

Nature Realms: heightening immersion in XR Nature Environments

Technical project report



European Project Semester

Academic year: 2022-2023

Team: Hello Hyperreality (HYPER)

Team members: Víctor Fernandez Moya (ES), Philipp Parteder (AT),
Bavo Debraekeleer (BE), Leire Lisón Nicuesa (ES) and Gabriel Copier (FR)

Clients: Jeroen Cluckers and Lowie Spriet from the Immersive Lab

Project coaches: Hiram Bollaert and Erwin Listhaeghe

EPS coordination team: Geert Marrin and Bie Dirkx

Table of Contents

Introduction.....	4
Technical Project Process.....	5
1 Artificial Intelligence Driven Design (AIDD) Process.....	5
1.1 Generative AI Tools	6
1.2 Tekst-to-Image.....	6
1.2.1 DALL-E 2	6
1.2.2 Midjourney.....	7
1.2.3 Stable Diffusion.....	7
1.2.4 Photosonic	8
1.3 Large Language Models	8
1.3.1 ChatGPT	8
1.3.2 Chatsonic	8
2 Virtual Reality Development Process	9
2.1 Overview.....	9
2.2 The VR Headset: Meta Quest 2	9
2.3 Unity Game Engine.....	10
2.3.1 Universal Render Pipeline (URP)	10
2.3.2 Oculus Integration SDK and Stage Tracking	10
2.4 Photogrammetry and 3D Scanning.....	11
2.4.1 Photogrammetry and Quixel Megascans.....	11
2.4.2 3D Scanning with Smartphone	11
2.5 SpeedTree	12
2.6 Custom 3D with Blender	12
2.7 Audio Design.....	12

Project Results	13
3 Artificial Intelligence Driven Design (AIDD) Results.....	13
3.1 Using AI for Ideation	13
3.2 Coming up with the experiments.....	13
3.2.1 Idea generation with Large Language Models.....	13
3.2.2 Idea visualisation with text-to-image AI tools.....	16
3.3 Visualising the Environments.....	24
3.3.1 Visualising the beach environment.....	24
4 Virtual Reality Development Results	26
4.1 Playthrough Video.....	26
4.2 Scene 1: Office	26
4.3 Scene 2: Forest.....	27
4.3.1 Overview.....	28
4.3.2 Location 1: Lake	31
4.3.3 Location 2: Uphill path	33
4.3.4 Location 3: Hill top	34
4.3.5 Location 4: Cave Entrance	35
4.3.6 Visual Effect: Aurora Borealis	36
4.3.7 Visual Effect: Fireflies	38
4.3.8 Visual Effect: Umbrella Raindrops	40
4.4 Scene 3: Cave	41
4.4.1 General Moodboard in Miro	41
4.4.2 Location 1: The Abyss	42
4.4.3 Location 2: The Tunnel.....	45
4.4.4 Visual Effect: Bioluminescent Mushrooms.....	47
4.5 Scene 4: Beach.....	48
4.5.1 General Moodboard in Miro	48

4.5.2	Overview.....	49
4.5.3	Location 1: The Beach.....	52
4.5.4	Location 2: The Cliff.....	54
4.5.5	Visual Effect: Ocean Water.....	56
4.6	3D Scans	61
4.6.1	Physical Setup	61
4.6.2	Fellow Student.....	61

Introduction

This report gives an overview of the technical aspects of the project that don't fit the final project report itself. It describes the use of Artificial Intelligence in the ideation phase and all the used tools, technology and techniques for the creation of the virtual reality aspect of the project.

Technical Project Process

1 Artificial Intelligence Driven Design (AIDD) Process

Another topic our clients, the Immersive Lab at AP University, are researching is the use of generative artificial intelligence tools in creative processes, or [AI Driven Design \(AIDD\)](#).

How AIDD is used in this project is documented in this section.



Figure 1 Image generated by DALL-E 2

Promt: a header image for documentation about the use of generative ai driven design in a creative process

1.1 Generative AI Tools

There are a lot of different [generative AI](#) tools available online to use.

The most known of which use *text to image* models, eg. DALL-E 2, or *large language models (LLM)*, eg. ChatGPT, but you also have *text to audio*, *text to music*, *text to video*, *real to stylized video* and *image to 3D models*.

Most of these however are closed source and require a paid subscription to be able to use them. Although they often offer free credits that can be used for experimentation and finding the right tool for the job at hand. These often also return after a certain period of time.

Other models are open source and free to use on your own systems or through platforms like Google Colab to offload the processing power needed for the model to run. In the nature of open source projects, these also sprout all kinds of variations and models trained for specific purposes or styles.

Note: all details are as of writing in Q1 2023.

1.2 Tekst-to-Image

Text to image models generate an image from a given prompt.

A prompt can be a full sentence, random words or a mix of both separated by commas, depending on the specific model.

Often there are also more advanced options like:

- Negative prompt: things you don't want in the image.
- Weights: with which you can give heavier weight to words, meaning making some words more important than others.
- Seed: number to control or limit the randomness of the image generation. This is often a random number, but by setting it to a fixed number you can more easily monitor the influences of changing a prompt.

1.2.1 [DALL-E 2](#)

- **Model:** text to image and image expansion
- **License:** closed source
- **Usage:** [website](#) and API
- **Pricing website:** 4 images per credit, 50 free credits, 15 free every month, \$15 for 115.
- **Pricing API:** \$0.02 per 1024×1024 image, \$0.018 per 512×512 image, \$0.016 per 256×256 image. Enterprise team subscriptions possible.

DALL-E 2 is probably one of the easiest and best tools for non experts.

It's a closed source *text to image* model by OpenAI with which you can generate photorealistic and artistic images. It also has some extra feature like expanding existing images or photos beyond their original frame.

[A short explanation is available on their website](#).

It works through a [webpage](#) where you can write a prompt or upload an image, as well as through an API.

1.2.2 [Midjourney](#)

- **Model:** text to image and photo to image
- **License:** closed source
- **Usage:** Discord channel
- **Pricing:** Free trial of 25 min. GPU time, Basic at \$10/month for 200 min/month, Standard at \$30/month for 15h/month, Corporate at \$600/year for 120h/year.

Midjourney is a model created by a researchgroup which generates more artistic images, often concept art style looking. It's still in beta and works through a Discord channel which you can join by clicking the *Join the Beta* button on [their website](#).

The pricing is counted in GPU time the AI model spends generating your images. You get a free trial in the beginning of 25 minutes GPU time which regenerates every month.

To generate images join the Discord channel, go to a *newbies* thread, and write /imagine. A pop-up will show up for this command /imagine prompt:. Select it and write your prompt. This is a shared thread, so all images being generated by other users also show up. You can follow the generation of your image in the command you posted. When it's finished a new message is created with the result mentioning your username.

1.2.3 [Stable Diffusion](#)

- **Model:** text to image and photo to image
- **License:** open source
- **Usage:** website, local, Google Colab Notebook, etc.
- **Pricing:** Free (virtual machine costs may apply.)

[Stable Diffusion](#) is an open source projects. As most popular open source projects it has sprouted a lot of different versions trained for specific tasks and styles. Here are the easiest and most useful ways to use it:

- [Stable Diffusion Playground web interface](#) : this is an easy webinterface similair to DALL-E to demonstrate the model. Just write the prompt and press *Generate image*. There is also a [prompt database](#) available to get ideas of possible prompts.
- [Hugging Face Demo web interface](#) : Hugging Face is an open source AI organization. They also provide a free web interface to demo Stable Diffusion similar to the Stable Diffusion Playground.
- [Google Colab Notebooks](#) : if you just google it you can find a lot of different Google Colab Notebooks created by users to use. This is one example using a specific model, but a lot of other models are available to use as well. You can also couple your Google Drive to a Notebook to auto save the resulting images to it. This approach is for a bit more advanced users, but provides the possibility to change and add code in Python.

- [Dream Studio](#) : is another web interface that features faster generation and makes some parameters available as sliders that are available in code when you use a Notebook.
- [Running locally from GitHub repo](#) : another possibility is cloning the Stable Diffusion repository on GitHub and running it locally on your own system. How to do this is documented in the README of the repo.
- [Blender Plug-in](#) : there is also a plug-in available to use Stable Diffusion within Blender. It's still only 2D image generation, but can be used to for example generate textures and 2D animations. The generation is not run locally, but through an API for which you'll need a key.

1.2.4 [Photosonic](#)

- **Model:** text to image
- **License:** closed source
- **Usage:** website
- **Pricing:** Free Trial 2500 words/month, Long-form \$13-1749/month for 19-5750k words/month. 1 image generation gives you 2 images and costs 100 words.

Photosonic is a tool by Writesonic that offer a range of AI tools. It uses a web interface similar to DALL-E 2. It uses words as it's credits because it also has a chat AI called Chatsonic where this system is easier to understand.

1.3 Large Language Models

1.3.1 [ChatGPT](#)

- **License:** closed source
- **Usage:** [website](#)
- **Pricing:** Free Trial 2500 words/month, Long-form \$13-1749/month for 19-5750k words/month.

A chat AI by OpenAI.

1.3.2 [Chatsonic](#)

- **License:** closed source
- **Usage:** website, Chrome plug-in
- **Pricing:** Free Plan available when demand is low. Plus plan for \$20/month giving access even when demand is high, faster response speed and priority access to new features.

A chat AI by Writesonic.

2 Virtual Reality Development Process

To build the virtual reality aspect of the prototype experience, and collaborate iteratively, a wide variety of tools, technologies and techniques were utilized. This section gives an explanation of the whole process.

2.1 Overview

- VR headset: Meta Quest 2
- Software:
 - Game engine: Unity
 - Software editor: JetBrains Rider / Visual Studio Code
 - Story-, mood- and referenceboards: Miro
 - 3D modeling: Blender
 - 2D editor: Adobe Photoshop
 - Photogrammetry library importer: Quixel Bridge
 - 3D tree creation tool: SpeedTree Modeler
 - Audio editor: Audacity
 - 3D scanning app: Polycam
 - Meta Quest development tools:
 - Meta Quest Developer Hub (Windows)
 - Oculus App (Windows and Android)
 - OVRMetrics (debugging on headset)
- Assets:
 - Unity Asset Store
 - Quixel Megascans photogrammetry library
 - SpeedTree Models
 - Audio from Freesound
 - 3D assets from Sketchfab
 - HDRI maps for skyboxes from Poly Haven and HDRI Haven
 - Visual Tween path editor (Unity plugin)
 - Oculus Integration SDK and examples

2.2 The VR Headset: Meta Quest 2

The VR headset used for this project is the Meta Quest 2, previously named the Oculus Quest 2. It was chosen for its accessibility, previous experience by the team, and above all its mobility. Most headsets run on a desktop computer or gaming console which they are attached to by cable. They can deliver better performance and graphical quality compared to the mobile Android hardware of the Meta Quest 2, but the cable limits the user's mobility and can be a distraction that breaks immersion. This headset was provided by the clients of the Immersive Lab.

2.3 Unity Game Engine

The Unity game engine was chosen as the main development tool because of the experience of the team using it and the vast developer community around the engine.

2.3.1 Universal Render Pipeline (URP)

Within Unity, the Universal Render Pipeline was chosen for its great performance on mobile hardware and its flexibility in render settings and custom shader capabilities. For a couple of features in the environments build for the prototype experience of this project the Shader Graph was used. These features are highlighted in the Project Results chapter.

2.3.2 Oculus Integration SDK and Stage Tracking

Because the experience was developed solely for the Meta Quest 2 and to be able to use experimental features of the headset the Oculus Integration SDK was used within Unity. One of the things it provides which makes development easier is a tool that checks if all settings are set up correctly for any Quest headsets.

An experimental feature it provides that was used in this project is Stage Tracking. Standard tracking does not keep track of the user's position in the virtual world relative to your real-world surroundings. Only the headset's position relative to the boundary. For example, if you take the headset off, turn around and put it back the virtual world will be aligned the same in your view, meaning its alignment towards the boundary and the real world has changed. When using a physical setup that needs to match the virtual world this is not wanted. Stage tracking provides a better solution. It requires a Room Setup as well as a boundary. The Room Setup works by highlighting the walls around the playing area to make a closed space (they don't have to be the actual walls of the room). With this the headset can track the orientation of the room and keep the orientation relative to the virtual world. However, it wasn't designed to perfectly match the virtual world to the real world. Rather to bring elements of your room into an mixed reality experience. For example, depending on the direction you are facing when starting an application, the virtual world can be rotated by increments of 90°. To fix this a new boundary has to be drawn. Another example is that your starting position, or virtual world origin point, as set by calculating the centre point of the boundary rather than that of the room. This means the boundary has to drawn almost exactly the same way every time or the physical elements in the experience won't align correctly with their virtual representations. The tracking is also still in experimental face resulting in occasional slight shifts of the virtual world. For example, the floor level getting raised and lowered, mostly after coming out of standby mode when it needs to recover the tracking. But

this can also occur while drawing the boundary.

A another downside of using Stage Tracking is that it can't be used simultaneously with Hand Tracking. Hand tracking can help with immersion and be more intuitive, especially for people who don't like controllers, but it has its own technical problems and limitations.

2.4 Photogrammetry and 3D Scanning

The goal of the project is to give users the experience of actually being in nature. That is why the focus of the visuals is hyper-realism. The headset's mobile hardware has many limitations, but with photogrammetry and the right texture density, a lot can be achieved.

2.4.1 Photogrammetry and Quixel Megascans

Photogrammetry is the technique of scanning real-world objects to be used in digital visualization. It can be scans of whole environments and everything in it, separate objects and persons, or only a specific surface. The goal is to capture all the detail and random nuances the real world has to offer. Something digital tools try to replicate, but never quite succeed at. A great source of photogrammetry assets is the Quixel Megascans library. They have a vast amount of surfaces, 3D assets and 3D plants available. With their Quixel Bridge software and accompanying plugin for Unity it is very easy to get the assets in the engine. Special care does need to be taken with setting up the export settings. The correct map names and channels need to be set for the Materials and Terrain Surface to be used for texture exporting. For example, the Universal Render Pipeline (URP) Lit material uses a Base (Albedo) map, while the Terrain Surface uses a Diffuse map. Very similar, but not quite the same. The Unity documentation can be consulted for the correct setup.

For the 3D model export settings, it is best to check the individual triangle counts of every available LOD level on a per-asset basis to choose the right ones for the project's needs. It is recommended by Meta that the triangle count in your camera's view on the Meta Quest 2 should not exceed 500,000 to 1,000,000 triangles at any given time. The LOD system can greatly help with this, though every extra LOD will increase application size and load times, so they must be chosen with great consideration.

2.4.2 3D Scanning with Smartphone

Custom 3D scans were also used in the project. The physical setup was scanned and imported into Unity to be able to match the virtual world to the real world more closely. This was done with the Polycam app on an iPhone utilizing its LiDAR scanner for better results.

A fellow EPS student was also scanned this way and put in the experience as an easter egg for other students to find during the user tests. Sadly the texture quality is quite low due to technical limitations, but still, a fun addition for creating more enthusiasm and willingness among fellow EPS students to participate in user testing.

2.5 SpeedTree

SpeedTree is a long-time industry standard for creating convincing digital trees. This results in very easy integration in Unity. A SpeedTree model can have LOD's and Wind animation generated and set up in the SpeedTree Modeler. Next, it simply has to be exported as a SpeedTree file for Unity and when imported into Unity it automatically has the LOD system working and the wind animations work with Unity's wind system. The Trees aren't as realistic looking as photogrammetry assets, but they are convincing enough and very easy to use, especially for the wind animations. Quixel Megascans does have a similar system, though since they were bought by Unreal this only works with Unreal Engine. SpeedTree on the other hand got bought by Unity so in the future it might be improved to compete more closely with its Unreal counterpart.

2.6 Custom 3D with Blender

For very specific objects, or objects good 3D models couldn't be found for, custom models were developed. The software of choice is Blender, a free and open-source 3D modelling software package. More on this in the Project Results chapter.

2.7 Audio Design

The audio design is a very important aspect to make virtual environments convincing. Unity offers a 3D audio system that works very well out of the box.

The process of creating the audio design was to first decide what sounds fit the specific scene and list them up. Next sounds were gathered from Freesound, a free online sounds library, and edited down using Audacity to get the right part and/or length. Most of the sounds also need to be able to loop. This is because the user can stay in the same spot as long as they want while the audio needs to keep going without noticeable stops or cuts. It also greatly helps with reducing the application file size and scene load times. Unity also doesn't support every file type, so audio needs to be exported preferably in the .wav file type. Once imported into Unity the sounds can be placed in the scenes with 3D sound turned on for the best effect.

Project Results

3 Artificial Intelligence Driven Design (AIDD) Results

3.1 Using AI for Ideation

Generative AI tools can be very handy in the ideation phase of creative work.

It can be viewed as an extra thinker on the team coming up with ideas depending on the topic or description you give it.

It can also be an easy and fast tool for idea visualisation. The result it gives you can be very different from what you expected to get or what you had in your mind, but that is part of the creative spark it can provide.

Here it's mostly the text to image generation that can provide the most value.

To come up with innovative ideas the LLM chat AI's are less useful, because they mostly reproduce existing ideas.

3.2 Coming up with the experiments

The first thing that had to be decided on this project is what experiments would be conducted. For this we first did individual brainstorming to then bring those ideas together in a team brainstorm.

Here was an opportunity to use generative AI for both coming up with ideas as well as visualising ideas to be able to present them more clearly to the group.

3.2.1 Idea generation with Large Language Models

Because to use the text to image generators you should already have some idea of what you're looking for you can start with an LLM like ChatGPT or Chatsonic.

Below are some of the questions and results from Chatsonic.

Q: Give some research ideas to make a virtual reality experience in nature as immersive as possible.

A: To make a virtual reality experience in nature as immersive as possible, there are a few research ideas that could be explored. First, one could explore using artificial intelligence and machine learning techniques to develop algorithms that can accurately map and recreate the user's movements and environment within the virtual reality experience. Additionally, one could research methods of **enhancing the sensory experience of the virtual environment**, such as adding soundscapes, adjusting lighting, and developing more realistic textures and colors. Finally, one could explore the use of haptic technology to provide tactile feedback to the user and increase the realism of the experience. By researching and exploring these ideas, one could create a more immersive and realistic virtual reality experience in nature.

Q: Give some ideas around creating more immersion in a virtual reality nature environment by using physical objects.

A: To create more immersion in a virtual reality nature environment, one could utilize physical objects. For example, one could **use smells and fragrances to provide an olfactory experience** of the environment. Additionally, one could **incorporate wind, mist, and other air-based sensations** to further immerse the user in the experience. Furthermore, one could use tactile objects such as rocks, trees, and other objects to give the user a more realistic physical experience of the environment. Finally, one could use haptic technology to provide tactile feedback to the user, such as the feeling of grass or water when the user touches an object. By exploring and utilizing these ideas, one could create a more immersive and realistic virtual reality experience in nature.

Q: What are non obvious things that heighten the immersion in virtual reality.

A: To heighten the immersion in a virtual reality experience, one could incorporate several non-obvious elements. For example, one could use sound design to bring the environment to life, such as using ambient noises and soundscapes to create a realistic atmosphere. Additionally, one could use **3D audio** to create a more realistic sound experience, such as using positional audio to give the user the feeling of being close to an object or a creature. Furthermore, one could incorporate **subtle movement, such as swaying trees, water ripples, and other small movements**, to make the environment feel more alive. Finally, one could utilize visual effects such as depth of field and motion blur to create a more realistic experience. By exploring and utilizing these ideas, one could create a more immersive and realistic virtual reality experience.

Q: Give me a list of sensory contrast you find in nature.

A: Nature provides a plethora of sensory contrasts that can be found in any environment. From the way wind passes through the trees and **rustles the leaves, to the way fire crackles and pops, to the sound of thunder rolling through the sky**, each of these contrast provide an interesting and unique experience. Furthermore, the **contrast between hot and cold, soft and hard, light and dark, and wet and dry** can all be observed in nature. **The sight of a sunrise or sunset, the feeling of the sun's warmth on a cold day, and the taste of a fresh rain shower in the summer** are just some of the many sensory contrasts that nature has to offer.

Q: Give me some ideas an electronic sensor can bring to a virtual reality experience of nature.

A: Virtual reality offers users the ability to explore and experience settings that are not available in the physical world. With the implementation of electronic sensors, the virtual

reality experience of nature can be enhanced in many ways.

One way is to use sensors to detect the user's presence in the virtual environment and adjust the environment accordingly. For example, the user's location could be tracked in the virtual environment and trees and other elements of the landscape can be adjusted to create a more immersive experience. If the user is in a forest, for example, trees could be added to the surrounding area and other elements of the environment could be adjusted based on the user's location.

