TOLEDANO Gabriel

MERLE Timothé

PINSON Alexis

REPORT

LockedUp

First Defense

08/03/2022

Table of Content

1. Introduction

Through this report of defense, you will be able to follow our evolution since the beginning of the project. Indeed, since the project was accepted, we have been working on the good realization of it. Thanks to the specifications we already knew where to start and how we were going to achieve it. The only problem could be the time that it was going to take us or the obstacles that we could have met during the development of this one. So, our topic is a game called LockedUp. Originally, we were not very comfortable with the idea of making a game. However, when our first project was rejected, we changed our mind to something that we felt like and that could be interesting to create. That's why we came up with this idea, all of us fonder of doing escape games and more enthusiastic to join Epita, we decided to create an escape game in this school. We first thought of the first scenario, from which two others followed.

The team consists of three members, Gabriel, who is the team leader, Timothé, and Alexis. We didn't have a hard time dividing the tasks. Indeed, we all had more or less already our areas of predilections and thus we already knew what we wanted to do. However, during the course of the project, some members realized that it would be more beneficial for them to exchange some of the tasks. For example, Gabriel already knew how to make a multiplayer system and Timothé was more attracted to this task. That's why they all simply changed the person in charge of this task. As for Alexis, he liked the realization of the scenarios but wanted to do more mini games. So, with Gabriel became in charge of the voice chat letting him more time to improve the different games.

Once everyone knew roughly what they had to do, the project could begin. From modeling the game objects to developing the different rules. Everyone worked hard to meet the deadlines we had set for ourselves. And thanks to all this hard work we managed to meet them and even exceed them.

So, for this defense, we succeeded in keeping all our objectives and we moved forward for the next one.

In order to describe our progress on the project, we have divided our report into several parts, each dealing with a different part of the project.

Have a good reading.

2. Outline Of the First Presentation

Presentation of the project

- Description of the subject
- New book of specifications

Presentation of the project schedule:

- Description of common tasks
- Description of individual tasks
- Description of the estimated future presentations

Presentation of the individual tasks:

- Research & Design

Conclusion:

- Reminder of what is done
- Reminder of what is to be done

3. Presentation of the Project

a. Presentation of the team

The “Jupiter”. team is made up of three men studying computer science. Gabriel Toledano, the project manager who also gives graphic ideas, Timothé Merle who takes care of the technical questions Alexis Pinson who takes care of the game design. We are all passionate about video games in general, but also about life-size games. Indeed, playing a game is one thing, but living it is another. When we participate in games such as escape rooms, we are in the center of the action, we, decide how the game is going to go. If we find the necessary clues, we win, otherwise lose. This creates a real stress for us, but a positive one, the one that drives the game forward. The frustration of not finding the clues makes us look harder to succeed in our mission.

We aim to reach this level of experience for our project.

b. Description of the Subject

i. Nature of the project

In recent years, we have seen a great growth in the development of escape games. These usually take place in a single room and require players to find clues to escape. Each game can be played in teams or alone.

Thus, the Jupitr. team is going to develop a video game taking the form of an escape game. It will be called LockedUp.

ii. Goals

What is LockedUp’s goal? Entertain players as much as a real escape game would. At first glance, one might think that this is going to be complicated, because nothing can really replace the cognitive sensations of a game. Indeed, the smell and texture of the elements surrounding the players is complex to transmit through a screen. Therefore, Jupitr. aims to overcome the virtual barrier with sound and visual effects.

With these objectives, we want to create a unique experience for each game. This means getting as close as possible to real-life experiences. Indeed, as when you choose the theme of the escape game in which you will participate in real life, several atmospheres will be available in the game. This will allow a diversity of gameplay and will prevent the player from getting bored by the game too quickly.

c. Game development

In this section, we will present the different aspects of the game. First, we will explain how the game plays in general. Next, we will see how artificial intelligence will bring life to the game. Finally, we will discuss the multiplayer mode.

i. General characteristics

LockedUp is a first-person puzzle video game which will be played on Windows. Since am escape game is always more entertaining with a team, our game will be playable in solo or in multiplayer. After building its team, or not, we give to the player the opportunity to choose between three different levels. Each one of them has a different level of complexity.

Three environments with modifications will therefore have to be modelized in 3D.

We will now present what the scenes have in common. Firstly, each game will last a maximum of one hour. It means that some games can last less than that. For instance, we think that one of the levels could be done in 20 minutes. Nevertheless, the choice for the maximum time is not random, it corresponds to the maximum time of a real-life escape game. A countdown will then be displayed on the player's screen to indicate the remaining time.

Then, a bag system will be available allowing players to store items such as keys, for example.

Similarly, if the player retrieves visual information, they will be able to do a "screemember". This will take a screenshot of what the player is looking at and store it so that they can use the clue later.

The previous two points involve interaction with the environment offered to the players. In this way, we will allow users to jump on furniture or open cabinets to find clues.

ii. Artificial Intelligence

This part gathers all the AI present in the game. Indeed, several AI will be used for different purposes. First, the Game Master will be used to manage and help the players if necessary. His program will constantly analyze the scene in which the characters are playing to get different information. This information includes mainly: the time and the number of remaining puzzles.

Then we have the AI of the different non-player characters that will appear in the games. They will represent, for the most part, students stuck in the current scene. These, like the Game Master, will be able to talk to the players. Again, their speech will depend on the current state of the game.

Finally, the last AIs to be implemented are the ones presented in the mini games. We count 3 games for the moment: a chess game, a puzzle and finally a Connect 4. First, the AI that will control the chess games will have an intermediate or even beginner level of difficulty depending on the time left. Indeed, a chess game can sometimes take an hour while a game on LockedUp can take less time. The AI will adapt its game according to its opponent and the state of the team.

