The Dungeon Generator is a Unity tool to quickly generate random levels best used for first person experiences. This document will explain how to use the Dungeon Generator, how to customize the Dungeon Generator by adding your own models and how to apply the generated levels to your own work.

We have made certain choices during the programming of the scripts that we think will give the most freedom in generating usable and interesting dungeons. We have spent many an hour walking through randomly generated dungeons checking and double checking the objects, corridor layouts and room buildups. We decided early on to solely focus on the building of a really interesting environment that can be used by people to build their games in and around. We already received the question: "where are the monsters?" a lot of times, but we have left that part up to you. We hope you enjoy using the dungeon generator to provide settings for your games. If you see anything that raises your eyebrows (there have to be bugs left) or if you really would like to see some additional choices, give us an email and convince us to add them ;)

***D4Games** and **Digital Forest**.*

1. Dungeon Generator settings

Using our tool

The Dungeon Generator tool is a script, allowing you to go through all the different settings and options to generate unique dungeons. Different settings will generate dungeons with different layouts, sizes and structures. Here we go over all the different settings to guide you in the use of the tool. We will also explain some of the background of the choices you can make.

- Open the scene **'EmptyScene'** This is located in the **RandomDungeonGenerator/Scenes** folder
- Click on the '**DungeonGenerator**' GameObject in the Hierarchy window and go to the DungeonGenerator script in the inspector.

The DungeonGenerator script is the tool that creates a Random Dungeon and where all the 'magic' happens!

Follow these steps:

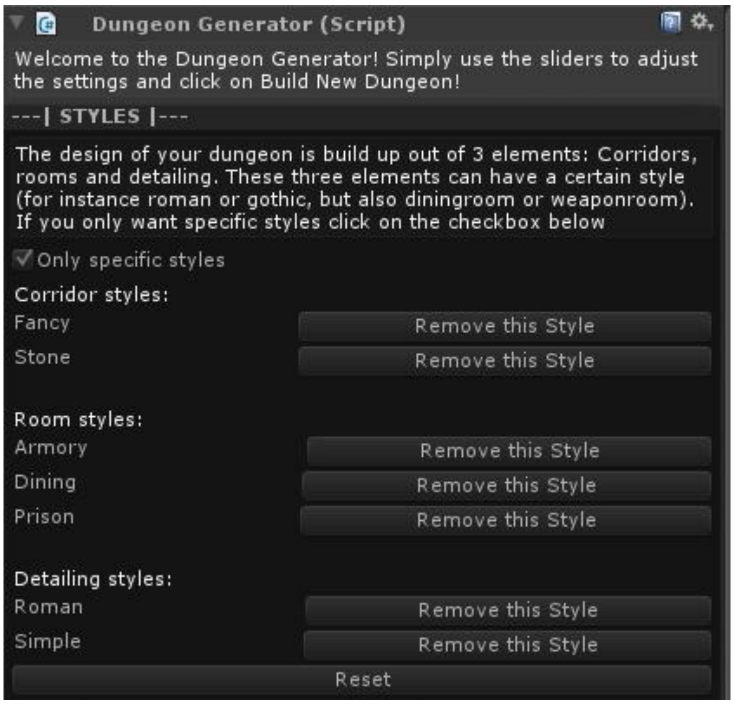
Step 1: styles

The Dungeon Generator tool works using different styles. All elements in the dungeon will be generated according to these styles. We have made a distinction between Corridors, Rooms and Detailing.

Corridors form the main path of the dungeon

Rooms are created from corridors

Detailing is a style for more decorative models like railings, staircases, portals and windows.


Dungeon Generator (Script)

Welcome to the Dungeon Generator! Simply use the sliders to adjust the settings and click on Build New Dungeon!

---| STYLES |---

The design of your dungeon is build up out of 3 elements: Corridors, rooms and detailing. These three elements can have a certain style (for instance roman or gothic, but also diningroom or weaponroom). If you only want specific styles click on the checkbox below

✓ Only specific styles

Corridor styles:

Fancy	Remove this Style
Stone	Remove this Style

Room styles:

Armory	Remove this Style
Dining	Remove this Style
Prison	Remove this Style

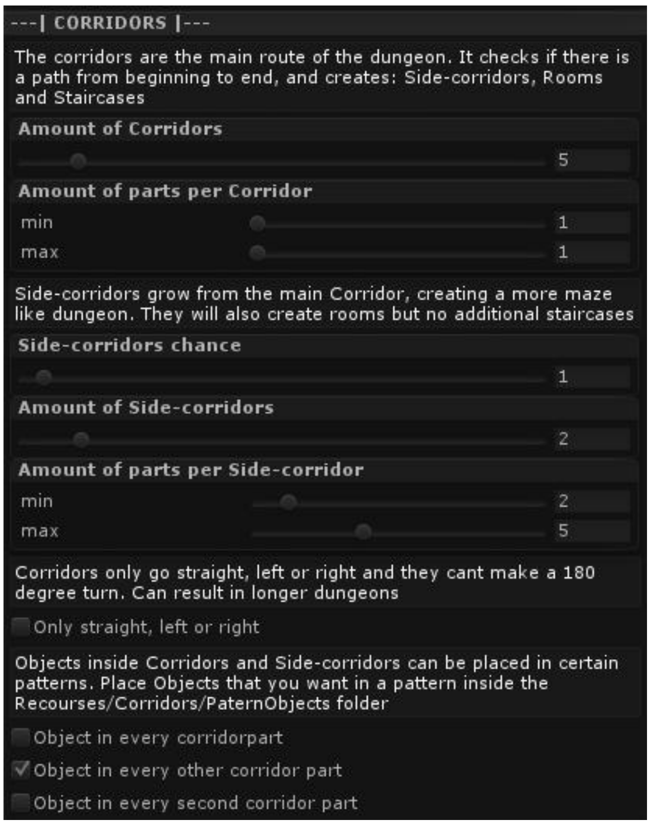
Detailing styles:

