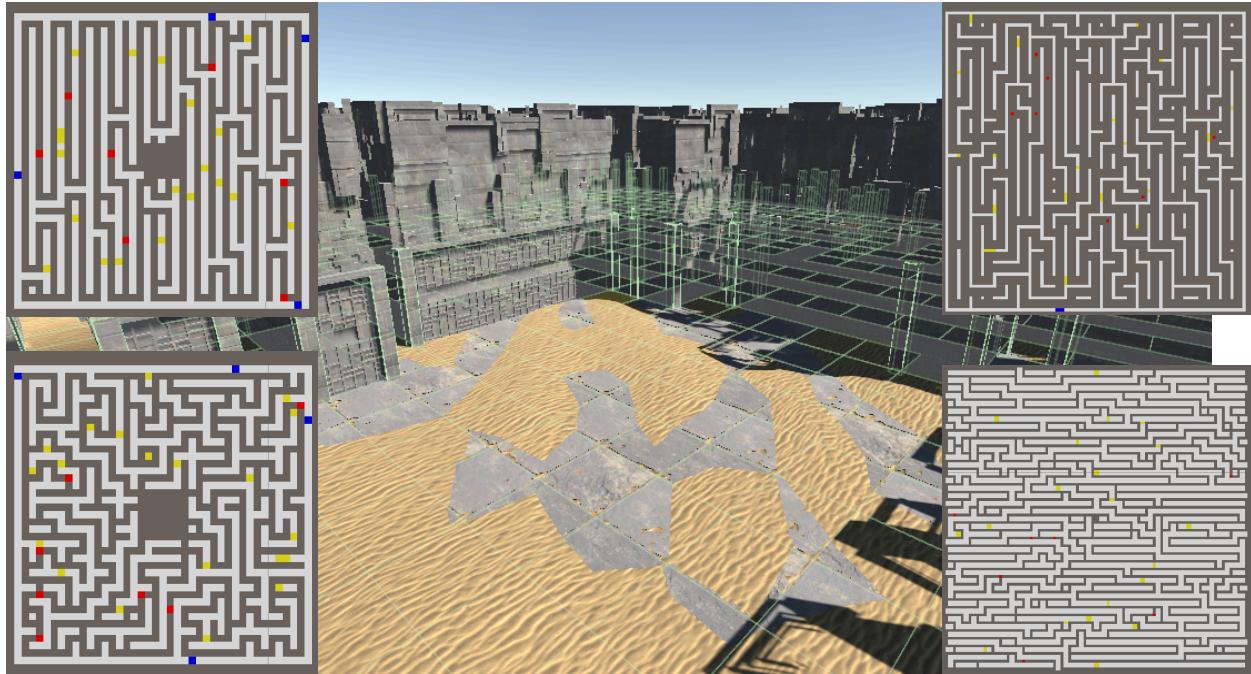


Maze Generator documentation

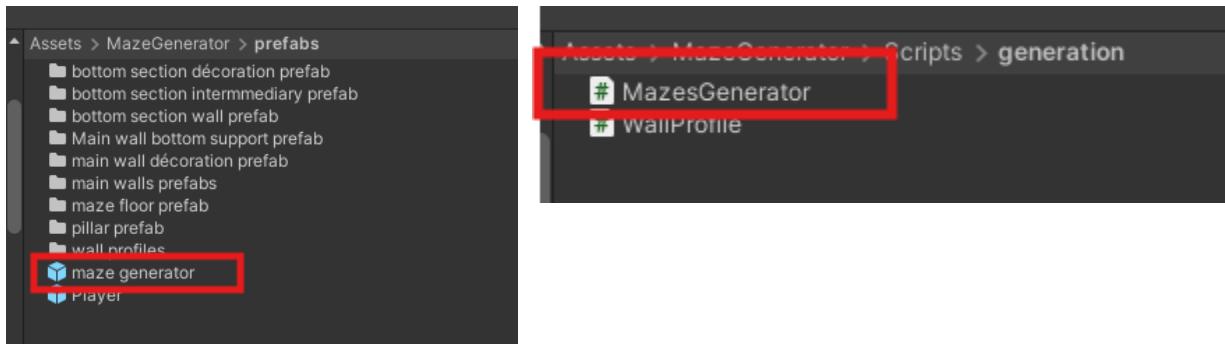


Summary :

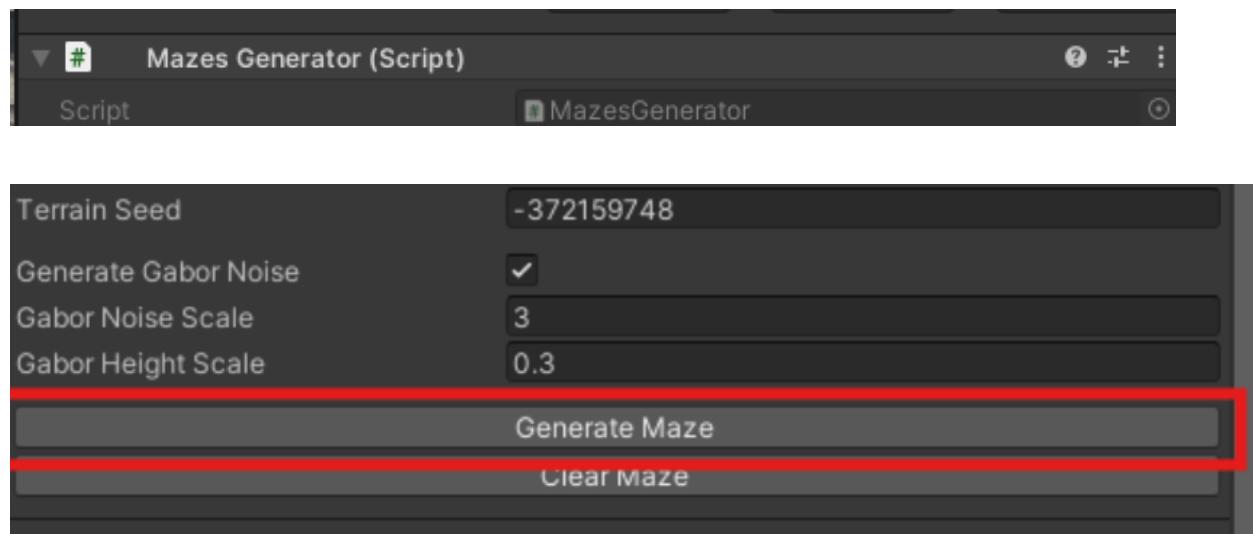
- 1 - Basic generation and player setup
- 2 - Main customization options
 - 2.1 - Dimensions
 - 2.2 - Maze options
- 3 - Mesh generation
 - 3.1 - Floors
 - 3.2 - Pillars
 - 3.3 - Wall profiles
 - 3.3.1 - Wall profile layers
 - 3.3.2 - Wall profile layers variants
 - 3.3.3 - Wall profile layers decorations
- 4 - Entrance/exterior walls removal
- 5 - Central empty zone
- 6 - Special floor cells
- 7 - Random wall passages
- 8 - Terrain generation

1. Basic generation and player setup

To get started, drag and drop the ‘maze generator’ prefab and the Player prefab into your scene, or simply open the example scene. If you prefer to configure everything from scratch, you can add the `MazesGenerator` script to an empty GameObject (detailed setup instructions in the sections below)



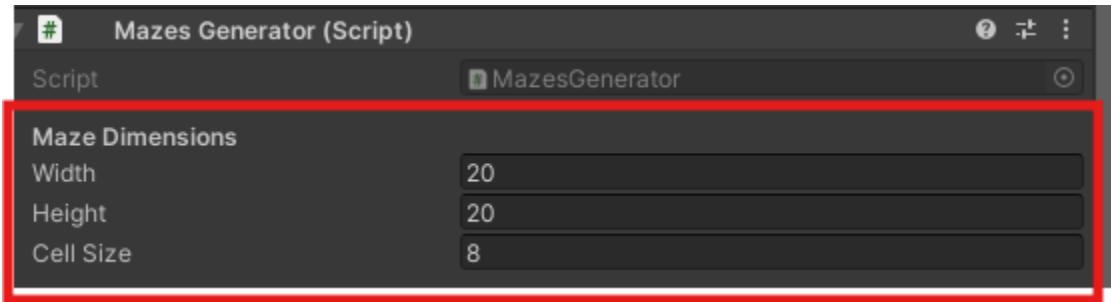
Once you have the configured prefab, you can just click the generate button in the editor after selecting the ‘maze generator’ prefab containing the script



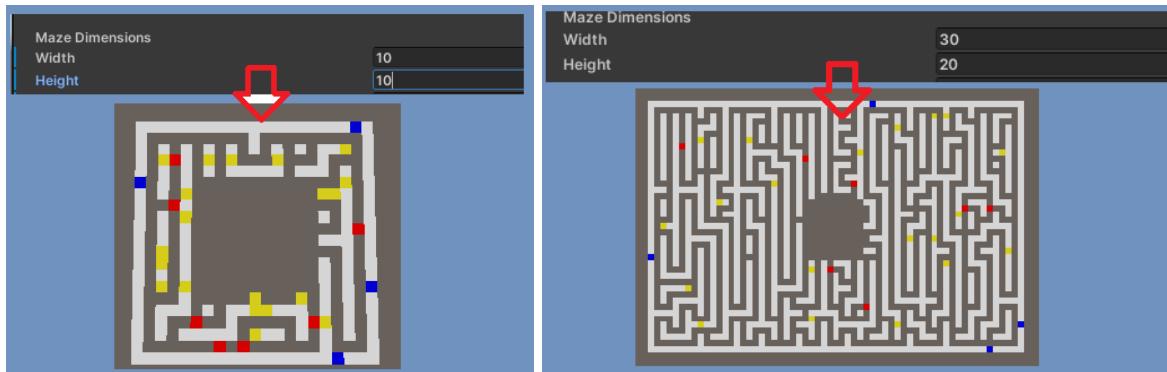
2. Main customization options

2.1 - Dimensions

On the top of the `MazesGenerator` script in the editor you can see the main options to customize the maze shape, first the options to define the maze dimensions :



Changing height and width define the maze size, this can be changed without needing to modify anything else



The ‘CellSize’ setting scales each cell of the maze grid. Changing it also requires adjusting the scale of the meshes, I recommend keeping the default value of 8 at first which matches the size of the example maze walls and floors.

