UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

KAIJUS AS ENVIRONMENTS:

DESIGN & PRODUCTION OF A COLOSSAL MONSTER FUNCTIONING AS A BOSS LEVEL

Guillem Álava Burgueño

Josep Serrano Recuero

Video Game Design & Development

Course: 2022/2023

Universitat Politècnica de Cataluña, CITM

Thanks

To my family and friends for believing in my capabilities and achievements. Their moral support and ideas made this project as unique as it is to me.

To my tutor, Josep Serrano Recuero, for guiding me in the right direction and giving me the necessary tools and resources throughout the development. This project would not have been possible without his help.

To Joan Pons and the class of Level Design in Barcelona for taking an interest in my project and providing helpful feedback that perfected the design of the final result.

To everyone who participated in playtesting the game, which made this game a collaborative effort.

And finally, to my partner Sergi Colomer for being with me throughout the journey, supporting me until the end and believing in me unconditionally. I would not be where I am right now without you.

Index

Abstract

Boss fights are a staple in most video game genres. They are milestones in the adventure, designed and intended to test the skills that the player has acquired throughout their adventure. In some cases, they even define the whole experience of the game, especially one type of enemy that has appeared in several instances and every genre: colossal bosses, monsters of giant proportions usually used as a matter of spectacle and a simple yet effective way to showcase the sheer power that players have achieved up until that point in the adventure. Titles like God of War, Shadow of the Colossus and even many Super Mario titles use this concept in their video games in imaginative ways to create Kaiju-like creatures working as a living environment the player has to traverse to defeat them. However, what is the process behind creating a colossal boss that works as a breathing environment, and how can it be achieved?

This project aims to study the process of colossal boss creation and design and apply level design and asset creation. To do this, the author will investigate the main aspects and key-defining features of these bosses, analyzing the strengths and weaknesses of existing bosses in videogames such as God of War 3's Cronos and Shadow of the Colossus and Solar Ash's bosses in terms of art production and game design. From this study and following the art process for creating creatures in the video game industry, the author will conceptualize, design and produce a working, playable prototype of a boss fight, showcased in the final presentation.

Keywords

3D, Art, Bosses, Kaiju, Modelling, Sculpting, Texturing, Rigging, Animation, Game Design, Unity, Shaders

Links

Trailer video

https://www.youtube.com/watch?v=HspBNoqm8kw

Final Product

https://github.com/KiwiTrek/Kaiju_Environment_TFG_Project/releases/download/Gold/Alava_Guillem_Kaijus_As_Environments_Gold.zip

Figure Index

Table Index

1. Introduction

In most cases, for a game to be fun, they have to offer obstacles for players to overcome with the tools and capabilities available. One of the major milestones in most video games designed and intended to test the skills that the player has acquired during a specific section of the narrative is bosses. From the formidable dragon at the end of a dungeon to a mighty foe who has achieved God-like powers, they have become iconic elements in most players' adventures.

They can come in varied shapes and forms, offering specific experiences for different chapters in the narrative. However, this project focuses on one type of boss in particular: colossal bosses, monsters of giant proportions usually used as a matter of spectacle and a simple yet effective way to showcase the sheer power players have achieved until that point in the adventure. These types of encounters against monsters have their origin in ancient mythology. Some of the most memorable battles in general culture are David against Goliath and St. Jordi against the mighty dragon. All these fights share the premise of a hero defeating a beast of enormous proportions against all the odds. In media, they have inspired the creation of the Kaiju[1] genre, a Japanese film genre which involves fights between giant monsters of the same name, usually in urban environments.

In the case of video games, colossal bosses serve as a clear candidate for a challenge to be overcome by players, at times becoming the most memorable events in them. Sometimes, they are used as a glorified playable cutscene to demonstrate a studio's technical and artistic capabilities, which is the case of Cronos from the video game God Of War III (more information later). However, developers in the past have designed memorable experiences involving direct interaction with the player, from being a hazard in the environment to the level itself, which are worthy of study.

The main objective of this study is to analyse what makes an excellent colossal boss fight and how to make one, focusing mainly on the creation and design of monsters that are also levels that must be navigated onto themselves. The main focus of this project is the artistic perspective, but the author also mentions topics from a design and technical point of view, which entails investigating existing examples of bosses from different genres, all compiled in a single document.

With the acquired knowledge, the student has created a working prototype of a video game boss level, showcased in this thesis's presentation. In this prototype, a player must defeat a formidable foe of gigantic magnitudes in the middle of some ruins by traversing and climbing its body towards the top, where the true enemy behind the chaos awaits.

1.1. Motivation

As of 2022, the video game industry has become one if not the most profitable entertainment media in the world, amassing almost 200 billion dollars in game-related sales alone, half of which are from mobile games (Wijman, 2023). Games have become crucial to many people's lives, including mine, whether used as a service,

[1] Kaiju: From Japanese's 怪獣 (*kaijū*), "strange beast".

a valuable tool for healthcare and education or an artistic gateway. As such, in the situation that video games are in, there is a need to study and analyse them.

Many scientific papers investigate both video games' beneficial and detrimental effects on children and patients suffering from mental illnesses. However, games as a focus of study are less common and, therefore, more valuable — a quick search through scientific research portals is enough to prove this fact. Studies tackling the many theories of fun, the essence of games in general, and elements inside them, such as level design, are some of the most cited works in papers. However, one surprisingly underdeveloped concept is that of game enemies, villains and bosses. My research uses such studies for its theoretical framework, but I still believe there is much to explore and investigate.

Hence, this project's primary motivation is to contribute meaningfully to the research of video game bosses, offering new perspectives to the discussion.

I also want to develop my artistic portfolio and demonstrate and put to the test my artistic skills while learning and developing them. This project will require many skills related to video game development, specifically design and programming, which I have recorded and appropriately explained throughout this study. However, despite their influence in defining the final product, these other aspects are not the focus of this project — as mentioned before. Therefore, I will use external sources such as tutorials or assets made by others to build the prototype.

As an aspiring 3D artist, I mainly want to tackle the artistic point of view by conceptualizing, designing, creating and documenting the process behind it all. I would consider this project a personal challenge, a test of all my accumulated knowledge and a synthesis of the prowess of what I can achieve at these moments of my professional life.

1.2. Problem Formulation

Many video game companies have achieved what this project proposes to do. Titles like God of War, Shadow of the Colossus or even many Super Mario titles use this concept in their video games in imaginative ways to create Kaiju-like creatures working as a living environment the player has to traverse to defeat them. The first of these titles has even done a talk at GDC, recording with excruciating detail all the hardships and methodologies used to create such a boss (O'Neill & Velazquez, 2012). Nevertheless, many of these projects are efforts made by hundreds of people throughout extended periods of time.

The main problem the writer wants to answer is how can a single person create an experience worth the quality of a professional studio in a relatively short time and to which extent they can do it.

To solve this problem, the student will study the necessary processes each of the sectors of a video game studio have to take to produce the elements needed for a boss of this calibre, focusing on the artistic aspects of the boss itself: what steps are needed, how can they be done and what are the results obtained by the student.

1.3. General Objectives

In general terms, the aim of this project can be divided into the following points:

- Analyse existing colossal bosses in video games.
- Define the main aspects of what makes a memorable gigantic boss.
- Understand the mistakes behind the least popular examples of some enemies and bosses.
- Study the process of boss design.
- Implement level design concepts in the boss' navigation phase.
- Research the pipelines used for boss creation in the industry and why they use them.
- Investigate what documentation and tools are required and why.
- Define a visual language of colour palettes, shapes, ideas and modular parts of a boss by creating concept art.
- Build the volumes of a boss and its different parts using primitives in a blockout.
- Sculpt the monster's elaborated shapes, crevices and other details of all the elements that make it up.
- Reduce the polygons and develop optimized versions of all the sculptures created.
- Distribute the polygon's texture information in UV Maps and paint them.
- Build a rig and animate all the boss's attacks, phases and stages.
- Produce a working prototype of the aforementioned boss' concept and design.

1.4. Specific Objectives

From these objectives, the intrinsic value of the project will be enhanced by the following specific objectives:

- Follow a professional creative process.
- Challenge players with an entertaining experience.
- Test the student with a creative process of significant magnitude to be showcased in his portfolio.

1.5. Project Reach

As it is artistically focused, this project aims primarily at 3D artists eager to learn about the experiences of other like-minded developers and how they can apply elements from other parts of development to their work. Likewise, since this production tackles other aspects of boss development, the public can also be from other sectors, such as riggers, animators, concept artists and game designers.

As a direct public, they can be either students from around the author's age — that is, between 16 and 25 old —, single developer studios that want to investigate the extent of producing this kind of project with limited resources and time, or other researchers like the student who are looking for sources of information regarding boss development and design, precisely the type described in this study.

As an indirect public, the project can be attractive to video game enjoyers and other individuals curious about the world of game development and the process behind it, especially those who have enjoyed titles like Shadow of the Colossus and Solar Ash, some of the games analyzed in this study.

2. State of the Art

2.1. Definition & History: From dragons to cosmic threats

Initially, the basic concept of a video game boss can be defined as a significant and challenging enemy that blocks a player's progression and acts as the climax/ending to the game's environment, level, or world (Rogers, 2014). In other words, it is a significant CPU enemy that the player must defeat in an epic confrontation that serves as a microcosm of the game while temporarily breaking its rules to achieve a goal or continue progress (Keren, 2018). As such, bosses aim to bring unforgettable experiences to video games, marking an impasse between game stages or before its ending.

Bosses as a concept did not come from thin air. As such, several years of workarounds and technological advancements had to pass to start seeing monsters with a considerable scale compared to the player. To better understand the evolution of this concept through time, the author has comprised a timeline of bosses based on size: how did they fit inside the screen or the game itself, what were the design philosophies behind them and which were the biggest bosses at the time.

The origin of the first boss fight is in the videogame in 1975's Dnd for the PLATO system, a game based on the proclaimed tabletop game Dungeons & Dragons in which players had to find and defeat a dragon guarding an ancient orb at the end of the adventure. There is no concrete evidence regarding the exact origin of the word "boss" to refer to these encounters with formidable foes. As Mike Fahey puts it in the podcast "Splitscreen", the term goes way back to criminal and mafia gangs that have a leader whom they call boss, appearing in media in movies such as Bruce Lee's "The Big Boss", in which the actor mentioned above has to fight gang members to get to the boss of the criminal cartel (Segarra, Gach, & Fahey, 2021). However, during the 80s, arcade and console games started expanding by adding these types of challenges in their video games, which game manuals and other official documentation referred to as bosses. The term boss is coined to two possible origins: either in the Metroid Instruction Booklet to refer to the "Mini-Boss" Ridley (Figure 1) and another in the game "Galaga" with the official name of its titular boss being "Boss Galaga".

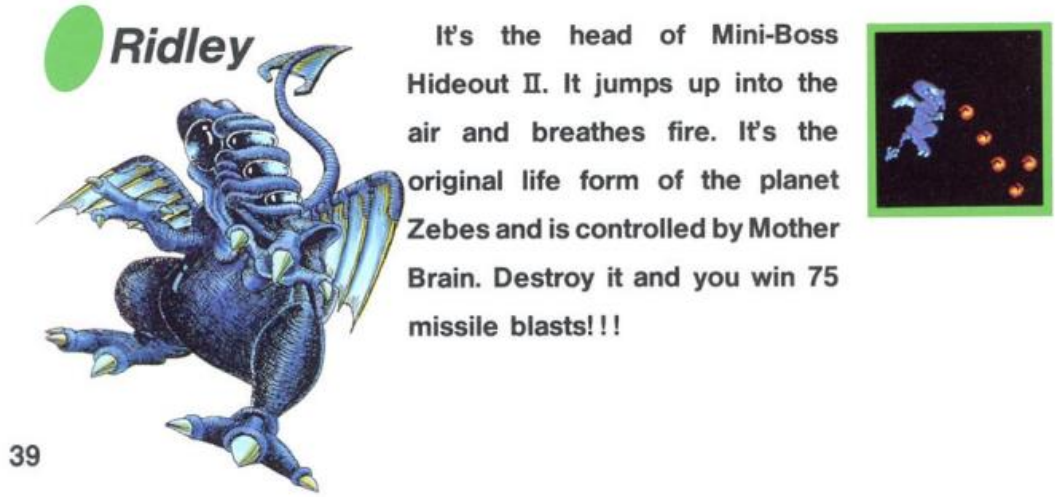

FIGURE 1. RIDLEY'S DESCRIPTION, FOUND IN METROID'S INSTRUCTION BOOKLET

As such, the term and concept of bosses organically grew in the video game industry, becoming a staple in most of the games at the time. The challenge offered in Dnd originally started as a monster of similar size to the player; however, designers often used the space the player could see on screen to showcase the amount of power and destruction these Goliath-sized creatures were able to inflict. A common mantra in design by Louis Sullivan states that "Form follows function", so developers used this concept to its maximum potential. The first of these examples appears in 1980's Phoenix by Amstar Electronics, presenting a boss in the form of Mothership, a giant alien UFO occupying most of the screen, as its own challenge separate from the rest of the game (Lee, 2015).

This boss also introduces a recurrent concept in these monsters: the weak spot, a vulnerable point located somewhere in a creature that, when struck, acts as a vital killing blow. Much like bosses themselves, this concept comes from mythology, in the form of the Cyclops' eye or Achilles' heel; and in literature, like Smaug's missing scale in The Lord of the Rings. It is a recurrent element seen in colossal bosses,

FIGURE 2 (LEFT). MOTHERSHIP, *PHOENIX*

FIGURE 3 (RIGHT). BIG EYE, *GRADIUS*

Despite its many achievements, there were also several constraints at the time. The camera had to be focused on the character at all times, and not many sprites[2] could be loaded simultaneously, so game elements were constrained to the borders of what players could see and reach. If a game attempted to convey gigantic monsters by current standards, it opted to divide each of its parts into separate screens or camera positions, frequently focusing the attention on defeating one of these parts at a time, but screens could never hold the entirety of the boss in a single image. If elements acted outside of view, players saw it as a cheap and unfair method to increase difficulty. Some examples of this phenomenon can be found in many games from the "Shoot 'em Up" genres, such as Phoenix, 1985's Gradius by Konami and 1987's R-Type by Irem.

The limitations in early 2D games disappeared with the arrival of 3D video games. With a new dimension of depth to explore, new possibilities arose as now, with different camera angles and perspectives at the designer's

[2] Sprite: A two-dimensional bitmap graphic object integrated in 2D games

disposal, bosses could appear more prominent than ever before. The first instance of games containing bosses above the 100 ft tall mark height can be found in 2001's Serious Sam: The First Encounter by Croteam, with its final boss, Ugh Zhan III, standing at 100.5 m. according to official sources[3]; 2003's Drakengard 3 by Access Games, with its final boss, The Flower, being the size of an entire city; and 2004's Painkiller by game studio People Can Fly, which makes titan-sized creatures as its default type of boss players have to fight.

FIGURE 4. NECROGIANT, ONE OF THE FIRST COLOSSAL BOSSES, *PAINKILLER*

The main characteristic of these bosses is that players had to fight them at a distance and worked as spectacular and glorified targets to be shot at in the case of first-person shooters, for example. Thus, at best, these enemies' challenges are reduced to slow, telegraphed attacks, projectiles or minions. Hence, after their first production, Team Ico decided to develop a game in which colossi had to be destroyed not by attacking from a distance but rather by having to figure out how to climb and navigate their bodies to damage their weak spots. Released in 2005 by the name Shadow of the Colossus, this game would become a massive success, offering a new take on boss fights never seen before in a videogame[4].

FIGURE 5. SCREENSHOT OF SHADOW OF THE COLOSSUS IN ITS 2006 VERSION

[3] See the boss' description in the HD version of the game downloadable in https://store.steampowered.com/app/41000/Serious_Sam_HD_The_First_Encounter
[4] More information in point 2.6.1.3

From this point forward, the only way was to go bigger and better. The concept of a boss as a navigable environment became an idea bound to happen eventually. It was not until 2006's Gears of War 2 that this concept would be developed. In this game, players have to fight a giant worm named the Riftworm by entering its insides and killing it from the inside as part of chapter 6 in act 2. However, it is dubious to consider this level a boss fight, as it does not follow the usual structure associated with boss fights.

FIGURE 6. PART OF INTESTINAL FORTITUDE; *GEARS OF WAR 2*

Nowadays, the honour for the most gargantuan examples of bosses in videogame history would fall to 2012's Asura's Wrath, known for having absurdly enormous bosses the size of planets. The most immense of them all is its final boss, Chakravartin, the deity of the game who, in one of their phases, can hold entire galaxies in its hands and throw stars to the player. It will be almost impossible to create something of equal size to the conceivable universe.

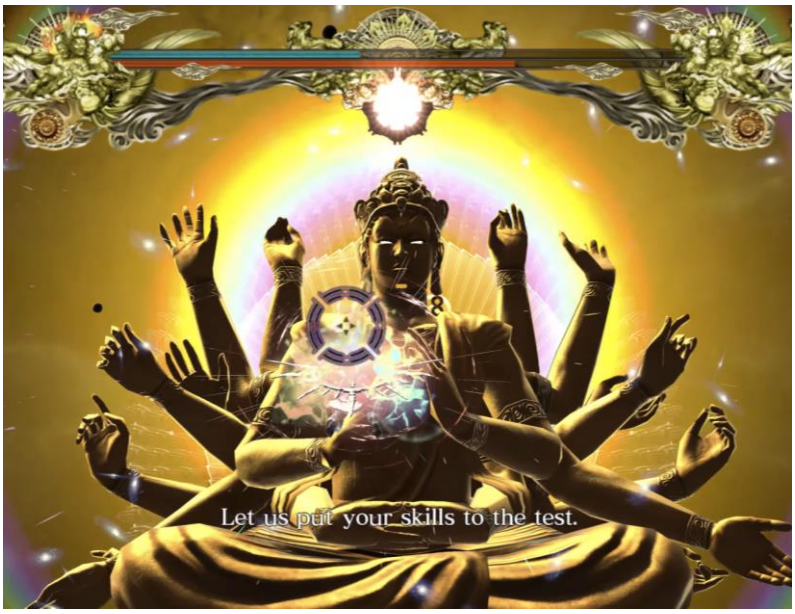

FIGURE 7. CHAKRAVARTIN, IN ITS COSMICAL-SIZED FORM, *ASURA'S WRATH*

2.2. Bosses Design: The anatomy of a good fight

From a design standpoint, most games aim to reach a goal by sorting and overcoming obstacles using the tools at the player's disposal. Thus, bosses are usually the most prominent obstacle in video games. In the best cases, they are memorable experiences that stay in the collective mind of players; in the worst, however, they become annoyances in the way.

The primary purpose of bosses is to dazzle the player with an incredible playing experience or confrontation, which in turn creates a lasting first impression in them; a way of engaging the player in the story by using the battle to advance it and create an emotional attachment to a significant event; and a challenge to test acquired skills or introduce new ones in preparation for what is next (Keren, 2018). In other words, bosses have three general roles: they are a change of pace mechanically, breaking the main game loop given by the game and its levels; they are a narrative device, usually being connected to the game's story and affecting its actions directly and meaningfully in the best of cases; and they are a test of skill, sort of like an exam that makes players learn and test the skills they have learned throughout their journey (Design Doc, 2019). With this objective in mind, there are several elements and rules to consider that make up the boss' structure and understand the proper execution of a standard fight, which the author has divided into four points:

2.2.1. The Core: Attributes, abilities & archetypes

As previously mentioned, an essential rule in the design involved in boss design is that "Form follows functions". The attributes and abilities that define the boss must define and convey the purpose and mechanics intended by the designer; hence, a correct definition of these can create dynamic experiences for the player.

Firstly, bosses have to be robust and intimidating. They are tests of skill, guarding the rest of the game for players, so they have to see that they are a menace and a threat that contrasts with other enemies. Based on the design mantra mentioned above, a typical take to showcase power is what this entire thesis is about: increase the physical attributes of the creature compared to other monsters, usually in size, health and force. This way, they change the game's rules and take the player by surprise. However, that does not mean they should only be bigger regular enemies. Otherwise, they become nothing more than giant walls players have to beat down, or, in other words, a damage sponge, which they may feel as a false sense of difficulty.

FIGURE 8. GONGEN WYZEN, LOOMING OVER EARTH, *ASURA'S WRATH*

What makes them feel different from regular minions is that they play with an element of surprise. Usually, game bosses use new abilities, often creating unique mechanics for the fight. How they challenge the player mentally and make them think outside the box or how they look and operate can throw the player out of the loop. Usually, this comes down to defining how they fight and respond to the player and what themes and objectives they want to tackle or subvert.

FIGURE 9. GIGA DRY BOWSER, BEGINNING A FIRE ATTACK, *NEW SUPER MARIO BROS 2*

A good boss fight balances fairness and difficulty in a constant state of total immersion defined by the term "flow" (Csikszentmihályi, 1990). Game designers must think of exciting challenges and respect the player's skills to the point of the story in which the fight happens. In defeat, players should not feel frustrated because a fight seemed unfair, but instead, they should feel rewarded with the knowledge of how to overcome the obstacle on the next try. Thus, designers have to provide a level of clarity to the player for each attack from the monster. There has to be anticipation and, if deemed necessary, telegraphing of attacks with the usage of preemptive animations indicating the start of an attack, shadows that indicate the landing position of an attack, repeating patterns that the player can memorize and use to sort the waves of attacks from a boss and clear indications that the boss is vulnerable and therefore safe to approach and attack.

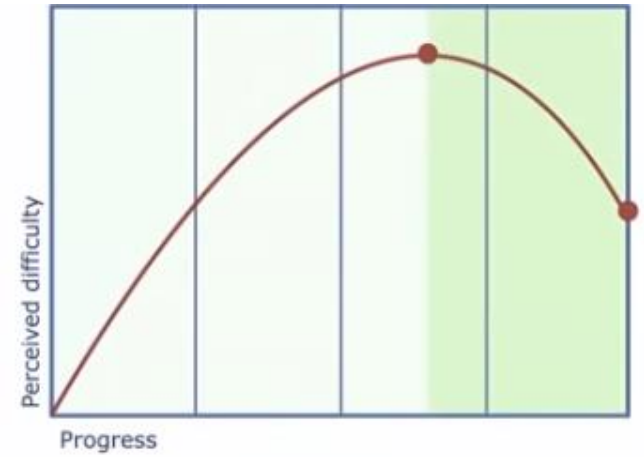

FIGURE 10. GRAPH SHOWCASING THE IDEAL RELATIONSHIP BETWEEN DIFFICULTY AND THE DURATION AND PROGRESS OF AN ATTACK OR STATE OF A BOSS

To find exciting challenges to put to a player, a good starting point for creating a boss is by beginning with an archetype and twisting it to catch players by surprise. In his GDC talk, Luis Barriga mentions nine standards for behaviours commonly found across many games in history (Barriga, 2003). These include the rusher, who dashes towards the player and has to be dodged; the jumper, who jumps into certain spots and deals damage upon landing; the spawner, who can summon minions to attack; and the bomber, a flying boss that drops harmful entities like bombs.

FIGURE 11. MORTON KOOPA JR., AN EXAMPLE OF A DASHER AND A SUMMONER; *NEW SUPER MARIO BROS 2*

Finally, another critical factor in making memorable boss fights is the theme, aesthetics and subversions behind them. Creating a contrast between slower-paced, average levels and the freneticism of the boss fight can help convey the difficulty of overcoming this challenge. Thus, visual and sound design are essential to get right (Millard, 2018). It can also help convey clues that will help beat the boss, using patterns to the designers' advantage to teach players how to tackle a situation subtly. In the best cases, they can indicate different methods of defeating a boss, giving the player a sense of freedom and creative liberty that makes each experience personalized and remarkable in the game's experience.

FIGURE 12. INTI, A BOSS FROM ICONOCLASTS WHOSE SYMMETRY IS A CLUE TO DEFEATING THEM

2.2.2. The Structure: Phases and pacing

After many iterations, the game industry's consensus regarding how a boss should be structured is that a boss fight is divided into three stations or eight beats, depending on the approach. It is important to carefully respect and understand these steps, as they divide and balance the rhythm of the good battle.

The first phase of a good fight is the introduction, divided into two beats. The first one happens before the fight even begins. There has to be a build-up and anticipation towards it, which in turn helps generate engagement for the fight. The most common way to do this comes in environmental storytelling, showcasing the destruction the boss has created alongside the path, or a cutscene, in which the plot advances and the boss is presented. However, anticipation for a boss fight can be done way before in the overarching narrative. The most memorable and scary bosses are recurrent characters throughout the story, like a long-lost rival appearing from the dead or a recurrent reminder of impending doom.

An excellent example is Jetstream Sam, a boss from Metal Gear Rising: Revengeance, who players fight twice throughout the game — once at the beginning and the 2nd at the end. The anticipation for the latter starts way back at the start of the story, in an unwinnable fight that shows how weak the player is compared to this man who can deflect and parry all their attacks. After having destroyed a behemoth minutes prior, being humiliated by a character with a cocky attitude creates enragement in the player, making the player want to defeat him even more, thus engaging them with the story.

FIGURE 13. JETSTREAM SAM IS ABOUT TO DUEL RAIDEN FOR THE FIRST TIME, *METAL GEAR RISING: REVENGEANCE*

The battle can begin when the player engages with the boss and arrives at the arena. The battle is divided into different phases, the first of which is the reveal, the second beat of the first station. From his experiences developing the first boss of his game Philophobia, game designer Tim Ruswick mentions that all good boss battles need something that triggers the start of the encounter. Before starting, players should get a feel of the surroundings, even if they do not understand at first glance what elements that make them up will mean for the fight. This exploration also builds anticipation, culminating in the above mentioned introduction, an animation or initial cutscene that showcases how menacing they can be by making them roar at the player or appear provocatively or menacingly threatening (Ruswick, 2017).

FIGURE 14. MALENA, BLADE OF MIQUELLA, INTRODUCES HERSELF TO THE PLAYER, *ELDEN RING*

After this, the proper battle begins, with the basic patterns of attacks and defences stipulating the experience's third beat. The sequence of a boss fight alternates between different patterns or states, usually in the form of an offensive pattern, the main form of attacking in which the boss tests the player's defensive skills, and a defensive pattern, in which the boss becomes invulnerable to attack the player in a frenzy relentlessly. In between these patterns, there are windows of opportunity in which players can attack bosses when either they do a taunt or another animation that characterizes the boss or after an offensive pattern that gives the player a clue that can make the boss enter its vulnerable state, which usually involves a recently taught skill or obtained tool. How these phases are intercalated and how predictable they determine the difficulty of a boss fight.

FIGURE 15. GOHMA, AFTER BEING STUNNED BY LINK, *THE LEGEND OF ZELDA OCARINA OF TIME*

When the boss reaches certain thresholds, often determined by the amount of health they have left, the battle has to evolve and increase in difficulty qualitatively by changing patterns and elements of the fight and quantitatively by reducing windows of opportunity or increasing wave count of minions and attacks. Such changes define the following beats of the battle, from escalating gradually until the climactic moment in which everything is on fire". The key to keeping a fight refreshing and entertaining is knowing how to balance the battle's phases and how much they last. There must be a clear indicator that signifies that a boss has changed phases, which comes in the form of another cutscene, animation or change in the opponent's appearance, indicating a sense of progress and success in the player.

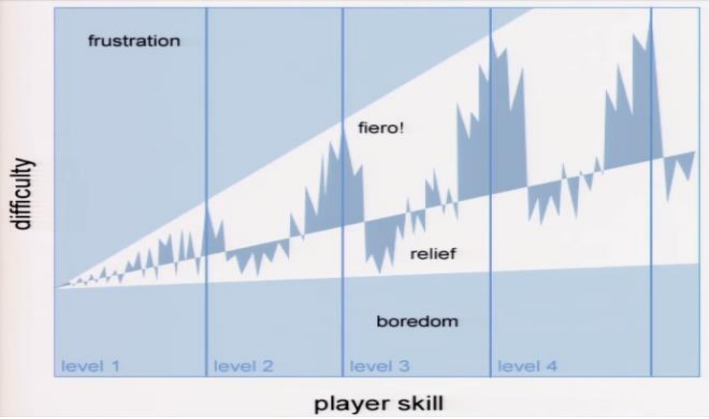

FIGURE 16. GRAPH SHOWCASING THE IDEAL RELATIONSHIP BETWEEN PROGRESS, SKILL AND DIFFICULTY OF A BOSS FIGHT AND THE SENSATIONS THEY PRODUCE IN A PLAYER

The final station is the outro to the fight, also divided into two beats. When the boss is stunned after throwing everything in their power towards the player, it is up to them to deliver the "kill". While it can be satisfying to kill a boss after their best attempts at destroying the player, sometimes it is even better to make them undergo the "Humiliation Conga", the process of ridiculizing the boss and making them suffer for what they have done (TV Tropes). After dealing with the last hit, a celebratory break showcasing how the boss falls on their knees or explodes into pieces has to let the player relish, gloat and feel relief after an intense battle, which also gives players their long-awaited reward and release in the form of an advancement of the story, an item or a new ability unlocked. Nicole Lazzaro refers to this type of sensation after a complex challenge as Fiero, which is more intense when the challenge is more demanding (Lazzaro, 2013).

2.2.3. The Arena: Safe spaces and environmental hazards

Bosses work best in an enclosed system and limited space that encourages confrontation encompassing the game's mechanics (Keren, 2018). The fight becomes meaningless if the player's fight or flight response allows them to leave or avoid the confrontation entirely by distancing themselves enough spatially from the battle. Hence, the arena definition is vital for the fight, as it can directly affect it positively and negatively.

FIGURE 17. GRIM MATCHSTICK, A BOSS FOUGHT ON SCROLLING CLOUDS, *CUPHEAD*

They have classically been cylindrical arenas or pathways containing all the battle's elements. These layouts allow for staging the camera in the middle or back of the room and then moving it around the area, always focusing on the boss (Rogers, 2014). The boss and all related elements and dangers must always be on the

screen so players can react accordingly and defend themselves against their opponent. Ruswick comments that the elements in the environment have a function and encourage interactability with the boss (Ruswick, 2017), serving as tools for the player that may help defeat the boss, such as falling statues or stalactites that can fall on top of the boss or hazards that serve as obstacles triggered by the trembling force of the boss. If the case is the latter, the obstacles must be taken into account as part of the difficulty of the boss, as the amount of information that the player can handle simultaneously is limited and may lead to frustrating levels of difficulty.

However, different rules apply when the bosses are the size of entire levels. In the following section, the author will describe how the size of an enormous boss changes the way of thinking about the space the player will occupy or see at a time.

2.2.4. Colossal Bosses: When big is too big

The specific case of a colossal boss is tricky to pull off but works in most genres and situations. They serve as a test of skill, as they must have skillsets that differ from regular enemies based on size alone, and they are threatening and menacing for the same reason. The mere presence of a creature of gargantuan size should already be menacing enough for anybody outside and inside the game, and they usually indicate a level of difficulty greater than the average because size indicates power. They are great tools for designers, but creating and designing a colossal boss fight has some essential aspects they must consider beforehand.

Firstly, a boss designer has to decide if the mechanics of their game are compatible with handling bosses like this. Games with melee-based mechanics, such as those from the fighting genre, have trouble making fair fights because these games involve a symmetrical confrontation, a mechanic that colossal bosses do not allow. If developers must include a boss that will not work for their game, the best solution for them is to make these fights glorified cutscenes with minimal interaction (only quick-timed events appearing from time to time). The most significant issue these bosses pose is the positioning of the camera: it is hard to keep the entirety of the boss inside of the screen — as seen throughout the history of big bosses. Bad camera handling causes battles to become dull or, even worse, break the immersion and flow entirely. There are many ways to solve this problem, from creating custom systems that accommodate them to workarounds and intelligent solutions.

FIGURE 18. NOCTIS FIGHTING THE ADAMANTOISE WHILE ALSO CLIPPING THROUGH ITS BODY, *FINAL FANTASY XV*

The first of these options is to fight parts of the boss at a time, a method of handling a boss fight in which the player gets the general scope of the fight but abstracts it away by reducing the real threat to a smaller, more affordable size. Other options include fighting them at a distance, which, as seen before, is the case of most "shoot 'em up"s and makes them more manageable, or making the boss a force of nature that creates hazards to be sorted by the player before reaching a particular goal, acting as a challenge modifier for a certain level. A combination of all of these elements is what makes the most memorable fights.

FIGURE 19. MG RAY, THE FIRST BOSS OF METAL GEAR RISING: REVENGEANCE

In the author's opinion, the most intriguing option is when fights completely ignore this issue with the camera and zoom in as much as possible by making the boss a living environment. In this case, the navigation across a level and the combat against the boss become the same challenge, frequently having to apply level design principles to the boss' concept. The author further analyses these principles in the following section and dissects how games use these principles to their advantage and what their results are in the market study.

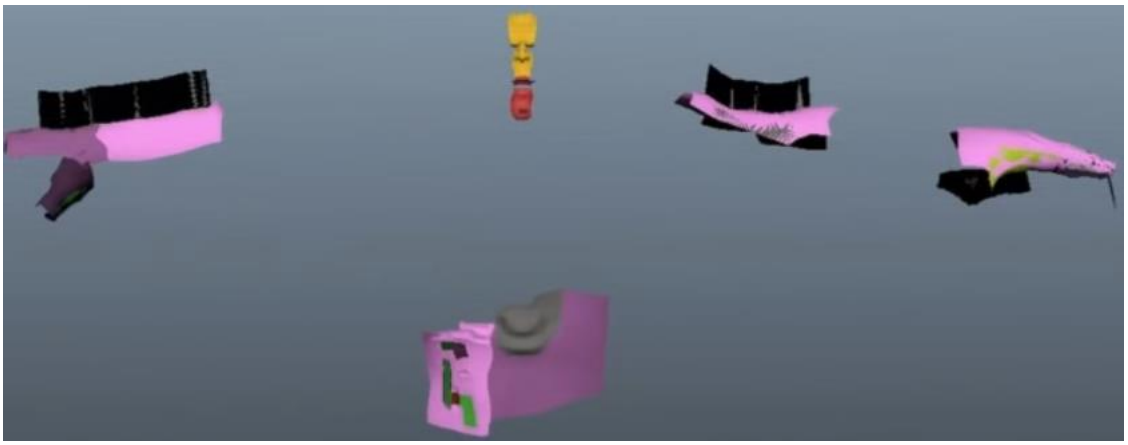

FIGURE 20. THE LAYOUT OF THE ACTUAL NAVIGABLE PARTS FROM CRONOS' BOSS FIGHT IN GOD OF WAR 3

2.3. Level Design: CCST theory applied to boss design

Like boss battles, levels are also divided by stations of increasing difficulty, oscillating between times of local maximum difficulty, in which a pivotal moment of action occurs, and relative safe zones, calming the player before tackling the following challenges. The methodologies designers follow to create levels have evolved and changed as they gain experience; however, the key to an entertaining level also involves balancing fairness and difficulty to keep a good sense of flow.

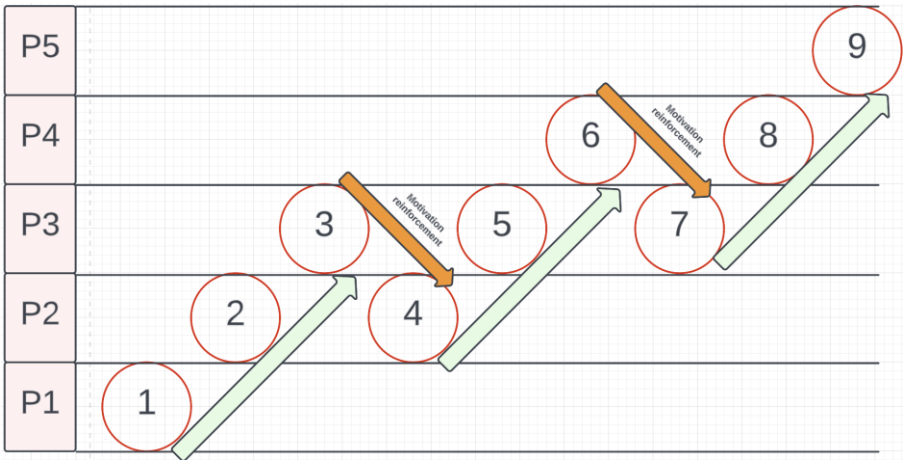

FIGURE 21. TYPICAL DIFFICULTY PROGRESSION GRAPHIC OF ACTION BUBBLES RELATING LEVELS(P) TO TIME

During the 2010s, Patrick Holleman began publishing a series of essays called "Reverse Design", which would later arrive in book form to stores. While playing the game, the designer noticed that "Super Mario World", a videogame created by Nintendo in 1990, follows a pattern that conceptually unites the levels by principles. Thus, in his third entry, "Reverse Design: Super Mario World", he presents a new thesis on level design theory that would revolutionize how game developers should look at level design, which he would call "the Super Mario World Method", "the Nintendo Method" or the CCST theory. He stated that each of "Super Mario World"'s levels could be deconstructed into interrelated elements across a dimension, usually a space consisting of a certain amount of units — in the case of Super Mario World, he considered the coin block as the smallest unit of distance measurement (Holleman, Measurements and Their Meanings, 2011c). The factors constituting CCST theory are challenges, cadences and skill themes, hence the acronym.

In this context, challenges refer to the different tasks players overcome across a specific level, from what enemies the player has to destroy to what obstacles in the level they have to surpass. Surrounding each challenge are some periods of relative safety for players to take a breather before tackling the next obstacle. Challenges also change across the level, continuously increasing in difficulty to avoid getting the player bored. Based on what variables are changing in the obstacles at play, the type of difficulty increases are divided into two types: expansions, which delimit quantitative changes (for example, in the number of hindrances, space between them, and the speed at which they move); and evolutions, which delimit qualitative changes (such as colour, behaviour, what weapon is the entity functioning as an obstacle using or their behaviour). They can also offer a triangular option, in which the player can choose their difficulty level by providing multiple and riskier paths with higher rewards. These paths are usually two, which, with the player, form a triangular shape in space, hence the name.

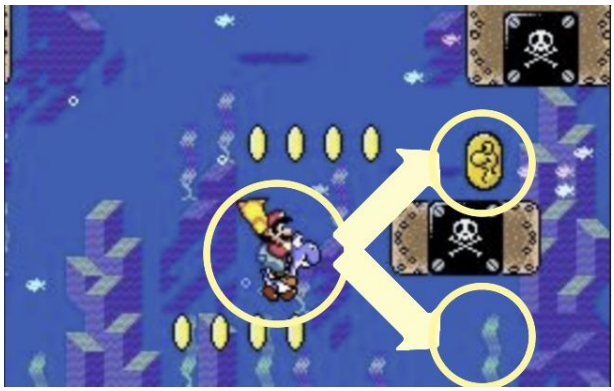

FIGURE 22. EXAMPLE OF TRIANGULARITY, WITH THE CORNERS OF THE TRIANGLE REPRESENTED, *SUPER MARIO WORLD*

Evolutions in a level can demand different methods and skills from the players to beat one another. These skill changes are what delimit the skill themes of a level, usually relying on changing game genres to keep the player on their toes. Varying game genres inside a level defines games as composite, allowing for more opportunities to make the player enter a state of flow.

Finally, how these skill themes and challenges relate to one another and are distributed across space defines a level's cadence. Defining a good level cadence can help level designers not to clutter their levels with unnecessary elements, overbearing the player. The cadence must start with a playground or setup section, in which the game showcases the central skill themes of the level in a low-risk environment (basically where it is difficult for the player to die). Based on these, evolutions and expansions of challenges ensue, considering the difficulty curve presented initially. These challenges culminate in a climatic act in which the player puts all the skills learned in the level to the maximum skill test, which usually ends the level.

FIGURE 23. EXAMPLE OF A LEVEL CADENCE IN "MINE CART CARNAGE" BY HOLLEMAN, P., *DONKEY KONG COUNTRY*[5]

These guidelines and methodologies come in handy when applied to boss design, considering each phase or possible movement the boss can do as challenges across time or, in the case of colossal traversable bosses, defining the navigable section itself as a level. It helps divide boss battles into different vital points, also known as action bubbles, allowing one to visualize and tweak aspects of an original design by conceptualizing its pacing and mechanics into a single flowchart.

[5] A more in depth analysis of this level can be found in Holleman, P.'s channel: https://www.youtube.com/watch?v=HC-oGDnhXZY&pp=ygURbWluZSBjYXJ0IGNhcm5hZ2U%3D

2.4. Boss Creation: The artistic pipeline

Several and varied processes exist to create game assets in the video game industry. Depending on the needs of the game, the artistic intention, the level of realism, stylization and other characteristics, how artists create 3D models may differ between productions, and thus their methods may change between studios.

When it comes to boss creation, much like any other 3D character, there is a generalized pipeline used in most triple-A companies that involves a process of nine steps, from drawing the idea on paper to making it a reality in a game engine, requiring a variety of skills, experts and sometimes even entire departments to bring new ideas to life. This pipeline is very similar to the one seen in the cinema, lining up with the 3D Production Pipeline seen in Andy Beane's book 3D Animation Essentials, in which he showcases the pipeline used in animation movies by studios like Pixar or Dreamworks (Beane, 2012).

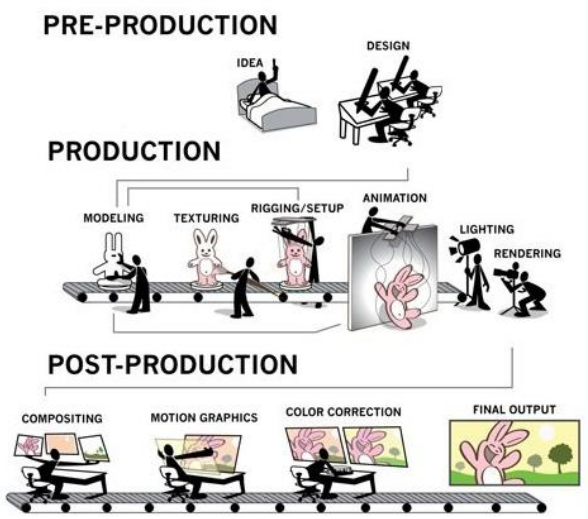

FIGURE 24. SIMPLIFIED VERSION OF THE 3D PRODUCTION PIPELINE

In the following section, the author applies this pipeline to video game creation, from conceptualising the idea to giving it life in its animation.

2.4.1. Pre-production: Ink, paper and machines

There are always many ideas behind any design that lead to it. Before starting, artists need to define who the entity in question is and what the story and reasoning behind the elements that make up the design are. Such definitions will be reflected in our character's physical, mental and emotional characteristics and developed in its model, textures and animations in a process summarized in "four Ds": discover, develop, design and deliver (Brighid, 2022).

In the discovering phase, artists come up with the inspiration to create their characters. A common mantra in design is that all ideas are made of other ideas (Berkun, 2020). As such, designers and artists do a brainstorming session where creators combine the best ideas and inspirations from the world and what interests them. Any media can be a good source of ideas, such as books, films, music or other video games. Usually, the more distant the source, the more interesting the ideas become. An excellent way to recollect these ideas is with

moodboards, which are image boards in which photos from other sources convey all the details and ideas related to the design. Moodboards may include facial and body structure, hair, clothing, shape language, stuff they may carry, characterization and mannerisms of the character.

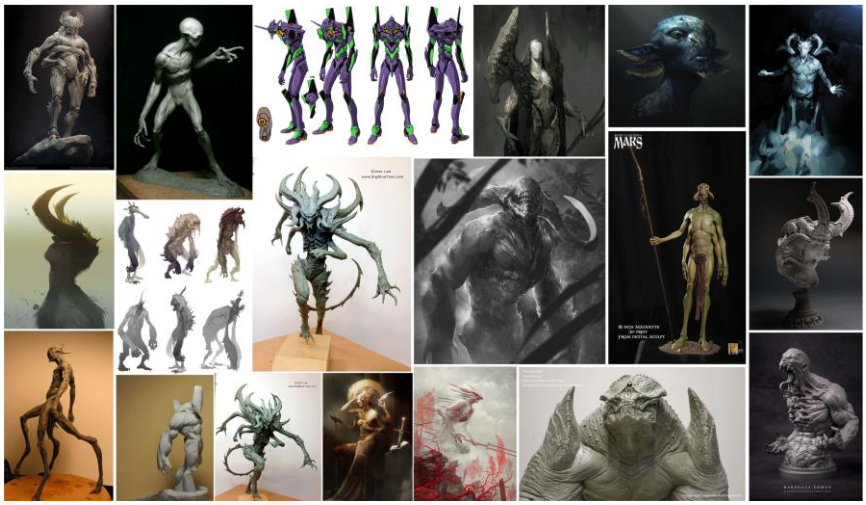

FIGURE 25. MOODBOARD FOR A MONSTER CHARACTER

By the time of this report, technology has also allowed creators to use artificial intelligence as an artistic tool for generating high-quality images in significantly less time than a person would, such as DALLE-2, MidJourney or Stable Diffusion. There are still a lot of ethical questions regarding the integrity and rights of artists and their art. As these algorithms require references from which to create images, many artists and experts in the industry consider AI the end of art and have expressed their disdain throughout social media. However, like many other tools, artificial intelligence is another tool future generations will use to create initial concepts and add more visual references during development. The author believes that image-generation tools will not disappear but become a starting point for artists to explore and create new ideas and concepts, as these machines will always need a basis from which to create, which they can only achieve with human input. After all, it is a human who has to be a "prompt engineer", someone who understands the AI models and can create good sentences to create the best images seen on social media.

FIGURE 26. EXAMPLES OF CONCEPT ART GENERATED BY ARTIFICIAL INTELLIGENCE

After discovering, there is a second phase for developing the idea that consists of exploring different character characteristics variations, which means iterating the same idea with slight changes or ideas. A common way to fast iterate is with thumbnailing, a process that involves drawing quick sketches in a limited space, usually with no more than 2 or 3 colours focusing on specific characteristics of the character at a time. From these quick ideas, creators choose some to further develop them in future iterations with increasing levels of detail.

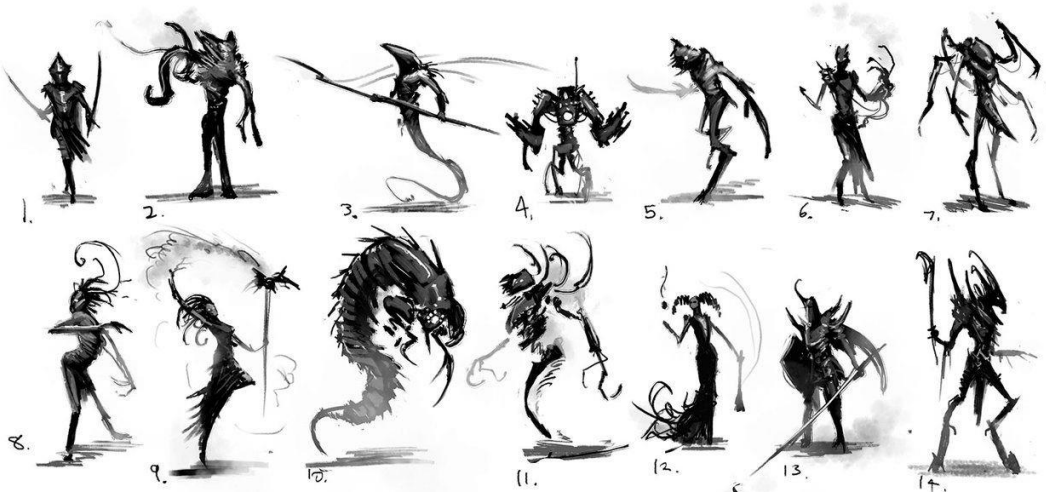

FIGURE 27. THUMBNAILING OF A CHARACTER

Finally, these ideas are appropriately designed and delivered after rounding up a few final sketches. In studios, this means creating a reference sheet or concept art, an expressive drawing of a character with turnarounds — showcasing the character from different angles, key poses, expressions and colour palettes. This way, 3D artists have a guideline from which they can start developing the character on 3D software.

FIGURE 28. FINAL CONCEPT ART OF A HUMANOID CREATURE

2.4.2. Blockout: Creating the foundation

After the creature is conceptualized, a blockout of it is done. This part of the process consists of defining the main volumes of the different parts of the body, starting from spheres and deforming them. Usually, it is as simple as recreating the character's silhouette with simple forms; however, depending on the level of realism the model has, more or less definition for parts like muscles or bone structure will have to be done to make models more anatomically accurate. These definitions will help to understand the size and shape language upon which the character will be built, so it is a crucial step to get it right.

FIGURE 29. BLOCKOUT OF A CHARACTER

In the case of the colossal boss, it is even more critical, as it will have to be paired with the blockout of the level design with which the player will have to navigate. For environmental and level design, there is also a definition of volumes of all the obstacles the player will have to sort through in a process called white boxing. Since the boss works as a level onto itself, the process will have to be adapted to conform to the main shapes of the creature while also being able to hold the level on its body, which means pairing the main character blockout with white boxing for the navigable parts.

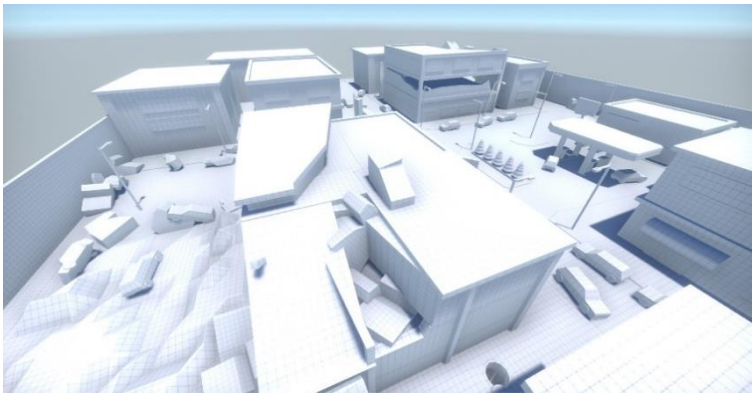

FIGURE 30. URBAN ENVIRONMENT BLOCKOUT

Additional modelling can be done to create more complex shapes like clothing, horns or other significant volumes from the character if needed. Any key aspects of the character that extrude and give form to their silhouette should be defined in the blockout.

2.4.3. Sculpting: Giving it detail

After the base is done, it is time to give them a more defined look. At the beginning of 3D digital modelling's history, the number of polygons a model could handle simultaneously in software could not exceed the thousands without sacrificing performance, limiting models' level of detail. However, with technological advancements, most software can handle millions of polygons at the same time on the screen, so there are programs prepared to work as if working with clay figures. This working methodology is known as digital sculpting, a process in which, with the help of different types of tools named brushes, artists properly define minor, more intricate details like the face, hair, scales, bandages, wrinkles, bones and spikes— frequently ending with what is called a high-poly model, a model with a polygon count surpassing the hundreds of thousands.

FIGURE 31. MODEL CREATED USING DIGITAL SCULPTING

If the model in question is more robotic or formed by hard surfaces, another process related to sculpting, called kitbashing, can be applied. This methodology comes from the world of mecha figurines and consists of creating high poly models using previously modelled modular parts like cubes, nuclei, tubes, wrenches and cogs. This process speeds up character creation in the case of robots and machinery while keeping consistency between the elements that form the machine. Some software has specific tools that can help instancing elements on top of the topology to add these modules procedurally, such as the Nanomesh tool in ZBrush.

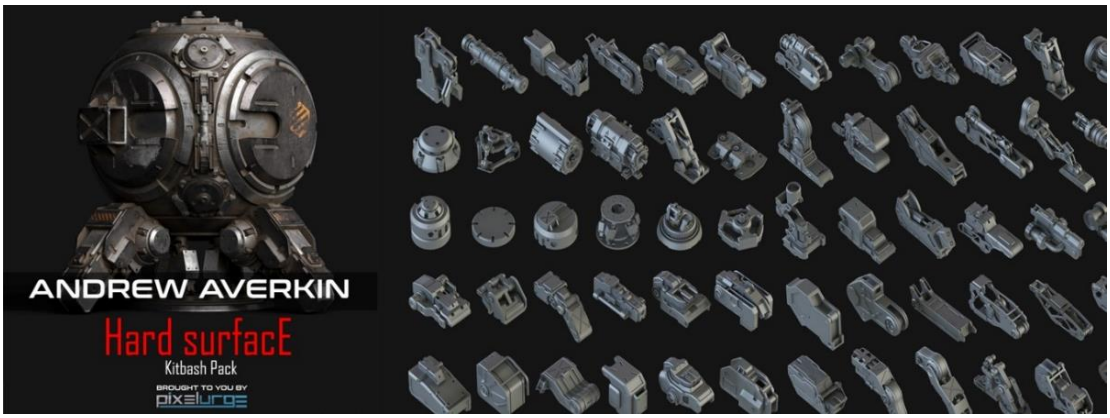

FIGURE 32. EXAMPLE OF KITBASH PACK AVAILABLE FOR MOST 3D SOFTWARE

2.4.4. Retopology: Optimization and loops

Since game engines cannot handle many assets with the number of polygons usually contained in a digital sculpture, there has to be a process of optimizing the model by creating a low-poly version, usually referred to as the mesh. This process is called retopology because it is the process of redoing the topology, that is, simplifying the number of polygons in such a manner that it still keeps most of the original model's silhouette and details while drastically reducing the number of polygons in it, thus creating what is called a low-poly model.

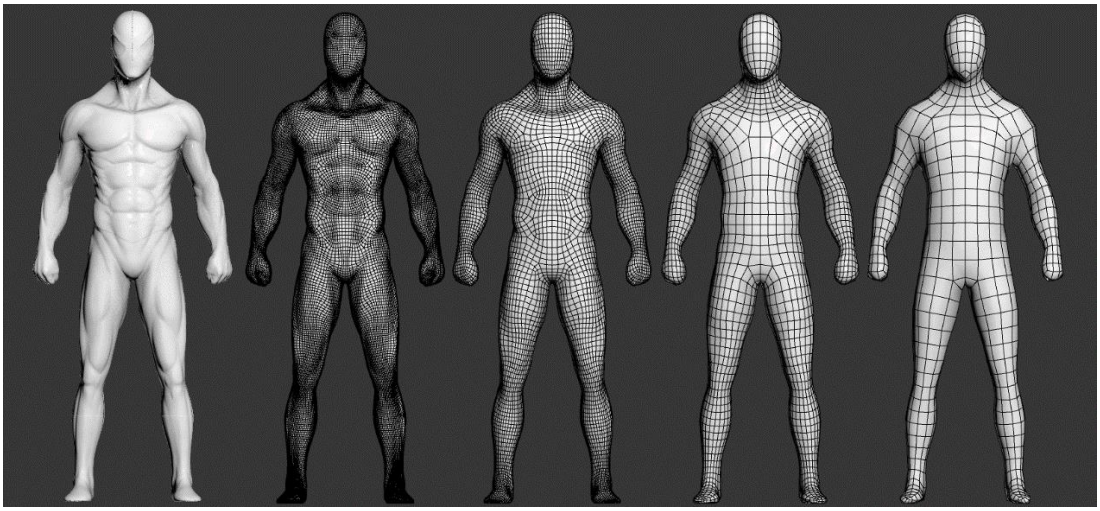

FIGURE 33. A HIGH-POLY MODEL WITH ITS LOW-POLY COUNTERPART

This part of the process is one of the most important to ensure a high-quality result, as avoiding lighting artefacts, errors in shadows cast by the models and other mistakes related to rendering the models in 3d software are entirely dependent on how the polygons are distributed throughout the model or, in other words, how the topology of our model is. Some programs can handle the automatization of this process; however, in the case of characters, it is recommended to do it by hand, as technology is not advanced enough to create adequately optimized models due to the definition of polygon chains called loops.

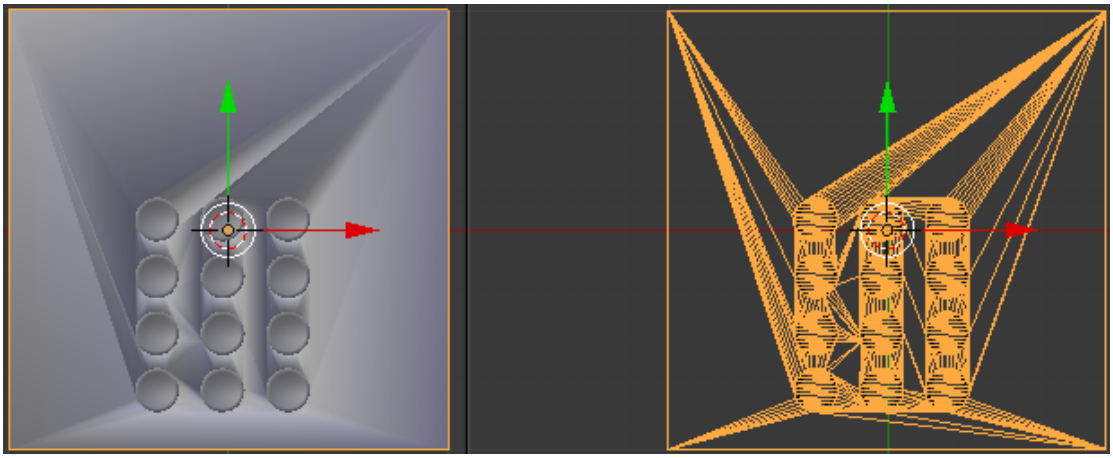

FIGURE 34. EXAMPLE OF LIGHTING ARTEFACTS IN A QUESTIONABLE TOPOLOGY

For characters especially, the correct definition of loops is crucial to ensure that the way the polygons bend when being animated does not deform the mesh in unwanted ways, especially in articulations such as the neck, armpits, elbows and knees of a humanoid and to define the volumes created in the blockout, depending on the level of realism creatives want to achieve. It also helps with future parts of the process, as 3D modellers use loops as guidelines for some automatization processes of the following steps. Hence, a good retopology involves the creation of loop chains around the volumes of the model that define its structure and then filling up the spaces with the number of polygons deemed necessary to be optimal while also generating the volumetric information needed to conform to the original high-poly sculpture.

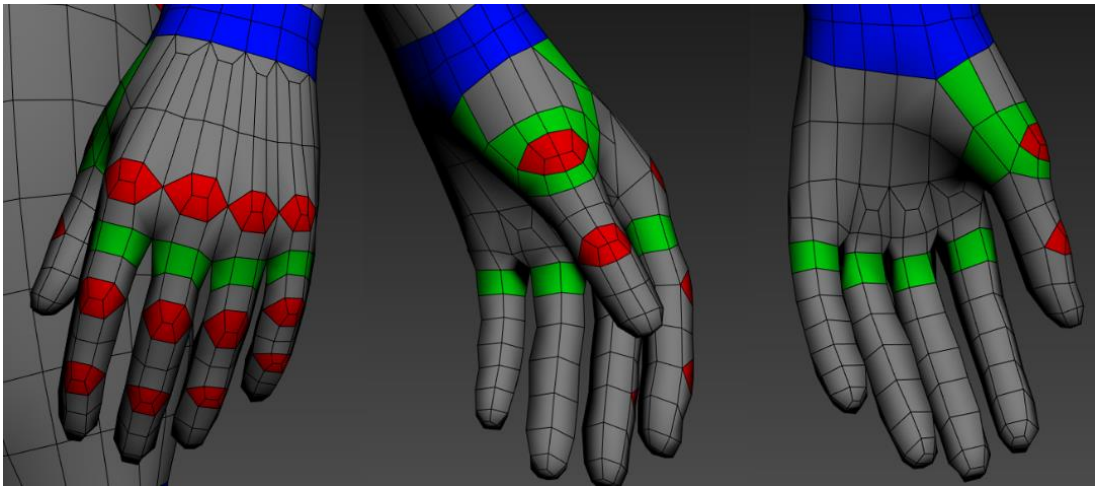

FIGURE 35. OPTIMIZED HAND AND ITS MOST ESSENTIAL LOOPS HIGHLIGHTED

2.4.5. UV Mapping: Unstitching a model

The model's file has to contain information assigned to all its polygons regarding which pixels each contains from the original image for it to be added as a texture. Thus, the model has to be projected in a two-dimensional space using UV Mapping. The letters "U" and "V" refer to the coordinates of an image in 2D, in the same way that 3D utilizes X, Y and Z to refer to each axis in space.

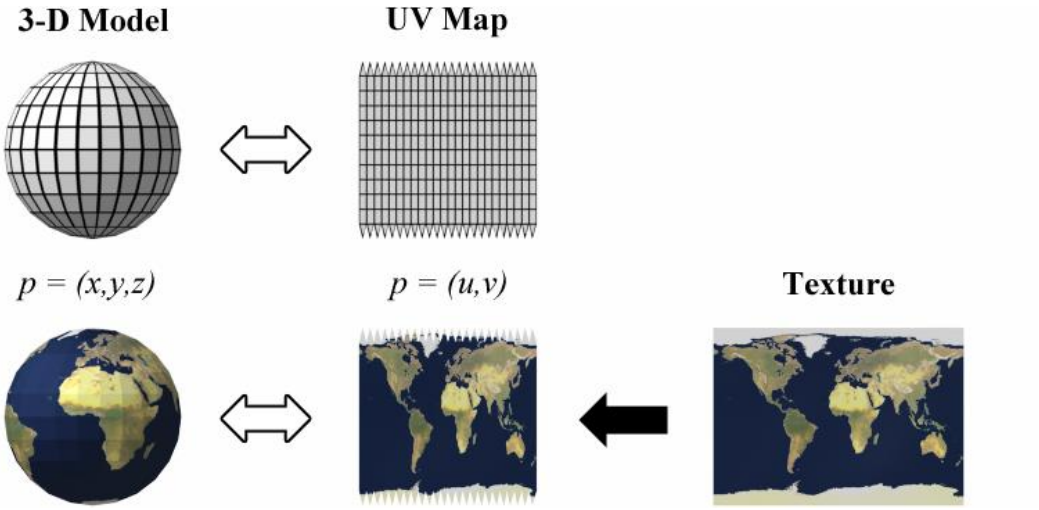

FIGURE 36. VISUAL EXPLANATION OF UV MAPPING USING THE EARTH

Most programs use a process similar to real-life knitting and unstitching, in which models are divided into seams that indicate how the models must be "cut" and divided to be unwrapped in a 2D space. These polygon zones — also called islands — have to be evenly distributed based on size and resolution. The more space in the image, the more pixel information it will contain, thus the more resolution it will have. A good packing of islands also helps create texture atlases, which are textures used for multiple models at once, thus optimizing the number of assets loaded at once in a game while sacrificing the resolution of the models.

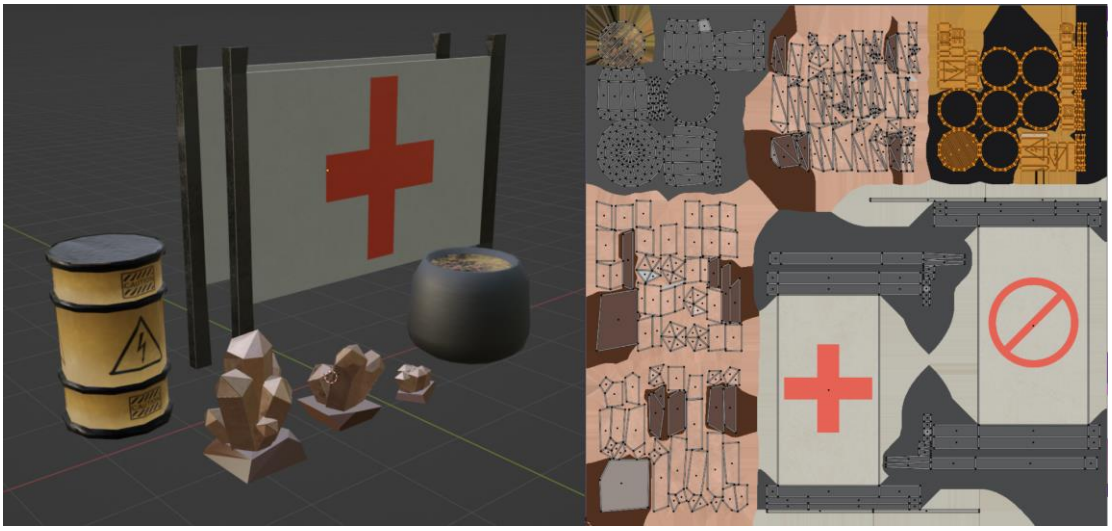

FIGURE 37. UV MAPPING OF MULTIPLE MODELS FOR THE SAME TEXTURE

Conversely, the same model can contain multiple textures simultaneously by dividing the mesh into multiple UDIMs— short for "U-Dimension", each for a different texture. In the case of colossal models, much like the subject in this project, creating multiple texture maps allows the distribution of different levels of detail for each of the parts that form the character, thus not having to rely on fewer images of enormous resolution and instead using multiple images of lower resolutions, which are easier to load while running an application.

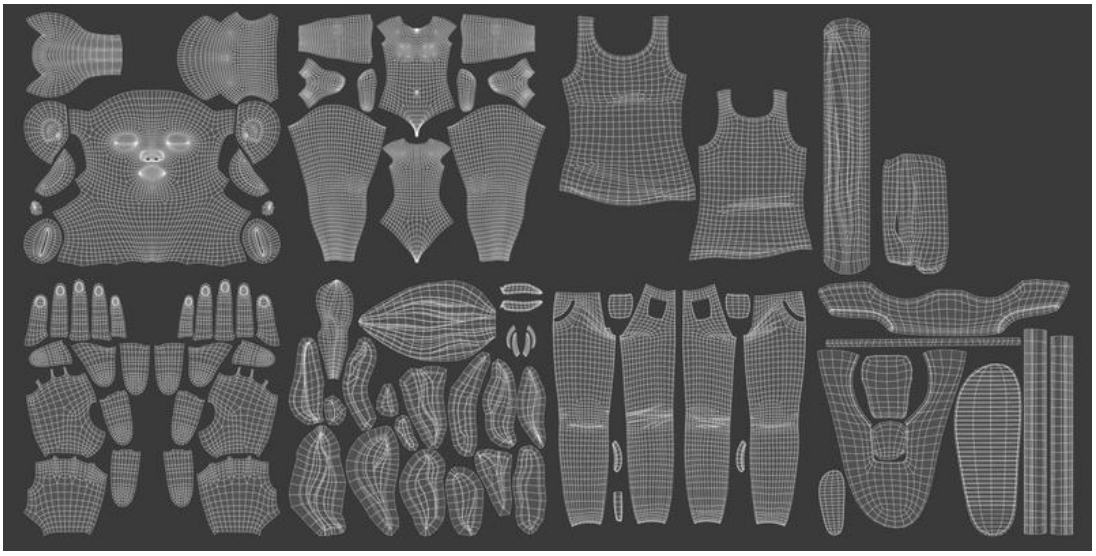

FIGURE 38. UV MAPPING OF A 3D CHARACTER IN DIFFERENT UDIMS

2.4.6. Texturing & Baking: Bringing colour

After UV mapping a model, the textures have to be created. Texturing refers to developing the images required to render a model in a game engine. Sometimes, it may be as simple as colouring the models in a diffuse or albedo texture; however, textures can be used to contain much more information for game engines. Depending on the rendering pipeline of the rendering engine in which the character will be rendered, different types of textures can be created as information for the model.

