# Playmaker For STE Manual


## Creative Spore

## Support: creativespore@gmail.com

## Web: http://www.creativespore.com

## Unity Forums: user CreativeSpore

## Twitter: @CreativeSpore

# Index

# 1 Introduction

This is the user manual for the Unity asset Playmaker For Super Tilemap Editor (STE).

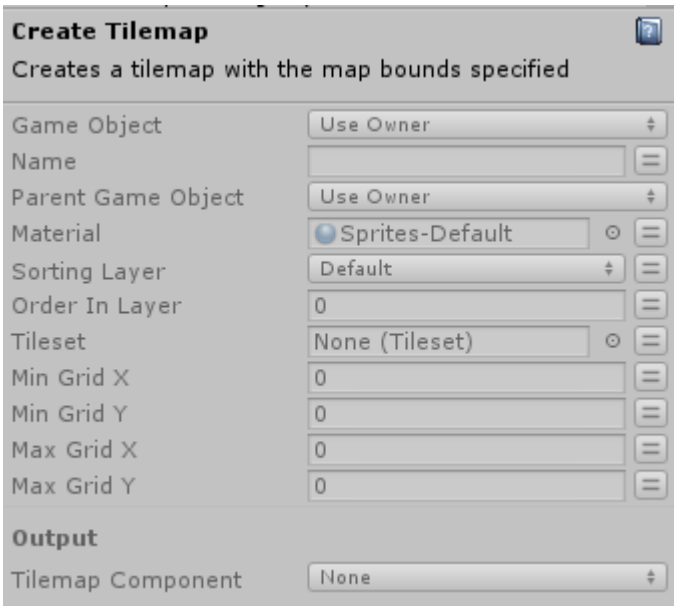
This manual will describe you each actions with examples of use.

# 2 Actions

## *2.1* Tilemap Actions

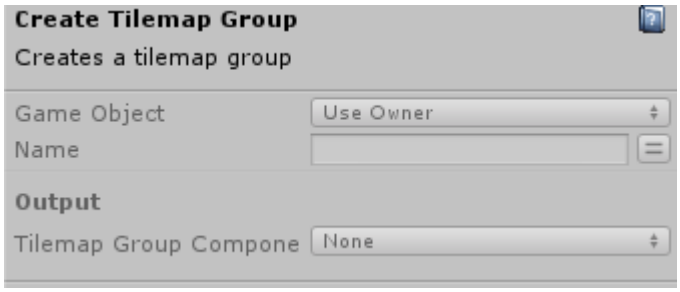
### 2.1.1 Create Tilemap

Creates a tilemap and set the parent object to the Parent Game Object specified. If Game Object is not specified, a new game object will be created with the name set in the Name property. If a Game Object is specified, the Tilemap Component will be attached to that game object.



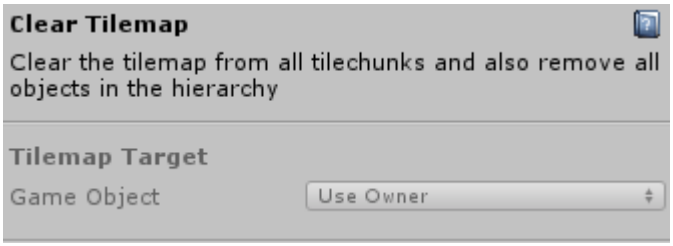

### 2.1.2 Create Tilemap Group

Creates a tilemap group to manage a list of tilemaps as children game objects. To add a tilemap to a tilemap group use the Create Tilemap action with the tilemap group object set as Parent Game Object.




### 2.1.3 Clear Tilemap

Clear the tilemap from all tilechunks and also remove all the objects in the hierarchy.

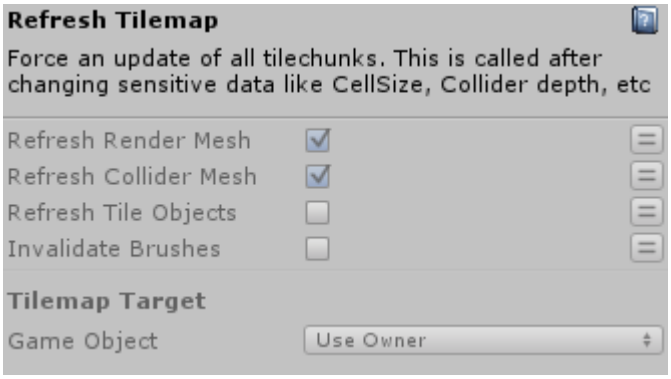



### 2.1.4 Update Tilemap Mesh

Updates the render mesh and mesh collider of all tile chunks. This should be called once after making all modifications to the tilemap with SetTileData.

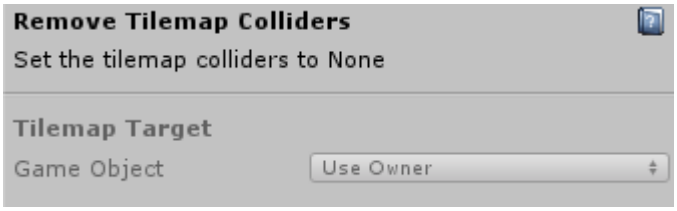

### 2.1.5 Refresh Tilemap

Force an update of all tilechunks. This is called after changing sensitive data like CellSize, Collider depth, etc.




