Water 2D Tool v1.x

Table of Contents

1. Introduction

Water 2D Tool is a tool for Unity3D that allows you to quickly create 2D and 2.5D water with dynamic properties for your 2D or 2.5D games.

2. Creating a Water Object

To create a new water object you can access the menu (***GameObjects->2D Water*)**. Here you will find 2 option: the CPU based water and the GPU based water.

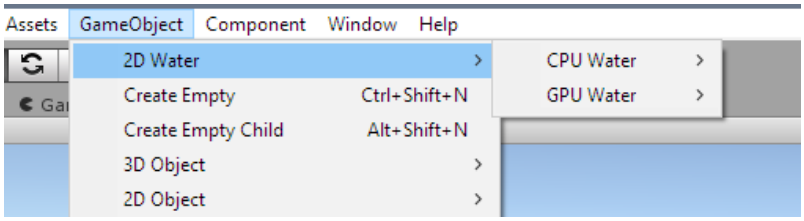

Figure 1

Alternatively you can right click in the Hierarchy window and you will find the 2D Water menu there too.

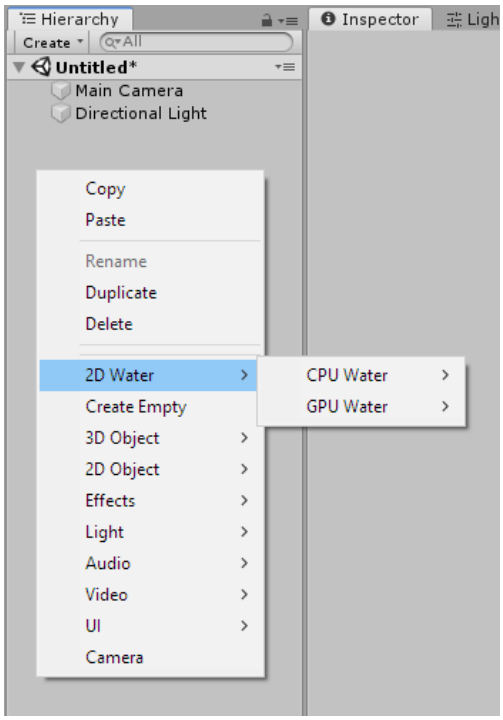

Figure 2

2.1. CPU Based Water

In the ***CPU Water*** menu you can choose to create a water object that has a 2D collider or a 3D collider, *Figure 3*.

Figure 3

After the water is created you can use the 4 handles to change its size and the position of its 4 edges, *Figure 4*.

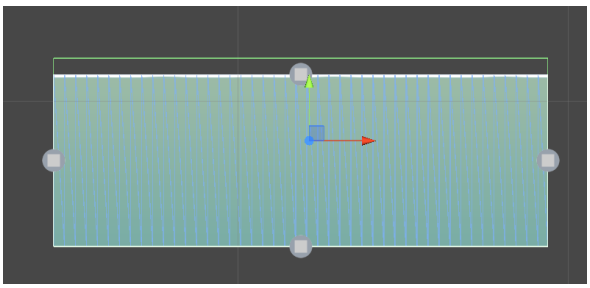

Figure 4

2.2. GPU Based Water

In GPU Water menu you will find 4 options, *Figure 5*:

- Creates a water object that has a 2D collider.
- Creates a water object that has a 3D collider.
- Creates and obstruction polygon. This polygon is used to create an obstruction texture.
- Creates a ripple source object. This objects can be placed on the surface of the water to generate ripples.

For more information about the GPU based water please go to section 5 of this guide.

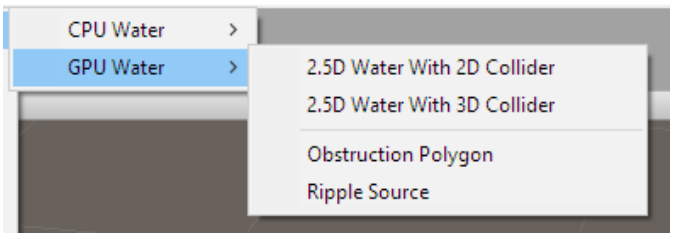

Figure 5

3. Rules

Below are a few rules that should be followed to make sure Water 2D Tool works correctly.

- The scale of the water should always be (1, 1, 1). Use the water handles to change the size of the water.
- Do not place a water object as a child of an object that doesn't have the scale (1, 1, 1) as this will also change the scale of the water object.
- The water objects should not be rotated.
- The Player must have the tag "Player" for the Water 2D Tool player settings to work.
- Only one object with the tag "Player" should enter the water.

4. Mobile Optimization

Water 2D Tool can be used not only on PCs and consoles, but on mobile phones too. If you are developing for low end or old phone models, you may experience some performance drops with the default water settings. Below are some tips on which settings you should change in order to improve your game performance.