Roman	Remove this Style
Simple	Remove this Style

The tool will give you the option to use all styles and will automatically look for all the different styles it reads from the ***Resources*** folder. For more information about these styles and the naming conventions go to chapter 2 '**changing the dungeon generator**'. Clicking on 'Remove this Style' will completely remove all objects used by this style when generating a new dungeon.

Once all corridors, staircases and rooms are generated the generator will check all attached corridors and rooms and set them to the same style. This will generate zones in the same style and give better defined boundaries between styles. The detailing style is the boundary between these styles.

Step 2: Corridors



---| CORRIDORS |---

The corridors are the main route of the dungeon. It checks if there is a path from beginning to end, and creates: Side-corridors, Rooms and Staircases

Amount of Corridors 5

Amount of parts per Corridor
min 1
max 1

Side-corridors grow from the main Corridor, creating a more maze like dungeon. They will also create rooms but no additional staircases

Side-corridors chance 1

Amount of Side-corridors 2

Amount of parts per Side-corridor
min 2
max 5

Corridors only go straight, left or right and they cant make a 180 degree turn. Can result in longer dungeons

☐ Only straight, left or right

Objects inside Corridors and Side-corridors can be placed in certain patterns. Place Objects that you want in a pattern inside the Recourses/Corridors/PaternObjects folder

☐ Object in every corridorpart
✓ Object in every other corridor part
☐ Object in every second corridor part

Main Corridor length

The dungeon is built by starting with a main corridor. These settings will determine the total size of the dungeon. The settings are split in two categories.

First you determine the amount of separate corridor parts and then the minimum and maximum length of each corridor part.

After each corridor part the tool will generate a new random direction of the corridor. So if you want short corridors with a lot of corners set the amount high, but the minimum and maximum length short and if you want long corridors without a lot of corners set the amount lower with higher minimum and maximum values.

Side Corridor setup

The first slider will determine if there is any chance for a side corridor to start. Each main corridor part will check to this chance. So if you have a high chance a lot of side corridors will be generated.

Each side corridor has a starting point on the main corridor and will branch off with the same type of settings as the main corridor. Each side corridor will have a setting for the amount of parts and then the minimum and maximum length of each part.

Side corridors can generate rooms but will not generate additional staircases or more side corridors.

Corridor Objects

We have made it possible for a user to select a certain pattern for objects to be placed in corridors. These 3 patterns are: ‘every corridor part’, ‘every other corridor part’ and ‘every second corridor part’. You can select all 3, or a sub selection.

We use these pattern objects mainly for light sources to make sure the corridors are well lit. But other uses can be thought of, like support beams and such. Remaining object spaces are placed according to the percentage slider for objects later in the list.

Step 3: Rooms

To generate rooms in your dungeon you first select the chance for a *Room* to spawn. Each corridor part will check to this chance including side corridor parts. A *Room* will then choose a random direction in which it will be build.

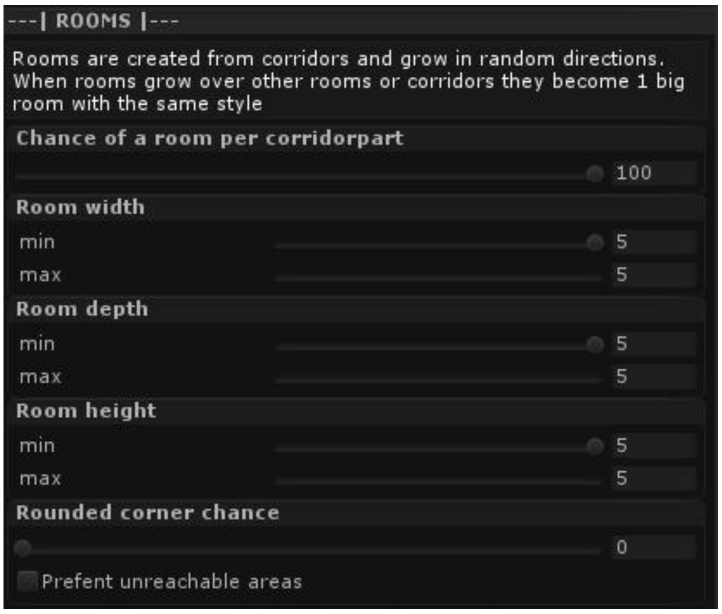
Setting the minimum and maximum size will determine the size of the room.

Setting a high chance will generate a lot of Rooms and this will mean that the rooms will overlap a lot, resulting in larger rooms. The rooms will overwrite corridor parts, but will generate floors in these spots, creating interesting walkways.

