3D modeling and game generation for the UJI Smart Campus

Marcos Álvarez Lázaro

Final Degree Work
Bachelor's Degree in
Video Game Design and Development
Universitat Jaume I

Academic year 2020 - 2021

Supervised by: Michael Gould Carlson, Full Professor

Acknowledgments

First of all, I would like to thank my Final Degree Supervisor Michael Gould Carlson for his patience and guiding me through this project.

I also want to thank Juan Camilo Gómez for supporting me with all my doubts and giving me material to learn this program.

Finally I would like to dedicate this work to my family for always supporting me, to my grandparents for being able to bring out the best in me despite the distance, to my friends for trusting me and helping me to finish this project.

Abstract

This document presents the process, the obstacles and the final result of the creation of the Medical Faculty and how I give it my own version, making a little comparison between both of them to be able to see a convenient result using shaders and a little of modeling, using AutoCAD 2020, CityEngine, Blender, Substance Designer and Unreal Engine 4.

In this project it can be seen how to implement buildings built with the City Engine program in Unreal Engine 4, a small comparison with a remodelling that will make it possible to approach the building in a completely different way, giving a futuristic air where nature takes over the building.

The process of creating City Engine and how to work in the programme will be reflected and then leave everything ready to be correctly implemented in the SmartCampus model of the Uji, in which the 3D representation of all the buildings of the university created in City Engine can be found.

Contents

Figure index

1

INTRODUCTION

Contents

1.1 Work Motivation

When I was a child I loved architecture, I thought several times about trying to do architecture as a career, but I soon discovered that this was not the path I wanted to follow. Even so, that ability to admire the buildings around me and learn about them has stayed with me until now.

This project was a bit of a rescue of that feeling, combined with one of my passions, 3D modelling. So it seemed like the perfect opportunity to learn how to use a new software while creating one of the university's buildings and finally trying to take it to my field. Although it wasn't an original proposal, I think it really connected with my interests and concerns, so it generated a lot of curiosity and desire to explore the tool and learn.

1.2 Objectives

- Create a model of the new medicine building in order to modernise the UJI Smart Campus application.
- Use the ArcGIS portal for the geolocation process of the building.
- Learning to work with CityEngine and its procedural techniques.

- Implementation of the faculty model in game engines.

- Creating a scene in a game engine with a personal touch.

1.3 Environment and initial state

This project was started without any prior knowledge of CityEngine, ArcGIS and UnReal Engine 4. The first weeks were spent getting used to the software and following tutorials. In addition to analysing all the data and documents provided.

2

PLANNING AND RESOURCES EVALUATION

Contents

2.1 Task and subtasks

- Get familiar with the software to use, which is CityEngine
- Start creating the medicine building
- Finish the building and export it in the necessary formats
- Implement the building in a game engine with ArcGIS
- Create my own version of the building
- Make an animation with a post processing to see the results

2.2 Time planning

Planning:

- Learning new technologies 10%
- Data cleaning and processing 20%

- Modelling/programming 60%
- Writing the final report 10%

2.3 Tools

- City Engine
- ArcGIS
- Unreal Engine 4
- Substance Designer
- AutoCAD
- Blender
- Krita

2.4 Expected results

The modernization of the UJI Smart Campus through new models. Also the implementation of these models in some of the main video game engines. Finally, give a personal style and another point of view to the campus in a procedural way.

3

BUILDING CREATION

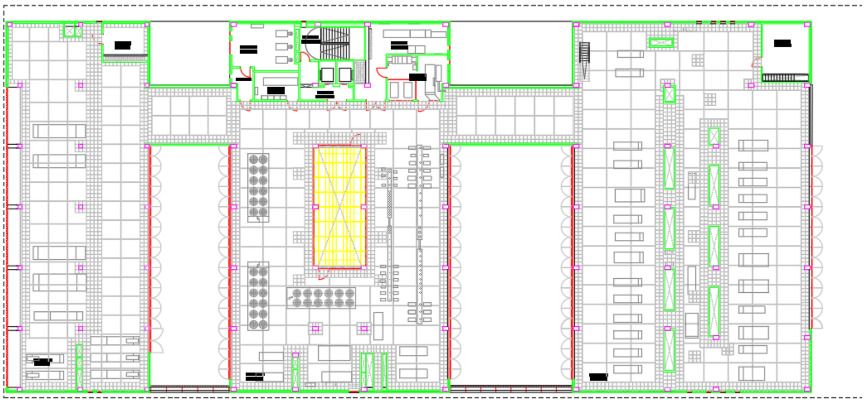
Contents

This chapter explains the process of creating the building step by step, with its problems and how they have been solved. This part has been perhaps the most complicated because the building is not yet fully constructed.

3.1 Autocad

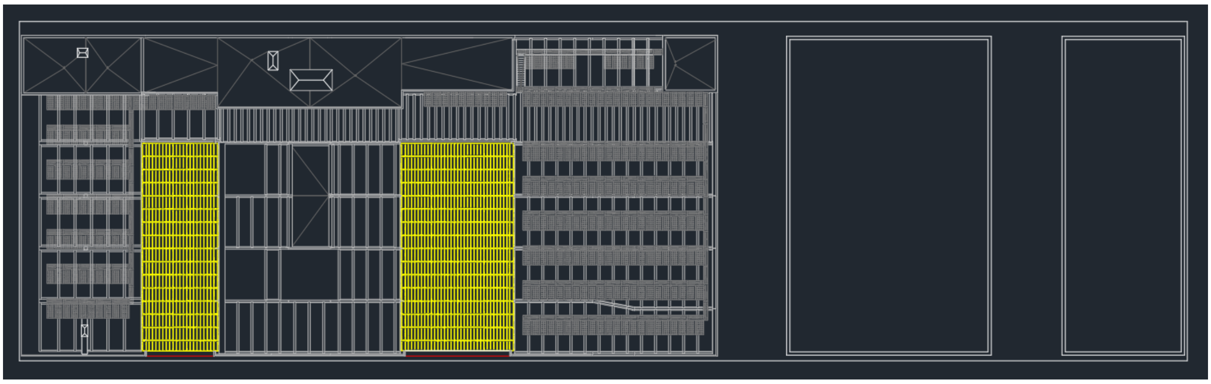
The first available information was AutoCAD floor-by-floor plans of the current building, which were useful in determining the base form of three of the five modules in the building. As they were only plans of the top view, it was somewhat complicated to determine the height of the building, as this information was not available. This part will be explained later on how it was solved.

Figure 3.1: Top view of the building (AutoCAD)



This first image (see Figure 3.1) is the one used as a reference to be able to appreciate the shape and dimensions of the modules.

Figure 3.2: Top view of the 5 modules (AutoCAD)



As this image shows (see Figure 3.2), you can see how the building is incomplete and two modules are missing. It is from this image that the structure of the remaining two modules can be seen, allowing the building to be recreated in its entirety.

3.2 Using CityEngine

When it came to working with this programme, not having the necessary knowledge of its use, it was necessary to start with the basics and how to move properly through the interface, learning the rules of use and how to edit objects and the rules themselves.

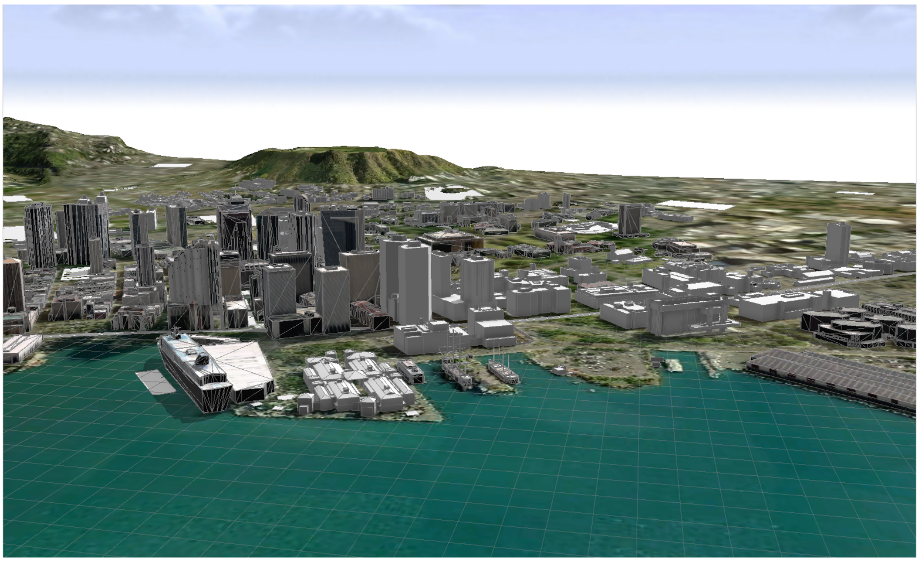
To achieve this, the first step was to build the city of Honolulu thanks to a tutorial, which taught us how to apply some of the basic rules and how to import files into the program.

The first thing this tutorial taught us is how to create a terrain, with its volume and its real parameters, as we can see in the image (Figure 3.3), in which we can see relief in the horizon and in the city itself. In this image you can also see the basic shapes of the buildings. This will be the next step, where we get to include in the terrain the buildings of Honolulu, with its basic shape.

In order to make sense of the city and allow us to recognise the important areas so that we can work on them better, it is necessary to separate the areas according to their utility, such as parks, ports, roads, railways, buildings...

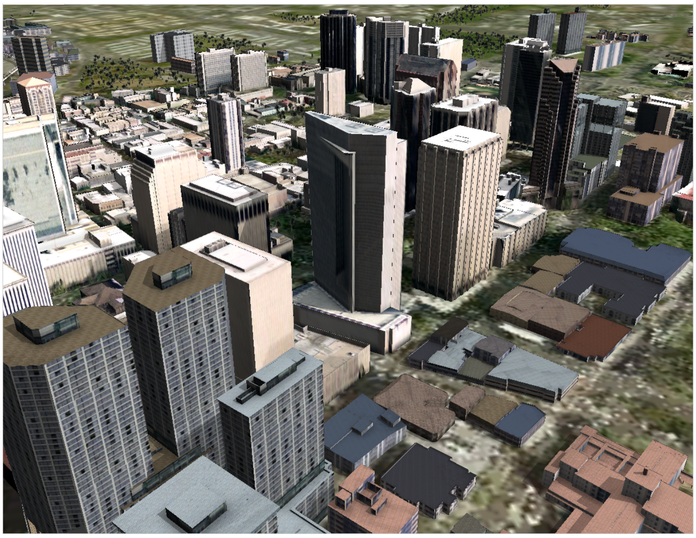
Because the terrain has volume, it is important to adapt the created structures to the terrain to give that sense of volume and realism to the scene using the "Align to terrain" tool.

Figure 3.3: Honolulu terrain and basic forms (City Engine)



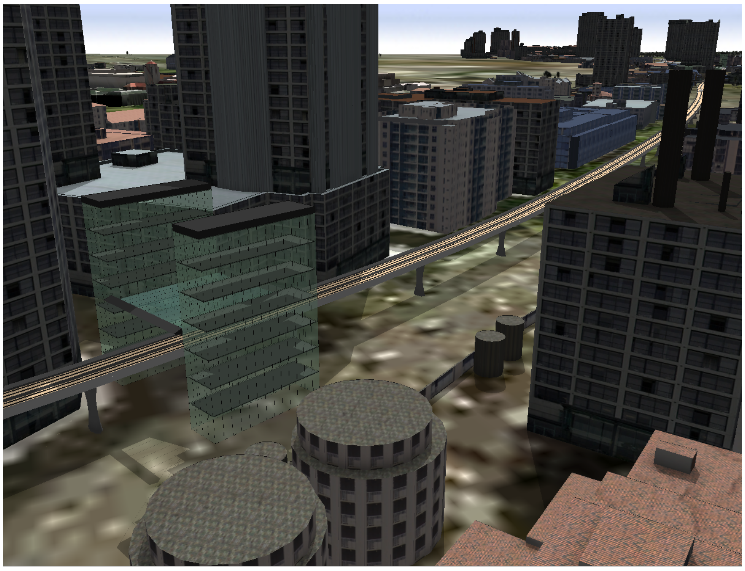
Once these parameters are defined, the next step is to learn how to apply the textures, in this case they are textures already defined by rules, so we don't have to create or edit them or anything. Each region created in the previous section must have its own textures, as they will not all be governed by the same rule.

Figure 3.4: Honolulu textures (City Engine)



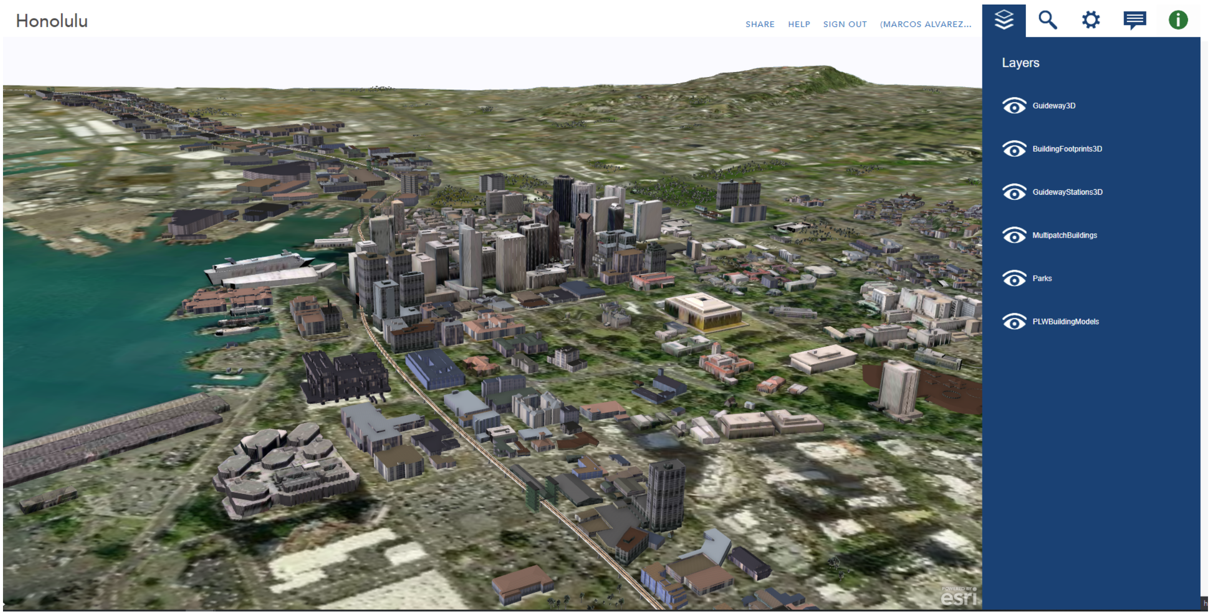
Before moving on to the last step in the creation of this city, it is important to complete the creation of roads, railways and other public roads. Also, in this case, it was essential to create and implement train stations, as these were not well geolocated and a series of parameters had to be edited to fit in with the rest of the city. See Figure 3.5 for more details

Figure 3.5: Honolulu railroads (City Engine)



Finally, we must learn how to import this type of projects in ArcGIS format in order to be able to view them from the browser, knowing how to differentiate each of the zones that we previously separated in our project. This is how the UJI Smart Campus is structured, separated by faculties, allowing us with a small time-lapse to visualise everything created previously. In this case we can see in Figure 3.6 the recreation of the city of Honolulu in ArcGIS.

Figure 3.6: Honolulu ArcGIS (ArcGIS)



Once the basic concepts of CityEngine are understood, the next step is to try to apply the concepts to the medical school building. It is here when problems start to appear, the first one independent of the building, since due to a folder failure the programme was stuck for several days until, thanks to the help of an IT specialist, the programme was able to work again.

This time in which the programme was inactive was used to gather information and techniques that would allow us to work better with the modelling and editing of surfaces, textures and the use of rulers that would allow us to work properly. To do this we looked for the correct way to model the structure of the building, as well as looking for ways to give texture to it, such as the Fecade Wizard or the texture shape tool.

Once the problem was fixed, they wanted to put everything they had learned during those days to work, but this is when another important problem appeared, as the licence for the use of the programme expired. Fortunately, the problem did not last as long as the previous one, thanks to the fact that both Michael and Juan Camilo obtained new licences and were able to start working with it again.

The first step when modelling the building is to import the data provided with the geolocated building base. If this information were not available, the building would not have any reference when it is located on a map, and therefore it would not be placed in its respective location in the UJI.

