WATER OBJECT MANUAL

CONTENTS

Tip: Hover your mouse over editor fields to see an explanation.
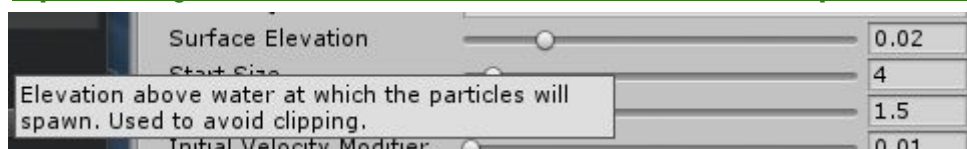
Quick Start

Making An Object Float

For demonstration purposes a primitive sphere will be used (GameObject > 3D Object > Capsule).

- Add *WaterObject* component to you object.

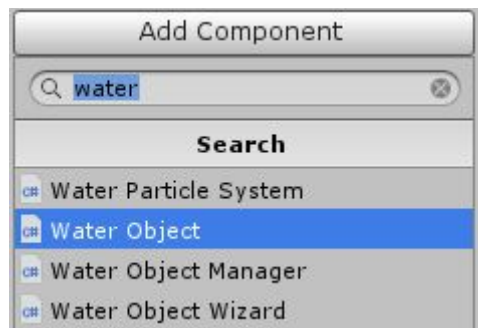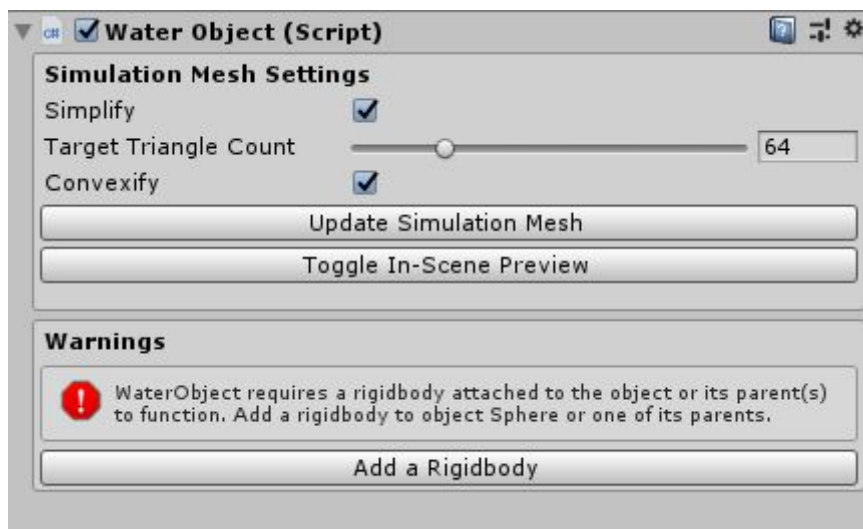- In case your object does not have a rigidbody, or there are some other setup issues, *WaterObject* will warn you. Follow the warnings. In this case our capsule does not have a rigidbody which is required for *WaterObject* to function. Click on 'Add a Rigidbody' or add it manually.

- By default *Rigidbody* has *Mass* of 1 which is way too light for the capsule of this size. Set the *Rigidbody Mass* to 300.

- Alternatively, you could add component *RigidbodyMassFromChildren* to your rigidbody which will enable Material Settings section on the *WaterObject*. Here you can choose a *Material Preset* or manually set either *Density* or *Mass* (*Volume* is auto calculated from the mesh). If one *Ribidbody* has multiple *WaterObject* children, the mass of that *Rigidody* will be a sum of all the child *WaterObject* masses.

Material Settings	
Material Preset	Custom
Density	400
Mass	419.5175
Volume	1.048794

- Add component *Water Object Manager* to any object in the scene. Exactly one *Water Object Manager* needs to be present in the scene since it is responsible for all the physics calculations.

- Press Play.

