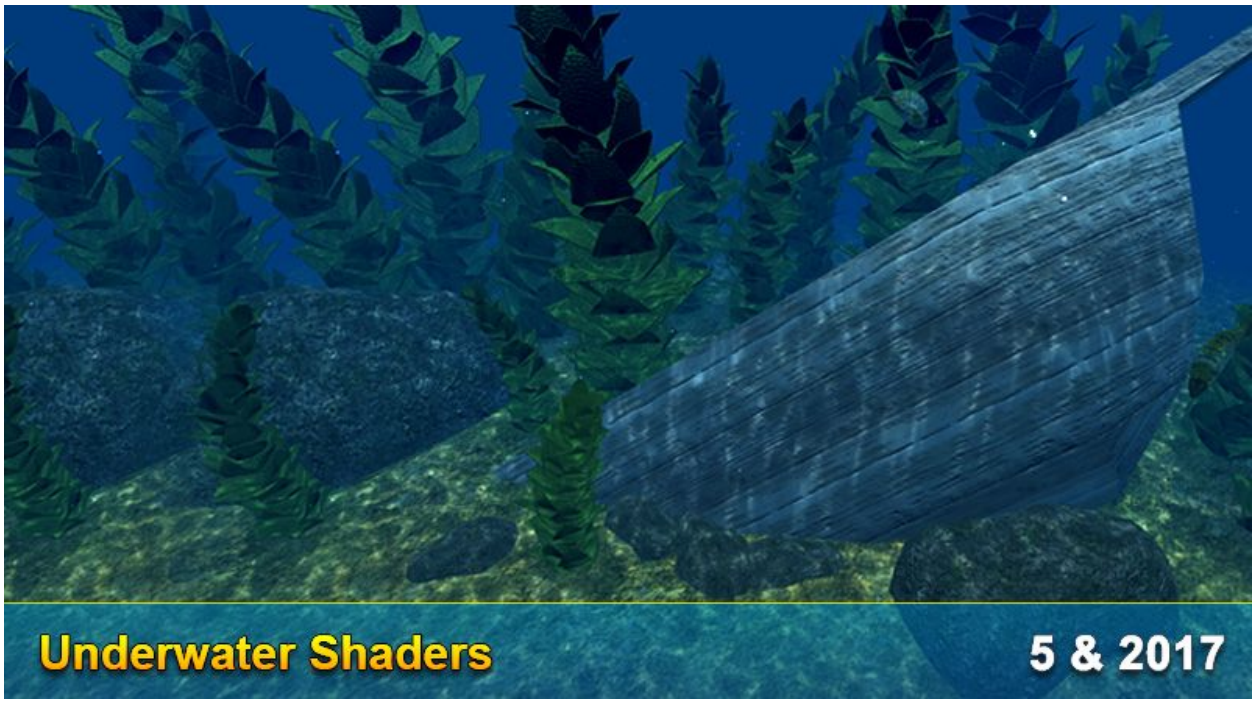
Underwater Shaders - Version 1.5



Thank you for purchasing Underwater Shaders!

Underwater Manager (Optional)

The Underwater Manager prefab can be dragged into your scene. It is a simple game object with the UnderwaterManager.cs script attached to it. Drag as many underwater materials as you would like into the “Materials” property array. All materials that are part of the array will inherit the settings from the manager script. This makes it simple to set the same caustics and fog properties on multiple materials at the same time. Please open the “Shipwreck” example scene to see this in use.

Please note that the Underwater Manager prefab and the UnderwaterManager.cs script are not required for the shaders to work. They are provided as an optional convenient way to manage the properties on multiple underwater materials.

Shaders

All shaders have been written to appear and behave as similar as possible.
Differences between them are as follows:

“Underwater/Standard” *Underwater-Standard*

- Based on Unity 5’s physically plausible shader.
- Supports shadows and bump maps.
- Multiple light types are supported with shadows.
- All lighting and fog is calculated per fragment.
- Includes the standard shader parameters “Smoothness” and “Metallic”
- Does not include all of the features of Unity’s standard shader.

"**Underwater/Desktop**" *Underwater-Desktop*

- Based on Unity’s “Legacy/Bumped Diffuse” shader.
- Supports shadows and bump maps.
- Multiple light types are supported with shadows.
- Non-directional lights don't contribute to the caustic effect.
- All lighting and fog is calculated per fragment.

"**Underwater/Mobile**" *Underwater-Mobile*

- Based on Unity’s “Mobile/Diffuse” shader.
- Supports 1 directional light only (With a shadow).
- All other lights are computed per-vertex (No shadows).
- Non-directional lights don't contribute to the caustic effect.
- All lighting and fog is calculated per fragment.

"**Underwater/Mobile Fast**" *Underwater-Mobile-Fast*

- Calculates all lighting and fog in the vertex function (Fast).
- Does not support shadows or bump maps.
- Supports 1 directional light only.
- May not work well with large distances between vertices.

"**Underwater/Terrain**" *Underwater-Terrain*

- Based on Unity’s “Built In Legacy Diffuse” terrain shader.
- Bump maps are supported.
- Supports shadows from 1 directional light only.
- Non-directional lights don't contribute to the caustic effect.
- All lighting and fog is calculated per fragment.
- *Underwater-Terrain-AddPass* is required when painting with more than 4 textures on the terrain.

