UNIVERSITAT
JAUME·I

Design and implementantion of the art and visual aspect of a video game in virtual reality platforms

Joaquín Soler Tomás

Final Degree Work

Bachelor’s Degree in Video Game Design and Development

Universitat Jaume I

May 20, 2019

Acknowledements

I would like to thank this project to my colleagues Antoni Arlandis and Daniel Castro for the work contributed in their respective parts of this project and without them, this project could not have been realized.

I would also like to thank my tutor Marta Martín for the advice, knowledge and support provided for the realization of this project, as well as the other teachers for the different knowledge I could acquire from them and have facilitated the development of this project.

Finally, I would like to thank my family for their unconditional support.

Abstract

This document presents the Final Report of the Final Degree in Design and Development of Videogames, which consists of the development and implementation of the general aesthetics of a horror videogame in virtual reality, from the first designs and concept art of the characters and environments, to the animation of these characters and incorporation to the game engine, passing through 3D modeling, illustrations, user interface design and programming, particle systems and dynamic change of textures by code, that provide the videogame with a realistic and credible aspect that helps the player to get immerse into the game.

The engine to be used will be Unity, and de different tools and opportunities offered by the virtual reality platform will be used for the development of all sections mentioned above.

Key words

Virtual Reality, Design, Art, User Interface, Videogame.

Index

Acknowledements 2
Abstract 3
Key words 3
Figure Index 6
1. Introduction 7
1. 1 Work motivation 7
1. 2 Objectives 7
1. 3 Environment and Initial state 7
1. 4 Planning 9
1. 5 Related Courses 10
2. Game design 11
2. 1 Synopsis 11
2. 2 Overview 11
2. 2. 1 Game concept 11
2. 2. 2 Genre 11
2. 2. 3 Target audience 12
2. 2. 4 Look and feel / game experience 12
2. 2. 5 Gameplay & mechanics 12
2. 3 Gameplay progression 13
2. 3. 1 Representation of pressure 14
2. 3. 2 Level structure 14
2. 3. 3 Movement & Controls 15
2. 3. 4 Physics & Objects 15
2. 3. 5 Actions 16
2. 3. 6 Screen flow 16
2. 3. 7 Game flow 16
2. 3. 8 Replaying & Saving 17
3. Narrative Design 17
3. 1 Storyline 17
3. 2 Characters 17
3. 3 Game world 17
3. 4 The diferent parts of the game structure 18

3. 5 Narrative structure 18
4. Artistic Design 19
4. 1 Characters 19
4. 1. 1 Concept art & Reference images 19
4. 1. 2 3D Modeling 20
4. 1. 3 Texturing 27
4. 1. 4 Animation 29
4. 1. 5 Import to game engine 30
4. 2 Game World 31
4. 2. 1 Modeling 32
4. 2. 2 Texturing 33
4. 2. 3 Import to Game engine 35
4. 3 Particles systems 36
4. 4 HUD & MENUS: 37
5. Functional & Technical specification 38
5. 1 Tools 38
5. 2 HUD & Menu programming 39
5. 2. 1 Raycast 39
5. 3 Dialogue System 40
5. 4 Dynamic Textures 42
6. Results 44
6. 1 Images of Final Result 44
6. 2 Project Access 46
7. Conclusions & Future Work 47
8. References 48
10. Appendix 49
10.1 One sheet: 49
10.2 Character's files: 50
10.3 Zledic's modeling process in Zbrush: 54
10.4 Zledic Biped Skeleton Adjustment 58
10.5 Sanja's modeling & texturing process from Zbrush to Substance 60
10.6 Texturing object process, video: 63

Figure Index

Figure 1: Gantt Chart.. 9
Figure 2: Resident Evil 7, famous horror video game saga [8] 11
Figure 3: House and Bunker plans 15
Figure 4: Call of Duty: Black Ops, Visual and art reference for bunker scenes........... 18
Figure 5: Zledic, Sanja and Dajana final designs 20
Figure 6: Basic human structure using Zspheres in Zbrush....................................... 21
Figure 7: Levels of subdivision 22
Figure 8: Modeling the details of Sanja's skin... 22
Figure 9: Retopology guides and final result in Zbrush 23
Figure 10: UV map of Sanja’s low poly model.. 24
Figure 11: Sanja’s low poly model: With and without normal map (5600 polygons each) 24
Figure 12: Poly-strips example in Zbrush (done manually)...................................... 25
Figure 13: Poly-strips in 3ds Max using Cloth modifier... 26
Figure 14: Modeling process of Sanja's dress (basic structure, finished model & with Cloth modifier) 27
Figure 15: Albedo-transparency, Ambient Occlussion, Metallic and Normal maps.... 28
Figure 16: Textured in detail of Sanja's skin... 28
Figure 17: Adjustment of the Biped skeleton with the Zledic model 29
Figure 18: Model and textures imported in Unity.. 30
Figure 19: Material configuration in Unity... 31
Figure 20: Layers of Fear screenshot, the project's biggest artistic scenario reference. 32
Figure 21: House structure in 3ds Max 33
Figure 22: 3D furniture models in 3ds Max 33
Figure 23: UV maps of the house corridor... 34
Figure 24: Texturing of a room in Substance Painter... 35
Figure 25: Aniso Level in Normal maps 36
Figure 26: Particle Creation Process Scheme.. 36
Figure 27: UI design in Illustrator 37
Figure 28: Sanja's final render, front & back 44
Figure 29: Sanja's in-game screenshots: Corridor 45
Figure 30: Sanja's in-game screenshots: Sanja's room... 45
Figure 31: mist particles result 46

1. Introduction

1. 1 Work motivation

Virtual reality is a booming market and provides another vision compared to the classic game, currently this market is reduced to a large number of experiences or small games, but the lack of horror video games with a developed narrative and created specifically for this type of platform, taking advantage of all the mechanics it offers, made us consider the possibility of creating one ourselves that could easily enter this market.

1. 2 Objectives

- To design from scratch, scenarios and environments within an aesthetic and history, including objects and textures.
- To create a series of textures with photorealistic style and a good definition when viewed from the virtual reality glasses, Oculus Rift.
- Creation and artistic development of NPC characters with a realistic human appearance, with their own models and animations.
- Development of particle systems in Unity in an optimized way that doesn't slow down the processing and obstruct the game experience and contribute to the generation of the atmosphere.
- Creation and programming a menu & Hud system for a virtual reality game.

1. 3 Environment and Initial state

Vornik is a virtual reality video game project in which three people are involved, so the work that each one would do in the different areas had to be specified.
The work has been divided into three parts:

- Game design and narrative.
- 3D design, interface and particles.
- Programming: implementation of avatar and puzzles.

The part that I have developed and therefore will be explained in this document is dedicated to 3D design, interface and particles.

Besides, taking into account that all members of the group had to accomplish with minimum programming requirements, aspects related to the mechanics provided by virtual for navigation between menus, as well as the change of textures in a dynamic way by code or the programming of different dialogues in-game will also be developed. Finally, mention the constant communication between the three members of the group, vital for the development of this project.

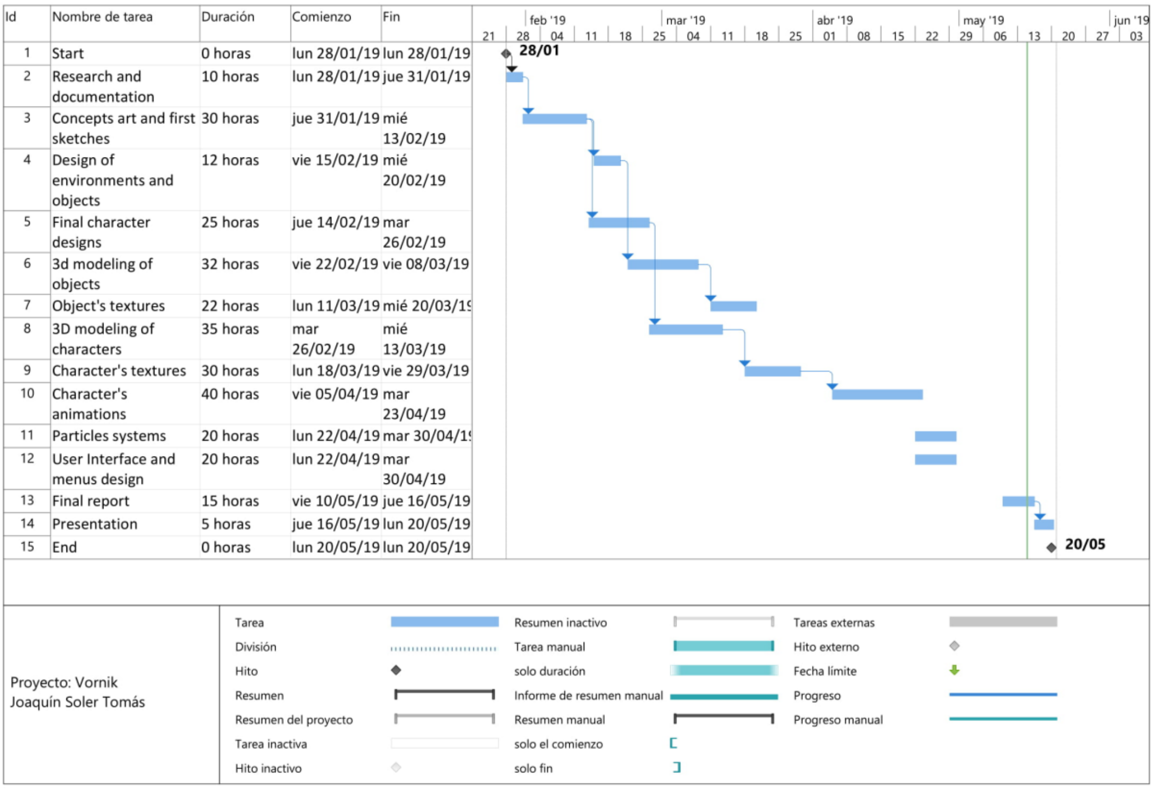
1. 4 Planning

During the first phases of the development of the project it was necessary to focus on clarifying the aesthetics, as well as on the elaboration of the different concepts-art to define it.

Then I decided to focus on the creation of 3D models of the different scenarios as well as the objects in them because my other two colleagues had to work with these objects to program them or to perform lighting tests with them.

Then began the phase of character creation, the longest and most complicated due to the number of different processes that exist in it, something that is reflected in the hours invested.

Finally, when the programming part of my colleagues was practically finished, I focused on developing the programming of menus and interface, dialogues and final visual retouches with particle systems.



Id	Nombre de tarea	Duración	Comienzo	Fin
1	Start	0 horas	lun 28/01/19	lun 28/01/19
2	Research and documentation	10 horas	lun 28/01/19	jue 31/01/19
3	Concepts art and first sketches	30 horas	jue 31/01/19	mié 13/02/19
4	Design of environments and objects	12 horas	vie 15/02/19	mié 20/02/19
5	Final character designs	25 horas	jue 14/02/19	mar 26/02/19
6	3d modeling of objects	32 horas	vie 22/02/19	vie 08/03/19
7	Object's textures	22 horas	lun 11/03/19	mié 20/03/1
8	3D modeling of characters	35 horas	mar 26/02/19	mié 13/03/19
9	Character's textures	30 horas	lun 18/03/19	vie 29/03/19
10	Character's animations	40 horas	vie 05/04/19	mar 23/04/19
11	Particles systems	20 horas	lun 22/04/19	mar 30/04/1
12	User Interface and menus design	20 horas	lun 22/04/19	mar 30/04/19
13	Final report	15 horas	vie 10/05/19	jue 16/05/19
14	Presentation	5 horas	jue 16/05/19	lun 20/05/19
15	End	0 horas	lun 20/05/19	lun 20/05/19

Figure 1: Gantt Chart

1. 5 Related Courses

VJ1204 – ARTISTIC EXPRESSION

VJ1216 – 3D DESIGN

VJ1223 – VIDEO GAME ART

VJ1227 – GAME ENGINES

2. Game design

2. 1 Synopsis

Horror video game for Oculus Rift, where we control Zledic, a mysterious character with a terrible background. Throughout the game, the player will have to solve different puzzles with mechanics designed for VR.

2. 2 Overview

2. 2. 1 Game concept

Vornik is a horror videogame where we control Zledic, a former sergeant in a bloody war who has been captured and interrogated to discover the reality of his actions during the war.

The player must solve different puzzles to advance and discover the dark past that concerns the protagonist. The game action is made in first person in virtual reality, using the platform Oculus Rift.

Vornik's visual style is realistic 3D, using various types of textures to achieve a result as close as possible to reality. The different objects of the game are well differentiated as far as modeling, texturing, physical behavior, and interaction with the characters, are concerned.

2. 2. 2 Genre

Horror video game.

Figure 2: Resident Evil 7, famous horror video game saga [8]

2. 2. 3 Target audience

Adult players interested in horror video games who are patient, like to investigate, enjoy the sensation of being in an unknown and hostile environment and are also users of the Oculus Rift platform.

2. 2. 4 Look and feel / game experience

The player must feel like the protagonist, with intuitive controls that help this feeling and allow him to interact with objects in the environment in a comfortable way.

2. 2. 5 Gameplay & mechanics

Thanks to the use of the virtual reality platform, Oculus Rift, a series of mechanics has been implemented to enhance the immersion and game experience of the player:

- *Interaction with objects:* The direct interaction with the different objects that appear within the game space are be the basis of the mechanics of this videogame, the player is be able to pick them up, throw them, place them in different places or fit them together among other actions, all so that the player can complete the different puzzles that are posed to him.

  On the other hand, the NPC characters that appear in Vornik are not directly interactive, the interaction between player and certain NPC is through events controlled by the game designers.

- *Well-designed environment:* The game space within Vornik is not large, on the contrary, it is limited to a relatively small space but not uninteresting. One of the important aspects in the design of this environment was to maintain a feeling of pressure, which was achieved through this reduced play space, with narrow corridors and small rooms that cause the feeling of psychological terror over the direct and conventional fear.

  In order to show the three main areas of the game, a realistic structure has been followed, both in size and design, collecting data from real stories and generating the corresponding blueprints for the subsequent 3D design.

  One of the main goals when designing the environment was how to represent the memories of the protagonist, which was solved by placing the player within the moment in which the memory was made, ie, the house of the protagonist.

- *Clear main objectives:* It is important that the player knows at all times what his main objective is, as well as that he has the necessary information to complete it getting the key objects and pushing forward in the story.

  In addition, the inclusion of dialogues by interrogators, both in the interrogation of the protagonist himself and in his memory, help to achieve this objective.

  Also note that there are secondary goals for players who are interested in researching and want to know more about the story that Vornik has to offer.

- *Coherent puzzles:* offer the player different ways of interacting with the objects of the game world, always from the coherence of the environment and story.

  These have a progressive difficulty, being the second memory scene the one that offers the most complex puzzle, taking the player from one object to another until completing them all.

  To represent the fulfillment of the puzzles, three images are used in the initial room, which form the three memories into which Vornik's structure is divided. Each of these images is cut into as many parts as puzzles are in the scene, so to complete the image, the player must solve all the puzzles.

2. 3 Gameplay progression

This section was developed mainly by Daniel Castro Arjona

The player has to advance through the puzzles designed in each scene, allowing him to get a key object that will open up one of the important memories he is looking for. Once he has reached that memory, the player will return to the real world and proceed to the next level.

The objective of each of the levels will be known by the player through the interrogation that is taking place in the real world where interrogators intervene, the main interested in knowing the story of the protagonist, Zledic.

2. 3. 1 Representation of pressure

The pressure for the player, besides the small game scenario, is represented through two elements:

- *Interrogators:* Once inside the level, in case the player doesn’t know where to advance or is not able to solve the puzzles that are posed to him, the interrogators will intervene and ask the player why he doesn’t complete the objective, besides giving him certain clues that could be useful to advance.

  We also have the fact that an induction session cannot last forever, so a counter is used, which will make interrogators intervene more frequently and more vehemently as it reaches 0.

  If it reaches 0, the interrogators will remove Zledic from the induction and the player will have to repeat the phase again.

- *Sanja (Zledic’s daughter):* A character who represents Zledic's internal struggle against the interrogation he is suffering, since through it he is forced to reveal the acts he carried out in the past, but he doesn't really want to reveal it. Therefore, as the player progresses and advances in the story, Sanja will be much more aggressive to prevent the interrogators from knowing the truth about Zledic.
  Sanja appears in two different ways:

  - Mandatory appearance: obligatorily due to the development of the story and for the player to advance to the next phases of the game.
  - Secondary appearance: modifying the environment as for example painting the walls with crayons or making different noises in different points of the house.

2. 3. 2 Level structure

The game in its full versión has three phases or levels, but the work to be done for this project is a demo with a duration of approximately 15 - 20 minutes.

Each phase takes place inside the interrogation room or inside the house of the protagonist, both are made through plans or blueprints with real measures and through references. *(Figure 3)*

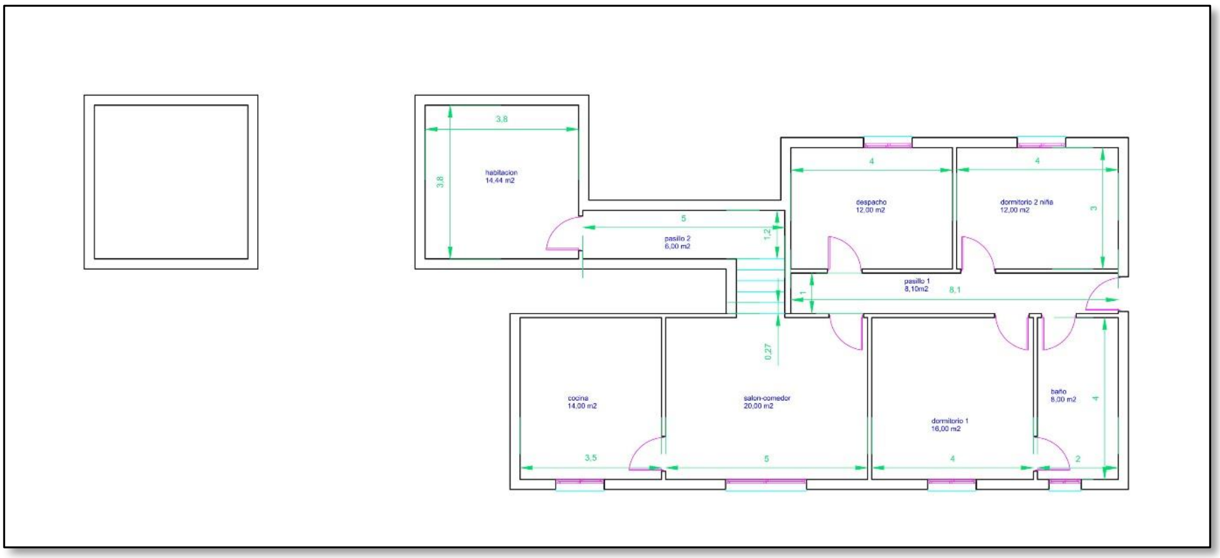


Figure 3: House and Bunker plans

2. 3. 3 Movement & Controls

For the movement of the character, taking into account that we are developing this video game for virtual reality, two different types of movement are implemented since the movement through the joystick, as if played with a conventional controller, can cause dizziness in different people.

- *Movement with Joysticks:* traditional, movement of the left joystick move the character in all directions of the horizontal axes, while the right joystick rotate the camera in small angles to make the handling more comfortable.

- *Teleport Movement:* Instantly move to any walkable área of the scene, by pressing a button and pointing the Oculus Rift controls to calculate the point at which the teleport will take place.

These two types of movement were developed by Antoni Arlandis Daviu.

2. 3. 4 Physics & Objects

As the story develops in two different realities, the real world and the memories of the protagonist, the objects behave according to where the player is, in case he is inside the real world (interrogation), the objects behave in a realistic way, with the corresponding rigidbody's parameters according to its design.

On the contrary, in the scenes of memory, the objects generally present a realistic behavior, because they represent a memory, something that actually happened, but certain objects can have other different and incoherent behaviors with the reality but it does makes sense with the history that is developing.

2. 3. 5 Actions

In general aspects, Vornik is a guided video game, however the player can stop to explore within the game space to complete secondary objectives and learn more about the story of the video game.

It also depends on where the player is since in reality, the interrogation, the protagonist is tied up and the player will not be able to move, he will simply be able to look around and observe the scene, but on the other hand, if the player is in the memory scenes, he will have more freedom of movement and exploration within the environment of the video game.

In addition to movement and exploration, the player is able to perform different actions with the objects he finds, such as pushing them, breaking them, saving them, pressing buttons, joining pieces, opening locks, turning keys etc, in order to complete the main objective and complete the actual phase

2. 3. 6 Screen flow

In order for the tension of the story to be reflected in what the player sees, the vision has been reduced through the use of lighting and post-processing effects, also giving the feeling of being more like human sight.

For the investigation of the game scenario to be resolved correctly, the use of the flashlight is essential when moving through the structure of the game.

Clear descriptions: the different ways of interacting are shown on screen by using UI, so that the objects that need explanation of how to be used have a description, so that the player know what to do with them.

2. 3. 7 Game flow

- First Splash screen: Unity cinematic.
- Second Splash screen: Team cinematic.
- Main Menu: a 3D screen appears listing the following options:
    - New game.
    - Load game.
    - Options -> secondary screen.

2. 3. 8 Replaying & Saving

Vornik is a linear story with no ramifications, so the game's replaying lies in obtaining all its achievements: get all the information provided about the character's story and go through secondary and optional objectives.

It also include an auto-save system in each phase, so that at the end of each memory phase the video game will be saved automatically to continue the game from that point.

Upon completion of the story the game gives the player the possibility to return to a specific point in the story and replay it without having to start from the beginning.

3. Narrative Design

This section was developed by Daniel Castro Arjona:

3. 1 Storyline

A former sergeant in a cruel war, has been caught and now a prisoner, he will have no choice but to tell everything that happened.

3. 2 Characters

- *Zledic Vučković:* protagonist and the only playable character, former sergeant of a hidden squad of his government, in charge of the "ethnic cleansing" of his country.

- *Dajana and Sanja Vučković:* wife and daughter of Zledic, they support their husband and father without really knowing their actions.

- *Interrogators:* Military police who chased and caught Zledic. They want to know everything about his past.

3. 3 Game world

Vornik names the military police operation to capture Zledic and all those who cooperate with him.

3. 4 The diferent parts of the game structure

The structure of the world in Vornik is closed, representing in each of the scenes, import key points of history, where the player is guided to a certain point giving him the freedom of movement necessary to explore and investigate, but wuthout going beyond the limits of the scenario.

