



Universitat Jaume I de Castellón

Design and Development of Videogames Degree

Technical Report of the Final Degree Project

Psychological terror experience
with
photorealistic models and illumination

Student: **Manuel Arroyo Arrebola**
Tutor: **Francisco Ramos Romero**

Summary

This Project contain the design and development of a realistic 3D experience with the main goal of transmit fear, loneliness and suspense to the player. This experience is based on a realistic environment and characters to increase the amount of immersion. Then to improve the atmosphere of the game it is added sounds, realistic real-time illumination and shadows.

This project is mainly focused on the artistic part of a videogame, specifically 3D. The main objective of this game is to create a realistic environment formed by models and character completely credible. To transmit feelings to a player, such fear, it is so much easier with a photorealistic environment. The second objective of this project is to create a realistic and unique story, with survival horror and conspiracies elements. The third objective of this experience is to create puzzles in the game, making a challenge for those who wants to finish it.

The game is a big loop, each loop has iterations, and in each iteration, the history is dosed and distributed around the environment, by notes. This notes will be in some places of the environment; the user has to read them to continue the loop. The history will be told by some elements like a TV, a phone or even voices in his head. Also there some important part of the history that will be told by the pictures in the walls, those pictures has to be found first, and they will be hide around the environment. Meanwhile the player tries to find and understand it, it will be haunt by a character.

The game has been made in four steps, brainstorming, artistic elements creation, scripting, effects and testing.

The first step, brainstorming, thinking how to orientate and create the history, the environment, the gameplay and the game cycle. To create the history was necessary just two words, then it was constructed around that. What took more time was the structure of the environment because the main objective was to get visual impact and it took some sketches to get the final result.

The second step was the creation of artistic elements; this phase was the one that took more time to develop. The environment is plenty of assets, this go through different processes (concept, modeling, texturizing, animation and implementation) it takes time to develop just an asset.

The third step is the scripting process. The main objective of this experience is to get visual impact and easy gameplay. This step was not the main objective of the game even though this process was one that took more time because there are some tricky mechanics in the game.

The four step is testing and image-effects and post production. When something was done, it was necessary to test it and check if everything done last and before was working properly, if not it was fundamental to check where the bug is, debug and test again. In this process it was added all the post production effects to the camera, it has to be modified all the values of the post processing script to get the right visual effect for the experience.

Key Words

Survival horror, experience, puzzle, photorealism, 3D environment, narrative

1. Introduction	10
1.1 Motivation	10
1.2 Objectives	10
1.3 Tools	11
1.4 Justification	11
1.4.1 Narrativa Hipermedia y Analisis de videojuegos	11
1.4.2 Diseño 3D	12
1.4.3 Motores de Juegos	12
2. Game Design	13
2.1 History	13
2.2 Level design/environment design	14
2.3 Gameplay	14
2.4 Sounds	14
3. Art creation and Development	15
3.1 Artistic elements creation	15
3.1.1 Inorganic models creation	15
3.1.1.1 References and Concept Art	15
3.1.1.2 3D Modeling and UV's	15
3.1.1.3 Texturizing	18
3.1.1.4 Model Export	20
3.1.2 Organic models	25
3.1.2.1 Zbrush and starting a new project	25
3.1.2.2 Mary's face creation	28
3.1.2.3 Mary's body creation	29
3.1.2.4 Marvelous Designer and clothes modeling	30
3.1.2.5 From high poly to low poly	32

3.1.2.5.1 Retopology	32
3.1.2.5.1.1 ZRemesh	32
3.1.2.5.2 Projection	34
3.1.2.5.3 UV mapping	34
3.1.2.5.4 Maps	36
3.1.2.5.5 Decimation Master	37
3.1.2.6 Texturizing	38
3.1.2.7 Rigging	40
3.1.2.8 Skinning	41
3.1.2.9 Animation	42
3.1.2.10 Export organic model	43
3.2 Scripting	46
3.2.1 Character controller	46
3.2.2 Zoom in	48
3.2.3 Pick up and setVisible	48
3.2.4 Show description	50
3.2.5 Changing between scenes	51
3.3 Image effects	52
3.3.1 Antialiasing	52
3.3.2 Ambient Occlusion	52
3.3.3 Depth of Field	52
3.3.4 Motion Blur	53
3.3.5 Eye Adaptation	53
3.3.6 Bloom	54
3.3.7 Color Grading	54
3.3.8 Chromatic Aberration	55
3.3.9 Grain and Vignette	55
4. Results	57

5. Project deviations and objectives done	63
6. Hours control and planning	64
7. Conclusion	65
8. Bibliography	67

Figures - Index

<i>Fig. 0 – Modeled Assets in Maya</i>	16
<i>Fig. 1 – Uv editor, UV mapping asset</i>	17
<i>Fig. 2 – Substance Painter HUD</i>	18
<i>Fig. 3 – TV fully texturized</i>	19
<i>Fig. 4 – Vase fully texturized</i>	19
<i>Fig. 5 – Rocking Chair fully Texturized</i>	20
<i>Fig. 6 – Material with maps in Unity, Material menu.</i>	20
<i>Fig. 7 – Rocking chair and vase in Unity</i>	21
<i>Fig. 8 – Rocking TV in Unity</i>	21
<i>Fig. 9 – Lamp in Unity</i>	21
<i>Fig. 10 –Empty environment in Maya</i>	22
<i>Fig. 11 –Empty environment texturized in Unity (1)</i>	23
<i>Fig. 12 –Empty environment texturized in Unity (2)</i>	24
<i>Fig. 13 – Zbrush Interface</i>	25
<i>Fig. 14 – Project Started with a predefined female model</i>	26
<i>Fig. 15 – Project started with a simple sphere</i>	26
<i>Fig. 16 – All references images loaded into Zbrush</i>	27
<i>Fig. 17 – Differences between topology</i>	27
<i>Fig. 18 – Modeling Mary’s face. Steps</i>	28
<i>Fig. 19 – Mary ´s body modeling. Steps</i>	29
<i>Fig. 20 – Marvelous Designer Interface</i>	30
<i>Fig. 21 – Garment finished</i>	30
<i>Fig. 22 – Completed model in Zbrush</i>	31
<i>Fig. 23 – ZRemesher menu</i>	32
<i>Fig. 24 – Retopology difference</i>	33
<i>Fig. 25 – Projection</i>	34
<i>Fig. 26 – UV master calculating seam with paint</i>	35

<i>Fig. 27 – Opened model</i>	35
<i>Fig. 28 – Texture Map or Albedo Map</i>	36
<i>Fig. 29 – Normal Map</i>	36
<i>Fig. 30 – Decimation Master menu</i>	37
<i>Fig. 31 – Simplification difference</i>	37
<i>Fig. 32 – Mary in Substance Painter</i>	38
<i>Fig. 33 – Export options</i>	38
<i>Fig. 34 – Mary’s model finished</i>	39
<i>Fig. 35 – Completed rig and model</i>	40
<i>Fig. 36 – Rigging</i>	40
<i>Fig. 37 – Painting weights</i>	41
<i>Fig. 38 – Skinning problem</i>	41
<i>Fig. 39 – Animation process</i>	42
<i>Fig. 40 – Attacking animation</i>	42
<i>Fig. 41 – FBX export options</i>	43
<i>Fig. 42 – Prefab animation menu</i>	44
<i>Fig. 43 – Mary animator</i>	45
<i>Fig. 44 – FPS Controller Hierarchy</i>	46
<i>Fig. 45 – Camera Menu</i>	46
<i>Fig. 46 – Cookie, alpha mask</i>	47
<i>Fig. 47 – Without and with cookie</i>	47
<i>Fig. 48 – Full FPS controller</i>	47
<i>Fig. 49 – Zoom script</i>	48
<i>Fig. 51 – Zoom in</i>	48
<i>Fig. 52 – Crumbled photo</i>	49
<i>Fig. 53 – Picture</i>	49
<i>Fig. 54 – Raycast</i>	49
<i>Fig. 55 – Crumbled paper detection</i>	50

<i>Fig. 56 – Picture Description</i>	50
<i>Fig. 57 – Detect picture and show description</i>	51
<i>Fig. 58 – Subtitles timer</i>	51
<i>Fig. 59 – Load Scene</i>	51
<i>Fig. 60 – Example of Antialiasing</i>	52
<i>Fig. 61 – Environment rendered just with Ambient Occlusion</i>	52
<i>Fig. 62 – Depth of Field in Skyrim</i>	53
<i>Fig.63 – Motion Blur in Crysis 2</i>	53
<i>Fig. 64 – Eye Adaptation in a Demo in Unreal 4</i>	54
<i>Fig. 65 – Bloom effect in The Witcher 3</i>	54
<i>Fig. 66 – Color grading in a Unity Demo</i>	54
<i>Fig. 67 – Chromatic Aberration in Destiny</i>	55
<i>Fig. 68 – Left for Dead grain and vignette</i>	55
<i>Fig. 69 – Environment with different visual effects</i>	56
<i>Fig. 70 – Full completed environment</i>	57
<i>Fig. 71 – Environment left side</i>	57
<i>Fig. 72 – Environment right side</i>	58
<i>Fig. 73 – Environment top side</i>	58
<i>Fig. 74 – Environment top side, rest zone</i>	59
<i>Fig. 75 – Environment top side, hall</i>	59
<i>Fig. 76 – Lobby different illumination 1</i>	60
<i>Fig. 77 – Environment top side with Mary</i>	60
<i>Fig. 78 – Environment lobby different illumination 2</i>	61
<i>Fig. 79 – Environment top side, hall with Mary</i>	61
<i>Fig. 80- Loop Zone</i>	62
<i>Fig. 81 Hours planning</i>	64

1. Introduction

In this Final Degree Project, it wants to be achieve a realistic, surrounding and credible survival horror experience. The terror videogames have been expanding in the last years and they have been growing a lot. Even that, the videogames users still want more terror games. Be able to transmit a feeling as strong as fear is, is the main reason why this project has been chosen to be developed. Some elements like illumination and sounds takes the main role of this project, those elements become in one of the main axis to impact the player, if any of them fails and it is not realistic enough the whole setting will fall, the player won't take it seriously and everything done will be useless. So it is really important to look out for those elements.

Nowadays, many terror games leave narrative in the background and also that narrative has been used many times before. In this project, the narrative is one of the protagonist of the game, it has an innovative aspect and unique, differentiating it from the rest of the games of this style.

1.1 Motivation

There are many motivations to work on this project, the most important will be described next. Since the beginning of this degree it has been wanted to do a terror experience, and after all the knowledge acquired in this four years, it was the right time and the chance to do it. In all this four years, the only way to work was with teams, the individual ideas were always not used or modified by the rest of the team. The chance to work alone, to know their own limitation and capabilities, to work in an own project is one of the biggest motivation. Also, the videogames market is really competitive, it is fundamental a good portfolio to be hired for a company, and this Final Degree Project is the best way to achieve a good project to add to the portfolio. This theme for the game has been chosen because terror is one of the most powerful feelings that any visual content can provide and a good resource for a videogame.

1.2 Objectives

- Create a surrounding, credible terror experience, where the player can feel fear and tension and suspense.

- Learn how to create photorealistic assets, understand most possible how Autodesk Maya, Zbrush, Unity and Substance Painter works

- Learn how to create visual impactful lights and transmit with lights and sounds, using real time lights, placing them in the right spot to create the right shadow or effect.

- Create a deep and elaborate narrative. Many of histories in terror has been told, it is necessary to pick up a good base history and transform it with no boundaries. And if the history is based in something real, it will add more realism and credibility.

- Learn how to create a good postproduction to improve the realism on the scene, the difference between a scene without postproduction with a scene with postproduction is abysmal. It is completely necessary to add camera effects to improve the final result.

- Create interesting mechanics, puzzles.

- Learn to scripting and program the gameplay

- Learn the workflow to create a game. How it is the best way to obtain a model.

1.3 Tools

This experience has been created for midrange computers. The game has been developed with Unity 5.6 with the Operating System macOS Sierra. Now its going to be listed all the tools that have been used on this project:

- Autodesk Maya 2017, has been used to model all the environment and the assets
- Pixologic Zbrush, has been used to created the enemy.
- Substance Painter, used for texturizing all the models.
- Marvelous Designer 6, used to create all the garment. Clothes physics.
- Monodevelop, used to scripting
- Microsoft Word, used to create the Final Degree Report.
- Adobe Photoshop, to modify and paint some textures.
- Audacity, modify and recording audio.
- Crazybump, to obtain normal maps from a texture.

1.4 Justification

In this section is going to be explained how the project has a relation with some subjects of the degree. This Final Degree Project has been based on three subjects mainly “Narrativa Hipermedia y Analisis de Videojuegos” (VJ1218), “Diseño 3D” (VJ1216) and “Motores de Juegos” (VJ1227).

1.4.1 Narrativa Hipermedia y Analisis de Videojuegos

In this subject has been perfect for the student to learn how to create a videogame history, how to develop a character and how provides to the player the right amount and way to explain the history. How to use the environmental narrative to provide a different way to tell the history using assets. This project has a history that it is told by the tools that was learned in this subject.

- E11 – Capacidad para generar y analizar recursos expresivos y narrativos aplicados a discursos audiovisuales, con especial énfasis en el multimedia, hipermedia y videojuegos.
- G01 – Capacidad de análisis y síntesis
- E11 – Generar guiones audiovisuales.
- E11 – Usar la terminología más adecuada como consecuencia de una auténtica – comprensión de los procedimientos de generación, evolución y cronología básica de los discursos audiovisuales desde la perspectiva signficante.
- E11, G01 - Utilizar las nuevas tecnologías audiovisuales emergentes para la elaboración de discursos audiovisuales en el terreno de la publicidad, del diseño gráfico, la infografía y los productos multimedia.
- E11, G01 – Analizar los relatos audiovisuales tanto lineales como no lineales.
- E11, G01 – Comprender los mecanismos básicos de los discursos audiovisuales y sus procedimientos de construcción a través de la asimilación de los recursos expresivos y narrativos que los hacen posibles.

1.4.2 Diseño 3D

This subject provided the right base to learn 3D modelling, how to start and how to continue, also this subject was one of the main start to get interest on 3D modelling. Since the making of the project that has to be done for that subject the need to learn more about 3D has been growing each day. Using 3DS max and the basic tools to modelling in that subject was the base of this project.

- E04 – Capacidad para el diseño y la creación de elementos gráficos.
- IR07 – Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
- R13 – Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los Sistemas de información, incluidos los basados en web.
- E04 – Analizar las características técnicas de las herramientas de diseño 3D.
- E04 – Usar aplicaciones de modelado tridimensional para videojuegos.
- E04, IR07, IR13 – Explicar los principios que permiten la definición de elementos gráficos tridimensionales.

1.4.3 Motores de Juegos

This subject was fundamental to understand how an engine works, how Unity works and also scripting.

- E08 – Capacidad para diseñar y construir modelos que representen la información necesaria para la creación y visualización de imágenes interactivas.
- E12 – Capacidad para evaluar, usar y extender motores de juegos.
- IR06 – Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
- IR07 – Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
- IR10 – Conocimiento de las características, funcionalidades y estructura de los Sistemas Operativos y diseñar e implementar aplicaciones basadas en sus servicios.
- IR14 – Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.
- E08, IR06, IR07 – Explicar los principios básicos de la creación de imágenes interactivas.
- E12, G09 – Evaluar las características técnicas de los motores de juegos como tecnología para la creación de videojuegos.
- E12, IR06, IR07 – Explicar el funcionamiento de un motor de juegos.
- E12, IR06, IR07 – Realizar extensiones y modificaciones sobre motores de juegos.
- E12, IR07, IR14 – Usar motores de juegos para la creación de videojuegos.

2. Game Design

2.1 History

This history tells the life of Zack Trenton. He was an old soldier in the Vietnam War. In 1978, he lived in Chicago, EEUU, with a paid house, a perfect family, economic stability and affective. He was decorated with honors after the war, he was a true National Hero. The CIA was involved in the cases of an experiment done in war. The CIA used this example to prove that the experiments done in Vietnam was nothing to worry about and telling the American society and the rest of the world that they did nothing wrong there. This experiments had a media impact around the globe.

Two years later, after what Zack lived in that war he did not remember anything of it. Something happened that last year that he started to remember some things of the war and the experiments that he went through. He was one of the patients in that famous MK-Ultra program done by CIA, also one of the top patients. Zack has been suffered amnesia because the overdosed of LSD that he received, till something change. All that memories came all together making Zack to suffered paranoid schizophrenia and post-traumatic stress. All the perfect live was falling apart, Zack started to drug himself and drink everyday at almost every hour, no sleeping either, he started to feel fear in his own house, the faces of his own family began to fade into horrible faces, demonic faces, also he started to hear some voices and hallucinations. The sound of the guns and the cries of suffering was increasing every day that passes through. That voices on his head, was called by Zack "Them".

One day, his wife Mary, chose to end it up that relationship, even that she loved him so much, she could not live like that anymore. She grabbed their son and she tried to leave the house. In that moment Zack hear the voice of "Them" and told him that he would lose his wife and his son, that they will not live in heaven without him.

"They" told Zac to sacrifice his wife and his son to enter the heaven, that he would go with God just when he kills them. Mary was killed by Zack, uses as a human sacrifice, but his son, when he was about to be killed by his father a white fog cover everything, the time stopped. Mary was found dead at the next day, no one knows anything about where Zac and his son Scott went, they just disappeared.

The media reported that the murder was done by thieves who broke inside the house, then kidnapped Zac and his son, and they are missing. The CIA bought the media, they could not afford two more dead on their background, they made every link between them and Zack's family disappear.

This history uses as a base a narrative element called MacGuffin, the history will not tell what happened to Zac and his son, and what "They" are really are, if they are real or just voices on his head, the history will develop around that unknown elements.

2.2 Levels/environment design

The environment structure is composed of two halls and one lobby. This environment is looped, in each iteration the player has to find, explore, read, listen or watch something like pictures, notes, video, audio to know more about the history and open the door to the next iteration. Each iteration changes the environment in different ways, illumination, audio, effects, etc... more abrupt and scary in each level. All the game is placed inside Zack's head, he has his own version of the night of the murder which haunts Zack head constantly. He does not remember anything; he is constantly trying to understand what happened that night because after that he suffered post traumatic amnesia. So as it was said, the environment is distributed in a big loop composed of iterations, the history will be dosed and will be explain a bit of it in each one.

