Ocean Toolkit

version 1.12

The Ocean Toolkit is a code package for the Unity3D game engine that brings realistic infinite ocean rendering to your project. The water surface is offset according to a custom wave function which can be used to control large scale waves. Small scale waves are realized using scrolling normal maps. The toolkit is easy to use and does not require any coding.

Contents

1 What's new

v1.12

Changes:

- Updated to Unity 2019.4 LTS

v1.11

Bug fixes:

- Fixed a bug where the scene file could contain the screen space ocean mesh resulting in a large file size
- Removed a runtime heap allocation

v1.1

New:

- Optimized projection math in vertex shader
- Increased maximum wave length to 80
- The water volume can now be clipped so that it can be used as a lake
- It is now possible to have multiple water game objects in a scene as long as they have different materials
- SSR parameters are now exposed in the Ocean material

Changes:

- The default render queue for the Ocean shader was changed to Transparent-100 in order for the ocean to better mix with particles
- The default render queue for the Ocean Mask shader was changed to Geometry+400

Bug fixes:

- Fixed a bug where the ocean would get clipped before it should if the camera was lower than the highest wave
- There is no longer a half-texel offset problem on D3D9 when Anti-Aliasing is enabled (Internal Unity fix)

v1.0

- Initial release

2 Features

- Realistic infinite ocean rendering
- Customizable wave function that offsets the water surface
- Small scale detail realized using scrolling normal maps
- Refraction and underwater light absorption to accurately convey depth
- Reflections using Screen Space Raytracing or conventional reflection probes
- Shoreline foam
- Optional caustics for the seafloor
- Clean look that fits most projects
- 4 example scenes showing different presets and use cases

3 Set up guide

1. Import the Ocean Toolkit package into your project
2. Open up the scene you are working on
3. Drag the Ocean prefab (located in Core/) into your scene
4. Drag the Ocean Camera prefab (also located in Core/) into your scene
5. You should now see an ocean stretching out to the horizon
6. Change the surface position by moving the game object until it matches your scene

4 Workflow

Typically, an Ocean prefab is added to a scene in edit-mode when building the scene. First, the appearence of the ocean is tweaked to match the mood of the scene by changing the parameters

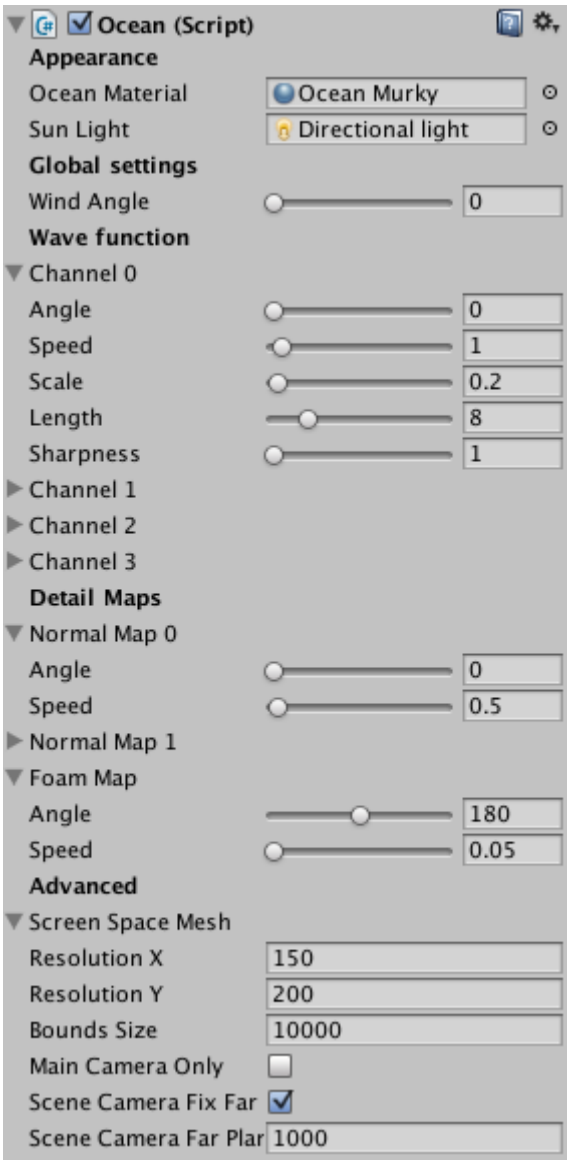
of the Ocean material. The wave function of the Ocean component is then set up so that the large scale waves animates in a believable manner.

