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

Procedural Environment Generator

Carles López Iglesias

Director: Marc Ripoll

Grau: Videogames

Curs: 2021-22

Universitat: CITM

Index

Summary

This thesis consists of the development of a plugin for a 3D engine that allows the user to generate a 3D environment procedurally. How it will work is by taking a reference to 3D models previously created and displaying them in a finite area inside the world. This area, as well as the elements displayed on screen will be able to be editable by the user without entering or touching any script. The tool will contain an user interface where there will be the assets chosen, the area where the environment will be and the quantity of assets to display.

The decision of using a commercial engine instead of creating it from scratch has been taken in base of the accessibility for the user, due to installing an external plugin is more simple. Another feature to take into account is the distribution and commercialization of the project, due to an engine from a big company containing its own marketplace where the users can upload projects and plugins.

The main purpose of this project is also the possibility of implementation of this plugin in a videogame, due to it can save a lot of time while designing big scenarios taking advantage of procedural generation technology, which has been growing up during the last years.

Glossary

3D Modeling: The process of creating a three dimensional representation of any kind of object using determined software.

Array: An ordered collection of elements that can be addressed and manipulated by using indices.

Seed: A value used to be applied into a random number generator to determine the sequence of generated random numbers.

Topology: The structure of polygons in a 3D model, determining its shape and characteristics.

UV Mapping: Process of projecting a 2D texture into a 3D model's surface, corresponding the vertices with the coordinates of the image.

Variable: A named storage location that holds a value that can be modified by using a program.

Key words

Procedurality, modeling, algorithm, environment, generation.

Links

Link to the project in Github:

https://github.com/carlesli/TFG-Procedural-Environment-Generation

Link to the build of the playable demo in Google Drive:

https://drive.google.com/drive/folders/1-QNqc_elmna9Y2AM7KxlZPhro1MHwq3v?usp=sharing

Table index

1. Introduction

1.1 Motivation

Since I started playing videogames as a child, a concept that has always called my attention is the representation of how levels are built, specially 3D environments, due to the fact that I saw a possibility to create my own worlds using different tools. As soon as I played more and more videogames, I just realized that maybe building levels and the worlds I imagined as a kid was a real possibility after discovering game development was a job.

Since then I started investigating how I could become a videogame developer, but I found that it was a huge world with a lot to investigate. As mentioned before, the building of an environment was a huge influence of mine, as well as the photography and traditional paintings, so this mix of visual inputs has awakened the artist in myself.

Although coding is not my strongest point, it is also pretty interesting and deep to get knowledge about, so the idea of being able to combine both disciplines into a project also increases my curiosity and my thirst for knowledge.

Recently I discovered the world of procedurality applied to videogames, so it looked like the best opportunity to develop my own tool that could combine my artistic knowledge of 3D modeling with my programming skills.

1.2 Description of the problem

While developing 3D content for a videogame, an animation or a movie, there is a part of the process where it needs an environment to be modeled and in some cases, timelines and budget are limited, as well as the production of many different assets can interfere with the progress.

For the reason mentioned above, among others, the proposal of this project is to create a tool for an existing 3D engine that allows the user to create a whole environment from scratch taking into account few parameters. The first parameter introduced by the user would be a group of 3D models previously created that will be taken as reference to the second parameter, which consists of generating an environment displayed in a plane which can be edited.

1.3 General objectives

This project has many points to establish one single objective, so it is divided into different big objectives:

- Procedural system to generate a 3D environment ready to be used in a videogame.
- Generate the 3D props that will conform the scenario, being coherent and sharing the same artistic style.
- Publishing the project on the internet with the highest quality possible.

1.4 Specific objectives

To achieve the objectives mentioned above, it is necessary to divide them into more specific objectives to have a better management of the project:

- Generation of 3D models that will conform the environment.
- Make a correlation among all the props to have an immersive level for the player.
- Creation of textures and materials for the models following the PBR (Physically Based Rendering) pipeline.
- Understanding the creation of 3D environment assets as well as working with procedural content.
- Showcase of the project in a 3D commercial engine.

1.5 Reach of the project

Due to the fact that the main objective of this project is to develop a tool for creating 3D environments, it can help from a student developing a personal project to an indie videogame studio that needs a quick production. The main idea of the result for this project is to reach as many users as possible, facilitating its accessibility, so the project will be distributed in the marketplace of the 3D engine where it will be developed, instead of launching an external plug-in.

2. State of Art

In the industry there are already different working pipelines created for the different stages of production, but the first of all it is important to define what is a pipeline, which consists in a combination of different processes that take place in the creation of any 3D content (videogames, animations, movies, etc), starting with the concept and finishing with the final result that needs to be achieved. The importance of this initial phase of the project is due to management, having each step planned and organized, as well as also adapting to different situations that can emerge during the production process[1].

In the case of this project, there are two remarkable pipelines. This first pipeline can be defined as the production of the 3D models that will be needed to take as reference to build the environment in the tool. The second pipeline that conforms to the second part of this project is the tool development.

The decision of dividing the project into two branches is for better organization and time management, which will be explained in the methodology section.

2.1 Modeling pipeline

In the situation of the videogame industry there are many different pipelines, each one depending on the company's decisions, but as a general point of view it can be divided into four big stages[2].

2.1.1 Pre-Production: Definition of an art style

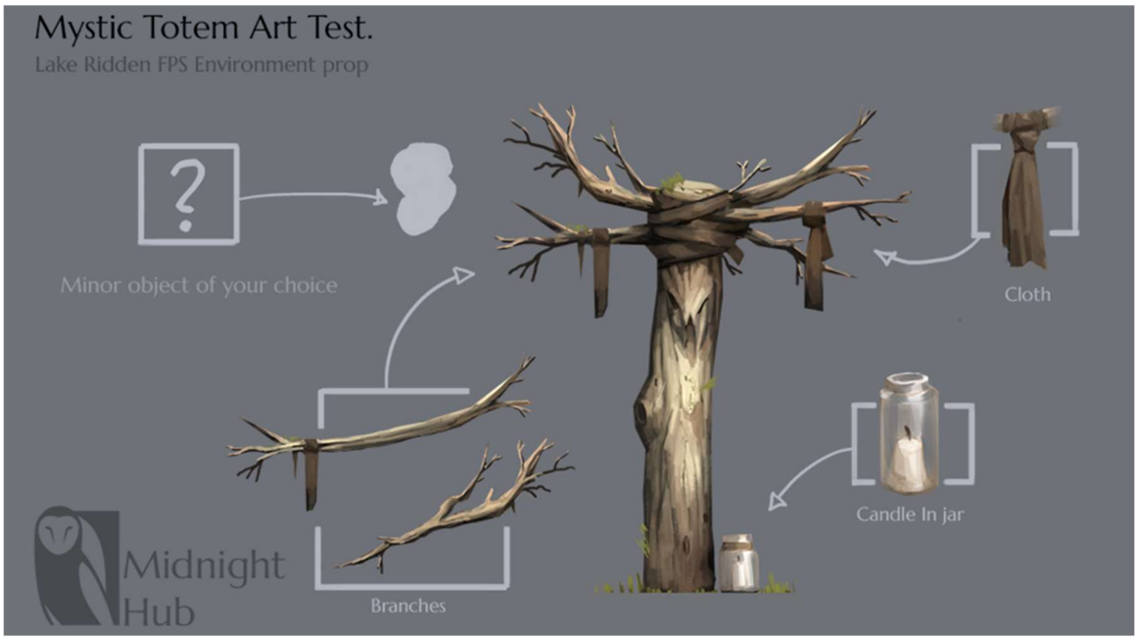
This is the first stage of the pipeline, which consists of creating a concept that will be used as the reference for the final model. In this stage it is common to have different versions of the concept to explore different ideas, defining the art direction and the overall vision of the project.

In the case of the concept of 3D assets, in the pre-production phase there are multiple draws of the same prop from different angles to show how detailed it will be. Although in this step there will be the "final" versions, some changes can be made until the final result[3].

[1] Kaedim - Medium - "10 Tips for creating efficient 3D pipelines" - November 23, 2022

[2]Chris Casmenco - Anim8 - "3D Production Pipeline" - February 21, 2018

[3] Tiger Collins - Medium - "3D Modeling Pipeline" - June 22, 2018



[4]Figure 1: Concept art of a prop.

Another important aspect of the pre-production process is defining the style of the art that will be established for the rest of the project. The objective of setting the visual style is to create an harmonious combination of all the components that the project will have to convey the desired atmosphere to the audience. This statement does not mean the number of polygons or how realistic a texture is, it is keeping in mind the resources available and combining them to get an atmosphere.
In the case of game development, defining the art style is directly related to the mechanics of such videogame to deliver the right experience and immersion to the players[5].

2.1.2 3D Modeling

Once the concepts are established and the approaches of the final versions are tested and polished, it is time to give a step to the modeling phase. To briefly define what 3D modeling is, it could be considered as the manipulation of the basic figures that are available, like cubes, cylinders, spheres, etc. At the start of the process, the first approach to the final model is named “White-box” or “Blockout”, which consists of testing and achieving the correct scale and fit.

[4] Image taken from “A kickass Game Art Portfolio” - Midnight Hub - erik - June 2, 2016
[5] Kevuru Games - “Choosing an art style for your videogame” - October 21, 2022

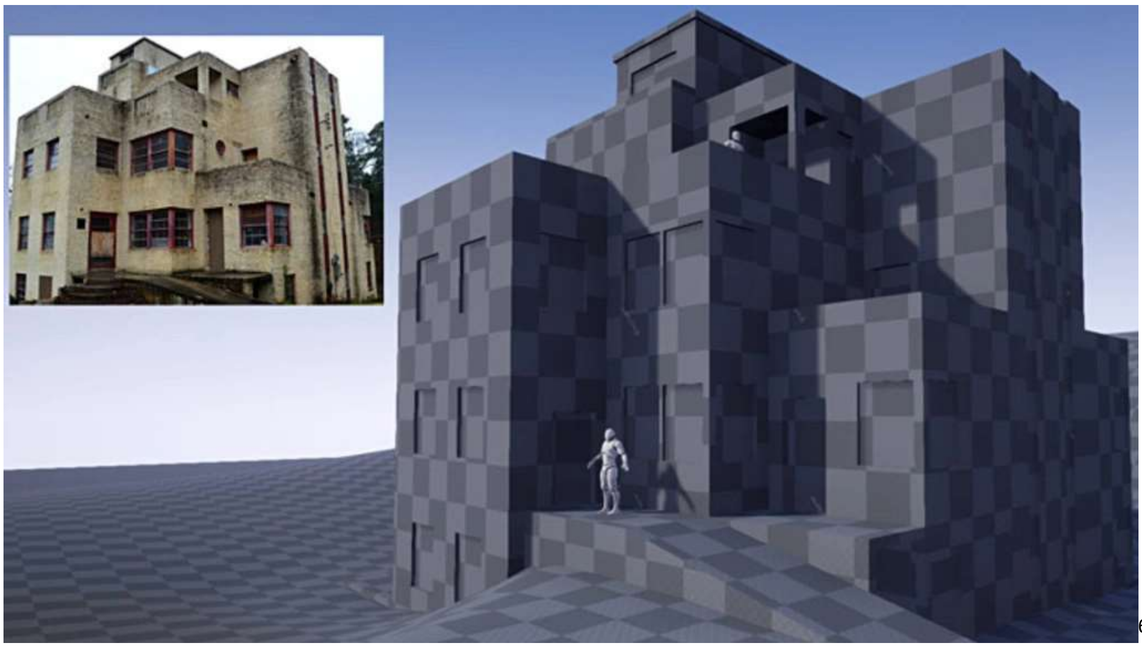

[6]

Figure 2: Blockout of a building.

In this situation there are only basic shapes and no detail in the model, so the next step is to make the “Low Poly” version of the model. This version will be the one displayed in the final result, so it is important to have a simple version of the model with less polygon count. The most important step in this situation is to create the retopology of the object, which consists in optimizing the count of polygons, avoiding unnecessary loops.
The next and final step of this stage is the “High poly” version of the model. In this situation, there is no need of optimizing the model due to it not being displayed in the game, so it can be added as much detail as the user wants[7].

[6] Image taken from “3D Modeling Part 1: The Basics of Building a Clean 3D Model” - Shutterstock - Francesco Furneri - December 8, 2021
[7] Ratloop Games Canada - “The Art Pipeline”

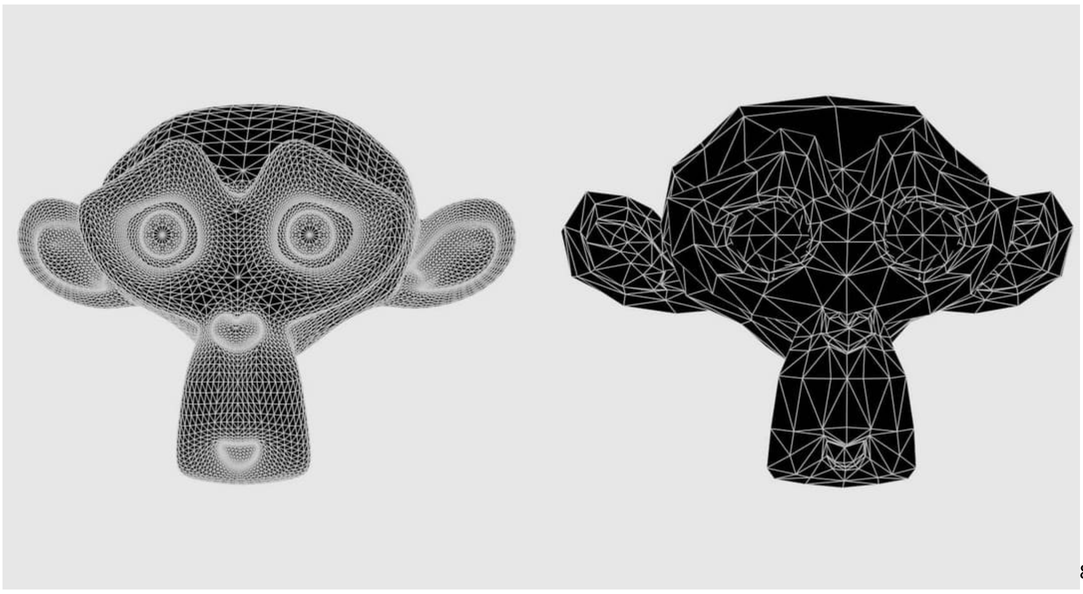
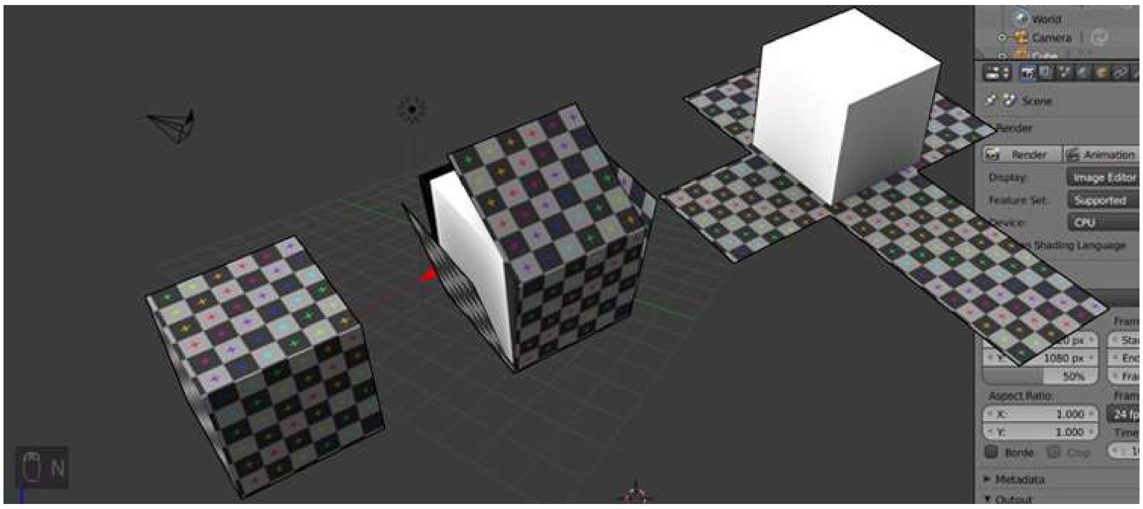

8

Figure 3: High poly and Low poly comparison.

2.1.3 UV Mapping

In this stage of the pipeline, the low poly version of the model mentioned above will be needed to be unwrapped. This method consists of the flat representation of the surface of the model to wrap textures, so instead of being represented in 3D it must be represented in 2D (taking as reference the U and V axis due to X and Y being already used in 3D). This process can be considered the most tedious one, but it is also necessary to give personality and composition to the models created through the textures[9].



[10] Figure 4: Representation of the UV unwrapping of a cube.

[8] Image taken from "What is the difference between High Poly and Low Poly models in 3D modeling?" - Modelry - Sarunas Vendelskis - September 1, 2022

[9] Thomas Denham - Concept Art Empire - "What is UV Mapping and Unwrapping?"

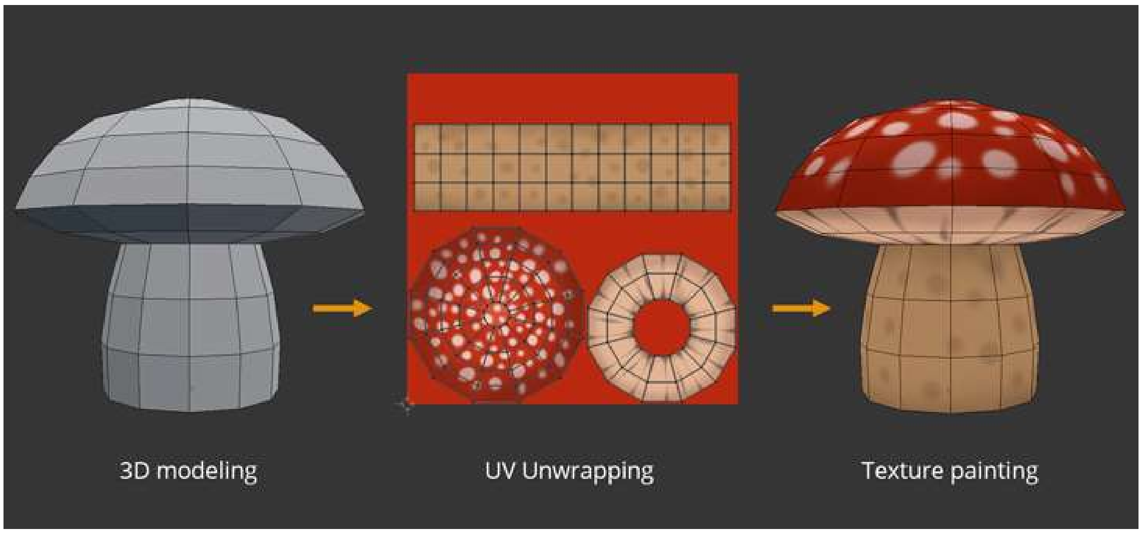
[10] Image taken from "What is UV Mapping and Unwrapping?"

2.1.4 Texturing

Once the model is optimized and unwrapped, the texturing is the final stage of this process, which consists of giving 3D objects colors and properties, matching with the concept first established and giving the user a sense of substance about the object, simulating a material from the real world. When a user adds a material texture to a model, it has to take into account that this material can react to the light of the environment by the reflection or refraction, among other features, so the detail designed for the texture must be established also in the pre-production so it correlates with the visual approach of the final project.

As it is developed above, this is generally how 3D models are made in the videogame industry, but it does not mean that there are other pipelines that arrive at the same objective with different steps, each project carries different situations and it has to adapt to them.

The second big part of the state of art of this project is defined by an existing development of the tool presented at the beginning, which consists of the production of a procedural environment tool[11].



[12] Figure 5: Showcase how a texture gives a 3D model material.

2.2 Creation of a 3D environment for a videogame

During the mid-to-late 1990s were the years where 3D models started being implemented in videogames more commonly, taking more popularity than 2D games. For this reason, the processes to create those levels in a 3D environment have been changing throughout the years, as well as new technologies have been developed[13].

[11] A23D - "What is 3D texturing?"

[12] Image taken from "Tutorial: Modeling, UV Unwrapping and Texturing a Mushroom" - Blendernation - Widhi Muttaqien - April 22, 2017

[13] Tech shout - "3D Environment Design for Production: Explained" - Carla Levi

In a videogame, the creation of an environment has many features to take into account, involving a lot of planning and preparations from drawing and sculpting in 3D[14].
This process includes the art of building virtual worlds that give the player the feeling of being and transport into another world. With the pass of time, 3D scenarios have got deeper layers which have allowed to develop qualities and elements that define the virtual space created through three-dimensional graphics and technology. These characteristics contribute to:

- Depth and dimension. This allows objects and elements to have height, width and depth, being a more accurate representation of the real world than 2D environments[15].
- Realism and detail. This is a crucial part of the production, being a mix of different concepts like the textures, the lighting, shadows and physics, aimed to create a visual coherence and immersive experience.
- Interactivity and navigation. One of the best features of videogames is the interaction between the game itself and the player, so in a 3D environment there must be objects that the player can manipulate to get a more engaging experience and not to break the immersion.
- Dynamic and changing environments. A scenario can also be designed to change dynamically, like for example the most common change of many videogames, the weather, adding a sense of realism in a virtual world.
- Spatial audio. Just like in the film industry, the audio is one of the most important aspects of a videogame, helping to set the scene at any time and play with the emotions of the player, so with the audio displayed in a 3D space, it is possible to have a perfect immersive experience[16].
- Scale and proportions. It refers to the dimensions and the relative size of the level, having to develop a consistent sizing system, visually adding the player a challenge or to look for showing some emotions indirectly[17].

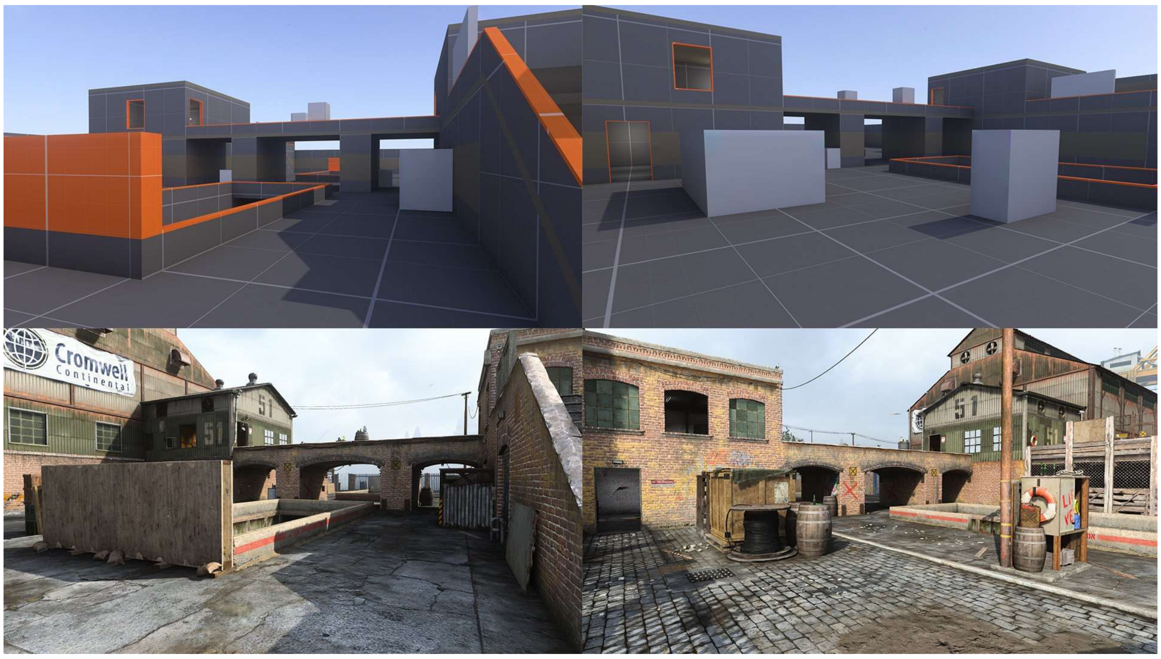
As it can be seen in the points mentioned above, what is more searched for in the creation of a 3D environment is the immersion of the player to create the perfect atmosphere to make a memorable experience with the goal to feel like the player is actually there.
Once the definition of a 3D environment is established, it is necessary to know how they are made in the videogame industry and which steps are needed to achieve the final result. Although there are different pipelines, most of them share the same steps[18]:

- Concept and layout phase. This is the first step of the process, so the most important part is to conceptualize the environment by taking references and receiving information about other similar works to achieve the final direction.

[14] Argentics - "The Science of creating 3D Environments" - March 10, 2022
[15] Polydin Studio - "3D Environment Design | Everything You Need to know" - March 1, 2023
[16] TotalNtertainment -"Why music is so important in gaming" Lee Holmes -
[17] Game Designing - "In-Game Sizing & Scale: Consistent Character Sizing and More" Dustin Tyler - June 10, 2023
[18] Quora - "What is the process to create 3D environments for the videogame industry?" - JetBrains - 2018

What is most common in this situation is to create a moodboard that gives the feeling of the final art to take as reference.

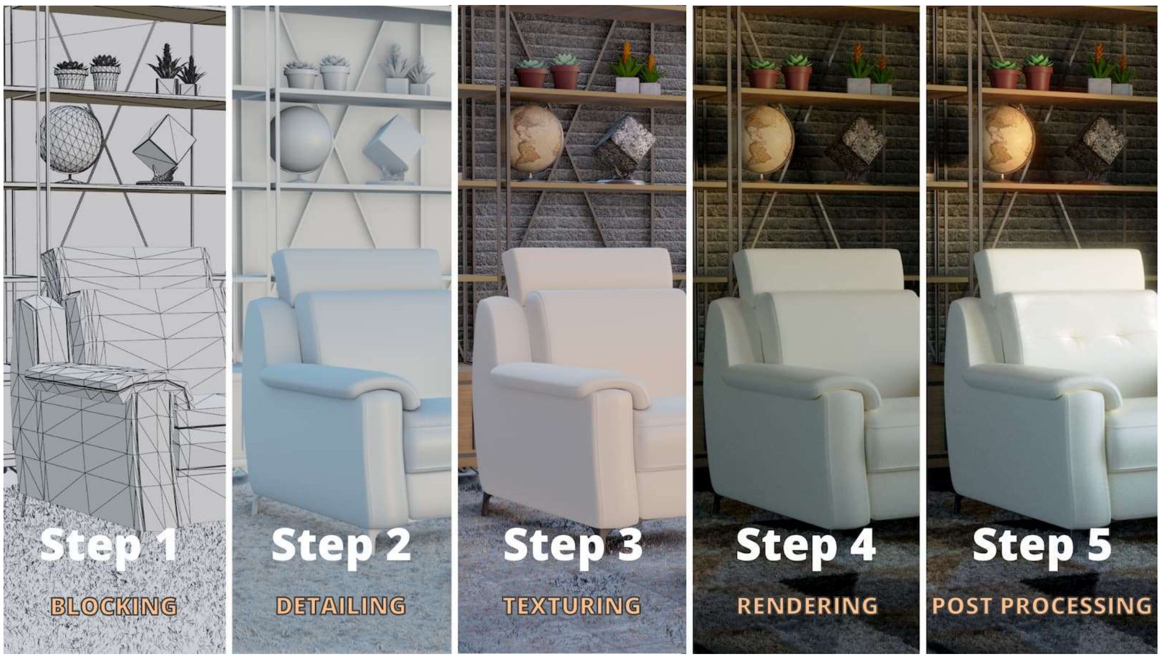
- Blocking. This step is also considered in the planning section, due to it consists of creating the level with white basic models to have a clear vision of the future models, allowing to test how the level works, the shapes,etc[19].



[20]Figure 6: Image of how firstly is a blocking of the level

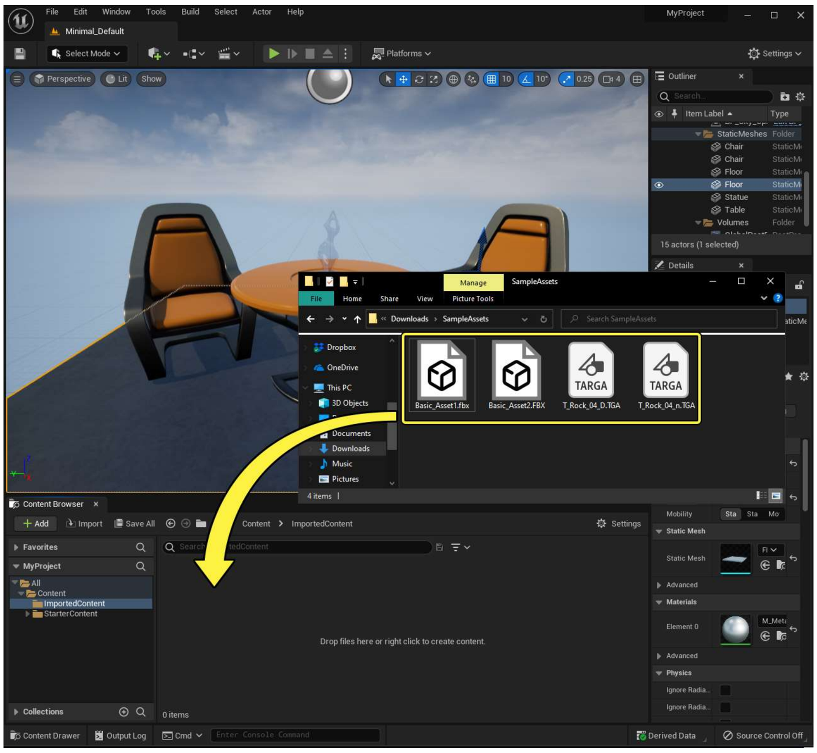
- Asset creation. This is the process of creating the 3D models that will compound the level, so the steps to follow are the mentioned above about the creation of assets, but to resume it can be divided into seven steps: concept, blocking, low poly, uv mapping, high poly, texturing and baking.

[19] Game Ace - "Game Environment Modeling: The Essence of Building a Stunning World" - September 29, 2021
[20] Image taken from: "The Level Design Book - Blockout" - January, 2023



[21]Figure 7: Image of the process of 3D modeling.

- Importing the assets to the game engine. This step is based on taking the models and textures created previously and displaying them in the engine that will be used for the game.



[22]Figure 8: Image to show how to import a mesh into 3D engine.

- Lighting. This is also an important part of a videogame, so it owns the power to make or break the visuals, theme and atmosphere. This feature has had more

[21] Image taken from: "[5 Steps To Creating an Amazing 3D Model]" - Elodie Larour - September 17, 2020
[22] Image taken from: "[Unreal 5 Documentation - Importing Assets Directly]" - Unreal 5 - 2022

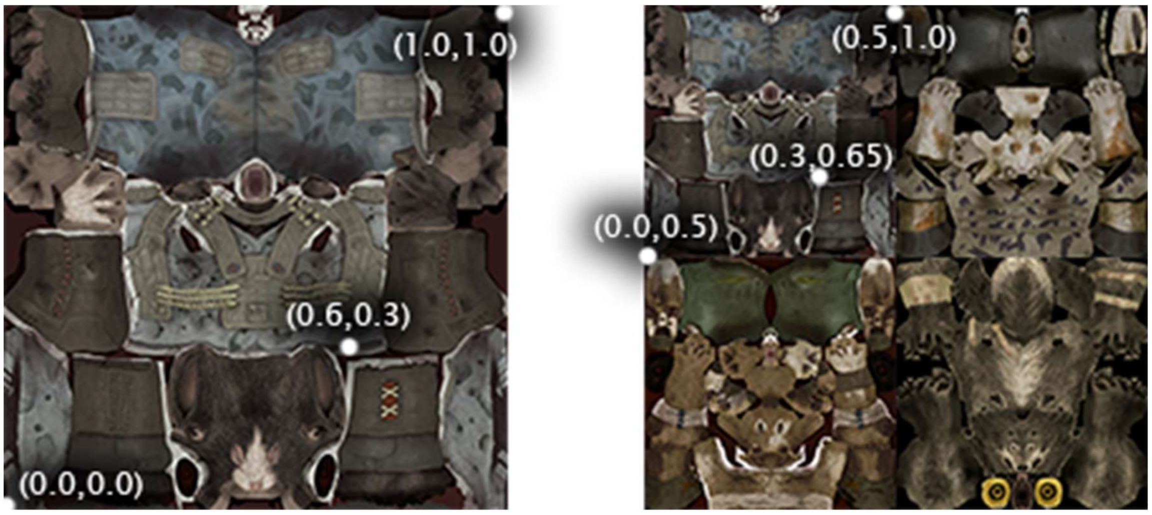
influence in the last few years due to the upgrade of the technologies, allowing it to give a visual impact greater than only displaying 3d assets on screen.

[23]Figure 9: Image that makes a showcase of the effect of a good light

- Finalization and testing phase. Once everything has been created, it is time to test the final result to adjust the achievement of the game.

One of the biggest challenges while making a videogame environment is the optimization of resources inside the game and also inside the engine, and it can be made using texture atlases.

A texture atlas is an image that contains data from several smaller images that have been packed together, so instead of having one texture of a mesh, this new texture will have more than one mesh[24].



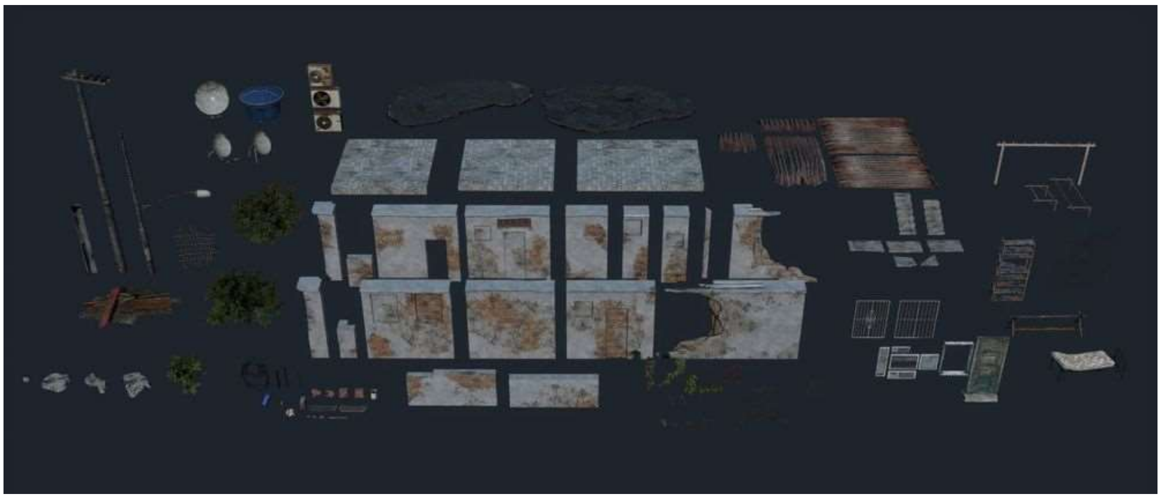
[23] Image taken from: "The ultimate guide to lighting fundamentals for 3D" - Dream Farm Studios - Arash Naghdi - 2021
[24] Arm Developer - "Real-time 3D Art Best Practices - Texturing"

[25]Figure 10: Showcase of how an atlas texture works

Another optimization technique commonly used in the industry is the modularity of the assets. This concept is an approach also used in traditional architecture to facilitate the off-site construction of large building sites, being a cheaper and faster process, but having a different design logic than other processes.