Another way is to use sensors to detect the user's movements and adjust the environment accordingly. For example, as the user moves, the trees and other elements of the landscape could move and change in the virtual environment. The user's movements could also create ripples in a virtual pond or lake, which could be used to create a more realistic and immersive experience.

Finally, sensors can be used to detect temperature, humidity, and other atmospheric conditions of the virtual environment and adjust accordingly. For example, if the user is in a virtual forest and the temperature of the environment changes, the trees and other elements of the landscape could be adjusted based on the new conditions. This could allow for a more realistic and immersive experience for the user.

Follow up Q: What about touch or pressure sensors?

A: Touch or pressure sensors can also be used to enhance the virtual reality experience of nature. These sensors can detect a user's movements and detect changes in the pressure exerted by the user's body. **This could be used to create a realistic experience of walking on grass, through a forest, or on a beach.** These sensors could also detect the user's hand movements, allowing for a more interactive and realistic experience when manipulating objects in the virtual environment. For example, if the user was picking flowers, the sensor could detect the user's hand position and the virtual flowers could be adjusted accordingly. Additionally, the sensors could detect the user's movements and move objects in the virtual environment to ensure a realistic and immersive experience.

Q: Give me some types of sensors to detect touch.

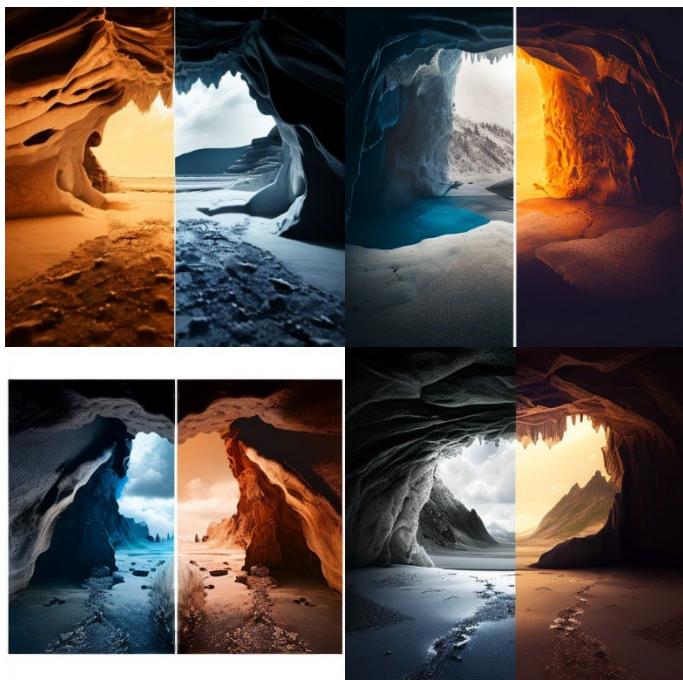
A: There are a variety of types of sensors that can be used to detect touch. One type of sensor that is commonly used is a **force-sensitive resistor (FSR)**. FSRs are resistors that increase their resistance when a force is applied to them. Additionally, **capacitive sensors** can also be used, which measure changes in capacitance when touched. Furthermore, **piezoelectric sensors** can be used to measure the pressure applied to them. Finally, **strain gauges** can also be used to measure the force applied to them. By utilizing these sensors, one can create a more immersive and interactive virtual reality experience with touch detection.

3.2.2 Idea visualisation with text-to-image AI tools

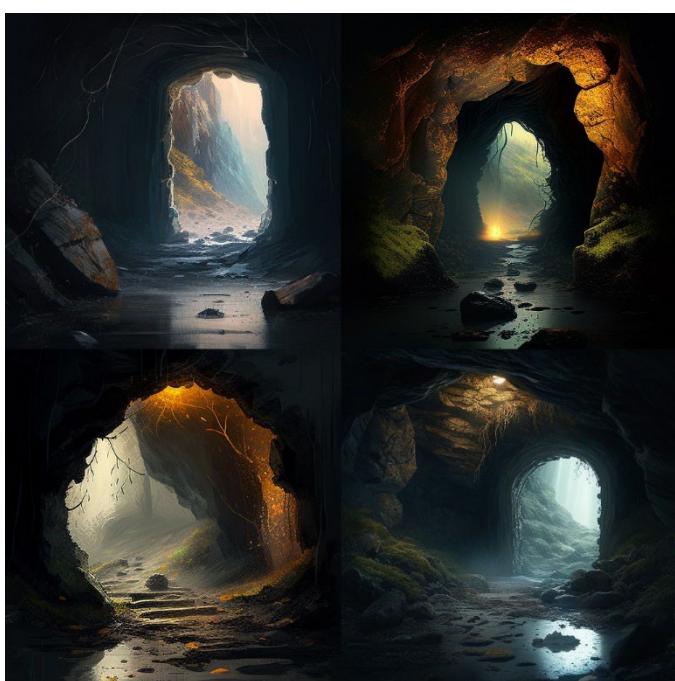
Once you have a clearer general direction to be able to write prompts for a text to image generator you can use visualisations of ideas to spark new ideas and/or present them to others. The following are examples of the resulting images that were used as inspiration.

3.2.2.1 Midjourney

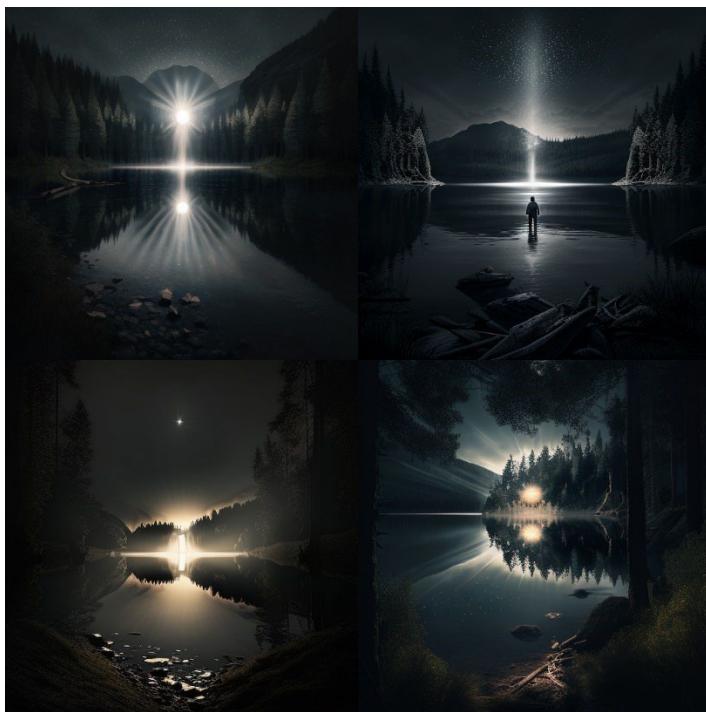
Contrasts in nature



Tight spaces

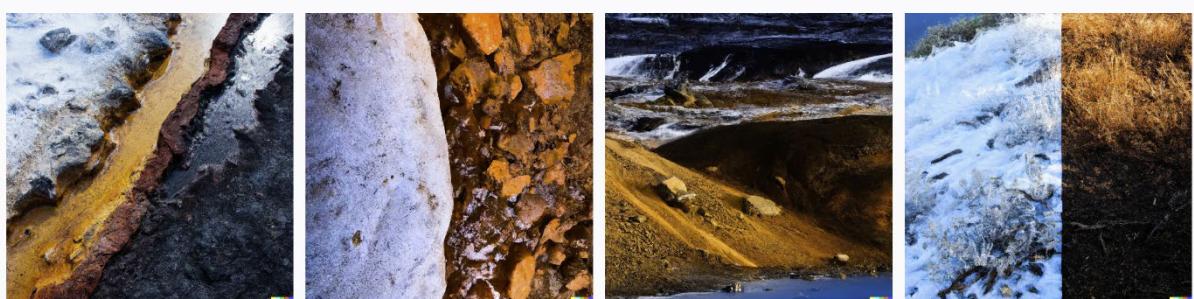


Light in the dark

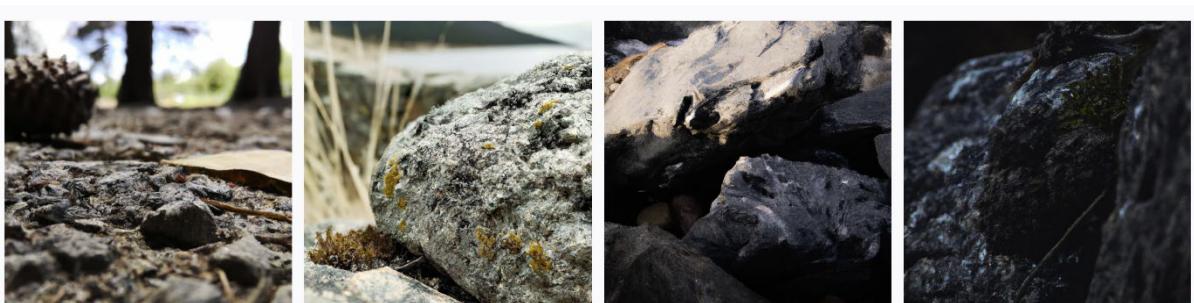


3.2.2.2 DALL-E 2

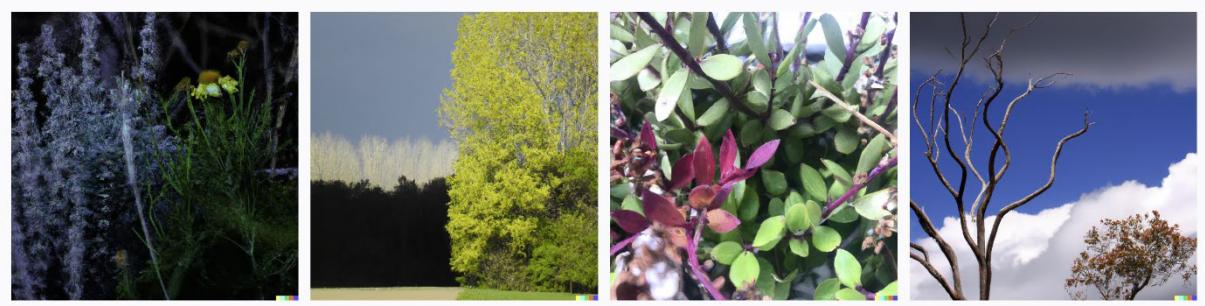
Prompt: contrast between hot and cold, soft and hard, light and dark, and wet and dry in nature



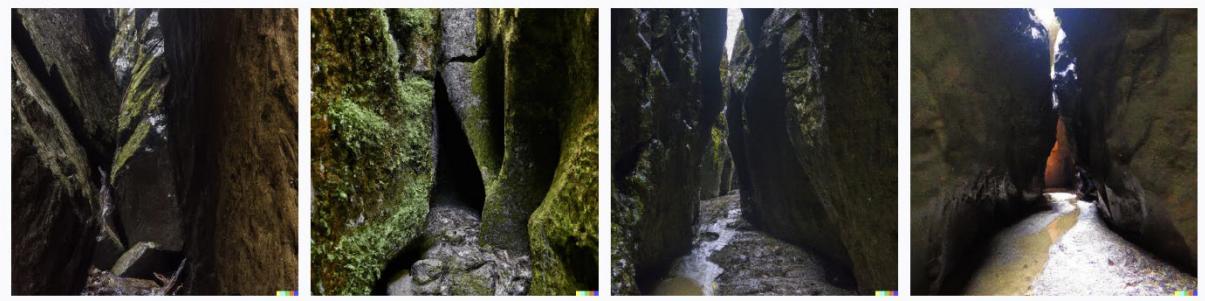
Prompt: contrast hard and soft in nature



Prompt: contrasts in nature



Prompt: damp cave narrow passage open air chasm canyon



Prompt: damp cave narrow passage



Prompt: damp cave



Prompt: narrow space in nature cold dry



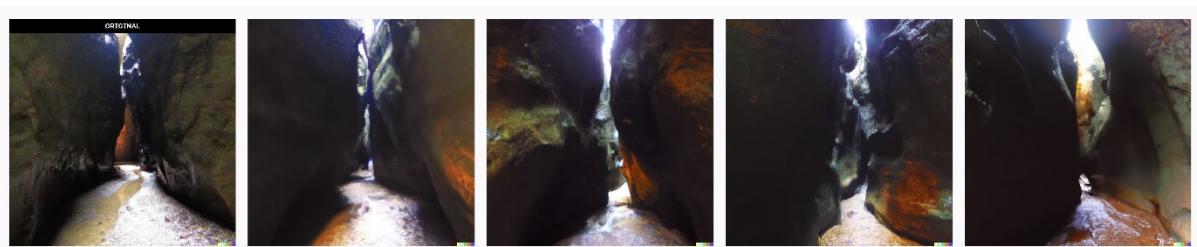
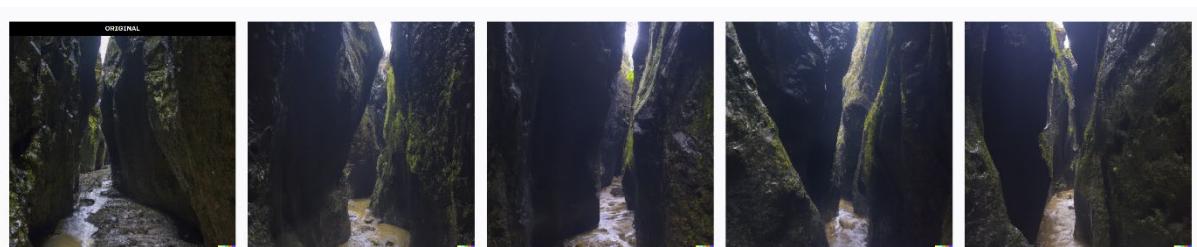
Prompt: narrow space in nature hot and dry



Prompt: narrow space in nature



Prompt: narrow space variations





Prompt: sensory contrasts in nature

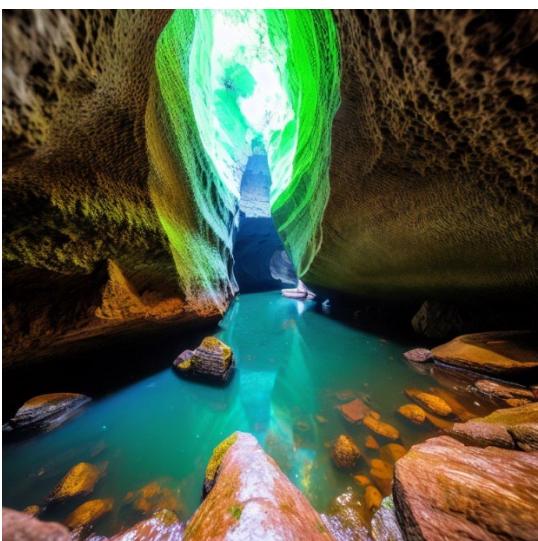


Prompt: the taste of a fresh rain shower in the summer



3.2.2.3 Stable Diffusion

Prompt: very narrow thight cave nature



Prompt: contrasts in nature



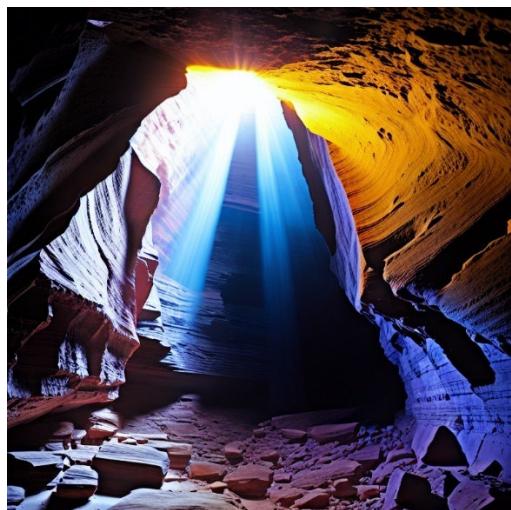
Prompt: sensory contrasts in nature



Prompt: hyper reality nature



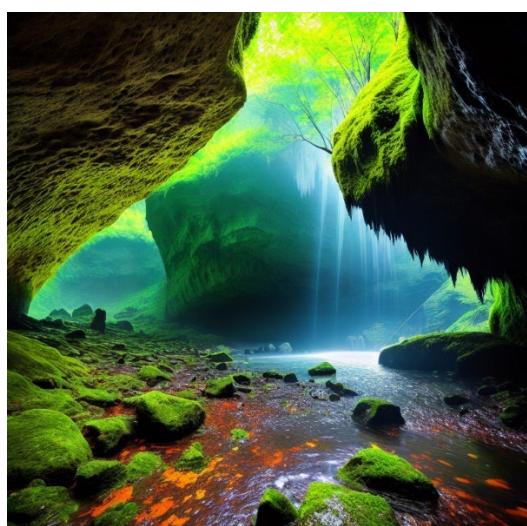
Prompt: dark cave shining light high contrast narrow walls closing in chasm canyon



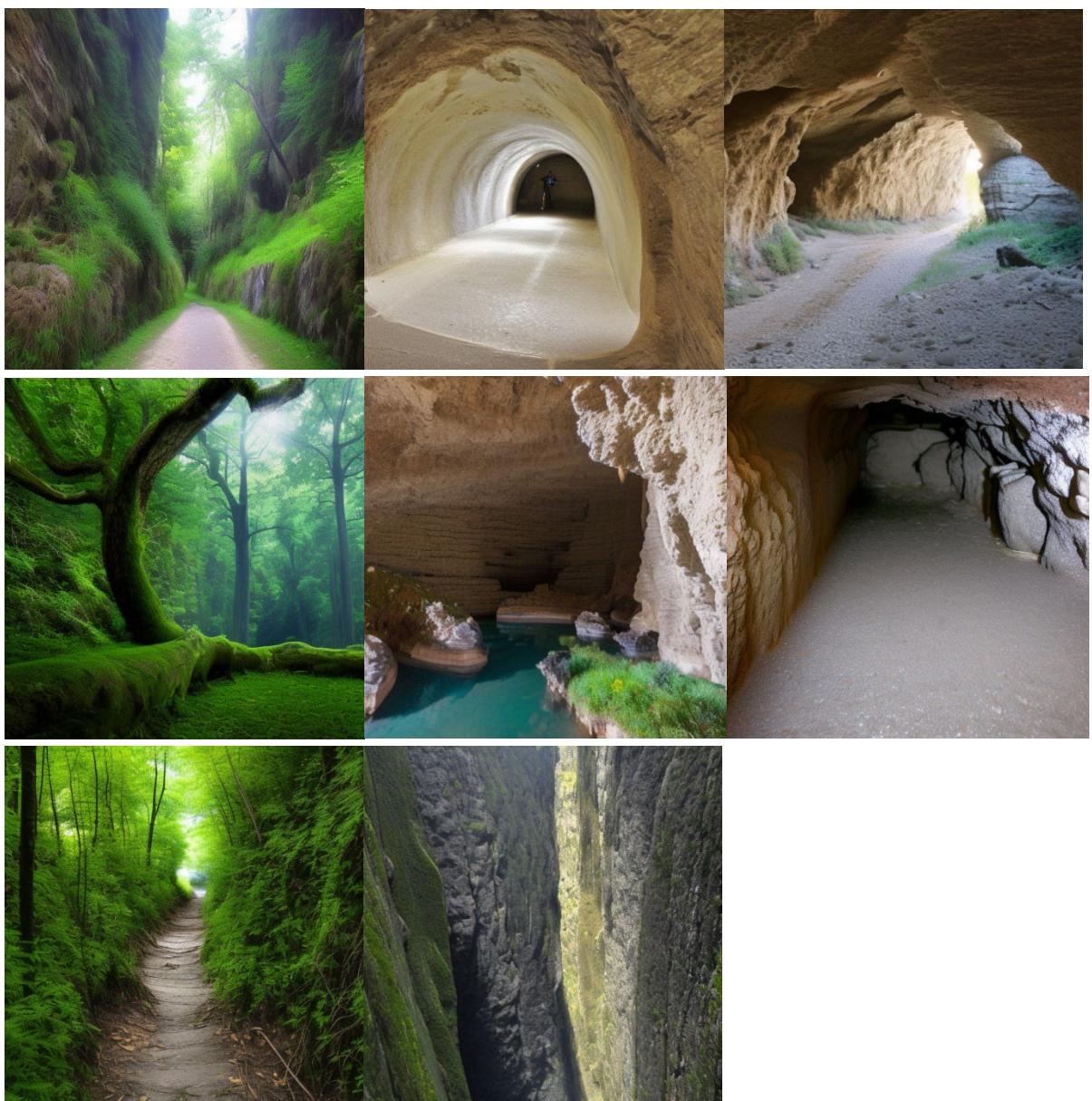
Prompt: wet dark cave shining light high contrast narrow walls closing in chasm canyon