### 2.1.6 Remove Tilemap Colliders

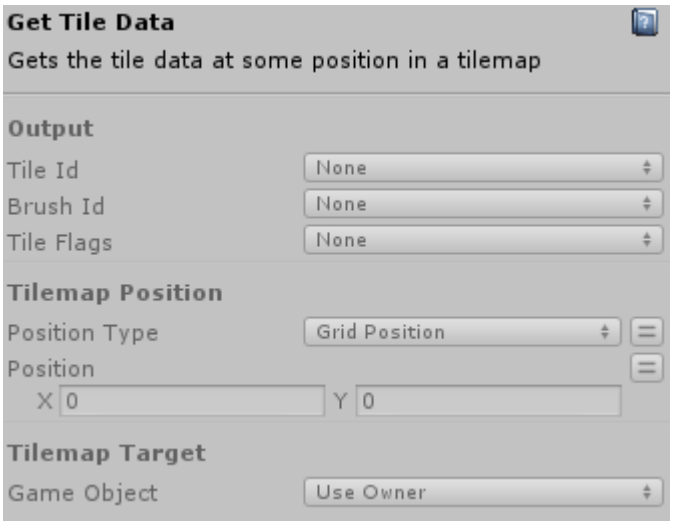
Set the tilemap colliders to None




### 2.1.7 GetTileData

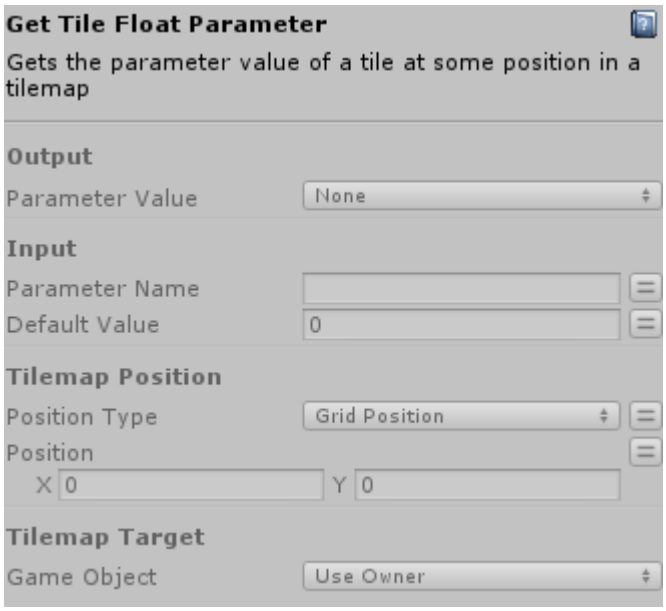
Gets the tile data at some position in a tilemap.

Note: SetTileData is named DrawDot and it's in the Painting category.



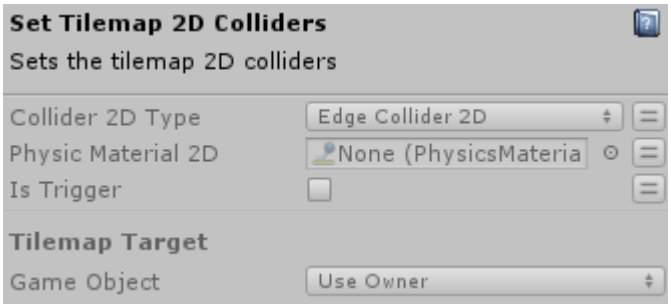
### 2.1.8 Get Tile Parameter (Bool, Float, Int, Object, String)

Gets the parameter value of a tile at some position in a tilemap.



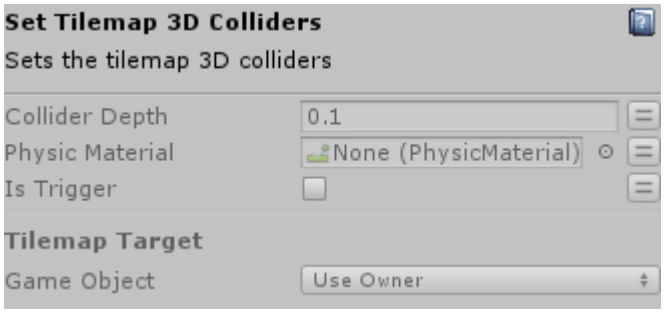

### 2.1.9 Set Tilemap 2D Colliders

Sets the tilemap 2D colliders.



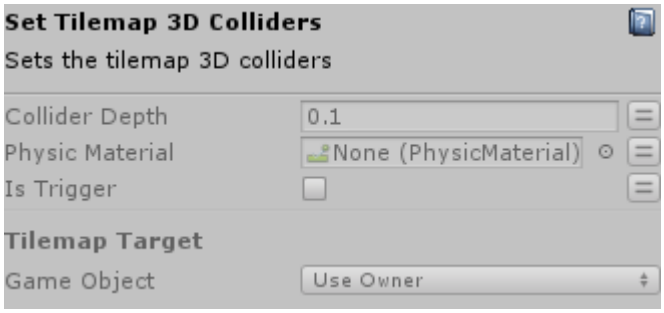

### 2.1.10 Set Tilemap 3D Colliders

Sets the tilemap 3D colliders.



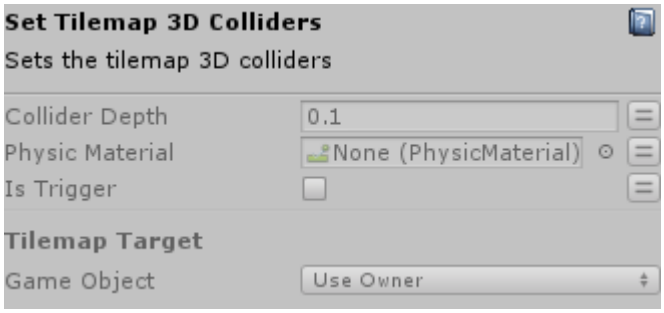

### 2.1.11 Set Tilemap Cell Size

The size of the cell containing the tiles. You should call Refresh() after changing this value to apply the effect.



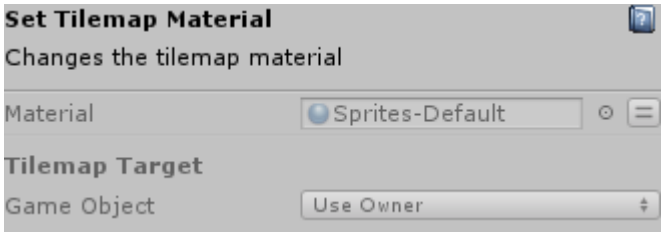

### 2.1.12 Set Tilemap Chunk Renderer Properties

Changes the renderer properties of all the chunks in a tilemap.



