\providecommand\hyper@newdestlabel[2]

# Dungeon Architect

## Code Respawn

Dec 21, 2019

# TUTORIALS

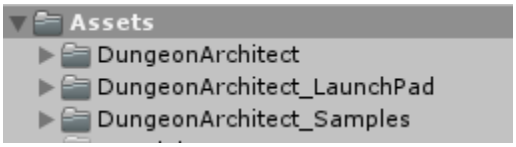
CHAPTER

ONE

# CREATE YOUR FIRST DUNGEON

## 1.1 Install Dungeon Architect

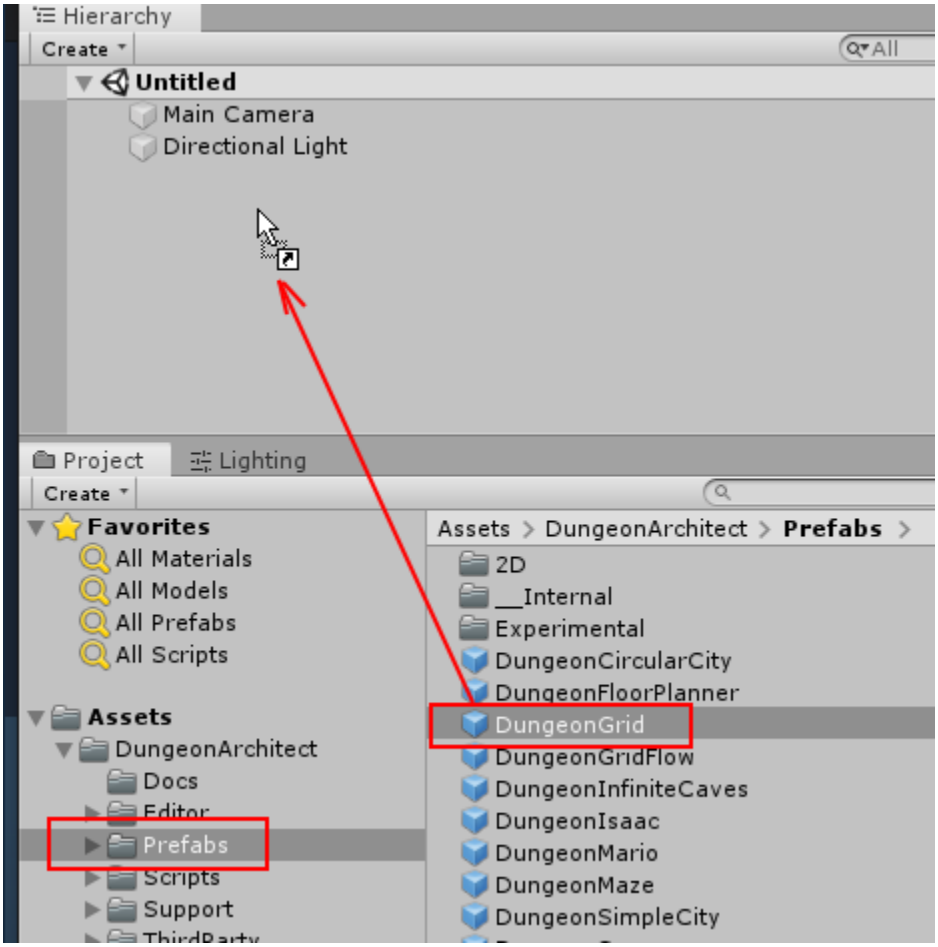
Install and import Dungeon Architect from the Asset Store. You should see the following folders:



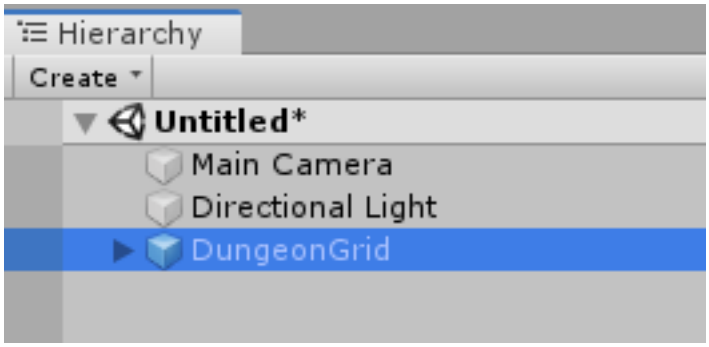

## 1.2 Setup Dungeon Prefab

Create a new Scene

Navigate to `DungeonArchitect > Prefabs` and drop in the `DungeonGrid` prefab on to the scene



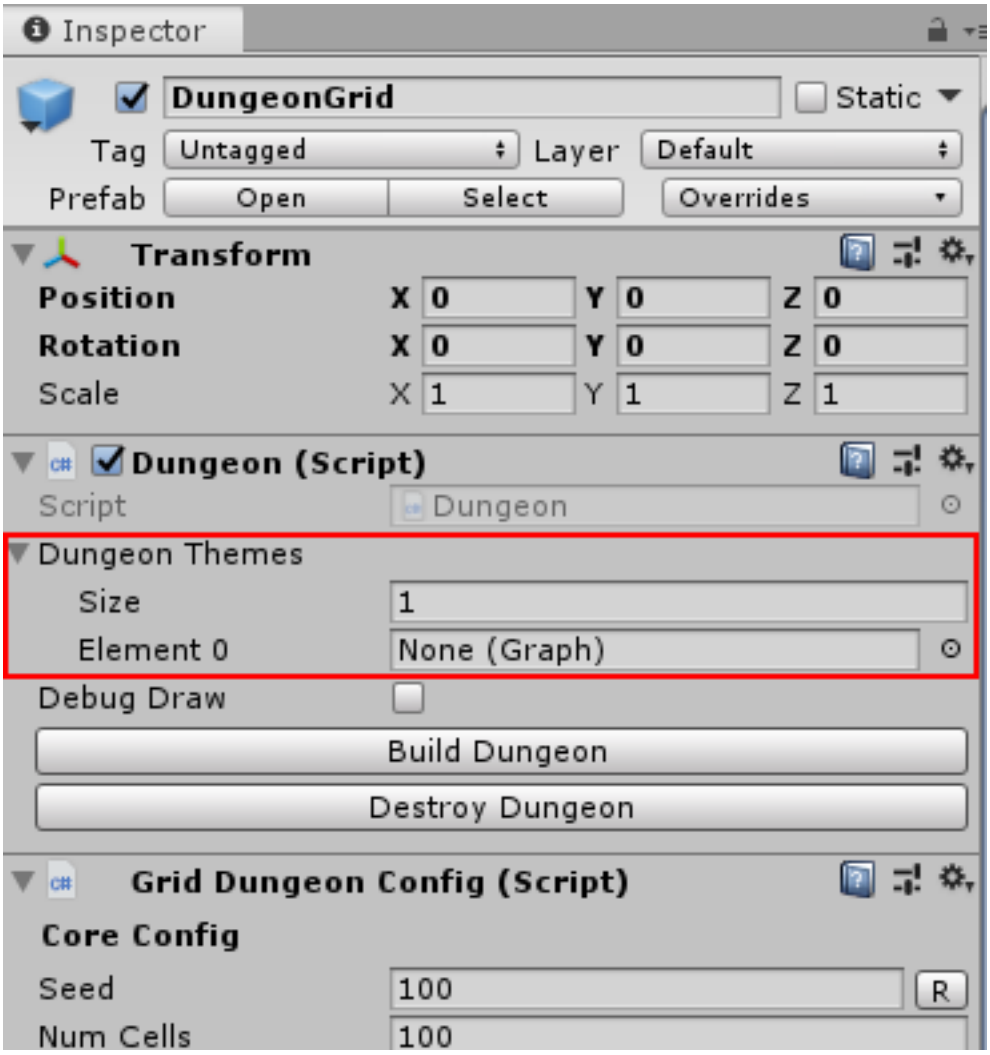

Select the dungeon game object and inspect the properties. We'll need to assign a new theme before we can build the dungeon

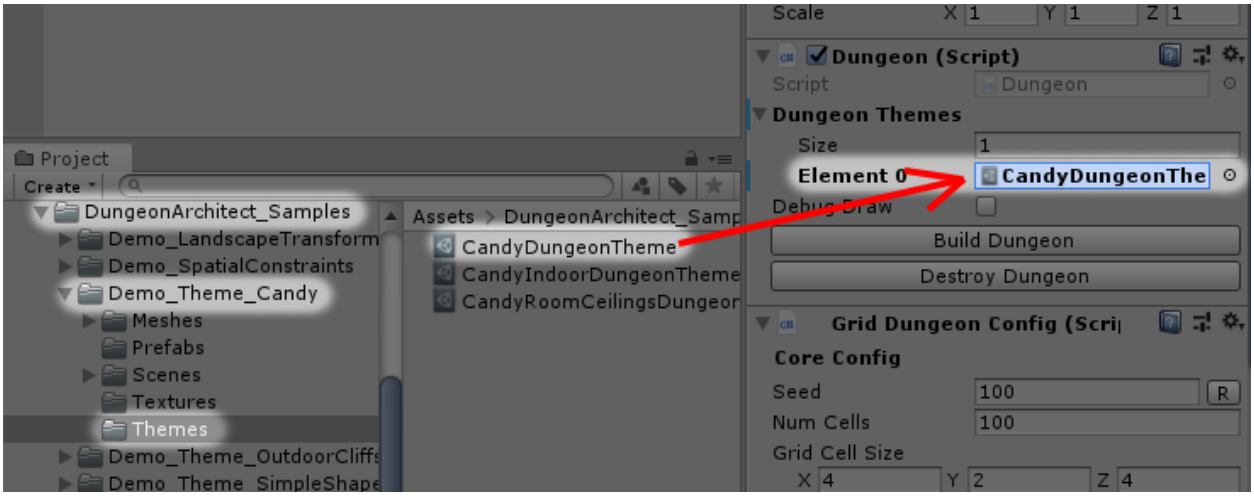



## 1.3 Assign Theme

Assign an existing theme to the dungeon actor

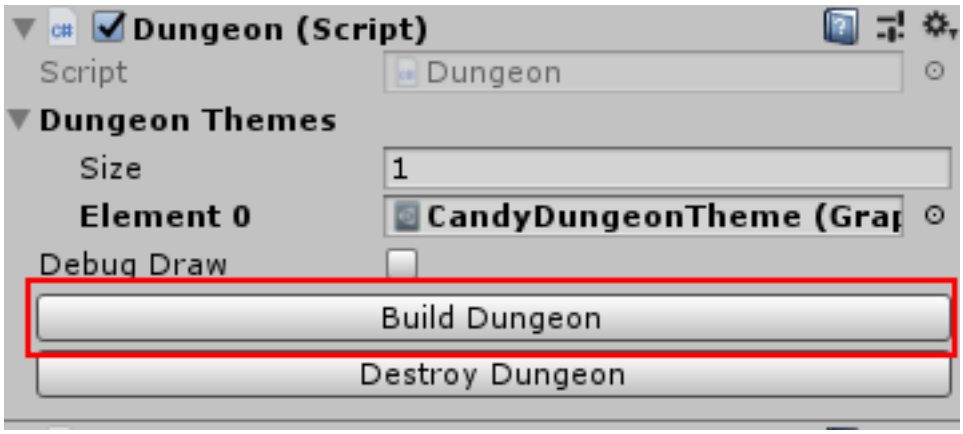
- Navigate to `Assets\DungeonArchitect_Samples\Demo_Theme_Candy\Themes`
- Assign theme file named `CandyDungeonTheme` as shown in the image below

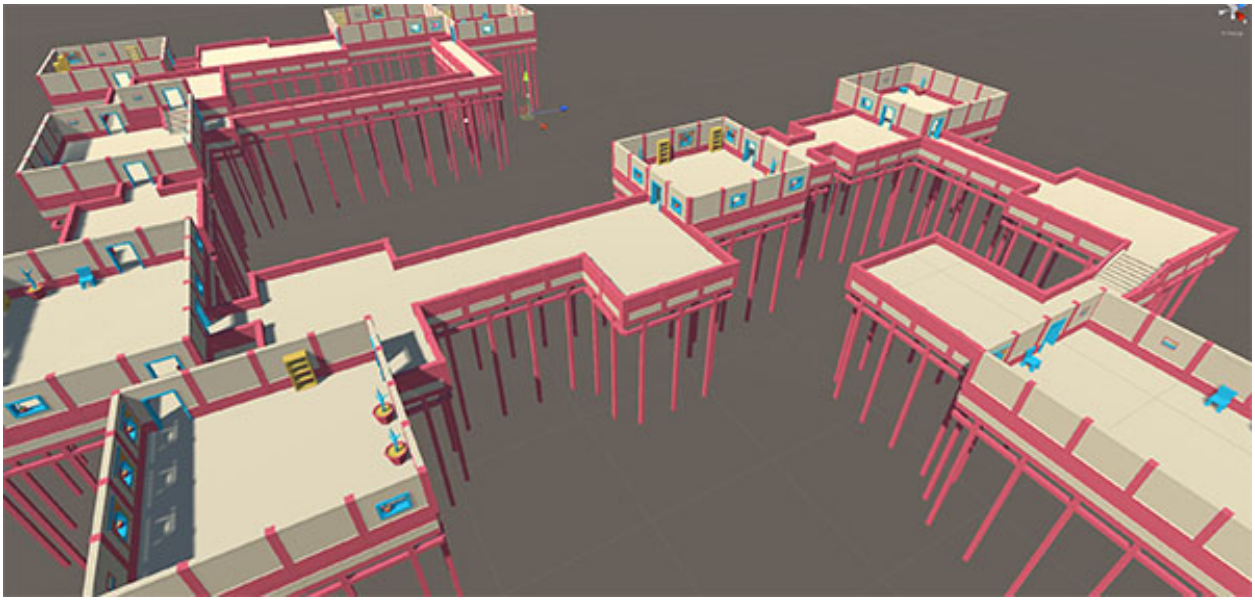

Inspector
DungeonGrid
Static
Tag
Untagged
Layer
Default
Prefab
Open
Select
Overrides
Transform
Position
Rotation
Scale
Dungeon (Script)
Script
Dungeon
Dungeon Themes
Size
1
Element 0
None (Graph)
Debug Draw
Build Dungeon
Destroy Dungeon
Grid Dungeon Config (Script)
Core Config
Seed
100
R
Num Cells
100


Project
Create
DungeonArchitect_Samples
Demo_LandscapeTransform
Demo_SpatialConstraints
Demo_Theme_Candy
Meshes
Prefabs
Scenes
Textures
Themes
Demo_Theme_OutdoorCliffs
Assets > DungeonArchitect_Samp
CandyDungeonTheme
CandyIndoorDungeonTheme
CandyRoomCeilingsDungeon
Scale
Dungeon (Script)
Script
Dungeon
Dungeon Themes
Size
1
Element 0
CandyDungeonThe
Debug Draw
Build Dungeon
Destroy Dungeon
Grid Dungeon Config (Scri
Core Config
Seed
100
R
Num Cells
100
Grid Cell Size
X 4
Y 2
Z 4

## 1.4 Build Dungeon

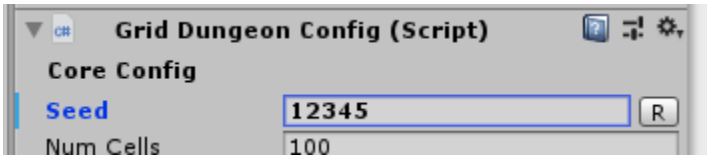
Select the `DungeonGrid` game object and click the `Build Dungeon` button in the Inspector window



## 1.5 Randomize Dungeon

Select the `DungeonGrid` game object and change the Seed value in the configuration. Changing this value will create a dungeon with a different layout




Click `Build Dungeon` button to rebuild the dungeon with the new seed

## 1.6 Organization

All the dungeon objects are created on the root hierarchy and makes it difficult to organize. We'll configure it so all objects are spawned under a certain game object

Lets destroy this current dungeon, configure it for better organization and then rebuild

### 1.6.1 Destroy Existing Dungeon

Search for `dungeon` on the hierarchy search box

Select the DungeonGrid game object and click the `Destroy Dungeon` button

Clear out the search text box in the hierarchy. Your hierarchy should now look like this

### 1.6.2 Configure Parent Object

Create an empty Game Object. All our dungeon items will go inside this parent object

Rename the parent object (e.g. `Dungeon Items`)

Select the parent object and **Reset the transform**

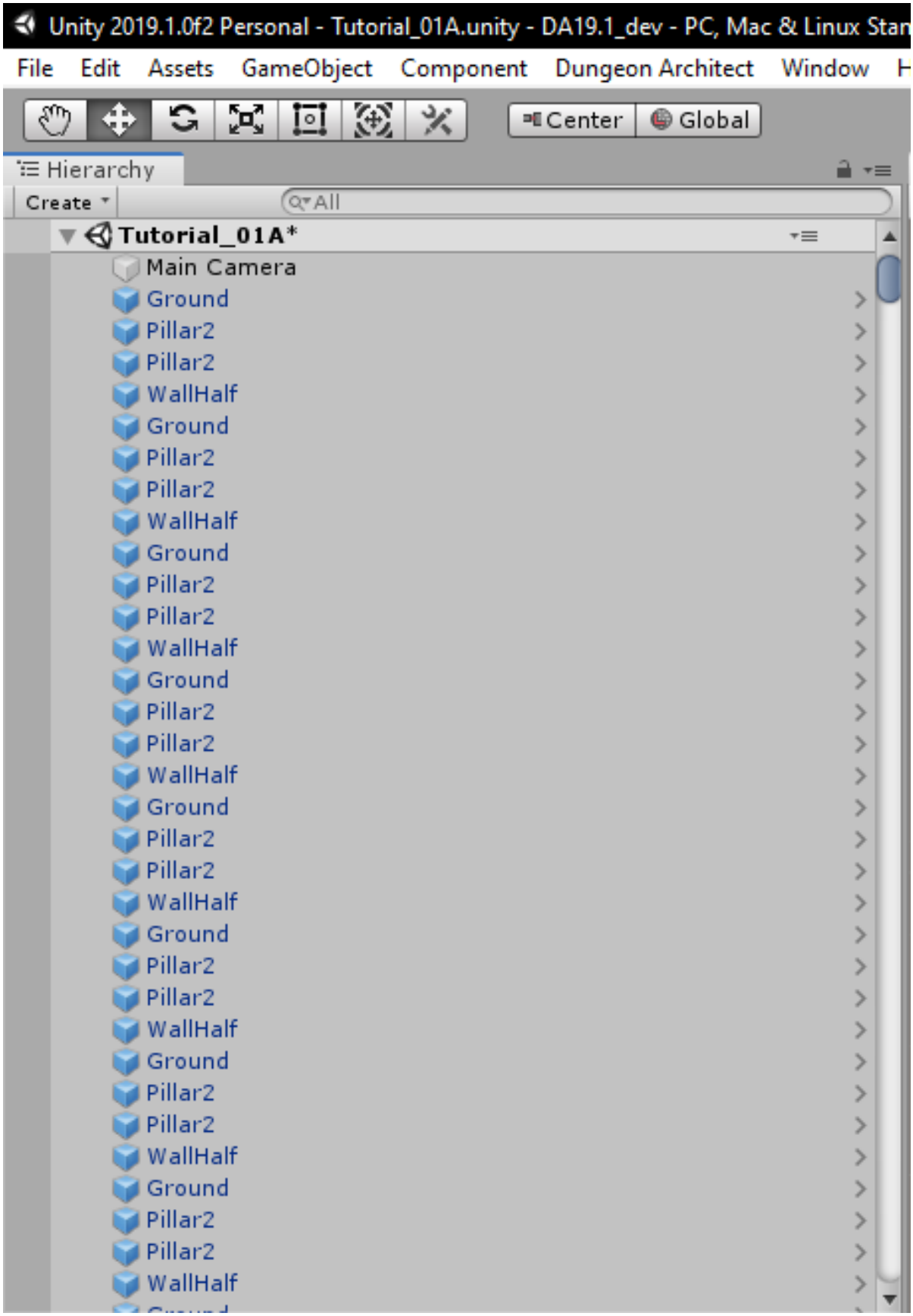
Select the parent object and set it to **static**

Assign the parent object to the GridDungeon game object

### 1.6.3 Rebuild Dungeon

Select the GridDungeon game object and click `Build Dungeon`

All your dungeon game objects will be organized under the parent object


Unity 2019.1.0f2 Personal - Tutorial_01A.unity - DA19.1_dev - PC, Mac & Linux Stan
File
Edit
Assets
GameObject
Component
Dungeon Architect
Window
Center
Global
Hierarchy
Create
All
Tutorial_01A*
Main Camera
Ground
Pillar2
Pillar2
WallHalf
Ground
Pillar2
Pillar2
WallHalf
Ground
Pillar2
Pillar2
WallHalf
Ground
Pillar2
Pillar2
WallHalf
Ground
Pillar2
Pillar2
WallHalf
Ground
Pillar2
Pillar2
WallHalf
Ground
Pillar2
Pillar2
WallHalf

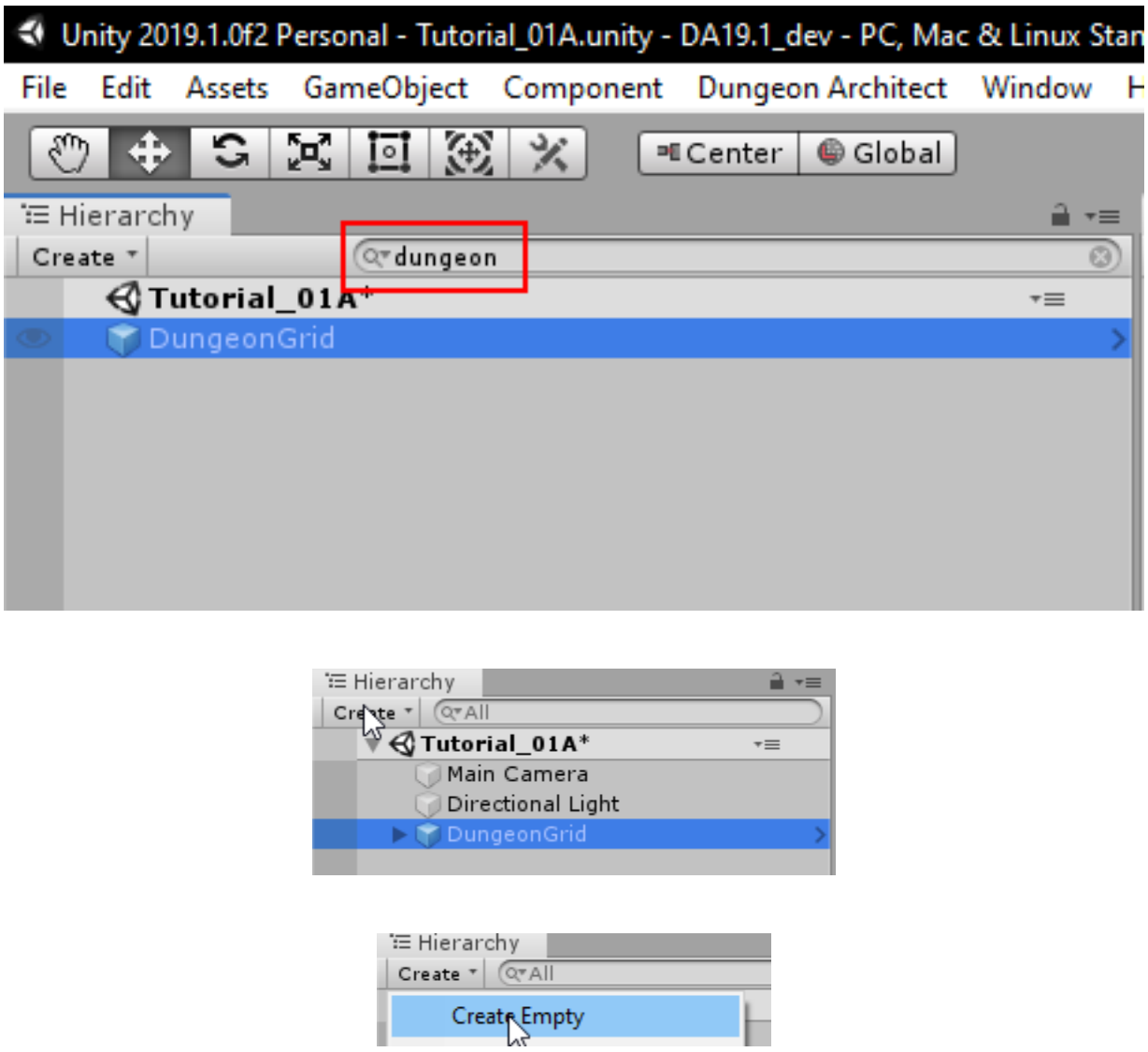
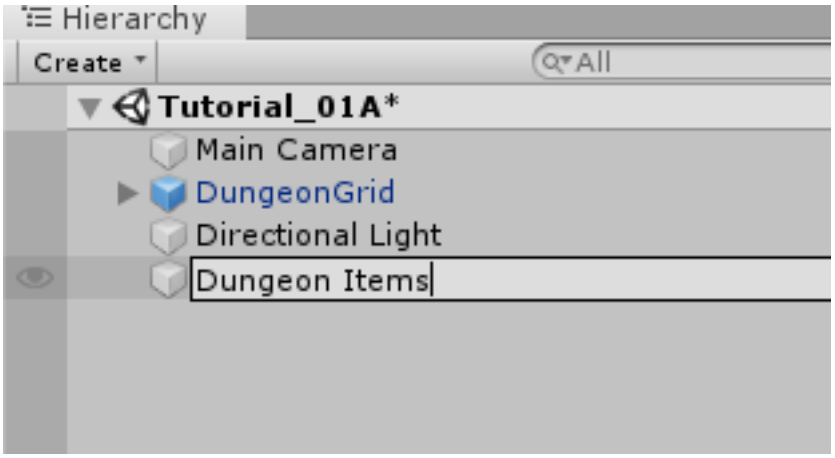
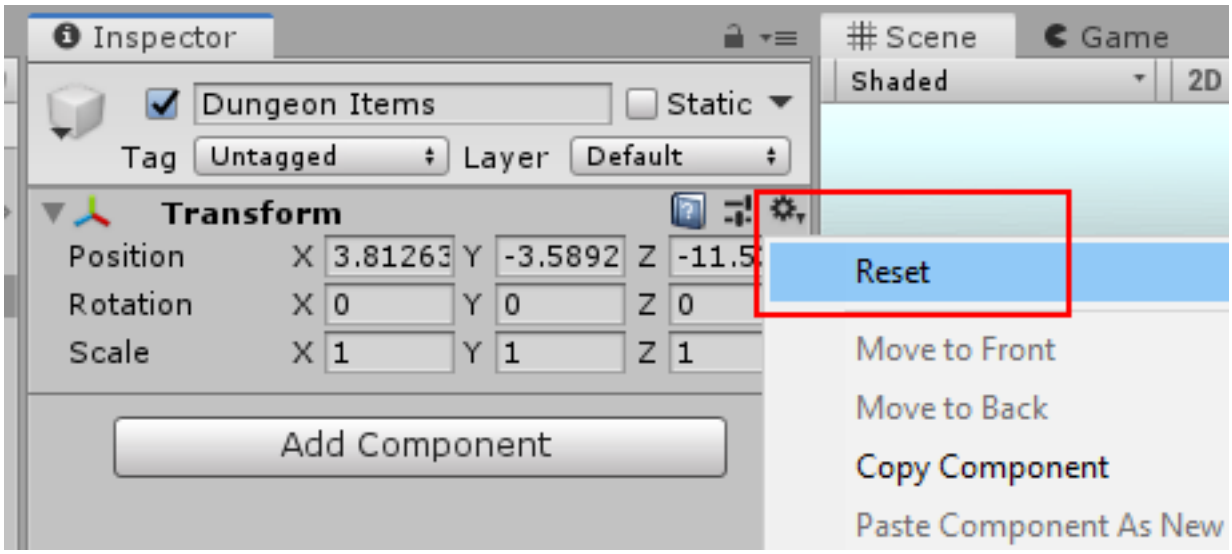
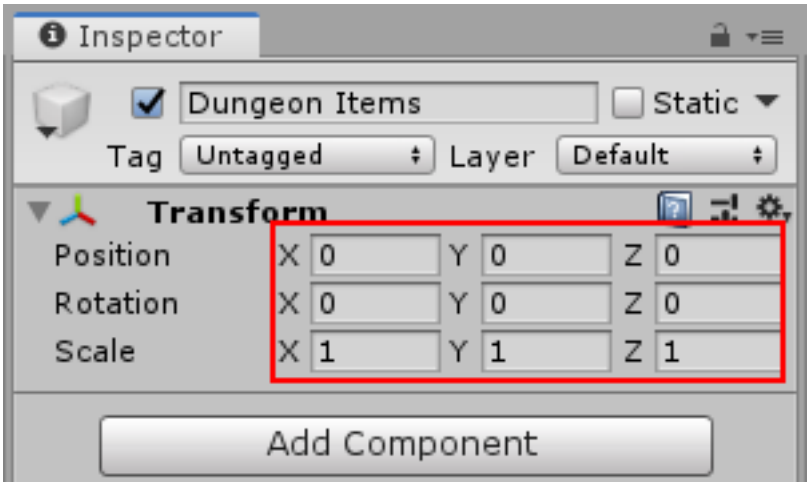
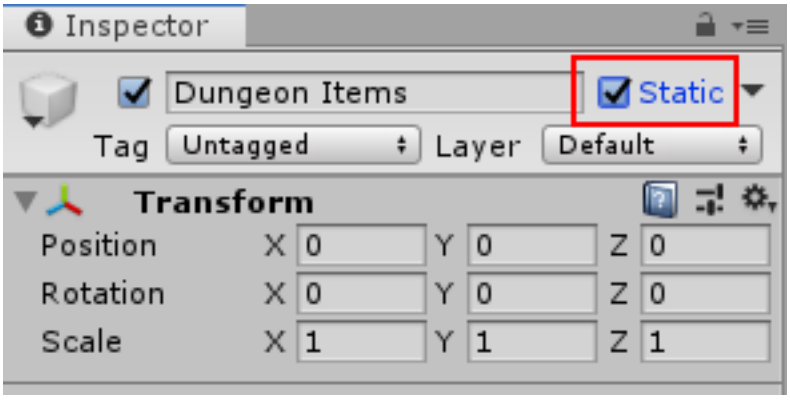


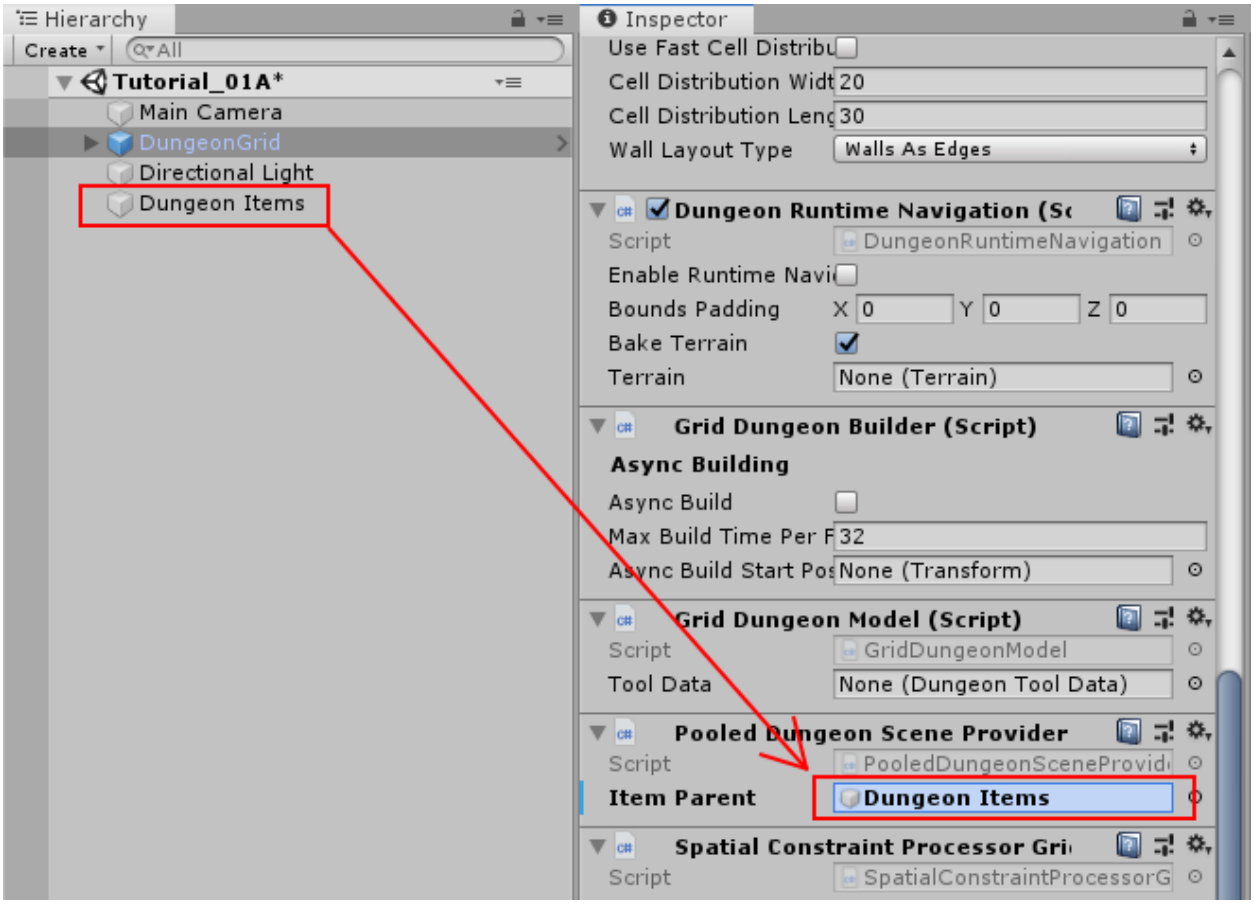

Fig. 1: Create an empty game object

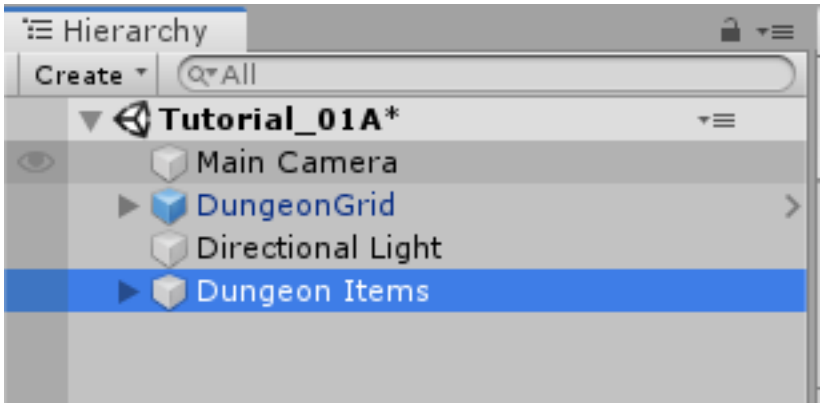



Inspector
Scene
Game
Shaded
2D
Dungeon Items
Static
Tag
Untagged
Layer
Default
Transform
Position
X 3.81263
Y -3.5892
Z -11.5
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Add Component
Reset
Move to Front
Move to Back
Copy Component
Paste Component As New


Inspector
Dungeon Items
Static
Tag
Untagged
Layer
Default
Transform
Position
X 0
Y 0
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Add Component


Inspector
Dungeon Items
Static
Tag
Untagged
Layer
Default
Transform
Position
X 0
Y 0
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1


Hierarchy
Create
All
Tutorial_01A*
Main Camera
DungeonGrid
Directional Light
Dungeon Items
Inspector
Use Fast Cell Distribu
Cell Distribution Widt 20
Cell Distribution Leng 30
Wall Layout Type
Walls As Edges
Dungeon Runtime Navigation (Sc
Script
DungeonRuntimeNavigation
Enable Runtime Navi
Bounds Padding
X 0
Y 0
Z 0
Bake Terrain
Terrain
None (Terrain)
Grid Dungeon Builder (Script)
Async Building
Async Build
Max Build Time Per F 32
Async Build Start Pos None (Transform)
Grid Dungeon Model (Script)
Script
GridDungeonModel
Tool Data
None (Dungeon Tool Data)
Pooled Dungeon Scene Provider
Script
PooledDungeonSceneProvid
Item Parent
Dungeon Items
Spatial Constraint Processor Gri
Script
SpatialConstraintProcessorG

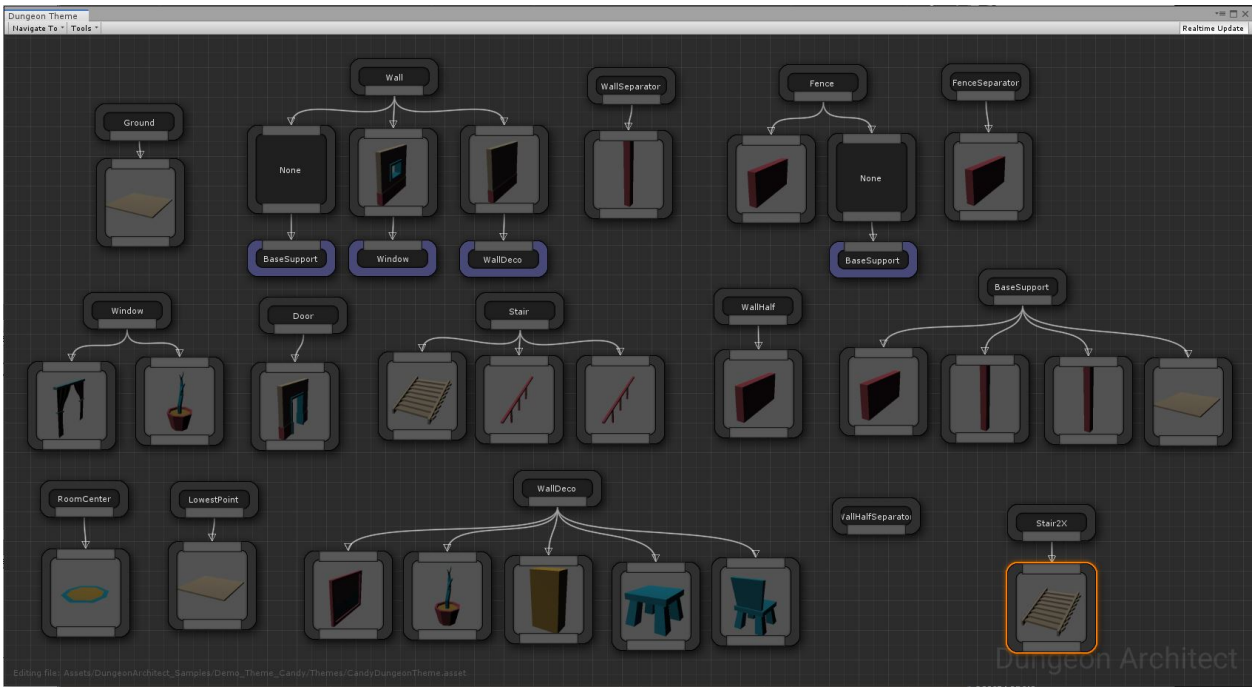

Hierarchy
Create
All
Tutorial_01A*
Main Camera
DungeonGrid
Directional Light
Dungeon Items

CHAPTER

TWO

# DESIGN YOUR FIRST THEME

A theme file is a mapping between **marker** names (like Walls, Ground, Door etc) and the meshes that you provide. The prefabs you map here will be used to build your dungeon




In the previous section, we created a dungeon with an existing theme file (CandyDungeonTheme). In this section, we'll create one ourselves

## 2.1 Create a Theme File

Create a new theme file from either the Main menu or the Create menu

This will create a theme file in the current open directory in the Projects window

## 2.2 Open the Theme File

Double-click the theme file to open the **Theme Editor**. Dock the theme editor so you can see both the Scene view and the Theme Editor at the same time

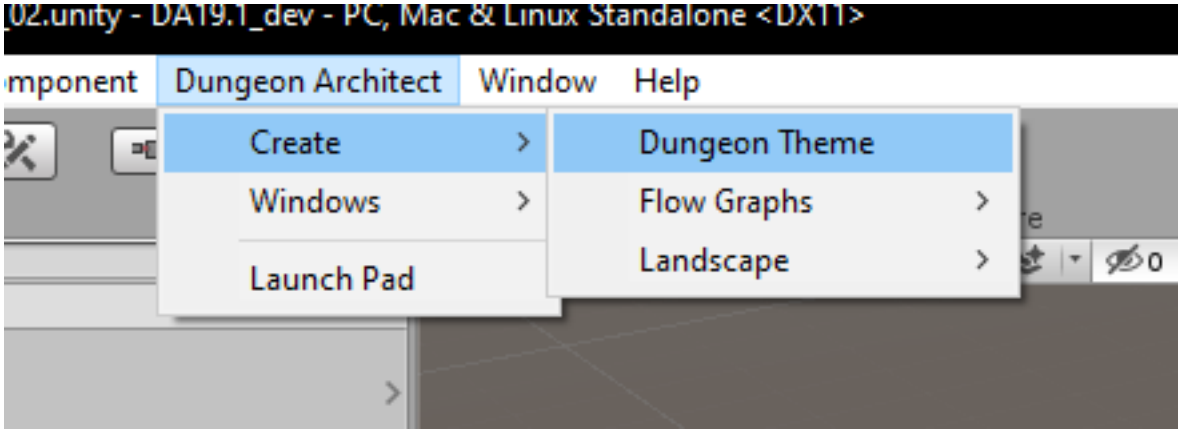



Fig. 1: Create from Main menu

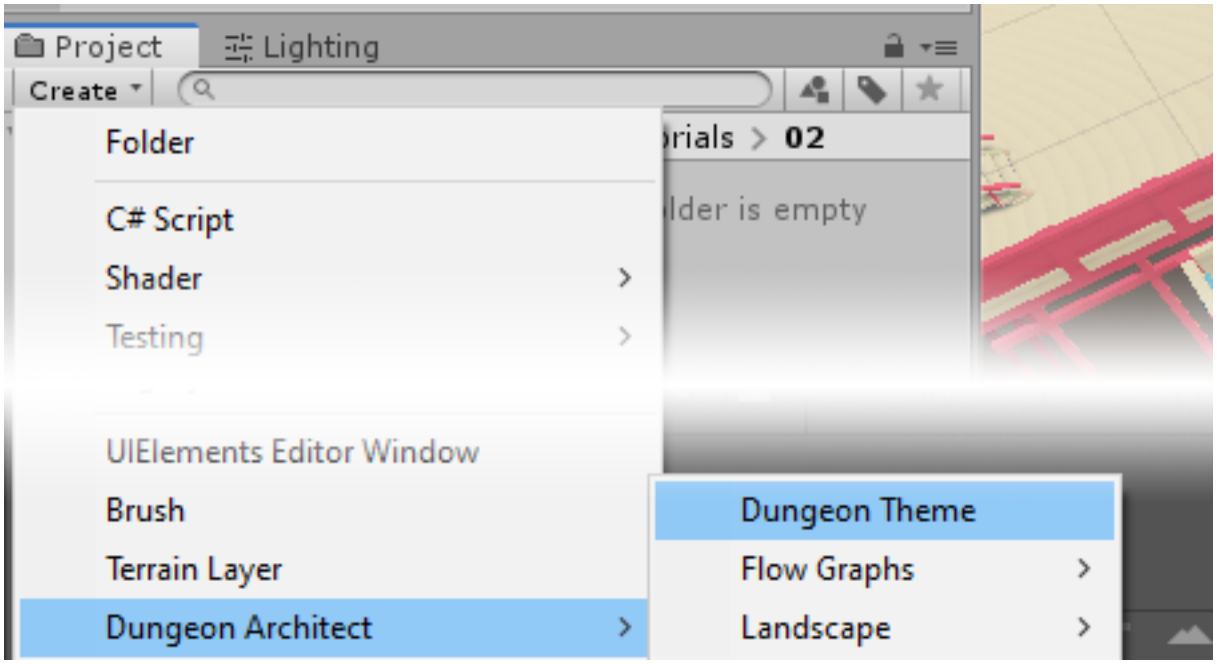
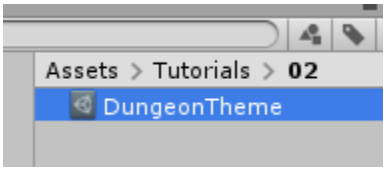
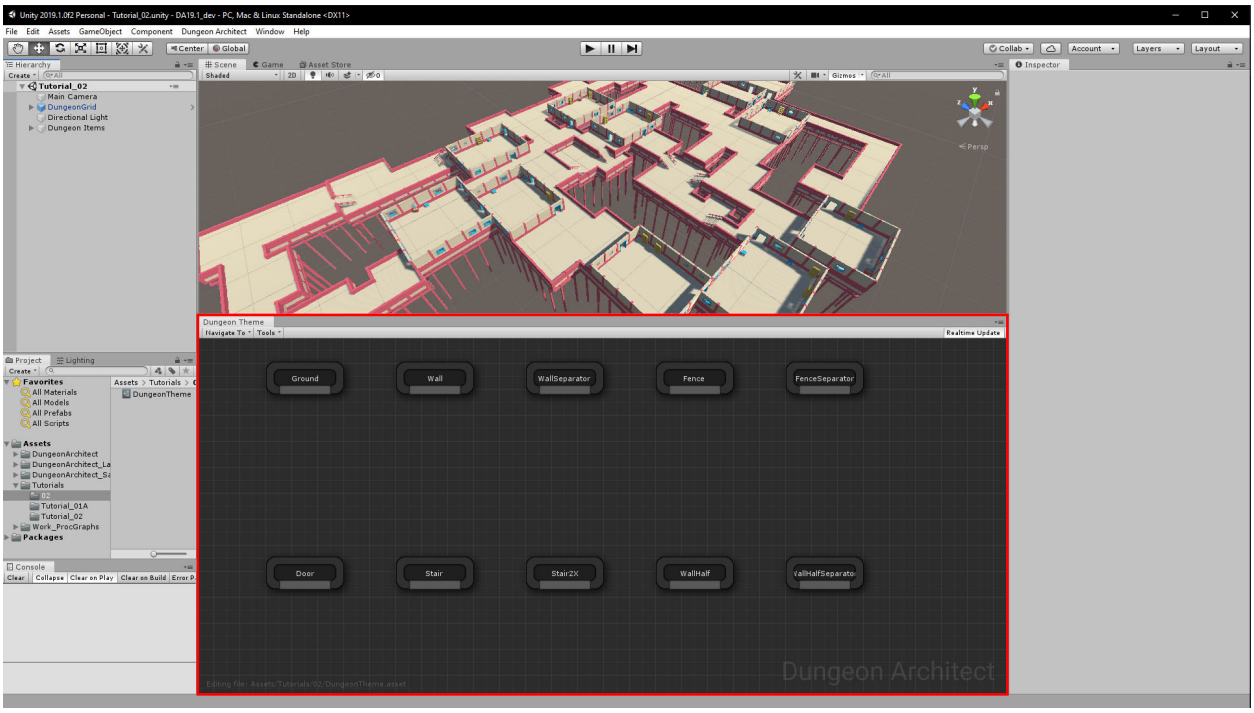



Fig. 2: Create from Create menu






## 2.3 Assign the Theme File

Destroy the existing dungeon by selecting `DungeonGrid` game object and click `Destroy Dungeon` button

Assign the theme file that you just created, on to the DungeonGrid game object

Click `Build Dungeon` and nothing should appear. That's because we haven't added prefab mappings on the theme yet

## 2.4 Design the Theme

Navigate to `Assets\DungeonArchitect_Samples\Demo_Theme_Candy\Prefabs`. This folder contains a set of mesh prefabs we can use for our dungeon

### 2.4.1 Add Ground

Drag-drop the ground prefab mesh on to the Theme Editor

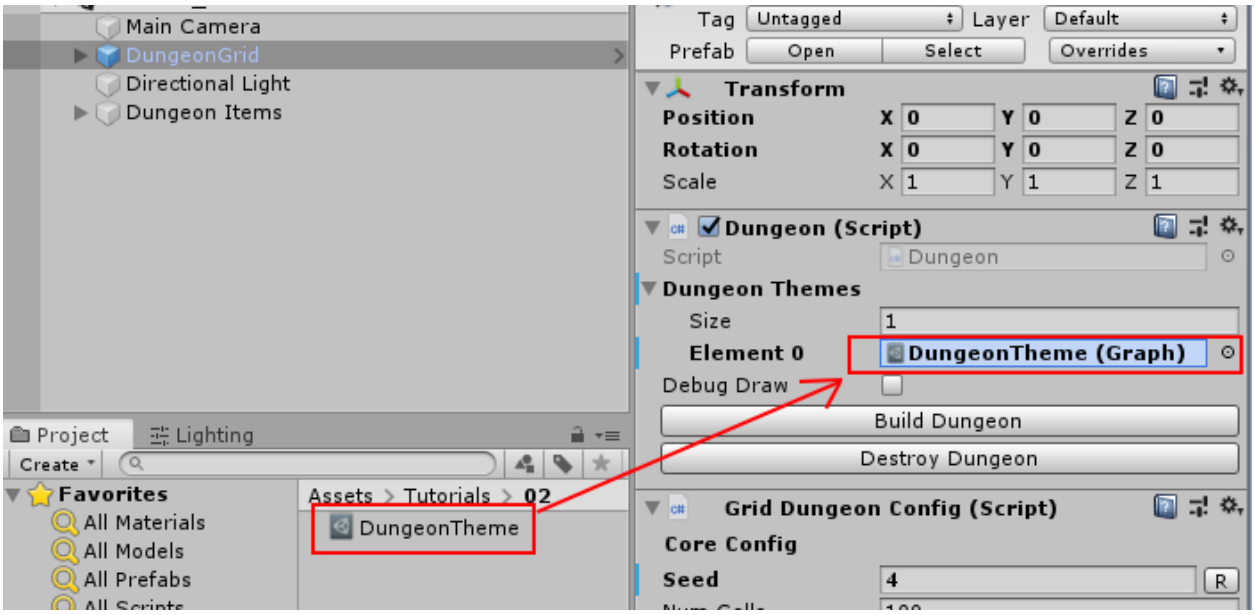
Link up the mesh node with the `Ground` marker node. When you do, you should see a live preview on the scene view with the dungeon using this ground mesh

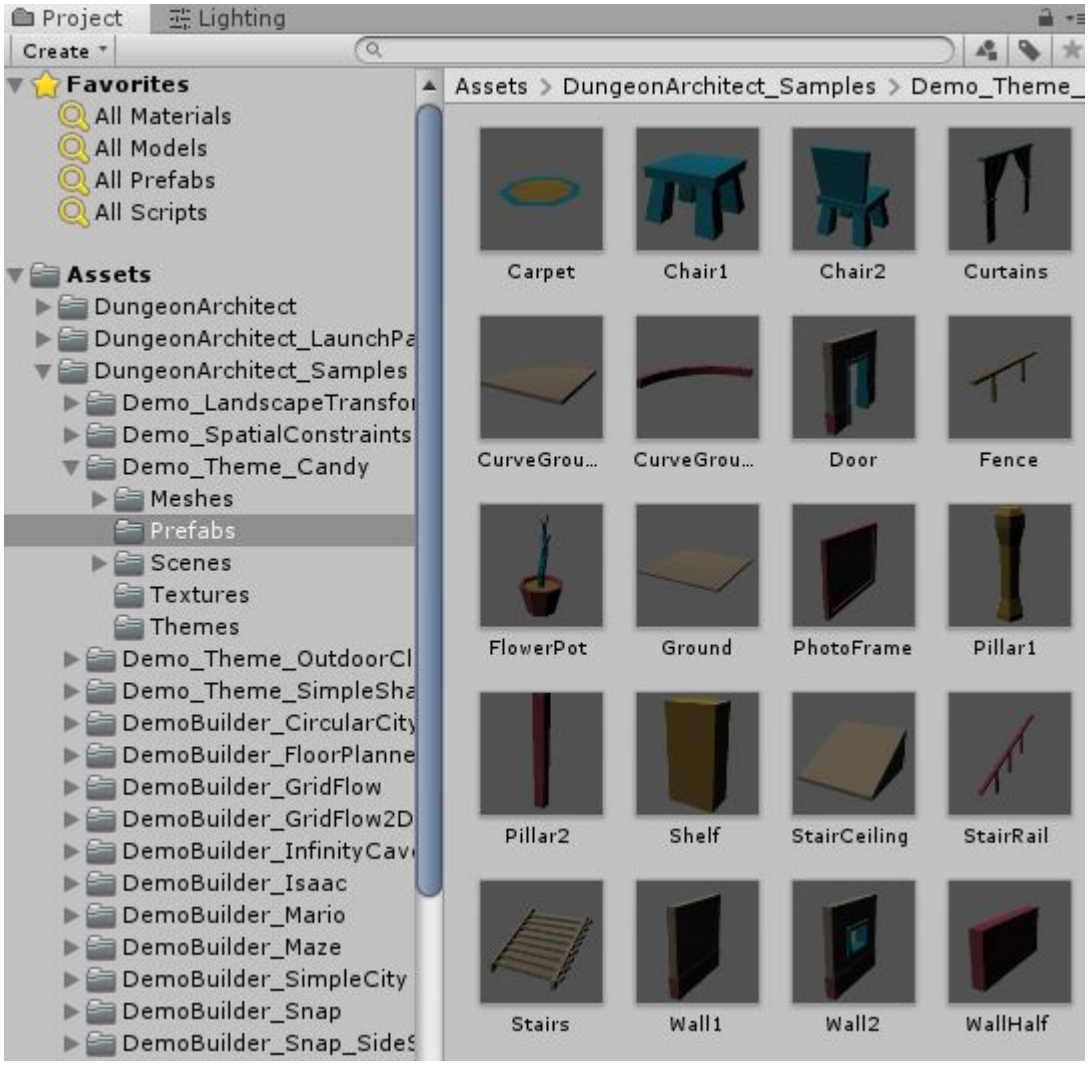
For the live preview to work, make sure the "Realtime Update" button is enabled in the Theme Editor

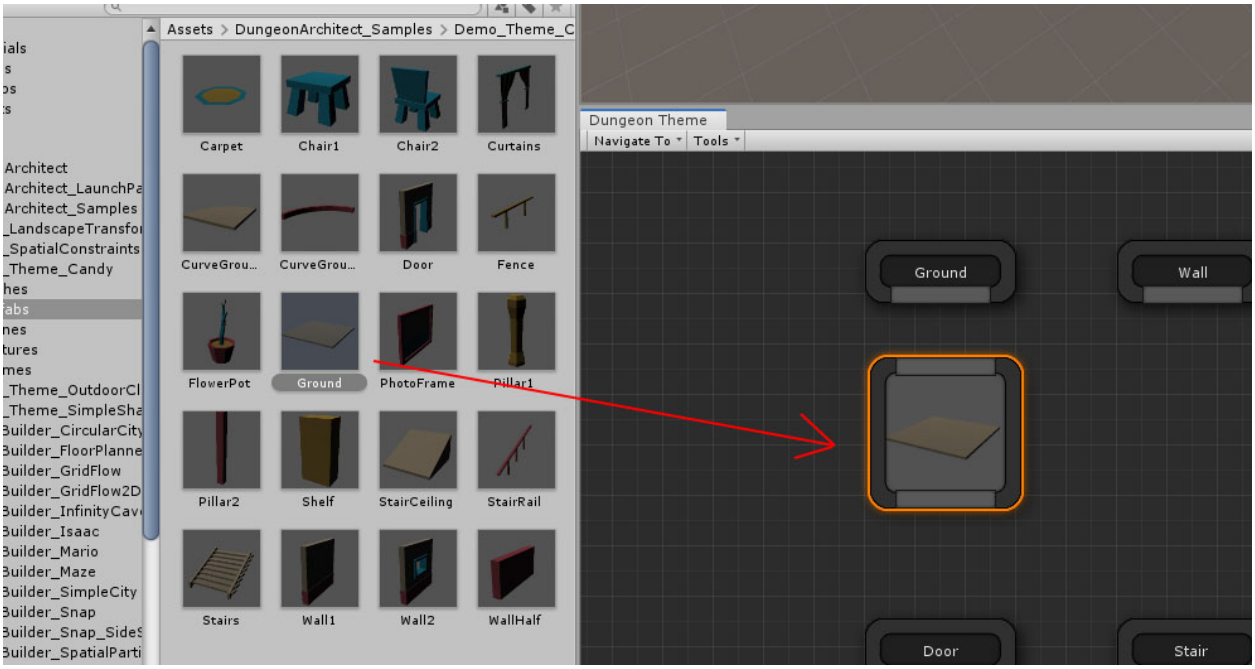
Another criteria for the live preview to work is that a dungeon in the scene should reference the theme that is currently being edited in the theme editor

### 2.4.2 Add More Prefabs

Go ahead and add prefabs under the following markers: **Wall**, **Fence** and **Door**


Main Camera
DungeonGrid
Directional Light
Dungeon Items
Tag
Untagged
Layer
Default
Prefab
Open
Select
Overrides
Transform
Position
Rotation
Scale
Dungeon (Script)
Script
Dungeon
Dungeon Themes
Size
1
Element 0
DungeonTheme (Graph)
Debug Draw
Build Dungeon
Destroy Dungeon
Grid Dungeon Config (Script)
Core Config
Seed
4
R
Project
Lighting
Create
Favorites
All Materials
All Models
All Prefabs
Assets > Tutorials > 02
DungeonTheme


Project
Lighting
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
DungeonArchitect
DungeonArchitect_Samples
Demo_LandscapeTransfo
Demo_SpatialConstraints
Demo_Theme_Candy
Meshes
Prefabs
Scenes
Textures
Themes
Demo_Theme_SimpleSha
DemoBuilder_CircularCity
DemoBuilder_FloorPlanne
DemoBuilder_GridFlow
DemoBuilder_GridFlow2D
DemoBuilder_Isaac
DemoBuilder_Mario
DemoBuilder_Maze
DemoBuilder_SimpleCity
DemoBuilder_Snap
Assets > DungeonArchitect_Samples > Demo_Theme_
Carpet
Chair1
Chair2
Curtains
CurveGrou...
CurveGrou...
Door
Fence
FlowerPot
Ground
PhotoFrame
Pillar1
Pillar2
Shelf
StairCeiling
StairRail
Stairs
Wall1
Wall2
WallHalf




### 2.4.3 Add Wall Pillars

Drag drop the `Pillar2` prefab to the theme editor and link it to the `WallSeparator` marker node

We'll need to make this pillar a bit bigger. Select the node you just dropped and modify the Scale parameter under Offset category

### 2.4.4 Add Windows

We have two wall meshes in the samples folder

The other one (`Wall2`) has a window. Lets configure the theme to sometimes use this second mesh so we have windows

Drag drop `Wall2` prefab on to the theme editor and place it **before** (left of) the existing wall prefab node

When you connect this to the `Wall` marker node, you'll notice it has picked up the window node for all the walls

This is because Dungeon Architect starts executing the nodes from left to right. When the condition was satisfied to pick the first node, it stopped execution and never came to the second node.

There are multiple ways you can control this condition, the simplest being adjusting the probablity of selection.

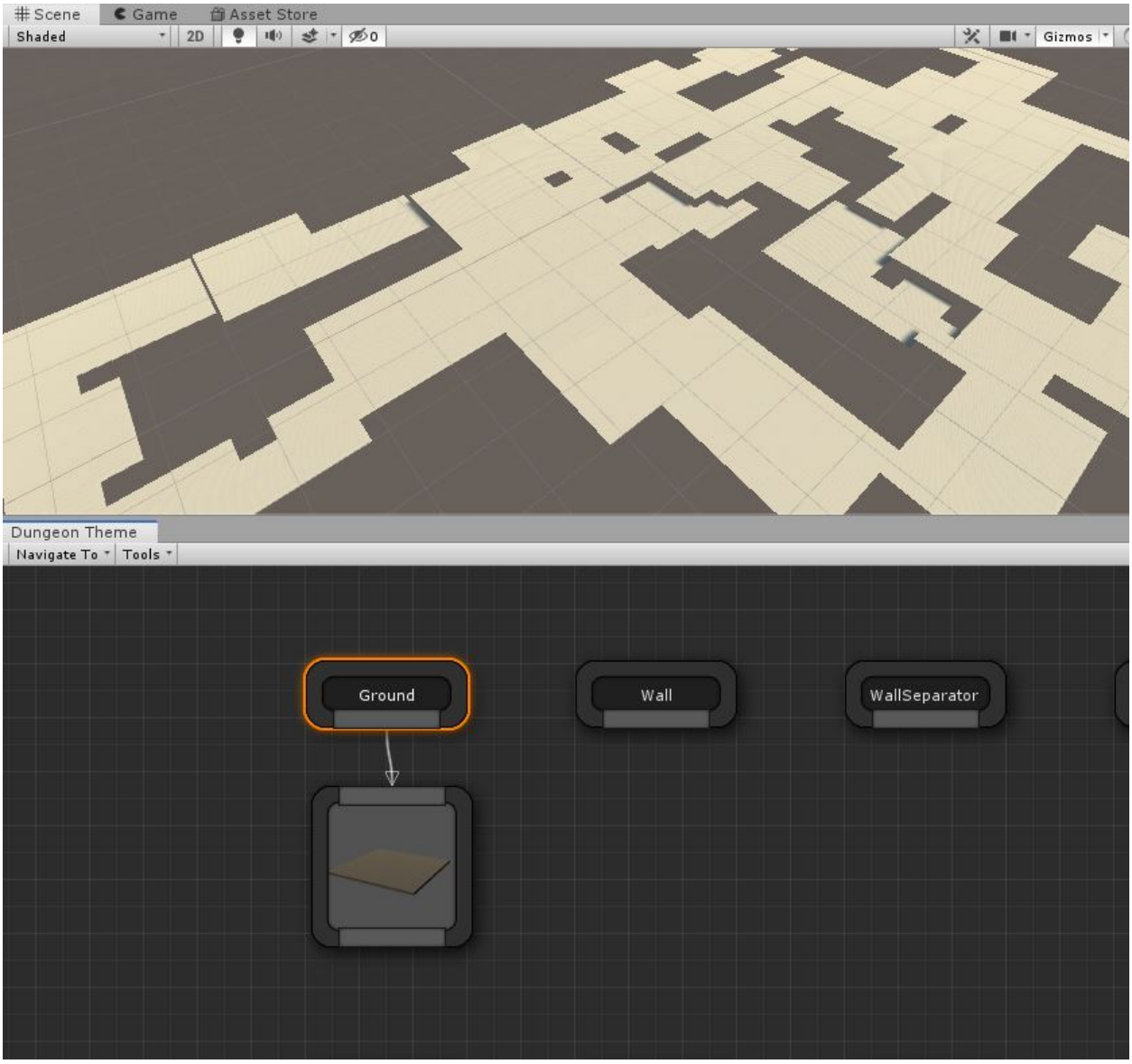
Select the node you just dropped and change the probability to `0.5` (this would mean it gets selected 50% of the time). The other 50% of the time, it would not be selected and the execution would then move to the next node, and hence selecting the non-windowed wall node

### 2.4.5 Add Wall Decorations

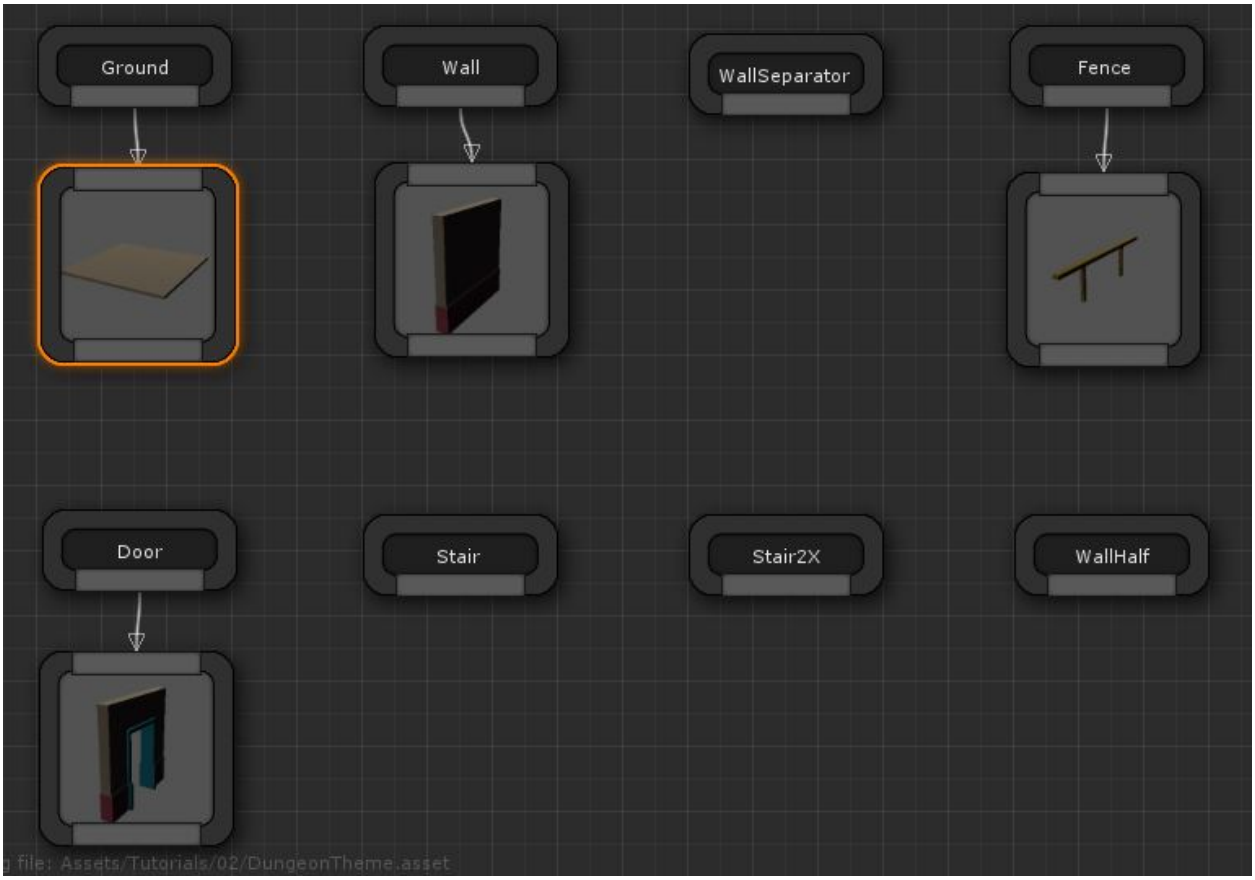
There's a photo frame prefab we'd like to attach to every wall

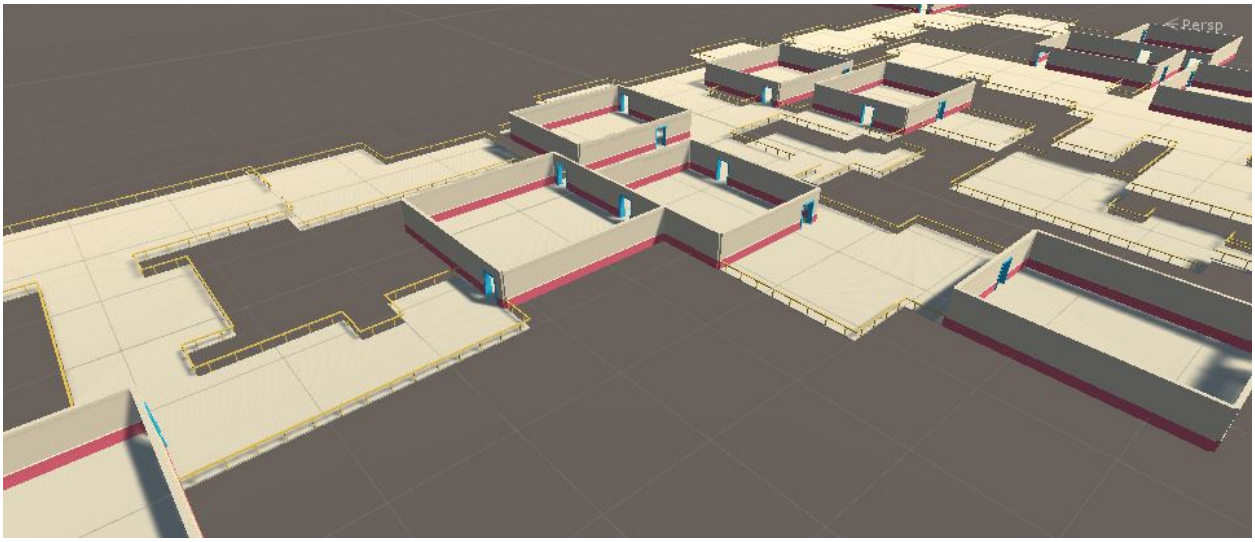
Drag drop this prefab to the theme editor **before** the two existing nodes and link it to the `Wall` marker node

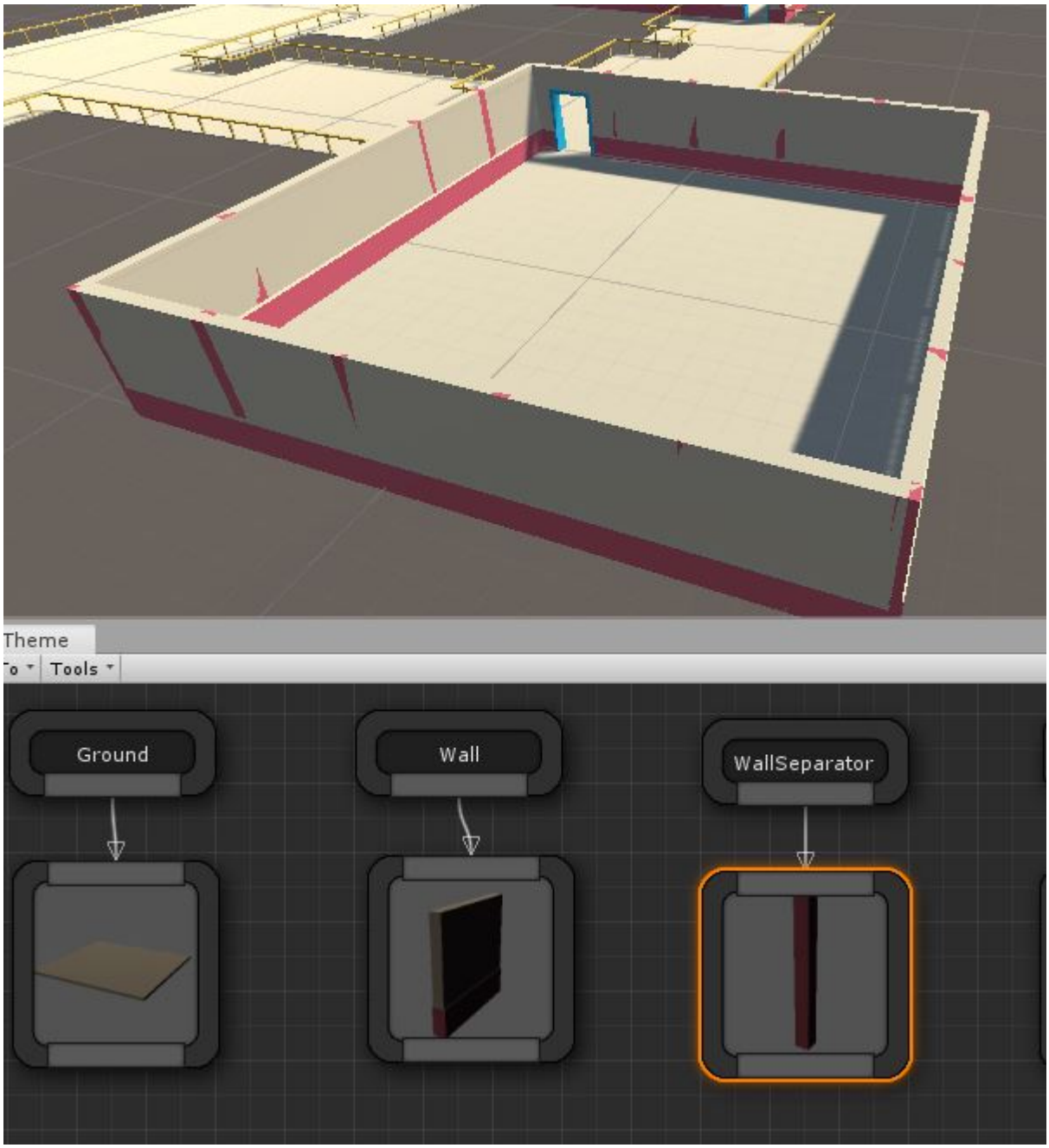
This will cause all the walls to disappear and be replaced with this photo frame

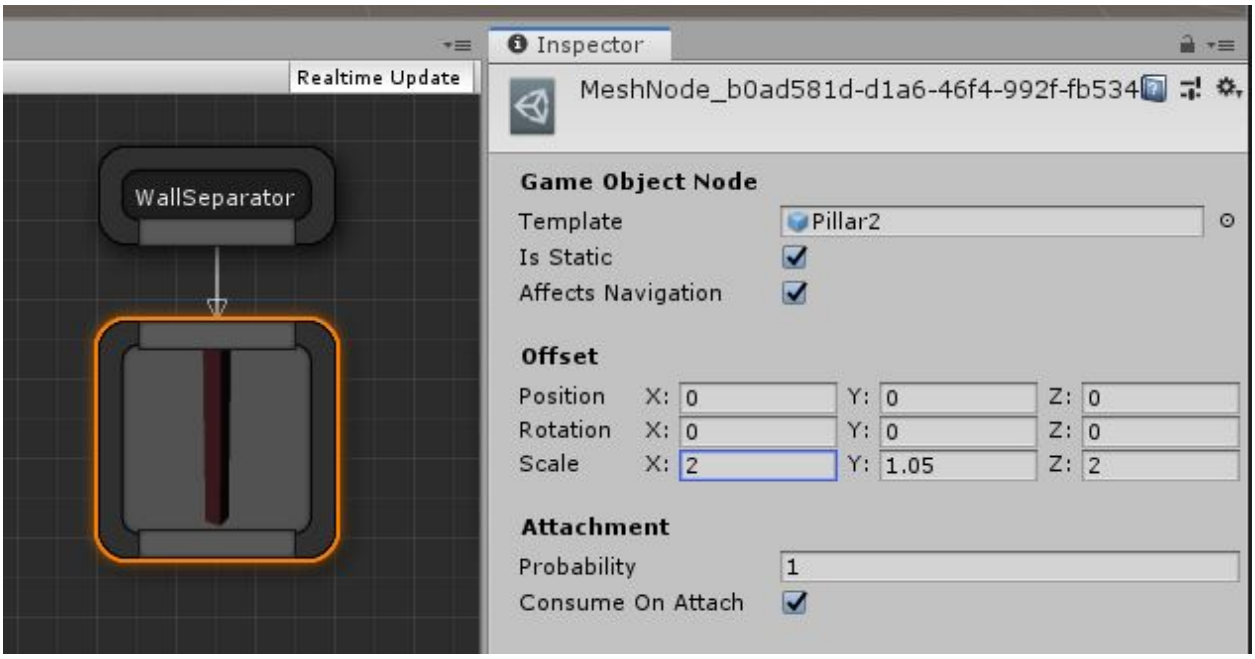

Scene
Game
Asset Store
Shaded
2D
Gizmos
Dungeon Theme
Navigate To
Tools
Ground
Wall
WallSeparator

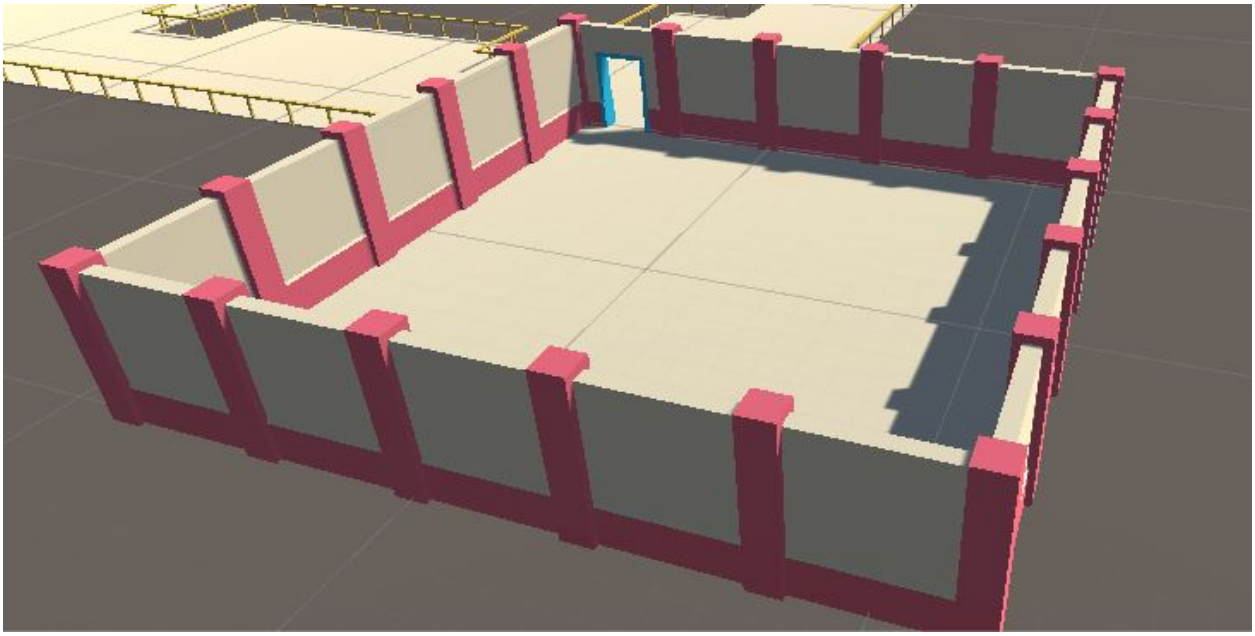

Dungeon Theme
Navigate To
Tools
Realtime Update

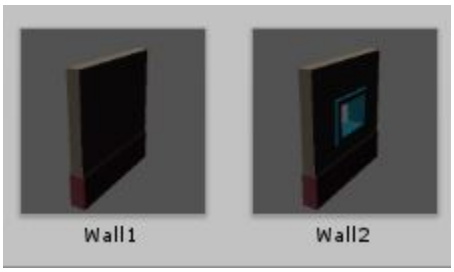

Ground
Wall
WallSeparator
Fence
Door
Stair
Stair2X
WallHalf
file: Assets/Tutorials/02/DungeonTheme.asset

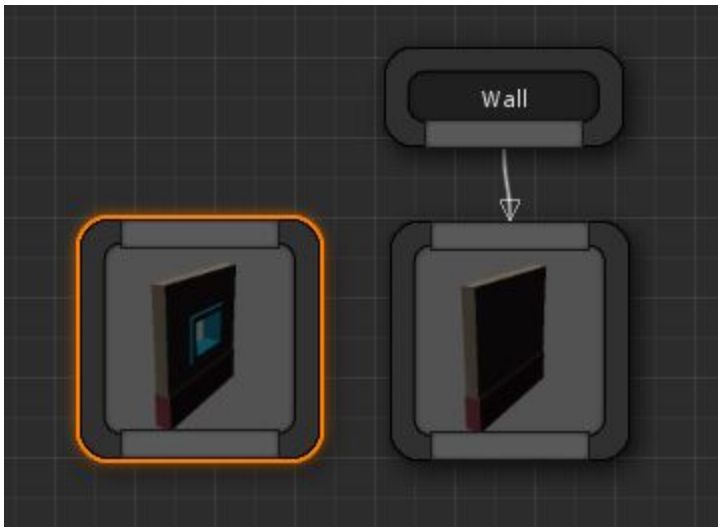

Persp


Theme
Tools
Ground
Wall
WallSeparator


Inspector
Realtime Update
MeshNode_b0ad581d-d1a6-46f4-992f-fb534
WallSeparator
Game Object Node
Template
Pillar2
Is Static
Affects Navigation
Offset
Position X: 0 Y: 0 Z: 0
Rotation X: 0 Y: 0 Z: 0
Scale X: 2 Y: 1.05 Z: 2
Attachment
Probability
1
Consume On Attach




Wall1
Wall2




This is because once the photo frame was selected, the execution stopped and the the wall nodes further down the line were not executed.