The structure of the game is divided into the following parts:

- *Interrogation:* the real world, where Zledic is locked up and we enter his memory.
- *Zledic's memory:* interpreted as the protagonist's home, made up of the different rooms and areas of the house, where the most important memories will be accessed

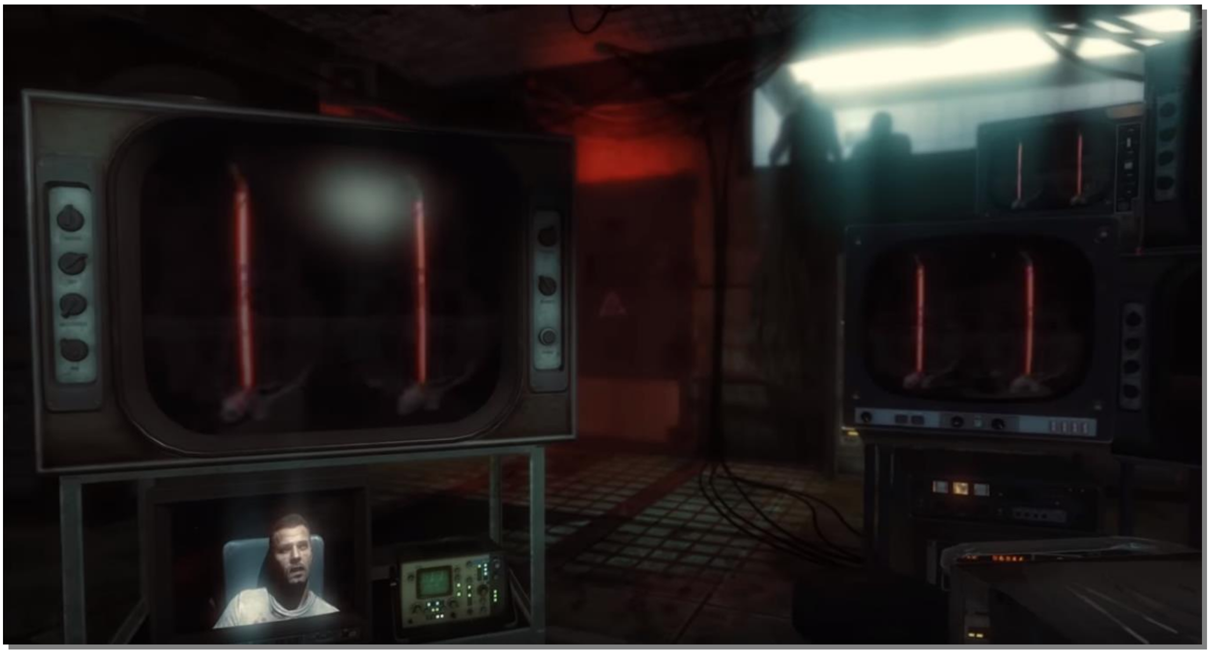


Figure 4: Call of Duty: Black Ops, Visual and art reference for bunker scenes

3. 5 Narrative structure

The video game is divided into three acts, each of which corresponds to a part of the story that the interrogators want to know, starting with Zledic, his life and way of being, followed by a family phase, in which the whereabouts of his wife and daughter will be known, ending with the last phase where the acts that Zledic carried out during the war will be seen and turned him into what he is now.

4. Artistic Design

To talk about the artistic design I have developed for this project, this section will be divided into 3 main parts: Character design, environment design, HUD and particles.

Each of these parts will explain the process carried out for its development, as well as the errors produced, how they were fixed and other possible options and processes that could be developed.

4. 1 Characters

The design of the characters has been divided into different phases, from their conceptual design to their incorporation into the game engine, including their 3D modeling, texturing and animation.

4. 1. 1 Concept art & Reference images

In this case we have that Vornik, takes its name from the ancient city of the Serbian Republic Zvornik, therefore its protagonist Zledic, as well as his family must contain characteristics of that region, and for it, as in any design process, the search for references was first made to specify the features of each of the characters, also taking into account the different profiles of each of them.

With all these reference images, as well as the profiles of each character, begins the sketching phase and concept art of each of the characters, capturing in the best possible way these features in the illustration. Once the result of the design is satisfied, the final illustration of the character is made, a drawing much more finished and polished than the previous ones. *(Figure 5)*

In addition, to facilitate the work of the later stages, different views of our character, front and profile view, in 'T-pose’, are made.

For this phase I have used Adobe Photoshop CC, a program more than known worldwide for this type of work.

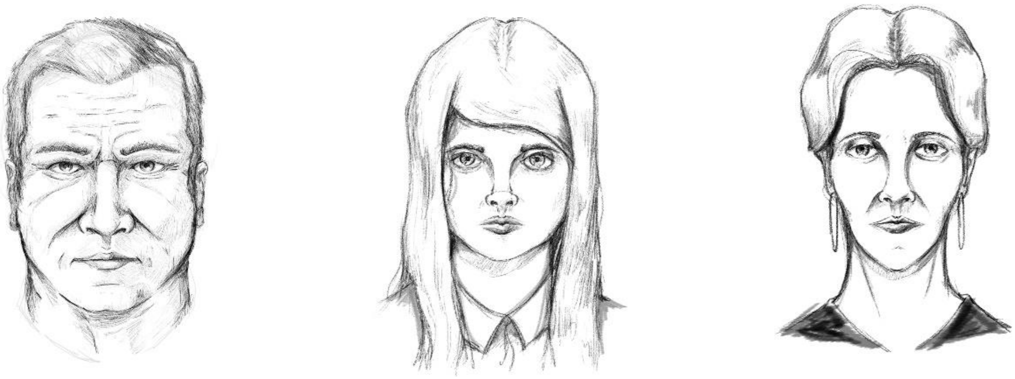


Figure 5: Zledic, Sanja and Dajana final designs

4. 1. 2 3D Modeling

Once the concept and basic design phases of the character have been completed, the 3D model of the character is created. This process depends on how our character is going to be used, whether to make a render and get images of it, for a game engine or many other options, in this case the models were made to be introduced into a game engine, so this process has a greater number of phases and complications, starting with the high-poly sculpting of the character.

I have worked with the Zbrush program, a 3d modeling, sculpture and digital painting software widely used in the video game and film industry [2].

This phase is divided into 3 distinct parts, body modeling, hair modeling and clothing modeling.

4. 1. 2. 1 Body Modeling

With the help of the front and profile images of our character, and with Zbrush's ZSpheres tool, we build a base skeleton with the right proportions to have a first 3D model to work with. *(Figure 6)*

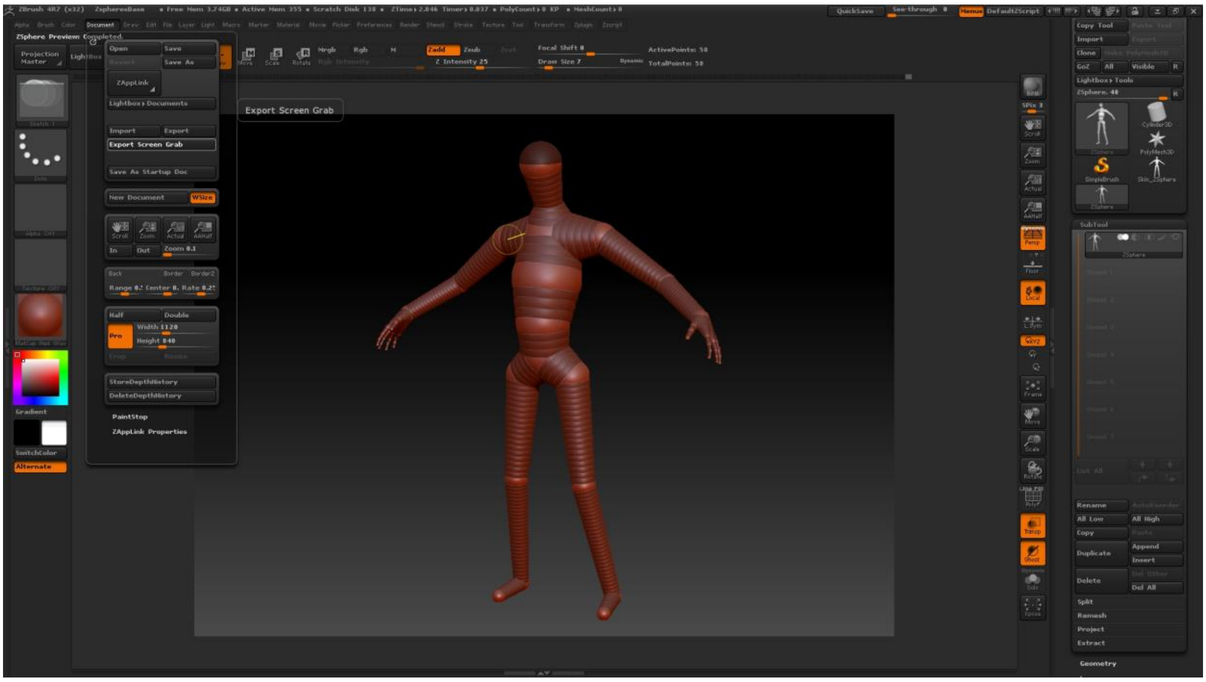


Figure 6: Basic human structure using Zspheres in Zbrush.

Before continuing with the model creation process, the concept of subdivision levels of a polygon mesh should be explained: A subdivision surface is a method of representing a smoothed surface from a simpler polygon mesh. This is represented in the form of different levels, and the difference between two levels of subdivision is the initial number of polygons multiplied by four. In this way, the higher the level of subdivision the more refinement and detail we can have on our surface, but less overall control we have on it.

Therefore, from the base mesh that has been created, levels of subdivision have been added to go more and more detailing the character.
Always bearing in mind that at that moment it is not relevant the number of polygons it has, only its definition and final aspect.

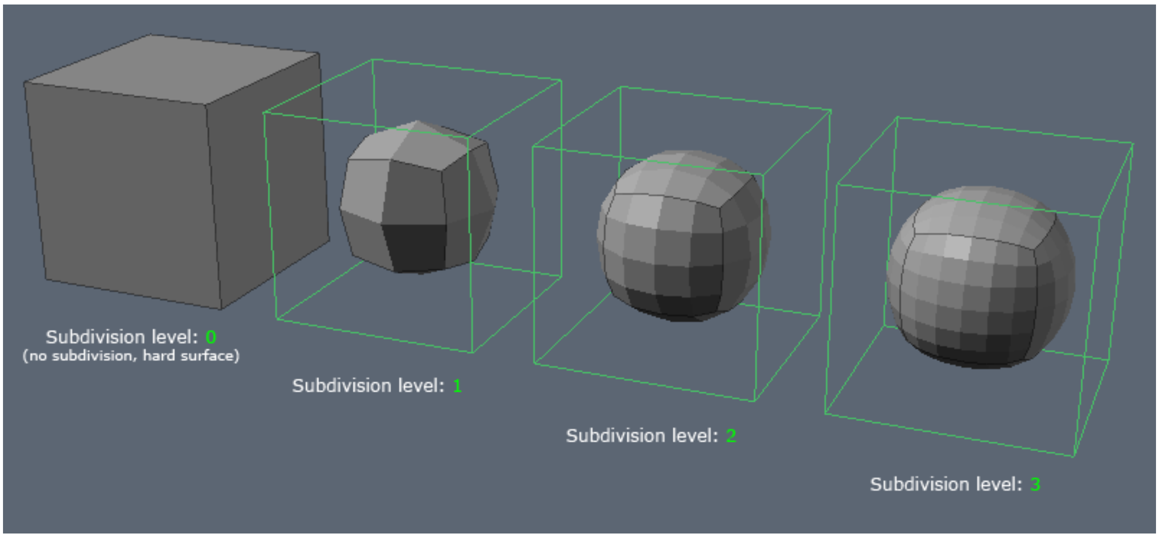


Figure 7: Levels of subdivision

Once the character's model has been refined and has the desired appearance *(Figure 8)*, a process known as "retopology" has to be applied to it. It is a process through which the amount of polygons in the model is reduced to a considerably smaller amount and in a better organized way for later processes such as rigging and animation. This is necessary because the high-resolution model or high-poly mesh that has been made has such a high number of polygons that the current game engines are not able to support, and this way it is not possible to incorporate it into the engine or prepare it for animation.

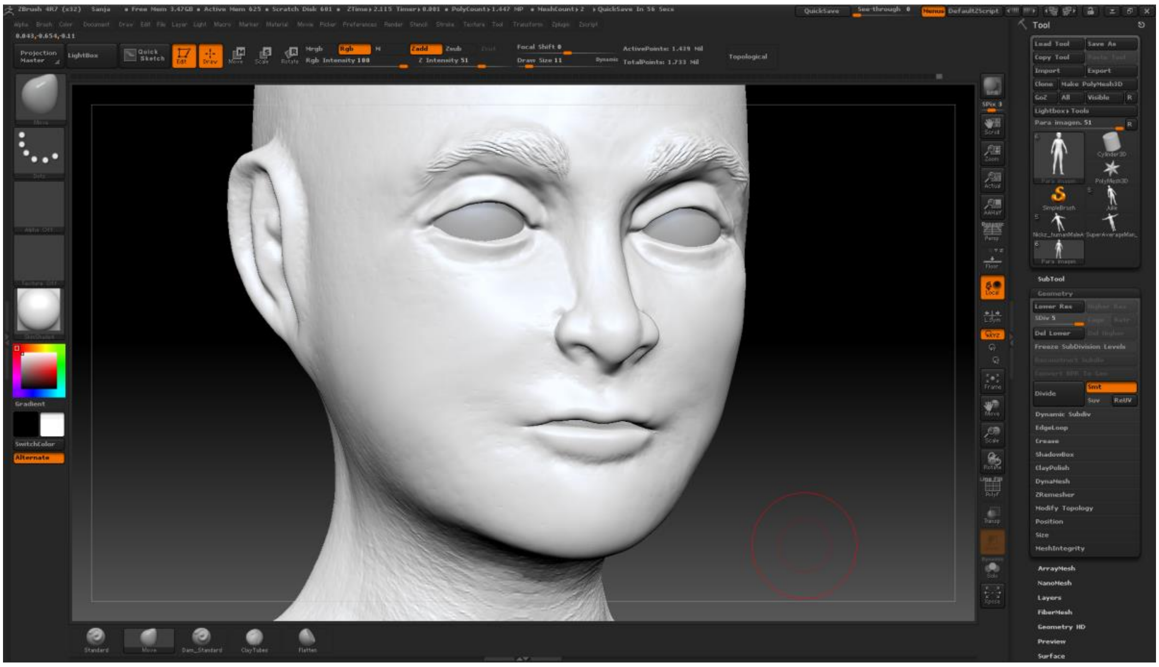


Figure 8: Modeling the details of Sanja's skin

This retopology process can be done manually, creating from 0 the new mesh, or automatically thanks to Zbrush, in this case has been done automatically.

The first thing that has been done is to duplicate the model of the character: one is maintained with high polygonage and the other is applied retopology, so that when this process has been done, through the levels of subdivision we can 'project' the details achieved in the high polygonage model to the low polygonage model.

Once I have the model to which I want to do the retopology, a series of guides, provided by the program, are drawn *(Figure 9).* This guides must be respected by the program (to a certain extent) when creating the new mesh under polygon, and once they have been placed, the program is told to perform the retopology automatically.

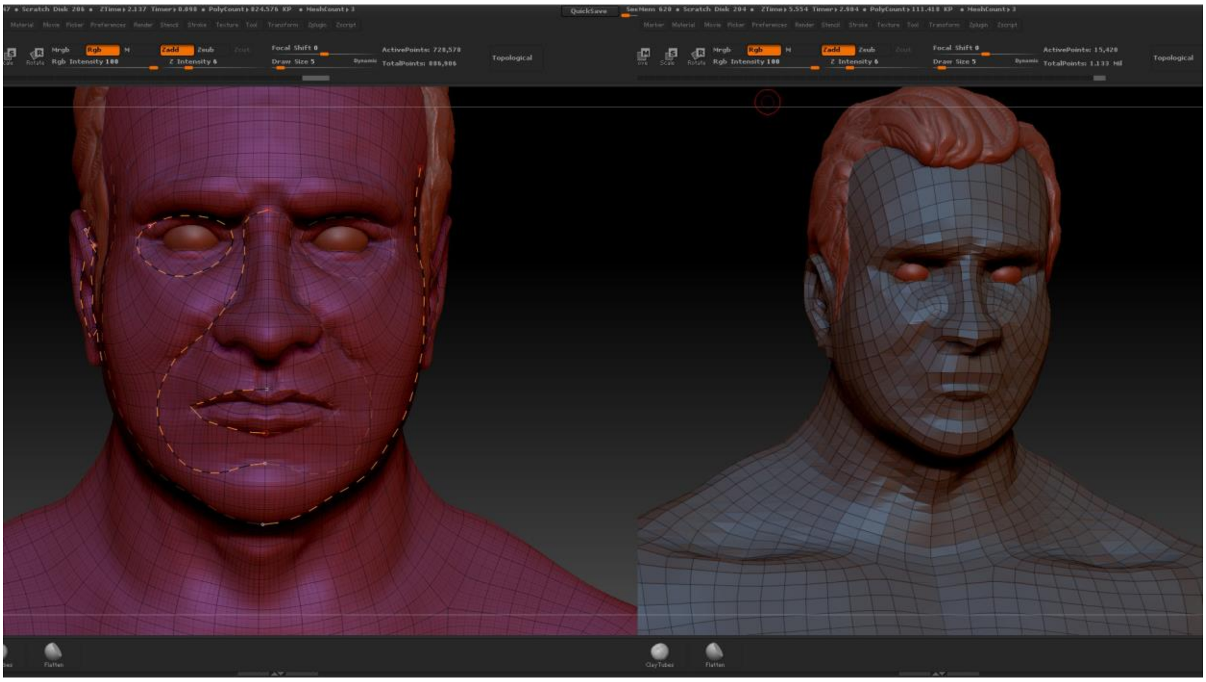

Scratch Disk 206 • ZTime 2.137 Timer 0.098 • PolyCount 824.576 KP • MeshCount 3
Material Movie Picker Preferences Render Stencil Stroke Texture Tool Transform Zplugin Zscript
Scale Rotate Mrgb Rgb M Zadd Zsub Zcut
Rgb Intensity 100 Z Intensity 6
Focal Shift 0 Draw Size 5 Dynamic
ActivePoints: 728,578 TotalPoints: 886,906
Topological
QuickSave
SeeMem 620 • Scratch Disk 204 • ZTime 5.554 Timer 2.984 • PolyCount 111.418 KP • MeshCount 3
Marker Material Movie Picker Preferences Render Stencil Stroke Texture Tool Transform Zplugin Zscript
Move Scale Rotate Mrgb Rgb M Zadd Zsub Zcut
Rgb Intensity 100 Z Intensity 6
Focal Shift 0 Draw Size 5 Dynamic
ActivePoints: 15,420 TotalPoints: 1.133 Mil
Topological
Flatten
ClayTubes
Flatten

Figure 9: Retopology guides and final result in Zbrush

Thanks to this, the polygons of the Sanja model have been reduced from approximately 1, 439,750 to just 5,600. While the Zledic model has been reduced from 728,578 to approximately 11,850. The retopology of the Zledic model has been less exhaustive because its model will hardly be used to be statically located.

Once finished, I took this moment to create the UV map of the character I was modeling (directly from Zbrush), as it was essential for later texturing *(Figure 10).* Automatically then I made the process of projecting details from our high-poly model, to the low-poly model, and finally I created a normal map that gathers all these details and information so that when using the low-poly model in combination with this normal map, has the right appearance.

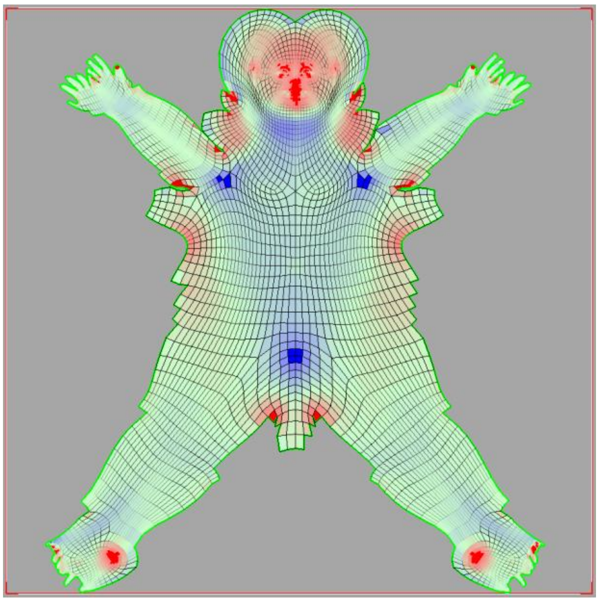


Figure 10: UV map of Sanja's low poly model

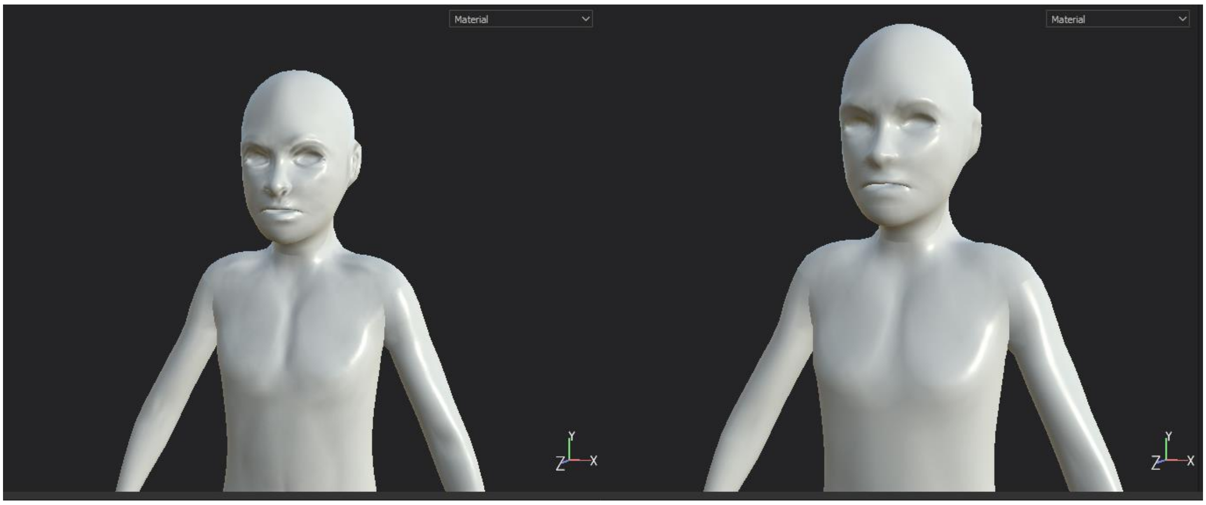


Figure 11: Sanja's low poly model: With and without normal map (5600 polygons each)

4. 1. 2. 3 Hair modeling

On the other hand is the modeling of the hair of the character. It is a very different modeling process and can be done in two different ways: by creating an own model of the hair in the same way as the body, or by using Poly-strips.

The Poly-strips technique was used because the number of polygons needed is much lower, thus improving performance, as well as giving the possibility of applying physics to the hair, making it move and wave with the movement of the character.
Poly-strips are strips of polygons of a single face that cover the entire surface of the hair you want to put, being each of the poly-strips, a lock of hair.