Once the plans were imported, we encountered another problem, and that is that since we did not have the plans already geolocated, we provided plans with less information than was necessary. Moreover, in this first contact, these plans did not match the original shape of the building, because the modules were not distinguishable and there were parts that were not understood. (Figure 3.7)

Figure 3.7: First base form (City Engine)

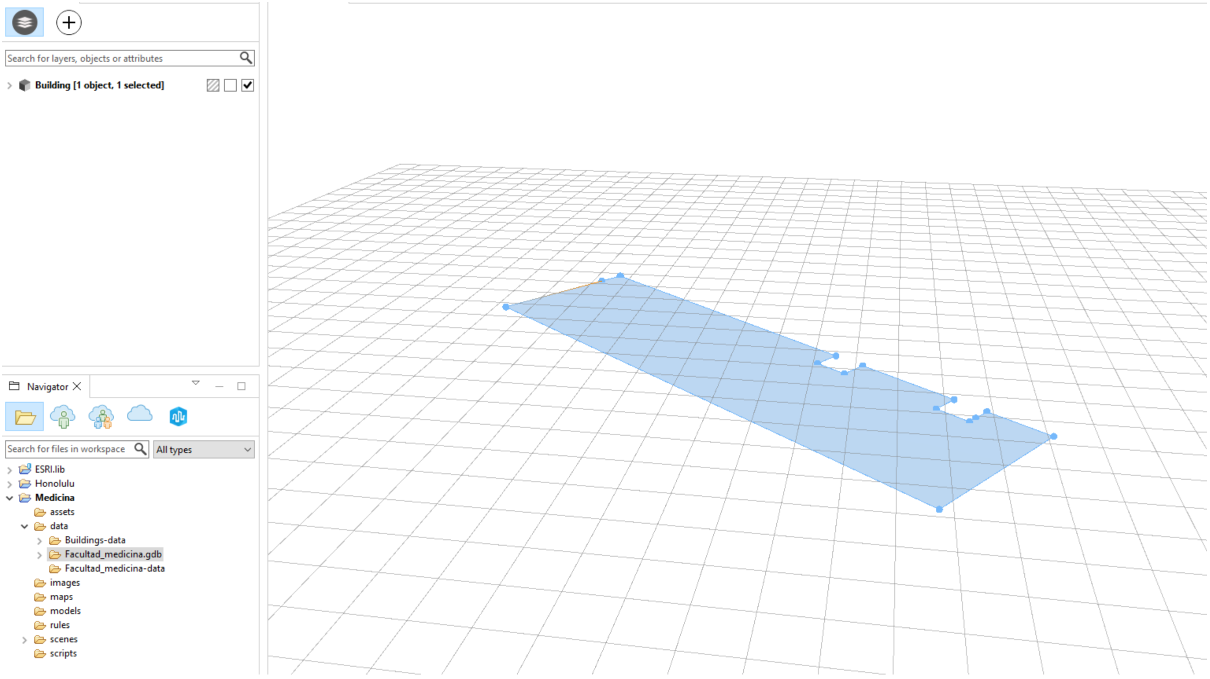
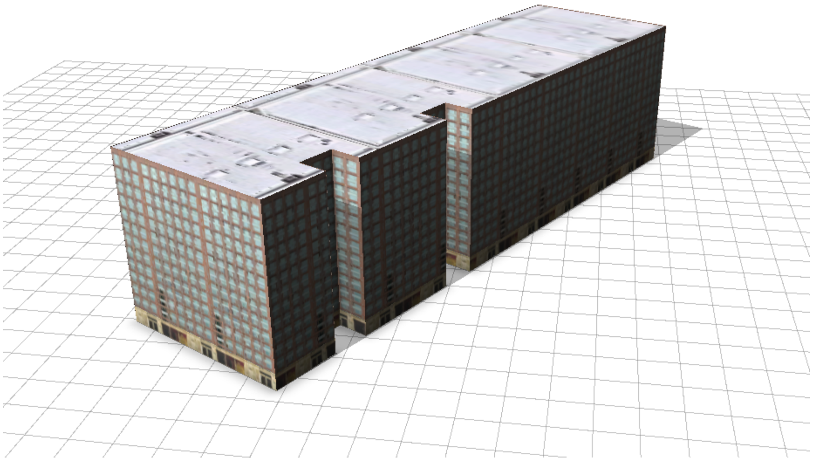


Figure 3.8: First base form with a default rule (City Engine)



As we can see in figure 3.7, it is a top view in which the modules are not distinguishable and have some small gaps that would lead us to understand that they are the separation of these modules. After being reviewed, as expected, the base form was not correct, so a meeting had to be arranged to discuss the issue. Once this meeting was over, a visit to the faculty was arranged in order to check the current shape, as the building is still under construction and has more and more content. From this visit we also took the opportunity to take some photos that will allow us to generate textures as will be explained later on.

This was followed by a further meeting where the overall shape of the building was clarified and the geolocated plan was redrawn. It took a few days to be ready but in the end it was a shape that we could work with properly, this process could be carried out thanks to Juan Camilo for taking care of the geolocation of the building.

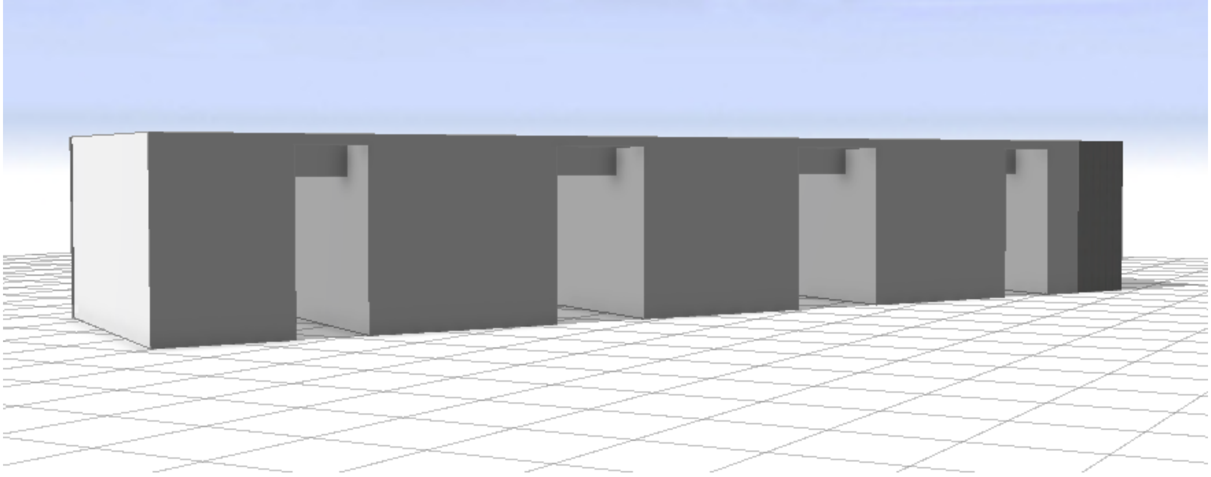
Once the base shape has been obtained, in this case a rectangular figure, which is what can be seen from above, we proceed to adjust the missing parameters by reviewing the AutoCAD documents mentioned above (Figure 3.2), to interpret the range of the 2 modules for which we did not know the information and to determine their dimensions.

The first step to create the building in 3D once the base is determined is to give it height, in this case it was somewhat complicated because in the available plans the height was not found, so we took as reference in the photos that were taken of the building some measures that allowed us to determine the height of the building in a relatively effective way.

Once the basic 3D shape in the form of a rectangle had been obtained, the next step was made possible thanks to the measurements taken from the plans (AutoCAD), which made it possible to create the gaps separating the modules as the rectangle to be moved had to be adjusted to the exact distance. To make this possible we used the tools "Rectangular Shape Creation" to determine the shape of the face to be edited, and "Push Pull Tool" which allows us to adjust the new face to its position, repositioning the new face in its exact place and creating the separation of the modules. This process must be repeated until the desired result is achieved in the building's partitions, both for the roof and for the plants at the back, as well as for the doors, windows and marquees that are visible on the sides of the modules.

The previously mentioned tools will not only be used to give volume to a structure, but also to delimit areas in which the textures to be added will be different from each other. For this process we will use another important tool, "Polygonal Shape Creation", to create more complex shapes that do not necessarily have to be a rectangle.

Figure 3.9: First steps into the 3D model (City Engine)



As we can see in Figure 3.9, the piece is no longer a plan and has some volume, with some first separations created according to the available data, however the building is nowhere near completion at this point. This is why the following weeks consisted of polishing details and even redoing things that were already done previously in order to obtain a more reliable final piece.

Figure 3.10: Modeling process 1 (City Engine)

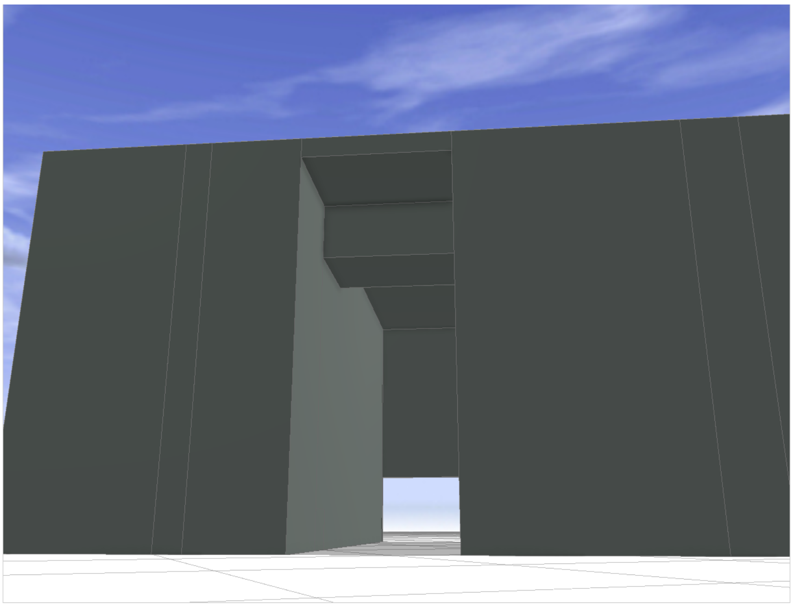


Figure 3.11: Modeling process 2 (City Engine)

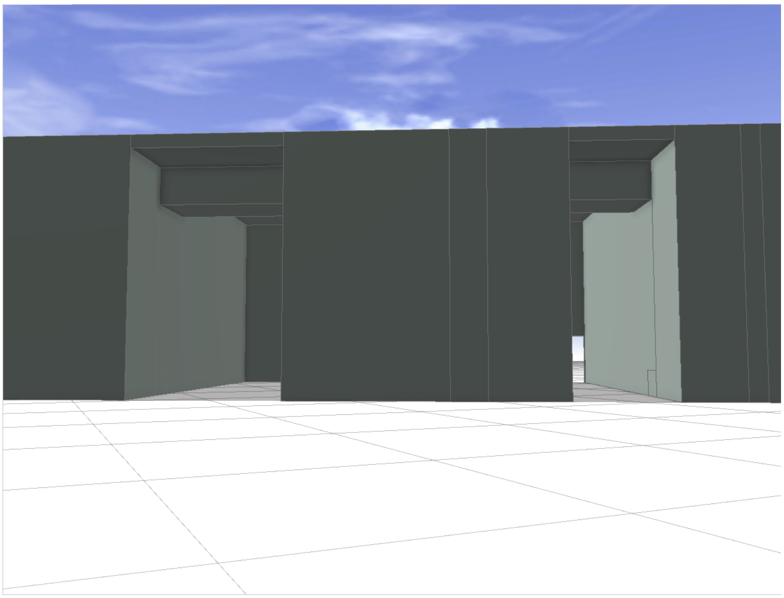


Figure 3.12: Modeling process 3 (City Engine)

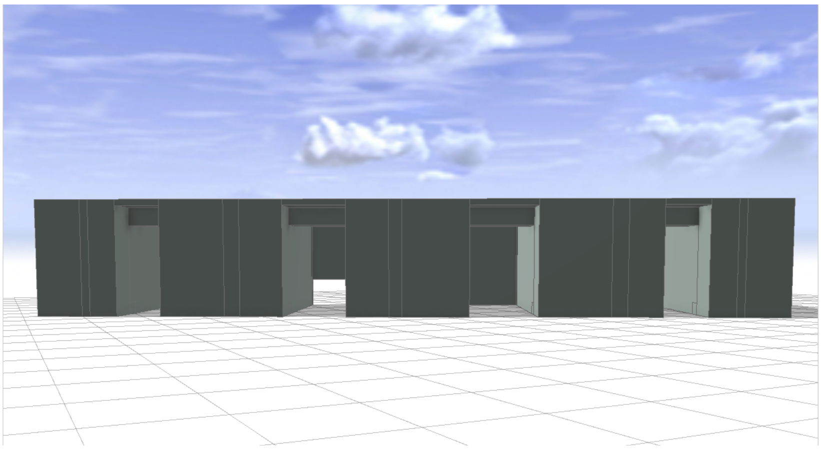
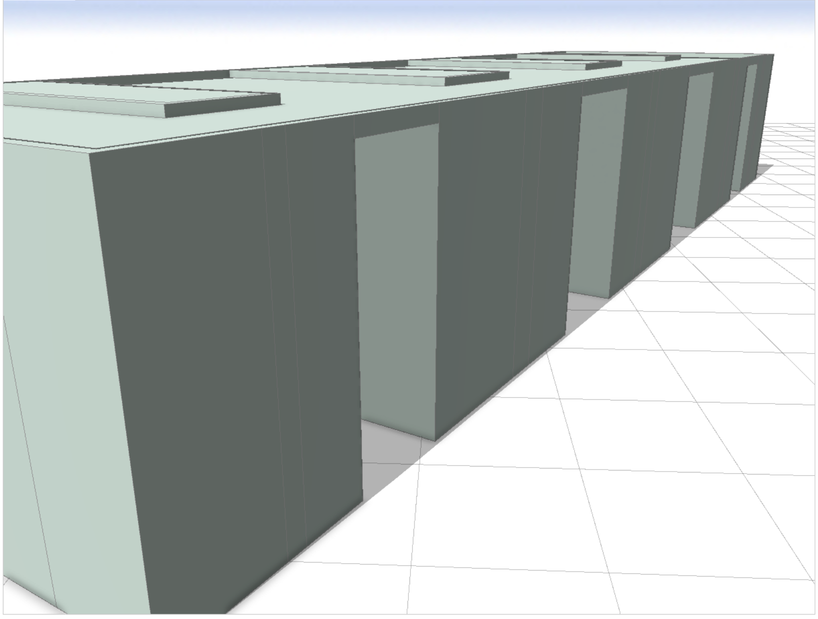


Figure 3.13: Modeling process 4 (City Engine)



During this process we can see in the images shown that the modules do not maintain an identical shape between them and that the building is not symmetrical, which we will also see later when we talk about the textures.

The two unfinished modules were somewhat complicated to determine how they would be joined together, because as we can see in Figure 3.12, some joints are open so that they can cross, but there is another one that takes up the whole of them. In this space there is a wall full of plants that we will see later. This is why after visiting the faculty several times and seeing the development of the faculty, it was deduced that in these modules there would be a gap to be able to cross from one side to the other.

When modelling the roof, it was decided to keep the structure simple and clean, because when we see elements of the Smart Campus, we can see that the roofs are smoothed surfaces with little detail as the purpose does not require it. Even so, some volume was given with the Polygonal Shape Creation tool, which allows us to delimit the roof windows and the edges of the building.

3.3 Problems

The medical school is a simple-looking building, but unfortunately it has not been so. The main reason why it has not been easy is that the building is not finished yet and still has some time to go, according to some sources it will be finished in 2022 or even 2023. This was a problem when we went to take photos for the textures, as we were not allowed to get close enough to see some things in detail, nor were we allowed to take photos from close up to see the textures better.

Other notable problems are those already mentioned above, such as the problem with the licence or the failure with the folders, which made the process difficult and even led to the loss of all the progress made (fortunately it was not close to being the final modelling and they were only the first steps). This problem of folders was in one of the folders that is created when you start a project, in this case in the Model, that to fix it was necessary to ask for help to computer scientists who discovered this failure and had to change some parameters and create a new folder to save the data called Model1.

