



PATileTerrain

Installation

Import all files from the PATileTerrain unitypackage file.

Requirements

Texture must have the following flags:

1) Read/Write Enabled - enabled

Read/Write Enabled ☒

2) Generate Mip Maps – disabled

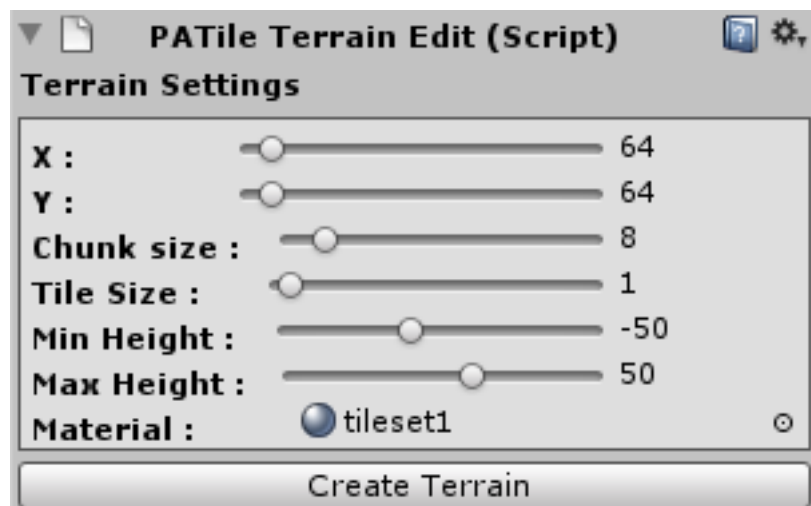
☐ **Generate Mip Maps**

If you want to edit the vertex colors, you need to use a shader, which uses the vertex colors. You can use the **PATileTerrainVertexColors.shader** located in “Assets\PATileTerrain\Shaders” or create your own.

IMPORTANT: You can have several types of tiles, but they should all have a common type that is transitional between all the others. For example, we have 4 types of tiles. This is a grass, a dirt, a sand and a rock. All four types have the common type – dirt. If this requirement is not fulfilled, the paint-algorithm will not work correctly.

How to use

1) Create a new GameObject and add an instance of **PATileTerrainEdit.cs** to this new GameObject. Select the new GameObject.