Prompt: damp cave nature

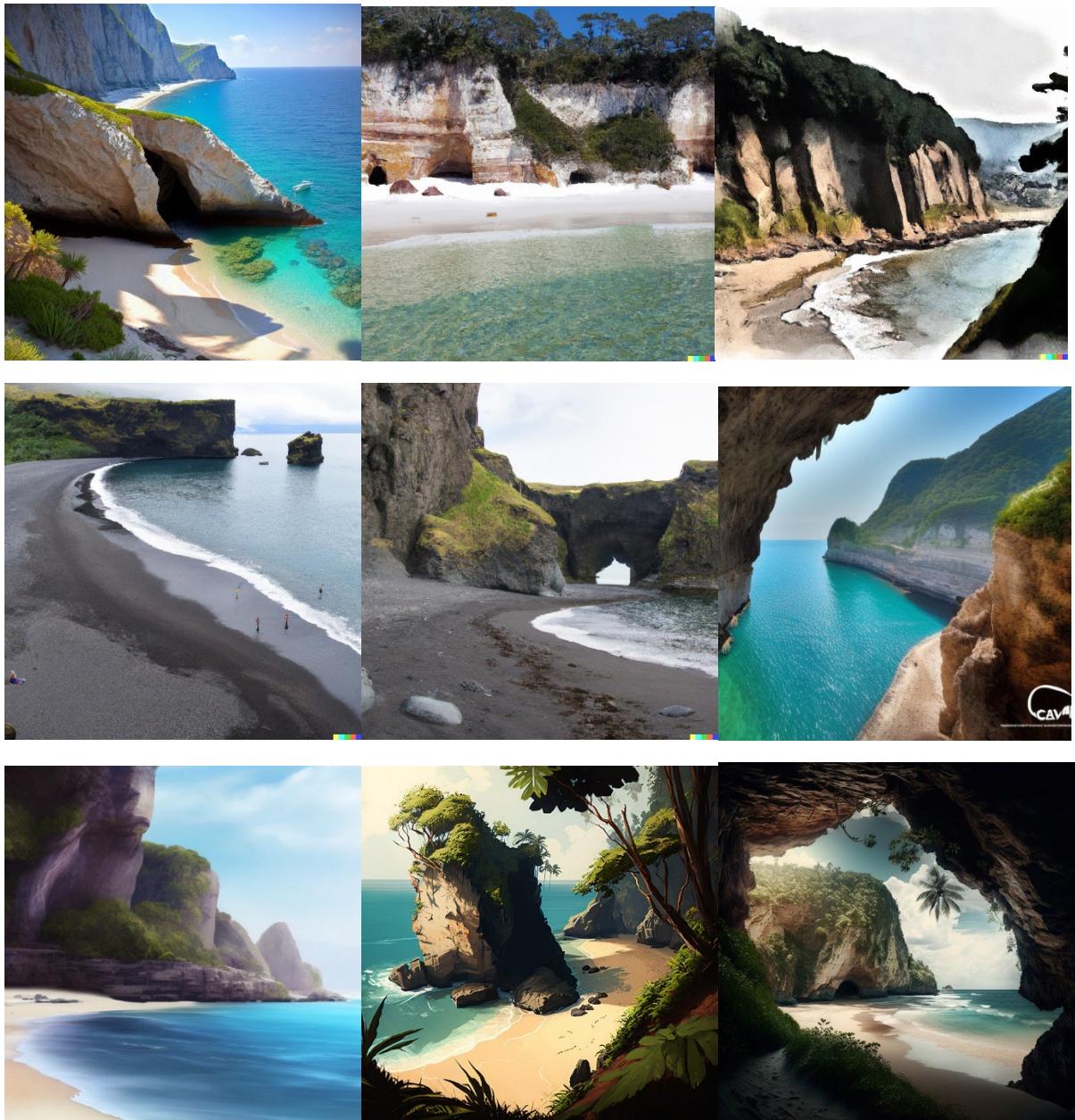


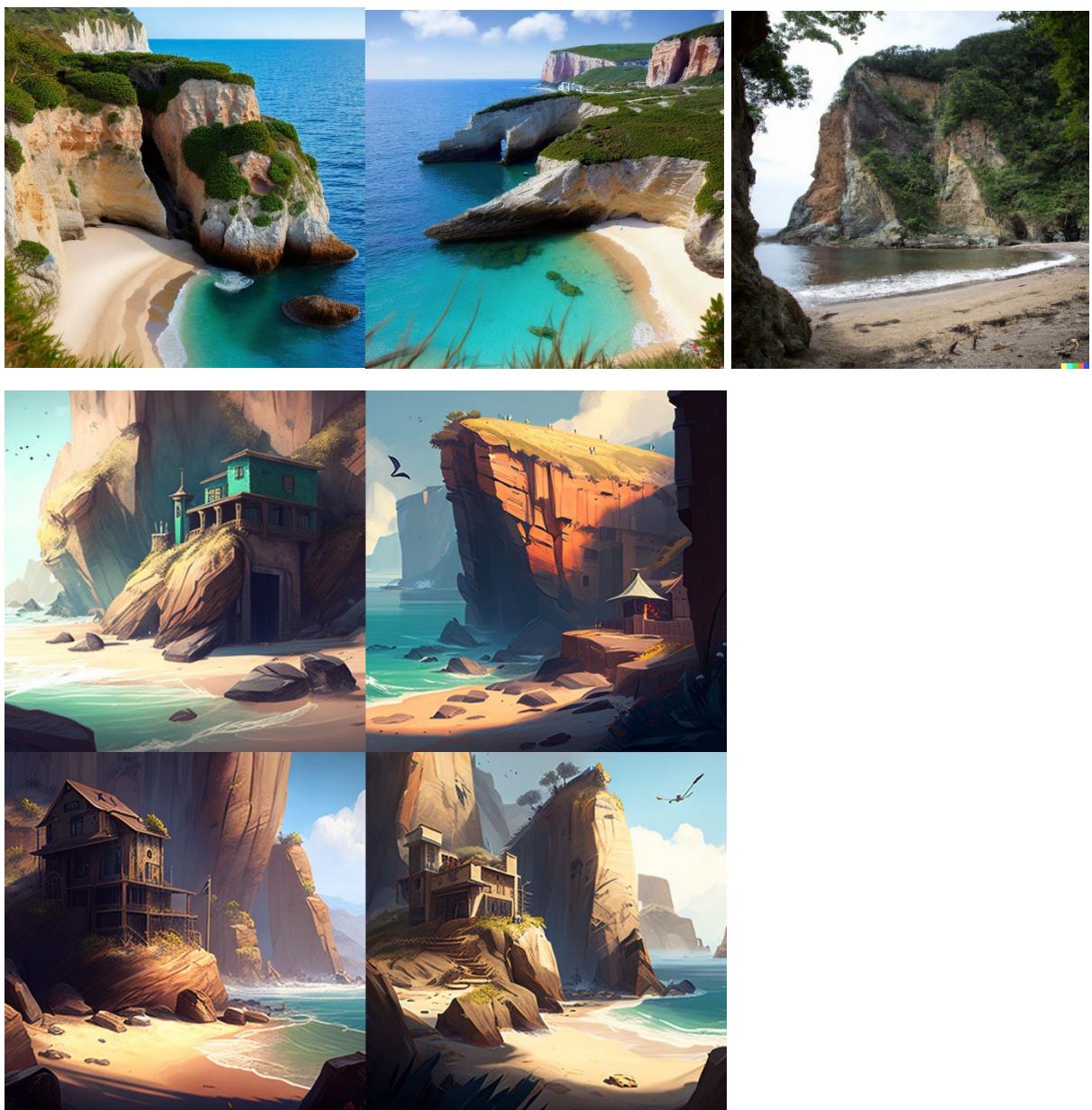
3.2.2.4 Photosonic



3.3 Visualising the Environments

3.3.1 Visualising the beach environment





4 Virtual Reality Development Results

To be able to do test the goals set for the project a custom virtual environment needed to be created. Out of the first user test we concluded that in order to be really immersed the user needed some sort of task or story that lasted long enough. If it is too short the user gets bored and the immersion breaks. It was already decided to build three different biomes to test different things. A Forest, Cave and Beach. So a story was constructed around these. As a starting scene an office environment was added, so the user could then, when ready, escape to nature. It follows a simple story structure where you journey through an environment that gets darker and darker and closes in around you to heighten the tension, to end with the relief of escaping the darkness and ending on the beach with a beautiful sunrise.

4.1 Playthrough Video

[Here](#) is a video of a complete playthrough of the experience as it was at the final user test.

4.2 Scene 1: Office

The office was added to give the user time to be set up with the headset and get used to the virtual reality. The user sits down at their desk and when ready they can press a button to escape to nature.

To build this scene assets from a low poly office asset pack were used together with custom 3D models made with Blender. Some Quixel Megascans materials were used to make surfaces look more convincing and real.



4.3 Scene 2: Forest

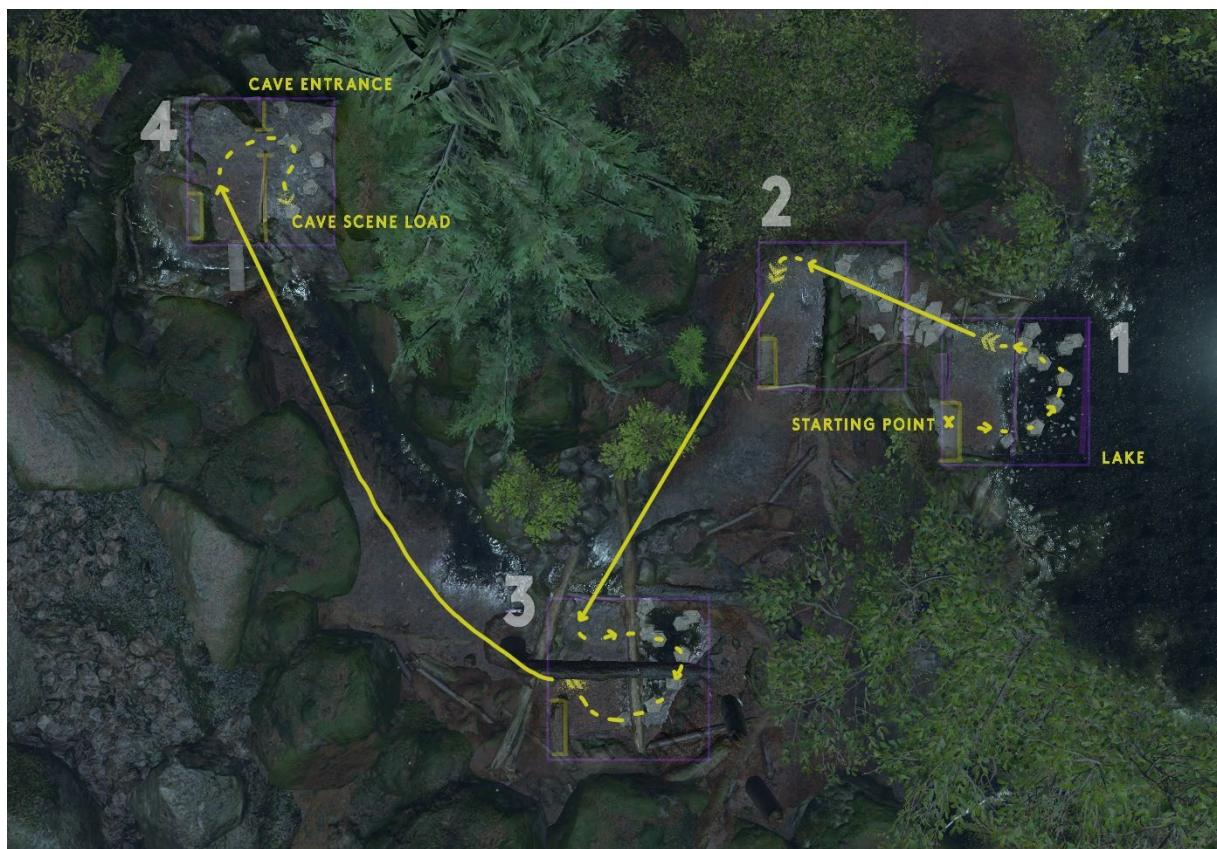
This scene consists of four different locations the user travels through. All four locations include the whole 4x4 meter physical setup.

To transition from the office to nature you start out sitting in front of a lake with a wide open view of where you can enjoy the northern lights. A phenomenon you normally have to travel

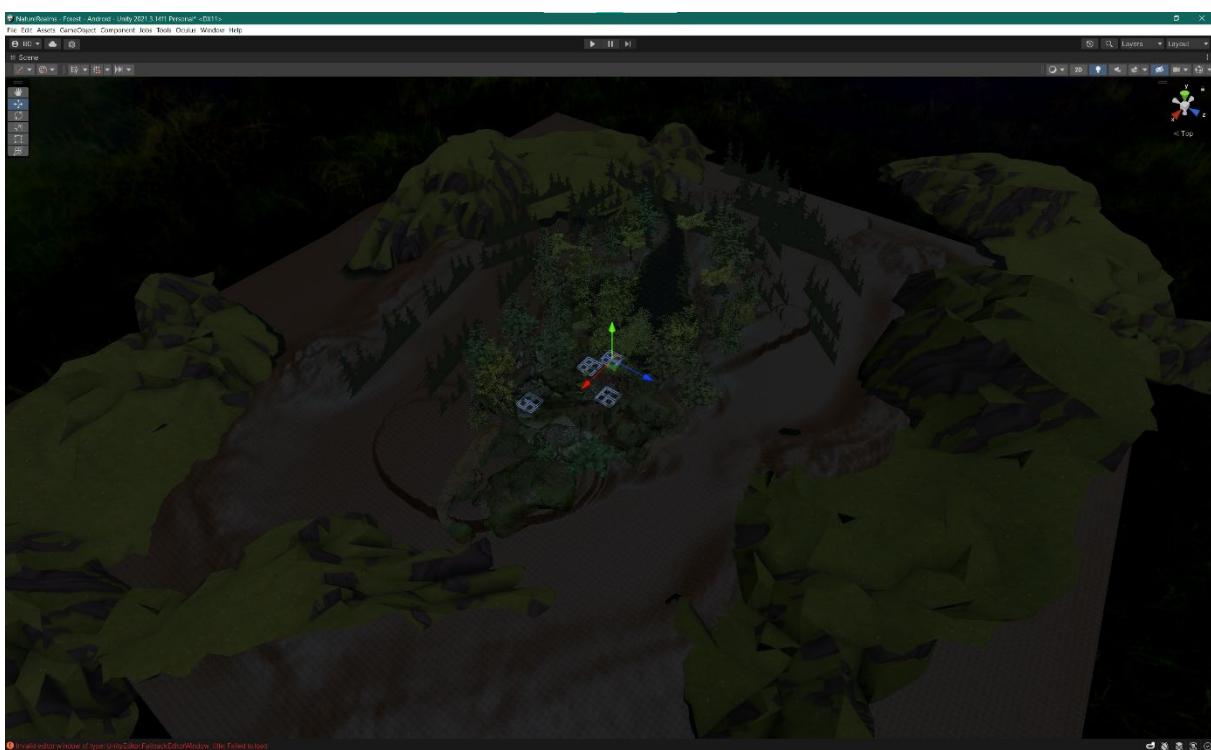
to Scandinavia for to be able to experience it. And even then you might be unlucky and don't get to see it. With a flashlight and umbrella in hand, represented by the Meta Quest 2's controllers, you travel through to forest to the Cave. Along the way there are fireflies that show the way. There are teleportation points on the ground indicated with an arrow and a beam of fireflies that show where the user will end up.

To build the scene mainly 3D assets from Quixel Megascans were used, together with tree models from SpeedTree with wind animations, and Unity's terrain tool with textures from the Quixel Megascans library. For the cave entrance, guidance arrows and aurora borealis effect, custom 3D models were made in Blender.

4.3.1 Overview





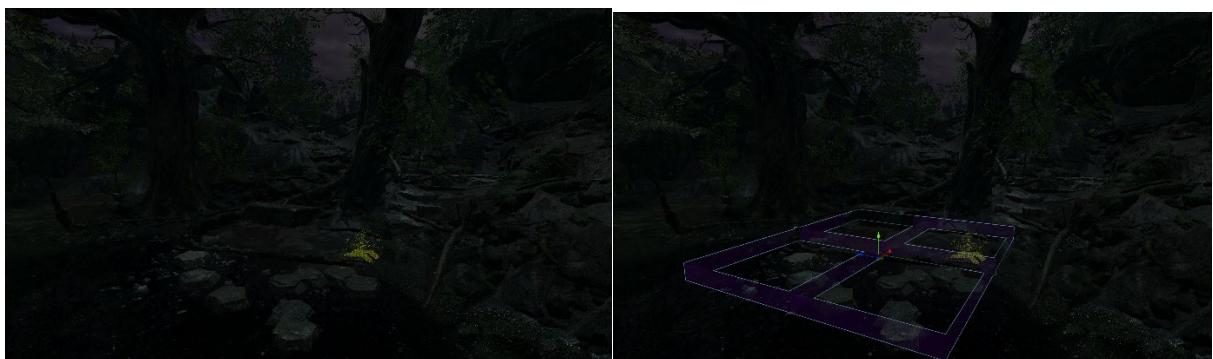


4.3.2 Location 1: Lake

4.3.2.1 Moodboard in Miro



4.3.2.2 Screenshots



4.3.3 Location 2: Uphill path

4.3.3.1 Moodboard in Miro



4.3.3.2 Screenshots



4.3.4 Location 3: Hill top

4.3.4.1 Moodboard in Miro



4.3.4.2 Screenshots



4.3.5 Location 4: Cave Entrance

4.3.5.1 Moodboard in Miro



4.3.5.2 Screenshots



4.3.6 Visual Effect: Aurora Borealis

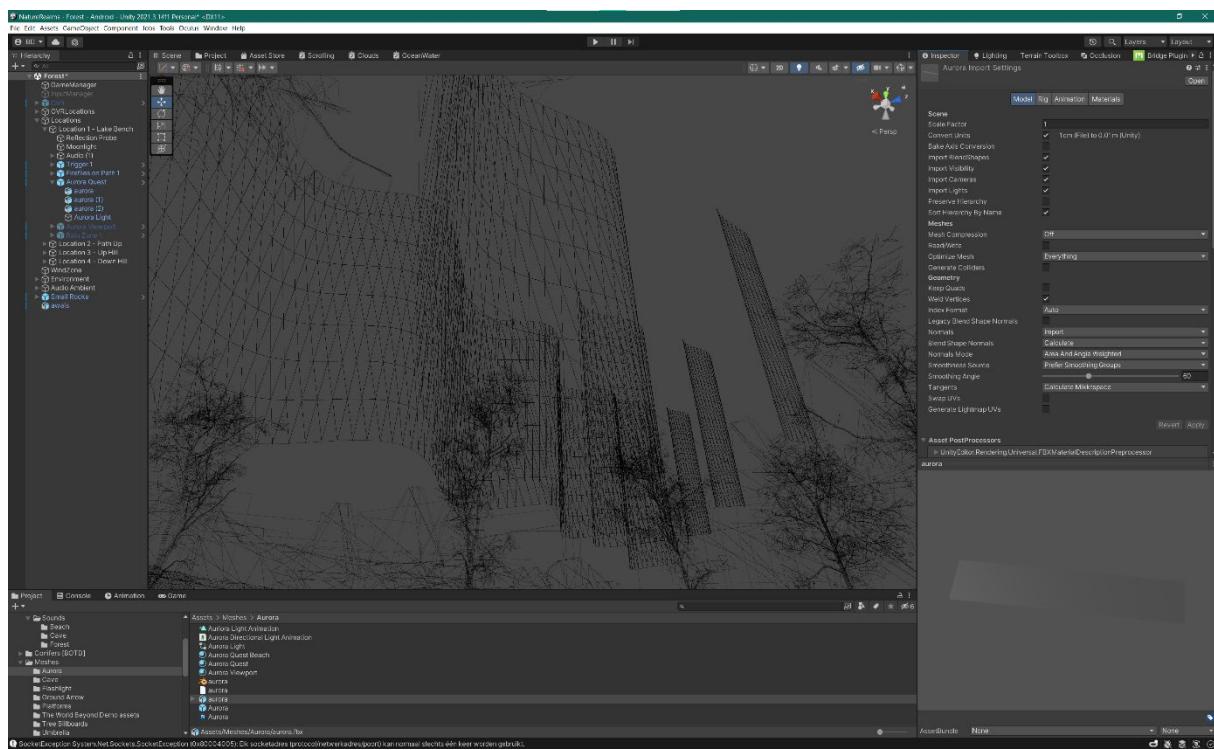
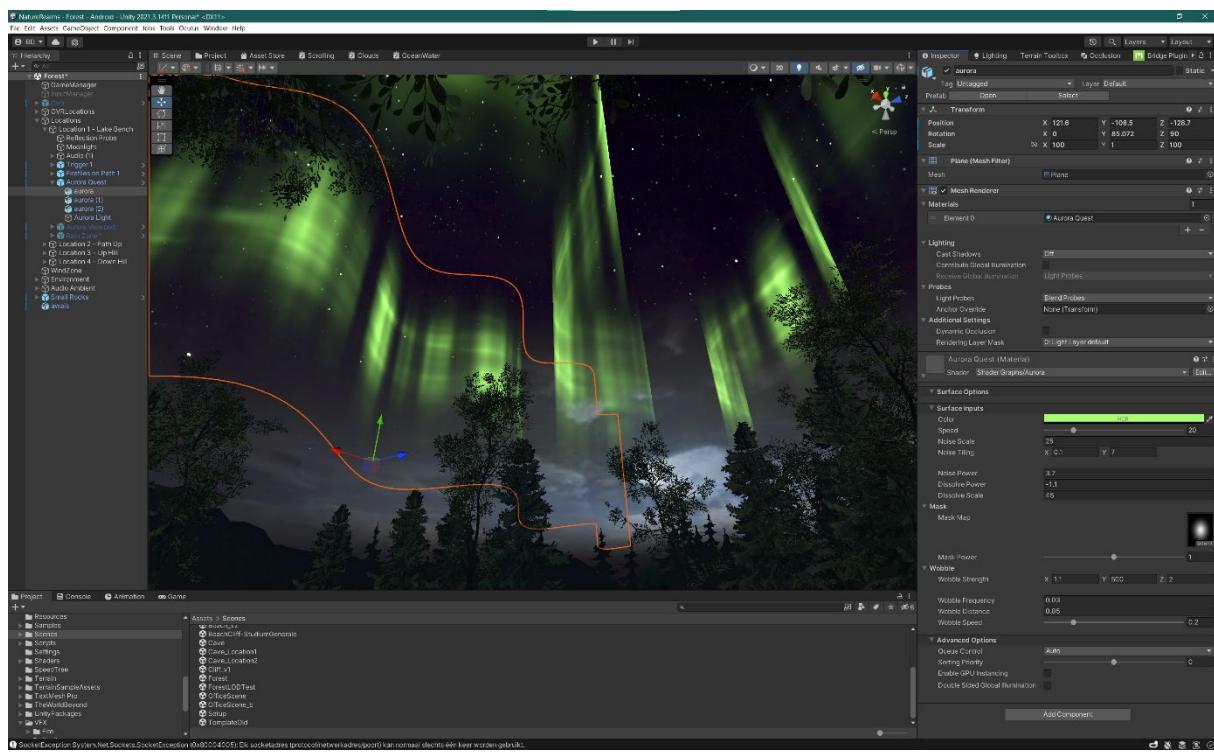
For the Aurora Borealis effect a custom 3D model was created as well as a custom shader with ShaderGraph and URP.