2.3 Gameplay

The gameplay will be based in investigate and find, then some history will be dosed as reward. Finding all the pictures that are hidden around the environment will unlock some part of the history, this history is like a puzzle that has to be completed by each player.

Also there will be notes in the environment that will tell the basic history, and picking up this will open the door to the next iteration. Also it is possible to unlock the next level by watching the TV or listening the voices in the phone.

Also Mary will appear in the environment and it is necessary to not get caught by her.

2.4 Sounds

The experience has to have sounds that will make easier to the player to involve into the game. Like the sound of raining, of a phone and an environmental sound not so strong that will make the player more easily to scared. Also Mary will have a sound of crying or laughing, it depends of the iteration of the loop. There will be also sounds of wood cracking, doors opening and closing, lamp moving, knocking...

3. Art creation and development

In this chapter is going to be explained how the different tasks that compound the game has been created. This chapter its going to be divided in different steps. First it is going to be explained how the artistic elements have been made (modelling, texturizing, rigging, animation...), second how the the mechanics of the game has been developed and finally how the post effects have added to the experience.

3.1 Creating the artistic elements

All the artistic field has been done handmade, all assets has been designed, modeled, texturized, rigged and animated by the creator of this project. There are no assets downloaded by internet and placed inside this Final Degree Project. It is a completely original project.

The artistic elements that gives form this project, mostly 3d models, can be divided in two sectors:

- Inorganic Models (Assets and environment)
- Organic Models (Characters, animals, insects...)

3.1.1 Inorganic Models

It is going to be explained first the inorganic models because they are the majority in this project. Each inorganic model goes through for a four steps process

3.1.1.1 References and Concept Art

A search is made in seek of references, if it is not possible to find any references that fit in the game for that object, a concept art is made by hand with the elements that is going to have. This images have to be appropriated to the general artistic style, an exhaustive search is made till the right images are found. The references have to fit the time where the project is framed and also the interior design. Once the images have been selected, before going to the next step, the photos can be modified adding some elements or taking away some.

3.1.1.2 3D modelling and UV's

Once the references images are chosen, it turns over to the 3d modelling program. In this case, the program which has been used to work with is Autodesk Maya 2017. The references images are placed into the viewport with basic primitives called Image Plane. This process is done this way because will make so much easier to work modeling the assets, the reference is constantly in the viewport and allows the artist to make a better and faster model than without any reference. At the time of modeling, there are some rules to take into account to avoid future problems of optimization when the model is finally into the engine.

First, the model has to be done with the less amount of object possible. If a model is done using a big amount of independent object and materials when it is imported into the engine the Batches grow up and the activity of the CPU will overload making a drop of the frames per second (FPS) and therefore a bad experience for the player. It is recommended to make a single piece model with the less amount of materials possible.

Second, it is a must not use booleans operators when modeling, this kind of operations will destroy the mesh making the posterior edition, like extract the UV mapping or texturizing a really hard work.

Third, it has to have especial care to the double faced faces and the not merged vertex. The doubled faced faces will drop the performance significantly and the not merged vertex will not have continuity when texturing.

Finally, the normal of each face and vertex has to have special care. Sometimes when modeling, the normal can flip over and that part of the mesh will show as a black face even if it has a texture pasted to it. So it is important to check that every face and vertex has no black face around it, that will avoid future problems. If it is any doubt about how the normal is oriented there is a tool called Vertex Normal Edit Tool in Maya, that will show every normal in the model.

There are many rules when modeling to take care of, but those described before are the most important and basic. With that in mind, the game that is going to be developed will be optimized and will not give further problems.

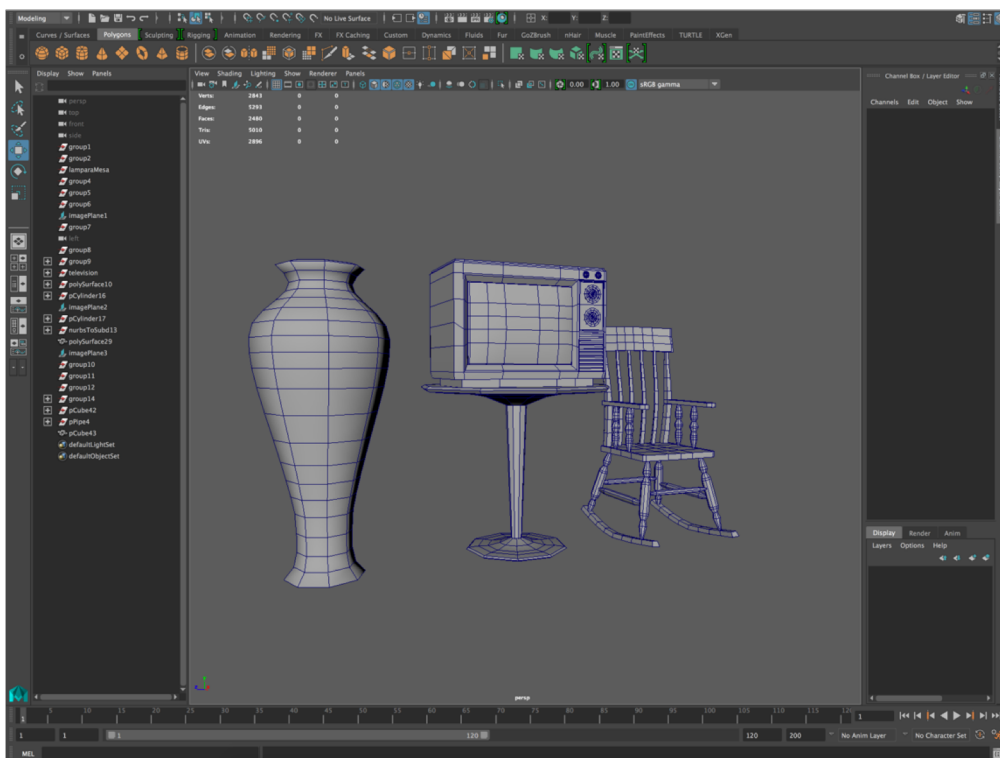


Fig. 0 – Modeled Assets

Once the model is completely modeled, the UV map has to be created. This will allow the artist to paste the texture over the model. To extract this map coordinates, it is necessary a plugin named UV editor that comes by default in Maya. In this texture space coordinates the object has to be distributed, this space has to be divided according the size of each sub-object that compounds this. The bigger the object is, the bigger the space that it will fill in the space texture. Also the objects that are not visible will have a small amount of space. It is fundamental to take advantage of this space, less free space, the more detail the object will have when texturing.

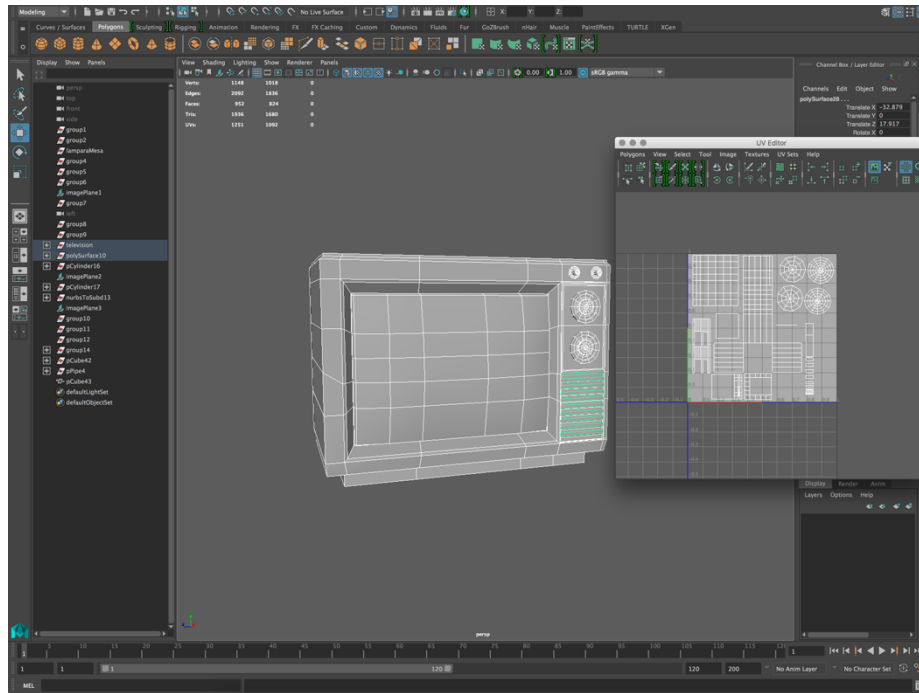


Fig. 1 – UV's asset

Right now, all the models share the same material information, all the sub-objects that create the asset has the same material. So it is necessary to group all the objects with the same material and apply a new one for each type. For example, if an object has wood and plastic, it has to be organized all the object in two categories, then apply a new material with a distinctive name for each group

Once the model has all the material assigned and the UV's done, it is necessary to import it in a valid file to be read by other future programs.

3.1.1.3 Texturizing

The next step is texturizing the model to provide it with more realism and detail. This task is done with a program called Allegoritm Substance Painter. This program provides numerous ways to paint and improve the model with a simple interface. Once the asset is completed and exported in FBX or other export file, it has to be loaded into Substance Painter.

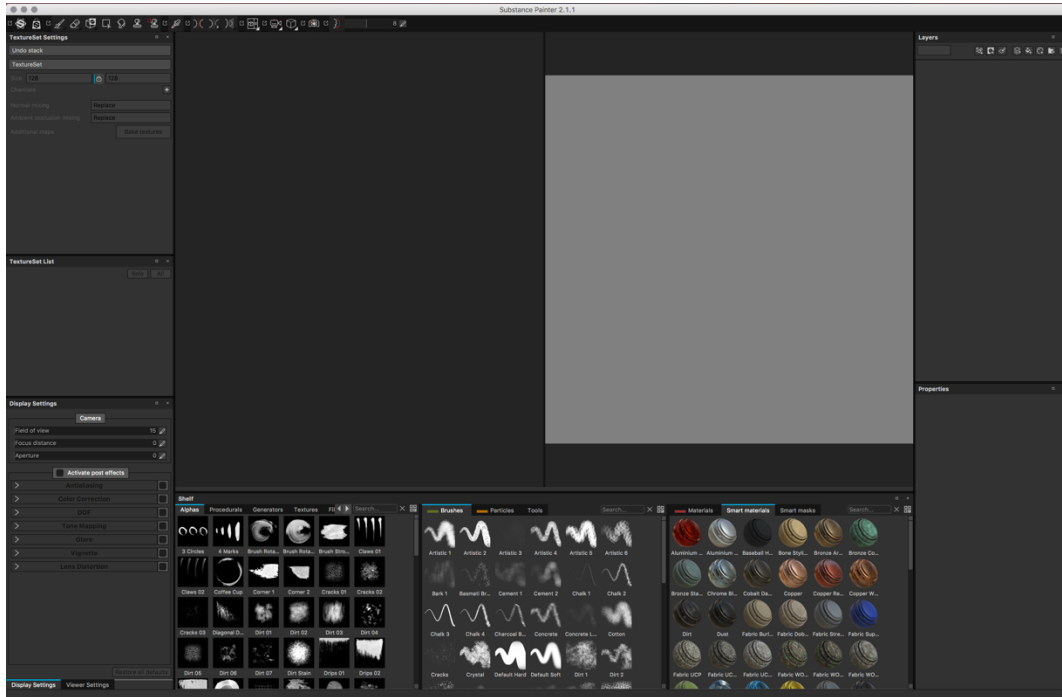


Fig. 2 – Substance Painter HUD

Firstly, it is necessary to create a new project, this can be done by loading the finished model into the program. Next, the model is loaded and prepared to be painted. In one of the viewports the model can be pre-visualized and how it is going to look when its loaded into Unity. Substance Painter provides a large number of materials (wood, metal, concrete, plastic, aluminum...) each one has different variations, and each variation has a huge number of parameters that it can be changed to provide almost a uniqueness material. If there is not a material that can fit into the model, there is a huge data base on the main page of Substance Painter where it is almost impossible to not find the right material for the model.

Secondly, at the right part of the program there is a layer menu, where all the materials can be dragged and therefore to give color to the object. It can be added many layer as the user wants. Each material is in one layer or different sub-layers. It can be added noise, dirt, particle erosion as as well. So, when the model is loaded, it has to be chosen the base material to work on, then it has to be adjusted all the parameters to give the right look to the model and finally add some kind of distortion to the model, like noise, dirt or erosion to give more realism and evading the not good looking perfection that some model has, so perfect that is not real.

Thirdly, the model is painted and texturized, it is necessary to export it and see how it looks in a game engine. At the left top of the screen there is an option called “export texture”, Substance painter provides many export options, one of them is exclusively for Unity 5. It exports three textures, Albedo for the texture (color, paint elements...), Metallic for the shine and Normal map for the details like cracks, wrinkles or any minor detail. Now, it has to be, one file with the model, and three textures each one with the information necessary to provide the asset with realism and details.



Fig. 3 – TV textured

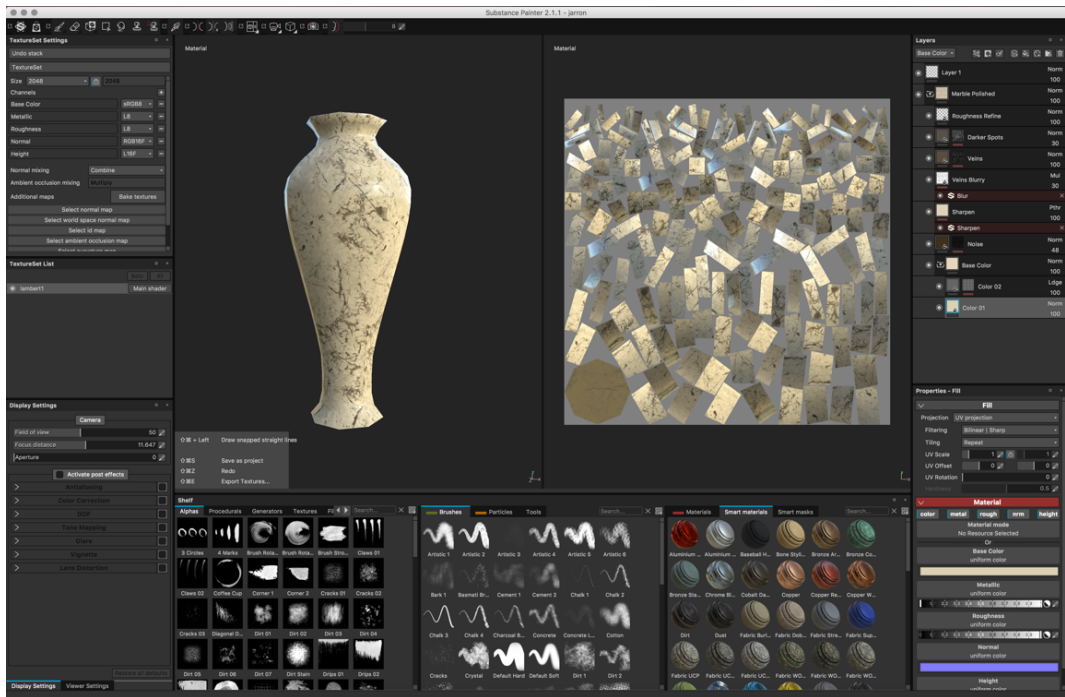


Fig. 4 – Vase textured

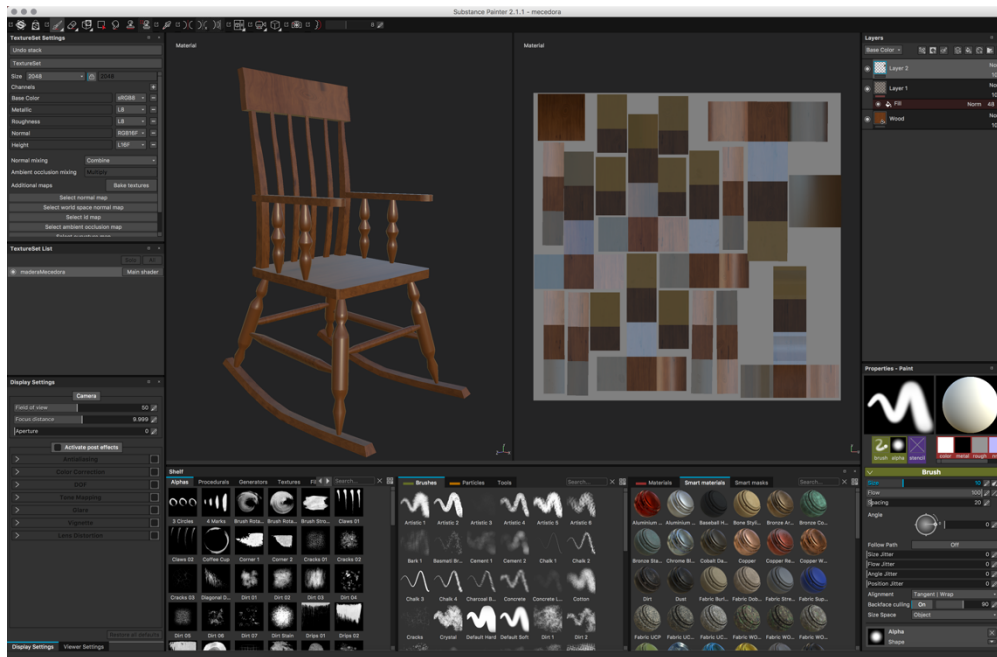


Fig. 5 – Rocking Chair Texturized

3.1.1.4 Model Export

Pick and drag all this files into Unity, place the model into the hierarchy and select the empty material that belongs to the model. Unfold the parameters and drag each texture into the box with the same name, albedo, metallic and normal map.

