WaterObject Manual

Quick Start

This quick start guide will explain how to set up a primitive sphere to interact with water.
An empty scene will be used for the example.

Initial Project Setup

Project Settings > Player > Api Combatibility Level needs to be set to .NET 4.x.

Assembly Definitions

Since all the NWH assets have been updated to use assembly definitions here is a disclaimer to avoid confusion when updating:

This asset uses Assembly Definition (.asmdef) files. There are many benefits to assembly definitions but a downside is that the whole project needs to use them or they should not be used at all.

- If the project already uses assembly definitions accessing a script that belongs to this asset can be done by adding an reference to the assembly definition of the script that needs to reference the asset. E.g. to access AdvancedShipController adding a NWH.DWP2 reference to MyProject.asmdef is required.
- If the project does not use assembly definitions simply remove all the .asmdef files from the asset after import.

Using, for example, Lux Water (which does not fature assembly definitions) will therefore require an addition of .asmdef file inside the Lux Water directory and a reference inside NWH.DWP2.asmdef or removal of all .asmdef files from the asset if you do not wish to use assembly definitions. Some assets such as Crest already feature .asmdefs and adding Crest as a reference to NWH.DWP2 is the only step needed.

Water Object Manager

`WaterObjectManager` has been removed in v2.5. All the settings and simulation is now handled by the `WaterObject`

Water Object

Any physics object that is active and has `WaterObject` attached will interact with water. There are two requirements for `WaterObject` to work: a `Rigidbody` and a `MeshFilter`:

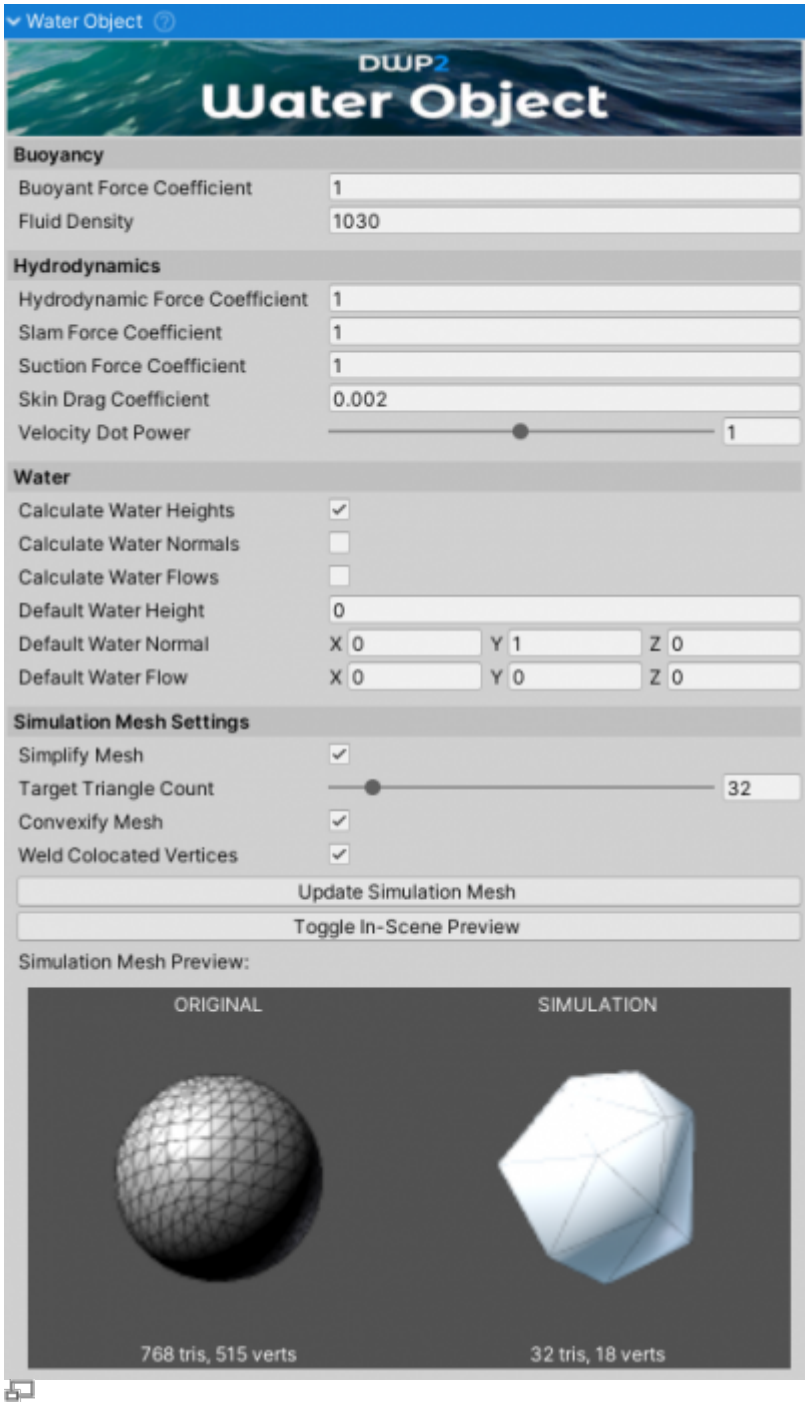
- `MeshFilter` is required so that the `WaterObject` knows which mesh to use for simulation.
- `Rigidbody` does not have to be attached to the same object as `WaterObject`, but it must be present in one of its parents. This allows for composite objects; one Rigidbody with multiple hulls - such as a trimaran.

Example Manual Setup

1. Add a *3D Object* ⇒ *Sphere* to the scene.
2. Add a `Sphere Collider` to the Sphere if is not automatically added.
3. Add a `Rigidbody` to the Sphere and set its mass to 300. There is also a script called `MassFromMaterial` which can calculate and set the `Rigidbody` mass based on material density and mesh volume, but it is a helper script and not required.
4. Add `WaterObject` to the Sphere. Since the sphere by default has 768 triangles `Simplify Mesh` option should be used. This option automatically decimates the mesh to a `Target Triangle Count`. A good triangle count is 30 or less for simple objects and around 60 for ship hulls. Using higher triangle count will only have a minor influence on simulation quality but will have a linear performance penalty (doubling the triangle count will about halve the performance). Therefore, adjusting the triangle count until the object starts to lose its shape is recommended. In the case of the example sphere 36 will be enough:

Example Auto Setup

1. Add a *3D Object* ⇒ *Sphere* to the scene.
2. Attach `WaterObjectWizard` to the sphere and press *Auto-Setup*.



Example WaterObject setup.