Second, the puzzle will also contain some kind of AI. Indeed, the mixture of pieces contained in it must not be random at the risk of having a game without solution! From then on, the player will play against the clock so no AI should be developed in addition.

Lastly, the Connect 4 AI will be easier to implement. Indeed, since Connect 4 is a fairly simple game, its level will not depend on the state of the game. Thus, the AI will just have to assign notes to each available token location in order to know where to place them. In this way, it is possible that some people will lose to the AI.

iii. Multiplayer

The multiplayer part has been, at least for moment, entirely done by Timothe. This includes several steps. To simplify the management of the server on which the multiplayer games take place we have opted for the Photon solution. This service allows to directly link a multiplayer server to our Unity project. By doing so, we set up a lobby and waiting room system. Each of them is managed by the team leader: the player who created the lobby. The latter is the only one who can choose the mode in which the team will interfere. Once the game is launched, several systems are set up. First, a spatial voice chat will allow more realism between each player. Then, if some players collect elements and want to share them with the other team members, this will be made possible. An inventory will be available on each character.

d. Project Schedule

Let's now talk about our planning.

This one has been perfectly respected. Indeed, the common and individual tasks, which had been planned, are either on schedule or even ahead of schedule.

i. Common Tasks Schedule

These can be quickly summarized. First, the mock-ups, which, although done by one person, needed general advice and modification, were completed at the beginning of the project. Then, the website was done by Timothé and Gabriel. Thanks to our experience in the web, it was done quickly, allowing us to concentrate on other tasks. So, except for the download page, everything else is done.

ii. Individual Tasks Schedule

The tasks representing the 3D modeling in general, as well as the game design of the game modes were done at the beginning of the project. This gave us a solid base to work from. Then we have two important parts, also at the estimated or exceeded level: the multiplayer and the different AI. The first one is at a very advanced stage with the lobby system. The second one is composed of NPCs and mini games. It is mainly the AI behind the Game Master that has been worked on and made functional for this first defense, concerning the NPCs. For the mini games, two of the three planned have been started or finalized. Since one is not, the progress is respected.

iii. Estimated Schedule for The Next Defense

As briefly stated in the introduction, we have exchanged our tasks in order to diversify our work. This means that we are updating our schedule and our division of tasks.

e. Work Breakdown

This table represents the distribution of tasks within the group. Each of them is assigned to two developers: the one in charge and his substitute in this order: “In charge; Substitute”

Tasks	**Members**
Game and Web Development	
Website	Timothé; Gabriel
Script writing	Alexis; Timothé
Minigames	Alexis; Gabriel
NPCs	Gabriel; Alexis
Setting up the lobby system	Timothé; Gabriel
Voice Chat	Gabriel; Alexis
Sound Effects	Gabriel; Timothé
Interact with the environment	Timothé; Alexis
Sharing Items	Alexis; Gabriel
Game and Web Design	
Website Mock-Up	Gabriel; Timothé
Game Mock-up	Gabriel; Alexis
3D modelling of environments	Gabriel; Timothé
3D modelling of NPCs	Gabriel; Alexis

f. Progression

Tasks	Presentation 1	Presentation 2	Presentation 3
Website	80%	100%	100%
Script writing	100%	100%	100%
Minigames	50%	100%	100%
NPCs	50%	75%	100%
Setting up the lobby system	50%	100%	100%
Voice Chat	10%	70%	100%
Sound Effects	0%	50%	100%
Interact with the environment	30%	60%	100%
Sharing Items	0%	50%	100%
Website Mock-Up	100%	100%	100%
Game Mock-up	100%	100%	100%
3D modelling of environments	80%	100%	100%
3D modelling of NPCs	50%	100%	100%

N.B:

- All the percentages in this chart may not be exact but they are our minimal objectives.

- All percentages highlighted in red represent tasks that have progressed further than expected.

4. Tasks Made by Each Member

a. Gabriel Toledano

i. Mock-Ups

To begin with, I had two mock-ups to do: one for the website and another one for the game in general way. Meaning, I made a quick design to explain to what the menu could look like for example. These two mock-ups were done at the beginning of the project and finished without any problem.
Even if this step adds some time to the global development of the game it decreases the time we spend when implementing the real game. Indeed, if the developer already knows where the elements should go, he will only have to make sure that what he does look like the model.

ii. 3D Modelling

The 3D Modelling seemed to be essential for the rest of the game development. In fact, since we need to make three different environments for the three levels that we are going to do, a base is needed for scenery. From that on, the last 20% that need to be done include the decorations and lights of each level.

To get a quick visual easily I, with the help of Timothé, modelled the whole building and most of the assets directly in Unity. In fact, the Prefabs, those 3D models that are dynamically editable, are a very powerful tool. The 3D modelling, in a low poly style, took the entire time from the beginning of the project to this first defense. This includes modeling the building, setting up the environmental lighting, creating all the prefabs used in the game and finally modeling the characters. Indeed, once the building was almost finished, I focused on the appearance of the characters. Those were taken in the Unity Asset Store and then modified, the skeletons and the mesh, directly in Unity.

Nevertheless, modelling everything in Unity made us encounter a major problem: a decrease of fps. In fact, the number of "Vertices" and "Triangles" which could be explained as being the level of detail of the scene, is too high. This is due to the unique number of elements. To counter that, I will model some elements together in blender and then import them in Unity.

For the moment, a glimpse of what we have modelled is in the appendix and are the figures 3, 4, 5 and 6.

