Terrain Engine 2D
User Manual - V1.20

INTRO ▾
GENERAL ▾
MAIN PROPERTIES ▾

Fluid

In depth information on the fluid simulation in the engine.

Table of Contents

The Fluid Simulation

The Basics

The fluid in Terrain Engine 2D is simulated using cellular automata. You can think of the Terrain Engine 2D world as a large grid, and in each grid space you can either have a block or some amount of fluid.

Terrain Engine 2D Fluid

Fluid Data

Similar to the terrain, each block of the world has information stored about its fluid properties. This data is stored inside the **FluidBlocks** array of the **FluidDynamics/AdvancedFluidDynamics** class. The main data stored for each block is listed below:

- Weight The amount of fluid in the block
- Stable Whether the fluid has settled, meaning no more fluid is flowing into or out of that block
- Color The color of the fluid (only in Advanced)
- Density The type of fluid (only in Advanced)

The Physics Algorithm

Fluid is simulated each frame by looping through the array of fluid blocks. Each fluid block compares the amount of fluid it currently has with those around it. If that block contains more fluid then those adjacent to it, a small amount of that fluid is transferred. In this way fluid is continually moved between adjacent blocks until a state of equilibrium is reached. This is how the algorithm works in its simplest form, however there are also other factors and complexities to consider.

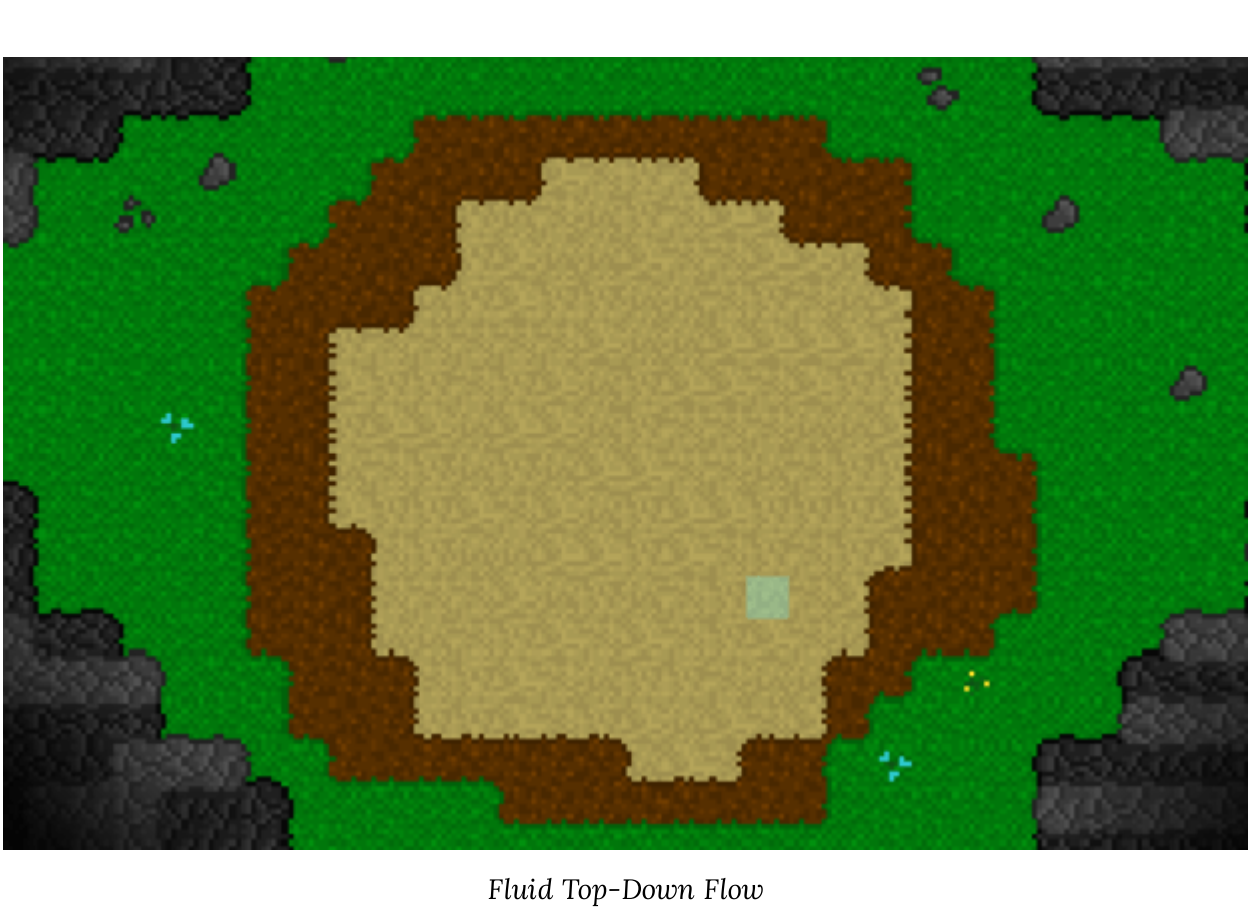
Fluid can not flow into grid spaces that contain blocks, which means blocks will inhibit the motion of fluid, and force it to flow in different paths.

