Hi, thank you for downloading the Blocky Level Generator tool. In this tutorial I will show you what the different variables are and show the process of setting up the tool. So without further ado, let's get started.

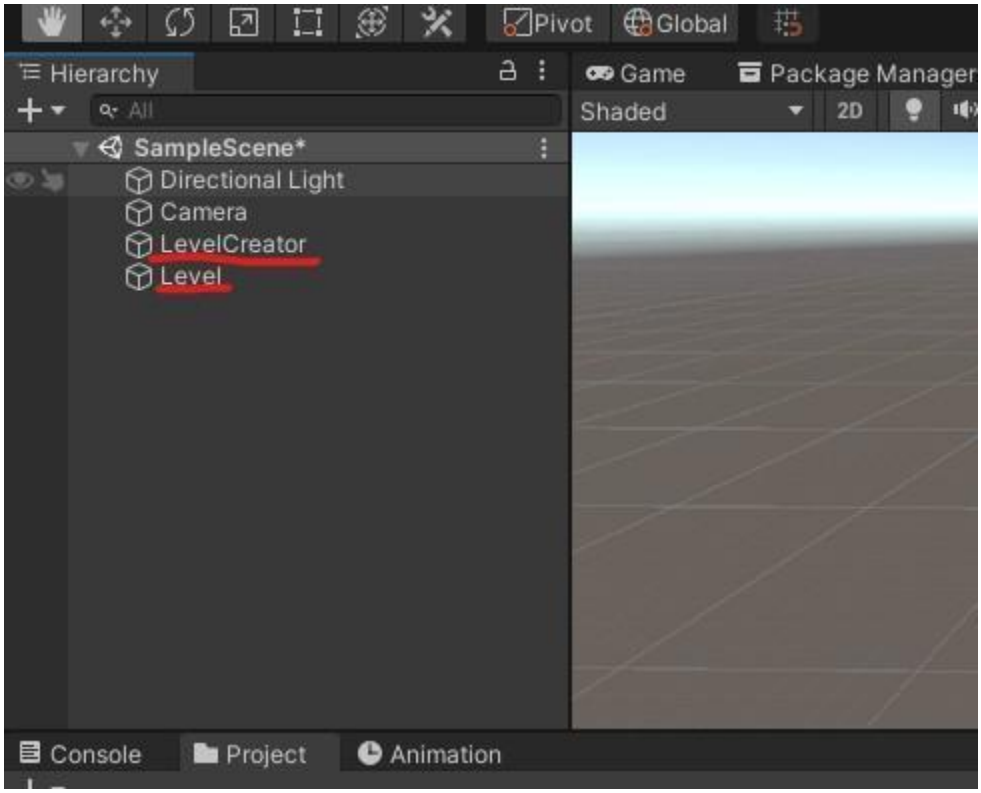
Part One – The Inspector Setup

1.1

Firstly you will need to open a new project/use your existing one, and download/import the package (seeing as you are reading this I will assume you have this first part done).

1.2 - Empty Objects

Next you will need two empty objects - however, if you use just one, a level object will be created to parent everything.

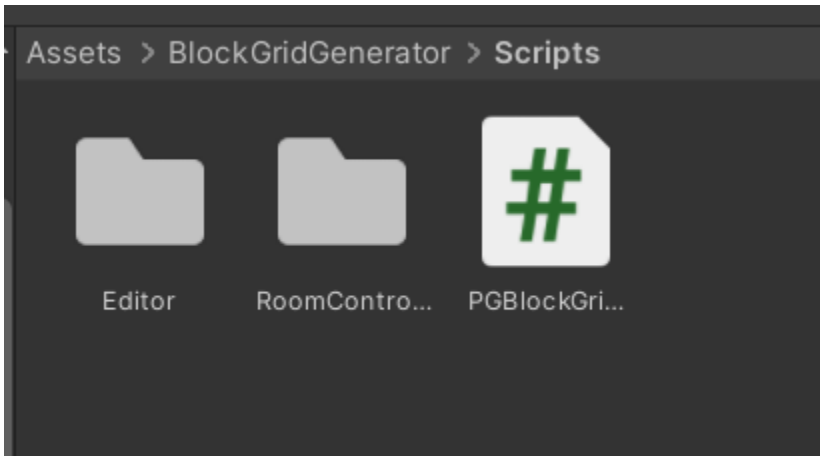


In the example above I have simply created two objects, "Level" and "LevelCreator".

NOTE *You do not actually need two objects, I just prefer it. One object will hold the script, and the other will be the parent for all the rooms and objects placed into the scene.*

1.3 - Scripts

We will now attach the script to the GameObject (I add mine to the level creator object). Go into the Block Grid Generator package and locate the scripts folder. Select the “PGBlockGridLevel” script. Drag in onto your GameObject.



And now we can start setting up our script.

- The Editor File Contains our Editor Script
- Room Controls Contains our Room Control Script (this is attached through code)

Part Two – The Script Setup

Alright, let's start setting up our script. I will break the script down into three parts, and explain what each of the variables do.

Remember, in Blocky Level Generator, we assign many min and max values, so that every room and corridor look the exact same - we want a variety of widths and lengths.

Note

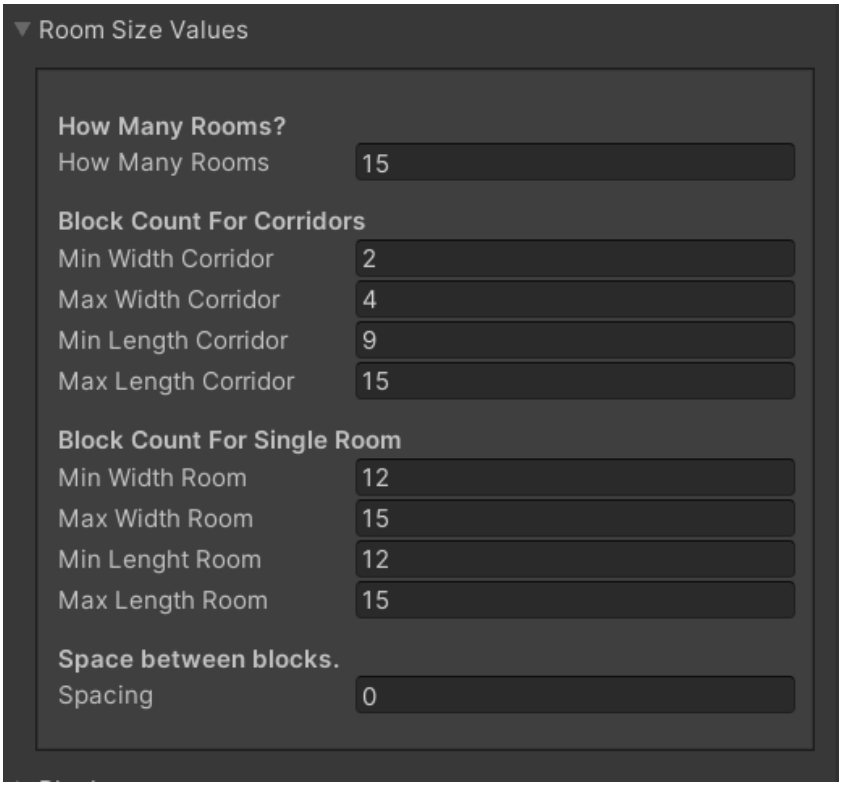
Only Room SIze Values, Floor Block, and Wall Block are required, everything else is optional

