Video game design and development degree

Lifelines
Development and implementation of the mechanics of a First Person
Adventure video game

Final Degree Project's Technical Report

LIFELINES

Marc López Gramaje

Final Degree Project

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

May 30, 2020

Supervised by: Diego José Díaz García

ACKNOWLEDGMENTS

To start with, I would like to thank Alex, Carlos and Mica for their participation in this project and thank Diego for his guidance.

I also want to thank my parents for their constant support during these months.

I also would like to thank Sergio Barrachina Mir and José Vte. Martí Avilés for their inspiring LaTeX template for writing the Final Degree Work report, which I have used as a starting point in writing this report.

ABSTRACT

Lifelines is a First-Person Adventure (FPA) game that narrates the last days of an old man who, after getting away from their loved ones, decided to spend his last moments in his parents' old boat and recall all his life memories. The work consists in the implementation of game mechanics and the creation of 2D art. The implemented mechanics range from the simplest to the most difficult ones and have been developed in the best possible way. Academically, this document consists of the final degree project report of the Video game Design and Development bachelor's degree at the Jaume I University.

CONTENTS

1

INTRODUCTION

Contents

When my friend Alejandro Estaben and I agreed to work on a joint final degree project, after discussing about what could we do, we decided to create a First Person Adventure (FPA) video game. This is a video game genre that focuses in the narrative and which main mechanics consists of exploring to discover the story.

The origin of this genre is the First Person Shooter(FPS) one, where the main goal was to defeat enemies using the shooting mechanic, which were also the main one. But in the FPA games, the main mechanic is to walk, that is why this genre is also colloquially known as "walking simulator". Of course, it is not the only mechanic, since there are other ones, but it is thanks to which the game move on.

Since this kind of games have to count with a consistent narrative part, we thought that we should include in the development of this project a third person that was responsible of making this part. That was when we decided to contact with our friend Carlos Granell, so he could start to create the story of the game while Alejandro focused on the art part and the sounds and I took charge of the programming part as well as 2D art.

1.1 Work Motivation

The main motivation that made us develop this game was the fact that, the previous year, Alejandro and I worked on another joint project where the results where not as

good as expected due to our inexperience and we wanted to create a game with a more solid idea and with more experience.

Since one of our main goals was to develop a game that was polished enough, we decided to create a FPA game, which is not too difficult to manage while developing it. We thought that a good theme would be a journey without a destination where the player would recall the memories of a character. After thinking about this we also agreed to create a game with little space because we had no time to model bigger spaces and more characters nor to program more mechanics than needed.

For all these reasons we decided that the game would be about a boat journey where an old man starts to recall all his memories through different objects that are on the boat. Given that all these objects are available from the start of the game, the player can discover the story in any order or even discover just some parts of it, so he/she has complete freedom.

Another aspect that motivated me to participate in a joint project was the fact that I wanted to completely focus on the programming part of a video game and learn as much as I could about how to properly develop every technical aspect of it in *Unity*, so I considered this a great chance to accomplish that goal.

1.2   Objectives

The main objective of every team member was to develop a game that was as polished as possible, taking into account the time and the resources we had, and document it in this report, that allows us to obtain the Video Game Design and Development bachelor's degree at the Jaume I University.

Next, while the 3D designer created the 3D models and the sounds and the narrative responsible wrote the story, the game mechanics and other aspects of the game as well as the 2D art for the interface were created. These objectives can be classified in:

- **Interaction with objects:** To achieve a good immersion in this kind of games it is important to allow the player to touch and interact almost every single object in the scene and receive visual and/or auditory feedback. That is why it was decided to add interactions with many objects, since they are also important to discover the story in some way. There are also some interactions that are not key for the story but that have been added so the player has different activities to do.

- **Environment:** One important objective that needed to be accomplished was to create an environment that feels active, that is, that the world around the character keeps moving without the need of any interaction, like the passing of the days, the movement of the sea, or the clouds. An aspect that joins the previous objective and this one is an AI that moves by itself but can also be interacted.

- **Art:** Since the 3D designer models were pretty simplistic, the 2D art made had to be simplistic as well. That is why it was decided to create 2D assets with little variety of colors and simple forms.

1.3 Environment and Initial State

This project has been made in group, which members are Alejandro Estaben, Carlos Granell and myself, Marc López. Alejandro is the responsible of the art, Carlos is the responsible of the narrative, and I am the responsible of programming. We decided to organize this way so we could work at the same time without bothering each other.

During the development we have followed a schedule, where Alejandro and I have worked 4 hours a day during 4 days a week, since the rest of the time we had to do external practices. On the one hand, to keep track of the current state of the video game and decide which were the weekly tasks to do, we met every Monday. On the other hand, every Thursday we met with Carlos to define different aspects about the story and what 3D models or functionalities we should implement to complement it.

Also, every two weeks, Alejandro and I met with our final degree project tutor, Diego Díaz, so we could show him our progress to decide what we had to fix or improve and what we had to do the next two weeks.

Before starting the development and after agreeing about this work plan, we divided every aspect that our video game should contain so we could easily plan weekly tasks, prioritize them and even remove them if necessary. We used *Discord* as our main communication tool during our work sessions, given that we could easily broadcast each other screen and communicate in real time.

The initial state of the project was based in a free asset [17] for *Unity* that offered a pretty realistic first person point view and basic object interaction that has been modified over time. Once we had this starting point, we began to model, program and define the story.

2

PLANNING AND RESOURCES EVALUATION

Contents

This chapter is the most technical part of the work. Here the followed planning is explained as well as the software and resources that have been used used.

2.1 Planning

In this section, it is explain how the time in this work has been invested divided by tasks(see Table 2.1) and a Gantt chart.(see Figure 2.1). In this planning almost every task is an implementation of a functionality or minor design tasks within *Unity* like create menus or arrange the animations. Furthermore, there is also some 2D art tasks added to complement the programming tasks in the project. Of course, some time from the total has been dedicated to writing mandatory documents, like this one.

Task	Type	Hours
To make the game design	Design	5h
To modify First Person Controller interaction scripts	Programming	15h
Day/Night cycle	Programming	30h
Stars particle system	Programming	10h
Random clouds generator	Programming	10h
Boat driving system	Programming	15h
Book turning pages system	Programming	30h
Book text and textures system	Programming	15h
Seagull AI system	Programming	40h
To draw UI assets	Art	20h
To create menus typography	Art	5h
To create two main menus	Design	15h
To implement all objects interactions and animations	Programming/Design	10h
Save and load system	Programming	25h
To write memory	Documents	40h
Presentation	Documents	10h
Technical proposal	Documents	5h
Analysis and design document	Documents	5h
Total Hours		305h

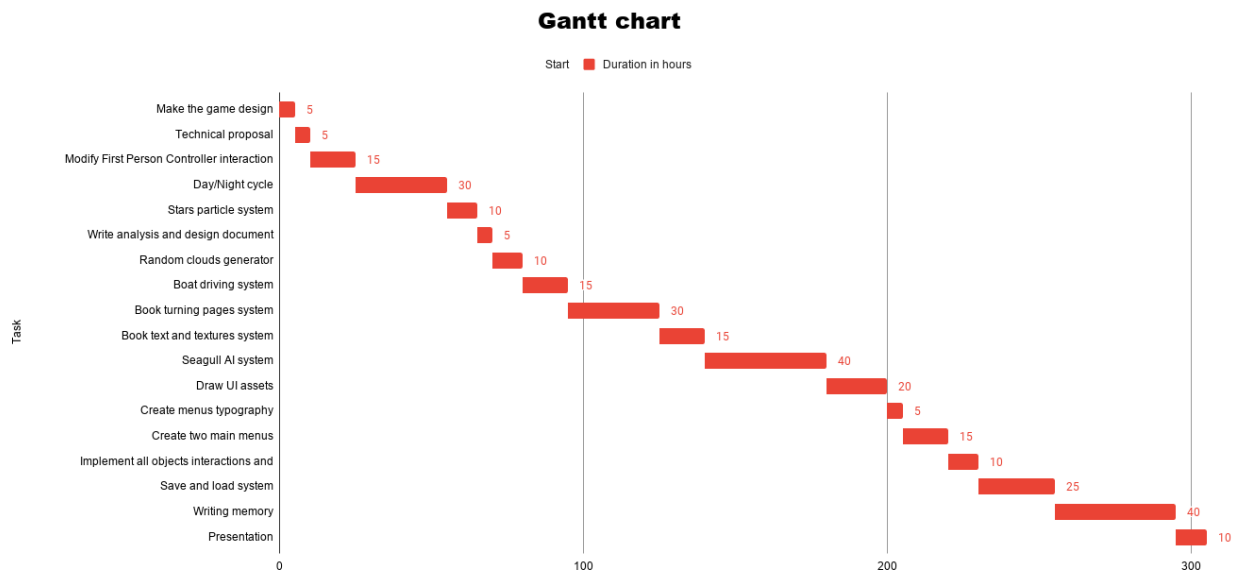Table 2.1: Tasks

Gantt chart

Start ■ Duration in hours

Figure 2.1: My Gantt chart

2.2   Resource Evaluation

Here, an analysis about the theorical amount of money and time that the project would cost is made. The amount of money per person is different depending on each job since some are harder or need more time than the others.

On the one hand, in this project the number of hours that were dedicated by the designer and the programmer was 80 hours a month for four months, so each one spends a total amount of 320 hours in the development. About the costs, the junior programmers on Spain earn 8.39 euros per hour on average [10], so its final cost would be 2684.8 euros, the 3D designers earn 9.4 euros per hour on average [9], so its final cost would be 3008 euros.

On the other hand, the third person works in the narrative part of the game spending 160 hours in total, since this part does not require as much time as the programming part o the artistic part. About the cost, the narrative responsible earn 10.25 euros per hour on average [4], so its final cost would be 1640.27 euros.

