

# APPLICATION OF MODELING AND TEXTURING TECHNIQUES FOR GAME READY LEVELS

---

FINAL DEGREE WORK

BACHELOR'S DEGREE IN VIDEO GAME DESIGN AND DEVELOPMENT

UNIVERSITAT JAUME I

Author: Adrián Bautista Gil

Supervisor: Sven Casteleyn

July 2021, Valencia



## ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor for the Final Project, Sven Casteleyn, for his assistance before the development started, as well as for his advice during the duration of the project.

I also want to thank my girlfriend for her constant support during these months and for helping me during the hardest and most difficult moments.

## ABSTRACT

This document establishes and summarizes the creation of a game level focusing on 3D art and optimization techniques. This project explores the development of a demo level that, ideally, could be in a game. It will explore modeling and texturing techniques that improve the reusability of the 3D models and employ diverse methods to optimize the final result. Another crucial point is the application of methods that bolster up the level of detail in the scene and break the repeatability.

## KEYWORDS

- ❖ UE4
- ❖ Level Design
- ❖ Texturing
- ❖ 3D Modeling
- ❖ Optimization

## CONTENTS

Universitat Jaume I .....	1
Acknowledgments .....	3
ABSTRACT .....	4
KEYWORDS .....	5
1 INTRODUCTION .....	9
1.1 Work Motivation .....	9
1.2 Subject Areas.....	9
1.3 Objectives .....	9
1.4 Environment and Initial State.....	10
1.5 Technologies and tools .....	10
2 Planning and resources evaluation .....	12
2.1 Planning.....	12
2.2 Resource Evaluation .....	14
2.2.1 Hardware.....	14
2.2.2 Software .....	14
2.2.3 Salary .....	15
2.2.4 Conclusion.....	15
3 System Analysis and Design.....	16
3.1 System Analysis.....	16
3.2 Design.....	19
3.2.1 Planning .....	19
3.2.2 Top Down .....	20
4 Work Development and Results.....	21
4.1 Work Development.....	21
4.1.3 Blockout.....	21
4.1.4 Landscape .....	22
4.1.5 Vegetation And Foliage.....	28
4.1.6 Architecture / Modularity .....	33
4.1.7 Set dresssing .....	35
4.1.8 Lightning and Post-Processing .....	40
4.2 Results.....	41
4.2.1 Time balance and deviations .....	41
4.2.2 Achieving the objectives .....	42
4.2.3 Links .....	42
5 Conclusions and Future Work.....	43
5.1 Conclusions .....	43
5.2 Future work .....	43
Bibliography.....	44

List of Figures .....	46
List of Tables .....	47
List of Schemes .....	47





## 1 INTRODUCTION

### 1.1 WORK MOTIVATION

The idea of developing this work has been something I had been thinking about for some time. Throughout the degree we have been provided with basic notions of 3D modeling, but it has not been deepened at a practical level in the different techniques that are applied in the industry, both when planning the creation and reusability of the assets, as in the different workflows to apply.

In this project I seek to explore the application of different techniques for the creation of 3D assets, as well as the construction of a scenario from these props and the application of techniques to give greater variety.

In this project I also seek to apply optimization techniques and what that may imply when creating 3D assets.

### 1.2 SUBJECT AREAS

Subject	Code
3D Design	VJ1216
Graphic Computer Science	VJ1221
Video Game Engines	VJ1227

Table 1 - Subject areas related to the project

### 1.3 OBJECTIVES

My main goal with this project is to develop a relatively large virtual scene, within a game engine. Creating a set of assets and textures that are ready to be implemented in a real game. As well as to document the entire development process and reflect it in this report.

In order to create a scene like the one I describe, it will be necessary to explore different techniques and tools that allow me to optimize the limited time I have. Some of the key points are:

- ❖ Use a variety of modeling techniques to favor reusability and time optimization.
- ❖ Apply different texturing techniques to create large seamless terrains.
- ❖ Add VFX and use shaders to improve the final result and break repeatability.

## 1.4 ENVIRONMENT AND INITIAL STATE

For the development of this work and the application of the different work techniques that I want to use, I require the use of different software tools that allow me to address the different phases of each workflow. For the development of this project, I need a powerful computer and a graphic tablet.

I wanted to plan the development of the work with the same phases that would be used in a real project, although these phases are not completely rigid.

- ❖ **Stage 1:** Planning - Consist in making a gathering of pictures or concept arts that could be useful as references.
- ❖ **Stage 2:** Level Design / Top Down - A 2D representation of the level which consist in a composition for the level depending on the objectives that the player has to accomplish and the gameplay mechanics available to the player.
- ❖ **Stage 3:** Blockout - The 3D representation of the top down.
- ❖ **Stage 4:** Landscape and Vegetation - Develop the textures and shaders used for de terrain, as well as the vegetation props that will cover the surface.
- ❖ **Stage 5:** Architecture - Generate the modules used to construct all the buildings in the scene, as so as the textures they will use.
- ❖ **Stage 6:** Set Dressing - Create the smaller props that will cover the scene.
- ❖ **Stage 7:** Lightning and Post-processing.

## 1.5 TECHNOLOGIES AND TOOLS

As mentioned previously, I will use an extensive assortment of Softwares in the workflow of the project. These are the main tools:



- ❖ **UE4** - Unreal Engine is going to be the game engine I will use for the project, the main reason is that it offers some facilities to create some visual effects or shaders which, in other engines, would require more technical knowledge.



- ❖ **3ds Max** - This is a widely known modeling software, in this project, I will use 3ds Max to create low poly models that are going to be included directly at the scene or used as a base to produce more detailed high poly versions. This software will be used to unwrap the UVs of the meshes as well.



- ❖ **ZBrush** - It is the current standard software used to sculpt, this software enables the creation of highly detailed meshes. In my case, I will use these meshes to bake all the data into low poly models that the engine can handle better.

- ❖ **Substance Suite** - It is a set of tools to create and apply materials for 3D



- **Substance Painter** - Substance Painter is a 3D painting software that allows you to texture and render your 3D meshes. It works with a layered structure similar to photoshop.



- **Substance Alchemist** - Substance Alchemist allows you to create and iterate on material collections by tweaking and mixing existing materials, or by extracting new materials from scans (single or multiple images).



- **Substance Designer** - Substance Designer is a node-based non-destructive application for creating materials.



- ❖ **Google Earth** - Google Earth is a geobrowser that accesses satellite and aerial imagery, topography, ocean bathymetry, and other geographic data over the internet to represent the Earth as a three-dimensional globe. Combined with street view it is a really useful tool to get real world references.



- ❖ **Terre Sculptor** - Terresculptor started as a utility to convert heightmap file formats to Epic's proprietary format, today it is more powerful and feature-rich.



- ❖ **Photoshop** - It is a graphics editor developed by Adobe that is mainly used for photo retouching.

## 2 PLANNING AND RESOURCES EVALUATION

### 2.1 PLANNING

This section explains the distribution of time that has been made for the different tasks, these have been represented both in a table and in a Gantt chart. The tasks have been divided mainly according to the techniques that each of one will require, and the tasks resulting from the development of all the necessary documentation have also been accounted for.

Task	Type	Hours
Reference gathering	Design	8
Technical Proposal	Documentation	4
Level Top-Down	Design	8
Analysis and design document	Documentation	8
Generate Terrain from Real-World Data	Modelling	2
Blockout	Design	14
Create Terrain Material	Shading / Texturing	40
Create Vegetation and Trees	Modelling / Shading / Texturing	40
Create Architecture	Modelling / Texturing	40
Set Dressing	Modelling / Texturing	46
Particles	VFX	18
Lightning	Lightning / VFX	20
Memory	Documentation	40
Presentation	Documentation	10
<b>Total Hours</b>		<b>300</b>

Table 2 - Initial planning

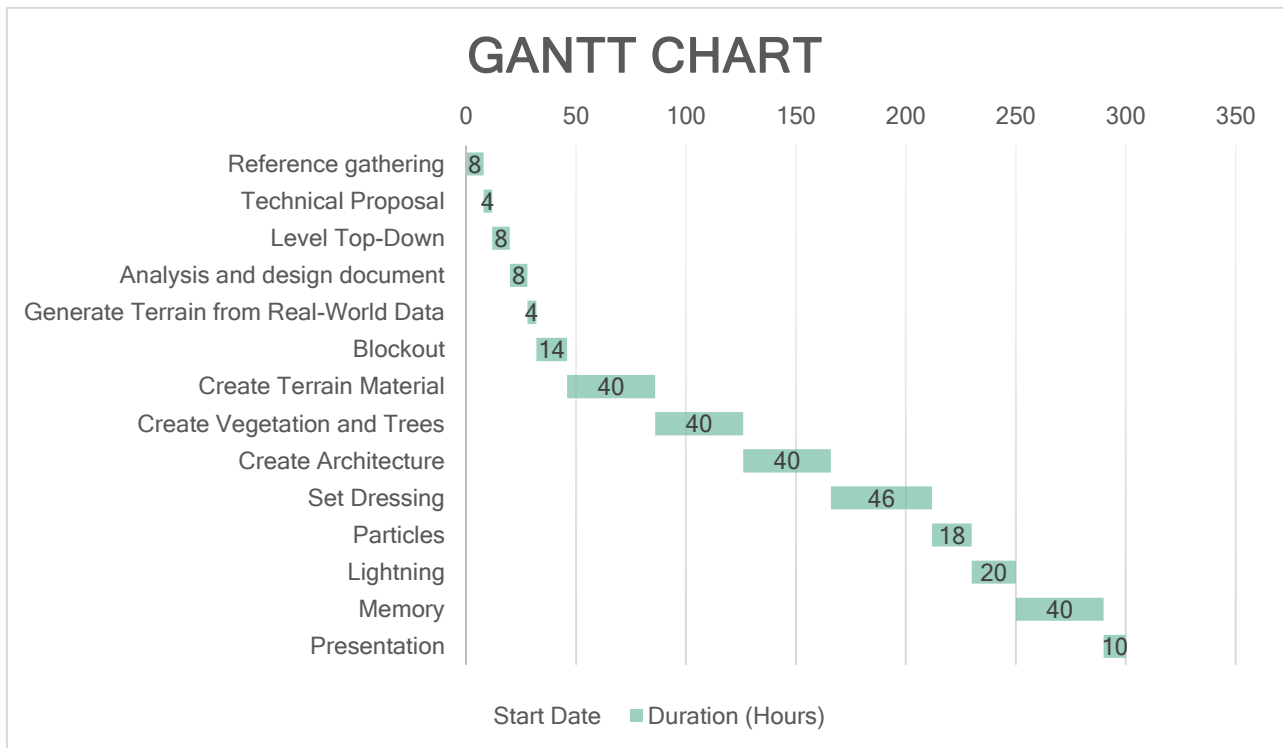


Fig. 1 - Gantt Chart: Tasks Initial hours distribution

## 2.2 RESOURCE EVALUATION

The cost of the work depends on multiple factors such as the softwares used or the quality of the equipment, as well as the worker salary. Here it is an analysis about the theoretical amount of money and time that the project would cost.

### 2.2.1 HARDWARE

The main specifications of the computer used for the project are the following:

Laptop specifications	
Processor	Intel(R) Core(TM) i7-10870H
GPU	Geforce RTX 3070 8 GB
RAM	32 GB

Table 3 - Laptop specifications

Any PC with similar specs will give a good performance to reduce the production time, either to improve the creation of high poly models and high resolution textures or the work in the game engine. This kind of PCs have a cost rounding between 1500 - 2000 €. In case the computer is only required for this project it is possible to opt for a rental, the cost per day could be around 8 - 9 €.

With the rental option, the cost would be 810 € for three months of work.

### 2.2.2 SOFTWARE

Apart from the hardware, the software can significantly increase the costs of a project like this, the majority of programs used have some kind of free license for students, nevertheless in a real case this license wouldn't be valid:

- ❖ Substance Suite has a subscription license-based system, the license cost depends on the enterprise revenue:
  - Revenue under \$ 100k: \$19.90 /month.
  - Revenue from \$100K to \$100M: \$99.90 /month.
  - Revenue under \$ 100k: Tailored for each enterprise.
  - Student license: Free.
- ❖ 3ds Max has a subscription system as well:
  - Paid monthly: € 279.
  - Paid annually: € 2.245.
  - Paid every 3 years: € 6.063.
  - Student license: Free.
- ❖ Zbrush use two types of license, by subscription and perpetual:
  - For Individuals:
    - Single user monthly subscription: \$39.95 /month.

