

# PLAYWAY WATER



Document Version 1.0

# Contents

Introduction.....	3
Getting started .....	3
Create Water Component.....	3
Water Profile .....	3
Water Camera .....	4
See it working.....	4
Quality and performance .....	5
Global quality settings.....	5
Individual water settings.....	6
Hardware limitations .....	6
Water profiles.....	7
Components overview .....	8
Water .....	8
Water Waves FFT .....	9
Water Waves Gerstner.....	9
Water Foam .....	10
Water Spray .....	10
Water Volume Subtract .....	11
Water Volume Add.....	11
Water Physics.....	11
Water Float .....	12
Planar Reflections .....	12
Water Camera .....	13

## Introduction

During the development of our water system we have put a lot of care to make it like Unity itself. It's very rich in features, but still friendly in use, practical and multiplatform. By using it, you can create outstandingly looking fluids on all devices ranging from cheap mobile phones to gaming PCs.

We have developed and tested this system thoroughly for months in few projects, but as new features are added, bugs may appear. If you will experience any problems or would like to suggest a feature, please contact me at mbartniczak ( at ) gmail.com. I will be glad to help.

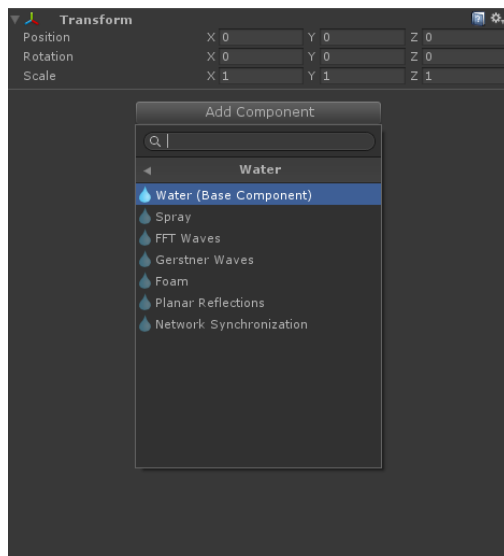
Please remember that all mobile platforms are currently in BETA. Bugs are to be expected on them.

## Getting started

We strongly recommend to start by analyzing the sample scene located at "PlayWay Water/Samples/Galleon Demo.unity". It will present you some of the ways to use our water system in real-world situations.

To add functional water to your scene, you have to perform three steps below.

### Create Water Component

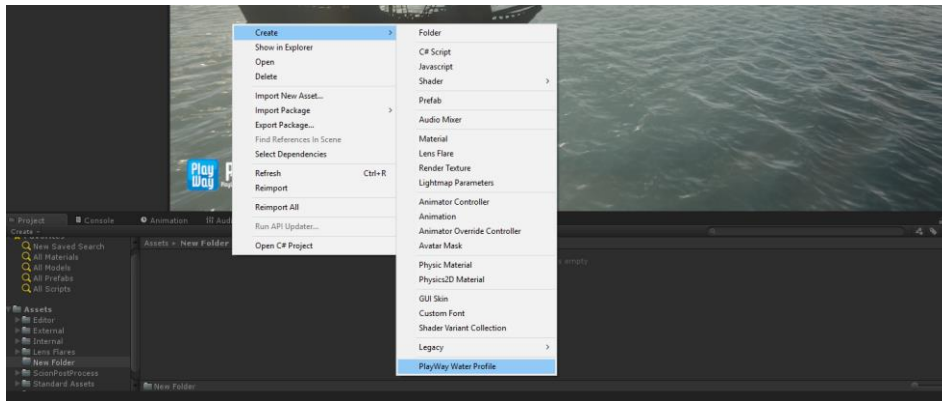


Create empty game object and add Water component. You can do so by selecting "Water/Water (Base Component)" in the menu presented on the left. Few additional components will be added along with it by default. You may remove them later if you don't need them.

This and all other water-related components control the process of rendering the water. In their inspectors you define purely technical means like the type of meshes to use, how reflections should be produced etc.