The system is composed of separate components, which are called modules, and these can be connected or integrated together, allowing components to be added or replaced without affecting the whole system. The scale of the space is not a problem anymore, due to it can be achieved through the repetition of the modules, for this reason it allows the addition or subtraction of components and can enhance the flexibility of usage and maintenance of a built structure.

Modular game environment art is a great help when it comes to the creation of big scenarios, needing a good functionality, due to it is very important that all meshes respect their assigned grid space, following the same pivot logic and standard transitions[26].

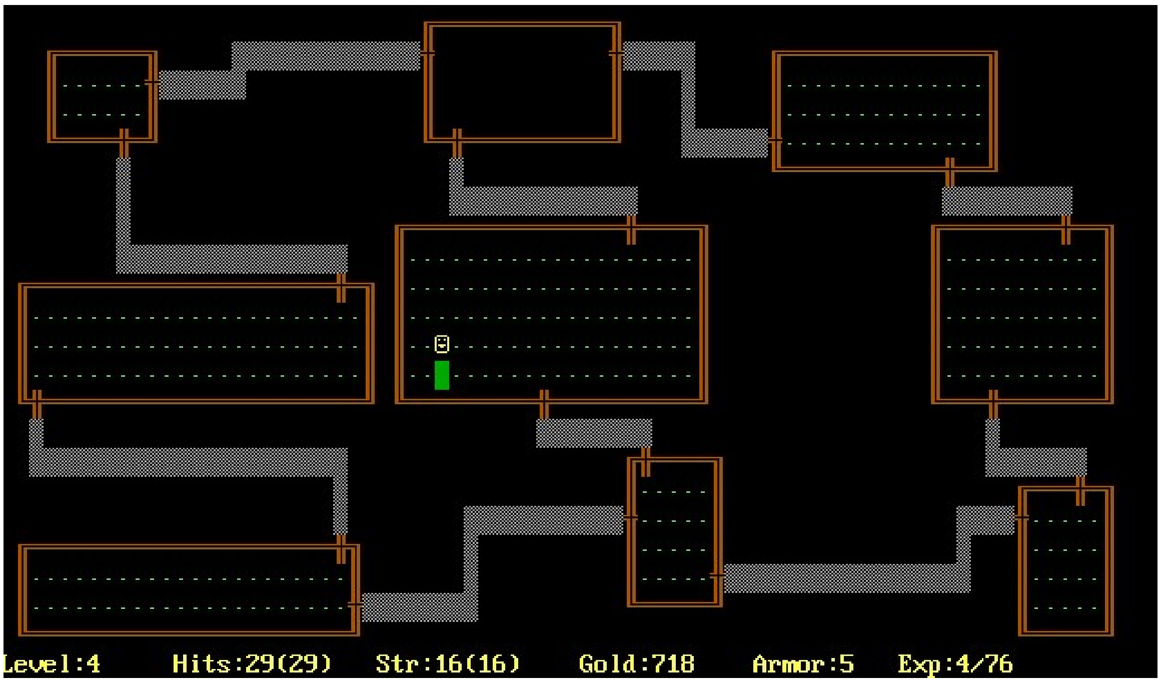

[27]Figure 11: Image to show how a modular environment works

2.3 Introduction to procedural generation

Another main topic of this project is related to procedural generation, which consists of data creation using an algorithm during runtime, being opposed to manually creating the data[28]. This data is a pseudo-random process that results in an unpredictable range of possible spaces, taking more importance the concept of randomness, due to it taking from a few parameters a large number of types of result.

[25] Image taken from: "Using Texture Atlases"- Game Developer - March 27, 2010
[26] Science Direct - "Game environment art with modular architecture" - Nataska Statham - March 2022
[27] Image taken from: "A Guide to Creating Modular Environments in Unreal Engine" - Máté Válent - November 27, 2022
[28] Medium - "Procedural Generation: An Overview" - February 19, 2021

Although there is no information about which was the first game to apply this technology, there is a game that has a very similar approach. The game is Rogue, from 1980, which defined the genre of rogue-like games. This game featured procedurally generated levels where the player navigates a labyrinth environment, creating a big dungeon.



[29]Figure 12: Image of the game Rogue

2.4 Procedural Generation Techniques Adapted to Game Development

The first step of developing each kind of project is to set the definition of the concept it is going to be represented in the production. In this case, the concept that will be developed and worked with is the procedural generation, which consists mainly of a computer algorithm that automatically generates content based on some pre-written instructions, also named procedures. This technique has been having bigger relevance with the pass of time, due to the fact that it has allowed the users to create many tools that with other pipelines would have been less efficient to work with.

Although there is a wide variety of applications where procedural generation could be implemented, the applications that will be shown are mostly related with the scope of this project, so they are mainly focused on environment generation[30].

2.4.1 Terrains

[29] Image taken from: "Rogue" - Wikipedia - August 31, 2019

[30] Fernando Bevilacqua, Marcos Cordeiro d'Ornellas, Cesar Pozzer - "A survey of procedural content generation techniques suitable to game development"- November 2011

This is an important part of the content of the whole game, due to it is the area where it will be played and plays an important role of keeping the player motivated to explore and discover new places.

Some procedural techniques that can be seen in other types of content are also applied in the videogames, like the noise, L-systems and fractals, being combined or implemented on their own to get results like an erosion effect.

Although it looks like adding a procedural effect is displaying randomness, the reality is pretty different, so it can be parametrically controlled. In coding, parameterization is the combination of multiple methods using a parameter that will pass the special value. In this case the group of methods perform similar actions that are different only in their internal values, numbers or operations[31].

To achieve this process there are different methods, such as adjusting division limits or controlling the distortion that can be generated. Create other parts of the environment such as caves, it can also be generated using volumetric discrete data structure.

An optimal solution to this process is to make an assisted method that uses a compact vector-based model that can be used to efficiently control the generation of the terrain, using control curves with the corresponding elevation gradient and noise parameters attached to them, generating a set of maps that are further combined in order to produce the final terrain[32].

In the industry this process of generating terrain is most used in videogame genres like roguelikes and sandbox[33]. In the case of roguelikes is due to the fact that a mechanic of the game is the randomness of each match, so it is also applied to the scenario, resetting the levels each time the player starts a new game. Some known roguelikes that apply this technique are:

- Crypt of the NecroDancer. This roguelike is combined with the rhythm, where the player moves the character to the beat of the music.

[31] Refactoring Guru - "Parameterize Method"

[32] Fernando Bevilacqua, Marcos Cordeiro d'Ornellas, Cesar Pozzer - "A survey of procedural content generation techniques suitable to game development"- November 2011

[33] Jasmine Bonner - "10 Games with Procedural Generation" - March 9, 2022

[34] Figure 13: Front page of the videogame Crypt of the NecroDancer.

- Enter the Gungeon. This is a bullet-hell roguelike with the dungeons being procedural generated. In the situation of this game the dungeon rooms are separately pre-set, but the order of its layout is completely random, giving the feeling that each run is in a different place than the last one.

[35] Figure 14: Front page of the videogame Enter the Gungeon.

- The Binding of Isaac. This game benefits from randomized loot, bosses and the layout. The scenarios are built more or less the same as the last point, but adding the condition of secret rooms, which need some conditions to be displayed on screen.

[34] Image taken from "Crypt of the NecroDancer: Nintendo Switch Edition" - Nintendo Switch - February 8, 2018
[35] Image taken from "Enter the Gungeon" -Nintendo Switch - December 18, 2017

[36] Figure 15: Front page of the videogame The Binding of Isaac.

In the other case, the sandbox games, a pillar is that every time the player starts a new world, it is generated differently every time, being identified with a seed, which is like a direction that makes each world unique. Some games that apply this process are:

- Minecraft. This is a survival game which can be played single player or with more in a world that each time is created is completely different, so the location of the resources and wonders are randomized each time, having interesting combinations in the landscapes.

[37] Figure 16: Front page of the videogame Minecraft.

[36] Image taken from "The Binding of Isaac: Repentance" - Epic Games Store - March 31, 2021
[37] Image taken from "Minecraft" - Nintendo Switch - June 21, 2018

- Terraria. This game is a 2D adventure that has a wide variety of unique bosses to conquer. The fact that the terrain is procedurally generated makes each bossfight unique, due to the fight area changes each run.

[38]Figure 17: Front page of the videogame Crypt of the NecroDancer

- No Man's Sky. This title can boast an entire galaxy procedurally generated, having unique planets, flora and fauna inhabiting them.

[39]Figure 18: Front page of the videogame No Man's Sky.

[38] Image taken from "Terraria" - Nintendo 3DS - December 10, 2015
[39] Image taken from "No Man's Sky" - Instant Gaming - August 12, 2016

2.4.2 Cities

The cities are the detonation of the human presence in the scenario and widely used in many games as a decoration, but the process of modeling the whole city is a time consuming task and it requires many aspects and details to take into account. As a consequence of it, it is important to generate a diversity of buildings, roads and places according to the worldbuilding and design of the game.

A solution to this problem is to generate a complex road network for cities based on Bezier curves and taking as reference real world data. In the case on a Bezier curve, it consists of a mathematical description of a smooth curve that is defined by representative points[40].

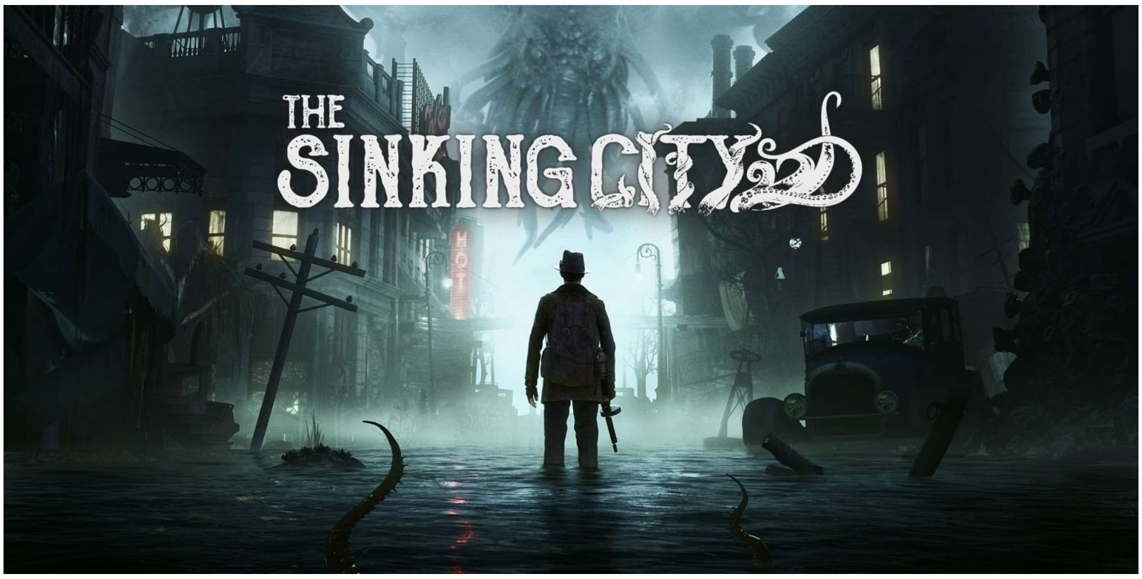
After the implementation of that method, the curves are interpolated by finding important characteristics and enhancing them such as interceptions. During the process new elements are added taking as base these curves in a real-time process by using polygonal approximation.

Once the roads and sidewalks are established, it is important to generate the urban ecosystems, using the interactive procedural system. This system consists mainly of using the process of urban model generation, a socio-economical and geometrical simulation approach that is used in order to generate the urban model.

In the videogame industry, each title released that uses this technology is different, due to the companies developing it in an internal way. Some games that apply the procedural generation of cities:

- The Sinking City. What this horror game does for the city's environment is the creation of some building prefabs using tools and keeping them on disk. Then the tool puts them along the streets using presets that contain the rules for building placement, ordered and chaotic environment, noise generation and bump on roads.

[40] Computer Hope - "Bézier Curve" - July 2, 2022



[41] Figure 19: Front page of the videogame The Sinking City.

- 7 Days to Die. The map generation of this survival game is based on seed generation, meaning that each world the player starts is different from the last one. The process used is using the cells rule, which the map is broken into cells whose size is specified by the source code, including specific rules to set a value.

[42] Figure 20: Front page of the videogame 7 Days to Die.

2.5 Creation of an immersive environment

During the process of 3D scenarios and levels creation a key element of the process is engaging the players with memorable and exciting experiences. What makes an

[41] Image taken from "The Sinking City" - Nintendo Switch - September 12, 2019
[42] Image taken from "7 Days to Die" - Instant Gaming - December 13, 2013

environment well designed is developing a coherent world at attention-grabbing levels, but there is also the doubt of how to set this immersion in a procedural environment.

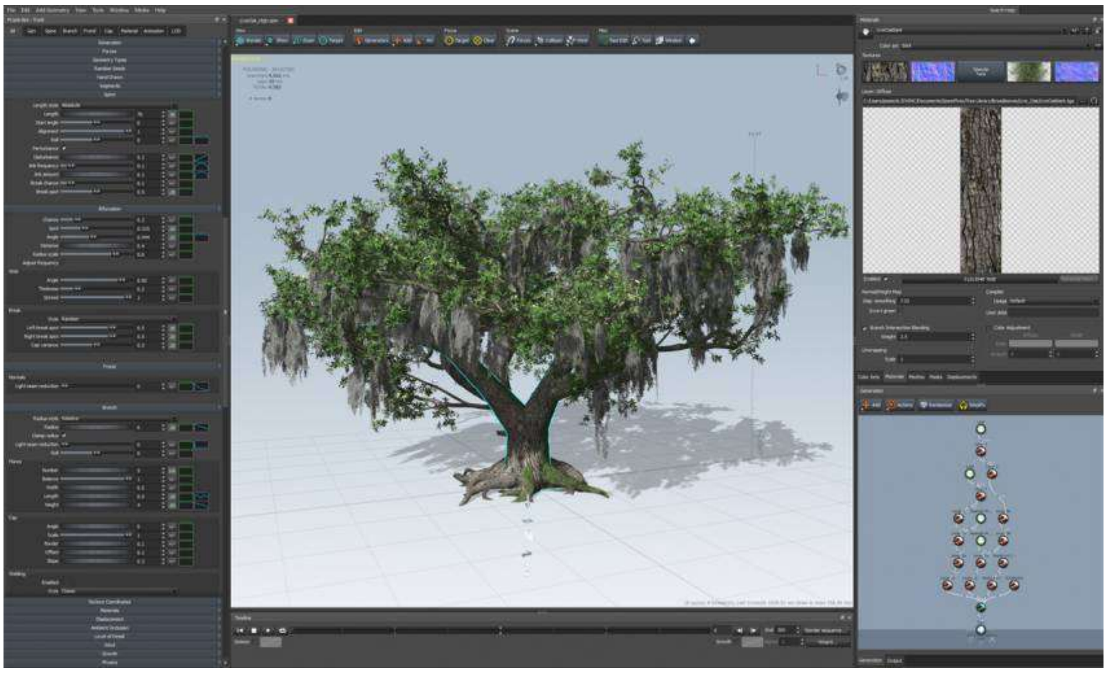
To implement the immersion experience to the player inside an environment procedurally generated is with the game design and taking into account three aspects, the player, the mechanics of the game and the own content of the game, from the narrative to the audio.

One of the best examples of procedural games that work with procedural generation are those that have exploration as one of the mechanics like open world and some sandbox games. As focusing more on the creation of the map, making a procedural creation allows it to imitate nature better, improving its performance and giving more replayability[43].

2.6 Existing tools

As procedural generation is not a new concept in the industry, there are tools that have been developed with this purpose in mind. There is a wide variation of software that consist on third party libraries and tools, which can be defined as content, features, functionality and components that are external to the original application and are created by external developers (they are also known as "third parties")[44].

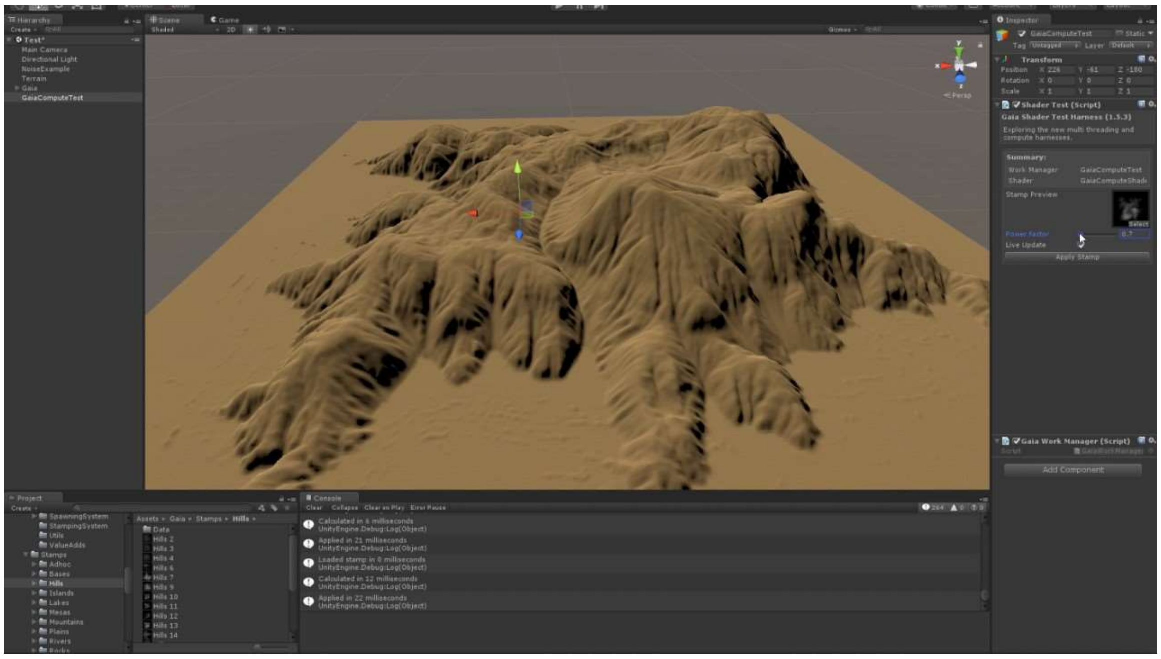
Some of the most popular software used in the industry are SpeedTree and Gaia. In the first case, SpeedTree is a program to 3D model with the objective of designing foliage, animations and architecture in real time[45].



[43] RiseAngle Team - RiseAngle - "Generating Excitement: The Impact Of Procedural Level Generation on Player Experience" - March 10, 2022
[44] Wolters Kluwer - "Third Party Libraries"
[45] Speedtree - "Speedtree games"

[46]Figure 21: Capture from the webpage of the software

The other software used is Gaia, which is a library that contains a landscape, terrain and scene generation system for the Unity engine. Although this software allows the user to create a lot of procedural content, it also allows changes to be made manually, using either sample assets or assets created by the user[47].



```
Calculated in 6 milliseconds
UnityEngine.Debug:Log(Object)
Applied in 21 milliseconds
UnityEngine.Debug:Log(Object)
Loaded stamp in 0 milliseconds
UnityEngine.Debug:Log(Object)
Calculated in 12 milliseconds
UnityEngine.Debug:Log(Object)
Applied in 22 milliseconds
UnityEngine.Debug:Log(Object)
```

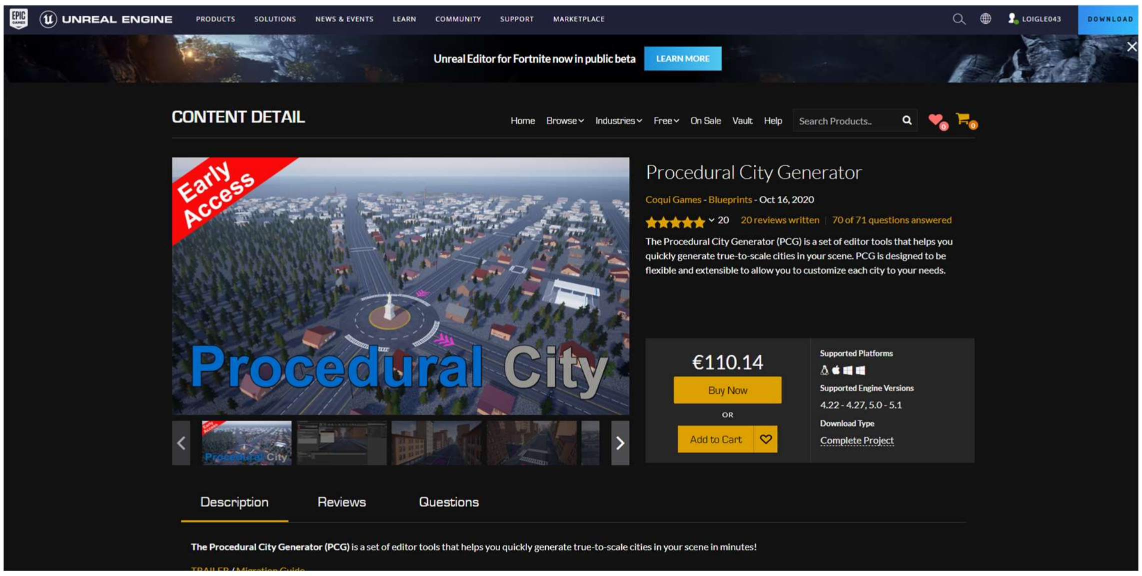
[48]Figure 22: Capture from the webpage of the software

Some third party software that is not as known as the programs mentioned above are also interesting due to their approach to the goal of this project. An example of this software is a procedural city creation in the Unreal Marketplace, which allows the creation of cities in base of models previously created, but the main problem is that this plugin is only focused on creation of cities, so it does not work well for different environments.

[46] Image taken from "Reference Manual Overview" -Speedtree documentation - 2014

[47] Unity Asset Store - "Gaia 2 - Terrain & Scene Generator"

[48] Image taken from: "Gaia - AAA terrain generator, procedural texturing, planting and scene creation" - Unity forums - January 24, 2017



Unreal Editor for Fortnite now in public beta

Procedural City Generator

Coqui Games - Blueprints - Oct 16, 2020

20 20 reviews written | 70 of 71 questions answered

The Procedural City Generator (PCG) is a set of editor tools that helps you quickly generate true-to-scale cities in your scene. PCG is designed to be flexible and extensible to allow you to customize each city to your needs.

€110.14

Supported Platforms

Supported Engine Versions
4.22 - 4.27, 5.0 - 5.1

Download Type
Complete Project

The Procedural City Generator (PCG) is a set of editor tools that helps you quickly generate true-to-scale cities in your scene in minutes!

[49]Figure 23: Capture taken from the Unreal Marketplace.

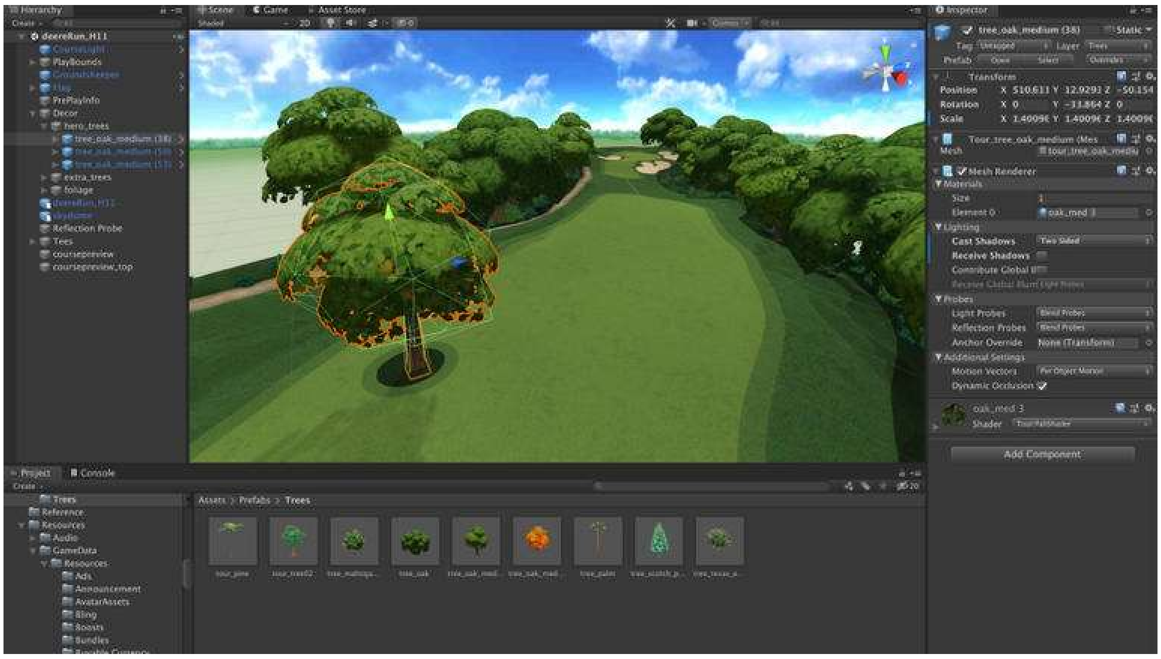
Those programs mentioned above are very complete and allow the user many options, each one having their own features and possibilities. As it can be seen, they are either purchasable in the marketplace of the engine they are used for or in their own webpage, having big price variations.

2.7 Technology used in the industry

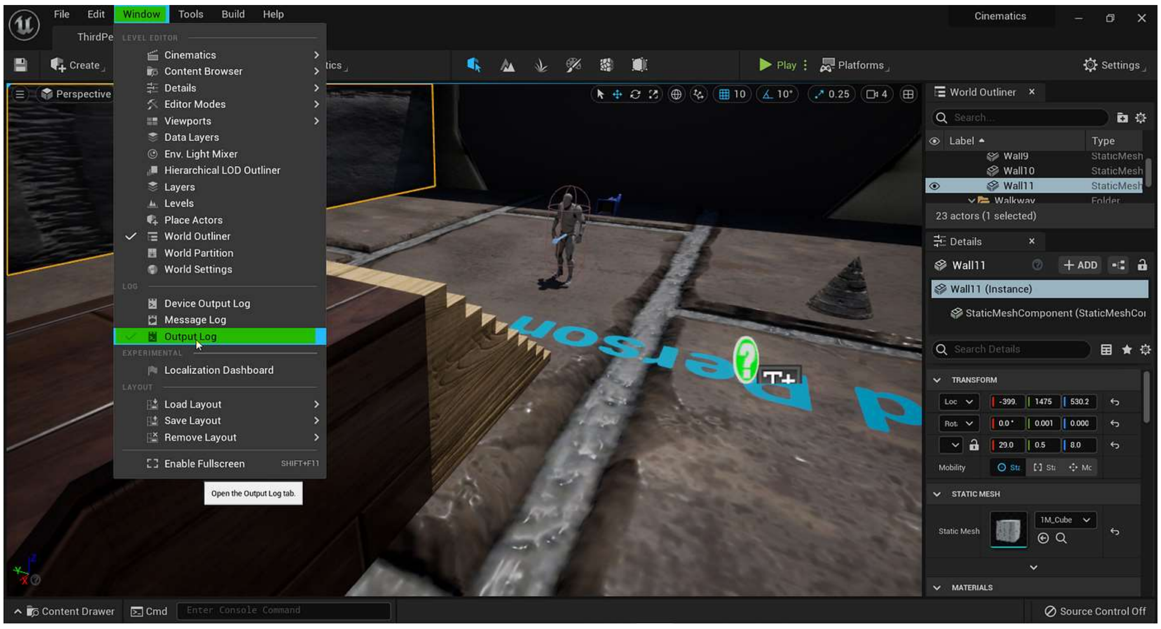
In this section there will be the most used programs by the videogame industry, allowing some to generate procedural content. To start with engines, the most known can be considered:

- Unity. This engine was firstly introduced to develop 2D or 3D games, but nowadays is used for other purposes like interactive simulations and software. The way it works is with scripts that use its internal programming language, C#, an extension of the C language, but with features previously created by the engine[50].

[49] Image taken from Unreal MarketPlace - "Procedural City Generator"
[50] Unity webpage


[51]Figure 24: Capture of Unity.

- Unreal Engine. This engine is mostly focused on 3D content creation in real time. The scripting system it uses is based on blueprints, which are a way of visual programming, as well as C++ scripts, which use the language C++, a variation of C but focused on software and games development[52].


[53]Figure 25: Capture of Unreal Engine.

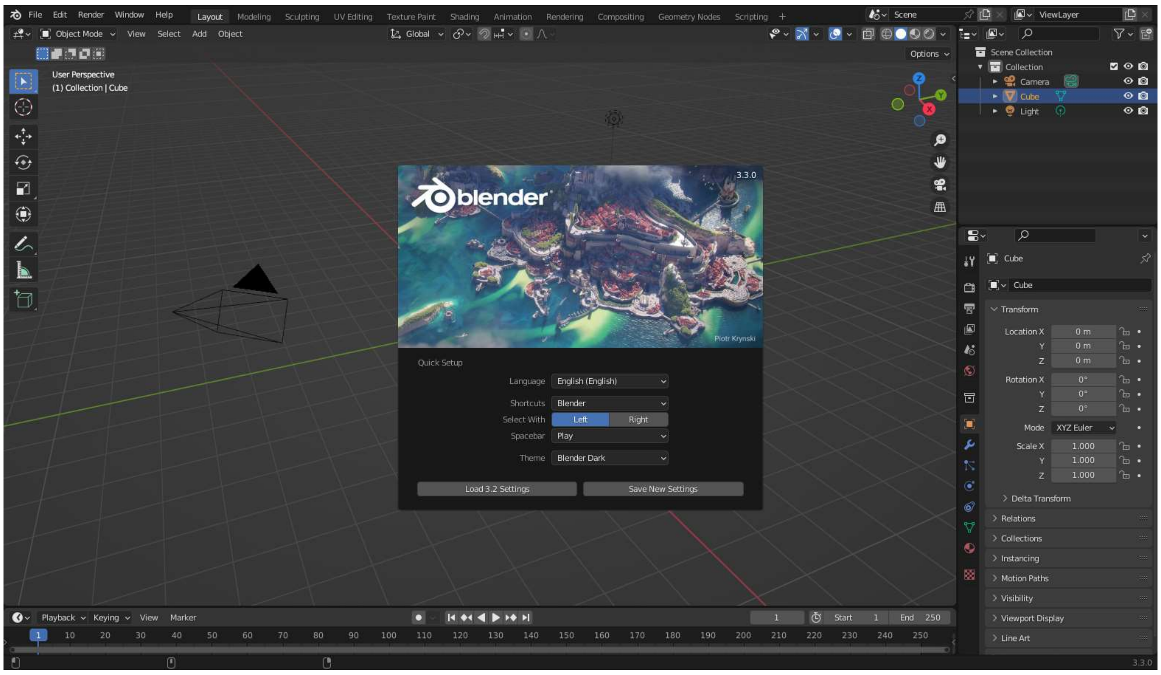
The game engines correspond a great importance to videogame industry, due to they allow the user the creation of software more easily than making it from scratch. With the

[51] Image taken from: "Why struggle with a cdn when you have games to create?" - Unity webpage
[52] Unreal Engine - "Unreal main page"
[53] Image taken from: "How to Capture High Resolution Screenshots in Unreal Engine 5?" - Medium - Prashant Mhatre - September 12, 2021

pass of time, the engines have evolved more not only to make games, but also other features like visualizing data or products.

In terms of 3D modeling there is also a big variety of programs, each one with their own features:

- Blender. This is a free and open source 3D computer graphics software that is used for modeling, animation and visual effects, among others. This tool is widely used due to its versatility and unique characteristics, such as the add-ons, which are extensions of the program that facilitate some tasks[54].



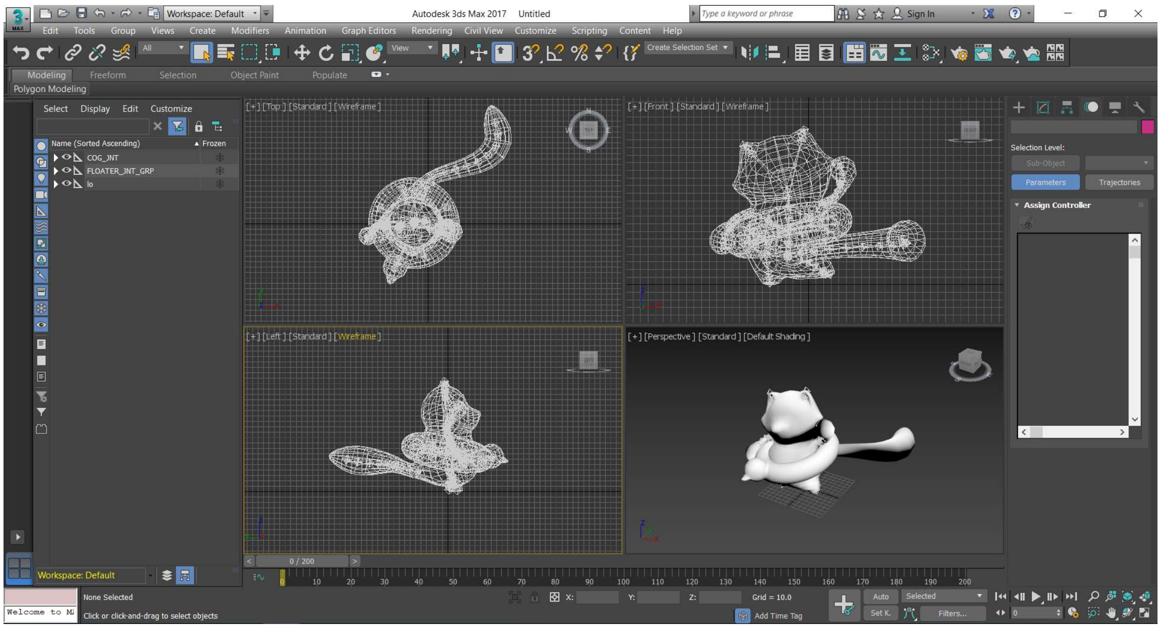
Quick Setup
Language: English (English)
Shortcuts: Blender
Select With: Left | Right
Spacebar: Play
Theme: Blender Dark
Load 3.2 Settings
Save New Settings

Location X 0 m, Y 0 m, Z 0 m
Rotation X 0°, Y 0°, Z 0°
Mode XYZ Euler
Scale X 1.000, Y 1.000, Z 1.000
Start 1 End 250

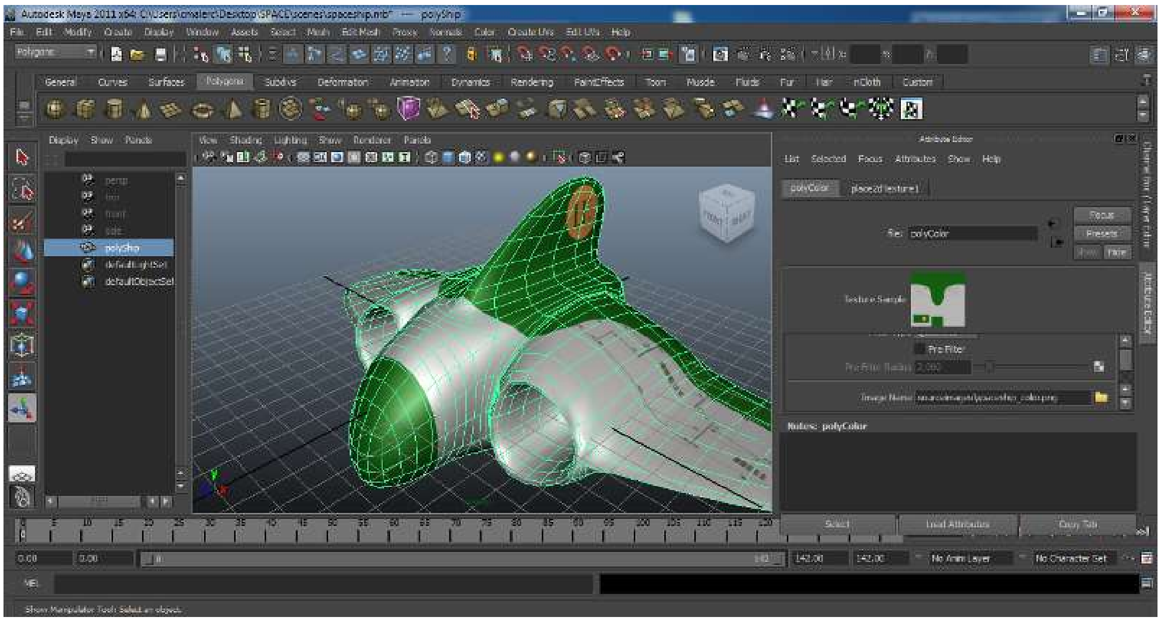
[55]Figure 26: Capture of Blender.