X, Y – terrain size

Chunk Size – how many tiles in one chunk

Tile Size – tile size in units of Unity3D

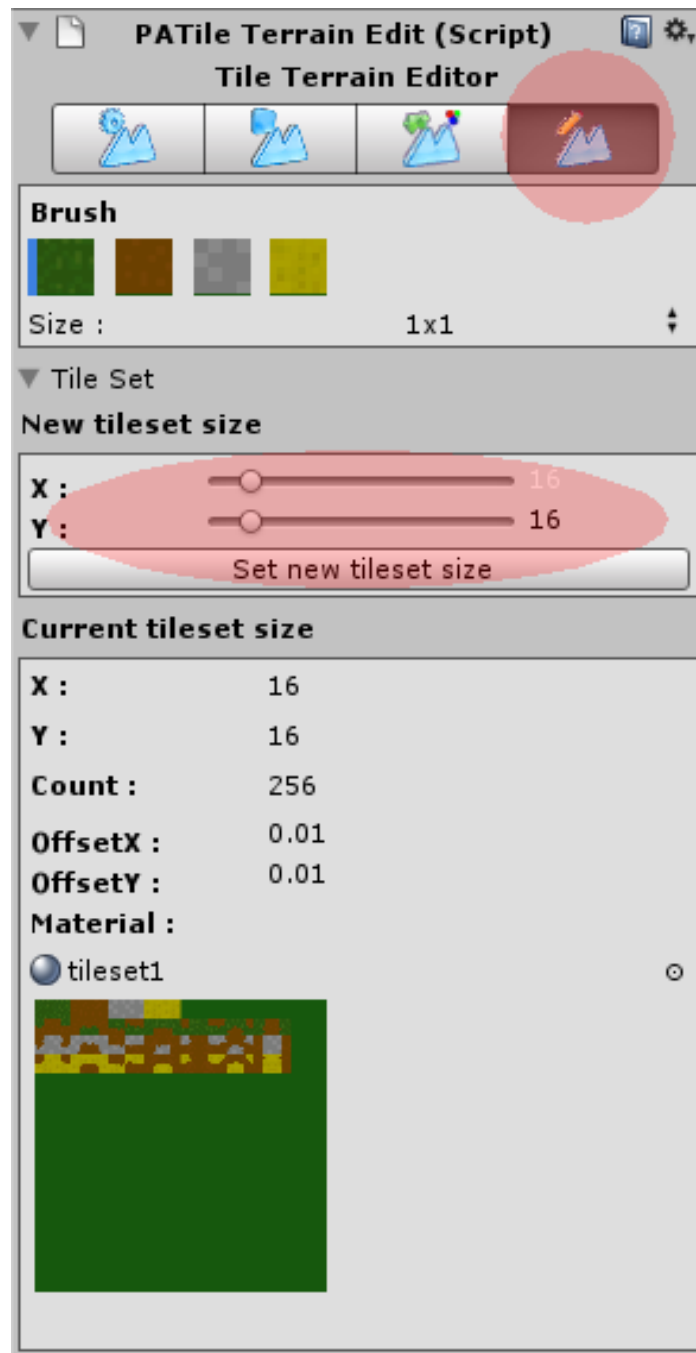
Min Height - the minimum allowed height

Max Height - the maximum allowed height

Material - material with a texture tile

2) Select the settings for a new terrain and then click **Create Terrain**.

3) Now you must configure the tileset. You need to specify the number of tile pieces. Set values for X and Y in the section **New tileset size**, then click **Set new tileset size** and confirm the change.



[illegible]

Add new tile types. Adjust the base index and name for each of the new types.


▼ Tile Type

▼ Type - grass

Id : 0

Name : grass

Base indexes :

Index 0: 0 

Remove base index 0

Fill terrain

Add base index


Remove type

▼ Type - dirt

Id : 1

Name : dirt

Base indexes :

Index 0: 2 

Remove base index 0

Index 1: 3

Remove base index 1

Fill terrain

Add base index

Remove type

► Type - rock

► Type - sand

Add Type

Add new transitions tiles. Then click the **Add Transition** and set all the indices of the new transition.

▼ Tile Transition



▼ Transition - grass-dirt


From : grass


To : dirt

Name : grass-dirt


Transitions :


