



Creation of a complex narrative using emergent multidimensional storytelling

Daniel Garín Boira

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

July 4, 2023

Supervised by: Antonio Loriguillo López, PhD.



To all of those that have supported my endeavours

ACKNOWLEDGMENTS

To begin with, I would like to thank my Final Degree Work supervisor, Antonio Loriguillo López, PhD, for his tutoring and contribution in making this project a reality.

I want to also thank my parents, who have always stayed by my side, not only within the context of this project, but throughout all the hardships that I have endured during my life.

In addition, I also want to thank my best friend Paula, who has been a constant ray of light during my time at university, which would have been so much harder without her presence.

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

This document aims to present a technical report of the Bachelor Thesis "Creation of a complex narrative using emergent multidimensional storytelling". This work consists in the creation of an entire video game experience, that focuses on the use of an ambiguous game design methodology that tries to portray a narrative in an emergent and immersive manner, enticing the players to engage with the story without the need of restricting their freedom and choices. This is further enhanced with a unique mechanic that not only gives the player spatial freedom, but also temporal freedom, allowing the travel between two different eras of the same scene that connect with each other to create one cohesive story.

Through the use of a specific level and gameplay design that cater towards an emergent narrative, the narrative design strives to tell a story in an indirect (hence emergent) manner, enticing the player to search and interpret the information they find in order to make up their own interpretation of the narrative.

CONTENTS

Contents	v
1 Introduction	1
1.1 Work Motivation	1
1.2 Objectives	2
1.3 Environment and Initial State	3
1.4 Additional Information	4
2 Planning and resources evaluation	5
2.1 Planning	5
2.2 Resource Evaluation	8
3 System Analysis and Design	11
3.1 Requirement Analysis	11
3.2 System Design	13
3.3 System Architecture	23
3.4 Interface Design	23
4 Work Development and Results	27
4.1 Narrative Design	28
4.2 Level Design	31
4.3 Gameplay Design	34
4.4 Programming	35
4.5 Results	39
5 Conclusions and Future Work	41
5.1 Conclusions	41
5.2 Future work	42
Bibliography	43

INTRODUCTION

Contents

1.1	Work Motivation	1
1.2	Objectives	2
1.3	Environment and Initial State	3
1.4	Additional Information	4

The paragraphs below explain the reasoning of why this project was born, from a personal motivation to how it can objectively contribute to the video game medium. Furthermore, several objectives have been set with the intent of justifying the previously mentioned motivations.

1.1 Work Motivation

From a personal standpoint, my goal in the video game field has always been to become a designer and be able to offer experiences that make use of all the different aspects of creating a video game (such as level, gameplay and narrative design) to create works that can not only fulfill my life, but also inspire and instill emotions in the people that play my games, making them think and reflect about the experience they are going through. This project to me was the true beginning of that, since it involved using the best of my abilities that I possess in the several fields that compose the designing aspect of a video game.

Besides, linking to the more objective motivation for this project, I wanted to create a story that made use of an underappreciated form of storytelling such as emergent narrative, which tends to be overshadowed by other, more direct storytelling disciplines like embedded narrative. However, my aim was not to replace one another, but to use them in conjunction in order to enhance the experience. Now, from a less individual point of view, the more objective motivation was to, as I said in the paragraph above, shine a light on the underused emergent narrative. Emergent narrative is able to achieve a symbiotic relationship with video games thanks to the unique interactive aspect that they possess [1]. Several times, it has been proven how it can enhance a story, with great examples like *Souls-like FromSoftware* games, such as *Elden Ring* [2] that make use of this type of narrative to add a certain unique aspect that is now characteristic to all their games, *XCOM 2* [3], with procedurally generated characters and systems that interact with each other to give every player a unique story along with the embedded narrative; and the biggest inspiration for my project, *Gone Home* [4], which perfectly combines both types of storytelling, and portrays a touching and engaging story that can be interpreted differently by each player.

Although these are great examples, they are still a minority when compared to the whole industry. Therefore, with my game I aim to add new ideas that can prove that video games, although inspired by other mediums such as film-making, can stand out on their own using their unique structural capabilities.

1.2 Objectives

The objectives to fulfill can be identified as those aspects of game design that need to be properly implemented for the whole system to work. Each one of them cannot properly exist without the other. Therefore, for the narrative to be engaging each design objective will need to be adequately implemented.

- **Level Design:** Emergent narratives are known for telling their stories through unconventional channels like item interaction and environmental storytelling. Therefore, a level like this needs to have a deliberate object placement that guides the player vision. Furthermore, since finding items to interact with is also a core component, the level requires to be design in a way that entices the players to approach the places where those objects can be found. This can be achieved in several ways, such as placing objects that stand out above the rest or using visual paths that guide the players vision towards the desired places (like sunlight rays or object in the scene pointing towards a specific direction).

- **Gameplay Design:** As it has been previously mentioned, emergent narrative is a storytelling device that greatly benefits from the interactivity of video games. Therefore, designing gameplay in way that connects with how the story is told is a requirement for the project to function. For this to work, the actions and events implemented in the game will all relate to the narrative in some form, such as item interaction prompting a certain dialogue for the player to interpret, or puzzles and exploration design in way that requires the player to analyze the environment in order to progress.
- **Narrative Design:** Both objectives mentioned above cohesively connect with the narrative design, meaning that, for the narrative to work it has to be integrated in both the level and the gameplay. The game will still display embedded narrative through written notes, but these will be vague enough so that the player can make their own assumptions about them, making them both emergent and embedded in nature. For fully emergent environmental storytelling, the narrative needs to be designed in a way that permeates the setting, and is able to tell a story solely through the player interpretation of what and how objects are placed in the scene.

1.3 Environment and Initial State

This project was developed individually by myself, being in charge of all the design and programming of the video game, aiding myself in the artistic department with assets acquired from the Unreal Engine Marketplace, such as the foundation for the house [5], several assets for the rooms [6] and surface materials to personalize both versions of the house (inhabited and dilapidated) [7]. Working by myself posed a challenge, due to the different disciplines involved in the creation of a video game, but also proved to be a great learning experience for a software new to me (Unreal Engine) and for learning about the different aspects of design. Furthermore, being the sole creator offered complete creative freedom and greatly reduce time investment in organization.

During development, an appropriate schedule was followed, where I worked around 3 to 4 hours 5 days a week, with some days adding extra time to finalize implementations or research about systems that I aimed to implement throughout development.

The initial state of the project came to be thanks to an Unreal Engine course about creating a *First-Person Survival Horror* [8] and also due to having access to several free assets within the Marketplace to create my own level and environment.

1.4 Additional Information

1.4.1 Keywords

Emergent narrative, video game design, multidimensional, exploration game, puzzle game

1.4.2 Related Subjects

- Hyper Media Narrative and Video Game Analysis (VJ1218).
- Video Game Conceptual Design (VJ1222).
- Theory and Practice of Audiovisual Production (VJ1230).
- Artistic Expression (VJ1204).
- Software Engineering (VJ1224).
- Algorithms and Data Structures (VJ1215).

PLANNING AND RESOURCES EVALUATION

Contents

2.1	Planning	5
2.2	Resource Evaluation	8

This chapter will explain the planning followed for the development of the video game. Furthermore, it will detail the approximate cost of resources used for the creation of the game, using both real calculations and simulated situations.

2.1 Planning

Taking into account how important planning is for the development of a video game, great consideration was taken into creating and following the scheduled plan. This tasks comprise from the conception of the game to the implementation of all its parts, also adding the time taken to write official documents such as this one.