Water Data Provider

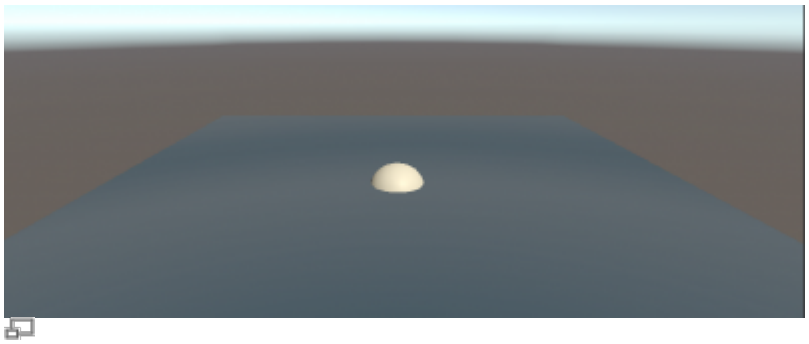
`WaterDataProvider` is a script that tells `WaterObject` where the water is.
It is an interface between water systems/assets and DWP2 and allows the two to communicate. All flat water assets/shaders use the same `WaterDataProvider`: `FlatWaterDataProvider` while for wavy assets such as Crest, an asset-specific `WaterDataProvider` has to be used, e.g. `CrestWaterDataProvider`.
As of version v2.5 an option to use multiple water surfaces in the same scene has been added. This is done by attaching a `Collider` with `isTrigger = true` to the `WaterDataProvider`. As long as the object is inside the trigger it will use data from that `WaterDataProvider`.

Minimal Setup

1. Add a Cube (or any other mesh) to the scene.
2. Attach `WaterObject` to the Cube. Make sure that a Rigidbody has been added.
3. Press Play. The object will now float at default water height (set under `WaterObject` settings).

Adding Water

To add water to the scene check the Water Assets page.
`FlatWaterDataProvider` can be used to make water height follow a flat primitive plane which results in something like this:

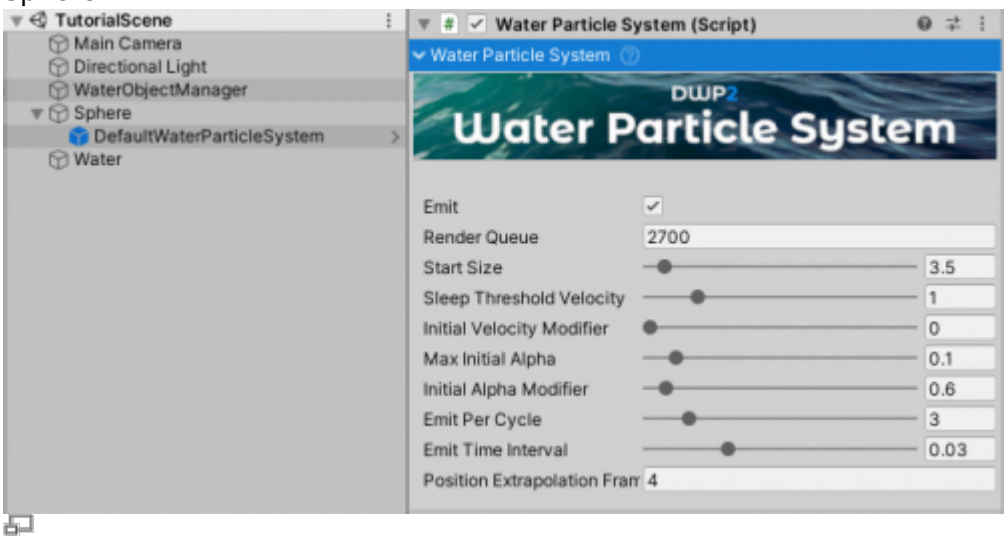


A basic primitive plane used as water.

Water Particle System

`WaterParticleSystem` can be used to generate foam. It works with any flat water.

1. Drag *DefaultWaterParticleSystem* from *DWP2* ⇒ *Resources* into the scene and parent it to the Sphere.

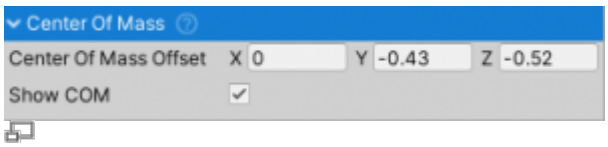


Example WaterParticleSystem setup.

2. Move the Sphere to be above the water and press play. Sphere falling into the water will generate foam around it based on simulation data. `WaterParticleSystem` and `ParticleSystem` values can be tweaked to suit the needs of the project.

Center Of Mass

`CenterOfMass` is a simple script that offsets the center of mass of a Rigidbody.
Unity calculates center of mass of the object from all of its colliders, as if the object were homogenous. In many cases this is not correct - a ship has ballast, a crate could have some load in it, a barrel could have oil, etc.
To adjust the center of mass of an object simply attach `CenterOfMass` script to the same object that contains the `Rigidbody` and adjust the `Center Of Mass Offset` - a value in local coordinates which tells how much to offset center of mass from the Unity-calculated point. Want a ship to be less prone to capsizing? Lower the Y component of COM.



CenterOfMass inspector.