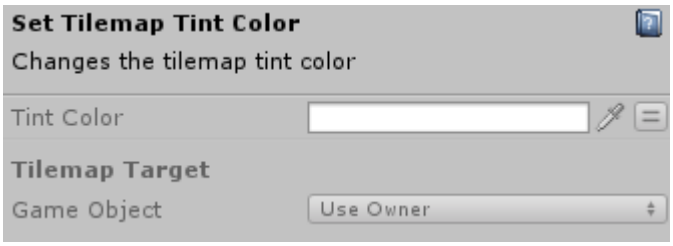

### 2.1.13 Set Tilemap Material

Changes the tilemap material.



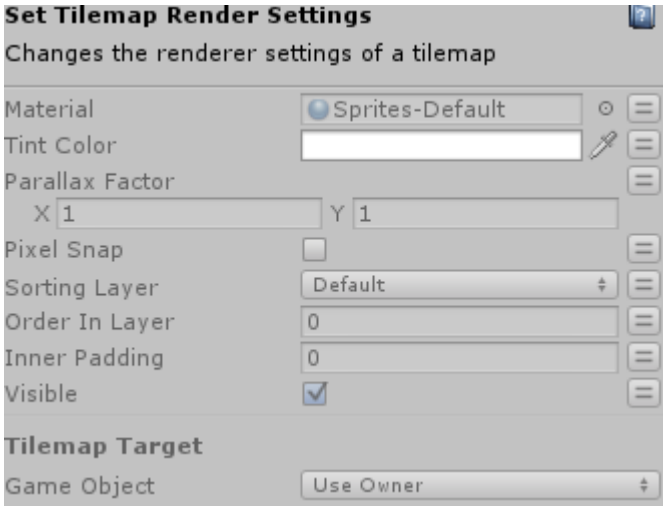

### 2.1.14 Set Tilemap Tint Color

Changes the tilemap tint color



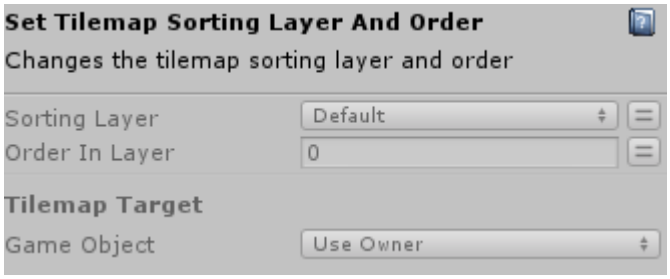

### 2.1.15 Set Tilemap Render Settings

Changes the renderer settings of a tilemap.



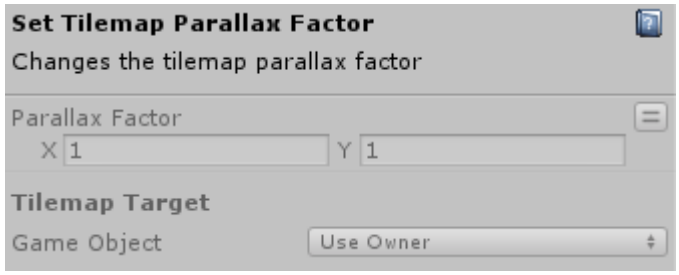

### 2.1.16 Set Tilemap Sorting layer and order

Changes the tilemap sorting layer and order.



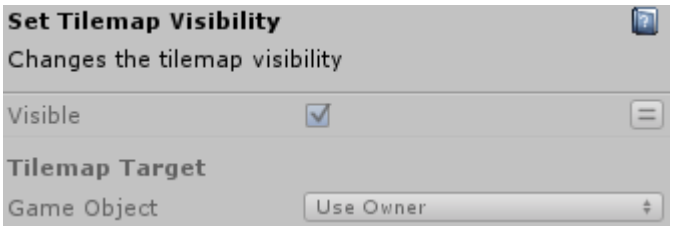

### 2.1.17 Set Tilemap Parallax Factor

Changes the tilemap tint color.




### 2.1.18 Set Tilemap Visibility

Changes the tilemap visibility.