### Water Profile

Next thing to do is to assign some Water Profile to the water that you have just created. You may use one of the default profiles located in your assets library at "PlayWay Water/Samples/Profiles" or create new ones by choosing "Create/PlayWay Water Profile".



Water profiles are the place to define your water look and feel – colors, intensities, wave types etc. It's the artists domain.

Profiles are similar in concept to materials, but they may be blended together at runtime and also describe some non-material properties like types of waves produced by the simulation or the physical density.

## Water Camera

Each camera that is supposed to see water needs “Water Camera” component for that. You may add it by selecting “Water/Water Camera” in Unity’s component menu.

Although it could be done automatically for each camera in scene, we have wanted to make that a conscious decision for few reasons. Beyond that it allows you to configure a few settings per camera.

Editor cameras are the exception to this rule and are automatically included for the water rendering.

## See it working

You should see your water now. Hit “Play” to see how it animates.

## Quality and performance

If you don't get decent enough FPS at the default settings on your target platform, don't worry. You may fix that in few ways.

Performance depends on the methods used to render water, which in turn are driven by the settings. As most settings may be set in few places, a final value actually used is always determined by choosing the lowest one of these:

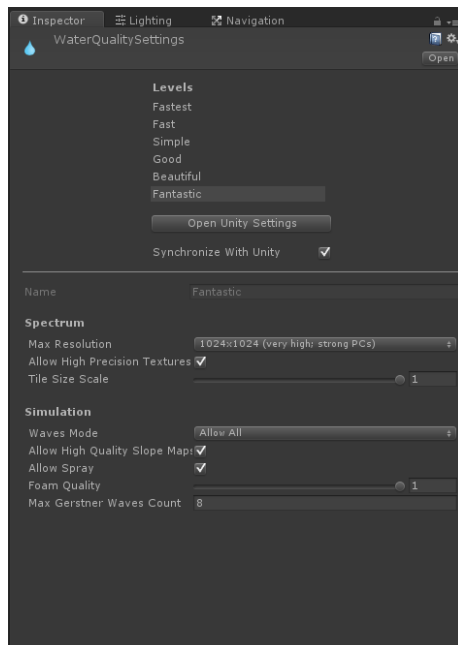
- Individual water component setting.
- Global water quality setting.
- The best setting that the current hardware may support.

Individual component settings allow you differentiate desired quality level for each fluid. For example, you will generally want to use advanced simulation methods for oceans, but keep them simple for puddles etc.

Global settings allow you to put some global limits for all fluids in your application depending on the target hardware.

### Global quality settings

You may get to them by selecting "Edit/Project Settings/Water".



Water quality settings are organized just like the Unity quality settings. You have few quality levels defined here and for each of them you may customize some performance-related settings.

By default, names, order and current quality level selection are synchronized between Unity settings and the PlayWay Water, but you may choose to not do so.

**Spectrum - Max Resolution:** All simulations will be performed at most at this resolution. It's performance-critical setting.

**Spectrum – Allow High Precision Textures:** Determines if high precision textures (32 bit floating point per channel) may be used, if supported by the target hardware. Please refer to the table below to choose the optimal value:

RESOLUTION	HIGH PRECISION TEXTURES
512 OR LOWER	False – using them makes no difference.
1024	Disputable.
2048+	True – low precision textures produce noticeable artifacts.

**Spectrum – Tile Size Scale:** Allows you to globally scale tile size of all water simulations. Setting it below 1.0 will increase quality and compensate drawbacks of using small simulation resolution, but will make the repeating pattern of water more noticeable.

**Simulation – Waves Mode:**

- Allow All – All types of simulations are allowed (FFT or Gerstner).
- Allow Slope FFT – Gerstner waves (low-quality) are allowed and used as a fallback to FFT. FFT may be used only to generate high-quality water normal maps. Supported on all platforms.
- Allow Gerstner - Gerstner waves (low-quality) are allowed and used as a fallback to FFT.
- Disallow All – No waves will be produced.