- Single user 6 month subscription: \$179.95 / 6 months.
  - Single user perpetual license: \$895.
- There are students versions, but still paid.
  - And license packs for companies.
- ❖ Unreal has a royalty system: 5% royalty applied to the product over \$1 million lifetime gross revenue.

---

### 2.2.3 SALARY

The salary can vary depending on the experience of the 3d artist, let us say that the average hourly salary is 15 €.

---

### 2.2.4 CONCLUSION

Considering the data provided in this section, it is possible to simulate what it would entail to develop a work of this style. Taking the hours of work required for the development of this project, as shown in section 2.1, and assuming that these would be distributed over approximately three months of work, the total cost of the project in terms of the company's income would be:

	Revenue under \$ 100k	Revenue from \$100K to \$100M	Revenue under \$ 100k
<b>Hardware</b>	810	810	810
<b>Software</b>	987,99	1.186,8	-
Substance	50,19	249	-
3ds Max	837	837	837
Zbrush	100,8	100,8	100,8
<b>Salary</b>	4500	4500	4500
<b>Total</b>	6297,99	6496,8	-

Table 4 - Total project cost in €

### 3 SYSTEM ANALYSIS AND DESIGN

In this section I will discuss the elements related to the artistic design, the requirements on which it depends, as well as the work schemes that will be applied to generate this content.

The first thing I want to analyze is the type of look I intend to emulate in my scene, my main goal is to generate a level that has a realistic look, similar to what can be seen in games in the industry such as Ghost of Tsushima (Fig. 2), Sekiro or any other game that could be considered "AAA".



Fig. 2 - Ghost of Tsushima: Gameplay capture

In my case, as in the two games mentioned above, I have chosen to use a style inspired by the feudal period of Japan, also looking for a mountainous location but containing artificial structures such as those that can present a small settlement.

As for the interaction of the player with my level this is reduced to a minimum, using the keys W,A,S and D the player can move around the environment.

Before starting with the development of the work as such, I want to dedicate this space to explain a little the different procedures that are intended to be applied in this project as well as the tools applied at each point of the process.

#### 3.1 SYSTEM ANALYSIS

The first thing would be to organize the different phases of the project, in each of these phases will come into play different processes, in the Scheme. 1 you can see the different phases of the project.



The first two phases of the project would correspond to what could be indicated as planning or design work, in these would be collected the references that in the future could be consulted to guide the design of the different props to be generated, also a first sketch of how the level will be designed and then check if the sketch works well to translate it to 3d through an initial blockout.

In the next phase we would start to replace the architectural elements of the blockout and begin to generate the textures and the necessary material for the terrain. At this point, some of the techniques that I want to explore start to come into play, among them the creation of modular kits, the generation of tileable textures and shader techniques to break the tiling.

- ❖ The **modular techniques** try to split big volumes into pieces that can be combined and recombined between them, this allows us to reduce the complexity of the models and give us the flexibility and variety in use. the pipeline used for this technique is at Scheme. 3.
- ❖ **Tileable textures** are really useful for covering large meshes, and I will explore different techniques for creating these textures and similar ones called trim sheets which I will go into in more detail on the next phase.

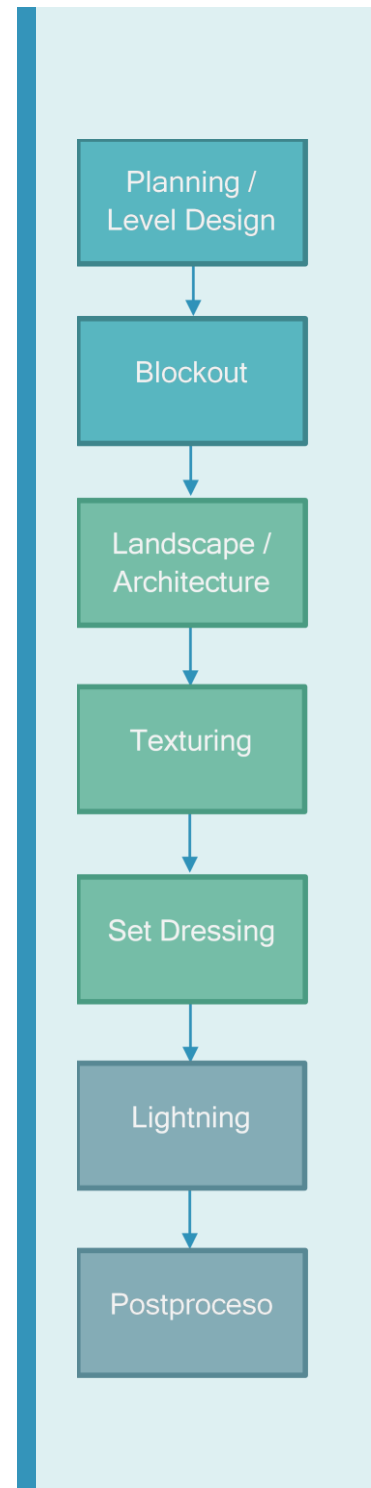
In the texturing phase I will create the missing textures for the modular kit, as well as more tileable textures.

- ❖ As I have mentioned, the seamless textures work great for texturing terrains, walls, etc. But sometimes, you need to texturize some part of a mesh with other materials, and these parts are not big enough to justify the creation of an entirely new texture. So instead of creating a new texture, it is possible to allocate some space in the same texture and still being reusable for different meshes, this technique it is called **Trim sheets**, for both tileable textures and trim sheets I will use the same working pipeline represented at Scheme. 4.

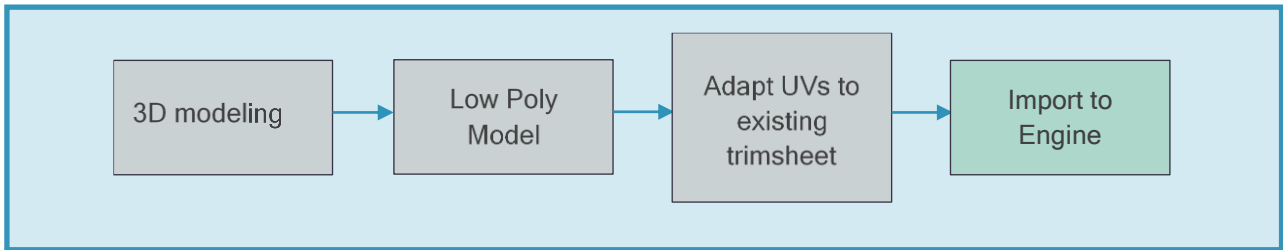
The last phase where I would continue exploring different modeling and texturing techniques would be in the set dressing section, in this section I would dedicate myself to generate props to fill the scenery. The workflow to generate the set dressing can be seen in the Scheme. 3.

The last two phases correspond to lighting and post-processing. In these sections I will not go into the tools to work on these processes and effects, but rather into the impact they can have on the performance of the scene.

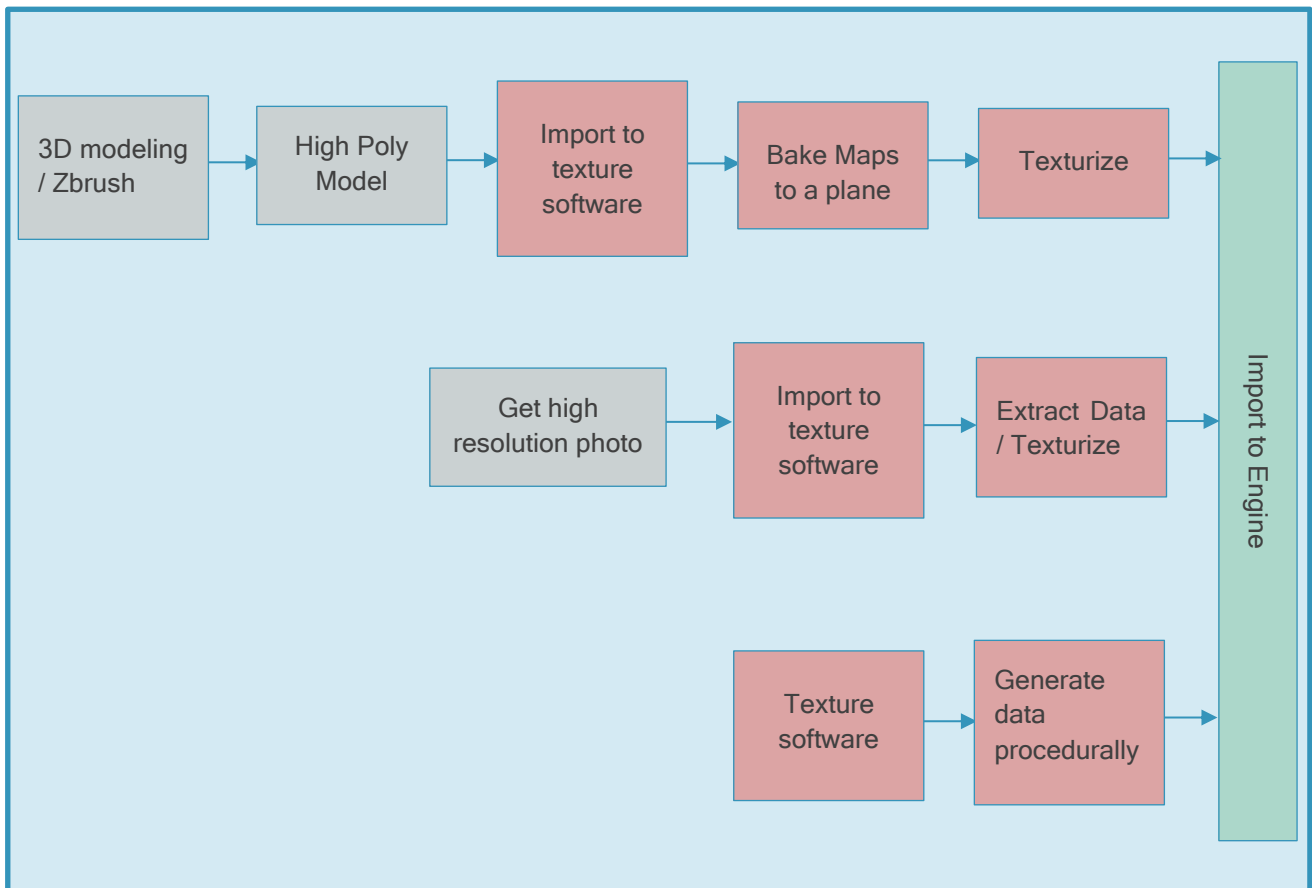
Finally, I would also like to remark that throughout the different techniques I will discuss the advantages of some of them as referred to optimization, and I will explain other possible techniques that can also be applied.



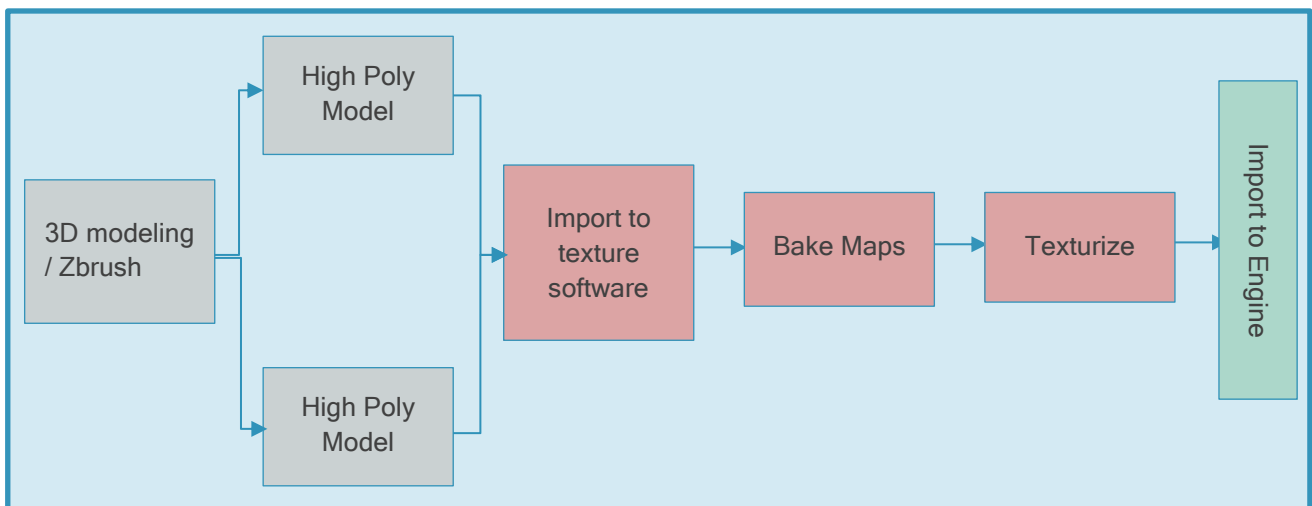
Scheme. 1 - Project phases



Scheme. 2 - Workflow to generate modular kit



Scheme. 4 - Workflows to generate tileable textures and trimsheets



Scheme. 3 - Standard Workflow to generate 3D Props

## 3.2 DESIGN

Once I had an idea of the style of scene I wanted to create and the techniques I was going to use to generate the different props and texture them, the first thing I had to do was to collect references that would facilitate my work during the following phases. This would allow me to speed up the work when creating the props, generating the textures or adjusting the lighting.

