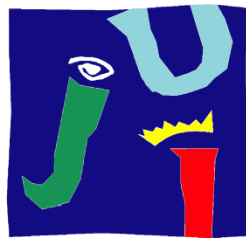


Creation of a narrative platform video game with decision making



UNIVERSITAT
JAUME·I

Degree in Video Game Design and Development

Author: Raúl Martínez Espinosa

Tutor: Marta Martín Núñez

Abstract:

This work consists in the creation of a platform video game where the player takes an active role in the narrative development. In that sense, the player can change the attitude of other characters, their dialogues and the story depending of the conversations that he/she has, the options that he/she selects during those conversations, the characters that he/she kills and the objects that he/she gets.

The game is divided in four levels and has some restricted areas which can only be accessed if we have accomplished certain requirements before.

The story takes place in a post-apocalyptic world where the protagonist, a man that wants to avenge the death of his son, has to find resources to survive, as well as face competitors who also want to obtain those resources. For that reason, the levels transmit destruction and loneliness.

Keywords:

Platforms, narrative, decisions, elections.

INDEX

- 1 Introduction → 7
 - 1.1 Objectives → 7
 - 1.2 Justification → 7
 - 1.3 Initial planning → 8
 - 1.4 Tools → 8
 - 1.5 Related subjects → 8
 - 1.6 Expected results → 9
 - 1.7 Technical specifications → 9
- 2 References → 10
 - 2.1 Narrative design references → 10
 - 2.2 Artistic design references → 11
- 3 Game design → 12
 - 3.1 Overview → 12
 - 3.1.1 Game name → 12
 - 3.1.2 Game concept → 12
 - 3.1.3 Genre → 13
 - 3.1.4 Target audience → 13
 - 3.1.5 Look and feel → 13
 - 3.2 Gameplay and mechanics → 14

3.2.1	Gameplay overview	→ 14
3.2.2	Game progression	→ 14
3.2.3	Mission challenge structure	→ 15
3.2.4	Objectives	→ 15
3.2.5	Movements and controls	→ 16
3.2.6	Physics and objects	→ 16
3.2.7	Screen flow	→ 16
3.2.8	Game options	→ 17
3.2.9	HUD	→ 19
4	Narrative design	→ 20
4.1	Storyline and transformation arc	→ 20
4.2	Beginning of the game	→ 20
4.3	Main characters	→ 22
4.4	Bosses	→ 25
4.5	Secondary characters	→ 27
4.6	Embedded narrative	→ 30
4.7	Dialogue system	→ 30
4.8	Decisions tree	→ 33
5	Artistic design	→ 35
6	Level design	→ 39

6.1 Levels sketches	→ 39
6.2 Harmful game objects	→ 41
6.3 Helpful game objects	→ 42
7 Programming	→ 44
7.1 Programming summary	→ 44
7.2 Programming details	→ 45
7.2.1 Player movement	→ 45
7.2.2 Enemy movement	→ 45
7.2.3 Dialogue system	→ 46
7.2.4 HUD	→ 48
7.2.5 Option menu	→ 48
7.2.6 Level change	→ 49
7.2.7 Camera follow	→ 50
7.2.8 Object interaction	→ 50
7.2.9 Data manager	→ 51
8 Results	→ 52
8.1 Hours dedicated approximately	→ 52
8.2 Project results	→ 52
8.3 Access to the code	→ 52
8.4 Access to all the project files	→ 53

9 Conclusions → 54

10 Future work → 55

Bibiliography → 56

1 INTRODUCTION

1.1 Objectives

- To design a game based in a narrative that makes the player feel that he/she is living in a post-apocalyptic world where he/she has to survive.
- To create paths and dialogues that allow the player to notice the importance of the decisions he has chosen. For example, presenting situations where other characters can die.
- To maintain a balance between gameplay and narrative by making the player's actions closely related to narrative changes.
- To develop an immersive narrative.

1.2 Justification

In this work we will develop the narrative, the modeling of the scenarios, the design of levels and the programming of a video game.

The narrative will be focused on the player knowing the events that have caused the destruction of the world, how the protagonist has arrived at the current situation, and the different choices that he can make to achieve his own benefit or help other characters to survive in that world.

Furthermore, the scenarios will have few lighting and we will use dark colors to create a solitude atmosphere, while the levels will be formed mainly by demolished buildings and closed areas that can only be accessed by meeting certain objectives.

On the other hand, is important to note that this project is focused on giving greater narrative complexity to a video game genre that traditionally has not allowed players to interfere in the story.

1.3 Initial planning

February: Design of the game concept, investigation and search of references. (25 hours) Creation of the tridimensional models. (50 hours)
March: Level design. (50 hours) Programming. (50 hours)
April: Programming. (50 hours)
May: Creation of the character dialogues and other narrative elements. (50 hours) Development of the project report and the presentation. (25 hours)

Figure 1 - Task and temporal planning

1.4 Tools

- Unity: A cross-platform game engine used to create different types of video games.
- Visual studio: An integrated development environment that serves to develop computer programs, websites, web apps, web services and mobile apps.
- Maya: A computer graphics application that can be utilized to create 3D models.
- Paint: A program useful to edit and create images.

1.5 Related subjects

VJ1216 - 3D Design[1].

VJ1218 - Hypermedia narrative and video game analysis[2].

VJ1227 - Video game engines[3].

1.6 Expected results

- An initial project with three levels, with an approximated duration of 30 minutes, that can be developed in greater depth in the future and transmits the feeling of being in a post-apocalyptic world.
- To achieve that the players replay the game to discover the different stories that can be developed.
- To experiment narratively in a genre that traditionally has not needed a deep and developed narrative.
- To involve the player with the story.

1.7 Technical specifications

The game has been developed in a PC with the following specifications:

- OS Windows 10.
- Nvidia GeForce GTX 960M.
- Intel i7 6700.
- 16 GB of RAM.

2 REFERENCES

2.1 Narrative design references

The dialogue system that we have used for the game is similar to the one applied in the *Fallout franchise*[4] (Interplay Entertainment, Black Isle Studios, Micro Forté, Bethesda Game Studios, Obsidian Entertainment, 1997 - 2019) to let the player see all the actual options he/she can select and the output of choosing that options.

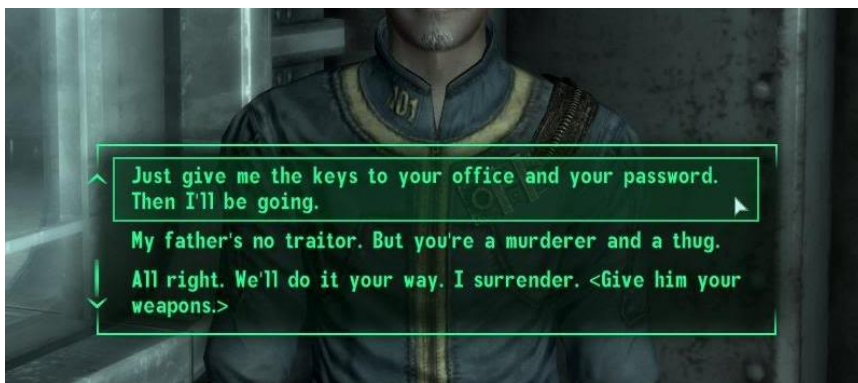


Figure 2 - Dialogue system of Fallout

Also, we have used the test dialogues of these games as reference to make the dialogue between White (the character that is controlled by the player) and Orange (one of the main characters of the game).

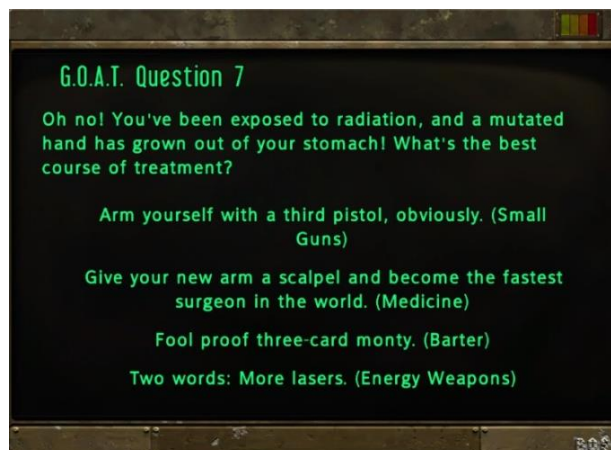


Figure 3 - G.O.A.T test of Fallout

On the other hand, the world, the characters and the interaction between them is based in stories that take place in post-apocalyptic worlds. In that sense, one of the stories that we have taken as main reference is the movie *The Road*[5] (John Hillcoat, 2009).



Figure 4 - Frame of the movie *The Road*

2.2 Artistic design references

The game has an illumination like the game *Inside* to create an atmosphere of solitude, sadness, mystery and danger.

Moreover, the camera position is similar to the one utilized in the game *Inside* to generate the feeling in the player that he is outside the action and he is an spectator of the things that are happening.



Figure 5 - Illumination and camera position of *Inside*

3 GAME DESIGN

3.1 OVERVIEW

3.1.1 Game name

The Dark Age.

We have decided to put this name to the game because it takes place in a period of time in which all civilizations have been destroyed and humanity has lost almost all its technology and advances.

3.1.2 Game concept

The player begins his/her adventure in a world almost destroyed by a war that has happened several years ago and has caused that most of the humanity and other living beings have disappeared.

The goal of the protagonist is to eliminate the people that have killed his son in a fight at the beginning of the game. Moreover, he has to survive collecting food, or other resources, which can be accessed by jumping obstacles and avoiding or killing enemies.

These objectives get more complicated as he advances through the levels meeting new characters, because they force him to make a series of decisions that modify the story, the relationships with other characters and the access to certain areas of the game.

Therefore, the player must decide through dialogues and his actions who to help and who he/she wants to face to get the few remaining resources in the world, know where the killers of his/her son are and have access to important areas.

In addition, his/her choices are essential to obtain information about what has happened before the war and what has led him to be in the current situation.

In that sense, it is important to say that this information serves to know more about the personality of the protagonist, the characters that he encounters during the game and the people that have killed his son.

With this information, the player can decide which characters he wants to help, if he has to continue with his revenge or forgive the killers of his son.

3.1.3 Genre

Platforms with decision making.

The game belongs to the platforms genre because the player needs to avoid a series of traps, enemies and obstacles jumping.

Furthermore, the game has decision making because the decisions of the player can change the game.

3.1.4 Target audience

The Dark Age is a product for people who like the platform genre, post-apocalyptic worlds and video games where they can choose how the story will be developed.

On the other hand, is important to say that the age of the target audience is greater than sixteen due to the topics that are treated and the violence.

3.1.5 Look and feel

The characters are cubes and the only way to recognize them is through their different colors and names. This way, the player has to talk with them to know how they are and their motivations.

