



Procedural Tilemap Generator 2D Documentation

Armin Ahmadi
DEVELOPERS HUB

Overview

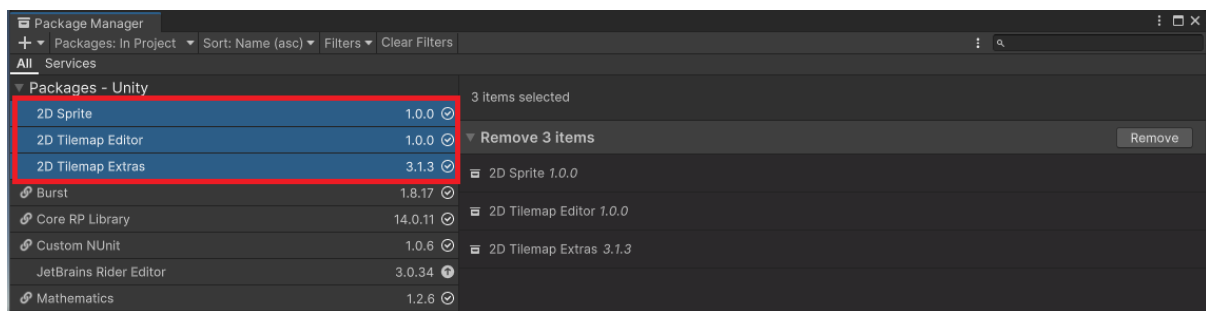
The **Procedural Tilemap Generator 2D** is a powerful Unity asset designed to help developers create dynamic, procedurally generated 2D tilemaps with ease. This tool is perfect for generating both static and infinite tilemaps, with support for various tile types, rules, and object placement. It is highly customizable, allowing developers to create unique and complex tilemaps for their games.

Note

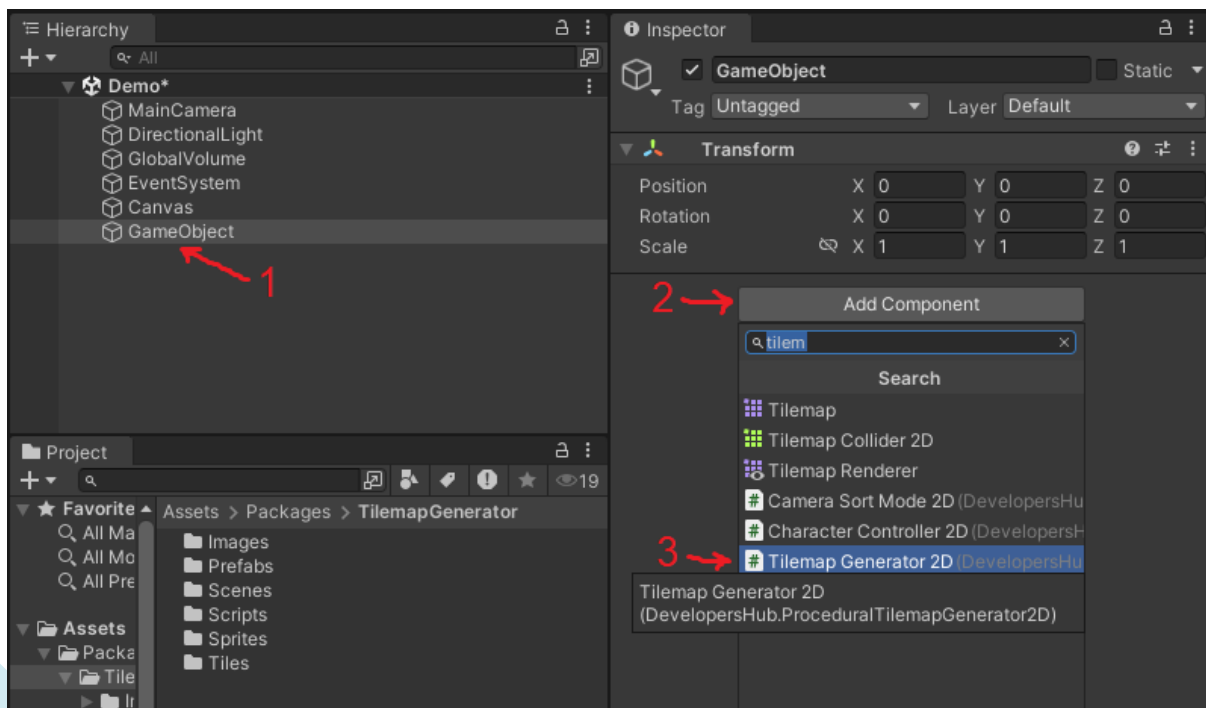
To get the proper result using this package, you need to follow all the instructions and your tile sprites and rule tiles must be in specified shape and have specified tiling rules.

Getting Started

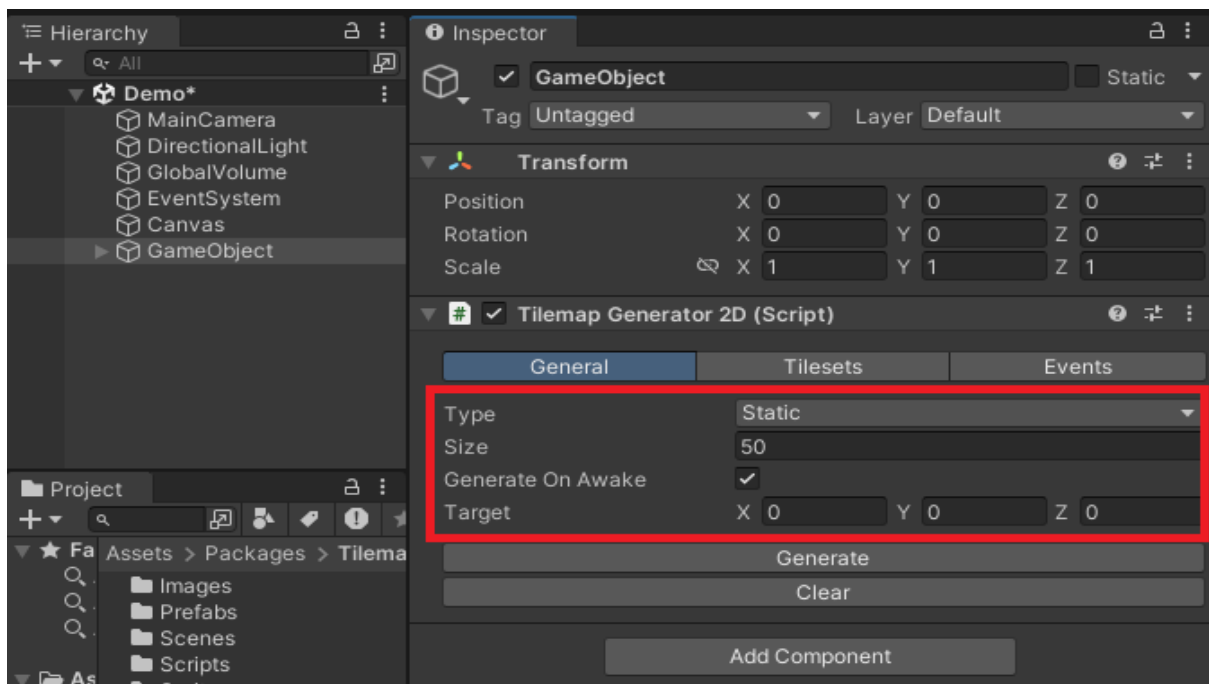
Before importing the package, make sure you have these packages installed.



