

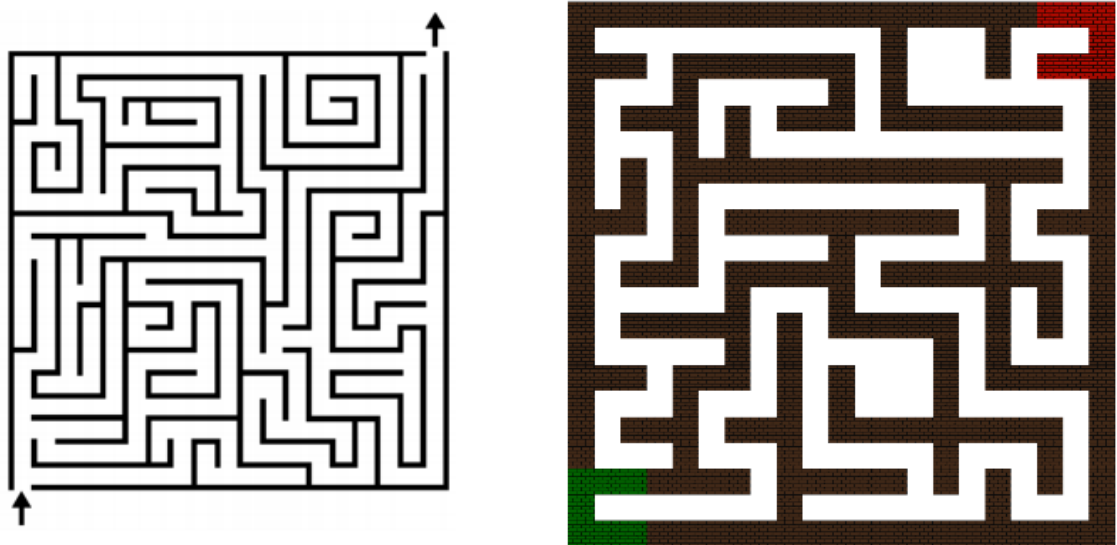
How to use MazeGenerator2D

Description

MazeGenerator is using for creating custom 2D maze. You can use custom wall to create your custom maze. Maze is using backtrack algorithm and cell system: each cell is rectangle and maze size is measured in cells. Also you can customize your maze by using room creation, you can set room count, room width and height. MazeGenerator can use seed for create same mazes(example: you want save/load generated level and you just can save/load only seed value) and mazeGenerator can create everytime random maze. MazeGenerator based on Maze Skeleton class. Each class you can serialize to JSON and save to file(maybe I will add it in future).

To create maze you can use prefabs(maze will automatically created) or use maze skeleton(get wall positions). At this moment walls using rectangle prefabs(walls) to create maze.

Maze have one entrance and one exit, as it shown on the picture 1.



Picture 1. Left picture - entrance and exit, right picture – maze example.

You can get access to maze, when scene is active, example: you want change level in your game. Maze script will help you to solve this task.

- At first step you need to get gameObject “MazeGenerator”
- At second step you need to get maze script
- And at third step you can use public method’s of MazeScript for your tasks.

Main functions of script:

1. Delete maze
2. Create maze new width and height
3. Create maze with seed(to create same mazes)
4. Set new prefab for wall, start and end walls
5. Get all impasses
6. Get all crossroads
7. Get all cell centers
8. Get all tunnels
9. Get way from start to end
10. Get gameObject of the start and end wall
11. Verify the position on the wall
12. Add custom rooms with width and height
13. Get all cell centers of rooms
14. Get room entrances(for doors)
15. SetSeed rules
16. Get longest way in maze

17. Set random start and set exit to end of longest way
 18. Use custom walls
- In demo version you can see how to use maze generator and maze script to create your game!