- Autodesk 3Ds Max. This is a 3D program that focuses on modeling and some animation that enables the creation of 3D designs with a great result. The unique feature of this program are the modifiers, which are internal tools to make the modeling easier[56].

[54] Blender - “About window from Blende website”
[55] Image taken from: “Blender Screenshots” - Github Topics - username: seanpm2001 - November 16, 2022
[56] Autodesk - “Autodesk 3Ds Max”

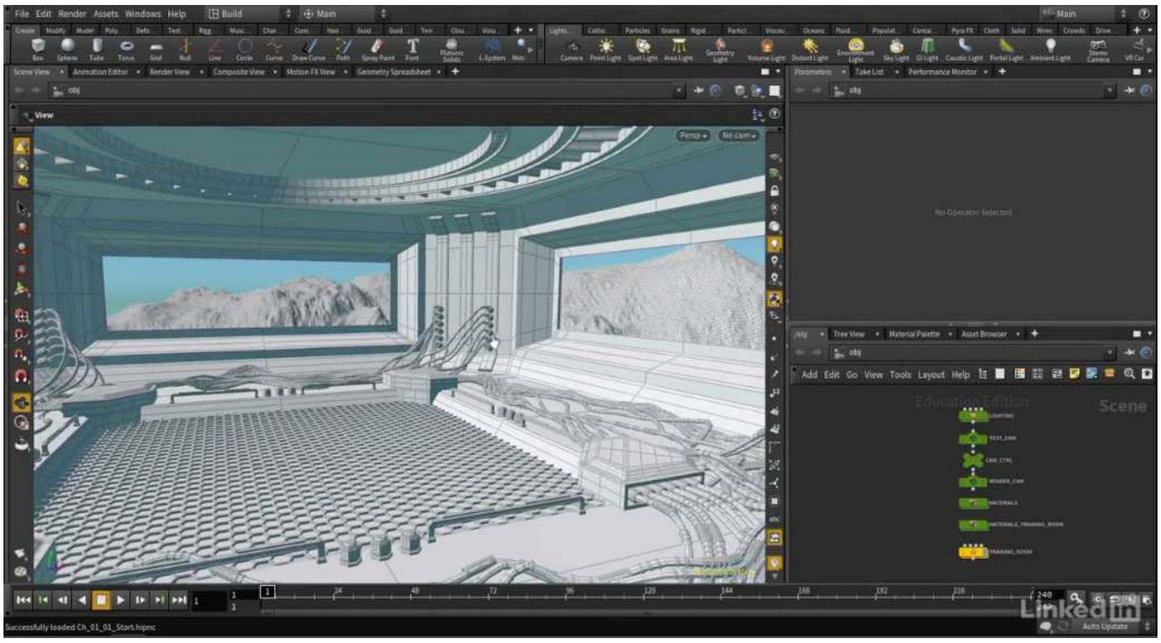

[57]Figure 27: Capture of 3Ds Max

- Autodesk Maya. This software is from the same company as 3Ds Max, but this is mainly focused on 3D animation and modeling, being the most popular for animating characters for games as well as movies[58].

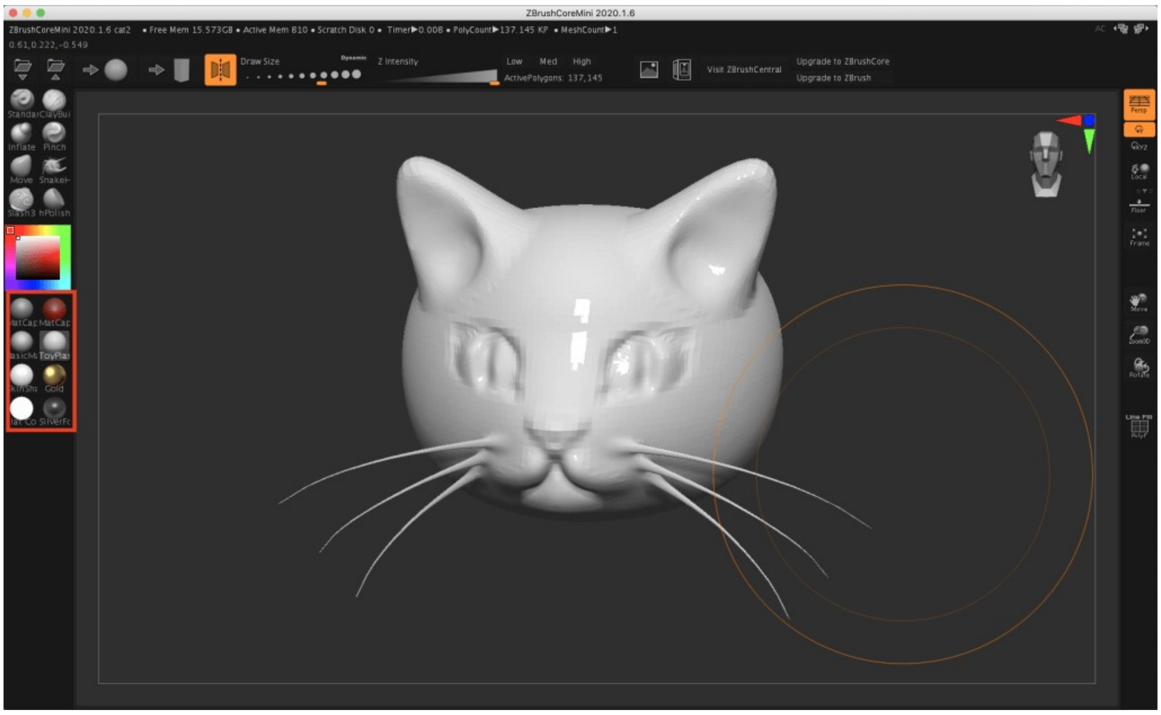

[59]Figure 28: Capture of Maya

- Houdini. This is another 3D computer graphics program that works with nodes, but it is mainly focused on procedural workflow. The nodes of this program are wired into networks which define a recipe that can be tweaked to refine the outcome creating unique results[60].

[57] Image taken from: "3ds Max, 3D Object Export" - Snap Docs
[58] Autodesk - "Autodesk 3D Maya"
[59] Image taken from: "Maya export" - x3dom
[60] SideFX - "Houdini | Home Page"


[61]Figure 29: Capture of Houdini

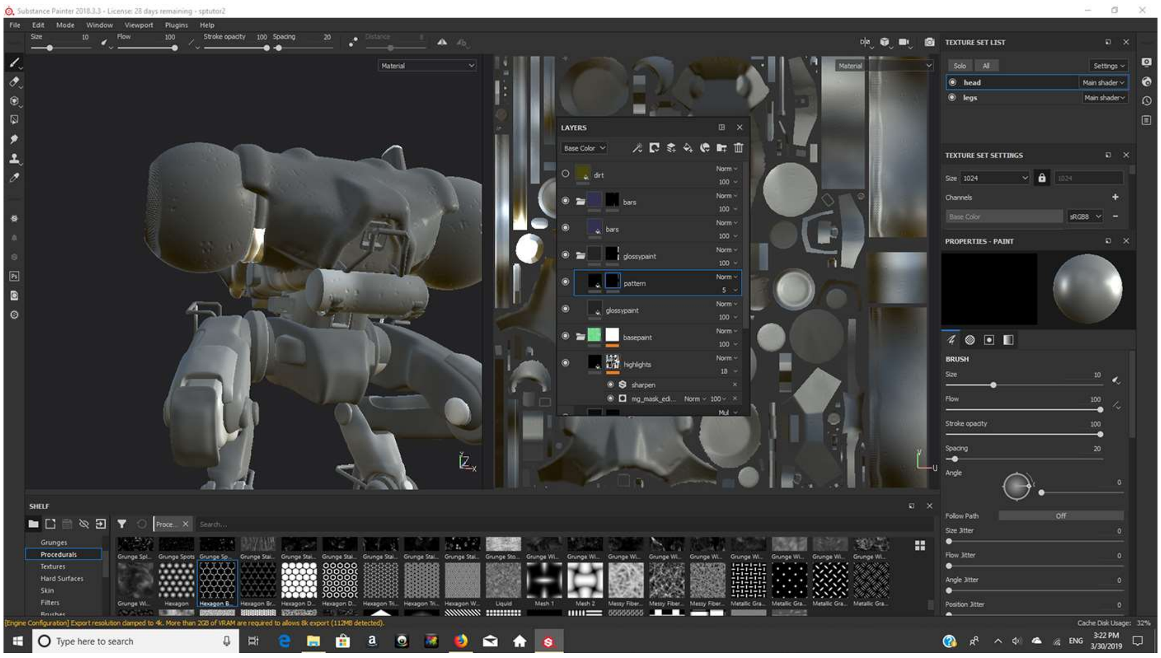
- Zbrush. This tool is also a 3D software which allows the creation of 3D models, but in this situation, it is mainly used for sculpting, so it is ideal for the creation of the high poly version. This powerful software is one of the most used in the art of videogames with multiple options and plug-ins to modify the meshes[62].


[63]Figure 30: Capture of Zbrush

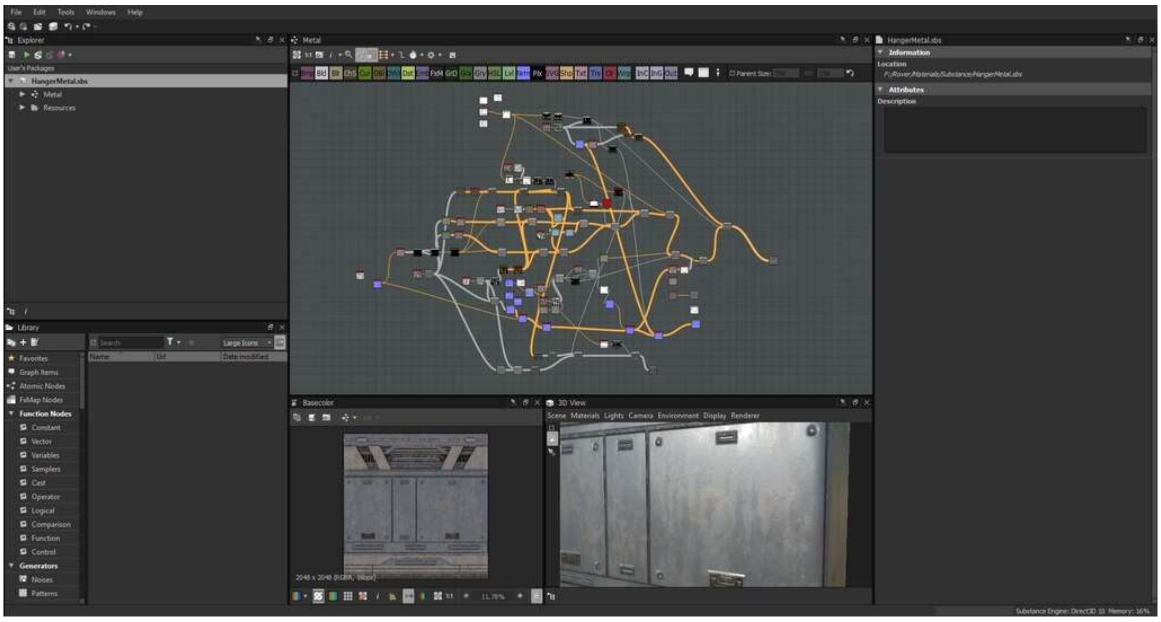
[61] Image taken from: "Houdini Essential Training" - SideFx - June 27, 2017
[62] Pixologic - "Zbrush"
[63] Image taken from: "Introduction to Sculpting with ZBrushCoreMini" - Styly Magazine

For the creation of textures and materials there are two main programs that consist on:

- Adobe Substance 3D Painter. It is a paint engine, recently bought by Adobe, which allows the user to paint the UVs of the 3D models previously created and the possibility to create textures from scratch with thousands of combinations and options[64].


[65]Figure 31: Capture of Substance Painter

- Adobe Substance Designer. This software is from the same company than Painter, but this program is focused on material creation, having also a production based on nodes, it allows to implement procedural generation to the textures[66].



[64] Adobe - "Adobe Substance 3D Painter"

[65] Image taken from: "Substance Painter Screenshots" - Emily Reinhardt - April 1, 2019

[66] Adobe - "Substance 3D Designer Documentation"

[67]Figure 32: Capture of Substance Designer

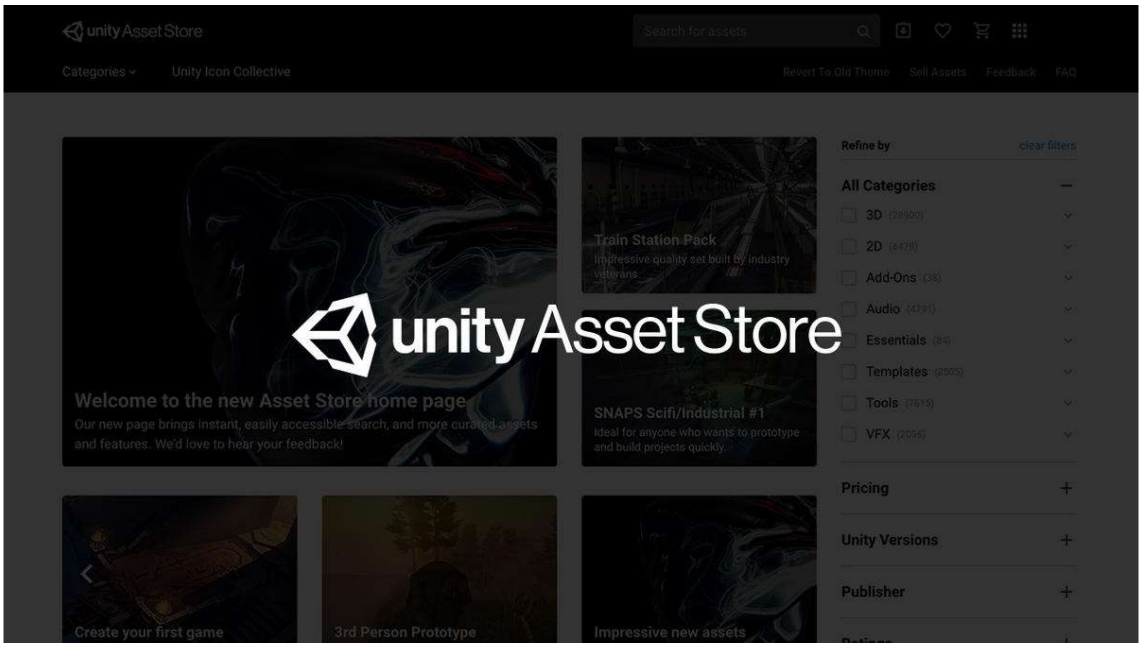
2.8 Publish a tool

Once the project is finished, it is time to upload it and accessible to the target audience to be able to download it. As this project will be developed in a commercial engine, each one has its own market, which makes available their creations. To publish the tool, it is necessary to complete the requirements of each engine. The most common ones are Unity and Unreal, which already have their own pipelines to publish any feature.

In the case of Unity[68], there is the Asset Store and the workflow consists on five steps:

- **Creation of a publisher account**. This is a kind of account different from the user one. This consists of giving an user access to creating and publishing packages, setting up payouts, viewing sales and revenues, and managing all aspects of the profile.
- **Creation of a new package draft**. This step consists in creating a draft package in the own Asset Store. An Unity package is a collection of files and data from Unity projects which are compressed and stored in one file.
- **Upload the assets into the package**. In this situation the user will need to get the Asset Store Tools package and select the upload option to start with the publishing. Here the user will need to insert its own credentials and validate the package to check that there are no errors.
- **Fill the package details**. It is a window that allows the user to note different details of the package created, like the state of the version or even the price it will have in case it is monetized.
- **Submit the package for approval**. This step is changing the status of the draft package to review, having few options like publishing it automatically once it has the approval and sending a message to the Asset Store team to add some notes of how it works.

[67] Image taken from: "Substance Designer 6" - Steam Community - user: Doomologist - April 16, 2019
[68] Unity Documentation - "Publishing to the Asset Store" - March 17, 2023



[69] Figure 33: Web of the Unity Asset Store.

In the other case, Unreal Engine, there is the Unreal Marketplace, where the user can choose among all the platforms that Unreal has supported[70].

[71] Figure 34: Web of the Unreal Marketplace.

This publishing process is a bit more complex than the one mentioned above, due to it gives many options depending on the scope of the project created, but all have the same steps in common:

- **Build**. Compilation of the executable file for the selected platform (PC, Xbox, Switch, etc).
- **Cook**. Executes the Editor in a special mode.
- **Stage**. Copies the executables and content to a staging area.
- **Package**. Packages the project into a platform's native distribution format.
- **Deploy**. Deployment of the build to a target device.

[69] Image taken from "The Asset Store, new and improved, starting today" - Unity - Janis Daniel Pascal Ewald, Eduardo Oriz - April 30, 2019
[70] Unreal Engine - "Sharing and Releasing Projects" - 2023
[71] Image taken from "Unreal Engine MarketPlace" - Epic Games Store - 2023

- **Run**. Starts the packaged project on the target platform.

When the user needs to package the final version of the project, it also has available different package methods, due to the fact that it is also important to debug in release to check for any bugs or errors.

2.8.1 Portfolio showcase

The portfolio is a key tool to show the projects made, being academic, professional or in a freelance way. This consists in a combination of materials that exemplifies the capabilities, experience and way a person works.

During the process of a project is important to document the process, so it can be useful for the portfolio later, but a part from this, the upload of a portfolio should have the following points[72]:

- **Statement of originality**. This is a short paragraph describing the work made with the own words of the user. Here is remarkable to indicate the parts that do not want to be copied.
- **Work philosophy**. A small description of the user's profile and the industry it wants to work in.
- **Resume**. This is the overview of the user's skills and experience,it is usually written as a presentation letter.
- **Work samples**. Those are the items that will be shown in the portfolio, so the most recommendable is to upload the projects with the most quality on it, prioritizing quality over quantity.
- **Academic Plan of Study.** This consists in showing the studies and preparation of the user, like degrees, courses, etc.
- **References**. A list of people that recognize the work made and corroborate the skills of the user.

As for the portfolio itself, there are different ways to build it, from web pages that build one with templates to the creation of one by scratch. Although there are many options, not all options are for all the users, due to it depends on which profile the user has. For the videogame industry the most common ways to showcase a portfolio are:

- **ArtStation**. This page is the most common platform for videogames, films, media and entertainment, but is mainly focused on artists, due to it being focused on showing visual content. In this portal, each user owns an account and there is a system for contacting people or finding jobs related to art. Here each page is different due to it has many options to customize the works, like creating albums that cover a group of works, or changing the theme.

[72] Clarke University - "What is a Portfolio?"

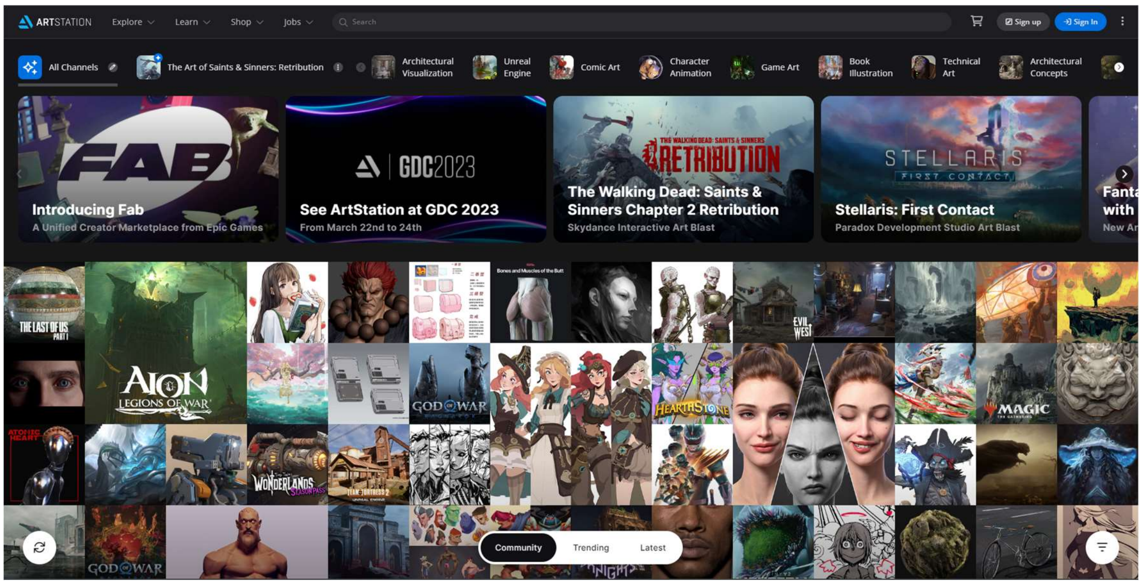

ARTSTATION
Explore
Learn
Shop
Jobs
Search
Sign up
Sign In
All Channels
The Art of Saints & Sinners: Retribution
Architectural Visualization
Unreal Engine
Comic Art
Character Animation
Game Art
Book Illustration
Technical Art
Architectural Concepts
Introducing Fab
A Unified Creator Marketplace from Epic Games
See ArtStation at GDC 2023
From March 22nd to 24th
The Walking Dead: Saints & Sinners Chapter 2 Retribution
Skydance Interactive Art Blast
Stellaris: First Contact
Paradox Development Studio Art Blast
Community
Trending
Latest

Figure 35: Capture of ArtStation website.

- **Github pages**. This tool is mainly used for uploading projects and to share code, but it has a web builder which allows you to create a website with a free domain, allowing the user to customize the structure to make it more personal and unique. To build a page, the user first needs to create a repository, which is a work space of a project where is used to upload different versions of the project. In this situation, the github pages have two options of building, one being with the use of editable templates that come with the application using markdown language, and the other by programming in HTML.

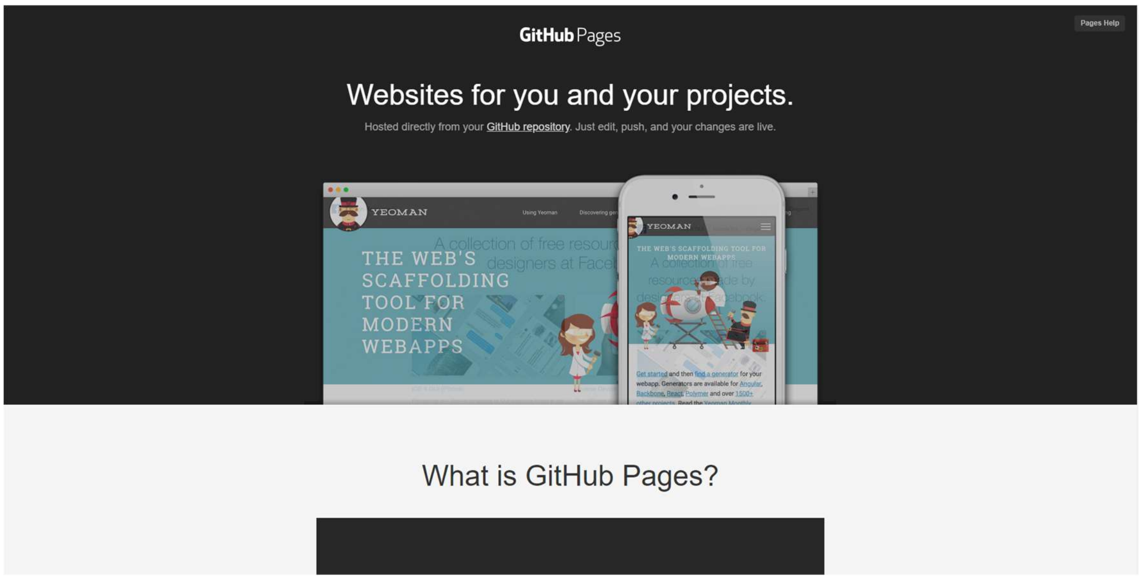
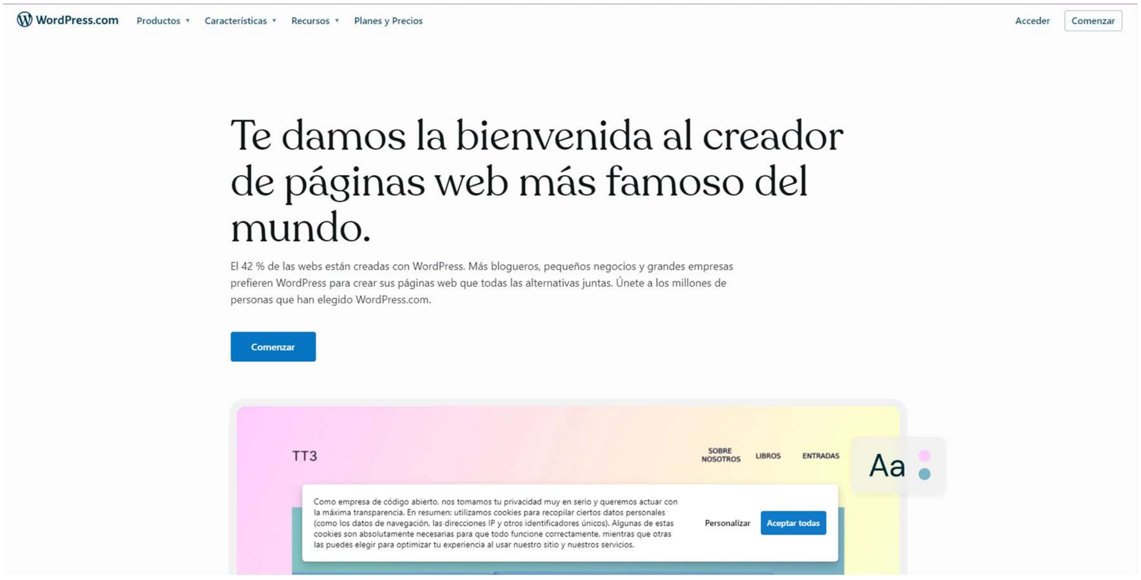

GitHub Pages
Pages Help
Websites for you and your projects.
Hosted directly from your GitHub repository. Just edit, push, and your changes are live.
YEOMAN
THE WEB'S SCAFFOLDING TOOL FOR MODERN WEBAPPS
What is GitHub Pages?

Figure 36: Capture of Github pages website.

- **WordPress**. Although this is a website builder, it is more focused on the creation of a blog, but also gives freedom of creation for any kind of web. This program is also self-hosted, but it is not totally free, so the user has to purchase a web hosting and a domain name to start.



Te damos la bienvenida al creador de páginas web más famoso del mundo.

El 42 % de las webs están creadas con WordPress. Más blogueros, pequeños negocios y grandes empresas prefieren WordPress para crear sus páginas web que todas las alternativas juntas. Únete a los millones de personas que han elegido WordPress.com.

Como empresa de código abierto, nos tomamos tu privacidad muy en serio y queremos actuar con la máxima transparencia. En resumen: utilizamos cookies para recopilar ciertos datos personales (como los datos de navegación, las direcciones IP y otros identificadores únicos). Algunas de estas cookies son absolutamente necesarias para que todo funcione correctamente, mientras que otras las puedes elegir para optimizar tu experiencia al usar nuestro sitio y nuestros servicios.

Figure 37: Capture of Wordpress website.

3. Project Planning

Once there are different pipelines investigated, the next step to the development of any project is to define a plan to manage three main pillars, the scope, the budget and the timeline, as well as the prevention of possible constraints.

To start the planning it is important to understand the structure of the project that will be developed, taking into account how many people participate and how much time is available to start dividing the tasks. Project planning can be divided into three pillars[73]:

- Scope. This point determines what will be made during the project and what will be discarded. It is a balance between the vision of the project and the determination of what can be achieved with the time available.
- Budget. Defines which manpower and other resources required to meet the project goals to estimate the cost. This point will be explained later.
- Timeline. Consists on the length of time expected to complete the different phases of the project, including the milestones.

Before starting the project, it is important to declare which goal is going to be achieved, so the main goal for this project is a tool that could create environments ready to be used for a videogame and implement it in a 3D engine. The process to arrive at this objective is going to follow through the design and development is the scrum framework. This consists of a process that has been used to make complex product development since the early 1990s. Scrum is the most optimal option for management due to it allows to adapt the planning with the current state of the project. How this framework works is by using prescribed events to create regularity. All events defined are time-boxed events, so each event can have its own duration. During the process, there are also the sprints, which are time-boxed events with a determined duration, so each sprint should be the same length, and only there is a change of sprint once the previous is finished, so it allows better work organization.

The timeline will be divided into different milestones that will conform to a certain number of sprints each. First of all, a milestone consists of a set of deadlines set at the beginning of the production in order to give a clear goal. In the case of this project, there are three milestones, also named rubrics for this thesis:

- Rubric 1. This milestone consists of planning which is the goal of the project as well as how it will be organized to achieve the timelines. During this rubric the most important parts will be searching information about the final version of the project. There will also be an exploration and conceptualization for the different stages to define which vision the project is going to take.
- Rubric 2. Once all the concepts have been set up in the previous milestone, this rubric defines the production of the project itself, so it will be the phase where the base will be developed.

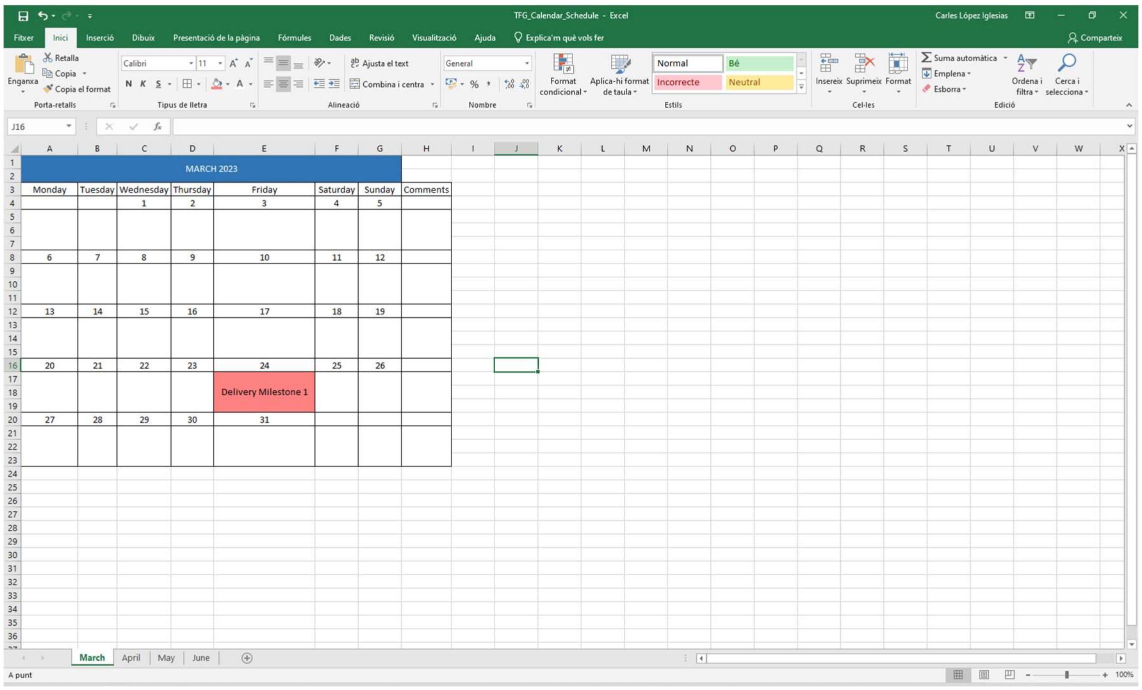
[73] Ben Lutkevich - TechTarget.com - "Project Planning: What is it and 5 steps to create a plan"

- Rubric 3. The final milestone will be the end of the development phase, meaning that the project is almost complete, but it also needs some polishing and optimization to reach the definitive version.

As management is a core key for the development of this project, there are two tools that will be essential to this task:

- Microsoft Excel/Google Sheets. This is a software program that uses spreadsheets for numbers and data organization with formulas and functions. It will be used for scheduling the timeline, defining the sprints to be completed and calculating the budget.
- Hack N' Plan. It consists of a project management tool specifically designed for videogame development teams, allowing the user to organize the design and development cycle.

To implement the management plan for this project it has been necessary to create an excel file that acts like a calendar, which is distributed by each sheet being a month with the main milestones written in the respective cells to note puntual objectives like deliveries or ideas that could be applied later.