If you still want to reduce or increase the cell size you will have to rescale the prefabs in the “Maze environment prefab” directory accordingly or to use your own scaled prefabs

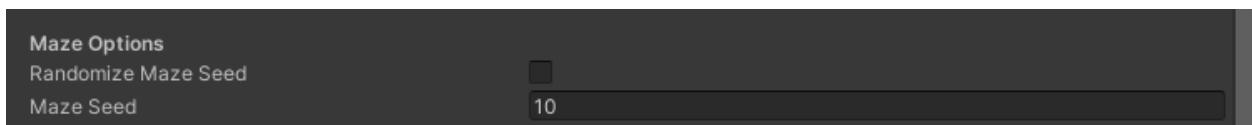
Directory : Assets\DsproPulse\MazeGenerator\Prefabs\Maze environment prefabs

2.2 - Maze options

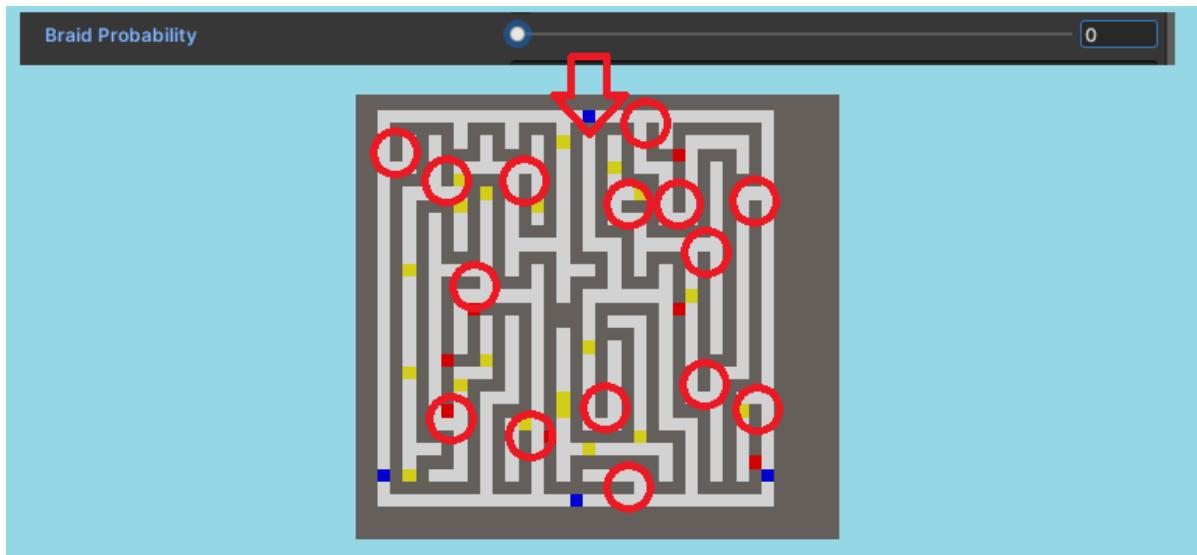
Below the dimensions are the maze options :



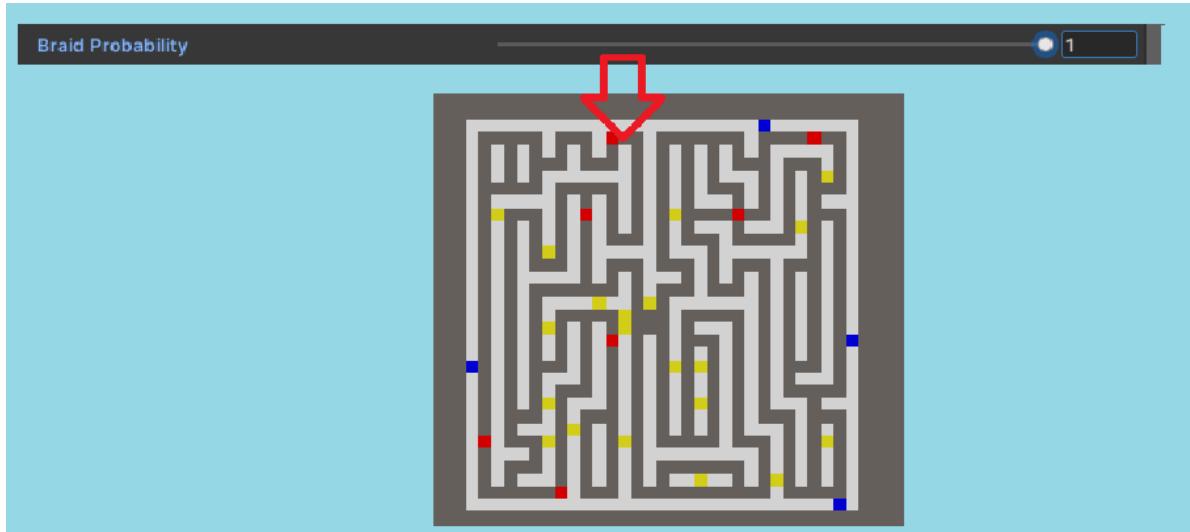
The seed can be manually set, or you can tick the '*RandomizeMazeSeed*' boolean to make it random, a similar value will make the global layout similar but will still randomize the rest (special tiles/entrance placement, meshes etc.)



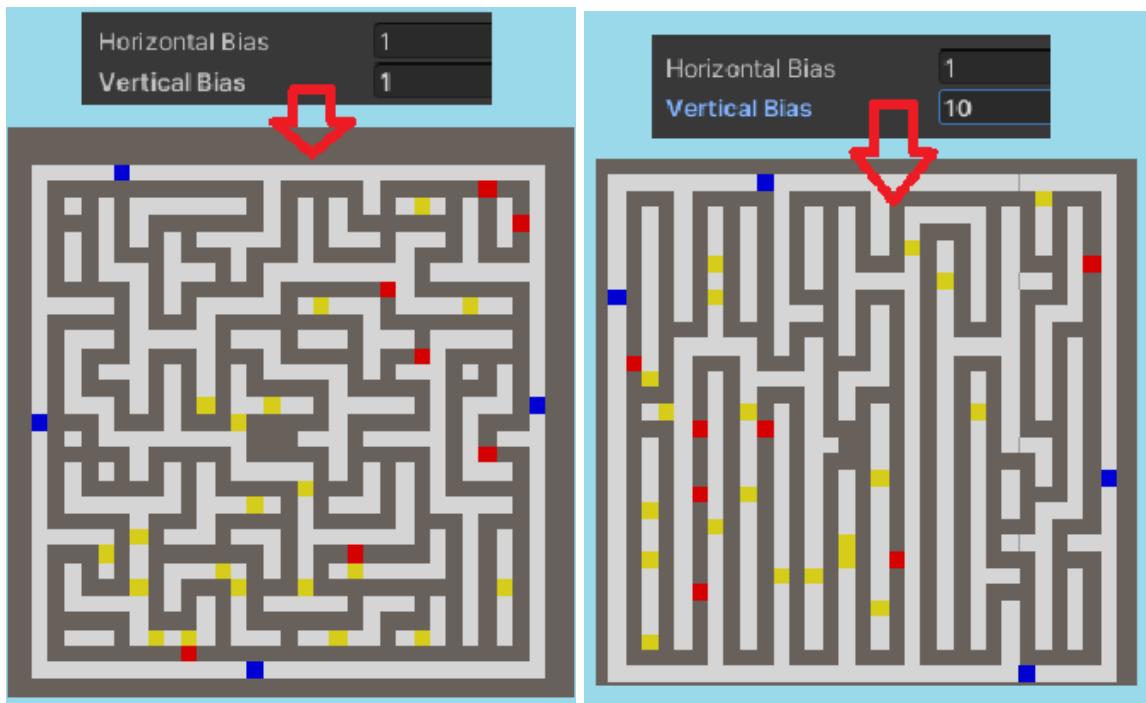
The boolean and slider to braid dead ends allow to reduce the number of dead ends, a value of 0 will mean no braiding so the maze will have many dead ends :



And increasing the value increases the number of braids and reduce dead ends :

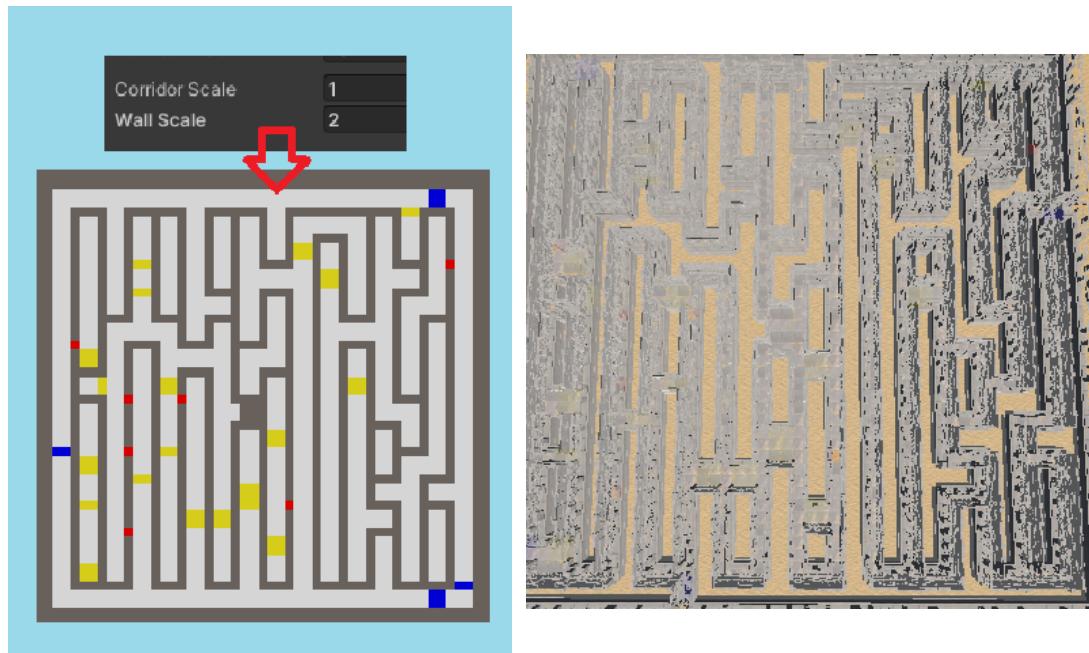


Vertical and horizontal bias define how likely the maze is to be oriented in one direction, so values of one will make a maze with a lot of turns, and increasing values in one way will increase corridors length in this way