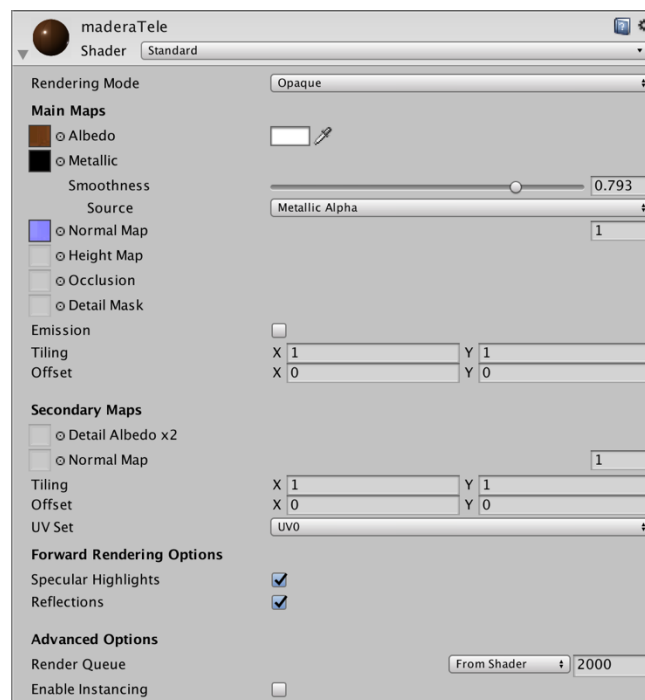


Fig. 6 – Material with maps in Unity

Finally, the model is in the game engine, now it is time to check how it looks with the illumination in the scene. It is important to check how the model integrates with the rest of the models in the environment and if it is necessary to change the albedo in Photoshop. Even if it changed in Photoshop the texture and the model still not working visually, it has to redo all the process, because one model can destroy all the visual impact



Fig. 7 – Rocking chair and Vase in Unity



Fig. 8 – TV in Unity



Fig. 9 – Lamp in Unity

All the assets on this project has been done with this workflow, also the environment. Fig(X)

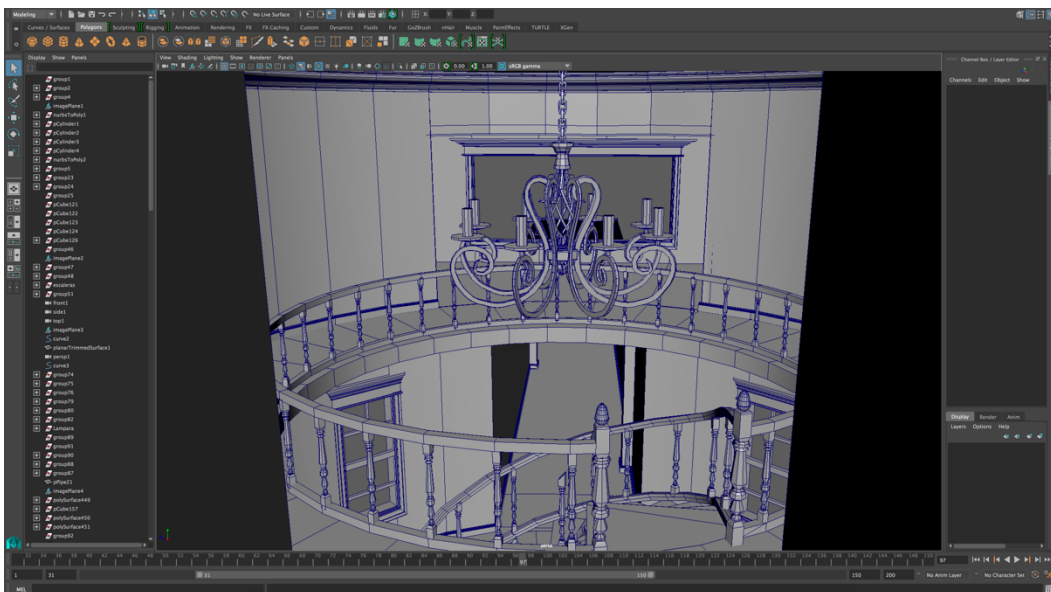
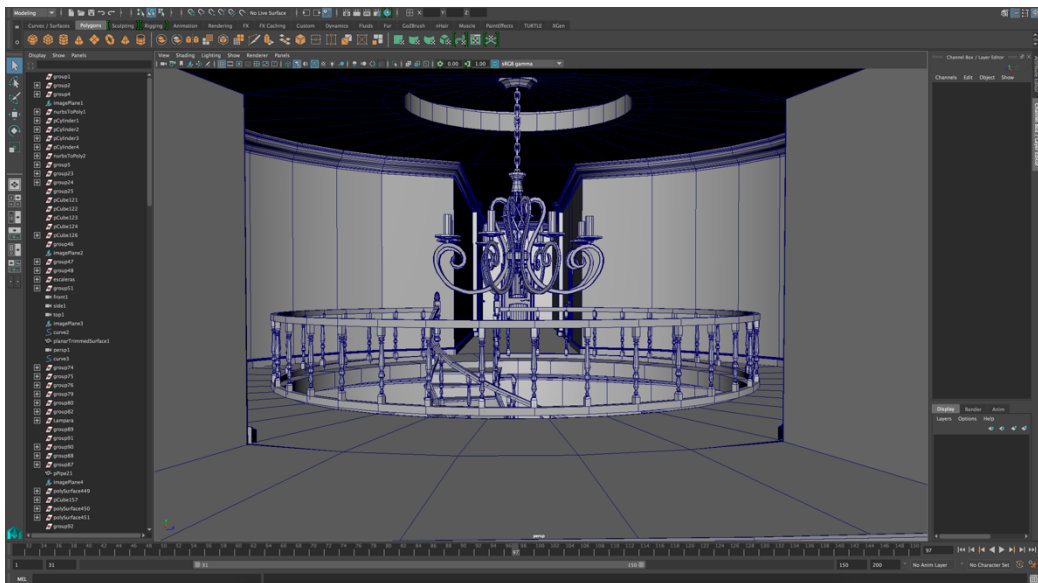
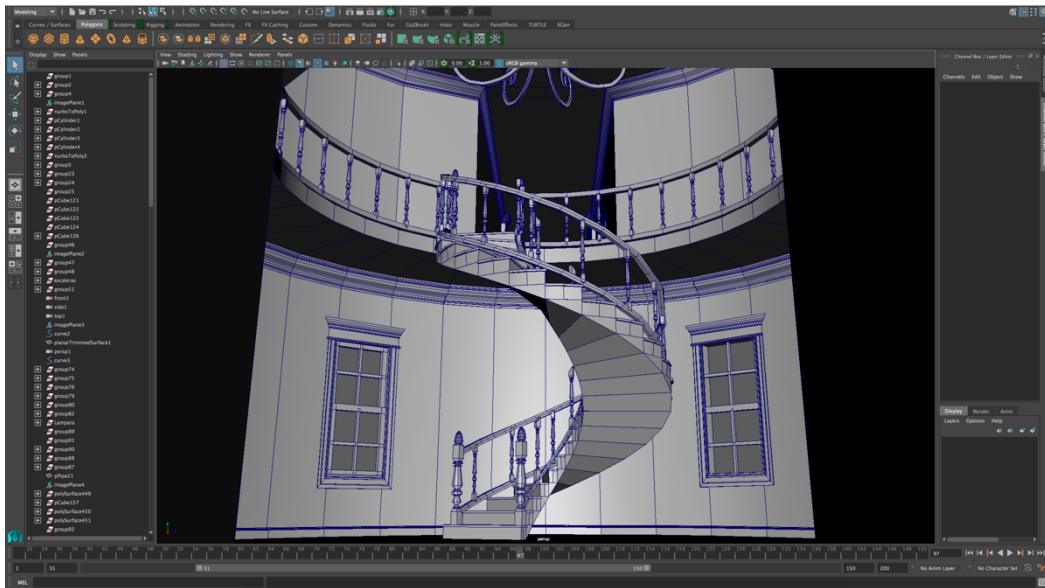


Fig. 10 –Empty environment in Maya



Fig. 11 –Empty environment texturized in Unity (1)



Fig. 12 –Empty environment texturized in Unity (2)

3.1.2 Organic Models

3.1.2.1 Zbrush and starting a new project

Now it is going to be explained how the organic model was done, the references, the phases of sculpting, texturizing and the phases to export the model.

The brainstorm of the history was done and it came to the conclusion that the enemy must be a tormented woman that was killed by her husband. The woman will haunt the main character through all the game. Then a concept art was done with all the ideas and details get for the brainstorm, face, body, objects... The concept art must be clear and easy to interpret for the modeler, if not the sculpt will not be perfect.

The program used to sculpt is Pixologic Zbrush, is industry's standard digital sculpting application. Sculpt with virtual clay, texturing, modeling, etc. All in real time.

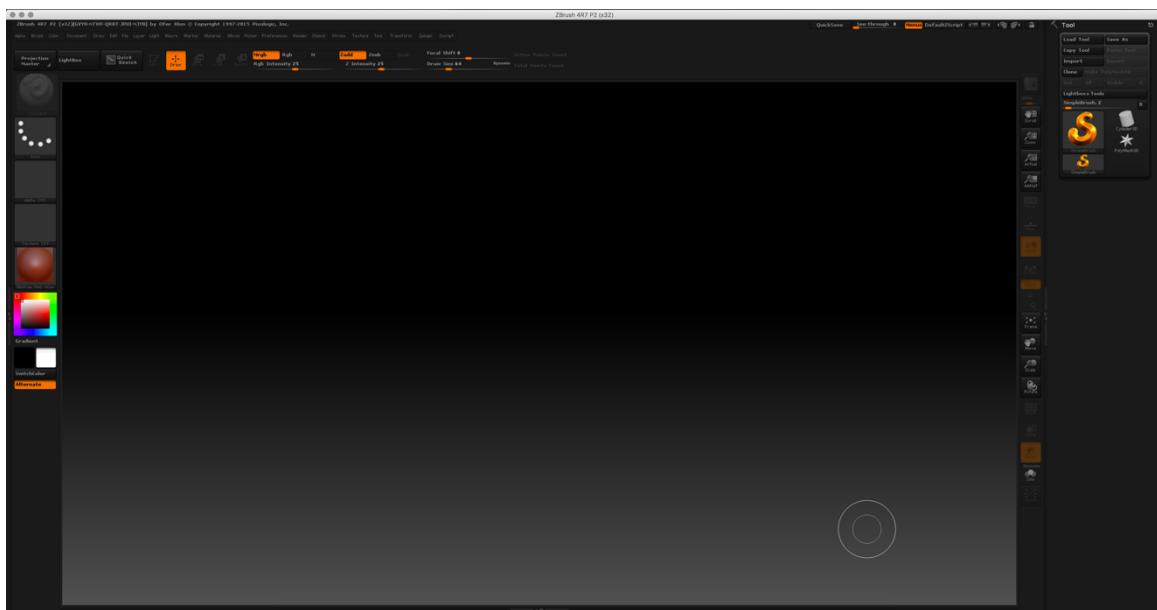


Fig. 13 – Zbrush Interface

The process of sculpting has many phases, all of them are really important for the optimization, realism and perfection of the model. Each of them has to be done knowing the program and the topology of the mesh to avoid future problems.

Zbrush at the beginning can intimidate, there are many buttons, options, parameters and menus. Some artists start sculpting with a base model, fig. 14, generally a body structure where it is brushed over it and deform the base to obtain other model different. It is a good resource when the artist has no time to expend in create a base model, but if the artist bases all his work on predefined models, it will be impossible to understand anatomy and volume. Other artist starts with a basic primitive, fig. 15, generally a sphere, and then the artist starts to sculpt and adding more sphere to create the full body. It is a good way to learn anatomy and volume, but it takes too much time to create a full body. In this case the organic model was done like the second example, from 0, every time that a model is created something new about volume, anatomy and something new about Zbrush is learned, so it is always recommended to create a sculpt from a primitive.

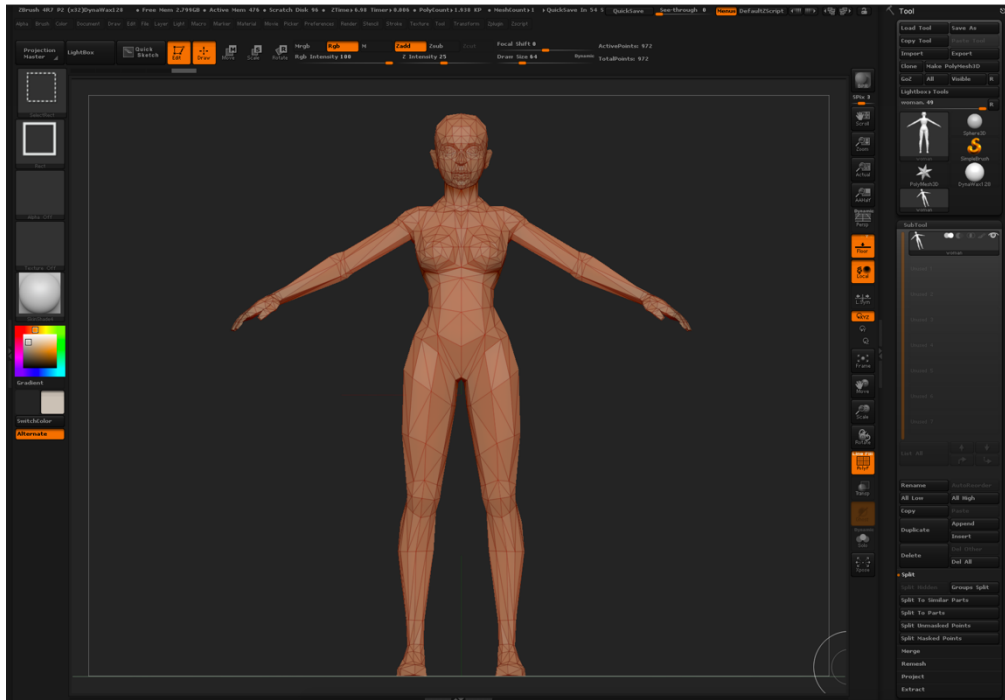


Fig. 14 – Project Started with a predefined female model

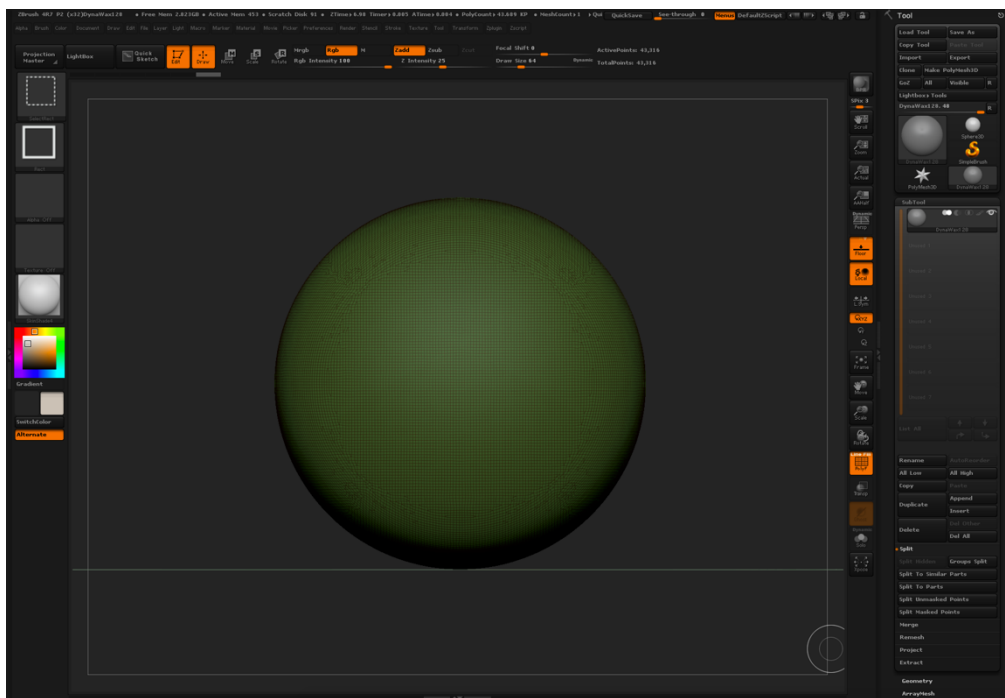


Fig. 15 – Project started with a simple sphere

Once the way to sculpt the model is chosen it is time to load all the image references that it is going to be used to sculpt the organic model. There is a plugin in Zbrush called Spotlight, it allows the user to place, size, modify, etc, all the images into an upper layer to use them as references when the artist is sculpting. It is a must to use references images when sculpting, it is possible to not use references images but it will take double time and the result will not be as perfect as other with references images. When the primitive and the references images are loaded, it is time to start sculpting.

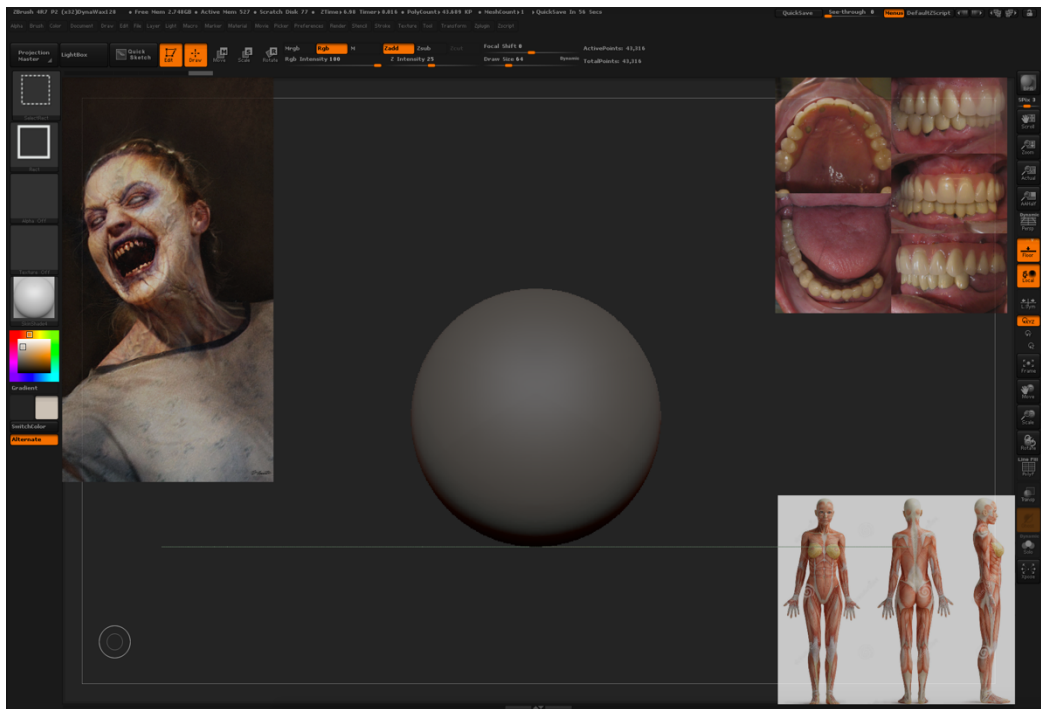


Fig. 16 – All references images loaded into Zbrush

It is necessary to understand how the polygons can be distributed, how the topology is formed in a mesh, there are two types of topology. One topology created for sculpting and detailing, which is really heavy and the polygons are distributed without any rules, called in Zbrush Dynamesh; and other topology perfect for videogames and animation, this one use the polygons to take advantage of the space of the model, the zones more visible and with more detail occupy more polygons and the zones more hidden areas or without any detail use less polygons. If the model is going to be animated it is important to distribute all the polygons of the face properly, creating edge loops around eyes and mouth, this will improve the realism, the mesh will deform like a face will do and the ability to transmit feelings to the player.

