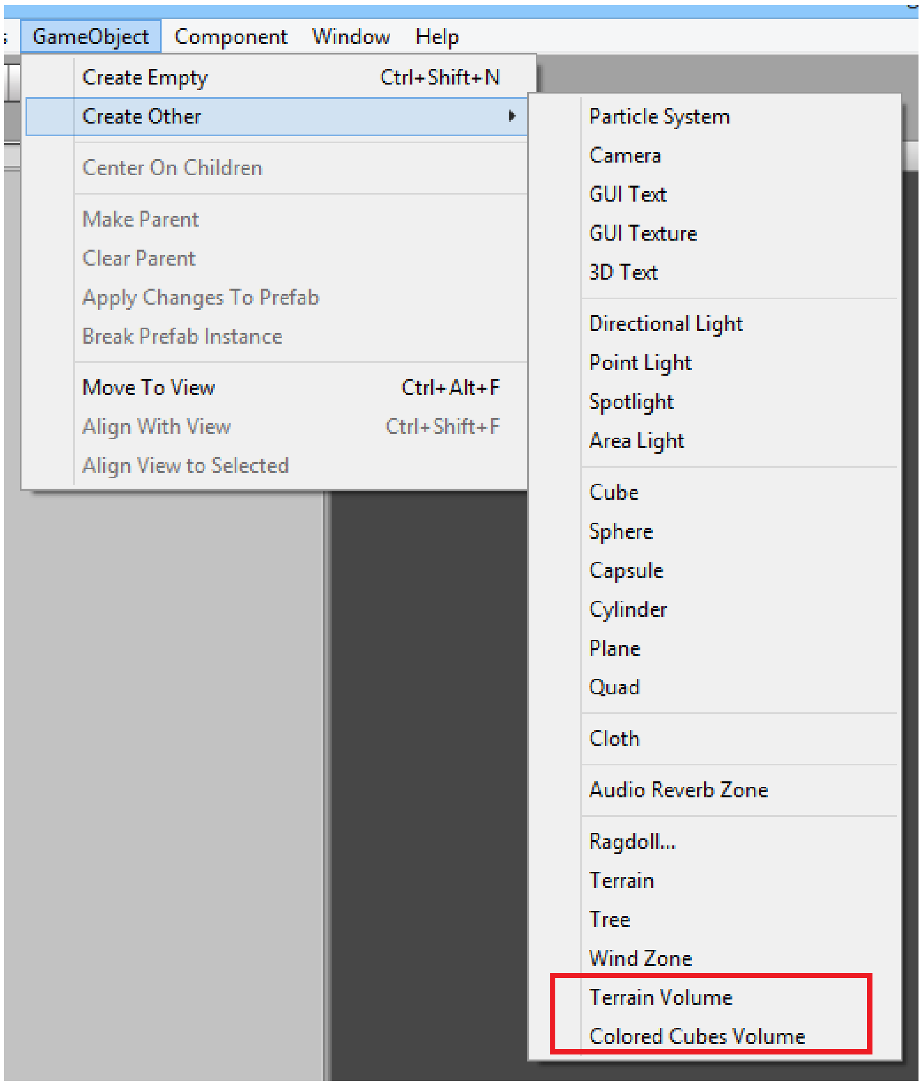
User Interface

Cubiquity for Unity3D is tightly coupled with the Unity3D editor and strives to follow similar design and usage approaches to those used in other parts of Unity. This means that concepts such as component-based design, assets, inspectors, transform gizmos, and materials should all work in similar way to what you are used to, and the interface to the TerrainVolume also takes significant cues from Unity's standard terrain.

We hope that you will therefore find the system intuitive, but none-the-less we use this section of the manual to outline the key user interface elements and how they fit together.

Adding Volumes To The Scene

A new terrain volume or colored cubes volume can be added to the scene via the main menu.

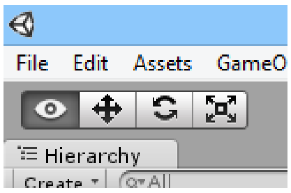


Creating a new volume through the main menu

This adds a new game object to the scene and automatically attaches Volume, Volume Renderer, and VolumeCollider components. It also creates a small volume data with just the floor filled in so that you have something from which to start your editing. Note that you are not given any options during the creation process (it is intended to be a simple one-click-and-you're-done' interface), but you can later create bigger/different volume data as discussed in the section on **Creating New Volume Data And Assets**.