During runtime, specific parameter values can be changed using the scripting interface in order for the ocean to change its appearence over time or as a response to events that occur in the game.

5 Components

5.1 Ocean component

The Ocean component controls all the parameters of the water simulation and makes sure that the ocean is rendered correctly. When working with the Ocean prefab you should only need to change the properties of the Ocean component, as it updates the other components accordingly.



Inspector view of the Ocean component

Ocean Material

The material used for rendering. Only materials that use the "Ocean Toolkit/Ocean Shader" can be used. A set of presets can be found in Core/Presets/.

Sun Light

The Light component used to represent the sun. If set to None, the sun is assumed to be in zenith.

Wind Angle [degrees]

A global setting that controls at what angle all other angle parameters start at.

Channel 0-3

A group of settings for wave channel 0 through 3. Each group controls one wave repeated across the ocean.

Channel 0-3 ⇒ Angle [degrees]

The direction of the wave in the XZ-plane.

Channel 0-3 ⇒ Speed [units/s]

The speed of the wave.

Channel 0-3 ⇒ Scale [units]

The maximum height of the wave.

Channel 0-3 ⇒ Length [units]

The distance between two wave crests.

Channel 0-3 ⇒ Sharpness

A value that controls the sharpness of the wave crest.

Normal Map 0-1

A group of settings for normal map 0 and 1. Each group controls how one normal map scrolls across the ocean.

Normal Map 0-1 ⇒ Angle [degrees]

The direction of the normal map movement in the XZ-plane.

Normal Map 0-1 ⇒ Speed [units/s]

The speed of the normal map.

Foam Map ⇒ Angle [degrees]

The direction of the foam map movement in the XZ-plane.

Foam Map ⇒ Speed [units/s]

The speed of the foam map.

Screen Space Mesh

A group of settings for the screen space mesh that is projected onto the water surface and used for rendering. This “projected grid” makes sure that the triangles used to render the ocean is evenly spaced across the viewport in screen space. The mesh is a rectangular quad.

Screen Space Mesh ⇒ Resolution X

The number of quads along the horizontal screen axis.

Screen Space Mesh ⇒ Resolution Y

The number of quads along the vertical screen axis.

Screen Space Mesh ⇒ Bounds Size

The bounds of the mesh in world space. This value needs to be large in order for Unity not to cull the game object when rendering.

Main Camera Only

If enabled, only cameras tagged with “MainCamera” will render the ocean. Useful for debugging purposes.

