ProcGenDungeon

Version 0.6

Andrew McAllister

Table Of Contents

Script Overview

ProcGenDungeon is a Unity script designed for generating 2D customisable dungeon levels that are ideal for roguelike games.
The package is inspired by The Binding of Isaac (Edmund McMillen and Florian Himsl) and others like it, offering a system generating connected rooms that have a variety of layouts, enemies, loot and generation settings.

Key features of the package include:

- Two map generation methods (Branch Generation, Drunken Wanderer)
- Choice of how many boss rooms spawn
- Adjustable spawn chances for Mid-Rooms, End Rooms and Special Rooms
- Enemy spawn and respawn settings
- Loot and Enemy pools to spawn from
- The ability to copy the dungeon generation script variables as a scriptable object and use it as an override
- Fully compatible with Unity's Tile Palette tool, ideal for users designing levels with the Tile Brush

How To Install

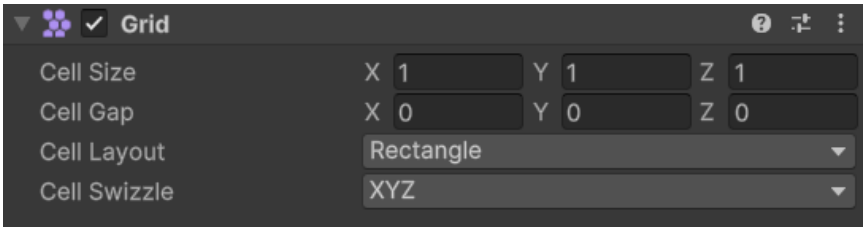
To get the Dungeon Generator set up, you must first:

1. Install and Import the Proc-Gen Dungeons UnityPackage into your Unity Project

2. Create an Empty GameObject in the Hierarchy and name it appropriately (e.g. DungeonGenerator)

3. Attach the ProcGenDungeon.cs script to the Empty GameObject

4. (Optional) If using a Tilemap Palette for Room Layouts, add the Grid Component to your Dungeon Generator GameObject. The values should match the ones shown below.



How To Use

1. Import your room designs into the project OR if using a Tile Palette create your room designs in the scene view.

How to create a room Design:

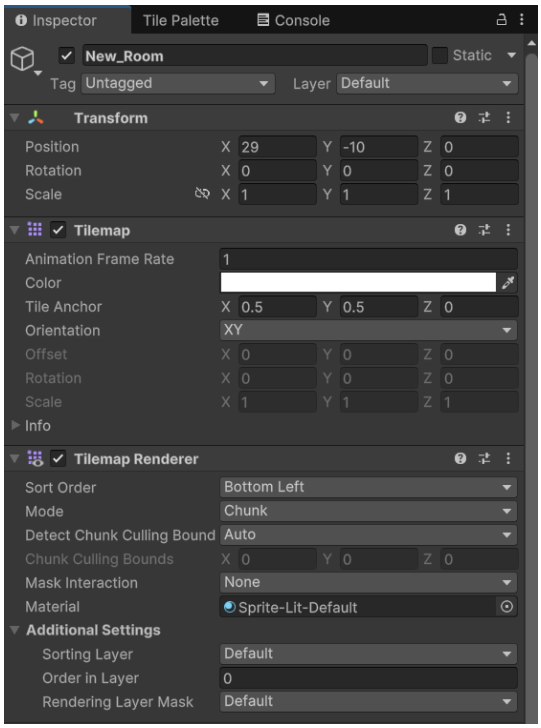
a. Create an Empty GameObject as a child of the DungeonGenerator

b. Add the Tilemap and Tilemap Renderer components to your room design

c. Navigate to Window > 2D > Tile Palette to open the Tile Palette window

d. Import your Tilemap to the Tile Palette and use the Paint Tools to draw your room design in the scene.

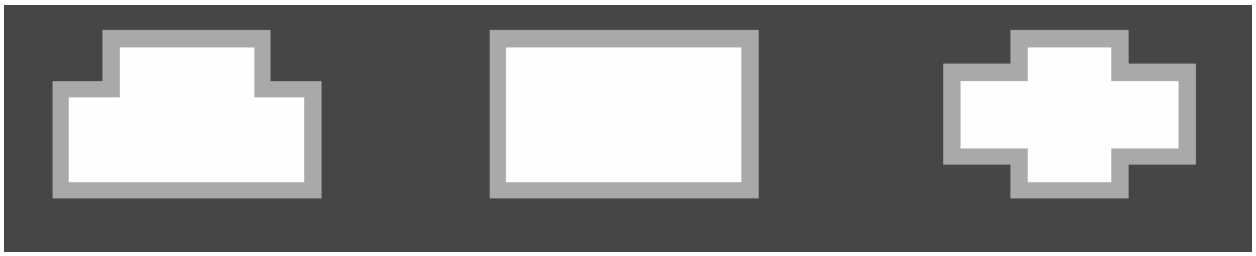
e. Once you've created the Room Design save it as a prefab in your Project Menu.