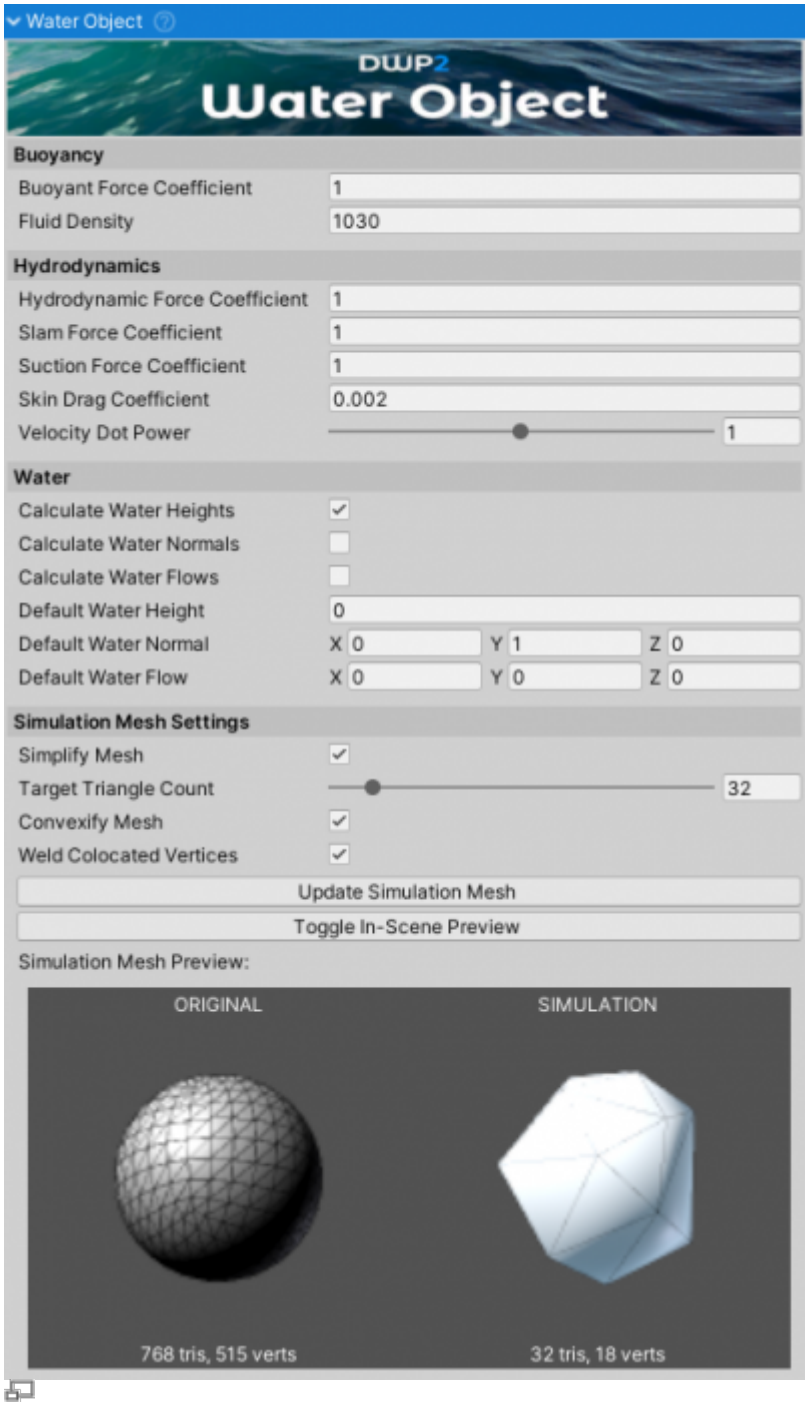
Water Object Wizard

`WaterObjectWizard` is a helper script that sets up a `WaterObject` automatically. It is still recommended to have knowledge of manual setup and how things work, but this script can automate and speed up the setup process.
A primitive Sphere will be used, same as in the rest of the guide above.

1. Add a *3D Object* > *Sphere* to the scene.
2. Add `WaterObjectWizard` to the newly create Sphere.
3. Tick *Add Water Particle System* (optional). This option is self-explanatory.
4. Click `Auto-Setup` and press Play after the setup is done. The Sphere now floats and generates foam. Next step would be to manually check and tweak the default values, such as `Target Triangle Count`, center of mass, etc.

2020/04/26 12:56 · Aron Rescec

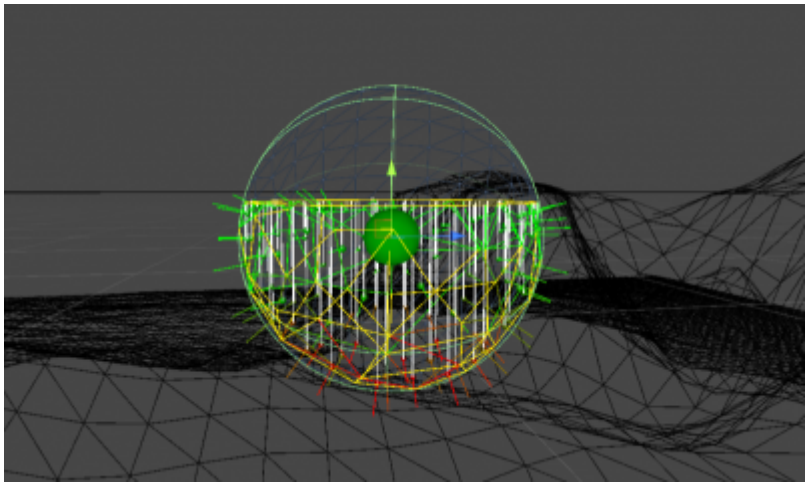
Water Object



WaterObject inspector.

`WaterObject` is the main script of DWP2. It handles all the buoyancy and hydrodynamics calculations.

`WaterObject` gets the data needed for simulation from the *Simulation Mesh*. This is a static mesh that will be used for simulating water/object interaction and can either be the original mesh or a simplified version of it. `WaterObject` provides basic tools for mesh triangle decimation, removal of co-located vertices, and convexification - if needed.



WaterObject with WaterObjectManager's Debug field enabled.

Fields

To get info about individual fields hover the mouse over the field and a tooltip will pop up.

