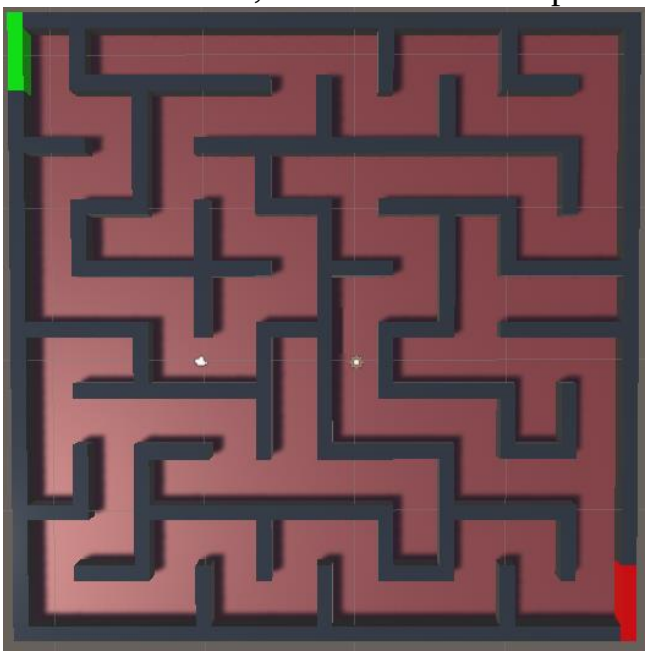
How to use MazeGenerator

Description

MazeGenerator is free demo version, it is using for creating custom 3D maze. You can use custom wall, floor and ceiling textures to create your custom 3D maze.
Maze is using backtrack algorithm.
Maze have one entrance and one exit, as it shown on the picture 1.



Picture 1. Maze example with entrance(green wall) and exit(red wall).

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 gameObgect “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. Set new prefab for wall, ground, ceiling, start and end walls

4. Get all cell centers (cell center positions in array of Vector3)

In demo version you can see how to use maze generator and maze script to create your game!

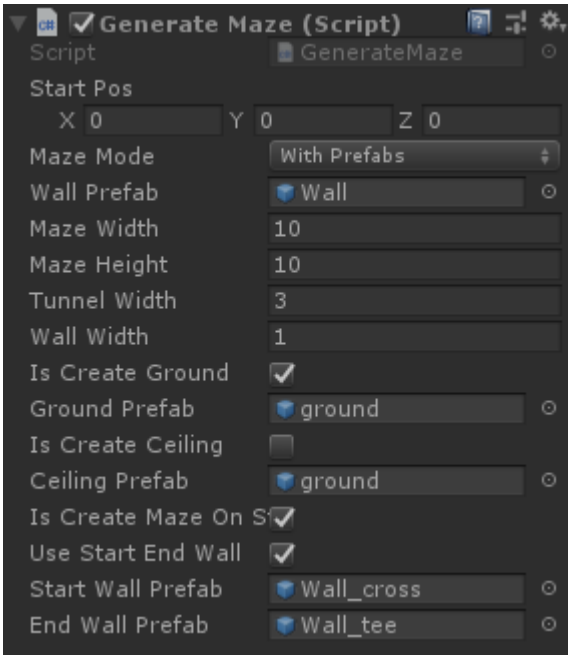
User's manual

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

Picture 2. MazeGenerator prefab.

On next step you can chose: use prefabs or get skeleton of maze.

If you chose prefabs mode, you must set wall prefab(size Vector3(1x,your depth,x1), floor(ground) prefab, also you can set ground, ceiling, start and end wall prefabs(pic.3).



Picture 3. Maze options.

Also you can set maze width and height in cellSize, maze start position(position maze center). You can set TunnelWidth and wall width, it means how mush wall will be created on one side of cell (TunnelWidth > WallWidth, use integer values). If flag "Create maze on start" is set – maze will be created after initialization of the MazeGenerator object. If flag "Create ceiling" is set – ceiling will be created.

