3D Tile Terrain Generator

By Lucas Fierfort

User manual

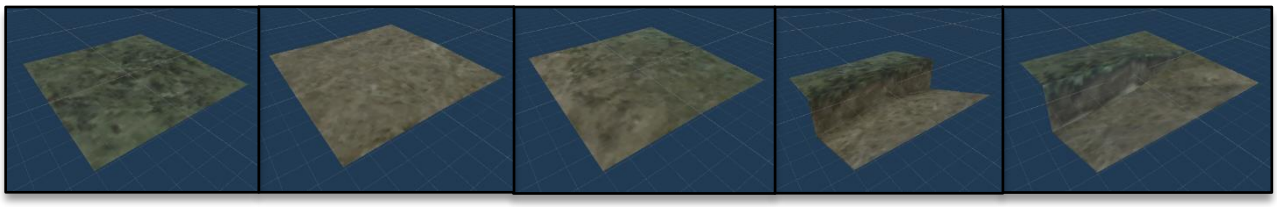
Introduction

Square Tile Terrain Generator is a simple tool for Unity providing user a mean to generate terrain (following a grid pattern) from single prefabs, with different features such as tile type/orientation handling, height management and tree generation.

The goal of this tool is to recreate low-poly/PSX squary style terrains with the simplest code as possible.

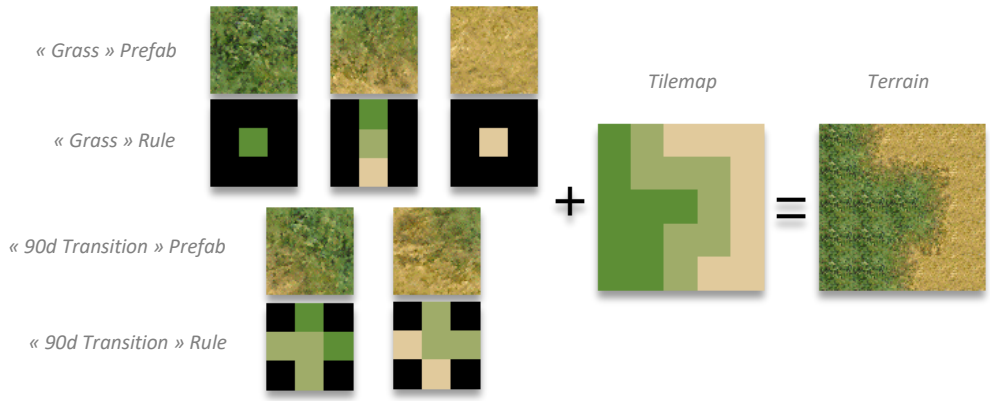
Features

User can use his own prefab terrain tiles to fill the terrain.