- **Software and component installation:** After careful evaluation of what software would be needed for the development of the project, the necessary programs were installed and research was made about how to use them.
- **Concept:** This time was spent conceptualizing the future stages of the game, from the first strokes of the narrative to what type of gameplay the game was going to have. The foundation of the level was also started at this stage in order to test ideas.
- **Narrative creation:** Most of the time spent was in creating a cohesive narrative and thinking how it could be implemented in an immersive and emergent manner.

The remaining time was invested in implementing a few narrative focused events (such as notes) to prototype its functioning.

- **Programming:** As it was to be expected, programming was going to take most of the allotted time for the game to function appropriately. This included basic gameplay like character movement, which was done first and took a relatively short time, to more complex actions like an interaction and inventory system or a save system that would allow for every taken or displaced object to be properly stored when exiting the game.
- **Game implementation:** This stage involved converging all of the previously mentioned implementations into one cohesive unit. The level was created in a way that would benefit the narrative, using an assortment of different assets and materials to decorate the house in a way that would incite environmental storytelling. As per the gameplay, it involved adding all of the systems created during the programming phase to the level created, making sure that everything worked well together. Finally, the narrative element was introduced through level design and gameplay implementation, using these systems in a way that would relay a story to the player.
- **Testing and optimization:** An important part of development was making sure that the gameplay would actually feel immersive and engaging and that the game itself would properly work, in both a lack of bugs and an adequate performance.
- **Final report:** This encompasses the time spent writing all the required documents, such as this report and the presentation. Some of that time was spent researching information and creating diagrams to better explain certain concepts.

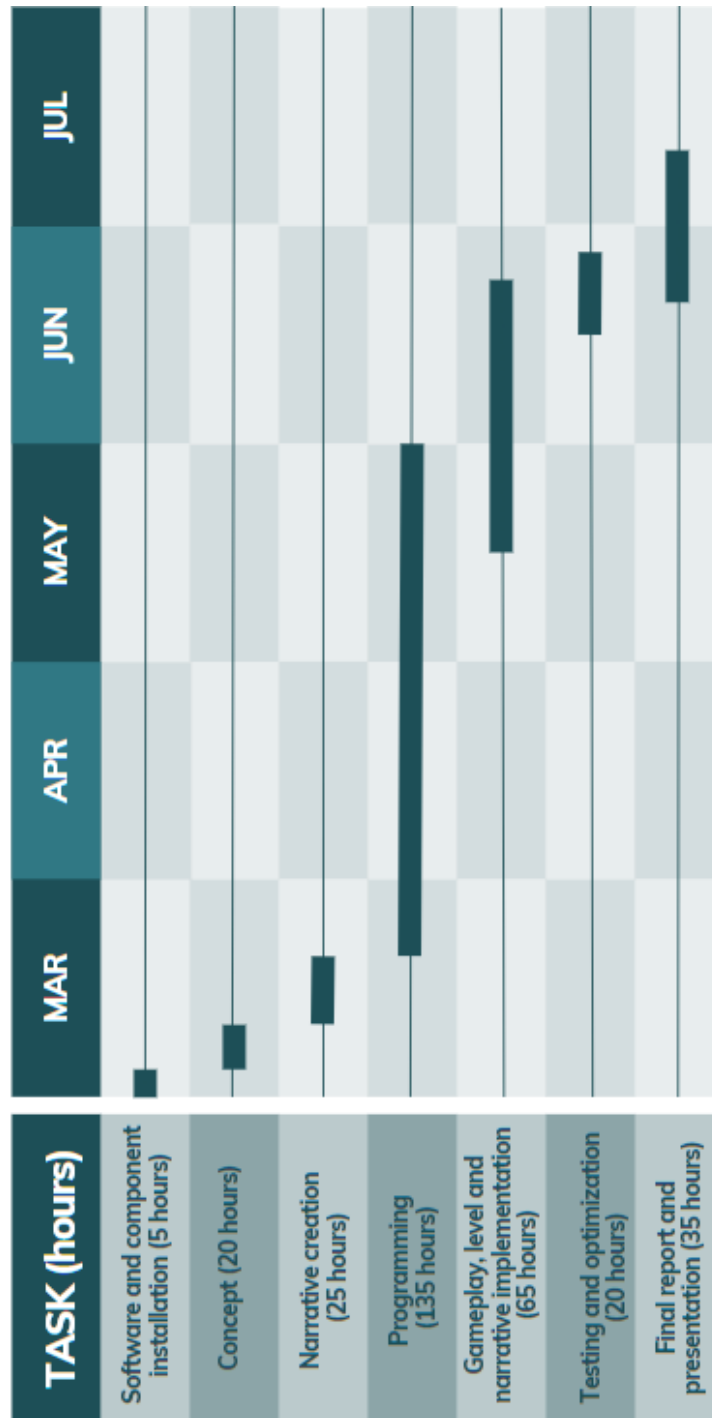


Figure 2.1: Example of a Gantt chart (made with Kplato)

2.2 Resource Evaluation

Before getting into the cost of the software and hardware used, it's important to talk about the human cost of developing a video game. In this case many disciplines were involved to develop the game, so we will envelop all of it under the Unreal Engine Developer umbrella. In this case, the average salary of a UE developer is around €36.667 gross yearly [9]. Taking into account the time worked in this project, this will result in €1.147,05 (€2.294,10/2 due to working 4 hours instead of 8 [10]) monthly net income, totalling to €4.588,2 net income during the four months of development.

Now pertaining to software and hardware costs, beginning with hardware:

- **Hardware:** Most of the components are not up to date but they were powerful enough to be able to comfortably develop the game. The cost totals to €1.112,60.
 - **CPU:** Intel Core i5-7600K (€205,88)
 - **GPU:** GeForce GTX 1050 Ti (€199)
 - **RAM:** Kingston FURY Beast 36GB (€150)
 - **Motherboard:** Gigabyte Z270-HD3P-CF (€172,05)
 - **Storage:** Crucial P3 Plus 1TB PCIe M.2 2280 SSD (€125,83)
 - **Power Supply:** Nox Urano VX 650W (€49,99)
 - **Monitor:** AOC 2476WM (€160,50)
 - **Keyboard:** Logitech G203 (€25)
 - **Mouse:** Mars Gaming MKAXPES TKL H-Mech (€24,35)
- **Software:** Unreal Engine was the software mostly utilized for development, but some other additional software for asset acquirement and modification has been used. The cost totals to 15€.
 - **Windows 10 Home 64.bit (€145):** Operative system installed within the computer used to develop the project.
 - **Unreal Engine 4.27 and 5.1.1 (Free):** Chosen for its visual scripting capabilities and its many ready to use systems to begin development without the need of implementing foundational systems.
 - **Unreal Engine Marketplace (€15):** The Marketplace offers a great assortment of permanent free assets and paid assets given for free on a monthly basis. However, €15 were spent in acquiring an asset pack.
 - **Blender (Free):** Used for editing certain assets that didn't fit in certain aspects.
 - **Megascans (Free):** Thanks to a alliance between Quixel and Epic Games, all assets inside Megascans can be freely used in Unreal Engine.

-
- **Canva (Free):** For creating some of the diagrams used in this report.
 - **Overleaf:** LaTeX online editor used to write this document.
 - **Replica:** To generate an AI voice for the character whenever it interacts with certain objects.

SYSTEM ANALYSIS AND DESIGN

Contents

3.1	Requirement Analysis	11
3.2	System Design	13
3.3	System Architecture	23
3.4	Interface Design	23

This chapter talks about the requirement analysis, design and architecture of the system and the design of the UI and HUD.

3.1 Requirement Analysis

This analysis will evaluate the specific requirements that have to be met for the game to properly work. In this case, these are not objectives to be met but compulsory requirements that have to be accomplished for the game to become a cohesive and meaningful experience. Before starting with said requirements, an explanation of the game functioning will be given.

