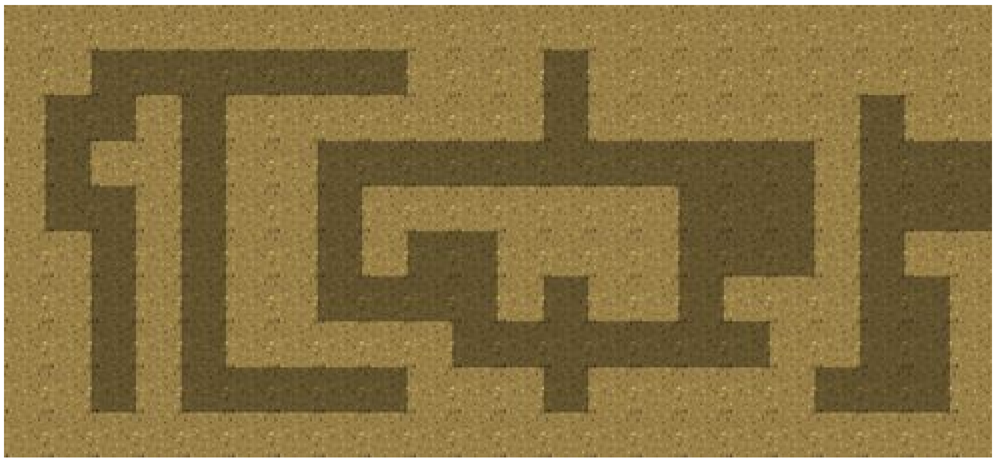
Advanced Rule Tiles Documentation & Guide

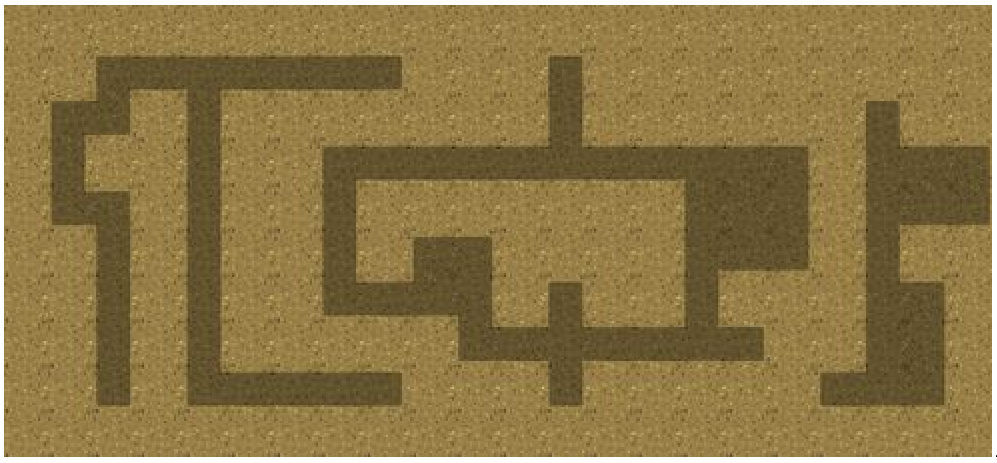
Contents

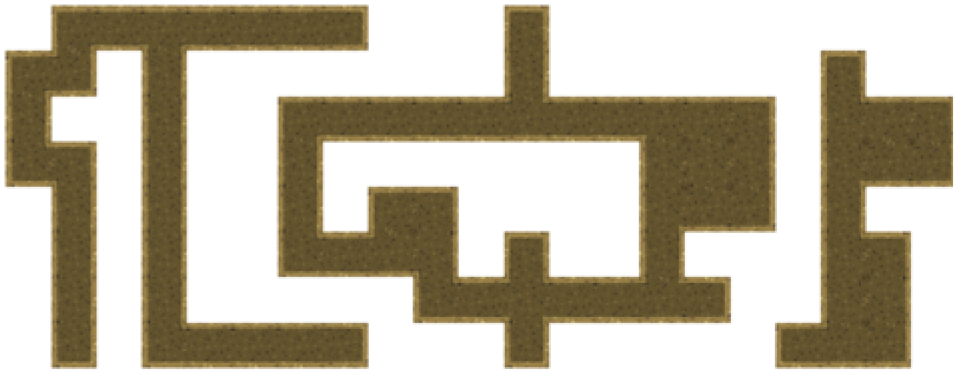
When and why to use Advanced Rule Tiles
Importing Advanced Rule Tiles into your project
Creating a new advanced rule tile
Customizing your advanced rule tile (no scripting required)
What to do after you make your advanced rule tile

When and why to use Advanced Rule Tiles

Rule tiles are used to change a tile's appearance based on the tiles around it. This will make your project look far more realistic and not so grid-like. It can also be used to create narrow paths that automatically connect visually.