2.1. - Editor Mode

Whether you wish to run the generator in the editor or in play mode. When unselected the coroutine will be called at the start of the game.

(Note the same function to be called at Start() in the the same called by the "Build Level" button, and can be called to build (first destroying the old if it exists) a level from any script).

2.2 - Room Size Values



The values here are the default values and can be seen when script is attached to an object.

A. How Many Rooms:

How many rooms explains itself, since it is how many rooms you want built.

B. Block Count For Corridors:

- Min Width Corridor: The minimum width of a corridor
- Max Width Corridor: The maximum width of a corridor
- Min Length Corridor: The minimum width of a corridor
- Max Width Corridor: The maximum width of a corridor

You assign the min and max widths/lengths you want your corridors to be (this is done like this to allow for random sized corridors).

C. Block Count For Rooms:

- Min Width Room: The minimum width of a room
- Max Width Room: The maximum width of a room
- Min Length Room: The minimum width of a room
- Max Width Room: The maximum width of a room

You assign the min and max widths/lengths you want your rooms to be (this is done like this to allow for random sized rooms).

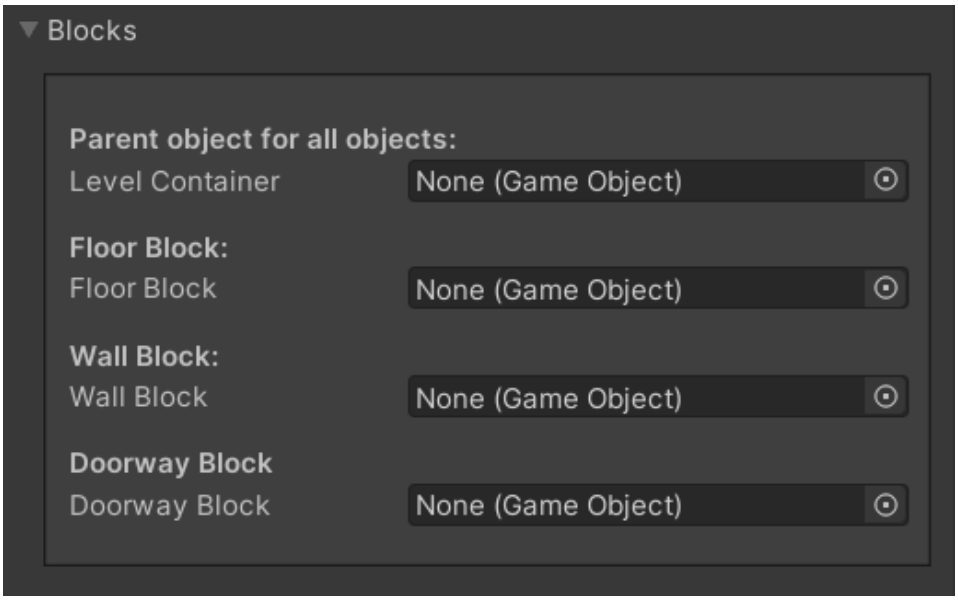
D. Spacing:

This allows you to add space between the blocks.

(Though it will take a float, it will only work with an int. E.g. If you put in 0.5, there will be no difference. But if you put in 1, then it will place that much space between every block.)

We will leave the default values as they are, and continue on.

2.2. - Blocks:



A. Level Container:

This will contain everything. You can use the empty "Level" GameObject we created earlier. Or you can use the "Level Creator" itself. It doesn't really matter as the object will not be destroyed when clearing. Also, if no object is assigned, it will simply create its own.

B. Floor Block:

This is the floor block which will make up all your floors - *for the best outcomes, use a copy of this block as the base of your next two blocks when creating prefabs*.

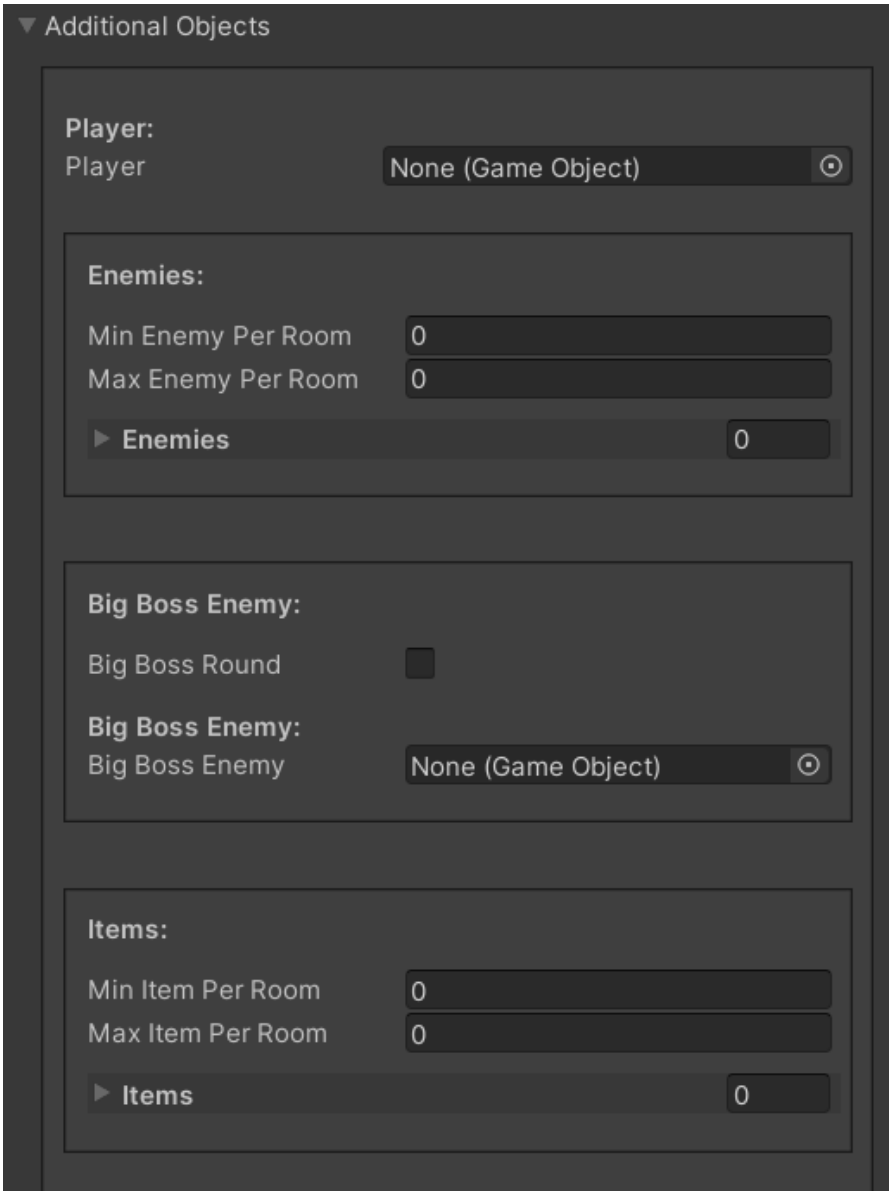
C. Wall Block:

This is the wall block which will be used to make your walls.

D. Doorway Block:

This is the block which will be used as our doorways - *if not selected, a wall block will be used instead*.

2.3. - Additional Objects:



A. Player:

This will instantiate your Player into the first room. (This instantiation happens before enemies and objects.)

B. Enemies:

- Min Enemies Per Room - Is the minimum amount of enemies you want to instantiate in each room.
- Max Enemies Per Room - Is the maximum amount of enemies you want to instantiate in each room.
- Enemies - Is the list of enemies you want to instantiate.

This is a list of enemies you want instantiated, and the min/max you want per room (again this is done to randomize the amount).

C. Big Boss Round:

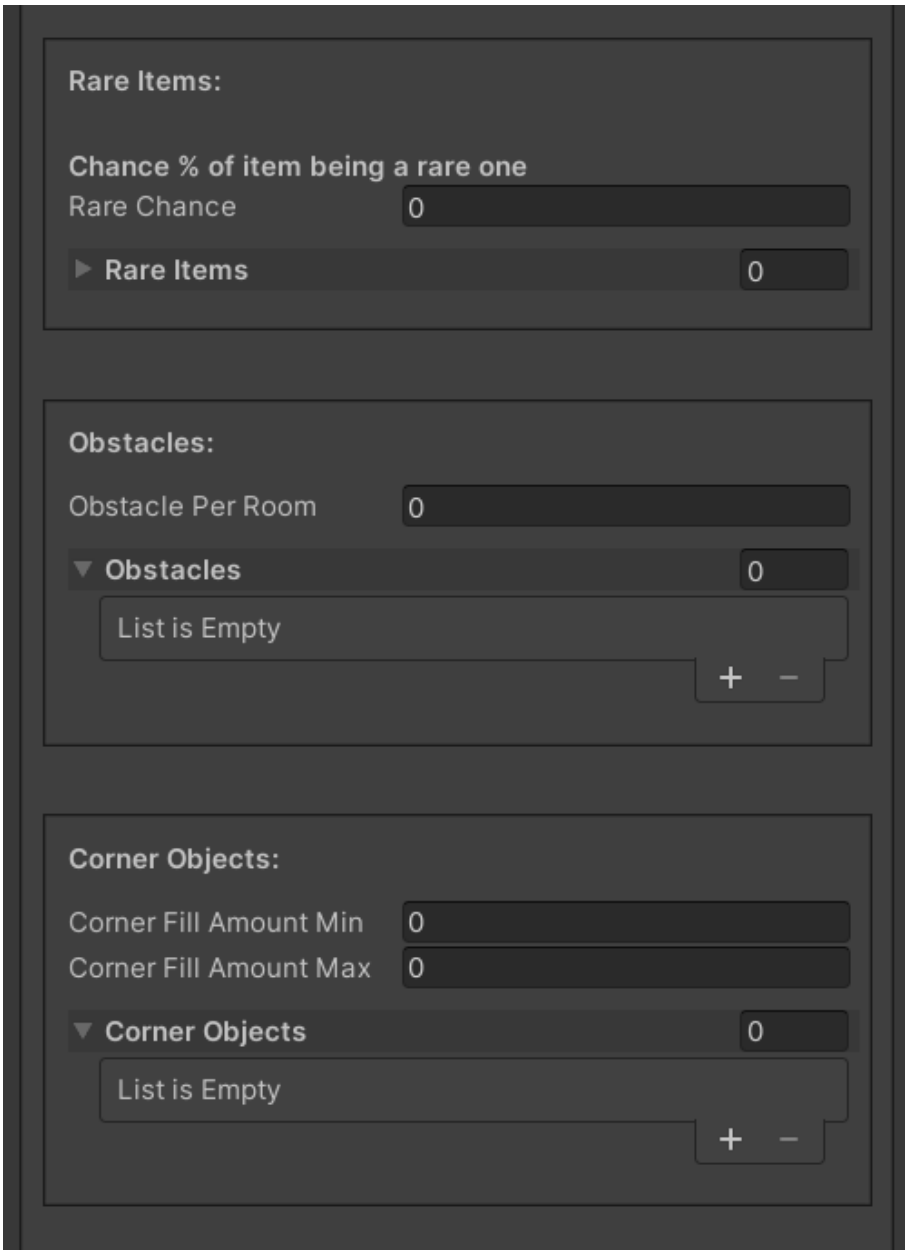
- Big Boss Round (bool) - is the option to have a big boss enemy instantiated.
- Big Boss Enemy - is the big boss enemy you want instantiated.

When the “Big Boss Round” bool is set to true, it will instantiate whatever boss GameObject you place with the variable. (If you are planning to place a large in size boss, then I would advise making sure your rooms are all large enough.)