FIGURE 39. 3D CHARACTER TEXTURED USING A SINGLE IMAGE

The most common of these are normal maps. In short terms, normal maps refer to textures that rendering systems use as guidelines that convey information related to how light is reflected in each polygon, indicated by the image's colour. Depending on the direction in which light bounces in the model, parts will lighten and darken depending on the model's normal maps.

Usually, creating a normal map involves a process called baking, which essentially boils down to projecting the information contained in a high-poly model into its low-poly counterpart, allowing it to mimic the details in the sculpture and giving the sensation of depth and higher resolution while still keeping the small number of polygons from the optimized model.

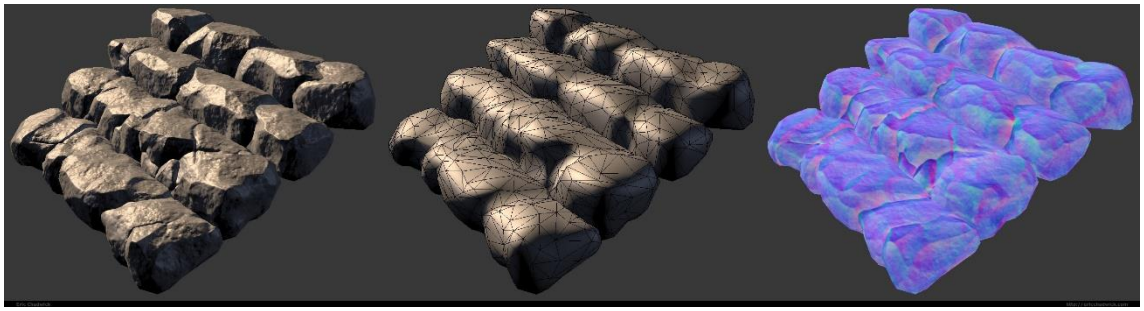

FIGURE 40. NORMAL MAP APPLIED TO A LOW-POLY MODEL AND THE RESULTS

Other texture maps can be used to improve the properties of a model's materials. The most common rendering pipeline in the industry is the Physically-Based Rendering pipeline — shortened to PBR. The maps involved in this pipeline are black-and-white, signifying a value going from 0 (black) to 1 (white), indicating how light affects the model (specular), how rough or glossy the material looks (roughness/glossiness) and how metallic it looks (metallic). They can be created separately using black and white channels or combined in a single RGB image that uses each colour channel for each property.

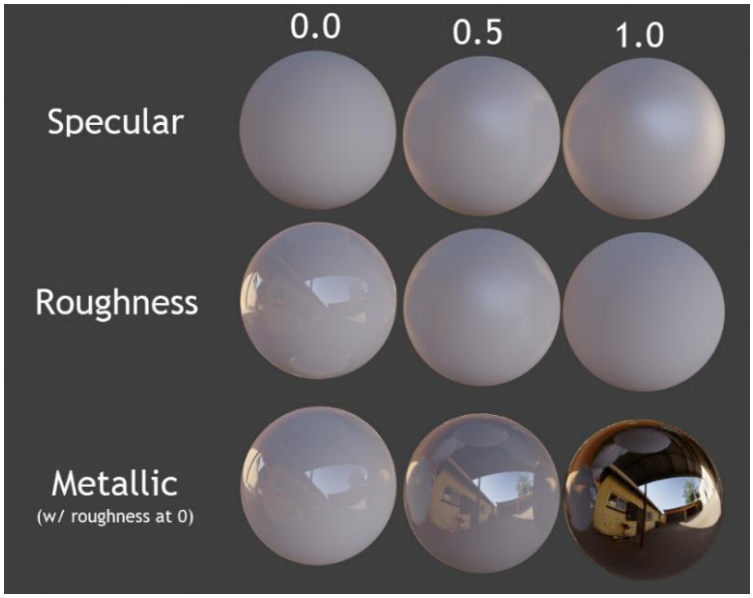

FIGURE 41. SPHERES WITH DIFFERENT VALUES OF SPECULARITY, ROUGHNESS AND METALNESS

In total, PBR Materials require between 3 and 6 images to work, depending on the level of optimization the developers want to achieve. In the end, depending on the materials' properties, how light interacts with them and what information the engine will need for post-processing procedures, more images will be needed for the same model, allowing for more detail beyond simple colours.

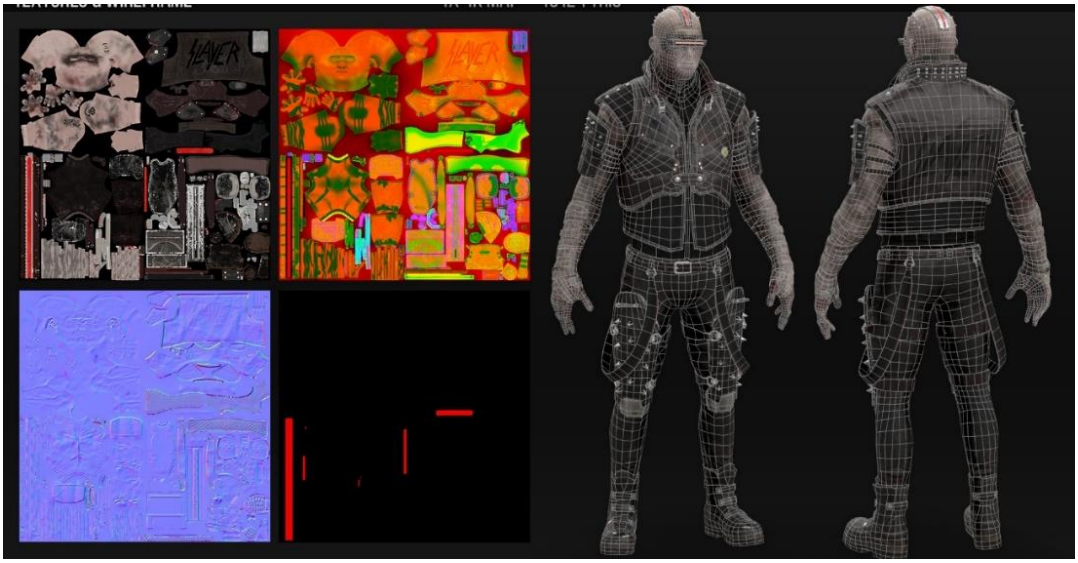

FIGURE 42. CHARACTER CREATED USING THE PBR PIPELINE

2.4.7. Rigging & Skinning: Defining the skeleton

Animation of characters in 3D media usually comes from the deformation of the polygons in their models. There must be indicators in critical parts of the character bodies indicating conjunctions and joints to indicate how and from where these polygons bend and stretch. This is where rigging comes in, a technique used to create a structure of elements called bones, forming a skeleton for the 3D model from which animators can manoeuvre and control, similar to a puppet.

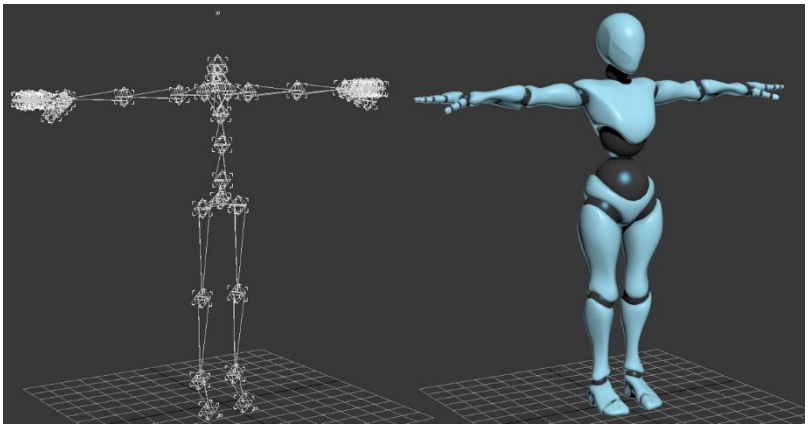

FIGURE 43. EXAMPLE OF A SKELETON CREATED FOR A HUMANOID CHARACTER

Just like the name conveys, a good rig imitates the main plausible bone structure that the creature to be animated would have in reality. For example, many quadrupeds have two joints for their back legs (like the horse in the top image), meaning three bones are needed to bend the legs correctly. Meanwhile, bipeds such as humanoid creatures only have one, meaning two bones are required to bend the structure correctly. This way, those in charge of creating rigs for characters — also known as riggers — can divide the influence of polygons that each bone in the skeleton has evenly in a process called skinning.

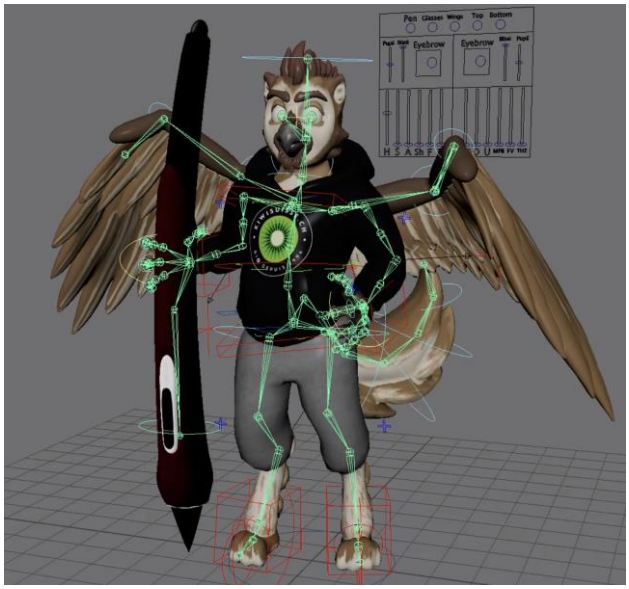

FIGURE 44. EXAMPLE OF A RIG WITH CONTROLLERS FOR THE SKELETON AND FACE (MADE BY THE AUTHOR)

However, controlling each bone can make some animations challenging or even downright impossible for animators. For this reason, riggers also create what is known as controllers, which are handles distributed across the skeleton that control it based on different conditions. The most relevant examples of these conditions are inverse and forward kinematics, two methods of controlling articulations such as arms and legs based on the point of reference from which the entire limb moves. If the most critical positions to control are the ends of the limb (such as in a walking animation), then animators will animate using inverse kinematics (or IKs); however, if it is the rotation of the arm what matters in the animation at play, then it is a better practice to use forward kinematics (or FK).

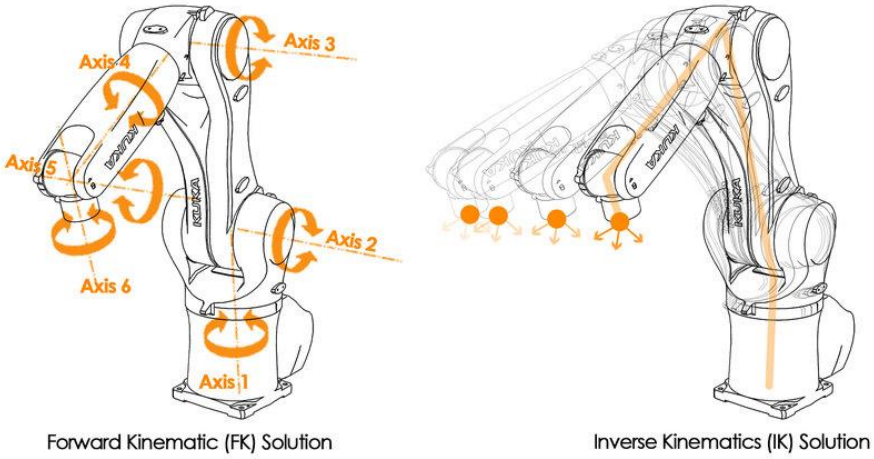

FIGURE 45. VISUAL REPRESENTATION OF FK VS. IK USING A ROBOTIC ARM

Some software even allows for animating mesh modifications for more minor animations, such as facial expressions or lipsynching — which means matching a voice with the animation. Controllers come into play when having to animate complex movements, and the more complex a rig is, the more movements, life and characterization animators can bring to characters and even props in an environment.

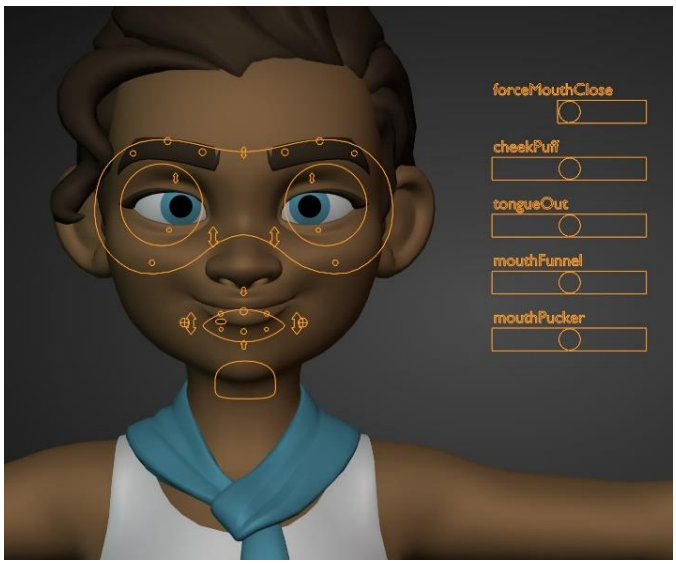

FIGURE 46. FACIAL RIG OF A GIRL WITH CONTROLLERS FOR MANY FACIAL ACTIONS

2.4.8. Animations: Bringing the life

Finally, with the rigging done, animations can be done. In this part, animators convey all the characterization and personality in the posing, movements and mannerisms of the character produced, following the 12 principles of animation (Thomas, and others, 1995). These principles were created by some of the most renowned animators at Disney, having been the key aspects they considered to make good 2D animation. In no particular order, these are Squash and Stretch, Timing and motion, Anticipation, Staging, Follow Through and Overlapping Action, Straight Ahead and Pose-to-Pose Action, Slow In and Out, Arcs, Exaggeration, Secondary Action, Solid Drawing and Appeal.

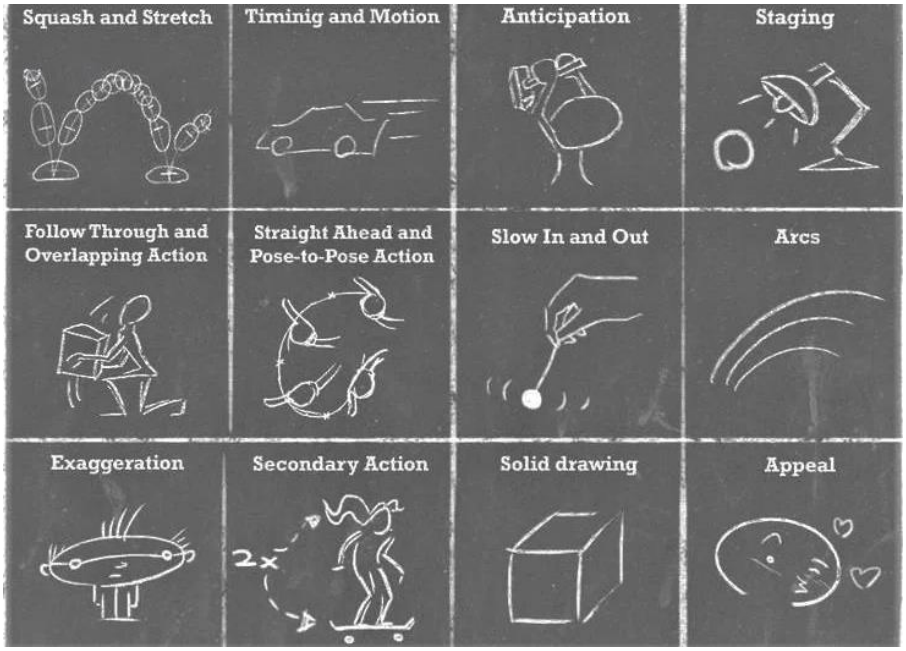

FIGURE 47. VISUAL REPRESENTATION OF THE 12 PRINCIPLES OF ANIMATION

However, 2D animation principles primarily serve their medium. In traditional 2D animation, the process involves drawing all of the many frames detailing a step of the characters and environment's movements and changes in space, based on a fictional camera's position, which then will be changed in succession at a rate of 24 frames per second (shortened to FPS), giving the illusion of movement. In 3D media, while still having to animate at the same framerate, animation usually relies on an automatization process called interpolation.

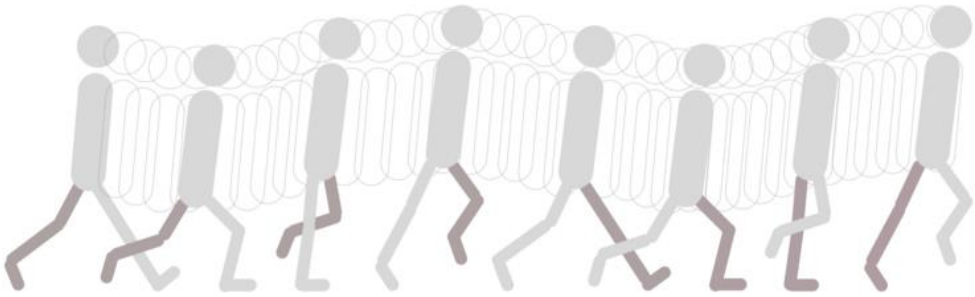

FIGURE 48. FRAMES OF ANOTHER WALKING CYCLE WITH SILHOUETTES INDICATING INTERPOLATED FRAMES

In a timeline, animators save the position, rotation and scale of each bone in the skeleton at specific frames in time, creating different poses that define the main action of the animation — also known as keyframes. From these keyframes, 3D software automatically creates in-between frames that give fluidity to the animation, using splines as points of reference from which animators create the curvature of the animations at play, thus upping the framerate from 24 FPS to the standard 60 FPS in all video games nowadays. With all the animations done, the characters can finally be implemented in the game.

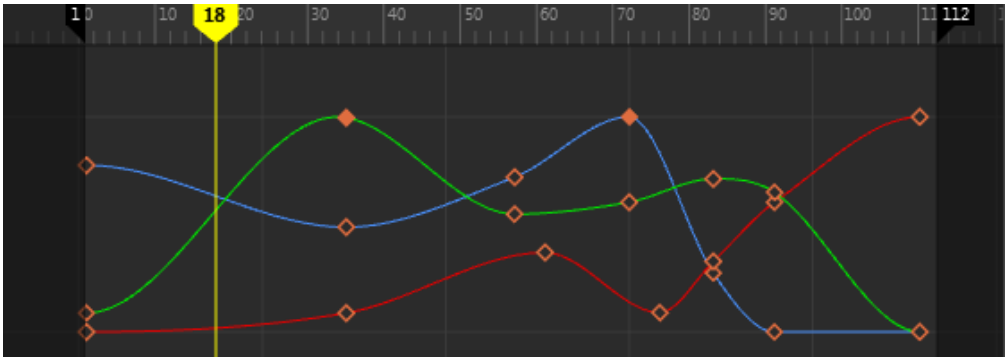

FIGURE 49. ANIMATION REPRESENTED USING A GRAPH EDITOR, WITH EACH LINE REPRESENTING THE INTERPOLATIVE MOVEMENT BETWEEN KEYFRAMES

2.4.9. Shaders: The art in code

Just like any other virtual asset, computers have methods to visualize data and information through a screen. How a graphic is visualized in computers is determined by several steps created by programs in GPUs called shaders (de Vries, 2014), a term with its first instances appearing in 1989's 3.1 Renderman Interface Specification by Pixar (Pixar, 1989). These programs control the lighting, darkness, glossiness and metalness based on inputs given by variables like numbers, words or, in most cases, materials. Such is the case of the previously mentioned PBR pipeline, which uses the information of different black-and-white maps to create information about how light interacts with the object's surface. The primary language for shader programming is the OpenGL Shaders Language (GLSL for short), and it can be used to create everything we see on a screen, from simple triangles to beautiful 3D models and animations.

FIGURE 50. 3D RENDER OF A SNAIL BY IÑIGO QUILEZ USING ONLY CODE IN SHADERTOY

In video games, using shaders is crucial for creating the aesthetics of a game's looks, as these can change the feel and look of an experience with simple mathematical operations. In this case, the role of these shaders can range from complementing the materials with simple visual effects to overhauling how they look in-game, as these programs are applied as the final step between the original image and the final rendering. They can also reduce the impact that rendering a topology or material can take in a game by reducing the number of mathematical operations the GPU requires to render specific effects on-screen.

FIGURE 51. EXAMPLES OF DIFFERENT USES FOR SHADERS FROM THE UNITY SHADER BIBLE

While programmers and technical artists are in charge of using shaders to create these effects, some tools have been developed for artists to use more user-friendly that do not require coding. These come in many software in the form of shader graphs, visual representations of code using interconnected nodes that call for functions sequentially from left to right, based on the user's needs. Hence, artists can create astonishing visual effects using easy-to-program nodes instead of code.

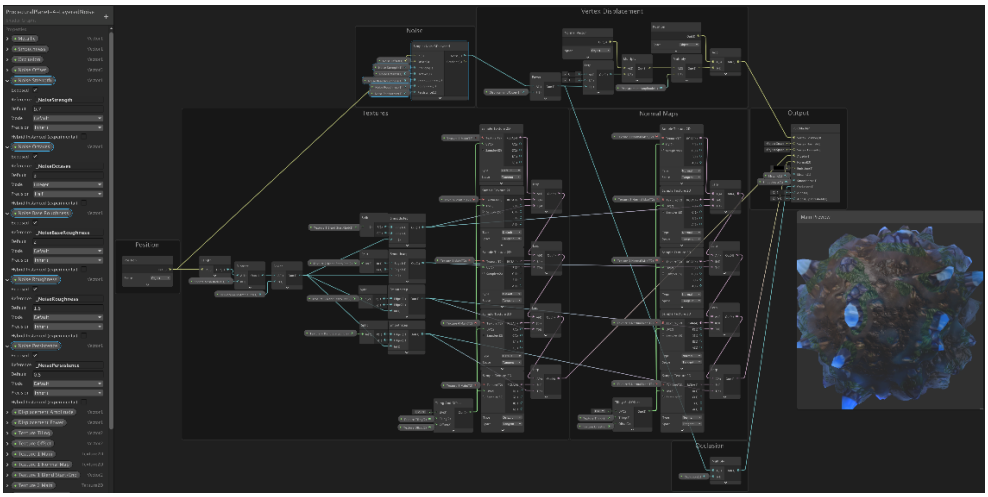

FIGURE 52. EXAMPLE OF SHADER USED TO CREATE PROCEDURAL PLANETS

2.5. Software for 3D Art

The software is the most critical tool for creating 3D graphics in digital creation. By the time the author writes this project, many programs allow for creating and visualising models, textures, rigs and animations that give life to several games in the industry. They usually try to supply all of the processes in 3D asset creation; however, developers usually use several staples in professional contexts in the industry. Some programs are built with different purposes and thus have more efficient tools for different parts of the process. Ultimately, companies choose programs based on the license cost and team workflow. For boss design, just like any other character, this is the assortment of programs typically used in game studios:

2.5.1. Modelling

Software that artists use for model creation has to allow for generating and modifying vertices, triangles and polygons that will constitute the shape of the mesh, starting from a primitive shape such as a cube, sphere or plane and ending with the complex characters and environment assets in video games (Petty, 2019). The pipeline to achieve this varies from program to program, but the most used software in the industry, as recommended by experts (Coggan & Guilcher, 2022), is the following:

2.5.1.1. Autodesk Maya (Modelling)

Autodesk Maya — usually abbreviated to Maya — is a program from the Autodesk suite created to design and produce game models. 3D artists heavily use it to create, visualise and render low poly assets, with other capabilities for texturing and animation. It is very versatile and has many other tools for the entirety of the 3D art process. In the case of 3D modelling, Maya offers all the tools for low poly creation, with special mention of the Polydraw workflow, which allows for the fast creation of polygons and meshes. However, it can get complicated for inexperienced individuals. It offers heavily advanced tools not often used in the video game industry but in the cinema industry or architecture, such as rendering features or the possibility of creating realistic hair using fibres, a rendering technique to create hair-like textures.

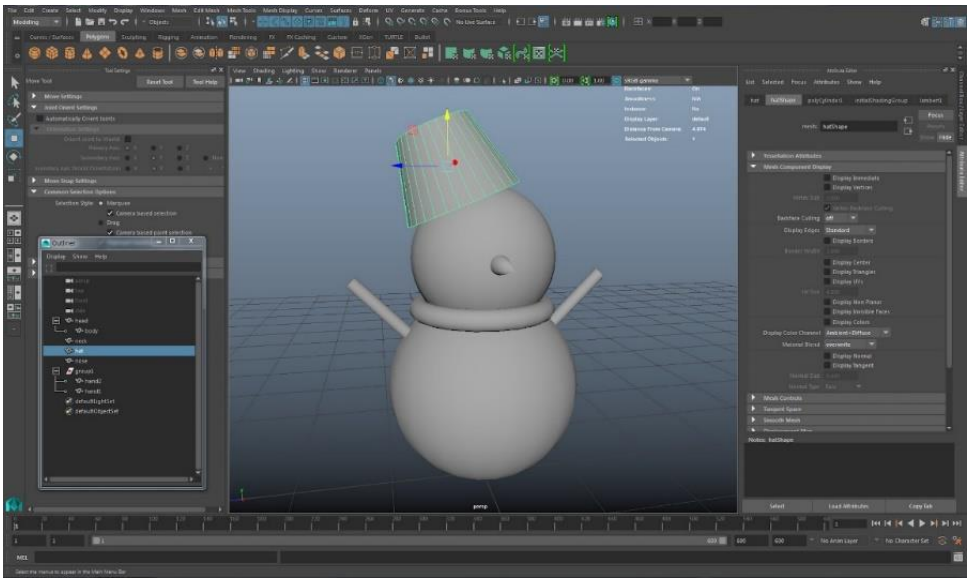

FIGURE 53. AUTODESK MAYA'S INTERFACE

2.5.1.2. Autodesk 3Ds Max

Like Maya, as they belong in the same suite, Autodesk 3Ds Max — also known as only 3Ds Max — is another modelling tool designed to create 3D assets. It is similar to Maya in most of its capabilities; however, in terms of video game art, it lacks animation capabilities and many tools for faster asset creation when it comes to low-poly creation. However, it is superior for primitive editing without modifying them by vertices or faces. With modifiers, artists can bend, twist, extrude and manipulate the overall shape of meshes as they please, allowing for the quick creation of blockouts and general shapes without getting into detail.

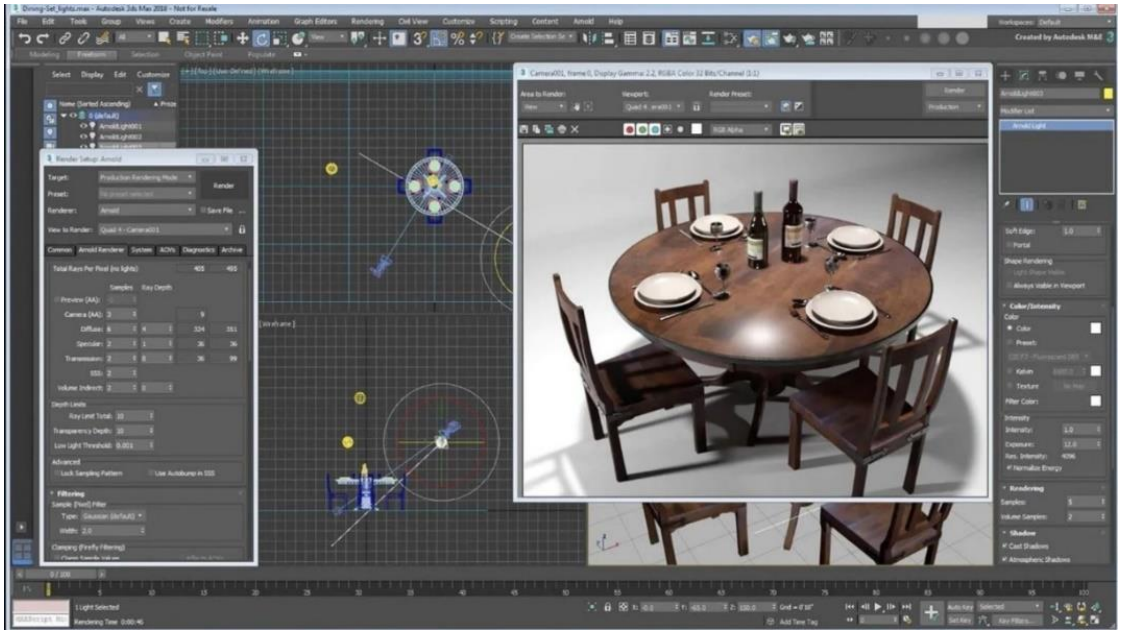

FIGURE 54. AUTODESK 3DS MAX'S INTERFACE

2.5.2. Sculpting

As mentioned, sculpting software allows for creating details, crevices and more elaborated shapes than typical modelling software. These programs must simultaneously allow quick rendering and manipulation of hundreds of thousands of polygons, as all details must be represented in the model's topology. For that, the software that is best prepared to deal with this is the following:

2.5.2.1. ZBrush

ZBrush is a tool developed by Pixologic, mainly focused on digital sculpting. It is used for organic 3D sculpting and offers compelling options and brushes for character creation, simultaneously allowing for the visualization and rendering of millions of polygons. It also contains several algorithms and tools that automatically remake mesh topologies, create intricate surface patterns, and generate textures and highly detailed shapes. Despite the default UI being intuitive initially and thus offering a harsh learning curve, it is one of the most used software in the industry, with an assortment of add-ons made by the community that allows the creation of beautiful high poly models.

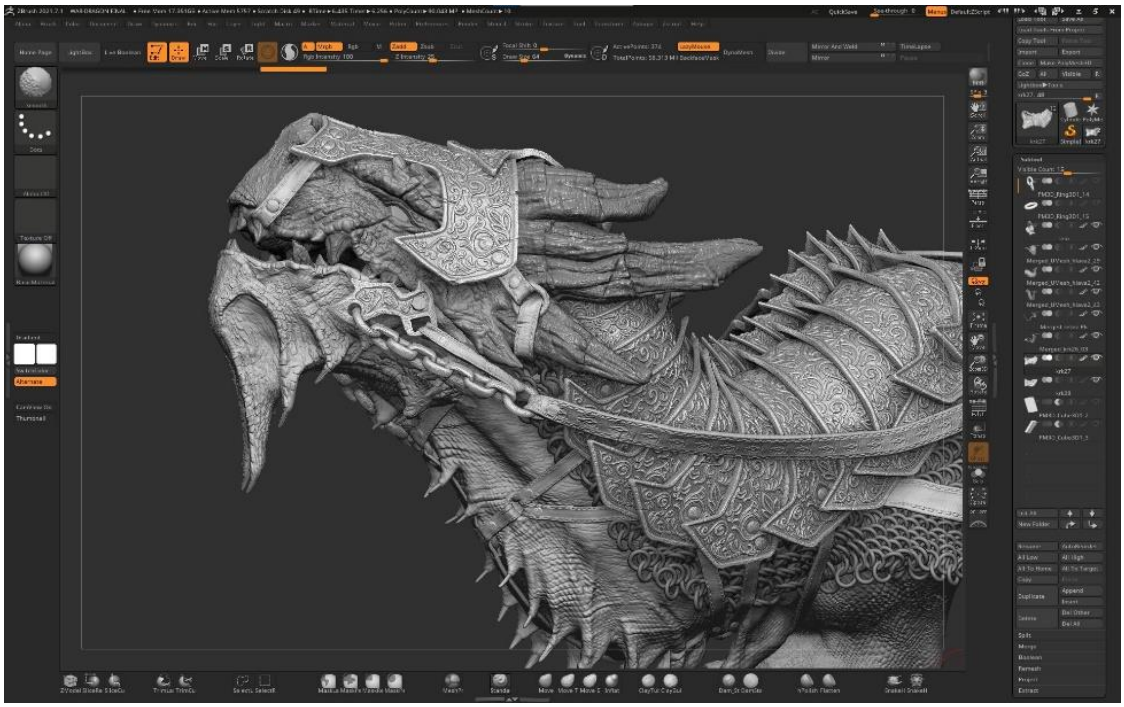

FIGURE 55. ZBRUSH'S INTERFACE

2.5.2.2. Autodesk Mudbox

Autodesk Mudbox — usually abbreviated to Mudbox — is also a program from the Autodesk suite focused on creating high poly sculptures and models. While it offers many of the utilities seen in ZBrush, it lacks the power and capabilities that the former holds and pales in comparison to functionalities and manipulation of models. As such, many artists use Mudbox in the case that their entire pipeline relies on using other software in the same suite, as it allows for quick compatibility between programs in the same family and also uses a similar UI structure to the other programs mentioned so, in terms of a learning curve, it is simpler than other sculpting programs.

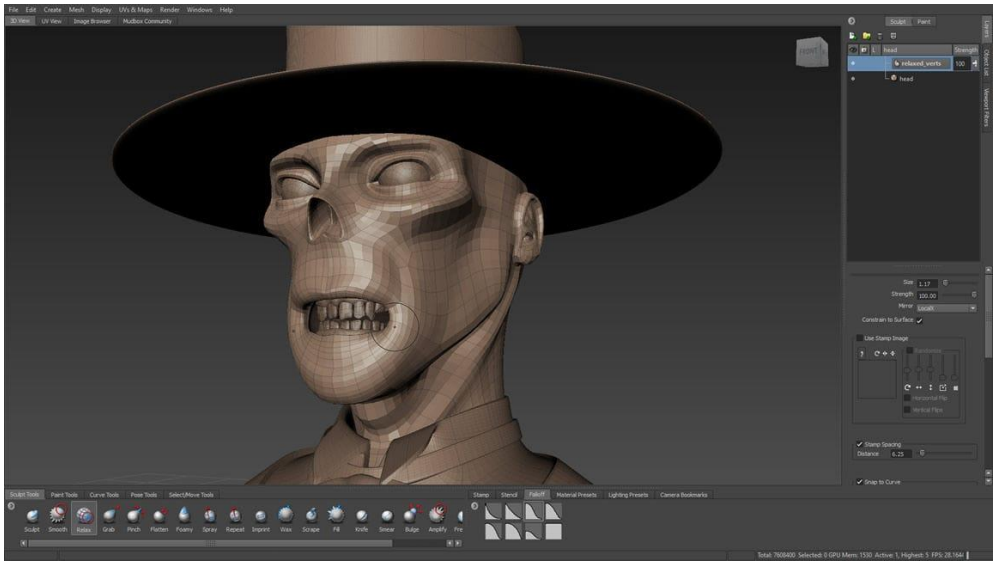

FIGURE 56. MUDBOX'S INTERFACE

2.5.3. Texturing

Regarding texturing, programs that allow for painting models offer capabilities similar to 2D image creation software but for 3D meshes. The industry's best software allows for blending seams' capabilities and procedurally generating textures that do not stretch due to poor UV optimization. Hence, some of the programs that are great for this are the following:

2.5.3.1. Mixer

Mixer is a program from the Quixel suite created by the company of the same name. It has many of the capabilities already seen in Substance Painter, such as Smart Materials, a layer workflow or baking capabilities. However, it is compelling for hyper-realistic textures, as it is compatible with other software such as Megascans or Unreal Engine, which usually opt for creating hyper-realistic environments and games with 3D models scanned from real life.

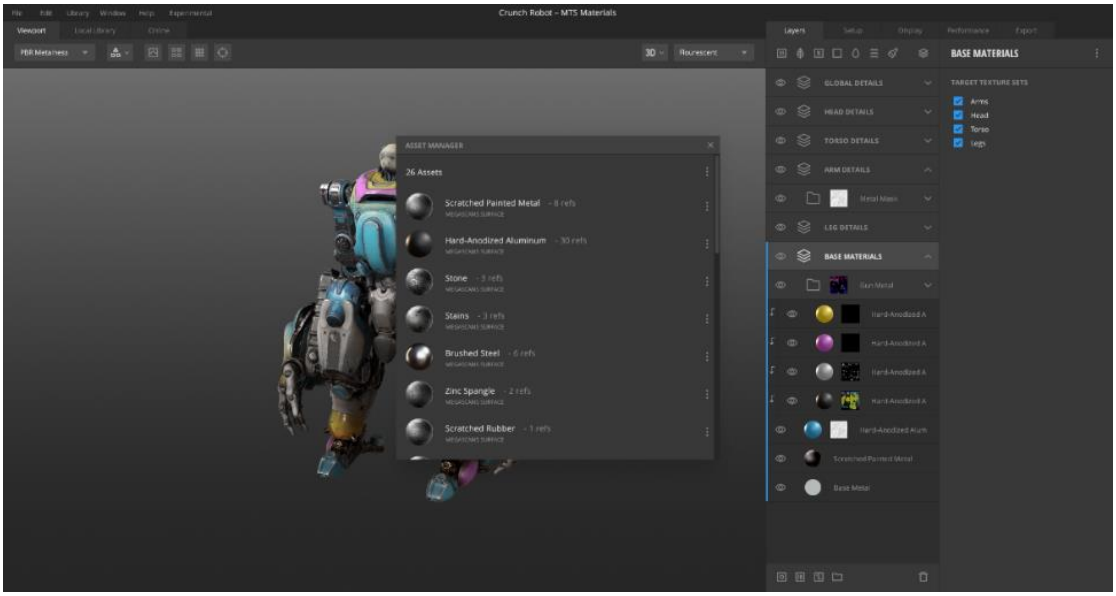

FIGURE 57. QUIXEL MIXER'S INTERFACE

2.5.3.2. Substance Painter

Substance Painter is software from the Substance Suite created by Allegorithmic, released in 2014 and currently owned by Adobe. It mainly focuses on creating textures for 3D models, allowing for procedurally generated materials (which the program refers to as "Smart Materials") or hand-painted ones in a fashion similar to Photoshop, working with layers and masks. As a more advanced capability, it also allows for baking maps of information using high poly models as a reference to create better procedurally generated textures. It is used in the industry as it allows for direct integration with other 3D software and game engines.

FIGURE 58. SUBSTANCE PAINTER'S INTERFACE

2.5.4. Animation

For animation, the capabilities software offers to create lifelike movements are already mentioned in the boss creation section, such as the creation of skeletons, controllers and keyframes (2.4.7 & 2.4.8). In this case, there are many software that allow for relatively easy creation of animations, such as:

2.5.4.1. Autodesk Maya (Animation)

The animation capabilities of Maya are some of the best already seen in the industry. With the possibility of editing keyframes with the graph editor, using blend shapes for animations that require mesh modifications in real time and controller creation for IK and FK options, Maya is the standard for animating complex movements. With add-ons such as Advanced Skeleton 5, which offers auto rigging capabilities, animating in Maya is super easy, and many experts in the industry recommend it for those just starting in the business (Coron & Guilcher, 2022).

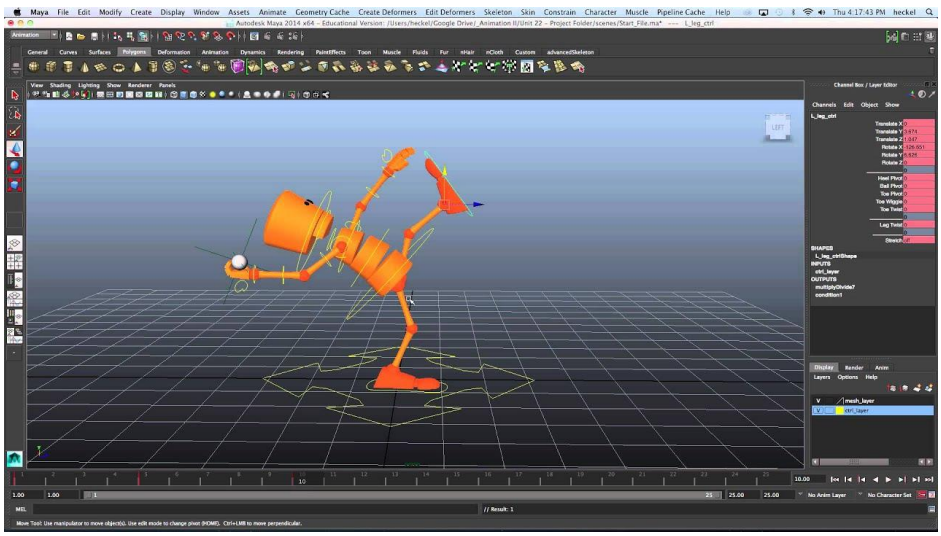

FIGURE 59. MAYA'S INTERFACE WITH A MODEL BEING ANIMATED

2.5.4.1. Mixamo

Mixamo is a website allowing auto rigging capabilities and exporting animations for humanoid figures. The software allows people without knowledge of animation and rigging to detect the limbs of the character in character and create a skeleton prepared to use animations from the vast library that the company offers uses. Being completely free, Mixamo is one of the fastest ways to create simple animations for games, so solo developers can use this website if necessary.

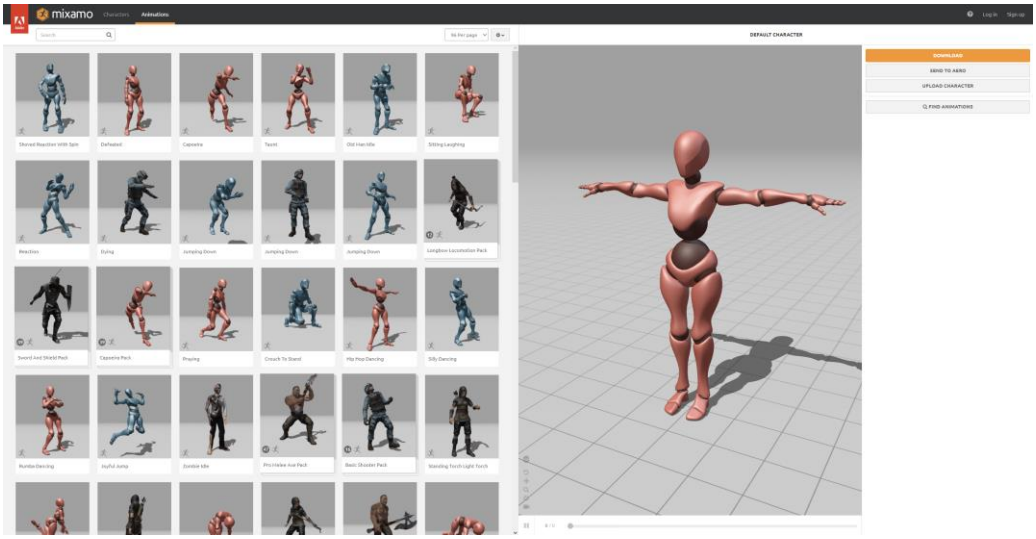

FIGURE 60. MIXAMO'S INTERFACE

2.5.5. The case of Blender

As a special mention in its separate category, Blender is a free application for 3D asset production. It is constantly updated and offers many capabilities for all the process steps, as it is incredibly modular and customizable. It is an excellent tool for starters, very intuitive and capable of creating complete 3D games, which is why it has been gaining popularity in the past few years.

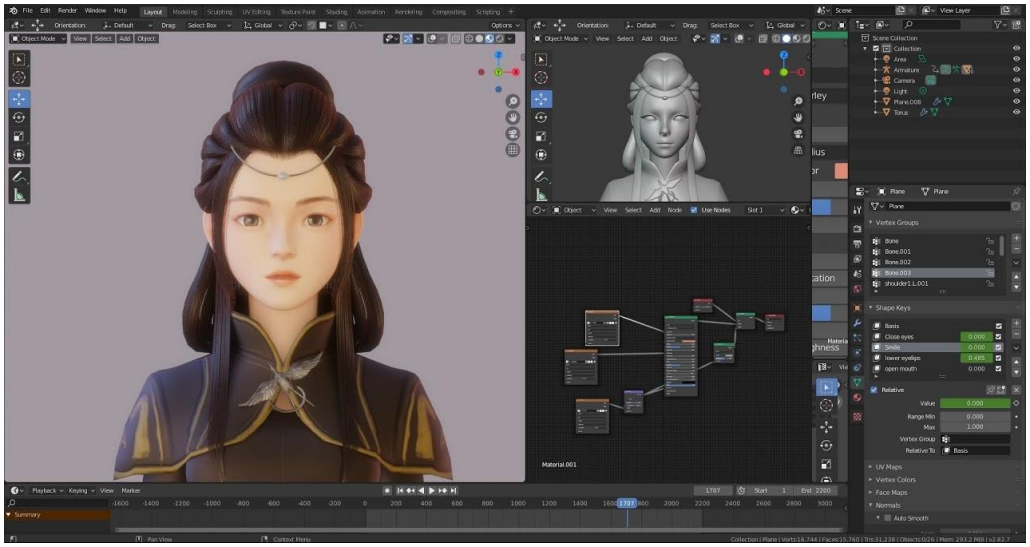

FIGURE 61. BLENDER'S INTERFACE

As it turns out, it offers modelling tools for both low-poly and high-poly creation, UV mapping capabilities, texturing features and a complex system of nodes for shader creation. Since the community is very active, many users have created add-ons that upgrade the experience to new levels that, considering it is a free program, offer different and sometimes better capabilities than the previous programs. However, some argue that it is an example of a jack of all trades and a master of none, as other established staples in the industry have more developed and optimized tools that Blender does not offer.

Nevertheless, as all the pipeline of asset creation is inside the same program, the learning curve of this software is not very steep. The amount of tutorials and tools the community and developers create for the public has made Blender one of the most accessible programs in the market and a good recommendation for those starting in the industry.

2.5.6. Artificial intelligence

For asset creation as mentioned earlier, artificial intelligence can help in the creation of assets by generating images for the concept. They work essentially using an extensive library of images of specific concepts and, based on a prompt, using the library as a reference to generate an original image after many iterations and steps based on a neuronal network (Riley, 2022). However, how these algorithms work can vary based on the parameters, capabilities and models used to create them.

2.5.6.1. DALL·E 2

DALL·E 2 is the first AI image generator on the market, created by OpenAI in 2021. It offers high-quality images based on simple prompts, creating four instances from that sentence. It also allows for creating variations of one of the instances for more fine-tuning of the perfect image. While being initially free, non-paying users have a limited amount of credits they can use for each prompt, but as it is effortless to use, with the correct prompt engineering, artists can create fast concepts in no time.

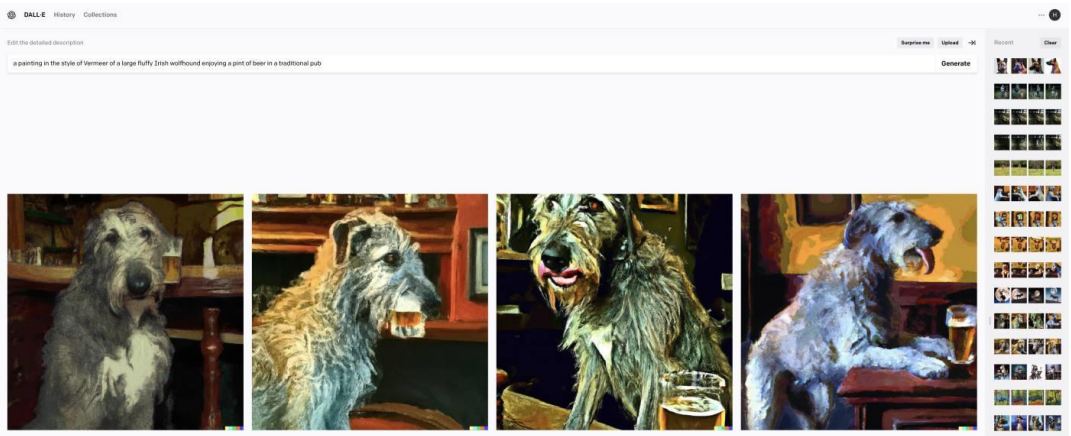

FIGURE 62. DALL·E 2'S INTERFACE

2.5.6.2. Midjourney

Midjourney is a direct competitor of DALLE·2, developed by a team of 11 people. As they put it on their website, "Midjourney is an independent research lab exploring new mediums of thought and expanding the

imaginative powers of the human species" (Midjourney, n.d.). It works by using commands directly inputted in their discord server, starting with the input "/imagine", which, much like DALLE·2, creates four variations of the same image and allows permutating one of the options if the user wants to. It also requires payment, but its results are more coherent, lifelike and realistic than those other AI generate.

FIGURE 63. IMAGES OF DOGS IN A BAR GENERATED BY MIDJOURNEY

2.5.6.3. DreamStudio

Finally, to finish the trifecta of artificial intelligence software in the market is DreamStudio, known mainly by the public as Stable Diffusion, the name of the program's algorithm. Developed by Stability AI and unlike the former, DreamStudio is open source, downloadable and highly customizable. It comes with parametric sliders to determine the size and amount of the images to generate, how close it resembles the prompt given and even how many steps the model takes. However, it also needs credits, which users must buy with money.

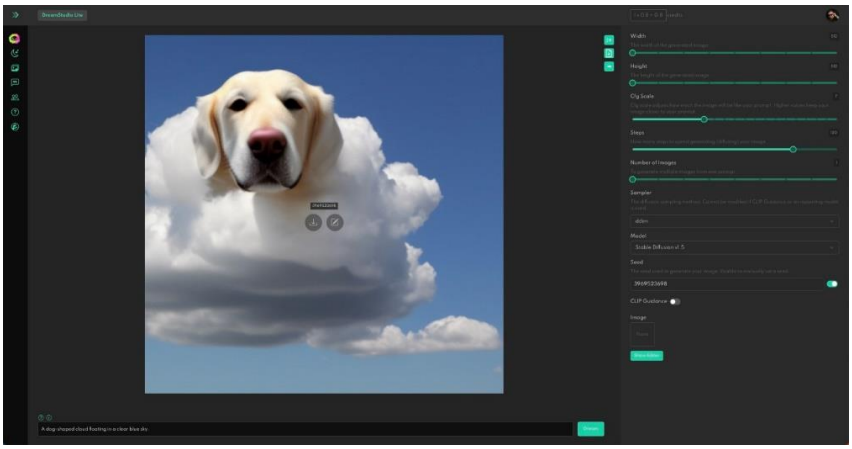

FIGURE 64. DREAMSTUDIO'S INTERFACE

2.5.7. Game Engines

Finally, to create games, many developers make use of programs named engines, which allow for the quick designing and creation of games with integrated tools for scripting, audio configuration, resource management, artificial intelligence, rendering, shading and UI systems. All of these elements are required for creating professionally looking videogames, and for 3D games, the best programs in the industry are the following:

2.5.7.1. Unity

Unity is a free program designed for video game production. It is one of the most versatile and easy-to-use game engines in the market, used by many companies for console, PC and mobile games, offering a vast assortment of tools and in-built scripts for every need. It works brilliantly for triple-A productions and relatively fast and small prototypes like the project. In artistic terms, it allows for both 2D and 3D games, allowing for highly personalized rendering pipelines and shaders that can offer products an individualized sense of identity and a unique look.

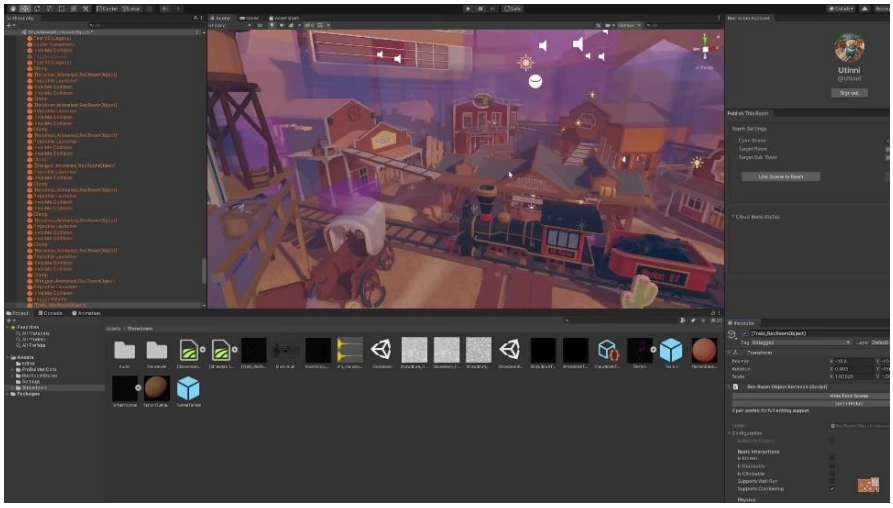

FIGURE 65. UNITY'S INTERFACE

2.5.7.2. Unreal Engine

Unreal Engine is another free game engine well-established in the market developed by Epic Games, initially created for programmers for their first-person shooter Unreal. Nowadays, however, it is the most powerful tool in the market for game creation and real-time rendering, allowing for hyper-realistic interactive experiences or even the creation of sequences for cinema. It is very artist-friendly, mainly using visual scripting tools to code and program games, and it also offers several great tools for environment and lighting creation.

FIGURE 66. UNREAL ENGINE'S INTERFACE

2.6. Market Study

The key to a compelling market study is achieved by initially searching titles that directly tackle the objective settled for this project. In a project related to artistic creation, analysing game models used in the industry is vital to showcase how developers use the pipelines explained above.

However, game assets are tough to obtain for most video games, as developers often protect them using data encryption and packaging. Nevertheless, in the case of older games, some avid individuals have figured out methods to extract game models from their original data. Therefore, getting some of the bosses' models used in games and importing them into 3D software for this analysis has been possible.

For this project, the student has chosen six cases of study from all ranges of novelty and production value to compare, test and analyse the pros and cons of what made them great or improvable. The author has also analysed their models and peculiarities, if available, to contrast with the pipeline showcased in previous sections.

2.6.1. God of War III: Cronos

God of War III is the fifth instalment in the wildly acclaimed series God of War, originally released for the PlayStation 3 on the 16th of March 2010 by Santa Monica Studio. Players control Kratos, former god of war, in an action game from the hack-and-slash genre relying on combo-based combat with multiple weapons obtained throughout the game, with the main objective being destroying the Olympian pantheon as vengeance for the deaths of the protagonist's wife and daughter.

At some point in the story, midway through the game, the protagonist has to retrieve the Omphalos Stone, an ancient artefact believed to be the stone that the titan Cronos swallowed, mistakenly believing it was Zeus when the god was a baby, to create a powerful weapon to fight the gods. Thus, he embarks on a journey to Tartarus, where he finds the gigantic Titan and a battle of biblical magnitude ensues. Kratos must travel across different parts of his body to defeat him, destroying minions appearing along the path and harming several parts of it. In the end, after being swallowed alive by the titan, he slices open his intestines, safely retrieving the stone and ending the tortured giant's misery.

FIGURE 67. CRONOS TRYING TO CRUSH KRATOS BETWEEN HIS FINGERS, *GOD OF WAR 3*

It is one of the most expensive, complex and epic sequences created by the studio, lasting for two years of the game's development for an average playtime of 25 minutes, with 35 people involved — 5 of them entirely dedicated to this battle alone. At 500m. tall, it has more than 18 million polys and 500 joints for 420 animations.

2.6.1.1. Production Process: A multimillion beast

As stated by its most relevant employees, Santa Monica's philosophy has always been to start big and take risks not to shorten the creativity of the developers, despite them having to scale it down after many playtesting sessions. In a GDC talk done in 2012, senior-level designer Chris O'Neill and lead in-game animator Bruno Velazquez explained the process behind the conception and production of such an endeavour, starting with establishing the vision of the sequence with ideas, key points and epic set pieces that would then define the experience.

For the project, the team started brainstorming ideas that would then be voted on to have a clear idea and objective of what to expect from the battle. From a design perspective, they wanted to achieve the feeling of scale and how minuscule Kratos and other elements of the environment are in comparison to this beast, so, to achieve this, they had to rely on relativeness: with waves of some of the gigantic bosses the player has fought throughout the adventure appearing as mere normal monsters that pale in comparison to the titan. They also wanted to keep a comfortable and plausible rhythm while offering gratifying real-time payoffs.

With this in mind, both artists and programmers start with concept art and storyboarding, depicting the major events that will and prototyping with premade assets such as a cyclops model from the previous instalment of the series. For this boss alone, they had to develop new technologies and systems to relate character state animations with object orientations and adapt the collisions for meshes by using multiple primitives in the primary model. The team did not want to develop heavily complex systems, as it may lead to severe time losses for the production. However, that did not stop the team from developing new collision and animation systems.

FIGURE 68. CONCEPT ART & EARLY PROTOTYPES OF CRONOS' BATTLE, *SANTA MONICA STUDIOS*

The process let the studio realize many of the challenges of these types of productions. For starters, character and environment artists had to work together to create the proper geometry to better define textures and paths throughout the body. Other experts, such as level designers and animators, had to work in unison to create the level itself, as they were heavily intertwined. Animations had to be stable for the player to stand on the boss, but they also had to be aesthetically pleasing.

Regarding software, Santa Monica Studios uses some of the best staples in the industry to create a monster of such gigantic proportions. In the talk, the studio showcases images of the model as seen in Autodesk Maya, showcasing the program's capabilities in modelling and animations. In Tyler Breon's Artstation, he posted screenshots of how he made Cronos' sculpting in ZBrush, which offered the best tools for creating a beast of such magnitudes at the time of development.

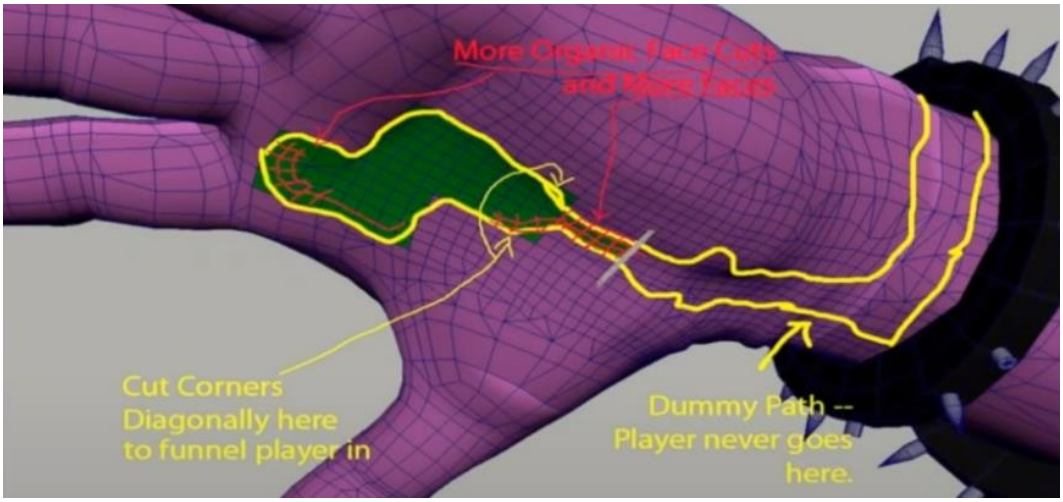

FIGURE 69. CLOSE-UP OF CRONOS' HAND WITH NOTES FROM DEVELOPMENT, *SANTA MONICA STUDIOS*

To finally sell the experience, they had to bring to life a believable yet emotional buildup for a climactic end. Programmers had to cut many elements due to technical and time constraints, but with the usage of several elements such as slow, oversized and heavy animations, grandiose visual and sound effects, camera work and other post-production effects, they kept the dynamic motion and frenzy that comes from asking how a man would kill a giant.

FIGURE 70. RENDER OF CRONOS' FACE EXTRACTED FROM TYLER BREON'S ARTSTATION

2.6.1.2. Design Analysis: A changing environment

This battle is a pretty effective pace change from the puzzles and static wave-based action the player has had to tackle. Instead of traversing typical structures, Kratos has to go through a seemingly constant action-packed living level, showcasing his power throughout several body sections to navigate and fight against minions.

The scenario keeps changing based on the Titans' actions, with the camera stopping at certain key angles that showcase the size difference between the two characters. The animation of the titan is created so that its most significant arcs appear during cinematic shots while staying still during gameplay bits, creating several challenges like navigating an arm from below or while the other one chases the player to make them fall. However, despite being seemingly chaotic to navigate, it achieves its objective of feeling organic, both in terms of pacing and level design, as it uses clear paths delimited by rocky skin that serves as a way for the player to know where to go next.

FIGURE 71. IN-GAME SCREENSHOT OF A PATH IN CRONOS' ARM DURING HIS BOSS BATTLE, *GOD OF WAR 3*

Mechanically speaking, it is hard to consider it a boss battle based on standard definitions. For the battle to work, the designers had to sacrifice many direct interactions between the player and the titan's attacks and movements. Hence, the sequence becomes a glorified series of trim levels between QTE[6]-based cutscenes. The challenge, if any, does not come from the boss's behaviour but rather the player's navigation throughout the creature's body, and so the biggest threats end up being the minions that keep blocking Kratos' path.

Nevertheless, this has not stopped the titan from becoming one of the most prominent achievements of the studio. It is important to remember that the initial objective was never for the titan to be a challenging obstacle but rather a spectacular showcase of raw power emanating from the player. From an artistic and technical point of view, Cronos is truly an example to follow for immersive and frenetic experiences.

[6] QTE: Abbreviation for "Quick Time Event"

2.6.1.3. Model Analysis: The complexity of a titan

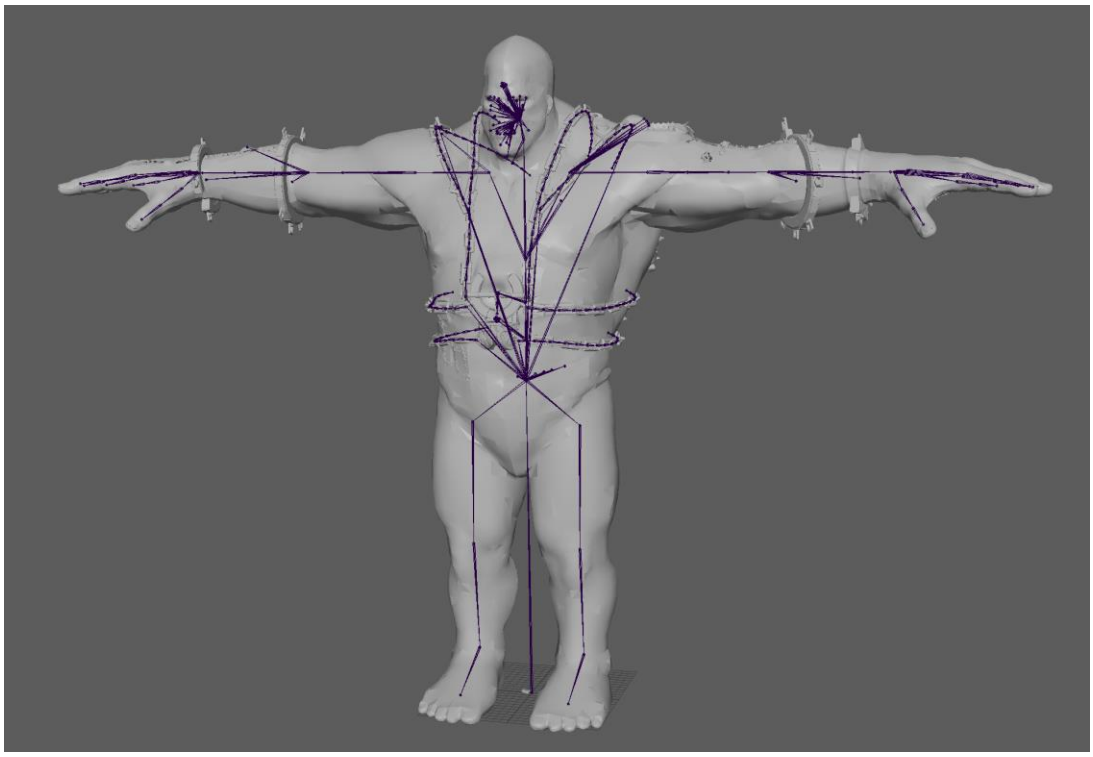

FIGURE 72. CRONOS' MODEL, AS SEEN FROM THE FRONT

Analyzing the model used in the fight makes apparent the vast number of elements behind the model's functioning. At first glance, the model is divided into several meshes for each of the titan's parts and accessories. As seen in Figure 20, the boss had to be divided into different meshes and surfaces to distinguish the navigable parts and those that are only aesthetic. As such, the model is divided into parts based on polygon density, amount of detail and animation. These details are more noticeable in the character's limbs, as seen in Figure 69.