Corridor and wall scale define how large the corridors/walls are, a value of 1 means that the grids for corridors and walls are the same size, changing the values to for example

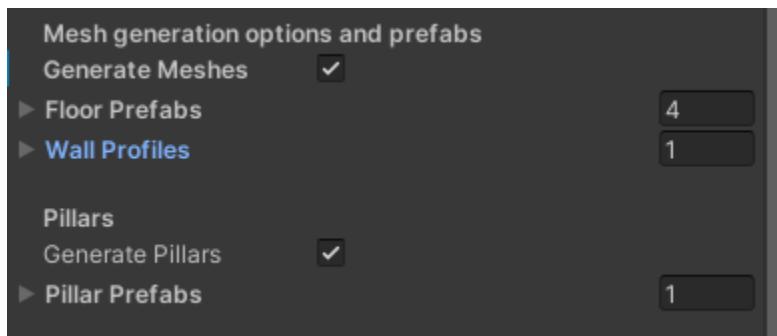
1:2 instead of 1:1 would make walls twice as large as corridors, and it also increases the global maze size



3. Mesh generation

Below the general option are the mesh options, the maze meshes are composed of the following components :

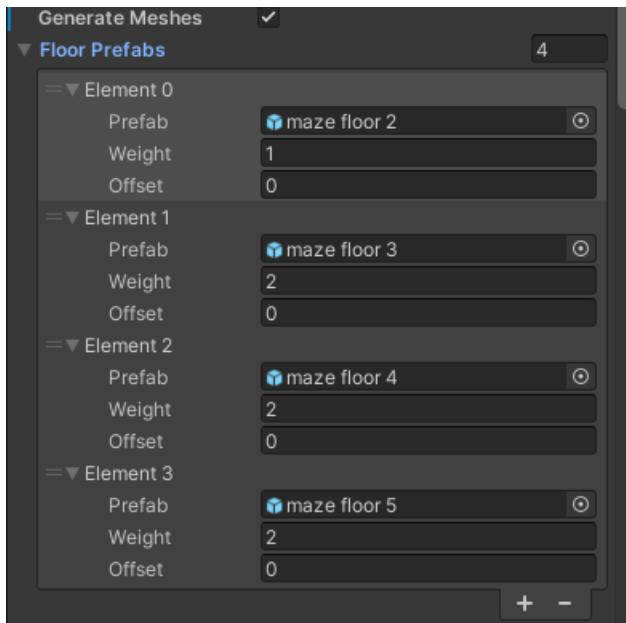
- floor
- pillars
- walls



If 'GenerateMeshes' is not enabled, only gizmos will be drawn (no meshes are spawned)

3.1 - Floors

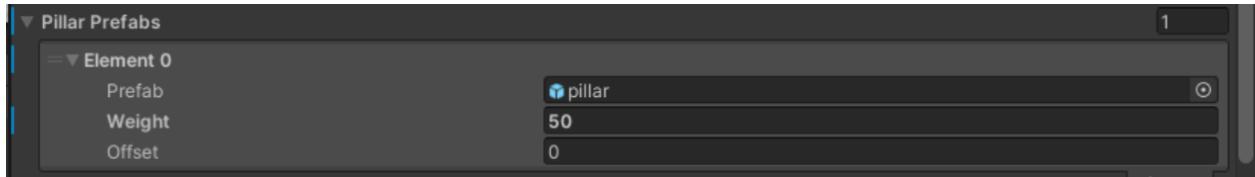
The floors are relatively straightforward: just add floor prefabs to the list and set their weight and offset. Weights act as relative probabilities. For example, if you have 2 items both with weight 2, they each have a 50% chance of spawning. If one has weight 1 and the other has weight 2, the second will spawn twice as often ($\approx 66\%$ vs 33%).



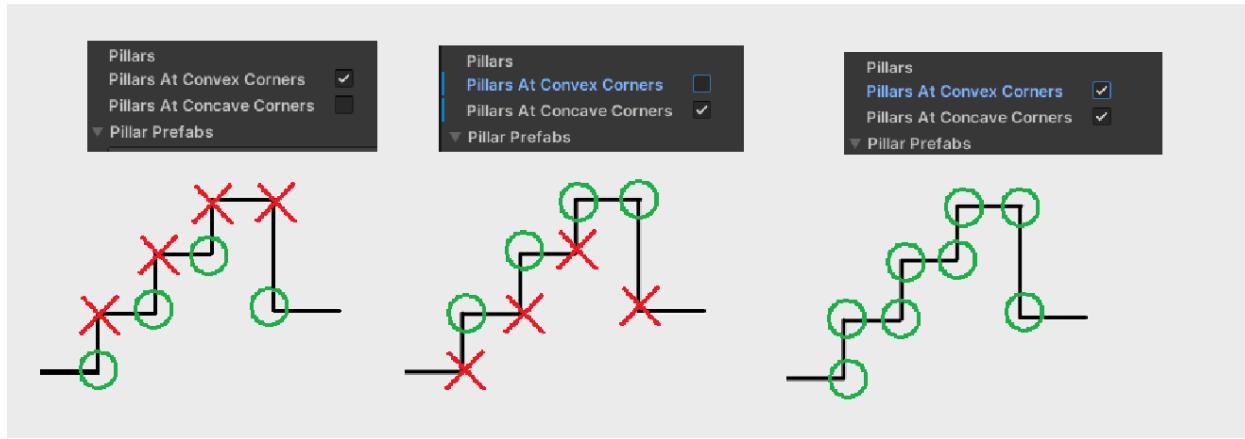
For all list containing a weight the weight is not at least 1 the prefab wont spawn

3.2 - Pillars

The pillars work in a similar way to floors, when generating a random prefab is picked depending on its weight (in the example there is only one prefab so only this one is picked)



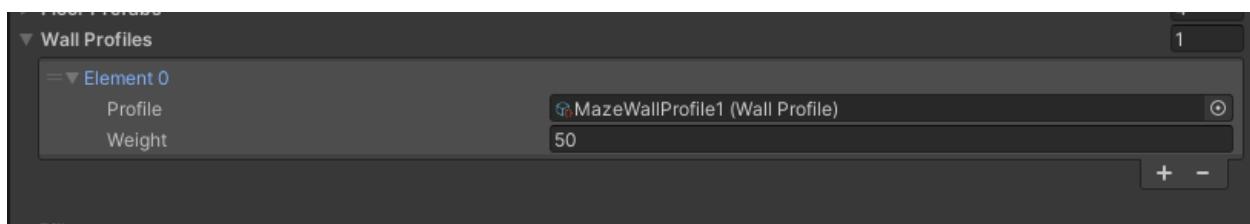
there are also two booleans allowing to choose whether pillars spawn only on outward/convex/external corners or inward/concave/internal corners, or both/neither



3.3 - Wall profiles

Walls are the most complex and customizable part of the mesh generation

Instead of choosing randomly between prefabs, it chooses randomly between wall profiles :