**Simulation – Allow High Quality Slope Maps:** It is very taxing for performance, but barely noticeable. You may consider enabling it on the highest quality level though.

**Simulation – Allow Spray:** Determines if “Water Spray” component is allowed on this quality level.

## Individual water settings

These are the settings that you find on each water-related component inspector. Following chapters describe them in detail.

## Hardware limitations

Actually used settings may still be lower than what you have chosen, when target hardware puts some limits.

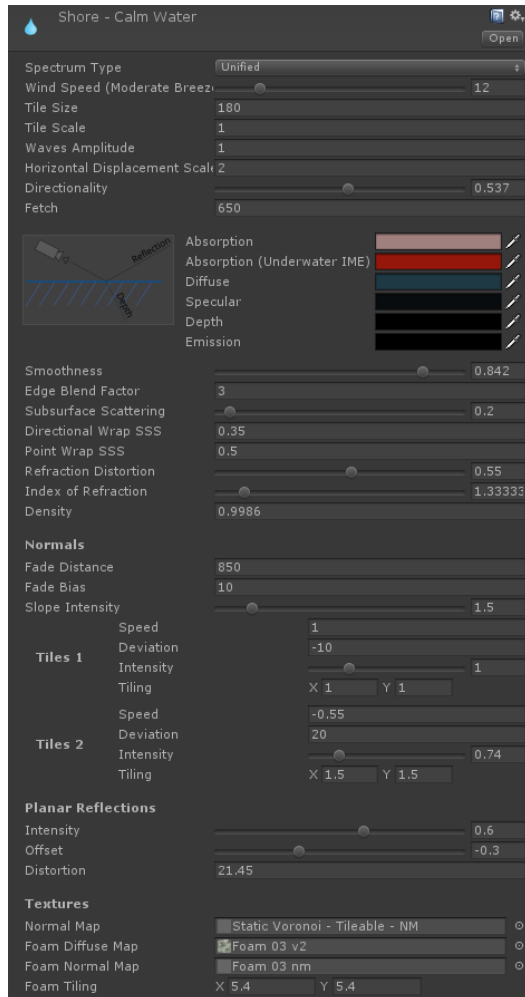
Please refer to the table below for the hardware-limited features. It will give you a general idea of platform limitations, but please keep in mind that some devices may be an exception from this classification.

SHADER MODEL	PLATFORMS	FEATURES
SM 5.0	Most desktops and new consoles	Tessellation, spray, FFT waves displacements and slopes, improved microfacet model, high precision textures.
SM 3.0	Old desktops, consoles and most mobile platforms	FFT waves displacements and slopes, high precision textures
SM 2.0	Extremely old desktops and many mobile platforms	FFT waves slopes.

Other features are generally not platform-limited.

# Water profiles

Water profiles determine the look and feel of the fluid. The best way to understand and correctly configure its parameters is to just play with them a bit and see their effects. Nevertheless, their descriptions are below.



**Spectrum Type:** Two spectrum types are currently implemented: Phillips (legacy) and Unified (recommended).

**Wind Speed:** Wind speed in knots. Hint on the left will display its Beaufort classification.

**Tile Size:** Determines area of water simulation in real world units. Low values increase quality, but make the repeatable pattern of water noticeable. Also, too low values may define area smaller than some of the big waves created by the current wind speed. When that happens a warning will appear.

**Directionality:** Eliminates waves moving against the wind. Should be 0 for oceans.

**Fetch:** It is the length of water in meters over which a wind has blown. Usually a distance to the closest land in the direction opposite to the wind.

**Absorption:** How much light of each color (R, G and B) should be absorbed per meter by the water volume.

**Absorption (Underwater IME):** Same as above, but used by the underwater image effect.

**Edge Blend Factor:** Controls water fading near the intersections with other geometry.

**Subsurface Scattering:** How much light should be scattered back from the underwater. Should be set high in profiles with small and medium waves, low in profiles with big waves.