To begin with, as per any other game, when the game is started the player is prompted into a main menu where several options can be chosen. These are *Continue* which will resume the game where it was last left off, *Start*, which will begin the game from the beginning, getting rid of any previous save, *Options* which will open an extra menu with options to change the audio and the graphic quality, and to show the controls for the game. Finally there's *Exit*, which will close the game.

When the option to start or continue the game is chosen the player will be promptly taken into the level. This single level will hold all spaces that the player can traverse, including both instances of the houses and a few micro levels where certain events will be played. The player can interact with the level in several ways, being able to pick up items distributed around the environment and adding them to their inventory. Other objects will be able to be interacted with so that certain narrative events (such as a note, a dialogue or simply the nature of an object) can be related to the player. The level will also have several puzzles, some involving the search of a key and others more specific mechanics. Finally, there's the core mechanic of being able to travel between both instances of the house. This will be possible through the interaction with certain objects that are of great narrative importance and will involve the game fading to black so that behind the screen the player can be teleported to the appropriate place in the other house.

3.1.1 Functional Requirements

A functional requirement defines a function of the system that is going to be developed. This function is described as a set of inputs, its behavior, and its outputs. These requirements define the entire system in a concrete way so that it is perfectly understood what can be done with it at any given moment. These requirements are:

- **R1:** The player can resume the game by pressing *Continue*.
- **R2:** The player can start the game by pressing *Start*.
- **R3:** The player can access the options by pressing *Options*.
- **R4:** The player can close the options by pressing *Exit*.
- **R5:** The player can change audio settings by modifying sliders within *Audio*.
- **R6:** The player can change graphic settings by picking options within *Graphics*.
- **R7:** The player can move the character by using *WASD*.
- **R8:** The player can crouch by pressing *C*.
- **R9:** The player can interact with items by pressing *E*.
- **R10:** The player can open and close the inventory by pressing *TAB*.
- **R11:** The player can examine, drop or use an item in the inventory by clicking on the item and choosing an available option.
- **R12:** The player can open physical objects like doors, drawers and suitcases by left clicking and dragging the mouse.
- **R13:** The player can unlock a locked physical object by right clicking them while having the required key in their inventory.

- **R14:** The player can interact with a lock and introduce a number combination by clicking on it and then clicking on each number to rotate them.

3.1.2 Non-functional Requirements

Non-functional requirements impose conditions on the design or implementation. These can be imposed for reasons like platform compatibility or to achieve a certain degree of polish when creating the product. In this case the non-functional requirements are:

- **R15:** The game will be available on PC.
- **R16:** The narrative should be understandable solely through interaction with items and environment.
- **R17:** The user interface should be extremely minimalist, only showing dialogue said by the character and icons when looking at certain objects.
- **R18:** The controls will be simple and easy to use.

3.2 System Design

The tables below will define the different use cases with functional requirements as a basis. Diagrams will be used to portray how all of these tie together:

Requirement:	R1
Actor:	Player
Description:	The player can resume a saved game by pressing <i>Continue</i>
Preconditions:	<ol style="list-style-type: none"> 1. Player must be at the main menu 2. Player must have a saved game
Normal Sequence:	<ol style="list-style-type: none"> 1. Player presses <i>Continue</i> 2. Level loads and becomes playable
Alternate Sequence:	None

Table 3.1: Case of use «Continue game»

Requirement:	R2
Actor:	Player
Description:	The player can start the game by pressing <i>Start</i>
Preconditions:	<ol style="list-style-type: none"> 1. Player must be at the main menu
Normal Sequence:	<ol style="list-style-type: none"> 1. Player presses <i>Start</i> 2. New game is created 3. Level loads and becomes playable
Alternate Sequence:	None

Table 3.2: Case of use «Start game»

Requirement:	R3
Actor:	Player
Description:	The player can access the options by pressing <i>Options</i>
Preconditions:	<ol style="list-style-type: none"> 1. Player must be at the main menu
Normal Sequence:	<ol style="list-style-type: none"> 1. Player presses <i>Options</i> 2. Options menu opens
Alternate Sequence:	None

Table 3.3: Case of use «Options menu»

Requirement:	R4
Actor:	Player
Description:	The player can close the game by pressing <i>Exit</i>
Preconditions:	<ol style="list-style-type: none"> 1. Player must be at the main menu
Normal Sequence:	<ol style="list-style-type: none"> 1. Player presses <i>Exit</i> 2. Game closes
Alternate Sequence:	None

Table 3.4: Case of use «Exit game»

Requirement:	R5
Actor:	Player
Description:	The player changes the sound by moving sliders within <i>Audio</i>
Preconditions:	<ol style="list-style-type: none"> 1. Player must be at the options menu
Normal Sequence:	<ol style="list-style-type: none"> 1. Player presses <i>Audio</i> 2. Player changes audio
Alternate Sequence:	None

Table 3.5: Case of use «Change audio»

Requirement:	R6
Actor:	Player
Description:	The player changes graphic setting by picking within <i>Graphics</i>
Preconditions:	<ol style="list-style-type: none"> 1. Player must be at the options menu
Normal Sequence:	<ol style="list-style-type: none"> 1. Player presses <i>Graphics</i> 2. Player changes graphics
Alternate Sequence:	None

Table 3.6: Case of use «Change graphics»

Requirement:	R7
Actor:	Player
Description:	Player moves character by using <i>WASD</i>
Preconditions:	<ol style="list-style-type: none"> 1. Player must be inside the game
Normal Sequence:	<ol style="list-style-type: none"> 1. Player presses the keys 2. Character moves
Alternate Sequence:	None

Table 3.7: Case of use «Character movement»

Requirement:	R8
Actor:	Player
Description:	Player makes the character crouch by pressing <i>C</i>
Preconditions:	<ol style="list-style-type: none"> 1. Player must be inside the game
Normal Sequence:	<ol style="list-style-type: none"> 1. Player presses the key 2. Character crouches
Alternate Sequence:	None

Table 3.8: Case of use «Character crouching»

Requirement:	R9
Actor:	Player
Description:	Player can interact with certain objects by pressing <i>E</i>
Preconditions:	<ol style="list-style-type: none"> 1. Player must be inside the game 2. Player must be close to an intractable item
Normal Sequence:	<ol style="list-style-type: none"> 1. Player presses E 2. Specific interaction with that item occurs
Alternate Sequence:	None

Table 3.9: Case of use «Character interaction»

Requirement:	R10
Actor:	Player
Description:	Player opens or closes the inventory by pressing <i>TAB</i>
Preconditions:	<ol style="list-style-type: none">1. Player must be inside the game2. <ol style="list-style-type: none">a) To open inventory, it must be closedb) To close inventory, it must be open
Normal Sequence:	<ol style="list-style-type: none">1. Player presses the key2. Inventory opens
Alternate Sequence:	<ol style="list-style-type: none">1. Player presses the key2. Inventory closes

Table 3.10: Case of use «Inventory opens/closes»

Requirement:	R11
Actor:	Player
Description:	Player can use items in certain ways by clicking on them
Preconditions:	<ol style="list-style-type: none">1. Player must be inside the game2. Player must have the inventory open3. Player must have at least an item inside the inventory
Normal Sequence:	<ol style="list-style-type: none">1. Player clicks on item2. Player chooses an option
Alternate Sequence:	None

Table 3.11: Case of use «Inventory item usage»

Requirement:	R12
Actor:	Player
Description:	Player can open physical items by left clicking and dragging the mouse
Preconditions:	<ol style="list-style-type: none">1. Player must be inside the game2. Player must be close to the object3. Player must left click and drag
Normal Sequence:	<ol style="list-style-type: none">1. Player left clicks on object2. Player drags mouse3. Object opens
Alternate Sequence:	None

