



**ESCAPE ROOM ORIENTED TO AWARENESS OF THE  
PROBLEMS AND SOLUTIONS OF A PANDEMIC:**

**Óscar Seguí Rodrigo**

Final Degree Work  
Bachelor's Degree in  
Video Game Design and Development  
Universitat Jaume I

June 21, 2020

Supervised by: Joaquín Huerta Guijarro





*To my dad, Jose Manuel,  
who helped me find this career and for his great support  
during all this time. Since I was a child I have always been passionate  
about video games, and playing them with him was  
the best thing in the world.*

\*\*\*

*To my mother Cristina,  
thank you for all the support you have given me throughout  
this stage of my life and for having always been there.  
Thank you for being as you are.  
Thank you for being the best mum in the entire world.  
Just, Thank you.*

\*\*\*

*To my friends Angel, Miguel Angel, Sergio, Hector and Jaume,  
who have always supported me and with whom I have shared  
incredible moments during these four years.  
I hope our friendship never ends, even though we are far from each other.*



## ACKNOWLEDGMENTS

First of all, I would like to thank my Degree Final Project supervisor, Joaquín Huerta Guijarro, for his help provided before and during the development of this project, which was crucial in all the part related to the beacon synchronization, as well as for his advice during the development of the project. Without him, this project would not be possible.



## ABSTRACT

This document presents the project report of the Video Games Design and Development Degree Final project by Óscar Seguí Rodrigo. It is a video game titled *Laboratory 93* that consists of a 2D escape room based on the actual pandemic we are living at this moment, but in the year 2050, and using BLE beacons for the interaction with the different rooms in the real world.

The player will be able to move through different rooms for changing between the game scenarios, looking for the necessary items and resolving the clues in this laboratory with the objective of creating the vaccine to save the people and escape this place.

This will create a greater challenge considering that the synchronization with a game and a beacon is very complex and, of course, a great form for the player to get a more immersive experience.

As well, it is necessary to comment that the part related to the development of the beacons interactivity is still in progress, so in the corresponding section it is mentioned the work developed until the first day of July.





---

# Contents

<b>Contents</b>	v
<b>1. Introduction</b>	11
1.1 Work Motivation . . . . .	11
1.2 Objectives . . . . .	12
1.3 Environment and Initial State . . . . .	12
<b>2. Planning and resources evaluation</b>	14
2.1 Planning . . . . .	14
2.2 Resource Evaluation . . . . .	16
<b>3. System Analysis and Design</b>	19
3.1 Requirement Analysis . . . . .	20
3.2 System Design . . . . .	21
3.3 System Architecture . . . . .	28
3.4 Interface Design . . . . .	28
<b>4. Work Development and Results</b>	30
4.1 Work Development . . . . .	38
4.2 Results . . . . .	38
<b>5. Conclusions and Future Work</b>	40
5.1 Conclusions . . . . .	40
5.2 Future work . . . . .	41
Bibliography	43
Bibliography . . . . .	43
Appendices . . . . .	45



# INTRODUCTION

## Contents

---

1.1 Work Motivation . . . . .	1
1.2 Objectives . . . . .	1
1.3 Environment and Initial State . . . . .	2

---

This chapter shows what the purpose of the work was in the beginning, why and how this project was going to be developed.

### 1.1 Work Motivation

We live in a complicated and difficult time, all due to the pandemic that is ravaging the entire world, and people are worried about what is going to happen in the future. However, currently the situation is starting to stabilize. That is why I decided to base my project on this fact.

I love escape rooms since I was a child. In my childhood I spent a lot of hours playing online games in several pages like *minijuegos.com* [1], and my favorite games were the escape rooms. I have played dozens of them, so when I saw this project as a possibility, I knew it: I needed to choose this project.

So, the main idea of this project is to create an Escape Room-style video game [2] that deals with this topic from a very close perspective to reality but with several fantastic nuances to explain this concept and, thus, through the fun of this style of games.

## 1.2 Objectives

Based on the motivation of the work, these are some goals to achieve:

- Understand how the process of the creation of a vaccine is followed.
- Understand how a 2D escape room game is programmed.
- Make an escape room with some real interactivity.
- Learn more about programming in Unity for mobile devices.
- Learn how to synchronize a mobile phone with a bluetooth beacon for spatial recognition.

## 1.3 Environment and Initial State

To get started, I tried to search which is the procedure to create a vaccine. I had millions of results, and most of them were scientific articles for doctors and medical students, and I didn't understand a lot of things. So, I needed to narrow down the search, so I tried with "vaccine ingredients" and I found a really graphic webpage [3] in which this process is explained for people with less knowledge in this field. Thus, I learned the basics about the creation of the vaccines and their main components: antigens, adjuvants, stabilizers, preservatives, surfactants, waste products and diluents. These components have their specifications for every single vaccine, but, roughly, they are always the same.

When I learned all this process, the next step was to think about the art style of the game. Because it is a mobile game, the screen will be smaller than in a computer. So, I thought that the best idea was to create line-art [4] scenarios. Line art always improves the visibility of the objects and elements in the scene. Also, for people without great drawing skills, this will be the easiest way, because the main feature will be the synchronization with the beacons.



## PLANNING AND RESOURCES EVALUATION

### Contents

---

2.1 Planning . . . . .	3
2.2 Resources Evaluation . . . . .	5

---

This chapter deals with a technical part of the work. It also shows the planning for the development of the project.

### 2.1 Planning

Planning is a very important task in the project to monitor and keep the workflow. Some things have changed from the initial days until now. Here it is detailed how the development of the project is divided:

- Task 1 ( 5 hours): reading and learning about the creation process of a vaccine.
- Task 2 ( 15 hours): Design of the puzzles and connection between the object
- Task 3: (30 hours): Creation of the art of puzzles and collectable objects.
- Task 4 (100 hours): Creation of the 3 scenarios and HUD elements (main menu, end screen, etc.)
- Task 5 (50 hours): Programming of the puzzles, game flow and UI elements.
- Task 6 (40 hours): Programming the interactivity of the game with the bluetooth beacons.
- Task 7 (40 hours): Project report and other documents;

- Task 8 (20 hours): finishing the game with the 3 scenarios implemented, all the puzzles and object interactivity, adding sounds and several features to complete it.