**Directional Wrap SSS:** Crude approximation of subsurface scattering. Use it along with with the full volumetric scattering defined above to get the best results. This setting is used for directional lights.

**Point Wrap SSS:** Same setting, but for point lights.

**Refraction Distortion:** Determines refraction distortions intensity.

**Index of Refraction:** Refer to the sources around the web to find the correct value for the fluid you need. Should be set to 1.33333 for water.

**Density:** Physical density. Dense fluids apply stronger forces to objects moving through them.

**Normals – Fade Distance and Bias:** Water gets really noisy at a distance and SMAA or FXAA won't handle that. This parameters will let you fade water's normals to avoid this problem.

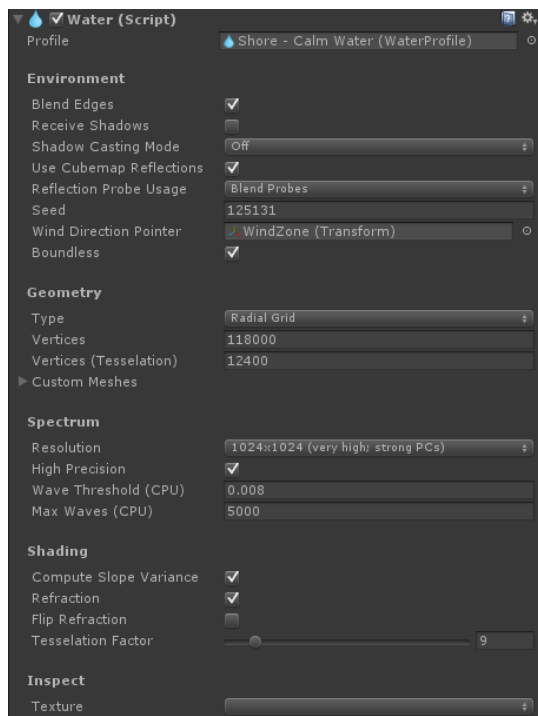
**Normals – Slope Intensity:** Intensity of wave normal when using FFT.

**Normals – Tiles:** Define parameters of two normal map tiles used by the shader.

**Planar Reflections – Offset:** Vertical offset of the planar reflection. Fixes some common artifacts produced by them.

## Components overview

### Water



#### Environment

**Blend Edges:** Makes shader softly mask water near intersections with other geometry.

**Use Cubemap Reflections:** Determines if reflection probes should be used to render reflections on water. May be used in conjunction with other methods of generating reflections.

**Seed:** Setting it to 0 makes water behave differently at each run. Other values make it predictable. It is useful for cutscenes, network games etc.

**Wind Direction Pointer:** Transform that points the wind direction.

**Boundless:** Should the water be considered endless in all directions? Used by physics etc. to determine where the water is.

#### Geometry

**Type:** Possible values are Radial Grid, Projection Grid, Uniform Grid and Custom Meshes. Radial grid is the recommended geometry type. It is flat radial grid placed on the sea level, covering field of view as closely as possible. Projection Grid may be a good choice for platforms not supporting tessellation but creates noticeable artifacts at horizon. This method will be improved in the future, but not all drawbacks of this method may be removed. Uniform grid is used by some internal effects and not recommended in regular use. Custom meshes is the geometry of choice if you plan to create small fluids like puddles. It gives you exact control over vertex count and surface area of fluid.

**Vertices:** Determines vertex count of water if not using custom mesh.

**Vertices (Tessellation):** Same setting, but used for systems with tessellation support. You want it to be lower as tessellation puts lots of well distributed additional vertices.



**Resolution:** Determines quality of waves simulation, foam and spray. Performance-critical. Quality-critical.

**High Precision:** Described in detail in [Global Quality Settings](#).

**Wave Threshold (CPU):** How small waves should be considered by the CPU. Lower values may make physics computations more precise, but also decrease performance. Default setting is pretty low actually. There is little to no reason to decrease it further.

