Creating Custom Scripts for Spatial Projects

Table of Contents

Basics

-What does a Script do? Slide 3

-How do we add Scripts to Objects? Slide 4

-What are Visual Scripts? Slide 5

Creating Visual Scripts

-Adding a Visual Script to an Object Slide 7

-Writing your first Visual Script Graph Slide 9

-Spatial Nodes Slide 15

-Spatial Data Store Slide 17

Prefab Objects

-What are Spatial Prefab Objects? Slide 17

-Adding Prefab Objects to your Space Slide 22

-Creating custom Prefab Objects Slide 25

-Getting your Prefab Objects in a Space Slide 29

Basics

What does a Script do?

Inside of your custom Spaces, you can have scripts attached to objects that customize the object's behavior.

For example, you can make objects move, clone themselves, have certain interactions with the user, and more.

Spatial's golf cart object in their Golf Cart Driving example scene. It can be driven using scripts.

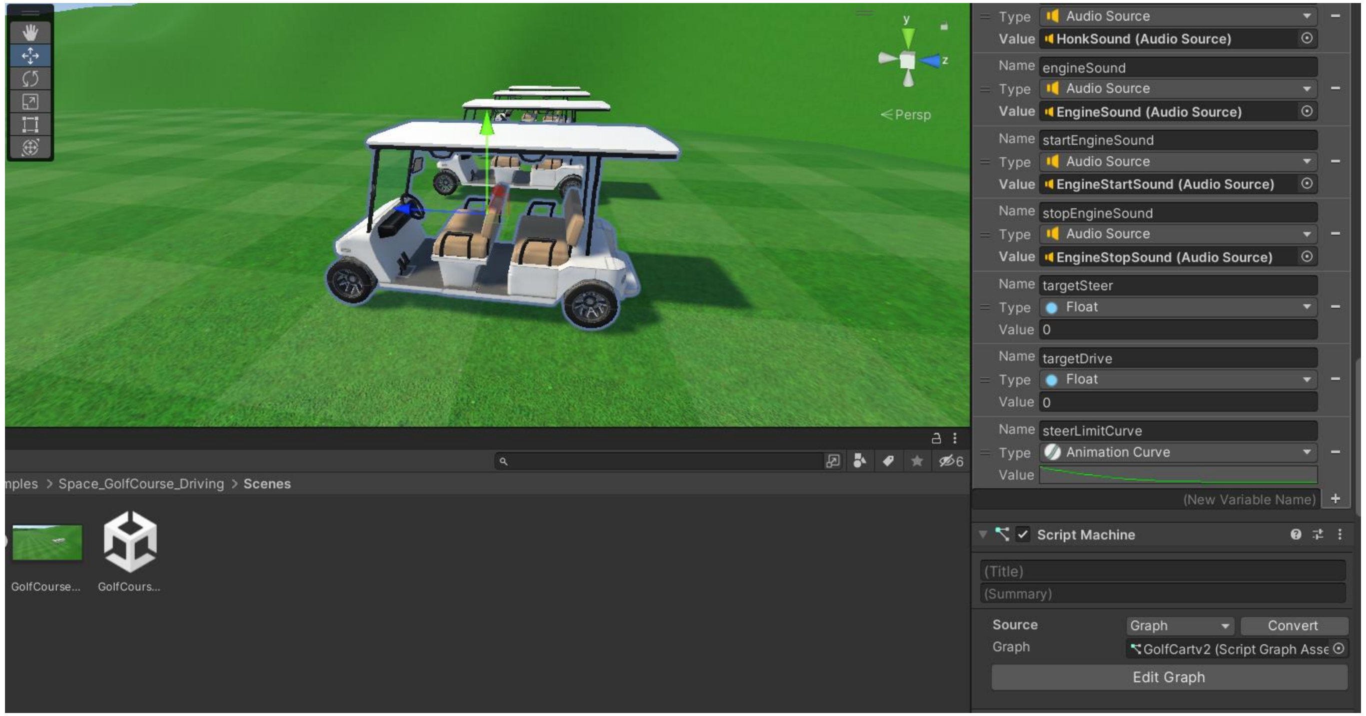
⬇ Some slides will have additional information in the "Notes" section below. Look for the down arrow!

Basics

How do we add Scripts to Objects?

Scripts must be added to objects inside of your Spatial project in Unity Editor.

To learn how to create Spatial projects in Unity, see my documentation on Creating Spaces.



Type: Audio Source
Value: HonkSound (Audio Source)

Name: engineSound
Type: Audio Source
Value: EngineSound (Audio Source)

Name: startEngineSound
Type: Audio Source
Value: EngineStartSound (Audio Source)

Name: stopEngineSound
Type: Audio Source
Value: EngineStopSound (Audio Source)

Name: targetSteer
Type: Float
Value: 0

Name: targetDrive
Type: Float
Value: 0

Name: steerLimitCurve
Type: Animation Curve
Value

(New Variable Name) +

☑ Script Machine

(Title)
(Summary)