In addition, I need to model the decors for the three levels to be able to test our different scenes.

iii. Players Interactions in the Environment

This task consisted in making a player move in the scene and making him know how to interact with the different present element. First thing was to choose the type of view we wanted to give to the players. We decided to go for a first-person view since we want to give the most immersive experience. To achieve that, Gabriel linked the camera to the player thanks to a script that made the user turn his head using his mouse. At the same time, the script to move the 3D model of the player was made. Its purpose was to make the character go in the 4 directions, jump, and fall thanks to basic physics. Using box colliders, I determined the boundaries through which a 3D model could not go.

The problem in this case was to compute the velocity of the player to make its fall look realistic. To achieve that, we added a tag to all elements on which a character could step. Example: the ground. Then a not visual object has been added to the bottom of the character that recognizes this tag. Thanks to this, we set the velocity to a certain value when the player is "Grounded" (on the ground), otherwise we make the value increase according to gravity.

iv. Website

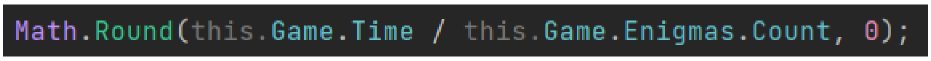
I have worked, as Timothé's assistant on the website. My main goal was to make the landing page. This one consists of a clean design with a classic navigation bar. When you click on "See More" the page goes down and you can see the timeline appear. It is accompanied by a section listing the various problems encountered during the development of the project. These two elements are not filled for the moment. Indeed, we are waiting to have more content and this filling will be done for the next defense. The timeline will list the most important steps of the projects. It includes the 3D modelling, the development of mini games but also the one of NPC's (Non-Player Character).

v. Game Master Development

Now we get to the interesting part of the project: the main A.I. The Game Master handle everything that can happen during a game. I chose to make its program look like a decision tree. In fact, its decision will be taken according to answers to specific questions. I first started by literally write the decision tree (see figure 1 in appendix). Then, once I had my idea, I started the Game Master in a C# console application. I wanted to try it out instead of directly putting it in a unity scene. The current state of the Game Master allows to do multiple tasks. At the beginning of the games, it computes the time needed for each enigma. This depends on the number of enigmas and the amount of time the players have, to finish a game. The enigmas are loaded at the beginning of the game and each one of them is linked to a clue.

After that, a function constantly calls the Game Master to help or not the players. This decision is, for the moment, made this way:

- Each time the Game Master is called, it computes the amount of time that could be used to finish the current enigma and still win the game. Winning the game means succeeding in all enigmas/steps of it before the global time goes to 0. The computation is a division between the remaining time and the number of remaining enigmas.
- If this time is less than the value computed in the first place, we chose to give the players a clue.



```
Math.Round(this.Game.Time / this.Game.Enigmas.Count, 0);
```

I planned to add multiple parameters to the decision tree of the Game Master. First, not all enigmas have the same difficulty. In fact, finding a key in a room and winning a chess game don’t exactly take the same time. This will result in computing a level of complexity for each enigma. According to this level, the Game Master will allow more or less time to solve one problem.

Secondly, not all games are going to be played in solo. It is easier to win in team than alone. For instance, when there are multiple players, they can cover a larger space of the environment to find an item. This will also be considered when calculating the time needed for one enigma.

vi. Voice Chat

This is the last part that I worked on before this defence. For the moment I use agora voice channels. Agora is a company providing vocal services to make calls but also to make your own vocal chat project. I started implemented this on a different scene than the one where Timothé developed the multiplayer.

We work like this to limit our number of merge conflict in git. Doing so, when playing the scene, you can:

- Create a lobby/channel
- Join it or a previously created one
- Leave the joined channel

Nevertheless, there is something that I still want to improve. Indeed, we want our game to be as immersive as possible. To do this, a space vocal chat would be the most suitable. This means that the sound of another player's voice can decrease or increase depending on his distance to you.

b. Timothé MERLE

i. 3D Modelling

For this first defense, I was mostly concerned with the development side of our project, I didn't touch too much on the graphic side of the project. I worked as an assistant for Gabriel in his 3D modeling task. Indeed, I helped him to model the main building of Epita, the VA, and other objects necessary for the good functioning of our game. I helped him by modeling the 2nd floor of the building and the last one with the machine rooms following the plan provided by Epita, to try to have the best representation possible. For the next defenses we would like to add small details in all the building as for example small Leds of colors, a more realistic ceiling...

To save as much memory as possible, and therefore the game lags as little as possible for the players, we must try to have as few objects as possible because each object, when it is designed directly from Unity. Will have a certain number of "Vertices" and "Triangles" and the problem is that the higher this number is, the more time Unity will take to recalculate them. That's why using Blender, we can reduce the number of "Vertices" and "Triangle". So, we will have to re-designate most of our objects from Blender.

Also, to make it easier for us if we want to add elements or this kind of things, we worked with prefabs. Which allow to make a modification only on the prefab and this modification will be propagated on all the elements using this prefab.

ii. Menus

I also made different menus at the beginning of the game to guide the player, like joining or creating a game or playing in multiplayer or solo. I didn't design these menus; I simply followed the models made by my friend Gabriel. For these menus, I didn't meet any major problem during their realization, only at the time of realizing the menu which is displayed when the players are in the lobby and wait for the beginning of the game. During the design I realized that when the group leader changed the game mode, it was not updated for the other players, so I had to rectify this.
To facilitate, the design of the new menus, if we were to create new ones like for example for settings or things like that, I went through several different ones. Indeed, I created a scene for each menu and when the player clicks on a button like for example "multiplayer" then I change the current scene with the predefined function of Unity on the scene named "Multiplayer". This system also allows to find better in the organization of the game because all the objects are not all in the same scene and superimposed on each other.