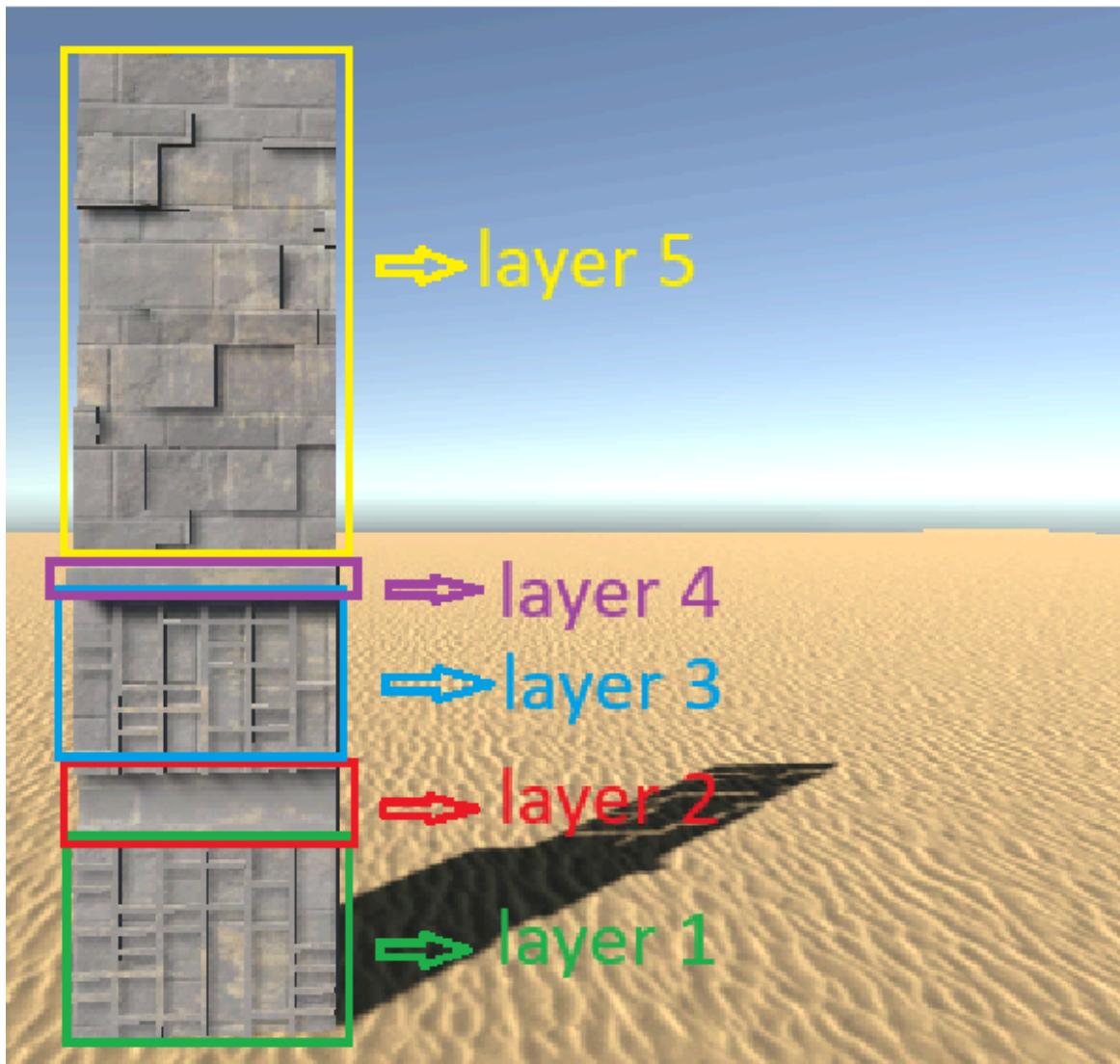


3.3.1 - Wall profile layers :

Wall profiles are made of layers



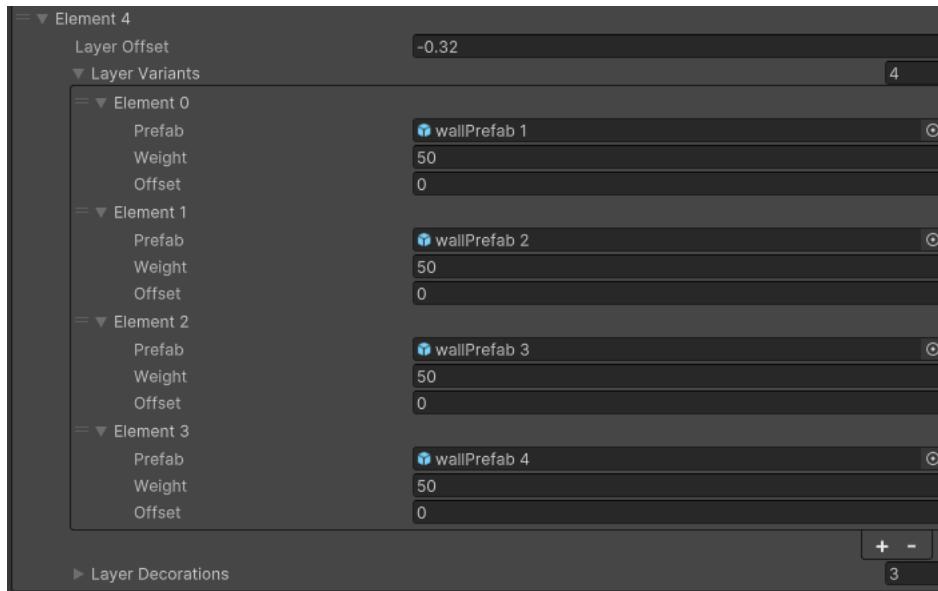
As you can see in the above example the wall profile is made of 5 elements (layers), each one spawning on top of each other like so :



3.3.2 - Wall profile layers variants :

Each layer is made of variants (one is randomly chosen by weight like the floors and pillars)

In this example, element 4 of the wall layers list (which corresponds to layer 5, since indexing starts at 0) is composed of 4 wall variants. Each has the same weight value, so they each have a $\frac{1}{4}$ chance of spawning.

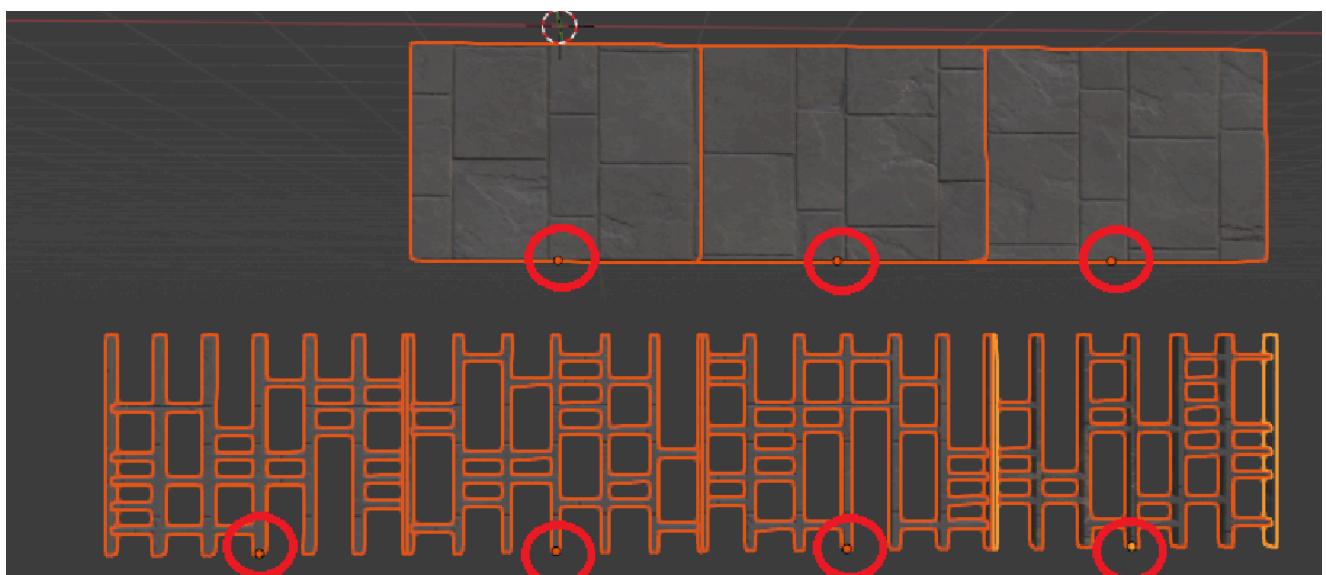


and in the previous example wallPrefab 3 was chosen :

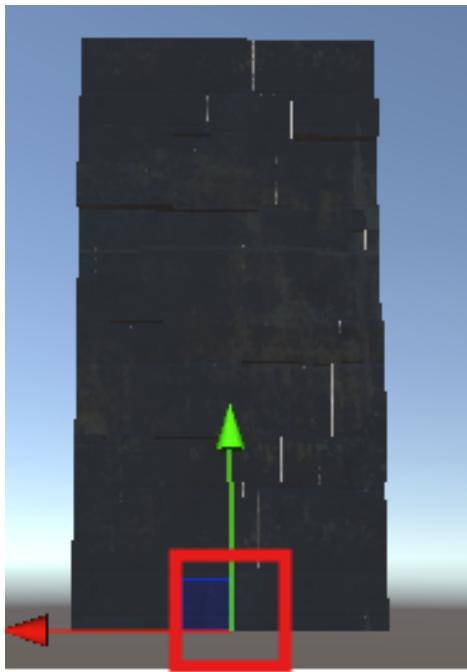


In order for each layer to correctly spawn on top of the previous one, I set the origin to the bottom center of each variant's mesh in Blender when modeling them.

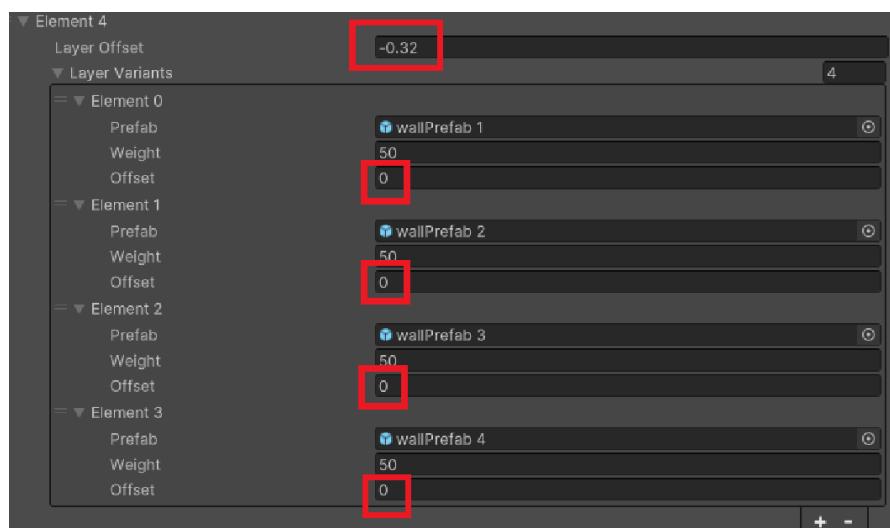
Screenshot of the variant's origin in blender :



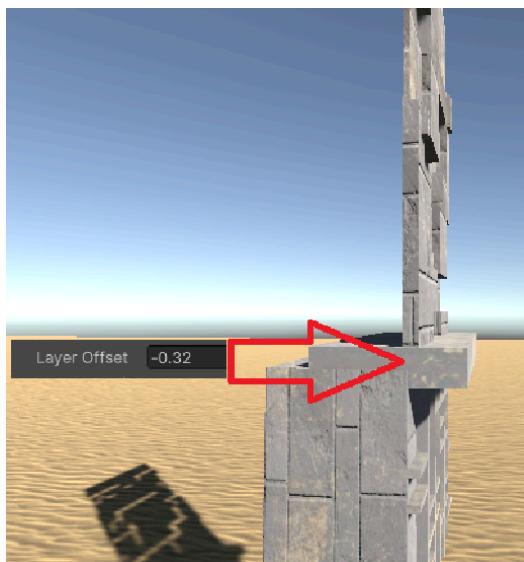
It can also be done in unity by putting a mesh under an empty object and moving the mesh up until the parent origin is at the bottom center of the children mesh :