Example of the Room Design GameObject

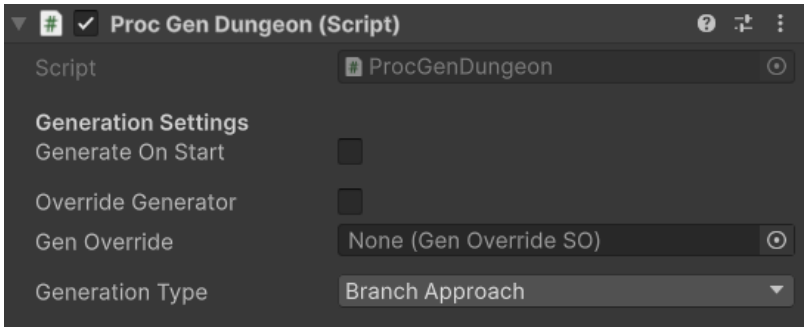
NOTES: The Room Design shouldn't contain any of the door assets, enemies or loot as part of its prefab. GameObjects that you can add to the prefab are ones that aren't expected to be moved or destroyed by the player.

Ensure that the transform positions of the Room Design prefabs are 0, 0, 0.



Examples of some base Room Designs for the dungeon

2. Assign your Generation Settings in ProcGenDungeon.cs



a. Generate On Start (bool) -

   When enabled, the dungeon will start generating at the start of the scene, if disabled the StartGeneration() function can be called during runtime from another script.

b. Override Generator (bool) -

   If enabled will use the values from the Gen Override Scriptable Object assigned instead of the values already stored in the script.

c. Gen Override (Gen Override SO) -

   Assign the Scriptable Object you wish to use for the Override to this variable. This will cause the script to adopt the values of the

Scriptable Object instead of the values stored in the script if the Override Generator bool is enabled.

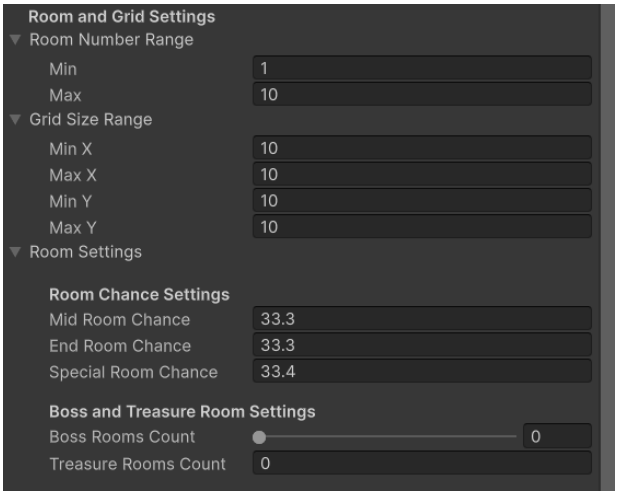
d. Generation Type (Enum) -

Select the generation algorithm that you want the script to follow when building the dungeon.

Branch Approach: This generation type will branch new rooms from an already existing room at random when generating the dungeon layout.

Drunken Wanderer: This generation type will choose the most recent valid room to branch from.

3. Assign the Room and Grid Settings of the Dungeon



a. Room Number Range (IntRange2) -

Insert minimum and maximum values for the script to draw a random number between which will act as the target for how many rooms to spawn in the dungeon.

If you want the target room number for the script to spawn to be an exact number enter that number into both the minimum and maximum values.

b. Grid Size Range (IntRange4) -

Enter a minimum and maximum for both the x and y values of the dungeon grid. The dungeon grid will determine how far away rooms are allowed to spawn from the starting room.