Total hours: 300 hours.

Here is a summary of these tasks in the order that have been done (see Figure 2.1).

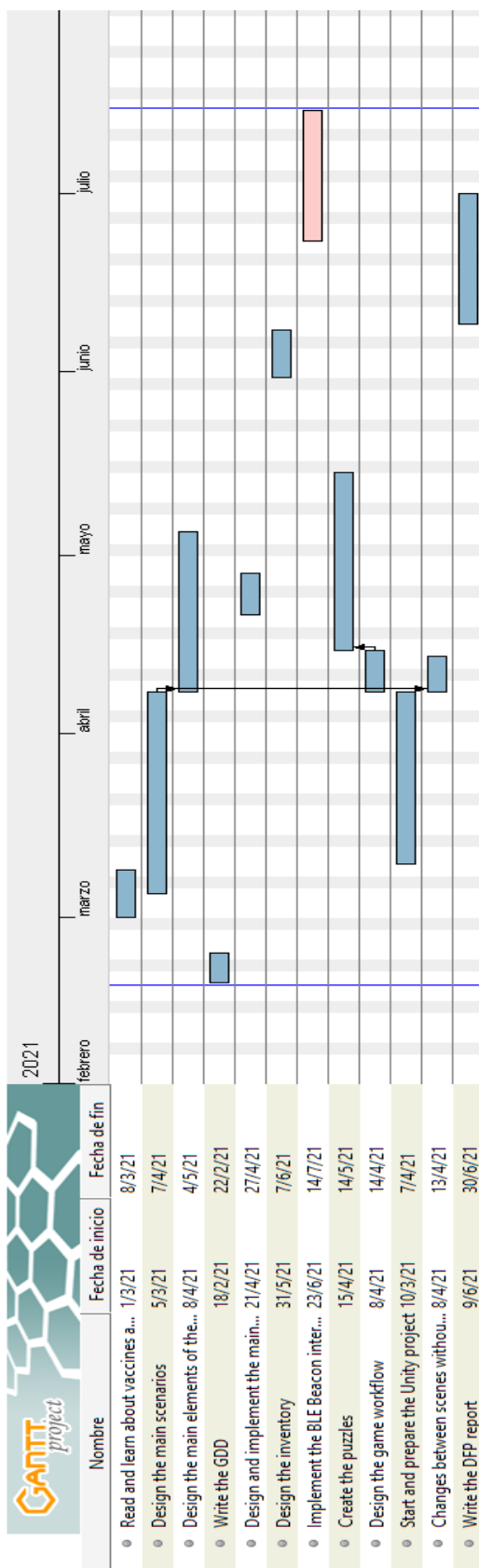


Figure 2.1: DFP task planner (made with Gantt Project [5]).



## 2.2 Resource Evaluation

In this point is listed the software and hardware needed to do this project:

- PC ( i7 CPU, 16GB RAM and GPU GTX 1070 Ti). I used this pc for programming a great part of the game.
- Laptop ACER (i7 CPU, 12 GB RAM and GPU GeForce 820m). I used this laptop for programming the beacon synchronization, because the other PC does not have bluetooth incorporated.
- Unity 2D 2020.2.3f1: the engine of the project [6].
- Visual Studio 2019 for Windows: with Unity 3D is a very useful tool for coding and allows switching between Unity and Visual. Besides, Visual Studio has a powerful debug function which will help me to find and solve bugs. I will use C# language because it is very powerful and versatile [7].
- Github (Github Desktop): a git repository where I will upload my work. It is a good tool to avoid losing progress in the project if a problem occurs on the computer [8].
- Adobe Photoshop CC 2020: it will be used to design every single asset of the game, as well as the scenarios and the user interface (GUI) [9].
- Trello: a website used to control the work. Divided on pending, doing and done it can ve used to divide the whole work into tasks. In helps a lot with the organization [10].
- Unity Asset Store: a website for buying and selling assets [11]. This page offers a very large amount of free assets and plugins, so it is very useful to download, in my case, a visual novel plugin (Fungus) for the conversations.



## SYSTEM ANALYSIS AND DESIGN

### Contents

---

3.1 Requirement Analysis . . . . .	6
3.2 System Design. . . . .	7
3.3 System Architecture . . . . .	13
3.4 Interface Design . . . . .	14

---

This chapter deals with the requirements analysis, design and architecture of the proposed work. It also shows the design of the GUI, commonly known as the game user interface.

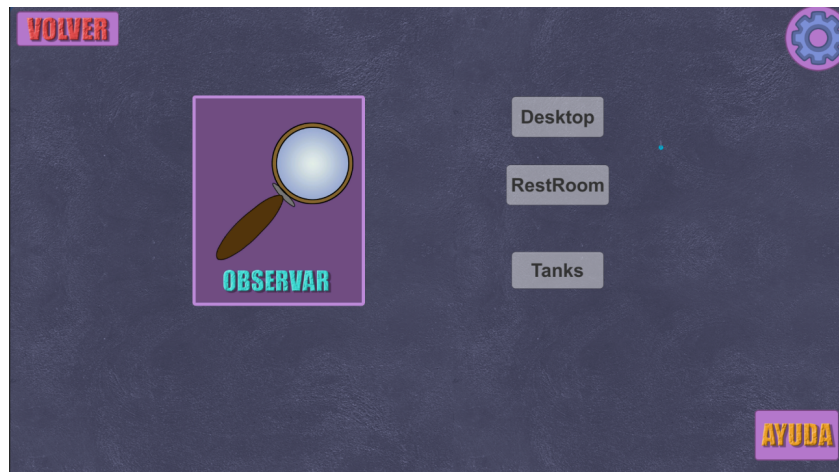


Figure 3.1: Image that shows the Game User Interface.