From -  To - 

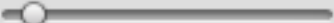





16










17




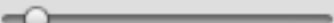





18










19




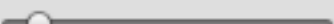





20










21




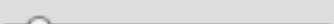





22




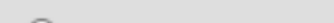





23




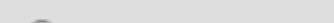





24




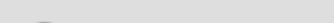





25




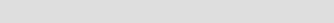





26




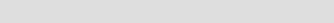





27




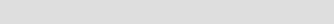





28





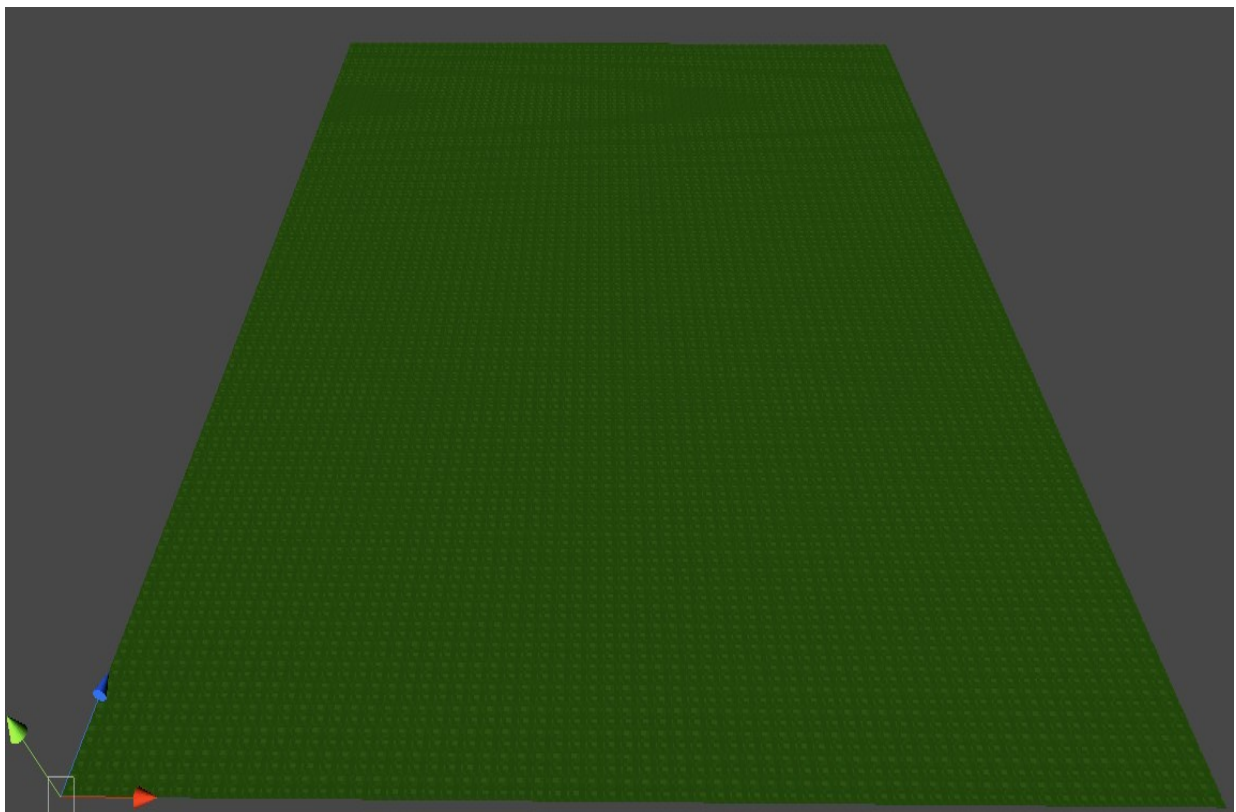
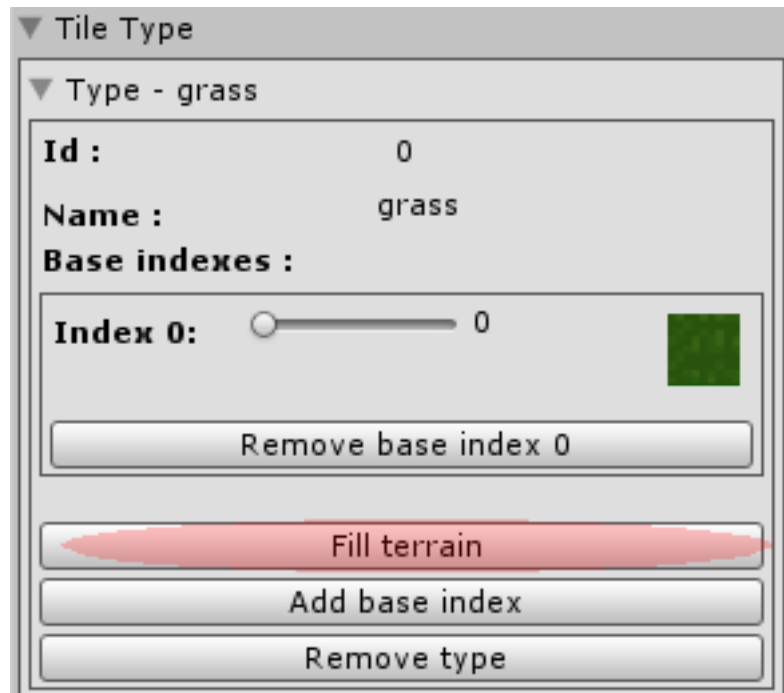


