

Computer Graphics for Games 2019

Coconut Island – Coral Reef

The beauty and simplicity of watching water ripple around an island is mesmerizing. The goal of this project is to try to capture it as much as possible.



Marcos Pêgo

86472

CGJ

marcosppego@gmail.com

Abstract

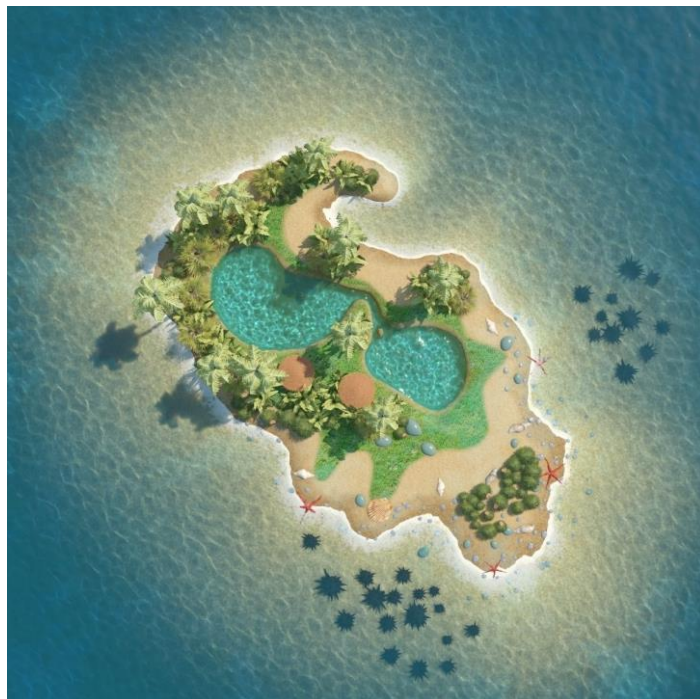
The main concept tackled in this project is the creation of realistic water around an island. Given the challenge we couldn't just use reflections and refractions to create the material as we would need to also create a rippling effect. So I decided the best approach was to follow this [tutorial](#), it explained fairly in depth how to create textures of the current scene to create refraction and reflection. Also, how to create the ripples, how to add on a normal map so we can have highlights and how to polish a bit the water to get the final result. To add the island and corals I used blender, an obj loader and a texture loader to add them into the scene, then using shaders created the effect of light. The use of light helped not only create the realistic effect of the objects but also helped make the water seem more real when looked at. In the end, I also decided to implement a simple underwater mechanic using shaders and caustic textures.

1. Concept

I first got the idea whilst playing *Bad North: Jotunn Edition*. The vibe the game gave of playing in a calm island and the simple controls gave for an immersive gameplay which made me feel like I was a commander trying to defend my homeplace.



I suggested the idea of making a real island to my, at the time, partner and she agreed. We decided a tropical island would look nice and even though the style wasn't exactly like the islands mention in *Bad North*, it would look good in a final project as a demonstration of our own game engine.



With this said, we decided to split task somewhat like this: I made the water and the light, she made the stuff that would be added to the island such as trees and coconuts and would make a mouse picking so we could pick up objects on the scene, and we would both help create the island mesh a guarantee all of the scene looked nice and cohesive.

So, I set my goal into making a realistic looking water like the water portrayed above and this was my initial result:



Given that eventually we had to split up the project, I now had nothing good to put up in the island itself. So, after dwelling on the subject for a few days, I decided to change the concept and make a coral reef on the shore of the island to give a more focus to the water rather than the island itself.

After a while, every time I added a new coral and for any reason looked below water it always seemed of, so I decided to implement an underwater system and for that I used [this](#) as a reference.

2. Technical Challenges

Since this part was all done by me there is no need to specify who did what.