### 3.1 Requirement Analysis

*Laboratory 93* is a 2D Escape Room game which belongs to the *Point and Click* genre [12], but mixed with some real world interactivity, because the player needs to move through some rooms to change the scenario in the game.

As soon as the game is opened a main menu () will appear with two options: Play, Quit and Configuration. Pressing the latter will close the game. And with the *Play* tab will enter in the main screen of the game.

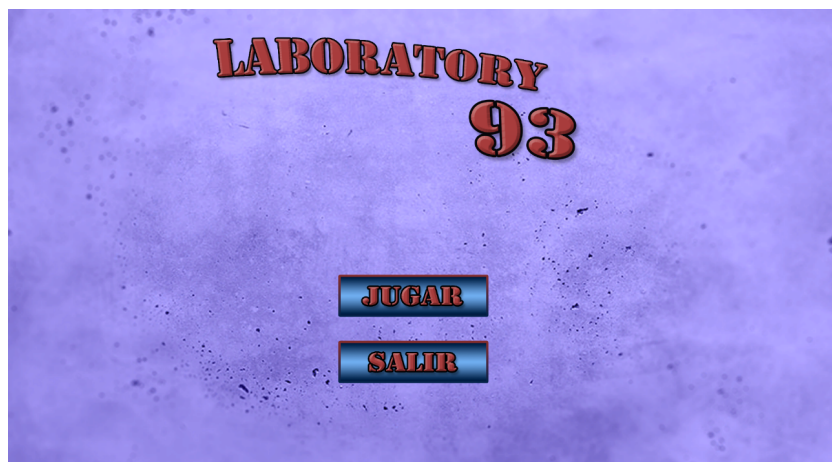


Figure 3.2: TheMain Menu screen.

Once the play has started, it is possible to perform different actions. For example, the player can go to the *Options* menu for changing some properties like the volume or the brightness. As well, it is possible to return to the main menu by pressing the button *Return* in the left up corner of the screen. And, finally, the player can press the big button *Observe* to look for the nearest room in his position.

Once this button is pressed, and the game finds the nearest beacon, the player will be moved to the specific room. He can interact with the entire room, to look for some relevant elements such as objects, features, etc. In the room, he will see his inventory on the bottom of the screen, in which the elements he finds in the game will be shown. Also, the button *Return* will be at the same corner, and it will send him to the previous screen.

The main objective of the player is to find the cure of the virus and test it on the three main living beings: a plant, an animal and a human. Each of them needs to take the vaccine in several ways for acquiring the immunity. To achieve that, it is necessary to look for, first time, the components of this vacuum, and once the player gets them, craft it in the crafting table located in one of the rooms and create the perfect mix for each patient. Finally, when the player gives them the cure, the door will open and the player will have saved the world (and will be free, as well).

### 3.1.1 Functional Requirements

A functional requirement defines a function of the system that is going to be developed.

Let's see these requirements:

- R1. The player enters the game by pressing the Play button
- R2. The player searches for the nearest room according to his real position.
- R3. The player interacts with the elements of the scene by touching them on the screen.
- R4. The player can collect objects by interacting with them by touching the screen.
- R5. The player can enter the options menu by pressing the Gear button.
- R6. The player can choose the volume of the game.
- R7. The player can return to the previous screen by pressing the Return button (the player needs to be in a different scene inside the game, not in the *Search scene*).
- R8. The player can select an object from the inventory (he needs to take it first).
- R9. The player can use the object to proceed in the game (he needs to select the correct object in the inventory).
- R10. The player can quit the game by pressing the "Quit" button.
- R11. The player can return to the main menu by pressing the "Return" button on the search scene.

### 3.1.2 Non-functional Requirements

Non-functional requirements are requirements that impose restrictions on design or implementation such as restrictions on design or quality standards. These are properties or qualities that the product must have.

- R12. The aesthetics will be cartoon line-art.
- R13. The game puzzles must be simple, not frustrating and funny.
- R14. The player can't get stuck in the game line for too long.

## 3.2 System Design

This section presents the logical and operational design of the system to be carried out. To do the logical design is shown with a Case Use diagram. And the descriptions of these cases are here:

---

<b>Requirements:</b>	R1
<b>Actor:</b>	Player
<b>Description</b>	At the beginning, the player will have the option to select “Play” to start the game which will be on the title screen.
<b>Preconditions:</b>	1. The player is on the title screen.
<b>Steps normal sequence:</b>	<ol style="list-style-type: none"><li>1. The player selects “Play”.</li><li>2. The game loads the search screen.</li></ol>
<b>Alternative sequence:</b>	None.

---

Table 3.1: Case of use «CU01. Play»

---

<b>Requirements:</b>	R6
<b>Actor:</b>	Player
<b>Description</b>	The player can change the volume of the game.
<b>Preconditions:</b>	1. The player must have entered the options screen.
<b>Steps normal sequence:</b>	<ol style="list-style-type: none"><li>1. The volume of the game is adjusted to the player’s selection.</li></ol>
<b>Alternative sequence:</b>	None.

---

Table 3.2: Case of use «CU02. Change Volume»

---

<b>Requirements:</b>	R10
<b>Actor:</b>	Player
<b>Description</b>	Allows the player to quit the game.
<b>Preconditions:</b>	1. The player must be on the title screen.
<b>Steps normal sequence:</b>	<ol style="list-style-type: none"><li>1. The player selects "Quit".</li><li>2. The game is closed.</li></ol>
<b>Alternative sequence:</b>	None.

---

Table 3.3: Case of use «CU03. Quit Game»

---

<b>Requirements:</b>	R11
<b>Actor:</b>	Player
<b>Description</b>	Allows the player to exit to the main menu.
<b>Preconditions:</b>	1. The player must be on the search scene.
<b>Steps normal sequence:</b>	<ol style="list-style-type: none"><li>1. The player selects "Return".</li><li>2. The player is taken to the main menu.</li></ol>
<b>Alternative sequence:</b>	None.