On the other hand, the environment is formed by simple 3D models of furniture, mutated vegetation and destroyed buildings.

Also, it is important to say that the levels have little illumination, especially in closed areas, and the colors used are mainly dark.

3.2 GAMEPLAY AND MECHANICS

3.2.1 Gameplay overview

Although the game is made with 3D models, the player can only move through levels in two dimensions: vertical (up or down) or horizontal (left or right).

Moreover, the actions that he can perform are walking, jumping, shooting and interacting. It should be noted that the player can only interact when he/she gets close to an object, to use or take it, or when he/she is with another character, to begin a conversation with him, while the actions walk and jump are the main movements and can be done whenever the player wants and shoot only is available if he collects bullets before.

In that sense, the style of gameplay is similar to traditional 2D platform video games.

It is relevant to say that the reason to make the game 3D instead of 2D is aesthetic. We think that the game looks better in a 3D environment.

3.2.2 Game progression

The game uses a branching path storytelling system because the player can choose from different options during the game and that choices change the story and the areas that he can access.

In that sense, it is important to say that the game has four endings according to the player's decisions:

- The protagonist commits suicide because he/she does not want to live anymore: this happens if the player kills all of the bosses.
- Nothing happens at the end of the game: this ending appears if the player has killed at least one boss but not all of them.
- The protagonist joins a group that is dedicated to protect people from raiders: this end triggers if the player does not kill any boss.

- The protagonist finds a place that has not been destroyed by the war and lives there with other people: this situation occurs if the player does not kill any boss, helps Red and Red friendly counter is high.

In the case of the ending were nothing happens the game over screen appears after the game has finished, while in the other endings it is shown first the ending with a text over a black screen and later appears the game over screen.

In addition, to the different endings there are twelve ramifications in the story, two for each of the characters that the player can help. These ramifications act as sub stories and affect the areas he/she can access, the dialogues with the main characters, the attitude of other characters towards the protagonist and the information he/she receives. Also, some ramifications allow the player to spare the bosses.

On the other hand, it is relevant to note that there are four levels, one for each of the people that were in the fight where the protagonist's son got killed. At the end of each level, we can fight or forgive one of these murders, which have the role of bosses.

3.2.3 Mission / challenge structure

The mission of the player is to collect resources to fill his bar hunger, because if it reaches an empty state, the player dies and he begins from the start.

Also, he can do the missions that other characters tell him to obtain access to other levels and areas or learn more about the story.

In that sense, he has to read carefully what those characters say to give a proper answer because they can react in different ways. For example, if the player goes against their personal interests, they can attack him.

3.2.4 Objectives

The main objective is to face the people that have killed the son of the protagonist and decide whether to eliminate or forgive them.

Nevertheless, during the game the player can choose to accomplish other objectives like: save himself, help others to survive, learn more about the world where he lives, get access to restricted areas, kill everyone, etc.

3.2.5 Movements and controls

There are ten controls in the game: left movement (a key), right movement (d key), jump / scroll up (w key), scroll down (s key), interact (e key), eat (u key), heal (i key), left shoot (j key), right shoot (k key) and open options menu (q key).

The scroll up and scroll down actions can be used in the menus and during the conversations to select options.

3.2.6 Physics and objects

If the player collides with an enemy character or a trap, he receives damage but if he hits an object or a neutral / friendly character, a take / talk / open / go to / use option appears on screen.

Furthermore, it is relevant to highlight that characters hit by bullets receive damage.

3.2.7 Screen flow

There are only three screens: the main menu, where the player selects the difficulty of the game, the game, which is divided in different levels, and the game over screen, where the player can return to the main menu.

The camera seeks the player when he moves and it gets black when the player goes to another level or he opens the options menu. If he goes to another level, the name of that area appears over the black screen, as shown in Figure 6, while if he opens the option menu instead appears the main menu and resume options. (Figure 6)



Figure 6 - Level change screen of The Dark Age

3.2.8 Game options

In the main menu, the player can select the difficulty of the game: normal or hard. If he chooses hard, the hunger bar decreases faster, he needs more bullets to kill enemies and receives more damage from enemy collisions and bullets.

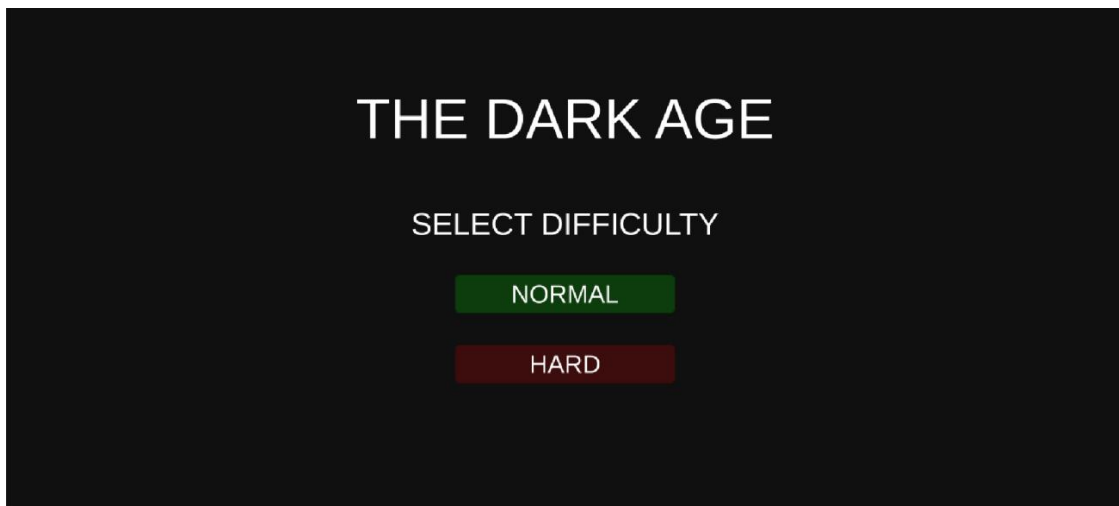


Figure 7 - Main menu of The Dark Age

Moreover, during the game he can pause the game opening the options menu. In this menu he can return to the main screen or resume the game.

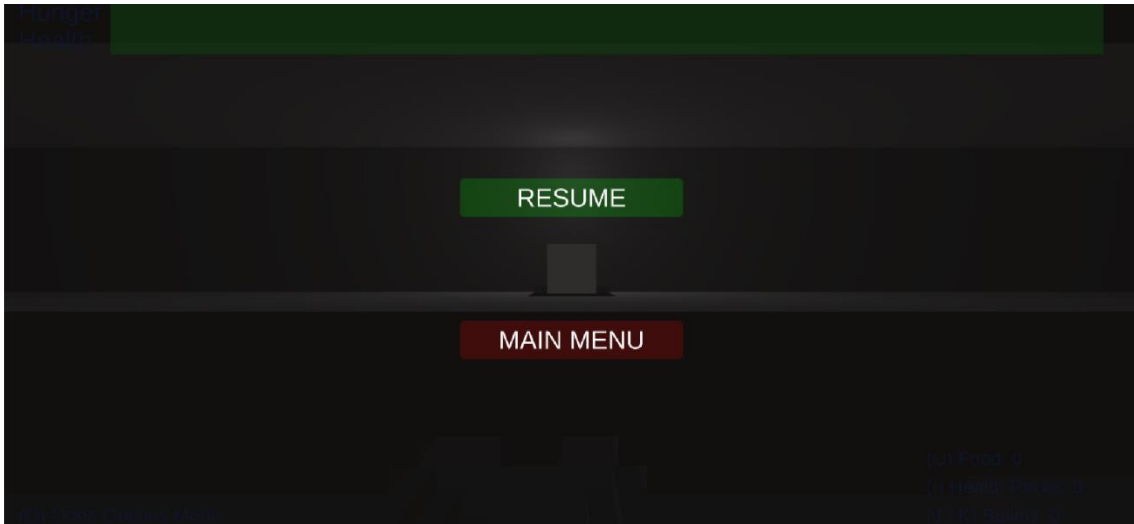


Figure 8 - Options menu of The Dark Age

On the other hand, the game over screen appears when the player dies or an ending is activated.



Figure 9 - Game over screen of The Dark Age

3.2.9 HUD

In addition to the hunger bar, during the game the player has a health bar and a counter of bullets, health packs and food.

The hunger and health bars are full and have a green color at the beginning. As he gets hungrier or loses health, they get reduced and change progressively to red. On the contrary, if he uses food or health packs, they are filled again and change to green.

On the other hand, the counters of bullets, health packs and food begin with a value of zero and their amount increases as he gets ammunition, health packs or food respectively. This way he/she can save items to use them later.

When the player shoots, heals himself or eats, the associated counter decreases.

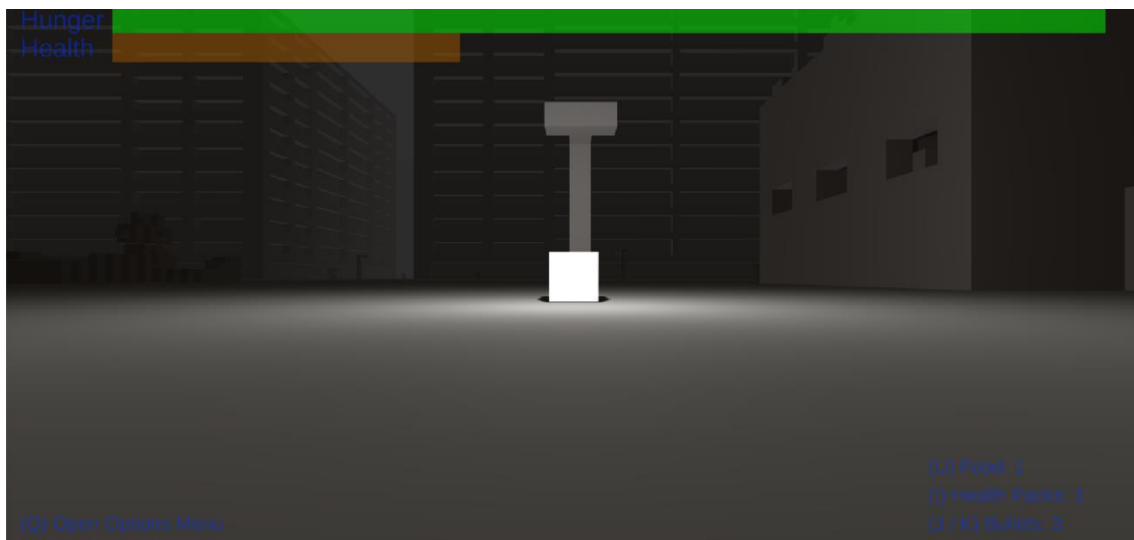


Figure 10 - HUD of The Dark Age

4 NARRATIVE DESIGN

4.1 Storyline and transformation arc

The story begins in the year 2050, twenty years after the Great War. At that moment the majority of the living beings have died, including the humans.