Before

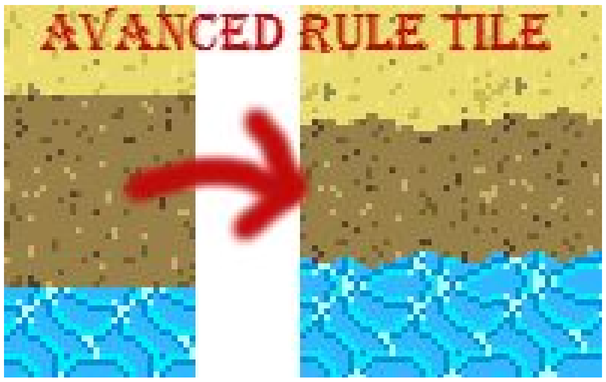

After


Without Dirt Tiles

Whether you're designing a tile-based map, randomly generating one, or allowing players to edit tiles, it's a good idea to use rule tiles instead of several different tiles for the same substance. Using a rule tile instead of several tiles will drastically reduce the time and effort it takes to design levels or randomly generate them, and it will make it a breeze for players to change tiles.

Additionally, having all the textures rolled into one tile will make scripts used for detecting tiles much shorter.

Advanced rule tiles allow you to test the tiles around them and automatically change to the texture best suited. You can use these features to blend together tile textures, such as to make sand meet water smoothly.



Importing Advanced Rule Tiles Into Your Project

If you're reading this, you probably already have Advanced Rule Tiles in your project. For every project you want to include Advanced Rule Tiles in, you only need to import the RuleTile.cs and RuleTileEditor.cs files. The RuleTileEditor.cs file needs to stay in the Editor folder.

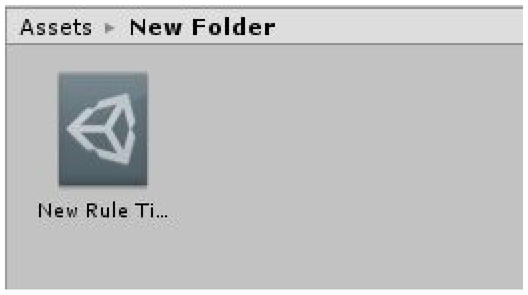
Important: If you already have the regular Rule Tiles scripts from 2d-extras in your project, you should replace them with the Advanced Rule Tile scripts. Don't worry, all of your old rule tiles will still work the same. Any rule tiles will automatically convert to Advanced Rule Tiles, but they will keep their current functionality.
If you replace the Advanced Rule Tile scripts with 2d-extras' Rule Tile scripts, you will lose any changes you made that affect advanced rules, but everything else will be kept for each rule tile.

Creating a New Advanced Rule Tile

Note: if you already have rule tiles in your project, they will automatically become advanced rule tiles so there's no hassle in converting them over.

In order to create an advanced rule tile, simply right-click in the assets area, which shows all your textures, assets, folders, et cetera - this is not the same as the hierarchy tab.
Next, left-click on Create>Rule Tile. You can also do this by going to the top of the project editor and clicking Asset>Create>Rule Tile.

Now the new rule tile will appear in your assets.