If needed there is also an offset value in each variant and in each layer to set an offset either to individual variants or to the full layer, which I used in the example scene between layer 4 :

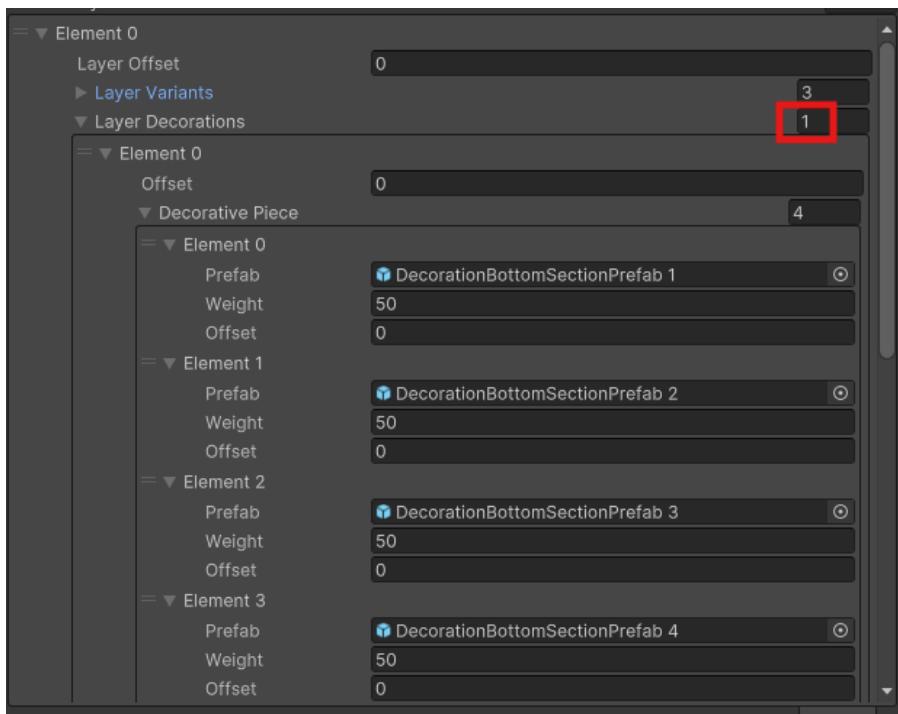


Offset example :

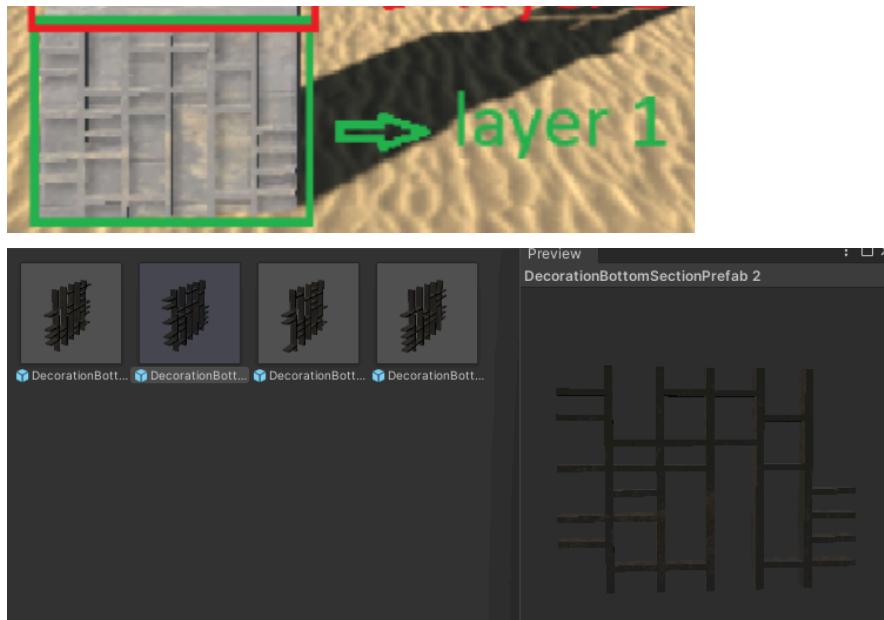


3.3.3 - Wall profile layers decorations:

Below the variants field is the decorations field, it works by spawning additional sub-layers within each layer. Each decoration has a weight that defines its chance of being chosen, allowing you to add more variety to a layer without extra variants modeling.



In this example layer 1 have one decoration layer composed of 4 decorations elements that can be chosen :



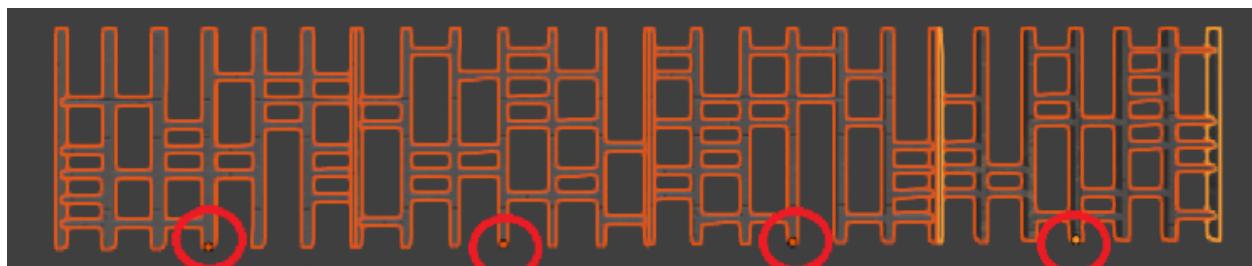
Without decorations :



With decorations :



Their origin also needs to match the layers to spawn correctly



Since the layer 1 has only one decoration layer its the easiest example to illustrate how decorations work, but as another more complex example the layer 5 is composed of 3 decoration layers each composed of multiple pieces :

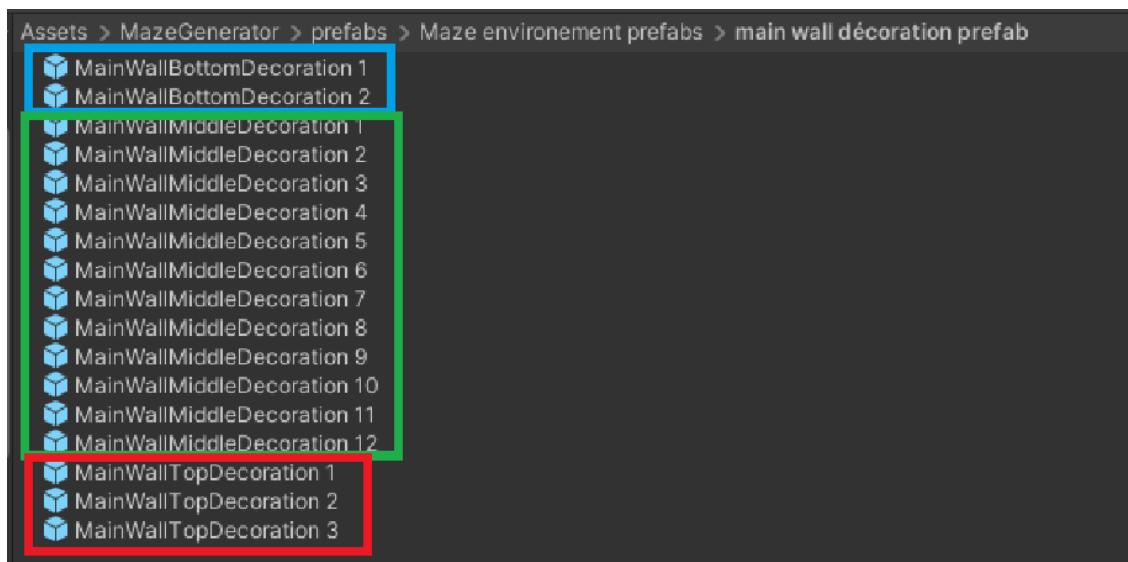


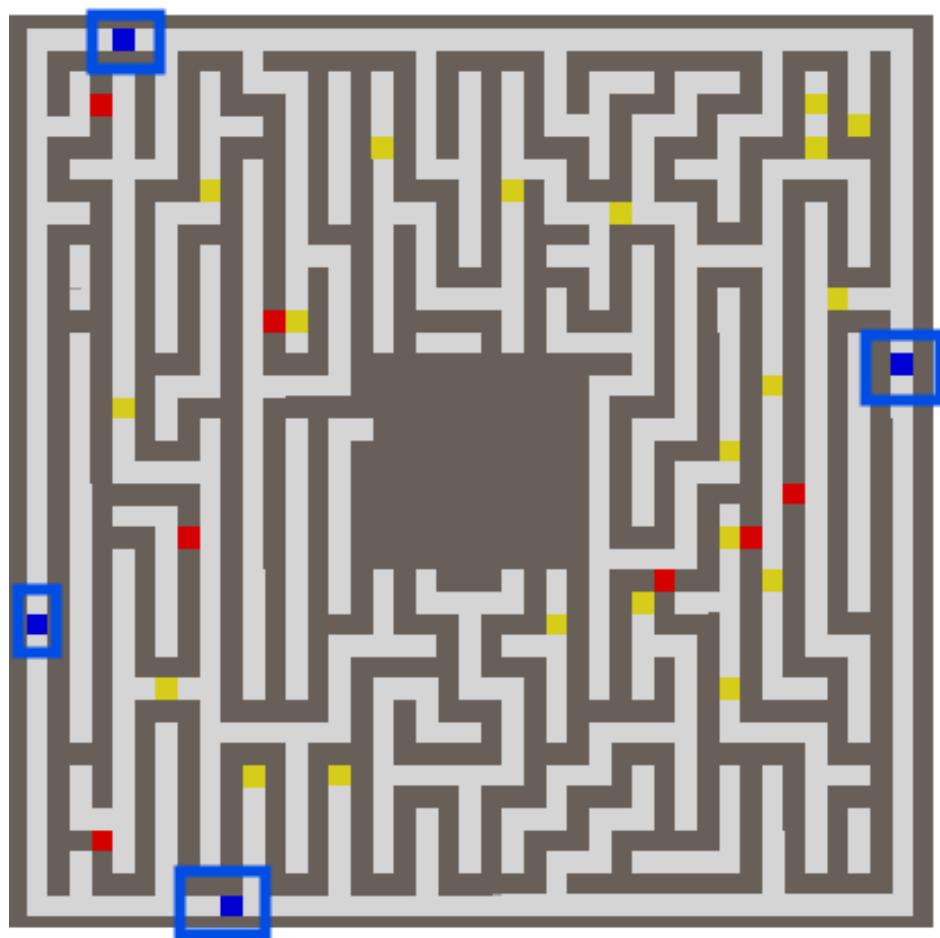
Illustration of top/bottom/middle decorations colored :