---

Table 3.4: Case of use «CU04. Return to main menu»

---

<b>Requirements:</b>	R7
<b>Actor:</b>	Player
<b>Description</b>	Allows the player to return to the Search scene.
<b>Preconditions:</b>	1. The player must be in one of the three main scenes.

**Steps normal sequence:**

1. The player selects "Return".
2. The player is taken to the Search Scene.

**Alternative sequence:** None.

Table 3.4: Case of use «CU04. Return to main menu»

---

<b>Requirements:</b>	R2
<b>Actor:</b>	Player
<b>Description</b>	The player searches for the nearest room according to his position.
<b>Preconditions:</b>	1. The player must be on the search scene.

**Steps normal sequence:**

1. The player selects "Observe".
2. The player is taken to the specific room.

**Alternative sequence:** [1.1] The player will attempt to search a scene but the game can't find it.

Table 3.5: Case of use «CU05. Search nearest scene»



---

<b>Requirements:</b>	R4
<b>Actor:</b>	Player
<b>Description</b>	The player collect an object from a room
<b>Preconditions:</b>	1. The player must be in the room where the object is located.

---

**Steps normal sequence:**

1. The player selects the object.
  2. The object is removed from the scene and taken to the inventory panel.
- 

**Alternative sequence:** None.

Table 3.6: Case of use «CU06. Take an object»

---

<b>Requirements:</b>	R8
<b>Actor:</b>	Player
<b>Description</b>	The player selects an object from the inventory
<b>Preconditions:</b>	1. The player must have taken the object first (see CU06 in 3.6).

---

**Steps normal sequence:**

1. The player selects the object in the inventory panel.
  2. The object is marked with a white square.
- 

**Alternative sequence:** None.

Table 3.7: Case of use «CU07. Select an object»

---

---

<b>Requirements:</b>	R9
<b>Actor:</b>	Player
<b>Description</b>	The player uses an object from the inventory to proceed in the game.
<b>Preconditions:</b>	1. The player must have selected the object first (see CU07 in 3.7).
<b>Steps normal sequence:</b>	1. The player selects one location of the room to use the object.
<b>Alternative sequence:</b>	The object can't be used in the selected place.

---

Table 3.8: Case of use «CU08. Use an object»

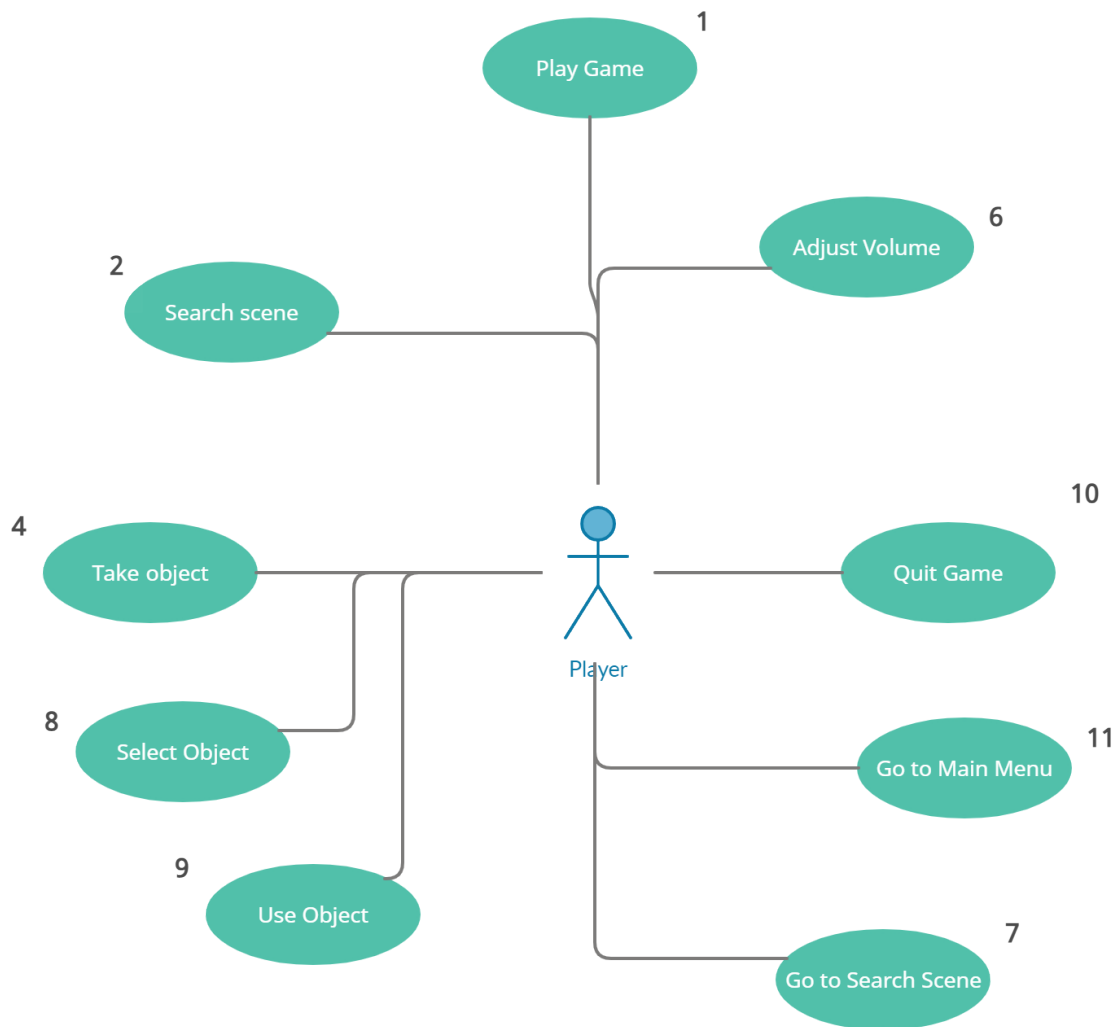


Figure 3.3: Case use diagram (made with <https://creately.com/es/home/> [13])

### 3.3 System Architecture

This section describes the architecture of the projected system. The video game is made with Unity2D engine, specifically with the 2020.2.3f1 version. For running games made with this engine the minimum system requirements are:

- Operating system
  - Windows 7 SP1+
  - macOS 10.2+
  - Ubuntu
- CPU: SSE2 instruction set support.
- GPU: Graphics card with DX10(shader model 4.0) capabilities.