- If you want the capsule to stand upright and not fall on its side, you can attach *Center Of Mass* component to its *Rigidbody*. Set *Center Of Mass Offset* to (0, -0.5, 0) and see what that does. Green sphere indicates current center of mass.

Adding Flat Water To The Scene

- The capsule now floats but it looks as if it is floating in thin air. To remedy this a water renderer needs to be added to the scene. This can be the included default water (DWP2 > DWP2 > Prefabs > Water), AQUAS or any flat water asset.

- Drag the DefaultWater prefab (or any other prefab you might have) into the scene and tag it with 'Water' tag. The objects will now float at the height the water object is at (water *Transform*'s y-position).
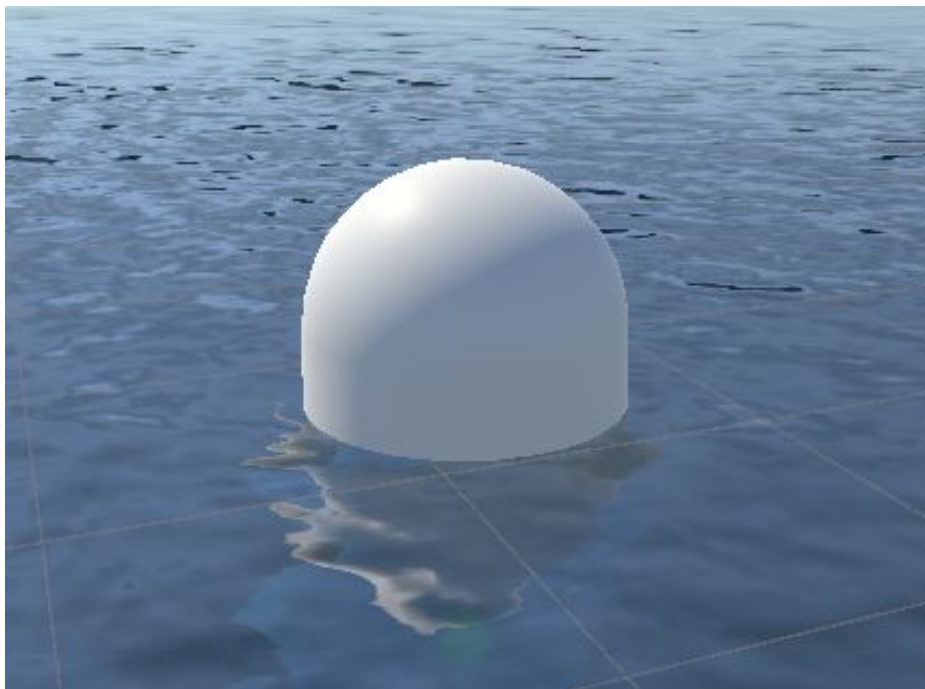
Adding Wavy Water To The Scene

NOTE: When using DWP2 with any non-flat water asset expect some overhead. If you need to simulate a large number of triangles, untick *High Resolution Water Queries* option on *WaterObjectManager*. This will reduce resolution somewhat but increase performance.

- DWP2 also supports water with waves. Currently supported are Crest, Ocean Community Next Gen and Lux Water. To use them with DWP2 set them up as you would flat water (previous section) and then under Edit > Project Settings > Player > Scripting Define Symbols add <u>one </u>of the following:
  - *DWP_CREST* for Crest
  - *DWP_OCEAN_NEXT_GEN* for Ocean Community Next Gen
  - *DWP_LUX* for Lux Water
  - *DWP_SUIMONO* for SUIMONO
  - *DWP_CETO* for Ceto Ocean

- Make sure that the object containing the relevant script (*Ocean Renderer* for Crest, *Ocean* for Ocean Community Next Gen) is tagged 'Water'.

- If the wavy water asset is not supported it is possible to add support for it by extending WaterDataProvider class IF the asset in question can provide water height at the given point. Alternatively, you could send an email to nwhcoding@gmail.com with the request.

Adding Water Particle System

Water Particle System is a 2D plane particle system and will not work with wavy water. It can be used with Crest / Ocean Community Nex Gen / etc. as long as wave magnitude is set to 0.