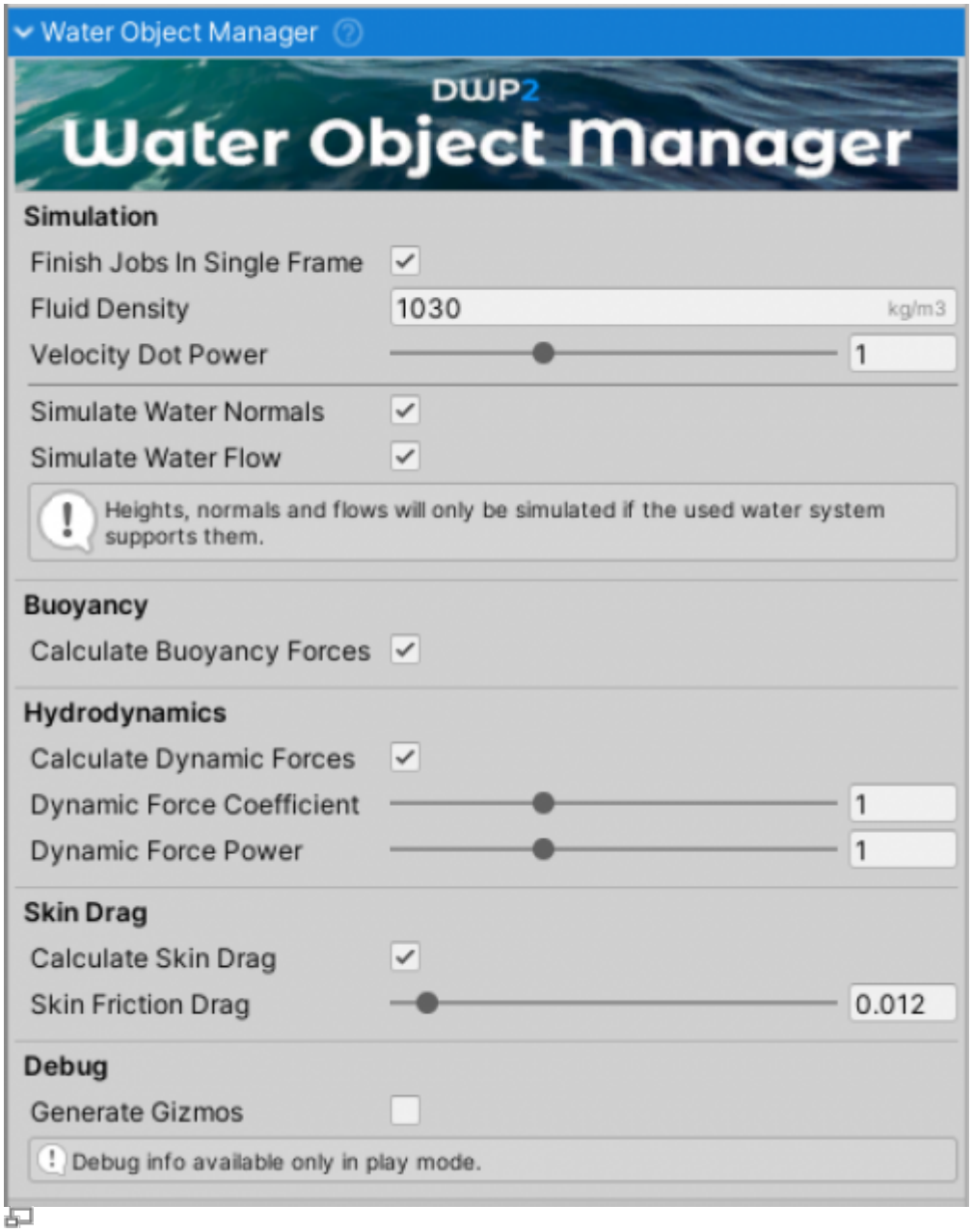
Instantiating at Run-time

Check WaterObjectManager page for more info about instantiating `WaterObjects` at run-time.

2020/07/16 13:28 · Aron Rescec

Water Object Manager

WaterObjectManager has been removed in v2.5. This page still exists for backwards compatibility purposes



WaterObjectManager inspector.

WaterObjectManager is the main script of Dynamic Water Physics 2 and has to be present in the scene for water/object interaction to work. It does not matter to which object it is attached, it just needs to be present. WaterObjectManager fetches the data from all the WaterObjects in the scene, processes it and sends it to a job which then does all the physics calculations and makes use of multiple CPU cores.

Fields

- `Finish Jobs In Single Frame` - if set to true jobs will be finished on the same frame they started. Disabling it will improve performance, but will also add a one-frame delay to water/object interaction. This is not significant in most cases. Default is off.
- `Fluid Density` - density of the simulated fluid in kg/m3. Default is 1030 (salt water).
- `Velocity Dot Power` - when set to one fluid will act as a Newtonian fluid (relationship between velocity and impact force is proportional). Values below 1 will result in pseudo-plastic liquid (relationship is logaritmic) and values above 1 will result in dilatant liquid (relationship is

exponential). This is a simplified explanation. Value of 1 should be used for normal water, but slightly higher value can also be used.
- `Simulate Water Normals` - if true water normals will be used when calculating forces. Has no effect if the water system does not use normals.
- `Simulate Water Flow` - if true water flow will be used when calculating forces. Has no effect if the water system used does not support flow.
- `Calculate Buoyancy Forces` - buoyancy will be calculated when this field is true.
- `Calculate Dynamic Forces` - hydrodynamics will be approximated when this field is true. Disable to use only buoyancy.
- `Dynamic Force Coefficient` - a coefficient by which the resultant force will be multiplied.
- `Dynamic Force Power` - an exponent to the power of which the resultant force will raised.
- `Calculate Skin Drag` - when true a drag will be calculated as a result of liquid passing over a surface.
- `Skin Friction Drag` - can be set to 0 or near 0 for water while goo-like fluids will have a larger value.
- `Generate Gizmos` - gizmos will be generated from simulation data when true.

Instantiating WaterObjects at runtime

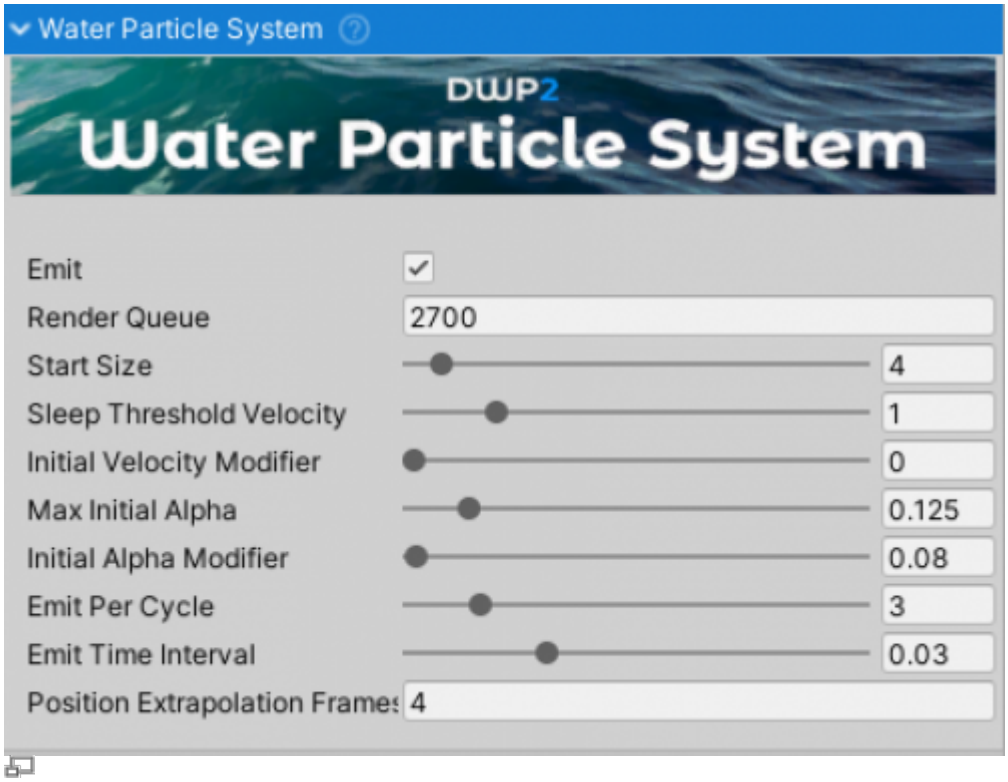
When Water Object is added to the scene during play mode it will not be automatically registered because this requires re-allocating memory for the jobs which is quite expensive and should ideally be done during the loading screens.