This reality is not strange for our protagonist as he has seen die all his family members and friends except his son, that gets killed in a fight for food at the beginning of the game.

When he saw him dying he loose all hope for a better future and his only mission in life is to survive long enough to get his revenge.

Depending of the actions that the player takes and the dialogues he has during the game he can change this objective and his opinion about the actual world.

In that sense, we can say that the player has full freedom to decide the future of the protagonist and the fate of the characters that he encounters.

4.2 Beginning of the game

When the game starts, a text over a black screen explains briefly the story to the player, as observed in Figure 11.

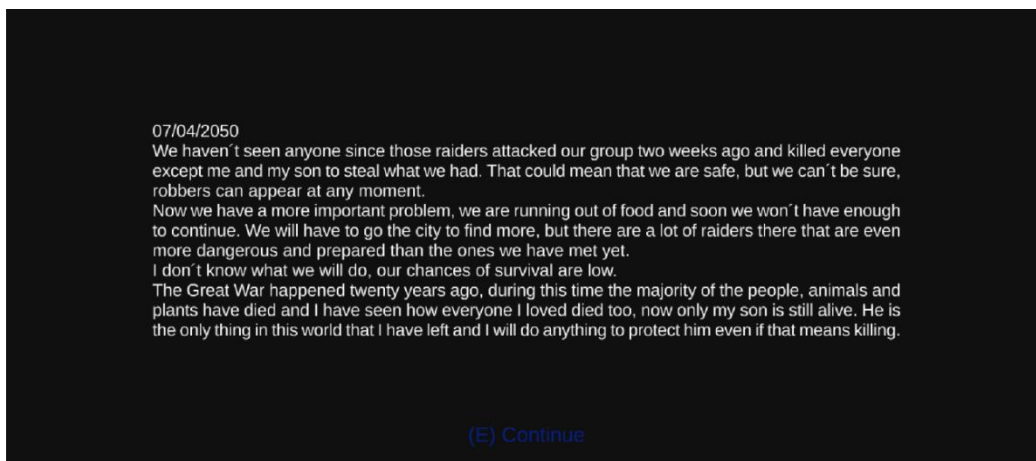


Figure 11 - Explanation of the story of The Dark Age

Later, a scene is shown where the player can see the encounter between White, White's son and the family of robbers that will be the main enemies during the game (Brown, Gray, Pink and Black). In this scene we see that White's son is killed by the family of robbers.

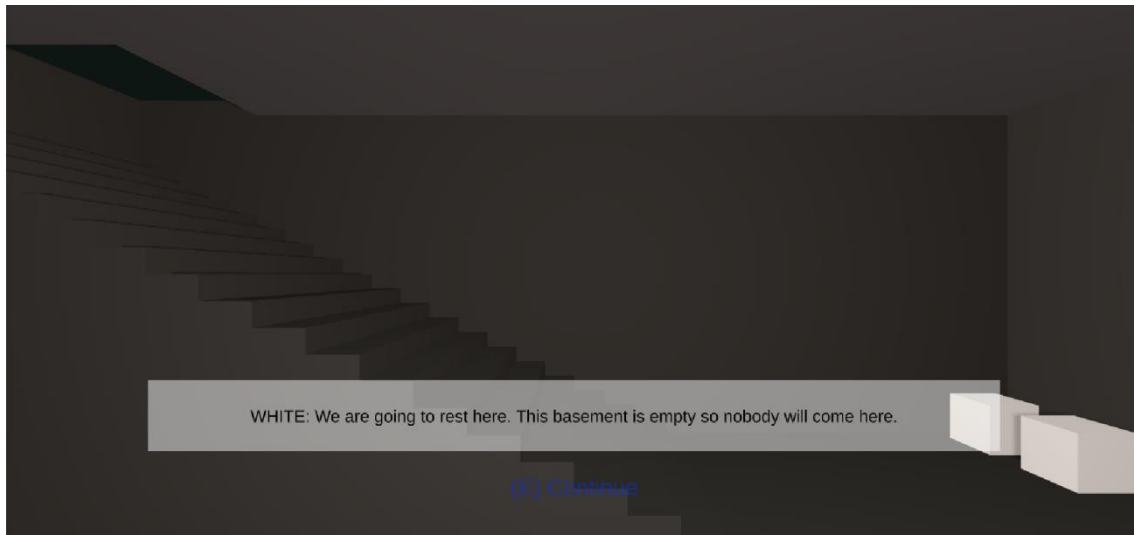


Figure 12 - (1) Initial scene of The Dark Age

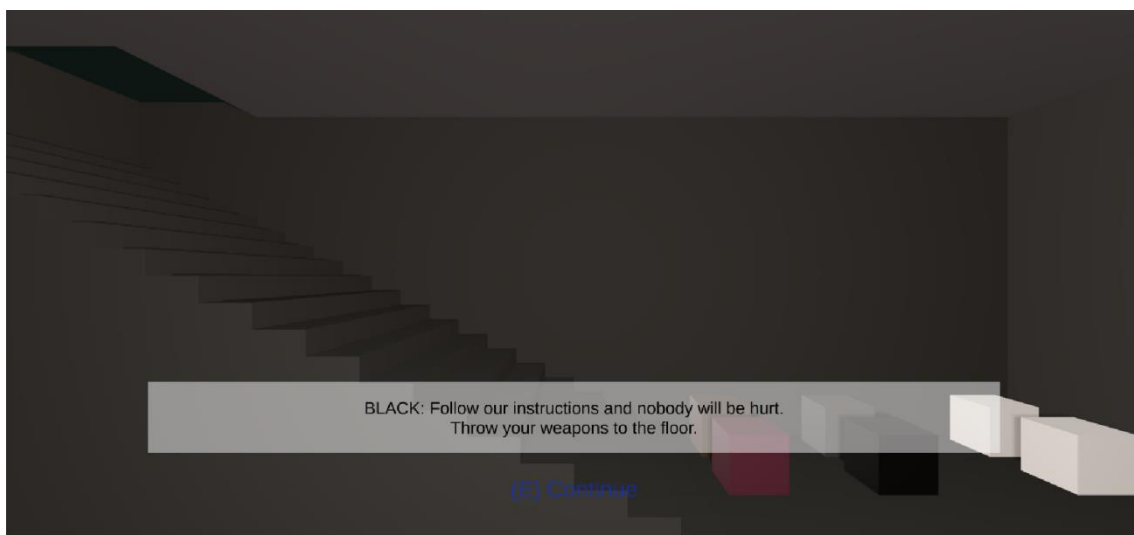


Figure 13 - (2) Initial scene of The Dark Age

It is important to say that the player can not perform any action during this scene (except going to the next part of the story pressing the e key) and the characters move in a predetermined way.

When this scene finishes, the player appears in the first level and a tutorial is shown to explain the player how to move and shoot, how to take and use items and their utility, the hunger and health system, who are the enemies, how to kill enemies, how to interact with other characters and how the dialogue system works.

4.3 Main characters

All main characters (except The Helper and White) have three types of dialogues: initial dialogue, friendly dialogue and hostile dialogue. The first one takes place when the player talks with the character for the first time, while the other two trigger depending on the decisions made during the previous dialogues with that character.

In that sense, it is important to say that all the main characters (except The Helper and White) have a friendly counter that increases or decreases with the options selected. If this counter is high in a character its friendly dialogue triggers, if is low its hostile dialogue appears and if is lower than 0, if the character is aggressive, he can attack the player and he will not be able to talk with him anymore.

On the other hand, it is relevant to say that before the player talks with a main character (except with The Helper) he needs to do a set of actions to unlock all the options. If he does not do some of them an advice appears on screen telling him that some options can not be selected.

Finally, it should be noted that after the player has done a dialogue with a main character it will not be longer available, so the decisions made in it can not be undone.

Main characters:

- The Helper: He helps the player during the game by telling him with texts the keys that he has to press to do actions and explains in the tutorial how to play the game.
- White: He is the protagonist of the story and the character controlled by the player. He was a cop before the Great War because he believed in justice and wanted to help other people. Nevertheless, when the world got destroyed, he lost this ideal

because everyone fought each other and his beloved ones began to die when he tried to help unknown people.

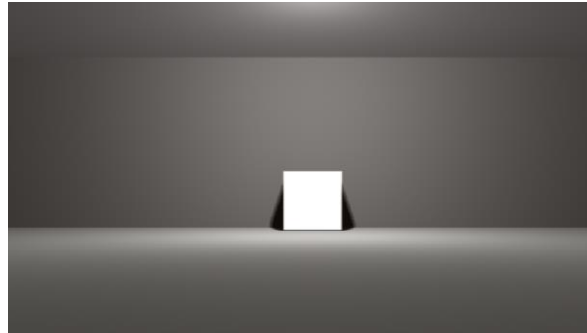


Figure 14 - White

- Blue: She is the ex-leader of a group of raiders. She was expelled from that band when she lost the support of his comrades after several months stealing very few resources. The player can help her to get information about the people that have killed his son and to get access to areas where there is more ammunition.

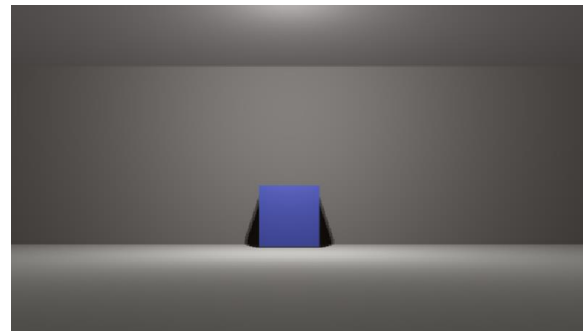


Figure 15 - Blue

- Red: He is a man that has lost his mind over the years after the Great War. He thinks that exists a place that had not been destroyed by the war and wants tools to fix his radio to listen to information about this place. If the player helps him he will be rewarded with a key to open an area where there are resources and information about what have happened before the war.

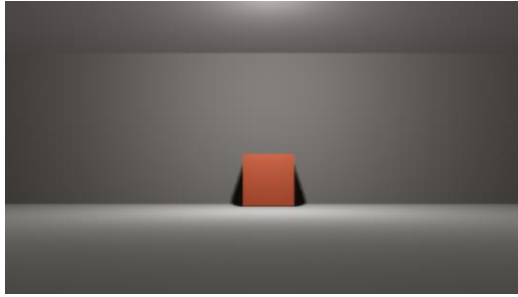


Figure 16 - Red

- Yellow: She is a doctor that cures people in a destroyed hospital with the last equipment that she has. She needs more health packs to save lives but a group of raiders is seeking that equipment to blackmail ill people. Helping her lets the player get access to conversations with her patients.

