# Stylized Water Shader

by Staggart Creations

# 1. Stylized Water Shader

Thank you for purchasing the Stylized Water Shader!

**Please consider rating the package through** your download list **or leave a review at** the store page **once you're familiar with it.**

Your feedback is instrumental to the future of this package!

Rate    Review

Feedback and suggestions can be made in the forum thread:

http://forum.unity3d.com/threads/stylized-water-shader-desktop-mobile.430118/

This manual is intended to clarify the use of the package and should be consulted for troubleshooting.

# 2. Getting started

1. Import the Unity package through the Asset Store download manager (Found under Window->Asset Store)
2. Drag one of the prefabs found under Assets/StylizedWaterShader/Prefabs into your scene.
3. Add the "EnableDepthBuffer" script to your main camera, this ensures that the depth and intersection effects work.
4. Use the StylizedWater script component to start modifying it to your liking.

Alternatively:

1. Create and assign a material to your water mesh and assign a shader found under */StylizedWater*.
2. Add the StylizedWater script component.
3. Add the "EnableDepthBuffer" script to your main camera.

**Note:** It is advised to create your own materials, so your settings are not accidentally overridden when updating the package.

If you're uncertain what a parameter is for, click the help icon to toggle descriptions for that section
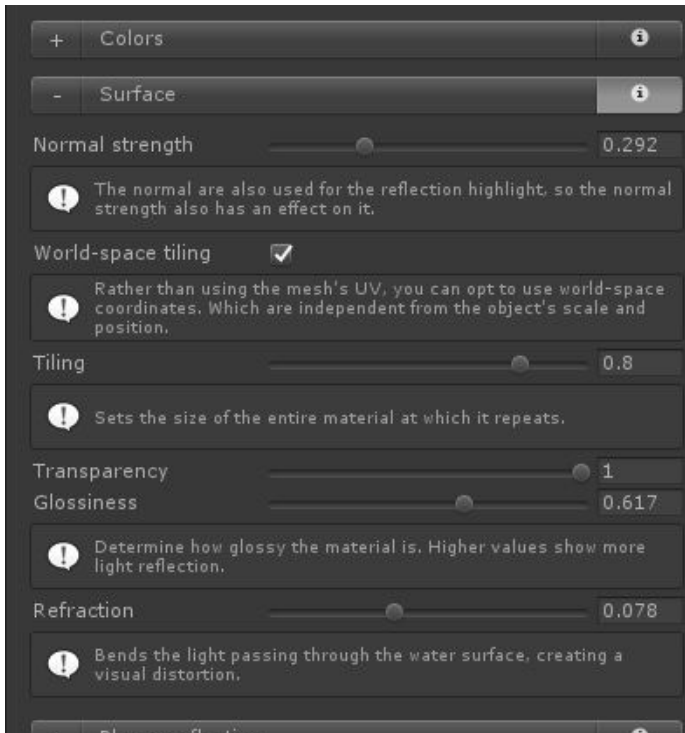
## 3. Support

Should you run into any issues or have questions/feedback, please do not hesitate to reach out! I will do my best to respond as soon as I can.

Unity forums thread: http://forum.unity3d.com/threads/stylized-water-shader-desktop-mobile.430118/

E-mail: contact@staggart.xyz

Twitter: @JonnyIO

## 4. Material instances

**Note: This workflow has been made obsolete since version 1.30. You now only need a regular material instance, since the inspector bakes PNG files now, rather than use Substance output textures.**