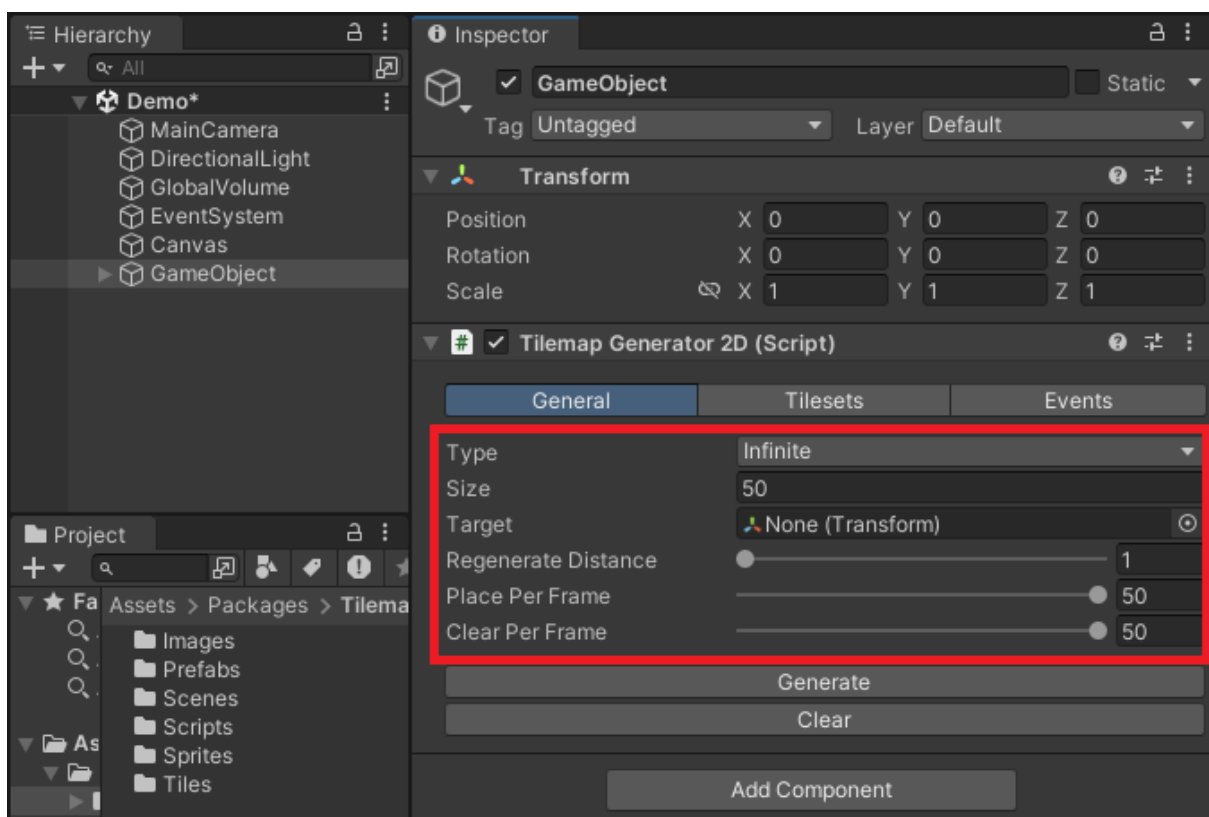
First create an empty game object and attach **TilemapGenerator2D** script to it.



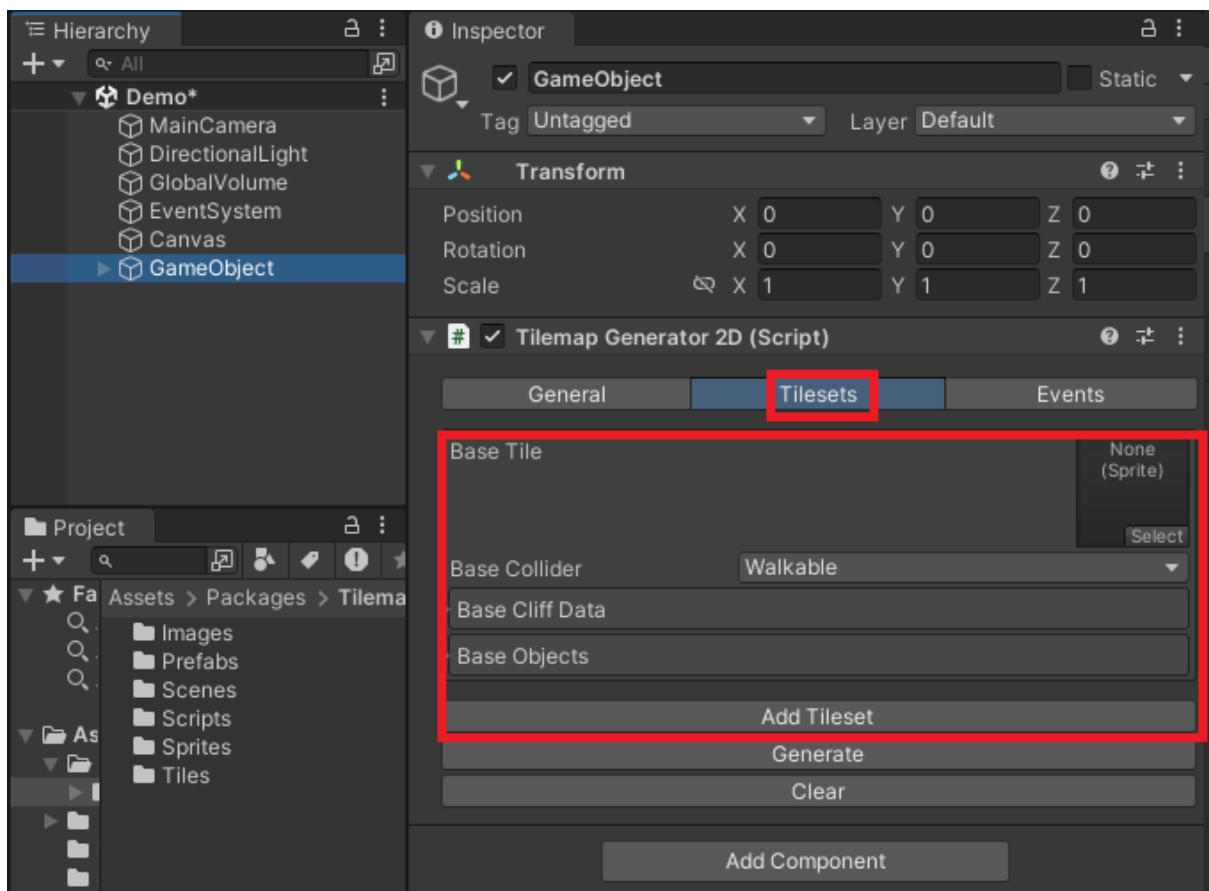
There are two types of generation. The first one is **Static** and it is used to generate around a specific position in your world with a specific size.