iii. Website

In parallel of what I'm going to explain you and what I have already explained to you, I developed the website on which our game is downloadable, and we quickly present the idea of the game. As for the menus of the game, I didn't design the website, I simply put in form what my friend Gabriel designed. The design of this website was fast and efficient, in fact with the expertise of my friend and myself in the field of the web, we finalized it in less than a day.
For the development of the website, we divided the tasks to go as fast as possible. Gabriel took care of the home page, the page where you arrive when you go on the site. While me, I took care of the page where we inform all the libraries that we use with unity, as well as the links towards the various social networks of each member of the group and the software that we used during the realization of our project. And I also have the page where you can see the different members of the group with a picture of each of us and a short description. We have already designed the page where you can download the game. However, we haven't developed it yet because the game isn't ready yet.

iv. Lobby System

I'm going to talk about the part that probably took me the longest to realize. Indeed, I had never made a lobby system allowing players to play with more than one person. After my friend Gabriel and I finished creating the first scenes, I asked about the different ways to set up this lobby system. Several methods were described, but the most practical and fastest to set up in the time we had was Photon Engine. Indeed, this library available for free on the internet allows to manage the lobby multiplayer system very easily.

However, this system has a small problem that I had to work around. Indeed, as this system works is very simple. First, we must create an account from the Photon Engine website, which will allow us to create an application. This application will give us what we call a UID, unique identifier, that we will then fill directly in unity. Once this is done, we can connect the players to each other. To do this, we must first connect the player to the Photon server, then the player can either join what we call a lobby created by another player or create his own lobby so that his friends can join him and play together. The problem with Photon, is that the lobby names must be unique, which is not practical if we want several players to use the same lobby name, so to solve this problem, I generate a UID, which I define as the name of the lobby, and with which other players can join the lobby. Once all the players are in the right lobby, the group leader can choose which game mode he wants to launch and then can launch the game by pressing the start button. Indeed, there are 3 different game modes, which simply represent 3 different escape game scripts.

The leader of the group then starts the game, once this is done, all players appear on the map at predefined locations, and can control their characters. To succeed in setting up the system that allows each player to control only his character and not the one of his friends, I encountered several problems. Indeed, Photon sees each player as the same if we don't tell him exactly that they are not the same. That's why at the beginning, each player was controlling all the characters including the one of the others. So, I looked more deeply into how Photon recognized the players and their cameras. To solve this problem, I left only one camera at the beginning of the game and assigned it directly to the leader of the group in code and then as soon as a new player appears in the scene, we create a new camera and assign it to him. Also, when the players move, Photon searches in all the objects of the scene what belongs to such or such player, that's why at the start of the scene or at each refresh of the scene, I check if the current camera is the one of the players or not and if it's not the case then I block all actions.

To make it easier to know if the camera is the one of the local players (the actual player, the one who is behind his computer), Photon has set up what they call a Photon View (see Figure 7), and which can be attached to any object of the scene.

For the next defense, I must make compatible the animations of movement of the characters made by my friend Gabriel with the multiplayer mode because, indeed, currently at the time of the launching of a game in multiplayer the characters do not have animation of movement. And, to return compatible the system of vocal chat.

v. Four-in-a-row

To add a little challenge to the players, they will have to face artificial intelligence in several mini games such as the four-in-a-row and beat them to get the clues. For the design of this game, I first developed it in a terminal to visualize it and then thought about an artificial intelligence. The artificial intelligence is not yet ready for this defense, but I am actively working on it. The method I'm going to use is the following: I'm going to take each square of the game board and transform it into a matrix of numbers that will represent the percentage of each square. Indeed, we will determine for each square a percentage or a power to know in which square it is more interesting to play the next move. This percentage is calculated according to the pieces that surround it and therefore the probability of winning.

c. Alexis PINSON

The following text contains spoilers about the different stories in our game. If you want to keep the surprise, please do not read.

i. Script writing

To realize the different stories, I played about 5 escapes games, and I also watched about ten videos. Indeed, to realize an escape game from scratch takes on average 2-3 years.

1. Night at school

The objective of this first game is to familiarize the user with the interface. Therefore, the difficulty is relatively low.

The escape game is broken down into several rooms. First, the players appear in a locked room. They must get out by simply finding a key in the room. Here, the user will be able to get used to using their inventory and moving around in the great place that is Epita.

After this first very simple part, you will have the possibility to walk around the ground floor to find a way to get out of the school. You will encounter several challenges, including letters written on dice, computers that must be turned on, and a 15 puzzle (which I will explain later).

All these tests will lead to the exit, and you will have finished the first level of this game.

2. *School Psycho*

This second level is already more complex, because it uses mechanics that are more difficult to understand.

This level is like a “cluedo”: players find the place, the weapon, and the killer. They put themselves in the shoes of a real detective.

To do this, players must first find and use a UV lamp to reveal the murder room, which is none other than the bathroom. Then, they must play with the light to find the murder weapon which turns out to be a knife. And finally, the players must find the name of the killer thanks to the fingerprints he left on the knife.

3. *The whisper of darkness*

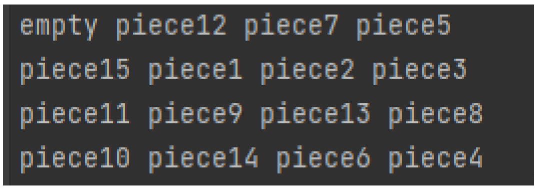
This level is much harder than the other two. Indeed, players will have to find a way to turn the light back on, especially with the help of cable-based puzzles. However, their mission will not be completely finished: they must then discover what really happened at the school. To do this, they will have other tasks to complete such as a game of chess, but they will also have to use their remaining architecture course as there will be xor doors with symbols.

ii. code of the mini game "15 puzzle”

I will now tell you about the code of one of the mini games: the “15 puzzle”.