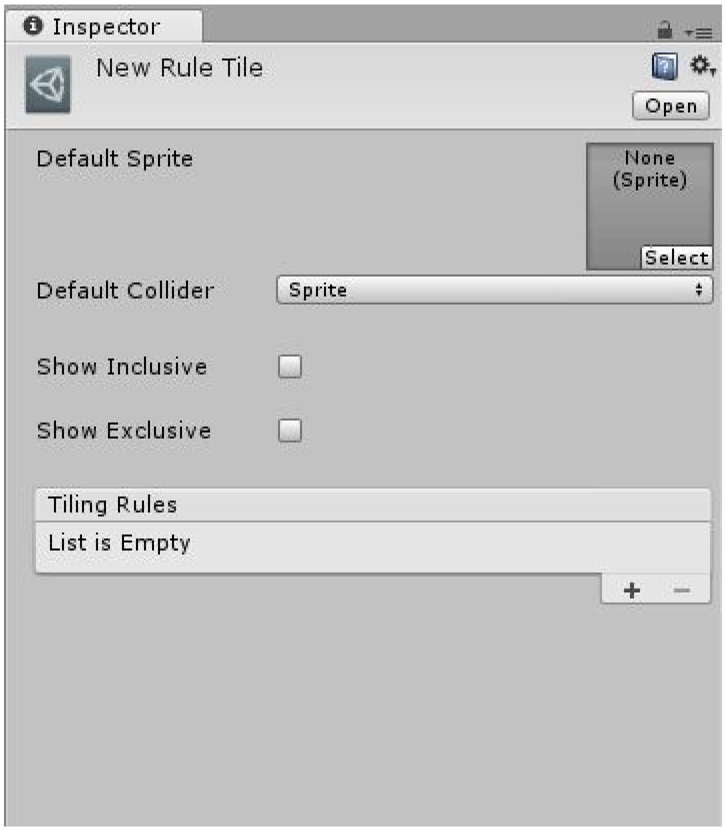


Customizing your advanced rule tile

(no scripting required)

If you're not yet familiar with rule tiles, you should watch the Unity tutorial first (linked below), before learning how to use advanced rule tiles.
https://unity3d.com/learn/tutorials/topics/2d-game-creation/using-rule-tiles-tilemap

To customize your rule tile, left-click on it in the asset manager and make sure the inspector tab is open. It should look something like this:



Default Sprite

Default Collider: Sprite

Show Inclusive

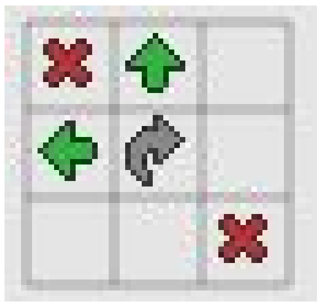
Show Exclusive

Tiling Rules

List is Empty

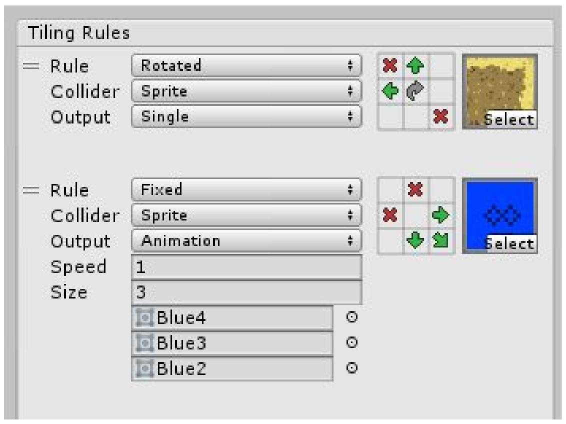
You should start by assigning a Default Sprite by clicking the [Select] button in the top right. This sprite will be shown whenever none of the tiling rules apply.

By now, you should have thought about what tiles will be in the vicinity of your rule tile and what textures you want to include to react to its surroundings.



Regular tiling rules will only check if each of the 8 neighboring tiles is the same type of rule tile(←↑→↓), is NOT the same type of tile (X), or either-or ( ). The middle of the grid will show whether the tiling rule is **fixed** (where the 8 neighbors are only checked in the exact rotation shown), **rotated** (where the rule is checked 4 times to see if it's true at @ 0°, 90°, 180°, and -90° rotation), **mirror x** (where the rule is checked twice, the second time reflected across the Y-axis), or **mirror y** (where the rule is checked twice, the second time reflected across the X-axis).

The first tiling rule (from top to bottom) to have all 8 neighboring tiles to follow their Same-NotSame-Either ruling will have its corresponding sprite shown for that tile. If the Output is set to animation, it will cycle through the sprites associated with that tiling rule. If the Output is set to random, it will be set randomly to one of the sprites associated with that tiling rule.