D. Items:

- Min Items Per Room - minimum amount of items you want instantiated per room.
- Max Items Per Room - maximum amount of items you want instantiated per room.
- Items - are the items you want instantiated.

This is a list of items you want instantiated, and the min/max you want per room (again this is done to randomise the amount).



E. <u>Rare Items:</u>

- Rare Chance - is the percentage chance of the rare item appearing.
- Rare Items - is the list of rare items you want instantiated.

This is a list of rare items you want potentially instantiated. The **<u>Chance</u>** variable, allows you to give a percentage out of 100 if any of the items instantiated are a rare one.

(E.g. If we put in 5, that would be the same as saying that every item placed has a 5% chance of being rare).

F. **Obstacles:**

- Obstacles Per Room - is the amount of obstacles you want per room.
- Obstacles - is the list of obstacles you want instantiated.

These are objects you can scatter randomly around the room, to allow for greater room interior variations.

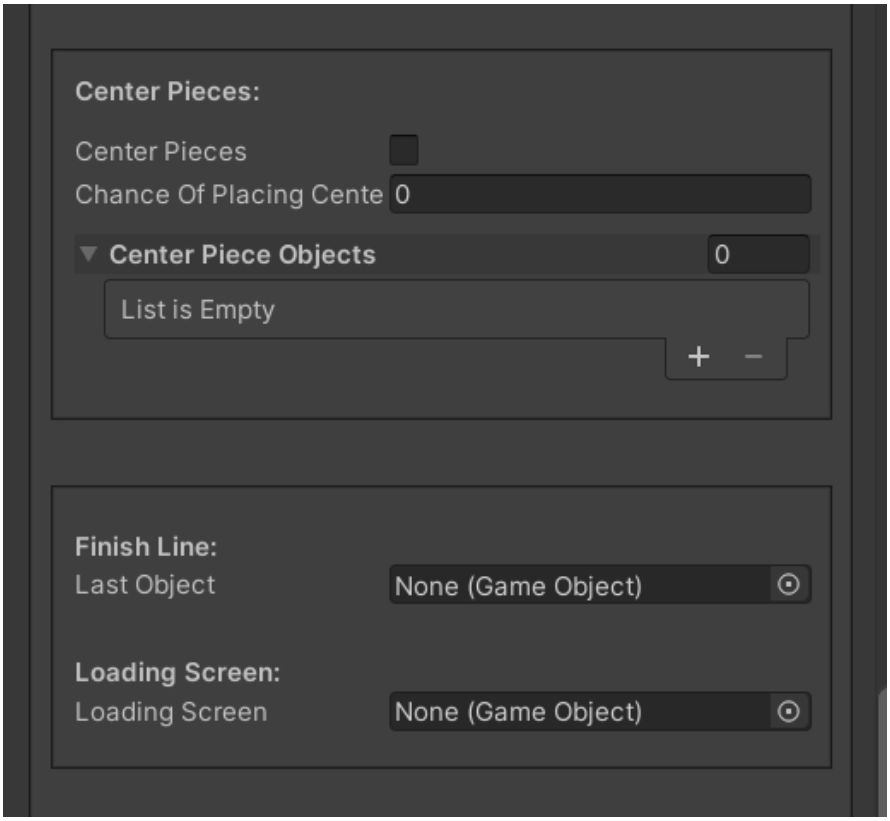
To allow for more functionality, the obstacles blocks were changed to obstacles.

G. **Corner Objects:**

- Corner Fill Amount Min - is the minimum amount of corner objects you want per room.
- Corner Fill Amount Max - is the maximum amount of corner objects you want per room.
- Corner Objects - is the list of obstacles you want instantiated.

These are objects you want placed in the corners of your rooms - remember, the amounts here account for four corners.

To combat the uniformity of simply placing the same amount of corner objects in each corner, the code instead places them randomly from a list of all the corner spots in a room.



H. **Loading Screen:**

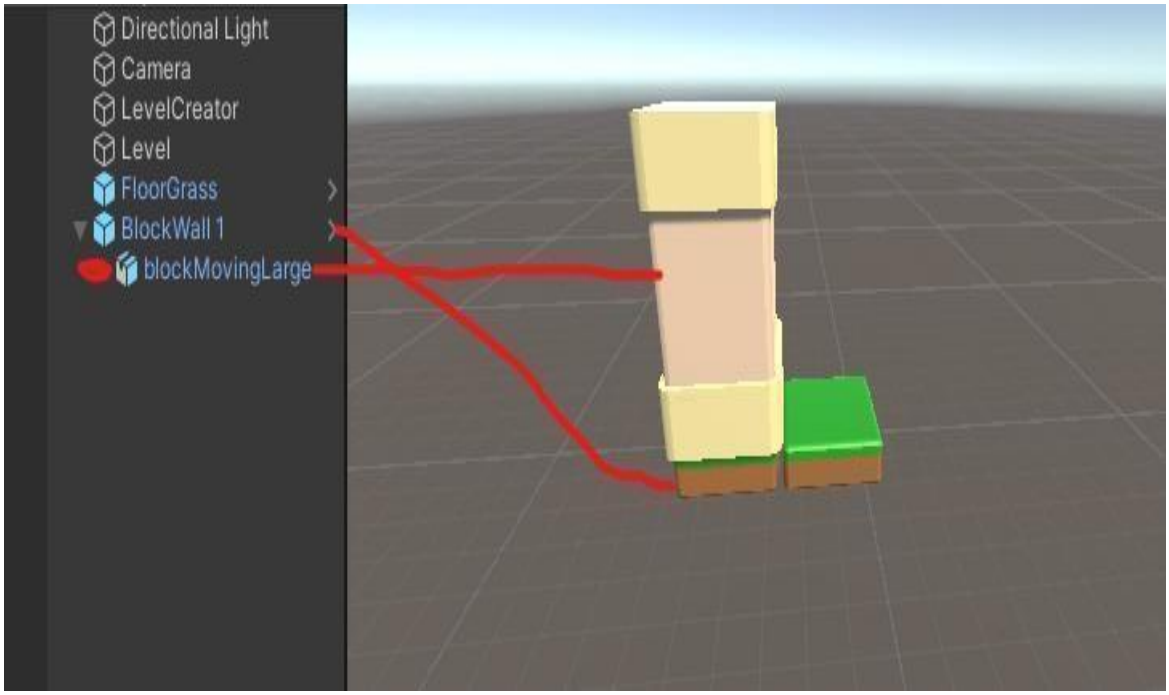
When the coroutine starts, it will set your loading screen object to active. And when it finishes, it will set your loading screen object to inactive.