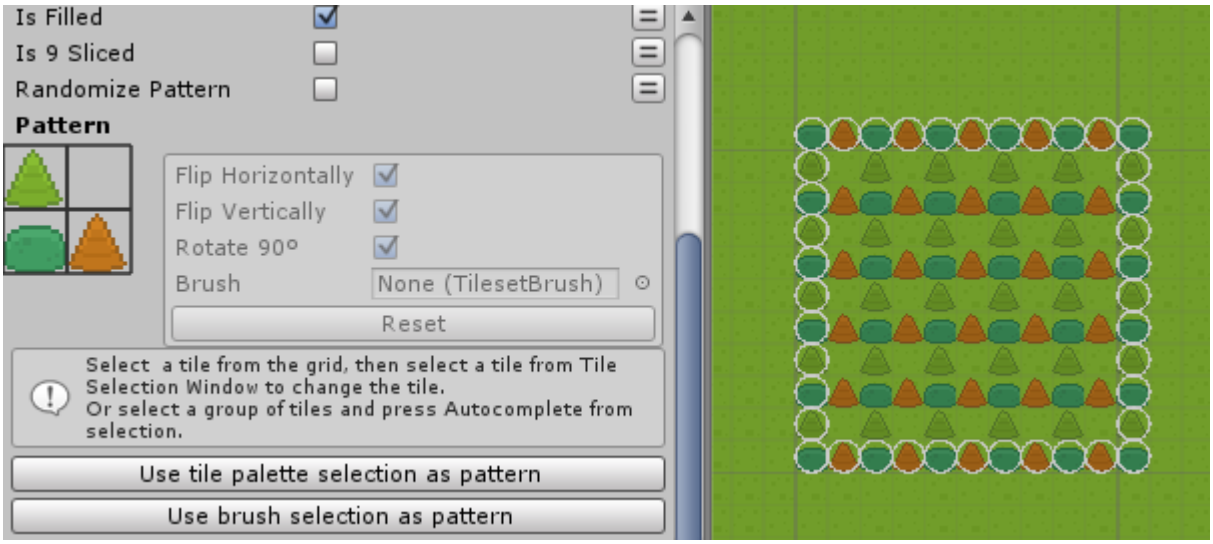

## 2.2 Paint Actions

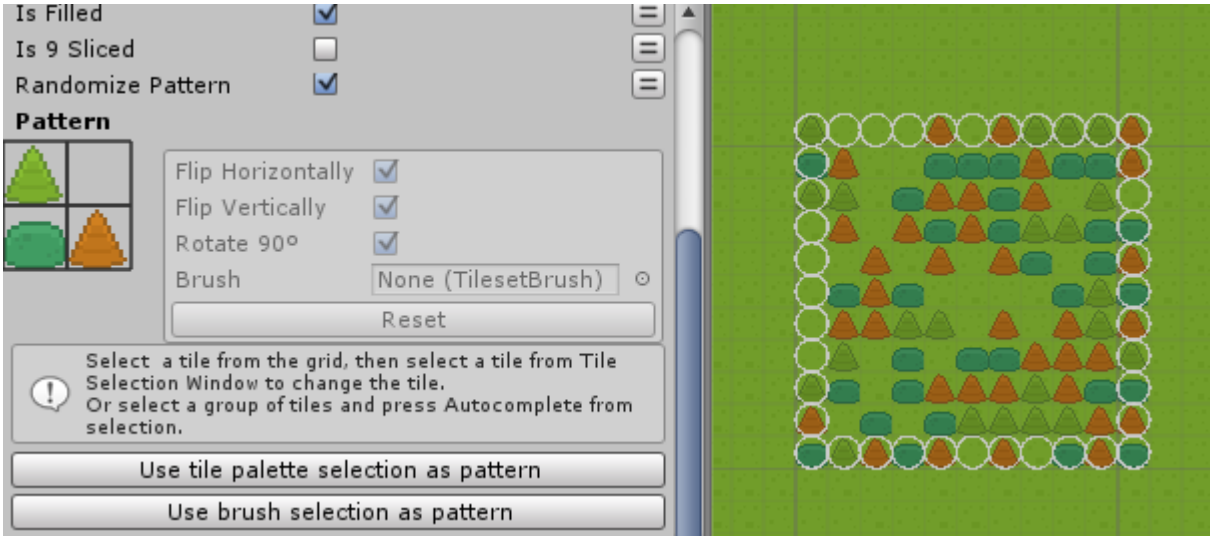
All paint actions share in common a target tilemap, a start painting positions and sometimes an end painting position and a pattern.

The pattern can be just a tile or a set of tiles. When the paint action is painting each tile, it will use a tile from the patter repeating it along the tilemap surface. If Randomize Patter is checked, the tile will be taken randomly.

Example of Patter being painted with the Draw Rect Action with a

without Randomize Pattern.




To create the patter you have two options. You can make a selection of tiles from the tile palette and press the button “Use tile palette selection as pattern” or you can make a selection from a tilemap and press “Use brush selection as pattern”. This is the only way to increase the patter size.

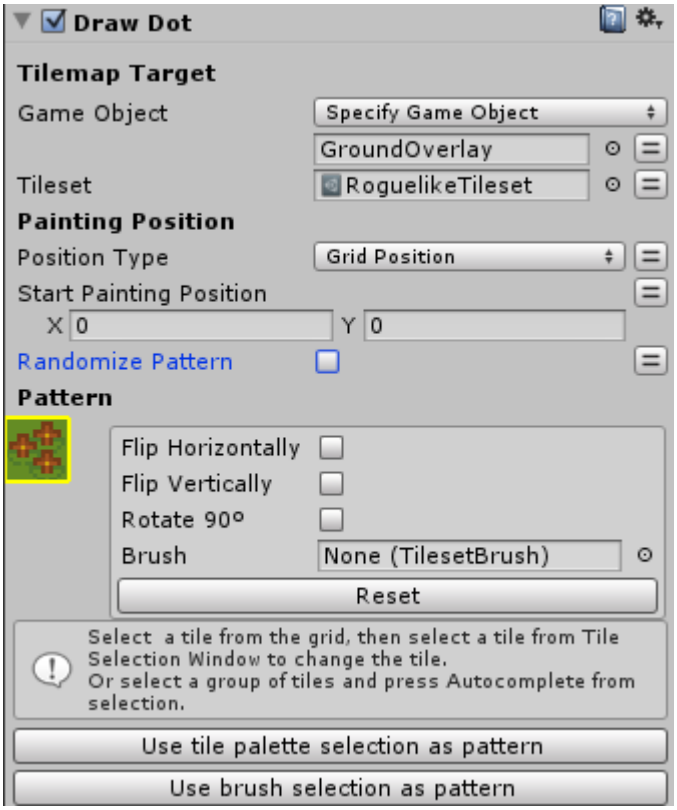
Once you have a pattern, you can select one by one each tile of the patter and select another tile or brush and modify the flags.

The paint actions also draw a gizmo in the scene with the outline of the final shape, for example a rectangle for the Draw Rect or an ellipse for the Draw Ellipse action. This won't be showed if the cell size cannot be

calculated. To calculate the cell size, the tilemap needs to be a reference to a game object with a created tilemap, not a variable. If the tilemap component cannot be found, the tileset will be used, calculating the default cell size through the PixelsPerUnits property and the tile size in pixels.

