

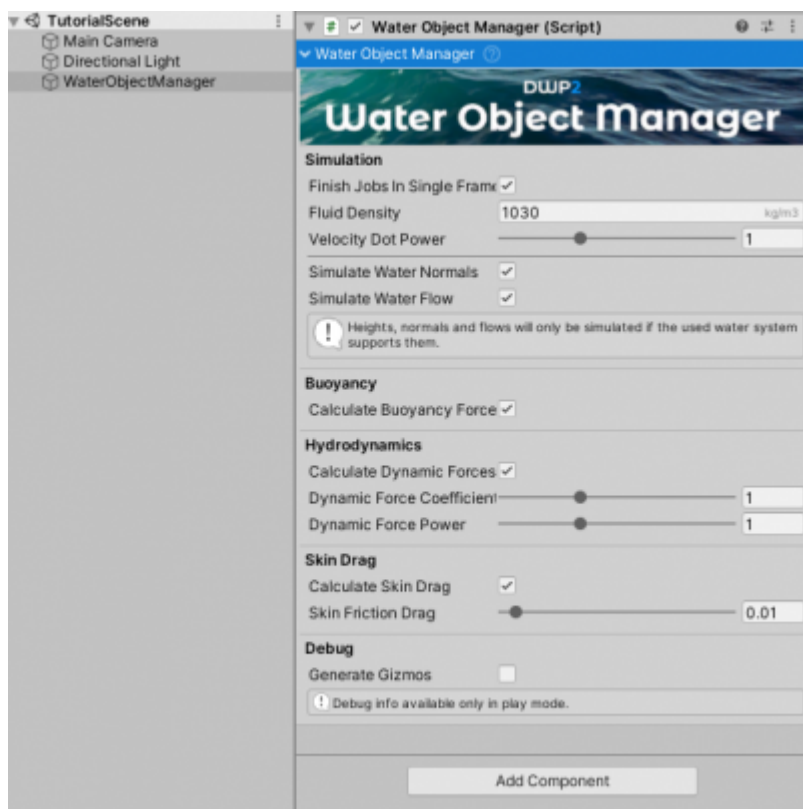
Quick Start

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.

Water Object Manager

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.

1. Add a WaterObjectManager to the scene. No need to change any settings. Example:



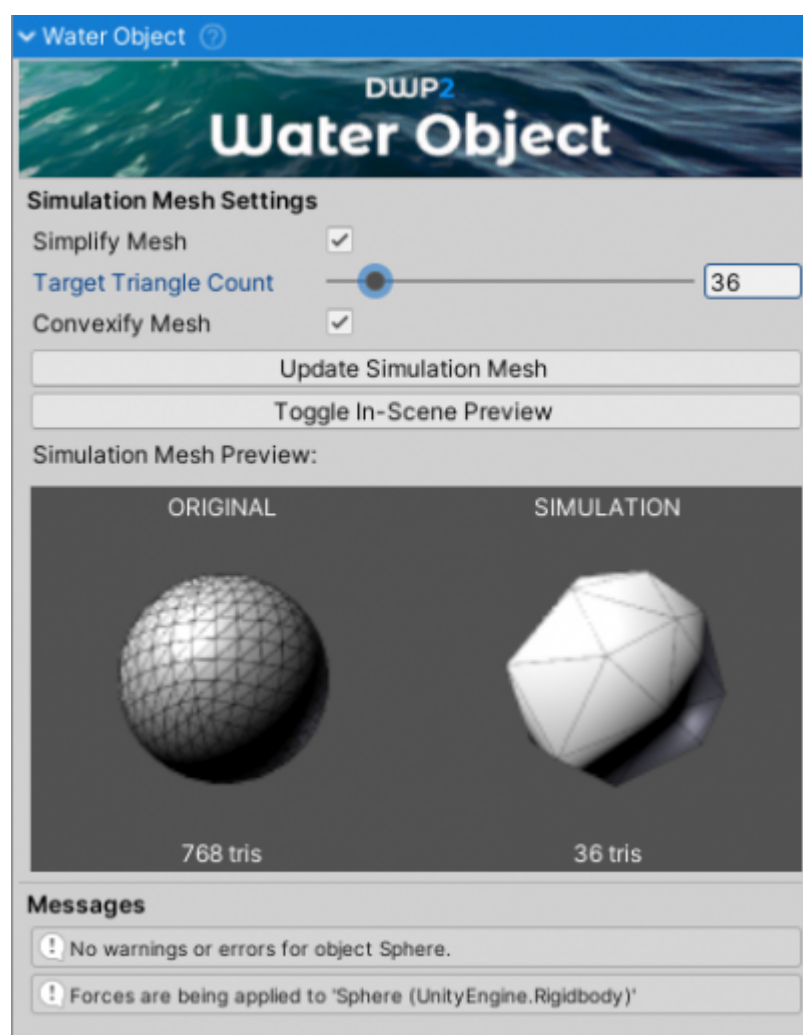
Example WaterObjectManager setup.

Water Object

WaterObject is a script that tells WaterObjectManager which objects should interact with water. Any object that is active and has this script will interact with water. There are two requirements for WaterObject to work: a Rigidbody and a MeshFilter from which the script will fetch the mesh (DWP2 uses mesh data to simulate object/water interaction). 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.

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 `WaterObject` setup.