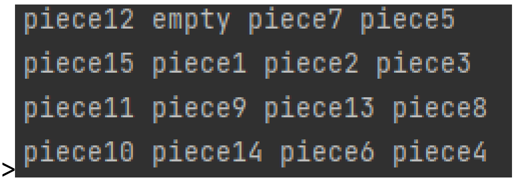
What is this game? It is a square made up of 15 small squares with a free space. The objective of this game is to reconstitute a pattern by sliding the pieces in the free space. (Figure 2)

For example, in this situation, if you want to move piece number 12, the game will look like this.



```
empty piece12 piece7 piece5
piece15 piece1 piece2 piece3
piece11 piece9 piece13 piece8
piece10 piece14 piece6 piece4
```

=> 12=>



```
piece12 empty piece7 piece5
piece15 piece1 piece2 piece3
piece11 piece9 piece13 piece8
piece10 piece14 piece6 piece4
```

At the code level, I used two classes: a class called "Piece" which manages the different pieces and the class "Board" which manages the board.

The "Piece" class has 3 variables: the x and y variables which allow to know its location on the board and the image variable which will store the image that this piece should display. There are also the getters, the setters, the constructor, and another method. This method called "GoEmpty" returns a Boolean indicating if the concerned part can go to the free location placed in parameter. Moreover, if it can, the two parts are reversed.

The "Board" class has only one variable which is a list of "Piece" but has 4 methods. The first one is "Shuffle". It allows to mix as its name indicates the pieces. However, it is in this part that we find the greatest mathematical complexity. It is important to know that it is possible to tell in advance whether the problem posed is soluble or not. Indeed, the initial configuration of a teaser is a permutation of its final configuration. This permutation is called even if it can be obtained by an even number of successive exchanges of two squares, adjacent or not, empty or not, also called transpositions. We show that this notion does not depend on the choice of the sequence of exchanges. It is odd otherwise. We also associate to the empty square a parity: the empty square is even if we can go from the initial position of the empty square to the final position in an even number of moves, odd otherwise. The problem is solved if the parity of the permutation is identical to the parity of the empty square. The difficulty is therefore to mix this puzzle without making it insolvent. The second method is called "Display" and allows the user to see the board in the console. The third one is called "Move". It takes a room as a parameter and returns a Boolean if it has been moved or not. For this method, I use the "GoEmpty" method of the part class. The last method "IsFinished" returns true if the puzzle is finished, false otherwise.

All these classes and methods now make it possible to have a very simple main program. Indeed, it is now enough to initialize the puzzle, to mix it, to display it and to create a loop which stops when the board is finished.

5. Conclusion

In conclusion of this report, we have done all the bases of our project. That is to say that we have for example the bases of the modeling of the game, a functional multiplayer system but to be improved and mini games in development. That's why for the next defense we must settle the few details we have left on the modeling level.

We will also have to start implementing the voice chat, and the ambient sounds of the game. As for the improvements, we must implement the mini games like the four-in-a-row or the puzzle directly in the game. And we also need to tweak 2-3 things like the multiplayer and the website. We also must improve the game master, which is the main part of the game. For this one we will improve his ability to recognize at which state the players are in the scenario so that he can give the best clues. As well as the different NPCs that will appear in the game. For the NPCs, we will simply develop an AI that will allow their movements to have a minimum of sense and that their dialogue is coherent.