This can be done manually, placing each of the poly-strips in the desired position, shape and length, however, it's a very expensive procces with a poor and artificial result if you dont have experience doing it. *(Figure 12)*

Therefore, in this case I've used the modifier Cloth of 3ds Max, the same one that I've used to apply physics to the clothes of the character. In this way it is applied to each of the poly-strips that make up the hair a force of gravity and attributes that give the sensation of hair much more natural, also giving the possibility of moving along with the head when making the animations of the character. *(Figure 13)*

When the whole surface is covered with the necessary poly-strips, the UV map is made of all the poly-strips together, this is because I wanted there to be an overlap between the textures of each of the tufts of hair to give the feeling of randomness in the hair.

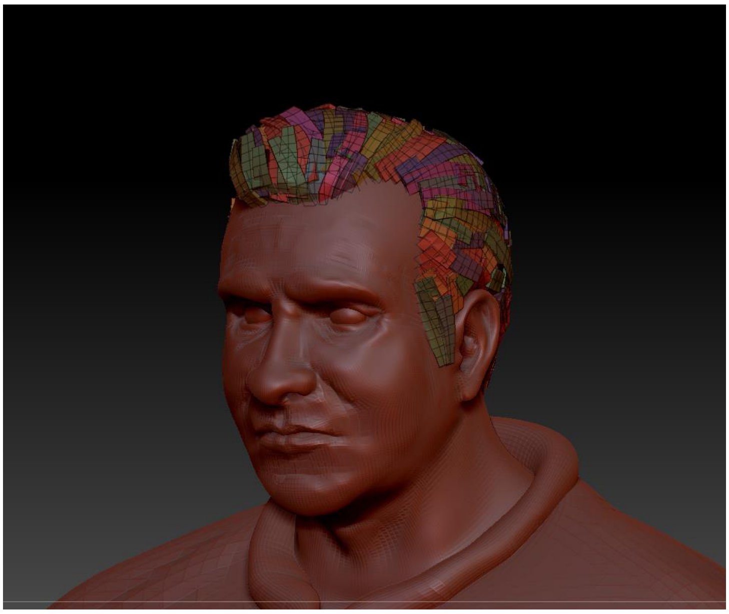
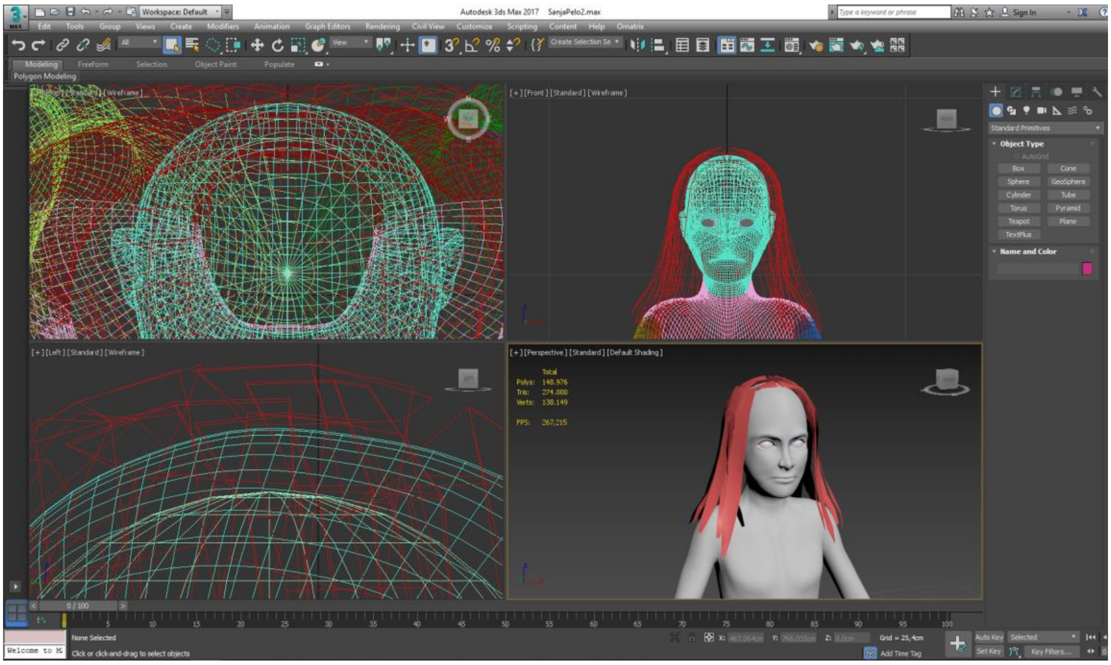


Figure 12: Poly-strips example in Zbrush (done manually)



```
        Total
Polys:  148.976
Tris:   274.000
Verts:  138.149

FPS:    267,215
```

Figure 13: Poly-strips in 3ds Max using Cloth modifier

4. 1. 2. 2 Cloth modeling

With the low-poly model of the body already created, i begun the modeling of the clothes for the character, a process that can be done in many different ways and programs, but in this case Autodesk 3d Studio Max has been used due to the good control over the amount of polygons it provides, as well as the different modifiers of clothes and physics, such as the modifier "Cloth", which provide a textile aspect to the clothes of the character *(Figure 14)*. [4]

Finally, and in the same way as the base body of our character, the corresponding UV map must be created for each of the pieces of clothing that are modeled.

The process of creating the UV maps of the objects has been done thanks to the UVLayout program.

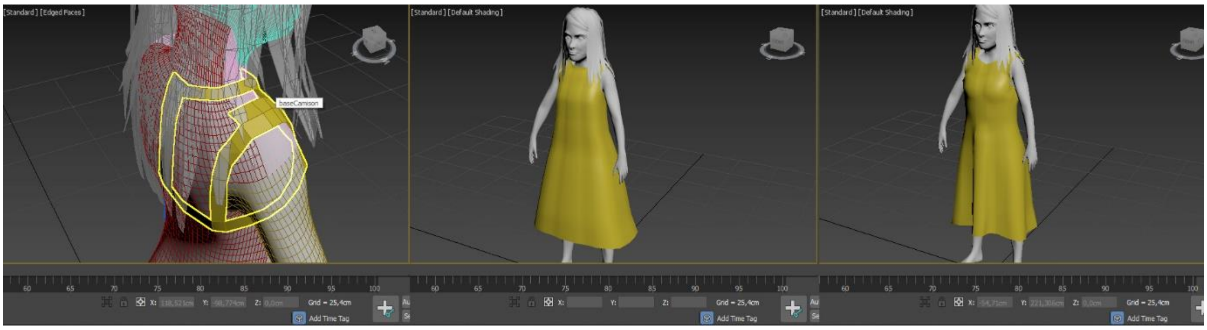


Figure 14: Modeling process of Sanja's dress (basic structure, finished model & with Cloth modifier)

With all this, including Sanja's hair and clothing, the model ends up with a total of 15,800 polygons, a more than reasonable amount to be supported by the Unity games engine and with a considerable result.

4. 1. 3 Texturing

The texturing of each of the parts of the character: body, hair and clothing, is done within the program Substance Painter. [3].

Substance Painter offers a great variety of materials and tools to give a realistic texturing to our objects, besides providing multiple options to export a great amount of texture maps, but in this case and considering that the game engine I'm going to work on is Unity, I'm only interested in four types of texture maps:

- *Albedo-Transparency map:* Represents only the base color of the textured object.
- *Ambient Occlusion map*: pre-computed calculations of ambient light bounce on the surface
- *Metallic Smoothness map:* Describes which parts of a material are metal and which parts are not metal
- *Normal Map:* 3D bumps map that modify Vertex normals to give the appearance of higher detailed geometry.

It’s also needed to include the *Alpha map* for the transparency level in case we are working with any kind of transparent object like crystal or even hair. [6]

Another aspect to take into account when exporting texture maps is the format and compression of them, this is because if you export texture maps in formats with compression such as .jpg, on screen the textures will be displayed correctly and without much loss of information, but once viewed from virtual reality will be completely blurred.

For this reason, to work in virtual reality they must be exported in uncompressed formats such as .bmp or .tiff. Finally it was decided to use the .tiff format due to the size considerably smaller than .bmp and whose visualized in virtual reality is correct.

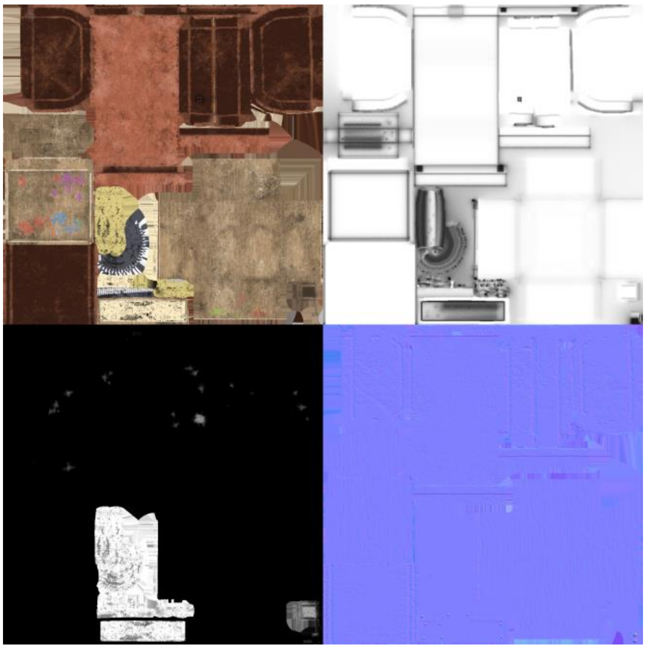
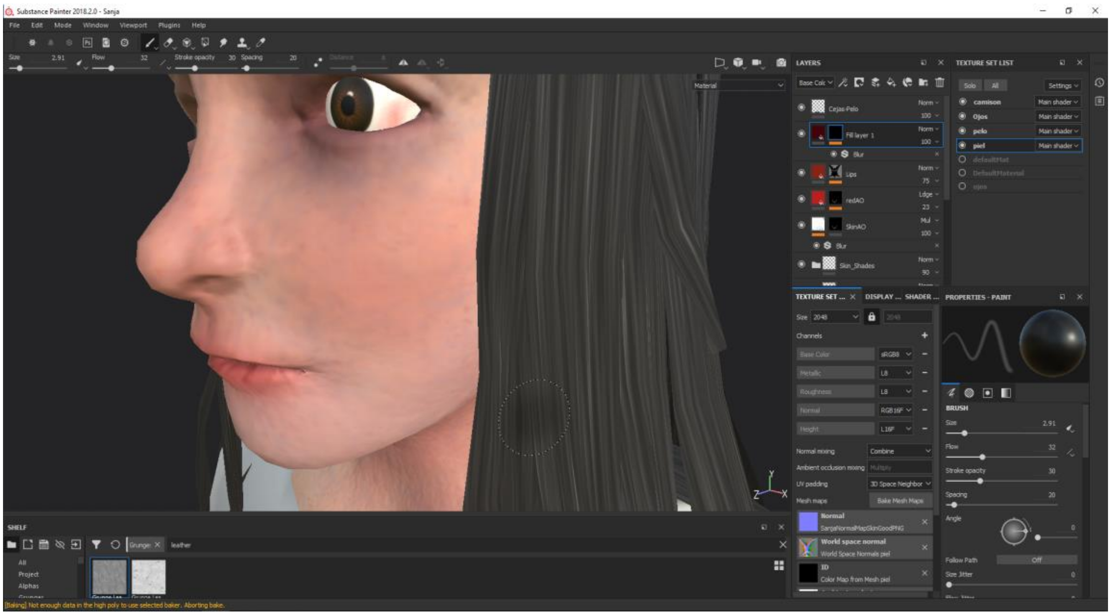


Figure 15: Albedo-transparency, Ambient Occlussion, Metallic and Normal maps



[Baking] Not enough data in the high poly to use selected baker. Aborting bake.

Figure 16: Textured in detail of Sanja's skin

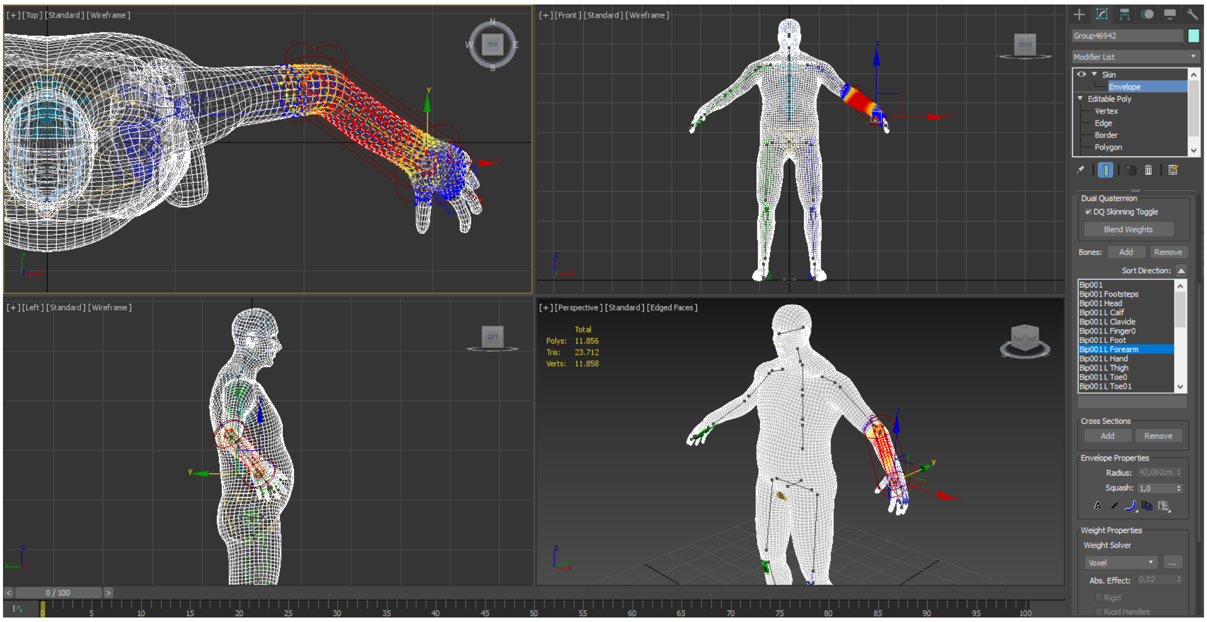
4. 1. 4 Animation

The animation of the character can be done before or after the texturing of the character, but in this case it was done after. This process has been done inside 3d Studio Max thanks to the skeleton tool "Biped".

For this the first step is the creation of a skeleton and its rigging. Rigging is a process through which a series of vertexes of a polygonal surface is associated with each of the bones present in a 3D skeleton, so that when one of these bones moves, all the associated vertexes follow in movement or rotation.

This can be done in two different ways, automatically thanks to free tools and web pages (such as "Mixamo", s.f.-b, [9]) or manually, associating each vertex to each of the necessary bones.
In this project, both techniques have been used, automatically for the Sanja model, and manually with the Zledic model.

In the manual case, the vertices must be associated with each of the bones, and making movement tests and adjusting the associations, checking that they all move in a natural way and there are no deformations. *(Figure 17)*



Total
Polys: 11.856
Tris: 23.712
Verts: 11.858

☑ DQ Skinning Toggle
Radius: 42,062cm
Squash: 1,0
Voxel
Abs. Effect: 0,02

Figure 17: Adjustment of the Biped skeleton with the Zledic model

When the rigging process is done and the model moves properly thanks to the skeleton, animations will be created for the character through keyframes: the key position and rotation of each bone will be defined from time to time, thus creating an animation based on the interpolation of positions and rotations of each bone.

It is important to mention that due to the fact that making animations manually, frame by frame, the result can be very unnatural and fluid, especially in complex animations such as running, free galleries of animations have been accessed to obtain a correct base of movement, and subsequently the key frames necessary to adapt the base of animation to the desired movements have been modified, obtaining a much more realistic result.

4. 1. 5 Import to game engine

When our character is finished, all that's left is its incorporation into the Unity games engine, where all the parts of the process must be joined: incorporating the 3D model (body, clothes and hair), to which the different animations will be added through the Unity Controller animator component, and the textures of each part of our character thanks to the Unity material component, to which these textures will be associated. [5]

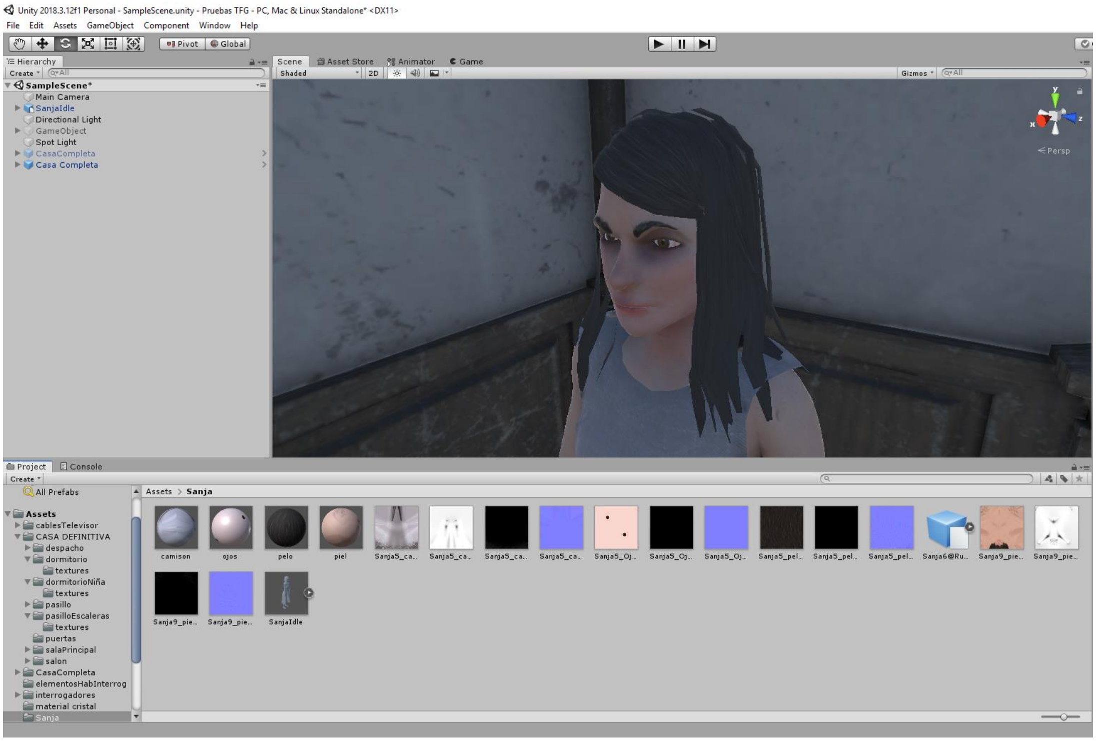
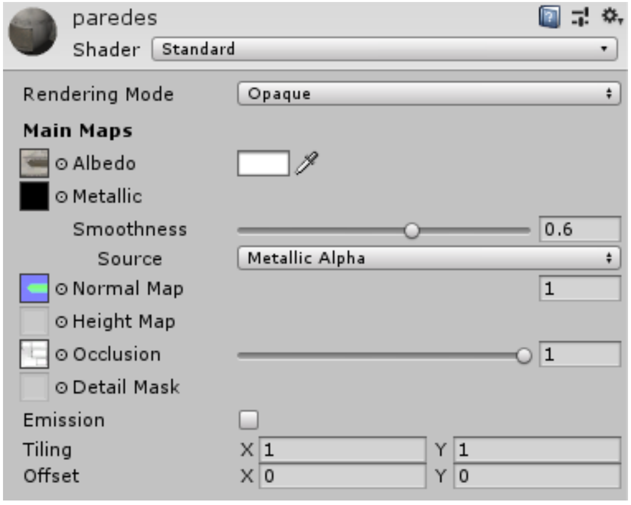


Figure 18: Model and textures imported in Unity



paredes			
Shader	Standard		
Rendering Mode	Opaque		
Main Maps			
Albedo			
Metallic			
Smoothness	0.6		
Source	Metallic Alpha		
Normal Map	1		
Height Map			
Occlusion	1		
Detail Mask			
Emission	□		
Tiling	X 1	Y 1	
Offset	X 0	Y 0	

Figure 19: Material configuration in Unity

4. 2 Game World

The process for creating all the elements that compose the gameworld, rooms through which the player will walk and objects with which to interact and complete the scene, is similar to the process followed to create all the characters as explained above.
The main difference lies in the way 3D modeling is performed, as well as the programs used for it.

The first step, as with the characters, is the search for references, for the genre of terror in which Vornik is included, it was decided to search for closed spaces, with wooden decorations in the house of the protagonist, with great influence of another video game of the genre LAYERS OF FEAR *(Figure 20)* [7], while for the bunker something more conventional and simple was searched for, with few elements in scenes except the essential ones.

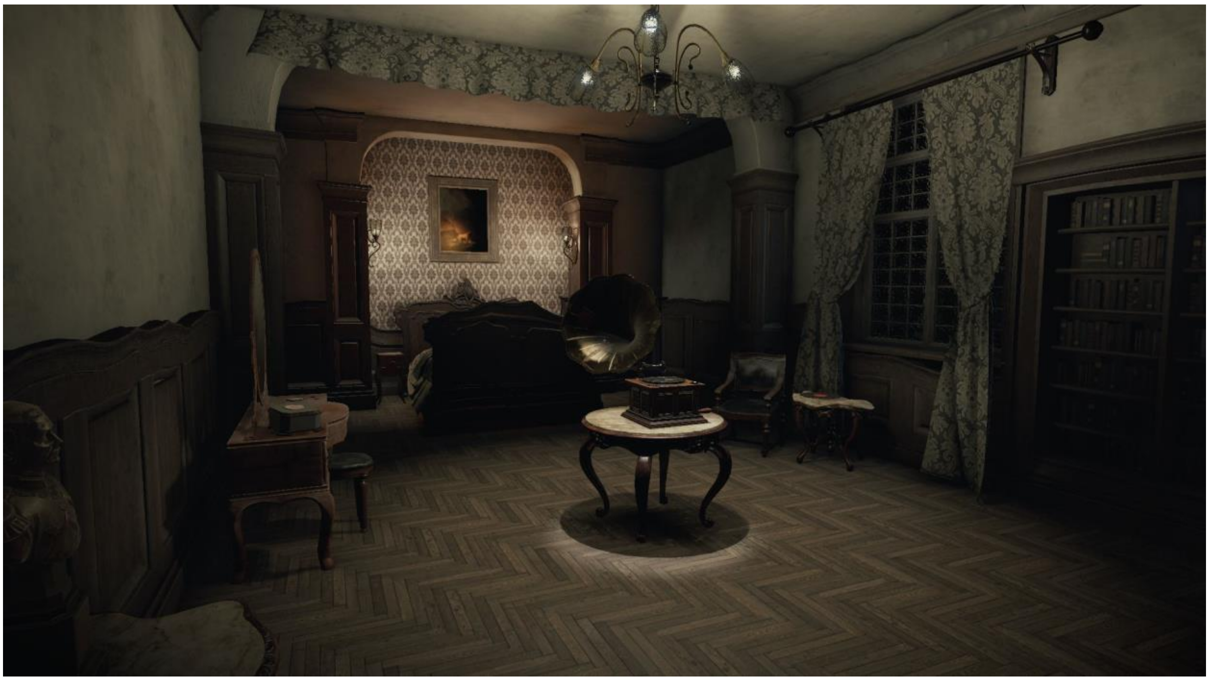


Figure 20: Layers of Fear screenshot, the project's biggest artistic scenario reference

4. 2. 1 Modeling

With these references, and together with the plans of the different areas previously prepared by my colleague Daniel Castro, the 3D model of the basic structure of the house and the bunker was built, following the real measures proposed *(Figure 21).*

However, there was a problem here, and that is that although the measures collected for the width, length and height of the walls of the house were real measures and should work perfectly, when introduced into Virtual Reality, gave the feeling of being smaller than they really were, so they had to be adjusted, making the rooms larger and higher than normal ceilings to give a feeling of being a house with realistic measures.