- 2.1. Creating Refraction and Reflection**
- 2.2. Applying the Fresnel formula**
- 2.3. Applying Du/Dv maps to create ripples**
- 2.4. Applying a normal map to create Highlights**
- 2.5. Polish**
- 2.6. Creating a Bling-Phong lighting**
- 2.7. Underwater Caustics and blue like filter**

3. Proposed Solutions (max 5 pages / student)

Since I found the water tutorial early on the development. Most of the challenges were straight to the point. The most hardships I had were based on implementing everything on c++, since the tutorial was in java and it wasn't always obvious the conversion between the 2, and bugs and roadblocks that appeared more often than not.

Creating Refraction and Reflection

The water refraction and reflection were made by creating a brand-new texture using fbos and loading them on to a flat plane. With more detail, to create the refraction texture we render the current scene (without the plane in it) from the camera's perspective and store it in an fbo, and to create the reflection we use the same method but use a different camera inverted on the y axis.

There is no point in showing the refraction (because it will look invisible) but here is what the reflection looks like:



Applying the Fresnel formula

Simple Fresnel

For the first iteration I decided to use a simple version of Fresnel which just made the dot product between a vector $(0,1,0)$ and the camera vector, this gave a factor that could be used to simulate the Fresnel effect by mixing the refraction and the reflection based on the angle between the normal of the plane. However, this formula is not the most accurate at describing the real Fresnel effect of water.

Fresnel Formula with the Schlick simplification

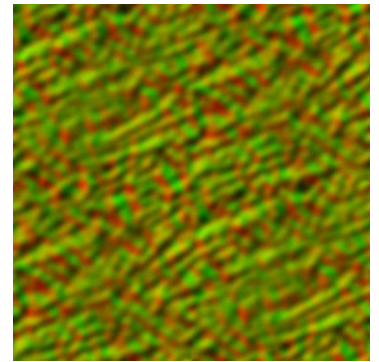
By using the Schlick simplification: $F = f_{\lambda} + (1 - f_{\lambda})(1 - \mathbf{v} \cdot \mathbf{N})^5$ and using the real f_{λ} for the water we ensure the most realistic Fresnel effect possible.



Applying Du/Dv maps to create ripples

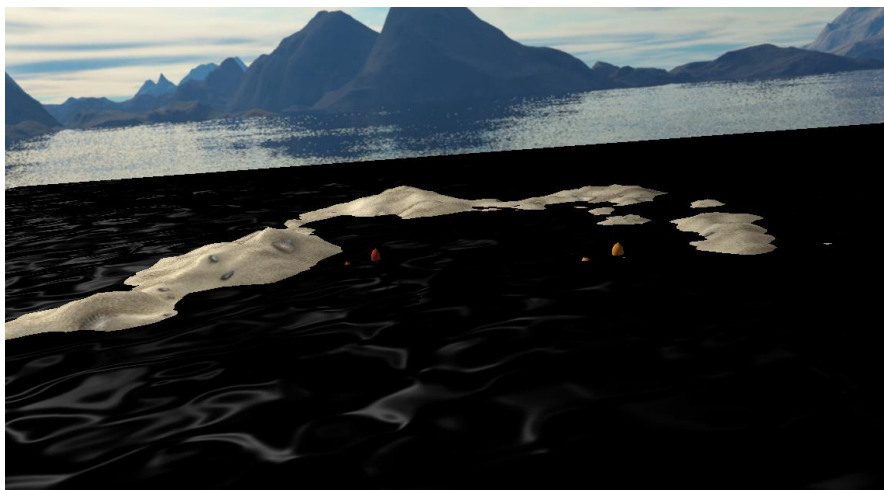
This was done by loading a texture like this:

Where the red pixels meant a low point in the wave and the green pixels meant a high point in the wave. This way we can simulate what the texture of the mix between refraction and reflection would look like if a wave was passing through it. This is nothing more than distorting the picture with the right values to seem like the texture is moving.



Applying a normal map to create Highlights

Much like the technique used before we load a texture which helps us predict what would be the normals where the water would be high or low. This way we can simulate normals whilst still allowing us to use only a flat plane. We use this so we can calculate what and where the Highlights would be.