29

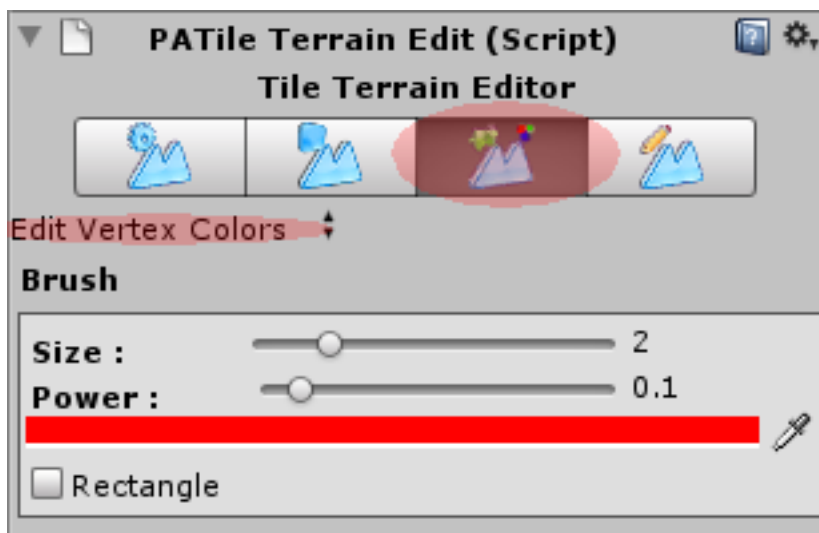


Remove transition

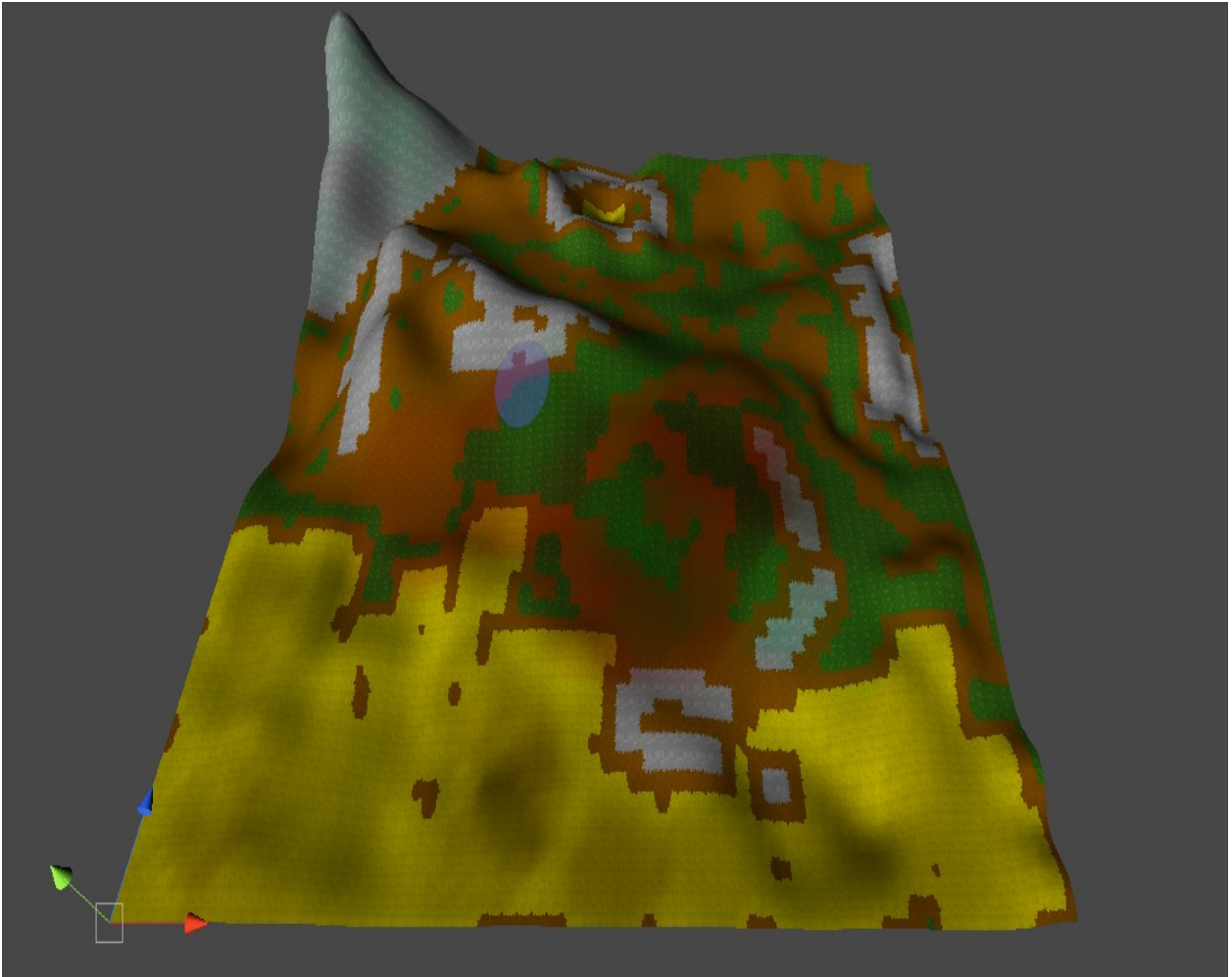
4) Choose one of the types and click **Fill terrain**.



5) Now you can edit the heights and color of the vertices.



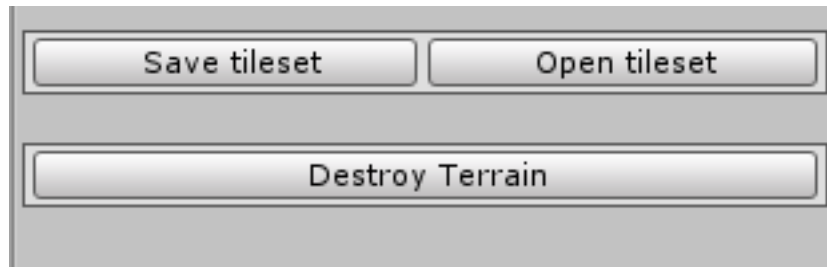
6) The final terrain.



Other

You can save and load tilesets.

You can also destroy terrain and create new.



Scripting

For access to the terrain, use PATileTerrain.cs.

```
PATileTerrain terrain = GetComponent<PATileTerrain>();
```

You can use only the following methods and properties:

bool FindPath(int startX, int startY, int targetX, int targetY, out PathData path)

Simple pathfinding from point A(startX, startY) to point B(targetX, targetY). If the method returns true, you can use PathData, which contains the path data.

PATile GetTile(int x, int y)

PATile GetTile(int index)

Returns information about the tile. For the tile you can specify the Id, Name, Walkability and Custom Data.

bool GetWalkability(int x, int y)

bool GetWalkability(PATile tile)

Returns walkability of the tile.

float GetHeight(Vector3 pos)

Returns the height in the 'pos' or 0 if it can not be obtained.

float GetPointHeight(int x, int y)

Returns the height of the specified point.

X range [0, <PATileTerrain>.settings.width].

Y range [0, <PATileTerrain>.settings.height].

You can also make some changes in a PATile (walkability, custom data, id, name).