The game has been developed and tested on a computer with these hardware and software features:

- Operating system: Windows 10 Home 2004
- Processor: Intel(R) Core(TM) i7-8700k
- Memory: 16GB
- Graphics: Geforce GTX 1070 ti

The resolution of the mobile device screen is not necessary to be a specific one. The game is designed to work with the most generic resolutions. But it is necessary to play in landscape instead of portrait mode.

### 3.4 Interface Design

The game user interface, GUI, is very simple and unobtrusive. It only shows the necessary buttons for the player to adjust the game features (See figure 3.2).



## WORK DEVELOPMENT AND RESULTS

### Contents

---

4.1	Work Development . . . . .	15
4.2	Results . . . . .	??

---

After having a solid idea about the work to be done, the requirements about the video game and the gameplay style it will have, in this chapter the development of the project will be described. How it has progressed, the mini goals that have been achieved and the changes that have been made along the way.

### 4.1 Work Development

In this section it will be discussed what has been done and the decisions that have been taken at certain times, but only the most relevant aspects of the project will be deepened. Great part of the information for the realization of this project comes from web pages, puzzles and tutorials on Youtube.

The first thing that was developed were the principal scenes. As the game takes place in a laboratory, the scenarios needed to be related to it. Thus, they are divided in three rooms, which each one includes: the main desktop of the scientist, the living beings' tanks with the exit and the tools room. All of them were made by myself with Photoshop. The sprites, including the objects the player will be finding along the game, are drawn in line-art style, because as the game is for mobile devices, the screen is smaller than pc, so it is necessary to be very differentiated with a first sight.



Figure 4.1: The Desktop scene.

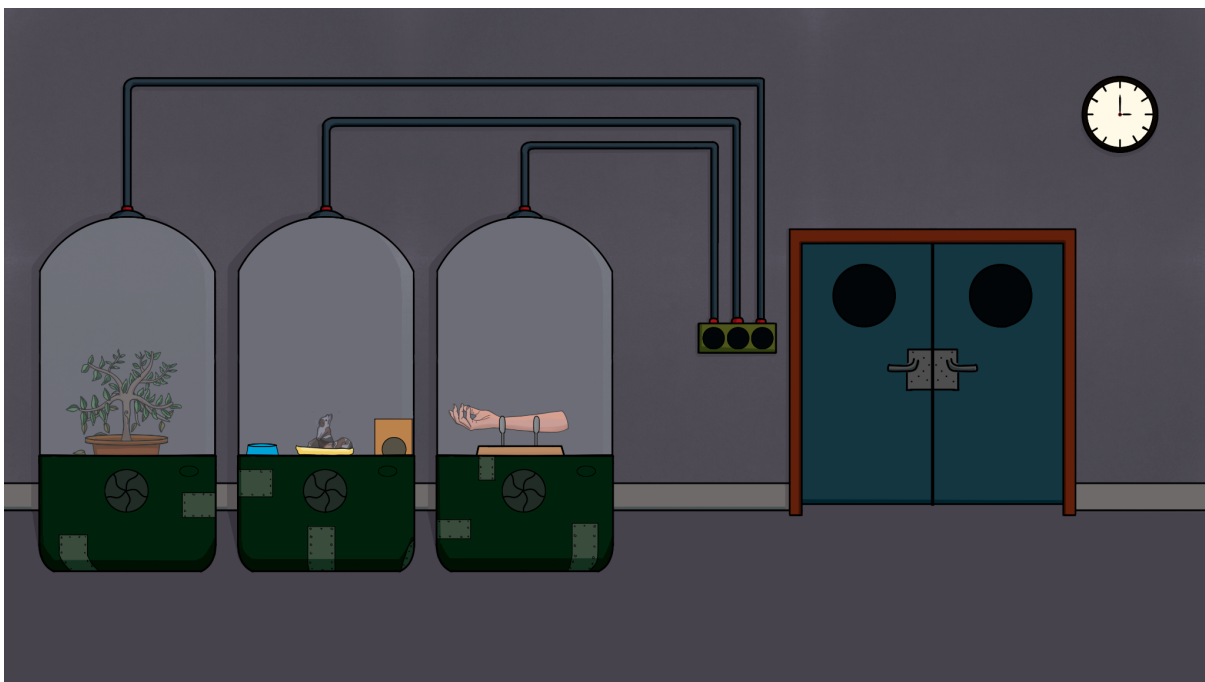


Figure 4.2: The Tank Scene.



Figure 4.3: The Restroom Scene.

#### 4.1.1 Interactivity

Related to the interactivity, the most UI elements in the game are buttons and toggles. The inventory will be always shown when the player is in some room. It will contain several toggles that will be hidden until the player finds an object. At this moment, the object will disappear from the scene and its linked toggle will be enabled. These objects in the scene act like a toggle group (a property in Unity). It means that only one can be selected at the same time. So the player can change the selected object just by clicking on the desired one. It will be surrounded by a white square.





Figure 4.4: The Inventory with the Fertilizer selected.

On the other hand, in the rooms, some sprites are buttons that make several functions: some of them, like the crafting table, just improves the objects the player has taken before like the syringe or fertilizer to inject the vacuum on them; another ones are the HUD buttons, which each one accomplish a specific function: Play, Return, Options, Observe, etc.

