



Crest - Shallow Water API

Release 1.3.3

Wave Harmonic

Sep 04, 2025

TABLE OF CONTENTS

1 Crest.ShallowWater 1

1.1 ShallowWaterSimulation 1

1.2 ShallowWaterSimulationInjection 11

1.3 ShallowWaterSimulationMode 12

1.4 ShallowWaterSimulationPreset 12

CREST.SHALLOWWATER

1.1 ShallowWaterSimulation

Class in *WaveHarmonic.Crest.ShallowWater*, WaveHarmonic.Crest.ShallowWater

Runs a shallow water simulation at global sea level in a domain around its transform, and injects the results of the simulation into the water data.

1.1.1 Properties

AdditionalWater

Adds meters of additional water into the simulation domain on initialization.

Requires resetting the simulation.

Declaration

```
public float AdditionalWater { get; set; }
```

BlendDepthRange

Affects depth-based blending of the simulation and animated waves (m).

For the minimum, when the water depth is less than this value, animated waves will not contribute at all, water shape will come purely from this simulation.

Declaration

```
public Vector2 BlendDepthRange { get; set; }
```

BlendPushUpStrength

The intensity at which waves inject water into the simulation.

Declaration

```
public float BlendPushUpStrength { get; set; }
```

BlurMaskIterations

Filters the mask.

Declaration

```
public int BlurMaskIterations { get; set; }
```

BlurShapeForRender

Filters the shape prior to rendering to smooth out sharp features.

Always enabled when baking.

Declaration

```
public bool BlurShapeForRender { get; set; }
```

CourantNumber

Stability measure - limits velocities. Default 0.5.

Declaration

```
public float CourantNumber { get; set; }
```

CullDistance

Disable simulation if viewpoint is more than this distance outside simulation domain.

Declaration

```
public float CullDistance { get; set; }
```

CullTransitionSpeed

The speed of the transition to its culled state.

Declaration

```
public float CullTransitionSpeed { get; set; }
```

Depth

The depth of the water in the shallow water simulation (m).

Any underwater surfaces deeper than this depth will not influence the sim. Large values can lead to instabilities / jitter in the result. Requires resetting the simulation.

Declaration

```
public float Depth { get; set; }
```

DrainWaterAtBoundaries

Rate at which to remove water at the boundaries of the domain.

Useful for preventing buildup of water when simulating shoreline waves.

Declaration

```
public float DrainWaterAtBoundaries { get; set; }
```

DynamicSeabed

Recompute ground heights every frame.

Only enable this if terrain used by water system changes at runtime.

Declaration

```
public bool DynamicSeabed { get; set; }
```

EnableDistanceCulling

Disable simulation when viewpoint far from domain.

Declaration

```
public bool EnableDistanceCulling { get; set; }
```

Evaporation

Rate at which to remove water at any location.

Useful for removing remaining water from lowering tides.

Declaration

```
public float Evaporation { get; set; }
```

FlowStrength

Multiplies flow output to scale when injecting.

Declaration

```
public float FlowStrength { get; set; }
```

FlowTexture

The baked texture for flow.

Declaration

```
public Texture2D FlowTexture { get; set; }
```

FoamStrength

Multiplies foam output to scale when injecting.

Declaration

```
public float FoamStrength { get; set; }
```

FoamTexture

The baked texture for foam.

Declaration

```
public Texture2D FoamTexture { get; set; }
```

Friction

Friction applied to water to prevent dampen velocities.

Declaration

```
public float Friction { get; set; }
```

InjectFlow

Add the resulting flow velocities to the water system.

Declaration

```
public bool InjectFlow { get; set; }
```

InjectFoam

Add the resulting foam to the water system.

Declaration

```
public bool InjectFoam { get; set; }
```

InjectShape

Add the resulting shape to the water system.

If blending data from waves or height then use Waves or Level respectively. Requires resetting the simulation.

Declaration

```
public ShallowWaterSimulationInjection InjectShape { get; set; }
```

LevelTexture

The baked texture for water level.

Declaration

```
public Texture2D LevelTexture { get; set; }
```

MaximumDisplacementVertical

Inform water how much this input will displace the water surface vertically.

This is used for culling water tiles.

Declaration

```
public float MaximumDisplacementVertical { get; set; }
```

MaximumResolution

Maximum resolution of simulation grid.

Safety limit to avoid simulation using large amount of video memory. Requires resetting the simulation.

Declaration

```
public int MaximumResolution { get; set; }
```

Mode

Whether to use the baked textures.