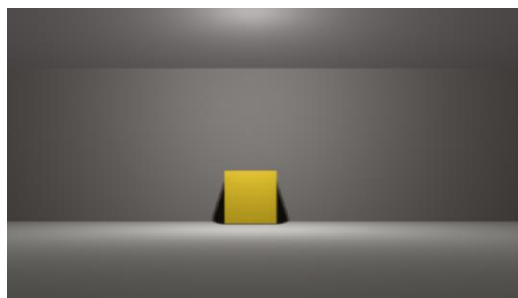


Figure 17 - Yellow

- Green: He is the leader of the band that wants the health packs. He can reward the player with ammunition for each health pack.

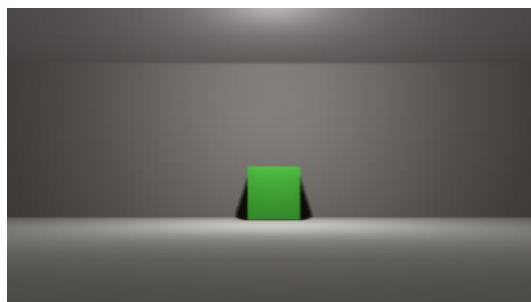


Figure 18 - Green

- Orange: He is a robot that was designed to protect the research facilities of a military laboratory. In that place, the player can increase the damage that his gun does, but needs to earn the trust of Orange before. To get that trust, the player has to do the missions that the robot gives him.

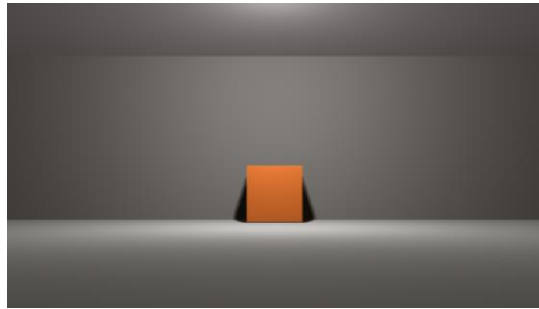


Figure 19 - Orange

- Purple: He is a child who hides from a group of raiders. The player can help him to look for his parents. Also, the player can discover that the mother of this child was killed by that group of raiders and his father is hiding until he can counterattack.

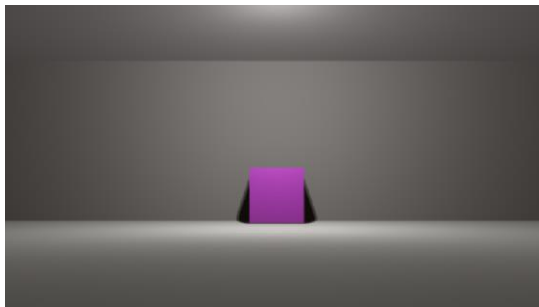


Figure 20 - Purple

4.4 Bosses

Each of the bosses have a dialogue that can be unlocked if the player has done some actions before. With this dialogue he can forgive them if he selects the proper options.

Also, it is relevant to say that these dialogues, like the dialogues of the main characters, can only be done once.

Bosses:

- Black: He is the person who kills White's son. He and his family (Brown, Gray and Pink) survive in this post-apocalyptic world stealing other people food, but they have never killed anyone until they encountered White and his son. He kills White's son during a fight for food. After the murder, he and his family disappear but White wants to find them to kill them.



Figure 21 - Black

- Brown: He is Black's son. He becomes mad after the death of White's son and thinks that his death is caused by demons because he does not want to believe that his family killed him. He can give resources to the player.

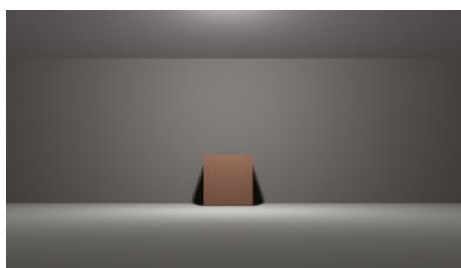


Figure 22 - Brown

- Gray: She is one of the two daughters of Black. Before the war she was a robber because she had a debt with the mafia and she could not pay it with the cash she made in her job. She teaches

all that she knows about robbery to the rest of her family. She can give resources to the player.

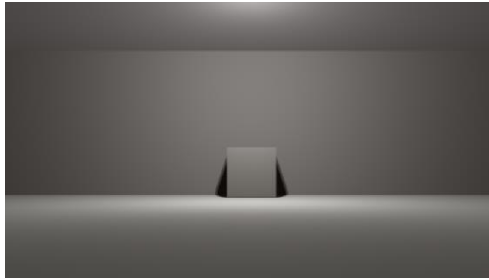


Figure 23 - Gray

- Pink: She is the other daughter of Black. Before the war she was an activist that tried to avoid the destruction of the world. The player can give all his resources to her.

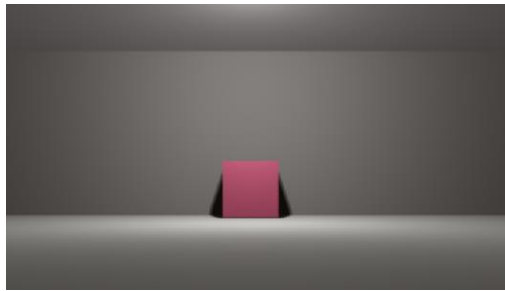


Figure 24 - Pink

4.5 Secondary characters

The secondary characters only have one dialogue and all their options are unlocked when the player talks with them.

When the player has done a dialogue with a secondary character it will not be longer available, so the decisions made in it can not be undone.

Secondary characters:

- Crimson: She is an elder woman that has seen a lot of things during his lifetime. These experiences serve her to give advices

to other people, to prevent them from making bad decisions. She can give a bullet to the player.

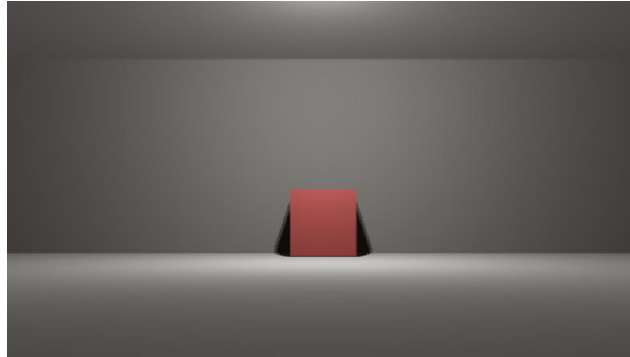


Figure 25 - Crimson

- Ochre: He is Yellow's brother. He has been with his sister since the war began and, like her, he wants to help other people and make decisions that benefit everyone. However, they sometimes argue about what is the best way to achieve these objectives. He can give a bullet to the player.

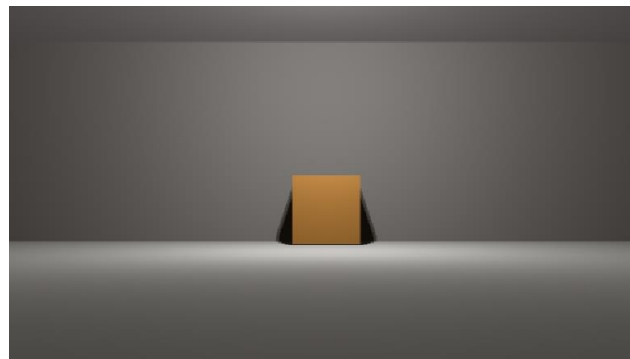


Figure 26 - Ochre

- Lime: The player can not know anything about her because when he meets her, she is sleeping.

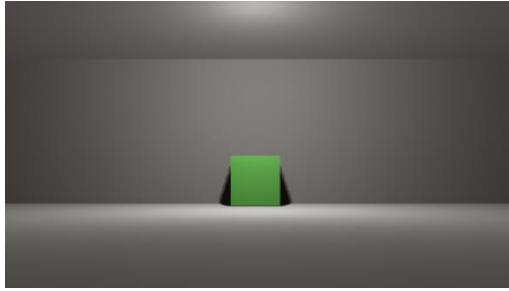


Figure 27 - Lime

- Azure: He wants to seize power inside the hospital and eliminate everyone that opposes him, for that reason he tries to expel from the hospital all the people that could be a menace. He has followers that see him as their leader and protector.

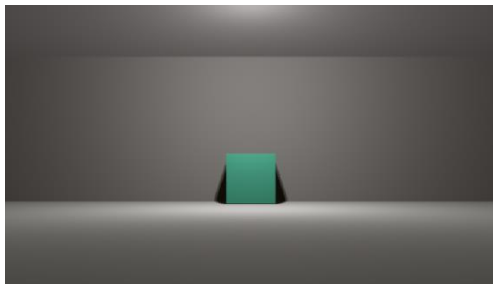


Figure 28 - Azure

- Obsidian: He is Purple's father. He wants to find his son to save him from the raiders who have attacked them. When he meets White, he thinks that he is one of those raiders.

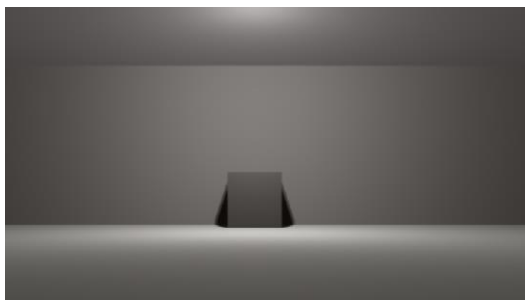


Figure 29 - Obsidian

4.6 Embedded narrative

Some items give information to the player about what have happened before the story of the game when he uses them. These objects, unlike the others, are not destroyed after being used.

In the game there are four of these items:

- Propagandistic letter announcing the beginning of the Great War.
- Propagandistic letter announcing that a plan has begun to defeat the Allied Forces.
- Propagandistic letter announcing that the missiles have been launched to the cities of the Allied Forces.
- Letter of a man that tells the consequences of the Great War.

The first three are written by the Resistance Union, the enemies of the Allied Forces during the Great War, while the last one is written by Cyan, a man who committed suicide after the Great War.