- In the **Mesh Properties** group of options, reduce the value of the field ***Segment To Units***. This will reduce the number of vertices the mesh has.
- Change the Buoyancy from **Physics** based to **Linear**.
- In the **Miscellaneous** group of options, **Collider Detection** drop down menu select **Bounds Based** option.
- If possible do not animate the left or right sides of the water because Water 2D Tool will recreate the water mesh every time the water size changes. The animation of the top and bottom edges of the water doesn't recreate the mesh.

The above list applies to the CPU based water as well as the GPU based.

If the water object you created is GPU based, there are a few additional settings that you can change to improve the performance.

- In the **2.5D Water** options group enable the toggle **Z Segments Limit.**
- Reduce the value of the field **RT Pixels To Units** to 10 or 8.
- Use the basic version of the GPU based water shader.

5. GPU Based Water Guide

All the properties of the GPU based water can be customized in the water objects Inspector, *Water2D_Ripple* script, Figure 6.

Figure 6 – GPU based water settings

As you can see from Figure 6, the fields **Radius** and **Strength** can be encountered multiple times.

- **Radius** – This field represents the initial radius of the ripple, or you could look at it as the radius of the object that creates that ripple

- **Strength** - Strength determines how high the ripple waves rise or go down.

To find more about the function of a particular field, you can hover the mouse over its name and a short info message will be displayed to you.

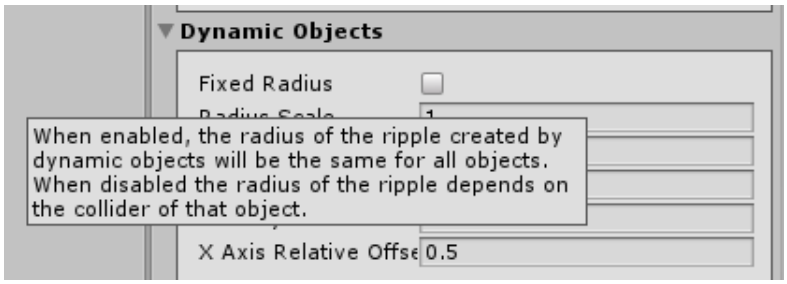

Figure 7

.

5.1. Water Interactions

Dynamic Objects Interaction - A dynamic object that interacts with the surface of the water will generate ripples. The properties of the ripple can be customized in the *Dynamic Objects* options group, Figure 8.

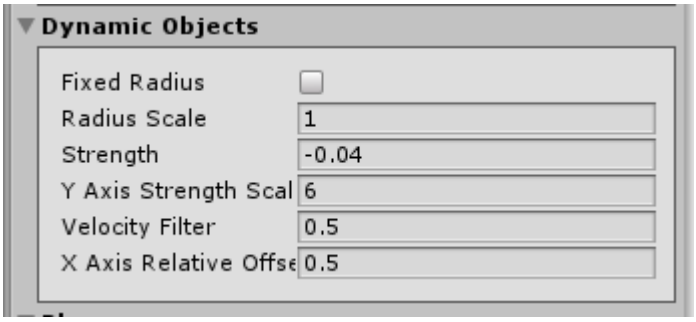

Figure 8

There a several conditions a dynamic object must satisfy in order to be able to generate a ripple.

- The collider of the object must intersect the surface of the water. Objects that are above or below the surface of the water will not generate ripples.
- The Abs values of the velocity on the X or Y axis must be greater than the value of the field *Velocity Filter*.

When **Fixed Radius** toggle is disabled, the radius of the ripple created by a dynamic object depends on that objects collider bounding box length on the Z axis and the value of the field **Radius Scale.** When Fixed Radius is disabled, the ripple radius is the same for all dynamic objects. If the water object uses a 2D Collider, **Fixed Radius** is enabled by default and can't be disabled.

Player Interaction – You can use a physics based character controller or a ray cast based character controller for the player. While a player that uses a ray cast based character controller will generate ripples, the buoyancy will work only with physics based character controllers. The properties of the ripples created by the player can be customized from the *Player* options group, in the *Water2D_Ripple* script, Figure 9.

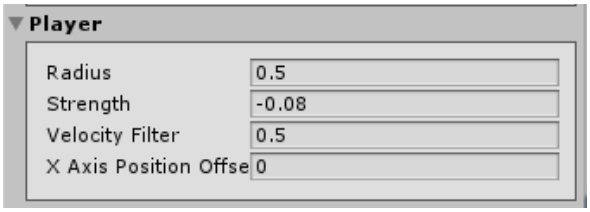

Figure 9 – Player options for the GPU based water

The conditions a dynamic object must satisfy in order to generate a ripple also apply to players that use a physics based character controller. In the case of the players that use a ray cast based character controller, only the first condition must be satisfied.