DWP2 water effects use standard Unity *Particle System* combined with *Water Particle System* script that takes the simulation data from *Water Object Manager* and determines when, where and how to emit particles.

- To add water effects to an object add *Water Particle System* to it. *Water Particle System* will automatically add a *Particle System* to the object with defaults already set. If you want to change the defaults edit *WaterParticleSystemPrefab* (*DWP2 > DWP2 > Resources > WaterParticleSystemPrefab.prefab)*

- Enter play mode and try dropping the object from e.g. 10 meters. You will see foam forming around the object. Dropping from larger height will generate more / more intense foam.
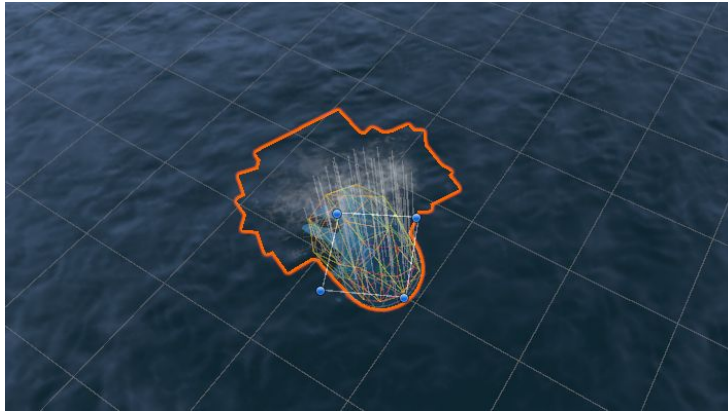
A Quicker Quick Start

There is a quicker way to setup a *Water Object* than the one above. It is just a good idea to know how to set it up manually first.

Steps:
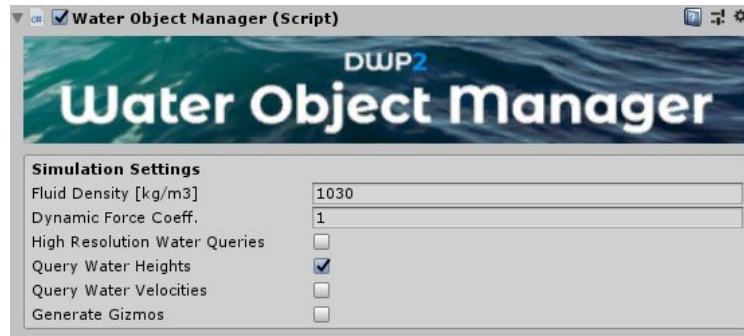- Attach *Water Object Wizard* to the object(s) you want to set up.

- Tick 'Add Water Particle System' if you want to add *Water Particle System*.
- Press Auto-Setup. It floats!

(if it does not contact the developer at nwhcoding@gmail.com)

COMPONENTS

Water Object Manager

Water Object Manager is the main script of Dynamic Water Physics 2 and it needs to be added to every scene that will have water physics enabled. All the physics calculations for all the Water Objects in the scene are done in here. This is different to DWP1 where each object did its own calculations. Centralising all the calculations has been done to make use of Unity Jobs (multithreading) and this is actually the main reason for existence of DWP2.

You can attach Water Object Manager to any object in the scene, as long as exactly one is present.

Fields
- *Fluid Density* - Density of the fluid in kg/m3. Default is 1030 which equals salt water.
- *Dynamic Force Coefficient* - Roughly equal to the viscosity of the fluid. 1 is default value for water. All forces except for buoyant forces are multiplied by this value.
- *High Resolution Water Queries* - Only effective is Crest, Ocean Community Next Gen, Lux Water or some other wavy water asset is used. Instead of querying for water height at the center of each triangle, water heights will be queried for each point of each triangle tripling the number of queries. Enabling this option is recommended for when higher accuracy is required (e.g. ship simulator).
- *Query Water Heights* - Only effective when wavy water asset is used. If disabled wave heights will not be queried and instead script will assume flat water. It is useful to toggle this option if you are using wavy water asset but with wave amplitude set to 0 (e.g. Crest but without waves) as it improves performance.
- *Query Water Velocities* - Should water surface velocity be queried (rivers, streams)? At the moment there are no supported water assets that support this feature but the option has been added as some assets have announced this feature. If you do not need detailed water velocity simulation simply using Rigidbody.AddForceAtPosition() function to imitate water flow will be adequate.
- *Generate Gizmos* - If enabled and gizmos are turned on in the editor Gizmos for all water objects will be generated and drawn.