I. **Finish Line:**

This will instantiate your Player into the first room. (This instantiation happens before enemies and objects.)

WALL BLOCK EXAMPLE (Can also be applied to doorways):

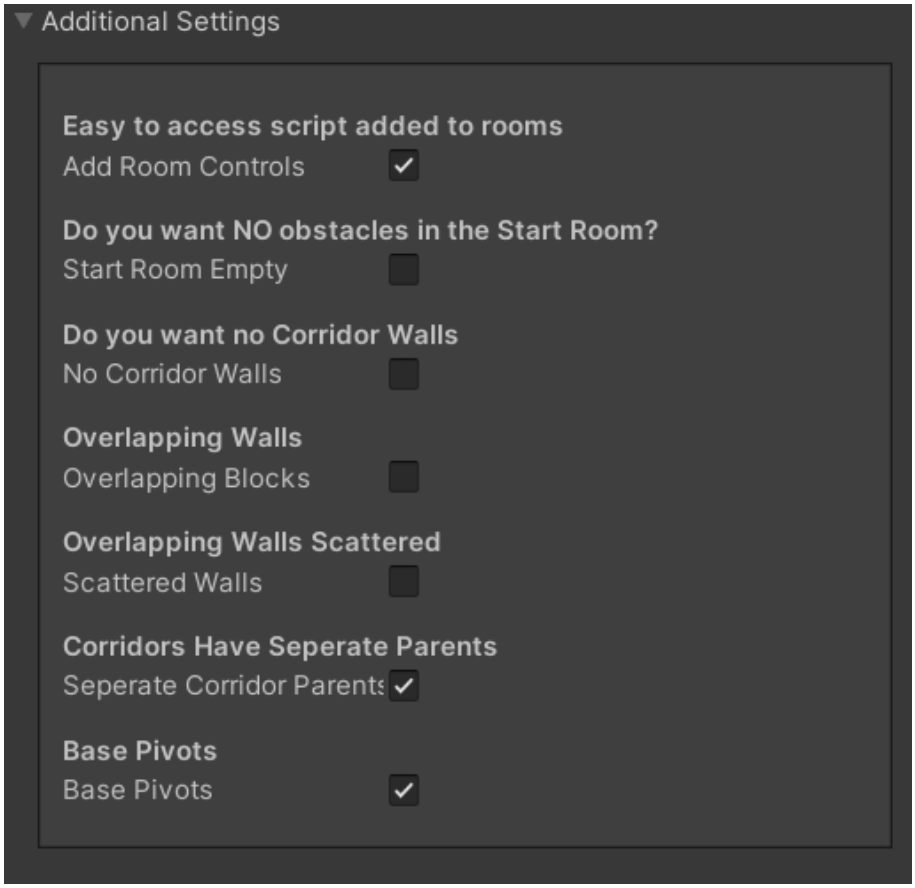
E.g.:



As you can see we have a Floor Block (“Floor Grass”). Then we have a second Floor Block (“Block Wall 1”) with a child wall (“blockMovingLarge”).

The same logic can also be applied to the doorway. If you doorways are coming out in the wrong direction, that just means you need to rotate it’s (the child object’s) Y by 90 and save the prefab.

2.3. - Additional Settings:



A. <u>Add Room Controls:</u>

(Set True Default) When Add Room Controls is active, the Blocky Grid Room Controls script will be attached to a room, and its blocks assigned their positions.

(Set false if overlapping walls is true.)

B. <u>Start Room Empty:</u>

By default, the start room will have obstacle blocks placed in like any other room. If you do not want this, you can select the option.

C. <u>No Corridor Walls:</u>

Corridors with walls are the default, you can select to not have them here.

D. <u>Overlapping Wall:</u>

When this option is selected, the generator will no longer take into account the distance from rooms, but instead allow them to overlap, and then clear. This creates greater variety in rooms, but makes it more difficult to attribute a single function to a single room - since many will be spliced - so Add Room Controls and Separated Corridors is set false.

E. **Scattered Walls:**

The Scattered Walls option, is essentially, a second check that runs through all the blocks, and ensures there are no overlapping blocks, unnecessary walls or missing walls. Due to the level of the check, it is slower (more so than the checks in overlapping blocks), and it removes the walls on corners, creating a more jagged look to edges.

F. **Base Pivots:**

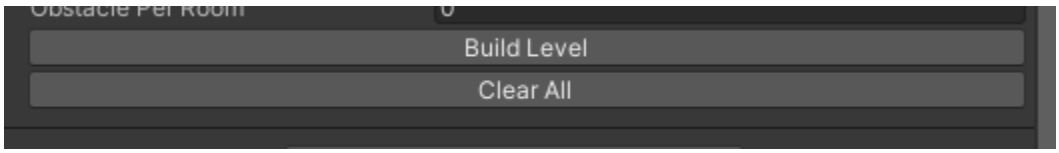
Base Pivots, is how the objects (mainly enemies, obstacles, and items) are placed on the blocks. Blocks are placed on the Y position at 0 (zero). So when your Floor Block is placed, it is in a different position on the Y, slightly, or more significantly, because of a center or base pivot.

If the blocks and the objects do not have base pivots, turn this off. But remember, this does not affect the blocks, just the objects placed atop during the build.

Part Three – Run The Code

There are two ways to run the script: in Editor Mode, in Play Mode. Play Mode will run the Builder if Editor Mode is false.

(If you don't want it doing so - e.g. you want it to be idle in the background of a game - simply remove the Start() function from **PGBlockGridLevel**.)



A. **Build Level:**

When clicked this will build the rooms. This button simply calls a function that will build the variables that you have placed inputted. The function will also clear any old rooms from the Level Container before creating.

(If editor mode has not been turned on, this will be very slow, as the coroutine it will be trying to run "WaitForSeconds").

B. **Clear All:**

This clears everything that was done when you clicked "Build Level". Think of it as a reset. And don't worry, you can click it as much as you like. (You do not need to press this button every time, as "Build Level" will clear and build at once.)

Note

In Editor mode, if your computer is really slow, building 100+ levels can take a couple of minutes. Especially if you run overlapping walls.

In Play Mode, there is no problem building lots of rooms.