The total cost of the project would be 7333,07 euros.

Next, it is important to differentiate between hardware and software:

- **Hardware:** Even if the used computer does not have the most actual components it is still pretty decent and has a price of about 700 euros:

  - **CPU:** Intel Core i5-6600

- **GPU:** Nvidia GTX 1060 3GB
- **RAM:** 8GB

- **Software:**

  - **Unity(Free):** It is a cross-platform game engine developed by Unity Technologies to create 2D and 3D games, even VR games.[14]
  - **Visual Studio(Free):** It is an IDE developed by Microsoft that is compatible with almost every highly used programming language, like C sharp, C++ or .NET. It is pretty easy to use with *Unity*, so, obviously, it is the script writing program that has been used.[12]
  - **GitHub(Free):** This GIT repository hosting service has been really useful for this project since two people are working in the same one.[11]
  - **Adobe Illustrator(19,66 euros/month):** This program is a vector graphics editor that has been mainly used to create all the 2D art. It is a pretty useful tool if the user does not count with enough artistic skill.[2]
  - **Adobe Photoshop(19,66 euros/month):** It has been mainly used to texturize the 2D art since with Adobe Illustrator the results were not too good. Apart from that, this program has only been used by the 3D designer.[3]
  - **Overleaf(Free):** This online LaTeX editor is the one that has been used to write this report since the used template has also been written by using LaTeX.[1]
  - **Google Docs(Free):** This tool to create and edit documents has been pretty useful to write some ones that did not require too much quality. It has also been used to write down the results from the weekly meetings with the team.[8]

3

SYSTEM ANALYSIS AND DESIGN

Contents

In this section, the different parts of the game where it has been participated are explained, that is, the artistic part, the game design part and the interface design part. It is also explained why each decision has been made and how it affects the game. In the next section it will be explained how each part has been developed with more detail.

3.1 Artistic Design

At the start it was discussed about the visual style that the game should have, and, according to our capabilities, it was decided that a good-looking and simplistic style would be a low-poly one with plain colors. While the main artist took care of the 3D modeling, two parts related to the art were designed:

- **2D art:** The game logo, the menus typography and some UI assets, like dropdowns or sliders, were drawn. Since the 2D art had to fit the 3D models style, it was decided to use pretty simple forms and colors, like in many similar games, for example *What Remains of Edith Finch*(2017)[13](see Figure 3.1). After checking some references like this one, it became obvious that the aesthetics of the 2D art had to be based on paper, more specifically a notebook, given that most of the story would be told by letters and a book instead of experience the story itself, so

using a paper-based art style that evoked these elements within the game made a lot of sense.

The first design that was made was the game logo, since the rest of the designs had to follow the same style. At first, it was tried to create a colorful design with many details, but it did not fit the 3D models art style. That is why it was decided that the 2D art had to be quite simple and with not too many colors, another reason why the designs had to be based on paper. After the checking that this style fitted in the game, the rest of assets were designed the same way.
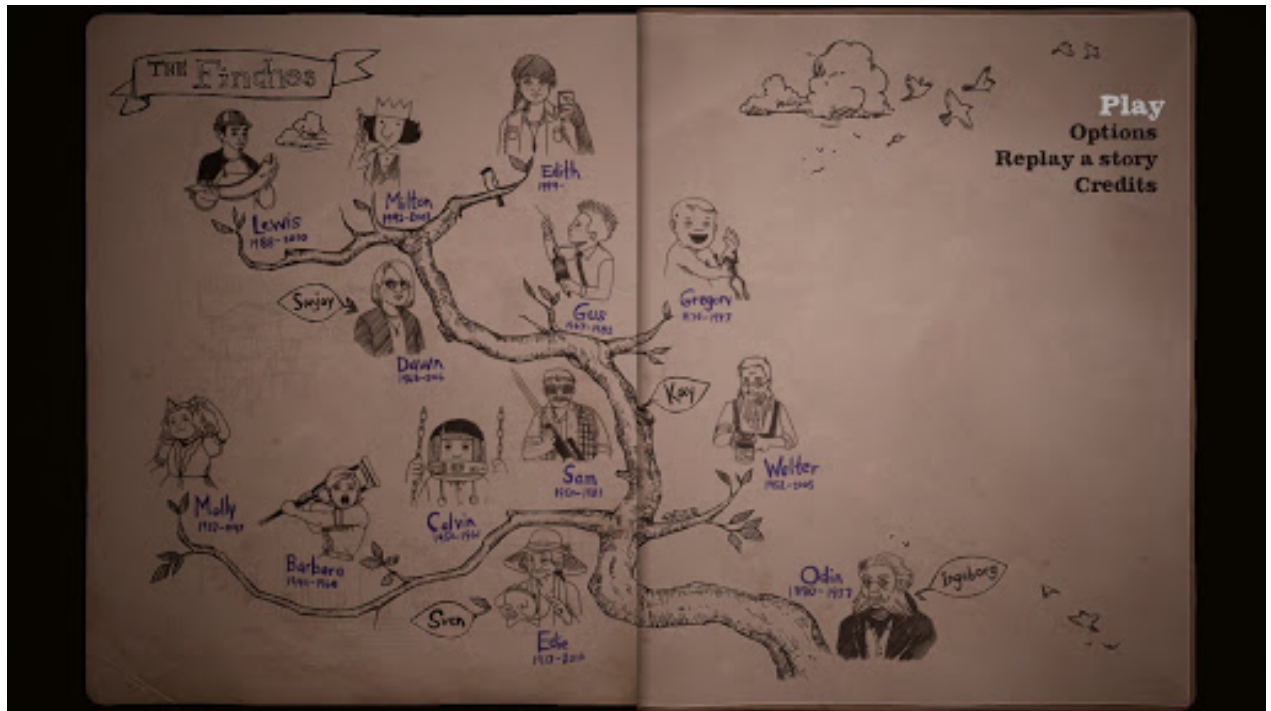

Figure 3.1: *What Remains of Edith Finch(2017)*

- **Lighting:** This was one of the most difficult parts since it had to fit in with the day/night cycle as well as with the artificial lights. During the day, the environment is pretty bright due to the Sun and pretty dark at night except for some light bulbs and the Moon, simulating a realistic day and night, since it was wanted to partially represent a real world (see Figure 3.2). There are two light bulbs in the outdoor area of the boat that are automatically turned on at night so the environment is not completely dark, given that the other two light bulbs that are inside the cabin and the ship cockpit can be turned off at any time. These two indoor light bulbs are also helpful to indicate the player the two main areas of the boat, since the core elements of the story and the survival elements are in these areas. The mood

of the light is also important when it comes to bringing calm to the environment, since they are neither very strong nor very soft.
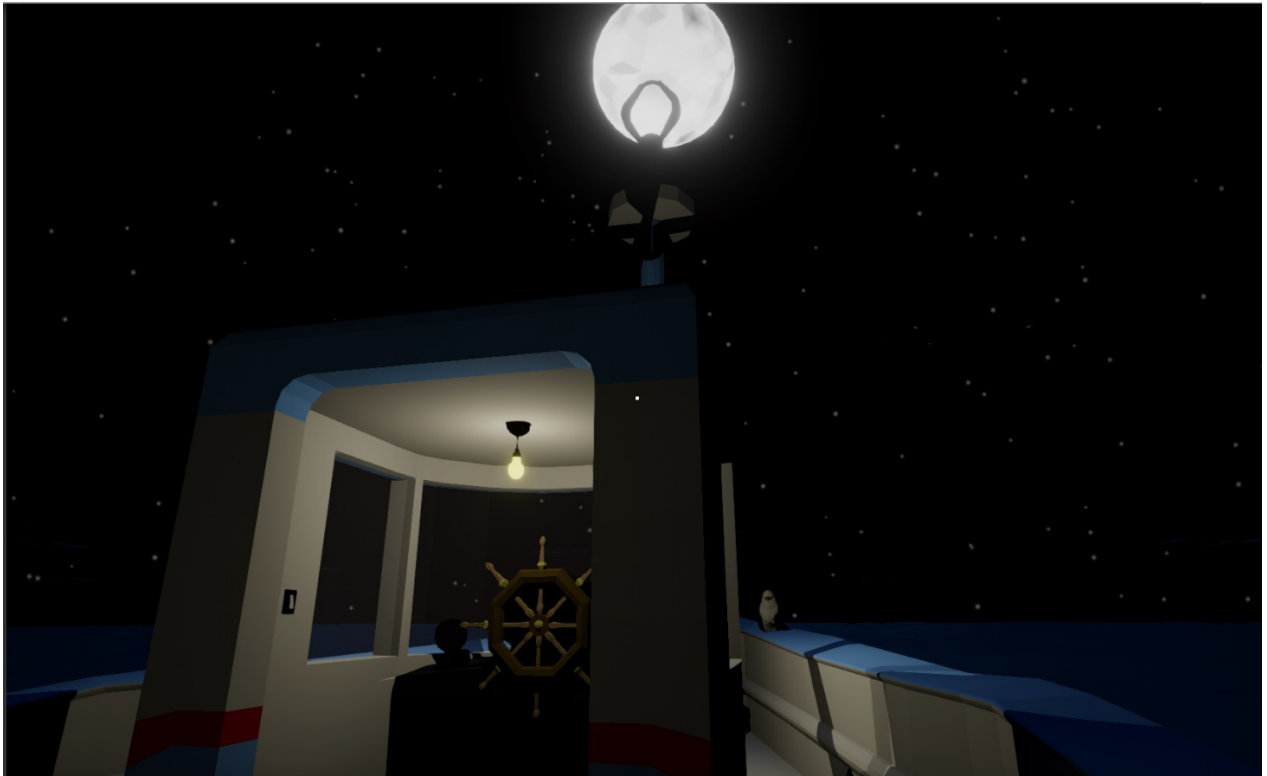