Figure 3.14: Building under construction 1

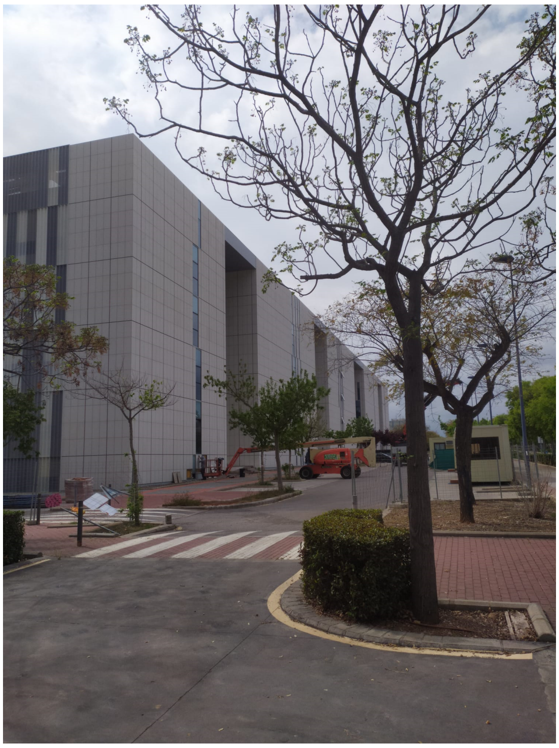


Figure 3.15: Building under construction 2

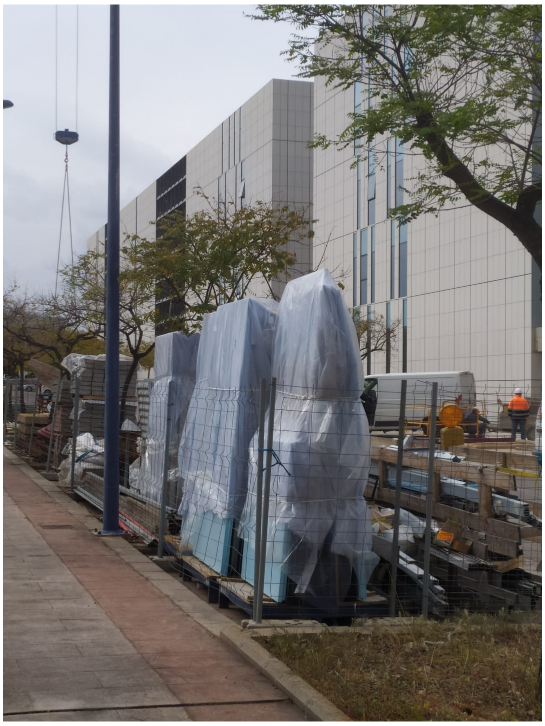


Figure 3.16: Building under construction 3

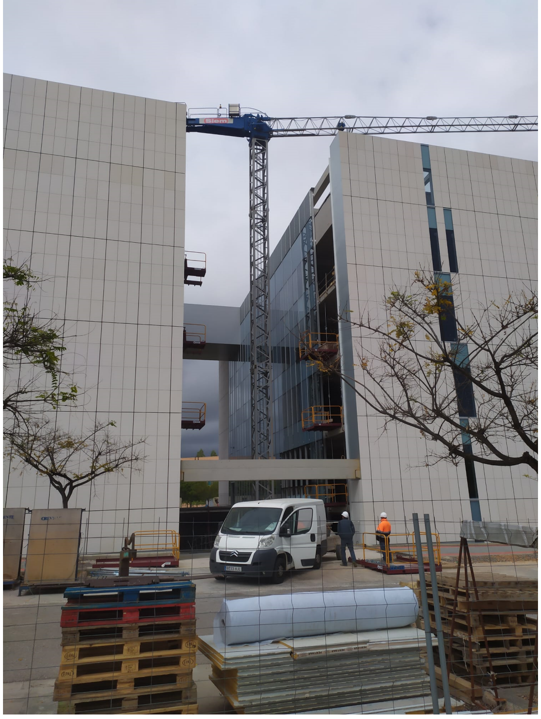
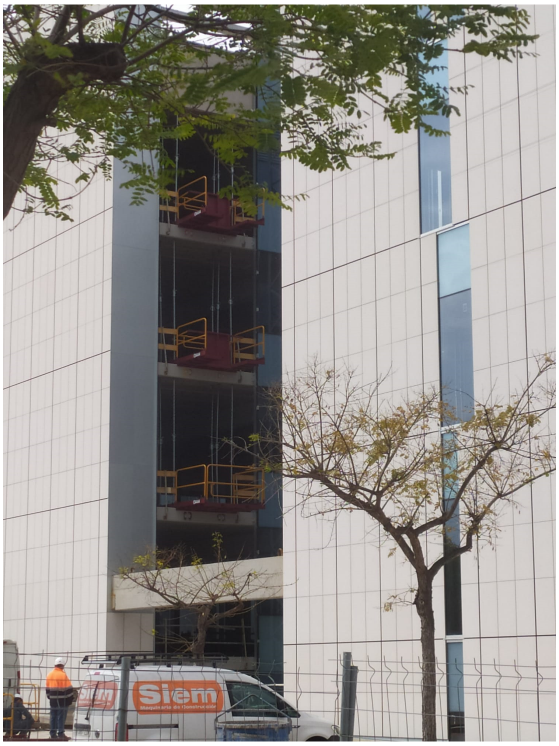


Figure 3.17: Building under construction 4



3.4 Results

Despite the problems encountered, the 3D modelling could be completed, adjusting to the dynamics of the Smart Campus UJI buildings, as it does not seek excessive complexity. This does not mean that later on, when it comes to giving it a personal touch, it won't undergo some changes to achieve a more realistic result.

In the images below we can see the result of the base form.

Figure 3.18: Final 3D base model 1 (City Engine)

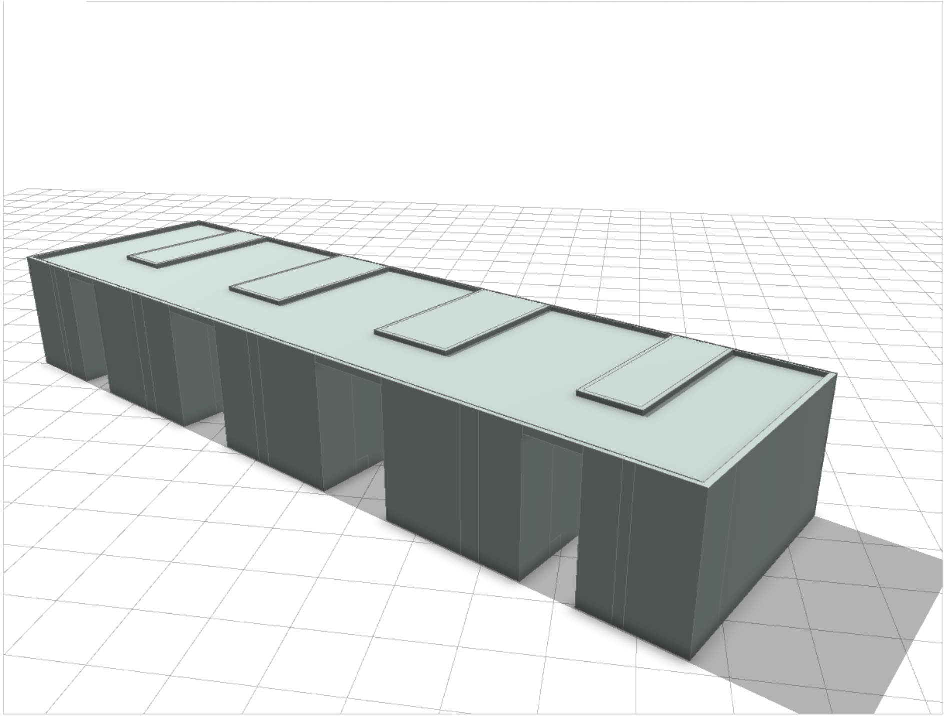
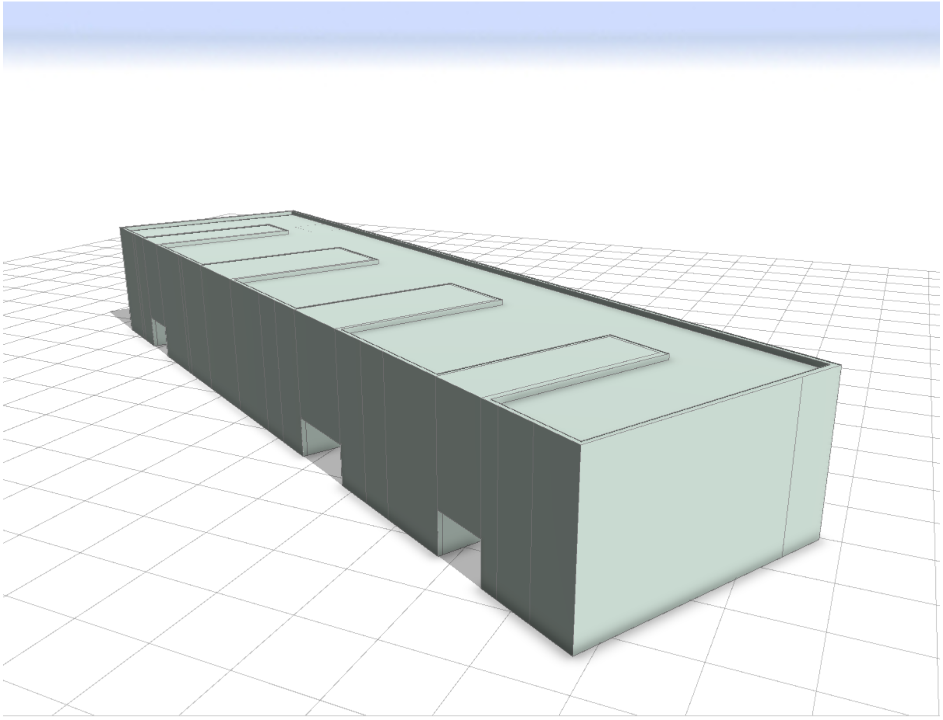


Figure 3.19: Final 3D base model 2 (City Engine)



4

TEXTURING THE BUILDING

Contents

4.1 Photos

As previously mentioned, it was necessary to go to the faculty to take photographs, not only to see the exact shapes, but also to use them to obtain the textures of the building. For this, angles were searched in which the materials of the building were clear and clean, allowing later the edition of these images in order to obtain the desired texture.

These photos were made possible thanks to the help of my friend Pablo Calvo Zorrilla, who offered to take some photos with his camera, with the aim of finding the highest possible quality to create the textures.

The most important parts to photograph when looking for images were the white materials of the main structure, the side metal structure, all types of windows, the doors and the plants we can see in the background.

Figure 4.1: Creating texture 1 (Main material)

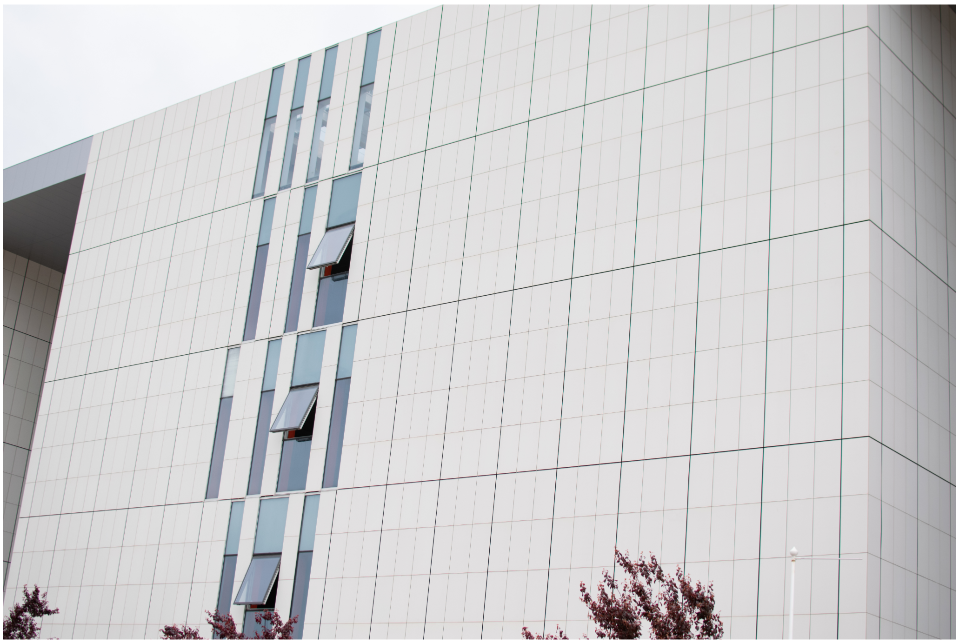


Figure 4.2: Creating texture 2 (Side metal material)

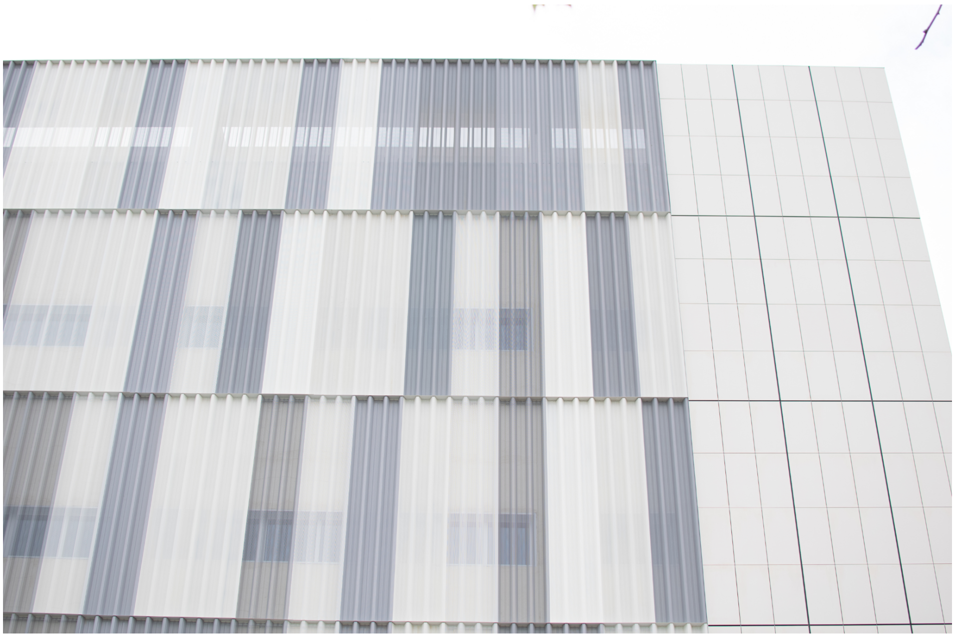


Figure 4.3: Creating texture 3 (Window 1)

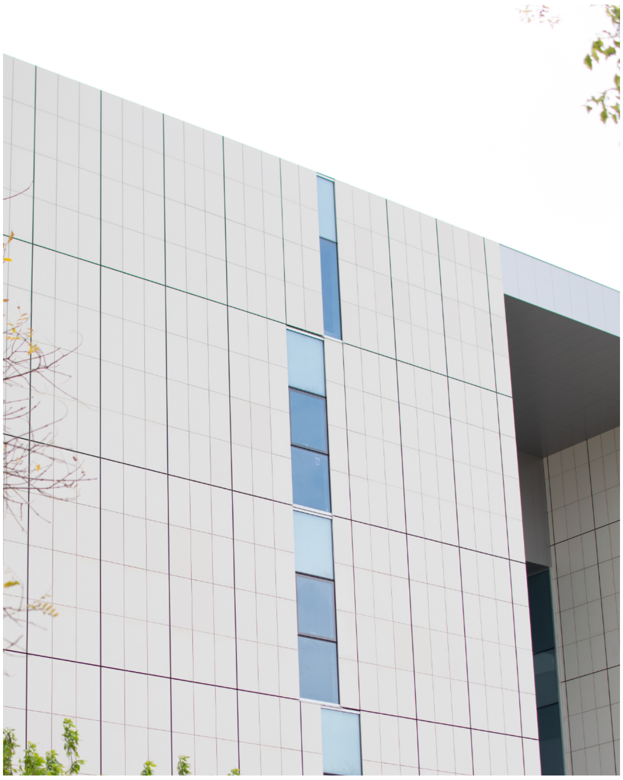


Figure 4.4: Creating texture 4 (Window 2)

