

# Level Designer's Tools Handbook

- Introduction
- SmartGit
  - Introduction
  - Repositories Window
  - Branches Window
  - Graph Window
  - Commit Window
- SmartGit Walkthrough
  - Start Of The Day - Pull
  - Commit & Push Prep
  - Commit & Push (Really this time)
- Unity
  - Introduction
  - World Editor Modes
    - Object Placement of World Objects
      - World Objects Filter
      - Picking World Objects In The Scene Tab
      - Selecting Multiple World Objects Via Marquee Select
      - Set Dressing World Objects
      - Gated World Objects
      - Adjusting a World Objects's polygon
      - Tagging Objects As Critical Path
      - Get Stats
      - Danny's Confluence Page on Placing World Objects
    - Save All
    - Fog Placement
      - Fog Types
        - AutoFog
        - GroupFog
      - Erasing Fog
      - Fog Groups Tied To World Objects
    - Fog Inspector
      - Fog Inspector (Advanced Controls)
      - Gated World Objects & Fog Groups
      - Auto Fog Clear Distance
      - Object Opacity Under Fog
  - Danny's Confluence Fog Placement
  - Layout & Nav
    - Layout Tool - Ground & Water
    - Nav Tool
    - Danny's Confluence Nav Page
  - Build Area
  - Characters
  - Locators
    - Pathing Locators
  - Day-Night Setting
  - Save As An Entire Level
  - Play Your Level
- QuestGraph
  - Introduction
  - Creating New QuestGraphs
  - QuestGraph Nodes
    - Main UI:
    - Quest Index:
  - Nodes
    - Start Node
    - Story
    - Connecting Story Nodes to Next Quest
    - Quest
    - Task
      - Types of Tasks
        - Obstacles and Decorations
        - Remove
        - Interact
        - Travel To
  - QuestGraph Flow
  - Quest, Task, Names, and Dialogue - The Loc File
- Create World Object Prefabs
  - Step By Step Instructions
  - Danny's Confluence/Video Example

- [Best Practices](#)
  - [Level Design Style Guide](#)
  - [Plant Placement Process](#)

## Introduction

This document covers everything you need to know about how to build a level on Project Eden. It will walk you through how to Pull the latest from the project's repository, how to check in edited files, how to use Unity and our Eden specific level design features, and its scripting tool. If you have any questions, corrections, or additions that you think needs to be added, please contact [jeff.sondin-kung@jamcity.com](mailto:jeff.sondin-kung@jamcity.com).

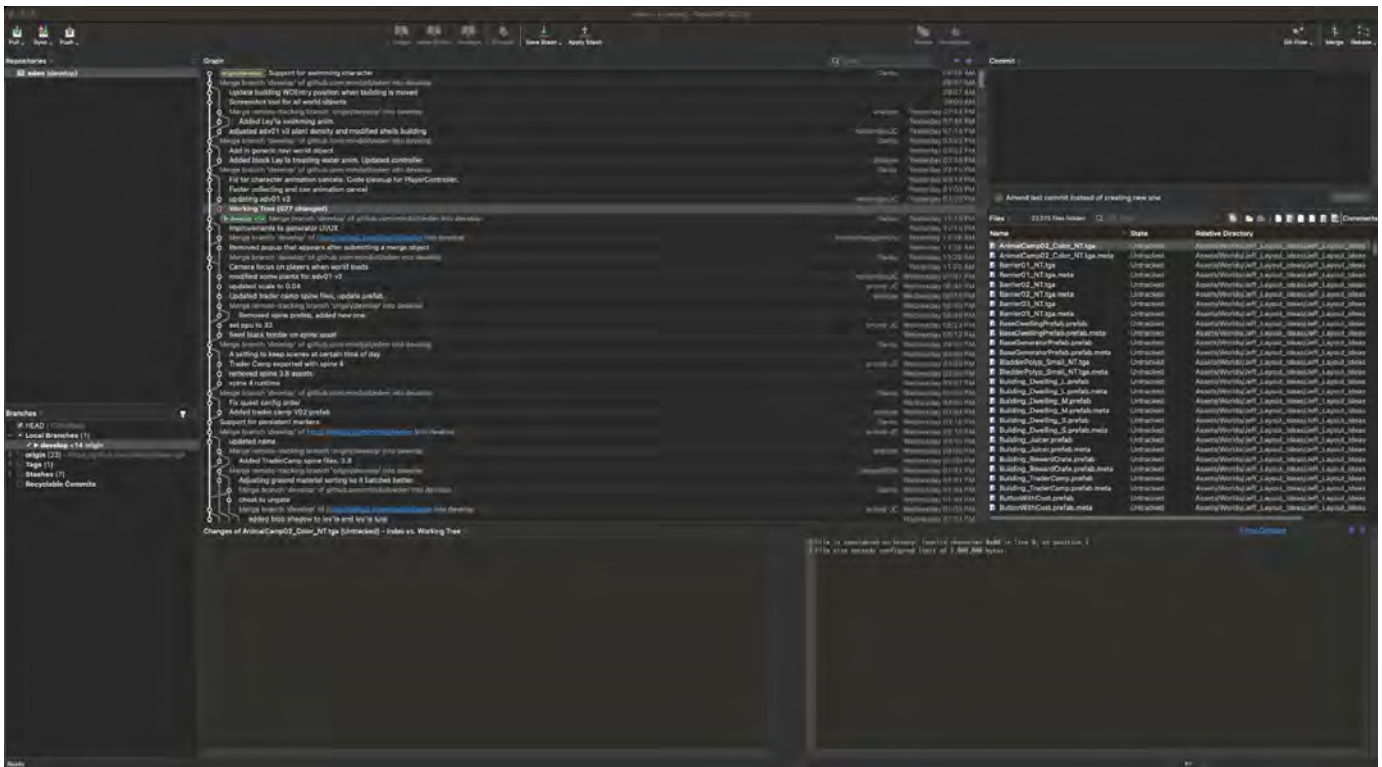


## SmartGit

### Introduction

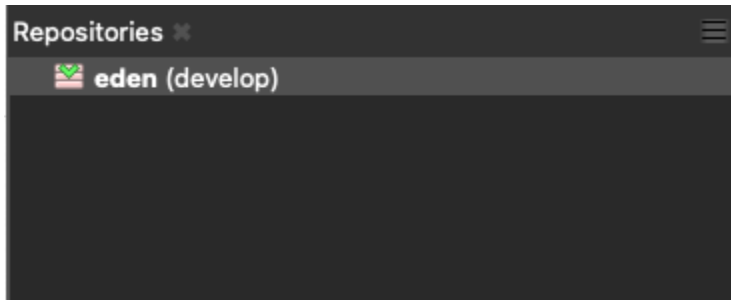
SmartGit is Project Eden's repository all all game assets and code.

Once you have SmartGit set up (Danny Lum or Jim Schuler will help with that), your SmartGit UI will look like this:



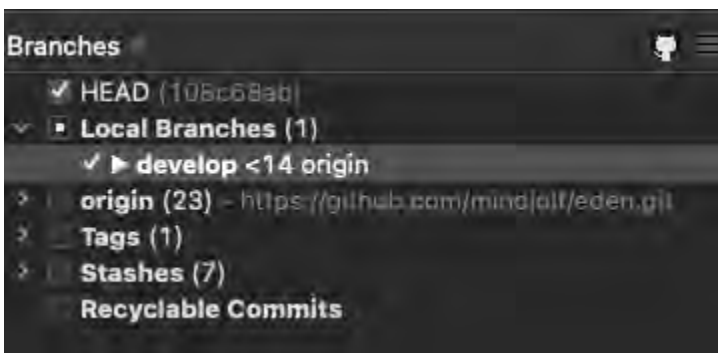
First, we'll highlight each section of the UI and point out what they do. Later, we'll go into [more detail](#) as we go through an actual session with SmartGit.

### Repositories Window



develop is our current repository for the game. Always have this selected. If you don't have it selected, nothing will show up in any other window. (Don't freak out if everything is blank). :)

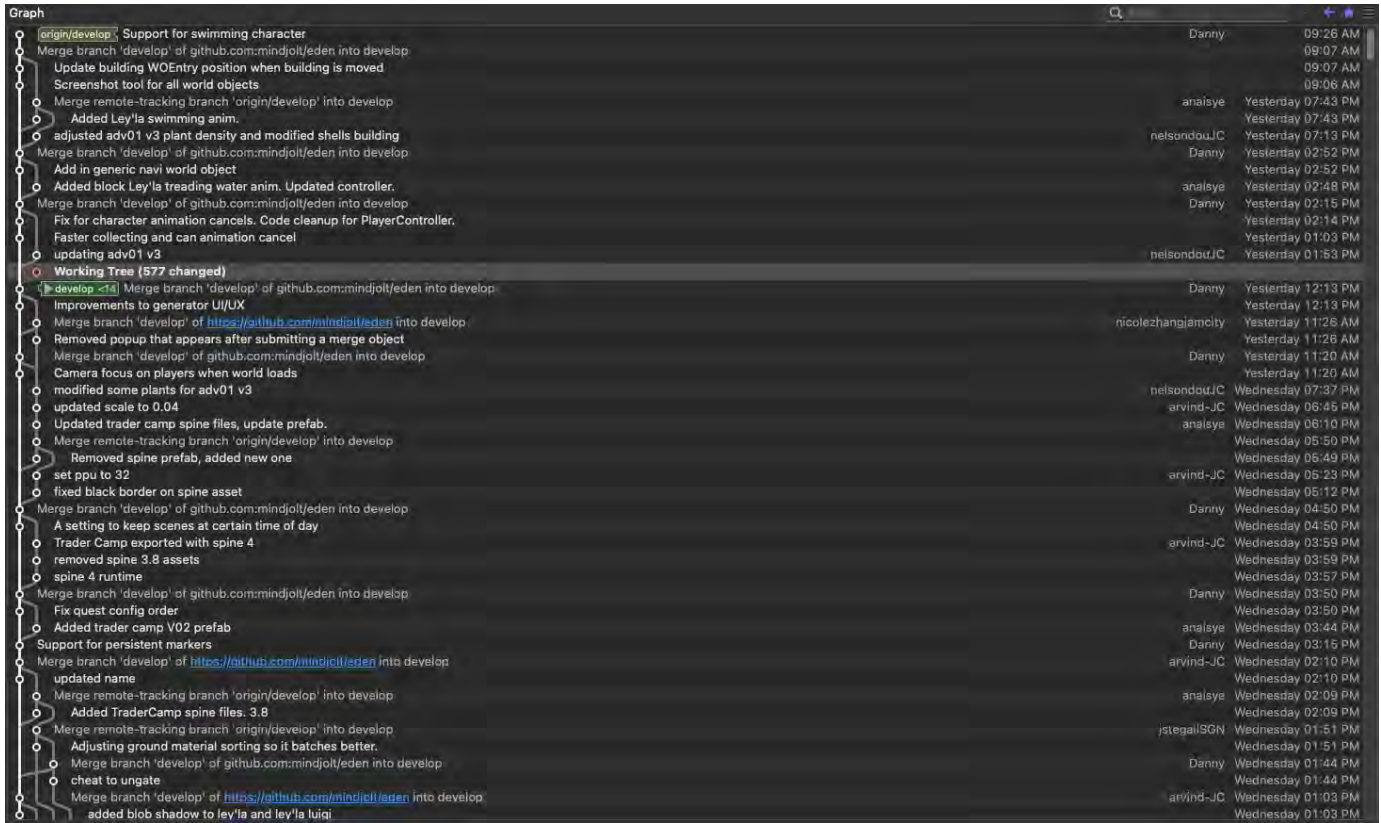
### Branches Window



If branches are made, they will show up here. For the most part, LDs won't need to branch as their work is on-going.

In the screenshot above, it shows that develop is a child branch of Local Branches (and is the only one here). Head is also checkmarked. These are the only lines you need to make sure are checked for your work.

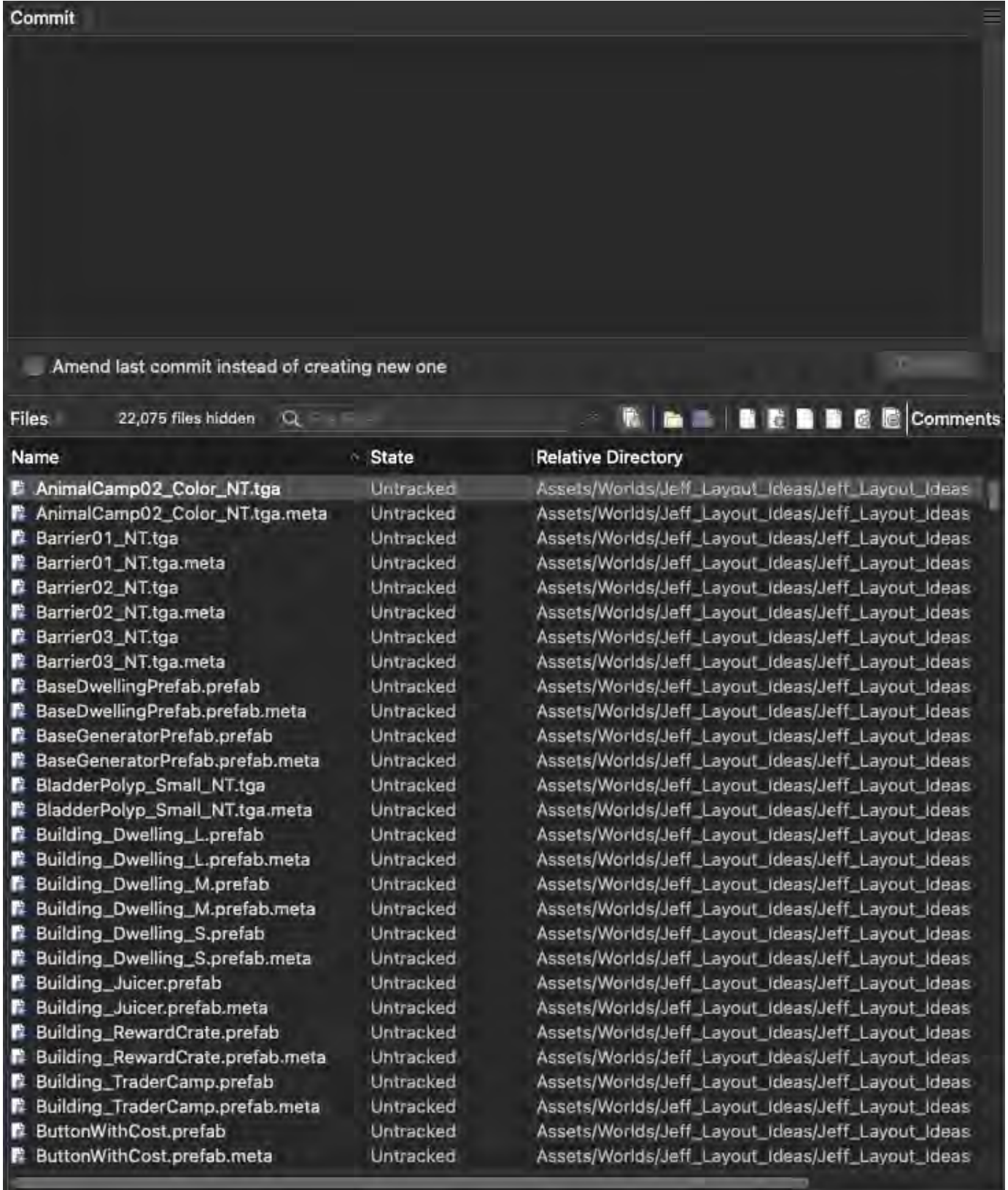
### Graph Window



This is a running history of everything that is committed and pushed into the develop branch. In P4 speak, this is everything that has been checked into the main repository.

### Commit Window





We'll go into [more detail](#) about this window but essentially, this is where you'll be picking files that you want to commit and then push to develop (checking in files in P4 parlance). This is also where you can write your commit notes (check in notes).

## SmartGit Walkthrough

This will be a walkthrough of a typical session for a level designer.

### Start Of The Day - Pull

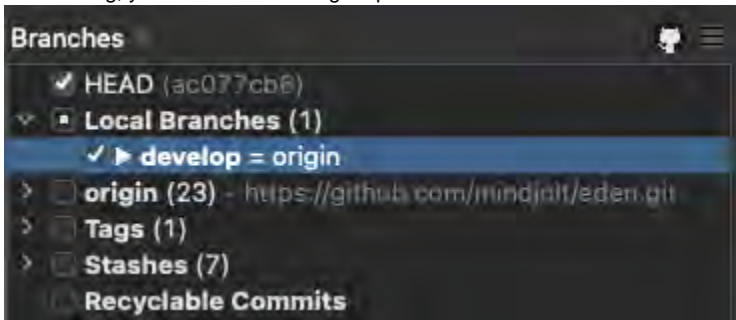
The first thing you should do is Pull (sync to head in P4 speak). You're already set to be pointing to develop so there's no pointing to anything. Just Pull. Upper left corner of the SmartGit UI.



You're most likely going to be behind the develop branch. Here, there are 19 pushes to develop that you are behind.

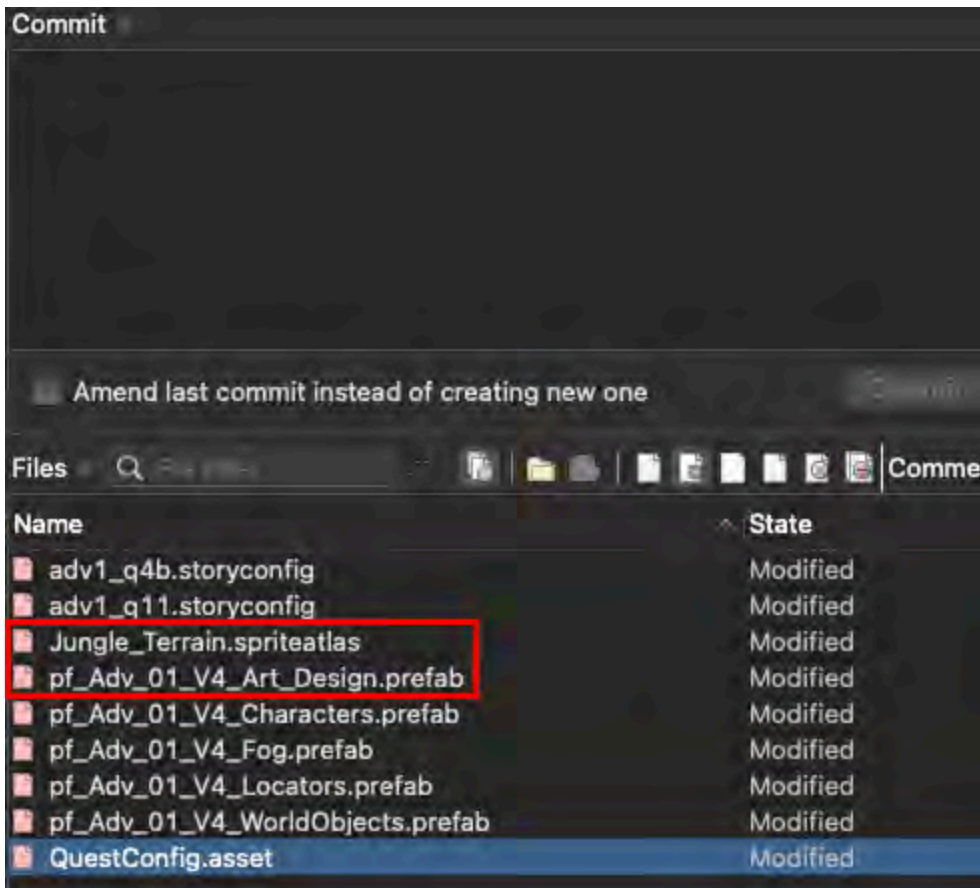


After Pulling, you should be all caught up.



### Commit & Push Prep

At the end of a significant session of work or at the end of the day, you should Commit and Push your work up to the develop branch.

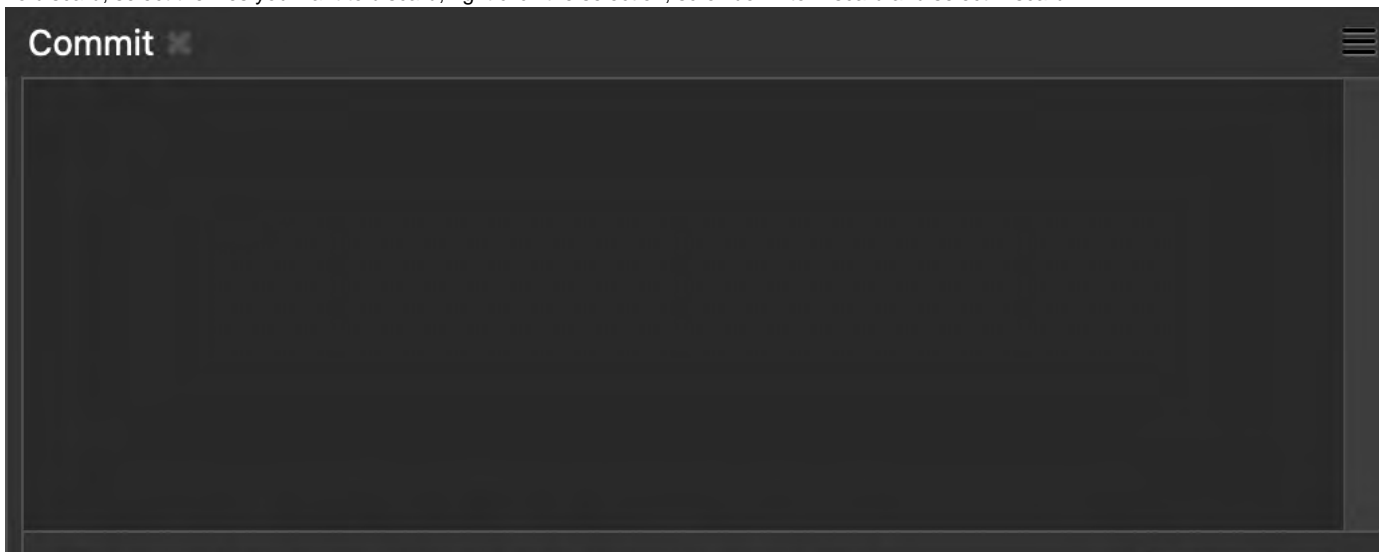


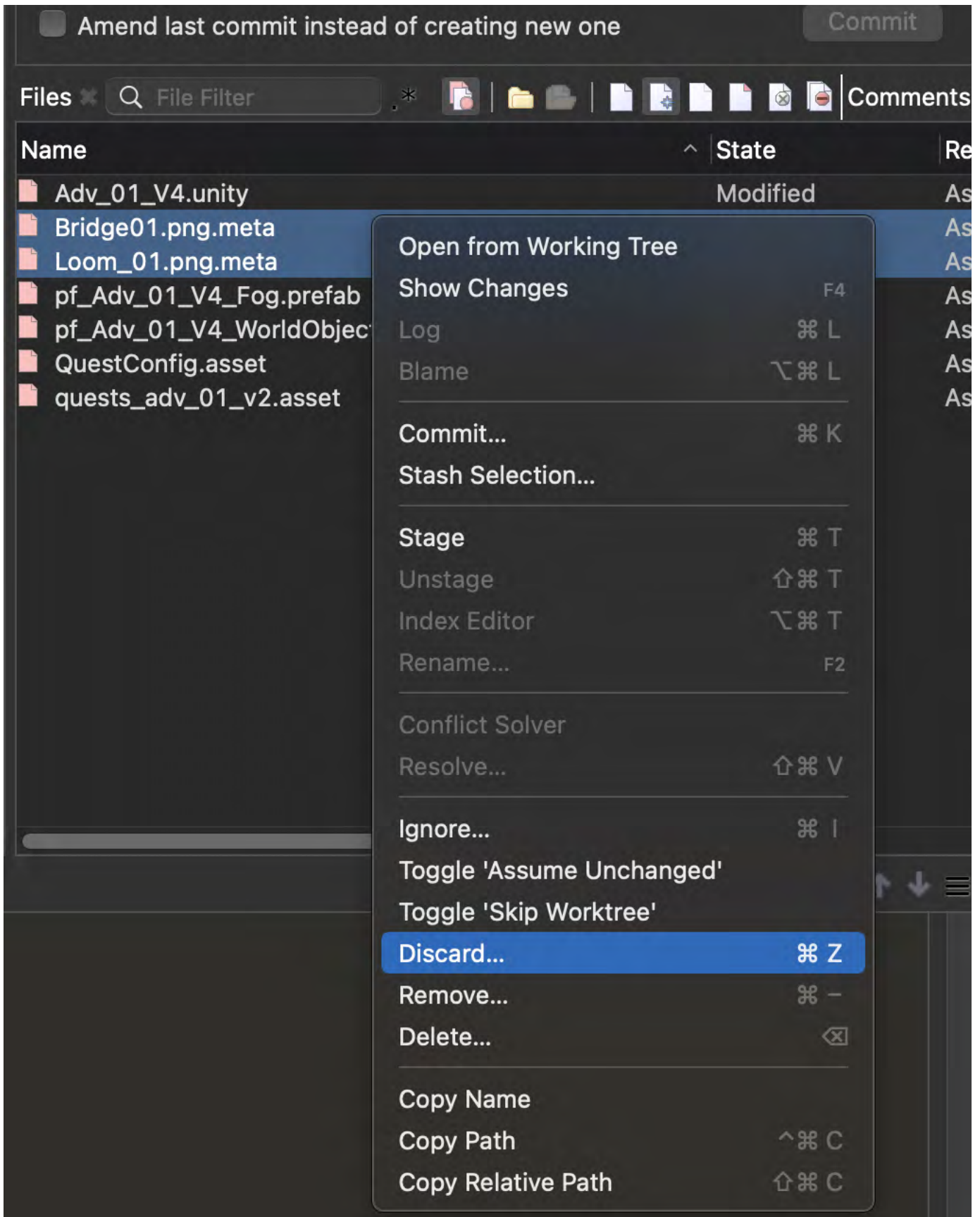
But before you make your Commit, make sure to Discard files that you did not work in (I'll go over which files these are) or leave certain files along (checked out locally).

Files you'll never work in and should be discarded because they're auto-checked out:

- (level).unity (yes, this is not something you need to save in Unity)
- \*.spriteatlas
- \*Art\_Design.prefab
- \*WorldLights.prefab
- \*FanLizardManager.prefab
- \*.meta
- \*.png
- \*.asset
- Anything coding .cs

To discard, select the files you want to discard, right click the selection, scroll down to Discard and select Discard.





Files you will be working with in rough order of frequency:

- \*WorldObjects.prefab
- quests\_\*.asset



- \*Fog.prefab
- \*Locators.prefab
- \*Characters.prefab

If you create a new level, under State in the Commit window, the term Untracked will be used instead of Modified. This means that the game trunk, develop, hasn't had these files pushed to it yet. Meaning, they are only local for the time being. So in this case, you'll be Committing and then Pushing these new files up to develop. All of the files you expect to be working with will be on this list as well.

- \*WorldObjects.prefab
- quests\_\*.asset
- \*Fog.prefab
- \*Locators.prefab
- \*Characters.prefab

But will also include:

- \*.meta files related to all of the above
- WorldConfig.asset (you're adding to the list of available levels)

### Commit & Push (Really this time)

Before you Commit and Push your files, be sure to Pull from the latest in develop.

When you've cleaned up your Commit file candidates (Discarded any auto-checked out files) and you've visually verified what you actually want to Commit, select the files you mean to Commit.

Name	State
pf_Adv_01_V4_Fog.prefab	Modified
pf_Adv_01_V4_WorldObjects.prefab	Modified
QuestConfig.asset	Modified
quests_adv_01_v2.asset	Modified

Right click the selected files, find Commit:

Name	State	Re
pf_Adv_01_V4_Fog.prefab	Modified	As
pf_Adv_01_V4_WorldObjects.prefab	Modified	As
QuestConfig.asset	Modified	As
quests_adv_01_v2.asset	Modified	As

Open from Working Tree

Show Changes F4

Log ⌘ L

Blame ⇧ ⌘ L

---

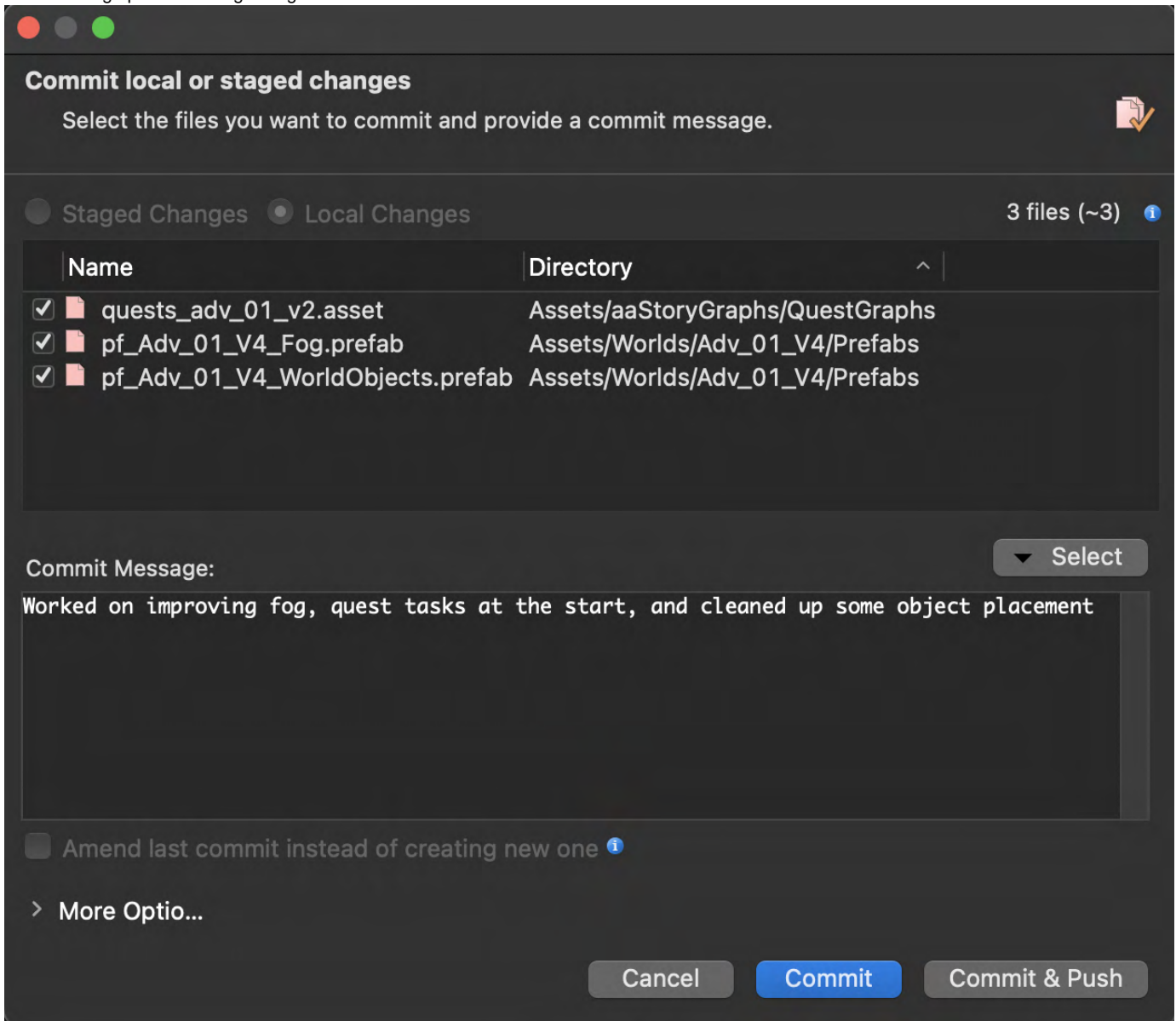
**Commit...** ⌘ K

Stash Selection...

You'll notice that QuestConfig.asset is Modified, in the Commit list but NOT included in the commit itself. This is because QuestConfig allows you to order the quests in the game. But for testing purposes, you will need to order your current working level at the top so that it plays first every time you play/test the level in Unity.

There's no other way to keep this file checked out but not included in your Commit/Push. So you have to manually keep it out of your commits. This is something that's fairly easy to remember once you get used to it.

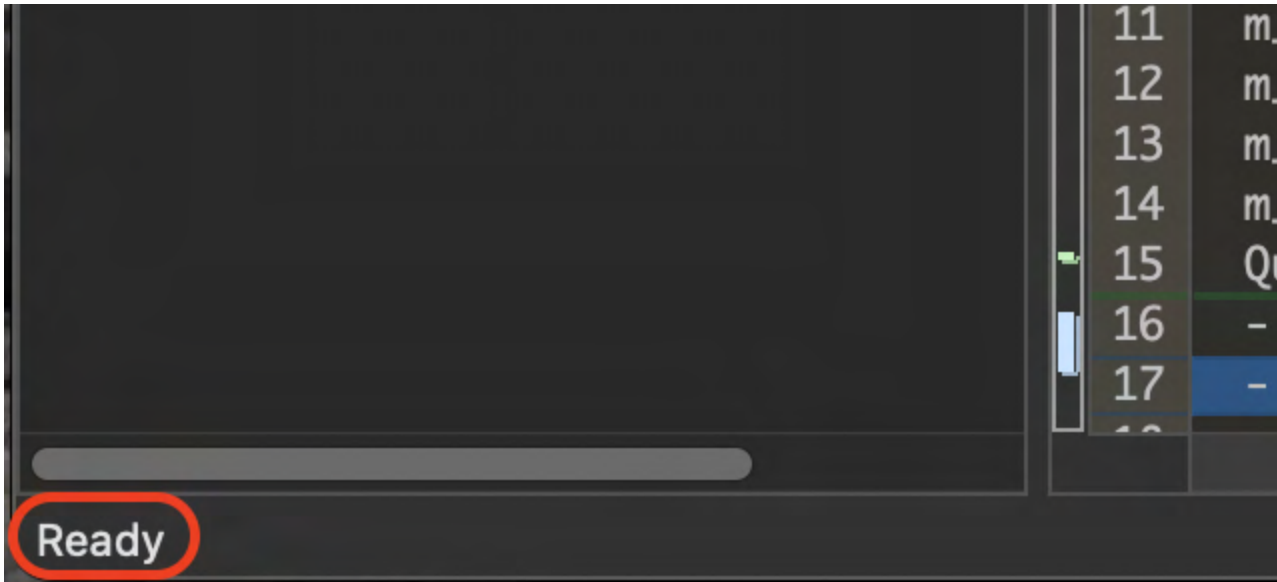
This will bring up the following dialog box:



Go ahead and Commit your files. But wait, you're not done yet. Hit the Push button:



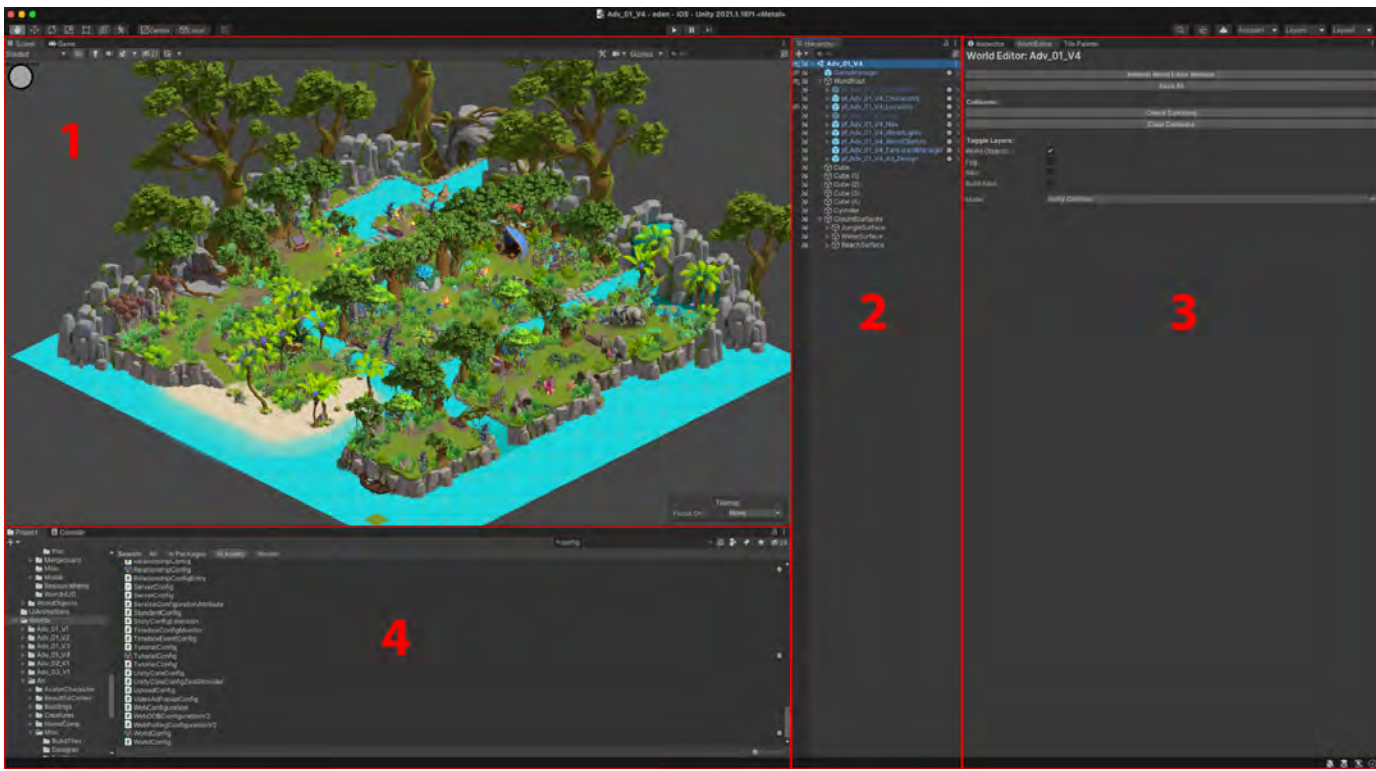
Alternatively, you can just hit the Commit & Push button in the Commit dialog box above. And that's it! You've successfully Pushed your work up into develop. And if successful, you'll see the Ready note at the bottom left corner of GitHub:



NOTE: If an error is showing for i2 when attempting to pull, try discarding, pulling newest, and replacing the i2 spreadsheet again.