There are two ways to add new objects to the scene during runtime:

- When there is a low number of triangles instantiating the object and immediately calling `WaterObjectManager.Instance.Synchronize()` is the best solution.
- Instantiating all the objects during scene load and deactivating them. `WaterObjectManager` will allocate memory for inactive objects too, but will not simulate them. When the object is needed it can simply be activated, without calling `Synchronize()`.

2020/07/16 13:26 · Aron Rescec

Water Particle System



WaterParticleSystem inspector.

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.

2020/07/16 13:28 · Aron Rescec

Supported Water Assets

`WaterDataProvider` scripts are interfaces between the 3rd party water assets and Dynamic Water Physics 2. They tell the WaterObject where the water is.

Due to inclusion of assembly definitions into this asset with v2.4 an additional step is required when setting up 3rd party water assets. The asset either needs to be referenced inside `NWH.DWP2.asmdef` or all the .asmdef (assembly definition) files need to be removed from DWP2. If the 3rd party asset does not include an assembly definition file one should be added manually to the root of that asset, or the .asmdef files need to be removed from DWP2.

If removing .asmdef files make sure to tick *Project Settings > Player > Allow '**unsafe**' code*. This code will not harm your device, it just allows unmanaged memory access (i.e. pointers) which are used for performance optimization inside DWP2.

Flat Water

`FlatWaterDataProvider` can be used for all flat water systems.
It can even be used with wavy water systems if the waves have 0 amplitude to improve performance (sometimes drastically as the water heights are always queried with wavy water system, even if there are no waves).

Setup

- Attach `FlatWaterDataProvider` to the `GameObject` representing the water.
- `WaterObjects` will now float at the water `transform.y` position.

2020/07/16 13:50 · Aron Rescec

Crest

Screenshot of demo scene for DWP2 with Crest.

Dynamic Water Physics 2 is compatible with Crest v10 or newer. Older versions have different API.

Crest supports water heights, normals and flows.

Crest Import

1. Download and import Crest and Crest-Examples from here. For HDRP and URP version check Unity Asset Store.
2. Open Crest-Examples ⇒ Main ⇒ Scenes ⇒ main scene.
3. Remove the existing boat from the scene.

Adding a CrestWaterDataProvider

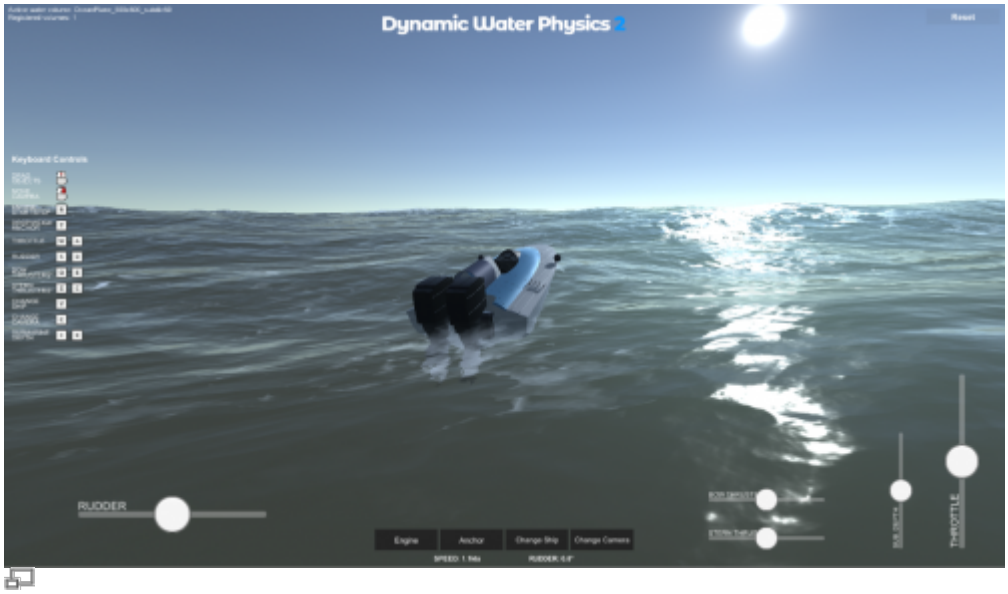
1. Add `DWP_CREST` to *Project Settings > Player > Scripting Define Symbols*.
2. Find the `GameObject` containing the `OceanRenderer` component.
3. Attach `CrestWaterDataProvider` to the object.

Adding a WaterObject

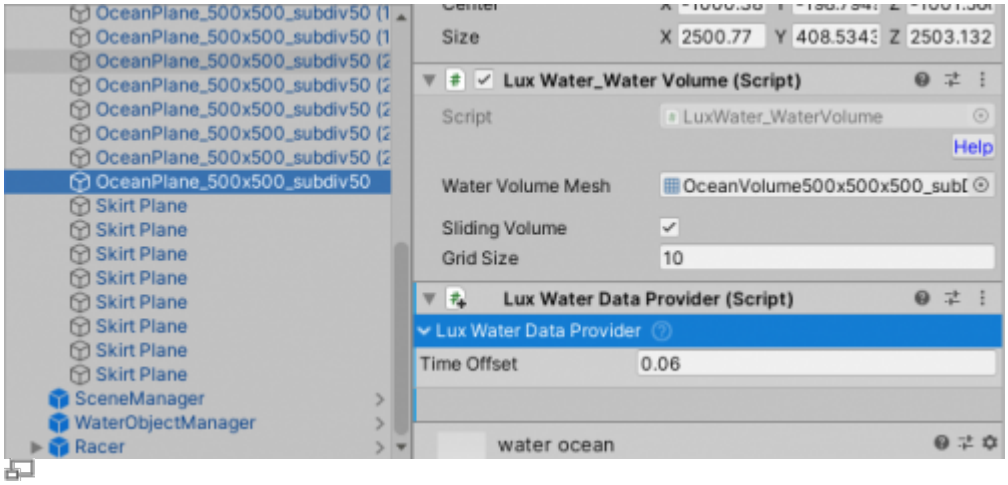
1. Drag `Racer` prefab into the scene.
2. Press play. The boat will now float and follow the waves properly. If this is not the case check that the console states `DWP: Using Crest` and there are no errors or warnings.