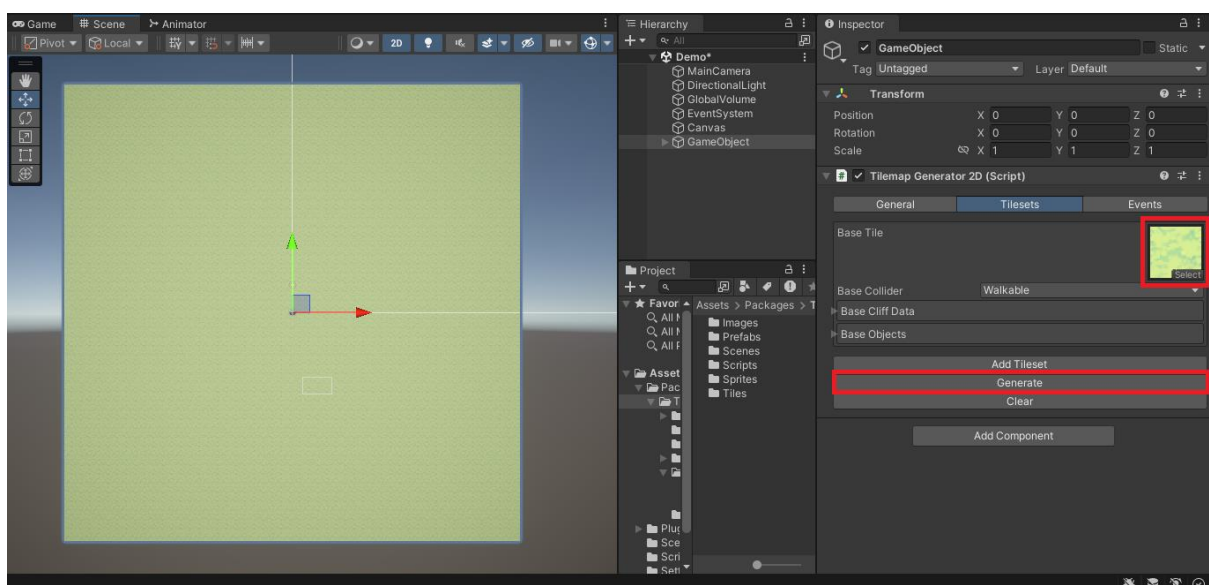
The second type is **Infinite** and it will generate around a specific transform in the world and if the target transform moves, the generation will follow.



Right now if you click on generate in the inspector nothing will happen because you haven't assigned your tiles yet. In the **Tilesets** tab, there is a variable named **Base Tile** takes a sprite and as the name suggests, it will be the base of your tilemap.

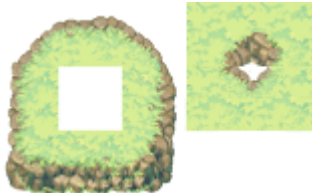


So if I assign a grass tile (in my case it's 32x32 pixels) to base tile and click generate, all tiles in my tilemap will change to that grass tile.

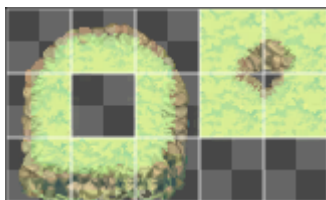


There is a variable called **Base Collider** which determines if the base tile is **Walkable** or it is **Not Walkable** and the colliders for it automatically be generated.

We also have **Base Cliff Data** which we can set a specific noise for it and create one level of heights for our tilemap (only one level of height is supported in this package). So to create the cliffs, we need the proper sprites for it. Here is the example.