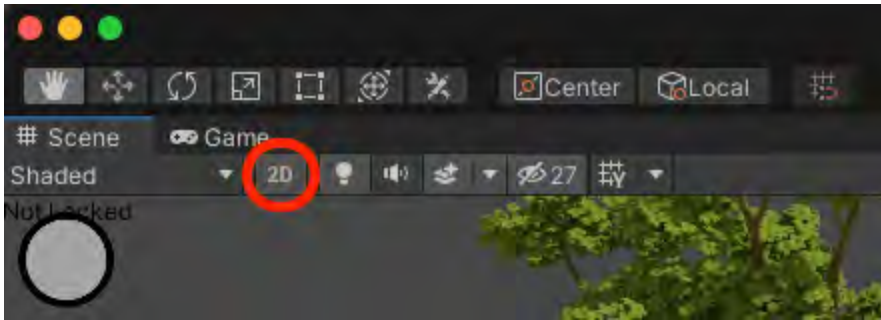
## Unity

### Introduction



This is the main UI in Unity. You can configure each panel however you like. This happens to be mine.

- 1. This is the main view port for a level, the Scene tab
  - a. Make sure to have 2D selected in the upper left menu bar



2. This is the Hierarchy

a. All editable prefabs will be here



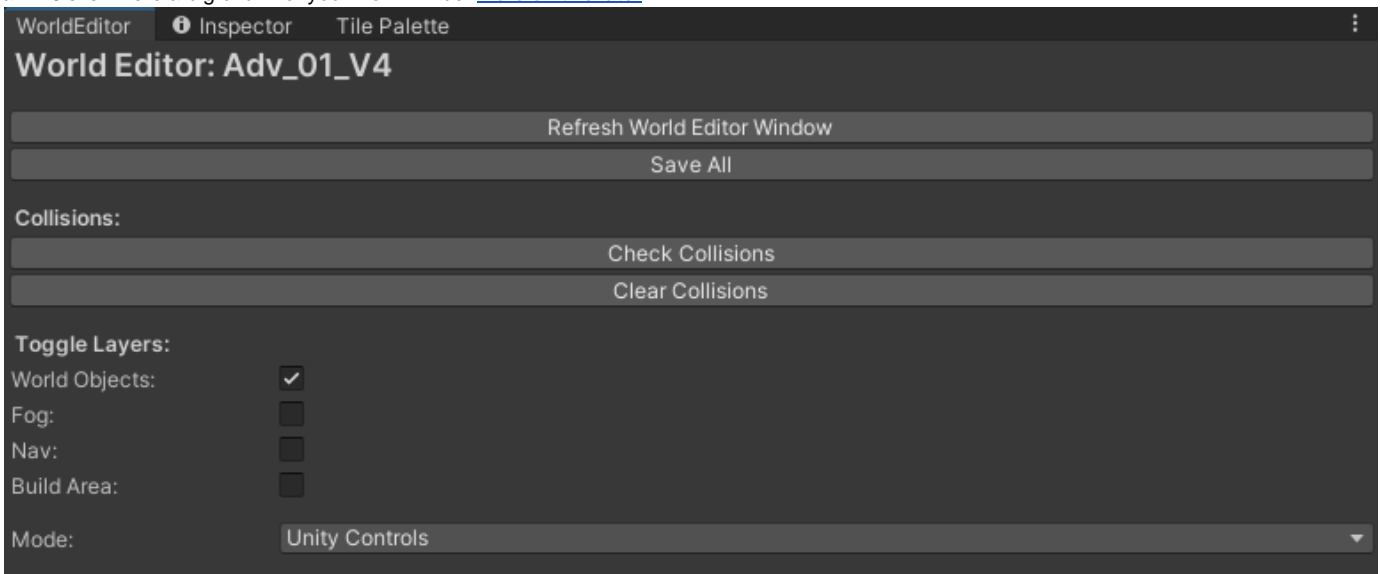
Each prefab can be hidden/viewable by clicking the “eye” area on the far left of

each prefab. The second icon to the right of the eye is the pickable/not pickable toggle.

By far, the most populated prefab will be the WorldObjects PF. [More on this later.](#)

3. This is the WorldEditor panel

a. This is where a big chunk of your work will be. [More on this later.](#)



5. Project Tab

a. This tab is helpful in finding certain files

b. Mostly, it'll be used to find the I2Languages prefab

c. Used for importing text from Google Sheets Localization file

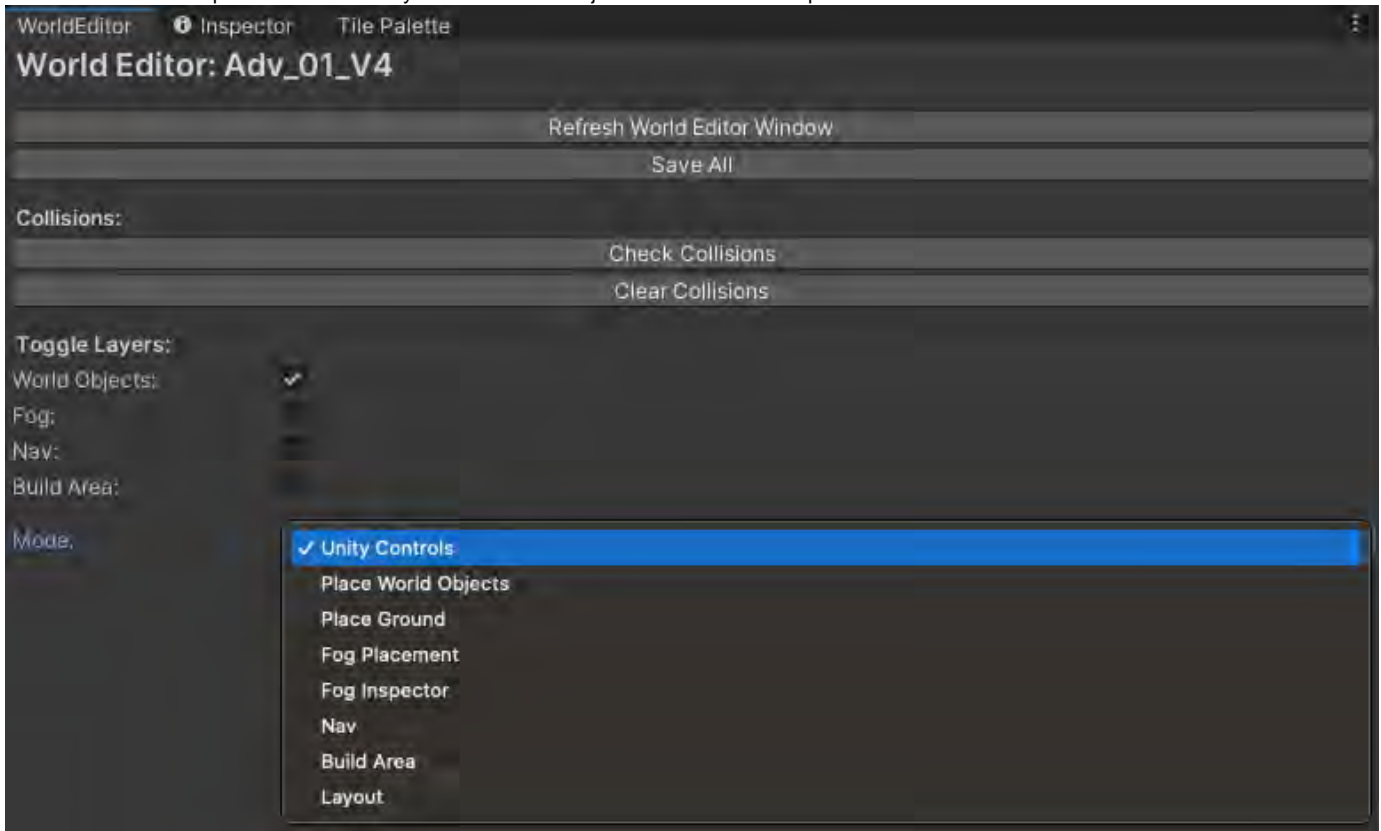
d. [More on this later](#)



## World Editor Modes

There are 8 Modes a level designer can work in. Each with its own set of tools that will allow the LD to place objects, create basic ground planes, add and edit fog, and more.

Easily, the most used Mode will be the Place World Objects mode followed closely by the Fog Placement mode and Fog Inspector. Unity Controls will be used quite a bit as well as you move World Objects around on the map.



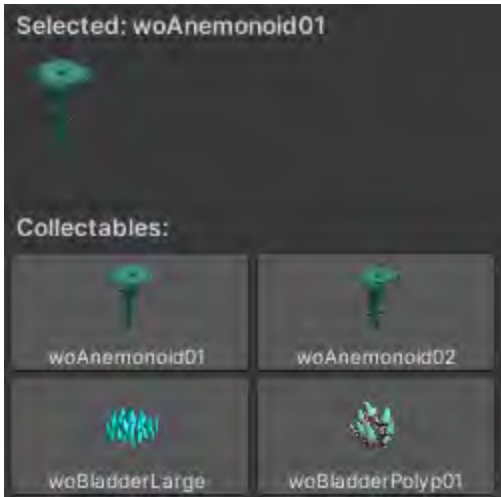
## Object Placement of World Objects





It's a good idea to hit the Refresh World Editor Window button after a fresh pull from GitHub. Once you've opened the Place World Objects mode, go ahead and click on the Refresh World Editor Window button. If the objects below weren't showing up, they will now.

Each thumbnail is an object you can place in the level in the Scene tab. Select an object from the library. A preview of the object will appear above the library to show you what you've selected.



Now, in the Scene view, your cursor will now be the object you've selected along with the footprint grid below it.



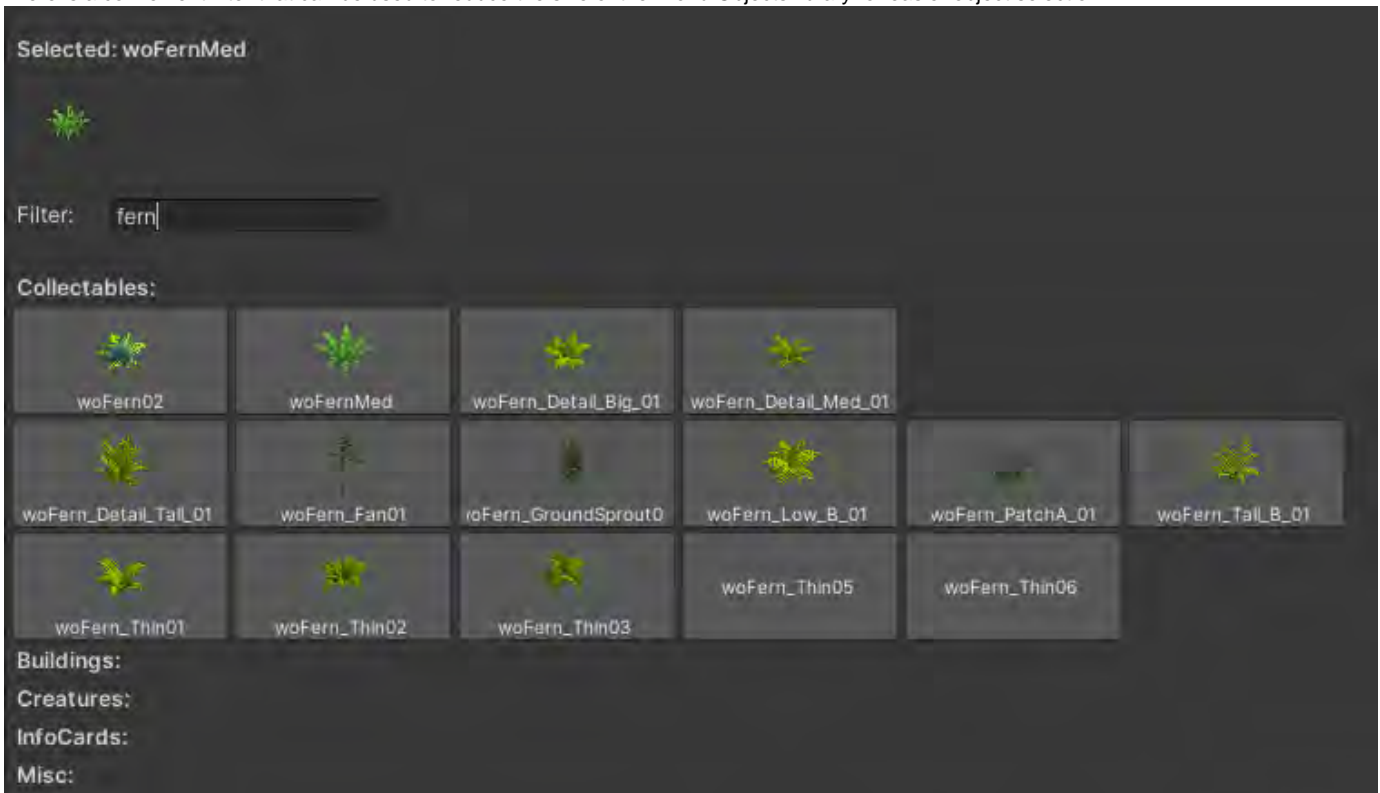
Left click to place the object.

Right click to erase the object.

- Do this by moving your cursor/object's footprint grid square over the placed object's grid square and right click
- You can erase any World Object with this cursor. Doesn't matter what you have selected

## World Objects Filter

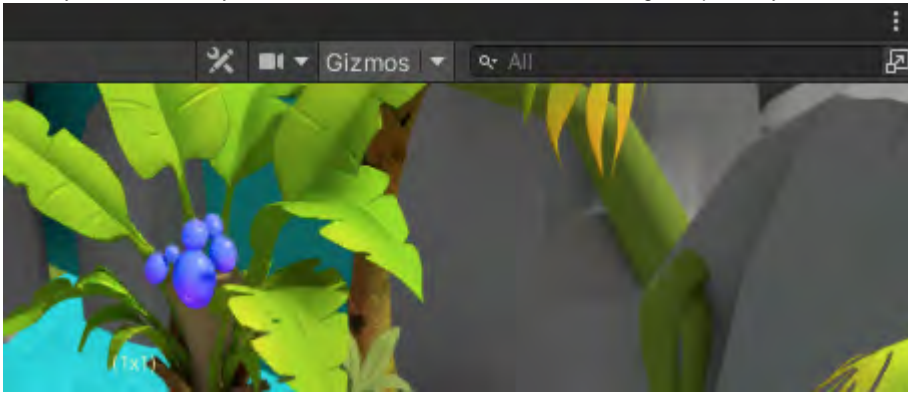
There is a convenient filter that can be used to reduce the size of the World Objects library for easier object selection.



Type in the base object type and every object of that base type will appear as selectable items.



TIP: If you need to see your level without all the noise from the blue grid squares, you can turn them all off with the Gizmos toggle.



This:



Vs This:





## Picking World Objects In The Scene Tab

You will need to pick/select certain objects in the Scene tab to make it [gated](#), [add it to a weight pool](#) for lifting Fog Groups, or to find it's file name to add to a task. Here's how to select a World Object.

- Switch Mode to Unity Controls



- In the Scene tab, press the W key toggle on the Unity move widget (and the little hand cursor will change to an arrow cursor)
- Move the cursor over a blue grid square at the center of a low lying object or at the base of a tall object and press the left mouse button to select it
- It should look like this for a low lying plant below:



- And this for a tree below:



- The footprint grid for the tree is the square in the red circle, at the base of the tree
  - Also notice, trees have a yellow bounding box around them (useful as a selection confirmation) and the center point is higher up on the asset
  - But you select the tree at the base of the tree

## Selecting Multiple World Objects Via Marquee Select

[multiselect.mov](#)

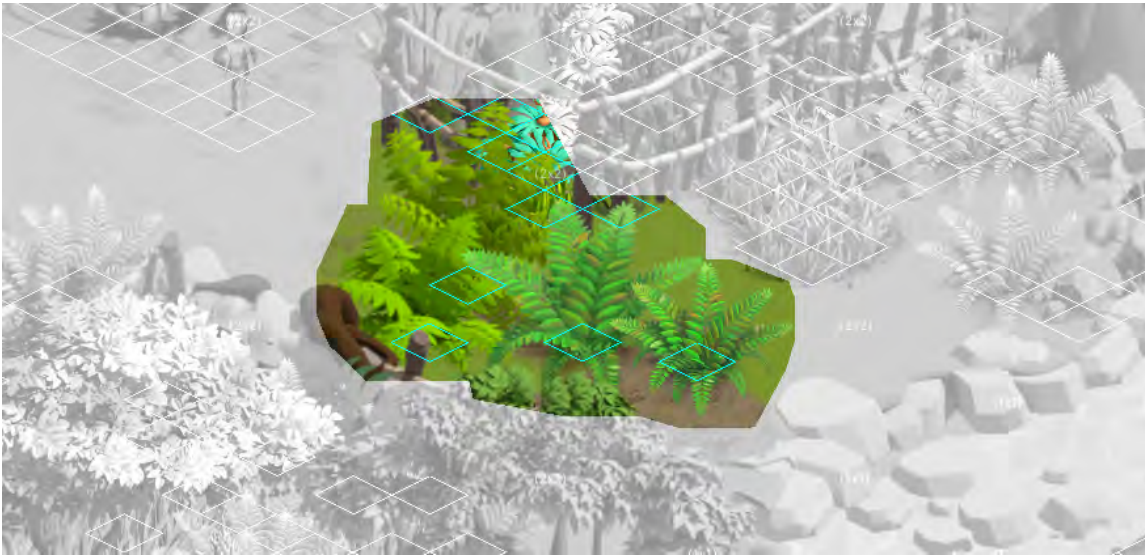


TIP: If you need to search for specific objects in a Fog Group, make sure you append the name of whatever objects are in your Fog Group with the Fog Group ID number. For example, here in the screenshot, I've appended each object with F5. If you multiples of the same object, you'll also want to included numeric increment before the F#.

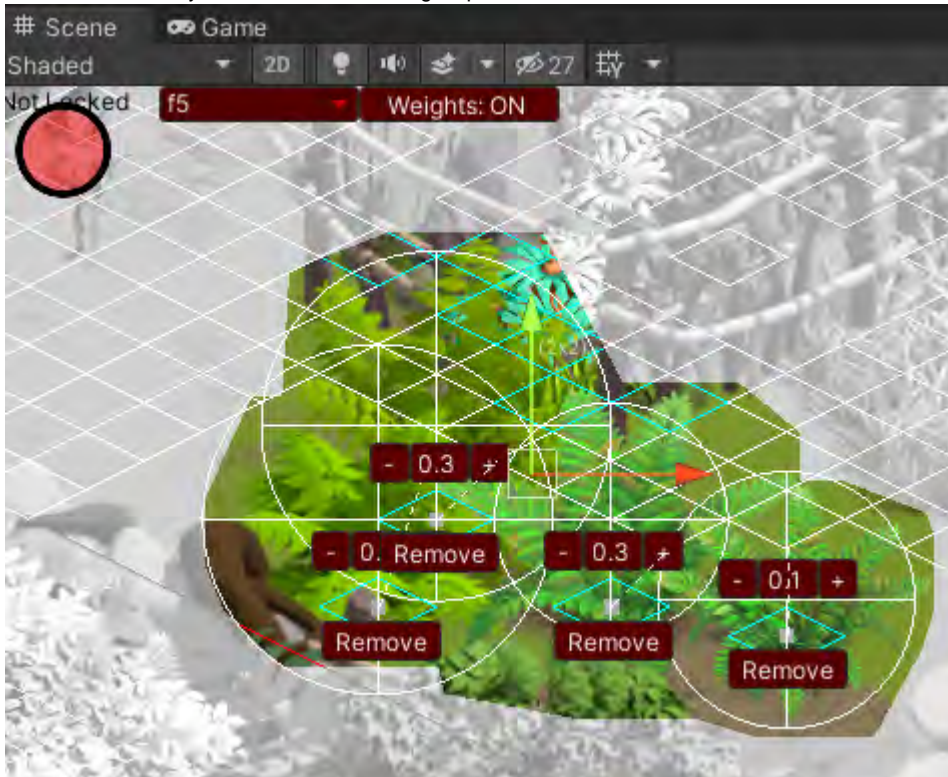


So all I have to do to find them all is use the search field at the top of the Hierarchy and that will filter out everything except the F5 objects.

And this is what it'll look like in the Scene tab:



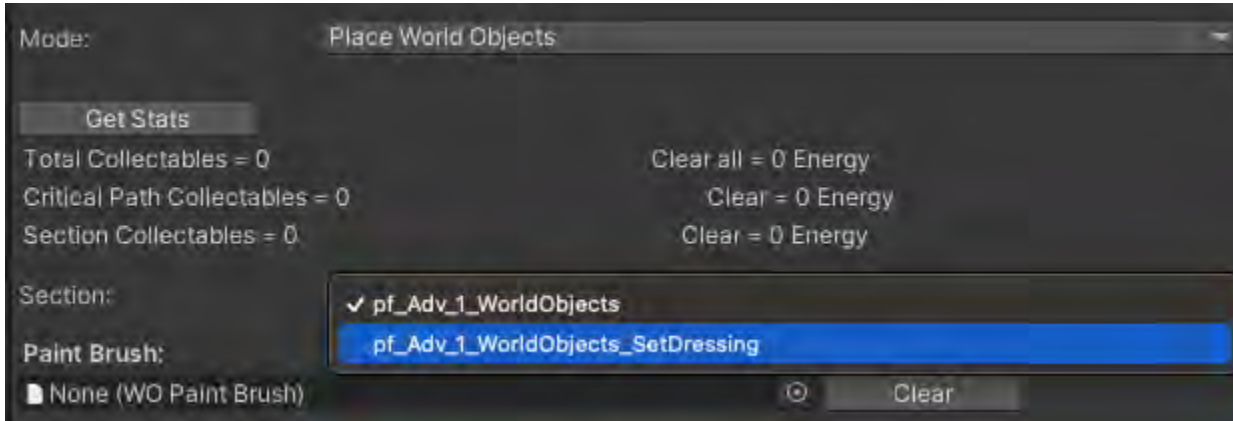
And this is what they look like selected in Fog Inspector mode:



## Set Dressing World Objects

When placing set dressing world objects, create a separate section for them so that they won't be counted toward the playable area's data. These objects will be in places a player will never access so they shouldn't count toward the energy cost and level's object total.

To create this section, right click the WorldObjects prefab> Eden, Add World Object Section. Rename this tab to differentiate from your regular World Objects and move the section into the World Root. Make sure to select this parent when adding in the World Objects from the drop down menu.



If you want to move objects into this section from a different parent, right click the WO parent> prefab> unpack. Then drag the object into the correct place and save all.

**Whenever you are adding a prefab to your level, make sure you check in your .unity file.**

## Gated World Objects

There will be instances where you will want an object to be gated (locked) in order to force the player to complete a task, require a minimum home camp level, or a relationship level.

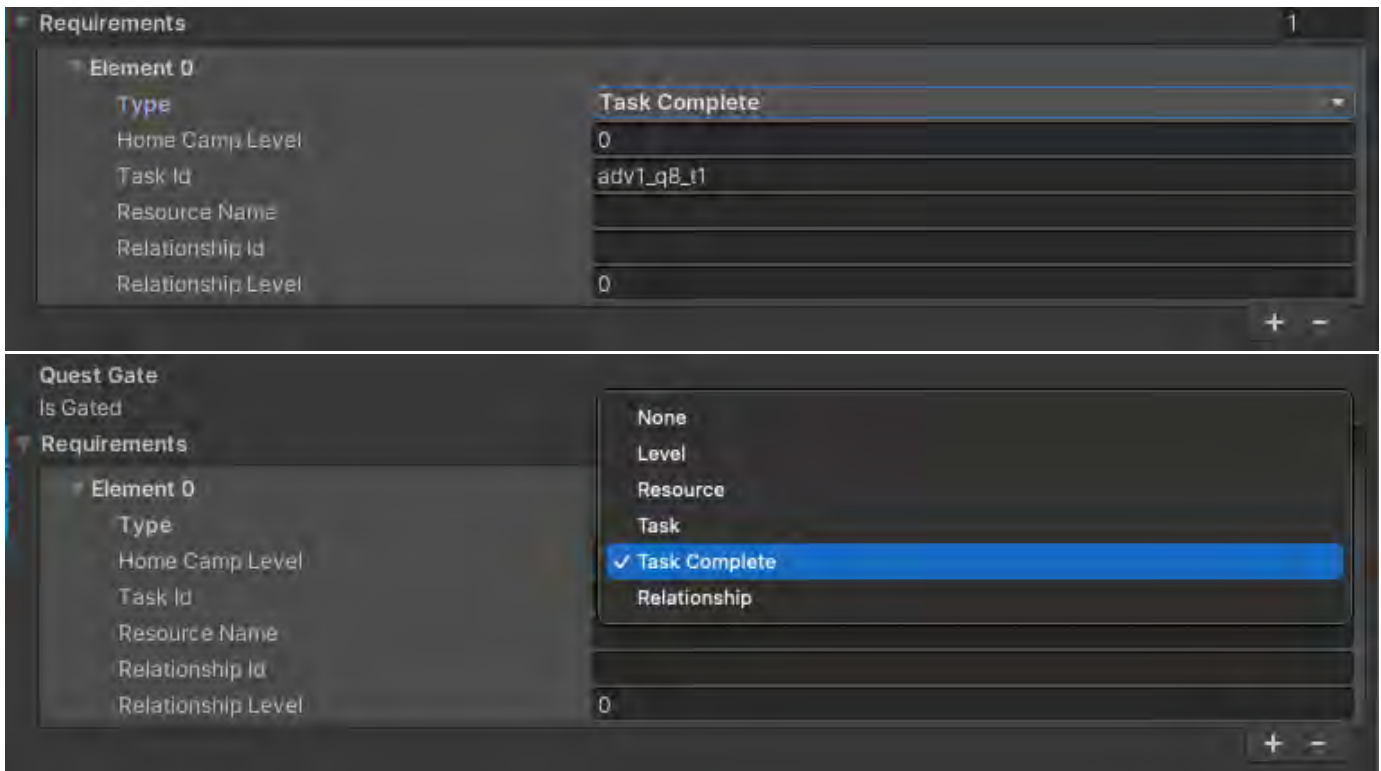
In this example, a woLog will be used as a gated World Object. Here's how you do it:

- Select the object you want to be gated. In this case, woLog2

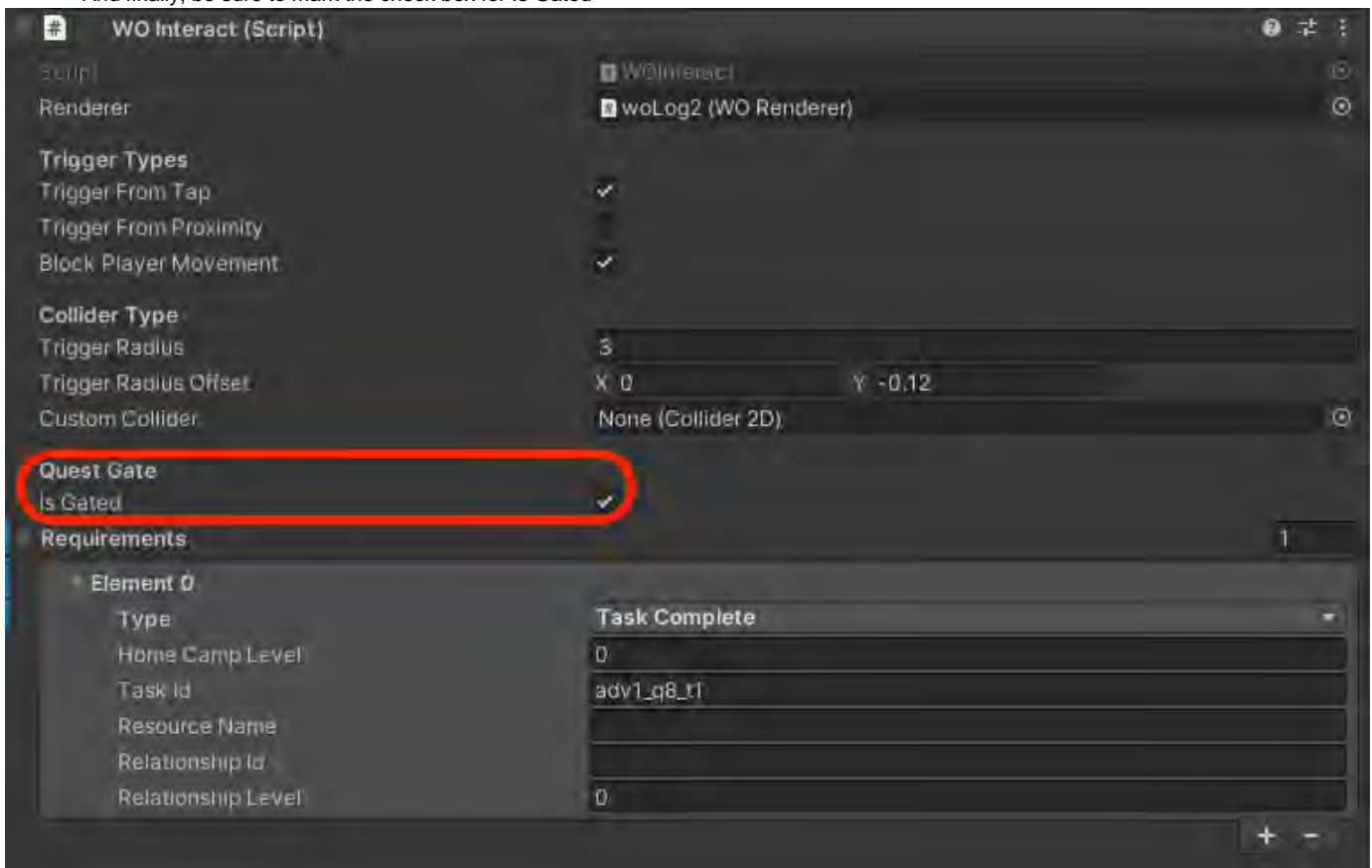


- You'll see the object is also selected in the Hierarchy
- Click on the Inspector tab
- Scroll down to WO Interact -> Requirements -> Type -> dropdown arrow in field



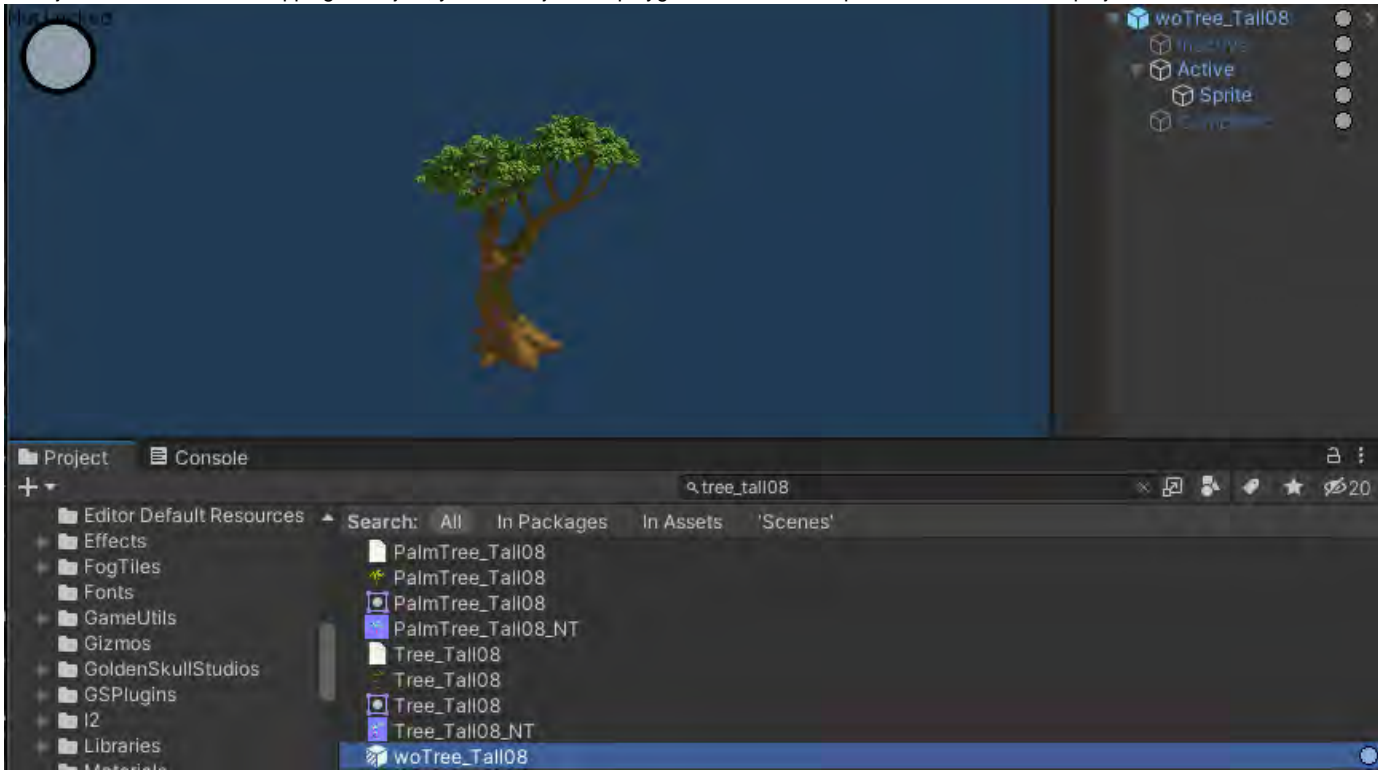


- And in this case, choose Task Complete
- Next to Task ID, type in the name of the task that needs to be completed to ungate this World Object. [More on Tasks later.](#)
- And finally, be sure to mark the check box for Is Gated

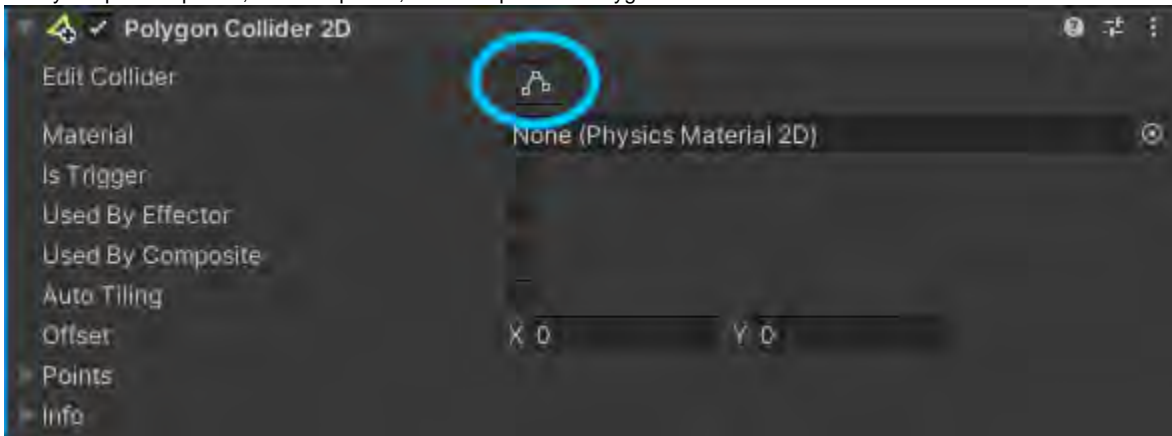


## Adjusting a World Objects's polygon

To adjust the "hitbox" when tapping an object, you can adjust the polygon. Search for the prefab of the item in the project window.