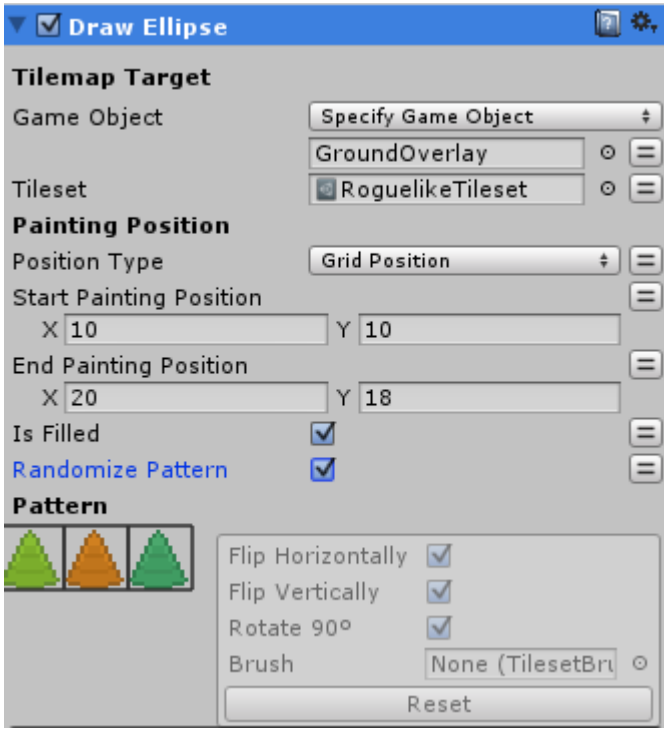
### 2.2.1 Draw Dot

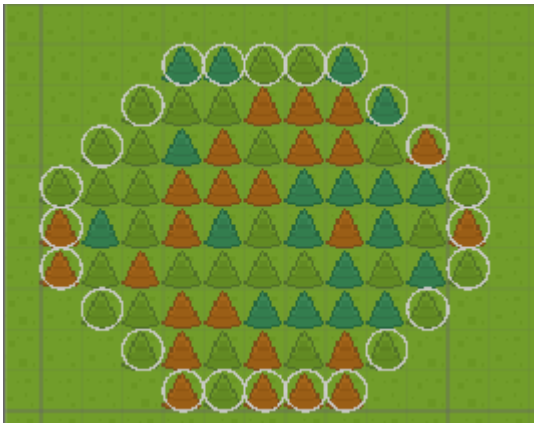
Draws a dot using a patter over a tilemap.




### 2.2.2 Draw Ellipse

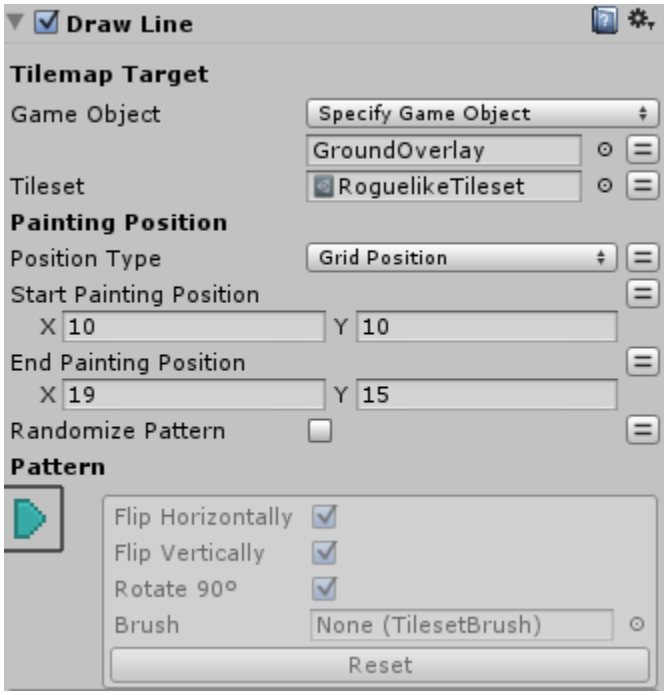
Draws an ellipse using a patter over a tilemap.

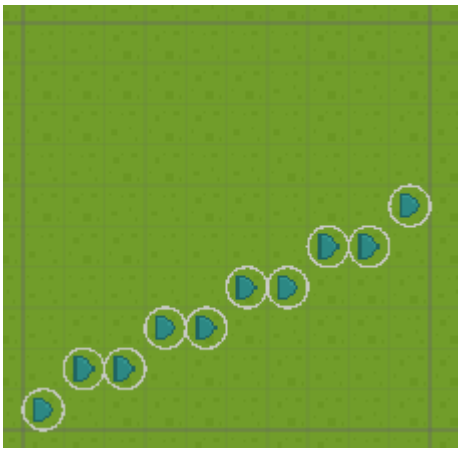


## 2.2.3 Draw Line

Draws a line using a patter over a tilemap.