### 3.2.1 PLANNING

At this point I decided to organize the references in different documents, in each of these mood boards I put together the references looking to capture a specific feature of the scene. In total I made three mood boards:

- ❖ **Mood board Key References** - In this board, I collect the main references for the objects, the architecture, and some points of interest that can be included in the scene.
- ❖ **Mood board Key References Interior** - This collection is the same idea as the last one but oriented to the interior of the buildings.
- ❖ **Mood board Lightning** - At Fig. 3 it is possible to see the collection of references for the mood of the project scene: Lightning, fog, color palette, etc.
- ❖ **Mood board AAA Quality benchmark** - This collection lump together images of projects or games that have been developed by other studios and has a quality similar to what is expected to be achieved. It is a really useful reference to visualize the level of detail in the objects, the density in the vegetation and foliage, etc.



Fig. 3 - Mood board Lightning

### 3.2.2 TOP DOWN

After getting some references for the project, I start thinking about the layout that the level was going to have. Usually, in a big studio, this work is made by the level designers, they would take care of creating the right composition for the level depending on the objectives that the player has to accomplish and the gameplay mechanics available to the player.

All this work it is reflected in the top-down, a 2D representation. At the top down you can reflect the player's path through the level, as well as some of the main objects that the player will need.

In my particular case, I am not going to need to create my own Top-down, because I don't have to care about specific game mechanics, so I'm just creating a scene to build a composition with the props that I'm modeling.

Instead of creating a Top-down, I have used Google Earth to find a good reference for my layout, Fig. 4, Google Earth and Street View allows you to get a lot of references for the project and the top-down.



Fig. 4 - Google Earth project reference

At this point I decided the dimensions of the scene I wanted to generate. The main elements of the scene, Fig. 3, were going to be spread over a space of 200 m<sup>2</sup> but the scene as a whole, counting the boundaries of the terrain extends to 8 km<sup>2</sup> which would add the distant elements of the scene.



## 4 WORK DEVELOPMENT AND RESULTS

### 4.1 WORK DEVELOPMENT

#### 4.1.1 BLOCKOUT

Once I had gathered enough references, I decided to start the scene in the game engine and begin the blockout stage. The first thing I needed to do was to choose a terrain that would serve as a base to build all the other elements. Unreal makes use of heightmaps to generate its terrains, and provides different tools to generate them, some of them allow you to generate the height maps directly in the engine and others allow you to use height maps already generated in other programs. In my case I decided to make use of maps obtained from real world data, pages like OpenTopography give access to high resolution topographic data. Once the data was obtained I had to process them through TerreSculptor to give them a format that unreal could interpret, since by default they come in. GeoTiff format. The final map for unreal can be seen in Fig. 5.

After generating the data map, it was only necessary to introduce it in Unreal, at this point it was interesting to take into account the technical restrictions to be applied to obtain the best possible performance of the terrain. Without going into too much detail, the terrains in unreal are built from:

- ❖ **Components:** These components are squares and have the same size between each other inside a terrain, they are the minimum unit of rendering and visibility calculation.
- ❖ **Sections:** These components can be optionally divided into one or four sections that are used for LOD calculations.

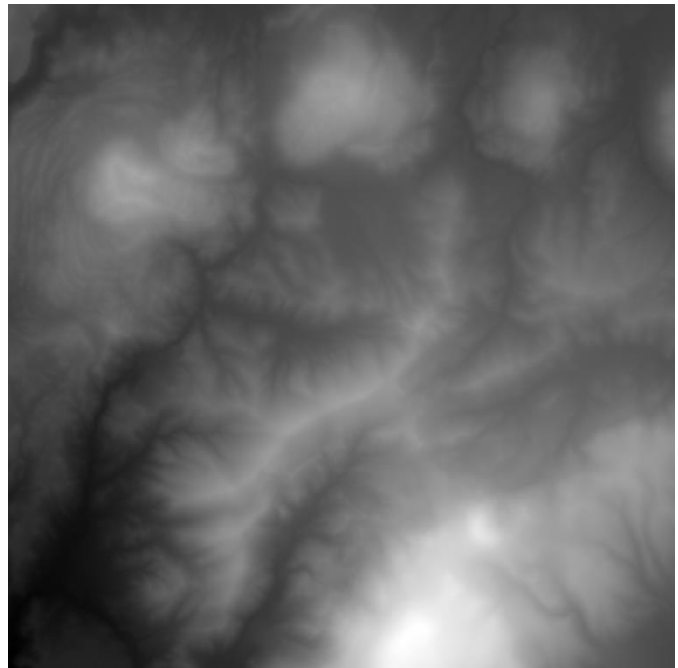


Fig. 5 - Heightmap data

The size for the components in contrast to the total components used is a matter of performance, the smaller the component the faster the LOD transition will be and will allow the occlusion of a larger part of the terrain, however, it also implies that a larger amount of components will be required and each component has a processing cost in the CPU thread, besides, each section involves a drawing call so it is important to keep it as low as possible.

In my case I have followed the recommendations that Unreal provides, and keeping a ratio of vertex per meter I have generated a terrain of 8.129 km such as:

Total Size (Vertex Resolution)	Quads per section	Section per component	Component size	Total Components
8129 x 8129	127	4	254 x 254	1024

Table 5 - Number of sizes that maximize the area while minimizing the number of Landscape components

Once the terrain was built, I began to shape what would be the large and medium volumes of the scene. This allows me to get an idea of what will be the distribution of the elements as well as their dimensions, apart from that I generated some first lighting passes following the references that I had looked for. In Fig. 6 you can see part of the results of the blockout.

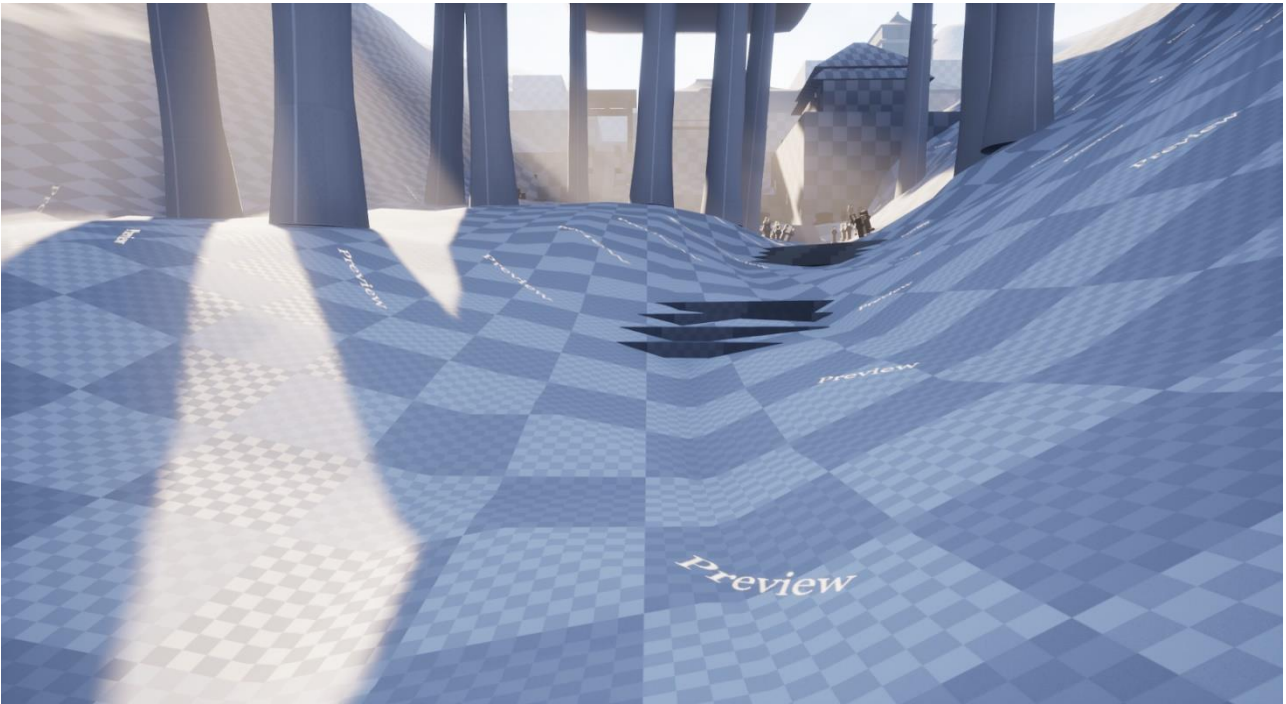


Fig. 6 - Blockout scene

#### 4.1.2 LANDSCAPE

In this section I will talk about the techniques that I have applied to be able to texture the complete terrain, as well as the techniques that I have used to break the repetitiveness, I will also introduce the optimization techniques used, some unique to the terrain and others that apply more generally to other elements as well.

In order to texture such a large element, I needed to use tileable textures, the first task was to generate these textures, for this I made use of Substance Designer that in a nodular and procedural way allowed me to generate textures with a high level of detail, in Fig. 7 and Fig. 8 you can see the node structure used to generate the heightmap of a material and the final result of the material visualized in the substance designer itself.

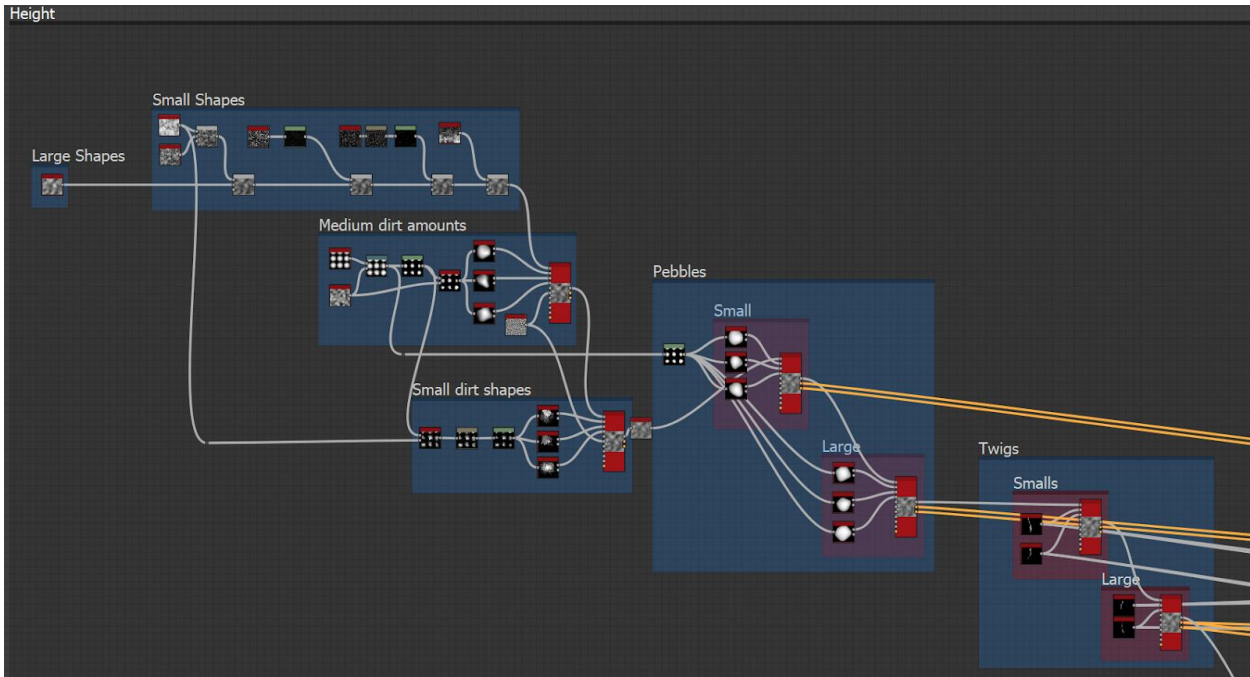


Fig. 7 - Leafy ground height map generator

In total, for the terrain I made use of three materials, two versions of the ground and a third rocky one for the stony parts of the scenery. Each of these materials requires color information, roughness, normal, ambient occlusion and height, the size of each texture is 4096 x 4096.