Then there is a slider for the chance that corners will generate so called round corners. This will result in less cubic looking dungeons.

The last setting is an option for the dungeon, to build floors on each room part that is generated out of one Room-Builder, to prevent area's to be build that are out of reach by a corridor.

The dungeon will always be generated with a guaranteed path from the start-point to the end-point. But sometimes rooms can overlap and generate side area's that are not normally reachable. The dungeon will be more open without this selection.


---| ROOMS |---

Rooms are created from corridors and grow in random directions. When rooms grow over other rooms or corridors they become 1 big room with the same style

Chance of a room per corridorpart: 100

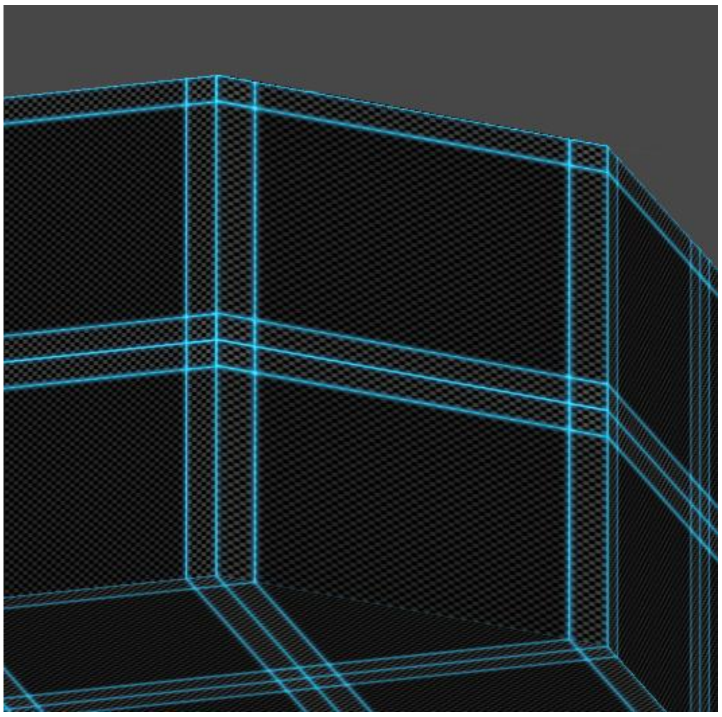
Room width
min: 5
max: 5

Room depth
min: 5
max: 5

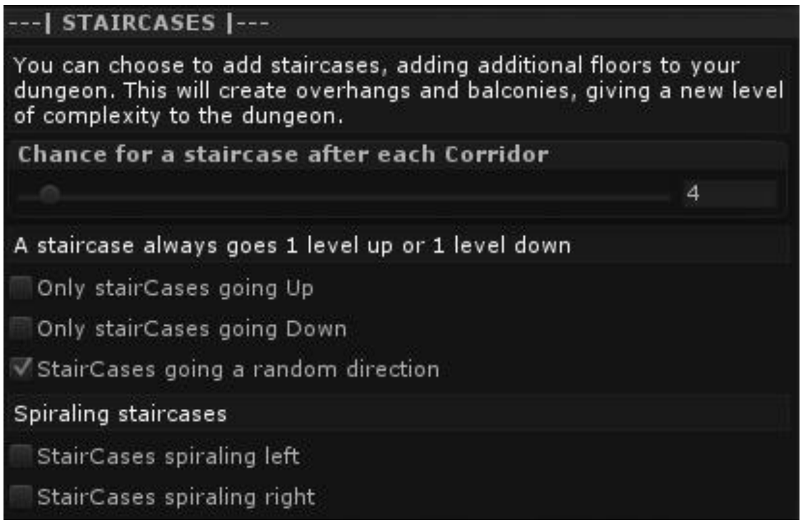
Room height
min: 5
max: 5

Rounded corner chance: 0

☐ Prefent unreachable areas



Step 4: Staircases


---| STAIRCASES |---

You can choose to add staircases, adding additional floors to your dungeon. This will create overhangs and balconies, giving a new level of complexity to the dungeon.

Chance for a staircase after each Corridor: 4

A staircase always goes 1 level up or 1 level down

- [ ] Only stairCases going Up
- [ ] Only stairCases going Down
- [x] StairCases going a random direction

Spiraling staircases

- [ ] StairCases spiraling left
- [ ] StairCases spiraling right

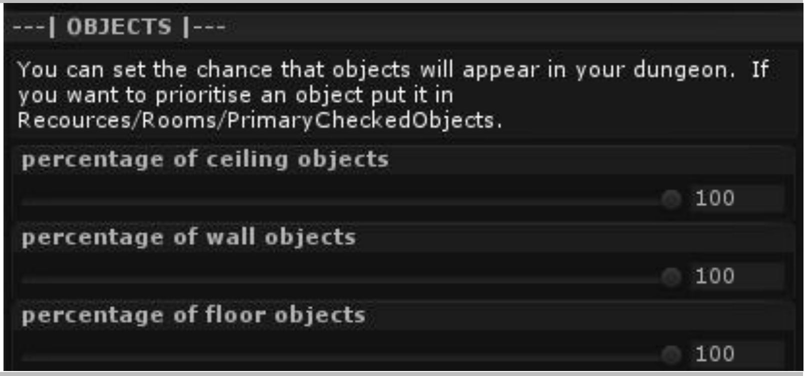
First you select the chance a staircase is built after each main corridor part. After each corridor is built it will check to this chance to try and build a staircase.