After you open the prefab, in the inspector, Add Component> Polygon Collider 2D> Edit Collider



This will either add a green polygon shape to your object or allow you to edit the existing one. When adjusting the shape, less is more. Try to include only what you need to.

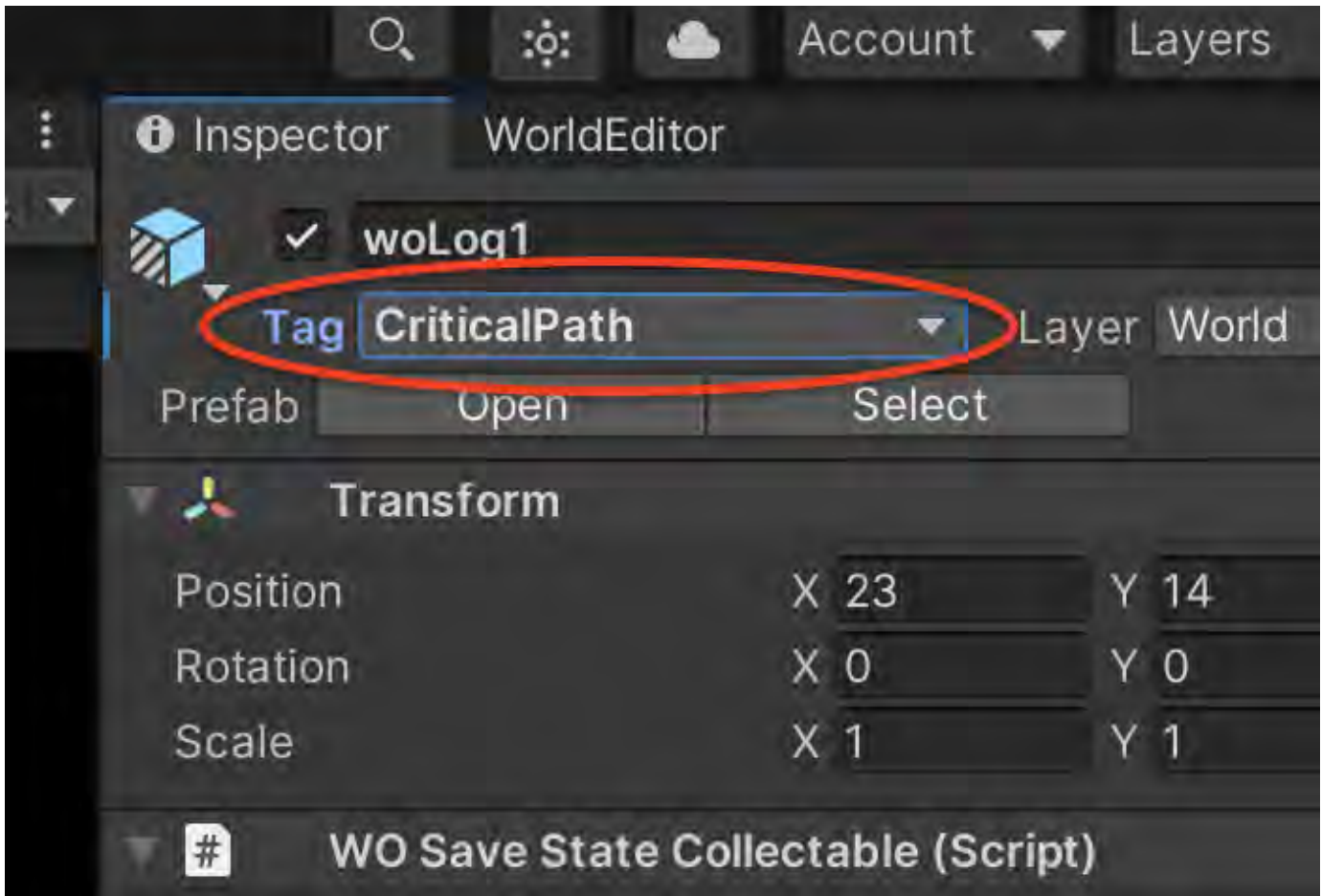


To make the WO use your new shape, drag the header text in the Polygon Collider component into the Custom Collider in the WO Interact component.



### Tagging Objects As Critical Path

The term Critical Path refers to the minimum objects required to be collected to complete a level. To tag an object as a Critical Path object, first select the object, then go to the Inspector tab. At the top of the Inspector section, there is a dropdown field called Tag. Choose Critical Path and that will mark that object as a Critical Path object.

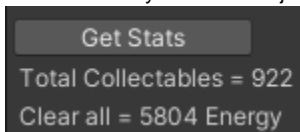


Get Stats now also has a Critical Path breakdown on top of the full level's object stats.



## Get Stats

There's a handy tool/readout just under the Mode: Place World Objects selector.



Hit the Get Stats button from time to time to see what your object count is and how much energy it would take to clear everything from the level. Future improvements - eventually, engineering will split up the level by sections and we can get a more granular view of object totals per section.

## Danny's Confluence Page on Placing World Objects

[Placing World Objects](#)

### Save All

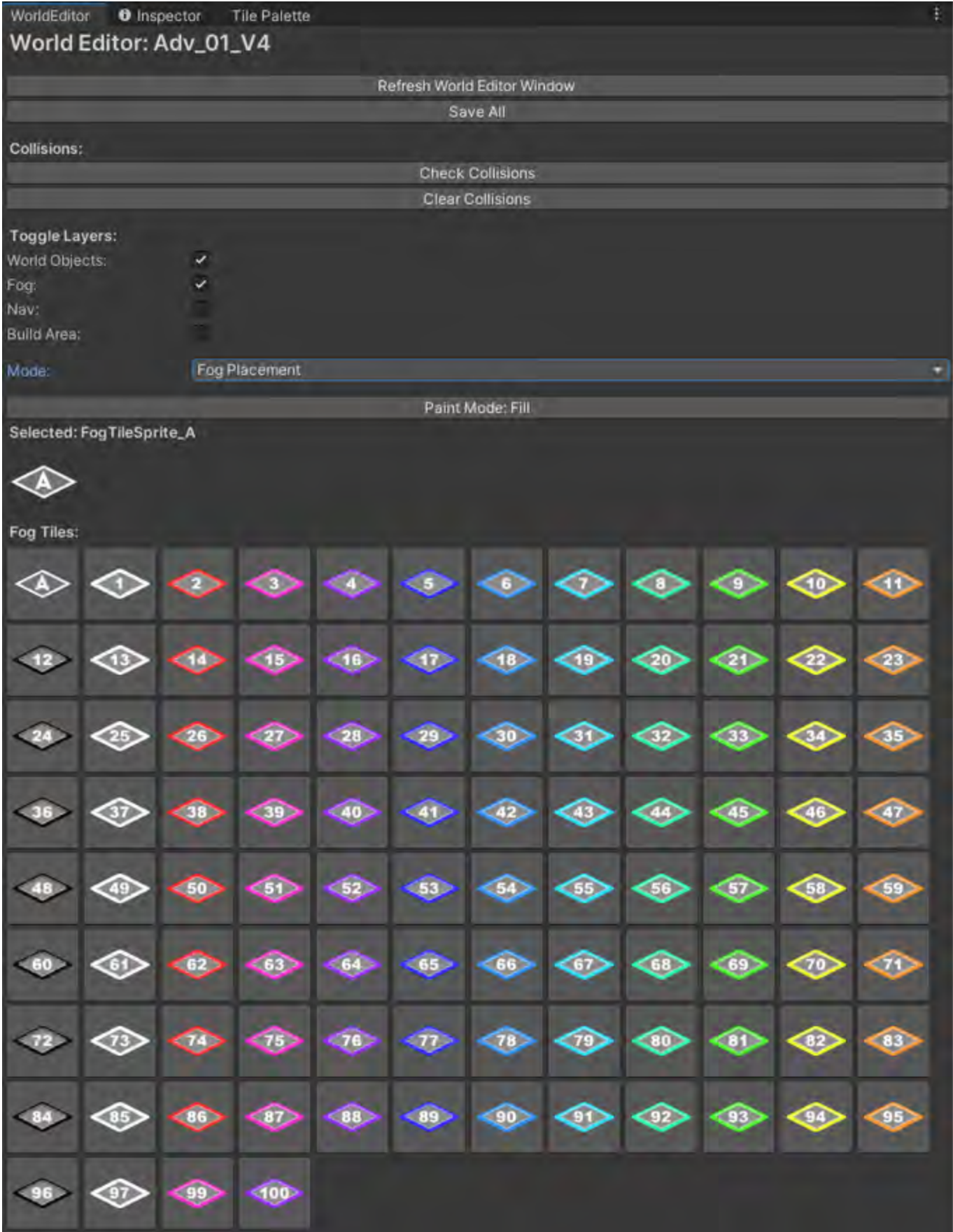
Hit that Save All button (near the top, under Refresh World Editor Window) frequently and often and then hit it some more. This is the only Save button you need to hit. You DO NOT need to (and shouldn't) hit the Save from File or from right clicking any prefabs. Not even the top, root file name for your level.



TIP: If you accidentally place a ton of unwanted objects in your level (I've done this many times), it's easier to Discard (revert in P4 terms) your \*WorldObjects.prefab file in GitHub and then task back into Unity. A popup window asking you if you want to reload your scene will appear. Hit reload and your errant objects will disappear. If you haven't saved in a while, you will also lose your work. So hit that SAVE ALL button many, many, many times while you work!

### **Fog Placement**

In this genre of games, fog is an integral part of the gameplay. It's used to obfuscate objects just beyond the player and the rest of the level. Opportunities for discovery for the player abound and, as a level designer, you can figure out where and how you want players (paying or non-paying) to find hidden secrets as well as the critical path items.



Once a fog tile number is in use, it will show green in the editor.



## Fog Types

There are 2 types of fog. A = AutoFog and Numbered = FogGroup.

By far, the AutoFog will be the fog you use the most as it's not tied to any particular World Object. And it works like this:

### AutoFog

- Select <A>
- Select Paint Mode: Fill or Paint Mode: Paint
  - Fill = Draw a rectangular and the fog unit will fill the space (below)







- Drag a rectangle (left); let go to fill (right)
- Paint = Places individual grid squares with fog unit (below)



Remember to [Save All](#).

GroupFog

See [Fog Inspector](#) for details around how GroupFog works and how it's set up.



## Erasing Fog

- To Erase any type of fog:
  - Right mouse click + Fill & release right mouse click or Paint will erase any fog unit

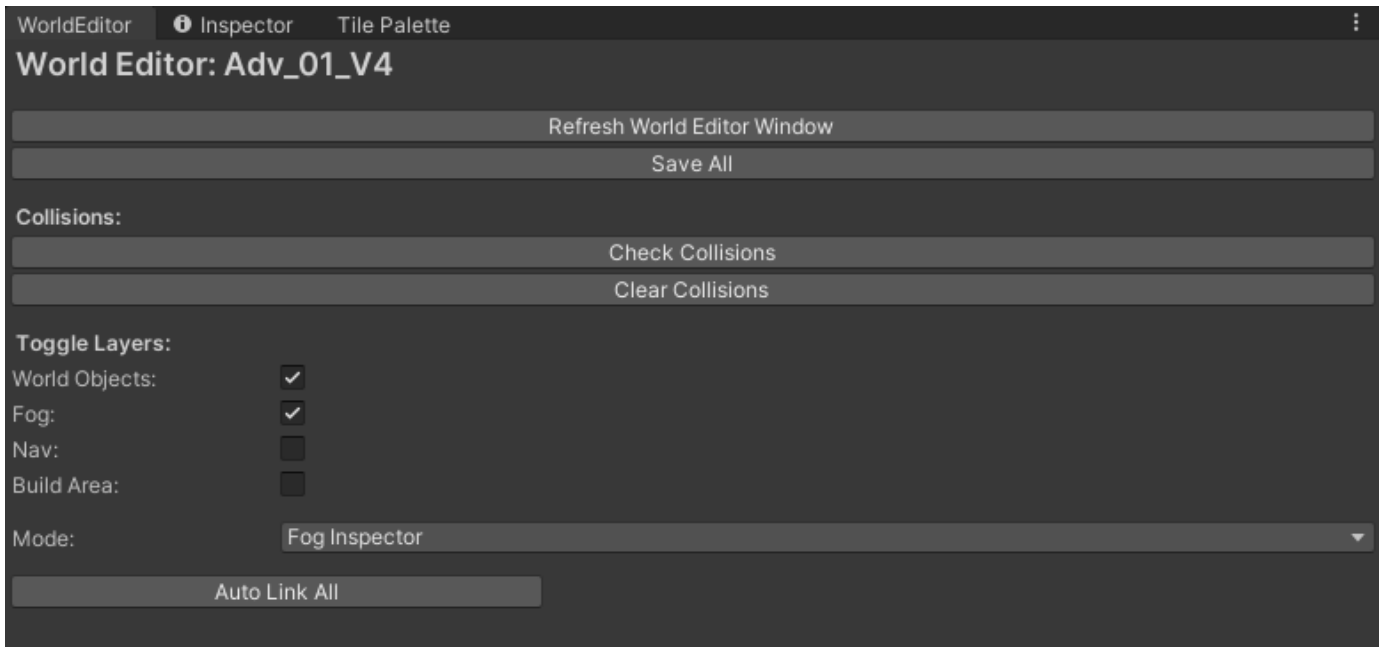
## Fog Groups Tied To World Objects

This is a way to uncover an entire group of fog by tying at least one object to a Fog Group <##> such that when that object(s) is interacted with by the player, that associated Fog Group will lift.

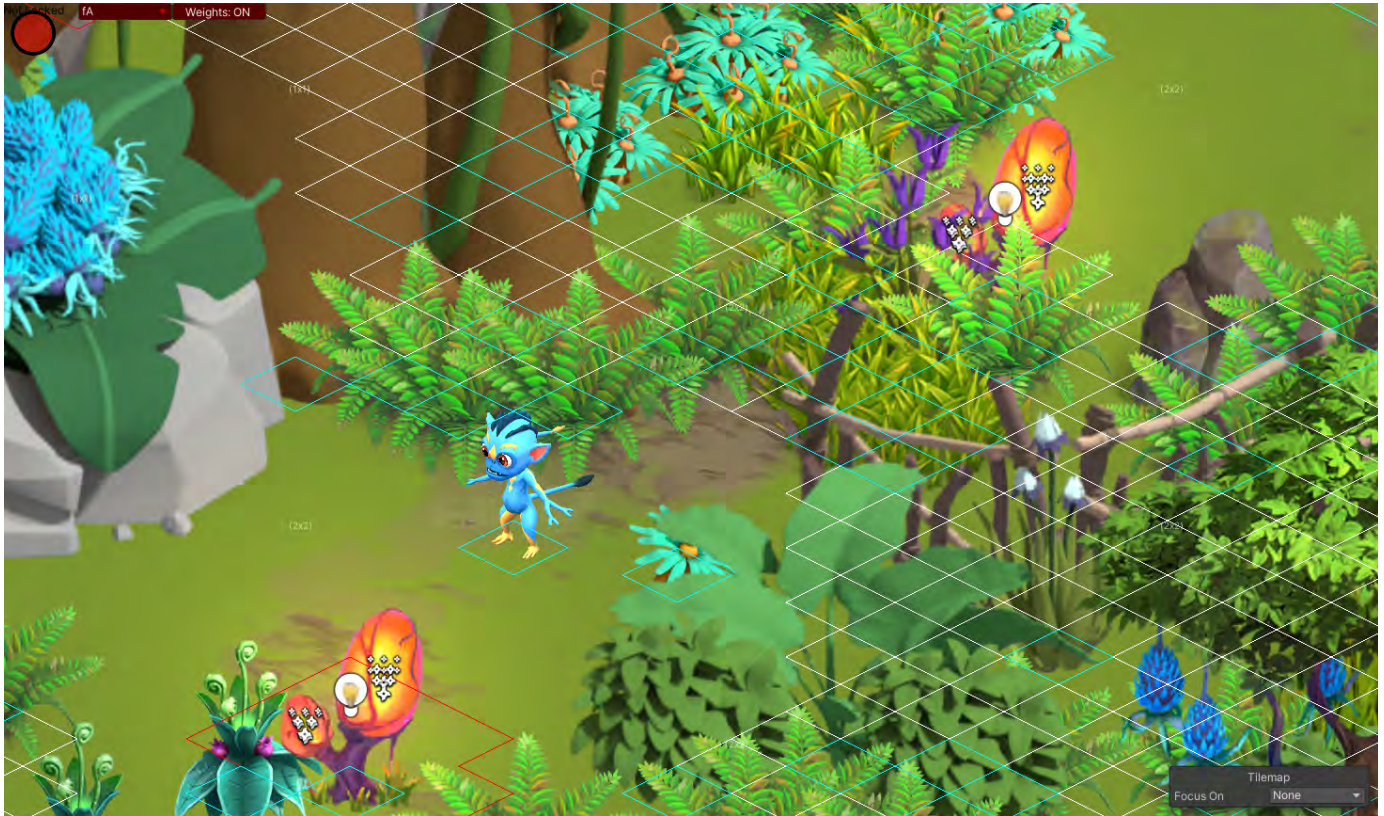
A weighting system assigns weights to the object(s) (.1 through 1.0) and if the player collects at least a total weight of 1.0 in objects, that associated Fog Group will lift.

The next section will explain how to author the tying of World Objects to a Fog Group.

## Fog Inspector



This is the Fog Inspector mode. Once in this mode, the work is actually done in the Scene tab.



In the upper left corner of the screen, you'll see a pair of new, red menu items.



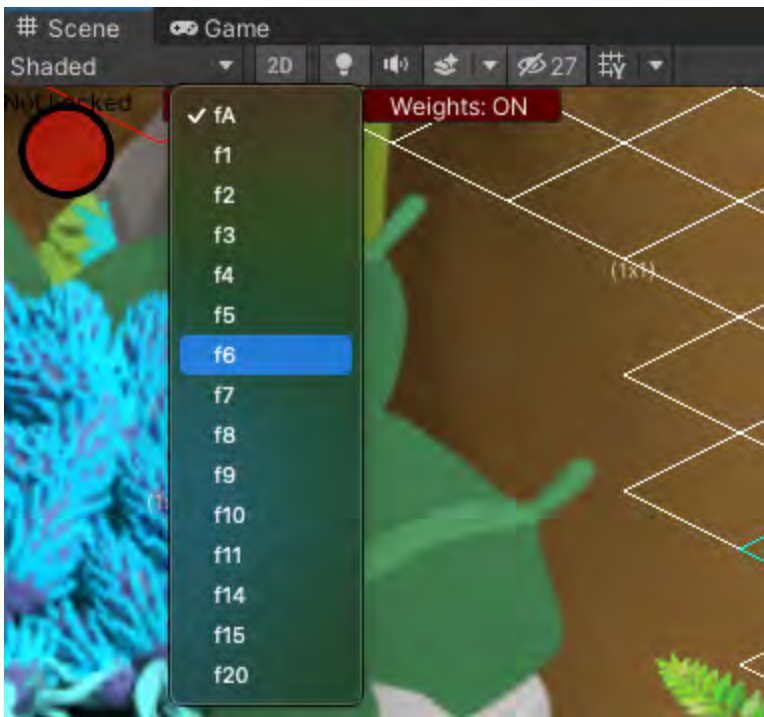
The menu item with fA and the dropdown arrow is the Fog Group selector.

The second menu item, Weights: ON/OFF is a toggle. I have not yet found a use for Weights: OFF so leave it in the ON position for all Fog Group work for now.

In this example, we'll be looking at Fog Group f6.







Here's a closer look at the objects tied to Fog Group f6:



Each object tied to f6 has a +/- on either side of a number (.1 to 1.0). These are weights associated with each object. If a total added weight of 1.0 is met from picking these objects, the associated fog will lift.





While in Fog Inspector mode, select any WO that you want to include in the fog unlocking weight pool of objects and click on the Add button for that object.

This will add the object to the weight pool of objects (below).



You'll now see that associated object with a [-][1.0][+] above it's grid footprint and a dotted line pointing to the closest edge of the Fog Group (in this case, its Fog Group f6).

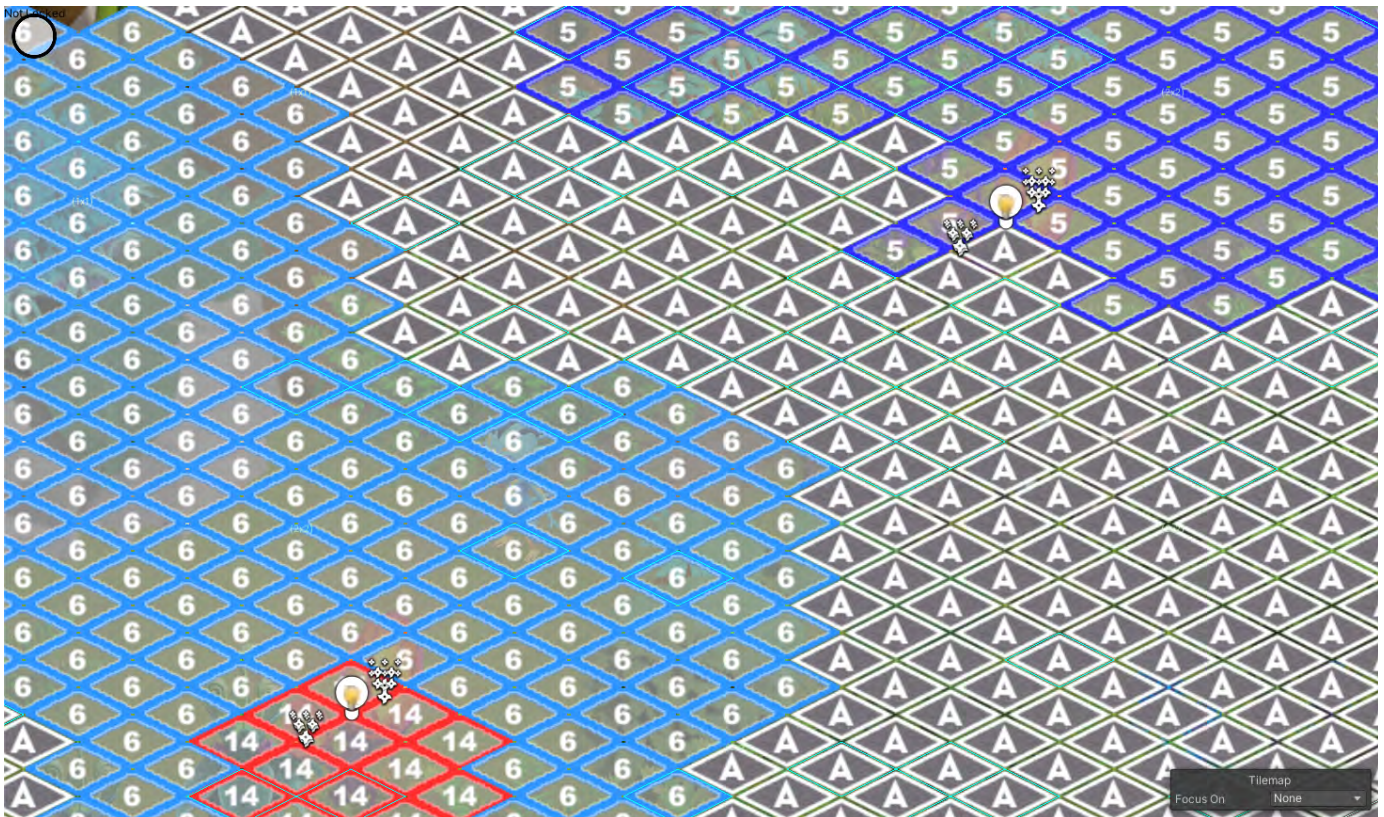


You can decrease/increase the weight of the object (.1 to 1.0).

Also, with the object selected, there will be a Remove button. This will remove the object from the weight pool. It won't, however, delete the object. Just the weight.

Stacked: Fog Inspector Mode vs Fog Placement Mode





When you're done editing your Fog Groups, remember to [Save All](#).

### Fog Inspector (Advanced Controls)

Once you're comfortable with using the Fog Inspector, here are some shortcuts to quickly remove, convert, or select fog groups.





- Remove Fog Group - Enter a fog group number and hit "Remove". This will delete all the fog tiles that are assigned to that fog group.
- Convert Fog Groups - Used to convert an existing fog group to another specified number.
- Active Fog Groups - Click any of the tiles and the camera will frame that fog group in the scene.

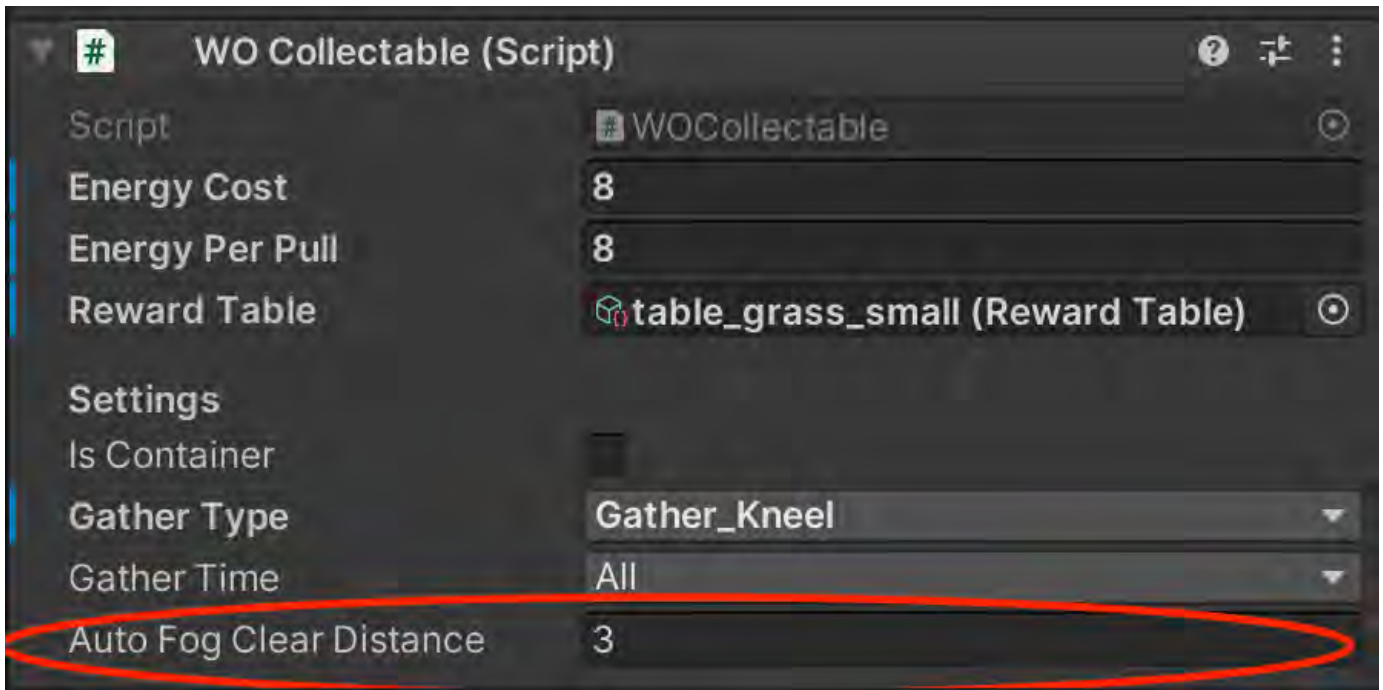
**\*NOTE:** Using the "Remove Fog Group" and "Convert Fog Groups" options will automatically save the changes, so be absolutely sure before hitting any of those buttons.

## Gated World Objects & Fog Groups

Gated World Objects, [more on that here](#), can also be tied to Fog Groups. They just require the player to satisfy a predetermined requirement to ungate the object and then, when the object is interacted with, the Fog Group associated with that object will lift.

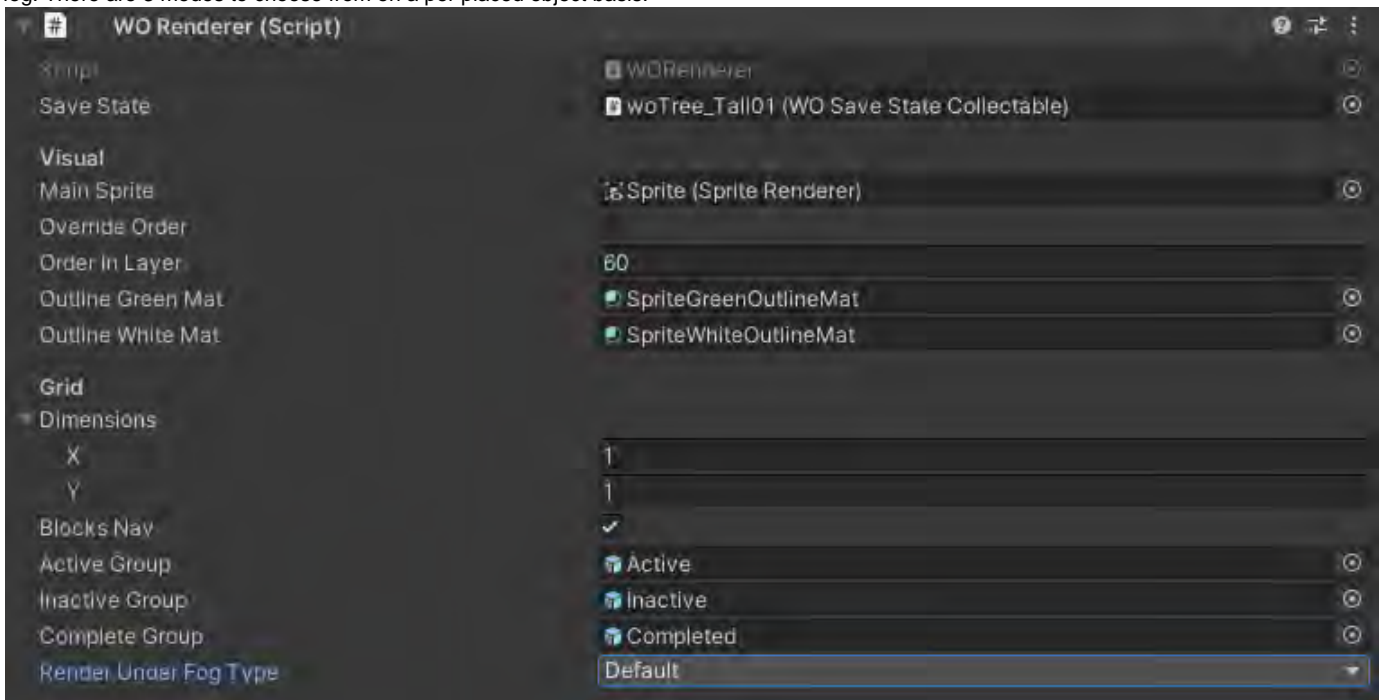
## Auto Fog Clear Distance

By inspecting a World Object in the hierarchy, the distance of fog revealed when removing an object can be adjusted.

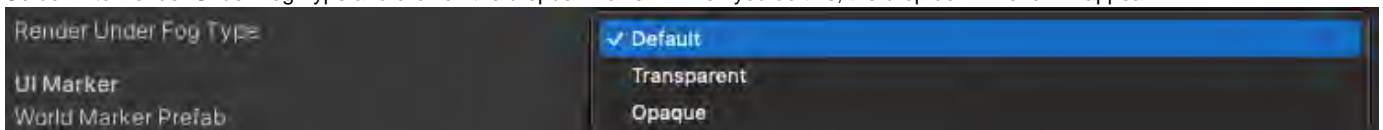


### Object Opacity Under Fog

World Objects are defaulted to be hidden under fog. However, there will be times when you'll want to have certain WOs stick out from under the fog. There are 3 modes to choose from on a per placed object basis.



Go down to Render Under Fog Type and click on the dropdown arrow. When you do this, the drop down menu will appear



3 settings: Default, Opaque, Transparent

- Default - objects are hidden under fog and then shown when they are revealed OR when near edge of fog
- Opaque - fully visible under fog
- Transparent - peeking through fog



Use case examples:

- POIs will sometimes be needed to poke through the fog to attract the player
- Semi-opaque treasure can be seen to lure players into non-critical path location

Danny's Confluence Fog Placement

[Fog Placement](#)

Layout & Nav

In Layout mode, a basic ground plane can be added. The grid can be either Filled or Painted in, grid square by grid square.