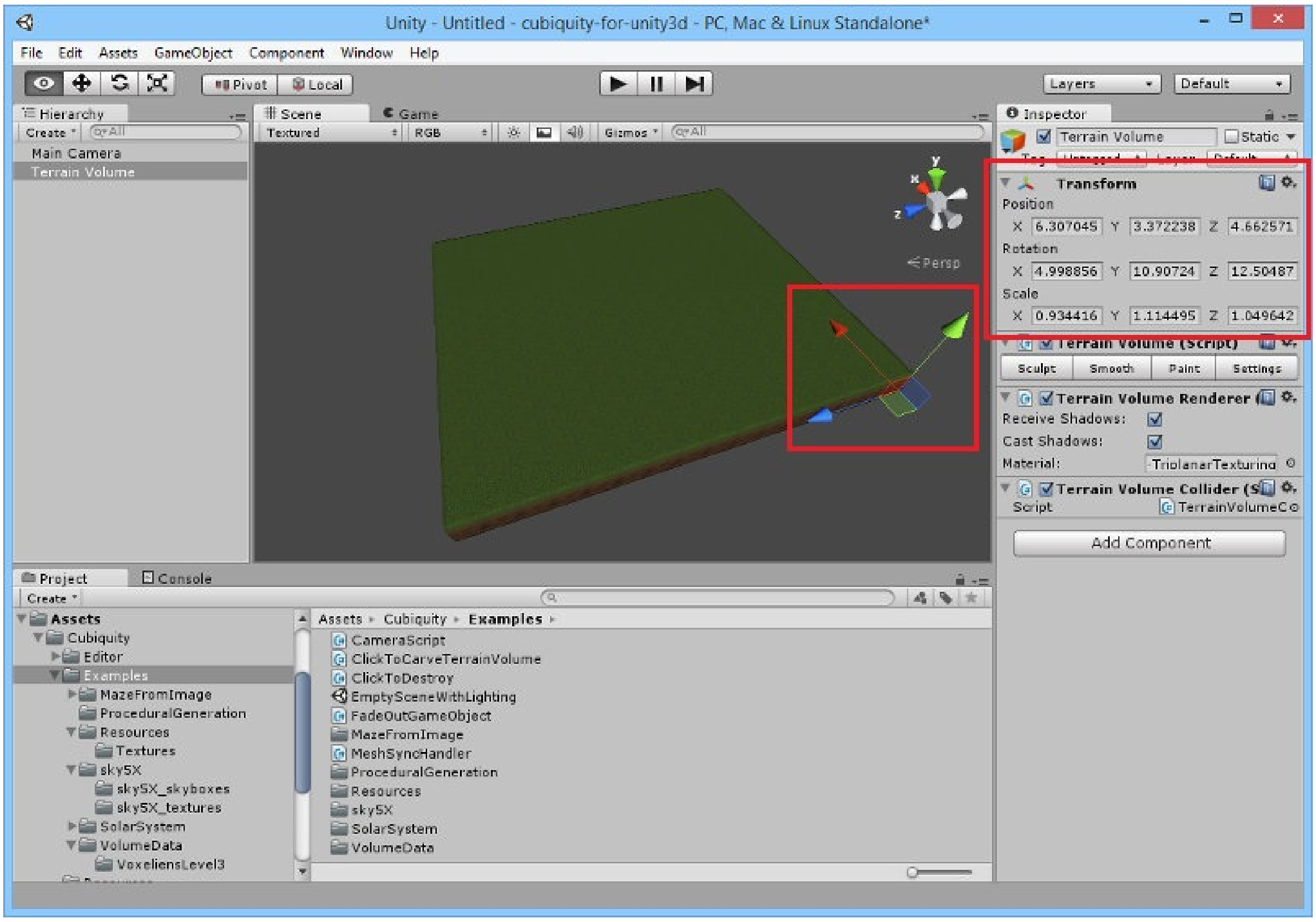
Transforming Volumes

Because a Cubiquity volume is just a GameObject with some specific components attached, it can be transformed much like any other object in Unity. Try selecting a volume and clicking one of the following buttons on the toolbar:



The standard controls can be used to translate, rotate and scale your volume

You should see the corresponding transform gizmo appear on the volume and you can use this to manipulate it. As you drag the mouse you will see the transform changing in the inspector, and you can also also enter the desired transform directly.