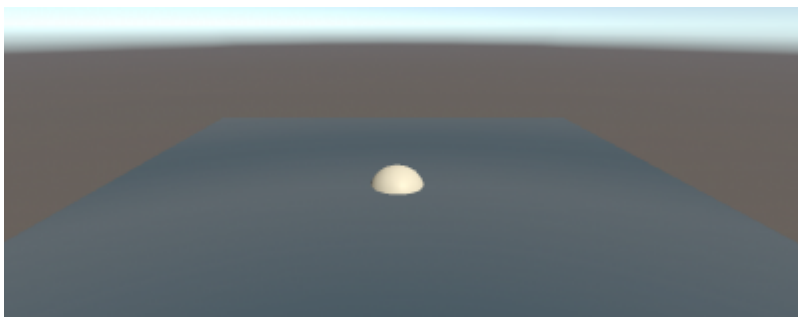
Water Data Provider

`WaterDataProvider` is a script that tells `WaterObjectManager` where the water is. It is an interface between water systems/assets and Dynamic Water Physics 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 an example a flat water plane will be used:

1. Create an empty object using *RMB* \Rightarrow *Create Empty* and name it *Water*. Name does not matter.
2. Add FlatWaterDataProvider to it and press Play. The object will now float but it will appear as if it floating in empty space (which it is). FlatWaterDataProvider assumes that water is at `transform.position.y` of the object it is attached to and does not care about any shaders, colliders, etc. Changing the Y value of Transform's position will make the Sphere float at different heights.
3. To make this look more realistic, a plane can be added instead. Delete the *Water* object created in step 1 and create a *3D Object* plane instead. Attach FlatWaterDataProvider and make sure to remove MeshCollider from it or the sphere will just sit on the plane through physics collision.
4. Press play again. The sphere floats the same as in step 2 but it appears it is in 'water'. At this point Default-Material of the plane can be replaced with a material that uses some kind of flat water shader - e.g. the included WaterProDaytime material which is just a standard Unity water.

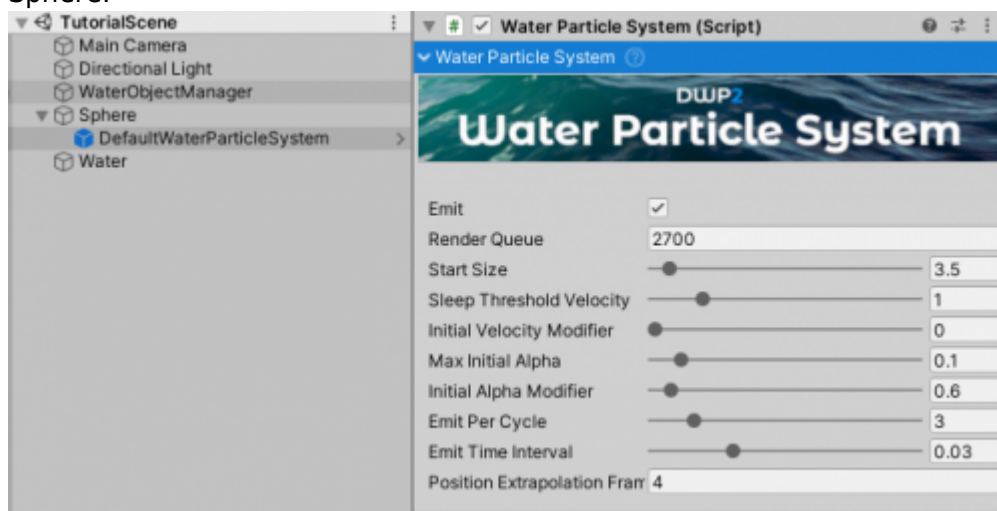


A result of following this guide. Not pretty, but fully functional.

Water Particle System

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

1. Drag *DefaultWaterParticleSystem* from *DWP2* \Rightarrow *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

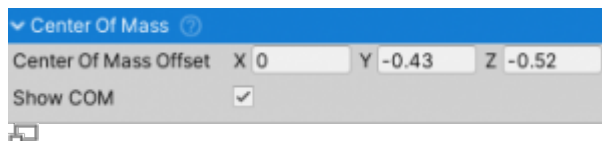
ParticleSystem values can be tweaked to suit the needs of the project.

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.

Enabling Burst

Burst is a compiler, it translates from IL/.NET bytecode to highly optimized native code using LLVM. This improves job performance.

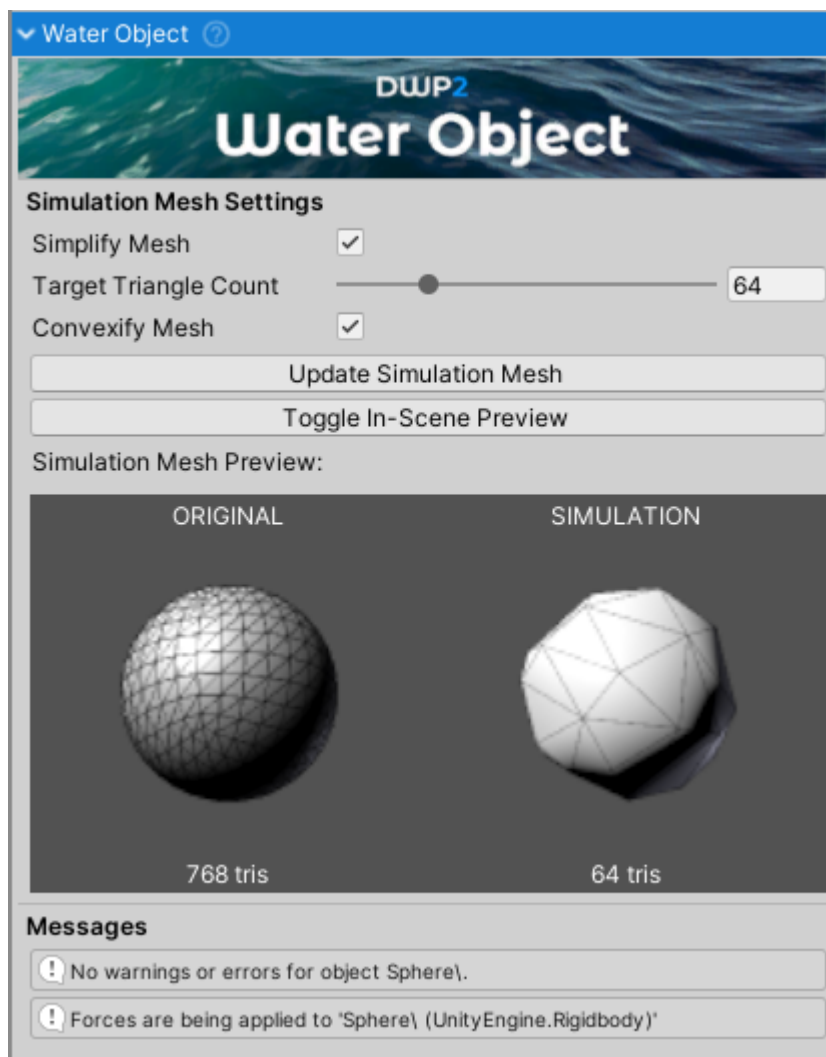
It can be installed from Unity PackageManager.