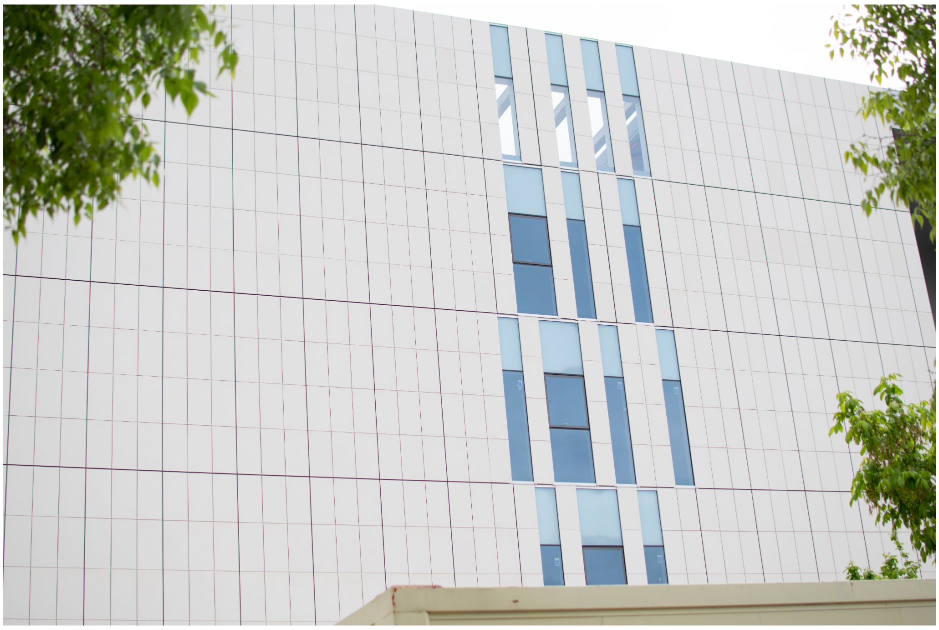


Figure 4.5: Creating texture 5 (Window 3)

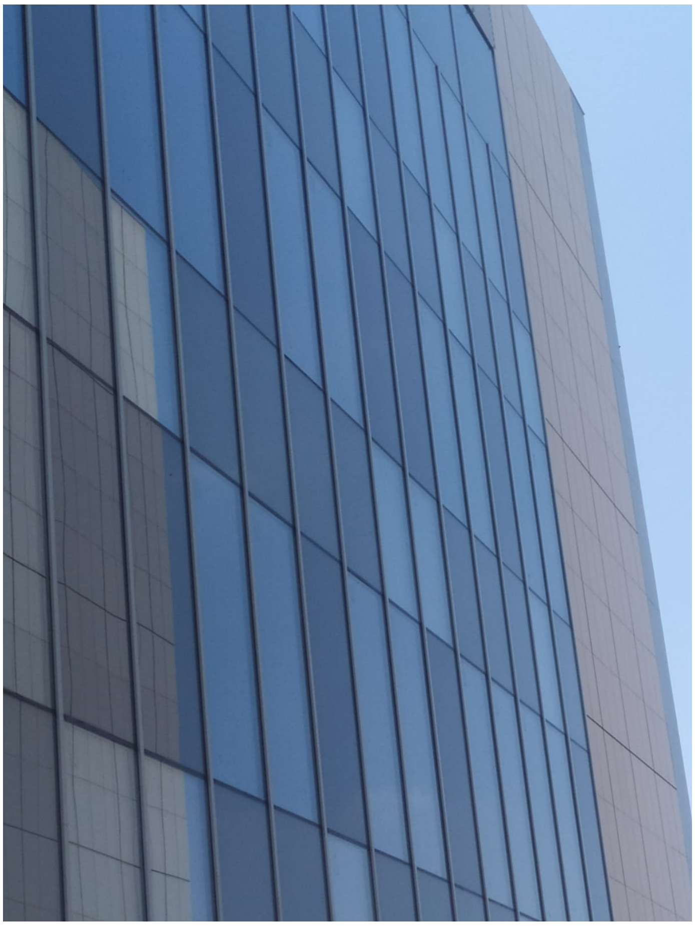


Figure 4.6: Creating texture 6 (Plants)

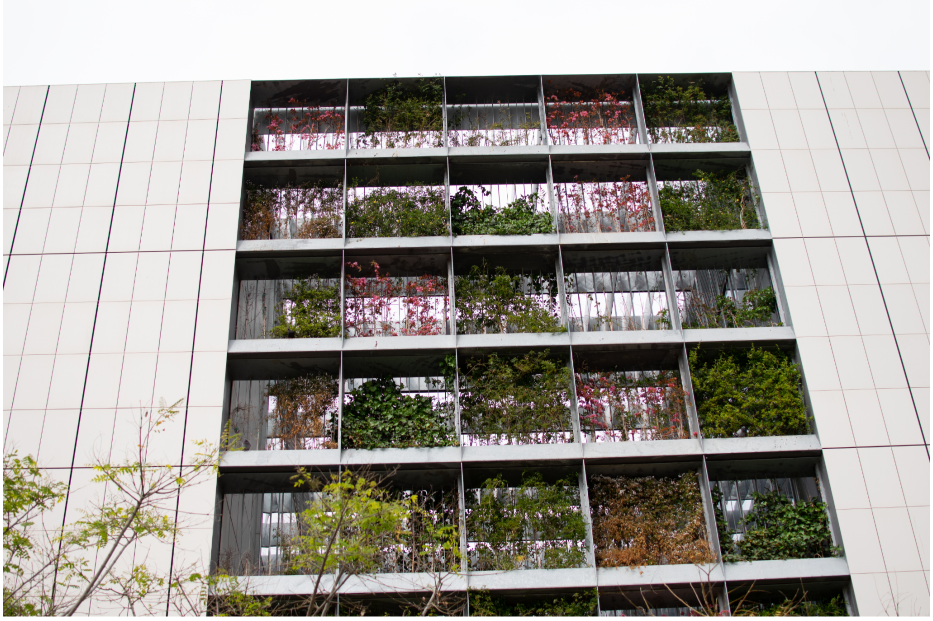
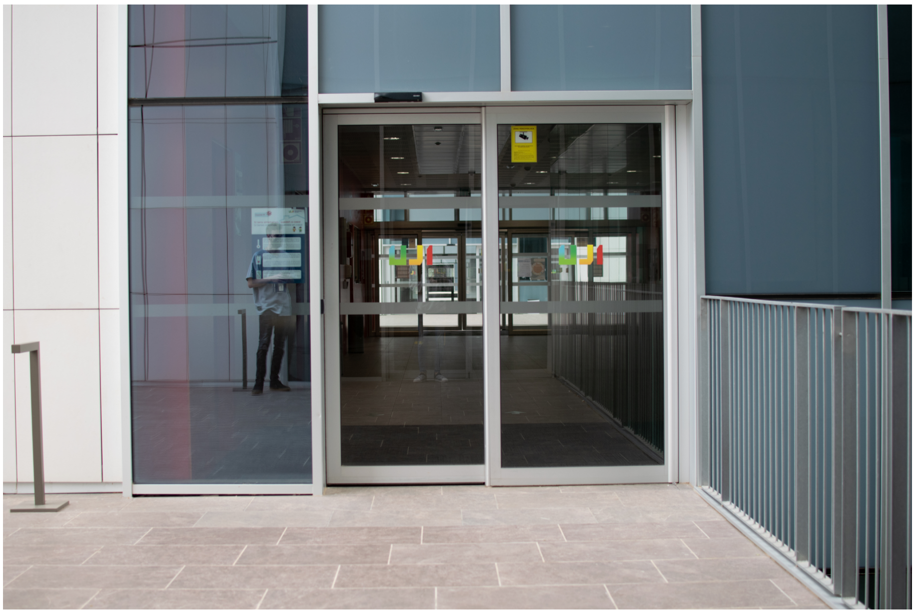
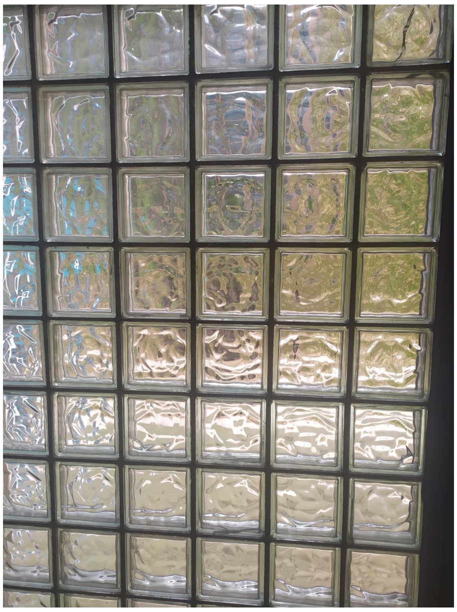


Figure 4.7: Creating texture 7 (Door)



It is from this image (Figure 4.7) that the height of the building and all the other objects could be calculated, because knowing my height, it was only necessary to make an approximation with the rectangles of the wall, and by means of a simple calculation to find out the height of each rectangle individually.

Figure 4.8: Creating texture 8 (glass)



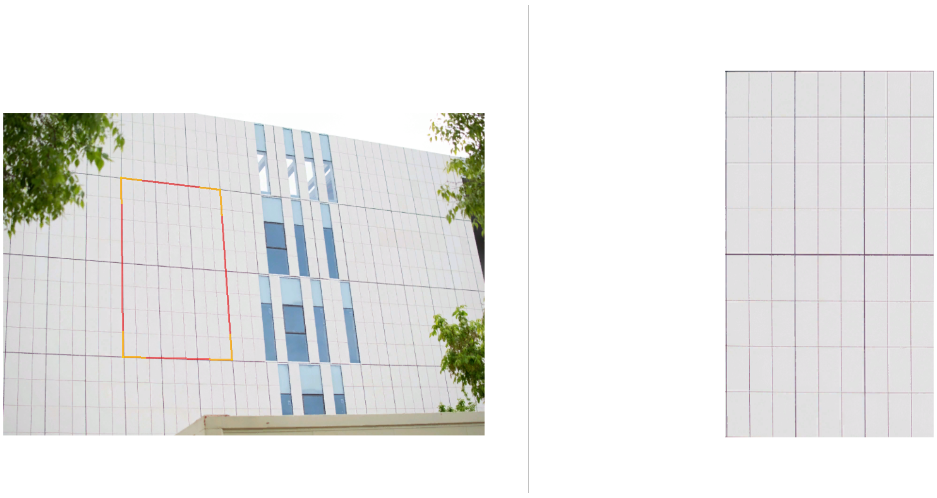
All these images were those that were useful, as many others in the process failed to provide any information and those already shown were sufficient. It is from these images from which the textures are taken that I will explain below how they were implemented.

4.2 Implementation

The images that were used in the programme, in order to convert them into a texture, first had to go through an editing process, which would make them look like a texture regardless of the angle at which the image was taken. To do this we will have to cut the image with a tool provided by the programme, which serves only for this purpose. This tool is in charge of delimiting firstly with an estimate of the area that it interprets that we want to crop. Once it delimits this area, if it does not coincide with our objective, we can move the polygon that it generates to obtain what we want. This polygon can be adapted to any angle of the surface, allowing us to create textures of practically any image.

It is with this method that the images shown above will be converted into a texture that we will work with to adapt to each area.

Figure 4.9: Cropped image (City Engine)

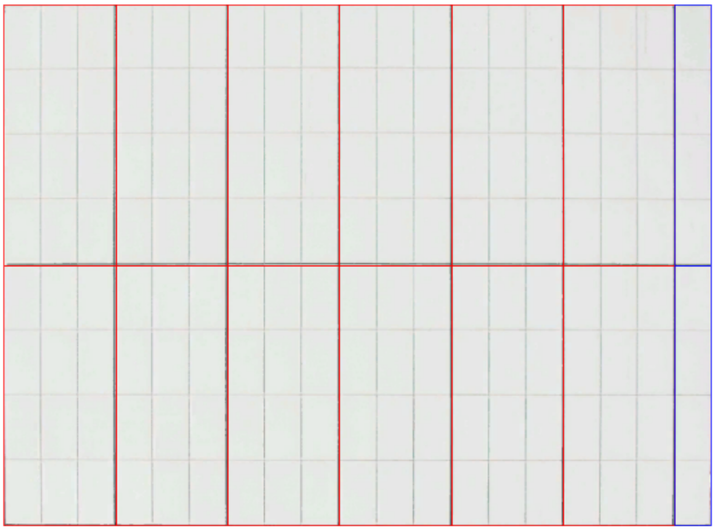


As we can see in Figure 4.9, the created rectangle provides us with some handles which will have a different function, the red lines will allow us to move the whole side of the polygon without deforming its adjacent angles, while the yellow corners allow us to edit the angle of the polygon to better adapt the figure, thus obtaining the result we can see on the right side.

Once we have all the textures cut with this tool, regardless of their result, they will surely have to go through another tool that allows us to apply it in the most useful way.

First of all we tried the "Facade Wizard" tool, which creates rules in the program that define how to work with the texture regardless of the way it is to be applied. In this tool we could separate parts of the texture creating repetitions of some areas that allowed them to be generated along the surface. These repetitions can be created by delimiting the area individually, which can be seen in blue, or as a whole, where the programme separates the area proportionally and creates the repetition, which can be seen in red. (see Figure 4.10)

Figure 4.10: Facade Wizard (City Engine)



This tool, although it is very useful and at first seemed easy, had to be discarded as not all the textures that were created adapted well to its surface and produced strange effects, which is why it was decided to continue searching for a suitable form.

At this point, the only thing that seemed feasible was to create textures that fit perfectly in each section, as it did not seem to take much time. First mistake, it did take a long time, which is why for a few days the time invested in both creating these textures and searching for information, was focused on what was the best way to apply the textures to the whole building in the most appropriate way for what we were looking for without the repetition of the original texture being noticeable.

After some time researching in pages, videos and Esri's own documentation, an idea came up that would allow us to mix two methods without wasting all the work invested in hand-created textures. For this we used the "Shape Texturing" tool, which had been tried before but the repetition of the texture was noticeable. Unlike other times, instead of putting in the texture cut by the program, we put in the texture that had been created by hand, adjusting it to the height of the building exactly. The next step was to change some parameters that allowed us to adjust the texture to the horizontal edges, thus maintaining the exact height that we had defined, but then controlling the measurement so that the texture maintained the shape across the width of the surface.

This made the work much easier, as it allowed all the textures to be applied to their respective areas in a very short time. It was only necessary to adjust the parameters that deformed the image on the unselected axis and that was it.

4.3 Final results

After this long process, in which the methodology had to be reworked twice, a way was found to obtain the most accurate result while consuming the least amount of resources.