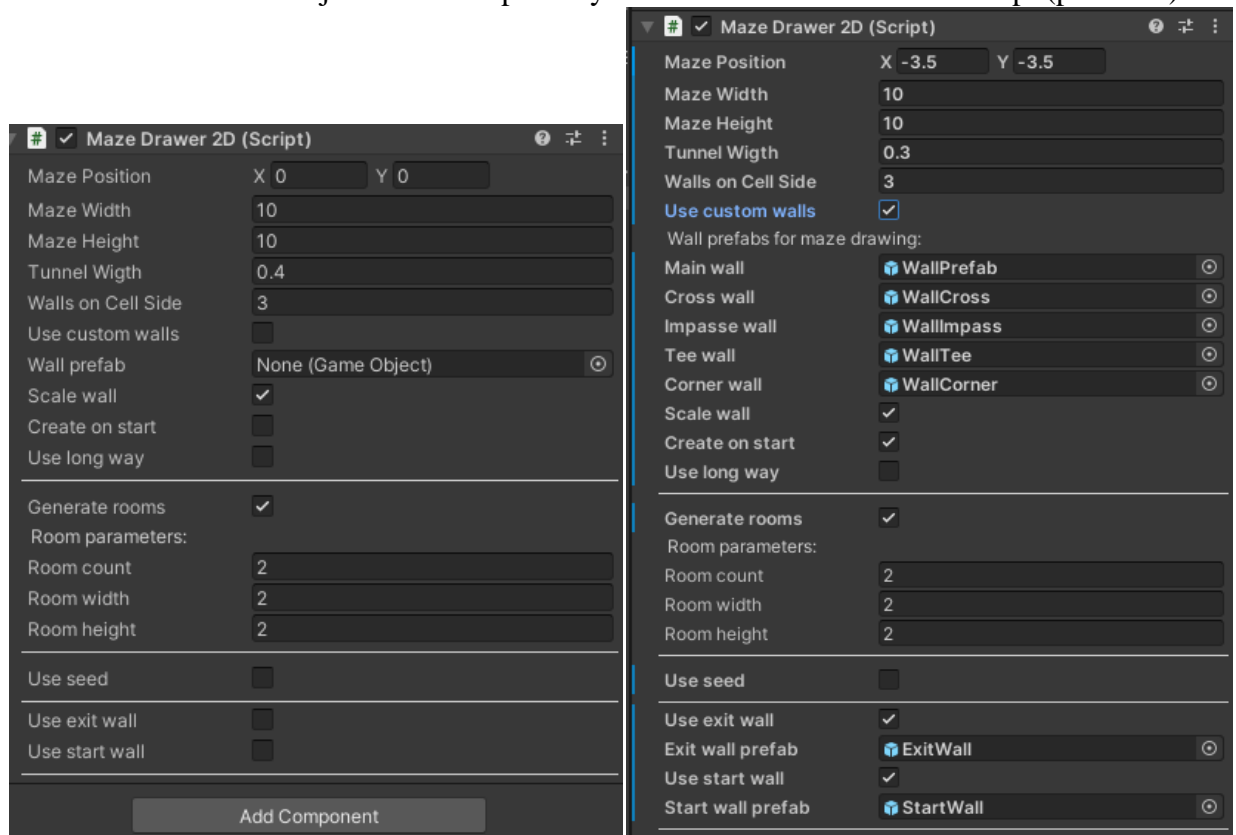
User's manual

If you want add a maze to the scene, you need to use “Maze” prefab(pic.2), you must transfer it to scene from folder “Prefabs”.



Picture 2. MazeGenerator prefab.