Position
X 6.307045 Y 3.372238 Z 4.662571
Rotation
X 4.998856 Y 10.90724 Z 12.50487
Scale
X 0.934416 Y 1.114495 Z 1.049642

Receive Shadows: ✓
Cast Shadows: ✓
Material: TriplanarTexturing
Script: TerrainVolumeCo

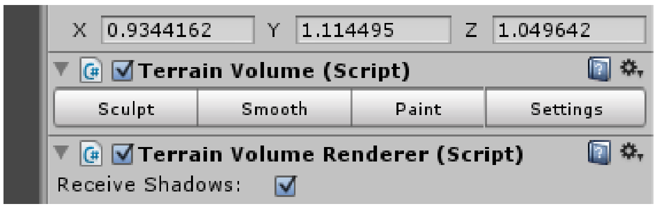
A volume after undergoing some transformations, with the transform gizmo and transform component highlighted.

Editing Volumes

We provide tools for editing the volumes from within the Unity editor (you can also edit the volumes in play mode but we don't provide tools for this - just use the API to implement your own game-specific tools).

Editing Terrain Volumes

When a terrain volume is selected you should see the 'Terrain Volume (Script)' component in the inspector.

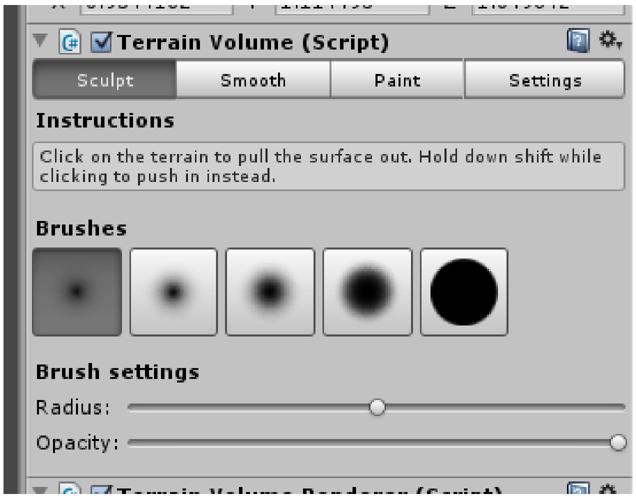


X 0.9344162 Y 1.114495 Z 1.049642
Receive Shadows: ✓

Click the tool buttons to open the corresponding options

You can cycle through the tabs to access the various functionality which is available:

Sculpt Mode



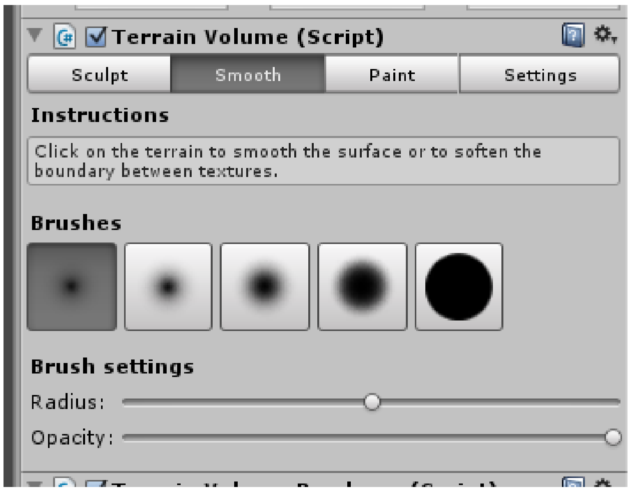
Click on the terrain to pull the surface out. Hold down shift while clicking to push in instead.

Sculpt mode options for changing the terrain shape.

This mode is used to make most changes to the shape of the terrain. You can choose the desired brush (gentle vs. sharp falloff) and set the brush's radius. The 'Opacity' setting controls how quickly the changes are applied, so having the slider to the left will allow finer control than having to the right.