Figure 4.11 Final texture 1 (City Engine)

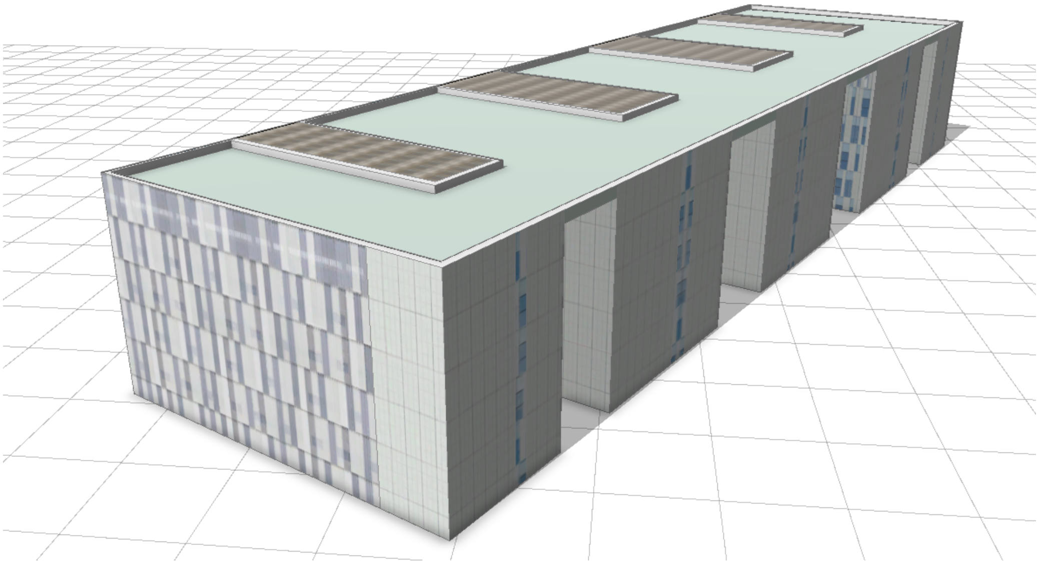


Figure 4.12 Final texture 2 (City Engine)

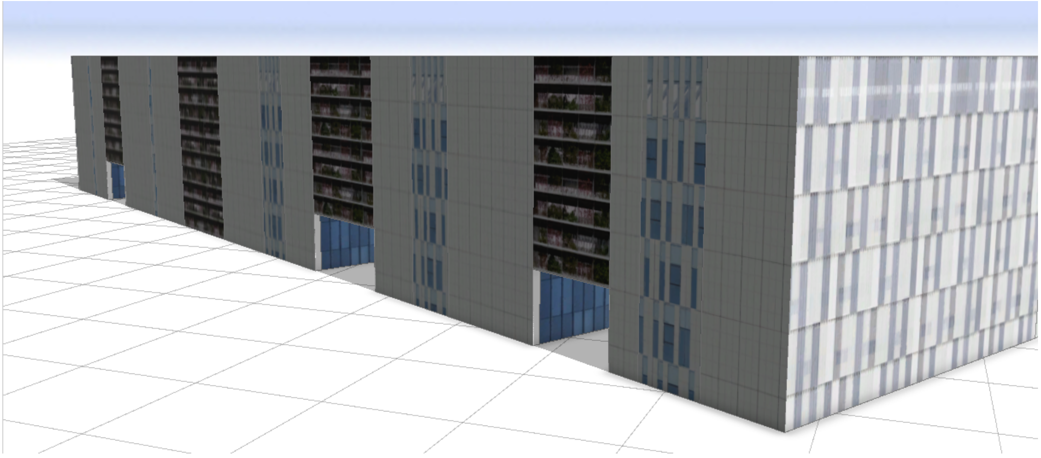


Figure 4.13 Final texture 3 (City Engine)

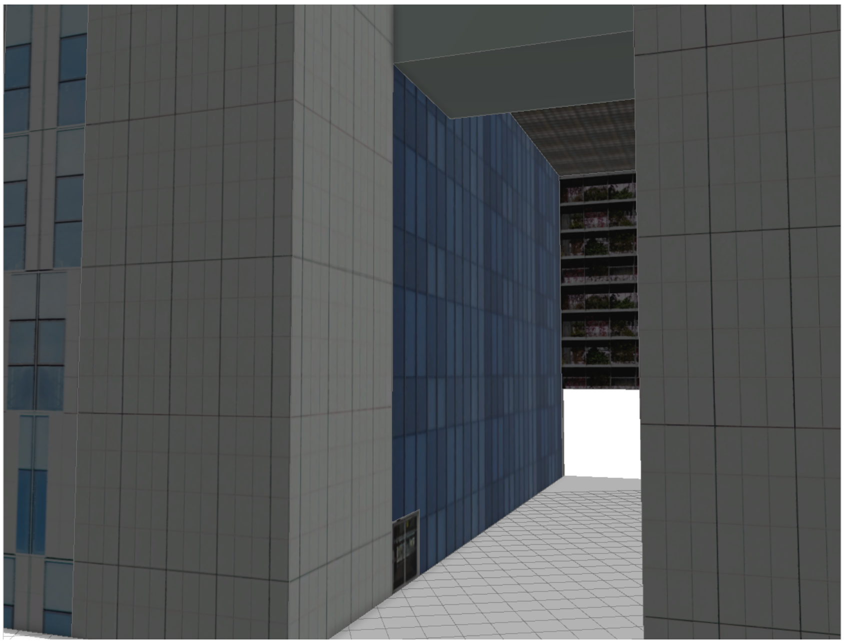
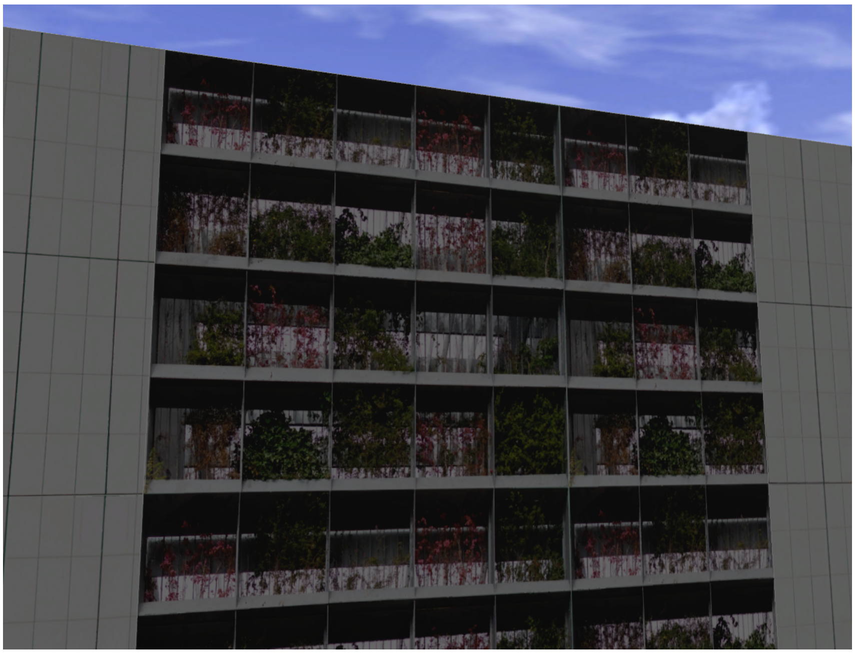


Figure 4.14 Final texture 4 (City Engine)



5

UNREAL IMPLEMENTATION

Contents

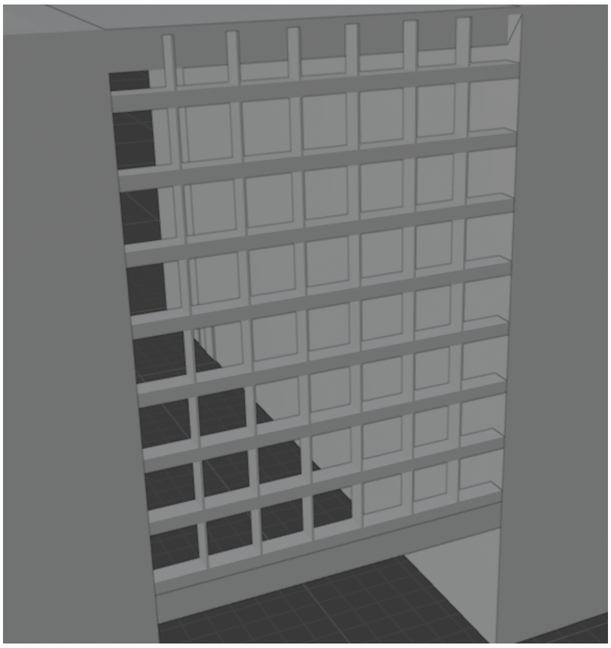
5.1 “Re-doing” the building

When the time came to introduce the building in Unreal Engine 4, the building was placed outside the scene, but not in an adjustable place, it was placed at a very significant distance which gave a few problems when working with it and repositioning it. This is due to the geolocation of the object, which for some reason Unreal could not interpret well.

Once the building was well implemented, it became noticeable that when looking for simplification for the smart campus the building was too simple, and that the use of images to recreate the textures was more striking and visible. This is why it was necessary to do a little remodeling of some parts of the building and added some more detail.

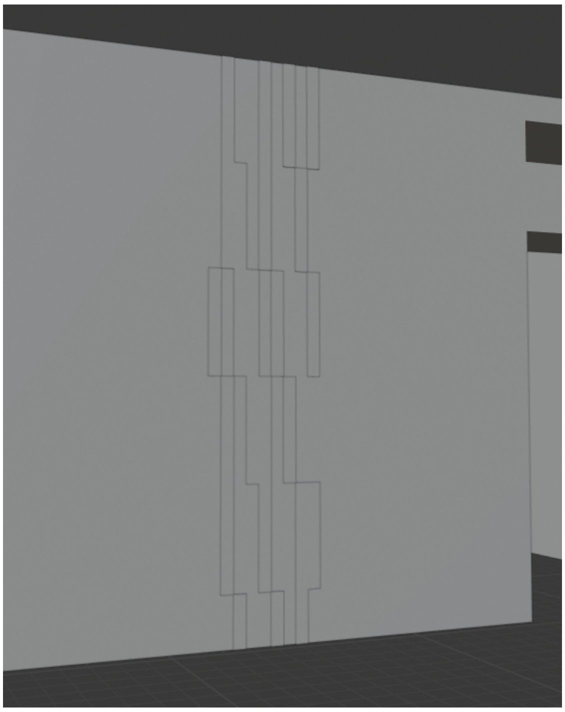
The first thing that needed to be rebuilt was the vegetation supports at the back of the building, as the image was clearly visible and took away from the feeling of authenticity. That is why a hole was made in those walls to implement a grid to give it a more realistic point. To be able to do this, it was necessary to take reference from the image that was there before, so that, using the "Array" tool in Blender, the grid could be recreated in the most appropriate way, always keeping the same distance between them. (See Figure 5.1)

Figure 5.1: Grid (Blender)



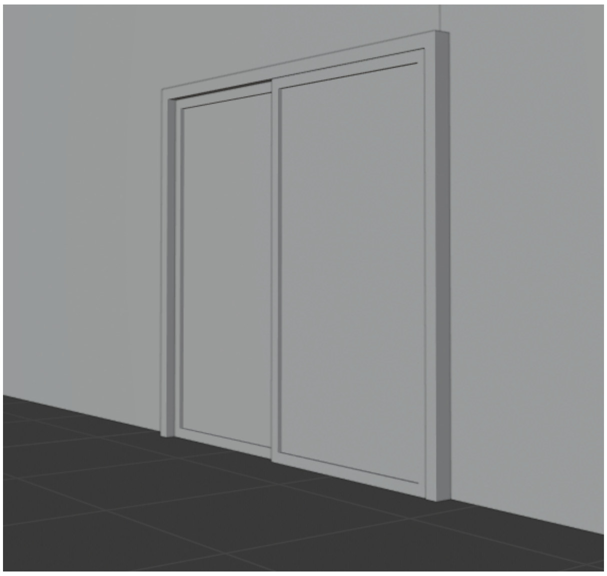
The next important addition would be the windows, this is because in the CityEngine model you can see at a glance that it is a texture taken from a photo and it can become repetitive. Modelling the windows allows the model to apply a single texture on the walls and then place the windows in the right place. This also gives a more realistic feel to the model because now up close, it will give a sense of volume and not be a flat piece. (Figure 5.2)

Figure 5.2: Windows (Blender)



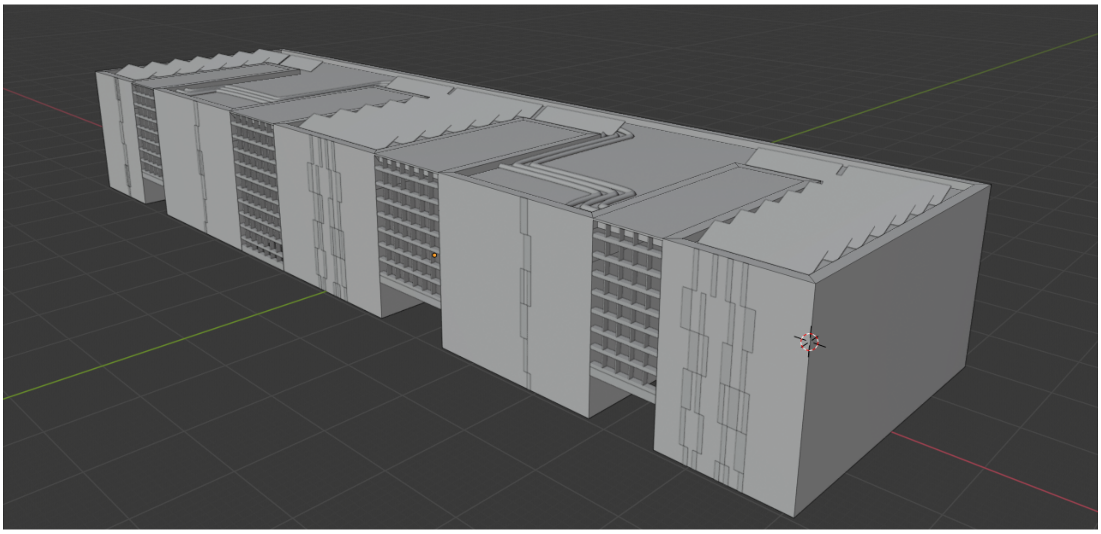
After the windows, doors were introduced for the same reason, as they lacked volume with the texture and looked too forced. Another reason why they were added is that in the texture you can see the reflection of the person who took the photos, making it less believable that he was seen in all of them. First the frames were created, and from them some sliding doors. (Figure 5.3)

Figure 5.3: Doors (Blender)



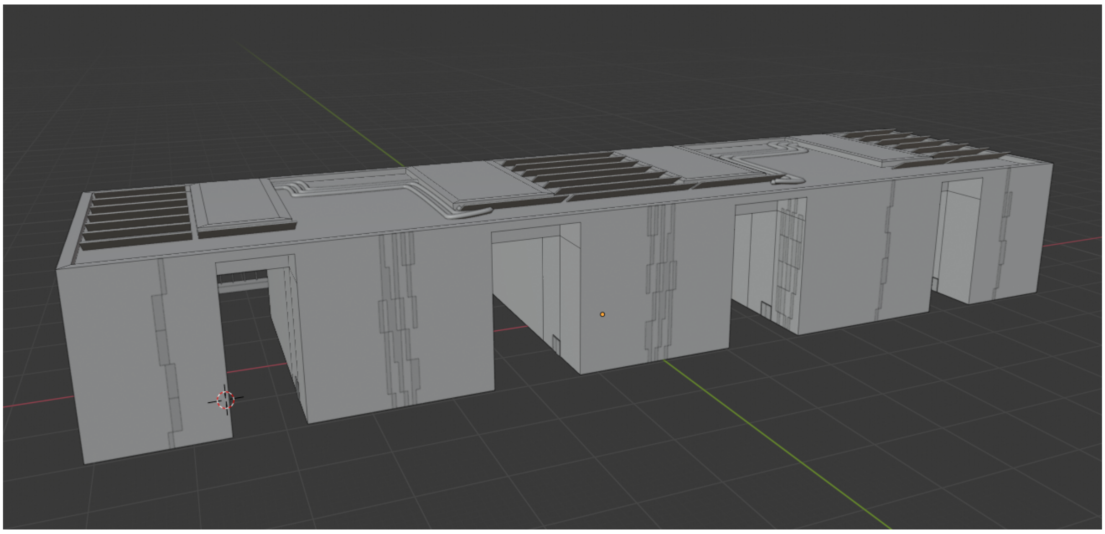
Finally, the roofs had to be redone, as they looked good for the Smart Campus, but when it came to adding them to a game engine like Unreal Engine 4, they were a bit too simple and too short. That's why the solar panels that can be seen at the top of the building and a series of pipes that are found in the building were added. (Figure 5.4)

Figure 5.4: Full model (Blender)



These changes were necessary because with the available knowledge levels of CityEngine we could obtain the expected result for the Smart Campus, but when implemented in Unreal it is not enough to reflect the desired amount of detail.