#### 4.1.2 Puzzles and Codes

In the game there is a puzzle which is called a Nonogram [14]. This type of puzzle works in the following way: the game has a grid with a lot of cells. The player can activate some of them by clicking, but some can not be pressed. As well, there are some numbers at the heading of each row and column of this grid. These numbers can be represented in several ways: there can be a unique number or two or more numbers separated. This means that, in this row or column, there are this number of cells pressed in the final solution, and if there are more than one number, means that, between them, it must be one or more non pressed cells. The main objective of the player is to press all the correct cells, but without pressing the non correct ones. This will reveal a drawing and the puzzle's door will be opened:

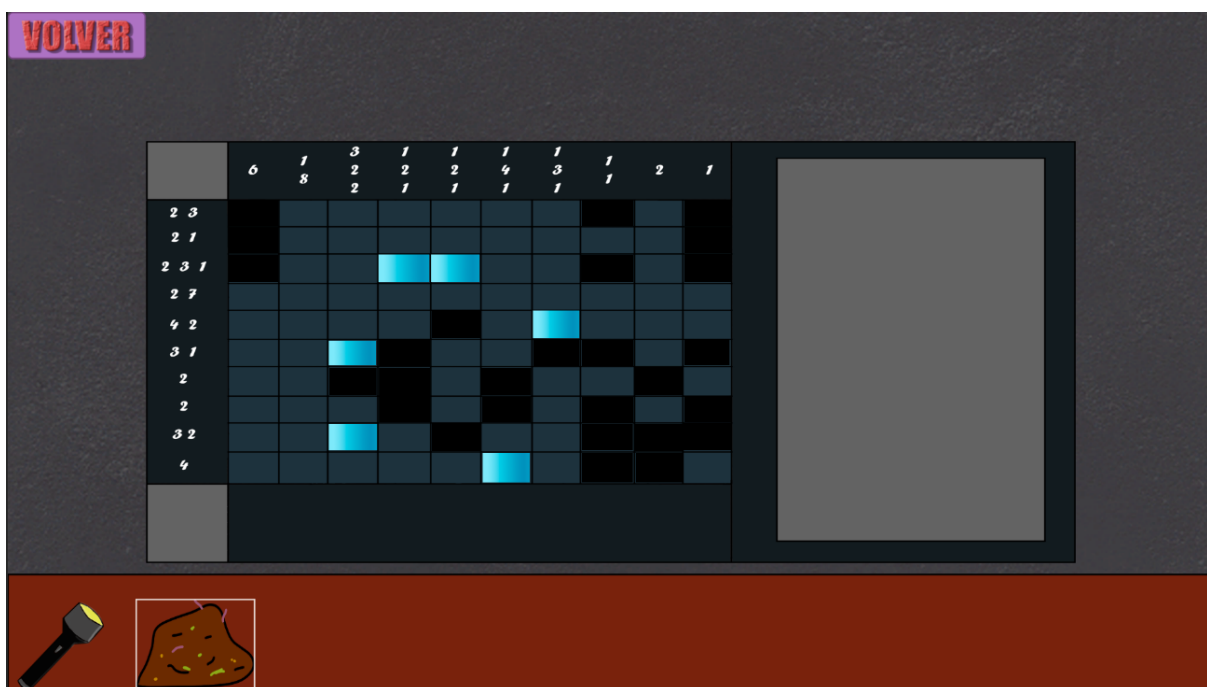


Figure 4.5: The Nonogram style puzzle.

This is the main idea for implementing this puzzle. First, it is necessary to create a list of toggles, they will be all of the cells. Each time a correct cell is pressed, the counter of the correct cells pressed will be incremented, and the same for the incorrect cells. When all of the correct cells are pressed and none incorrect one is, the puzzle will be solved:

Besides, there is a 4 colors code implemented, which performance is very simple: the player has to find the correct color sequence to be able to open the box. This code is hidden in any part of the game. For changing the values of each position of this code, it includes some arrows to change between the next and the latest color of the sequence, as it is shown in this image:

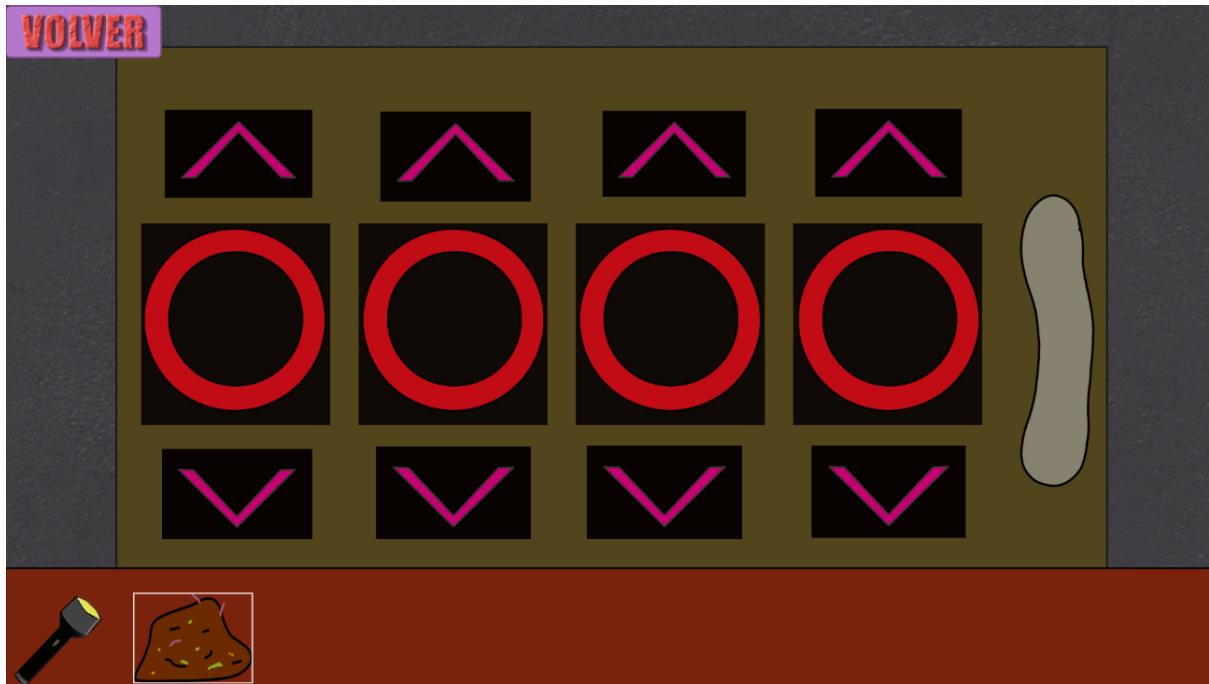


Figure 4.6: The Color code.