Scripting

List of public variables:

Prefabs:

```
public GameObject wallPrefab, groundPrefab, ceilingPrefab, StartWallPrefab,
EndWallPrefab;
```

Maze size in cells:

```
public int MazeWidth = 10, MazeHeight = 10;
```

Position of left top maze corner

```
public Vector3 startPos = new Vector3(0, 0, -10);
```

Flags to create maze on start and flag to create ceiling:

```
public bool CreateMazeOnStart = true, createCeiling = false;
```

Tunnel width and wall width(set wallWidth = 1 as constant)

```
public float TunnelWidth = 1;
public float WallWidth = 1;
```

List of public functions:

Function to set flag for create Ceiling

```
public void SetCreateCeiling (bool create)
```

Function for set wall prefab

```
public void SetWallPrefab (GameObject newWall)
```

Function for set ground prefab

```
public void SetGroundTPrefab (GameObject newGround)
```

Function for set ceiling prefab

```
public void SetCeilingPrefab (GameObject newGround)
```

Function for set start wall prefab

```
public void SetStartWallPrefab (GameObject newGround)
```

Function for set end wall prefab

```
public void SetStopWallPrefab (GameObject newGround)
```

Function returns size of the maze(in cells)

```
public Vector2 GetMazeSizeInCells ()
```

Function for deleting all maze objects

```
public void DeleteMaze()
```

Function for delete maze and create new maze with new width and height(in cells)

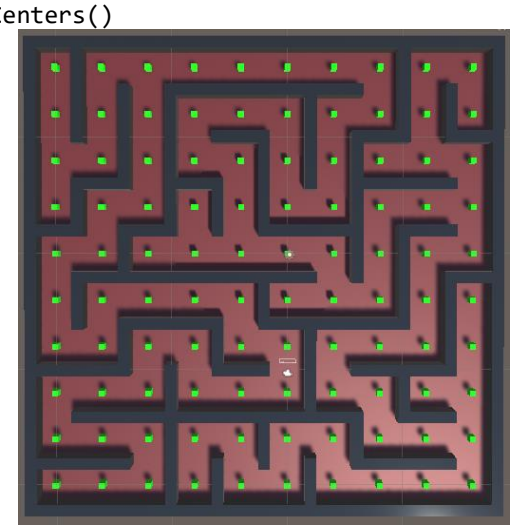
```
public void ClearAndGenerateNewMaze(int newMazeWidth, int newMazeHeight)
```

Function for creating new maze with new width and height(in cells)

```
public void GenerateNewMaze(int newMazeWidth, int newMazeHeight)
```

Function return list of all cell centers in maze (cell center positions, pic.4)

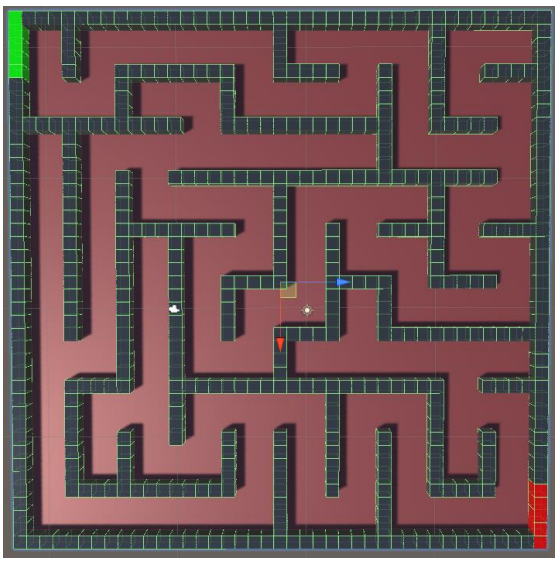
```
public Vector3[,] GetGridCenters()
```



Picture 4. Example of way from start to exit. Function will return start position in maze (cell center position)

Function for get wallArray. It is a skeleton of maze, you must create wall on each position from list. (pic.5)

```
public List<Vector3> GetWallArray()
```



Picture 5. Maze skeleton.

Function to getSize for ground

```
public Vector2 GetSizeForGround()
```

Help

Next code will help you to get the gameObgect “MazeGenerator”:

```
private GenerateMaze maze;
void Start () {
maze = GameObject.FindGameObjectWithTag("MazeContainer").GetComponent<GenerateMaze>(); }
```

If you have some questions or found some bugs, errors, you can write me on e-mail: **radiomaster71@gmail.com**