# Index

# 3. Water Surface

# 1. Creating MultiWater Material

MultiWater  material can be created almost like any Unity™ standard material. Go to *Menu/Assets/Create/Suyu Assets/MultiWater Material* or right click mouse button in project window *Assets/Create/Suyu Assets/MultiWater Material* .

To change an existing material in MultiWater material go to material Inspector, in *Shader* field, select SuyuAssets/MultiWater as new shader.

# 2. Configuring MultiWater Material

## 2.1. Surface Options

The following properties of surface options tab are very important to MultiWater material work correctly, make sure that your MultiWater has the same values:

- *Surface Type* – Transparent.
- *Rendering Pass* – Default.
- *Blending Mode* – Alpha.
- *Back Then Front Rendering* – True.
- *Depth Test* – True.
- *Test Mode* – LessEqual.
- *Alpha Clipping* – True.
- *Double-Sided* – True.
- *Normal Mode* – None.
- *Double-Sided GI* – Auto.

All properties in Surface Options foldout are defined by Unity™, we only suggest some values that can represent the best results. For more information consult Unity™ documentation.

## 2.2. Water Editor

This area defines the type of surface of water and some technical aspects of the water.

### 2.2.1. Water Type

MultiWater has four types of surface:

*TIP: The words in parentheses are the "_KEYWORD" or "_PropertyName " strings used in scripts. For more information consult Unity™ documentation.*

- *None(_WATERTYPE_NONE)* – This is a non-activity water that is primary loaded before any type is selected.

- *Ocean(_WATERTYPE_OCEAN)* – This type is used for simulating ocean behavior, especially the seashore. Can be used also wherever environment requires the wave break phenomenon.
- *Course(_WATERTYPE_COURSE)* – This is the type to use wherever the water forms some kind of path like a river, stream or a waterfall.
- *Contained(_WATERTYPE_CONTAINED)* – This is the type to use when the water forms a closed system like lakes, lagoons and pools.

### 2.2.2. Additional Options

- *World Anchor(_WorldAnchor)* – Mark this option to work in World Space. So even when you move the plane, all features of water surface will still static on world space (very useful when you want an infinite plane behavior).
  - *Anchor Position(_AnchorPosition)* –This is override value to world origin (0,0,0). Once the anchor position is defined, all other values can be filled as working in Object Space.*(Only when World Anchor is ON.)*
- *Height Map(_HeightMap)* – This option will tint the water surface with a pre-determined color gradient height scale, what facilitates to visualize the behavior of water surface.
- *Edit Mode(_EditMode)* – This option puts your water in a static condition. Very useful to set texture values.
  - *Edit Time(_EditTime)* – When Edit Mode is on the time is fixed to zero. This time will override the time (very useful to sync animations).*(Only when Edit Mode is ON.)*
- *Show Seabed(_ShowSeabed)* – MultiWater material uses a 2D gray-scale texture to simulate the interaction of ground on the surface of the water. Mark these option to turn visible this image into the plane, and visualize the result on the water surface.

## 2.3. Surface Inputs

These settings determine the look and feel of the water surface itself.

### 2.3.1. Color Settings

- *Shore Color(_ShoreColor)* – The color of water in the area when intercept the firm land, and the next few meters.
- *Shallow Depth(_ShallowDepth)* – The depth of transition between shore color and shallow color.
- *Shallow Color(_ShallowColor)* – The color of water at where the water is classified as "shallow water".

- *Deep Depth(_DeepDepth)* – The depth of the transition between shallow color and deep color. In real world usually this happens at 200 meters depth.

- *Deep Color(_DeepColor)* – The color of the water at where the water is classified as "deep water". As we do with the definition of shallow water, let´s consider that deep water occurs at depth greater than 200 meters.

- *Brightness(_Brightness)* – Controls how much light the water surface will reflect. Looks like "Smoothness" property in Unity´s™ standard material.

- *Reflection(_Reflection)* – Controls how much of surroundings will be reflected on water surface.

### 2.3.2. Surface Waves *(Only Ocean Type.)*

- *Amount(_SWAmount)* – The quantity of undulations larger the ripples but not large enough to be considered as waves formations.

- *Height(_SWHeight)* – (0~1) The relative height of surface waves.

- *Disturbance(_Disturbance)* – (0~1) The relative random movement of these waves on the water surface.

- *Speed(_SWSpeed)* – The relative speed of the surface waves along the Z-axis.