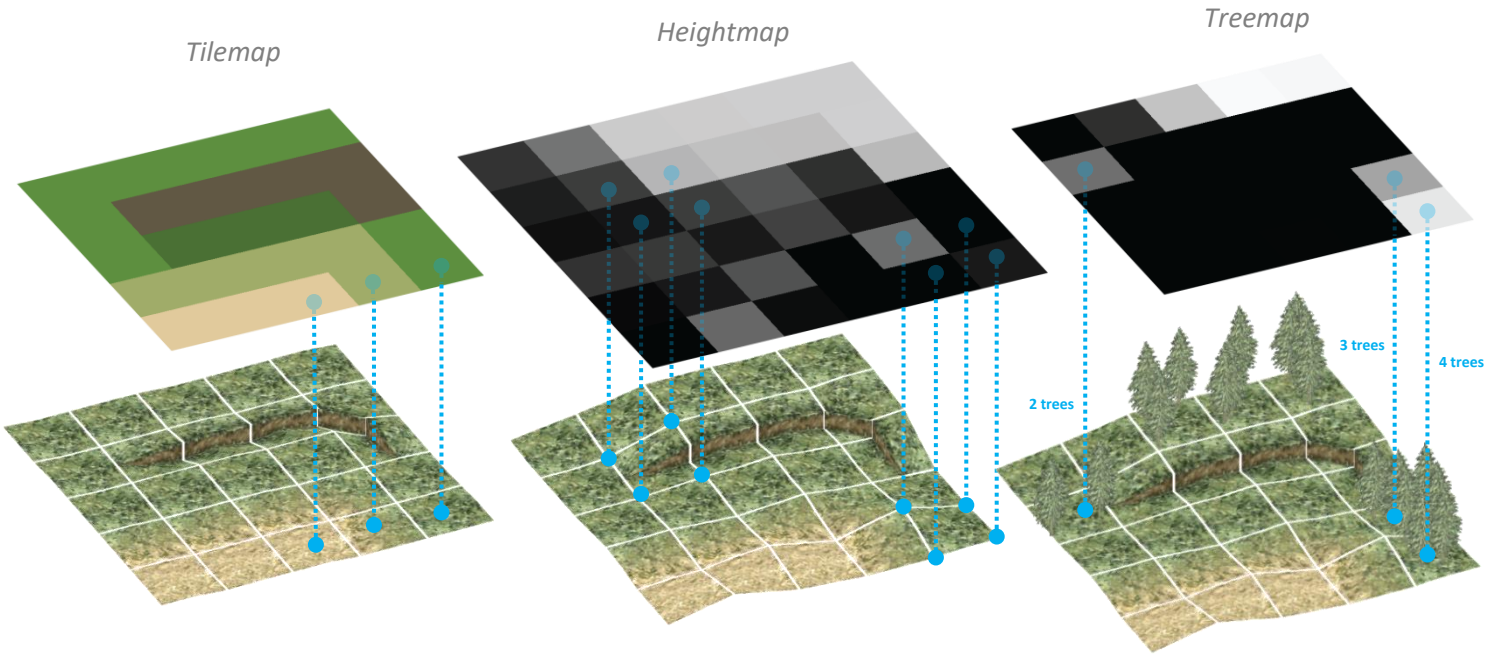


Examples of prefabs

By associating 3x3 images with his prefabs, user can create a set of “rules” determining tile arrangement and orientation one to another :

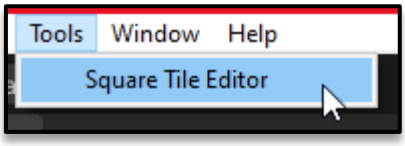


In addition of a tilemap, a heightmap and a treemap allow user to set respectively different heights to each prefab corner and add tree prefabs in various density.

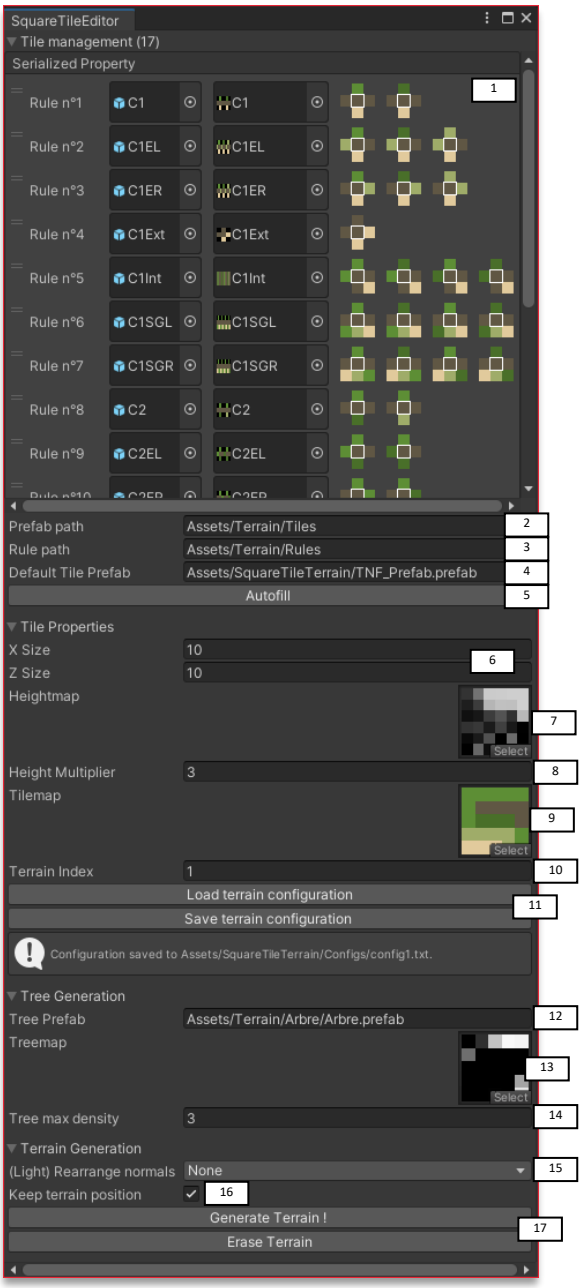


Setup

The tool can be found in the *Tools* tab.