Source: Graph — Convert
Graph: GolfCartv2 (Script Graph Asse

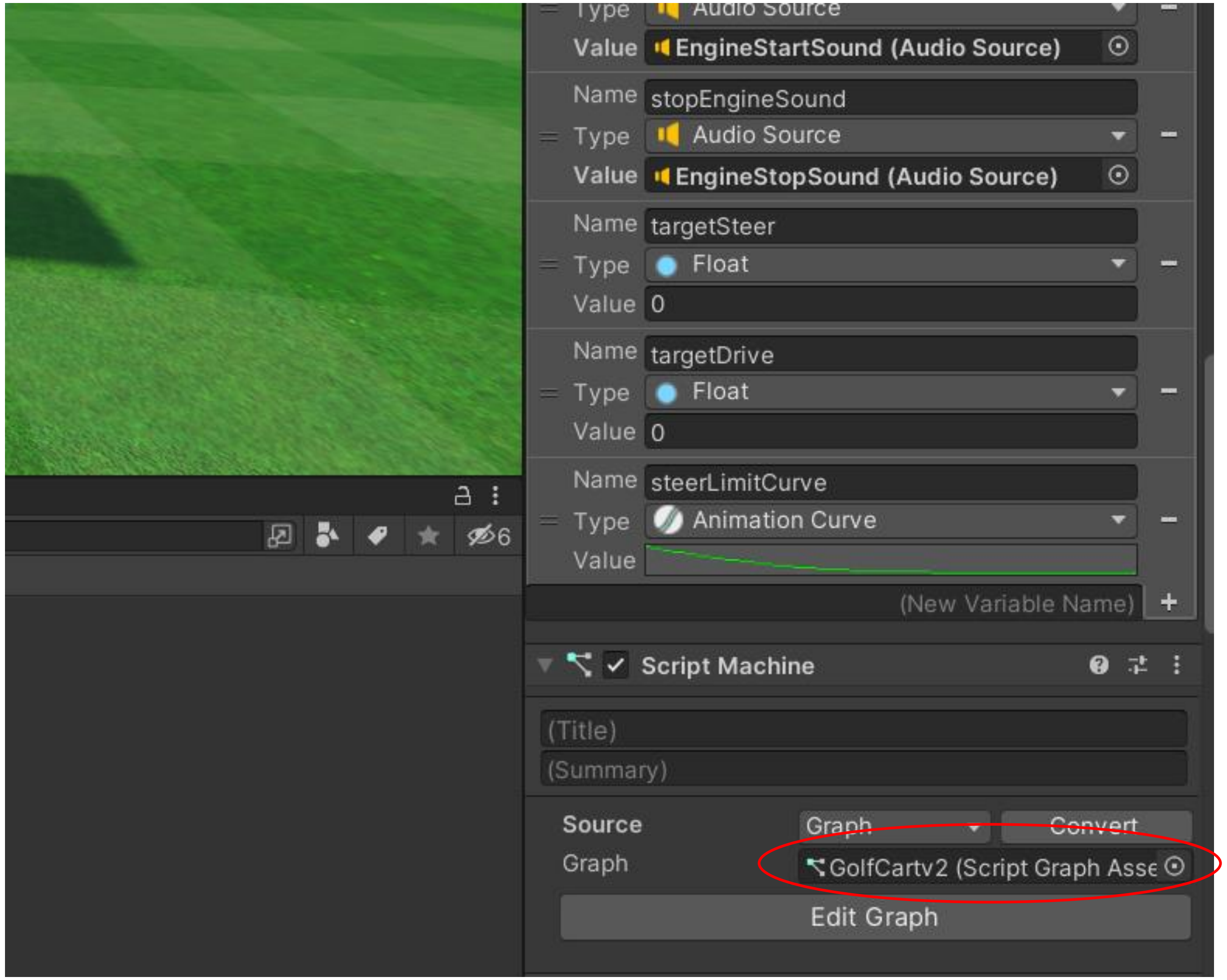
Edit Graph

Basics

What are Visual Scripts?

"Visual Scripting" is a type of scripting that is created using nodes that are connected in a graph.

Spatial only supports visual scripting. C# scripts or scripts in other languages will not compile in Spatial*.



Type Audio Source
Value EngineStartSound (Audio Source)

Name stopEngineSound
Type Audio Source
Value EngineStopSound (Audio Source)

Name targetSteer
Type Float
Value 0

Name targetDrive
Type Float
Value 0

Name steerLimitCurve
Type Animation Curve
Value

(New Variable Name)

Script Machine

(Title)
(Summary)

Source Graph
Graph GolfCartv2 (Script Graph Asse

The visual script that is used for Spatial's golf cart, GolfCartv2.