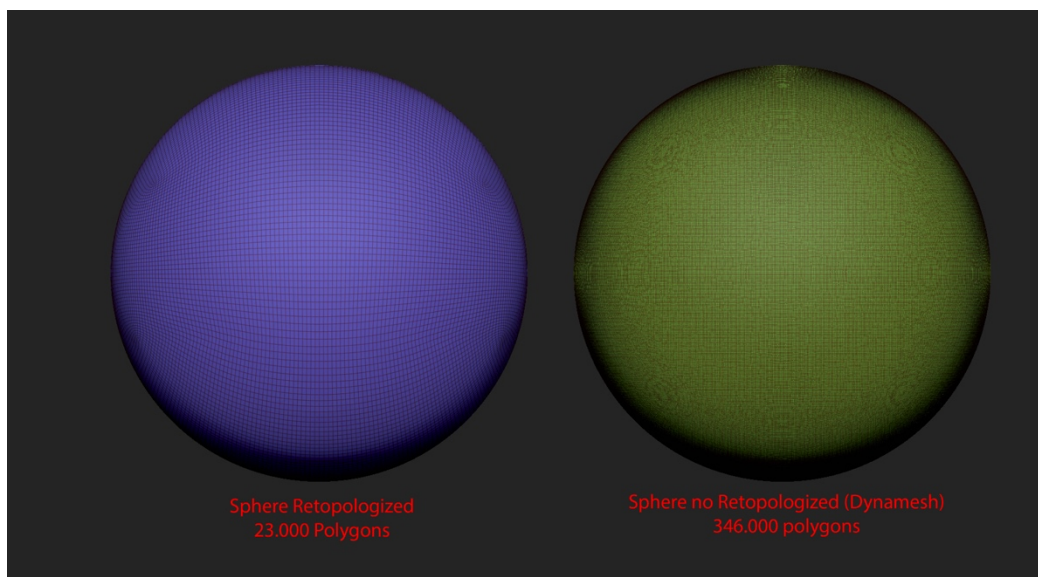


Fig. 17 – Differences between topology

3.1.2.2 Mary's face creation

At the beginning, it is necessary to work with the sphere roughly, making the basic form of a face with the brushes provided by Zbrush. At this early stage of sculpting just with three brushes it is possible to obtain the basic form of a female face. Also the model in this stage has a low level of polygons, so it is impossible to give too much detail to it. Then is necessary to increase the number of polygons increasing the number of Dynamesh resolution and add more detail. When the second phase of detail is done it is necessary to add more resolution, this process has to be repeated until the model is detailed enough, fig. 18.

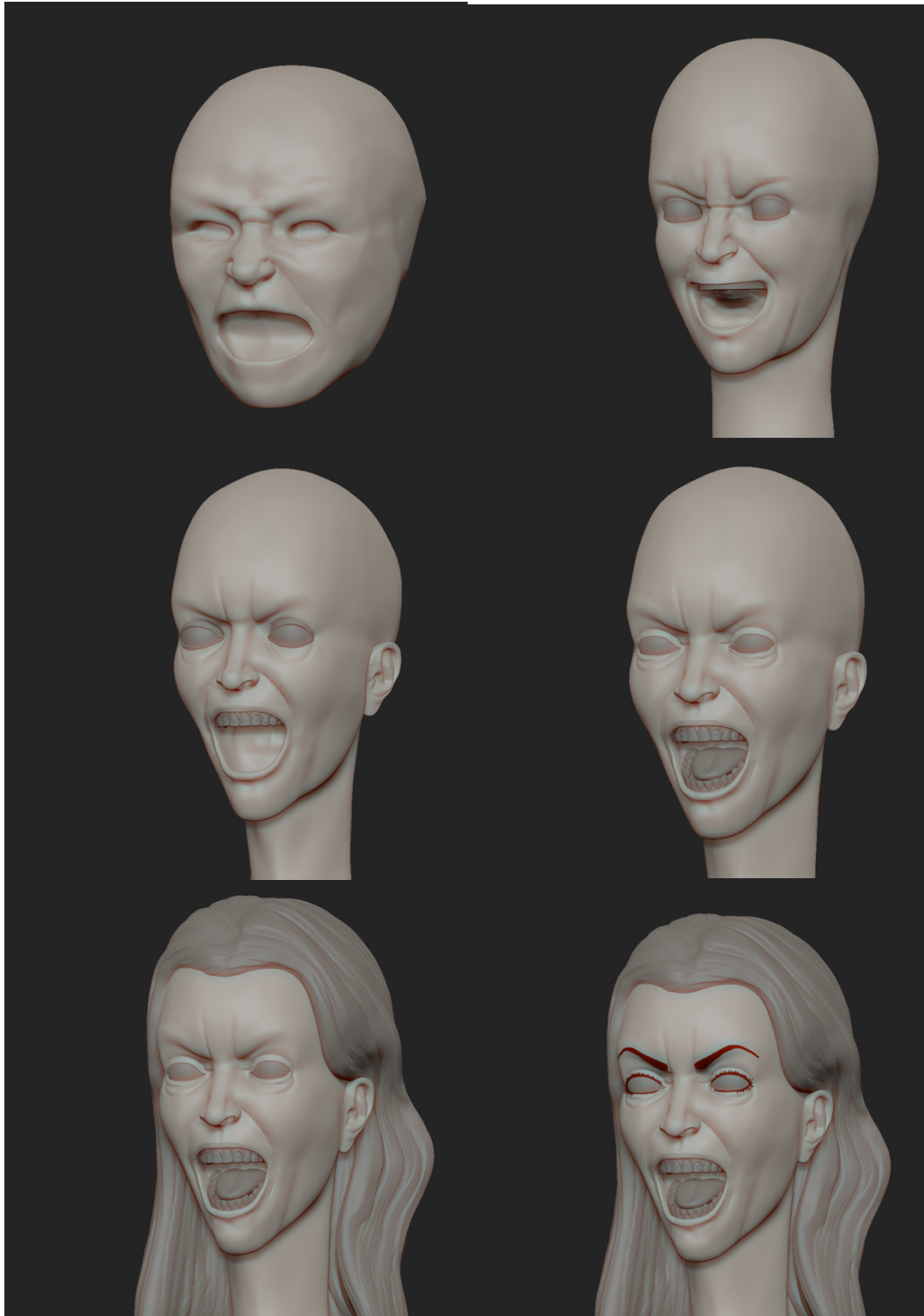


Fig. 18 – Modeling Mary's face. Steps

3.1.2.3 Mary's body creation

Once the face is done is time to start with the body, as it was told before, the body has been created with spheres, each one representing one part of the body, fig. 19. Then is necessary to change the references images to sculpt the body. There are many parts of the human body hard to sculpt without any reference (hands, foot, back...). It is not completely necessary to sculpt it following any order but it is recommended to start from the top of the model to the bottom.

When the face and the body is done, now its time to add the clothes and details that the character is going to wear.

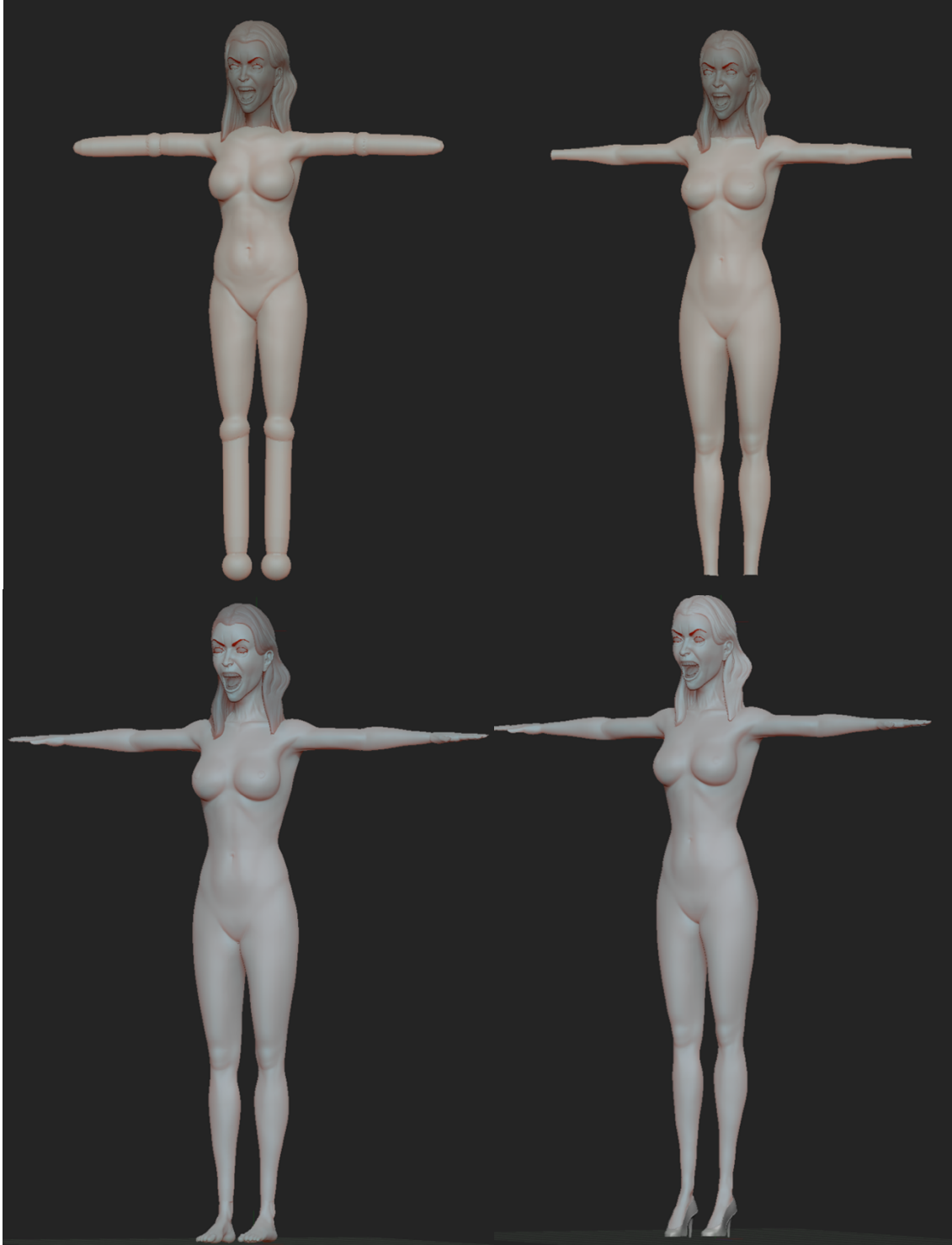


Fig. 19 – Mary's body modeling. Steps.

3.1.2.4 Marvelous Designer and clothes modeling

The clothes have been done with a program called Marvelous Designer. This program allows the user to create 3D virtual clothing. Marvelous Designer is the most used program in this field. The artist can create clothes as the traditional way, with patterns, needles, mannequins and seams. In the figure 21, it is possible to observe how the interface of Marvelous Designer is, at the left is the zone where the mannequin wears the clothes and where the clothes physics are simulated, at the right is where the artist draws the pattern and where sews the garment. In order to create the dress that the enemy is going to wear, it is necessary to create first the pattern of the dress, then get the references of the concept art done at the beginning and then draw the pattern on the left zone of Marvelous Designer. Then it is time to sew the pieces of the patterns, there is a tool called “Sew Machine” that allows the user to merge two or more piece of garment together. Once the patterns are drawn and sew together it is time to simulate the garment with the avatar loaded in the scene. Next it is exported to Zbrush and added to the original model.

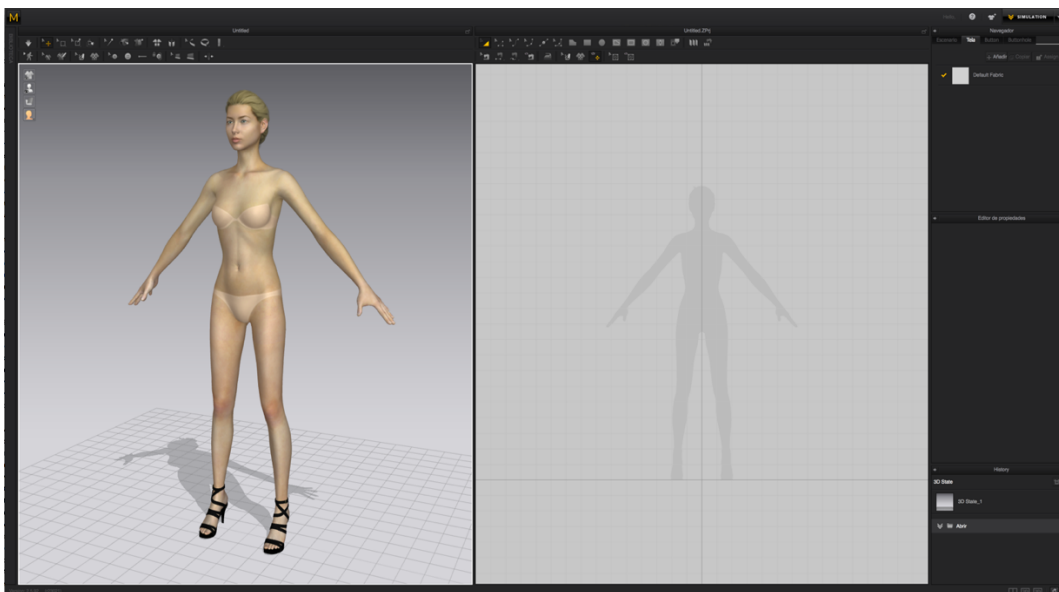


Fig. 20 – Marvelous Designer Interface

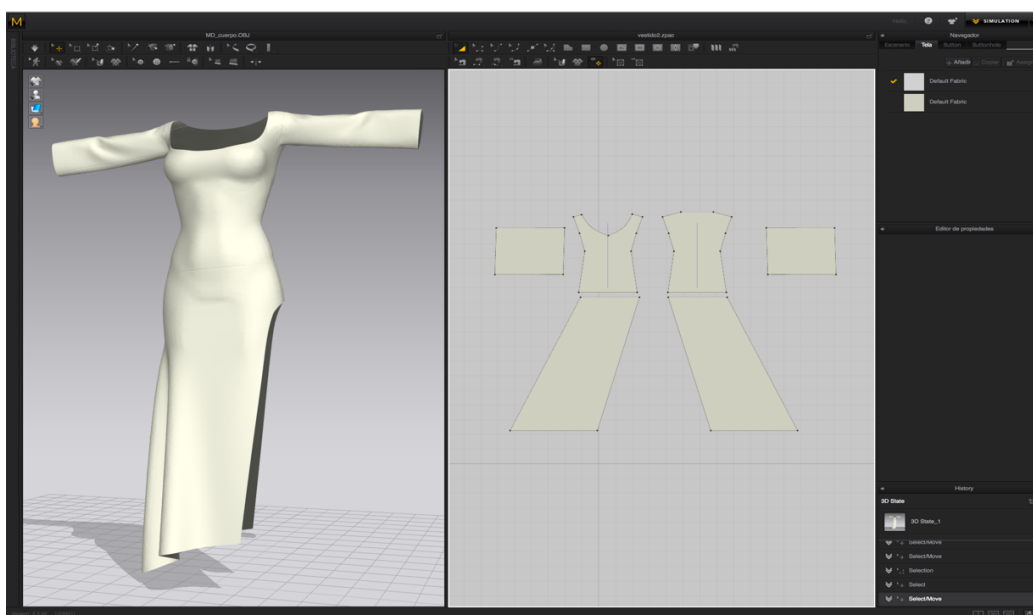


Fig. 21 – Garment finished



Fig. 22 – Completed model in Zbrush

3.1.2.5 From high poly to low poly

The model is completely done, it is composed by different subtools. This group of subtools has 7.14 millions of polygons which means that is impossible to export the model with that resolution to an engine. The model right now has a topology perfect for modeling but not for videogames or animation. This model is right now in high resolution, it is fundamental to reduce the number of polygons and try to keep the maximum details possible. With a good distribution of the polygons, UV's and maps, both models can be almost the same visually but with a large different of polygons between them, more optimized. This process is separate in different steps, retopology, project, making UV's, Map generation and Decimation Master. All this steps have to be done for each subtools that gives form to the model.

3.1.2.5.1 Retopology

As it was told before, the model has too many polygons, it is fundamental reduce it to be viable for videogames or animation. So retopology is the process to obtain an existing model with more optimal geometry. This process can be done in almost every 3D editing program, in this case the model is going to be retopologize in Zbrush. Some artist rather doing this process handmade, to get a perfect topology for facial animation or maximum optimization, polygon by polygon. In this case, it has been used a plug-in added in Zbrush, which automatize the process and its not necessary to model quad by quad to get a perfect result. Also one of the reason why this process was not done handmade is because the face of the model wasn't going to be animated. The process to retopology in Zbrush is done by a tool called Zremesher.