Scene Camera Fix Far Plane

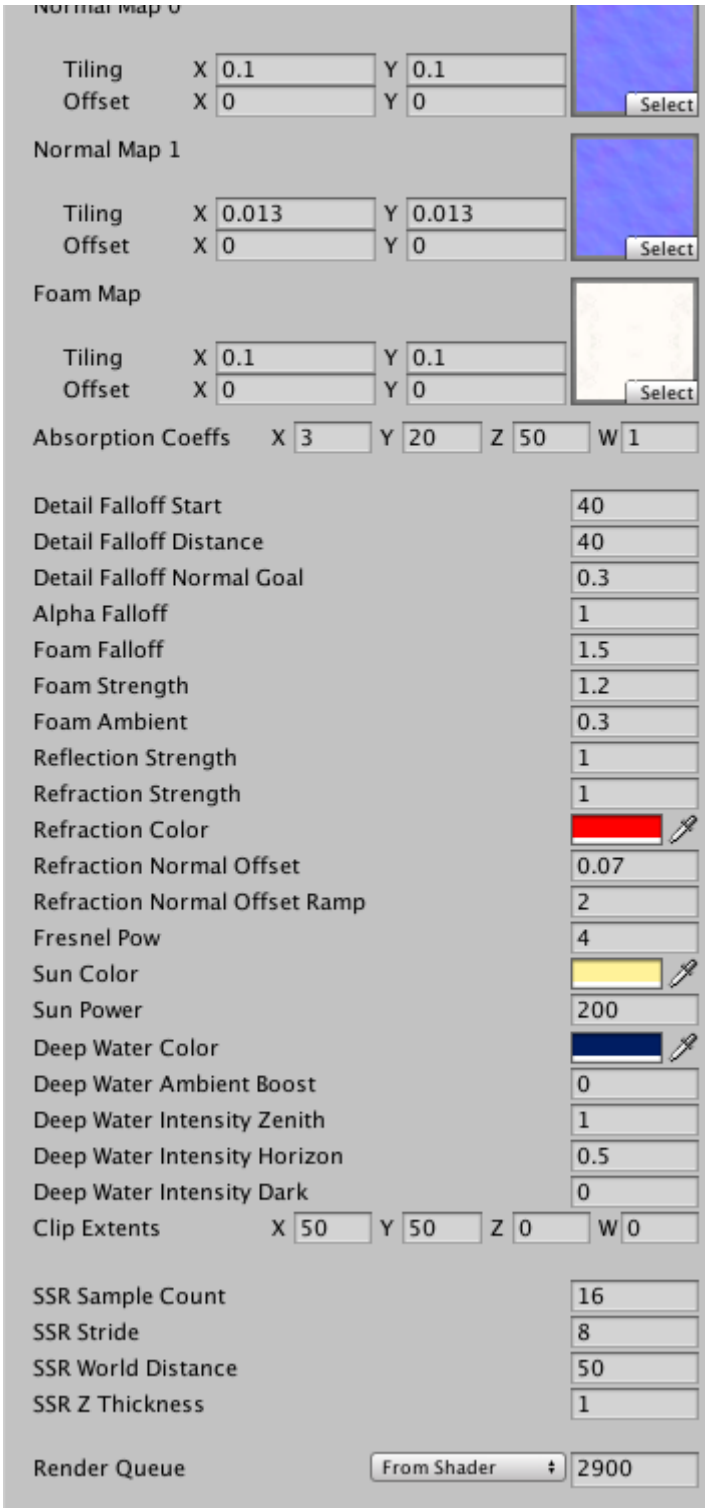
If enabled, the far plane of the scene view will be fixed to the value set to “Scene Camera Far Plane”.

Scene Camera Far Plane

If “Scene Camera Fix Far Plane” is enabled, this is the value that the scene view far plane will be set to before rendering.

5.2 Ocean material

The Ocean material controls the appearance of the ocean. Any material that uses the ocean shader (“Ocean Toolkit/Ocean Shader”) can be used in the Ocean component. To get an idea of what is possible, there are a number of example materials in the Core/Presets/ folder.



Inspector view of the Clear Deep Water Blue preset

Reflection [Off, Sky Only, Screen Space Raytracing]

Determines the quality of the reflections. Off: No reflections. Sky Only: Reflection probes are sampled once. Screen Space Raytracing (SSR): The reflection ray is traced in screen space to find the exact intersection with the environment. When using SSR, not every pixel along the ray is taken into account in order to improve performance, but since the water surface is usually not flat the effect is plausible.

Refraction [Off, Color, Normal Offset]

Determines the quality of the refraction. Off: No refraction. Color: The "Refraction Color" property will decide the color of underwater geometry. Normal Offset: The underwater color will be sampled at points offset by the normal of the water surface at each location.

Clip Bounds

Determines whether or not the bounds of the water volume, as defined by the "Clip Extents" property, should be clipped.

Normal Map 0

The first normal map that will be used for small scale wave detail.

Normal Map 1

The second normal map that will be used for small scale wave detail.

Foam Map

The texture that will be used to give detail to the foam. Currently, only the alpha channel is used.

Absorption Coeffs

This property determines the color of the ocean. Each value indicates at what distance light (of a particular color) travelling through the water is completely absorbed. The X, Y and Z of the vector correspond to the Red, Green and Blue color channels respectively. (The W component is ignored) If X is set to 0.3, for example, the Red color component of the seafloor will be gone at a depth of 0.3 units. That is, the red tint of the seafloor will fade out very quickly as the geometry along the shore decends into the water.