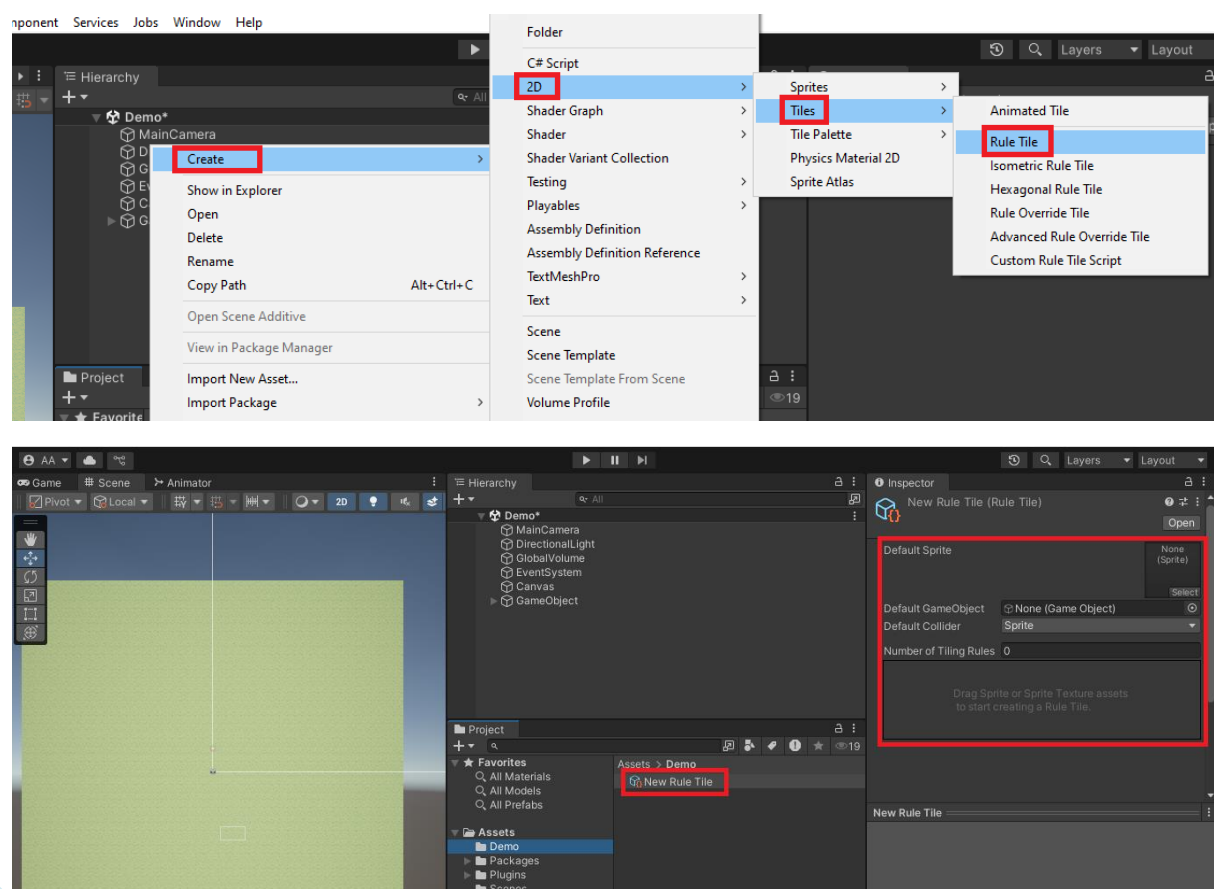


As you can see, it is a sprite sheet that contains 12 transitions between grass and cliff. You can import it as a single sprite sheet like this and slice it using sprite editor or import 12 individual sprites separately.

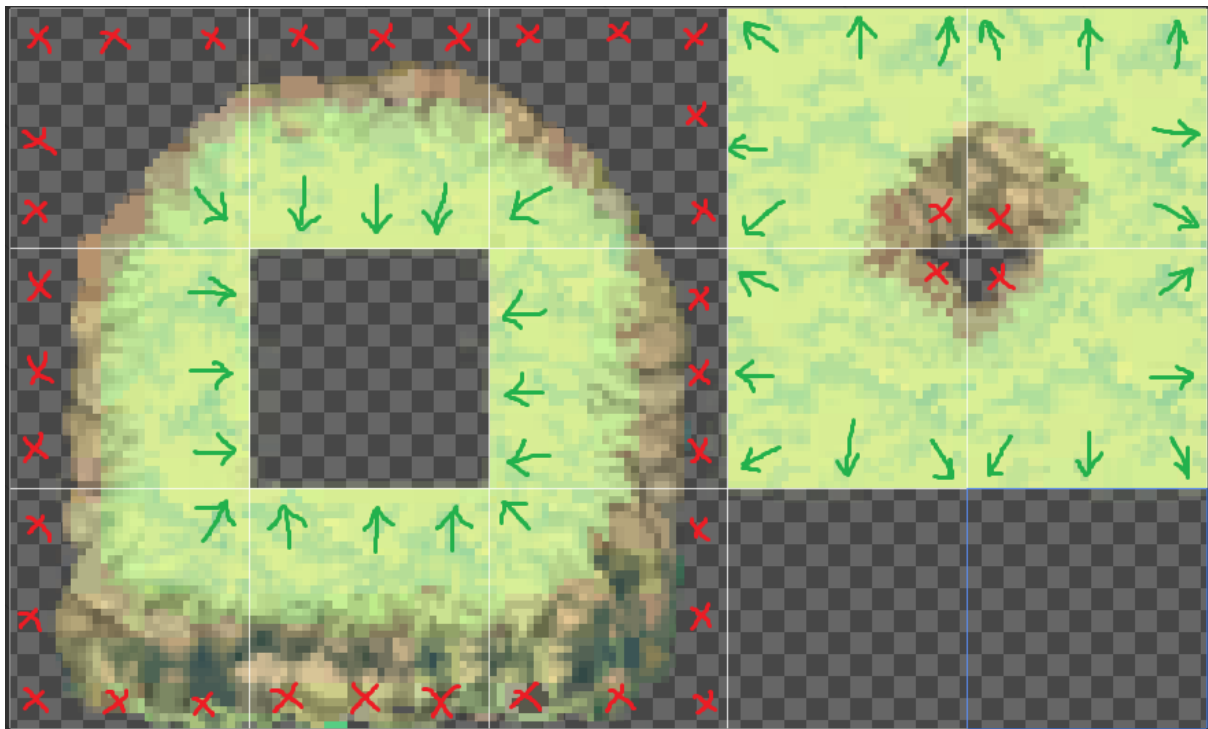


If you have the as a sprite sheet, make sure to set **Sprite Mode** to **Multiple** and click on sprite editor button and use the provided tool to slice it to separate sprites.