The parameters are the following:

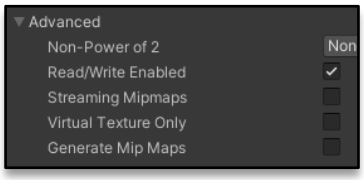


1. **Rule list**: Associate your tile prefabs with images representing rules for sorting this one over another prefab. Note that the n° of the rule also represent their priority when script picks adequate prefab during terrain generation. Rules previews are also visible on the right side for better visibility.
2. **Prefab path**: Folder containing tile prefabs.
3. **Rule path**: Folder containing rule textures.
4. **Default tile prefab**: Default prefab, picked when no rule can be applied over a pixel of tilemap.
5. **Autofill**: Automatically fills rule list using prefab and rule directory mentioned above. Each prefab and rule are associated by alphabetical order.
6. **Tile size**: Tile prefab width and depth. Determines spacing between each tile.
7. **Heightmap**: Map representing height of each grid intersection (or each prefab corner). Heightmap must be one pixel higher and wider than Tilemap to cover every corner.
8. **Height multiplier**: The intensity of heightmap.
9. **Tilemap**: Map representing tile placement according to their colors and their neighbor's color.
10. **Terrain index**: Number given to your terrain. Allows to handle multiple terrains.
11. **Load/Save terrain configuration**: Retrieve or store current parameters. Each field value will be stored in *SquareTileTerrain/Configs* folder.
12. **Tree prefab**: tree prefab that will be placed all over the terrain according to treemap.
13. **Treemap**: Map representing tree placement over the terrain. Black means no tree, white means full tree quantity.
14. **Tree max density**: Maximum amount of trees on the same tile, obtained with a white pixel in treemap.
15. **Rearrange normal**: As tiles are deformed to fit heightmap, normals (vectors on each vertex used to determine lighting behaviour) can be deformed. Set all vector to a cardinal vector (up, right or forward) if you wish.
16. **Keep terrain position**: If terrain is generated while already existing, its current position, rotation and scale will be kept. Otherwise, terrain will go back to 0,0,0.
17. **Generate/Erase terrain**: As the name suggests!

Notes

Mandatory settings

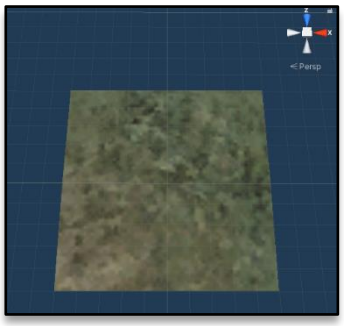
1. For the terrain to be generated, several conditions must be reunited :
- Tilemap (9) must be given.
- Tile rule list (1) must not be empty.
- If treemap (13) is given, tree prefab path (12) must be valid.
- If default tile path (4) is not empty, it must be valid.
2. Tilemap, heightmap, treemap and rules must have "*Read/Write*" setting enabled.

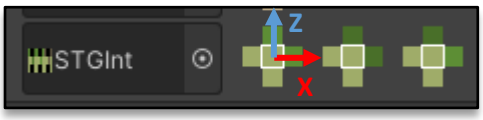


Rules

A single prefab can contain multiple rules. Associated rule image must be 3 pixels high and multiple of 3 wide. Note that black pixel means that corresponding neighbor is not important.

Rule must be oriented as left-right represent de global X axis and up-down represents global Z axis of prefab.

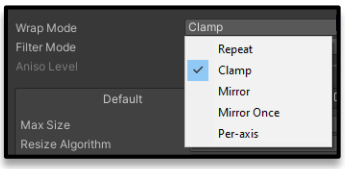




Heightmap

As a tilemap and treemap pixel represent one tile, a pixel in the heightmap represent a corner. It must be one pixel wider and higher than tilemap/treemap then.

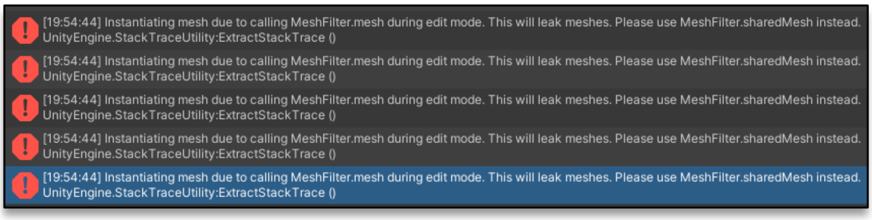
If not, no error will be raised, as Unity uses wrap to extrapolate image beyond its existing size.



Known issues

Mesh and Shared Mesh

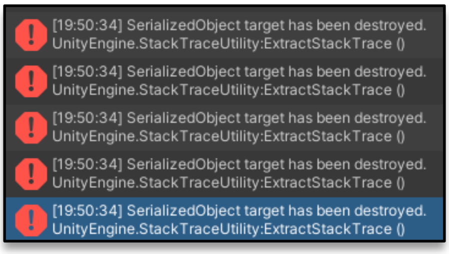
After generating the terrain, Unity will pop several error messages stating that meshes are deformed using the wrong mesh.