Selected the photo frame and uncheck the flag "Consume on Attach". This will cause the execution to continue further even though this node was selected by the theming engine

Lets adjust the offset of the photo frame (position and rotation) to make it properly align with the inner walls

Select the photo frame node and change the position to `(0, 2, -0.22)` and rotation to `(0, 180, 0)`

The photo frame is aligned with the walls

### 2.4.6 Marker Emitters

We have an issue with the photo frames. They also spawn near windows

`Marker Emitters` allow you to emit marker names from any of your dropped prefab nodes. This means, we can define a new marker node (e.g. `MyWallDeco`) and then emit that marker from the wall node that doesn't have a window (`Wall1` prefab). All our wall decorations can now go under this `MyWallDeco` marker and it will show up only near solid walls

Right click on an empty area in the theme editor and select `Add Marker Node`

Select the node and change its name to `MyWallDeco`

Break the link to the photo frame

Connect this under `MyWallDeco` marker node. All the future wall decorations can also go under this marker

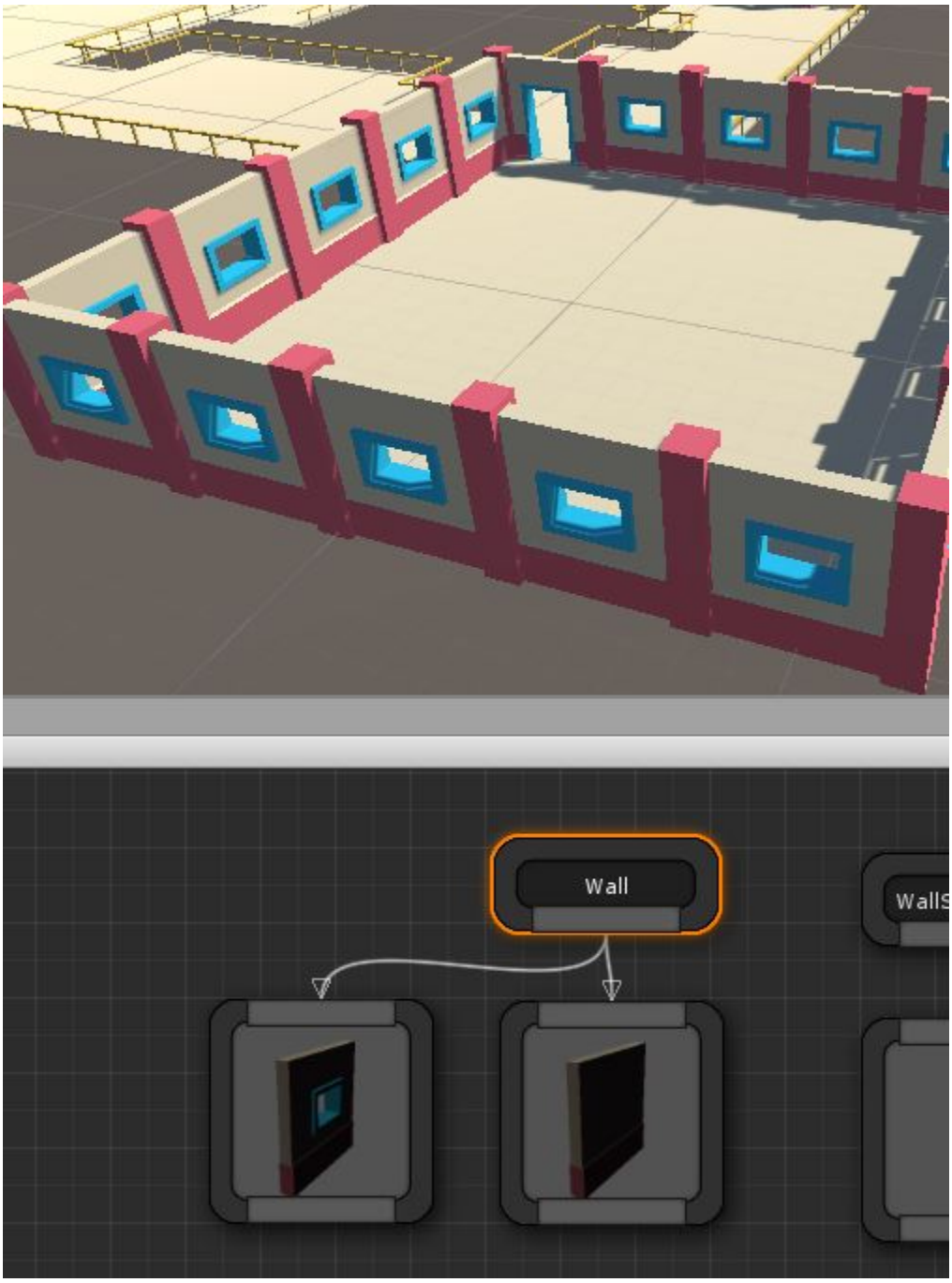
Now emit this marker from the wall node that doesn't contain a window

Drag a link out of the bottom of the solid wall prefab node and release the mouse in an empty area

You can follow the same method to create another type of decoration (e.g. MyWindowDeco) and emit it from under the windowed wall node. In this example, I've added a flower pot in the windows

### 2.4.7 Align with Offset

Dungeon Architect can adapt to any modular asset regardless of the prefab pivot position. If the pivots are off, you can always adjust them from the Offset section of the node's properties


Wall

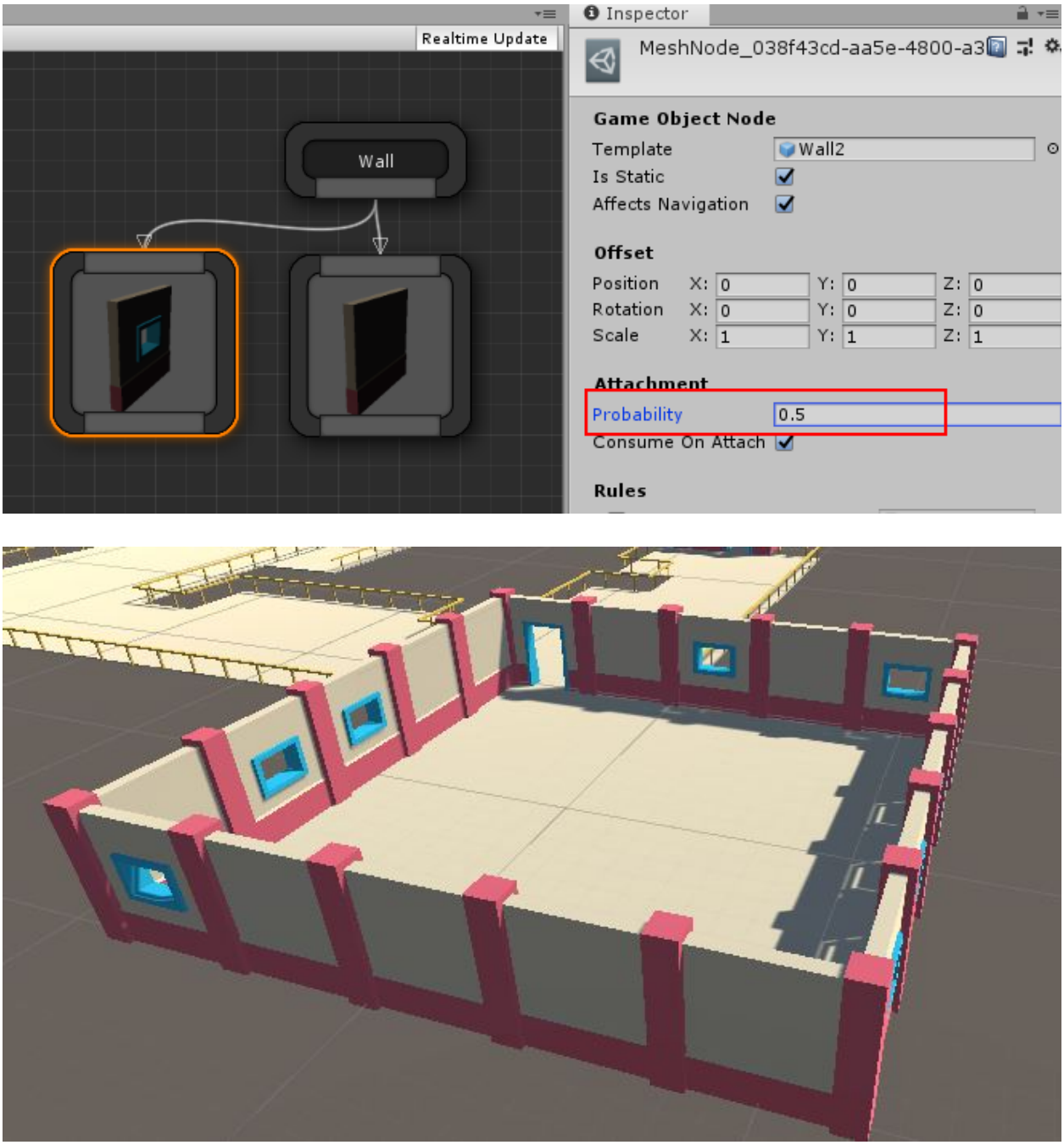
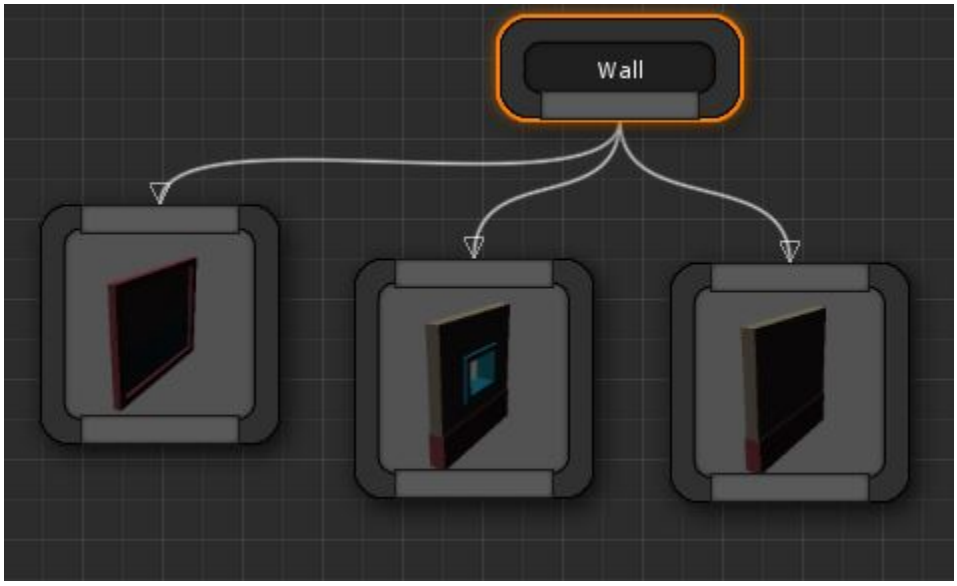
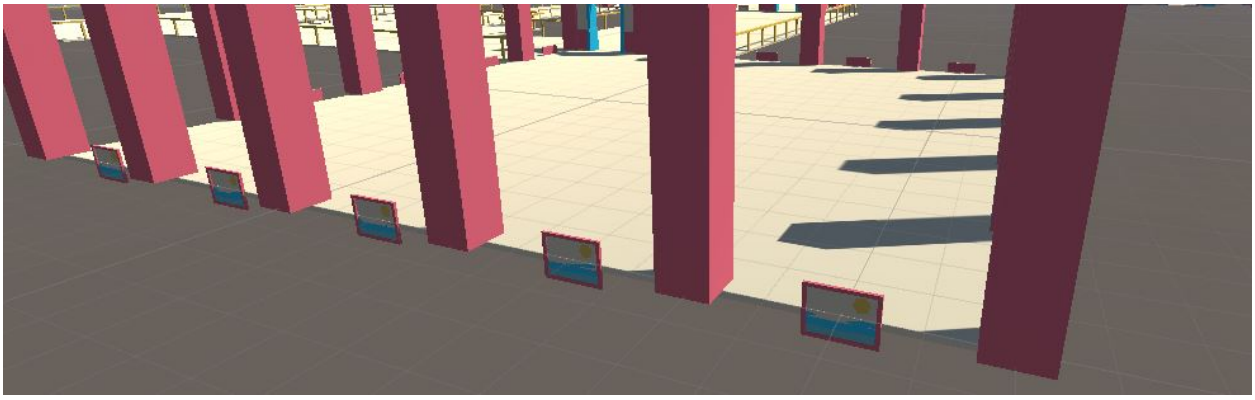
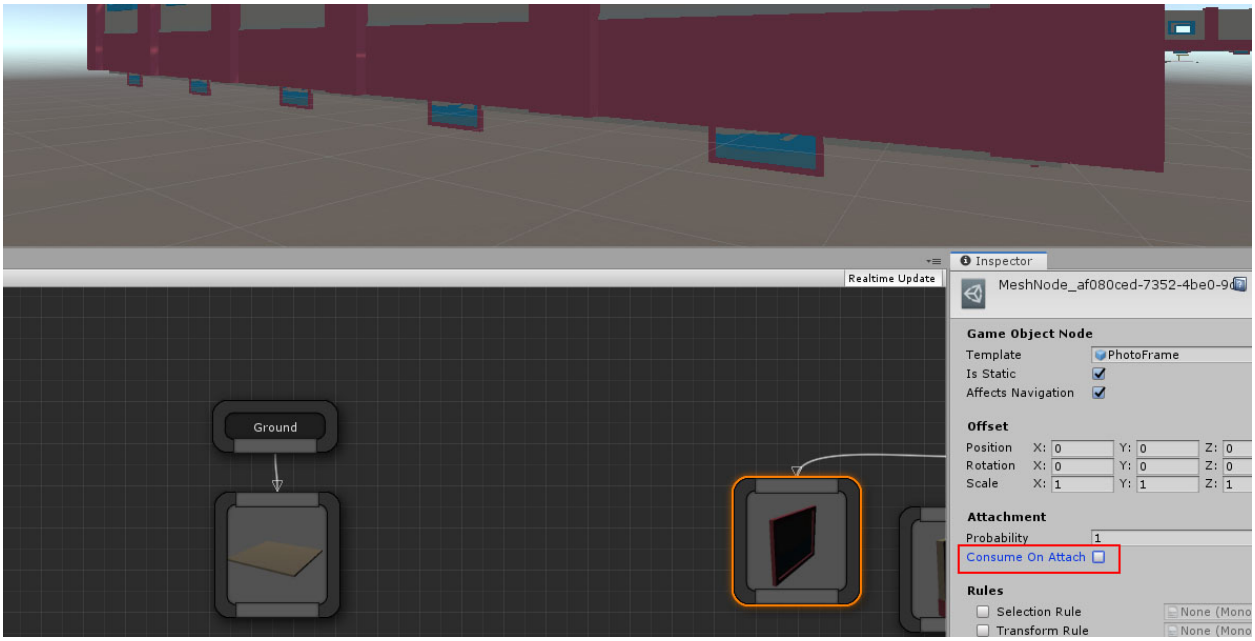
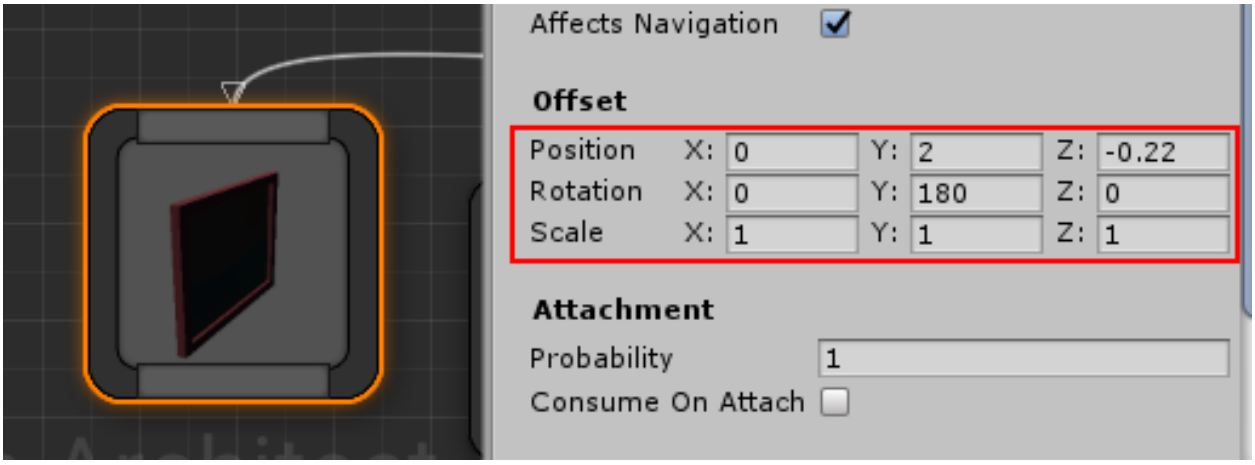


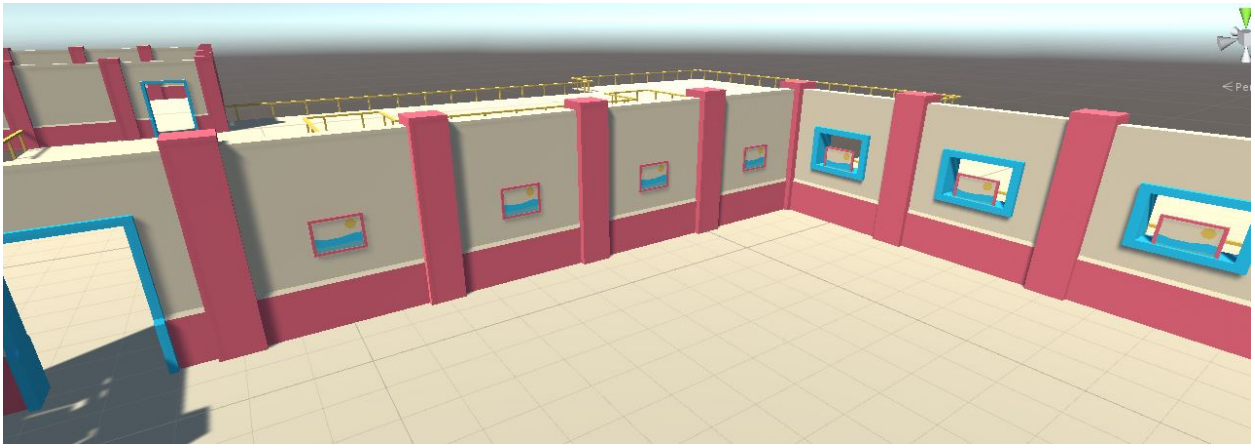

Fig. 3: Half the walls have windows

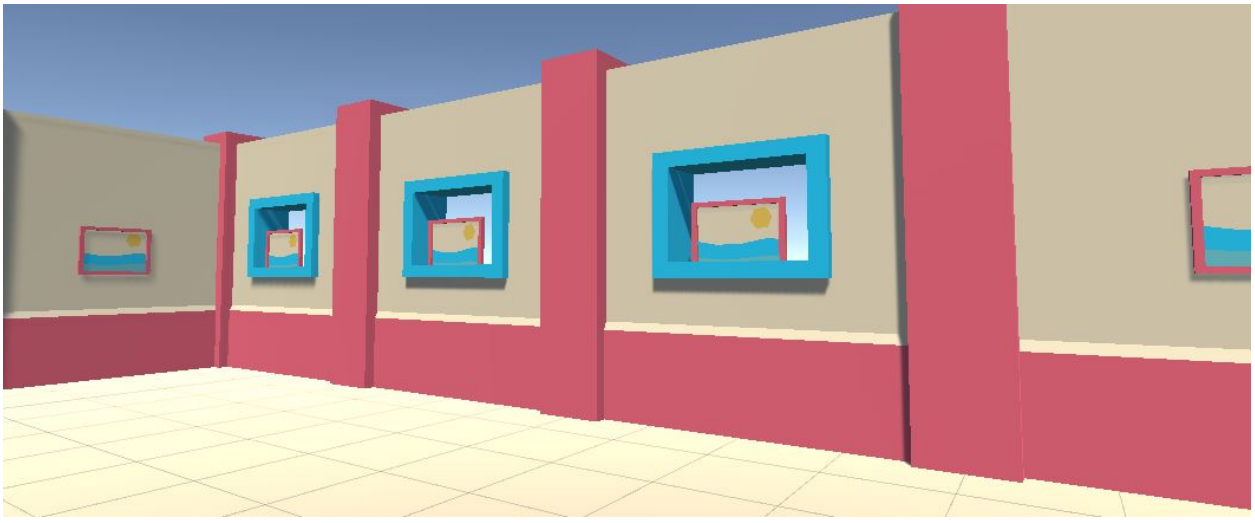

Wall

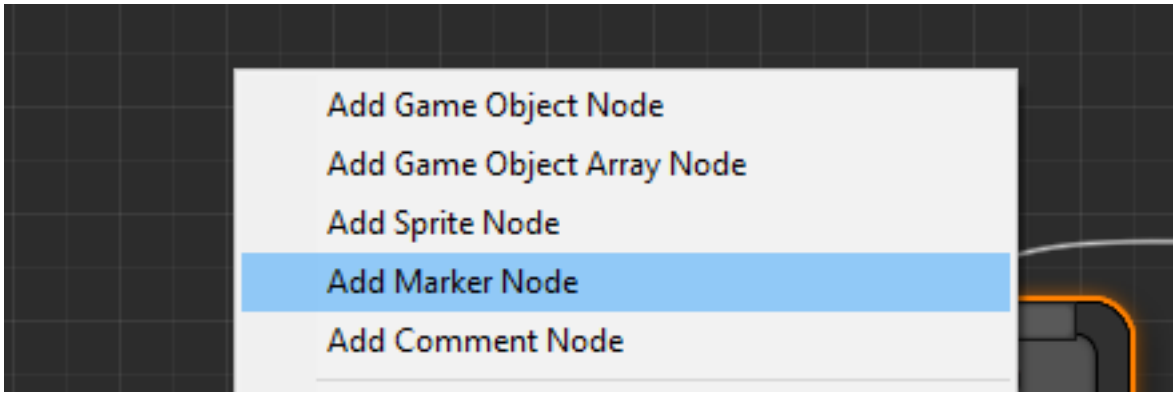


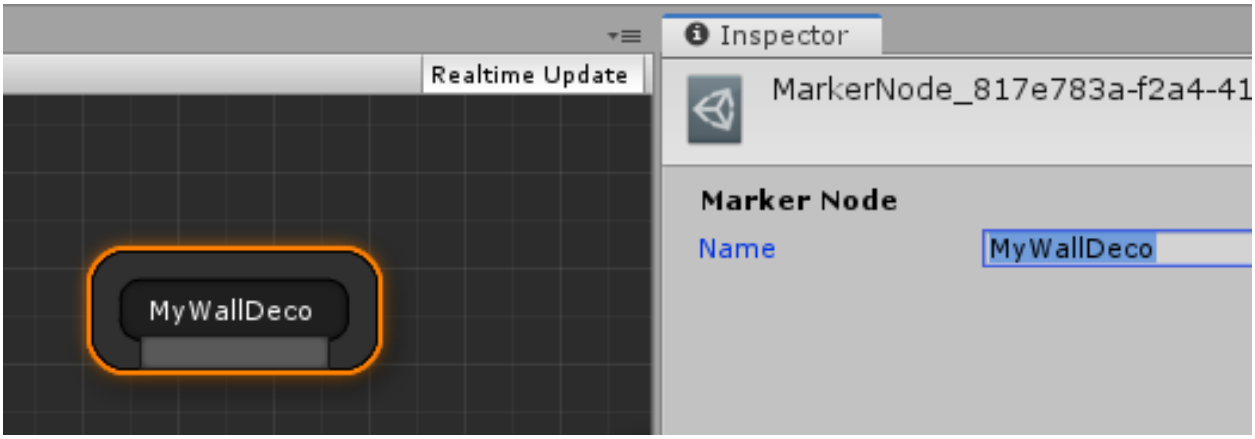

Realtime Update
Inspector
MeshNode_af080ced-7352-4be0-9
Game Object Node
Template
PhotoFrame
Is Static
Affects Navigation
Offset
Position
Rotation
Scale
Attachment
Probability
Consume On Attach
Rules
Selection Rule
Transform Rule
Ground


Affects Navigation
Offset
Position X: 0 Y: 2 Z: -0.22
Rotation X: 0 Y: 180 Z: 0
Scale X: 1 Y: 1 Z: 1
Attachment
Probability 1
Consume On Attach








Sometimes, it is difficult to line up the ground node, as there is no point of reference to compare with.

In that case, turn on Debug Draw and build the dungeon.

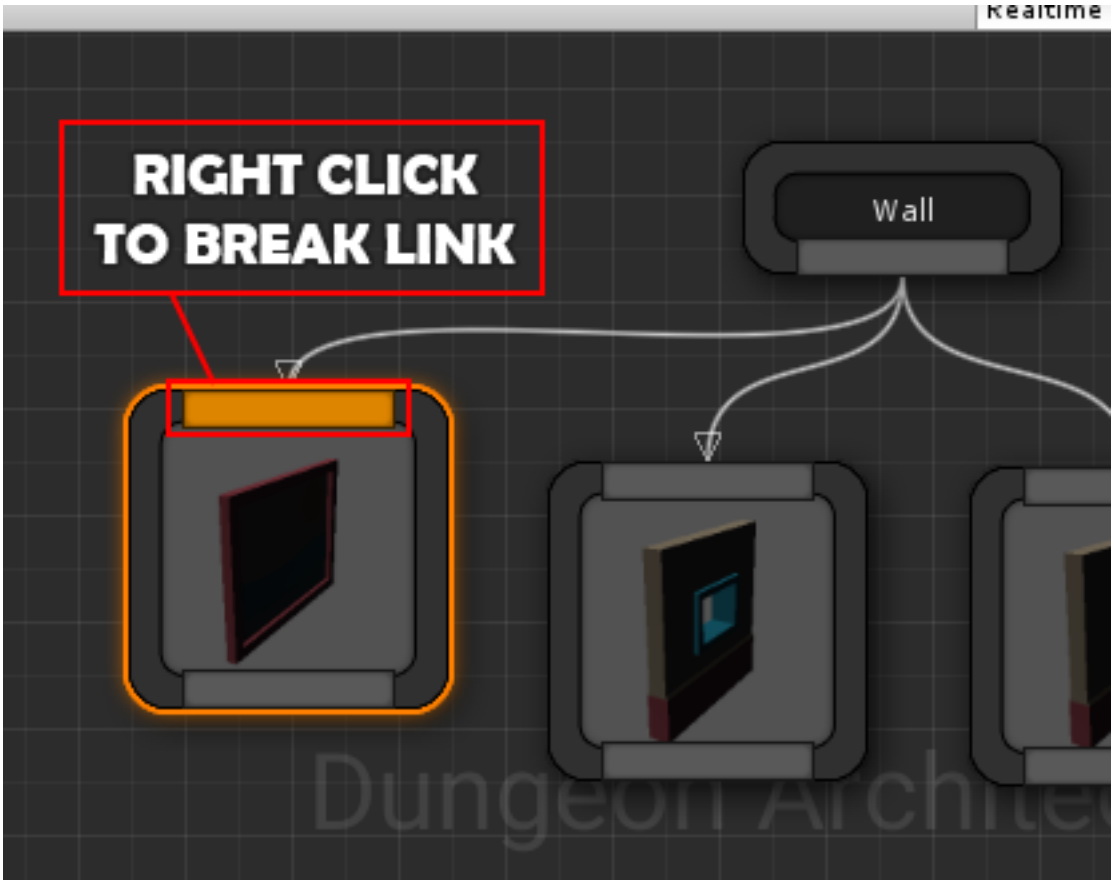
This ground asset has its pivot on the corner instead of the center. We'll add an offset of position `(-2, 0, -2)` to fix it and align with the debug drawn boundaries

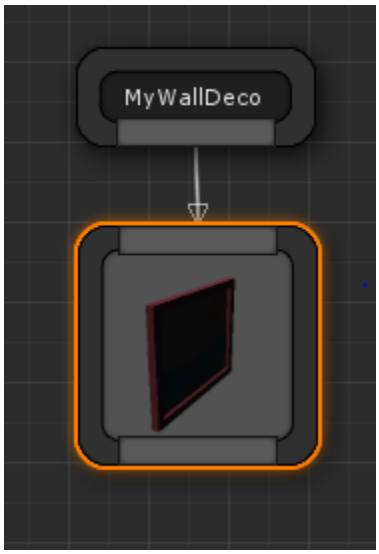
It is important to first align the ground mesh and then use that as a reference to align your walls and fences. If the ground is not aligned correctly, the rest will also not align (since you will be using an incorrect point of reference for aligning the rest)

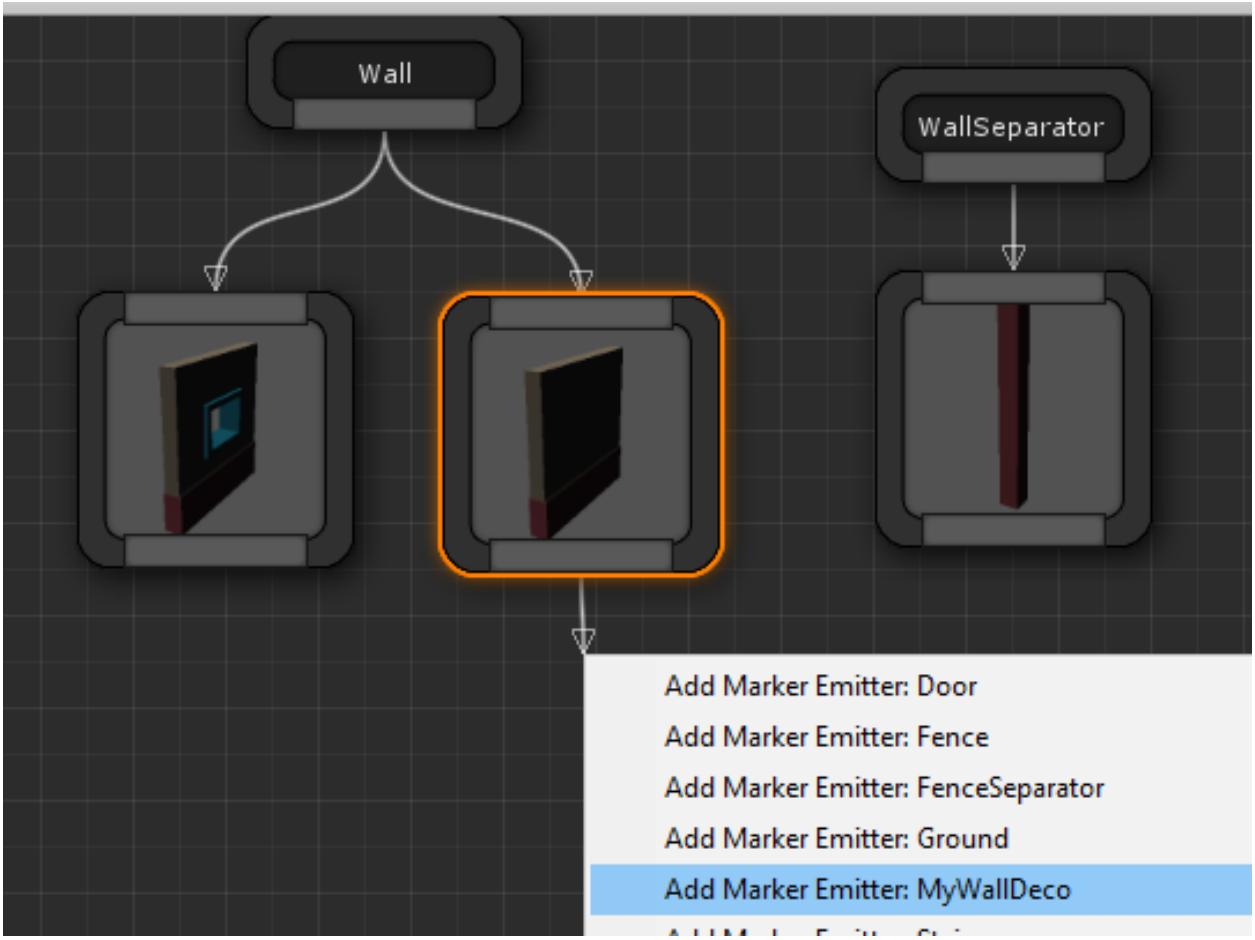
### 2.4.8 Recap

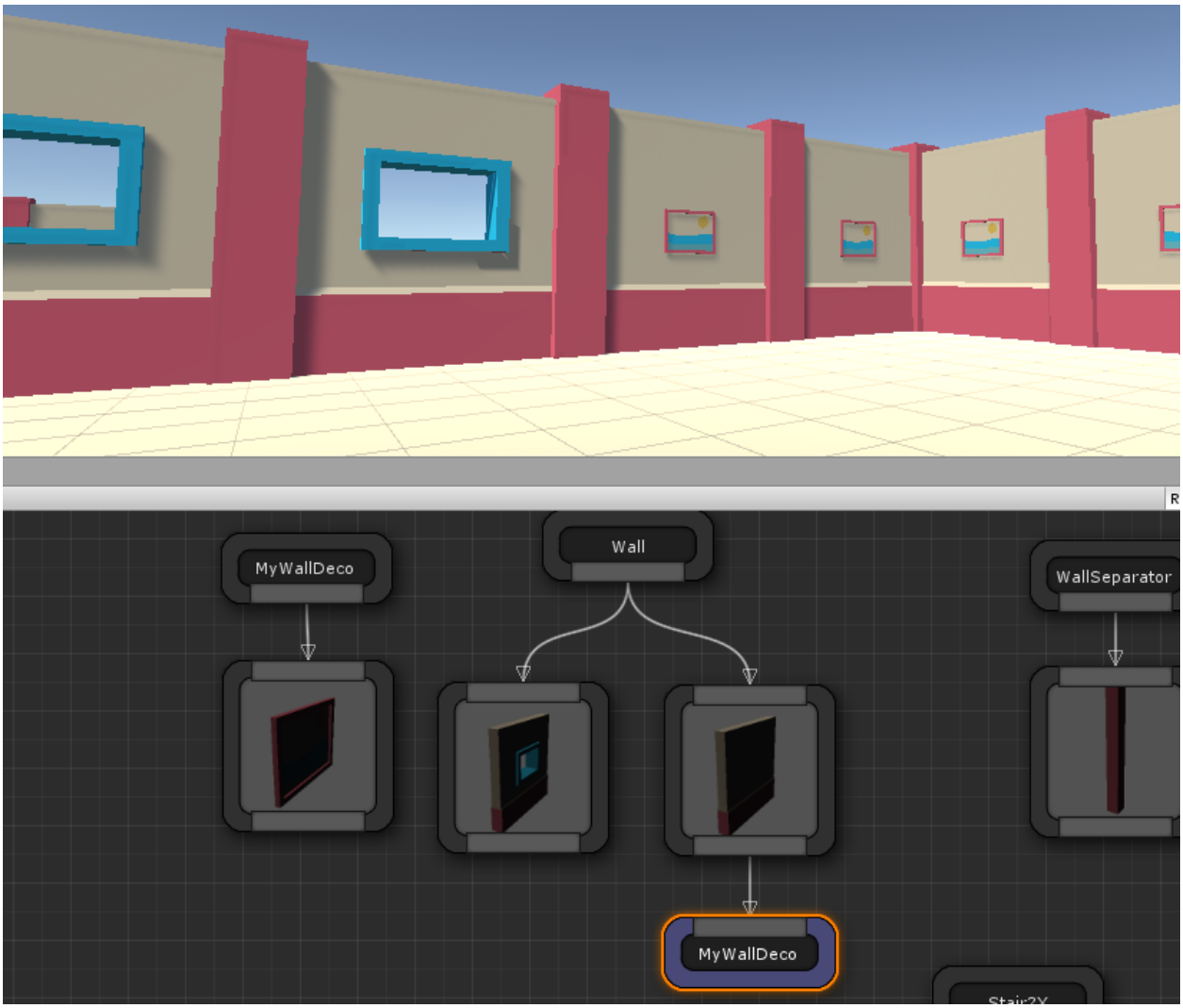
In this section we learnt the following:

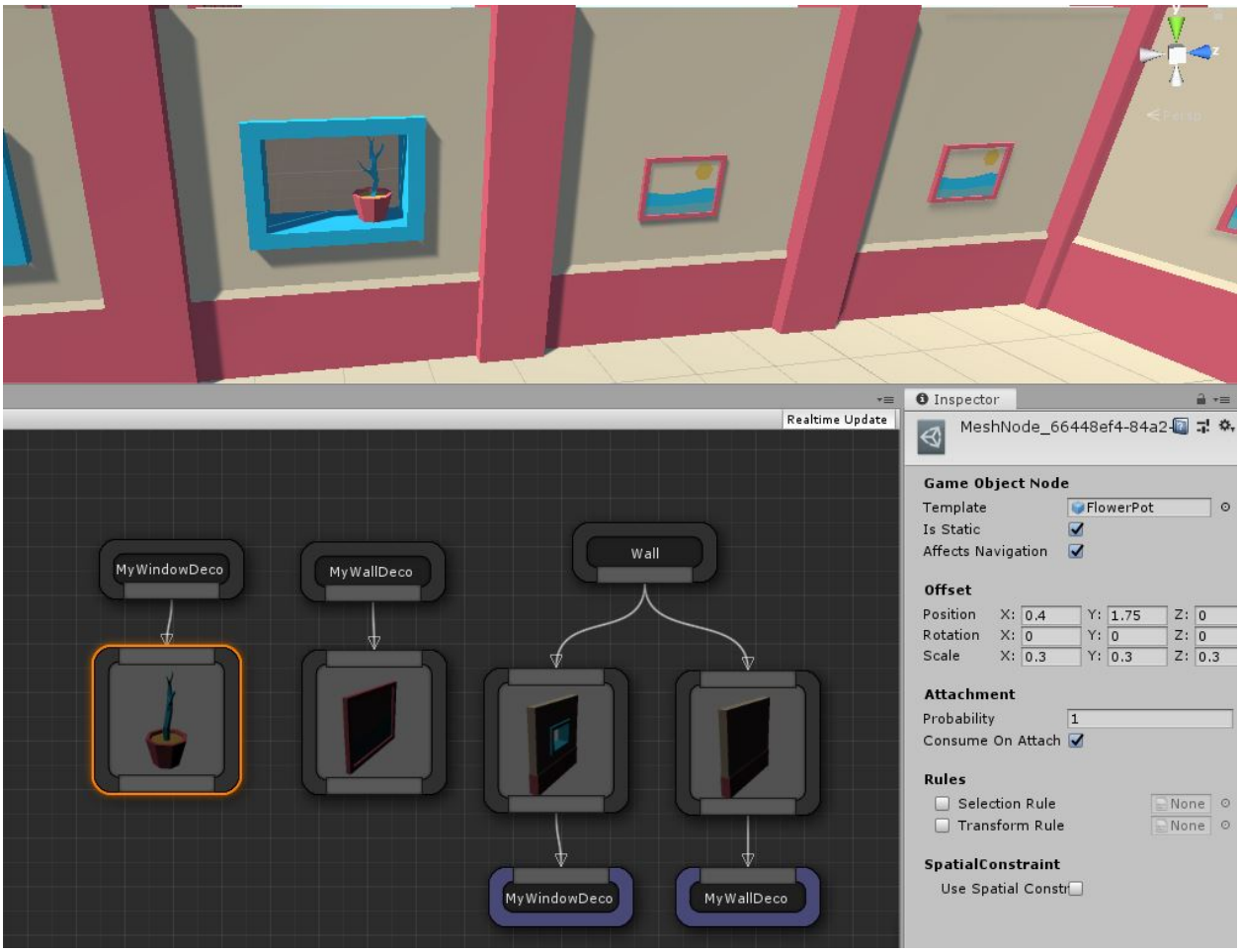
- *Probablity* - Controls the percentage chance of a node being selected. A value of 1 means 100% selection chance. A value of 0.25 means 25% selection chance
- *Execution Order* - The theme engine executes all the nodes under a marker node from left to right. If it selects a certain node, it stops executing, unless the `Consume on Attach` flag is unchecked
- *Marker Emitters* - You can create complex hierarchies with your own marker nodes, giving you more freedom to decorate your dungeons

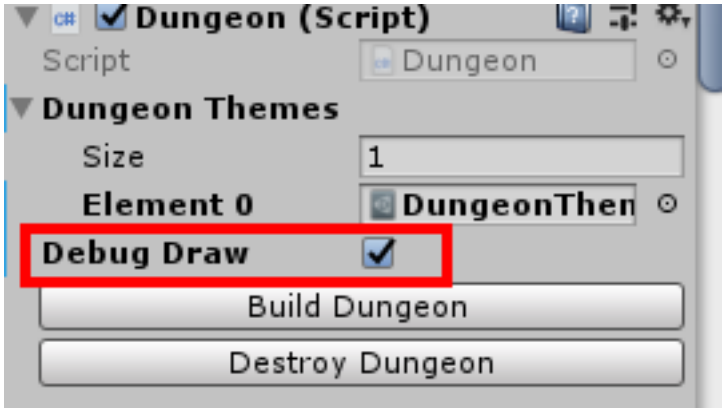

RIGHT CLICK
TO BREAK LINK
Wall

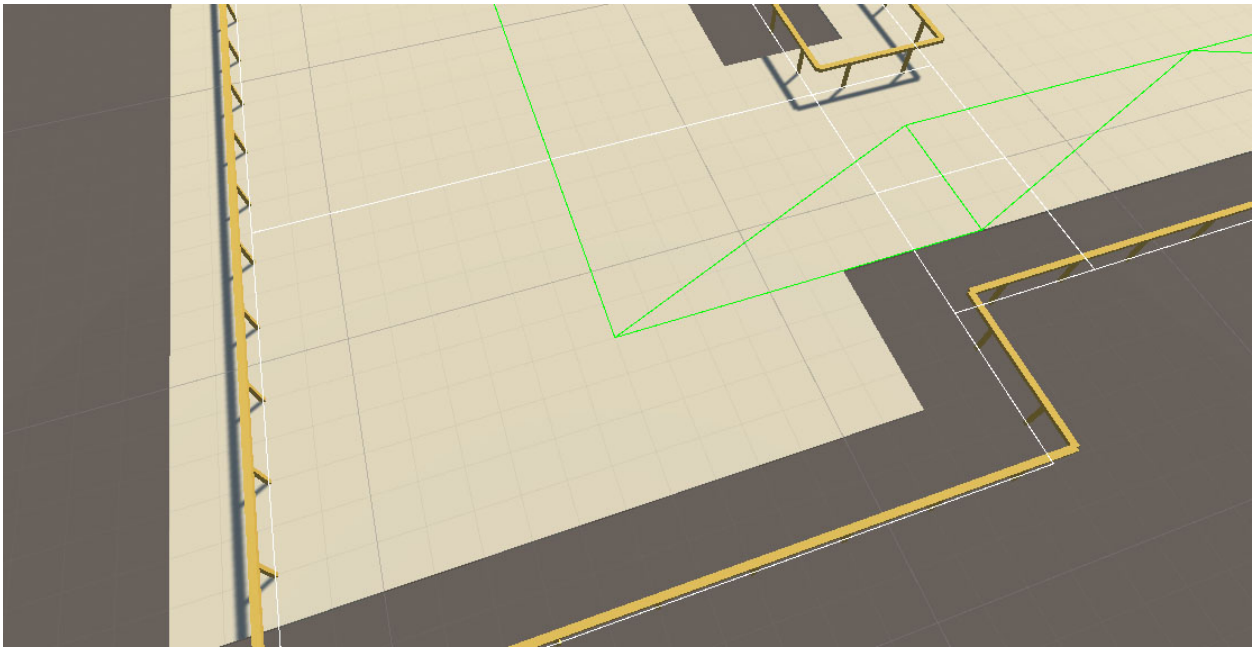

MyWallDeco

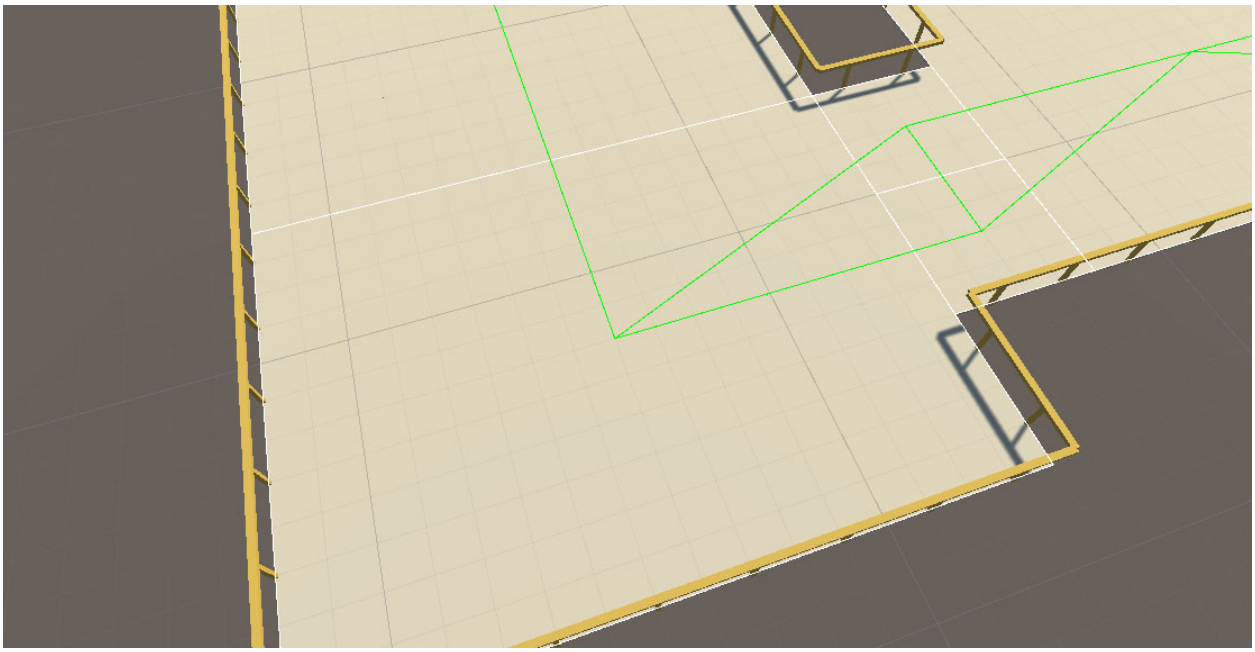

Wall
WallSeparator
Add Marker Emitter: Door
Add Marker Emitter: Fence
Add Marker Emitter: FenceSeparator
Add Marker Emitter: Ground
Add Marker Emitter: MyWallDeco


MyWallDeco
Wall
WallSeparator
MyWallDeco


Realtime Update
MyWindowDeco
MyWallDeco
Wall
MyWindowDeco
MyWallDeco
Inspector
MeshNode_66448ef4-84a2-
Game Object Node
Template
FlowerPot
Is Static
Affects Navigation
Offset
Position X: 0.4 Y: 1.75 Z: 0
Rotation X: 0 Y: 0 Z: 0
Scale X: 0.3 Y: 0.3 Z: 0.3
Attachment
Probability
1
Consume On Attach
Rules
Selection Rule
None
Transform Rule
None
SpatialConstraint
Use Spatial Constr


Dungeon (Script)
Script
Dungeon
Dungeon Themes
Size
1
Element 0
DungeonThen
Debug Draw
Build Dungeon
Destroy Dungeon





CHAPTER

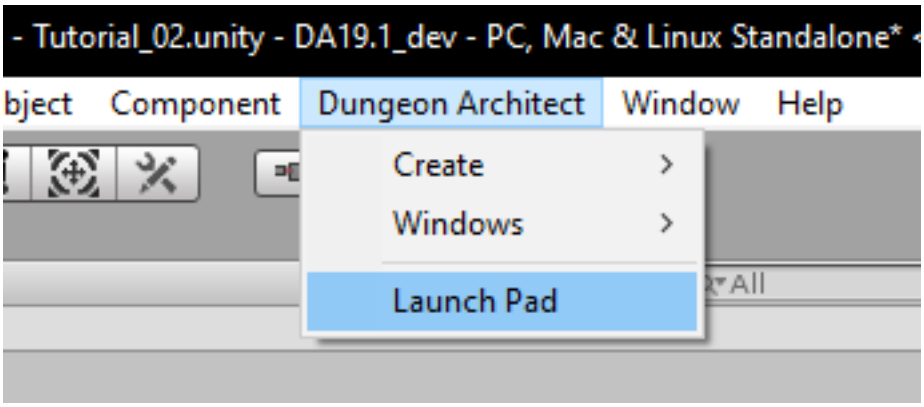
THREE

# LAUNCH PAD WINDOW

Use the Launch Pad window to setup new dungeon scenes, browse the samples, clone from templates and much more

## 3.1 Open Launch Pad

From the Main menu, open the Launch Pad window `Dungeon Architect > Launch Pad`




## 3.2 Navigation

Select the various sections from the left.

Use the navigation bar on the top to go back to a previous page. This is useful for retaining the scroll positions of the previous page (especially for larger pages like the Samples section)

## 3.3 Builder Templates

Dungeon Architect supports many different types of dungeon layout methods and is designed in a way that new layout methods can be easily added in the future

These layout methods are called *Dungeon Builders* or **Builders** in short

This section lets you create a new scene preconfigured with one of the builder templates. Click on any of the builders. In the next screen, click the `Clone Scene` button

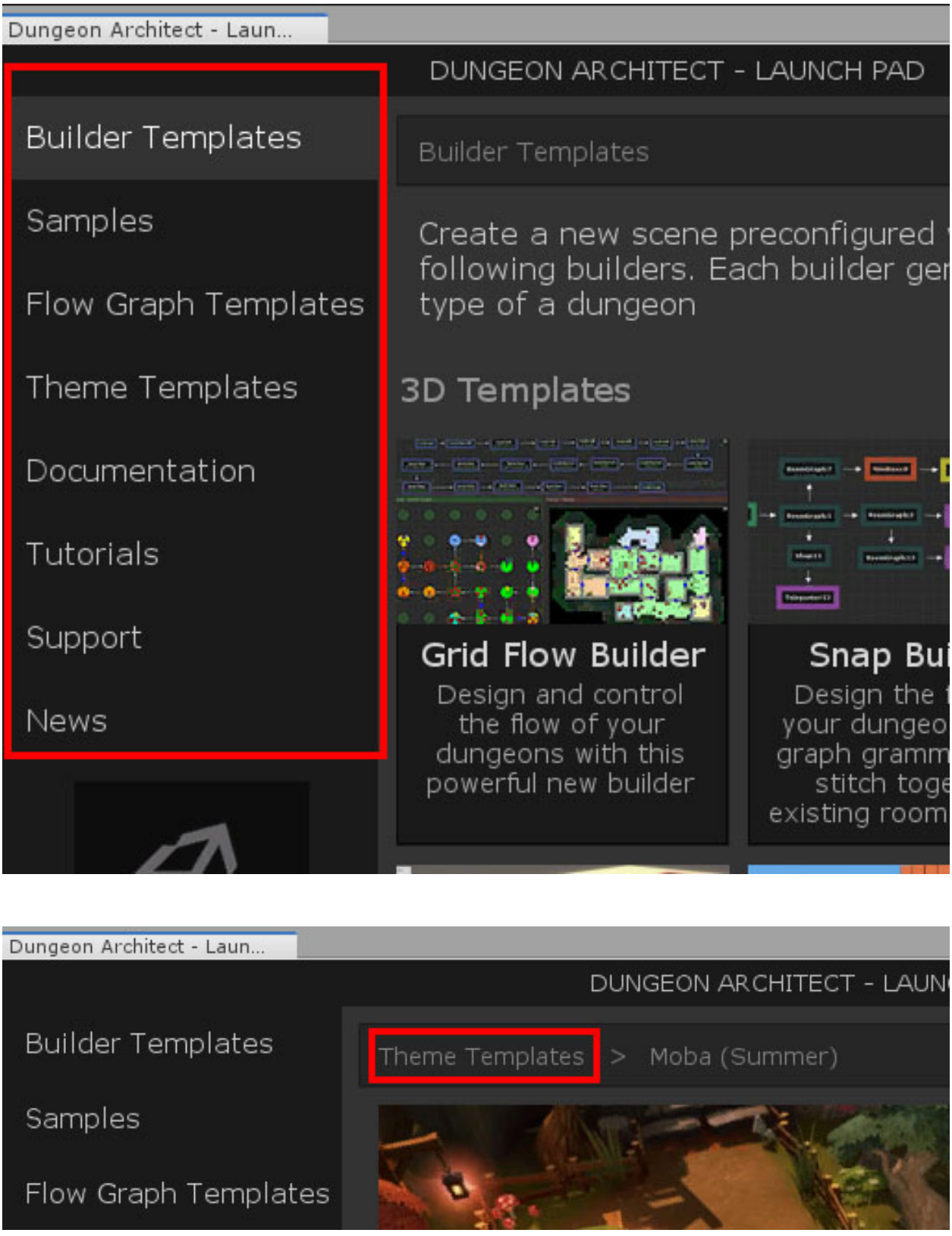



Fig. 1: Navigation bar

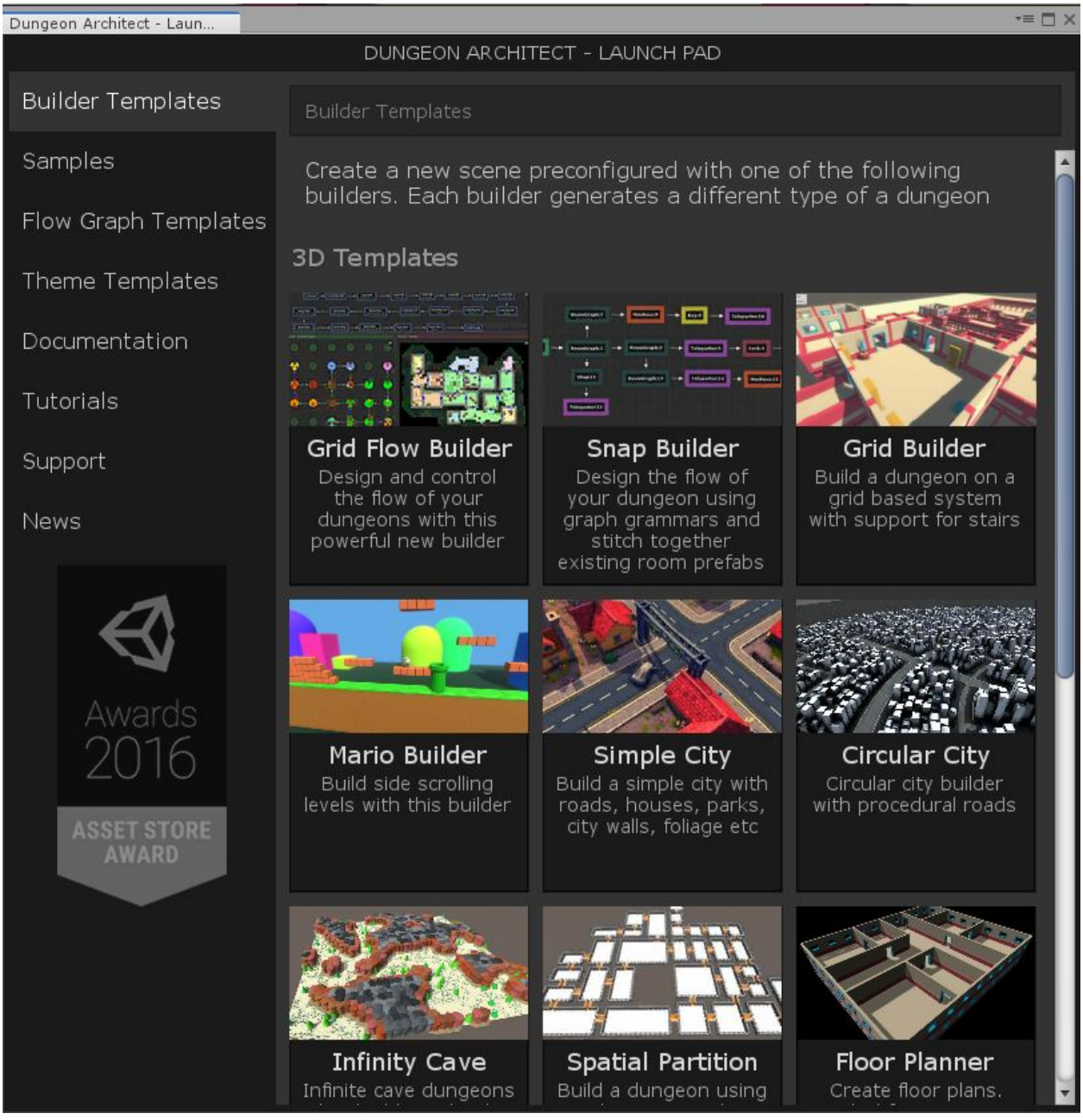
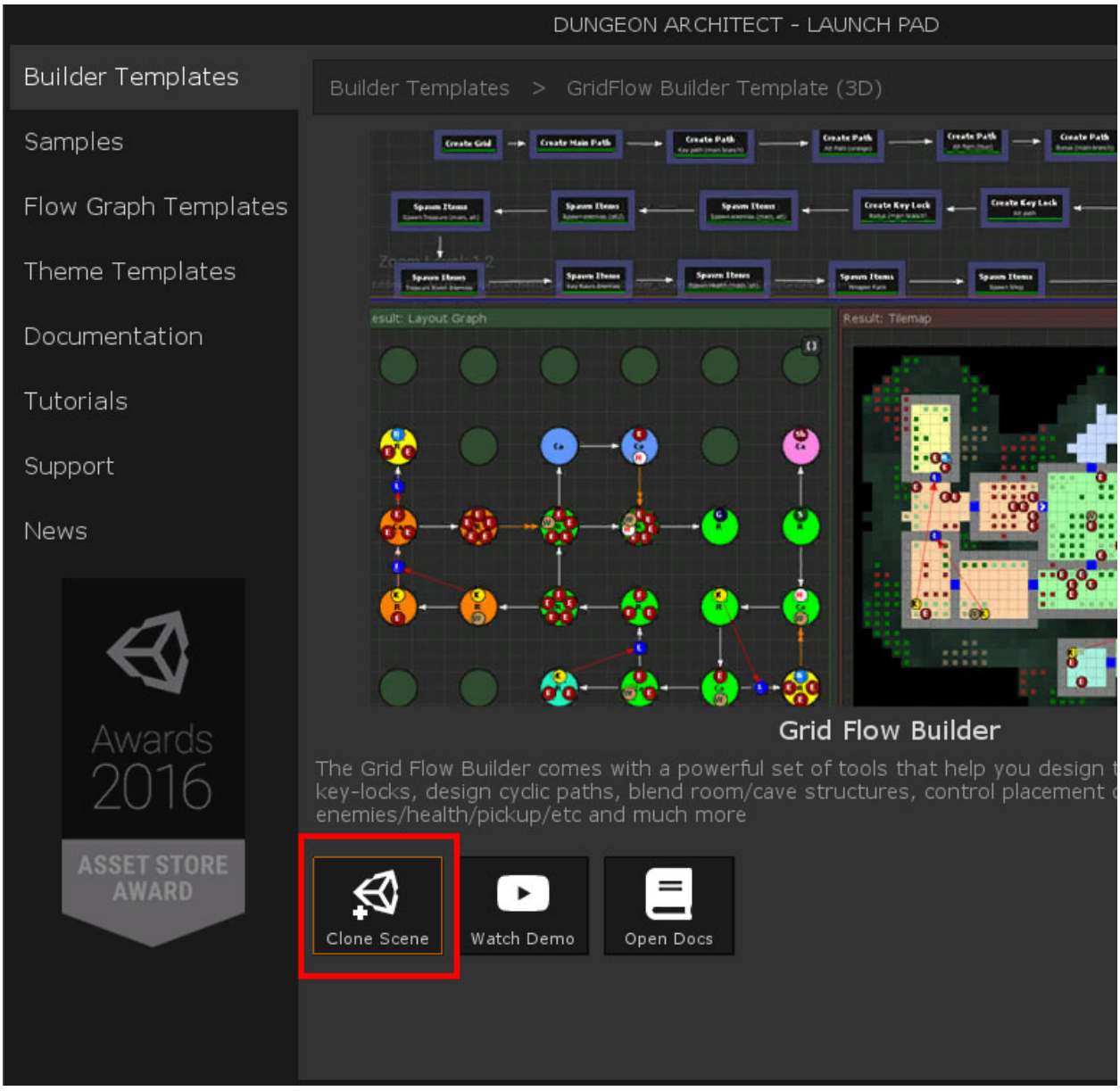



Fig. 2: Dungeon Architect - List of Builders

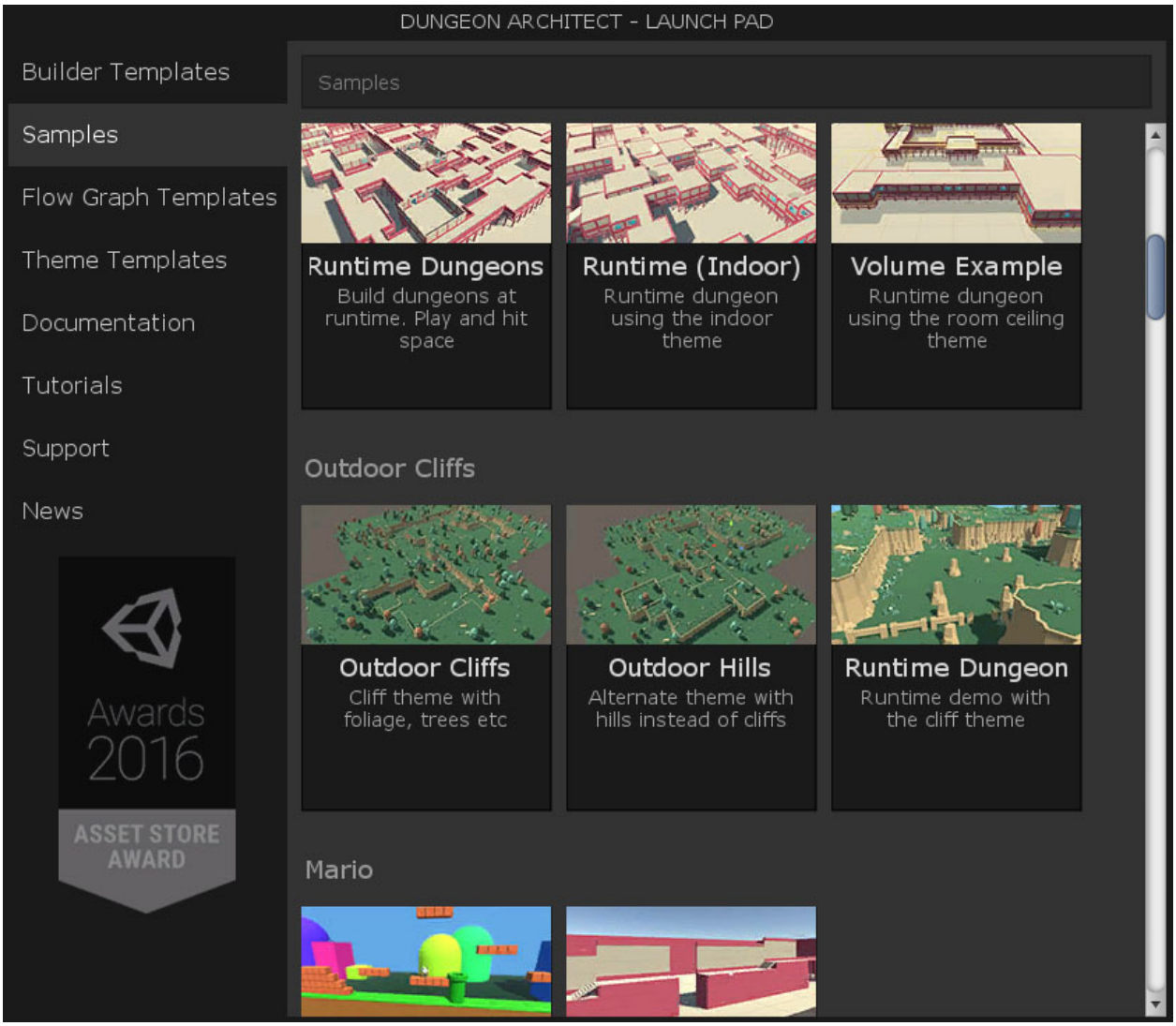

DUNGEON ARCHITECT - LAUNCH PAD
Builder Templates
Samples
Flow Graph Templates
Theme Templates
Documentation
Tutorials
Support
News
Builder Templates > GridFlow Builder Template (3D)
Result: Layout Graph
Result: Tilemap
Grid Flow Builder
The Grid Flow Builder comes with a powerful set of tools that help you design
key-locks, design cyclic paths, blend room/cave structures, control placement
enemies/health/pickup/etc and much more
Clone Scene
Watch Demo
Open Docs
Awards
2016
ASSET STORE
AWARD

This will create a new scene based on the template, fully configured with the appropriate dungeon. It will also clone a starting theme file (and any other flow graph assets) and set everything up.