Once the basic structure of the house and the bunker was done, I began modelling the decorations and elements of each of the rooms present. This whole process was carried out in "Autodesk 3ds Max" *(Figure 22).*

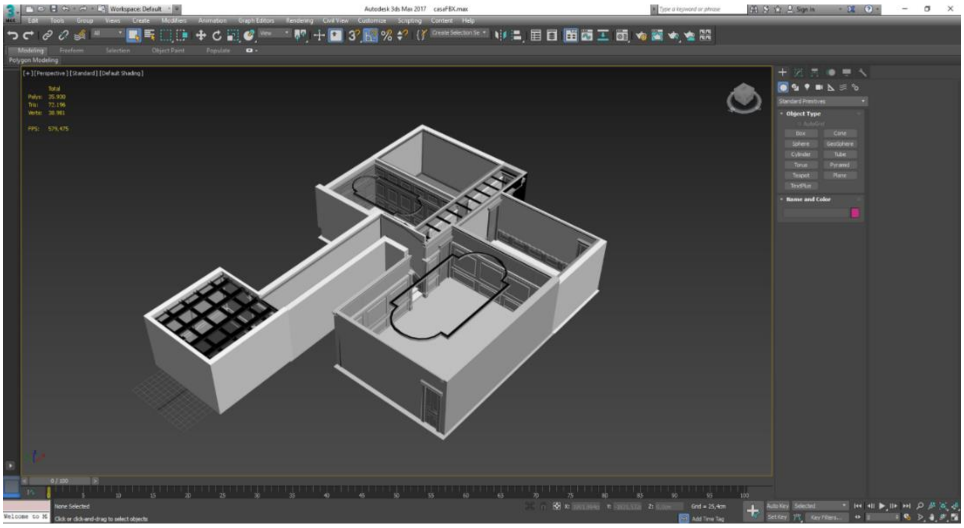

Autodesk 3ds Max 2017 casaFBX.max
[+] [Perspective] [Standard] [Default Shading]
Object Type
Name and Color

Figure 21: House structure in 3ds Max

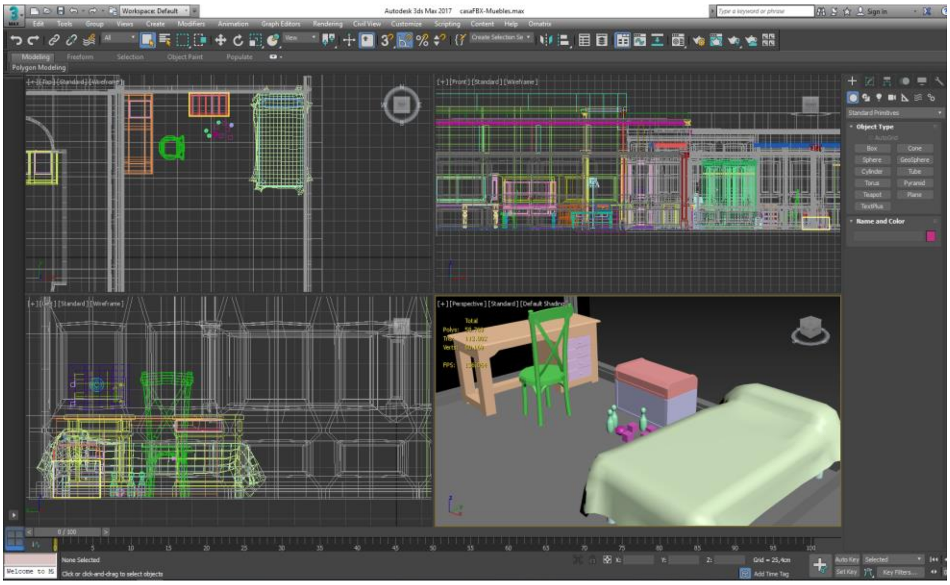

Autodesk 3ds Max 2017 casaFBX-Muebles.max
[+] [Front] [Standard] [Wireframe]
[+] [Perspective] [Standard] [Default Shading]
Object Type
Name and Color

Figure 22: 3D furniture models in 3ds Max

4. 2. 2 Texturing

The process of texturing starts with the creation of UV maps of each of the objects present, and then create different texture maps that will give the proper visual appearance to our objects.

For the different objects that compose each of the rooms, chairs, doors, tables, desks, etc. is a simple process, as each object has its own UV map, however for the house we can not create a unique map that includes all the rooms because the result we get with this is a blurred texture with hardly any definition, so the strategy followed was to divide the house into the different rooms, and in turn divide each room into the different elements that composed it, ceiling and floor, walls, and wood decorations, creating a total of 3 UV maps per room to get the proper resolution. *(Figure 23)*

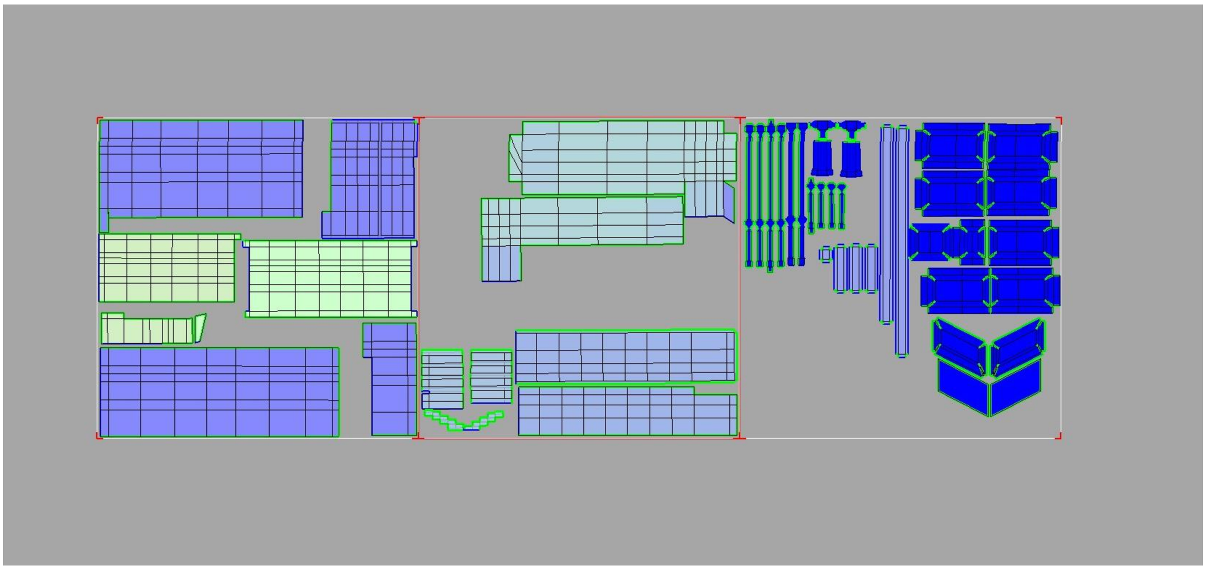


Figure 23: UV maps of the house corridor

All objects with their respective UV maps are then exported to FBX format and reimported into Substance Painter for texturing. *(Figure 24)*

As previously explained in the texturing of characters, we will only be interested in working within the four maps needed for the materials in Unity: Albedo-Transparency map, Ambient Occlusion map, Metallic Smoothness map and Normal Map.

In addition to the appropriate format and compression level as already mentioned for the correct visualization of textures in virtual reality.

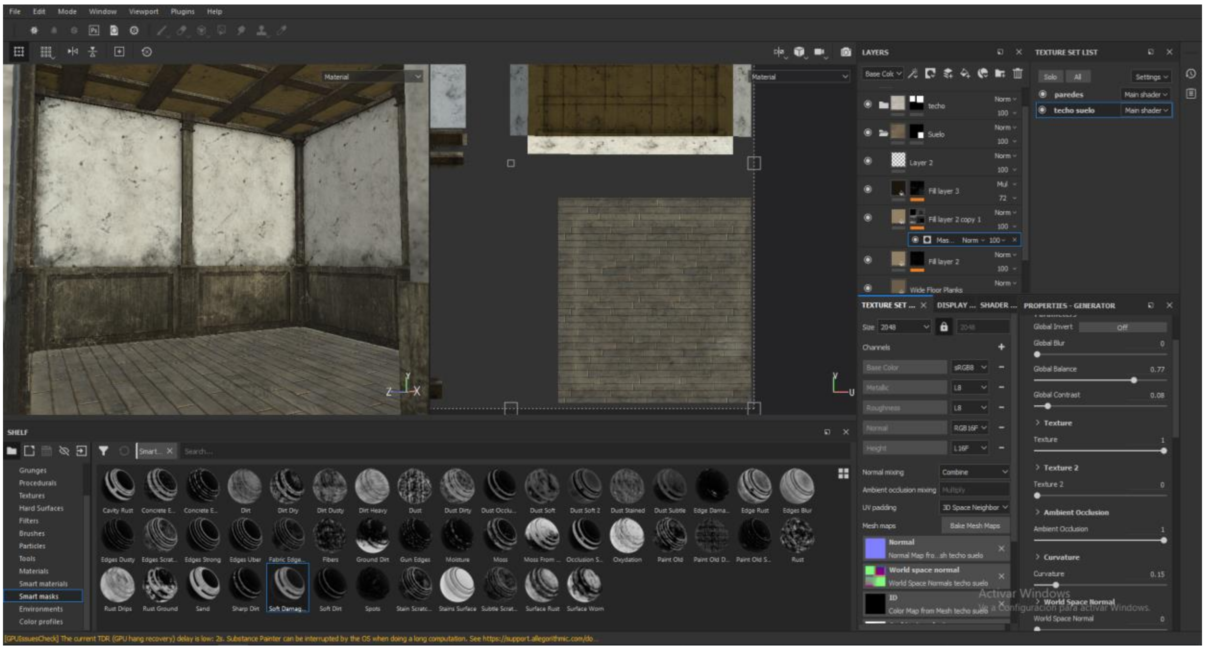


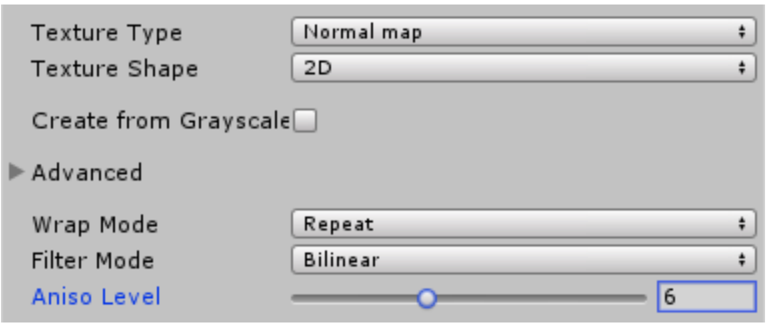
Figure 24: Texturing of a room in Substance Painter

4. 2. 3 Import to Game engine

Finally, all the elements, .fbx files and textures must be imported to the game engine, Unity and in the same way as with the characters, create the necessary materials, colliders and rigid bodies and apply them to the respective objects so that the scenarios are ready.

An important aspect to bear in mind is that for materials created specifically for the floor, an adjustment must be made to its "aniso level" variable in the associated normal map *(Figure 25).*

This variable controls the quality with which this map will be viewed from a low viewing angle, that is, in virtual reality, if this variable is at a low level, when we bend down slightly, the ground will lose quality and at a certain distance will not be seen properly, however if it is at a high level, it will not have this effect and will look perfectly even if we are crouched.



Texture Type: Normal map
Texture Shape: 2D
Create from Grayscale □
▶ Advanced
Wrap Mode: Repeat
Filter Mode: Bilinear
Aniso Level: 6

Figure 25: Aniso Level in Normal maps

4. 3 Particles systems

In this project they have been created through the particle editor that Unity itself provides, and have been used to highlight different objects that are important within the game or to simulate smoke or fog effects, so that they do not affect performance.

For its creation, 2D textures have been drawn in grey scale thanks to Photoshop CC, with the different effects that were necessary to reproduce, and once inside Unity, the necessary variables were adjusted to create the desired effect. *(Figure 26)*

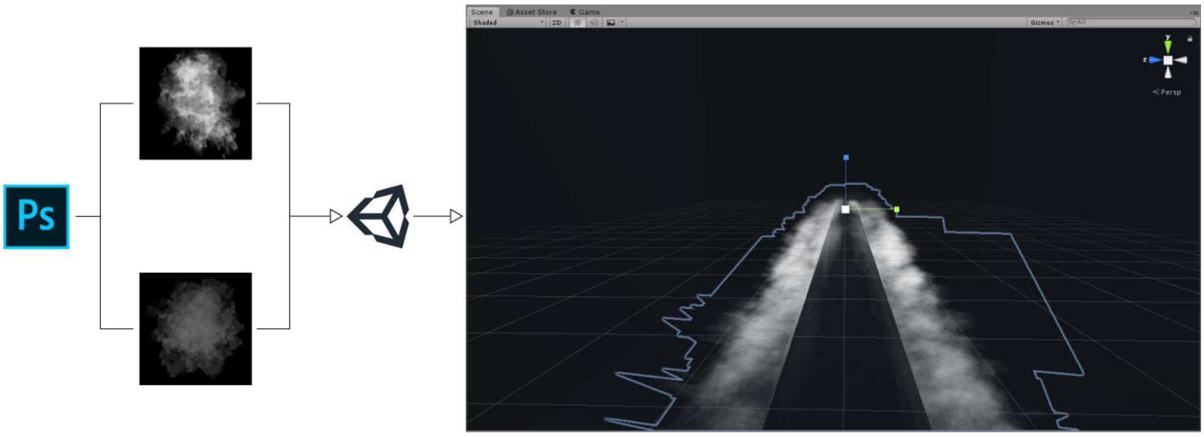


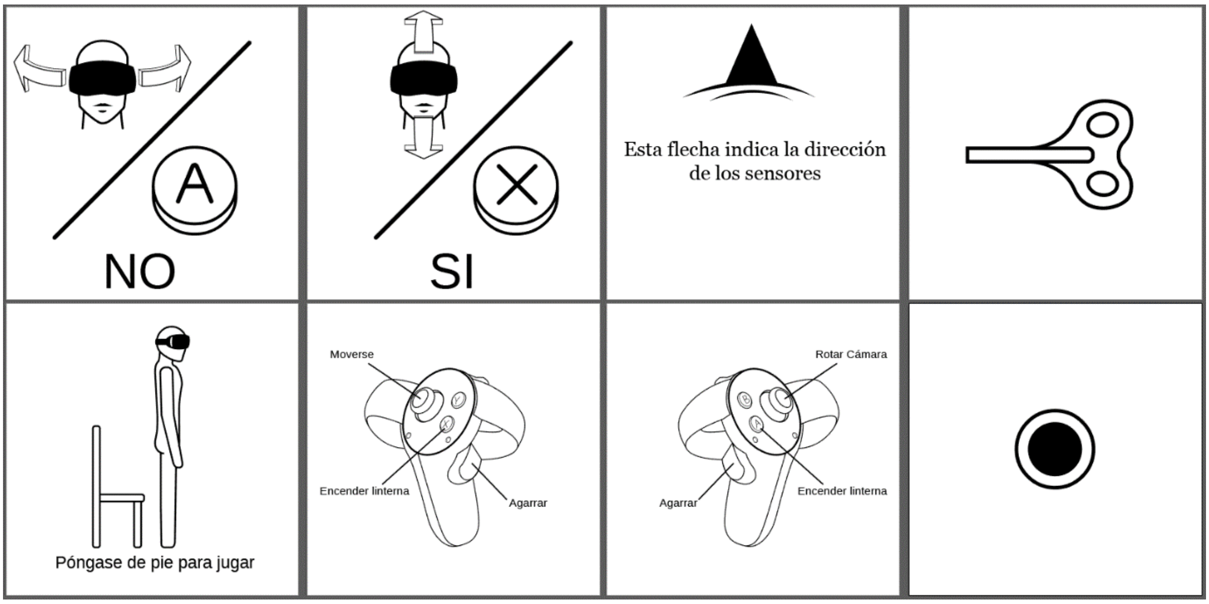
Figure 26: Particle Creation Process Scheme

4. 4 HUD & MENUS:

Within the main menu, the different buttons that can be selected to change the menu and access the different options will be made through text by TextMeshPro. This will be explained later in the technical section.

On the other hand, at the design level, the menu elements, texts and images have been made in two different ways: The buttons are simply selectable text as already mentioned, while the images and icons that appear in the interface, such as those that show the controls or different indications, have been designed thanks to the Adobe Illustrator CC tool *(Figure 27).*

While the player is in the menu he will be located inside the living room of the protagonist's house, being able to look around and observe the scene in which he is going to play, but he will not be able to move or interact beyond choosing the different options of the menu.



A
NO
SI
Esta flecha indica la dirección de los sensores
Póngase de pie para jugar
Moverse
Encender linterna
Agarrar
Rotar Cámara
Agarrar
Encender linterna

Figure 27: UI design in Illustrator

5. Functional & Technical specification

5. 1 Tools

3 MAX	Autodesk 3D Studio Max: 3D animation and graphics creation program developed by Autodesk
ZBRUSH	ZBrush: 3D modelling, sculpture and digital painting software created by Pixologic.
SUBSTANCE PAINTER	Substance Painter: Software for painting materials directly on three-dimensional objects, developed by Allegorithm
Ps	Adobe Photoshop CC: raster graphics editor developed by Adobe Systems Incorporated
UV LAYOUT	UV Layout: is a stand-alone application for the creation and editing of UV texture coordinates for 3D polymeshes and subdivision surfaces, developed by headus.
	Unity: Video Game Engine created by Unity Technologic
Ai	Adobe Illustrator CC: is a vector graphics editor developed and marketed by Adobe Inc.

5. 2 HUD & Menu programming

The TextMesh Pro tool will be used to create text and HUD elements and menus, because TextMesh Pro uses a rendering technique known as Signed Distance Field (SDF), which makes it possible for text to be rendered cleanly at any size and resolution, so that in this case, Virtual Reality, and unlike Unity's default text tool, allows texts and dialogues to be displayed clearly and without problems.

In addition to using a collection of shaders designed to take advantage of the power of SDF rendering, Text Mesh Pro makes it possible to dynamically change the appearance of text by simply changing certain variables to add visual styles such as dilation, contours, shadows, brightness, etc. including the ability to save and load those visual styles by creating/using pre-defined materials.

Inside a Unity canvas object the different text objects necessary have been created, these objects show the different options available from the menu, each of them with a script that indicates its identification number.

5. 2. 1 Raycast

Through a raycast launched from the virtual reality camera it is detected which option the player wants to select, that is to say, the one he is looking at, and when he selects it by pressing the A button of the control, the camera script is in charge of accessing the identification number of the selected text, showing and hiding the corresponding text objects depending on the identifier number.

In addition, each of the text objects has its own sound that is played when they are selected.

```
Update()
{
        Ray ray = new Ray(Camera.main.transform.position, Camera.main.transform.forward);
        // We create a ray that goes from the position of the camera to its positive z-axis.
        RaycastHit rayCastHit;

        If(Physics.Raycast(ray, out rayCastHit, distance, layermask.value)) {
        /*uses Unity's internal physics to check that the ray created from the camera has struck an object with the layer required (in this case "UI") at the set distance and saves it in the collision object rayCastHit.*/
                If(Input.GetButtonDown(acceptButton))
                      rayCastHit.collider.getComponent<UIInteraction>().Press()
```

```
      /* The Press() function of the UI, in this case text, performs the necessary behavior, for example, new game makes a launch of scene 1 of the project (Interrogation I).*/
                      else
                      rayCastHit.collider.getComponent<UIInteraction>().Pointed();
                      /* The text has in its Pointed function, the necessary behavior to signalize the text, bold and increases size.*/
      }
      }
      }
```

5. 3 Dialogue System

In order to manage the dialogue system, a specific script has been created that shows the appropriate lines of text when indicated.

We have a general controller within the scene, which will be in charge of telling our code when to start reading and displaying text.
The code has an array of texts where all the lines of text to show are stored, an array of sounds in which the dubbing of each one of these lines of text is saved, and an int variable that will indicate if the current line of text is interactuable or not.

The controller will indicate to our code that it reads the following line, this through a corrutine will show the line of text by an object of the canvas, activate the object of sound with the corresponding audio and it reproduces this audio.

Then wait for the audio to finish, so that the text appears on the screen the same time it takes the sound to be heard, add a unit to the counter so that the next line to be shown is the corresponding one and check if the text is interactive or not.

If it is, the controller calls another script developed by my colleague Antoni Arlandis to provide an answer. When he receives it, he will tell the dialogue script to continue reading and showing the corresponding lines of text depending on the answer obtained.

The following code is accompanied by comments to support the explanation:

```
using TMPro;
/*Necessary library for the objects of textmesh pro, for the classic tool of Unity the library would be :
using UnityEngine.UI;*/

public class SubtitleControl : MonoBehaviour
{
    [SerializeField] private AnswerWithHead answerScript;
/* Reference to a code developed by my colleague Antoni, to answer yes or no by moving the head or using the buttons.*/
    [Header("Entradas para la configuración de los subtitulos")]
          [Tooltip("Array de strings para modificar de forma automática")]
          public string[] texts;  //All dialogues ordered
          [Tooltip("Array de audios para la modificación de forma automática")]
          public AudioClip[] clips; //All audios ordered
          public int interactive;
          public bool action = false;

          public TextMeshProUGUI text;
          /* We create an object of type Text of text mesh pro, the normal case would be:
              public Text text;*/

          public AudioSource audioSource; //Audio output for subtitles
          int flag; //General counter

          ControllerManager controller; //reference to scene controller


          private void Awake()
          {
              controller = GameObject.Find("Controlador").GetComponent<ControllerManager>();
          }

          void Start()
          {
              text.text = "";
              flag = 0;
          }



          public void Answer(bool answer)
          {
              flag += answer ? 0 : 1;
              StartCoroutine(nameof(ReadLineSound));
          }

          public void ReadLine()
      //La primera vez se llama a esta function desde el controlador de escena, luego es recursivo
       {
              if (flag == interactive) // If the text is interactive, the script to answer is enabled
              {
                  answerScript.enabled = true;
              }//Trigger interaction
              else if (flag < texts.Length)
                  StartCoroutine(ReadLineSound()); // Call corroutine
              else
              {
                  controller.ActivateTransition();
```

```
                // Finish the texts and tell the controller to finish the scene.
                text.text = "";
            }


        }

        IEnumerator ReadLineSound()
        {
            text.text = texts[flag]; // Place the corresponding text
            audioSource.clip = clips[flag]; // Place the corresponding audio
            audioSource.Play();//Activates the sound
            yield return new WaitForSeconds(clips[flag].length); // Wait till the audio's over.
            flag++;
            ReadLine(); // Read dialogue again
        }
    }

```

5. 4 Dynamic Textures

At certain moments of the game it is necessary to change material textures in Unity to modify the visual appearance of objects. This can be done by creating a different material and changing it when necessary, however this causes more textures than necessary, increasing unnecessarily the size of the project.