### 2.3.3. Foam Settings

#### 2.3.3.1. Shore Foam

Refers to every foam formed near to the intersection of water with the land.

- *Foam Depth(_FoamDepth)* – The maximum depth where this foam is visible. This range starts at zero (the interception of water and land) and goes until this. Aside the color of foam can be defined. This color will be used in all foam types and occurrences.

#### 2.3.3.2. Surface Foam

- *Formation(_FoamFormation)* – (0~1) Controls the amount of foam that is randomly formed at water surface.

- *Persistence(_FoamPersistence)* – (0~1) Controls the time that this foam will be visible before it dissolves

## 2.4. Control Inputs

These settings control the behavior of the water and are different for each type of water.

### 2.4.1. World Settings

- *Day Length(_DayLength)* – Control in real world time(seconds) the length of a complete day(24hrs) on game world.
- *Tidal Band(_TideBand)* – Control the difference of water level between hide tide and low tide. The tide will change gradually every six hours (game time).
- *Shore Line(_ShoreLine)* – Control the position on Z axis (object space), from the object pivot, of a virtual line, that represent the average of intersection of water and land.
- *ShowLine(_ShowLine)* - When on, will tint all area after shore line by the color selected. *(only Ocean Type)*
- *Waves Type(_Swell)* – (0~1) Control which type of wave will be formed, that can be:
    - *Wind Waves* – Narrow waves that are generated in different positions along the Z axis.
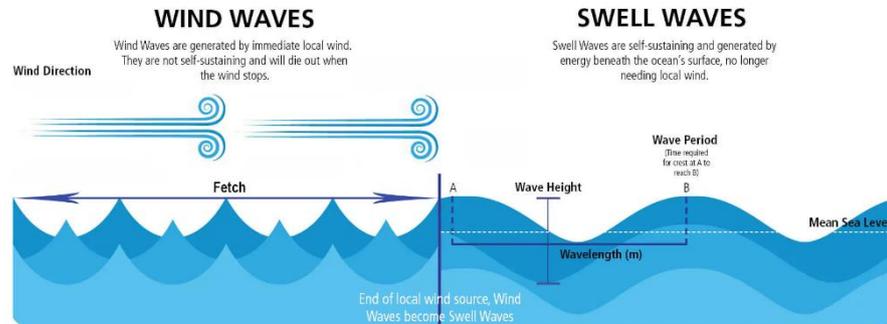    - *Swell Waves* – Very wide wave in most of cases well aligned.



*Fig. 1 Wave formation diagram*

- *Stormy(_Stormy)* – (0~1) Changes the appearance of surface, so it looks like in a storm weather.
- *Choppy(_Choppy)* – (0~1) Changes the behavior of water, so its look like more turbulent.
- *Water Speed(_WaterSpeed)* – Control the speed that the water flows through. *(Only Course and Contained Type)*

### 2.4.2. Water Settings

Specific controls that are loaded for water type Ocean. All values are in object space.

### 2.4.2.1. Ocean Settings

- *Wave Point(_WavePoint)*
  - *X. Breakpoint* – The position, on Z axis, from object pivot, where the waves breaks.
  - *Y. Wave Height* – The maximum height that the wave can reach, that is few meters before it breaks.
  - *Z. Left Edge* – The minimum value, on X axis, from object pivot, of the wave formation.
  - *W. Right Edge* – The maximum value, on X axis, from object pivot, of the wave formation.
- *Interval(_Interval)* – Control the amount of time (in seconds in real world time) between two consecutive waves. This value will determine the speed of all components of water.
- *Direction(_Direction)*
  - *X.* – *(*-1~1) Control the relative amount that the wave will move in the x-axis.
  - *Y.* – (0~1) Control the relative amount that the wave will move in the z-axis.
- *Segments(_Segments)* –  Define the number of divisions, on X-axis, of the wave formation. Each represent a wave phase(one peak and one valley).

MultiWater material allows up to three independent wave points, each one with own origin, interval and segments, so combining these points, exploring all variations possible, you can create a realistic beach environment.

*TIP: If you want an open sea environment, with no wave breaking effect, DON'T set the wave height to zero. This will make any behavior of the respective wave point as inexistent. To avoid the wave break effect, set the break point value with a position that exceeds the object max boundary at Z axis, that way, all wave break effects will occur outside of the visible range, and only the open sea parts will be visible.*

### 2.4.2.2. Course Settings

Course Water Type is based on the segmentation of the plane. Each segment is defined by Start Point and End Point. All controls affects only the segment area, and the area between each segment will follow the configuration of the previous End Point.