Figure 3.2: Lighting at night

3.2   Game Design

- **Mechanics:** As the main programmer, most of the main mechanics have been developed, such as reading or interacting with some objects, while the 3D designer made some secondary ones, like sleeping or watering plants. The most important one is the reading mechanic given that the completeness of the story is narrated with a notebook, some letters and notes and a mobile phone. Within this objects, the most complex one is the book, since the player can interact with it just like it was a real book where he can turn pages and read each one (see Figure 3.3). This is a pretty common mechanic in games of this genre that forms an important part of the core of the game since this kind of games do not necessarily have a playful component and just want to tell a story. Therefore, our game counts with some trivial interaction actions such as examining objects or smoking and core actions like reading.
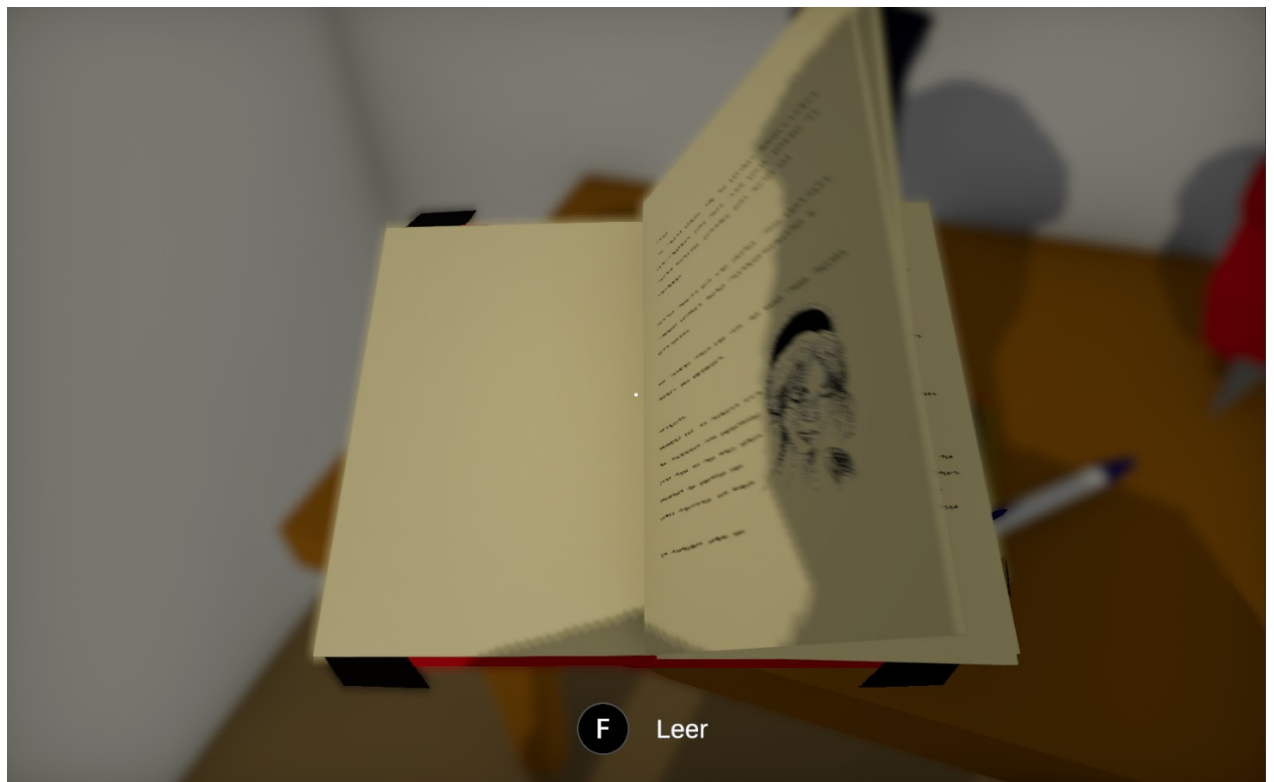
Figure 3.3: Book

Among the interaction actions, different ones have been made:

1. Examine objects: This way the player can pick up some objects distributed throughout the ship to observe them in detail rotating them. Even if this is a trivial action, it supports the story because these examinable objects have some relevance in it.

2. Boat driving system: This mechanic allows the player to redirect the ship, but it is just another trivial action, because the ship actually rotates but it does not move forward (see Figure 3.4).

3. Open/close closets,chests...: As some letters or other objects can be inside a closet, a locker or a chest, the player must be allowed to open or close everything to find them.
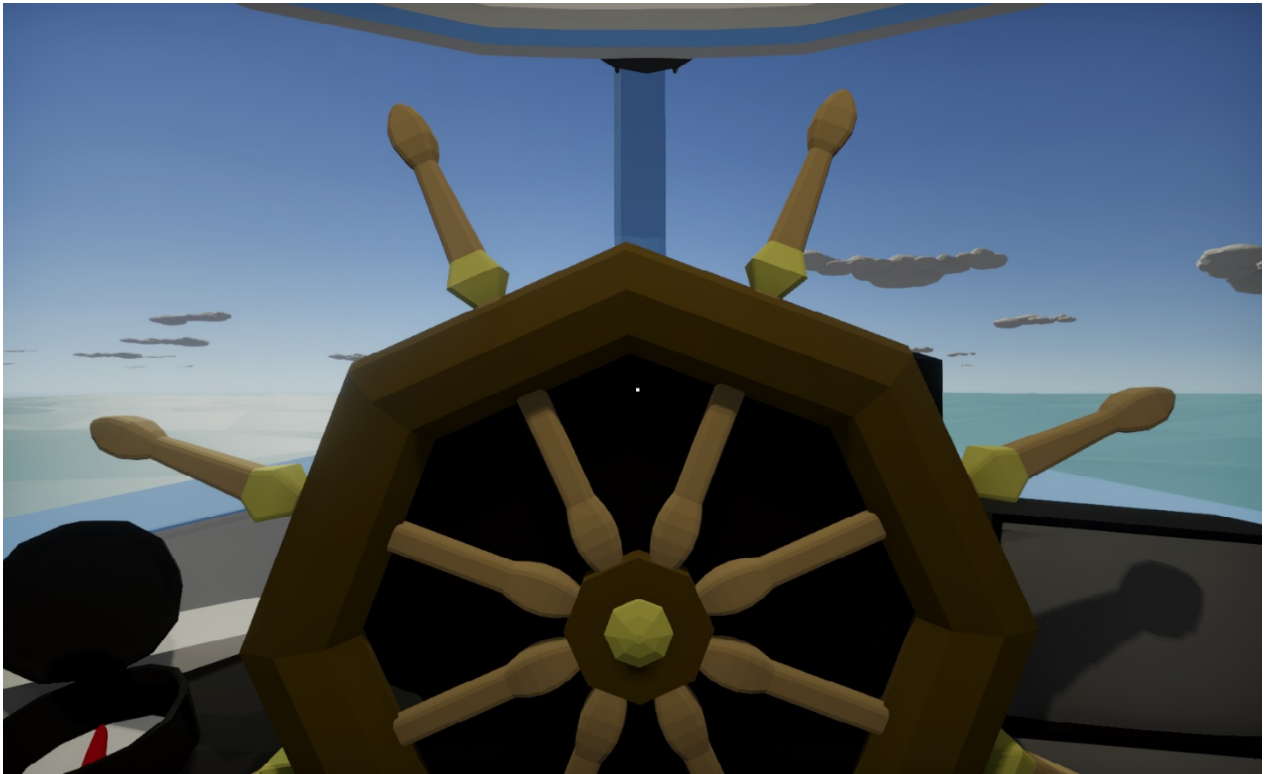
Figure 3.4: Boat driving system

The rest of actions have been made by the 3D designer: smoking, watering plants and sleeping, which basically consists on smoking a finite number of cigarettes, helping some tulips to grow up and advancing time, respectively.

- **Game Feel:** The most important goal that was wanted to be accomplished in this project was to offer a mostly realistic experience where the player felt as the character did, so this part has a special importance. That is why the movement, and more concretely the point view, had to be as real as possible to obtain an immersive experience. To achieve this the camera had to feel like the real sight of a human being with properties like swaying when walking or a little blurring when turning its head.

  Also, if the player examines an object, everything behind it is blurred to simulate depth of field, focusing his sight in the picked up item(see Figure 3.6).

  In addition, to support the sleeping mechanic, using the interface of *Unity*, the screen fades to black while blurring, imitating the eyes of a person.

  Finally, since it is played from a first person point of view, the sounds are the other important aspect to complement the gaming experience, and are included in

almost every interactable object and every action that the character can do, like watering plants, opening closets, or even walking. Given that there are also some important actions that are not explicitly explained, the sounds are used to offer some feedback about the character status, like some belly roars to indicate hunger.

- **Level Design:** In this project, since the controls are pretty intuitive, there was no need of adding a tutorial level, so the player is put on a boat to discover by him/herself which are the objectives. During all the game, the player can only move inside a boat and see the days pass by while performing some secondary activities like watering plants or increasing the confidence with the seagull, or main activities like reading or use the mobile phone.

At the start there are different objects scattered across the ship that do not appear to have any meaning, but as the player progresses through the story, they all start to have a reason to be there. Also, there are no objects that are locked or anything like that as the game progresses, everything is on the boat from the start and is accessible, that is because it was wanted to tell the story but in the order the player decides.

The ship where the game occurs is basically divided in three different places(see Figure 3.5):

1. The ship cockpit: Here the player can drive the ship and examine the compass. This is the simplest area.

2. The deck: This is the widest area and is basically dedicated to contemplate the landscape, try to approach to the seagull and water the plants in addition to examine some objects and letters.