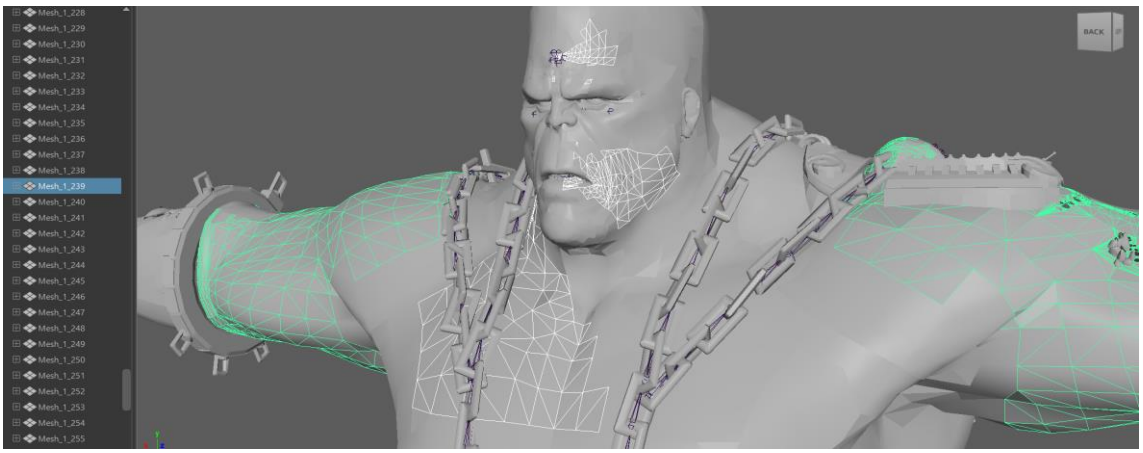

FIGURE 73. CLOSE-UP OF CRONOS WITH SOME OF THE MESHES SELECTED

The number of joints mentioned in section 2.6.1 can also be seen when importing the model inside 3D software. There is a high concentration of joints focused on the face for facial expressions and animations, the hands, as

it is one of the parts with more requirement of precise movements; and all the chains that bind Cronos to the structure in its back, for physically and procedural animations for the chains, one of the many capabilities that Maya offers.

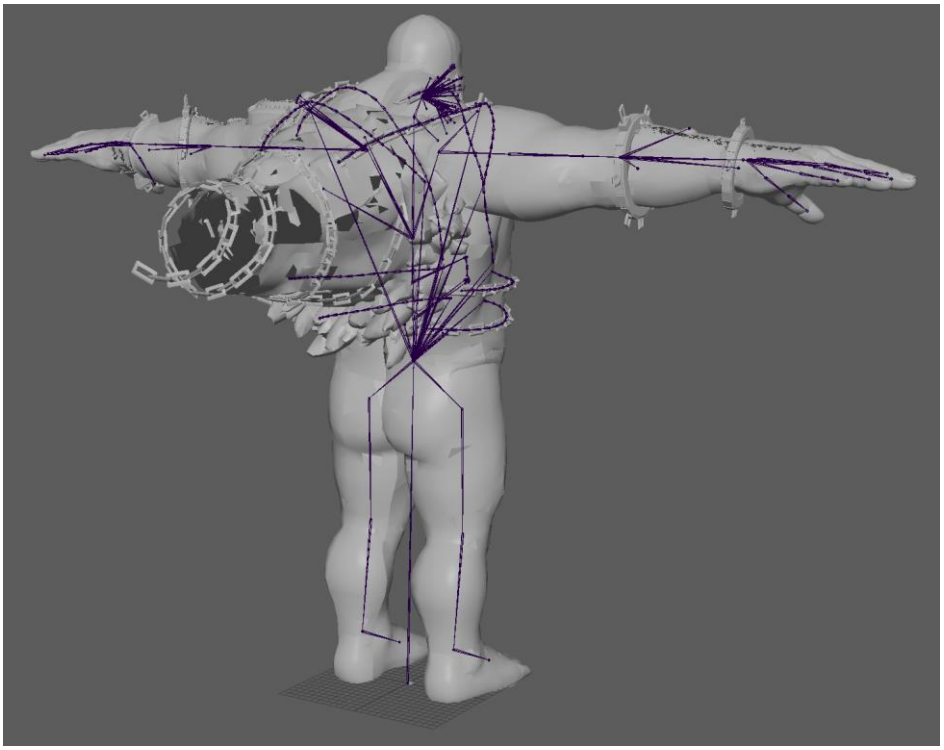

FIGURE 74. CRONOS' MODEL AND SKELETON, AS SEEN FROM THE BACK

Finally, for the textures used in the model, all meshes use four types of images for their materials: one for colouring, one for height or opacity, one for normal maps and another for properties of the material, such as its roughness and metallic properties, saved in the images in each of the colour channels of the image. This method of saving properties inside each of the RGB channels of an image is a technique that artists use to reduce the number of images loaded simultaneously in-game, which saves a considerable amount of memory inside the GPU.

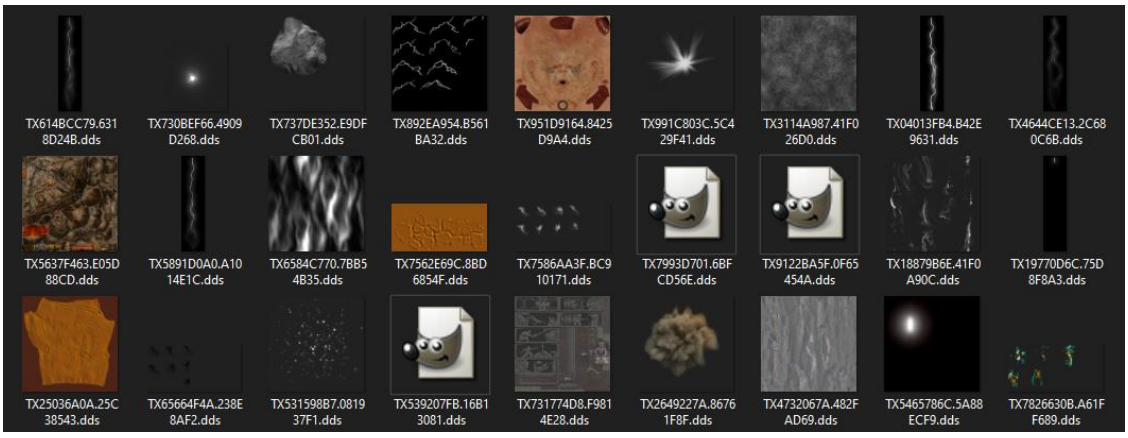

FIGURE 75. SOME OF THE TEXTURES USED IN CRONOS' MODEL

Overall, Cronos is an excellent example of how vital optimizing models is in video games. Despite the complexity behind the model and its textures, the work and expertise required to develop a model of such magnitudes is indubitably a signal of prowess for everyone involved.

FIGURE 76. RENDER OF CRONOS AS SEEN IN-GAME

2.6.2. Shadow of the Colossus: Malus & the Colossi

Shadow of the Colossus is the second title developed by Team Ico, a small Japanese game studio now disbanded and succeeded by genDESIGN, which, as explained before, was released in 2005 for the PlayStation 2. It is considered one of the best games of all time by many creditable sources[7], as it is a hidden gem amongst the vast collection of titles from the console, which, to this day, is still a topic of discussion due to the mystery and mysticism surrounding this work of art.

FIGURE 77. WANDER FIGHTING THE SECOND COLOSSUS, *SHADOW OF THE COLOSSUS*

The game tells the story of Wander, a young traveller that, armed with an ancient, presumably stolen sword and carrying a young recently deceased lady named Mono in his horse named Agro, embarks on a journey to the

[7] See Metacritic's reviews: https://www.metacritic.com/game/playstation-4/shadow-of-the-colossus

Forbidden Lands, a giant peninsula in which, guided by an entity by the name of Dormin, he will have to defeat 16 colossi to resurrect the girl.

Thus, from start to finish, this game is a boss rush involving these giant creatures as the main objective to conquer, and its mechanics, environments and overall gameplay revolve around them. However, the title presents an interesting take on level design, as it explores the verticality of a level in a way never seen before. To defeat the colossi, players have to navigate the boss' body in a means to reach their goals, as the only way to defeat them is usually in spots all across the giants' body, normally unreachable from ground level without getting crushed or vaporized if the player dares to attempt it unprepared.

2.6.2.1. Design Analysis: Game loop & mechanics

In terms of mechanics, the game uses standard phases for boss fights in its main loop, as seen in point 2.2.2, but twists them to adapt them to the gameplay and themes. The first of these phases appears on the path towards the boss. From the beginning of the game loop — usually after defeating a colossus, the player starts from the centre of the territory, called the Shrine of Worship. From there, their only guidance towards the next objective is the light they can project using their sword, which, if pointed in the right direction, will leave a trace of light that marks their next objective. Thus, the player spends a considerable chunk of the time going towards their next objective.

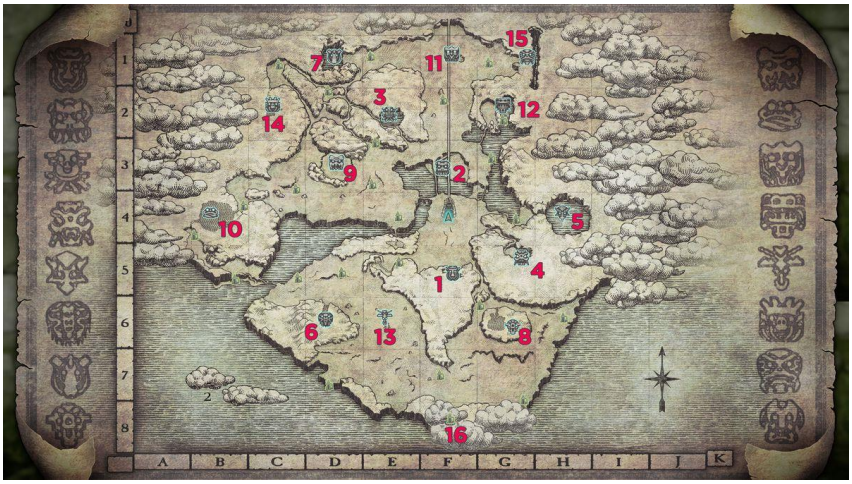

FIGURE 78. MAP OF SHADOW OF THE COLOSSUS WITH THE BOSSES' LOCATIONS AND ORDER

The fastest way to traverse the Forbidden Lands is with the help of Wander's only companion in this journey, Agro, the horse accompanying the traveller from the beginning of the game. She is usually unreliable in traversing through colossi territory, but she still feels like an unconditional companion to the player more often than not, as it is sometimes necessary to beat specific battles.

Depending on the player's playstyle, they can go straight towards the objective, which is a puzzle in itself, as there is no clear indication of where to go next unless the player uses the sword's beam in their favour. Hence, the main objective of this phase is for the player to wander and wonder: the game hides many secrets, like health improvements not mentioned anywhere in the game, as it aims for an atmospheric experience of a tiny traveller adventuring in lands not meant for them.

Eventually, they will reach enemy territory and begin the second phase of the loop, which is the study of the colossus's behaviour and surroundings. Each of the bosses follows a particular pattern with which they will attack the player or avoid him, sometimes just minding their own business as if they were not even there until the player provokes them. Their defence mechanisms come in the form of sword swings, laser attacks, electricity, shoving or stomping, which the player will have to adapt against to incapacitate the creature to strike in the many weak spots scattered throughout their bodies.

FIGURE 79. POSEIDON, THE TWELFTH COLOSSUS, ABOUT TO STRIKE WANDER WITH ELECTRICITY, *SHADOW OF THE COLOSSUS*

The player has one of two options. On the one hand, they can use some movement options such as rolling, running or climbing, or the tools at their disposal, such as the ones previously mentioned or their bow and quiver, which, while not being an offensive tool, is sometimes a way to provoke the boss and hit temporal weak spots that will help advance the fight. On the other hand, they can use environmental hazards like geysers, fire, stone platforms, tunnels and breakable towers that will advance the fight if used to the player's advantage. In any case, each colossus always aims to make players learn new ways to defeat bosses, working like a puzzle they have to solve rather than a task of raw force.

Once they figure out a way to incapacitate the giant, phase three starts with a change in music. In this phase, the player has to climb the giants to strike them in their weak spots with Wander's sword, which can be revealed by illuminating them with the sword's light beam. Like Cronos, the available path of the player differentiates itself from the stone-like obstacles in the form of the grey fur-like textures that the colossi have. To traverse it, the player has a limited amount of stamina that they can use to climb and grab onto the creature, which is the only way in which they will be able to stabilize themselves on a surface in the boss while the colossus's behaviour becomes more defensive, in the form of shake-offs that try to take out the player off their bodies to go into a more offensive position. Sometimes, they become curious about the player's behaviour, and their movements become a defensive byproduct.

Once they strike their weak spots enough, the colossus falls to the ground defeated, and Wander is teleported back to the middle of the peninsula, with a new objective tasked by Dormin.

FIGURE 80. WANDER ABOUT TO STRIKE HIS SWORD ON KNIGHT, THE THIRD COLOSSUS, *SHADOW OF THE COLOSSUS*

2.6.2.2. Aesthetic Analysis: Pacific grandiosity and hostile insignificance

From an artistic perspective, the game aims to convey a sense of solitude and atmosphere in an unknown, untouched paradise. The colour palettes of the game's environments are desaturated, and the sky is always covered in clouds. It gives out a sense of staleness, a pacific territory with landscapes blending in with one another that is being disturbed by an intruder. However, not everything is dark and lugubrious: light guides the player and offers a sense of goodness and direction in their journey, metaphorically and mechanically.

Another concept to convey is the size difference between the player and the surrounding environment. Apart from the apparent size difference between the player and the colossi, the landscape reflects a sense of vertigo. Despite being a calm landscape, the spaces between each area of the map are vast and barren, with few elements in between, often ending in giant cliffs, enormous ancient structures, tall trees and long pathways. The staging and angles the camera offers further expand the sensation of smallness, which moves the player towards the bottom corners of the screen to isolate the player and usually leaves the gigantic creatures in the majority of the player's plane of vision and uses low angles to amplify the sensation of grandiose further.

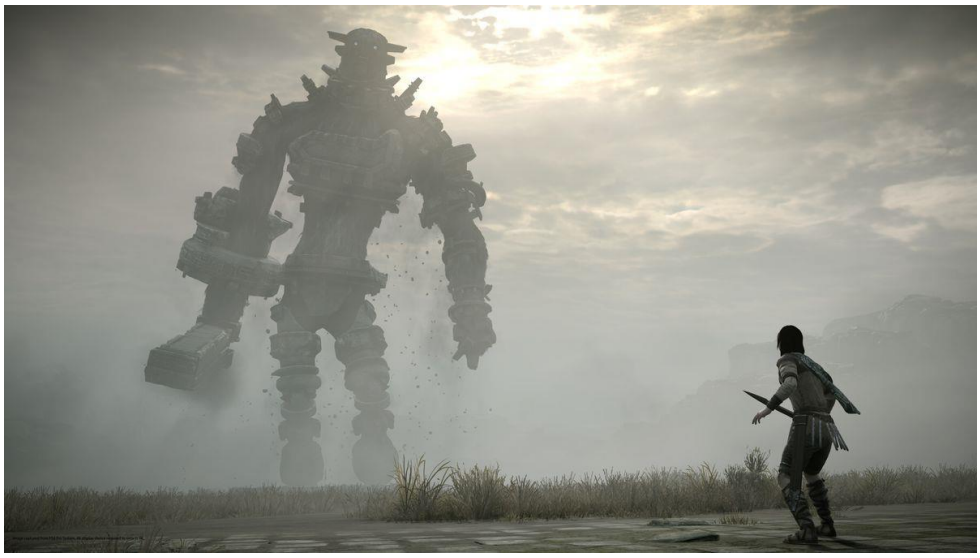

FIGURE 81. THE CAMERA'S POWER IN STAGING IN SHADOW OF THE COLOSSUS

The architecture and the rest of the civilization left across the map contribute to the game's sense of scale. They are reminiscent of and inspired by classical and middle-aged Western structures and Aztec and Mayan cultures. With the amount of overgrown nature in them, there is also a temporal distance between the player and their surroundings, offering another dimensionality of distance apart from a spatial one (Fourcade, The Art of Shadow of the Colossus (4/6): Visual Design, 2014d).

FIGURE 82. LANDSCAPES AND ARCHITECTURE, *SHADOW OF THE COLOSSUS*

Finally, in terms of music and sound design, it differentiates between each phase in the game loop. Menus have profound, booming sounds behind every selection, reflecting the colossi's influence even outside the game. During the exploration, there is no music: only sound effects like the howling of the wind and the galloping of Agro accompany the game to emphasize the loneliness of the Forbidden Lands. During combat, the music starts as an indicator of each beast's behaviour: the more aggressive they are, the more threatening the music is. As the player incapacitates and climbs the creature, the music does the same with a crescendo, leading to a climactic end from towering down the colossus.

2.6.2.3. Boss Analysis: Malus, Grandis Supernus

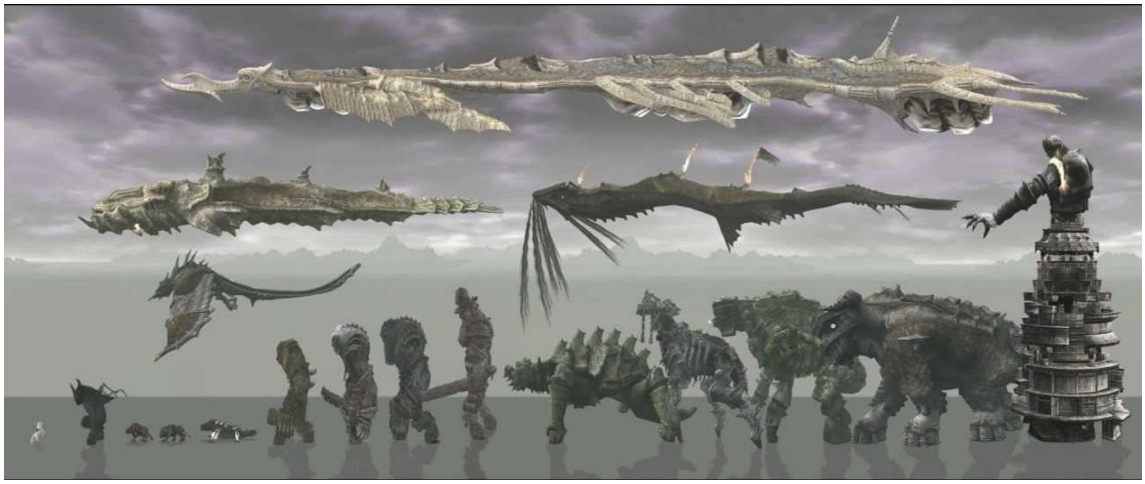

FIGURE 83. LINE-UP OF ALL BOSSES FROM SHADOW OF THE COLOSSUS, WITH MALUS BEING THE RIGHT-MOST CREATURE

This game's bosses vary in several design aspects, such as size, behaviour, animations and abilities. However, they mostly share the same mixture of animal fur delimiting the grabbable area of the body, stone from a similar architectural direction as the structures from the peninsula — which covers their face in the form of a mask with shining eyes; and even plants overgrowing on them. They can be bipedal, quadrupedal, avian or even wyrm[8]-like and appear inactive until the player arrives to disturb their peace.

More often than not, they do not seem like menaces disturbing the environment. Despite each one being unique to the world, the fact that Dormin, the entity guiding us through the quest, does not mention them by any name emphasises the mystery surrounding them and their enigmatic nature. Narratively speaking, the game implies that the player's actions and consequences are morally grey, as they seem to be killing innocent, pacific creatures for their own goals. This idea is more evident when fighting the last of the bosses and the most difficult one to navigate: Buddha, the sixteenth colossus, named by fans as Malus, Grandis Supernus.

Located in the south of the map but not accessed directly towards the south, finding a way to sort the mountains blocking the direct path is already a challenge. The only entrance leading to Malus is an ancient door blocked if the player has not defeated the 15 previous bosses yet. The way towards the boss is narratively significant, as it is when Agro sacrifices herself to save Wander, leaving him weak and alone at the end of everything. As he approaches the scenery, the most significant change in the environment appears: the light accompanying him throughout the journey now comes from an aggressive and threatening thunderstorm, which darkens the scene and makes the boss the most straightforward light source.

[8] Wyrm: Type of dragon caracterized by not having any kind of limbs

FIGURE 84. MALUS, AS SEEN FROM THE BOTTOM, *SHADOW OF THE COLOSSUS*

The boss is located at the end of a series of walls and tunnels carefully positioned for the player to traverse stealthily. It uses one of the least used mechanics of navigation: rolling, which tends to leave the player vulnerable if not carefully planned, as the boss will attack with a deadly laser each time he sees an opening towards the player, which is usually hard to avoid and have few opening windows. As Wander approaches it, the true nature of this creature reveals itself: the colossus is a half-organic torso and a half-living fortress.

The living fortress part is the easiest to traverse, as it works as both a skirt/kilt for the boss and a castle-like structure the player can use to climb. Once the player arrives at the back, the player must hit a series of hitpoints that the boss will react accordingly in pain, as it moves its hands all over his body for Wander to go further and further upwards in his weak point. As the player climbs, the boss's behaviour that used to be defensive fear suddenly becomes curiosity but then turns back to fear as soon as the protagonist approaches the most prominent weak spot at the top of its head, ending the adventure with an empty feeling of success.

Malus is possibly the greatest challenge Shadow of the Colossus has to offer, as it challenges the player to use all of the mechanics they have learned throughout the journey and serves as the culmination of the most core aspects of the game's design and aesthetics. Many lessons can be learned from this game, as it pushes the boss rush genre to new horizons that few games have tried to replicate while still offering an atmospheric piece of artistic and technical endeavour to the public.

2.6.2.4. Model Analysis: The complexity of a Colossus

For the proper viewing of models inside Shadow of the Colossus, the author has used a program called "SotC-Viewer" by Matheus Garbelini and additional notes by Léna Piquet, Senior Product Manager and Technical Artist at Adobe. This program allows one to extract data from a pack and visualize its contents in a 3D viewer. The analysis done in this project corresponds to the models found in the PS2 and PS3 versions of the game, which means that they will not adhere to the models found in future game remakes.

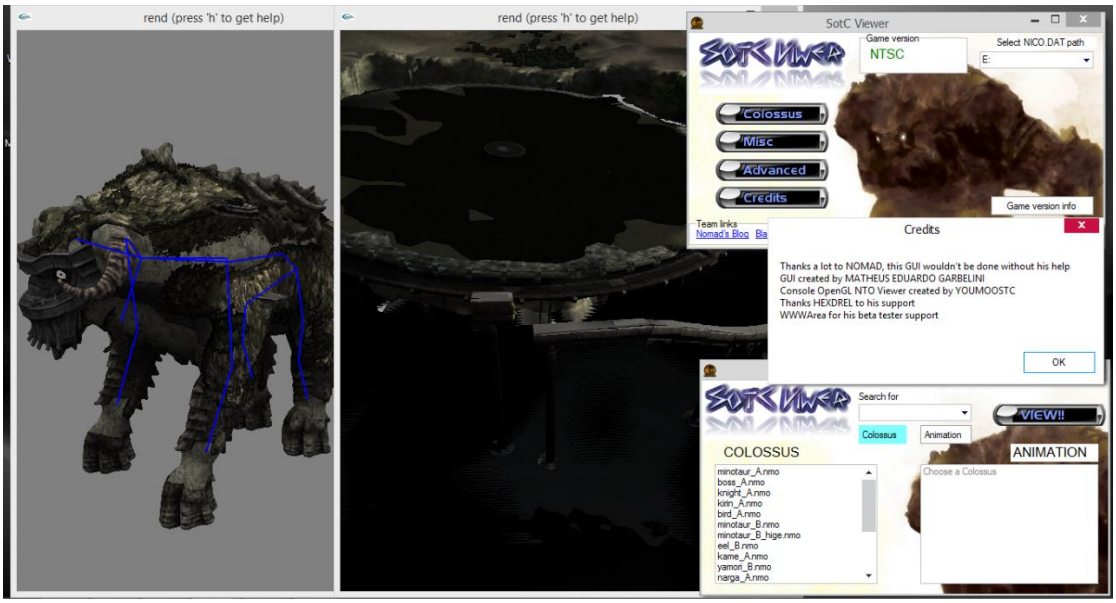

FIGURE 85. SOTC VIEWER'S INTERFACE, WITH ONE OF THE COLOSSI RENDERED

At first glance, it is apparent that all models are primarily low-poly with simple skeletons for the animations. They had very low polycounts based on the size of the monsters, oscillating between 6k and 21k triangles (or 3k and 10k polygons, respectively). The number of polygons meant that some stretching was expected in most models when these were in movement. Still, despite these drawbacks, the colossi were already heavy, incredibly detailed models to be loaded at runtime for the console at the time. Some parts of the colossus had to be entirely separated meshes that would then adhere to the colossi in the engine, ranging from their weapons to their heads.

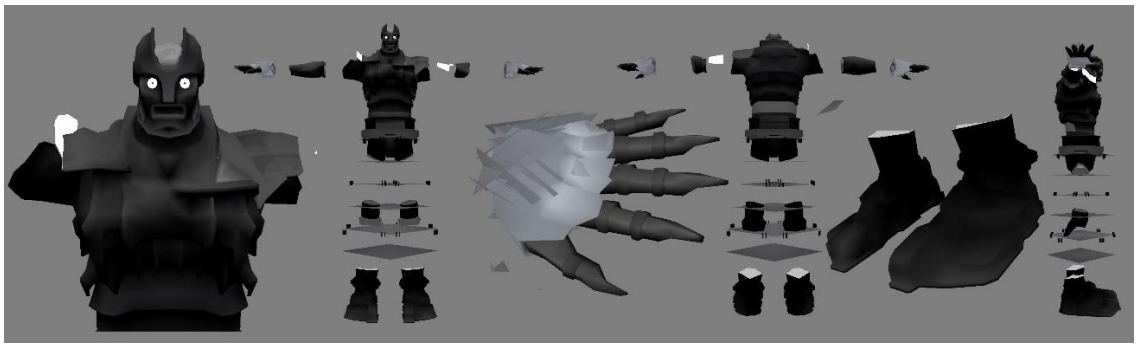

FIGURE 86. BUDDHA'S CHARACTER MODEL, WITHOUT THE WALLS FROM ITS SKIRT STRUCTURE

The console's limitations meant artists could not create detailed meshes for the bosses. This setback meant that the model's textures did most of the heavy lifting, something very typical for the era. The colossi use textures ranging from 16x16 pixels to 128x128 pixels, mainly consisting of either albedo textures (only containing colour information) or black-and-white textures for images that contain transparent parts, like fur. Each polygon would then contain either a part or the entirety of these textures displayed on it.

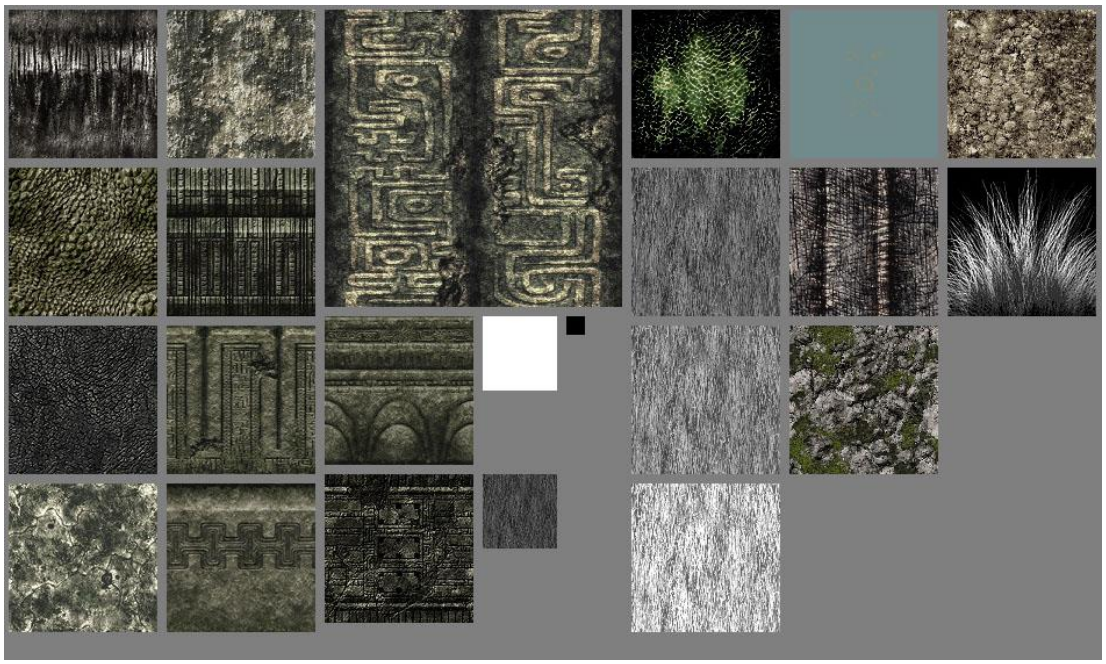

FIGURE 87. KAME_A'S TEXTURES, RANGING THROUGH ALL SIZES

Speaking of which, one of the most characteristic attributes of these creatures is their furry-like textures. Shadow of the Colossus's hair is one of the most curious and mesmerising cases of rendering a fibre-like texture from the era, as it was impossible to generate natural fibres like nowadays. In order to simulate hair, the artists used two methods. The first one involved creating several polygons containing an image of the hair, which, when grouped, would generate hair tufts.

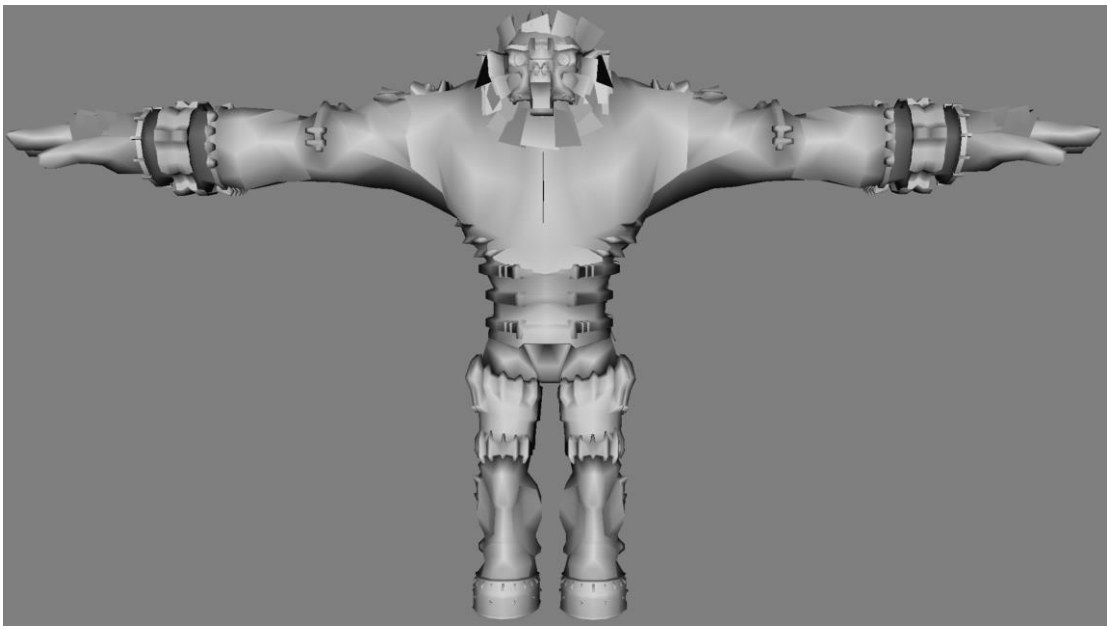

FIGURE 88. MINOTAUR_B'S MODEL, WITH THE TUFTS SEEN ON THE FACE

The second and most intriguing is the usage of polygon layers spaced with a distance no more prominent than the player's model's hand. As seen in Figure 87, each colossus used three textures of different opacity that,

when combined in layered succession — usually two layers per map, could replicate a sense of depth and puffy hair. The hair's colour would then be obtained thanks to the colour information in each vertex — also known as vertex colour (Piquet, 2012).

FIGURE 89. MINOTAUR_A'S MODEL WITH THE FUR TECHNIQUE IN ACTION.

For the flow and hair direction, the artists used the model's UVs and polygon placement to their favour. To create symmetry between both sides of the colossi, the UVs of one of these were created first so that the other half would need to have them inverted. To create varied textures and dynamic flow, the layers shifted in position ever so slightly so that the hair would generate a lean towards one specific direction.

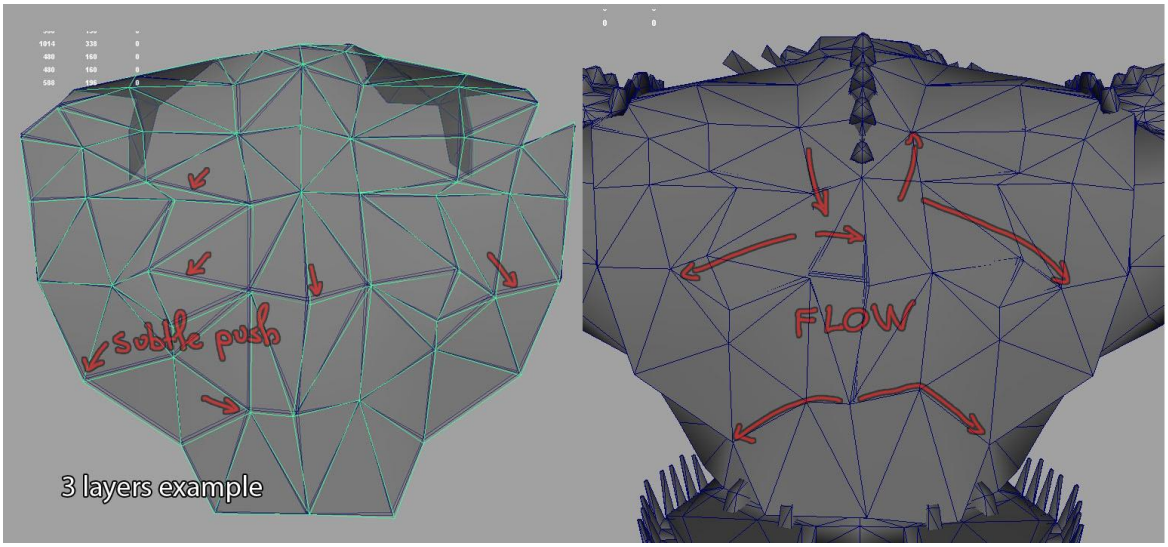

FIGURE 90. EXAMPLE OF SOME LAYERS BEING STACKED ON TOP OF EACH OTHER WITH A SLIGHT SHIFT IN POSITION

Finally, regarding the animations, the models do not use a significant number of bones in their skeletons, as they are primarily quadrupeds or humanoids. Their skeletons are very simple, ranging from 19 for the simplest to 114 for the more humongous ones like Snake_C. The number of bones is also a factor to consider as one of the reasons for stretching in most models when they are moving.

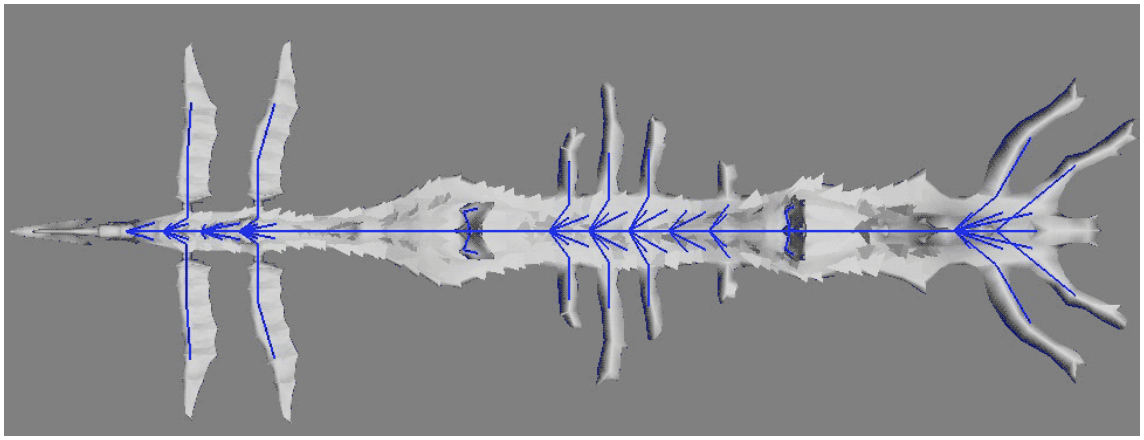

FIGURE 91. SNAKE_C'S MODEL AND SKELETON

Overall, these models exemplify the technical prowess that the technical artists at Team Ico showcased to fight against the era's consoles' limitations. Despite the difficulties and challenges, it presents techniques still used today, like the methods seen when creating fur. In the words of Lena, "It is a magnificent achievement and a good example to follow" (Piquet, 2012).

2.6.3. Solar Ash: The Remnants

Solar Ash is a game developed by Heart Machine, a video game company primarily known for their previous title, Hyper Light Drifter. Released on the 2nd of December 2021, it is a platformer action game that mixes parts of exploration with the fighting of giant enemy creatures. It is very similar to Shadow of the Colossus in its game loop structure; however, rather than focusing on the bosses themselves, Solar Ash's main mechanics revolve around the flow of movement and the environmental storytelling of its scenery.

FIGURE 92. SOLAR ASH PROMOTIONAL ART

The game follows the story of Rei, a Voidrunner trying to save her home planet from the Ultravoid, a black hole sucking away the remnants of what once was a place where she lived with the rest of the crew members of her fleet. On her way, she will have to release the world from a mysterious and intrusive hot miasma and fight gigantic, grotesque creatures that work as the bosses of each area, which she must destroy to activate the machine and return her home to its former glory.

2.6.3.1. Design Analysis: The sense of Flow

To navigate the environment, the player can perform several actions such as running, jumping, double jumping and triple-hit instant multidirectional combo attacks. The environments adapt to such movement so that the player can slide and dash to instantly achieve high speeds with movement similar to skating with rollerblades, with grind surfing in many parts of each level.

FIGURE 93. SCREENSHOT OF SOLAR ASH SHOWCASING THE SLIDING MECHANICS OF THE GAME

Other mechanics related to the boss fights in this game are a whip that players can use to swing to specific points usually unreachable with basic movements and timeslips, a focus-based mechanic used to stop time to reposition oneself or reach points further away from the grappling hook's average reach distance.

FIGURE 94. SCREENSHOT OF SOLAR ASH DEPICTING REI USING HER GRAPPLING HOOK

These actions offer the player a sense of flow-like motion that is incredibly satisfying to manoeuvre through the game's levels. The pacing is never interrupted by awkward spaces and platforming, as most of the environments are usually set in open spaces that sometimes even change gravitational pulls, depending on the area. It feels fast while also not overwhelming or complicated to keep up with, allowing for an exquisite game like none other.

2.6.3.2. Aesthetic Analysis: Neon & Synthwave

The art of the game is visually engaging and exciting. The colour palettes abuse the usage of secondary, complementary palettes in most of its settings, with models using plain colours without much texture, always connecting in a very organic way. This contrast is heavily reflected between the antagonist and the protagonist, whose colour palettes complement one another.

FIGURE 95. ECHO, THE ANTAGONIST OF SOLAR ASH, HOLDING REI IN HER HAND

The general artistic direction of the game falls in the category of low-poly synth-wave neon aesthetics. Light is heavily present throughout the game as a source of contrast and guidance for the player. The mournful art and storytelling are reminiscent of artistic movements such as retro-futurism and borrow heavily from Jack Vance's Dying Earth series, whose architectural design can be seen in many game environments.

FIGURE 96. SCREENSHOT OF THE BROKEN CAPITAL, ONE OF THE LOCATIONS IN SOLAR ASH

The protagonist's wavy lines and shape language help convey a sense of movement and flow in her design, which often contrasts with the blocky nature of the environments and the monster Rei fights against throughout the journey. It also separates the protagonist from the rest of the world and helps her be easily readable in the camera view, which is fast and responsive to the players' movements (as opposed to the orthopedical controls of Shadow of the Colossus, for example).

Overall, the game's use of contrasts allows it to become a visual experience that gets players in a hyper-focused mindset while not getting too eye-straining or overbearing. It also helps the visual language of the game's design elements by differentiating danger from navigable platforms, creating a trance-like experience that helps to convey dynamism and action.

2.6.3.3. Boss Analysis: Test of reflexes

Each area of the game is guarded by a giant creature known as a Remnant, formed by bone-like chunks of matter over a slimy, slick and slender black mass that, when interacted with, becomes extremely hot exceptionally quickly. They seem to be influenced by other media, like 80's anime such as Neon Genesis Evangelion's Angels, which had the same organic yet alien structure that came in different shapes and forms. After liberating the zone from a pulsating miasma that seems connected to the Remnant of that area, they become active and start wandering around, looking to strike the heroine at their first opportunity.

FIGURE 97. THE WITHERED EYE, ONE OF THE MANY BOSSES IN SOLAR ASH

After avoiding their attack, the player can grapple onto one of its limbs and start traversing its body while they are still moving. The objective is to hit one of the creature's many eyes by jumping between the blocky bone-like structures while avoiding the hot black miasma and hitting different objectives that work as small checkpoints. These checkpoints reset a hidden timer in the form of the creature's ability to become a mass of instant death, which keeps the player on their toes towards one of the many eyes they have to strike down. Rei has to hit these targets three times per boss, each time making the boss's body lose some of its bone structure and ultimately altering the layout the next time the player has to traverse, giving a sense of weakening while increasing the difficulty.

FIGURE 98. REI JUMPING ACROSS ONE OF THE BOSSES IN SOLAR ASH

Overall, Solar Ash's boss design is simple yet challenging and fun, becoming a challenge of agility, skill and reflexes that increases in difficulty over time. However, they are not an archetypical boss fight, but rather an elaborated timed obstacle course level, as they do not have significant attack or defence patterns throughout the fight that change as it progresses. Nevertheless, it is a demonstration that it is not required to have unique mechanics to create memorable boss fights.

2.6.4. Super Mario Galaxy: Megaleg

Super Mario Galaxy is a platformer game and one of the many entries for the Nintendo Wii, released in 2007 by Nintendo. In this instalment of the Super Mario Bros. series, players will have to embark on an adventure in outer space, where they will fly across varied galaxies, constellations and comets to save Princess Peach from the ferocious claws of King Bowser. The game's main gimmick is that Mario can use the gravitational pull of the planets to his advantage, rolling around them, going up walls, and even sticking to ceilings with his feet. With these abilities, players must collect stars to power up a ship and rescue the princess.

FIGURE 99. MEGALEG, LOOMING OVER MARIO AND THE TINY MOON ON WHICH THEY ARE STANDING, *SUPER MARIO GALAXY*

The game's first main boss is Megaleg, a three-legged mechanical construct commanded by Bowser Jr. with a face reminiscent of a past enemy from the franchise — the Snifit. Multiple gears, pistons and cannons form the machine's body, all intended to crush and stomp Mario on the mechanical moon where the battle occurs. However, as the legs keep hitting the ground, the player will notice some arrows indicating that the legs are climbable and divided into two sections: one with bullet bill cannons blocking the way and gears turning that can make Mario fall off the creature. As they reach the head, the player will finally have to guide bullet bills to the top of the head, where the core of the machine stands, to break the reinforcements and barriers blocking it from being free.

FIGURE 100. MARIO REDIRECTS A BULLET BILL TO HIT THE CORE OF MEGALEG, SUPER MARIO GALAXY

Overall, the boss is a simple test for the game's first hours; thus, it is not very difficult to defeat per se. However, the camera angles, the size differences between Mario, the moon and the monster and the mechanical sound design of the boss truly sell the experience of overcoming and defeating a living robotic fortress.

2.6.5. Rayman Origins: Big Mama

Rayman Origins is the fourth instalment of the main Rayman series, released on the 15th of November 2011 and developed by Ubisoft. After years of spinoffs in the Raving Rabbids series of questionable success, the studio returned to the main formula of 2D platformer games, with massive public acclaim and a fresh take on the formula that had never been done since 2003.

Regarding its bosses, Rayman is not known to have especially memorable boss fights per se. Design-wise, they are crudely hand-crafted goofy abominations of nature that serve as the main obstacles in our protagonist's way, with simple mechanics and attack patterns as is common in these games. However, one example worth mentioning is that of Big Mama — also known as Voodoo Mama — which is an optional boss fight at the end

of the game. She is a nymph in the Land of the Livid Dead; however, when she first appears, she looks like a gigantic Lovecraftian pink monster.

FIGURE 101. SCREENSHOT OF VOODOO MAMA AS SEEN IN THE VIDEOGAME RAYMAN LEGENDS[9]

The objective is to burst open pimps appearing in her body with a hit, making her squirm and Rayman jump into her face, where her real weak spots are. To achieve this, Rayman must use her arms to his advantage as they work as giant platforms that elevate and rotate all around her. At the same time, several environmental hazards in the form of lava at the bottom, bracelets too big for her bone-like arms that are affected by gravity, thus moving constantly across them; and spikes both on her elbows and hands, marking the limits that Rayman can navigate in.

With the usage of his attacks, gliding and jumping abilities, players must test their skills in an ever-changing environment that, while being reasonably simple mechanically, is an effective way to create and embody a giant boss in a simple game mechanically speaking.

2.6.6. Monster Hunter World: Zorah Magdaros

Monster Hunter: World is the second-to-last instalment of the Monster Hunter franchise, released on the 26th of January 2018 for the PlayStation 4 and XBOX One and developed by Capcom. It is an action RPG from the boss rush genre whose main objective is to defeat and hunt giant creatures distributed worldwide for studying purposes. The player can create and use different weapons and armour to fight these monsters in their preferred playstyle, manipulating the environment to their advantage and farming them to obtain better gear and abilities. What is revolutionary about this entry in the saga is that it is a complete overhaul of the formula and the first one offering online capabilities.

[9] In a later release of the sequel, Rayman Legends, the player has the possibility of reviving the battles from Origins as part of its extra content.

The artistic design of the game's monsters is very reptilian-based and prehistoric, with creatures reminiscent of ancient Jurassic dinosaurs filled with spikes, hard plaques and long tails. They signify a menace to the creatures surrounding the ecosystem, and the most prevalent examples are usually apex predators in the food chains of the local fauna. Mechanically speaking, each of them has its methods of defeating: some can be destroyed using brute force, but others do not have to be killed but rather be ripped off their armour, scales, wings or other valuable parts to have a profitable reward by using elements in the environment or the creature's behaviour to the hunter's advantage.

FIGURE 102. EXAMPLES OF MONSTERS FROM THE MONSTER HUNTER SERIES

At the end of the game, the player has to stop the advancement of a colossal monster known as Zorah Magdaros towards the main home settlement. Just like Cronos from God of War 3, this beast of colossal magnitudes was a development challenge for the team at Capcom, lasting one year of its development in only this sequence alone and being a spectacular showcase of the artistic and technical prowess of the generation's console capabilities. However, compared to the design of other monsters in the game, this battle is considered one, if not the most underwhelming final boss fights in the franchise. It is a siege assault with not many mechanical complications that change the game's main loop of battles against monsters, as it uses mechanics unique to the battle, which, in turn, led to many design overlooks.

FIGURE 103. ZORAH MAGDAROS, DESTROYING EVERYTHING IN ITS PATH, *MONSTER HUNTER: WORLD*

The battle is divided into two phases. In the first one, the protagonist is suggested, not obligated, to take out three flaming cores located on Zorah's surface, which consists of a changing terrain with lava in between that does not offer a challenge on itself. The player also has the option to throw stalactites at the creature, but these do not affect the creature's health in the slightest; they also have the option to fight Nergigante, another well-known monster from the game that, while it is a good contrast of size to compare with the colossus, it is entirely optional, as players can avoid it due to the vast size of Magdaros. Hence, this phase becomes a tediously fixed timer and a glorified mining level in an unappealing, annoying terrain that becomes a nuisance rather than a spectacle.

FIGURE 104. NERGIGANTE ON TOP OF ZORAH MAGDAROS, CAMOUFLAGED BY THE ENVIRONMENT, *MONSTER HUNTER: WORLD*

In the second one, the player is placed on a giant bridge with cannons and ammo, and they are supposed to manage the constant barrage of damage that will impede the advancement of the beast by using cannons spread throughout the bridge. However, Zorah Magdaros does not feel like a threat, as it advances incredibly slowly and cannot destroy a giant wooden barrier, erasing any sense of urgency in this phase. This missing engagement makes the entire experience a dragging, pointless one.

This battle is an example of overproducing a boss and placing it in a game unprepared to handle these battles regarding camera placement and texture design. Changing an already well-established formula for a specific battle, especially the last one, is one of the worst things a designer can do, as it clashes with everything the player has learnt in their adventure, and it usually leads to terrible playing experiences if not adequately prepared. Despite the long production time and the technical spectacle it promised, the absence of threat and the dragging glorified timers make this experience underwhelming.

3. Project Management

In this section, the student has reunited all the most efficient tools used for correctly managing the final product's procedures and validations, as well as an analysis of possible risks, threats and potential solutions for them to save and optimize the workload over time. This analysis aims to ensure excellent quality for the project while keeping a healthy and realistic workflow.

3.1. Follow-up Tools & Procedures

Firstly, based on different methodologies used for time optimization in many video game companies and the knowledge of the student on the usage of such, here are the tools used by the author for many of the functions crucial for the project:

3.1.1. Work Diary: Discord

Discord is an instant messaging application similar to Slack and Team Speak. It uses several text and voice chats spread across servers to communicate with friends, teammates and coworkers alike. What differentiates it from other applications is that, apart from having most of its services for free, it has many functionalities oriented towards video game usage, which, at first glance, would not benefit the project.

However, the student has created a private server that, using several chat channels, has served as a way to record all the actions and tasks completed throughout the project by sending a message to the chat board at the end of each working day. Since the messages indicate the time the author sent them, it works as an automatic recording board, allowing for the retrieval of images and notes recorded on a specified day, thus allowing the recording of the information quickly in this study.

To save these chats in the cloud, the author used a plugin in Google Chrome that allows exporting chat logs from Discord in an HTML format, which can then be opened in any navigator. This chat exporter saves messages and media sent, so it is a reliable contingency plan in case of information loss due to external factors.

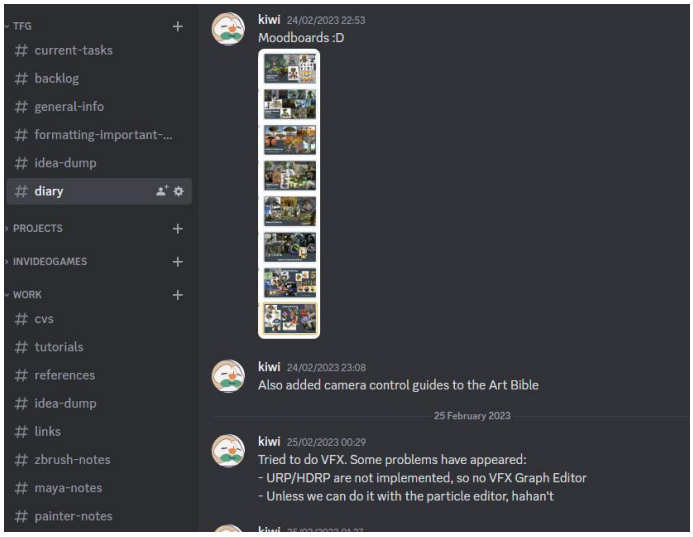

FIGURE 105. DISCORD DIARY RETRIEVAL

3.1.2. Agile Methodologies: Scrum & Kanban with Trello

In the video game industry, especially in smaller teams, there has been a rise in the usage of Agile methodologies, which have switched from a waterfall-like work structure to a more cyclical one, in which developers test and revise features as they do them. Two of these methodologies are Scrum and Kanban.

On the one hand, Scrum refers to the division of work into smaller tasks spread throughout the weeks and milestones of a project and classified by the amount of time required to complete them. These are then stored in a backlog revised by the Scrum master, who is in charge of selecting and creating short periods in which a selection of tasks workers must complete, called sprints. This process allows teams to analyse and focus the work on specific needed parts at a time.

On the other hand, Kanban is a funnelling method that filters what tasks are being worked on at the moment, by whom and their status. It allows scrum masters to manage the work in progress and to be done in a sprint while also capping the number of tasks simultaneously, thus creating more efficient and focused work.

A great visual tool that allows the visualization of Kanban tables is Trello while also being able to categorize tasks with tags and create deadlines and checklists. It is an indispensable tool that helps visually represent the project's advancements and achievements by dividing each task into columns based on the status of such, the level of urgency and its type, which can be either programming, design, research or art.

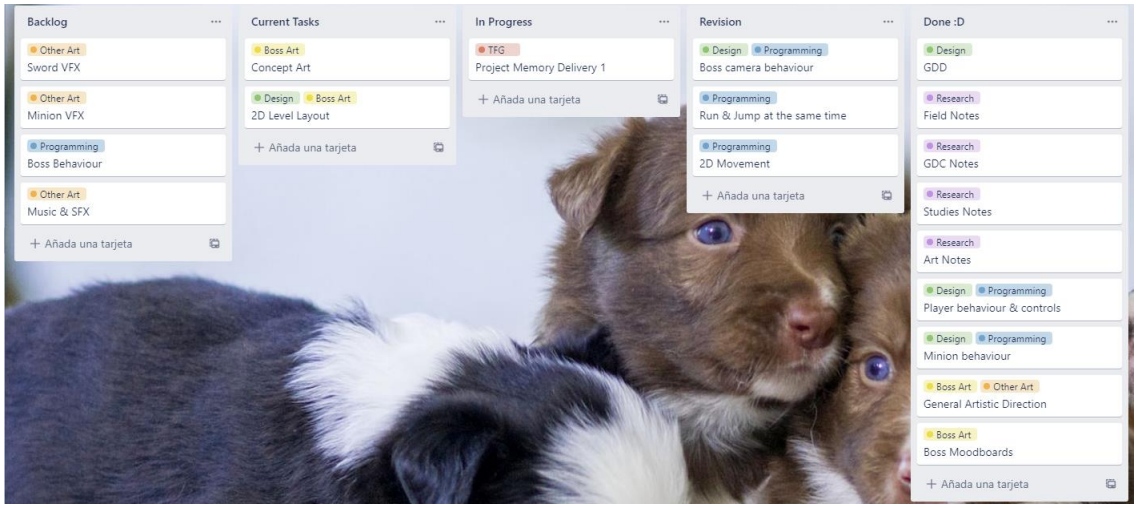

FIGURE 106. TRELLO TABLE RETRIEVAL

3.1.3. Cloud Storage & Version Control: Google Drive & GitHub

Finally, the author has used another two software to save the progress, documents and backups of every asset, document and log of this project. One of the many services that Google offers users is Google Drive, a cloud storage service that allows for creating and saving files. It has been instrumental in managing version control of backups, documents and such, as it also contains a log of all the changes and versions each of the files saved has passed through, which has helped record the production process.

Nombre	Propietario	Última modificac...	Tamaño de archivo
References	yo	1 mar 2023	–
Deliveries	yo	1 mar 2023	–
TFG Proposition	yo	3 jun 2022	5 kB
Notes (For State of Art)	yo	3 mar 2023	2,3 MB
GDD	yo	1 mar 2023	1,1 MB
Diary Backup.html	yo	18:46	55 kB
ArtBible	yo	3 mar 2023	213,7 MB

FIGURE 107. GOOGLE DRIVE DOCUMENTS RETRIEVAL

Another service to store the project's data is GitHub, another cloud service developers use to save substantial project repositories related to software creation, like video games. Usually, programs create meta files and other types of temporal files irrelevant for correct project preservation. GitHub allows programmers to filter these files and commit and push[10] all the necessary assets and changes to pre-existing assets and scripts. It also keeps up a version history of all the states the project has had, allowing to fetch and pull[11] the project into other PCs to work outside the central working station and the possibility of discarding changes of last pushes by reverting to a previous version in history. It has served as a vital tool for the version control of the prototype in Unity.

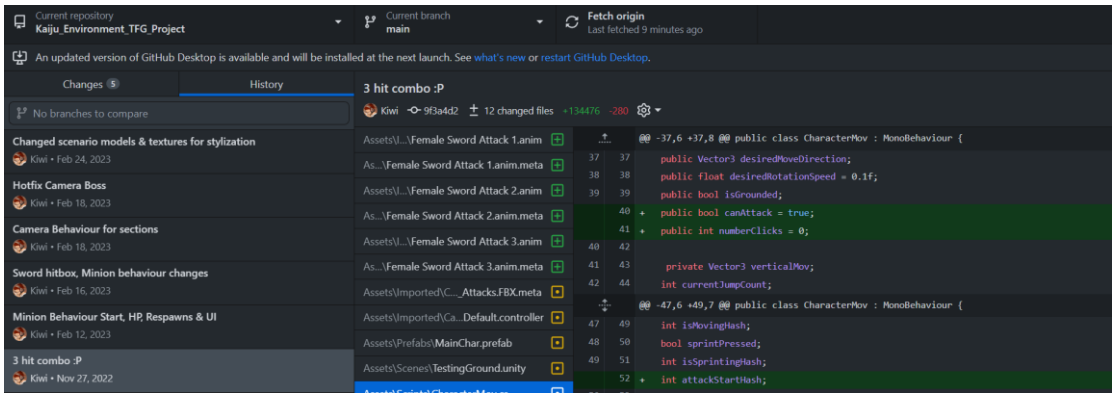

FIGURE 108. GITHUB HISTORY RETRIEVAL

3.1.4. Time Management: Gantt Chart

For an accurate analysis of the project's milestones, objectives and time required for each process, chronograms in the form of a Gantt Chart were created using a free Gantt Chart creator[12]. This type of diagram allowed the author to line up all the tasks involved in this project and what time will it take to complete them all. It also checks what percentile of work has been completed for each task, relates each task with the next one by connecting them with arrows and allows to add annotations to each assignment, characteristics similar to Trello.

[10] Commit & Push: Git commands that allow to save changes into the cloud repository
[11] Fetch & Pull: Git commands that allow to retrieve the current state of a repository and download it onto a computer
[12] Gantt chart creator found in https://www.onlinegantt.com/#/gantt

Since this project has many uncontrollable risks, the writer has created multiple iterations of a Gantt chart to better adapt the project to the time constraints.

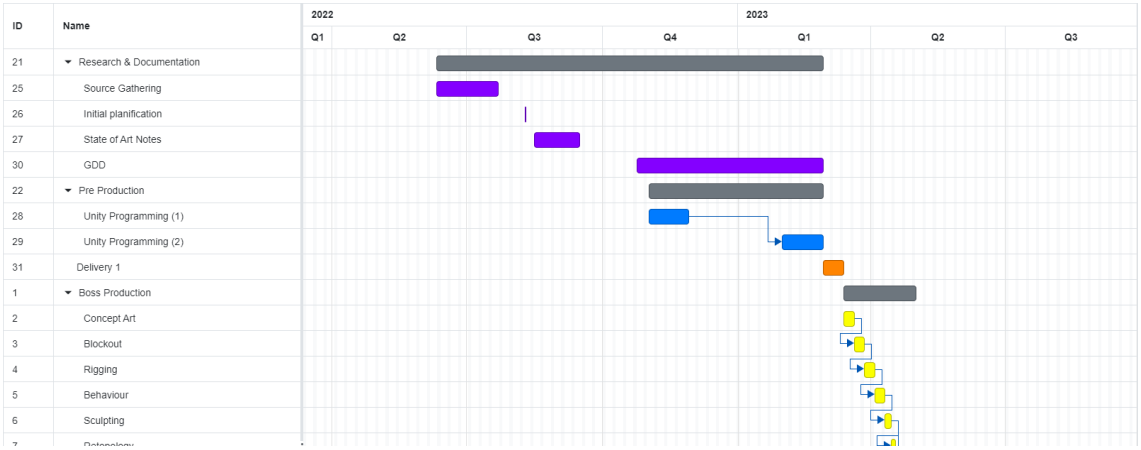

FIGURE 109. EXAMPLE OF THE EARLIEST GANTT CHART RELATED TO THE PROJECT

3.1.5. Validation: Playtesting sessions

Finally, to ensure the project achieves its purpose perfectly, at least two playtesting sessions will be conducted to test the prototype's state. Each of these will test the boss fights in different development parts — the first to improve on the first iterations created for each phase and the second to fix undetected bugs and improve on the final result with minor additions — and, if possible, with different publics, as it will serve to recollect information with fresh eyes.

The methods of collecting information in these sessions will include recording all user comments, observing their reactions, and, most importantly, using Unity scripts developed by the user to record heatmaps of the user's actions based on position. These heatmaps will generate information for when they attack, jump, move and die, allowing the author to fix issues related to high-difficulty spikes, points of unfair deaths, and other mistakes not found during development.

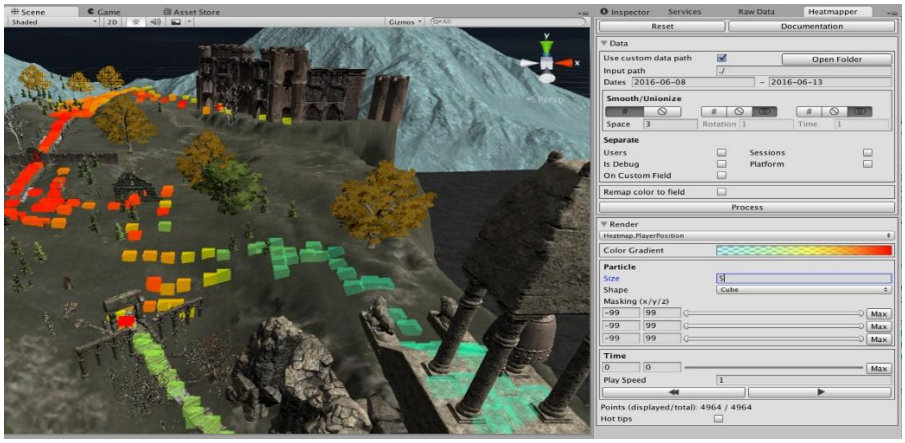

FIGURE 110. EXAMPLE OF HEATMAP GENERATED FOR A 3D LEVEL SIMILAR TO THE ONE USED IN THIS PROJECT

3.2. SWOT

	Positive	Negative
Intrinsic origin	Strengths	Weaknesses
	Preexisting knowledge of the degree related to design, programming and 3D asset creation. Willingness and inspiration to create.	Neuroticism and feelings of overwhelmingness for a project of this scope. Perfectionism for insignificant details. Lack of expertise in areas such as animation and concept art.
Extrinsic origin	Opportunities	Threats
	A new theme in the field of study of video games. There is much-existing documentation, guidelines and support from experts.	Regarding professional diffusion, high competencies at junior levels with other degrees.

TABLE 1. SWOT ANALYSIS

3.3. Risks & Contingency plan

Risks	Solutions
Other elements such as programming, design and assets other than the boss itself make the project feel unfocused or jeopardize its development	Reduce other aspects unrelated to art to the simplest necessary minimums for the boss to be functional while not sacrificing creativity.
Other assets such as environment and character models, animations, VFX and SFX are needed to have a working and presentable prototype.	Download and use assets from accessible sources on websites like the Unity asset store. Use websites such as Mixamo for extra animations required.
The prototype suffers from performance issues due to a high quantity of loaded models or poor optimization.	Reduce the number of assets used in the environment or use automatic optimization processes for 3D models.
The main PC used for the project becomes unusable or has a technical issue.	For redacting this document, use other computers available at home. For developing the prototype and models, use the computers at the university or virtual machines.
Partial or complete loss of files related to the project due to corruption or malfunction of hardware or software.	Use a GitHub repository to save the prototype's data. Create daily security backups in the main working PC and cloud services like Google Drive.
Work weight becomes too big for the final delivery in June.	Deliver the project in September or opt for one of the other plans in the time analysis.
Times stipulated in Gantt charts are either estimated to be too long or short for the part of the process in question.	If the task is too short, advance to the next step. If it is too long, opt to cut time from other parts not deemed vital for the final presentation.

TABLE 2. RISKS & CONTINGENCY PLAN ANALYSIS

3.4. Initial Costs Analysis

Any assets showcased and used for this project have been downloaded licitly for free. Based on the analysis made for other products, this project's complexity would require a handful of experts to create, which, depending on the team size — in this case, indie- may vary the salaries. Considering the specs of the project, that it lasts from June of the previous year to June of this year with a total of approximately 450 hours spread across approximately nine working months, and that it is taking place in Spain, the initial costs of the project are the following:

Software Licenses			
Name	**Cost (Monthly)**	**Usage time**	**Total**
Autodesk Maya	279€	4 months	1,116 €
ZBrush	43.05€	4 months	172.2 €
Substance Painter	49.99€	4 months	199.96 €
Adobe Creative Cloud	62.99€ (19.66€ for students)	4 months	251.96 € (78.64 € for students)
		Total	1740.12 €
Salaries			
Career	**Cost (yearly)**	**Time of development**	**Total**
Concept Artist	17K €	1 month	1,416 €
3D Artist	26.8K €	2 months	4,467 €
3D Animator	19.1K €	1 month	1,591 €
Lighting Composer	26K €	1 month	2,167 €
Designer	22K €	2 months	3,667 €
Programmer	21K €	4 months	7,000 €
		Total	20,308€

TABLE 3. INITIAL COST ANALYSIS OF SOFTWARE AND WORKFORCE

4. Methodology

For this project, the methodology used for the artistic part will be the most common process for character creation in the video game industry, divided into three phases, all with necessary procedures and tasks dependent on one another. This way of working is called an incremental methodology, which means that each of the tasks done uses the results of previous ones to work.

4.1. Pre-production

Before starting with the modelling software, since bosses are meant to be tests of skill for the abilities learned by the player throughout a game's phase, there must be game mechanics and pillars the student has to test in the first place. Hence, the first part of the project is to define and create such mechanics to base our boss on and program them in a game engine.

For this project, the student has chosen Unity as his engine of choice, as it is the one with which he is more capable and knowledgeable. Even though Unreal Engine would be the one that ensured better results to create a proper prototype, the programming tools at his disposal required knowledge that would slow the production of the focus matter of the project, so the option was discarded very early on in the development.