Table 3.12: Case of use «Physical object interaction»

Requirement:	R13
Actor:	Player
Description:	Player can unlock a locked physical object by right clicking on them while have the necessary key
Preconditions:	<ol style="list-style-type: none">1. Player must be inside the game2. Player must be close to the door3. Player must have the necessary key in their inventory4. Player must right click the door
Normal Sequence:	<ol style="list-style-type: none">1. Player right clicks on locked physical object2. Physical objects unlock
Alternate Sequence:	None

Table 3.13: Case of use «Unlock locked physical object»

Requirement:	R14
Actor:	Player
Description:	Player can interact with a lock to try and solve it
Preconditions:	<ol style="list-style-type: none">1. Player must be inside the game2. Player must be close to the lock3. Player must interact with the lock
Normal Sequence:	<ol style="list-style-type: none">1. Player interacts with lock2. Player solves the lock3. Lock opens
Alternate Sequence:	None

Table 3.14: Case of use «Lock interaction»

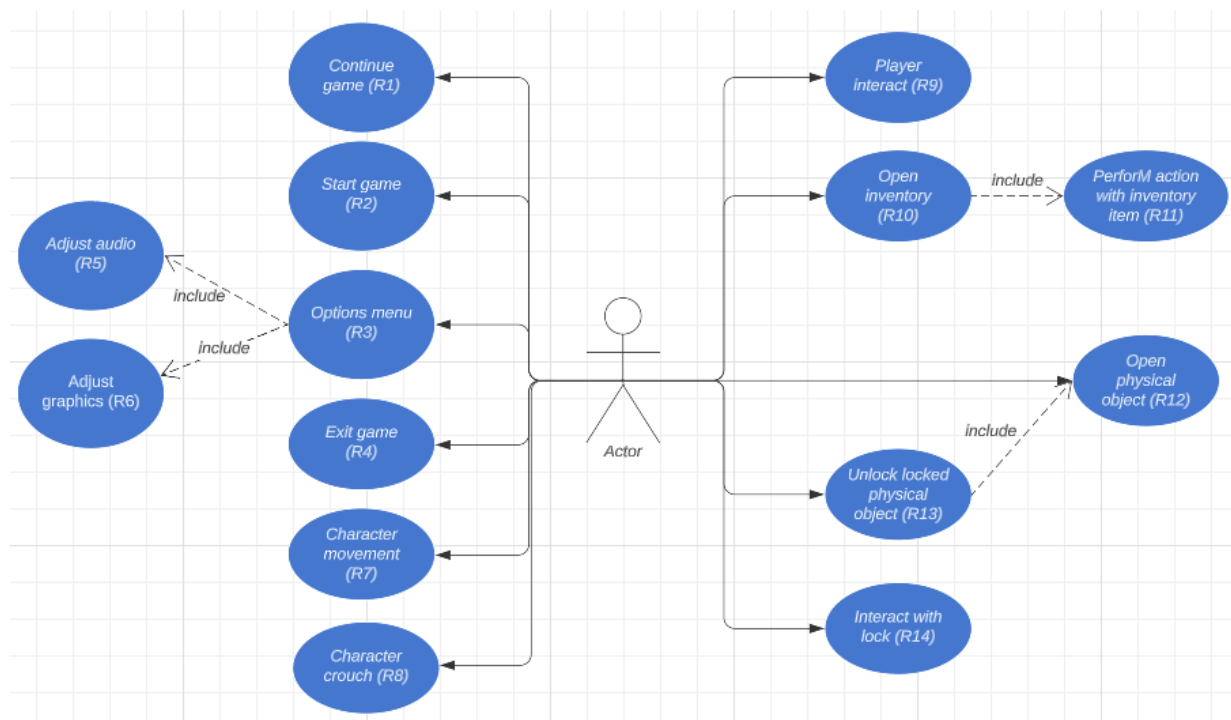


Figure 3.1: Case use diagram (made with Lucidchart)

3.3 System Architecture

The game has only been tested on my own computer, but with different graphic configurations. Therefore the following minimum requirements are just an approximation:

- **Operating system:** Windows 7 64-bit.
- **CPU:** Quad-core processor 2,5GHz.
- **GPU:** Nvidia GeForce GTX 960.
- **RAM:** 8 GB.

3.4 Interface Design

The menu will simply allow the player to transition into the game or change a few settings such as graphics and audio.

The interface seen during most of the game is just a single point in the middle of the screen to show the player where they are look at.



Figure 3.2: Main menu before starting the game

The interface seen during most of the game is just a single point in the middle of the screen to show the player where they are look at.



Figure 3.3: HUD point

When being nearby an intractable and/or pick up-able item, the prompt to tell the

player which key to press (E) is shown on screen. Its design is quite simple to not detract any attention from the actual game. The prompt is also shown as part of the item instead of being part of the player's HUD. This way is much easier to identify where these intractable items are within the scene.



Figure 3.4: Item interaction

Other user interface implemented is the one shown when clicking an inventory item. It shows the different actions that can be taken with said item, such as dropping it to the ground or examining it.



Figure 3.5: Item interaction

When clicking *Examine*, an extra interface appears, where a 3D model of the item is shown and can be rotated and examined by the player. Furthermore, its name and a short description of what it is appears to the left and right side of the screen, respectively.



Figure 3.6: Item interaction

WORK DEVELOPMENT AND RESULTS

Contents

4.1	Narrative Design	28
4.2	Level Design	31
4.3	Gameplay Design	34
4.4	Programming	35
4.5	Results	39

This section will be a very detailed analysis and explanation of the work done to develop the video game. This includes systems and mechanics used and the reasoning of why those decision were taken. In order to make everything easier to understand, the information in this section will be contained within several subsections pertaining to each one of the three main design components of the video game, these being narrative, level and gameplay design. This has been done because in chronological order, several aspects of each design were developed in conjunction with each other, which would make it too confusing to be put into words on this report.

4.1 Narrative Design

4.1.1 Idea and intention

The original idea for the development of this game came from a growing interest in several aspects. One of them was wanting to create an entire video game in Unreal Engine, which was chosen for its ease of use, great graphic fidelity and the recent rise in popularity within the industry. This can be seen in many studios, from indies to AAAs, switching to Unreal Engine 5 [11] and in the recent virality of certain UE5 projects such as *The Matrix Awakens* [12] and *Unrecord* [13]. The other aspect which incited a deep interest was wanting to create a video game that could prove that emergent narrative is one of video game's greatest strength to differentiate themselves from every other medium.

For these reasons, and taking into account that the game was going to be developed single-handedly, it was decided that the best idea was for the game to have very little mechanical complexity, and instead base everything around its narrative.

As for the narrative itself, the objective was to convey a feeling that everyone has experienced up to some degree at some point in their life, and this feeling ended up being **guilt**. Moreover, thanks to how emergent narrative is naturally prone to interpretation, this was used as a basis that would allow any player to understand the story and the guilt the character felt using their own experiences as a benchmark. Deciding what things one is truly guilty of and how much one deserves to suffer or to overcome that guilt.

4.1.2 Story

The story is that of a man who, after losing his family in a horrible accident, becomes ridden with guilt and decides to leave the house and go far away. Years later, with a creeping feeling of guilt that was beginning to consume him, he decides to come back to the house in order to finally overcome his guilt or end up being completely consumed by it. From this point onwards is when the game takes place, as the player will explore the environment and find out where that feeling of guilt comes from.