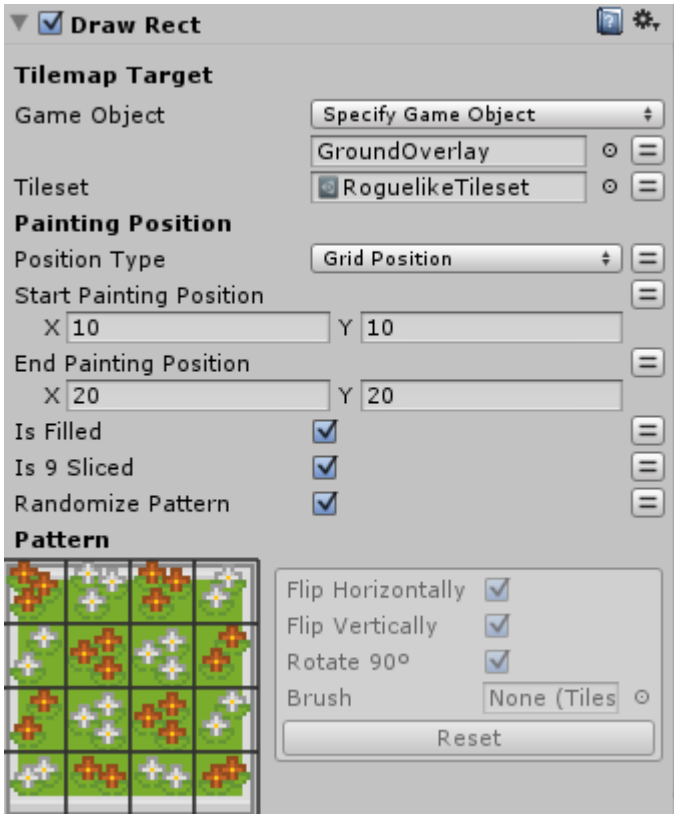



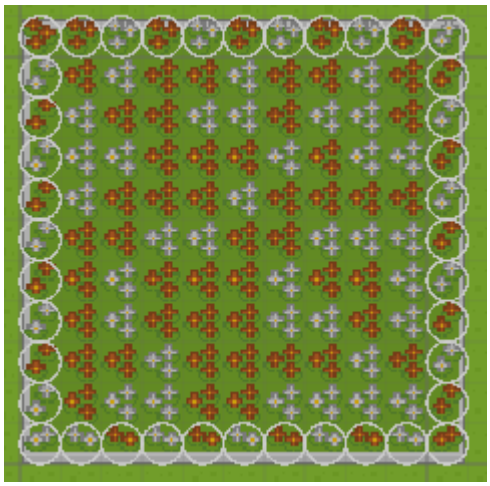
### 2.2.4 Draw Line Mirrored

Like DrawLine, draws a line from the start position to the end position and another one from start position in the opposite direction.

### 2.2.5 Draw Rect

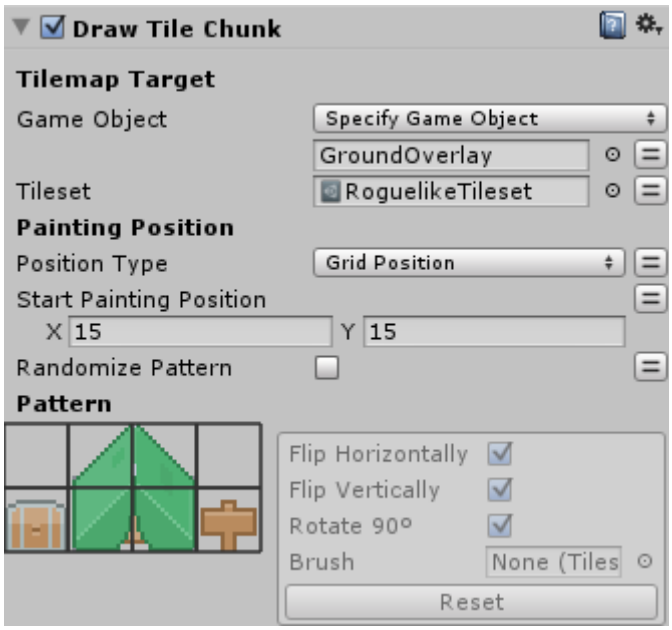
Draws a rect using a patter over a tilemap.



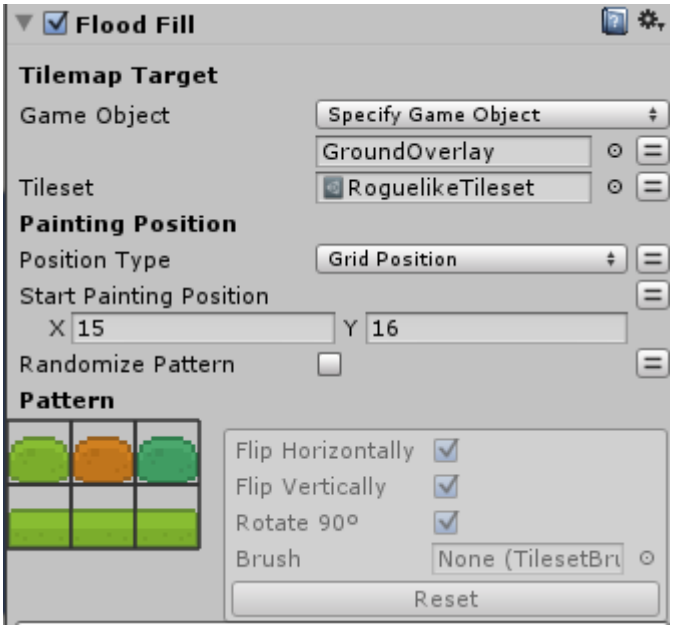
### 2.2.6 Draw Tile Chunk

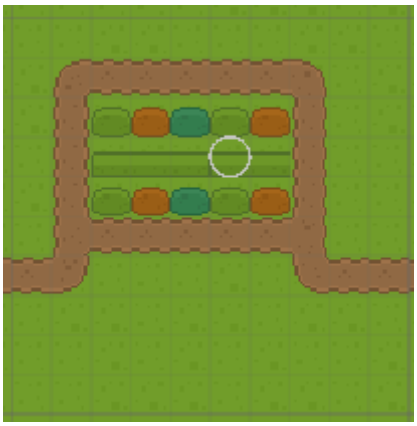
Draws a pattern centered in the start painting position.




### 2.2.7 Flood Fill

Use the flood fill tool over a tilemap.



## 2.3 Color Paint Actions

These actions modify the color channel of a tilemap. Each tile is a tilemap is rendered using 2 triangles and 4 vertices. You can change the color of each vertex using these actions, clear the color channel color to a single

color, remove the color channel to save memory or get the color of a tile.