Therefore, the change of only one texture map of the corresponding material, the "AlbedoTransparency map", will be made to avoid this increase in size.

The first thing we have to take into account is that Unity materials are shaders, so when the code created to make this change of textures is called in the necessary moment, it must access the renderer component of the object in which it is.

Then the keyword needed to access the texture you want to change, "albedo_Transparency" or "main Texture" will be enabled, and finally, the texture is changed to the new one to be incorporated.

Code:

```
public Texture mainTexture, newTexture; //The initial and new texture.
public Material materialWall; // corridor wall material
Start()
{
        materialWall.SetTexture("_MainTex", mainTexture);
}

Public void newTexture()
{
              materialWall.SetTexture("_MainTex", newTexture);
      }
```

6. Results

- It has been achieved the creation of characters based on features given with photo realistic look for a virtual reality environment

- Creation of a series of textures with a realistic look and good definition when viewed from the reality glasses Virtual Oculus Rift and optimized.

- Design and realization of an entire realistic environment, including 3D models, textures and particle systems for a horror videogame in virtual reality, with a remarkable finish, visualization and optimization.

- Programming of a dialog system, as well as menus and HUD taking advantage of the Virtual Reality environment.

6. 1 Images of Final Result

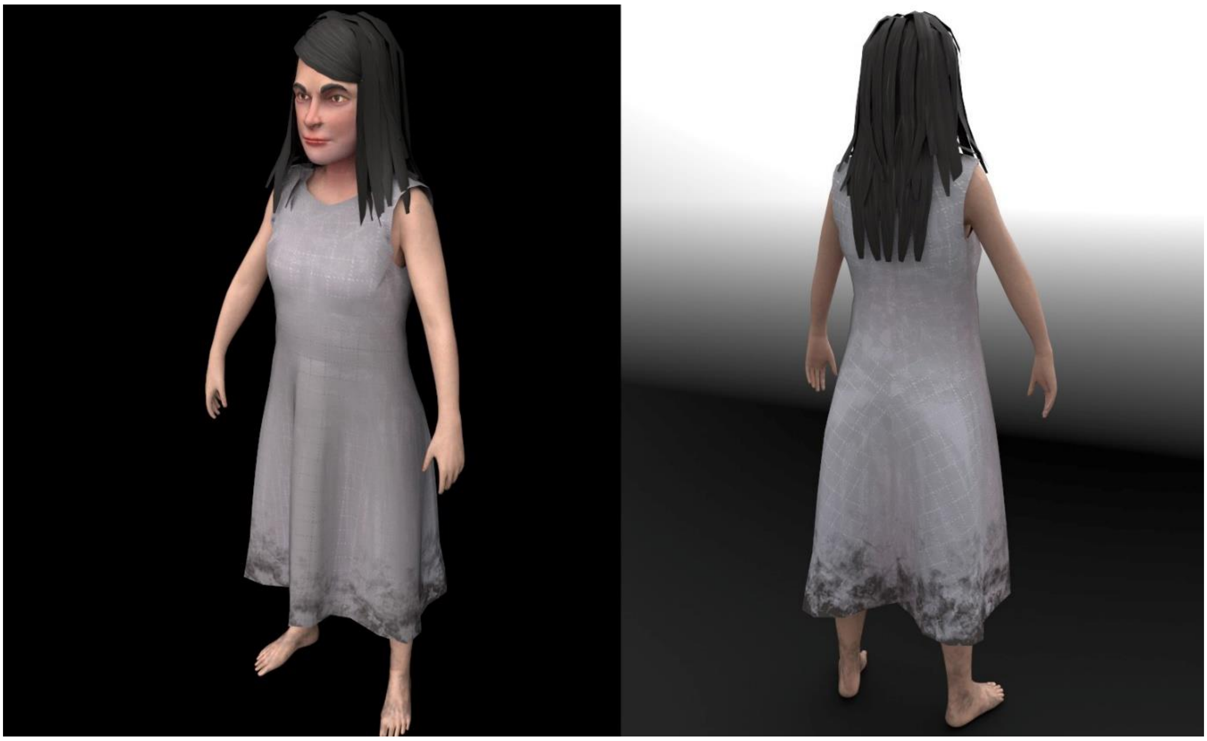


Figure 28: Sanja's final render, front & back

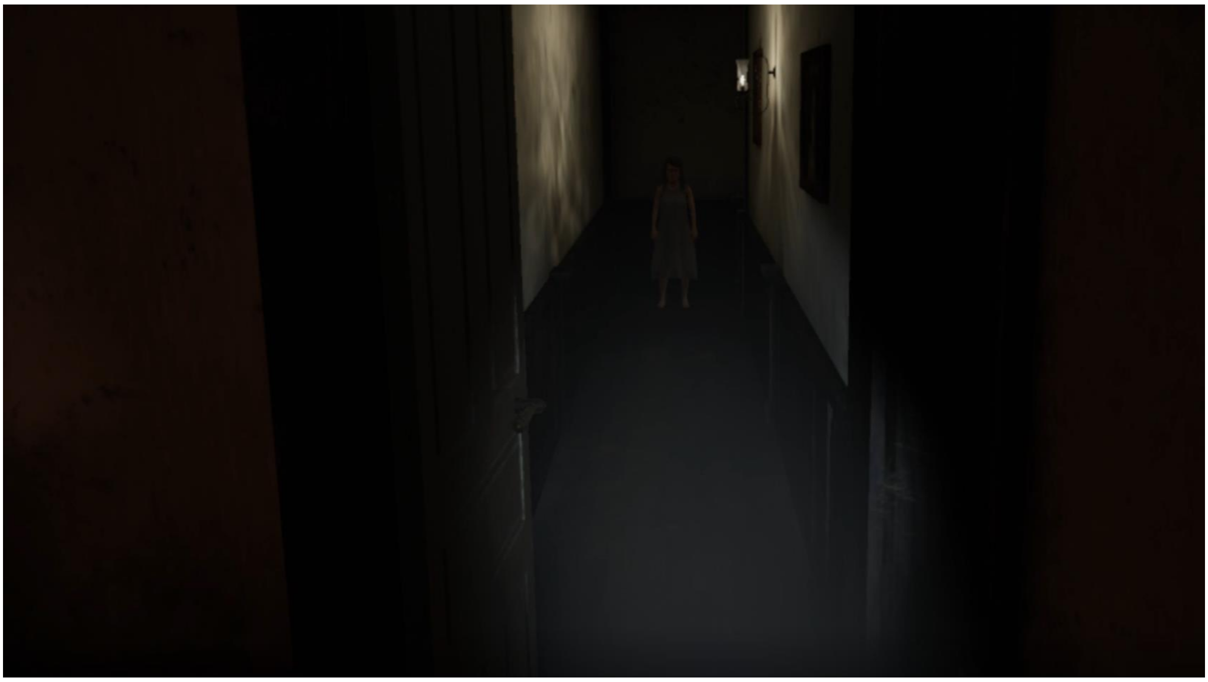


Figure 29: Sanja's in-game screenshots: Corridor

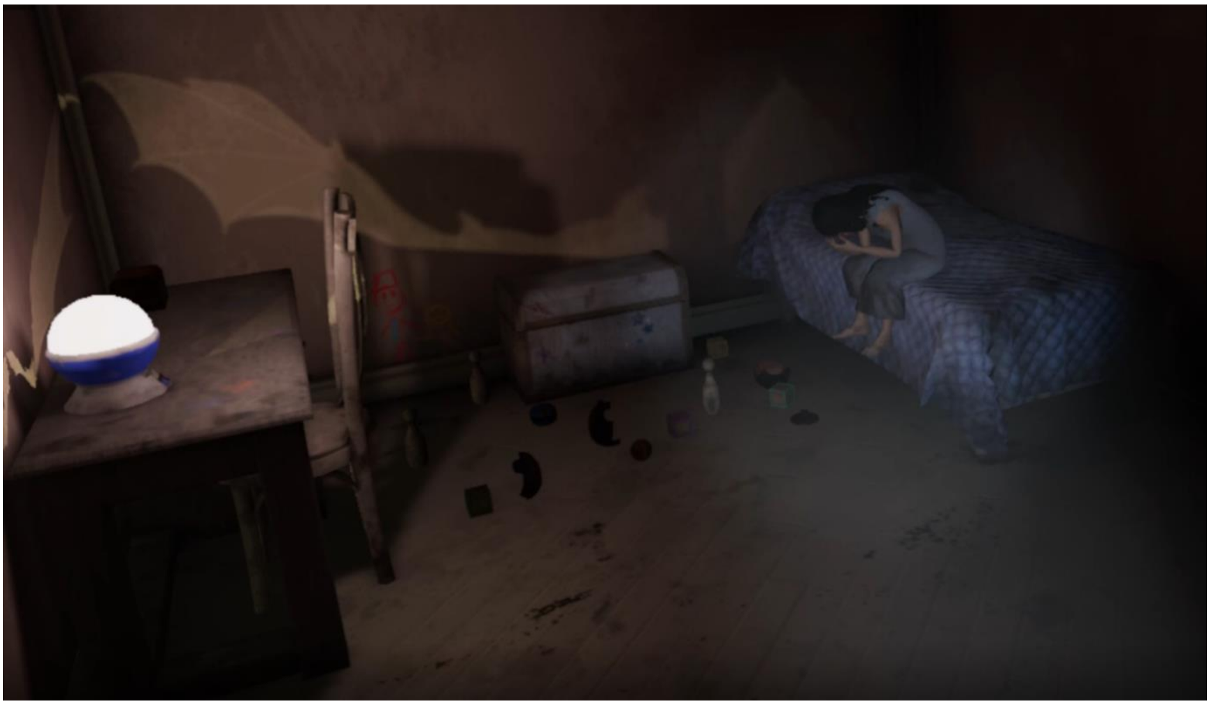


Figure 30: Sanja's in-game screenshots: Sanja's room

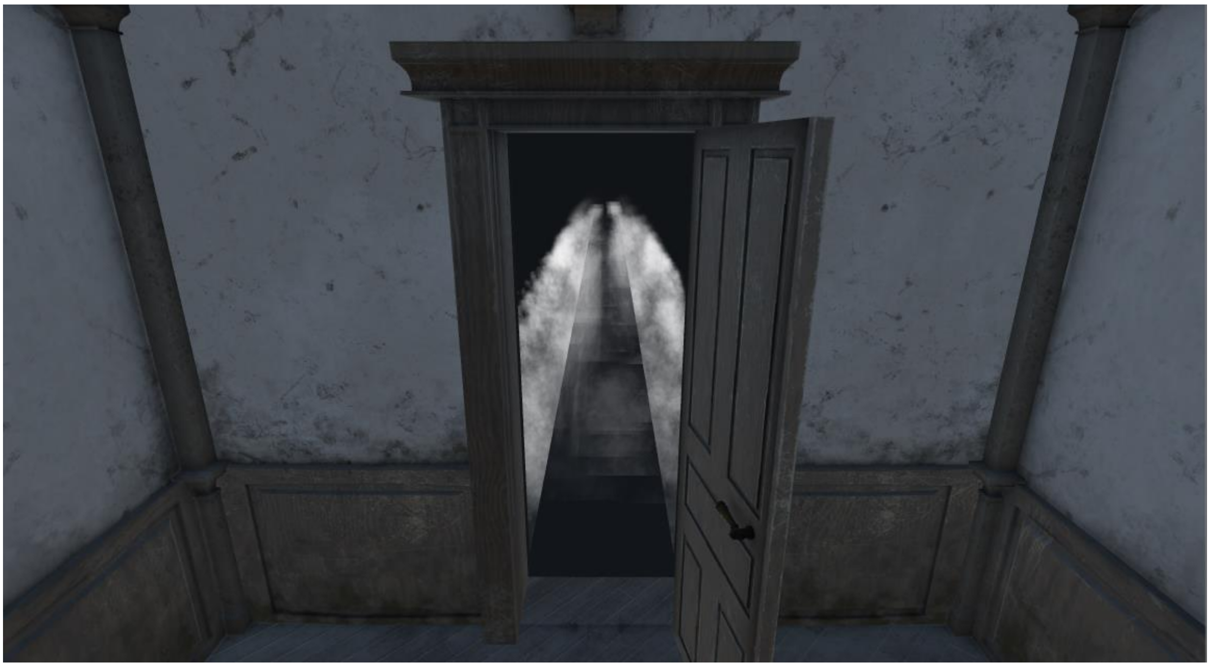


Figure 31: mist particles result

6. 2 Project Access

- Github: https://github.com/danicas92/Vornik

- Executable: https://drive.google.com/file/d/1Zm0iAGuxle1NxKVKEG410bIvxQBCgSZa/view?usp=sharing

- Youtube Video: https://www.youtube.com/watch?v=NA_bOkD3Fvc&t=2s

7. Conclusions & Future Work

Vornik is an important project and the biggest one carried out to date by me, it has been a challenge in multiple aspects due to the novelty of working in virtual reality, but without a doubt it has been a project that has allowed me to put into practice all the knowledge acquired during the years of career, as well as to expand them to a large extent, especially at the level of design and technical art without underestimating the programming. Together with everything else, I am very happy with the results obtained, as well as with the great collaboration and atmosphere between the different members of the team.

Vornik already has the lines of work marked, just need to continue in the same way to get to finish the experience completely, so from the team we have decided to continue the project to get a final version.

8. References

[1] Unity Technologies. (2019) Oculus Unity Getting Started Guide. *Retrieved from* https://developer.oculus.com/documentation/unity/latest/

[2] Pixologic. (2019). Zbrush Documentation User Guide. *Retrieved from* http://docs.pixologic.com/user-guide/

[3] Allegorithmic. (2019) Substance Painter DOC<. *Retrieved from* https://support.allegorithmic.com/documentation/spdoc/substance-painter-20316164.html

[4] Autodesk. (2017). Autodesk 3DS Max 2017 HELP. *Retrieved from* http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Tutorials/

[5] Unity Technologies. (2019). Unity User Manual (2019.1). *Retrieved from* https://docs.unity3d.com/Manual/index.html

[6] Polycount Wiki. (2015, May 24). Texure types. *Retrieved from* http://wiki.polycount.com/wiki/Texture_Types

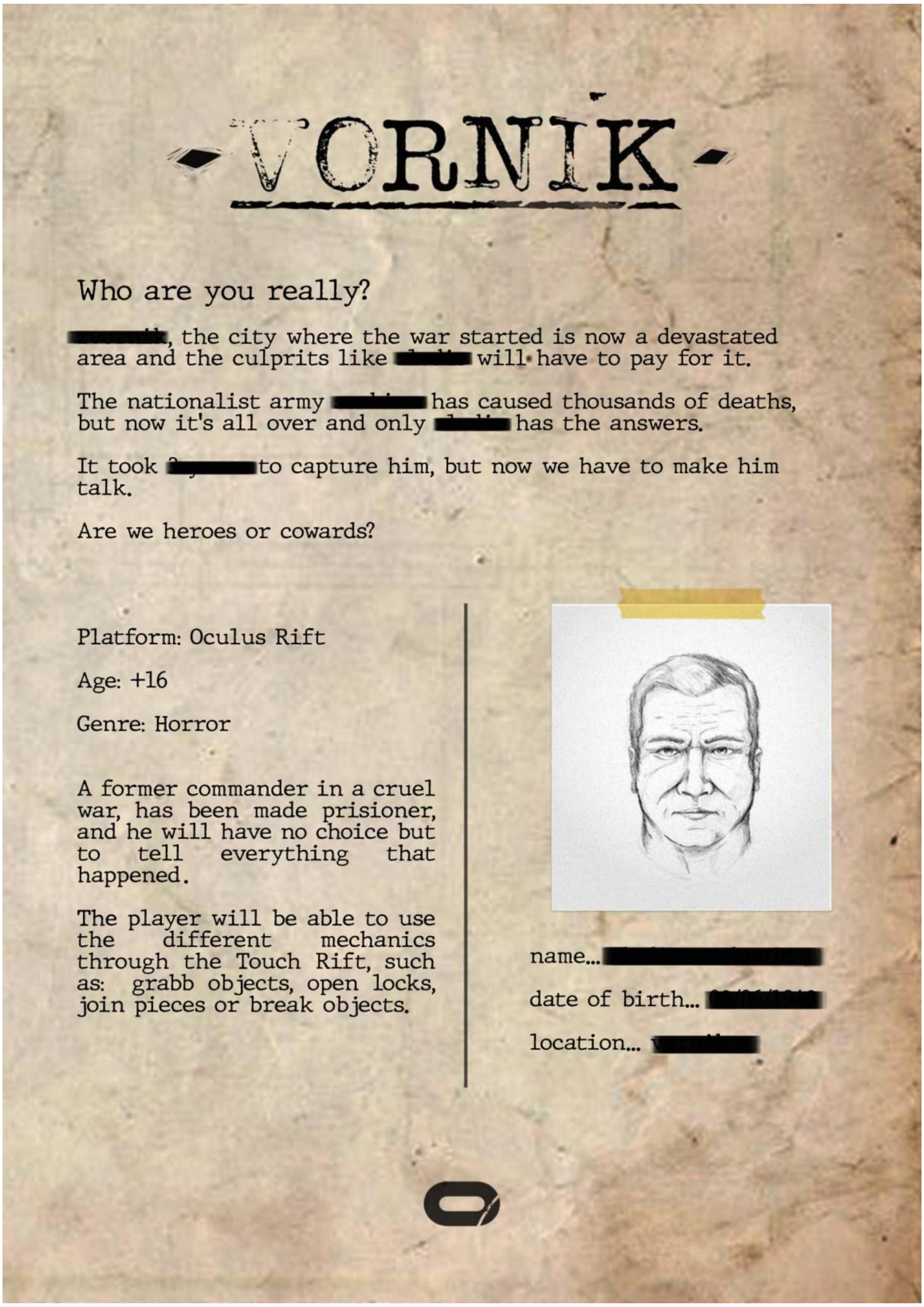
[7] Bloober Team. Layers of Fear – A Psychedelic Horror Game. *Retrieved from* https://layersoffear.com/main.html

[8] CAPCOM. RESIDENT EVIL 7 biohazard. *Retrieved from* http://residentevil7.com/es/#_top

[9] Mixamo. (n.d.). *Retrieved from* https://www.mixamo.com/#/

10. Appendix

10.1 One sheet:



VORNIK

Who are you really?

[redacted], the city where the war started is now a devastated area and the culprits like [redacted] will have to pay for it.

The nationalist army [redacted] has caused thousands of deaths, but now it's all over and only [redacted] has the answers.

It took [redacted] to capture him, but now we have to make him talk.

Are we heroes or cowards?

Platform: Oculus Rift

Age: +16

Genre: Horror

A former commander in a cruel war, has been made prisioner, and he will have no choice but to tell everything that happened.

The player will be able to use the different mechanics through the Touch Rift, such as: grabb objects, open locks, join pieces or break objects.

name... [redacted]

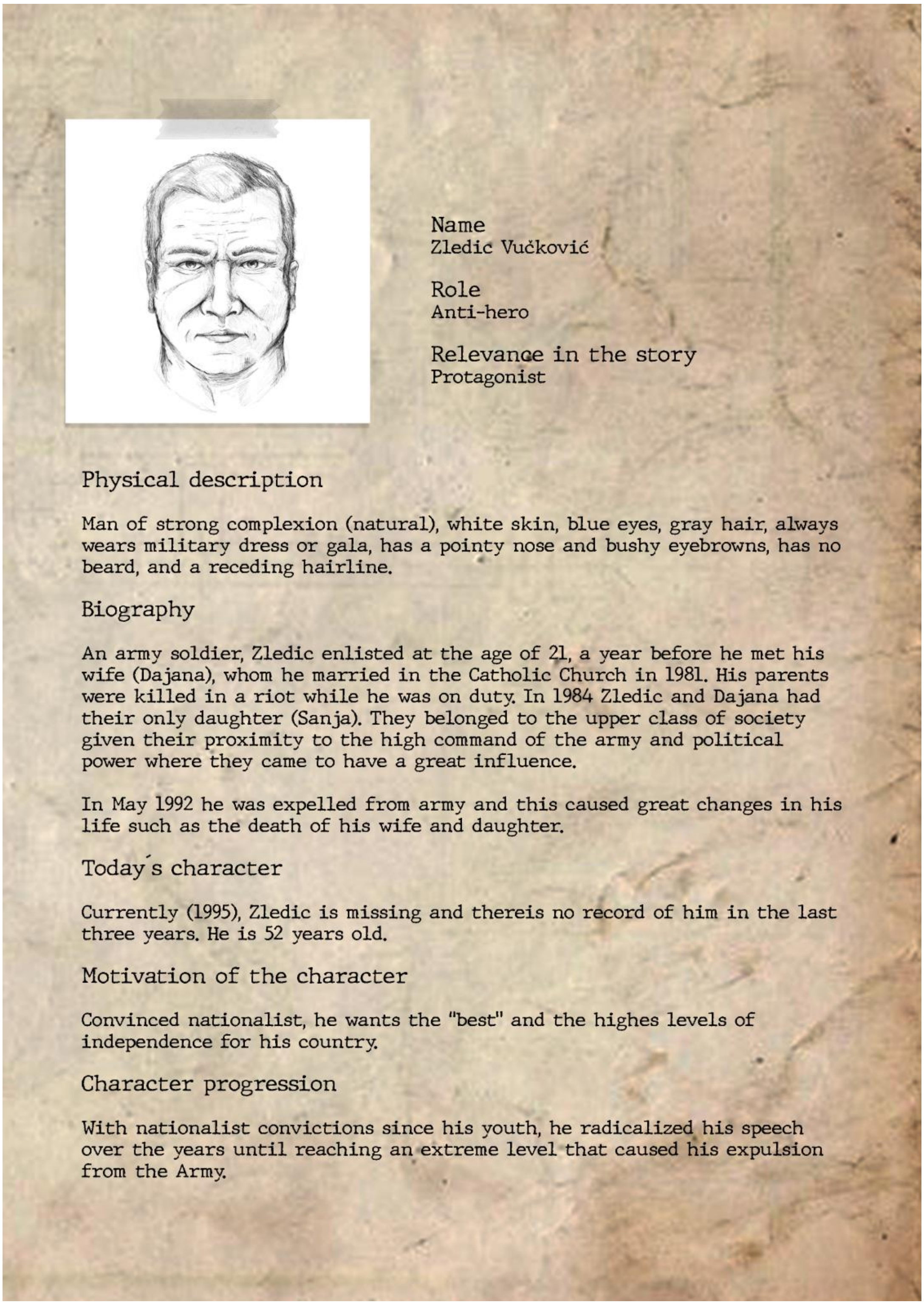
date of birth... [redacted]

location... [redacted]

[1]

10.2 Character’s files:

Character’s files made to show the physical and psychological characteristics of each one of them.