Ripple Sources Interaction - To create a ripple source, you can access the menu *GameObject->2D Water->GPU Water->Ripple Source* or instantiate a ripple source prefab from *Water2D_Tool/Assets/Prefabs* folder. There are 2 ways you can use a ripple source. You can place the ripple source object in the water objects inspector, *Ripple Sources* options group, Figure 10.

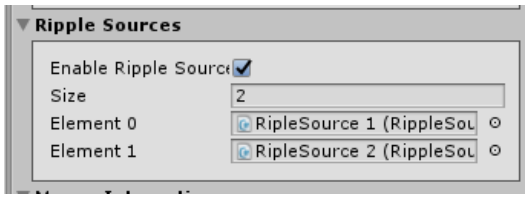

Figure 10

Alternatively you can add the **RippleSource** script to a kinematic object. It is not recommended to add the **RippleSource** script to a dynamic object.

A ripple source has 3 main options that determine when a ripple is generated:

- **When Moving** – When this option is enabled, a ripple will be generated only when the object changes its position in world space.

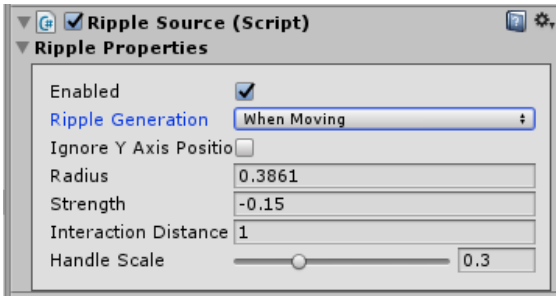

Figure 11

- **Random interval** – When this option is enabled, a ripple will be generated at random intervals of time. After a ripple is generated, the ripple source will calculate a random value between the values of *min* and *max Delta Time*. This value represents the time to wait before a new ripple is generated.

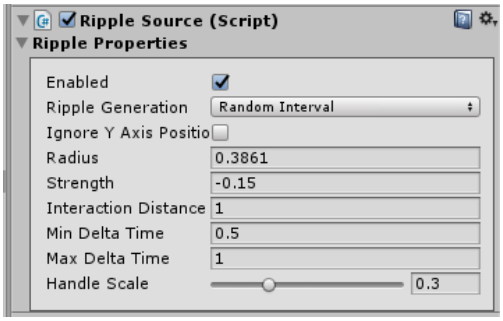

Figure 12

- **Fixed Interval** – When this option is enabled, a ripple is generated at a fixed interval of time.

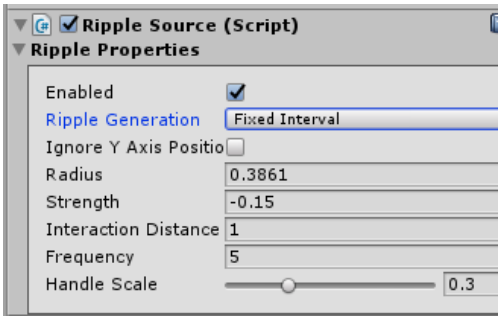

Figure 13

When you select a ripple source in the editor, you will see a couple of handles. The Green square handles can be used to change the value of the field **Interaction Distance.** The current value of the field **Interaction Distance** is represented by the green line. For a ripple source to be able to generate ripples, the green line must intersect the surface of the water. This rule is ignored if the toggle **Ignore Y Axis Position** is enabled.

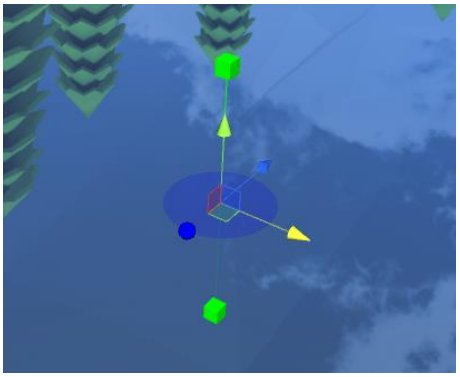

Figure 14

The blue sphere handle can be used to change the value of the field **Radius**.

Mouse Interaction – you can use the mouse arrow to generate ripples on the surface of the water. The radius and power of the ripples can be set in the *Mouse Interaction* options group, Figure 15. This option is only available if the water object uses a 3D collider.

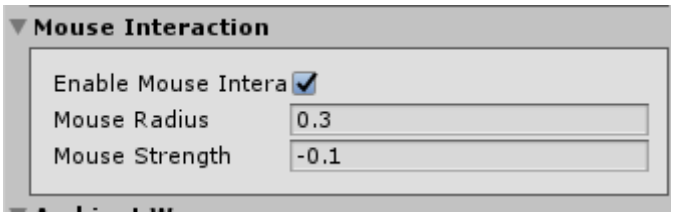

Figure 15

NOTE: Currently only 10 ripples can be generated per frame.

5.2. Water Obstructions

The GPU based water supports 2 types of obstructions: dynamic and texture.