### 2.3.1 Clear Color Channel

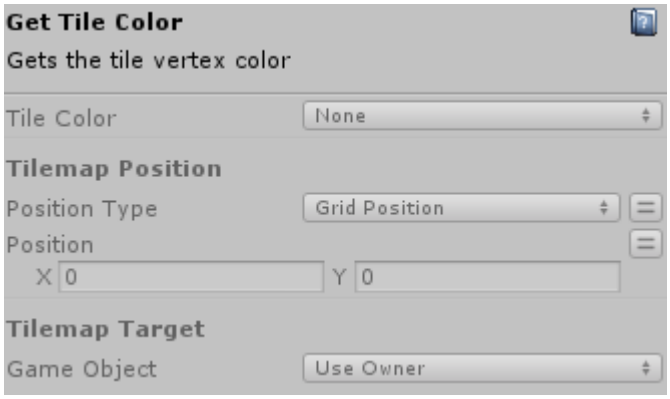
Clear the color channel with a single color.


### 2.3.2 Remove Color Channel

Remove the color channel of a tilemap.

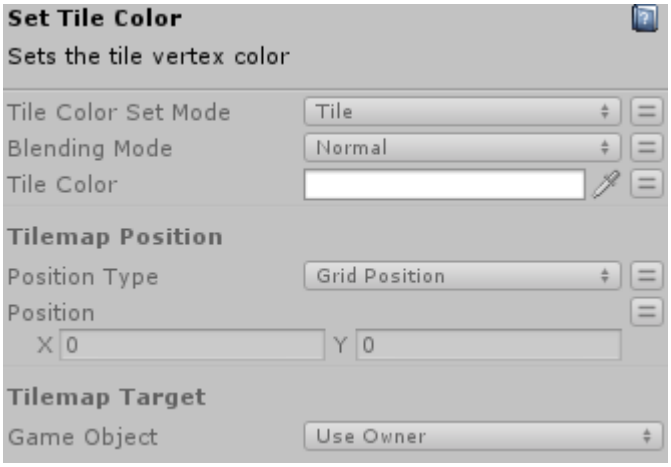

### 2.3.3 Get Tile Color

Gets the tile vertex color.




### 2.3.4 Set Tile Color

Sets the tile vertex color.




### 2.3.5 Tilemap Vertex Paint Circle

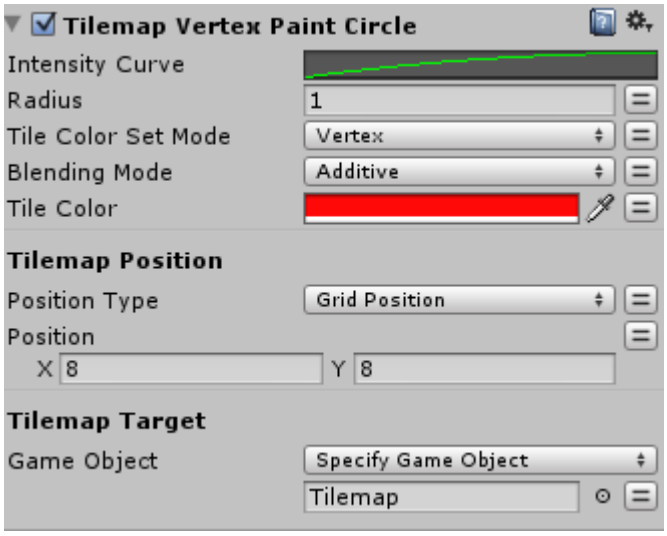
Paints the tilemap tiles using a circle shape. You can use the **Intensity Curve** to change the intensity of the Tile Color along the radius (the intensity will be multiplied to the color).

The **Tile Color Set Mode** changes how the color is applied:

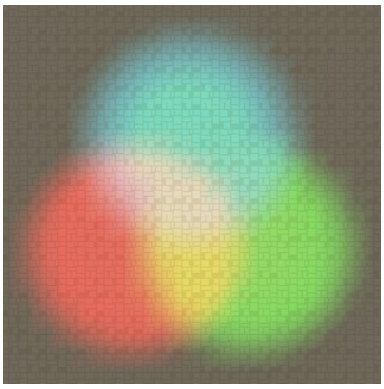
- **Tile**: will apply the same color to the 4 vertices of a tile.
- **Vertex**: will apply the color per vertex. Use this option for a smoother color effect.

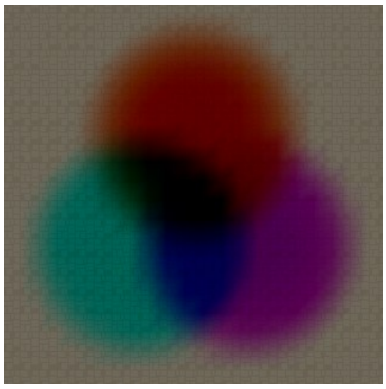
**Blending Mode** is used to specify how to apply the color over the previous color:

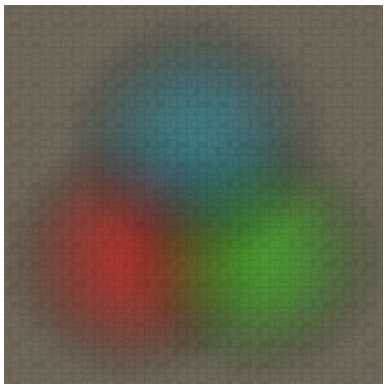
- **Normal**: overwrites the previous color.
- **Alpha** Blending: mix the color using the alpha component.
- **Additive**: sum the color components.
- **Subtractive**: rest the color components.
- **Multiply**: multiply the color components.