Fluid blocks are allowed to hold more weight then the desired maximum. This creates **fluid pressure** and can influence the flow of fluid.

In the engine there are two different fluid physics functions, one for side-scrolling games called **Down Flow** where fluid flows down with gravity and one for top-down games called **Top-down Flow** where fluid flows equally in all directions.

Fluid Down Flow

Down Flow is the default fluid physics function where fluid will fall with gravity. This makes calculating the flow of fluid vertically quite more complex then in the top-down variation. In Down Flow fluid will continue to flow down until it reaches a solid (terrain) block or a space filled to the max. As stated before, fluid blocks can hold extra weight in a kind of pressurized state. This means that a lower fluid block can hold a small amount more fluid than the block above, so fluid pressure increases with depth. As a result fluid also has the ability to flow up with this pressure. When a block has more then the maximum amount of fluid (plus the extra amount allowed according to its depth) and can not flow horizontally it will be forced to flow up.



Fluid Top-Down Flow

Top-down Flow is much simpler then Down Flow since fluid can flow equally in all directions and thus the complexities of fluid pressure do not apply. When fluid is placed in a simulation using Top-down Flow the fluid will simply flow out in all directions until the fluid amount in adjacent blocks has equalized. However, fluid pressure can still play a role, as blocks can still become pressurized and hold more fluid then their maximum if it is added directly.

Basic Versus Advanced

There are two different fluid options which you can utilize depending on the kind of game you wish to create, your desired features and the target platform(s). The **Basic** fluid simulation and the **Advanced** fluid simulation. The Advanced fluid simulation was added to the engine in version 1.20, and with it came a bunch of new features. The main difference being the ability to have multiple different kinds of fluids as opposed to just one in the Basic fluid simulation.

Basic Fluid Simulation

The **Basic** fluid simulation has a single fluid type whose color is controlled by a **Main** and **Secondary** fluid color. The color of a fluid block is set by interpolating between those two colors. The greater the amount of fluid (or weight) the fluid block has, the closer the block will be to the Main color. In this way you can vary the fluid color depending on the fluid pressure.

Advanced Fluid Simulation

With the **Advanced** fluid simulation, you can have up to 256 different fluid types. Each fluid type has a **Default Color** and **Density**, fluid blocks of the same density are able to mix, where as fluid blocks of different density will remain separated. When two pools of fluid of the same type, but different color mix their colors will also mix resulting in a new color forming.