Additional Notes Below

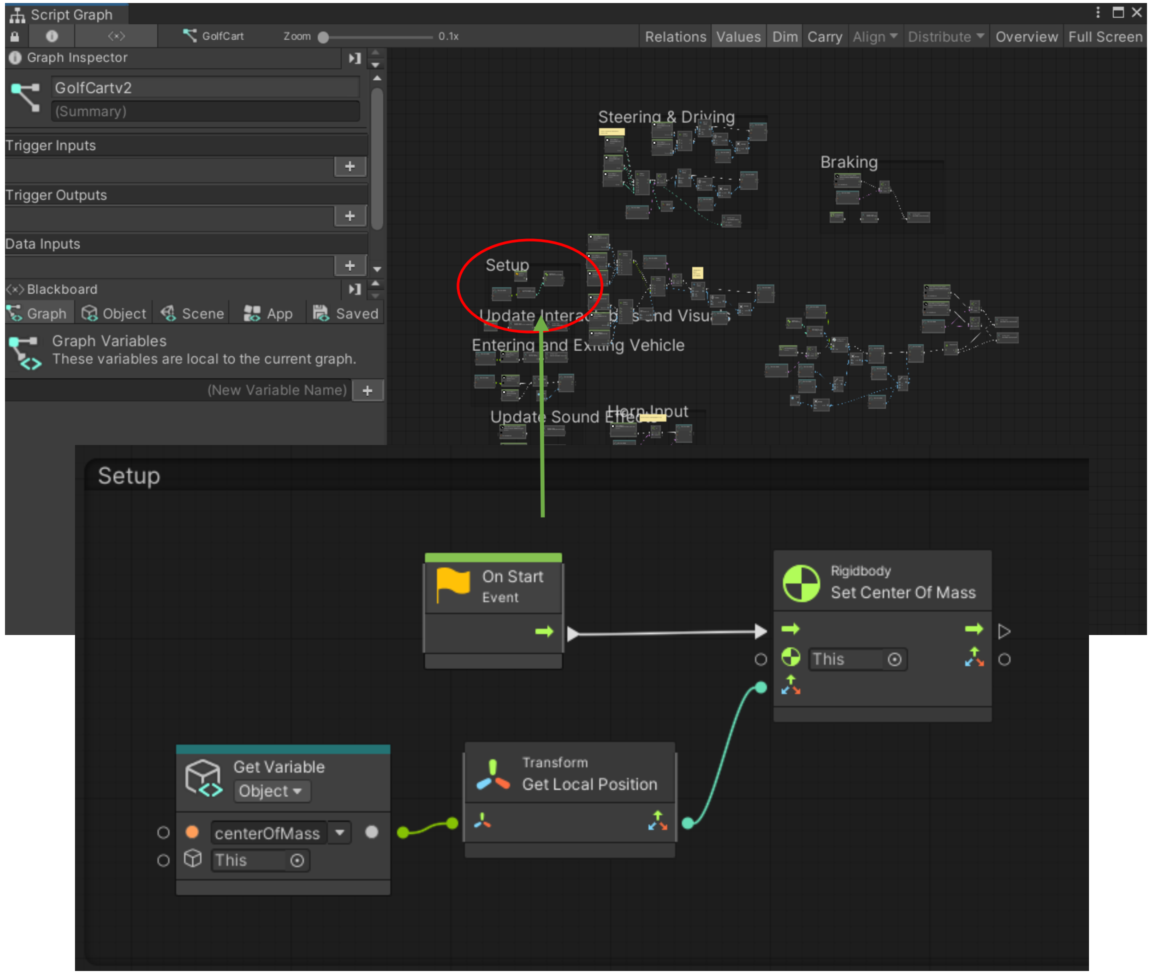
Basics

What are Visual Scripts?

In a visual script, nodes represent objects, functions, events, or even other graphs.

Nodes are connected by lines, which determines the order that the nodes are processed.

Visual scripting graphs can be quite complicated or quite simple.



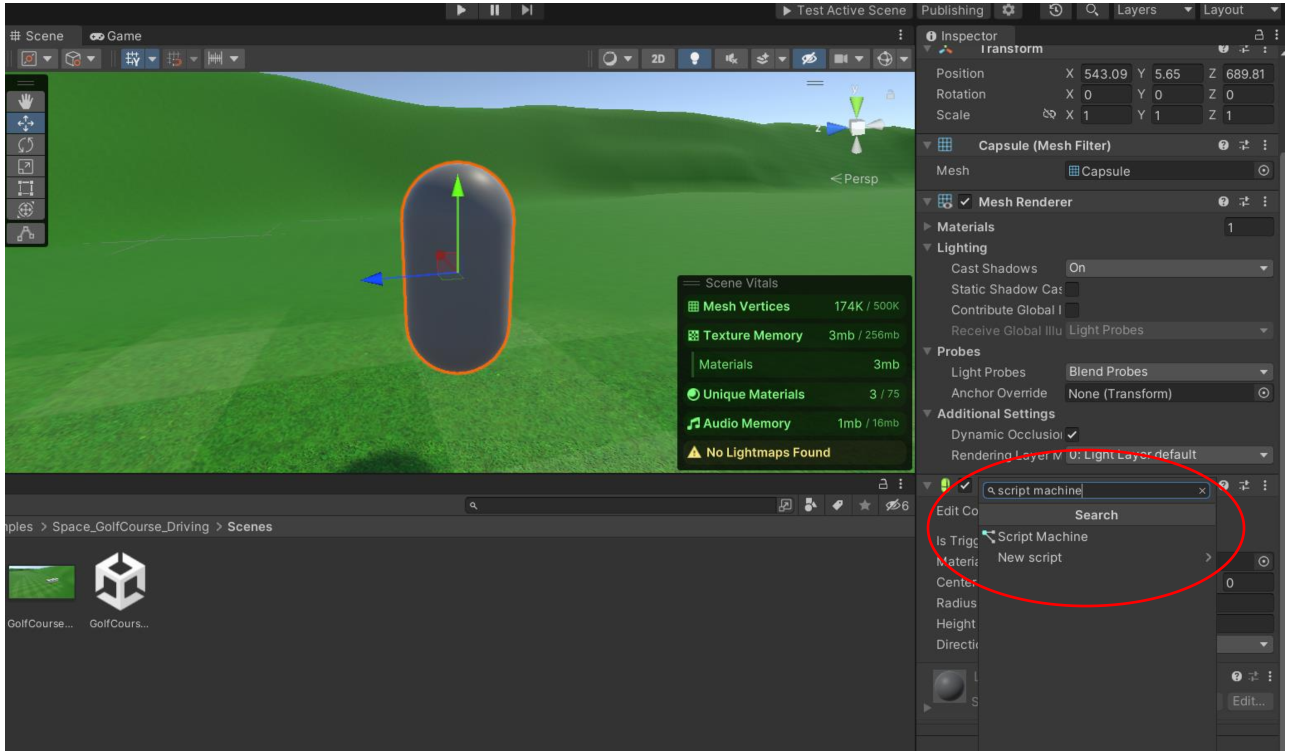
A small portion of GolfCartv2. On the start of the scene, we get the object variable centerOfMass and set the golf cart's center of mass to it.

Creating Visual Scripts

Adding a Visual Script to an Object

To give an object a new visual script, select the object in the Unity Editor.

In the Inspector, scroll down to Add Component. Search for "Script Machine" and select it.