As it has been mentioned several times in this report, the narrative is an emergent one, meaning that the player will make up their own story with the information found. Any narrative made up by the players' minds when analyzing all the different narrative components will be a valid interpretation of the story. This way, as it has been mentioned, using their own experiences each player will be able to uncover and judge information regarding the accident and the family's dynamics and relationships, coming to a conclusion regarding how the character should deal with his guilt and, in the process, make the player see how they would react or would have reacted to a similar situation themselves.

4.1.3 Narrative implementation

The way narrative has been implemented is deeply interconnected with all of the design, not only the narrative one. The nature of this type of narrative is to be told not directly to the player, but through the interaction with the different systems that comprise the game. For a narrative focused game that wants to tell a specific story but leave most of it up to interpretation, emergence cannot be the only narrative base, an embedded type of narrative is also required to rely the information that the player will interpret. This embedded narrative involves objects such as notes written by the different characters of the game, members of the family or people adjacent to them. This is embedded in the way the information is relied to the player, through clear written text, however it has also an emergent nature due to how vague and open to interpretation some of that information is.

Beginning with the notes, these provide information relating to things like how the relationships between certain characters were, what the behaviour of the characters was, events previous or after the accident, and many more. Some of this information is not crucial, but is useful to make the world and characters feel more alive, like an actual family living in a house and not just an excuse for the story to exist. However, there is also information of greater importance, that talks about specific events that help the player understand more about the core part of the narrative. Most of the important information is more accessible to the player, allowing them to get a grasp of the story first and foremost, while the less important information will usually be found near and more hidden (the method how will be explained more in depth when level design is touched upon), in order to act as contextual compliment.

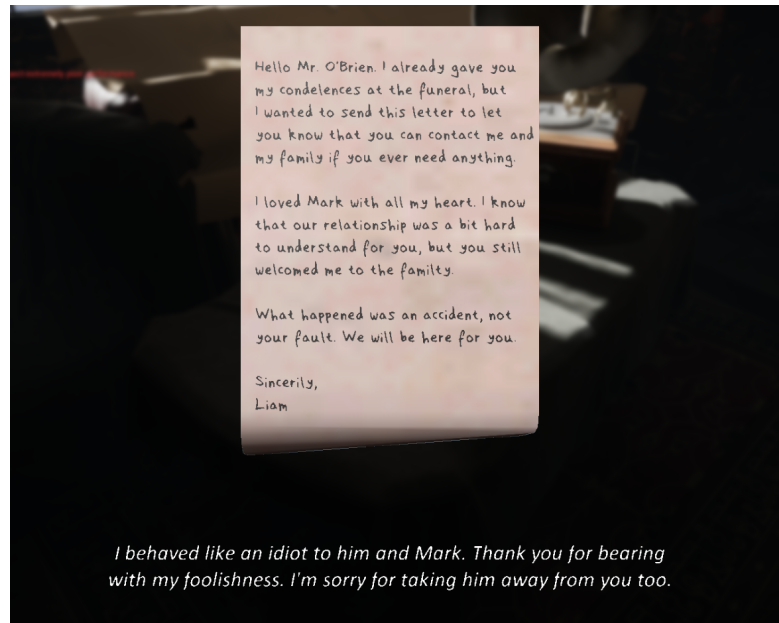


Figure 4.1: One of the note actors that the player can interact with

Now, the other type of narrative, which is more emergent in its form is the environmental one. In this case the information is relied entirely by object placement (of both intractable and prop items) and the aspect of the house, specially the changes between the two instances of the house when the player travels from one another. The information relied in this case is most of the time within that realm of complimentary information, that helps give context and understand more about characters' personalities and relationships. However there are specific cases where the environment gains importance and relies key information about the story. The best example of this is a scene that can be reached by accomplishing certain steps in a storyline. When the player finds out how to fully progress through this one storyline, at the end it will lead to the garage, where a set of suitcases lies near the family's car, and when interacting with the car, a scene will appear where the car is upside down and in flames, telling the player what exactly happened in the accident that took the family.



Figure 4.2: Snippet from the accident cutscene

4.2 Level Design

As it has been said before, all aspects of design have been made as a way to enhance the emergent nature of the narrative. In the case of the level design this involves creating the level in a way that guides the player towards those narrative events that have been mentioned. This type of guiding is not direct like an arrow, but indirect, using object placement and visual paths to guide the player's vision. Examples of this are things like sunlight rays bathing certain places within the scene or object being placed in a way that points towards a central focus where an important piece of narrative lies.

Another crucial part that I wanted to tell through the level design was the freedom of choice offered to the player [14]. This is another characteristic of emergent design, where the lack of a very clear embedded story gets rid of the rigidity of having to follow a set path in order to experience the story. In the specific case that is this level, from the get go players are free to explore different parts of the house, with no clear starting point. However, certain places inside the house cannot be accessed without the necessary keys or puzzles solved, this way, without limiting the player's options, a certain degree of restriction is imposed in order to not overwhelm the player. This high degree of freedom also offers being able to experience a different telling of the story depending on the order of exploration followed, since most narrative elements are highly susceptible to interpretation, and this interpretation depends on the information possessed.



Figure 4.3: Visual path using environmental lighting

Now, it is important to add that, even though there is no set path that the player is forced to follow, there still is certain progressions or storylines that are suggested to the player via level design. The reason is that, although some narrative elements are somewhat self-contained and can be uncovered in any order, a few other can be more specific and therefore better understood if followed in a specific order.

The best example to finally put all this level design into practice, will be the example given in the previous section about the truth of what specifically happened in the accident, however it will now be explained from a level design point of view. Basically, when players are in the past instance of the house when the family was still alive, and get to the kitchen, which is next to the house entrance (and therefore will be one of the first destinations of most players), they will find a note stuck on a locked door that gives access to the garage (this note will be very visible when entering the room, using those previously mentioned visual paths to guide the player's view). This note, written by the main character's wife tells the player that by mistake she placed the car keys inside a suitcase within the garage, and that the garage key should be in their bedroom. This will incite the player to look for said key in order to uncover more about this storyline, which means that the player, not knowing where the bedroom actually is, will at the same time explore other places of the house, engaging in new contained narratives and storylines. This example shows that by effective object placing and narrative design, the player can be suggested places to go, inviting them to explore the house in the order they desire and without forcing them in where to go.