#### 4.1.3 Game logic

This section shows the main game logic of the game. This includes all of the incetactive actions that take the player to the end of the game. Because the game is an escape room, the logic is essential for the player to not get stuck in any part of it. For example, when the player is in a dark place, he needs to have the flashlight or at least a way to get it, because if it is not the case, the player can't proceed in the game. It took some time to think about the logic of the object, the place they will be found and its use in the game riddles.

#### 4.1.4 Beacons

The player, when presses the “Observe” button, scans to look for the room in which is. It is necessary to place a Beacon in the corresponding room for the data transmission. BLE beacons [15] just send a message to the device which is trying to connect to them. This message will be used to find out which room the player is in.

So, to configure each beacon, I used a mobile application called “iBKS Config Tool” [16]. This application can change the name of the device and the message the beacon sends to the bluetooth device when it is connected.

Once the beacon is configured, the next step is to program the interaction in Unity. To do that, Google has developed a Nearby API [17] which is free to use and lets all the programmers implement it in the language they need for their projects.

The behaviour of the code is as follows. First, when the player selects the “Observe” button, a new instance of the nearby messages’ event system is created. Once created, the game starts looking for the nearby beacons and receives their messages.

When the nearest place is located, the game sends the player to the desired location.

```
3 referencias
public override void Show()
{
    _nearbyMessages = NearbyMessagesEventSystem.Instance;
    _nearbyMessages.OnNearbyMessageFound += OnPlaceNearby;
    base.Show();
    _timeout = StartCoroutine(SearchTimeout());
}

4 referencias
public override void Hide()
{
    _nearbyMessages.OnNearbyMessageFound -= OnPlaceNearby;
    CancelTimeout();
    base.Hide();
}

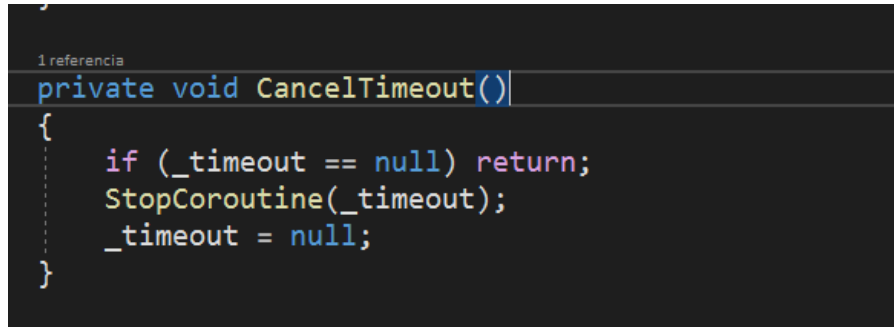
1 referencia
private IEnumerator SearchTimeout()
{
    yield return new WaitForSeconds(searchTimeout);
    _timeout = null;
    Hide();
    nearbyPlaceNotFoundPanel.Show();
}

2 referencias
private void OnPlaceNearby(NearbyMessage message)
{
    Hide();

    //Set desired scene to change
}
```

Figure 4.7: Beacon detection.

However, a problem may occur, because it is possible that a beacon near the device can't be detected. Therefore, it is necessary to establish a stop condition in the search function so that it does not try to permanently search for a beacon signal (see Figure 4.8).

A screenshot of a code editor showing a C# method named `CancelTimeout()`. The code is as follows:

```
private void CancelTimeout()  
{  
    if (_timeout == null) return;  
    StopCoroutine(_timeout);  
    _timeout = null;  
}
```

The code is displayed in a dark-themed editor with syntax highlighting. A small tooltip above the method name indicates '1 referencia'.

Figure 4.8: The search of nearby beacons is cancelled if the timeout is large.

#### 4.1.5 Problems

In this section I would like to mention the problems I have had throughout the development of the project.

One of the problems I had during the development was with the mobile application Joaquín (my DFP tutor) for configuring the beacons. When I tried to configure it, an error pop-up appeared on the screen. It says that my firmware was not updated, and when I click on the “update” button it seems to do nothing. After that, I tried to update the firmware. To do that, it is necessary to download the beacon application Kotlink.io [18], but when the app detects these beacons, a pop-up message appears that says “you are not the owner”, and doesn't let you update it.

I had some problems with the resolution of the game. At first, I was not keeping in mind that problem, and when I did the first build to test the game, the only thing that appeared in the scenes was the background image. So, when I realized that, I spent a lot of time configuring every single canvas of the scene to be adjusted to each resolution in which the game is played. Now, the mobile screen does not matter anymore.

In Android Studio, it is very easy to configure any app or game to be represented in portrait or landscape screens, but in Unity it is not that easy. At the moment, as it is not a critical error, I can go without taking account of it and focusing on more important issues.

## 4.2 Results

Based on the objectives cited in section 1.2, I can ensure that most of the main objectives have been completed including also the secondary ones.

Thanks to the development of this project I have managed to learn more about the creation of 2D escape room style games, with all that it includes (game logic, flow of the game, etc). As well, during this development, I have learned so much about the process of the creation of a vaccine, not in a deep way but enough to know about the great job the scientists do for protecting the society.

The game flow is another thing in which I have spent a lot of time, because in this type of game it is the most important aspect. That thing can make the difference between a fun game and a tedious one. And, in this case, some people have tested the game and they were able to proceed with the game perfectly.