**Layout Tool - Ground & Water**



Inspector WorldEditor Tile Palette

# World Editor: Jeff\_Layout\_Id1

Refresh World Editor Window

Save All

Collisions:

Check Collisions

Clear Collisions

Toggle Layers:

World Objects:

Fog:

Nav:

Build Area:

Mode: Layout

Paint Mode: Fill

Selected: Ground

Layout Tile:

Ground Water

Unity File Edit Assets Game Object Component Tools Edit Scene View Hierarchy Inspector WorldEditor Tile Palette

Jeff\_Layout\_Id1 - v4.0 - Unity 2021.1.10f1 (64-bit)

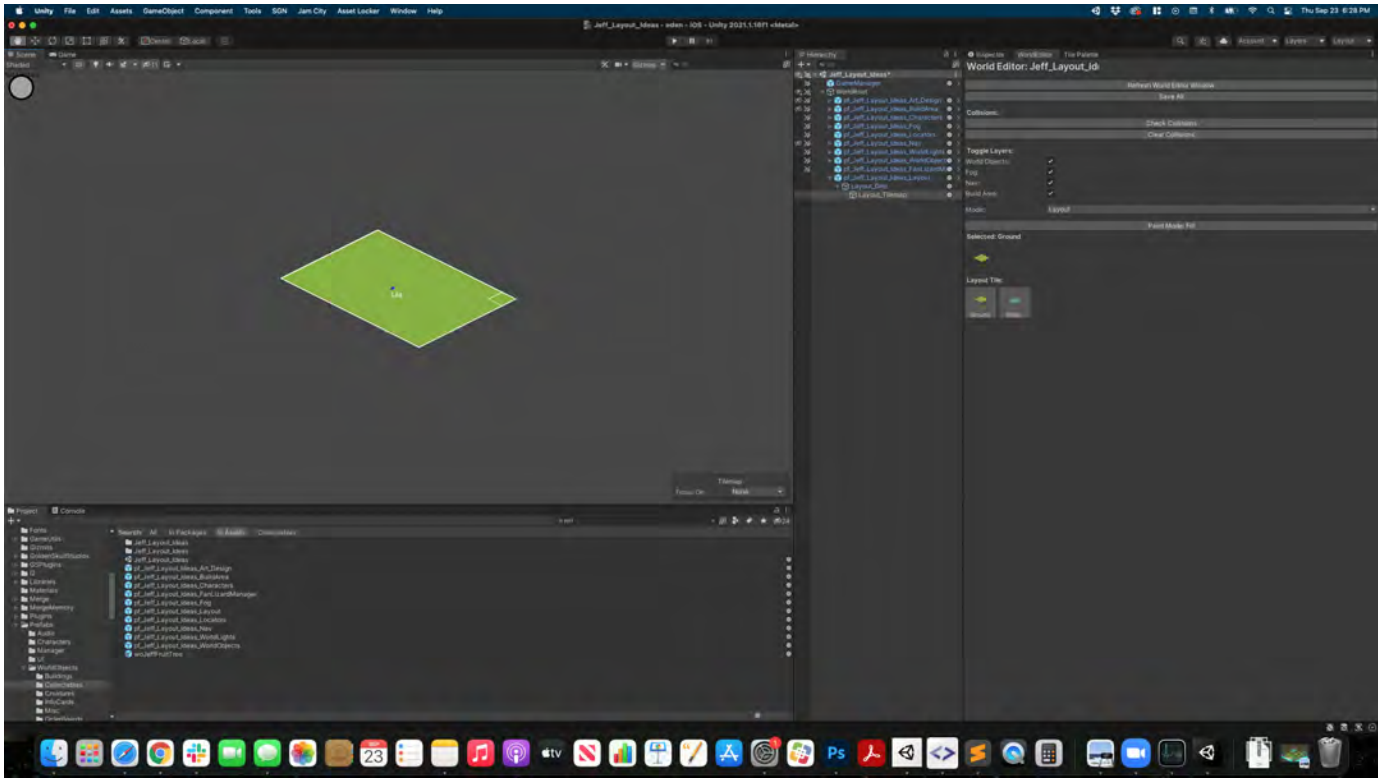
Scene View: A top-down view of a game world showing a large, diamond-shaped area filled with small, light-colored tiles. A single tile is highlighted in a darker shade, indicating it is selected.

Hierarchy: A tree view showing the scene's structure. The selected tile is highlighted in blue. The tree includes folders for 'WorldEditor' and 'Tile Palette', and various objects like 'Ground' and 'Water'.

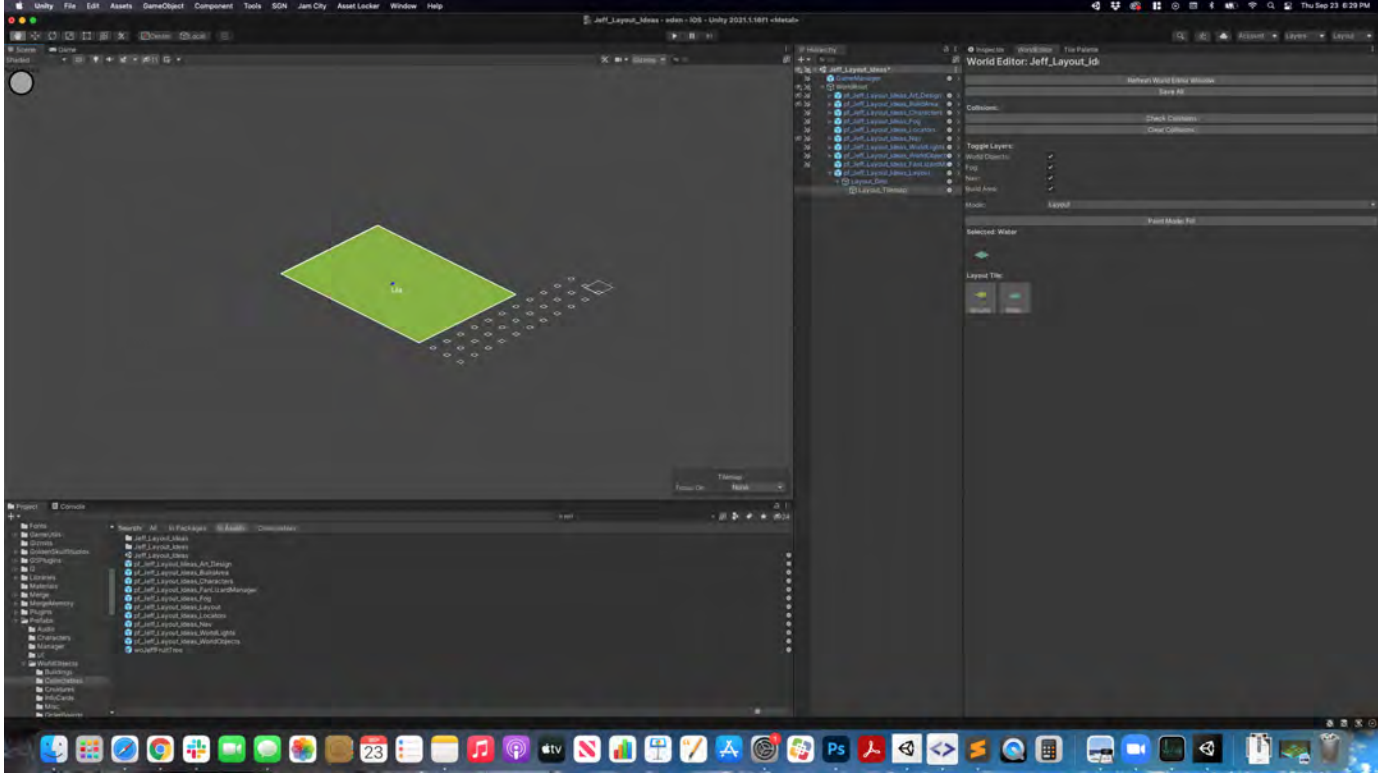
Inspector: A panel on the right side of the scene view, showing the properties of the selected tile. It includes sections for 'Collisions', 'Toggle Layers', 'Mode', and 'Paint Mode'. The 'Mode' is set to 'Layout' and 'Paint Mode' is set to 'Fill'. The 'Selected' tile is 'Ground'.

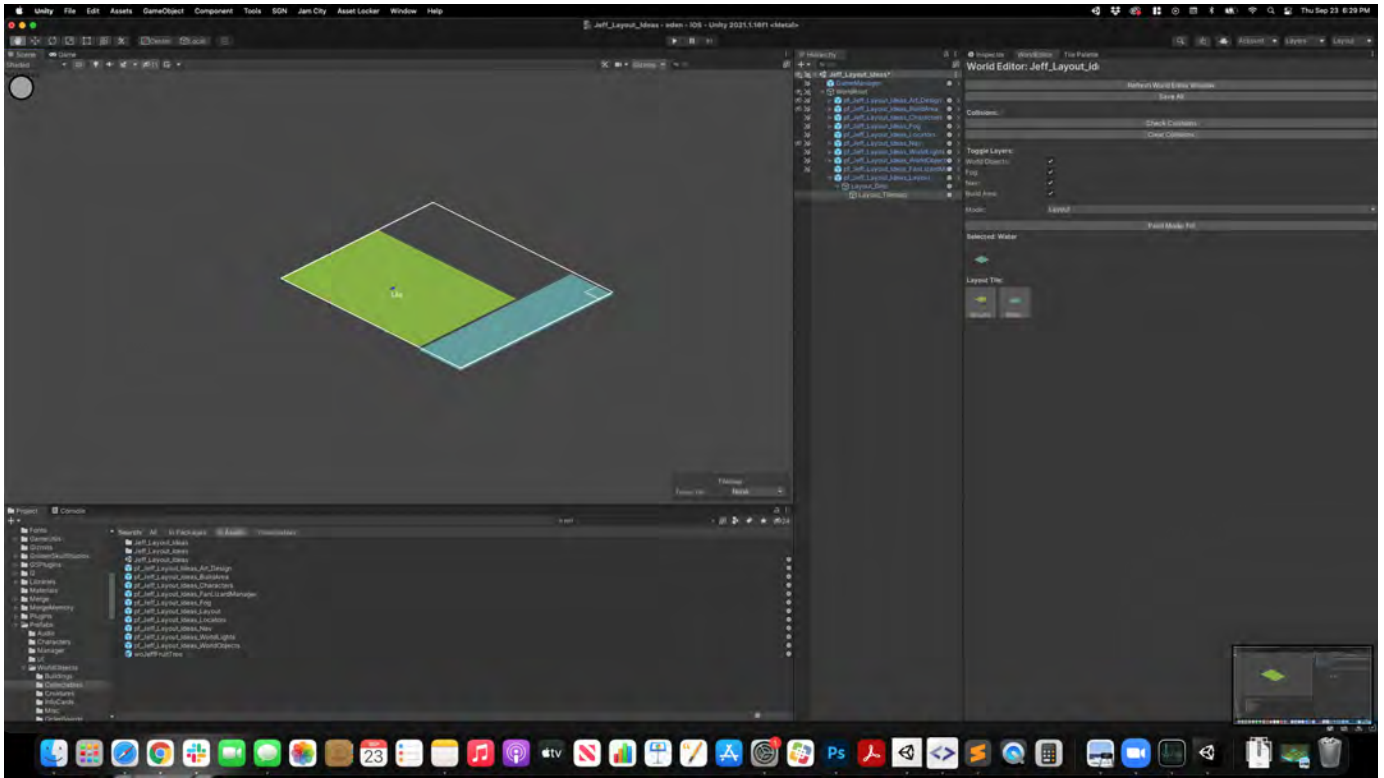
Project: A panel at the bottom left showing the project's file structure. It includes folders for 'Assets', 'Scripts', and 'Prefabs'. The selected tile is highlighted in blue.

MacOS Dock: A row of application icons at the bottom of the screen, including Finder, Safari, Chrome, and various utility apps.



Along with the ground, you can add in water tiles:

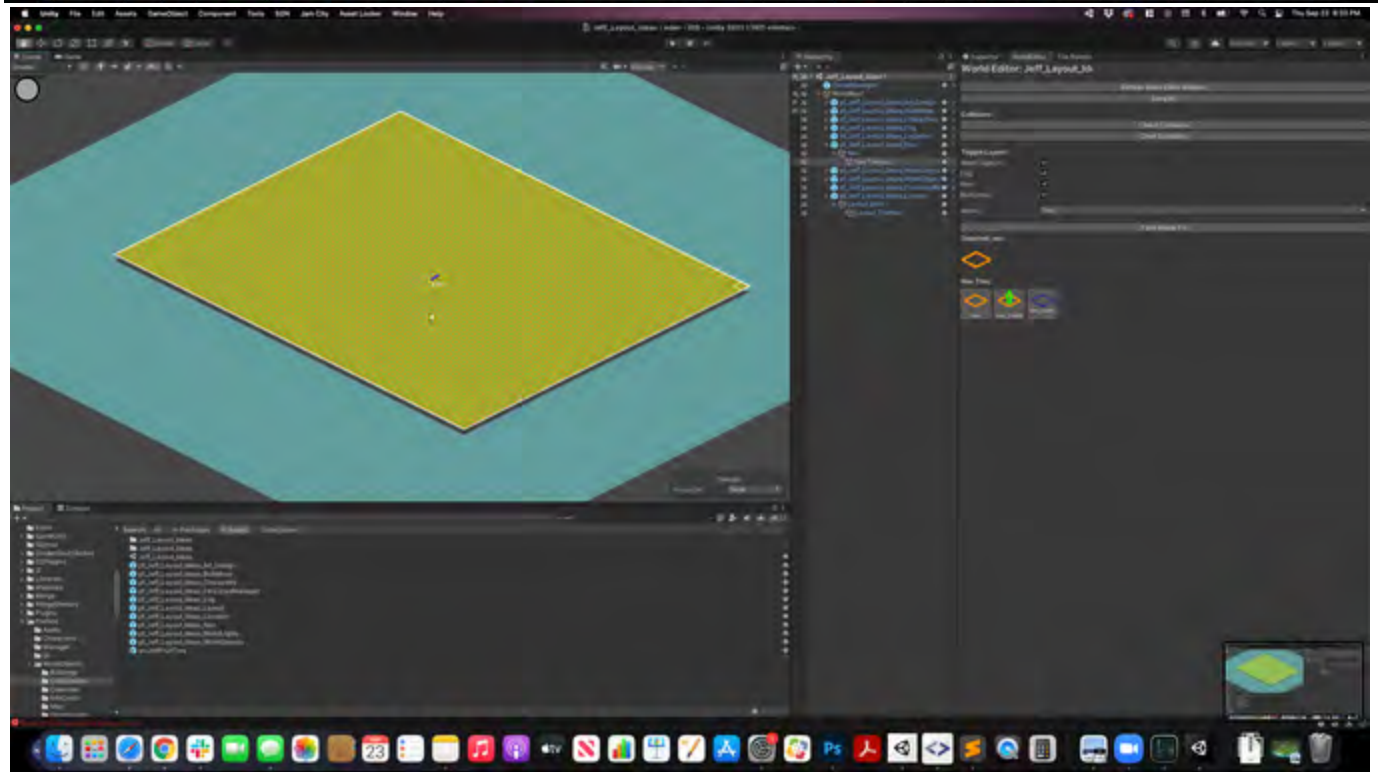
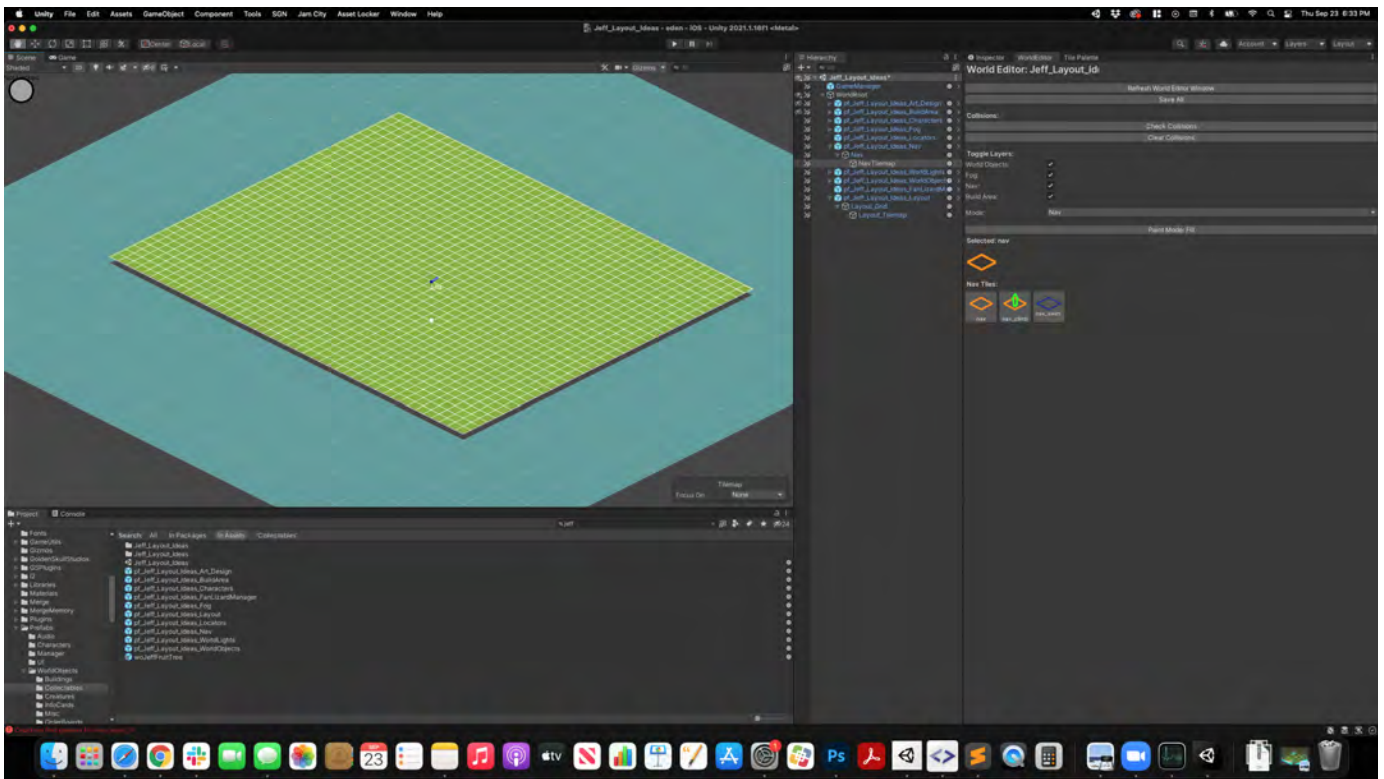




### Nav Tool



The Nav Tool is used to define areas where the player can walk. Simply draw over the ground like you would a ground plane.



## Danny's Confluence Nav Page

[Nav](#)

[Build Area](#)

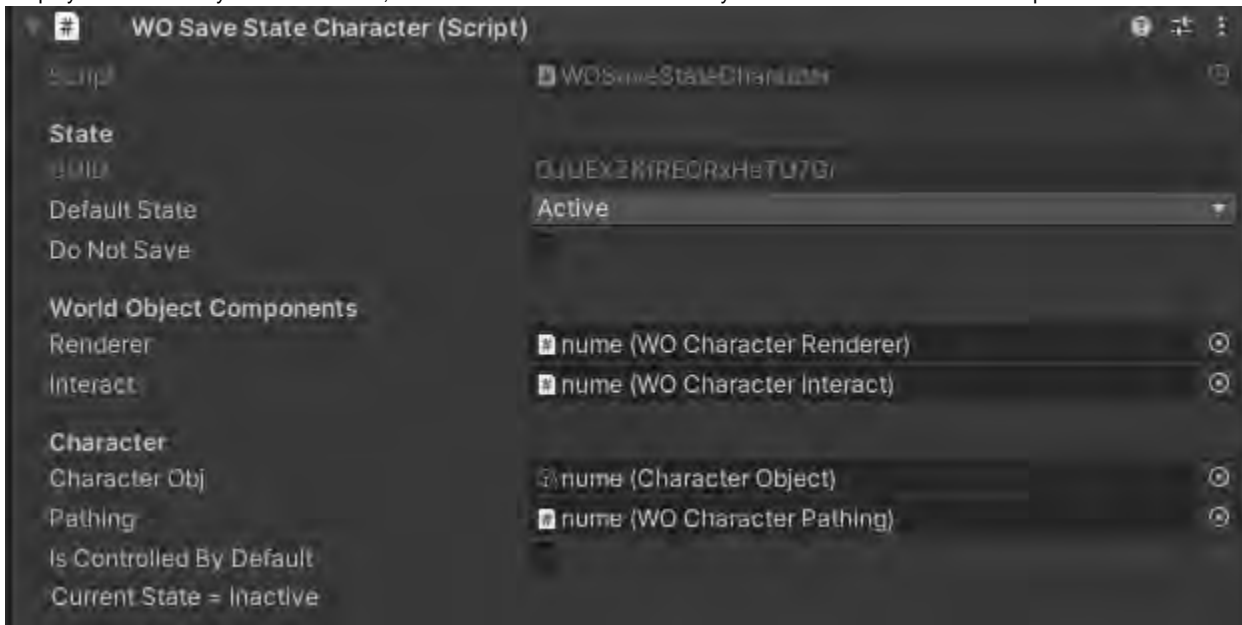
Danny does a great job of describing this feature in the following Confluence page.

[Build Area](#)



## Characters

To add characters to a level, select Place World Objects mode, and place them like you would other world objects. If they will not be controlled by the player from the very start of the level, make sure the box "Is Controlled by Default" is unchecked in the inspector.

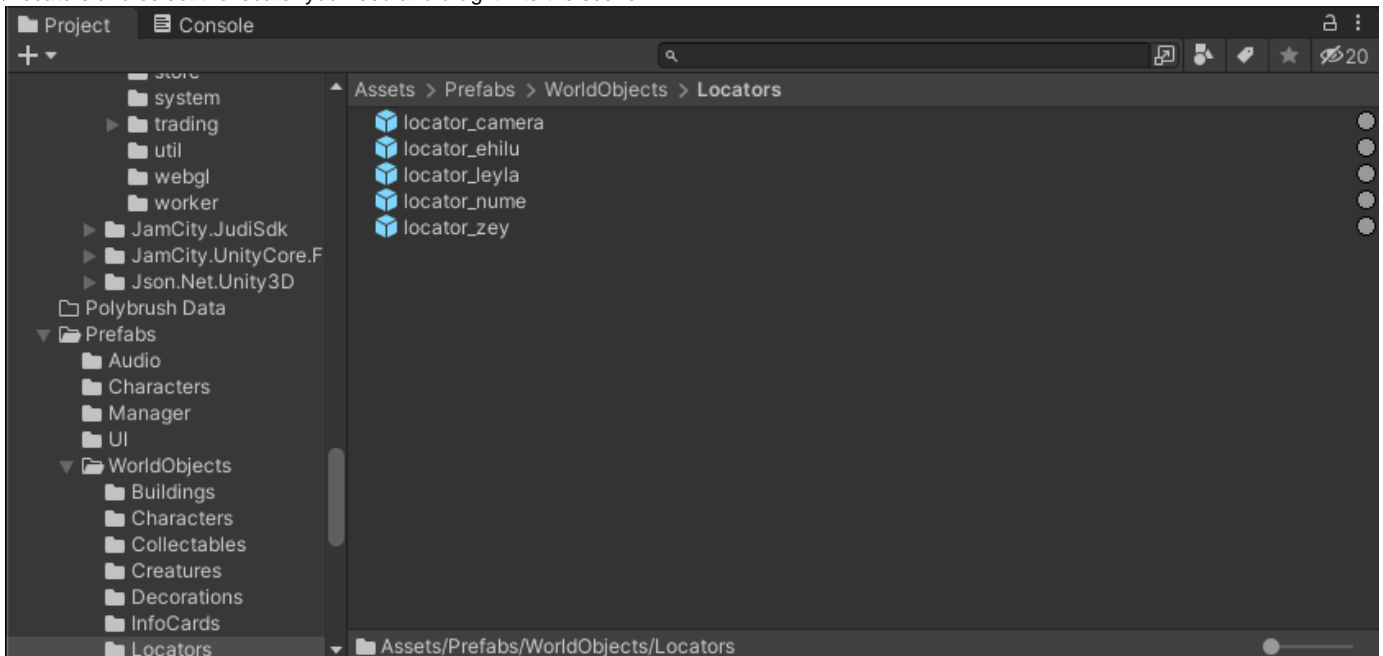


If you want them to accompany the player later on in the level, you can unlock them by using the `player_controls [main character, new character]` command in your script.

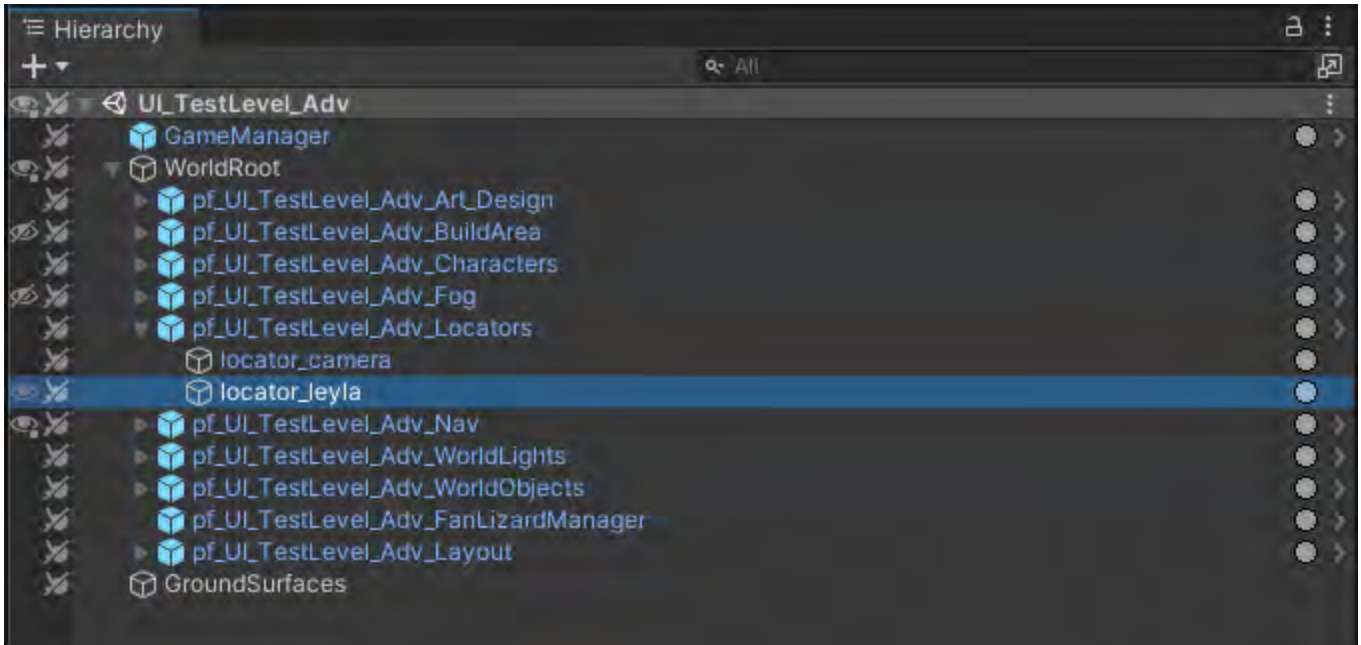
## Locators

\*Note: At the time of this writing, the Locator Mode in the World Editor was not working.

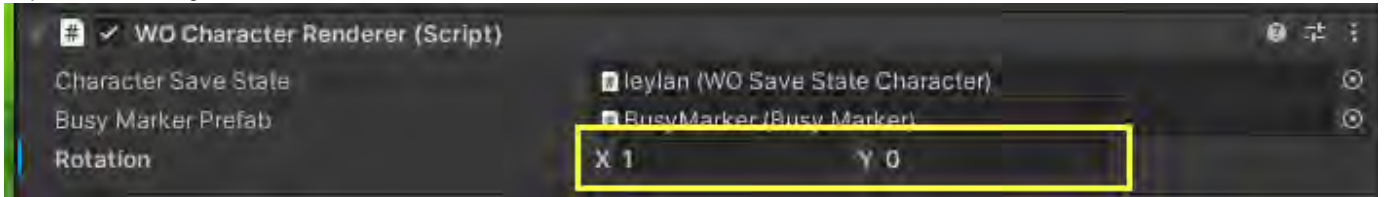
Locators are used to position characters and cameras during cutscenes. To get a locator into the scene, navigate to `Assets/Prefabs/WorldObjects/Locators` and select the locator you need and drag it into the scene.



Once the locator is positioned where you want in the scene, go to the Hierarchy and move the locator into the Locators prefab and rename it as needed.



To make the character face a certain direction, select the character in the hierarchy and look for the WO Character Renderer component in the Inspector tab. Change the Rotation value.



To see the change, look at the blue dial below the character.



The blue dial points towards the X,Y coordinates given. Think of a 2D grid that is on top of the dial. Below is a quick reference for commonly used directions.

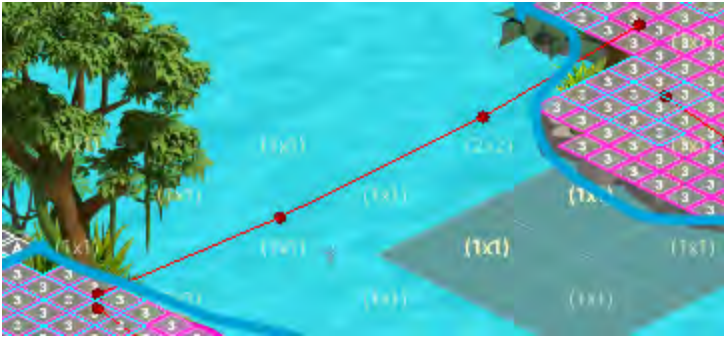
Direction	X	Y
Up	0	1
Up-Right	2	1
Right	1	0
Down-Right	2	-1
Down	0	-1
Down-Left	-2	-1
Left	-1	0
Up-Left	-2	1

### Pathing Locators

To create a path: right click the Locators prefab in the hierarchy > Create Empty > Rename to your path. Inspect your new path > Add Component > Scriptable Path > Add Path Node.

The path nodes are the children of your path and will appear as red dots. Use W+ Left click to move them around.



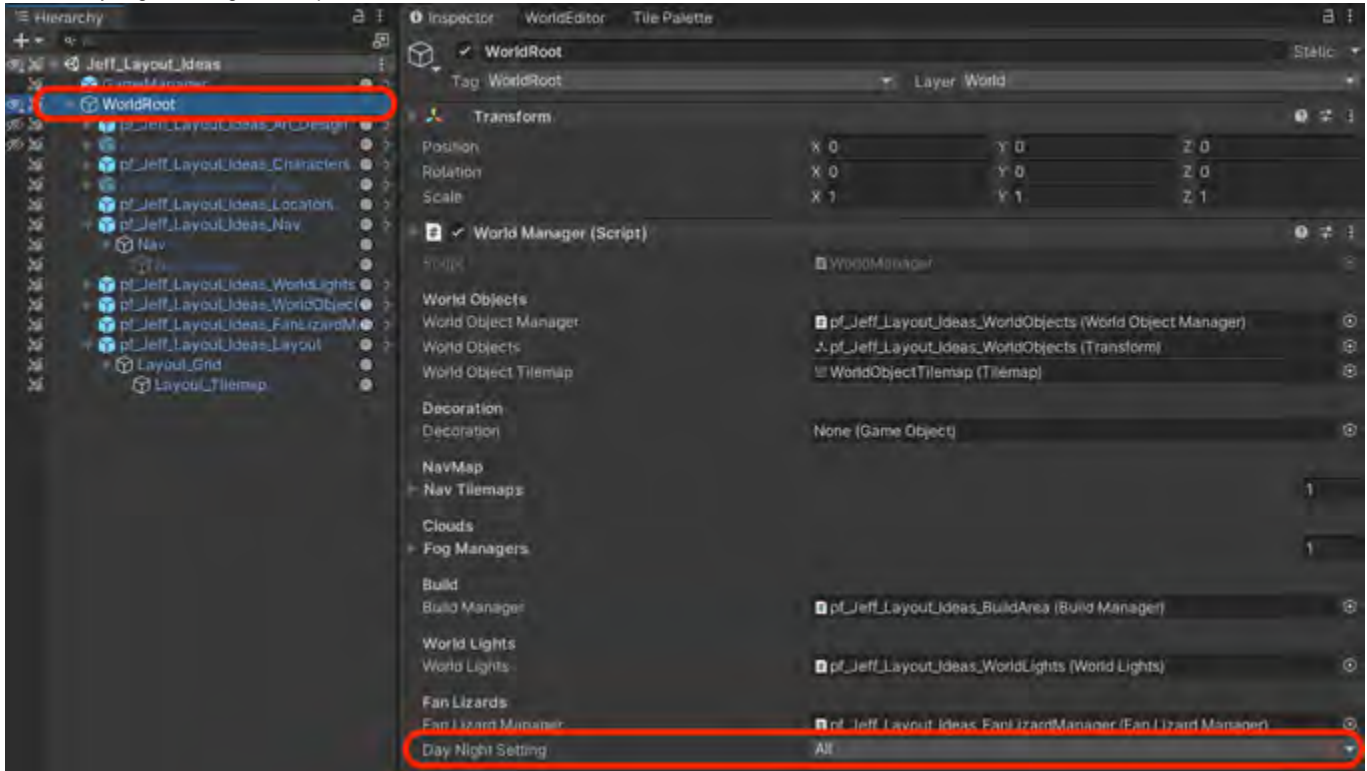


Flying paths will use the same method.

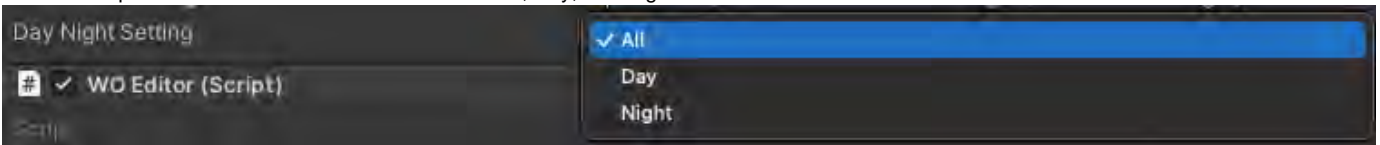
## Day-Night Setting

Eden will have day/night cycles in the Home Camp. However, adventure levels will be either light or dark (there is no notion of day or night on Pandora). Here is the setting for that for each level.

With the WorldRoot selected in the Hierarchy of your level, click on the Inspector tab. At the bottom of the World Manager (Script) section, you'll find the Day Night Setting and dropdown field.



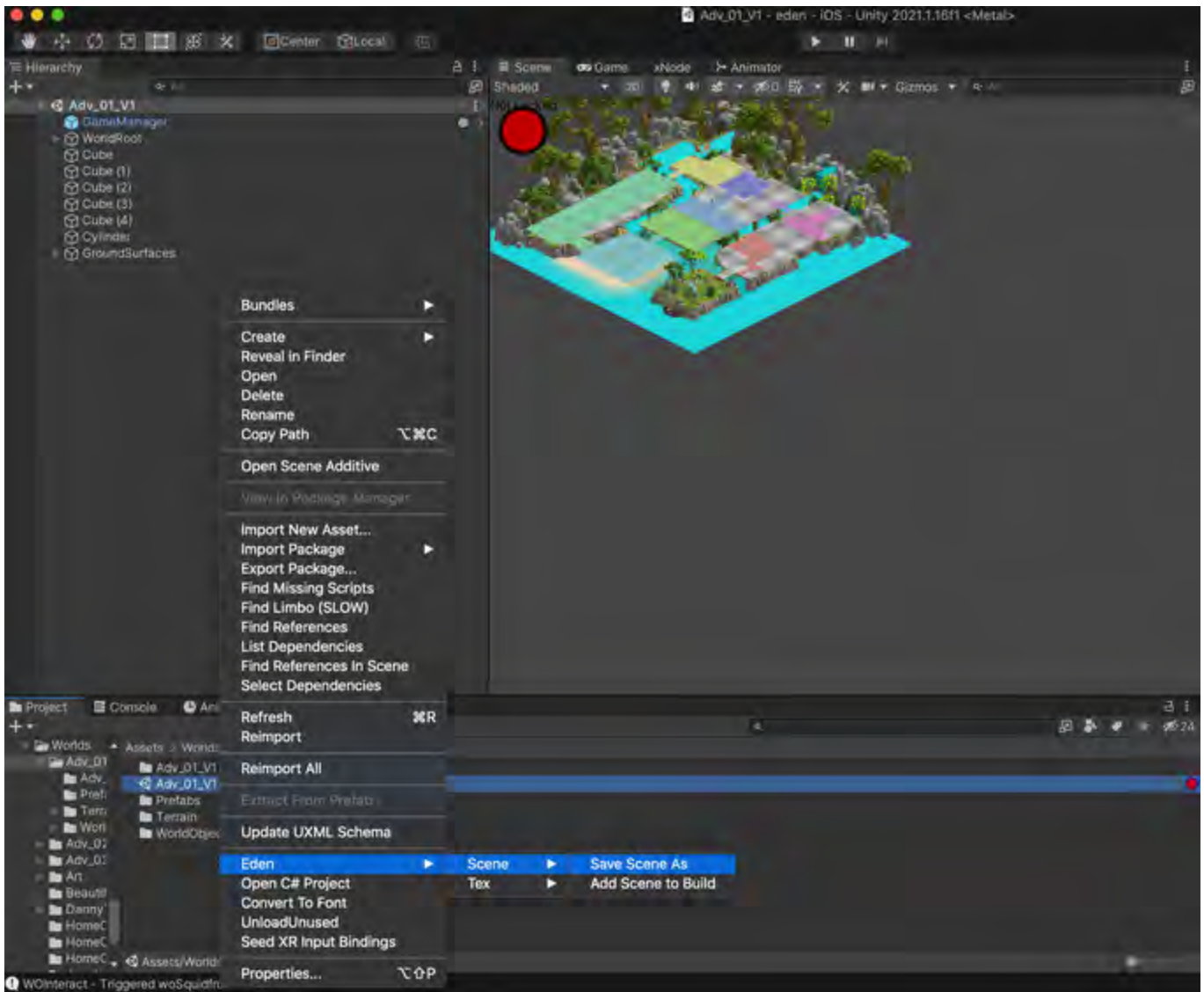
Click the dropdown arrow to reveal choices between All, Day, and Night.