No solution was found to prevent those errors. However, it doesn't affect terrain generation. You can ignore them.

SerializedObject target has been destroyed

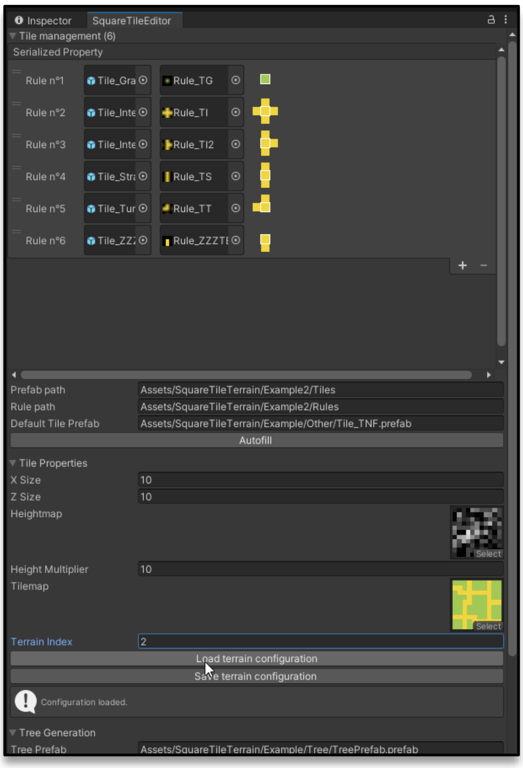
When the Tile Generator is destroyed (after a script compilation, exiting game etc.), Unity will warn user about serialized list being destroyed.

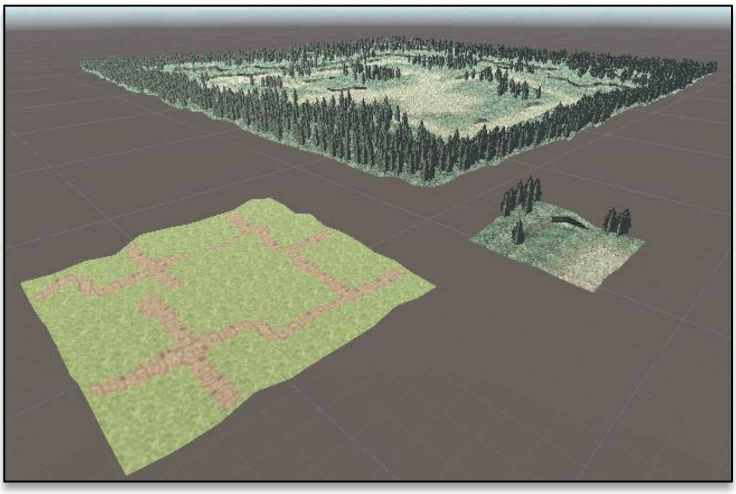


You can fix this by closing the tool and open it again.

Examples

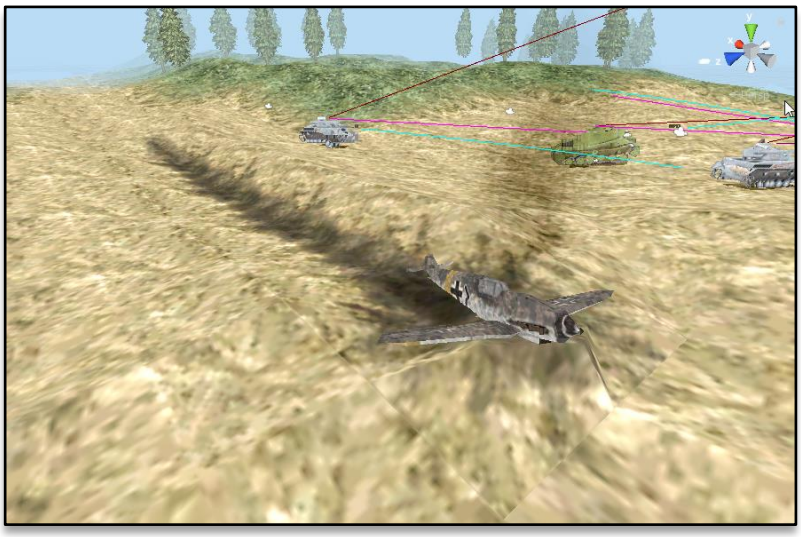
Three examples are given in the package. Load the configs to know how each have been built.





Special thanks

Thanks to Triple Axis, creator or [PSX Effects shader](), for giving me a lot of inspiration and the taste of PS1 demakes. During lockdown times, Unity was a real great deal to learn new aspects of coding.



Lucas Fierfort™

JackCarter / Drelias