Normally, to create a variation of a material you'd simply create another material with the same shader. However, since this shader relies on certain bitmaps generated by a Substance, any material instance should be accompanied by a Substance material instance.
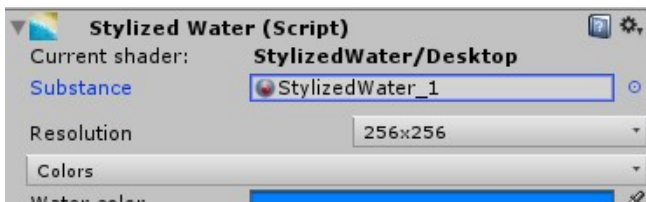
This is only necessary if you want to have different wave- or intersection styles for another water material.
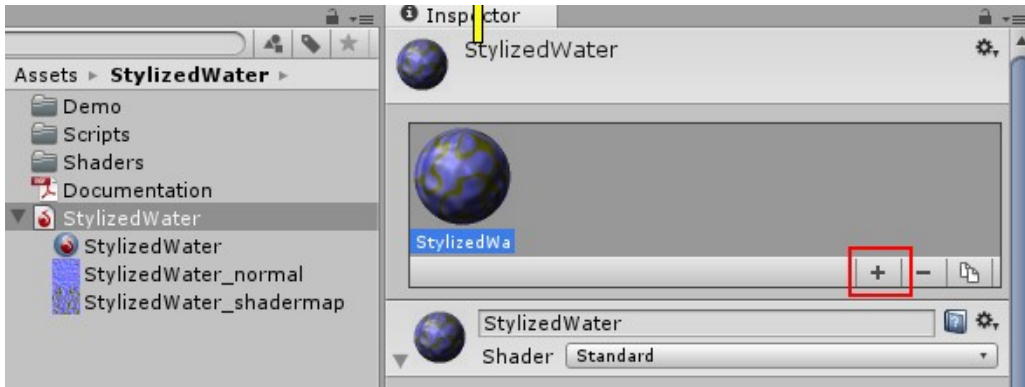
Select the Substance container and press the '+' icon to create a new instance.



You can assign this new material to Substance field in the inspector:



# 5. Shader Forge

If you own Shader Forge you are free to modify the shaders at your will. However, please do so at your own risk. The custom inspector relies on certain shader keywords to be known, and will likely break should these be removed from a shader. It should be noted that the inspector is not required for the shader to function, it is merely a front-end for the various parameters.

Please note that the shaders are currently being rewritten, so Shader Forge editing will not be possible in a later update.

# 6. Parameter glossary

Note: not every parameter is available in all shaders versions.

| Feature | Desktop | Mobile Advanced | Mobile Basic |
|---------|---------|-----------------|--------------|
| Normal map | V | V | V |

| | | | |
|---|---|---|---|
| Transparency | V | V | V |
| Glossiness | V | V | V |
| Intersection effect | V | V | V |
| 7 different styles | V | V | V |
| World-Space tiling | V | V | V |
| Fresnel | V | V | X |
| Depth | V | V | X |
| Surface highlights | V | V | X |
| Wave animation | V | V | X |
| Multi-light support | V | X | X |
| Advanced surface highlights | V | X | X |
| | | | |
| Refraction | V | X | X |
| Wave heightmap | V | X | X |
| Tessellation | V | X | X |
| Cubemap reflection | V | X | X |

## 6.1. Colors

| | |
|---|---|
| **Water color:** | The most self-explanatory parameter. |
| **Fresnel color:** | Controls the color for the glancing reflection, the Alpha channel controls the intensity of the Fresnel effect. |
| **Rim color:** | Color of the foam intersection. Alpha controls its transparency |
| **Wave tint:** | Controls how bright or dark high waves should be tinted based on the heightmap |

## 6.2. Surface

| | |
|---|---|
| **Normal strength:** | Increases or decreases the normal map intensity<br>**Note:** this does not work when using custom normal maps |
| **Tiling:** | The amount of tiling |
| **Transparency:** | Controls the opacity of the surface. |
| **Glossiness:** | Defines how reflective the water looks. Fully reflective water shows the sun reflecting in the waves and the cubemap reflection. |
| **Fresnel:** | Exponentiate the Fresnel effect, essentially moving it closer or further away from the horizon. |
| **Refraction:** | Simulates light bending as it moves through the waves. |
| **Reflection cubemap:** | Assign a cubemap to be reflected in the water, depending on the glossiness amount. Real-time reflection is not supported. |

## 6.3. Intersection

| | |
|---|---|
| **Custom texture:** | Assign a grayscale texture to be used for the intersection effect. |
| **Style:** | Dropdown menu to choose one of the pre-built styles from. |
| **Size:** | Controls how wide or deep the intersection effect should be. |
| **Falloff:** | Adds a linear transition to the intersection. |

| | |
|---|---|
| **Distance:** | Add a smooth offset for the effect |
| **Tiling:** | The amount of tiling for the intersection texture. The buttons here can be used to increase or decrease the tiling. |