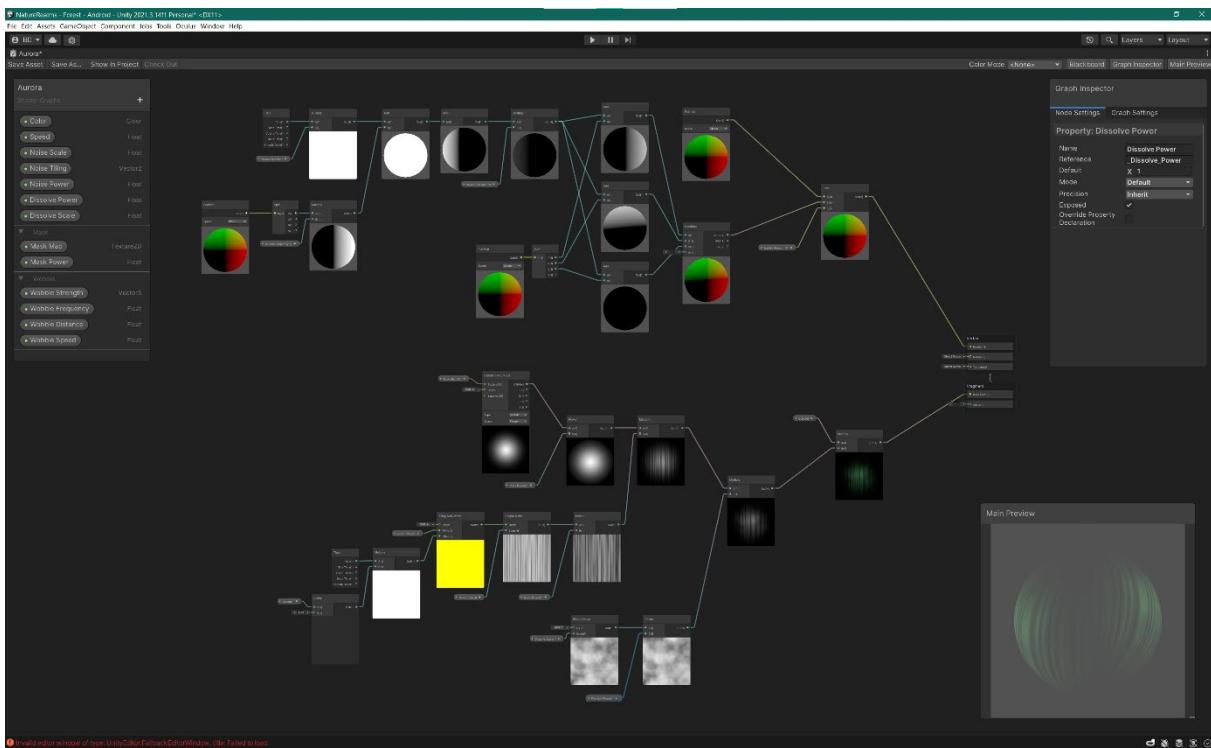
4.3.6.1 Referenceboard in Miro



4.3.6.2 Screenshots







4.3.7 Visual Effect: Fireflies

The guidance arrow and fireflies are a combination of a couple of things.

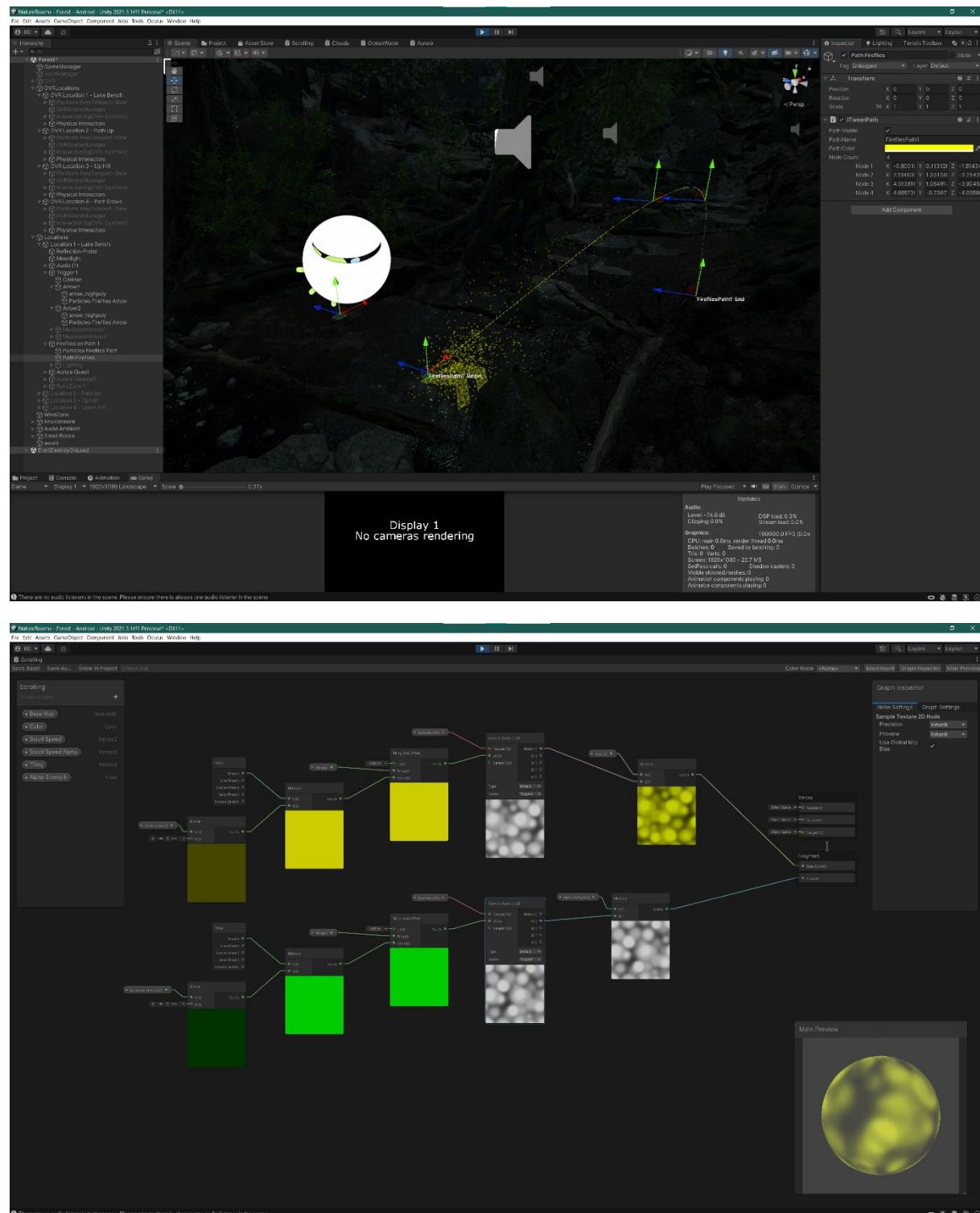
The guidance arrow is made of a custom 3D model with a custom shader build in ShaderGraph and a particles system.

To create the beam of fireflies to the next location the same particle system is used, although with slightly different settings, which are animated to follow a path set up with the Visual iTween path editor and an additional script.

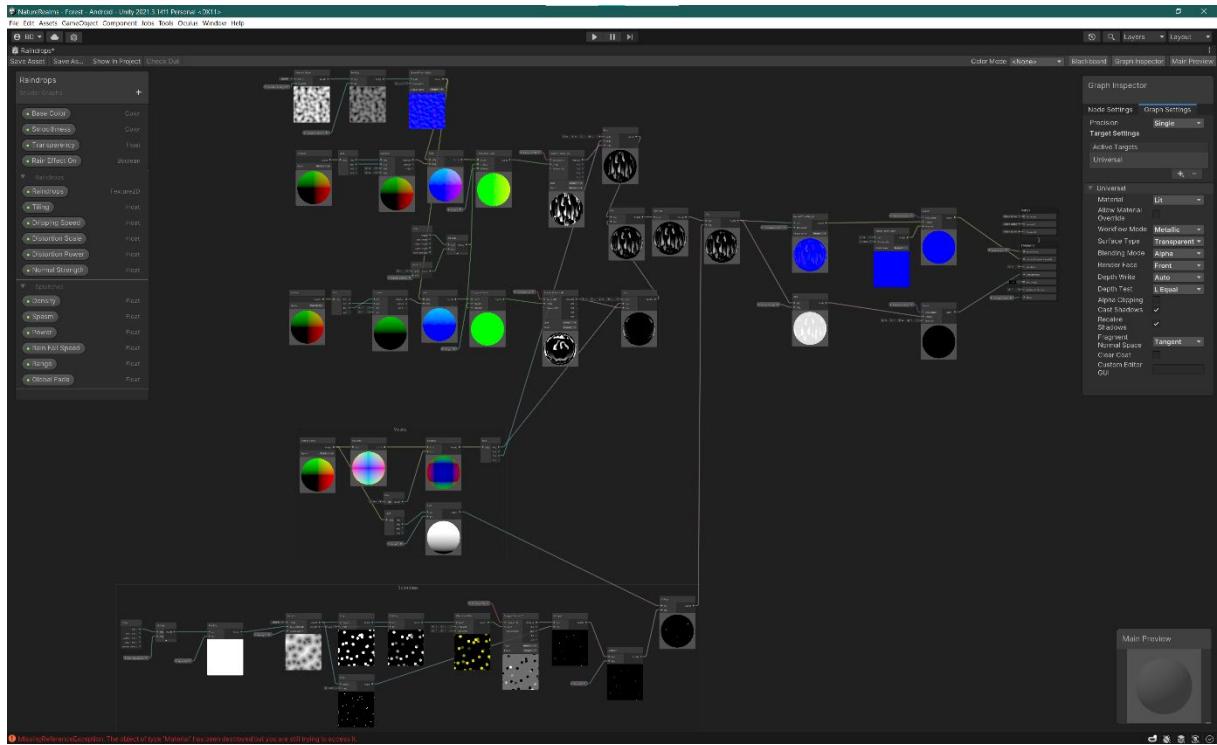
4.3.7.1 Referenceboard in Miro



4.3.7.2 Screenshots



4.3.8 Visual Effect: Umbrella Raindrops

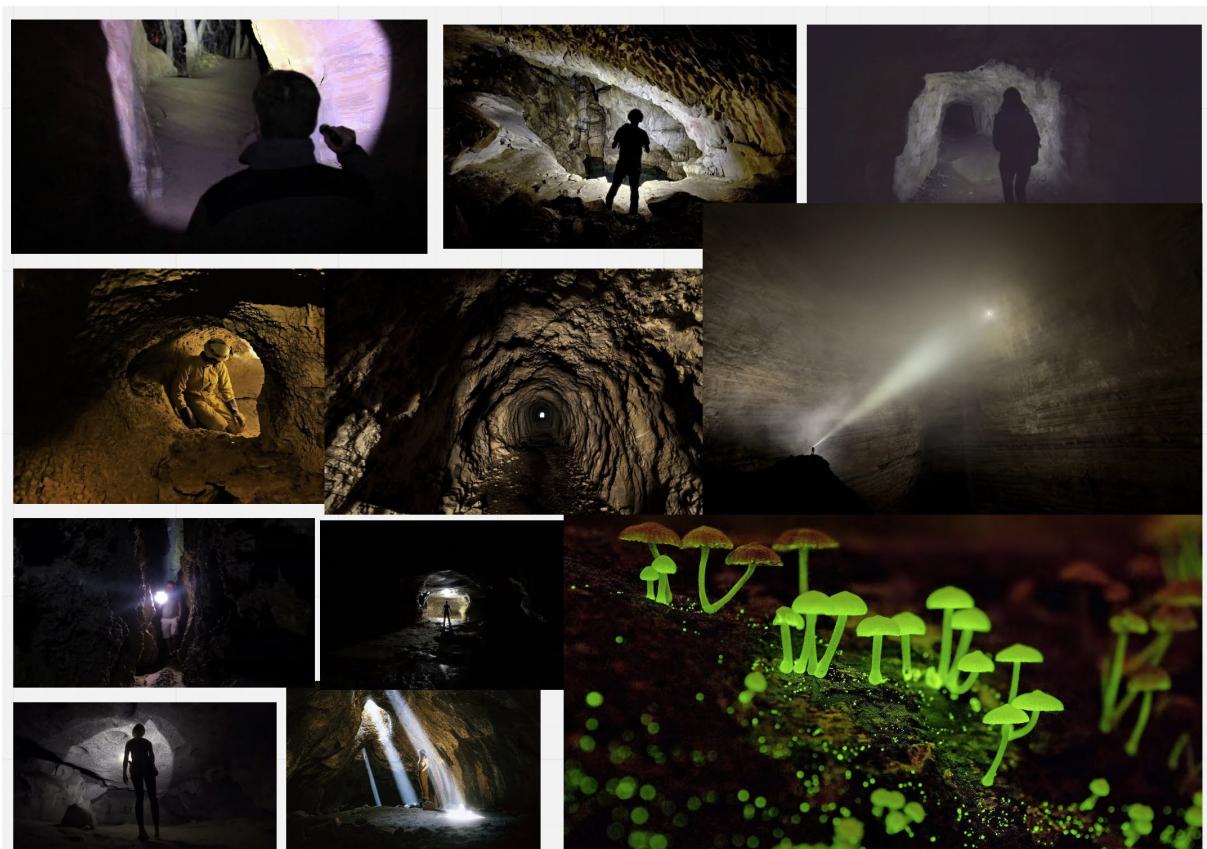


4.4 Scene 3: Cave

The cave consists of two locations. In the first the user experiences a wide open space, but with a very narrow passage next to an abyss. In the following the ceiling gets very low and the user has to crawl through towards the light. Along the way there are luminescent mushrooms to guide the way.

For the walls and structure of the cave a custom 3D model was build. The rest of the assets and materials are from the Quixel Megascans library. The bioluminescence for the mushrooms was easily achieved by slightly editing an existing texture and using it as an emission map.