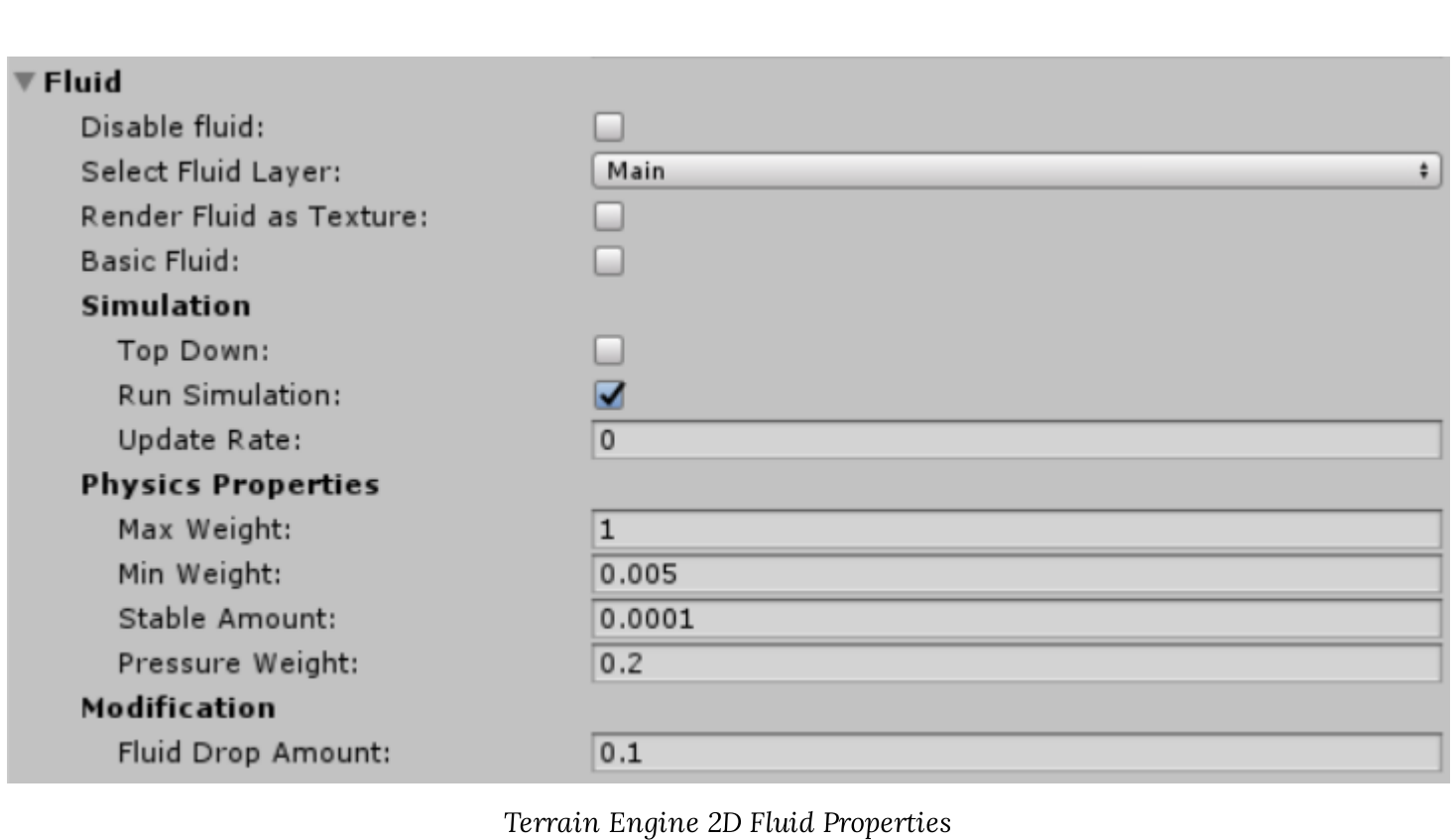
Below is a table showing the **Pros** and **Cons** of the **Advanced Fluid Simulation**:

Pros	Cons
Multiple fluid types	Uses more memory
Fluid color mixing	More CPU intensive
Fluid density separation	

These features are explained in greater depth in the **Fluid Properties** section below.