"**Underwater/Terrain Fast**" *Underwater-Terrain-Fast*

- Calculates all lighting and fog in the vertex function (Fast).
- Does not support shadows or bump maps.
- Supports 1 directional light only.

- *Underwater-Terrain-Fast-AddPass* is required when painting with more than 4 textures on the terrain.

"**Underwater/Transparent/Mobile**" *Underwater-Mobile-Transparent*
"**Underwater/Transparent/Desktop**" *Underwater-Desktop-Transparent*
"**Underwater/Transparent/Standard**" *Underwater-Standard-Transparent*

- These are the same as their counterparts described above with the addition of transparency via the alpha channel of the main texture.

How to use the terrain shaders

1. Create a new material and assign either "Underwater/Terrain" or "Underwater/Terrain Fast" shader.
2. Select the terrain asset in the hierarchy.
3. Open the terrain settings in the inspector.
4. Change the Material drop down to custom.
5. Drag and drop the material you created into the "Custom Material" slot.

   Note: If you assign a new shader to the terrain material, you may need to temporarily change the material in the "Custom Material" slot to something else and then change it back to see the new shader take effect.

Parameters

- Main Texture (_MainTex)
    - Main color texture. (RGB)
    - Alpha when used with a transparent shader (A)
    - *Does not apply to the terrain shaders.*

- Bumpmap (_BumpMap)
    - Normal bump map.
    - *Desktop shader only.*
    - *Does not apply to the terrain shaders.*

Caustics

- Caustics Texture (_Caustics)
    - Uses a 4 channel image (RGBA).
    - Requires an alpha channel.
    - Each channel is animated separately in the shader.
    - UV independent always projected from world vertical axis.
    - Tiling and offset are ignored.

- Tiling(XY) Offset(ZW) - Overrides texture coords (_CausticsCoord)
    - Use to modify the caustic texture tiling and offset
    - *Required for custom light functions*

- Speed (_CausticsSpeed)
    - Sets the speed of the caustic effect animation.

- Boost (_CausticsBoost)
    - Boost lighting to the caustic effect.
    - A value of 0 simulates a multiplicative blend mode, where the areas not lit by caustics are darkened.
    - A value of 1 simulates an additive blend mode, where the areas not lit by caustics are unchanged and the areas lit by caustics are doubled in intensity.

- Intensity A (_CausticsIntensity0)
    - Intensity for caustics at height (World Y) Position A
    - 0 = No Caustic effect (Basic lambert lighting).
    - 1 = Full caustic effect.

- Intensity B (_CausticsIntensity1)
    - Intensity for caustics at height (World Y) Position B
    - 0 = No Caustic effect (Basic lambert lighting).
    - 1 = Full caustic effect.

- Position A (World Y) (_CausticsPosition0)
    - Position (Height in world Y) for intensity A.

    - Caustic intensity A will fade to B starting at this World Y position.

- Position B (World Y) (_CausticsPosition1)
    - Position (Height in world Y) for intensity B.
    - Caustic intensity B will fade to A starting at this World Y position.

Fog

- Color A (_FogColor0)
    - Color of the fog at world y position A.
    - Independent of Unity's fog.

- Color B (_FogColor1)
    - Color of the fog at world y position B.
    - Independent of Unity's fog.

- Intensity A (_FogIntensity0)
    - Intensity of the fog at world y position A.
    - 0 = No fog.
    - 1 = Full fog.

- Intensity B (_FogIntensity1)
    - Intensity of the fog at world y position B.
    - 0 = No fog.
    - 1 = Full fog.

- Position A (World Y) (_FogPosition0)
    - Position (Height in world Y) for intensity and color A.
    - Fog intensity and color A will fade to B starting at this World Y position.

- Position B (World Y) (_FogPosition1)
    - Position (Height in world Y) for intensity and color B.
    - Fog intensity and color B will fade to A starting at this World Y position.

- Start (_FogStart)
    - Start distance from the camera's position.

- End (_FogEnd)
    - End distance from the camera's position.

Changelog

- Version 1.5
    - Added alpha transparency versions of the Mobile, Desktop and Standard shaders.

- Version 1.4
    - Modified sample scenes.
    - Optimized performance of shaders.

- Version 1.3
    - Added “Underwater/Standard” shader.
    - Added “UnderwaterManager.cs” script and Underwater Manager prefab.

- Version 1.2
    - Added Boost (_CausticsBoost) property.
    - Renamed “Underwater/Mobile” shader from version 1.1 to “Underwater/Mobile Fast”.
    - Added new “Underwater/Mobile” based on Unity’s “Mobile/Diffuse”.
    - Modified “Underwater/Desktop” to support multiple lights.
    - Added “Underwater/Terrain” and “Underwater/Terrain Fast”.
    - Added “Pool” and “Terrain” demo scenes.

- Version 1.1
    - Changed shaders from Low/High to Mobile/Desktop
    - Added shadows and bump mapping to desktop shader
    - Added height (World Y) based intensity to caustics
    - Added height (World Y) based color and intensity to fog
    - Modified demo scene to reflect changes

- Version 1.0
    - Initial release