After these mechanics, a Game Design Document is generated after a template found in Level Up!'s Bonus Level 8 (Rogers, 2014), modified and adapted for this project, in which all the aspects related to the boss are specified and enumerated. The document is not to be kept still throughout the project, as future steps of the process will modify it significantly; rather, it is meant to be a specification of guidelines upon which the boss will be created, which roughly include a description of the boss, the narrative behind it, some key aspects that had to be respected at all times, the arena in which the boss battle is going to take place, the required animations & SFX for the boss and the battle's phases and transitions, as well as the level design of the navigable parts of the boss itself.

In tandem, another document is used to define the artistic direction of the project: the Art Bible. In this document, all the information related to how the game's general aesthetics are going to look is compiled, including ideas, assets from the internet, moodboards, concept art, technical details, camera control, music and sound and all the possible information that the student will need for the production phase.

4.2. Production

From the concepts and guidelines stipulated in the pre-production phase, the student must convey it in the previously mentioned 3D software. For this project, the programs used by the author have been 3Ds Max, Maya, ZBrush and Substance Painter, with some assistance from Blender if deemed necessary, as the methodologies found in the latter are usually more straightforward than the other software.

Using the capabilities of primitive manipulation using modifiers that 3Ds Max offers, a blockout is to be created as a basis for the final boss, which will be rigged in Maya to create the key poses that will directly impact the boss' layout and changes throughout the phases. Maya's tools allow the author to create all the controllers

necessary to create professional animations and precisely control the beast's movements. These poses and blockouts serve as a guideline to test out the game feel, balancing and pacing of the entire boss fight, as it will be a task that will involve both level design and character creation concepts at the same time.

From this blockout, the student will generate a high-poly model using ZBrush. Using this program and its powerful rendering and sculpting tools seen in 2.5.2.1, the author will create all the details, volumes and shapes in great detail of all the elements of the boss. Afterwards, the author will optimize the sculptures using either ZBrush's retopology tools— ZRemesher and ZModeler— for simpler shapes or by creating a low poly version in Maya, using the PolyDraw tool exclusive to this software (for more complex topologies). In both cases, the objective is to create a mesh that maintains the general shape and silhouette of the high poly model with fewer polygons.

Moreover, the model is to be texturized in different materials to ensure a high-quality product, as it is an enormous model that will require a higher resolution than typical models. For this, the student has chosen Substance Painter as his modus operandi, as its baking tools and Smart Material workflow allow the creation of procedurally generated textures for both cartoon and hyper-realistic models. Finally, the model must be adequately animated from the previously created key poses and rigs, where Maya's animation tools for animation graph editing and interpolation come into play.

A similar production process will be applied if other characters or enemies are involved in the boss fight, such as sub-bosses or minions. Each of them will follow the same process of blockout, high poly model, retopology, texturing, rigging and animation.

4.3. Post Production

To ensure that the quality is up to standard, the student will perform some revisions and testing with both the tutor and peers, from which an analysis of all the compiled opinions and tips will help fix issues arising from gameplay and aesthetics related to the boss.

Finally, for the final composition of the project, different assets and illumination treatments will be done to give the final touches to the project. This process entails creating proper visual effects and adding music and sound treatment to the boss.

All the assets involved in the post-production process will be extracted and created using external sources and tutorials, for the most part, to give it a professional look and prepare it for the final project presentation.

4.4. Time Analysis

For this project, the author has created three different Gantt charts, each with different objectives, such as showcasing different parts of the process and time estimations for the methodology described. The project will orbit around these chronograms as time guidelines for each step to adapt to emergent risks.

This first diagram showcases the work done for the first half of the project (from June 2022 to the time of delivering the first half of this document). The tasks include research, documentation and the preproduction process, which means defining the mechanics of the prototype and designing the boss around them, and coding.

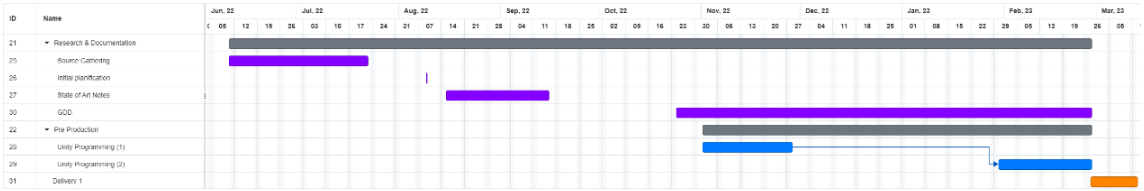

FIGURE 111. GANTT CHART 1: PRE-PRODUCTION

The second diagram showcases the distribution of tasks aimed to deliver all of the elements involved in the boss' design for the first sitting in June. The total time spent on each part of the production involves an intensive weekly workload for each step involved in character creation that does not correctly contemplate the time involved in other matters outside the prototype. Depending on the matter, it may take less or more time to complete each step, but the student decided to put one week per task and two for redacting documents as an approximate estimate obtained from the time spent on other assignments, which means dividing the work in one week for each task.

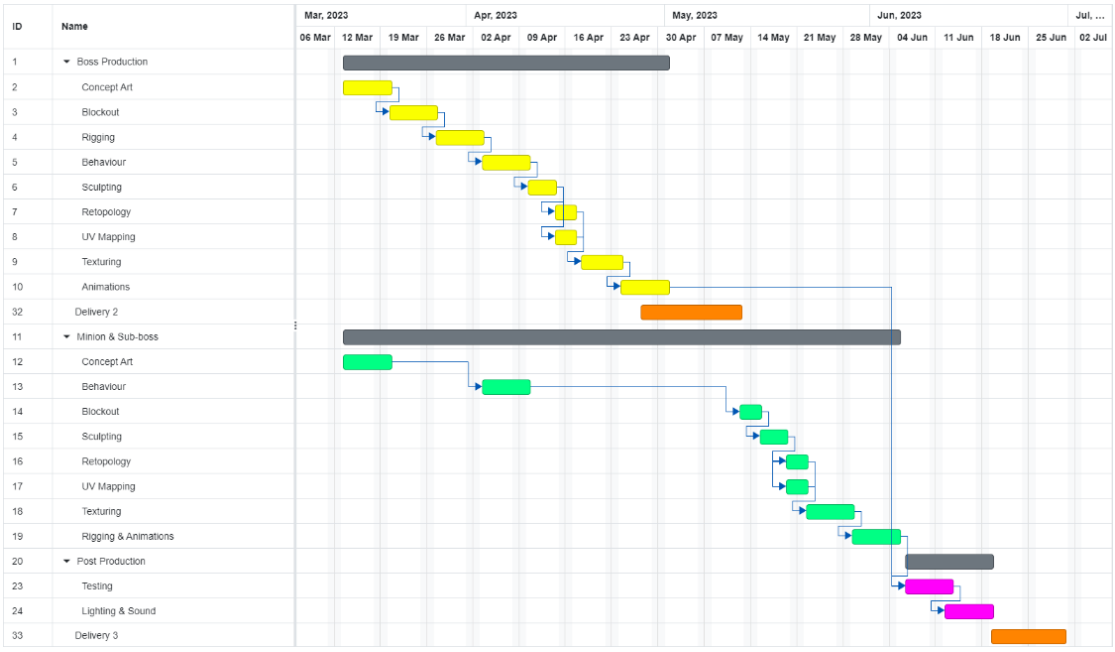

FIGURE 112. GANTT CHART 2: PROSPECTS FOR JUNE

The third diagram was created to ensure better results at the end of the project. Studies and past experiences from the author have shown that simultaneously dedicating an absurd amount of continued hours leads to worse products. Hence, the student considered another time distribution if he delivered the entire product in September, which would entail developing only the main boss itself while leaving other elements in a conceptual/blockout state.

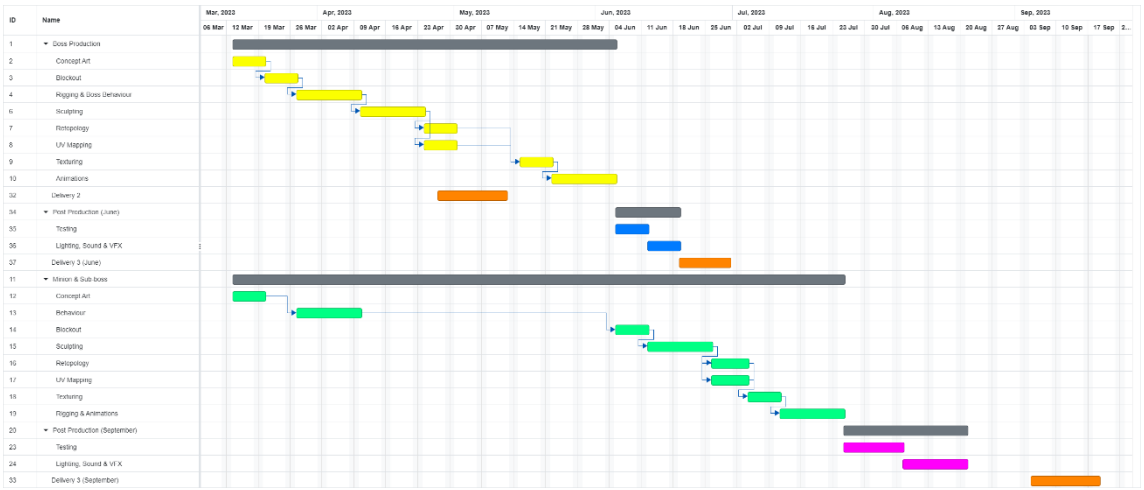

FIGURE 113. GANTT CHART 3: PROSPECTS FOR SEPTEMBER

Throughout the project, some plans changed based on the project's circumstances; thus, the Gantt charts above became obsolete before the second delivery of the project. As seen in the previous Gantt charts, adding a playtesting session in the production plan and the writing division were some defining factors that shifted the initial plans. The following chart showcases how the author more accurately distributed the time spent in each process step.

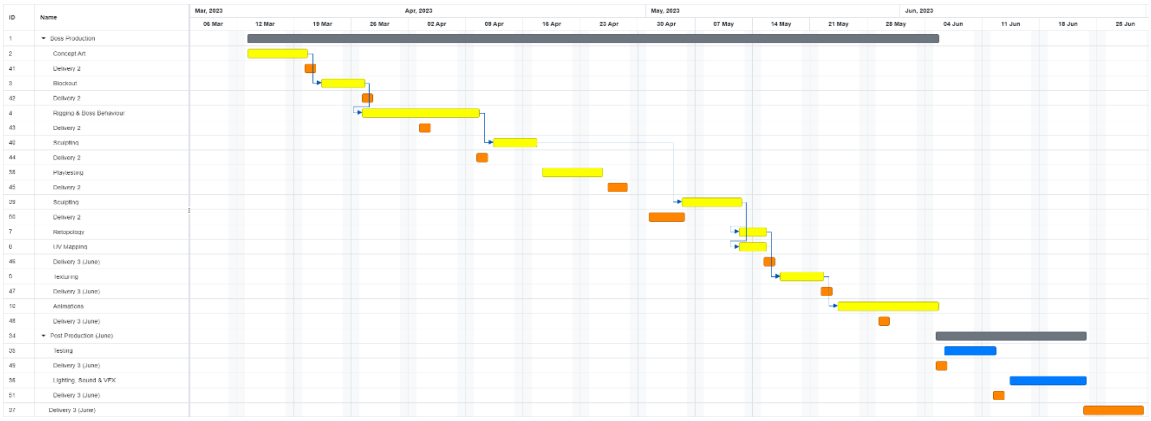

FIGURE 114. GANTT CHART 4: NEW REDISTRIBUTION OF TIME (AS OF THE 3RD OF MAY)

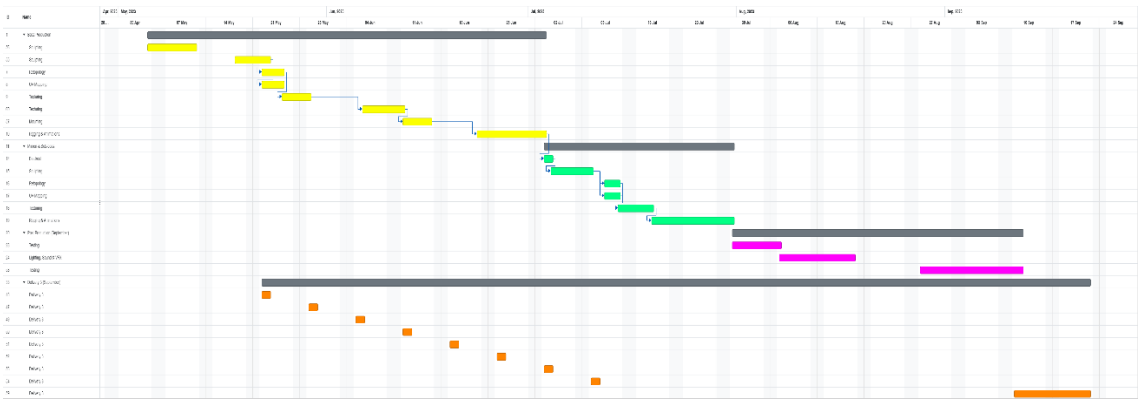

FIGURE 115. GANTT CHART 5: REDISTRIBUTION OF TIME AFTER THE 3RD OF MAY

5. Project Development

Based on the explained theoretical framework, the student visually conceptualizes the prototype's development and results of the methodology described in the previous section. The following are the results, difficulties and solutions the author has developed throughout the project.

5.1. Pre-production

For the first few months after the topic research, the author has established a basis upon which he has created the boss. The preproduction phase is crucial, as it helps limit the project's scope and sets the beginnings of the outcome. In this case, it has helped with many artistic and design choices for the boss and the prototype.

In this section, the student will focus on the initial aspects of the prototype outside the artistic production. These refer to both the design of the game and the boss' mechanics, and the programming of general elements mainly focused on the player and elements not intrinsically related to the colossal creature.

5.1.1. Design

5.1.1.1. Defining the mechanics

All videogames in the industry define their project's characteristics and mechanics around a list of immovable objectives and specs followed throughout the project. These guidelines receive the name of the four Ts and the game pillars. The former specs refer to the Theme, Tools, Time and Team specified in previous sections. The game pillars, however, are a set of values from which all game elements and gameplay stem.

Based on the market study of titles like Shadow of the Colossus and Solar Rush, the intention was to merge both styles into a single enjoyable experience. From the student's perspective, combining dynamism and flow based on the latter example with the size difference play of the former defined the essential core pillars of the game. Movement should feel sound, interconnected and flow-like as much as possible, and the camera should show how minuscule the player compares to our opponent.

Initially, the game would have had puzzle elements to make the gameplay more engaging, just like Shadow of the Colossus, and movement-related mechanics like a grappling hook, just like Solar Ash. However, it is vital to consider the project's time constraints, team limitations and contingency plan. While it is true that the prototype's design and programming are equally necessary, the latter had to be kept as simple as possible so it only consumed a small quantity of the author's time and kept the player's information front-loading low. This limitation kept the main mechanics to only three: running, double jumping and a 3-attack combo. The following page contains a visual representation of all of these mechanics in play. To teach the player about them, the student deemed it necessary to create a miniature obstacle course that teaches the mechanics as quickly as possible.

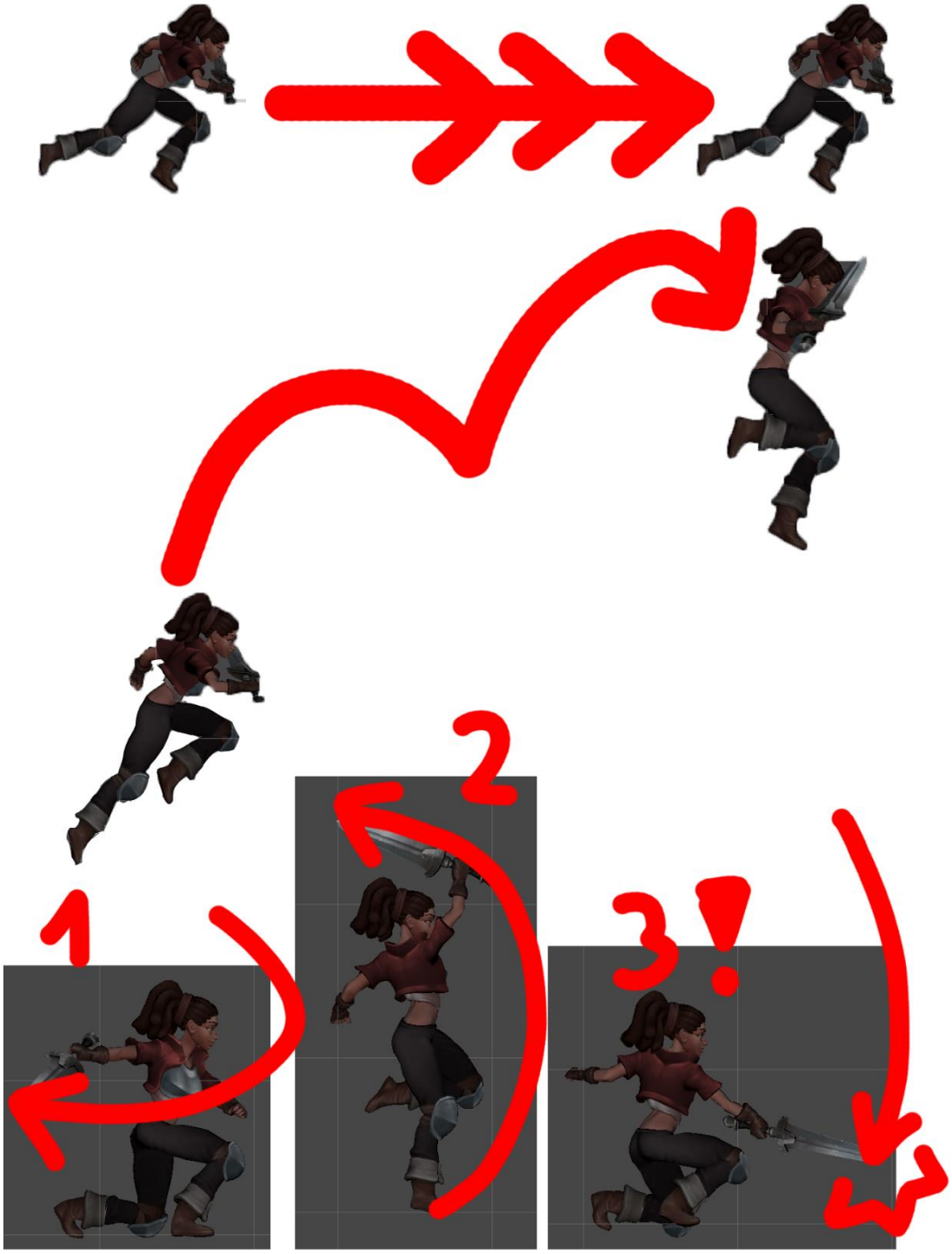

FIGURE 116. VISUAL REPRESENTATION OF THE GAME'S MECHANICS

Some things to consider about the GDD are that the student has written it in tandem with the prototyping and the game's development, explained in the following sections. The development notes mainly tackle the project's final directions and how the author reached such conclusions while glossing over the thought process in each subsequent area. Other aspects, such as VFX and game-feel elements, have been considered during the post-production phase.

5.1.1.2. Defining the boss

Scott Rogers' Level Up's Bonus Level 8 contains guidelines for creating a boss that will be used for this project (Rogers, 2014). He presents a series of points that work for many game bosses, such as narrative, animations, movement and attack patterns and assets required. However, these points must be adapted and combined with level design concepts for this battle.

While the prototype does not contain a narrative extrinsically, it follows the adventures of a young adventurer intending to defeat a colossal monster tree wreaking havoc in the middle of an evacuated city's ruins. Escárgotree, the above mentioned creature, was awakened by an evil woodpecker, Carpintroyer, who intends to destroy the locals invading its forest. The player must stop them before they destroy more of the surroundings. Thus, the battle occurs amid the boss' destruction: a big open field surrounded by trees and unclimbable thrashed skyscrapers. This narrative background would not appear until later in development, so the following explanation combines the thoughts at this moment of the process and the ending result.

FIGURE 117. FIRST ARENA ITERATION, WITH THE BOSS' LOCATION REPRESENTED AS A RED CIRCLE

The author intended to divide the boss into three phases, imitating the structure found in Shadow of the Colossus of destroying a weak spot to climb the boss to reach a spot where an event occurred, focalizing the action in different parts of the structure rather than the whole not to overwhelm the player. This structure is one of the reasons why the monster is tree-like, as the central design of a tree is also divided into three parts. A level 0 was considered to teach the player how to play the game, but the writer discarded it at this point of development due to initial time constraint problems.

Hence, the boss starts with slow, cyclical, telegraphed attacks towards the player. The player's first objective is to destroy the weak spot of the parts causing the attacks in the first place. Later in development, these ended up being the beast's roots, which, when stomping the ground, generated pulses that hurt the player and stunned them shortly.

FIGURE 118. THE PLAYER ATTACKS ONE OF THE LEGS WHILE ANOTHER STOMPS THE GROUND

When all the required weak spots break down, the creature stops and stays on the ground, allowing the player to climb on top of one of its limbs and reach the body, a 2.5D labyrinth that the player will have to traverse to reach the top. The level contains danger zones, minions and other interactable elements like keys and hearts to help the player in their journey. However, many of these elements would not be conceived until a level cadence analysis was determined, which was created as the boss concept was defined.

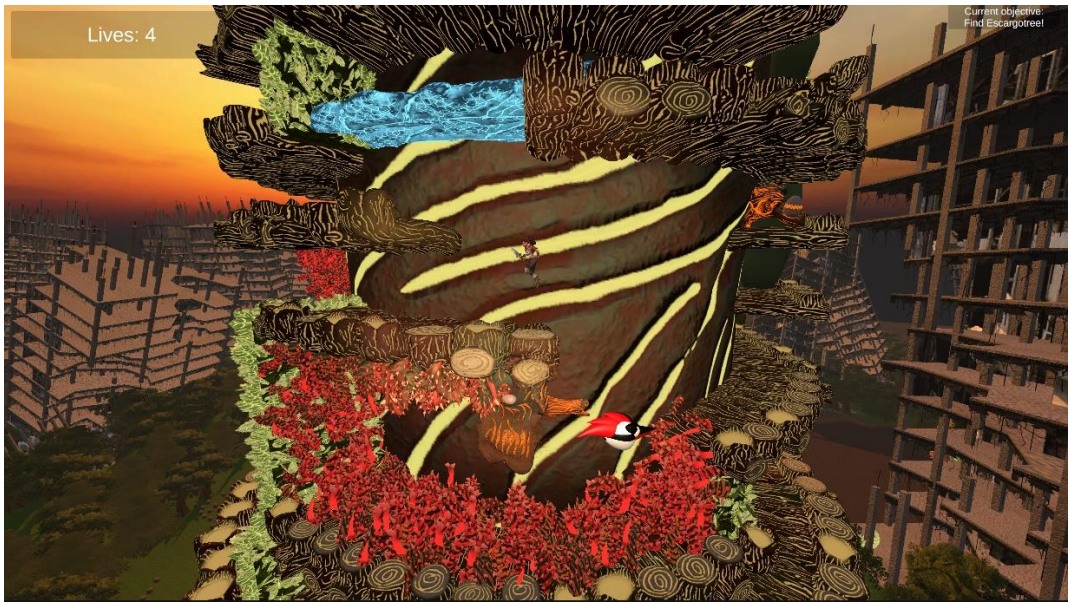

FIGURE 119. THE PLAYER TRAVERSING AN EARLY VERSION OF THE 2.5D SECTION

Finally, at the top of it all, the evil mastermind (in this case, the woodpecker) awaits to attack. The mini-boss pattern follows a simple telegraphed thrusting attack that, if the player avoids it by jumping over the generated shockwave, will make the woodpecker get stuck on the ground, thus allowing the player to hit it repeatedly to kill it. Whenever the boss stops, it cycles between two invulnerable phases: one that spawns several minions seen across the previous level and another in which they thrust thrice in quick succession.

FIGURE 120. THE TOP OF THE BOSS, WHERE CARPINTROYER BATTLES THE PLAYER

These battle phases were the focus and unmovable aspects of the boss. While several minor changes were made in between, the author respected these general concepts and ideas across the project, and all elements that conform to the boss surrounded this vision, keeping it as easy to learn as possible for the reasons previously mentioned.

5.1.2. Programming: Prototyping & Testing

For the first few months of the project, the leading game mechanics described in the GDD were scripted and tested to see what mechanics were plausible to do in the pre-established time. This step has adapted, shaped and even helped create many elements in the design documentation, as it started before the proper boss production.

As mentioned above, the student initially intended to develop the project using Unreal Engine 5, as it offered many excellent artistic and rendering tools that would improve the image quality tremendously. However, due to the inability to learn how to use the engine's blueprint system in time to deliver, this decision was quickly scrapped and swapped to Unity. Hence, the first days involved looking for assets that could serve as an

environment and main character, limiting the number of options and ideas for the boss to the context that the arena and the protagonist allowed.

Using dummy models to test before the final one was implemented, the author programmed the main character's basic movements, including moving based on the camera's direction, double jumping and sprinting. When the final model for the young adventurer was chosen, the student created the animation triggers for the model and 3-hit combo, which does more damage with each subsequent hit (similar to Action RPGs).

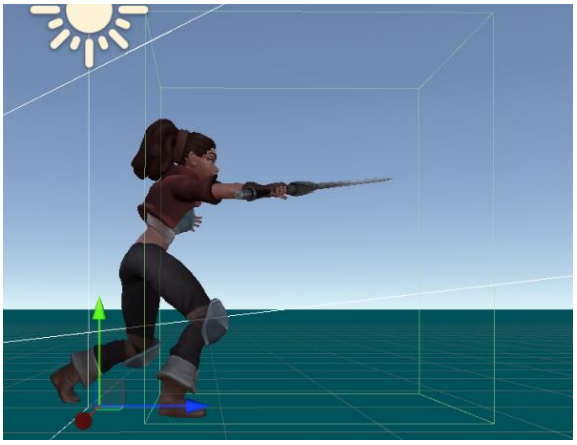

FIGURE 121. SCREENSHOT OF THE PLAYER SHOWCASING THE ATTACK HITBOX

To add some difficulty to the level, the student created many challenges that appeared alongside the boss. The first of these challenges were flying minions that were shot from a cannon, followed the player, hurt them on impact, and could die with a sword strike, inspired by the Bullet Bills from the game Super Mario Bros. The problem with these creatures is that they were tough to strike with a sword, even if the hitbox was huge compared to the player. The reason is that they would come in front of the player, making the space between the player and the minion arduous to notice. This issue led to the decision to make the second section a 2D platformer similar to the pixel sections in the video game Super Mario Odyssey, where these monsters were easier to see and, therefore, easier to kill. With these creatures into play, the author created an austere health and reviving system to add challenges and punishments for error.

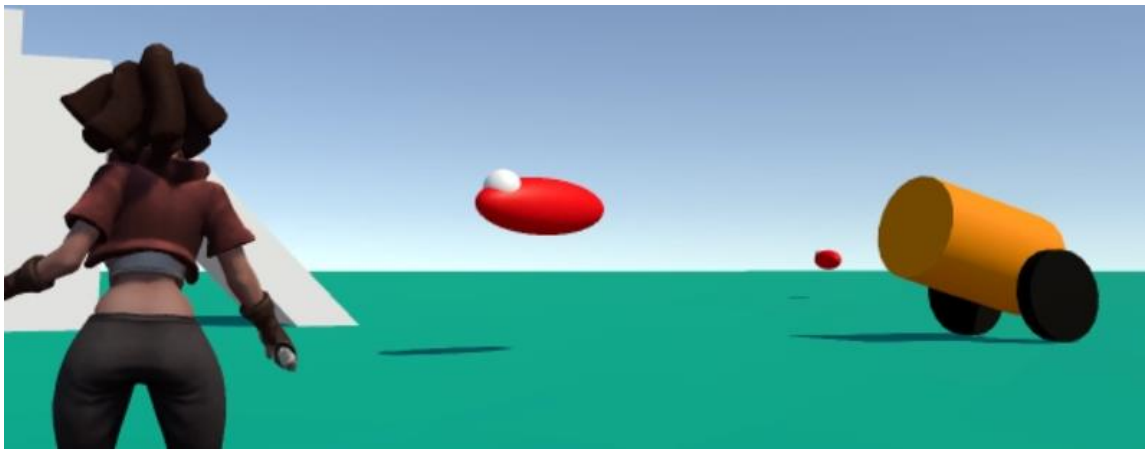

FIGURE 122. SOME MINIONS FLYING THROUGH THE SKY

For the main creature, the first problems that appeared were the colliders of its body and how the player could navigate the level using them. Based on the experience from Santa Monica Studios on creating colossal bosses, creating floors for organic models was a significant and complicated part of the boss' development, so the student had to try more accessible alternatives and ideas.

In the first versions, the initial proposal was to use a script that updated the mesh's collider for each animation frame, which worked well for simpler models. Nevertheless, it started getting costly in performance when applied to the final environment, so it was quickly discarded for simpler, more straightforward collision boxes as soon as the author started with the boss' concepts. This decision limited the scope for the boss design, as now it had to have flat surfaces that could stand these colliders, which meant creating something out of blocks or platforms, like a robot or a mushroom-based creature.

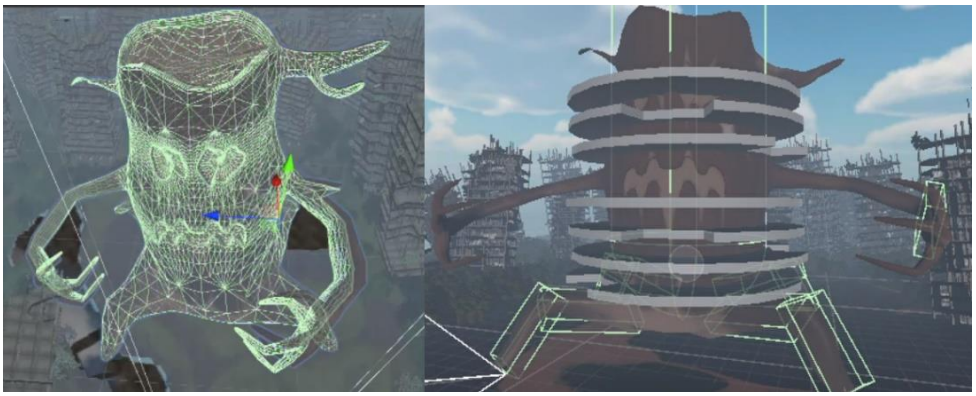

FIGURE 123. BOTH OPTIONS FOR THE MESH COLLISIONS. ON THE LEFT: MESH COLLIDER; ON THE RIGHT: SIMPLE COLLIDERS

Finally, with all these decisions taken, the last step for this preproduction phase involved the creation of all the different camera views used in the space. For that, the author opted for zone triggers and a camera system called Cinemachine, allowing for smooth transitions between cameras based on the area the player was traversing (especially the body and the top of the head). Since the player's movement depends on the camera, creating the movement for the 2D section was more manageable to control, as they already revolved around the main body's platforms. Hence, the only things necessary were some additional tweaking of each camera's properties.

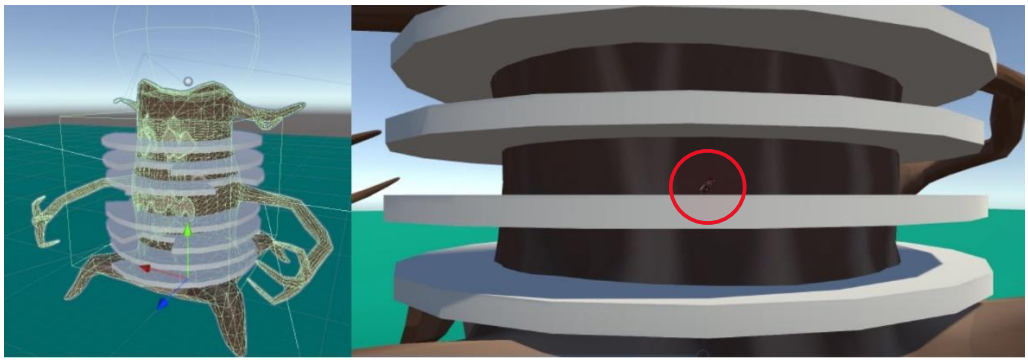

FIGURE 124. CAMERA COLLIDERS AND CAMERA BEHAVIOUR SHOWCASED, WITH THE PLAYER CIRCLED IN THE MIDDLE

5.2. Production I: The Tree

With all the conditions set from the preproduction phase, the production of the boss proper began in February. The student will mainly focus on the prototype's artistic production in this section, referring to creating the necessary models from concept to animation. He also advanced the prototype in unison to make it functionally playable.

5.2.1. Concept Art: Moodboards & Thumbnailing

The student started pitching ideas for the project in the initial phases of preproduction, as he required a basis from which to start creating mechanics and scripts that he would later use in the final build. Hence, the first step was brainstorming ideas for a boss, ranging from a tattoo inside a person navigating through environments created with other tattoos, giant aliens and motherships, robots, or even directly imitating other media in the form of gods, humanoid giants or kaiju, like the initial proposition. However, what would be a good monster concept with these proportions that could be plausible to develop within the stipulated time?

Ultimately, the author created the Art Bible, a document much like the GDD where all artistic decisions can be visualised. Early in development, the student conceived initial aspects he had to respect across the project. These critical characteristics aligned with the writer's abilities and skills in the 3D pipeline and affected the decision-making of assets outside the boss (such as the protagonist or the environment). Later, extra additions, such as the camera angle inspirations and other technical details, would be included.

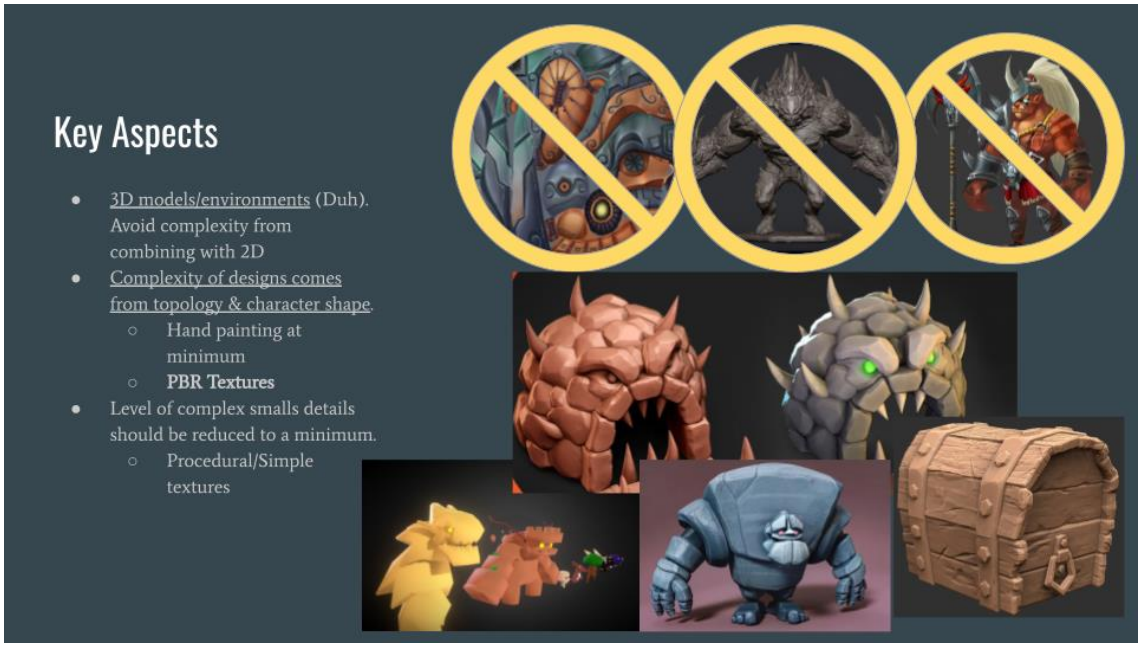

FIGURE 125. KEY ASPECTS PAGE OF THE ART BIBLE

With these goals in mind, it was time to search the Unity Asset store for a protagonist and an environment, two elements that did not enter the student's project prospection. Such assets had to be ultimately free to keep the budget as low as possible. They also had to adapt to the necessities listed in the GDD regarding animations for the characters, space for the environments and theme between one another. The deciding factors that

influenced the final decisions were especially character-focused, as there were fewer choices. Ultimately, the author presented two art-style propositions, two theme propositions and five boss proposals, each with a respective moodboard for information. All art styles and themes were interchangeable, but the boss options were intimately tied with the theme propositions.

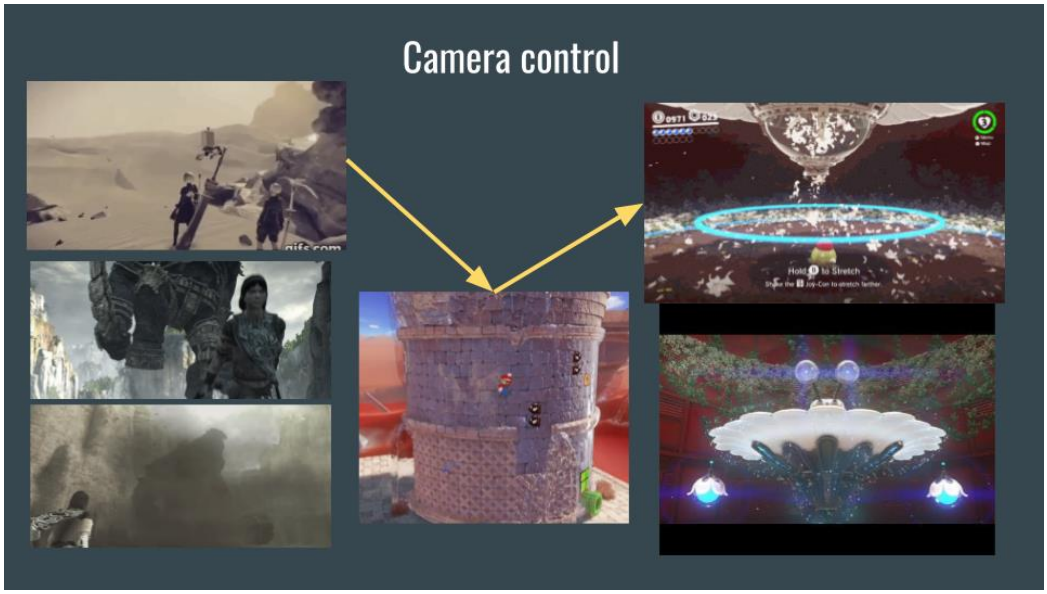

FIGURE 126. CAMERA CONTROL REFERENCES

The first proposal was to create a stylised art style with a modern theme, using urban environments and either Janmo or Unity-chan as characters, which were highly complex models with many animations. In the latter's case, some shaders must be used to improve the character's colours, limiting the number of usable landscapes. This art style is one of the most complicated and varied to pull off exceptionally, yet one of the most beautiful. From this option, there were three boss possibilities.

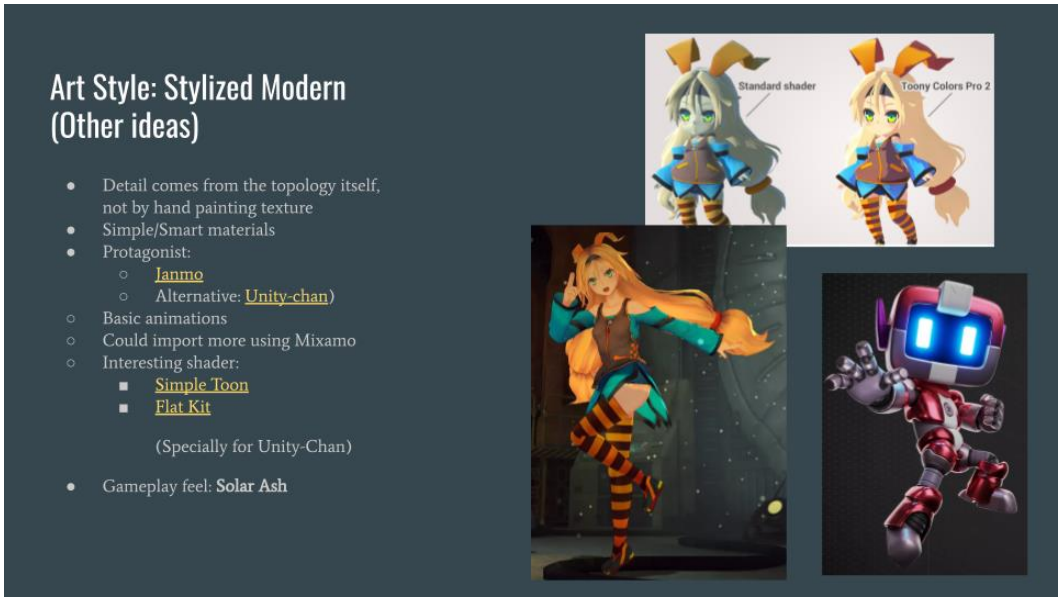

FIGURE 127. NOTES FROM THE DISCARDED ART STYLE "STYLIZED MODERN"

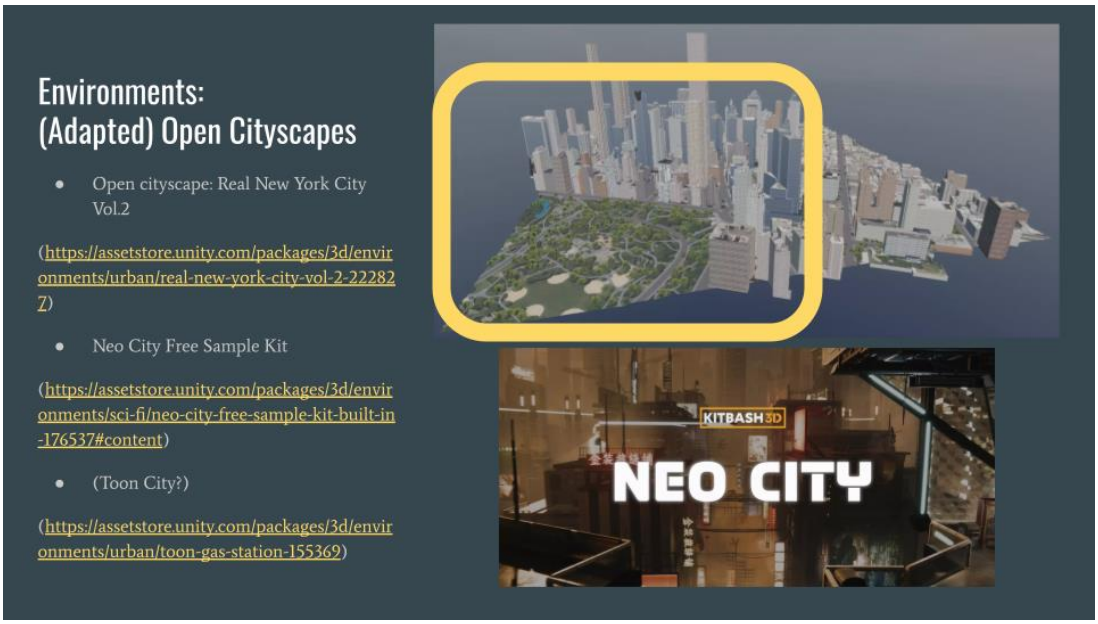

FIGURE 128. POSSIBLE ENVIRONMENTS FOR "OPEN CITYSCAPES"

The first one was a trash monster, basically a monster made out of parts of the environment like lamp posts, traffic signs, roads and general trash. This concept would have required creating modular assets to generate interesting silhouettes with urban elements while leaving the general structure's textures as a massless blob with a seamless trash texture. It was a fascinating idea, but the writer quickly discarded it due to the complexity of the assets required and the shape's openness to possibilities.

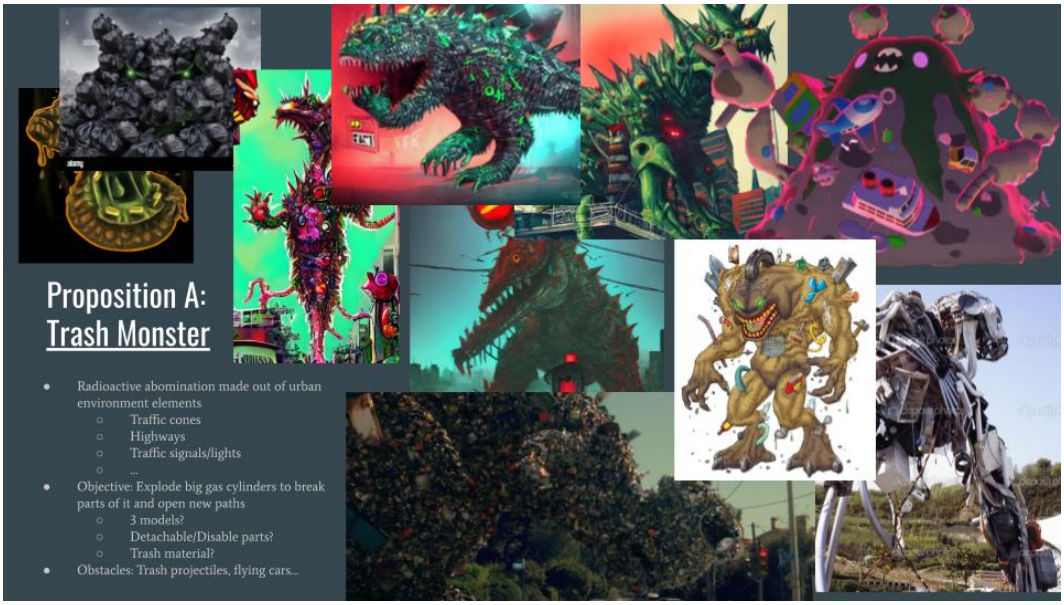

FIGURE 129. MOODBOARD FOR THE TRASH MONSTER PROPOSITION

The second one was creating an alien structure or creature, a mothership with either squid-like or insectoid characteristics. The objective was to traverse its many appendixes to reach the top, avoiding lasers, mini UFOs and the ever-changing structure of the appendixes. While interesting, the problem with this idea is that the

addenda working as an environment would require a level of programming and collider tweaking that did not align with the project's objectives. This idea could have been plausible if the initial collider script had been optimal, but it was not worth the risk.

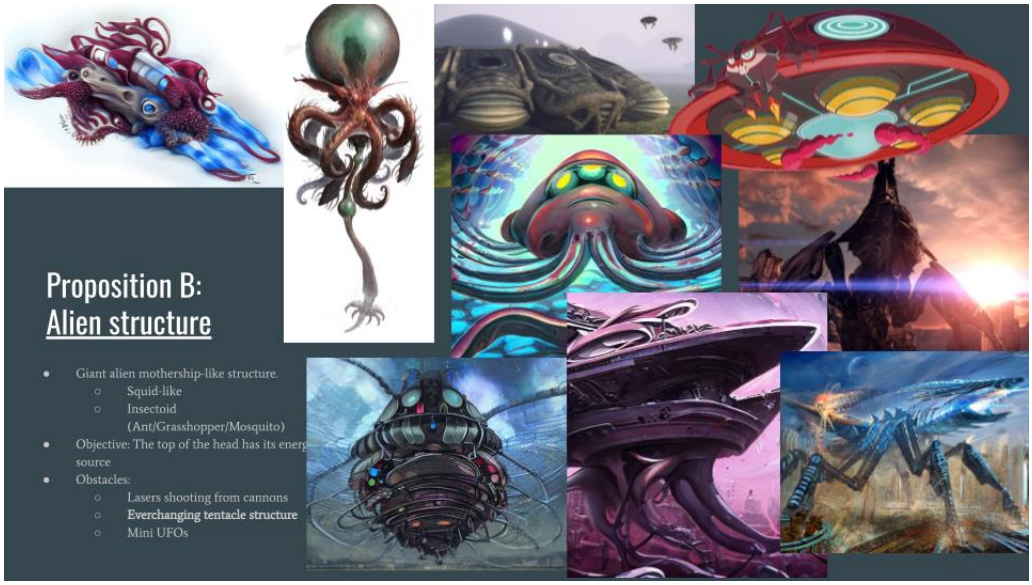

FIGURE 130. MOODBOARD FOR THE ALIEN STRUCTURE PROPOSITION

As unoriginal as it may seem, the final proposition was to create a kaiju proper. Based on the designs of existing creatures like Godzilla, Mothra or King Ghidorah, design and conceive a kaiju that could be navigable. The vision involved using the creature's dry skin and scales as platforms, but the concept could have been more attractive based on the others. It felt uninspired while being a pretty open idea that could work in multiple settings.

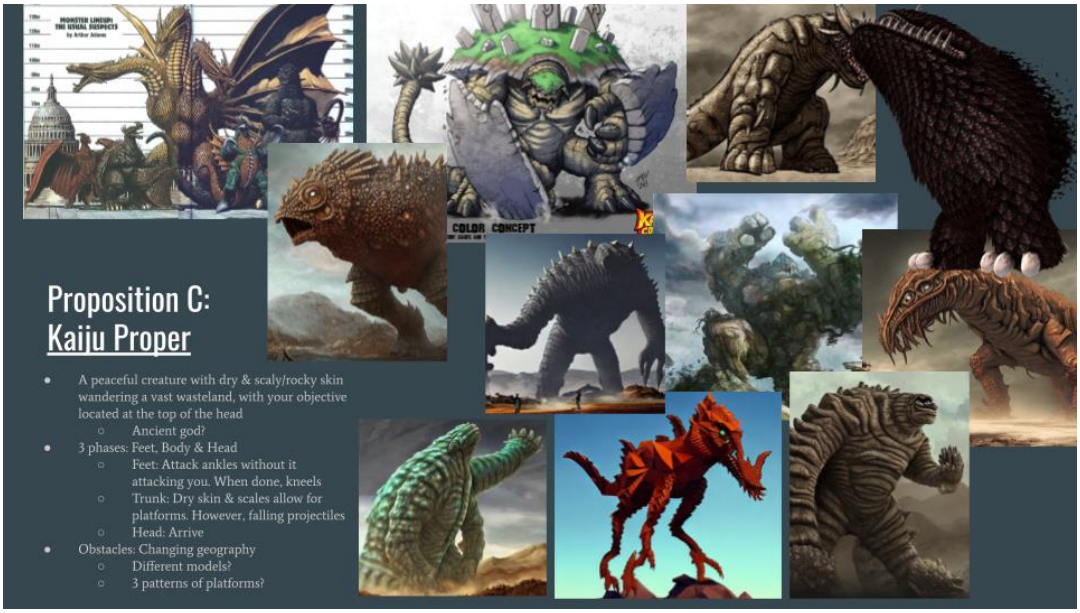

FIGURE 131. MOODBOARD FOR THE KAIJU PROPOSITION

The second art style and theme proposals were a low-poly aesthetic in a fantasy setting, using the RPG Tiny Hero Duo as a base character and many of the low-poly environments available in the Unity Asset Store, especially wastelands and deserts. This style required a high level of stylisation and optimisation, with little sculpting involved and the option of using shaders to enhance the minimalist topology of the models involved. From this, there were also three propositions, one being the kaiju proper already described in the previous paragraph.

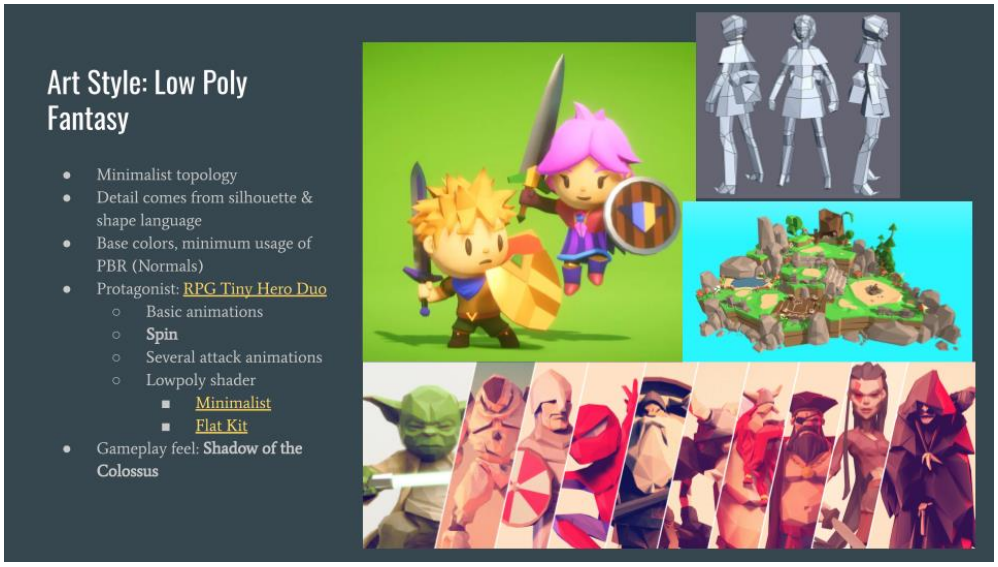

FIGURE 132. NOTES FROM THE DISCARDED ART STYLE "LOW-POLY FANTASY"

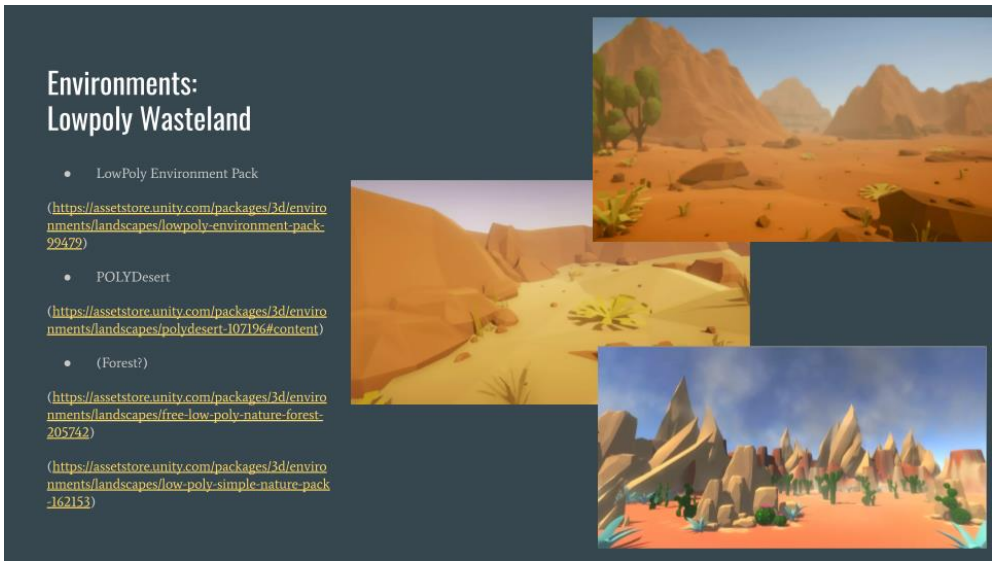

FIGURE 133. POSSIBLE ENVIRONMENTS FOR "LOW-POLY WASTELANDS"

The first was a colossus made of stone and rocky formations, similar to the kaiju and the trash monster but with different materials and assets. However, this concept did not lead anywhere. It was incredibly dull and overshadowed by the other ideas in complexity and interest, so the author did not even make a moodboard for it. In contrast, the second proposition was more intriguing and, ultimately, the final decision developed in this project. The concept involved an Ent or tree-like structure, in which its stages could be easily divided into three

regions (the roots, the trunk and the top), and the objective was to climb from the roots up to the top. It seemed simple, but unlike the kaiju, there are many types of trees, structures, colour palettes and references from which to get inspiration. It was worth the exploration, and so it was the chosen proposal.

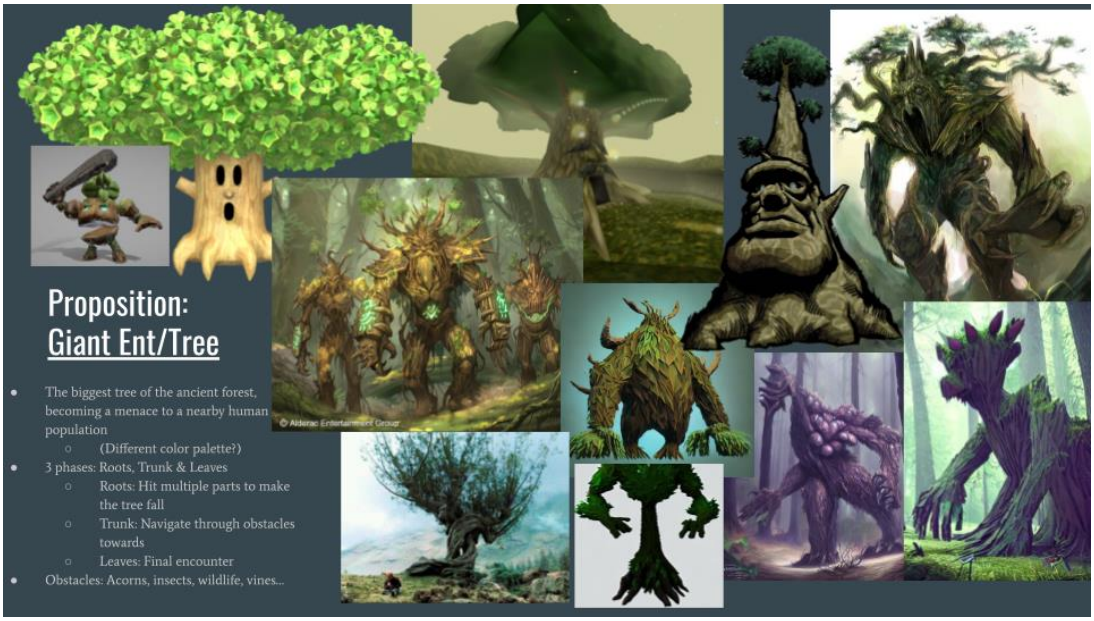

FIGURE 134. MOODBOARD FOR THE ENT PROPOSITION

This decision required a change in style-theme combination, opting for a stylised art style in a fantasy setting, using the asset "Medieval Cartoon Warriors" as the main protagonist model, which included all the basic animations required for the game, and the usage of multiple assets from stylised environment asset packs. The final setting was the "Destroyed City", a relatively simple environment containing ruined buildings and general vegetation that could be adapted with other stylised assets. Having set it, the exploratory phase could begin.

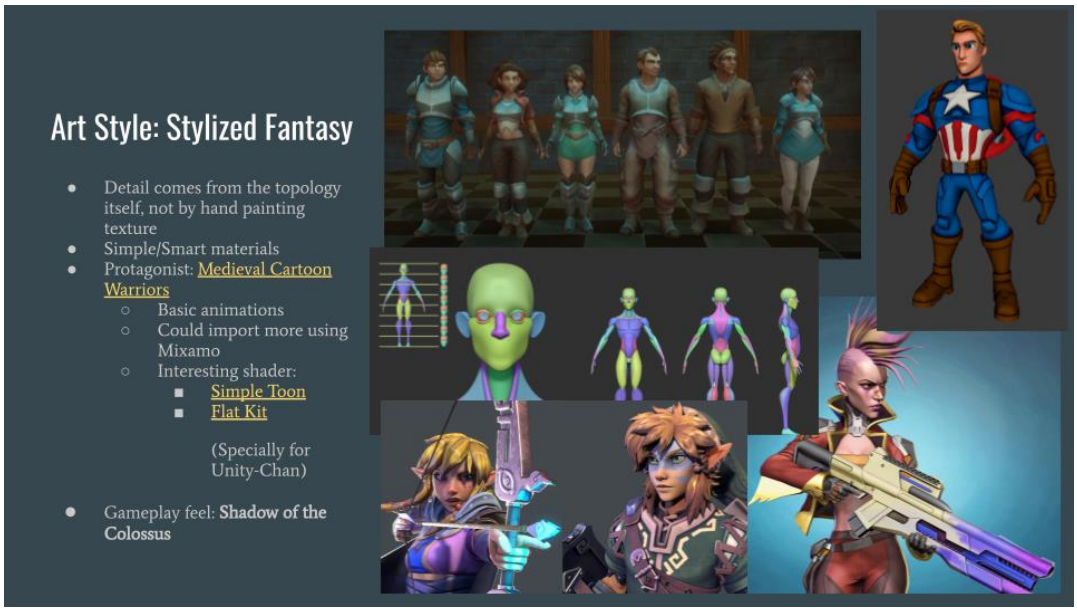

FIGURE 135. NOTES AND SCREENSHOTS FOR THE ART STYLE "STYLIZED FANTASY"

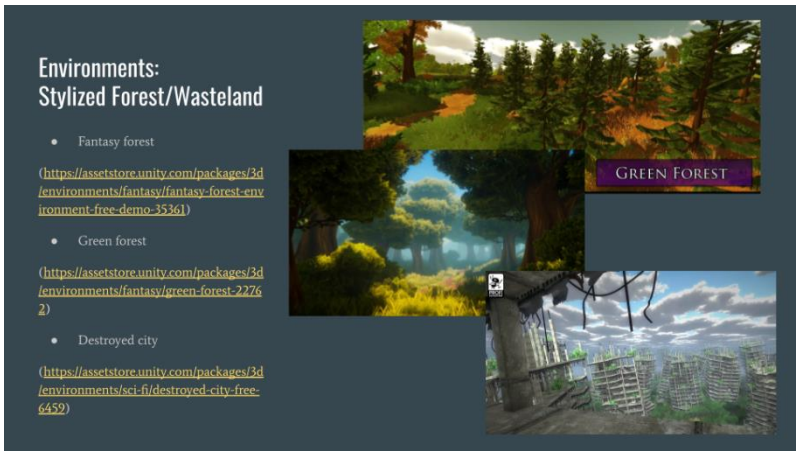

FIGURE 136. ENVIRONMENT ASSET PACKS FOR A STYLIZED FOREST AND WASTELANDS

With this basis as a starting point, the first objective was to construct many moodboards to convey ideas and possible outcomes of the final product. From the get-go, it was clear that the tree could not be grounded, as they should be movable objectives based on the boss fight design. This rule even suggested making the boss machine-like, inspired by games like Sonic the Hedgehog or Megaman, in which these types of machinery were common. However, the idea was dismissed for approaching the concept in its original nature-like idea. As such, the author created an initial concept based on the idea of the forest' spooky tree, using several inspirations from video games like Kingdom Hearts, Pokemon and Halloween artwork. The primary idea behind the moodboard is to showcase forms strictly based on the concept of a tree divided into three discernible parts, all stacked on top of each other with some radial symmetry in the middle area. While functional, the student wanted to explore other possibilities.

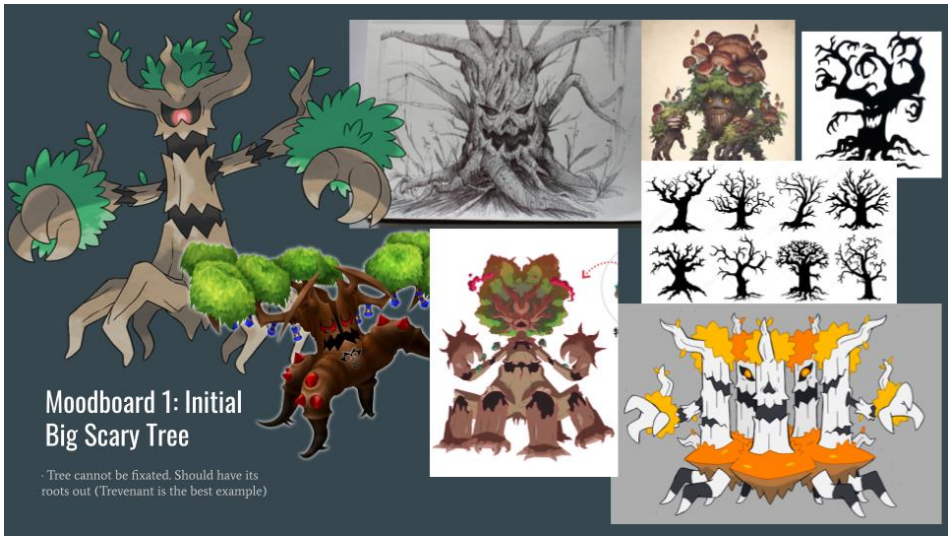

FIGURE 137. MOODBOARD 1: INITIAL - BIG SCARY TREE EXPLORATION

As a possible alternative, there is the mythical creature approach of basing the boss in treants or ents, forest creatures from Tolkien's fantasy world Middle-Earth that take the form of giant humanoid trees. This concept has been explored in many fantasy settings, usually oscillating around the idea of an old bearded man covered in bark, bushes and sometimes even bird nests. However, many games and media have created variants

combining tree-like structures with other mythical creatures like the centaur or the minotaur and animals like horses or turtles. The student found these types of creatures bewitching. So, one of the objectives for future concept art was to create variations of treants using different animals, especially insects or reptiles.

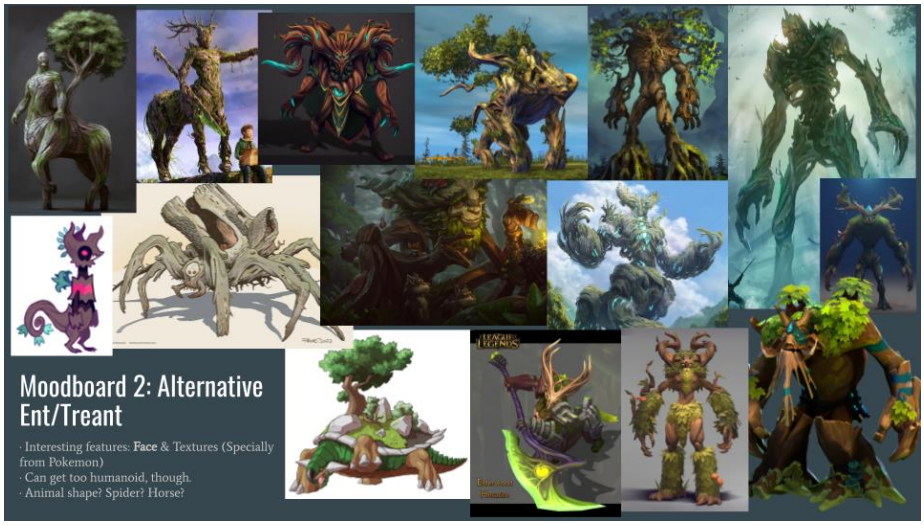

FIGURE 138. MOODBOARD 2: ALTERNATIVE - ENT/TREANT EXPLORATION

After finding the creature's basis, the author researched the types of trees the monster should be based on regarding shape and materials. For this, there were three moodboards conceived. The first explored the idea of trees that could be functional design-wise yet aesthetically pleasing simultaneously. The structure that conveyed the concept of platforms was fungi or mushroom trees, such as the baobab. This notion helped solidify one of the many discarded ideas for this creature: instead of being plant-based, it could be more fungi-based. Nevertheless, it was quickly scrapped to avoid deviating from the original idea, much like the robotic notions from earlier, and because the shape language ought to be more varied.