Another important thing that I have learned has been how to program mobile games using Unity. In my opinion, this was a great objective because in these 4 years I have learned a lot about this engine, but I haven't learned anything about programming and exporting these games for use in mobile devices.

And finally, and the most important thing, is that I am still learning how to configure BLE beacons for using them in Unity. This process is so tedious because there are a lot of steps to configure and include them in the game, but I think that the result is totally worth it, because the interactivity in the real world makes the game to be more immersive for the players

My last objective was to have a demo available that would show the result of having achieved the above objectives. The demo is available for Android here:

[https://drive.google.com/file/d/1-HdDE\\_4cxfRowmRwol07z\\_Y07uomS-Oy/view?usp=sharing](https://drive.google.com/file/d/1-HdDE_4cxfRowmRwol07z_Y07uomS-Oy/view?usp=sharing)

You can also access the entire project from Git:

<https://github.com/OscarSegui/Laboratory-93>



## CONCLUSIONS AND FUTURE WORK

### Contents

4.1	Conclusions . . . . .	15
4.2	Future work. . . . .	??

In this chapter, the conclusions of the work, as well as its future extensions are shown.

### 5.1 Conclusions

I think the DFP is a great form for every student to show what he has learned along these 4 or more years of this degree and, of course, a great presentation sample for all the companies in which the student wants to leave a CV when he or she is looking for a job. As well, in this project it is reflected in the personality of the student, as it is his most important game he has realized until this moment.

What I am clear about is that the world is constantly changing in terms of technology and interactivity with the environment. And, in the case of the games, I am sure this is the future for them. This is the great point where the games want to arrive at any moment, the most interactive experience for the players.

And, finally, the most important part is everything that is learned by doing this final degree project, since it is where all the knowledge obtained during these years is really shown and remembering during the project all the moments for which each one has passed up to this moment.



## 5.1 Future work

The final result of the game is very pleasant for me, but I would like to have more time to improve in every single aspect. For people without great drawing skills, but I always try my best to accomplish my expectatives mainly, and of course, with the others as well. In my case, I was very busy this semester and my intention was to create a basic game but with all the features that needed to be included, adding my own ideas to the project.

The game is almost done, but for example, some improvements like animations in the elements, adding a tutorial for the main scene or adding features like a video intro can improve the player;s game experience, since all this improves the visual quality of the video game and makes it more attractive.

I'm still improving the game. With the help of friends who test the game, I'm correcting errors, removing or adding things, etc. The main work is almost done. So now I'm polishing the game, making it entertaining and getting a flow in the gameplay.



---

## BIBLIOGRAPHY

- [1] Minijuegos.com. <https://www.minijuegos.com/>. Accessed: 2020-07-01.
- [2] The Escape Room. “What is an Escape Room”.<https://theescapegame.com/blog/what-is-an-escape-room/>  
Accessed: 2020-07-01.
- [3] OMS. “¿Cómo se desarrollan las vacunas?”  
<https://www.who.int/es/emergencias/diseases/novel-coronavirus-2019/covid-19-vaccines/how-are-vaccines-developed>  
Accessed: 2020-07-01.
- [4] Line art. “¿Qué es y cómo hacer dibujos de arte lineal?”  
<https://www.crehana.com/es/blog/disenio-grafico/line-art/>  
Accessed: 2020-07-01.
- [5] Gantt Project. Free Project Management Application. <https://www.ganttproject.biz/>  
Accessed: 2020-07-01.
- [6] Unity. <https://unity.com/es>  
Accessed: 2020-07-01.
- [7] Visual Studio. <https://visualstudio.microsoft.com/es/>  
Accessed: 2020-07-01.
- [8] Github. Github Desktop <https://desktop.github.com/>  
Accessed: 2020-07-01..
- [9] Adobe Photoshop CC 2020. <https://www.adobe.com/es/products/photoshop.html>  
Accessed: 2020-07-01.
- [10] Trello. <https://trello.com/es>  
Accessed: 2020-07-01.
- [11] Unity Asset Store. <https://assetstore.unity.com/>  
Accessed: 2020-07-01.
- [12] Wikipedia. Point and Click. [https://en.wikipedia.org/wiki/Point\\_and\\_click](https://en.wikipedia.org/wiki/Point_and_click)  
Accessed: 2020-07-01.
- [13] Wikipedia. Point and Click. [https://en.wikipedia.org/wiki/Point\\_and\\_click](https://en.wikipedia.org/wiki/Point_and_click)  
Accessed: 2020-07-01.

[14] Creately. Diagram Maker. <https://creately.com/es/home/>

Accessed: 2020-07-01.

[15] BLE Beacons. “Using Nearby with Beacons (100 days of Google Dev)”.

<https://www.youtube.com/watch?v=7InjGgP15E>.

Accessed: 2020-07-01.

[16] IBKS Config Tool. “IBKS Config Tool User Manual”.

<https://accent-systems.com/support/knowledge/ibks-config-tool-user-manual/?v=04c19fa1e772>

Accessed: 2020-07-01.

[17] Google Developers. Google Nearby. <https://developers.google.com/nearby>

Accessed: 2020-07-01.

[18] Kontakt.io. Beacons & Tags. <https://kontakt.io/beacons-tags/>

Accessed: 2020-07-01.

---

## APPENDICES

2.1	DFP task planner . . . . .	16
3.1	Image that shows the Game User Interface . . . . .	18
3.2	TheMain Menu screen . . . . .	19
3.3	Case Use Diagram . . . . .	27
4.1	The Desktop scene . . . . .	29
4.2	The Tank Scene . . . . .	29
4.3	The Restroom Scene . . . . .	30
4.4	The Inventory with the Fertilizer selected . . . . .	31
4.5	The Nonogram style puzzle . . . . .	32
4.6	The Color Code . . . . .	33
4.7	Beacon Search . . . . .	36
4.8	The search of nearby beacons is cancelled if the timeout is large . . . . .	37