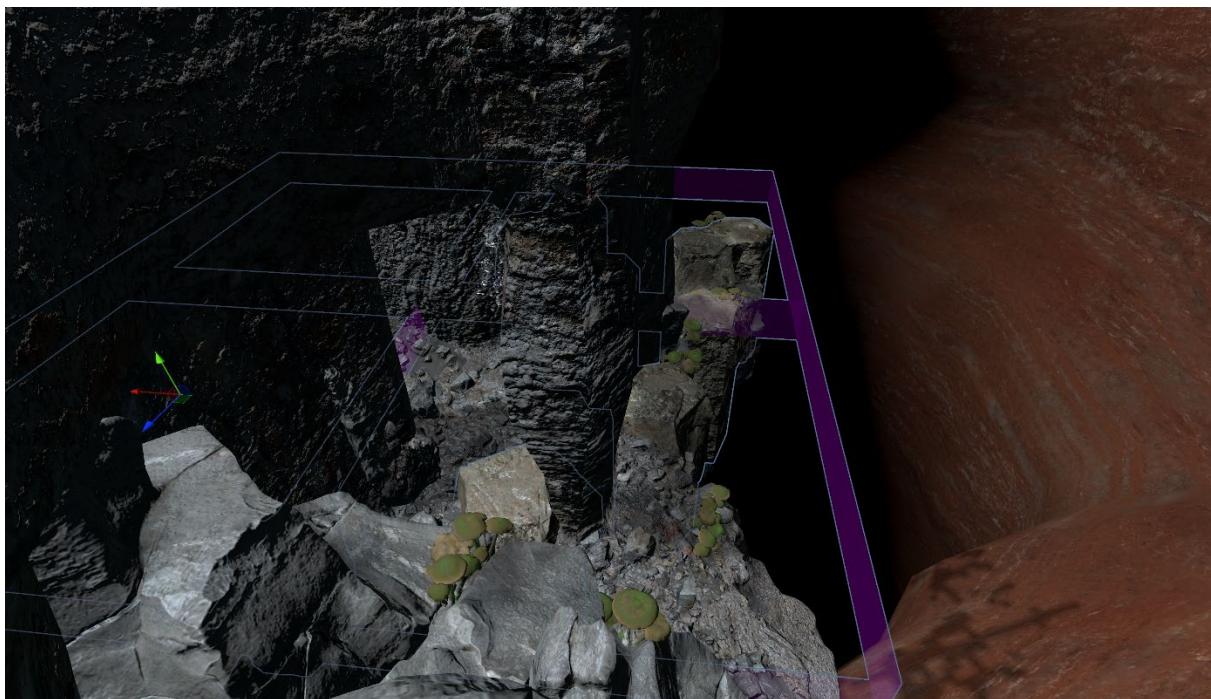
4.4.1 General Moodboard in Miro

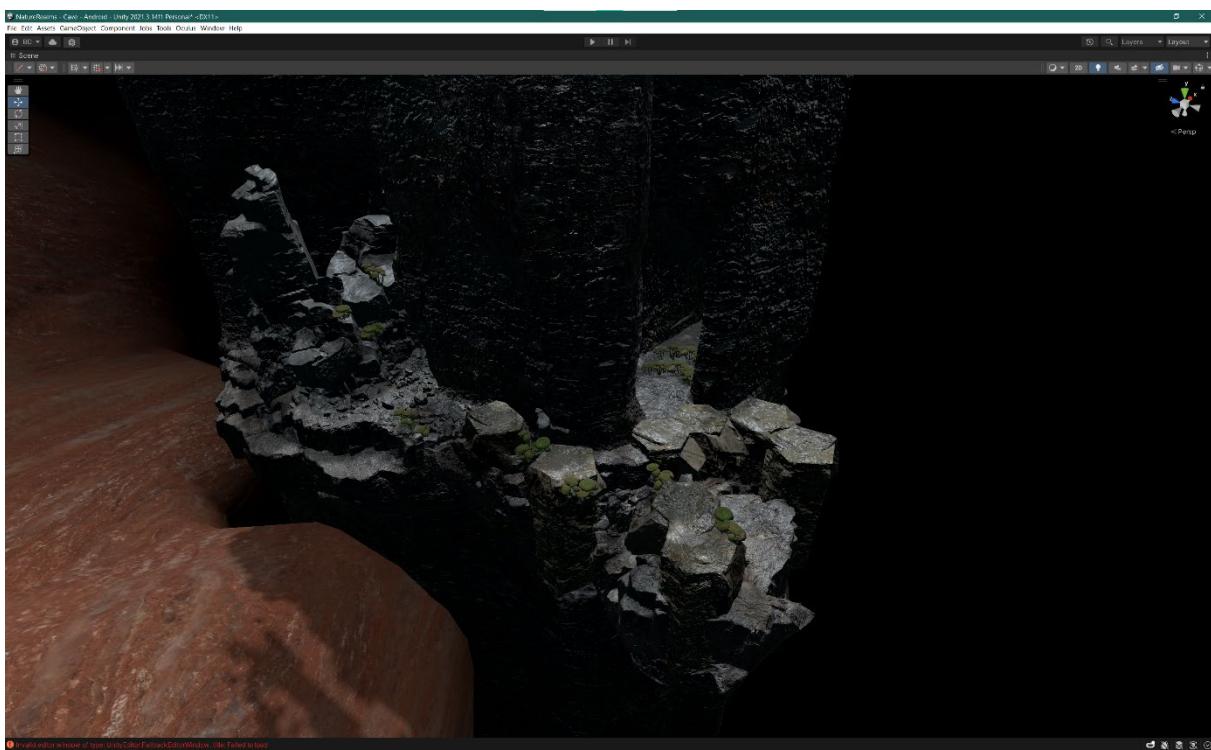


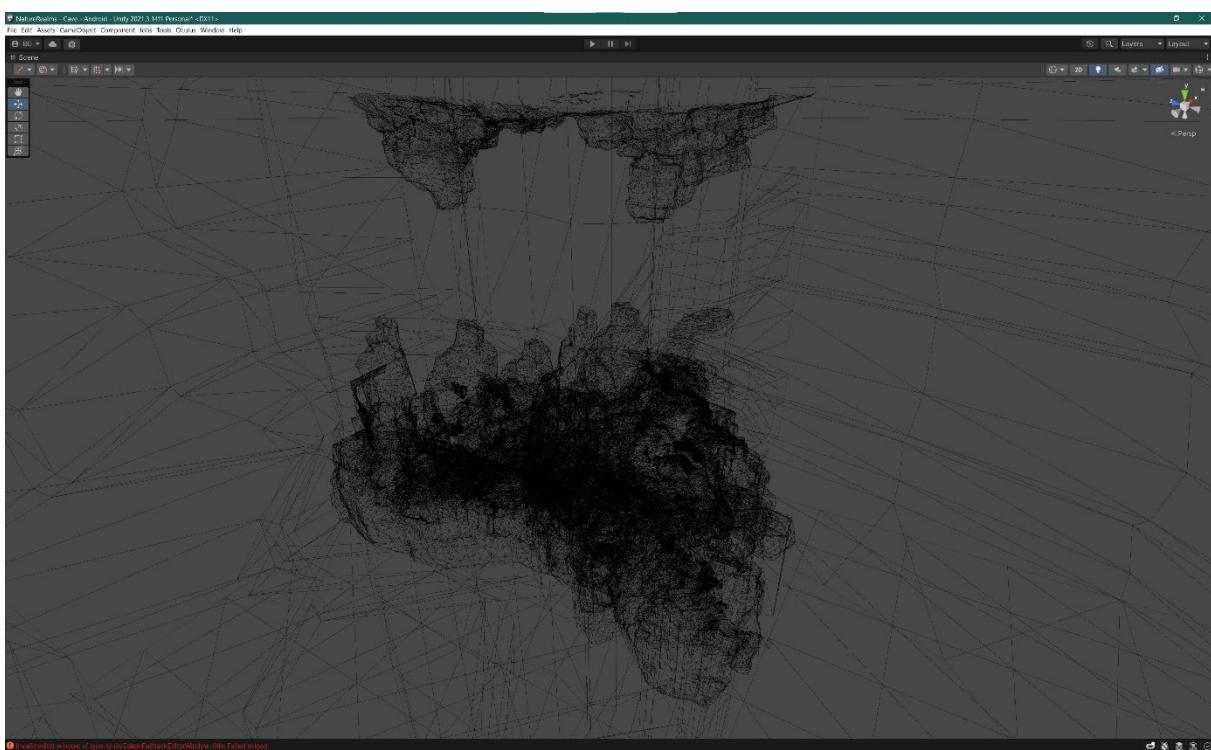
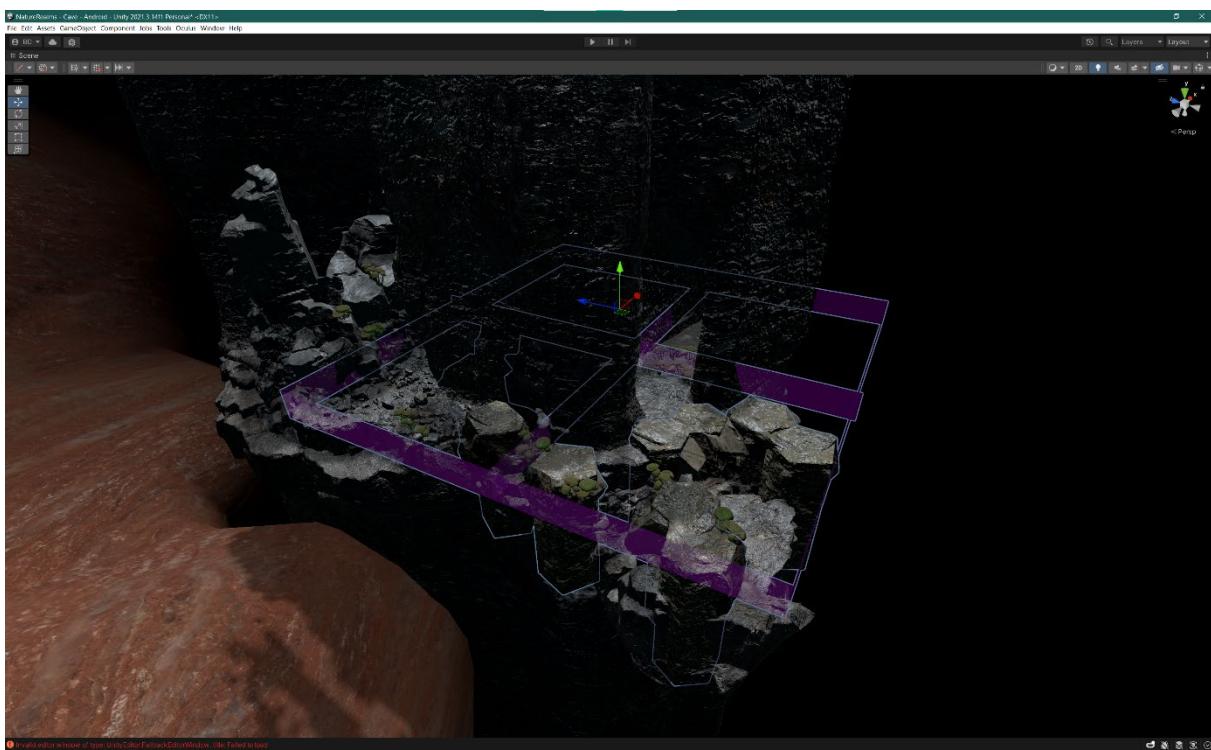
4.4.2 Location 1: The Abyss

The purple lines represent the physical platform size. The screenshots are significantly lit up.

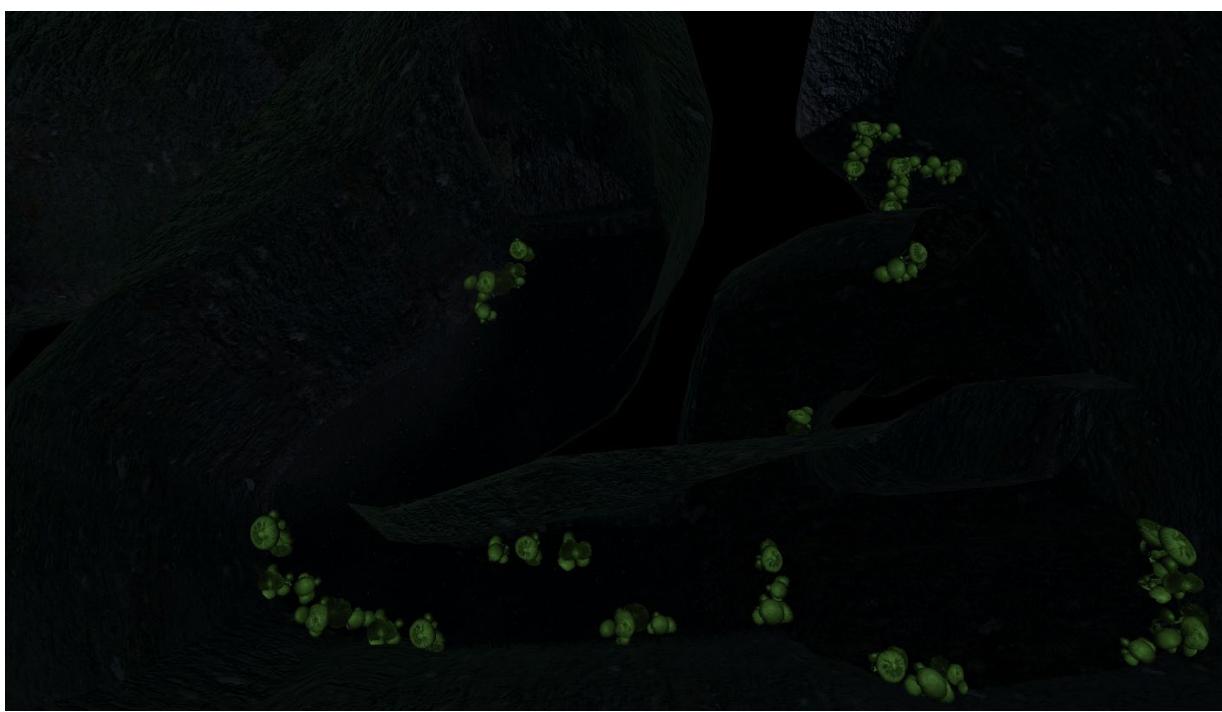
The actual lighting off the scene is the dark screenshot where you can only see the mushrooms in the darkness.

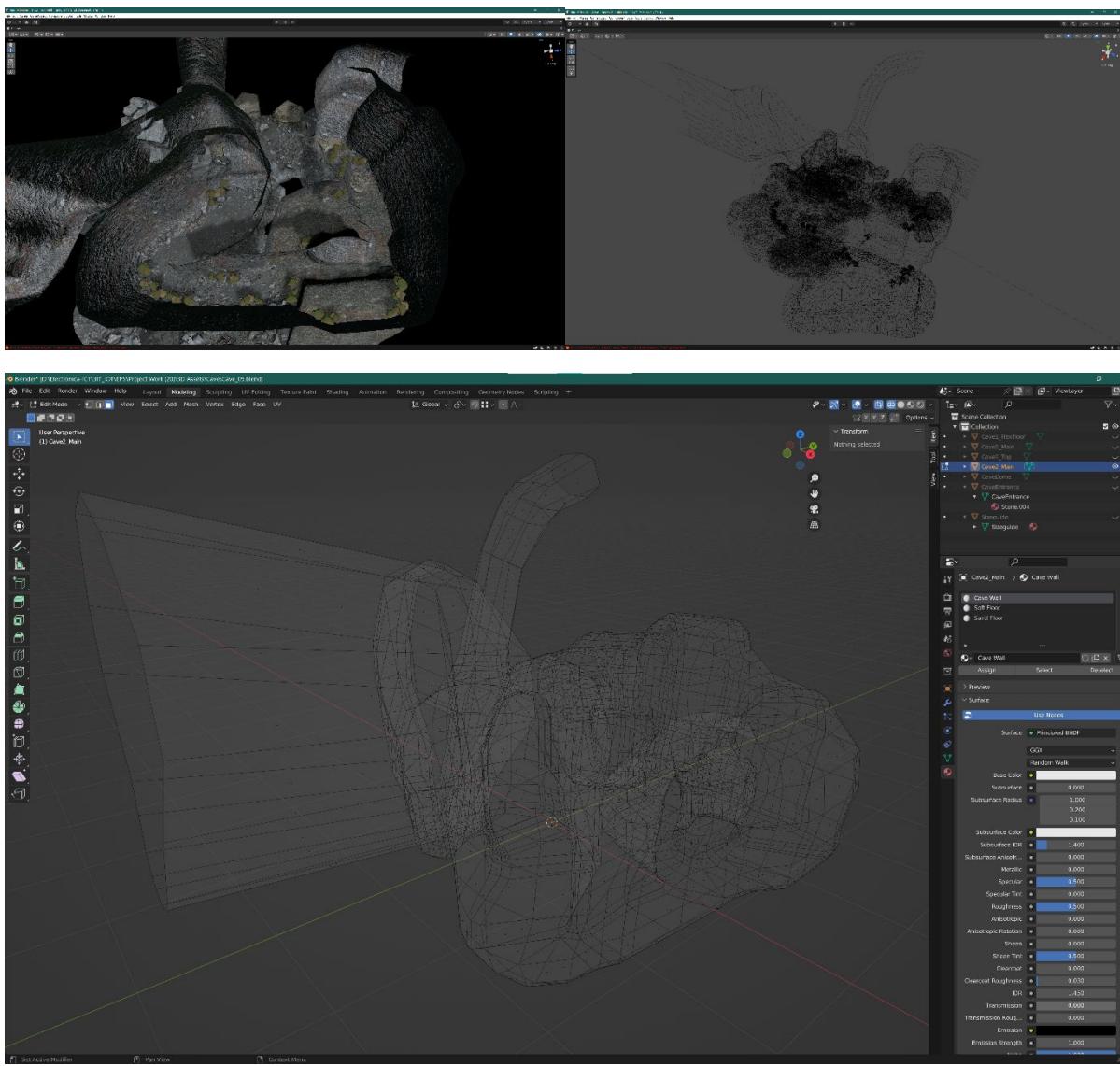




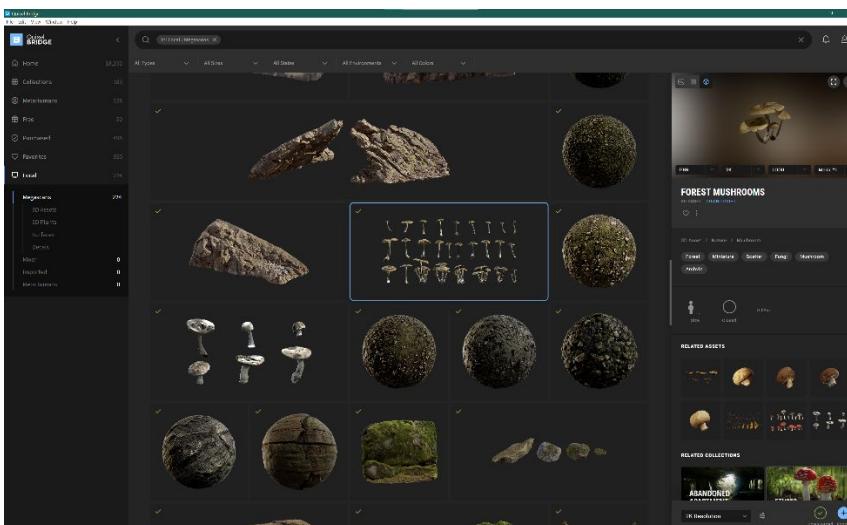
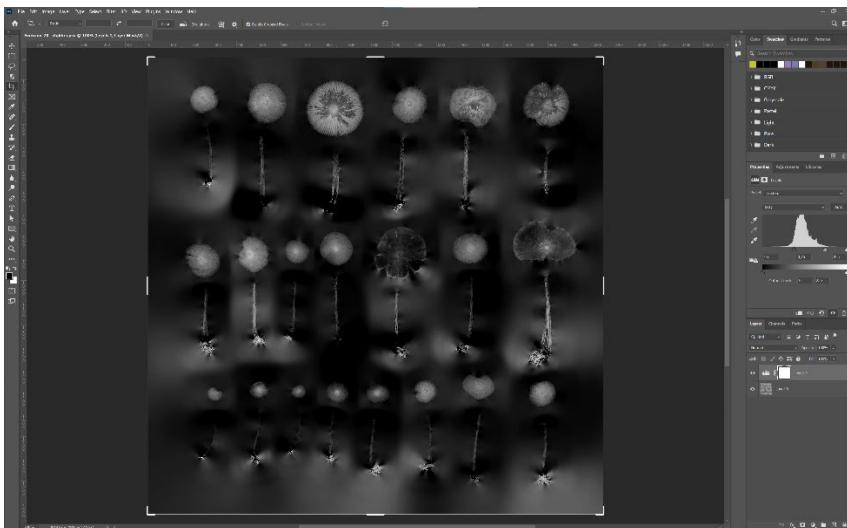
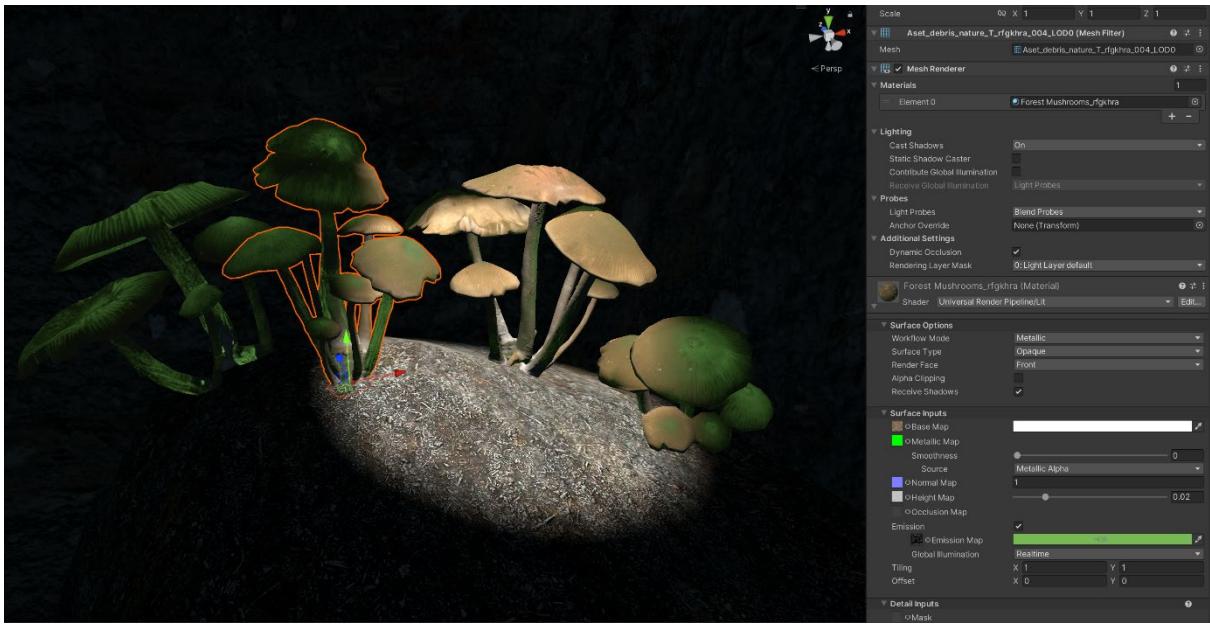