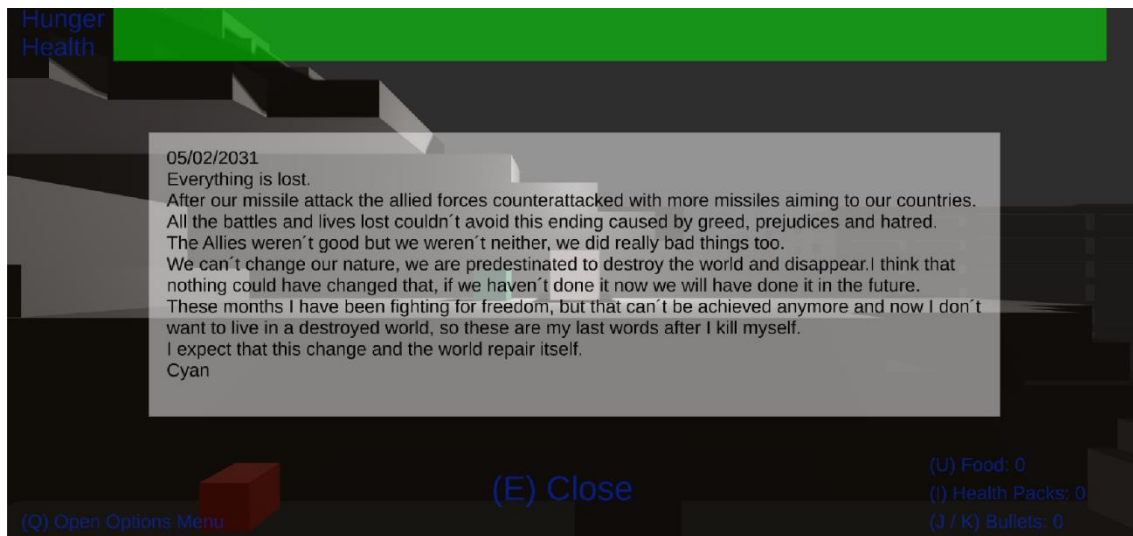


Figure 30 - Embedded narrative of The Dark Age

4.7 Dialogue system

The player can interact in the dialogues by choosing one of the three options available to him when he has to speak.

In that sense, is important to note that, regardless of the option selected, the next three options that he will have, in that conversation, are the same.

Nevertheless, that options are important because they change what the characters say, can let him enter in some areas, give him items, finish a conversation, make the character he is talking with attack him, etc. Those choices determine how the story and the events of the game will be developed.

Example of the dialogue system:

Green A:

Hey, you, wait there!

Where do you think you are going?

Don't move and give me everything you have.

White A:

- Option A1:

Ok, came here and take it.

(Attack Green)

Output: Green B1, Green becomes an enemy, ends dialogue

- Option A2:

You have met the wrong person.

I'm not going to give you anything.

The best you can do is run before I get angry.

Output: Green B2

- Option A3:

I only have this, take it.

Output:

Food > 2 and bullets > 2: Green B3.1, White loses all resources, blocks dialogues with Green, ends dialogue

Else: Green B3.2, Green becomes an enemy, ends dialogue

Green B1:

Nice, I'm going to enjoy this, I like that sheep fight before they die.

Green B2:

I like that, I like the people that doesn't bend, I like wolves.

Green B3.1:

Thanks.

I give you permission to leave.

Green B3.2:

You think that I'm joking, you think that I'm a fool?

This isn't enough.

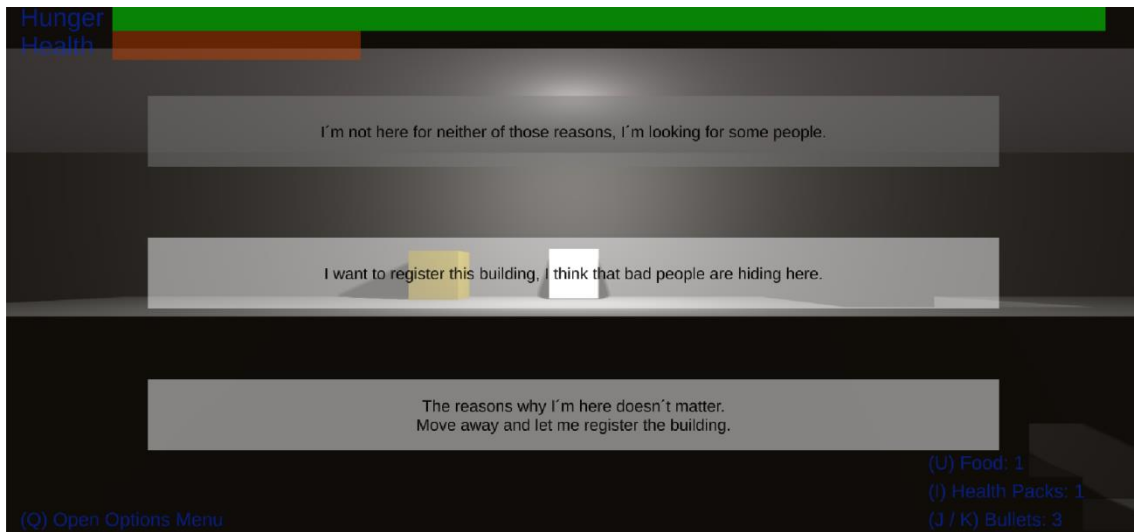


Figure 31 - Dialogue system of The Dark Age (player options)

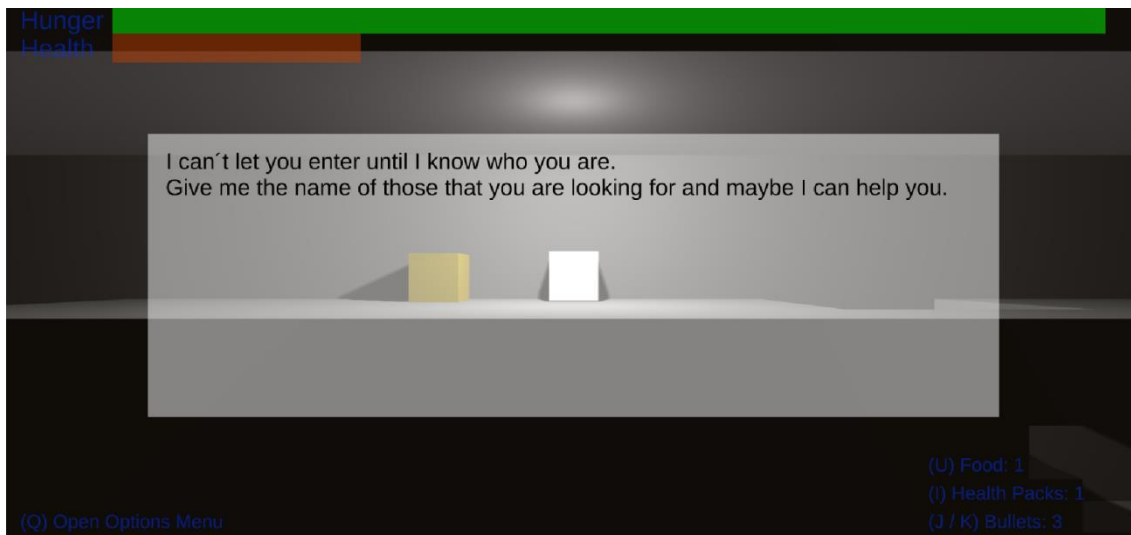


Figure 32 - Dialogue system of The Dark Age (character dialogue)

4.8 Decisions tree

To understand better the ramification system of the game there are Twine[6] schemes were it can be seen the options of the game and the consequences of selecting that options. Each of the schemes represents the decision tree of one character.

To do these diagrams we have to create a new passage (a new independent part of the story), insert in that passage the text that we want to show and include links to connect that passage to other passages. Optionally we can add variables to allow or deny the access to other passages if those variables have a determined value.

```

No, I haven't killed them yet. Give me the key and I will ↗ ✕
-----
+ Tag
-----
• Extortion doesn't work with me.
• I will be fine without your help.
•
• (if: $blueFriendlyCounter >= 15) [[Blue Friendly Dialogue 1 - A (Level 3)]]
• (if: $blueFriendlyCounter < 15) [[Blue Hostile Dialogue 1 - A (Level 3)]]
•
• (set: $blueFriendlyCounter to $blueFriendlyCounter - 10)
•
• |

```

Figure 33 - Example of Twine passage

In the Figure 33 we can see in the first two lines a text, in the third and fourth lines conditions that only allow the use of links if the variable `blueFriendlyCounter` has a determined value and in the last line a code that change the value of a variable. In the upper part is shown the name of the passage.

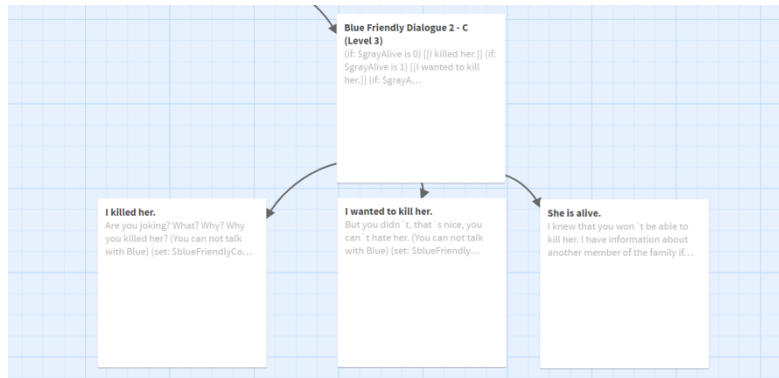


Figure 34 - Example of Twine scheme

These schemes can be played as interactive stories in the website philome.la

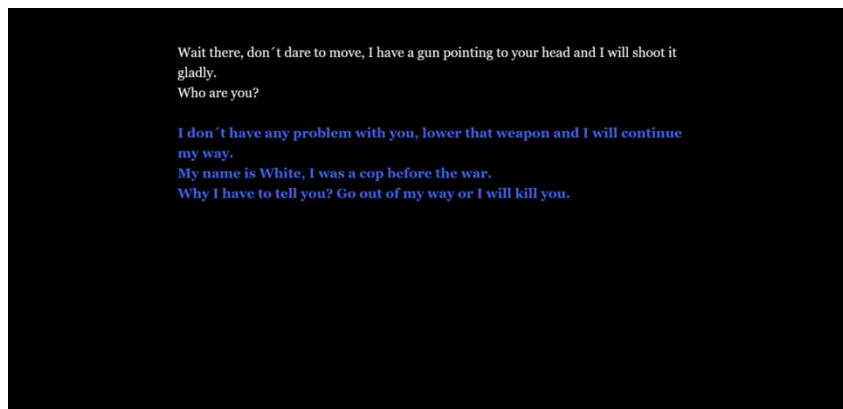


Figure 35 - Interactive story of the character Blue

Link to the Drive repository where there are the Twine files and the links to the philome.la interactive stories:

5 ARTISTIC DESIGN

The environment is formed by low poly models (3D models with a small number of polygons) with cubic shapes to match the style of the characters. Nevertheless, they have some details to do not make them monotonous.