- All - Day-Night cycles automatically. This is used for Home Camp only
- Day - Toggles on Day mode for the level. Will stay day the entire time in this level
- Night - Toggles on Night mode for the level. Will stay night the entire time in this level

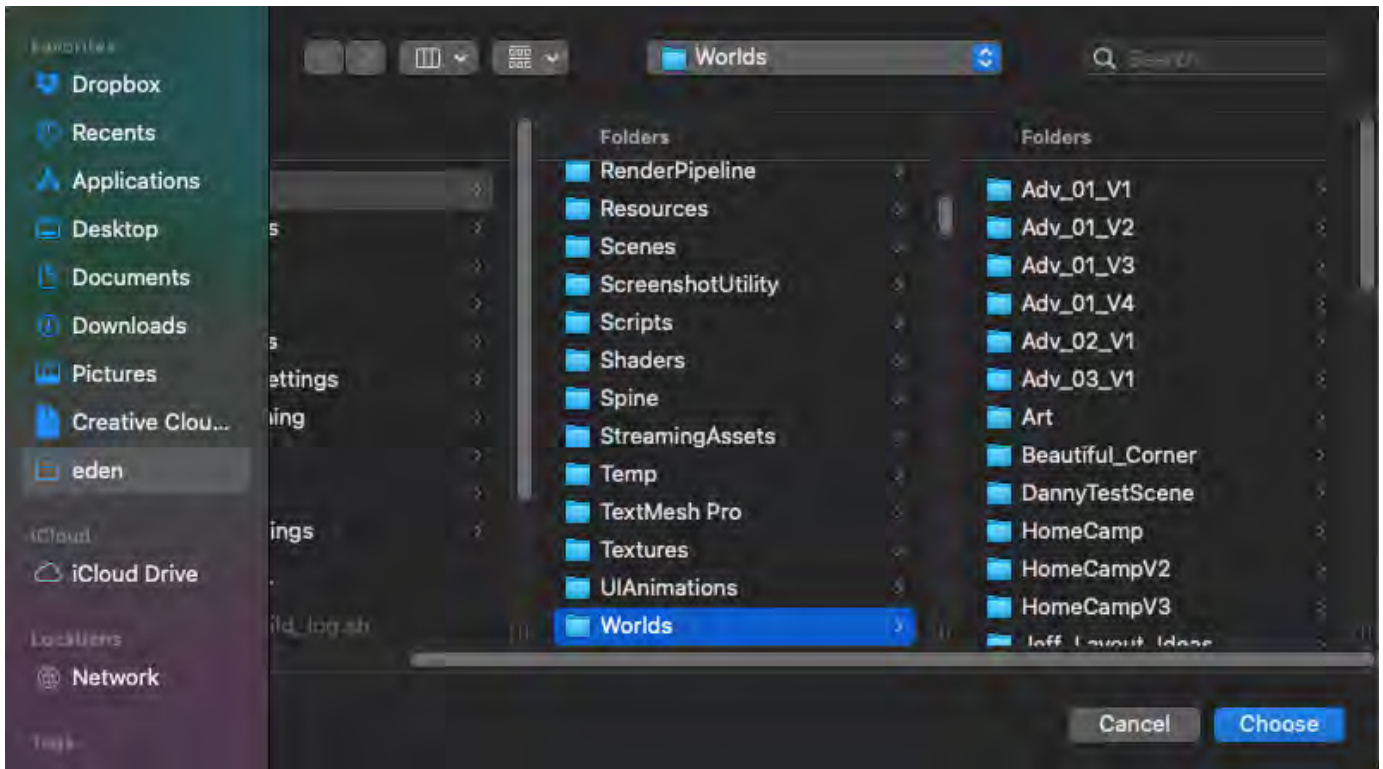
## Save As An Entire Level

This is functionality that allows you to essentially duplicate an entire level. This is great for creating a test condition of an existing level without harming the original.

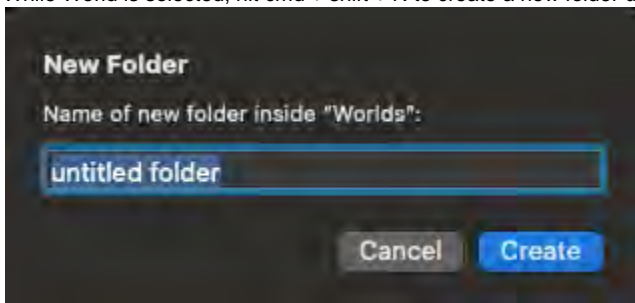


- In the Project tab, search for your level.
- Right click on the level's Unity file.
- Go to Eden > Scene > Save Scene As

A dialog box will appear asking you to choose a folder. World will be selected. This will bring up this dialog:

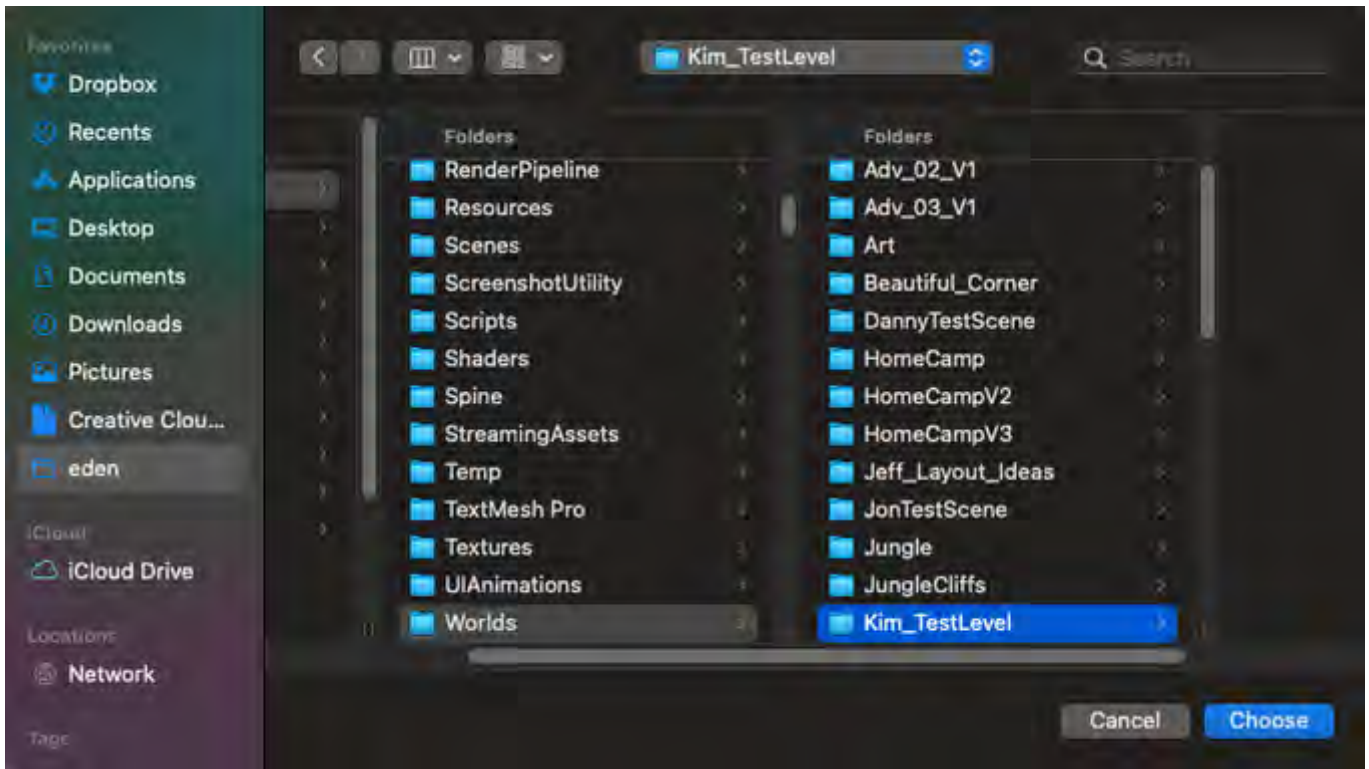


While World is selected, hit cmd + shift + N to create a new folder under world. This will pop up:



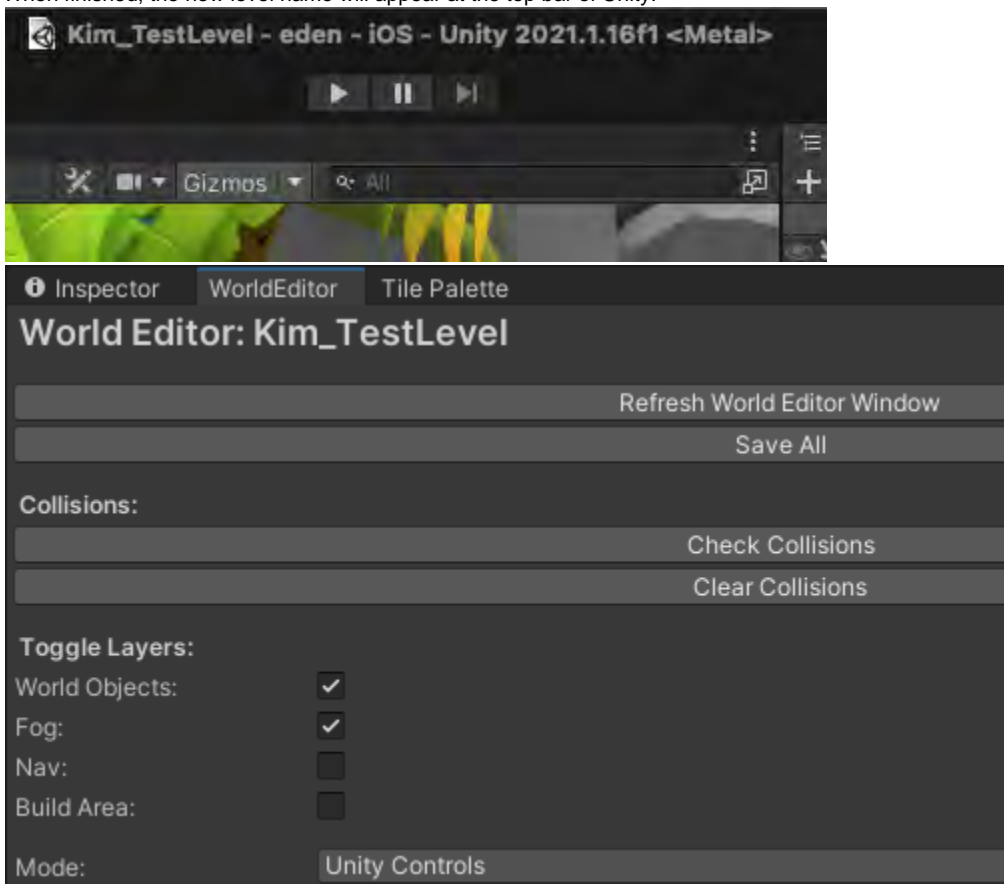
Name you new level name (which is the same as the folder name). Results should look like this:





Now hit Choose and everything that was in the previous level will be copied over to the new folder but with all of the names changed to the same folder name. This might take a minute or two to complete.

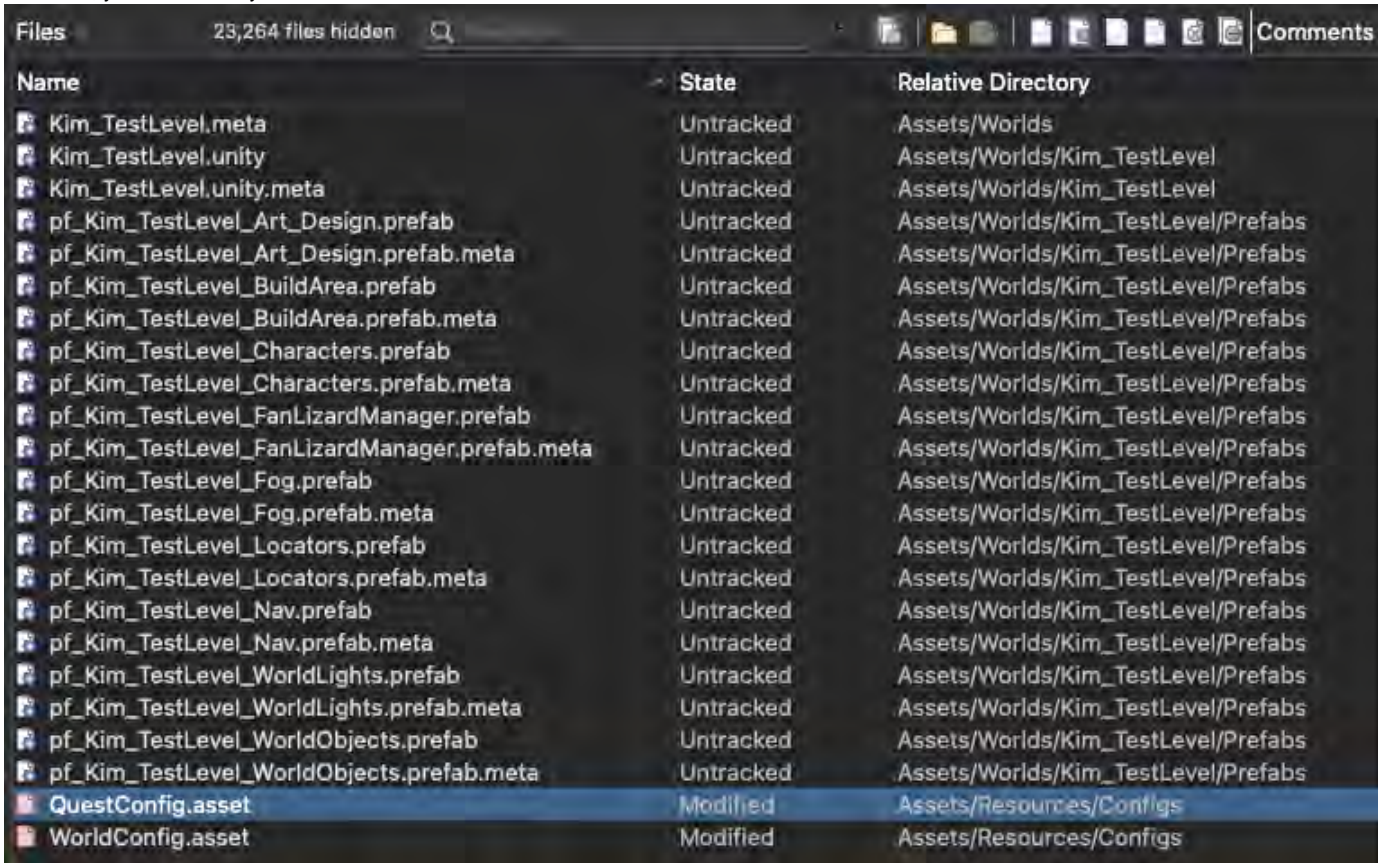
When finished, the new level name will appear at the top bar of Unity:





Next, you'll want to add this new level into the develop repository.

In GitHub, you'll see all of your new files associated to the new level.



You'll notice all of the new files are "Untracked." This means they are not registered/recognized by develop. You have to commit and push these new files onto develop so they can be tracked. So let's go ahead and Stage all of the untracked files. We'll take care of pushing the WorldConfig.asset file in a different commit.

Name	State	Relative Directory	Renamed Path
QuestConfig.asset	Modified	Assets/Resources/Configs	
WorldConfig.asset	Modified	Assets/Resources/Configs	
Kim_TestLevel.meta	Added	Assets/Worlds	
Kim_TestLevel.unity	Added	Assets/Worlds/Kim_TestLevel	
Kim_TestLevel.unity.meta	Added	Assets/Worlds/Kim_TestLevel	
pf_Kim_TestLevel_Art_Design.prefab	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel...sign.prefab.meta	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel_BuildArea.prefab	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel...Area.prefab.meta	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel_Characters.prefab	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel...ters.prefab.meta	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel...rdManager.prefab	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel...ager.prefab.meta	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel_Fog.prefab	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel_Fog.prefab.meta	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel_Locators.prefab	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel...tors.prefab.meta	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel_Nav.prefab	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel_Nav.prefab.meta	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel_WorldLights.prefab	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel...ghts.prefab.meta	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel...ldObjects.prefab	Added	Assets/Worlds/...tLevel/Prefabs	
pf_Kim_TestLevel...ects.prefab.meta	Added	Assets/Worlds/...tLevel/Prefabs	

And now, above, we've successfully added all the new files. Next, we need to Commit these files with a Commit note.



Name	State	Relative Directory
QuestConfig.asset	Modified	Assets/Resources/Configs
WorldConfig.asset	Modified	Assets/Resources/Configs

Name	State	Relative Directory	Renamed Path
Kim_TestLevel.meta	Added	Assets/Worlds	
Kim_TestLevel.unity	Added	Assets/Worlds/Kim_Test/	
Kim_TestLevel.unity.meta	Added	Assets/Worlds/Kim_Test/	
pf_Kim_TestLevel_Art_Design.prefab	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel...sign.prefab.meta	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel_BuildArea.prefab	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel...Area.prefab.meta	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel_Characters.prefab	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel...ters.prefab.meta	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel...rdManager.prefab	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel...ager.prefab.meta	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel_Fog.prefab	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel_Fog.prefab.meta	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel_Locators.prefab	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel...tors.prefab.meta	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel_Nav.prefab	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel_Nav.prefab.meta	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel_WorldLights.prefab	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel...ghts.prefab.meta	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel...ldObjects.prefab	Added	Assets/Worlds/...tLevel/P	
pf_Kim_TestLevel...ects.prefab.meta	Added	Assets/Worlds/...tLevel/P	

- Open from Working Tree
- Show Changes F4
- Log ⌘ L
- Blame ⌘⇧ L
- Commit... ⌘ K**
- Stash Selection...
- Stage ⌘ T
- Unstage ⇧⌘ T
- Index Editor ⌘⇧ T
- Rename... F2
- Conflict Solver
- Resolve... ⇧⌘ V
- Ignore... ⌘ I
- Discard... ⌘ Z
- Remove... ⌘ -
- Delete... ⌘ X
- Copy Name
- Copy Path ⇧⌘ C
- Copy Relative Path ⇧⌘⇧ C

### Commit local or staged changes

Select the files you want to commit and provide a commit message.



Assets/Worlds/Kim\_TestLevel

21 files (+21)

Name	Directory
Kim_TestLevel.unity	Assets/Worlds/Kim_TestLevel
Kim_TestLevel.unity.meta	Assets/Worlds/Kim_TestLevel
pf_Kim_TestLevel_Art_Design.prefab	Assets/Worlds/Kim_TestLevel/Prefabs
pf_Kim_TestLevel_Art_Design.prefab.meta	Assets/Worlds/Kim_TestLevel/Prefabs
pf_Kim_TestLevel_BuildArea.prefab	Assets/Worlds/Kim_TestLevel/Prefabs
pf_Kim_TestLevel_BuildArea.prefab.meta	Assets/Worlds/Kim_TestLevel/Prefabs
pf_Kim_TestLevel_Characters.prefab	Assets/Worlds/Kim_TestLevel/Prefabs

Commit Message:

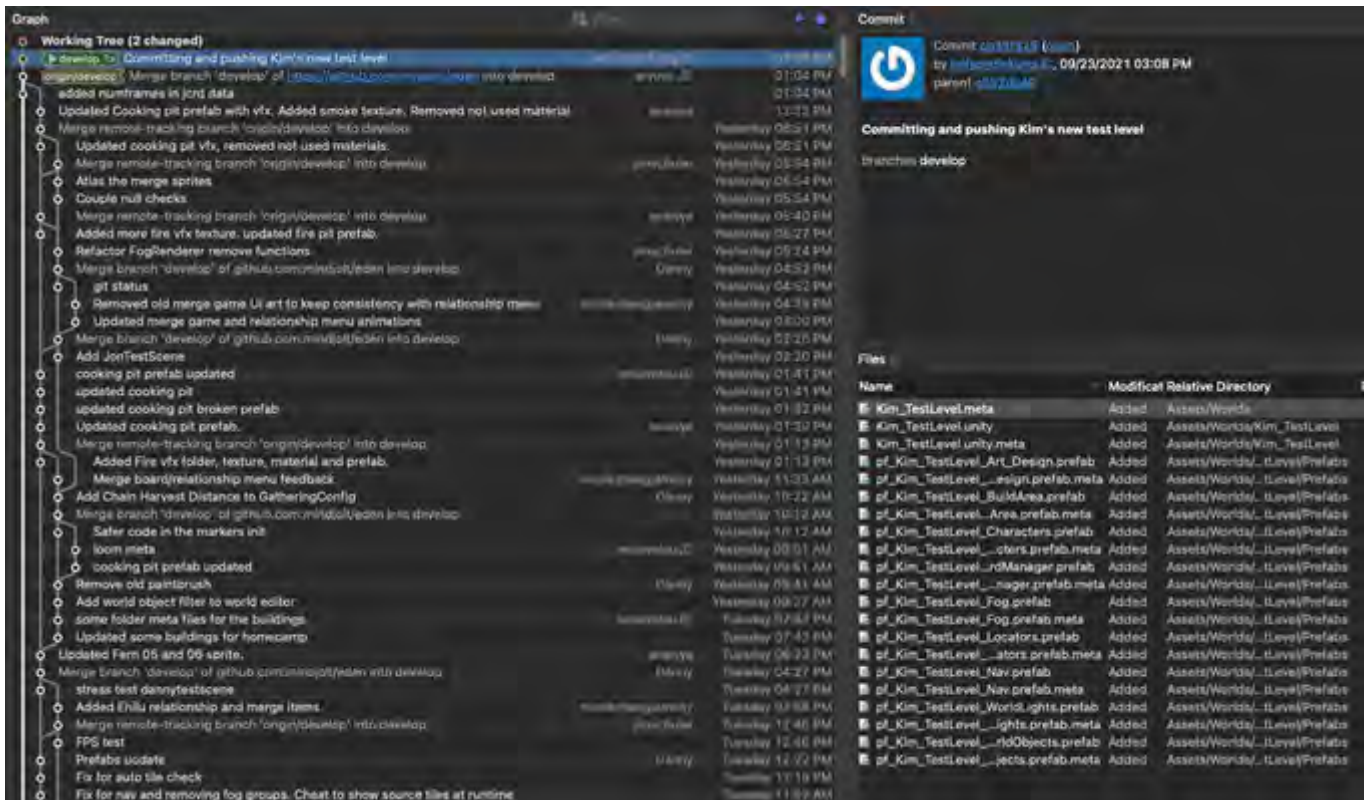
Select

Committing and pushing Kim's new test level

Assets/Worlds/Kim\_TestLevel

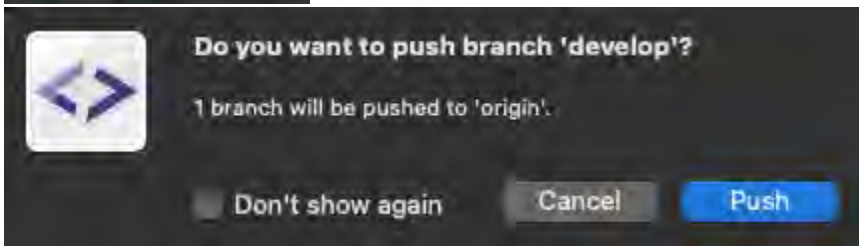
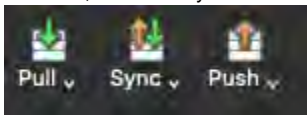
More Optio...

Cancel Commit Commit & Push



Committed!^

And now, we're ready to Push this level to develop.

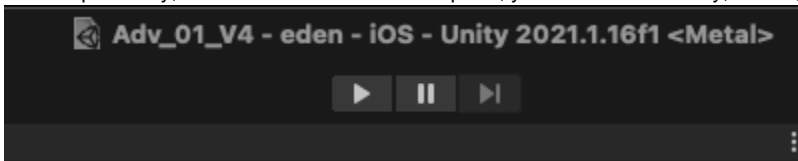


Commit and Push the WorldConfig.asset file separately. And that's it.

NOTE: When creating a new level, it is best to check in WorldConfig.asset right away so that it isn't written over or conflicts with someone else's check in.

## Play Your Level

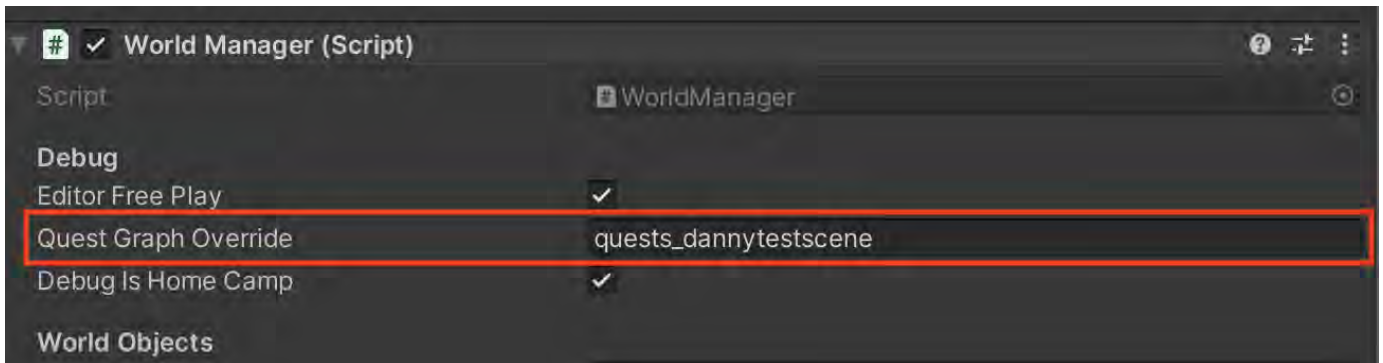
At the top of Unity, under the level title on the top bar, you'll see a set of Play, Pause, and Frame Forward buttons.



Press the Play button and the game will start. But if you want a specific level to start (the level you're working on), you'll have to change the order of the main quests in the QuestSystemManager.prefab file.

NOTE: There is a new/better way to do this. Danny created an override that will allow you to specify exactly what Quest Graph you want to start.

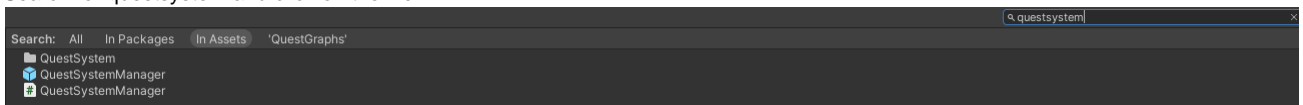




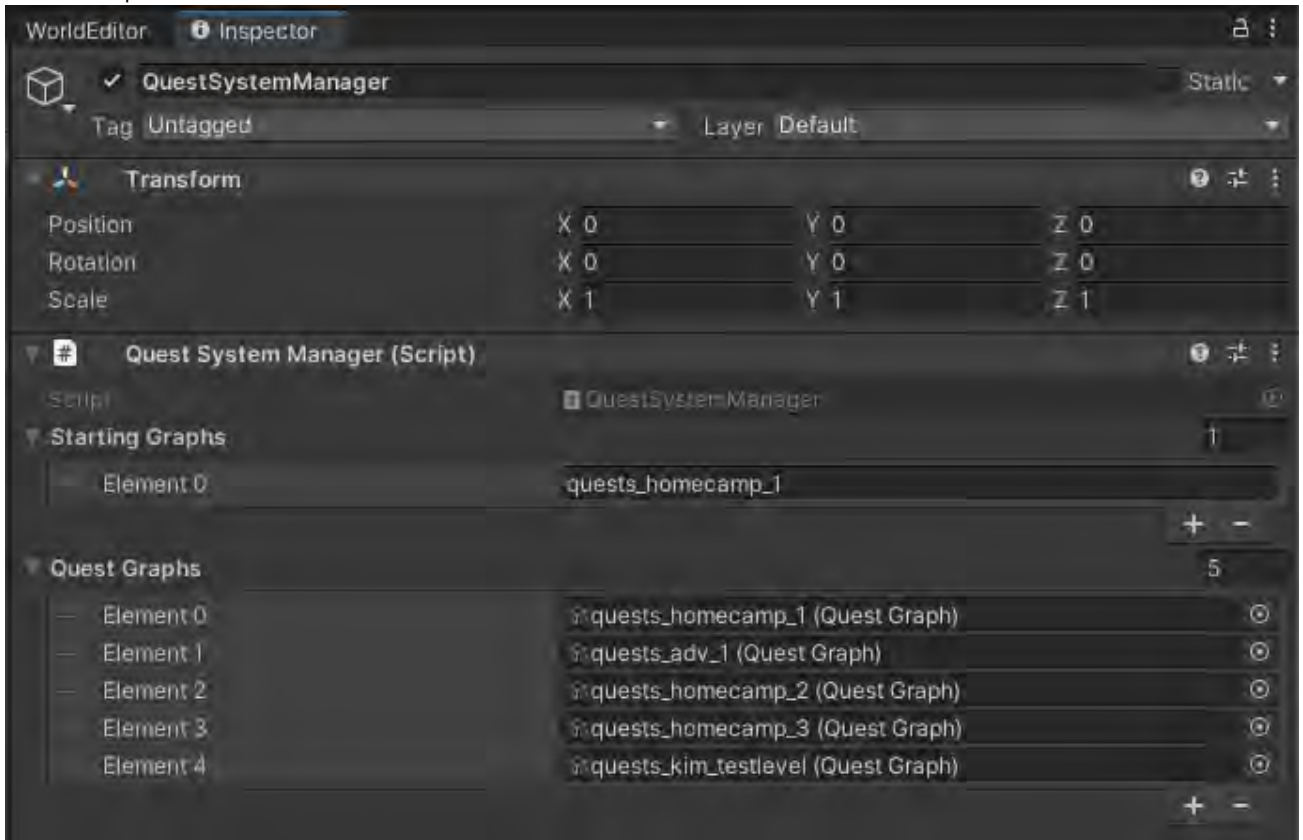
The rest is still valid but Danny's new method is a lot easier and a lot safer (no more accidentally committing the QuestSystemManager prefab).

Here's how you do that:

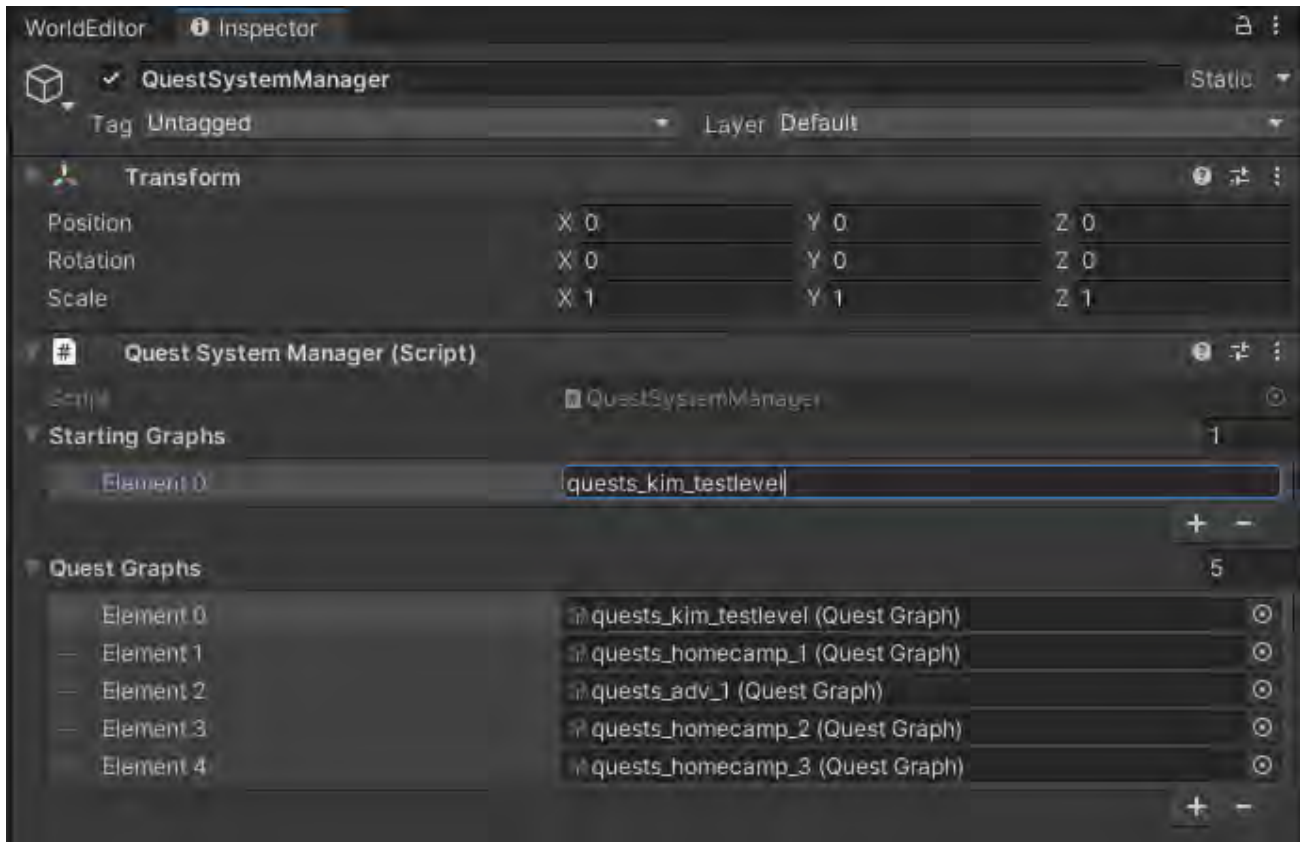
- Go to your Project tab
- Search for questsystem and click on the file



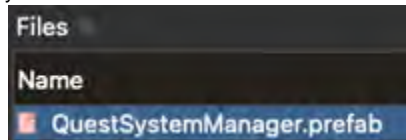
- Go to the Inspector tab and it'll look like this:



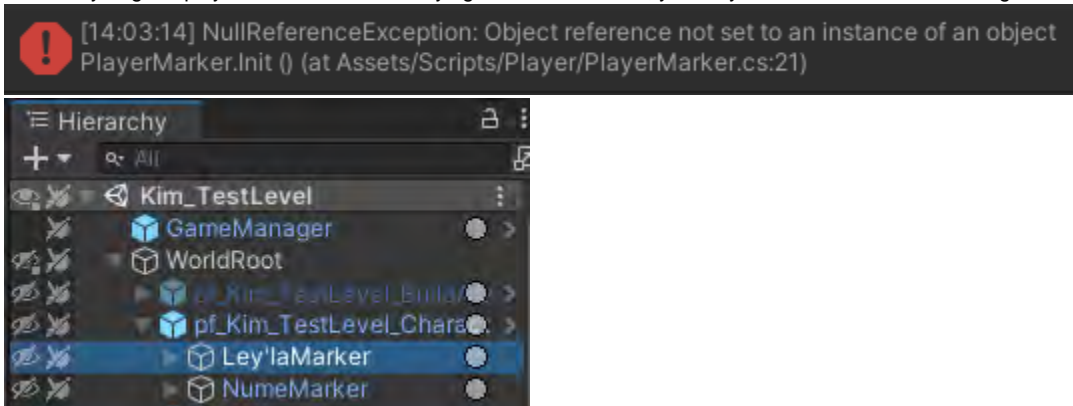
- Grab the = sign of the quests you want to put on top, on the far left side under Quest Graphs, and drag it up to the top. Replace the Starting Graphs with the one you want. After, hit Save All. Now it'll play first when you hit the play button.