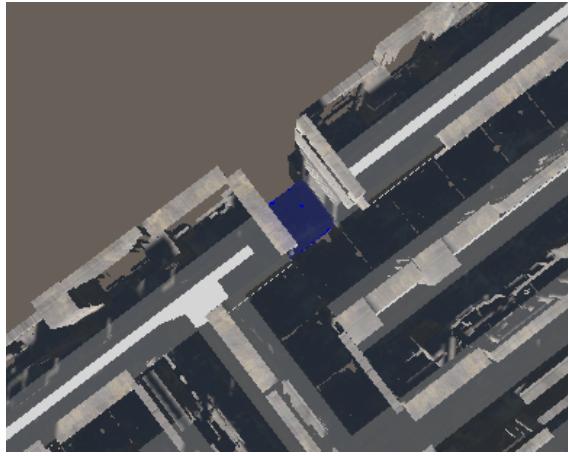


4. Entrance/exterior walls removal

The entrance count on each side defines how much maze exit it will create on each side (represented by blue squares on the gizmos)

Maze entrances & exterior walls removal	
Bottom Entrance Count	1
Top Entrance Count	1
Left Entrance Count	1
Right Entrance Count	1
Max Entrance Attempts	100





'MaxEntranceAttempts' defines the maximum number of times the generator will try to place a valid entrance connected to a path.

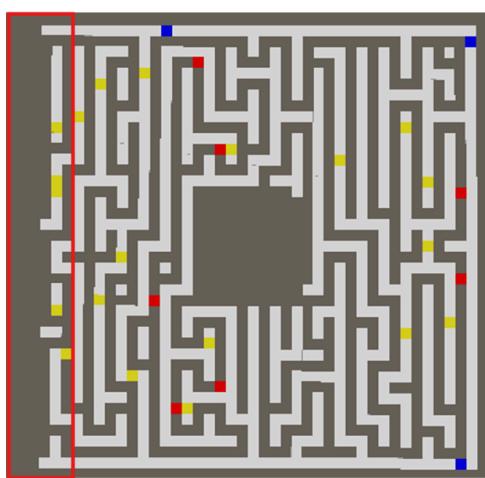
Below those are options to fully remove one wall side

- Remove Bottom Wall
- Remove Top Wall
- Remove Left Wall
- Remove Right Wall

For example :

- Remove Bottom Wall
- Remove Top Wall
- Remove Left Wall**
- Remove Right Wall

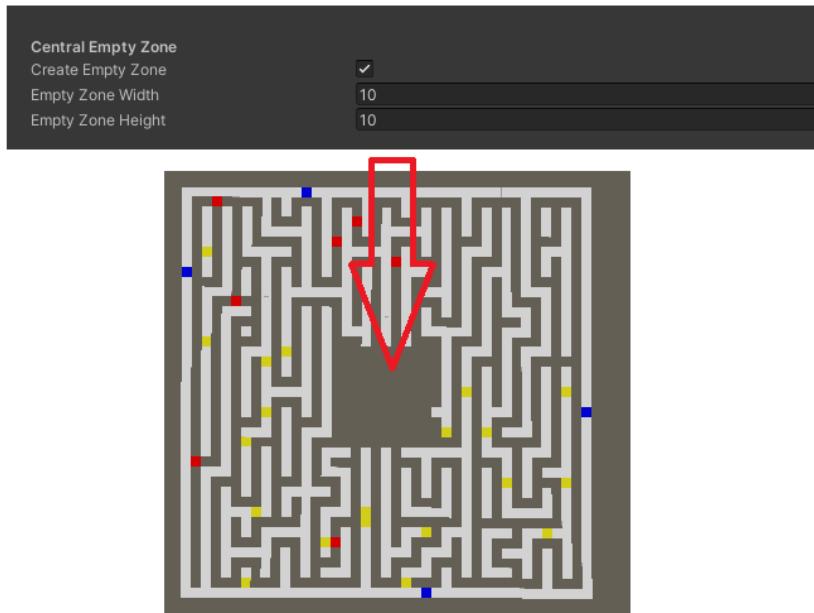
Screenshot with left wall removed :



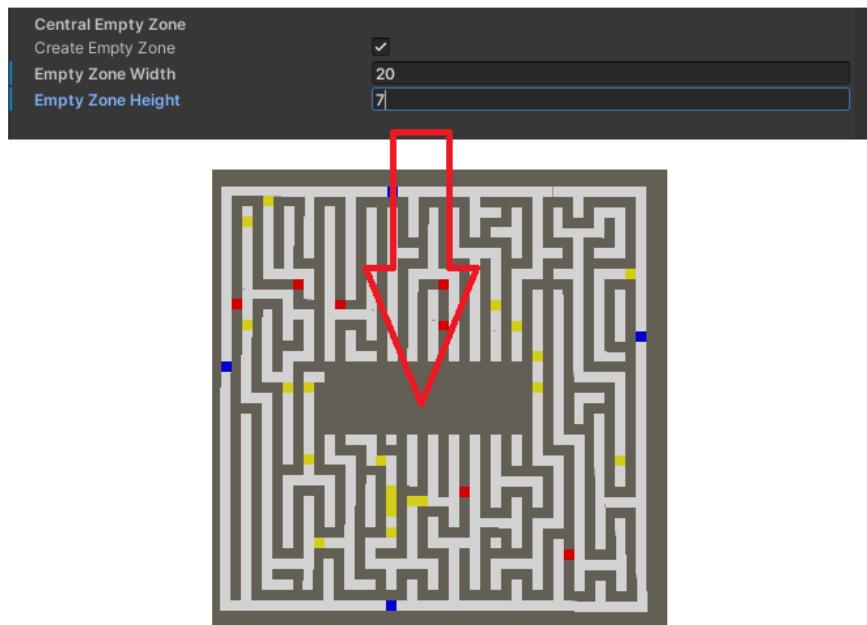
5. Central empty zone

Relatively straightforward, it defines if you want a central empty area and its size

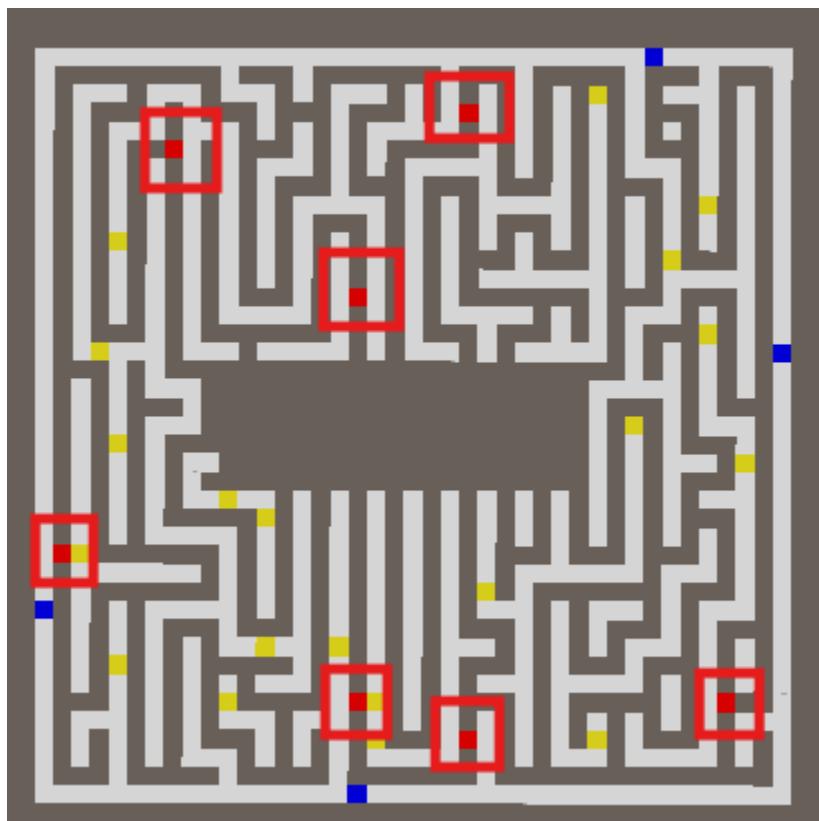
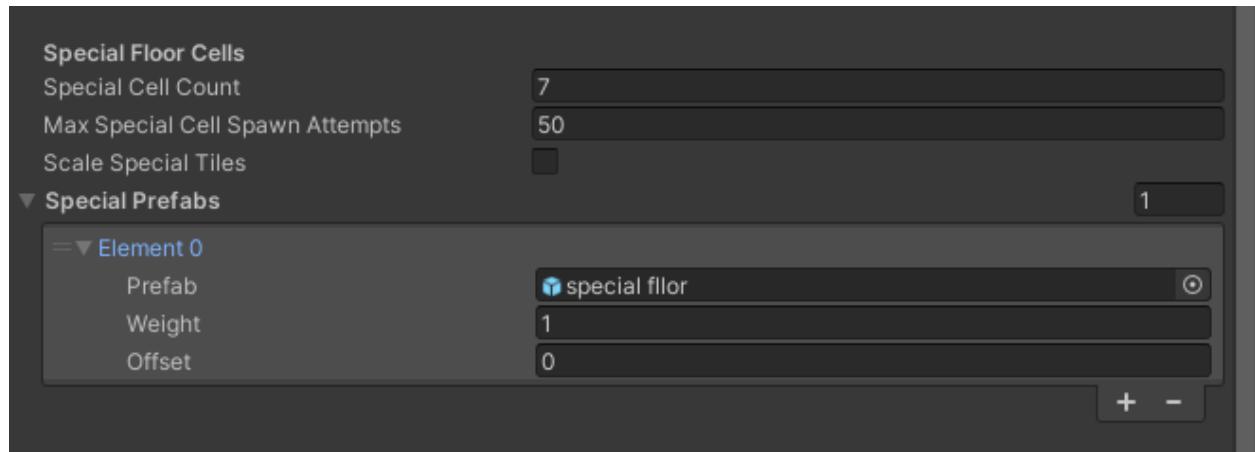
Example for 10x10 central empty zone :