3. The cabin: Here are the most important elements: the book, the bed, the mobile phone and some letters. In this place is where the player can discover everything about the story stored in the book and the mobile phone and save the game by sleeping.
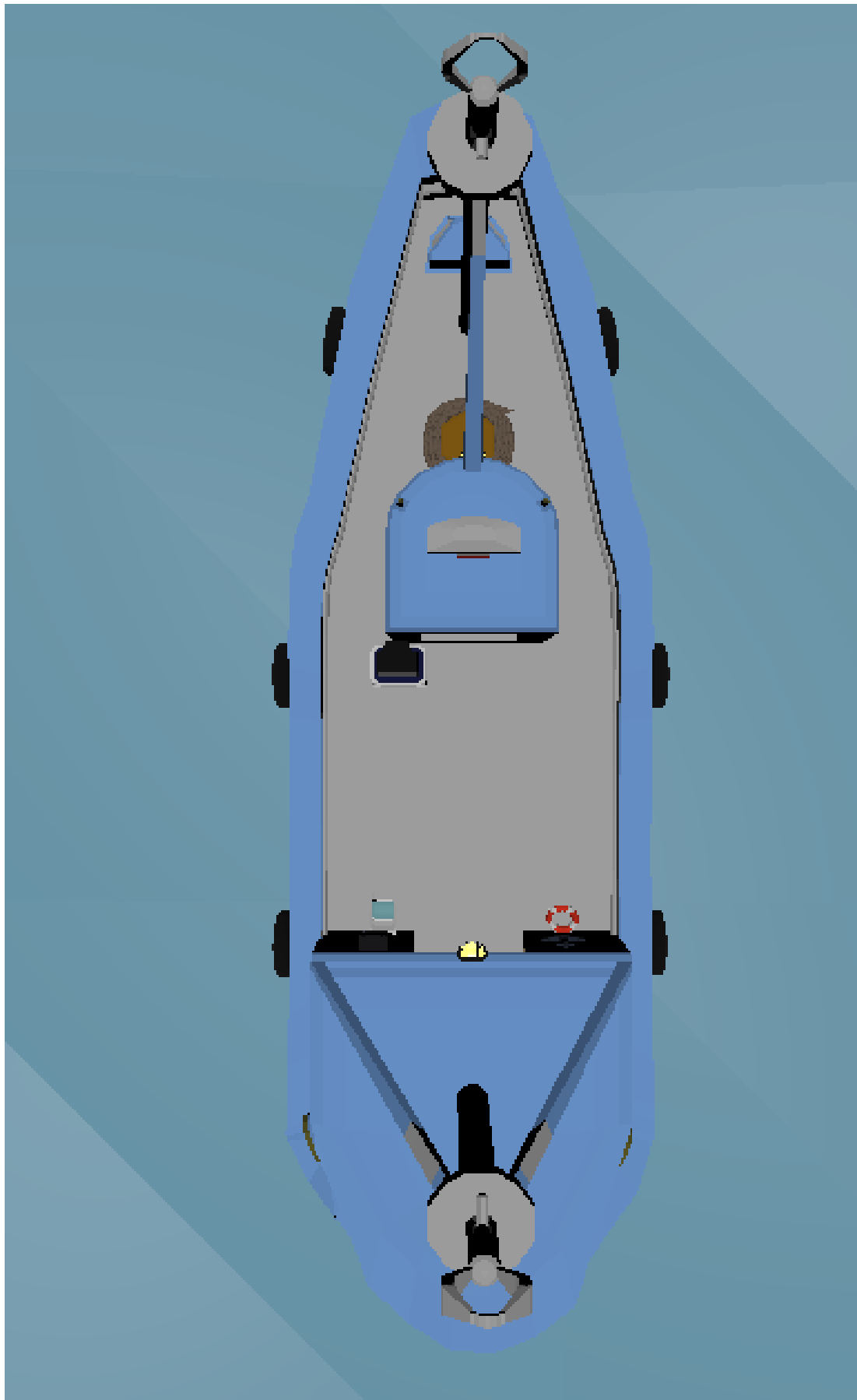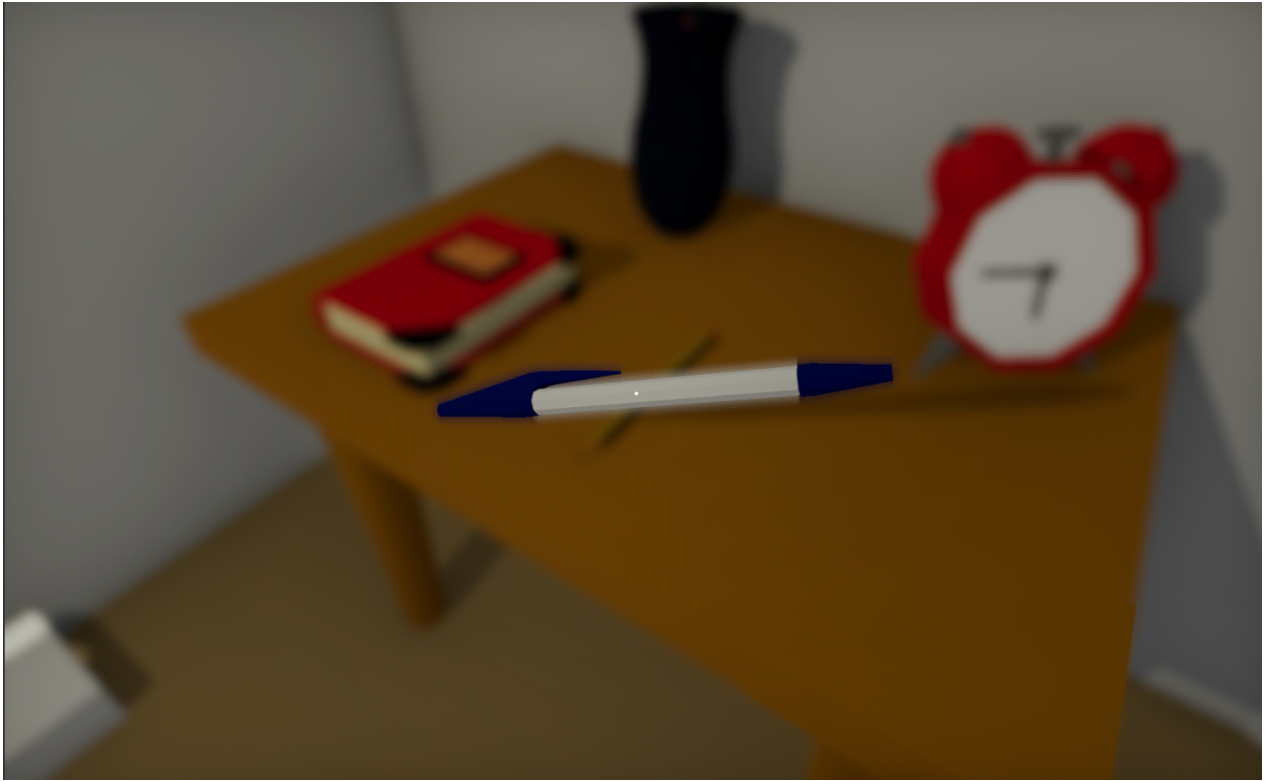
Figure 3.5: Ship parts

Figure 3.6: A blur generated when examining a pen

3.3   Interface Design

Now, about the interface, since it is wanted our game to be simplistic there is just the necessary elements, that is, two main menus, a pause menu and a black screen that fades between scenes and when sleeping.

First, one main menu is a basic one projected over a table with a paper and has three main buttons: a new game button, to start writing a letter and then start the game, a settings button and a button to exit the game (see Figure 3.7). Inside the settings menu the player can set the volume, the resolution, the graphics quality and the full screen option (see Figure 3.8).

Figure 3.7: One of the main menus

Figure 3.8: Game Settings

Next, the other menu has the same buttons and settings options as the first one, but in this case it also has a continue button and is a menu placed in a 3D space on the ship of the game. Also, a little feature added in this menu is a rotation that the camera does to travel between the main menu and the settings menu (see Figure 3.9).
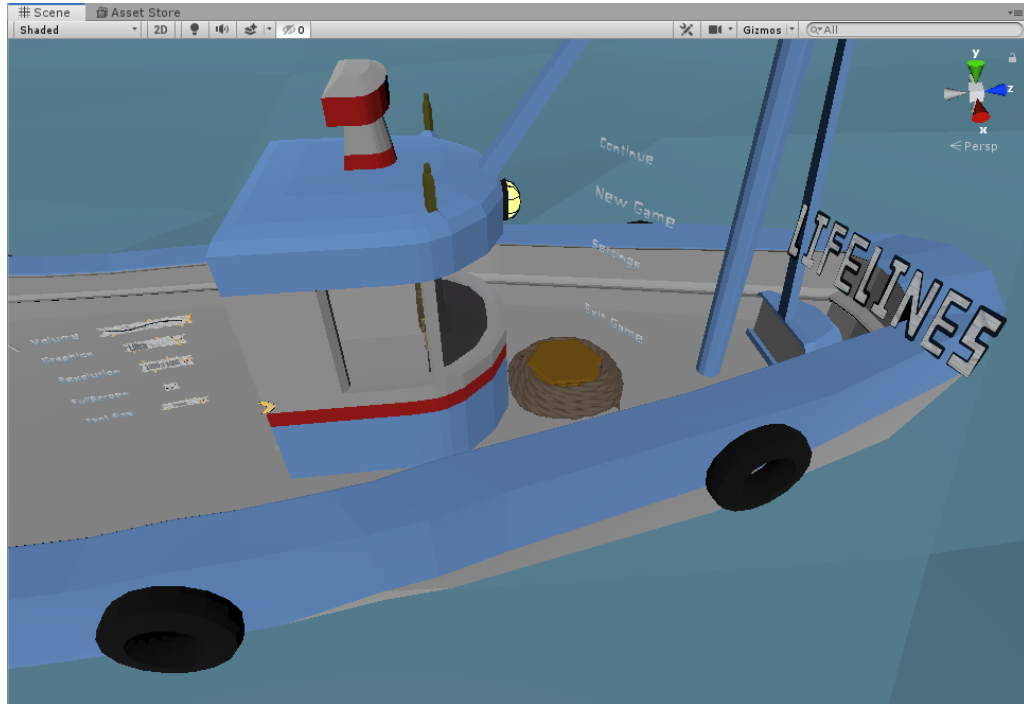

Figure 3.9: The 3D main menu

Lastly, the pause menu also counts with the settings menu along with a resume button and a button to exit to the main menu.