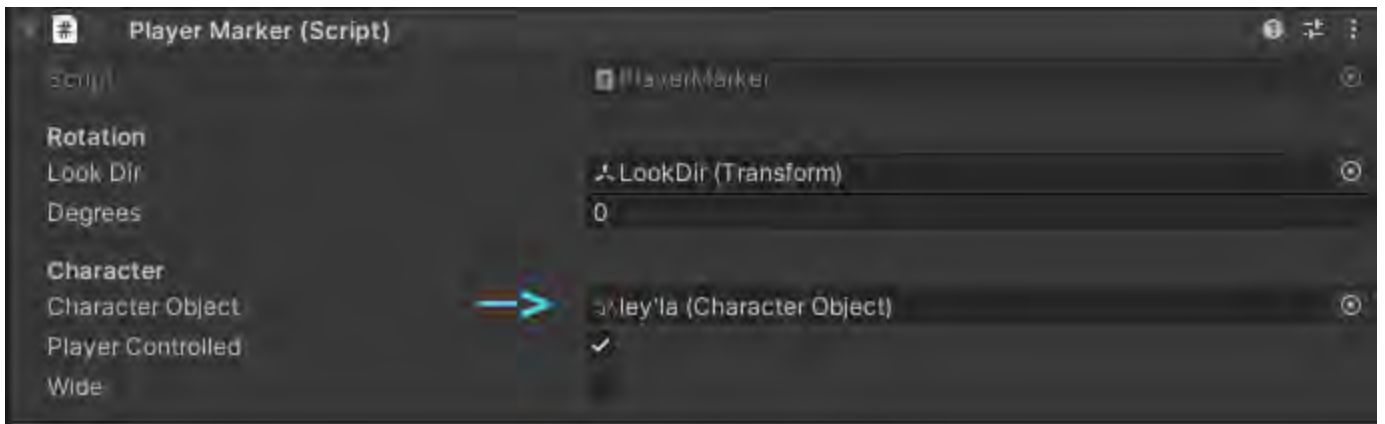


In GitHub, you'll get a check out of a file called `QuestSystemManager.prefab`. This tweak you did above is from this file. And for as long as you need to test your level, make this `QuestSystemManager.prefab` perma-checked out. DO NOT commit the `QuestSystemManager.prefab` file unless you would like the changes to show for everyone.



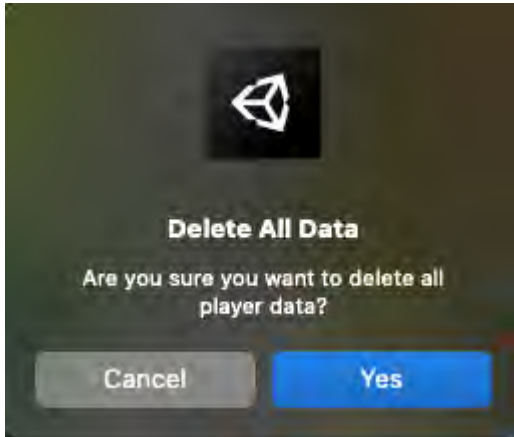
NOTE: If you get a playermarker error when trying to run a new level, you may need to set the character again in the inspector.



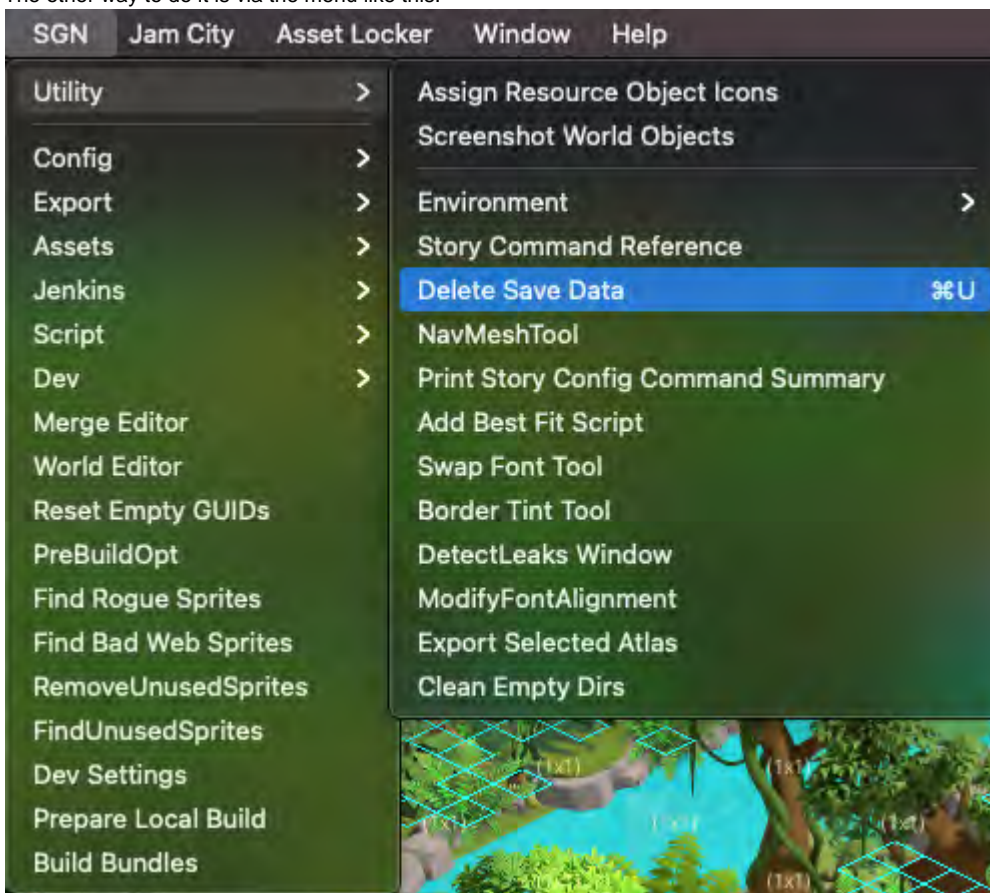




TIP: To reset your fog back to the beginning as well as all of the quests, so that you can replay your level, press + U. A dialog box will appear, click on the Yes button.



The other way to do it is via the menu like this:



## QuestGraph

### Introduction

The QuestGraph is the guts of a level. This is our scripting tool and this is what keeps our players constantly moving forward in our game.

A level is a series of quests with tasks that, when completed, trigger a story script that may contain as little as a command to start the next quest to extensive camera zooms, camera vignetting, delays, and character chat lines. We can even move characters on the play field.

### Creating New QuestGraphs

The video in this Confluence page shows how to create a new QuestGraph:

[New QuestGraph](#)

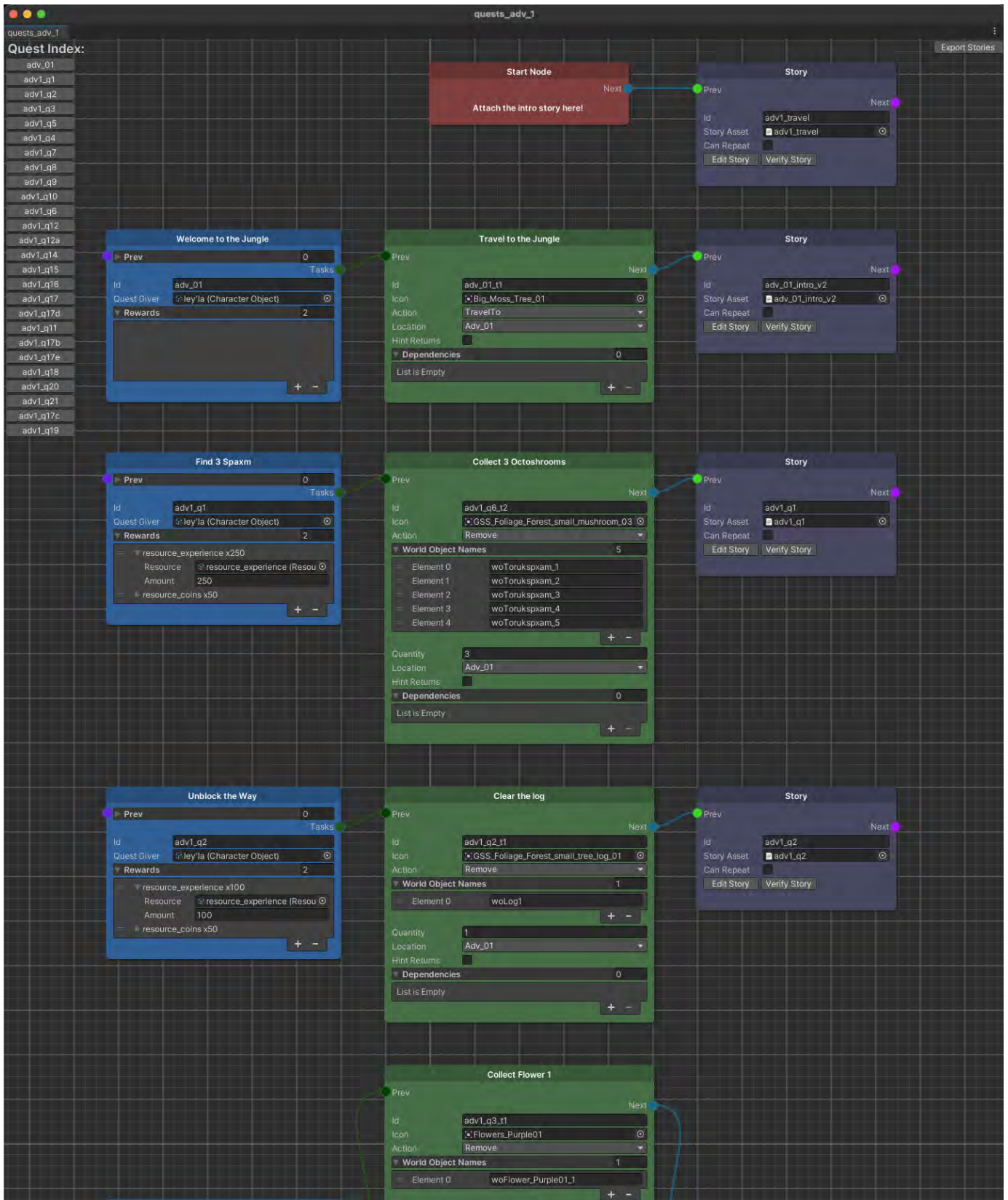
Essentially the steps are as follows:

- From the Project tab, go to Assets>aaStoryGraphs>QuestGraphs and right click in any space and choose Create > Quest Graph
- This will create a new Quest Graph in Assets > aaStoryGraphs > QuestGraphs called New\_Quest\_Graph
- Rename this to quests\_ \*whatever\* **NOTE:** Keep quest IDs lowercase, this is how they are imported from the Loc file.
- From the Project tab's search, type in QuestSystemManager and select the QuestSystemManager (blue cube/red bracket icon)
- In the Inspector tab, go down to the Quest Graphs section and hit Plus icon to add an element.
- Click the radio dial button. This will bring up the Select Quest Graph window. Choose the new QuestGraph
- Now move the QuestGraph element to the top of the order. Now when you test your level, it will load your level first. Reminder, don't submit the checked out QuestSystemManager.prefab Just keep it locally.

## QuestGraph Nodes

Here are the main parts of the QuestGraph.

**Main UI:**



Quest Index:



quests\_adv\_01\_v2

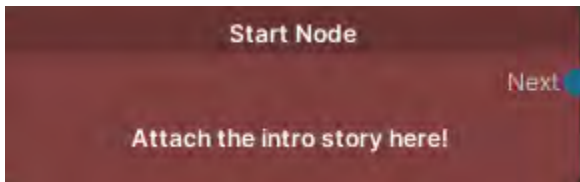
**Quest Index:**

adv1_q1
adv1_q2
adv1_q3
adv1_q4
adv1_q4b
adv1_q5
adv1_q6
adv1_q7
adv1_q8
adv1_q9
adv1_q10
adv1_q11
adv1_q12
adv1_q13
adv1_q13b
adv1_q14
adv1_q15
adv1_q16
adv1_q17
adv1_q17c
adv1_q17b
adv1_q18
adv1_q19
adv1_q20
adv1_q21

#### Nodes

- This is essentially a bookmark table of contents. Each button will take you too that particular quest

#### Start Node



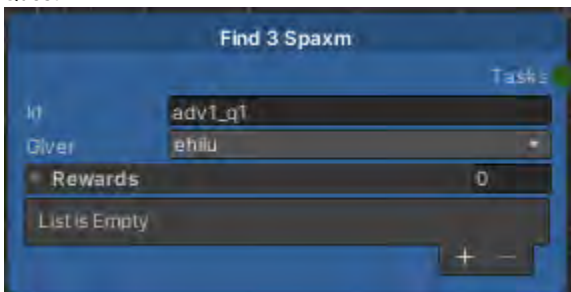
- There's only one function, to set the order of the level and push to the first story

#### Story



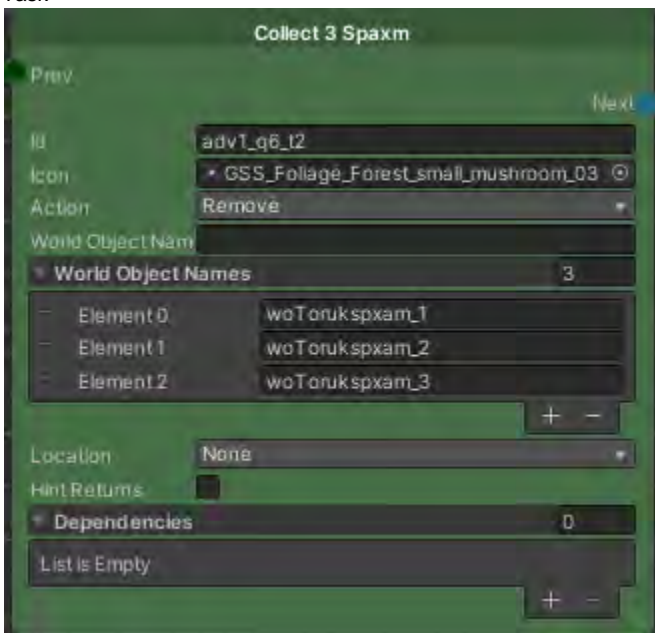
- This handles the script for everything from starting the next quest to triggering camera cutscenes, starting character dialog, etc.

#### Quest



- The quest handles the start of a quest with the quest ID and any rewards for completing the quest via its tasks.

#### Task

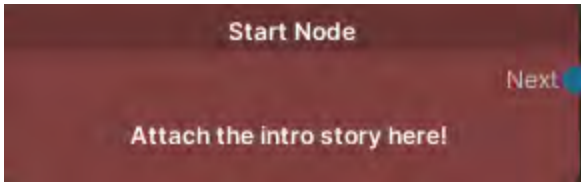


- The task window is the meat of the scripting system. This is where you'll connect World Objects with the script.

Let's get into more details about each of these windows and their functions. Then we'll go through a use case as an example.

#### Start Node

Chapter 1 will be the Home Camp, where the player and the story will start. Chapter 2 will be the first adventure level and so on. Hence, the Chapter window in the above example is set to 2, which is the first adventure level the player will visit partway through, early on, in the Home Camp.

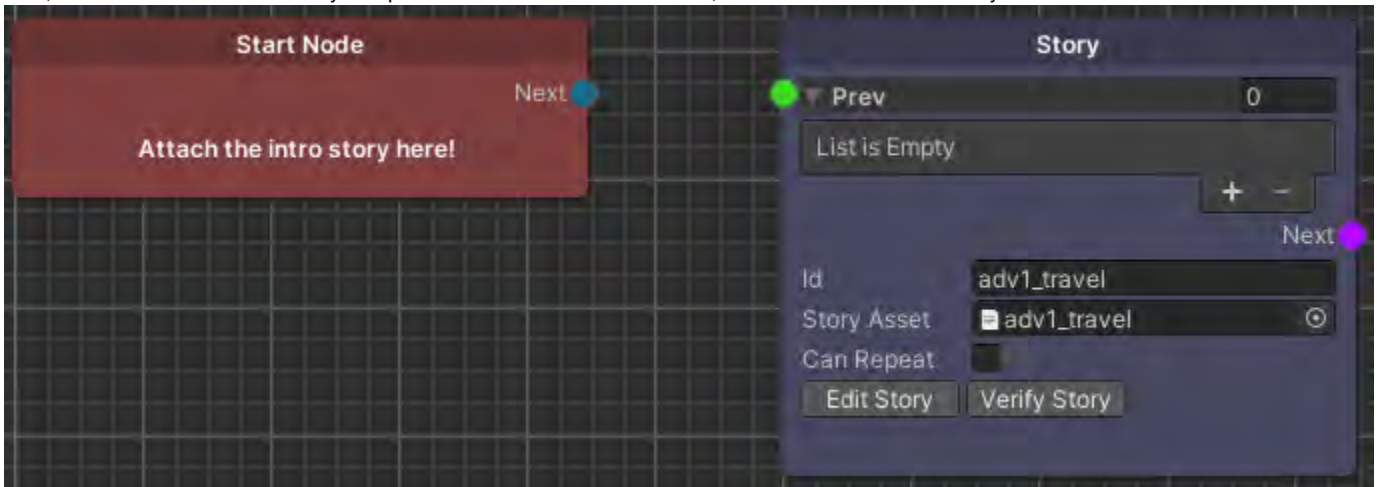


Assuming you're starting from scratch. Start by right clicking the QuestGraph field. A menu will appear:

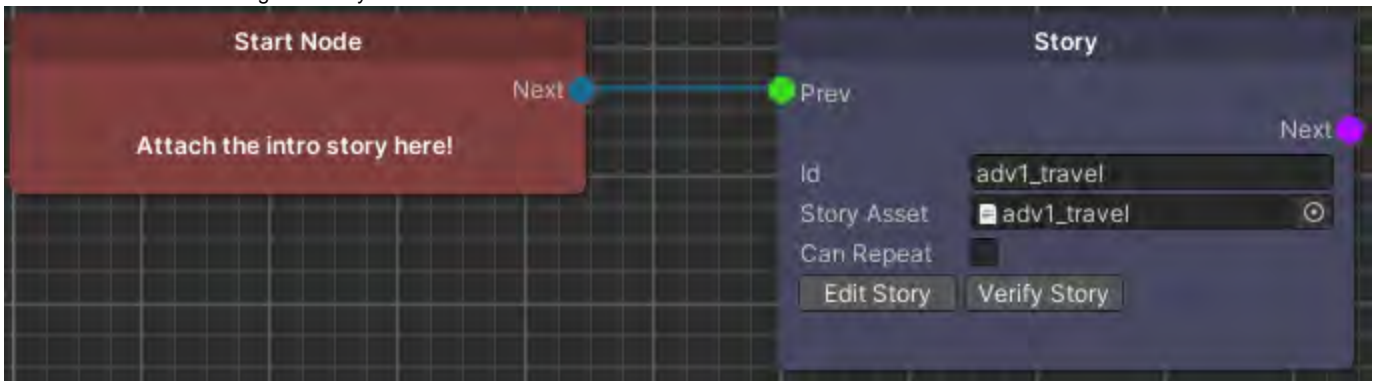


Choose Start to start the level.

Next, do the same but choose Story and put the two nodes near each other, Start node on the left of Story. It should look like this:



Now click and hold on the Next node and drag across towards the Story node. A tether is attached to the Next node and the cursor will draw the rest of the line towards the green Story node. This is how it'll look when connected.





## Story

Now, click on the Edit Story button. This will bring up the \*.storyconfig file in your text editor\*\*. If a \*.storyconfig file doesn't already exist, a default one will be created.



Note: The Edit Story Object button is a way to do the same kind of scripting as Edit Story does but it does it in Unity itself instead of in text. I prefer the text method so that's what will be focused on here.

If a previous Task or Chapter output is connected to the green input point, the upper list field disappears as seen below:



The ID, taken from [I2Loc Eden Localization](#), will automatically fill in the Story Asset with an existing asset or create an empty one with the correct name.

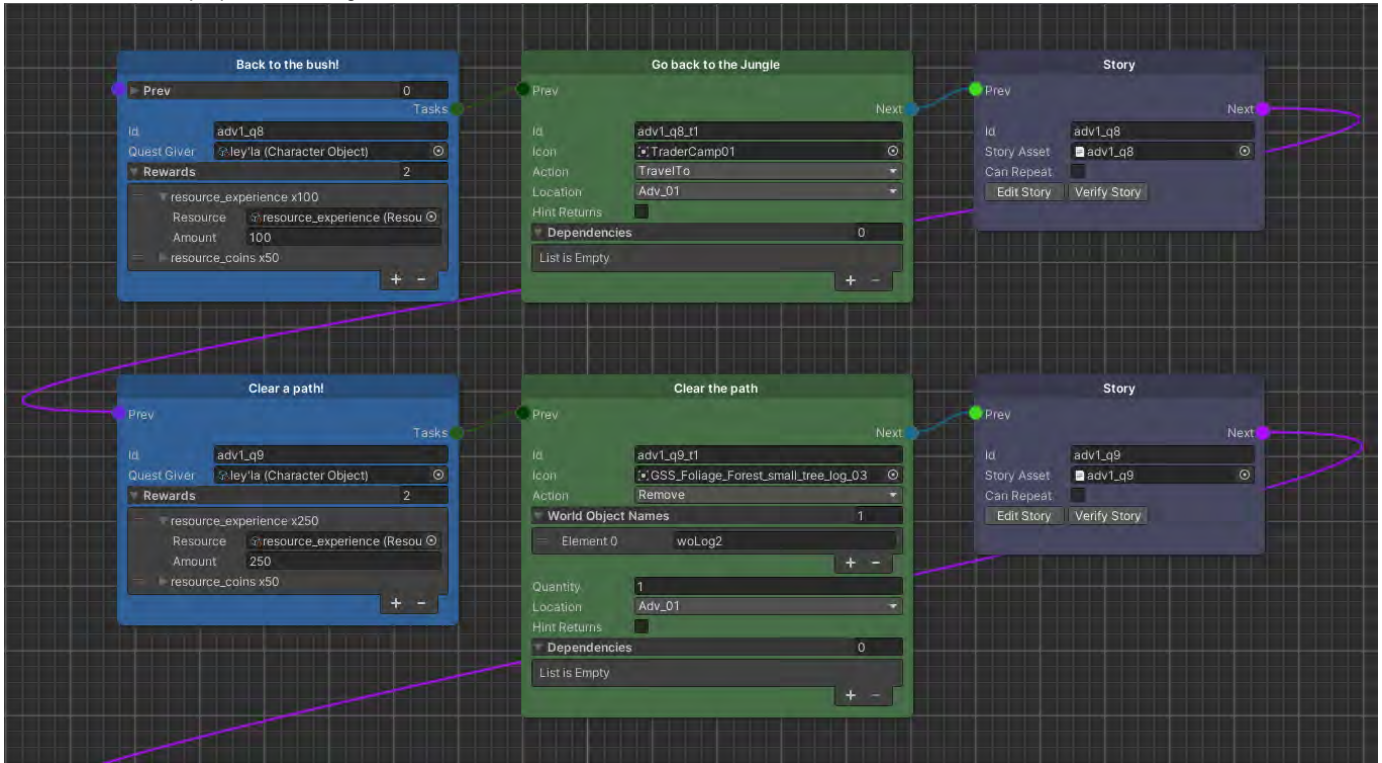
[PLACEHOLDER]

Naming convention hasn't been established yet. Will edit this when a new naming convention is made.

**TIP: You can copy/paste any node or select any number of nodes to copy/paste! Super handy when creating multiple but similar node scripts.**

## Connecting Story Nodes to Next Quest

To start the next quest upon completion of a Story, simply connect the Next output from the first story to the Prev input of the next quest as shown below in the purple connecting lines:



\*\*Use a text editor of your choice. I like Sublime Text. Download here for free: <https://www.sublimetext.com/download>

Here is an example of a StoryConfig:

```
adv_01_intro_v2.storyconfig
1  # ADV_01_INTRO_V2
2  text cliffs_area_title cliffs_area_subtitle
3  activate_quest adv1_q2
4  delay 3.5
5  move_camera Cam_Start1 2 .21
6  spotlight true Spot_Start
7  delay 2.0
8  chat adv1_leyla_01
9  chat adv1_leyla_01b
10 spotlight false Spot_Start
11 move_camera Cam_Start1 2 .6
12 activate_quest adv1_q1
13 delay 2
14 quest_arrow
```

Let's dissect these lines:

**# ADV\_01\_INTRO\_V2**

- # makes the line unreadable to the game, so we can use this comment with

**text cliffs\_area\_title cliffs\_area\_subtitle**

- Taken from [I2Loc Eden Localization](#) Rows 7 & 8 in Chi\_1 Tab

**activate\_quest adv1\_q2**

- Activates this quest so it'll show up in the Quest Log

#### **delay 3.5**

- Literally a time delay of 3.5 seconds

#### **move\_camera Cam\_Start1 2 .21**

- Moves the player camera (Cam\_Start1) over 2 seconds to .21 zoom distance in

#### **delay 2.0**

- Another delay. 2 seconds

chat adv1\_leyla\_01

- ~~Chat dialog from 92Lee Eden Localization in Quests\_Adv\_01 Tab~~

chat adv1\_leyla\_01b

- ~~Chat dialog from 92Lee Eden Localization in Quests\_Adv\_01 Tab~~

Dialog will no longer be done as above. Instead, follow either format below:

- start\_dialogue ley'la nume  
line adv1\_leyla\_01  
line adv1\_num\_01  
end\_dialogue
- dialogue leyla nume adv1\_leyla\_01,adv1\_num\_01

#### **move\_camera Cam\_Start1 2 .6**

- A camera zoom out using Cam\_Start1 over 2 seconds

#### **activate\_quest adv1\_q1**

- Activates this quest so it'll show up in the Quest Log (This is now obsolete as you may now connect a Story to the next Quest via connecting lines. See <https://socialgamingnetwork.jira.com/wiki/spaces/EDEN/pages/2240020715/Level+Designer+s+Tools+Handbook#Connecting-Story-Nodes-to-Next-Quest>).

#### **delay 2**

- Another delay. 2 seconds

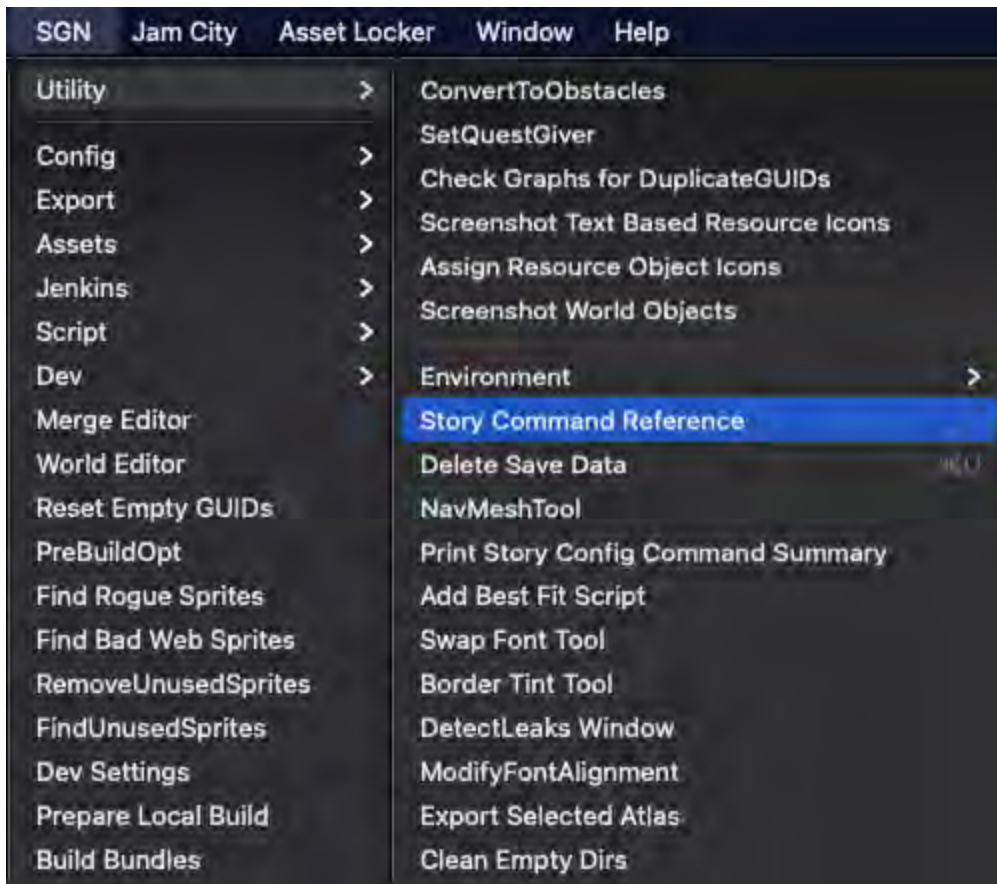
#### **quest\_arrow**

- Plays a left pointing yellow arrow showing where the Quest Log is

Here's a good resource for the most up-to-date Story Config Commands:

- First go to SGN > Utility > Story Command References





- Which leads to this:

<b>unlock_hud:</b>	<i>Unlock a hud element {0}</i>	
	Hud Element Name:	String
<b>unlock_scene:</b>	<i>Unlock scene {0}</i>	
	Scene Name:	String

## Task Action Descriptions:

<b>Get:</b>	<i>Earn resources from any source</i>
<b>Use:</b>	<i>Use resources from any source</i>
<b>Have:</b>	<i>Currently have a balance of specified resource</i>
<b>Interact:</b>	<i>Tap specific world object</i>
<b>MoveTo:</b>	<i>Move character to a specified locator</i>
<b>Remove:</b>	<i>Interact with an object to remove it from the world</i>
<b>Gather:</b>	<i>Get a resource from gathering in the world</i>
<b>Harvest:</b>	<i>Get resource via plot or limited harvester</i>
<b>Craft:</b>	<i>Craft a specified resource</i>
<b>Trade:</b>	<i>Trade at an order board with the specified resources</i>
<b>CompleteOrder:</b>	<i>Complete an order at an order board</i>
<b>Purchase:</b>	<i>Buy an resource in the storefront</i>
<b>Spend:</b>	<i>Use a specific resource in the storefront</i>
<b>Plant:</b>	<i>Use a specific resource to at a plot</i>
<b>Repair:</b>	<i>Repair a building type</i>
<b>Buld:</b>	<i>Buld a specific building</i>
<b>RelationshipLevelUp:</b>	<i>Level up a relationship to a specific level</i>
<b>Reveal:</b>	<i>Reveal a specific object from under fog</i>
<b>TravelTo:</b>	<i>Travel to a specific scene</i>
<b>Generate:</b>	

StoryCommandReferenc...

## Story Config

### Characters:

Ehlu  
Teylu  
Lila  
PewukeTeU  
Ley'la  
Hexapede  
Nume  
Ley'la\_Lulgi

## Command Reference:

Filter:

<b>activate:</b>	<i>Set World Object {0} to Active</i>	
	Object Name:	String
<b>activate_quest:</b>	<i>Activate Quest {0}</i>	
	Quest Id:	String
<b>animate_character:</b>	<i>Animate {0} with {1}</i>	
	Character Id:	Character
	Animation Name:	String
<b>chat:</b>	<i>{0} says: {1}</i>	
	Localization Key:	Localization

<b>clear_fog:</b>	<i>Clear fog area (0)</i>	
	Fog Id:	String
<b>day:</b>	<i>Change to Day</i>	
<b>delay:</b>	<i>Wait for (0) seconds</i>	
	Duration:	Float
<b>infocard:</b>	<i>Show Info Card (0)</i>	
	Info Card Name:	String
<b>map_arrow:</b>	<i>Show the map arrow on hud</i>	
<b>menu:</b>	<i>Show Menu (0)</i>	
	Menu Name:	String
<b>move_camera:</b>	<i>Move camera to (0) over (1) seconds at zoom level (2)</i>	
	Location:	String
	Duration:	Float
	Zoom Level (0-1):	Float
<b>move_character:</b>	<i>(2) (0) to (1). Timing: (3).</i>	
	Character Id:	Character
	Location:	String
	Movement:	WalkOrTeleport
	Timing:	DelayOrContinue
<b>next_quest_graph:</b>	<i>Use next quest graph</i>	
<b>night:</b>	<i>Change to Night</i>	
<b>player_controls:</b>	<i>Set (0) as controllable</i>	
	Character Ids separated by comma:	String
<b>quest_arrow:</b>	<i>Show the quest arrow on hud</i>	
<b>set_active:</b>	<i>Set World Object (0) to Active</i>	
	Object Name:	String
<b>spotlight:</b>	<i>Spotlight (0) at (1)</i>	
	Show:	Bool
	Object Name:	String
<b>text:</b>	<i>Show Text Overlay (0):(1)</i>	
	Title key:	Localization
	Subtitle key:	Localization
	true = Fullscreen style : false = Ban	Bool
<b>trigger_obj:</b>	<i>Trigger World Object (0)</i>	
	Object Name:	String
<b>unlock_hud:</b>	<i>Unlock a hud element (0)</i>	
	Hud Element Name:	String

- I would just keep this open somewhere on your desktop. It will open again if you close Unity. It's a great reference and it's kept up to date by Danny.

**Move\_character** should be written like this:

Move\_character has separate move commands for each movement type:

move\_character\_walk = force character to walk

move\_character\_run = force character to run

move\_character\_teleport = force character to teleport

move\_character = game will decide based on distance

Move\_character (actor name) (Mark\_\* [pointer]) (true=walk/false=teleport( (true=delay/false=continue)



Example:

```
move_character Ley'la Mark_Leyla_Juicer true false
```

- Ley'la is the player character
- Mark\_Leyla\_Juicer is the pointer where the player character will go to
- True means she'll walk to pointer
- False means once the script above is called, the rest of the Story just continues

To move two or more characters at the same time, omit the true false at the end of the move\_character command. Otherwise each character will wait for the previous character's move command to finish before they begin theirs.

NOTE: It is best to move a character and camera all at once by command + selecting both.



## Dialogue

NOTE: When scripting dialogue with more than two characters, use a comma to group them together.

Ex. Start\_dialogue Leylan,Zey Nume

Leyland and Zey will be on the left and Nume will be on the right when the letterbox appears with the talking heads.

## Pathing

```
delay 3
move_on_path zey path_test 5.0
delay 5.0
animate_character zey rope_climb_intro
delay 1
move_on_path zey zey_path 5.0
move_camera end_loc 5.0
animate_character zey rope_climb 0.1
delay 5.0
animate_character zey rope_climb_outro
delay 1
move_on_path zey path_last 3.0
animate_character zey idle
delay 3.0
```

Dissection:

### delay 3

- wait 3 seconds

### move\_on\_path zey path\_test 5.0

- [move command] [character] [name of path] [amount of time to move]

### delay 5.0

- This should match the time in the previous line to allow the character to finish moving on the previous path before starting the next.

### animate\_character zey rope\_climb\_intro

- [start animation] [character] [name of animation] for the path line above.