Water Object
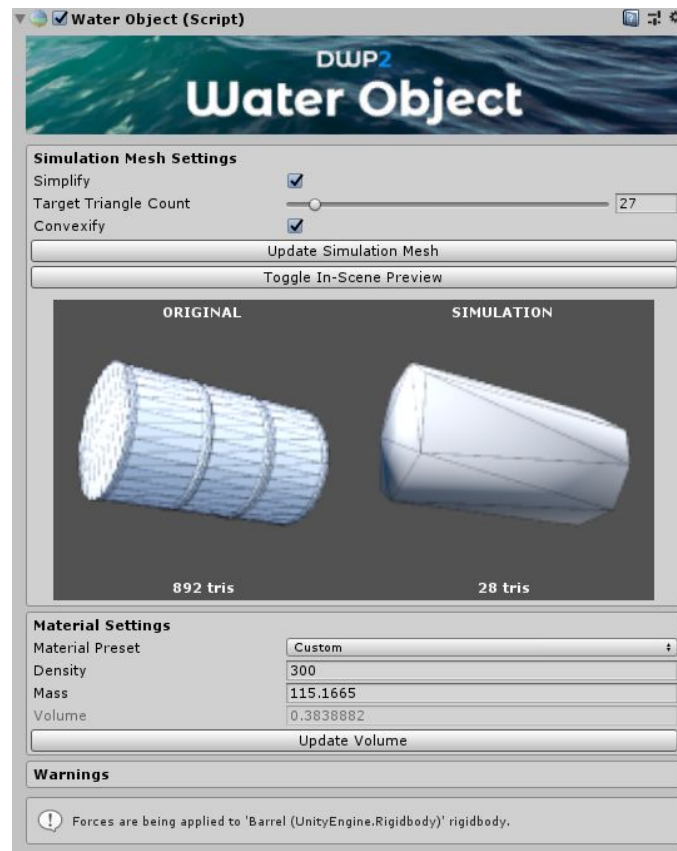
Water Object is a script that sets up the object for simulation. All the objects containing *Water Object* will interact with water if there are no setup errors and *Water Object Manager* is present. *Water Object* requires the object to have a *Mesh Filter* present (mesh renderer is not needed). *Water Object* uses a *Simulation Mesh* that is an (optionally) simplified mesh.

Fields

Simulation Mesh Settings:

- *Simplify* - Should the simulation mesh be simplified? If the mesh has more than 30 triangles it is usually a good idea to enable this option.
- *Target Triangle Count* - How many triangles the simplified simulation mesh will have?
- *Convexify* - Should the simulation mesh be made convex? While DWP supports both convex and concave shapes, enabling this option is recommended when there are holes in the mesh (e.g. a crate without the top or a ship hull without deck). In some cases it can also be used to lower the triangle count additionally.
- *Update Simulation Mesh* - Generates the simulation mesh from the given options. If *Simplify* and *Convexify* are left unticked the original mesh will be duplicated.
- *Toggle In-Scene Preview* - Besides using the preview icons inside the inspector it is also possible to preview the simulation mesh inside the scene. When clicked this button will toggle between original and simulation mesh on the current object.

Material Settings (only available if Rigidbody has *Rigidbody Mass From Children* component attached):

- *Material Preset* - A list of common material densities. When selected *Mass* will be calculated from the auto-calculated volume of the mesh (scale-sensitive) and the selected material density. Set this field to *Custom* if you want to use a custom density or mass.