In that sense, is relevant to note that we have made these models with the program Maya using the Extrude tool and the scale and position modifiers.

Moreover, is important to say that the predominant colors of the environment are white, gray and black. But other colors are also used to highlight important objects and differentiate characters.

On the other hand, the HUD, menus and dialogue system are formed by simplistic designs to maintain the style coherence and focus the attention of the player in the information they provide. To distinguish them from the environment they use green, red and blue colors but have low brightness to maintain the atmosphere of solitude, sadness, mystery and danger that the game wants to provide. We have made these elements using a canvas (an area where all the UI elements must be to work properly), the tool TextMeshPro (a gameobject that shows texts), images and buttons (interactive images).

Furthermore, to create the atmosphere of the game we need to change in Unity the illumination values (to darken all the scene), use a directional light, and rotate it (to illuminate the back faces of the objects instead of the parts of them that the player sees), and activate the modifier fog (to cover the backgrounds of the levels with a dense fog).

Finally, it is relevant to highlight, from the artistic design, that each level is different from the others:

- The first level takes place in the periphery of the city. There the player can see the full devastation of the Great War because the majority of the buildings are now ruins and the last that are still standing are almost destroyed. (See Figure 35)



Figure 36 - Level 1 of The Dark Age

- In the second level, the player is at the center of the city. There, the buildings are better preserved and some of them do not have visible damages, as shown in Figure 36.



Figure 37 - Level 2 of The Dark Age

- The third level takes place in a forest and is full of plants that have been mutated, as a consequence of the radiation of the missiles launched during the Great War. Also, in this level the player can go to cabins and other buildings.



Figure 38 - Level 3 of The Dark Age

- The fourth level has two differentiated zones: the research facility and the underground. In the research facility the player can see a big mountain in the background, a building and an elevator (Figure 38), while the underground is formed by four chambers where there are robot enemies, mobile platforms and switches. (Figure 38)

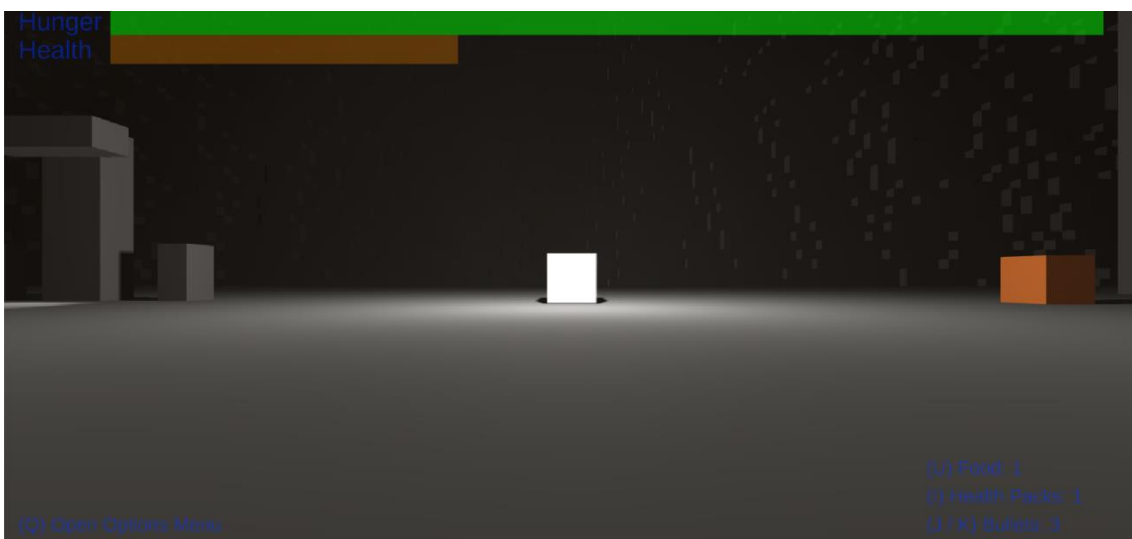


Figure 39 - Level 4 of The Dark Age (research facility)

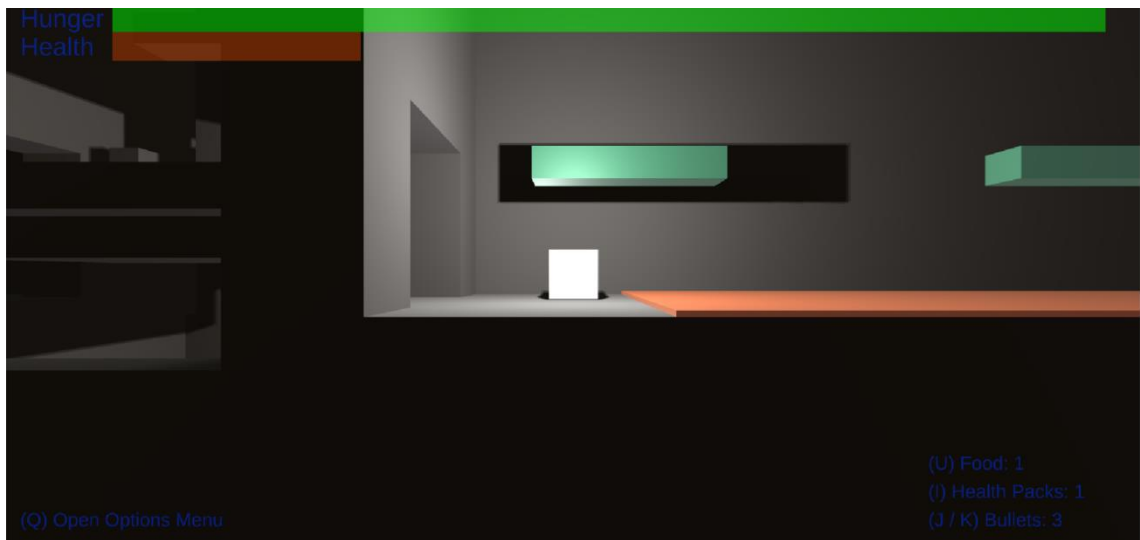


Figure 40 - Level 4 of The Dark Age (chambers)

6 LEVEL DESIGN

6.1 Levels sketches

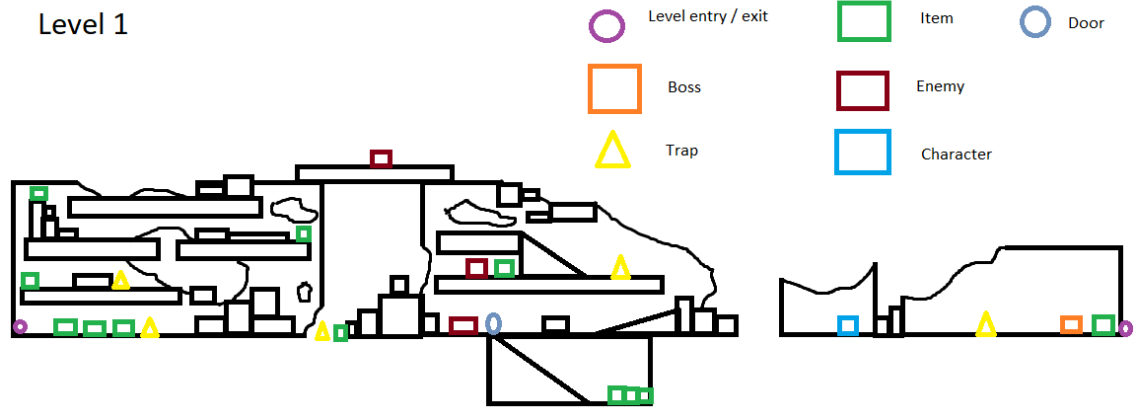


Figure 41 - Sketch of the level 1 of The Dark Age



Figure 42 - Sketch of the level 2 of The Dark Age

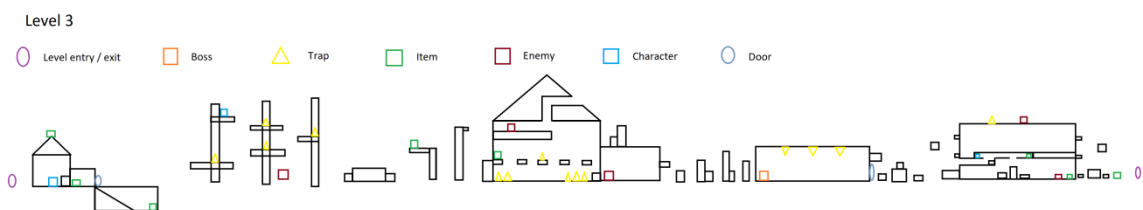


Figure 43 - Sketch of the level 3 of The Dark Age



Figure 44 - Sketch of the level 4 of The Dark Age

We have used these sketches as a reference to do the design of the levels of the game. However, they have some differences with the final design of the levels.

In them, we can see information about the position and number of different objects in each level. These objects are:

- Level entry / exit: They serve to go to other levels and know where the player will appear when he changes of level.
- Boss: Is the enemy that we have to beat or spare to get access to the next level.
- Trap: These objects can deal damage to the player. They can not be destroyed by bullets.
- Enemy: These objects can deal damage to the player. They can be destroyed by bullets.
- Item: When used they give information or resources to the player.
- Character: They are the people with whom the player can talk. Some of them can become enemies.
- Door: They block the access to some areas. The player has to do some missions, select the proper options or use switches to unlock them.
- Elevator: We can use them to go to some areas.
- Mobile platform: They are platforms that move.
- Upgrade station: We can use it to increase the damage of the bullets of the player.
- Switch: It serves to open doors.

In the next two sections we are going to explain these objects with greater depth.

6.2 Harmful game objects

All the harmful objects that the game has are traps and enemies. The main differences between these two types are that the traps are activated by triggers and can not be destroyed by bullets, while the enemies have collisions and can be destroyed by bullets. Although traps are immune to bullets, some of them are destroyed when they come in contact with the player.

Moreover, it is important to say that the gunner enemies can harm the player if they collide with him or if their bullets hit the player, while traps and contact enemies can only deal damage if they collide with the player.

Also, it is noteworthy that permanent traps and enemies deactivate their contact damage for a period of time when they hit the player, when that time passes, their contact damage is reactivated.

The types of traps that the game has are the following:

- Static trap: This trap can not move.
- Moving trap: This trap can move.
- Permanent trap: This trap is not destroyed when it comes in contact with the player.
- Destructible trap: This trap is destroyed when it comes in contact with the player.

It is relevant to keep in mind that these types can be combined. For example, we can have a static trap that is destructible, a static trap that is permanent, etc.