Dynamic Obstructions – To enable this option, in the **Obstruction Type** dropdown menu select **Dynamic Obstruction**.

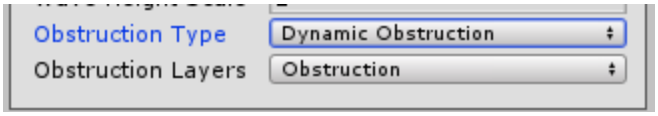

Figure 16

Objects that have a Box Collider or a Sphere Collider and are on the Obstruction Layers mask, will prevent the ripples from passing through them. These objects can be moved in the water and the obstruction region will move with them. Currently there are some restrictions to this feature. You can use only 5 objects with Sphere colliders and 5 with Box colliders. If there are more than 5 of each type,

they will be ignored. Another restriction is that the objects with box colliders should be rotated only around the Y axis and only by 0, 90, 180, 270 or 360 degrees. This is because the bounding box of the collider is used to calculate the obstruction region.

Texture Obstructions - To enable this option, in the **Obstruction Type** dropdown menu select **Texture Obstruction**.

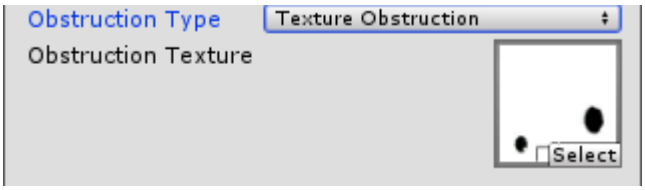

Figure 17

In the obstruction texture white means the ripples can move freely while black means that they can't pass through that region and will be reflected back.

You could use photoshop or other tools to create the obstruction texture, but there is a much simpler and quicker way to do it. Water 2D Tool has a build in tool that allows you to create an obstruction texture right inside Unity.

To create a texture obstruction, first go to the menu (*GameObject->2D Water->GPU Water)* and create an **Obstruction Polygon** object, Figure 18**.**

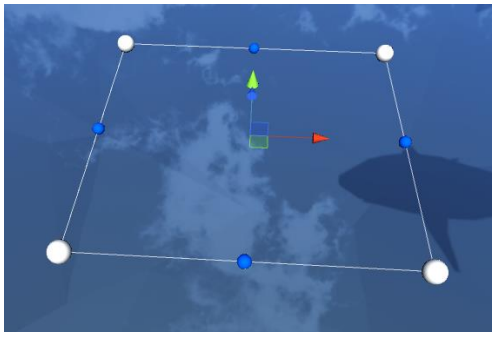

Figure 18

To change the shape of the obstruction polygon click, hold and drag one of the white handles. Clicking on the one of the blue sphere will add a new polygon point at that position. To delete a polygon point hold ALT key and press on one of the red spheres, Figure 19.

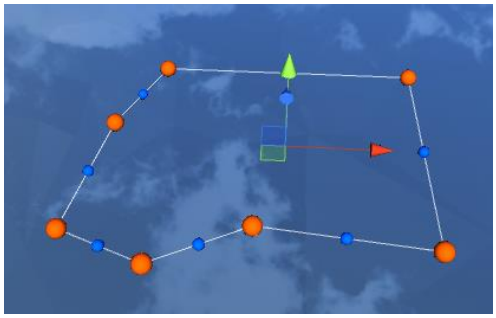

Figure 19

When editing the shape of the polygon, to have a better view, it is recommended that you change the camera view to Top. You can do that by right clicking on the axis gizmo and selecting Top option.

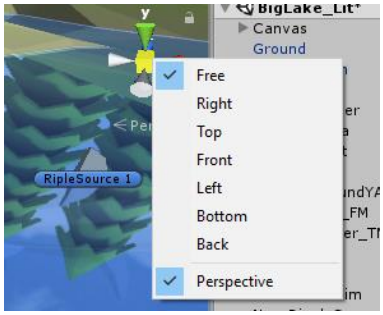

Figure 20

After you give the obstruction polygon the shape you need, go to Windows menu and click on **Texture Obstruction Creator**.

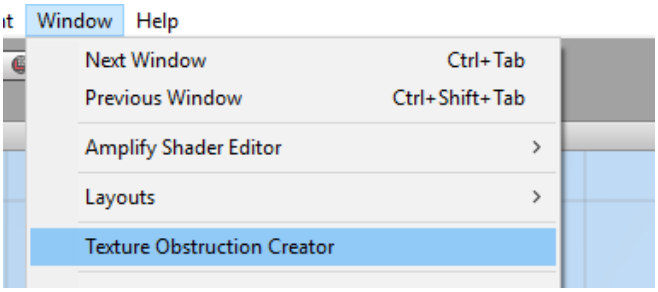

Figure 21

This will bring up the **Texture Obstruction** window Figure 22. Here you can give the texture a name, set its scale. By default the texture scale is set to 1, this means that the obstruction texture will have the same resolution as the render texture that is used to store the height map.