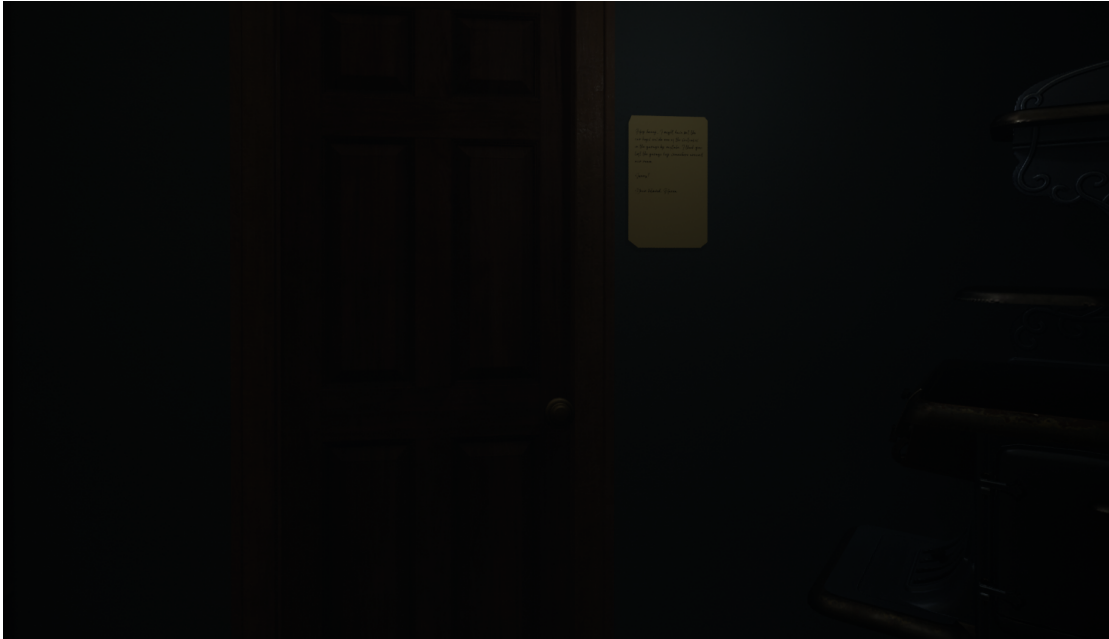


Figure 4.4: Note that kick-starts the garage storyline



Figure 4.5: Instance of the house from the present, dilapidated

4.3 Gameplay Design

Due to its narrative focus, the intent for the gameplay was to be quite simple and act purely as a complimentary aspect to the narrative. For this reason most of the gameplay was based around exploring the houses. However, simple pure exploration without any kind of impediment would not only make the experience more monotonous, but also act a detriment to the narrative. For this reason, and in conjunction with the level design, the exploration was made in a way that would require the player to acquire certain necessary items (such as keys) in order to access certain locations. As it has been mentioned in the example given within the level design section, this type of free but loosely guided exploration favors the narrative, offering a certain degree of cohesion and inciting the player to explore more than if total freedom was given.

Although exploration puzzles comprise most of the gameplay, there are still some other more classical puzzles implemented in the game. One of the most prominent is the existence of locks with a numeric combination. Its prominence is due to their nature as a typical object in everyone's home for its security, and how easy it is to connect to narrative elements (like hiding the code within a note or giving the code through certain elements in the environment to count).

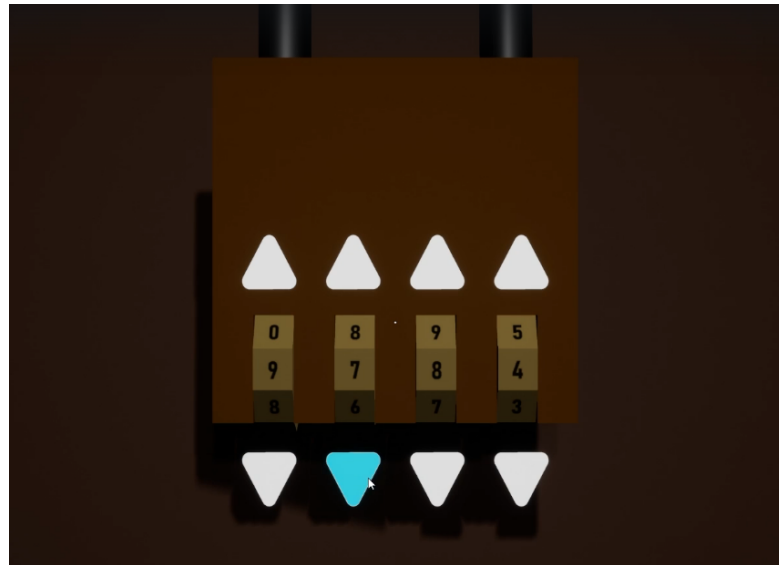


Figure 4.6: Lock puzzle

In short, being able to link not only the puzzles' solutions but also its significance and why they exist within the narrative, ends up providing a result that not only enhances the player experience, but also adds value to the narrative, which now doesn't need to rely purely on object examination and environmental design to be told.

4.4 Programming

This section will hover over all of the programming implemented but will only deepen when necessary, going into functions that have a great influence to the game itself.

The programming, following the emergent nature of the design, consist on a variety of systems that interact with each other and hence, that interaction provides an experience. Therefore, acting as a link of most of these systems is the most important blueprint of this project, L1_Character, which is the blueprint that implements the character and all of its functionalities. Since most systems in the game are based around the player interacting with them (such as intractable items, doors, puzzles, etc.), it only makes sense that the blueprint for the player will hold the information of what the player is currently interacting with. Aside from these interactions, there's also some other more specific systems that pertain purely to the character, such as movement or crouching.

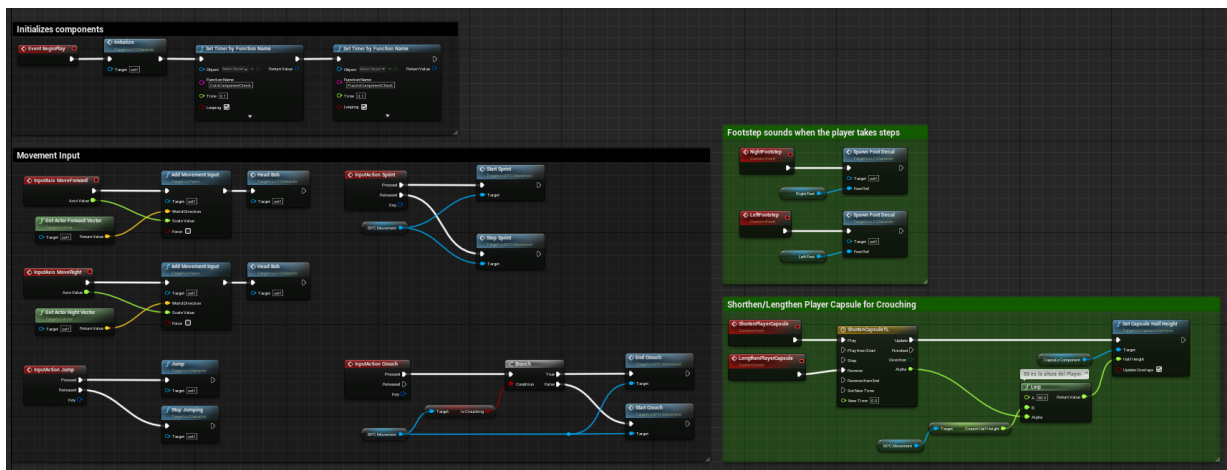


Figure 4.7: Character movement BP

When it comes to the specifics of the interaction system, there are two main ones, interaction based and physics based. Interaction based involve mainly objects like notes, attachable items or other miscellaneous that execute an event when interacted with, such as making a widget appear on screen with information about said item or lighting up a room. These are implemented by the use of line traces and collision spheres. When the conditions are met (being close to the object or be looking at it) the interaction will become available and pressing the key will activate the event attached to the object.