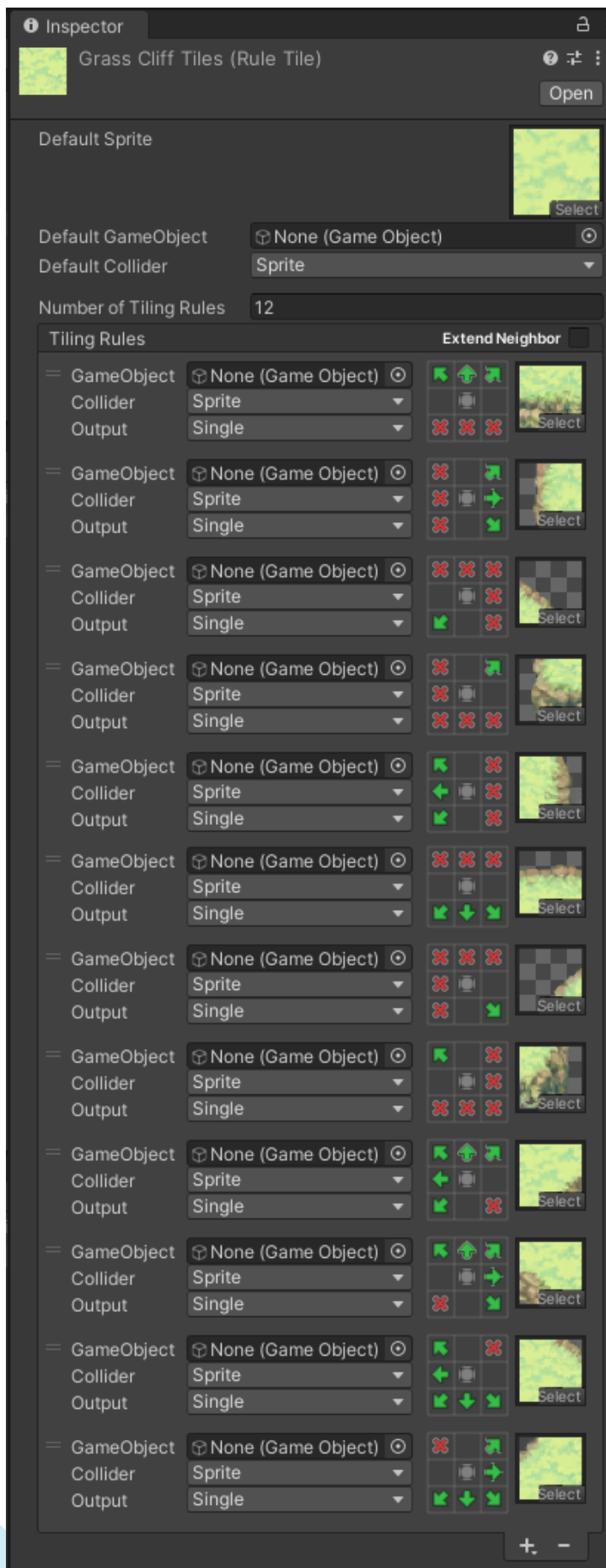
Now you need to create a rule tile for your cliffs.



For the **Default Sprite** we assign the grass tile that we chose for the **Base Tile** since my cliff is also have grass on it. We change the **Number of Tiling Rules** to **12** and set their values like this. The order of tiling rules is not important.

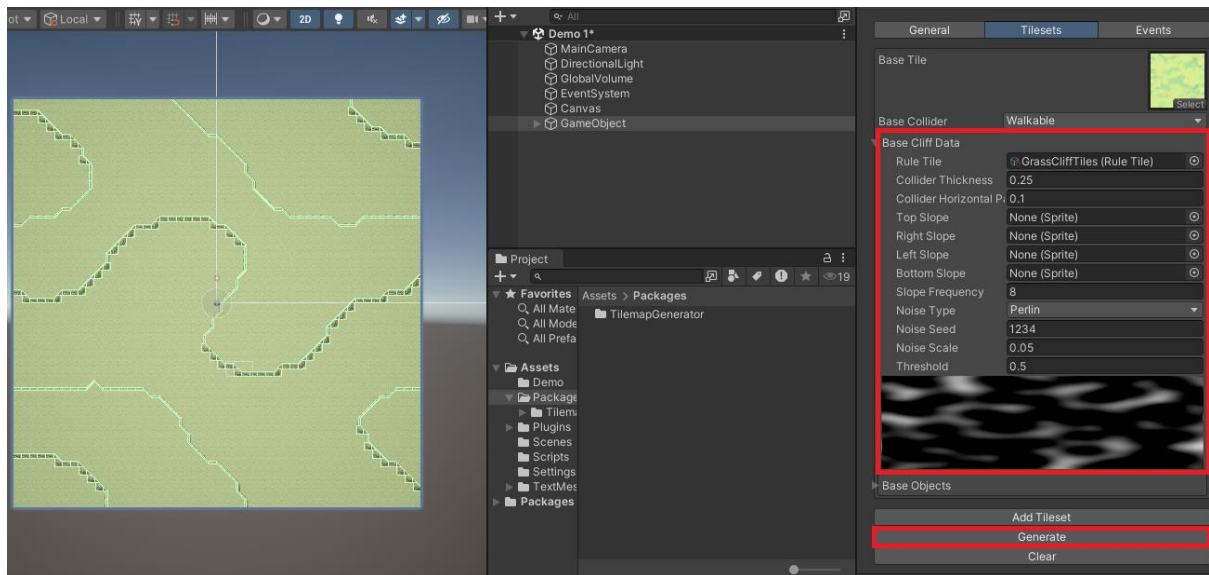


Do not change any other settings of the rule tile, only assign the sprite and set the rules based on the given example. Note that if you don't follow these instructions exactly, you will not get the proper outcome.

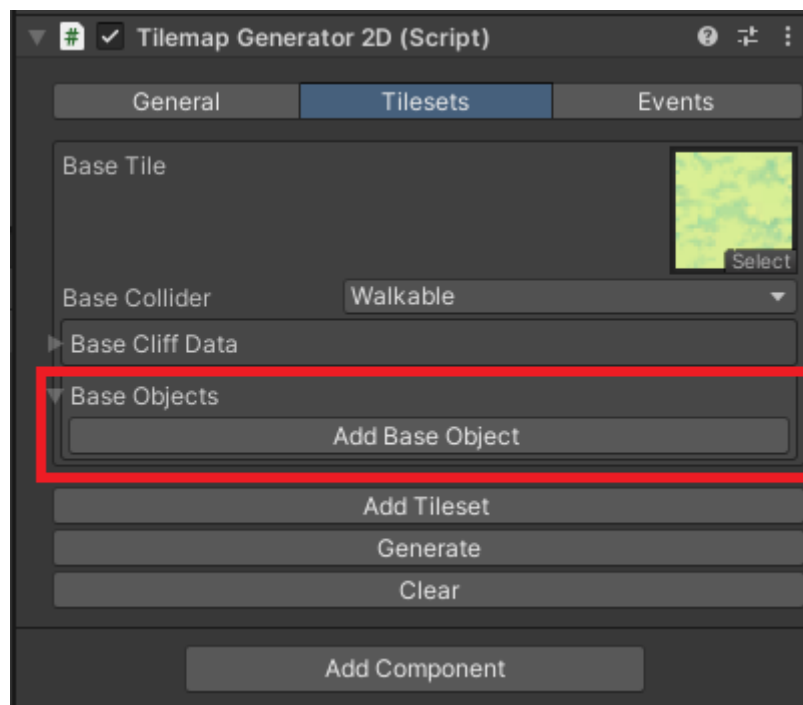


Here is a full view of the rule tile. As mentioned before, the order of them is not important.