3.1.2.5.1.1 Zremesher

It's a tool that comes predefined with Zbrush, fig 23. It automatizes the retopology process, also it is possible to paint over the model with different colors to predefine the resolution that the mesh is going to have in each zone. In the figure 24, the model was not painted, and it is possible to observe the difference between topologies, it is possible to observe how many detail it is lost even when they have both the same amount of quads, 18k polygons. In the figure 24, the topology is visible and it can be seen how at the right of the picture the polygon's distribution has rules, edge loops in mouth and eyes and it is distributed accorded to a pattern. Instead, at the left of the figure, the topology has no order at all and the quads are distributed just the main reason of sculpting.

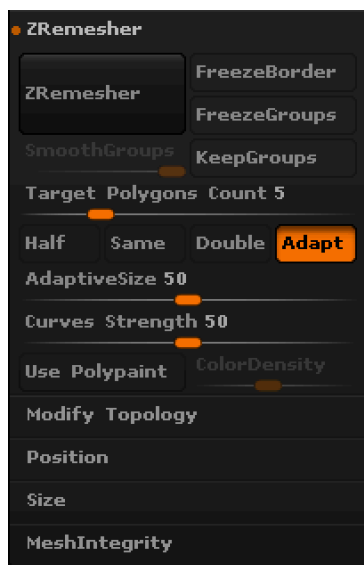


Fig. 23 – ZRemesher menu

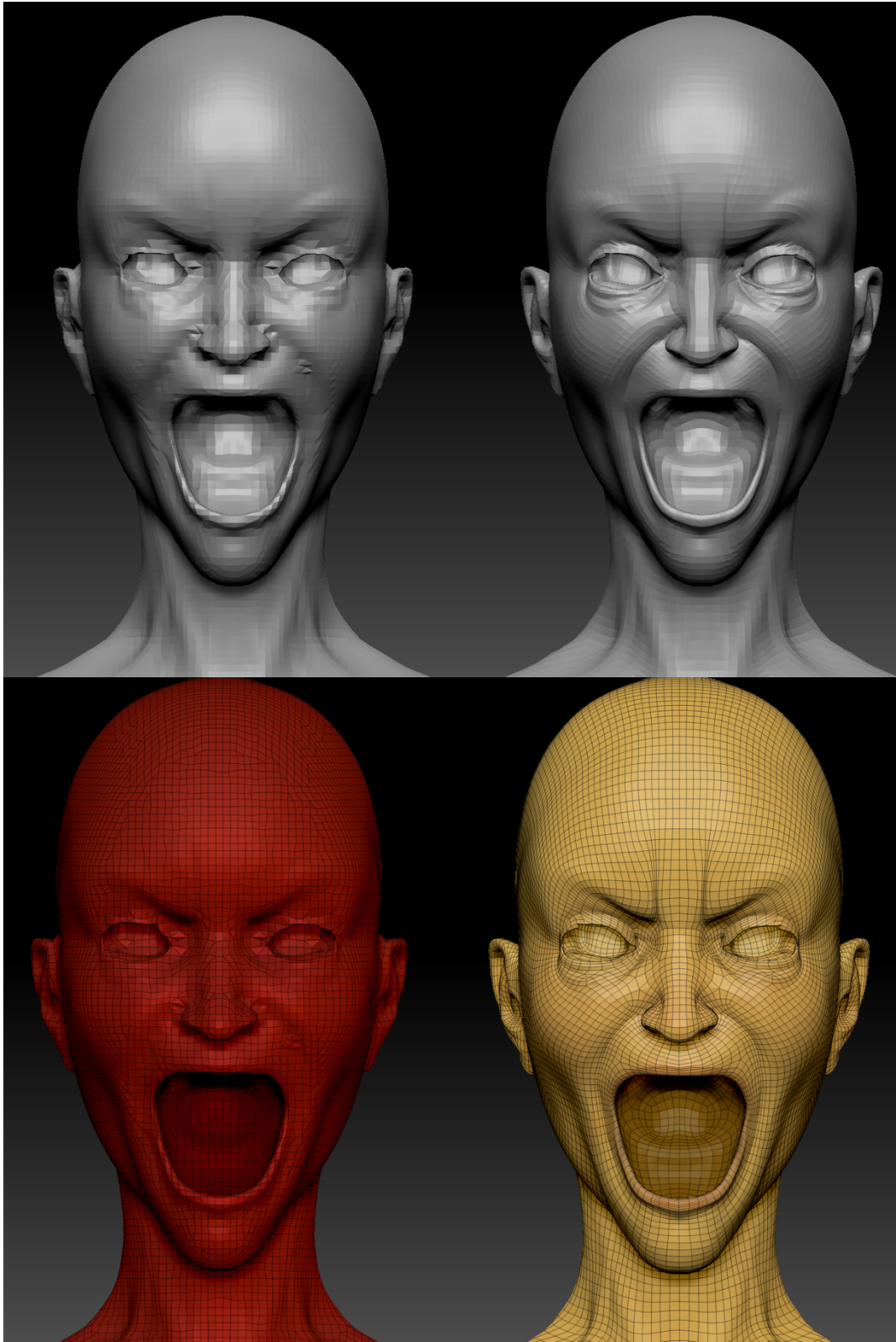


Fig. 24 – Retopology difference.

3.1.2.5.2 Projection

Next step in a sculpting workflow is projection. When the model is almost finished and ready to create minor details it is retopologized as was told before, then all the detail from the first model has to be transferred to the retopologized one. This is done projecting the low poly one into the high poly, the latter acts like mold where the less resolution model is projected. Once is projected, the retopologized model has to be subdivided and projected again. This process has to be repeated until the optimized model has the same amount of details than the first one. In the figure 25, it is visible how the low poly model (grey) is selected and inside of it, it is the high poly model (yellow) ready to provide the detail to the low resolution model.



Fig. 25 – Projection

3.1.2.5.3 UV Mapping

Once the model is with the right topology and with the right amount of detail, it is time to prepare the model to add the maps that will give details in the low poly model. But first it is fundamental indicate to the model where is going to start the maps and how is going to be pasted. Imagine that the mesh has a zipper, this allows the model to be completely flat and capable of being projected into the maps. This technique is called UV mapping. This is also done in Zbrush by a plugin called UV master, this process allows the user to select where the model is going to be opened, where the seams is going open the model and what parts of the model are forbidden to be opened, for example the face. First of all, the program creates a low poly clone to work with,

making easier select the zipper. This information is provided by the user to Zbrush painting the model with different colors, in red the zone of the mesh where is impossible to contain the zipper, and blue where the program has to create the seam, fig. 26. In some especial cases, it is fundamental to understand how the mesh is going to be opened and how the space texture is going to be organized for a better quality of the maps and optimization. When all of this is done then, the “Unwrap” button has to be clicked, it calculates all necessary to open the model and to create the UV’s, fig. 27. Finally, the UV’s are copied in the clone and pasted into the original model.

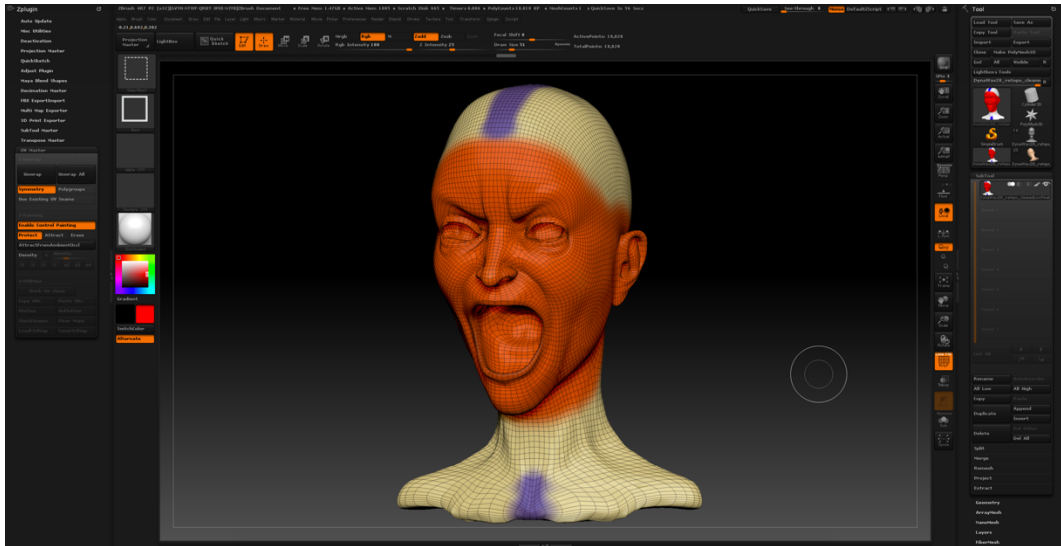


Fig. 26 – UV master calculating seam with paint.

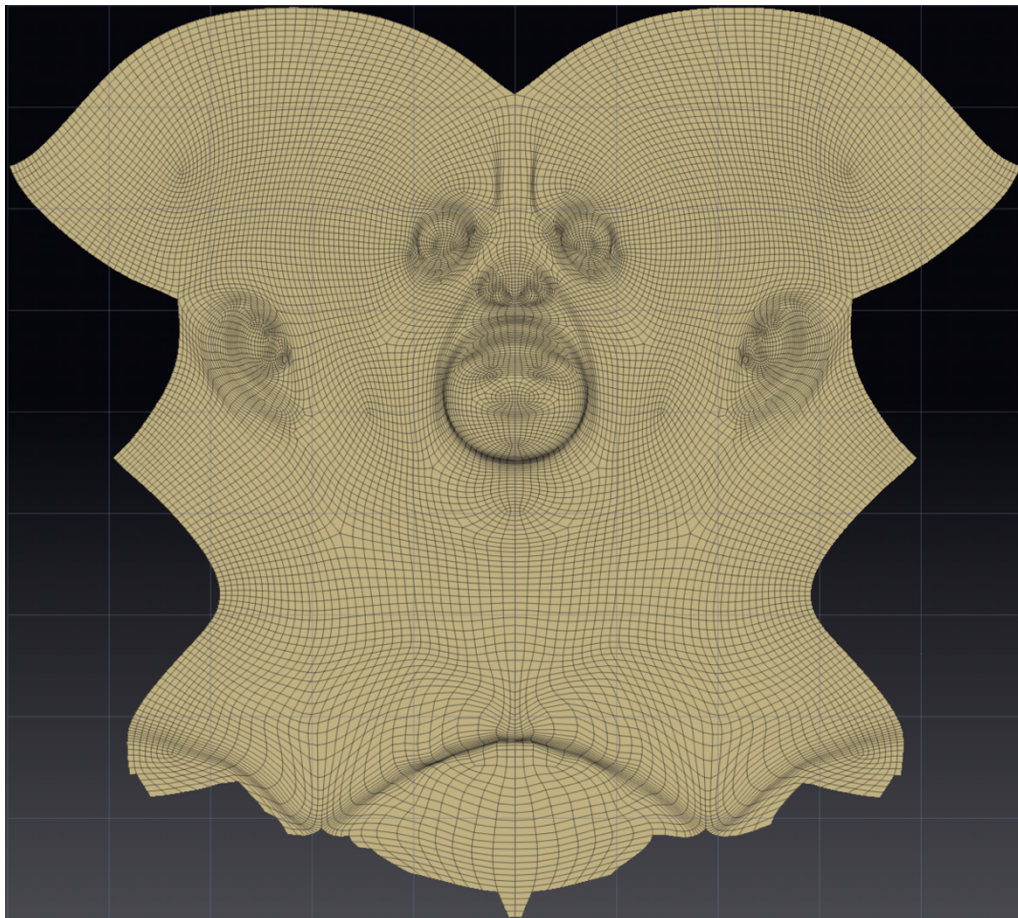


Fig. 27 – Opened model

3.1.2.5.4 Maps

The low poly model is prepared to being pasted with different maps. It is possible to create and paste many maps into the model, texture map, displacement map, normal map, vector displacement map... All of them are possible to be obtained in Zbrush, in this case only the normal map is going to be obtained by this way. The rest of the maps like texture map or metallic map are going to be obtained by Substance Painter, which will give a better and easier result to obtain than Zbrush. To export the model into the engine it is only necessary three maps, texture map (or Albedo in Unity), normal map and metallic. The normal map in Zbrush is obtained with by the higher resolution of the model. First, the model has to be in low resolution then Zbrush is going to increase the resolution until the maximum and generating detail into the map. The texture is assigned automatically.



Fig. 28 – Texture Map or Albedo Map



Fig. 29 – Normal Map

3.1.2.5.5 Decimation Master

Decimation Master is a plugin from Zbrush which with a high poly model makes an approximation model with much less polygons trying to keep the most details from the original. This plugin uses simplification algorithms. This process is used when the model is modeled completely, the retopology and the UV's done. Once the model has been retopologized, if it has been done right, there will not be any problem with the engine, but if the optimization is one of the main concerns, all the models have to be simplified. In this case one of the objective of this project is the implementation of optimized models, so this has to be done. Decimation Master is a simplification tool. This tool will allow the artist to reduce the amount of polygon of her models without a minor lose of detail. Also it is possible to keep the UV's and the borders. The amount of detail to reduced can be selected in the menu of this plugin. In the figure 30, it is possible to observe the Decimation Master menu, also in the figure 31, it can be seen the difference between a model retopologized and another simplified with Decimation Master.

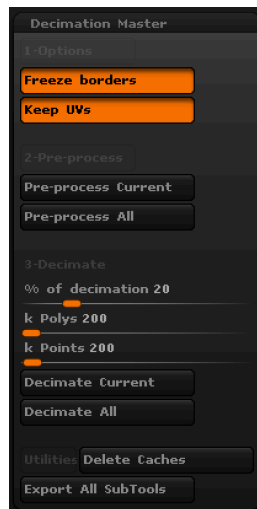


Fig. 30 – Decimation Master Menu

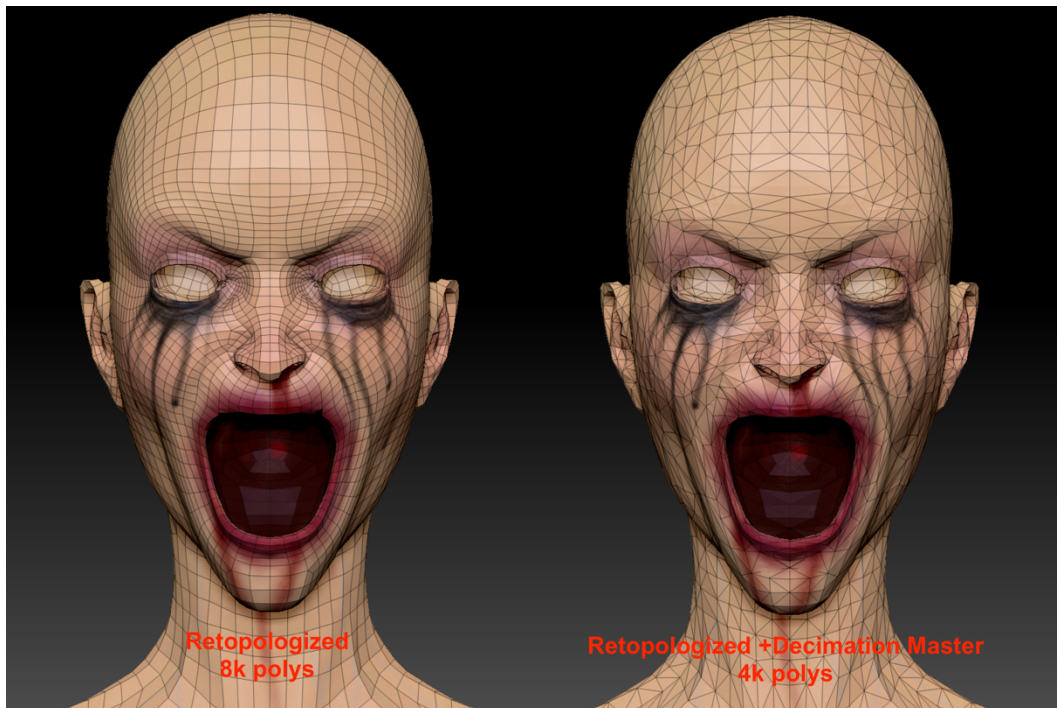


Fig. 31 – Simplification difference

3.1.2.6 Texturize

Once the model is in low poly (from 7.14 millions to 30k polygons), it has to be exported to paint it, this process is done by a tool explained before, called Substance Painter. This task can be made by Zbrush as well, but it is more recommendable to use Substance Painter since the results that is going to be obtained will be so much better and realistic with less time and effort. Each subtool that form the model, has a material assigned to them and an own space in the space coordinates. Once the model is loaded in Substance Painter it is time to paint it and add materials appropriate to each subtool. For the example, if the head is going to be painted it is crucial to choose a Skin material, modify the parameters and apply it correctly, to give realism and credibility, then it is added more paint details like lip contour, cheeks, nose, eyes... The process is similar with the rest of the subtools but choosing the right material in each case.

