Procedural Maze Generator

Setting up the grid

Auto Grid:

1. Open the template scene provided with the package and select the Maze Generator object in the inspector.
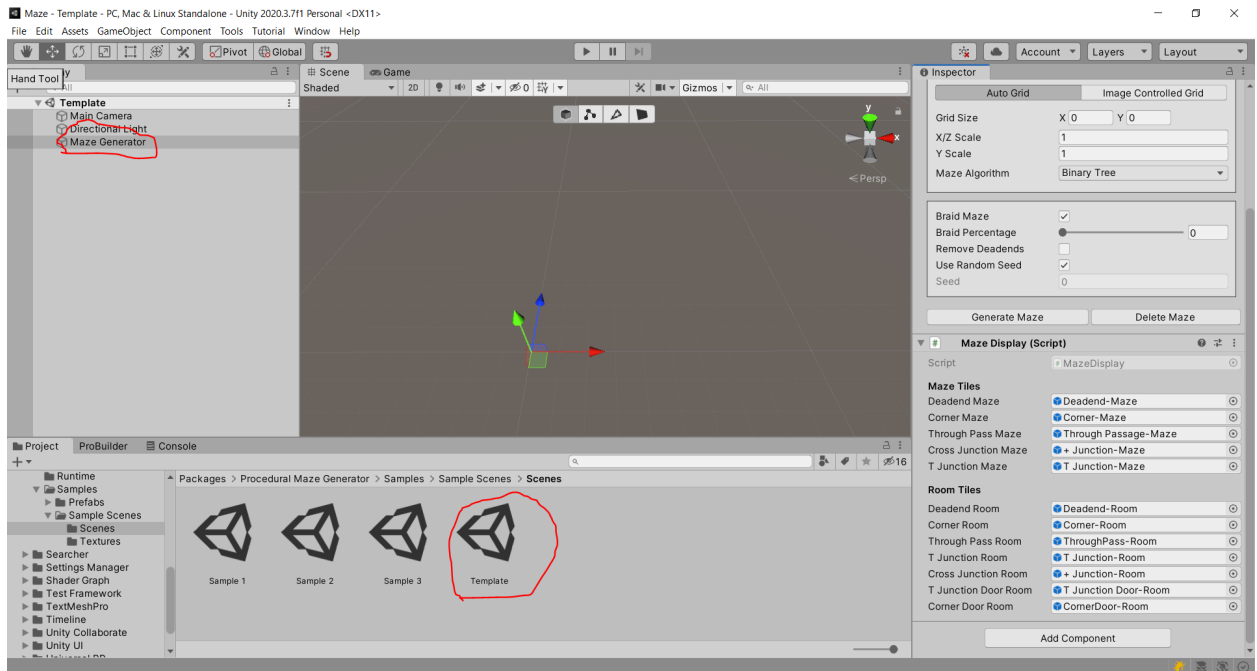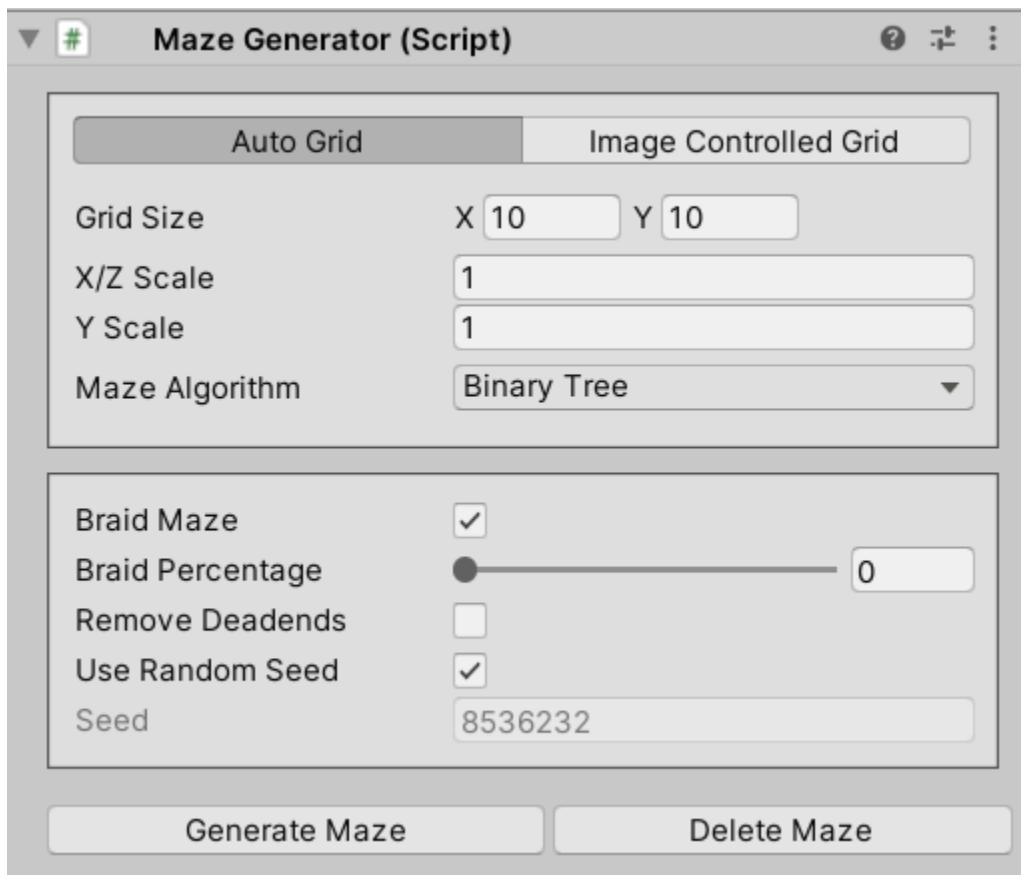

