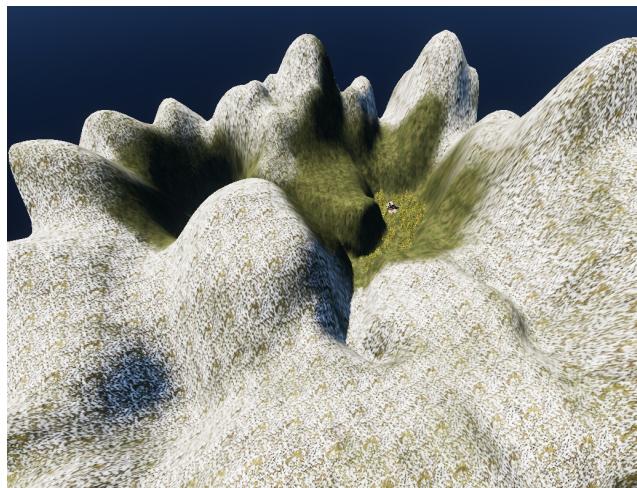


Landscape

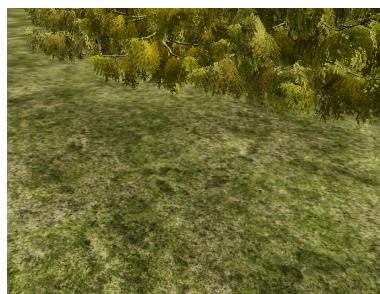
Introduction

In this activity, we will work on several techniques to build a landscape, an enclosed space and the lighting to make both of them look good.

Landscape



We have built a landscape consisting of a group of mountains, a forest area with a cabin in it, a wet area with a lake and a temple (the player's goal in the game) and some paths connecting all of them. There are three kinds of ground surfaces: mossy grass, snowy grass and moist grass. There is a global Material Function for the surfaces and each of them has a Material Function Instance associated and all of them are used by the landscape material.



Mossy grass



Snowy grass



Moist grass

Lighting

There are many elements that compose the lighting of the landscape. Firstly, we have a directional light representing the sun. We wanted the sun to be located more or less to the west, so we gave this light a rotation of $(0^\circ, -64^\circ, 120^\circ)$. There is also a Sky Atmosphere actor providing the sky of our landscape, a Sky Light actor providing the ambient illumination of our scene and an Exponential Height Fog providing a foggy environment.

Lake

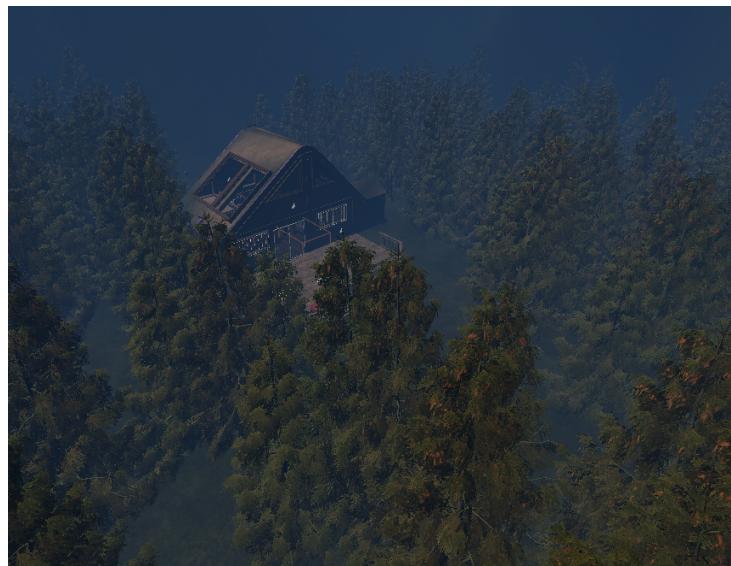
In order to make the moist grass make sense, we created a very simple lake in the corresponding area:



For making the lake, first we added a hole on the landscape by sculpting it downwards. Then, we placed a plane slightly below the height of the ground filling the hole. Then, we assigned a water material to that plane. Basically, it uses a SingleLayerWater shading model and a water normal texture that pans over time to resemble the water of a lake.