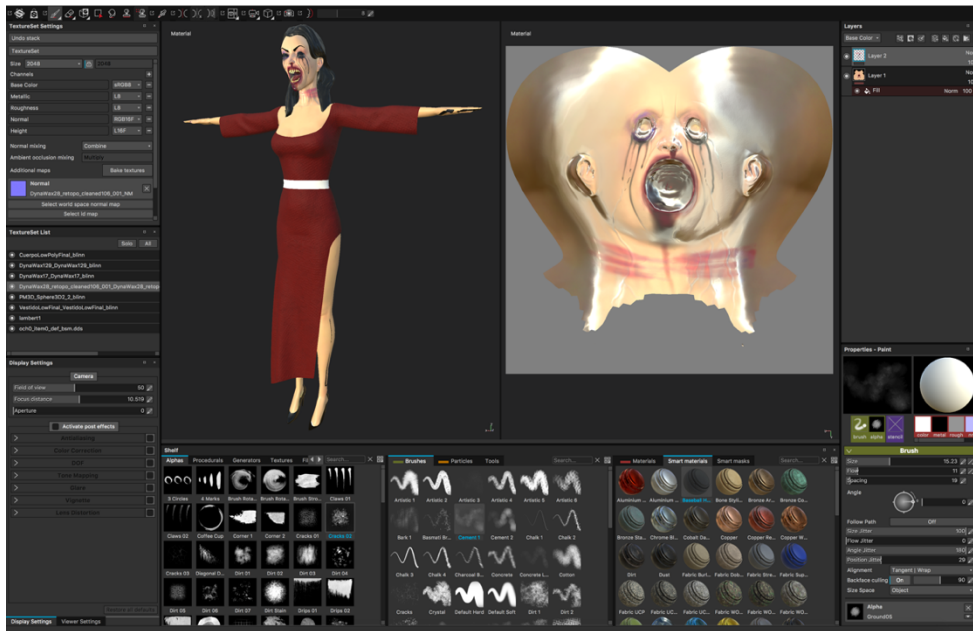


Fig. 32 – Mary in Substance Painter

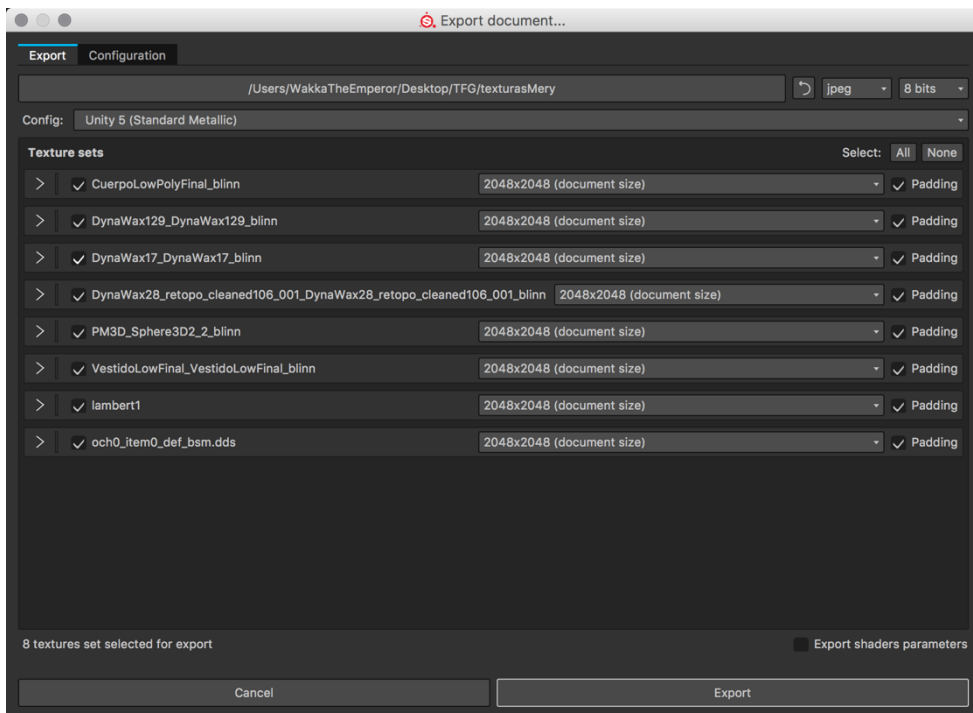


Fig. 33 – Export options



Fig. 34 – Mary's model finished

3.1.2.7 Rigging

The model is fully texturized but the model it is still in T pose, it needs some life, some movement, it is time to bring that essence to the game, one of the main source of fear here, comes from the animations. Before animate it, it is fundamental to create the bones that will conform the structure of the model, once the model is completely rigged, it will be possible to be animated. To rig the model, it is going to be subdivide in parts, legs, spine, arms, and head. Each part will have controls, this controls will be form by curves, generally circles, and will be attached to the joints, this can be done by parenting both or making constrains between them. This curves will allow the rigger to move the joints of the skeleton without touching the joints, making so much easier and satisfying. It is possible to rig without curves, but this makes the animation much difficult and not optimal, so it is recommended to use curves to rig the model.

Each curves take care of the rotation or movement of a joint or a group of joints. If the curve is transformed the joints associated to it, will move accord that transform. All curves are parented to the SuperMover, this means that if the SuperMover is displaced all joints of the scene will be moved or escalated. When the rig is completed, if a curve is moved the model doesn't follow yet, this occur because the weights are not done, it is time to skin the model.

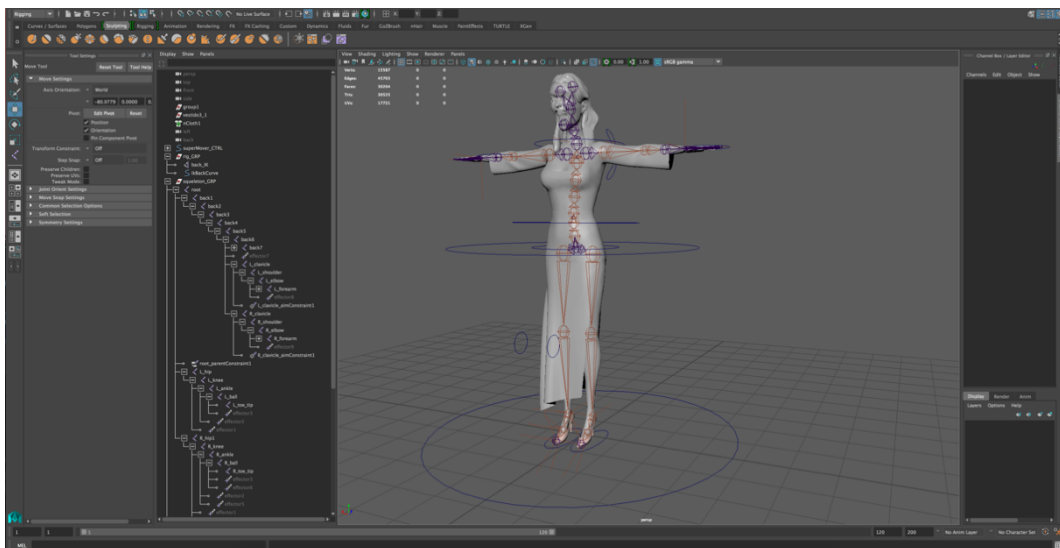


Fig. 35 – Completed rig and model

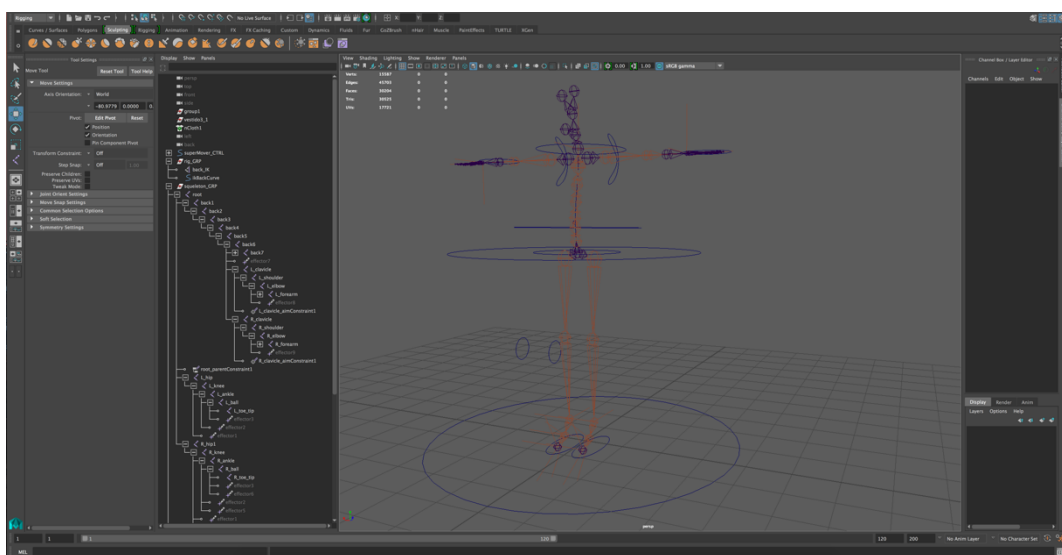


Fig. 36 – Rigging

3.1.2.8 Skinning

As it was told before, the rig right now it is completely separate from the model. So the objective is that the model move accords the rig. The skinning consists in bound the rig with the model making weights between each join and the part of the mesh near. This can be done automatically by Maya, but almost never this process works perfect, it is necessary to paint the weights in the model for each joint that compounds it. First, the model and the rig must be selected and then click on bind skin. The mesh is right now attached to the rig. Now it is time to check which joint needs more or less painting, selecting the curves and moving each makes possible to see where the skin destroys. Pressing Skin>Paint Skin Weights, with the mesh selected, will turn off the colors of the model by applying a different material to making easier paint over it. At the left of the screen an additional menu will appear, where it is possible to select any joint of the rig and see the force that exerts to the mesh. Also at the bottom of this menu, there are some option that modifies the brush, like adding or resting color, increasing the size of the brush... Every time that a part of the rig it is finished, it is fundamental to check if the mesh is bending correctly. When the mesh does not deform any more, it's time to start animate it.

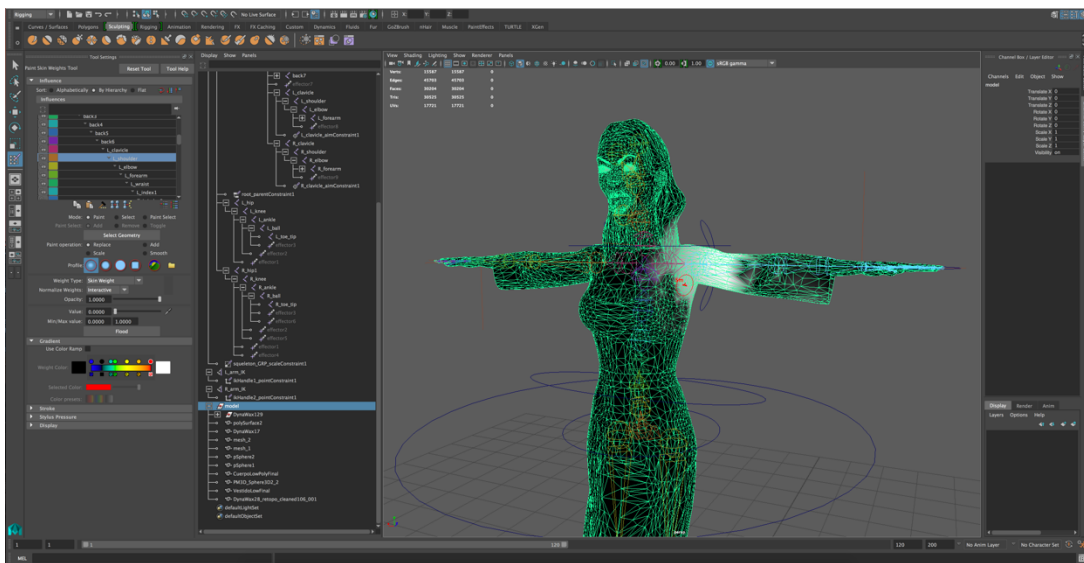


Fig. 37 – Painting weights

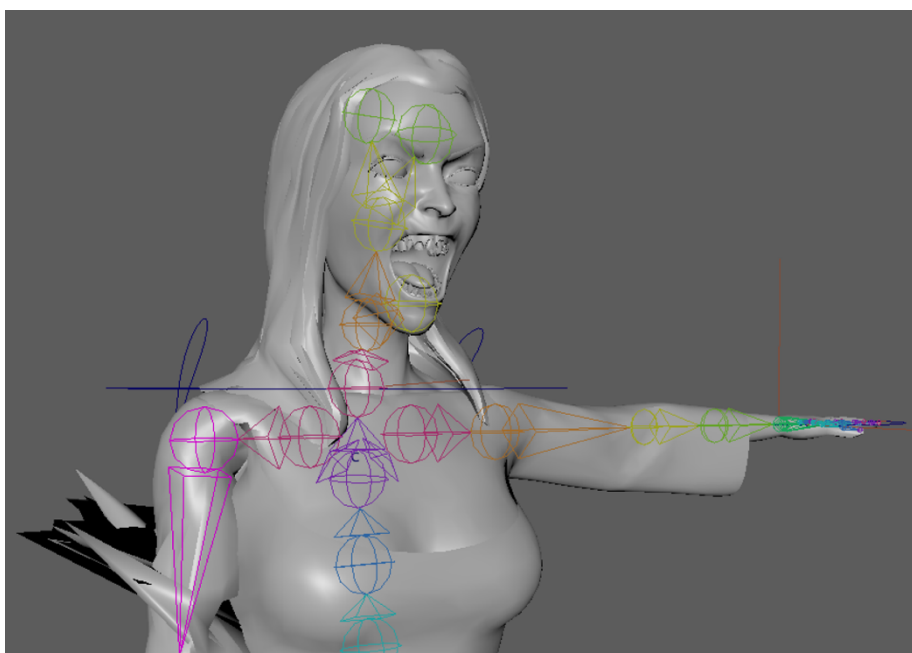


Fig. 38 – Skinning problem

3.1.2.9 Animation

The rigging and skinning are finally done. Now it is possible to animate the model, but first it is a must have a list of all the animations that the model is going to have, even the one that it may be utilized, because if an animation has to be added later, the next step (exporting) will have to be done again and a long time and effort will lose. The model starts in T pose, all the controls has to be initialized at this pose, because it is the initial pose. A pose is fixed in the timeline by keys, this keys save the information of one or multiple control in a frame. So the animation is a group of keys in a timeline. Animation for videogames has common animations for each character, idle, running, jumping, attacking... In this case each animation is going to have another version of each one, version with spams making a total of 9 animations. It is fundamental take care of the animation curves, and use it to loop the animation or check if the animation is being animated correctly.

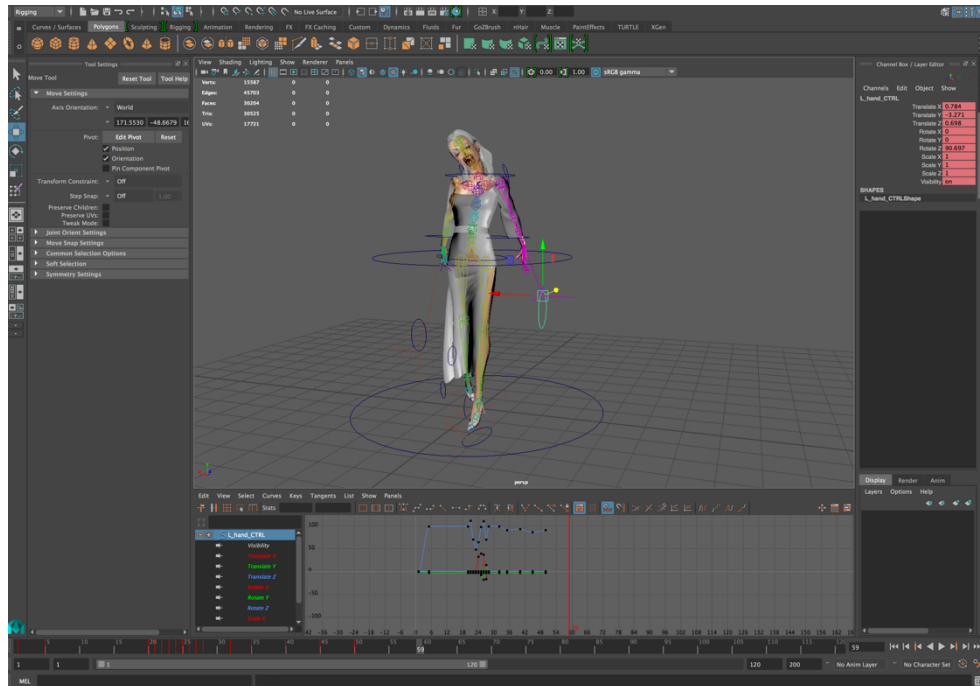


Fig. 39 – Animation process

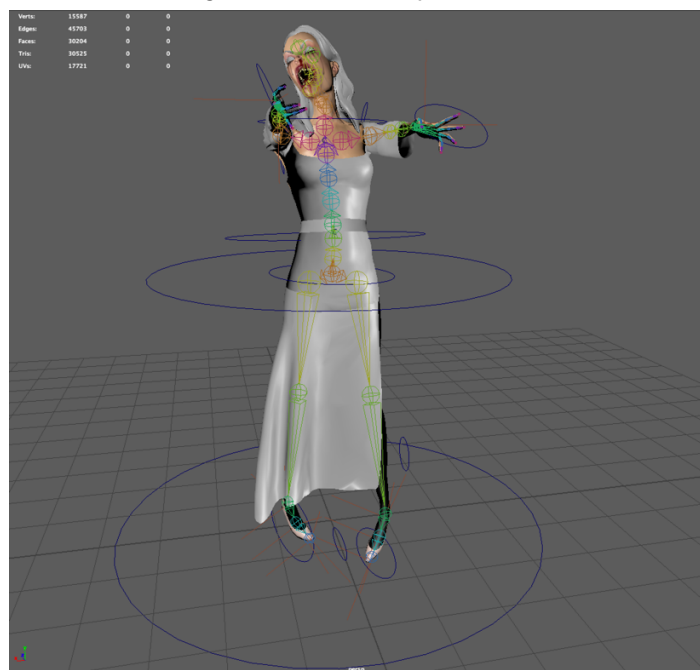


Fig. 40 – Attacking animation

3.1.2.10 Export organic model

The model is finally done, modeled, optimized, rigged and animated. Now it is time to export it from Maya to Unity. The export process is really delicate, first of all it has to be checked that every joint has no negative values on scale transform, check the right values into FBX exporter, like embed media, bake animation, set the start and the end of the baked animation correctly... In the figure 41, it is possible to observe the right options to export an animated model. When everything is checked and tested, click on export and finally the FBX will be exported. It is necessary to drag it into the unity project and select the prefab generated. All the animations are together in the same timeline, as it was done in Maya, so it has to be separated each one. The figure 42 shows how the animations menu looks like.