**Max Waves (CPU):** How many waves from the spectrum may be used by the CPU.

### *Shading*

**Compute Slope Variance:** Makes water smoothness depend on the angle and distance of looking in a physically correct manner. If water in your game is viewed from predictable perspectives (like in RTS games) you may disable this and just set water smoothness correctly. Otherwise it's recommended to leave it on.

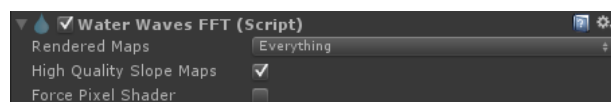
**Tessellation Factor:** How many vertices should be added by tessellation.

### *Inspect*

You may choose most of the water maps here and observe how they animate.

## Water Waves FFT

Produces tens of thousands of waves at a time. Height and displacement generation works only on SM 3.0+. Slope map generation works on all platforms.



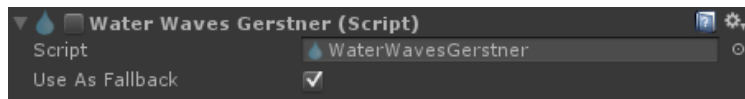
**Rendered Maps:** Determines what maps should be rendered using Fast Fourier Transform. Available choices are: height map, horizontal displacement map and slope map.

**High Quality Slope Maps:** Described in detail in [Global Quality Settings](#).

**Force Pixel Shader:** Disables FFT generation through compute shaders. Not recommended as compute shaders are lot faster.

## Water Waves Gerstner

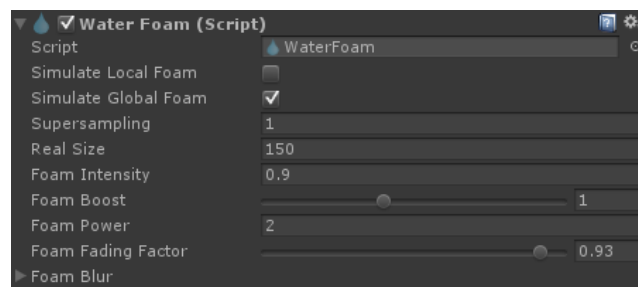
Produces only few waves, but works on everything and is not limited by simulation resolution.



**Use As Fallback:** Makes this component enable itself when Water Waves FFT cannot be used.

## Water Foam

Produces foam.



**Simulate Local Foam:** Performs foam simulation around the main camera. Currently there is no reason to use this. It will be more useful in the future.

**Simulate Global Foam:** Performs foam simulation on the whole simulation tile and repeats result over the whole world. Recommended: on.

**Supersampling:** Simulation resolution multiplier. Recommended: 1.

**Real Size:** Applies only to local foam and determines size of simulation area in world units.

**Foam Intensity:** Intensity scale.

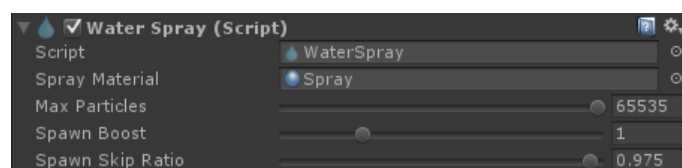
**Foam Boost:** Higher values produce more foam in more places.

**Foam Power:** Foam edges sharpness.

**Foam Fading Factor:** How long foam should float on water after being spawned.

## Water Spray

Produces spray when wave gets too big and breaks. Greatly improves storms.



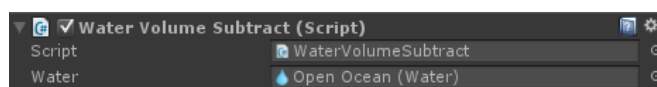
**Spray Material:** Material used by the spray particles. Default (and recommended) material is located at: “PlayWay Water/Samples/Materials/Spray.mat”.