Figure 5.5: Full model 2 (blender)



5.2 Texture creation & implementation

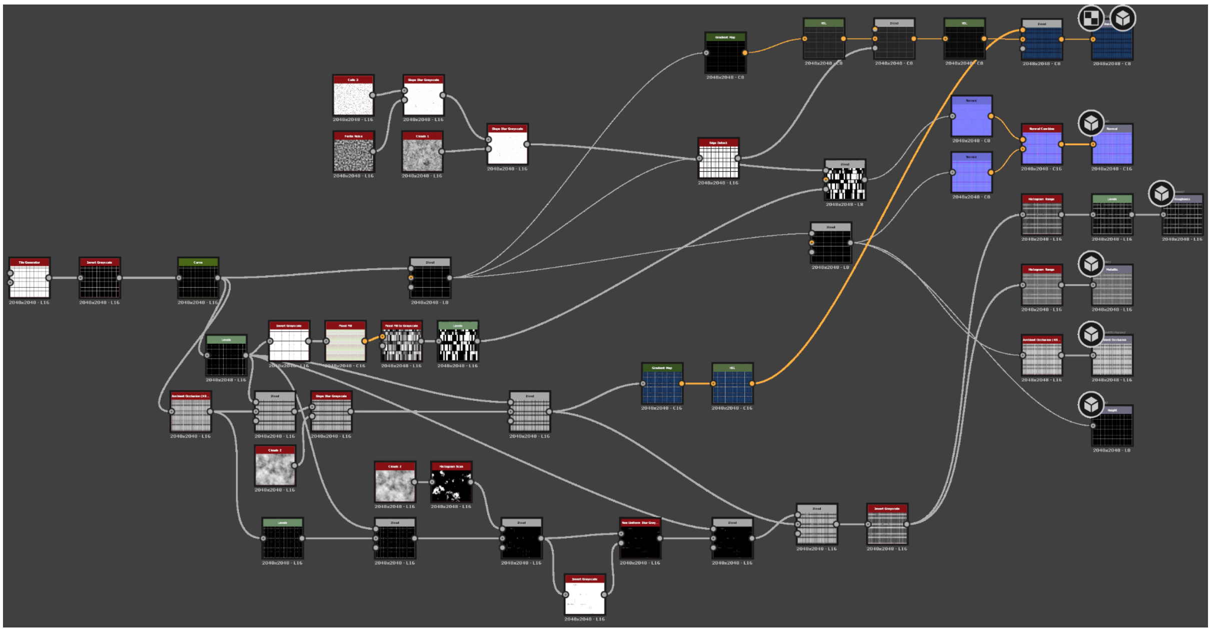
For the textures of this building, the aim was to use the CityEngine textures but having an improved version of the building, one of the decisions taken was to use Substance Designer to create the textures of the new materials, giving them a more realistic look.

The first material to be created was the windows and walls with windows, because in CityEngine they looked good but were too flat and did not fit as desired to the new pieces.

When creating this material, the aim was to create the rectangular shape of the windows without them all being the same size, first a row of taller windows and then a row of smaller ones. To create this mesh it was necessary to start with a Tile Generator, which allowed to create the rectangular shapes of the base, resulting in the edges of the windows. In order to focus on the windows first, the greyscale had to be inverted, allowing it to work with the glass area.

Once the base shape was obtained, the work began with curves and levels that allowed the intensity of what was being worked on to be adapted. These parameters were mixed using the "blend" tool, several noise layers such as "clouds", "perlin noise", "cells" and many others, all of which were processed to give a realistic tone to the window. All this was finally converted into colour layers with bluish tones, normal maps, metallic reflections, ambient occlusion and the height to provide some volume. (See figure 5.6 to see the hole map)

Figure 5.6: Nodes of the material (Substance Designer)



Once this material was finished (see Figure 5.7) and seeing that it was close to the desired result, all that remained was to implement it in the model.

Figure 5.7: New window material (Substance Designer)

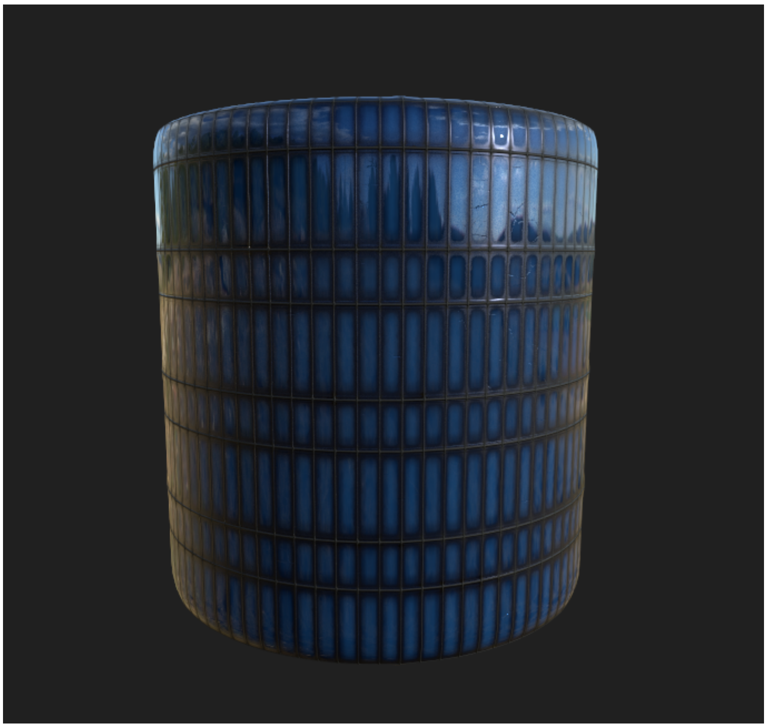
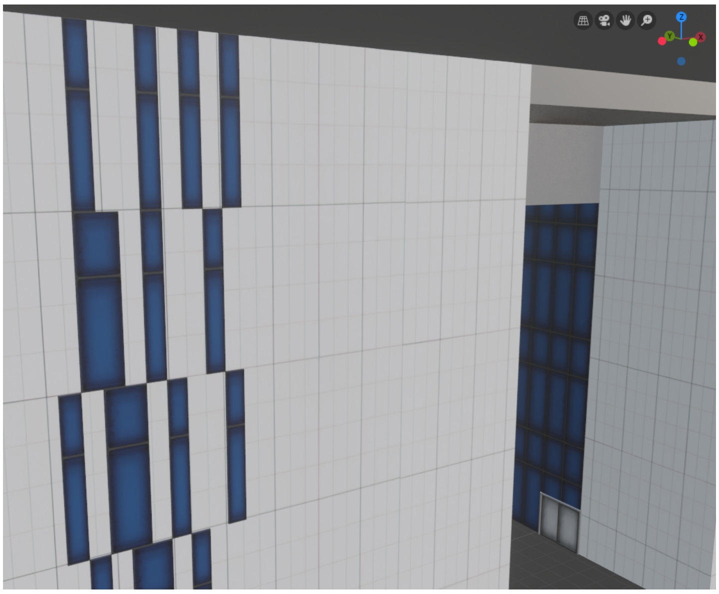


Figure 5.8: New window material 2 (Blender)



The other materials created for the building followed a similar process of creation, changing the necessary parameters to obtain the desired result.

As can be seen in Figures 5.8, 5.9 and 5.10, the images are only orientative, this is because only the texture of the base color is being used, this means that neither the reflections, nor the normals, nor the height nor the rest of the elements necessary for the material can be seen.

Figure 5.9: Textures in blender (Blender)

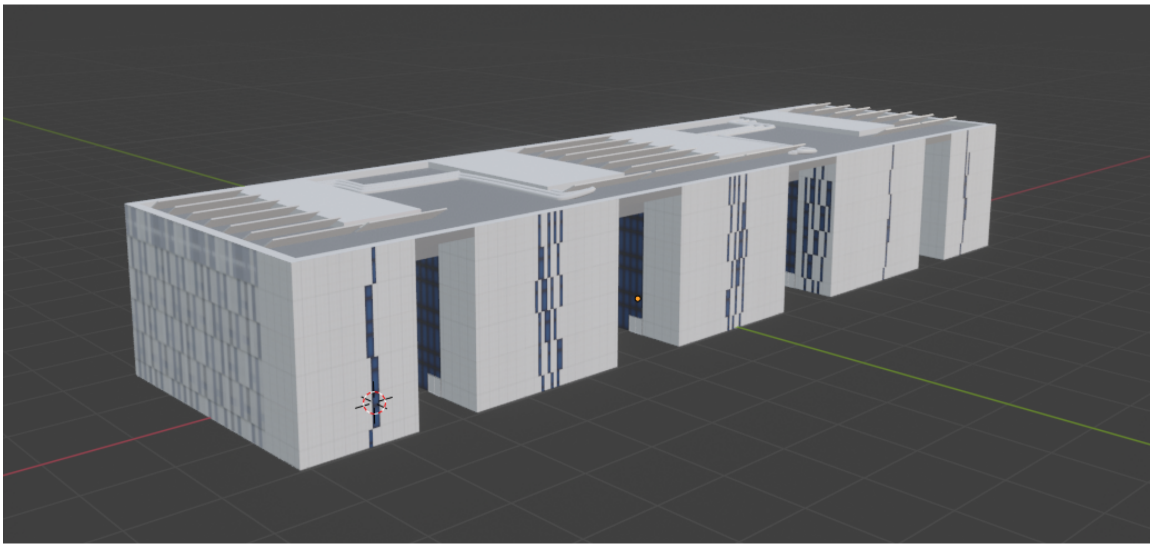
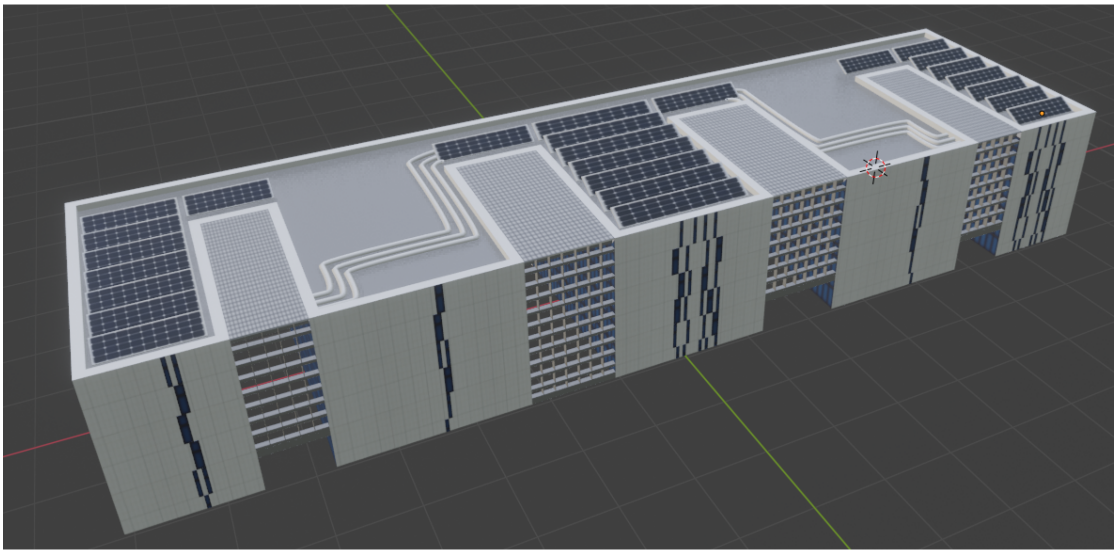
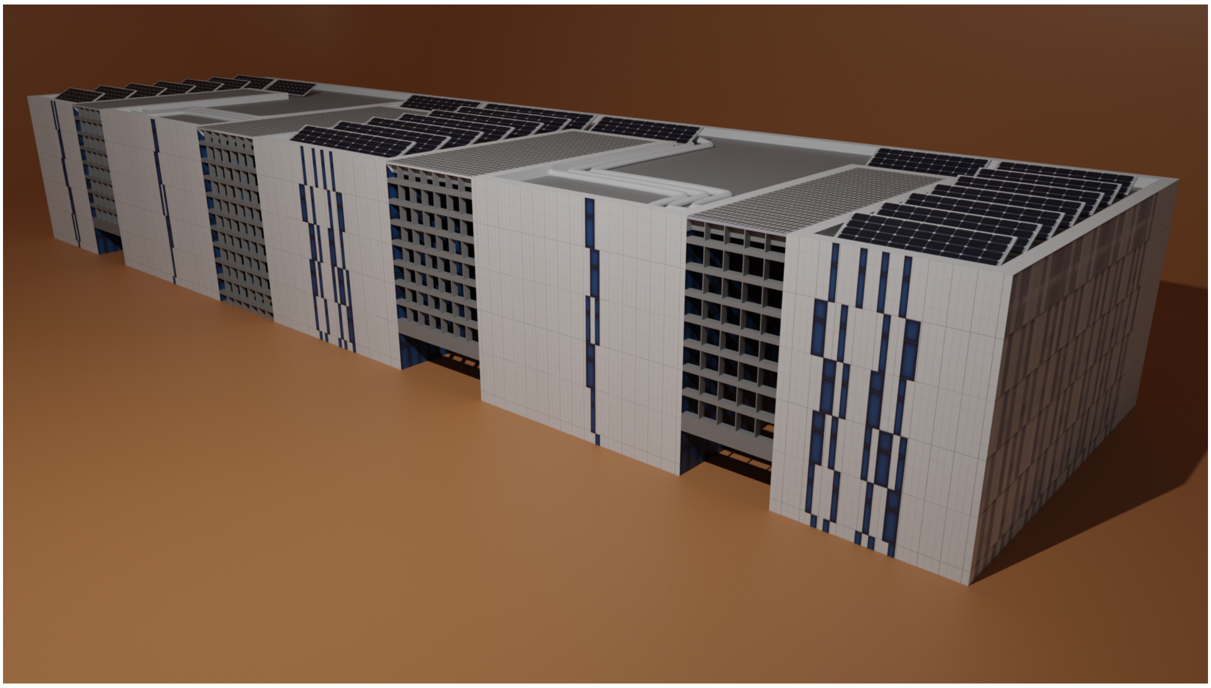


Figure 5.10: Textures in blender 2 (Blender)



Finally, to better see the final remodeling with the textures applied, it was decided to make a render with the Cycles engine in Blender in order to see the approximate final result of what will be implemented in Unreal Engine 4. (See Figure 5.11)

Figure 5.11: Render of the building (Blender Cycles engine)

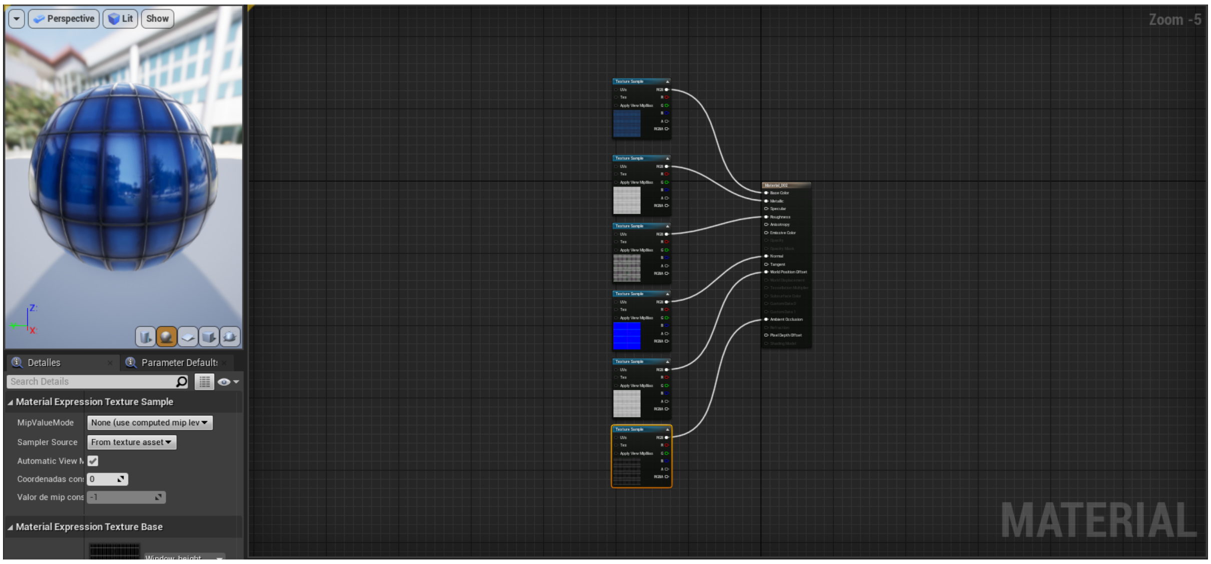


5.3 Setting up the scene

To assemble the scene the first thing that was done was to use an empty scene in Unreal Engine 4. The detailed model of the faculty was imported directly from blender, this allowed the materials to be imported immediately, facilitating the work.