FIGURE 139. MOODBOARD 3.1: MUSHROOM TREES EXPLORATION

The second board explored one of the most beautiful trees in shape and dynamism: the bonsai. How they grow, morph, adapt and expand across the medium is such an enticing concept that it was worth exploring to create

mesmerising shapes and forms. The author believes that limiting the idea of trees as straight shapes could become tasteless genuinely fast; however, in terms of design and programming, it took more work to create a navigable structure, as most bonsais branch out horizontally rather than vertically. Some more convoluted shapes could become troublesome if they were required forms for the trunks' structure. Nevertheless, the researcher extracted many notions he could apply in future concept art.

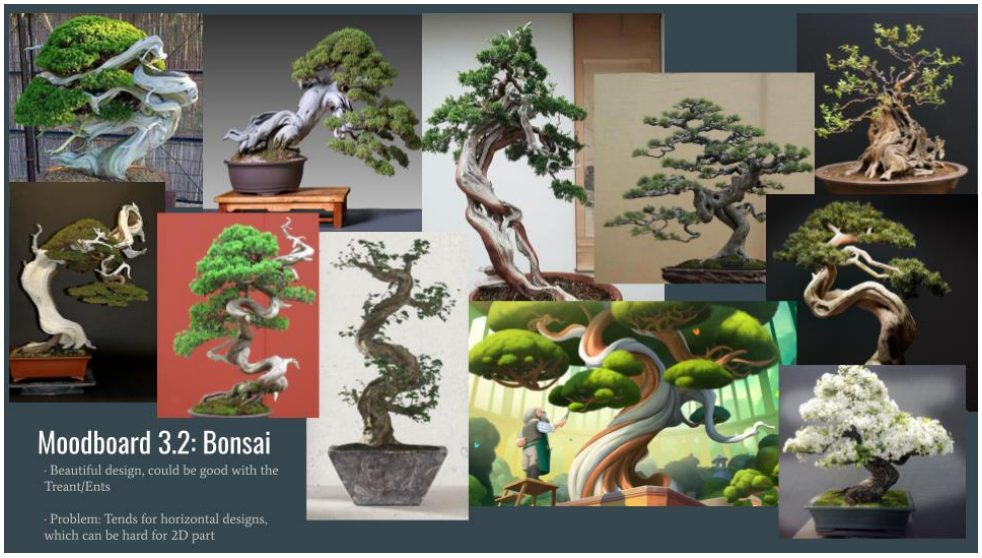

FIGURE 140. MOODBOARD 3.2: BONSAI EXPLORATION

Finally, the third moodboard explored concepts from the student's territory of origin: Catalunya. The pupil deemed taking into account his roots (pun intended) as something important for the project. Henceforth, research regarding some of the most memorable trees in his homeland was done to compile images of the most impressive bark materials. The ending result showcases a clear contrast between the overgrowing and twisting directions the roots dig through the ground — reminiscent of the bonsais and the straight yet imponent shade their trunks cast upon us — reminiscent of the mushroom trees.

FIGURE 141. MOODBOARD 3.3: CATALAN TREES

The writer used artificial intelligence to generate more reference images and created some pictures already seen before and even entire moodboards based on specific prompts. The algorithms used for this project have been Stable Diffusion and DALL·E 2 due to their affordability and the quality of the images they could produce if the commands used were specific enough. Using adjectives like "tree-like", "bark", "leafy", "colossal", and "huge" and nouns like "game boss", "monster", "ent", or many animal names like "deer", "horse", or "spider" served to forge interesting images stemming from the same concept.

FIGURE 142. MOODBOARD 3.4: EXPERIMENTING WITH DALLE·2

The next idea to clarify was the aspect of the minions. Based on the boss' nature theme, it must adapt its minions to this state and, as form follows function, these creatures should also have the form of a bullet. Therefore, two ideas appeared: they could be birds (specifically woodpeckers) or acorn-like creatures. As such, moodboards for both concepts were created. The author even considered exploring concepts aligning with more robotic notions to see if some attractive characteristics could appear.

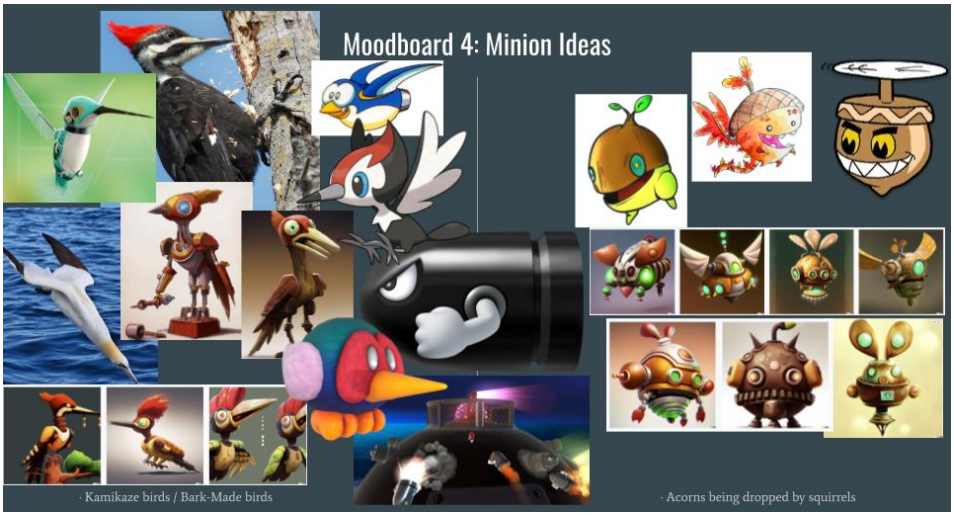

FIGURE 143. MOODBOARD 4: MINION IDEAS

In the end, the idea chosen for the minions was the birds, as they were more exciting for the sub-boss at the top than the acorns. Speaking of which, the author deemed it necessary to make the sub-boss at the top of the

tree have a thematic resemblance to the minions that spawned across the 2D section. As such, a moodboard for the mastermind was conceptualized, proposing three different ideas. The writer created conceptions for an evil mastermind inside a small bird-shaped ship, a parasitical worm-like plant, or the idea that finally came through, a giant woodpecker bird torturing the tree.

FIGURE 144. MOODBOARD 5: THE MASTERMIND

All of these moodboards were crucial in creating concept art for the boss. In these drawings, the artist proposes many options for all aspects of the boss, taking upon the notions expressed in the concept discovery. As such, the sketches explore general shapes for the boss' structure and its roots, trunk, platforms and consumables. The author's forte was never drawing in a notebook, so some drawings may appear crude to some. Nevertheless, these pages' comments, decisions and annotations cemented the idea of the amalgamation that would become Escárgotree.

FIGURE 145. CONCEPT ART PAGES FOR THUMBNAILING

Taking into note pages 7 and 8, the author notes one engaging idea to tie up the boss to its environment: making apparent that it surged from the ground, thus carrying part of the environment on its back (in this case, some of the buildings from the "Destroyed City" asset pack) as a conch. Hence, the final idea for Escárgotree ended up becoming a chimaera between 2 animals: a snail (or shrimp, depending on how it may be looked at), as it proposed an exciting silhouette for the body, and a camel, due to the necessity of having exterior roots and for having a similar structure on its back.

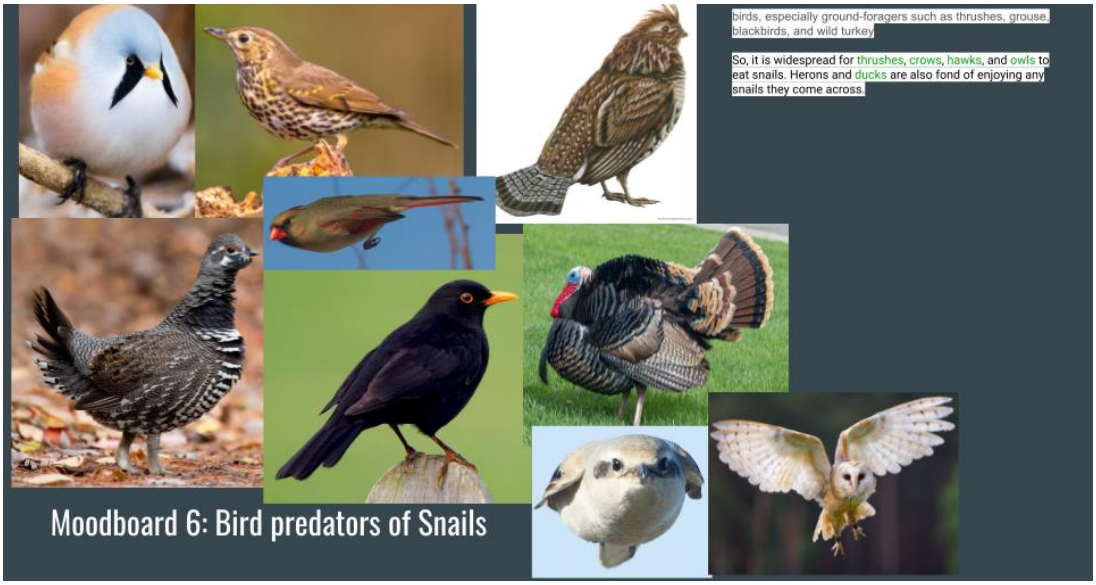

FIGURE 146. MOODBOARD 6: BIRD PREDATORS OF SNAILS

With these ideas in mind, the author did the final moodboards for two: researching the natural predators of snails for the bird's conception and using DALL·E 2 and references from other games as a final concept exploration. Having finished these, the concept of Escárgotree was born.

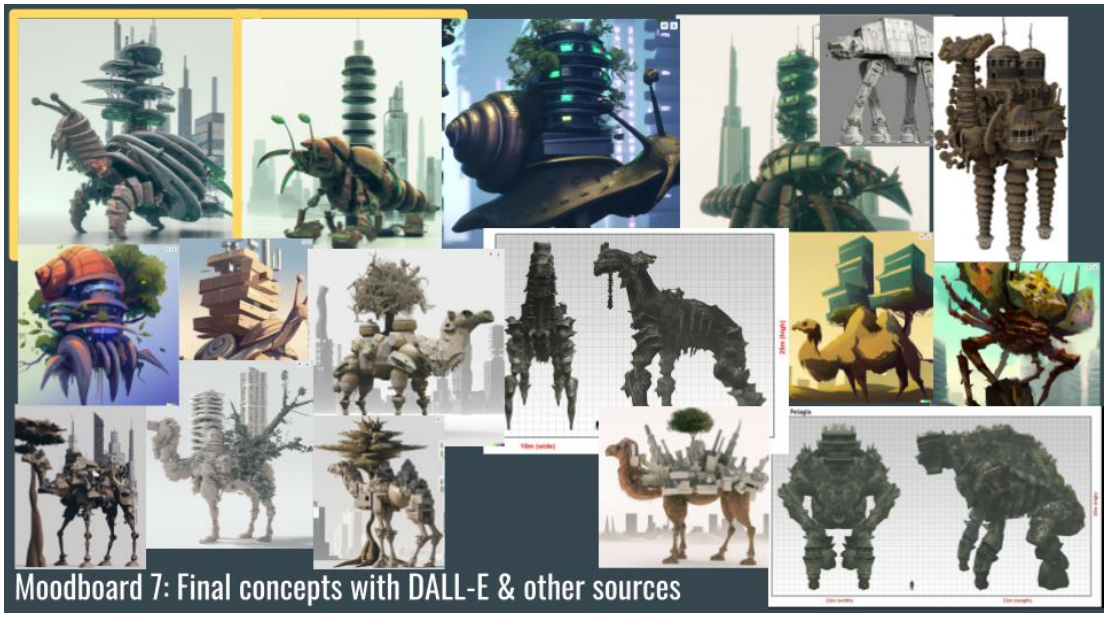

FIGURE 147. MOODBOARD 7: FINAL CONCEPTS WITH DALL·E & OTHER SOURCES

5.2.2. Blockout: Applying level design to the boss design

With the concept of the boss in mind and all the foundations for the battle settled it was time to bring Escárgotree to life. While the idea for the navigational part being a 2.5D section was decided early in development, it became one of the most prevalent concerns the author envisioned could quickly become troublesome. Thus, it was the first part designed and blocked out.

To begin with this structure and following Patrick Holleman's CCST Theory, the author measured the most fundamental unit of measurement experimented across this level: the jump height. For this, taking the character as a measurement unit, the author lined up the base model of a cylinder against the heights reached by the character's movement values at the time. This idea benefited the plan, as it sedimented the grounds to start with the level design. The author prepared the cylinder with some quick adjustments in Blender to prepare the model for Substance Painter. Using this software, the author could envision the final result of the level in a 3D environment while also being able to edit it in real-time.

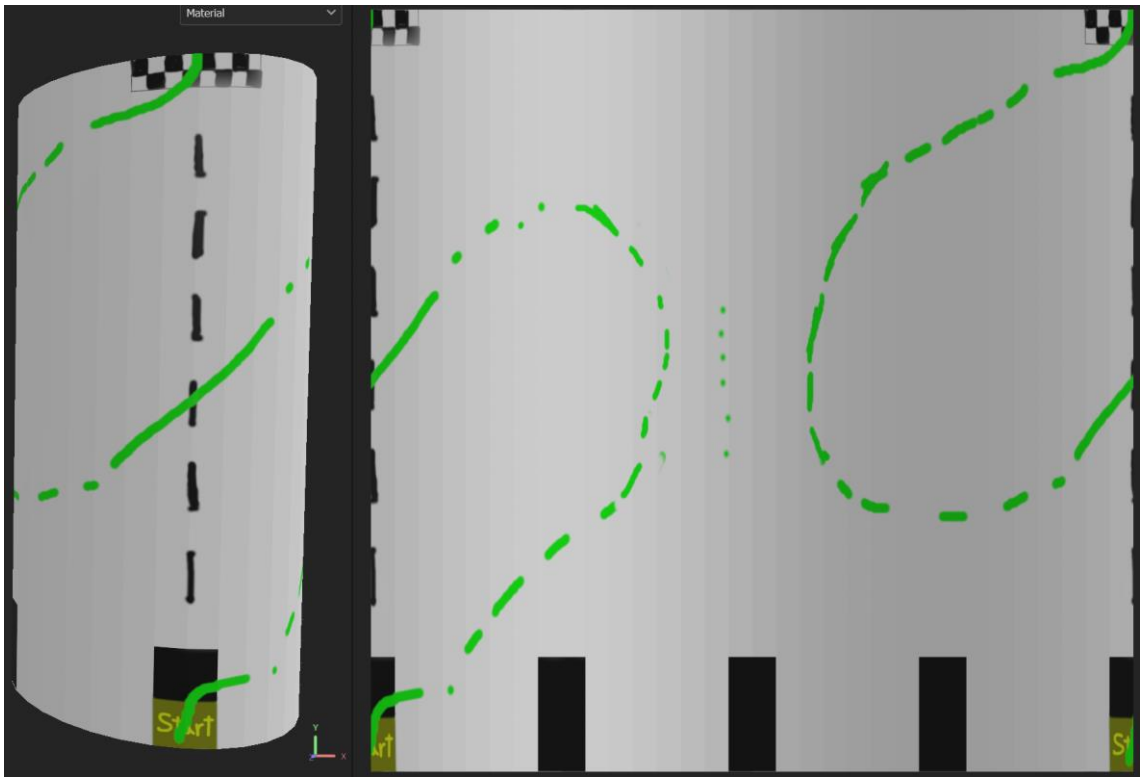

FIGURE 148. SCHEME OF THE 2.5D SECTION'S MEASUREMENTS AND THE PLAYER'S MANOEUVRE TRAJECTORY

As seen in the image above, the author planned the player's movement to follow directions similar to an asymptotic formula adapted to the entrance and exit of the level. The reasoning behind it was that it would make the player move across the entirety of the boss while also giving some foreshadowing and glimpses of what could be lying at the top. The boss was divided into platforms surrounding the surface of its side, and hence, the player began opening holes and placing elements.

FIGURE 149. EARLY SCREENSHOT OF THE PROCESS OF DESIGN

During the creation process, the author created some elements like pickups, keys, doors and spike pits that he would later program into the Unity project using existing scripts from both this same project's preproduction phase and other past projects. As base models for these parts, the student used primitive shapes like stars and cones to represent these features, but these were added obstacles and rewards that would be contextualized in the universe (as seen in Figure 145. Concept Art pages for thumbnailing).

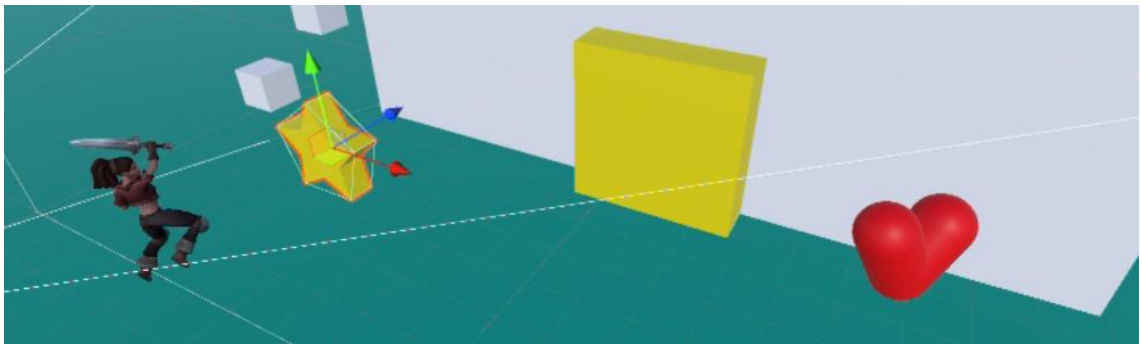

FIGURE 150. EXAMPLES OF INTERACTABLE ELEMENTS ADDED DURING THE LEVEL'S CONCEPTION

At 75% completion of the 2.5D section, the writer realized that the level could only be completed by first considering where to put the creature's face in this model. As it has been a long time since the student used 3Ds Max, he opted to practice by first creating Carpintroyer, the woodpecker sub-boss. Using primitive spheres and revolving bodies like cylinders and cones, he crafted the creature's shape using modifiers such as Bend, FFD (an acronym for Free Form Deformation), which, as the name implies, is used to deform the model by dividing the model into sections of equal measurement called "Lattice box"; and Array, a custom modifier that duplicates a model and spreads it across one or multiple axes. The sub-boss was completed with some final additions using Symmetry to unify the topology and assigning a material with a simple base colour to each polygon. The writer used its head as their model to unify it with its minions. Much like the minion concept, the idea of having a monster move as a missile was reflected in its design by making it look like a rocket ship, with its beak signifying the direction the bird took across the map.

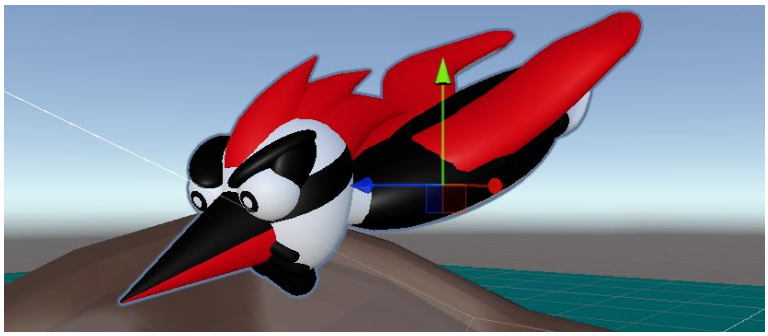

FIGURE 151. CARPINTROYER, AS SEEN INSIDE THE UNITY ENGINE

With this practice shape done, the player revisited the concept art done for the boss. As such, some of Escárgotree's final decisions regarding its facial structure and other aspects related to the 2D level were made. Before starting with the blockout proper, the author analysed his level to detect problems with its cadence. Using CCST theory, he identified its obstacles and skill themes and represented them in a table, with which mistakes regarding the level's cadence were detected predominantly in the earlier sections.

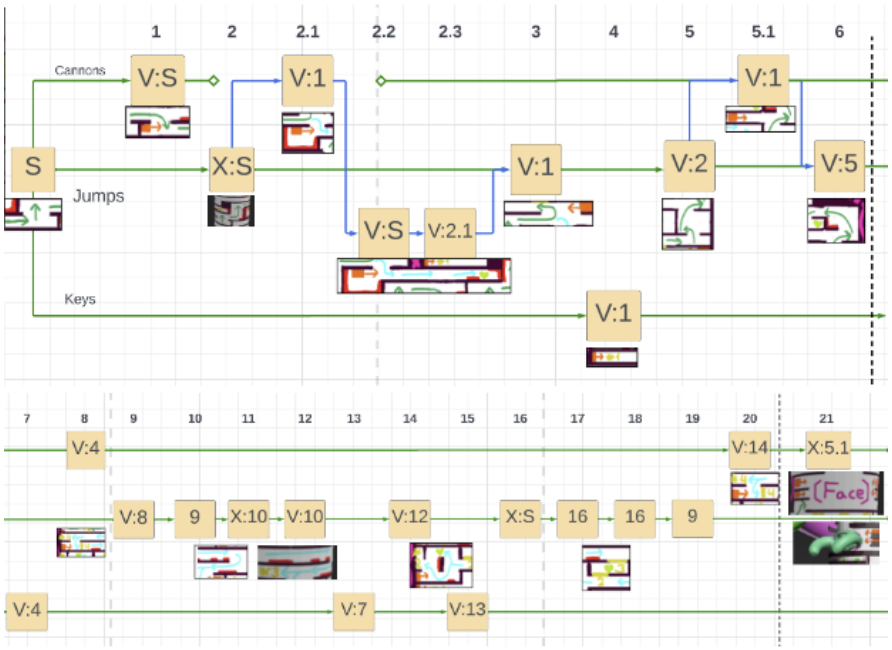

FIGURE 152. LEVEL CADENCE ANALYSIS OF THE 2.5D SECTION'S FIRST ITERATION

The level's first iteration has three skill themes: jumping across platforms, the minion-shooting artillery and the keys. It begins by presenting the most significant threats in safe areas so the player may feel they are avoiding a somewhat difficult obstacle; instead, it is just psychological. In section 2.1, the player can either get a secret heart they should have seen from the floor below or continue moving, adding a sense of exploration to the whole level that will be seen again moving forward. Section 4 presents the keys by placing them next to a cannon directed at the player. The idea is for players to try and hit the minions with their swords, which, at the same time, would trigger the key, thus teaching them that they can kill minions with their weapons and activate stuff with them. How the player moves across the level overall allows them to foreshadow what lies ahead,

their objectives and how to deal with them, all reflected in sections 6 through 8. After some linear platforming, the player should have been able to open all the doors blocking their way towards the face, so sections 17 and 18 open as a shortcut to reach the top faster. Finally, to add an element of surprise, section 21 would make the player reach the monster's face, which they would have to either use to their advantage or avoid at all costs, something that was unclear, as the blockout had not been defined yet.

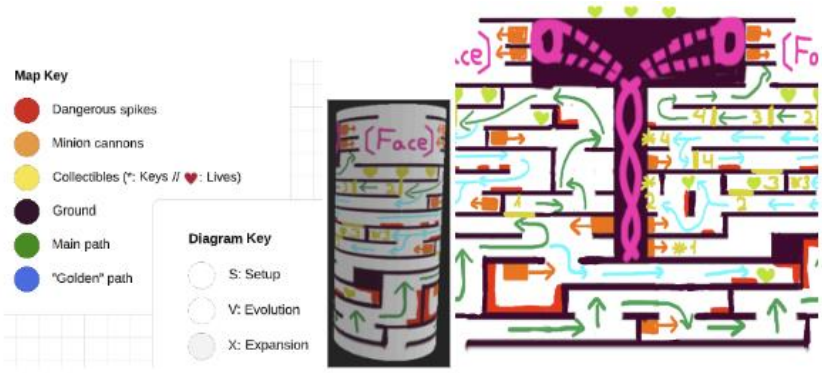

FIGURE 153. THE 2.5D SECTION'S FIRST ITERATION

However, with the platforming level done, Escárgotree's blockout could start. After importing the original model used to create the platforming section, the first step taken was to develop a disc-like structure divided into sections, each for each side of the original cylinder, that would serve as a base for each of the platforms, much like the platforms contained inside the scheme. To prevent players from falling, the shape of these discs was altered by using 3DsMax's Lowpoly editing tools to extrude a border surrounding these disks that would help prevent players from falling. This shape was repeated across the trunk for each subdivision, and with the low-poly editing tools, the artist removed all the unnecessary polygons. With some added cubes as walls, 20% of Escárgotree was complete.

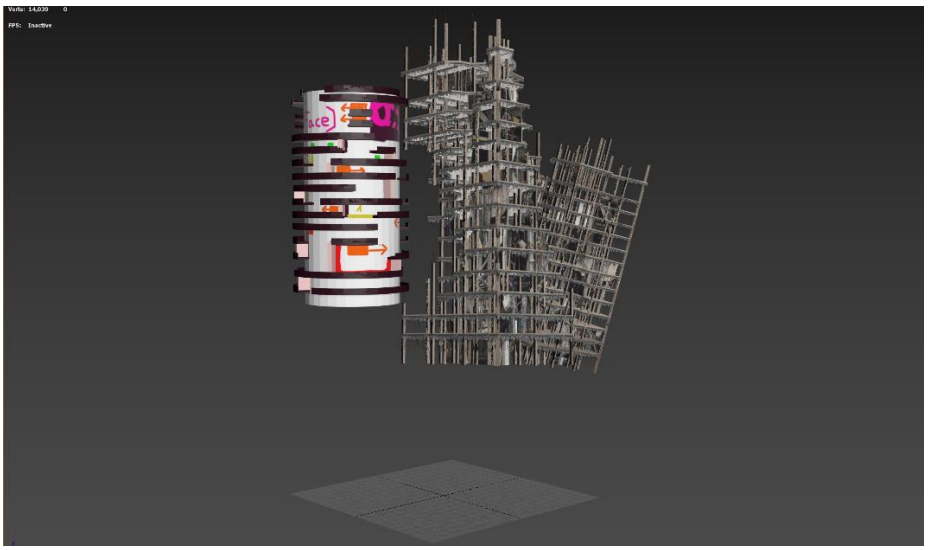

FIGURE 154. ESCÁRGOTREE'S TRUNK AND CONCH

For the body, the author followed a similar process used for the sub-boss creation. The concept art annotations mention that the conch is made of the same buildings as the battle arena's surrounding environment. As such, these were extracted using an FBX exporter included in Unity and imported into 3DsMax. The same primitives and modifiers were used for most of the boss' ground chunks and body, but there was another lingering dilemma: How would the player reach the trunk in the first place?

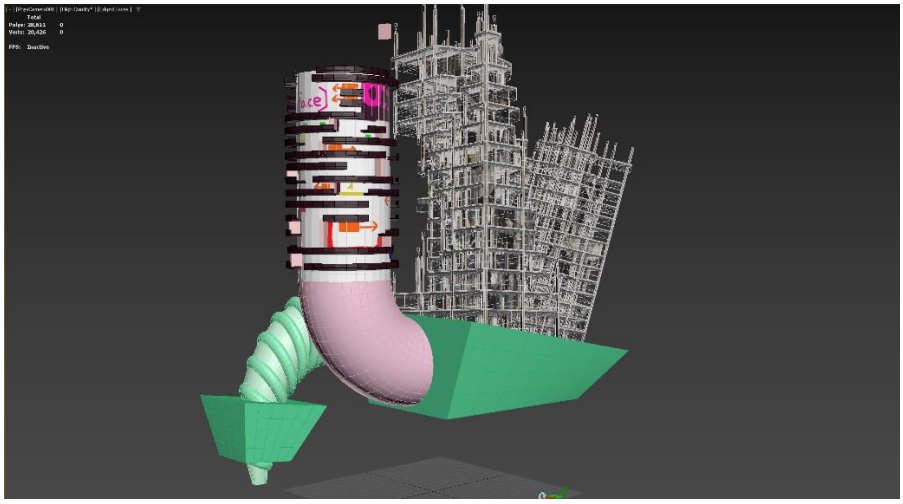

FIGURE 155. ESCÁRGOTREE'S BODY AND LEG ELEMENTS

At first, it was inevitable that the roots had to be used partly as platforms. However, what was somewhat undisclosed was the interconnection between each part —an initial idea explored in the concept discovery involved using dirt chunks or mushroom floors. However, several nuisances could appear if this idea was carried out. The author wanted to keep the platforming section contained to the trunk to maintain its rhythm and flow, and adding platforming to the legs meant accounting for it in the level's cadence. Adding to that the trouble it would generate having to move all the collision boxes of those parts with the leg's mesh, that those colliders must line up with the animation so that the player can jump on them, and that there should also be chunks like it around the other legs that could potentially become troublesome in the rigging and skinning process, it just was not worth the risk.

After tinkering with the program a bit, a brilliant idea occurred: based on the bonsai's twisted design and some thumbnails, the user used splines to create tubular shapes that could pass as vines or roots intertwining with the ground. In 3DsMax, splines are linear guides that can be used for animation controllers or, in this case, to generate a topology with evenly distributed polygons. These objects can come in many shapes and forms, from simple lines to hand-made drawn figures. In particular, the one that caught the artist's attention was the helix, which allowed for the creation of outlines in spring with customizable characteristics like the radius of the beginning and end of such, the number of turns from side to side, and the direction of the spiral. With these tools and the other modifiers mentioned, the author created vine-like structures that twisted around the body, similar to roots twisting around the leg. These forms would become a motif for all future plant-like elements of the boss.

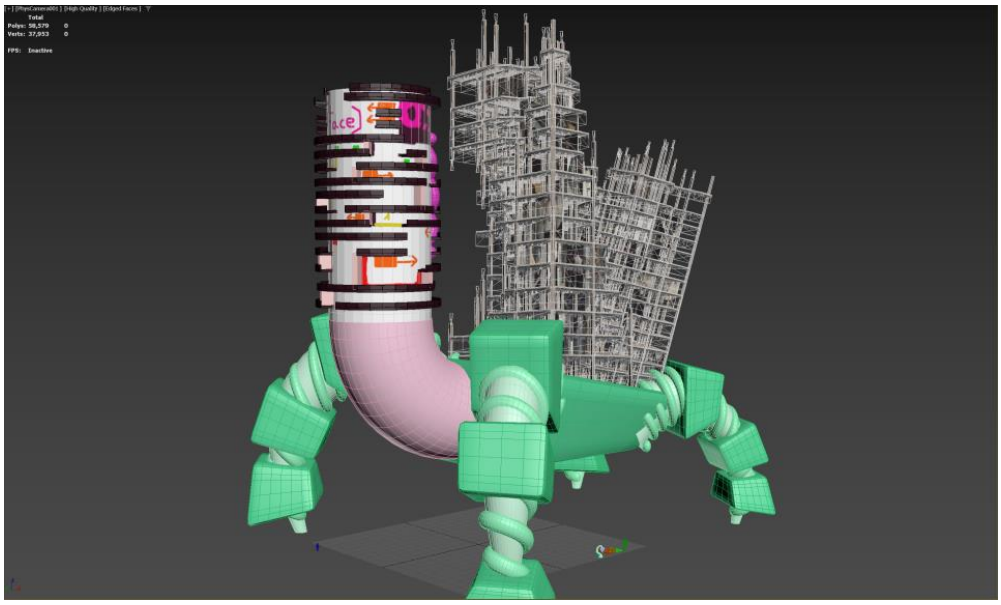

FIGURE 156. ESCÁRGOTREE, WITH EACH OF ITS LEGS PLACED DOWN

To remove the hard edges created by the cubes, a Bevel modifier was applied to them, which smoothened their edges ever so slightly. Regarding the legs, to respect the "Rule of Threes", one of the oldest rules in design that states that things that come in threes are more aesthetically pleasing (Ramirez, 2021), each leg has three trapezoidal chunks of dirt. Depending on whether or not a side was accessible, the biggest of the sides was either at the bottom or at the top. In this case, the initial plan was only to make one of the legs climbable — in this case, the left one at the front. Finally, to add the final touches to the trunk, the face was developed as two intertwining splines that converged into tubes with eyes at the end and a mouth with two pairs of branches similar to shrimp whiskers or claws. With this, the blockout was finished, and Escárgotree was born.

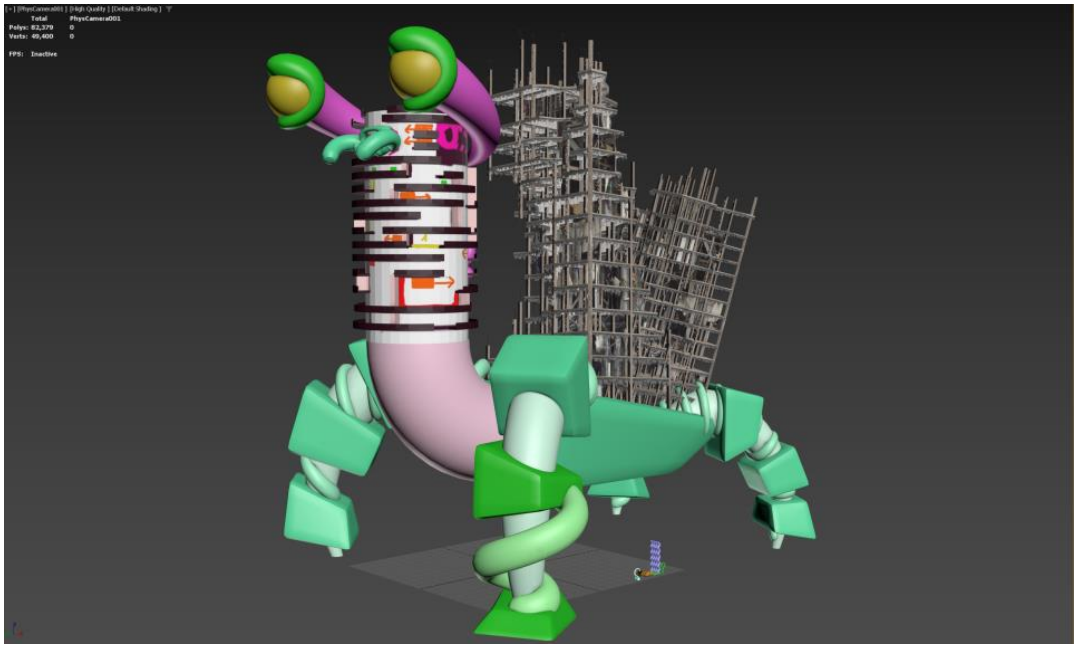

FIGURE 157. ESCÁRGOTREE'S COMPLETED BLOCKOUT

In addition, the artist created blockouts for the modular parts of the boss, such as the platforms, walls and interactable assets. Most interactable parts became vine-like and leaf-like formations with shapes representing their function. The lives became life fruits, stemming from a single vine; the keys became vine-shaped hooks with a silhouette similar to a key, and all platforms became wood surfaces that would branch out from all parts of the trunk. There was an exciting idea to make the platforms branch out from what could be the "spinal cord" of the boss, wrapping it up and popping it out of the trunk to make up the platforms. However, as the boss was open to design changes, it was best to leave the platforms as modular as possible.

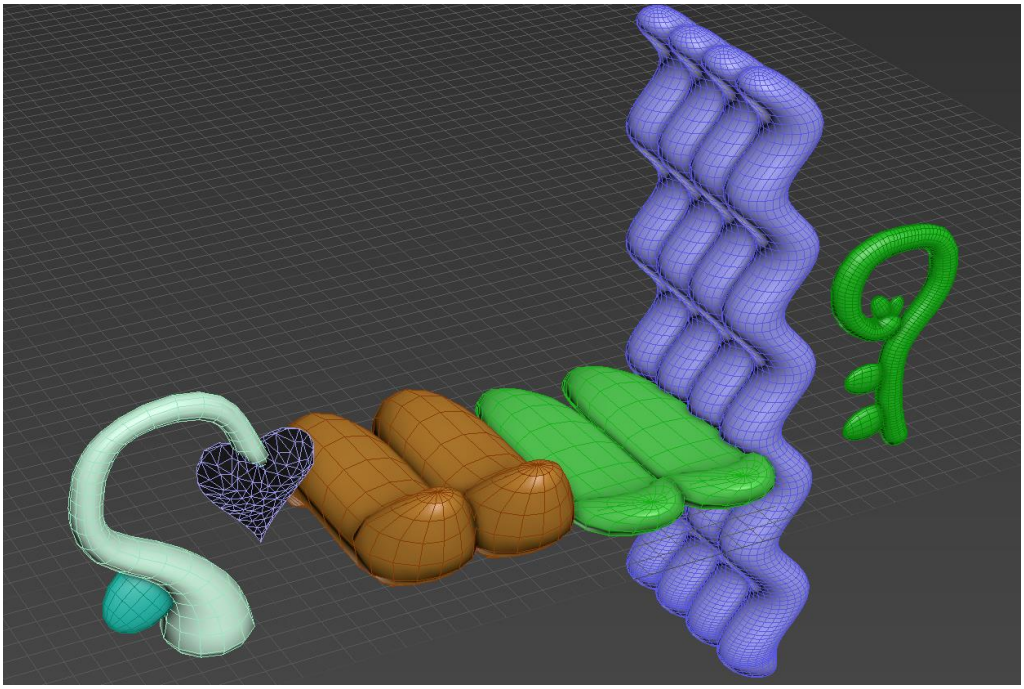

FIGURE 158. BLOCKOUT FOR THE TREE'S MODULAR PARTS.

5.2.3. Rigging & Behaviour: Preparing for the first build

Instead of moving straight to sculpting, the author created the monster's rig and some basic animations to advance in programming the boss battle. As such, it was time to import the project into Maya, where the author utilised a plugin called "Advanced Skeleton 5"[13] to create the monster's skeleton (Animation-Studios, s.f.).

To avoid modifying data from the original mesh, Maya uses a system of references that takes topological data from another file and imports it into another as a reference, which cannot be edited unless the user modifies the original file. This way, riggers, animators and modellers in big gaming studios can work in their fields separately because if some data is changed on the original file, it will be reflected in other scenes.

Hence, the first step required is creating a model reference to prepare it for rigging. Afterwards, it is time to create the bone structure. Advanced Skeleton 5 includes several auto-rigging tools that facilitate the creation of complex skeletons by following simple instructions. It also contains several joint structures already programmed by the plugin to create controllers that will facilitate the model's animations. One of these presets takes the

[13] By the time of writing this document, it has been updated to version 6.0 — simply being called "Advanced Skeleton". In this section, the author did not know about the existence of the update, but it will be applied for the following rigs in this project.

form of an insect, which includes joint chains for the legs, antennae and tail. The antennae were repurposed for the eye tubes, and some other bones were created for each building for better control. This modified basis served as the frame to create the skeleton after being accommodated into the structure.

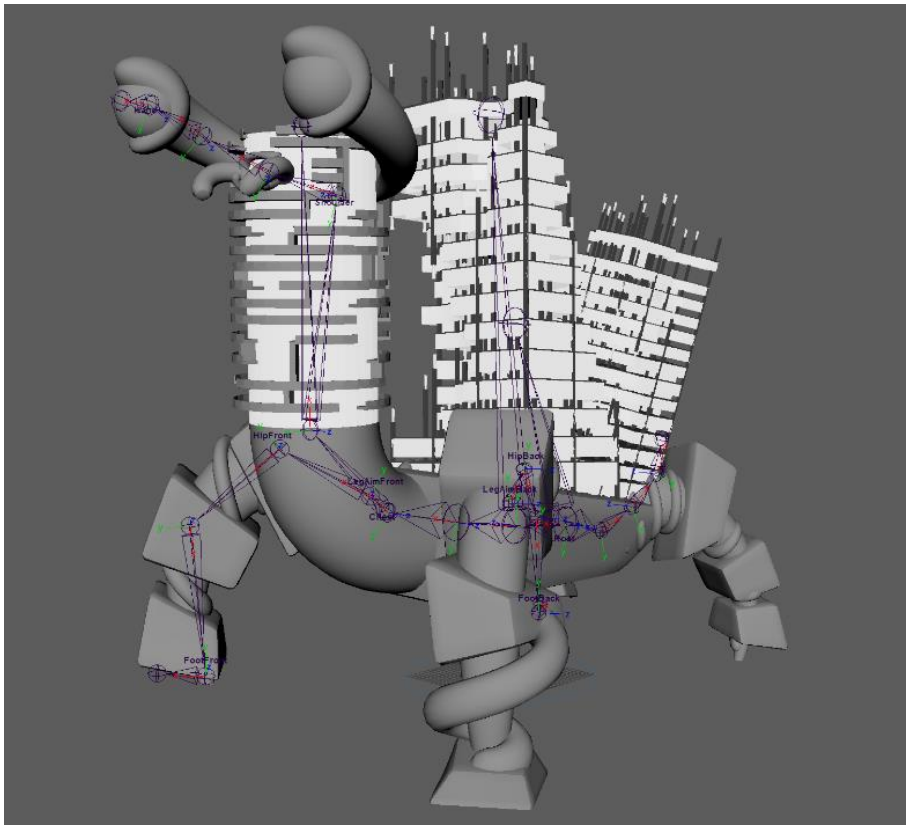

FIGURE 159. THE BASE SKELETON ACCOMMODATED INSIDE THE BOSS.

When the basis is created, the plugin automatically creates symmetric chains of joints for the other half of the mesh. It also makes controllers for each bone chain, used to move appendixes using FKs or IKs or to orient the bones in the direction needed.

After this, it commences the skinning process. This process is primarily automatic; however, many tools can make it more precise in line with the artist's desires. One of these is already included in Maya, the Geodesic Voxel Binding system, which divides the mesh into cubes (also named voxels). Based on the distance between the voxels on the skeleton and those in the topology, it calculates an automatic weight for each polygon (Autodesk Interactive, s.f.). While not perfect, this process saves up 80% of the work in most cases; however, as the skinning process was crucial to get right in this project, all help the author could harness was welcome. After painting some erroneous weights using the program's manual skinning tools, animations were created to test the monster's attacks and how the elements moved with the skeleton. For version 1.0, the artist created two basic animations for stomping and kneeling after defeat, with one leg close to the trunk to make it reachable. In the future, more controllers would be added to control mesh deformations for facial expressions like blinking or opening and closing the mouth. However, Escárgotree was now alive and roaring for a fight.

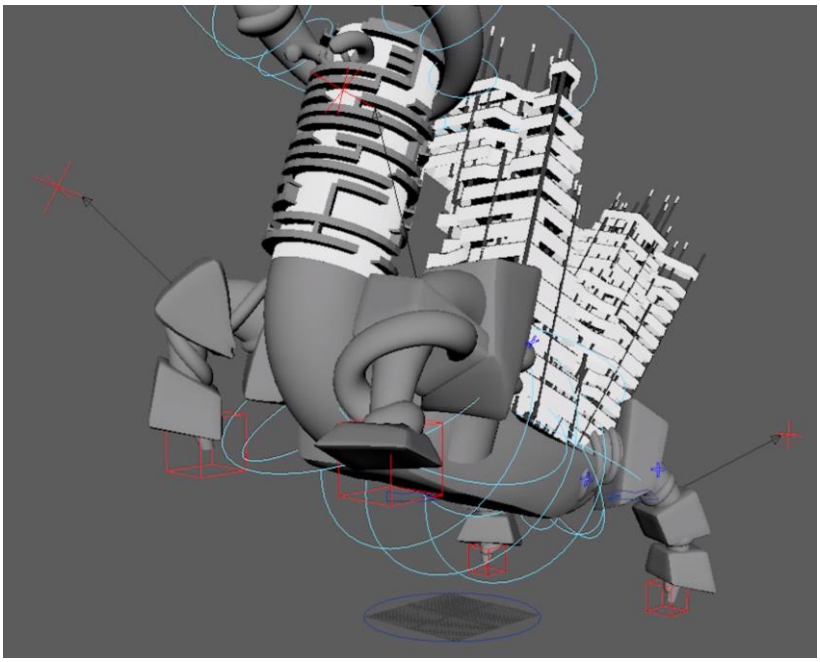

FIGURE 160. ONE OF ESCÁRGOTREE'S TEST ANIMATIONS TO SHOWCASE THE SKELETON'S CAPABILITIES AND MOVEMENTS.

The author moved on to creating the boss' behavioural patterns and attacks, as all the required models and animations were completed. The first step was putting the collision boxes, interactable items and camera behavioural triggers. Most of the inaccessible parts of the boss contain simple box or capsule colliders to avoid overloading the prototype with collision checks. However, places that would indeed have player interaction use a more complex type of collider called mesh collider, which, as the name implies, can have the form of the specified mesh. Putting each collider into place by hand is monotonous at best, but as many precautions were taken beforehand, eventually, the job gets done.

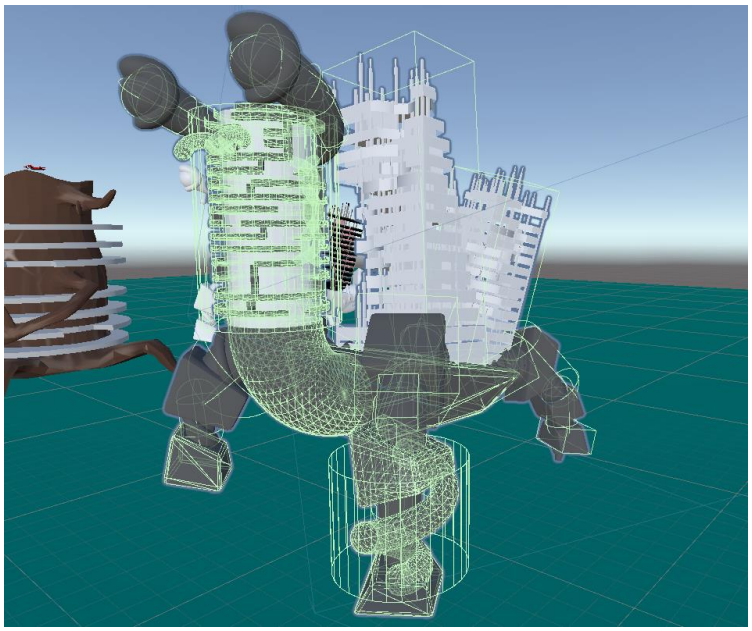

FIGURE 161. THE FIRST ITERATION OF THE COLLIDER PLACEMENT IN THE BEAST

When starting with the camera behavioural triggers, several significant issues appeared. Some had easy fixes (like camera culling problems and additional scripting that had to be adapted for the new model and attack patterns being developed during the process); however, the most crucial problems were related to the beast's animations. The first was focused on the climbable leg: managing the monster's animations when kneeling and colliders was more challenging than expected, as some mesh colliders and their skinned mesh counterparts needed to line up. This problem was solved by flooding the skin weight of the leg's elements in the animation, but it also meant having to redo all the collision placement from the ground up.

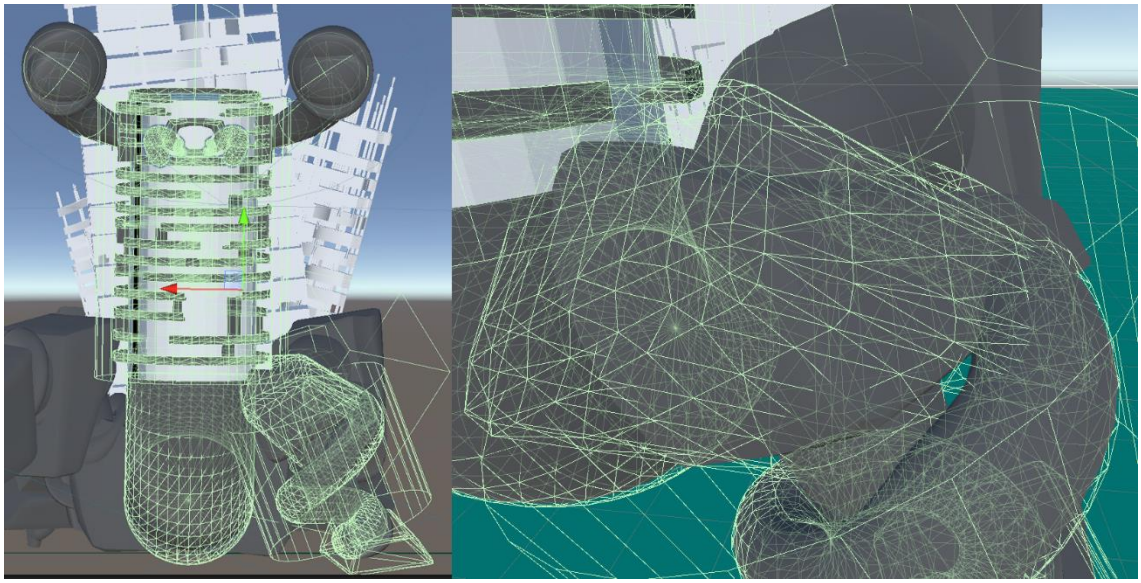

FIGURE 162. EXAMPLE OF LEG AND TRUNK BUGS WITH COLLISION BOXES WHEN KNEELING

Another issue was creating the 2.5D level. The initial animation was not looking straight up on its first iterations, which meant the player could easily fall despite the small barrier created in each of the disk's platforms. To exacerbate, the minion's behaviour was overly glitchy due to the number of collision meshes surrounding them and because its behaviour allowed it to leave the part of the level they were supposed to act upon and thus airstrike the player from many directions, something that was not planned for the level. A partial solution at the time for both of these problems was to create a cylindrical barrier around the trunk that allowed the player to be unable to leave the boss. For the most part, it worked, but these issues persisted for many weeks due to how the minion shooter scripts were programmed at the time. Other issues with unsurpassable obstacles, such as spikes that could not be jumped over, were solved by changing their behaviour and role. In this case, the obstacle was changed to moving spikes, making the challenge time-based.

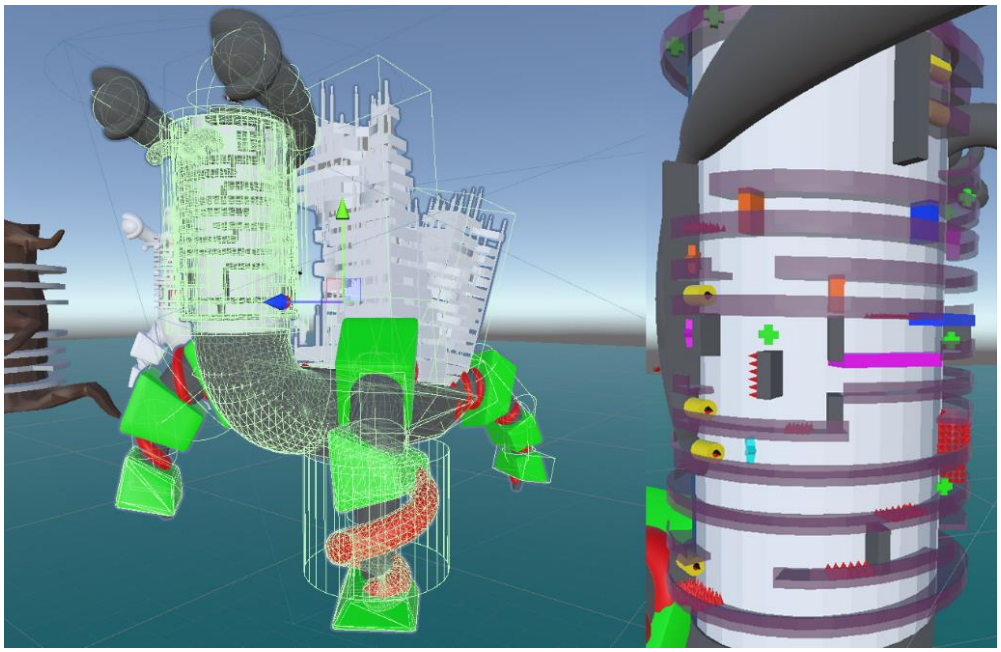

FIGURE 163. THE SECOND ITERATION OF COLLIDER PLACEMENT, WITH THE INTERACTABLE ELEMENTS IN PLACE

When the trunk was done, it was time for other behavioural patterns, such as the stomping, the shockwaves and the sub-boss. Without entering into too much detail, for the pulsations, the legs created a preemptive red zone based on the leg's height, which would disappear after striking down, in which a shockwave spawned. Similar behaviour was added for the sub-boss, but instead, their behaviour was attached to its lunging attacks. These were relatively easy but still issued some problems, especially regarding player feedback bugs with materials and the shockwave collision boxes. These were unsolvable problems at the time due to needing more indispensable assets to function, but they did not hinder much in the programming.

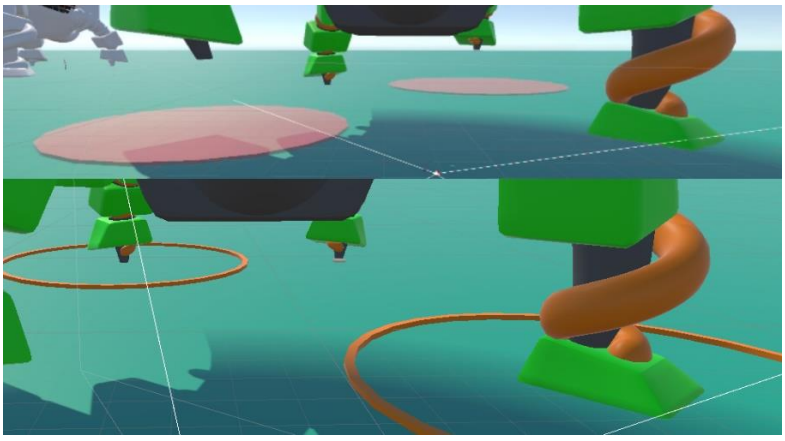

FIGURE 164. SHOCKWAVES IN ACTION

By this time, the author commented on their design teacher Joan Josep Pons López about the project and suggested doing a playtesting session of the prototype's state to detect and fix bugs, test playability, and get some information and feedback. For this, the writer added some game-feel elements like life bars, created a build with an additional script that stores placement data of many actions that could be done in-game, and

constructed a Unity scene that could create a heatmap based on that data, which would help gather extra data. With some additional tweaking in the movement and more bug fixing when placing the boss inside the final environment, version 1.0 of the build was prepared, and it was time for the playtesting session.

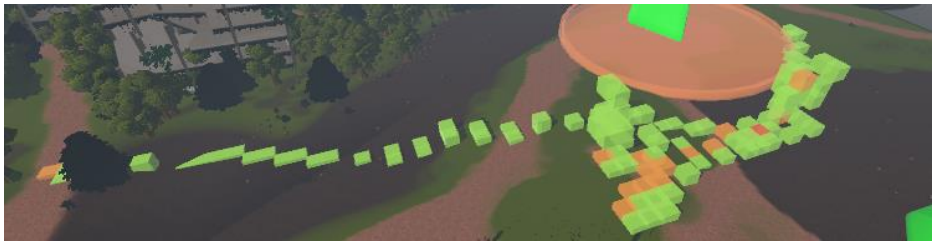

FIGURE 165. HEATMAP GENERATION TESTING

5.2.4. First Playtesting Session: Revamping the boss

The playtesting session was conducted on the 18th of April, 2023 and lasted the entirety of the Level Design class, that is, two hours. In this testing session, 12 students and the teacher tried this build. The data recollected from this session involved the heatmap data collection and additional notes and voice recordings of the participant's comments and feedback. It was an eye-opening experience that helped solve many bugs and mistakes in the playable scheme.

The session began with blind testing of the game. Some players already knew about the goals in this game, but most people felt lost in the first part of the battle. They did not know what to do; they found many lingering bugs the author could not solve, and some even broke the game in ways I could not imagine. Adding the fact that the player's camera felt wrong to all the members, as it was inverted, imitating aviation controls, it was clear that the experience required polishing. As seen in the heatmap data, players were all over the place, not knowing what path to follow or how to do specific tasks and reaching places they were not supposed to because of the monster's affordability from its animations.

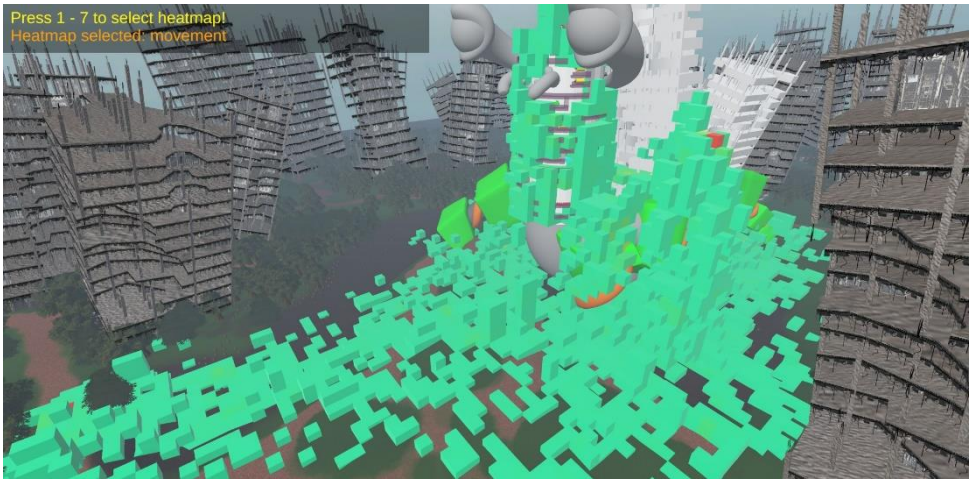

FIGURE 166. HEATMAP GENERATED WITH PLAYER POSITIONS

Despite that, to have constructive feedback beyond playability issues, they observed how the author played the game in its intended way, the thinking process he followed to reach the previous conclusions and all the

information explained in the previous sections. With this contextual information, the testers used it in the following weeks to improve the battle.

Firstly, regarding bugs mentioned in previous passages, many were solved with the help of the aspiring programmers from that class. These included the shockwave and minion behaviours, two of the most troublesome bugs from the prior section, and other undetected ones related to the spawning system and other poorly placed colliders requiring attention. The author also added an options window to change camera settings such as orientation and sensitivity.

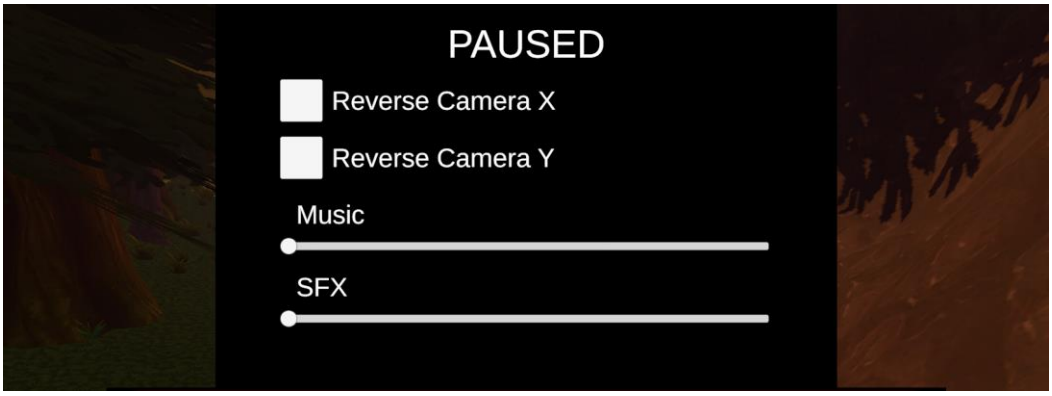

FIGURE 167. PAUSE MENU SCREEN AS OF VERSION 2.0

Another necessary change was the renderer pipeline. Unity has preset shaders that create basic illumination for most projects; however, to add visual effects and other elements such as decals (essentially projections of 2D images into 3D surfaces, working as graffiti) that would be needed to achieve the changes presented, a new shader pipeline had to be installed. This process requires all the materials involved to use the shaders provided by the pipeline, which, for this project, ended up being the Universal Rendering Pipeline. This shader pipeline is simple to use and optimal for the project's requirements, allowing for similar results in lower gamma hardware compared to the High Definition Rendering Pipeline.

After this, going from phase to phase, there was a consensus that there was a need for player guidance and feedback from most elements. During the preproduction phase, one of the many ideas to improve the player's experience was creating a playground zone or level 0 that taught the player the tools they must use to kill the beast. This concept was discarded to favour the boss' creation; however, as seen in this playtesting session, it was a necessary feature. Hence, a small section of the map was dedicated to teaching players the controls in the minimum time and space possible to keep up with the pre-established schedule.

The playground was designed to hold and showcase most of the mechanics that would appear in battle in mind. The players had to learn about double jumping and the shockwaves they would encounter in battle, so the author placed part of a fallen building in the middle of the road with a radio antenna that emits pulses that, rather than hurting the player, push them aside, becoming an easily avoidable hindrance. They also needed to learn what to hit in the first phase, so a little dummy was created that would get destroyed when attacked thrice quickly. This dummy's function is to teach players about the ability to attack, the triple-hit combo and what targets they can attack with their sword, all in one concise section. With all these tools, they were now ready to

face the boss. For players not to miss it, it was placed in an illuminated area with a single shadow pointing towards it as a small indicator to guide them.

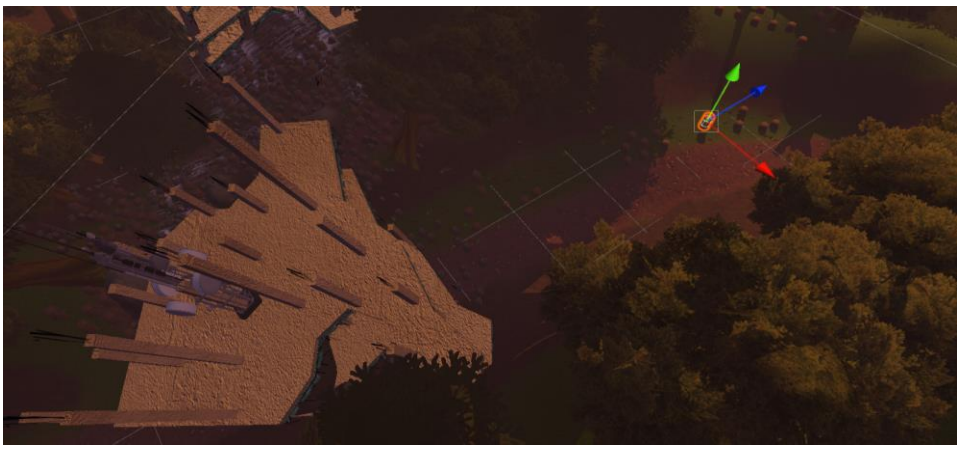

FIGURE 168. THE PLAYGROUND'S ASSETS AND OBSTACLES, AS OF VERSION 2.0

The first phase received changes from its first iteration, mainly on the terrain. They noted how the boss had most of its elements be extra-diegetic, these being how players can predict when the beast will attack. The area of effect was made out of elements that initially did not form part of the level but instead were only meant to fulfil their goal. To solve this, the terrain was adapted to mark the pulsations' area of effect on the ground, as if the shockwave had destroyed those areas in particular. A decal was used instead of a 3d model to add a preemptive shadow to the attack, which was a more elegant solution overall. To improve the player's guidance across the first action beats of the battle, the writer slightly shifted the boss' position so that the player had the front leg as the closest objective to reach. The author also elevated the terrain to better observe the situation and the boss' behaviour before entering into battle, creating a barrier that divided the playground from the battlefield area. All of these elements instantly enhanced the player's experience vastly.

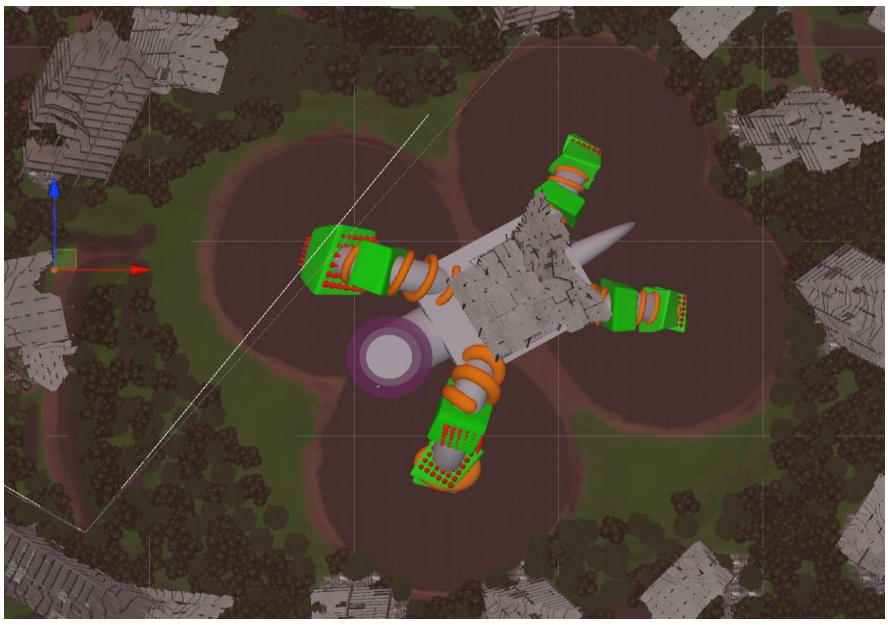

FIGURE 169. TOP-DOWN VIEW OF THE 2ND ITERATION OF THE BOSS' ENVIRONMENT

Before the second phase, to solve the issue of roaming players without much guidance, the writer added negative affordance with spikes and some changes in the blockout's shapes to impede players from climbing sections they are not supposed to. Whenever the legs are destroyed, some of the spikes for the climbable leg disappear, allowing users to climb to the tree. The initial idea to get to the 2.5D section was to go around the creature's knee to get on top; however, players would instead climb on top of the root, thus accessing parts of the level earlier than intended. To favour this instinct, the artist modified the level's access method by enlarging the trunk and making it mandatory to climb the knee and double jump onto the platforming level. The bottom disk was also enhanced so players could effortlessly get on top.