## 6.4. Surface highlights

| | |
|---|---|
| **Use intersection texture** | Use the intersection texture for the surface highlight effect. |
| **Cross-panning textures** | When using the intersection texture, this toggle adds a secondary cross-panning texture. |
| **Opacity:** | Controls the overall opacity of the highlight effect. This effect is derived from the normal map. |
| **Size:** | Dynamically shrink the size of the effect. |
| **Tiling:** | Amount of tiling. |

## 6.5. Depth

| | |
|---|---|
| **Depth:** | Adds murky water the deeper the water is. Note: This effect depends on the amount of transparency. |
| **Darkness:** | The depth color is derived from the water color and can be brightened or darkened here. |

## 6.6. Waves

| | |
|---|---|
| **Custom normal map:** | Assign a custom normal map to be used for the waves. |
| **Style:** | Dropdown menu to choose one of the pre-built normal maps from. |
| **Heightmap** | Dropdown menu to choose one of the pre-built heightmaps from. |
| **Strength:** | Controls the height of the waves. If you need higher waves, you can scale your water plane up on the Y axis. |
| **Size:** | Controls the size of the waves. |

## 6.7. Other

**Tessellation:**

Add polygons to the mesh to create a more detailed wave animation effect

## 7. Mobile

The mobile shader is of stripped refraction, for performance. While it is optimized for mobile devices, the features that provide the water effects are still quite taxing. Please keep this in mind when developing.

By default, the camera doesn't render the depth map on mobile platforms, which is required for the shader to work. To fix this, attach the included "EnableDepthBuffer" script to your camera and you're good to go!

Two shader variants are provided:

**Advanced:** Same features as desktop variant expect for refraction. With added support for OpenGL 3.0 and iOS Metal.

**Basic:** Bare minimum, no cross-panning of textures, directional light only.

Both require a minimum Shader Model of 3.0.

## 8. Troubleshooting

- If you're using Unity's Ambient Occlusion image effect, make sure the EnableDepthBuffer script is executed after it.

**Waves aren't high enough, even with the highest "Wave strength" value**

In this case, your water mesh must be very big. If you need higher waves than the shader value allows for, increase the scale of your mesh on the Y-axis to the same effect

**Waves look very pointy, triangular**

This is an indication that the vertex-density/polycount of your water mesh is too low. Add subdivisions to your mesh to increase the wave fidelity.

Much like a pin-art toy; the more pins it has, the more detailed the cast becomes.

**Water surface flickers white and the intersection effects disappears**

This is usually caused if there is no light source in the scene, or your scene-view has lighting switched off. This also occurs when using an orthographic or 2D camera, which is not supported at this time.

**Water shows banding artifacts**

Some of the shader features rely on the rendered depth buffer. If your device does not render this accurately banding artifacts will show. This appears to happen on some devices using a Mali-450MP4 GPU. And has been fixed in version 1.24. Should it still occur, a potentially solution is to set the display buffer to 32-bit as some users reported this fixed the issue.
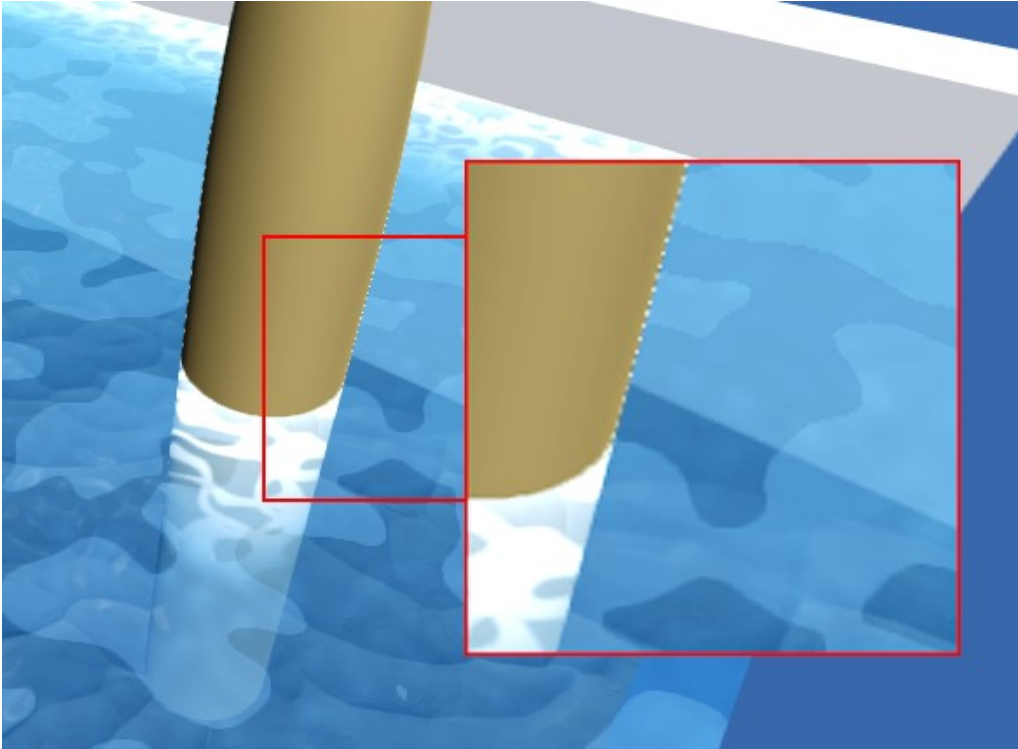
**The new waves/intersection styles added in the update aren't visible in the dropdown**