In my case I am generating a fairly basic terrain with only three materials, and I would be using a total of 15 textures each of them 4K, this is not optimal, especially when some of these textures are grayscale information and only requires a texture channel to be stored. A very convenient technique in these cases is the texture packing, that consists of using the different RGB+A channels of an image to store different types of grayscale information, video games like Assassin's Creed Odyssey use a texture to store color (RGB) and height (A) and another to store two channels of the normal map (RG), metalness (B) and Glossiness (A), in Fig. 9 it is possible to see how I have packed the different maps in two textures.



Fig. 8 - Leafy ground Substance Designer render

Having the materials created, the next step is to create a shader in unreal that is able to combine all the textures and apply techniques to break the tiling that can be generated.



Color - Roughness - Normal - Height - AO

RGB - A - RG - B - A

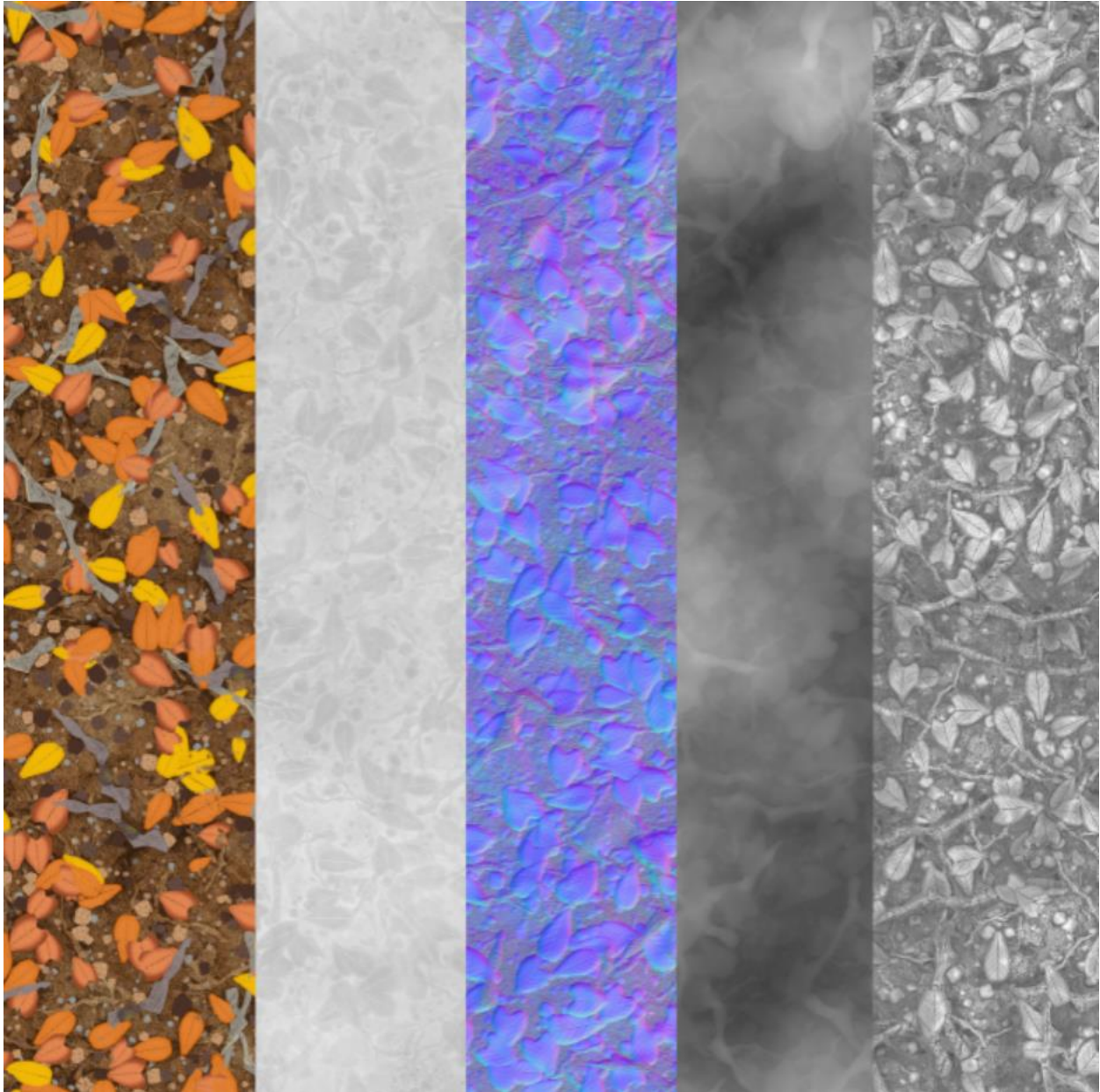


Fig. 9 - Leafy ground packed channels, in green one texture, in black the other.

## BREAKING TILING

There are different techniques that can be used to break the repetition of textures, some can be combined with each other, and there are some that may involve a higher computational cost or may involve the use of extra textures. In this section I will discuss the different techniques that I have explored, to break the tiling in the near, mid and far plane.



## NEAR PLANE

---

This first technique is very simple to implement, it basically consists of taking the texture we already have and rotate it 90 degrees, then we use some kind of mask to do a blend between the normal texture and the rotated texture. The good thing about this technique, is that it does not require any extra texture (beyond the texture used as a mask) so the shader will save some memory.

## MID PLANE

---

The first technique works well to eliminate the repetition in areas closer to the camera, the problem is that the patterns reappear immediately, to eliminate them at a greater distance we can use this next technique that has two versions, one version that offers better performance but a worse visual result, and another that consumes more memory but whose visual result improves significantly.

In this case, we will also make a blend between textures, this time the blend is not done through a noise mask but it is camera dependent, that is, we make use of the depth of the camera to generate a mask that shows us a texture near us but replaces it with another texture as it moves away from the camera. In Fig. 10 you can see the depth-dependent mask.

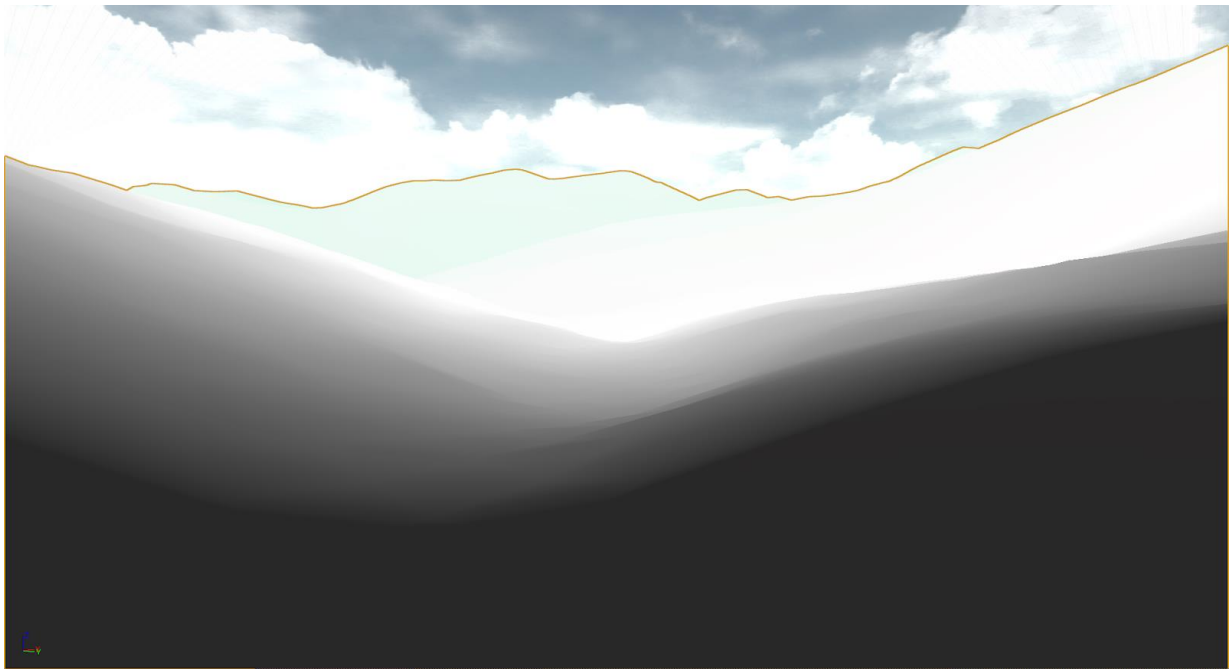


Fig. 10 - Depth camera mask

The first version of this technique consists of taking the texture that we already have but scaling it, in such a way that when the plane moves away from the camera the texture will occupy a larger space, by occupying a larger space the tiling is reduced. In Fig. 10 you can see the before and after, in this case the original texture is designed to occupy 2 m of terrain, the scaled version occupies 8 m, the problem of this first version is that by scaling the original texture the details that were designed to look good in a space of 2 m will now look very large.

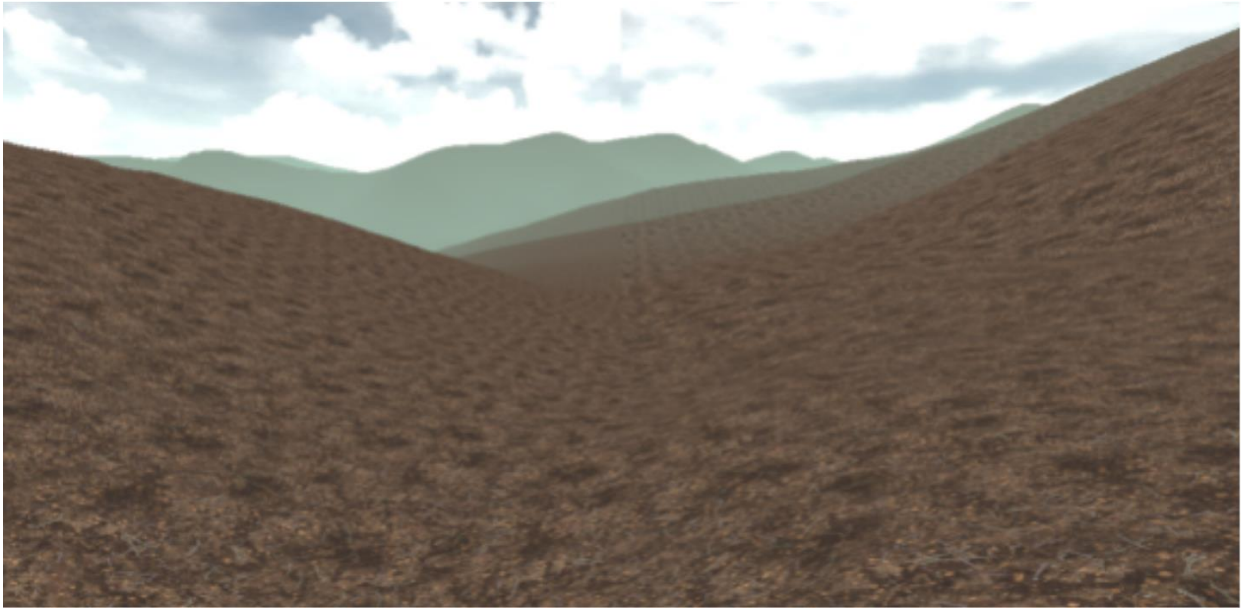


Fig. 11 - Before/After mid mask, with original texture scaled.