Only supported by water level.

Declaration

```
public ShallowWaterSimulationMode Mode { get; set; }
```

OvershootReductionStrength

Overshoot is artifacts where the water will spike sharply.

Declaration

```
public float OvershootReductionStrength { get; set; }
```

PaddingWidth

Padding area for zero activity at the edge of the domain.

Adds padding so that simulation edge can overlap with height inputs without causing a seam (m). It is important that the padding fully overlaps the water level input, or there may be a gap. Visualized by the gizmo.

Declaration

```
public float PaddingWidth { get; set; }
```

PlaceAtSeaLevel

Places the simulation at sea level.

Requires resetting the simulation.

Declaration

```
public bool PlaceAtSeaLevel { get; set; }
```

Placement

Where the simulation is placed.

The default performs the best.

Declaration

```
public Placement Placement { get; set; }
```

Preset

Preconfigures the simulation for a specific use case.

Requires resetting the simulation.

Declaration

```
public ShallowWaterSimulationPreset Preset { get; set; }
```

SampleDepthProbeDirectly

Whether to sample the Depth Probe directly.

If disabled then data will come from the LODs which will have precision losses depending on camera position. Only disable if necessary. Requires resetting the simulation.

Declaration

```
public bool SampleDepthProbeDirectly { get; set; }
```

SampleWaterLevelInputsDirectly

Whether to sample water level inputs directly.

If disabled then data will come from the LODs which will have precision losses depending on camera position. Only disable if necessary. This option only supports water level inputs using the Geometry, Spline or Renderer modes. Requires resetting the simulation.

Declaration

```
public bool SampleWaterLevelInputsDirectly { get; set; }
```

TexelSize

Simulation resolution - width of simulation grid cell (m).

Smaller values will increase resolution but take more computation time and memory, and may lead to instabilities for small values. Requires resetting the simulation.

Declaration

```
public float TexelSize { get; set; }
```

TimeStep

Time step used for simulation (s).

Smaller values can make simulation more stable but requires more runtime computation.

Declaration

```
public float TimeStep { get; set; }
```

WaterEdgeMargin

Adds a margin around the water level to prevent water from degenerating at a distance and prevents water wall creep (px).

Declaration

```
public bool WaterEdgeMargin { get; set; }
```

WaterEdgeMarginBakedWidth

Same as Margin Width but for baking only.

Will use the larger value of the two.

Declaration

```
public int WaterEdgeMarginBakedWidth { get; set; }
```

WaterEdgeMarginWidth

Width of the margin.

The default is the optimal width.

Declaration

```
public int WaterEdgeMarginWidth { get; set; }
```

Width

The width of the simulation area (m).

Enable gizmos to see a wireframe outline of the domain. Requires resetting the simulation.

Declaration

```
public float Width { get; set; }
```

1.1.2 Methods

ResetSimulation

`ResetSimulation(bool)`

Resets the simulation. Needs to be called after certain property changes.

Declaration

```
public void ResetSimulation(bool populateDepthProbe = false)
```

Parameters

populateDepthProbe	Whether to also re-populate/re-bake the DepthProbe , if applicable.
--------------------	---

Bake

Bake()

Bake the simulation to a texture. Only works with ShallowWaterSimulationInjection.Level and in the Editor.

Declaration

```
public void Bake()
```

1.2 ShallowWaterSimulationInjection

Enum in *WaveHarmonic.Crest.ShallowWater*, WaveHarmonic.Crest.ShallowWater

Where to inject the displacement.

1.2.1 Properties

Waves

Inject into the Animated Waves.

Declaration

```
Waves = 1
```

Level

Inject into Water Level.

Declaration

```
Level = 2
```

1.3 ShallowWaterSimulationMode

Enum in *WaveHarmonic.Crest.ShallowWater*, *WaveHarmonic.Crest.ShallowWater*

ShallowWaterSimulation 's update mode.

1.3.1 Properties

RealTime

Simulate every frame.

Declaration

```
RealTime = 0
```

Baked

Simulation output has been baked to a texture.

Declaration

```
Baked = 1
```

1.4 ShallowWaterSimulationPreset

Enum in *WaveHarmonic.Crest.ShallowWater*, *WaveHarmonic.Crest.ShallowWater*

Preset options for the *ShallowWaterSimulation*

1.4.1 Properties

None

All options available.

Declaration

```
None = 0
```

Shoreline

Only shows options for shorelines.

Declaration

```
Shoreline = 1
```

Stream

Only shows options for streams.

Declaration

```
Stream = 2
```