****Most Variables are public, so that they can be changed through script in the case you want to build another room when the player reaches the end goal (and so on).**

Simply:

- **Get your reference to the PGBlockGridLevel Script.**
- **You could change, say, the amount of rooms, add enemies, more items, higher rare item chance.**
- **Then call StartBuilding - once - to clear the last level, and make a new one.**

Here are some relevant variables you can adjust from whatever rounds system you have created (or when the player reaches the end).

–Blocks–
FloorBlock - GameObject
WallBlock - GameObject
DoorwayBlock - GameObject
loadingScreen - GameObject

–Objects–
ParentObj - GameObject
Player - GameObject

–Enemies–
Enemies - List
minEnemyPerRoom - int
maxEnemyPerRoom - int
BigBossEnemy - GameObject
bigBossRound - bool

–Items–
Items - List
minItemPerRoom - int
maxItemPerRoom - int

–Rare Items–
RareItems - List
rareChance - List

–CenterPieces–
CenterPieceObjects- List
ChanceOfPlacingCenterPiece - int

–Obstacles–
Obstacles - List;
obstaclePerRoom - int

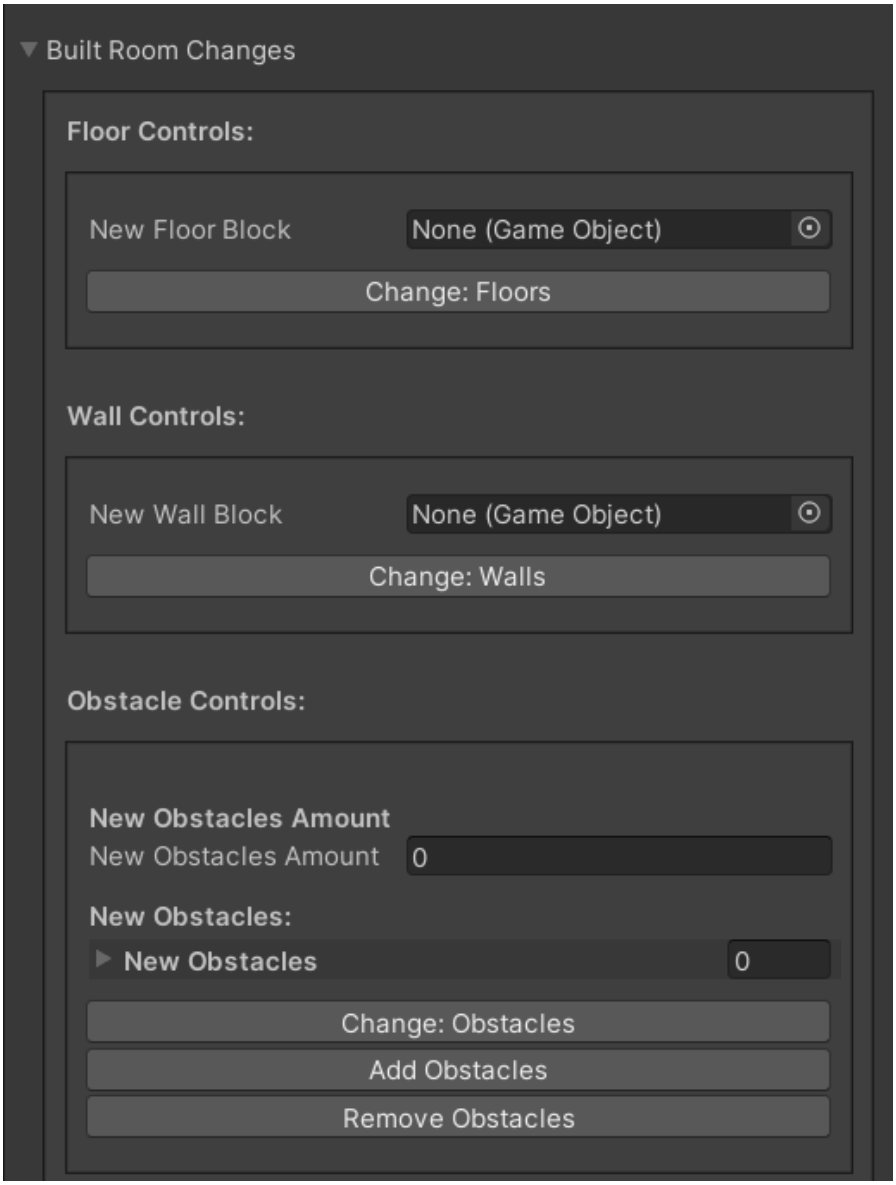
//Corner Spots Variables
CornerObjects - List
cornerFillAmountMin - Int

cornerFillAmountMax - Int

Part Four – Built Room Controls

Built room controls are functions you can call in Editor (or Play) Mode to make changes to the rooms and corridors.

These functions work by executing the functions contained in the BlockyGridRoomControls script of each room. **If AddRoomControls in Additional Settings is false, these functions will not have a function to call, and will not do anything.**



A. **Floor Controls:**

- New Floor Block is the new floor block you want to change for the current ones in every room.
- Change: Floors is a button that changes all the floors in every room.

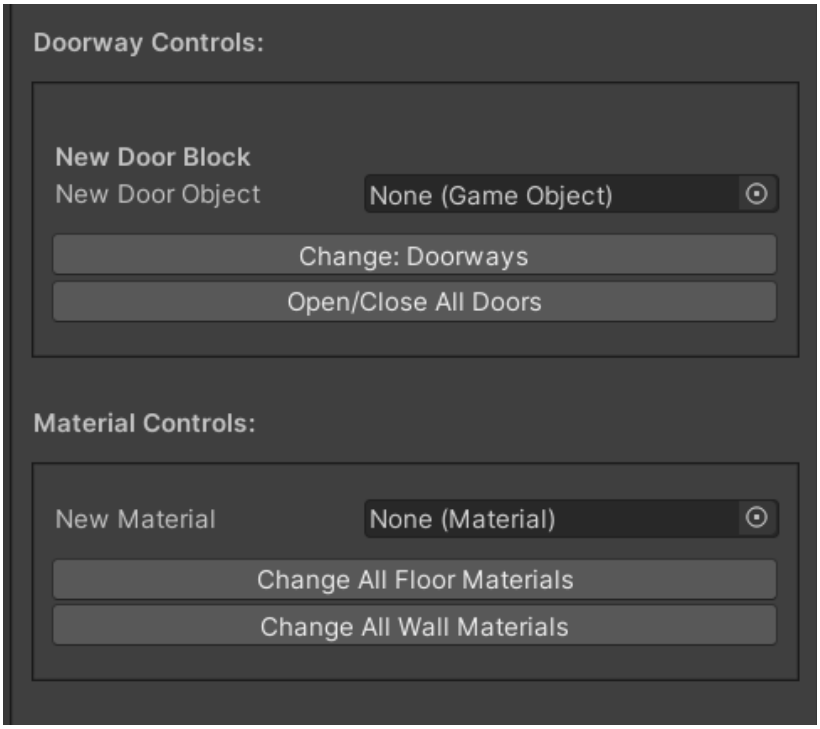
B. **Wall Controls:**

- New Wall Block is the new wall block you want to change for the current ones in every room.
- Change: Wall is a button that changes all the walls in every room.