Detail Falloff Start [world space units]

The distance from the camera where the waves start to fade out.

Detail Falloff Distance [world space units]

The distance it takes for the waves to completely fade out from the “Detail Falloff Start” point.

Detail Falloff Normal Goal [0 to 1]

A value indicating the amount of normal map influence at distances over (“Detail Falloff Start” + “Detail Falloff Distance”).

Alpha Falloff [world space units]

The depth at which the ocean should start to fade out when interacting with world geometry.

Foam Falloff [world space units]

The depth at which foam should start to appear when interacting with world geometry.

Foam Strength

A value indicating the intensity of the foam.

Foam Ambient [0 to 1]

A value indicating the intensity of the foam at sunset.

Reflection Strength

A value indicating the intensity of the reflections.

Refraction Strength

A value indicating the intensity of the refraction.

Refraction Color

The color to use for the seafloor when the “Refraction” property is set to “Color”.

Refraction Normal Offset

The amount to offset the sample point along the normal when the “Refraction” property is set to “Normal Offset”.

Refraction Normal Offset Ramp [world space units]

The depth during which the normal offset should ramp up to the value set to “Refraction Normal Offset”, when the “Refraction” property is set to “Normal Offset”. This smooths out the transition betweem geometry above the surface and refracted geometry below the surface.

Fresnel Pow

The power/sharpness of the fresnel term. The fresnel term decides how much light is reflected and how much is refracted at any given point on the ocean.

Sun Color

The color of the sun reflections.

Sun Power

The power/sharpness of the sun reflections.

Deep Water Color

The color used for deep water. This is the ambient color of fully absorbed rays of light. (See the “Absorption Coeffs” property)

Deep Water Ambient Boost

A value determining the intensity boost of deep water. Higher values increase the wave definition but too high values gives the effect of a luminescent seafloor.

Deep Water Intensity Zenith

Determines the intensity of the deep water color when the sun is in zenith.

Deep Water Intensity Horizon

Determines the intensity of the deep water color at sunrise.

Deep Water Intensity Dark

Determines the intensity of the deep water color when the sun is opposite to zenith (ie. when the light travels straight up).

Clip Extents

The extents of the water volume used for clipping when the “Clip Bounds” property is enabled.

SSR Sample Count

The maxmimum number of texture samples that should be used to determine whether or not a reflection ray intersects an object, when the “Reflection” property is set to “Screen Space Raytracing”.

SSR Stride

The number of pixels to skip (in each iteration) when traversing the reflection ray, when the “Reflection” property is set to “Screen Space Raytracing”.

SSR World Distance

The maximum world space distance to travel on the reflection ray, when the “Reflection” property is set to “Screen Space Raytracing”.

SSR Z Thickness

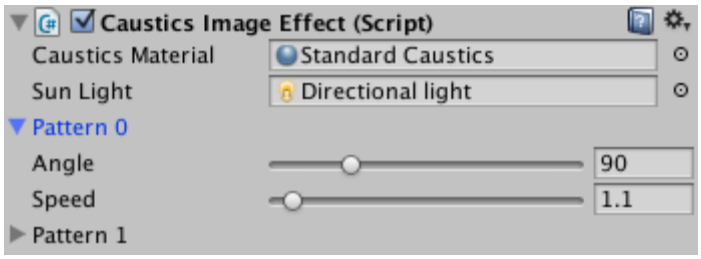
The assumed thickness of hidden parts of objects in the scene, when the “Reflection” property is set to “Screen Space Raytracing”.

5.3 Caustics Image Effect component

The Caustics Image Effect component adds a caustics effect to underwater parts of a scene to any given camera it is attached to. The effect is most notably used for tropical settings but can be an effective depth que even in murky conditions. The Caustics Image Effect component controls the rendering and animation of the caustics.

Caustics Image Effect enabled

Caustics Image Effect disabled



Inspector view of the Caustics Image Effect component

Caustics Material

The material used for rendering. Only materials that use the “Ocean Toolkit/Caustics Shader” can be used. See the Standard Caustics preset located in Core/Presets/.

