Stylized Water Shader

by Staggart

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.
2. Drag one of the prefabs into your scene.
3. Add the "EnableDepthBuffer" script to your main camera.
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. 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/

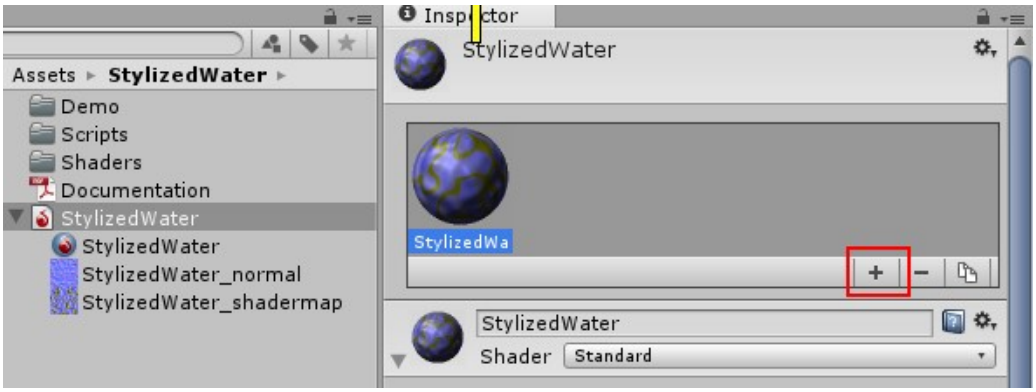
E-mail: contact@staggart.xyz

Twitter: @JonnyIO

4. Material instances

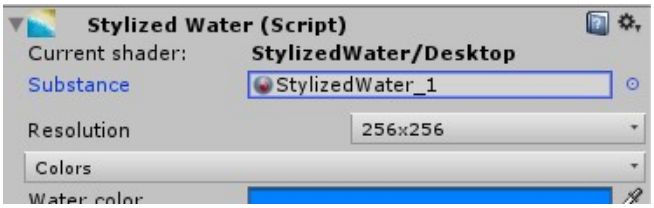
Normally, to create a variation of a material you'd simply create another material with the same shader. However, since this shader relies on certain bitmaps generated by a Substance, any material instance should be accompanied by a Substance material instance.

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.

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

6.2. Surface

Normal strength:	Increases or decreases the normal map intensity **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.
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

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 styles from.
Strength:	Controls the height 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.

Water surface flickers white and the intersection effects disappears

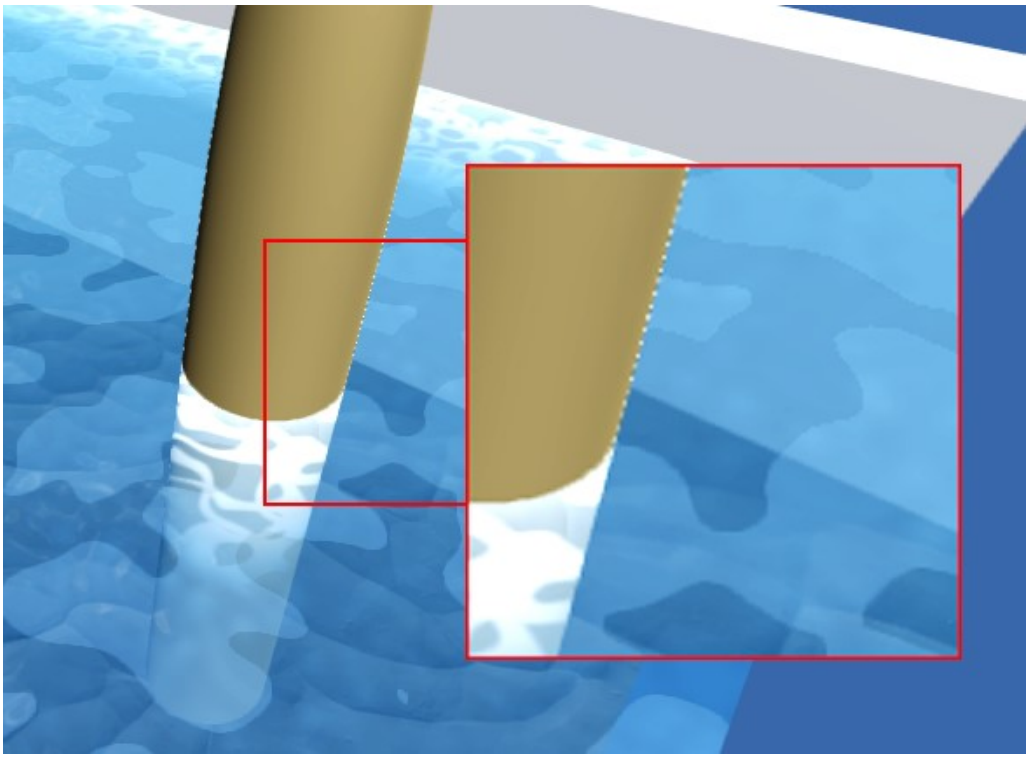
This is usually caused if there is no light source in the scene, or your scene-view has lighting switched off. This also occurs when using an orthographic or 2D camera, which is not supported.

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.

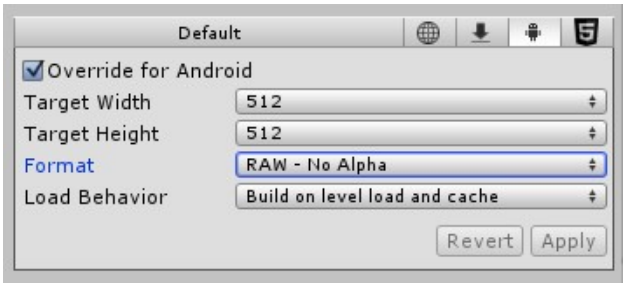
Objects show a pixelated white outline

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 that takes place on mobile platforms. You can override this compression in the Substance material, by setting the format to ‘RAW – No Alpha’. This will impair a slight memory cost as the two textures will be larger in size.



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