By default Burst compilation is disabled. This is due to the fact that cross compilation is not supported in some cases. E.g. building for Mac OS on Windows will fail if Burst is enabled.

To enable Burst compilation add `DWP_BURST` under *Project Settings* \Rightarrow *Player* \Rightarrow *Scripting Define Symbols*. Make sure that the *Burst* package installed beforehand.

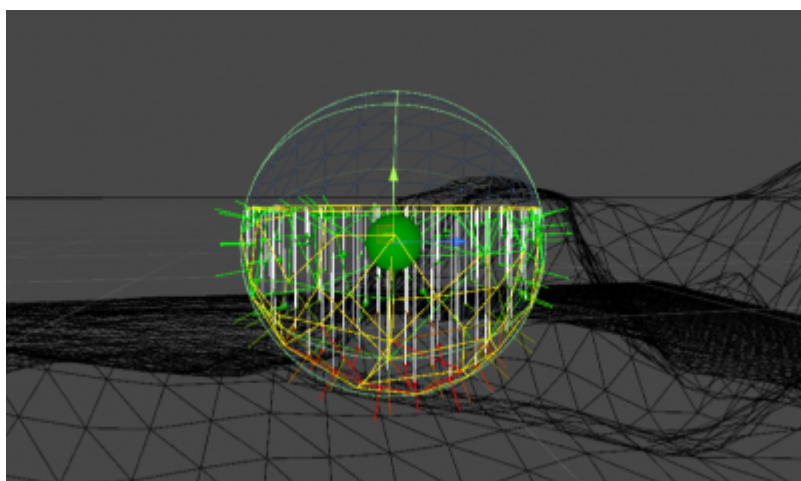
2020/04/26 14:56 · Aron Rescec

Water Object



WaterObject inspector.

WaterObject is a script that tells WaterObjectManager that the object should interact with water. It also handles *Simulation Mesh* generation. *Simulation Mesh* is the 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.



WaterObject with WaterObjectManager's Debug field enabled.

Fields

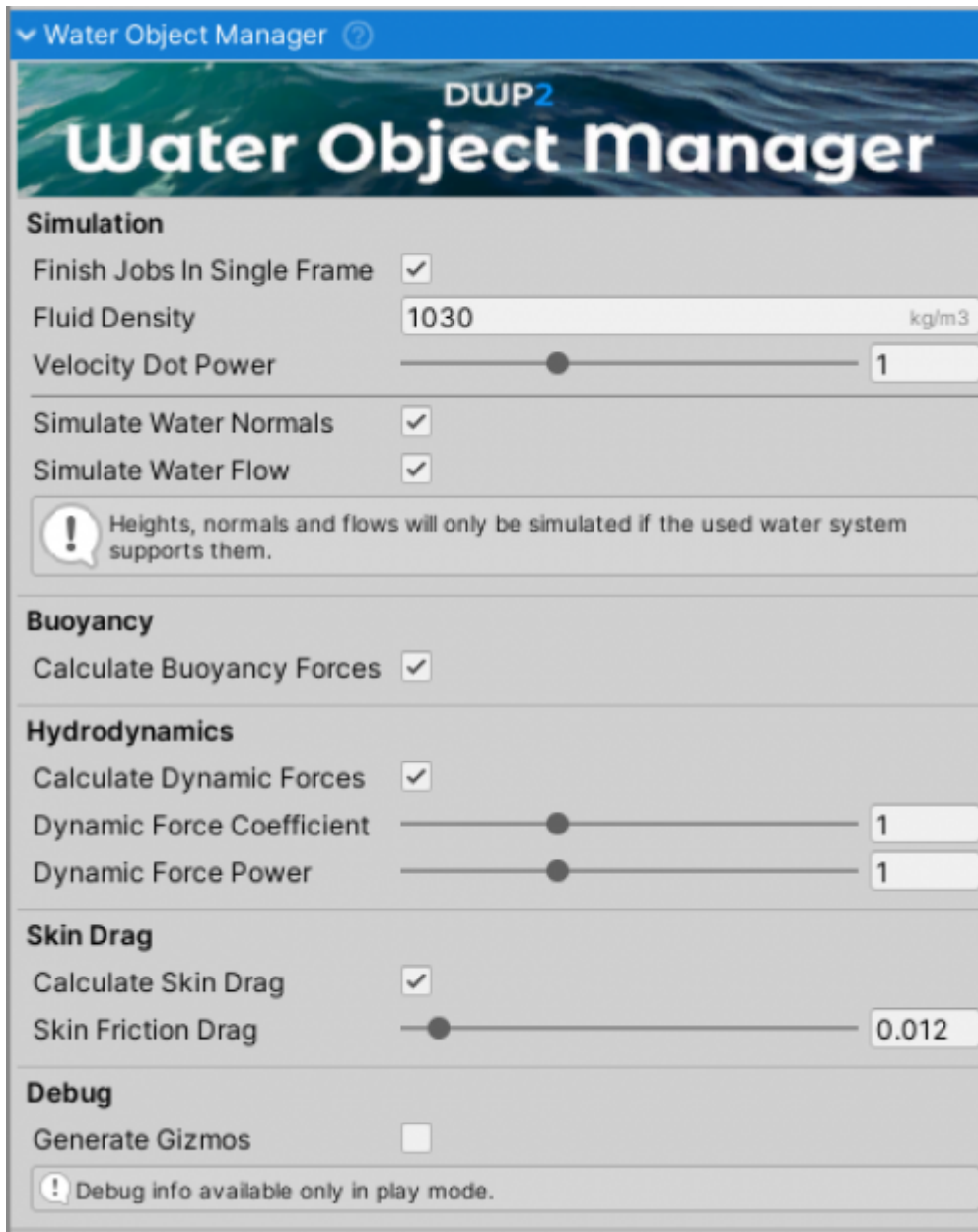
- **Simplify Mesh** - Should the simulation mesh be simplified? If the mesh has more than 30 triangles it is usually a good idea to enable this option.
- **Target Triangle Count** - How many triangles the simplified simulation mesh should have?
- **Convexify Mesh** - Should the simulation mesh be made 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 used.
- **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.