Choose a folder to saved your scene file. Once saved, the launcher would do the following:

- Open the new scene
- Open any theme editor windows associated with the referenced assets (Theme Editor, Flow Graph Editors etc)
- Select the dungeon game object (so you see the properties in the inspector by default)

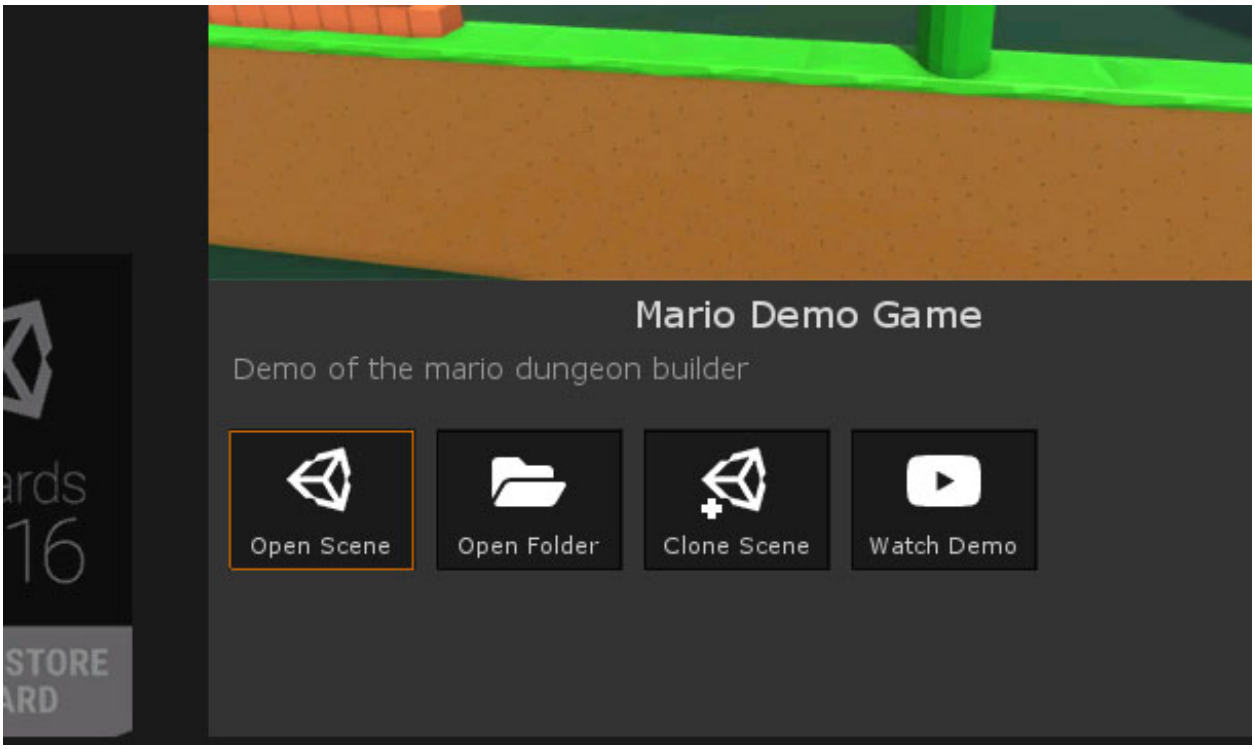
## 3.4 Samples




There are tons of samples to explore. Select a sample you like and perform one of the following actions

- **Open Scene**: Opens the sample scene (usually under DungeonArchitect_Samples folder)
- **Open Folder**: Opens the folder containing the scene
- **Clone Scene**: Clone a scene and also clone over the referenced assets (themes, flow graphs etc) so you can modify them without affecting the sample scene

- **Watch Demo**: Watch a video, if it exists




## 3.5 Flow Graph Templates

Flow graphs allow you to control the flow of your dungeon (more on this in the later tutorial sections). This section contains a list of flow graph templates you can use as a starting point for your project

## 3.6 Theme Templates

Clone one of the many themes and use it in your project or as a starting point for a new theme

Select a theme and clone it

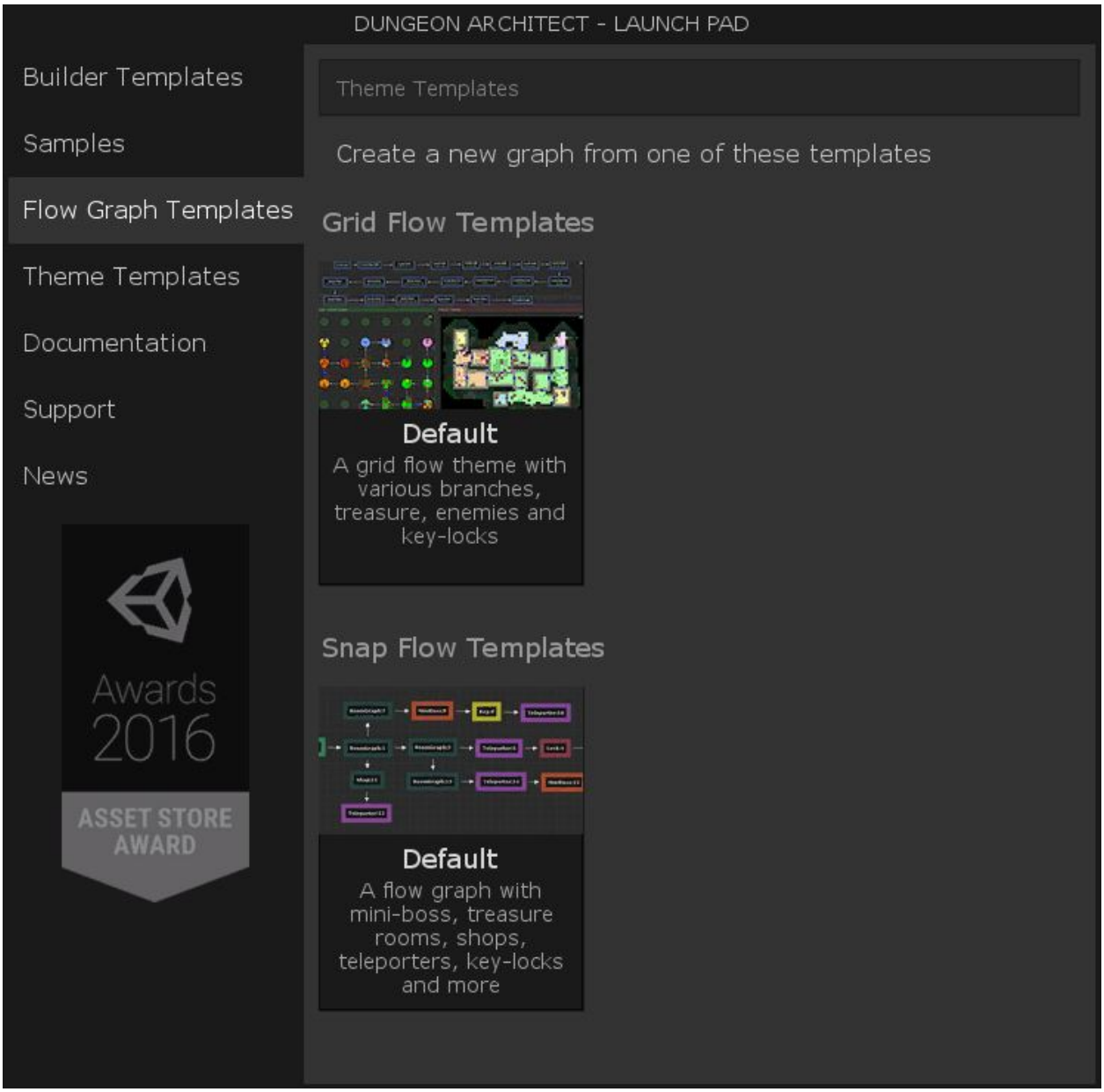
There's also a category on *External Themes*, the ones that use external paid art assets from the Asset Store. You can use these themes if you also own the art asset.

## 3.7 Documentation

Links to various online documentation, including this one

## 3.8 Support

Reach the developers through any one of these channels. Interact with the community and the devs in Discord chat and forums or reach directly through email


DUNGEON ARCHITECT - LAUNCH PAD
Builder Templates
Samples
Flow Graph Templates
Theme Templates
Documentation
Support
News
Theme Templates
Create a new graph from one of these templates
Grid Flow Templates
Default
A grid flow theme with various branches, treasure, enemies and key-locks
Snap Flow Templates
Default
A flow graph with mini-boss, treasure rooms, shops, teleporters, key-locks and more
Awards 2016
ASSET STORE AWARD

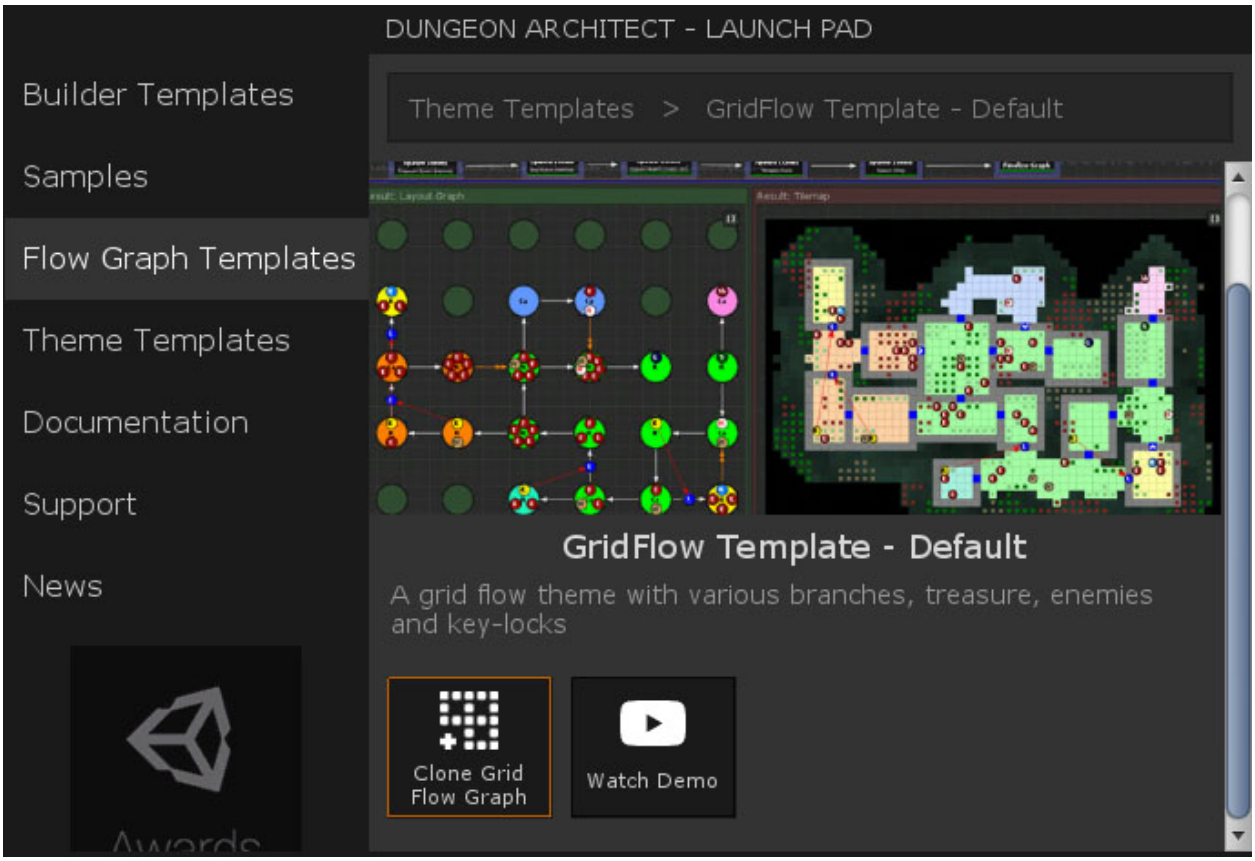



Fig. 3: Clone Grid Flow Graphs

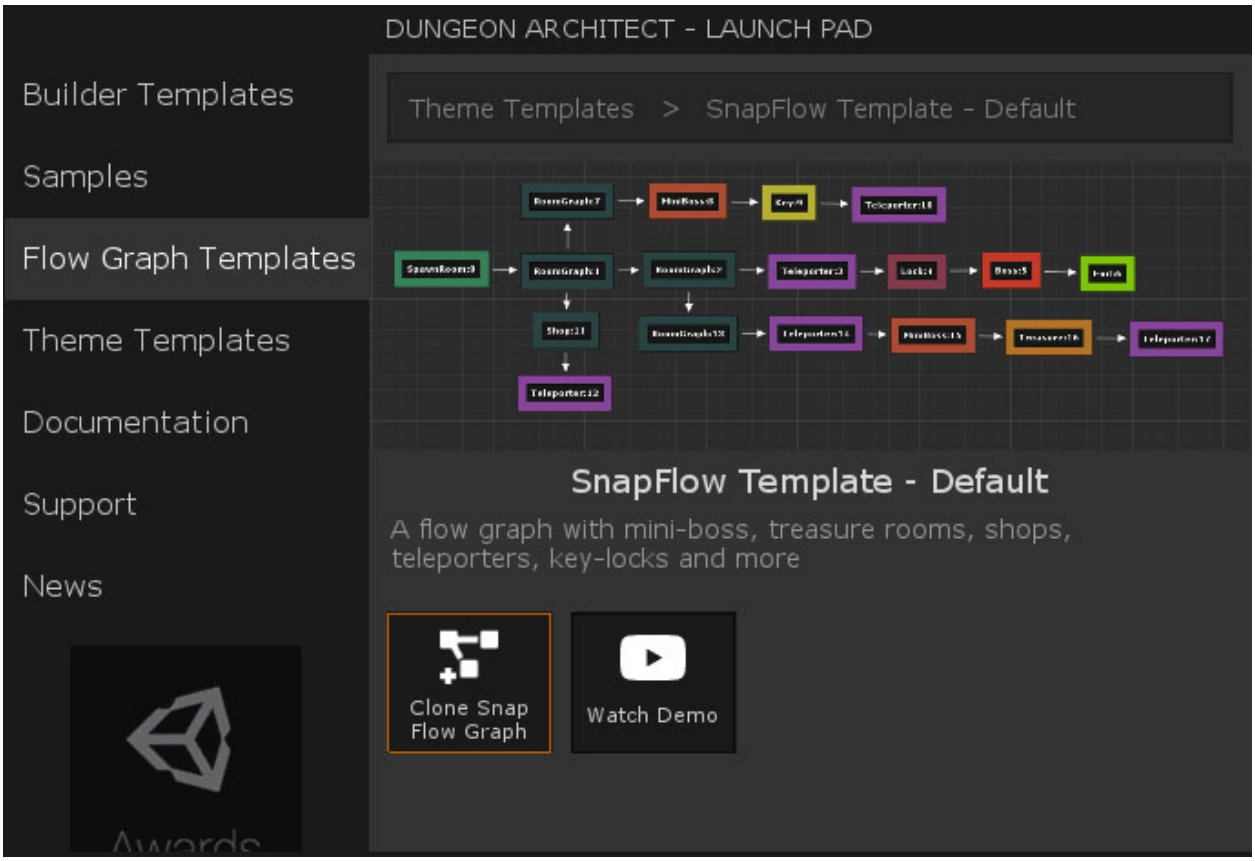



Fig. 4: Clone Snap Flow Graphs

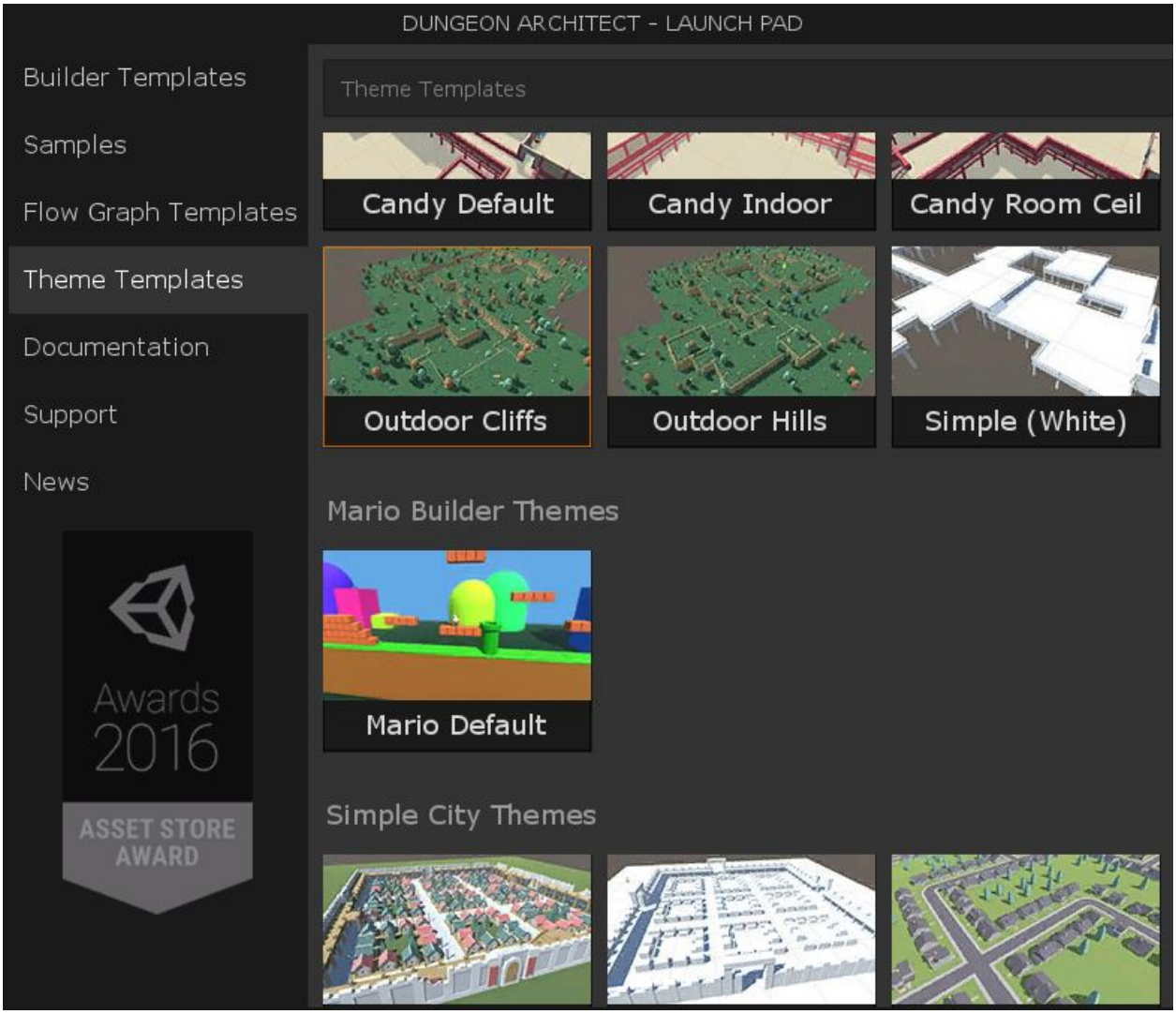
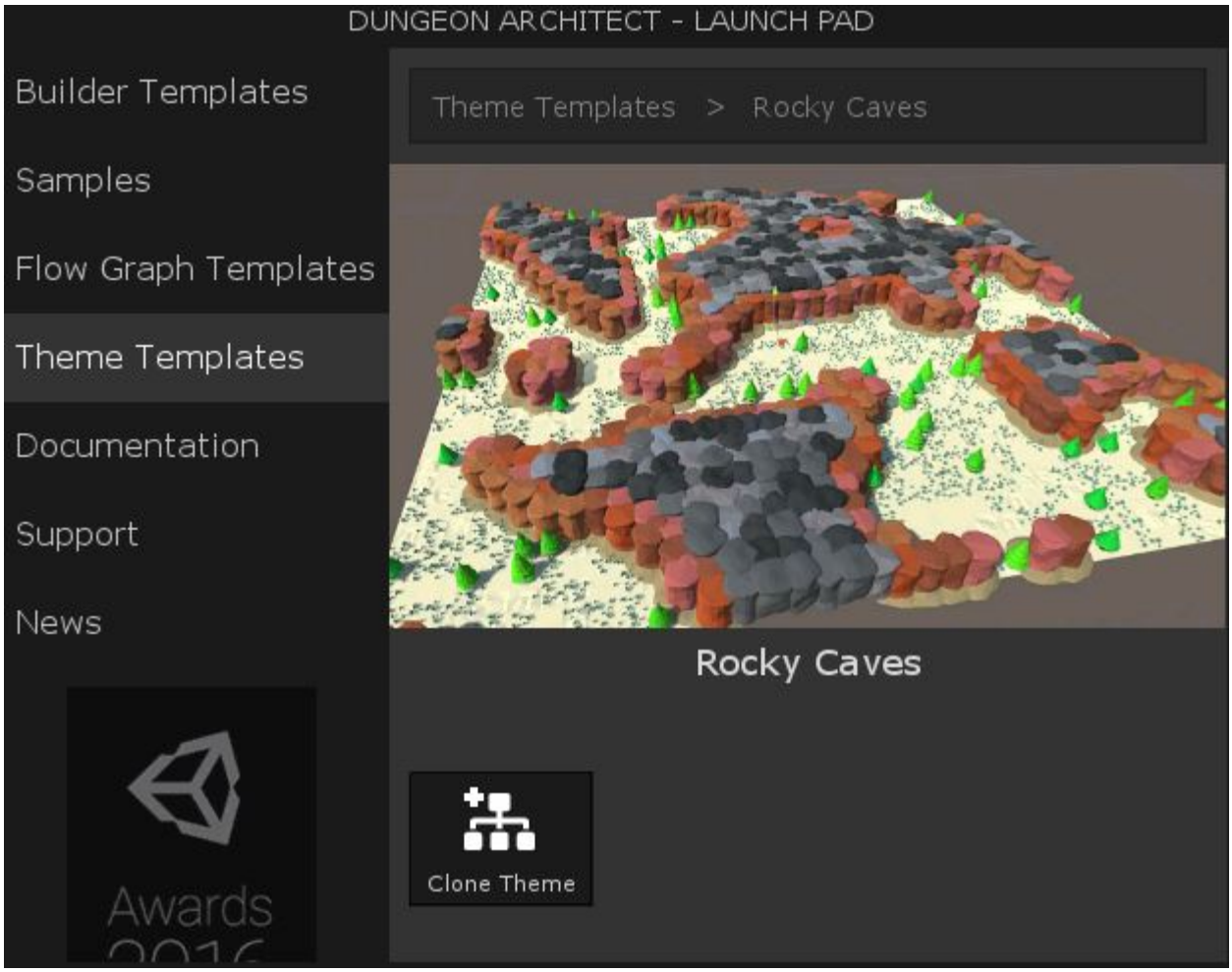
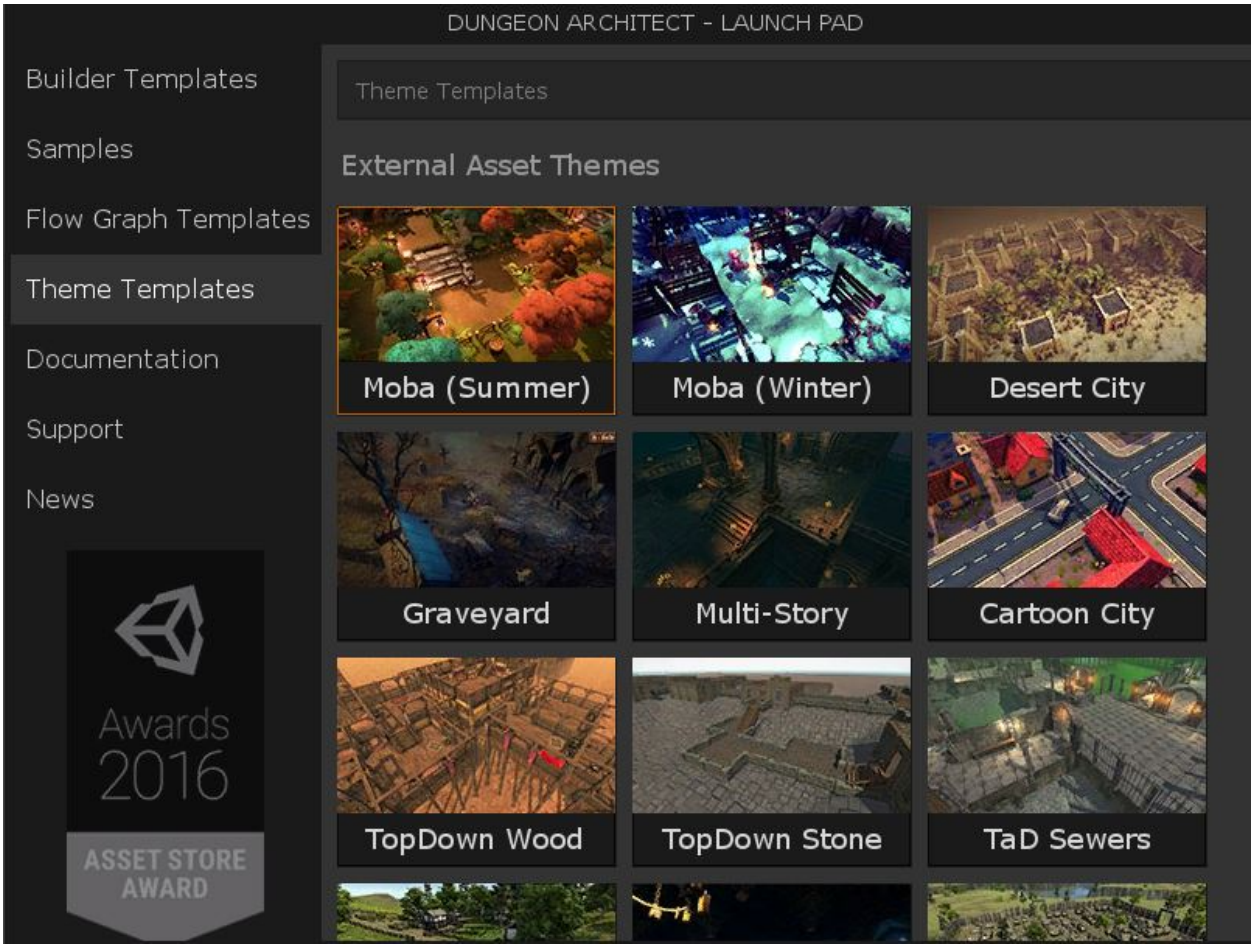
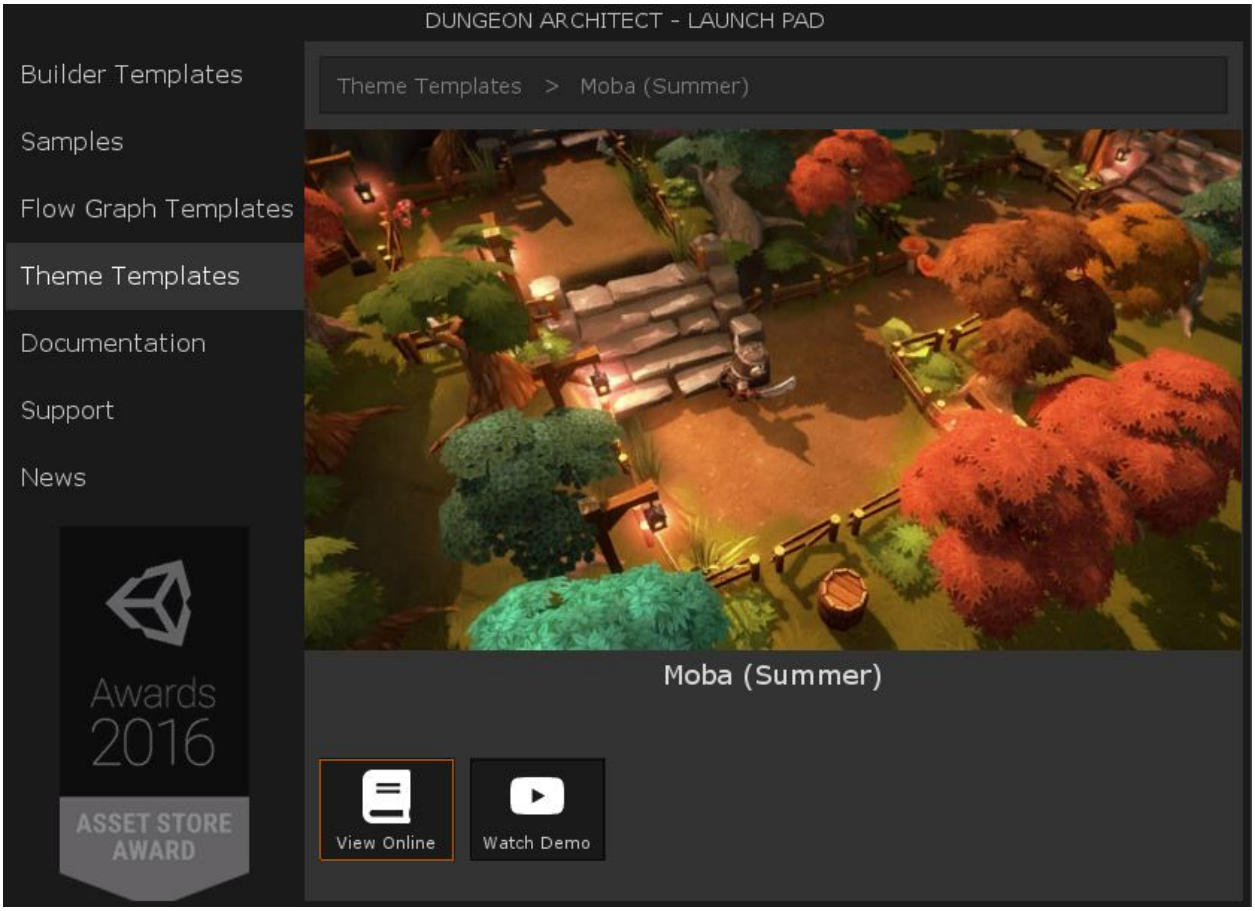
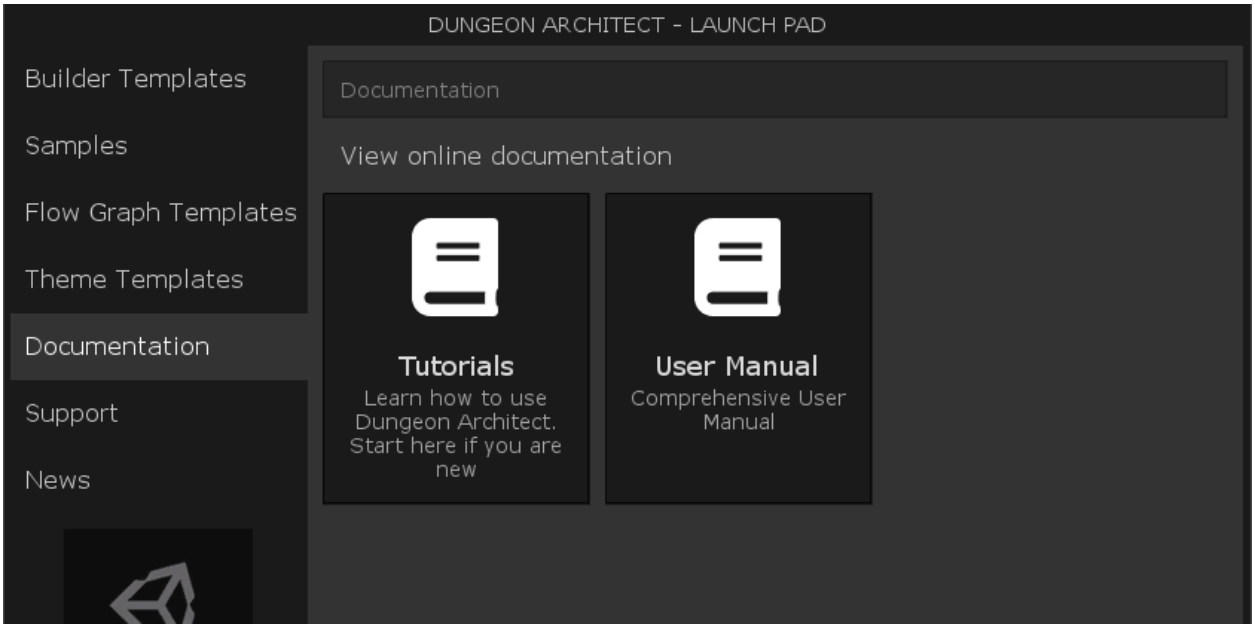


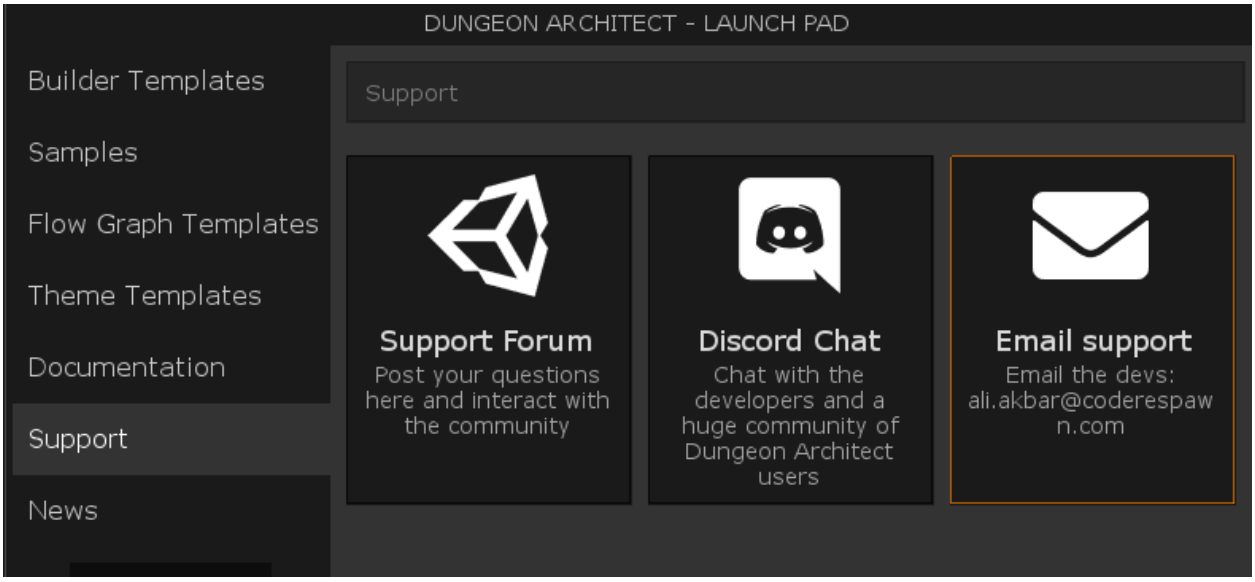

Fig. 5: Theme template browser


DUNGEON ARCHITECT - LAUNCH PAD
Builder Templates
Samples
Flow Graph Templates
Theme Templates
Documentation
Support
News
Theme Templates > Rocky Caves
Rocky Caves
Clone Theme
Awards


DUNGEON ARCHITECT - LAUNCH PAD
Builder Templates
Samples
Flow Graph Templates
Theme Templates
Documentation
Support
News
Awards
2016
ASSET STORE
AWARD
Theme Templates
External Asset Themes
Moba (Summer)
Moba (Winter)
Desert City
Graveyard
Multi-Story
Cartoon City
TopDown Wood
TopDown Stone
TaD Sewers

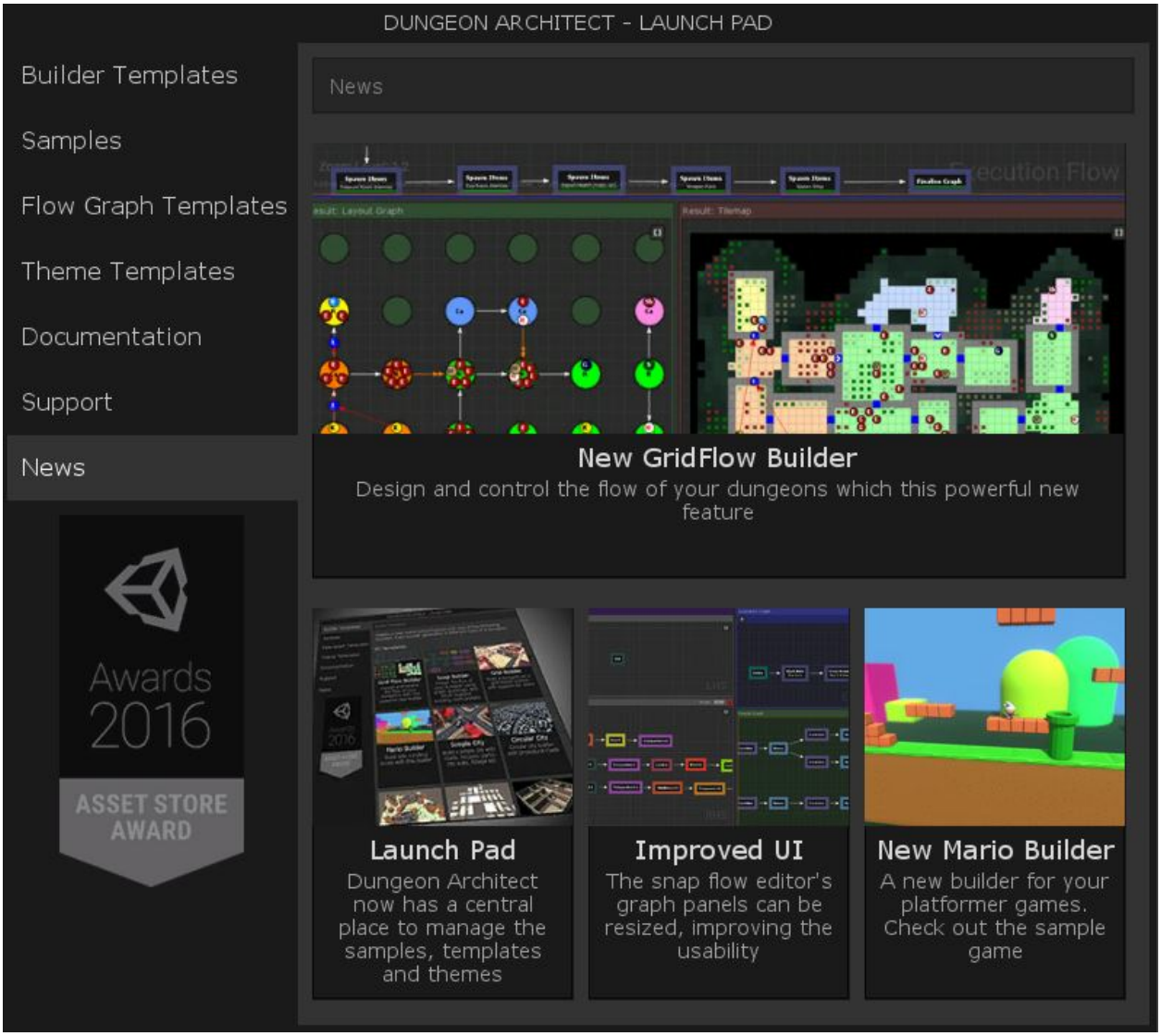

DUNGEON ARCHITECT - LAUNCH PAD
Builder Templates
Samples
Flow Graph Templates
Theme Templates
Documentation
Support
News
Theme Templates > Moba (Summer)
Moba (Summer)
View Online
Watch Demo
Awards
2016
ASSET STORE
AWARD


DUNGEON ARCHITECT - LAUNCH PAD
Builder Templates
Samples
Flow Graph Templates
Theme Templates
Documentation
Support
News
Documentation
View online documentation
Tutorials
Learn how to use Dungeon Architect. Start here if you are new
User Manual
Comprehensive User Manual




## 3.9 News

Dungeon Architect News! Find out whats new since the last update


DUNGEON ARCHITECT - LAUNCH PAD
Builder Templates
Samples
Flow Graph Templates
Theme Templates
Documentation
Support
News
Awards
2016
ASSET STORE
AWARD
News
New GridFlow Builder
Design and control the flow of your dungeons which this powerful new feature
Launch Pad
Dungeon Architect now has a central place to manage the samples, templates and themes
Improved UI
The snap flow editor's graph panels can be resized, improving the usability
New Mario Builder
A new builder for your platformer games. Check out the sample game

CHAPTER

FOUR

# GRID FLOW BUILDER

The Grid Flow Builder offers a rich set of tools to control the flow of your dungeons and item placement

## 4.1 Create a Grid Flow Dungeon

### 4.1.1 Setup Dungeon Prefab

Create a new scene. Navigate to `Assets/DungeonArchitect/Prefabs` and drop in the DungeonGridFlow prefab on to the scene

Select the DungeonGridFlow game object you just dropped and reset the transform

### 4.1.2 Setup Parent Object

Create a new Parent object where all the spawned dungeon items will go in.

Reset the parent object's transform and set it to static

Assign the parent object

This makes sure all the spawned dungeon objects are placed under this parent object

### 4.1.3 Setup Theme

There's a theme available in the samples folder which we'll use for this tutorial section

Navigate to `Assets\DungeonArchitect_Samples\DemoBuilder_GridFlow\Theme` and assign the theme `ThemePrehistoric` to the DungeonGridFlow game object

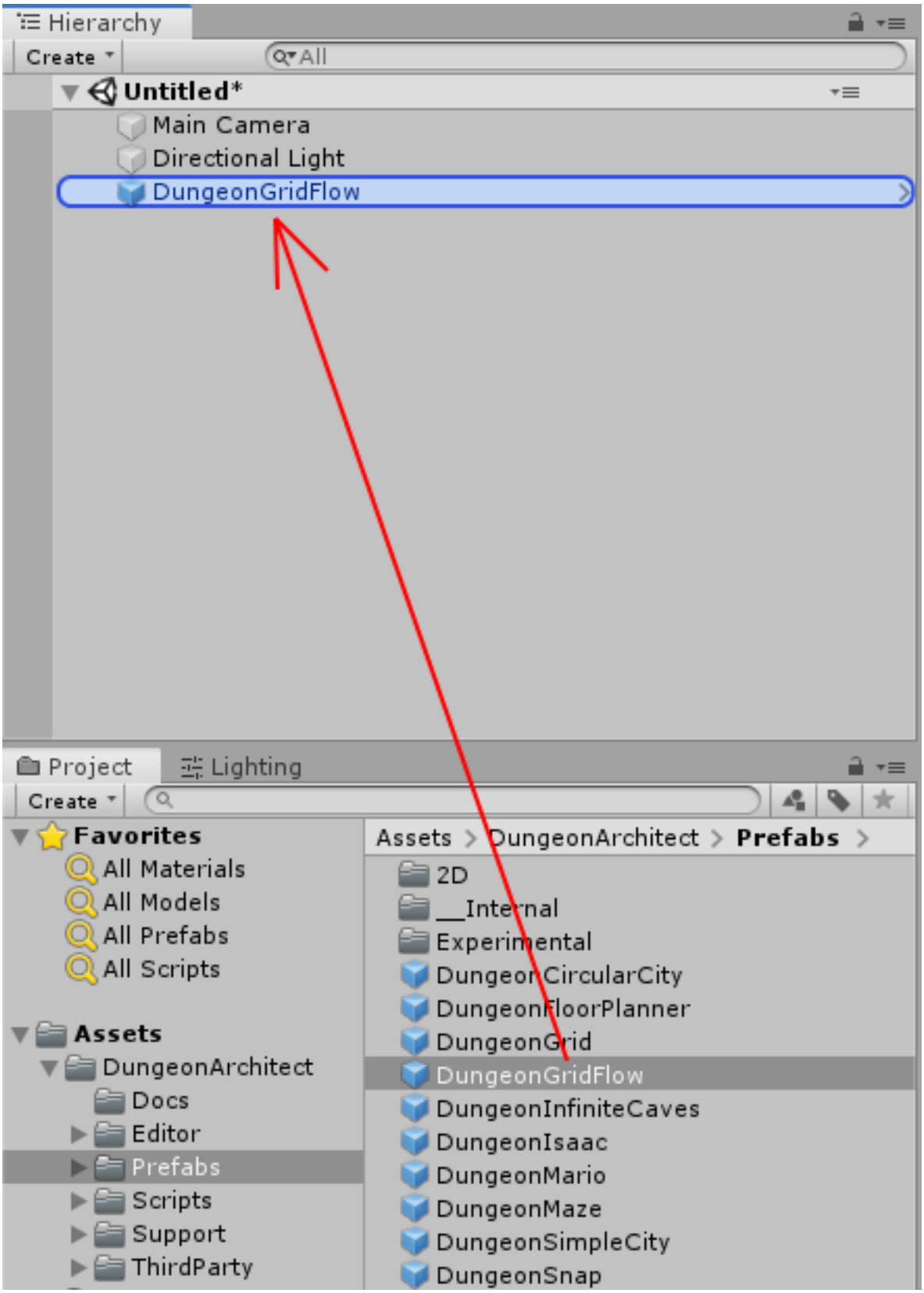
### 4.1.4 Setup GridFlow Graph

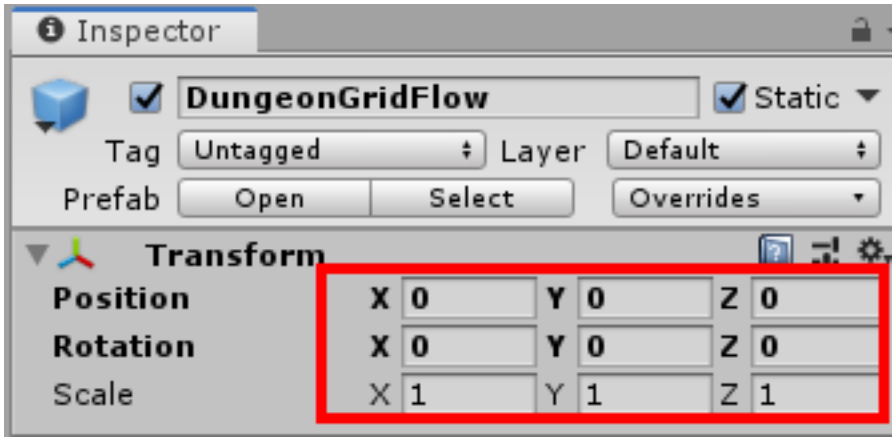
This builder requires another asset called the *Grid Flow Graph*. This is a graph that helps you control the flow of your dungeon. In this section, we'll use an existing graph from the samples folder

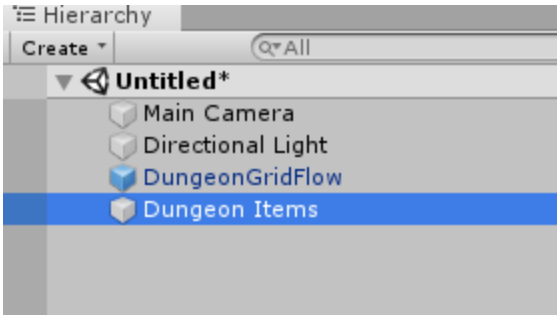
Navigate to `Assets\DungeonArchitect_Samples\DemoBuilder_GridFlow\FlowGraph` and assign the graph `DemoGridFlow` to the *DungeonGridFlow* game object's `Flow Asset` property

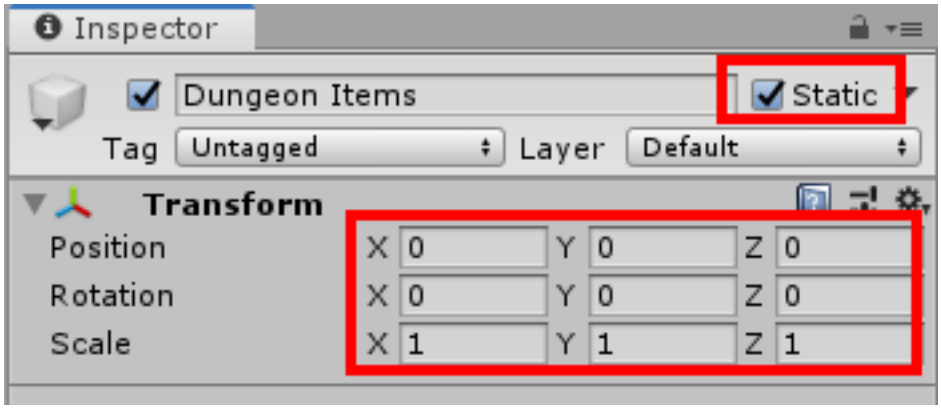
### 4.1.5 Build Dungeon

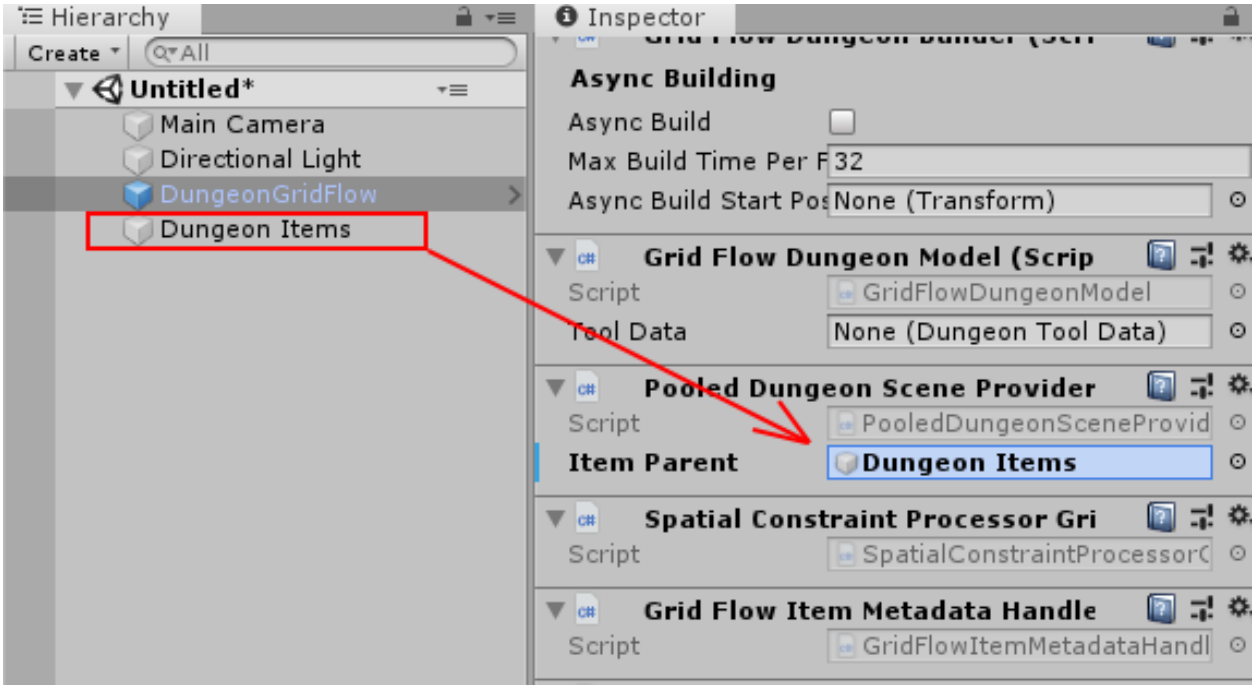
Select the `DungeonGridFlow` game object and click `Build Dungeon` button from the inspector window


Hierarchy
Create
All
Untitled*
Main Camera
Directional Light
DungeonGridFlow
Project
Lighting
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
DungeonArchitect
Docs
Editor
Prefabs
Scripts
Support
ThirdParty
Assets > DungeonArchitect > Prefabs
2D
__Internal
Experimental
DungeonCircularCity
DungeonFloorPlanner
DungeonGrid
DungeonGridFlow
DungeonInfiniteCaves
DungeonIsaac
DungeonMario
DungeonMaze
DungeonSimpleCity
DungeonSnap


Inspector
DungeonGridFlow
Static
Tag
Untagged
Layer
Default
Prefab
Open
Select
Overrides
Transform
Position
Rotation
Scale
X 0
Y 0
Z 0
X 0
Y 0
Z 0
X 1
Y 1
Z 1


Hierarchy
Create
All
Untitled*
Main Camera
Directional Light
DungeonGridFlow
Dungeon Items


Inspector
Dungeon Items
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
X 0
Y 0
Z 0
X 0
Y 0
Z 0
X 1
Y 1
Z 1




### 4.1.6 Open Grid Flow Editor

Let's open the GridFlow asset in the editor:

Navigate to `Assets\DungeonArchitect_Samples\DemoBuilder_GridFlow\FlowGraph` and double click on `DemoGridFlow`.

Dock the editor window so you see both the scene view and the grid flow editor

Click the Play button on the top left of the flow editor to build a new dungeon in the Grid Flow Editor

### 4.1.7 Link Editor with Dungeon

We are going to link up the dungeon we have on the scene (*DungeonGridFlow* game object) with the Grid Flow Editor so when we generate a new dungeon in the editor, it syncs up the dungeon on the scene

Click an empty space (grey area) in the Execution Graph

This will show up the execution graph properties in the Inspector window

Assign the dungeon game object by dragging the DungeonGridFlow over

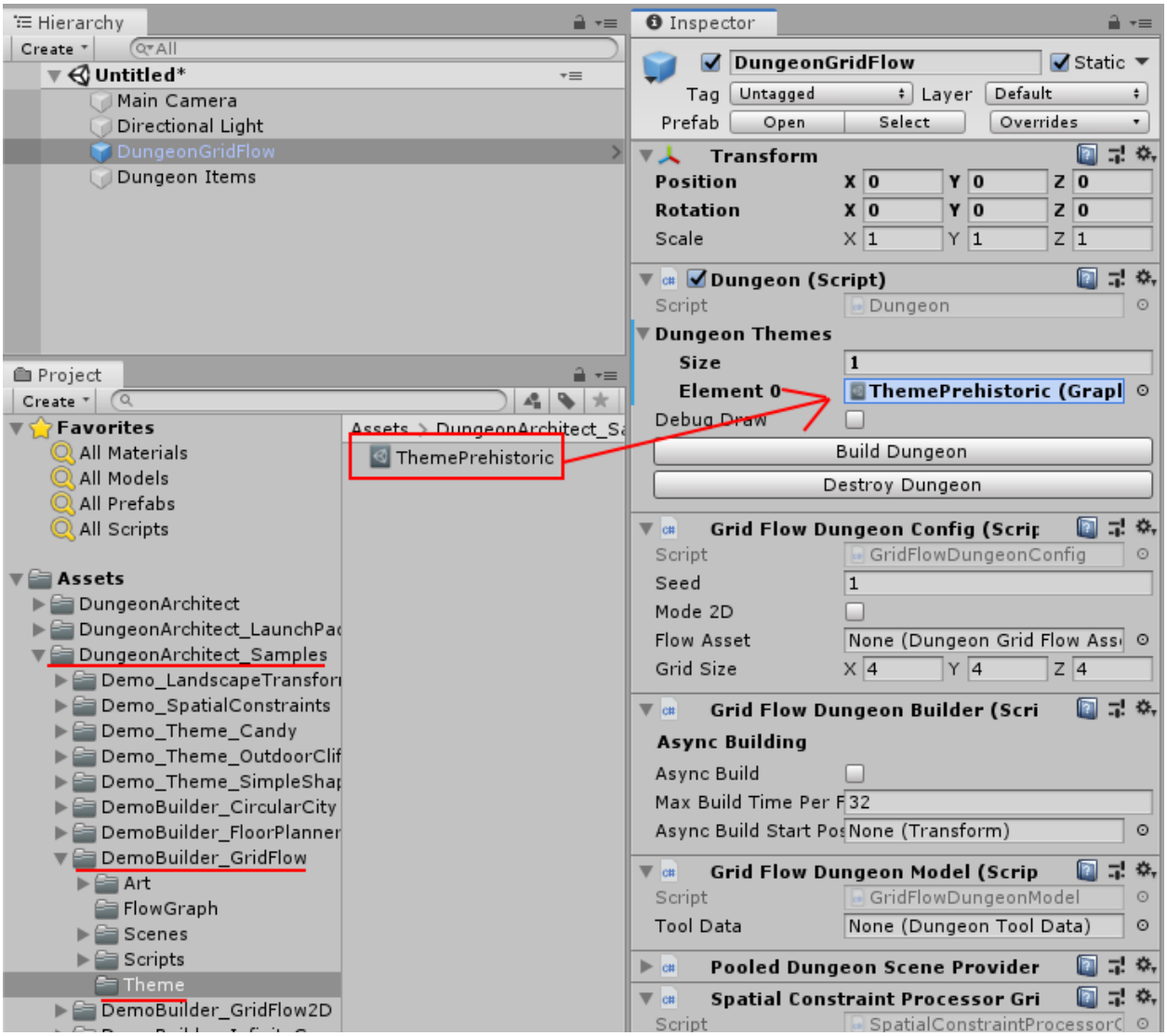
Now build the dungeon again by clicking the Play button in the Execution Graph Window

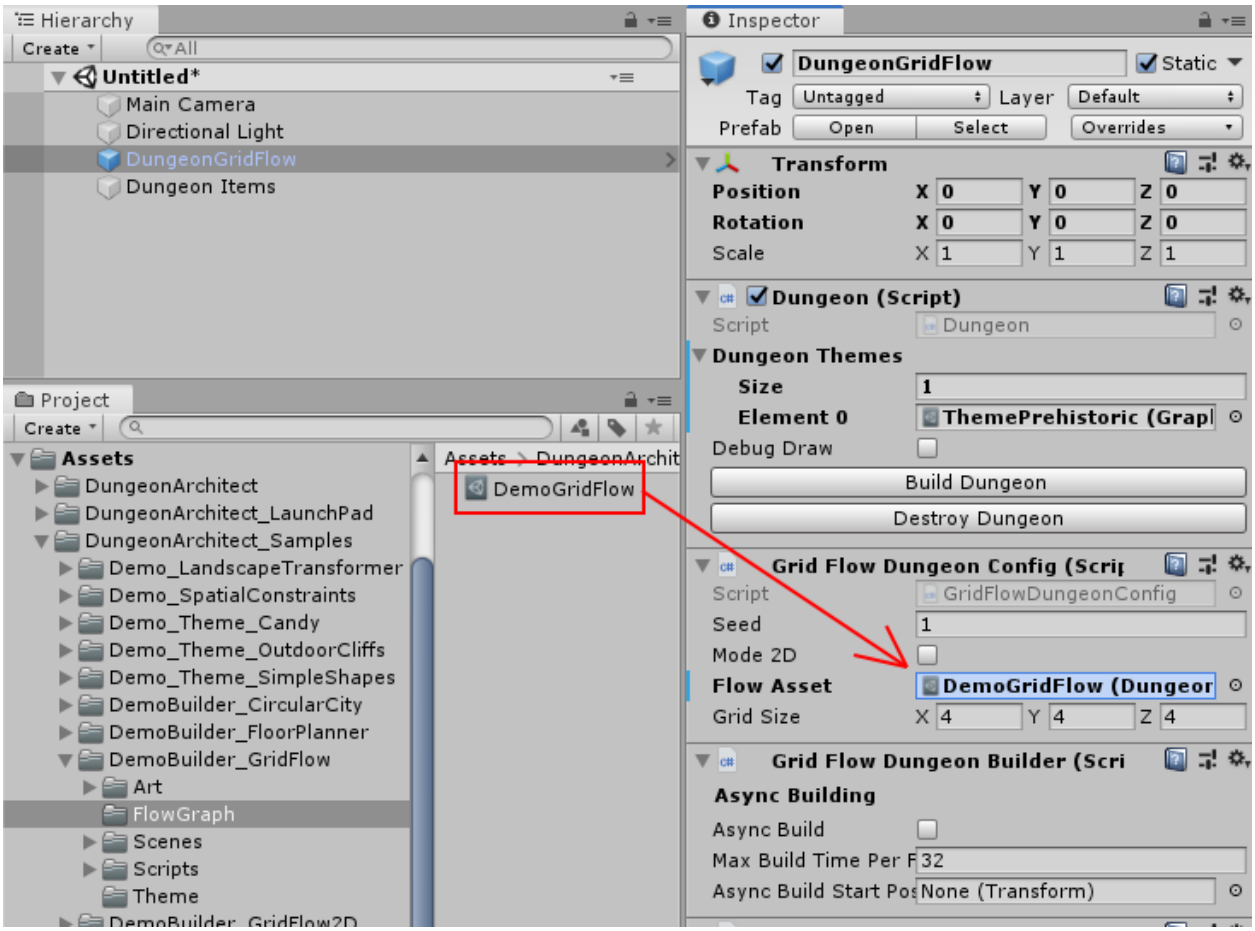
This will recreate the dungeon in the scene view. The dungeon in the editor window is now synchronized with the dungeon in the scene view

If you double click on any of the tiles in the Tilemap window, the scene view should focus on that tile/item

Double click on the Bonus Item (`B`) in the tilemap.