Once the building was introduced in the scene, it was necessary to adjust the parameters of the materials, since they were imported from blender and only had configured the base color. To do this we had to introduce the rest of the texture parameters in the project and apply them in the new material in the corresponding section. (See Figure 5.12)

Figure 5.12: Windows Texture (UE4)



Material Expression Texture Sample

MipValueMode: None (use computed mip lev
Sampler Source: From texture asset
Automatic View M: ☑
Coordenadas con: 0
Valor de mip cons: -1

Material Expression Texture Base

Window_height

This process will have to be repeated on all textures that have been previously created in substance designer, to create the realistic effect. Once all the materials have been arranged, the next step is to add lighting to allow the reflections to be seen, as well as post-processing effects to give a greater sense of dynamism.

The results obtained after this step are closer to what has been sought from the beginning, but there is still a lot of work to be done.

In this step, slight modifications will begin to be noticed that will later become part of the expected redesign as part of the project result. In this case, the most notable part is found in the windows (See Figure 5.13), since in the adapted version the building will not be new, and an ecosystem will have been created around it, which is why the windows will be somewhat broken, but not much, giving an effect of a more abandoned building where vegetation has grown around it (See Figure 3.14).

Figure 5.13: New Window material (UE4)

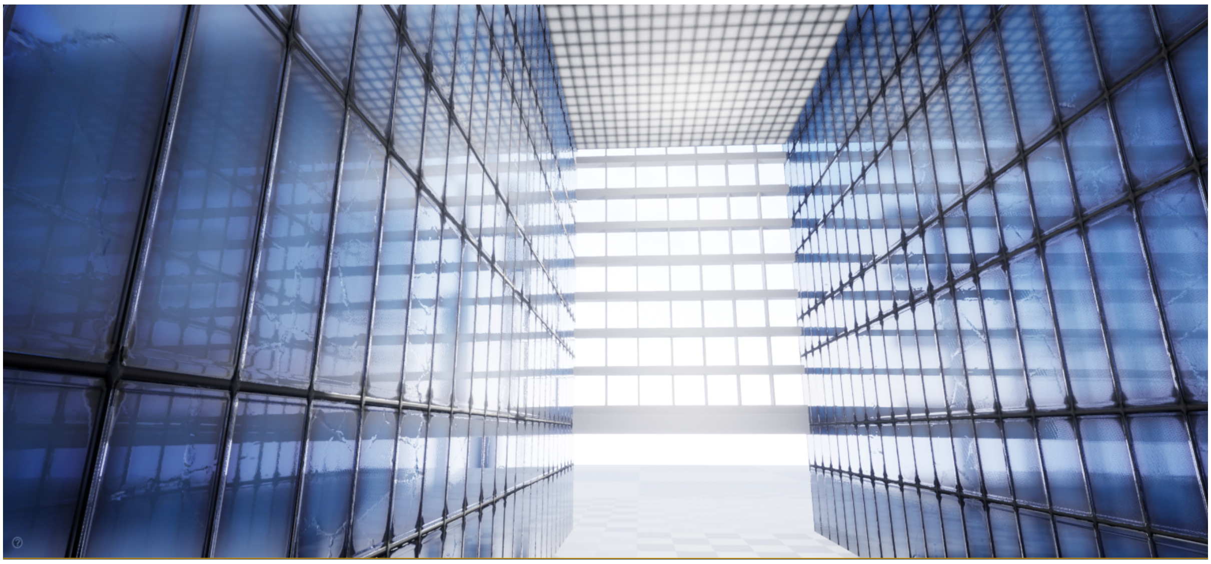
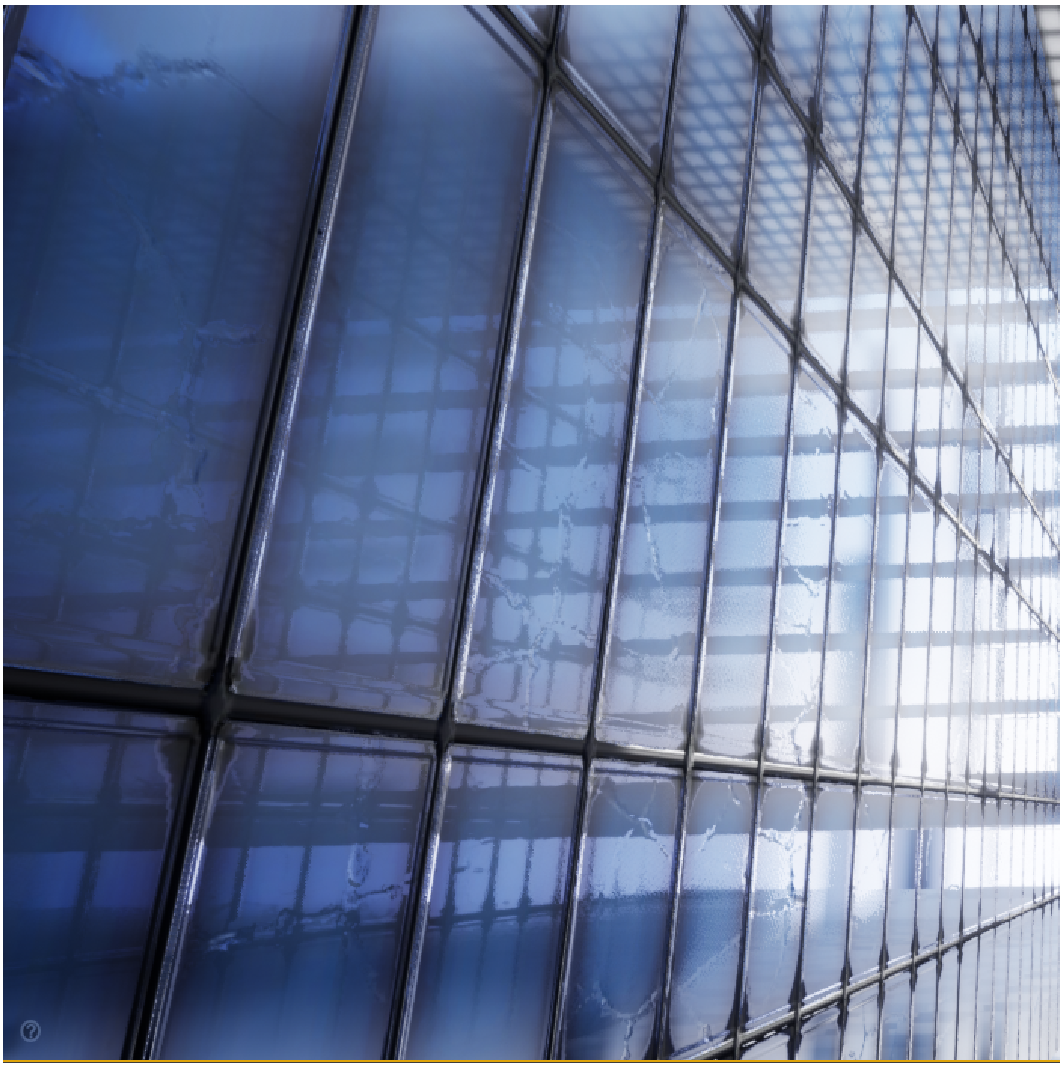
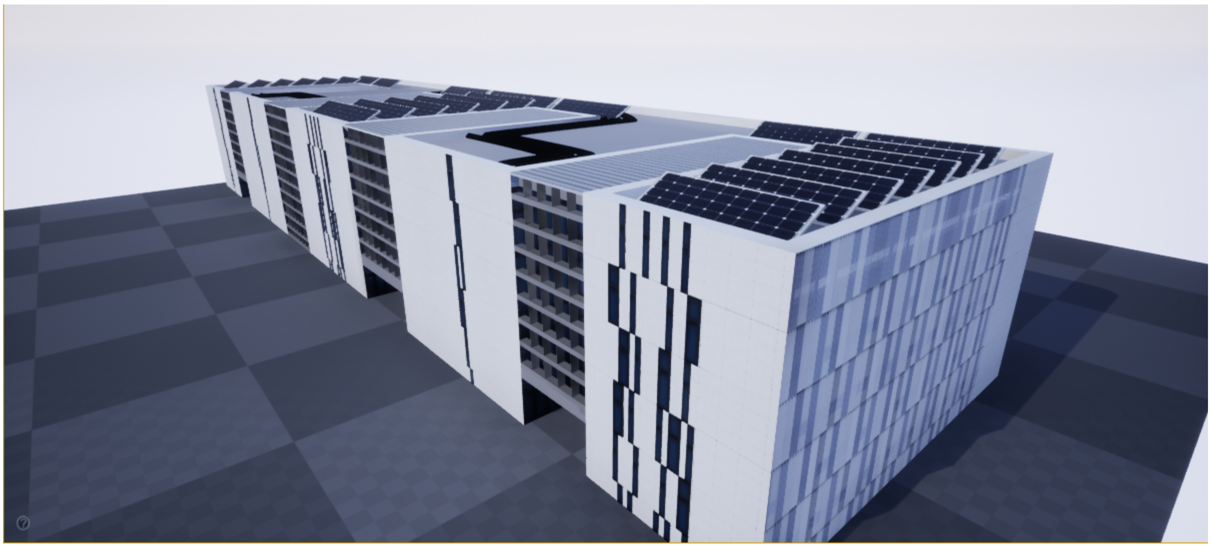


Figure 5.14: Closeup material (UE4)



Finally, once it has been imported, the materials have been adjusted to the desired texture and some lighting has been added, it can be said that the remodeled version of the building is now implemented (See Figure 5.15). After this, it only remains to set it in a scene that keeps some sense with the work and give it the personal touch that is sought in the project. We will see this next.

Figure 5.15: Model in UE4 (UE4)



6

SHADERS AND ORIGINAL LOOK

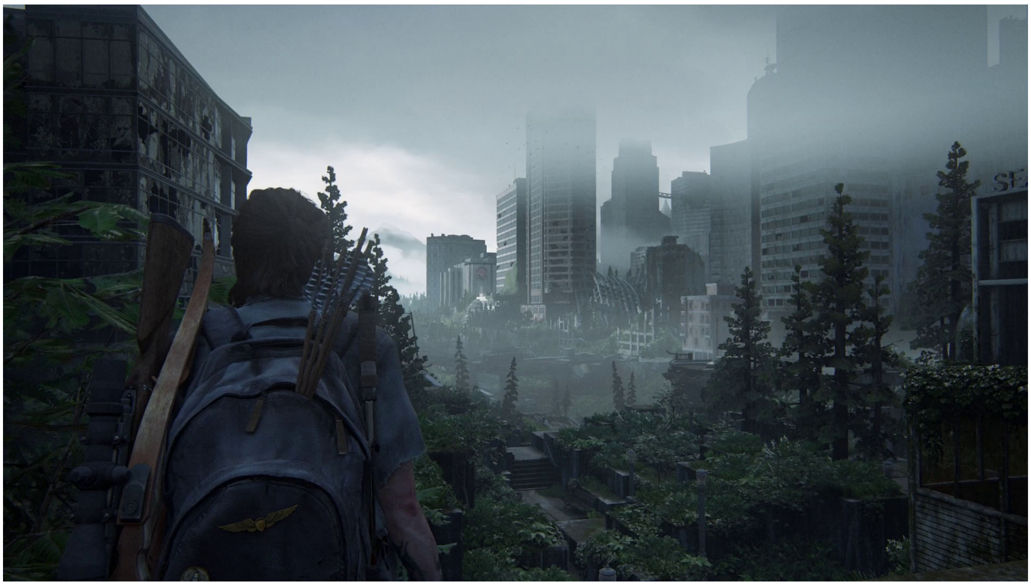
Contents

6.1 Introduction to the idea

What was sought at the time of the project was to give a more personal touch to the building, even to propose something that could be shown in a video game. In the first place, the idea was to focus on a futuristic environment, creating a modern faculty that would allow to interpret an option of how it could be in a few years. This idea gradually evolved to focus on a more Cyberpunk style, with the faculty located in a futuristic and hostile environment, where what would be most abundant would be the neon lights and holograms. This idea was discarded due to the complexity of creating an environment that would adapt perfectly to the idea of the building, as it would require a quite complex post-processing.

At this point, it was necessary to investigate more styles that would look good in the building. For a long time there was no luck and no solution was found, only ideas that were quickly discarded. Continuing with this futuristic atmosphere, in the end a mixture of several styles was found that fit more in line with the objective. The first style that was decided to use is a type of ecological brutalism, in which there are plenty of vines and lots of vegetation, if this is mixed with a more devastating environment where the faculty could become abandoned, the result would be a style very similar to that used in the The Last Of Us saga, where there is plenty of vegetation and abandoned buildings half destroyed (See Figure 6.1).

Figure 6.1:The Last Of Us (Google)



Once defined the style in which the final building will move, as it is not possible to apply all this style to a surface like the UJI (due to not having all the necessary files), it is necessary to rethink the design of the scene. Once defined the style in which the final building will move, not being possible to apply all this style to a surface like the UJI (due to not having all the necessary files), it is necessary to rethink the design of the scene. After several failed ideas, and giving it a refocus from the architecture, the conclusion is that it can be raised as a model of a version of the faculty, placing it in turn with other mock-ups that differ from it. In order to place these "models" a scene of a small room is recreated where, thanks to models that were previously available or some that were created in the process, the perfect environment is generated to place the mock-ups on top of an exhibitor.

This idea is inspired by the game "It takes two", in which the protagonists shrink, turning into wooden and clay figurines. In the scene this will be reflected in the fact that the player, if he approaches the models, will be able to reduce their size to move freely around them.

6.2 Modeling

To adapt the model to the idea explained above, it is necessary to create a series of ivys that later can be placed using the Foliage tool of Unreal Engine 4. To do this, first the branches must be created using bezier curves, which will then be converted to a mesh that will facilitate their manipulation. Once the mesh is obtained the next step will be the creation of the leaves, which will be modeled with the help of a reference sheet that will facilitate the shape of the plane. Once all this is done, the only thing left to do is to place the leaves, for this the blender hair modifier, with the appropriate parameters, is the right one for the function. Following these steps we created three different types of ivy that will give a little variety to the model. (See Figures 6.2, 6.3 and 6.4)

Figure 6.2: Ivy 1 (Blender)

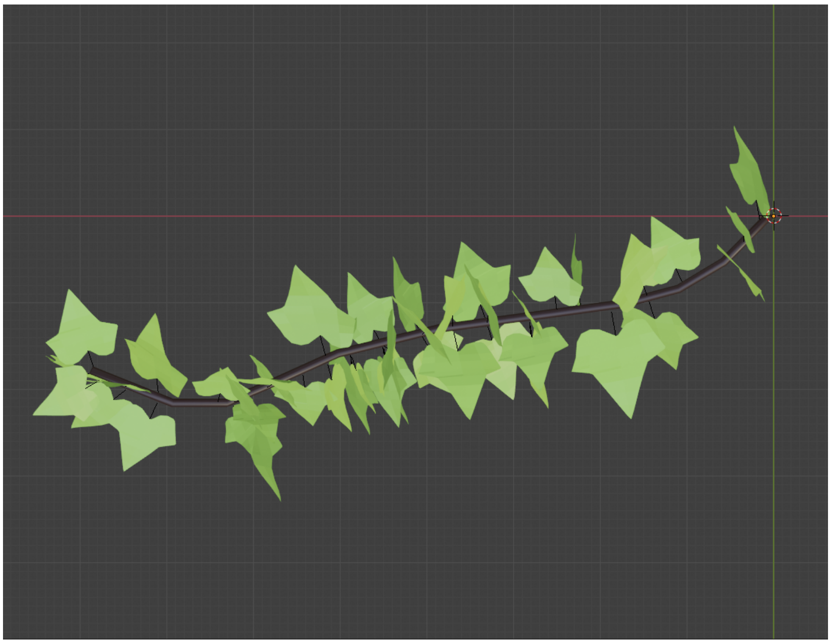


Figure 6.3: Ivy 2 (Blender)