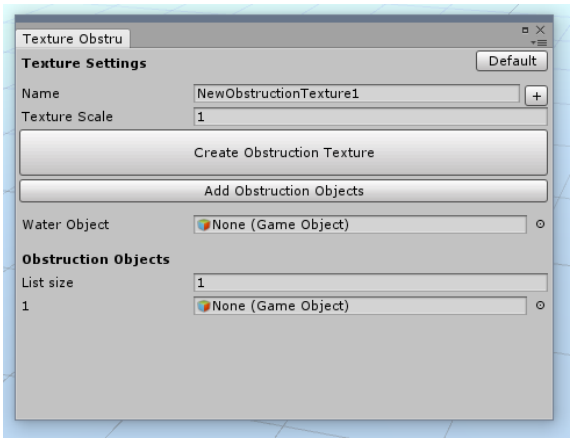

Figure 22

The next step is to place the water object for which we want to create an obstruction texture in the **Water Object.**

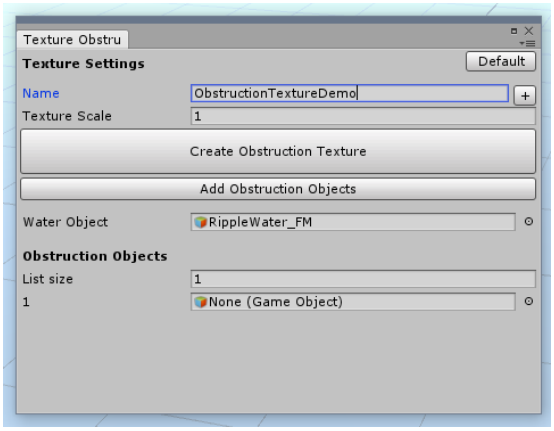

Figure 23

We do the same for the obstruction objects. We could place them one by one manually or we can just select the obstruction objects in the hierarchy panel and just click the button **Add Obstruction Objects.**

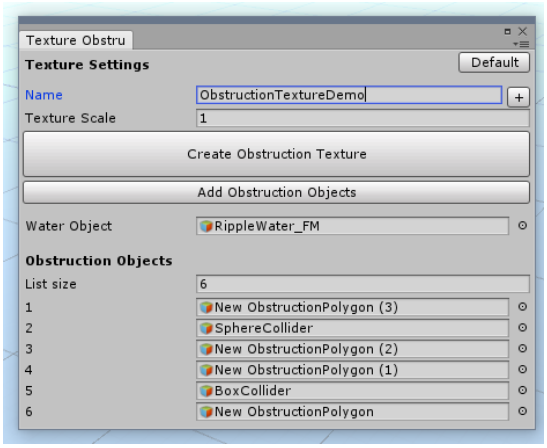

Figure 24

You can also use as obstructions, objects with Sphere and Box Colliders. The objects that have box colliders can be rotated only around the Y axis.

Now all that's left is to Press the button **Create Obstruction Texture**. The new obstruction texture can be found in the *Water2D_Tool/Assets/Textures/Obstructions* folder.

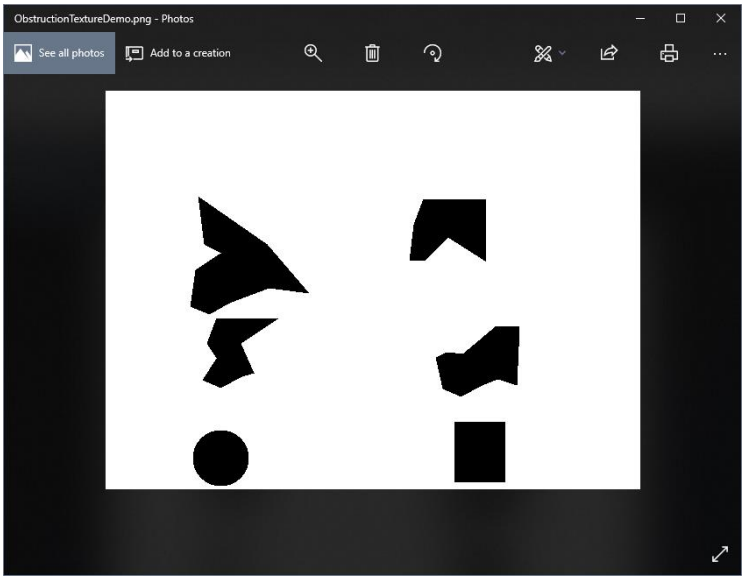

Figure 25

6. Water Settings and Concepts

For the CPU based water Water 2D Tool creates the waves, by modeling the surface of the water as a series of vertical springs, Figure 26. Each surface vertex has a spring attached to it that controls its position.