4.4.3 Location 2: The Tunnel





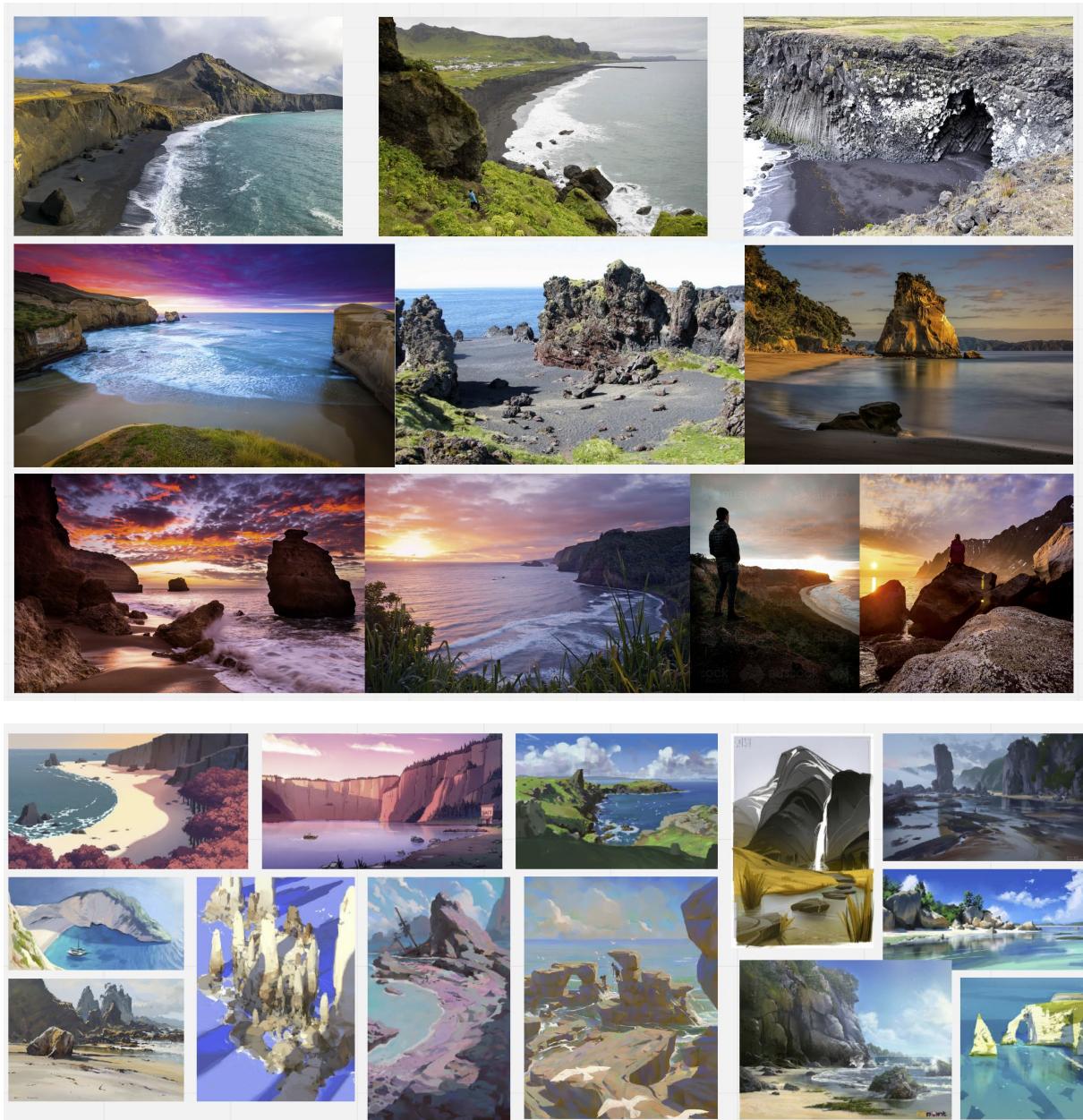
4.4.4 Visual Effect: Bioluminescent Mushrooms



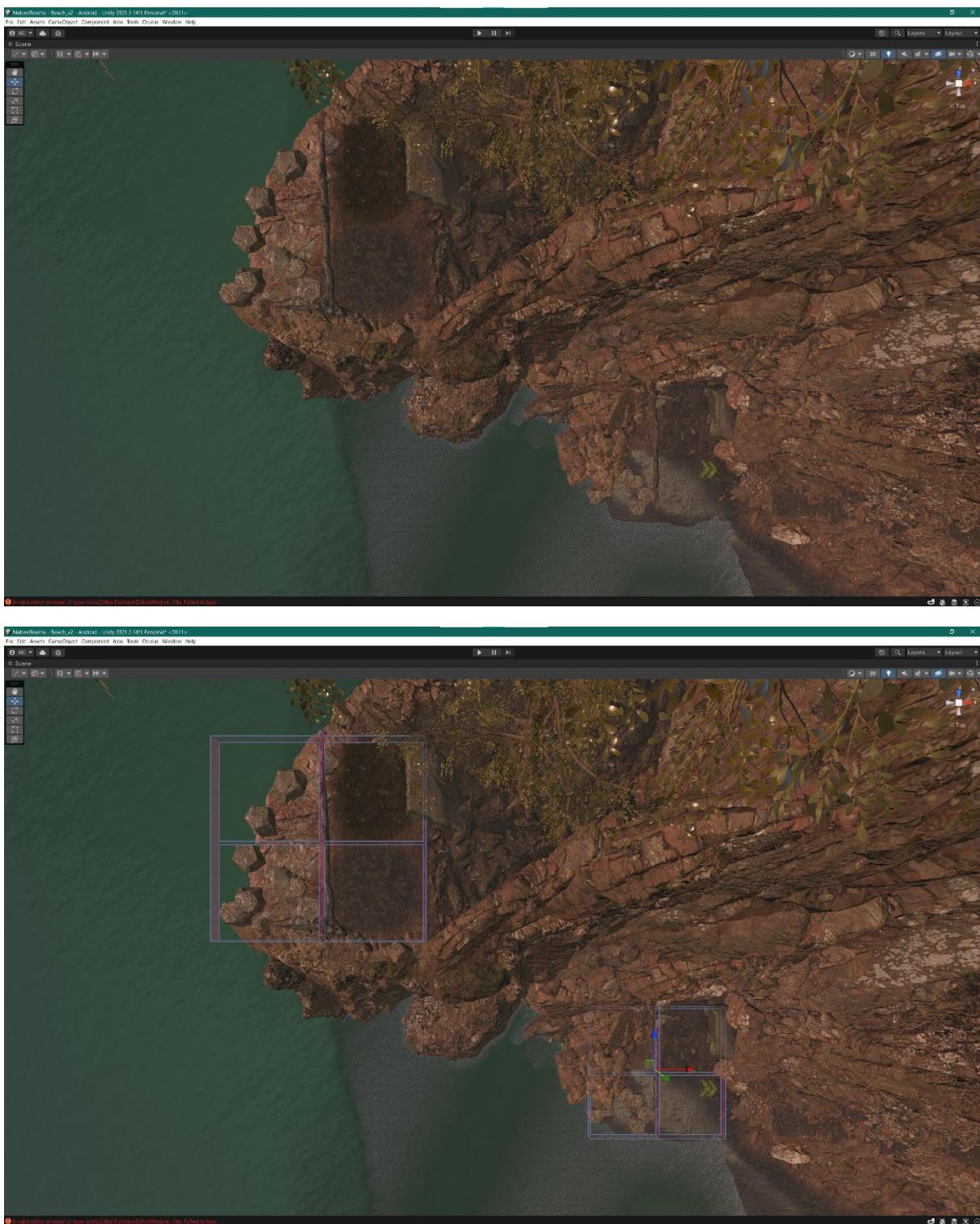
4.5 Scene 4: Beach

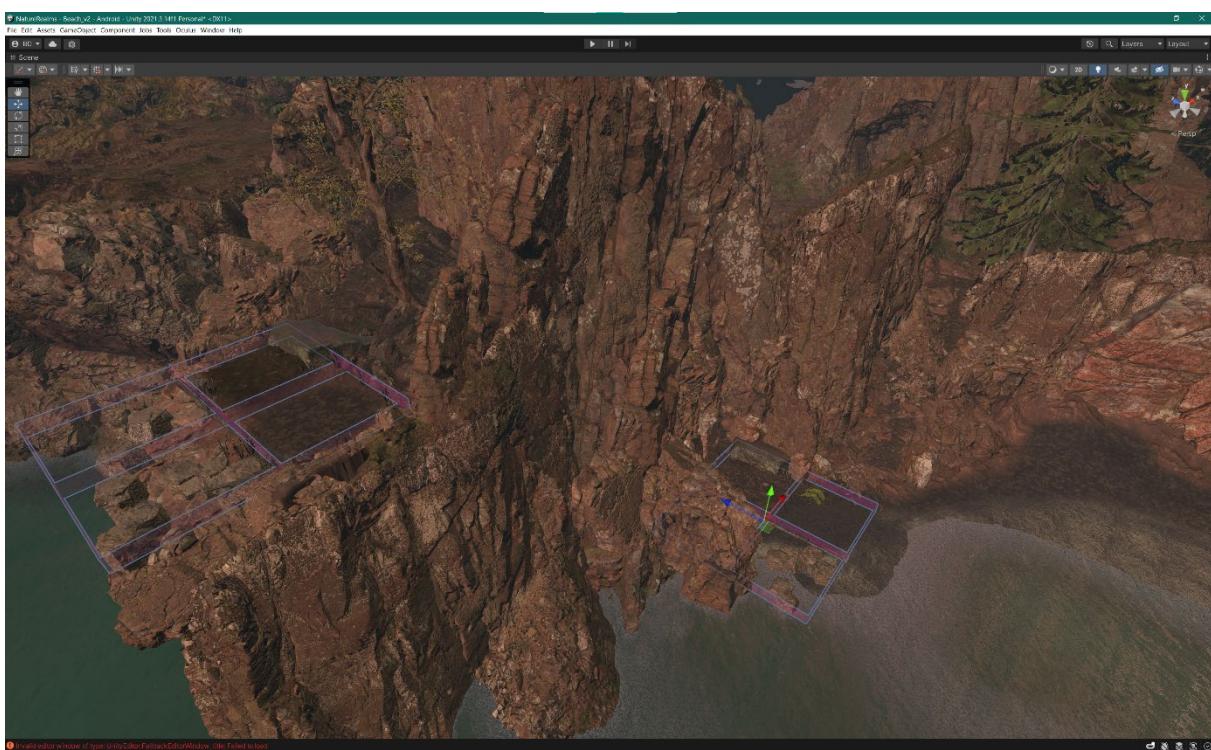
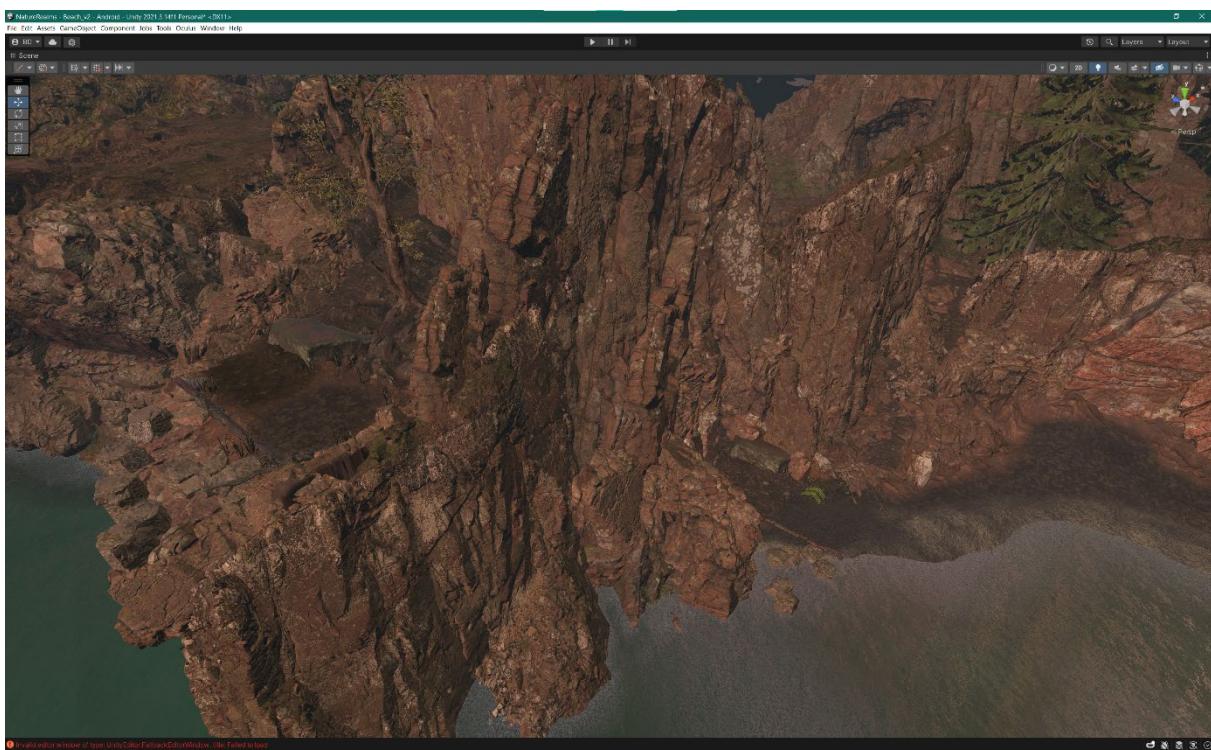
The Beach is meant to be a relief when you come out of the cave. It is a wide open view of a beautiful sunset. You can stay on the beach and enjoy the ocean waves or go to the top of a cliff to have an overview of your surroundings and enjoy the sunrise. Both standing up or sitting down.

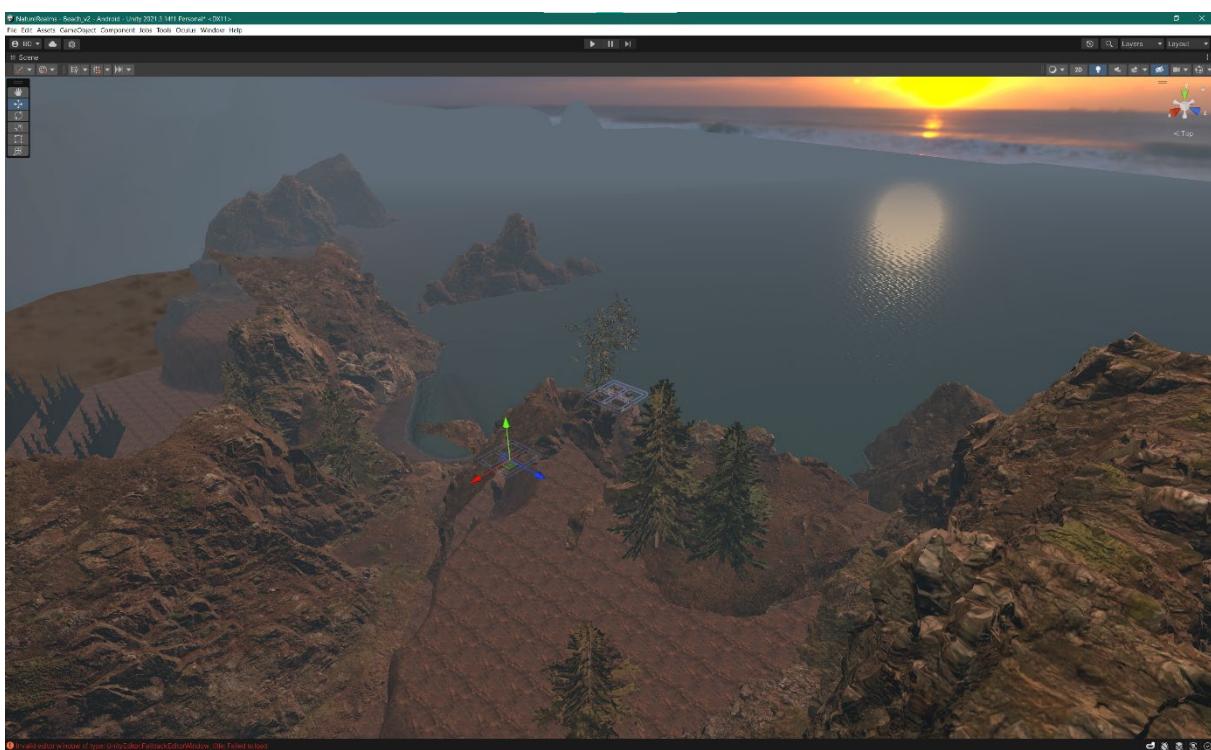
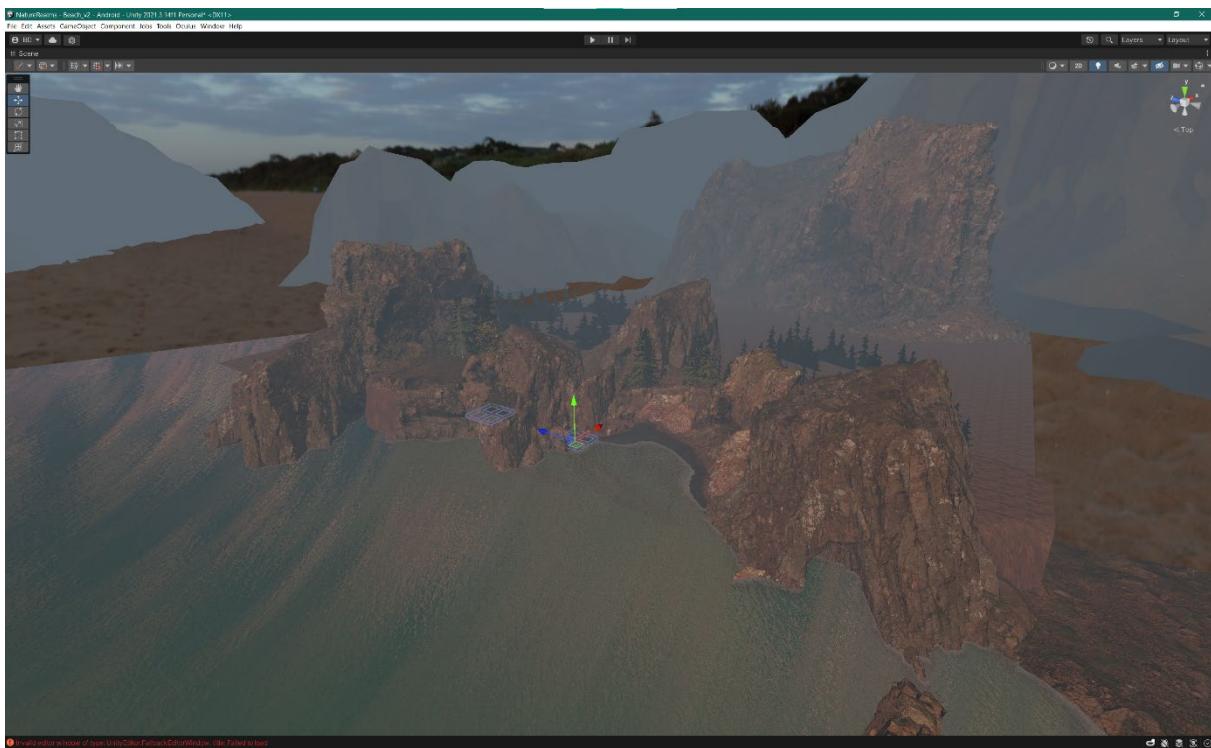
4.5.1 General Moodboard in Miro