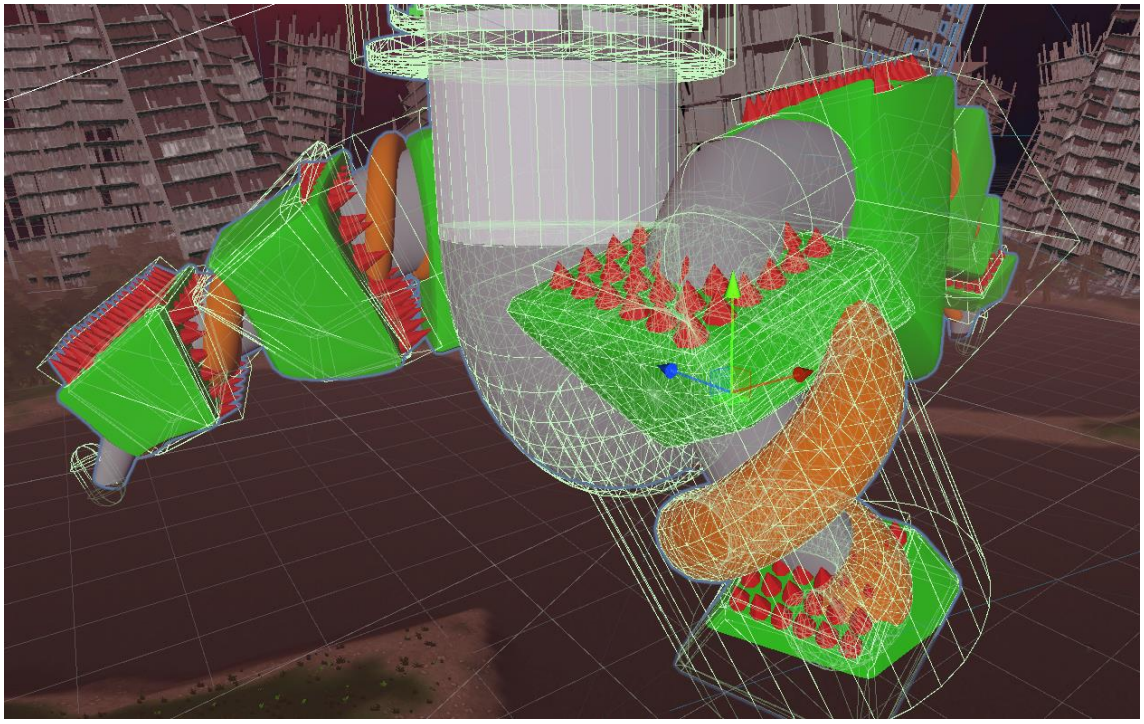

FIGURE 170. THE RESULTING LEGS AFTER ADDING NEGATIVE AFFORDABILITY

When analysing the level with the cadence scheme, most of them detected some sections that were harshly calibrated. Too many dangerous elements in the first sections had to be changed to give a safe space after the boss' first phase. As such, the first two disks were changed to have nominal obstacles that rewarded the player with lives so they could heal in case they arrived at the second phase with one life remaining (which was mostly the case). To add interest to the level, the minion's behaviour was changed to add a simpler version that did not chase down the player in the earlier sections and to make it easier for players to tell when the cannons would shoot, a charging animation was added. All of these changes that have been carried out since creating the boss behaviour right after the rig required a revamp of the level's cadence, adding a difficulty graph that examined the progression and how it added up to the flow level.

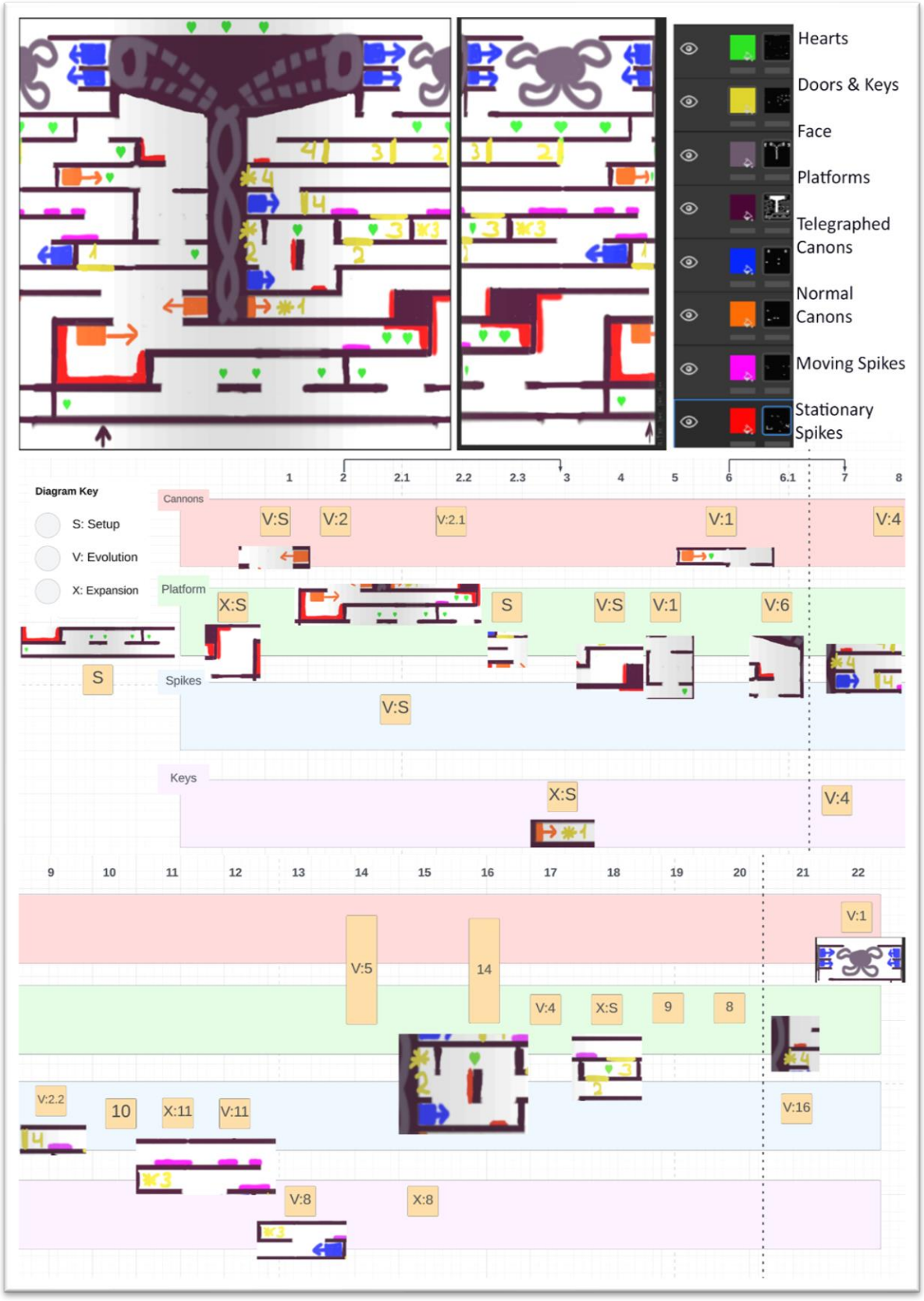

FIGURE 171. 2ND ITERATION OF THE 2.5D SECTION OF ESCÁRGOTREE WITH AN ADDED CADENCE ANALYSIS

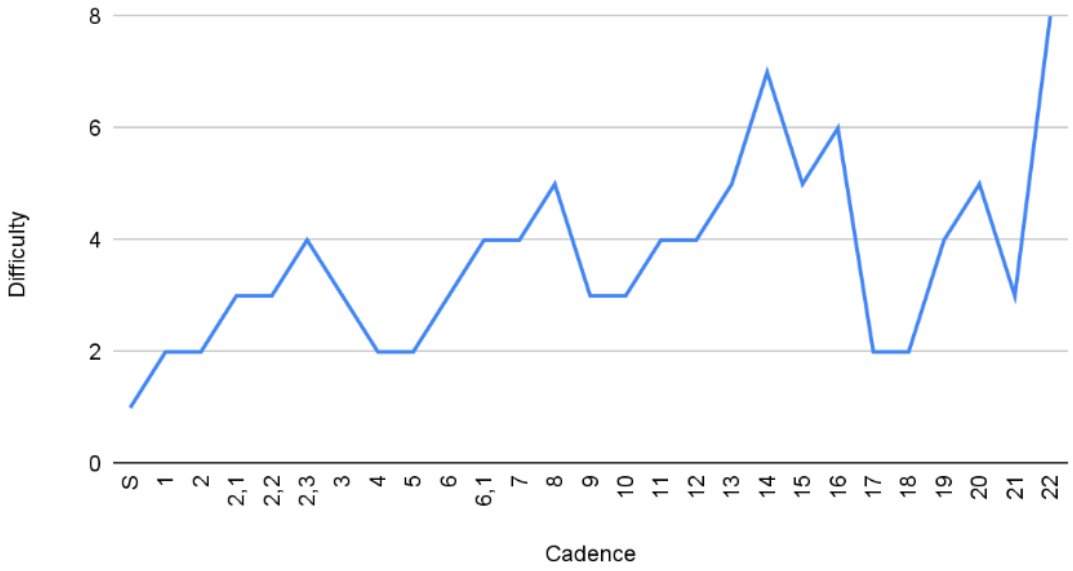

FIGURE 172. GRAPH RELATING THE 2.5D LEVEL'S DIFFICULTY LEVELS WITH EACH SECTION OF THE CADENCE

Finally, the sub-boss battle would also receive changes for the third phase. Its attacks became better telegraphed by preparing a small backing animation before it lunges towards the player, the preemptive attack zone was changed to a shadow decal like the legs, and the shockwaves it generated were faster and covered the entire top, meaning that the only way to avoid them was to jump over them, thus tying the boss towards the skill themes presented in the labyrinth. The boss layout also changed to have higher barriers so players could not leave the arena as smoothly, and in the form of a nest for thematical purposes.

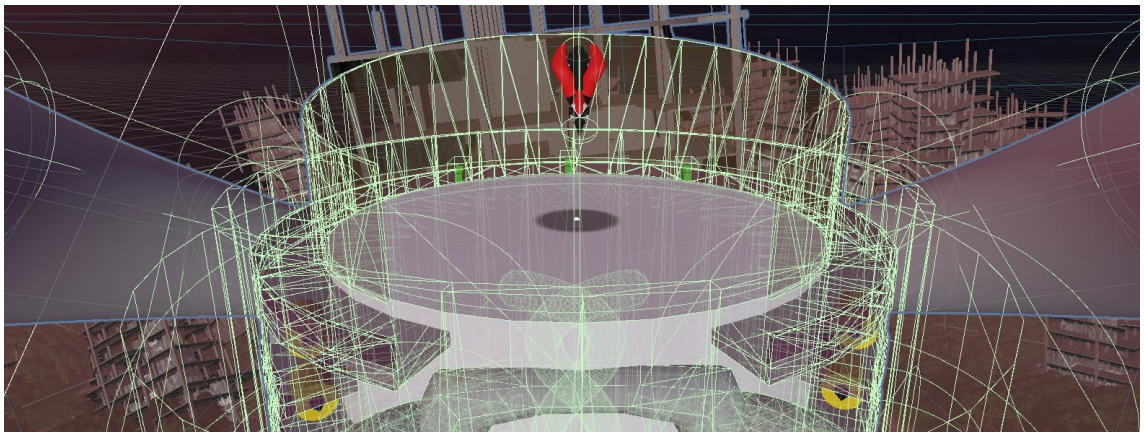

FIGURE 173. THE SUB-BOSS NEW ARENA FOR VERSION 2.0

Many other changes were suggested that would be added in the following paragraphs or that did not make it to the final cut due to time constraints. Nevertheless, version 2.0 was a groundbreaking point in the project's development, allowing the artist to start sculpting the assets that would replace the blocks in the level.

5.2.5. Sculpting: Bark, vines and beanstalks

This part of the process was one of the most troublesome regarding time constraints and creative freedom and restrictions simultaneously. Due to the author's inability to create quality drawings and the simplicity of the concept art's final drawings, his course in creating all the meshes' details was iterative. This approach implies that simpler models had to be done first to find all the tools in ZBrush and general directions and parameters used in each giant monster's meshes, with general inspiration taken from other references to convey the original objective of a stylized look. Most of this project's importance lies within this part, so it is crucial to get it right and dedicate as much time as necessary. As such, the first models sculpted for this project were those involved in the trunk or, in other words, the platforms, walls, spikes and interactable elements of the boss fight.

The general process for model creation follows the same structure. Firstly, the model's shape is created in relatively low resolution to create the silhouette and volumes of the model. Using Dynamesh, a tool that recreates the process of clay modelling, allows the preservation of the same polygon density across all extrusions and modifications by recreating the model with a uniformly distributed topology. The objective is not to get into too much detail but instead generate a base from which to add detail afterwards.

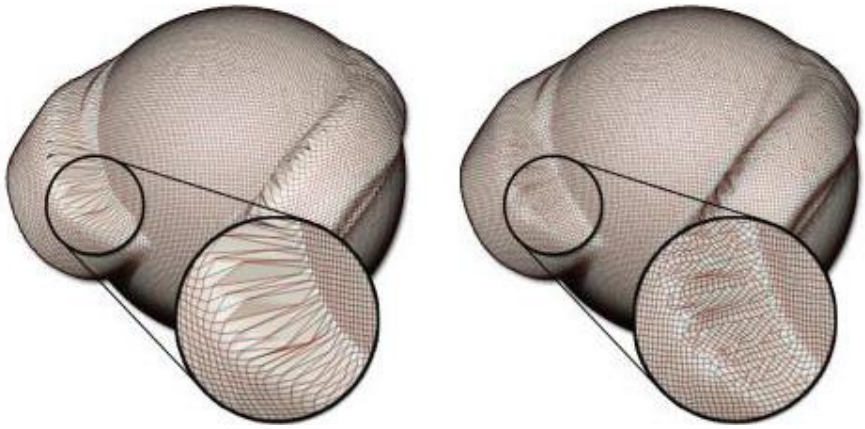

FIGURE 174. ON THE LEFT, STRETCHED POLYGONS. ON THE RIGHT, RETOPOLOGY CREATED USING DYNAMESH

Many brushes were used in sculpting all models, both included in ZBrush base data and downloaded for the internet for stylized modelling. Some of the main brushes used to manipulate the model include Move, which moves polygons; Clay, ClayBuildup and Inflat for volume generation; SnakeHook for branch-like extrusions; Smooth to smoothen the volumes between one another; TrimDynamic for creating flat surfaces; Pinch for creating sharpened borders; and the Orb brush pack created by Michael Vicente. Overall, the author used various tools to create the stylized look marked as the objective.

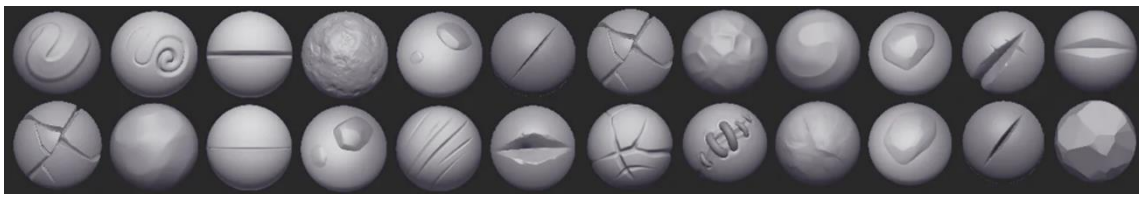

FIGURE 175. BRUSHES INCLUDED IN MICHAEL VICENTE'S ORB BRUSH PACK

After the model has the desired physique, it is time to give it details. ZBrush allows creating different levels of detail with subdivisions of the same model, allowing to approach the model from general, more visible attributes to the most minute of details like the pores of a face or, in the case of this project, the venation of a leaf. Firstly, the writer has to create a mid-poly version of the shape, which will serve as the first level of detail. The quickest method is ZBrush's ZRemesher, which retopologizes automatically by imputing parameters and the amount of optimization to be created. With this first version, the artist divides the topology into several levels, from the thousands of polygons to the millions, for each of the sculpture's minutiae.

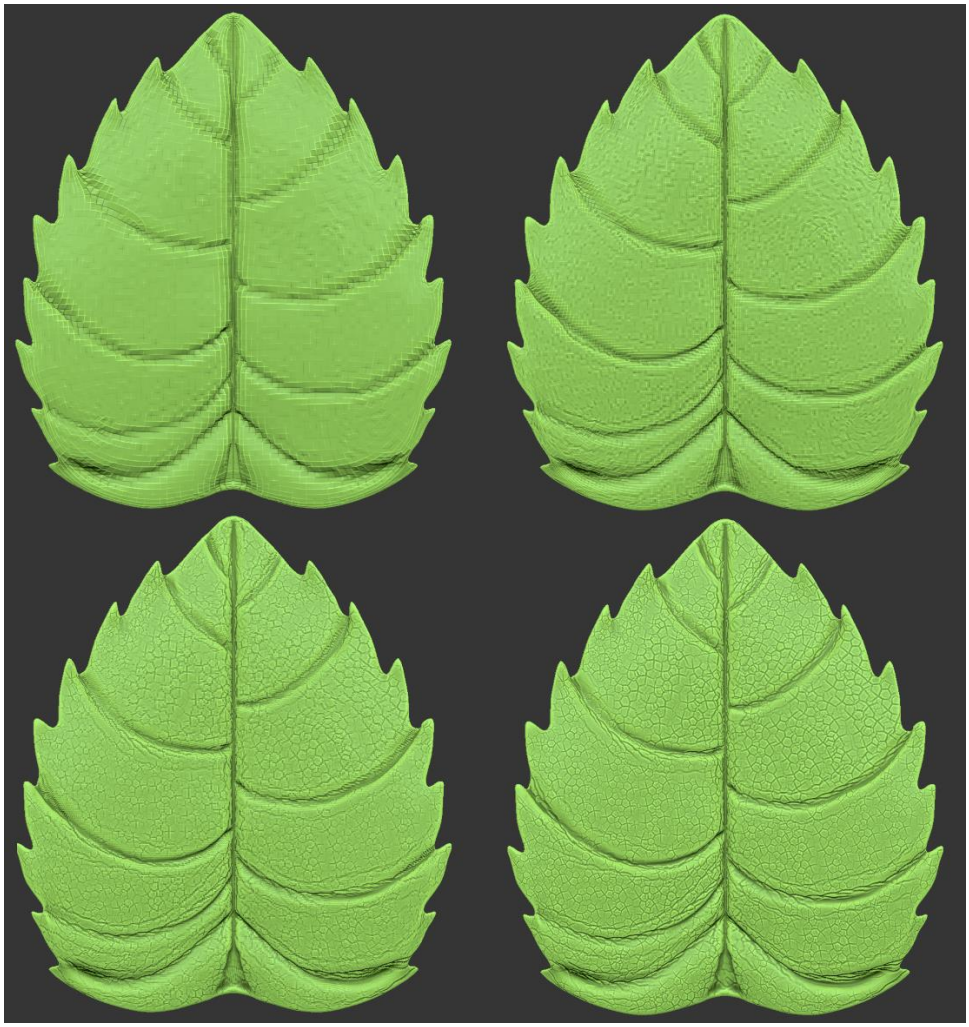

FIGURE 176. DIFFERENT LEVELS OF DETAIL OF ONE OF THE LEAF MODELS

This process is well exemplified with the first models created: the leaves. All the vegetation in this level shares the same leaf type: one inspired by the general shape of bonsai leaves but with the spikiness margin found in many Catalan trees. Technically speaking, this cordate shape with a serrate margin (Mukherjee, 2023) (Masons, 2021) gave the trees an aggressive, pointy structure, accentuated by the leaves' arcuate and reticulate venation. As the leaves could have many sizes, a significant limitation that occurred with them is that the venation gets lost if the leaves appear small in the camera, resulting in some image blurriness that should be treated in texturing. The artist created two similar versions to have variability in the vegetation.

The following model produced was the life fruits, heart-shaped giant berries that heal the player when eaten. As a reference, the artist searched for similar plants in other games, most notably Terraria, and even their takes on this exact concept by other artists. Inspired by the twisting volumes around Escárgotree's leg, the author used vines as a starting point. Two variations of the same concept were done by creating spiralling forms around a structural stalk, with roots grounding the plant on one extreme and the fruit dangling on the other. A specific tool used for the heart is the Noise tool, which allows for creating porosity and coarse surfaces on objects, using a Perlin noise map (an image with pixels of semi-random values on a scale of black to white) as a reference. This tool's purpose, in this case, was to give the berry a hairy and porous feel to imitate a peach's surface.

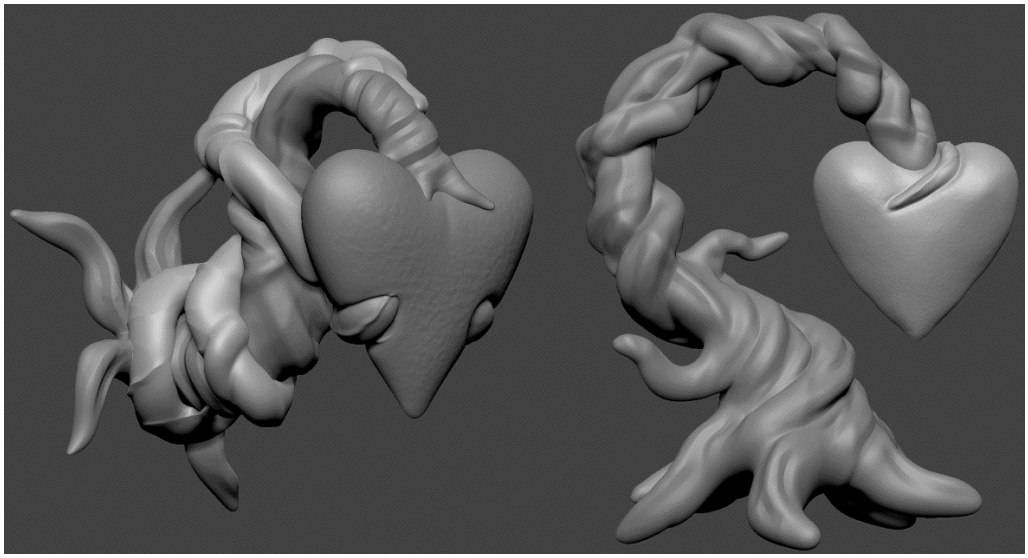

FIGURE 177. VARIATIONS OF THE LIFE FRUIT PLANT

With all these tools learned, more modular assets were sculpted. Following a structure similar to the previous models' leaves and vines, the keys and gates were also plant-like. Version 2.0 of the prototype had these models be simple blocks and stars. However, the blockout that would supplant these plants has an outline resembling traditional handheld keys, with a circular handle and two leaves as bits. For the sculpting, the key had a vine-like twisting shape resembling the life fruits, and the doors were shaped as two leaves combined in the middle, resembling a mantle or a tongue.

FIGURE 178. LEAF DOOR & VINE KEY SCULPTURES

Afterwards were the spikes. There were many possibilities, like spiked wood chunks, bushes, and twisted twigs. Nevertheless, as it had to be on multiple surfaces and in different sizes, the best outcome was using giant spikey brambles. Four variations were created with shapes resembling a simple spike, a hook, a half-arc and a combined amalgamation, all with the same twisting relief over it, to add modularity and randomness to these obstacles.

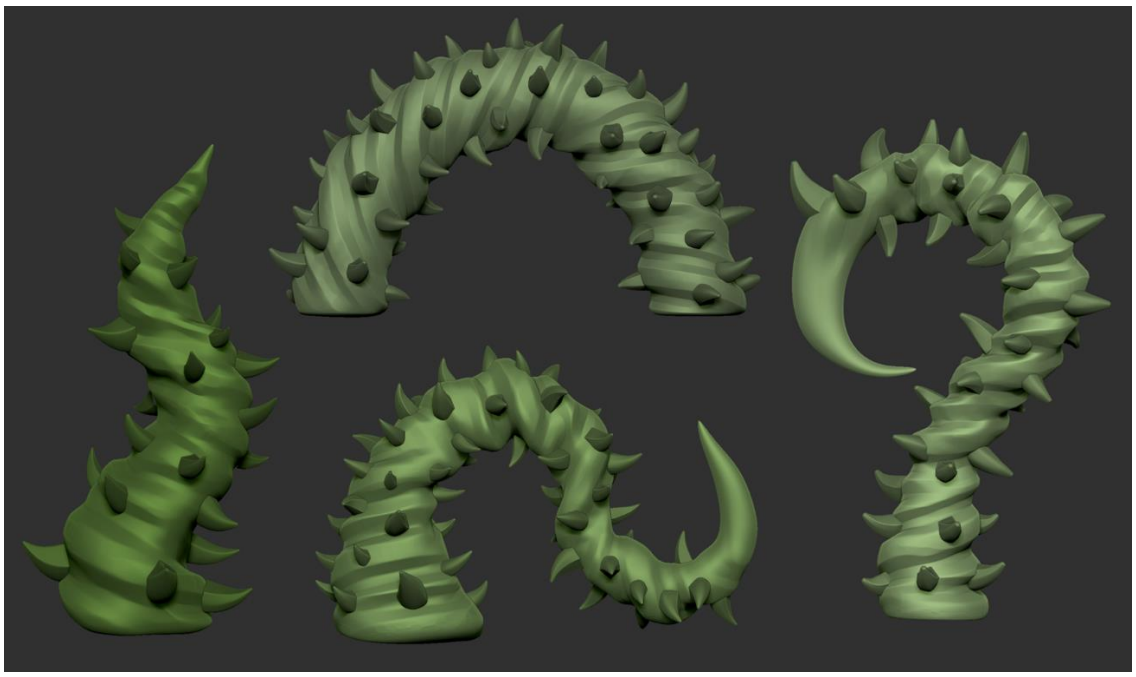

FIGURE 179. ALL BRAMBLE VARIATIONS

The problem came with the spikes, which initially seemed to be a time-taxing task. Luckily, ZBrush has tools for automatically placing assets on polygons named NanoMesh and MicroMesh. For this project, the former was used to place the same asset across the topology with the ability to adjust parameters for random arrangements. While the general distribution of barbs had to be adjusted manually at times, this tool was practical, resulting in one of the most used tools in this project.

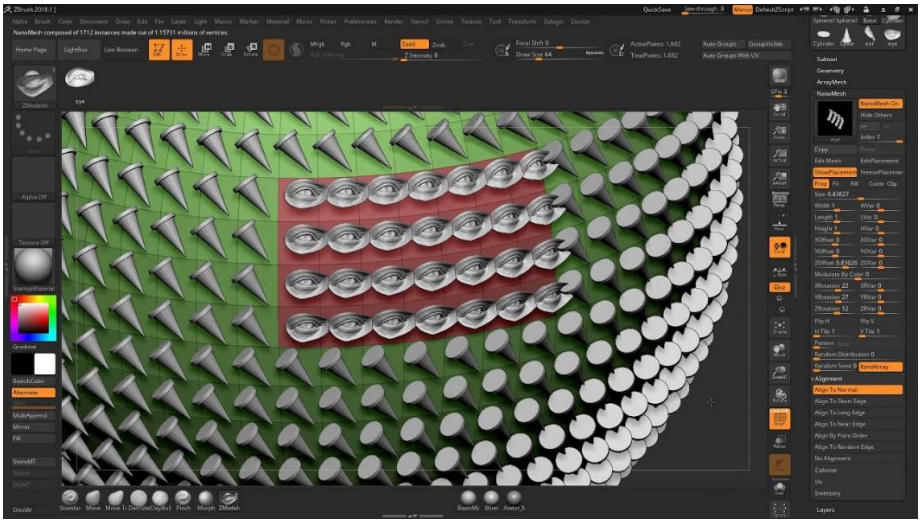

FIGURE 180. NANOMESH IN ACTION WITH TWO DIFFERENT GROUPS

Another usage of the NanoMesh was with the minion cannon. The initial idea was nebulous at best, being a simple hollow stump stemming from Escárgotree's walls and platforms. However, some of these artillery ended up in multiple directions, incompatible with the original design. Therefore, a new approach was taken. The minions appear from inside the gun as eggs waiting to burst on a nest when the player gets nearby, aimed towards the direction they will take and with a branch indicating such orientation. These nests used NanoMesh to create small branches on the surface to resemble small sticks or hay. While some details may become unnoticeable at a distance, the writer deemed this method the most suitable for the project's schedule.

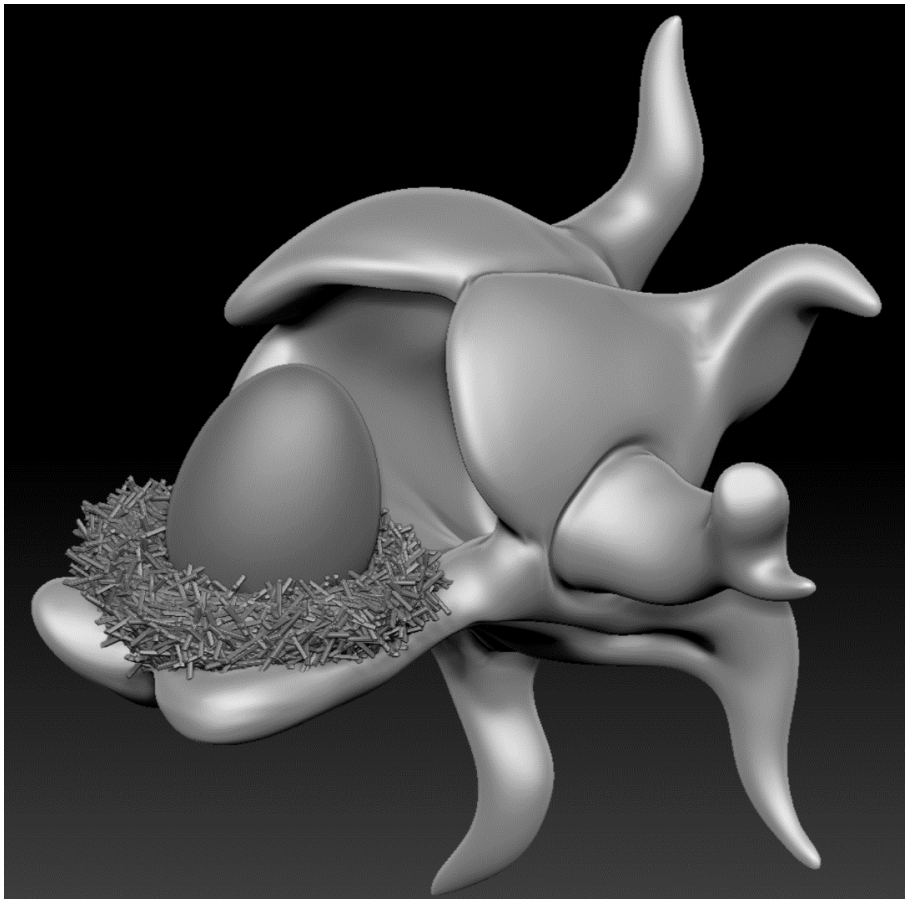

FIGURE 181. EARLY VERSION OF THE MINION CANNON

For the bark, it was time to introduce two new tools to the mix: the Masking tool and the Deformation window. Masking lets sculptors delimit areas as uneditable, allowing them to save certain regions' original topology and change the surrounding ones. Combined with the deformation window, which applies some brush effects like Inflate or Smooth but in a generalized manner, these tools were perfect to create the bark's surface.

Applying alpha textures (or, in other words, black and white images with detailed graphics) from Rafał Urbański's 20 Wood Stylized pack[14], it was simple to create intricate patterns in the wood. The exact process was also employed to create the five variations used for the platforms and the dummy figure, including an image of a bird carved on top. An additional branch could be placed to add more variations to the modules.

[14] You can find it here: https://www.artstation.com/artwork/3dLyXm

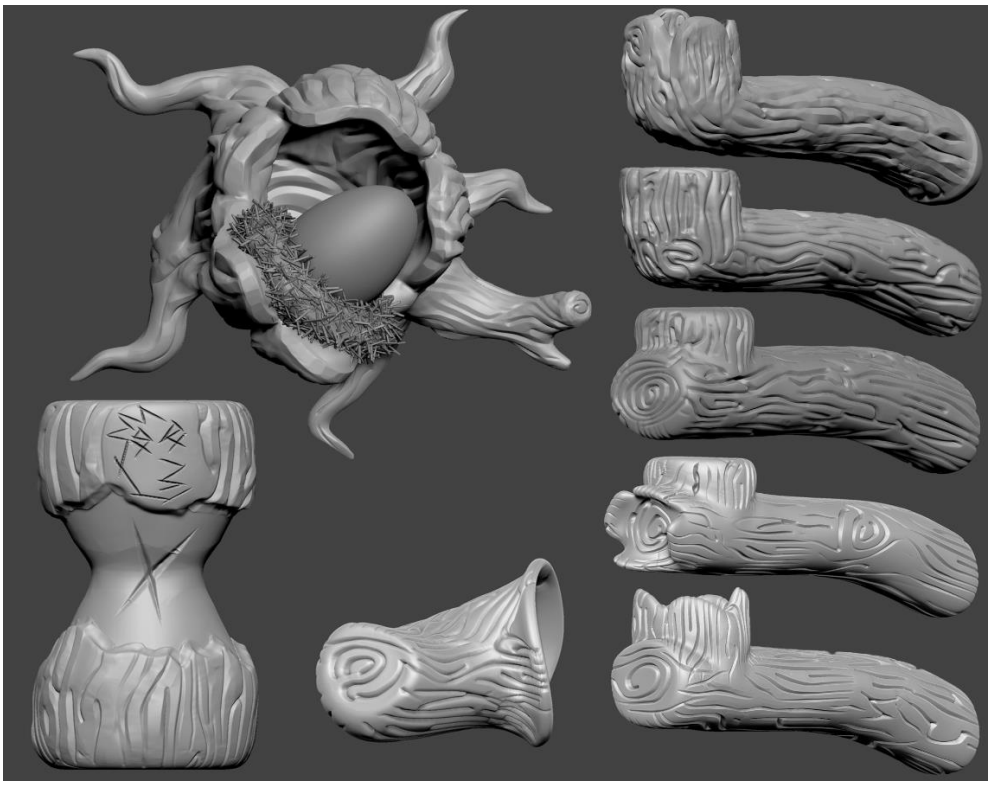

FIGURE 182. WOODEN MODULES AND STUDIES OF CANNONS AND PLATFORMS

One of the most challenging ideas to figure out was the walls. The original blockout was initially defined as serpentine trunks, much like the platforms. Regardless, after much testing and putting into perspective, it did not work well with the other modules, so they had to go in another direction. In the end, the best-looking wall combines many of the already created vegetation modules like leaves and brambles, which gives the idea of a wild bush. Due to optimization issues, this idea required the walls to be done in the future. Nevertheless, the author developed a proof of concept to serve as a reference for when the time comes.

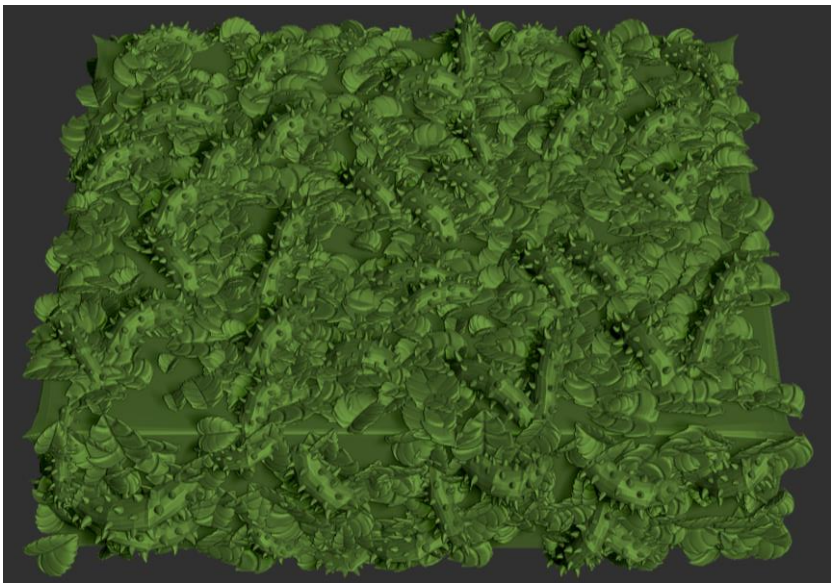

FIGURE 183. VINE WALL CONCEPT RECREATED IN ZBRUSH

To add final touches to the overall look of the modules, the artist used an automatic polishing tool called ClayPolish, which sharpens a model's edges and smooths its surfaces, giving a final layer of polishment emphasising the stylized point of view. With all the modules finished and many lessons learned, the sculptor moved on to Escárgotree proper.

Firstly, the author modelled the more defined parts of the design as the other concepts for the sculpture were coming to fruition. Starting from the top, it was decided earlier in the concept phase that the sub-boss arena would take place in the bird's nest, so the artist used NanoMesh to fill a croissant-shaped mesh with many austere branches, similar to the nests found in the cannons. Surrounding this roost, the trunk's bark was extruded to give it the shape of a stump, following the same process of moving and fine-tuning with smoothing brushes as the many platforms surrounding the structure. For the mouth, the claws were horned, and the teeth were sharpened and extruded. Combined with hollowing the back of the mouth to give it depth, the beast was already becoming more threatening as more parts were sculpted.

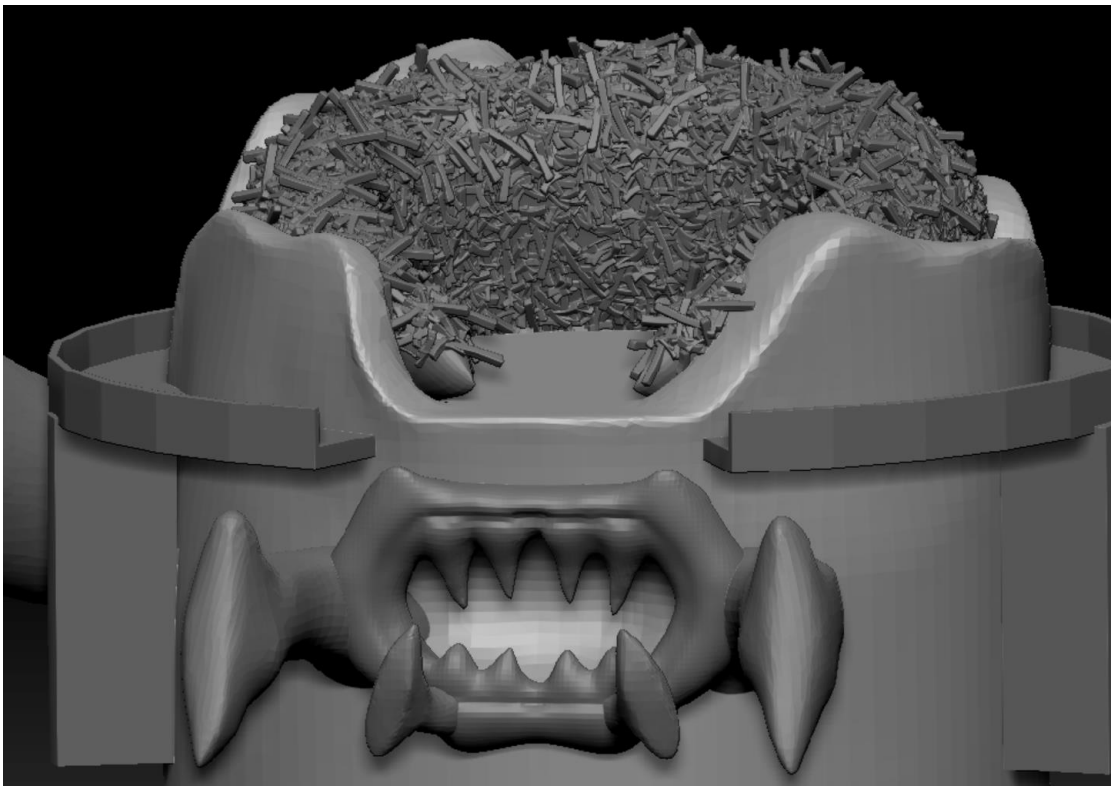

FIGURE 184. ESCÁRGOTREE'S FACE CLOSE-UP

Following down with the legs, the bottom had the weak spots marked just like the dummy, and each chunk incrusted in these had the same sculpting process applied to them. Using ClayBuildup, the sculptor raised all block's borders and top and bottom layers to create cavities and mud-like irregularities. These cavities were later solidified and elaborately defined with the same brush. To smoothen the blocks, the student combined the Smooth and Clay brushes to give the dirt a more stylized look while also being coarse. The top chunks also had roots from the vegetation ingrained in them to emphasize that the creature had risen from the ground, further giving variety and interest to the sculpture.

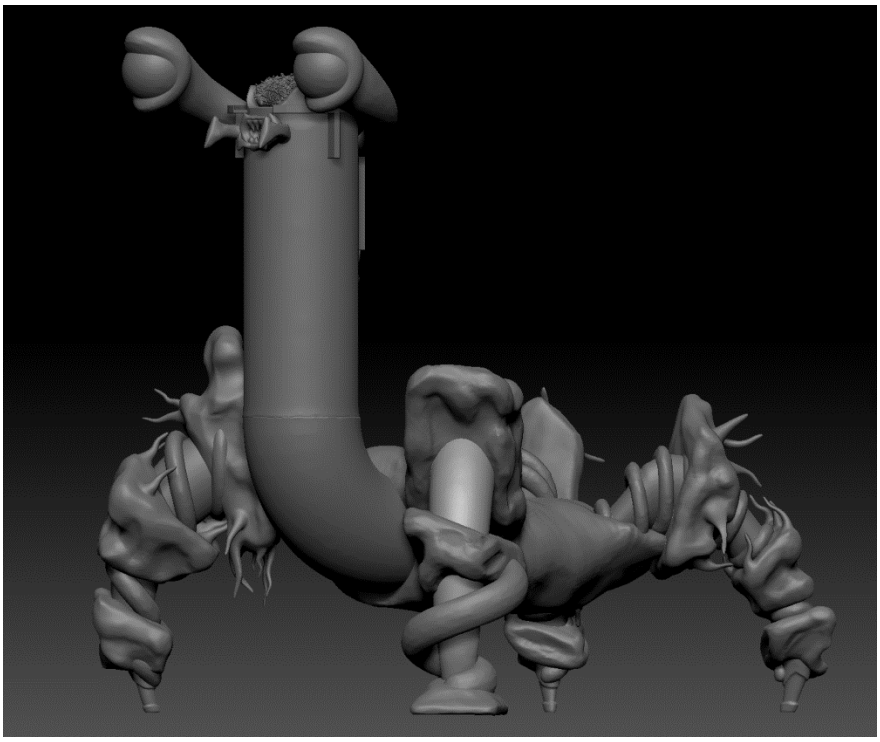

FIGURE 185. ESCÁRGOTREE, WITH ITS DIRT CHUNKS CREATED

Next up were the roots and vines. Each of them also had tendrils stemming all over the body. However, as these were defined, an idea sparked: similarly to the bark's bending being inspired by bonsais, the vines across the sculpture used as inspiration the twisting forms of a beanstalk. As such, a model was repurposed to be part of such plants (the life fruit, which was remodelled to look like a heart-shaped bean inside an open pod), and another module was sculpted: a small bean pod created with three spheres combined, resembling the beans inside. This model was used for the nanomesh topologies like the vine wall and some parts of the monster.

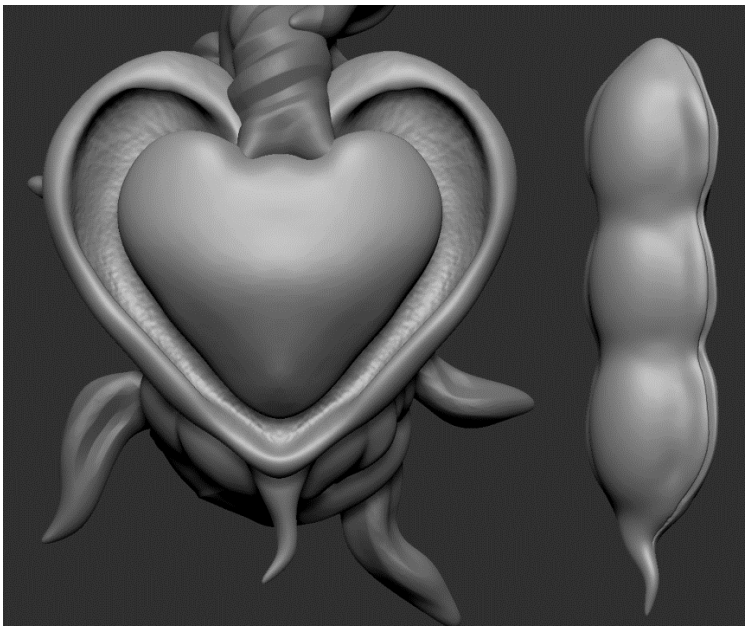

FIGURE 186. REPURPOSED LIFE BEAN AND THE BEAN POD ON THE SIDE

With this idea in mind, the vines' forms began twisting and combining each part, creating exoskeletons in the legs and reimagining some of the walls to be vines from a beanstalk intertwining like tubes in a system or directly giant pods with enormous beans inside. The eyes, in particular, had the most creative repurpose, as now they could become peas in a pod working as an eyeball inside its eyelid. One exception was the climbable leg, which was carefully sculpted to maintain the carefully constructed blockout while also having the characteristics of the other shapes. Some of the chunks even grew in size or were flattened to become better natural barriers than the negative affordance created using the spikes in the prototype in Unity. After all the vegetation and reshaping, the Dynamesh phase was completed, and it was time to add more detail to the beast.

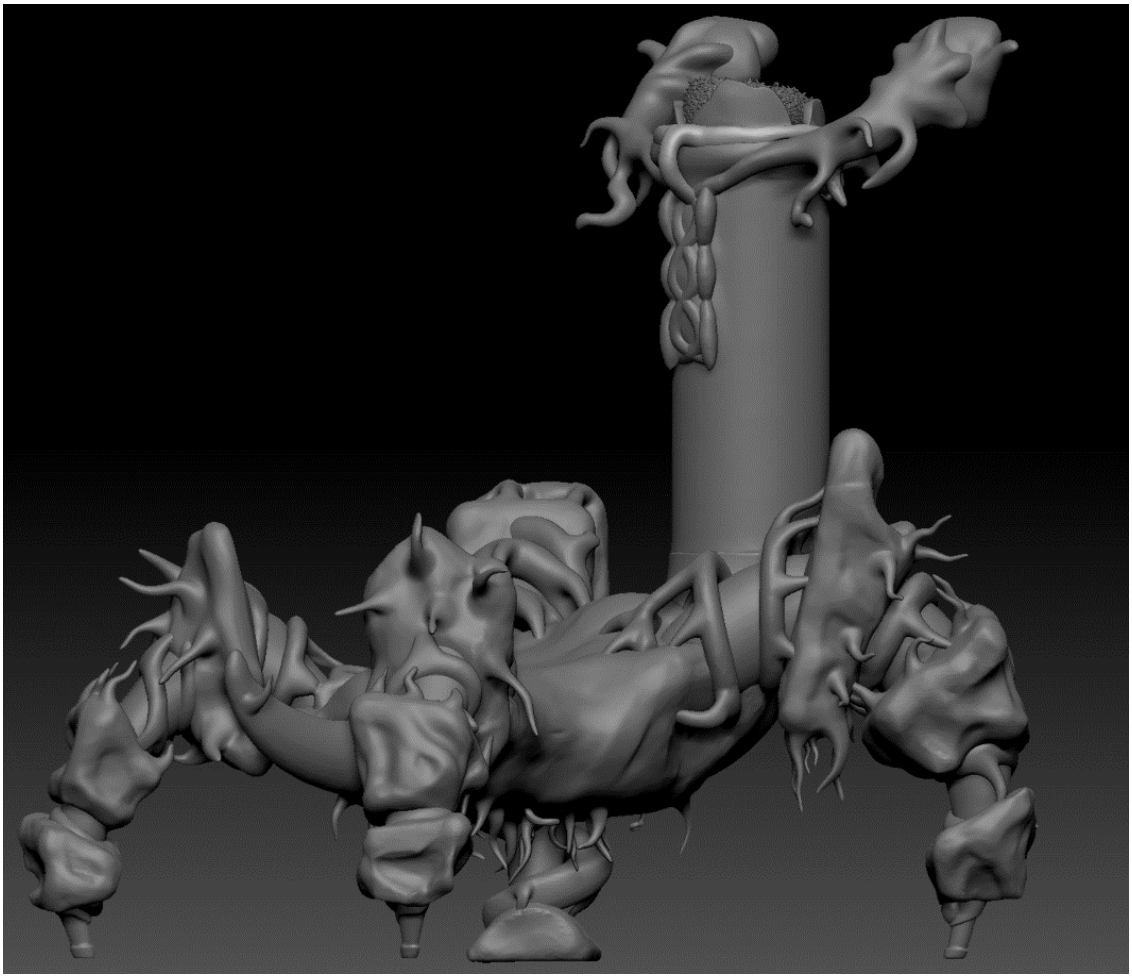

FIGURE 187. ESCÁRGOTREE'S BACK, WITH VINES AND ROOTS, MORE ELABORATED

The roots and the trunk use the same methodology as their modular counterparts. Using alpha masks, which are black-and-white images that determine what parts of the brush-selected areas are affected based on the whiteness level, the student created the wooden cavities across the surfaces. He then complimented it with some added ClayPolish treatment and features like ground with roots at the bottom of the central elements or rocks.

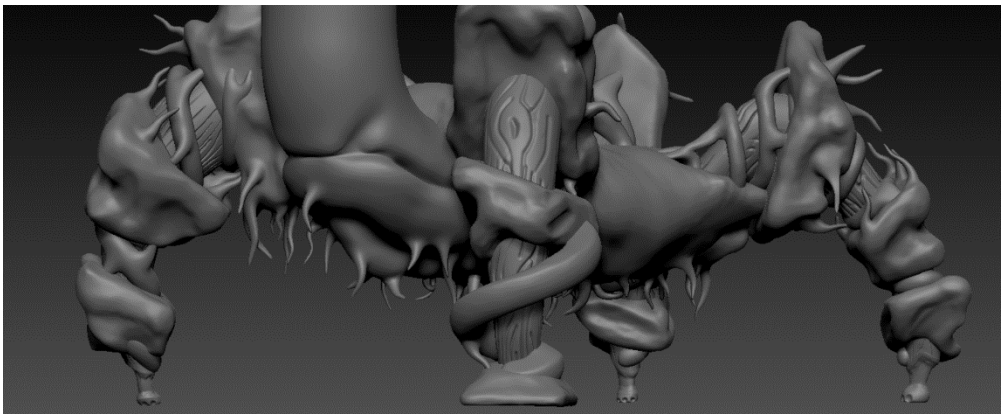

FIGURE 188. ESCÁRGOTREE'S ROOTS MORE ELABORATED

The trunk's cavity design had to be iterated to find the perfect pattern for the final design. At first, the initial concept was to create simple lines across the surface going upwards to the stump at the top. However, the author felt this lost the original essence of the concept art and moodboards' design; thus, he had to convey the idea of twisting wood on a relatively flat surface. The key to solving this issue involved recuperating the original asymptotic path created for the first iteration of the climbable area, hence making this zone's cavities follow this pattern, which gave the structure more interest than the first design.

The sculptor applied the same process to stylize and improve the current model topology for all its components, as seen in all the 2.5D section's modules. To each sub-tool, the author applied ZRemesher to fix the topology's loops and troublesome topology generated when using Dynamesh. To clean them up afterwards, he used ClayPolish again, which fixed most of the models' imperfections.

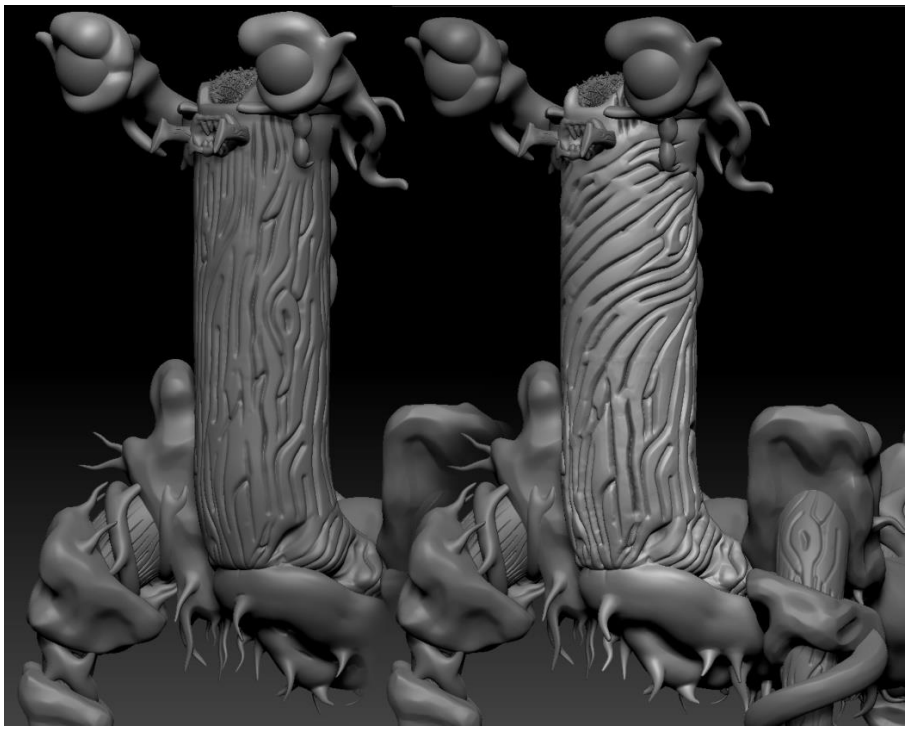

FIGURE 189. COMPARISON OF BOTH TRUNK DESIGNS, SIDE BY SIDE

With all of this set, more details were added in the form of small rocks and other natural marks. Looking into other references for beanstalks, the author deemed it necessary to create more superficial vine-like structures that could be implemented as add-ons across the various models. With all the high poly models created and exported, it was time to optimize them.

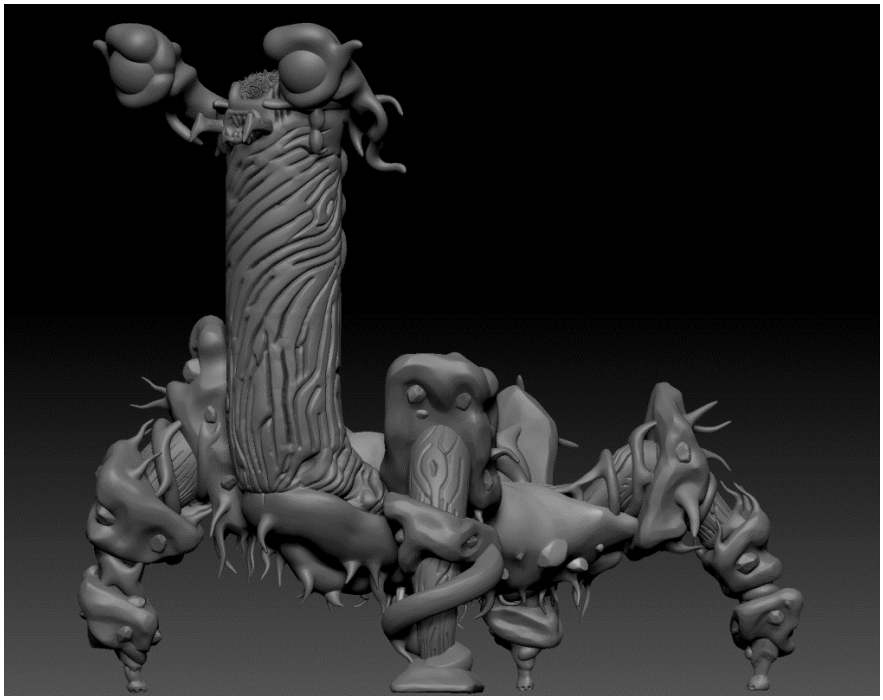

FIGURE 190. FINAL ESCÁRGOTREE SCULPTURE, FRONT

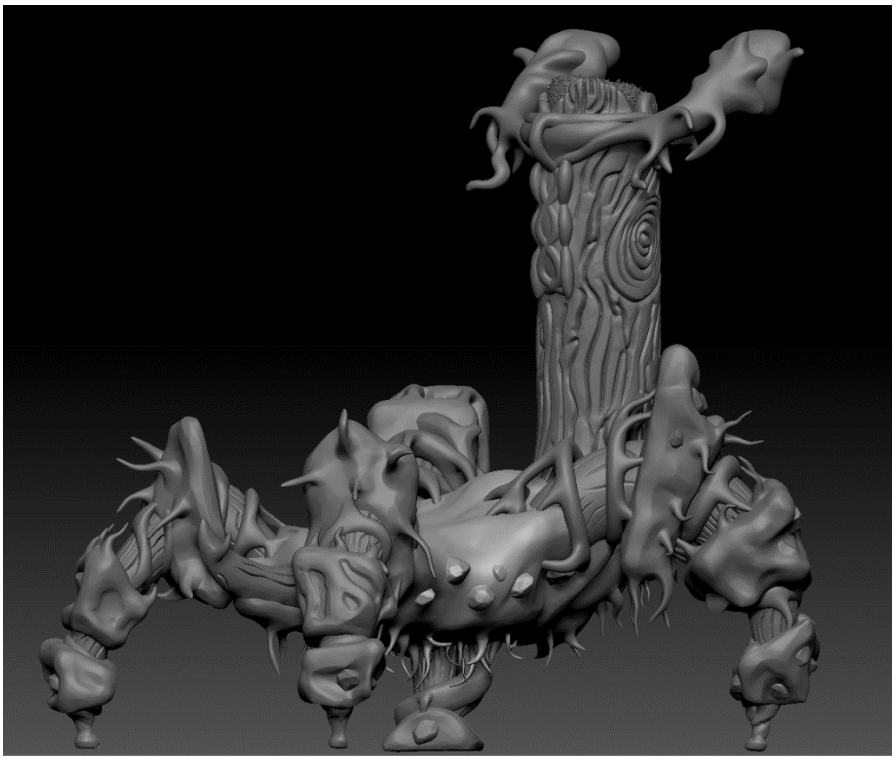

FIGURE 191. FINAL ESCÁRGOTREE SCULPTURE, BACK

5.2.6. Retopology & UV Mapping: Automating the process

Since most of the modules created did not have complex shapes nor had to be deformed for animation, the author opted to use procedural tools to optimize all these topologies. ZRemesher's capabilities allow artists to automatically retopologize models, reducing their polycount and optimizing them much better than most 3D software. However, these imperfect tools require accurate calibrations to avoid spiralling loops or other topology problems or, if no other options were available, optimizing by hand by removing unnecessary loops using ZBrush's ZModeler, a tool prepared for hard-surface or low-poly modelling.

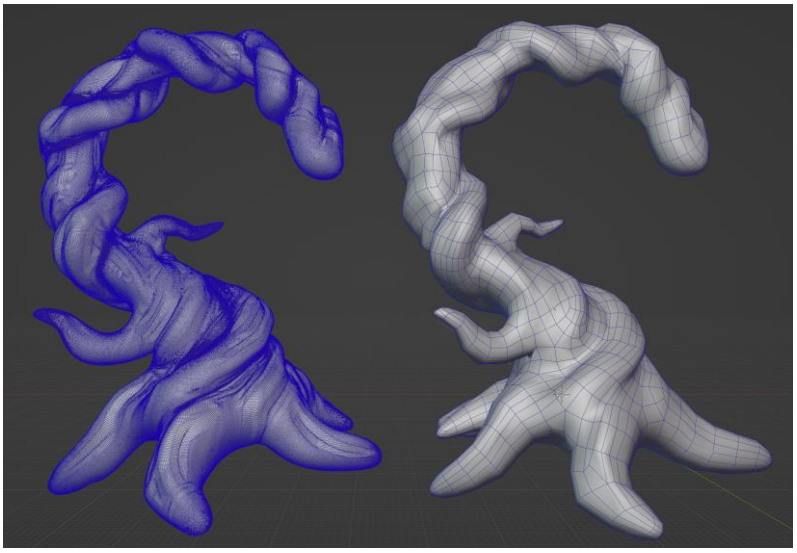

FIGURE 192. COMPARISON OF HIGH POLY MODEL WITH LOW POLY COUNTERPART, WITH WIREFRAME IN BLUE

Such problems appeared on the wooden platforms, for example, which caused many of these spirals to appear. Some other models, like the heart bean, only received the ZRemesher treatment in one of the instances, as the model used for the heart-shaped fruit is the same in both models. The student then imported the high and low poly versions into Blender for material distribution, renaming the models to fit with internal naming conventions and UV Unwrapping purposes.

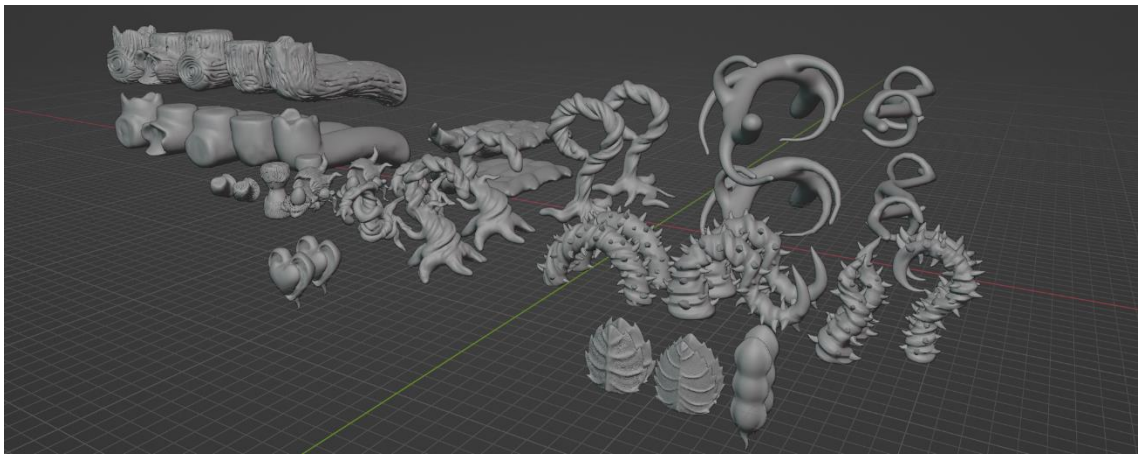

FIGURE 193. LINEUP OF ALL MODELS, BOTH HIGH POLY AND LOW POLY

This same process was also applied to the creature, whose sub-parts in total had a polygon density higher than all the modules combined since it is a bigger model than the rest. The modules have a polycount of 53k triangles, while the boss stands at 97k without other added assets. While this is a high value for most low-poly characters, it is still acceptable by today's standards, based on the hardware on which this will be played and the creature's dimensions. After all, God of War's Cronos had a polygon count of 18 million tris, 180 times the size of Escárgotree's. Nevertheless, if deemed necessary, there are multiple high-poly zones from which the user could remove loops.

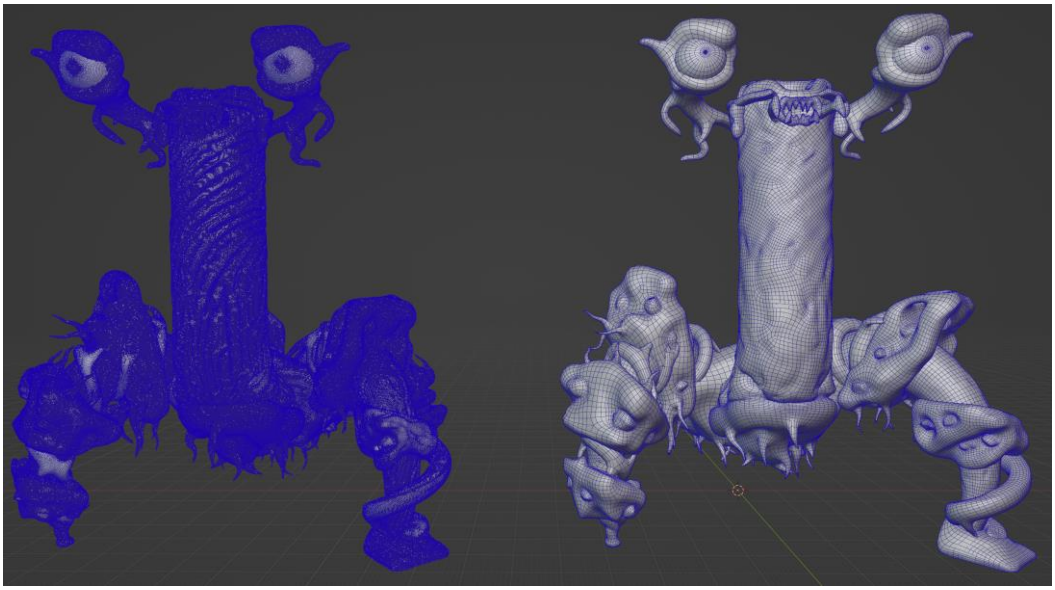

FIGURE 194. COMPARISON OF HIGH POLY ESCÁRGOTREE WITH LOW POLY COUNTERPART, WITH WIREFRAME IN BLUE

With all the models in place, the student divided the boss' assets into materials. In the end, they were separated into eight materials, these being one for the wood platforms and their add-ons, one for the cannon, one for the heart fruits, one for the key and door, two for the brambles (main plant and spikes), one for plant add-ons for the other models (like leaves and the bean pods) and finally one for the dummy at the beginning of the stage. Similarly, Escárgotree was divided into nine materials: one for each leg, one for the body, one for the main trunk, one for the eyes, another for its tubes and one for the top elements like mouth and nest. This allotment distributed the UV maps equitably based on polygon density and importance. Speaking of which, the UV Mapping process involved two methods: one automatic and another manual.