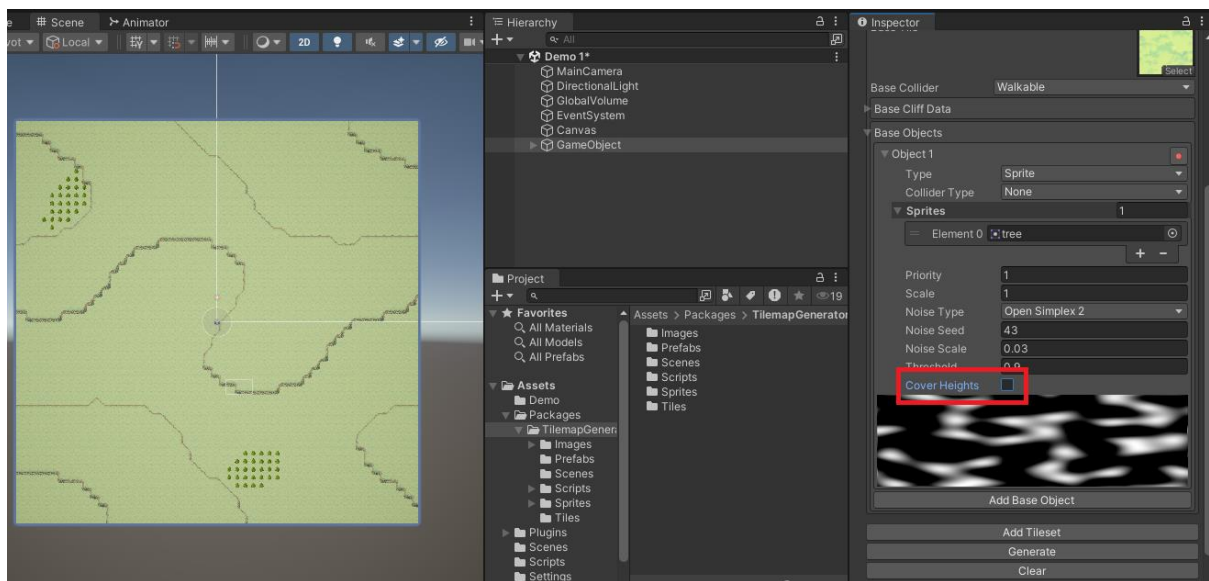
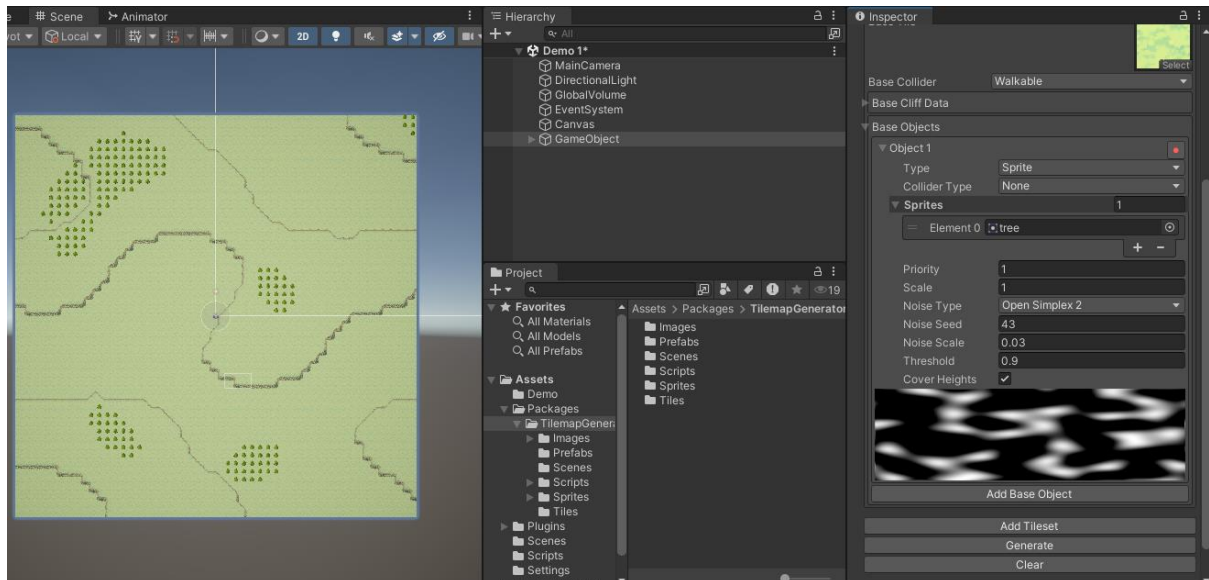
Now that we have a proper rule tile, we can assign it to **Base Cliff Data** and set our custom settings. **Noise Type**, **Noise Seed** and **Noise Scale** determine the shape of the noise and you can see the changes in the noise texture when you change those values. **Threshold** is a value between 0 and 1 which determines the acceptable level of noise. In the noise texture, the white areas have the value of 1 and the black areas have the value of 0 and the grey areas is a value between them. So the lower value for **Threshold** means more area of the noise will be accepted and wider cliff area. Finally, if you click the Generate button, you will see that the tilemap will be updated and the cliffs will be added.



There is another area named **Base Objects** that you can expand. Using this you can add objects like trees or rocks on your base tilemap.



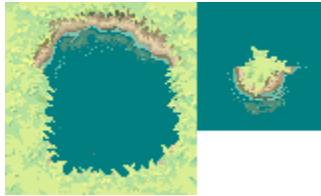
You can add a new object type by clicking Add Base Object. You can use a prefab to be instantiated in your world or use a sprite to be set on your tilemap. If you decide to use a prefab, then your prefab must have **TilemapObject2D** script attach to it. For both prefab or sprite, you have the option to assign multiple options and one of the will be selected randomly but consistent. Noise Type, Noise Seed, Noise Scale and Threshold are the settings for noise which determines how the object is scattered and the **Cover Heights** determines if the objects should be placed on cliffs as well. Here is an example of placing trees.



Note that if you put the **Type** on **Sprite**, a variable named **Collider Type** appears and you can use it to make a collider for you objects. If you put it on **Prefab**, then you'll have to create the collider for your prefabs manually.