The scene view should zoom in on the treasure chest


Hierarchy
Create
All
Untitled*
Main Camera
Directional Light
DungeonGridFlow
Dungeon Items
Project
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
DungeonArchitect
DungeonArchitect_LaunchPa
DungeonArchitect_Samples
Demo_LandscapeTransfor
Demo_SpatialConstraints
Demo_Theme_Candy
Demo_Theme_OutdoorClif
Demo_Theme_SimpleShap
DemoBuilder_CircularCity
DemoBuilder_FloorPlanner
DemoBuilder_GridFlow
Art
FlowGraph
Scenes
Scripts
Theme
DemoBuilder_GridFlow2D
Assets > DungeonArchitect_Sa
ThemePrehistoric
Inspector
DungeonGridFlow
Static
Tag
Untagged
Layer
Default
Prefab
Open
Select
Overrides
Transform
Position
Rotation
Scale
Dungeon (Script)
Script
Dungeon
Dungeon Themes
Size
1
Element 0
ThemePrehistoric (Grapl
Debug Draw
Build Dungeon
Destroy Dungeon
Grid Flow Dungeon Config (Scrip
Script
GridFlowDungeonConfig
Seed
1
Mode 2D
Flow Asset
None (Dungeon Grid Flow Ass
Grid Size
Grid Flow Dungeon Builder (Scri
Async Building
Async Build
Max Build Time Per F
32
Async Build Start Pos
None (Transform)
Grid Flow Dungeon Model (Scrip
Script
GridFlowDungeonModel
Tool Data
None (Dungeon Tool Data)
Pooled Dungeon Scene Provider
Spatial Constraint Processor Gri
Script
SpatialConstraintProcessorC



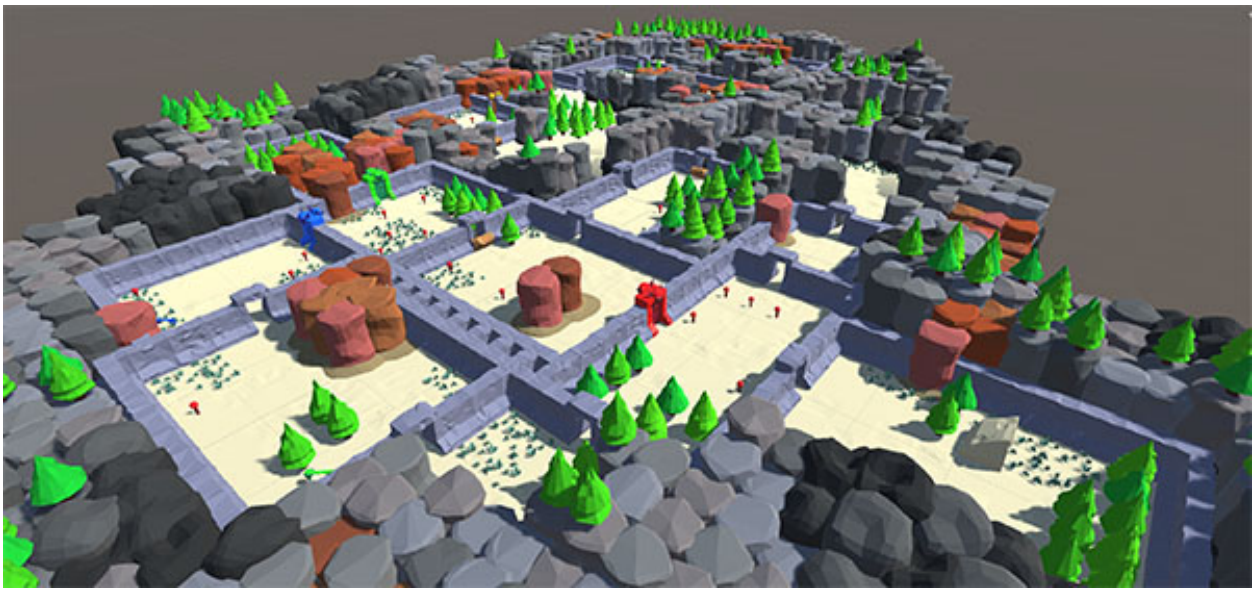



Fig. 1: GridFlow dungeon built using the Prehistoric theme

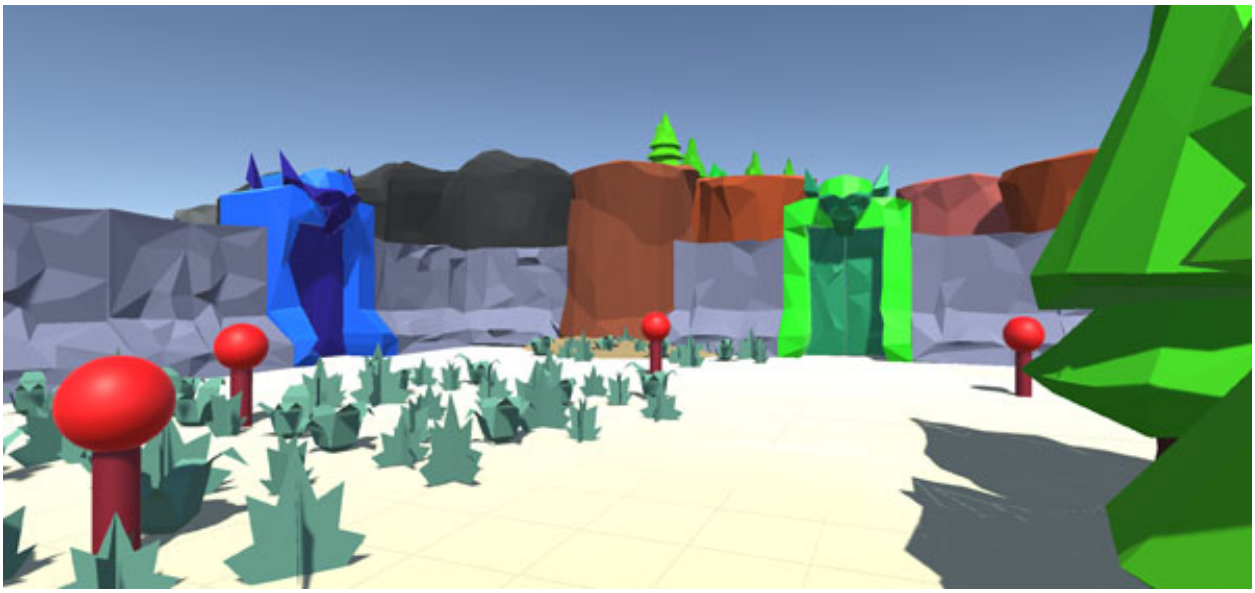
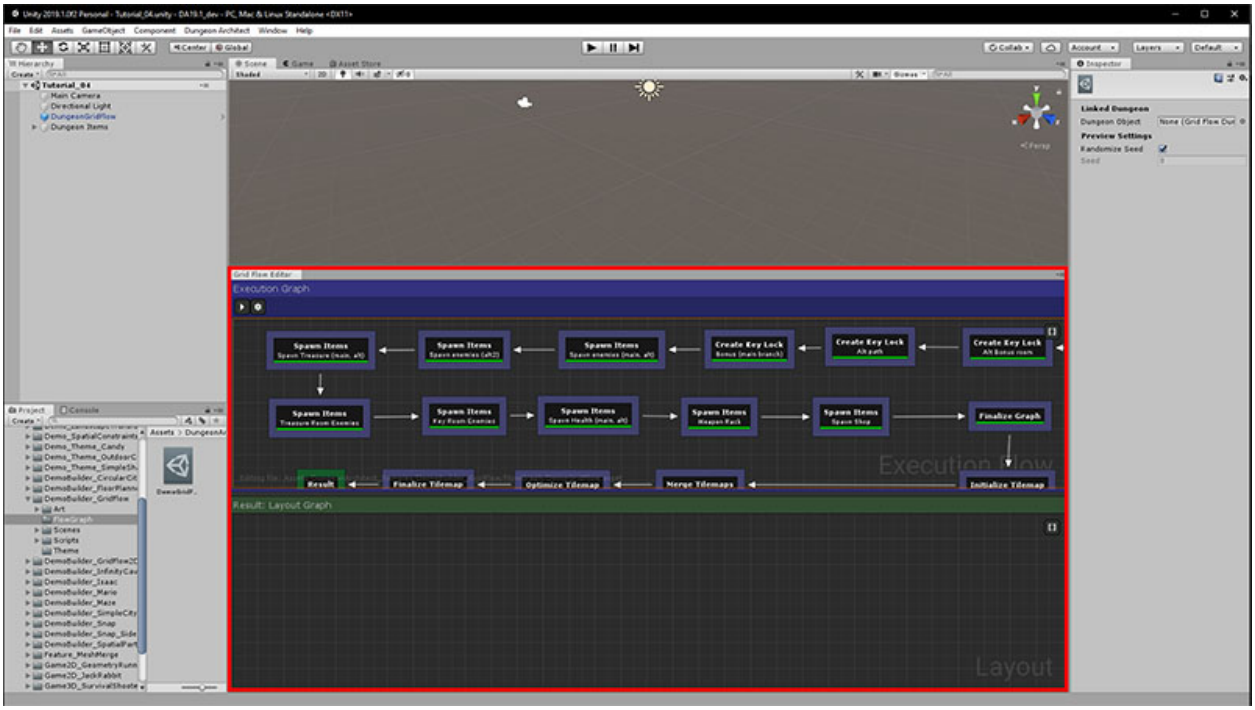
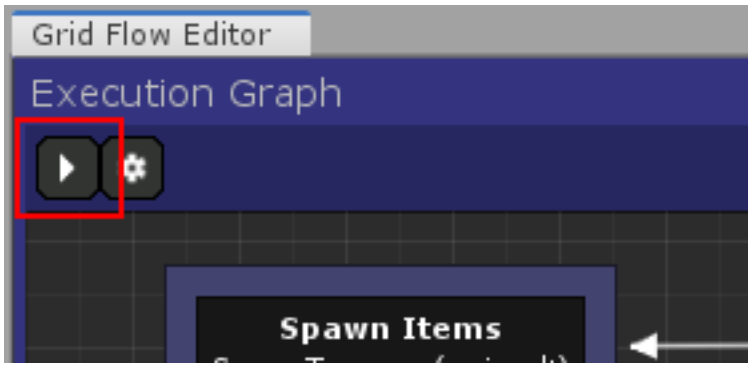


Fig. 2: GridFlow dungeons support key-locks





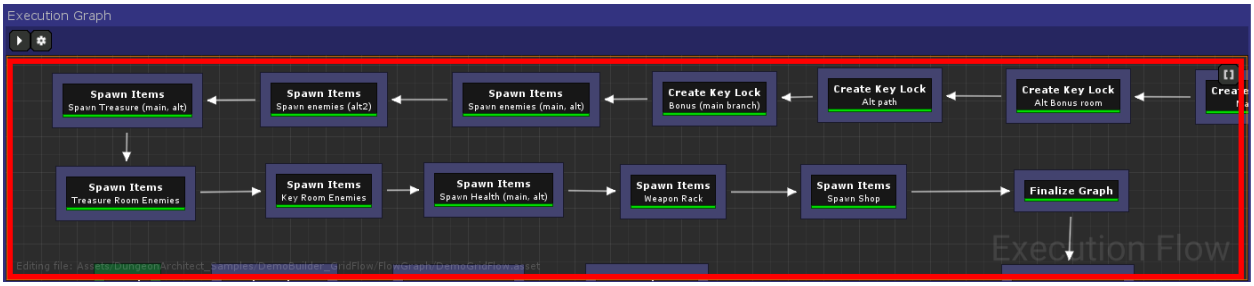




Fig. 3: Click anywhere in the empty area

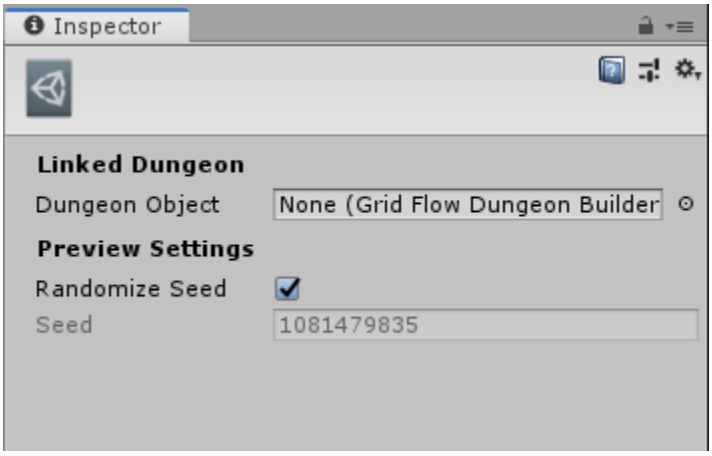
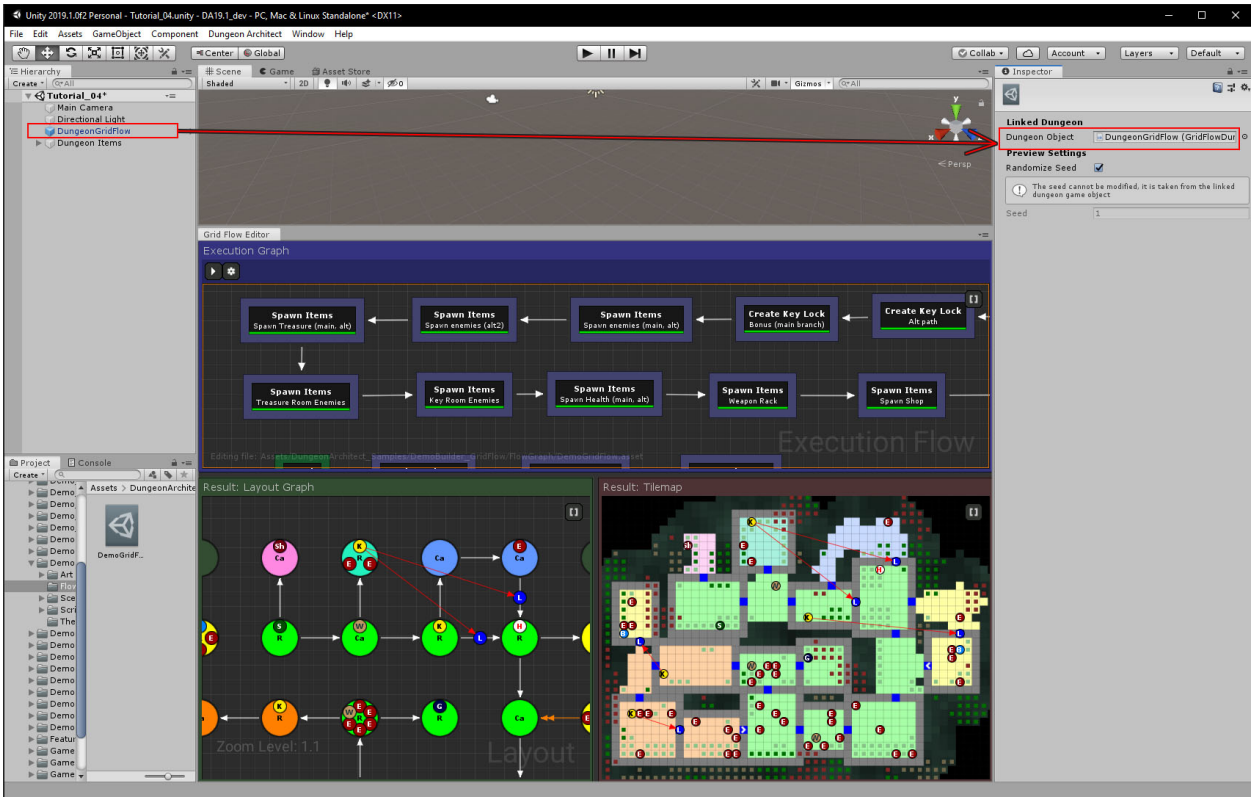
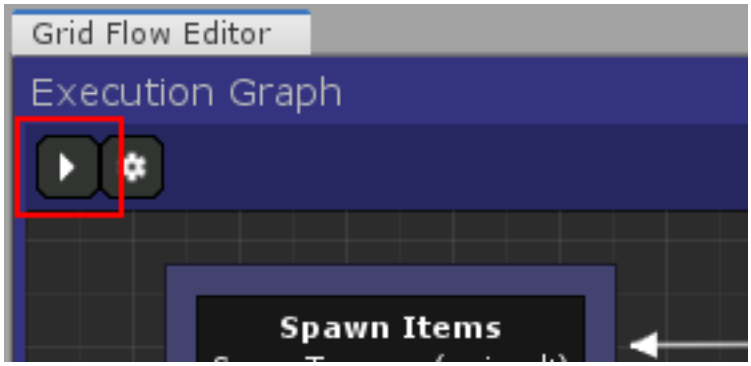
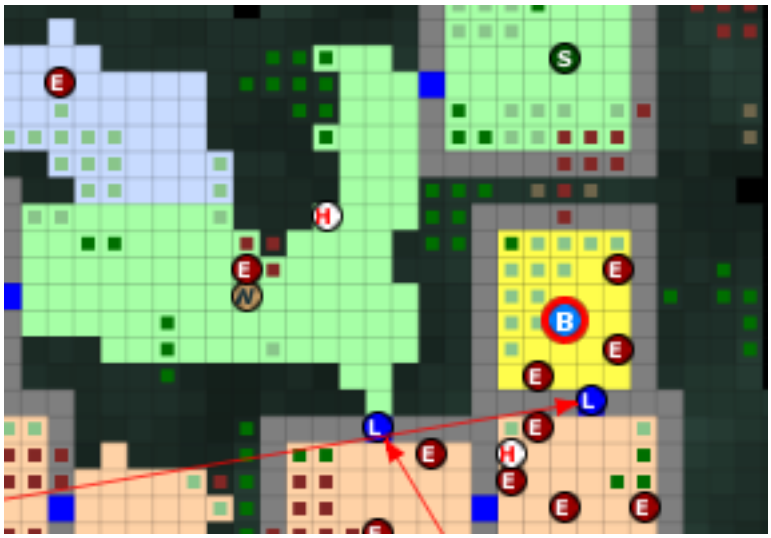
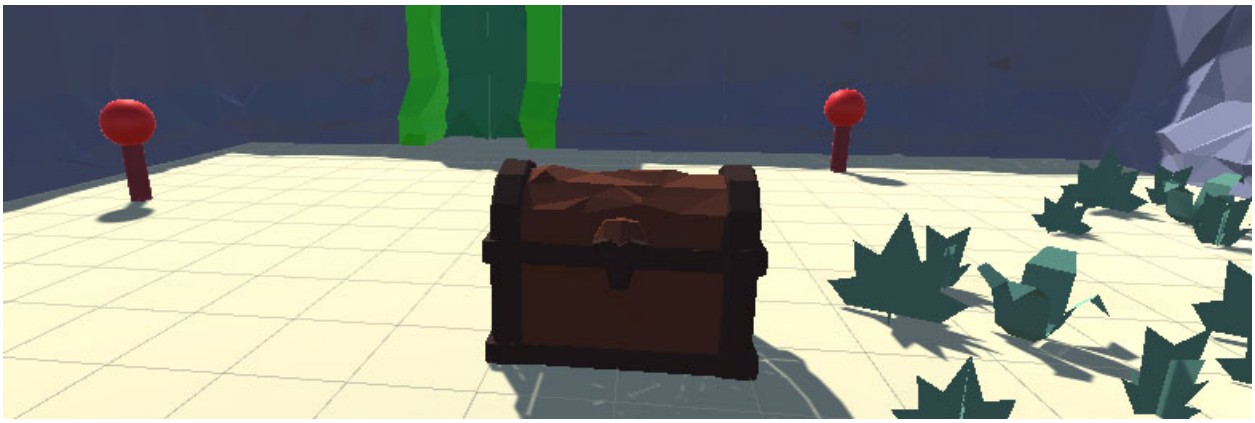



Fig. 4: Execution Graph properties


Grid Flow Editor
Execution Graph
Spawn Items
Create Key Lock
Execution Flow
Result: Layout Graph
Result: Tilemap
Zoom Level: 1.1
Layout


Grid Flow Editor
Execution Graph
Spawn Items





### 4.1.8 Explore Grid Flow Graph

After you've built a dungeon in the editor (by hitting the play button on the top left), you can select each node and see how the dungeon layout was built, as shown in the lower preview panels

Fig. 5: Select a node to preview the build process

## 4.2 Design a GridFlow Graph

In the previous section, we used an existing Grid Flow graph. In this section, we'll design one ourselves.

### 4.2.1 Setup

Destroy the existing dungeon and clear out the Flow Asset that we assigned earlier

Create a new Grid Flow asset from either the Create menu or the Main Menu

Rename to something appropriate and double click the grid flow asset to open it in the editor

We won't be needing the scene view for some time. Dock the editor so we have more working area

Notice that there is only one node in the Execution graph, the `Result` node. Our final output should be connected to this node

### 4.2.2 Create Grid

Right click on an empty area in the Execution Graph and from the context menu select `Layout Graph > Create Grid`

Connect this node to the `Result` node and hit play

This node creates an initial grid to work with

In this builder, you first design your level in an abstract layout graph like this and then move the final result to a tilemap

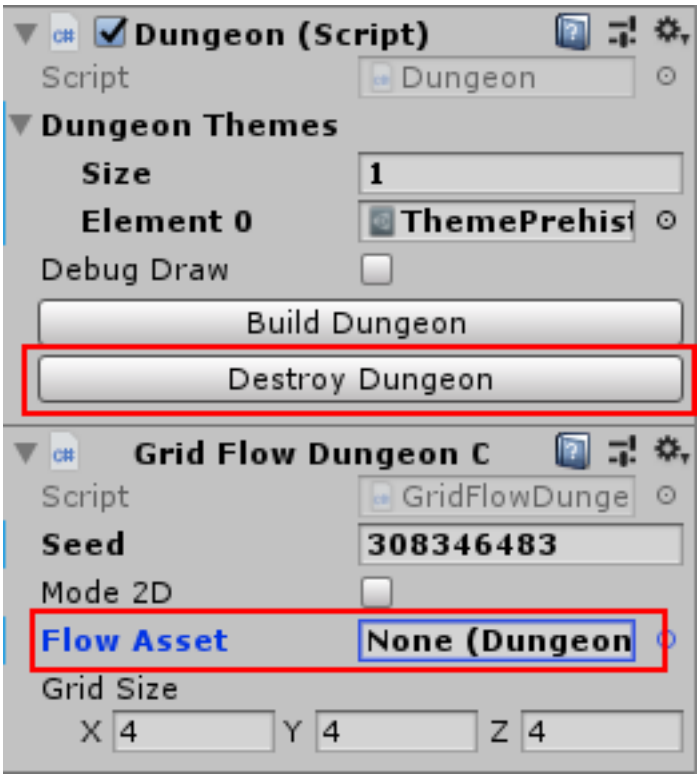



Fig. 6: DungeonGridFlow game object properties

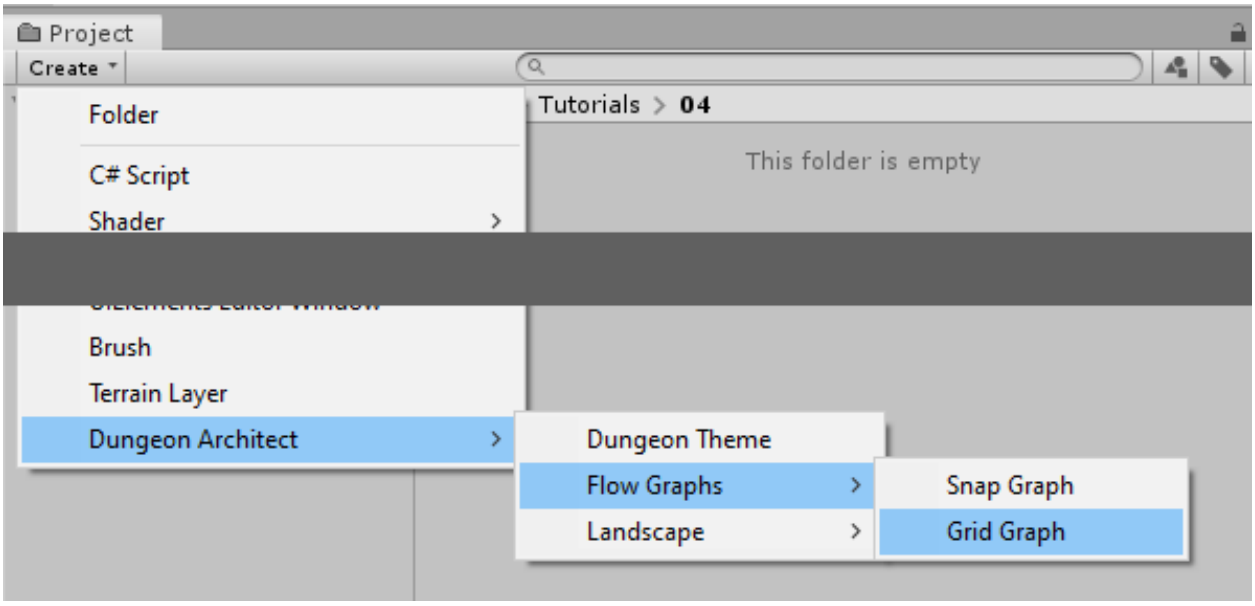



Fig. 7: Create Menu

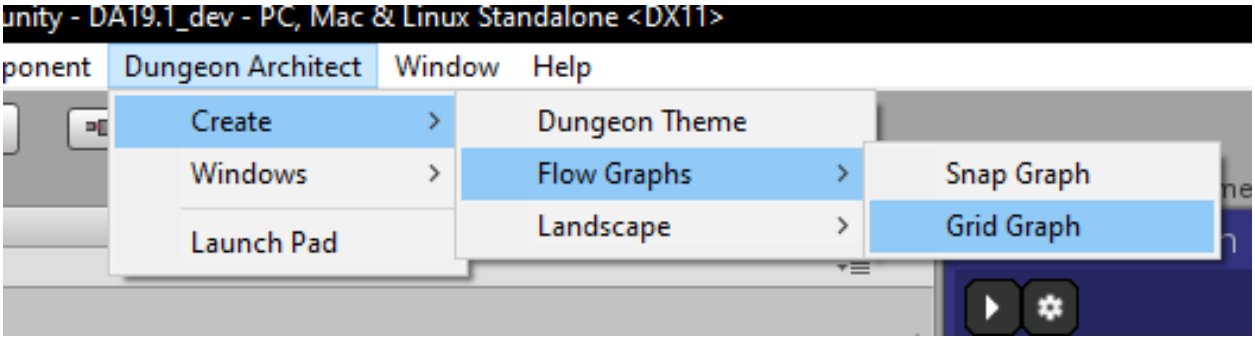
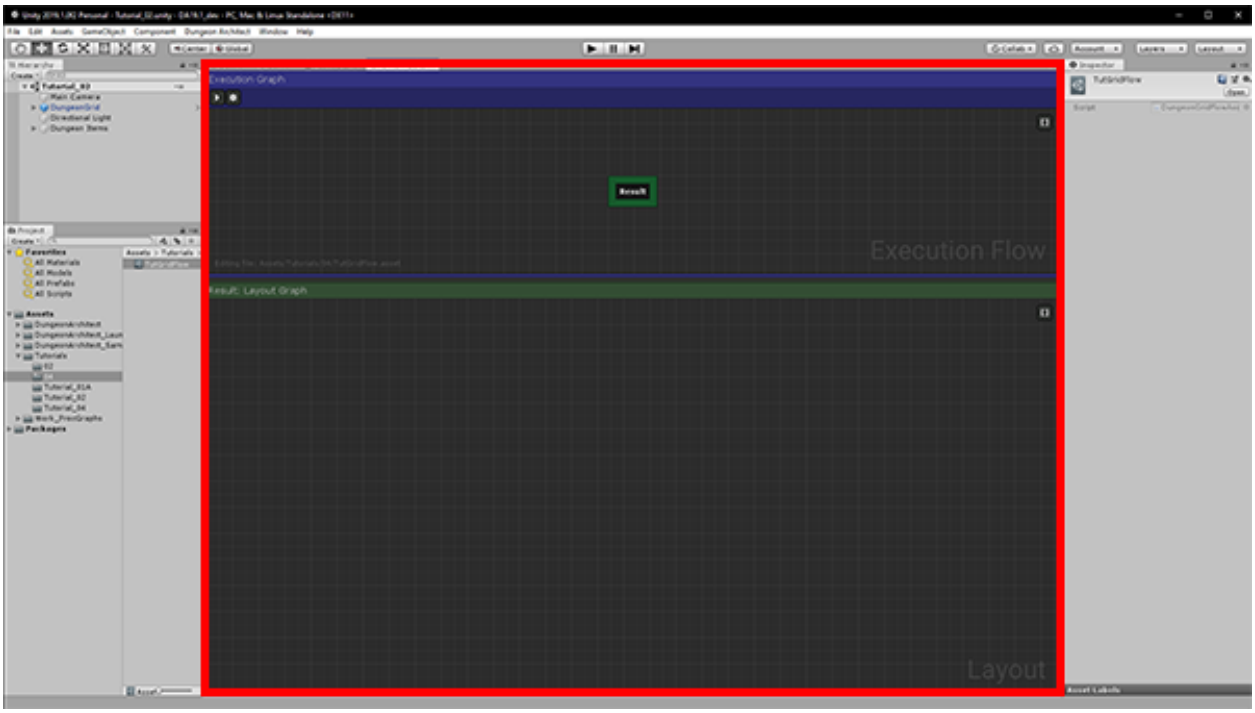
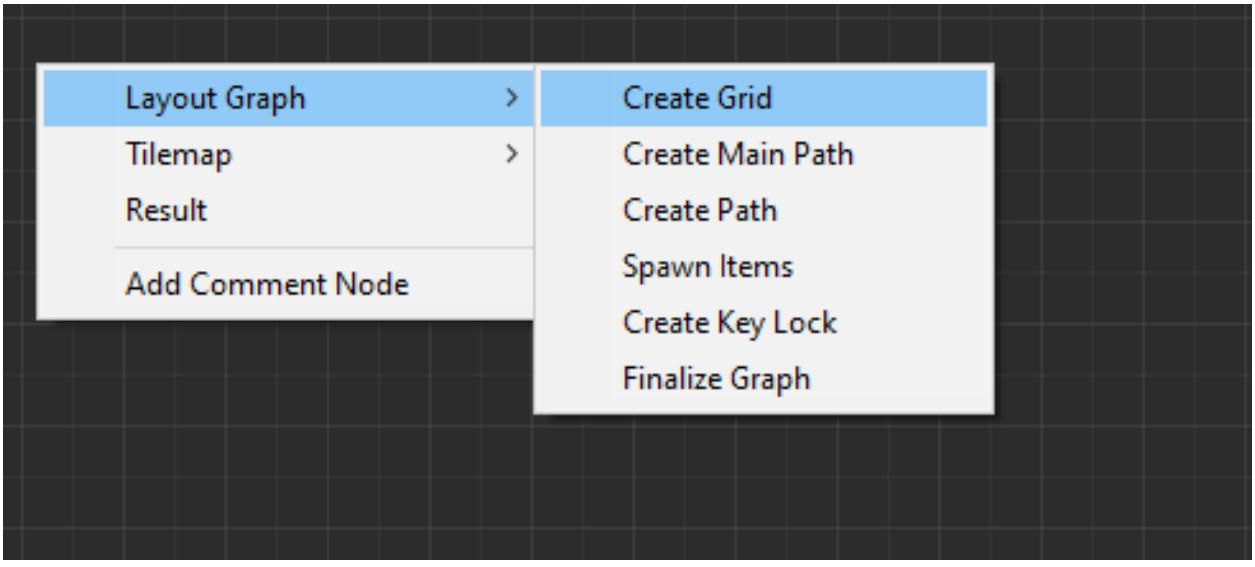
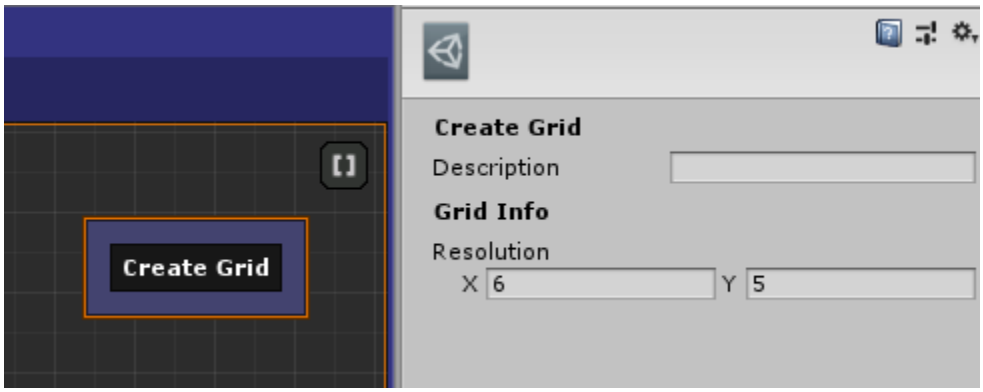
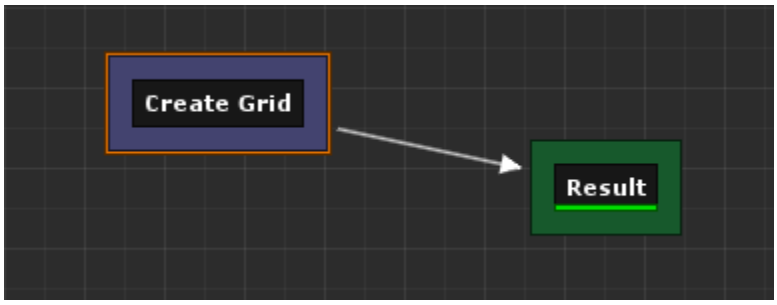


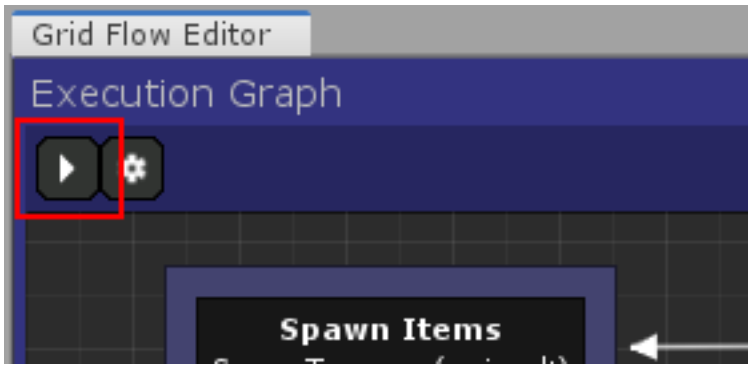

Fig. 8: Main Menu

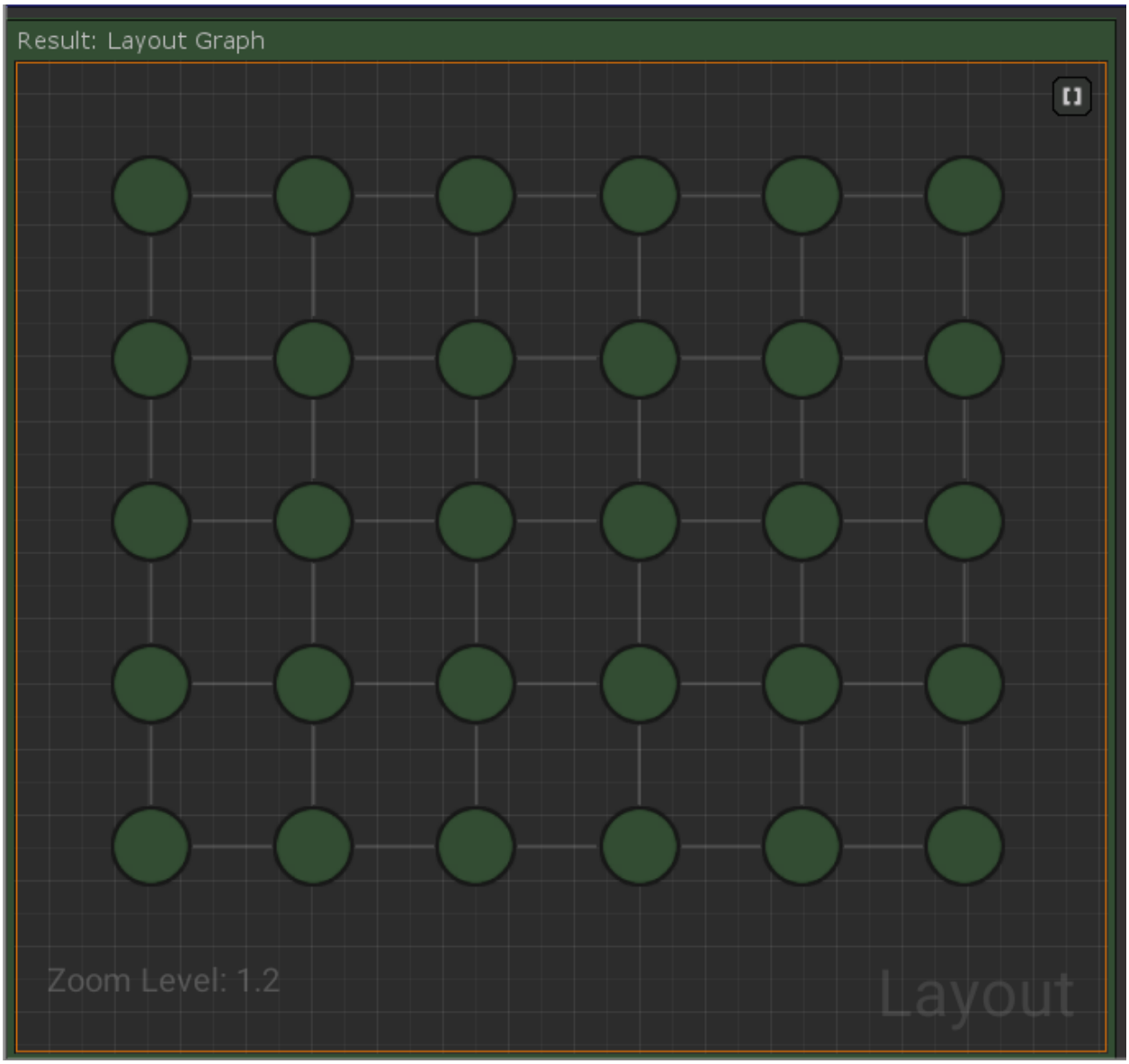



Create Grid
Create Grid
Description
Grid Info
Resolution
X 6
Y 5


Create Grid
Result

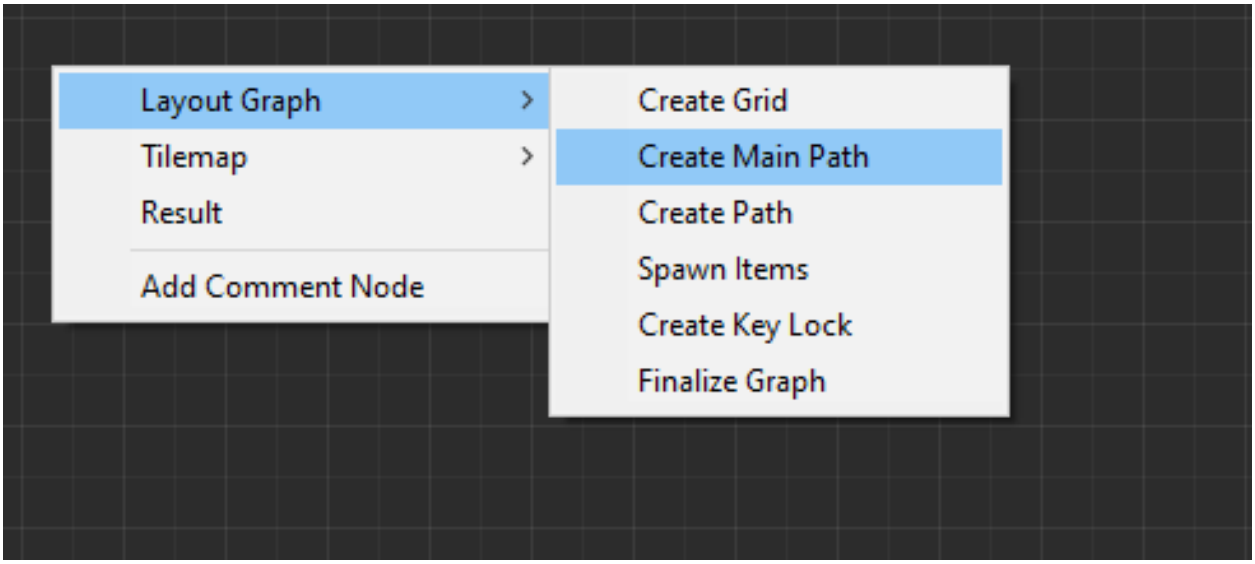

Grid Flow Editor
Execution Graph
Spawn Items


Result: Layout Graph
Zoom Level: 1.2
Layout

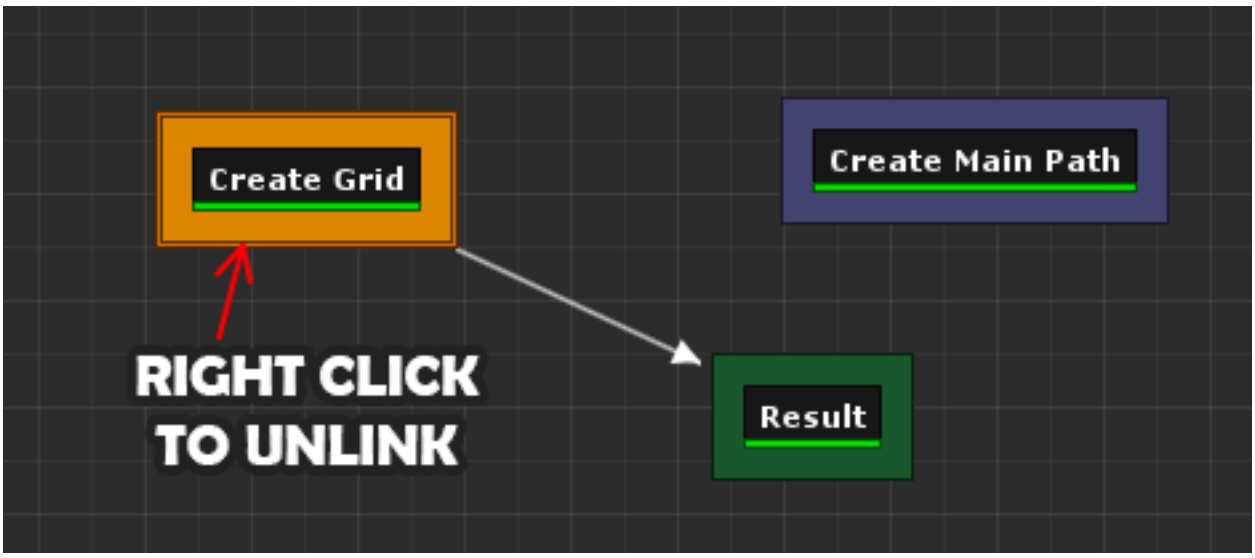
### 4.2.3 Create Main Path

We'll next create a main path within this grid. The main path has a spawn point and goal

Create a new node `Layout Graph > Create Math Path`




Unlink the `Create Grid` node from the `Result` node (do this by right clicking on the node's orange border)




Link the nodes up like below and hit Play

This node creates a main path in the grid. Keep hitting the play button for different result

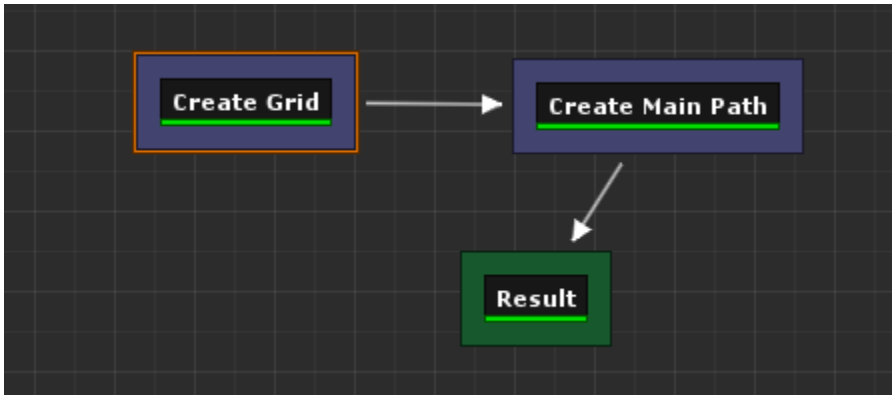
---

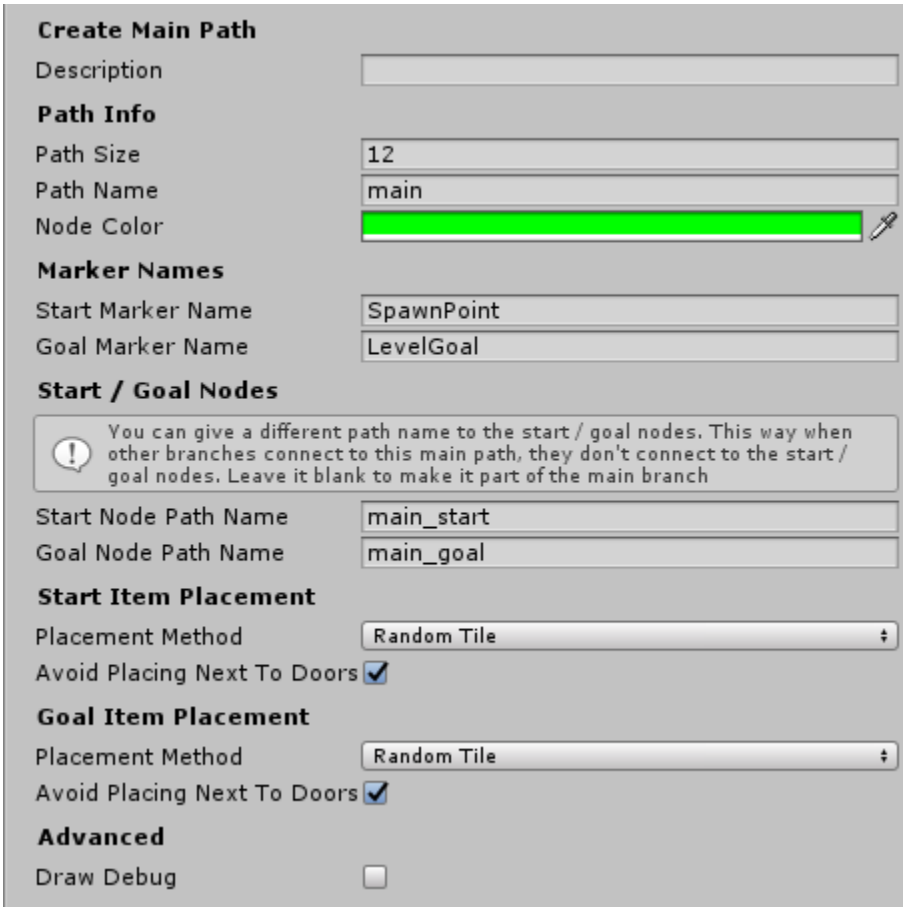
**Note:** If you do not see random results when you hit play, make sure randomize is enabled. Enable this by clicking on an empty area in the Execution Graph to show the properties. In the inspector, select `Randomize Seed`

---

Select the `Create Main Path` node and inspect the properties

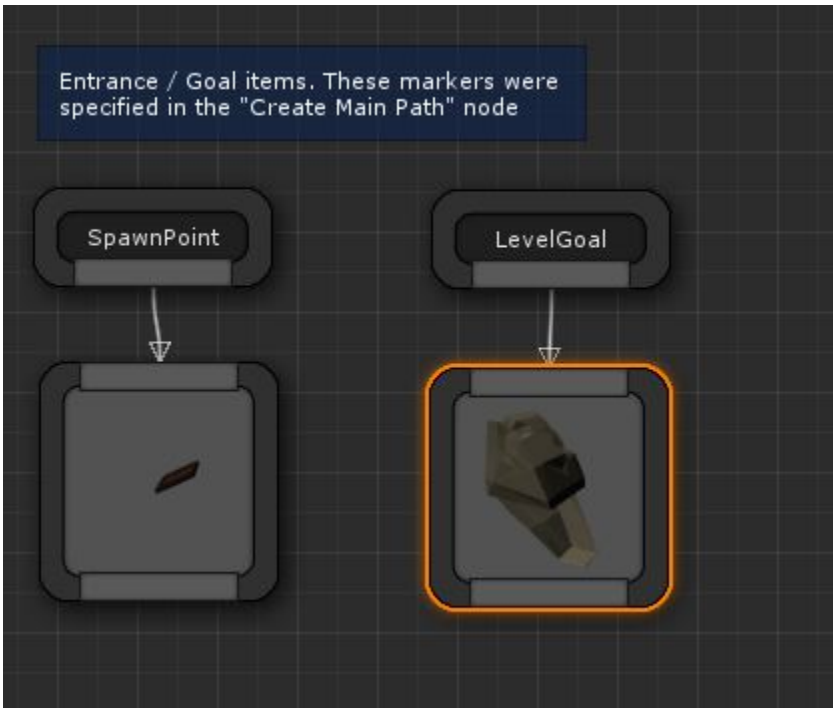
We'll leave everything to default for now


Create Grid
Create Main Path
Result


Create Main Path
Description
Path Info
Path Size
12
Path Name
main
Node Color
Marker Names
Start Marker Name
SpawnPoint
Goal Marker Name
LevelGoal
Start / Goal Nodes
You can give a different path name to the start / goal nodes. This way when other branches connect to this main path, they don't connect to the start / goal nodes. Leave it blank to make it part of the main branch
Start Node Path Name
main_start
Goal Node Path Name
main_goal
Start Item Placement
Placement Method
Random Tile
Avoid Placing Next To Doors
Goal Item Placement
Placement Method
Random Tile
Avoid Placing Next To Doors
Advanced
Draw Debug

Notice the *Path Name* parameter is set to `main` This is the name of the path and we will be referencing this path in the future nodes with this name

You can adjust the size of the path. `Start Marker Name` and `Goal Marker Name` lets you specify a name for the markers. You can then create these markers in the theme file and add any object you like. In the *Prehistoric* theme, there's a marker already created with these names and a player controller is placed under `SpawnPoint` marker and a level goal handler prefab is placed under `LevelGoal` marker




### 4.2.4 Create Alternate Path

We'll next create an alternate path pathing off the main path so the player has another way of reaching the goal

Create a new node `Layout Graph > Create Path`

Connect the nodes together like below

Leave all the properties as default and hit play

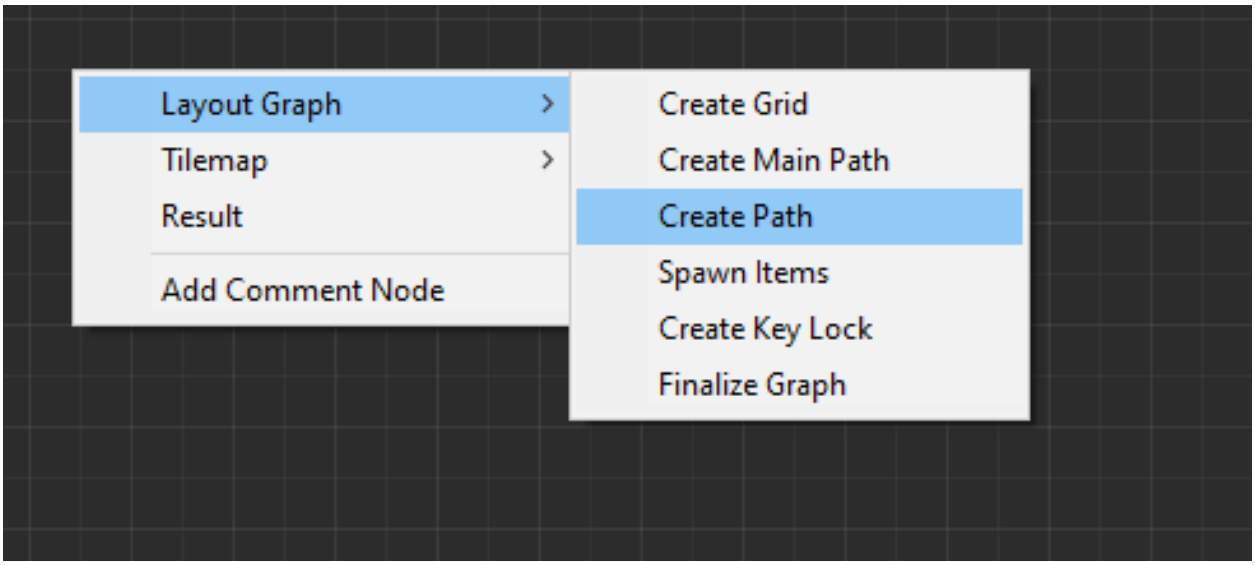
Select the `Create Path` node and inspect the properties

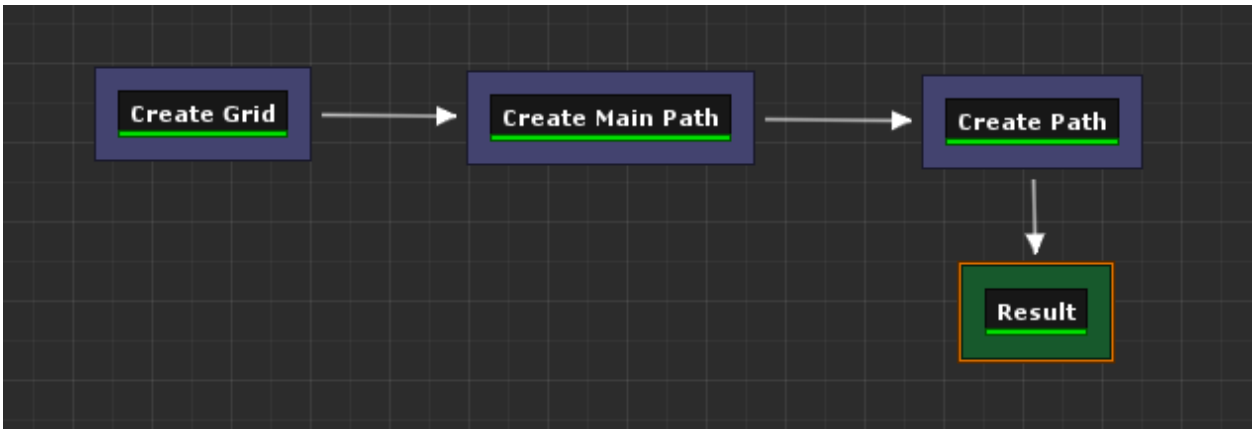
Change the *Path Name* from `path` to `alt`. We will be referencing this path as `alt` in the future

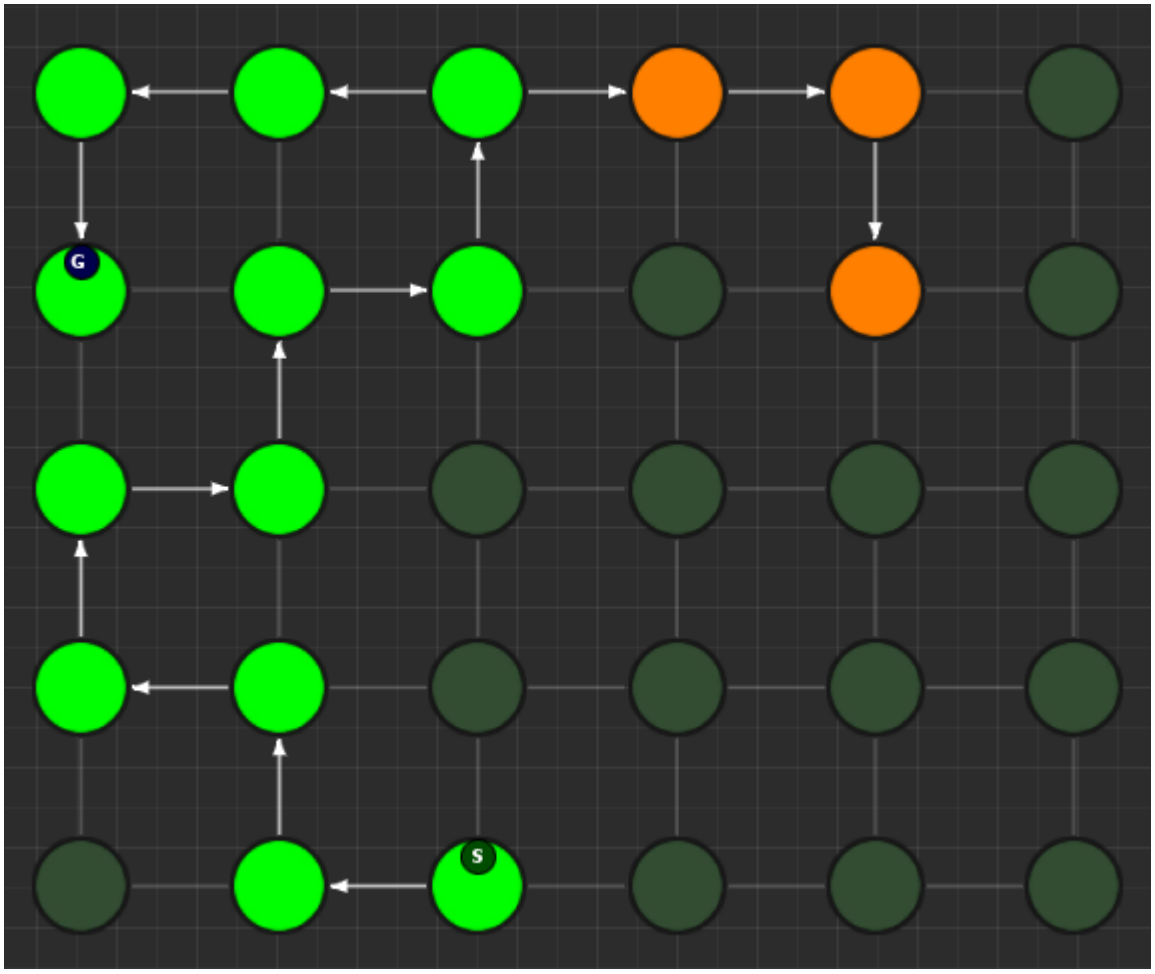
You can specify the paths from which this path should start and end. The *Start From Path* parameter is set to `main`, referencing the main path we created in the previous section

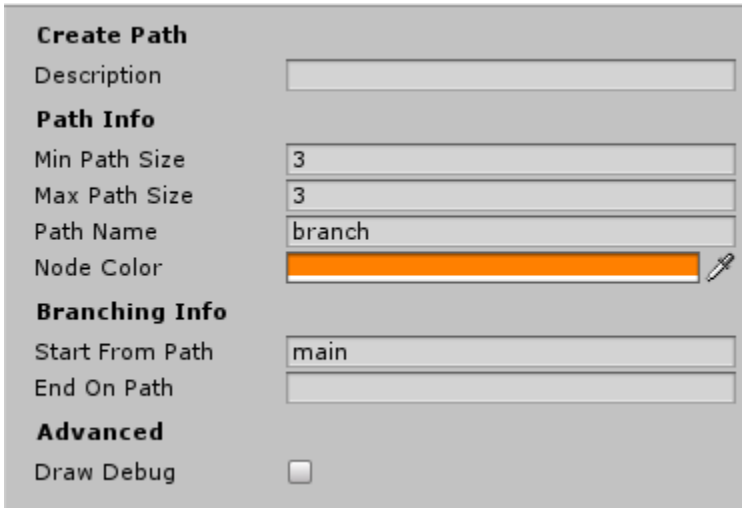
The *End On Path* is left empty, so the end of this path doesn't connect back to anything. We'd like this path to connect back to the main path.

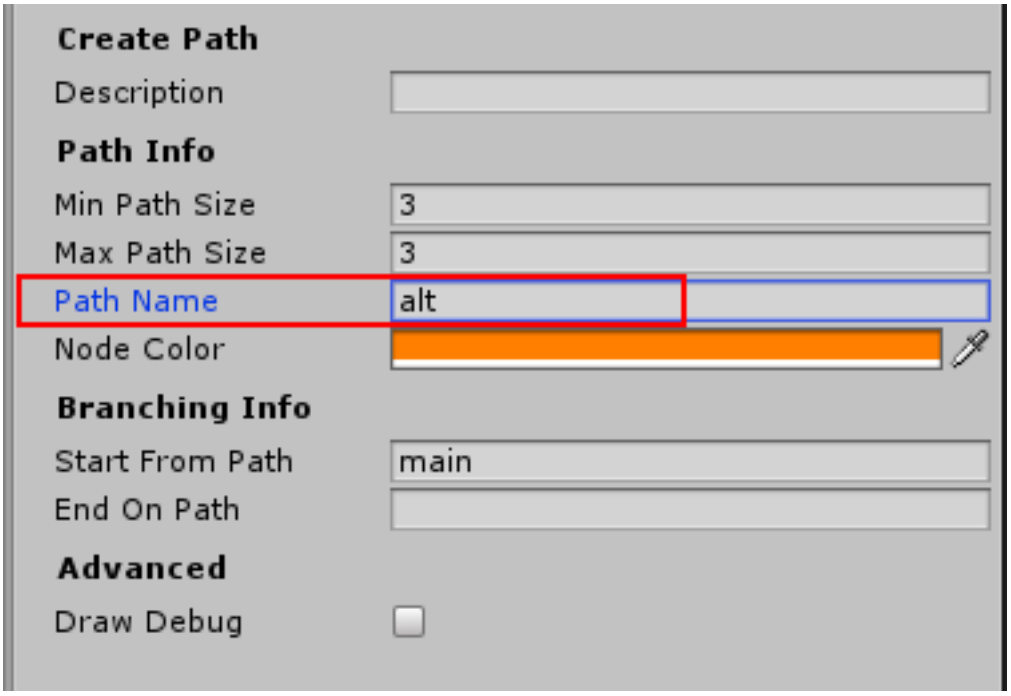
Set the *End On Path* parameter to `main`

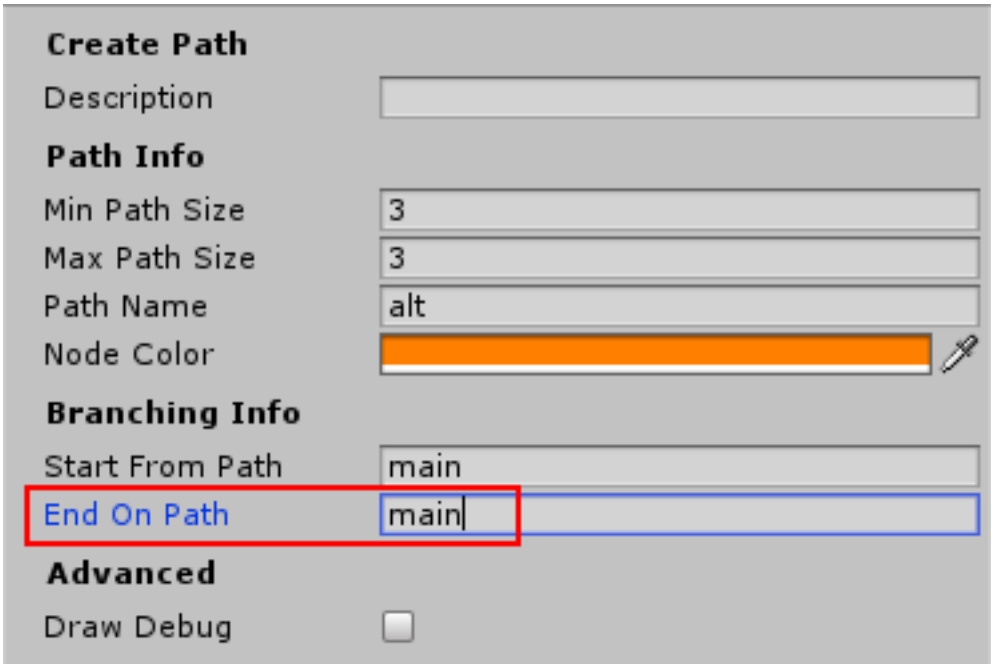

Layout Graph
Tilemap
Result
Add Comment Node
Create Grid
Create Main Path
Create Path
Spawn Items
Create Key Lock
Finalize Graph


Create Grid
Create Main Path
Create Path
Result


G
S

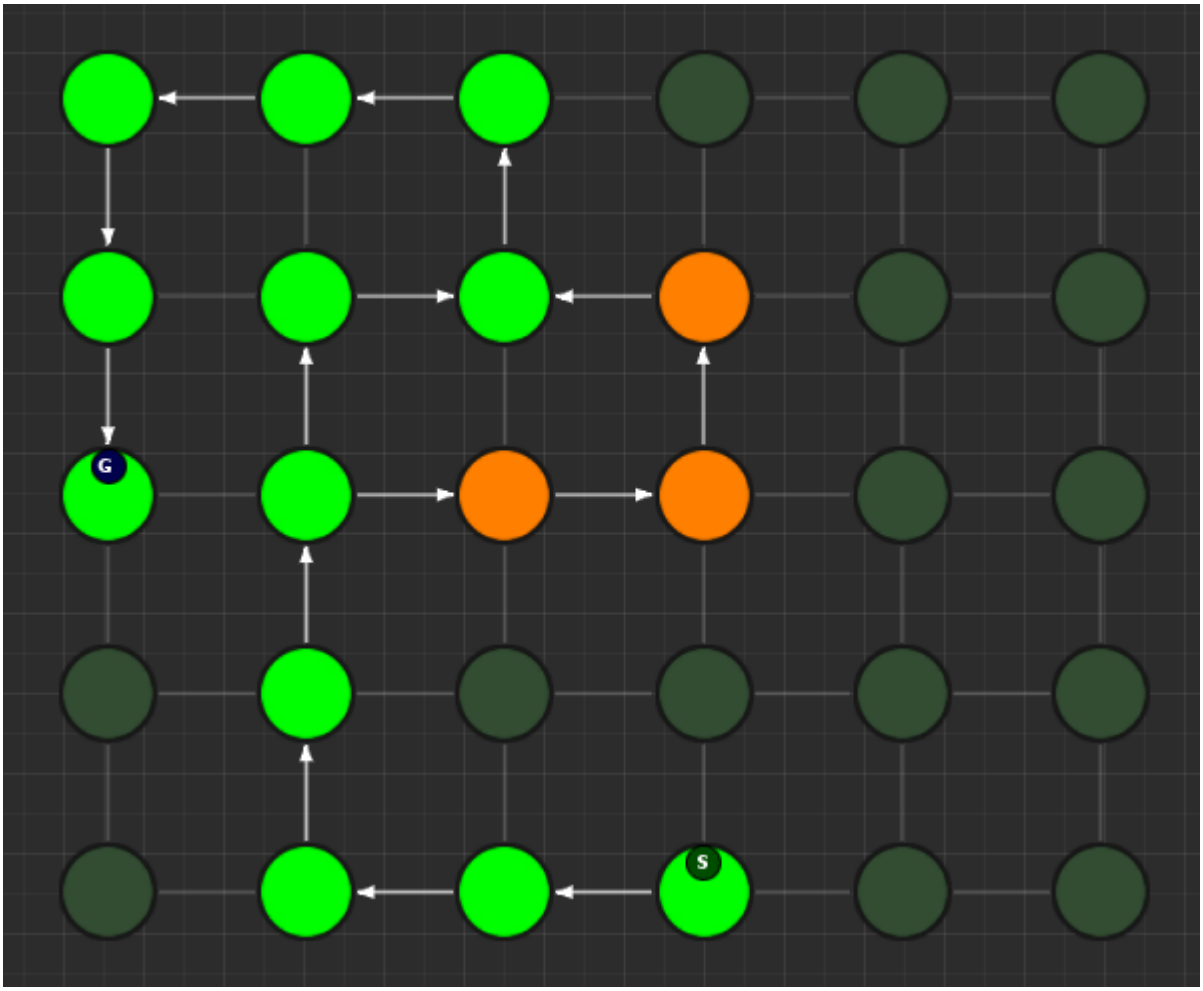

Create Path
Description
Path Info
Min Path Size
3
Max Path Size
3
Path Name
branch
Node Color
Branching Info
Start From Path
main
End On Path
Advanced
Draw Debug


Create Path
Description
Path Info
Min Path Size
3
Max Path Size
3
Path Name
alt
Node Color
Branching Info
Start From Path
main
End On Path
Advanced
Draw Debug


Create Path
Description
Path Info
Min Path Size
3
Max Path Size
3
Path Name
alt
Node Color
Branching Info
Start From Path
main
End On Path
main
Advanced
Draw Debug

| | |
|---|---|
| **Min Path Size** | 3 |
| **Max Path Size** | 3 |
| **Path Name** | alt |
| **Node Color** | orange |
| **Start From Path** | main |
| **End On Path** | main |

This will make the alternate path (orange) connect back to the main path (green)

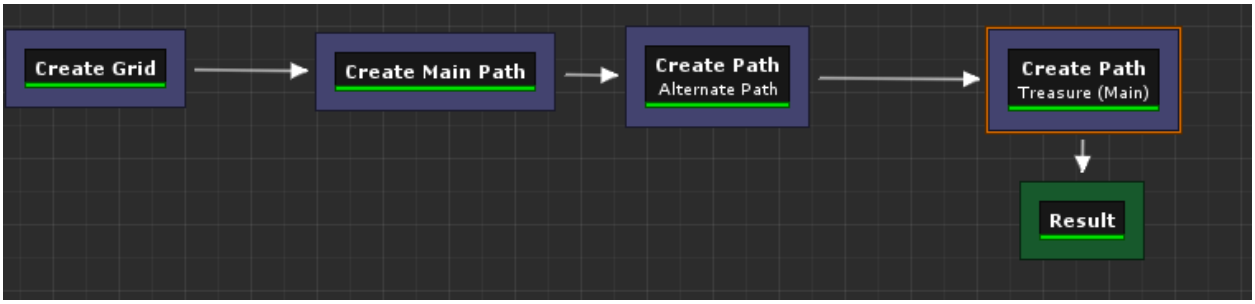


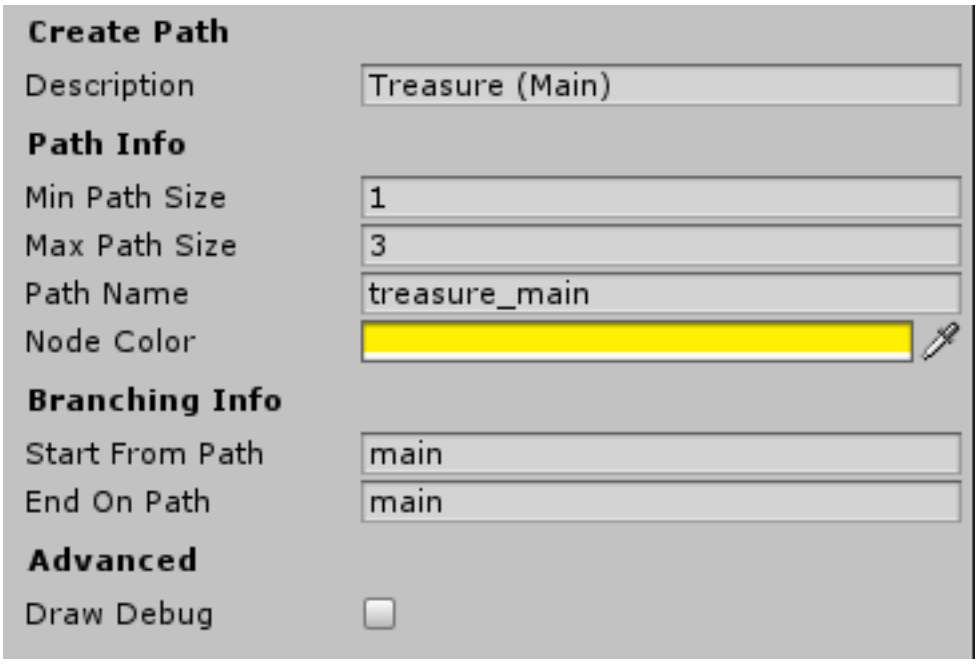

Keep hitting Play for different results

### 4.2.5 Create Treasure Room (Main)

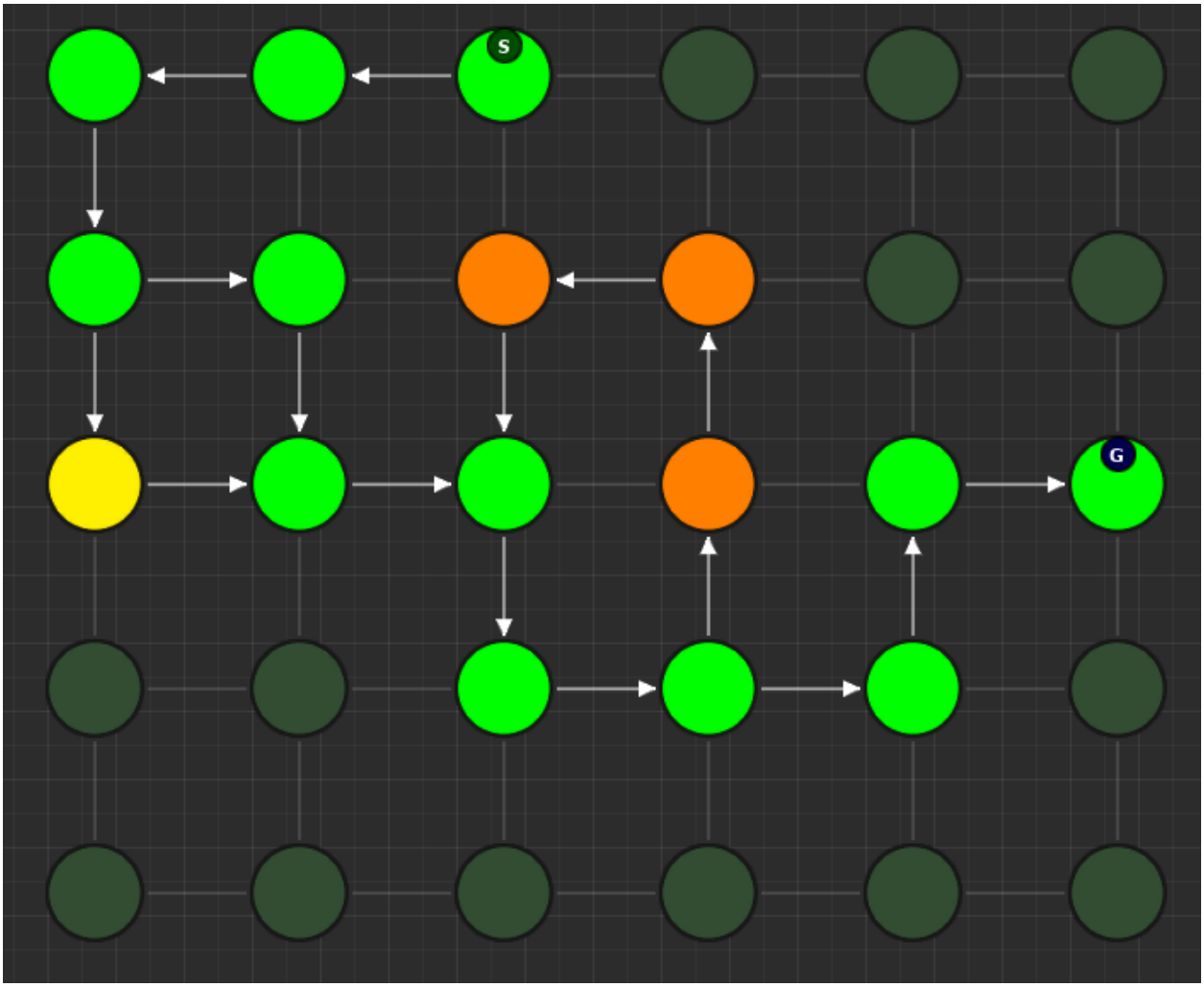
We'll add a treasure room connected to the main path