Name
Zledic Vučković

Role
Anti-hero

Relevance in the story
Protagonist

Physical description

Man of strong complexion (natural), white skin, blue eyes, gray hair, always wears military dress or gala, has a pointy nose and bushy eyebrowns, has no beard, and a receding hairline.

Biography

An army soldier, Zledic enlisted at the age of 21, a year before he met his wife (Dajana), whom he married in the Catholic Church in 1981. His parents were killed in a riot while he was on duty. In 1984 Zledic and Dajana had their only daughter (Sanja). They belonged to the upper class of society given their proximity to the high command of the army and political power where they came to have a great influence.

In May 1992 he was expelled from army and this caused great changes in his life such as the death of his wife and daughter.

Today´s character

Currently (1995), Zledic is missing and thereis no record of him in the last three years. He is 52 years old.

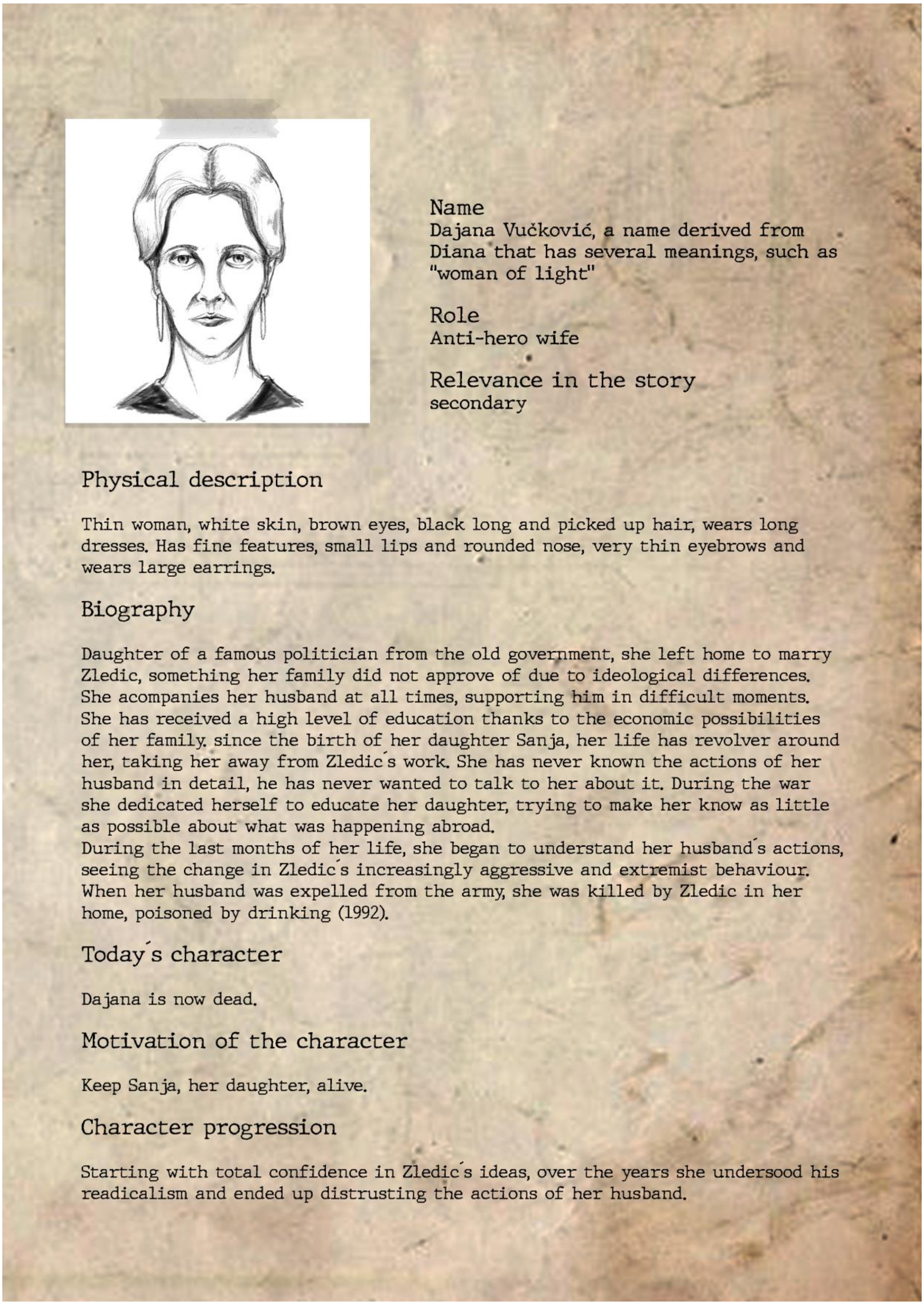
Motivation of the character

Convinced nationalist, he wants the "best" and the highes levels of independence for his country.

Character progression

With nationalist convictions since his youth, he radicalized his speech over the years until reaching an extreme level that caused his expulsion from the Army.

[2] Zledic’s file



Name

Dajana Vučković, a name derived from Diana that has several meanings, such as "woman of light"

Role

Anti-hero wife

Relevance in the story

secondary

Physical description

Thin woman, white skin, brown eyes, black long and picked up hair, wears long dresses. Has fine features, small lips and rounded nose, very thin eyebrows and wears large earrings.

Biography

Daughter of a famous politician from the old government, she left home to marry Zledic, something her family did not approve of due to ideological differences. She acompanies her husband at all times, supporting him in difficult moments. She has received a high level of education thanks to the economic possibilities of her family. since the birth of her daughter Sanja, her life has revolver around her, taking her away from Zledic´s work. She has never known the actions of her husband in detail, he has never wanted to talk to her about it. During the war she dedicated herself to educate her daughter, trying to make her know as little as possible about what was happening abroad.

During the last months of her life, she began to understand her husband´s actions, seeing the change in Zledic´s increasingly aggressive and extremist behaviour. When her husband was expelled from the army, she was killed by Zledic in her home, poisoned by drinking (1992).

Today´s character

Dajana is now dead.

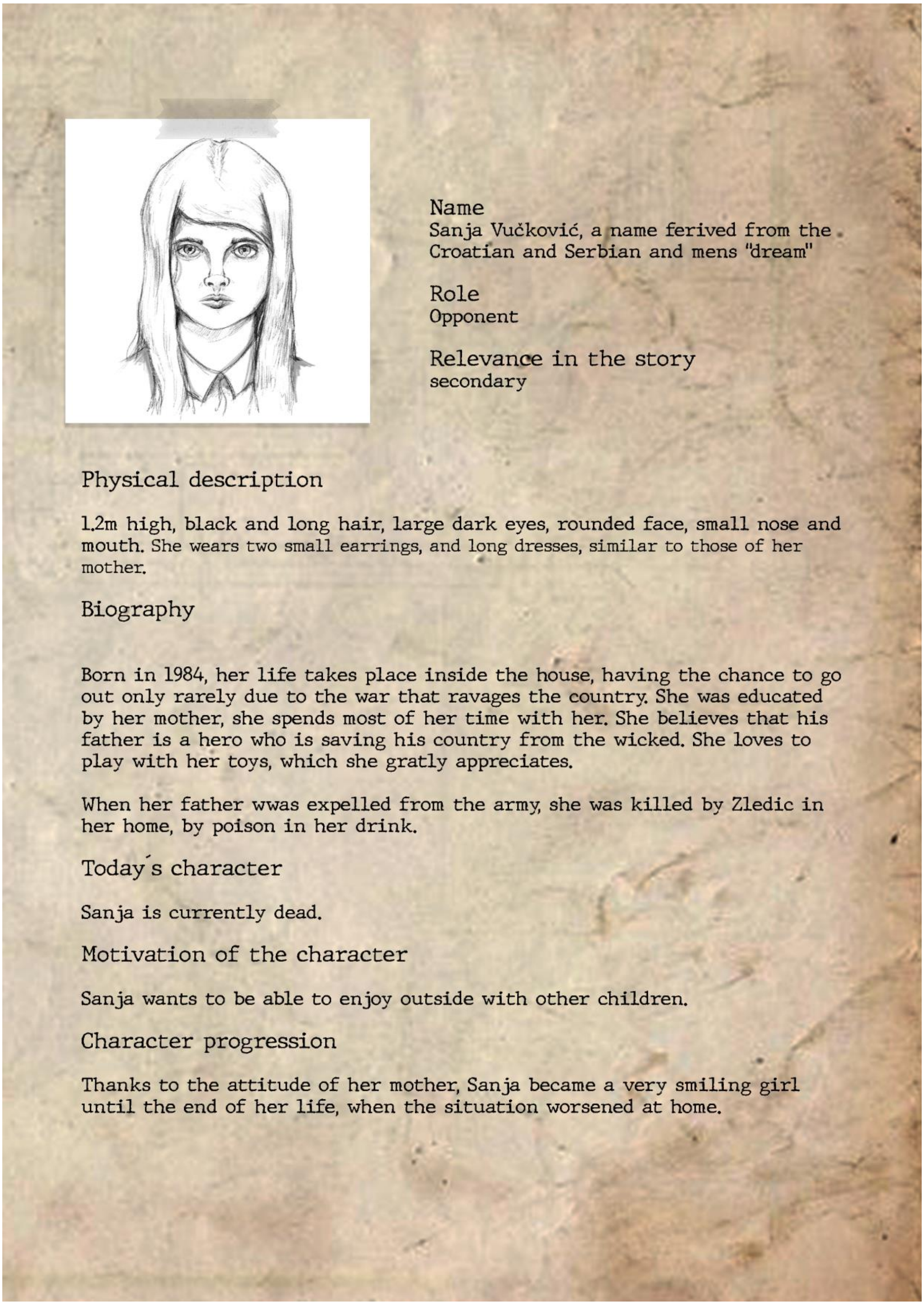
Motivation of the character

Keep Sanja, her daughter, alive.

Character progression

Starting with total confidence in Zledic´s ideas, over the years she undersood his readicalism and ended up distrusting the actions of her husband.

[3] Dajana’s file



Name

Sanja Vučković, a name ferived from the Croatian and Serbian and mens "dream"

Role

Opponent

Relevance in the story

secondary

Physical description

1.2m high, black and long hair, large dark eyes, rounded face, small nose and mouth. She wears two small earrings, and long dresses, similar to those of her mother.

Biography

Born in 1984, her life takes place inside the house, having the chance to go out only rarely due to the war that ravages the country. She was educated by her mother, she spends most of her time with her. She believes that his father is a hero who is saving his country from the wicked. She loves to play with her toys, which she gratly appreciates.

When her father wwas expelled from the army, she was killed by Zledic in her home, by poison in her drink.

Today´s character

Sanja is currently dead.

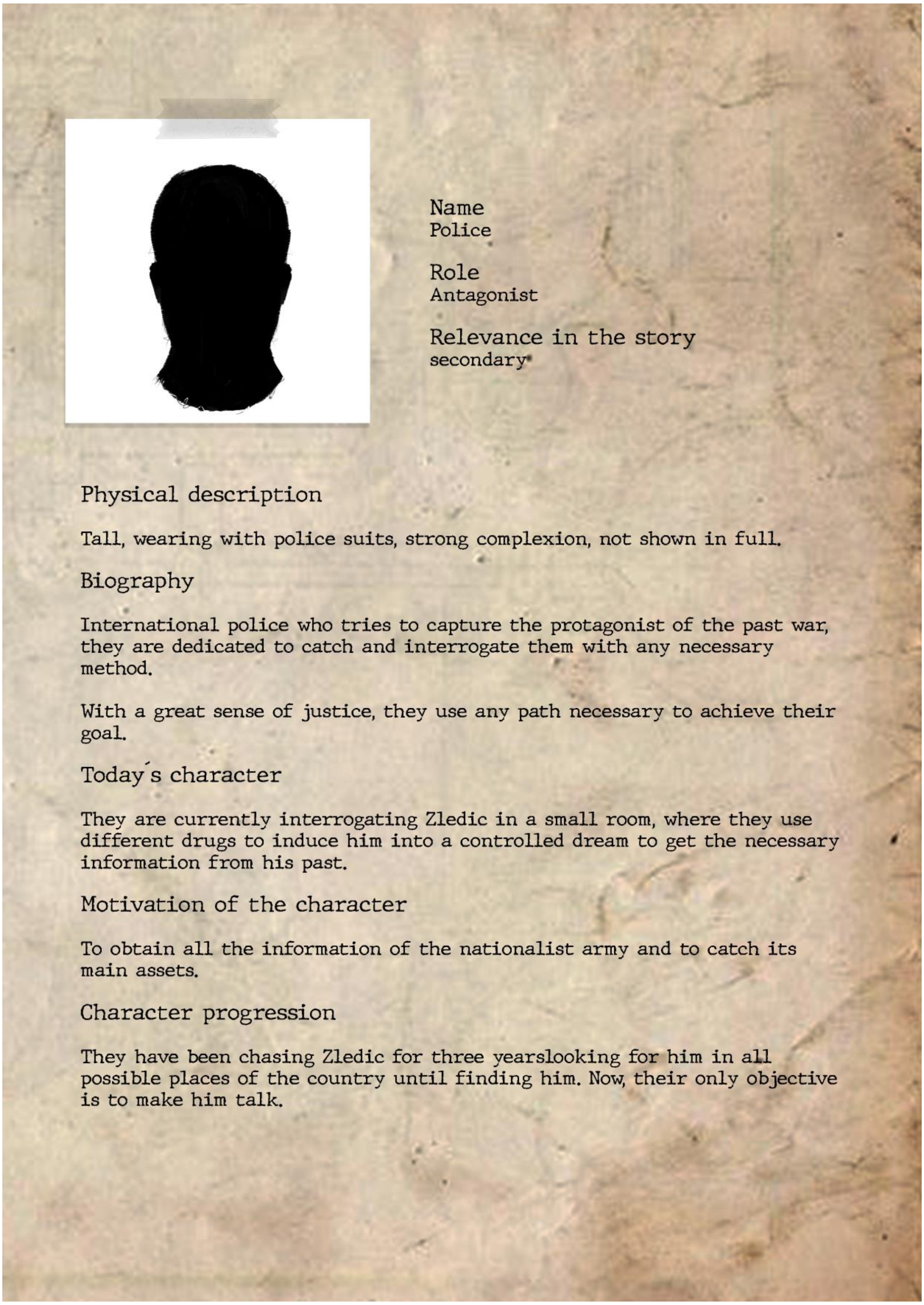
Motivation of the character

Sanja wants to be able to enjoy outside with other children.

Character progression

Thanks to the attitude of her mother, Sanja became a very smiling girl until the end of her life, when the situation worsened at home.

[3] Sanja’s file



Name
Police

Role
Antagonist

Relevance in the story
secondary

Physical description

Tall, wearing with police suits, strong complexion, not shown in full.

Biography

International police who tries to capture the protagonist of the past war, they are dedicated to catch and interrogate them with any necessary method.

With a great sense of justice, they use any path necessary to achieve their goal.

Today´s character

They are currently interrogating Zledic in a small room, where they use different drugs to induce him into a controlled dream to get the necessary information from his past.

Motivation of the character

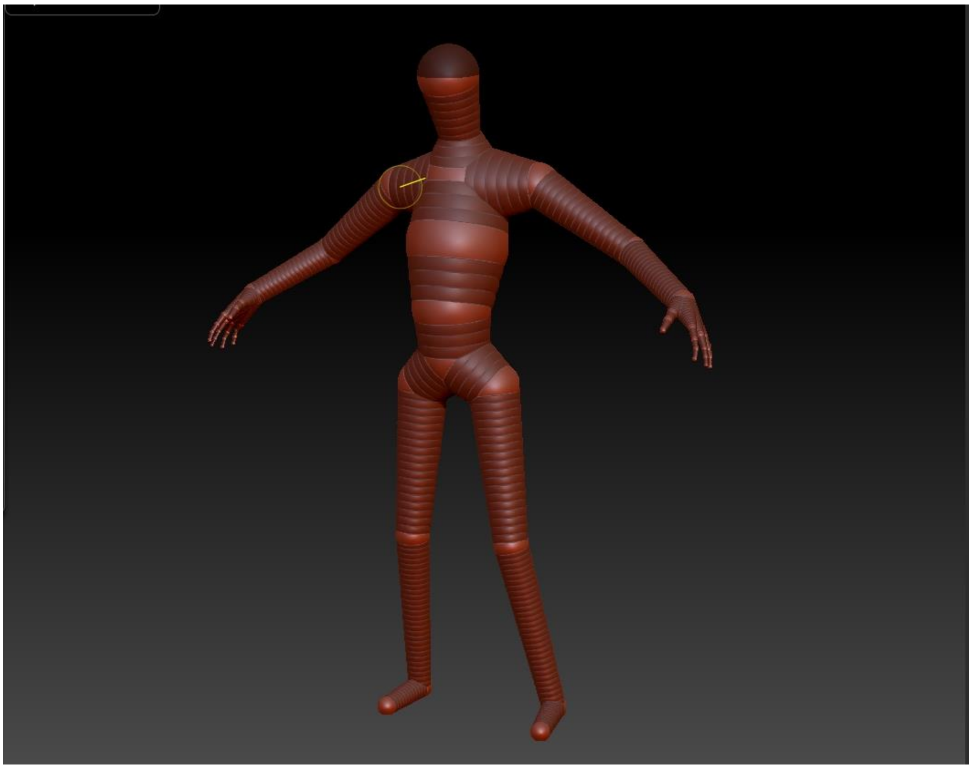
To obtain all the information of the nationalist army and to catch its main assets.

Character progression

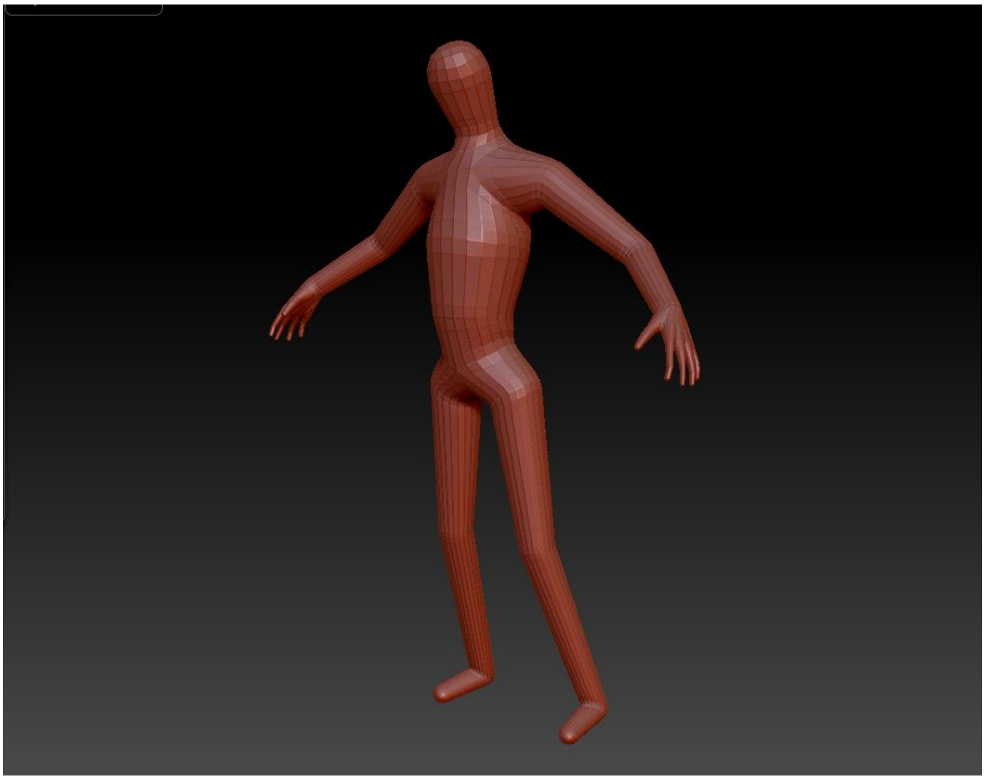
They have been chasing Zledic for three yearslooking for him in all possible places of the country until finding him. Now, their only objective is to make him talk.