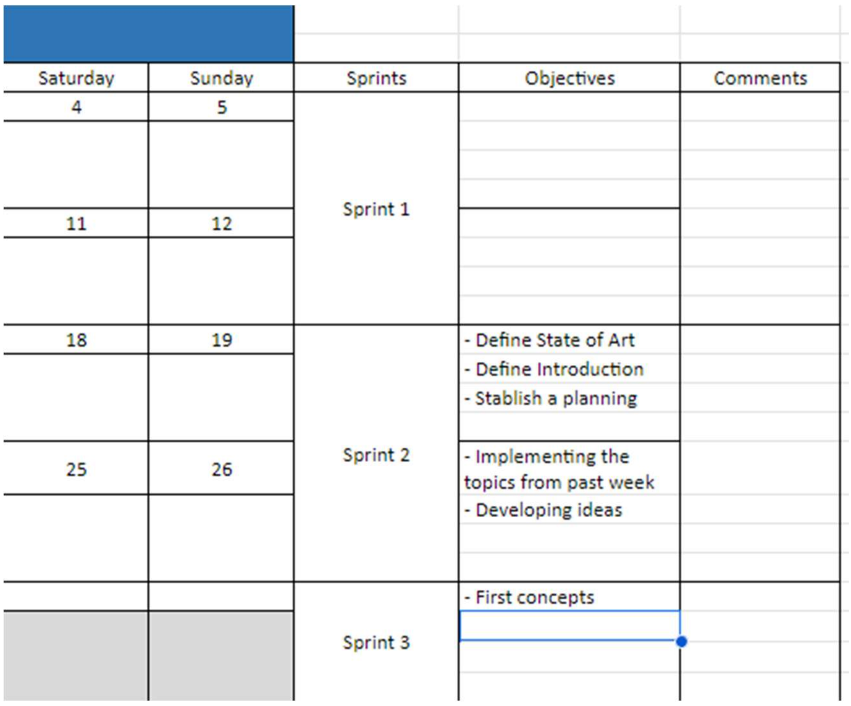


MARCH 2023

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Comments
		1	2	3	4	5	
6	7	8	9	10	11	12	
13	14	15	16	17	18	19	
20	21	22	23	24	25	26	
				Delivery Milestone 1			
27	28	29	30	31			

Figure 38: Capture of the calendar built in Excel.

As an annex to the calendar, there will be displayed the sprints with the objectives established corresponding to their duration. Next to them there will be a comments box to take notes of the progress or if it is necessary to make changes adapting to the project's progress.



Saturday	Sunday	Sprints	Objectives	Comments
4	5	Sprint 1		
11	12			
18	19	Sprint 2	- Define State of Art - Define Introduction - Stablish a planning	
25	26		- Implementing the topics from past week - Developing ideas	
		Sprint 3	- First concepts	

Figure 39: Capture of the sprints planification in the calendar.

To have a better understanding of the progress of the project and manage it in an efficient way, the software used will be HacknPlan, a project management tool for game development, brining game design and project management together to provide a better workflow and more intuitive organization. Although this application is thought for teams, it can also be used for single projects.

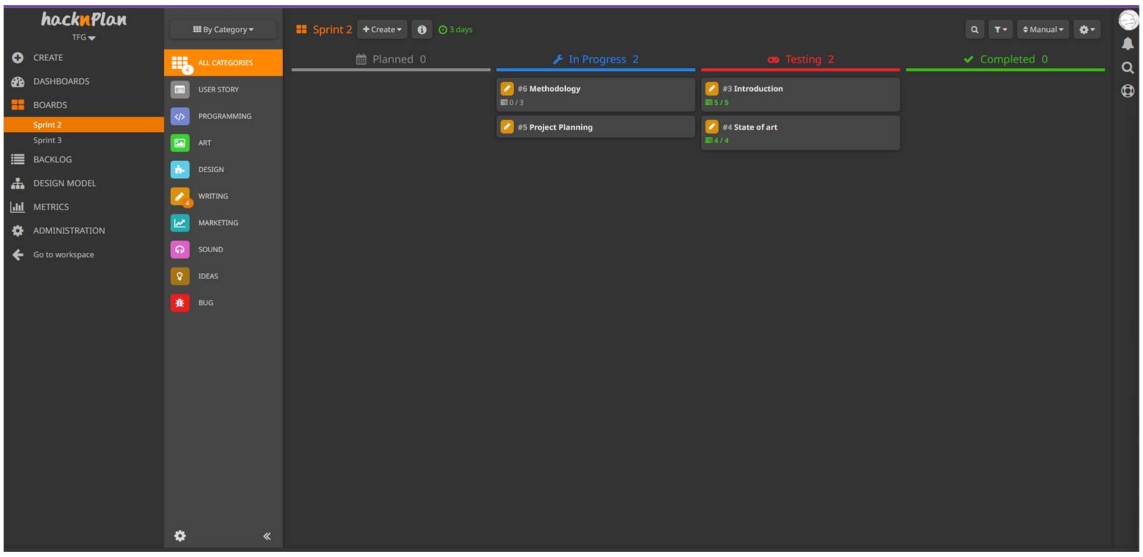
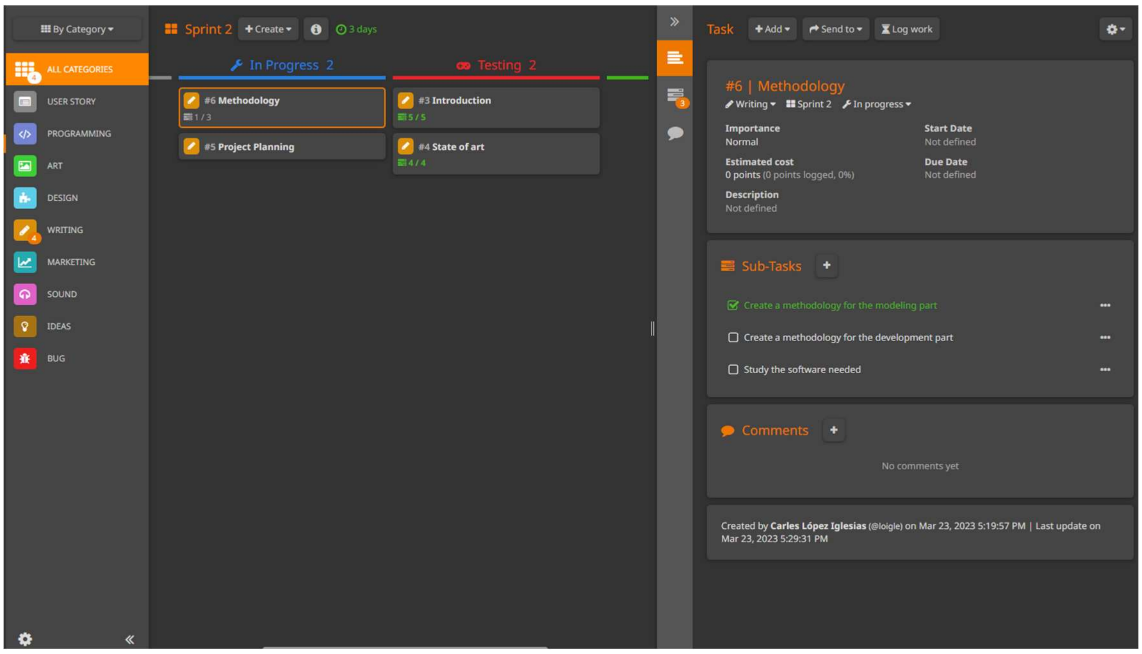


Figure 40: Capture of the management board in HacknPlan.

At first glance, this program is structured as if it was a physical dashboard divided into four stages, which are planning, in progress, testing and completed, with sticky notes where the user writes to them. Inside these stickers, the user can add sub-tasks, which come with a checkbox to see progress.



#6 | Methodology
Writing · Sprint 2 · In progress

Importance: Normal
Start Date: Not defined
Estimated cost: 0 points (0 points logged, 0%)
Due Date: Not defined
Description: Not defined

Sub-Tasks
- [x] Create a methodology for the modeling part
- [ ] Create a methodology for the development part
- [ ] Study the software needed

Comments
No comments yet

Created by **Carles López Iglesias** (@loigle) on Mar 23, 2023 5:19:57 PM | Last update on Mar 23, 2023 5:29:31 PM

Figure 41: Capture of the subtasks from the board in HacknPlan.

In the sidebar of the board there are multiple categories for the stickers, so the user has visual feedback of which part has more tasks. On the options of the project there are also other options that help the management, but the most used will be the "Boards" window, where the user can add sprints, setting up the timeline and being able to put a brief description to add context.

Although the objectives are displayed and set up from the beginning, there is a possibility of making small changes to the project to adapt better to the timeline, like developing discarded ideas or even adding new ones, as this also takes part of the development process.

3.1. DAFO

The DAFO analysis consists of an analysis tool used in project management to set its own characteristics. It is divided into four aspects:

- Strengths. Consists of the characteristics that are available and makes it easier to get the objective.
- Weaknesses. Defines the internal characteristics that are difficult to achieve the final result.
- Opportunities. Environment aspects that allow to get the smaller objectives to get to the final one.
- Threats. Contrary to the last point, they are the external aspects that are difficult in the development process.

This tool is useful to take a look at the different points of view of the project, the working environment and the team, being able to balance both situations to arrive at a final decision that is the most beneficial and counter possible weaknesses or setbacks.

As it can be seen in the definition of the points that conform the DAFO, it can be divided into two groups, being the internal aspects and the external aspects, corresponding each one with the perspectives mentioned above.

The objective of the DAFO is to identify the steps, actions and cautions to upgrade the management and planification of the project.

	Positives	**Negatives**
Internal Origin	**Strengths** New approach that could help the videogame industry adapt to new technologies. Mixing an artistic approach as is the 3D modeling with a technical approach which is the development of the tool.	**Weaknesses** As there are few tools in the industry similar to this project, the tool can look similar to others. Difficult to balance the artistic / technical parts of the project.
External Origin	**Opportunities** There are tools and software that make possible the objective of the project. Get experience in developing tools and improve the portfolio.	**Threats** Strict timelines and schedules that interfere the progress of the project. Trying to embrace a too big objective instead of going to achieve a more simple goal.

Table 1: DAFO

3.2. Risks and contingency plan

Although a project is strictly structured and with good timelines and management, it is not always possible to predict what will succeed in the long term or how the project will evolve. For this reason, among others, creating a risks and contingency plan allows to prevent different problems and threats that will come during the development process, making it a necessity for each project.

This table below represents the identified risks of this project, with its respective solutions, everything sorted from major to minor level of importance:

Risk	Solution
Adding too many features to the tool and not being able to arrive on time	Prioritize and define which functions of the tool have a key function or a secondary function
Technical problems with the software and hardware	Go to the university and ask for a computer that can stand the software needed or working on a virtual machine
Storage and file management of the project	Use a Google Drive file to upload different versions of the project and save them in the cloud
Not being able to achieve the objective of the project due to strict timelines	Organize the project development with management tools available

Table 2: Risks and solutions for the project

3.3. Budget analysis

This stage of the process consists of a detailed estimation of all the costs that are likely to be incurred before the project is completed. The costs taken into account in this analysis can be divided into three main categories:

- Hardware. This is the showcase of the devices that are needed for the project, as well as how it will be the amortization.
- Software. This point is how much the programs used in the project will cost.
- Salary. This is the amount of money that a professional needs to develop the project taking into account the time and planification.

Budget			
Hardware	Unit cost	Amortization	Cost sum
PcCom Silver	919,18 €	4 months	
Peripherals	34,99 €	4 months	
Screen MSI Optix 144Hz Full HD	159,00 €	4 months	
Total cost (Hardware)			1.113,17 €
Software	Unit cost	Amortization	Cost sum
Unreal Engine	- €		
Blender	- €		

Substance Painter	145,99 €	1 year	
Total cost (Software)			145,99 €
Salary	Cost per month	Duration (months)	Cost sum
Junior 3D Artist	1.250,00 €	4	**5.000,00 €**
Total cost of Budget			6.259,16 €

Table 3: Excel of the budget analysis

4. Project validation

This stage of the project refers to the process of confirming that an overall product meets its intended use and purpose. It can be also known as software quality control, which is a set of activities for ensuring and evaluating the quality of the product. It can be divided into different points[74]:

- **Unit Testing**. In this initial phase of testing, it consists of testing individual elements of the product, like individual functions or limited features, being the smallest testable part of any software.
- **Integration Testing**. This step is based on testing the interfaces and interactions between integrated components.
- **System Testing**. This part of the testing process is already with an early version of the software to check if the whole system works and adapts to the user's needs.
- **Acceptance Testing**. This is the final part of the testing levels, which is being tested to be accepted with the purpose of evaluating the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Taking into account the different processes of testing the software and the timelines, the final decision of testing will be during the sprint 7 of the project between the days 26th and 28th of May, where there will be a system testing to get as much feedback as possible. Once having information from users, the process of polishment will start on the basis of the data taken previously.

[74] Software Testing Fundamentals - "Software Quality Control" - August 31, 2020

5. Methodology

The methodology of this project will be pretty similar to some pipelines already seen in the videogame industry, but the most common are typically divided into three stages[75]:

- Pre-production.
- Production.
- Post-production.

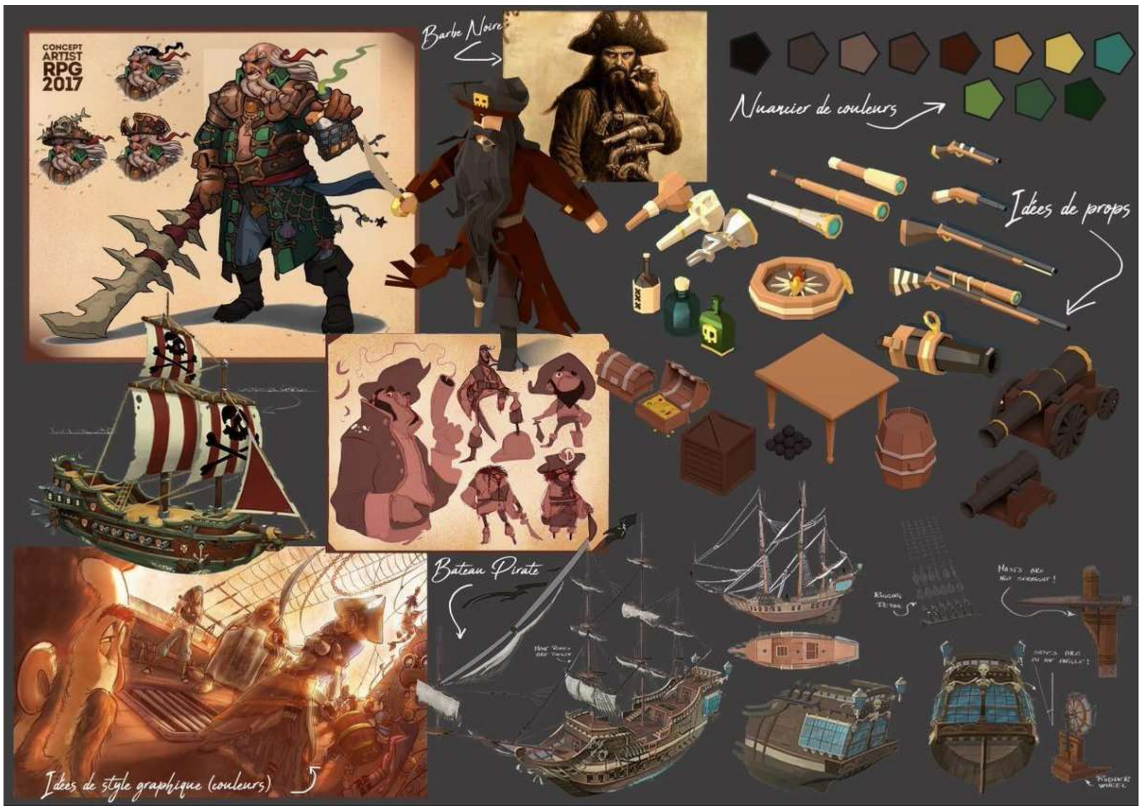
The reason for the choice of this method for the project is due to it is the most common in the videogame industry, as well as it allows a more manageable timeline and budget to reduce inefficiencies and bottlenecks, combining it with the planning from the last chapter, what remains is how the project will be developed.

5.1 Pre-production

This is the first step of the work pìpeline, which is the planning phase where the main focus is the concept development and idea that will be carried out during the rest of the process.

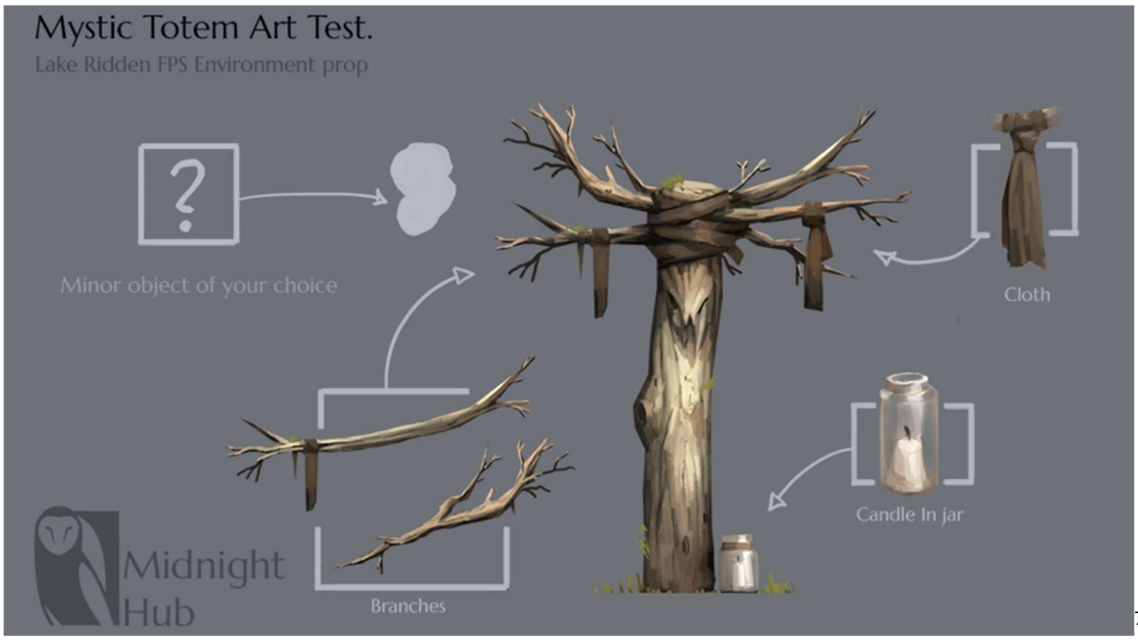
In the art part the most important phase is the concept art, which consists of the exploration of a central idea. In the case of this project, as the main idea is to create an environment, the models that can fit the most will be houses of a village. Once the idea is established, it is time to set the theme of the environment, which will take a certain atmosphere. The first reference that will be used is the creation of a moodboard, which is a collection of visual materials that evoke a certain style or concept, so the art style will also be taken from this point[76].

[75] Nadia Stefyn - CGSpectrum - "How videogames are made: the game development process" - September 5, 2022
[76] Marc Clancy - "How to make a moodboard in 10 easy steps" - March 10, 2023



[77] Figure 42: Showcase of a moodboard

When creating the concept arts needed for the final version of the models, there will be different sketches from different perspectives of the prop to take as much detail as possible.



[78]

Figure 1: Concept art of an environment prop

[77] Image taken from the user salemium - Twitter - January 10, 2019
[78] Image taken from "A kickass Game Art Portfolio" - Midnight Hub - erik - June 2, 2016

The first step is to define the tools that will be used during the development of the project, which function will have and in which phase will be needed for.

In this first stage of the project, it is essential to take into account how the tool will be developed or which art style the models will have, so it is necessary to take references and organize them. For the artistic part of this thesis the tools used for making mood boards and get inspiration there are three tools:

- ArtStation. This is a website which allows the user to showcase visual content as a portfolio, being focused on art. Here there is a wide variety of images, videos, short clips and scenes. There is the possibility for the user to create blogs, which is to share the work in progress, articles and highlights among others. Those blogs will be shown in the user's own website, which is self-hosted and can be edited with the themes it gives. This tool is very used to get references from other people with the possibility to contact them, with thousands of concepts[79].

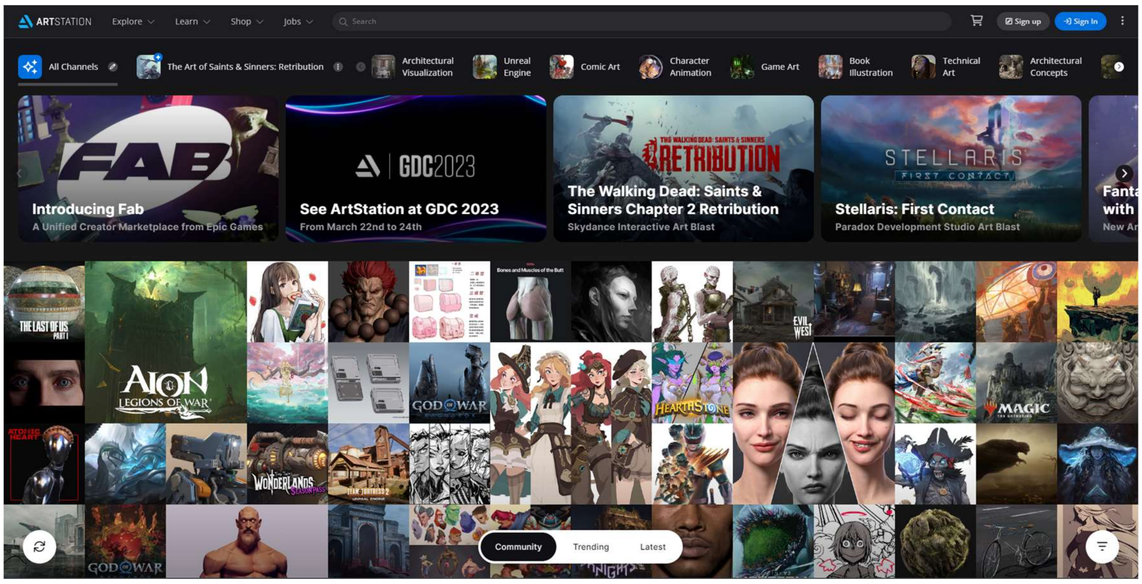

ARTSTATION
Explore
Learn
Shop
Jobs
Search
Sign up
Sign In
All Channels
The Art of Saints & Sinners: Retribution
Architectural Visualization
Unreal Engine
Comic Art
Character Animation
Game Art
Book Illustration
Technical Art
Architectural Concepts
Introducing Fab
A Unified Creator Marketplace from Epic Games
See ArtStation at GDC 2023
From March 22nd to 24th
The Walking Dead: Saints & Sinners Chapter 2 Retribution
Skydance Interactive Art Blast
Stellaris: First Contact
Paradox Development Studio Art Blast
Community
Trending
Latest

Figure 35: Capture of ArtStation website

- Pinterest. It is a visual content discovery engine for finding ideas. In this situation, an user can upload an image that will be named pin, which once published is public and free access to the rest of the users. When the user makes a search in Pinterest, the algorithm will recommend similar images to constantly provide information[80].

[79] ArtStation - "Overview of ArtStation"
[80] Pinterest - "All about Pinterest"

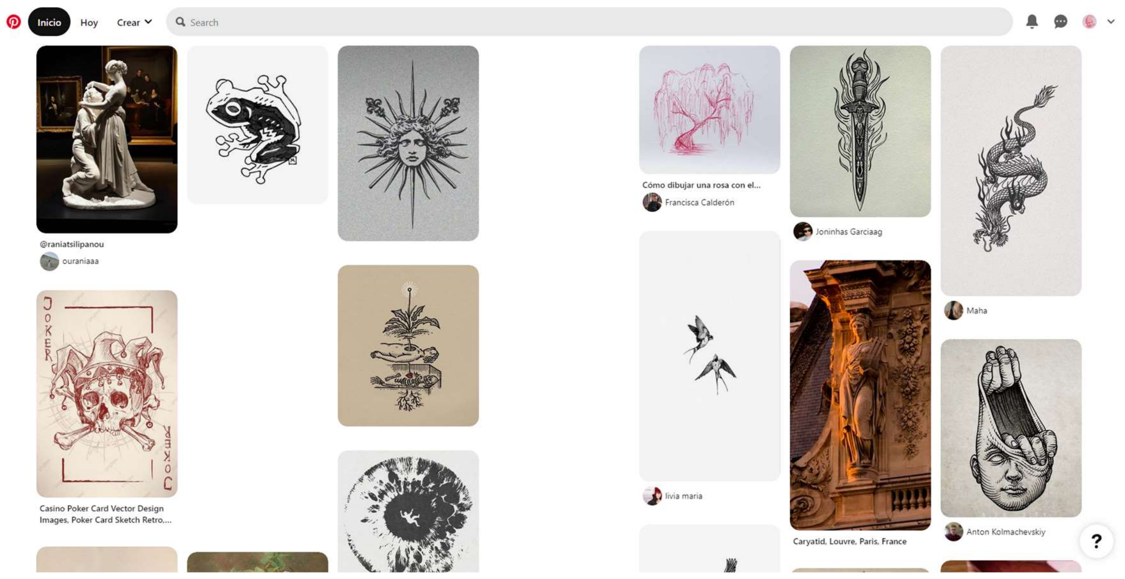

Inicio
Hoy
Crear
Search
@raniatsilipanou
ouraniaaa
Casino Poker Card Vector Design Images, Poker Card Sketch Retro,...
Cómo dibujar una rosa con el...
Francisca Calderón
Joninhas Garciaag
Maha
livia maria
Caryatid, Louvre, Paris, France
Anton Kolmachevskiy

Figure 43: Capture of the website of Pinterest

- Pure Ref. This is an application that forms a black canvas, which is resizable and intuitive to add images and notes, allowing the creation of schemas and moodboards, specially created for making moodboard[81].

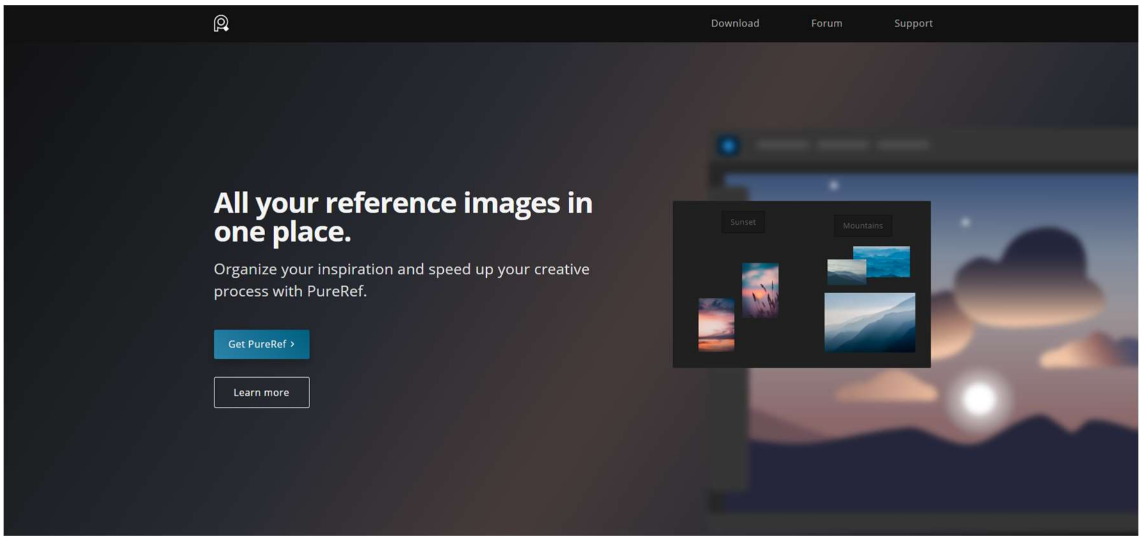

Download
Forum
Support
All your reference images in one place.
Organize your inspiration and speed up your creative process with PureRef.
Get PureRef
Learn more
Sunset
Mountains

Figure 44: Capture of the website of PureRef

5.2 Production

5.2.1 Modeling Production

Once the concepts are defined, the next step is the own creation of the models that will conform the environment, so the pipeline determined for this part of the project is divided into five steps:

[81] PureRef - "About | PureRef"

- Concept and moodboard.
- Low poly version.
- UV mapping.
- High poly version.
- Texturing.

The decision of this pipeline is due to it being pretty standard for the game industry, so it can adapt to others, adding some versatility to the production. Although the creation of a block out prop was mentioned in another chapter, this methodology has been removed. The reason for this decision is because the block outs are usually created firstly for sculpting later, so making the low poly version before saves a stage of the process and goes directly to the retopology of the prop. Another feature to take into account is the UV mapping right after the retopology, and this is preferable due to if the model is already optimized, it is easier to test textures and materials before making the bake with the highpoly.

The tools that will be used for this stage of the project are the following:

- Blender. This is a free and open source 3D computer graphics software that is used for modeling, animation and visual effects, among others. It is also used for creating unique and procedural materials. It will be mainly used for modeling and UV mapping, but also the program allows sculpting, but as it is not such developed, so it will be useful to add some details[82].
- Zbrush. This tool is also a 3D software which allows the creation of 3D models, but in this situation, it is mainly used for sculpting, so it is ideal for the creation of the high poly version. This powerful software is one of the most used in the art of videogames with multiple options and plug-ins to modify the meshes[83].
- Adobe Substance 3D Painter. It is a paint engine, recently bought by Adobe, which allows the user to paint the UVs of the 3D models previously created and the possibility to create textures from scratch with thousands of combinations and options. This is the standard of texturing models on different industries which need some creativity[84].

5.2.2 Development of the tool

For the second part of this project, the development of the tool will be in a 3D commercial engine, but firstly it will be set up in another software. Once there is a working prototype in the first program, it will be passed to the engine. The tools that will be used for this stage are the two following:

- Houdini. This is another 3D computer graphics program that works with nodes, but it is mainly focused on procedural workflow. The nodes of this program are

[82] Blender - "About window from Blende website"
[83] Pixologic - "Zbrush"
[84] Adobe - "Adobe Substance 3D Painter"

wired into networks which define a recipe that can be tweaked to refine the outcome creating unique results. This tool is widely used in the industry mainly used for FX and creation of huge worlds, being very intuitive and easy to learn[85].

- Unreal Engine 5. This is the 3D engine that will be used for the project and where most of the work will be done. This engine also works with nodes, but in this case they are for programming, which are named "Blueprints", and are pretty susceptible to the creation of procedural generation[86].

The reason for the choice of this software is due to the fact that both are pretty used in the creation of procedural content and they are well complemented to each other with the exportation and importation projects. In the case of Unreal Engine, it owns the Unreal Marketplace and it is an easier process to publish the final version.

5.3 Post-production

Finally, for the last steps of the project, the software will be pretty much the same as mentioned in the last two points.

In this stage of the project the main task will be optimizing the blocks of code for the tool as well as multiple testing sessions. For the visual part, it will be based on the creation of high quality textures and configuring some environment details like illumination.

[85] SideFX - "Houdini | Home Page"
[86] Epic Games - "Unreal Engine 5"

6. Development of the project

6.1 Pre-production

6.1.1 First concepts for the models

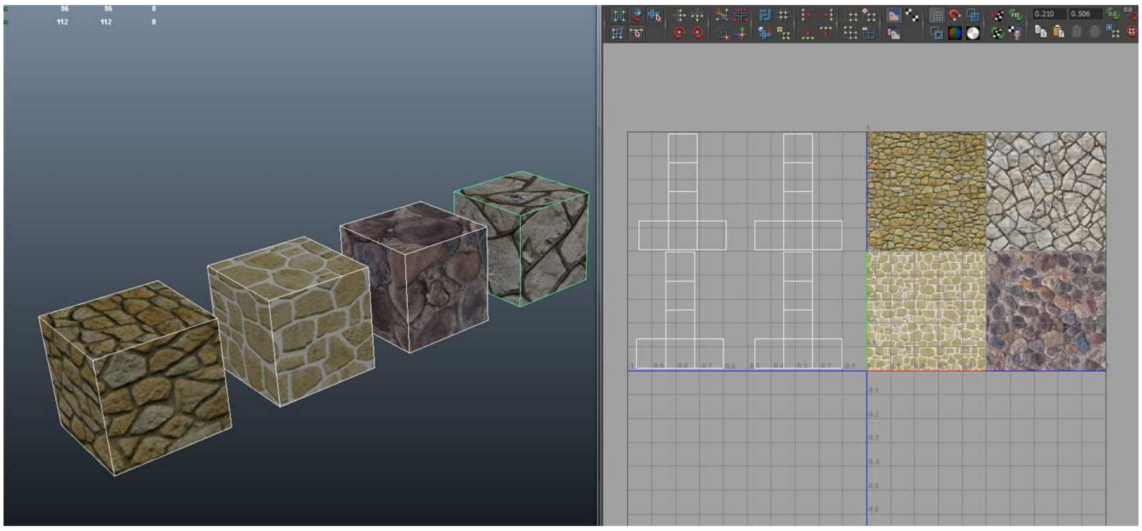
This chapter will consist of the process of the project, the software that will be used and how it is going forward taking into account the original planning. During this first month of development, most of the content established in the sprints is prototyping and trying to apply different ideas.

As mentioned in the previous section, inside the software that will be used there is Houdini, due to it allows many options in terms of creation of procedural content and offers a lot of versatility. However, the program has limitations in some sections, like in the process of making a 3D environment, due to the program taking other directions.

For this reason, the development of this project will be totally in Unreal Engine for the part of implementing the procedural creation and the setting of the project, but first of all, what is needed to create an environment is to create props.

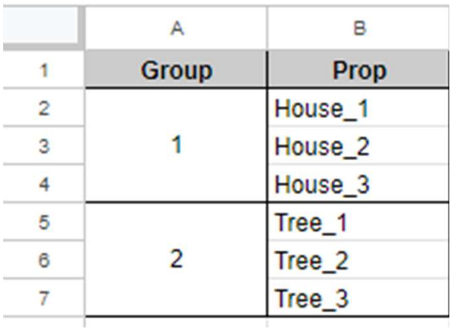
Although the user will be able to create environments with their own models, the tool will come with some made models that will work as an example of what can be done and to test the tool. During the third sprint, it was necessary to create a list to stipulate which essential props will conform to the environment.

Due to there will be many models created, each one will have its own UV mapping and materials, but there is a method that saves a lot of management and allows to optimize and organize the assets production. This is also known as texture atlas, which consists of grouping more than one uv map into one file. Sharing the same material allows to use less textures to get the same results[87].



[87] Joe Bernardi - Medium - "Texture Atlasing: An Inside Look At Optimizing 3D World" - January 24, 2018

[88]Figure 45: Image showing how a texture atlas works



	A	B
1	Group	Prop
2	1	House_1
3		House_2
4		House_3
5	2	Tree_1
6		Tree_2
7		Tree_3

Figure 46: Capture of the first version of the prop list

Once there are some principal groups established, the first iteration of the models will be the creation of Blockings, using basic shapes and without textures, so they will be used for testing the tool. In this step, the program that will be used for modeling is Blender, due to it giving great results in modeling as well as uv mapping and organization of the texture atlas. The first basic shapes that will be used for testing are a house and a tree.