Staircases can be set to either go only up, down or a random direction.

Then you can set the direction in which the staircases will grow. You can choose either right or left to generate spiraling dungeons.

Staircases will add a lot of additional depth to the dungeons. Be aware that the dungeon generator will always try and position a corridor after the staircase is placed so the path will never get stuck.

Step 5: Objects


---| OBJECTS |---

You can set the chance that objects will appear in your dungeon. If you want to prioritise an object put it in Recources/Rooms/PrimaryCheckedObjects.

percentage of ceiling objects: 100

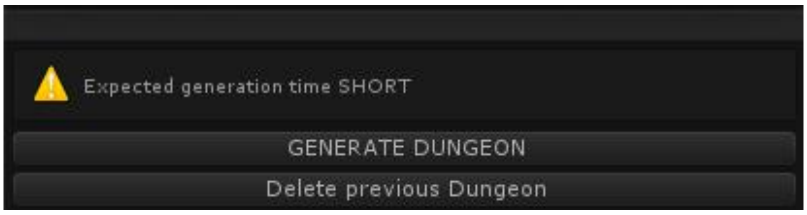
percentage of wall objects: 100

percentage of floor objects: 100

We have made a sub selection of 3 types of objects in relation to their placement. Here you can select the chance of each of these 3 types being: Ceiling, wall and floor objects.

We have made a distinction between primary objects and normal objects. For corridors these primary objects are placed in a pattern. For rooms these primary objects are chosen first and are placed more often if there is room for them. These are used for style defining objects. In our current version they are the prisons inside the prison style. The individual placement options for Objects are handled in Chapter 2 under ***'The object script'***

Step 6: Generate


Expected generation time SHORT

The generator will give you a guesstimation about the time it might cost to generate a new dungeon using the current settings. If you are happy with the generated dungeon you can press play to try it out or delete it and generate a new one. Be aware that generating a new one will also delete the previous dungeon.

As we have chosen for the dungeons to be generated in Edit-Mode unity will actually freeze during the generation time. If you have selected a very big dungeon we suggest a cup of coffee while you wait.

Using pre-generated scenes

We provided some scenes with the Dungeon Generator to give you some idea of how different a dungeon can become with just changing the settings.

Simply open one of the scenes found in: **RandomDungeonGenerator/Scenes/**

All the scenes contained are nothing more than a generated dungeon with a player in it, and the DungeonGenerator gameobject with the DungeonGenerator script attached.

Press play to test the dungeon, or go to the DungeonGenerator script, scroll completely down and press ‘GENERATE’ to generate a new dungeon. Each scene has different generate settings.

Each time you generate a new dungeon, the Level prefab (with the DGRandomLevel tag) will be completely cleared and be refilled with a new random dungeon. So be careful!

Setting up a scene from scratch

- Make sure you've got the DungeonGenerator asset inserted into your project.
- Create or open a scene.

*You don't necessarily need an empty scene to work in, but the dungeon might be (visually) generated over objects you've already got in your **hierarchy**. The complete level will however be generated in its own gameobject and can therefore be easily moved.*

- Drag the **DungeonGenerator** prefab and the **DGLevel** prefab into the **hierarchy** screen. These can be found here: **RandomDungeonGenerator/BuildingBlocks/**

*The important thing about the **DungeonGenerator** prefab is that it has the DungeonBuilder tag attached and the DungeonGenerator script component attached.*

*The **DGLevel** prefab has the DGRandomLevel tag attached.*

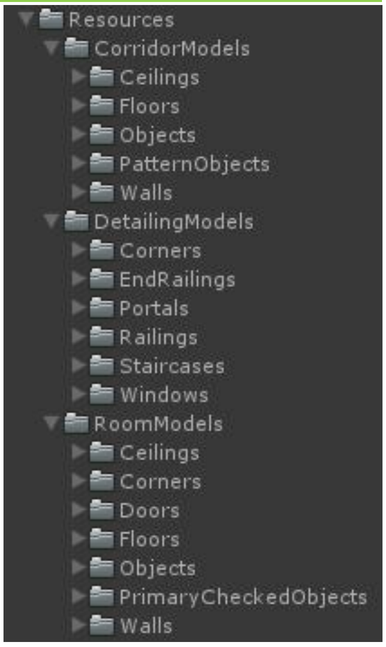
- Select both Prefabs and in the Unitymenu select GameObject/Break Prefab Instance.
- Select the DungeonGenerator gameobject and use the inspector to setup the dungeon parameters. You are now ready to generate a dungeon.

The dungeon will be generated by the script attached to the DungeonGenerator gameobject and will be placed in the DGLevel gameobject.

2. Changing the Dungeon Generator

The Dungeon Generator is built up out of corridors, rooms and detailing, which are again built up from smaller parts like: walls, floors, doors, ceilings, objects and so on. If you change these elements you can completely change the look and feel of your dungeon.