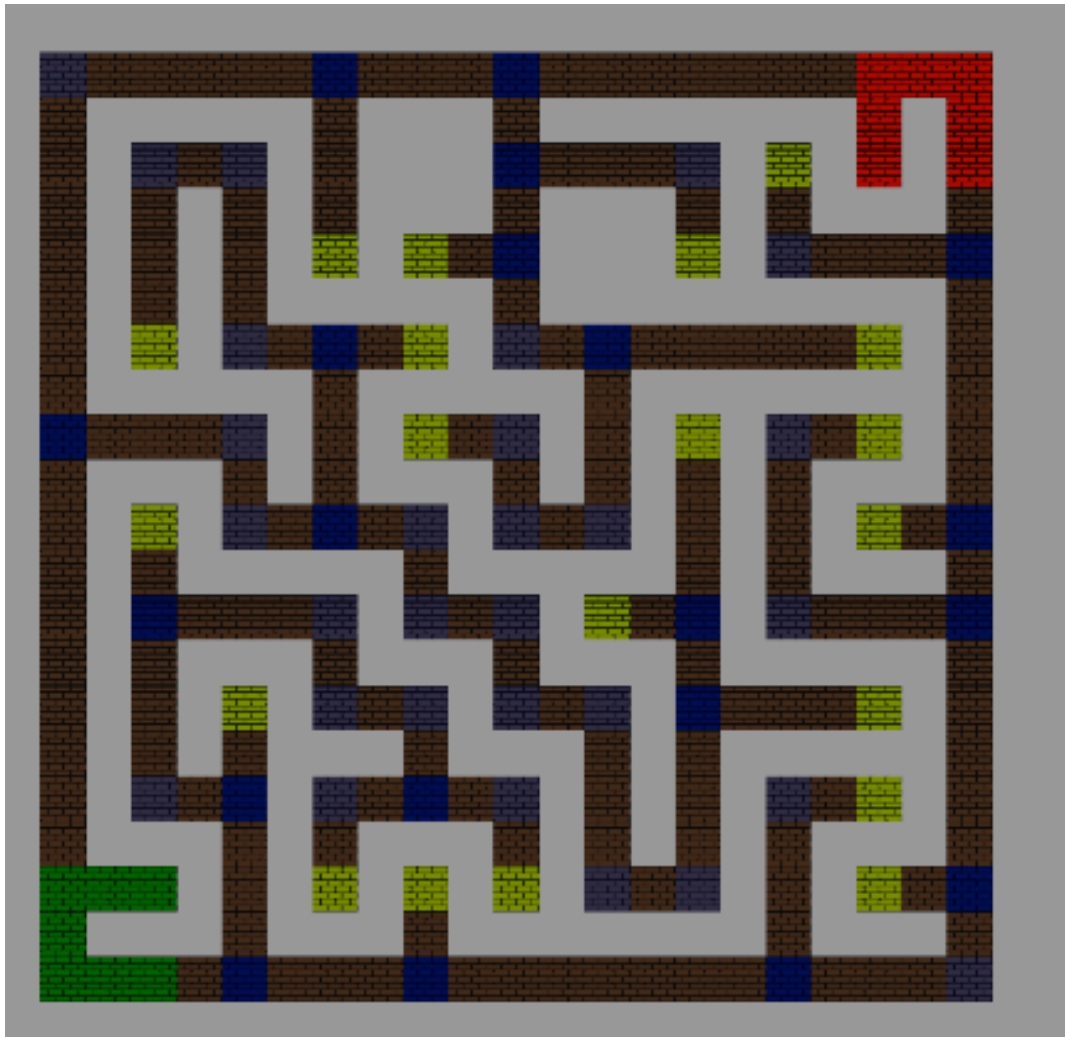
Select MazeGenerator object and in Inspector you can see Maze Drawe 2D script (picture 3).



Picture 3. Inspector view of Maze Drawe 2D script.

In this script you can set main parameters of the Maze: width, height in cell count, position on scene, tunnel width and wall count on each side of cell(you can set minimum 3).

If you want to use custom walls, check checkbox “Use custom walls” and set wall prefabs. Custom wall example you can see on picture 4.



Picture 4. Custom walls

If you will set Scale wall, walls will be automatically scaled. Source wall must have 1x1x1 size, and Pixels Per Unit must be equal sprite width.

If you will set “Use long way”, maze will set exit position to end of longest way in the maze.

If you want create rooms, set “Generate rooms” and fill room parameters.

If you want to create same mazes, set “Use seed” and set seed value.

Also you can set start and exit wall prefabs, for indication.

MazeGenerator contains 2 scripts: **MazeDrawer2D.cs** and **MazeSkeleton2D.cs**. Scripts have full documented code.

MazeDrawer2D.cs needs to draw maze, **MazeSkeleton2D.cs** contains MazeSkeleton class, MazeCell class, MazeWall class, MazeRoom class.

If you have some questions, you can write me on e-mail: radiomaster71@gmail.com