- *Density* - Only available if *Material Preset* is set to custom. Density in kg/m3. Setting this field will update the *Mass* field.
- *Mass* - Only available if *Material Preset* is set to custom. Mass in kg. Setting this field will update the *Density* field.
- *Volume* - Auto-calculated volume of the mesh. If you scale the mesh or otherwise change it click *Update Volume* to make sure it is up-to-date.

Instantiating at Run-time

Note that when *Water Object* is added to the scene during play mode it will not be automatically registered because this requires re-allocating memory for the jobs. So, if you need to add *Water Objects* to the scene at run-time first instantiate them and then call
`WaterObjectManager.Instance.Synchronize();`
This will signal to *Water Object Manager* that all the objects are instantiated and that it can synchronize the data.

Alternatively, you can instantiate your *Water Objects* on scene load in Awake() and immediately disable them. *Water Object Manager* will by default run *Synchronize()* in Start() and will register *Water Objects* even if they are disabled - and those objects can then later be enabled without calling *Synchronize().* Disabled *Water Objects* incur no overhead and you can enable/disable them at will.

Water Particle System
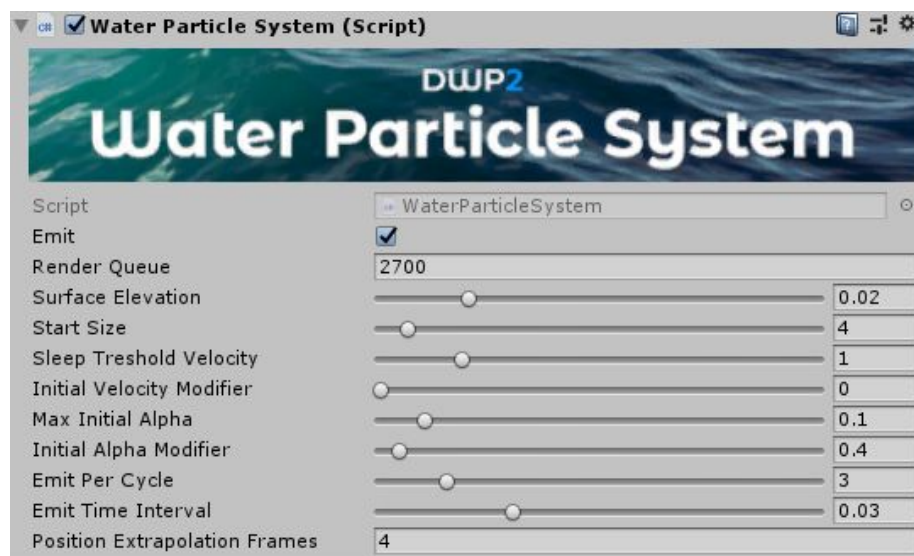
Water Particle System is a component that generates particles based on simulation data from *Water Object Manager*. It can be added to any *Water Object*. *Water Particle System* has been rewritten from the ground up for DWP2 and now it has virtually no performance or memory overhead except for the cost of the Unity's *Particle System* it uses to render particles.

Water Particle Systems emits only along X-Z axis and does not work with wavy water assets. For that asset-specific foam has to be used (if available).

Fields
- *Emit* - Particles will only be generated when this field is ticked.
- *Render Queue* - Render queue of the particle material. If particles are rendered behind the water increase the value to be just above the value of the water's render queue.
- *Surface Elevation* - Height above water surface at which the particles will be emitted.
- *Start Size* - Starting diameter of the particle.