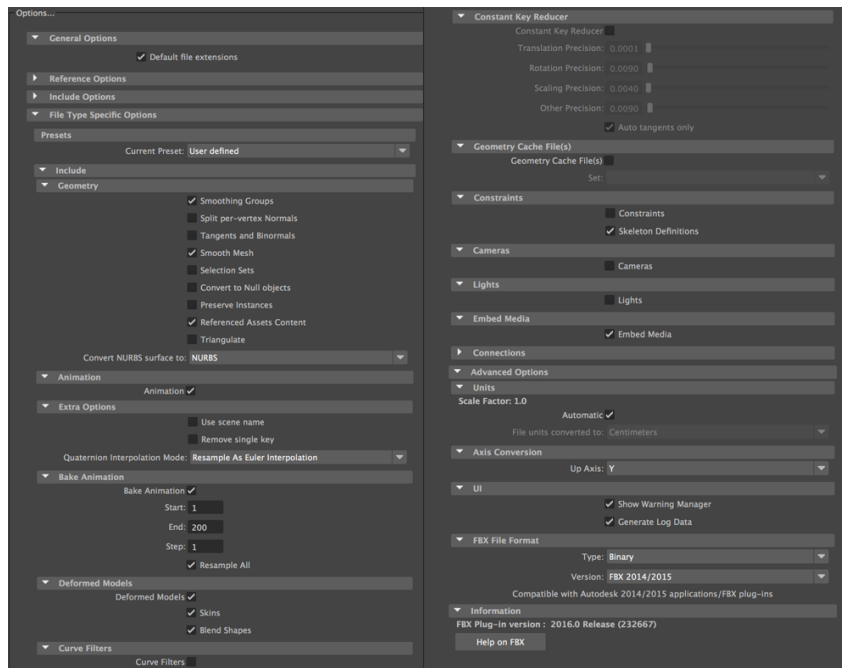


Fig. 41 – FBX export options

First of all, the animations have to be checked, that every animation is working fine, pressing play and analyzing it, the joints and the skin. If everything is working fine it is time to distribute the animations.

Second, it is moment to set the beginning and the end of the animation, then change the name and press plus. This will add the animation to the pile. This process has to be done for each animation of the model. When all the animations are done, the button apply is pressed and the changes done in the timeline will be saved in the prefab in unity. An animator file has been generated in that time.

Third, inside the prefab an animation file has been created for each animation that has been divide before. These animations have to be looped or not depending the use, checking the option loop in each file.

The animator file will allow the programmer to work with the animations when scripting, making the animation play or change between them. The animator has to be configured, making the default state of the animation and setting the condition of each transition between the animations (trigger, bool, int, float...) The figure 43 represents the animator for Mary, how almost every animation it is connected between each one, and the transitions between them.

Finally, each animation can be called from script, in a concrete time, and changed it between animation when collide or when something happens.

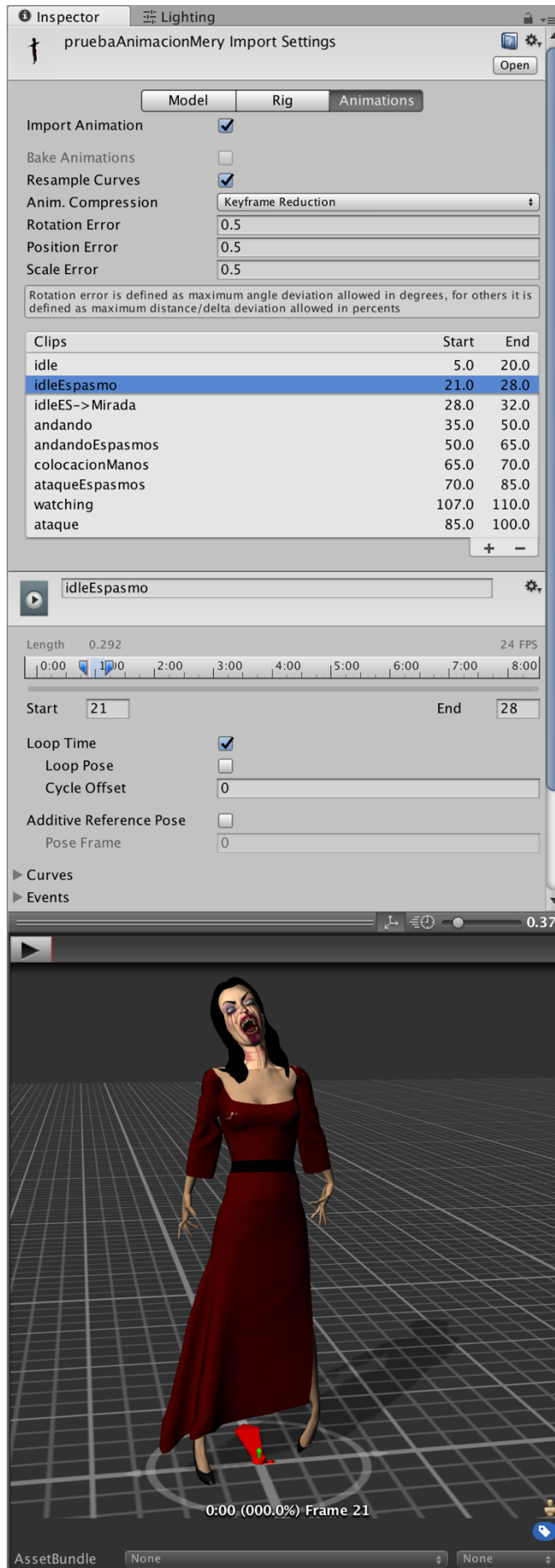


Fig. 42 – Prefab animation menu

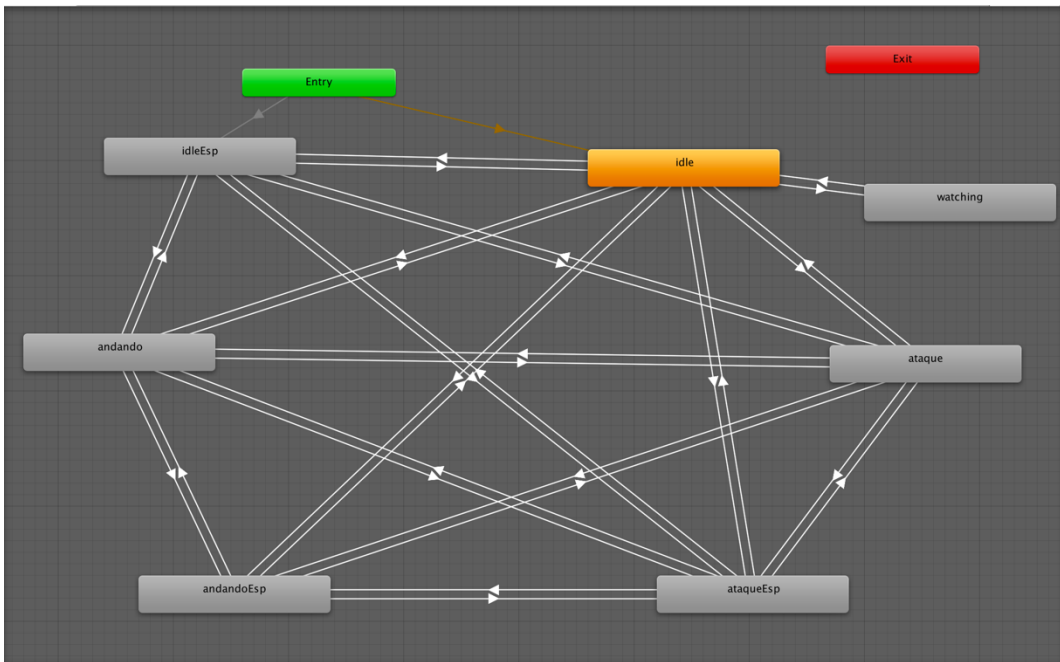


Fig. 43 – Mary animator

3.2 Scripting

All the artistic elements have been done already. Now it is time to move around the environment and interact with it. It is fundamental a controller, in this case, as the game is in first person will be necessary a first person controller. Also there will be tree mechanics, zoom in, pick up an object and translate in a determined position and read some descriptions from objects in the scene. All this mechanics will be explained next. Now its time to implement the fps controller and the gameplay, how the player it is going to interact with the different elements of the scene.

3.2.1 Character Controller

The character controller allows the player to move around the environment. In the standard assets, that comes by default in Unity, there is a FPS Controller created. This controller has a character controller with a rigid body attached as parent and as a child it has a camera that will follow the mouse and the character controller movement.

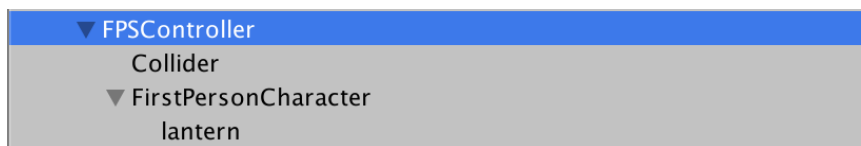


Fig. 44 – FPS Controller Hierarchy

The camera must be modified to add some imperfection, to avoid go through the environment and to add post processing effects that will be explained later. The field of view has to be modified to give a more realistic view, in this case a value of 70 will be perfect, making a more eye looking and realistic view. Also the occlusion culling must be checked to avoid draw polygons outside the camera planes. And finally the near plane has to have a value of 0.1, this plane will be near the origin of the camera making this not going through the collides of the environment, making visible the background.

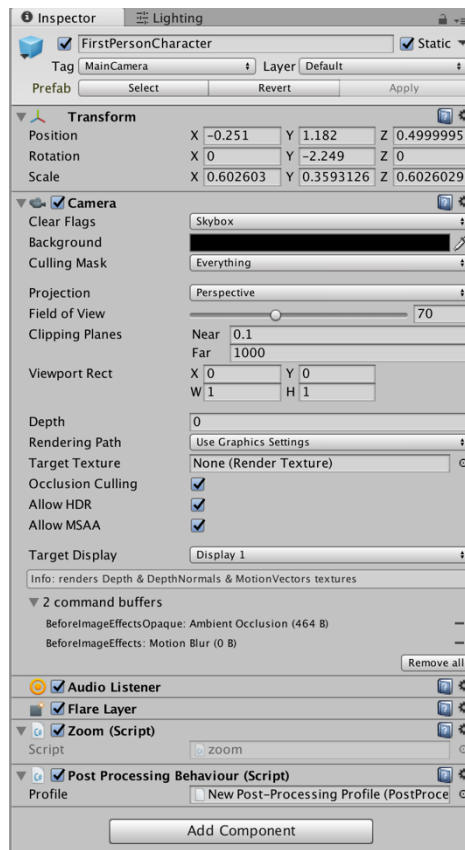


Fig. 45 – Camera Menu

This game will be dark and sometimes will be hard to see something in the environment, it is completely necessary something that helps the player to see, so a lantern was added to the fps controller. This was made by adding a spotlight under the camera child and placing a cookie to the spotlight to give a more realistic lantern looking. A cookie is an alpha image that will add noise or some kind of form to the projection of the light. The spotlight must be under the camera to follow the movements of the mouse.



Fig. 46 – Cookie, alpha mask

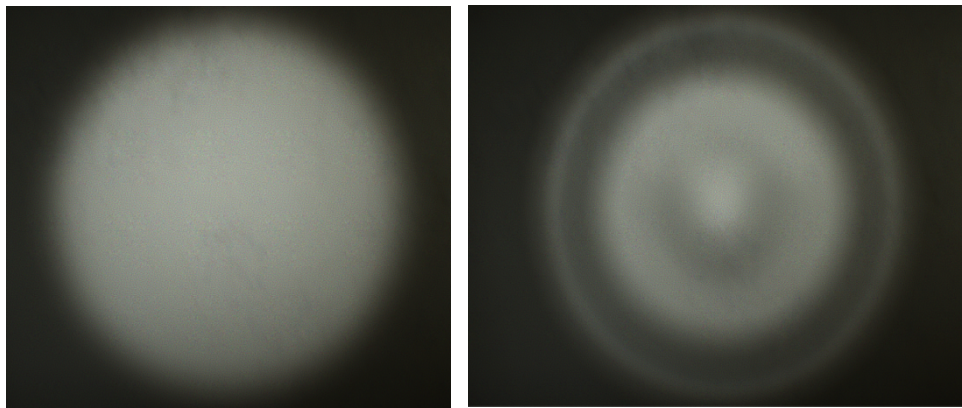


Fig. 47 – Without and with cookie

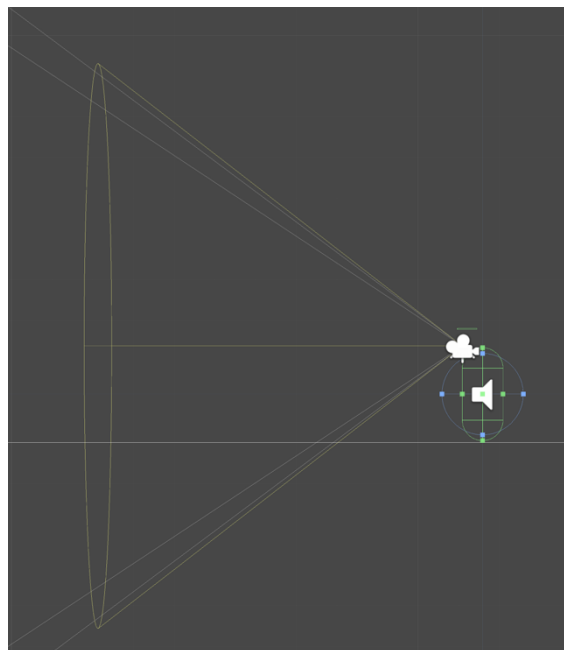


Fig. 48 – Full FPS controller

3.2.2 Zoom in

The main objective of this mechanic is when the player is pressing a button (E button) the camera will zoom to a limit, if the button is released the zoom will come back to the original position. This is the base for the rest of the mechanics, first it is necessary to zoom in and then the object will transform or will appear a description. This is done with the field of view of the camera, the value is increased for each frame in the update function till a limit, when the button is released the field of view is decreasing faster (than zooming in) till the base value, 70 in this case.

```
void Update () {  
    if (Input.GetButton("Zoom") && Camera.main.fieldOfView > 50) {  
        Camera.main.fieldOfView -= 0.5f;  
    }  
  
    if (Input.GetButtonUp("Zoom") && Camera.main.fieldOfView < 70) {  
        while (Camera.main.fieldOfView < 70) {  
            Camera.main.fieldOfView += 0.1f;  
        }  
    }  
}
```

Fig. 49 – Zoom script



Fig. 50 – Zoom out



Fig. 51 – Zoom in

3.2.3 Pick up and setVisible

Once the basic mechanic is done, it is possible to work with the rest. The mechanic that it is going to be explained is pick up and object and set visible another one. In this game this mechanic it is issued to tell the history, when a crumbled photo is picked up, this will be deleted, next a sound will be played and finally the photo will be visible in the right spot of the environment. Each photo has a description that will tell part of the history, but this mechanic will be explained later.



Fig. 52 – Crumbled photo



Fig. 53 – Picture

To get this mechanic it is fundamental to create a Raycast. Raycast it is an invisible ray casted from a point to another, as his name suggests. The original point in this case is the center of the camera, and the final point long enough to cover the whole environment. This mechanic could have been done by distance, but some crumbled papers are together and could generate problems, even this process was tested but the Raycast one was chosen instead of distance because the results were so much better and easier to work with. When the Raycast intercepts the collider with a determined tag and the zoom it is as his maximum the mechanic continues. Also the Raycast has a debug ray to check in the scene where collides and with what.

```
void Update () {  
    RaycastHit hit;  
    Ray ray = camera.ScreenPointToRay(Input.mousePosition);  
  
    Debug.DrawRay(ray.origin,ray.direction*70);  
}
```

Fig. 54 – Raycast

The environment has a total of 10 crumbled photos hidden on it, each one has a collider with a different tag and a GameObject with a different name each. In the figure (X) there just a example of tree photos. This mechanic it divided in three steps. First, when the player finds crumbled photo, the camera has to be zoomed till the maximum to activate the next step of the script, then the photo must be visible in the right spot. Second, the sound that indicates the player that the photo has been picked up has to be played. And finally, the crumbled photo has to be deleted.

```
if (Camera.main.fieldOfView < 51){  
    if (hit.collider.tag == "photo") {  
        hands.SetActive (true);  
        AudioSource.PlayClipAtPoint (paper, GetComponent<Collider> ().transform.position);  
        Destroy (hit.collider.gameObject);  
    }  
    else if (hit.collider.tag == "photo1") {  
        multipleFaces.SetActive(true);  
        AudioSource.PlayClipAtPoint(paper, GetComponent<Collider>().transform.position);  
        Destroy (hit.collider.gameObject);  
    }  
    else if (hit.collider.tag == "photo2") {  
        merySmiling.SetActive(true);  
        AudioSource.PlayClipAtPoint(paper, GetComponent<Collider>().transform.position);  
        Destroy (hit.collider.gameObject);  
    }  
}
```

Fig. 55 – Crumbled paper detection

3.2.4 Show description

Now that the photos can be placed in the right spot, each one its going to tell a part of the history by zooming in it. This phrases will appear at the bottom of the screen.