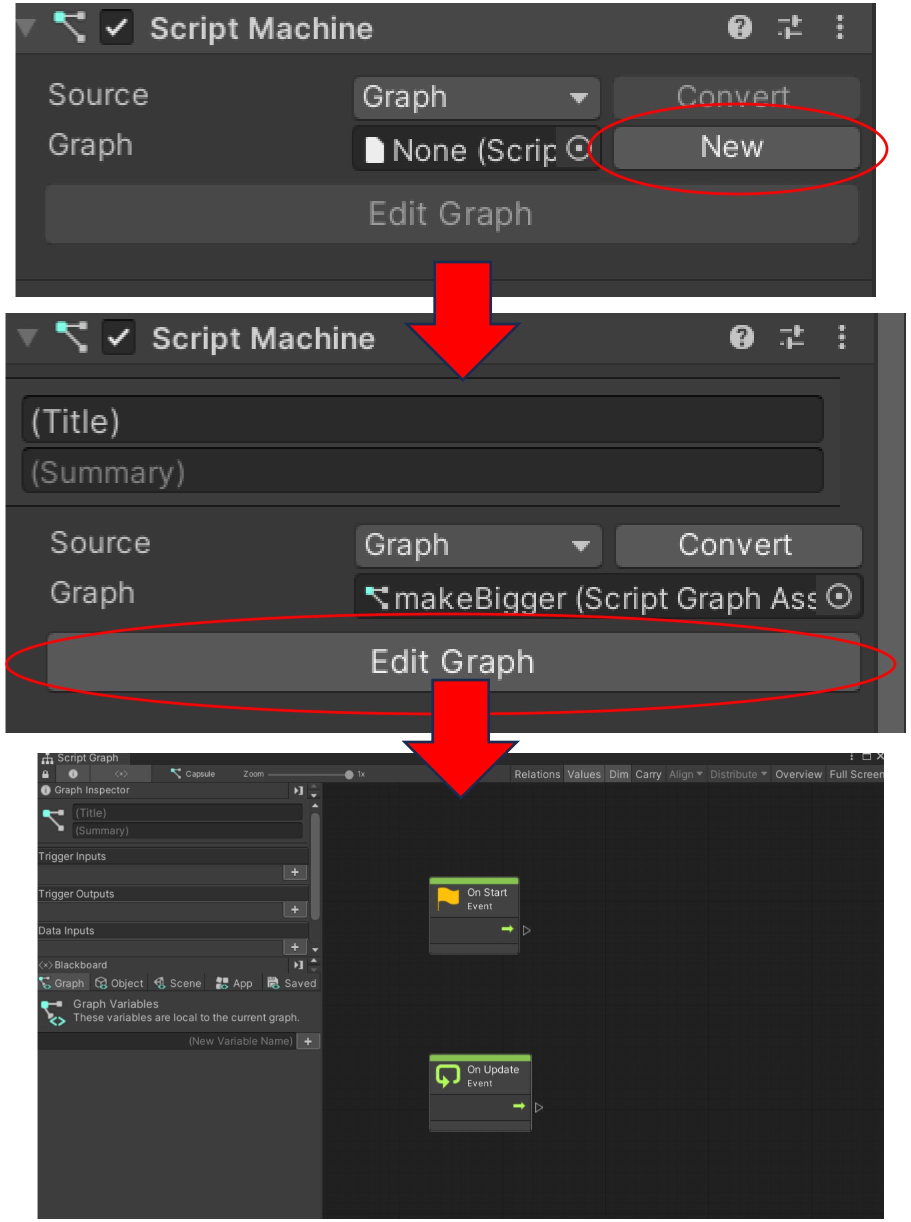
Position X 543.09 Y 5.65 Z 689.81
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Mesh Capsule
Materials 1
Cast Shadows On
Static Shadow Cas □
Contribute Global I □
Receive Global Illu Light Probes
Light Probes Blend Probes
Anchor Override None (Transform)
Dynamic Occlusio ✓
Rendering Layer M 0: Light Layer default
script machine

Creating Visual Scripts

Adding a Visual Script to an Object

In your new Script Machine component, click "New" to create a new graph. After naming and saving it, click "Edit Graph".

This is where you can see and edit the graph of your visual script.



Script Machine

Source: Graph

Graph: None (Scrip

Script Machine

(Title)

(Summary)

Source: Graph

Graph: makeBigger (Script Graph Ass

Script Graph

Graph Inspector

(Title)

(Summary)

Trigger Inputs

Trigger Outputs

Data Inputs

Blackboard

Graph Variables

These variables are local to the current graph.