Changes are applied by left-clicking on the desired part of the terrain. Note that changes are only applied when the mouse is moved, so you can't simply hold down the mouse on the same spot but must instead move it slightly. You can also hold down the Shift key when sculpting to remove material from the terrain instead of adding it.

Smooth Mode



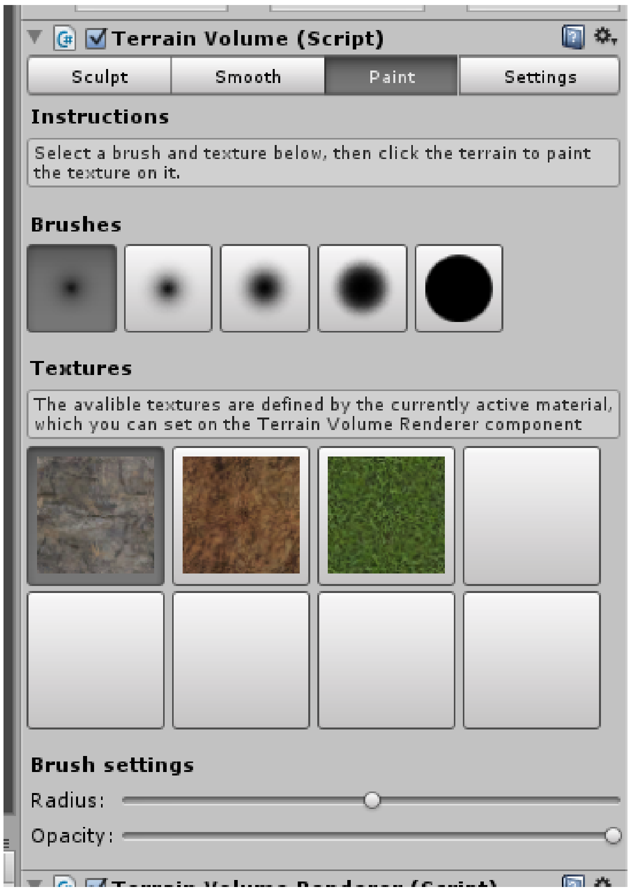
Click on the terrain to smooth the surface or to soften the boundary between textures.

Smooth mode options (currently the same as for sculpting)

The smooth mode is used to eliminate jagged edges and sharp features on the terrain, as well as softening the boundary between different textures. It does this by averaging together a voxel with it's neighbours - a kind of 3D equivalent to the 'blur' operation found in many 2D image editors. It is again possible to change the falloff, radius, and opacity of the brush performing the smoothing.

Paint Mode

Textures can be applied to the terrain by using Cubiquity's paint mode.


Terrain Volume (Script)

Sculpt | Smooth | Paint | Settings

Instructions

Select a brush and texture below, then click the terrain to paint the texture on it.

Brushes

Textures

The avalible textures are defined by the currently active material, which you can set on the Terrain Volume Renderer component

Brush settings

Radius:

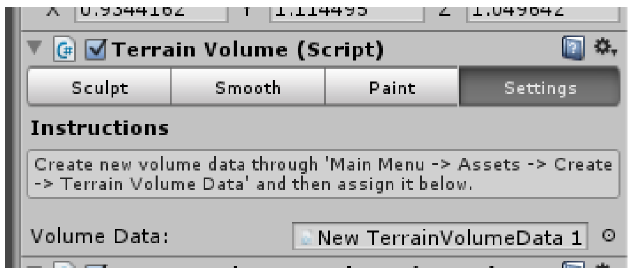
Opacity:

Paint mode options for applying textures to the terrain.

Note that the range of available textures is defined by the currently applied material, and this can be changed via **The Volume Renderer** as discussed later. The textures are not simply painted onto the mesh but are instead painted into the volume - i.e. the brush is not a circle but a sphere which also changes the texture of underground voxels. You can see the effect of this if you choose a large brush, paint part of the terrain, and then dig into the part which you have painted.

Settings Mode

Currently the only setting which is exposed is the volume data which is being rendered. You can drag new volume data onto this field, or click the small circle to select from a pop-up window. More settings will be exposed in future versions of Cubiquity.