The primary modus operandi the author followed for most UVs was using automatic procedures, which were quick to use and usually generated acceptable results. Blender's material and UV Unwrapping processes are comparable to standards in the industry if paired with external plugins. For this method, three tools were employed. The first is SmartUV, whose primary purpose is automatically creating seams based on the angle difference between polygons. While it can create unfavourable seams depending on the main loops a 3D model has, it created acceptable results for simpler models, like the bramble's spikes. If SmartUv's results had multiple undesired islands, the artist manually created the UV's seams by marking and unwrapping.

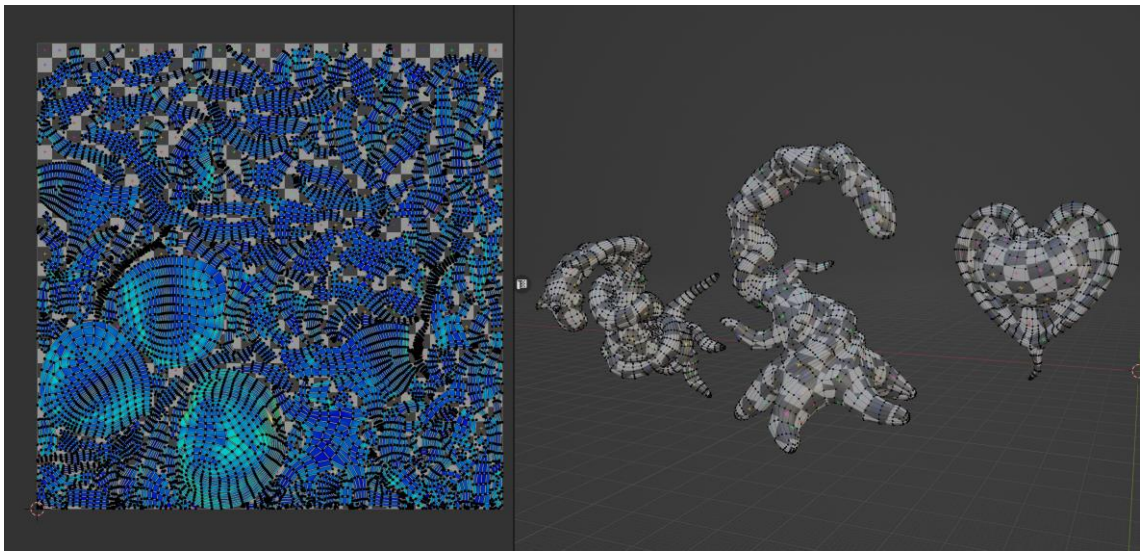

FIGURE 195. EXAMPLE OF MODELS AFTER BEING APPLIED AUTOMATIC UV PROCESSES

After unwrapping, further editions were done using ZenUV, an external plugin that adds new functionalities absent from the original product. The most important of these is UV relaxing, which helps reduce stretching on UVs to improve the texel density. Finally, to distribute the space and optimize it as much as possible, the student used UVPackMaster3, which automatically positions UVs to occupy the most space available.

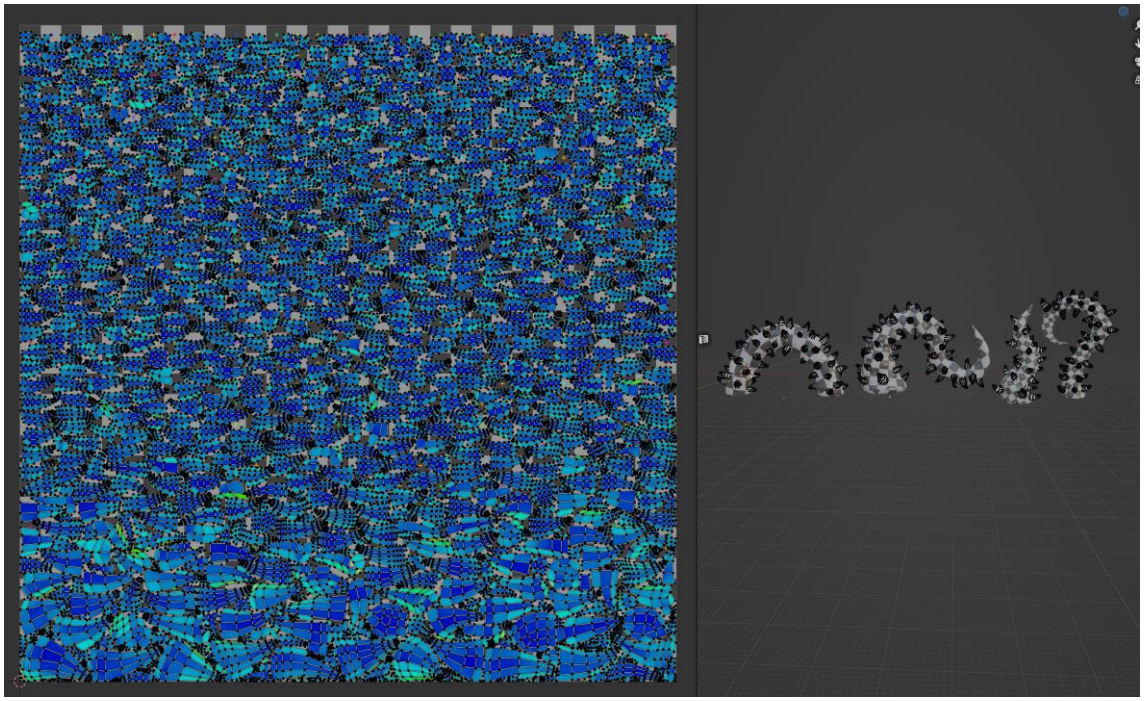

FIGURE 196. UV MAP FOR THE BRAMBLES' SPIKES

These processes were applied to each batch of models sharing the same material. While some models had to be repeated multiple times to avoid major texturing issues like seam cuts and texture whitening, all models were prepared to be textured in their majority from the get-go.

5.2.7. Baking & Texturing: The light from within

To pass all the details from the high poly to their low poly counterparts, the writer used Substance 3D Painter's automatic baking tools, which generate information regarding how light reflects on the model's textures based on the detail included in the high poly's surfaces. This process requires some calibrations and previous adjustments to work correctly, such as superposing the high poly models with their low poly pairs, including suffixes at the end distinguishing and pairing models to avoid baking artefacts, or colour ID maps, which allow distinguishing parts of the same model in different substances self-contained in the same material. These preparations saved a significant amount of time in future parts of the texturing process, as they allowed the artist to use many other procedural tools Substance 3D Painter has for non-painters like himself.

FIGURE 197. EXAMPLE OF WOOD TRUNKS WITH BAKED NORMAL MAPS

After baking, one thing the student had to develop ultimately before starting to texture was the colour palette of the monster and its related assets. The process of colour selection also requires an exploratory phase to seek inspiration and generate an ingenious idea. As such, he created a moodboard with multiple colours coming from real trees, using Adobe Colour as a basis to create exciting colour palettes made out of multiple references ranging from simple oak trees to birch and Sakura trees.

FIGURE 198. MOODBOARD 8: COLOUR PALETTE EXPLORATION

While beanstalks do not initially come from trees but rather are vine plants themselves, the fact that the original concept had to be an idea only suggested that the artist had complete liberty when choosing the type of Escárgotree's trunk. Thus, several quick concepts were applied to the optimized model using Substance Painter as an alternative to quick thumbnailing to see what colours fit better with the design in mind. The author generated seven images using different combinations of bark like oak, birch and Sakura, plant life with colourations inspired by the previously mentioned trees, and crevices colours, experimenting with emissive materials to generate interest in the duller concepts (like the oak tree).

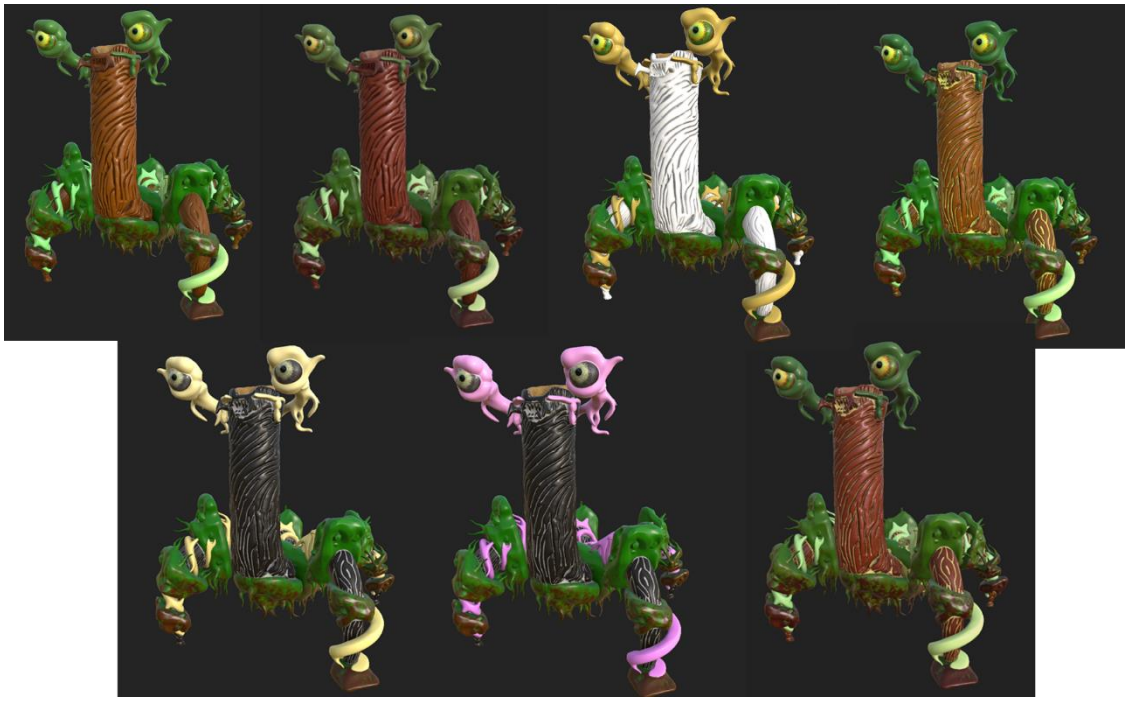

FIGURE 199. COLOUR PALETTE ITERATIONS APPLIED TO ESCÁRGOTREE'S PRIMARY MODEL

Eventually, the author settled for sticking to the classical look of an oak tree, but with certain variations in colour ranging from red and yellow for the bark and lime and dark green for other vegetation. While it was undoubtedly a classic look for a tree, it needed to keep some chromatic relationship with the environment surrounding the creature, so it was ultimately the best outcome. However, since the colours felt too familiar for his taste, the artist included emissive colours to add energy and life coursing through the "veins" of the beast (with the possibility of adding shaders in Unity to improve the light traversal through the wood).

These colour palettes can be seen in the primary wood platforms, where moss has grown on top of some trunks, in the cannon, whose egg changes depending on the type of bird it releases, and in the dummy, with a more transparent section not covered by bark. Most of the colouring has been created using both custom grunge maps included in the program, smart masks (which are preset masks that allow for the creation of procedural textures) and generators, which take information from the previously baked maps and use it to colour certain parts of the model automatically. In these cases, the normal maps and curvature generators autonomously painted all of the wood's crevices.

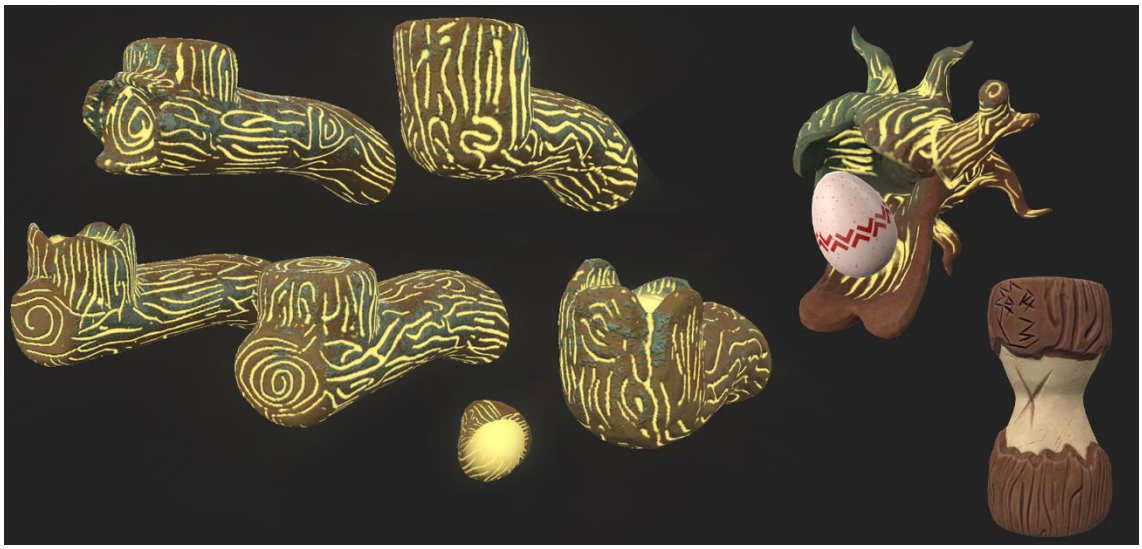

FIGURE 200. TEXTURIZED WOODEN MODULES OF THE BOSS FIGHT

Most leaves and vines' colour palettes are similar, and their lighting works mainly the same, containing lime and green emissive lights between the cavities; however, not all surfaces contain the same crevices. Such is the case of the brambles and the heart vines, whose lighting patterns appear on the outside surfaces rather than the interior. The lighting mostly worked like the trunks for the leaves and the key-door combo; however, in the latter case, they were colour-coded to pair them together, so the ending result was grey to change the colour later in Unity.

FIGURE 201. TEXTURIZED VEGETATION MODULES OF THE BOSS FIGHT

Finally, for Escárgotree, the artist used all the practice earned from the other assets to texturize all parts. However, the main beast's body had some throughout the model, having to redo many UV maps for different parts of the model due to overlapping faces, as the automatized mapping created unwanted clutters. This process slowed down the initial texturing significantly but was compensated in the future with better texturing. Most textures used for the monster were extracted from preexisting images to maintain chromatic consistency with the game environment and the previously designed materials.

On the one hand, the dirt chunks were painted using textures from the game environment created in Unity. Its surfaces were painted with grass and the inner parts with dirt, getting darker as they approached the ground. The only parts original to these chunks are the rocks and roots, created with preexisting materials in Substance Painter. On the other hand, Escárgotree's wood and vines were created using the same materials as the assets, with some modifications to adapt to the model and variations to keep the model attractive to the eye. The original parts of the creature were the eyes, created using another Substance Painter material, and the nest, whose colours resemble hay to avoid getting obscured by the surrounding environment and differentiate between the 2nd stage and the bird boss' stage.

FIGURE 202. MAIN MODEL'S FINAL TEXTURES

The student applied some optimizations to export the resulting images from the textures, as there were many images per material. Substance 3D Painter allows for the creation of different methods of image combinations that contain the necessary information to replicate a model's textures, as seen in the program. To optimize the amount of memory used for this battle, the author compressed all the maps into three or four types of images: one for the albedo, one for the normal map, one containing roughness, metallic and height information in the red, blue and green channels respectively; and finally, an emissive map for the models that require it. In some cases, as multiple colour variations for the same model were created, the number of images increased exponentially, making some models have as many as fifteen images for the same model. After texturing, it was time to build the beast's parts.

FIGURE 203. ALL TEXTURE MAPS FOR THE CANNON

5.2.8. Mounting: Building the beast

The mounting phase refers to the recreation of the blockout monster created for the game's first iteration with the newly created assets. Many of the assets created are single modules that will be rearranged in the central structure, so the author had to develop these variations in Blender. Firstly, much like the ground textures, some assets like foliage, trees and buildings were exported from the main scene to keep up with the theme of maintaining artistic consistency. The buildings were positioned just like the original blockout, and some trees were strategically placed to avoid unwanted climbing from the player and create a small ecosystem on top of the beast.

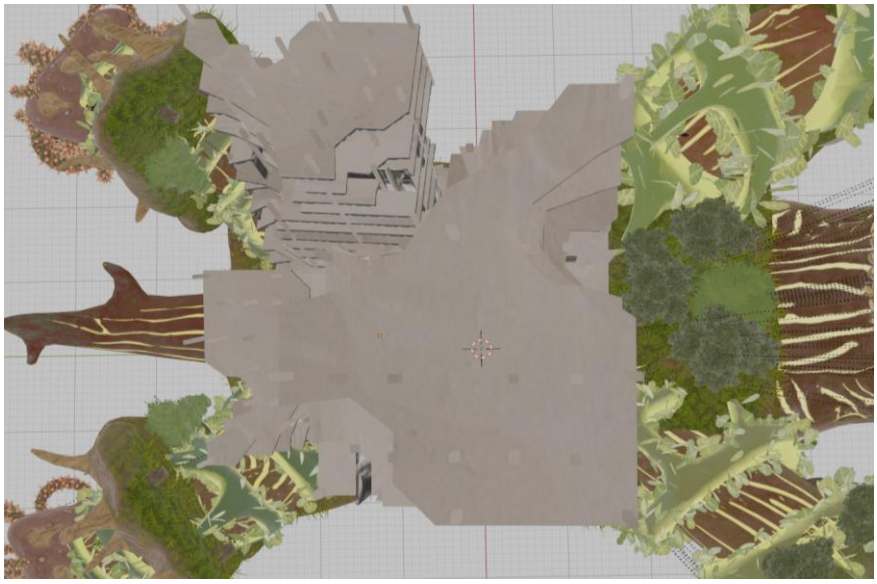

FIGURE 204. ESCÁRGOTREE'S BACKSIDE, AS SEEN FROM THE TOP

For the vegetation, the author used a feature in Blender called Geometry Nodes (or geonodes for short), which are custom-made modifiers that allow one to change the geometry of a model based on the arrangement of nodes programmed by a user. With these, the author instanced the foliage across the green parts of the model. Most followed the same structure as the image seen below but with minor changes such as masking nodes based on position or direction, density modifiers and other parameters to create the perfect foliage for each situation.

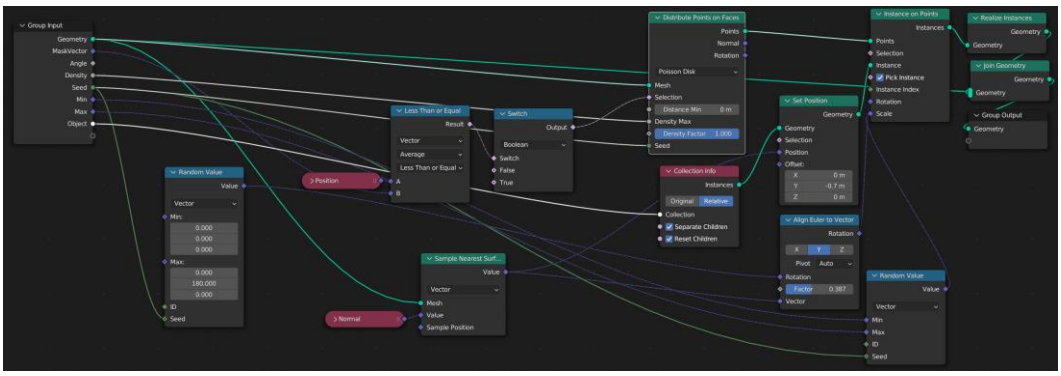

FIGURE 205. GEOMETRY NODES FOR LEAF INSTANCING

A common practice in the industry is for greenery like leaves, bushes, and branches to be kept as planes with an elaborated 2D Texture. As instancing some models would make the model too heavy, the author baked both the front and back sides of the leaves and bean pod models in 2D planes by rendering all the maps in different images using an orthographic camera. Such planes would then be instanced over several models using adapted versions of the previously created modifier.

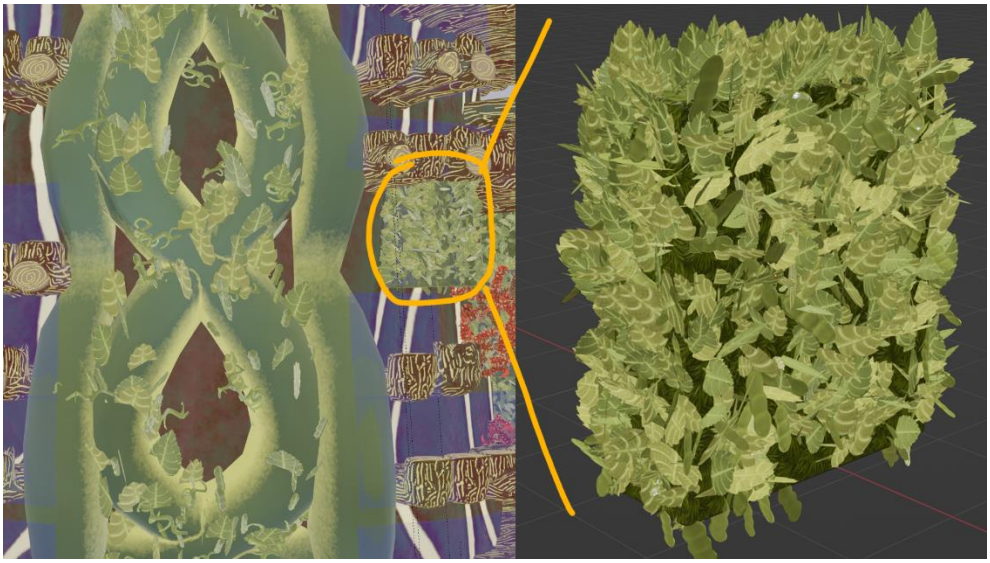

FIGURE 206. EXAMPLES OF LEAF INSTANCING IN VARIOUS MODELS

The platforms and brambles were tricky; they had to conserve the same collision boxes as the initial platforms, and they were by far the most significant contributors to the final poly count of the model. As such, both models (plus some additional ones from the main structure) were retopologized using a Boolean modifier to remove topology overlapping with the model and the Decimate modifier from Blender, which reduces a topology to fit a specific percentile while not breaking the UVs. This method, while unorthodox and generally not recommended due to the resulting meshes not having the best loop quality, was not as problematic in the models applied, as these models would not suffer deformations due to animations. This way, the heaviest models could have a reduced topology while still looking similar to the original model.

FIGURE 207. BRAMBLES INSTANCED OVER UNCLIMBABLE LEGS

Similarly to the foliage, both brambles and platforms used geometry nodes to instance models over surfaces. In the case of the brambles, it was as simple as creating a plane applying the same instancing geonodes but with the brambles models. However, the process was different for the wooden platforms, as they had to follow a path along a circle wrapping around the previously created platforms (which would serve as collider helpers) and also pick instances from a collection of models. As such, a new set of geometry nodes was created that did that same function, which would then be instanced for all thirteen platforms across the trunk.

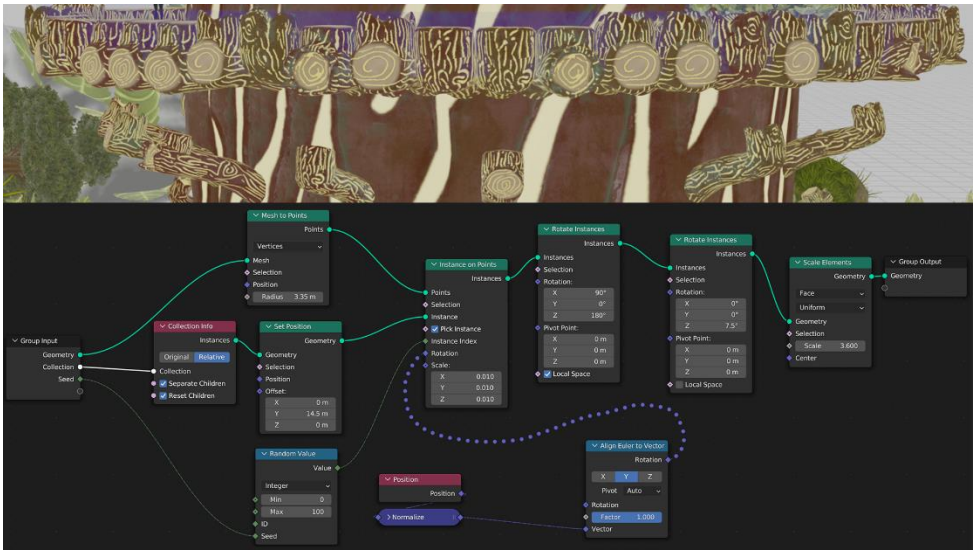

FIGURE 208. GEONODES FOR WOOD INSTANCING APPLIED TO THE PLATFORMS

After some readjustments, accommodations and optimizations, the final model for this boss battle was finally done, with a topology of more than one million polygons for the entirety of the model. A beast of a model that was getting closer to the numbers seen in other colossal bosses like Cronos. It was time to animate the beast and implement it in Unity for testing.

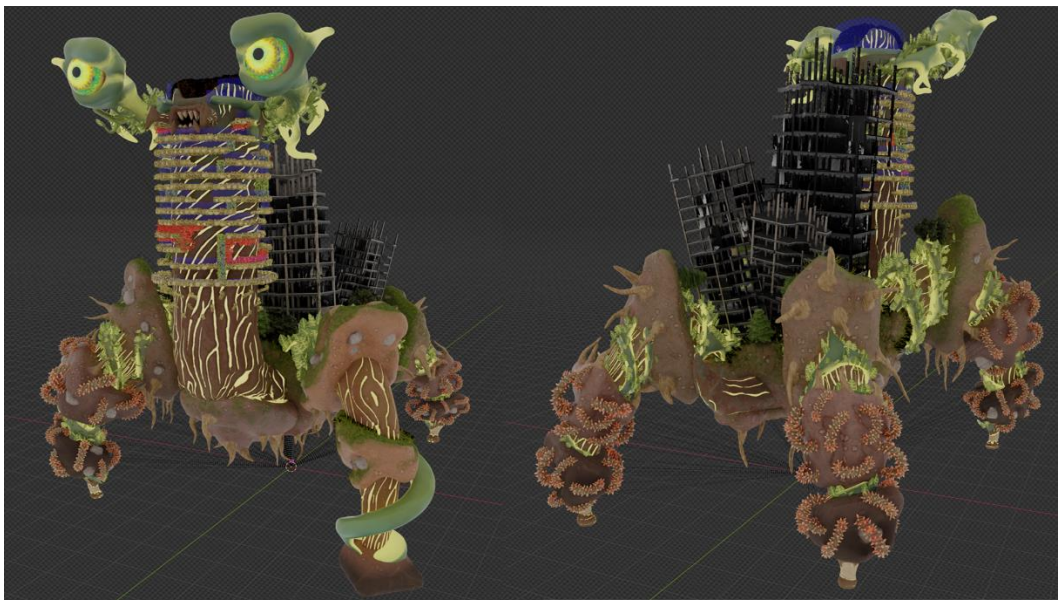

FIGURE 209. FINAL RENDER OF THE COMPLETE ESCÁRGOTREE MODEL

5.2.9. Rigging II & Animations: Bringing the beast to life

The rigging process followed for the final model is similar to the one used in 5.2.3, as it utilizes the same plugins and software to create the animations, updated from its previous iteration. The rig was remade from the ground up to avoid skinning issues and other shenanigans, only binding the deformable parts of the model. This decision resulted in two models being exported, one containing the complete unrigged model and the other containing the deformable parts with the bones. The author deemed it simpler to recreate the same rig based on the original model because of the number of added meshes and because the general shape of the beast received readjustments between production stages that made the original rig unusable.

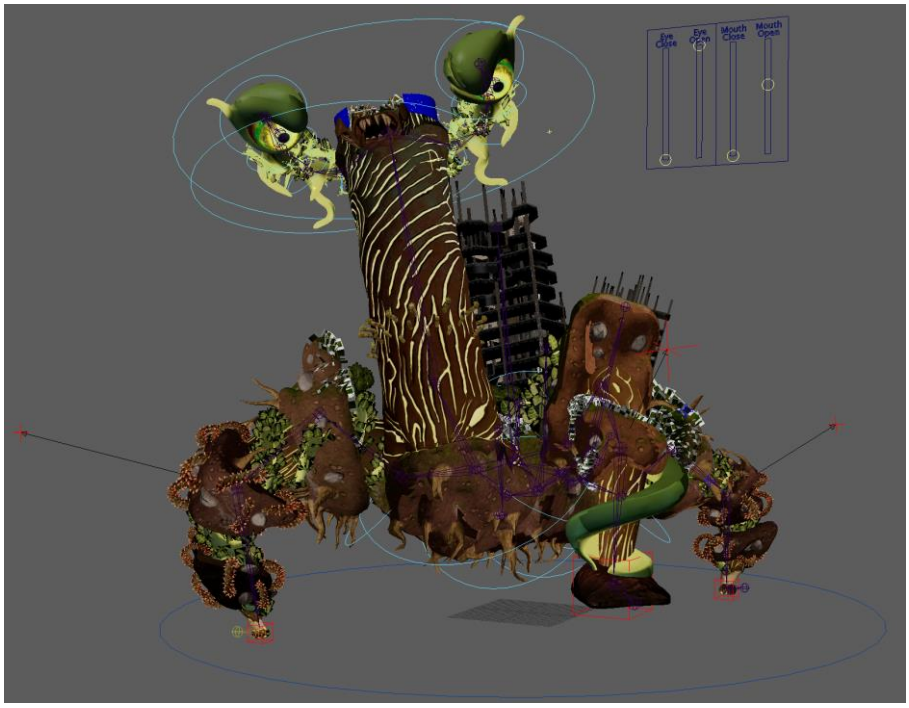

FIGURE 210. ESCÁRGOTREE RIGGED WITH CONTROLLERS AND POSED FOR STOMPING

For the end of this production phase, the principal animations created were a stomping animation for the attack and the final position of the monster being down. The difference between this iteration of the animation and the previous one is the number of secondary movements and deformations created to convey the weight and power of the creature's stomps, which was not reflected in the blockout phase.

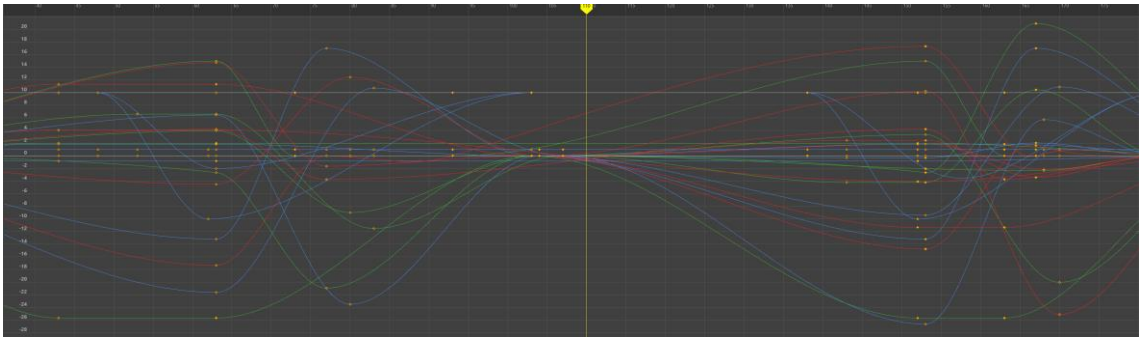

FIGURE 211. ANIMATION CURVES FOR THE STOMPING ANIMATION

The author developed other controllers for more specific movements like closing and opening both eyes and mouth using blend shapes, an alternate version of the model with a deformed topology that can be used as an animation key position from which the animator can interpolate between the base form and the deformed result. For example, this method can reduce the number of bones and skinning procedures required for precise facial expressions like those mentioned above.

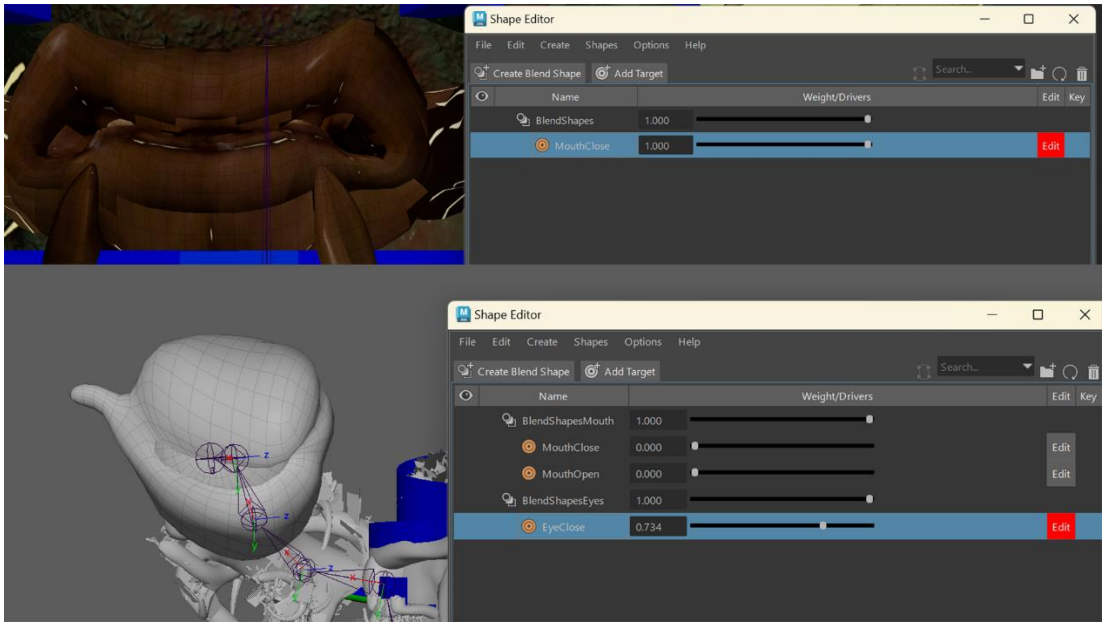

FIGURE 212. BLEND SHAPES IN ACTION, WITH THE WINDOWS INVOLVED IN THE PROCESS

The student also rigged and animated other modules for the placeholders. He created a simple animation for the leaf door from a chain of bones that rolls the leaf onto itself, similar to rolling up a rug, and another animation for the minion's egg rolling down the cannon, reading up in the cannon. Both these animations used the same processes as the boss: marking keyframes and interpolating between them using Bezier curves created in the graph editor. This way, the boss fights, and the whole experience will feel livelier and more dynamic.

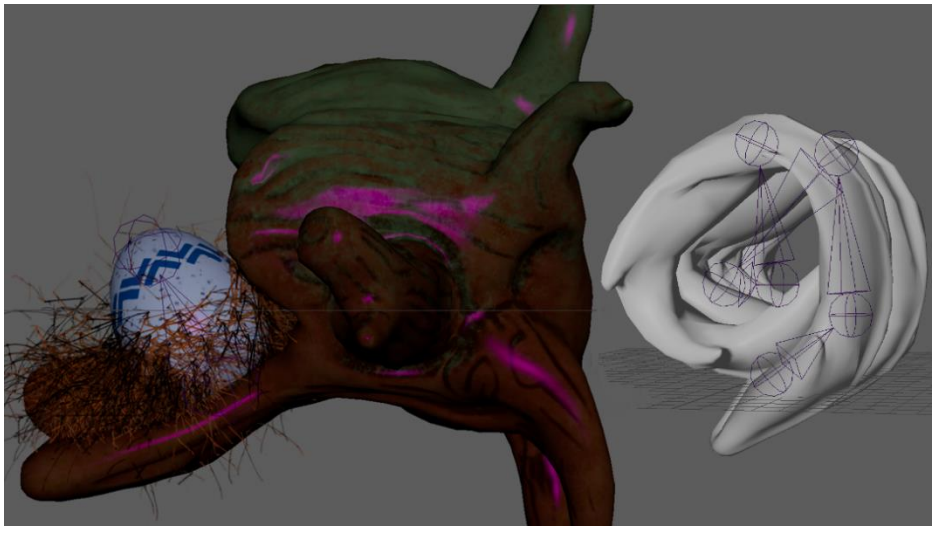

FIGURE 213. RIGGED MODULES WITH ANIMATIONS IN ACTION

5.2.10. Implementing & Improving: Blooming like a butterfly

With all functional animations complete, it was possible to substitute the old model with the new one to expand the game's aesthetics. The first step was to import it into Unity. As previously mentioned in 5.2.4, the rendering pipeline used in this project is the URP. Based on the texture pipeline created, the type of textures does not adhere to the one used in this rendering pipeline, as it requires six maps instead of four. The base map, normal map and emissive map were easy to implement, but for the roughness-metallic-height map, the student had to use external tools to create a new shader for each part's material. This shader is a recreation of URP's Lit shader. However, it uses the Shader Graph features instead of code with additional modifications that separate the RMH map into each RGB channel to assign it to its proper attribute in the material. Since Unity uses smoothness instead of roughness for its materials and roughness is the direct inverse of smoothness, the author also had to invert the red channel, resulting in an optimized shader adapted for Physically Based Rendering.

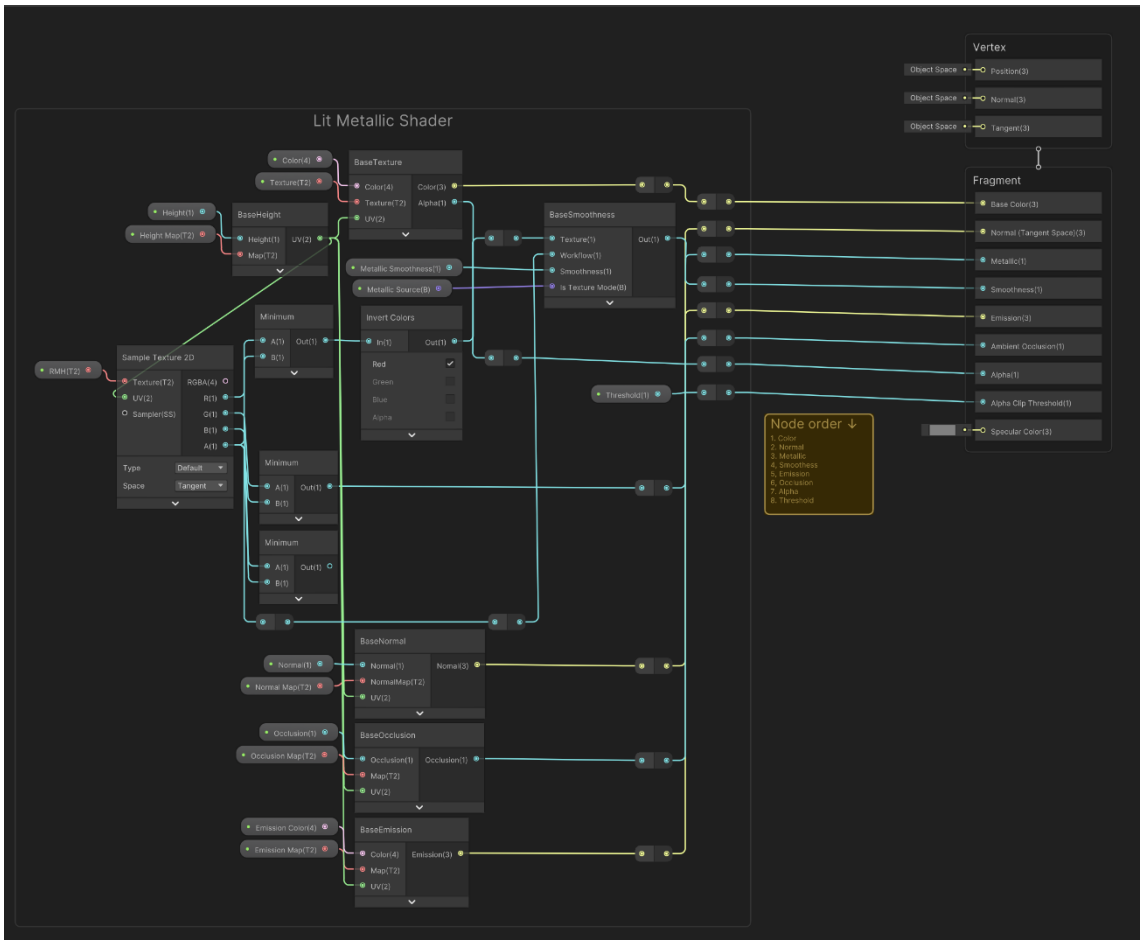

FIGURE 214. NODE GRAPH URP SHADER FOR MOST MATERIALS USED IN ESCÁRGOTREE

From there, it was as simple as recreating the boss' interactable entities, behaviours and areas and reassigning the properties to the new rigged models, similar to the process seen in 5.2.9. Some parts were as simple as dragging the item from one prefab onto the other, while others required methodical tweaking and adjustments, like the nest at the top of the arena, which invaded the top platform, thus making it unusable.

FIGURE 215. BACK OF CARPINTROYER'S NEST, NOW BEING INVADED BY THE BIRD'S HOME

The reimportation of models got tricky at times, as the new models had to have their scripts rewritten to accommodate the new features these models brought to the table and their transforms having become disparate from their blockout counterparts. These problems became apparent when substituting collision meshes and boxes from the main structure, as the new structures, having become more organic looking, did not always fit the cubical, symmetrical sides of the blockout. Nevertheless, despite these difficulties, the main structure and elements became functional, like the first playtesting build. However, it did not mean it was free from game bugs and design flaws, as there were still many collision errors and mistakes to fix with the top boss.

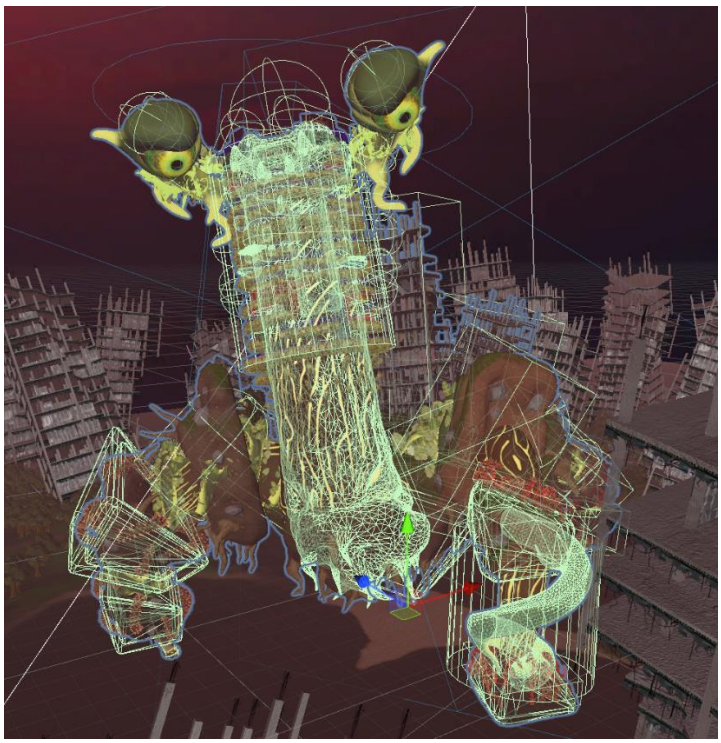

FIGURE 216. ESCÁRGOTREE'S COLLISION BOXES MOVING WITH THE STOMPING ANIMATION

One problem the author found early on when testing out was that the 2D section got too dark at times, making it difficult to distinguish the player and the other structures. The author implemented three different solutions: firstly, adding a light that would illuminate the player when traversing the level, which significantly helped it stand out from the background; secondly, adding contrasting emissive materials to the models, some of which initially did not have any, colour-coded so that the player can know which does what; and finally changing the player's clothes to be an electric emissive blue, contrasting with the warm colours of the main trunk.

FIGURE 217. EXAMPLE OF COLOUR-CODED LEVEL ELEMENTS OF THE GAME

Another issue appeared with the nest: the author did not consider the camera positioning for the boss fight when initially designing it. Usually, these types of cylindrical boss fights have their cameras positioned so that they are between the player and the wall behind them; however, this was not the case for this boss fight, which compromised it, as zooming in was not a viable option. In the end, the student created a modified shader with a culling mask that made the model invisible in the middle of the screen by subtracting a circle or texture from the final rendering image, thus making the fight visible again.

FIGURE 218. THE ADDED SHADER NODES FOR THE CULLING MASK

Finally, a bug dragged from the previous build that was becoming more apparent with the new model was the character controller's collisions with the tree trunk, as they kept getting stuck in unseemingly strange places. After investigating, it became clear that the extensive use of mesh colliders was taking a toll on the character controller's movement calculations. As such, the simplest solution was simplifying the collision box of the trunk with primitive meshes like cylinders. Not only did it improve the overall experience, but it also increased performance and fixed other collision bugs regarding the minions' movements.

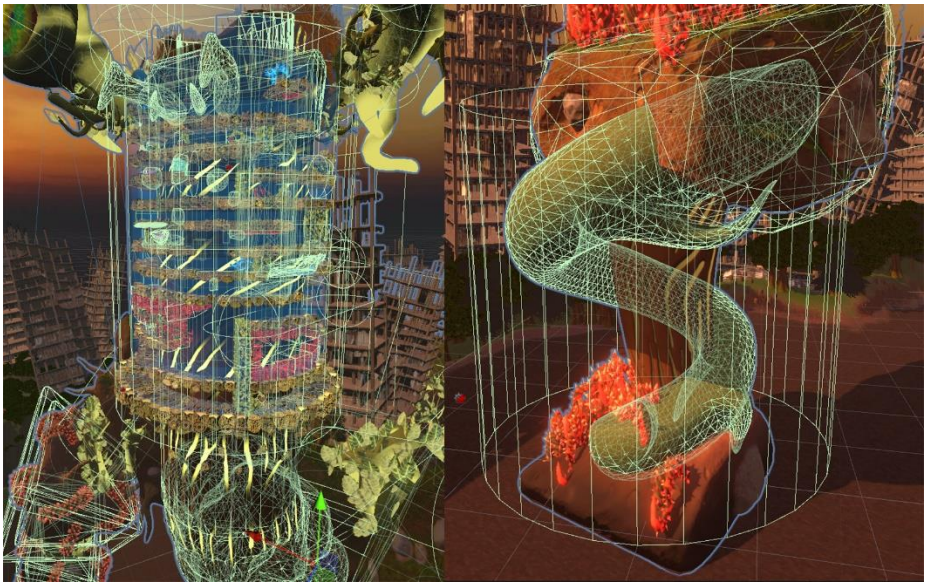

FIGURE 219. UPDATED COLLISION BOXES OF THE TRUNK AND CLIMBABLE LEG

With these many issues fixed, the author created some visual effects to improve the look and feel of the game's visuals. These effects were mainly created using Unity's shader graph and particle systems, made by the author, downloaded from the internet or created from tutorials. The ones developed during this phase created new and improved existing visual feedback for the user while providing aesthetically pleasing effects to the elements. The simplest was adding decals to the legs that activate whenever they broke from the player's sword swings. The peculiarity of these is that, by using Light layers, the decals only affect the mesh corresponding to the leg.

FIGURE 220. CRACK DECALS FROM DIFFERENT POINTS OF VIEW

For the minion shooters, a suggestion made during the first playtesting sessions was to make a preemptive charging animation to tell when they were ready to shoot. While this was already achieved with the moving egg, the author decided to include a shader that illuminates the cannon based on a value from 0 to 1 that generates a mask that exposes the emissive colour, which would then be attached in code to the charging animation, slowly lighting up as the egg gets ready to explode. When they did, a particle burst effect was instantiated to represent the egg exploding into smithereens.

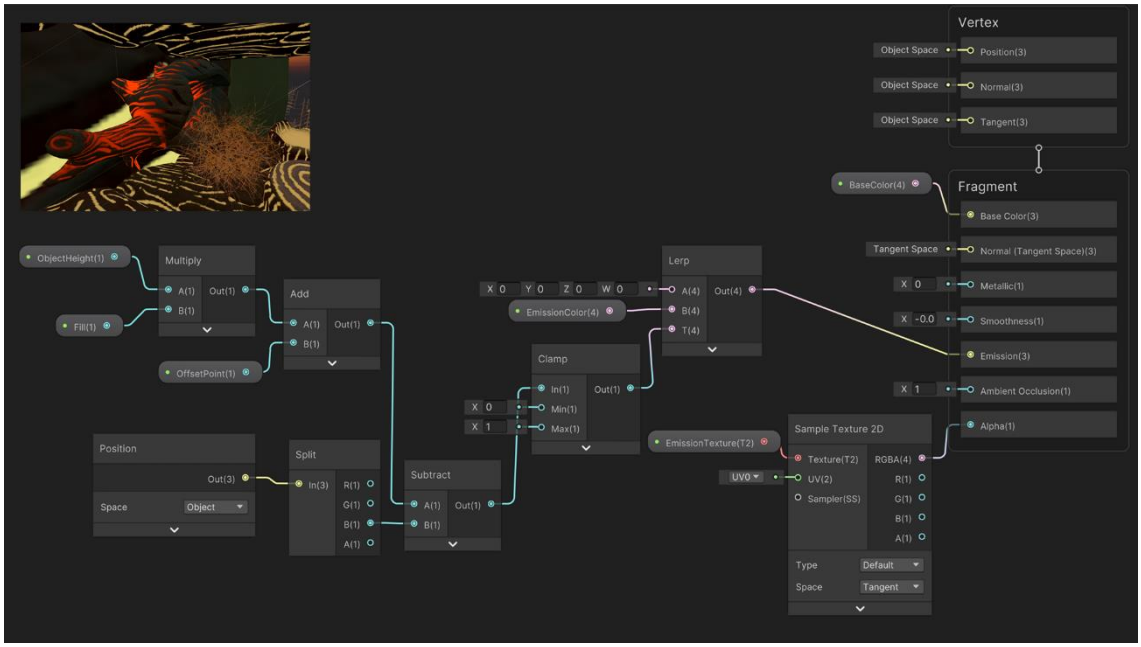

FIGURE 221. CANNON SHADER WITH VISUAL RESULT IN-GAME

For the surrounding trees and vegetation, the author created another shader that simulated the movement of rustling leaves by displacing the vertex position of the meshes based on a moving noise map that created a waving notion of the environment's vegetation by time.

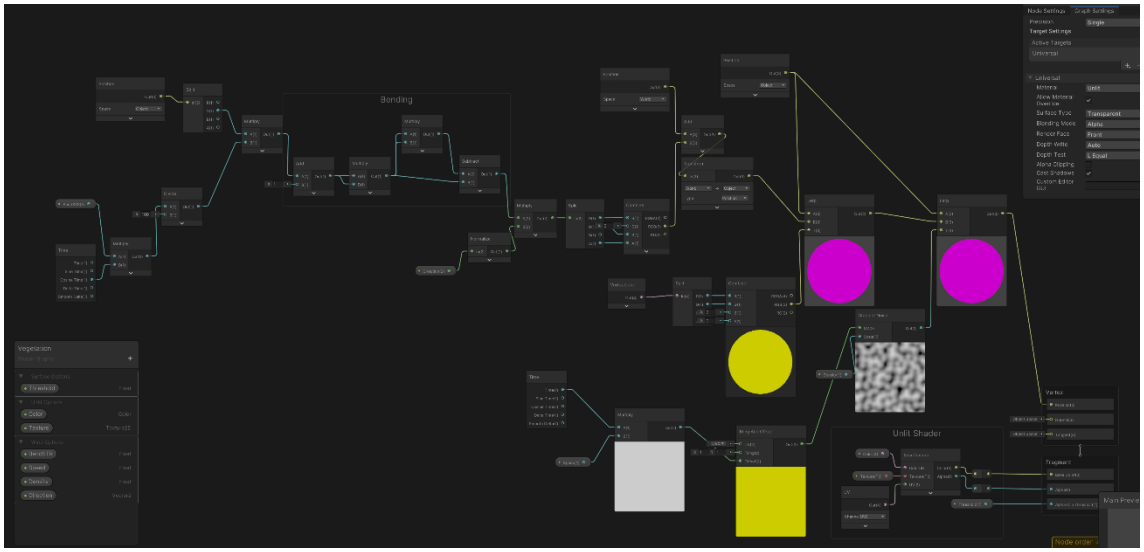

FIGURE 222. WIND SHADER FOR TREES

Finally, the player's attacks received slashing visuals with a moving texture on a rotating torus instanced in each of the player's attacks, based on the direction, speed and force that these attacks had to convey. These slashing effects also used a new visual effect graph tool created for complex particle systems, which includes the particle systems functionalities with the possibility to program them like a shader graph.

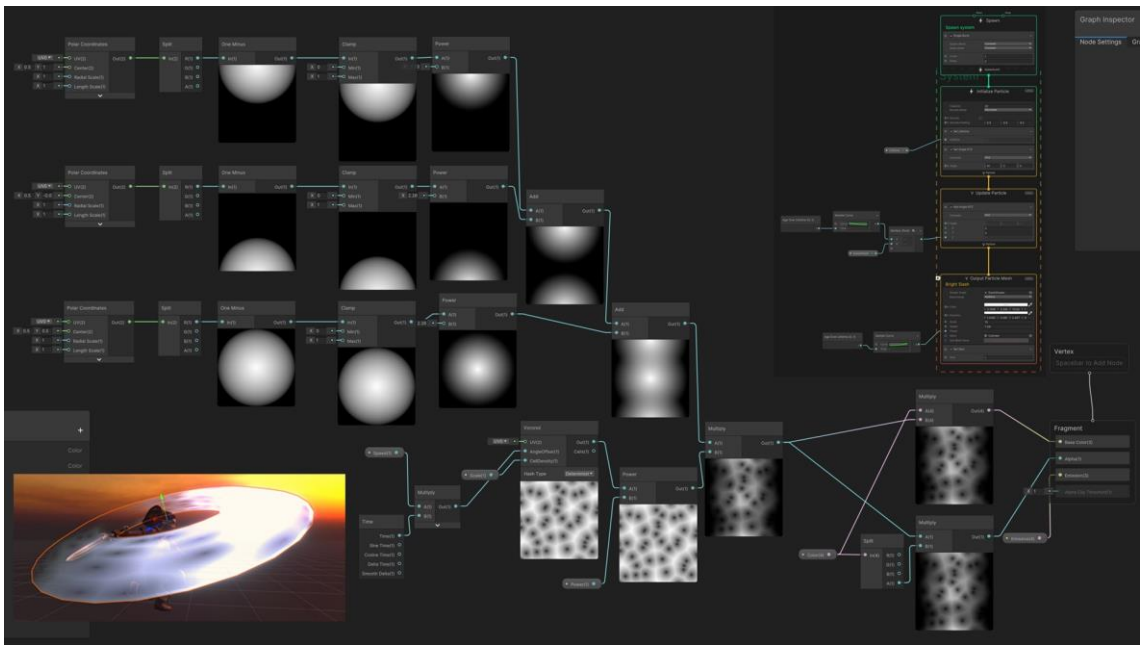

FIGURE 223. SLASHING SHADER AND VISUAL EFFECT GRAPHS WITH THE IN-GAME RESULT

At this time, this part of the process was cut short due to uncontrollable and unexpected schedule changes that prioritized producing other elements. As such, these elements will be continued in section 5.4 with other elements like sound and post-production. For now, to follow the set plan, the focus of this project shifted towards creating the sub-boss and its minions.

5.3. Production II: The Bird(s)

This section will be shorter than the previous ones, as many of the steps followed in this production are similar to those used for Escárgotree's primary model. As such, the artist will only describe the actions followed and additional notations unique to this model.

As explained before, the principal minions and mastermind behind all the chaos are Carpintroyer and its minions. These creatures were conceived during the discovery phase (5.2.1) as woodpecker birds with a design language resembling bullets and rockets, which already worked with the placeholder models to convey their functions to the player. Therefore, the main objective of this production is to make the models prettier and ready for texturing and animations.

5.3.1. Blockout: The second iteration of the design

At the beginning of the project, the models implemented were simple primitives combined to resemble a bird in the shape of a rocket, its head serving as the minion's shape. For this production, the author followed the

same design philosophy to create the designs and models for the main bird and its hatchlings: creating the big boss first and related assets and features, such as its feathers, so that the author can build up a smaller, unevolved version of itself with these pieces.

Since the model had to be animated, the blockout had to be mostly redone from scratch in 3Ds Max, as it had to be built in a T-Pose position as a resting pose for subsequent animations. The author only kept a few readjusted decorative pieces from the placeholder, but some parts, like the legs, had to be completely rebuilt to resemble a standing featherless bird.

However, this decision did not mean everything from the previous creation was lost. The original design created as a placeholder was a great place to start Carpintroyer's redesign. As such, the author got the main blockout design and improved it by adding more complexity to the original design, such as more feathers and hair on the face and legs. While birds typically do not have hair on their bodies, he wanted to anthropomorphise the design by giving it human features like hair and visible sclera. The modeller also wanted to increase the colour contrast on the face to reinforce the focus point of impact of the bird's shockwaves and the legs to resemble the explosiveness of a rocket when ignited while keeping the overall design less top-heavy.

For the feathers, the best course of action for these types of surfaces is to create quads covering the body and wings. Having learned from the leaf creation in 5.2.8, the artist created a model to serve as the basis for the feathers covering the bird's body. With these done, the author proceeded with sculpting the high-poly version of the creature.

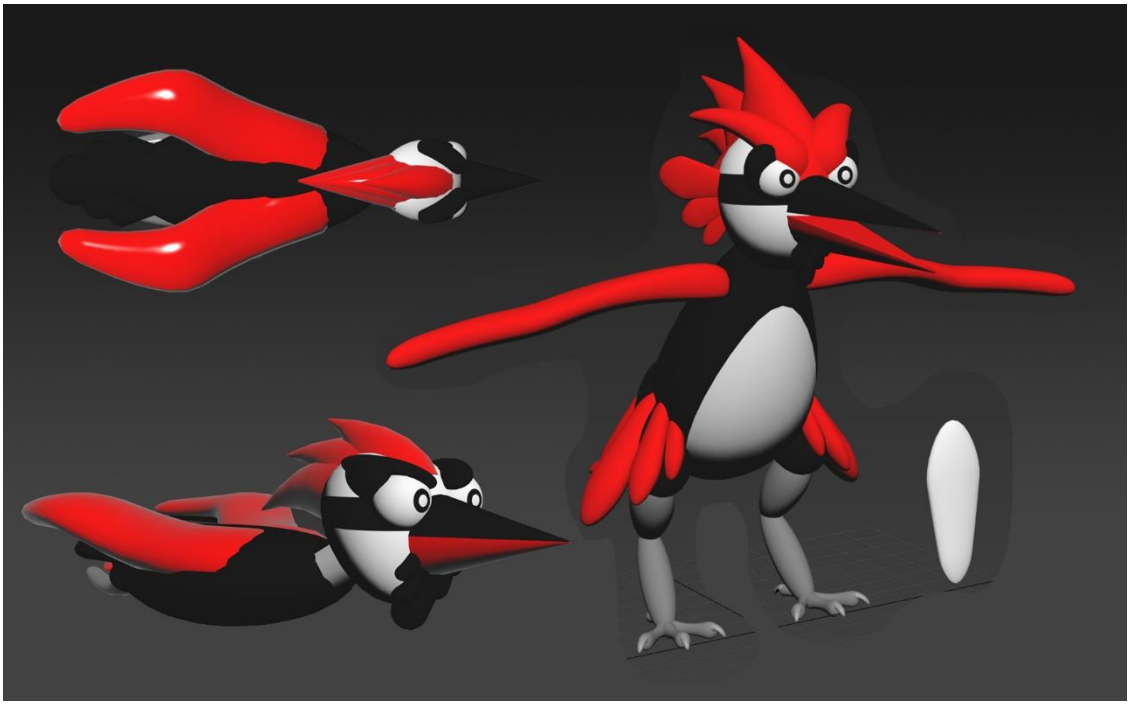

FIGURE 224. CARPINTROYER. ON THE LEFT: PREVIOUS DESIGN. ON THE RIGHT: NEW BLOCKOUT

5.3.2. Sculpting: Fur, feathers and coarseness

The sculpting process for this model can be summarized in two concepts: instancing and alpha masks. In this case, the main focus of sculpting is giving interest and life to the simplistic primitive shapes of the blockout. The fastest way to do it, especially with the hair, is with IMM Brushes in ZBrush — short for Insert Multiple Mesh — allowing for quick instancing of models from a preset library. As such, using the Fast N' Furry tool by Sergi Camprubí[15], the author could easily switch the pointy bits of hair with more slender, fiery shapes.

FIGURE 225. SET EXAMPLES OF FAST N' FURRY'S HAIR

While most of the models had little treatment and stylization required to improve on the original design — only having to use sculpting tools like ClayBuildup or slashing tools like the Dam Standard brush to sculpt all different body sections, the legs were the ones that required the most attention. Bird legs are characterized as having coarse yet intricate patterns on their surfaces, which the author focuses on getting right for this model. As such, the legs had several areas filled with recessed patterns created using alpha masks, something already seen in 5.2.5. Similarly, the rest of the body was covered in feathery patterns to help convey the bird's fluffiness. Other modifications in the body included pending the arms to indicate better the bird's joints and the lower part's feathers becoming more pointy, almost looking like autumn leaves, which combine well with the redness of the bird.

FIGURE 226. CARPINTROYER SCULPTURE, AS SEEN FROM THE FRONT AND ON THE SIDE

[15] You can find it here: https://www.artstation.com/marketplace/p/k61M/fast-n-furry-imm-brush-for-zbrush-2022

5.3.3. Retopology & UV Unwrapping: Another take on the process

At the beginning of the production, this model's retopology and UV unwrapping processes were considered to be done manually using Autodesk Maya's tools. However, due to time constraints and considering that most of the bird's body is pretty simple in shape language and that most of the bird's textures would not be extremely precise, the modeller opted to use automatic processes for retopology and UV Unwrapping. As such, the student used previously seen tools like ZBrush's ZRemesher and Blender's SmartUV, ZenUV and UVPackMaster tools to create the low-poly model of the bird.

Something to note that was different from Escárgotree's UV unwrapping was a technique to increase the area used by UV islands on the map. Since some parts of the bird would be symmetrical, the author created the UV maps for only half of the body. If a symmetry modifier is applied to models with UV maps, the resulting models will have the same UV islands, which can then be displaced to a quadrant to the side to not interfere with the baking process.

FIGURE 227. CARPINTROYER'S LEG UV MAPS

5.3.4. Texturing: The fire from within

Texturing Carpintroyer was as simple as adding three colours to the body in varying ways: red, white and black. The primary colours of this design's colour palette and distribution are based on the Pileated Woodpecker, the most known species of woodpecker, which is characterized by having a red mohawk with accents and black stripes on the face while being covered from neck to toes in black feathers for the most part. As such, the artist painted the bird using this same colour scheme.

Similarly to Escárgotree, Carpintroyer was divided into different materials, each for every UV Map, to keep the balance between quality and optimization for this model at maximum. Most white and black areas were as simple as covering them with plain colour. However, the red and featherless parts had other procedural textures

using curvature masks and ambient occlusion enhancers — in black for the lighter tones like the legs and white for the darker ones like the black feathers of the back — to contrast the sculpting cavities baked from the high poly model.

Finally, the artist painted the feathers in black-and-white tones to reuse the same texture for the feathers instead of creating one for each type, as the colour of these could be modified using shader modifications.

FIGURE 228. CARPINTROYER TEXTURED, COMPARED TO A PILEATED WOODPECKER

5.3.5. Mounting: Feathering the body and the birth of a minion

As stated at the beginning of this section, to keep consistency with the original design while not having to create a new model entirely, the minion's model was created using parts of the original design with different variations in size. As such, the feathers became the top of the head, the eyes and beak were increased and shrunk respectively to make the creature look child-like, the sideburns became the minion's wings, and the legs were shrunken and widened. Thus, the minion could reuse most of the big bird's textures, optimizing the size of the final project. The textures were also kept grey, like the feathers, to be coloured afterwards.

FIGURE 229. FINAL MINION DESIGN, AS SEEN FROM THE FRONT

With all the textures necessary created, the complicated part of the models begins: mounting the feathers into the body. Just as instancing plants over Escárgotree's body, Carpintroyer's torso, legs, tail and head were covered in different tones of feathers with similar Geonodes to those seen in Figure 205, taking as reference quads with the feather's render as a texture, but rotated so that they followed the natural form of the body outwards the tail. This solved the issue with the body quite nicely, but the problem came with the wings.

FIGURE 230. CARPINTROYER AND THE MINION, AFTER HAVING APPLIED THE FEATHER-INSTANCING GEONODES

The wing shape of a bird is one of the most characteristic attributes which determines its flight speed, capabilities of holding itself in the sky and manoeuvrability in the air. Studying the best reference led to the creation of a new moodboard with reference images about woodpeckers' wing types. Since they are tiny creatures capable of switching speeds in the blink of an eye, the ones that best fit this description were the elliptical wings, with a very open arch expanding across the entire wingspan of the wing.