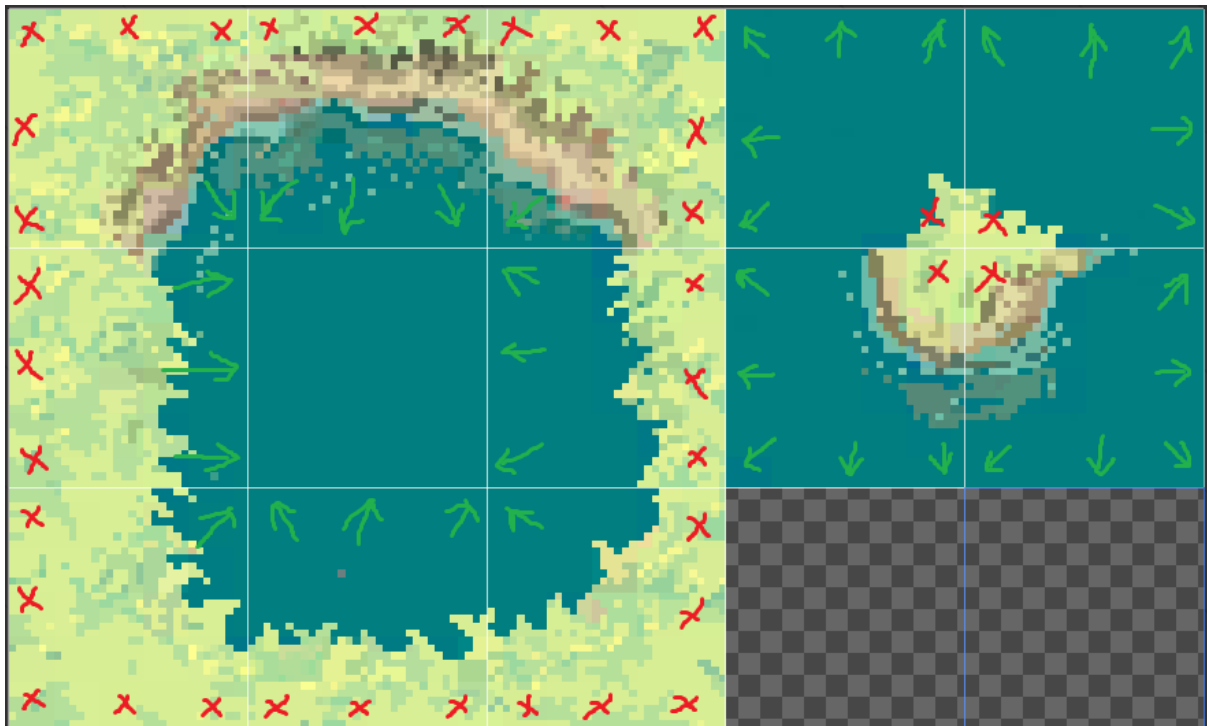
Priority is the importance of the object and when you have set multiple objects with different noise types, if in a specific area their noise overlaps, the object with higher priority will be placed and the one with lower priority will be ignored.

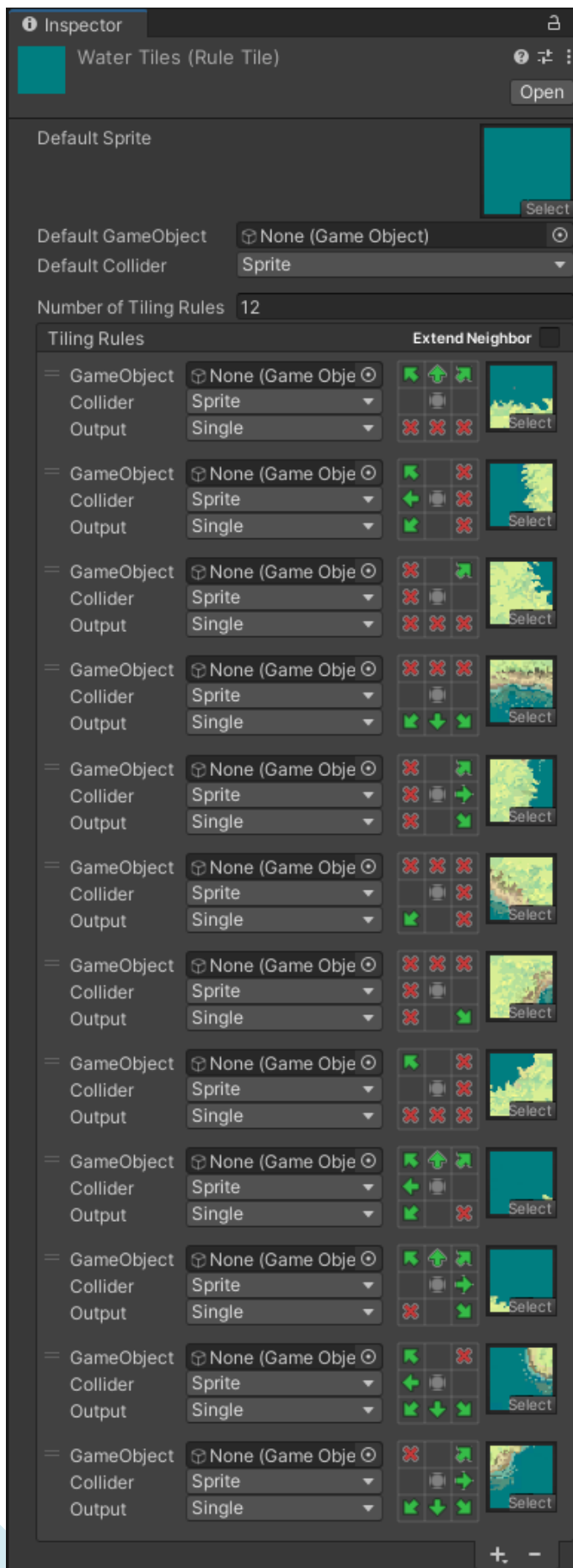
Now that we have covered the base tileset, we can have additional tilesets. For example, we want to have water areas on our base tileset which is grass, so we need grass-water transitions. You need 1 water tile and 12 transition tiles like this.



Basically any tilesets you want to use must have these 12 tiles to give you the proper result.