### delay 1

- one second wait

### move\_on\_path zey zey\_path 5.0

- [move command] [character] [name of path] [amount of time to move]

### move\_camera end\_loc 5.0

- [move camera] [camera name] [time to move]

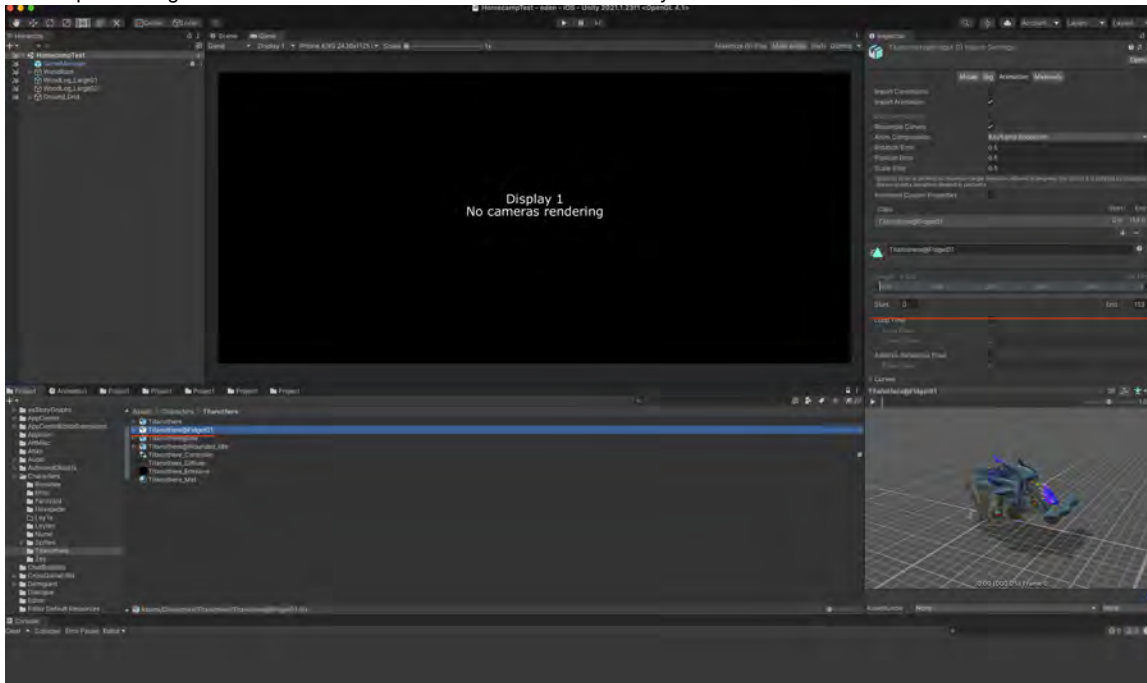
### delay 5.0

- This should match the time in the previous line.

A path sequence is made of 3 parts, an intro, the main path, and an outro. The intro and outro are animations to transition the character naturally into and out of the main path animation, but the scripted order sequences are the same.

### Animation Length

To look up how long an animation takes in order to add the correct delay:



30 frames is a second, so 153 frames would be 5.1 seconds.

### Follow Camera

A scriptable command to have the camera follow a game object

```
# FOLLOW CAMERA TEST

# set camera to follow leylan
follow_camera leylan

# move leylan to loc, wait until reaches destination
move_character_walk leylan loc true

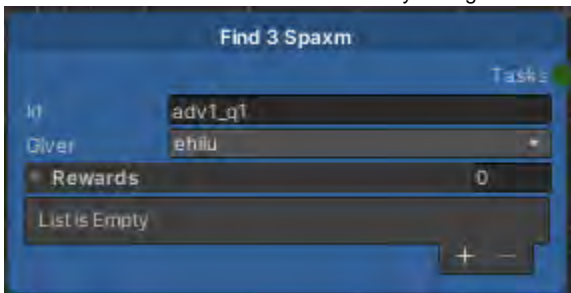
# set camera to follow NULL ( stop following )
follow_camera NULL
```

### Quest

Along with Chapter, this script node has very few entries required to be filled.



- Id - quest ID taken from [I2Loc\\_Eden\\_Localiaztion](#) in the level's tab
  - This will also automatically change the title of the Quest box to whatever is written in the Loc file like this:

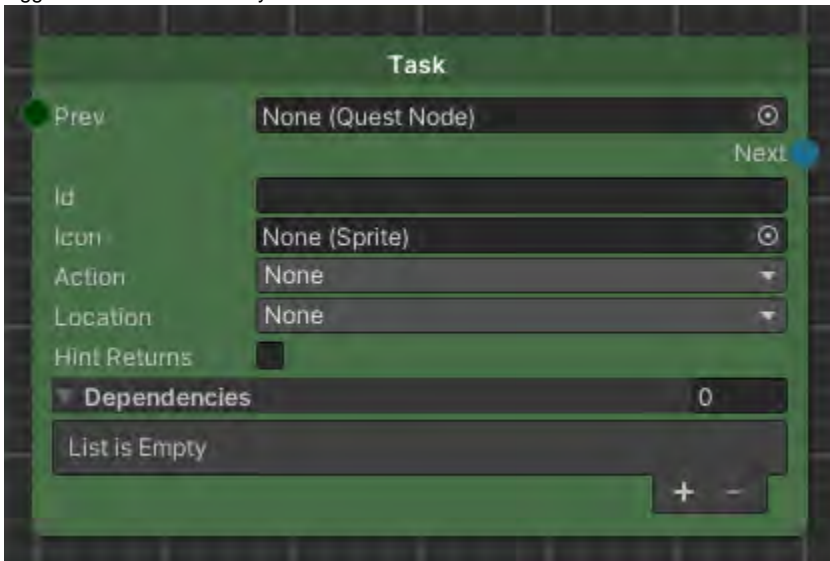


- Giver - character name (work with the Narrative team to determine who gives a quest)
- Rewards - established by systems design, so work with them for this information

This is mainly a start point for new quests, called upon from the previous quest's story.

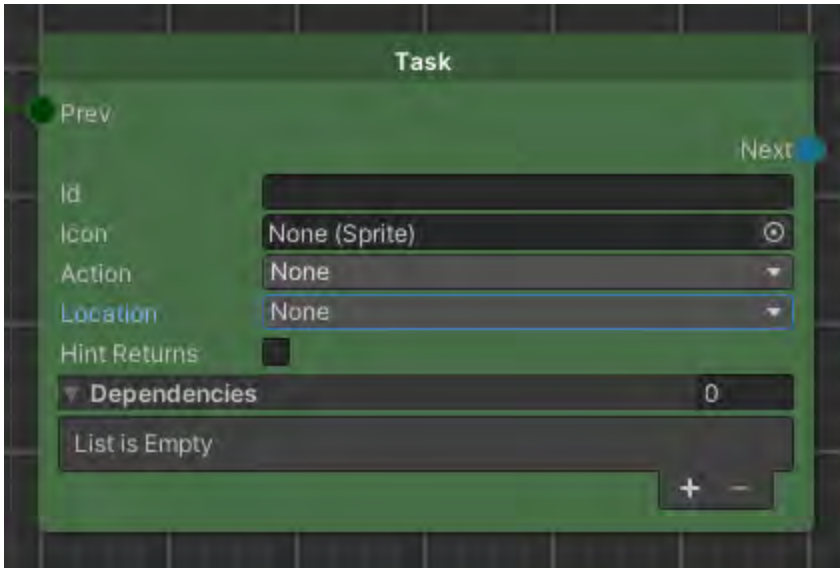
### Task

As mentioned, this is your bread and butter work in QuestGraph. This is the direct interface with your level where World Object interactions trigger events as dictated by the Task's Actions.



As with the Story node, when connected to the Task's input point, the upper most field will disappear like this:

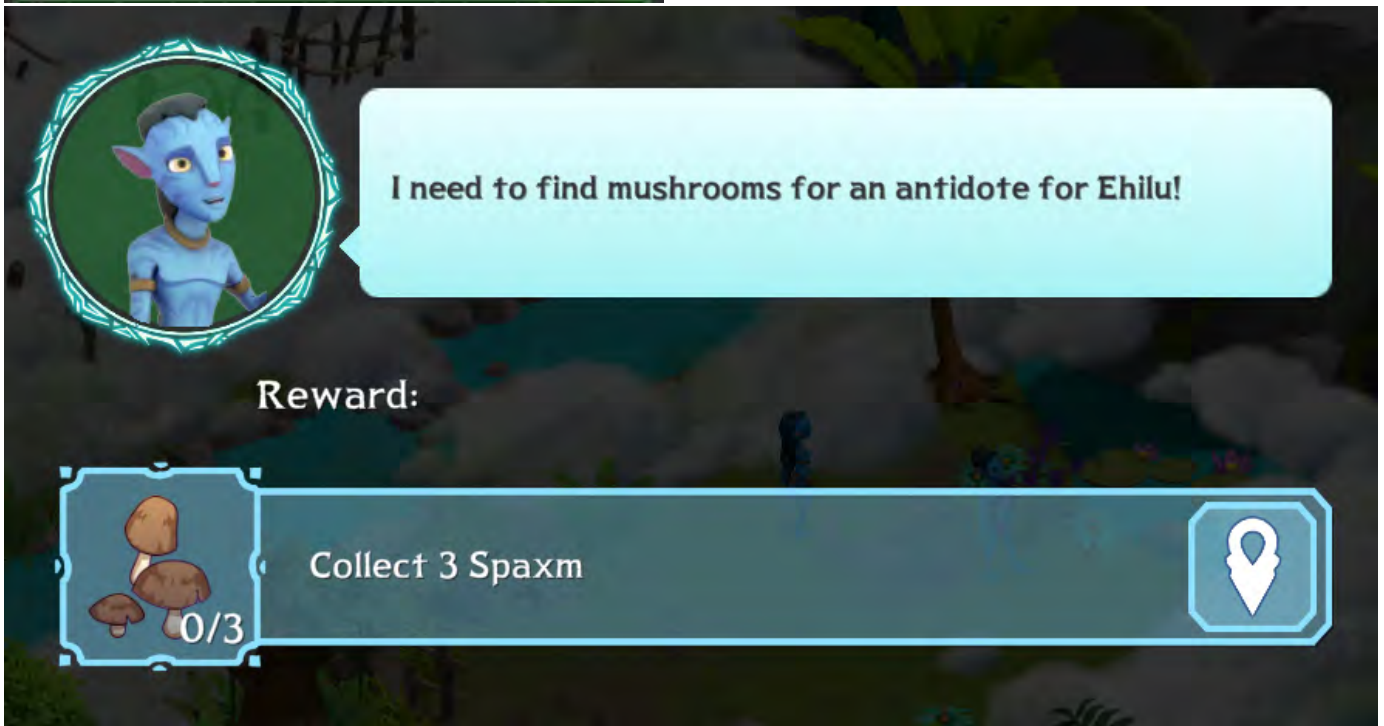
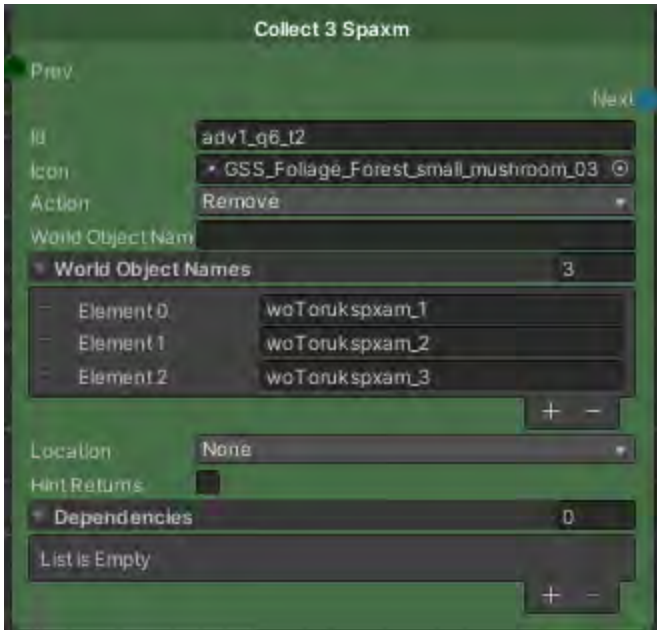




Adding in the Task Id will also populate the Task node's title with the name from [l2Loc\\_Eden\\_Localiaztion](#) and the tasks from the level's tab. The task Id is the quest Id with \_t# added to the end.

FIND 3 SPAXM





## Types of Tasks

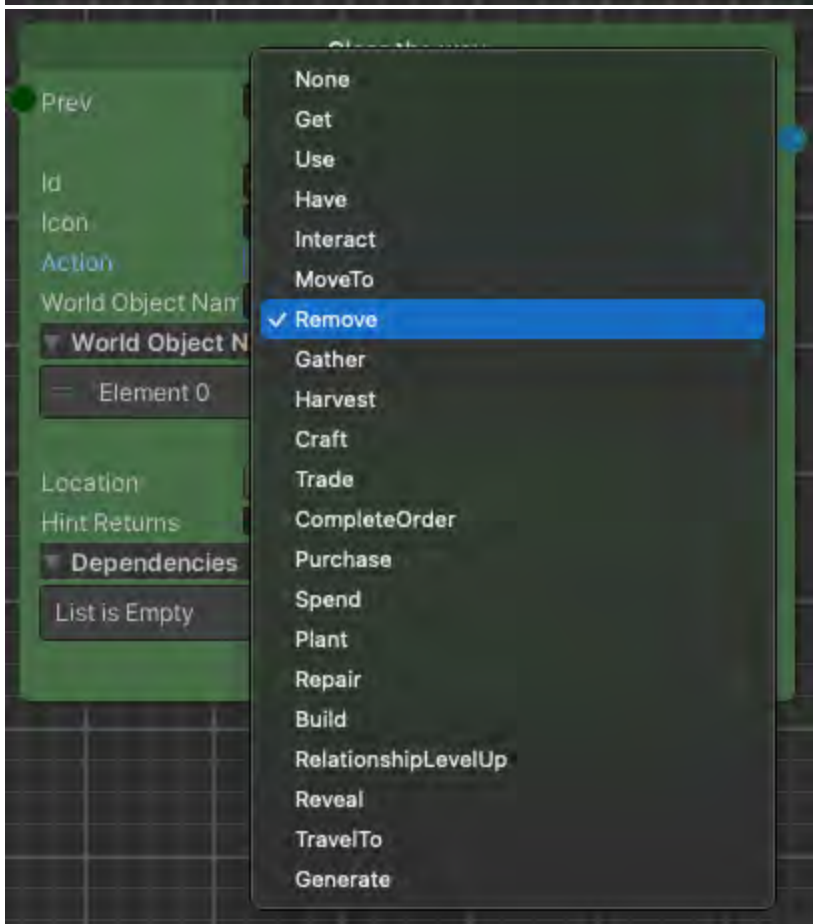
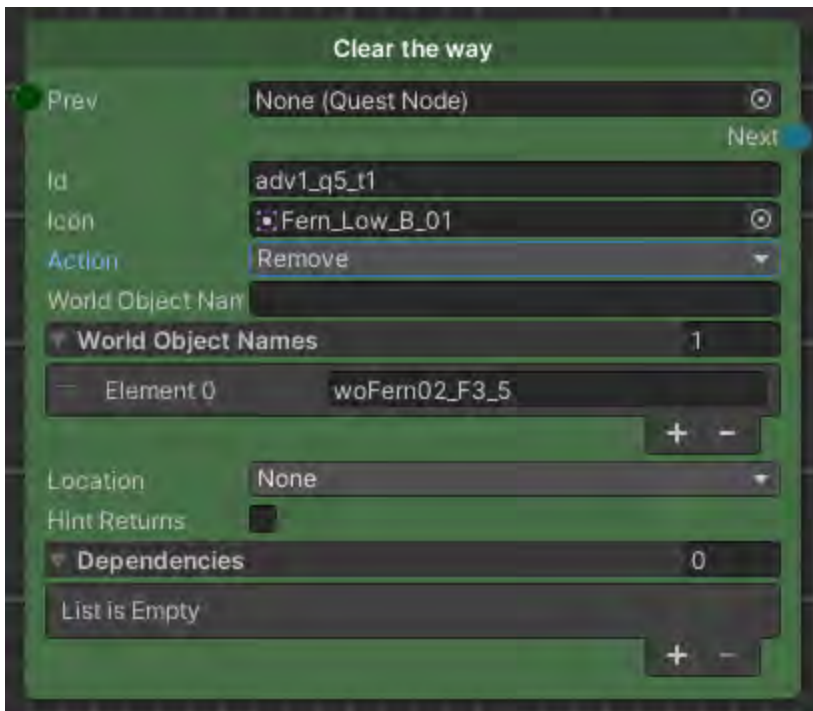
There are many types of tasks but in the examples below, we'll go over 3 of the most common Actions for tasks.

### Obstacles and Decorations

Remove

This is probably the most common task. Choose Remove from the dropdown menu for Action.





You can list as many World Object Names as you need by adding more lines under the World Object Names list by clicking on the + icon.



Once all of the listed WOs are interacted with, the task will be complete.

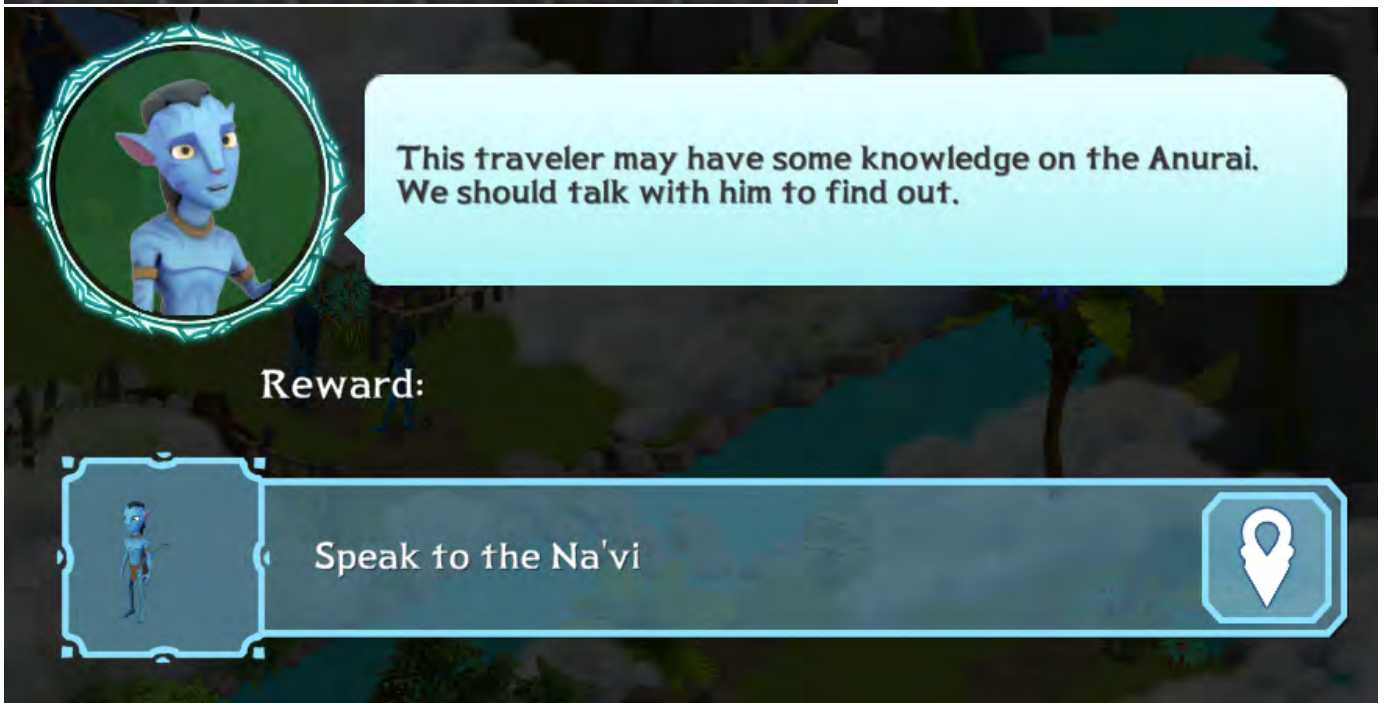
Interact

This is literally requiring the player to tap on the designated World Object.

# TALK TO THE STRANGE NA'VI





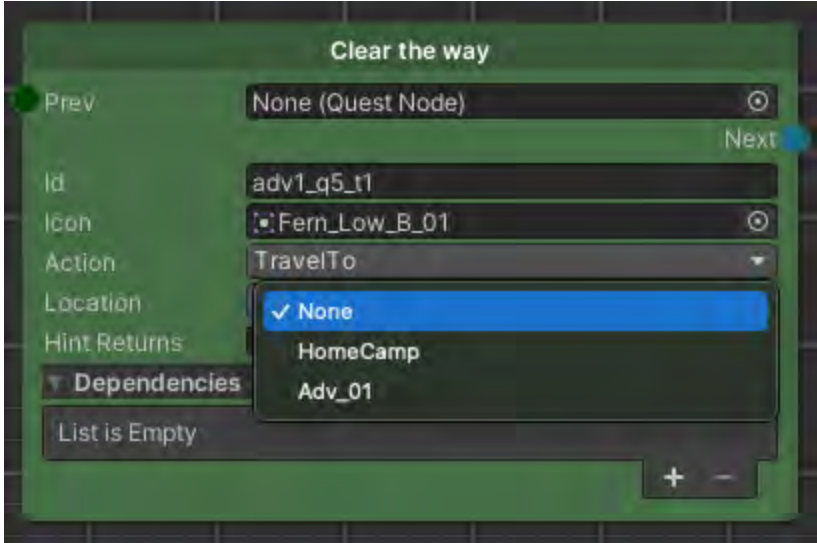
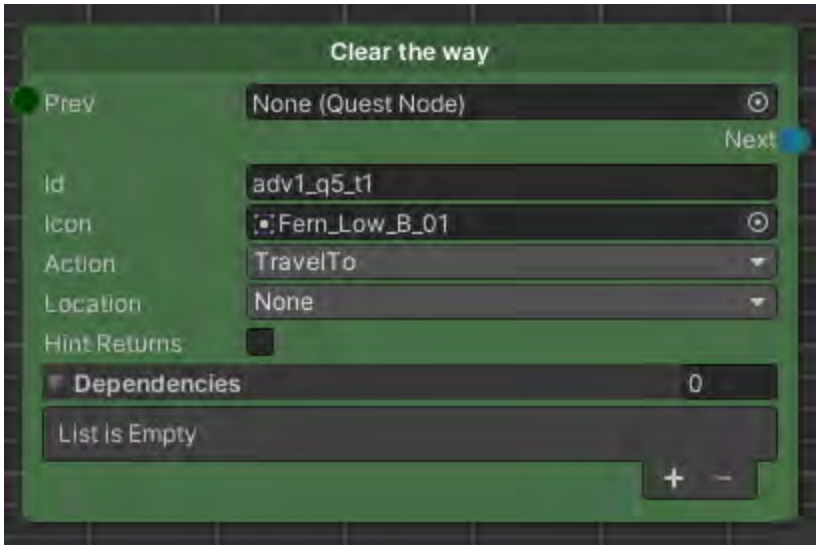


Here, the World Object is woNaviTrader.

Once the player taps this object, the task is complete and the Next output will be called, it goes to Story, and the next script will play.

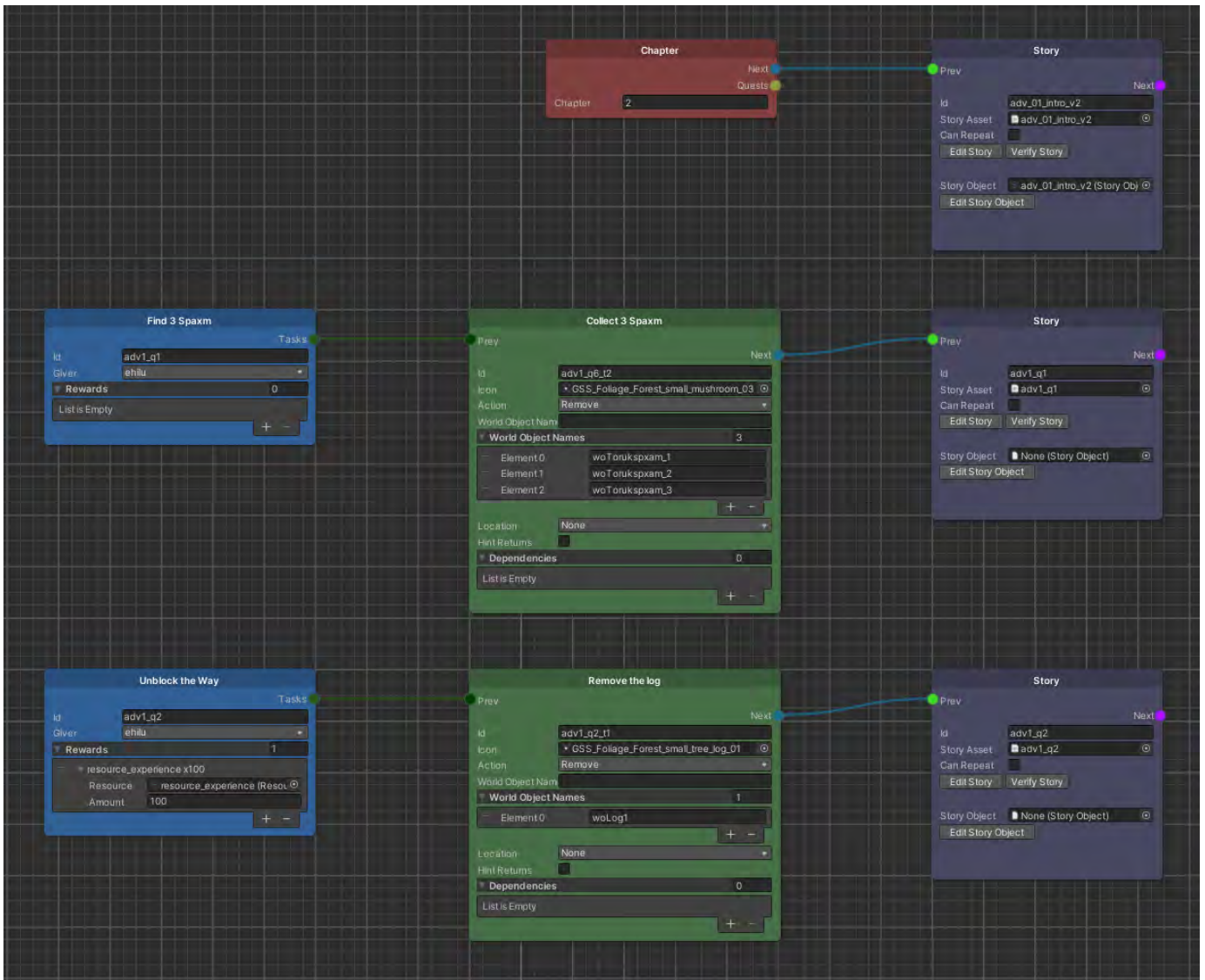
Travel To

This task is used if the story or task requires the player to go back to the Home Camp to craft or retrieve something that will be used to complete a task.



This will result in a Quest Log update that looks like this:

QuestGraph Flow

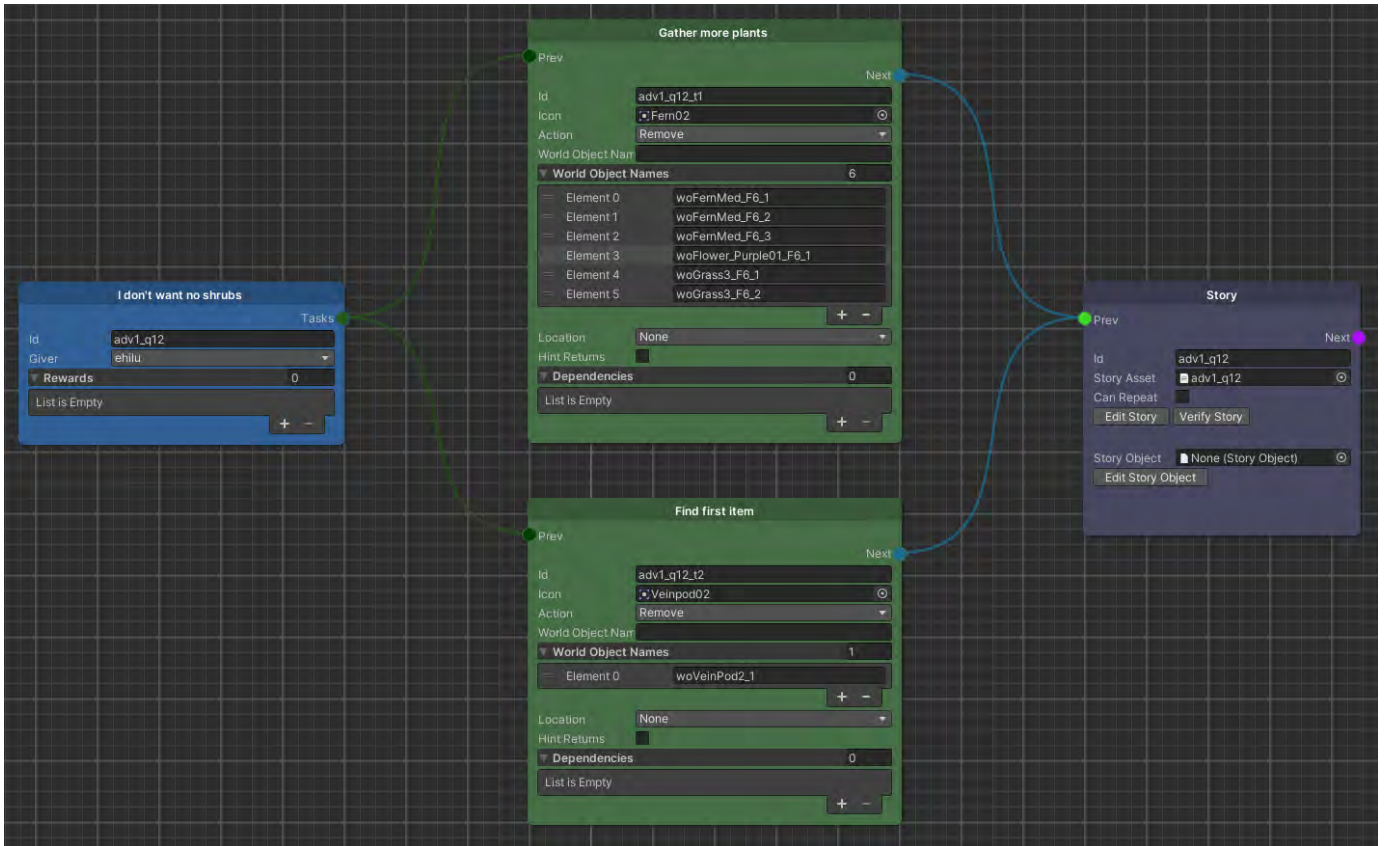


Typically, your QuestGraph quests come in 3s: Quest, Task, and Story from left to right.

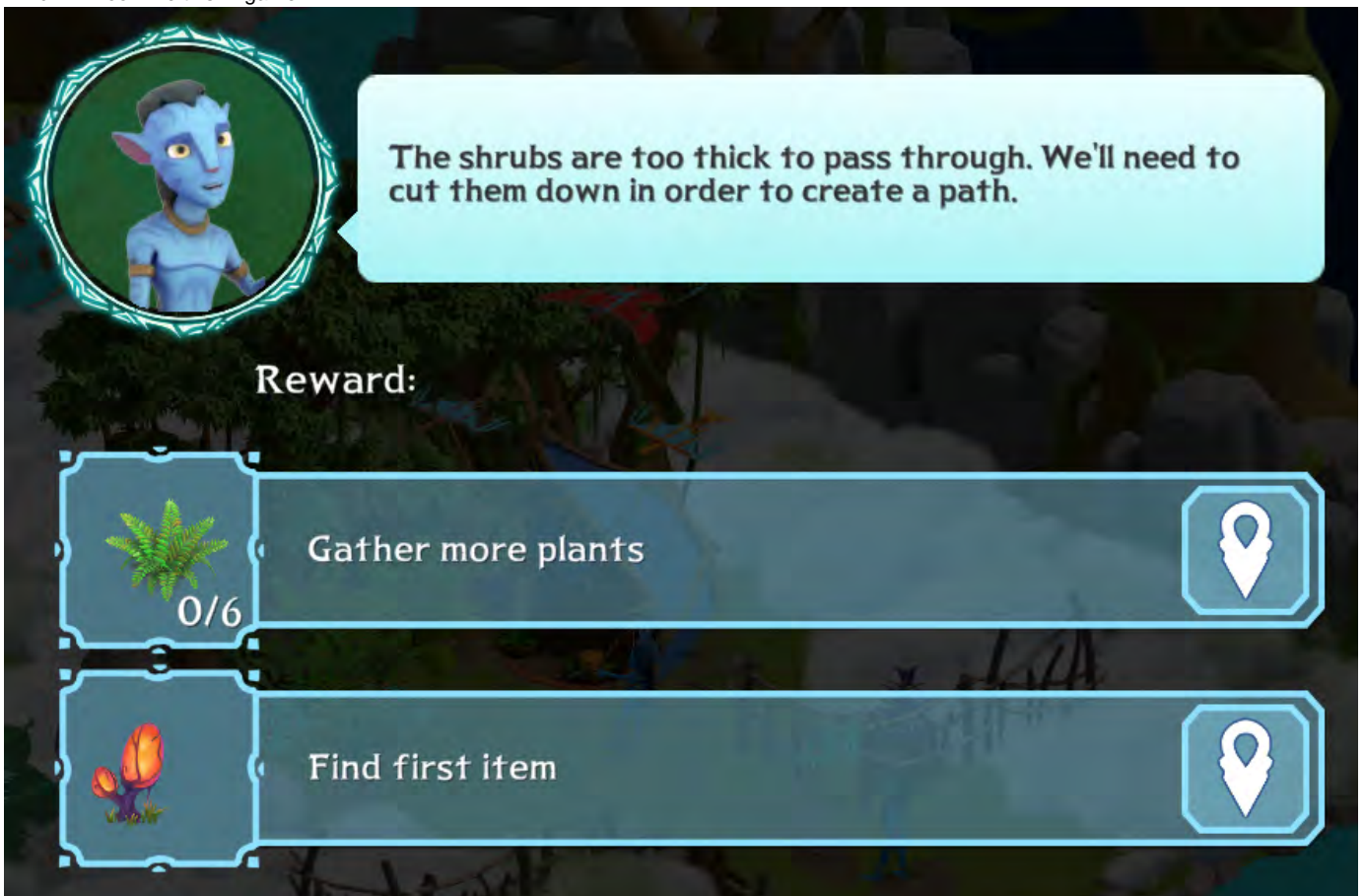
The three are linked by connecting splines. No need to draw an output from Story to the next quest as the next quest is always written in the previous Story's script.

You can have multiple Tasks that need to be completed before that quest's Story is triggered.