c. Room Settings (Struct) -

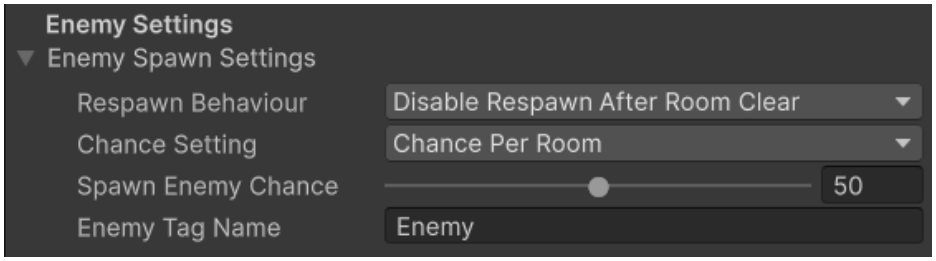
Room Chance Settings:

Input the chance percentage of MidRooms, EndRooms and Special Rooms to spawn. (All of the values should add up to 100%)

Boss and Treasure Room Settings:

Input how many Boss Rooms (max of 3) and treasure rooms that will spawn in the dungeon.

4. Select your Enemy Spawn Settings that you want your dungeon's enemies to follow



a. Respawn Behaviour (Enum) -

   Always Respawn Enemies: Enemies in the room will always respawn when the player enters, even if they were previously defeated.
   Disable Respawn After Room Clear: Enemies will respawn only if the player left the room without defeating all the room's enemies. Once the room has been cleared, enemies will no longer respawn when the player re-enters.

   No Enemy Respawn: Individual enemies won't respawn after being defeated even if the player re-enters the room.

b. Chance Setting (Enum)

   Chance Per Room: If selected, the room will either add all of the possible enemies entered in the RoomLayout's Enemy Spawns or not add any based on the Spawn Enemy Chance.

   Chance Per Position: The Spawn Enemy Chance is applied to each individual Enemy Spawn in the RoomLayout resulting in some positions in the rooms spawning enemies and some not, unlike Chance Per Room where either all positions spawn enemies or none will.

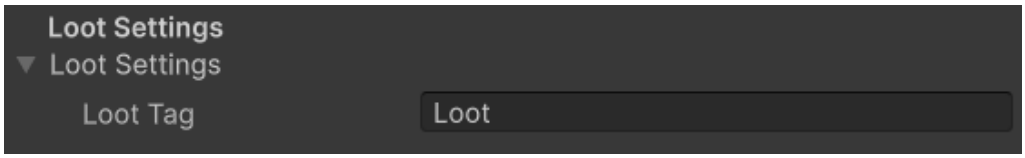
c. Spawn Enemy Chance ([Range(0, 100)] Int) -

   Use the slider to adjust the chance of enemies to spawn for either each spawn position or per room based on the Chance Setting.

d. Enemy Tag Name (String) -

   Type the name of the tag you want to assign to your dungeon's enemies.

5. Assign the name of the tag you want to assign to loot items in the loot settings.



6. Create RoomLayouts to use as rooms when building the dungeon

    a. Room Name (String) -

       Enter the name of the room to distinguish each room.

    b. Layout Prefab (Game Object) -

       Assign one of the Room Designs from your Project Window.

    c. Type Of Room (Enum) -

       Select the room type of the RoomLayout. It is recommended to include at least one layout for each room type in your RoomLayouts list.

    d. Grid Size (Vector2Int) -

       Specify the width and height of the room layout in grid tiles. Each tile in your Layout Prefab counts as 1 unit in the grid.

    e. Enemy Pool (Enemy List) -

       Assign an Enemy List from the Project Window to draw enemies from that you want to spawn in the RoomLayout.

       Enemy Lists can be created via Assets > Create > ProcGen > Enemy List.

NOTE: When adding enemies to the Enemy List, there are two types you can use:

Single Enemy: Simply drag and drop a single enemy prefab from the Project Window into the list.

Enemy Group: Create an empty GameObject in the Scene, add multiple enemy GameObjects as its children, and then turn the entire setup into a prefab in the Project Window. Drag that prefab into the list to spawn the entire group together.