6. APPENDIX

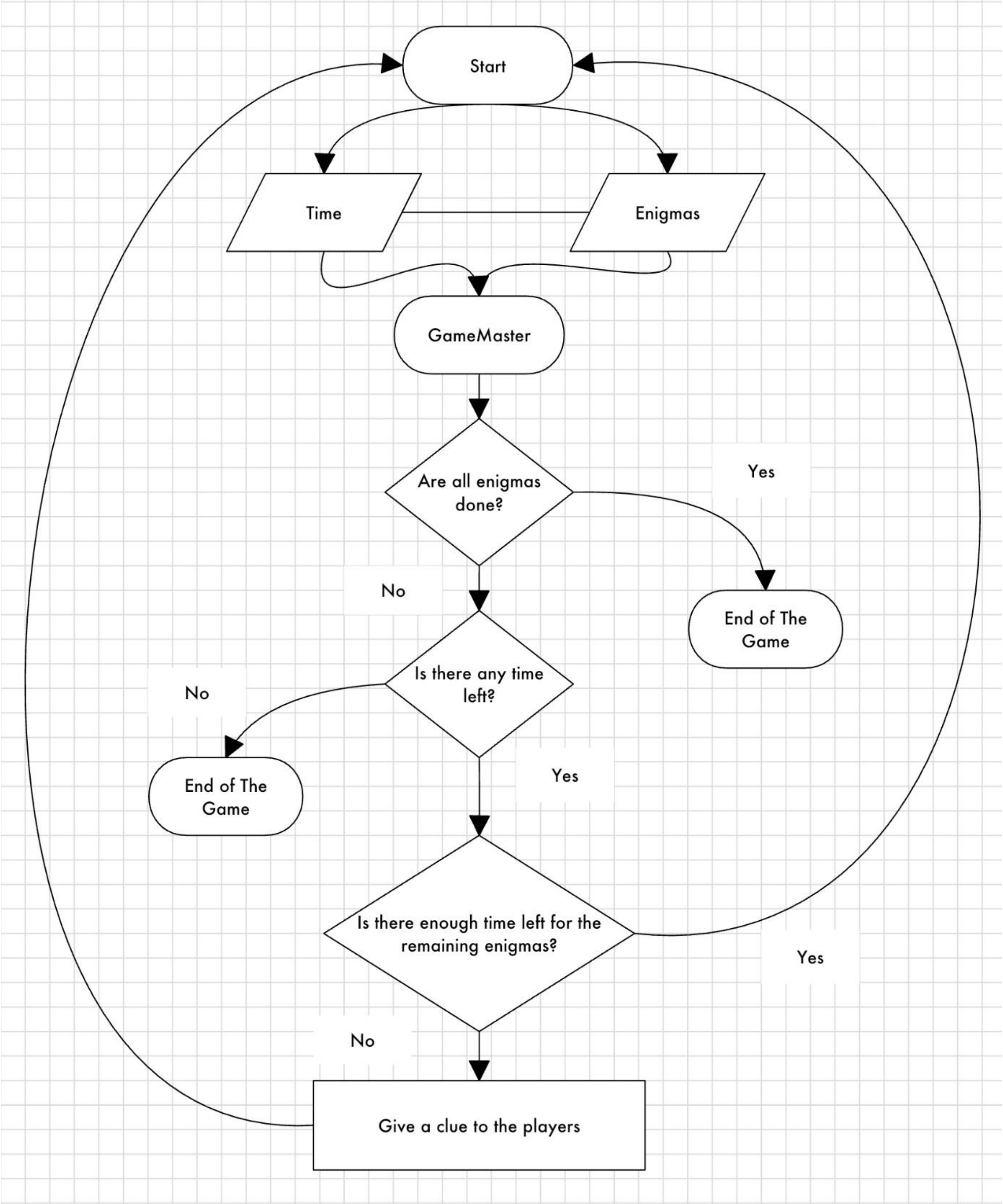


Figure 1: Decision Tree of the Game Master

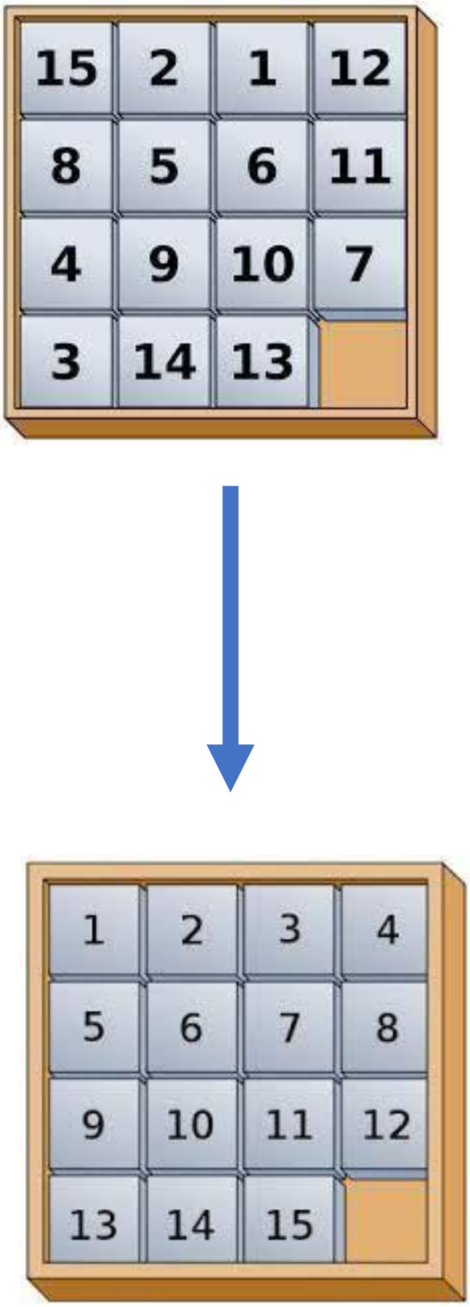


Figure 2: Puzzle Gqme

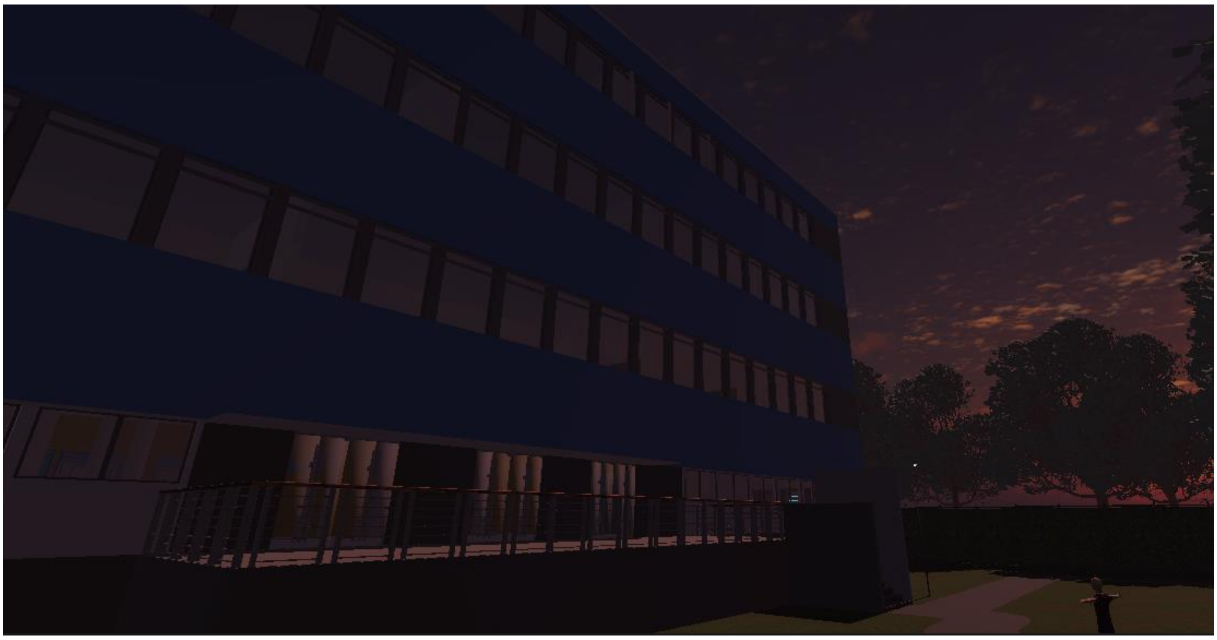


Figure 3: 3D Modelling from the outside

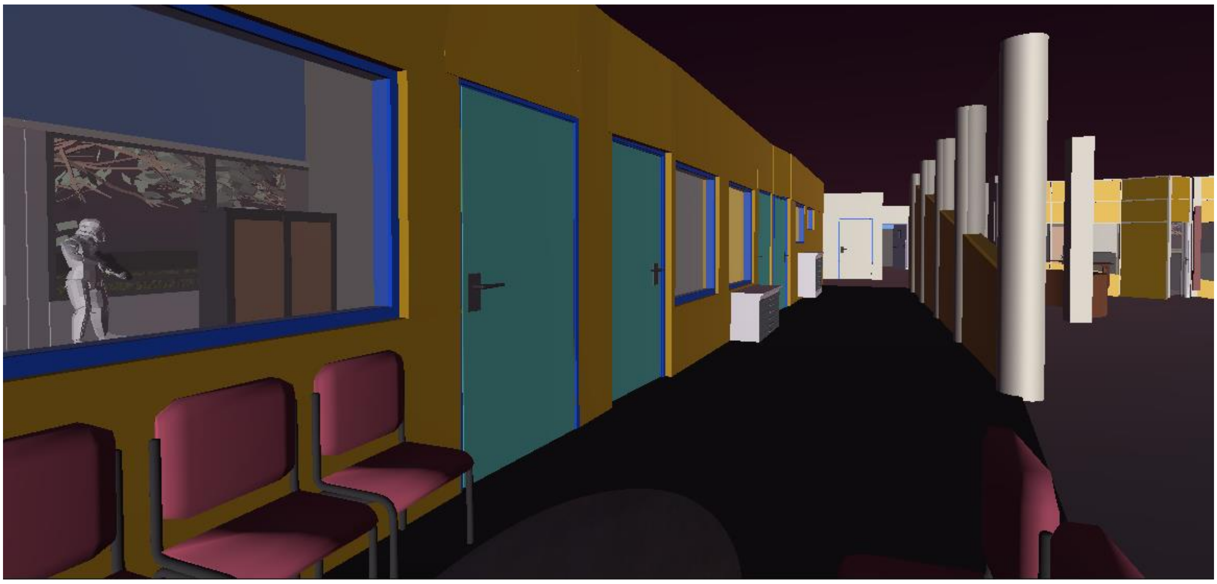


Figure 4: Administration Corridor

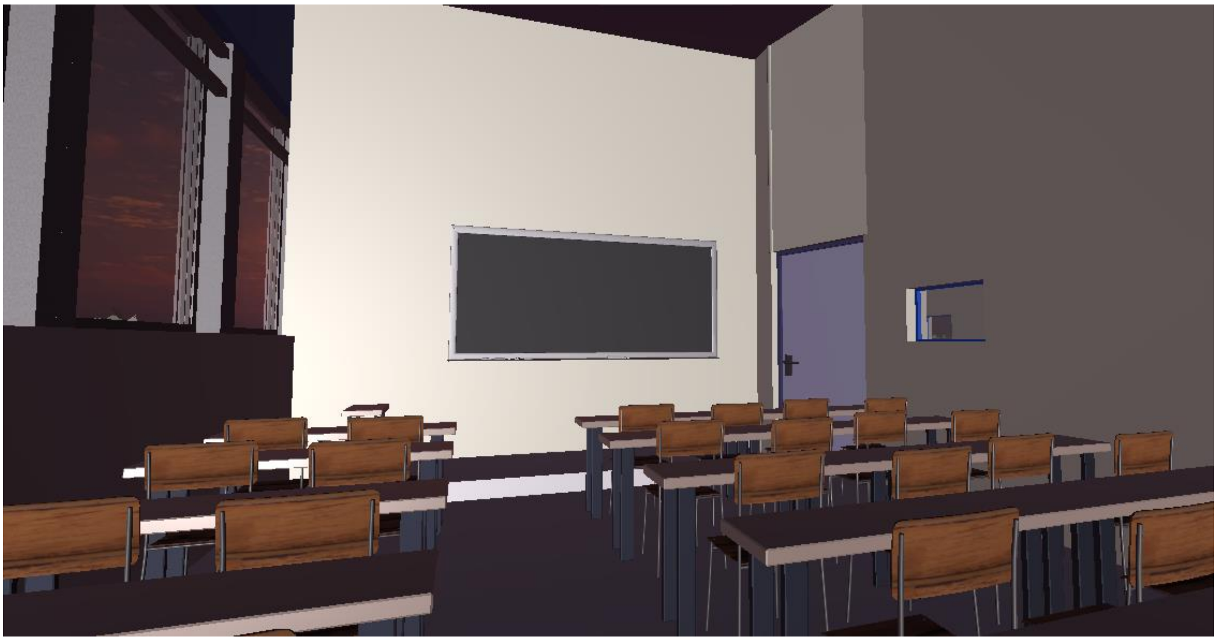