Which will look like this in-game:



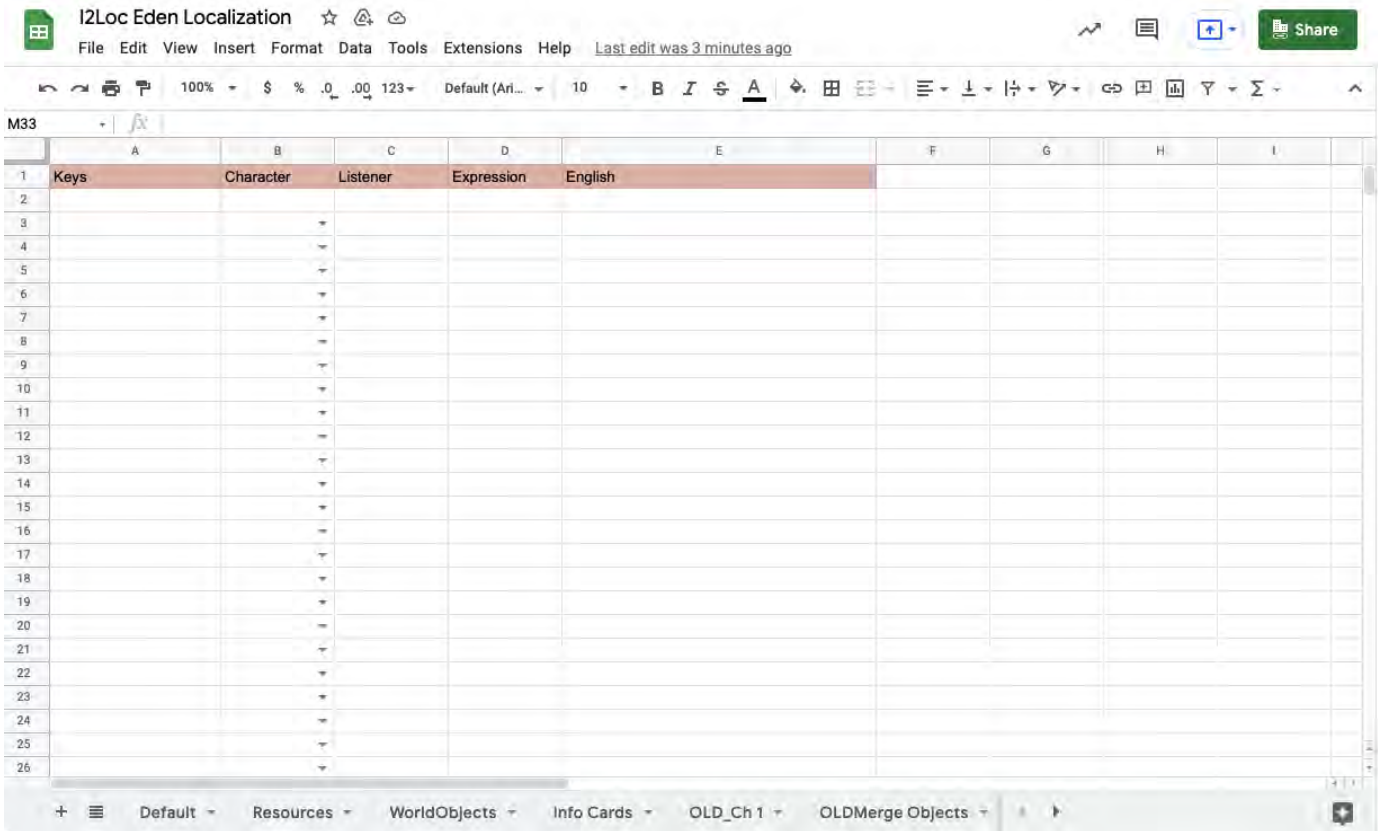




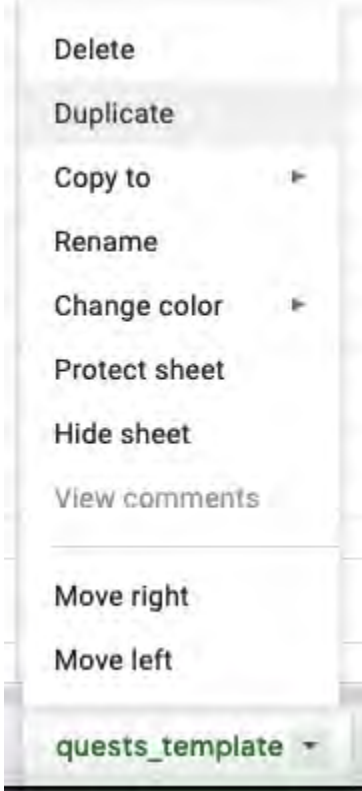
#### Quest, Task, Names, and Dialogue - The Loc File

There is a localization table that all of our quests, tasks, and dialog come from. The table is located here: <https://docs.google.com/spreadsheets/d/15pJCeN9YN0nQXY0Wums0v7zQRaT4CLBcAfetXpgUcrs/edit?usp=sharing>

This spreadsheet will have tabs that are level specific. Narrative will fill in the necessary text dialog but LDs will put together the quest structure, tasks associated with the quests, and names/titles.



Look for the **quests\_template** tab at the bottom of the spreadsheet and duplicate it. Rename it to **quests\_LEVELNAMEHERE**.

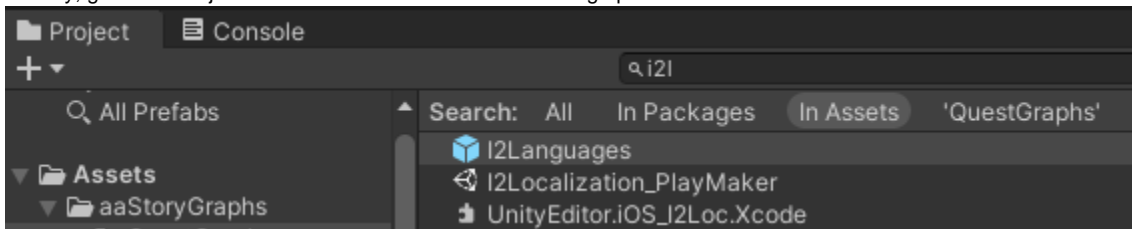


Fill in your quest text and details like the example below:

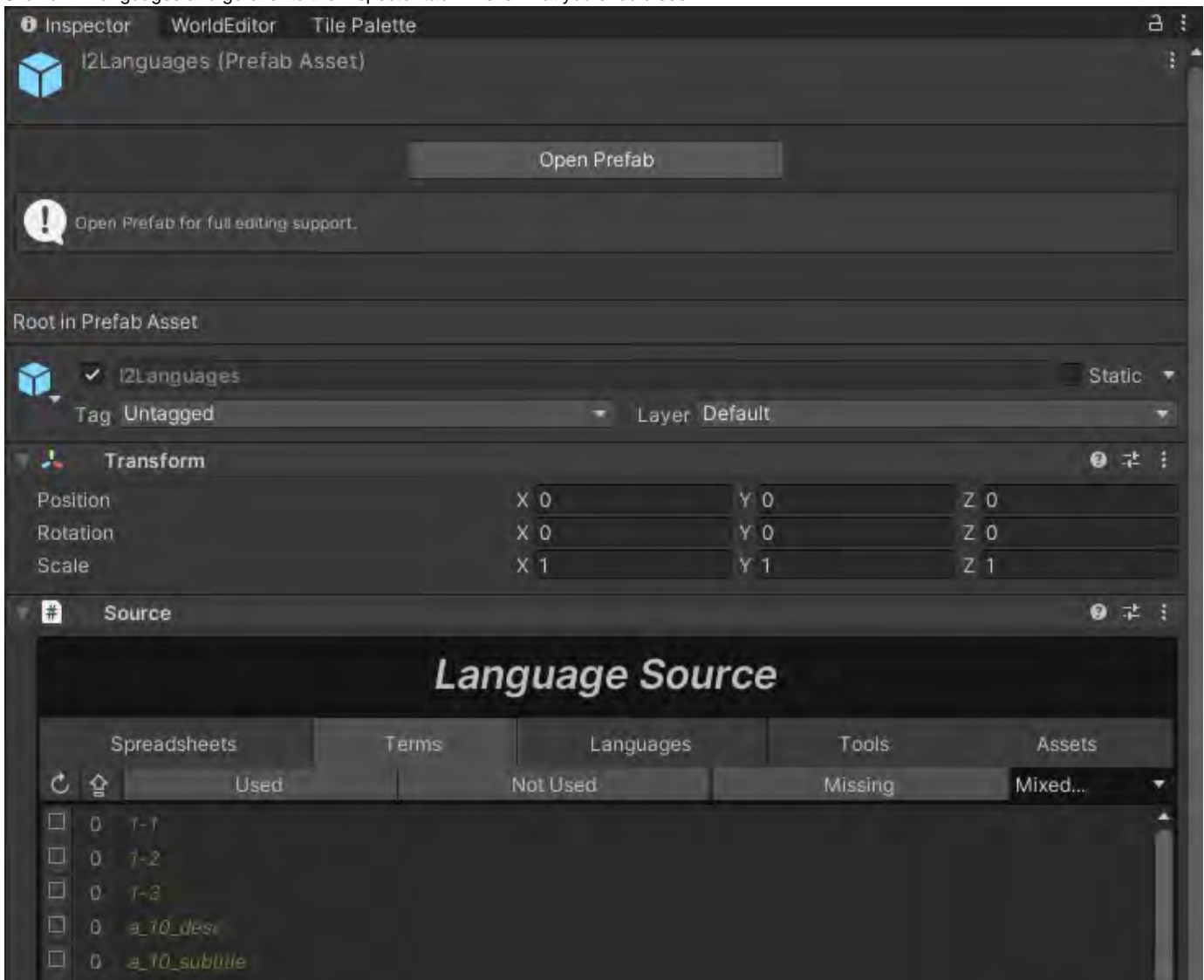
	A	B	C	D	E
1	Keys	Character	Listener	Expression	English
2	KimTest_q1_title				The cat's outta the bag
3	KimTest_q1_desc				Go to Meow Jungle
4	KimTest_q1_t1_desc				Travel to Meow Jungle
5					
6	KimTest_q2_title				A boquet for Oketra
7	KimTest_q2_desc				Collect flowers
8	KimTest_q2_t1_desc				Gather flowers for the cat god's tribute
9					

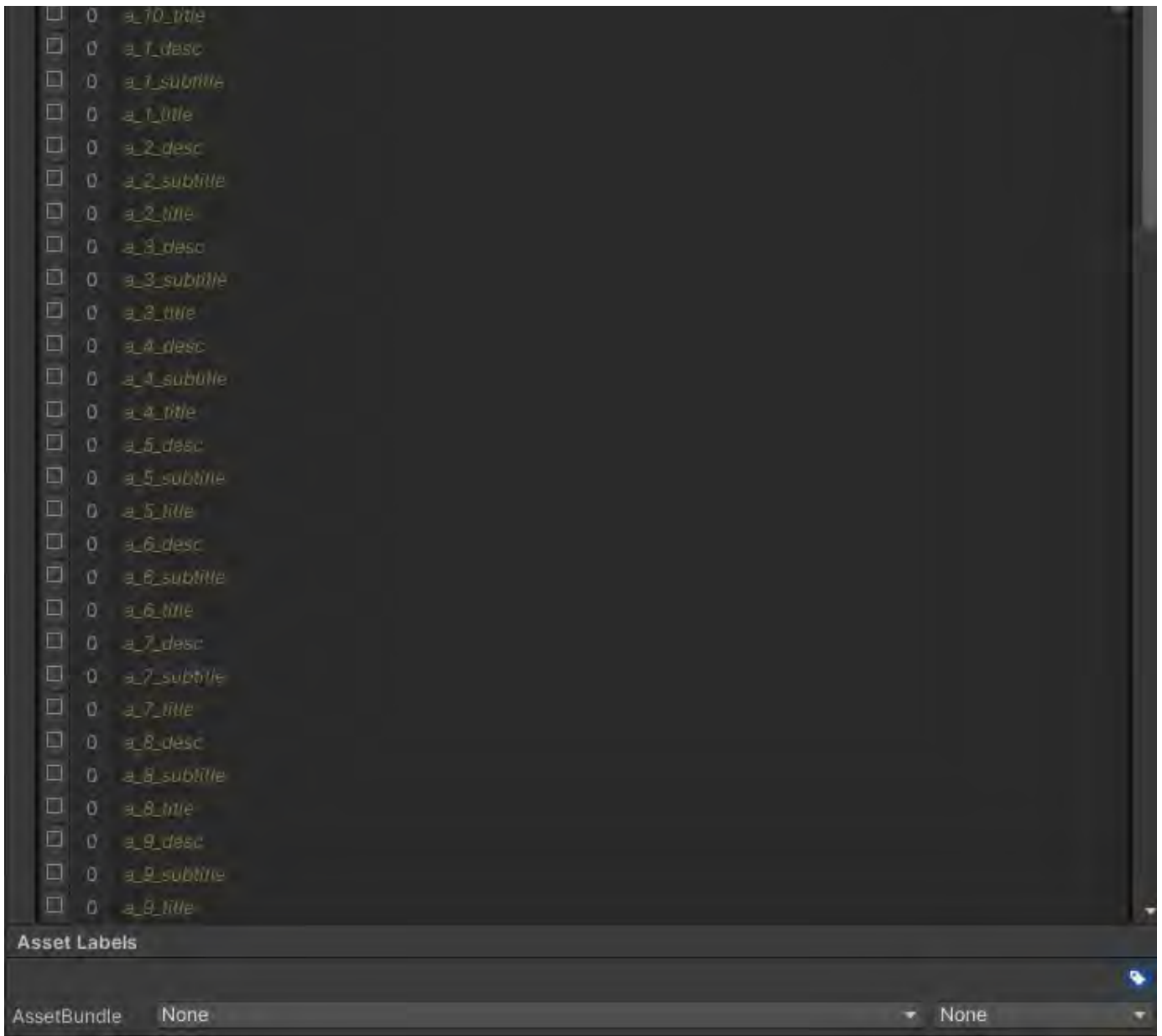
Once you're happy with your edits to this spreadsheet, you need to import them into the game. Here's how you do that:

In Unity, go to the Projects tab and search for i2l. This will bring up this result:



Click on I2Languages and go over to the Inspector tab. This is what you should see:





Click on the Spreadsheet tab under Language Source. The following should appear.



# Language Source

Spreadsheets

Terms

Languages

Tools

Assets

Local

Google

Web Service URL:

Install

Verify

In Google Drive:

New

Open

Refresh

Import

Replace

Merge

Add New

Export

Replace

Merge

Add New

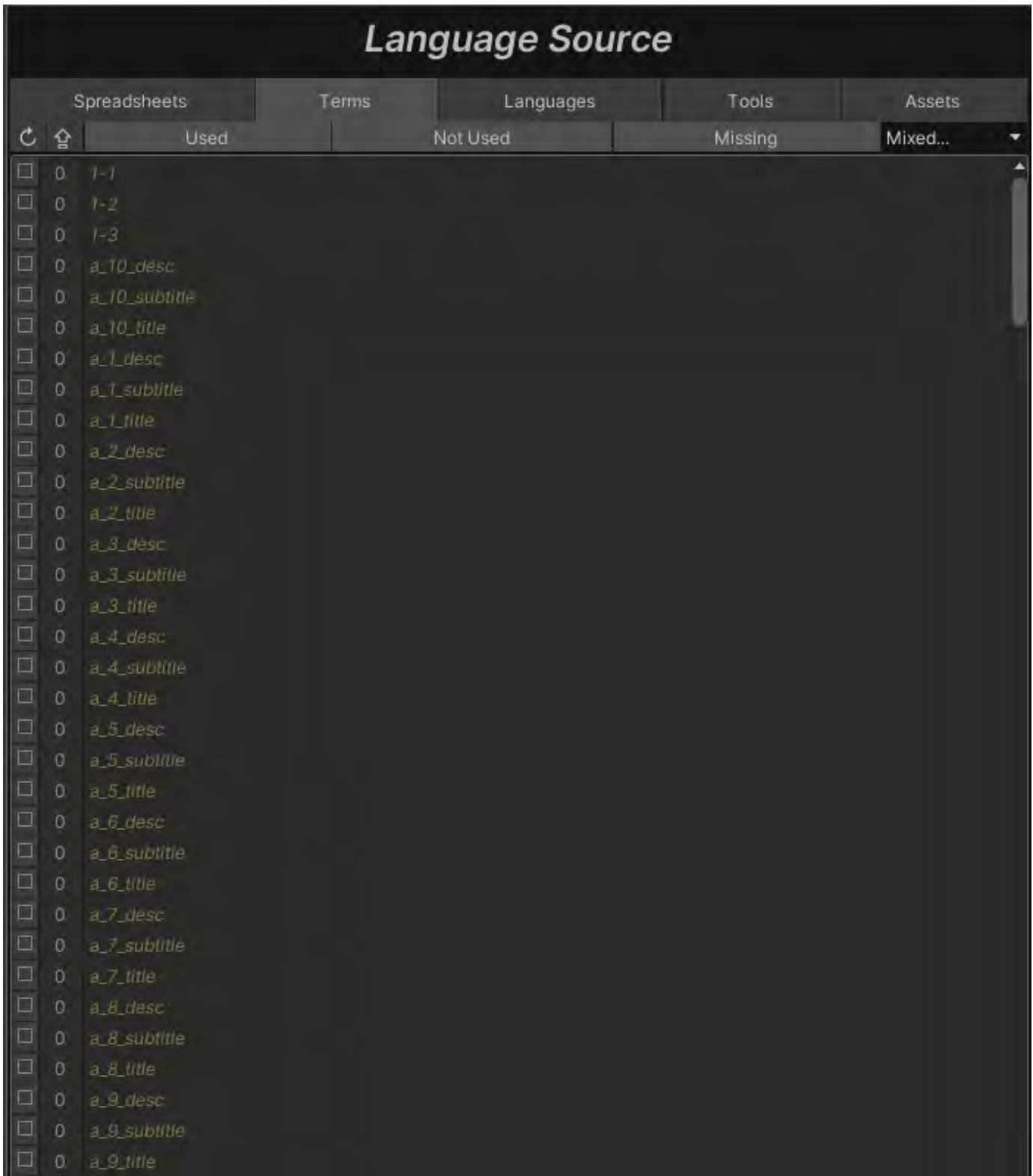
Open Spreadsheet after Export

v2.8.1 a7

[Ask a Question](#)

[Documentation](#)

Click on Replace in the Import section, bottom left of the above screen. When finished Replacing, the window will return to the following state:



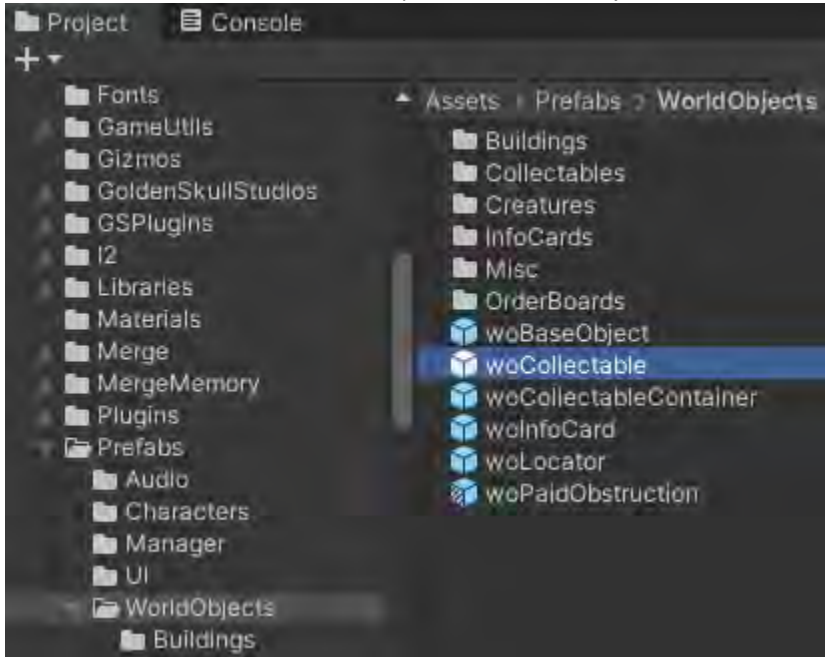
You've successfully updated the loc strings and you should immediately see the changes in your QuestGraph node titles and in-game chat bubbles.

## Create World Object Prefabs

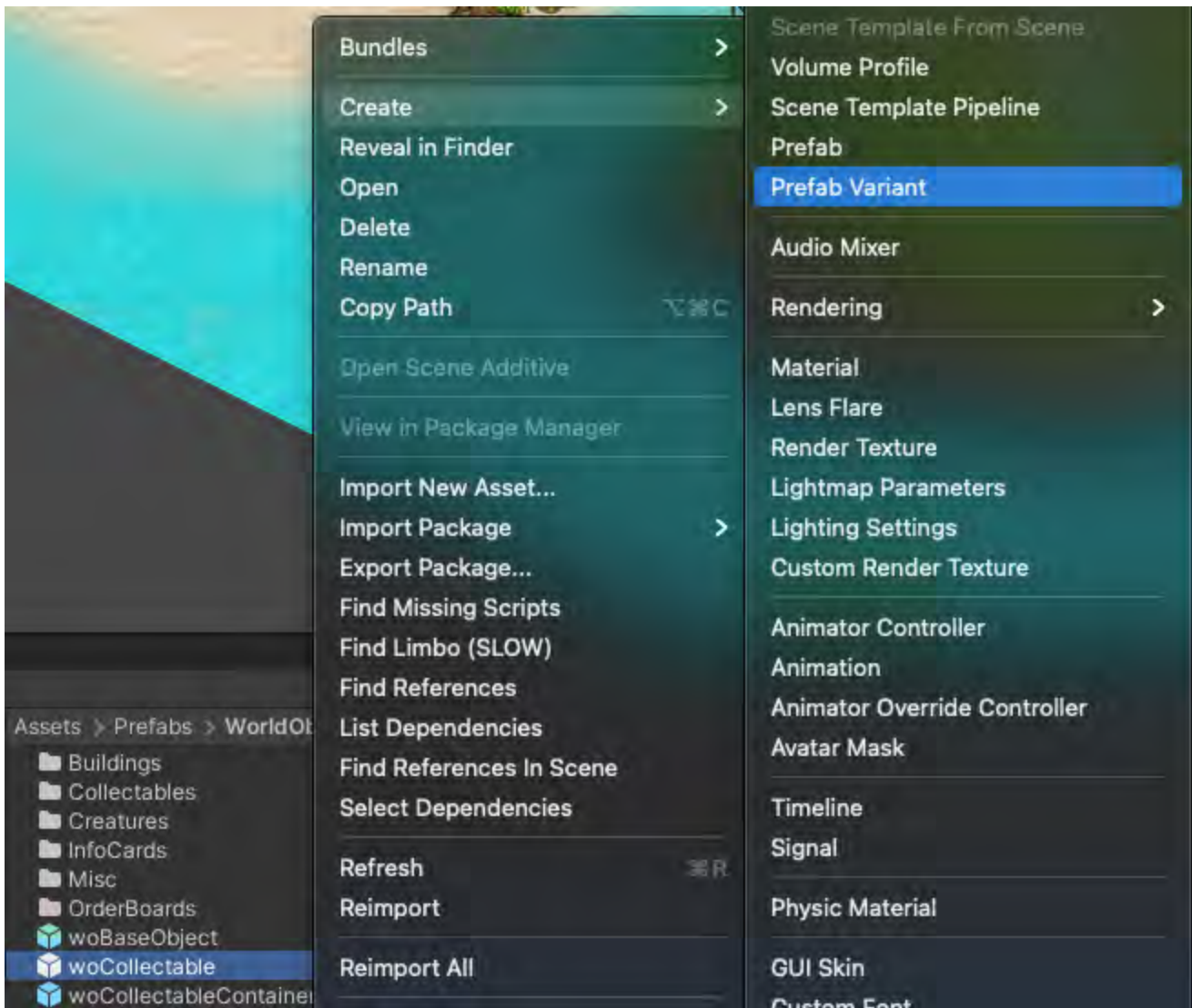
[Step By Step Instructions](#)

In Unity, in the Project tab, find Assets > Prefabs > World Objects

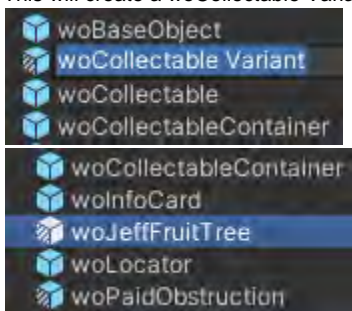
Find the woCollectable file. This is the template for collectable objects.



Right click on woCollectable > Create > PrefabVariant

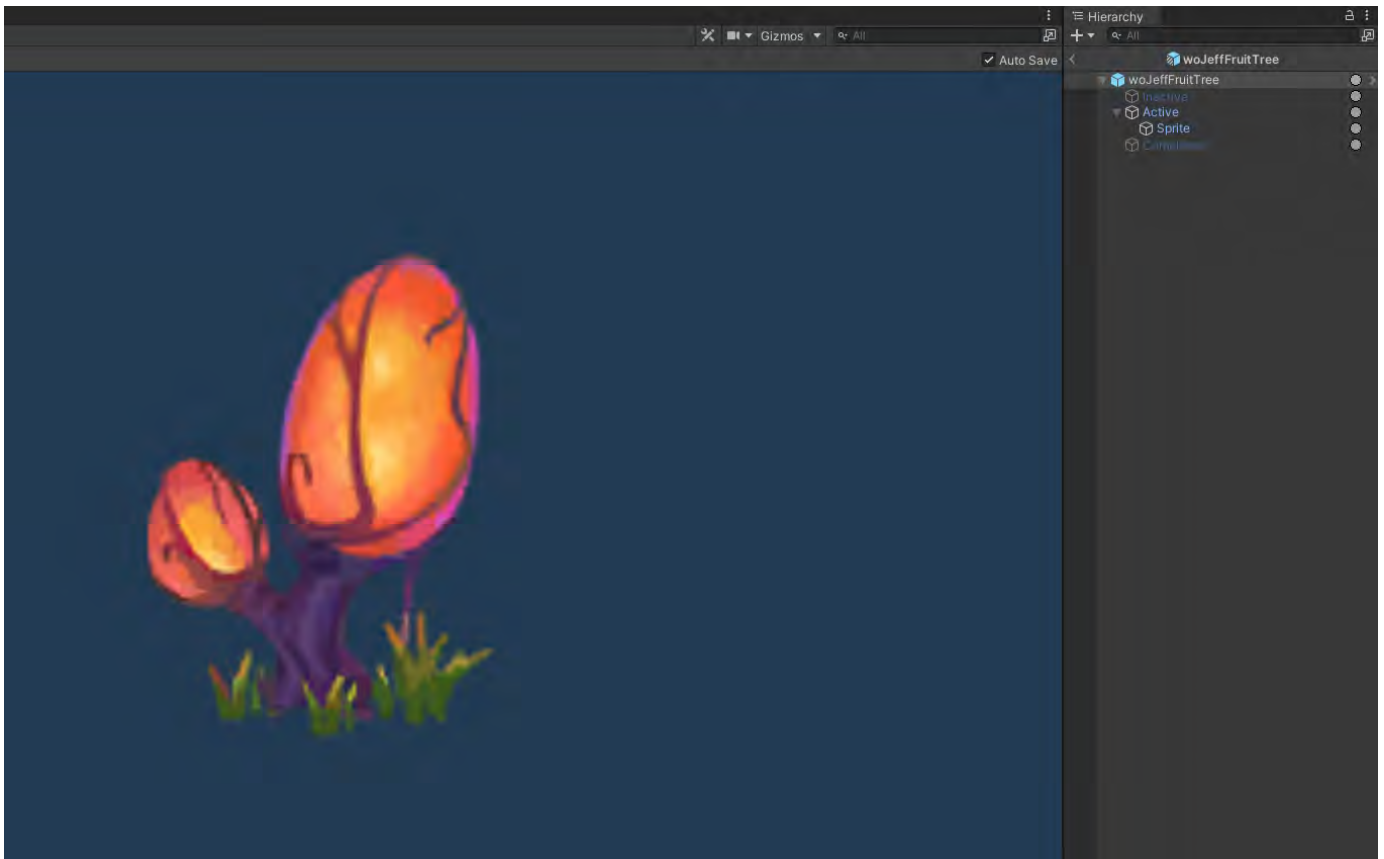


This will create a woCollectable Variant. Change this name.

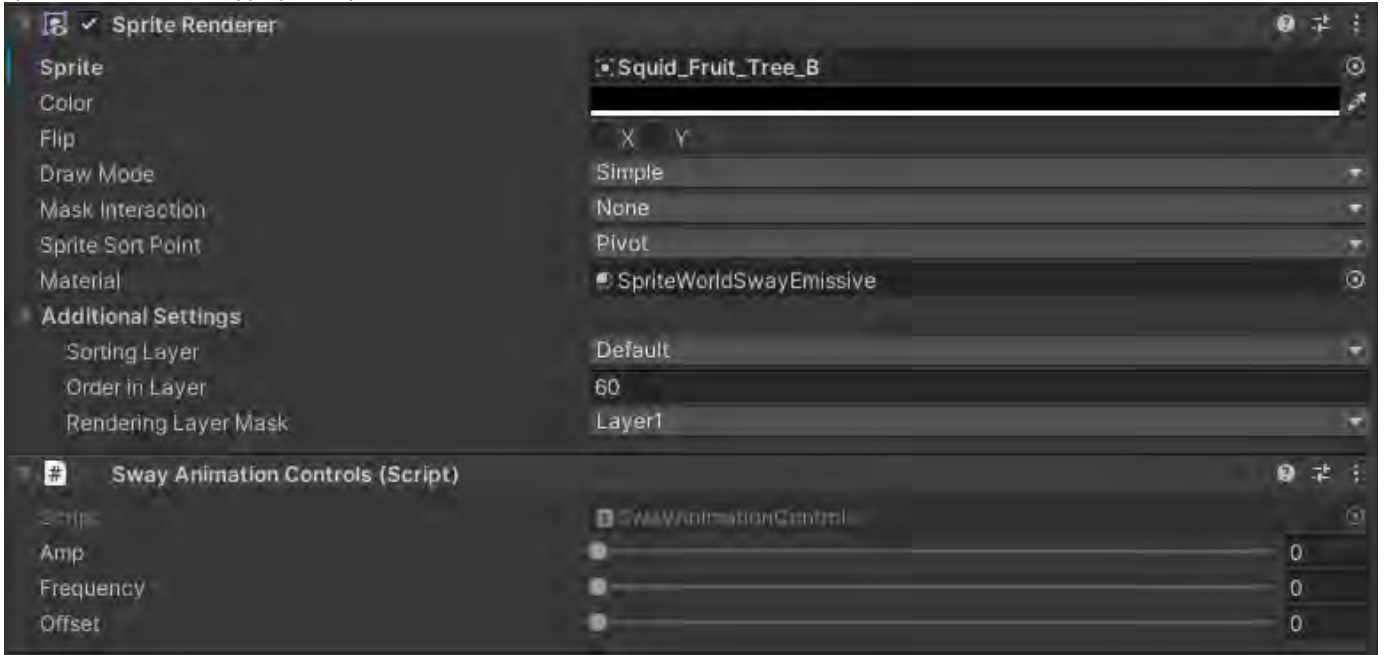


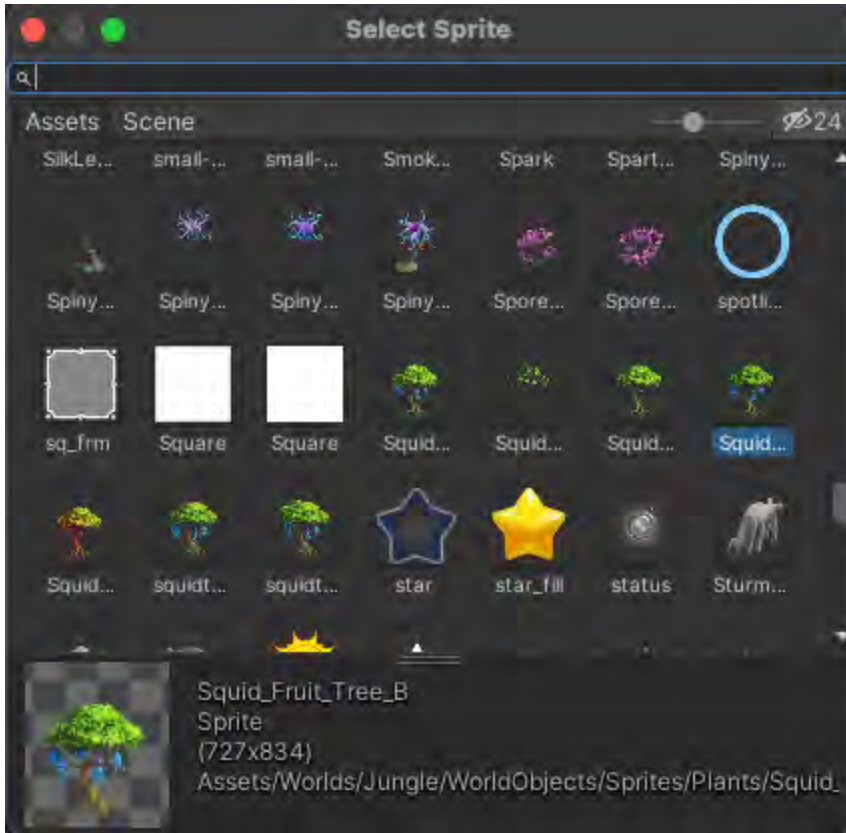
Double click on woJeffFruitTree (of course, whatever you called it). You should see an orange vein pod.





Go to the Sprite in the Hierarchy under Active. Now click on the Inspector tab, scroll down to Sprite Renderer and the first line there. Find the Sprite icon and look an appropriate sprite.





Now, your new prefab items should show up in the World Object library.



Danny's Confluence/Video Example

[Collectables - Making World Object Prefabs](#)

[Relationship Graphs](#)

[Edit World Object Tappable Area](#)

## Best Practices

This is list of best practices but are not in any particular order.

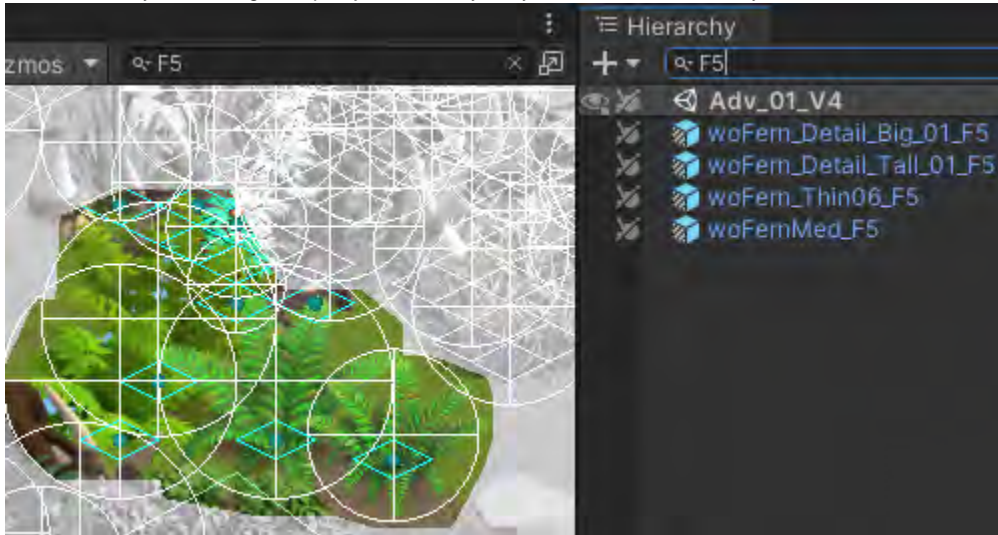
For the standard approach to design please refer to:

[Level Design Style Guide](#)

[Plant Placement Process](#)

- Prefab edits and pushes:
  - LDs are primary implementers on Adv levels. If you need to make changes in adv levels, check with that level's LD first and coordinate a time to make your changes.
  - And vice versa. If anyone needs to make changes in Home Camp, double check with Rose first and coordinate.

- Also suggest that everyone make incremental pushes all day instead of holding on to changes the whole day. That way if something goes wrong, you'll only lose a fraction of what you would if you didn't make incremental changes.
- When adding World Objects to a Fog Group, rename that object and append it with F#. For example:
  - woLog becomes woLog\_F3
  - This log belongs to the F3 fog group.
- In the Hierarchy, these Fog Group objects are very easy to find via the Hierarchy's search field



- When creating a new QuestGraph for a new level, please be sure to add a task to travel to the new level in the Story (press the Edit Story button) and unlock the level so that the level map that shows all the levels actually unlocks the intended level to be unlocked.