Customizing the model database



All the models that are used to generate a dungeon can be found in the **Resources** folder. The Dungeon Generator uses the resource folder to populate its database. Each folder has a specific name which divides all the models into different parts. The Dungeon Generator generates the **Corridors** first, then the **Rooms** and finally the **Detailing** models.

The styles that come with the Dungeon Generator are located in the FantasyPackage/Resources folder.

Except for the Objects, PatternObjects and PrimayCheckedObjects folders they all contain the GameObjects that are actually placed in the dungeon. *If you change these models, you will change the look of the dungeon.*

The Object folders work a bit differently. They contain GameObjects that all have the Object script attached. In the object script you can choose which Prefab to place, change the size of the object and how it is placed in the dungeon.

We chose to do this to speed up the generating of the dungeons, and so that you can easily edit or replace the Prefab without touching the settings.

The Prefabs we used for the Dungeon Generator can be found in the FantasyPackage/PrefabsUsedInDungeon folder.

Naming conventions

The Prefabs inside each folder starts with the '**style**' name, then a '_' and then the rest of the name. The Dungeon Generator uses the first part of the name to collect the Prefabs of the same style. If you would for instance rename of *Armory_Table* to *Prison_Table* and generate a new dungeon, the armory table will be placed in the prison rooms instead.

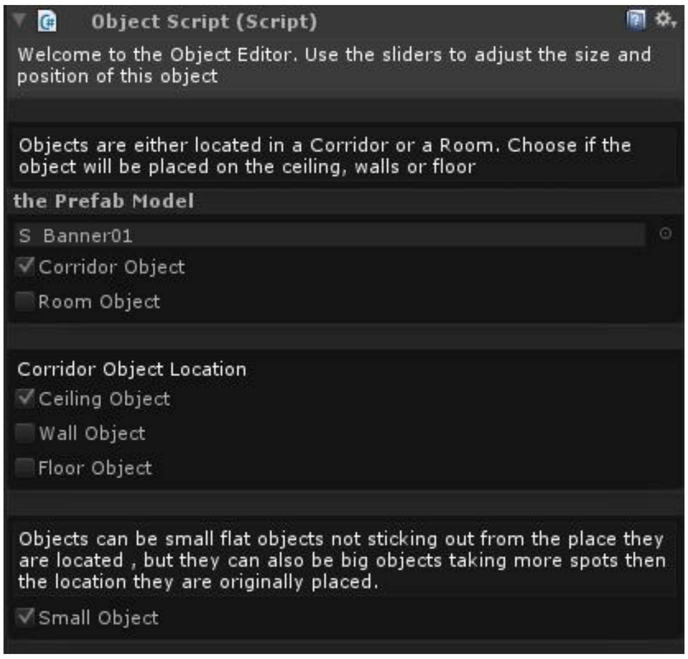
All objects will be stored in an object list. During dungeon generation this list will be used to pick random models. If you want a specific object to have a bigger change to be placed in a dungeon you can add '=' with a number to add more of this object in the list. For example you can add '=3' after the name and the chance to be chosen will triple.

The object script

The **Objects**, **PatternObjects** and **PrimayCheckedObjects** folders all contain empty GameObjects with the Object script attached. With this script you can choose which Prefab to generate, set the size of that Prefab and how it is placed in the dungeon. Here we will go through all these options:

- Drag the prefab that you want to be part of the dungeon into the first field under the text 'the Prefab Model'.

If there was no Prefab chosen a bunch of new options will appear after doing so.


Object Script (Script)

Welcome to the Object Editor. Use the sliders to adjust the size and position of this object

Objects are either located in a Corridor or a Room. Choose if the object will be placed on the ceiling, walls or floor

the Prefab Model

S Banner01

✓ Corridor Object

Room Object

Corridor Object Location

✓ Ceiling Object

Wall Object

Floor Object

Objects can be small flat objects not sticking out from the place they are located , but they can also be big objects taking more spots then the location they are originally placed.

✓ Small Object

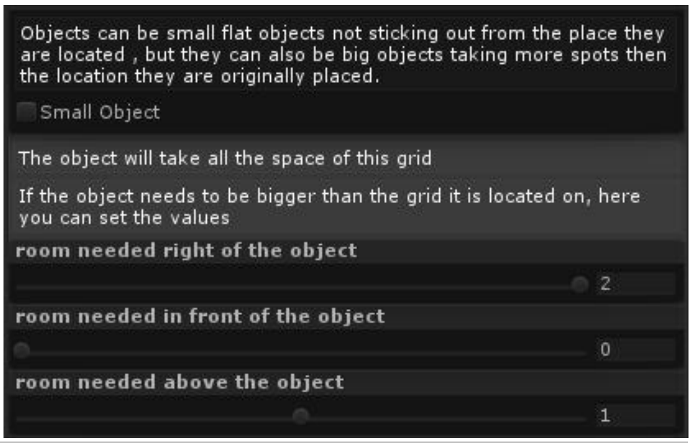
- Choose if it is a corridor or a room object. If the script is located in the **CorridorModels** folder select **'Corridor Object'**, if it is located in **RoomModels** select **'Room Object'**.

If you select Corridor while it is located in a Room or vice versa the object will not be part of the dungeon.