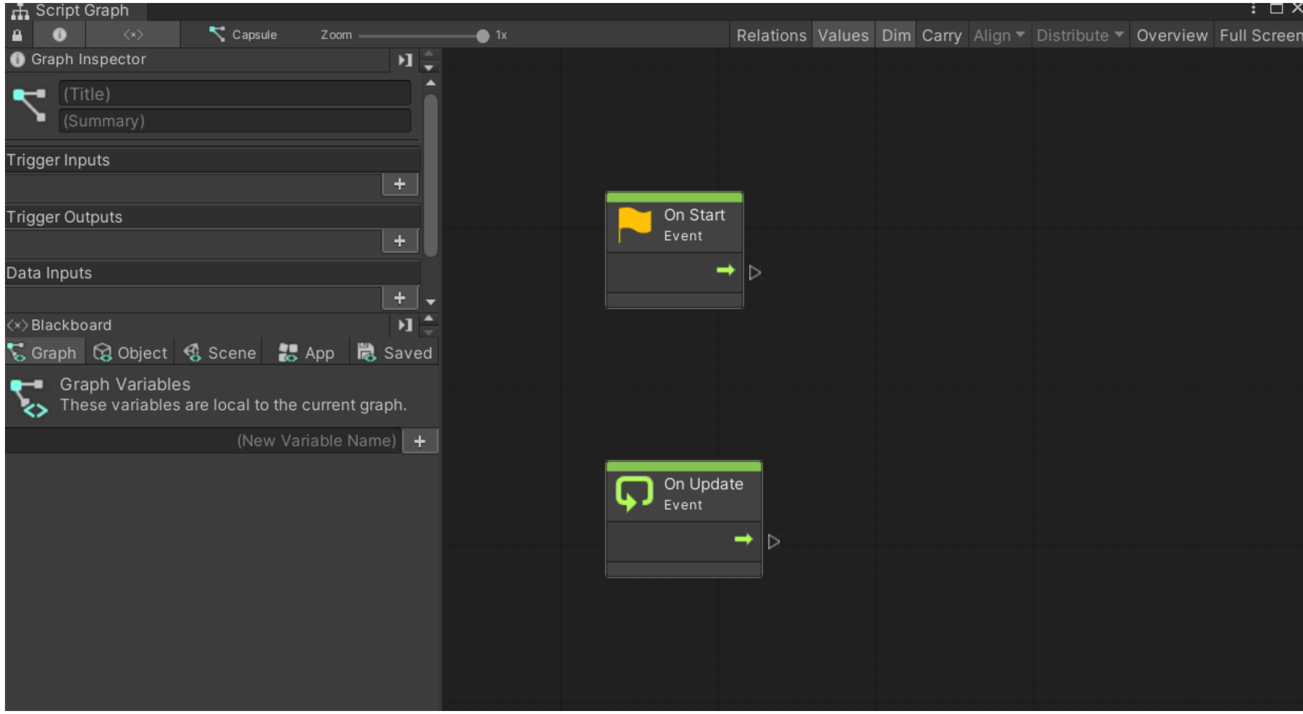
(New Variable Name)

Creating Visual Scripts

Writing your first Visual Script Graph

A new graph has two nodes. These are Event nodes, which are necessary for our script to do anything.

Different Event nodes are triggered during different events; For example, "On Start" simply triggers one time when the scene first loads. "On Update" triggers once every frame.



These variables are local to the current graph.

Additional Notes Below

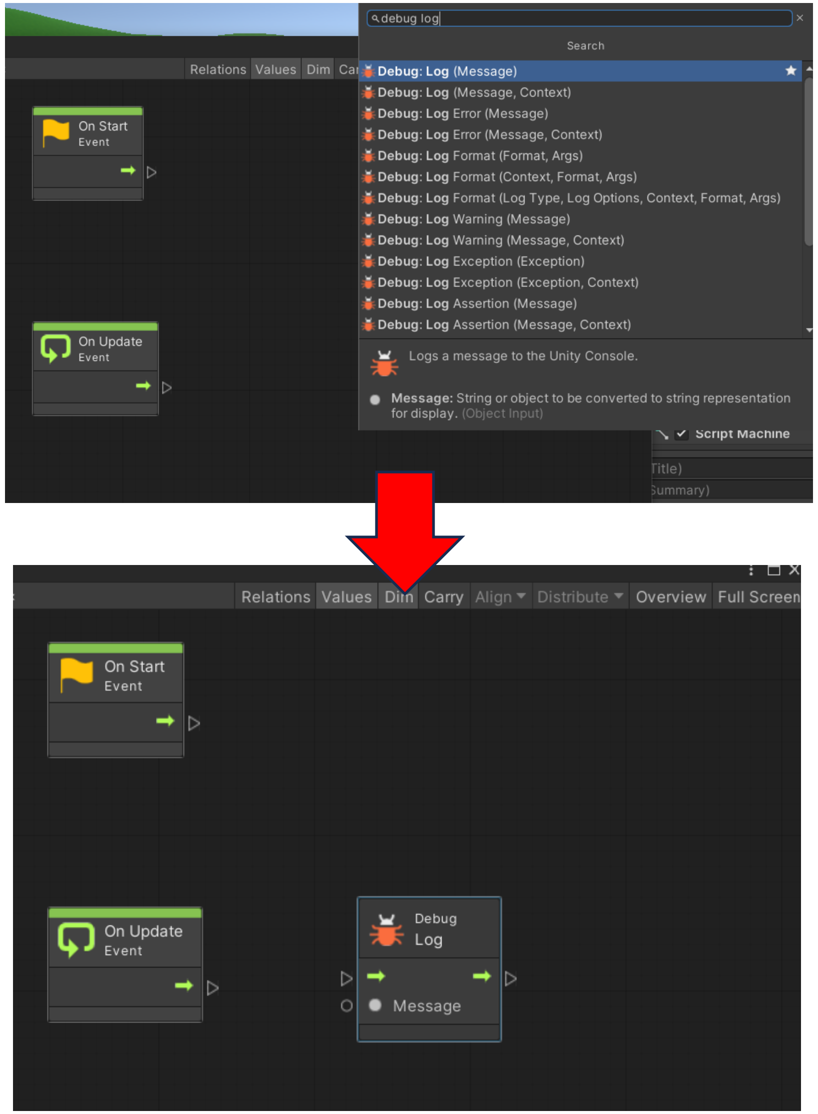
Creating Visual Scripts

Writing your first Visual Script Graph

Functions are also essential for our visual script to do anything.

One of the best functions for testing your script is the Debug Log node.