**Spawn Boost:** Higher values produce more spray in more places.

**Spawn Skip Ratio:** Percentage of spray particles that will be randomly discarded from spawning.

## Water Volume Subtract

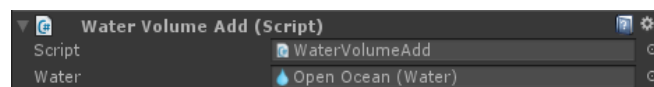
Removes water from the insides of attached colliders. Does that for both rendering and physics. Use it for boats, submarines, closed bottles etc. to remove not wanted water.



**Water:** Water which should be affected.

## Water Volume Add

Adds water to the insides of attached colliders. Does that only for physics. You should use custom meshes to do the same for rendering. Use it to create a glass of water, a puddle etc.



**Water:** Water which should be affected.

## Water Physics

Simulates drag and buoyancy forces applied by the fluids on the attached object.



**Sample Count:** How many sample points should be used to apply forces.

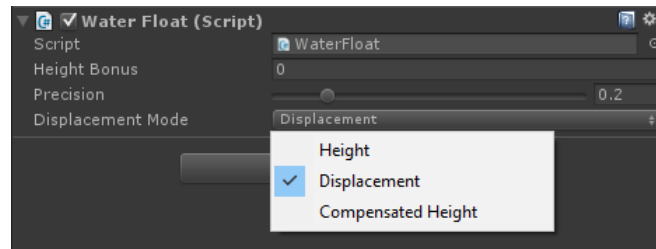
**Drag Coefficient:** Determines how much drag force should be applied to the object. For details, please refer to the Wikipedia page on the subject. It will be partially automated in the future.

**Precision:** Keep it low for objects bigger than most waves. Set it higher when imprecisions arise on small objects.

**Buoyancy Intensity:** Scale buoyancy force applied to the object. May be useful for beach balls and other objects with extremely low density as PhysX time step may be too high to handle them correctly.

## Water Float

Makes object float on water without use of physics.



**Height Bonus:** Vertical offset to the object position.

**Precision:** Keep it low for objects bigger than most waves. Set it higher when imprecisions arise on small objects.

**Displacement Mode:**

Displacement – Object will follow the original point on water in all directions. Recommended.

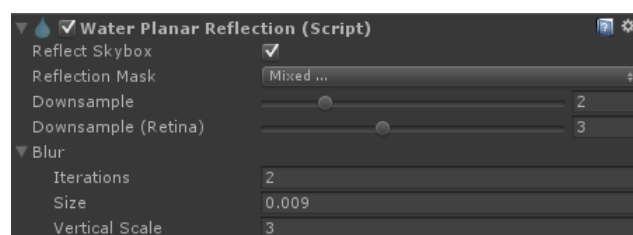
Height – Object will move only vertically. Horizontal displacements of water will be completely ignored.

Compensated Height – Same as height, but horizontal displacements will be compensated to produce more precise result.

## Planar Reflections

Generates planar reflections on the surface. Very precise, but works well only for relatively calm water.

You may tune its intensity in water profiles for each weather. This component is expected to be improved in the near future. New settings will appear.



**Reflection Mask:** Choose layers which should get reflected.

**Downsample:** Downsampling factor of the screen resolution. Planar reflections should be blurred anyway, so there is no reason to use the full resolution (1).

**Downsample (Retina):** Same setting, but used for screens with very high DPI.

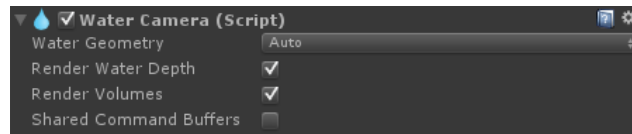
**Blur – Iterations:** Blur steps.

**Blur – Size:** Blur intensity.

**Blur – Vertical Scale:** Vertical blur multiplier.

## Water Camera

Renders water on camera along with all other screen-space data.



**Water Geometry:** Allows you to override water geometry used by the water to render. Useful for internal effects which, for example, won't work with projected grid.

**Render Water Depth:** Renders water depth into Unity's native depth map.

**Render Volumes:** Renders data used by subtractive volumes.

**Shared Command Buffers:** This component extensively uses command buffers. Check this option if you also use your own.