Tiling Rules

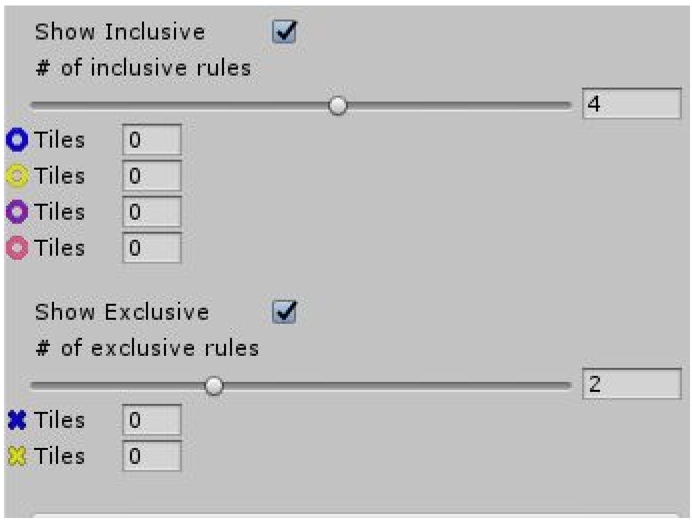
Rule: Rotated
Collider: Sprite
Output: Single

Rule: Fixed
Collider: Sprite
Output: Animation
Speed: 1
Size: 3
Blue4
Blue3
Blue2

When you are using Advanced Rule Tiles, you can add up to 13 additional rules. These will be in the same grid as the Same-NotSame-Either grid for each tiling rule. You'll still have to left-click on the small squares to cycle through the rules, however, you will not have to click through extra rules if you have not included them.

For each rule tile, you can have up to 7 different **inclusive** rules (represented by colored circles). You may also have up to 6 different **exclusive** rules (represented by colored X's).

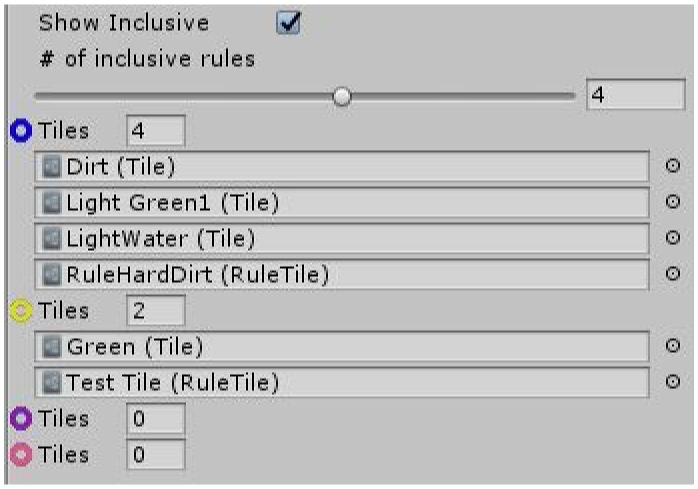
Increase the number of each rule type by checking [Show Inclusive] & [Show Exclusive] and using the slider to set the number of rules.



Show Inclusive ✓
# of inclusive rules 4
○ Tiles 0
○ Tiles 0
○ Tiles 0
○ Tiles 0

Show Exclusive ✓
# of exclusive rules 2
✖ Tiles 0
✖ Tiles 0

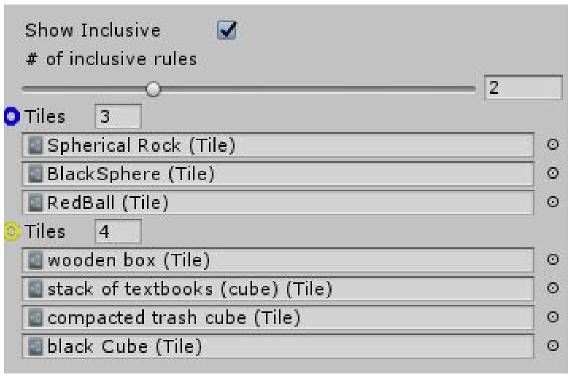
You'll notice that a row will be added for each rule where you can see the icon that will represent that rule. Each rule will also have a number input for the number of tile-types you'd like to include in that rule. When attaching tiles, they can be rule tiles, tiles, animated tiles, random tiles, or any other type of tile inherited from the TileBase class.
You can have an unlimited number of tiles included in each rule.
To prevent accidentally removing all of the sprites attached to a rule, you will have to click on another inspector element to update after changing the number.



Show Inclusive ✓
# of inclusive rules 4
○ Tiles 4
Dirt (Tile)
Light Green1 (Tile)
LightWater (Tile)
RuleHardDirt (RuleTile)
○ Tiles 2
Green (Tile)
Test Tile (RuleTile)
○ Tiles 0
○ Tiles 0

Inclusive rules will return true if that neighbor matches any of the tiles attached.
Exclusive rules will return true if that neighbor does **not** match **any** of the tiles attached.