Polish

When I imported the island, I could notice that around the edges between the island and the water was kind of blurry between colours. This ruined the experience as it just liked that the water was slicing up the island.

The work around was to implement a depth system which allowed me to see how close to the shore the water was. With this information I was able to tweak the amount of distortion near the shoreline. Making it calmer when near and slowly going back to the normal distortion, I also change the transparency of the water near shore so it's easier to see the floor under the water.

This simple tweak allowed for not only fixing the weird edges problem but also gave the overall look of the scene a more realistic feel as the water was calmer near the island.

The water depth (were black is near the water level and white is far):



Creating a Bling-Phong lighting

The Bling-Phong was a simple implementation as I already had done the ambient light and diffuse light calculations in the previous project. All I had to add was the specular lighting but since it's fairly similar to the highlights of the water there wasn't anything new to it.

Underwater Caustics and blue like filter

I put myself the extra challenge of creating an underwater feel for the scene. For this I searched the web on how this could be made. The first thing I noticed were the water caustics.

The water Caustics where simple, all I had to do was much like the highlights of the water, load a texture and add it like if it were a specular light.



Afterwards I tried to tackle the harder problem: Creating that blue like feel around the scene. There wasn't much information available that I could find, and my first guess was that I could disable the red component using the glColorMask, which just lead to a strange look. It could be considered good enough, but I didn't like it.



After a while I found out that it could be possible if I simulated fog but gave it a blue colour, so once again I found a tutorial on how to make [fog](#). I tweak it to my likings and get a much nicer a realistic looking underwater.



4. Post-Mortem (max 2 pages)

4.1. What went well?

The fact that the tutorial I found was so straight forward helped me stay on the right path. Even with many roadblocks in the implementation department at least I could be sure that it was the right way to go and somewhere along the line I had made a mistake.

Often, it was just a small bug keeping it from running smoothly which allowed for fast passed coding.

Most times there wasn't even any problem and the development was going as smooth as it could be, with me even sometimes doing things ahead of time before even watching the tutorial or searching it online.

The caustics on the water are a good example of that, I just looked at the video and decided I wanted to do that. Without even searching a tutorial I implemented it easily, by using knowledge I acquired during the development of this project. Even if it wasn't the best solution possible out there, it gave me a sense of accomplishment and satisfaction to have done something by myself and succeeding.

4.2. What did not go so well?

The obvious part would be my partner not coming through with her part. Me being busy on my own part and other projects from other courses kind of made me focus on my side and only realised too late that hers was far worse than I expected.

At one week from the delivery date she finally told me she wouldn't be able to make her part. I think it was for the best to split the project because it allowed me to perfect the water and even do a new challenge that I wasn't expecting.

The other part was just the number of bugs and roadblocks I found. By my fault, sometimes I was running on a tight schedule, which meant a lot of

nights with little sleep and, because of that, a lot of unnecessary mistakes or just a tired mind that couldn't fix the problem no matter how many times I tried.

This led to situations like the reflect camera, where I spent 8 hours trying to fix it to no avail and ending up going sleepless to the discussion that day with what I considered an unfinished project. Only to, the next week, fix it within 15 mins.

Situations like this could've been avoided had I prepared better for the semester and managed better my time.

4.3. Lessons learned

I think the biggest lesson I learned was to work on my time managing skills. Had I properly managed my time I wouldn't have wasted so much time on fixing things I couldn't fix because of sheer tiredness, and maybe, been more aware of the situation of my partner and tried to help her more than I did.

If I were to start again or at least give some kind of advice, would be to be consistent on your work, not only on this course but the others as well. One good week won't make up for a bad one, try as best to be consistent through the semester and don't tire yourself out needlessly because of it.

References

Other references were already mentioned and hyperlinked to the relevant spots.

Skybox:

<https://learnopengl.com/Advanced-OpenGL/Cubemaps>

Blender landscapes:

<https://www.youtube.com/watch?v=yrMee2gcS20&t=1325s>

Coral Assets used:

<https://free3d.com/3d-models/coral>