Terrain Volume (Script)

Sculpt | Smooth | Paint | Settings

Instructions

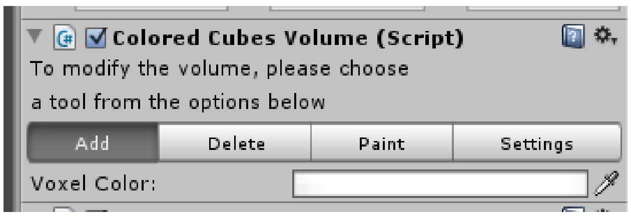
Create new volume data through 'Main Menu -> Assets -> Create -> Terrain Volume Data' and then assign it below.

Volume Data: New TerrainVolumeData 1

Settings currently lets you change the volume data which is being used.

Editing Colored Cubes Volumes

The built in editing capabilities of the colored cubes volume are currently very basic compared to those of the terrain volume, and it is expected that users will instead generate their volume in external applications or via other means (see **Obtaining Volume Data** for various options). However you can use them for making small adjustments.


Colored Cubes Volume (Script)

To modify the volume, please choose a tool from the options below

Add | Delete | Paint | Settings

Voxel Color:

The colored cubes volume inspector lets you add, delete, or paint cubes.

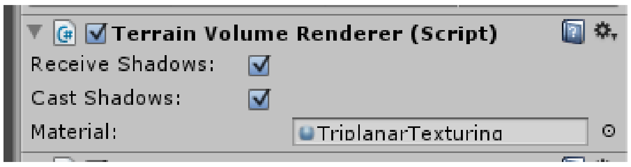
The Add, Delete and Paint options are all applied to a single cube at a time, with the Add and Paint offering the ability to chose a particular color for the voxel to be drawn. There is also a setting panel which, as in the case of Terrain Volumes, allows different volume data to be specified.

Other Volume Components

When creating a volume through the main menu Cubiquity for Unity3D ensures that certain other components are automatically added to the GameObject. These are optional though usually desirable, in particular the default added components provide the ability for the volume to be rendered and to participate in collisions.

The Volume Renderer

The VolumeRenderer component (actually one of its subclasses) needs to be added to any volume which you wish to be visible in edit mode or during gameplay. It serves a purpose which is conceptually similar to the standard Unity MeshRenderer, and exposes properties through the inspector shown below:

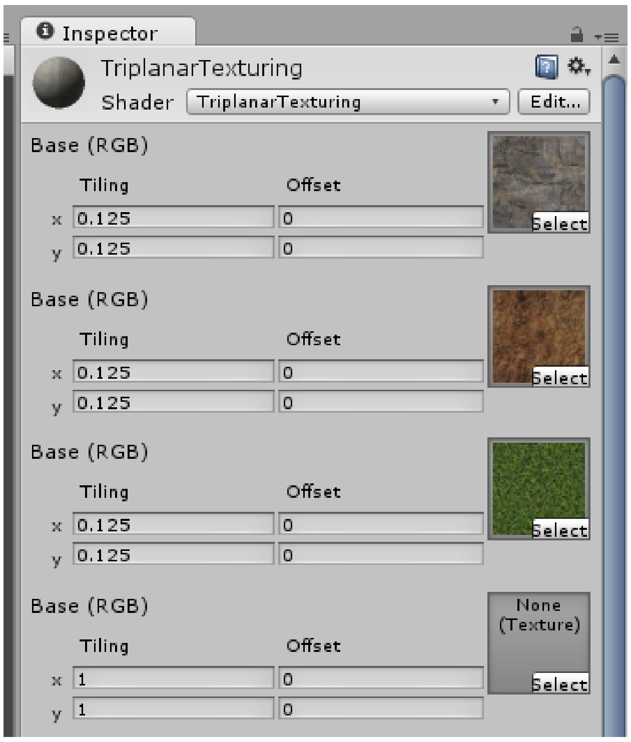


Terrain Volume Renderer (Script)
Receive Shadows: ✓
Cast Shadows: ✓
Material: TriplanarTexturing

Note that 'Material' only exists for the Terrain Volume Renderer and not the Colored CubesVolume Renderer.

The 'Cast Shadows' flag controls whether this volume get drawn into Unity's shadowmap (i.e. whether it can cast shadows on to other objects) while the 'Receive Shadows' flag controls whether the shadowmap gets sampled when rendering the volume (i.e. whether other object can cast shadows on to the volume). If you want the volume to exhibit self-shadowing then both of these flags need to be set, which is the default.