2. Copy this and paste it into the scene you wish to make a maze in. (Alternatively, create a new game object and add the MazeGenerator script to it. This will automatically add the MazeDisplay script but you will have to populate the tile fields for the MazeDisplay

script).
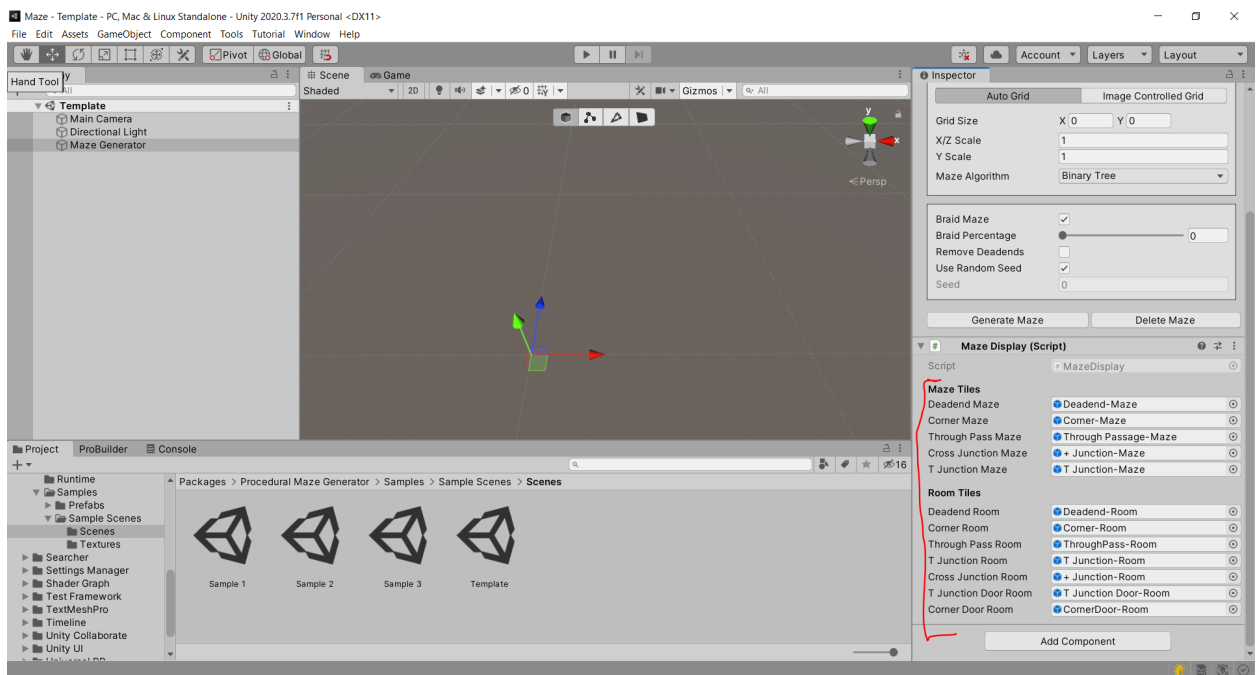
3. Select the AutoGrid tab.