Fluid Properties

The **Fluid** properties section of the World Inspector gives you full control over the fluid simulation.



Terrain Engine 2D Fluid Properties

Base

- Disable Fluid Disable the entire fluid simulation, select this if you do not wish to have fluid in your game.
- Fluid Layer The terrain layer the fluid simulation will use to determine which blocks are solid (blocks fluid can not flow through)
- Render Fluid as Texture Render fluid using a single generated texture of the loaded terrain as opposed to in chunks
- Basic Fluid Opt to use the basic fluid simulation instead of the advanced fluid simulation

Select **Disable Fluid** to disable the simulation if you choose not to use fluid in your game, and this will increase performance. If you choose to have fluid in your game you must select the **Fluid Layer** which the engine will use for the fluid simulation. This is the layer in which the fluid will interact with the blocks of your terrain. You also have the option if you wish to **Render the Fluid as a Single Texture** instead of chunks of meshes. This makes it easier to apply shaders and perform post processing on the fluid rendering. So if you wish to animate the fluid in some way, create waves or change the look, you will likely want this option selected. The downside is in some cases it may be the slower option. You then have the option of selecting the **Basic Fluid** simulation instead of the advanced. The differences of which are explained above.

Simulation

- Top Down Fluid simulation used in a top-down style 2d game
- Run Simulation Toggle the fluid simulation (this will freeze any fluid in your game)
- Update Rate The rate at which the fluid simulation updates (in seconds)

The properties under the **Simulation** section allow you to modify how the simulation will run. You can set the fluid physics algorithm to use the **Top Down** function meaning fluid will flow equally in all directions ignoring gravity. You can toggle the entire fluid simulation through the **Run Simulation** field. The fluid **Update Rate** lets you control how often the simulation updates, which in turn will control the speed that the fluid flows. These properties are all accessible at runtime to help with testing.

Physics Properties

- Max Weight The maximum amount of liquid a single block can hold (unpressurized)
- Min Weight The minimum amount of liquid a single block can hold
- Stable Amount If the amount of fluid flowing out of a block is less than the stable amount, the fluid block is stable
- Pressure Weight Fluid weight pressure factor (each fluid block can hold Pressure Weight more liquid than the block above it)

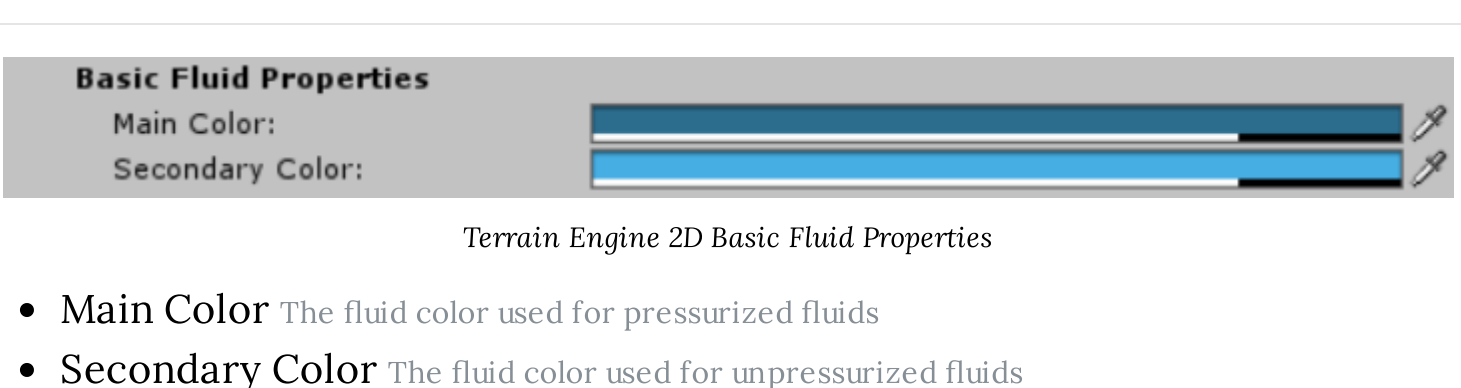
The **Physics Properties** section holds properties allowing you to change the fluid physics of the simulation. The **Max Weight** and **Min Weight** allow you to adjust the amount of fluid a block is able to hold. Changing the Max Weight will not effect the extra amount of pressurized fluid a block can hold. The Min Weight is used to stop fluid flow and is used as a threshold to determine when a block has approximately no fluid left (so the minimum weight should always be above but close to zero). The **Stable Amount** represents the threshold for determining fluid equilibrium. Since fluid is constantly flowing into and out of blocks to maintain equilibrium, this value is used to determine when the fluid amount flowing is small enough to stop the simulation from continuously performing calculations. The **Pressure Weight** is the amount of extra fluid a block can hold compared to that block above it. Note that fluid blocks only become pressurized when there is another block containing fluid directly above it. These properties are all accessible at runtime for the sake of testing. However you must be careful as changing these properties at runtime may cause unexpected results and could potentially break the simulation.