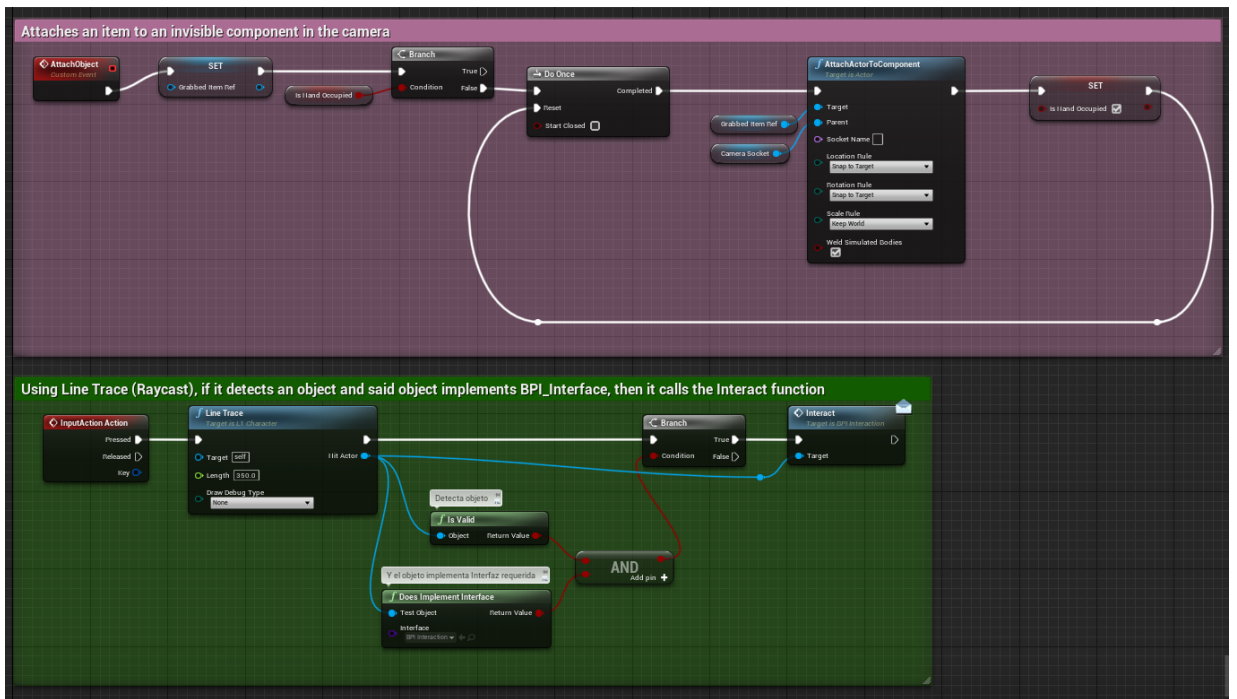


Figure 4.8: Interact with objects and attach objects to socket

On the other hand, physics based interactions involve moving or displacing objects that are simulating physics, like opening a door or grabbing an item and dropping it somewhere else. In this case line traces are also used, but these are complimented by other components like physics handles that register the specific place from where an object is being grabbed and updates on each frame so that the player can drag them across the scenery.

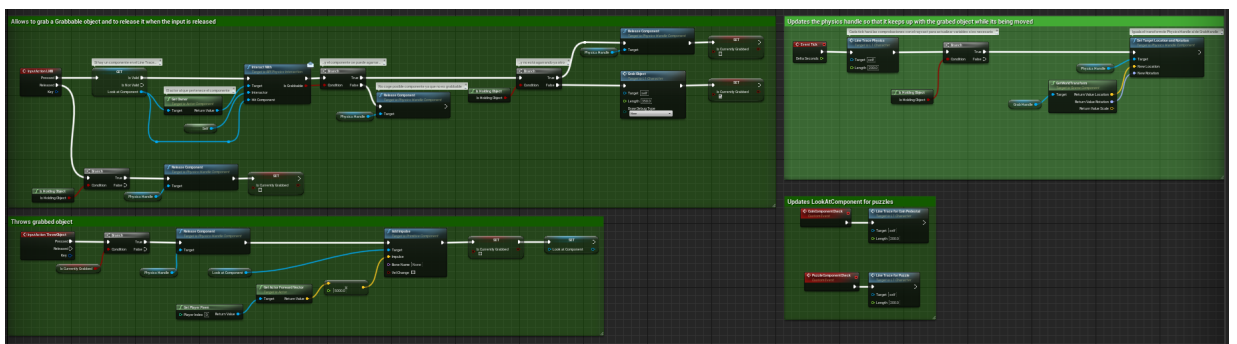


Figure 4.9: Grabbing physical items and interacting with puzzles

Another important script would be the one pertaining to the inventory. It will take care of all actions that involve the inventory, such as adding an item or removing it. Everything works by modifying in different ways an array where the information of each item is stored. This script is of type component, meaning that it has to be added to an actor and said actor will use its functions. The obvious choice for this actor is the Character, since the inventory will be used by him.

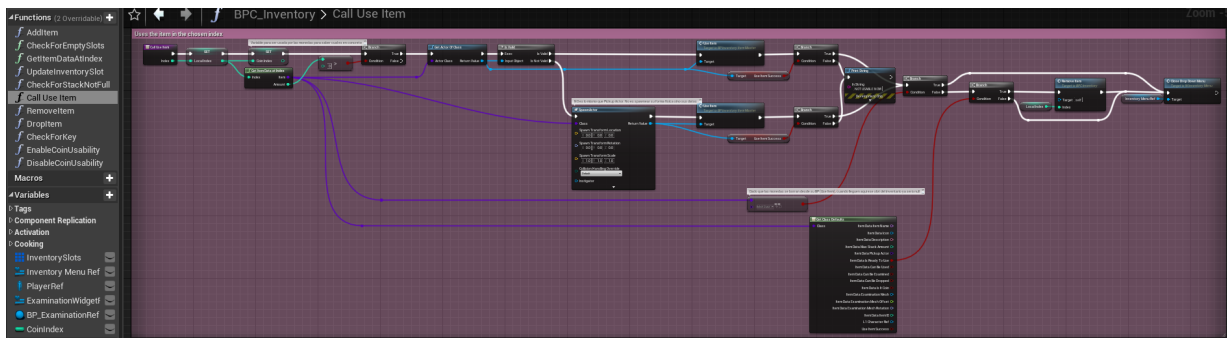


Figure 4.10: Left: inventory functions. Right: Use function implementation

A simple but crucial script is the one for the object examination. This script will simply detect when the player is nearby an object and it will activate its interactivity, meaning that it will allow the player to pick up the item. Once the player interacts with said object, a widget will be created where the object in question will be presented to be examined. In this screen the player is able to rotate the 3D model around to inspect it.

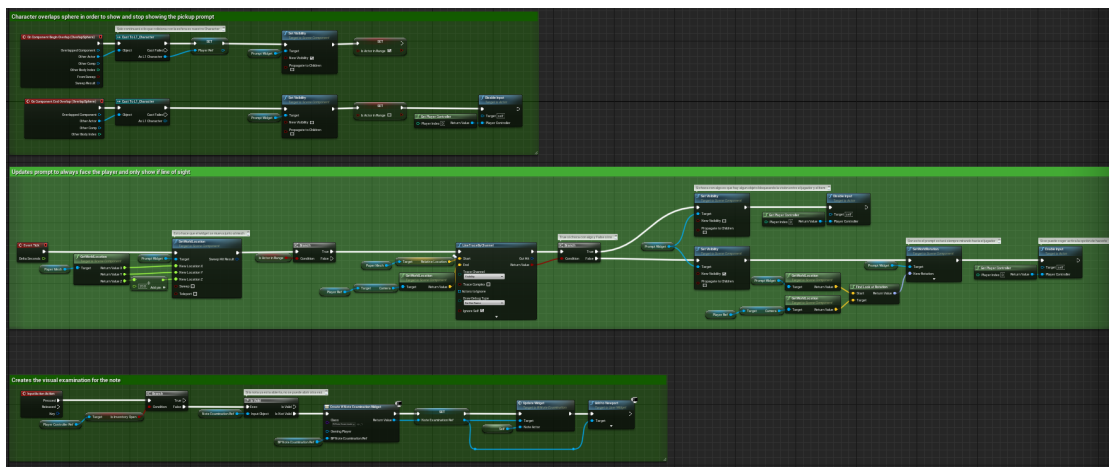


Figure 4.11: Note implementation

Finally, a script that is not necessarily part of the design but is key for the proper functioning of the game is the saving events. The saving will be done via the level blueprint, since most information to be saved will refer to that of the items distributed across the environment. The game is programmed to auto-save when certain actions or events take place. This may include things like using an item, unlocking a locked door, dropping an item, etc. When one of these actions is performed, a call is made from the object that triggered the event to the level blueprint. The blueprint will store the necessary values in a custom struct that has been designed to contain information about the different objects to be saved. Once the game is restarted the code in the level blueprint will trigger and it will check whether there's any save data to be recovered. In the case there is, that previously mentioned struct will be accessed and that information will be extracted and passed onto the objects that saved it in the first place.