Instantiating at Run-time

Check [WaterObjectManager page](#) for more info about instantiating WaterObjects at run-time.

2020/07/16 15:28 · Aron Rescec

Water Object Manager



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 logarithmic) 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 be 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

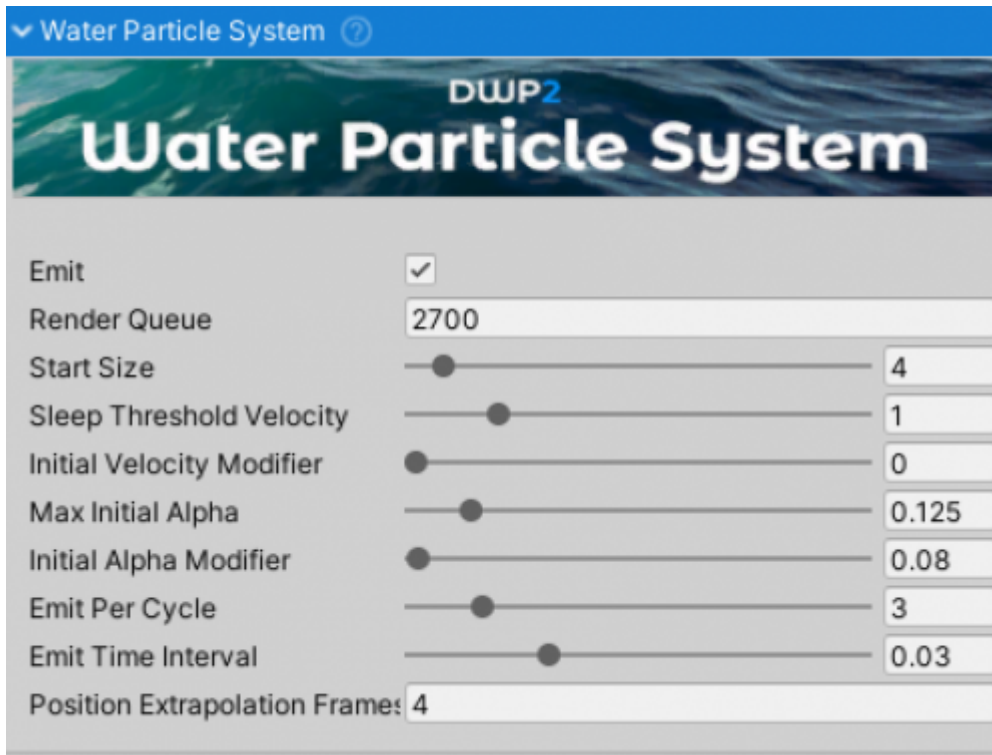
When Water Object is added to the scene during play mode it will not be automatically registered because this requires re-allocating memory for the jobs which is quite expensive and should ideally be done during the loading screens.

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 15:26 · 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 15:28 · Aron Rescec

Supported Water Assets

Flat Water Assets

FloatWaterDataProvider is 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).

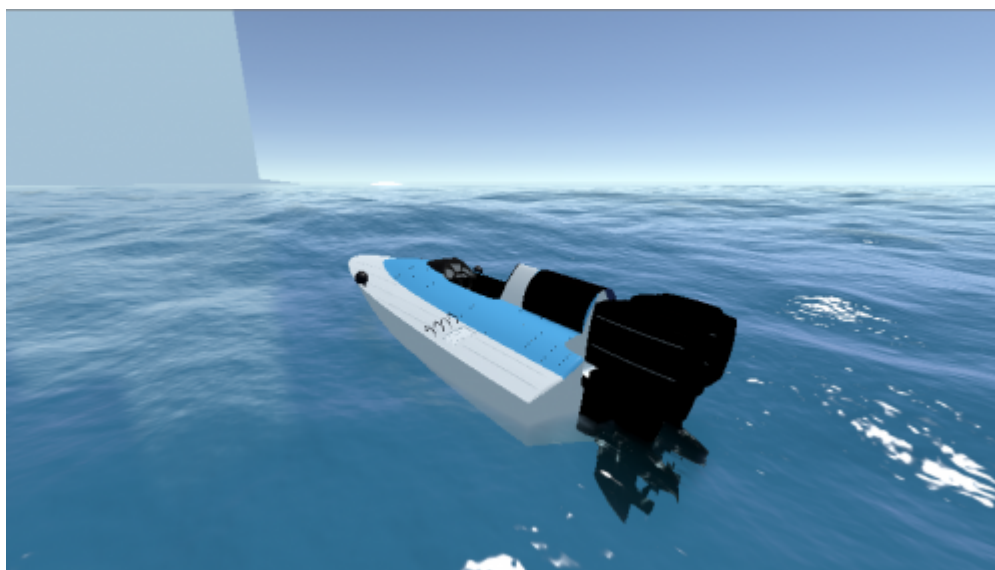
Setup

- Attach FlatWaterDataProvider to the GameObject representing the water.
- Water will exist at that object's transform.y position. No further action is needed.