[4] Interrogators' file

10.3 Zledic’s modeling process in Zbrush:

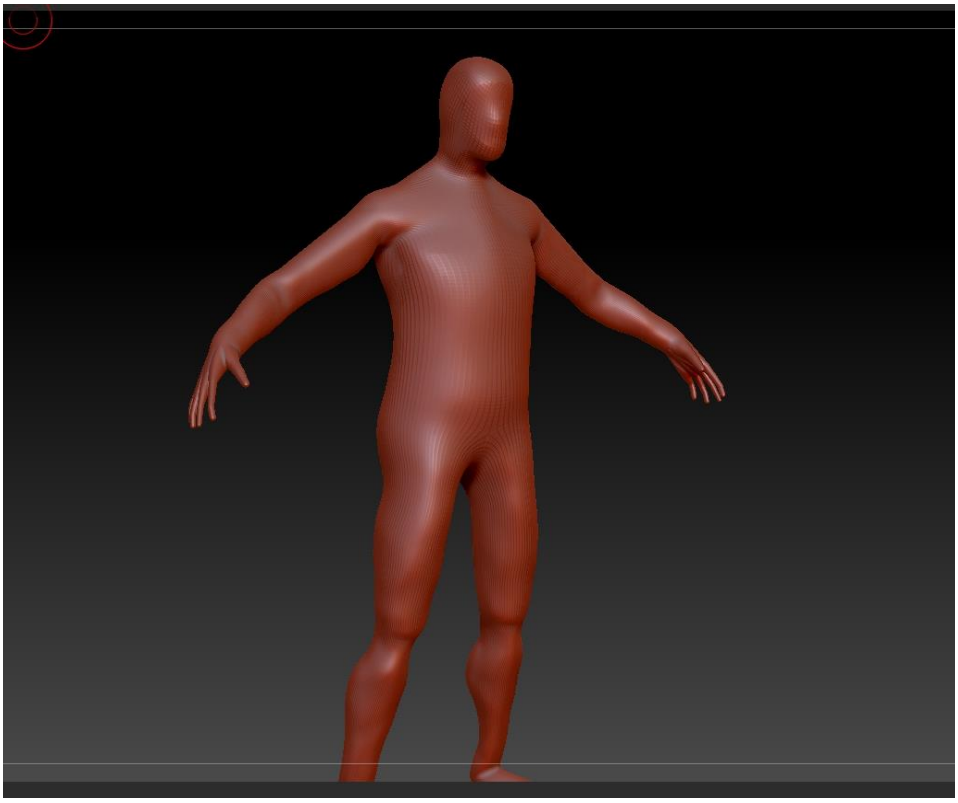


[5]

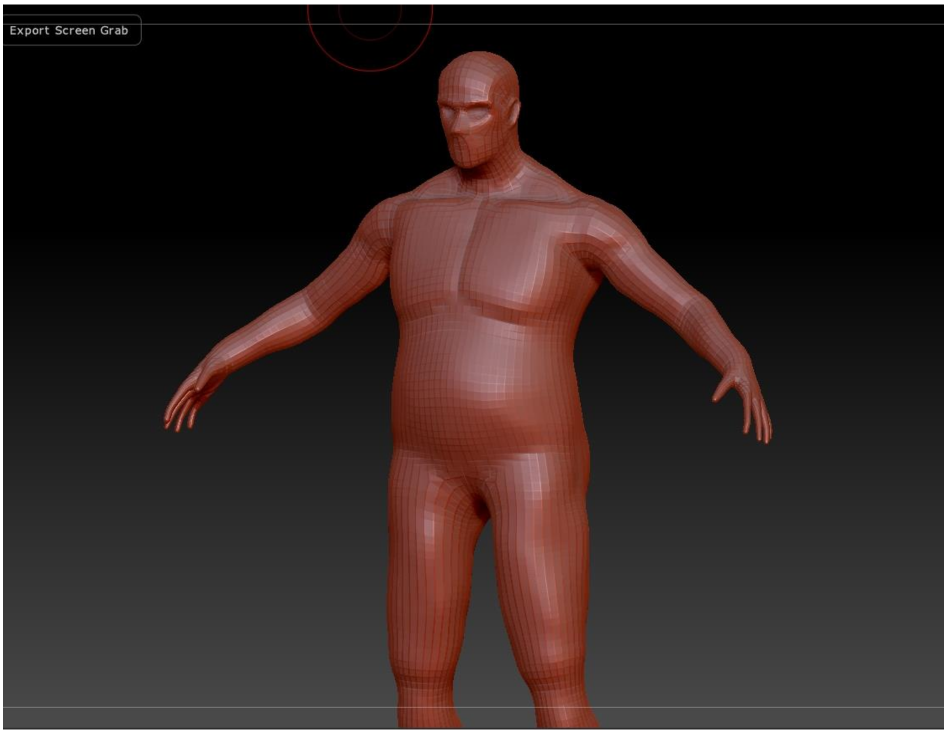


[6]

Zspheres base structure



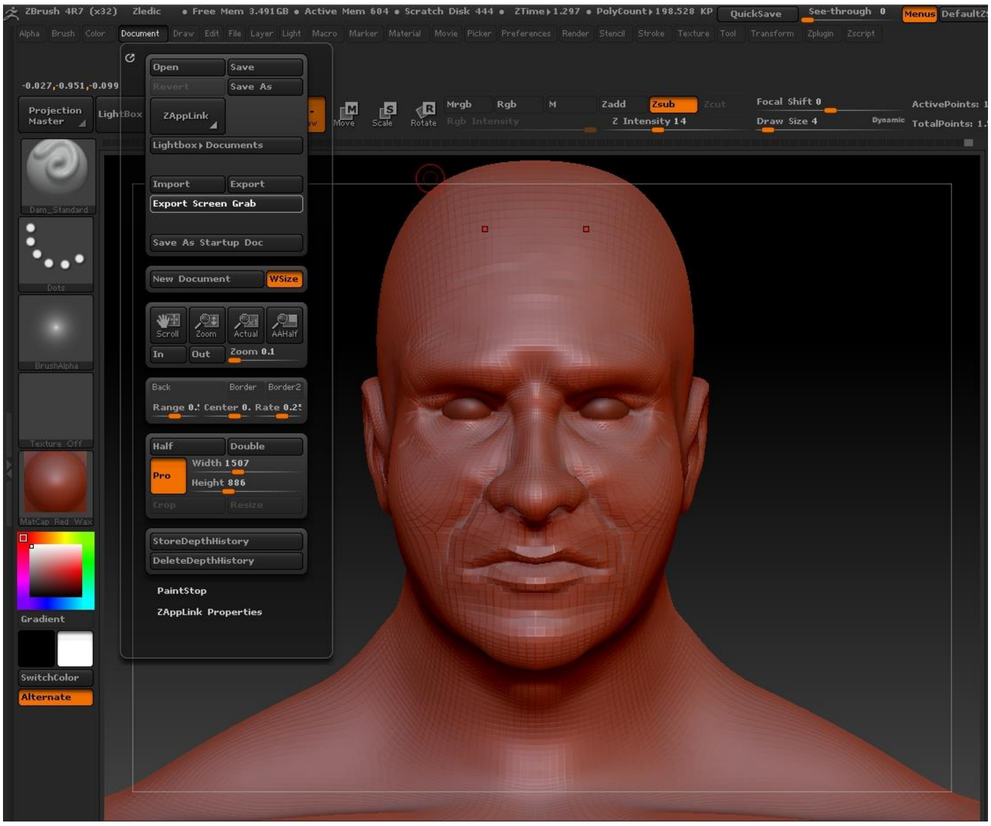
[7]



[8]

Body base modelling

Creating the model base, giving the first touches of volume and adjusting proportions to a low subdivision level.



ZBrush 4R7 (x32)   Zledic   • Free Mem 3.491GB • Active Mem 604 • Scratch Disk 444 • ZTime▸1.297 • PolyCount▸198.528 KP

-0.027,-0.951,-0.099

Zoom 0.1

Range 0.! Center 0. Rate 0.2!

Width 1507

Height 886

Z Intensity 14

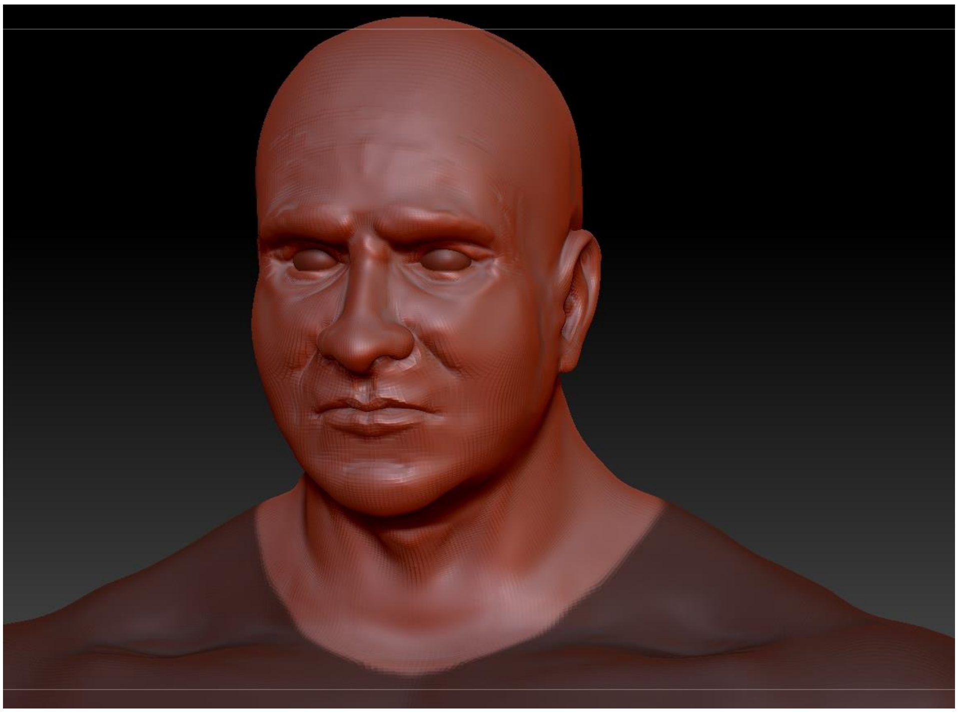
Focal Shift 0

Draw Size 4

ActivePoints: 1

TotalPoints: 1.

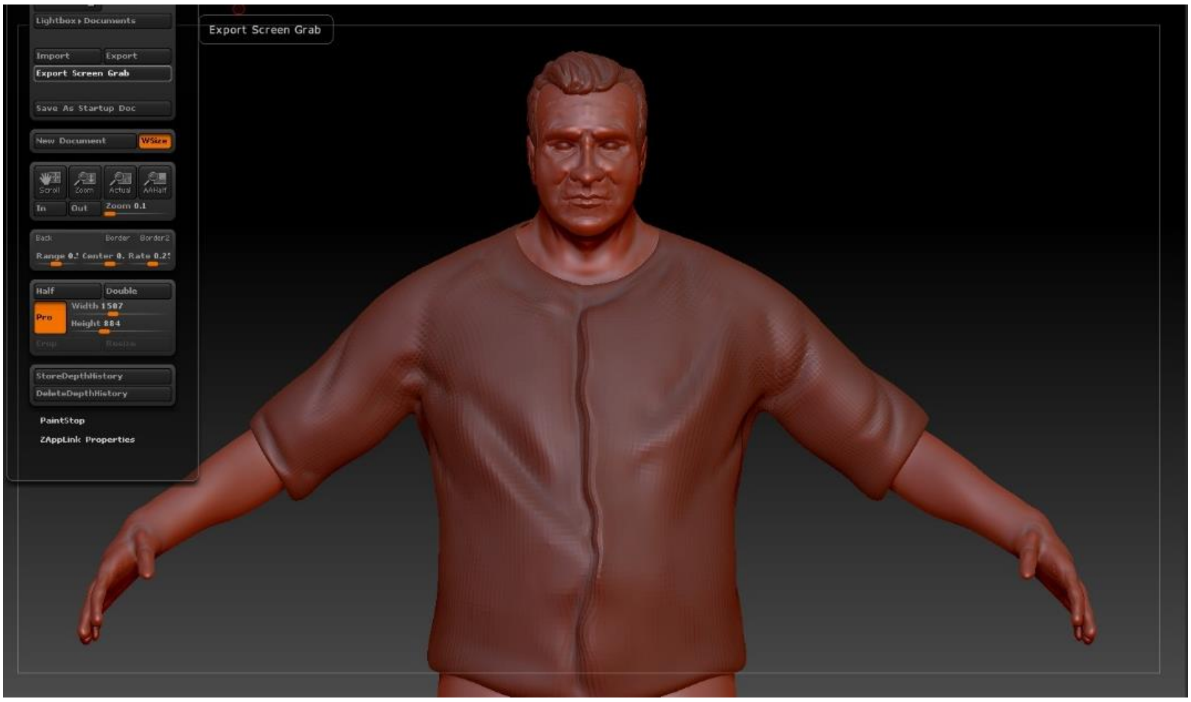
[9]



[10]

Face definition

With the well-adjusted geometry base, the subdivision levels are increased and the details and definition of the important parts are worked out.



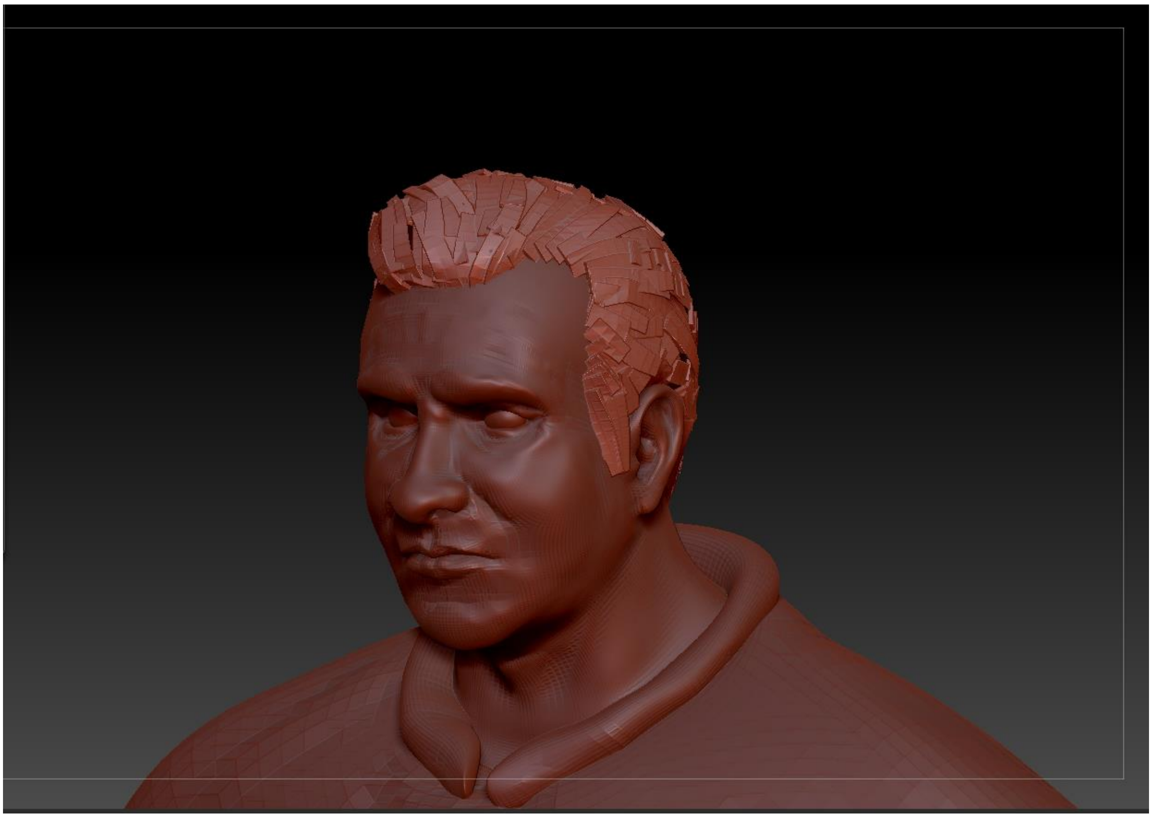
Zoom 0.1

Range 0.! Center 0. Rate 0.2!

Width 1507

Height 884

[11]

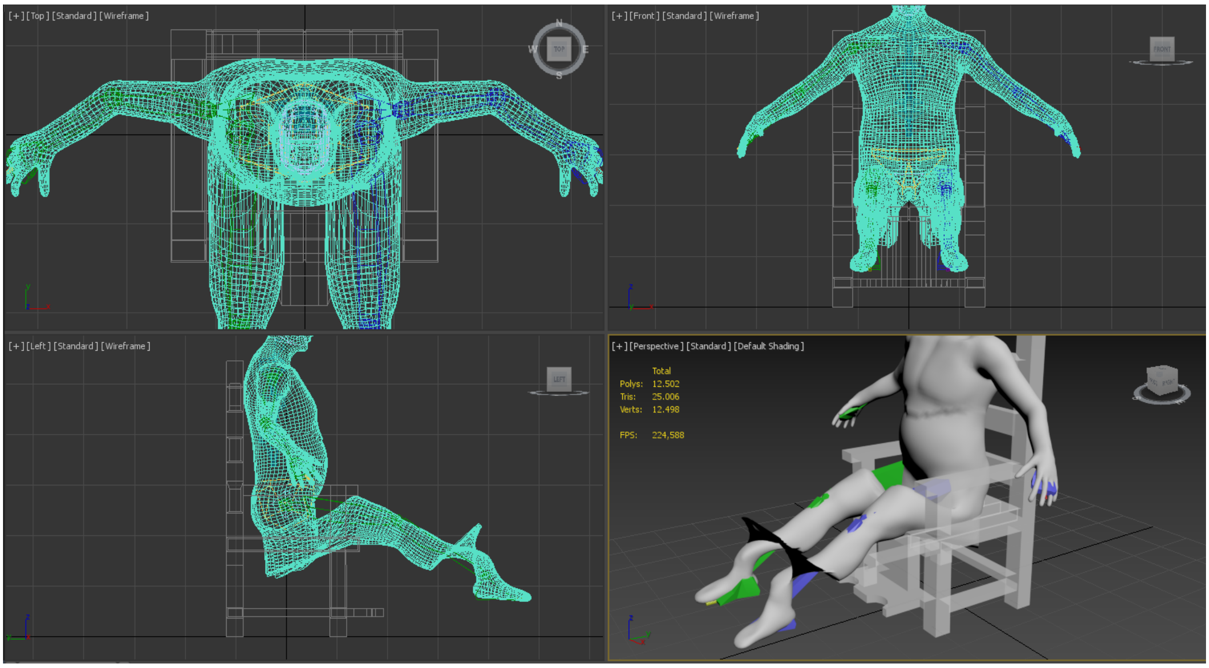


[12]

Base shirt and poly strips hair

These images show the first steps and attempts made for the modeling of clothes and hair inside the Zbrush interface, although as already explained in the memory these processes have been done in different ways in 3ds Max.

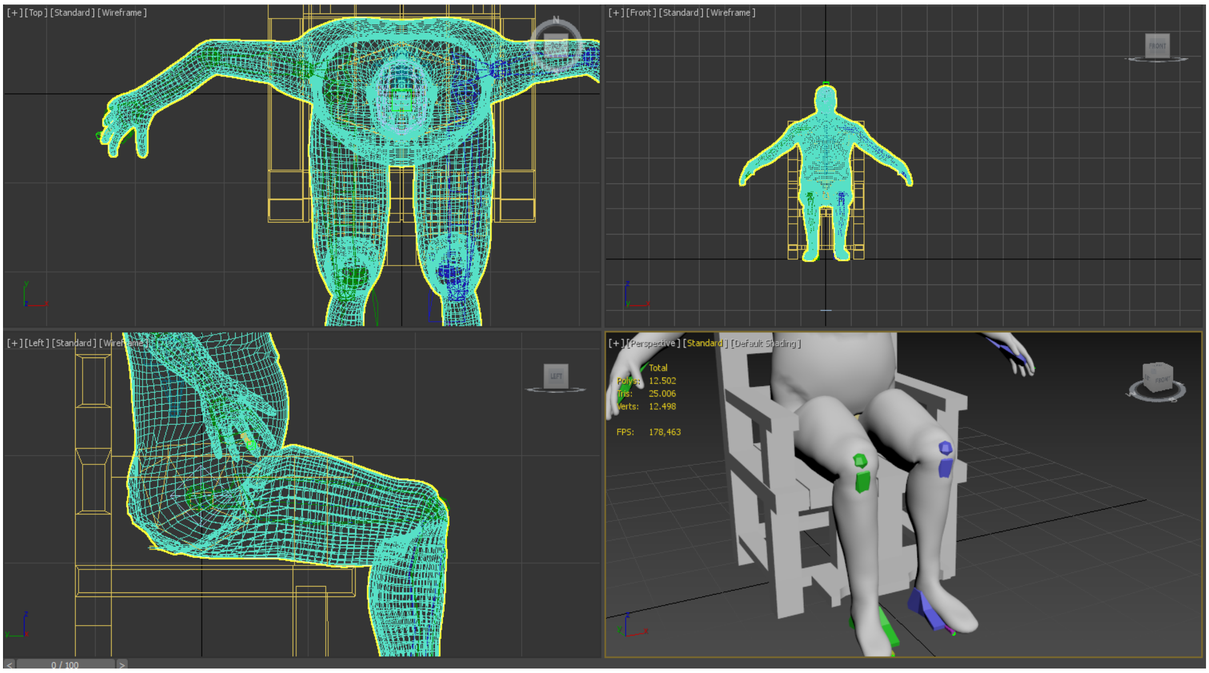
10.4 Zledic Biped Skeleton Adjustment


[+] [Top] [Standard] [Wireframe]
[+] [Front] [Standard] [Wireframe]
[+] [Left] [Standard] [Wireframe]
[+] [Perspective] [Standard] [Default Shading]

Total
Polys: 12.502
Tris: 25.006
Verts: 12.498

FPS: 224,588

[13]

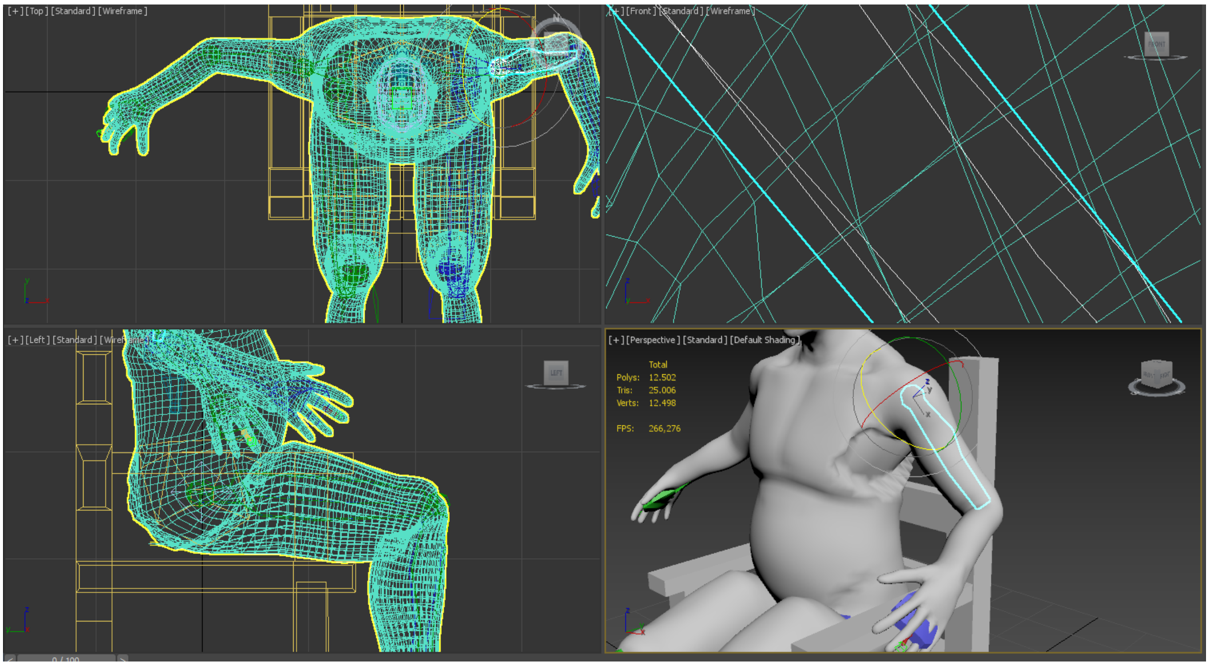

[+] [Top] [Standard] [Wireframe]
[+] [Front] [Standard] [Wireframe]
[+] [Left] [Standard] [Wireframe]
[+] [Perspective] [Standard] [Default Shading]
0 / 100

Total
Polys: 12.502
Tris: 25.006
Verts: 12.498

FPS: 178,463

[14]

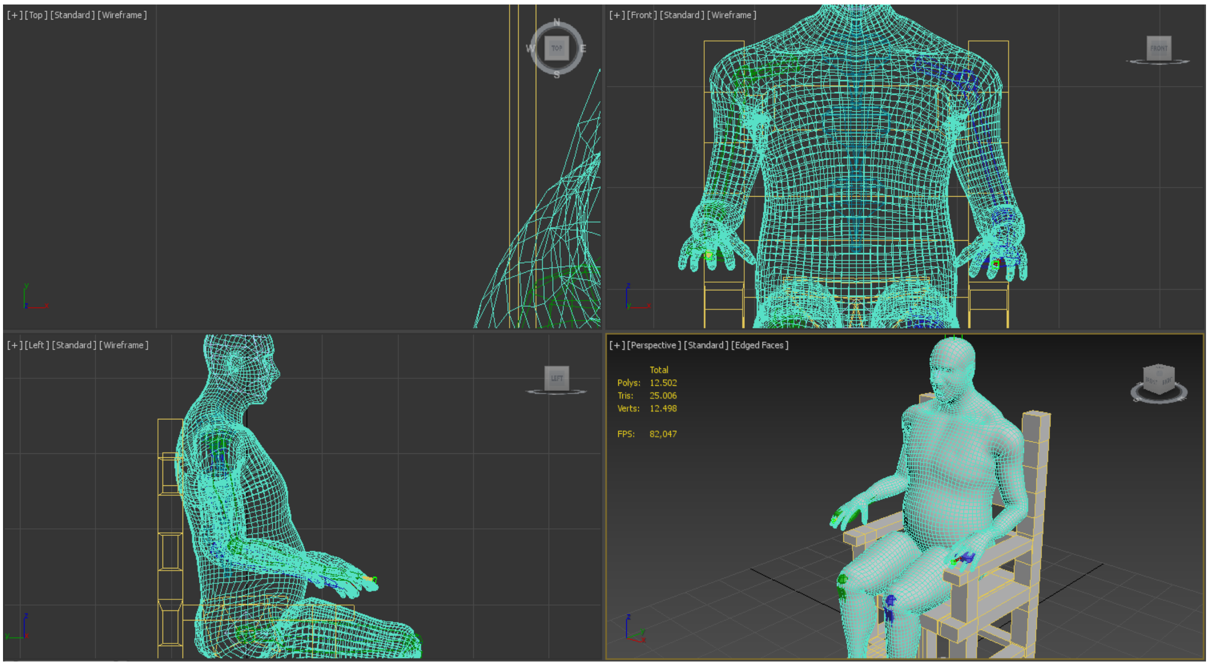
Biped placement with rotation tests, clear deformations and the result with the associations of adjusted vertices.