Examples using Additive, Subtractive and Alpha Blending:



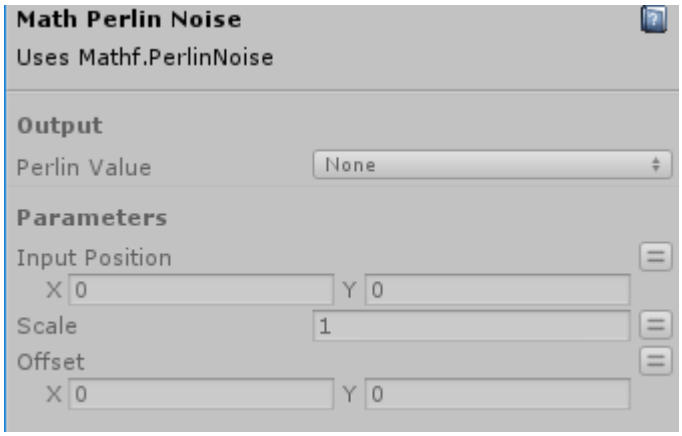




## *2.4* Utility Actions

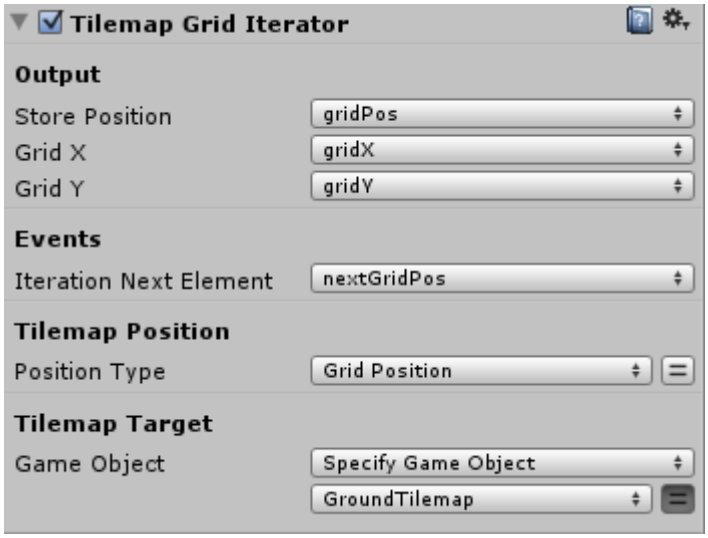
### 2.4.1 Perlin Noise

Uses Mathf.PerlinNoise with some more parameters like Scale and Offset. The Scale will be multiplied to the input X and Y parameters and after than, the Offset will be added.



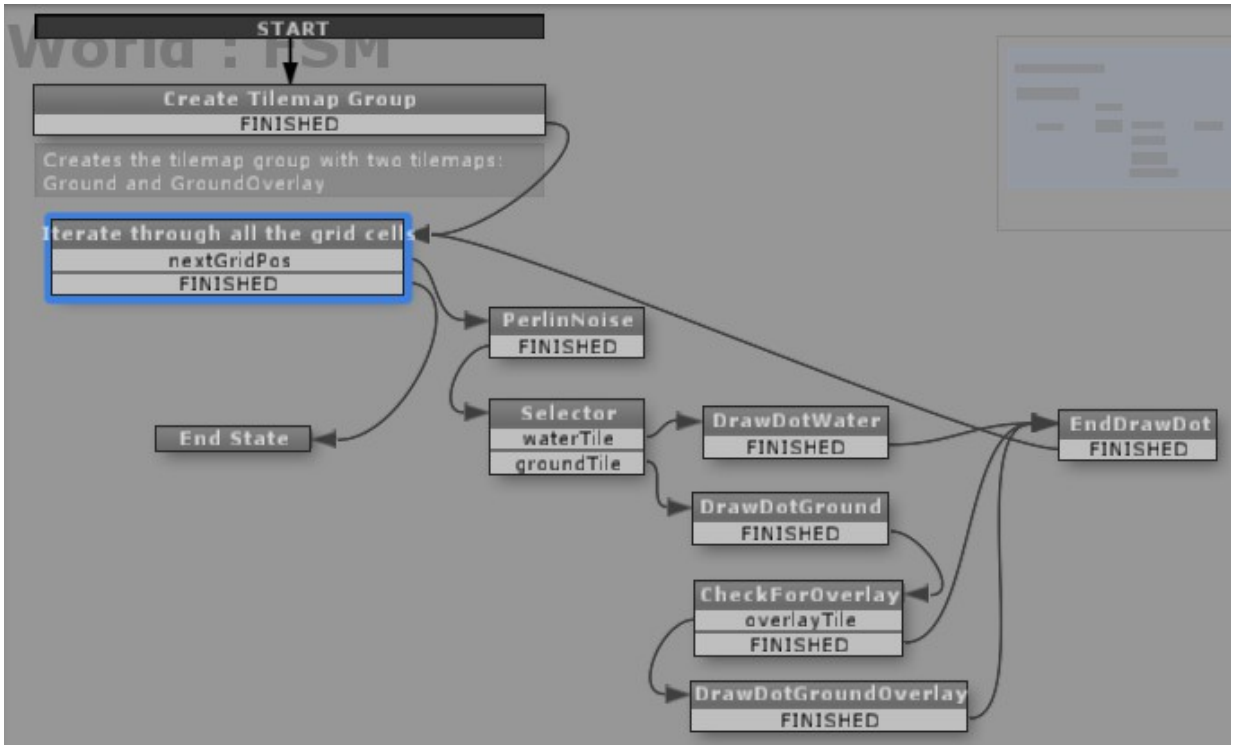

### 2.4.2 Tilemap Grid Iterator

Iterates through all the grid positions in a tilemap.




You can use this action to perform other action per each cell in the tilemap.

You can see an example of use in the demo scene: [PM4STE] Procedural Tilemap Demo.


START
Create Tilemap Group
FINISHED
Creates the tilemap group with two tilemaps:
Ground and GroundOverlay
Iterate through all the grid cell
nextGridPos
FINISHED
PerlinNoise
FINISHED
End State
Selector
waterTile
groundTile
DrawDotWater
FINISHED
EndDrawDot
FINISHED
DrawDotGround
FINISHED
CheckForOverlay
overlayTile
FINISHED
DrawDotGroundOverlay
FINISHED