Forest

The cabin had to be surrounded by a forest area. We got a plugin from the Unreal Marketplace with some nice foliage assets and created a forest around the cabin with a spruce tree with three different sizes from that plugin. This is how the forest looks like:



Enclosed Space

The enclosed space consists of a room with some furniture. It has interior lighting, some windows the sun passes through, signs of damage, a mirror on a wall, interactable doors and a key that the player needs to grab to solve the mystery in the game.

Lighting

The room also contains a point light inside a desk lamp and post processing volume that applies a color temperature effect (we wanted to produce a warm environment) with a value of 3768 and a bloom effect for that point light. We can see how this effect affects the point light by looking at the following images:



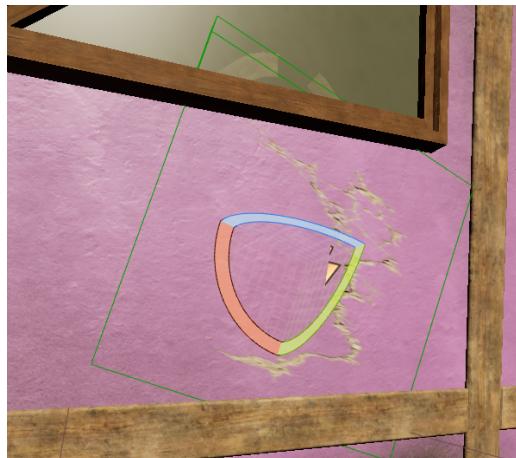
No bloom



Bloom with intensity 2

Deterioration

We have used two types of damage signs for the room: crack decals and a mossy material for the walls and roof.



Crack decals are just decals placed on the outer walls of the room and use a crack texture from Quixel.



We created a global mossy surface material that has two instances: one for the walls and one for the roof. By using the vector painting method and this mossy surface material, we painted some moss on the house to make it look damaged through time.

Mirror

We created a mirror material. We have placed it near the desk where the key we need to grab is located:



The material used by the mirror has a base color of (0.9, 0.9, 0.9), a specular value of 1 and a roughness of 0.1. A “perfect mirror” would have had a base color of pure white and a roughness value of 0, but we wanted to make it look dirty.

Doors

Doors are a problem. They need to be there but at the same time they bother. And we can not remove them or leave them open, because that would not make sense in a narrative point of view. That is why we needed to create a very simple system that detects when the player gets close to a door and opens or closes it as the player gets closer or walks away. As there are two different doors in the room (one of them rotates for opening and the other one slides), we had to create a different Blueprint for each of them. Each BP has an open and a closed position/rotation and contains a volume that detects overlaps with the player. Whenever an overlap with the player has started or ended, the BP transitions accordingly between the open and the closed states. These are the two doors:



The key

There is a key on the desktop that we can (and we have to) grab. It is just a blueprint with a static mesh of a key and a sphere volume with a certain radius, the pick radius, that detects overlaps with the player. When an overlap starts, a hint suggesting the player to grab the key is shown and when that overlap ends the hint is hidden. If the player presses the pick key (specified in the hint) while being inside the pick range, the key reveals the hints to find the secret in the level and then it destroys itself. The key mesh uses a material that mimics gold with a base color of (1,0.77,0.34), a metallic factor of 1 and a specular factor of 0.5.

The hints

Starting in front of the cabin, there is a path that leads to a bifurcation. One of the paths is a dead end and the other leads to the secret in the level. There are two hints that are wooden signs. There is one sign in the bifurcation telling to follow the path on the right and another sign in the dead end suggesting to turn back. These signs, by default, are hidden. When the key is picked, they are made visible. This is how they look like:



The goal

If we complete the game by choosing the correct path in the bifurcation, we will eventually find the lost temple. Next to it, there is a crystal capsule. The moment we enter that capsule, the game will end.



References

Game assets

/ChichenItza/: <https://opengameart.org/content/chichen-itza-el-castillo>

/EF_Sam/: <https://www.unrealengine.com/marketplace/en-US/product/ef-sam>

/Materials/T_Water_N: https://youtu.be/6JFBL_239M

/Meshes/SM_Key: <https://opengameart.org/content/card-suits-keys>

/PN_interactiveSpruceForest/:

<https://www.unrealengine.com/marketplace/en-US/product/interactive-spruce-forest>

/SurvivalWood/:

<https://opengameart.org/content/survival-wood-shackshelter-and-objects>