Modification

- Fluid Drop Amount Amount of fluid added on drop

The **Modification** sections holds a single property pertaining to modifying the fluid in game. The **Fluid Drop Amount** allows you to specify the amount (or Weight) of fluid added when placing fluid through the OSD or World Modifier script.

Basic Fluid Properties

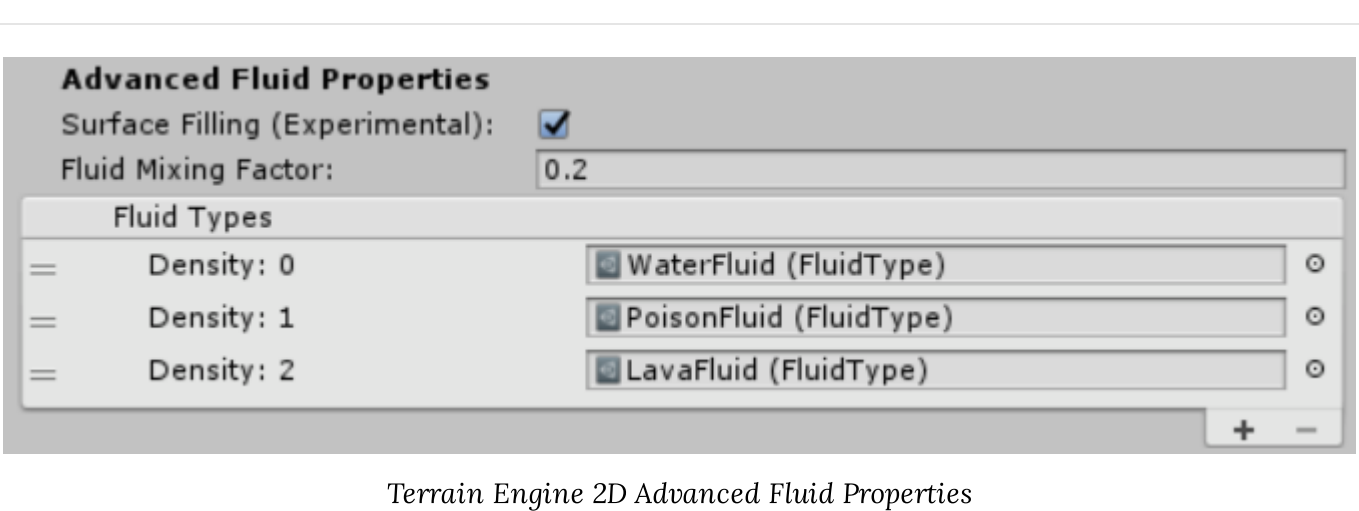


Terrain Engine 2D Basic Fluid Properties

- Main Color The fluid color used for pressurized fluids
- Secondary Color The fluid color used for unpressurized fluids

Under the **Basic Fluid Properties** section (which will only show up if Basic Fluid is enabled) you have the option to set the **Main Color** and **Secondary Color** of the fluid. The fluid color is set by interpolating between these two colors. Higher pressurized fluids will have a color closer to the Main Color where as fluid blocks with fluid less then the Max Amount will have a color closer to the Secondary Color.