C. **Obstacle Controls:**

- New Obstacles Amount is the amount of obstacles you want to add or remove from each room (this uses the Obstacles List in Additional Objects).
- Change: Obstacles is a button to change all the obstacles in every room.
- Add Obstacles is a button to add obstacles to every room.
- Remove Obstacles is a button to remove obstacles from every room.

Change and Add both use the Obstacles List in Additional Objects.



D. **New Door Object:**

- New Door Block is the new doorway you want to change for the current ones in each room.
- Change: Doorways is a button to change all the doorway blocks.
- Open/Close All Doors is a button that calls a simple function to activate and deactivate the child of the doorway blocks.

E. Material Controls:

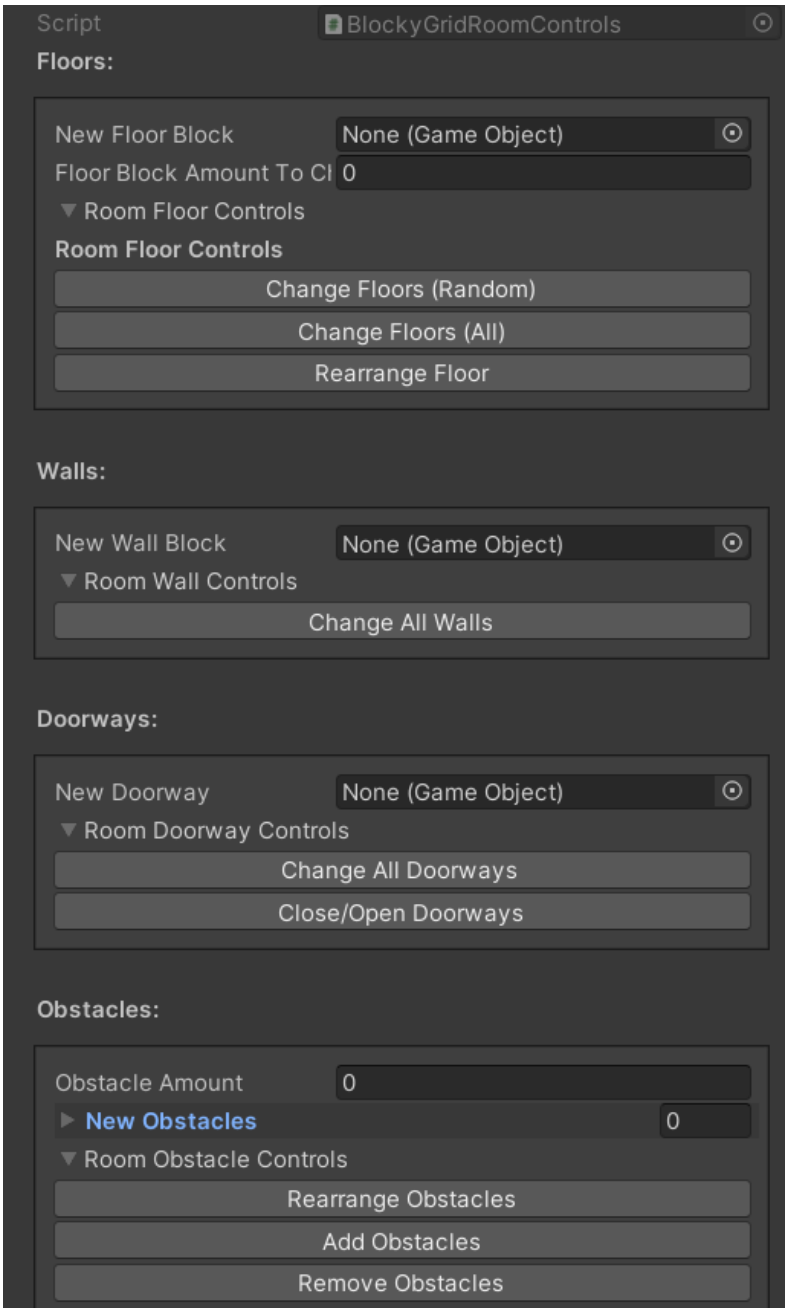
- New Material is the material you want to use when changing.
- Change All Floor Materials is a button to change all the floor materials.
- Change All Wall Materials is a button to change all the wall materials.

Floor changes happen to the floor gameObject, and the floor object of all walls.

Part Five – Individual Room Controls

A more extensive array of functions for making room changes is attached to every room, in the form of the BlockyGridRoomControls script. Simply click on a built room to see it.

For this script to be attached, Add Room Controls in Additional Settings has to be set true.



A. **Floors:**
   - New Floor Block is the new floor block you want to change for the current ones.
   - Change Floors (Random) randomly changes floor blocks to the new block.
   - Change Floors (All) changes all the floor blocks to the new block.
   - Rearrange Floor, rearranges the floor blocks.