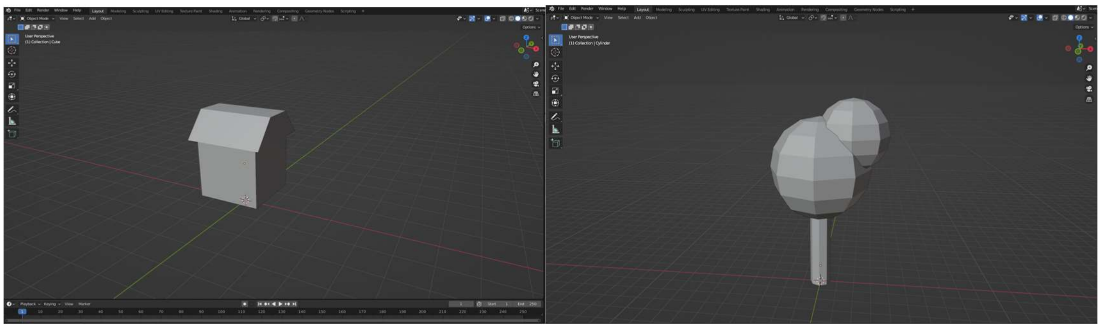


Figure 47: Captures of the first basic shapes

6.1.2 Set up of the project

Having created some models, the next step is to open the Unreal Engine and create a new project. Once the project is set up, it is necessary to create an empty scene where it will be possible to work with the tool. In this engine, to create a scene is as simple as right clicking in the window that shows the files of the project and selecting "Level".

[88] Image taken from Unity Answers - "The correct way of making Texture Atlas"

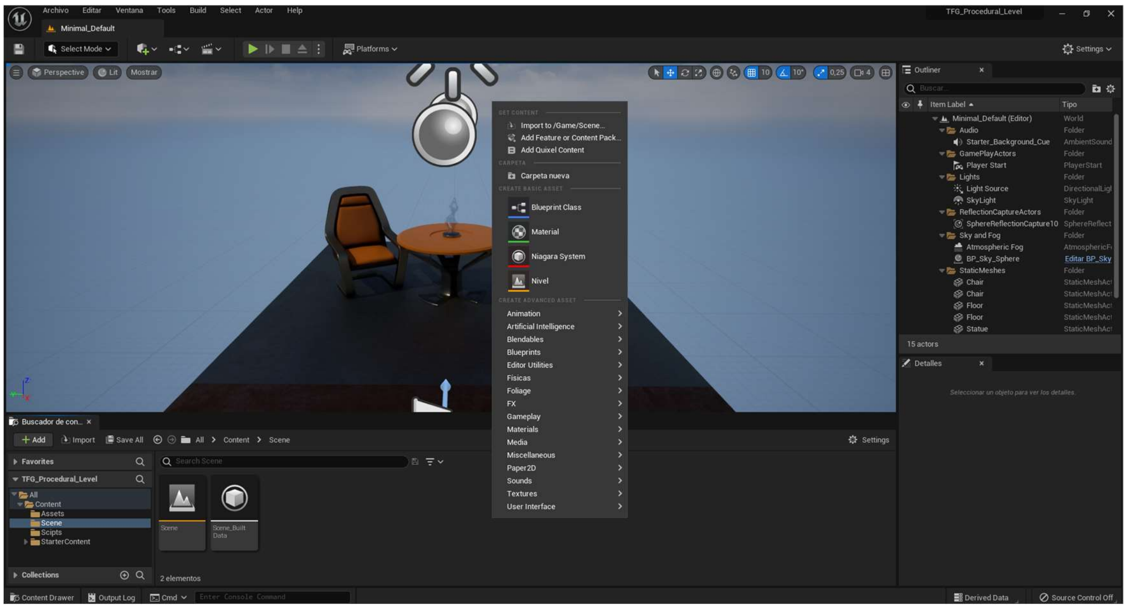


Figure 48: Capture taken from Unreal to create a new scene

To create an empty scene, the most common is to use the "Basic" option, so it consists of a plane and some basic lighting with the skybox as background.

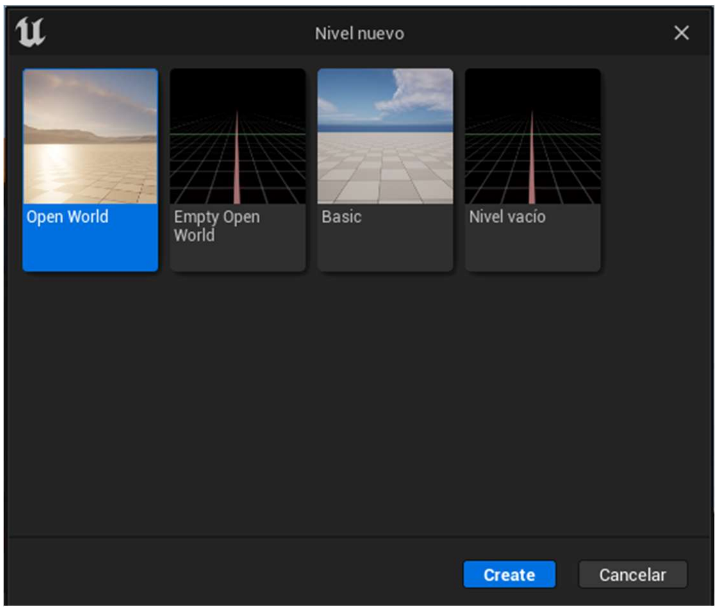


Figure 49: Capture of the scene selector

The way Unreal works with the meshes that are imported from other software is naming them "Static Meshes"[89], which are geometry pieces that consist of a static set of polygons that conform to the basic unit used to create geometry for levels. From the static meshes, they can be used to create objects that move, rigid body physics objects, foliage and terrain decoration. Those terms will also be used in this project.

[89] Unreal Engine Documentation - "Static Mesh Component" -

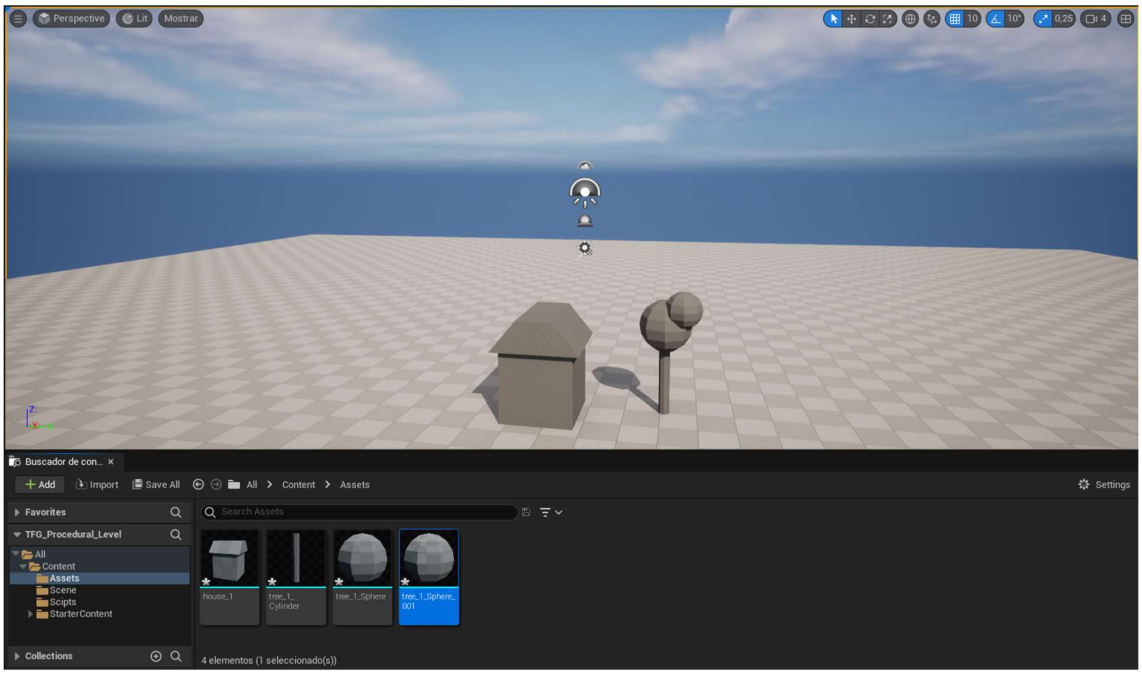


Figure 50: Capture of the models imported to the engine

For this project and for making a better showcase of its capabilities, the scene chosen has been an empty landscape that Unreal already has as a present. This is a single plane that can be sculpted and involves using a variety of tools that modify the underlying heightmap. These tools range from the simple sculpt tool that paints height values using a brush and strength scale to many complex algorithms such as erosion[90].

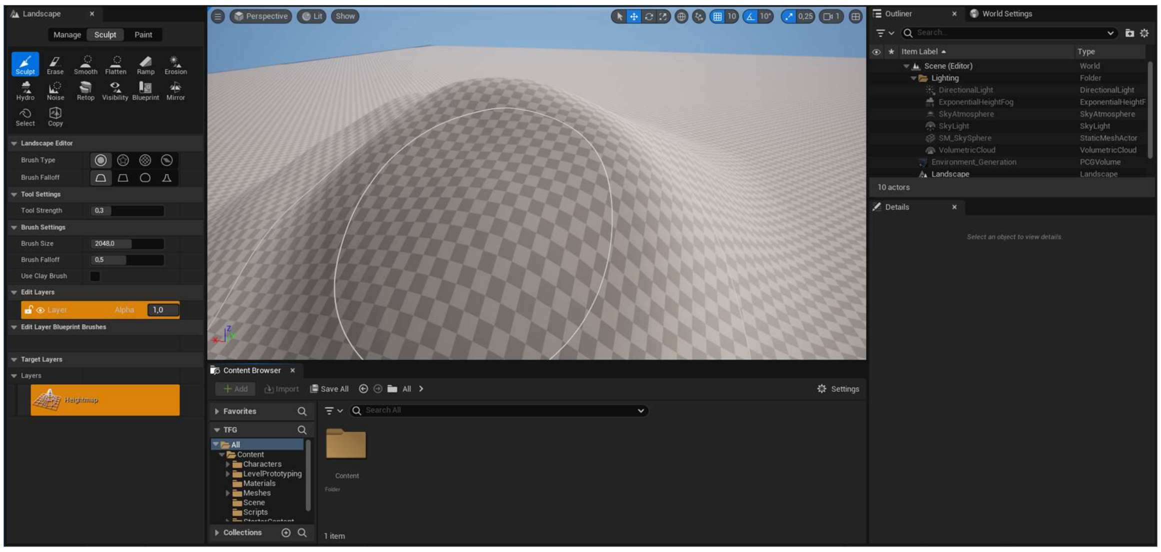


Figure 51: Capture of the landscape function of Unreal.

One of the biggest advantages of using unreal instead of other engines is the feature of procedural generation framework, which consists of a toolset that allows the user to make procedural content, enabling to define rules and parameters to populate large

[90] Unreal Engine - "Sculpt Mode" - Unreal 5.0 Documentation - 2022

scenarios. This component also works in runtime, which means that the system can run in real time, so the world can react to gameplay or geometry changes.

At this time of the project the tool can only spawn the model of the house taking into account the population, which appears in the details window of the engine, but in future sprints it is thought to be able to spawn more models at the same time. During this second month of development, it will be focused on the functionality of the tool, so the final version of the models as well as the User Interface will be part of the last steps.

6.2 Production

6.2.1 Creation of the environment elements

A big part of this project is based on the modeling and production of 3D assets, so it will be taking into account the processes mentioned in the last chapter to follow the pipeline established.

The first steps before starting the production of models is about the concept and the art style that will be used to have a coherence among the models. Although the project is thought to be used with any kind of environment, in this situation the scenario created will be a medieval city, having an art style that combines realism with a bit of stylized materials. This decision has been taken after comparing different environments of games already released, and taking into account the limitations of time, going to a hyper realistic style is not optimal due to the time required to the materials, as well as avoiding organic modeling, like trees or plants, which would add more complexity in the asset creation process.

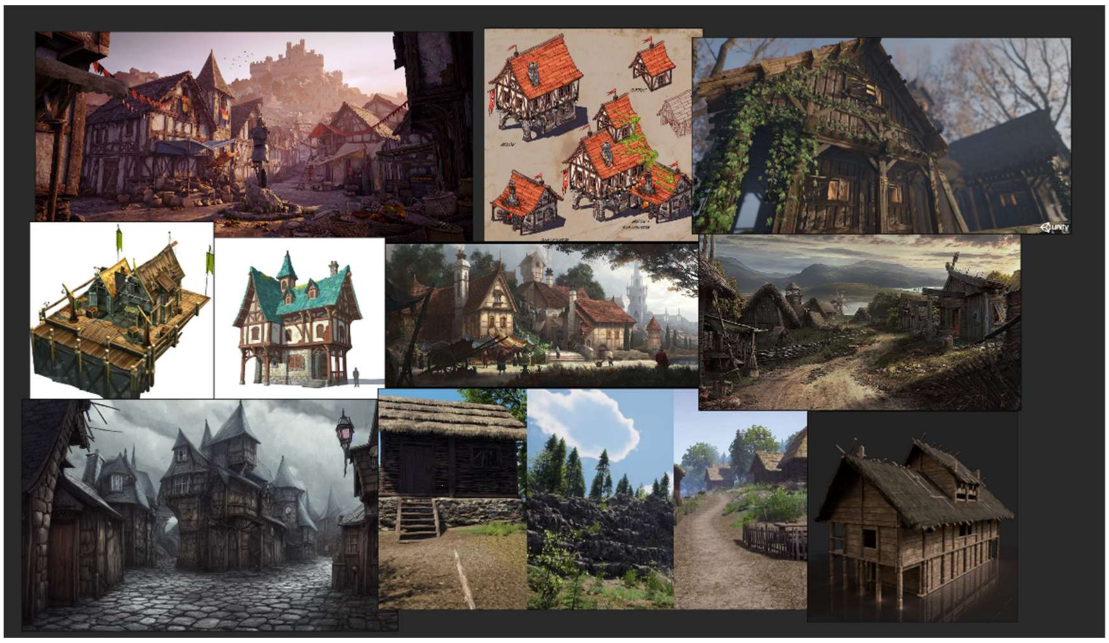


Figure 52: Image of the moodboard for the assets

As it can be seen above, this moodboard is to take references for the 3D models that will conform the environment and the materials that will be used for texturing. The main approach of this scenario is to simulate a medieval village that is located in the mountains, so most of the colors will be with darker wooden tones except for the tavern, which will contain few warm colors in the roof. The decision of this is due to in a videogame city, most of the buildings have the function of decoration and help to a better immersion, being the buildings that are intractable the ones with brighter colors, as well as using warmer colors help the player to think that this place is safe.

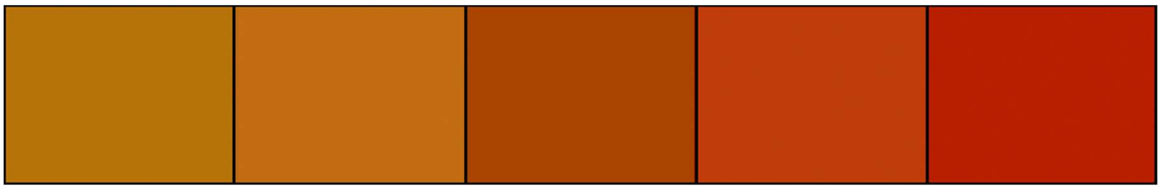
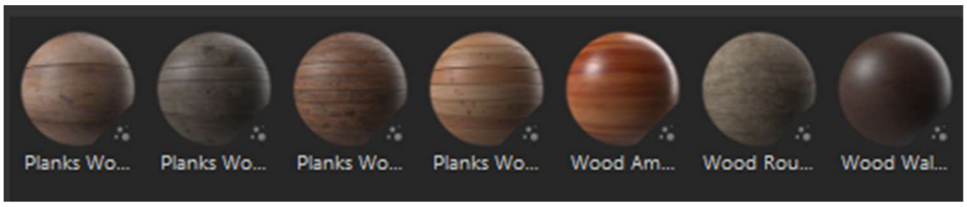
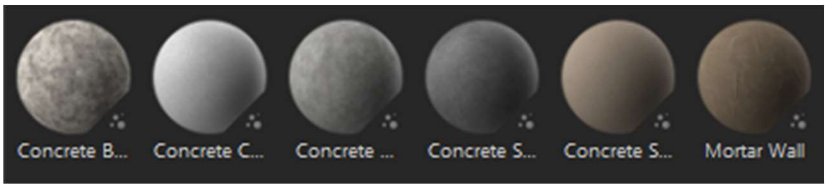


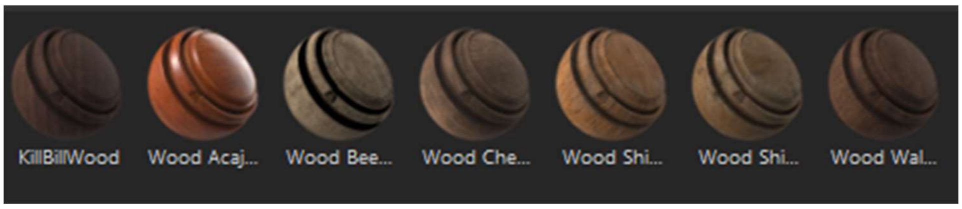
Figure 53: Showcase of the color palette for the environment

In the technical part, the decision for the capacity of the models oscillates between 4.000 and 1.500 polygons per prop, allowing to have certain detail at the time of modeling. As the engine chosen is Unreal Engine, this software has much more capabilities, allowing it to run up to 10 billion polygons, so it is easier to model with more detailed models.

The usage of textures and materials is also a crucial part of the process, because they give realism and detail to 3D models, defining the physic and visual properties to the object. In this project the main materials used are mainly based in wooden planks, which conform the majority of the houses, but to give variety to the props and not to look all the same, there are other materials, like the concrete that are used which also have the coherence of the moodboard and the concepts.







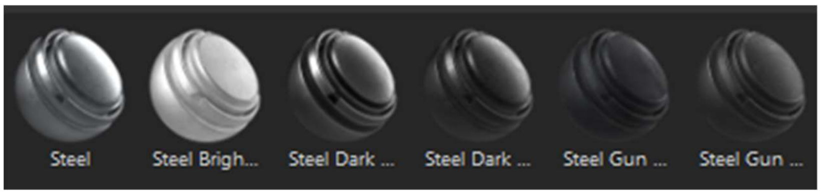


Figure 54: Showcase of the main materials used

As it can be seen in the images above, the wooden materials correspond to the color palette, allowing to play with the different tonalities available, letting the concrete as a complementary material and the steel to give unique details to the models.

Another feature of texturing models, there is the size of the texture, which consists in the quality that the image will have, and ending in how deep the detail is displayed on the model. Taking into account the number of models and how much resources it will consume, the final decision to the size of the texture will be 1024 x 1024 pixels with 8 bits depth, following the PBR pipeline, which stands for Physics-Based-Rendering material to simulate any kind of physical material to a better simulation of the 3D model.

For the individual models, each one counts with a concept that approaches more to the main idea, allowing to explore the idea deeper, as well as giving each model its unique identity.

This is the first prop made for this project, which consists of a dark wooden house with a plain roof and different brown tonalities, following the main palette. To arrive at the final result, the model has been the mix of two concepts taken from the Internet that accomplish the artistic view of the project.

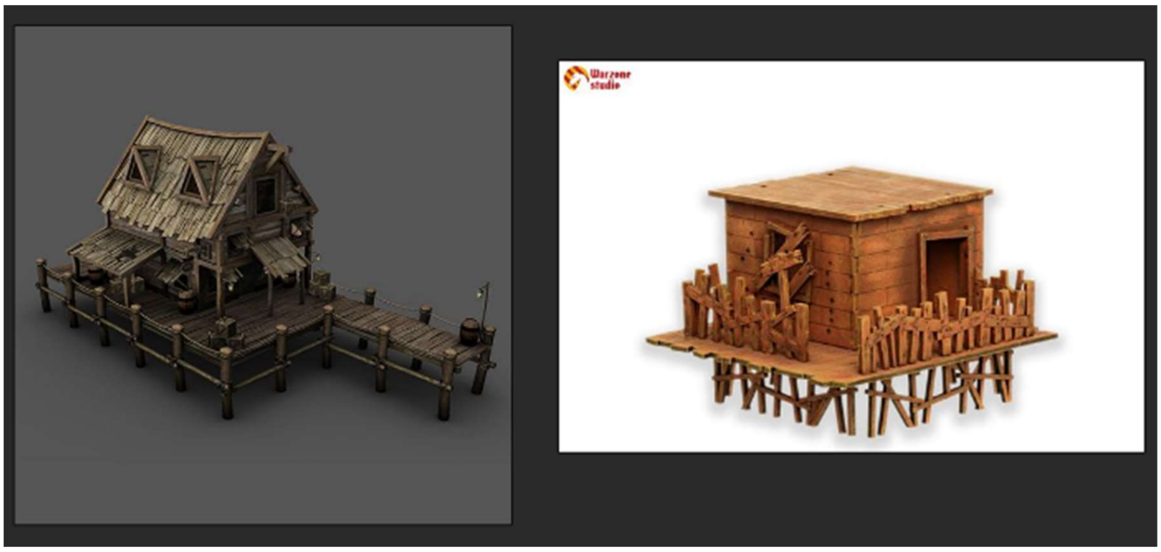


Figure 55: Concepts of the first house

To complete the next steps, it is necessary to open Blender, due to this program allows 3D modeling and UV mapping. The approach of this house is to have rectangular shapes, due to its material is based mainly on wooden planks.

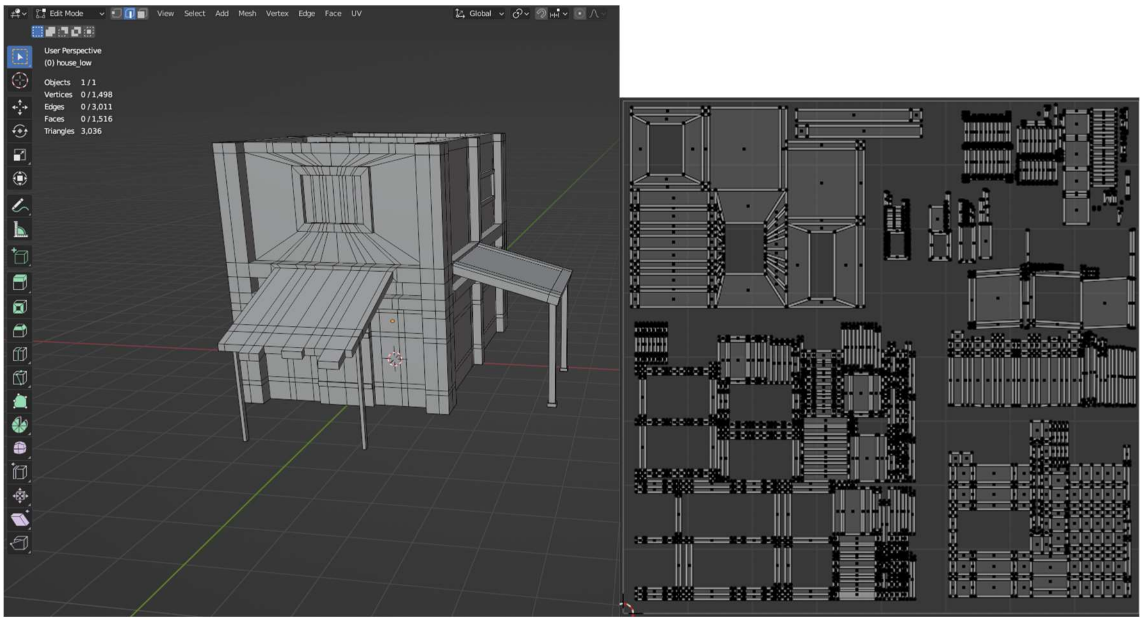


Figure 56: Wireframe and UV mapping of the model

The baking of this model consists in the process of saving information related to a 3D mesh into a 2D texture file, being this information transferred thanks to the UV mapping. As it can be seen in the image below, this first house corresponds to the use of a main clearer tone with darker details to make a bigger contrast.

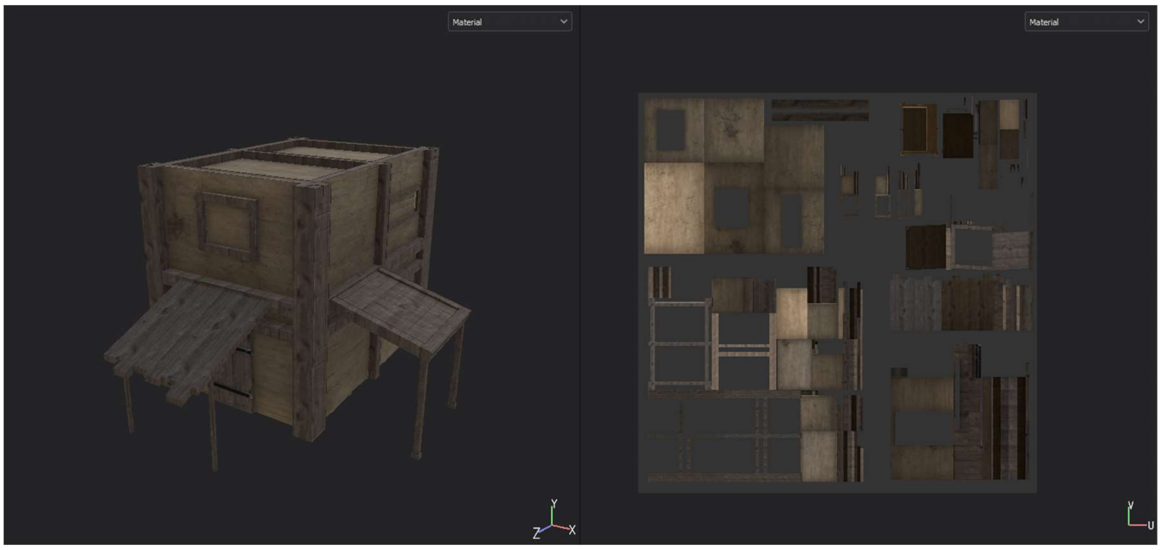


Figure 57: Showcase of the baking and texturing of the model

To import the model to Unreal, there are few changes needed, due to the software of Blender and Unreal being quite different, so for exporting from Blender it is needed to rotate the pivot so the engine works well. This is the final result displayed in Unreal:

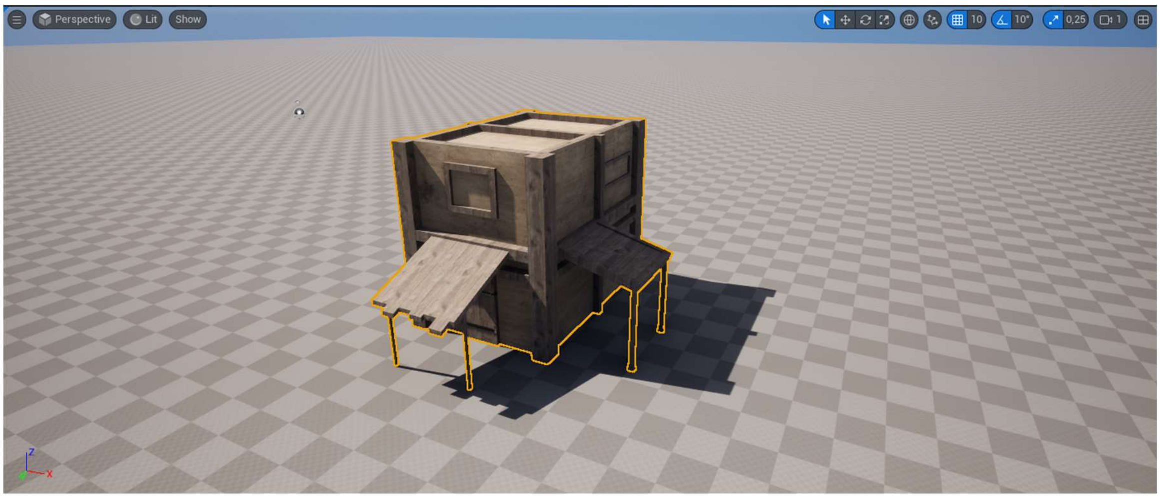


Figure 58: Display of the model in the engine

The second model consists of another house, but with a different touch, having another orientation and mainly changing a bit the color palette by using a concrete material, but respecting the main concept.

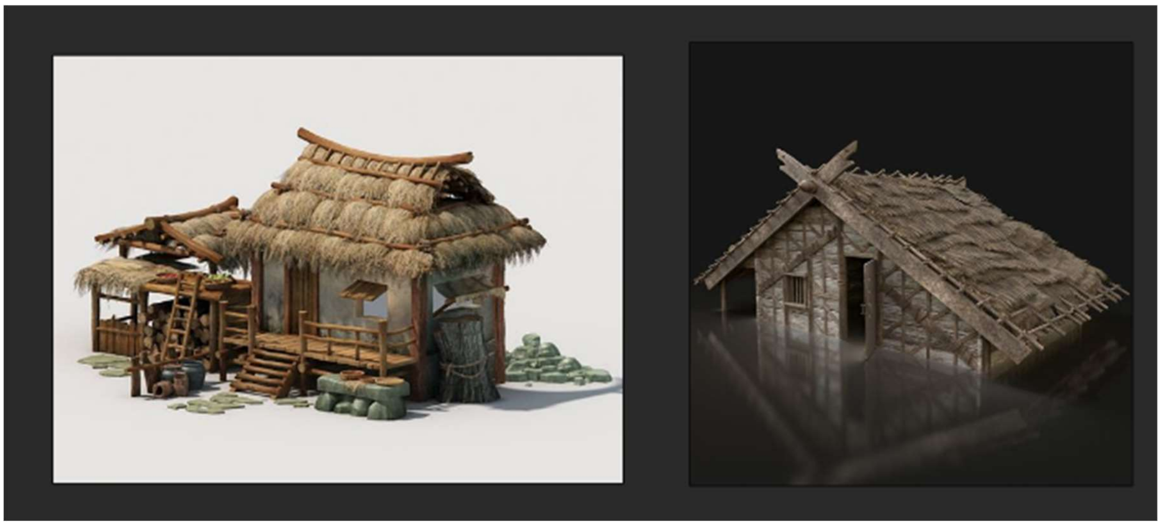


Figure 59: Concept of the second model

The objective of this model is to make a more rounded roof, but not too much because it can affect the texture of the wood and would look too cartoonish. This model contains way less polygons than the first house, being a bit more simple.

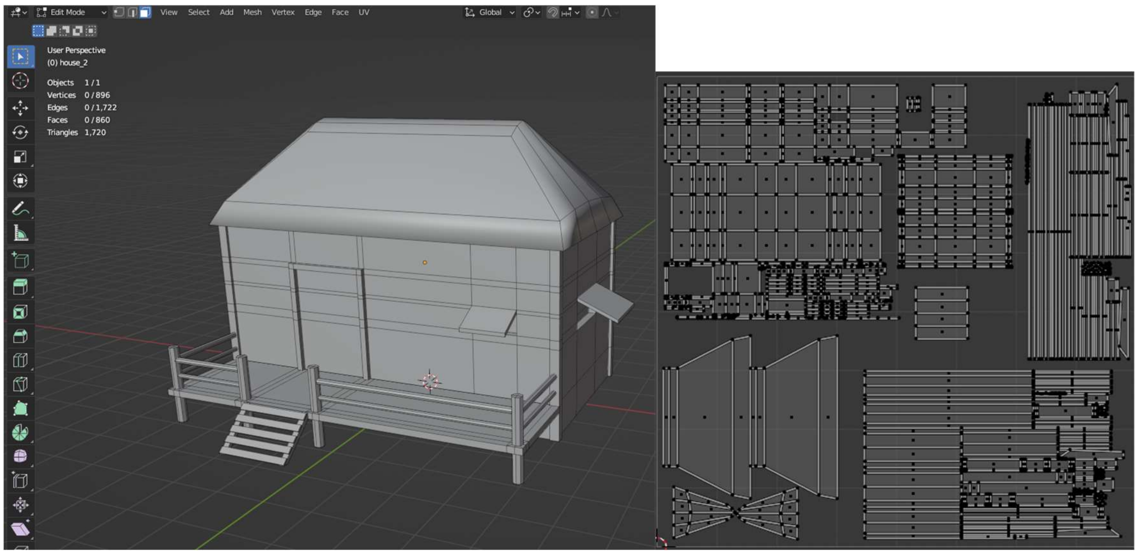


Figure 60: Modeling and UV mapping of the second model

For the texturing and material of this prop is mainly based in the combination of the same colors of the main palette, but changing the order of them, taking more center stage the darker wood, forming part of the roof and floor, and a clearer tone for the walls to play with the contrast:

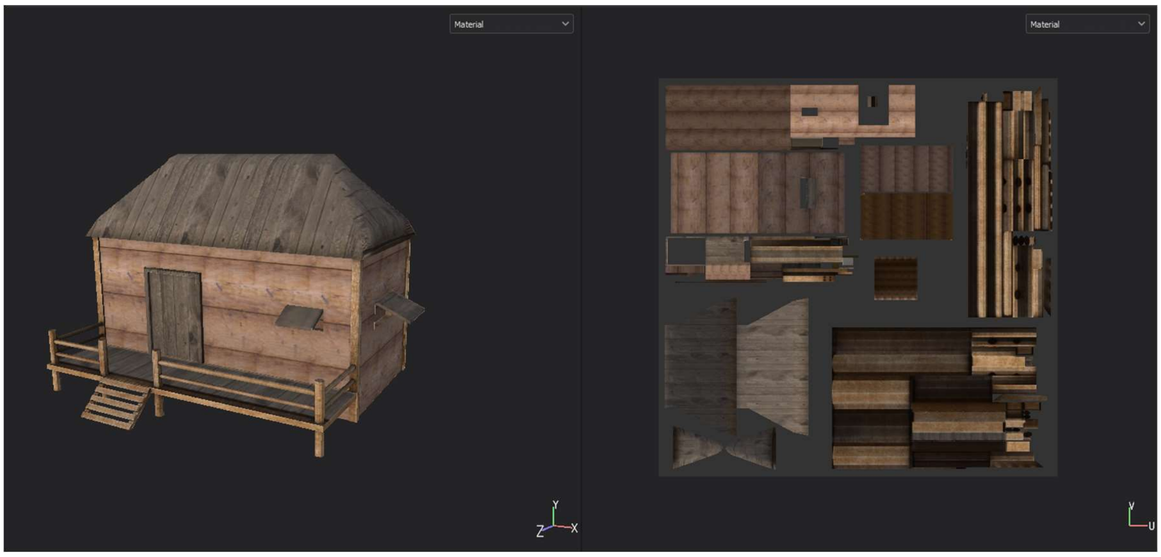


Figure 61: Baking of the second house

The importing process will be the same during all the props to maintain the same pipeline workflow:

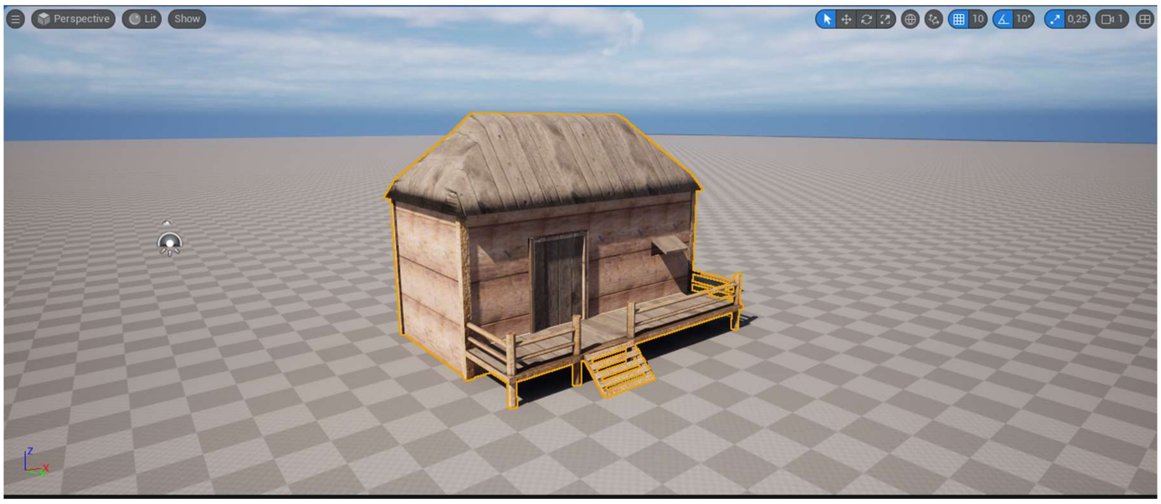


Figure 62: Showcase of the second house in the engine

For the third type of house, the idea is to follow the main color palette, but to make more contrast with the rest of the props by experimenting with the concrete material combined with some stones at the sides.