Figure 5: Example of a Classroom

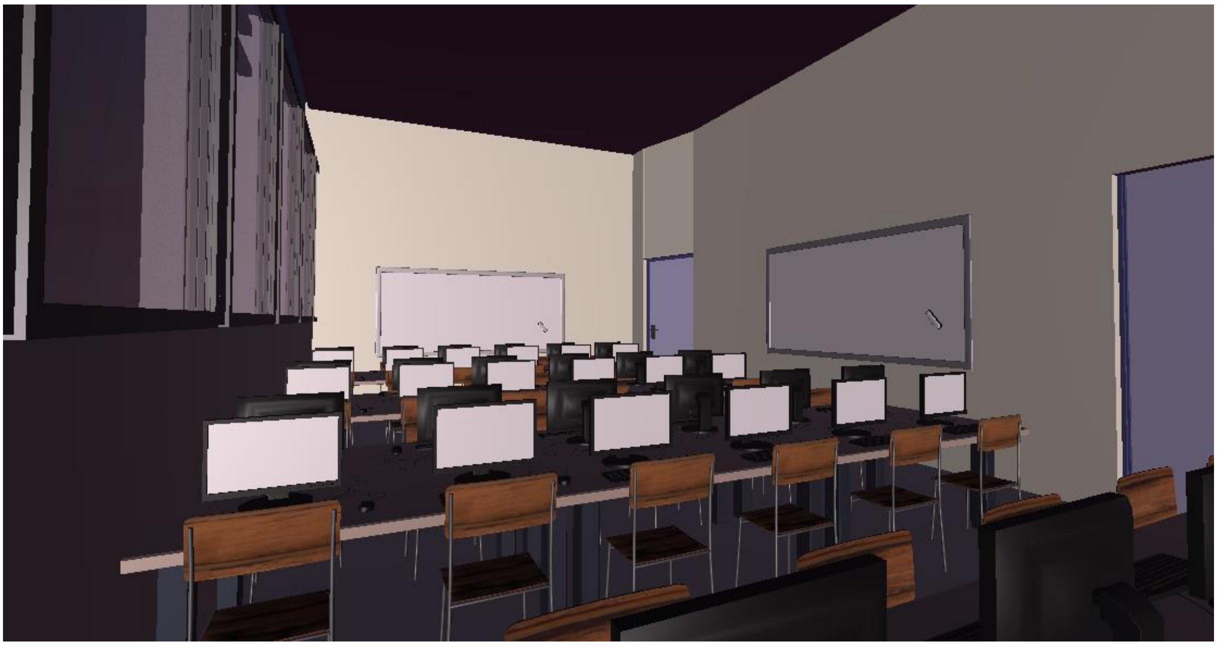
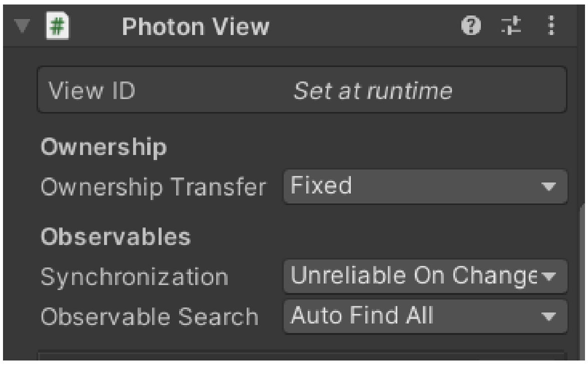


Figure 6: Example of an IT Classroom



Photon View

View ID Set at runtime

Ownership

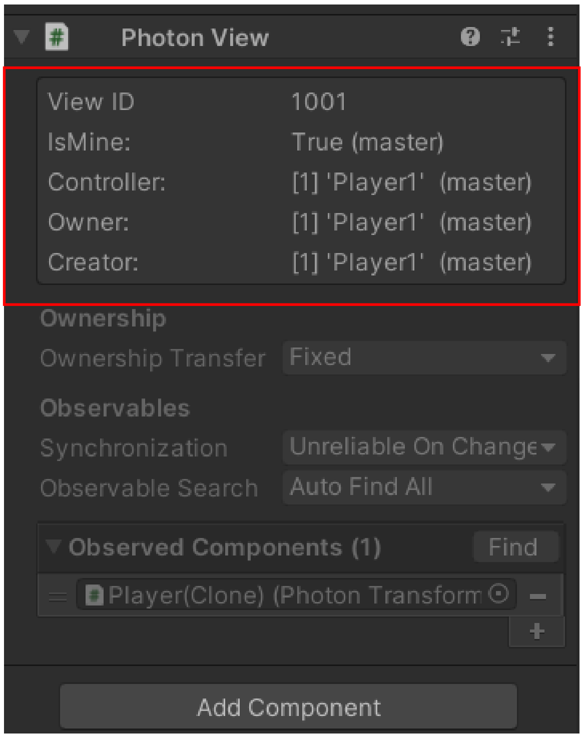
Ownership Transfer Fixed

Observables

Synchronization Unreliable On Change

Observable Search Auto Find All

Figure 7: Photon View not in game



Photon View

View ID 1001

IsMine: True (master)

Controller: [1] 'Player1' (master)

Owner: [1] 'Player1' (master)

Creator: [1] 'Player1' (master)

Ownership

Ownership Transfer Fixed

Observables

Synchronization Unreliable On Change

Observable Search Auto Find All

Observed Components (1)

Player(Clone) (Photon Transform

Figure 8: Photon View in game