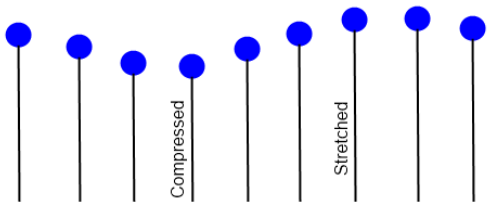

Figure 26

Figure 27 shows how the mesh vertex indices are arranged. For simplicity the first half of the water mesh vertices form the bottom of the water mesh while the second half form the top of the water mesh.

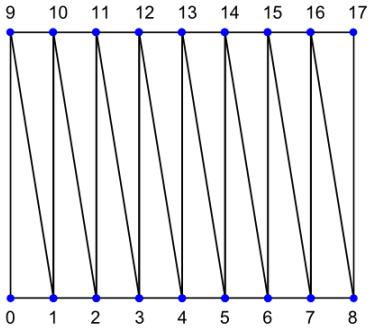

Figure 27

Segments to Units - The density of the mesh vertices on the X axis can be controlled using the value of the field ***Segments to Units*** from the ***Visual Properties*** group. The value of 3 means that in 1 m of Unity space on the X axis, 3 columns (or horizontal segments) will be placed. The triangles marked in green form a column, Figure 28**.**

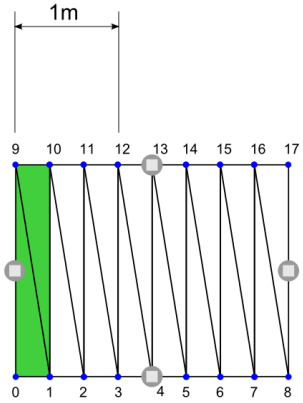

Figure 28

6.1. 2.5D Water

When this toggle is enabled, a new game object is created as a child of the current water object. This object has a mesh on the X and Z axis, Figure 29**.**

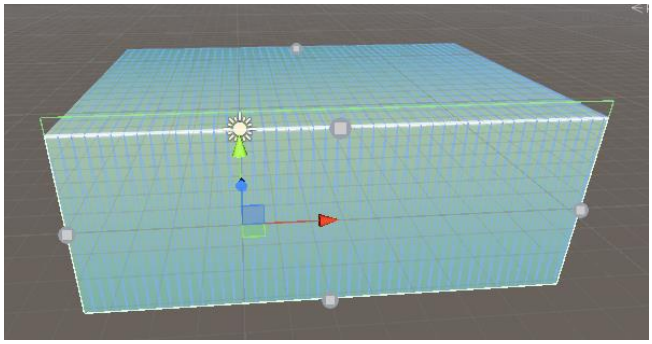

Figure 29

6.2. Buoyancy

Polygon Clipping - This dropdown menu contains 2 options that determine which method will be used to calculate the intersection between the object collider and water collider**.**

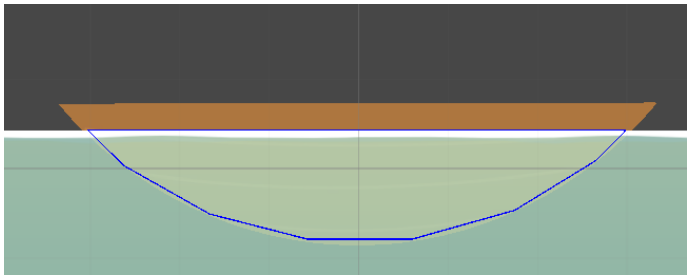

Figure 30 - The intersection polygon, simple clipping

The intersection polygon is represented by the blue lines.

The process of calculating the intersection between 2 polygons is usually called polygon clipping, where one polygon is the subject polygon and the other is the clipping polygon. In our case the collider of the object is the subject polygon.

Simple option – The *simple* clipping option uses the Sutherland Hodgman polygon clipping algorithm and the clipping polygon is always a horizontal line made of 2 points. The position on the X axis of this 2 points corresponds to the global positions of the water edges on the X axis, while the position on the Y axis is equal the global position of the vertex that is closest to the center of the object.

The *simple clipping* is the cheapest option in terms of performance, but it has a disadvantage. Because the clipping polygon is always a horizontal line, the objects behavior on big waves is not very real, Figure 31**.**

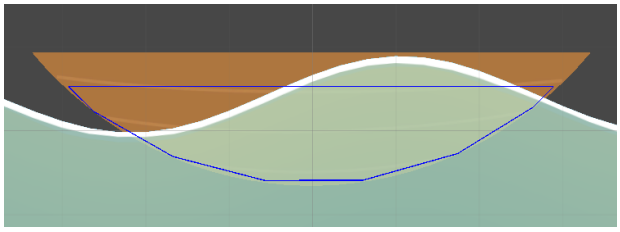

Figure 31 - The intersection polygon, simple clipping

This problem is not present in the complex option.