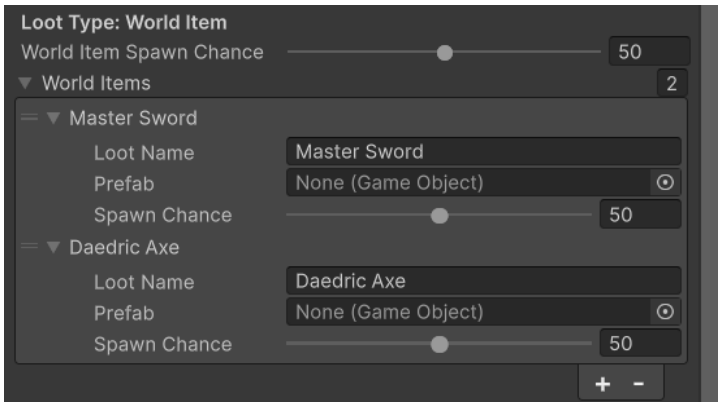
f. Enemy Spawns (List<Vector2Int>) -

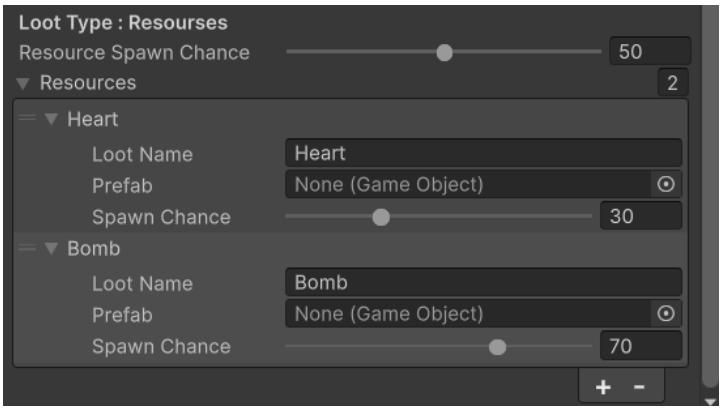
Fill this list with the grid positions of the room that you want your enemies to spawn .

g. Room Loot (List<LootPositions>) -

Loot Pool: Assign the LootList Scriptable Object that you want to draw world items and resources from. You can create a LootList via Assets > Create > ProcGen > LootList.

NOTE: World Item Spawn Chance and Resource Spawn Chance should add up to 100%, this also applies to each individual item in either list.





Chance To Spawn: Set the percentage chance for this entry to spawn a loot item at the spawn position.

Spawn Position: Put the coordinates of the tile that you want the item to spawn should the spawn chance trigger.

h. Door Data (List<DoorData>) -

Door Prefab: Enter here the prefab of the door you want to spawn which will lead onto the next room.

Door Direction: Select the direction that this door is going to lead in your dungeon. It is not recommended to have two doors that go in the same direction per room.

Door Position: Assign the values of the coordinates you want the door to spawn from the room's local position.

How To Use The Doors

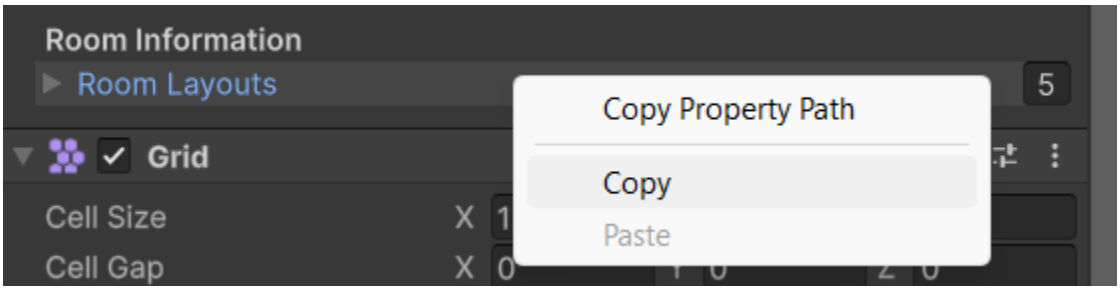
For the doors to be used, the OpenDoor() coroutine has to be called within the DungeonDoor.cs script. As it is, this is called by using the arrow keys but for a game you'd probably want to open the door with the player character.