The second version maintains the same logic, but instead of using a scaled version of the original texture, a new texture is used as the second texture. This new texture is a version of the first one tiled and scrambled several times, in Fig. 12 you can see the before and after. As in the previous case, the original texture is designed to occupy 2 m and the new version is designed to occupy 8 m.

The advantage of this second version is that the details of the new texture will be of the same size as in the 2 m version and in addition a greater reduction of the tiling will be observed.

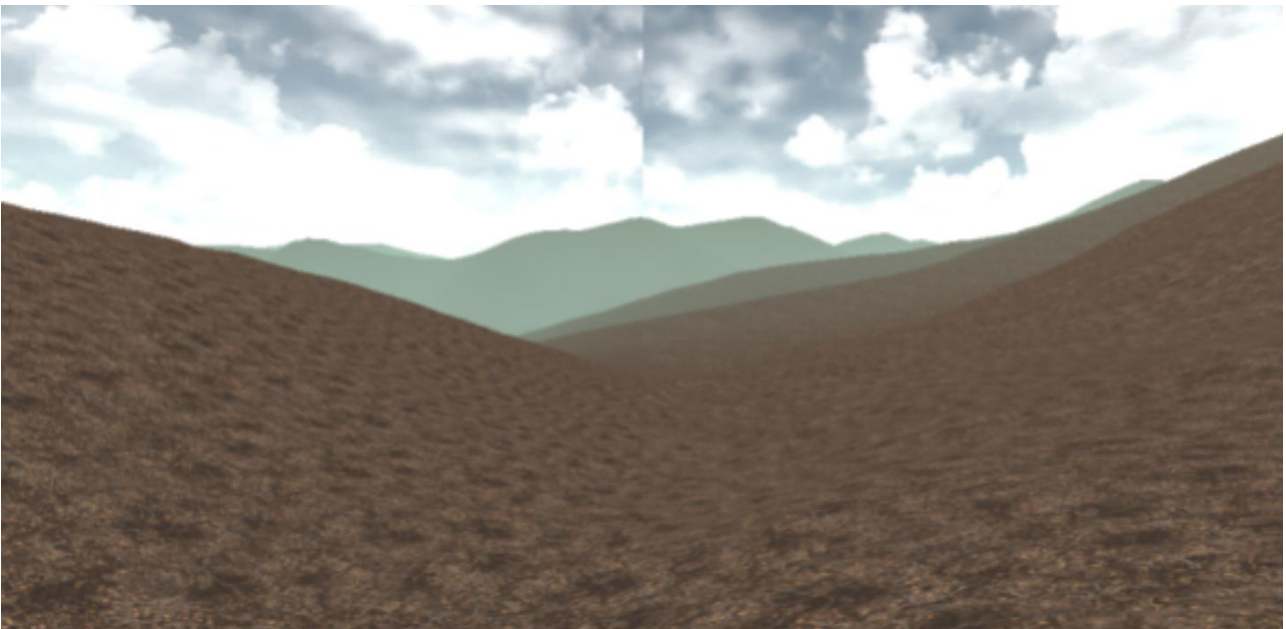


Fig. 12 - Before/After mid mask, with Secondary texture scramble

## FAR PLANE

---

This last technique is quite simple compared to all the others, it basically consists of a macro variation of the color of the terrain. You simply have to take a noise texture and add it to the base color of the material by some kind of operation, you can vary the color of the terrain undertones or you can modify the luminosity.

## SLOPE BLENDING

Once applied the techniques to break the tiling to the two versions of the ground and the texture for the cliffs, I need that somehow the shader knows where to paint each of these textures. This is known as auto material, in my case it is quite simple since I only need it to place earth texture in the flat areas and rocky texture in the cliffs, these materials can be more complex and, for example, make use of more textures or replace one by another depending on the height.

## RUNTIME VIRTUAL TEXTURE OPTIMIZATION

All the techniques I have used to build the terrain automaterial result in a very complex shader, and all the operations that this shader has to do are being calculated for each draw call. The RVT is going to take all these operations and render the result to a very large texture, this process is done once, and when you want to paint the terrain, instead of carrying out all the calculations you access the texture that is already generated.

At first I was going to use this optimization technique because it seemed quite interesting and useful, and it could be used to apply a blend effect between the set dressing objects and the terrain itself, so that a greater cohesion between all of them is achieved. However, certain aspects made me end up discarding the technique at least in this project.

On the one hand, although the shader created has a certain degree of complexity, it is quite basic, so implementing the RVT did not show a great improvement in performance. In the figures: Fig. 13 y Fig. 14 it is possible to see the information about the shader costs before and after implementation.

```

• Base pass shader: 235 instructions
• Base pass shader with Volumetric Lightmap: 315 instructions
• Base pass vertex shader: 43 instructions
• Texture samplers: 9/16
• Texture Lookups (Est.): VS(13), PS(22)

```

Fig. 13 - Automaterial shader instructions before RVT

```

• Base pass shader: 229 instructions
• Base pass shader with Volumetric Lightmap: 303 instructions
• Base pass vertex shader: 41 instructions
• Texture samplers: 2/16
• Texture Lookups (Est.): VS(0), PS(48)
• Virtual Texture Lookups (Est.): 6
• Virtual Texture Stacks: 1

```

Fig. 14 - Automaterial shader instructions after RVT

As you can see the instruction reduction is quite low, apart from that I was not able to make the tessellation work together with the RVT and the biggest problem is that the tiling system that I had implemented for the middle plane did not work, this last one is because the RVT generation is independent from the camera, so the mask shown in the Fig. 10 was not generated correctly, and finally the implementation of this technique affects a bit the quality of the textures. That is why at the end I eliminated the implementation of this technique.

### 4.1.3 VEGETATION AND FOLIAGE

In this section I am going to deal with the techniques applied to generate the vegetation, all the elements created in this section would be considered part of the set dressing, I have decided to deal with it here because the workflow is quite different from the rest of the set dressing and I think it is the logical conclusion of the point previously discussed.

To generate the vegetation that would be used in the scene, I made use of different techniques. The idea behind these techniques is the same, to obtain an atlas texture composed by the different vegetation elements, from which you can extract the models that will be applied in the scene.

The first technique I used was to generate the texture directly in substance designer, using the same processes I used for the terrain textures. Once the Atlas was generated I used 3ds max to adjust geometry cuts to the different elements of the texture, and from these pieces I built the plants. At Fig. 15 appears the implementation of the vegetation in UE4.

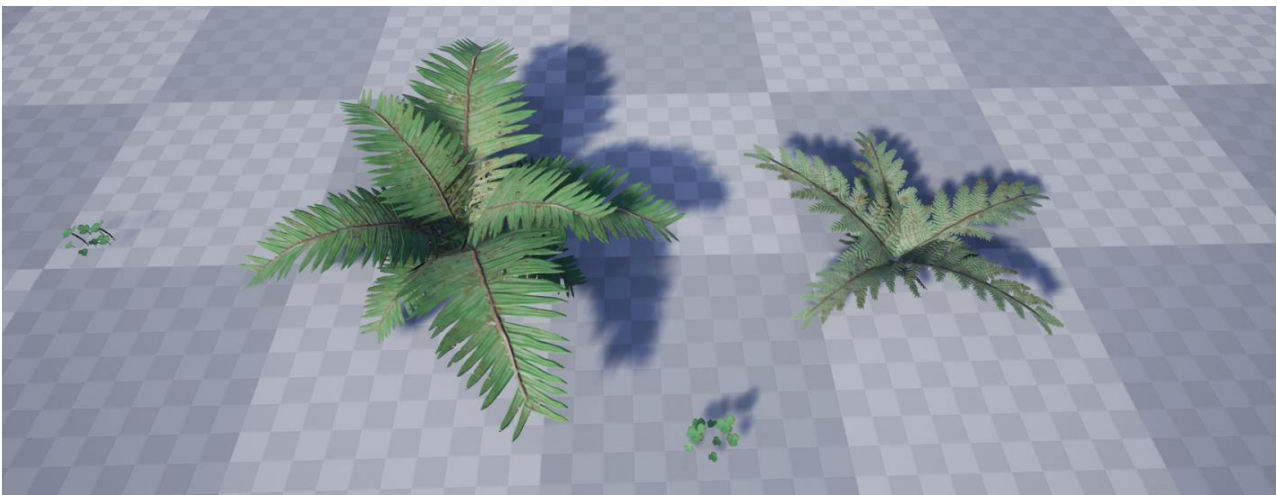


Fig. 15 - Vegetation in UE4

---

## TREES

The next technique is somewhat more complex and is the one I used to generate the trees. In this one we have to separate on one side what are the smaller branches and foliage, from what would be the trunk of the tree. The trees are composed of three different sets of textures, on the one hand we have an atlas that collects all the branches with leaves, on the other hand a set of textures that is used to store a tileable log texture that will be applied in the vast majority of parts that make up the trunk of the tree and finally a third set of adoc textures is used for the base of the trunk, in this way you can get trees that in view of the player have a greater variety.

## FOLIAGE

---

To generate the foliage of the tree, I started by creating a small initial atlas that would add color, roughness and normals information to the different elements of the branches: leaves, twigs, fruit, etc.. I generated it from cutouts extracted from photographs and retouched in Photoshop, Fig. 16, at this point it was not very relevant the resolution of the images used to generate this initial atlas, since at the end these elements would occupy a much smaller space in the final texture. With this initial atlas I generated high poly models of small sets of twigs, Fig. 17, I kept putting together different variations of these branches to obtain larger and more complex groupings. When the branches were big enough, I made a bake to generate what would be the final texture cards for the branches, Fig. 18.





Fig. 16 - Initial foliage trim,



Fig. 17 - Initial set of twigs



Fig. 18 - Final atlas of branches

## LOG

As I have already mentioned at the beginning of this section, the trunk is composed of two sets of textures, on the one hand a tileable texture is used which covers the vast majority of areas, and on the other hand a unique texture is used which adds unique characteristics to the base of the tree, in this way you can have different variations of a tree type that in essence share the vast majority of textures, but at the sight of the player they look different.

To obtain this result, I started by generating an initial low poly model, from that LP model I extracted the base and took it to zbrush, where I sculpted a high poly version (Fig. 19), applying decimation on the HP model I obtained an LP version whose geometry was much better adapted to the final shape of the trunk base after having passed the sculpting process, this new LP base I joined it with the rest of the trunk of the initial model, in Fig. 20 you can see the result.