To add a node to your graph, right click anywhere in the graph and search for the desired node.


debug log
Search
Relations
Values
Dim
Ca
Debug: Log (Message)
Debug: Log (Message, Context)
Debug: Log Error (Message)
Debug: Log Error (Message, Context)
Debug: Log Format (Format, Args)
Debug: Log Format (Context, Format, Args)
Debug: Log Format (Log Type, Log Options, Context, Format, Args)
Debug: Log Warning (Message)
Debug: Log Warning (Message, Context)
Debug: Log Exception (Exception)
Debug: Log Exception (Exception, Context)
Debug: Log Assertion (Message)
Debug: Log Assertion (Message, Context)
On Start
Event
On Update
Event
Script Machine
Title)
Summary)
Relations
Values
Dim
Carry
Align
Distribute
Overview
Full Screen
On Start
Event
On Update
Event
Debug
Log
Message

Logs a message to the Unity Console.

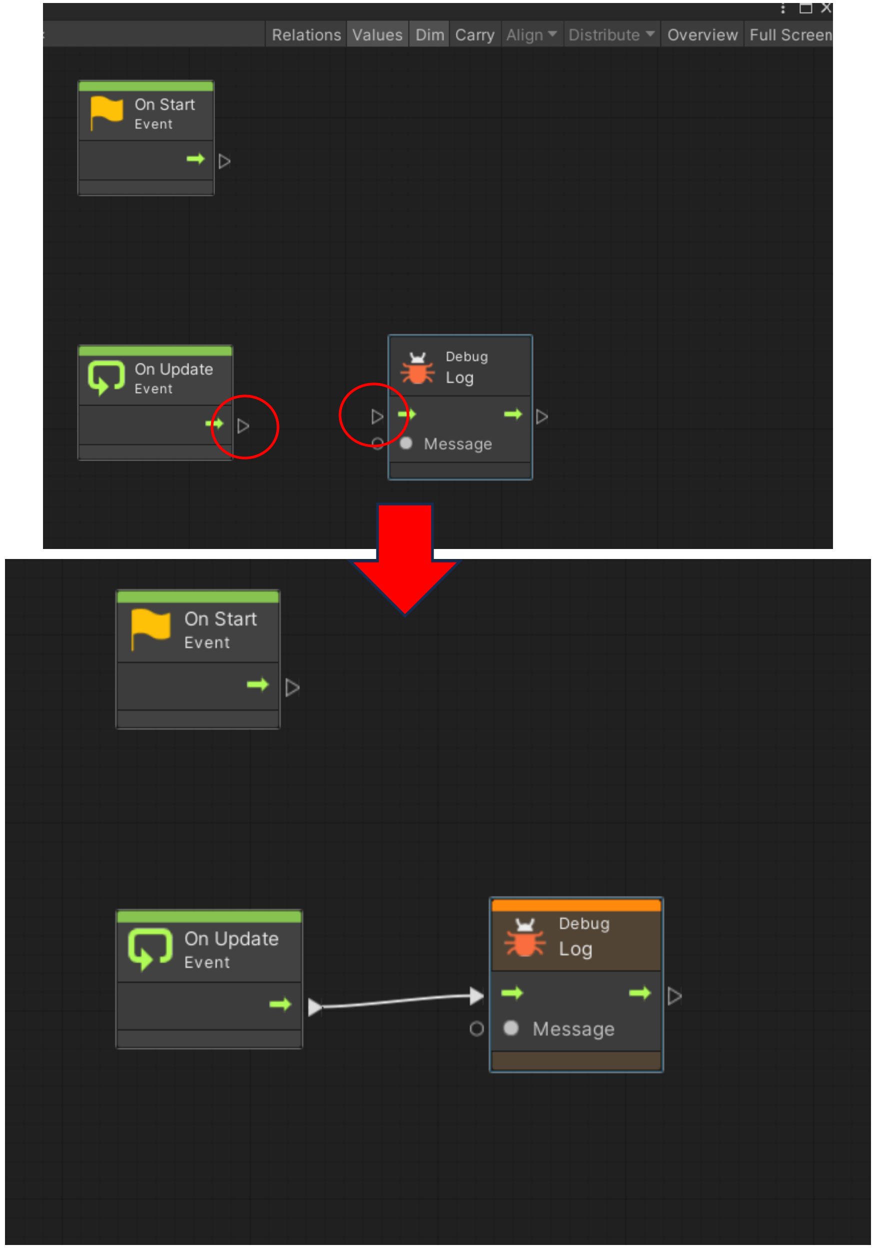
Message: String or object to be converted to string representation for display. (Object Input)

Creating Visual Scripts

Writing your first Visual Script Graph

To connect two nodes together, you need to connect one node's Exit flow to another node's Invoke flow.

Click the triangle next to the arrow pointing outward from the first node, and then click the triangle next to the arrow pointing inward to the second node.