2020/07/16 13:41 · Aron Rescec

Lux



Lux Water demo scene with DWP2.



Example Lux Water setup.

Lux Water supports water heights.

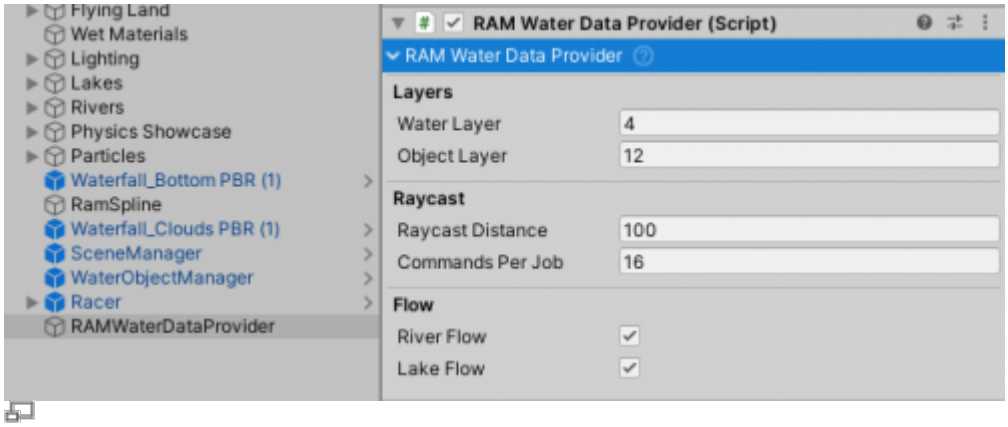
Setup

1. Add `DWP_LUX` to *Project Settings* > *Player* > *Scripting Define Symbols*.
2. Attach `WaterDataProvider` to `LuxWater_WaterVolume`.

2020/07/16 13:41 · Aron Rescec

River Auto Material (R.A.M.)

R.A.M. demo scene with DWP2 added.



Example River Auto Material setup.

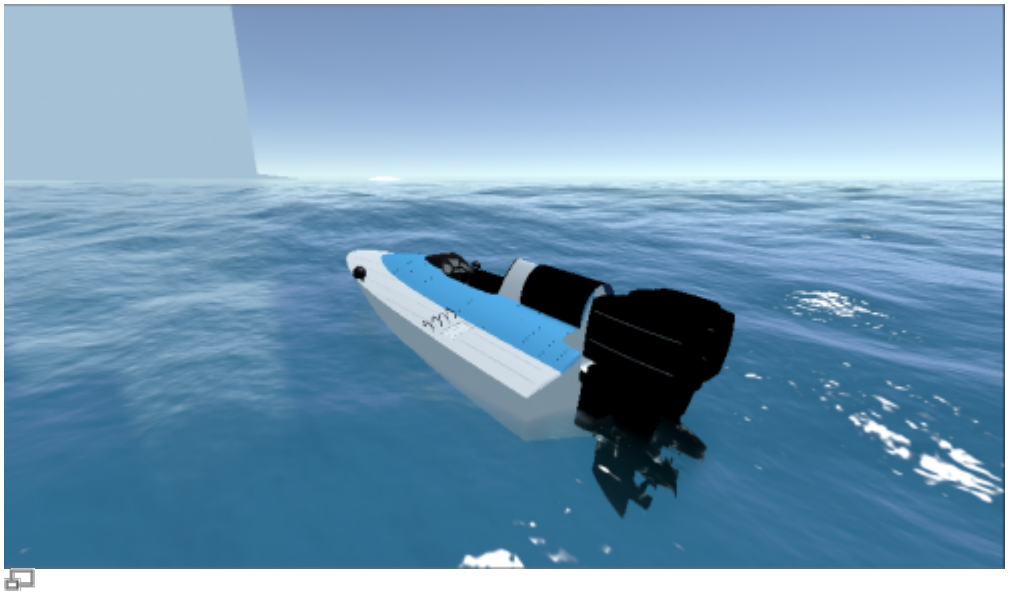
`RAMWaterDataProvider` supports water heights, normals and flow and it inherits from `RaycastWaterDataProvider`.

Setup

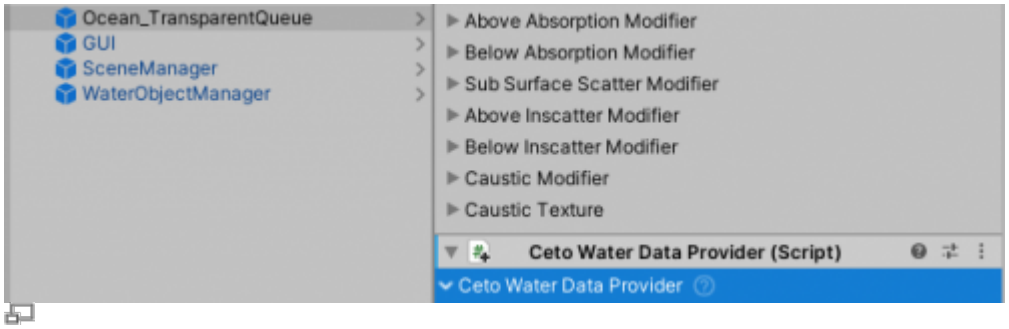
- Add `DWP_RAM` to *Project Settings* > *Player* > *Scripting Define Symbols*.
- Set up the scene as if using flat water, minus FlatWaterDataProvider.
- Add RAMWaterDataProvider to the scene. It does not have to be attached to any specific object.
- Make sure that the RAM object has a MeshCollider attached. This is required for Raycasts to work.
- Assign Water Layer to the River Auto Material water. Assign Object Layer to all the WaterObjects in the scene. This is an important step as the script will disable physical collisions between the two layers to prevent the WaterObjects from sitting on top of the mesh collider that R.A.M. uses instead of interacting with water.
- R.A.M. setup is ready to go.

2020/07/16 15:01 · Aron Rescec

CetoSetup



Ceto demo scene with DWP2.