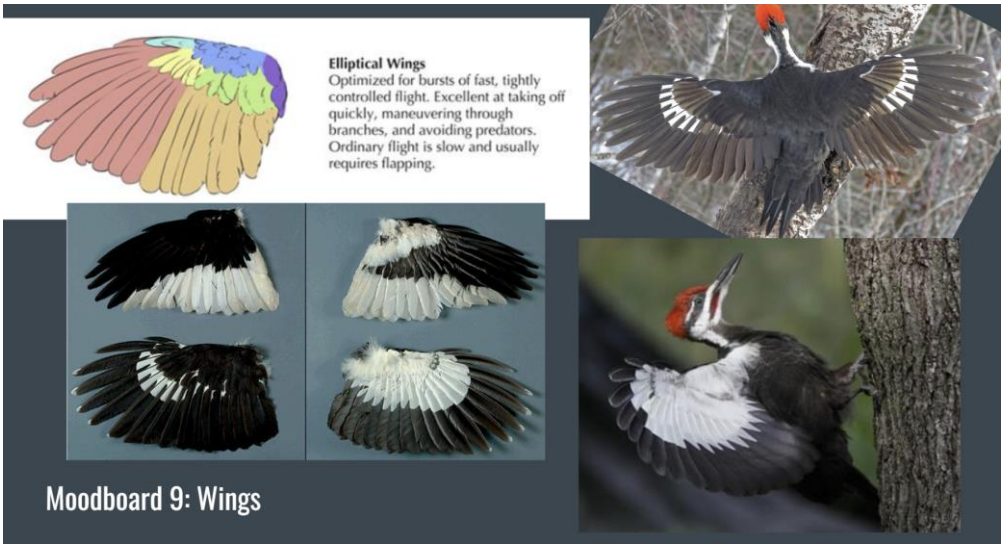

FIGURE 231. MOODBOARD 9: WOODPECKER WINGS

Several ideas were dabbled for the best way to create, from using array modifiers with specific curve trails to instancing them using Geonodes. Ultimately, placing each feather by hand was the best way to create these wings precisely as the author needed them. With that, many rows of varying quantities of feathers were meticulously placed across the wing, following the same shape as the one described in the moodboard. With the birds mounted, it was time for rigging and animation.

FIGURE 232. CARPINTROYER AND THE MINION, SEEN FROM THE FRONT

FIGURE 233. CARPINTROYER AND THE MINION, SEEN FROM THE BACK

5.3.6. Rigging & Animations: Bringing the birds to life

Much like Escárgotree's rigging process, Carpintroyer's skeleton was built using Advanced Skeleton for Maya, using the built-in template for bird rigs. This rig's features include all the necessary elements to create a functional bird skeleton; however, for other functionalities like eye movement, the author must add blendshapes to adapt the eye socket movement so that the eyeball does not leave the socket itself.

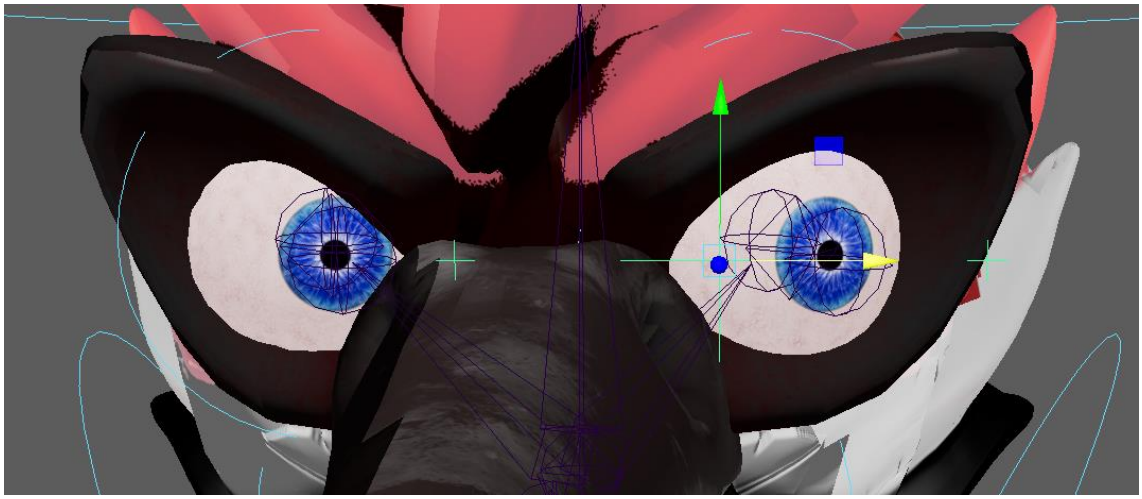

FIGURE 234. CARPINTROYER'S EYE BLENSHAPES IN ACTION

To animate some of Carpintroyer's animations, since some happen on the ground and others fly, the animator used reference primitive shapes representing the landing spot when the bird thrusts into the ground. However, this landing spot could vary, as well as the animation direction which Carpintroyer had to face. The author created all animations facing forward to solve this, as any future rotations would be done in code.

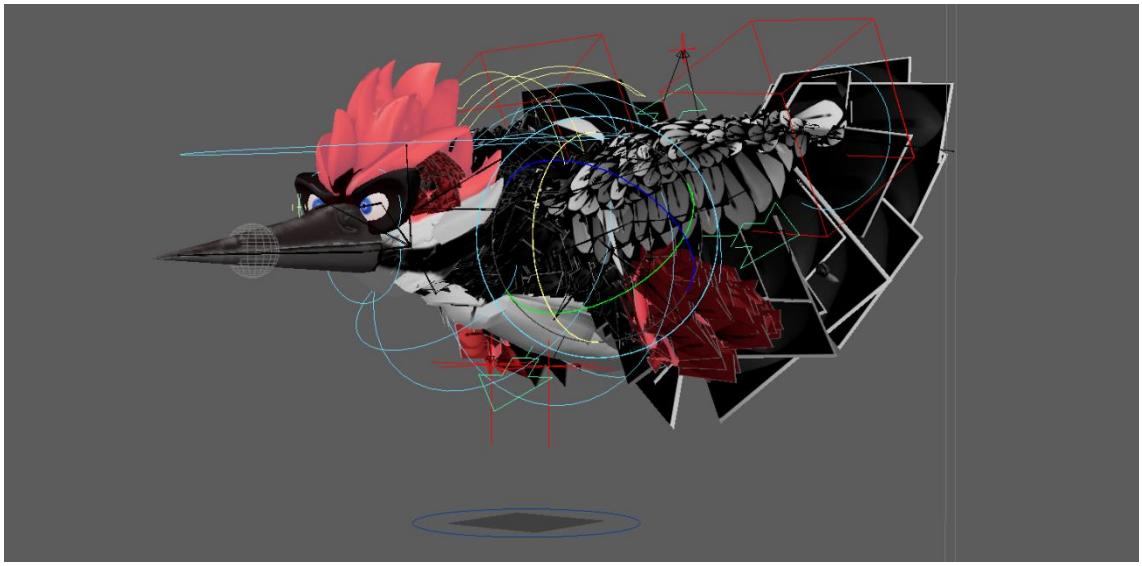

FIGURE 235. CARPINTROYER IN THE THRUST ANIMATION, WITH REFERENCE PRIMITIVES IN LIGHT GREY

In the case of the minion, its rig was a simple star-shaped rig, having its centre in the body and bones for each of the limbs all stemming from the core. Since its rig was very straightforward, so were its animations, these being a simple shake-up for idle flying and a more focused and angry-looking thrusting animation for when it is chasing the player, similar to Carpintroyer's.

FIGURE 236. THE MINION'S RIG, IN THE MIDDLE OF ITS FLYING IDLE ANIMATION

With the animations finished, the game asset production phases were finally complete, and it was time to go through post-production in-game.

FIGURE 237. CARPINTROYER, IN ITS IDLE ANIMATION

5.4. Post-production

In the post-production phase, the author implemented the final touches that improved the game's visual experience to the results seen in the final build. Post-production is usually used in film-making to refer to the final stage where footage is edited, sound is mixed and visual effects are added (ScreenSkills, 2023). However, for this production, it is used to refer to visual improvement and bug fixing.

This phase is divided into two parts. The first one tackles visual and sound effects for the entire fight and post-processing effects, cinematics, UI and other screens. The second one is the second playtesting session, which will help with the quality assurance of the product by getting feedback from users trying the game and thus will mainly be focused on bug fixing and optimizations.

5.4.1. Beta version: Sound, VFX, lights, cameras, action!

Starting with the post-production, the author created several filters for different game sections, improving the visuals by adding effects to the colour and camera. The main ones that affect the game's lighting are bloom, which increases the game's light diffusion by making things shinier, and colour adjustments to saturate the visuals and make them more vivid. Those which affect the camera are motion blur, which increases the blurriness of the image when moving the camera, and depth of field, which allows blurring of the image based on the distance from the player, thus centring the image on whatever the camera needs at each moment. Each effect varies in intensity based on the boss phase, as some effects that work well for one section are too intense in others.

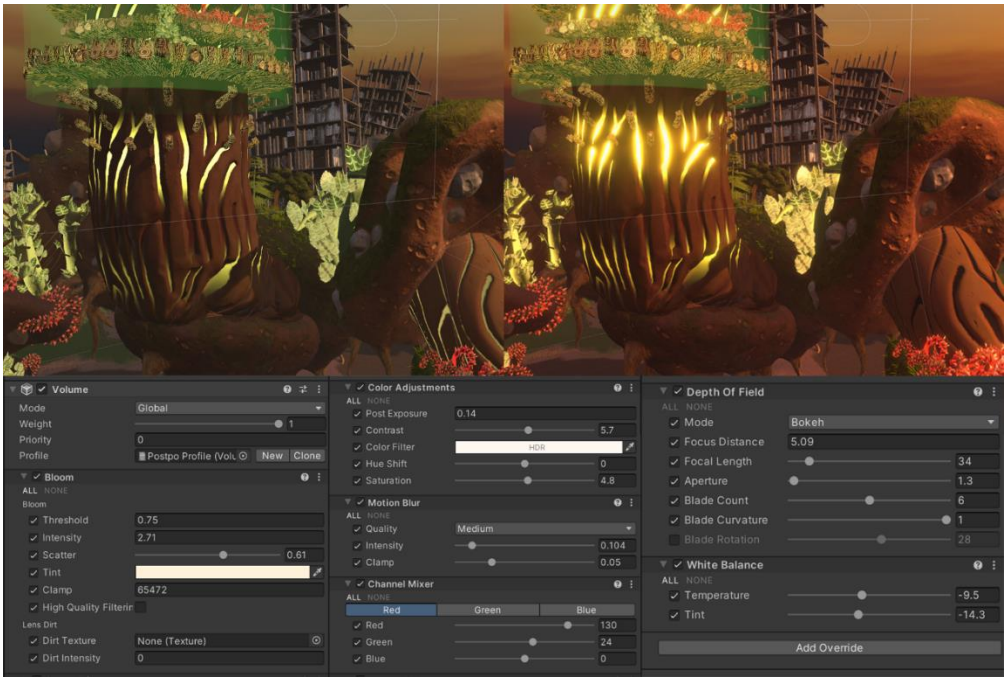

FIGURE 238. POST-PROCESSING SETTINGS FOR MOST OF THE GAME, WITH COMPARISON BEFORE (LEFT) AND AFTER (RIGHT)

Continuing with the effects started at 5.2.10, in this section, the author created other visual effects related mainly to the birds but also from both the intro sequence and the boss fight itself. Most of the game revolves

around a mechanic where most entities spawn expanding shockwaves. Initially, these shockwaves were simple primitive tubes that expanded over time; however, to give some flourish to these, the author created a shader to apply to them that makes them resemble cartoon air cycling outwards and particle systems of different kinds that burst when the shockwave spawns. This shader used a cycling Voronoi texture that moves across time, with parameters to switch speed and height.

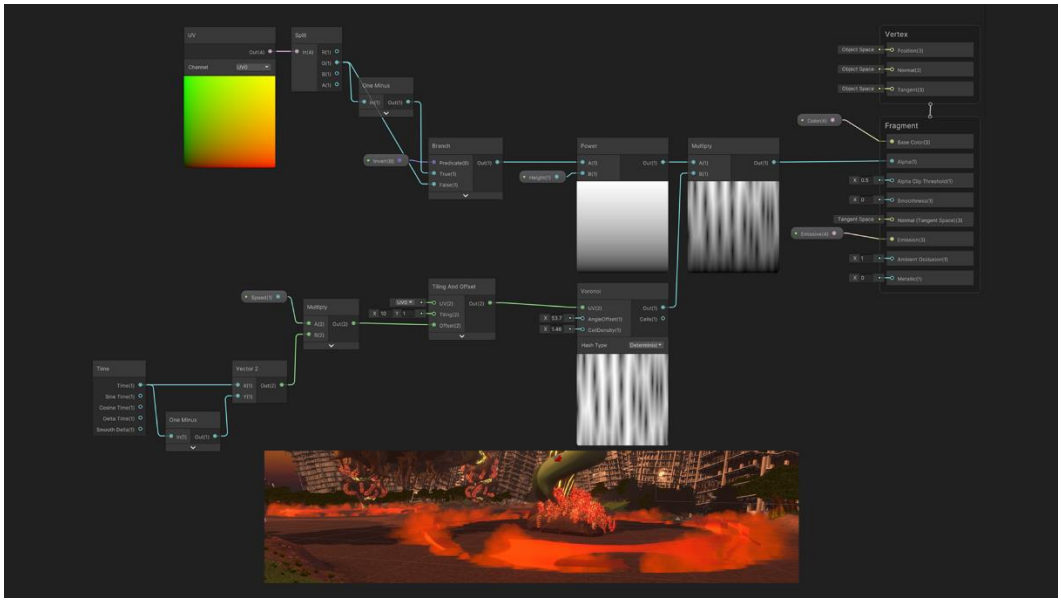

FIGURE 239. SHOCKWAVE SHADER WITH AN EXAMPLE OF ESCÁRGOTREE'S SHOCKWAVES

The author created a personalised shader for the electrical shockwaves at the intro sequence that gets a lightning texture and oscillates its opacity following a sinusoidal function, as well as a simpler shader for the loading energy tube to make it part of a material instead of another primitive. The twisting shapes of the particles result from the mesh being instantiated in each emitter burst.

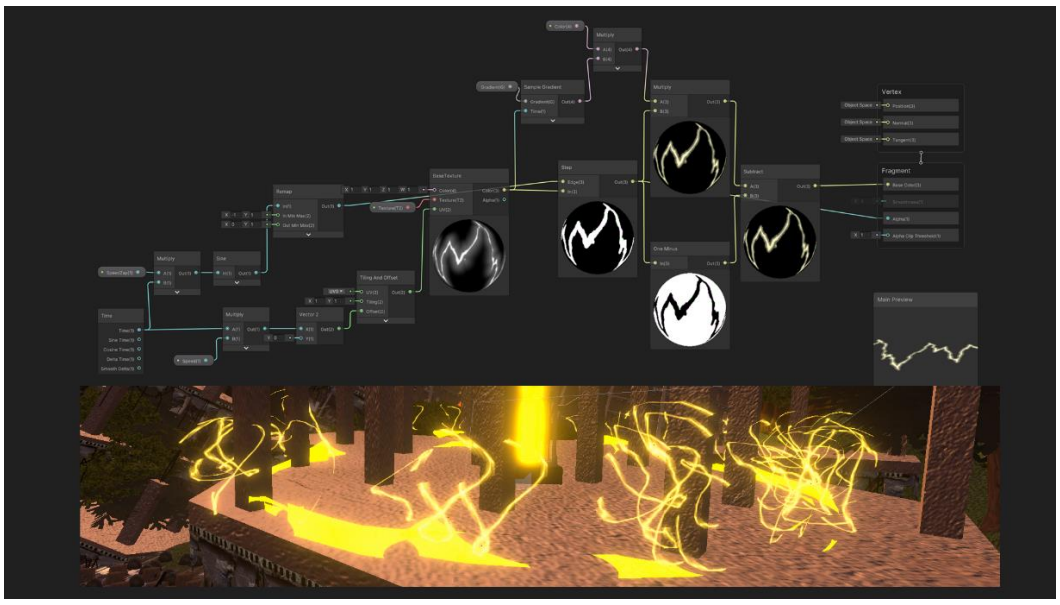

FIGURE 240. ELECTRIC PARTICLE SHADER WITH AN EXAMPLE BELOW

In the dummy's case, there were two particular visual improvements, the first of which improved how it was signalled to the player. Initially, the dummy was pointed at by a shadow created by a tree; however, this was too subtle for most players, who ignored the dummy completely. To answer this issue, particle systems in the shape of god rays were added, pointing to the target so that players would not miss it. Additionally, when it gets struck by a combo attack, it has a small animation that tumbles it down to the ground.

FIGURE 241. THE DUMMY BEING ILLUMINATED BY GOD RAYS

Most collectables and interactable hitboxes also received visual effects in the form of plain particle bursts. The heart fruits had the shape of pink hearts, while the keys had bursts of different coloured keys based on their colour. Objects like the dummy and Escárgotree's legs emit wood particles when struck in the shape of brown lines, and Carpintroyer sheds feathers when hit with the player's sword.

FIGURE 242. KEYS BURSTING FROM LIGHT-BLUE KEY INTERACTABLE

Speaking of which, Carpintroyer's design and visuals also received a colour overhaul. For the final battle at the peak, the author wanted the bird and the player to be as contrasting with the background as possible. For this purpose, the author changed the sub-boss colour scheme to bright purple instead of red because this colour

results from mixing red and blue, the hues of the minions. Its visual effects also use the shockwave shader, even as part of the bird's thrusting animation.

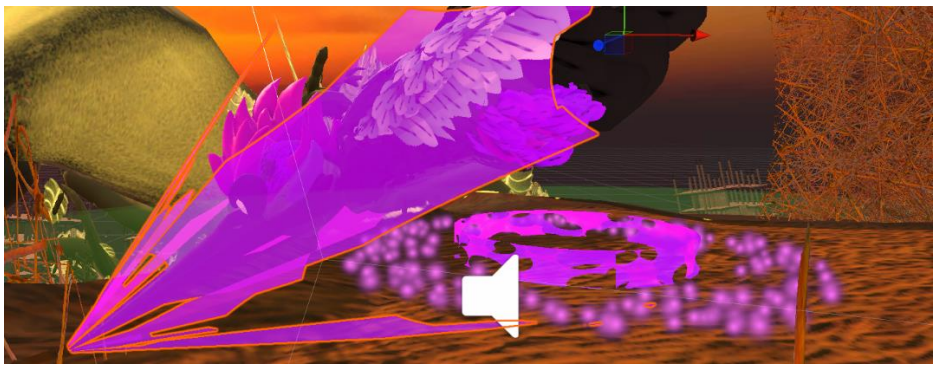

FIGURE 243. CARPINTROYER'S VISUAL EFFECTS

The author also produced a sound treatment of the entire game. Most sound effects come from Pixabay, a website publishing royalty-free stock resources like images, videos or, in this case, music and sound effects (Pixabay, s.f.). These would then be sound-treated using Adobe Audition, a program from the Adobe Suite specialized in audio equalization, mixing and post-production. Most of these treatments were related to volume regulation, mixing some sound effects with others and equalizing. These sounds would then be included in audio sources across the scene, activated using animation event callbacks or scripting calls and mixed using Unity's audio mixer, allowing for volume control and applying effects to the audio if needed. The author's objective was for every moving element of the scene to have visual and sound feedback so that players feel like everything is responsive.

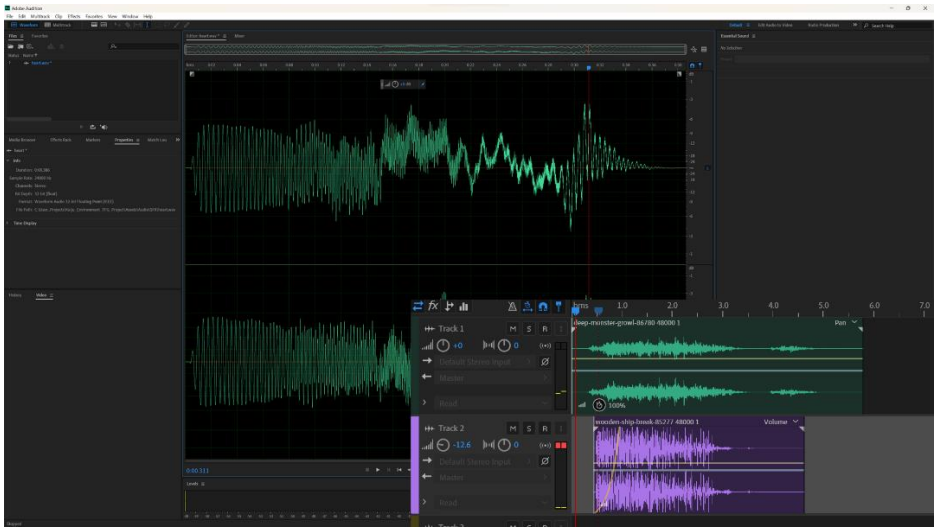

FIGURE 244. AUDIO EDITING USING ADOBE AUDITION

The writer also created cinematics across the game to give a cinematographic experience to the product and indicate crucial events on screen that were unclear during the first playtesting session (5.2.4). For each boss, an introduction and finishing animation were created, which would then be used in their respective cutscenes when they are introduced and when the first boss phase concludes for Escárgotree or, in the case of

Carpintroyer, when it dies. Other cinematics include the first key and door interaction since the feedback happens in the opposite direction of the interaction and when the player jumps into the leg. Foreseeing a possible difficulty for players, the author opted to automate getting on the 2D level by creating a scripted super jump that ensures they arrive at the beginning position of the maze.

FIGURE 245. INTRODUCTORY CUTSCENE TO ESCÁRGOTREE, LOOMING OVER THE PLAYER

Finally, the author ended up with three different screens regarding the UI. The main font used for this project was changed to REM, a font designed by Octavio Pardo, and the main interactable elements were left as Unity's default assets or stock images for symbols and logos. The first is the title screen, a simple screen that serves as a preamble to the game. The background was created using an edited screenshot of the background with several artistic blurs and effects that create a film-like image that keeps the elements on screen readable.

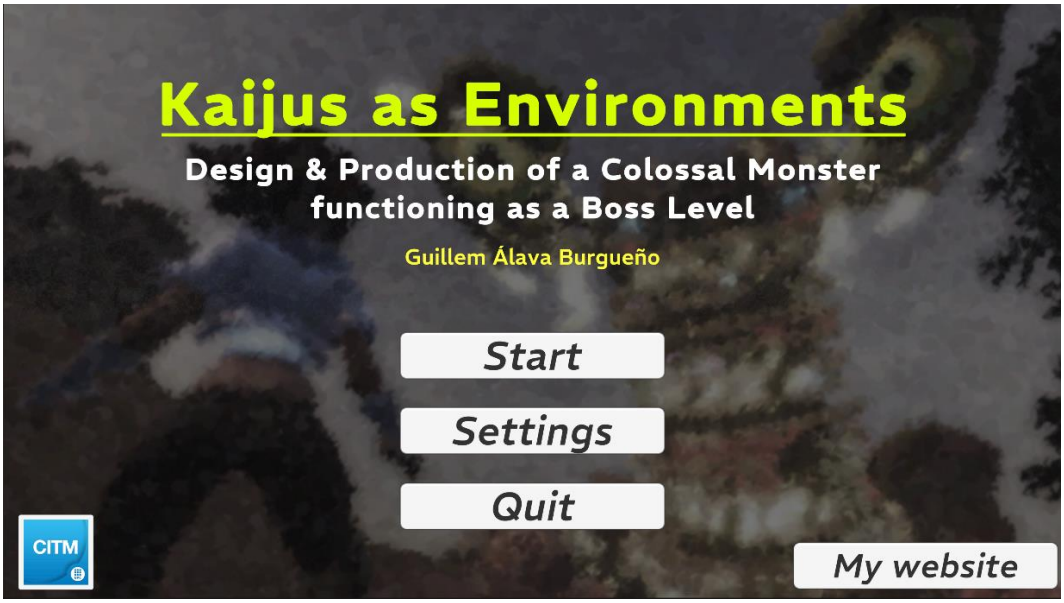

FIGURE 246. EARLY ITERATION OF THE TITLE SCREEN

The second one is the main game's UI. Using the following image as a reference for this explanation, the screen contains the following elements: from top to bottom and left to right. The first elements seen are always on screen, and they are a health bar composed of hearts based on the number of lives remaining and the player's current objective, with a yellow indicator that notifies when the objective changes so that players know what to do at any time. In the middle, the pause menu only shows when the player pauses the game, showcasing the game's controls and the changeable settings for the camera and volume, which can also be set from the title screen. Finally, at the bottom, the boss health bar is shown if major hitboxes are active, which is also divided into sections in the case of Escárgotree's Legs, each representing the life equivalent to a leg.

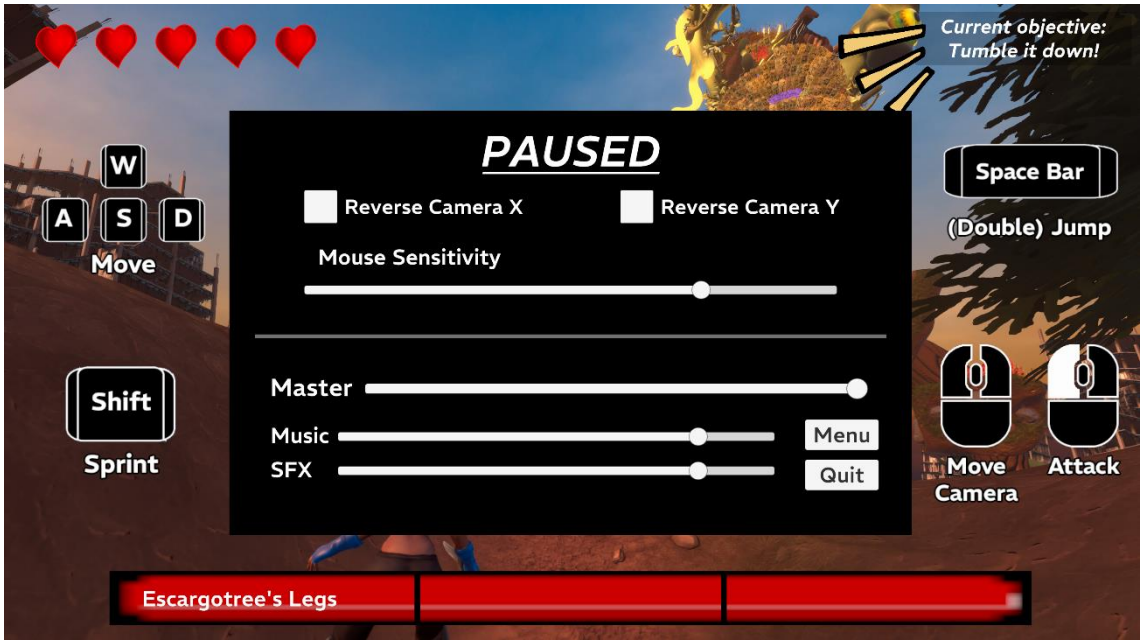

FIGURE 247. THE MAIN GAME'S COMPLETE UI

At last, there is the credits scene. At this point in development, the author and the tutor's names are listed, the project's primary purpose and licensing are mentioned for all external assets. After the second playtesting session, the author included additional thanks to the playtesters and other important people for the project. With this, the beta version is complete; thus, this game's second playtesting phase can begin.

6. Validation: Second Playtesting Phase

Before starting, the author conducted individualized tests to ensure no significant bugs or design flaws would interfere with crucial information for the game's quality assurance. Therefore, several errors were fixed before, mainly related to the player, such as the player's jittery movement, something dragged from several sections before that was caused by inconsistent camera updates based on the player's movements, as well as animation fixes and control of undesired actions and paths players may follow that could potentially break the game. These actions resulted in setting the 2D level's position to a predeterminate distance to the centre, something more effective than the platform colliders now made possible thanks to the cutscenes and adding invisible walls to prevent players from wandering off or falling out of places. The author started the playtesting when all detected bugs were fixed.

Organizing and coordinating a proper playtesting session like the first one was impossible at this point in development, so this playtesting phase was made as a publicly available form posted on the author's social media, discord communities like The Design Den[16] and the Game Design Study Group[17] from Barcelona and Reddit forums like r/playmygame and r/indiegames. This form was divided into three parts: playing the game, answering questions related to specific phases, and a final section in which people can freely express their other opinions about the game. It was translated into Spanish and English so more people could answer and understand the questions.

After many repeated promotions of the form through these channels throughout a week, 15 people wrote feedback using this form, similar to the first playtesting session. The target demographic of these groups primarily consists of people who play video games regularly and are passionate about game development, making the feedback obtained potentially valuable. However, the author asked additional questions about the respondents' personal information, such as their age range and the games they played that were studied in this project. In this form, most participants were between 19 and 25, leaning towards older ages over 20. Most of them have played the games listed in this form; however, it is essential to note that none of the contestants have played Solar Ash, probably due to the novelty and non-notoriety of the title compared to the other ones, as it is an indie title.

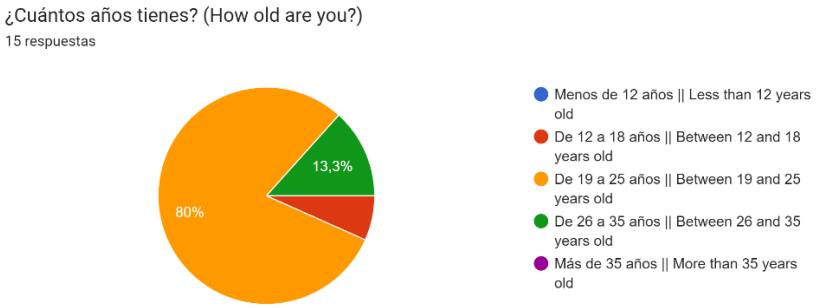

FIGURE 248. RESULTS FROM QUESTION 1 OF THE 2ND PLAYTESTING: AGE

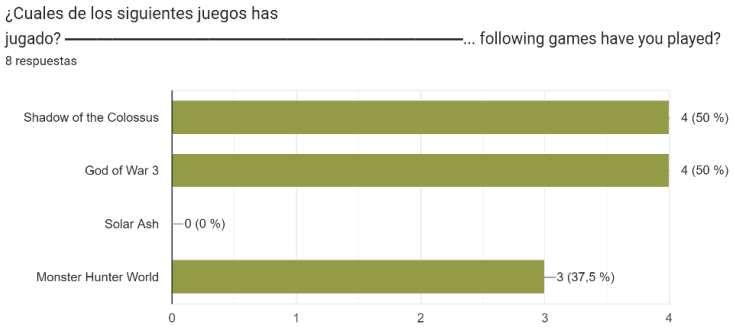

FIGURE 249. RESULTS FROM QUESTION 2 OF THE 2ND PLAYTESTING: GAMES PLAYED

For the first part, the author wanted to know elements regarding the experience's general aspects, like the duration and default settings. These questions helped significantly with the project's presentation, as they

[16] You can join here: https://discord.gg/thedesignden
[17] You can join here: https://discord.gg/DrT2cChC4R

delimited how it would be presented to the jury and the settings most people agree should be set by default. The duration of the experience for people was highly varied across the board, ranging between 5 and more than 30 minutes. Based on some testimonies from the survey, this disparity is primarily due to different skill levels from playing games — despite the playtime from some of them — the amount of bugs encountered that cut the experience short and the methodology followed to test the game.

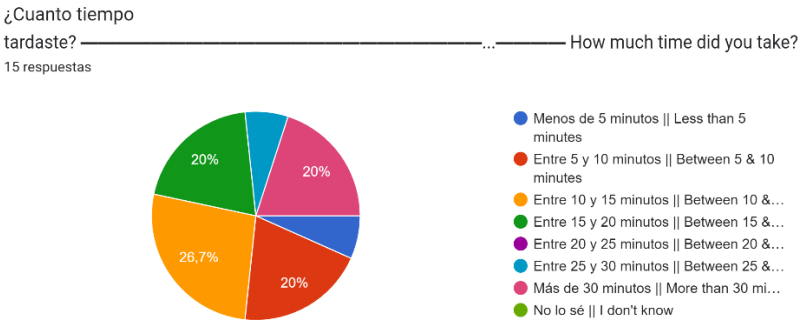

FIGURE 250. RESULTS FROM QUESTION 3 OF THE 2ND PLAYTESTING: TIME SPENT

Most of the settings tested were considered intuitive and well-regulated, with only a consensus regarding the music being louder than expected.

FIGURE 251. RESULTS FROM QUESTIONS OF THE 2ND PLAYTESTING REGARDING SETTINGS

Afterwards, there were questions related to the initial phases of the battle, including the player's learning curve during the introductory road and the first phase of the battle. The author wanted to ensure the respondents caught on to the main ideas shown and taught during these sections, as well as regulating the difficulty. Most players understood the game's main controls but struggled with concepts like the life bar and the three-hit combo. For the first phase, people understood the assignment and had little to no trouble understanding it.

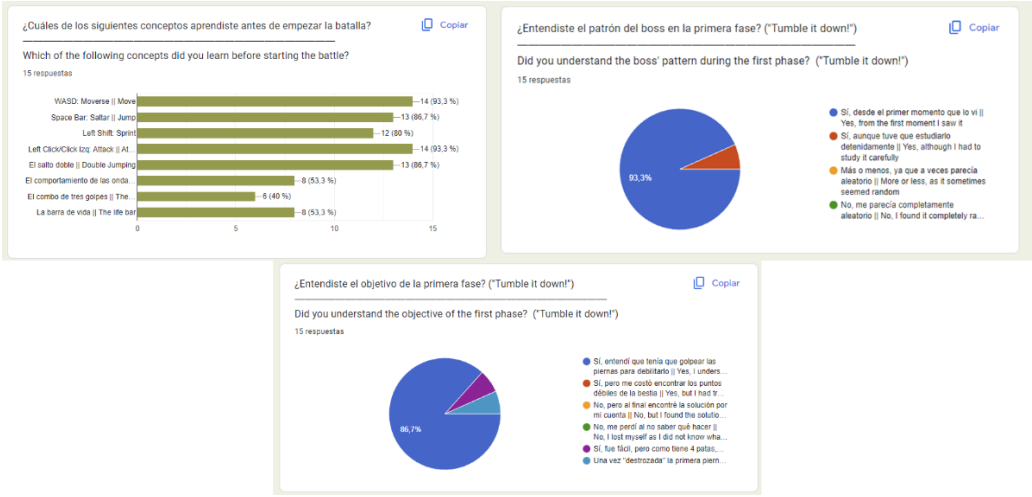

FIGURE 252. RESULTS FROM QUESTIONS OF THE 2ND PLAYTESTING REGARDING THE 1ST PHASE

The student also investigated the changes between phases, especially for the transition between the first battle and climbing through the tree. Since several mistakes appeared during the first playtesting session related to these parts, especially with the ever-changing cameras, getting lost without knowing where to go next, and getting on top of the beast, it was essential to check the improvements made for these parts worked. Most people understood where they had to go afterwards and how to navigate across the 2D sections, but they had trouble getting on top of the beast, even with the magnetized jump towards the trunk. The reasons varied from getting on top without any issues to not knowing how or what to do in the first place.

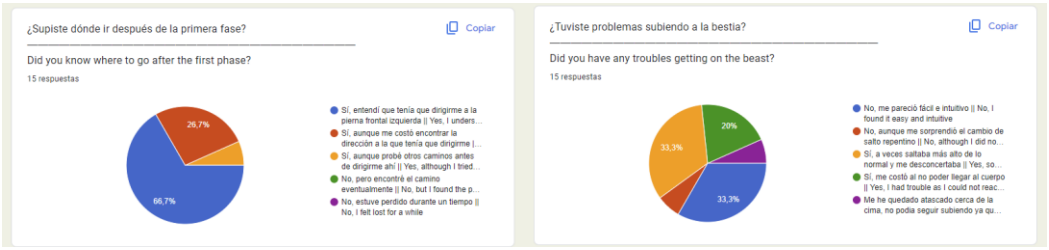

FIGURE 253. RESULTS FROM QUESTIONS OF THE 2ND PLAYTESTING REGARDING THE 1ST TRANSITION

Similarly to the first phase, the writer also studied the player's reactions to the second phase. These included the new controls for the 2D section, the objectives to achieve in the form of the keys, understanding all the elements involved in this part, and whether the number of lives included was enough. The questions aimed to see if this section was fair and easily understandable, and the results obtained revealed that players quickly understood most of the mechanics shown except for the keys and doors, especially the three located at the top, which mostly went unseen by players at first. In addition, the players estimated that an average of 0.6 healing

plants should be added, which the author disregarded due to the low amount of players that playtested, the majority voting that no plants should be added and the disparity of results that caused this result.

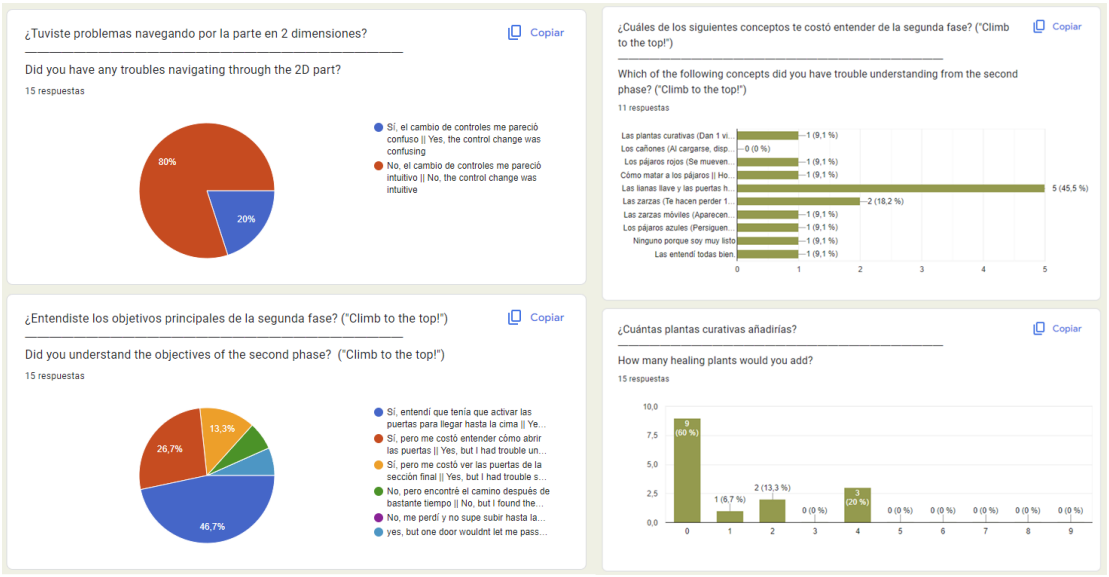

FIGURE 254. RESULTS FROM QUESTIONS OF THE 2ND PLAYTESTING REGARDING THE 2ND PHASE

Finally, it was time for the fight with Carpintroyer. The questions were related to the camera and the boss pattern, similar to the previous phases, to determine if the battle was fair and understandable. These questions revealed that reaching the top arena was complex and confusing, probably influenced by the cutscene trigger being lower than the final trunk platforms (those being Escárgotree's fangs) and the camera control increased the fight's difficulty unnecessarily, probably because the player cannot discern the height they jump as the camera points directly to the ground when jumping. Nevertheless, the boss pattern and fight were easily understood and challenging enough for them.

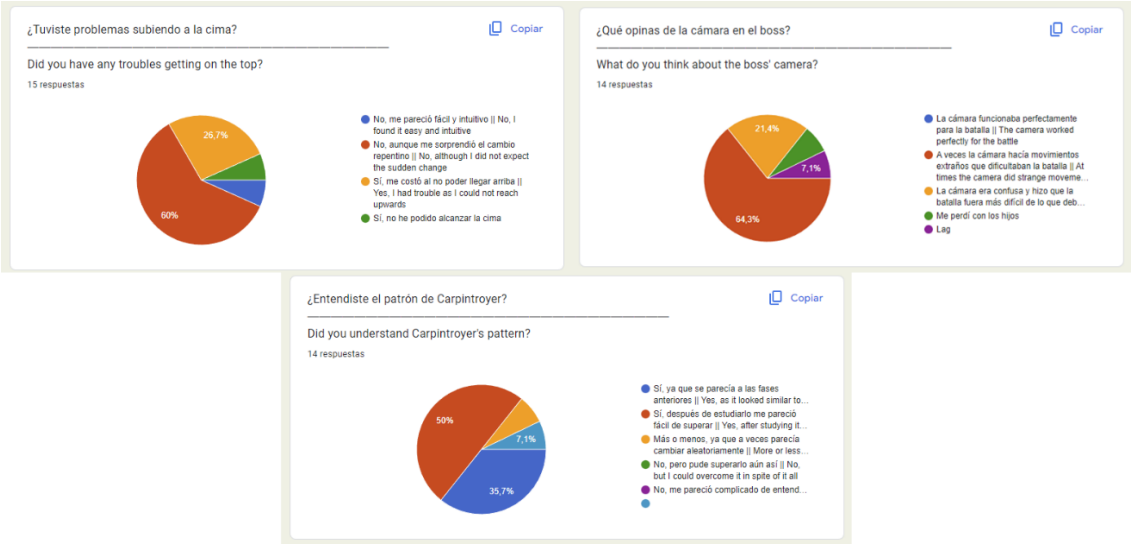

FIGURE 255. RESULTS FROM QUESTIONS OF THE 2ND PLAYTESTING REGARDING THE 3RD PHASE

An additional question was added to sum up the difficulty of the battle, which people mostly regarded as fair for the entirety of the fight. Specifically, the first phase was easy but fair, the second was fair but challenging, and finally, Carpintroyer's battle was just suitable for users, showcasing that the game's difficulty study and analysis was compelling.

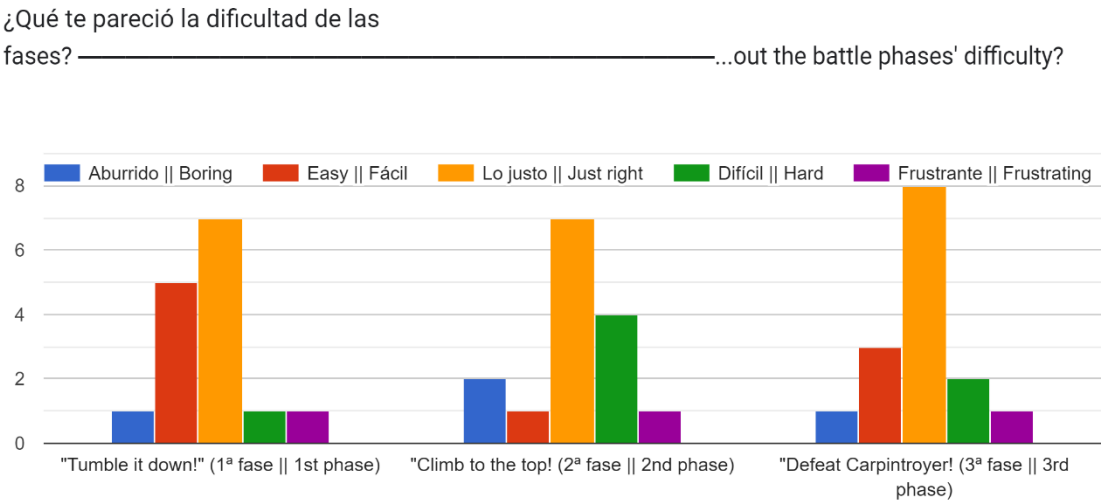

FIGURE 256. RESULTS FROM THE FINAL QUESTION OF THE 2ND PLAYTESTING: DIFFICULTY

In the free section, the general score from the public was a solid 7 out of 10. Many bugs were detected regarding collider issues that could get confusing sometimes from both platforms and shockwaves, a soft lock bug regarding a key that became inaccessible if skipped, and most user complaints were related to the first phase's camera controls, which became controversial for some and confusing for others, as it was a type of control not very seen in game's current standards. This correlates with the production phases, as the 3rd-person camera was one of the most underdeveloped parts of the project due to it not being deemed a priority by the student.

Nevertheless, people truly enjoyed the boss fight, highlighting the change between 3D and 2D as exciting, intriguing and visually attractive. Thematically, people loved the concept and the ideas presented for the fight, and everyone highly praised the art style, colour palettes and amount of detail. Despite some issues regarding the narrative cohesion of it all, the project was successful artistically speaking.

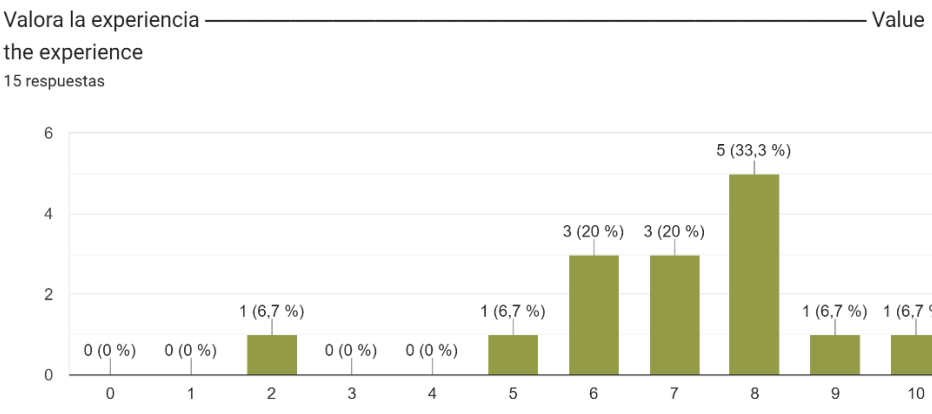

FIGURE 257. RESULTS FROM THE 2ND PLAYTESTING OF THE GENERAL EXPERIENCE

With all this feedback, the author proceeded to improve on the problems found by users. First and foremost, the initial area received changes to ensure that the tutorial concepts were understood before even entering the battle. An additional obstacle was added to teach players about a single jump, additional lighting was added to guide players towards the correct path, and finally, more dummies were added so that players do not miss on the three-hit combo. These changes made the path teach the information gradually and efficiently while guiding players towards the correct direction.

FIGURE 258. FINAL VERSION OF THE TUTORIAL AREA

Secondly, the transition areas received increased guidance while reducing the "movement magnetization" amount towards the corresponding areas like the trunk or the final area. These changes were implemented by adding a small text prompt and lighting in the part where the player jumps to the trunk and reducing the trigger area of the bird cutscene to allow players to reach the beast's fangs.

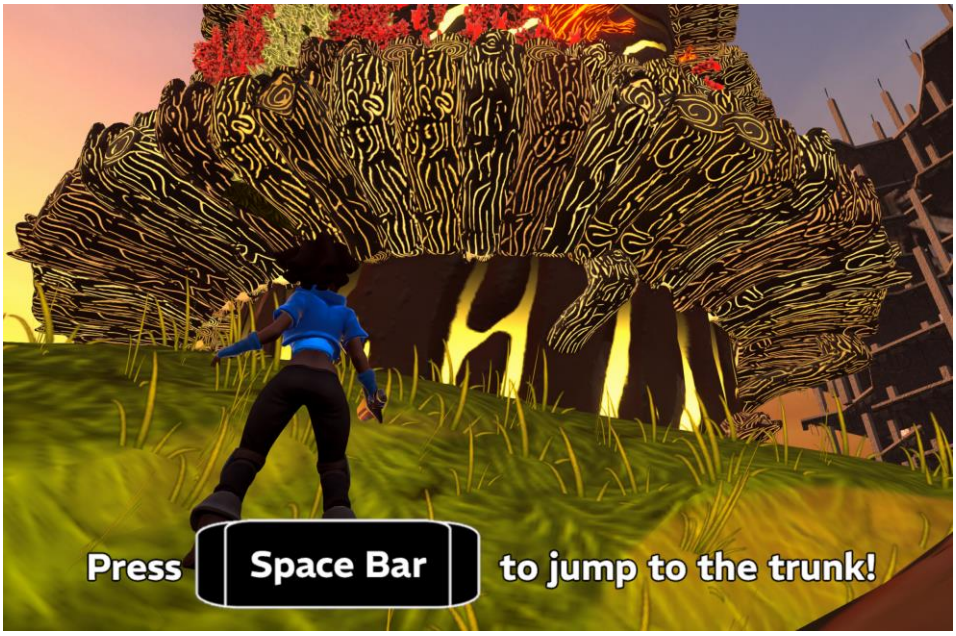

FIGURE 259. TEXT PROMPT TO JUMP TO THE TRUNK

Finally, it was crucial to improve the camera control of the game. The author considered improving the third-person camera, as it was the primary issue players found for the game; however, the camera remained unchecked due to time constraints and the inability to know and create a proper solution to this problem. Since it was the shortest part of the experience, the author did not prioritize this problem and instead focused on fixing the 2D cameras, especially the one in Carpintroyer's battle arena. The best camera found by the student fixes the rotations to only gyrate around the player while raising and falling depending on the player's height, following a similar behaviour to the camera found in the trunk section.

FIGURE 260. NEW CAMERA BEHAVIOUR FOR CARPINTROYER'S BOSS FIGHT

With additional fixes like the colliders and triggers, orthographical errors, the soft lock, increased contrast and settings regulations, the project is finally officially complete and ready to ship. In the following days before the delivery, the project only received minor hotfixes to correct mistakes not seen during the improving phase.

7. Conclusions & Lines of Future

At the beginning of this production, the author planned to accomplish seventeen objectives with the execution of this project. These objectives were based on studying existing colossal boss fights in the industry and how they applied level design elements to these fights to enhance the experience of battling against them and creating a boss from an artistic point of view. With this study, the author would create a playable boss fight that simultaneously served as a game level, focusing on following the artistic process of creating a 3D boss and developing a working prototype people could download and play, thus showcasing the author's artistic prowess and the extent a single developer can achieve by developing a product of this calibre.

After all processes and required tasks are completed at the end of the production phase, the final product consists of a three-phase boss fight with an additional introductory phase, allowing players to test the game's controls before going against the beast. The original design was respected as much as possible, requiring the minimum number of changes in order to present a boss fight that contains two entirely blocked, sculpted, retopologized, texturized, rigged and animated creatures, with additional environmental elements and other simpler creatures that complement the experience by contributing to the creation of a meticulously designed 2D level. All of this production follows the steps of a professional game studio, with additional support from external assets for indirectly related elements to the production of the 3D models, which combined make for a short experience that could serve as a tech demo for a game.

Technically, it could be argued that the project's general objectives were accomplished twice — one for Escárgotree and another for Carpintroyer. However, most of the production was focused on the former, as it required the most assets to be brought to life compared to the latter, whose model production lasted less time due to the practice and learned lessons obtained with the first boss. By the time of writing this text, the project required the most effort to complete, marking it as the most significant project for the author and the most incredible learning experience.

From a design perspective, creating a boss fight involves far more than this project showcases. Since this boss fight focuses on the battle itself only, it misses the required buildup these confrontations require to be effective and the taught lessons it has to test to the player. Most memorable bosses are always related with solid emotion and connection to the player, with some element contrasting the game. However, since no actual game precedes the experience, the fight ends up working on paper, but it does not have the same impact as other bosses in the industry nor an excellent adaptation to a curve difficulty since it goes from zero to a hundred in terms of quantifying the battle's difficulty. It is also worth noting the importance of the camera when developing a game of any kind. This project was a valuable learning experience for developing camera controls for other future games.

From an artistic perspective, this project has helped the author discover many new programs and tools from software already known by the author that will become helpful with future projects. For instance, the student used several new tools like Substance Painter's new baking tools or ZBrush's Nanomesh. However, the late inclusion of Blender has served to accelerate the process of creating the models' final versions with the usage of Geonodes, as well as creating simpler models and quick fixes for the game. Creating a project of this size

has served to test and play with most features Unity has to offer, having discovered new tools such as VFX and shader creation and a better understanding of already-known features.

However, outside of more extrinsic knowledge, the student also understood the importance of project playtesting and validation from oneself and others. In the former's case, some of the project's most visual features would not have been possible if the author had not distanced himself from the production to focus on unrelated projects. Returning to a project with fresh eyes and new knowledge has served as great importance to see mistakes invisible before. In the latter's case, without the collaboration of all the people who participated in both playtesting sessions, this project could not have achieved the level of care and meticulous quality assurance behind it.

Despite the previously mentioned issues related to the battle's design, these do not interfere with the battle in isolation. By the time of writing these lines, there are no more plans for this project other than being a standalone experience to be showcased in the author's portfolio. However, it is possible to use this battle's concept and assets in other future projects, adapting the game design elements and mechanics to the project it is imported. With that in mind, there are several elements to go through in the future to improve on this battle, such as better integrating the level's platforms with the main design with intertwining vines, an idea discarded at the beginning of production for being too tricky, time-consuming and challenging to improve with iterating but that would have probably been better for the general aesthetic the author wanted to apply to the creature at the end of production.

Nevertheless, may this serve as the presentation ticket to the student's beginning of the laboral life by showcasing the abilities learned during this degree.

Bibliography / Webography

Animation-Studios. (n.d.). *Advanced Skeleton | Maya Rigging Tools*. Retrieved from animation-studios: https://www.animationstudios.com.au/advanced-skeleton

Autodesk Interactive. (n.d.). *Geodesic Voxel binding*. Retrieved from Autodesk Maya Creative: https://help.autodesk.com/view/MAYACRE/ENU/?guid=GUID-5EFDB81B-E332-4D6C-B1BB-0B989AD2F2C7

Barriga, L. (2003). *Bossy Behaviour: Patterns and Techniques in Boss Design*. Retrieved from GDC Vault: https://www.gdcvault.com/play/1022625/Bossy-Behavior-Patterns-and-Techniques

Beane, J. A. (2012). *3D Animation Essentials*. Sybex Inc., U.S.

Berkun, S. (2020, September 21). *All ideas are made of other ideas*. Retrieved from Scott Berkun Web site: https://scottberkun.com/2011/ideas-are-made-of-other-ideas/

Brighid, F. (2022, July 5). *The Ultimate Guide to Character Design Concept Art*. Retrieved from Skillshare Blog: https://www.skillshare.com/en/blog/the-ultimate-guide-to-character-design-concept-art/

Coggan, G., & Guilcher, A. L. (2022, December 19). *The best 3D modelling software in April 2023*. Retrieved from Creative Bloq: https://www.creativebloq.com/features/best-3d-modelling-software

Coron, T., & Guilcher, A. L. (2022, November 30). *The best animation software in April 2023*. Retrieved from Creative Bloq: https://www.creativebloq.com/buying-guides/best-animation-software

Csikszentmihályi, M. (1990). *Flow: The Psychology of Optimal Experience*. Harper & Row.

de Vries, J. (2014). *Hello Triangle*. Retrieved from LearnOpenGL: https://learnopengl.com/Getting-started/Hello-Triangle

Design Doc. (2019, May 2). *Boss Battle Design ~ Design Doc*. Retrieved from YouTube: https://www.youtube.com/watch?v=wUPugc51udc

Design Doc. (2019, October 3). *Boss Battle Design Vol. 2 - Designing Engaging Boss Fights in Games*. Retrieved from YouTube: https://www.youtube.com/watch?v=o0CKHLumTG0

Design Doc. (2020, December 21). *Boss Battle Design Vol. 3 - Okami, Sonic Mania, and what's wrong with Zorah Magdaros*. Retrieved from YouTube: https://www.youtube.com/watch?v=6EIbdNIvl5Q

Design Doc. (2022, January 13). *What Makes A Good Colossal Boss?* Retrieved from YouTube: https://www.youtube.com/watch?v=k1hFnjUFSsY

Doodley. (2022, October 7). *Animating ARMS (FK vs. IK) - Doodley*. Retrieved from YouTube: https://www.youtube.com/watch?v=JnkAlwMjalc

ERI Economic Research Institute. (n.d.). *Salary Expert - Salary Data by Job Title and City | Salary Research*. Retrieved from Salary Expert: https://www.salaryexpert.com/salary

Fourcade, F. (2014a, March 12). *The Art of Shadow of the Colossus (1/6): Introduction*. Retrieved from Game Developer: https://www.gamedeveloper.com/design/the-art-of-shadow-of-the-colossus-1-6-introduction

Fourcade, F. (2014b, March 16). *The Art of Shadow of the Colossus (2/6): Gameplay and Story*. Retrieved from Game Developer: https://www.gamedeveloper.com/design/the-art-of-shadow-of-the-colossus-2-6-gameplay-and-story

Fourcade, F. (2014c, March 21). *The Art of Shadow of the Colossus (3/6): Designing Agro*. Retrieved from Game Developer: https://www.gamedeveloper.com/design/the-art-of-shadow-of-the-colossus-3-6-designing-agro

Fourcade, F. (2014d, April 3). *The Art of Shadow of the Colossus (4/6): Visual Design*. Retrieved from Game Developer: https://www.gamedeveloper.com/design/the-art-of-shadow-of-the-colossus-4-6-visual-design

Fourcade, F. (2014e, April 14). *The Art of Shadow of the Colossus (5/6): Music*. Retrieved from Game Developer: https://www.gamedeveloper.com/audio/the-art-of-shadow-of-the-colossus-5-6-music

Fourcade, F. (2014f, April 27). *Shadow of the Colossus (6/6): Conclusion*. Retrieved from Game Developer: https://www.gamedeveloper.com/design/shadow-of-the-colossus-6-6-conclusion

Glassdoor, Inc. (2023). *Sueldos de la empresa*. Retrieved from Glasdoor: https://www.glassdoor.es/Sueldos/index.htm

Guinness, H. (2023, March 14). *The best AI image generators in 2023*. Retrieved from Zapier: https://zapier.com/blog/best-ai-image-generator/

Holleman, P. (2011a). *Reverse Design: Super Mario World*. Retrieved from The Game Design Forum: https://thegamedesignforum.com/features/RD_SMW_1.html

Holleman, P. (2011b). *Super Mario World's Historical Context*. Retrieved from The Game Design Forum: https://thegamedesignforum.com/features/RD_SMW_2.html

Holleman, P. (2011c). *Measurements and Their Meanings*. Retrieved from The Game Design Forum: https://thegamedesignforum.com/features/RD_SMW_3.html

Holleman, P. (2011d). *Challenges, Cadences and Skill Themes*. Retrieved from The Game Design Forum: http://thegamedesignforum.com/features/RD_SMW_4.html

Holleman, P. (2015, November 12). *How to Design Levels With the "Super Mario World Method"*. Retrieved from Game Development Envato Tuts+: https://gamedevelopment.tutsplus.com/articles/how-to-design-levels-with-the-super-mario-world-method--cms-25177

Holleman, P. (2015, December 10). *The Super Mario World Method: Using Evolutions and Expansions*. Retrieved from Game Development Envato Tuts+: https://gamedevelopment.tutsplus.com/articles/the-super-mario-world-method-using-evolutions-and-expansions--cms-25295

Keren, I. (2018, November 6). *Boss Up: Boss Battle Design Fundamentals and Retrospective*. Retrieved from GDC Vault: https://gdcvault.com/play/1024921/Boss-Up-Boss-Battle-Design

Lazzaro, N. (2013, October 12). *Nicole Lazzaro | Games and the Four Keys to Fun: Using Emotions to Create Engaging Design*. Retrieved from YouTube: https://www.youtube.com/watch?v=EEmNRRRqgNc

Lee, T. (2015, September 28). *An annotated history of video game boss battles*. Retrieved from Polygon: https://www.polygon.com/features/2015/9/28/9333685/annotated-history-boss-battles

Masons, F. (2021, October 3). *Types of leaves*. Retrieved from The Tree Center: https://www.thetreecenter.com/types-of-leaves/

Midjourney. (n.d.). *Home Page*. Retrieved from Midjourney: https://www.midjourney.com/home/?callbackUrl=%2Fapp%2F

Millard, A. (2018, February 1). *The Anatomy of a Bossfight*. Retrieved from YouTube: https://www.youtube.com/watch?v=6BL0O7CL0N4

Mukherjee, S. (2023, February 2). *Different types of leaves with names and pictures*. Retrieved from Science Facts: https://www.sciencefacts.net/types-of-leaves.html

Nintendo. (1989). *Metroid: Instruction Booklet*. Redmond, WA: Nintendo of America Inc. Retrieved from https://www.nintendo.co.jp/clv/manuals/en/pdf/CLV-P-NAAQE.pdf

O'Neill, C., & Velazquez, B. (2012, March). *The Titans: Creating Living, Breathing Levels in God of War III*. Retrieved from GDC Vault: https://gdcvault.com/play/1015524/The-Titans-Creating-Living-Breathing

Petty, J. (2019, January 4). *What is 3D Modeling & What's It Used For?* Retrieved from Concept Art Empire: https://conceptartempire.com/what-is-3d-modeling/

Piquet, L. (2012, October 7). *[Breakdown] Shadow of the Colossus (PAL - PS2)*. Retrieved from Froyok - Léna PPiquet: https://www.froyok.fr/blog/2012-10-breakdown-shadow-of-the-colossus-pal-ps2/#:~:text=-A%20lot%20of%20tiling%20textures,128%20most%20of%20the%20time).

Pixabay. (n.d.). *4.1 million+ stunning free images to use anywhere*. Retrieved from Pixabay: https://pixabay.com

Pixar. (1989). *The RenderMan Interface Specification v3.1*. Massachusetts: MIT. Retrieved from RenderMan 20 Documentation.

Ramirez, D. (2021, October 23). *What is the mysterious rule of three? — Rule of three*. Retrieved from Rule of three: https://www.rule-of-three.co.uk/articles/what-is-the-rule-of-three-copywriting

Riley, S. (2022, October 4). *How AI Image Generators Work (Stable Diffusion / Dall-E) - Computerphile*. Retrieved from YouTube: https://www.youtube.com/watch?v=1CIpzeNxIhU

Rogers, S. (2014). *Level up! The Guide to Great Video Game Design*. Chichester: John Wiley & Sons, Ltd.

Ruswick, T. (2017, November 20). *Boss Battles - 5 Things Every Video Game Boss Needs*. Retrieved from YouTube: https://www.youtube.com/watch?v=o9DRLW6AnLs

ScreenSkills. (2023). *Post-production*. Retrieved from ScreenSkills: https://www.screenskills.com/job-profiles/browse/post-production/#:~:text=Post-production%20is%20the%20final,or%20even%20get%20%E2%80%9Csaved%E2%80%9D.

Segarra, L. M., Gach, E., & Fahey, M. (2021, February 19). *Gaming's Best And Worst Bosses (with Stephen Totilo)*. California, United States of America: Kotaku Splitscreen. Retrieved from https://podcasts.apple.com/us/podcast/gamings-best-and-worst-bosses-with-stephen-totilo/id1039413502?i=1000509773261

Snowman Gaming. (2016, November 26). *Good Game Design - Bosses*. Retrieved from YouTube: https://www.youtube.com/watch?v=YmwLPF11eos

Staff, C. (2018, March 31). *Introduction To Thumbnailing And Quick Sketching*. Retrieved from Concept Art Empire: https://conceptartempire.com/intro-to-thumbnail-sketching/

Take This, Inc. (2016, August 23). *Crunch Hurts: How unmitigated overwork harms employee health, productivity and your studio's bottom line*. Retrieved from Take This: https://www.takethis.org/wp-content/uploads/2016/08/CrunchHurts-TakeThis.pdf

Thomas, F., Johnston, O., Luske, H., Tytla, V., Moore, F., Spencer, F., . . . Babbitt, A. (1995). *The Illusion of Life: Disney Animation*. Disney Editions.

TV Tropes. (n.d.). *Humiliation Conga*. Retrieved from https://tvtropes.org/pmwiki/pmwiki.php/Main/HumiliationConga

Wijman, T. (2023, February 15). *The Games Market Will Grow by 2.1% to Reach $196.8 Bn in 2022*. Retrieved from Newzoo: https://newzoo.com/insights/articles/the-games-market-will-show-strong-resilence-in-2022

8. Appendixes

8.1. Trailer production: The best of both worlds

Like any media nowadays, the best way to advertise a video game is with a trailer. Trailers serve as short yet effective showcases of information to explain to the public what the developers are offering to them. Likewise, for these final degree projects, there must be an accompanying video that showcases the process or the final product the author has created, sometimes both. As such, the author wanted to create a proper short video that could serve as both a final degree project video trailer and a proper video game trailer.

For this, the student analyzed several videos from past university works and professional studios to extract the elements that made them successful in the first place. In both cases, unless it is a feature that is important to the game, they do not include voice-overs narrating the action. This practice was used decades ago to advertise movies but has decayed in recent years. In the case of the final project videos, they showcase processes in video format of how they did their projects, usually at fast speeds to shorten the video duration, with stock music in the background to avoid copyright infringements. In the case of video game trailers, they showcase gameplay and attractive, eye-catching bits to get people's attention and pique their interest.

As such, the best way the author organized the video trailer was to showcase the creation process during the first half of the video and the gameplay bits in the second half, with some before and after shots to compare the progress across the different phases. The program used for this project was Adobe Premiere, a powerful video editing tool created by the Adobe Suite that sufficed the trailer's needs.

For the latter half, the author recorded a gameplay video completing the game and trimmed it into parts that could interest viewers, like the intro sequences of the bosses and some action shots climbing up Escárgotree, with additional clips from an earlier development video that compare specific parts of gameplay to the game's final look. However, the author had a problem with the former half, as they did not record any part of the process beforehand. Nevertheless, the solution came about in the form of turnarounds from all the previously saved files from each stage of development, showcasing Escárgotree and its evolution across the entire production. Most turnarounds were created using a camera animation rotating around the target in question; however, some had to be created manually due to some program's limitations.

With some additional editing, text and copyright-free music, the student created a trailer lasting 1 minute and 40 seconds, working as both a final degree project trailer and a video game trailer proper.

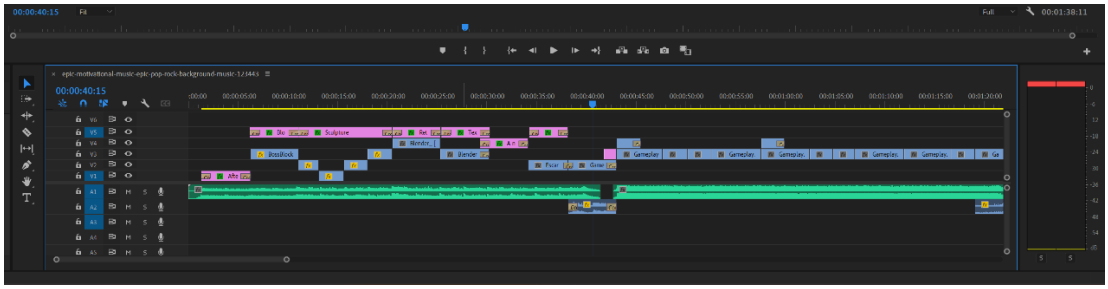

FIGURE 261. VIDEO EDITING CLIPS FROM THE TRAILER