Add a new node `Layout Graph > Create Path` and set it up as follows:


Create Grid
Create Main Path
Create Path
Alternate Path
Create Path
Treasure (Main)
Result


Create Path
Description
Treasure (Main)
Path Info
Min Path Size
1
Max Path Size
3
Path Name
treasure_main
Node Color
Branching Info
Start From Path
main
End On Path
main
Advanced
Draw Debug

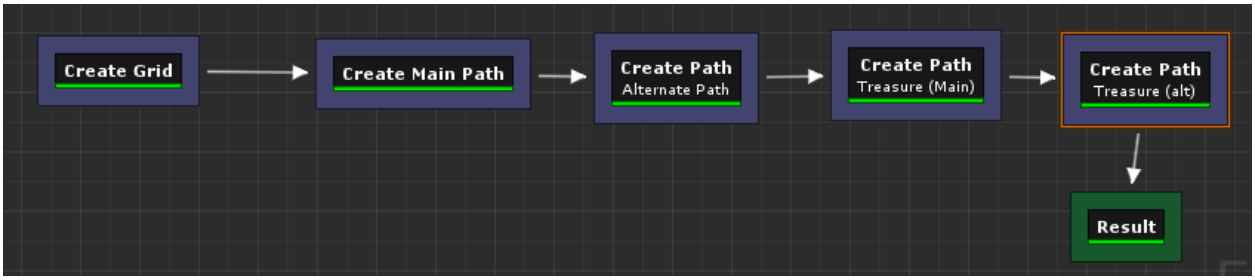
| | |
|---|---|
| **Min Path Size** | 1 |
| **Max Path Size** | 3 |
| **Path Name** | treasure_main |
| **Node Color** | yellow |
| **Start From Path** | main |
| **End On Path** | main |

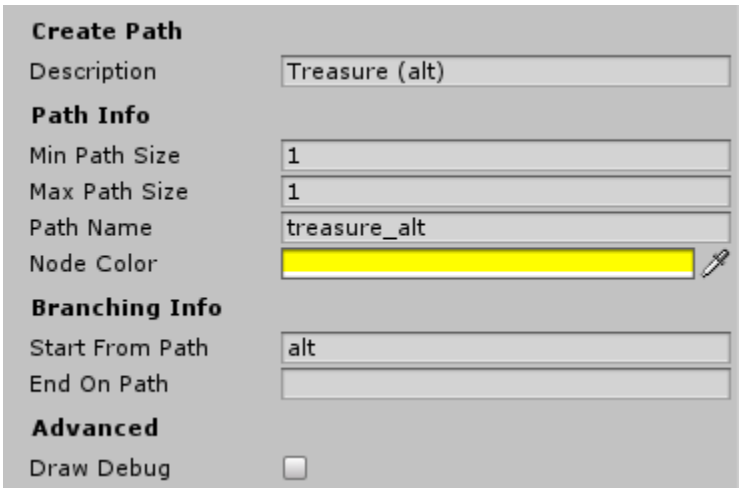



### 4.2.6 Create Treasure Room (Alt)

We'll add another treasure room connected to the `alt` path but keep the `End On Path` parameter empty so it doesn't connect back to anything:

Add a new node `Layout Graph > Create Path` and set it up as follows:




| | |
|---|---|
| **Min Path Size** | 1 |
| **Max Path Size** | 1 |
| **Path Name** | treasure_alt |
| **Node Color** | yellow |
| **Start From Path** | alt |
| **End On Path** | |

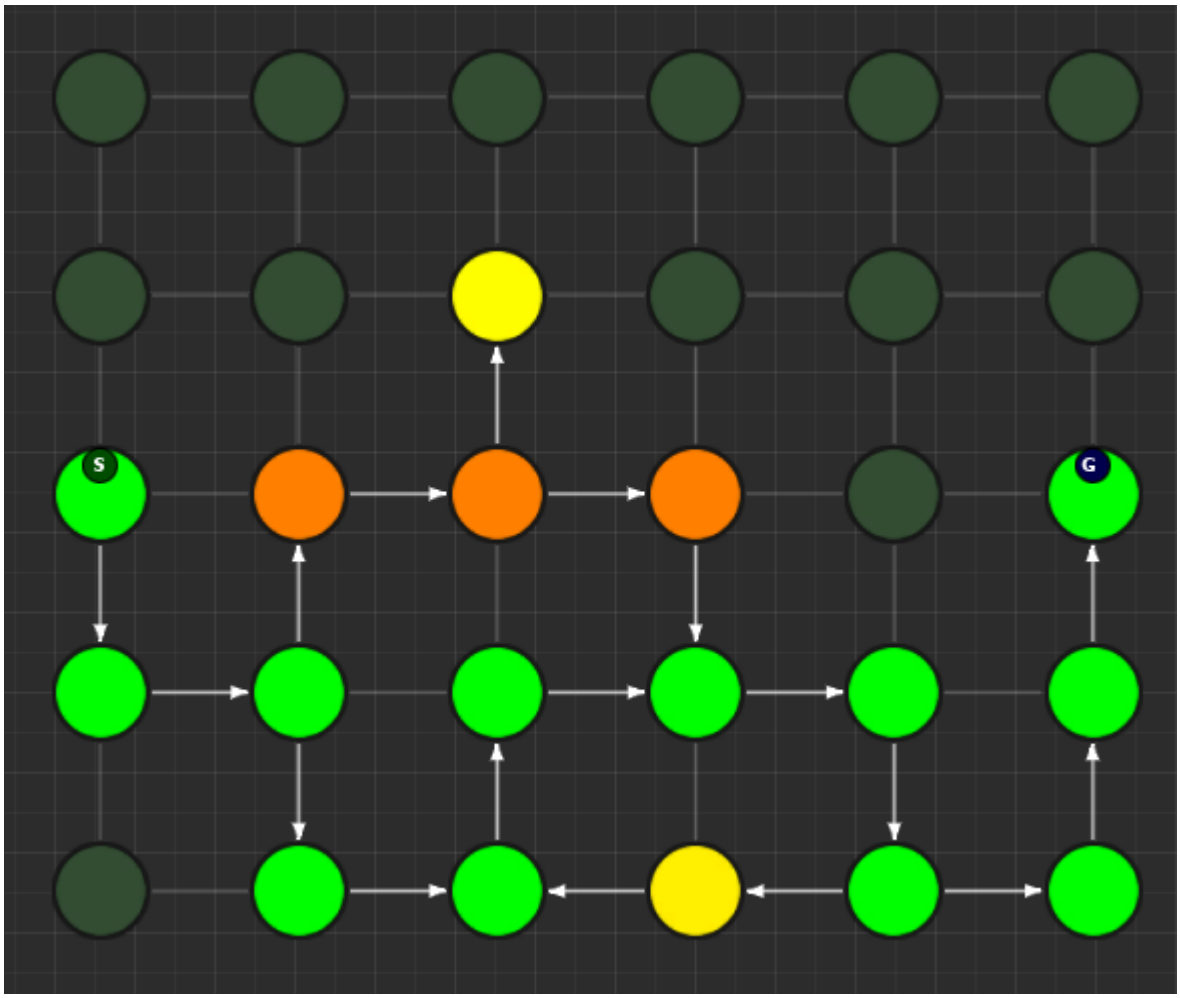
### 4.2.7 Create Key Room

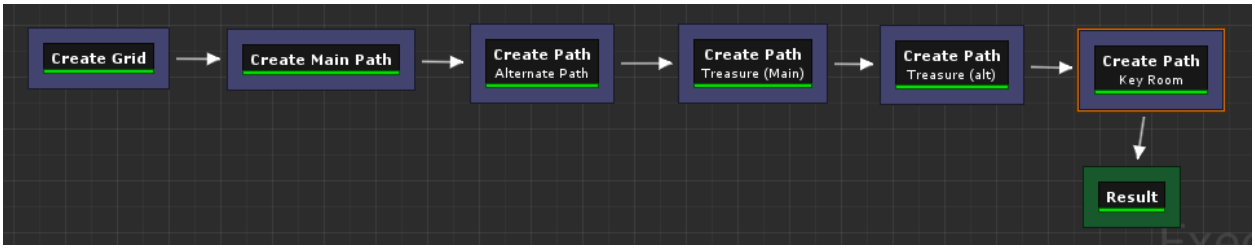
We'll create a room connected to the main path which will act as the key room. We'll later configure this room to have a key that opens up a lock in the main path. It will also have a NPC (key guardian) guarding the key

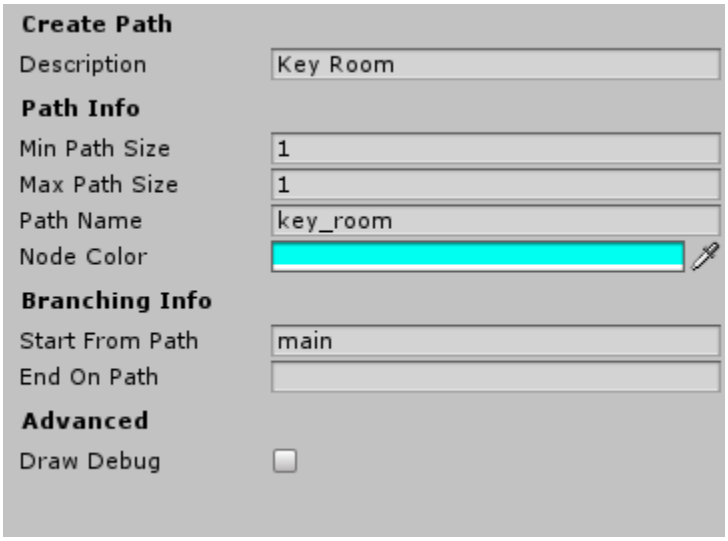
Add a new node `Layout Graph > Create Path` and set it up as follows:

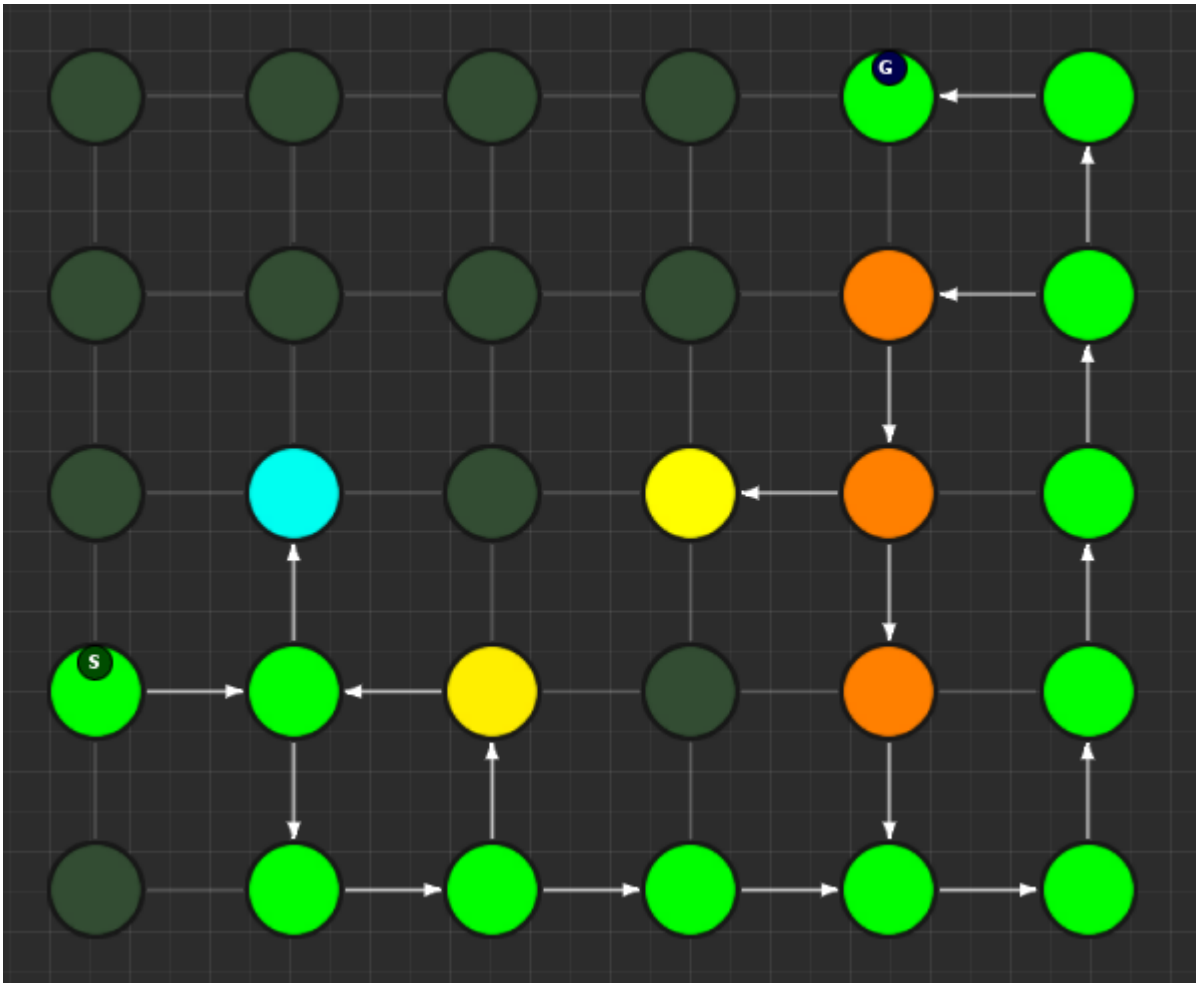
| | |
|---|---|
| **Min Path Size** | 1 |
| **Max Path Size** | 1 |
| **Path Name** | key_room |
| **Node Color** | cyan |
| **Start From Path** | main |
| **End On Path** | |

**Note:** We've named this path `key_room`. It will be referenced later on when creating the key locks


S
G


Create Grid
Create Main Path
Create Path
Alternate Path
Create Path
Treasure (Main)
Create Path
Treasure (alt)
Create Path
Key Room
Result

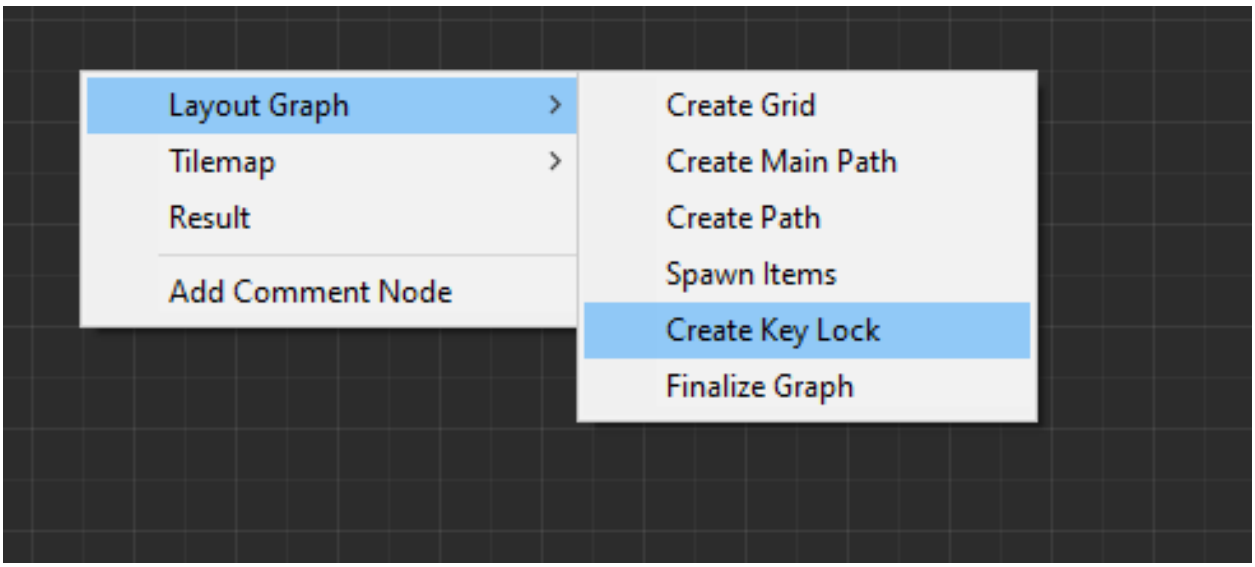

Create Path
Description
Key Room
Path Info
Min Path Size
1
Max Path Size
1
Path Name
key_room
Node Color
Branching Info
Start From Path
main
End On Path
Advanced
Draw Debug

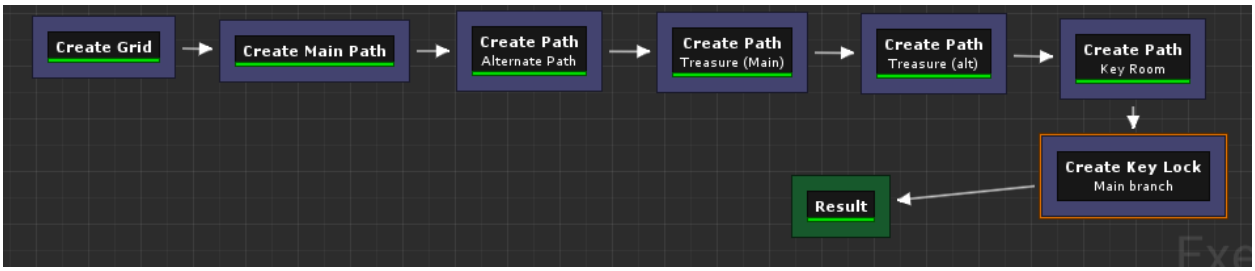

G
S

### 4.2.8 Create Key-Lock (Main)

We'll next create a key-lock system on the main path. Our key will go on the Key Room we created earlier (`key_room` path) and the lock will be somewhere in the main branch (`main` path)

Add a new node `Layout Graph > Create Key Lock` and set it up as follows:




| | |
|---|---|
| **Key Branch** | key_room |
| **Lock Branch** | main |
| **Key Marker Name** | KeyMain |
| **Lock Marker Name** | LockMain |

Specify the *Key Branch* as `key_room` and *Lock Branch* as `main`

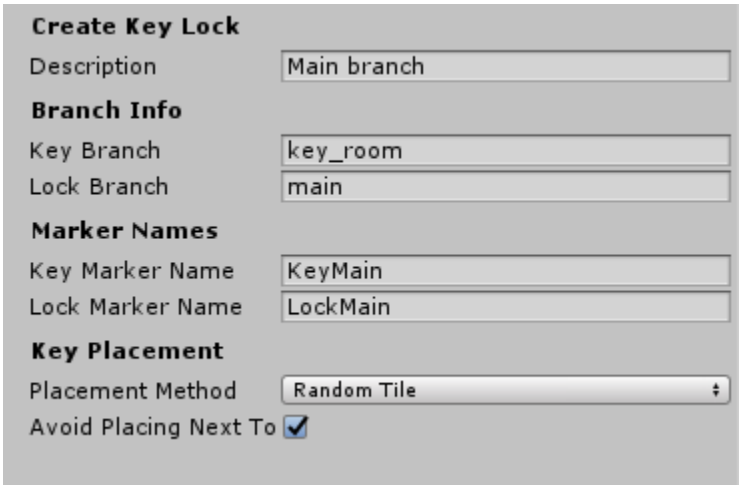
Set marker name for the key as `KeyMain` and lock as `LockMain`. Then in the theme file, you'd create marker nodes with these names and add your key and locked gate prefabs.

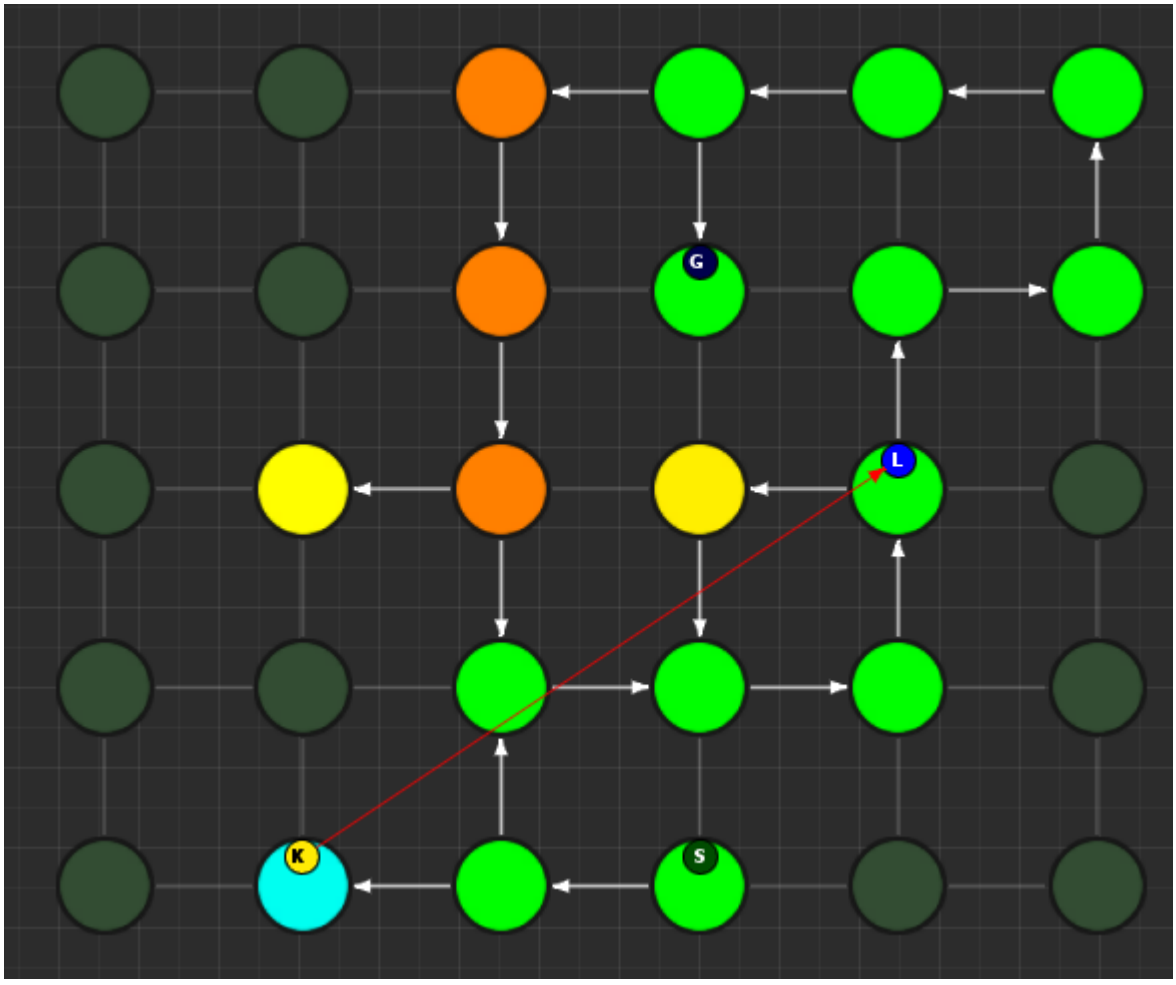
The prehistoric theme already has these setup

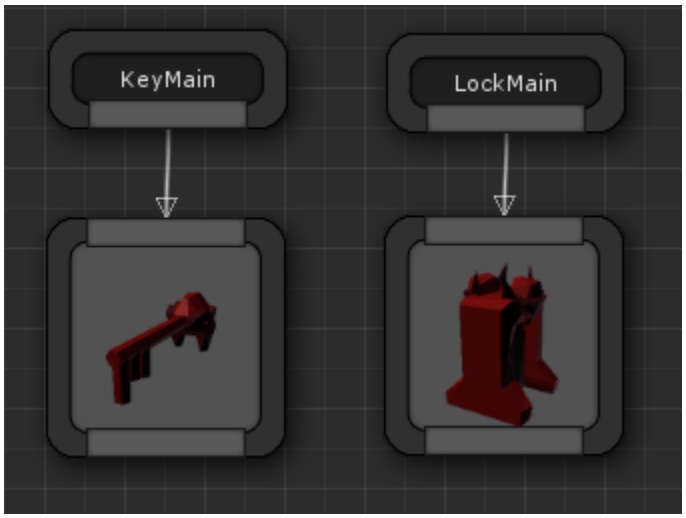
### 4.2.9 Create Key-Lock (Treasure Main)

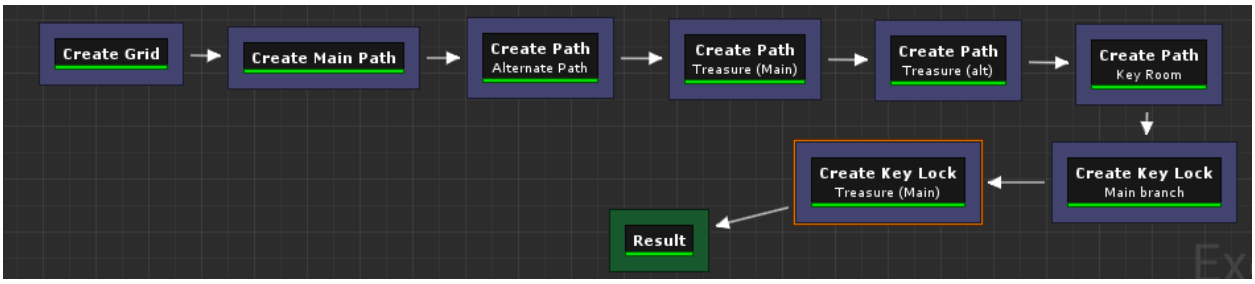
We need a key-lock to guard the treasure room in the main branch

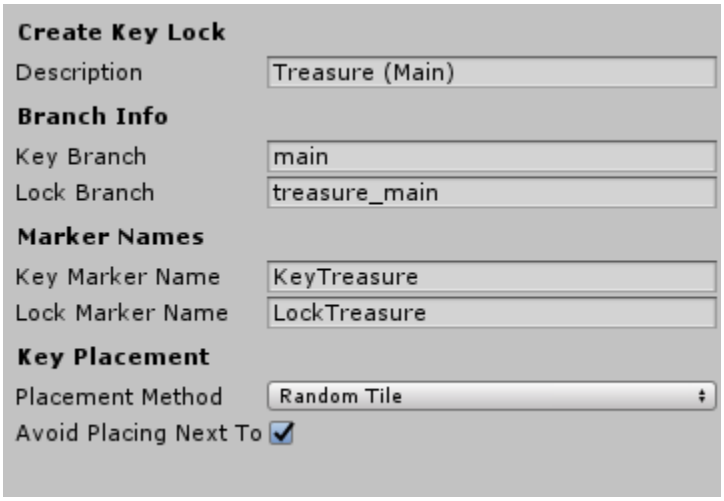
Add a new node `Layout Graph > Create Key Lock` and set it up as follows:


Create Key Lock
Description
Main branch
Branch Info
Key Branch
key_room
Lock Branch
main
Marker Names
Key Marker Name
KeyMain
Lock Marker Name
LockMain
Key Placement
Placement Method
Random Tile
Avoid Placing Next To

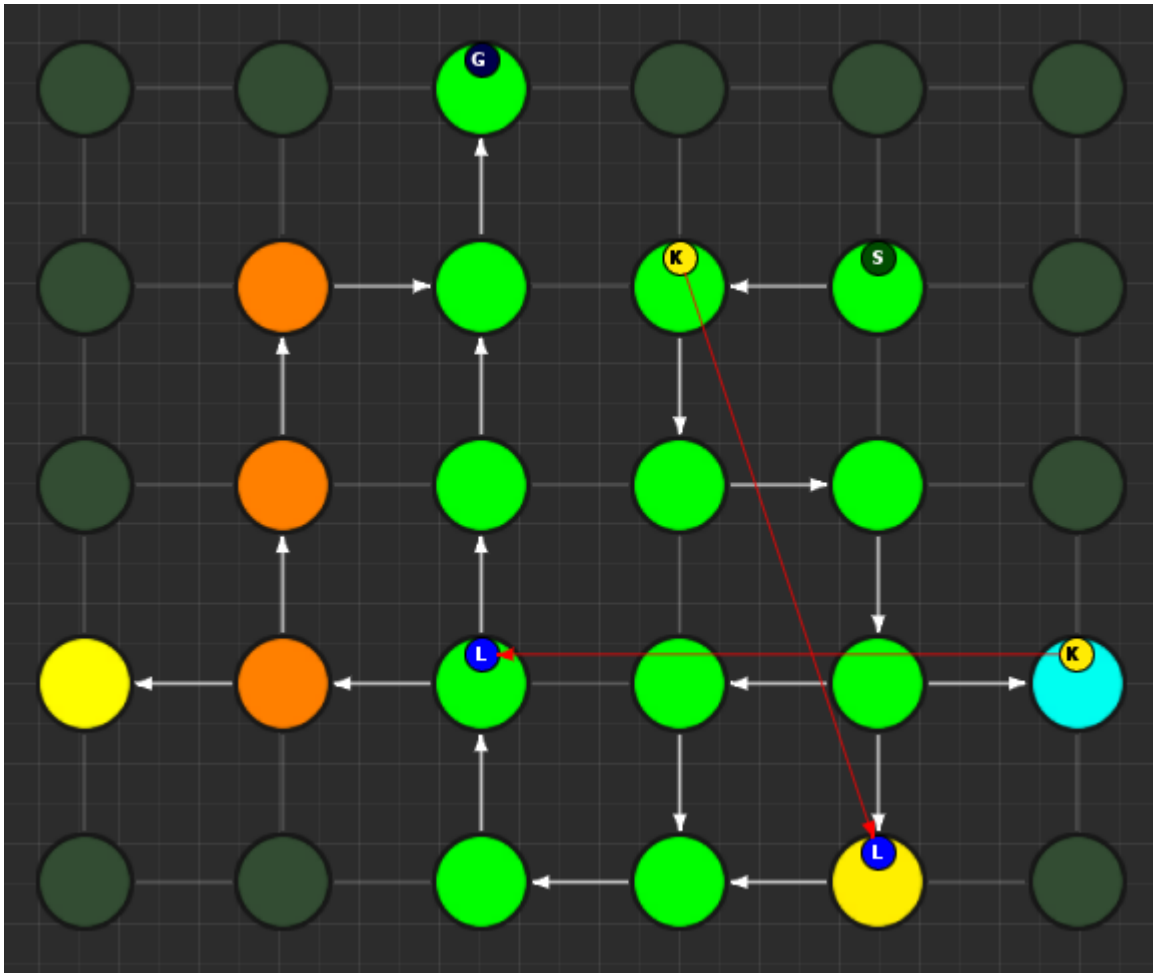

G
L
K
S


KeyMain
LockMain


Create Grid
Create Main Path
Create Path
Alternate Path
Create Path
Treasure (Main)
Create Path
Treasure (alt)
Create Path
Key Room
Create Key Lock
Main branch
Create Key Lock
Treasure (Main)
Result


Create Key Lock
Description
Treasure (Main)
Branch Info
Key Branch
main
Lock Branch
treasure_main
Marker Names
Key Marker Name
KeyTreasure
Lock Marker Name
LockTreasure
Key Placement
Placement Method
Random Tile
Avoid Placing Next To

| | |
|---|---|
| **Key Branch** | main |
| **Lock Branch** | treasure_main |
| **Key Marker Name** | KeyTreasure |
| **Lock Marker Name** | LockTreasure |




Set marker name for the key as `KeyTreasure` and lock as `LockTreasure`. Then in the theme file, you'd create marker nodes with these names and add your key and locked gate prefabs.

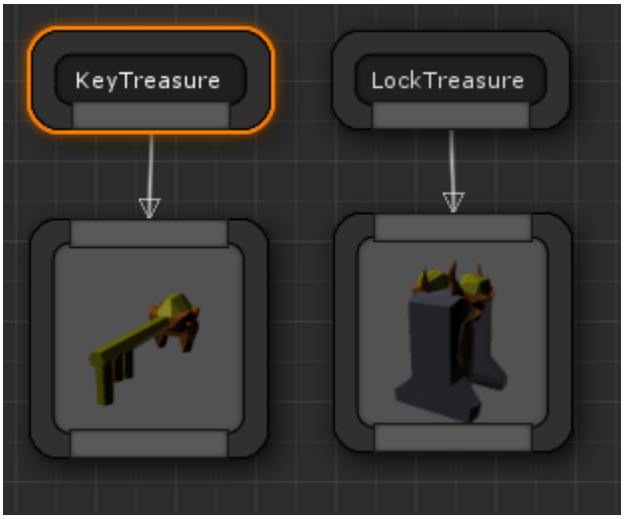
The prehistoric theme already has these setup

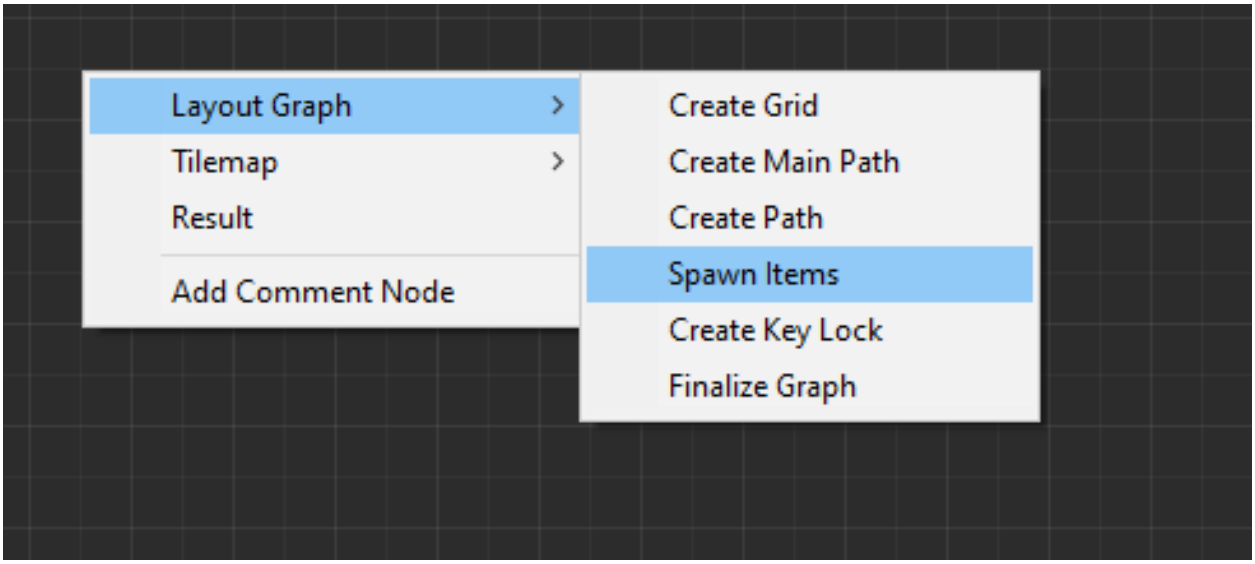
### 4.2.10 Spawn Enemies (Main, Alt)

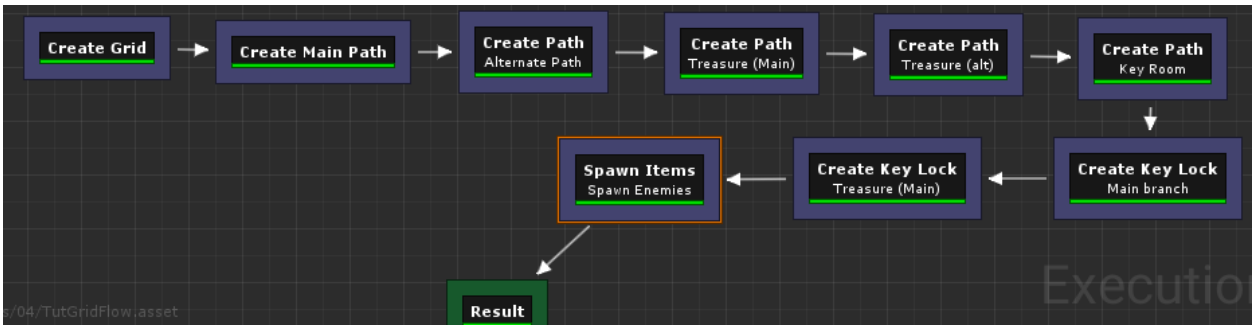
We'll use the `Spawn Items` node to spawn enemies on the `main` and `alt` paths

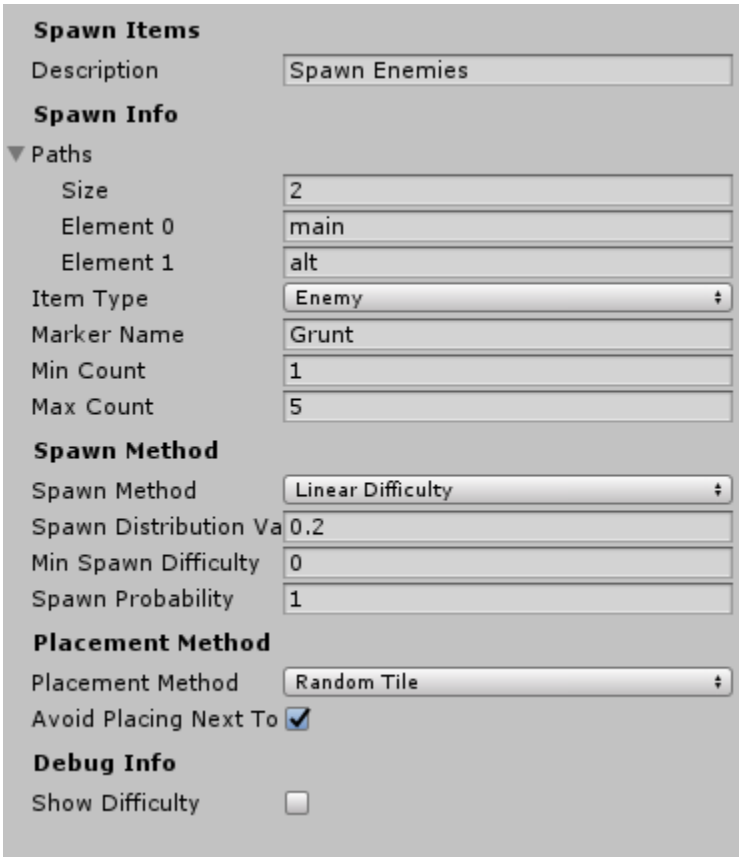
Create a new node `Layout Graph > Spawn Items` and set it up as follows:

| | |
|---|---|
| **Paths** | main, alt |
| **Item Type** | Enemy |
| **Marker Name** | Grunt |
| **Min Count** | 1 |
| **Max Count** | 5 |


KeyTreasure
LockTreasure


Layout Graph
Tilemap
Result
Add Comment Node
Create Grid
Create Main Path
Create Path
Spawn Items
Create Key Lock
Finalize Graph


Create Grid
Create Main Path
Create Path
Alternate Path
Create Path
Treasure (Main)
Create Path
Treasure (alt)
Create Path
Key Room
Create Key Lock
Main branch
Create Key Lock
Treasure (Main)
Spawn Items
Spawn Enemies
Result




This will spawn enemies in the nodes, gradually increasing the number of enemies based on the difficulty. The difficulty increases as we get closer to the goal. You can control this from the *Spawn Method* properties. Leave it to default for now

We've specified the marker name as `Grunt` and an appropriate marker node should be created in the theme file so we can spawn prefabs under it. The pre-historic theme already has this marker

You can control the placement of items (in the tilemap) from the *Placement Method* property section. Leave it to default for now

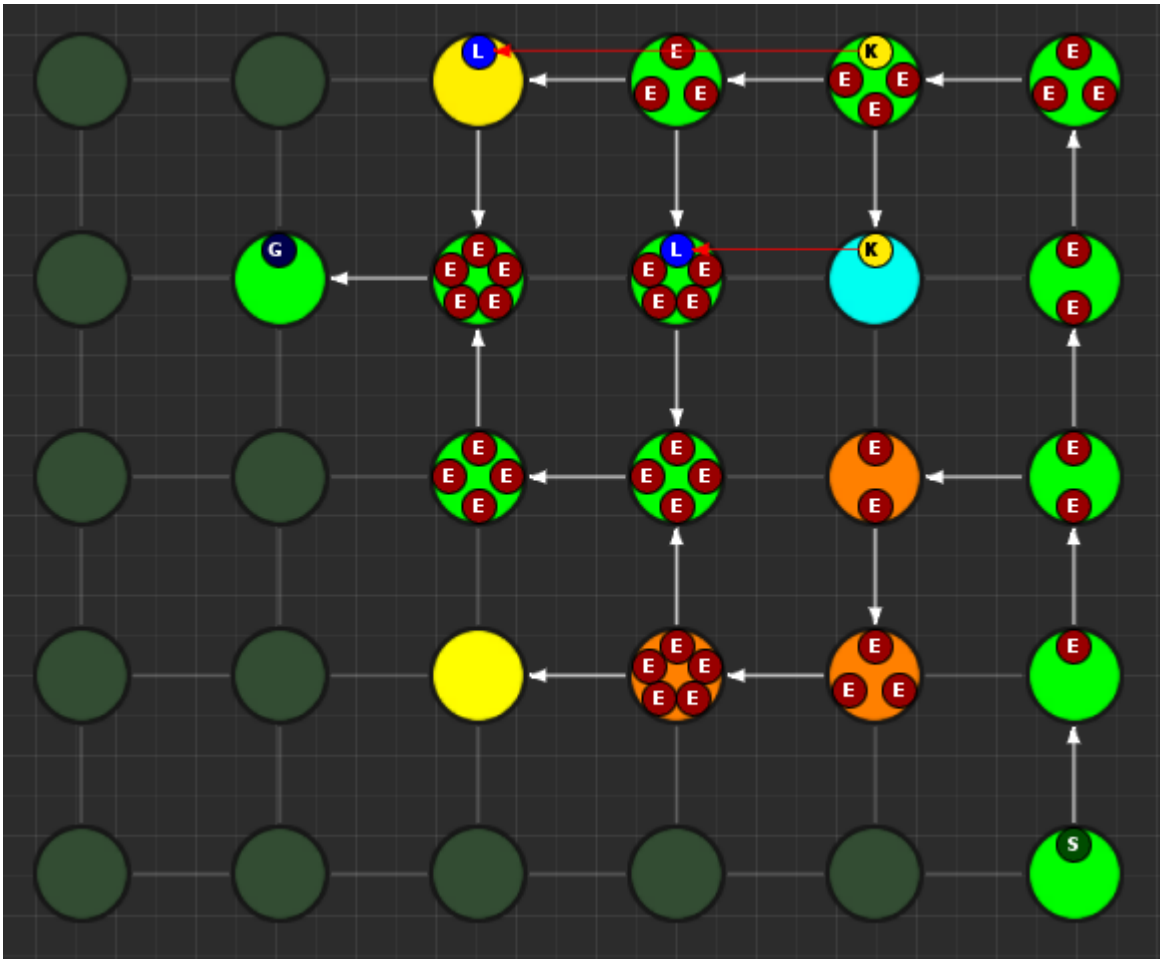
### 4.2.11 Spawn Bonus (Treasure Chests)

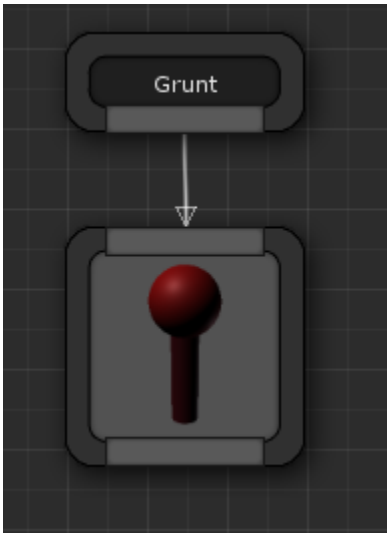
Spawn treasure chests in your bonus rooms using the *Spawn Items* node

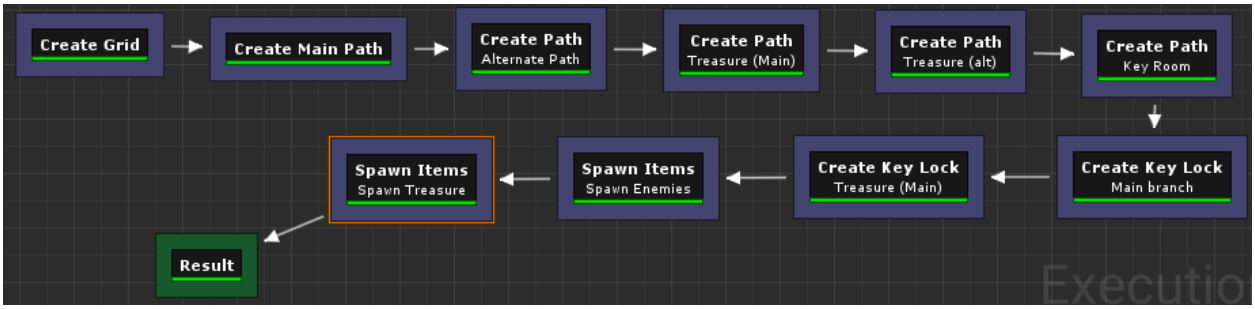
Create a new node `Layout Graph > Spawn Items` and set it up as follows:

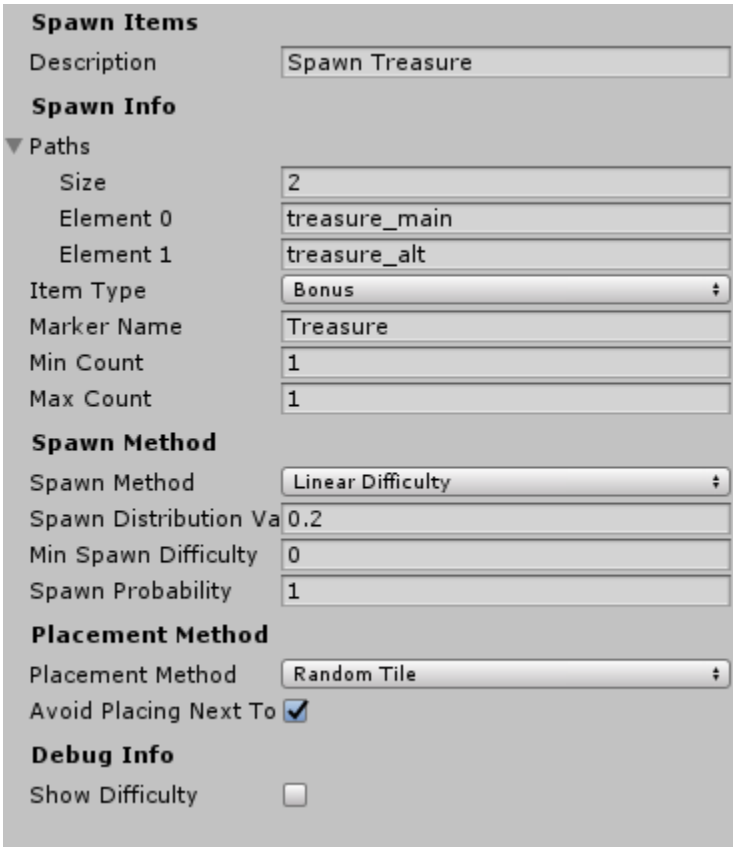
| | |
|---|---|
| **Paths** | treasure_main, treasure_alt |
| **Item Type** | Bonus |
| **Marker Name** | Treasure |
| **Min Count** | 1 |
| **Max Count** | 1 |
| **Min Spawn Difficulty** | 1 |

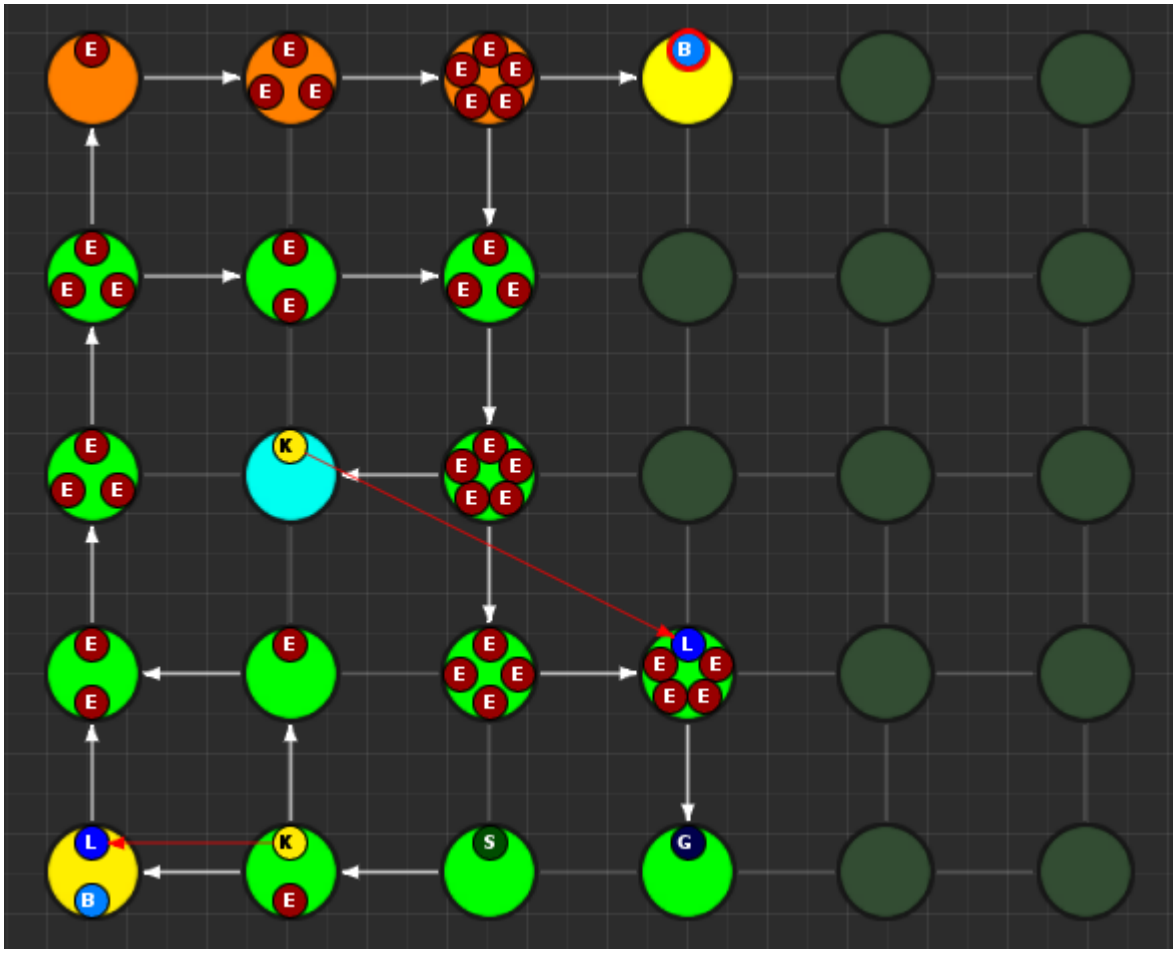
We've specified the marker name as `Treasure` and an appropriate marker node should be created in the theme file so we can spawn prefabs the treasure chest under it. The pre-historic theme already has this marker


L
E
K
E
G
L
K
S


Grunt


Create Grid
Create Main Path
Create Path
Alternate Path
Create Path
Treasure (Main)
Create Path
Treasure (alt)
Create Path
Key Room
Create Key Lock
Main branch
Create Key Lock
Treasure (Main)
Spawn Items
Spawn Enemies
Spawn Items
Spawn Treasure
Result


Spawn Items
Description
Spawn Treasure
Spawn Info
Paths
Size
2
Element 0
treasure_main
Element 1
treasure_alt
Item Type
Bonus
Marker Name
Treasure
Min Count
1
Max Count
1
Spawn Method
Spawn Method
Linear Difficulty
Spawn Distribution Va
0.2
Min Spawn Difficulty
0
Spawn Probability
1
Placement Method
Placement Method
Random Tile
Avoid Placing Next To
Debug Info
Show Difficulty


E
B
K
L
S
G

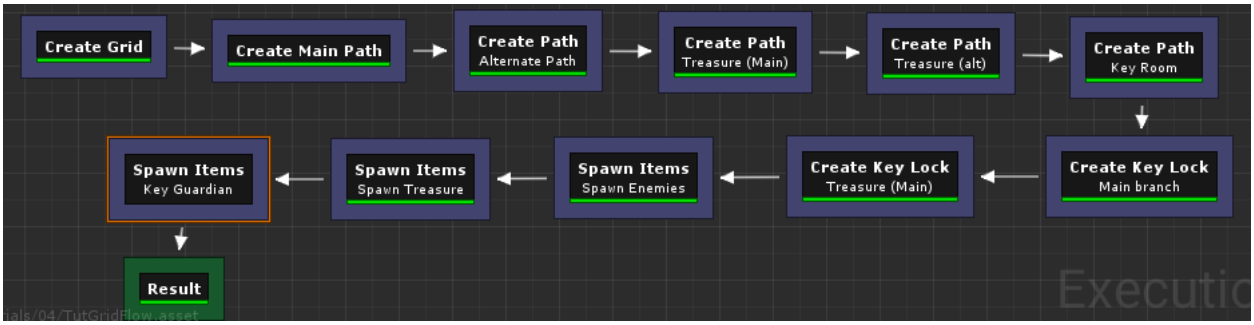

Treasure

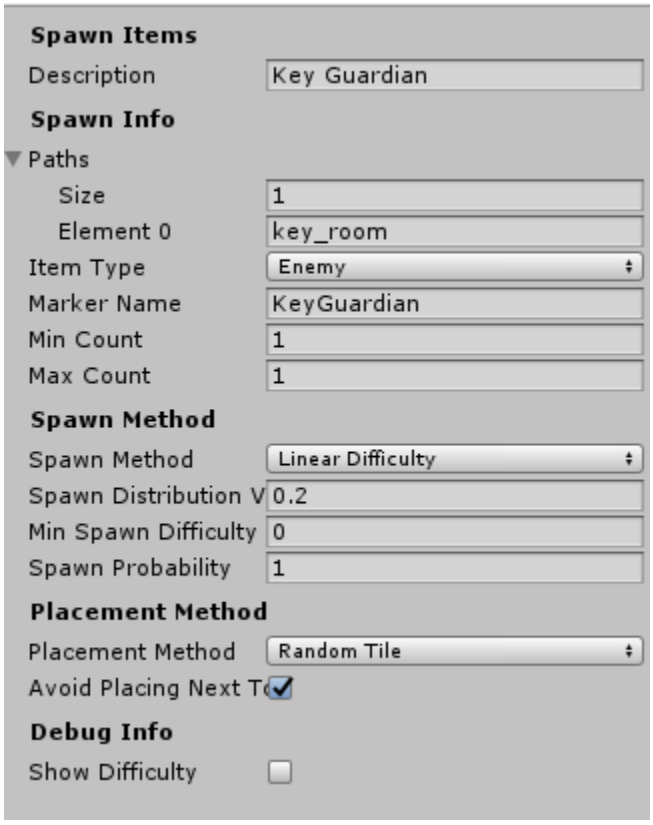
The *Min Spawn Difficulty* is set to `1`. The first node in the branch will have a difficulty of `0` and the last node `1`. Sometimes, the yellow branch may be 3 nodes long. Since we want the chest to occur only on the last node, we've set this value to `1`

### 4.2.12 Spawn Key Guardian

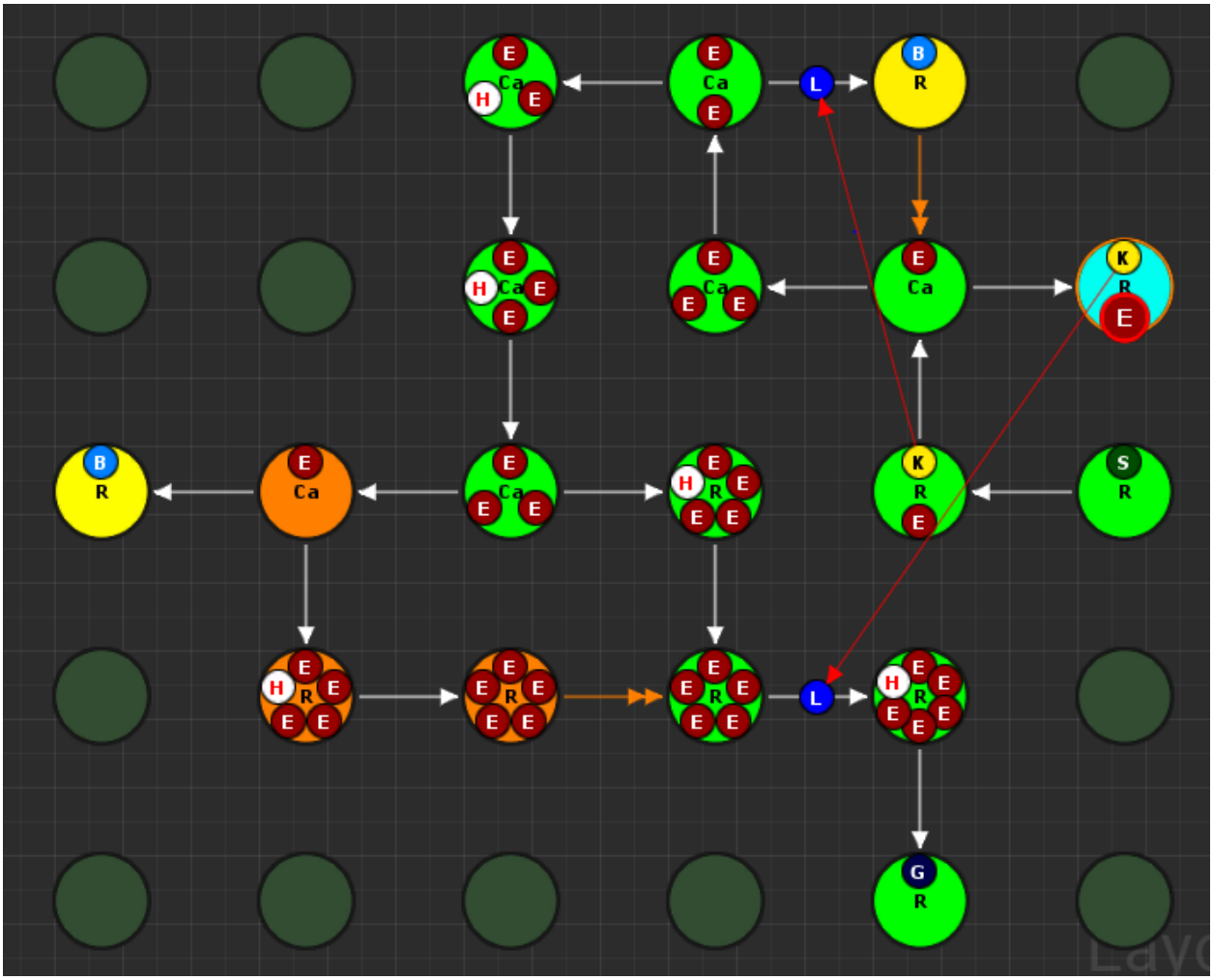
We'll add an NPC in the Key room guarding the key

Create a new node `Layout Graph > Spawn Items` and set it up as follows:




| | |
|---|---|
| **Paths** | key_room |
| **Item Type** | Enemy |
| **Marker Name** | KeyGuardian |
| **Min Count** | 1 |
| **Max Count** | 1 |


E
Ca
H
E
E
Ca
E
L
B
R
E
H
Ca
E
E
E
Ca
E
E
E
Ca
K
R
E
B
R
E
Ca
E
E
Ca
E
E
H
R
E
E
E
K
R
E
S
R
E
H
R
E
E
E
E
R
E
E
E
E
R
E
E
L
H
R
E
E
E
E
G
R

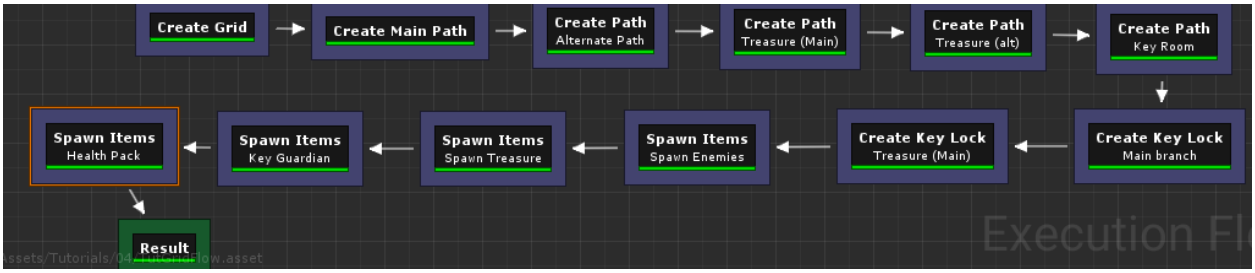
You'll need to create a marker named `KeyGuardian` in the theme file and place your NPC prefab under it. This marker doesn't exist in the *Prehistoric* theme and you'll need to create it yourself if you want to visualize it

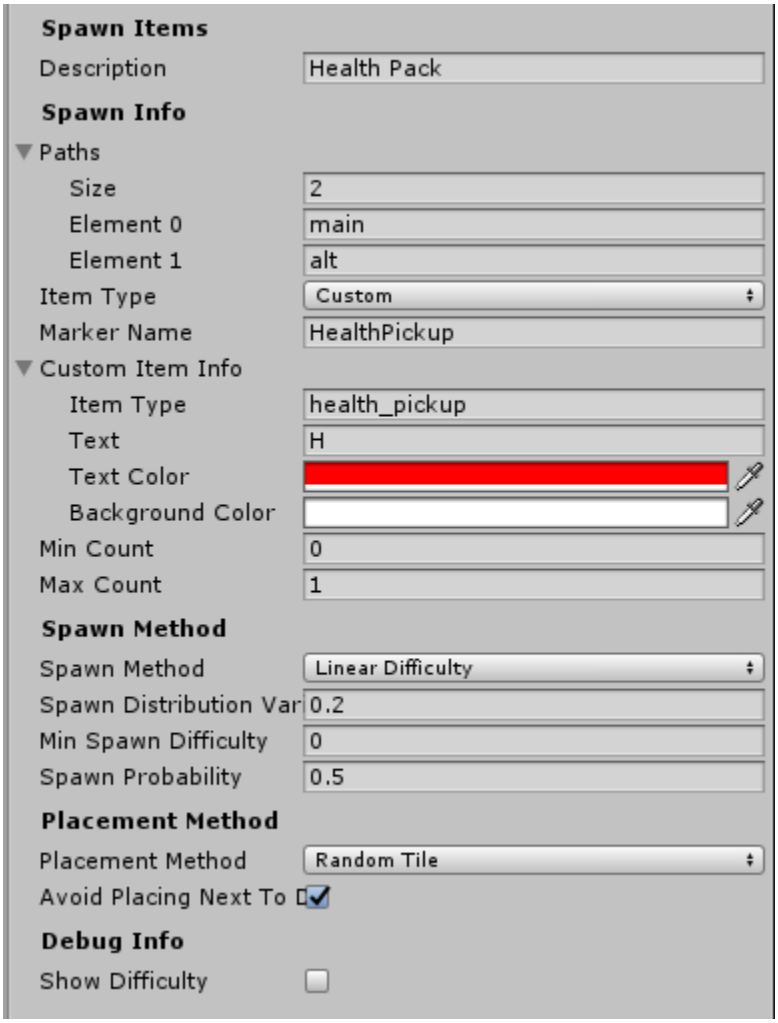
### 4.2.13 Spawn Health Pack

We'll use the *Spawn Items* node to spawn a few health pickups along the *main* and *alt* paths

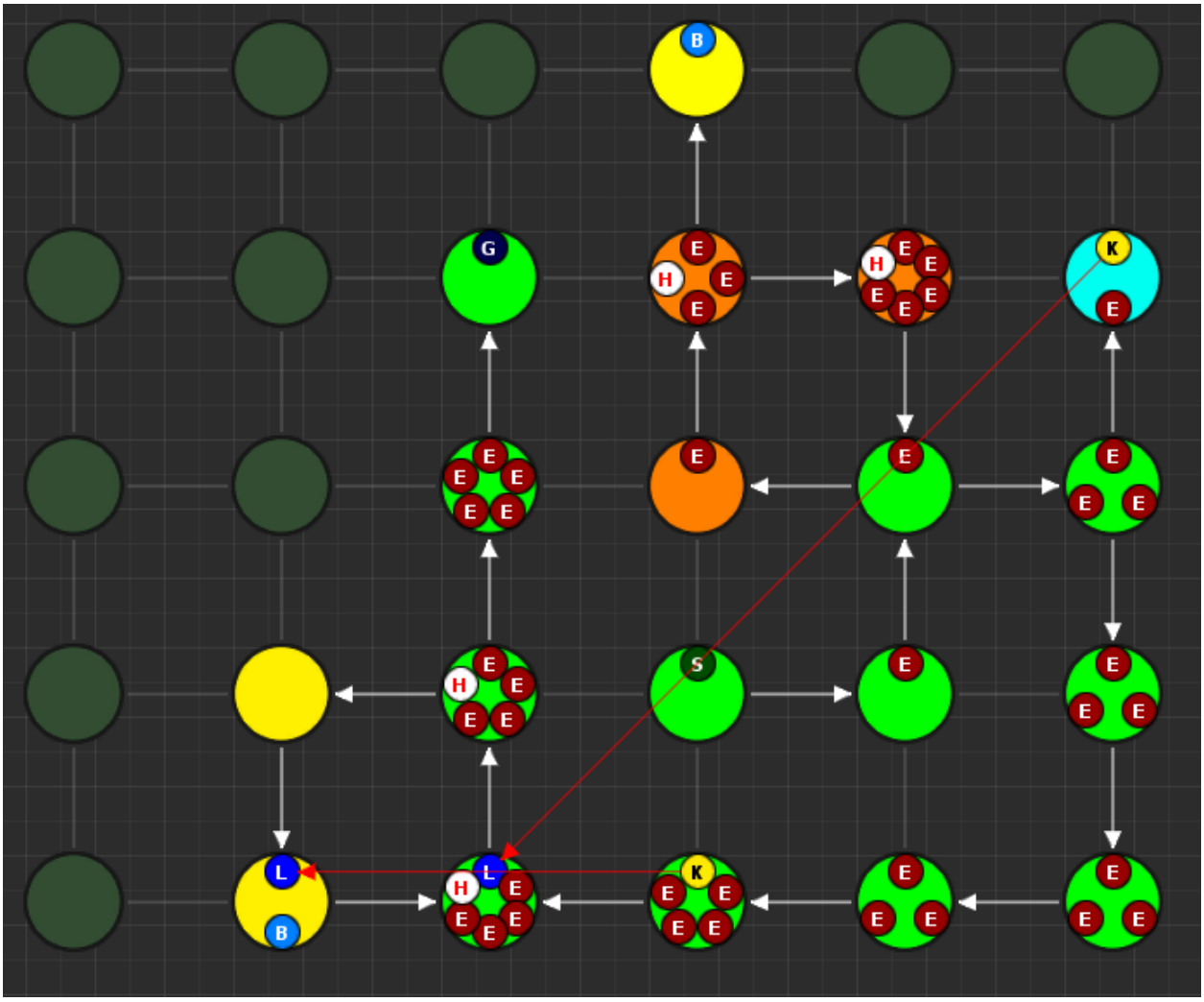
This section also shows you how to use the *Custom* Item Type

Create a new node `Layout Graph > Spawn Items` and set it up as follows:



| | |
|---|---|
| **Paths** | main, alt |
| **Item Type** | Custom |
| **Marker Name** | HealthPickup |
| **Min Count** | 0 |
| **Max Count** | 1 |
| **Spawn Probability** | 0.5 |
| **Custom Item Info** | |
| **>> Item Type** | health_pickup |
| **>> Text** | Health |
| **>> Text Color** | [Red] |
| **>> Background Color** | [White] |



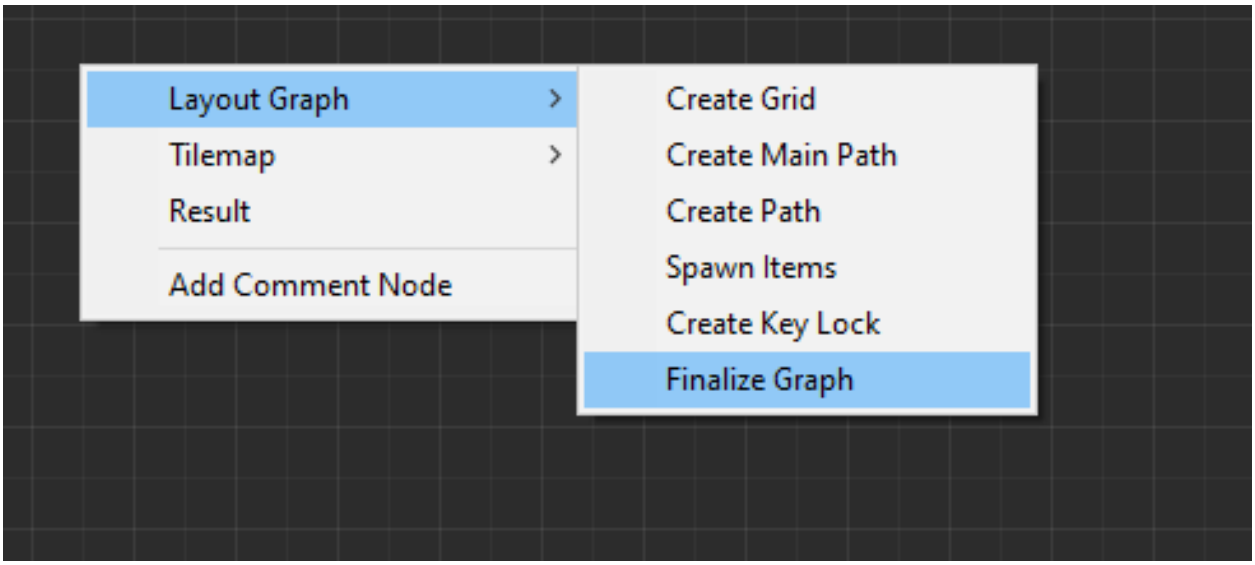

**Note:** You'll need to create a marker named `HealthPickup` in your theme file and add your health pack prefab

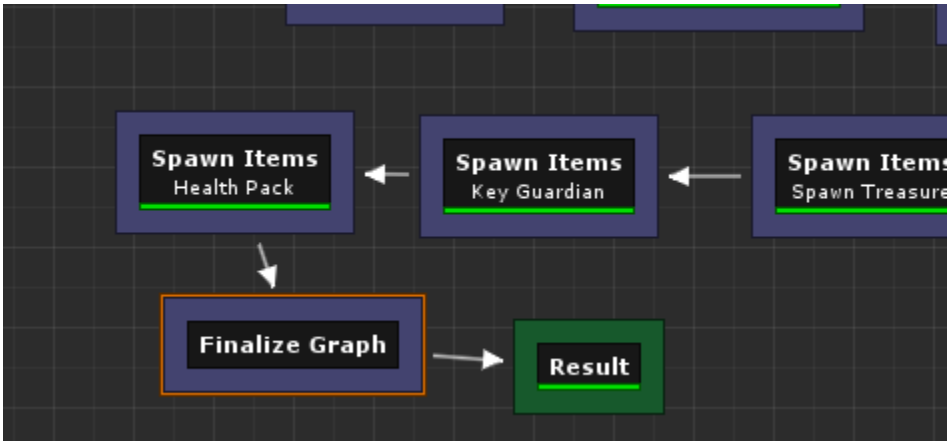
### 4.2.14 Finalize Layout Graph

After we are done designing the layout graph, we'll need to finalize it with the *Finalize Graph* node. This node does a few things:

- Move the locks from the nodes on to the links
- Create one way doors (so we don't go around locked doors)
- Assign room types (Room, Corridor, Cave)

Create a new node `Layout Graph > Finalize Graph` and set it up as follows:



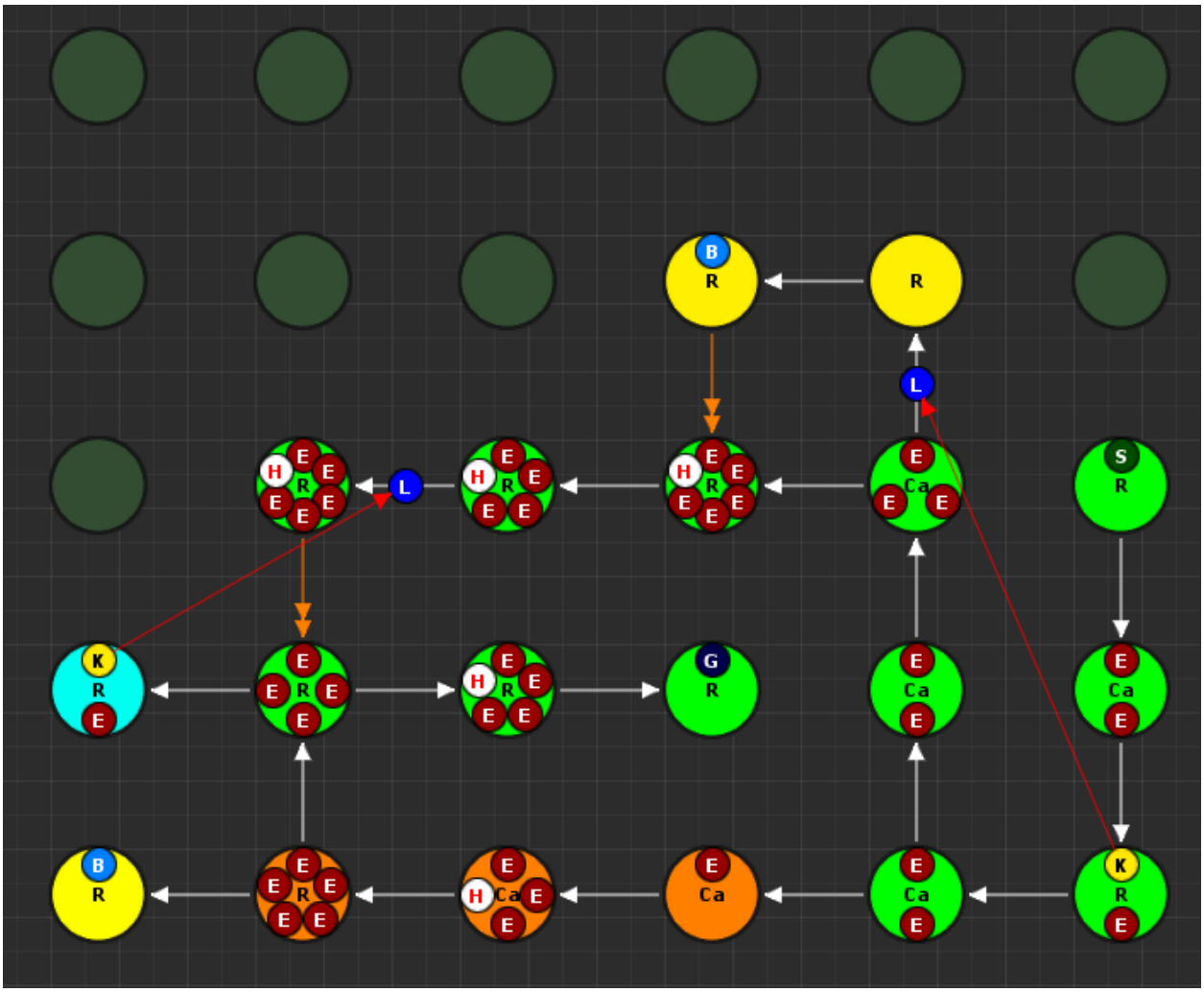

Leave all the properties to default

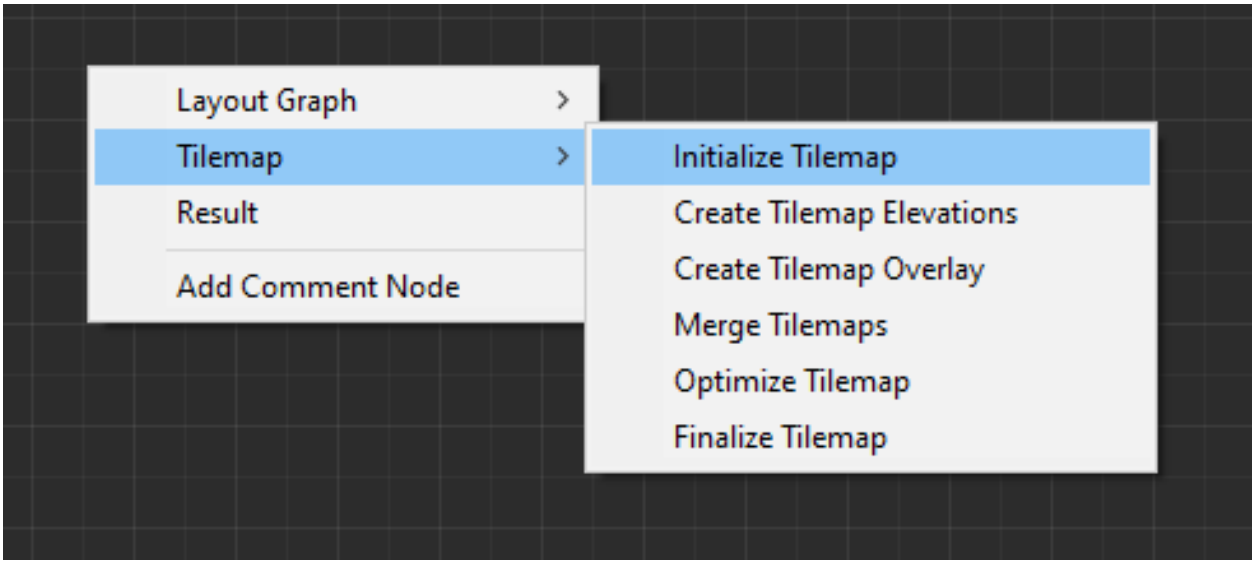
We are now ready to create a tilemap from this

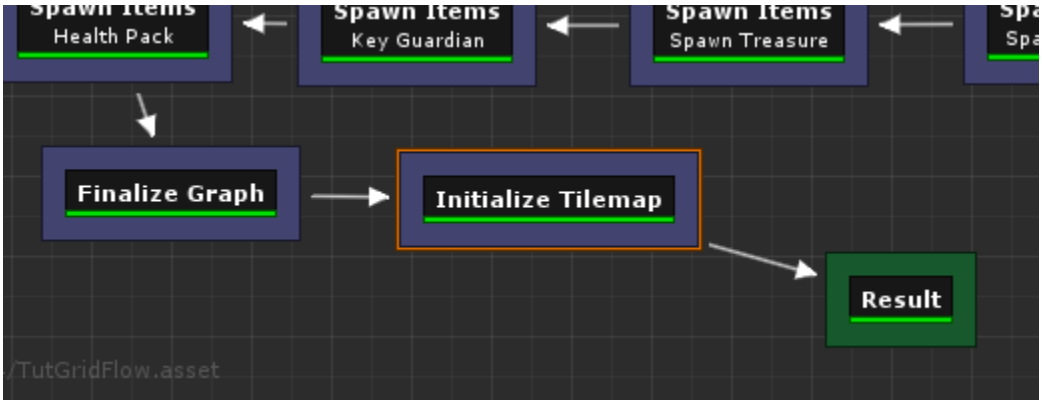
### 4.2.15 Initialize Tilemap

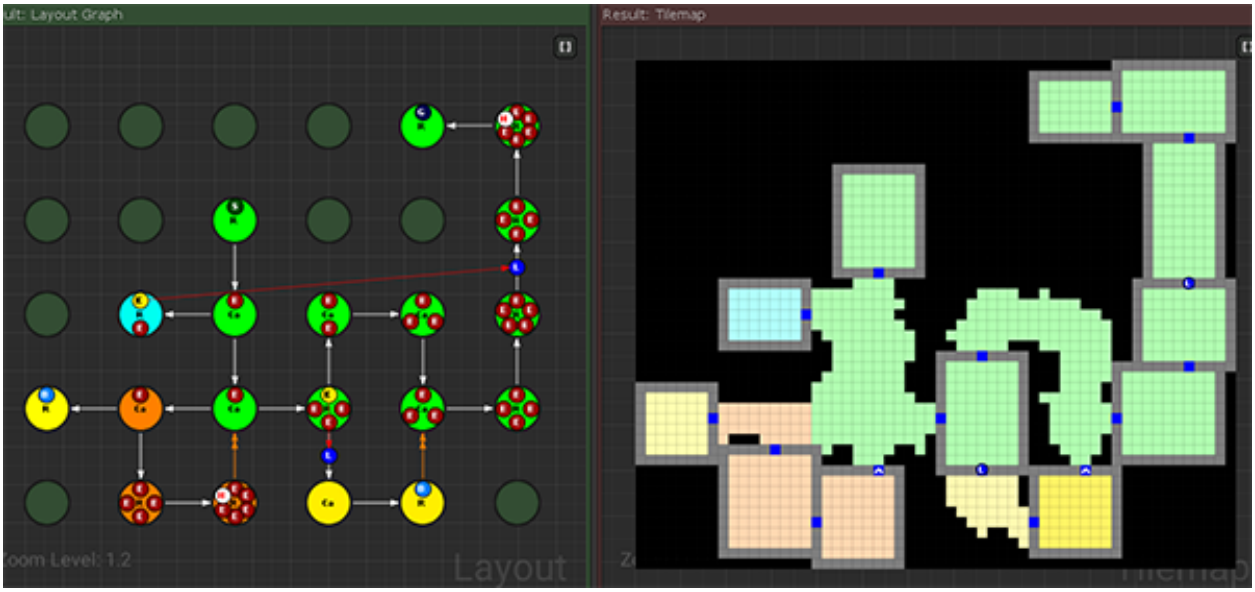
Create a new node `Tilemap > Initialize Tilemap` and set it up as follows:

You can control the thickness of the caves from the *Cave Thickness* parameter. Each node on the layout graph gets converted into rooms in the tilemap.


B
R
R
L
H
E
R
E
E
E
L
H
E
R
E
E
H
E
R
E
E
E
Ca
E
E
S
R
K
R
E
E
E
R
E
E
H
E
R
E
E
G
R
E
Ca
E
E
Ca
E
B
R
E
E
R
E
E
E
H
E
Ca
E
E
Ca
E
E
Ca
E
K
R
E


Layout Graph
Tilemap
Result
Add Comment Node
Initialize Tilemap
Create Tilemap Elevations
Create Tilemap Overlay
Merge Tilemaps
Optimize Tilemap
Finalize Tilemap




**Initialize Tilemap**

| | |
|---|---|
| Description | |
| **Layout Settings** | |
| Tilemap Size Per Node | 10 |
| Perturb Amount | 3 |
| Corridor Lane Width | 2 |
| Layout Padding | 5 |
| **Cave Settings** | |
| Cave Thickness | 2.25 |
| Cave Automata Neighbors | 5 |
| Cave Automata Iterations | 4 |
| **Color Settings** | |
| Room Color Saturation | 0.3 |
| Room Color Brightness | 1.5 |

The parameter *Tilemap Size Per Node* controls how many tiles are used to generate a room from the node. Bump this number up if you want more space in your rooms

If you want a more uniform grid like look on your rooms, bring the *Perturb Amount* close to `0`

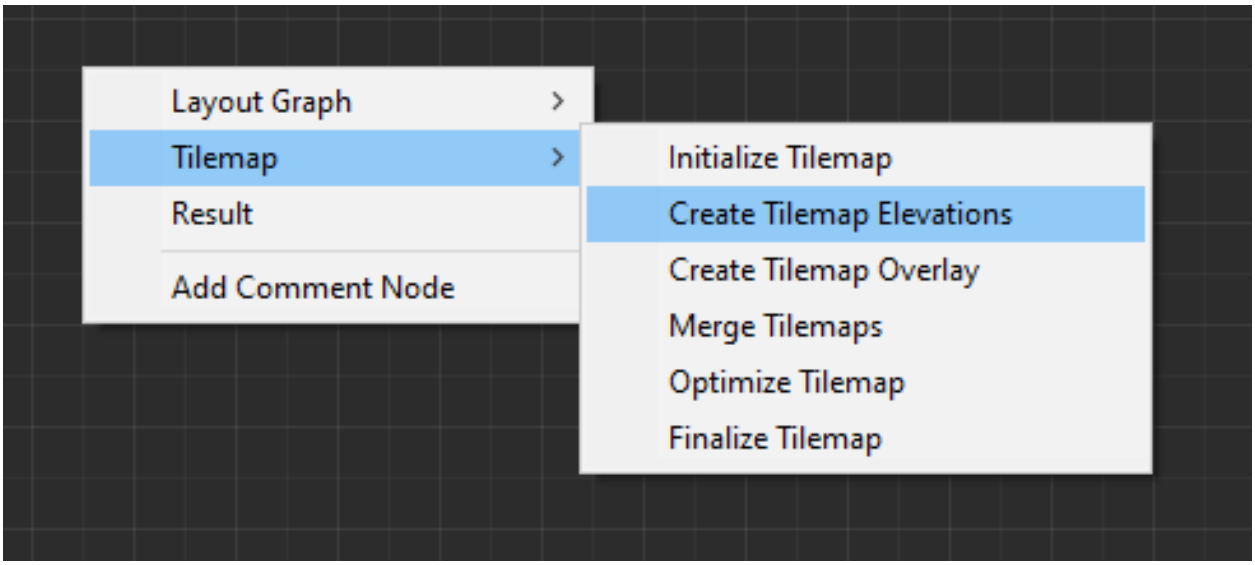
*Layout Padding* adds extra tiles around the dungeon layout. Set to `5` so we can apply some decorations outside the dungeon bounds

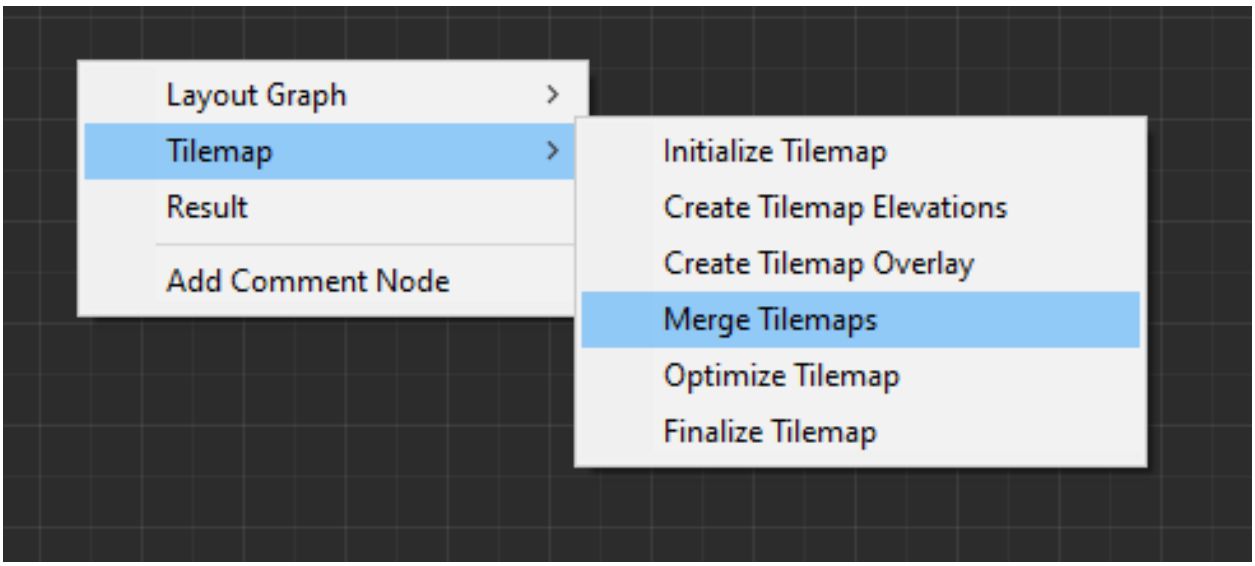
When you select a node on the layout graph, the tiles that belong to the node light up. This is controlled by the *Color Settings* parameters

### 4.2.16 Add Background Elevation

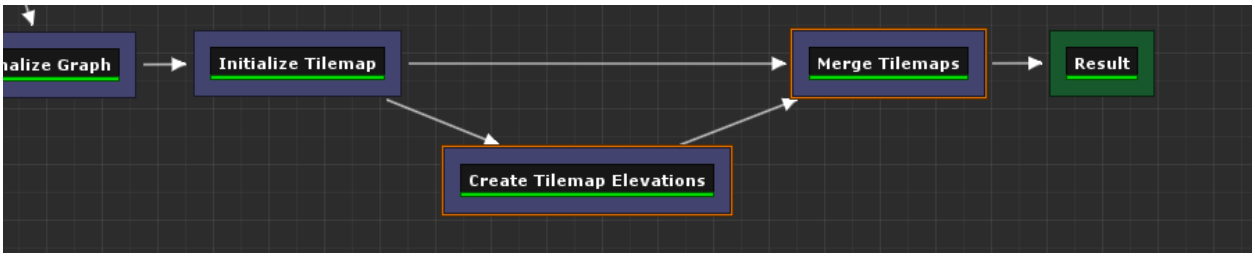
We are going to create overlays and merge them with the original tilemap. Create the following two nodes:

- Create a node `Tilemap > Create Tilemap Elevations`
- Create a node `Tilemap > Merge Tilemaps`



Link them up like below:



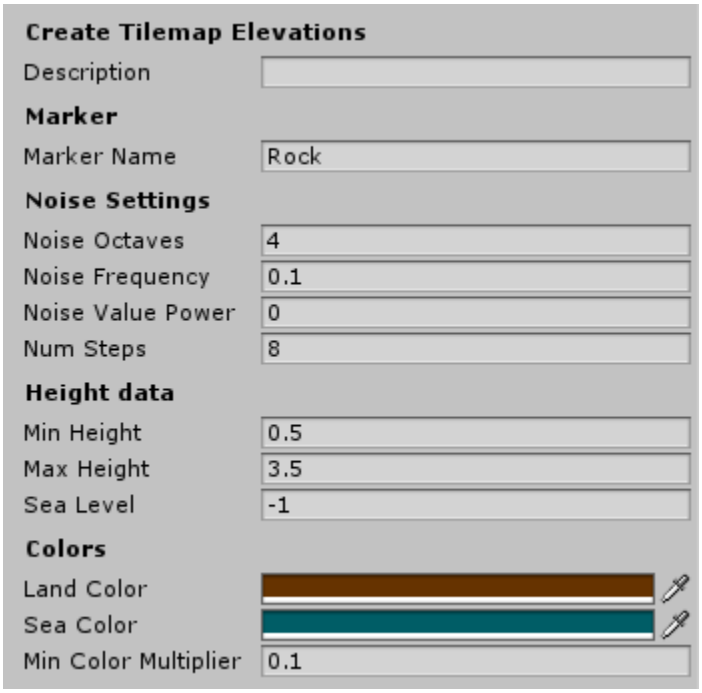




Fig. 9: Create Tilemap Elevation properties

Update the properties

| | |
|---|---|
| **Noise Frequency** | 0.1 |
| **Num Steps** | 8 |
| **Min Height** | 0.5 |
| **Max Height** | 3.5 |
| **Sea Level** | -1 |

We've specified the marker name as `Rock`. If you place objects under the specified marker node in the theme editor, they will show up on these tiles at the given height

---

**Note:** The Min/Max height is logical and will be mulitplied by the dungeon config's Grid Size Y value. If the GridSize is `(4, 2, 4)` in the DungeonGridFlow game object's config and the tile height happens to be `2.5`, the actual placement will be on `2.5 * 2 = 5`

---

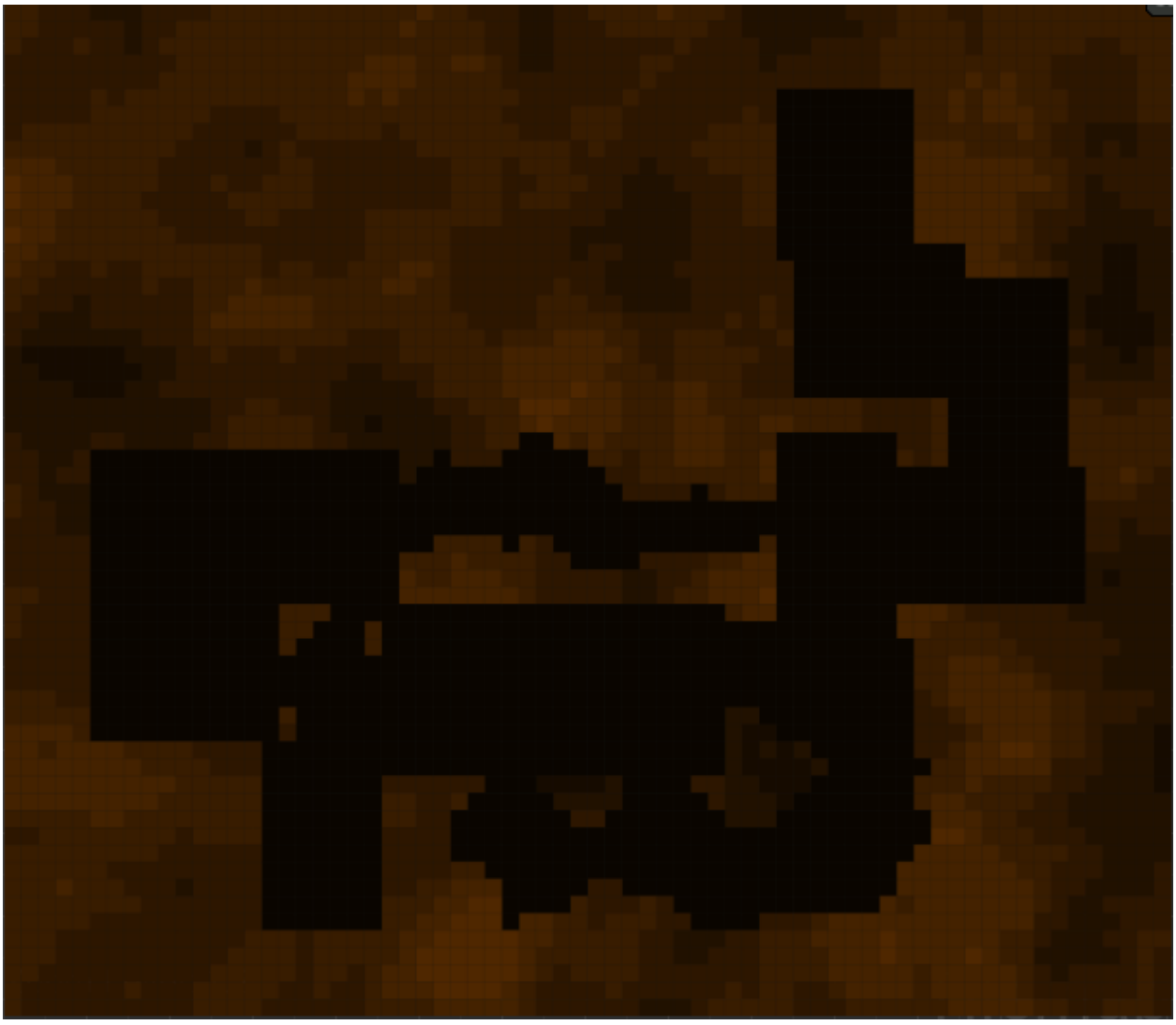


Fig. 10: Create Tilemap Elevation Node Result

### 4.2.17 Add Tree Overlays

We'll overlay trees on our dungeon using a noise parameter. These overlays will be placed such that they will not block the main path

Create a node `Tilemap > Create Tilemap Overlay`

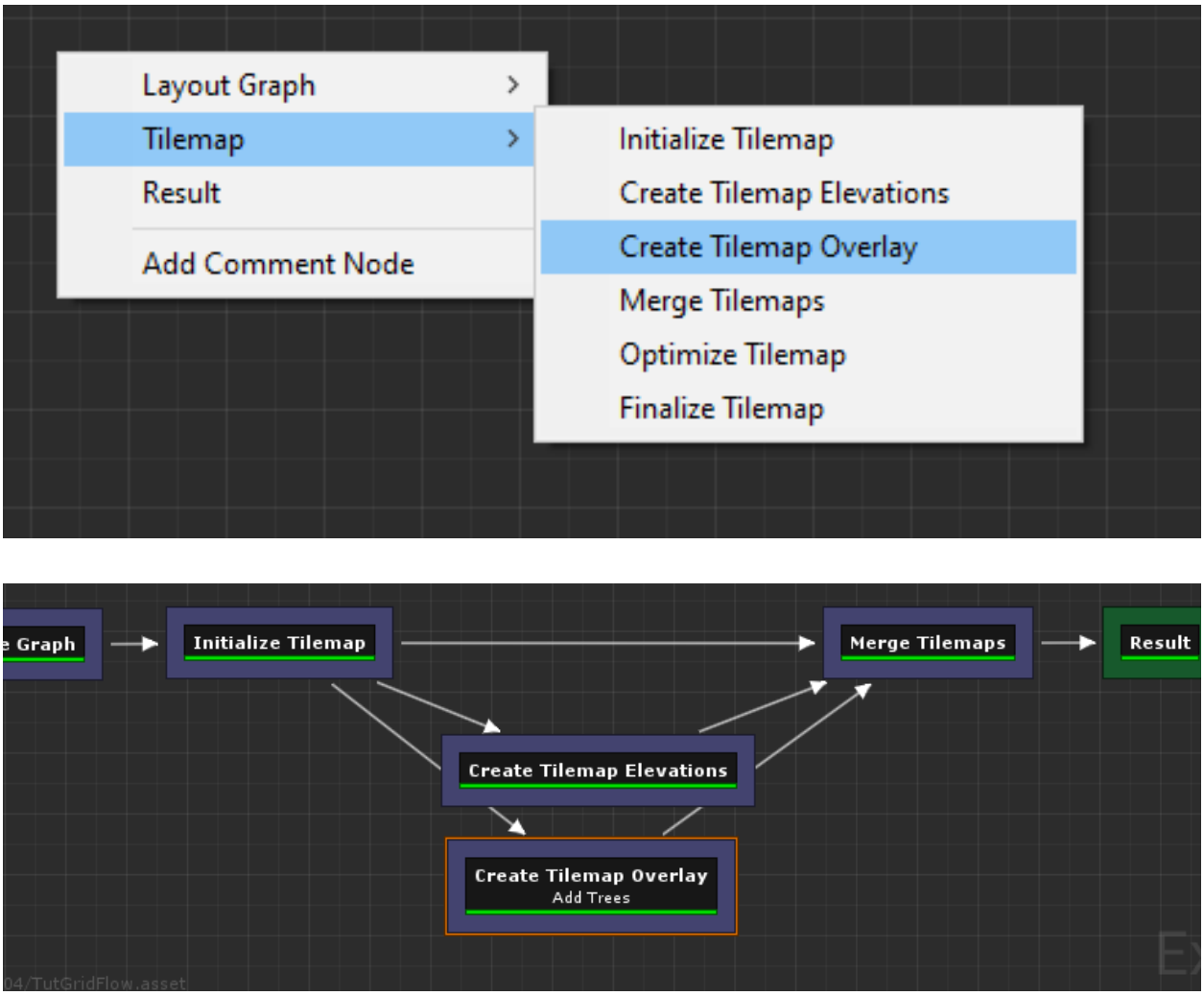



Fig. 11: Create Tilemap Overlay Node Connection

| **Noise Settings** | |
|---|---|
| **> Noise Frequency** | 0.2 |
| **> Noise Max Value** | 1.5 |
| **> Noise Threshold** | 0.75 |
| **> Min Dist From Main Path** | 1 |
| **Merge Config** | |
| **> Max Height** | 1 |

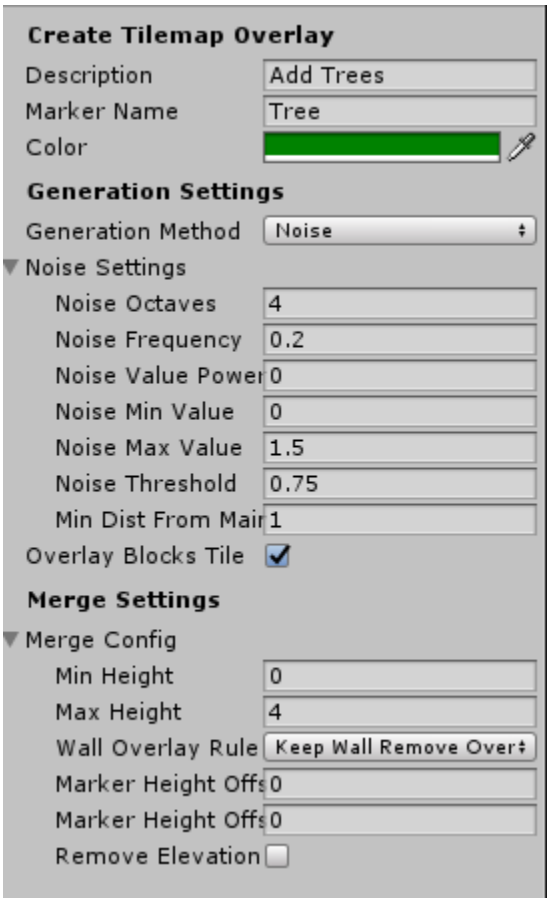



Fig. 12: Create Tilemap Overlay Node properties

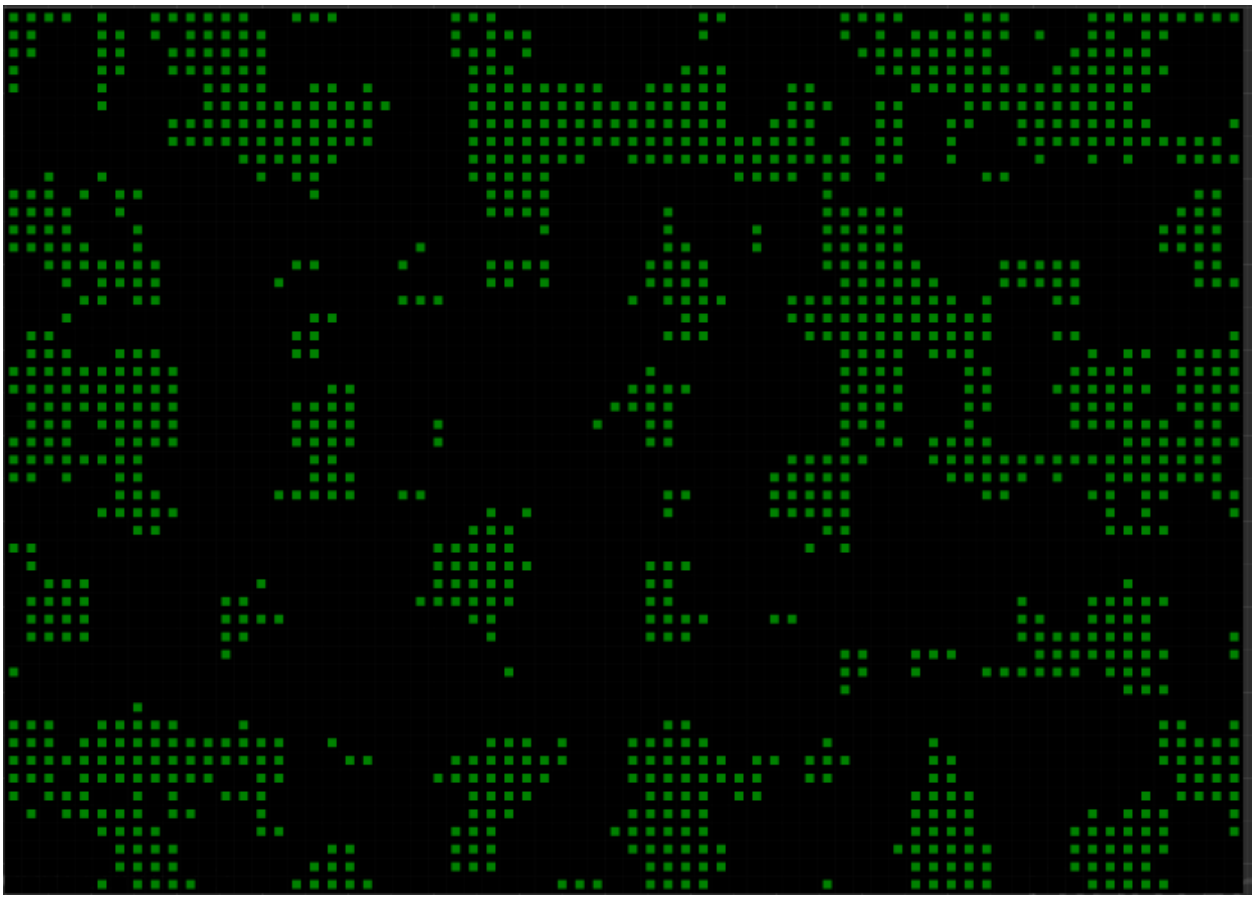


Fig. 13: Result of the Create Tilemap Overlay Node

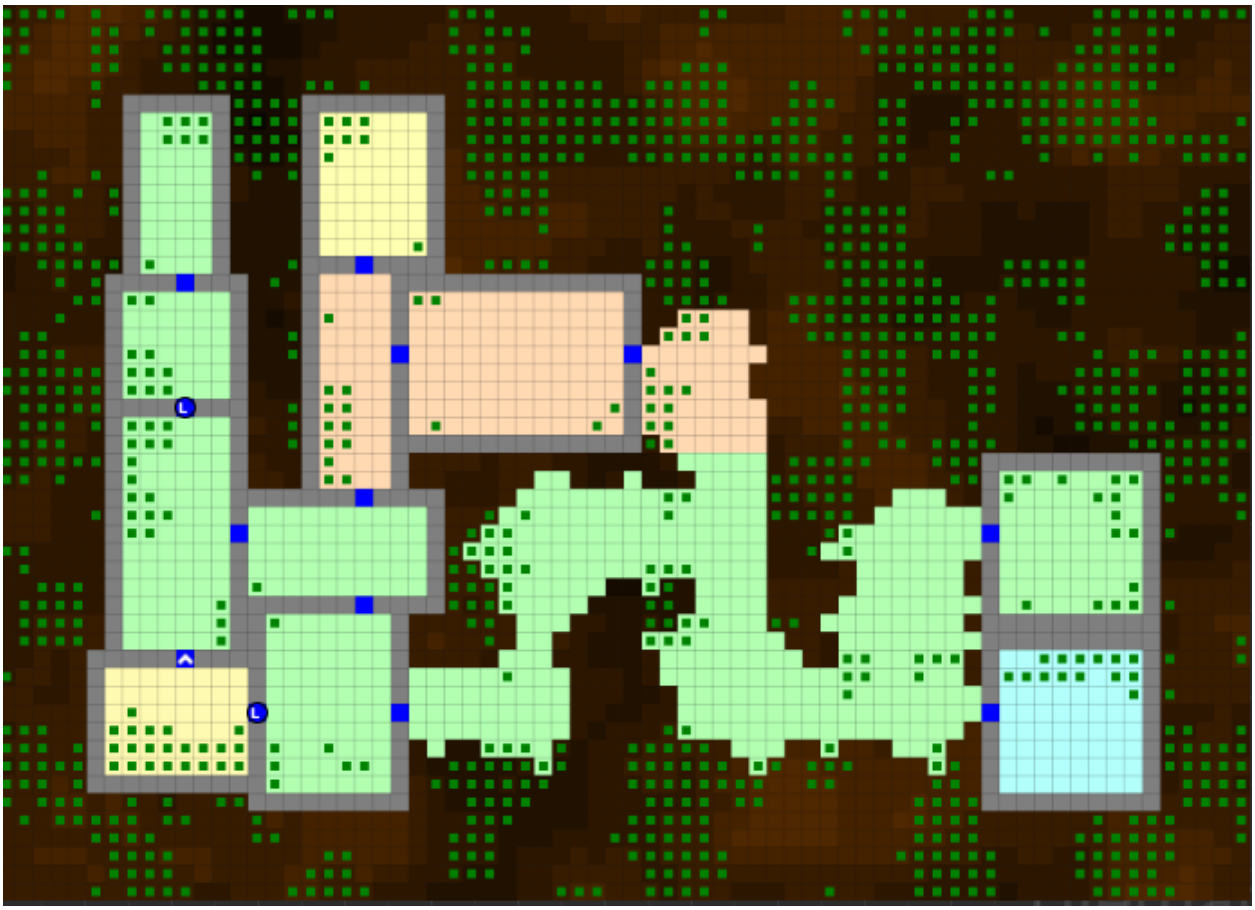
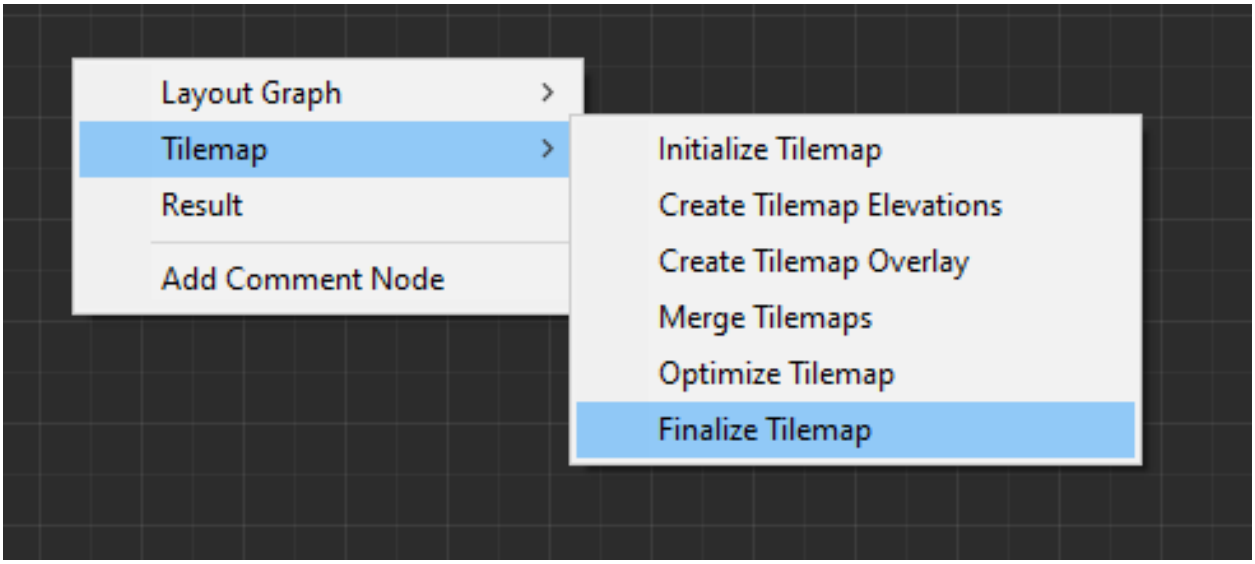


Fig. 14: Merged result

### 4.2.18 Finalize Tilemap

Finalize the tilemap to complete the grid flow graph

Create a node `Tilemap > Finalize Tilemap`





Finalize Tilemap node places all the items on to the tilemap (enemies, keys, bonus etc)

### 4.2.19 Build Dungeon

Assign this grid flow graph to your DungeonGridFlow game object and click *Build Dungeon*

### 4.2.20 Optimize Tilemap

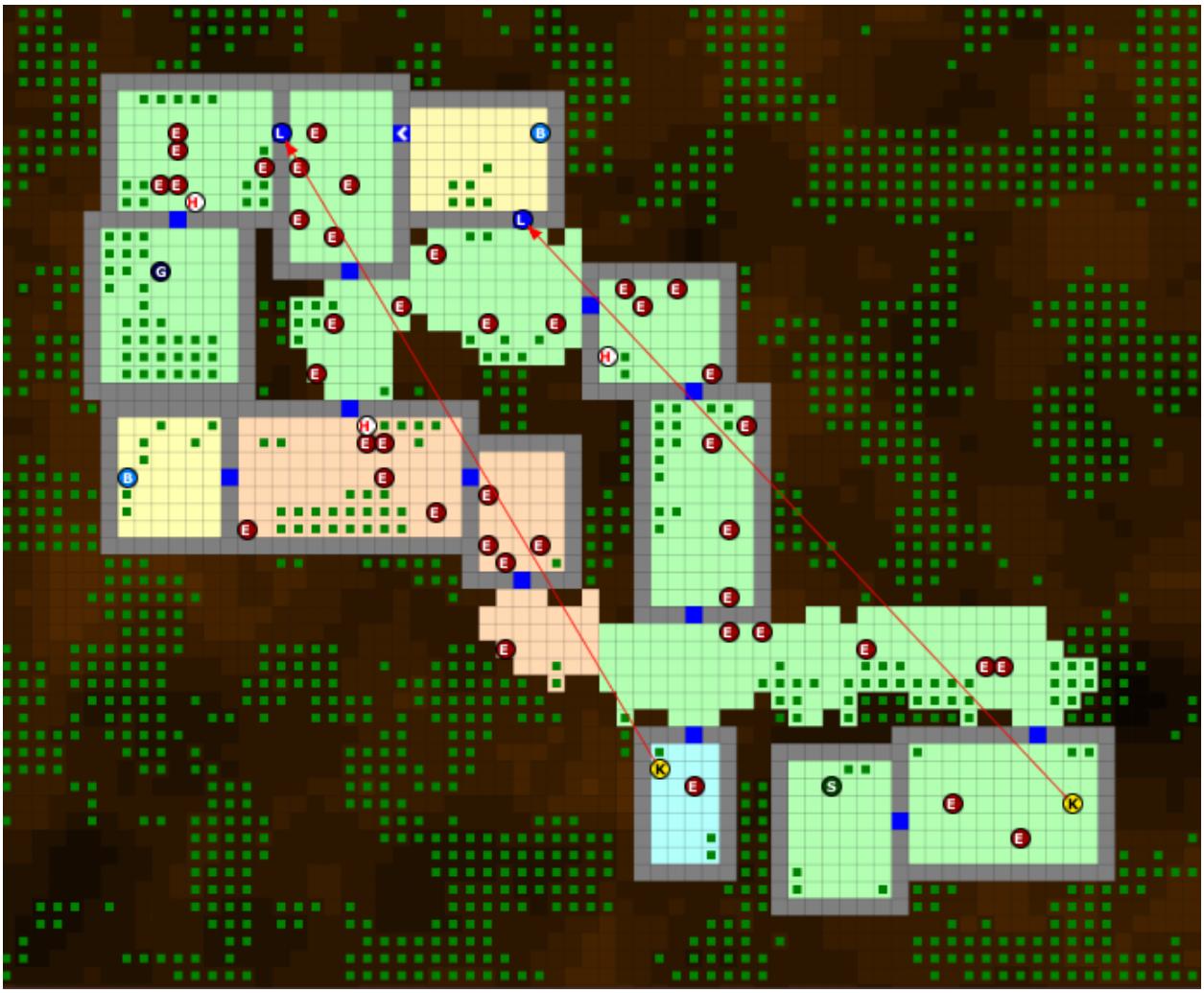
When the tilemap based level is generated, there are many tiles that the player might never see, as they are far away from the dungeon layout

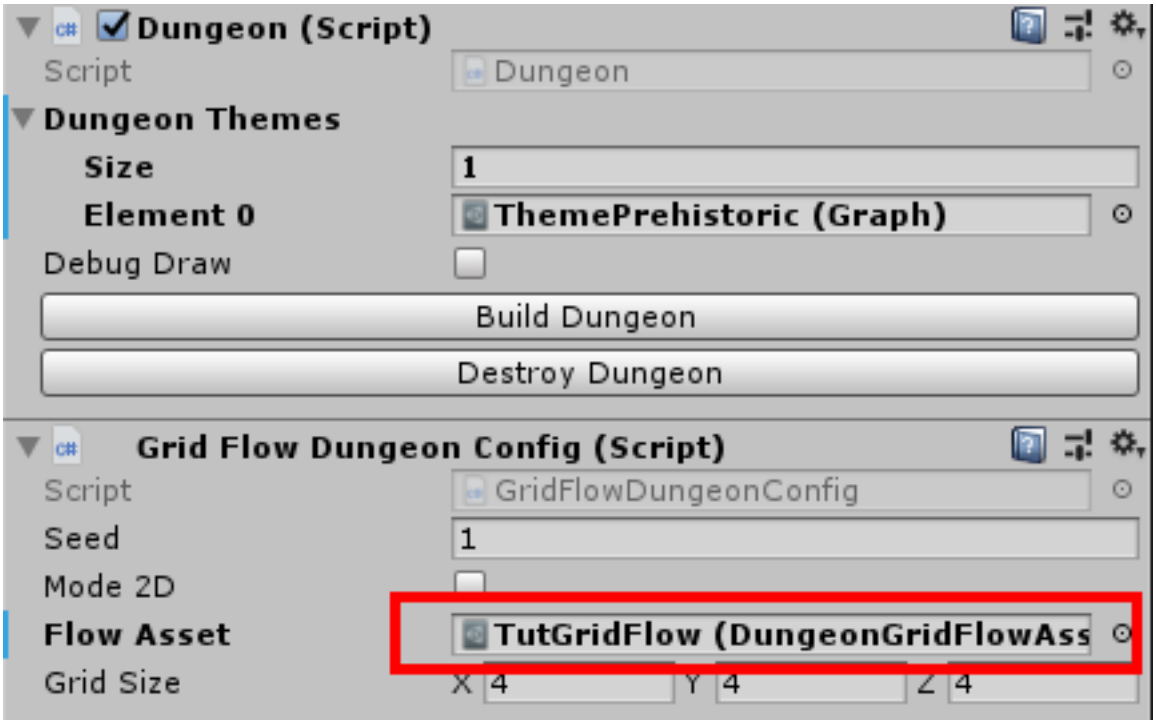
The *Optimize Tilemap* removes tiles that are away from the specified distance from the dungeon layout bounds

Create a node `Tilemap > Optimize Tilemap`

Connect it before the *Finalize Tilemap* node like below:

Rebuild the dungeon in the scene view




Dungeon (Script)
Script
Dungeon
Dungeon Themes
Size
1
Element 0
ThemePrehistoric (Graph)
Debug Draw
Build Dungeon
Destroy Dungeon
Grid Flow Dungeon Config (Script)
Script
GridFlowDungeonConfig
Seed
1
Mode 2D
Flow Asset
TutGridFlow (DungeonGridFlowAss
Grid Size
X 4
Y 4
Z 4

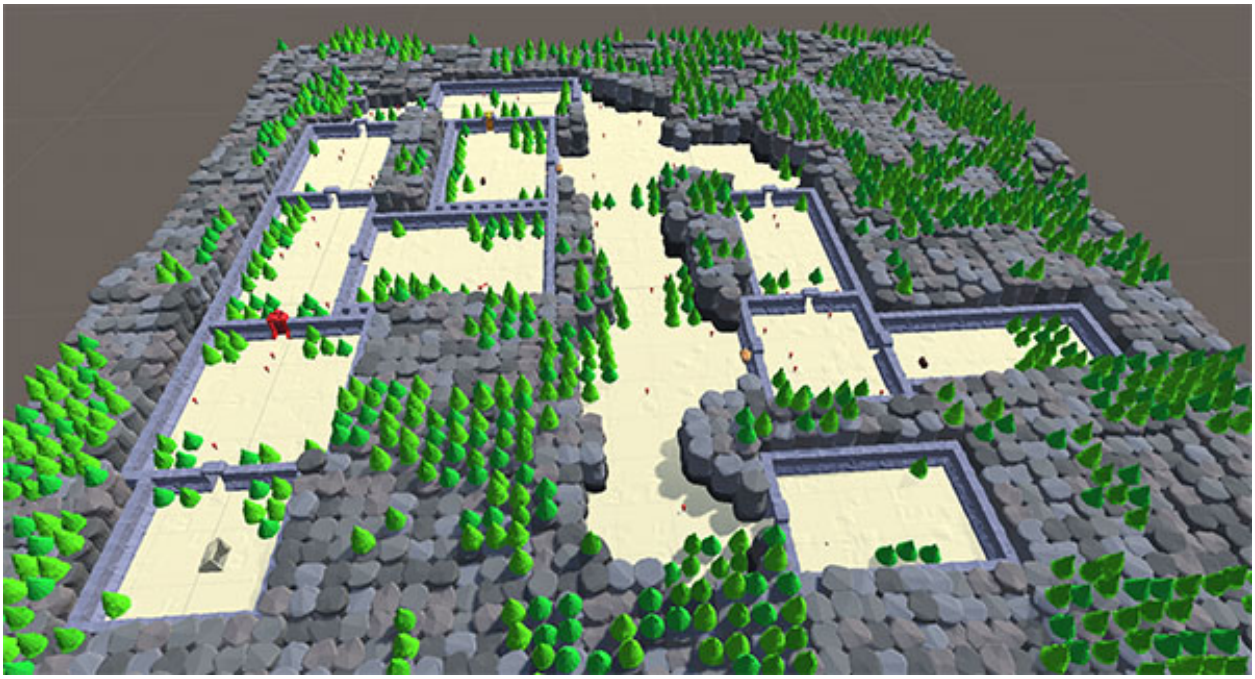
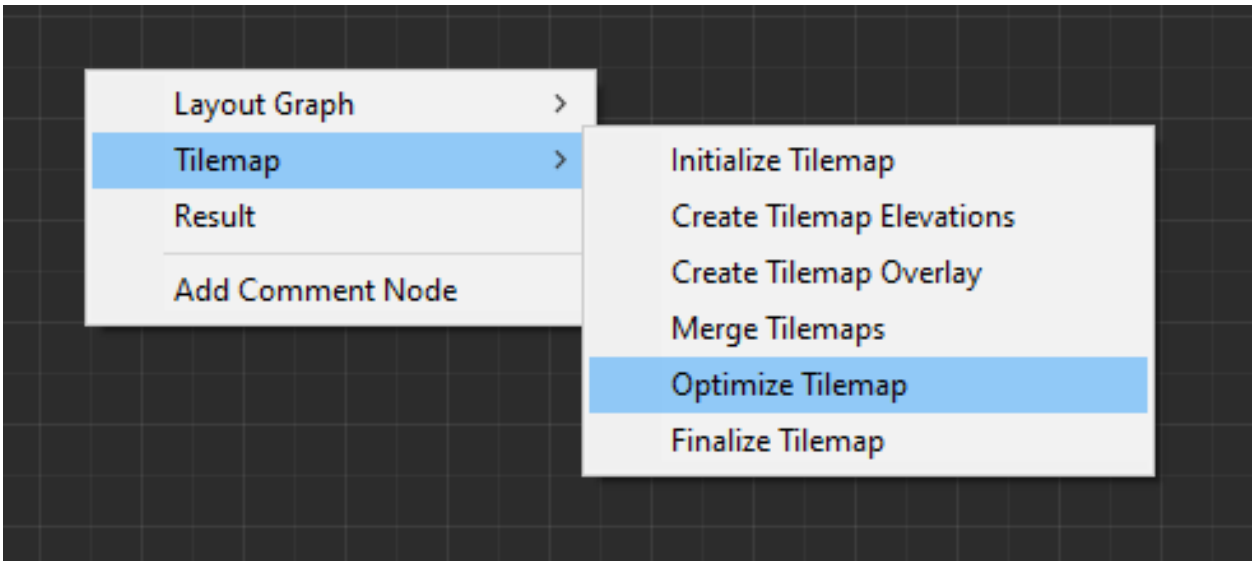


Fig. 15: Optimize Tilemap Node Result

Fig. 16: Optimize Tilemap Before / After





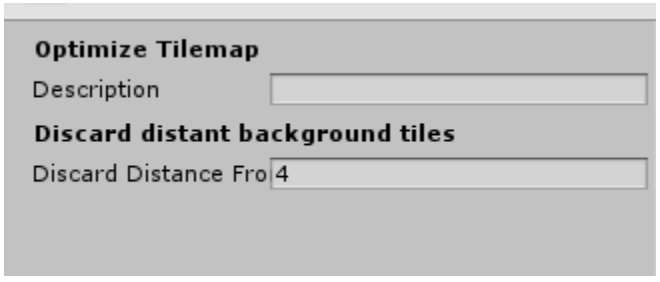



Fig. 17: Optimize Tilemap Node properties

Fig. 18: Optimize Tilemap Before / After

## 4.3 Key Lock System

The spawned Key and Lock game objects will have the following components attached to it by Dungeon Architect

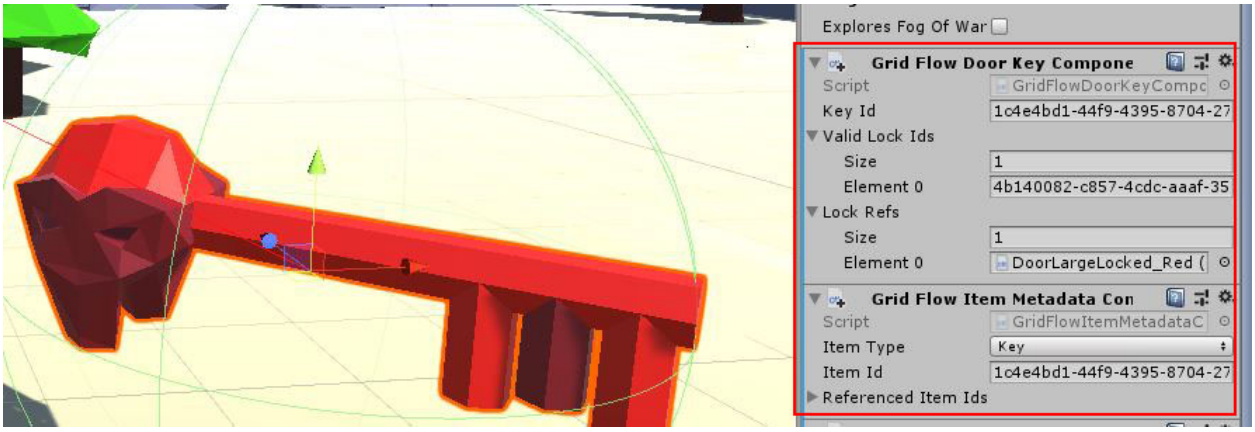



Fig. 19: New Components attached to the Key Prefab

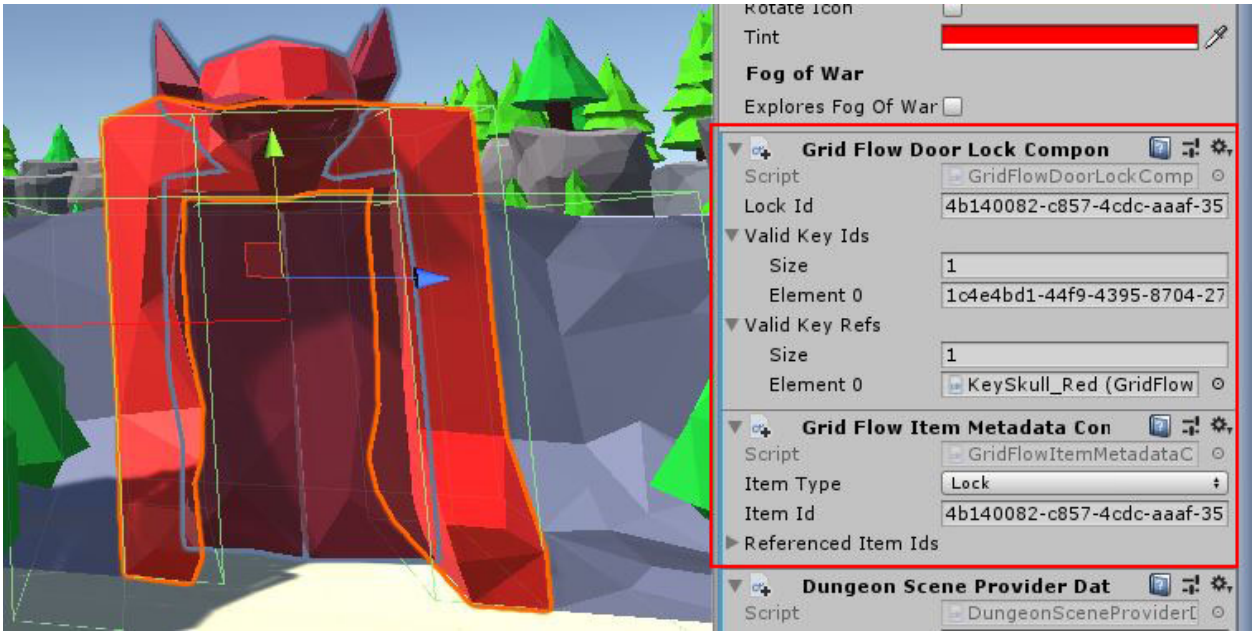


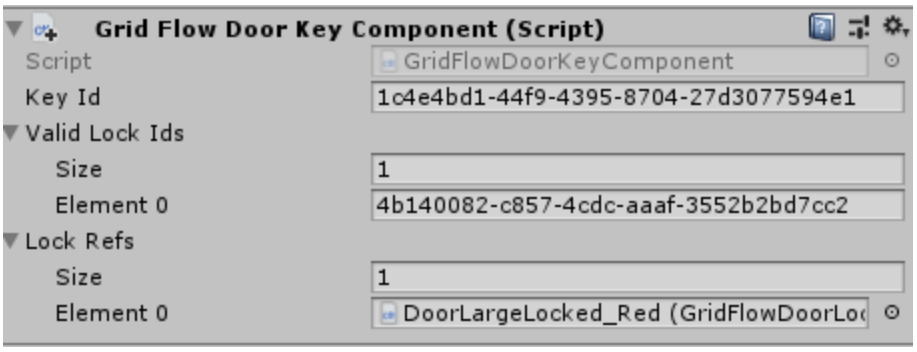

Fig. 20: New Components attached to the Locked Door Prefab

### 4.3.1 Key Component

The builder will attach a new component `GridFlowDoorKeyComponent` to the spawned key prefab

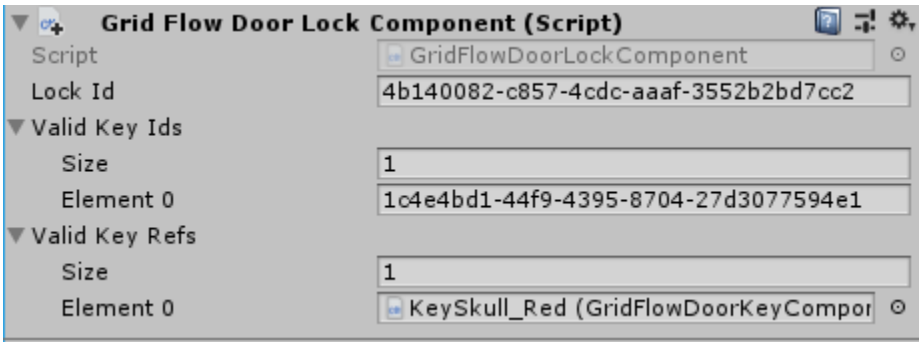
This component contains the KeyId and a reference to all the locks that this key can open

| | |
|---|---|
| **Key Id** | The Key Id |
| **Valid Lock Ids** | List of Lock Ids that can be opened by this key |
| **Lock Refs** | References to the spawned lock game objects that can be opened by this key |




### 4.3.2 Lock Component

The builder will attach a new component `GridFlowDoorLockComponent` to the spawned lock prefab




This component contains the LockId and a reference to all the keys that open this lock

| | |
|---|---|
| **Lock Id** | The Lock Id |
| **Valid Key Ids** | List of Key Ids that open this lock |
| **Valid Key Refs** | References to the spawned key game objects that open this lock |

### 4.3.3 Sample

Game Sample Scene: `Assets/DungeonArchitect_Samples/DemoBuilder_GridFlow/Scenes/GridFlowBuilderDemo_Game`

The GridFlow game sample contains a working example of how you can implement a key lock system. There are many ways of implementing this, this sample shows one such way.