To help you understand, here's an example: Imagine you have three types of tiles that display spheres and four types of tiles that display cubes. Assume you want all of your grass tiles to display 1. A circular shadow if the tile to the right of it is a sphere, 2. A square shadow of the tile to the right of it is a sphere, 3. No shadow if there is neither a cube or a sphere to it's right.



Show Inclusive ✓
# of inclusive rules 2
Tiles 3
Spherical Rock (Tile)
BlackSphere (Tile)
RedBall (Tile)
Tiles 4
wooden box (Tile)
stack of textbooks (cube) (Tile)
compacted trash cube (Tile)
black Cube (Tile)

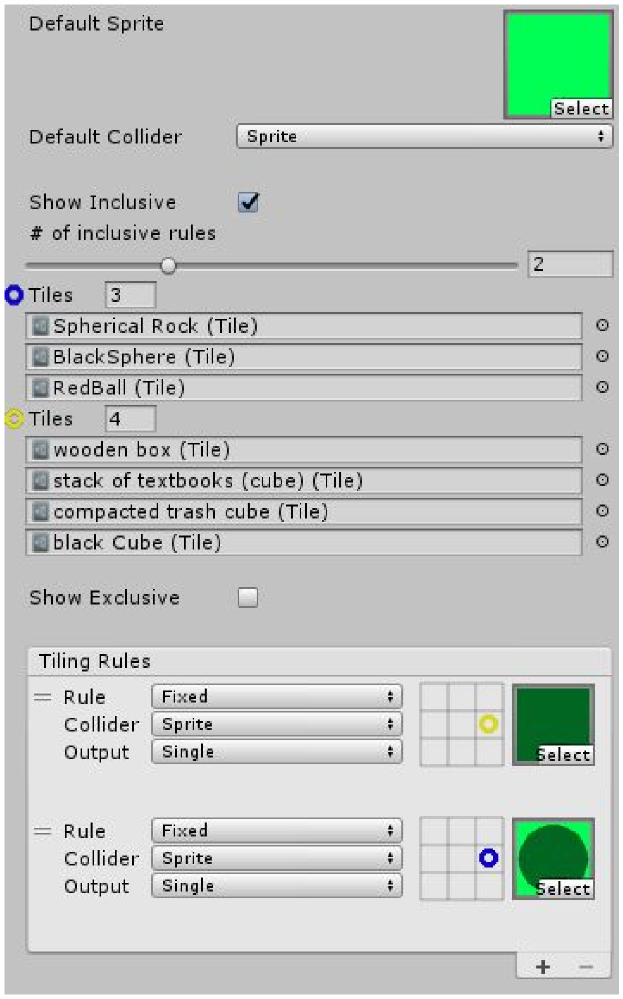
You would use two inclusive rules: one with three tiles and one with four tiles. In the slots for the first one, you would put all the tiles that are spheres. In the slots for the second one, you would put all the tiles that are squares.
In order to check what our grass is next to, we'll create two tiling rules.

In the first tiling rule, we will test if the tile to the right of the grass tile is a included in the yellow inclusive rule. If it is, the grass' texture will be set to the dark green to represent a shadow.

If the first tiling rule returns false, we go on to the second.

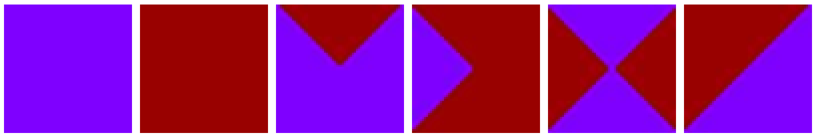
In the second tiling rule, we will test if the tile to the right of the grass tile is a included in the blue inclusive rule. If it is, the grass' texture will be set to light green with a dark circle to represent a circular shadow.

If both rules are false, then the default sprite is used and the grass will appear light green - aka without a shadow.