Advanced Fluid Properties



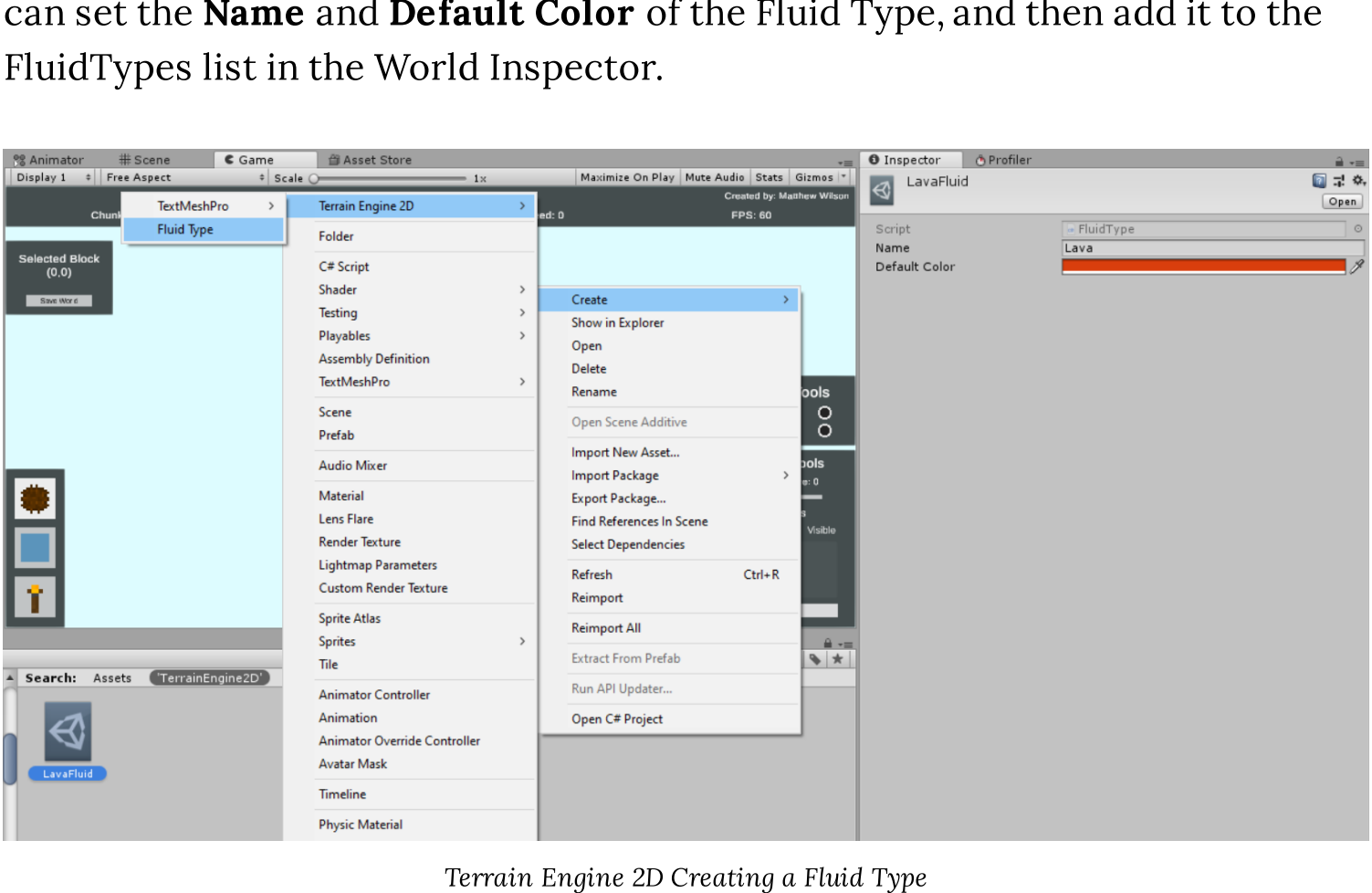
Terrain Engine 2D Advanced Fluid Properties

