how to manage objects and write scripts to make them interact with each other. Unity is a powerful tool that has been designed to be easy to use. Many features of this software facilitate the processing of each part of the project. There is a window for every thing you want to access. I really like the way you can edit objects through scripts, or access values in other

scripts. I had some difficulties getting used to it and many hours were spent debugging very simple things, but many tutorials allowed me to learn it quickly. Now that I know the basics of this software, I know what to do in designing the game, and I will be much faster in implementing new features. As far as the appearance of our game is concerned, the current state is as simple as possible. GitHub is an excellent software that allowed us to monitor every change we made and go back if we had problems. I had trouble understanding it at first, but It was very useful later on. The only problem is that we could not modify our files at that time, so we are considering using Unity Collab which is the solution to this problem. I had a lot of fun learning how to develop a game, but it took me a lot of time and work. I'm really looking forward to continuing to make the game more dynamic with single-player AI as well as implementing the story with our vocal acting game, to make our game more alive. Thanks to the help of my teammate Thien, the artificial intelligence of the turret does what we expected from it. I really enjoyed recording the voices for our game, as well as choosing the music for each level. I was able to learn more about how we could handle audio clips, for example when playing them during a level, for a certain action or even randomize them with the use of scripts. I also implemented power telekinesis which simply allows the player to take a box to open a door with a push button, and thus advance in the game. I found this project to be very interesting experience and might even get me into making a new game of my own in my free time.

4.4 Johan PARDO

Having never used most of the softwares. It was for me a brand new experience, using GitHub, at first it seemed strange why would I used something that assemble every folder that we did. I thought, at first, that using Google Drive was a better choice and we could assemble everything at the end but after a couple of weeks I

understood the power of git and all the benefits of using it in this environment.After a heavy use of GitHub I think I understand how it works not only using the GUI given by GitHub but also using the command line in Linux. I am still not a master but now have some bases in this part.

Then came Blender, the way that this software was implemented was kind of strange with right click to select something and then the amount of informations that you need to learn just to get use to it and then learn all the shortcuts to properly use it. For me learning all those things was long and fastidious. Then for Blender because I redid all the 3D modeling and animation I can easily use this software. What I like about it, is that with simple keyboard shortcuts you can quickly and efficiently do modifications of your character. Another aspect that I like about it is his physics animation (Hair, Fire, Water). Even if it is something that you could think is minor, for me it is something that I really like because it is not just a modeling software but also a software that can be used in CGI for movies and this is a great plus.

And finally Unity, I just used it the final week to implement some of our animations in our game but it is the only part were I could program in C#. First it felt very intuitive from the script to the texture. But something that was frustrating was the import of blender animation to Unity.

5 Conclusion

This game has helped us to tremendously improve our skills both in programming and in group-based projects. This project enriched our knowledge in every aspect of game development. Most importantly, it gave us an inside view of what it is to work in a group in a big project with all its advantages as well as its disadvantages. We know how to manage our time and resources, depending on our strengths and weaknesses.