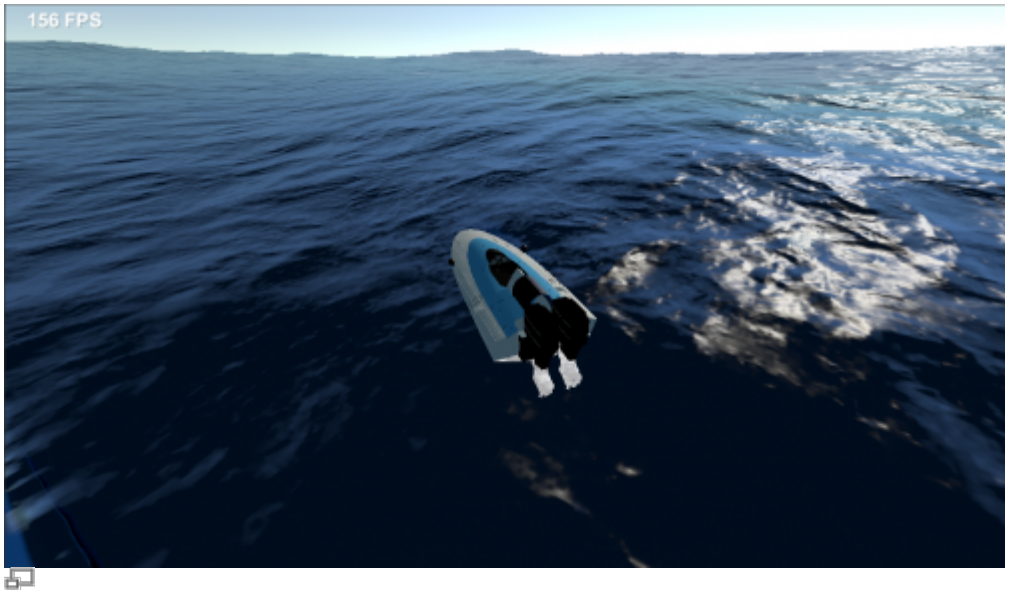
Example Ceto setup.

Setup

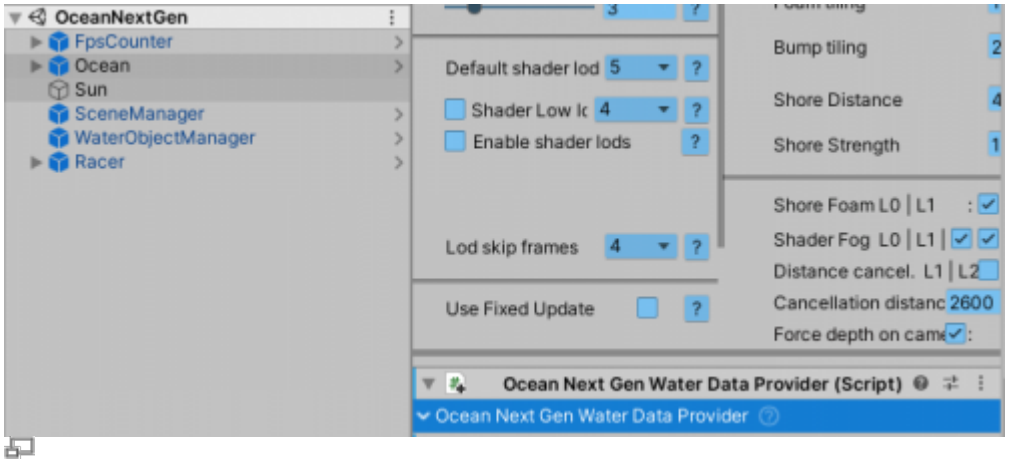
1. Add `DWP_CETO` to *Project Settings* > *Player* > *Scripting Define Symbols*.
2. Attach `WaterDataProvider` to `Ocean`.

2020/07/16 15:38 · Aron Rescec

Ocean Nex Gen



Ocean Next Get demo scene with DWP2.



Ocean Next Gen example setup.

Ocean Next Gen is supported but has not been updated regularly for over two years. Crest and Ceto will be better options.

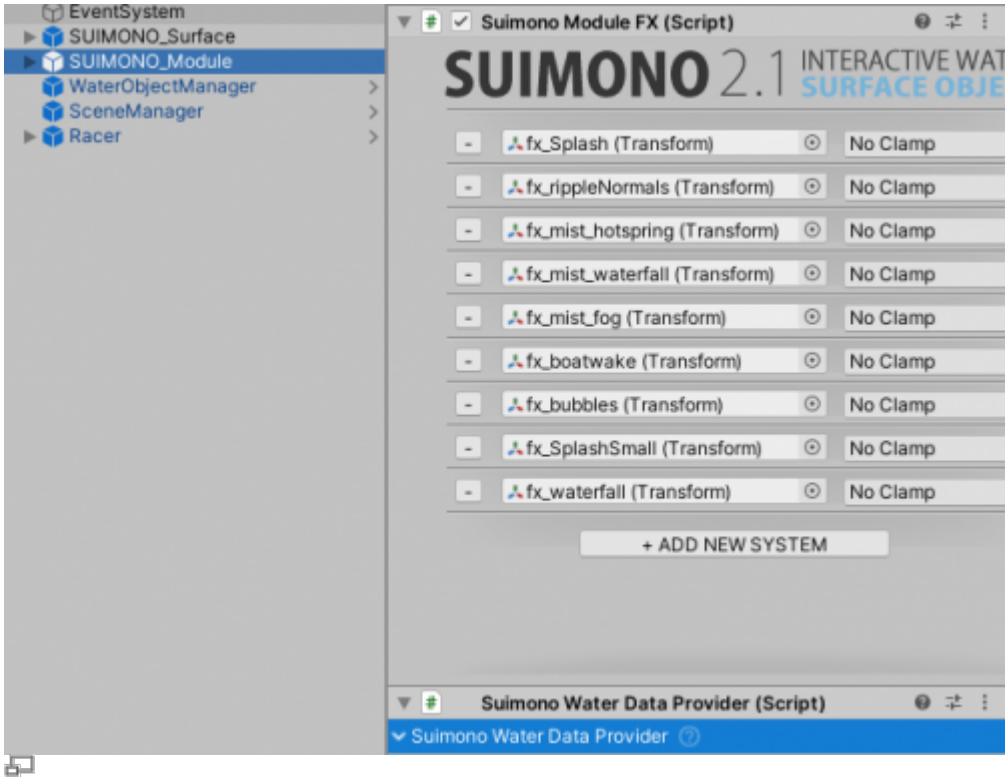
Setup

1. Add `DWP_OCEAN_NEXT_GEN` to *Project Settings* > *Player* > *Scripting Define Symbols*.
2. Attach `OceanNextGenWaterDataProvider` to `Ocean`.

2020/07/16 15:10 · Aron Rescec

Suimono

Suimono demo scene with DWP2 added.



Example Suimono setup.

Setup

- Rename SuimonoWaterDataProvider.cs.txt to SuimonoWaterDataProvider.cs.
- Add `SuimonoWaterDataProvider` to the object containing `SuimonoModule` script.