Important: Due to the complexity of vertex manipulation, sometimes the result on the plane object is far away from the expected. To solve this problem Course Water Type makes several calculations to fit one segment to another, so the values used to configure one segment are approximations of what will happen on the plane object, that is why minor adjustments on the segment configuration can result in no changes on the plane object. The most common cases are when the borders of one segment are close to the borders of the previous segment. In this case the space between the two segments will be eliminated and the result will be one segment with the two segments width together. There is a lot of adjustments that is done, and explain all very difficult, so just keep in mind little changes will be done on segment configuration, but they are very important to keep the result plausible with the reality.

- *Start Point(_StartPoint)*
  - *X. – Start on Z* – Control the segment start on Z axis, from object pivot, in object space.
  - *Y. – Start Height* – Start height equal zero mean that the current height at this point will be used. Any other value greater than zero will result in a height difference, that will be applied only a few meters before the start point.
  - *Z. – Left Edge* – Control the position of the left margin of water course.
  - *W. – Right Edge* - Control the position of the right margin of water course.
- *Restart(_Restart)* – Restart the plane at the end point of segment. It means that after end point in z-axis, all the plane will be at the same end height. If not, all parts out of edges will stay at start height.
- *Back Foam(_BackFoam)* – If segment width is smaller than the plane width, and the plane was restarted, a water flow interruption zone will be formed at the sides of segment. This option allow to control if an impact/reflux foam will be rendered or not. *(Only visible when Restart option is on.)*
- *Direction(_Direction)* - Control the direction of the water course. This is a vector2 property, where (0f,1f) represents straight forward, (1f,1f) represents 45º facing to right, and (1f,0f) represents 90º right direction. This direction only affects the space between start point and end point in Z-axis.

- *Foam Length(_FoamLength)* – (0~1) Control the length of the foam in the segment. It´s very useful when working with a waterfall case, that with this, you can control the foam along the fall.
- *End Point(_EndPoint)*
  - *X. – End on Z* – Control the segment end on Z axis, from object pivot, in object space.
  - *Y. – End Height* – End height is always a negative value and define the new zero value of next segment. There is nothing that prevents using positive values but  a lot of inconsistency will happens if you do that.
  - *Z. – Left Edge* – Control the position of the left margin of water course.
  - *W. – Right Edge* - Control the position of the right margin of water course.

The Course Water Type allows positioning up to five independent segments that works the same way explained before. For logical reasons, if the value of the left margin was equal to the value of the right margin, the segment will be ignored.

### 2.4.2.3. Contained Settings

- *Origin(_Origin)*- The origin position, in object space, of the disturbance that generates the undulations on water surface.
- *Wave Height(_WaveHeight)* – The height of main undulation on water surface.
- *Reflection Position(_ReflectionPos)* – The position where waves formed by the impact of the border will be placed. Only the position can be configured, all other values are consequences of the main disturbance.

The Contained Water Type allows the positioning up to eight reflection effects, that works at same way.

## 2.5. Control Inputs

### 2.5.1. Foam Textures

#### 2.5.1.1. Bubbles Texture

- *Albedo(_BubblesAlbedo)* – The albedo texture for bubbles.
- *Bump(_BubblesBump)* – The normal texture for bubbles.
- *Tilling*

- *X.* – The scale of texture on X axis.
- *Y.* – The scale of texture on Z axis.
- *Offset*
  - *X.* – The movement of texture on X axis.
  - *Y.* – The movement of texture on Z axis.
- *Normal(_BubblesNormal)* – The straight of normal map.
- *Speed(_BubblesSpeed)* – The relative speed that texture moves at given direction. Remember that this speed is inversely affected by the *Tilling* and directly affected by water speed .

**2.5.1.2. Blast Texture**

The same of *Bubbles Texture* but for the blast foam. Shader Properties are named as *_BlastAlbedo, _BlastBump, _BlastNormal and _BlastSpeed*.