Example for 20x7 central empty zone :

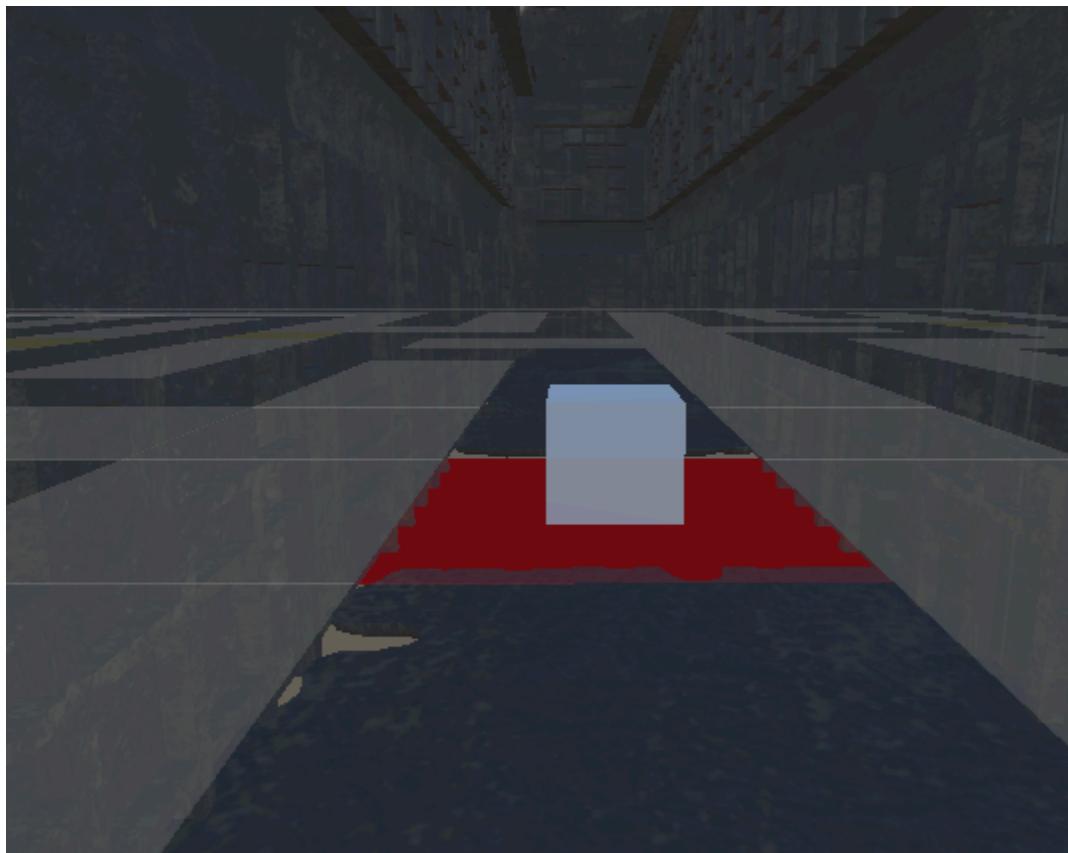


6. Special floor cells



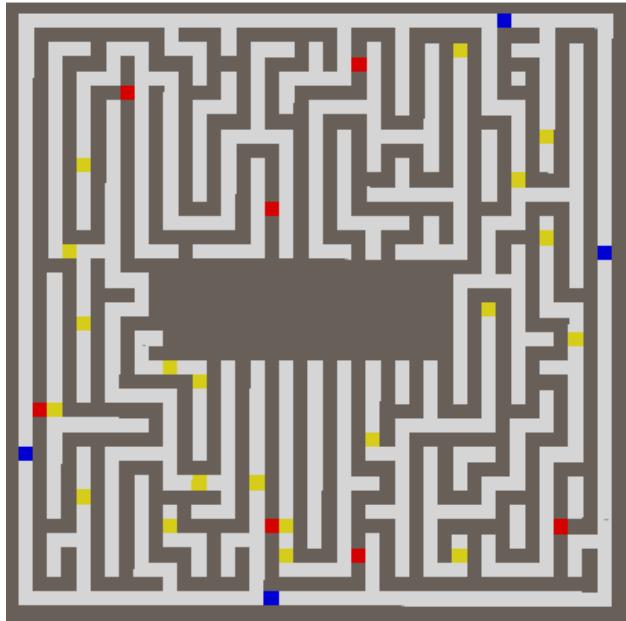
Special floor cells are represented by red cells in the gizmos, in this example the special cell count is 7, so 7 floor cells are replaced with special cells, and if meshes are

generated, a “special prefab” is randomly selected to replace the floor prefab depending on the same weight calculation explained previously, in the example scene it's just a red tile with a square in the middle but could be replaced with a point of interest, a broken tile, a trap etc.

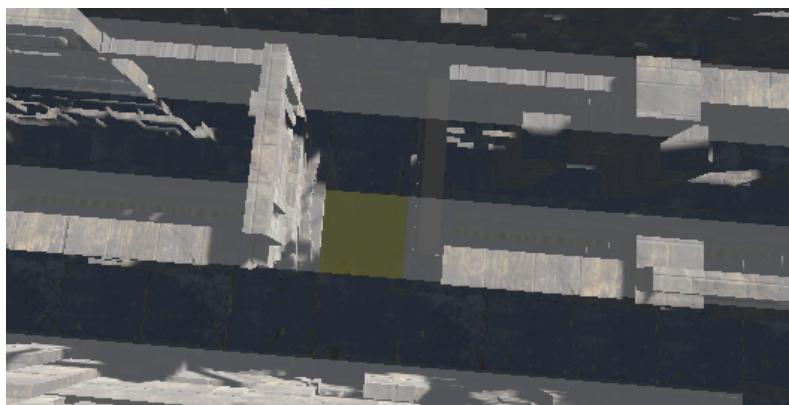


7. Random wall passages :



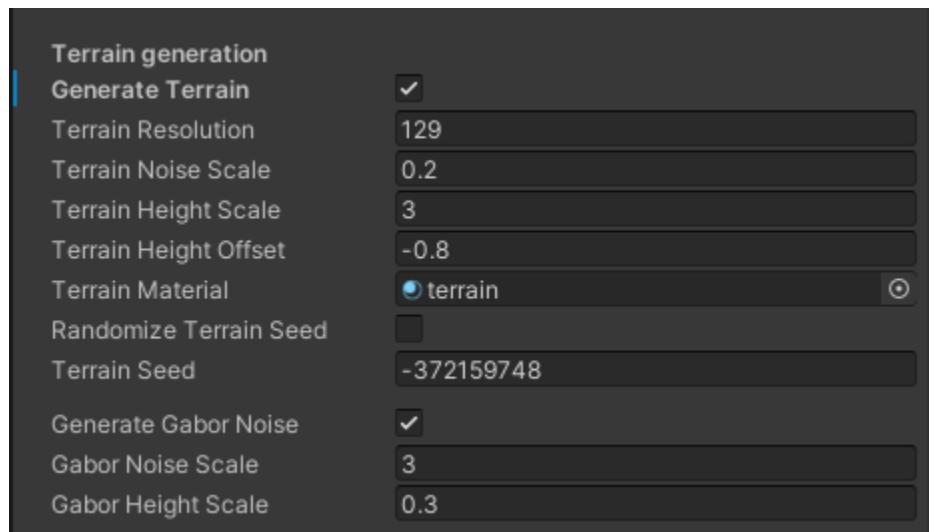


Represented by yellow gizmos, these indicate walls that will be deleted. In this example, **since 'WallsToRemove' = 20**, 20 wall cells are marked as yellow gizmos, and during mesh generation the script will attempt to remove them to create passages.