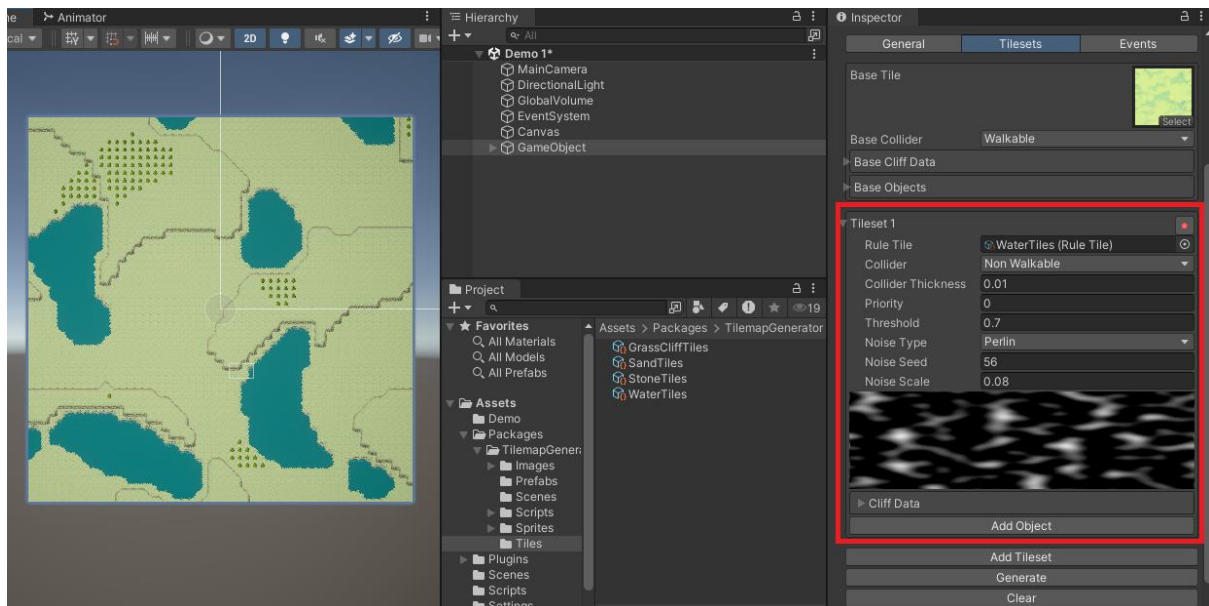
This is how you should set the rules.



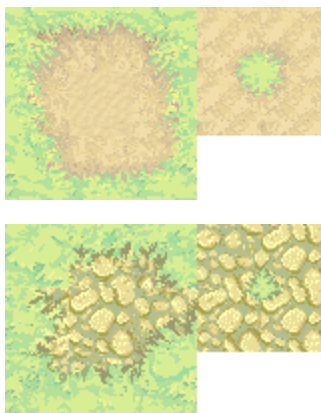


This is the full view of water-grass rule tile. Note that you should assign the water tile as **Default Sprite**.

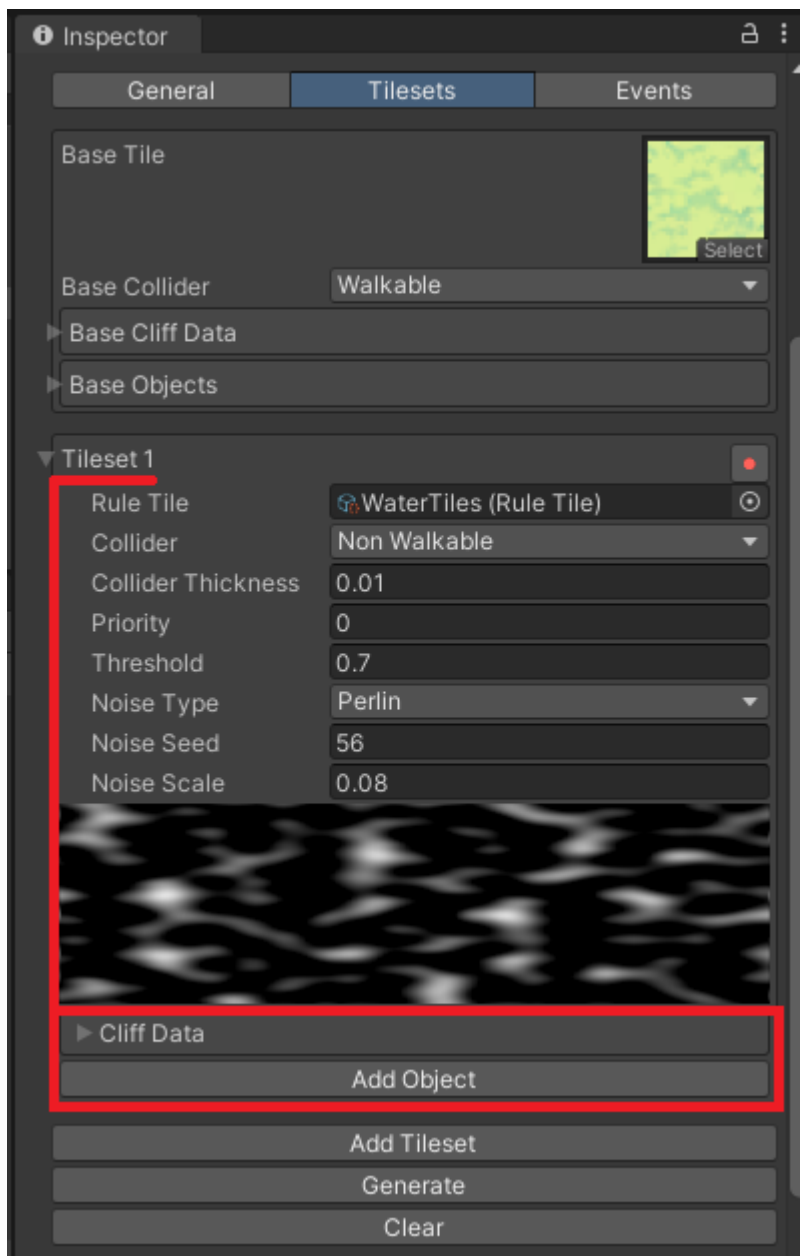
After we created our water rule tile, then we can click on **Add Tileset** and assign the rule tile. As shown below, when we set the settings of the noise and click generate, the water spots will be added to the tilemap. Note that we set the **Collider** of the tileset to **Not Walkable**.



Here are the example sprites for sand stone areas.



Basically it's the base tile that we draw on so any transition must be from base tile which in this case is grass to the additional tileset.



Similar to the base tile, other tilesets also have cliff data and objects list of their own. For example, you can use it if you want pine trees on your base tile and palm trees on your sand tileset and maybe water lilies on your water tileset the you can use these options.

Final Note:

If you want to create a player and move it around the world, make sure the sorting layer for its sprite render is 0 and also put the Transparency Sort Axis to (0, 1, 0) in your project.

Contact

You can contact me to ask questions, report bugs or request features using discord or email.

Discord: <https://discord.com/invite/FWwDgKjckQ>

Email: developershub.organization@gmail.com