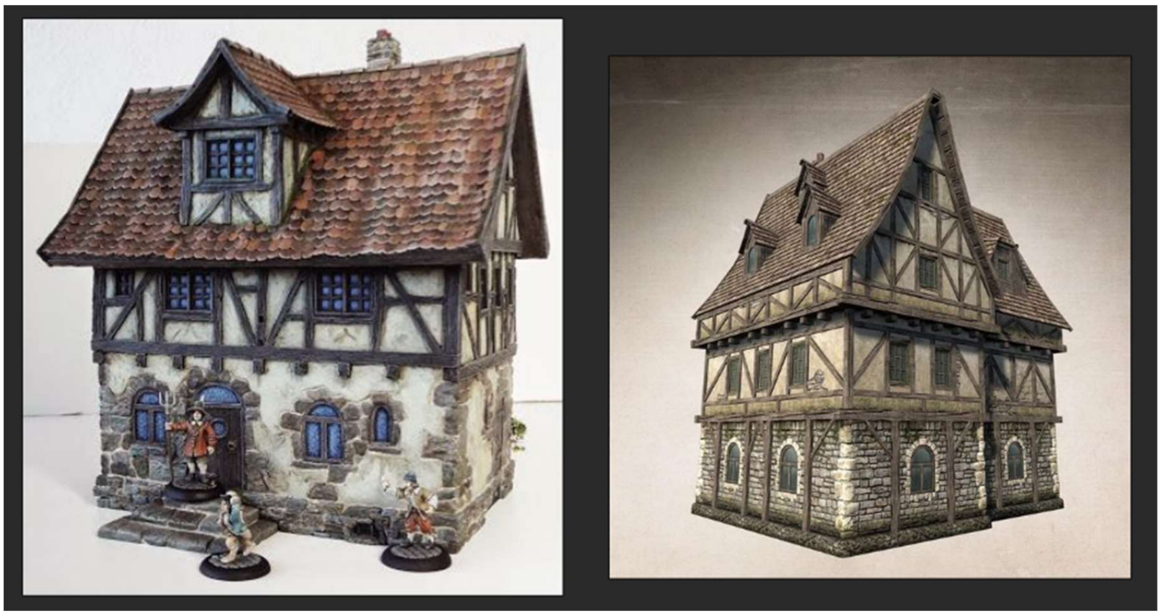


Figure 63: Concept of the third house

For the modeling of this prop, the biggest changes have been the flattened roof to follow the same style of the other houses of the village to keep the coherence. The purpose of this model is to make a house that could be distinguished from a larger distance, so the player could have a reference.

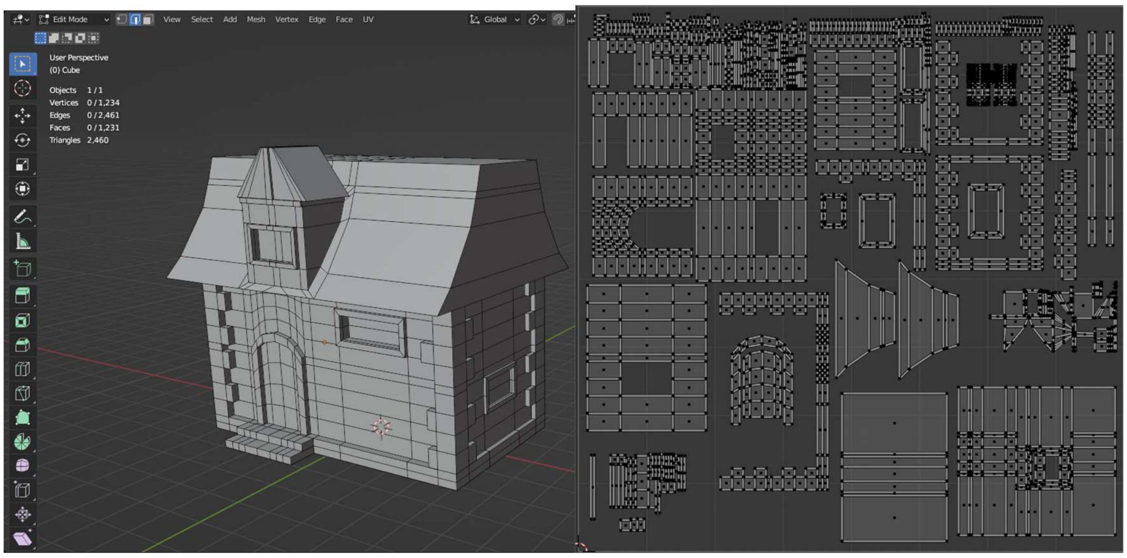


Figure 64: Modeling and UV mapping of the third house

For texturing and materials, the prop breaks the main palette a bit, due to it using only two colors from the main colors, but it uses the concrete material for the walls, giving an unique result.

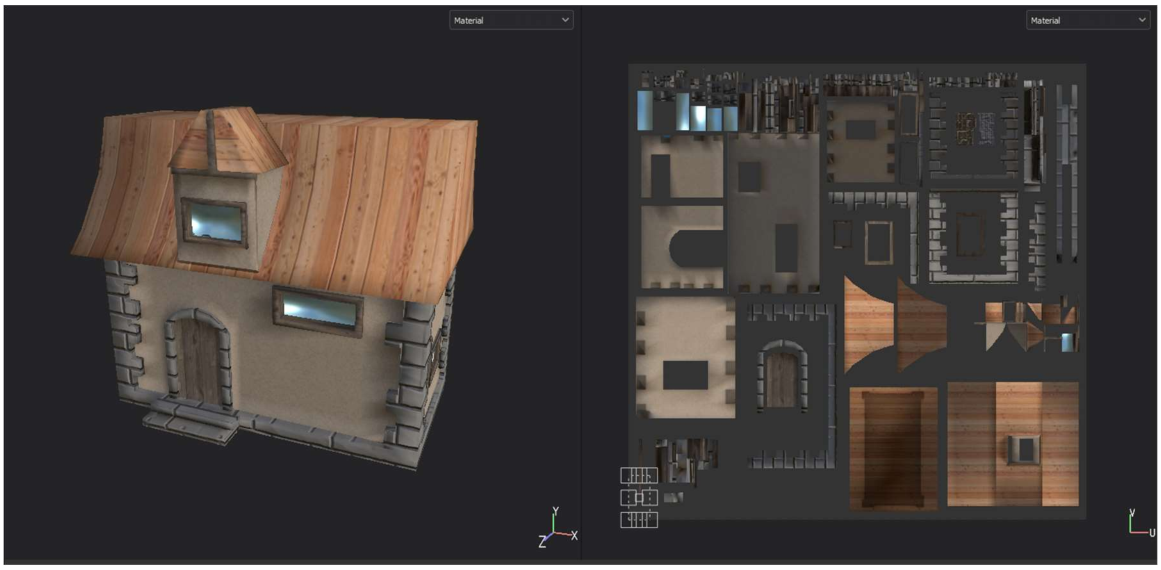


Figure 65: Texturing and baking of the third model

In this situation for the windows has been used a new material which consists in glass, which is based in removing the opacity and some roughness to give the feeling of reflecting the light.

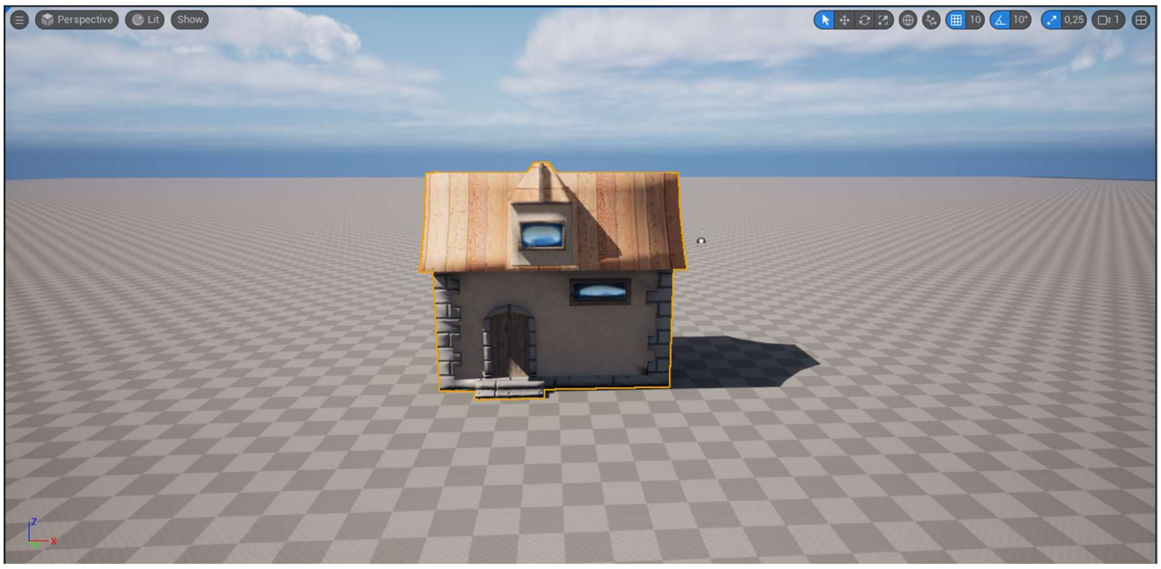


Figure 66: Display of the third house in the engine

For the final model, the choice was well, giving a bit of variation and richness for the environment, breaking the monotony of only displaying houses on screen.

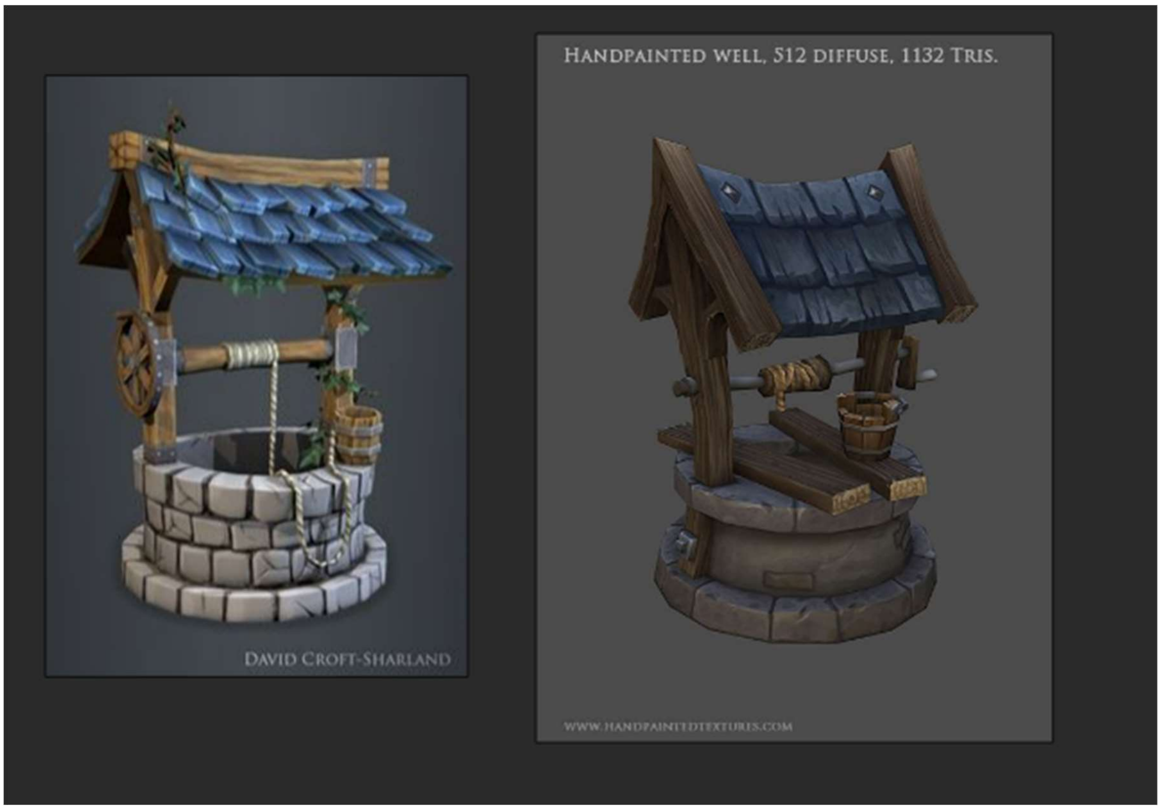
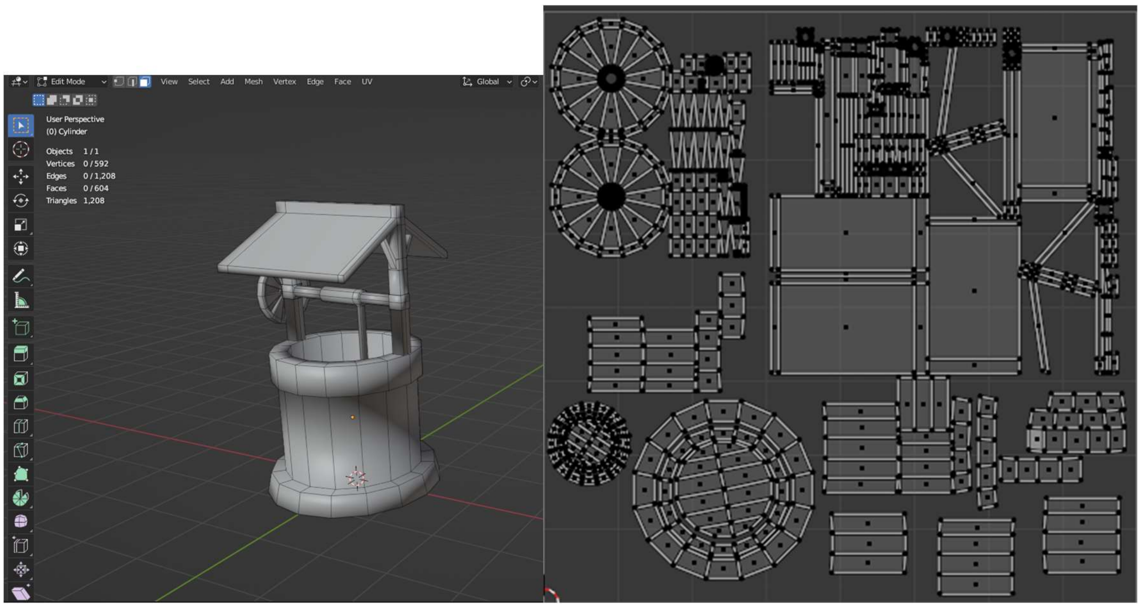

DAVID CROFT-SHARLAND
HANDPAINTED WELL, 512 DIFFUSE, 1132 TRIS.
WWW.HANDPAINTEDTEXTURES.COM

Figure 67: Concept of the well

Although the concepts are for a more stylized prop, the proportions of the final model has kept the “realistic” purpose of the art style maintaining the coherence.



User Perspective
(0) Cylinder

```
Objects    1 / 1
Vertices   0 / 592
Edges      0 / 1,208
Faces      0 / 604
Triangles  1,208
```

Figure 68: Polycount and UV mapping of the well

For the materials of the prop, the main colors have been replaced by stone bricks and blackboard material being the main colors, but adding wood and steel for little details so it also keeps coherence with the original color palette.

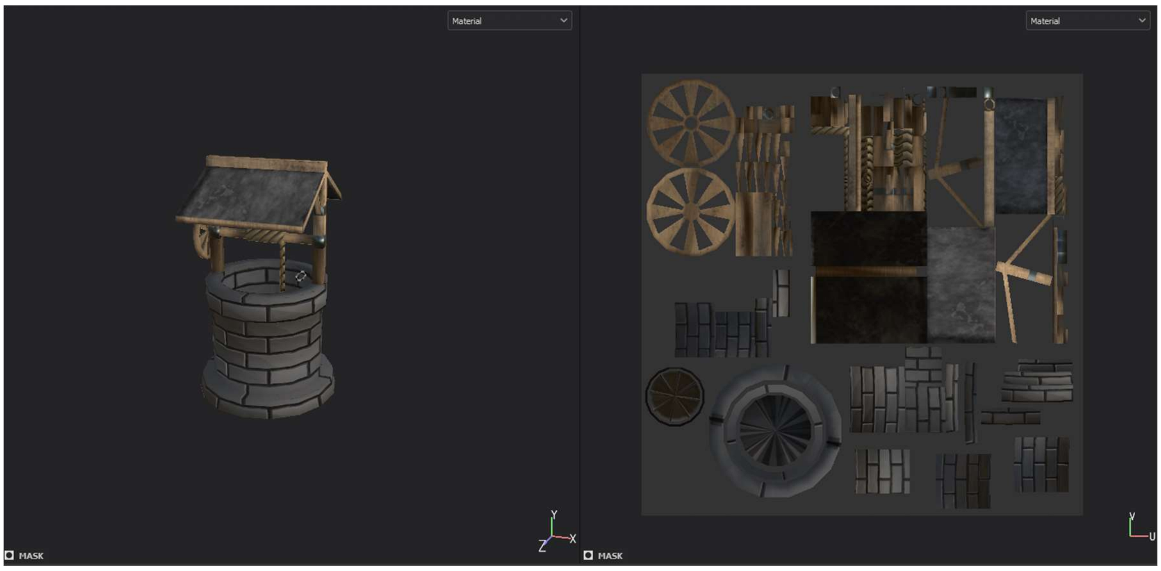


Figure 69: Texturing and baking of the well

The purpose of this prop on the environment is also decorative, but is more difficult to find in the scenario, due to being smaller than the other props and its spawn is quite limited to not saturate the player.

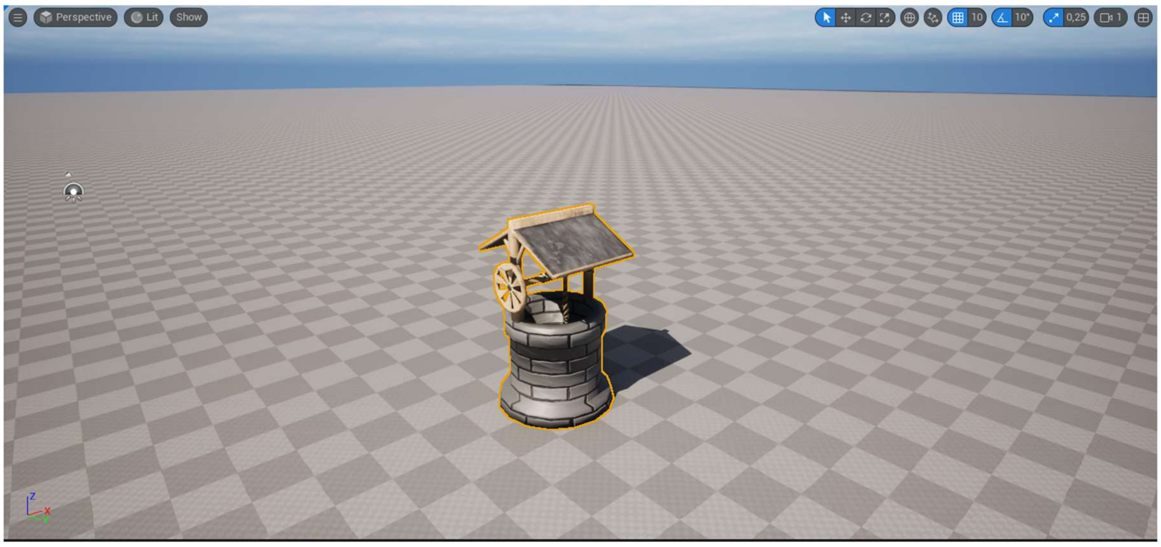


Figure 70: Display of the well in the engine

6.2.2 Creation of procedural graphics

One of the decisions for this project has been developing everything in Unreal Engine instead of firstly using Houdini because in the last updates of the engine there are functions that facilitate the creation of procedural content, being the procedural generation framework, which has a better approach for this project than other methods.

This new feature of the engine is based on a blueprint actor that creates an empty cube that contains an array of objects, which is a collection of similar data items stored at contiguous memory locations[91]. In this case the data stored inside this array will be the props previously created and imported in the engine, but there are few adjustments to make before.

The node editor that Unreal offers is based on a different way of coding, focusing on using blocks that contain code, but connecting them by using nodes, allowing a more intuitive experience in programming. Although this option is available, the engine also counts with the possibility to code entirely in C++ language.

What is needed to start spawning objects on screen, it is necessary to add a surface sampler in the procedural script, which is a function that allows the generation of points in a two dimensional domain that samples the cube previously created, which will conform to the surface of the environment.

[91] Geeks for geeks - “Array of Objects in C++”

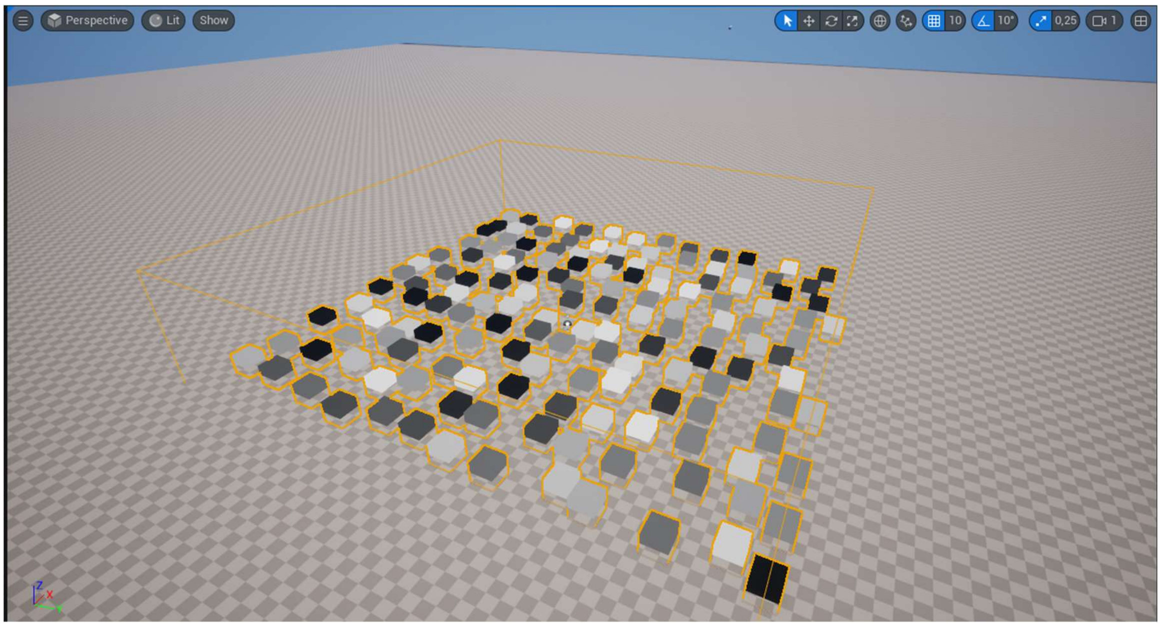


Figure 71: Capture of the first spawning of objects test

As it can be seen on the image above, all the cubes spawned correspond to different objects, but to implement a model it is necessary to make an additional step. The spawn of a 3D object works with the static mesh spawner, changing the default cubes with a model imported to the engine, like for example the model of the first house created.

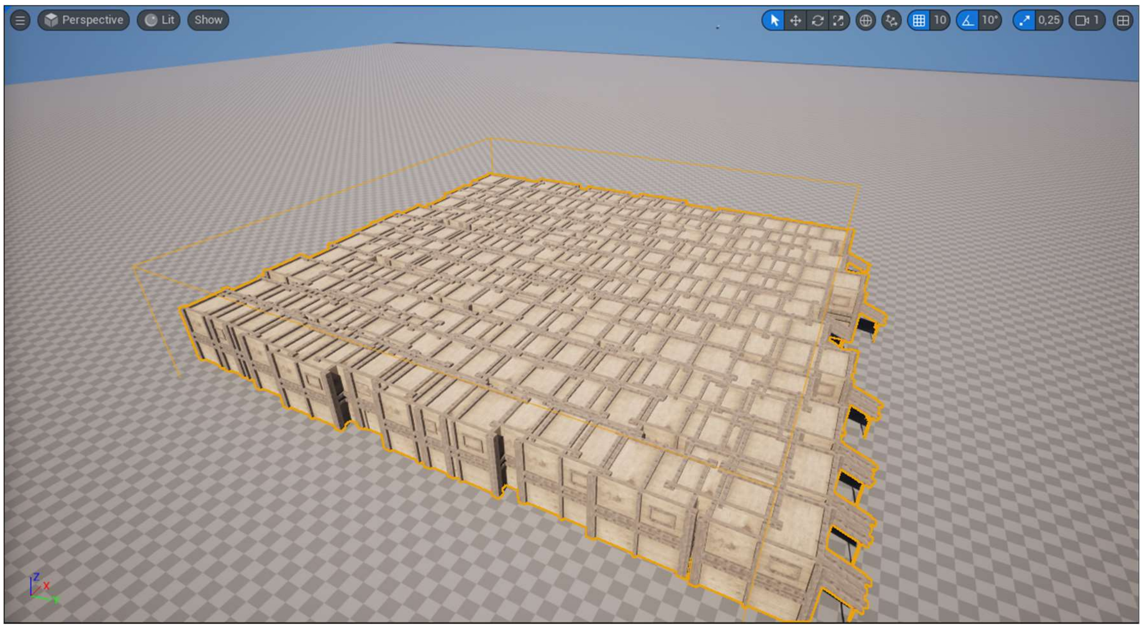


Figure 72: Display of the first model in the array

The problem in this situation is the point mesh that acts as the place to spawn the model, in this case the cube, because it does not take into account the bounding box of the model. Another problem to take into account is that all the models have the same local rotation, breaking the naturality of the procedural generation. What is necessary in this situation is to configure the points of spawn from the surface sampler as well as the looseness parameter, which is the parameter that determines how closely the generated points conform to a grid shape. This configuration is individual for each model, due to not all the props having the same size or the same bounding box, so those values must be changed manually.

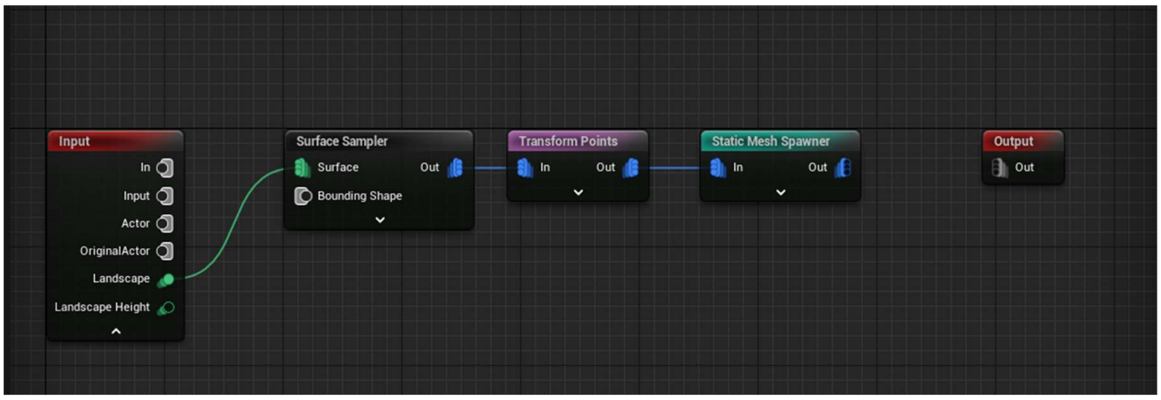


Figure 73: First simple script to spawn models properly

To solve also the problem of the rotation, between the nodes of surface sampler and static mesh spawner there will be necessary to create a transform points node, which allows the manipulation of different parameters of the model with ranges, so it would spawn the model with a random rotation inside the minimum and maximum values assigned.

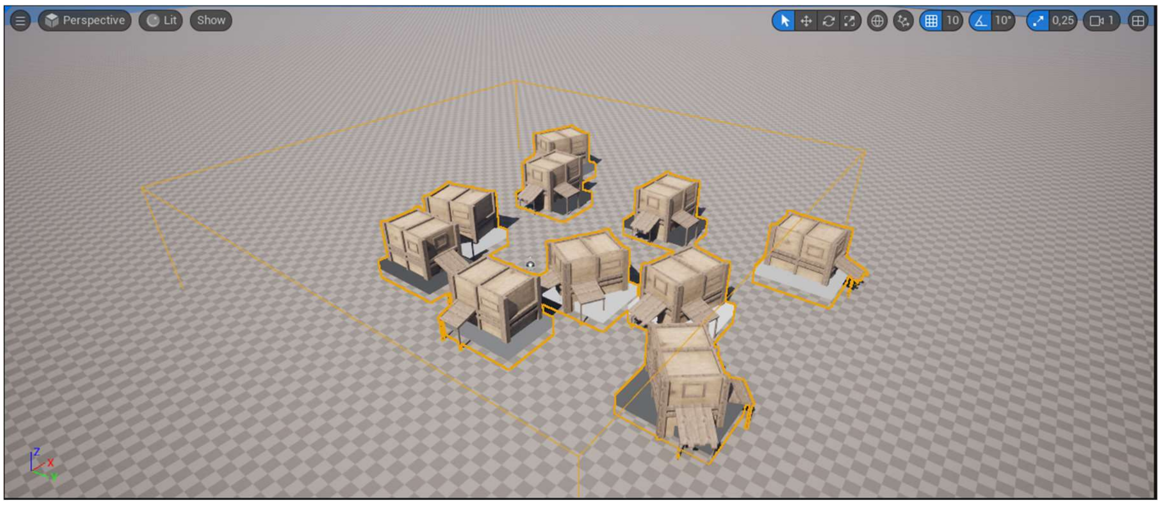


Figure 74: Spawning of models with some parameters changed

Once it is already configured of how the models should display on screen, it is time to combine the rest of the models, but firstly there are some changes necessary first, like watching the density of the props displayed, due to in bigger environments it is more complicated to change it manually. The node to use in this case is the normal to density and then the density filter, which consists of only spawning the cubes of a certain value inside the noise grid generated. Another important node is the self pruning, which consists of removing the meshes that are on the edges and also adjusting the ones that spawn too closely by checking their bounding boxes.

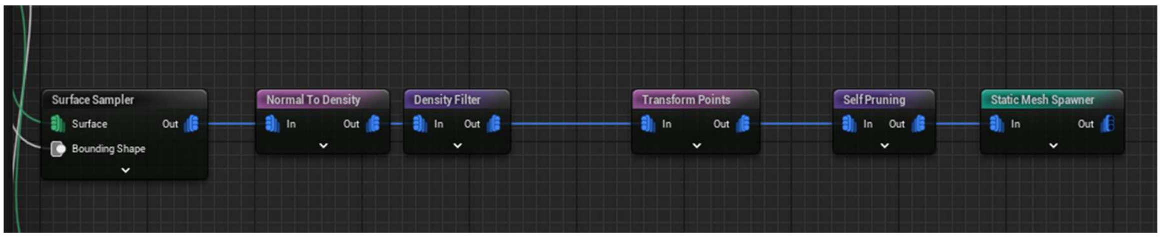


Figure 75: Application of density and pruning nodes to control the spawn of the model

The last step of this process is to spawn the rest of models by configuring the spawn parameter of each one depending on how many times the object will be repeated in the environment. One final feature to take into account is that in the spawn of two models that are common in the scenario, the best option is to use the difference node, making the insert of the props changing which one to display.

After configuring all the nodes and the models, this would be the final result, this space is resizable, and by changing the scale it continues generating the environment. The spaces created form like roads, which will be created in the next chapter.

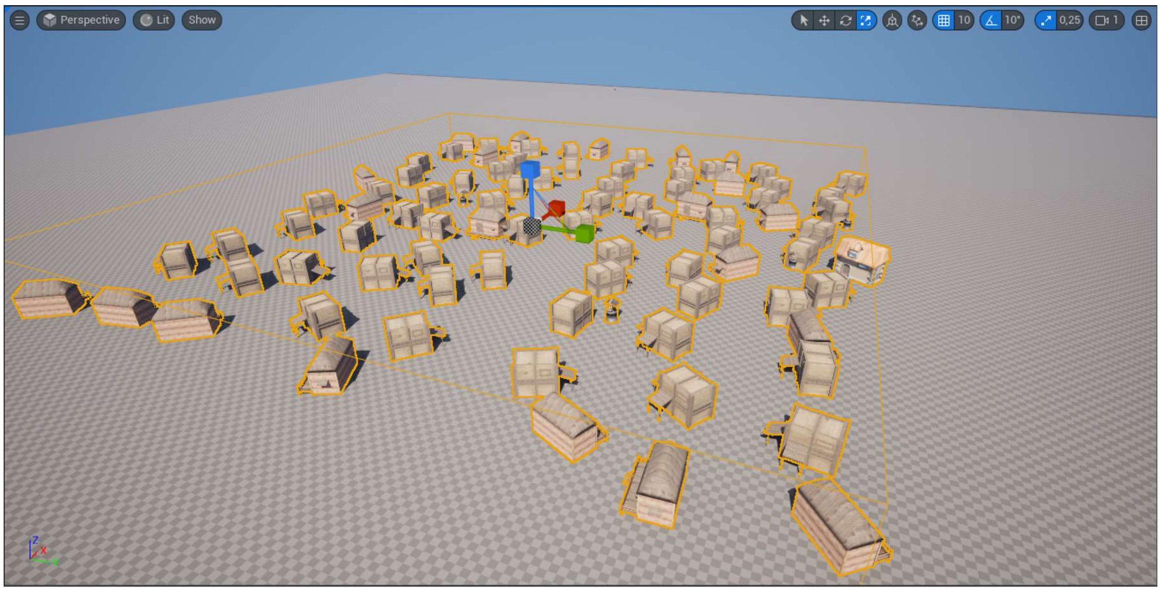


Figure 76: Display of a seed

As this is a procedural tool, it allows experimentation with different seeds and sizes of the bounding box, generating different results.