[+] [Top] [Standard] [Wireframe]
[+] [Front] [Standard] [Wireframe]
[+] [Left] [Standard] [Wireframe]
[+] [Perspective] [Standard] [Default Shading]

```
Total
Polys:  12.502
Tris:   25.006
Verts:  12.498

FPS:    266,276
```

[15]


[+] [Top] [Standard] [Wireframe]
[+] [Front] [Standard] [Wireframe]
[+] [Left] [Standard] [Wireframe]
[+] [Perspective] [Standard] [Edged Faces]

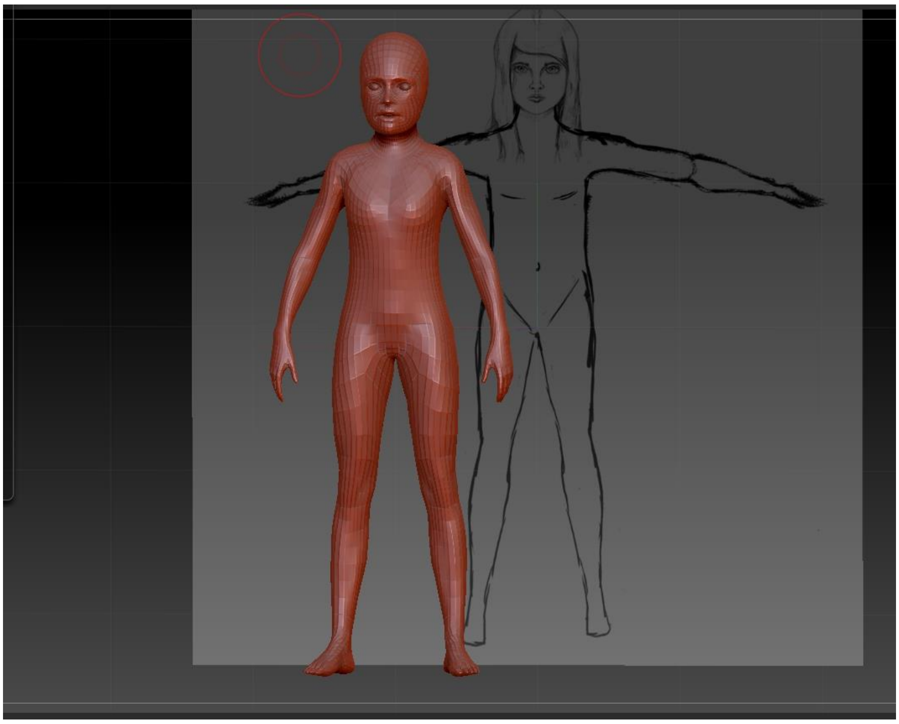
```
Total
Polys:  12.502
Tris:   25.006
Verts:  12.498

FPS:    82,047
```

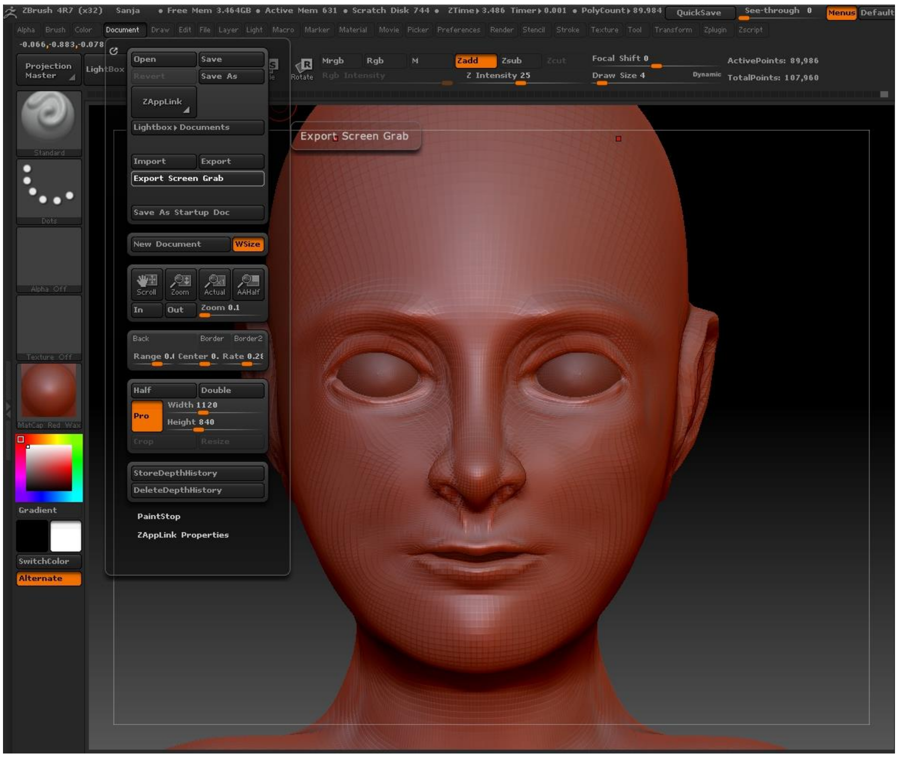
[16]

Same process as previous images, now with arms

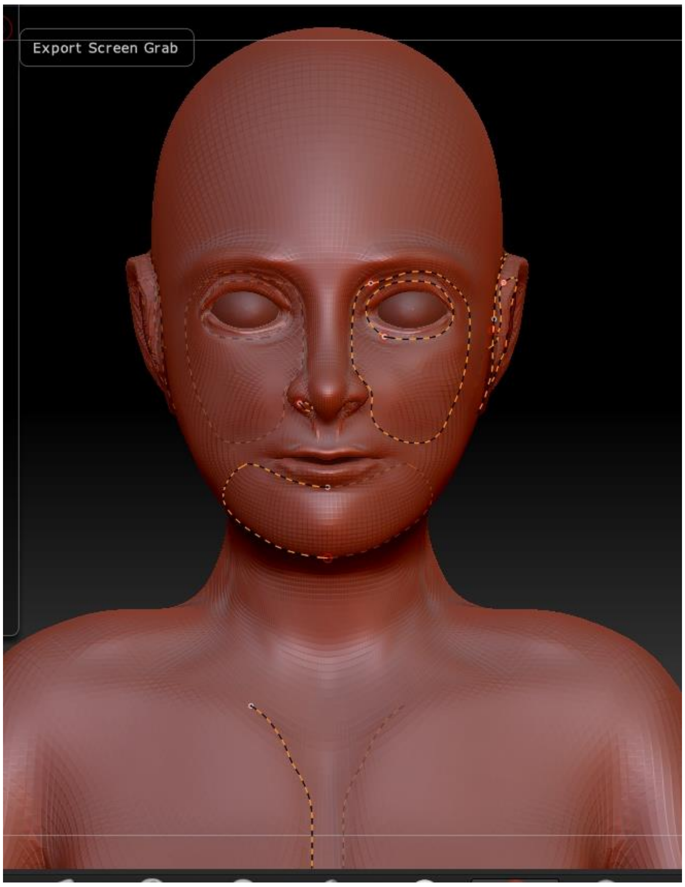
10.5 Sanja’s modeling & texturing process from Zbrush to Substance



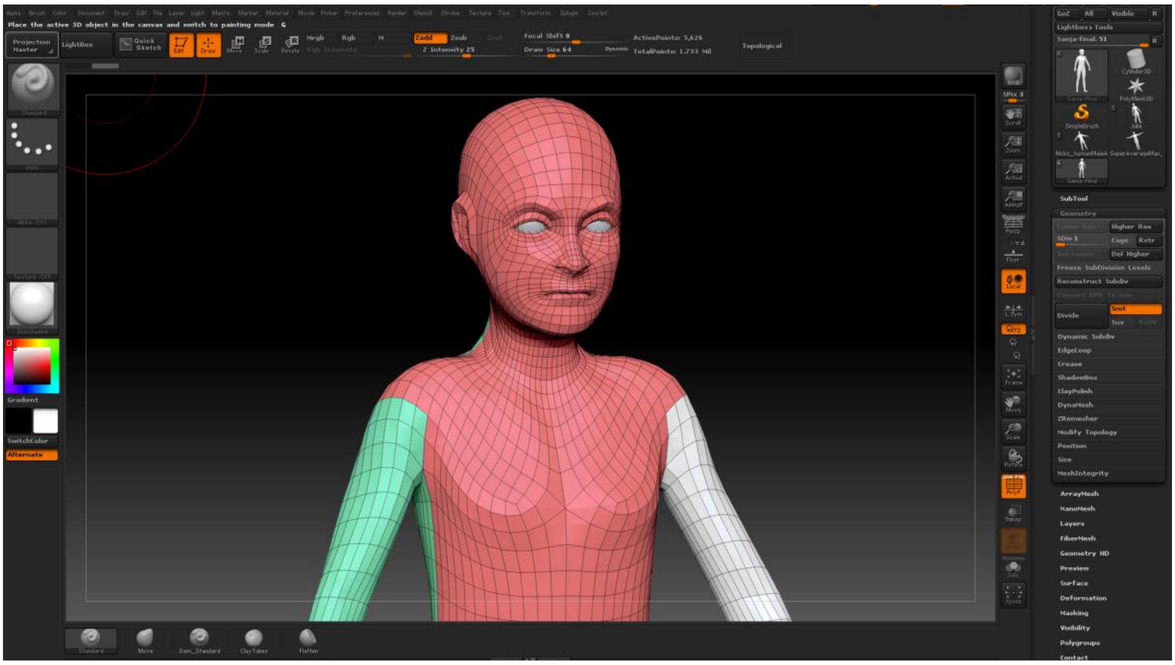
[17]



[18]
High poly definition



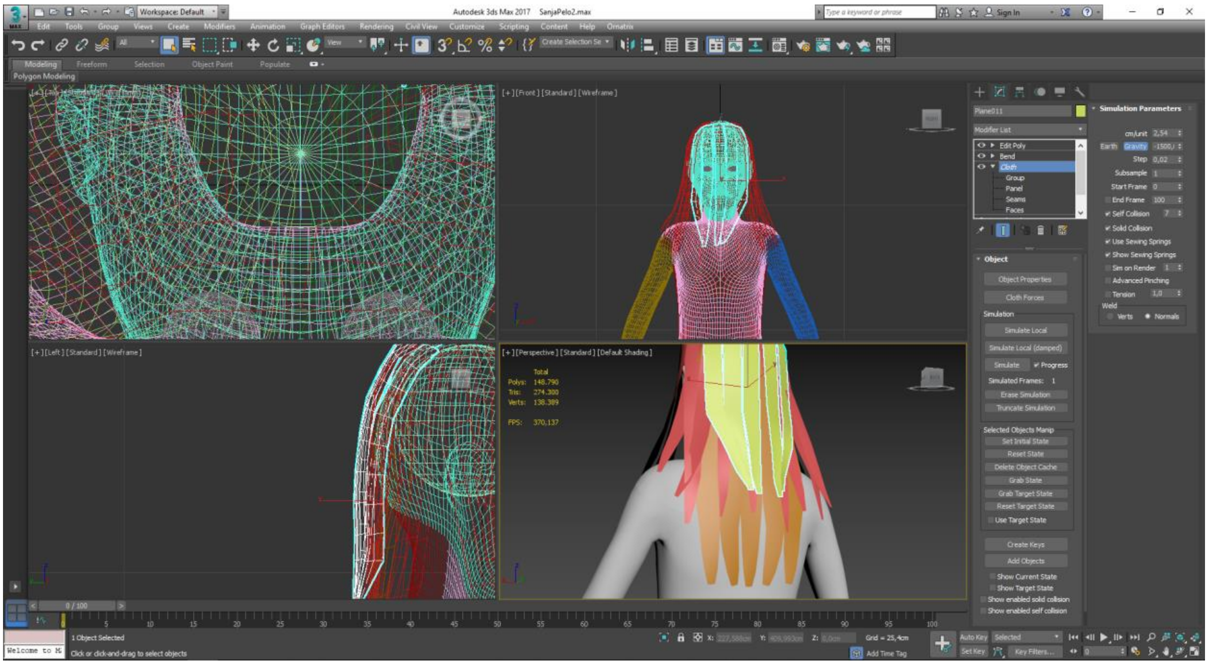
[19]



[20]

Retopology process

The retopology process at Zbrush, showing the retopology guides and the result they provide.



Total
Polys: 148.790
Tris: 274.300
Verts: 138.389

FPS: 370,137

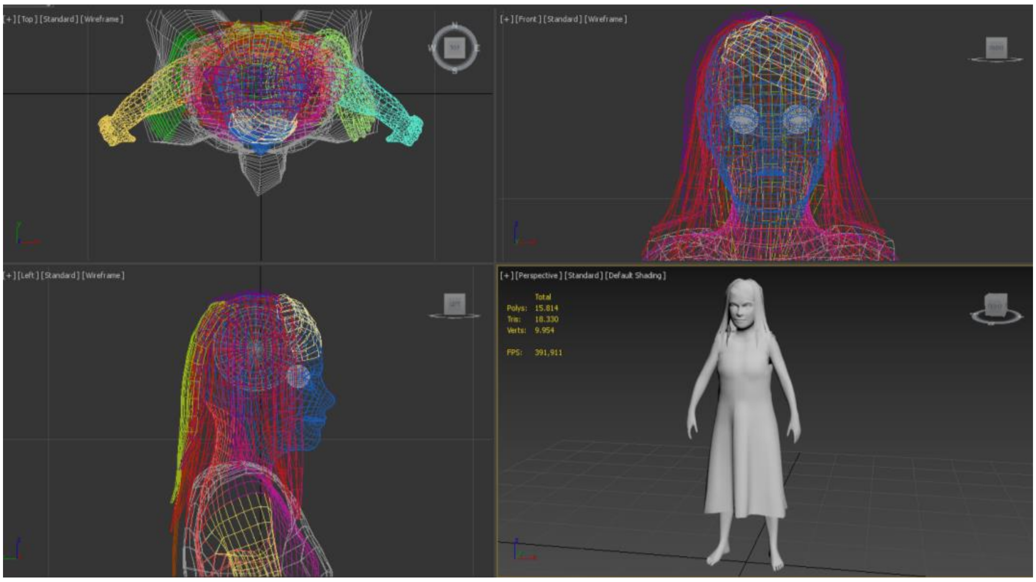
Simulation Parameters

cm/unit 2,54
Earth Gravity -1500,
Step 0,02
Subsample 1
Start Frame 0
□ End Frame 100
☑ Self Collision 7
☑ Solid Collision
☑ Use Sewing Springs
☑ Show Sewing Springs
□ Sim on Render 1
□ Advanced Pinching
□ Tension 1,0
Weld ○ Verts ● Normals

Simulated Frames: 1
☑ Progress

[21]

Poly-strips process with Sanja’s hair using Cloth modifier

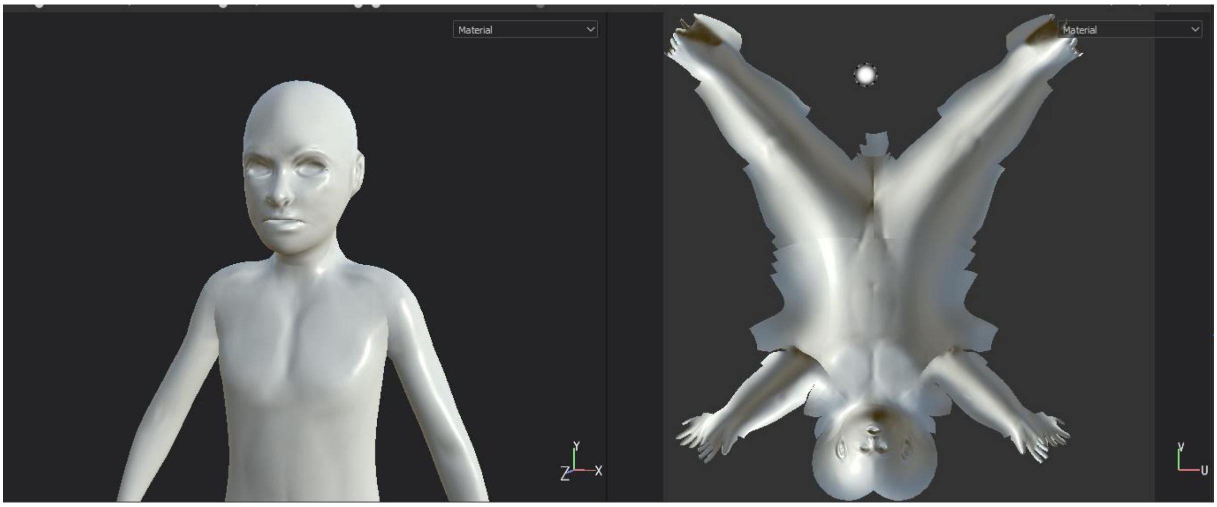

[+] [Top] [Standard] [Wireframe]
[+] [Front] [Standard] [Wireframe]
[+] [Left] [Standard] [Wireframe]
[+] [Perspective] [Standard] [Default Shading]

Total
Polys: 15.814
Tris: 18.330
Verts: 9.954

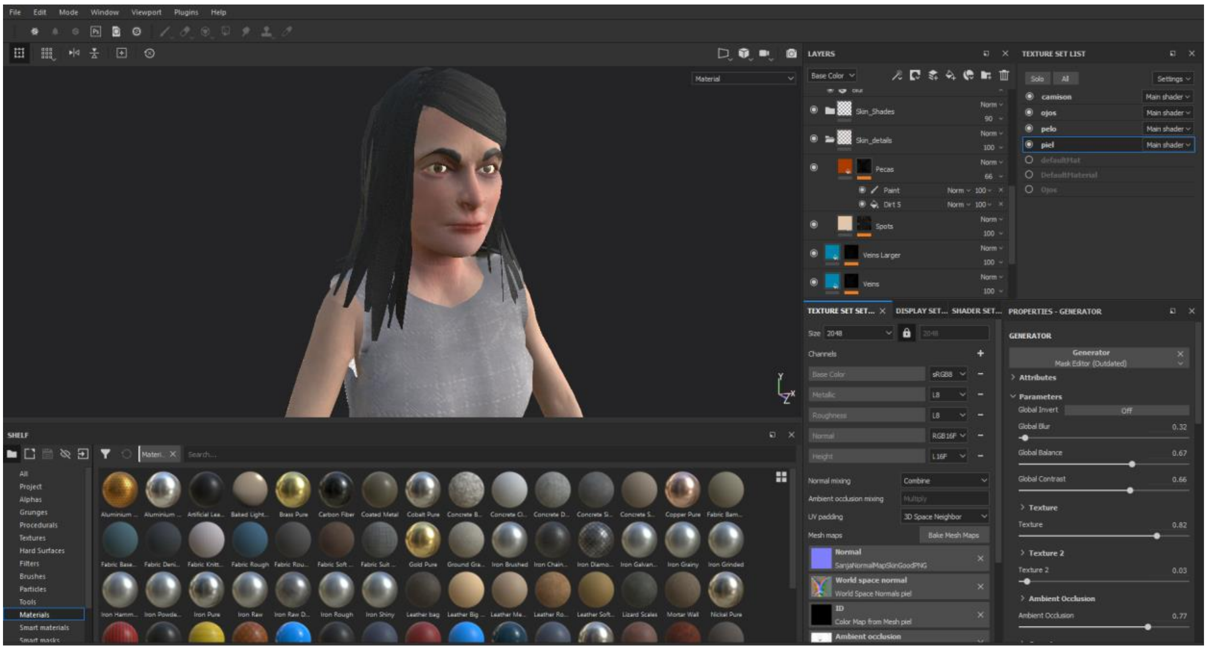
FPS: 391,911

[22]

Hair and Cloth
How the hair was finally modelled in 3dsMax instead of Zbrush as initially thought.



[23]



Base Color

Skin_Shades Norm 90

Skin_details Norm 100

Pecas Norm 66

Paint Norm 100

Dirt 5 Norm 100

Spots Norm 100

Veins Larger Norm 100

Veins Norm 100

camison Main shader

ojos Main shader

pelo Main shader

piel Main shader

defaultMat

DefaultMaterial

Ojos

Size 2048 2048

Channels

Base Color sRGB8

Metallic L8

Roughness L8

Normal RGB16F

Height L16F

Normal mixing Combine

Ambient occlusion mixing Multiply

UV padding 3D Space Neighbor

Mesh maps Bake Mesh Maps

Normal SanjaNormalMapSkinGoodPNG

World space normal World Space Normals piel

ID Color Map from Mesh piel

Ambient occlusion

GENERATOR

Generator Mask Editor (Outdated)

Attributes

Parameters

Global Invert Off

Global Blur 0.32

Global Balance 0.67

Global Contrast 0.66

Texture

Texture 0.82

Texture 2

Texture 2 0.03

Ambient Occlusion

Ambient Occlusion 0.77

[24]

Final texturing

10.6 Texturing object process, video:

https://www.youtube.com/watch?v=L3W_LFEoVjY