4. Set the desired size of the grid.

5. Set the desired scale of each tile. An xz scale of 2 will mean the floor and wall objects are scaled to 2 on the x and z axis. The same goes for the y scale on the y axis.
6. Select the desired algorithm. For more information about the different algorithms, see [here](#).

Image Based Grid:

1. Create a new image in your photo editing software of choice, such as Gimp. Each colour used will represent a different layer/mask/section and will have a different algorithm run on it.
    a. The size of the image will be the size of the grid. So a 50 pixel by 50 pixel image will create a 50x50 maze (2500 tiles)
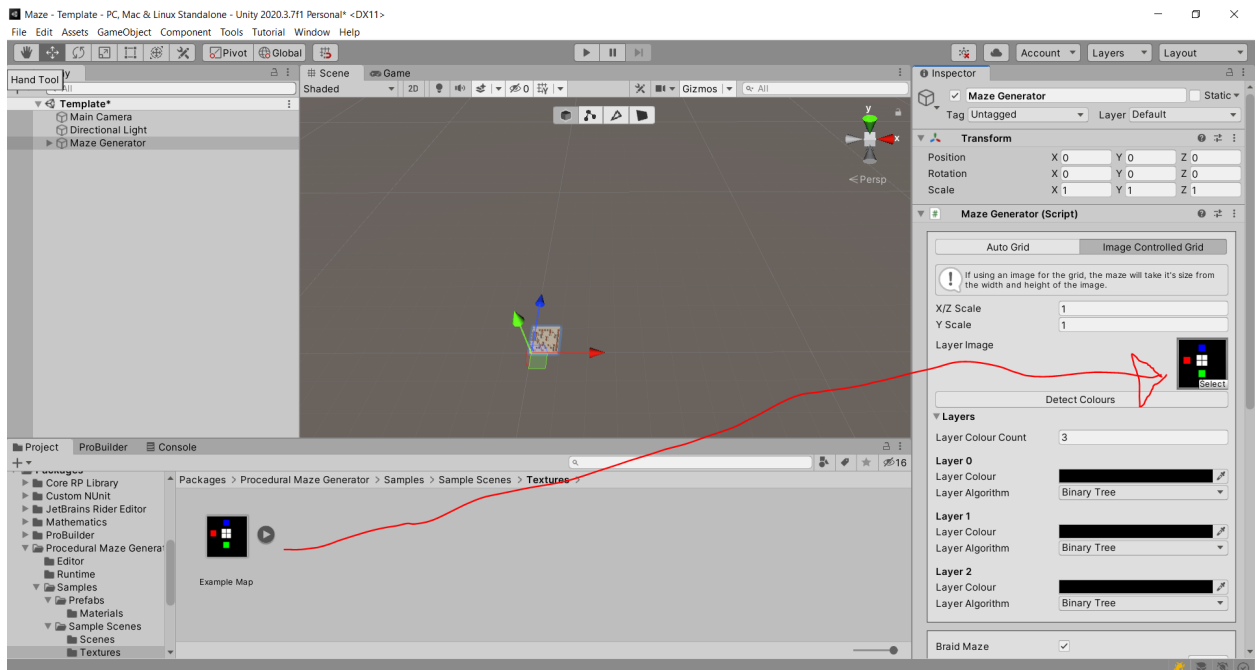