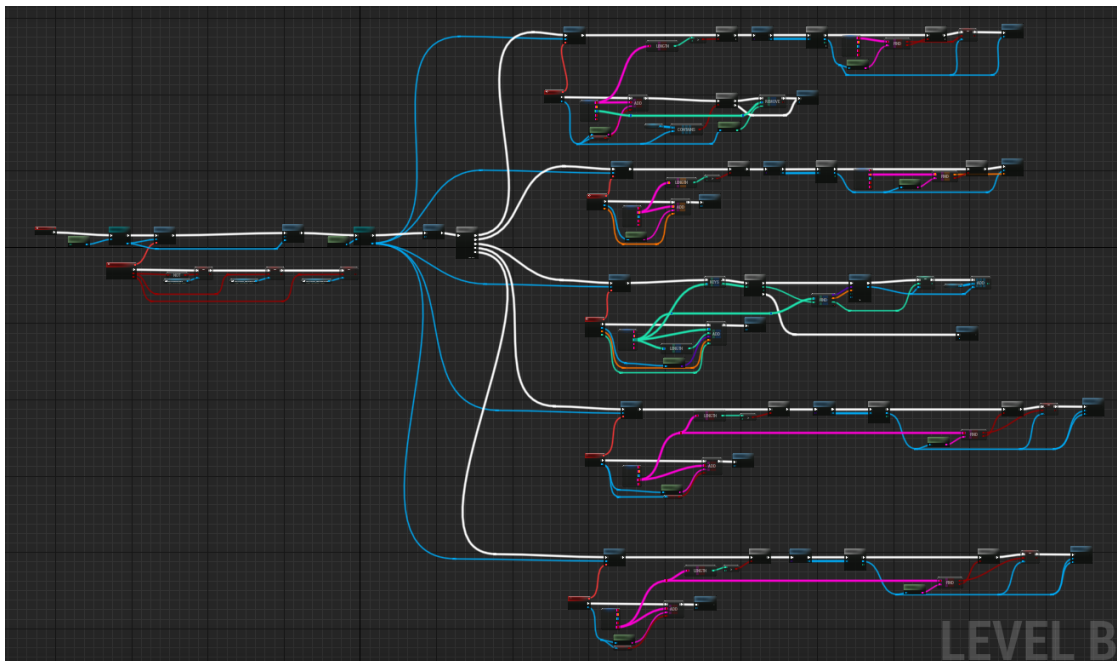


Figure 4.12: Level blueprint saving system

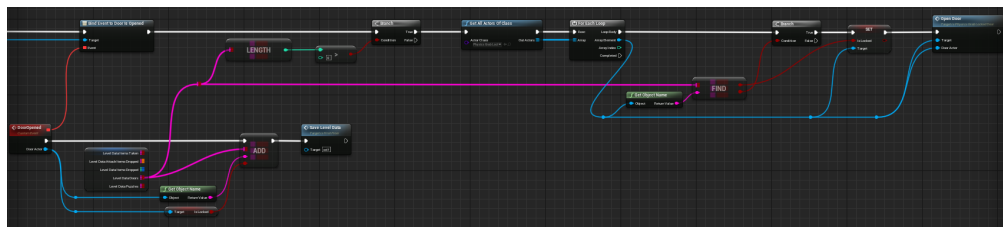


Figure 4.13: Branch in the level that saves doors that have been unlocked

4.5 Results

To conclude, thanks to a use of an ambiguous design methodology, all design objectives have been able to be implemented, creating one conjoined emergent design. To begin with, the level design has been successfully implemented within two instances of the same level, capable of telling a story purely through the environment. Aside from that, an effective object placement combined with the level design has been a key component in integrating a proper game flow that offers a certain degree of controlled freedom of choice to the player without overbearing them, thus achieving the objective intended for the gameplay design. Finally, thanks to the combine effort of level, gameplay and a carefully constructed story, an emergent narrative that fulfills the narrative design objective is attained. This emergent narrative permeates not only purely narrative elements like dialogues and written notes, but also all the other systems in place thus reaching a cohesive narrative design.

- **Google Drive Repository:** <https://drive.google.com/drive/folders/1uGAJPB-h4o4YR30lNILBVsVIFn3Dpnc-?usp=sharing>

CONCLUSIONS AND FUTURE WORK

Contents

5.1	Conclusions	41
5.2	Future work	42

5.1 Conclusions

As a personal note, design is and has always been my end goal as a game developer. I have been able to act as one up to some degree in some of the projects developed in the degree, however this was the first big project where I could pour my entire design philosophy with my own personal views influencing all other aspects of the game. This is not to say that working alone was an advantage, since having a team would have been a great addition in order to create a better product (i.e. an artist to create custom assets or make personalized music, a programmer that could better implement my design and therefore leave me more time to design and prototype more systems, etc). That being said, this work has helped me severely develop my design skills and has made me ready to embark on more ambitious and professional endeavors.

I would also like to add that this was my first project developed using a new engine (Unreal Engine). I had high expectations and it has managed to surpass them. Designing in this engine has been a great experience and I'm happy to say that it has reawakened a drive to keep creating experiences.

5.2 Future work

To preface things, I believe that the most crucial aspects of development have been successfully met, however a few additions that were not definitive but still included as possibilities (such as more small self-contained scenarios, more puzzles and an enhance environmental design with custom assets) have not been implemented. This were not systems expected to be part of the game, they were more of a backup plan in case the design objectives hadn't been met or the work had been completed in lesser time than expected. That being said this are things that I would like and would probably add before releasing the game in order to achieve a more professional final product. This will not be great expansions as I would like to star developing new ideas with my newfound knowledge, but they would probably be done to consider this a fully realized project.

BIBLIOGRAPHY

- [1] Destructoid Noelle Warner. The art of make believe: emergent narrative in games. destructoid.com/emergent-narrative-video-games-art-of-make-believe.
- [2] FromSoftware. Elden ring, 2022.
- [3] Firaxis Games. Xcom 2, 2016.
- [4] Fullbright. Gone home, 2013.
- [5] Giant Sparrow Games. Edith finch: House and common areas. unrealengine.com/marketplace/en-US/product/ef-house.
- [6] 1D.STUDIO. Triplex house villa. unrealengine.com/marketplace/en-US/product/big-triplex-house-villa.
- [7] Quixel. Megascans. quixel.com/megascans/home.
- [8] Aidan Perry. Unreal engine: Ultimate survival horror course. udemy.com/course/unreal-engine-ultimate-survival-horror-course.
- [9] Kiwi Remoto. Average salary of an unreal engine developer. kiwiremoto.com/sueldo/desarrollador-unreal-engine/.
- [10] Cinco Días. Monthly net income of ue developer. cincodias.elpais.com/herramientas/calculadora-sueldo-neto/s.
- [11] Game Rant. All games confirmed to be using unreal engine 5 so far. .
- [12] gamesradar Josh West. The matrix awakens is a truly unreal glimpse at the future of video games. .
- [13] Eurogamer Richard Leadbetter. Unrecord's 'viral' trailer is just a taste of the ue5 surprises to come. .
- [14] Víctor Manuel Navarro Remesal. Libertad dirigida. una gramática del análisis y diseño de videojuegos. .