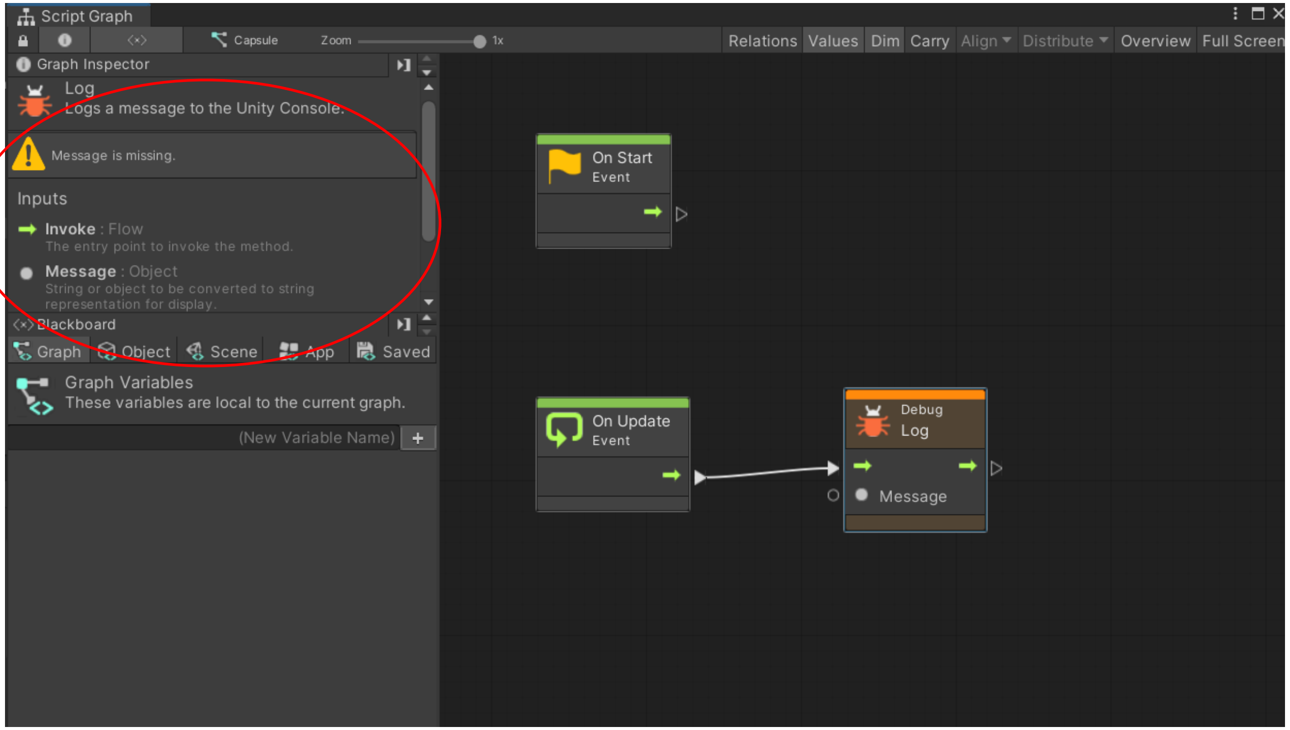
Creating Visual Scripts

Writing your first Visual Script Graph

Our Debug Log node is orange now because it is invoked but is missing a required argument.

Clicking on a node will display any of its warnings, as well as inputs and outputs.

We can see that our Debug Log node is missing a Message object, which should be a string.



Log
Logs a message to the Unity Console.

Message is missing.

Inputs

Invoke : Flow
The entry point to invoke the method.

Message : Object
String or object to be converted to string representation for display.

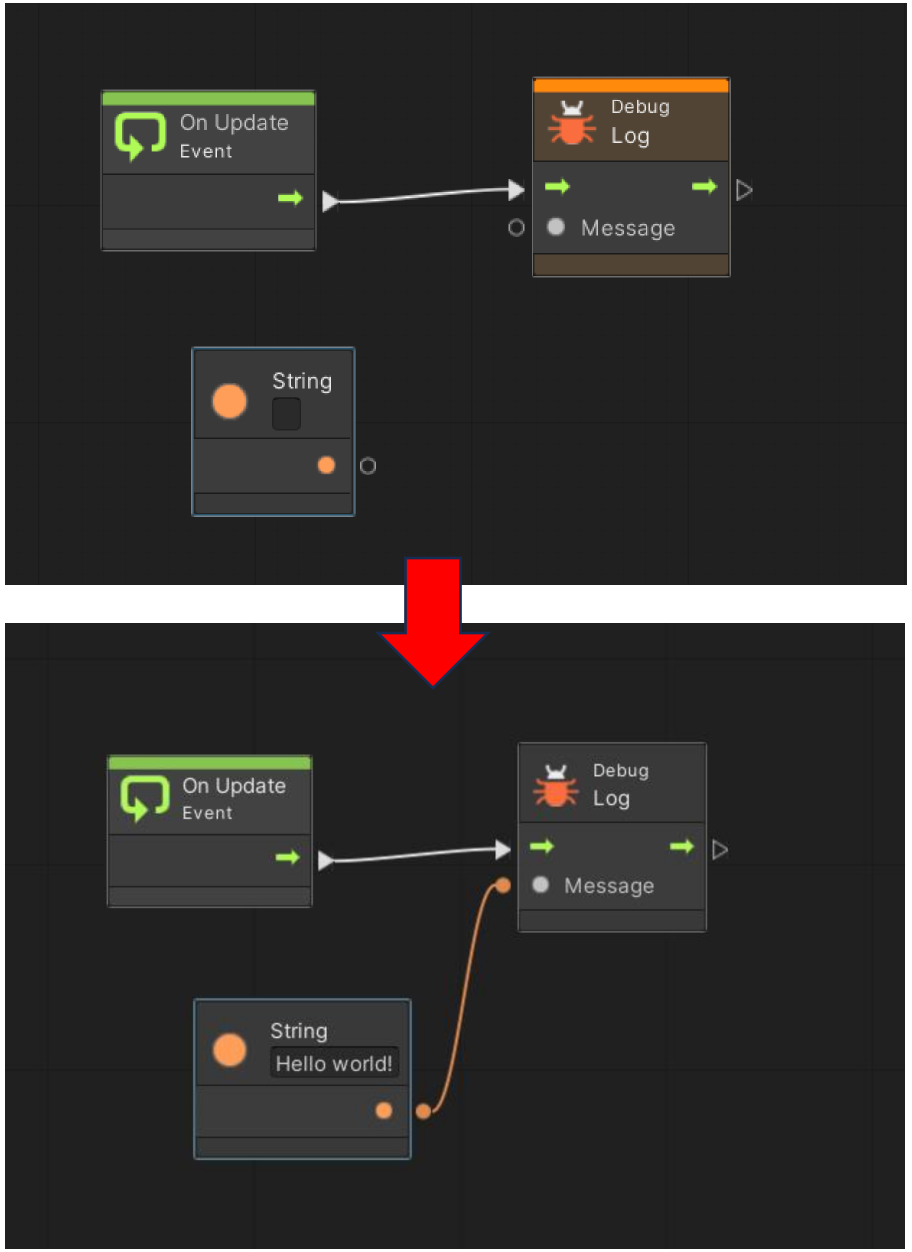
Graph Variables
These variables are local to the current graph.