Please remove and re-add the Stylized Water script component to force a refresh of the list

**Objects show a pixelated white outline**

This occurs when using Unity's built-in MSAA and be remedied by using an Anti-Aliasing image effect instead.
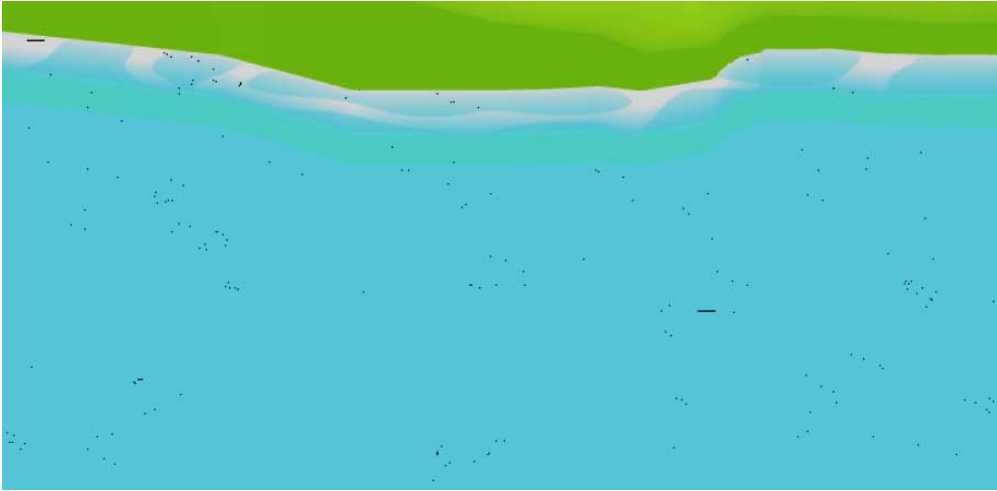
**Surfaces looks jagged on mobile**

This is due to compression of the shader textures. You can override this compression in the inspector under the "Advanced" tab. This will impair a slightly larger memory cost as the two textures will be larger in size.



**(iOS) Black dots appear on the water**

You can elevate this issue by doing taking the following steps:

- Unset the Sun object in the Lighting tab, if set
- Raise the Shadow Bias value on your Directional Light

## 8.1. Playstation 4

In order to make the shaders compatible with the Playstation 4, a few modifications have to be made to the shader files, namely:

- Replace all instances of the word "*COLOR*" with "*SV_Target*"
- Remove any lines that say "*#pragma exclude_renderers gles3 metal d3d11_9x xbox360 xboxone ps3 ps4 psp2*"

After these modifications the shaders will compile correctly on the console.

## 9. Refunds

Please refer to our [refund policy](#)

## 10. Credits

The assets included with the demo scene are from a student project called "[The Torch](#)", however you are free to use these under the same license.

Credits go to:

Hang Nguyen

Kassym Kuskimbayev

Ulvis Bariss