On the other hand, the types of enemies that the game has are:

- Static enemy: This enemy can not move.
- Moving enemy: This enemy seeks the player in the x axis when the player enters in his alert area.
- Contact enemy: This enemy only can deal damage with collisions.
- Gunner enemy: This enemy can deal damage with collisions and if their bullets hit the player. There are two types of this enemy: gunner enemy predetermined and gunner enemy positional. The gunner enemy predetermined shoots bullets to predetermined positions, while the gunner enemy positional aims his bullets,

when he shoots, to the place where the player is. The gunner enemy begins to shoot when the player enters in his alert area.

Once again, it is relevant to keep in mind that these types can be combined. For example, we can have a static enemy that is also a gunner enemy positional, a static enemy that deals damage only in contact, etc.

6.3 Helpful game objects

There are different types of objects that can help the player during the game:

- Doors: If the player manages to open them he can get access to areas where there are items or bosses.
- Keys: We can use them to open doors. To open a door with a key the player has to be colliding with that door, use the appropriate keyboard button and have the key that can open that door. The player can obtain keys through dialogues or can encounter them in the levels. The key items are destroyed after they are collected.
- Switch: They open doors from distance when the player uses them.
- Resources: These items give the player bullets, food and health packs, they are destroyed after they are used and they have a counter that determines the amount of each resource that they give to the player.
- Information object: When used, these items give information to the player.
- Elevator: In the actual state of the game the player can use this object only in the fourth level to access to the test chambers that are in the underground of that level.
- Moving platform: They can go in loop from a point to another or fall when the player touches them. If they are the falling types they return to their initial position when the player does not touch them.
- Character: As we have said before they are the most important narrative objects because the conversations that the player has with them determine how the story will be developed, the areas he can access, etc.

- Teleport (level entry / exit): The player can use them to go to other levels. In some cases they are not available until some requirements are completed.
- Upgrade station: When the player completes the missions that Orange gives him he can access to the research facilities, there he can improve the damage that his weapon does using the upgrade station. This upgrade can only be done once.

7 PROGRAMMING

7.1 PROGRAMMING SUMMARY

The game needs programming for the movement of the player and other characters, the dialogue system, the HUD, the menus, the change of levels, the movement of the camera, the scripted scenes, the interaction with objects and the saving of items and player actions.

The movement of the player is controlled by the keyboard and the physics associated to the game objects of the protagonist and the bullets he shoots.

The movement of other characters is managed by state machines and the physics of their bodies.

The dialogue system is composed of texts that display the words that the characters say and the options the player can choose, buttons to let the player select the choices he likes and a hierarchy of options to connect previous selections with future possibilities.

The HUD displays visually info on screen during the game that is updated according to external events that affect the player.

The menus are similar to the dialogue system because they display info with texts and the player can select options with buttons.

The change of levels happens when the player goes to the exit of a level and enters another.

The camera seeks the player when he moves.

During scripted scenes the movement of the player is disabled and the characters move in a predetermined way.

The interaction with objects is similar to the movement of the player because is handled with the keyboard.

Finally, the items are stored in variables to let the player use objects in the future while actions are stored in variables to trigger if-else conditions.

7.2 PROGRAMMING DETAILS

7.2.1 Player movement

To avoid that the player goes to a wrong z position, which would cause errors in the game, the position in the z axis is restored each frame.

The player can only move if he/she is touching the ground, this way the game is more challenging because is harder to move the character.

The player can move in the x axis and the y axis and his/her movement is controlled by the *CharacterController*[6][7]. (Figure 44)

```
1. void Movement()
2. {
3.     transform.position = new Vector3(transform.position.x, transform.position.y, 0.5f);
4.
5.
6.     if (characterController.isGrounded)
7.     {
8.         moveDirection = new Vector3(-Input.GetAxis("Horizontal"), 0, 0);
9.         moveDirection = transform.TransformDirection(moveDirection);
10.        moveDirection *= speed;
11.
12.
13.        if (Input.GetButton("Jump"))
14.            moveDirection.y = jumpSpeed;
15.
16.    }
17.
18.    moveDirection.y -= gravity * Time.deltaTime;
19.
20.    characterController.Move(moveDirection * speed * Time.deltaTime);
21.
22.
23. }
```

Figure 45 - Player movement

7.2.2 Enemy movement[8]

We use the variable *isAggressive* to block the movement and attacks of the enemy until some condition is reached. The main use of this variable is to make the characters harmless until an option that converts the character in enemy is selected.

The variable *alert* determines whether the player is in the alert zone of the enemy or not. If the player is in the alert zone, the enemy will go to his position in the x axis, if he is outside the alert zone, the enemy will return to his waiting position. The implementation is depicted in Figure 45.

```

1. private void FixedUpdate()
2. {
3.     if (isAggressive)
4.     {
5.         if (transform.position.x > thePlayer.transform.position.x + movementOffset && alert.playerInAlertZone)
6.         {
7.             waiting = false;
8.             enemyRigidbody.velocity = transform.TransformDirection(-Vector3.right * movementSpeed);
9.         }
10.
11.         else if (transform.position.x < thePlayer.transform.position.x -
12.             movementOffset && alert.playerInAlertZone)
13.         {
14.             waiting = false;
15.             enemyRigidbody.velocity = transform.TransformDirection(Vector3.right * movementSpeed);
16.         }
17.         else
18.         {
19.             if (!waiting)
20.             {
21.                 enemyRigidbody.velocity = transform.TransformDirection(Vector3.right * 0f);
22.
23.                 if (!alert.playerInAlertZone)
24.                 {
25.                     float step = movementSpeed * Time.deltaTime;
26.                     Vector3 targetPosition = new Vector3(waitingPosition.transform.position.x, transform.position.y, transform.position.z);
27.                     transform.position = Vector3.MoveTowards(transform.position, targetPosition, step);
28.
29.                     if (Vector3.Distance(transform.position, targetPosition) < 0.001f)
30.                     {
31.                         waiting = true;
32.                     }
33.                 }
34.             }
35.         }
36.     }

```

Figure 46 - Enemy movement

7.2.3 Dialogue system

Figure 46 presents the dialogue system. There, the variable *canTalk* serves to allow the player talk with the characters only if he is in the talking area of the character and the dialogue with the character has been unlocked.

The method *CheckIfTheOptionIsLocked* verifies if the option is locked. If the option is locked, when the player tries to select it nothing happens.

If the player selects a correct option the dialogue, he/she goes to the next character dialogue and player options. When there are no more options or character dialogues left, the dialogue ends and the player can move.

The variable *becomeAggressive* ends the dialogue and transforms the character into an enemy.

The variable *endgame* activates the ending, shows the game over menu and hides the game canvas.

The variable *die* destroys the character.

```
1. public virtual void Talk()
2. {
3.     if (Input.GetKeyDown(KeyCode.E) && canTalk && CheckIfTheOptionIsLocked())
4.     {
5.         if (playerTalking)
6.         {
7.             playerDialogues.SetActive(false);
8.             characterDialogueGameObject.SetActive(true);
9.             playerTalking = false;
10.        }
11.
12.        else
13.        {
14.            dialogueNumber++;
15.            if (dialogueNumber <= numberOfDialogues && !activatedEndDialogue)
16.            {
17.                playerDialogues.SetActive(true);
18.                characterDialogueGameObject.SetActive(false);
19.                playerTalking = true;
20.            }
21.
22.            else
23.            {
24.                talking = false;
25.                characterDialogueGameObject.SetActive(false);
26.                playerManager.canMove = true;
27.                canHaveThisConversation = false;
28.
29.                if (becomeAggressive)
30.                    enemy.isAggressive = true;
31.
32.                if (endGame)
33.                {
34.                    playerManager.canMove = false;
35.                    gameCanvas.SetActive(false);
36.                    gameOver.StartCoroutine("ShowGameOverScreen");
37.                }
38.
39.                if (die)
40.                    Destroy(transform.parent.gameObject);
41.            }
42.
43.        }
44.
45.        showOutput = true;
46.        canTalk = false;
47.        StartCoroutine("TimeToTalkAgain");
48.    }
49. }
50.
51.
52. public bool CheckIfTheOptionIsLocked()
53. {
54.     if(optionSelected == 1)
55.     {
56.         return playerDialogueText1.text != "(Locked Option)";
57.     }
58.
59.     else if (optionSelected == 2)
60.     {
61.         return playerDialogueText2.text != "(Locked Option)";
62.     }
63.
64.     return playerDialogueText3.text != "(Locked Option)";
65. }
```

Figure 47 - Dialogue system

7.2.4 HUD

The HUD class actualizes each frame the information about the resources of the player that are shown in the game canvas and changes the health bar and hunger bar according to the actual values of health and hunger of the player.

```
1. void Update()
2. {
3.     foodCounter.text = "(U) Food: " + playerManager.food;
4.     healthPacksCounter.text = "(I) Health Packs: " + playerManager.healthPacks;
5.     bulletCounter.text = "(J / K) Bullets: " + playerManager.bullets;
6.
7.     hungerBar.color = new Color32((byte)(255 -
8.     playerManager.hunger * 2.55), (byte)(0 + playerManager.hunger * 2.55), 0, 125);
9.     healthBar.color = new Color32((byte)(255 -
10.    playerManager.health * 2.55), (byte)(0 + playerManager.health * 2.55), 0, 125);
11.    hungerBarRectTransform.sizeDelta = new Vector2(playerManager.hunger * 7, 20);
12.    healthBarRectTransform.sizeDelta = new Vector2(playerManager.health * 7, 20);
13. }
```

Figure 48 - HUD

7.2.5 Option menu

When the option menu is activated the time of the game is stopped (*Time.timeScale = 0*) [9] and all the objects are paused.

If the player selects resume the game continues, and the option menu disappears, if the player selects return to main menu the application is restarted and the player goes to the main menu.

```
1. void Start()
2. {
3.     resumeButton.onClick.AddListener(Resume);
4.     mainMenuButton.onClick.AddListener(ReturnToMainMenu);
5. }
6.
7. void Resume()
8. {
9.     Time.timeScale = 1;
10.    gameObject.SetActive(false);
11. }
12.
13. void ReturnToMainMenu()
14. {
15.     Time.timeScale = 1;
16.     SceneManager.LoadScene(SceneManager.GetActiveScene().name);
17. }
```

Figure 49 - Option menu

7.2.6 Level change

The only way to go from a level to another is to use the teleport areas that are at the end and beginning of each level.

When the player teleports to another level the canvas changes to black and the name of the level is displayed in the canvas. After a few seconds the canvas disappears and the name of the game is eliminated from the canvas.

While the player is teleporting more teleports are not allowed until some seconds have passed to avoid errors.