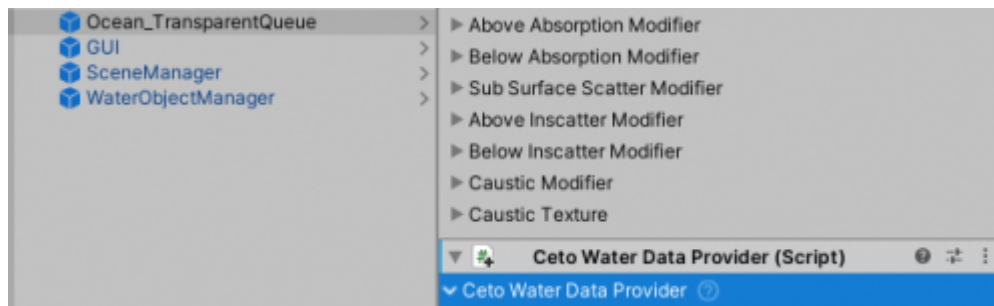
2020/07/16 15:50 · Aron Rescec

Wavy Water Assets

CetoSetup



Ceto demo scene with DWP2.



Example Ceto setup.

Setup

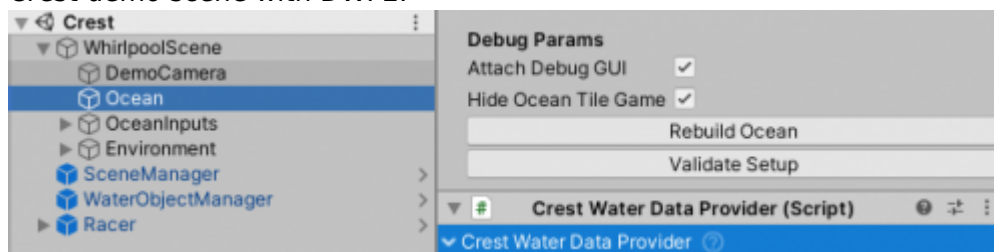
- Rename CetoWaterDataProvider.cs.txt to CetoWaterDataProvider.cs.
- Set up the scene as if using flat water, minus FlatWaterDataProvider.
- Add CetoWaterDataProvider to the object containing Ocean script.

2020/07/16 17:38 · Aron Rescec

Crest Setup



Crest demo scene with DWP2.



Example Crest setup.

Crest supports water heights, water normals and flow. It is one of the best wavy water systems to use with DWP2 and free to boot.

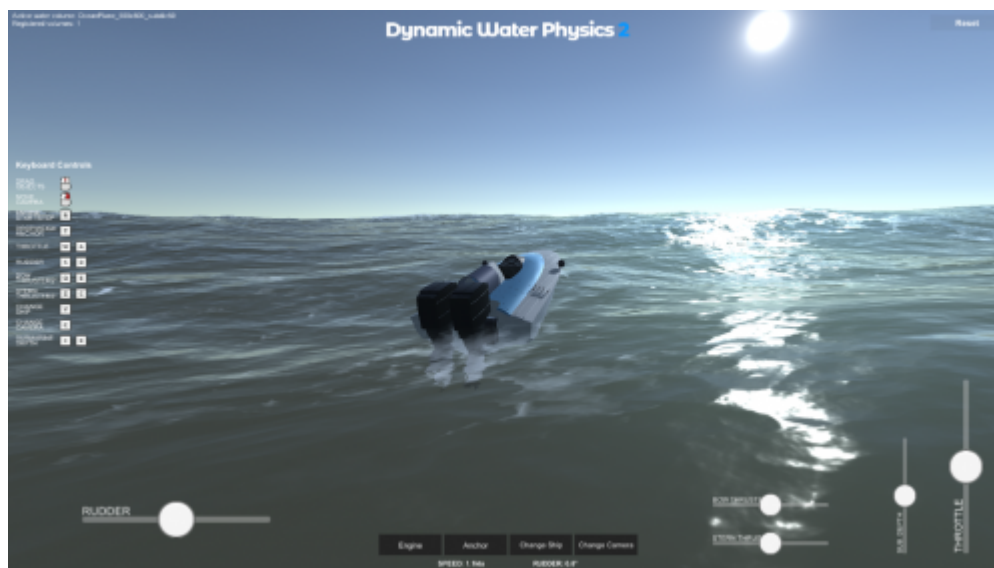
Please always use the latest Crest version with DWP2 as the Crest API has changed a few times in the past.