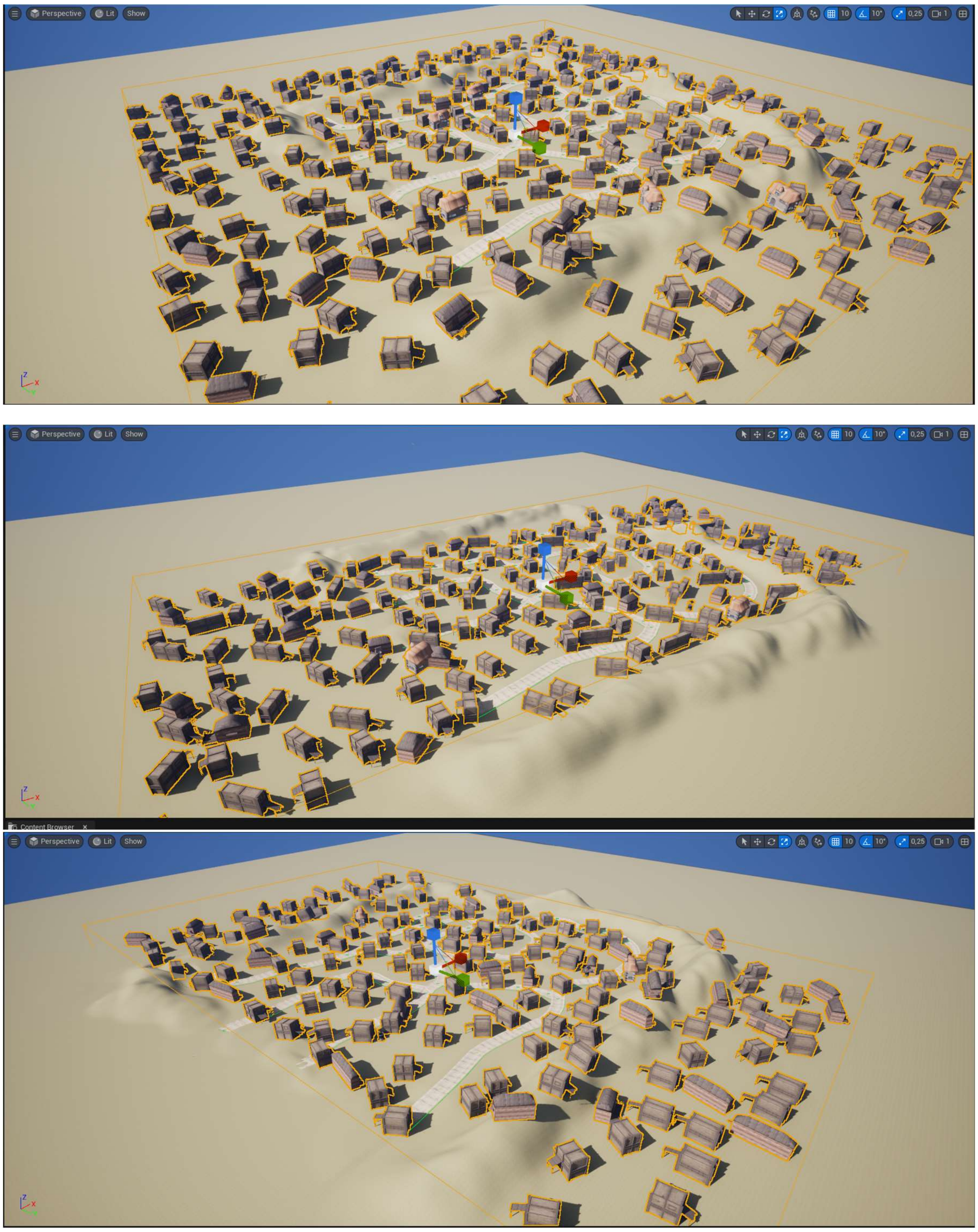


Figure 77: Compilation of different uses and sizes

One of the main characteristics of a procedural generation is the concept of the seed, which is the starting point of an algorithm. This seed is assigned to a number, so if that number changes, the position, rotation and distribution of the whole environment changes, making each seed unique[92].

[92] <Packt> - "Seeds"

For the playable demo of this project, there is a button that can generate a random seed each time it is pressed.

6.2.3 Creation of walks

For the roads of the city the material created will be dirt, which will be created firstly in Substance Designer. This program is focused on material creation, and working also with nodes, allows the creation of a lot of procedural materials. The material to create in this situation will be a dirt road, due to it combining with the thematic of the environment as well as respecting the color palette, adding coherence and more immersion to the level. Another reason behind this decision has been the usage of another tool of Unreal Engine, which is the Blueprint spline.

This component is a path to define and use positional data. It is used to move different components around the world, or place those components along the spline. This component is fully editable both in the Blueprint Viewport and in the Level Editor. How this works is by the deformation of a single static mesh along a two point spline, being those points controllable through Blueprints[93].

In the creation of the dirt road material, it can be divided into different layers, due to it having a bit of depth to give some detail and following the same PBR pipeline as in the models. To make the first dirt mounds, the main aspect has been making some nodes with noise, which is a procedural texture that contains light and dark colors that determine the height (the colors are composed by black and white). Another popular image used in this material is the clouds, which is a kind of noise, but giving an intensity.

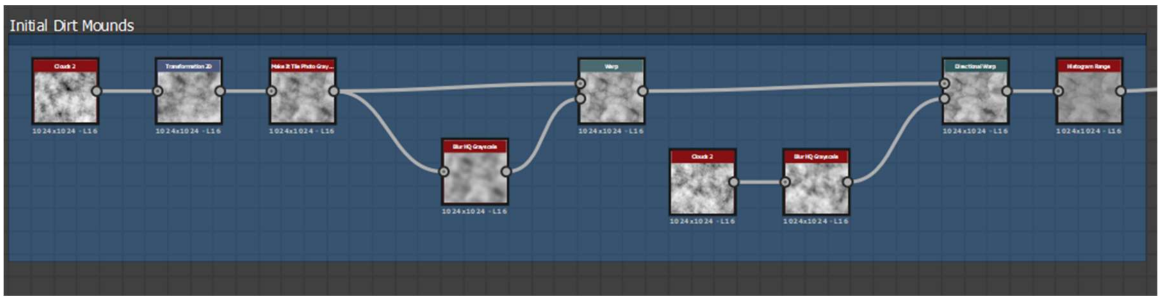


Figure 78: Initial dirt mounds nodes implemented

To simulate the road itself, the clouds and noise have been combined with the shape node, which is a kind of node that allows to create basic forms, like in this case, a rectangle to make the mark of the road, deforming it with some blur to keep the texture more natural.

[93] Unreal Engine - "Unreal 5 Documentation - Blueprint Spline Components Overview" - 2023

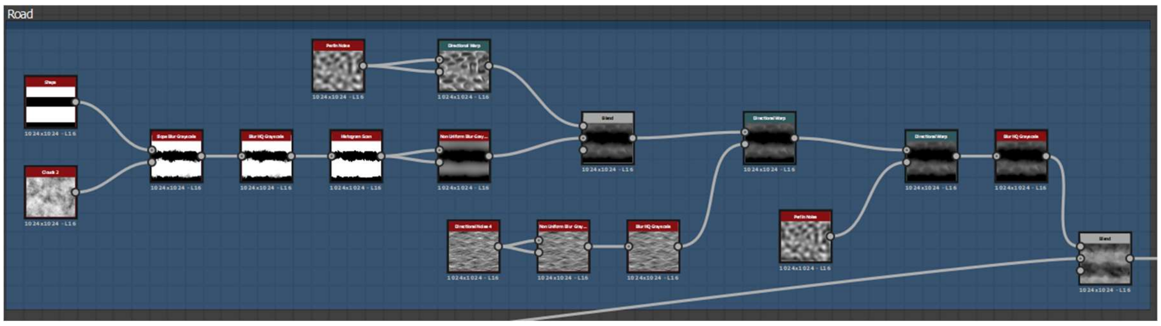


Figure 79: Road limits creation

With these nodes what is made is defining the normal and roughness map, letting the base color for other nodes. The main nodes used for the base color consist of the ambient occlusion, which takes a height map as input and generates an ambient occlusion map for that, then, the next step is to create a gradient map that takes a gradient of colors as reference and is combined with the ambient occlusion to make the different color tonalities.

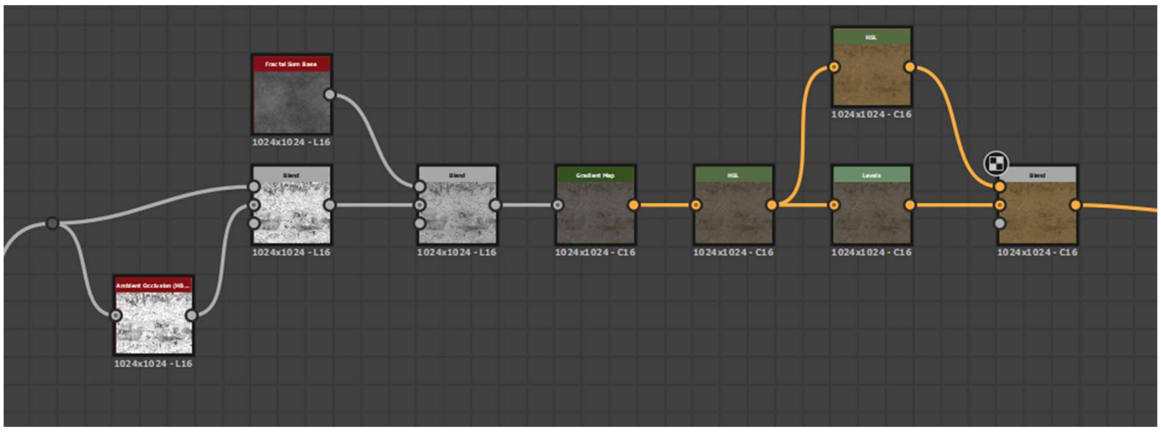


Figure 80: Creation of the base color of the texture

Once the material is created, what is time is to export it to Unreal Engine, which is a very accessible pipeline because it is the same as with Substance Painter, so the most efficient way is to export the textures with the PBR workflow and then creating a material in Unreal to apply the textures.

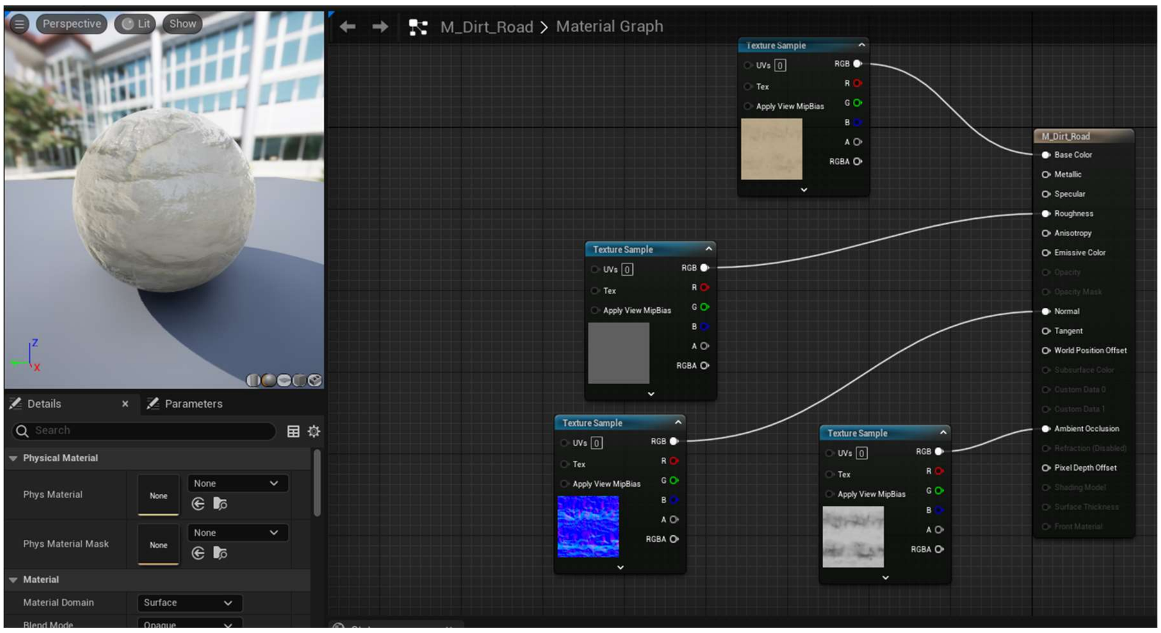


Figure 81: Creation of a material in Unreal Engine

Now what remains is to create the spline script to apply this component and finally create the procedural road for the environment by creating a new layer in the landscape mode of the engine to block this layer to only use the spline component so that it saves from a lot of problems. Once the new layer is created, the process that follows is to select the spline icon inside the management window to start adding the road.

Once the road is set up, the final step is to add a plane that will work as the road and add the material previously created to get the final result. In this situation, the color of the road is a tone clearer in comparison to the rest of the props to create some contrast.

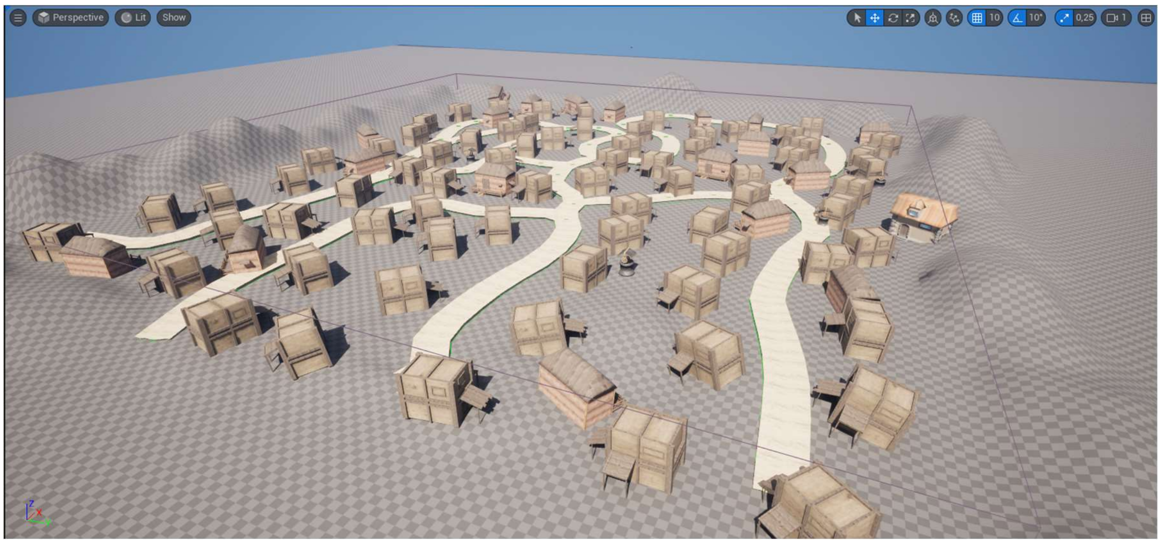


Figure 82: Capture of the road finally implemented.

6.2.4 UI Setting

For the design of the UI, it will consist of two simple buttons, one to generate a random seed each time it is clicked, generating different scenarios, and the other to quit the application.

To create a working UI in Unreal Engine is with the Blueprint class of Widget, which creates a canvas that allows to implement elements like buttons or slider bars so that the user can interact with. In this case the used elements will be the buttons and text, which allows the creation of a button and writing a text on it.

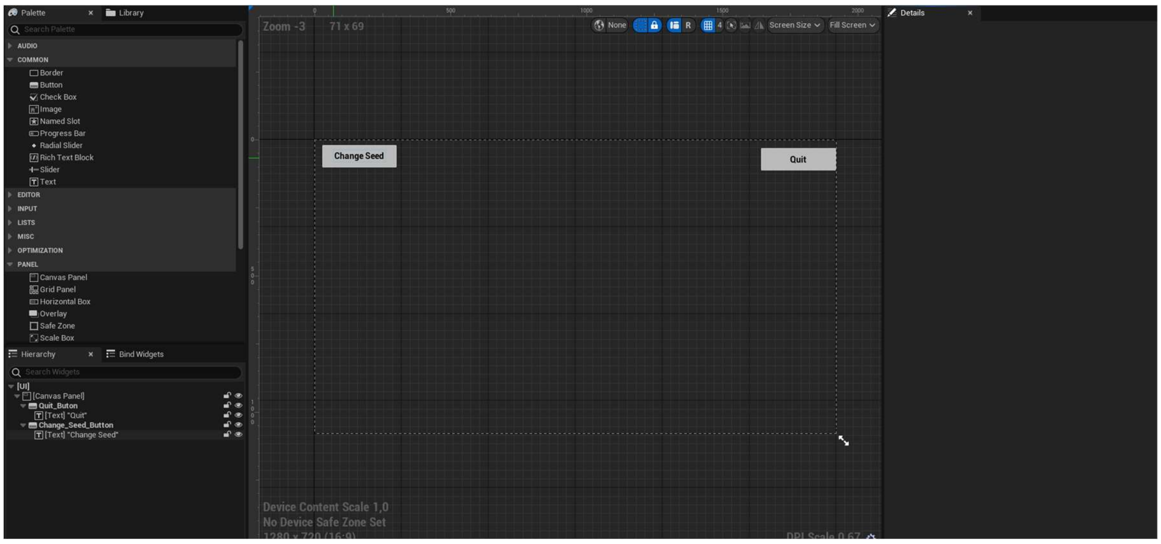


Figure 83: Capture of the Blueprint Widget

Once the design is set up, it is time to make it functional with the event system that Unreal offers. This event system allows the user to assign a certain event each time there is an interaction with the UI. In this case, the event is related when the user clicks a button. As there are two buttons that make different functions, the events assigned will be different.

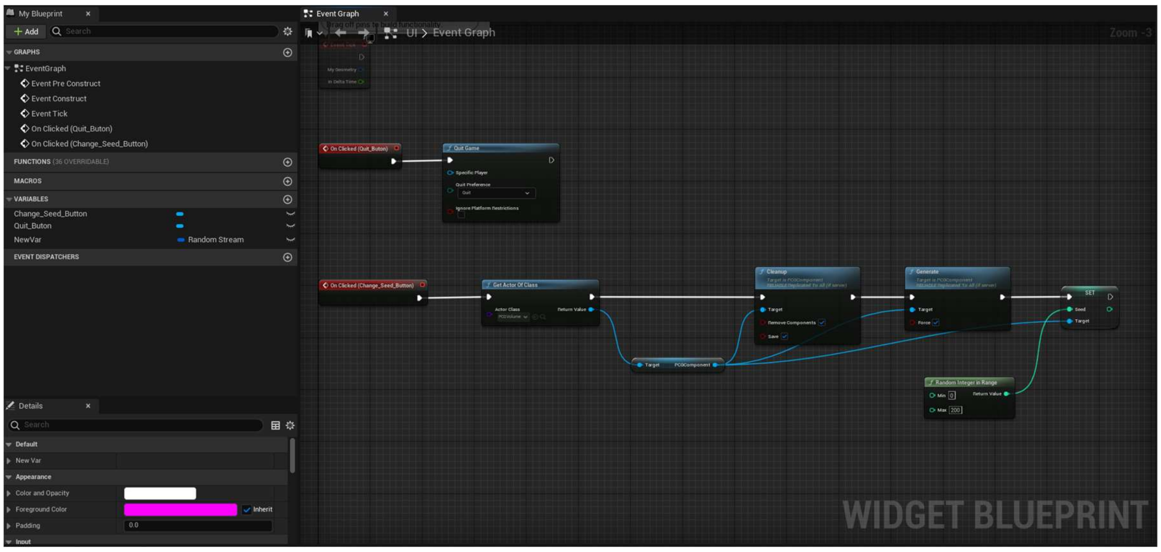


Figure 84: Events of buttons

In the case of the generate seed button, it takes as target the blueprint of the procedural generator and cleans up all the size of the component for generating instantly a new one, setting the seed, which is a random value. On the side of the quit button it only calls the function "Quit game" that Unreal has by default.

To implement this UI to the player, it is necessary to go to the blueprint of the character and establish that when the game starts, it created an UI Widget inside the camera of the player, allowing to display the buttons on screen, ending in this result:

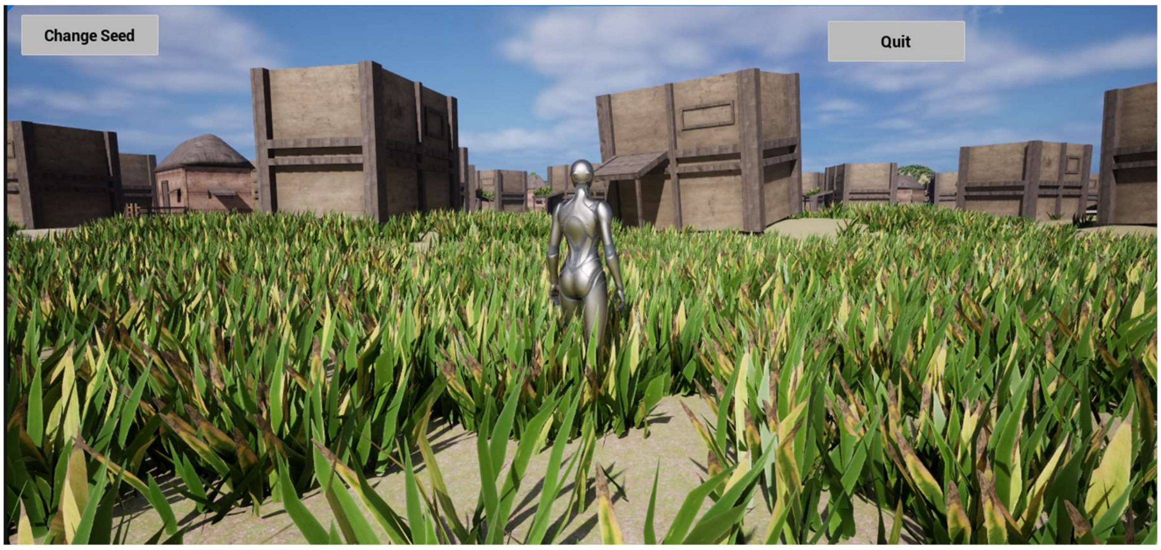


Figure 85: Capture of the UI implemented

6.3 Post production

This is the final stage of the project, and where most of the main tasks of the project are completed, so what only remains is the polishing of smaller details and adding few features that do not affect directly to the development.

6.3.1 Lighting set up

In videogame development the lighting in the environment has formed a bigger part with the pass of time and the updates of technologies, so it has allowed to get more presence at the time of designing a level. In this situation, the unique light source that will form part of the scenario is named the global illumination, which is the light source that will always be illuminating the main environment.

As the scenario to simulate is based on a village that is in the mountains, the source light should have cooler tones than if the environment was located in another place. Another reason behind this decision is that as the color palette for the props is with warmer colors, having a cooler light helps the player not to overload the sight, establishing a visual balance and an unique esthetic.

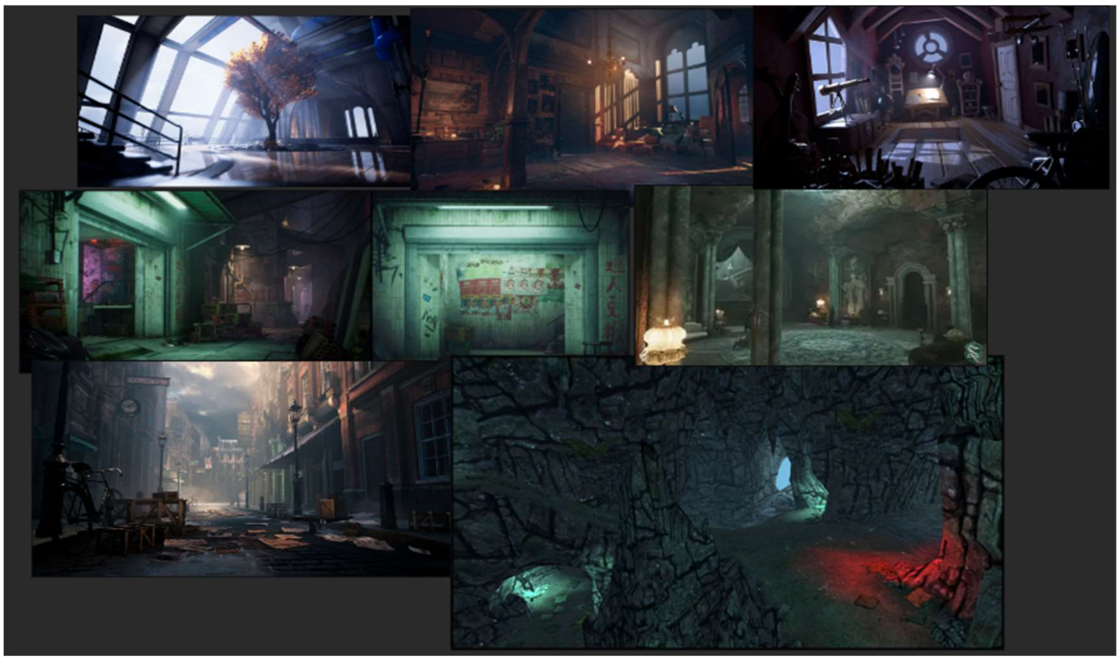


Figure 86: Moodboard of the light for the environment.

In terms of illumination Unreal Engine counts with a solid and accessible editor that facilitates a lot of work, due to it already prepares a light source by default, it is easier to edit the angulation and colors, as well as its temperature.

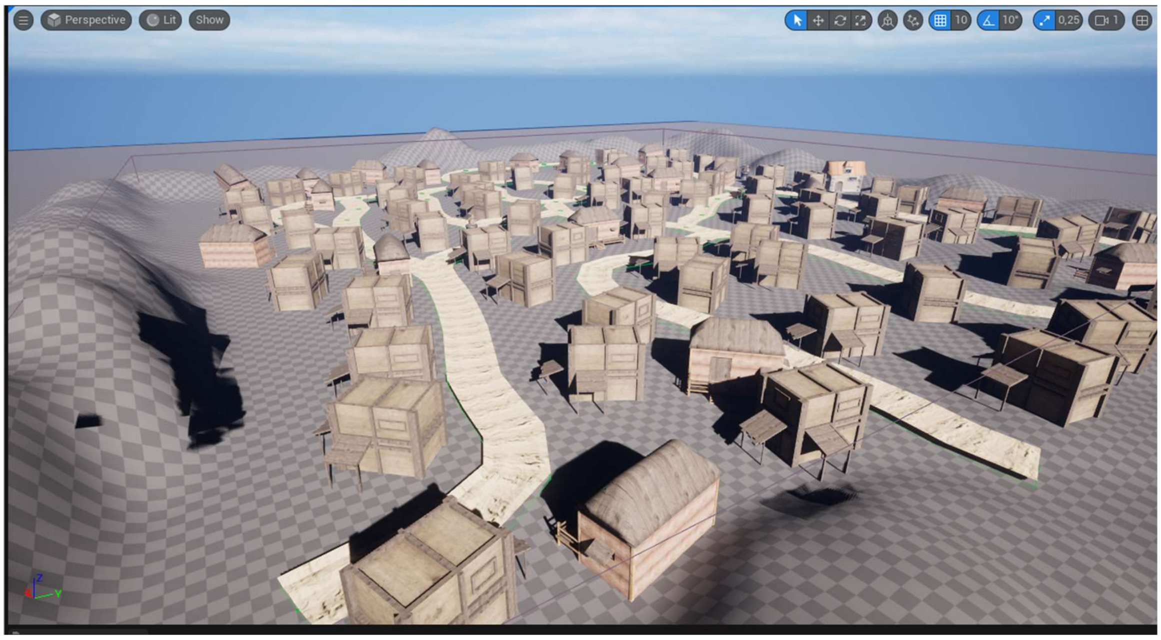


Figure 87: Lighting of the environment

6.3.2 Landscape texture and foliage

To add more detail into the environment, it is important to use a material for the ground, so it could feel more realistic and help the player's immersion, as well as some models for grass to add more detail to the scene.

For the grass material, it has been taken from the page of Poly Haven, which is an open content resource webpage where there are lots of different materials and tools for 3D artists. The main advantage of this gallery is that their license allows the users to use the resources available for any purpose without asking for any kind of credit, making a community that grows every day[94].

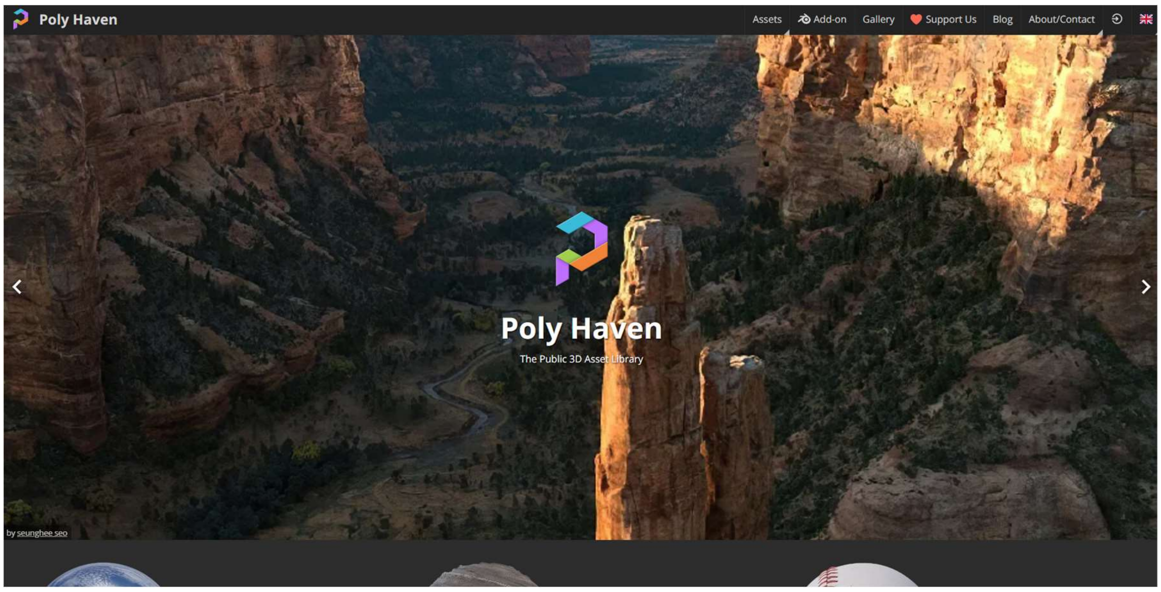


Figure 88: Capture of the main page of Poly Haven

The texture chosen is a grass texture that mixes with ground, so the dirt road does not highlight the environment, looking unrealistic.

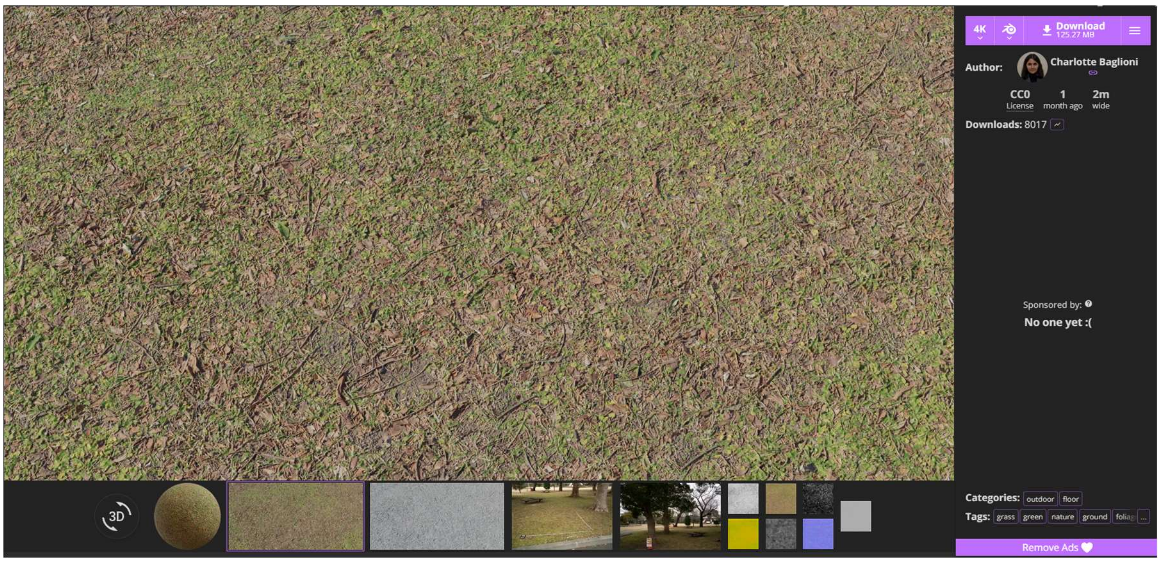


Figure 89:Capture of the texture for the landscape

[94] Poly Haven - "Home Page"

For the grass model, it has been downloaded from another webpage similar to the mentioned above, being Sketchfab, which is also a portal where people can upload their 3D models made as well as download either by free or by paying models made for other people[95].