2. Export your image as a png into Unity.
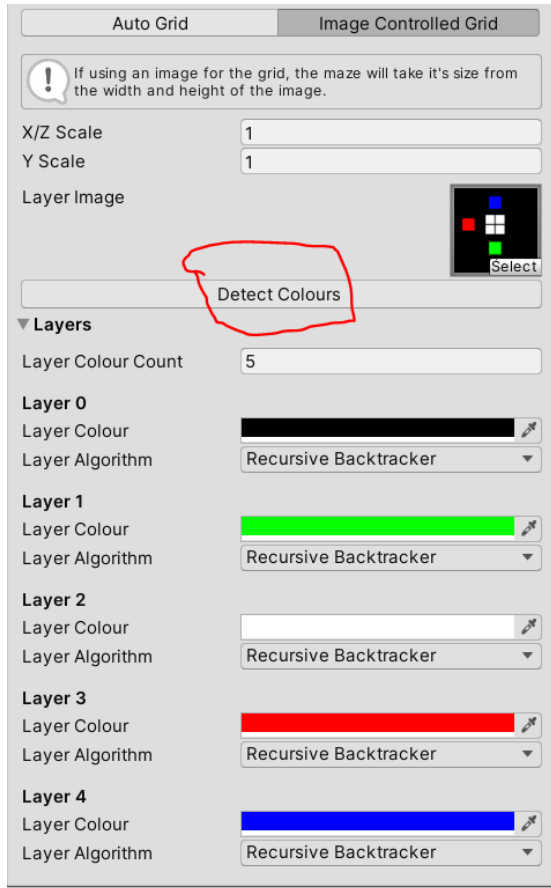
3. Set the texture type to Sprite (UI and 2D) and the read/write option to true.
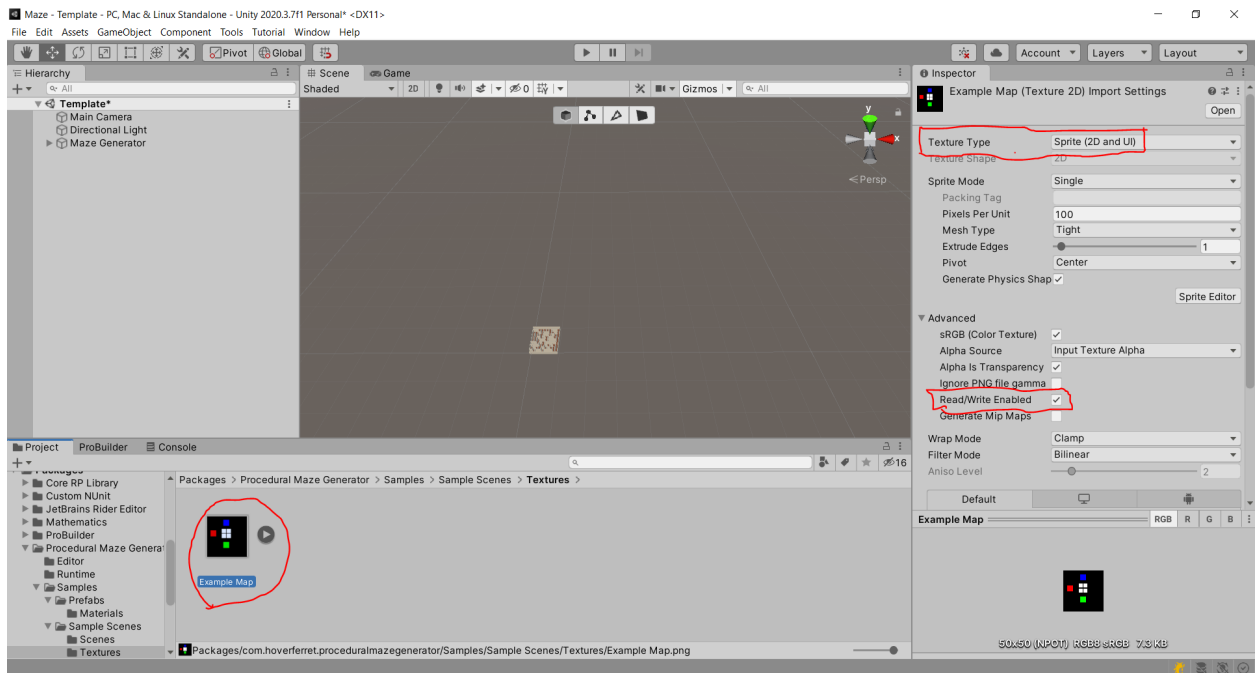
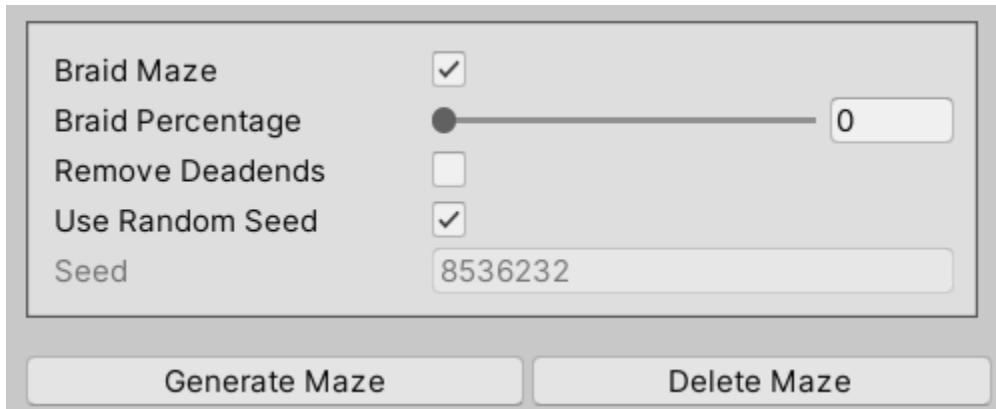4. Drag the texture into the texture field of the maze generator.