Complex option – The complex clipping option uses the Clipper library. Here the clipping polygon has at list 4 edges. The clipping polygon can be seen in Figure 32, represented by the green lines.

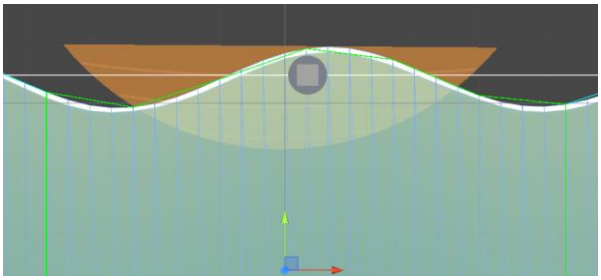

Figure 32 - Complex polygon clipping

Water Line Segments - To get a clipping polygon that has a shape that is close to that of the water wave, but still get a good performance, only the positions of certain vertices are used.

Which vertices are used is determined by the value of the field ***Water Line Segments*** from the ***Buoyancy*** group in the inspector**.**

As an example let's say the water line left most vertex has the index 9 and the value of the field ***Water Line Segments*** is 4, Figure 33**.** In this case the positions of the vertices with the indices 9, 13 and 17 will be used to build the clipping polygon. As you can see the value of the field ***Water Line Segments*** is equal to the number of horizontal segments between the vertices that are used.

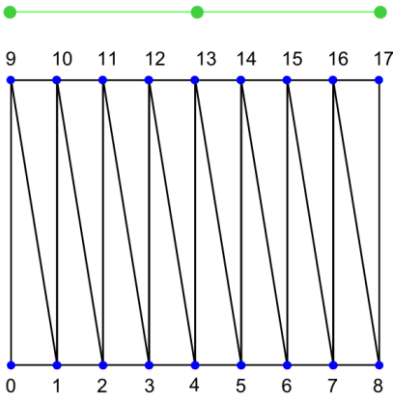

Figure 33 – The water mesh

Polygon Vertices **-** For objects that have a CircleCollider2D or a SphereCollider, an imaginary regular polygon is created based on the radius of the CircleCollider2D or SphereCollider. This is done so that a drag and a lift can be applied to this object, Figure 34**.** The value of this field determines the number of corners this polygon will have.

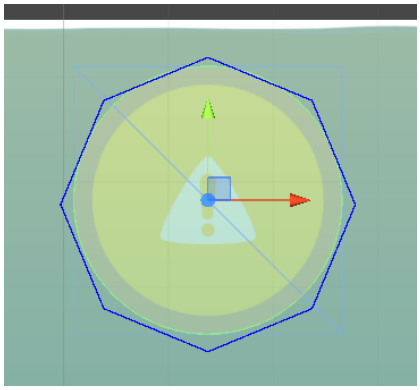

Figure 34 – An imaginary regular polygon for an object with a CircleCollider2D

Float Height determines how much force should be applied to an object submerged in the water. A value of 3 means that 3 m under the water the force applied to an object will be 2 times greater than the force applied at the surface of the water, Figure 35**.**

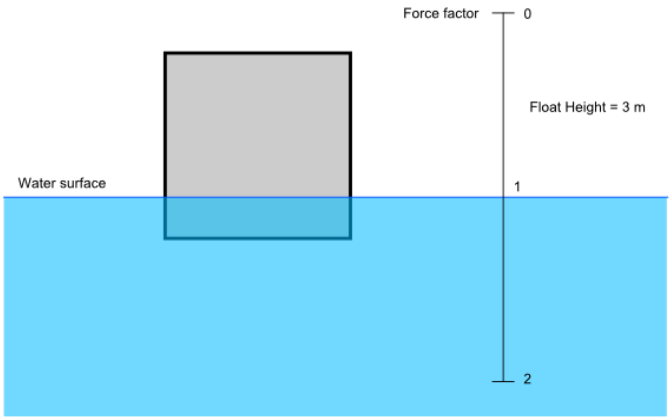

Figure 35 – Linear buoyancy, Float Height

6.3. Surface Waves

This are the waves that are not created by objects.

Random Splashes **-** When this option is selected, the velocity of random surface vertices will be changed based on a random velocity value generated between the values of ***Max Velocity*** and ***Min Velocity***. The velocity of its left and right neighbors are changed too.

This option can be used to simulate water rain.

Sine Waves **-** When ***Sine Waves*** option is selected in the ***Surface Waves*** drop down menu, the velocity of the surface vertices will be changed using a sine function.