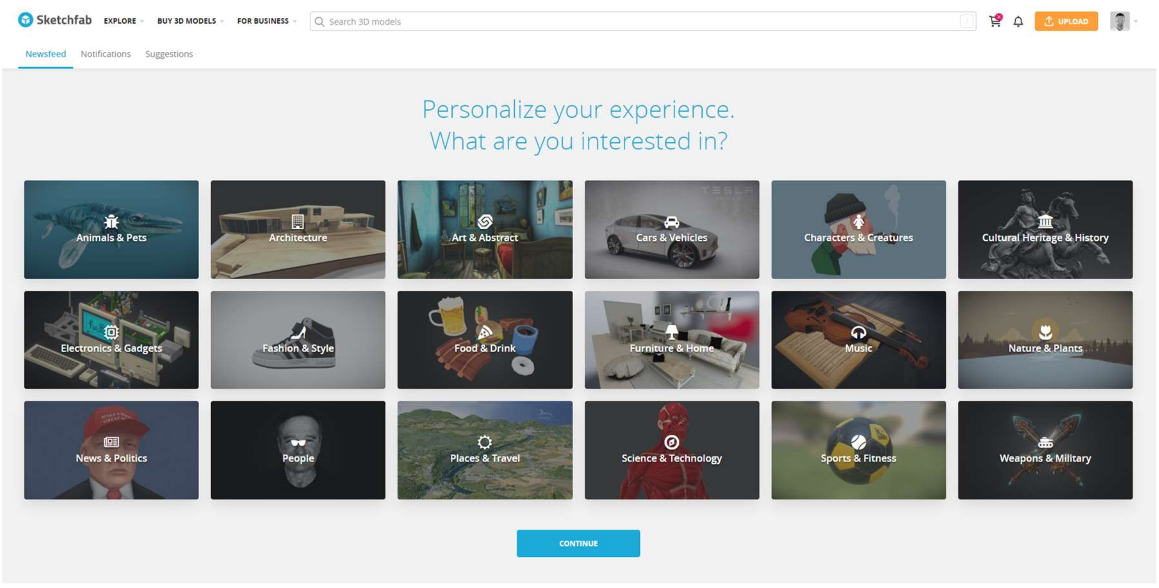
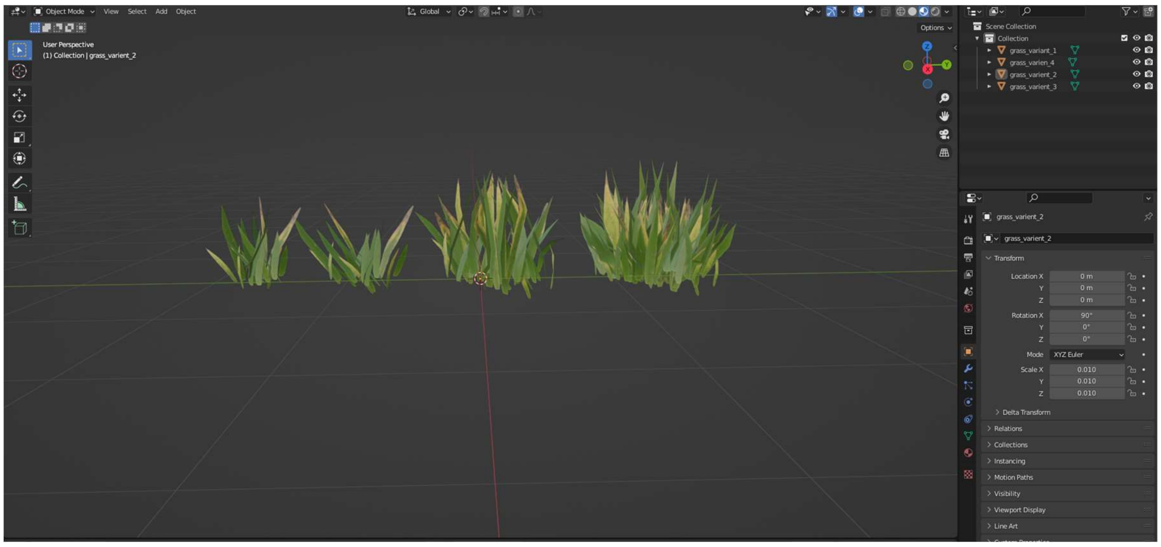


Figure 90: Capture of the main page of Sketchfab

The model consists of different variations of grass, allowing to generate random models in the foliage procedural tool of Unreal Engine.



grass_varient_2

Transform

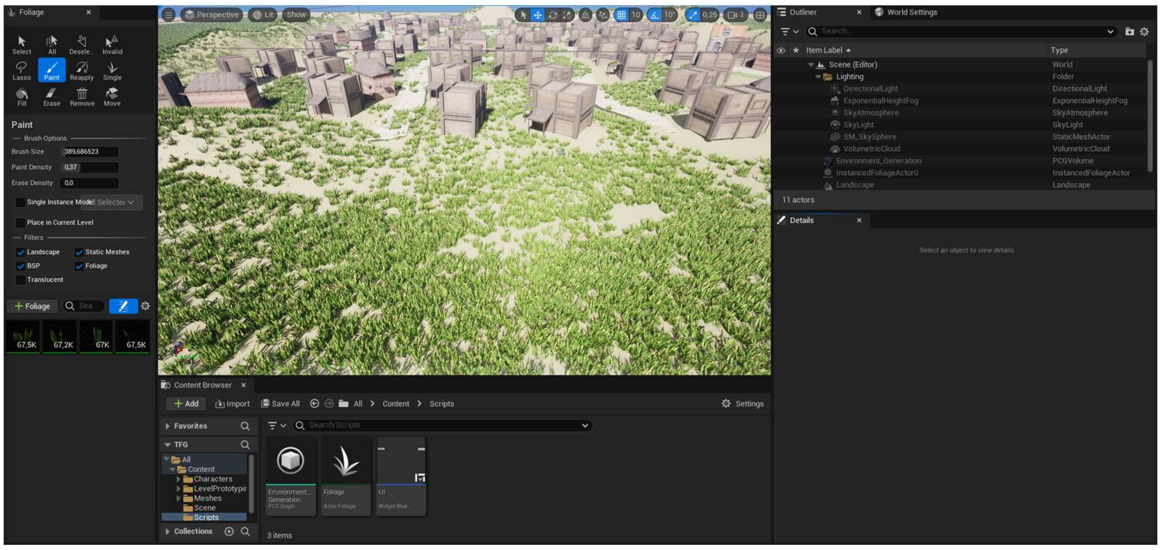
Location X 0 m
Y 0 m
Z 0 m
Rotation X 90°
Y 0°
Z 0°
Mode XYZ Euler
Scale X 0.010
Y 0.010
Z 0.010

Figure 91: Capture of the models for the grass

To implement both effects in the engine, what is necessary is to use the landscape tool. For the texture of the floor, it is necessary to assign the material to the mesh of landscape, being similar to the material assigning of the models for the props.

[95] Sketchfab - "Feed page"

For the grass, it is necessary to import the models to the foliage part of the engine, which allows to display them like if it was painted with a brush, allowing to make different effects of placing the model on the surface.



Paint

Brush Options

Brush Size 389,686523
Paint Density 0,37
Erase Density 0,0

Single Instance Mode

Place in Current Level

Filters

✓ Landscape ✓ Static Meshes
✓ BSP ✓ Foliage
Translucent

67,5K 67,2K 67K 67,5K

Item Label	Type
Scene (Editor)	World
Lighting	Folder
DirectionalLight	DirectionalLight
ExponentialHeightFog	ExponentialHeightFog
SkyAtmosphere	SkyAtmosphere
SkyLight	SkyLight
SM_SkySphere	StaticMeshActor
VolumetricCloud	VolumetricCloud
Environment_Generation	PCGVolume
InstancedFoliageActor0	InstancedFoliageActor
Landscape	Landscape

11 actors

Details

Select an object to view details.

Figure 92: Capture of the brush for the foliage

This would be the final result after applying the post processing effects.

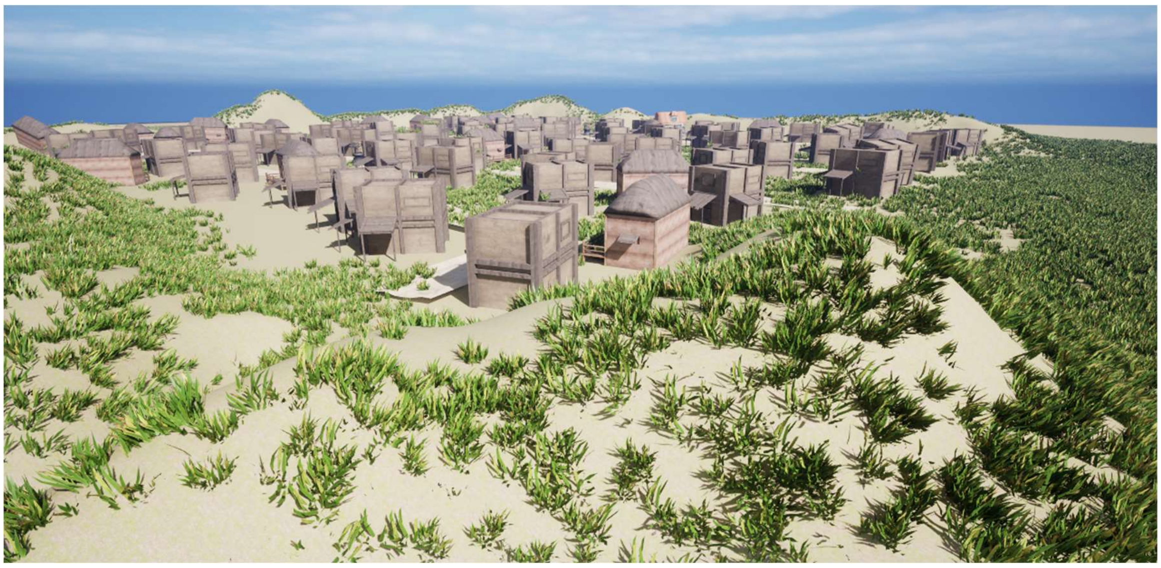


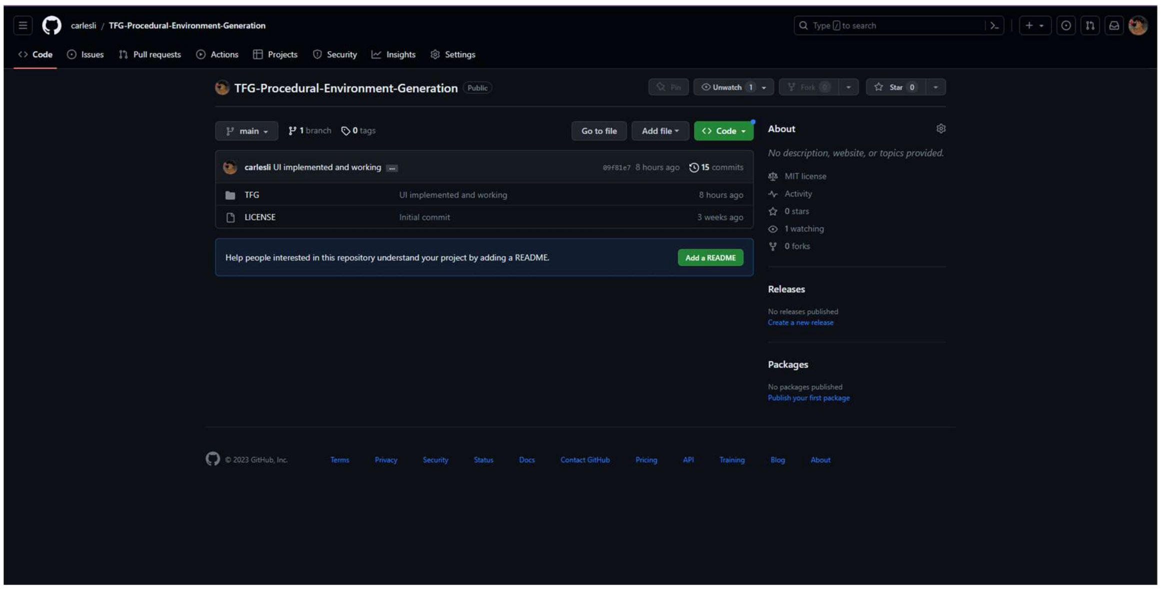
Figure 93: Render of the environment

6.3.3 Portfolio showcase

Once the project is finished, it is time to publish it on the Internet and make a showcase of the result. The most common process to upload any project of this type is to use portfolio pages that are built for this purpose, like ArtStation, Github or LinkedIn, allowing it to have certain repercussions in each community as well as a way to show someone's capabilities.

In the case of this project, the uploading of the project will consist in two main parts, one uploading the project to Github and a playable demo also in Github, where the users will be able to test the power of the tool before downloading it.

For uploading projects, Github is one of the best options for uploading the project, because the users can search for it and download the source code for free and without installing any external plugin. The users will only need the version of Unreal Engine 5.2 to download the project and start using it, counting with different files like basic scripts for the player and the foliage, as well as the tool itself with the models used in this project and an example scene to show its possibilities.



TFG-Procedural-Environment-Generation Public

main 1 branch 0 tags

carlesli UI implemented and working 09f81e7 8 hours ago 15 commits

TFG	UI implemented and working	8 hours ago
LICENSE	Initial commit	3 weeks ago

Help people interested in this repository understand your project by adding a README.

No description, website, or topics provided.
MIT license
Activity
0 stars
1 watching
0 forks

No releases published
Create a new release

No packages published
Publish your first package

© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

Figure 93: Capture of the repository of Github

The other part of the showcase is uploading a release which consists of a file in Google Drive with the playable demo of the project, allowing the users to understand better how it can be applied to a videogame and its behavior once a build is made. This demo consists of a character that can move through the environment and two UI buttons, one to generate a random seed each time it is pressed and another to quit the application.

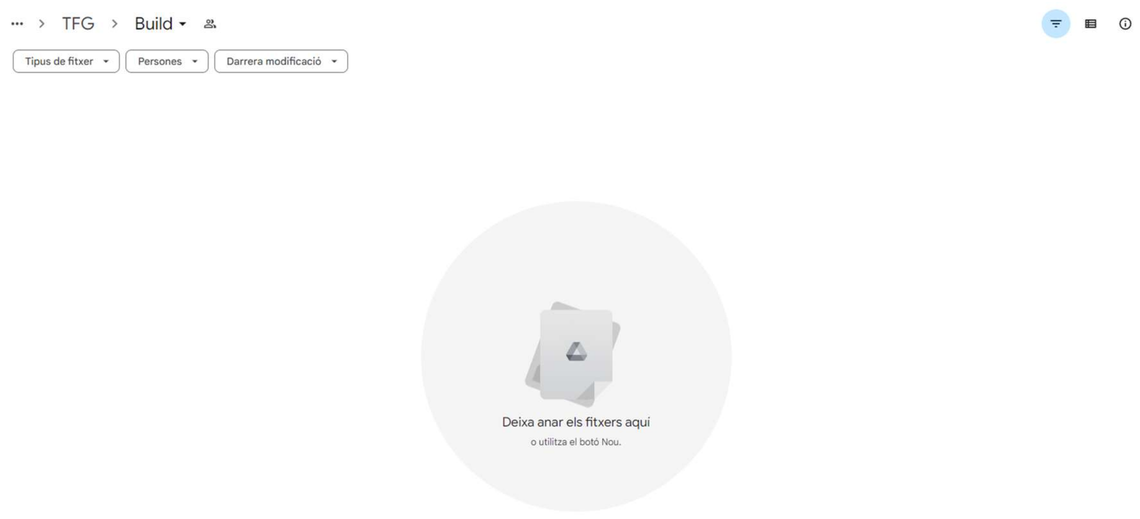


Figure 94: Capture of the build folder

This is the render image that will conform to the front page of the project.

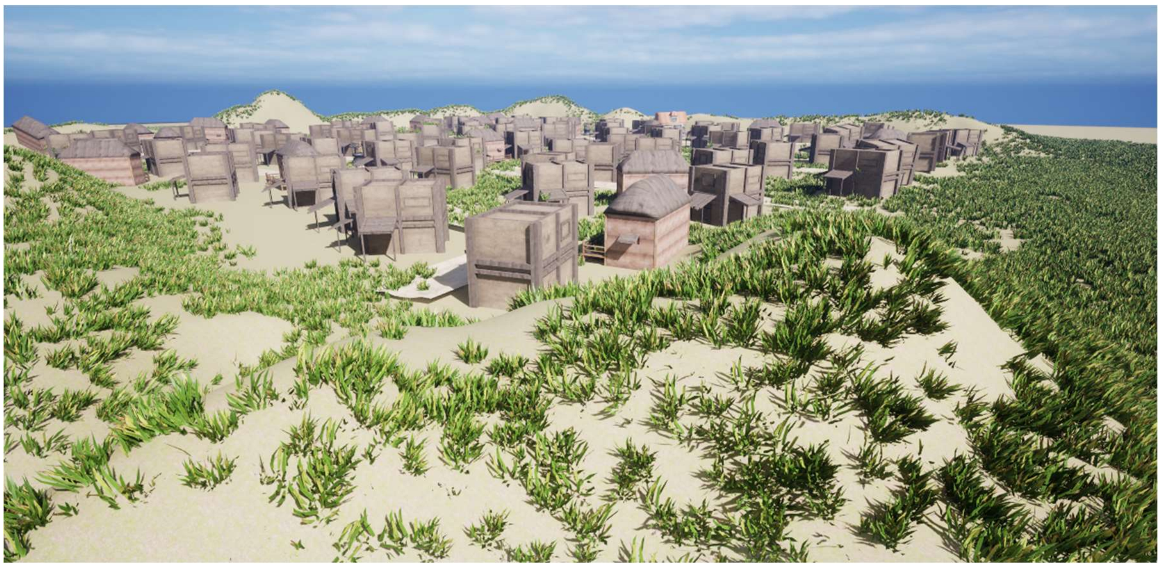


Figure 95: Render of the project

7. Objective test

A form has been created with the purpose to know if people thought what was procedural generation and to know their opinion about the concept, being made by many people, making the range from people who are in the videogame industry to people that do not belong to it, so the range to get is bigger to get more information. The questions have been divided into two different sections, one with the people who knew about the topic and the other with people who did not have much knowledge about it.

The first form asks questions about people who have previous knowledge about the topic, so the focus is to ask about methods and known techniques, possible challenges and the perspective about the future of the procedural generation.

On the other hand, the other form focuses on people who do not have that previous knowledge and try to introduce them to the topic with the possibilities it can reach, or the problems it can carry on in the future.

In the chapter of "Annexes" there is the complete form with detailed information about the answers, as well as the analysis of them is located in the next chapter, which is the "Conclusions".

8. Conclusions

From the questionnaires made it is remarkable that during the last years, more and more people are curious about how the videogames are made, carrying this curiosity to the learning about the techniques applied, making that people that are not inside the industry learn about those topics and how they are made to grow a better critical eye that determines which product is better.

Another remarkable conclusion to add from the questionnaires is that more or less the half of answers about the possibilities of the implementation of procedural generation in videogames are positive, seeing this technology as more an advantage than a disadvantage, despite the fact that a part of them are not very familiar with the games with this implementation.

On the other hand, the people that did not have much knowledge about the topic, the answers kept positive, even surpassing the statistics from the other section, taking into account that less people answered the negative question.

After analyzing both sections of the questionnaire, it is clear that this topic has been taking strength during the last few years, so as a consequence of it, most people have been investigating the topic and seeing the benefits it can bring, ending in a more positive reception.

Entering in the retrospective, comparing the objectives established at the beginning of the project and the objectives achieved at the end, many of them have been successfully achieved.

Starting with the specific objectives, they have been completed one by one, ending in a significant upgrade of the skills required to develop a project of this magnitude, one artistic to have a critical eye about the quality of the models and the coherence of the whole environment, as well as a technical part that has ended up in a deeper knowledge of the engine selected, either opening new opportunities for upgrading the project at a later time or starting the development of a bigger project.

As a personal conclusion after the development of the project and all the investigation, the final conclusion to be drawn is that procedural generation is a very useful tool that helps to the work of generating content by facilitating the working pipeline and it can be a great tool to apply in the videogame industry due to the possibilities it can reach. During the last few years, this technique has been making bigger and bigger steps to consolidate as what it is nowadays, meaning that it will grow up right with the upgrade of the technology, waiting for a promising future.

9. Bibliography

- Kaedim - Medium - "10 Tips for creating efficient 3D pipelines" - November 23, 2022
- Chris Casmenco - Anim8 - "3D Production Pipeline" - February 21, 2018
- Tiger Collins - Medium - "3D Modeling Pipeline" - June 22, 2018
- Kevuru Games - "Choosing an art style for your videogame" - October 21, 2022
- Ratloop Games Canada - "The Art Pipeline"
- Thomas Denham - Concept Art Empire - "What is UV Mapping and Unwrapping?"
- Image taken from "What is UV Mapping and Unwrapping?"
- A23D - "What is 3D texturing?"
- Fernando Bevilacqua, Marcos Cordeiro d'Ornellas, Cesar Pozzer - "A survey of procedural content generation techniques suitable to game development"- November 2011
- Refactoring Guru - "Parameterize Method"
- Fernando Bevilacqua, Marcos Cordeiro d'Ornellas, Cesar Pozzer - "A survey of procedural content generation techniques suitable to game development"- November 2011
- Jasmine Bonner - "10 Games with Procedural Generation" - March 9, 2022
- Computer Hope - "Bézier Curve" - July 2, 2022
- Unity Documentation - "Publishing to the Asset Store" - March 17, 2023
- Unreal Engine - "Sharing and Releasing Projects" - 2023
- Image taken from "A kickass Game Art Portfolio" - Midnight Hub - erik - June 2, 2016
- Joe Bernardi - Medium - "Texture Atlasing: An Inside Look At Optimizing 3D World" - January 24, 2018

10. Annexes

In this section there will be the questionnaire made to external people with the objective of data compilation. This questionnaire has been divided into two sections, the first one where people have a certain knowledge about the topic of procedural generation and the second where people do not have much knowledge about it. These sections have been determined by the answer of the first question.

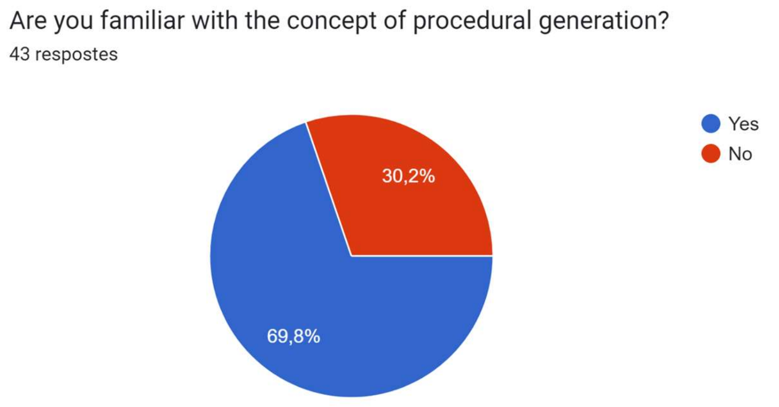
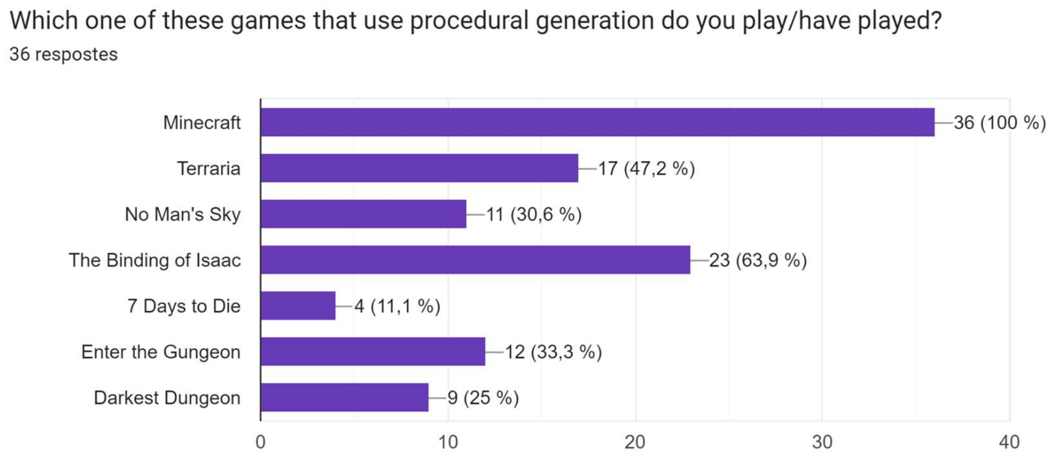
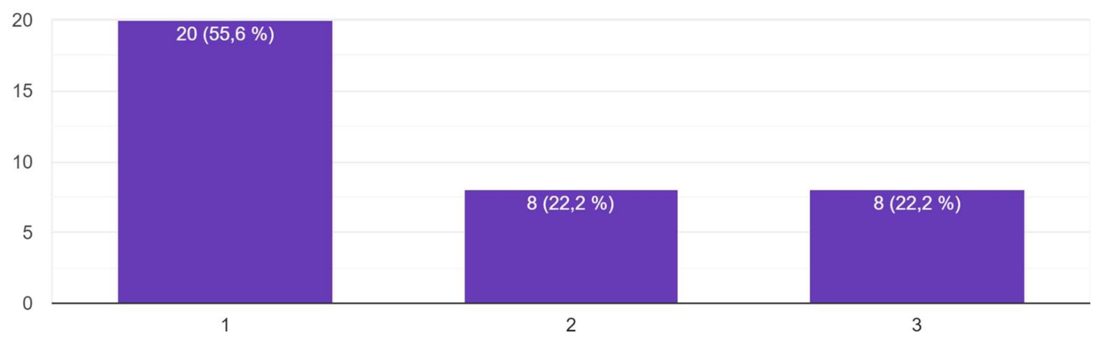


Figure 96: First question of the form

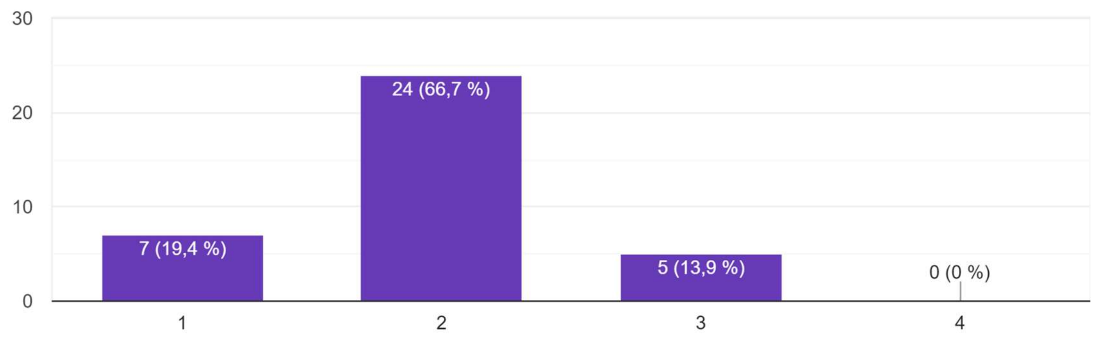
The first section has been filled with mostly technical questions, implying that this is a known topic.



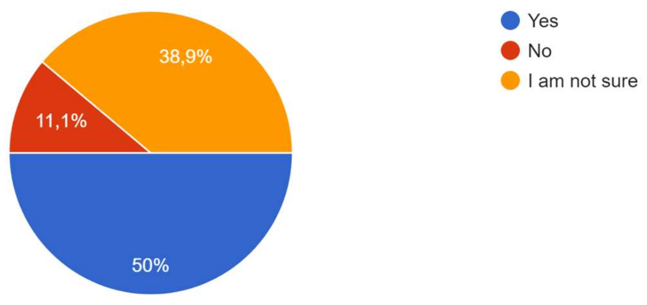
How familiar are you with videogames that apply procedural generation?
36 respostes

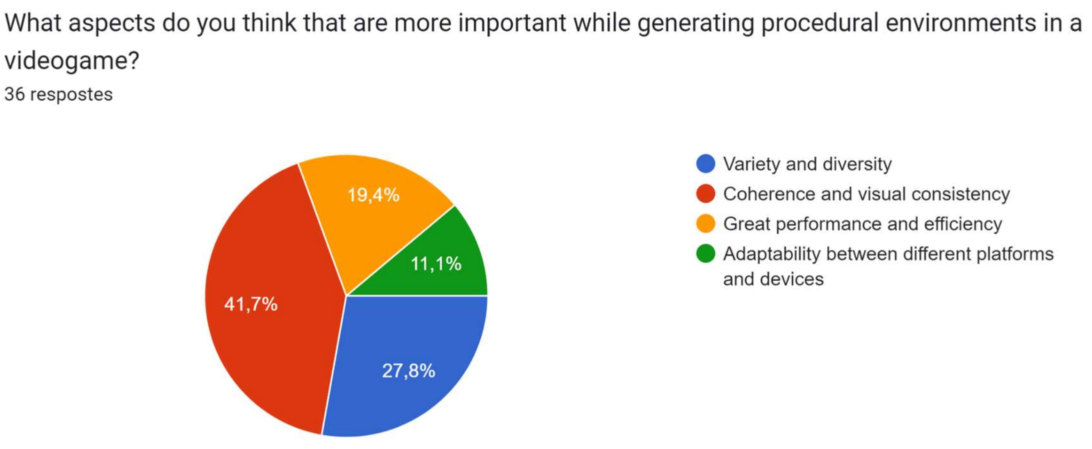


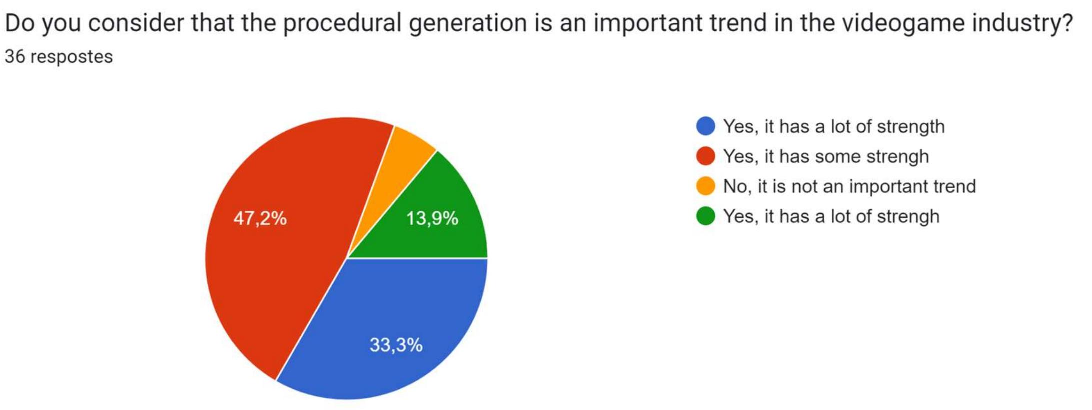
Related with the las question, what is your opinion about the procedural generation in the environments?
36 respostes



Do you consider that videogames with procedural generation in its environments can upgrade the game experience in comparison with scenarios created manually?
36 respostes







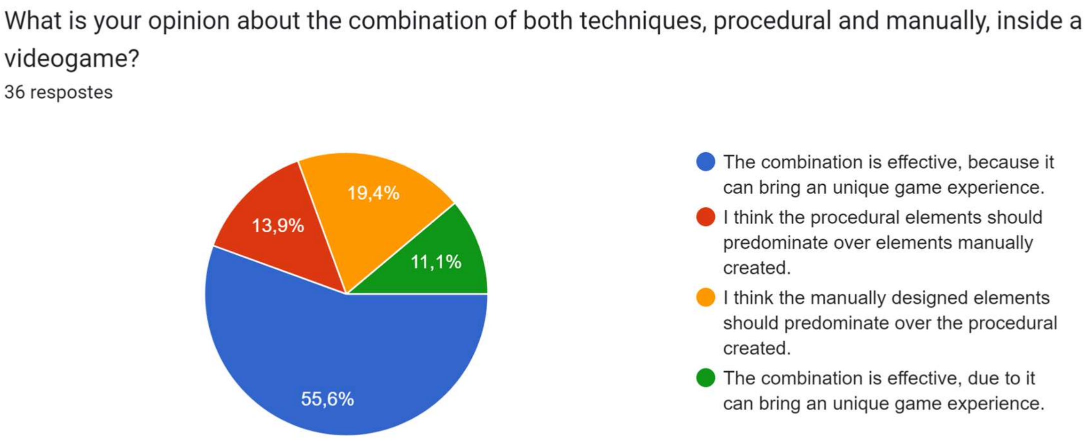
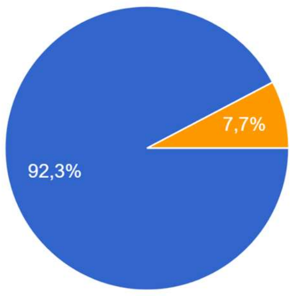


Figure 97: First section of the form

In the second section as the focus was for people that did not have much knowledge about procedural generation topic, the questions were to introduce the topics and let the users consolidate an opinion in base of basic explanations.

In the context of videogames, what do you think that "procedural generation" means?

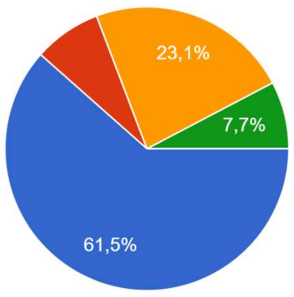
13 respostes



- ● Automathic creation of content in a videogame using parameters and rules.
- ● The content is manually designed by the developers
- ● I am not sure

What benefits do you think that can offer the procedural generation in an environment of a videogame?

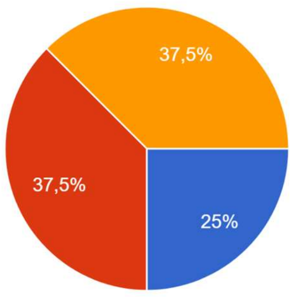
13 respostes



- ● More variety and diversity in the gameplay.
- ● More attractive visual and detailed levels.
- ● I am not sure.
- ● Opció 1

What kind of elements do you think that could be generated with procedural techniques?

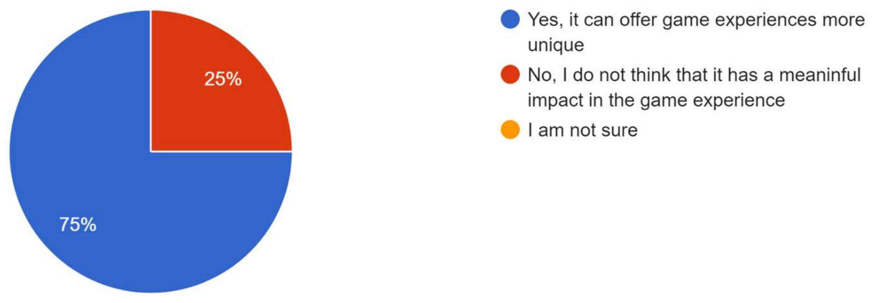
8 respostes



- ● Terrains and landscapes
- ● Buildings and structures
- ● Objects and non-decorative elements
- ● I am not sure

Do you think that procedural generation applied to videogames can affect in the experience of the gameplay?

8 respostes



In overview, do you think that procedural content generation is a promising trend in the videogame development?

8 respostes

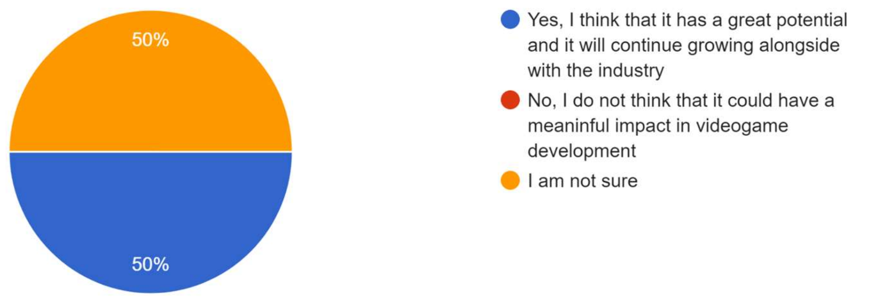


Figure 98: Second section of the form