In addition to the menus, the interface is also used to display a black screen, that fades to make smooth transitions between scenes, and a blur effect that activates during the sleeping process, as a background for the pause menu and as visual trick to focus the sight in the object that is being examined if there is any. It is also used to display the texts that can be contained by some elements like the letters, since they are not directly readable within the game.

Apart from these few cases, there are no more interface elements because it was thought that it would be better to have a clear screen with nothing else during the gameplay so the immersion was greater and also because there was no information that needed to be displayed.

4

GAME DESIGN DOCUMENT

Contents

4.1 Game Overview

4.1.1 Game Concept

This games consists of a man named Oscar who has lost everything, from people to his jobs. He will go on a journey where he will recall all his memories, mainly focusing on critical moments of his life, like when he lost his job or when he broke up with his girlfriend.

The whole game takes place on a boat that belongs to Oscar's parents and that contains many objects, notes and a book that will help him to awake all his memories. He will be alone during all the journey except for a seagull with which he will have to gain confidence so he can get closer to it.

The memories can be obtained at any time and in any order, so the player can discover them in a non-chronological order and even not obtain every one of them if he/she runs out of time.

4.1.2   Genre

The genre of the game is First Person Adventure(FPA). This genre consist of a first person point of view where the player is told a story of any type. Unlike the First Person Shooter genre or any other first person genre, in the FPA there is almost no action, all the mechanics are quite simple and the game main objective is to tell the player a story. An example of this genre is the game *The occupation*[7] (see Figure4.1).

Figure 4.1: The occupation

4.1.3   Target Audience

The main target audience is young people (+16 years old) and adults. Even though the game itself can be played by anyone, the story that is told contains offensive language, drugs and alcohol are mentioned and there are implicit sexual connotations. Also, one optional action is smoking, so this a limitation about who can play this game.

4.2 Gameplay and Mechanics

4.2.1 Gameplay

Game Progression

The only progression in the game is a narrative progression, since the only goal in the game is to discover all the story, which is completely available from the beginning. Inside the game the player can also notice a time progression due to the day/night cycle, but this is just an addition to give a bit more of realism to the game, given that the only thing that truly changes between each day is the status of the character. This status inevitably gets worse each day, but can be slowed down if the characters eats, drinks and sleeps every day.

The game ends when the character dies, which can happen before the player gets to discover all the story if he/she does not take proper care of the character by making him eat, drink and sleep. So the player can know which is the status of the character, the borders of the screen will start to get darker as it gets worse, so he/she can get enough feedback about if he/she is doing all the required actions.

Play Flow

The player will have total control over the character throughout the game, except when he sleeps. During the gameplay, the player can move the character as much as he/she wants around the boat and interact with some objects as well as examine some other ones. While performing any action with an object, the player will lose control over the character itself and will start to control the interacted object. For example, when picking up a letter, the player shall only rotate it or read it, but will not be able to move the character nor the camera. There are some objects like the book or the mobile phone that have controls that are a bit more complex, like turning pages or navigate through the menus of the mobile phone, but he player will still not be able of moving the character nor the camera.

4.2.2 Mechanics

Movement in the game

The movement in the game is pretty basic, since the main character is an old man and he can not run or bend. However, there is no need of special actions regarding the movement in this game, it is enough to walk so the game can be completed. There is the option to walk slowly, but its only goal is that the character can get closer to the seagull without scaring it. Also, if the player want to, he/she can zoom in to closely examine the objects.

Another reason why there is no need of more movement actions is because the game takes place on a boat that is quite small, so the implementation of more movements would only become a waste of time.

Actions and interaction with objects

There are different objects scattered around the boat, some of them have interactions, some other ones can be examined and the rest are only part of the decoration.

There are two types of interactable objects: the ones that the player has to hold a key, and the ones where the player has to press a key. The ones where the player has to hold a key are objects that have some kind of core behavior in the game, like eating, sleeping or smoking, while the other ones are quite basic, like opening doors.

The examinable objects do not have special behaviors, the player can pick them up, rotate them and drop them, but some of them have some special meaning regarding the story.

Screenflow

There are five different types of screens that the player can watch:

- **Initial credits:** As its name suggests, this is an initial screen with the game's logo.

- **Main menu 1:** This is the first menu that the player can see if he/she has not played the game before. Here he/she can change the settings of the game, like the volume or the resolution, start a new game or exit the game.

- **Main menu 2:** This menu can only be seen if the player has a save file. Here he/she can perform the same actions as in the other main menu, with the exception that he/she can continue the game and that the menu has now a background with the boat.

- **Game scene:** This is the main scene and the one that the player will be seeing most of the time, since here is where all the game takes place.

- **Final credits:** This screen is quite simple, with only containing the name of the participants in the project.

4.2.3   Saving

The saving in this game is performed when the character goes to sleep so the player can continue at the start of each day. The saved data contains:

- Character status

- Gained confidence with the seagull

- The flowers that have grown

- The remaining cigarettes

All this data is saved in a binary file that is loaded later to update the elements in the game.

4.3   Story, Setting and Character

4.3.1   Story and Narrative

The game starts with an old man that goes on a boat trip with no destination with his parents' old boat. When he decides to leave he writes a farewell note to all the people that he knows. During his journey he will recall all his memories through some letters, an old mobile phone and a book.

The most important memories start when he was a child and he went with his parents on the same boat. These memories will also cover some happy moments in his life, like when he met his first love or when he started to work, and some other moments that were tough for him, like when he lost his job or when his son and him started to be out of touch. In short, he will relive all the moments and decisions that have led him to the current situation.

4.3.2   Game World

The game world consists of the sky, the sea and the boat. All these elements have solid colors and light tones so the world can feel completely calm even at night. There are not many objects either, only the necessary ones, so the player can see a clear space where he/she can move.

The world around the player always moves calmly and does not suffer any important change, so it is a cosy place where he/she can feel comfortable and where there are not any kind of danger.

4.3.3   Characters

- Oscar,the main character

- Sofia, the person who sleeps with the protagonist

- Camila, the main character's wife

- Friends

- Son

- Father

- Mother

4.4   Interface

4.4.1   Visual System

The game is played in a first person point of view so the player can get more immersion. To enhance this immersion, there is no HUD in the game, with the exception of a little

point in the center of the screen and a sprite that appears in the same position when looking at some interactable object. There is also a black screen that fades in when the character goes to sleep and that fades out when he wakes up.

Also, when the player picks up an object, there is some blur in the background to simulate the depth of field and to focus the vision in the object. If the picked up object contains some text, the player can read it by displaying the text over the screen with a bit dark background. Finally, there is also a pause menu where the player can change some settings or return to the main menu.

4.4.2   Control System

The control system follows the same scheme as any other first person game: the character is moved with W, A, S and D keys and the camera is controlled with the mouse. The player can also move slower by pressing the left shift key.

Apart from this controls, the player can pick up objects with the left mouse button and rotate them with the central mouse button. If the picked up object can be read or turned on, the player has to press the F key to do it. There are two special objects: the rudder and the book. The player can interact with them after picking them up by pressing the right and left arrow keys. Another special object is the mobile phone, which after turning it on, can be used a real mobile phone by using the mouse and the left mouse button.

Regarding the interactable objects, these can be interacted with the E key, that has to be pressed or hold depending on the object. The player, after interacting with a cigarette, he/she can smoke by pressing the space key.

The player can activate the pause menu by pressing the escape key.

4.4.3   Audio, music, sound effects

All the objects that have some kind of interaction contain their own sound. For example, when opening a door or the chest, a squeak sound is played. This is done to give the player some sound feedback in addition to the obvious visual feedback.

There are also some ambient sounds, like screeching of the seagull or the sea that contribute to offer a more live environment.

The character also has its own sounds, like the footsteps sound, a coughing sound when he needs to drink, a rumbling sound when he needs to eat or a yawn when he needs to sleep. These are the most important sounds, since they indicate the player that he/she needs to take care of the character.

4.5   Artificial Intelligence

This game only counts with one AI since there is no need for more, because the game happens in a small space where the character is alone and there are no enemies, but it has been implemented to enhance the feel of an active world around the character.

This AI is a seagull that accompanies the character during almost all the game and that is often flying around the boat. Its flying routes are pre-determined so the player can expect which one is it following. This way, its easier for him/her to gain confidence with the seagull and acquire the capacity of approaching it without problems.

To increase the confidence with the seagull, the player has to approach it each day that passes by in the game so the next day he/she can get closer to it.

4.6   Game Art

Since the game looks for a simple visual style, the art style of the game is a 3D low poly one. It is an art that represents the objects with a few polygons and plain colors, obtaining 3D models that are not realistic.

The 2D art also has simple forms and plain colors so it fits with the 3D models in the game.

4.7   Additional information

4.7.1   Key words

Video game, First Person Adventure, Game Feel, Visual narrative, AI

4.7.2   Related subjects

- Video game art (VJ1223)

- Algorithms and data structure(VJ1215)

- Artificial Intelligence (VJ1231)

- Game engines (VJ1227)

- Hypermedia narrative and video game analysis (VJ1218)

WORK DEVELOPMENT AND RESULTS

Contents

Now, it is going to be explained how the work has been done. Full control when it comes to making scripts and managing the scenes in *Unity* has been had, so there has not been any problems about the coordination with the 3D designer.