- *Sleep Threshold Velocity* - If rigidbody's velocity is below this value particles will not be emitted. Do not set to 0 as that will result in (invisible) particles constantly being generated, even when object is still.
- *Initial Velocity Modifier* - Velocity at the point of contact with water is multiplied by this value to get the initial particle velocity. If set too high it will seem as if the particles are flying away from the object.
- *Max Initial Alpha* - Maximum initial alpha (transparency) of the foam. If set to 1 foam will be opaque, 0 and it will be invisible.
- *Initial Alpha Modifier* - Higher contact force with water will result in higher initial alpha (up to *Max Initial Alpha)*. This field sets the sensitivity of alpha related to the force.
- *Emit Per Cycle* - How many particles should be emitted in each cycle? If there are not enough contact points with water less particles may be emitted.
- *Emit Time Interval* - Interval between emission cycles in seconds.
- *Position Extrapolation Frames* - To counteract the initial fade-in and apparent lag of the particles, the emission position is predicted a number of frames in advance. If this number is set too high particles will appear as if emitting in front of the object.

Rigidbody Mass From Children
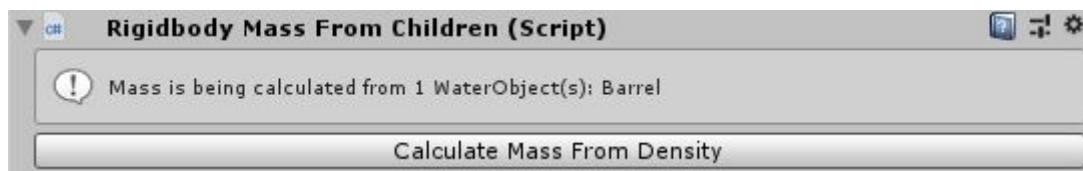
A helper script that calculates the mass of a *Rigidbody* from all the *Water Objects* attached to it. An example would be a trimaran - it has three hulls under one (parent) *Rigidbody*. If all three hulls have a *Water Object* the mass of the trimaran will be automatically calculated as the sum of the masses of all three hull *Water Objects.*

Center Of Mass
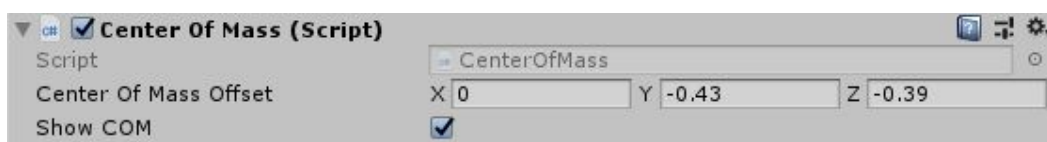
A helper script that adjusts the center of mass of a *Rigidbody*. Unity calculates the center of mass as a geometrical center of all the colliders that are children of the *Rigidbody* which is correct only if the object is made out of a single material with no voids. In most cases *Center Of Mass* needs to be adjusted, especially with ships which are designed to have very low centers of mass to prevent tipping over to the side.
Current center of mass is indicated with a green sphere.

Anchor
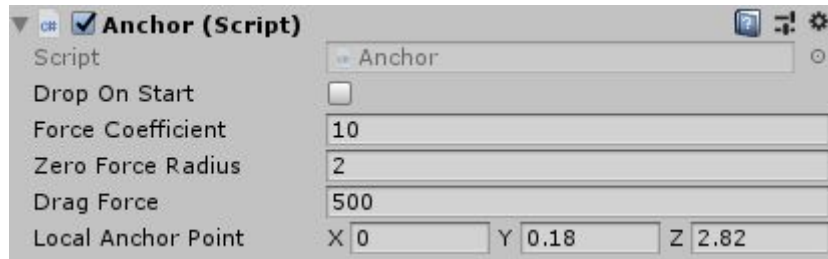
Anchor is a helper script that imitates the behavior of an anchor. It will try to keep the object in place exerting up to *Drag Force* at which point the anchor starts to drag and will move with the object, with the *Drag Force* being constantly applied in the direction of the anchor, pulling at the object. Once the object is outside of *Zero Force Radius* from the *Anchor.Drop()* point force will start to be applied as a function of the object distance from rim of the circle described by the *Zero Force Radius.*
Force Coefficient adjusts the amount of the force, while the *Local Anchor Point* adjusts where on the object the force will be applied (blue sphere gizmo).