The Sample has the following scripts:

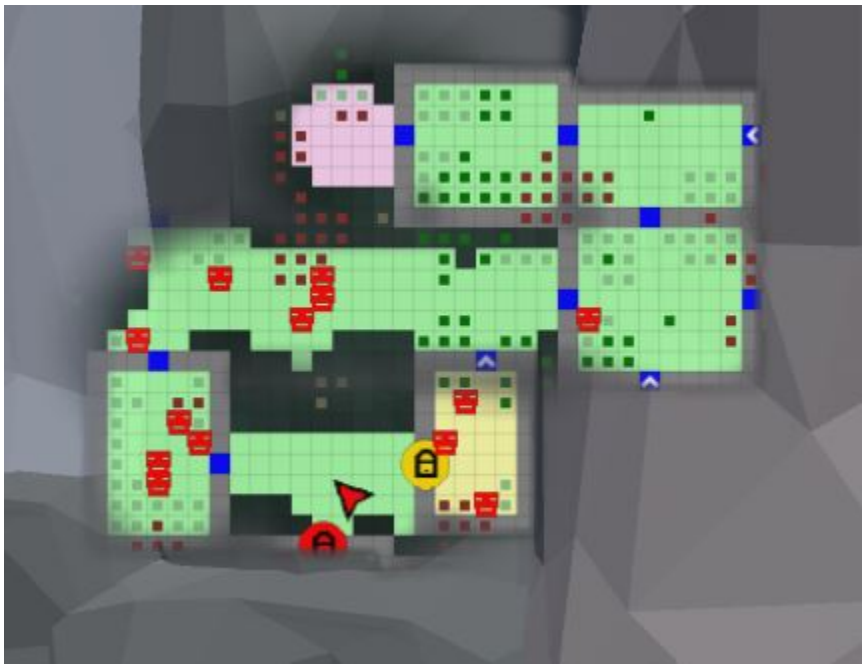
- Inventory: Saves the picked up keys in the inventory
- LockedDoor: A script that implements the door opening logic. This script is added to the locked door prefab. When something collides with the door trigger, it checks if it has an inventory. If it does, it checks if the inventory contains any of the valid keys that can open this door

LockedDoor script location: `Assets/DungeonArchitect_Samples/DemoBuilder_GridFlow/Scripts/DemoGame/Door/LockedDoor.cs`

```
bool CanOpenDoor(Collider other)
{
        var inventory = other.gameObject.GetComponentInChildren<Inventory>();
        if (inventory != null)
        {
                // Check if any of the valid keys are present in the inventory of the␣
↪collided object
                foreach (var validKey in validKeys)
                {
                        if (inventory.ContainsItem(validKey))
                        {
                                return true;
                        }
                }
        }
        return false;
}
```

## 4.4 Mini-Map

Display a 2D minimap with fog of war



The `DungeonGridFlow` prefab already comes pre-configured with the minimap. This is done with the `GridFlowMinimap` component:


| | |
|---|---|
| **Update Frequency** | Control the frequency of minimap updates. The updates can run at a lower fps for better performance |
| **Enable Fog of War** | Hides parts of the map that is not explored yet |
| **See Through Walls** | If this is disabled, unexplored area behind a wall will not be made visible. This works if Fog of War is enabled |
| **Minimap Texture** | The Render Target texture that the minimap will be rendered on |
| **Icons** | The icons to overlay on special tiles |
| **Init Mode** | |
| >> On Dungeon Rebuild | The minimap layout texture is regenerated when the dungeon rebuilds |
| >> On Play | The minimap layout texture is generated when you start play |
| >> Manual | The minimap layout texture is generated only when you manually build it from script |

### 4.4.1 Setup

The minimap requires you to provide a Render Texture asset in the *Minimap Texture* property. The minimap will be rendered in this texture. You can then apply this texture anywhere (in your UI elements, in a mesh etc)

Create a new Render Texture asset. Use the Create menu in the Project window: `Create > Render Texture`

Select the Render Texture asset and inspect the properties

Change the following:

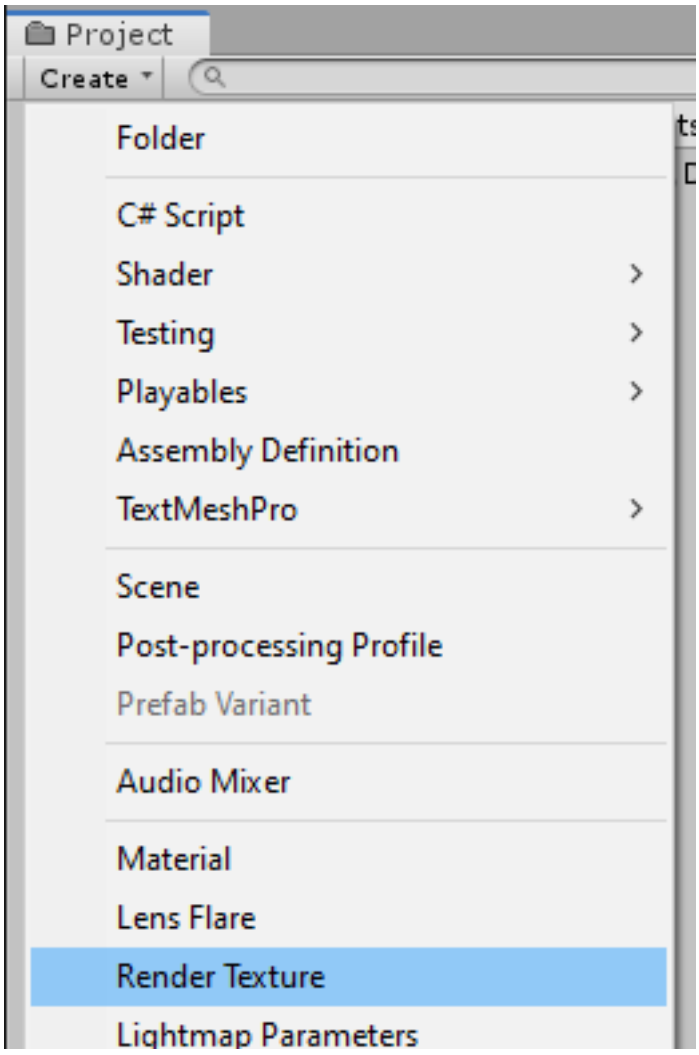
| | |
|---|---|
| **Size** | Change to 512x512 (or the quality you are comfortable with) |
| **Depth Buffer** | No Depth buffer (we don't need it here) |
| **Filter Mode** | Point (so we get sharp tile edges instead of a blurry image) |

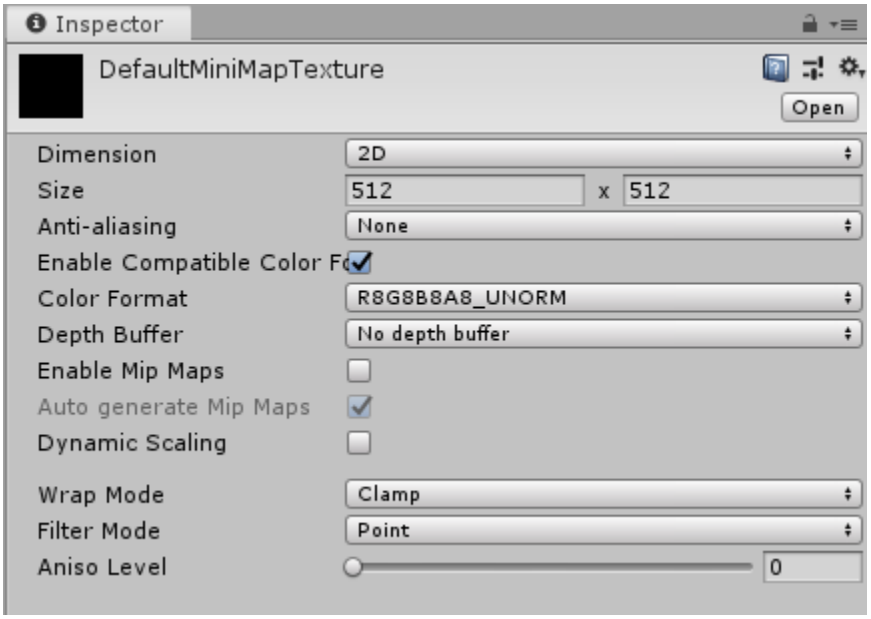
Assign this *Render Texture* asset to your DungeonGridFlow game object's minimap component

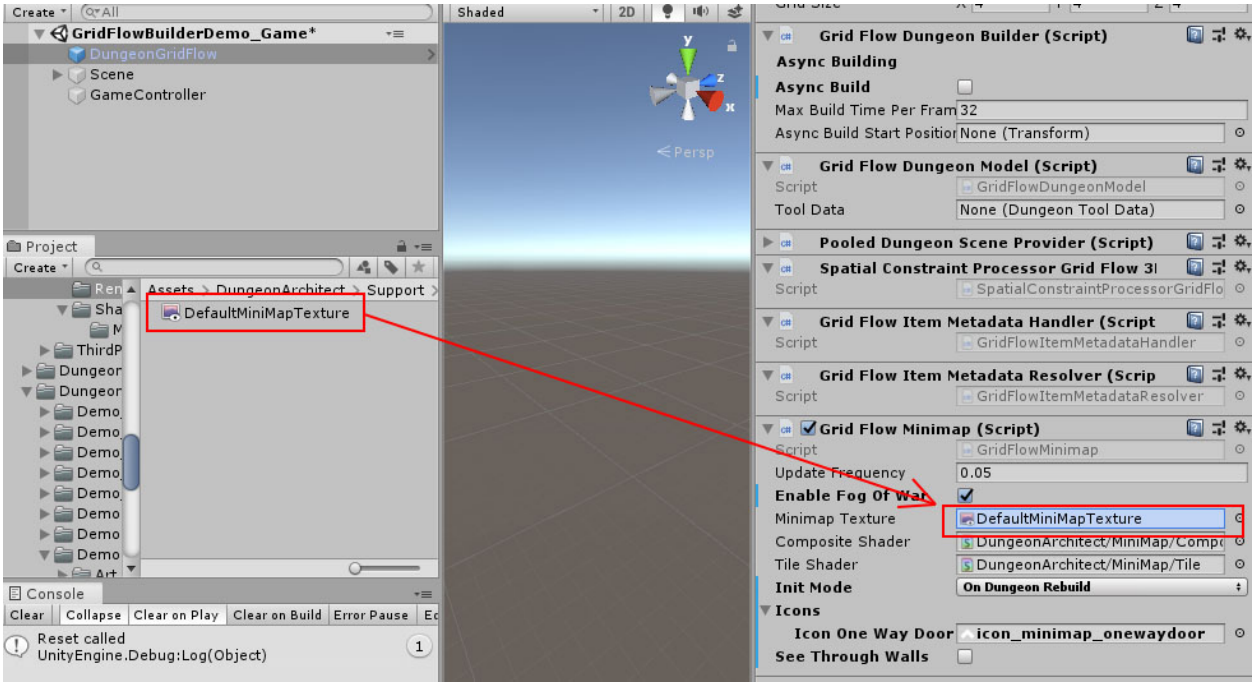
Dungeon Architect will automatically update this texture based on the specified *Update Frequency*. You can assign this texture anywhere on your UI. You can also attach it on a mesh

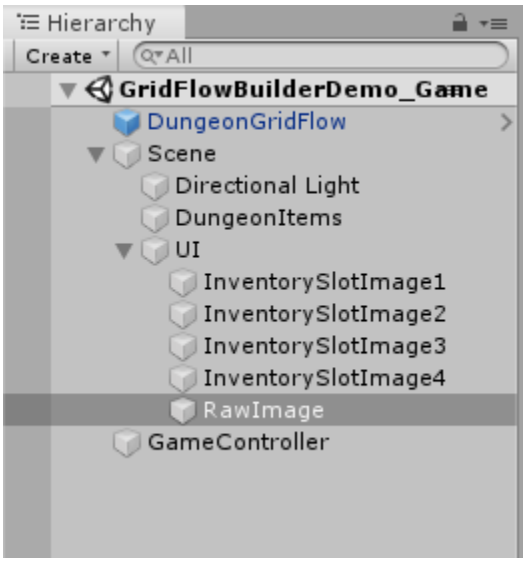
### 4.4.2 Show in UI

Open the game sample scene: `Assets/DungeonArchitect_Samples/DemoBuilder_GridFlow/Scenes/GridFlowBuilderDemo_Game`


Project
Create
Folder
C# Script
Shader
Testing
Playables
Assembly Definition
TextMeshPro
Scene
Post-processing Profile
Prefab Variant
Audio Mixer
Material
Lens Flare
Render Texture
Lightmap Parameters

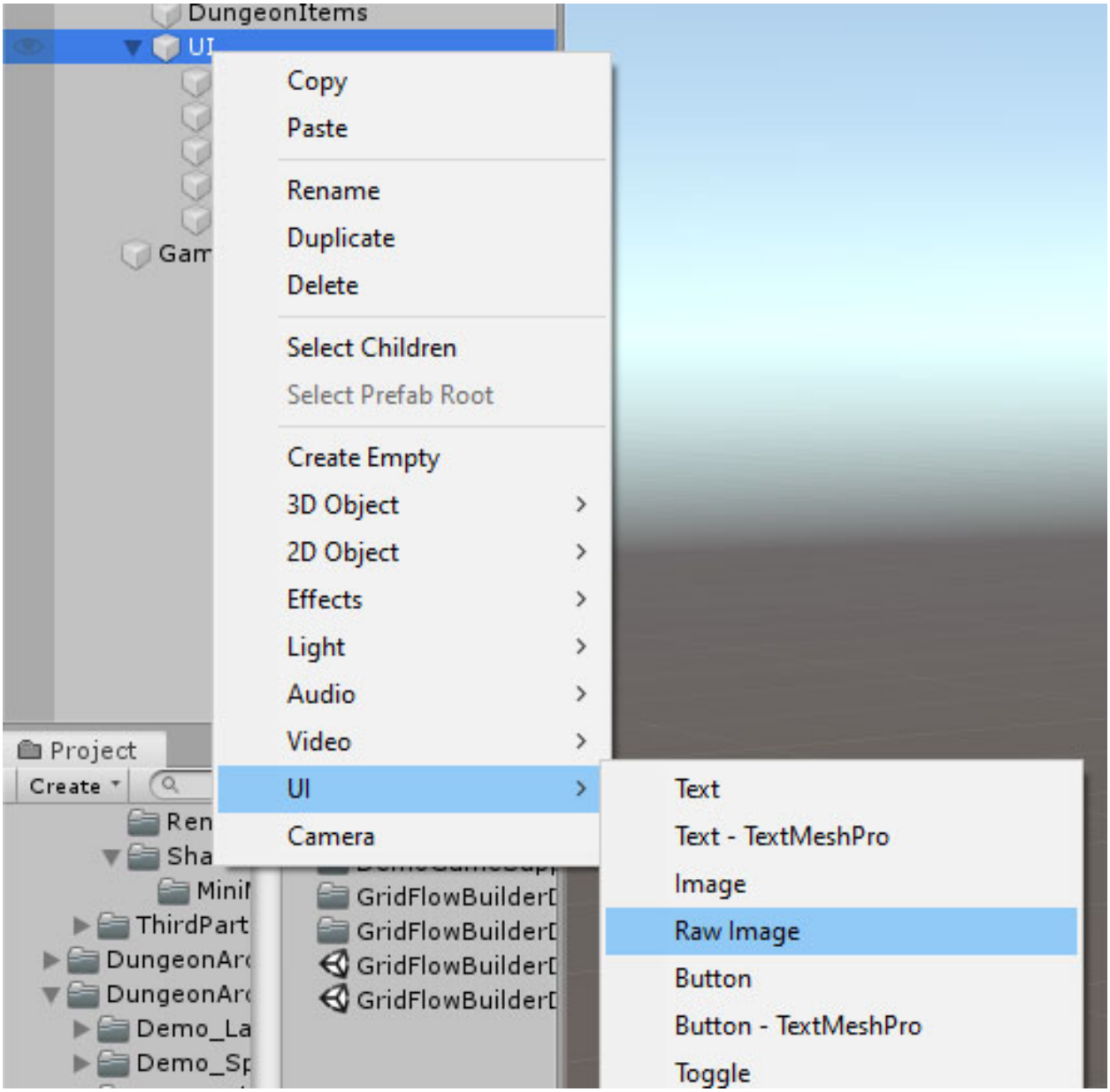

Inspector
DefaultMiniMapTexture
Open
Dimension
2D
Size
512
x
512
Anti-aliasing
None
Enable Compatible Color F
Color Format
R8G8B8A8_UNORM
Depth Buffer
No depth buffer
Enable Mip Maps
Auto generate Mip Maps
Dynamic Scaling
Wrap Mode
Clamp
Filter Mode
Point
Aniso Level
0


Create
All
GridFlowBuilderDemo_Game*
DungeonGridFlow
Scene
GameController
Shaded
2D
Persp
Project
Create
Assets
DungeonArchitect
Support
DefaultMiniMapTexture
Console
Clear
Collapse
Clear on Play
Clear on Build
Error Pause
Reset called
UnityEngine.Debug:Log(Object)
Grid Flow Dungeon Builder (Script)
Async Building
Async Build
Max Build Time Per Fram
32
Async Build Start Positior
None (Transform)
Grid Flow Dungeon Model (Script)
Script
GridFlowDungeonModel
Tool Data
None (Dungeon Tool Data)
Pooled Dungeon Scene Provider (Script)
Spatial Constraint Processor Grid Flow 3I
Grid Flow Item Metadata Handler (Script
GridFlowItemMetadataHandler
Grid Flow Item Metadata Resolver (Scrip
GridFlowItemMetadataResolver
Grid Flow Minimap (Script)
GridFlowMinimap
Update Frequency
0.05
Enable Fog Of War
Minimap Texture
DefaultMiniMapTexture
Composite Shader
DungeonArchitect/MiniMap/Tile
Init Mode
On Dungeon Rebuild
Icons
Icon One Way Door
icon_minimap_onewaydoor
See Through Walls


Hierarchy
Create
All
GridFlowBuilderDemo_Game
DungeonGridFlow
Scene
Directional Light
DungeonItems
UI
InventorySlotImage1
InventorySlotImage2
InventorySlotImage3
InventorySlotImage4
RawImage
GameController

There's a UI canvas in the hierarchy. Expand and inspect it:

There is a *RawImage* Canvas Item in there. It was created like this:




Select the RawImage item and configure it like this:

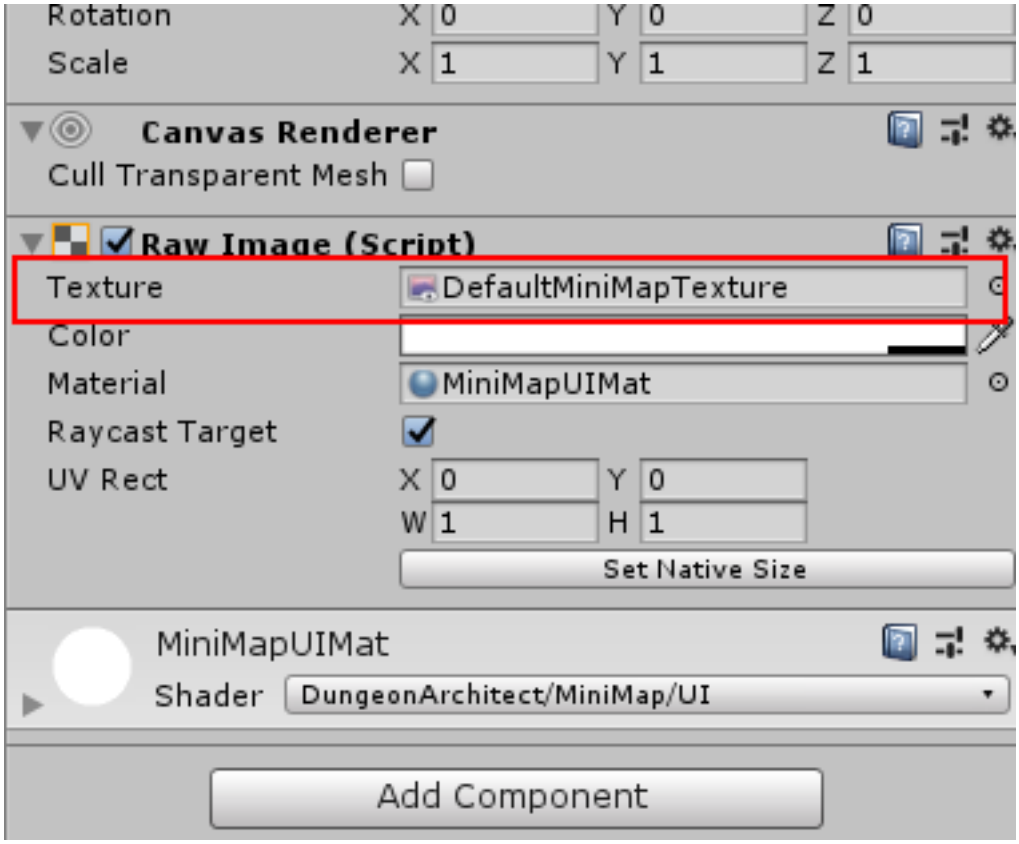
The Render Texture was assigned there so it will show our minimap

### 4.4.3 Add to a Material

While playing the sample game, if you look down, you notice the player holding a map in the hand (like in Minecraft). This map shows the minimap in realtime

The texture was simply added to an unlit material, and the material was then applied to that mesh

Create a Material as below:




- Set the Shader to `UI/Unlit/Transparent`
- Set the texture to your Render Texture asset

You can now apply this material anywhere (e.g. in a large billboard in your world, a small map that the player holds, dashboard of a vehicle etc)

**See also:**

Check the sample game to see how this was done

| | |
|---|---|
| **Tablet Prefab** | *Assets/DungeonArchitect_Samples/DemoBuilder_GridFlow/Art/Prefab/Tablet_Map* |
| **Material** | *Assets/DungeonArchitect_Samples/DemoBuilder_GridFlow/Art/Materials/Mat_TabletScreen* |

### 4.4.4 Minimap Tracked Objects

The minimap can track any object in the scene. You do this by adding the *GridFlowMinimapTrackedObject* component to the desired prefab

It has the following features:

- The tracked object can explore the minimap (e.g. player and allies)
- Specify an icon, color and scale of the object in the minimap
- the icon can rotate to indicate the game object's Y rotation (good for player game objects)

You'd want to turn on *Explores Fog of War* only for the player and other relavant objects. The icon can be greyscale and you can apply a tint on it with different colors (e.g. on key icon but different colors applied to the red key prefab,

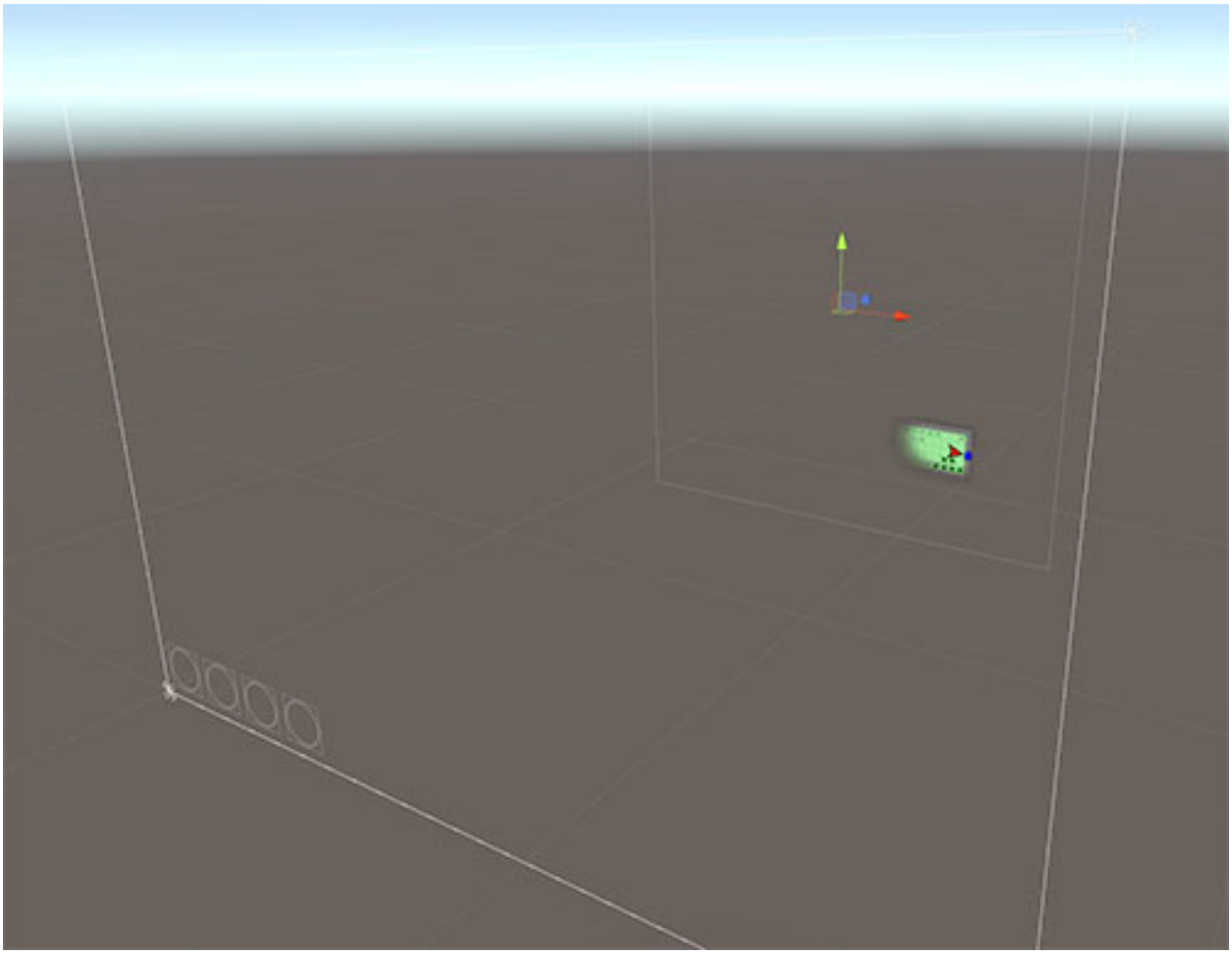



Fig. 21: Added to the Player prefab

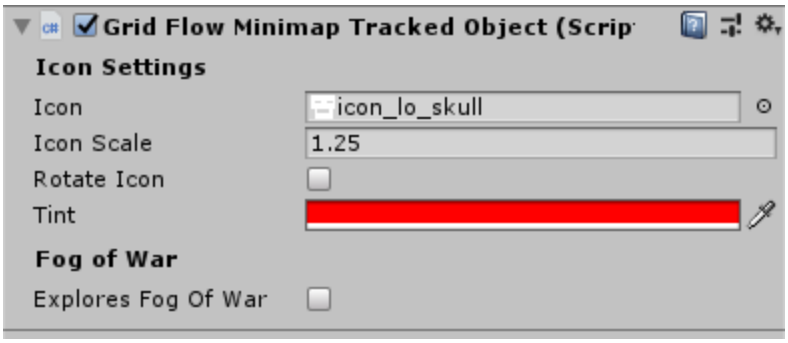



Fig. 22: Added to the Enemy prefab

blue key prefab and so on)

**See also:**

Check the sample game prefabs to see how the component was configured

| | |
|---|---|
| Player Controller | Assets/DungeonArchitect_Samples/DemoBuilder_GridFlow/Scenes/DemoGameSupportFiles/Prefabs/GridFlowPlaye |
| Grund NPC | Assets/DungeonArchitect_Samples/DemoBuilder_GridFlow/Scenes/DemoGameSupportFiles/Prefabs/Enemy |
| Key (Red) | Assets/DungeonArchitect_Samples/DemoBuilder_GridFlow/Art/Prefab/KeySkull_Red |
| Key (Blue) | Assets/DungeonArchitect_Samples/DemoBuilder_GridFlow/Art/Prefab/KeySkull_Blue |
| Door (Yellow) | Assets/DungeonArchitect_Samples/DemoBuilder_GridFlow/Art/Prefab/DoorLargeLocked_Yellow |
| Door (Green) | Assets/DungeonArchitect_Samples/DemoBuilder_GridFlow/Art/Prefab/DoorLargeLocked_Green |

CHAPTER

FIVE

# SNAP FLOW BUILDER

The Snap Builder generates a dungeon by stitching together pre-built rooms prefabs. The rules for stitching them is controlled by Graph Grammars

In this page, we'll walk through the creation process.

## 5.1 Preparing the Scene

Create a new scene and drop in a Dungeon Snap game object. This will allow you to build snap dungeons

## 5.2 Creating Module Prefabs

A module is a prebuilt prefab of a area (like room, corridor etc)

Design a room in the editor so we can turn it into a prefab for use with the snap builder

We've left holes at places where we want a possible door. We'll place a special Connection prefab later at these places to let Dungeon Architect know that it can stitch the rooms from these points

Go ahead and create a few more module prefabs. We've created one for a corridor below

## 5.3 Connections

A Snap Connection tells DA how to stitch the room modules together. They are usually the Door Entry / Exits.

The connection also contains references to two assets, a references to a Door prefab and a reference to a Wall prefab.

If DA stitches another module through that connection point, it would place the specified Door prefab in that place. Otherwise it would fill up the gap with the specified Wall prefab

Design a new Connection prefab by creating an emtpy Game Object

Reset the transform of the newly created emtpy Game Object and rename it

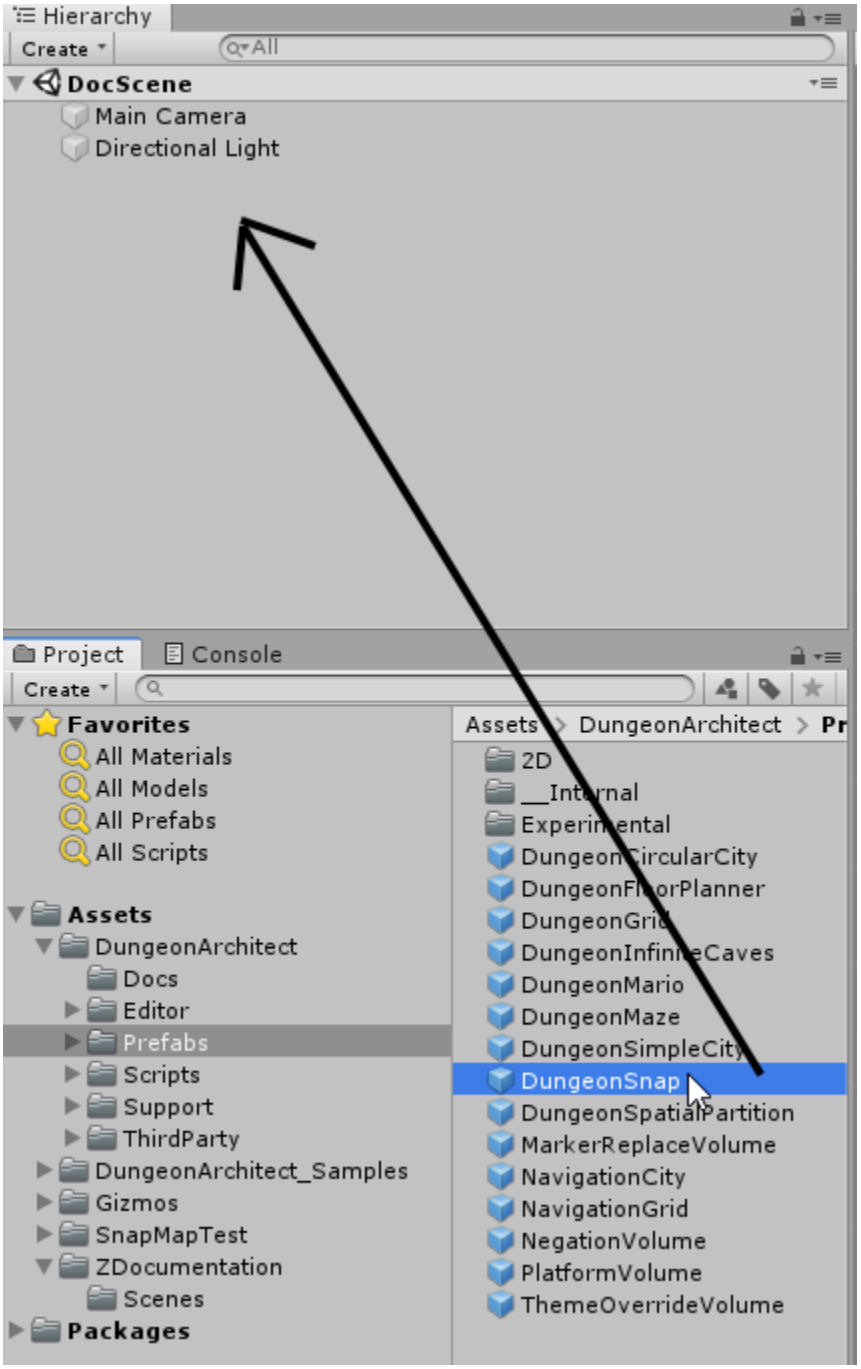
Add a SnapConnection script to it

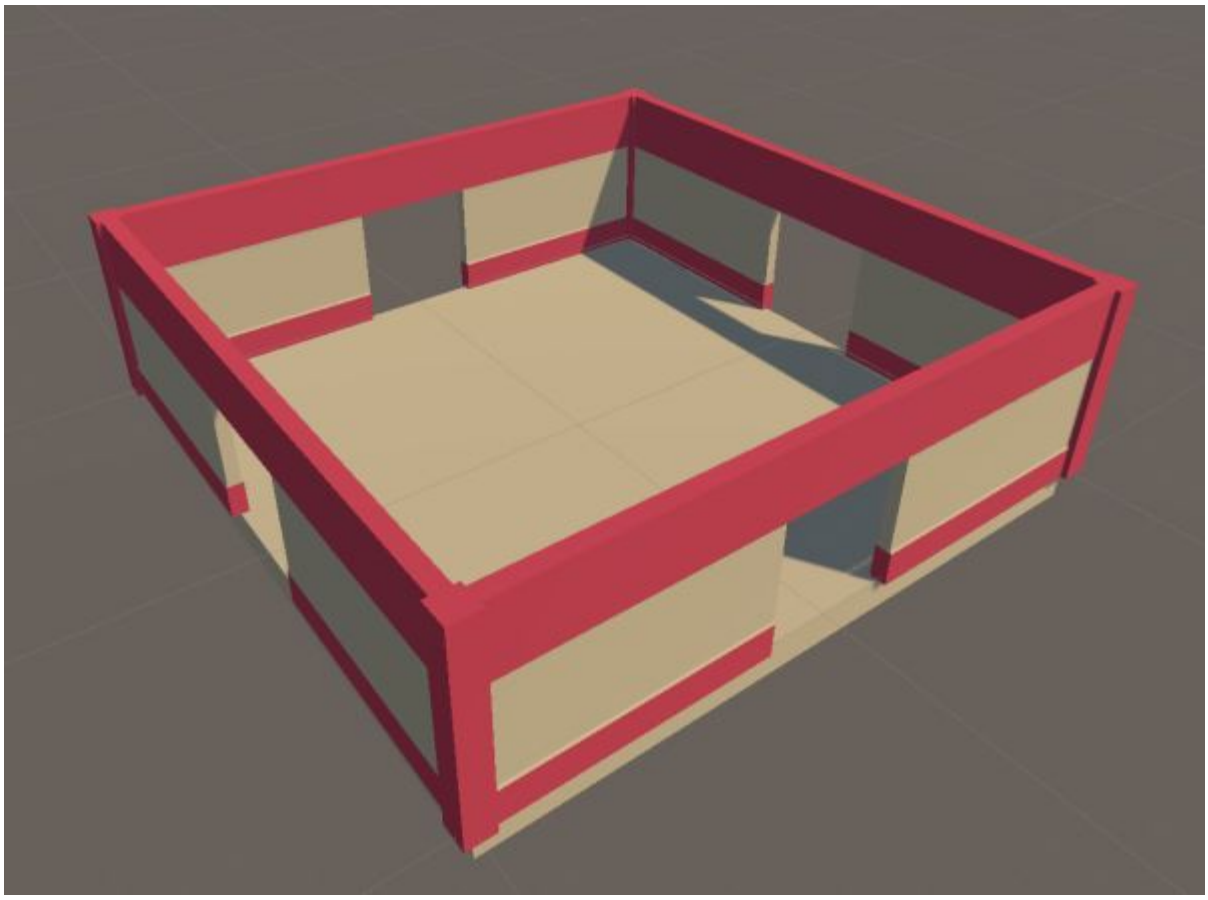
This script takes references to the above two mentioned prefab references, one for door and another for wall

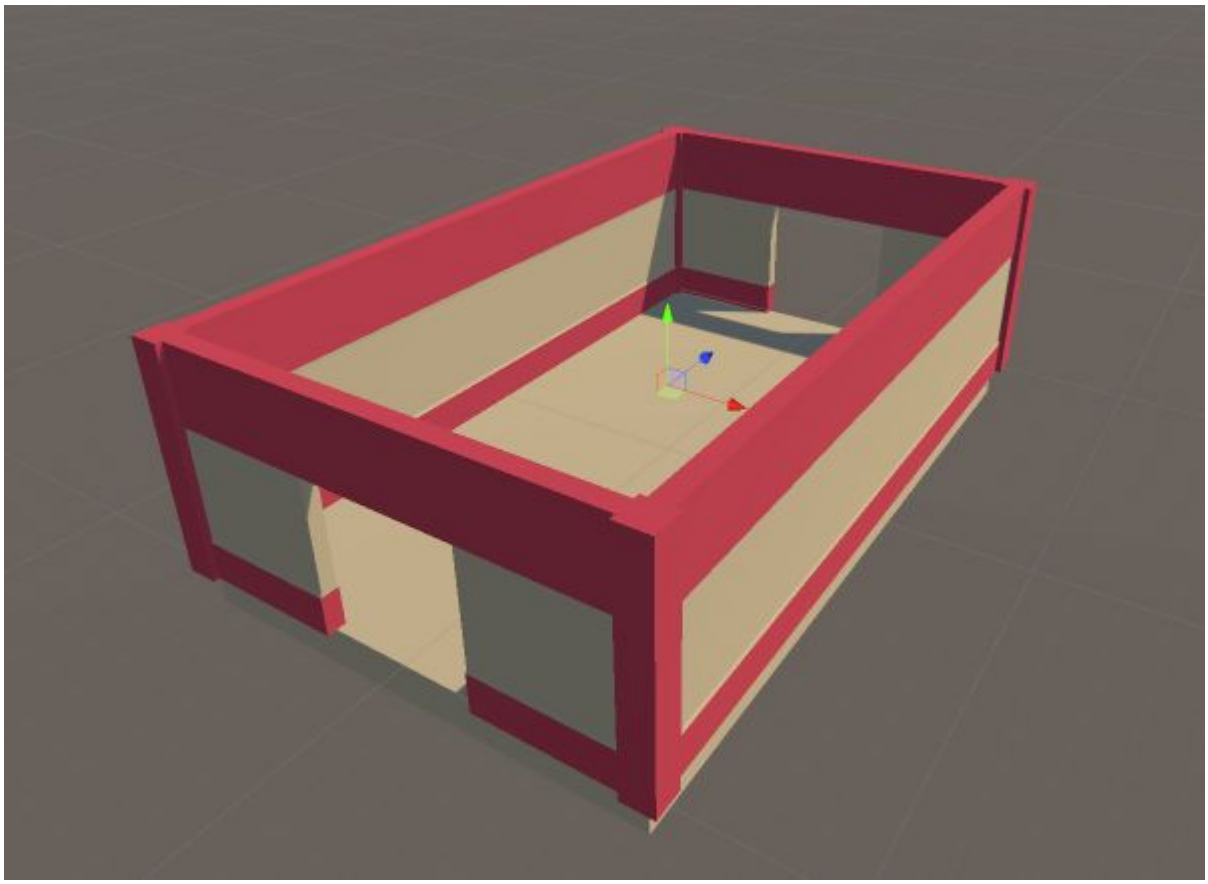
Drop your Door and Wall prefabs under the Connection prefab

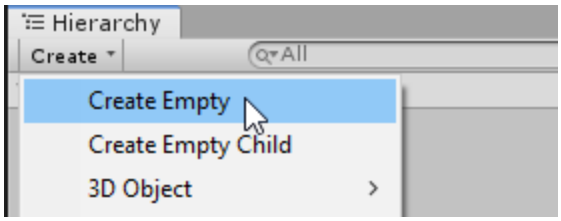
Align the door and wall prefabs so the Red line is perpendicular to it

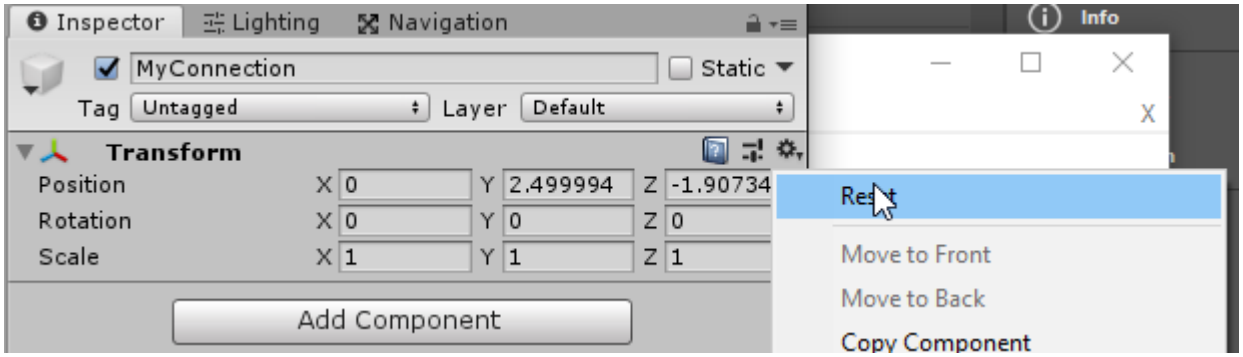
Now set these child door / wall prefab references on the SnapConnection script

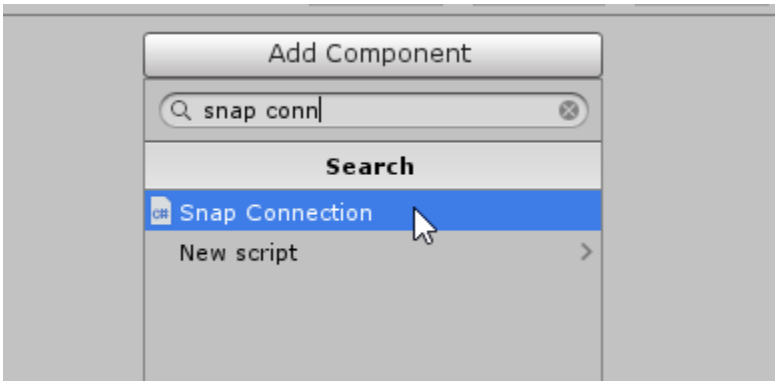

Hierarchy
Create
All
DocScene
Main Camera
Directional Light
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
DungeonArchitect
Docs
Editor
Prefabs
Scripts
Support
ThirdParty
DungeonArchitect_Samples
Gizmos
SnapMapTest
ZDocumentation
Scenes
Packages
Assets > DungeonArchitect > Pr
2D
__Internal
Experimental
DungeonCircularCity
DungeonFloorPlanner
DungeonGrid
DungeonInfiniteCaves
DungeonMario
DungeonMaze
DungeonSimpleCity
DungeonSnap
DungeonSpatialPartition
MarkerReplaceVolume
NavigationCity
NavigationGrid
NegationVolume
PlatformVolume
ThemeOverrideVolume

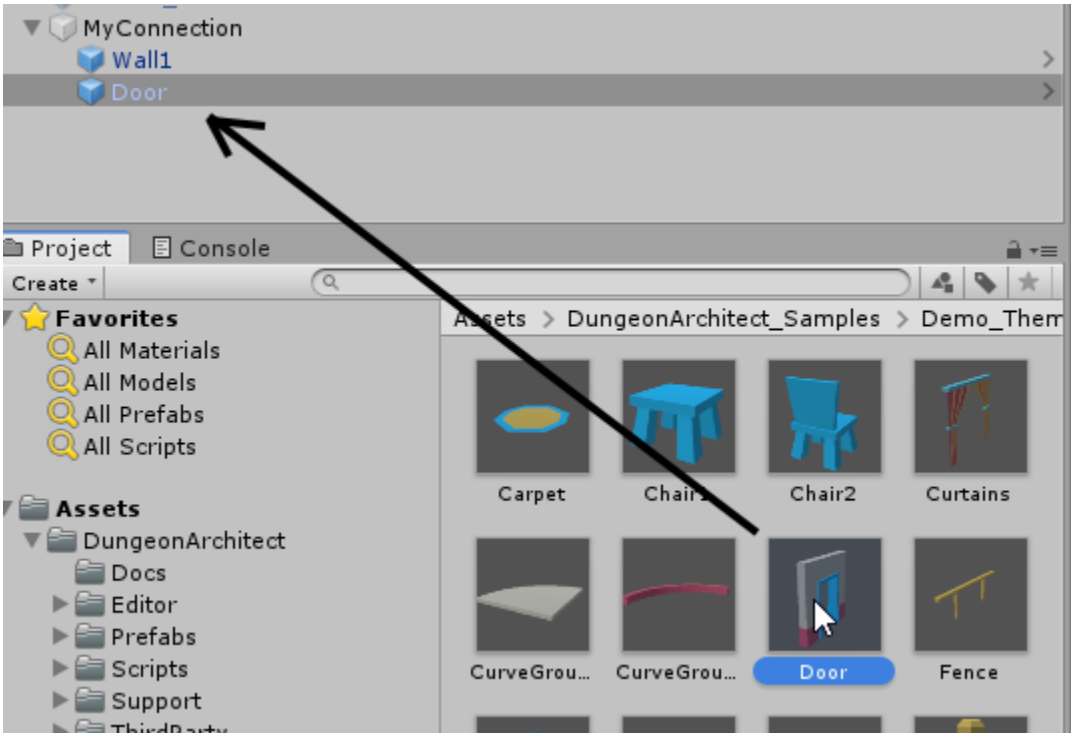


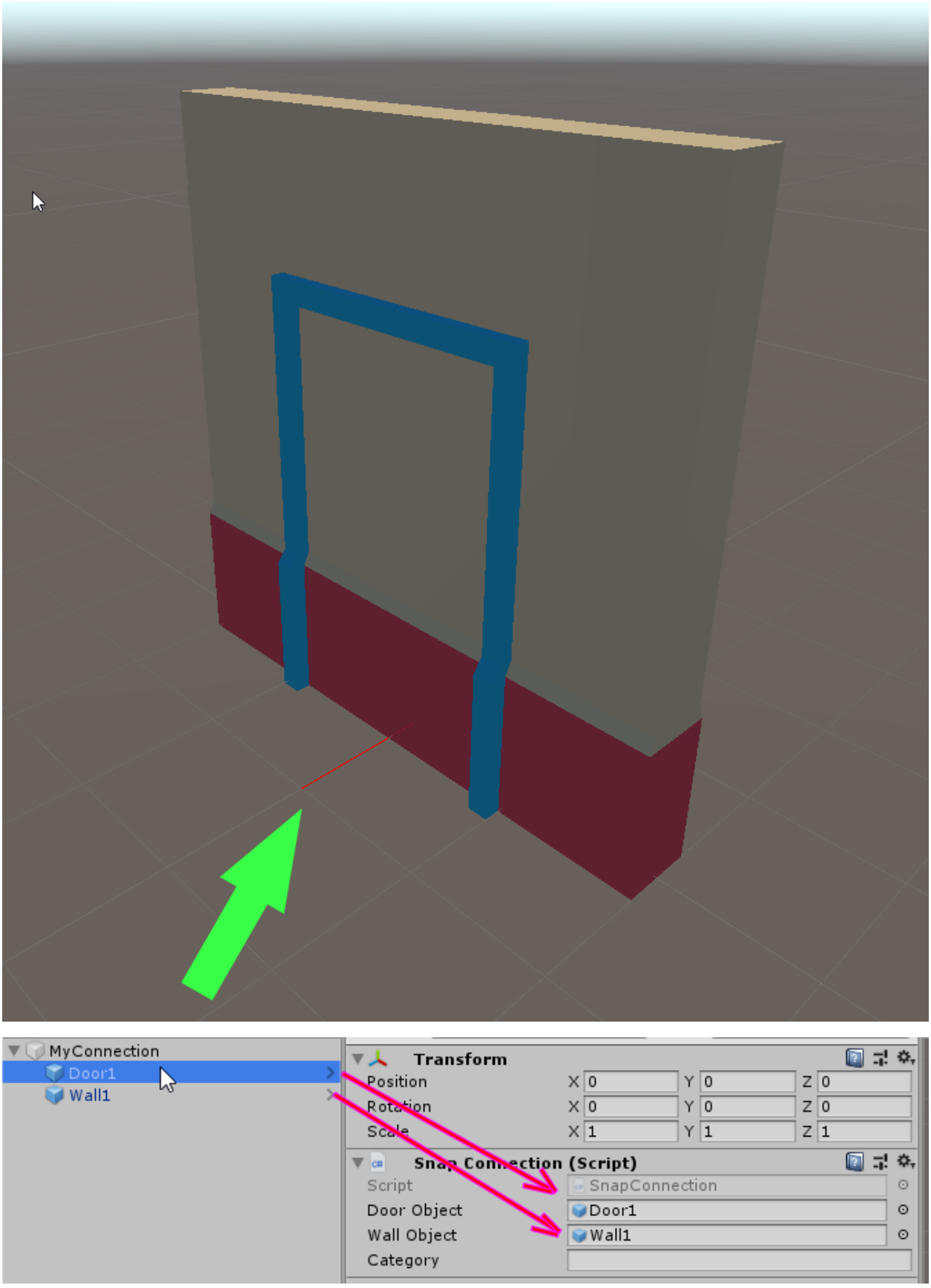



Hierarchy
Create
All
Create Empty
Create Empty Child
3D Object

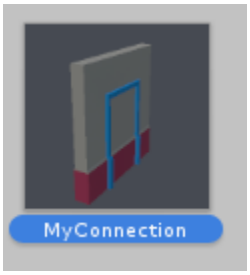

Inspector
Lighting
Navigation
MyConnection
Static
Tag
Untagged
Layer
Default
Transform
Position
X 0
Y 2.499994
Z -1.90734
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Add Component
Info
Move to Front
Move to Back
Copy Component


Add Component
snap conn
Search
Snap Connection
New script

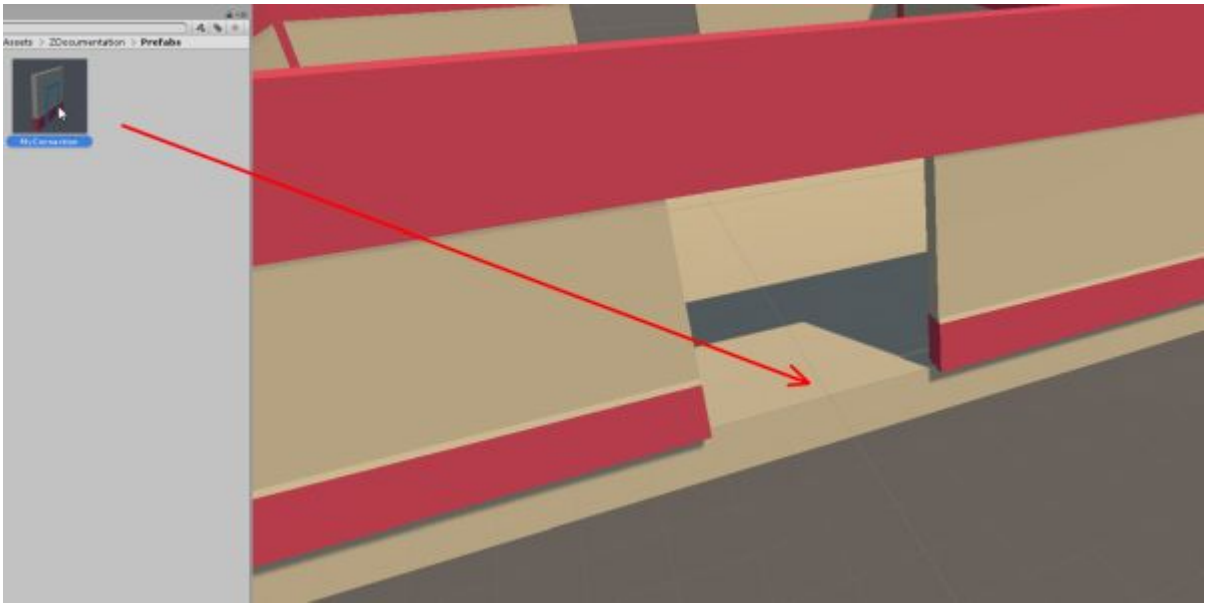

MyConnection
Wall1
Door
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
DungeonArchitect
Docs
Editor
Prefabs
Scripts
Support
DungeonArchitect_Samples
Demo_Them
Carpet
Chair2
Curtains
CurveGrou...
CurveGrou...
Door
Fence


MyConnection
Door1
Wall1
Transform
Position
Rotation
Scale
Script
SnapConnection
Door Object
Door1
Wall Object
Wall1
Category

Your connection is ready. Turn this into a prefab so we can reuse this in our modules




Drag and drop the connection prefab on your previously generated modules. Make sure the red line points outwards from the opening



Make sure the red line is pointing outwards and is on the edge of the module bounds

**Note:** It is a good practice to design with the snap settings (Edit > Snap Settings > Snap All Axis)

Repeat by drag-dropping on all the door openings. Do this for all the other modules as well (like the corridor module)

Save/Update your module prefab

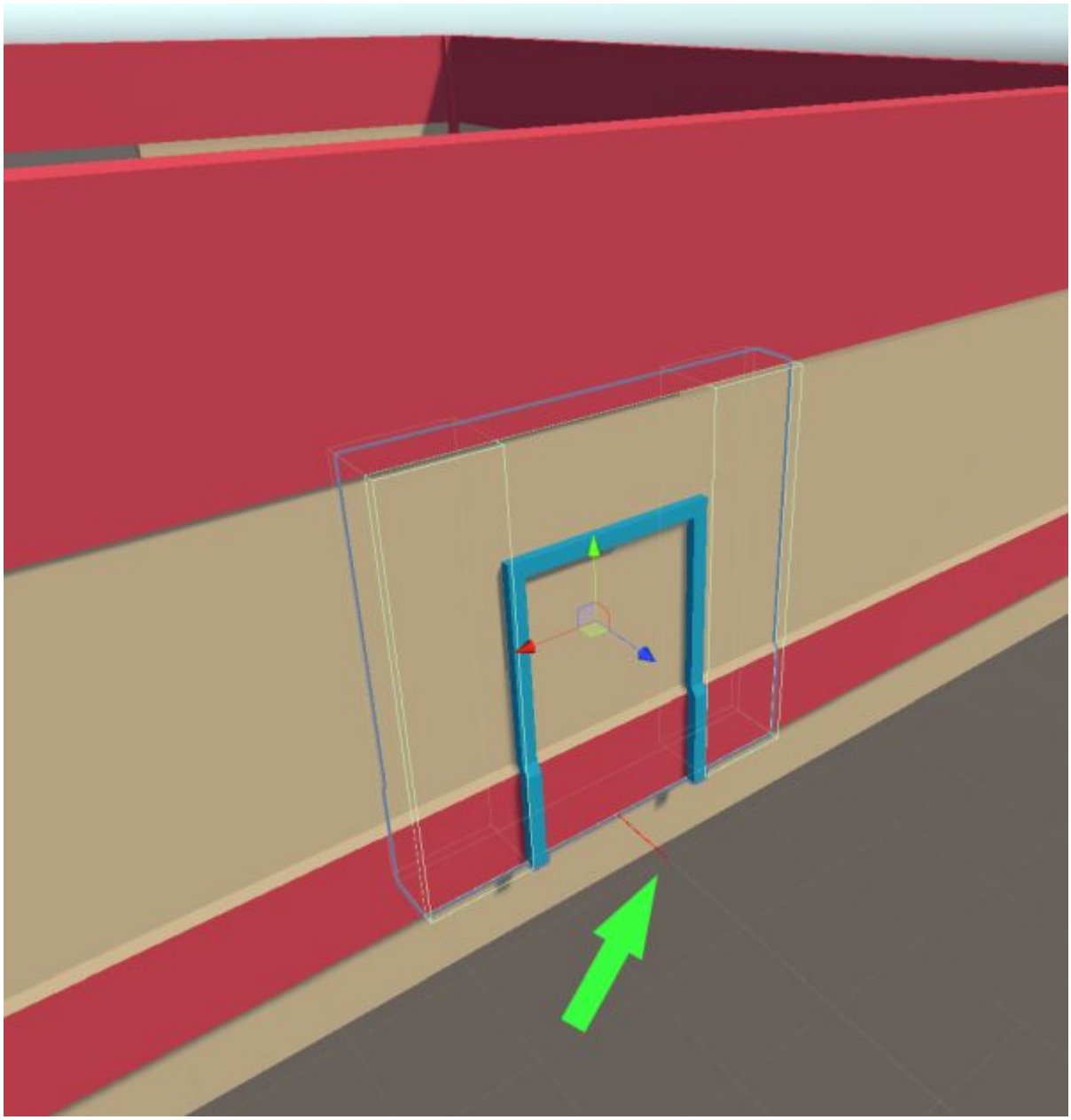
## 5.4 Register the Modules

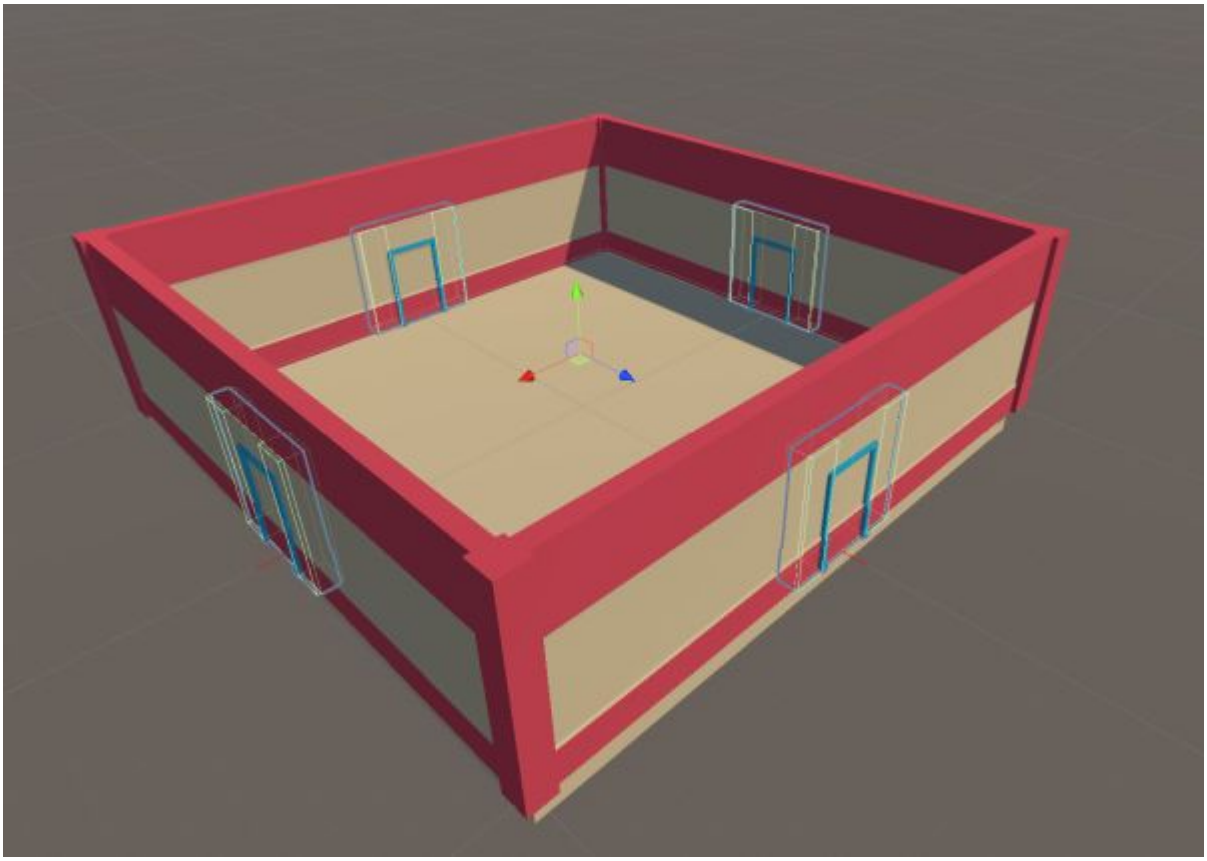
Register your modules in the DungeonSnap game object so Dungeon Architect can use it to build the dungeon

Inspect the DungeonSnap Game Object

Here we've registered the two modules and assigned a Category to them (e.g. Room, Corridor, TreasureRoom, Mini-Boss, MainBoss, SpawnRoom, Exit etc).

You can have multiple module prefabs assigned to the same category. These categories are used in the Dungeon Flow graph to design a procedural layout graph for your dungeon





## 5.5 Dungeon Flow

A Dungeon Flow graph allows you to control the layout of your dungeons using Graph Grammars. You can generate interesting graphs with simple rules

Create a new Dungeon Flow Asset by right clicking on the Projects window. (This can also be done from the Create menu in the Projects window or Assets > Create from the editor's main menu)

Double click the asset to open the Dungeon Flow Editor. Doc this window in a large area

Add two new nodes **Room** and **Corridor**. You can change the name of the nodes from the inspector window

These names map to the names you specified on the Module registration in the DungeonSnap game object

Select the *Start Rule* and on the RHS, delete the default T node and drop in a few Room nodes like this:

---

**Note:** Cycles are not supported by the SnapMap builder

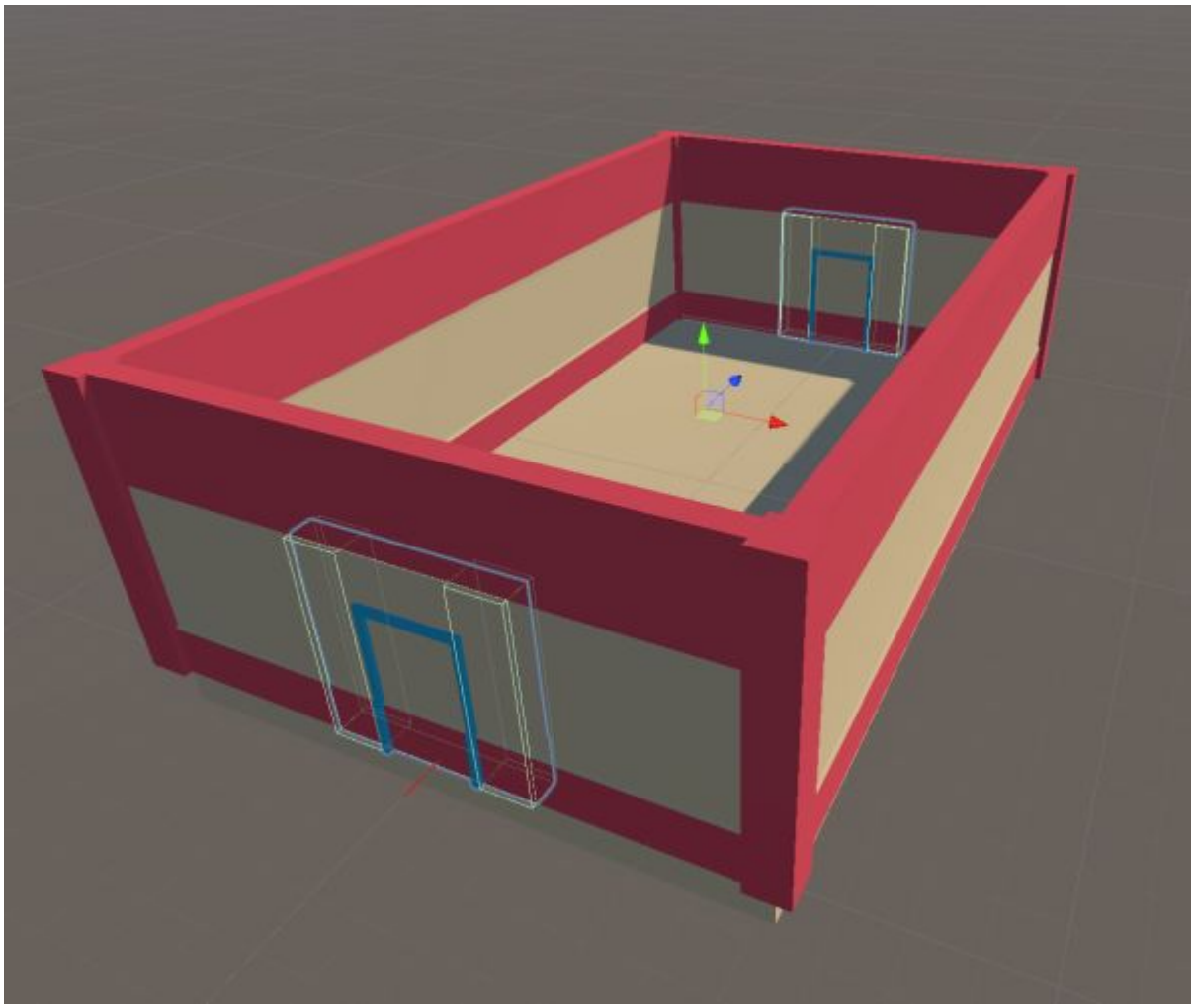
---

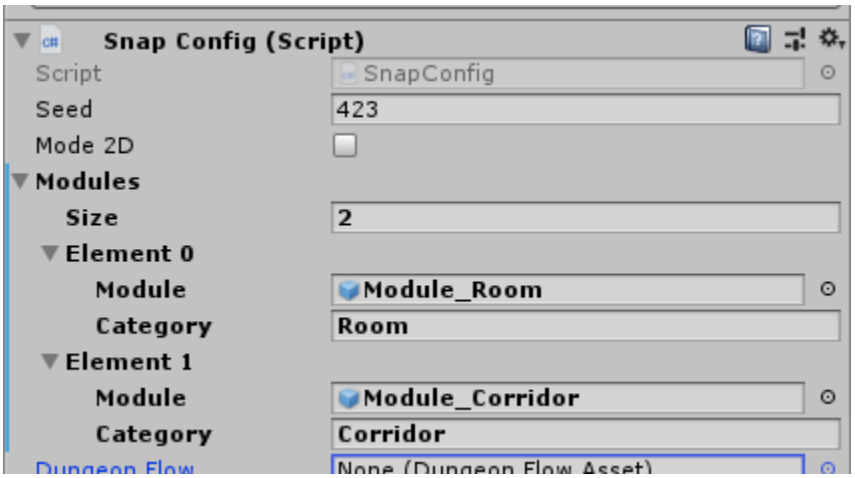
Execute the rule and see how the final graph is generated. You do this by clicking the Run icon on the Execution graph panel

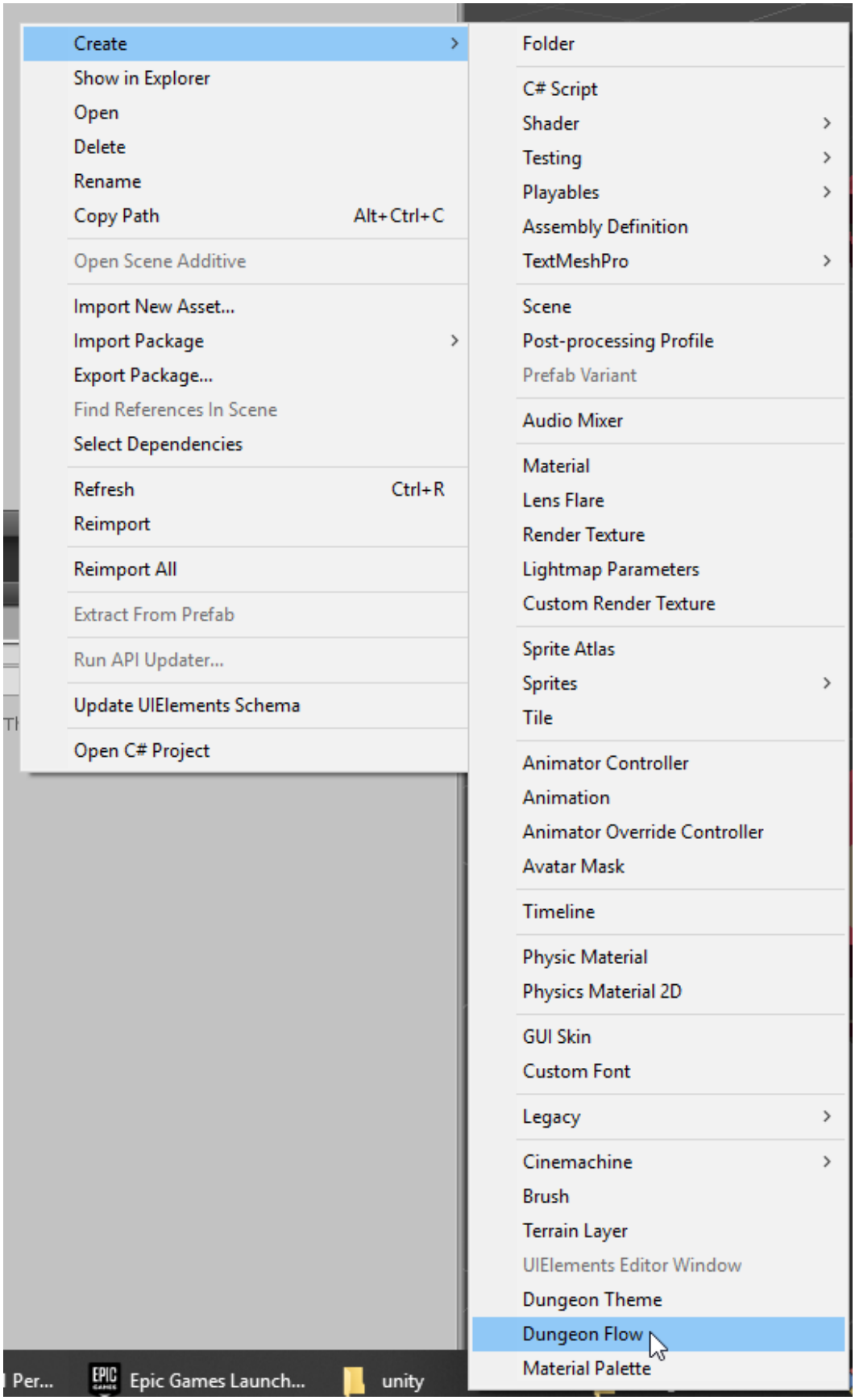
We'd like to insert Corridors between the rooms. Create another rule and give it a name (e.g. *Insert Corridors*)

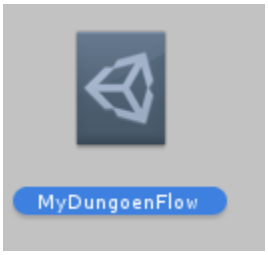
On the LHS, we want to find a patterns where two rooms are connected to each other like this (Room -> Room) and have it replaced with (Room -> Corridor -> Room)

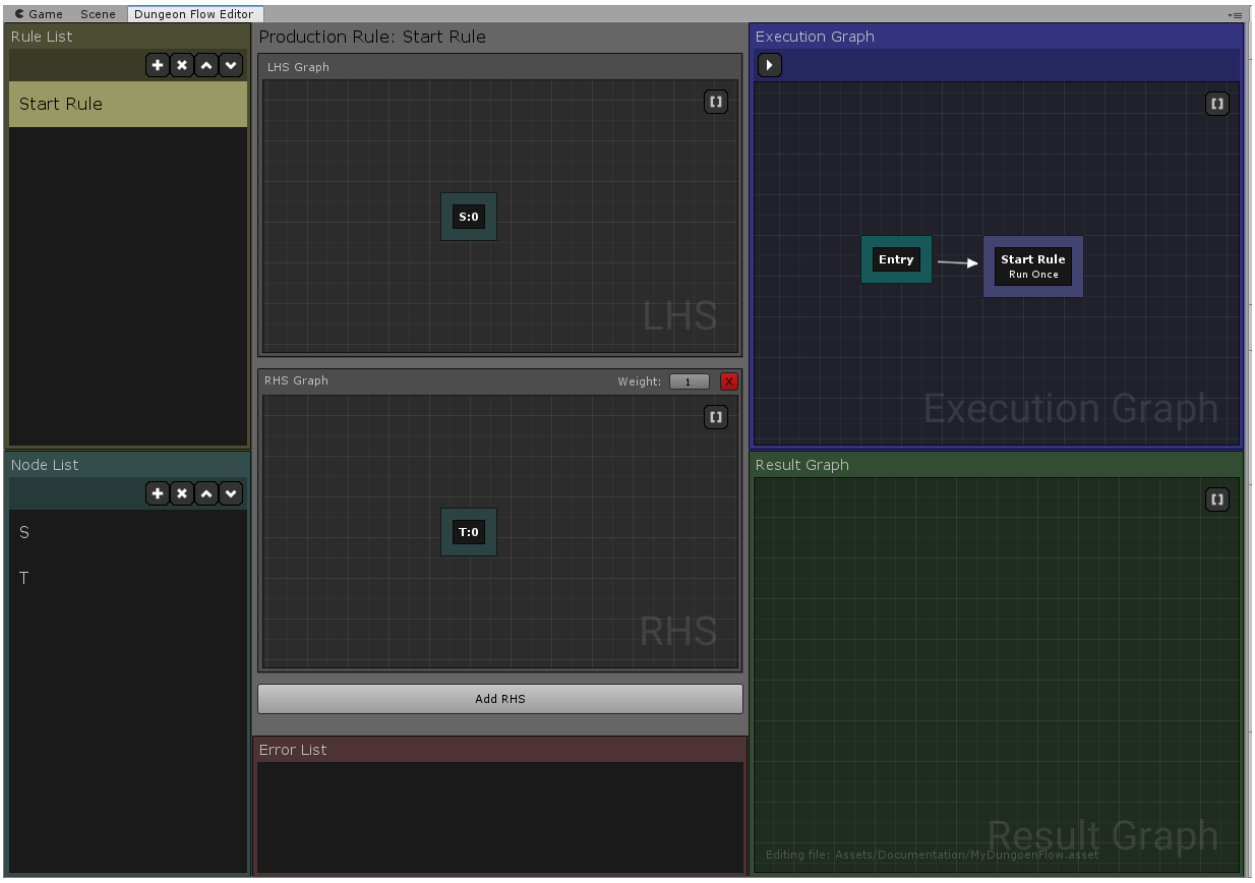
The Graph Grammar will find a pattern you specify on the LHS and replace it with the one you specify on the RHS

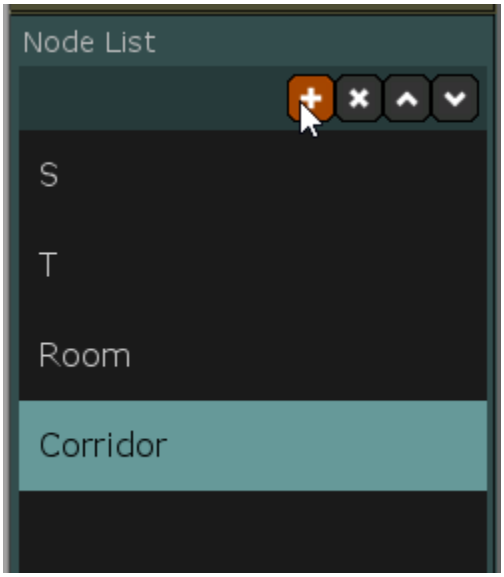



Snap Config (Script)
Script
SnapConfig
Seed
423
Mode 2D
Modules
Size
2
Element 0
Module
Module_Room
Category
Room
Element 1
Module
Module_Corridor
Category
Corridor
Dungeon Flow
None (Dungeon Flow Asset)

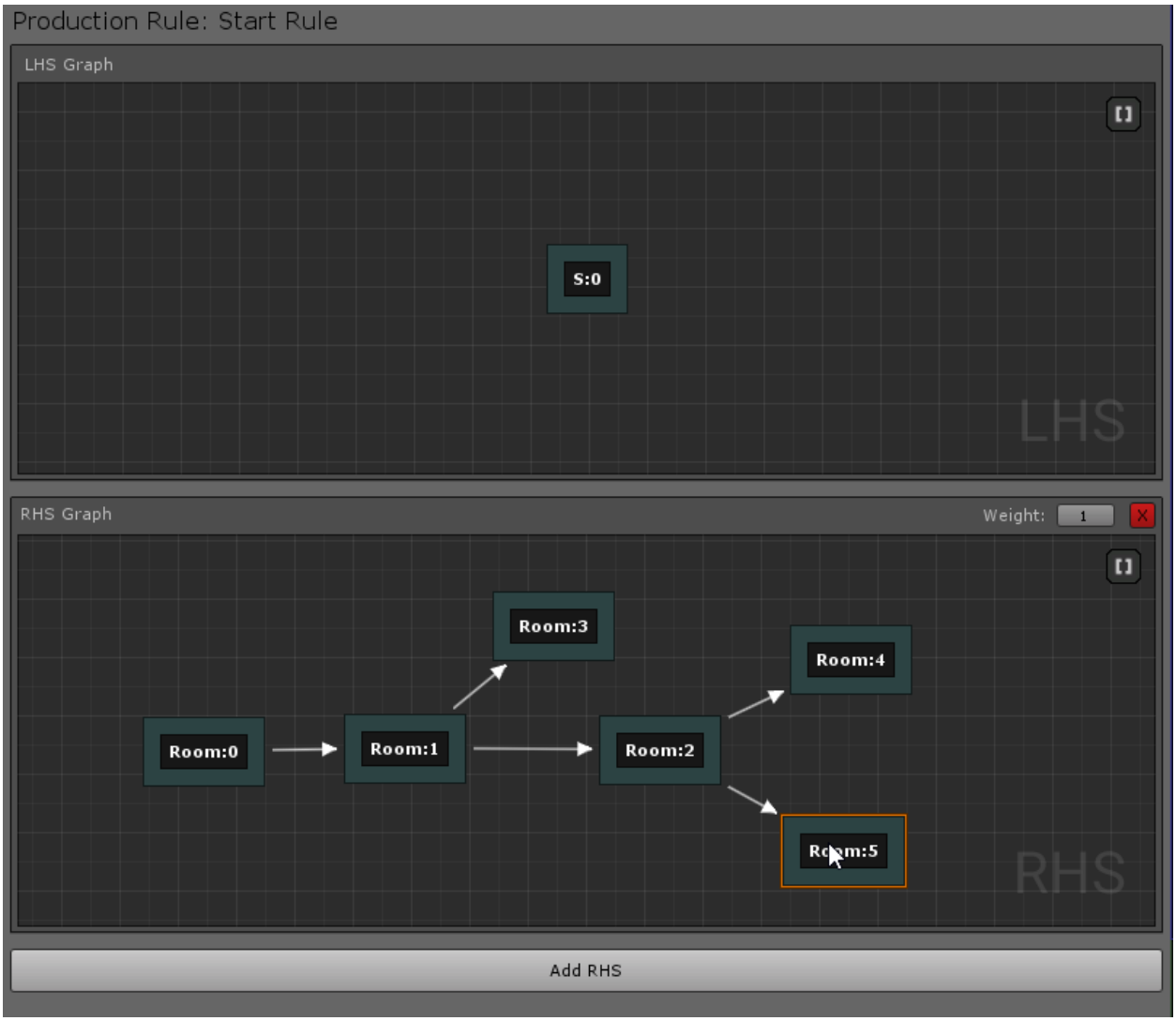

Create
Show in Explorer
Open
Delete
Rename
Copy Path Alt+Ctrl+C
Open Scene Additive
Import New Asset...
Import Package
Export Package...
Find References In Scene
Select Dependencies
Refresh Ctrl+R
Reimport
Reimport All
Extract From Prefab
Run API Updater...
Update UIElements Schema
Open C# Project
Folder
C# Script
Shader
Testing
Playables
Assembly Definition
TextMeshPro
Scene
Post-processing Profile
Prefab Variant
Audio Mixer
Material
Lens Flare
Render Texture
Lightmap Parameters
Custom Render Texture
Sprite Atlas
Sprites
Tile
Animator Controller
Animation
Animator Override Controller
Avatar Mask
Timeline
Physic Material
Physics Material 2D
GUI Skin
Custom Font
Legacy
Cinemachine
Brush
Terrain Layer
UIElements Editor Window
Dungeon Theme
Dungeon Flow
Material Palette
Per...
Epic Games Launch...
unity