To create some of the mechanics and to solve some trivial problems, the Unity manual [16] and some forums like Stack Overflow [5] and the official Unity forums [15] have been really important, since most of the possible problems that have been found, had already been solved in this websites.

5.1 Work Development

5.1.1 Technical Development

This is the area of work that is going to be larger than any other one. Next it is going to be explained every aspect that has been worked on.

1. **Player Mechanics:**

   - **Movement, camera and interactions:** The basics of first-person control are concentrated in the free asset used as a starting point of the project, which are:

     – A first person controller script to handle every movement that the character does, like rotating and walking.

– A camera controller script that, as the name suggests, handles where the camera looks.

– An interaction controller script to manage the inputs to pick up objects or to interact with them.

Since the controls offered by the interaction controller needed to be modified so the character and the camera could not be moved while an object was picked up because it was necessary to keep every object at its initial position. To do this, the first person controller script and the camera controller script are disabled, but obviously the interaction controller script is not, given that this is the one that manages the activation of the other ones and the action of dropping the picked up object. It was also important to previously store the orientation and the position of the selected object relative to the boat before picking it up to restore them when dropping it.

Also, for the interaction with the objects the loading bar above the selected object is changed for a circle that fills up and an e key sprite in the center of the canvas. (see Figure 5.1)

Figure 5.1: E key sprite at interactions

After this settings were made, a problem related to how the raycast was

colliding with the objects was detected. It was only searching for objects in the interactable or the pickable layer, ignoring others and allowing the player to interact with them through the wall. That is why this had to be adjusted to make the raycast stop ignoring the walls.

- **Reading:** On the one hand, there are some letters or notes that can be read by pressing a key when they are picked up to show a GUI box containing the text associated to that object. It is a pretty simple system, where the object has an assigned text extracted from a *.txt* file in a script in charge of showing or hiding the text when a key is pressed. This script is enabled when the object is picked up and disabled when the said object is dropped, otherwise the text box may appear when it should not.

On the other hand, there is the book(see Figure 5.2). This system is much more complex and took much more time to develop than the letters one took, given that it was wanted to represent a book that physically turned pages and that had different textures for each page, which is a bit complex. To do this, starting from the 3D model of a book made by the 3D designer, two canvas that contained an image, rendered in world space, were assigned to the three first pages: one to the front face and the other one to the rear face. Next, the three corresponding pivot points around which the pages would rotate were set and, in the script, six different positions where the pages could be when turning one of them were also set, so no texture could be seen changing in real time and to give a sense of depth. All textures are mostly stored in advance as well as texts, but methods to dynamically add more were also implemented. Finally, a list of texts that were not physically connected with the book were added, it only showed two texts in two *TextMesh Pro*, one for each page, that depended on how many pages the player had turned.
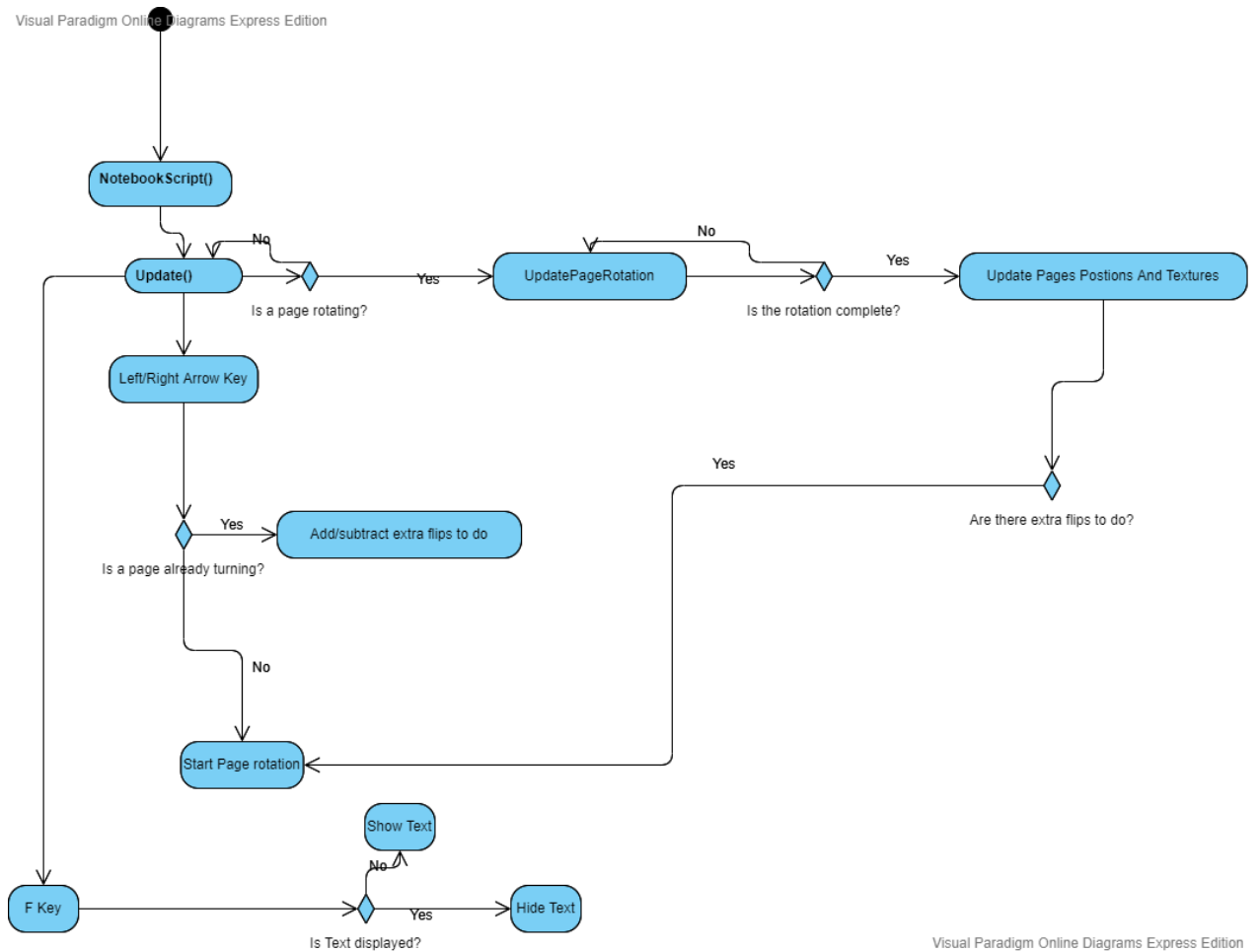
Figure 5.2: Activity Diagram of the book

- **Boat driving:** This mechanic is implemented in a pretty simple way. Given that the boat and its rudder had to rotate at the same time, it was implemented. When the player holds down the left or right arrow key, a float variable starts to slowly increase or decrease depending on the arrow key pressed and then rotation is applied to both elements by calculating using this variable. Also, if the objects are already rotating and the boat rudder is released, it does not stop rotating immediately. Instead, a coroutine is started to smoothly stop the rotation.

2. **Environment:**

   - **Day/night cycle and stars:** This part took a fairly long period of time given that, even if *Unity* has a pretty easy way to establish the Sun rotation

and its directional light(see Figure 5.3), no way of adding a moon was find, at least to add it as easy as the Sun. But, after checking some tutorials and forums, it was decided to improvise a moon made with a default sphere, a low poly texture and a material with some white emission that rotated opposite the Sun and around the ship, with its own directional light also opposite to the Sun one (see Figure 5.4). To add one last detail, some stars were also added by using a hemisphere-shaped particle system that emitted particles that last an entire cycle but are only visible at night thanks to the interpolation of their colors. [18]

