

Terrain Serializer Document

Terrain Serializer is a lightweight script that lets you Save/Load Unity's Terrain at runtime.

You can use it to save few hours of work/ headache, or to learn how to serialize and interact with terrain data. The codes are kept simple and can be easily modified/extended/integrated.

Features

- * Save/Load Tree Instances
- * Save/Load HeightMap
- * Save/Load AlphaMap
- * Save/Load DetailMap
- * Save/Load Terrain (All the maps/tree data)
- * Serialize/Deserialize at both runtime/editor
- * Source code included.

Limitation

Terrain Serializer does not serialize meshes(tree etc), textures(grounds etc) etc. That means, for example, if sand ground texture is replaced by mud texture on the terrain editor, mud will appear in place of sand when the terrain is deserialized.

Terrain Serializer does not create new Terrain. It applies serialized data(maps, trees and detail positions,size, etc) to the existing terrain.

It can completely restore a terrain given that the terrain has the same set of brushes(trees, details etc)

Example Usage

To serialize Terrain, call
`TerrainSerializer.Save(filePath, terrain);`

To deserialize Terrain, call
`TerrainSerializer.Load(filePath, terrain);`

Each map/tree data can be individually serialized/deserialized if desired.
For more methods and information, please refer to the documentation.

Support

Please feel free to ask questions or request new features.

Quick Start

Make a plain terrain, set textures, trees, details and save it to a prefab.

Upon loading a zone/scene in a game, Instantiate a prefab and call Load()

Upon saving a game, call Save()

TerrainSerializer class API and Examples

```
public static void Save(string path, Terrain terrain)
```

```
public static void Load(string path, Terrain terrain)
```

Saves the specified terrain's data, or loads the terrain data and applies to the specified terrain. You can either use file name or directory name as a path. It will throw an exception if the directory was not found.

Example:

```
TerrainSerializer.Save(Application.streamingAssetsPath+"/myData", myTerrain);
```

For each saved data, a suffix is automatically added to the provided path. In the above Case, heightMap data will be saved as myTerrainHeight, trees will be saved as myTerrainTree etc etc.

Calling the method will serialize/deserialize all the terrain data so it is not necessary to call other methods introduced below.

```
public static void SaveSettings(string path, Terrain terrain)
```

```
public static void LoadSettings(string path, Terrain terrain)
```

Saves/Loads terrain settings such as its position and size. In most case, saving and loading these data are optional. You can extend the method to serialize data your game require such as wind settings.

Example:

```
TerrainSerializer.SaveSettings(Application.streamingAssetsPath+"/myData", myTerrain);
```

```
public static void SaveTrees(string path, Terrain terrain, bool saveDetails = true)
```

```
public static void LoadTrees(string path, Terrain terrain)
```

Saves/Loads tree instances. By default, TerrainSerializer will serialize each tree's details such as scalings and colors. But if you have many trees, you can specify saveDetails=false when calling the method to strip the details. (Thus reducing file size and improving performance)

Example:

```
TerrainSerializer.SaveTrees(Application.streamingAssetsPath+"/myData", myTerrain);
```

```
public static void SaveAlphaMap(string path, Terrain terrain)
```

```
public static void LoadAlphaMap(string path, Terrain terrain)
```

Saves/Loads AlphaMap(splat maps for your ground textures).

Example:

```
TerrainSerializer.SaveAlphaMap(Application.streamingAssetsPath+"/myData", myTerrain);
```

```
public static void SaveHeightMap(string path, Terrain terrain)
```

```
public static void LoadHeightMap(string path, Terrain terrain)
```

Saves/Loads HeightMap.

Example:

```
TerrainSerializer.SaveHeightMap(Application.streamingAssetsPath+"/myData", myTerrain);
```

```
public static void SaveDetailMap(string path, Terrain terrain)
```

```
public static void LoadDetailMap(string path, Terrain terrain)
```

Saves/Loads DetailMap.

Example:

```
TerrainSerializer.SaveDetailMap(Application.streamingAssetsPath+"/myData", myTerrain);
```

Known Issues and Note

None right now. Please let us know if you found/encounter any problem.

Performance and Tips

While serializing 512x512 terrain takes about 0.3 - 0.5 sec on my desktop PC, 4096x4096 terrain takes about 1.5 - 2 secs. Serializing Unity Terrain is certainly not a light task.

We have tried other serialization methods like JSON.NET but the results were slower.

Using google's Protocol Buffers may dramatically improve the performance and may be added as a new feature if there are many requests.

Also, adding more Detailed brushes or increasing Detail Resolution affects the performance. Having high detail resolution will also generate more drawcalls so it is recommended to keep the resolution low and instead increase the Detail Resolution Per Patch.