If the player can teleport and he is in a teleport area he can use the e key to go to another level.

```
1. void Update()
2. {
3.     if (teleporting)
4.     {
5.         timeWaiting += Time.deltaTime;
6.         if (darken)
7.         {
8.             levelChange.color = new Color32(0, 0, 0, (byte)((timeWaiting * 255f) / 2));
9.             levelChangeText.color = new Color32(255, 255, 255, (byte)((timeWaiting * 255f) / 2));
10.        }
11.    }
12.    if (fade)
13.    {
14.        levelChange.color = new Color32(0, 0, 0, (byte)(255 - (timeWaiting * 255f) / 2));
15.        levelChangeText.color = new Color32(255, 255, 255, (byte)(255 -
16.        (timeWaiting * 255f) / 2));
17.    }
18. }
19. if (touched && playerManager.canTeleport && Input.GetKeyDown(KeyCode.E))
20. {
21.     switch (place)
22.     {
23.         case Place.CENTER:
24.             levelChangeText.text = "CITY CENTER";
25.             break;
26.
27.         case Place.PERIPHERY:
28.             levelChangeText.text = "CITY PERIPHERY";
29.             break;
30.
31.         case Place.FOREST:
32.             levelChangeText.text = "FOREST";
33.             break;
34.
35.         case Place.RESEARCH:
36.             levelChangeText.text = "RESEARCH FACILITIES";
37.             break;
38.     }
39. }
40. useObject.text = "";
41. teleporting = true;
42. playerManager.canTeleport = false;
43. darken = true;
44. StartCoroutine("NewLevel");
45. }
46. }
```

Figure 50 - Level change

7.2.7 Camera follow

During the game the camera goes to the place of the player but maintains his z axis position.

The "follow script" [10], in Figure 50, uses an offset and the method Lerp to go to where the player is in a smooth way. This serves to avoid blur effects.

```
1. private void FixedUpdate()
2. {
3.     Vector3 desiredPosition = target.position + offset;
4.     Vector3 smoothedPosition = Vector3.Lerp(transform.position, desiredPosition, smoothSpeed);
5.     transform.position = smoothedPosition;
6. }
```

Figure 51 - Camera follow

7.2.8 Object interaction

When an object collides with the player (object with the tag player) the variable touched is changed to true to say that the player is in contact with the item and he can use it.

When the player is not colliding with the object the *touched* variable turns to false to advice that the player can not use that item.

Depending on the type of object that he is colliding, the information text changes (*useObject*).

```
1. void OnTriggerEnter(Collider collision)
2. {
3.     if (collision.gameObject.tag == "Player")
4.     {
5.         touched = true;
6.
7.         if (gameObject.tag == "Resource")
8.             useObject.text = "(E) Take";
9.
10.        if (gameObject.tag == "Door")
11.            useObject.text = "(E) Open";
12.
13.        if (gameObject.tag == "Interactable")
14.            useObject.text = "(E) Use";
15.
16.        if (gameObject.tag == "RestrictedArea")
17.            useObject.text = "You can not access this area yet";
18.    }
19. }
20.
21. void OnTriggerExit(Collider collision)
22. {
23.     if (collision.gameObject.tag == "Player")
```

```

24.     {
25.         touched = false;
26.         useObject.text = "";
27.     }
28. }

```

Figure 52 - Object interaction

7.2.9 Data manager

The class *DialogueDataManager*[11] stores some variables that are accessed from other classes to determine if some options can be selected or not during the dialogues or to activate some of the dialogues.

```

1. [HideInInspector] public int blueFriendlyCounter = 15;
2. [HideInInspector] public int redFriendlyCounter = 0;
3. [HideInInspector] public int yellowFriendlyCounter = 0;
4. [HideInInspector] public int healthPacksGivenToYellow = 0;
5. [HideInInspector] public int greenFriendlyCounter = 10;
6. [HideInInspector] public int purpleFriendlyCounter = 0;
7. [HideInInspector] public int orangeFriendlyCounter = 0;
8. [HideInInspector] public int numberOfBossesDead = 0;
9. [HideInInspector] public int numberOfUpgradeStationKeysObtained = 0;
10. [HideInInspector] public bool materialsForRadioCollected = false;
11. [HideInInspector] public bool redEndingActivated = false;
12. [HideInInspector] public bool playerHasGivenFoodToBlue = false;
13. [HideInInspector] public int missionLevel2EnemiesEliminated = 0;

```

Figure 53 - Data manager

8 RESULTS

8.1 Hours dedicated approximately

Task	Estimated hours	Final hours
Game concept design, investigation and search of references	25	25
Creation of the tridimensional models	50	100
Level design	50	50
Programming	100	100
Creation of the character dialogues and other narrative elements	50	75
Development of the project report and the presentation	25	25
Total:	300	375

Figure 54 - Comparison between planned hours and real dedicated hours

8.2 Project results

Task	Result
Creation of the tridimensional models	118 models
Level design	4 sketches
Programming	113 scripts
Creation of the character dialogues and other narrative elements	144 pages + 14 Twine files

Figure 55 - Project results

8.3 Access to the code

Link to the Github repository:

<https://github.com/RaulMartinezEspinosa/The-Dark-Age>

Github user:

al225699@uji.es

Github password:

TheDarkage123

8.4 Access to all the project files

https://drive.google.com/drive/folders/1gUm9BFhwriTxhd-6-gWRWeXj_R19tung

9 CONCLUSIONS

The objectives of this project are, as we have said, to design a game based in a narrative that makes the player feel that he is living in a post-apocalyptic world where he has to survive, to create paths and dialogues that allow him/her to notice the importance of the decisions he has chosen, to maintain a balance between gameplay and narrative and to develop an immersive narrative. In that sense, it is important to say that in this section we are going to say why we think that we have accomplished all these objectives.

Firstly, the aesthetic of the levels and the dialogues that the player has with the characters transmit the sensation of living in a post-apocalyptic world where people have to fight to survive. Also, the artistic design election (using characters that are cubes and objects that have cubic shapes) does not affect negatively these objectives, as it could be thought, but they even help to transmit that perception.

On the other hand, the decisions that the player makes during the game have visible outputs, so he can see the importance of them immediately. Moreover, as the dialogues with each character advance these decisions change greatly the story, so he can see that his elections have a lot of importance. This sensation increase even more when the player begins a new level and tries different options because he can see the paths that were hid in previous games.

Furthermore, the balance between gameplay and narrative is achieved because the player's actions are essential to change the story of the game. Also, both characteristics (gameplay and narrative) need each other to make the game advance, so the player does not feel that neither of them is unnecessary.

Finally, the narrative is immersive because each character has its own personality and acts accordingly to it and the information that the player receives let him know many things about the world.

10 FUTURE WORK

We want to improve this game in the future in several ways:

- Adding more depth to the narrative: Currently the development of the characters is hard because the dialogues are short. For that reason, in order to make a more engaging narrative experience, we have planned to increase the number of dialogues that the player has with each character and the length of those dialogues.
- Increasing the content of the levels: The levels of this initial version have a few things to do, this causes that the player finishes them too fast and does not have time to appreciate all its features. To avoid this situation we want to make more enemies, buildings, characters, missions and places.
- Adding more levels: At the current state of the game there are four levels. In them there are some secret areas and unlockable places that can be accessed if the player completes some requisites before, but these locations are minor spaces inside those levels. It could be interesting to improve the importance of those areas by giving them their own spaces. To do it we want to create secret and unlockable levels.

A BIBLIOGRAPHY

Books

[1] Derakhshani, D. (2014). *Introducing Autodesk Maya 2015: Autodesk Official Press*. Sybex.

[2] Lebowitz, J. and Klug, C. (2011). *Interactive Storytelling for Video Games: A Player-Centered Approach to Creating Memorable Characters and Stories*. Focal Press.

[3] Smith, M. and Queiroz, C. (2015). *Unity 5.x Cookbook*. Packt Publishing.

Video games

[4] Interplay Entertainment, Black Isle Studios, Micro Forté, Bethesda Game Studios, Obsidian Entertainment. (1997 - 2019). Fallout franchise. [Video game]. Interplay Entertainment, 14 Degrees East, Bethesda Softworks.

[5] Playdead. (2016). Inside. [Video game]. Playdead.

Movies

[6] Hillcoat, J. (Director), & Wechsler, N., Schwartz, S., & Schwartz, P. M. (Producers). (2009). The Road [Motion picture] United States: Dimension Films.

Websites

[7] Twine. (2019). [online] Available at: <https://twinery.org/>

[8] Unity Documentation. (2019). CharacterController.Move. [online] Available at: <https://docs.unity3d.com/ScriptReference/CharacterController.Move.html>

- [9] Unity Documentation. (2019). Character Controller. [online] Available at: <https://docs.unity3d.com/Manual/class-CharacterController.html>
- [10] Unity Documentation. (2019). Vector3.MoveTowards. [online] Available at: <https://docs.unity3d.com/ScriptReference/Vector3.MoveTowards.html>
- [11] Unity Answers. (2016). A Proper Way To Pause A Game. [online] Available at: <https://answers.unity.com/questions/1230216/a-proper-way-to-pause-a-game.html>
- [12] YouTube. (2017). Smooth Camera Follow in Unity - Tutorial. [online] Available at: <https://www.youtube.com/watch?v=MFQhpwc6cKE>
- [13] Unity Answers. (2011). Referencing non static variables from another script? C#. [online] Available at: <https://answers.unity.com/questions/42843/referencing-non-static-variables-from-another-scri.html>
- [14] Unity. (2013). Inheritance. [online] Available at: <https://unity3d.com/es/learn/tutorials/topics/scripting/inheritance>
- [15] Unity Forum. (2014). Modify the width and height of RectTransform. [online] Available at: <https://forum.unity.com/threads/modify-the-width-and-height-of-recttransform.270993/>
- [16] Unity Answers. (2012). Destroying a prefab bullet after time. [online] Available at: <https://answers.unity.com/questions/250873/destroying-a-prefab-bullet-after-time.html>
- [17] Unity Documentation. (2019). Object.Instantiate. [online] Available at: <https://docs.unity3d.com/ScriptReference/Object.Instantiate.html>
- [18] Unity Documentation. (2019). Transform.LookAt. [online] Available at: <https://docs.unity3d.com/ScriptReference/Transform.LookAt.html>
- [19] Unity Answers. (2014). How to make an object go the direction it is facing? [online] Available at: <https://answers.unity.com/questions/616195/how-to-make-an-object-go-the-direction-it-is-facin.html>

[20] Unity. (2013). Overriding. [online] Available at:
<https://unity3d.com/es/learn/tutorials/topics/scripting/overriding>

[21] Unity Documentation. (2019). Button.onClick. [online] Available
at: <https://docs.unity3d.com/ScriptReference/UI.Button-onClick.html>

Assets

[22] Freesound. (2019). Dark_ambient.wav. [online] Available at:
<https://freesound.org/people/Aleks41/sounds/470643/>