Terrain Engine 2D
A 2D Block Engine for Unity
Out now on the Unity Asset Store

BUY NOW!

Terrain Engine 2D
User Manual - V1.20

Serialization

This page explains how data is saved in the engine.

Table of Contents

How data is saved

Data is saved in two different forms, the world settings which includes the Main Properties and Block Setup information is serialized as a Scriptable Object called World Data. The main Terrain properties (Name, Width, height, Seed) from the World custom inspector is saved in it's own seperate json file reffered to as the Base Data. The terrain data arrays holding all the data used to serialize the in game world, are saved in a binary file reffered to as the Terrain Data.

What data is saved

World Data

The World Data includes all of properties and fields that can be modified from the World custom inspector, including the Block Type and Layer information. The WorldData script is a ScriptableObject which serializes the data in the inspector. This is useful for many reasons. For one it allows you to have all the settings stored in one place for easy access from all your scripts. It can be accessed from any script using:

```
TerrainEngine2D.World.WorldData;
```

It makes it easy to export your Terrain Engine 2D projects, as well as keep your data safe during updates. It also allows you to keep multiple World Data objects, which could be useful for testing different block textures for example. Another bonus of doing this is it allows you to save modified settings during playmode.

Terrain Engine 2D Inspector World Data Object

World Data Objects can be generated from the World Custom Inspector.

Base Data

The Base Data is one of the two types of data which is saved to file when your world is saved. It is the most basic information about your world, it contains the **Name**, **Width**, **Height** and **Seed**. The kind of information you might want to show in a menu screen for example.

*The Base Data is saved as a **.json** file*

Terrain Data

The Terrain Data is the other type of data saved to file when you world is saved. It contains all of the block and fluid data for every single grid position in your generated world.

The block data stored to file includes the **BlockType** and **RenderBlock** information. This is the only data necessary to save as all the other information can be generated at program start. This does mean that the terrain variation will not be saved, and any changes to that at runtime will be lost. If you wish to save more data, it is easy to expand upon the serialization system. Refer to the BlockData and FluidData scripts, as they are good examples for how that can be done.

The fluid data stored to file depends on whether the basic or advanced fluid simulation is being used. If the basic fluid simulation is being used then only the fluid **Weights** are serialized. If the advanced fluid simulation is used, then the fluid **Density** and **Color** data is also serialized.

*The Terrain Data is saved as a **.bin** file*

Where data is saved

The World Data Object is saved directly to your main project folder, as it is created within the Unity editor.

There are two seperate places where the Base and Terrain data are saved depending on if the game is run from the editor or in a build.

In Editor

If the game is run from the editor and the game is saved, then the default save location is:

```
public static string EditorSaveLocation =
Application.streamingAssetsPath/Worlds;
```

'Application.streamingAssetsPath' will create a folder called StreamingAssets in the root of your main project folder. This allows you to access all of your saved worlds from one convenient place when testing, and makes sharing projects easier.

In Build

If the game is run from a build and the game is saved, then the default save location is:

```
public static string DefaultSaveLocation =
Application.persistentDataPath/Worlds;
```

'Application.persistentDataPath' is going to be different for everyone, for more information you can refer to Unity's documentation.

WARNING be wary that any time you make changes to the settings of your world, any previously saved worlds may no longer work!

Saving Worlds

In Editor

In the editor your world can be saved manually and automatically. If you enable **Auto Save** in the World inspector, then your world will be saved automatically when you exit playmode. Your world will also be saved if you manually select **Generate World** in the World inspector. Your world can also be manually saved during runtime from the OSD.

From Script

Saving your world from script is super easy, and can be called from anywhere using:

```
TerrainEngine2D.Serialization.Save();
```

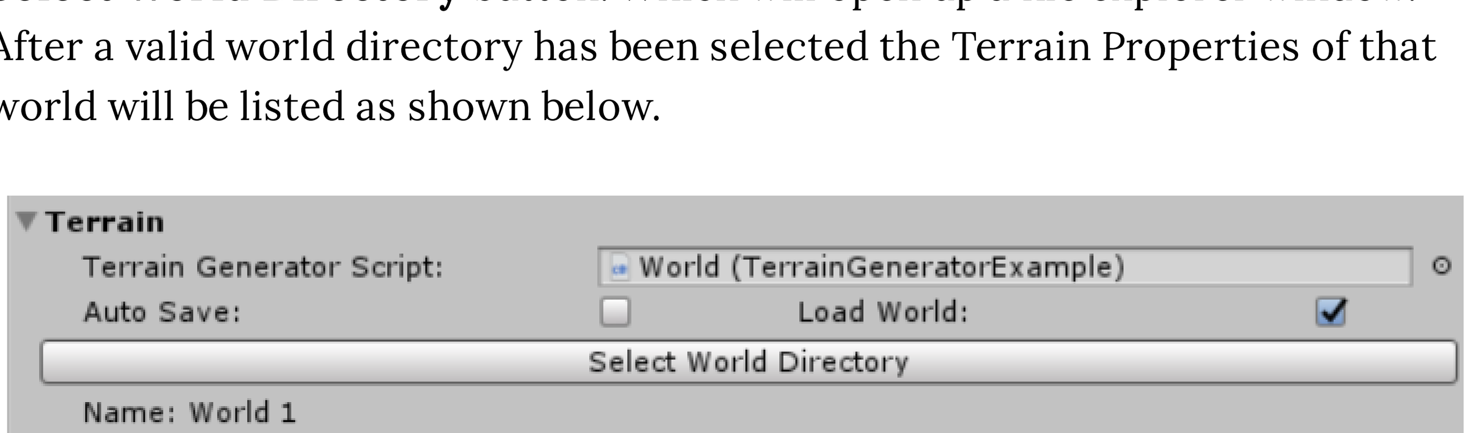
Calling that function will create a directory, then serialize the Terrain Data and Base Data to file at the appropriate location.

Loading Worlds

In Editor

Note that you can not load saved world files that were generated and saved with a different World Data object.

From the editor you can load a world in the World inspector by checking the **Load World** option found near the top of the inspector under **Terrain**. Then you must locate the directory of the world you wish to load by hitting the **Select World Directory** button. Which will open up a file explorer window. After a valid world directory has been selected the Terrain Properties of that world will be listed as shown below.



Terrain Engine 2D Terrain Properties

From Script

The process for loading a world from script is a little more complicated then saving. An example scene in provided in the asset called **MenuExample**, you can refer to that scene or particularly the **GameManager** scripts for how this is done.

If you choose to use the MenuExample ensure that you properly setup the BuildSettings for switching scenes detailed in Setup.

Note that every World Data Object has a GUID which will match the SID of the BaseData file in the directory of any world generated and saved with that World Data Object.

The SID value from a BaseData file can be attained using `Serialization.GetSID` from the path to the save directory.

This can be used to ensure you are not loading world save files from other TE2D projects.

The easiest way to load a world from script is to modify the **WorldData** properties; **LoadWorld** and **WorldDirectory**. You must set LoadWorld to true and set the WorldDirectory to the path of the world you wish to load. Then the next time you reload the scene your new world will be loaded. This can be done from your current scene or any other scene.