**2.5.1.3. Flow Texture**

To reduce the tilling effect, flow foam is a combination of two sets of textures, each one with its own tilling and offset values that allows you to compensate some size differences, but the normal straight and speed value is shared by two textures, so we strongly recommend  to use textures with little differences of size. The shader properties are named as *_FlowAlbedo_1, _FlowBump_1, _FlowNormal, _FlowSpeed, _FlowAlbedo_2, and _FlowBump_2*.

## 2.5.2. Ripple Textures

To generate ripple effect we also use two textures, but these textures share the same values for *tilling, offset, normal strength and speed, so it´s very important use textures that has same pattern size, because there is no way to change any value individually.*

- *RippleMap_1(_RippleMap_1)* – The main ripple texture. This texture will control the values of ripple effect.
- *RippleMap_2(_RippleMap_2)* – The secondary ripple texture.
- *Normal(_RippleNormal)-* The normal strength to be used by the textures.
- *Speed(_RippleSpeed)* – The relative speed of ripples on water surface.

## 2.5.3. Seabed Settings

As said on Water Editor chapter, MultiWater material uses a 2D gray-scale texture to simulate an interactive surface. These settings allows you to change the texture, size, position and movement of this texture.

- *Seabed Map(_SeabedMap)* – The texture that will simulate the interactive surface. MultiWater uses 0~1 values to calculate all the behaviors, so as many "unregular" your texture was, choppier your water surface will be. Be warning that MultiWater is a vertex displacement based shader so there is a lots of limitations that must be considered.
- *Tilling and Offset* – Works the same way as other textures.
- *Static(_Static)* – Mark this option if you don´t want any movement in your seabed map. If you don´t mark this option, the seabed will move by default one time in horizontal direction by day length, even if you mark the seabed speed with zero.
- *Direction(_SeabedDirection)* – Control the direction of seabed movement.
- *Inverse(_Inverse)* – Will change the black areas into white and white into black. Keep in mind that whatever effect was, black areas means the minimum and white the maximum.

## 2.6. Advanced Options

For more information consult Unity™ documentation.

# 3. Water Surface

Water Surface is an assembly of all are needed to start using Multi Water as a ready to use asset. To create one go to *Menu/GameObject/Suyu Assets/Water surface* or right click mouse button in Hierarchy Window *Suyu Assets/Water surface*. A flat plane will be created using Unity™ default material, to avoid the creation of unnecessary materials. Assign a MultiWater material in Procedural Plane material field or in the Mesh Renderer material field and start configuring the material.

## 3.1. Procedural Plane

- *Water Material* – Assign the MultiWater material here. In the moment that the plane is generated, the Unity™ default material is used to avoid the creation of unnecessary material. At this point a little red square on the left side of Water Material field will be present, indicating that a not

compatible material is been used. Soon a MultiWater material be assigned, the square will turn to green.

- *Width* – The size of the plane along the X-axis. We use the Unity™ measure unit, so one unit represents the side of primitive cube.
- *Length* – The size of the plane along the Z-axis.

Procedural Plane was developed to work exclusively with MultiWater material. We try to make it compatible with others material as much we can, but some problems can occur using other materials.

## 3.2. Water Sound Player

- *Play In Edit Mode* – Make sound player reproduce the sounds even when in Unity™ Editor.
- *Water Sound* – The Audio Clip that each Audio Source will play.
- *Sounds Library* -A collection with the Audio Clip for each MultiWater water type. If no library is assigned, a default library will be loaded, and the audio clip witch type matches with material water type will be selected as Water Sound.
- *Audio Source Controls* – A collection of means controls of Unity™ Audio Source controls.
  - *Min Sound Distance* – The distance where the audio source volume stop increasing.
  - *Max Sound Distance* – The maximum distance that audio source still audible.
  - *Spatial* – (0~1) Control the spatial attenuation of a 3D sound. When the value is zero, the spatial attenuation will be ignored, and one, the spatial attenuation will be applied at all sound.
  - *Start Volume* – (0~1) The volume of Audio Clip that will be used by this player to do all calculations to final volume. Almost all calculations of volume are related to height variable.
  - *Start Time*-(0~1) The start point of Audio Clip to start playing the audio. Use this to synchronize the sounds with the water surface.

### 3.2.1. Water Sounds Library

This is a library to group the sounds for each type of water. Each audio clip will be associated to a water type, so when water type is changed the proper audio clip will be loaded.

A new Water Sound Library can be created at Menu/ Window/Suyu Assets/MultiWater/SoundsLibrary or mouse right click Create/Suyu Assets/SoundsLibrary.