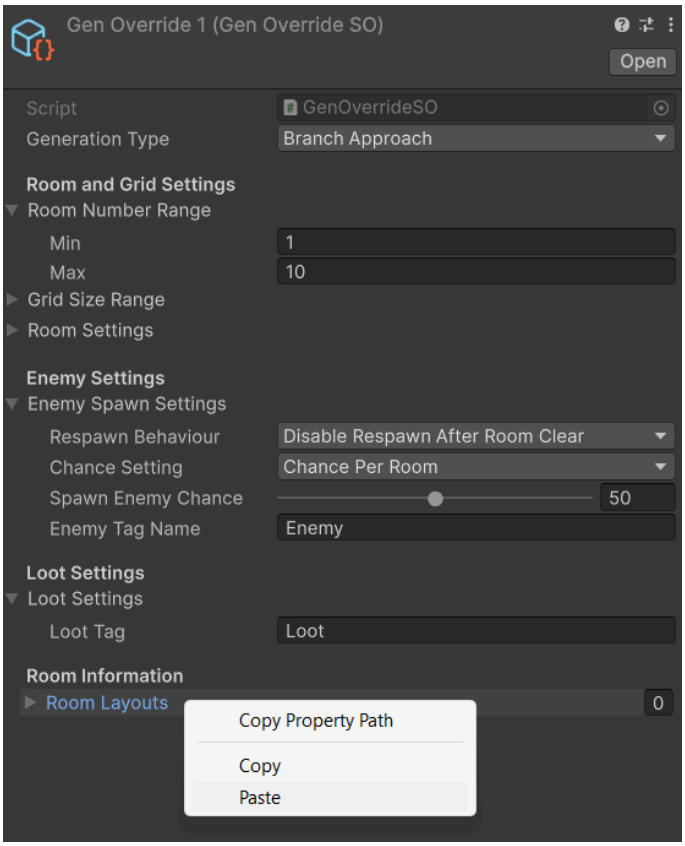
```
void Update()
{
    if (Input.GetKeyDown(KeyCode.UpArrow) && doorDirection == DoorDirection.Up)
    {
        OpenDoor();
    }
    else if (Input.GetKeyDown(KeyCode.RightArrow) && doorDirection == DoorDirection.Right)
    {
        OpenDoor();
    }
    else if (Input.GetKeyDown(KeyCode.DownArrow) && doorDirection == DoorDirection.Down)
    {
        OpenDoor();
    }
    else if (Input.GetKeyDown(KeyCode.LeftArrow) && doorDirection == DoorDirection.Left)
    {
        OpenDoor();
    }
}
```

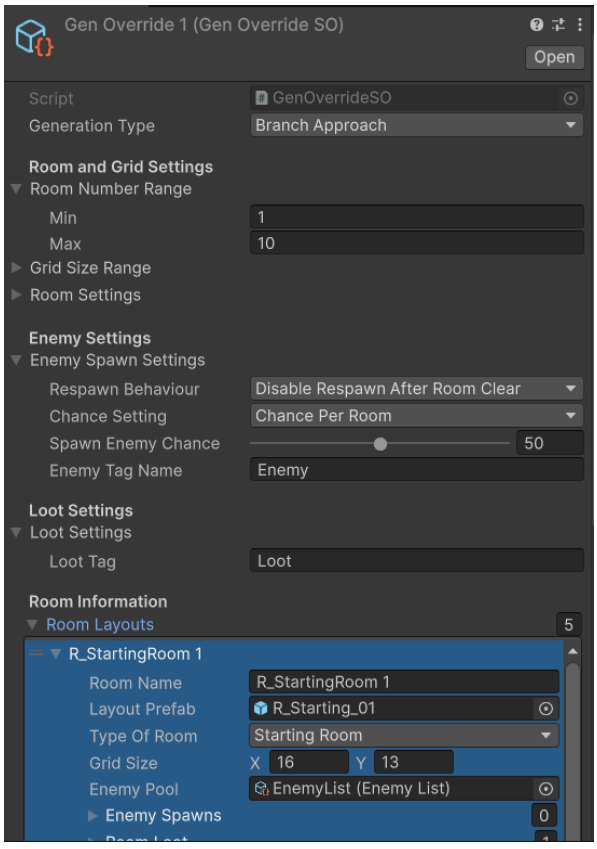
As a tip, you could attach a Trigger Collider to your door prefab that triggers a bool whether or not the player is in the radius. From there you can replace the current code in the Update() method with one that triggers OpenDoor() if the player presses the open input button.

How To Save Script Data As An Override

With the Scriptable Object, GenOverride that you can create via Assets > Create > Proc Gen > Generator Override. Then with each variable (e.g. Enemy Spawn Settings, Room Number Range, Room Layouts) you can copy and paste the data from the Script onto the Scriptable Object.







Future Features

- Loot Chests:

  Currency chests can be added into the loot pool but won't work properly in game unless the chest gameobject is deleted from the scene when the item is collected.

- Lockable Doors:

  I'll add a setting that will let the player choose the lock behaviour for doors. Examples: Doors never lock, Doors unlock upon room clearance.

- Larger room options and more than one door in each direction:

  I will find a method to allow the generator to spawn larger rooms (2x2, 1x3) that allows the user to comfortably add more than 4 doors for each room.

Contact

If you have any suggestions for future features or need assistance with any problems you may find with the tool you can contact me:

Email: amcallister789@gmail.com
Website: WIP
LinkedIn: linkedin.com/in/andrew-mcallister-b4b602258/