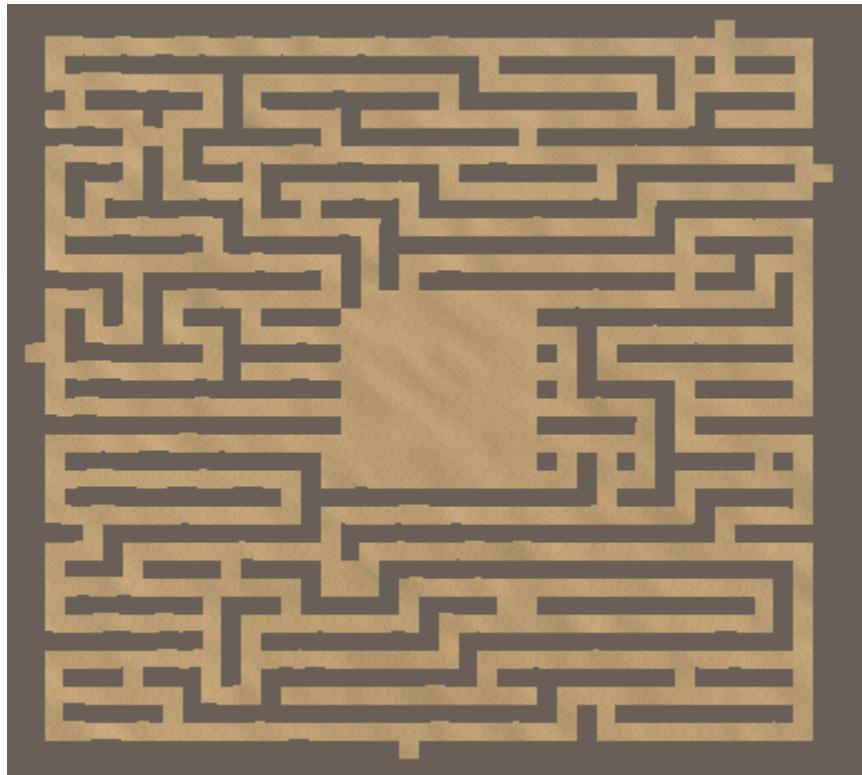


8. Terrain generation

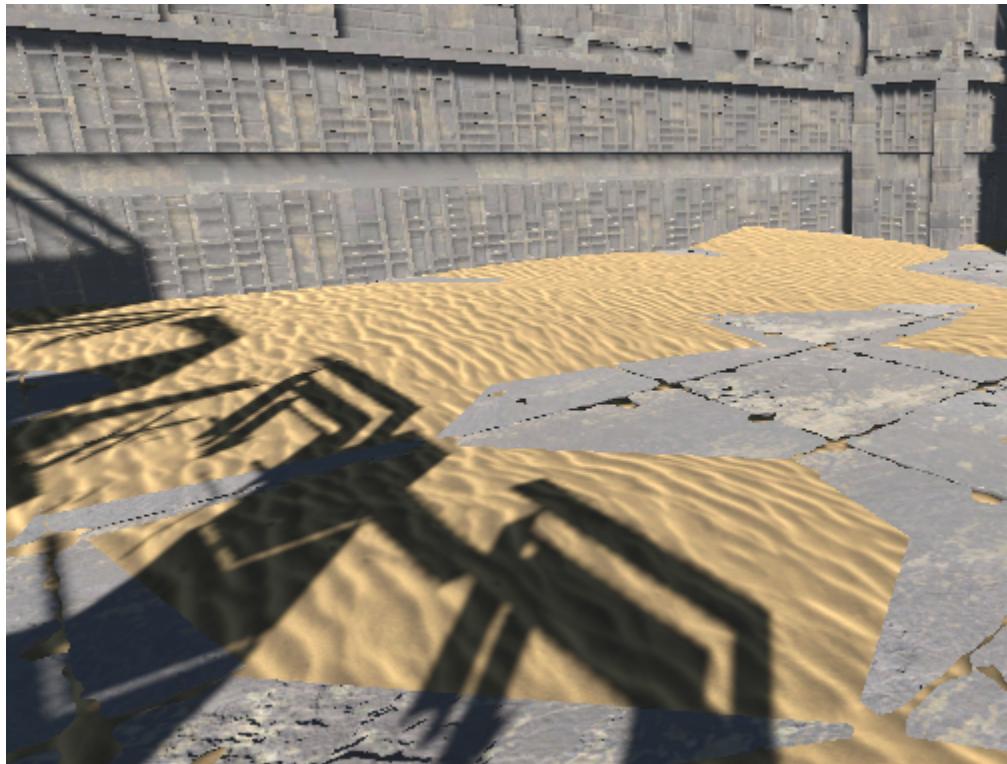
Last but not least is terrain generation



If the boolean '*GenerateTerrain*' is ticked a terrain that follows the floor shape will be created on top of floor



The Noise and Height Scale options control vertical variation, letting the sand rise above or sink below the concrete floor tiles.



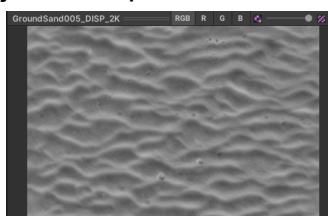
The terrain seed allows to randomize the height variation or to keep the same if uses the same number as an earlier generation (to randomize maze without randomizing the terrain sand)

Randomize Terrain Seed	<input type="checkbox"/>
Terrain Seed	-372159748

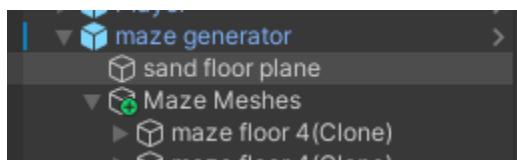
The gabor noise is an additional layer of noise for more height variation

Generate Gabor Noise	<input checked="" type="checkbox"/>
Gabor Noise Scale	3
Gabor Height Scale	0.3

(but the ripples you see on the sand are not handled by the gabor noise option — it's just a displacement texture I added in the albedo of the sand material)

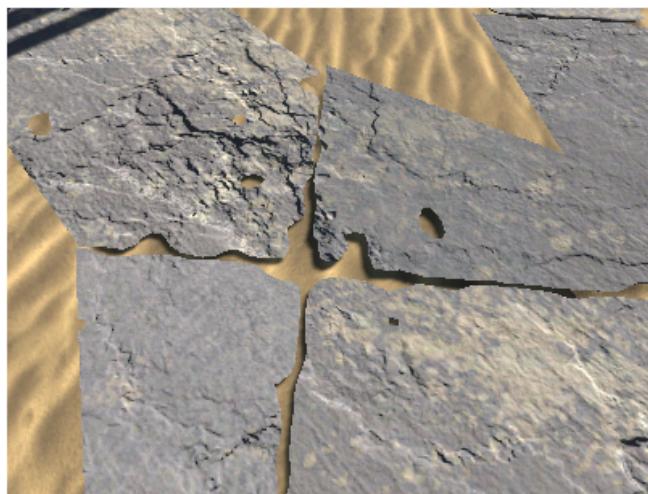


As an additional information, I manually put a flat plane with sand texture just below the floor level

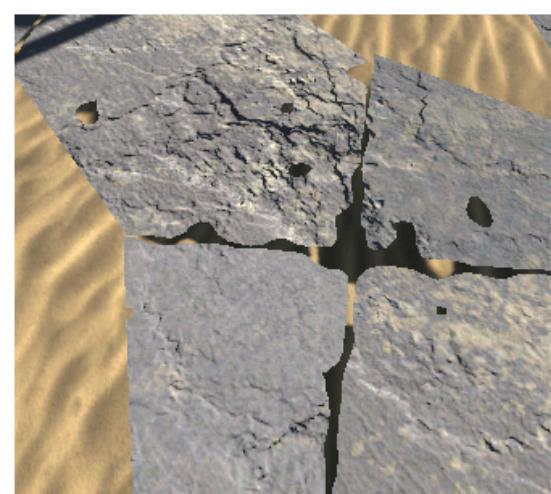


Without it, depending on the discrepancies caused by terrain noise, visible gaps may appear beneath the floor. :

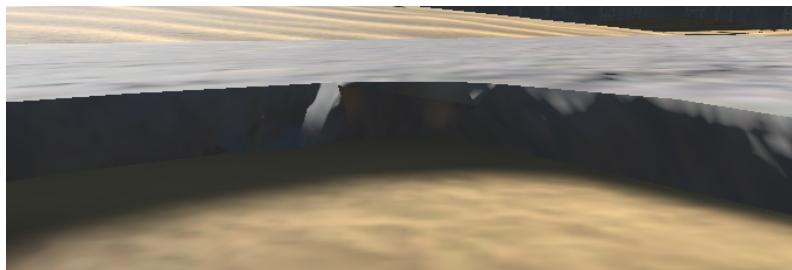
With sand plane :



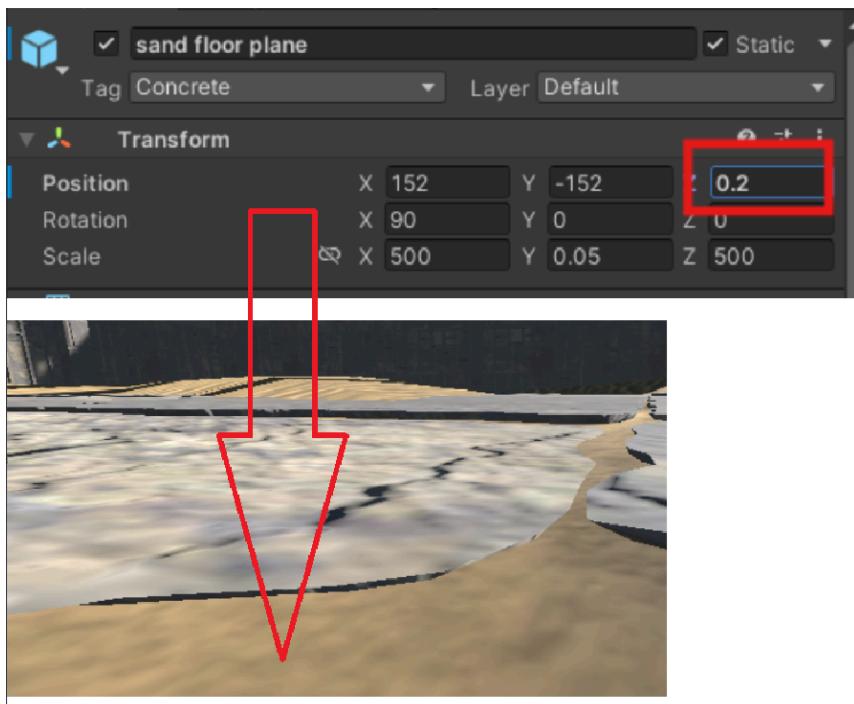
Without sand plane :



I prefer it to be at the lowest height possible where it touches the floor tiles and still creates some shadows



But if you don't want it to cast shadows you can just move it slightly higher like so :



(initial position is 0.13, I recommend a Z position offset between 0.125 and 0.2).

This concludes the Maze Generator documentation. For additional details or bug reports, please reach out via the contact information below.

Contact:

contact.dyspropulse@gmail.com