Sun Light

The Light component used to represent the sun. If set to None, the sun is assumed to be in zenith.

Pattern 0-1

A group of settings for pattern texture 0 and 1. Each group controls how one pattern scrolls across the ocean.

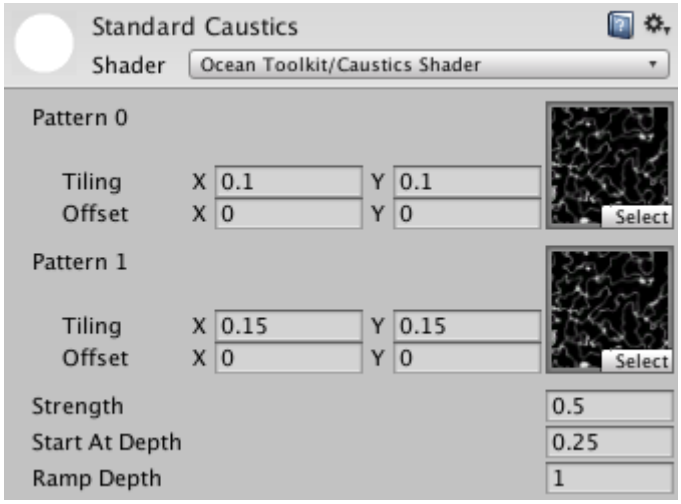
Pattern 0-1 ⇒ Angle [degrees]

The direction of the pattern movement in the XZ-plane.

Pattern 0-1 ⇒ Speed [units/s]

The speed of the pattern.

The Caustics material controls the appearence of the caustics and the size of the scrolling patterns.



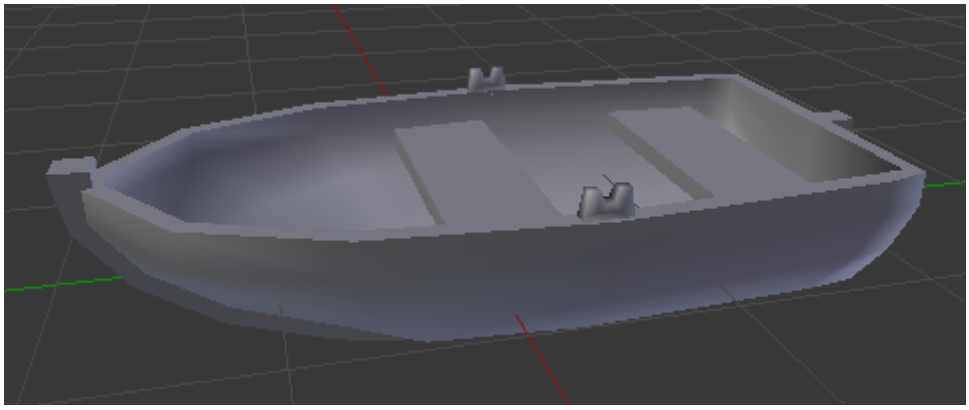
Inspector view of the Caustics material

Pattern 0	The first pattern texture.
Pattern 1	The second pattern texture.
Strength	The intensity of the caustics.
Start At Depth	The depth at which the caustics start to appear.
Ramp Depth	The distance it takes for the caustics to reach full intensity.