2020/07/16 14:48 · Aron Rescec

Stylized Water 2

Setup

Unlike other `WaterDataProviders`, the one for Stylized Water 2 is included with the Stylized Water

2 asset instead of DWP2.

- Set up the scene as per quick start guide for flat water, minus FlatWaterDataProvider.
- Go to Help → Stylized Water 2, and click the “Install integration” button. Wait until scripts have finished compiling.
- Add `StylizedWaterDataProvider` to the object containing OceanRenderer script.

2021/06/08 14:19 · Aron Rescec

KWS

KWS support is under development.

2021/07/12 10:42 · Aron Rescec

Multiple Assets / Water Types

Multiple water types can be used in the same scene at the same time. This is achieved through triggers (`Colliders` with `isTrigger` set to true) attached to the same `GameObject` as the `WaterDataProvider` in question.
By default these colliders are created automatically on `Awake` and are set to cover the whole world. However, it is possible to use a `WaterDataProvider` just for a small part of the scene - such as a lake.

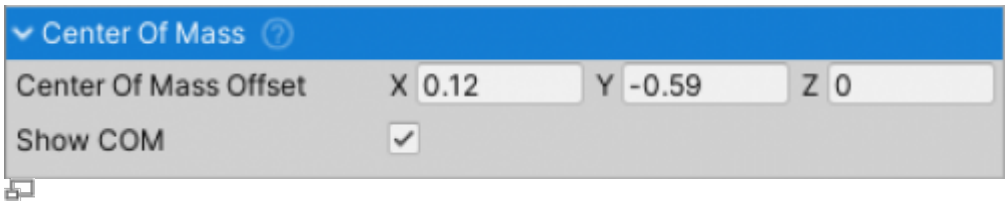
- Attach a `Collider` of any type (e.g. `SphereCollider`) to the `GameObject` containing `WaterDataProvider`.
- Tick `Is Trigger` on the collider.
- Adjust the size/radius of the collider to cover the area you want the `WaterDataProvider` to have the effect on.

If there are multiple trigger volumes they will act as a queue, meaning the one that the object last entered will be currently active.

2020/04/26 13:53 · Aron Rescec

Helper Scripts

Center Of Mass



CenterOfMass inspector.

This script has been replaced with `VariableCenterOfMass` in v2.5.

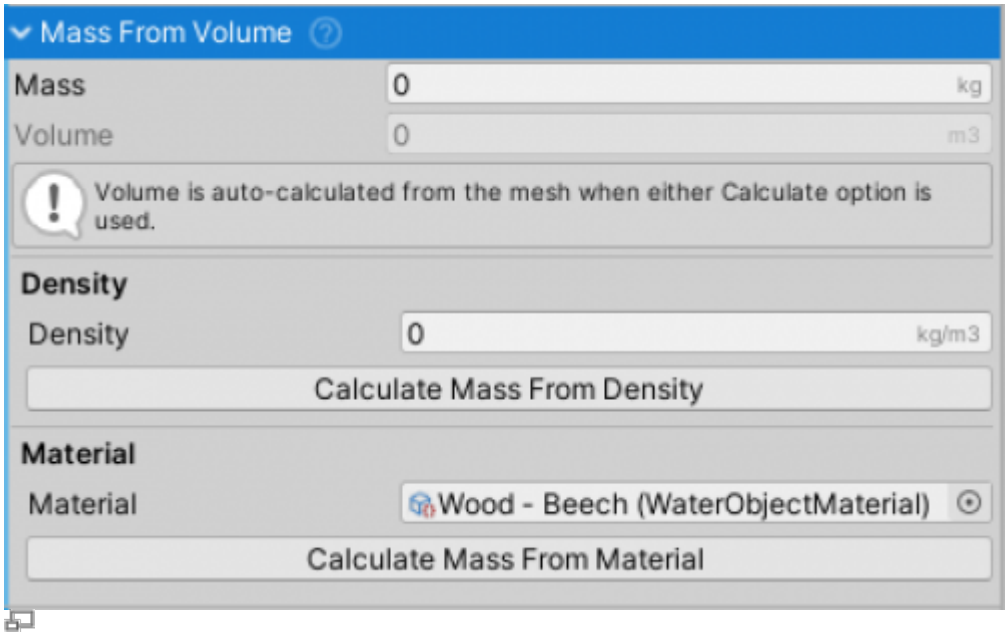
A helper script for setting center of mass of a `Rigidbody`.

Unity calculates center of mass as a center of volume of all the `Rigidbodys` `Colliders`. This can and will result in unrealistic center of mass for objects that do not have uniform density, i.e. center of mass will not have to be adjusted for a wooden log but a ship has a ballast and is noticeably heavier around the keel and therefore the center of mass is low which in turn prevents the ship from capsizing.

- `Center Of Mass Offset` - offset of the center of mass from the Unity calculated one in local coordinates.
- `Show COM` - when true a green gizmo sphere will be drawn at the current center of mass.

2020/07/16 19:24 · Aron Rescec

Mass From Volume



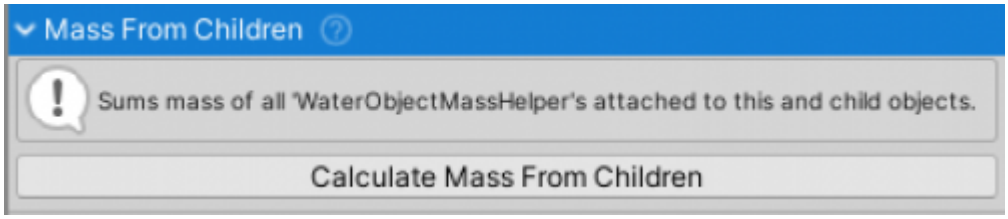
MassFromVolume inspector.

`MassFromVolume` is a helper script that calculates object's mass from volume of the mesh and the density.

- Volume of the mesh is calculated automatically.
- `Mass` field can also be set manually.
- Can be used together with `MassFromChildren` to calculate mass of complex objects (objects having more than one child `WaterObject`).

2020/07/16 21:42 · Aron Rescec

Mass From Children



MassFromChildren inspector.

A helper script for determining mass of a Rigidbody from the children. It sums the masses of all the children that have `WaterObjectMassHelper` scripts attached. This eliminates the need for guessing the mass of the object.

Usage

- Attach the script to the parent object. That object has to contain the `Rigidbody` component.
- Make sure that at least one child has `WaterObjectMassHelper` attached or the result will be 0 and ignored.
- Press *Calculate Mass From Children*. This will calculate and set the `Rigidbody`'s mass.

2020/07/16 19:32 · Aron Rescec
2020/04/28 13:15 · Aron Rescec