Setup

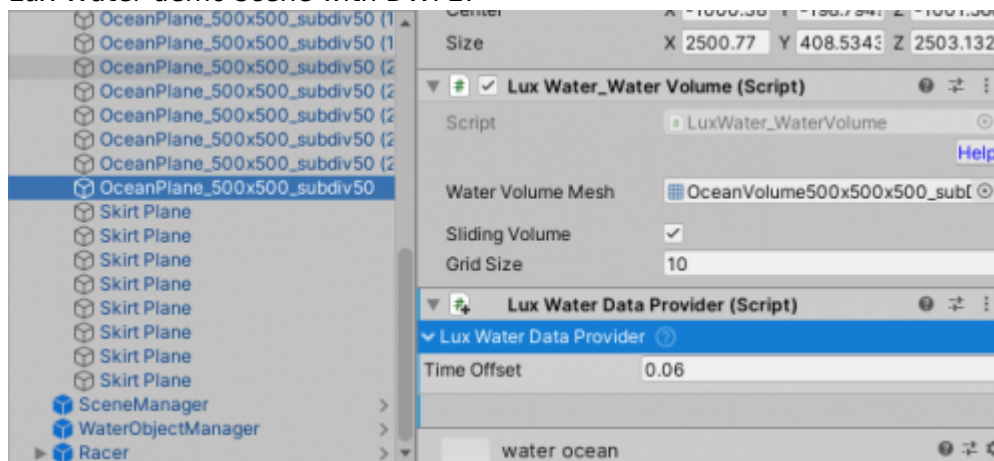
- Set up the scene as per quick start guide for flat water, minus FlatWaterDataProvider.
- Rename CrestWaterDataProvider.cs.txt to CrestWaterDataProvider.cs.
- Add CrestWaterDataProvider to the object containing OceanRenderer script.
- If there is a message Max query count (4096) exceeded, ... create a new AnimatedWaveSettings file, set the Max Query Count to a larger value than default and assign it to OceanRenderer.

2020/07/16 15:41 · Aron Rescec

Lux Setup



Lux Water demo scene with DWP2.



Example Lux Water setup.

Lux Water supports water heights.

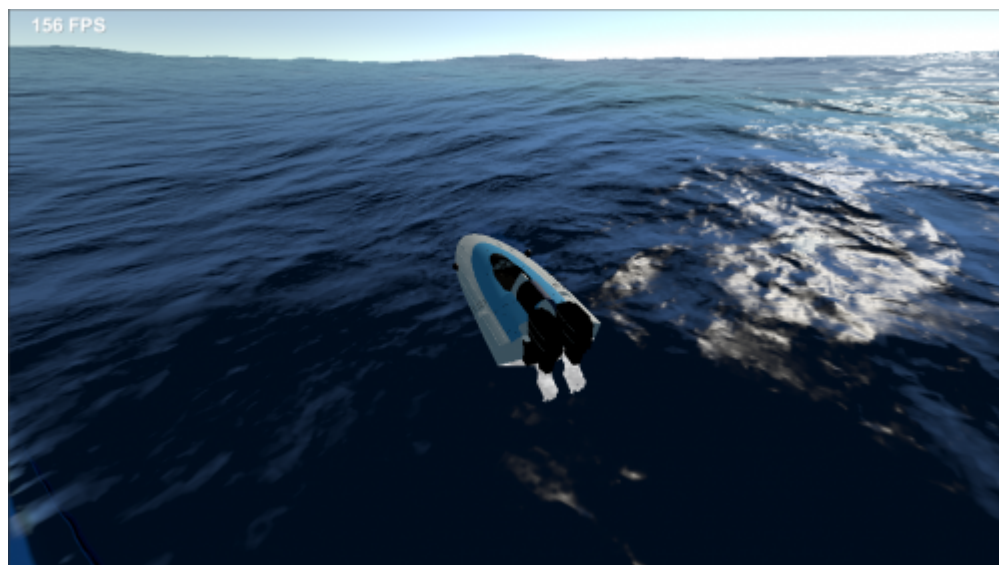
Setup

- Rename LuxWaterDataProvider.cs.txt to LuxWaterDataProvider.cs.
- Set up the scene as if using flat water, minus FlatWaterDataProvider.

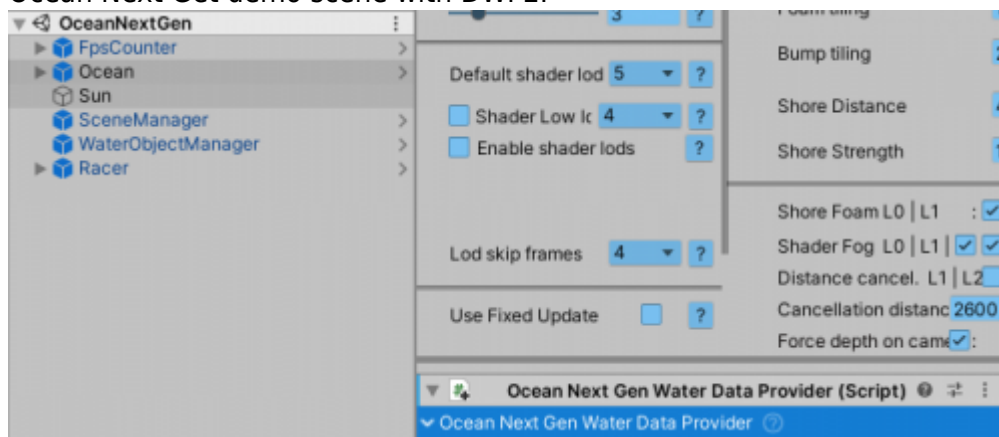
- Add LuxWaterDataProvider script to the object containing LuxWater_WaterVolume.