In the case of the *terrain volume* the 'Material' field lets the you specify which material should be used for drawing the terrain. The default material uses a technique known as 'triplanar texturing', though you can write and connect your own shaders and materials if you wish (strong graphics programming knowledge required). Even if you don't change the material you can also use this field to open the material so that you can specify which textures should be used. Double-click on the material to open it in the inspector as below:



Inspector
TriplanarTexturing
Shader TriplanarTexturing Edit...

Base (RGB)
Tiling Offset
x 0.125 0
y 0.125 0

Base (RGB)
Tiling Offset
x 0.125 0
y 0.125 0

Base (RGB)
Tiling Offset
x 0.125 0
y 0.125 0

Base (RGB) None (Texture)
Tiling Offset
x 1 0
y 1 0

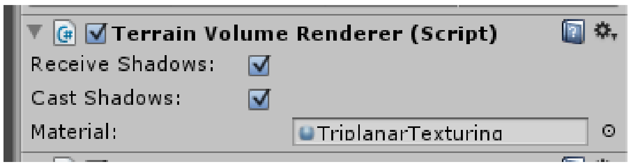
You can specify which images are linked to which texture slot.

From here you can change the images, offsets and scaling factors associated with the different texture slots (currently up to eight). You should see the rendered volume being updated in real-time as you make changes.

Note that it is not currently possible to change the material associated with a colored cubes volume renderer but this will probably change in the future.

The Volume Collider

The VolumeCollider should be added to a volume if you want it to be able to participate in collisions. This component currently does not expose any properties, but its presence causes a mesh collider to be created for the volume.



Terrain Volume Renderer (Script)

