Random Map Generator

Usage with Scripts

After setting up the script (check Setup and Usage PDF) you can use the generator in your scripts.

You can to get the script in the scene to call its functions.

```
    randomMapGenerator rmg = GetComponent<randomMapGenerator>();
```

For more info follow:

https://docs.unity3d.com/ScriptReference/GameObject.GetComponent.html

After getting the script you can call its functions. Following functions are available.

`GenerateMap()`	Generates a map with the setting in the inspector. Returns `rmgMap` which can be stored and drawn later.
`GenerateAndDraw()`	Generates a map with the setting in the inspector and instantiates it in the scene. Returns `rmgMap` which can be stored and drawn later.
`GenerateAndDraw( Vector3 )`	You can also pass a Vector3 to `GenerateAndDraw()` set the starting position of the map.

All variables of the *randomMapGenerator* script are public and editable with other scripts.

Options are the following:

- **List<roomList>** roomsToSelectFrom
- **string** mapName
- **float** roomScale
- **Vector3** startingPosition
- **int** roomCount
- **bool** limitSize
- **int** maxWidth
- **int** maxHeight

Storing and Instantiating Maps

You can store generated maps in a *`rmgMap`* class to store and instantiate them at a later time.

```
rmgMap myMap = rmg.GenerateMap();
```

You can call the Draw() function in the created `rmgMap()` class to instantiate the map. You can Also set the starting position of the first room by supplying the Draw() function with a Vector3, similar to the `GenerateAndDraw()` function.

```
myMap.Draw(new Vector3( 10, 0, 0 ));
```