2020/07/16 15:41 · 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

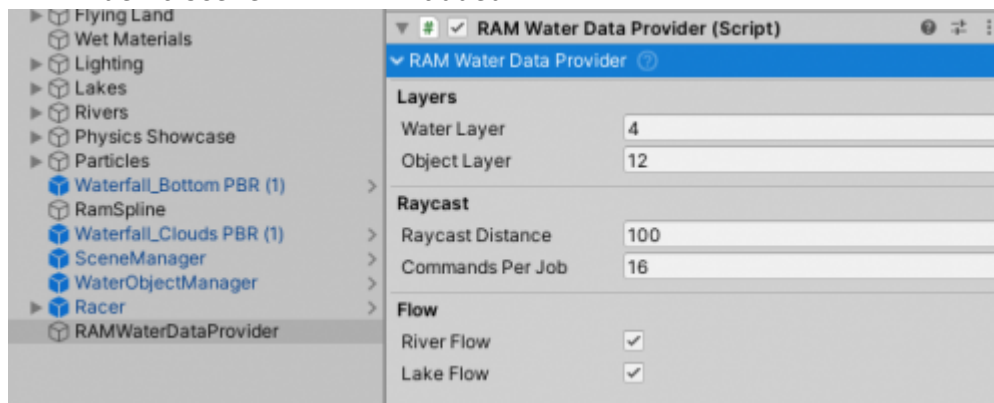
- Rename OceanNextGenWaterDataProvider.cs.txt to OceanNextGenWaterDataProvider.cs.
- Set up the scene as if using flat water, minus FlatWaterDataProvider.
- Add OceanNextGenWaterDataProvider to the object containing Ocean script.

2020/07/16 17:10 · Aron Rescec

River Auto Material (R.A.M.)



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



Example River Auto Material setup.

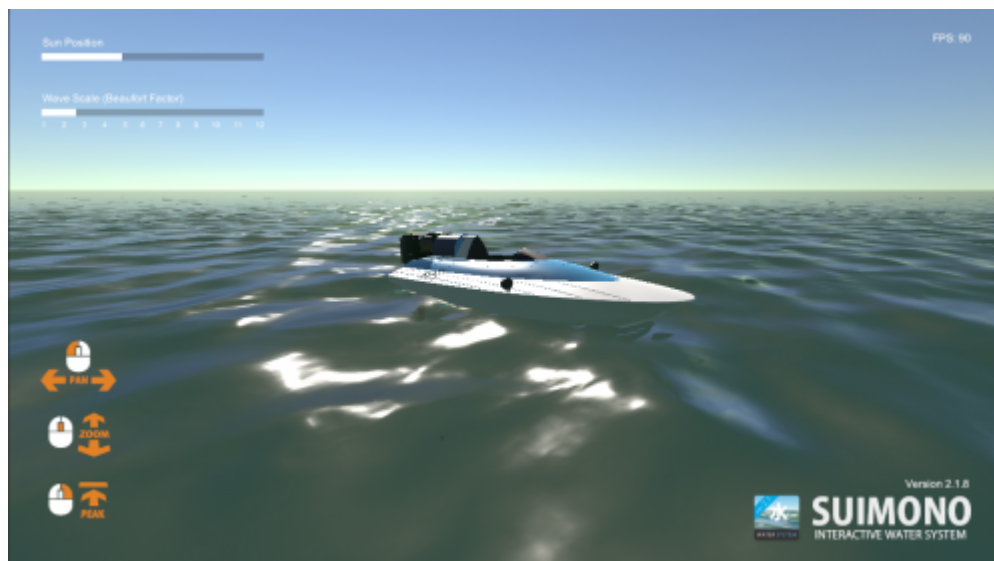
RAMWaterDataProvider supports water heights, normals and flow and it inherits from RaycastWaterDataProvider, meaning that the setup steps are the same.

Setup

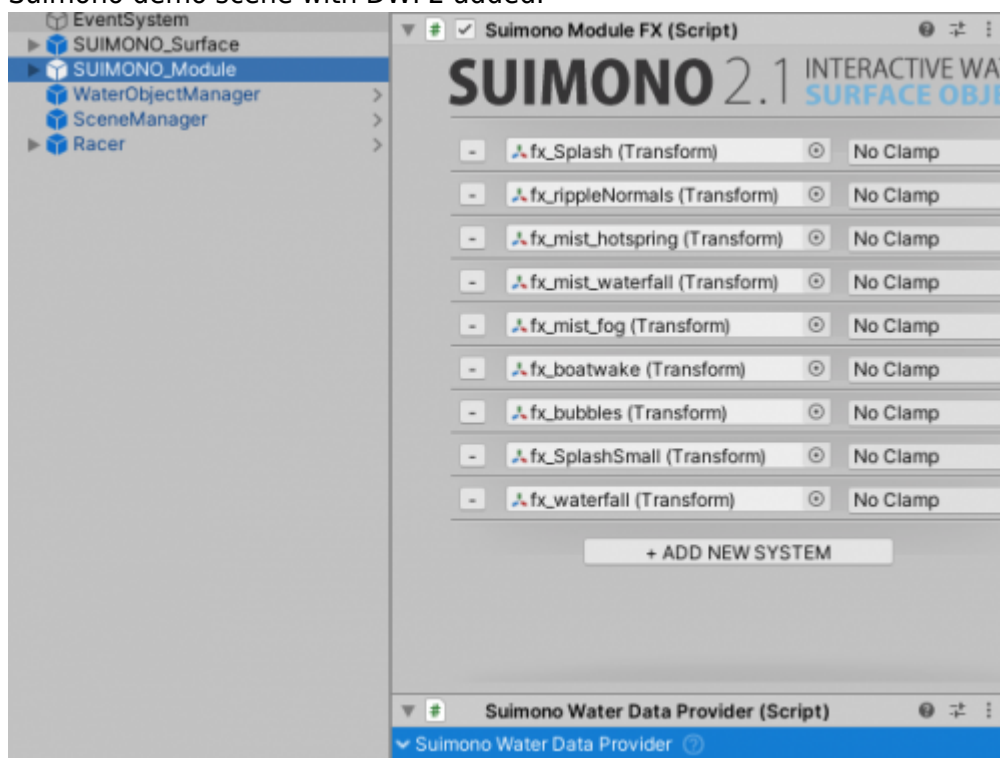
- Rename RAMWaterDataProvider.cs.txt to RAMWaterDataProvider.cs.
- Set up the scene as if using flat water, minus FlatWaterDataProvider.
- Add RAMWaterDataProvider to the scene. It does not have to be attached to any specific object.
- Assign Water Layer to the River Auto Material water. Assign Object 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 17:01 · Aron Rescec

Suimono



Suimono demo scene with DWP2 added.