- Choose the Object Location. Selecting **'Ceiling Object'** will place the object hanging from the ceiling. Selecting **'Wall Object'** will place the object hanging on a wall. Selecting **'Floor Object'** will place the object standing on the floor.

You will have different settings per choice of object location.

- Select if the object is a **'Small Object'**. If you select **'Small Object'**, the object will take little to no amount of space, and be assumed to have no Collider e.g. a painting on a wall or some small rubble on the floor. Multiple small objects can be placed in one ‘part’.


Objects can be small flat objects not sticking out from the place they are located , but they can also be big objects taking more spots then the location they are originally placed.

Small Object

The object will take all the space of this grid

If the object needs to be bigger than the grid it is located on, here you can set the values

room needed right of the object

2

room needed in front of the object

0

room needed above the object

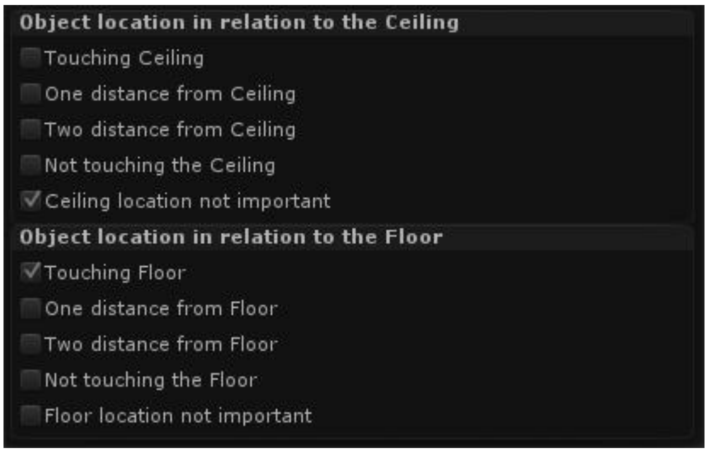
1

If you do not select 'Small Object' it will at least take all the space in the ‘part’ it will be generated on, which means no other objects will be placed on that ‘part’. You will also get extra options:

Increasing the room needed to the **right**, **in front** or **above** the object will make the object bigger than 1 ‘part’. These values have to correspond with the size of the prefab.

One part is 5x5x5 units where there will be 0.5 removed from the height if there's a ceiling in that part. So for example if you have an object that is 8 units wide 4 units high and 2 units deep, you should change the 'room needed right of the object' to 1, and the rest 0, giving it a space of 10x5x5 units.

For more information about large objects and how to make them fit your dungeon read the **Replacing the template models** chapter.



Object location in relation to the Ceiling

☐ Touching Ceiling
☐ One distance from Ceiling
☐ Two distance from Ceiling
☐ Not touching the Ceiling
✓ Ceiling location not important

Object location in relation to the Floor

✓ Touching Floor
☐ One distance from Floor
☐ Two distance from Floor
☐ Not touching the Floor
☐ Floor location not important

To give some control to the placing of the object in the dungeon we provided some choices. You will have different choices per 'Object location'

- ***Object location in relation to the ceiling***

Touching Ceiling means it HAS to touch a ceiling

One distance from Ceiling means it touches no ceiling but has at least 1 ceiling part at one distance

Two distance from Ceiling it has no ceiling for at least 2 parts above it and at least 1 ceiling part at two distance

Not touching Ceiling means it can't touch a ceiling and will always remain at least one distance away from a ceiling

Ceiling location not important means it doesn't matter if the object touches a ceiling or not

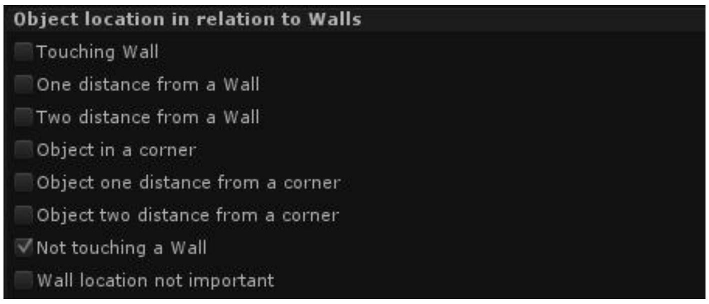
- ***Object location in relation to the floor***

Touching Floor means it HAS to touch a floor

One distance from Floor means it has no floor contact and at least 1 floor part at one distance

Two distance from Floor it has no floor for at least 2 parts below it and at least 1 floor part at two distance

Not touching Floor means it can't touch a floor and will always remain at least one distance away from a floor



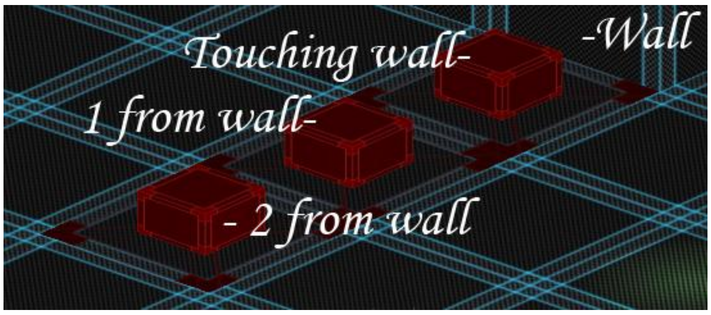
Object location in relation to Walls

☐ Touching Wall
☐ One distance from a Wall
☐ Two distance from a Wall
☐ Object in a corner
☐ Object one distance from a corner
☐ Object two distance from a corner
✓ Not touching a Wall
☐ Wall location not important