Figure 5.3: Day

Figure 5.4: Night

- **Cloud generator:** After doing the day/night cycle it was thought that the sky was still too clear, that is why it was decided to create a cloud generator. This generator has two parts, an area to randomly spawn some clouds at the start, and an invisible wall that spawns them with a time interval. After this wall generates a cloud it keeps moving forward until it collides with a second wall placed on the opposite side, destroying it (see Figure 5.5).
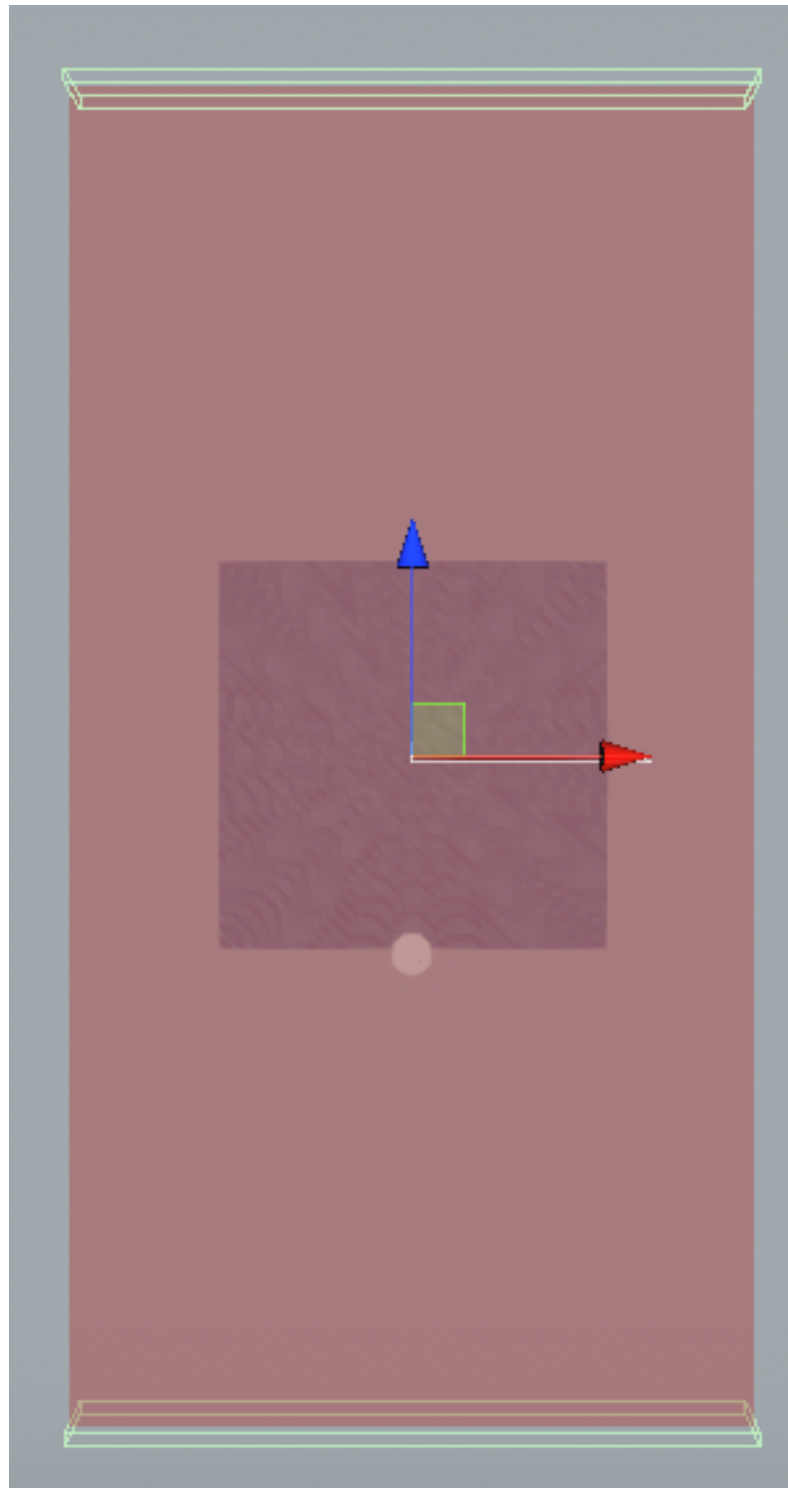
Figure 5.5: Clouds area and start and end walls

- **Objects:** While the 3D designer was the one who modeled and animated the objects, their animations in *Unity* were implemented with animators and the interaction scripts from the free asset of the first person controller were adapted so when interacting with some objects an animation was played. The workflow is that, when interacting with an object, a script that acts as an intermediary checks if the selected object has any animation to be played and if that is the case, the corresponding script is called.

3. **Seagull AI:** At first, an AI for the seagull was wanted to be created by using a navMesh that automatically navigated between preset points, but this idea had to be discarded given that it was a bit too complicated to use this to make an AI that could fly and above all, because this tool had to be previously baked, but since the ship could rotate this was not a good solution.

   Next, it arose the idea of implementing the A star algorithm, but this idea was quickly discarded, given that it could have taken a long time in addition to possibly having had to take into account many trivial cases that would have given problems.

   Finally, it was decided to create all the possible paths that the seagull could take and randomly travel between them. Even if this solution was not the best, there was not too much time left, since some other key functionalities also needed to be implemented.

   Now, the general behaviour that the seagull has consists of two areas around it with different sizes (see Figure 5.6), and when the player enters the big area and is walking too fast, the seagull picks a random route from the point where it is and starts to move. The same happens when the player enters the small area, regardless of player speed. When the seagull lands, if the player is still close, it immediately picks another route, and, if not, it waits for an entire minute playing idle animations until it starts to move again. As days pass by, the confidence with seagull increases so the character can get closer to it, but if he tries to slowly approach each day, the gained confidence is greater(see Figure 5.7). For now, this AI only works as a travel partner since there was no more time to work on it, but it was also wanted to add some interactions like feeding the seagull or petting it when the confidence level was at maximum, although it will be tried to add them in the future.
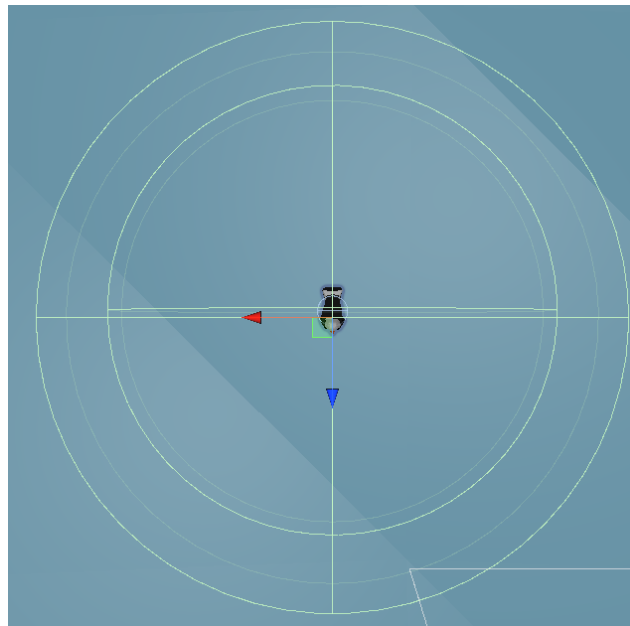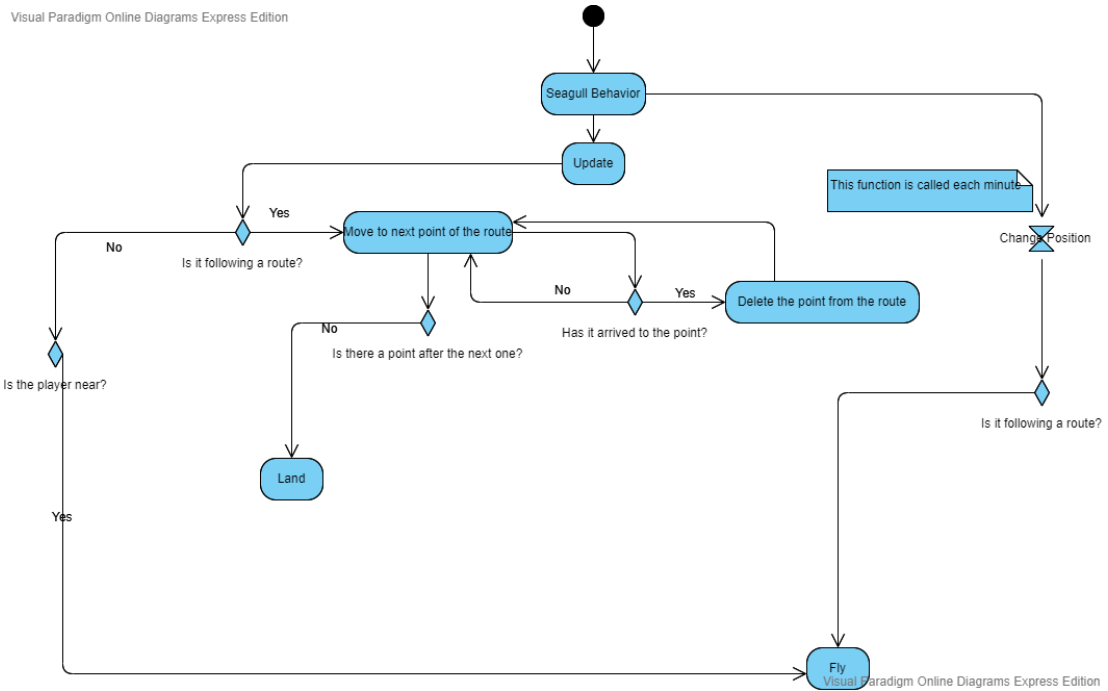
Figure 5.6: Seagull areas

Figure 5.7: Activity Diagram of the seagull AI

4. **Save system:** Initially, since the game was not going to be a too long one, no need of implementing a save system was found, but while it was being worked on it, it was noticed that it could take a considerable time to complete the game. That is when it was decided to implement it with a Brackeys tutorial [6], so the player could leave the game at any time.

Its basic behavior consists of, when the player performs the action of sleeping, a binary file stores all the key data of the game, like the confidence level with the seagull, the plants that had grown and the character status(see Figure 5.8). All this information is inside a custom class dedicated to save this data. The binary files were chosen since they were unalterable by the player, unlike the XML and .json files and also were easier to use to store many data, unlike the PlayerPrefs of *Unity*.