Fig. 19 - HP log base

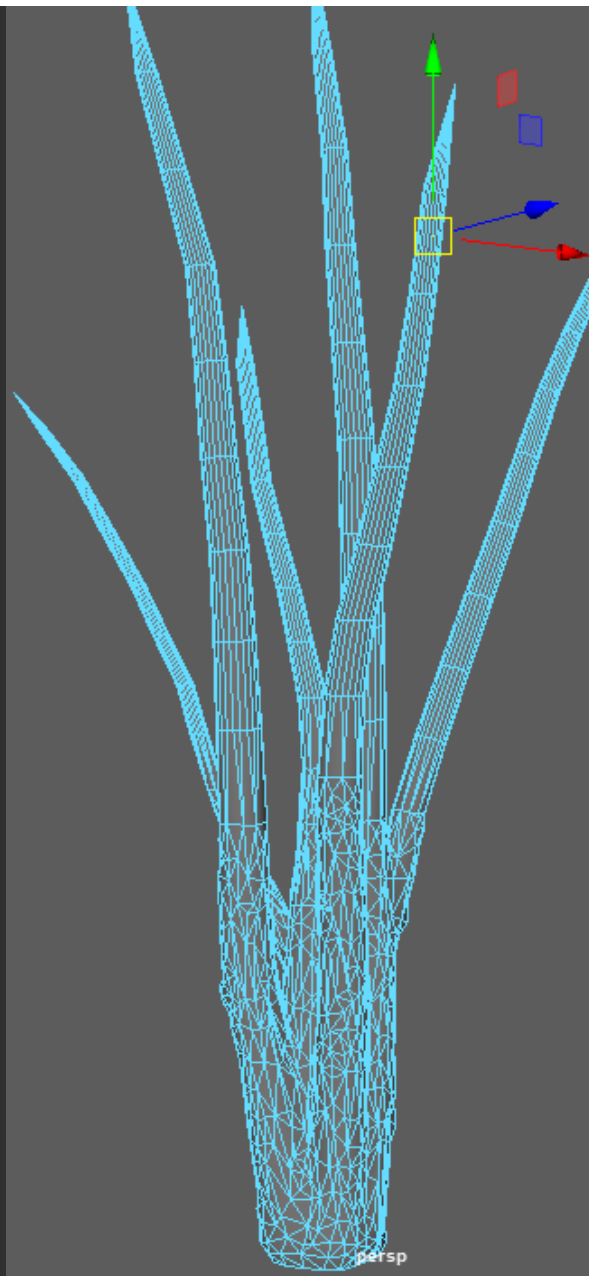


Fig. 20 - Decimated base attached to initial LP

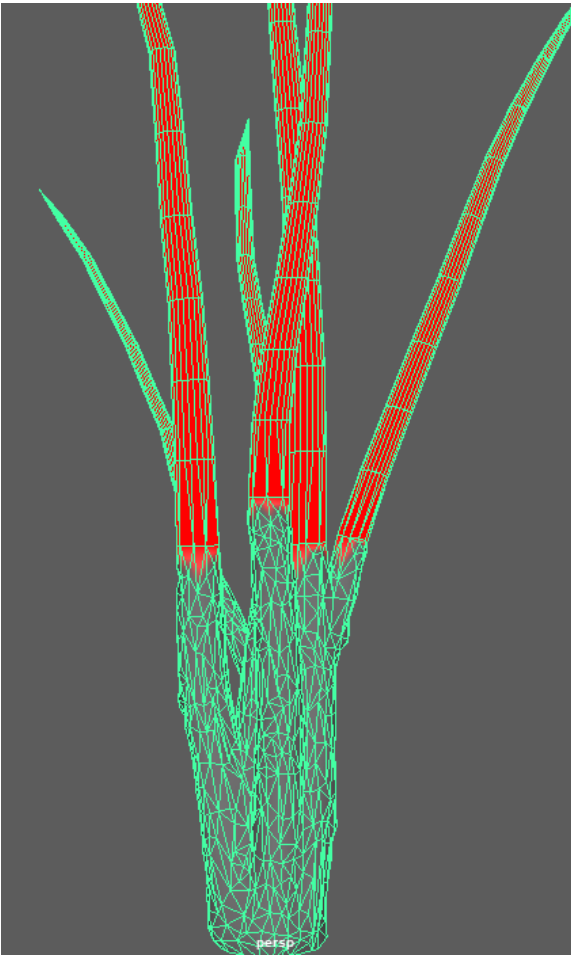


Fig. 21 - Vertex channel R used as mask

As for how to combine the adoc texture with the tileable texture and not show any border, I used a single UVs map and the vertex color property to generate a mask to make a transition from one texture to another, in Fig. 21 you can see the mask by vertex color.

---

## OPTIMIZATION

In both vegetation and trees I used a system of LODs, for bushes and vegetation I used the automatic LOD generation provided by Unreal, however for trees I generated the LODs manually, since they were going to be visible from a greater distance and required more control over the topology of each LOD, besides that, trees as last LOD make use of a Billboard that could only be generated manually, this billboard consists of a plane that is kept perpendicular to the camera view. In order to implement this optimization technique, an extra set of textures containing the complete tree information is required, Fig. 22 shows the base color texture of the billboard.



Fig. 22 - Billboard base color textures



#### 4.1.4 ARCHITECTURE / MODULARITY

Architecture is another element that requires some planning, in this section I want to talk about the concepts and techniques that I have applied. As I have already mentioned previously to generate this type of architectural structures, it is very common to make use of techniques such as modularity, to generate the modules of my scene I reviewed the references that I had collected in the moodboard of key references, after analyzing different photographs and after marking different elements that could be considered modules as well as possible sub-modules, in the Fig. 23 you can see one of the divided photographs, I built the base geometry of what would be the architectural modules.



Fig. 23 - Example of a photography divided into modules (Red and blue) and subcomponents (Green)

Having the base geometry of the modules, I introduced them in the engine and substituted the blockouts of the houses for the modules, this way I made sure that they worked and fit correctly. Once I checked that the created modules were working properly I went on to generate the textures that I would use to cover them, in the Fig. 24 you can see the modules inside unreal.

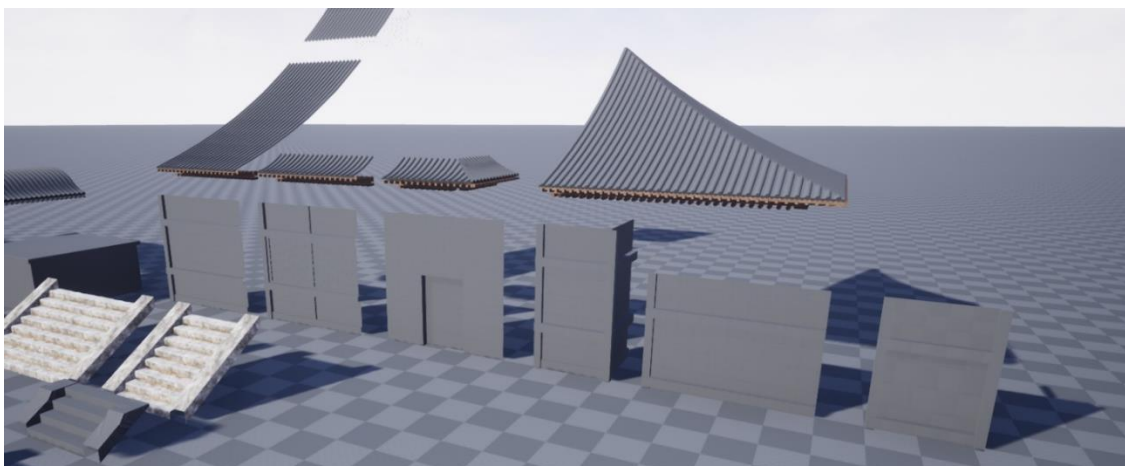


Fig. 24 - Architectural modules for buildings inside UE4

## OPTIMIZATION AND TRIM SHEETS

To develop the textures, I applied the trim sheets technique that I have already commented previously in section 4, as I have already explained with this technique what is intended is to maximize the space of the texture by combining, as its name suggests, different trims of materials that usually can be tiled in one direction, in these textures you can also add unique elements to be reused. In order to texture the modular kit of the buildings I had to generate two trim sheets and then fit the UVs of the modules into the appropriate trim. In Fig. 25 you can see the appearance of one of the generated trim sheets, tileable on the horizontal axis and with a couple of unique elements.



Fig. 25 - Base color of one of the two trim sheets

The advantage of doing it with this technique is that you can reduce the amount of textures needed to texture several elements. To generate this type of textures requires some planning and study of references when considering how much space will occupy each trimming in the texture.



#### 4.1.5 SET DRESSING

The workflow used for all the elements of this section comes to be the same, this can be seen in the Scheme. 3, Basically it starts with a base topology and very simplified that serves as part of the blockout, so that I can already get an idea of how it is inside the scene, then this geometry is passed to z brush where a HP version is generated, then, if the changes generated in z modify enough the initial form, it can be interesting to carry out a retopology of the mesh as it happened with the base of the trunk for the tree, in some cases the retopology has been manual and in others I have generated it with decimation. In the figures Fig. 26 and **¡Error! No se encuentra el origen de la referencia.** you can see the named steps for one of the props.

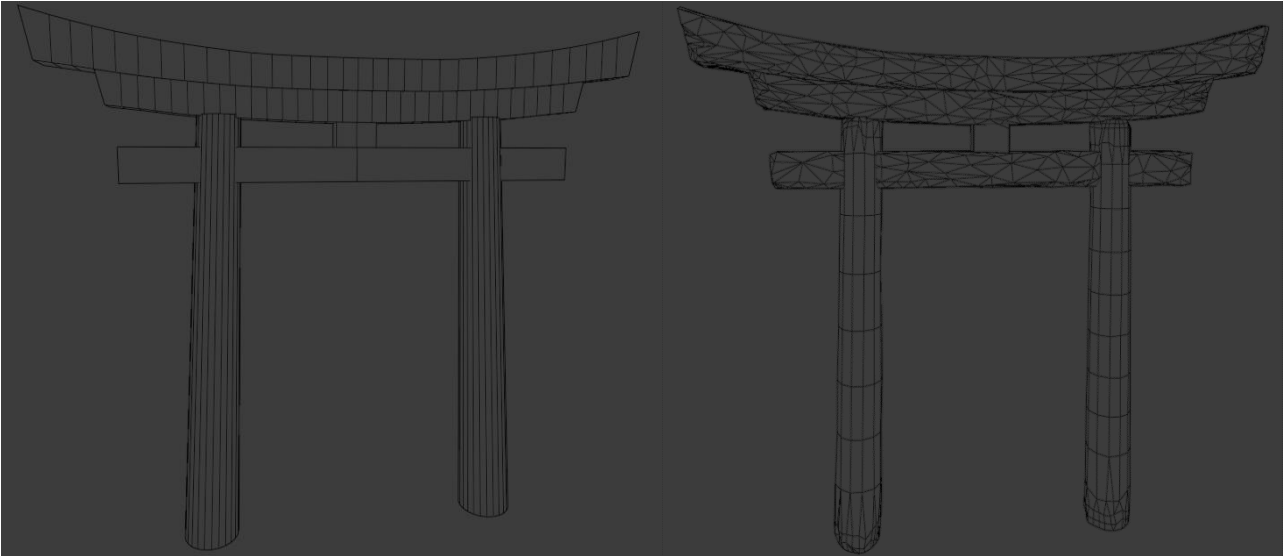


Fig. 26 - Left: Torii Gate base LP used as blockout, Right; Final Lp

When the final mesh is generated, we must prepare the UVs of the object and take it to Substance painter for texturing. All the props created, except for the Torii gate that I will describe in more depth later, make use of Adoc textures, that means that, unlike the textures seen before, the ones generated now are unique to the object, so they are not seamless.



Fig. 27 - Torii Gate HP sculpted in Zbrush

As I have already mentioned, one of the props uses a slightly different technique with respect to its texturing and uv maps. The prop in particular, is the Torii gate that can be seen in the figures above, the first thing to keep in mind is that unlike the other props of set dressing this element is quite large, being so large could not generate an adoc texture that would maintain an optimal resolution within the game, that is, if the player gets too close would be able to see the texture somewhat blurred or with artifacts. To solve this I had to use the trim sheet texture of the Fig. 25, The problem was that if I used only that texture the result would be very flat and I would not be able to take advantage of the information of the HP generated in Z. So I had to create two channels of uvs for the object, the first channel was adapted to the trimsheet, and the second was completely adoc, this way I could generate a set of unique maps that could be combined by a shader with the tileable textures, so when the player approached the object would keep the texture with a higher resolution. Fig. 28 y Fig. 31 show the two UV maps corresponding to the object.

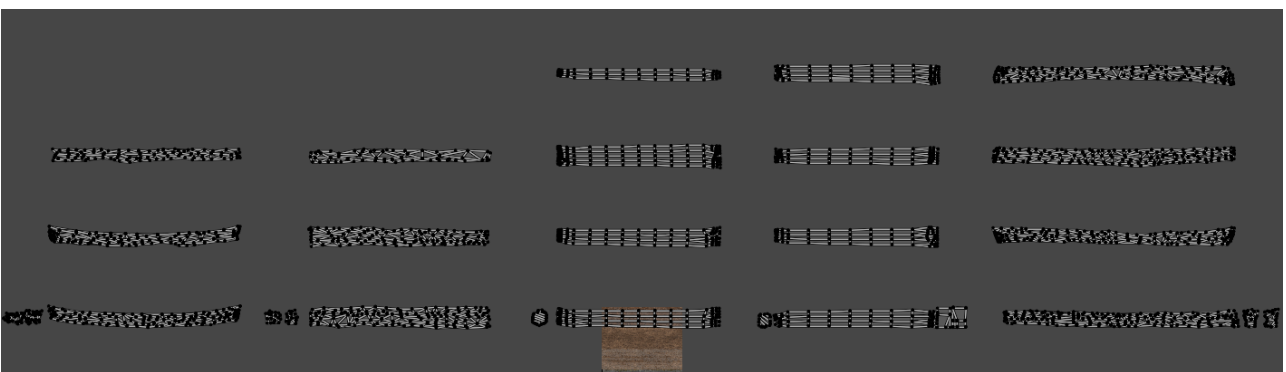


Fig. 28 - Torii Gate Uvs for Trimsheet texture

As for the information I generated using the adoc UVs, they were:

- ❖ Normal map
- ❖ Second material mask
- ❖ Moss mask
- ❖ Dust mask

The last three masks I have packed them in a single texture, while the normal ones I have left them untouched. Of the three masks that I have named before I am interested in explaining mainly the application of the first one, this basically consists of a mask generated mainly from the concavities of the object, the idea is to use it to apply in the concave areas a wood material different from the one used as a base. The base material used, is the first type of wood of the trim sheet, and I want to use the second material of the trimsheet for the concavity zones, for this, inside the shader I had to apply an offset in the uvs to adjust them to the second trim of the texture, and using the mask as a control parameter of an interpolation I have made a blend between the texture without offset and the texture with offset, the fragment of the shader can be seen in the Fig. 32.