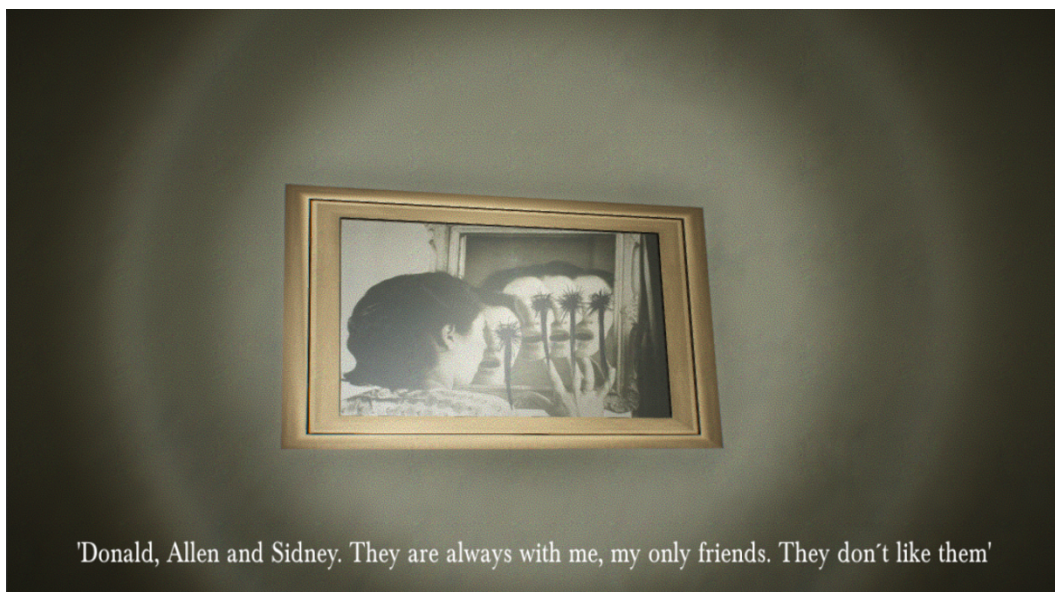


Fig. 56 – Picture Description

Each photo has a collider and tag, that will interact when the Raycast go through it at the maximum zoom of the camera. Also the script has a debug to check if the ray has collide with the collider or not and print a message in the console.

```

if (Camera.main.fieldOfView < 51){
    if (Physics.Raycast(ray, out hit)) {
        Debug.Log("HIT");
        if (hit.collider.tag == "photo1D") {
            texto.text = "";
            texto.text = "'Donald, Allen and Sidney. They are always with me, my only friends. They don't like them'";
            timer1 = true;
        }
    }
}

```

Fig. 57 – Detect picture and show description

To modify the text in the screen it is necessary to create a canvas first where all the text is going to be change, this text is changed inside the script. This canvas has to have a concrete configuration because the resolution can destroy the text and may not be visible. Also the text in the screen has to stay there a determined time, it is necessary a timer to count the time and delete the text when the time has passed.

```

if (timer1) {
    time += Time.deltaTime;
    Debug.Log (time);
}
if (time >= 8.0f) {
    timer1 = false;
    texto.text = "";
    time = 0.0f;
}

```

Fig. 58 – Subtitles timer

With this scripts the mechanics of the gameplay are done for each iteration of the loop of the game. Now it is necessary to program the change of scenes, change between each iteration.

3.2.5 Changing between scenes

The mechanics of each iteration is ready, now the player has to be able to change between the iterations that compound the game. This change between scenes has to be done with a fade in and a fade out to be progressive and smooth. Also the data of the past iteration has to be saved to the next iteration. For example, if 4 crumbled papers have been picked up, 4 photos have been placed on each spot, that information has to be saved between the future iterations. There are three function used in this script, LoadScene, DontDestroyOnLoad and StartCoroutine. When the player collides with a collider, a coroutine it is called which execute the fade in/out and the load scene.

```

void OnTriggerEnter(Collider collision){
    if (collision.gameObject.tag == "Player") {
        StartCoroutine (Fading ());
    }
}

IEnumerator Fading (){
    anim.SetBool ("Fade", true);
    yield return new WaitForSeconds (1);

    Application.LoadLevel ("iteracion2");
    SceneManager.LoadScene ("iteracion2", LoadSceneMode.Single);
}

```

Fig. 59 – Load Scene

3.3 Image Effects

All the models are done and animated, now its time to give more realism and detail to the scene adding some effects to the camera. This effects will have a huge impact in the scene, changing almost completely the experience.

3.3.1 Antialiasing

It is a set of techniques used to improve the final quality of the images visible. This process is used to delete or blur the jaggies that use to appear when different pixels are to big or they are not being processed correctly.



Fig. 60 – Example of Antialiasing

3.3.2 Ambient Occlusion

Rendering and shading technique which calculates the amount of lighting (ambient lightning) that a determined point in the scene is going to receive or get hit. How bright the light should shine in a specific part of the surface.

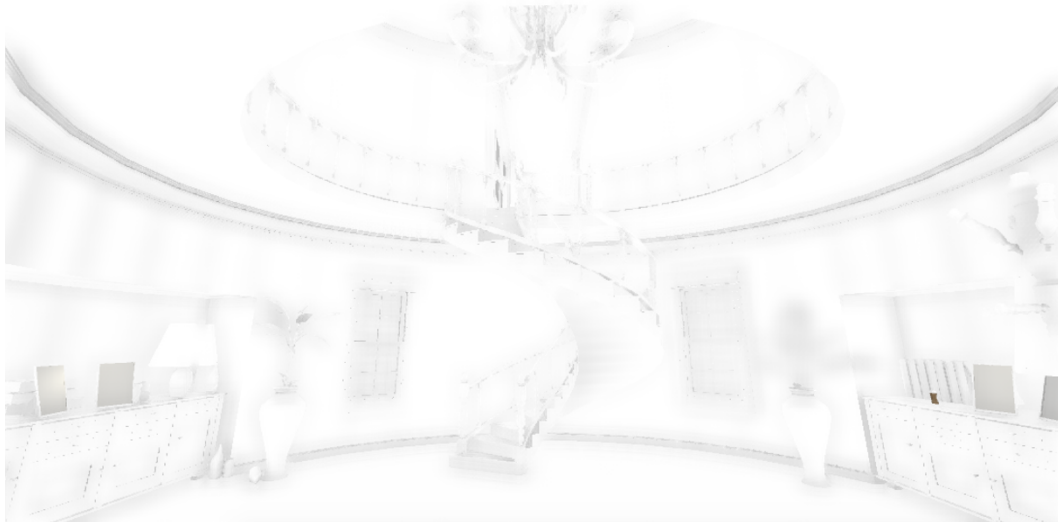


Fig. 61 – Environment rendered just with Ambient Occlusion

3.3.3 Depth of Field

This technique is used to add a blur effect, to difference the elements from the foreground to the background, simulating a lens focus. This process is used in First Person to add more realism on the scene.



Fig. 62 – Depth of Field in Skyrim

3.3.4 Motion Blur

This process adds a trail when the camera is moved. This effect will add and blur effect to the objects that are on the sides. Sometimes this effect can be annoying, it is necessary to be configured correctly.



Fig.63 – Motion Blur in Crysis 2

3.3.5 Eye Adaptation

This effect makes the exposure of the scene to be automatically adapted and adjusted to recreate the effect of human eyes when change passes between a lighted environment to another much darker or viceversa.



Fig. 64 – Eye Adaptation in a Demo in Unreal 4

3.3.6 Bloom

This graphical effect makes the light blocked by an object to shine brightly around the edges and extending around and objects around the blocked one.

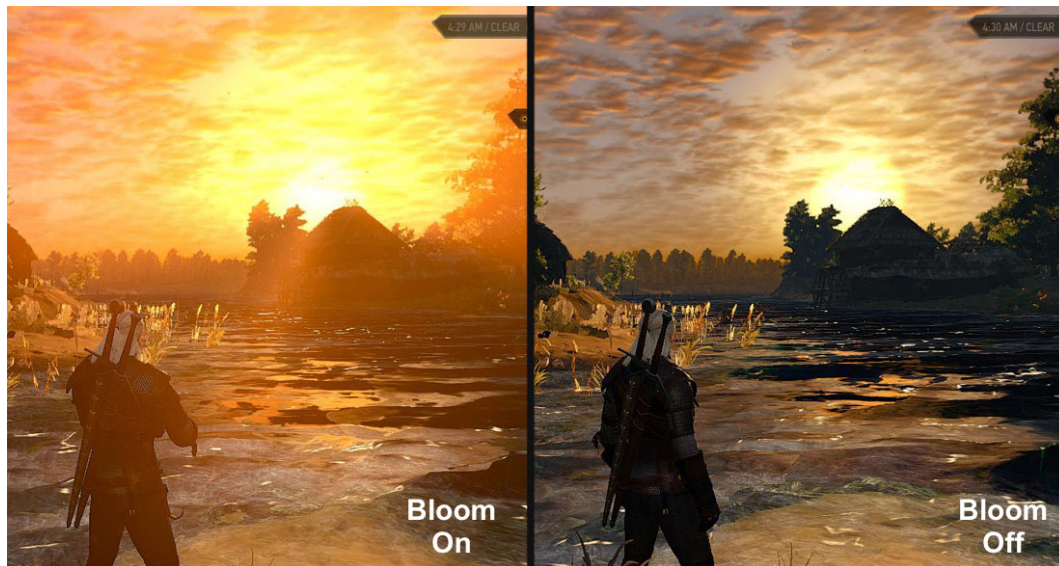


Fig. 65 – Bloom effect in The Witcher 3

3.3.7 Color Grading

This process changes the color of everything render by the camera.



Fig. 66 – Color grading in a Unity Demo

3.3.8 Chromatic Aberration

It is an effect which there is a problem of lens when tries to render all the colors on the same point. When colors are incorrectly refracted by the lens, making that the colors do not combine as the should.



Fig. 67 – Chromatic Aberration in Destiny

3.3.9 Grain and Vignette:

This process add noise to the image and darkens the corners and sides



Fig. 68 – Left for Dead grain and vignette

All this processes have been made in Unity with a free downloadable package in the asset store done by Unity. This package provides to the user the chance to add all this effects into an own project. To produce a better visual quality is necessary to modify all the parameters correctly, if a parameter is changed wrongly the whole visual effect can be destroyed. In the figure 68 can be observed how the environment has been changing with every filter.



Fig. 69 – Environment with different visual effects

4. Results

All the results done in this Project will be in this chapter. All will be presented with photos; all assets implemented, illumination and sounds with different illumination. The photos have been taken from different spots and angles in the environment trying to obtain the most visual impact.



Fig. 70 – Full completed environment



Fig. 71 – Environment left side



Fig. 72 – Environment right side



Fig. 73 – Environment top side



Fig. 74 – Environment top side, rest zone



Fig. 75 – Environment top side, hall



Fig. 76 – Lobby different illumination 1



Fig. 77 – Environment top side with Mary



Fig. 78 – Environment lobby diferent illumination 2



Fig. 79 – Environment top side, hall with Mary



Fig. 80- Loop Zone

5. Project deviations and objectives Done

The main objectives of this project have been done, although there is some aspect that has to go further. Now it is going to be listed all the objectives that were done in the technical proposal and if they have been finished or not.

- ✓ Create a surrounding, credible terror experience, where the player can feel fear and tension and suspense.
- ✓ Learn how to create photorealistic assets, understand most possible how Autodesk Maya, Zbrush, Unity and Substance Painter works
- ✓ Learn how to create visual impactful lights and transmit with lights and sounds, using real time lights, placing them in the right spot to create the right shadow or effect.
- ✓ Create a deep and elaborate narrative. Many of histories in terror has been told, it is necessary to pick up a good base history and transform it with no boundaries. And if the history is based in something real, it will add more realism and credibility.
- ✓ Learn how to create a good postproduction to improve the realism on the scene, the difference between a scene without postproduction with a scene with postproduction is abysmal. It is completely necessary to add camera effects to improve the final result.
- ✓ Create interesting mechanics, puzzles.
- ✓ Learn to scripting and program the gameplay
- ✓ Learn the workflow to create a game. How it is the best way to obtain a completed model

The game has ended like it was though at the beginning, a closed environment, with visual impact, with a background history, with simple mechanics and gameplay and with powerful sounds. But when the brainstorming started, one of the thing it was thought was how to create hair but not sculpting it. It was thought a different way to approach the hair in videogames, with strips and clipping maps. Actually this is the method used for hair in most videogames, and it was interesting to learn it and see how it would saw in real time inside the engine. But was impossible, the time was almost out and learning a new technique costs time and a lot of effort.

The objective “Create interesting mechanics, puzzles” is marked, because the basic steps of this objective are done, but would have been great to go a step further and create more interesting and innovative puzzles and mechanics. Although the basic in this field is done.

At the beginning of the project it was told to add a mental illness like schizophrenia in the history and gameplay, but later on it was impossible to get more information about the subject, all the information was so vague and alterative, the information in one page was completely different in another. So even this mental illness is added on the project it is not a protagonist as it was thought at the beginning.

Roughly, it is possible to say that the main objectives are done, the game is fully playable, it has an scary atmosphere, an animated character, an environment plenty of assets, a good illumination and sounds and a history. Also all the models and assets has been done handmade, it was a perfect exercise to learn how to create all this models and all of them optimized to be played in a midrange computer.

6. Hours control and planning

Once the project has been done, all the hours that have been noted on the calendar was divided in task. Each day of work, was noted on a notebook, with the number of hours, the task done and the date. When all the hours were added, the total has more hours than expected. The hours listed in the technical proposal were:

1. Creación del Game Design Document (GDD):
 - 1.1. Historia (5h)
 - 1.2. Personajes (5h)
 - 1.3. Diseño de niveles (10h)
 - 1.4. Gameplay y puzzles (10h)
2. Modelado Inorgánico:
 - 2.1. Modelar y texturizar los escenarios (30h)
 - 2.2. Modelar y texturizar assets (35h)
3. Modelado orgánico:
 - 3.1. Modelar, texturizar, riggear y animar los enemigos (45h)
 - 3.2. Modelar y texturizar otros elementos orgánicos (15h)
4. Implementación de mecánicas y de modelados
 - 4.1. Implementación del movimiento del protagonista (10h)
 - 4.2. Implementación de la cámara (5h)
 - 4.3. Implementación de los escenarios (10h)
 - 4.4. Implementación de los enemigos y assets (15h)
 - 4.5. Implementación de los puzzles (30h)
5. Iluminación y postproducción:
 - 5.1. Implementar luces (15h)
 - 5.2. Añadir filtros de cámara (5h)
6. Sonidos
 - 6.1. Búsqueda y grabación de efectos de sonido. (10h)
 - 6.2. Implementación de los efectos de sonido y música ambiental (5h)
7. Testear, documentar y presentar el proyecto:
 - 7.1. Testeo del juego (10h)
 - 7.2. Redactar la memoria final (15h)
 - 7.3. Preparar la presentación y la demo (15h)

The table 1 represent the total number of hours expended in each task done for this project.

Task	Hours
Task 1 – Brainstorming, GDD and Technical document	20
Task 2 – Modeling and texturizing environment	65
Task 3 – Modeling and texturizing Mary	60
Task 4 – Rig, Skinning and Animation	12
Task 5 – Modeling and texturizing assets	70
Task 6 – Scripting	30
Task 7 – Implementation (sounds, lights, colliders, assets)	30
Task 8- Testing	22
Task 9 – Final memory, presentation, and video	70
Total Hours	379

The number of hours for some task fits more or less the original hours proposed in the technical document, but the biggest difference between the proposed and the original is the number of hours done to modeling and texturizing the different assets in the scene. It took 70 hours total (30 hours of difference).

7. Conclusion

Since the beginning of this degree, always have been wanted to create a realistic terror experience. But some years ago, this task was not possible, it has to learn a lot of about videogames. Those years were passing and more knowledge was acquired, first a small 2D game, then the first 3D project and finally this Final Degree Project. In each project something new was learned, how to create sprites 2D for videogames, how to animate in 2D, how to program the mechanics, how to create basic assets...

All the things that have been learned in this for years are shaped in this project, the last project, the project that will give a step to something different. This project has been done with love and passion, because this moment, the moment that it was able to create an own project since the begging, finish it, and present it, it was a moment really waited.

There are plenty things that have been learned from this project, some are really important, like one by his own can create a game in a short time with earnings, effort and believing in oneself, also to work hard and create schedule to work everyday. The objectives that were told in the chapter number 5 was most about learning, next it is going to be explained what was learned in this project:

- In this project it has been learned how to create photorealistic assets since the scratch; when this project began the knowledge of Autodesk Maya was minimum, it was necessary to learn how to use the different tools and menus that compose this powerful program.

- Also it was learned how to distribute and create powerful and beautiful lights in a Unity project, how adding and changing some effects and lights can change the complete atmosphere of a scene.

- it has to be mentioned that at the beginning of this project it was not interest in scripting and programming, but thanks this project it has been learning how to program and create mechanics, it was a good exercise to open the mind and jump out the comfort zone, it was used to create just artistic stuff and not much programming but with this project the scripting and programming it is not a problem anymore.

- Learning how the basic workflow for a game is done by companies, it is fundamental to know how just with a piece of paper, with an idea, a game can be constructed and created. All the intermediate process, the optimization problems, exporting models and file types...

- Learning how to rig, skin and animate a character.

This project is a part of a bigger idea that will be continued for sure. This project will pass for some retouch in the next months and it will be posted in some platforms like itch.io, to be know the creator a bit and to transmit the game as many people as possible.

8. Bibliography

[1] Pixologic Zbrush
<http://pixologic.com/>

[2] Substance Painter
<https://www.allegorithmic.com/products/substance-painter>

[3] Substance Painter Materials
<https://share.allegorithmic.com/>

[4] Marvelous Designer
<https://www.marvelousdesigner.com>

[5] Autodesk Maya forum
<https://forums.autodesk.com/t5/maya-forum/bd-p/area-b201>

[6] Polycount forum
<http://polycount.com/forum>

[7] Unity forum
<https://forum.unity3d.com>

[8] Stackoverflow
<https://stackoverflow.com>

[9] References
<https://es.pinterest.com>

[10] MK Ultra
https://es.wikipedia.org/wiki/Proyecto_MK_Ultra

[11] Schizophrenia
https://es.wikipedia.org/wiki/Esquizofrenia_paranoida

[12] Antialiasing
<https://es.wikipedia.org/wiki/Antialiasing>

[13] Ambient Occlusion
<https://gamedev.stackexchange.com/questions/23/what-is-ambient-occlusion>

[14] Depth of field
https://www.reddit.com/r/truегaming/comments/15lvuf/depth_of_field_and_its_use_in_video_games/

[15] Motion Blur
<https://www.escueladevideojuegos.net/que-es-el-motion-blur/>

[16] Eye adaptation
<https://docs.unrealengine.com/latest/INT/Engine/Rendering/PostProcessEffects/AutomaticExposure/>

[17] Bloom effect
<https://www.giantbomb.com/light-bloom/3015-197/>

[18] Grain effect

<https://www.giantbomb.com/film-grain/3015-487/>

[19] Color grading

https://en.wikipedia.org/wiki/Color_grading

[20] Sounds

<https://freesound.org>

[21] Audacity

<http://www.audacityteam.org>