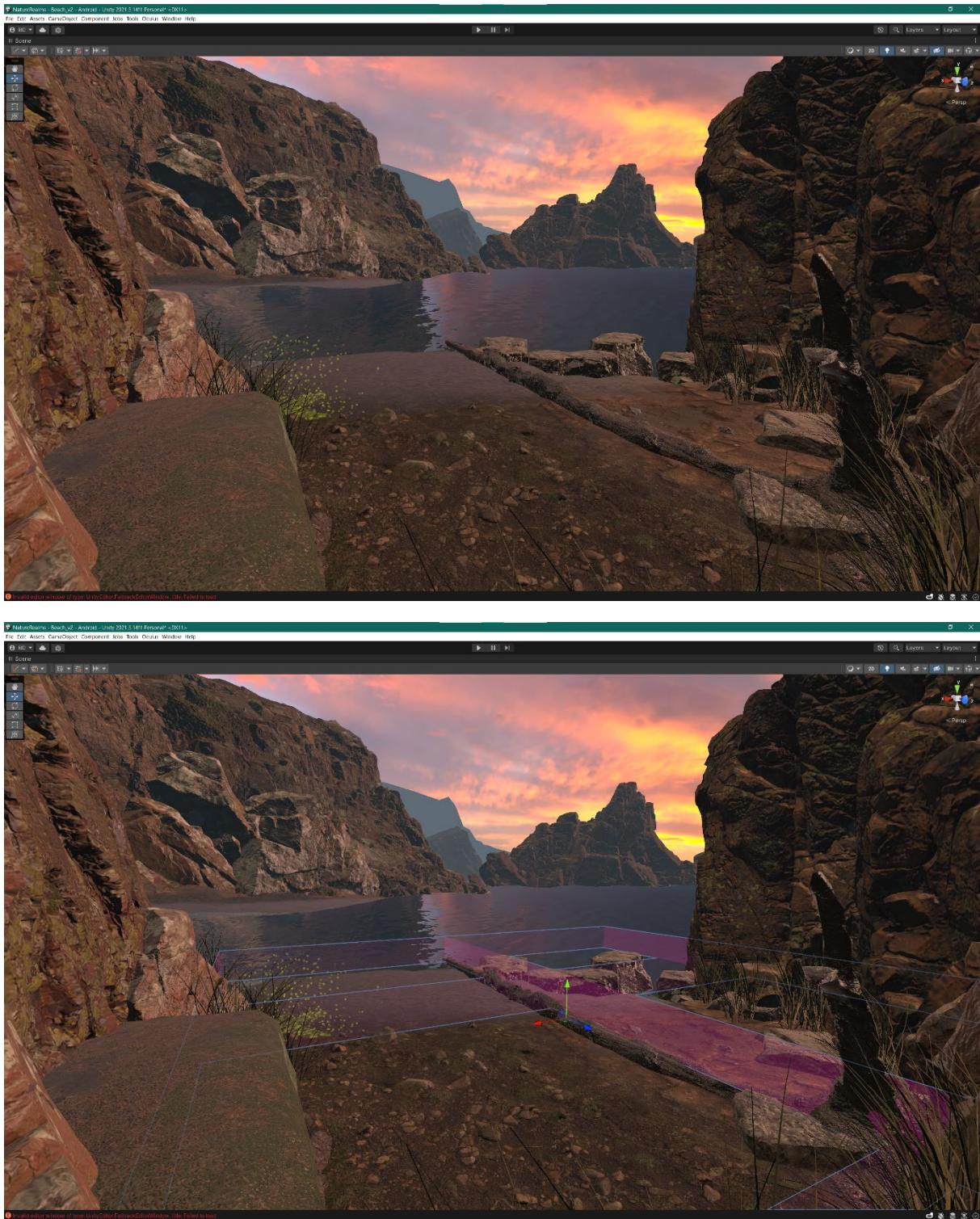
4.5.2 Overview

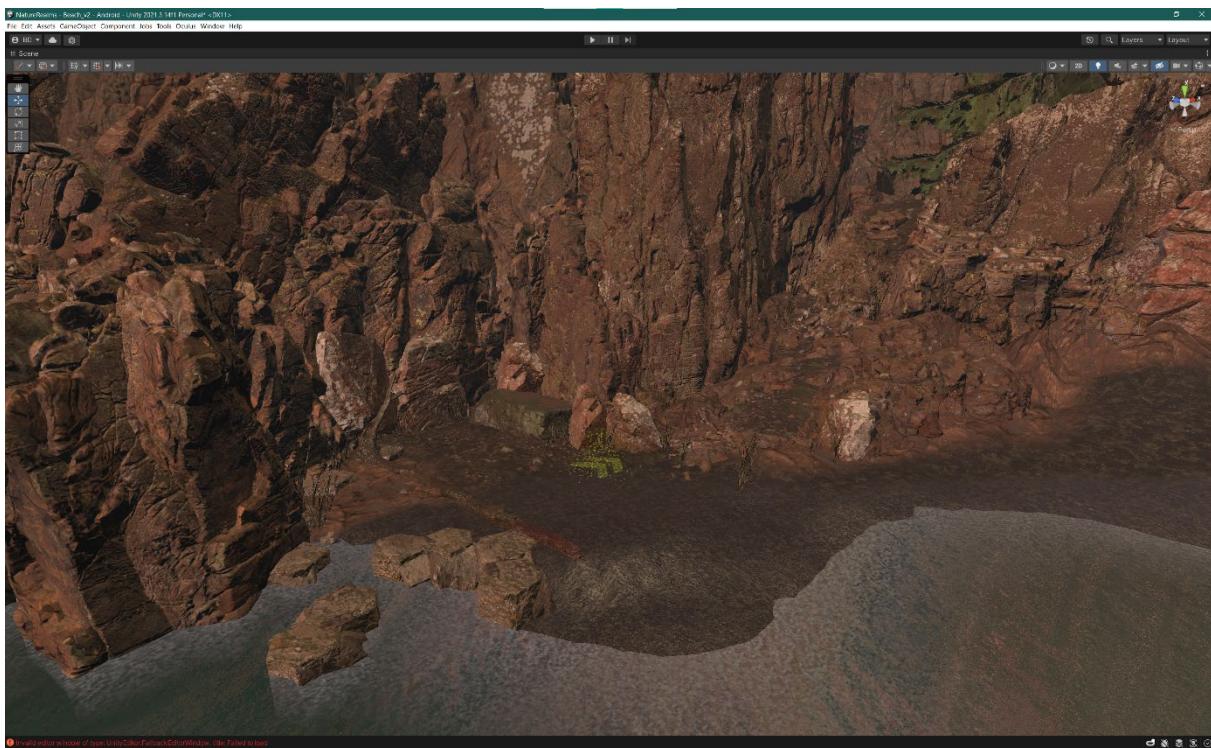






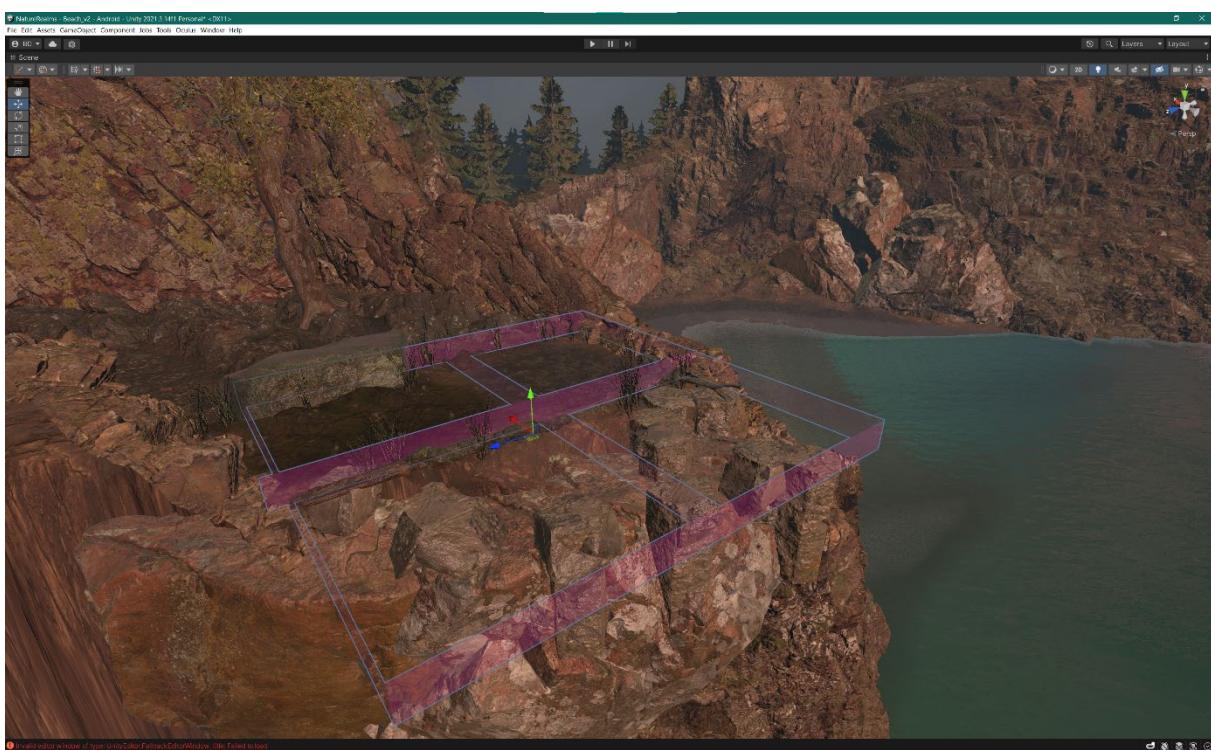
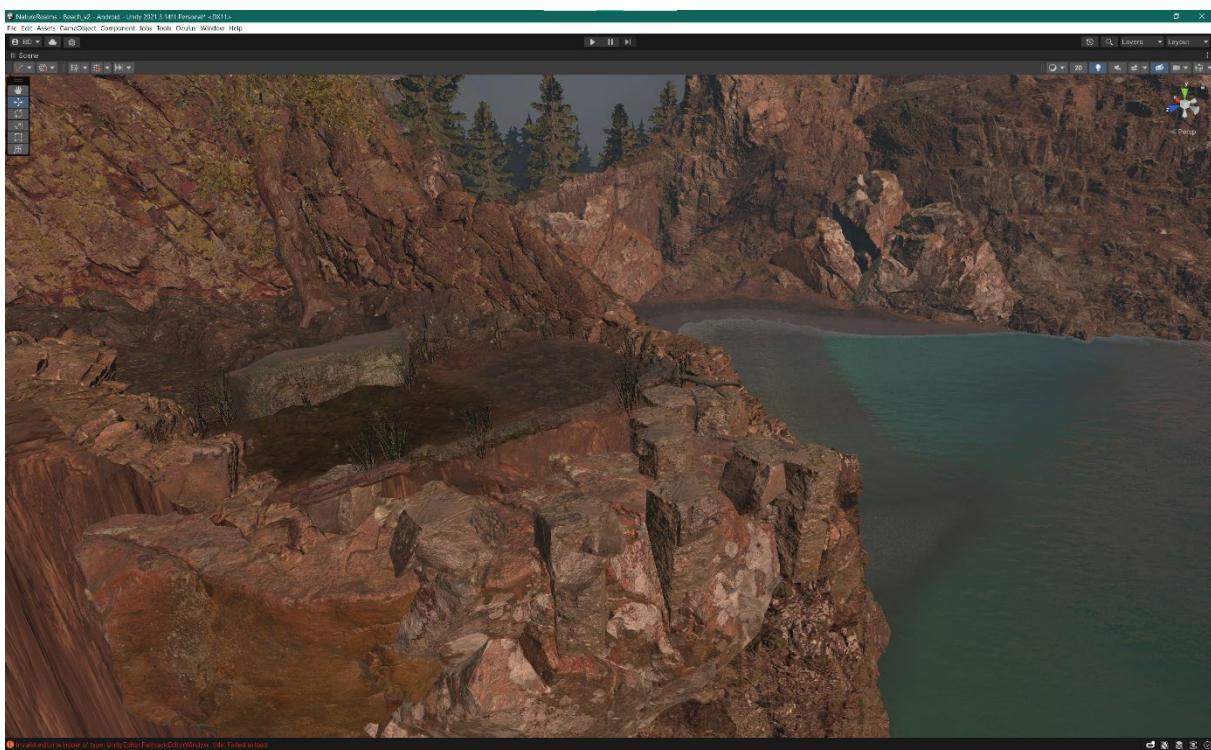
4.5.3 Location 1: The Beach





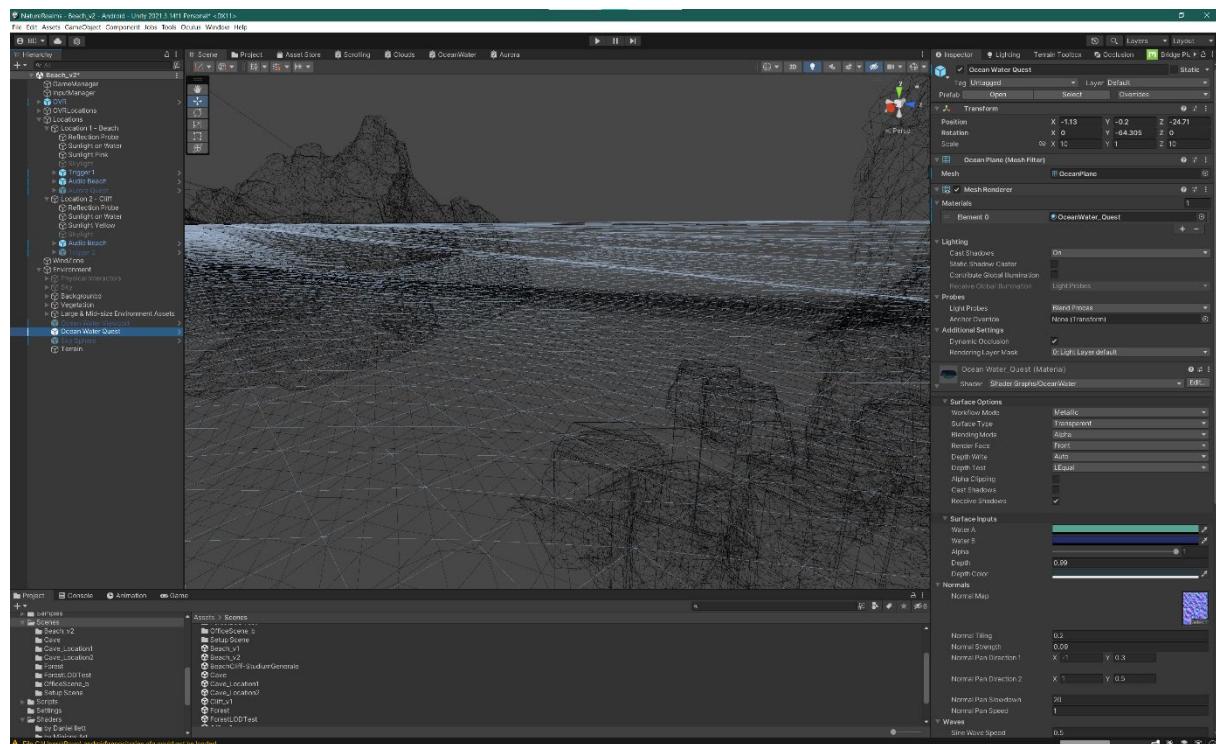
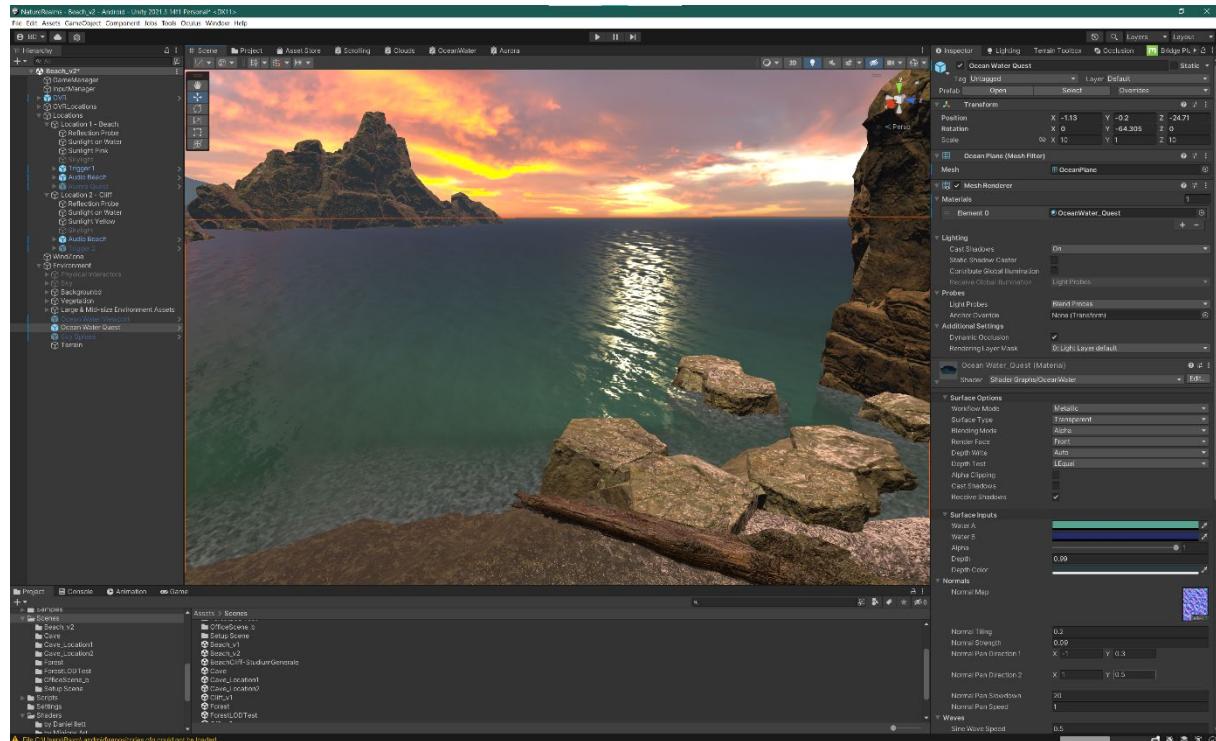
4.5.4 Location 2: The Cliff

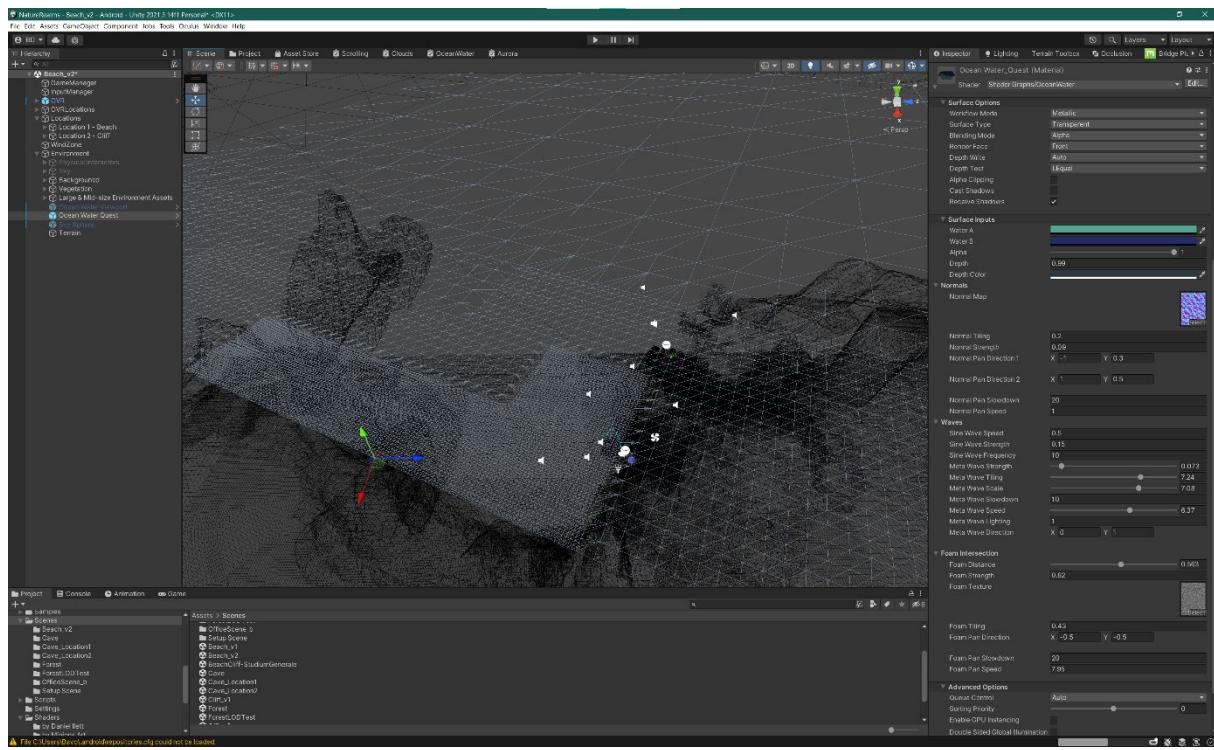
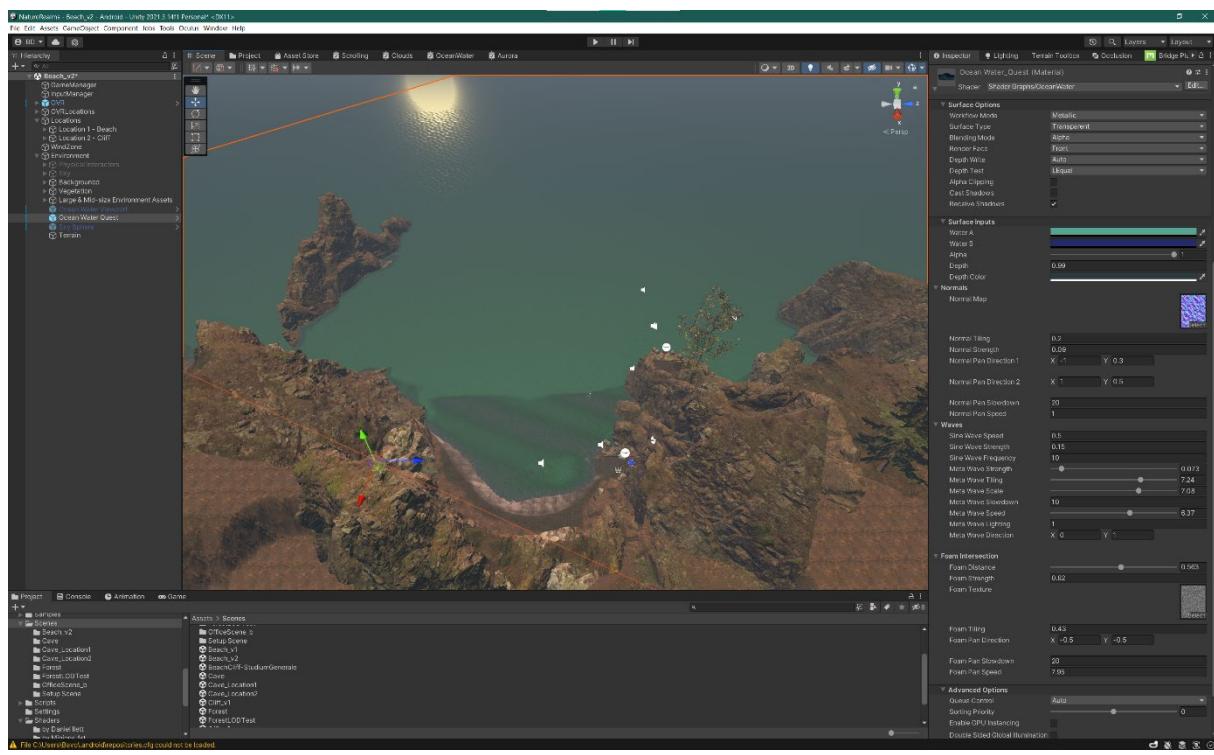