## OPTIMIZATION

Apart from the optimization techniques such as trim sheets, or the use of LP mayas, another element that has also been used and has been discussed in other sections is the use of LODs.

Although I have not collected the optimization data as I have been implementing the different props, I thought it would be interesting to see how it affects to deactivate all the implemented detail levels and loading in the scene only the original models, and as I imagined the change is more than remarkable, in the Fig. 29 you can see how it affects the number of triangles drawn on the screen as well as the FPS and the drawing times per Frame. (Fig. 30).

Counters	Average	Max	Min	Average	Max	Min
Triangles drawn	25,015,848.00	99,101,760.00	11,762,110.00	7,110,875.00	10,007,702.00	6,959,510.00
DrawPrimitive calls	1,877.03	2,394.00	1,710.00	1,706.27	1,839.00	1,698.00
Lines drawn						

Fig. 29 - Triangles Drawn LODs Off / LODs On

	12.45 FPS	60.00 FPS
	80.30 ms	16.67 ms
Frame:	79.08 ms	16.67 ms
Game:	7.49 ms	3.01 ms
Draw:	35.35 ms	6.20 ms
GPU:	79.21 ms	16.63 ms
DynRes:	Unsupported	Unsupported

Fig. 30 - FPS and Draw milliseconds LODs Off / LODs On

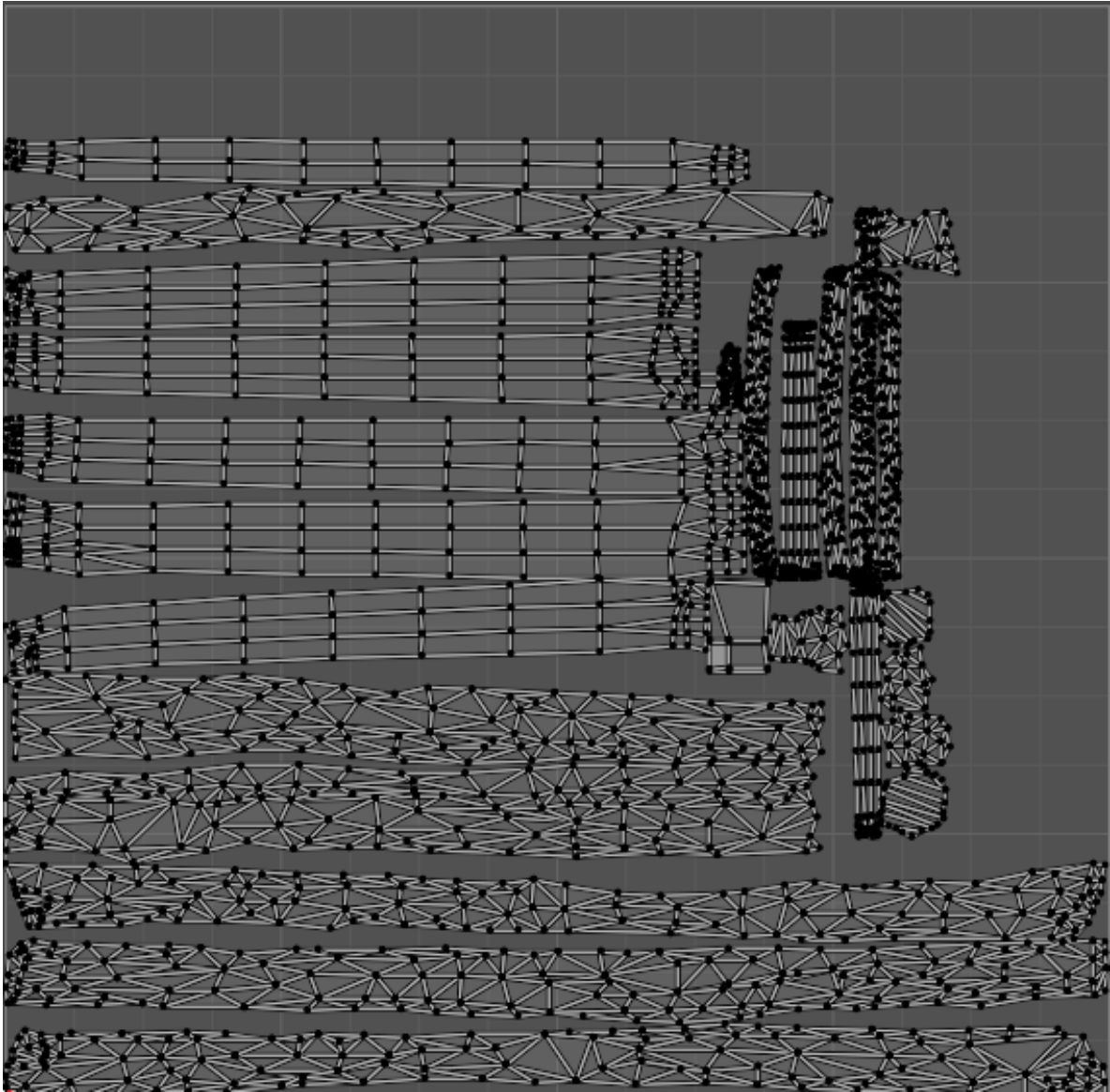


Fig. 31 - Torii Gate Adoc Uvs

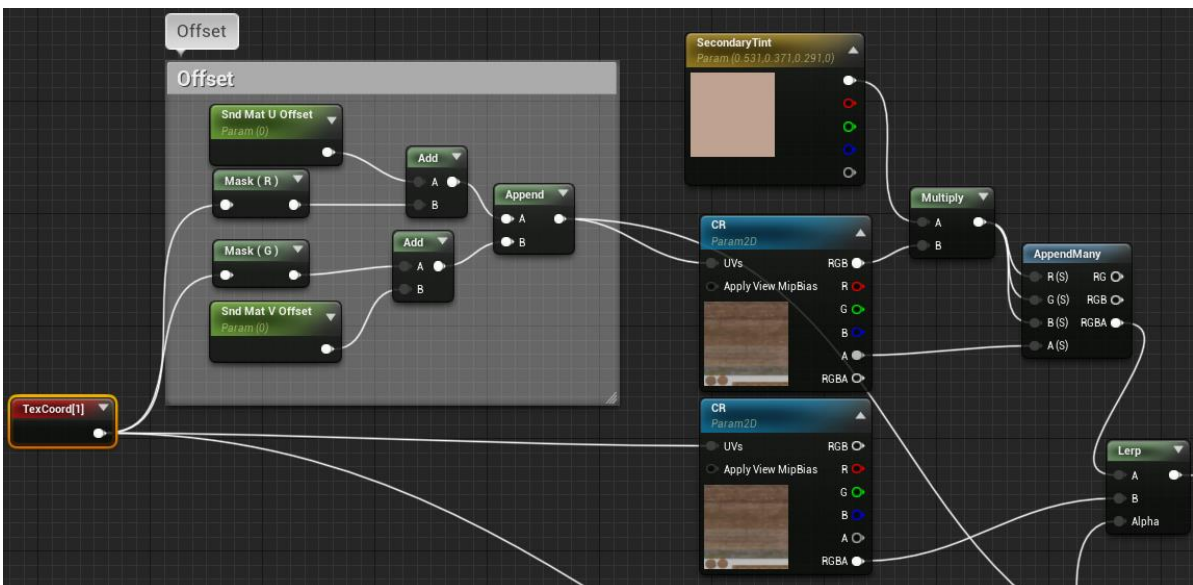


Fig. 32 - Torii gate shader: Interpolation between materials using offset

#### 4.1.6 LIGHTNING AND POST-PROCESSING

This section is going to be quite short; I am only going to talk about some of the techniques used to generate post-processing effects, and the impact they can have on performance.

It is very common in video games to make use of techniques that allow to adjust or modify some visual aspects of the render before drawing it on screen, you could say that they work as a kind of filters. Some of these tools can be used to vary the tonalities of the scene, increase or decrease contrasts, add effects like depth of field, etc.

These effects can greatly improve the result presented to the user, but they can also have a negative impact on the performance of the game and imply a higher RAM consumption.

In my scene I have decided to apply the following post-processing effects:

- ❖ Bloom
- ❖ Exposure
- ❖ Chromatic Aberration
- ❖ Lens Flares
- ❖ Vignette Effect
- ❖ Color Grading
- ❖ Depth of field
- ❖ Ambient Oclusión
- ❖ Screen space reflections

Apart from post processing, other effects that can also have a negative impact on performance are volumetric effects such as those generated by volumetric fog.

In the following figure you can see the impact that these effects have on processing times, the values shown in the figures are the milliseconds it takes the GPU to draw each frame.

GPU [STATGROUP_gpu]	Average	Max	Min	GPU [STATGROUP_gpu]	Average	Max	Min
Counters				Counters			
[TOTAL]	17.17	22.40	14.73	[TOTAL]	16.48	20.68	12.76
Shadow Depths	4.13	5.54	3.73	Shadow Depths	4.17	4.62	4.03
Basepass	3.08	4.70	2.50	Basepass	2.64	6.69	2.22
Prepass	1.76	3.02	1.56	VisibilityCommands	3.79	5.76	0.03
VolumetricFog	1.04	1.79	0.89	Prepass	1.23	1.86	1.18
DepthOfField	0.88	1.82	0.78	VolumetricFog			
VisibilityCommands	0.81	5.85	0.00	Postprocessing	0.52	0.68	0.49
Postprocessing	0.53	0.99	0.46	DepthOfField			
ScreenSpace AO	0.62	1.85	0.44	ScreenSpace AO	0.64	1.45	0.47
[unaccounted]	0.55	1.40	0.47	[unaccounted]	0.20	0.94	0.15
Lights	0.53	2.31	0.37	SkyLightDiffuse	0.64	1.83	0.88
SkyLightDiffuse	0.52	1.94	0.36	Lights	0.62	2.31	0.27
ScreenSpace Reflections	0.41	1.26	0.34	State UI	0.33	0.85	0.10
Reflection Environment	0.28	1.09	0.23	VolumetricCloud	0.17	0.21	0.15
TAA	0.40	1.28	0.33	Reflection Environment	0.33	1.20	0.27
VolumetricCloud	0.37	0.42	0.30	TAA	0.20	0.71	0.18
State UI	0.28	1.65	0.10	ScreenSpace Reflections	0.09	0.23	0.08
Shadow Projection	0.25	1.01	0.20	GPUSceneUpdate	0.01	0.03	0.01
Translucent Lighting	0.19	0.86	0.16	Shadow Projection	0.37	1.45	0.20
HZB	0.10	0.13	0.09	Translucent Lighting	0.20	1.03	0.16
Fog	0.09	0.10	0.09	HZB	0.12	0.64	0.11
LensFlare	0.08	0.67	0.05	Fog	0.09	0.10	0.08
GPUSceneUpdate	0.02	0.25	0.01	SkyAtmosphereLUTs	0.05	0.07	0.05
SkyAtmosphereLUTs	0.07	0.94	0.05	LensFlare			
ScreenSpace AO Setup	0.05	0.08	0.05	ScreenSpace AO Setup	0.05	0.07	0.05
17 more stats. Use the stats.MaxPerGr				17 more stats. Use the stats.MaxPer			

Fig. 33 - Left effects On / Left effects Off

Although at first glance it may seem that there is not much variation, it must be considered that the computer being used to carry out the development of the work is quite powerful, if the hardware on which the project is being executed were inferior, this change would be more pronounced.