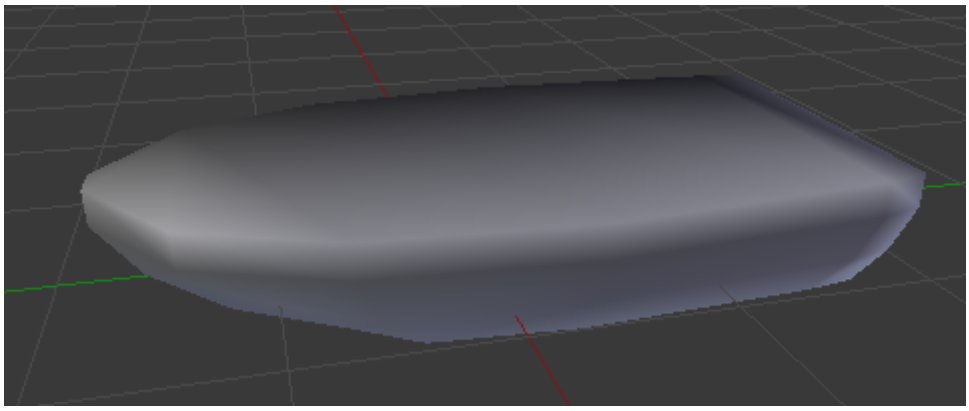
5.4 Ocean Mask material

The Ocean Mask material is used to prevent the ocean from being rendered through certain game objects. For example, the row boat in the example scenes does not have a deck and if nothing was done to prevent it, the ocean would show up through the bottom of the boat. In the example scenes, the Ocean Mask material is used to overwrite pixels in the depth buffer (right before the ocean is rendered) so the ocean behaves *as if* the boat had a deck. The shader only writes to the depth buffer so there will be no visual change to the scene.

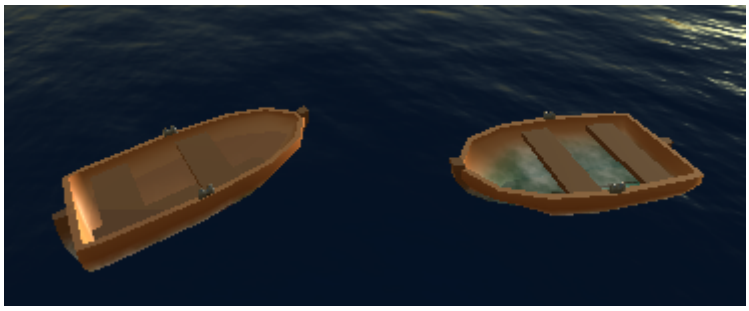
In order to use the Ocean Mask with a boat, first create a mask mesh where the boat has a deck. Then, in Unity, duplicate the hull game object of the boat and rename it to “Ocean Mask”. Then switch the Mesh of this new game object to the mask mesh and the material to the Ocean Mask material. This is the procedure I used for the row boat in the example scenes. (The model is called SmallBoat.fbx and the mask is in SmallBoatMask.fbx)



Small boat mesh



Corresponding mask mesh



With and without mask comparison

6 Scripting reference

6.1 Ocean component

In order to change the appearance or get information from the ocean during runtime, you can write a custom script that changes the properties or calls functions of the Ocean component.

To get a reference to the ocean component, use the following code and make sure that the script is attached to the Ocean game object.

```
public void Start()
{
    OceanToolkit.Ocean o = GetComponent< OceanToolkit.Ocean >();

    // Now use o
    o.OceanMaterial = ...;
}
```

Here is a list of the properties and functions of the Ocean component.

Name	(Return) Type	Property/Function	Note
OceanMaterial	Material	Property	See section 4.1
SunLight	Light	Property	See section 4.1
WindAngle	float	Property	See section 4.1
WaveAngle0	float	Property	See section 4.1
WaveAngle1	float	Property	See section 4.1
WaveAngle2	float	Property	See section 4.1
WaveAngle3	float	Property	See section 4.1
WaveSpeed0	float	Property	See section 4.1
WaveSpeed1	float	Property	See section 4.1
WaveSpeed2	float	Property	See section 4.1
WaveSpeed3	float	Property	See section 4.1
WaveScale0	float	Property	See section 4.1
WaveScale1	float	Property	See section 4.1
WaveScale2	float	Property	See section 4.1
WaveScale3	float	Property	See section 4.1
WaveLength0	float	Property	See section 4.1
WaveLength1	float	Property	See section 4.1
WaveLength2	float	Property	See section 4.1
WaveLength3	float	Property	See section 4.1
WaveSharpness0	float	Property	See section 4.1
WaveSharpness1	float	Property	See section 4.1
WaveSharpness2	float	Property	See section 4.1
WaveSharpness3	float	Property	See section 4.1
NormalMapAngle0	float	Property	See section 4.1