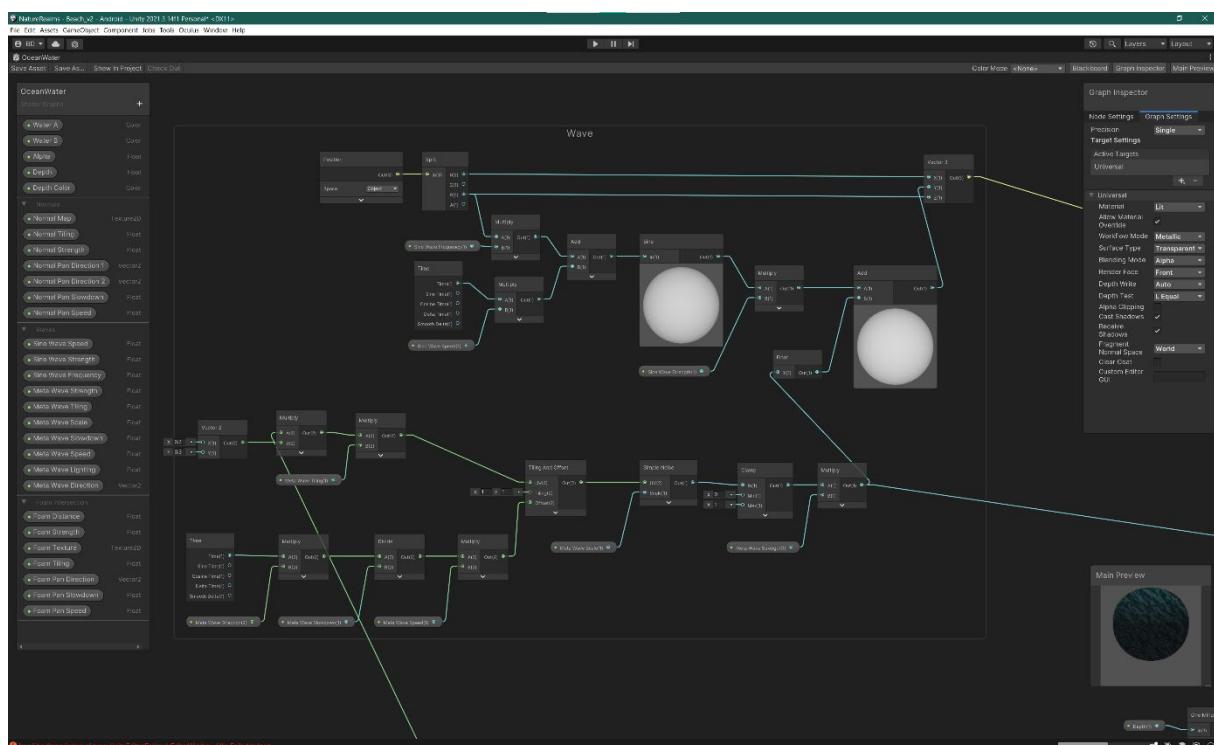
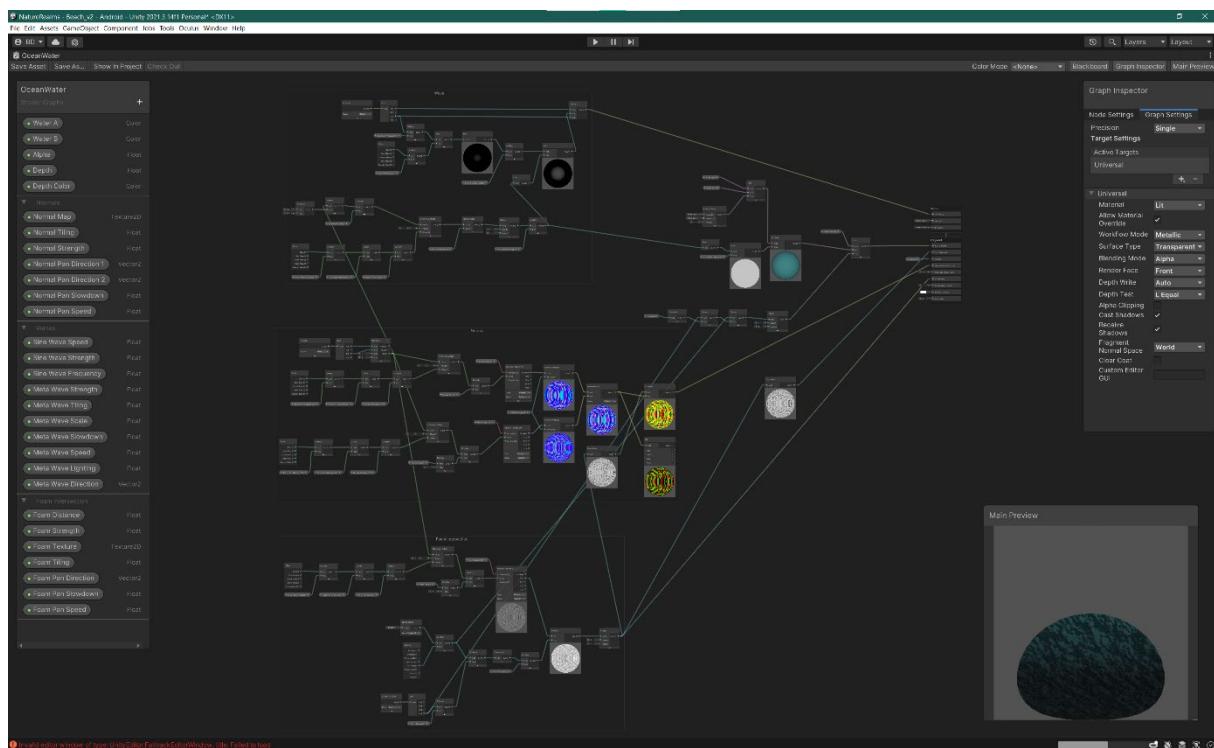


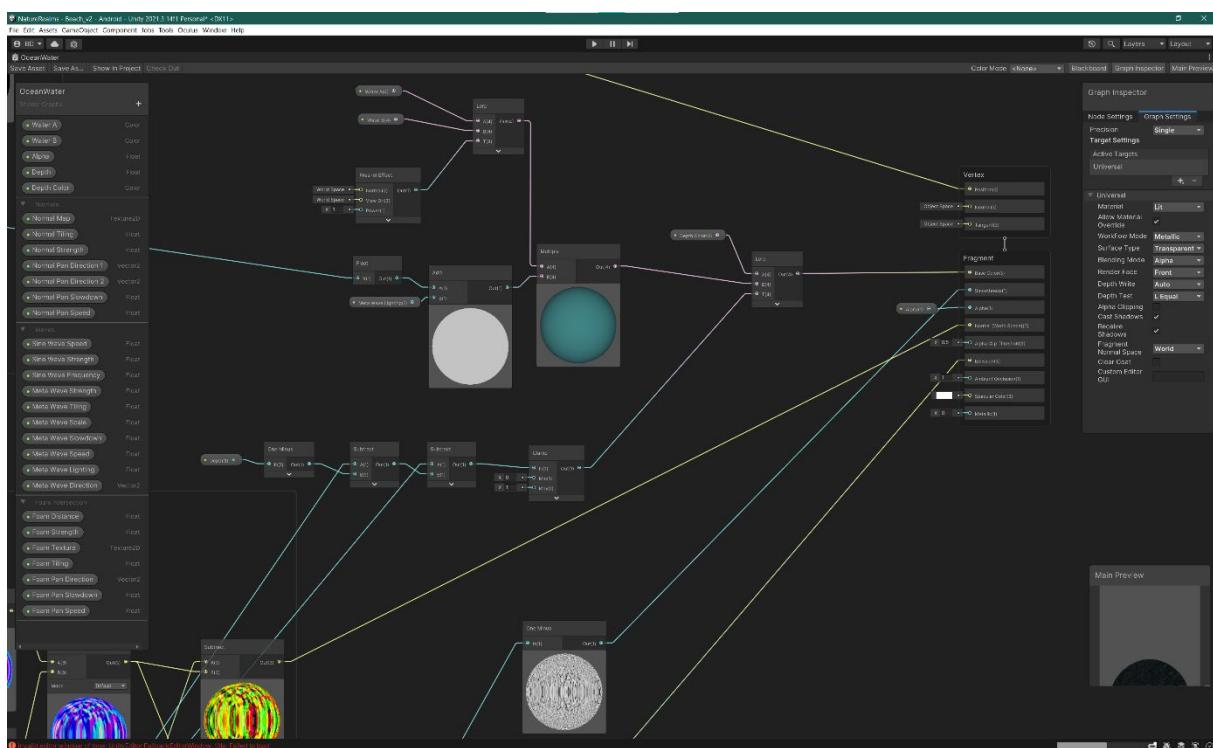
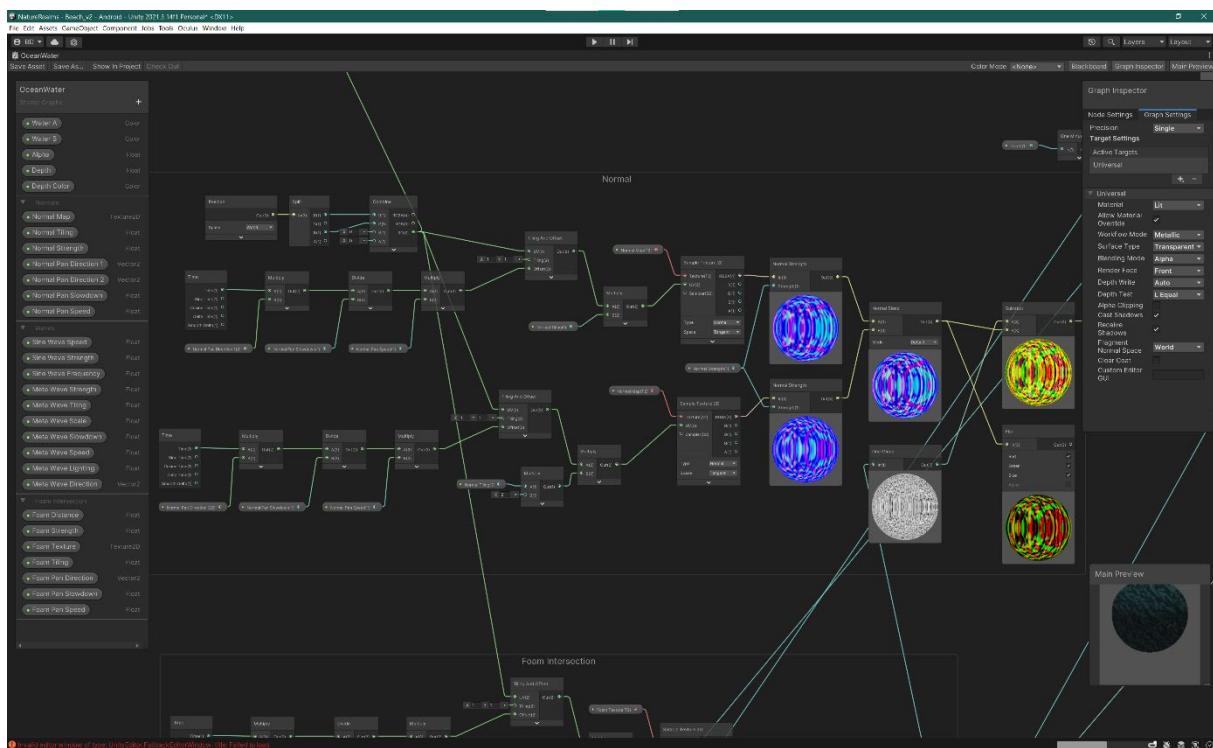
4.5.5 Visual Effect: Ocean Water

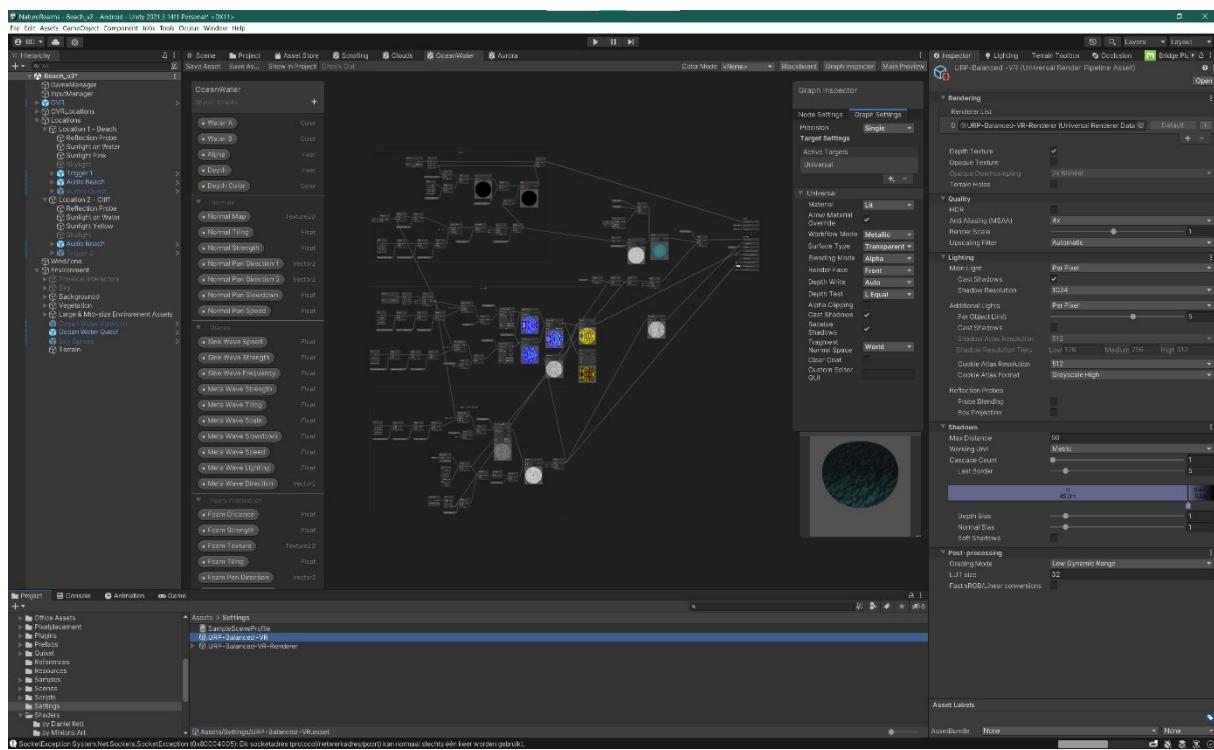
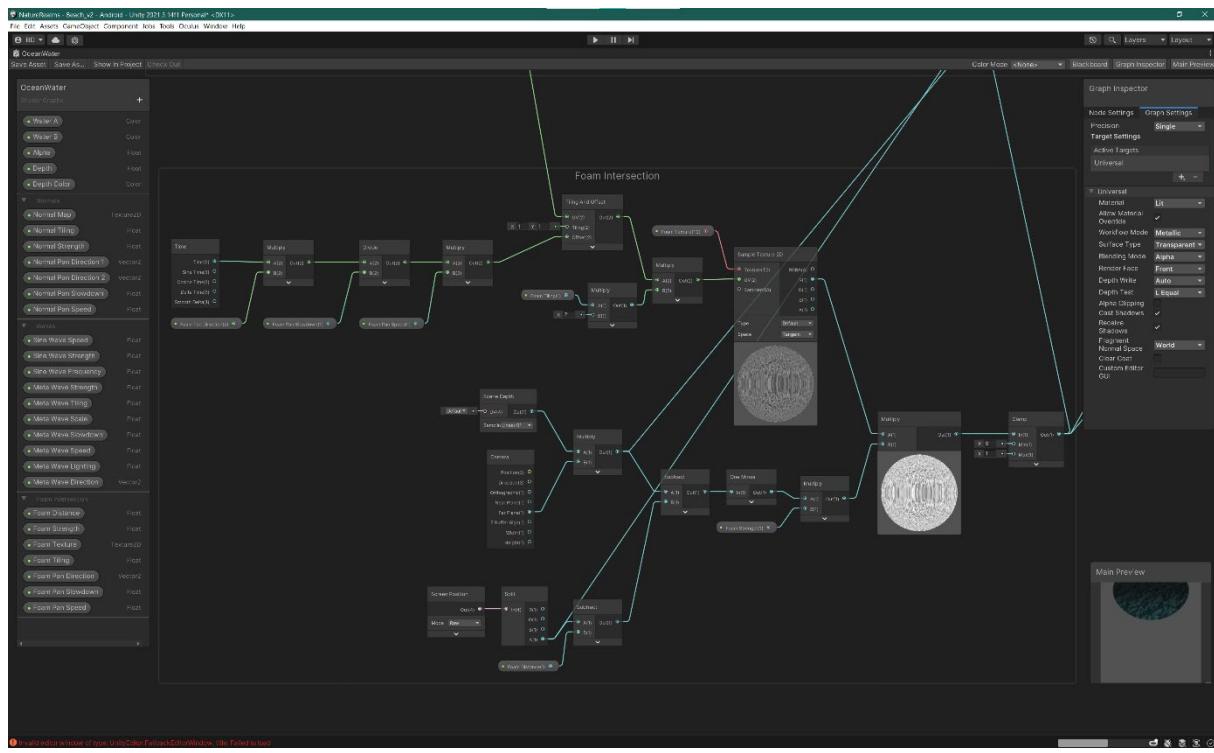
For the Ocean Water a custom 3D model was made to spread to spread the triangle count the where it is needed and a custom shader with ShaderGraph. To be able to use the Depth Texture needed for the shader this feature needs to be turned on in de render settings.







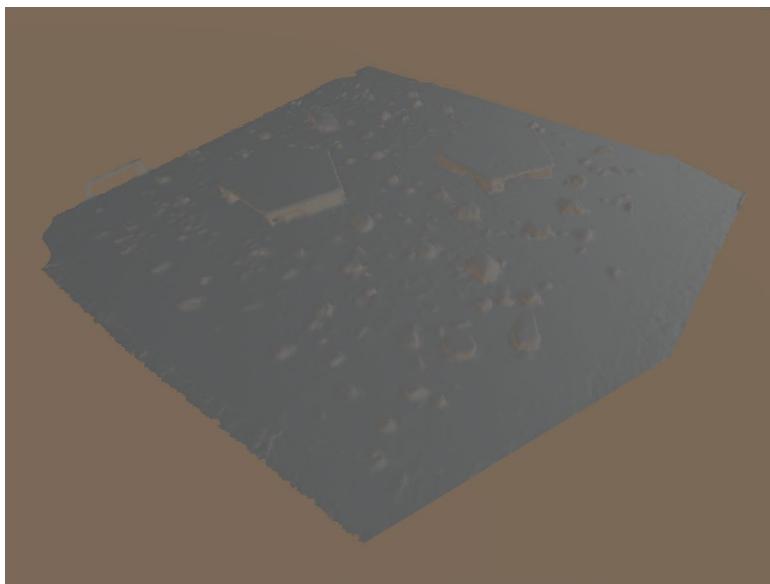




4.6 3D Scans

As explained in the process chapter two 3D scans were made for the project.

4.6.1 Physical Setup



4.6.2 Fellow Student