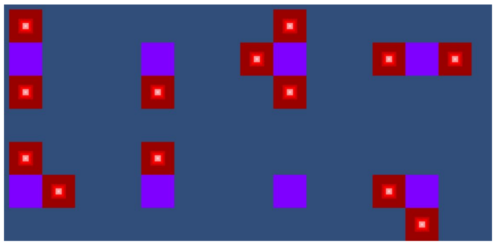


Default Sprite
Default Collider Sprite
Show Inclusive ✓
# of inclusive rules 2
Tiles 3
Spherical Rock (Tile)
BlackSphere (Tile)
RedBall (Tile)
Tiles 4
wooden box (Tile)
stack of textbooks (cube) (Tile)
compacted trash cube (Tile)
black Cube (Tile)
Show Exclusive
Tiling Rules
Rule Fixed
Collider Sprite
Output Single
Rule Fixed
Collider Sprite
Output Single

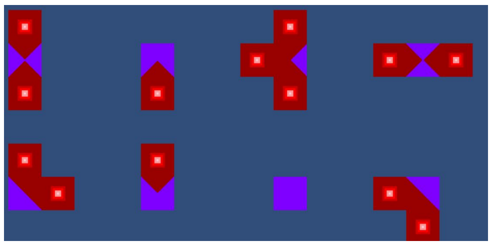
For another example: Imagine you have blue and red tiles, and you want to make a tile that is solid purple unless there are red tiles around it. Pretend you want to split the purple tile into four so that any side that is adjacent to a red tile will be red. You will need at least six textures: One that is solid purple, one that is solid red, one for when there is **one** adjacent red tile, one for when there is **three** adjacent red tiles, and two different textures for when there are **two** adjacent red tiles.



The end goal is to have this:

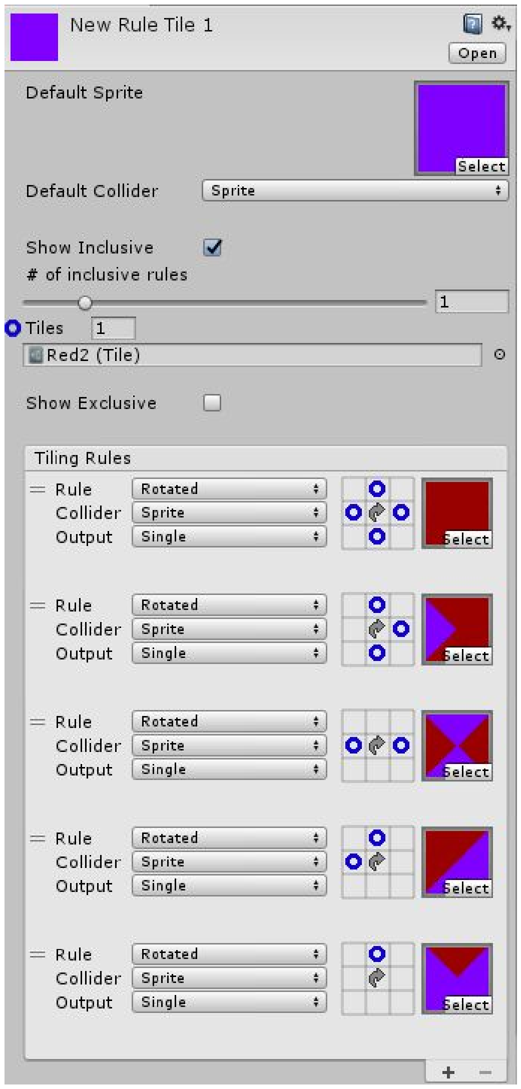


Become this:



The solution is shown here. Try to determine how it works on your own.

Note that because the first tiling rule to be true is chosen, you won't necessarily need exclusive rules to tell where red isn't in this example.



New Rule Tile 1

Open

Default Sprite

Select

Default Collider: Sprite

Show Inclusive ✓

# of inclusive rules: 1

Tiles: 1

Red2 (Tile)

Show Exclusive

Tiling Rules

Rule: Rotated
Collider: Sprite
Output: Single

Rule: Rotated
Collider: Sprite
Output: Single

Rule: Rotated
Collider: Sprite
Output: Single

Rule: Rotated
Collider: Sprite
Output: Single

Rule: Rotated
Collider: Sprite
Output: Single

What To Do After You Make Your Advanced Rule Tile

Like any other type of tile, you must first have a tilemap put into your scene. Then you may drag your Rule Tile from the asset tab into the #scene tab. You can also drag the Rule Tile into a tile palette in the tile palette tab. (You can open the Tile Palette tab by mousing over the [Window] tab at the top of the editor and clicking on [Tile Palette])

Additionally, you can add the rule tile into a tilemap using scripts found in Unity's Scripting API.