- Surface Filling (Experimental) Allows fluids of different densities to mix in order to fill the top surface level of a fluid
- Fluid Mixing FactorThe factor effecting how fluids mix together, a higher factor will result in a more drastic color change
- Fluid TypesA reorderable list of fluid types

The **Advanced Fluid Properties** section has options for setting up the Advanced fluid simulation. The **Surface Filling** option is an experimental feature which allows you to have fluids of different densities mix in order to fill the top surface level of a pool. This is to avoid having one fluid appear to float on top of another because the bottom fluid does not have its surface block level filled to the top. This works well with small pools of fluid, but not so well with large pools. There are some issues with this, because if the bottom pool is big enough, it may totally consume the other fluid. Which is why this is an experimental feature.

The **Fluid Mixing Factor** allows you to control how fluid of different colors (but the same density) mix their colors together. Fluid colors mix proportionally to the amounts of fluid in each block. The factor allows you to shift weighting between the color of the fluid flowing into a block, and the color of the fluid already in the block. A factor of 1 means fluid color will be based entirely on proportions. A factor less than 1 will place higher weighting on the fluid color already in the block. A factor greater than 1 will place greater weighting on the fluid color flowing into the block. A smaller Fluid Mixing Factor seems to yield the best looking results.

The **Fluid Type** list allows you to setup the different kinds of fluid which will be used in your game. These fluids can be referenced in your scripts by using their density value as index to the global **FluidType** array of the World Data Object. FluidType objects can be created in the **Create** menu, which can be accessed in either the top left of the **Project Window** or by right clicking in the Project Window. Select **Create -> Terrain Engine 2D -> Fluid Type**. You can set the **Name** and **Default Color** of the Fluid Type, and then add it to the FluidTypes list in the World Inspector.



Terrain Engine 2D Creating a Fluid Type

For information on how to use these Fluid Types in your terrain generation scripts refer to Terrain Generation. For examples on how to place these fluids in game refer to the **WorldInputHander** script.

Rendering Fluid

By default fluid is rendered the same as terrain, in chunks of meshes, however there is the option of rendering the fluid instead by generating a single texture. Both options have their pros and cons as described below.

Chunks

Rendering fluid in chunks is fast and efficient however since the fluid is ultimately split up into a number of different mesh pieces it can be hard to create animations or add cool shader effects to your fluid.

Texture

By rendering the fluid as a single texture it is much easier to manipulate the graphics of the fluid. You can apply shaders to the mesh and further manipulate the texture by applying post-processing effects. If you wish to apply your own shader to the generated fluid texture, the default material used by the Mesh Renderer is called **FluidTextured**. Any processing of the texture is handled in the **Fluid Renderer** script. The included **TopDownExample** project in the asset is an example of how you can create smooth looking fluid using this method, as seen below.

Fluid Texture Rendering

References

Special thanks to Janis Elsts for creating the algorithm for the fluid simulation, and Jon Gallant for converting it to C# in Unity. Both of these projects are licensed under the MIT software license. Links to these projects are included below:

Note that the fluid simulation algorithm from these projects has been heavily modified, updated and optimized for use in Terrain Engine 2D.

- https://w-shadow.com/blog/2009/09/01/simple-fluid-simulation/
- http://www.jgallant.com/2d-liquid-simulator-with-cellular-automaton-in-unity/