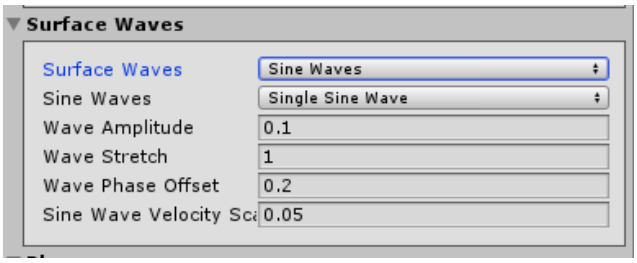

Figure 36 – Surface Waves group

You can use a single sine wave to change the velocity of the surface vertices by selecting ***Single Sine Wave*** from the ***Sine Waves*** drop down menu Figure 36, or multiple waves by selecting the option ***Multiple Sine Waves,*** Figure 37**.**

When Single Sine Wave option is selected you can change dynamically the ***Wave Amplitude***, ***Wave Stretch*** and ***Wave Phase Offset*** of the sine wave by changing the values of the public variables: waveAmplitude**,** waveStretch and wavePhaseOffset, using the animator or a scrip.

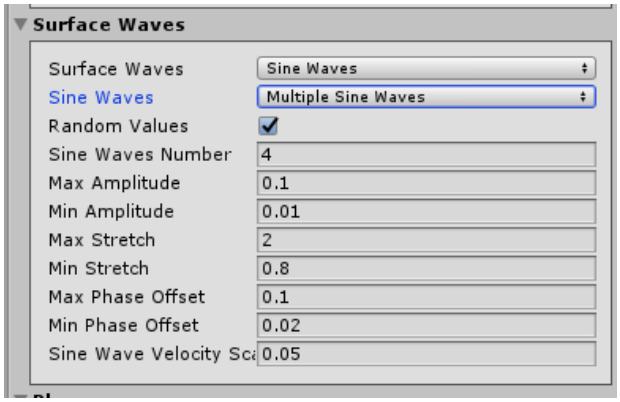

Figure 37 - Surface Waves group

Random Values **-** When this toggle is enabled, the amplitude, stretch and phase offset will be random values generated in the start method, for each sine wave.

You can disable this option and use your own values, Figure 38**.**

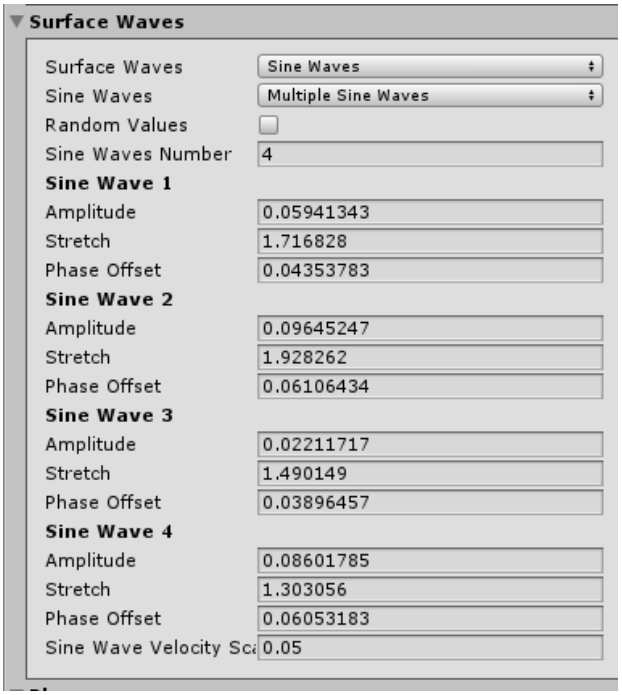

Figure 38 - Surface Waves group

6.4. Miscellaneous

Interaction Region **-** This is the bottom region of a colliders bounding box that can affect the velocity of a water vertex. This value is used to limit the ability of the objects with big bounding boxes to affect the velocity of the surface vertices. Only the velocity of the vertices that are inside the interaction region will be affected by an object, Figure 39**.**

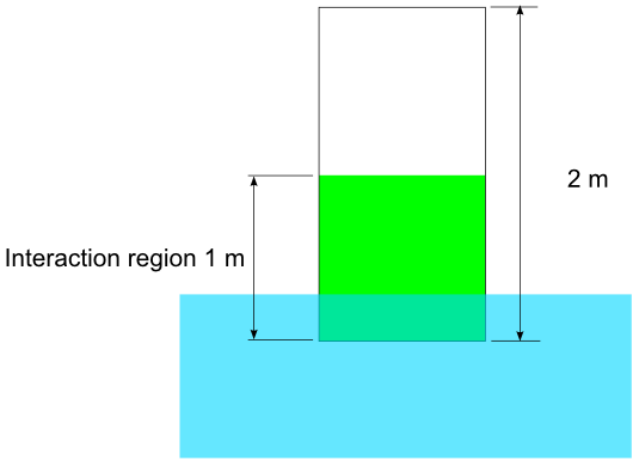

Figure 39 – Interaction Region

7. Customer Support Links

If you have any questions or difficulties with Water 2D Tool you can make a post on the forum thread or email me directly on my email.

Forum thread - https://bit.ly/2TEySHG

Email – johnq002@gmail.com