Now, if the player closes the game and opens it again, he/she can start a new game by deleting the binary file or load the previous data. When loading the previous data, everything in the game is updated to restore the game status as it was before closing the game.
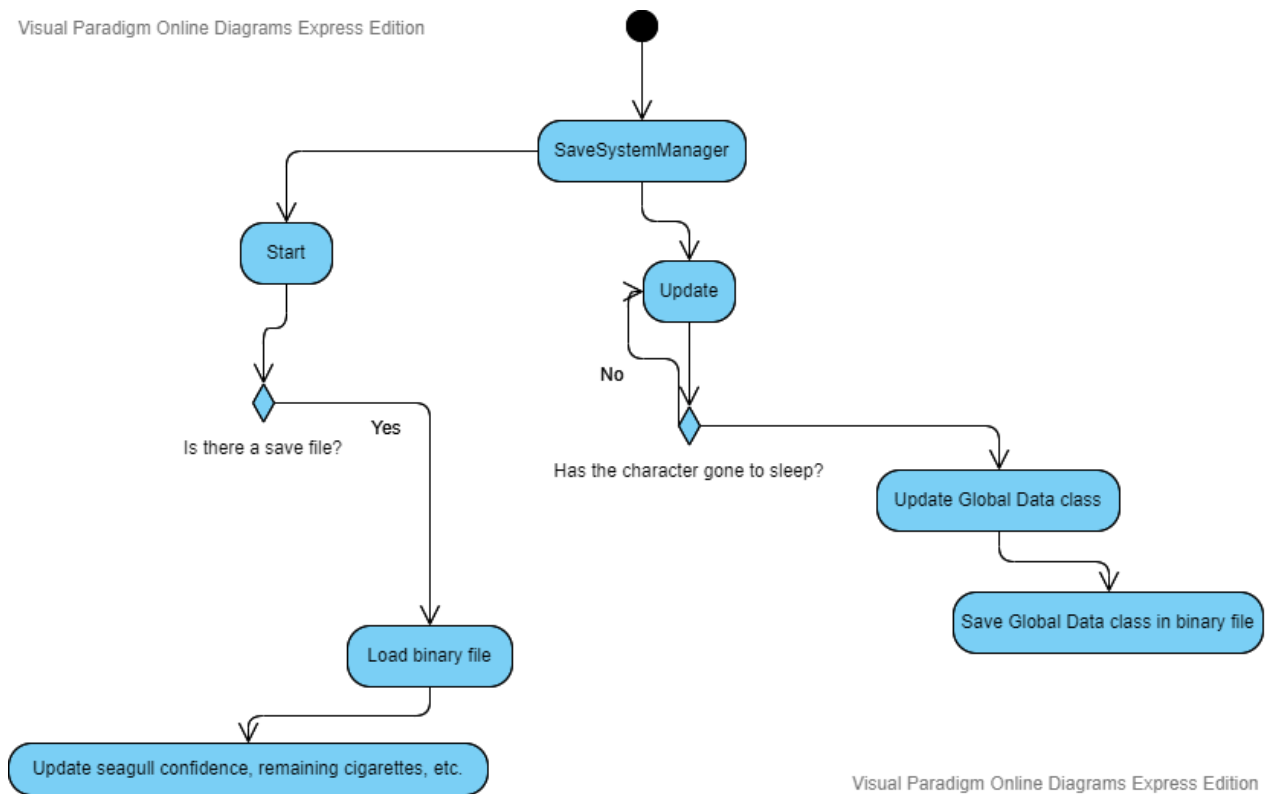

Figure 5.8: Activity Diagram of the save system

5.1.2   Artistic Development

This development was mainly leaded by the 3D designer since he has been the main artist of the game, but I also contributed with some tasks involving 2D art explained below.

1. **Logo:** This is the first design that was made given that the rest of the 2D art had to fit with the logo style. After a few attempts where it was tried to use gradients and a few colors and forms, it was decided to create a more simple logo based in a notebook. Next, with Adobe Illustrator, it was designed the game logo with lines with thickness variation and, with Adobe Photoshop, a notebook texture was added inside the letters(see Figure 5.9). After checking that this style fit pretty well, it was decided to create the rest of the assets also based in a simplistic style.
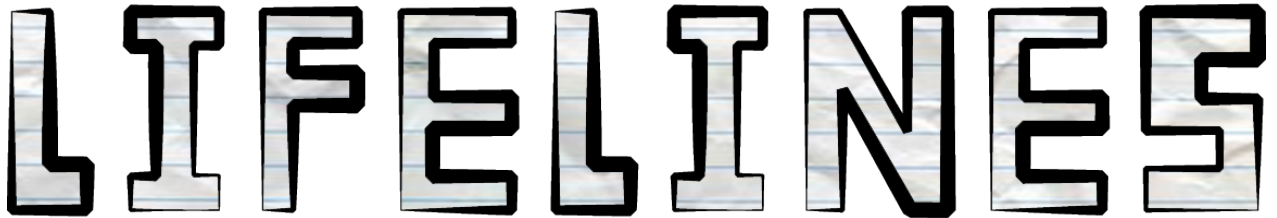
Figure 5.9: Game Logo

2. **UI assets:** After designing the logo, it was started to create assets for the settings menu. These assets had the shape of torn pieces of paper with adhesive tape on the border and had the texture of a notebook, just like the logo (see Figure 3.8).

3. **Menu Typography:** While the 3D designer created the in-game typography, the menus one was made. To do this, it was used a template from a website where it had to draw a character in the corresponding cell. After this, it was uploaded the *.png* to the website and generated a *.ttf* file to use as a typography. The letters design was pretty simple, with straight and bent borders combined (see Figure 3.7).

5.2   Results

In conclusion, I consider that every objective described in the first chapter has been accomplished, at least at a basic level and the game feels complete enough.

On the one hand, some things like the AI could have been better done, but mainly because of the available time for each task, it has not been possible to expand them. However it does not mean that they feel unpolished, since it has dedicated enough time to each task so that it is the best possible.

On the other hand, apart from completing all the assigned tasks, during the development it has also quickly implemented some functionalities like smoking that were not planned at first, but they added more immersion to the game.

About the deviations from the initial planning, the main ones are about the used time in each task. Even if most of them have taken the estimated time, the day/night cycle and the book ones have taken much more time due to similar problems. The main problem was that some assets that could do these tasks were searched to save up time but nothing came up. That is why these two functionalities had to be done from zero and took up much more time than expected, but the achieved quality was decent. Also, since the seagull AI was not as autonomous as expected, it required much more preparation than in the initial planning. All these problems made us extend the self-imposed deadline to finish the game from early April to mid-May. Apart from these three main deviations, all the original planning has been completed and have not been any more changes since everything has been possible to do.

In addition to these conclusions, I would like to also comment that by polishing some functionalities that I have developed like the book one or the day/night cycle could be published so some other game developers could use it in his/her game.

CHAPTER **6**

CONCLUSIONS AND FUTURE WORK

Contents

6.1   Conclusions

First of all, about complementing the development of this project and the labor practices, personally I have had no problem in this aspect, since I could follow a schedule that allowed me to work without stressing.

Next, focusing on the development of this project, I feel like I have lacked time. Even though we have made a complete game that can be played without problems and all objectives have been completed, I would have liked to make a bigger game, with more polished functionalities, AI and with even more elements, but it was not possible because of the available time.

Also, I have had no problem developing a joint project, since I have good coordination with my teammates and the tasks were clearly distributed so we could not ruin the other one's work. In this aspect, our biggest problem has been the difference of ideas because we had to develop every artistic and technical aspect of the game so we both were satisfied, which is something that we have accomplished.

In conclusion, I am pretty satisfied with the results and I think I have learnt much more by working on this project. But, as I have said, I would have liked to get results with a more professional touch, which was not possible so the game could have enough

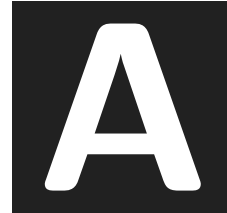more elements to make it a feel like a complete game and not a professional but empty one.

6.2   Future work

From the beginning we already decided that we were going to keep working on this project, since we consider that it can be easily expanded and polished, but we had no enough time. For example, we could tell a longer story and create more environments with more available actions or even use the current environment and add more secondary actions and some things like a weather system. Given that we have been focusing on finishing the project to deliver it, these expansions are still not clear, but our objective is to develop a complete and polished project that can be published.

BIBLIOGRAPHY

[1] Overleaf. `https://es.overleaf.com/`. Accessed: 2020-04-29.

[2] Adobe. Adobe illustrator. `https://www.adobe.com/es/products/illustrator.html`. Accessed: 2020-03-20.

[3] Adobe. Adobe photoshop. `https://www.adobe.com/es/products/photoshop.html`. Accessed: 2020-03-20.

[4] writers guild Alma. Fiction writer salary. `https://www.sindicatoalma.es/convenio-colectivo/%7D%7D`, 2015.

[5] Jeff Atwood. Stack overflow. `https://es.stackoverflow.com/`. Accessed: 2019-12-15.

[6] Brackeys. Save system. `https://www.youtube.com/watch?v=XOjd_qU2Ido`. Accessed: 2020-04-20.

[7] White Paper Games. The occupation. `https://www.occupation-game.com/`, 2019.

[8] Google. Google documents. `https://docs.google.com/`. Accessed: 2020-04-29.

[9] Indeed. 3d designer salary. `https://www.indeed.es/salaries/dise%C3%B1ador-3d-Salaries%7D%7D`, 2020.

[10] Indeed. Junior programmer salary. `https://www.indeed.es/salaries/programador-junior-Salaries%7D%7D`, 2020.

[11] Microsoft. Github. `https://github.com/`. Accessed: 2019-12-15.

[12] Microsoft. Visual studio. `https://visualstudio.microsoft.com/es/`. Accessed: 2019-12-15.

[13] Giant Sparrow. What remains of edith finch. `http://edithfinch.com/`. Accessed: 2020-03-20.

[14] Unity Technologies. Unity. https://unity.com/es. Accessed: 2019-12-15.

[15] Unity Technologies. Unity forum. `https://forum.unity.com/`. Accessed: 2019-12-15.

[16] Unity. Unity manual. `https://docs.unity3d.com/Manual/index.html`. Accessed: 2019-12-15.

[17] VeryHotShark. First person controller. `https://github.com/VeryHotShark/First-Person-Controller-VeryHotShark`. Accessed: 2019-12-15.

[18] Øyvind Strømsvik. Day/night cycle. `http://twiik.net/articles/simplest-possible-day-night-cycle-in-unity-5`. Accessed: 2020-01-25.

A

OTHER CONSIDERATIONS

A.1 Source Code

Since we have several scripts, instead of showing here some lines of code, we think it is better to offer the link to the github repository: `https://github.com/marklg5/TFG`

There is also an executable uploaded to Google Drive: `https://drive.google.com/file/d/1Apodh_Ewuqm1LmY6Qm_1PDhugsbhlszh/view?usp=sharing`