5. Click the Detect Colours button. This will populate the layers section.

Auto Grid	Image Controlled Grid

(!) If using an image for the grid, the maze will take it's size from the width and height of the image.

X/Z Scale	1
Y Scale	1

Layer Image

Select

Detect Colours

▼ **Layers**

Layer Colour Count	5

Layer 0
Layer Colour	⬛
Layer Algorithm	Recursive Backtracker ▼

Layer 1
Layer Colour	🟩
Layer Algorithm	Recursive Backtracker ▼

Layer 2
Layer Colour	⬜
Layer Algorithm	Recursive Backtracker ▼

Layer 3
Layer Colour	🟥
Layer Algorithm	Recursive Backtracker ▼

Layer 4
Layer Colour	🟦
Layer Algorithm	Recursive Backtracker ▼

6. For each colour/layer, select an algorithm to run. For more information about the different algorithms, see here.
   a. Warnings:
      i. there can only be 1 Ignore layer.
      ii. Each colour should be connected (except the colour used for Ignore Layer). As in the example texture, all of the black pixels are connected to other black pixels. The same goes for red, blue, and green. Only white is able to be separated because white is being used as the Ignore Layer.
7. After generating a maze, each layer/section will be connected to an adjacent one.

Advanced Settings

1. By default, the maze will be a perfect maze (only one path from any tile to any other tile. No loops). Braid maze will merge link some of the deadends with the neighbouring cells.
   a. After ticking the braid maze setting, you'll be able to choose the percentage of deadends that will be linked to their neighbours.

2. Remove deadends will remove deadend passageways.
   a. A deadend is defined as a tile with only one connection. Once one is found, it will be deactivated. The tile that linked to it will then be checked. If this is now considered a dead end, it will also be deactivated. This continues until the linked cell has more than 1 link.
   b. In a perfect maze (with no loops) this will remove the entire maze, so it is not recommended for this.
3. If Use Random Seed is selected, a different maze will be generated every time. If it is unselected, you will be able to enter your own seed.

Algorithms

Binary Tree:

Pros: Quick.
Cons: Biased towards mazes that lead North-East. The Northern and Eastern Sides are empty passageways.
Warning: May not work if the area is not square or rectangular.

SideWinder:

Pros: Quick.
Cons: Biased towards mazes that lead North. The Northern side is an empty passageway.
Warning: May not work if the area is not square or rectangular.

Aldous-Broder:

Pros: Unbiased, totally random maze.
Cons: Can take longer to generate for big mazes.

Wilson:

Pros: Unbiased, totally random maze.
Cons: Can take longer to generate for big mazes.

Recursive Backtracker:

Pros: Long twisty passageways. Quicker than Wilson/Aldous-Broder.
Cons: Less memory efficient (may be an issue if generating very large mazes.

Hunt and Kill:

Pros: Similar to recursive backtracker but potentially more memory efficient.
Cons: possible slower than Recursive Backtracker.

Create Room:

Connect all tiles in on the layer.

Ignore Layer:

Ignore the tiles with that colour.
Warning: there should only be one colour for Ignore Layer.