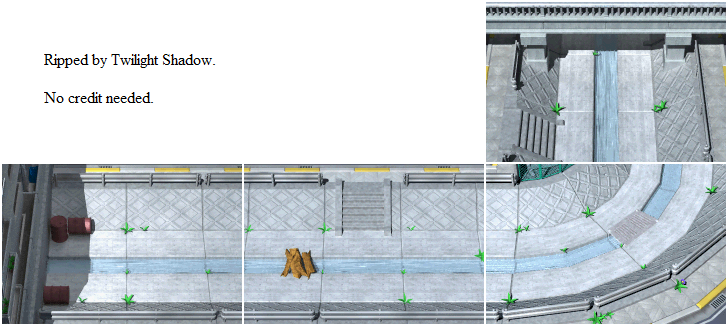
Water Shader en Normal map Shader

Table of Contents:

1. Concept and idea
2. Creating the environment
3. First version of water and normal map shaders
4. First conclusion
5. Second version of water and normal map shaders
6. Definitive conclusion

## Concept and idea



In the game Yugioh: the sacred cards you control a sprite that can move along these pre rendered images. There are no animations, there’s no scrolling and with that knowledge wouldn’t it be great if you could see this area in 3D with normal maps to represent the walls and a water shader to make the water and therefore the rest of the area feel alive.

<https://youtu.be/V_RvfoaWBhs?t=3099> here’s a timestamped video where you can see the area that I’ll be trying to recreate called “Bridge”

The goals for this assignment are:  
- The water must look like it could be from this game that means it should be:  
semi-transparent,   
have a water flow,  
have water refractions,  
- The Normal map must look it’s 3D like how the walls look in the image with the cross pattern.

## Creating the Environment

So, the way that I’ve tried to 100% recreate this area was by using 2 plugins which are Realtime CSG and probuilder.   
Realtime CSG is based on brush-based mapping which is something that started with quake mapping back in the day and the simplest way of describing a brush is that it’s a simple mesh that has properties for each vertex, edge and surface to give you a lot of freedom in how you want to shape your maps by being able to modify all of it.   
Example of a modern brush-based mapping tool is Trenchbroom and with it you can make maps for games like quake.   
I use CSG to create the straight parts and give ourselves the option to later make small changes if needed to the surfaces and edges.   
But one thing that CSG can’t do is place brushes in very specific places like the angle that I want for the 90 degrees turn, and that’s where probuilder comes in.  
Probuilder allowed me to make a torus and arch at a specific angle with the amount of turns that I wanted so that it would look exactly like the image.  
So by combining these 2 tools I was able to recreate the essential area where the shaders are going to be.

## First version of water and normal map shaders

The first version of the water shader was me Trying out shaders after 1 year and it was slow progress from trying out how to make waves and getting that done but then getting stuck on how to give them direction. To solve this, I started over and went into it with the intention of implementing direction as well into the waves. I thought at the time that having that and a normal map would create a flowing effect, it did but the result wasn’t like the concept so there had to be some changes.  
I started working on the normal map shader by first figuring out how normal maps worked and when I got that down I had to figure out how to mix it with a texture because before this every version was just the normal map. Second version used world reflections for the normal map which ended up working but now every normal mapped surface looked like a mirror because it was reflecting the skybox without any objects.

## First Conclusion

The shaders that I made worked but they weren’t up to the standards set by me.  
The world reflections on the normal map should be lighting instead and the water shader needs more transparency and lighter colors used to look more like the water from the image.

## Second version of water and normal map shaders

For the second version of the normal map shader I followed a tutorial by freya holmer that goes through making a normal map shader by using lighting instead of world reflections.  
And for the water shader we used a catlikecoding tutorial on flowmaps but because it’s for surface shaders we had to refactor it so it would work with our code. The end result is

Sources:

<https://docs.unity3d.com/Packages/com.unity.shadergraph@6.9/manual/Fresnel-Effect-Node.html>

<https://www.youtube.com/watch?v=6_-NNKc4lrk>

<https://www.youtube.com/watch?v=E4PHFnvMzFc>

<https://www.youtube.com/watch?v=lWCPFwxZpVg>

<https://catlikecoding.com/unity/tutorials/flow/texture-distortion/>

<https://answers.unity.com/questions/1494854/mathfrepeat-in-shaderlab.html>

<https://unitywatershader.wordpress.com/2018/05/14/sky-reflections-and-fresnel/>

<https://unitywatershader.wordpress.com/>

<https://docs.unity3d.com/2019.1/Documentation/Manual/HOWTO-Water.html>

<https://catlikecoding.com/unity/tutorials/flow/waves/>

<https://docs.unity3d.com/Manual/SL-ShaderPrograms.html>

<https://docs.unity3d.com/Manual/SL-VertexFragmentShaderExamples.html>

<https://gist.github.com/TarasOsiris/e0e6e6c3b8fdb0d8074b>

<https://www.youtube.com/watch?v=UfX9dzhBhg0>

<https://catlikecoding.com/unity/tutorials/flow/>

<https://marketplace.visualstudio.com/items?itemName=ShaderlabVS2019.ShaderlabVS&ssr=false#overview>

<https://summit2021a.game-lab.nl/>