NormalMapAngle1	float	Property	See section 4.1
NormalMapSpeed0	float	Property	See section 4.1
NormalMapSpeed1	float	Property	See section 4.1
FoamMapAngle	float	Property	See section 4.1
FoamMapSpeed	float	Property	See section 4.1
ScreenSpaceMeshResolutionX	int	Property	See section 4.1
ScreenSpaceMeshResolutionY	int	Property	See section 4.1
ScreenSpaceMeshBoundsSize	float	Property	See section 4.1
MainCameraOnly	bool	Property	See section 4.1
SceneCameraFixFarPlane	bool	Property	See section 4.1
SceneCameraFarPlane	float	Property	See section 4.1
GetHeightAt(Vector3 p)	float	Function	Returns the water surface height at the point p (both in world space)

6.2 Ocean material

The Ocean material is a normal Unity material and in order to change its parameters one has to use the scripting interface provided by the Material class. Here follows a few examples. See the Unity documentation for more information.

First, we get a reference to the material through the inspector:

```
public class MyScript : MonoBehaviour
{
    public Material m;

    ...
}
```

We can now use *m* in the script to control the material.

```
float falloffStart = m.GetFloat("ot_DetailFalloffStart");

m.SetFloat("ot_DetailFalloffStart", falloffStart + 10.0f);
```

The Reflection, Refraction and Clip settings are a bit special. Here is how you enable SSR reflections, color refractions and bounds clipping through a script:

```
m.DisableKeyword("OT_REFL_OFF");
m.DisableKeyword("OT_REFL_SKY_ONLY");
m.EnableKeyword("OT_REFL_SSR");

m.DisableKeyword("OT_REFR_OFF");
m.EnableKeyword("OT_REFR_COLOR");
m.DisableKeyword("OT_REFR_NORMAL_OFFSET");

m.DisableKeyword("OT_CLIP_OFF");
m.EnableKeyword("OT_CLIP_ON");
```

Here is a list of the properties of the material.

Name	Type	Note
ot_NormalMap0	Texture	See section 4.2
ot_NormalMap1	Texture	See section 4.2
ot_FoamMap	Texture	See section 4.2
ot_AbsorptionCoeffs	Vector	See section 4.2
ot_DetailFalloffStart	float	See section 4.2
ot_DetailFalloffDistance	float	See section 4.2
ot_DetailFalloffNormalGoal	float	See section 4.2
ot_AlphaFalloff	float	See section 4.2
ot_FoamFalloff	float	See section 4.2
ot_FoamStrength	float	See section 4.2
ot_FoamAmbient	float	See section 4.2
ot_ReflStrength	float	See section 4.2
ot_RefrStrength	float	See section 4.2
ot_RefrColor	Color	See section 4.2
ot_RefrNormalOffset	float	See section 4.2
ot_RefrNormalOffsetRamp	float	See section 4.2
ot_FresnelPow	float	See section 4.2
ot_SunColor	Color	See section 4.2
ot_SunPow	float	See section 4.2
ot_DeepWaterColor	Color	This is the raw color, before it is lit. See section 4.2.
ot_DeepWaterAmbientBoost	float	See section 4.2
ot_DeepWaterIntensityZenith	float	See section 4.2
ot_DeepWaterIntensityHorizon	float	See section 4.2
ot_DeepWaterIntensityDark	float	See section 4.2

ot_ClipExtents	Vector	See section 4.2
ot_SsrSampleCount	float	See section 4.2
ot_SsrStride	float	See section 4.2
ot_SsrWorldDistance	float	See section 4.2
ot_SsrZThickness	float	See section 4.2

7 Example scenes

7.0.1 Basic

Simple scene showing the ocean as it comes in the prefab. 3 static game objects for reference.

7.0.2 Murky

Scene with polluted water and a calm wave function. A number of static game objects for reference.

7.0.3 Ocean Mask

Calm sunset scene with no wave function. 2 small boats show the effect of the Ocean Mask material.

7.0.4 Island

Detailed scene showing the ocean and in a tropical island setting. This scene shows off the refraction, reflection and caustics effect the most.

8 Known Limitations

- There is currently no image effect for when the camera is below the water surface.

9 Contact

Please let me know if you run into any problems when using the toolkit or if you have feedback on how I can improve it in the future. I am also interested in seeing projects that use any of my toolkits in practice!

Website: https://gustavolsson.com/

Contact: https://gustavolsson.com/contact/