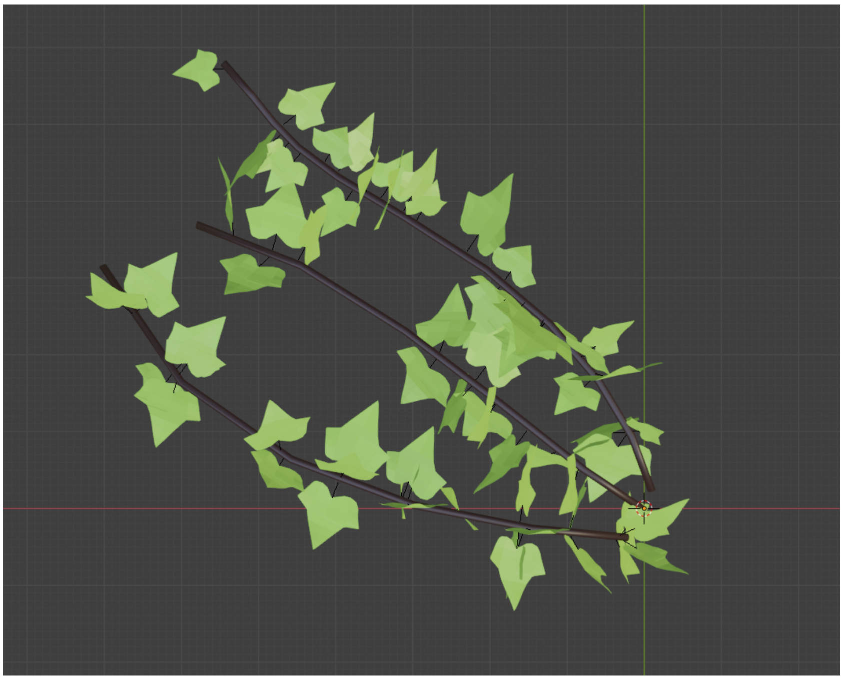
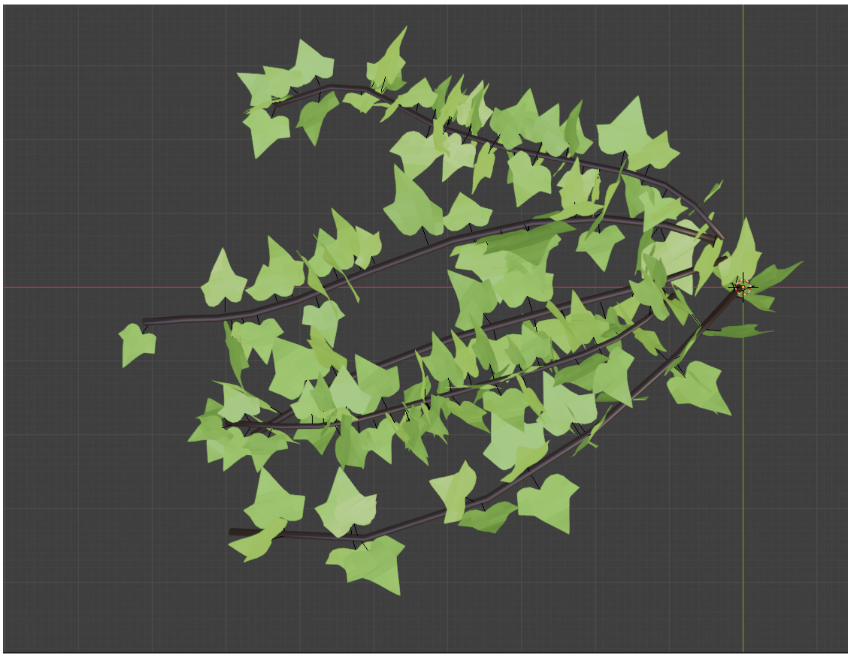


Figure 6.4: Ivy 3 (Blender)

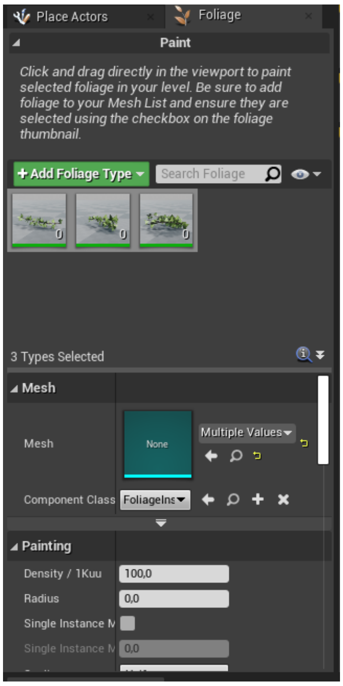


These models are simple since it is not necessary that they are very complex, this is because in Unreal will be added better created textures that will give more authenticity.

6.3 Implementation in Unreal Engine 4

When implementing it in Unreal, it was necessary to use the Foliage tool. This tool works as a brush, which once properly adjusted, allows you to create vegetation environments according to your needs (See Figure 6.5). This tool has to be adjusted perfectly to obtain the desired results, which is not easy, since it is necessary that it adapts well to all the desired surfaces and that it maintains a form that is not very far from reality. The first problem that arose with this tool is that it did not adjust to vertical objects that were not a terrain. This was solved after an extensive modification of parameters that made it work according to the needs. For this tool to work, it is necessary to enter the models that you want to place so that the tool itself mixes them together. At this point the texture of the leaves has already been changed for one with a little more definition, which will give it a more realistic approach.

Figure 6.5: Foliage tool (UE4)



Click and drag directly in the viewport to paint selected foliage in your level. Be sure to add foliage to your Mesh List and ensure they are selected using the checkbox on the foliage thumbnail.

3 Types Selected

Mesh

Mesh: None, Multiple Values

Component Class: FoliageIns

Painting

Density / 1Kuu: 100,0

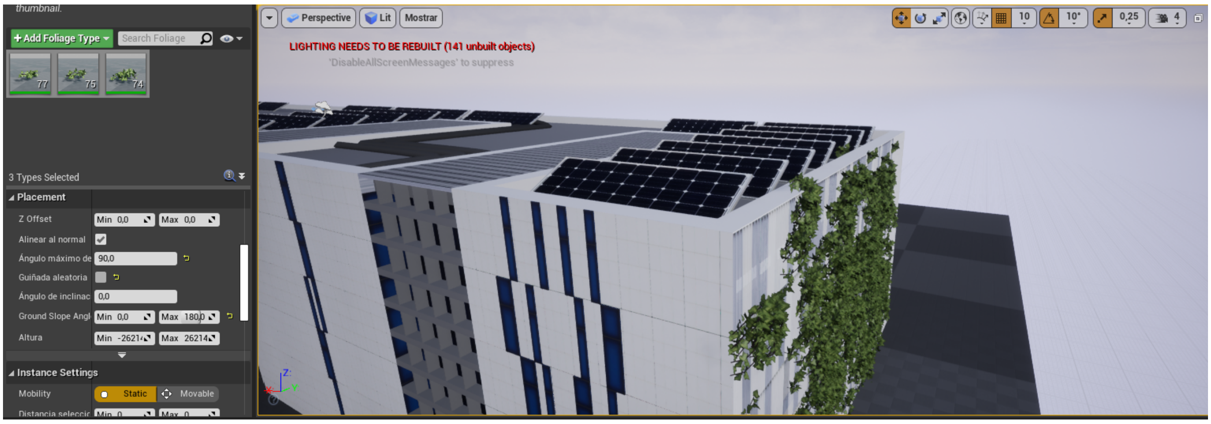
Radius: 0,0

Single Instance M: □

Single Instance M: 0,0

Once all these parameters have been adjusted, the only thing left to do is to apply this Ivy's layer with the brush we have adjusted, giving results according to the objective (See Figures 6.6 and 6.7).

Figure 6.6: Ivy in UE4 (UE4)



LIGHTING NEEDS TO BE REBUILT (141 unbuilt objects)
'DisableAllScreenMessages' to suppress

3 Types Selected

Placement

Z Offset: Min 0,0 Max 0,0

Alinear al normal: ☑

Ángulo máximo de: 90,0

Guiñada aleatoria: □

Ángulo de inclinac: 0,0

Ground Slope Angl: Min 0,0 Max 180,0

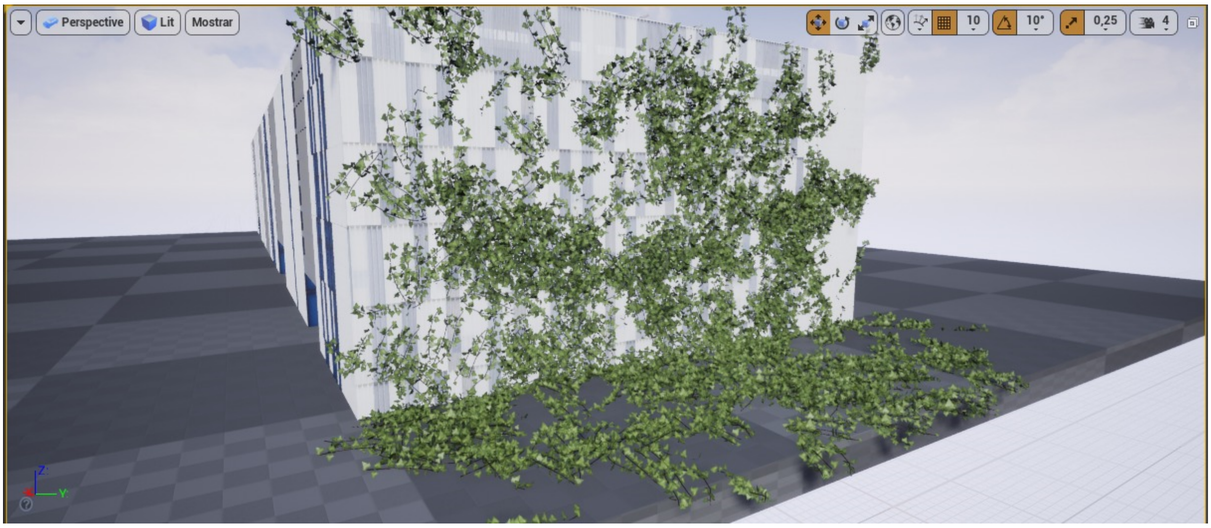
Altura: Min -26214 Max 26214

Instance Settings

Mobility: Static Movable

Distancia selecci: Min 0 Max 0

Figure 6.7: Ivy un UE4 2 (UE4)



Using all the tools used in this process, both for the ivy's and the materials, it will be possible to generate the final version available in the demo, thus seeing the different approaches that have been given to the building.

7

FINAL RESULT IMAGES

Figure 6.8: My own version result (UE4)

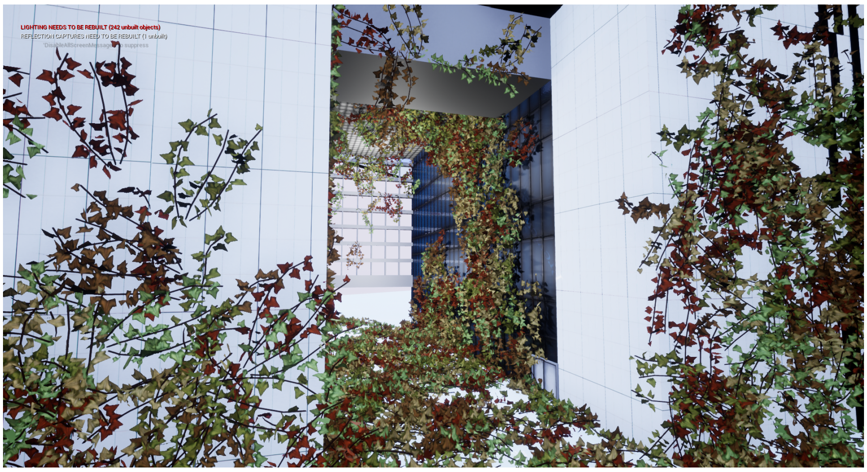
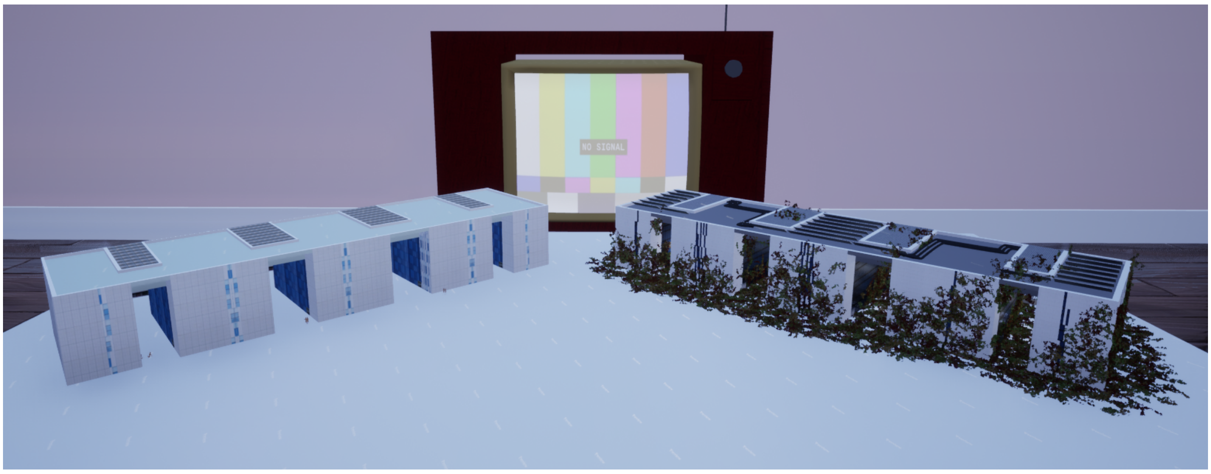

LIGHTING NEEDS TO BE REBUILT (242 unbuilt objects)
REFLECTION CAPTURES NEED TO BE REBUILT (1 unbuilt)
'DisableAllScreenMessages' to suppress

Figure 6.9: Final comparison (UE4)



8

CONCLUSIONS

Thanks to this project I have been able to learn how to use the CityEngine tool. It is a very powerful software with a lot of potential, and very powerful tools like the texture editor. Overall it was good to use CityEngine and to have been able to combine it with Blender and Unreal Engine 4. The comparisons of the CityEngine model and the Blender model are quite revealing. The CityEngine model is going to be implemented in the Smart Campus, so although it had to have a good amount of detail, there was no need for a model as detailed as the one achieved with Blender, which was going to be used in a first-person environment. Overall I think it was very interesting to see the comparison between modelling for a virtual information environment and modelling for a game. I am very happy with the result obtained in the two models, as well as the scene created and the window material that gives a very realistic touch to the building.

When carrying out this project, most of the proposed objectives have been achieved, as the final result has been a model that allows the modernisation of the UJI Smart Campus, using ArcGIS geolocation and learning to use CityEngine, as indicated at the beginning of the project.

As expected, the model of the building created with CityEngine was also implemented in the Unreal Engine 4 video game engine, both with the model imported from CityEngine with a Datasmith and a remodelled model from blender.

When it came to giving it a personal touch, it was also possible to implement it, making a small comparison between the models in the final demo.

The only section where a problem was encountered was the implementation in the Smart Campus, as not having all the necessary files, it was impossible to implement it. However, the building is ready to be implemented, it is correctly modelled and geolocated.

9

BIBLIOGRAPHY

[1] City Engine bug fix
https://community.esri.com/t5/arcgis-cityengine-questions/city-engine-crash/td-p/641299

[2] ArcGis & Smart Campus
https://www.geoexperiencecenter.com/video/como-trabajar-en-un-proyecto-smart-campus-con-la-plataforma-arcgis

[3] CAD referencing https://www.youtube.com/watch?v=T6oXdOU5ZGo

[4] ArcGis CityEngine tutorial and information about how to use the program (all the sections were used or checked for this project)
https://doc.arcgis.com/en/cityengine/latest/tutorials/tutorial-1-essential-skills.htm

[5] Ivy inspiration https://www.youtube.com/watch?v=Z1QIya0xKPc

[6] Glass material tutorial https://www.youtube.com/watch?v=DFJEgD_XBgc

10

TOOLKIT BIBLIOGRAPHY

[1] City Engine
https://www.esri.com/es-es/arcgis/products/arcgis-cityengine/overview

[2] ArcGis & Smart Campus
https://www.arcgis.com/index.html

https://uji.maps.arcgis.com/apps/CEWebViewer/viewer.html?&3dWebScene=e0042c32bd9f4687b914ddc2b4f9035a&view=381.73,-24.8,-113.79,654.3,176.74,237.5,0.95&lyr=1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1

[3] Unreal Engine 4
https://www.unrealengine.com/en-US/

[4] Substance Designer
https://www.adobe.com/products/substance3d-designer.html?gclid=EAIaIQobChMI4tm_8Pv08QIV05rVCh0pRwwrEAAYASAAEgJFCPD_BwE

[5] AutoCAD
https://www.autodesk.es/products/autocad/overview

[6] Blender
https://www.blender.org/

[7] Krita
https://krita.org/es/