Floor location not important means it doesn't matter if the object touches a floor or not

- ***Object location in relation to Walls***

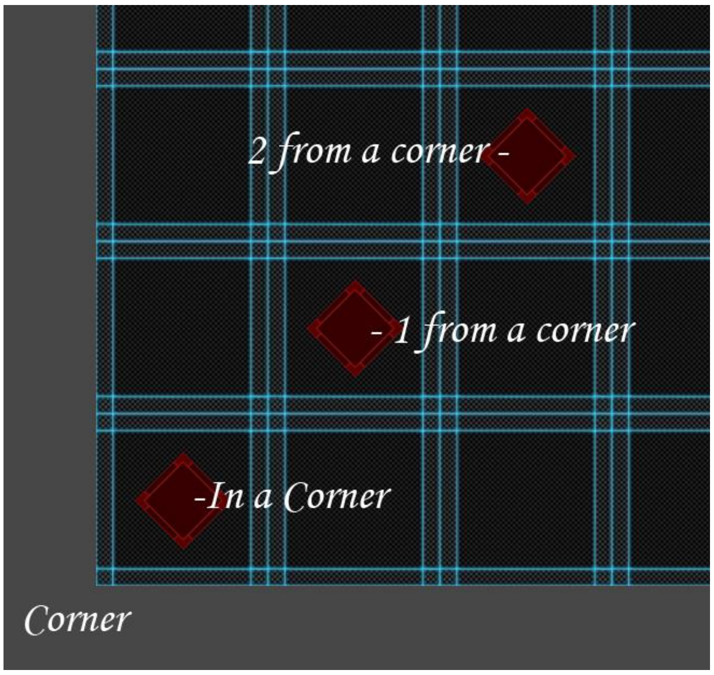
Touching Wall means it HAS to touch a wall



One distance from Wall means it touches no wall directly and has at least 1 wall part at one distance

Two distance from Wall it has no wall for at least 2 parts next to it and at least 1 wall part at two distance

Object in a corner will place the object if there is a corner directly next to it. If the object is no bigger that default it will also rotate the object 45 degrees making it look away from the corner.



Object one distance from a corner will place the object if there is a corner one away from it. If the object is no bigger that default it will also rotate the object 45 degrees making it look away from the corner.

Object two distance from a corner will place the object if there is a corner two away from it. If the object is no bigger that default it will also rotate the object 45 degrees making it look away from the corner.

Not touching Wall means it can't touch a wall and will always remain at least one distance away from a wall

Wall location not important means it doesn't matter if the object touches a wall or not .

The idea of all these settings is to be able to imagine a 'perfect' scene with all your objects in their best spots. Then using these settings to fix them in their 'perfect' locations and randomize it for constantly different settings but with the underlying 'perfect' scene still present. Setting these locations will also reduce the amount certain objects will appear in your scene. Especially small objects can appear almost everywhere. Forcing these objects into corners or only 2 away from a wall or ceiling will greatly reduce the amount of these objects, reducing the rinse-repeat feel of the dungeon. We also added some small additional scripts to randomize rotation and sometimes delete parts of the object. For example: the amount of cutlery on the dinner tables prefabs and the positioning of the chairs around them.

Adding your own models

Let's say you have a treasure chest 3D-model and want it to be generated into the dungeon. Find one of the Prefabs in the **FantasyPackage/PrefabsUsedInDungeon** that has a similar shape and size (for instance the **A_Table**)

- Drag the **A_Table** into the scene and open the parent Gameobject. Put the treasure chest model into the opened Gameobject, so that its bottom is at the same location of the table's bottom. Center the chest so that it overlaps the table.
- Delete the table inside the **A_Table** parent.
- Rename **A_Table** to anything you want.

*The first part of the name is not important, but makes it easier to locate it later, so we suggest you name it something like **A_Chest**.*

- Make it into a new Prefab.

Now that the prefab is ready we need to add an object in the **FantasyPackage/Resources** folder.

- Duplicate ***FantasyPackage/Resources/RoomModels/Objects/Armory_Table*** and rename it to 'Armory_Chest'.
- Select 'Armory_Chest' and drag the Prefab named 'A_Chest' into the top field of the Object script.
- The chest will now be generated into Armory styled dungeons and behave the same as the armory table (because we duplicated 'Armory_Table' *we also duplicated it's values).*

We have also added two useful scripts that can be added to the models, or parts of it to enhance the random look and feel of these models. The two scripts are:

1. ***FantasyPackage/Scripts/SecundaryScripts/RandomDestroy*** which can be used to delete certain parts of the object on play.
2. ***FantasyPackage/Scripts/SecundaryScripts/RandomRotate*** which can be used to slightly rotate the object on the spot it is placed. Be aware that unevenly shaped objects could potentially stick out of the reserved spot.

Create a completely new dungeon design

Let's say you want to make a Sci-fi dungeon instead of a fantasy dungeon. This means you will have to change the database completely.

To make this easier, we have provided a Template Dungeon which consists of raw shapes for walls, stairs, objects etc.