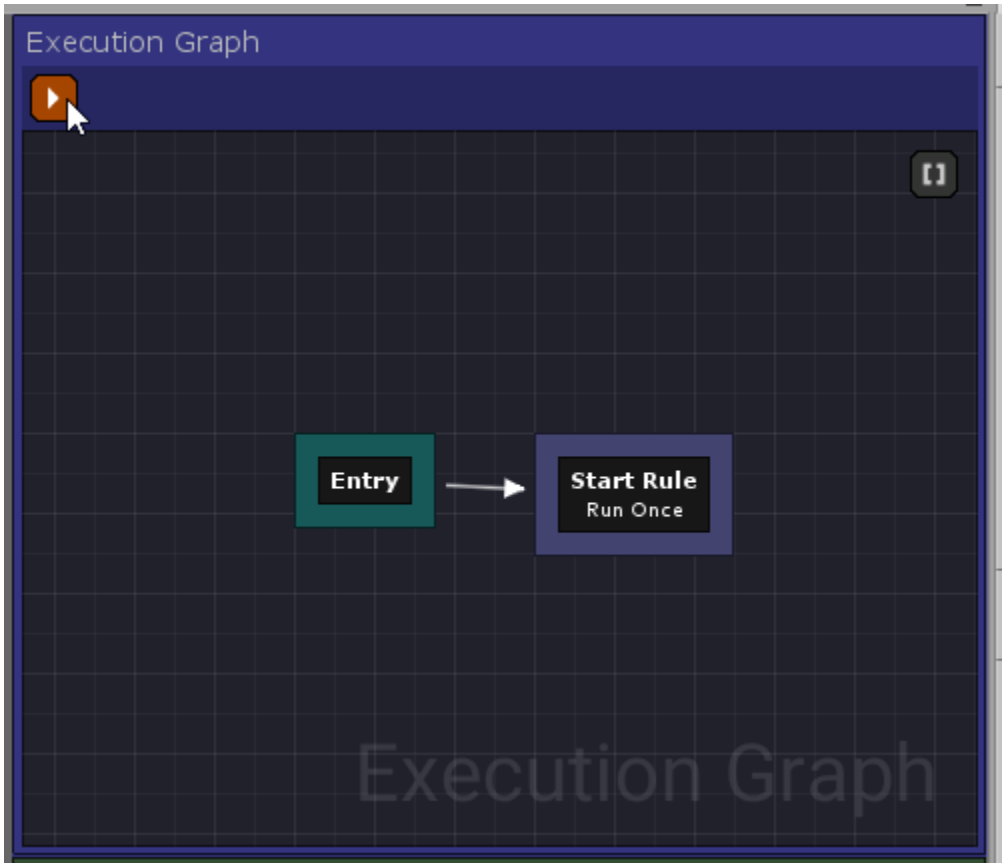

MyDungoenFlow

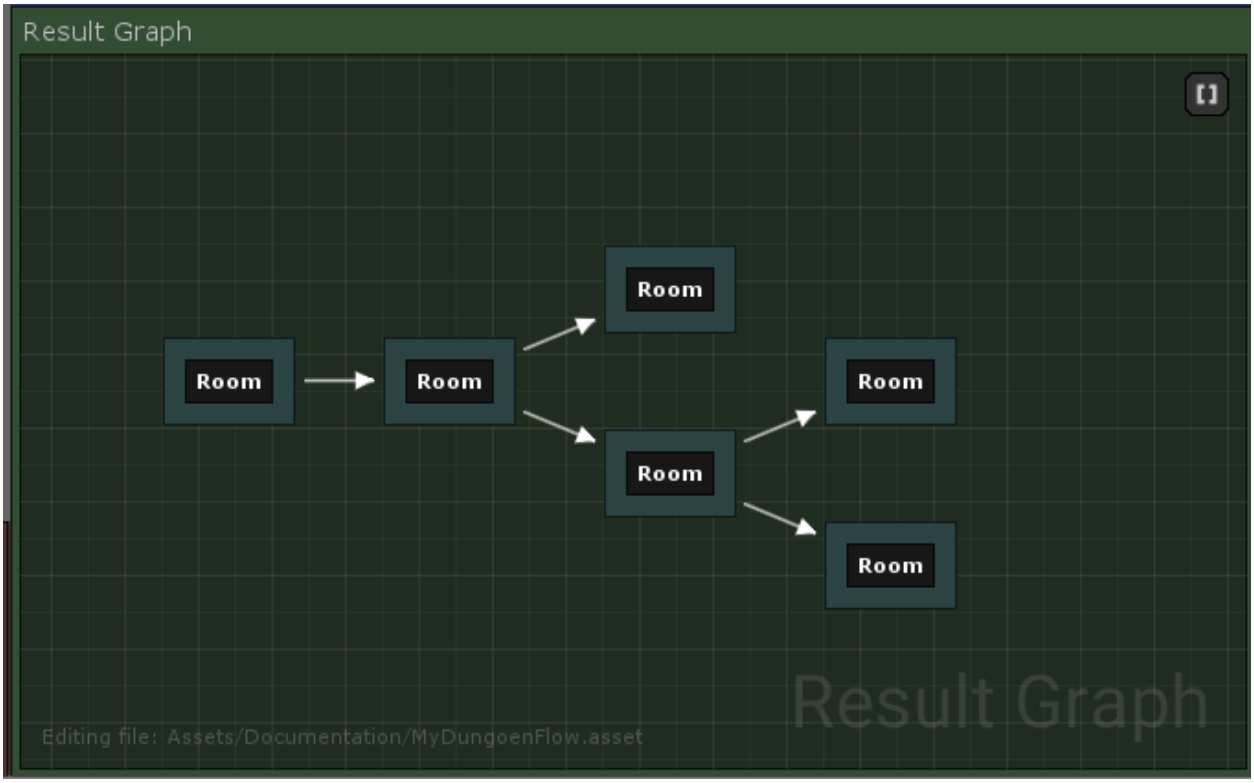

Game
Scene
Dungeon Flow Editor
Rule List
Start Rule
Node List
S
T
Production Rule: Start Rule
LHS Graph
S:0
LHS
RHS Graph
Weight:
1
T:0
RHS
Add RHS
Error List
Execution Graph
Entry
Start Rule
Run Once
Execution Graph
Result Graph
Result Graph
Editing file: Assets/Documentation/MyDungoenFlow.asset

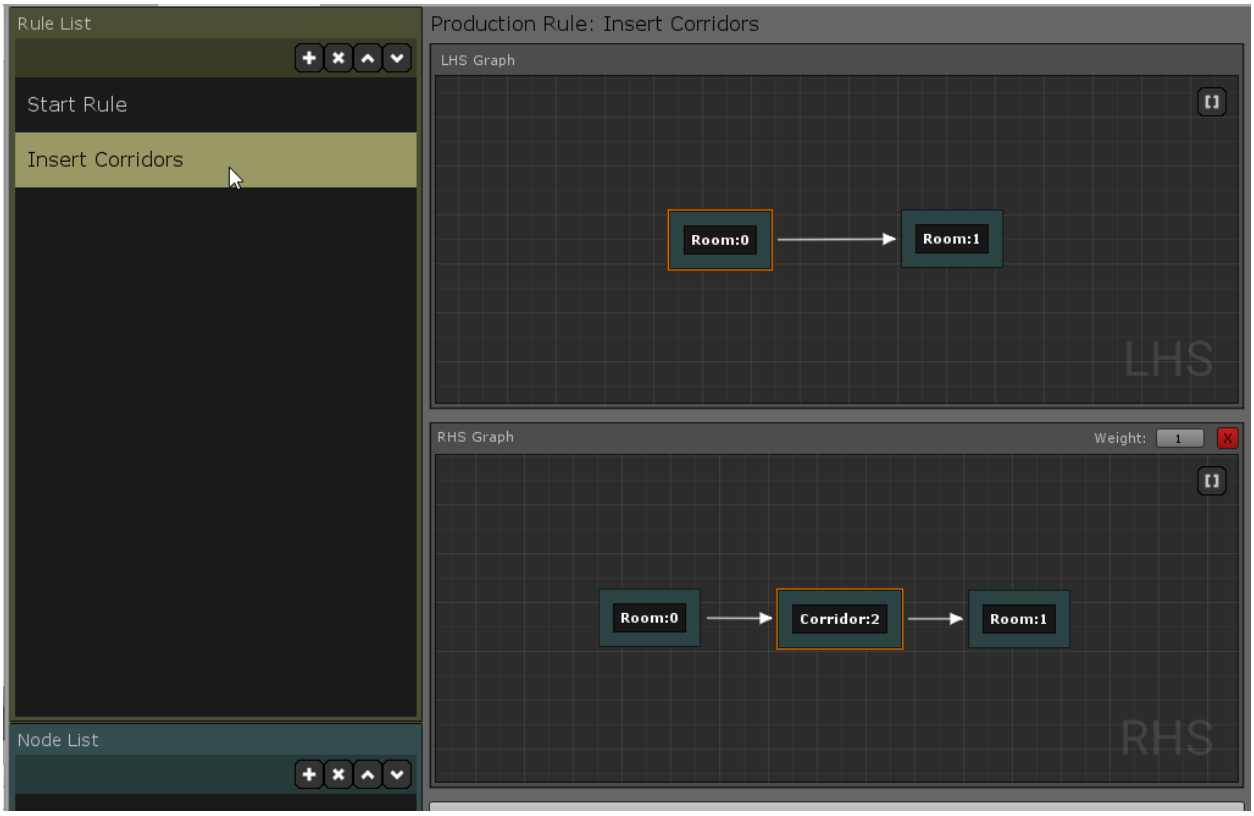

Node List
S
T
Room
Corridor


Rule List
Start Rule


Production Rule: Start Rule
LHS Graph
S:0
LHS
RHS Graph
Weight:
1
Room:3
Room:4
Room:0
Room:1
Room:2
Room:5
RHS
Add RHS

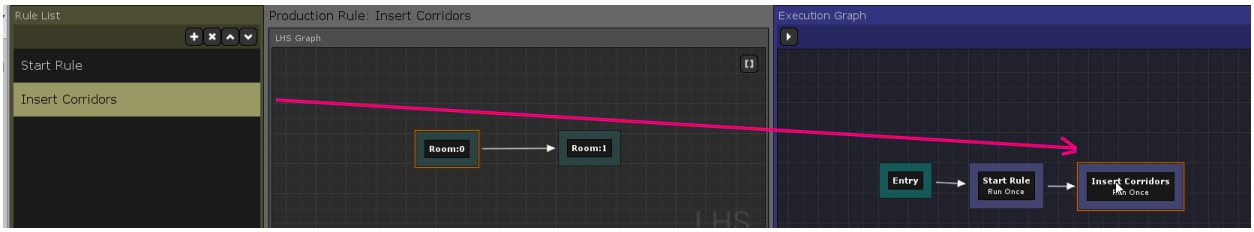

Execution Graph
Entry
Start Rule
Run Once
Execution Graph


Result Graph
Room
Room
Room
Room
Room
Room
Result Graph
Editing file: Assets/Documentation/MyDungoenFlow.asset




The Indices on the nodes (e.g. Room:0, Room:1) are important that helps in correct mapping. Since we properly specified 0 and 1 indices on the RHS, it knows the direction of the newly created links to the corridor. This will be covered in detail in the full documentation soon

You control how your rules are run from the **Execution Graph**. Drag drop your newly created **Insert Corridor** rule on to the execution graph and connect it after the *Start Rule*.



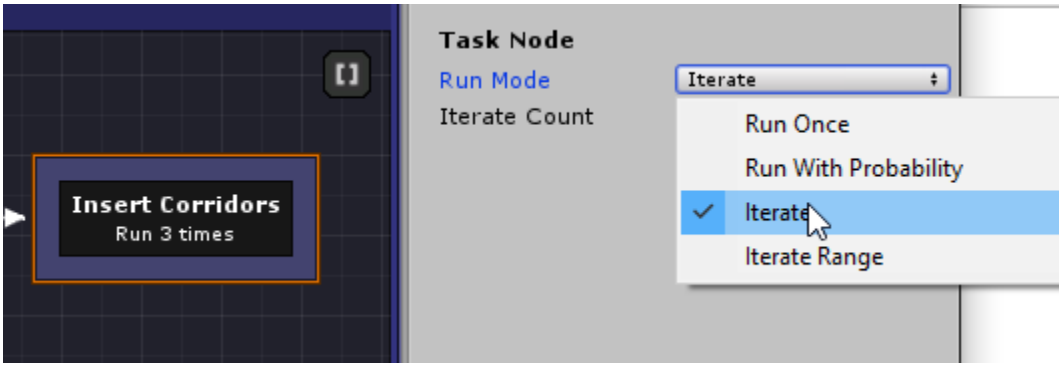

Select the newly placed node and from the details panel, change the execution mode to Iterate and set the count to 2 or 3 (This makes the rule run multiple times since the newly replaced Room nodes wont map with the adjacent older Room nodes by design and need to be run again)

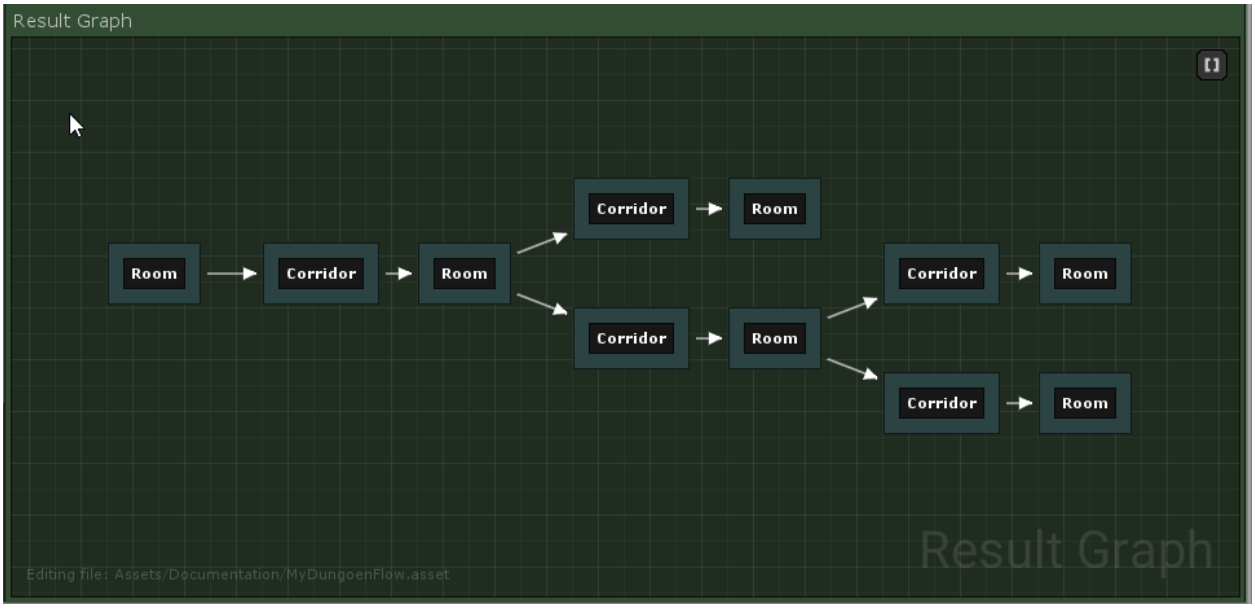
Execute the grammar and you'll now see corridors between your rooms

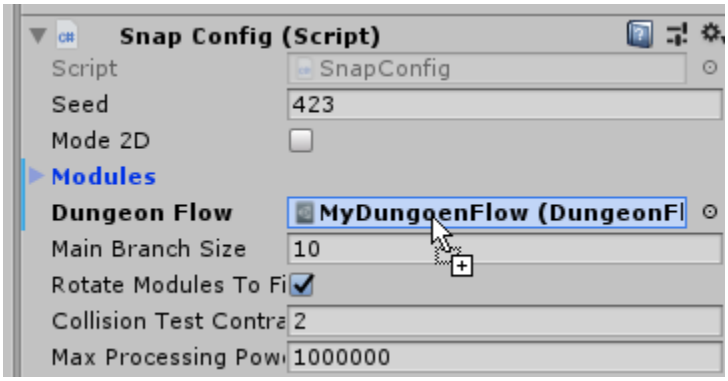
We will use this Dungeon Flow graph grammar to generate our snap dungeons

## 5.6 Generating the Dungeon

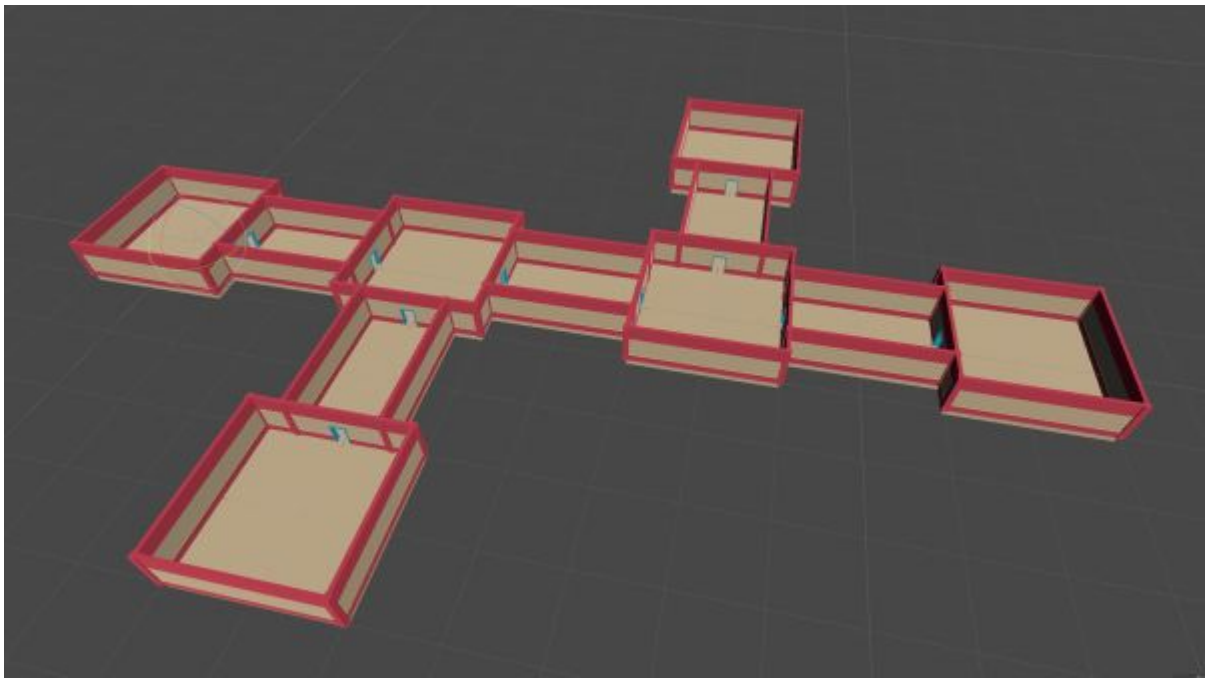
Assign the **Dungeon Flow** assets to the DungeonSnap game object


Task Node
Run Mode
Iterate Count
Iterate
Run Once
Run With Probability
Iterate
Iterate Range
Insert Corridors
Run 3 times


Result Graph
Room
Corridor
Room
Corridor
Room
Corridor
Room
Corridor
Room
Corridor
Room
Editing file: Assets/Documentation/MyDungeonFlow.asset
Result Graph

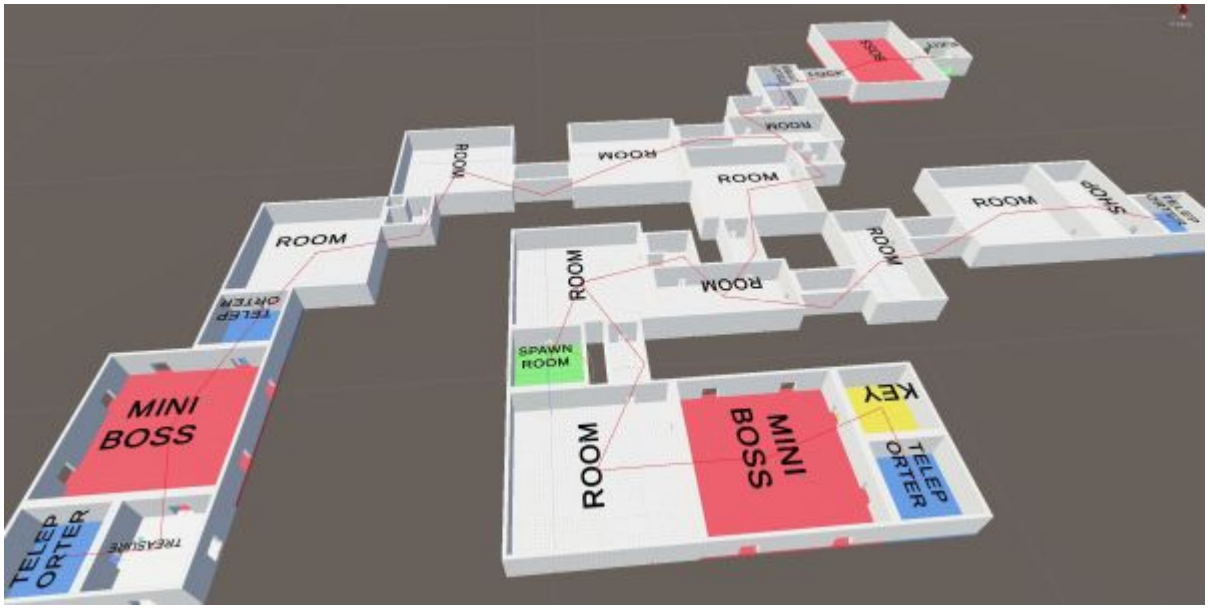

Snap Config (Script)
Script
SnapConfig
Seed
423
Mode 2D
Modules
Dungeon Flow
MyDungeonFlow (DungeonFl
Main Branch Size
10
Rotate Modules To Fi
Collision Test Contra
2
Max Processing Pow
1000000

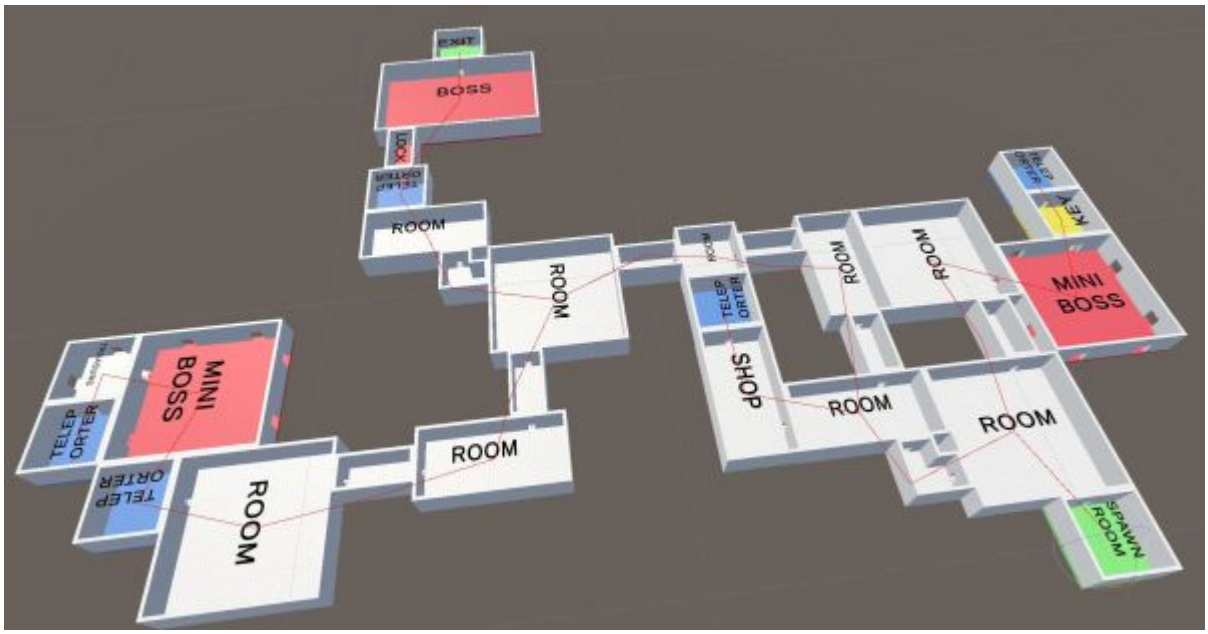
Hit **Build Dungeon**. Randomize the seed and get different configurations that satisfy the layout graph you defined in the flow asset. Change the Dungeon Flow graph and experiment further

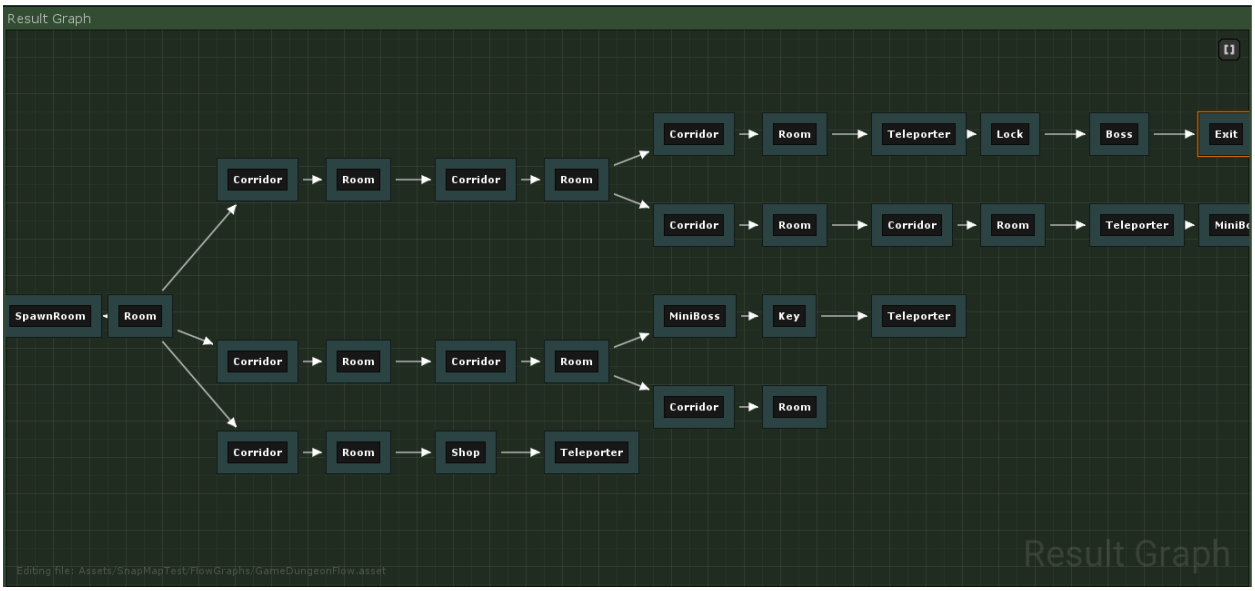


Explore the Sample for a more complex and complete dungeon map with multi key-lock system, treasure rooms guarded by miniboss, exist guarded by a Boss room which requires a key to unlock

Location: *DungeonArchitect_SamplesDemoBuilder_SnapScenesDemoScene*




EXIT
BOSS
LOCK
TELEP ORTER
ROOM
ROOM
ROOM
ROOM
ROOM
TELEP ORTER
KEY
MINI BOSS
SHOP
ROOM
ROOM
MINI BOSS
TELEP ORTER
TELEP ORTER
ROOM
ROOM
SPAWN ROOM

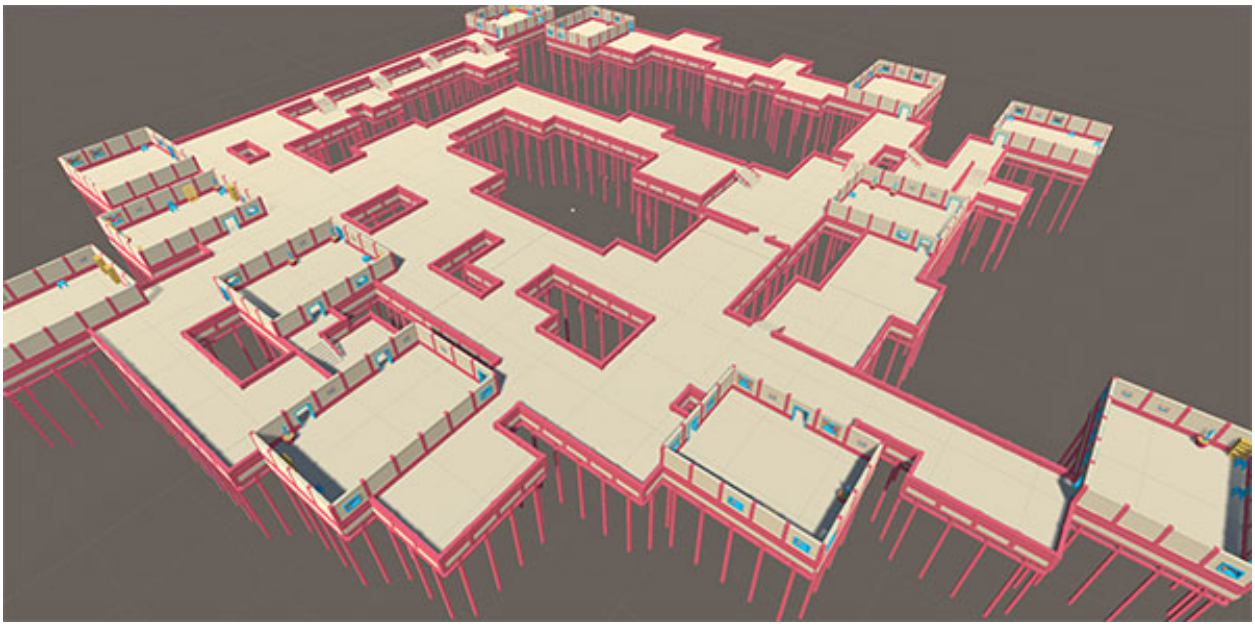

Result Graph
SpawnRoom
Room
Corridor
Room
Corridor
Room
Corridor
Room
Teleporter
Lock
Boss
Exit
Corridor
Room
Corridor
Room
Teleporter
Corridor
Room
MiniBoss
Key
Teleporter
Corridor
Room
Corridor
Room
Corridor
Room
Shop
Teleporter
Editing file: Assets/SnapMapTest/FlowGraphs/GameDungeonFlow.asset
Result Graph

CHAPTER

SIX

# GRID BUILDER

The *Grid Builder* generates a dungeon by scattering rooms across the map and connecting them with corridors. The builder supports height variations (stairs)



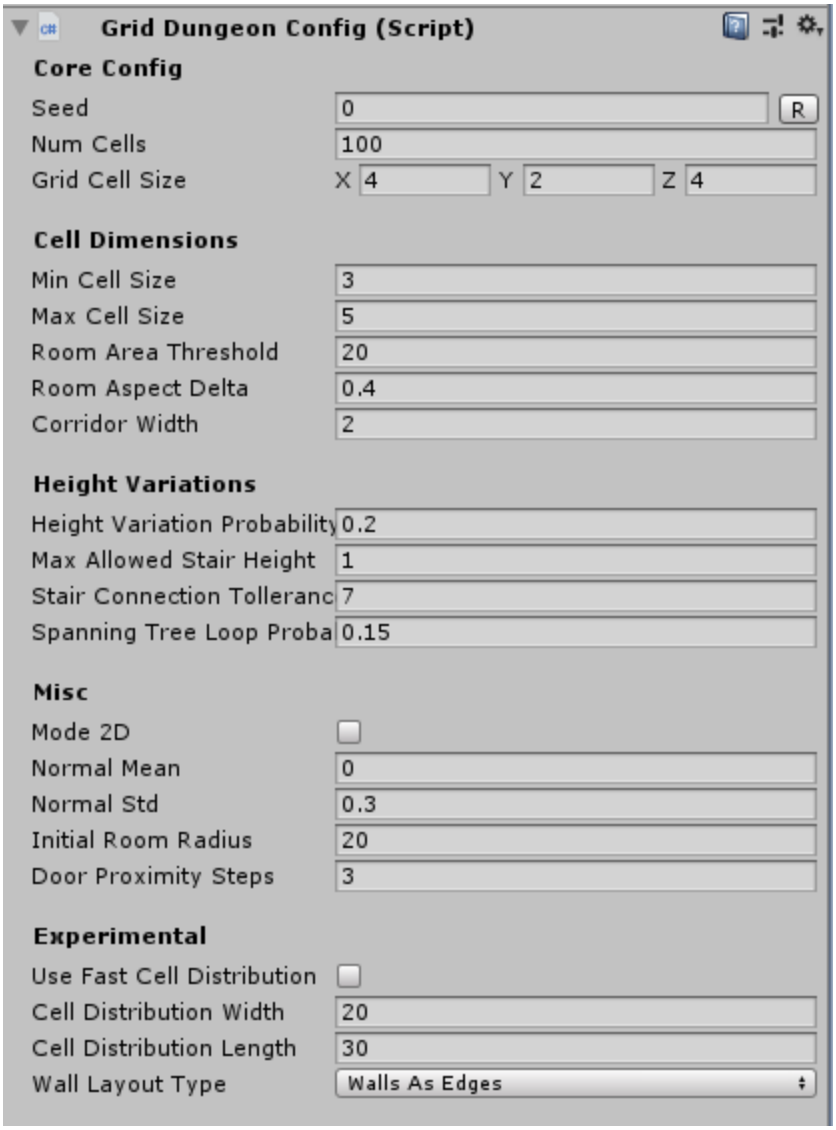
We've used this dungeon builder in the previous section *Create your first Dungeon* and *Design your first theme*

In this section, we'll explore more on this builder

## 6.1 Properties

Continuing on the scene created in the section *Design your first theme*, open the scene and select the `DungeonGrid` game object and inspect the properties

- Change the `Seed` parameter to build a different dungeon layout
- Set the `Grid Cell Size` parameter according to your moduler art asset. If the ground mesh is 4x4 and the stair mesh height is 2, set this to `(4, 2, 4)`
- The dungeon creation method first creates a number of cells (defined by `NumCells`) and spreads them across the scene.
- The size of these cells is define by the parameters `Min Cell Size` and `Max Cell Size`


Grid Dungeon Config (Script)
Core Config
Seed
0
R
Num Cells
100
Grid Cell Size
X 4
Y 2
Z 4
Cell Dimensions
Min Cell Size
3
Max Cell Size
5
Room Area Threshold
20
Room Aspect Delta
0.4
Corridor Width
2
Height Variations
Height Variation Probability
0.2
Max Allowed Stair Height
1
Stair Connection Tolleranc
7
Spanning Tree Loop Proba
0.15
Misc
Mode 2D
Normal Mean
0
Normal Std
0.3
Initial Room Radius
20
Door Proximity Steps
3
Experimental
Use Fast Cell Distribution
Cell Distribution Width
20
Cell Distribution Length
30
Wall Layout Type
Walls As Edges

- Some of these cells will be promoted to Rooms and the rest will be promoted to Corridors. If the cell area is large enough (parameter `Room Area Threshold`), that cell is promoted to a *Room*, otherwise a *Corridor*
- The rooms are connected together with corridors
- Control how much height variation is allowed with `Height Variation Probability`
- A new Stair will not be created between two tiles if there's another stair nearby that if traversed, takes N steps to reach this cell. This value is controlled by `Stair Connection Tollerance`. Bump this number up if you want fewer stairs
- Maximum allowed stair: Determine how high a stair tile can get. If set to one, the builder will create stairs that go only one level up
- Both Rooms and Corridors have a `Ground` marker. Rooms are surrounded by `Wall` markers while the corridors are surrounded by `Fence` marker

## 6.2 Platform Volume

Platform Volumes let you control the placement of the rooms or corridors. You do this dropping in a platform volume on to the scene and resizing / positioning it on the scene and you room will be built around it.

Navigate to `Asset/DungeonArchitect/Prefabs` and drag drop the `PlatformVolume` prefab on to the scene

Select the PlatformVolume game object and set the Dungeon reference

Click the button `Rebuild Dungeon`

Move the *Platform Volume* and scale it to control the position and size of your room. You can have multiple platform volumes in the scene. Check the samples in the Launch Pad for more examples

## 6.3 Theme Override Volume

Theme Override Volumes let you apply another theme on certain portions of your dungeons that are covered by this volume. These are useful for adding variations to your dungeons.

Navigate to `Asset/DungeonArchitect/Prefabs` and drag drop the `ThemeOverrideVolume` prefab on to the scene

Move and scale it to cover a certain portion of the dungeon

Select the theme override volume you just dropped and inspect the properties and assign the DungeonGrid reference

Navigate to `Assets/DungeonArchitect_Samples/Demo_Theme_SimpleShapes/Themes` and assign the theme `Theme_Basic_White` to the Theme Override Volume

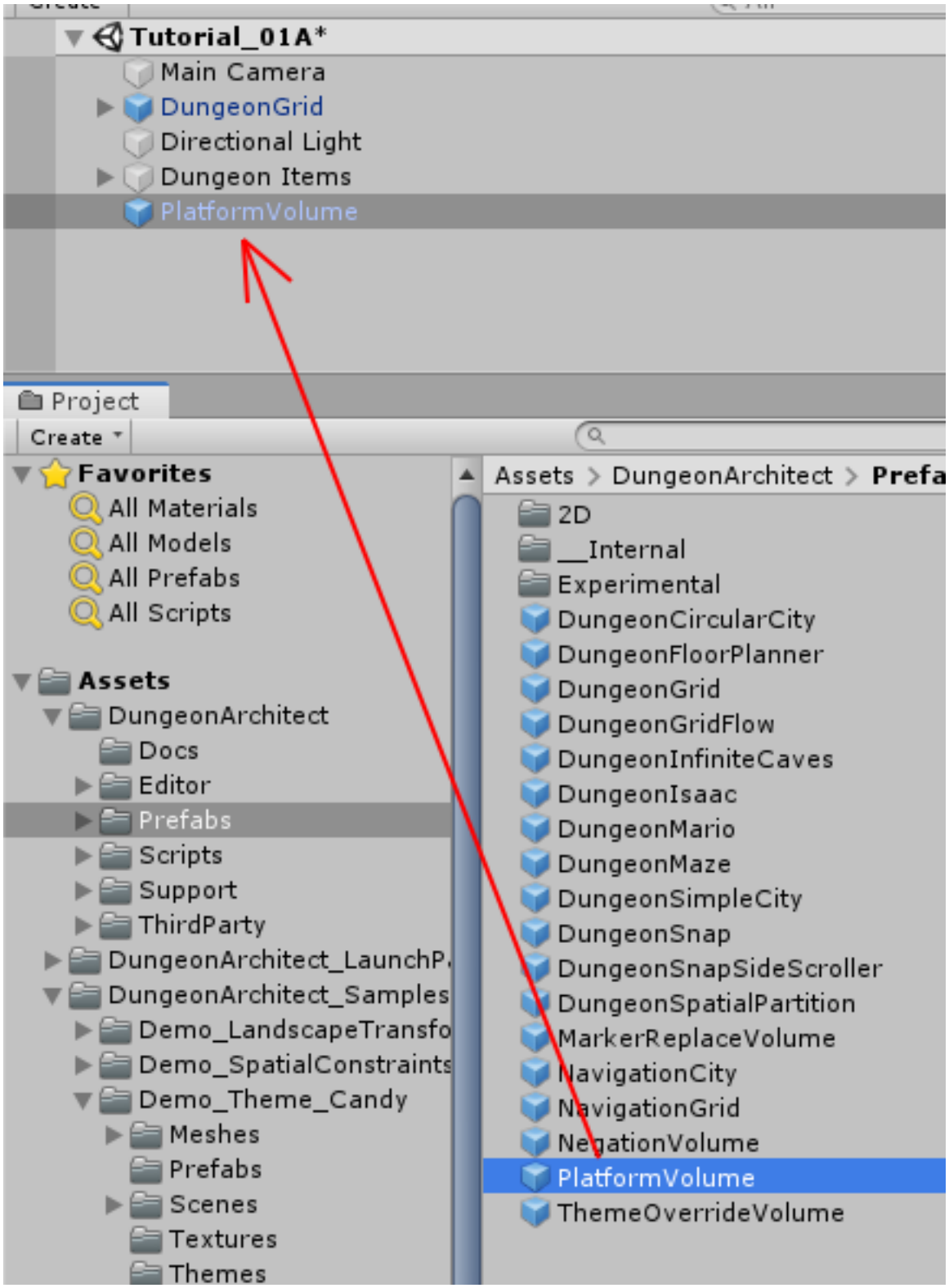
Click `Rebuild Dungeon`

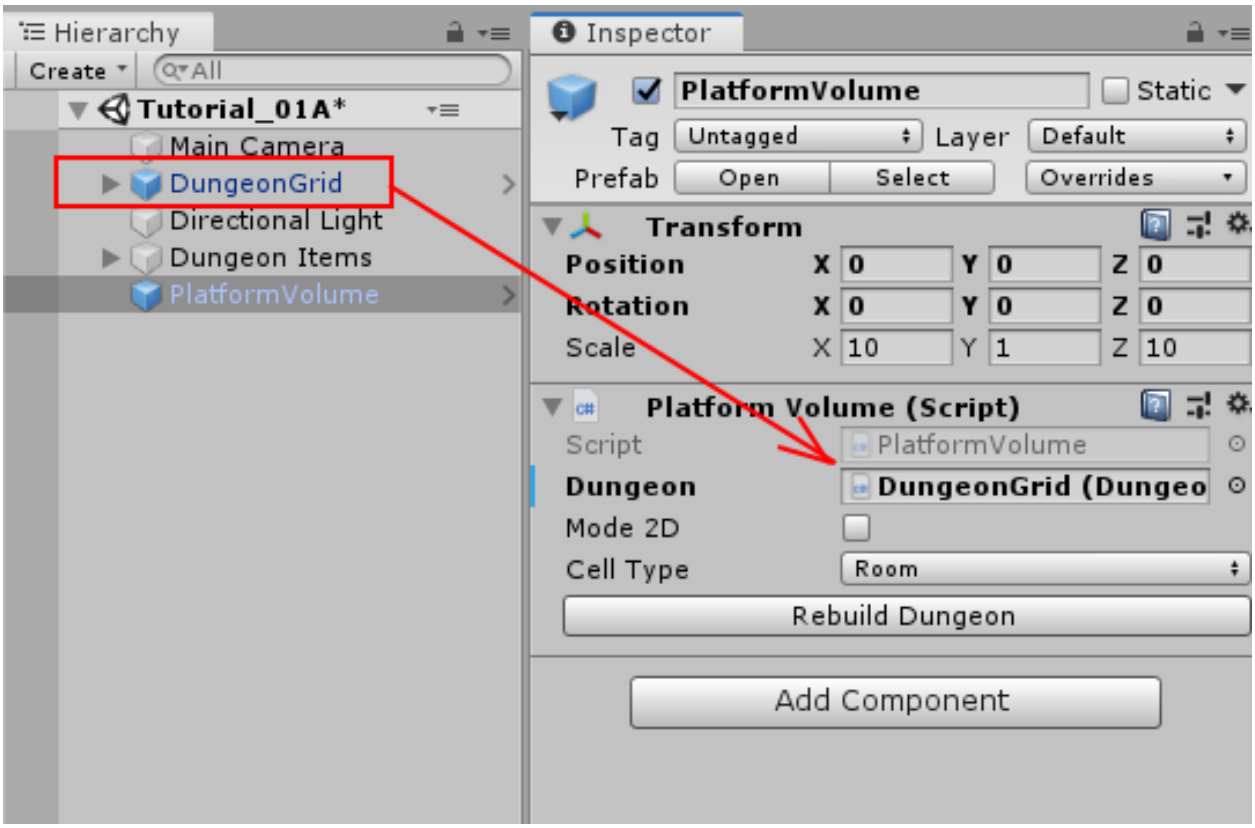
## 6.4 Paint Mode

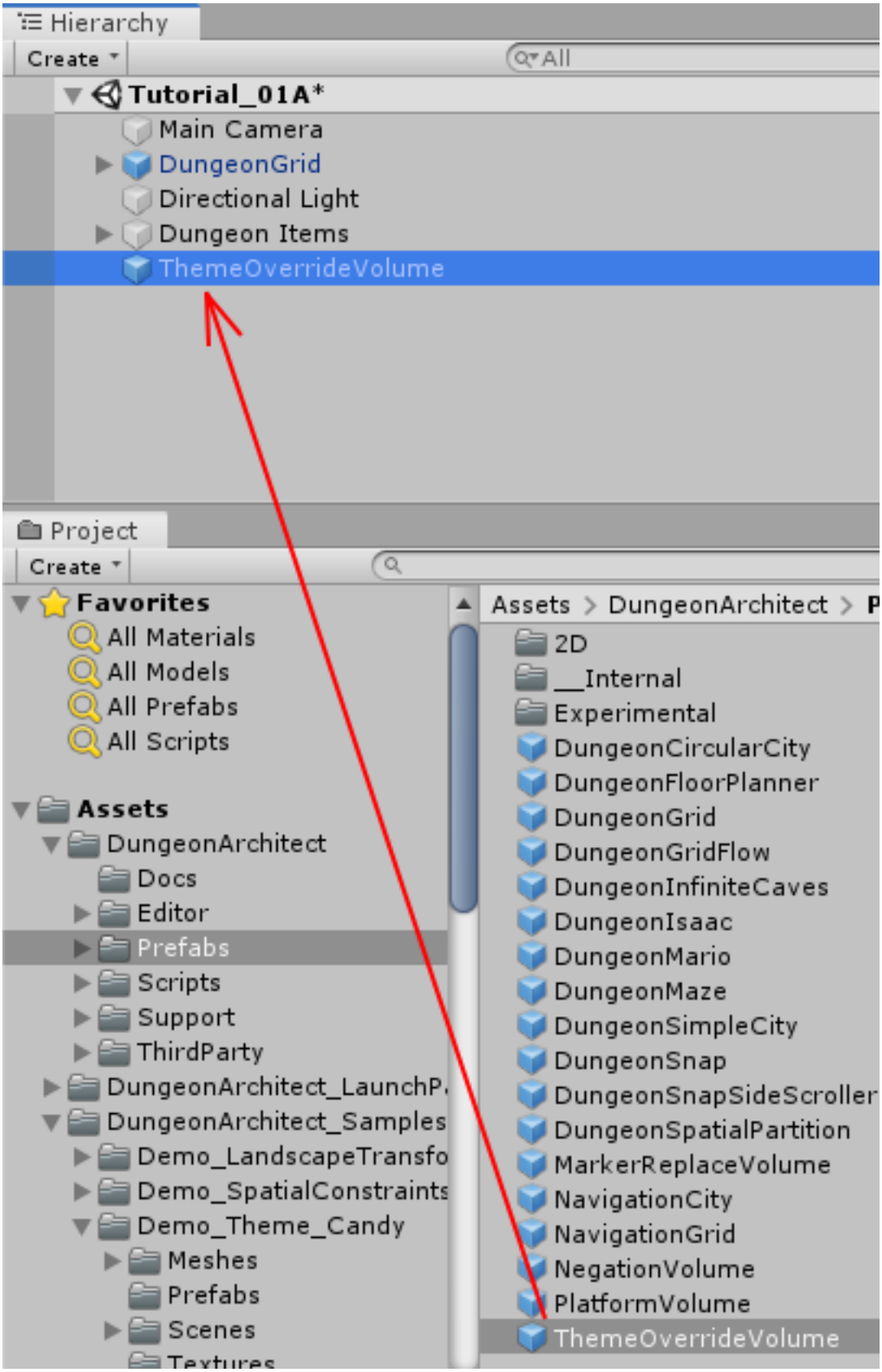
You can paint your own dungeon layout on top of the procedural dungeon

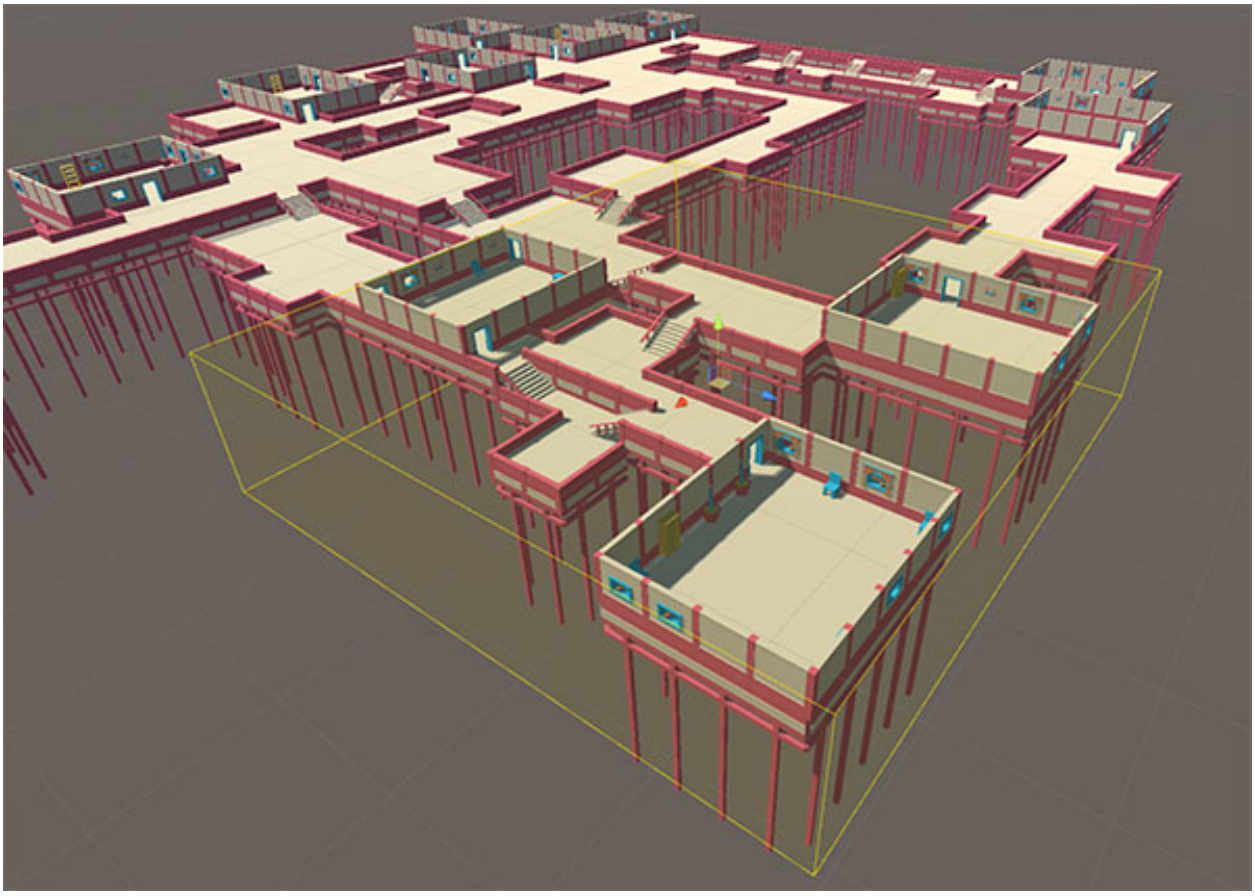
Select the DungeonGrid prefab and expand it. Select the `PaintMode` game object. This will activate the *Paint Mode*

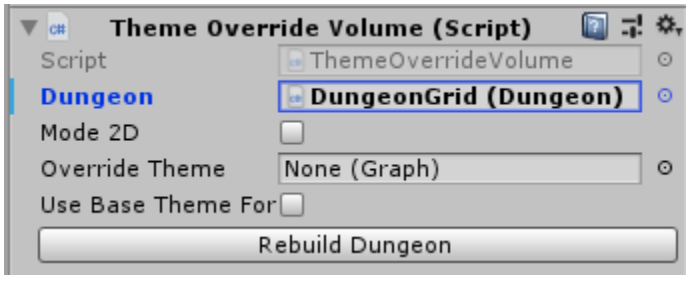
`Left click` and drag to draw dungeon cells. `Shift + Left click` to delete cells. `Scroll wheel` to move the cursor up/down

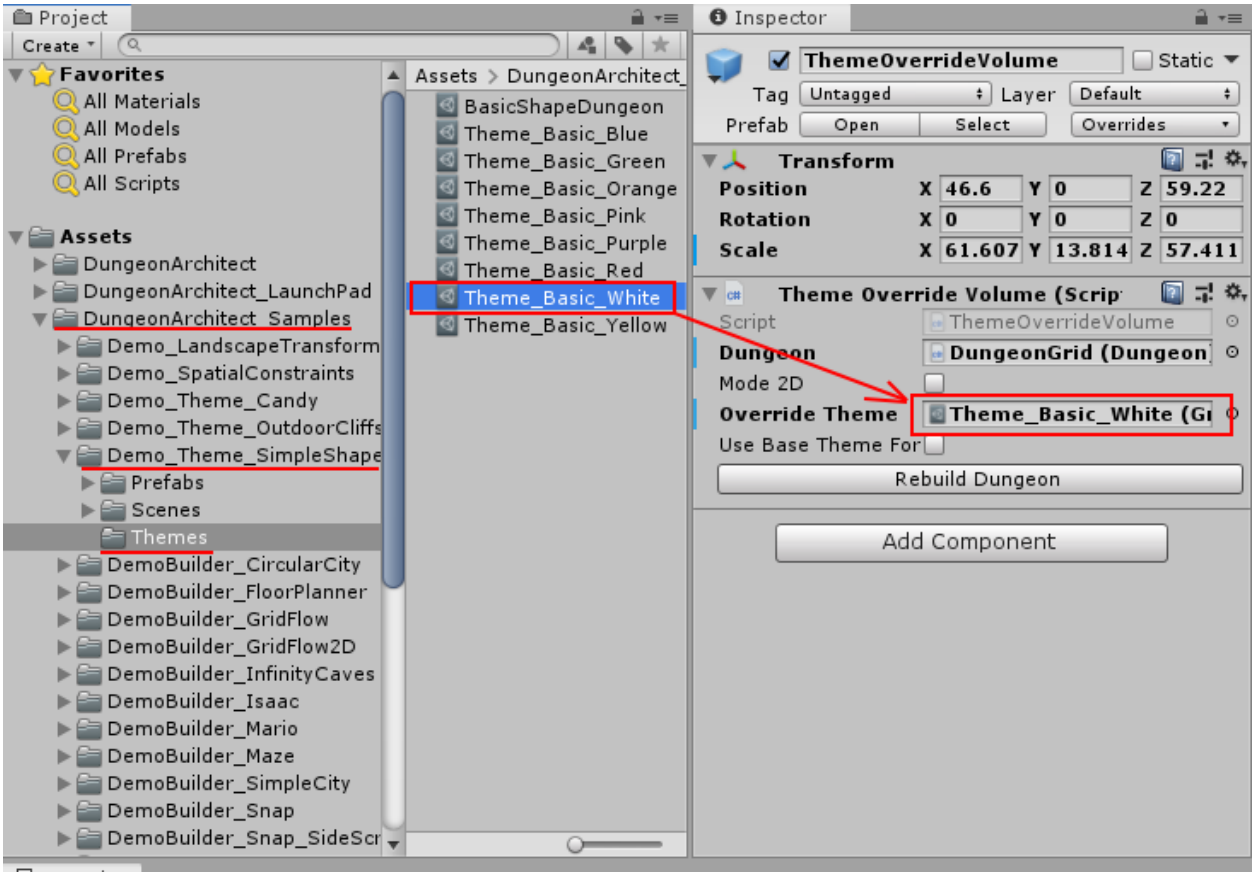

Tutorial_01A*
Main Camera
DungeonGrid
Directional Light
Dungeon Items
PlatformVolume
Project
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
DungeonArchitect
Docs
Editor
Prefabs
Scripts
Support
ThirdParty
DungeonArchitect_LaunchP
DungeonArchitect_Samples
Demo_LandscapeTransfo
Demo_SpatialConstraints
Demo_Theme_Candy
Meshes
Prefabs
Scenes
Textures
Themes
Assets > DungeonArchitect > Prefa
2D
__Internal
Experimental
DungeonCircularCity
DungeonFloorPlanner
DungeonGrid
DungeonGridFlow
DungeonInfiniteCaves
DungeonIsaac
DungeonMario
DungeonMaze
DungeonSimpleCity
DungeonSnap
DungeonSnapSideScroller
DungeonSpatialPartition
MarkerReplaceVolume
NavigationCity
NavigationGrid
NegationVolume
PlatformVolume
ThemeOverrideVolume

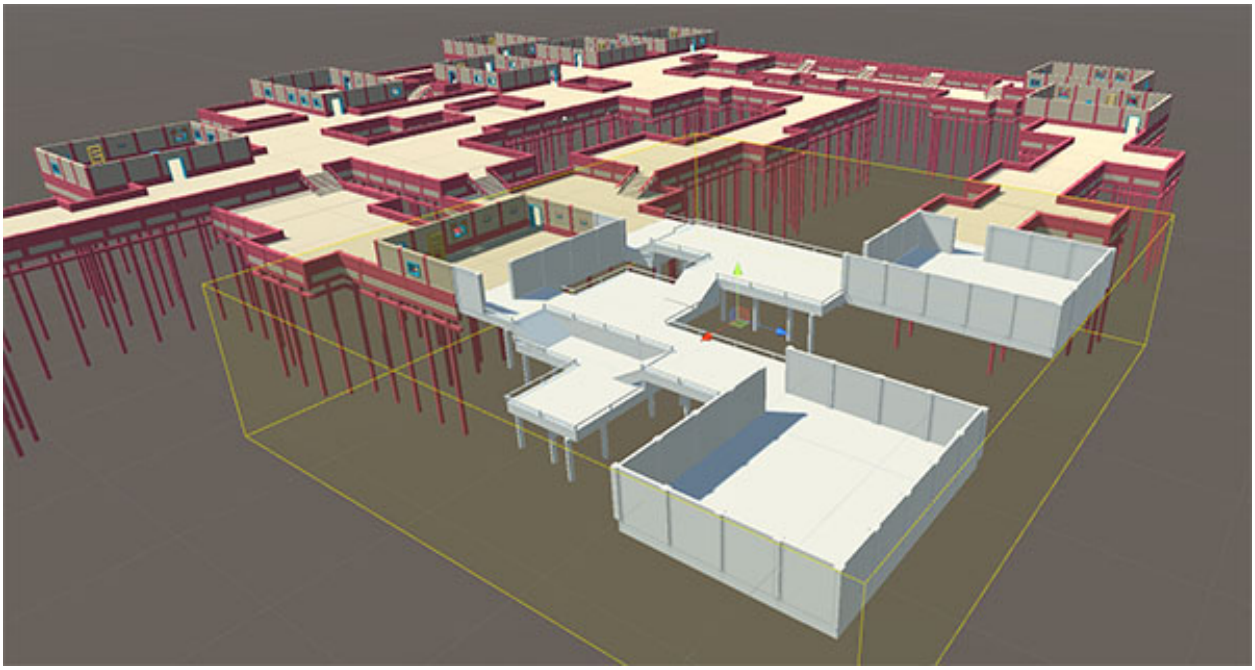

Hierarchy
Create
All
Tutorial_01A*
Main Camera
DungeonGrid
Directional Light
Dungeon Items
PlatformVolume
Inspector
PlatformVolume
Static
Tag
Untagged
Layer
Default
Prefab
Open
Select
Overrides
Transform
Position
X 0
Y 0
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 10
Y 1
Z 10
Platform Volume (Script)
Script
PlatformVolume
Dungeon
DungeonGrid (Dungeo
Mode 2D
Cell Type
Room
Rebuild Dungeon
Add Component

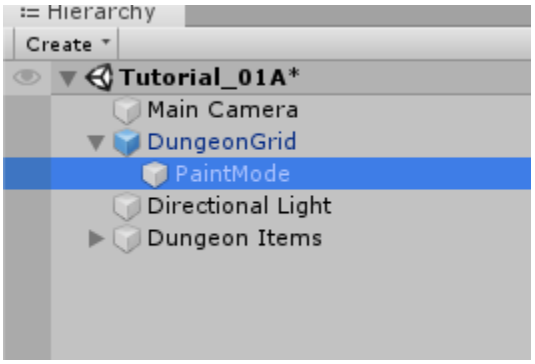

Hierarchy
Create
All
Tutorial_01A*
Main Camera
DungeonGrid
Directional Light
Dungeon Items
ThemeOverrideVolume
Project
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
DungeonArchitect
Docs
Editor
Prefabs
Scripts
Support
ThirdParty
DungeonArchitect_LaunchP
DungeonArchitect_Samples
Demo_LandscapeTransfo
Demo_SpatialConstraints
Demo_Theme_Candy
Meshes
Prefabs
Scenes
Textures
Assets > DungeonArchitect > P
2D
__Internal
Experimental
DungeonCircularCity
DungeonFloorPlanner
DungeonGrid
DungeonGridFlow
DungeonInfiniteCaves
DungeonIsaac
DungeonMario
DungeonMaze
DungeonSimpleCity
DungeonSnap
DungeonSnapSideScroller
DungeonSpatialPartition
MarkerReplaceVolume
NavigationCity
NavigationGrid
NegationVolume
PlatformVolume
ThemeOverrideVolume




Theme Override Volume (Script)
Script
ThemeOverrideVolume
Dungeon
DungeonGrid (Dungeon)
Mode 2D
Override Theme
None (Graph)
Use Base Theme For
Rebuild Dungeon


Project
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
DungeonArchitect
DungeonArchitect_LaunchPad
DungeonArchitect_Samples
Demo_LandscapeTransform
Demo_SpatialConstraints
Demo_Theme_Candy
Demo_Theme_OutdoorCliffs
Demo_Theme_SimpleShape
Prefabs
Scenes
Themes
DemoBuilder_CircularCity
DemoBuilder_FloorPlanner
DemoBuilder_GridFlow
DemoBuilder_GridFlow2D
DemoBuilder_InfinityCaves
DemoBuilder_Isaac
DemoBuilder_Mario
DemoBuilder_Maze
DemoBuilder_SimpleCity
DemoBuilder_Snap
DemoBuilder_Snap_SideScr
Assets > DungeonArchitect_
BasicShapeDungeon
Theme_Basic_Blue
Theme_Basic_Green
Theme_Basic_Orange
Theme_Basic_Pink
Theme_Basic_Purple
Theme_Basic_Red
Theme_Basic_White
Theme_Basic_Yellow
Inspector
ThemeOverrideVolume
Static
Tag
Untagged
Layer
Default
Prefab
Open
Select
Overrides
Transform
Position
X 46.6 Y 0 Z 59.22
Rotation
X 0 Y 0 Z 0
Scale
X 61.607 Y 13.814 Z 57.411
Theme Override Volume (Scrip
Script
ThemeOverrideVolume
Dungeon
DungeonGrid (Dungeon
Mode 2D
Override Theme
Theme_Basic_White (Gi
Use Base Theme For
Rebuild Dungeon
Add Component




Hierarchy
Create
Tutorial_01A*
Main Camera
DungeonGrid
PaintMode
Directional Light
Dungeon Items

CHAPTER

SEVEN

# ADVANCED THEMING


## 7.1 Selection Rules

One way to select the node in a theme graph is by setting the probablity. If you want more control, you can write you own selection logic in a script

### 7.1.1 Example #1

Different tiles for rooms and corridors

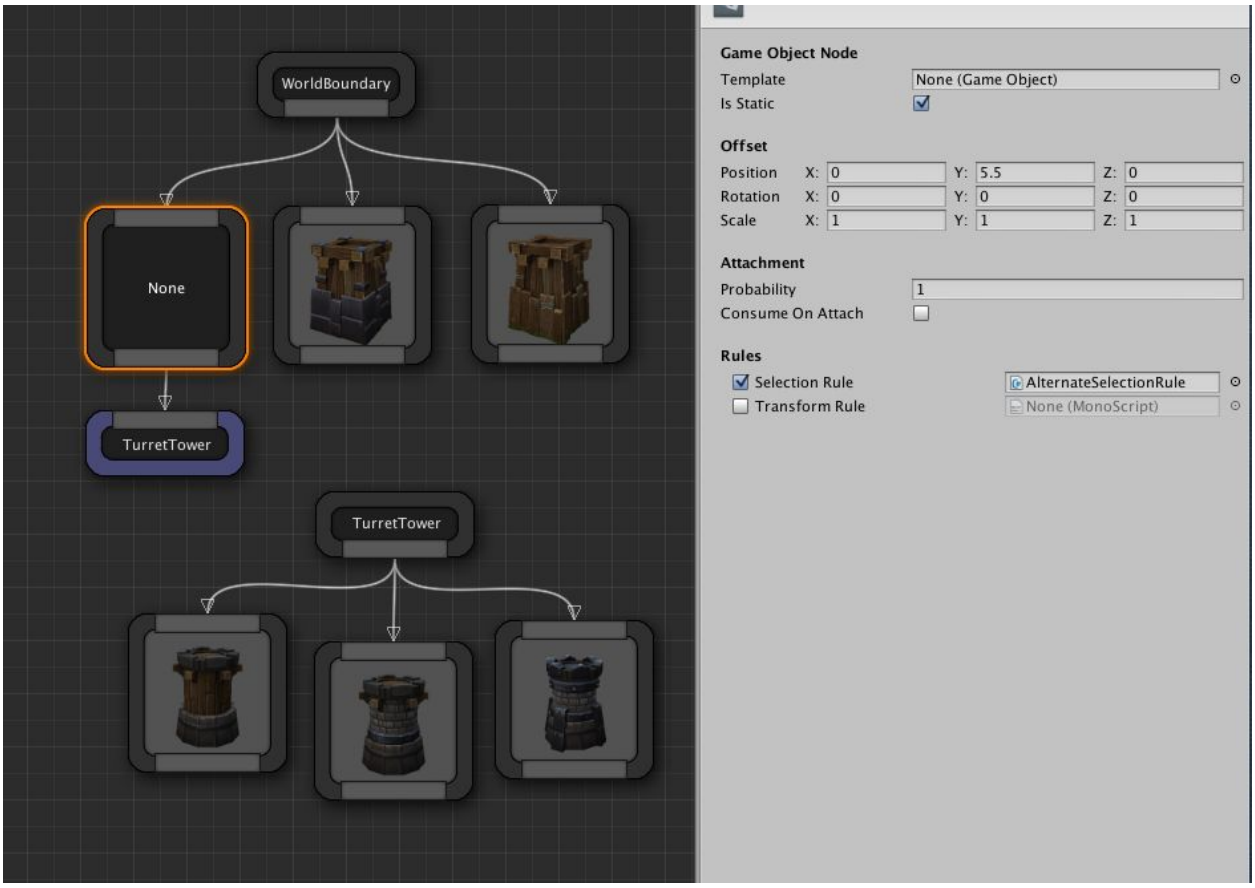
### 7.1.2 Example #2

In this example the towers are too crowded and close to each other.

A selector rule is created to select alternate cells

```
using UnityEngine;
using System.Collections;
using DungeonArchitect;
```

(continues on next page)


WorldBoundary
None
TurretTower
TurretTower
Game Object Node
Template
None (Game Object)
Is Static
Offset
Position X: 0 Y: 5.5 Z: 0
Rotation X: 0 Y: 0 Z: 0
Scale X: 1 Y: 1 Z: 1
Attachment
Probability
1
Consume On Attach
Rules
Selection Rule
AlternateSelectionRule
Transform Rule
None (MonoScript)

(continued from previous page)

```
public class AlternateSelectionRule : SelectorRule {
        public override bool CanSelect(PropSocket socket, Matrix4x4 propTransform,␣
↪DungeonModel model, System.Random random) {
                return (socket.gridPosition.x + socket.gridPosition.z) % 2 == 0;
        }
}
```

## 7.2 Transform Rules

A transform rule lets you apply an offset using a script. These are great for randomization or handling any special case of object alignment through script

CHAPTER

# EIGHT

## ADVANCED DUNGEONS

CHAPTER

NINE

# SIMPLE CITY BUILDER

CHAPTER

# TEN

## MARIO BUILDER

CHAPTER

# ELEVEN

## INFINITY CAVES BUILDER

CHAPTER

# TWELVE

## SPATIAL PARTITION BUILDER

CHAPTER

# THIRTEEN

# FLOOR PLAN BUILDER

CHAPTER

# FOURTEEN

## CIRCULAR CITY BUILDER

CHAPTER

# FIFTEEN

## ISAAC BUILDER

CHAPTER

SIXTEEN

# MAZE BUILDER