Example Suimono setup.

Suimono supports water height queries only.

Setup

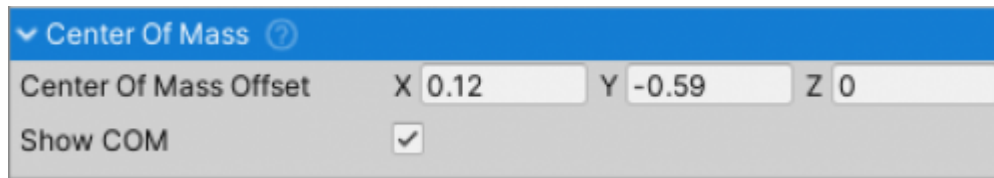
- Rename `SuimonoWaterDataProvider.cs.txt` to `SuimonoWaterDataProvider.cs`.
- Set up the scene as if using flat water, minus `FlatWaterDataProvider`.
- Add `SuimonoWaterDataProvider` to the object containing `SuimonoModule` script.

2020/07/16 16:48 · Aron Rescec

2020/04/26 15:53 · Aron Rescec

Helper Scripts

Center Of Mass

The image shows the 'Center Of Mass' inspector window. It has a blue header with a dropdown arrow and a help icon. Below the header, there are three input fields for 'Center Of Mass Offset': 'X' with value '0.12', 'Y' with value '-0.59', and 'Z' with value '0'. Below these is a 'Show COM' checkbox which is checked.

CenterOfMass inspector.

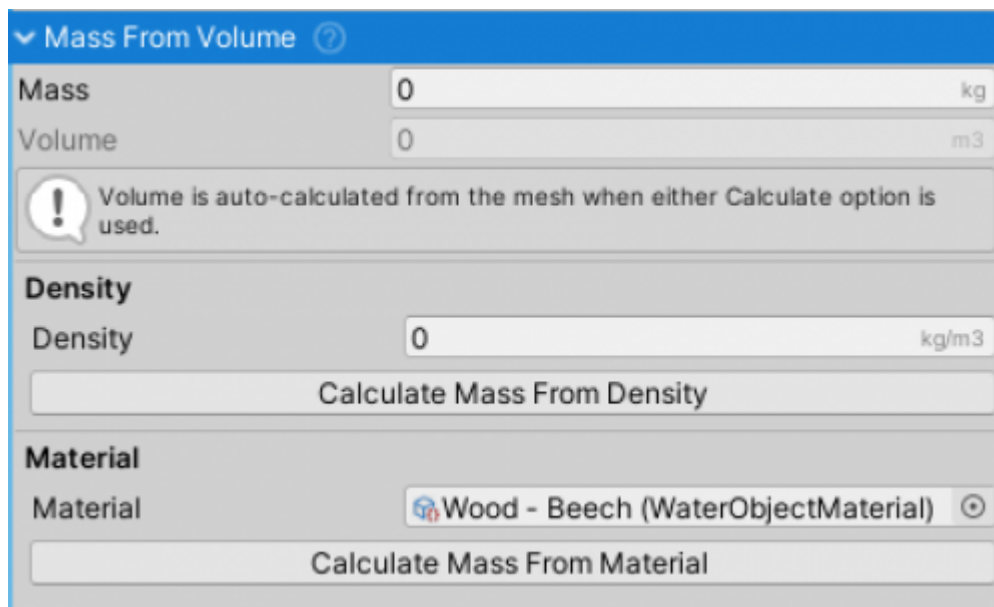
A helper script for setting center of mass of a **Rigidbody**.

Unity calculates center of mass as a center of volume of all the **Rigidbody's 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 21:24 · Aron Rescec

Mass From Volume

The image shows the 'Mass From Volume' inspector window. It has a blue header with a dropdown arrow and a help icon. Below the header, there are two input fields: 'Mass' with value '0' and unit 'kg', and 'Volume' with value '0' and unit 'm3'. Below these is a warning message: 'Volume is auto-calculated from the mesh when either Calculate option is used.' Below the warning, there is a 'Density' section with an input field for 'Density' with value '0' and unit 'kg/m3', and a button 'Calculate Mass From Density'. Below that is a 'Material' section with a dropdown menu for 'Material' showing 'Wood - Beech (WaterObjectMaterial)' and a button 'Calculate Mass From Material'.

MassFromVolume inspector.

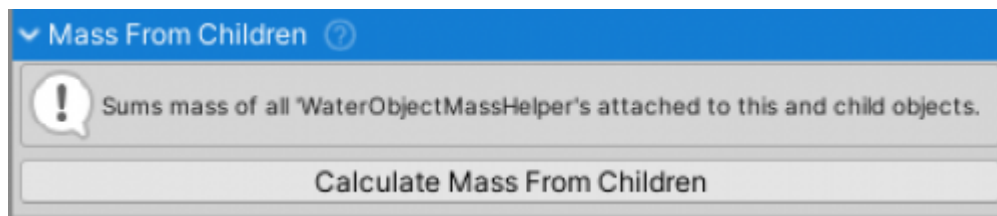
MassFromVolume is a helper script that calculates object's mass from volume of the mesh and the density.

- 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 23:42 · 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 21:32 · Aron Rescec

2020/04/28 15:15 · Aron Rescec

From:
<http://localhost/> - **Documentation**

Permanent link:
<http://localhost/doku.php/WaterObjectManual>

Last update: **2020/07/16 23:44**