- Open the **'Template'** Scene

As we said in the **Customizing the model database** chapter, all models used in the **Resources** folder a recognized by the Dungeon Generator.

- Duplicate the entire **Template** folder**.**
- Rename the duplicated **Template** folder to a name of your choice, for instance: **MySciFiDungeon**.
- Locate the **TemplateResources** folder in the duplicated **Template** folder and rename it to '**Resources'.**
- Select the **DungeonGenerator** object in the scene and go to the DungeonGenerator script in the Inspector.
- Click on the 'Reset' button located in the top of the script to reset the Styles.

The 'Template' Style will now be part of the style lists.

- Remove all other Styles with the 'Remove the style' buttons so only the 'Template' styles will be used.

You can also rename the Resources folder located in the FantasyPackage to FantasyResources so it won't be found by the DungeonGenerator script.

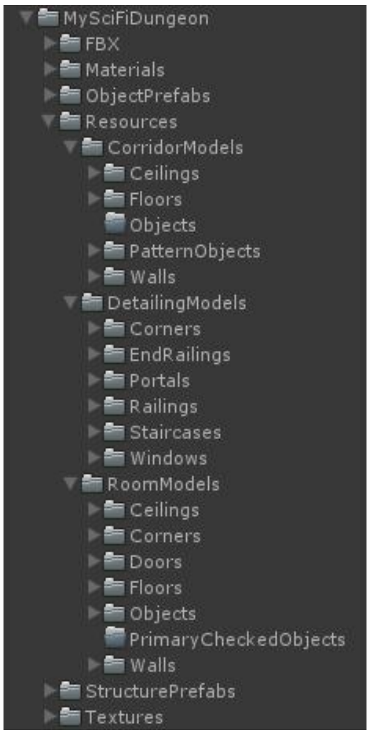
- Generate a dungeon to see the result.

Replacing the template models

Now that we've got a copy of our Template we can start replacing all the objects with our own models.

A dungeon is build up out of 3 elements: **CorridorModels**, **DetailingModels** and **RoomModels**. If you look in the **Resources** folder you will see these 3 folders. Inside them you can find all their sub elements.

For the Dungeon Generator to work you need to have at least 1 model of the same style in EACH folder inside each element. With the exception of Objects, PatternObjects and PrimaryCheckedObjects, these 4 folders (an Objects folder is located in both CorridorModels and RoomModels folder) can be empty.



```
MySciFiDungeon
  FBX
  Materials
  ObjectPrefabs
  Resources
    CorridorModels
      Ceilings
      Floors
      Objects
      PatternObjects
      Walls
    DetailingModels
      Corners
      EndRailings
      Portals
      Railings
      Staircases
      Windows
    RoomModels
      Ceilings
      Corners
      Doors
      Floors
      Objects
      PrimaryCheckedObjects
      Walls
  StructurePrefabs
  Textures
```

By changing the Prefabs in each folder with your own you will completely change the look of the dungeon. Drag the Prefabs into the scene, for reference of shape, size and location and replace it with your own models. Then replace the prefabs with the new ones.

Corridor-Ceilings, -Floors and -Walls as well as Room-Ceilings, -Corners, -Floors and -Walls need to be their exact size and have their exact location to make sure there are no holes in the Dungeon. We suggest you only change their material and maybe add some models to it, but leave the models themselves alone, unless you are an experienced 3D-modeler and can edit the models in a program like Maya or Max.

Adding Objects to your Styles works in the same way as explained in the **Adding your own models** chapter. We provided a few objects of different shapes and sizes to help you on your way. These are located inside the **ObjectPrefabs** folder.

Always keep the parent of the Prefab and replace all the children inside it with your own model. This way the location for placement by the Dungeon Generator will stay intact.

If you make sure that your model is not outside the template shapes you will always be able to walk around the object.

3. Using the generated dungeons

Saves scene

If you generated a very interesting dungeon you can use **File/SaveSceneAs** … to save the dungeon to use as a level in your own game.

Begin and Endpoint

Each dungeon is generated with a Beginpoint and Endpoint. The Beginpoint is currently a bright blue block on which the standard firstperson controller from unity is spawned when starting the dungeon. The Endpoint is a green glowing flame that will start the generation of a new dungeon using the same settings as the dungeon currently being played. This can be useful when you want to have a fully random game with always changing dungeons in a certain style, size or setting.

Removing this endpoint or deleting the attached CreateAnew script will not make it possible to generate new dungeons while playing.

Bake navigation, occlusion culling and light

The dungeons generated by our tool support Occlusion Culling (Pro only), navigation baking (Pro only) and Lightbaking.

Part structure

The generated dungeon will have a certain hierarchy buildup that can be used to quickly find certain locations. Each 5x5x5 part in our dungeon will be stored as a StairCasePart, RoomPart or CorridorPart in the hierarchy. Each cell has a sphere to locate the cells middle point. Each object attached to this cell will have a corresponding “middle point” overlapping the cells middle point. All prefabs we use will have this middle point for easy instantiation in this 5x5x5 grid. Another benefit of these parts can be the creation of a waypoint system for enemies patrolling the dungeon. We have added all floor prefabs with a tag “Floor” so they can be easily found within the different cells.