B. **Wall:**

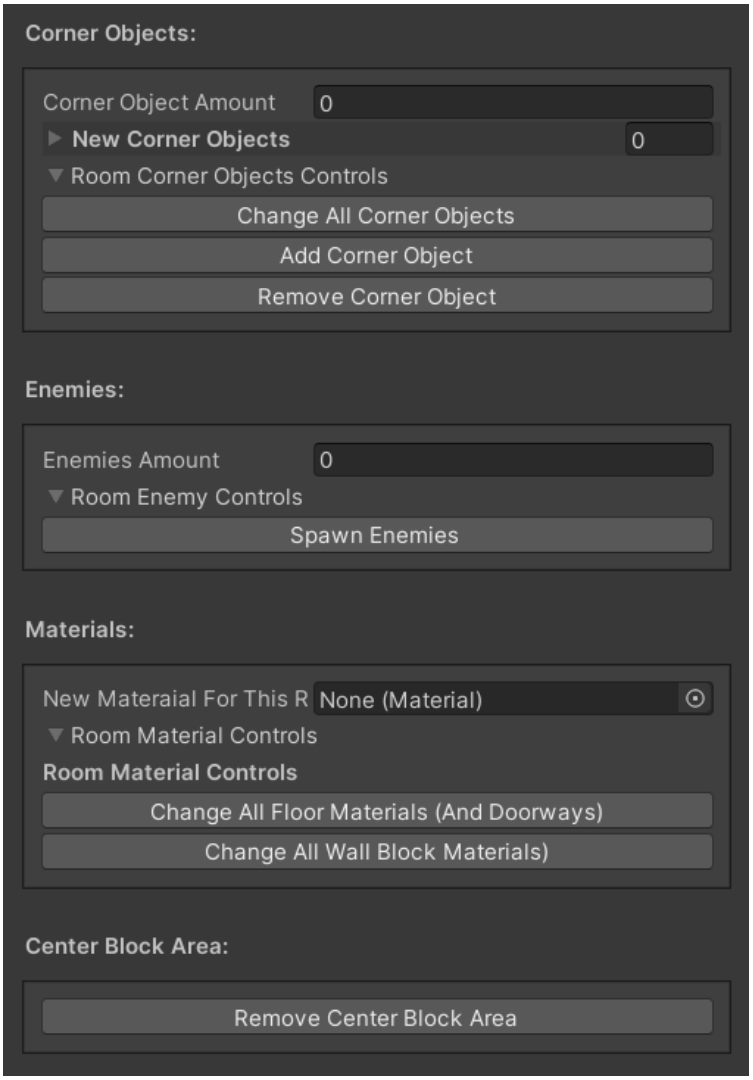
   - New Wall Block is the new wall block you want to change for the current ones.
   - Change All Wall changes all the walls for the new wall block.

C. **Doorways:**

   - New Doorway is the new doorway you want to change for the current ones..
   - Change All Doorways change all the doorway blocks leading into/out of the room.
   - Close/Open All Doors is a button that calls a simple function to activate and deactivate the child of the doorway blocks.

D. **Obstacles:**

   - Obstacle Amount is the amount of obstacles you want to add or remove.
   - New Obstacles is the list of obstacles you want to add.
   - Rearrange Obstacles rearranges the obstacles in the room.
   - Add Obstacles adds obstacles to every room.
   - Remove Obstacles removes obstacles from.



E. **Corner Objects:**

- Corner Object Amount is the amount of corner objects you want to add or remove.
- New Corner Objects is the list of corner objects you want to add or change.
- Change All Corner Objects changes all the corner objects.
- Add Corner Objects adds corner objects.
- Remove Corner Objects removes corner object.

F. **Doorways:**

- Enemies Amount is the amount of enemies you want to spawn in (enemies used are from the main scripts Enemies list).
- Spawn Enemies is used to spawn in the amount of enemies specified by the int above.

G. **Materials:**

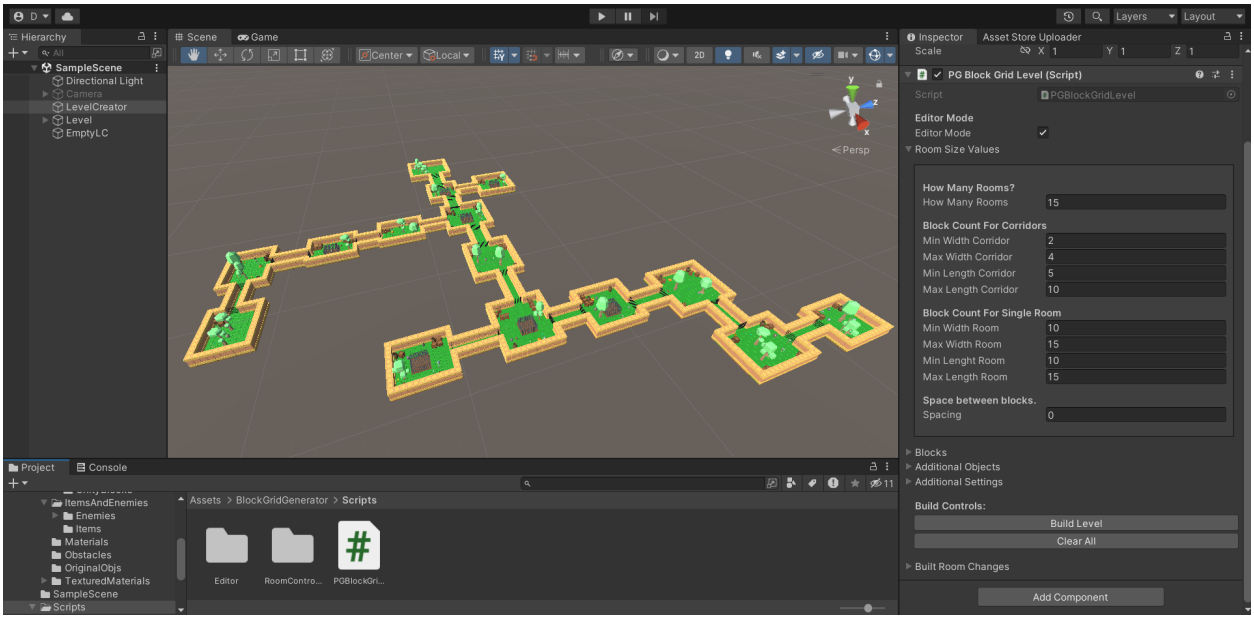
- New Material For This Room, is the material you want to use to change the walls or floors material.
- Change All Floor Materials (And Doorways), will change all the floors material - including the floor blocks under the walls, and the floor blocks under the doorways.
- Change All Wall Block Materials, is used to change all the wall materials - it changes the material of the wall block's first child.

H. Center Block Area:

- Remove Center Block Area, is used when you have a centerpiece, and you wish to edit the blocks beneath - by default, during the creation, rooms with a centerpiece have their center blocks name changed. This is so later instantiations and edits do not place objects in this area and into our centerpiece.

Part Four – Conclusion

So, with the variables we have put in (room amount, sizes, floor block, and wall block), lets run the code (in editor mode) and see what we get:



And there we have a simple set of rooms and corridors ready for whatever project you have in mind. Thank you again for purchasing my tool, and I hope you have tons of fun playing around and building your own stuff.