## 4.2 RESULTS

### 4.2.1 TIME BALANCE AND DEVIATIONS

At the beginning of the project, I planned the tasks by weeks of work, so that in periods of one or two weeks I was advancing through the different phases of the project, so that at the end of the project I would have made a couple of iterations in the field the vegetation and props, so that in the second iteration I could polish or fix mistakes that I had left.

It is remarkable to note that the work developed in this project was carried out in parallel to the study of a master, in the first tasks of the project I had no major problem, and it was easy to follow the planned scheme, and in general, the work times were quite adjusted or improved the estimated ones. As the course progressed the volume of work required by the master increased more than I had initially estimated, and there were times when the project was stuck, besides, I was unable to work for a week because I was sick as a result of covid-19, with minor symptoms but enough to be unable to work in front of the computer, this also increased the accumulated workload of the master.

In short, the initial planning that I had for weeks was completely broken down and I had to take every spare moment that my master's studies and vacations between academic terms left me in order to move the project forward.

Task	Initial Hours	Final Hours	Deviation Reason
Reference gathering	8	8	-
Technical Proposal	4	2	Were easier than I expected
Level Top-Down	8	6	I ended up using an image extracted from Google earth instead of designing it myself.
Analysis and design document	8	8	-
Generate Terrain from Real-World Data	4	2	Were easier than I expected
Blockout	14	10	I completed it faster than I had initially estimated.
Create Terrain Material	40	40	-
Create Vegetation and Trees	40	48	I ended up using a more advanced technique than I had intended.
Create Architecture	40	40	-
Set Dressing	46	20	I had to reduce time in order to complete other chapters
Particles	18	16	Were easier than I expected
Lightning	20	20	-
Memory	40	60	It has taken more time than I originally thought it would
Presentation	10	16	It has taken more time than I originally thought it would
<b>Total Hours</b>	<b>300</b>	<b>300</b>	

---

### 4.2.2 ACHIEVING THE OBJECTIVES

Taking as a reference the objectives mentioned in the introduction section 1.3, I can say that the objectives initially proposed have been completed.

- ❖ I have been able to apply techniques using shaders to texture large surfaces and break the tiling of the materials using masks generated from textures, Vertex color and with camera dependency. At the beginning I thought that for this work I would use mainly vertex painting, but finally the techniques explained in section 4.1.4 have ended up having more relevance. In general all the shader techniques used to break the tiling have given a good result.
- ❖ I have built a modular kit that I have been able to use to generate different architectural constructions and this kit has been textured using trim sheets. At first, I had planned to create the trim sheets from the information extracted from HP models that I would have worked on in Zbrush and textured in Painter, but finally I ended up extracting the information maps from high resolution photos using Substance Alchemist and combining it with an extra pass-through Substance Painter to add more detail. In doing so, I have been able to reduce the work time and the quality of the generated textures has been maintained.
- ❖ I was able to create set dressing elements using the workflow described in the system analysis section. During the development of the work I have ended up using techniques that combined the use of adoc and tillable textures in the same element, as is the case of the trees or the Torii Gates, these techniques were not planned at first and have taken me a little more time but they have been interesting and useful.

I would like to say that I have not been able to implement one of the tools that I had originally thought of using, which are the decals, it is one of the tools used to break the repetition, I have not been able to implement it, mainly due to lack of time. It would have been interesting to be able to apply some of them.

---

### 4.2.3 LINKS

- ❖ GitHub repository: <https://github.com/Macadrian/TFG.git>

## 5 CONCLUSIONS AND FUTURE WORK

### 5.1 CONCLUSIONS

The first thing I must say is that in general terms, I am satisfied with the development of this project and with everything I have learned, although I must say that I have a bittersweet feeling with the final result obtained.

On the one hand, regarding the development of the work as well as the research and application of the different techniques explored, I feel quite pleased. I have had the opportunity to investigate and use completely new techniques for me, it has been very interesting to generate shaders of certain complexity as in the case of the one used for the landscape, or to explore advanced workflows as in the case of the one applied to generate the trees, I have also had the opportunity to use tools that I had not used before such as Substance Designer.

On the other hand, although I have been able to meet the main objectives that I had proposed in the work, I think that I have not been able to meet the personal objectives that I had set myself with this project. Due to the fact that I have carried out the development of this work in parallel to my master studies, I have not been able to dedicate as much time to it as I would have liked, and I have had to adapt the roadmap of this project to the free moments that the master left me. I would have liked to have obtained a much more polished final result and to have been able to generate a scene with a greater density of elements.

Concluding this section and as a summary, I am quite happy with what I have learned and with the new techniques I have been able to apply, but I would have liked to have more time to obtain a more polished and professional final result.

### 5.2 FUTURE WORK

I think this work has accomplished what was intended, which was to explore different texturing and modeling techniques that are applied in the video game industry, therefore, there would not be much work left to do on this project and I would consider it finished.

In any case, in the short term, a post mortem document should be developed to reflect what has been learned during the process, as well as what can be changed to improve it. This document should include the mistakes that have been made and the problems that have been detected, analyze the causes of the problem as well as the damage caused and the impact it has had, and emphasize the actions that can be taken which contribute to overcoming the error.

Going into the medium to long term future, it would be best to develop new projects that seek to improve and refine both the techniques learned and the results obtained, and to apply the lessons learned and reflected in the post mortem document.

## BIBLIOGRAPHY

- [1] S. Kumar, "Medium," 30 5 2020. [Online]. Available: <https://medium.com/@legacykraft/trim-sheet-detailed-breakdown-bd35c78d16c3>. [Accessed 20 6 2021].
- [2] V. Zhdanov, "80 Level," 20 12 2017. [Online]. Available: <https://80.lv/articles/understanding-modular-environment-design/>. [Accessed 15 1 2021].
- [3] B. Cloward, "Youtube," 8 10 2020. [Online]. Available: [https://www.youtube.com/watch?v=qSU0uDr2wt4&list=PL78XDi0TS4IHwqv\\_PmXAdedT2SH1qeRqK](https://www.youtube.com/watch?v=qSU0uDr2wt4&list=PL78XDi0TS4IHwqv_PmXAdedT2SH1qeRqK). [Accessed 18 1 2021].
- [4] T. Simpson, "Youtube," 17 10 2018. [Online]. Available: <https://www.youtube.com/watch?v=fJps8QxOh4s&list=PLBi3xvwwY3dncU0vIOi7gov3AFmXhRTOm&index=1>. [Accessed 11 01 2021].
- [5] T. Anlauf, "80 Level," 18 12 2018. [Online]. Available: <https://80.lv/articles/001agt-004adk-17th-century-roman-modular-environment-in-ue4/>. [Accessed 16 1 2021].
- [6] B. Cloward, "Youtube," 21 01 2021. [Online]. Available: <https://www.youtube.com/watch?v=RjwyFIUo5p8>. [Accessed 21 01 2021].
- [7] P. Haslow, "Unreal Engine," [Online]. Available: <https://learn.unrealengine.com/course/3754335/module/7274128?moduletoken=UHxxnDLPW8TyzZqVFK40~5jSUBVF-h6q6GhguMDmgdUzIAobNIUn2s9Lwhr5Pcgo&LPId=0>. [Accessed 13 1 2021].
- [8] EpicGames, "Unreal Engine," [Online]. Available: <https://docs.unrealengine.com/4.26/en-US/BuildingWorlds/Landscape/TechnicalGuide/>. [Accessed 18 5 2021].
- [9] B. Cober, "Casual Distraction Games," 7 01 2017. [Online]. Available: <https://www.casualdistractiongames.com/post/2017/01/07/ue4-overview-of-static-mesh-optimization-options>. [Accessed 9 3 2021].
- [10] V. Deroizer and P. Fleau, "GDC Vault," 18 03 2019. [Online]. Available: <https://www.gdcvault.com/play/1026145/Texturing-the-World-of-Assassin>. [Accessed 9 3 2021].
- [11] A. Galuzin, "World of Level Design," 27 1 2015. [Online]. Available: <https://www.worldofleveldesign.com/categories/ue4/ue4-guide-to-scale-dimensions.php>. [Accessed 6 1 2021].
- [12] A. Galuzin, "World of Level Design," 30 1 2020. [Online]. Available: <https://www.worldofleveldesign.com/>. [Accessed 4 3 2021].

- [1 V. Loignon, "Vincent Loignon," 4 01 2016. [Online]. Available:  
3] <http://vincentloignon.com/blog/optimizing-and-profiling-games-with-unreal-engine-4/>.  
[Accessed 20 3 2021].
- [1 G. Sanders, "GDC Vault," 2018. [Online]. Available: [https://ubm-twideo01.s3.amazonaws.com/o1/vault/gdc2018/presentations/gilbert\\_sanders\\_between\\_tech\\_and.pdf](https://ubm-twideo01.s3.amazonaws.com/o1/vault/gdc2018/presentations/gilbert_sanders_between_tech_and.pdf). [Accessed 4 4 2021].
- [1 R. Yang, "The Level Design Book," [Online]. Available:  
5] <https://book.leveldesignbook.com/process/blockout/metrics/modular>. [Accessed 6 1 2021].
- [1 "Substance 3D," 15 1 2019. [Online]. Available:  
6] <https://substance3d.adobe.com/tutorials/courses/Getting-Started-with-Substance-3D-Designer/youtube-y8q6-tgQjZc>. [Accessed 17 3 2021].
- [1 A. Galuzin, "World of Level Design," 29 4 2020. [Online]. Available:  
7] <https://www.worldofleveldesign.com/categories/ue4/landscape-heightmap-guide.php>.  
[Accessed 9 1 2021].

## LIST OF FIGURES

Fig. 1 - Gantt Chart: Tasks Initial hours distribution .....	13
Fig. 2 - Ghost of Tsushima: Gameplay capture .....	16
Fig. 3 - Mood board Lightning .....	19
Fig. 4 - Google Earth project reference .....	20
Fig. 5 - Heightmap data .....	21
Fig. 6 - Blockout scene .....	22
Fig. 7 - Leafy ground height map generator .....	23
Fig. 8 - Leafy ground Substance Designer render .....	23
Fig. 9 - Leafy ground packed channels, in green one texture, in black the other. ....	24
Fig. 10 - Depth camera mask.....	25
Fig. 11 - Before/After mid mask, with original texture scaled .....	26
Fig. 12 - Before/After mid mask, with Secondary texture scramble .....	26
Fig. 13 - Automaterial shader instructions before RVT .....	27
Fig. 14 - Automaterial shader instructions after RVT .....	27
Fig. 15 - Vegetation in UE4 .....	28
Fig. 16 - Initial foliage trim, .....	29
Fig. 17 - Initial set of twigs .....	29
Fig. 18 - Final atlas of branches .....	29
Fig. 19 - HP log base .....	30
Fig. 20 - Decimated base attached to intial LP .....	30
Fig. 21 - Vertex channel R used as mask.....	31
Fig. 22 - Billboard base color textures.....	32
Fig. 23 - Example of a photography divided into modules (Red and blue) and subcomponents (Green).....	33
Fig. 24 - Architectural modules for buildings inside UE4.....	33
Fig. 25 - Base color of one of the two trim sheets.....	34
Fig. 26 - Left: Torii Gate base LP used as blockout, Right; Final Lp .....	35
Fig. 27 - Torii Gate HP sculpted in Zbrush .....	36
Fig. 28 - Torii Gate Uvs for Trimsheet texture .....	36
Fig. 29 - Triangles Drawn LODs Off / LODs On.....	38
Fig. 30 - FPS and Draw milliseconds LODs Off / LODs On .....	38
Fig. 31 - Torii Gate Adoc Uvs .....	39
Fig. 32 - Torii gate shader: Interpolation between materials using offset .....	39
Fig. 33 - Left effects On / Left effects Off.....	40



## LIST OF TABLES

Table 1 - Subject areas related to the project.....	9
Table 2 - Initial planning.....	12
Table 3 - Laptop specifications .....	14
Table 4 - Total project cost in €.....	15
Table 5 - Number of sizes that maximize the area while minimizing the number of Landscape components.....	21
Table 6 - Time Deviations in Tasks .....	42

## LIST OF SCHEMES

Scheme. 1 - Project phases .....	17
Scheme. 2 - Workflow to generate modular kit.....	18
Scheme. 4 - Standard Workflow to generate 3D Props .....	18
Scheme. 3 - Workflows to generate tileable textures and trimsheets.....	18