Receive Shadows: ✓

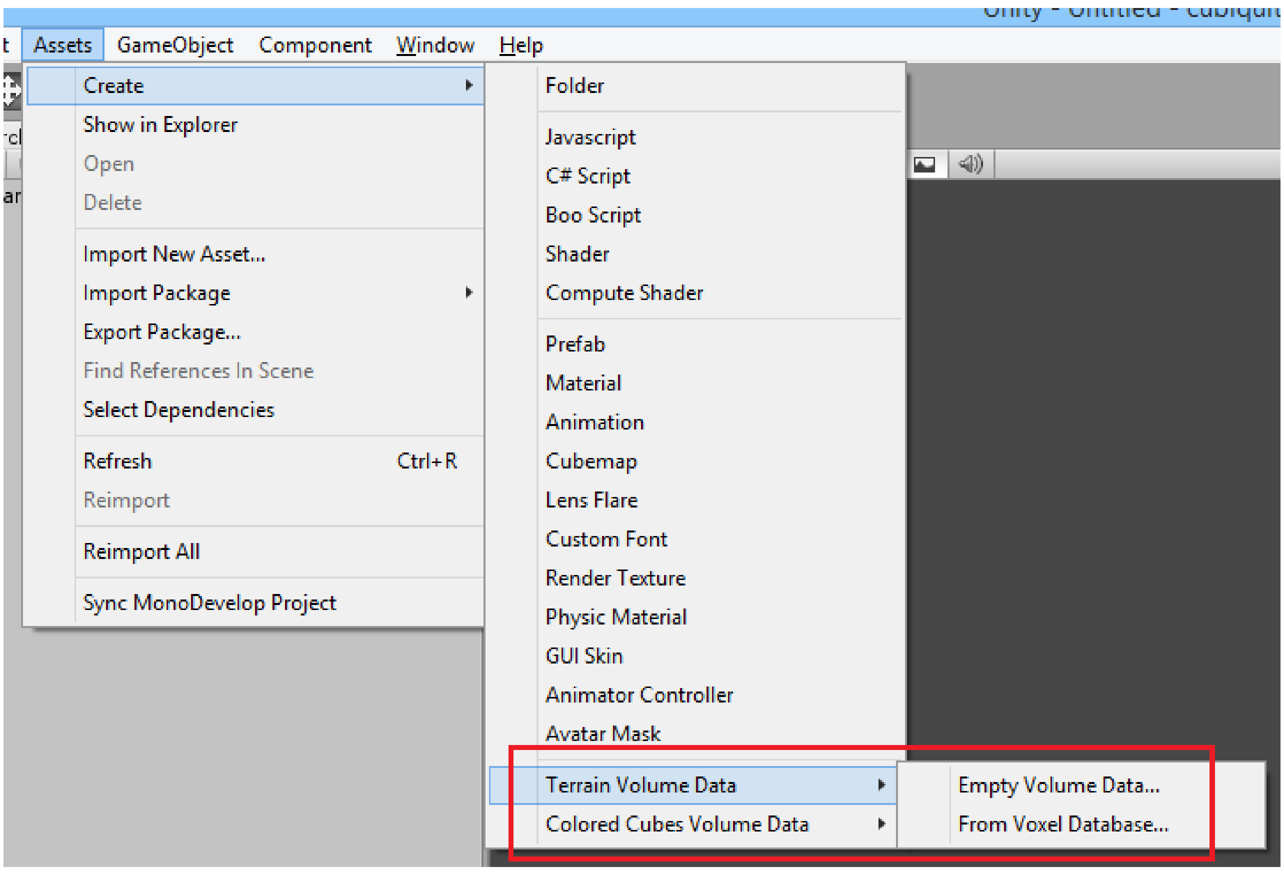
Cast Shadows: ✓

Material: TriplanarTexturing

The volume collider does not currently expose any properties.

Creating New Volume Data And Assets

When you create a volume through the main menu Cubiquity for Unity3D will automatically create an almost-empty volume data (with just a floor) to accompany it. It is quite possible you will want to replace this volume data with a larger empty one or to import a Cubiquity voxel database which you have generated elsewhere. Both of these can be achieved through the 'Assets' menu as shown below:

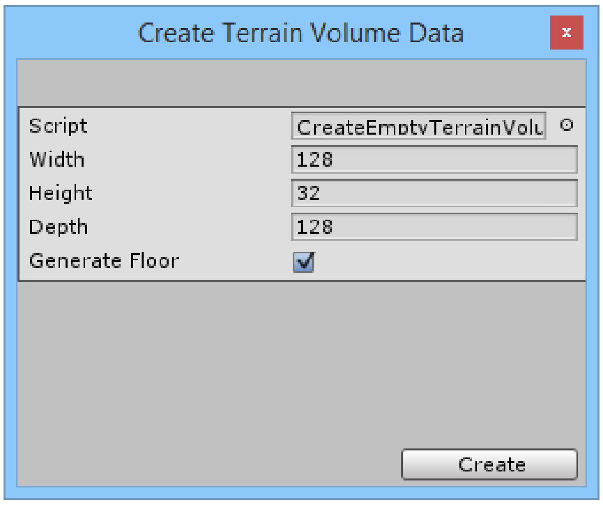


Note that there are separate entries for the two volume types (terrain vs. colored cubes).

For both terrain volumes and colored cubes volumes you have the option to create 'Empty Volume Data...' or 'From Voxel Database'. In both cases Cubiquity will also add the volume data to Unity's asset database, meaning an asset file will appear in your project. You can drag this on the the 'Volume Data' field in the 'Settings' tab of the volume inspector in order to set this as the active volume data for the volume.

Creating Empty Volume Data

Selecting this option will present a dialog similar to the one below:


Create Terrain Volume Data

Script	CreateEmptyTerrainVolu
Width	128
Height	32
Depth	128
Generate Floor	✓

You can configure the parameters of the new empty volume.

From here you can specify the size of your desired volume as well as whether it should include a floor. Note that the dimensions you specify should not be too large (probably 512 is the limit for a reasonable desktop PC). The floor flag controls whether the first few layers of the volume are filled solid - you almost certainly want this as otherwise you will not be able to see the volume and will have no starting point for further editing.

Creating From An Existing Voxel Database

If you already have a Cubiquity '.vdb' file then you can use this menu item to create a volume data which will reference it (see **Obtaining Volume Data** for ideas about how you might create an existing voxel database). You will be presented with a standard file selection dialog from which you can choose the .vdb file that you wish to wrap. Take care to ensure that the .vdb file you select contains a type of data (MaterialSet vs. QuantizedColors) matching the type of volume you are working with (terrain volume vs. colored cubes volume).