Creating Visual Scripts

Writing your first Visual Script Graph

You can create a "literal" as a node. Right click and search for "String literal". You can write a string right inside the node.

Then, connect its output to the Debug Log node's input.

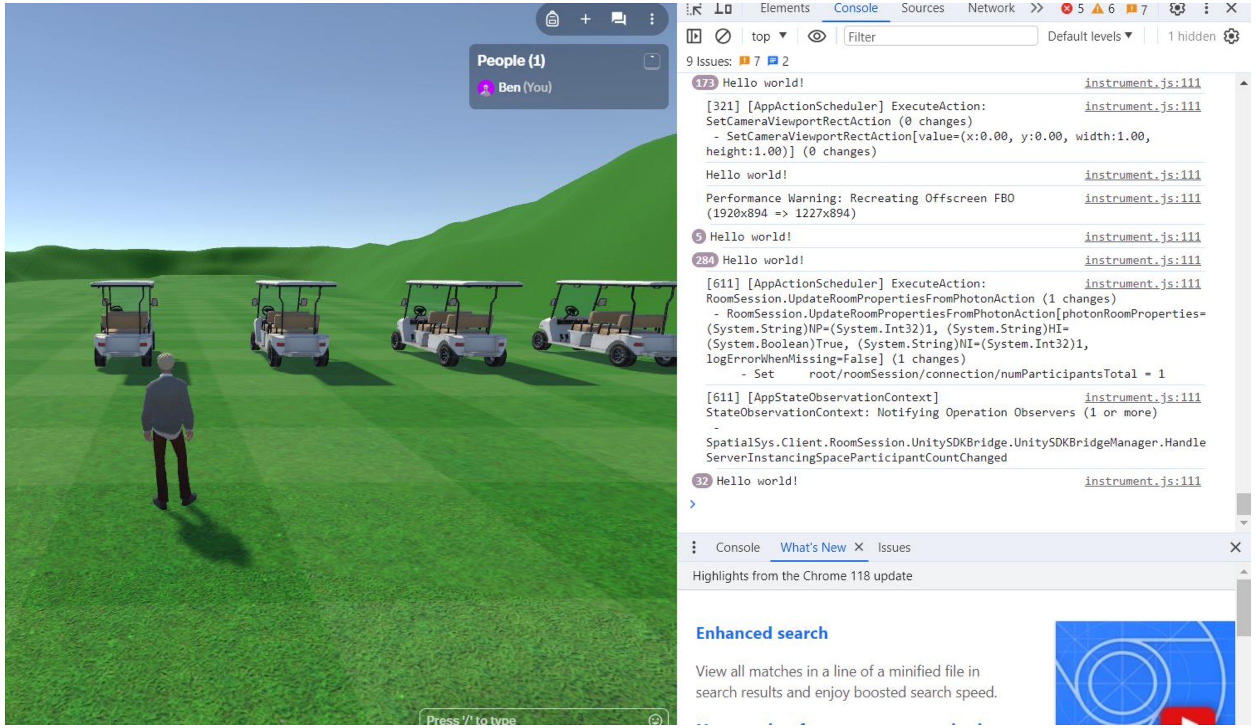


Creating Visual Scripts

Writing your first Visual Script Graph

Now that we have an event and a function, we can see our visual script in action. Test the scene that your script is a part of.

To see messages printed by the Debug Log node, you must right click the sandbox page and click "Inspect". Then, change the view to Console mode.



```
173 Hello world!                                                   instrument.js:111
   [321] [AppActionScheduler] ExecuteAction:                       instrument.js:111
   SetCameraViewportRectAction (0 changes)
    - SetCameraViewportRectAction[value=(x:0.00, y:0.00, width:1.00,
   height:1.00)] (0 changes)
   Hello world!                                                    instrument.js:111
   Performance Warning: Recreating Offscreen FBO                   instrument.js:111
   (1920x894 => 1227x894)
5 Hello world!                                                     instrument.js:111
284 Hello world!                                                   instrument.js:111
   [611] [AppActionScheduler] ExecuteAction:                       instrument.js:111
   RoomSession.UpdateRoomPropertiesFromPhotonAction (1 changes)
    - RoomSession.UpdateRoomPropertiesFromPhotonAction[photonRoomProperties=
   (System.String)NP=(System.Int32)1, (System.String)HI=
   (System.Boolean)True, (System.String)NI=(System.Int32)1,
   logErrorWhenMissing=False] (1 changes)
        - Set     root/roomSession/connection/numParticipantsTotal = 1
   [611] [AppStateObservationContext]                              instrument.js:111
   StateObservationContext: Notifying Operation Observers (1 or more)
    -
   SpatialSys.Client.RoomSession.UnitySDKBridge.UnitySDKBridgeManager.Handle
   ServerInstancingSpaceParticipantCountChanged
32 Hello world!                                                    instrument.js:111
>
```

Highlights from the Chrome 118 update

Enhanced search

View all matches in a line of a minified file in search results and enjoy boosted search speed.

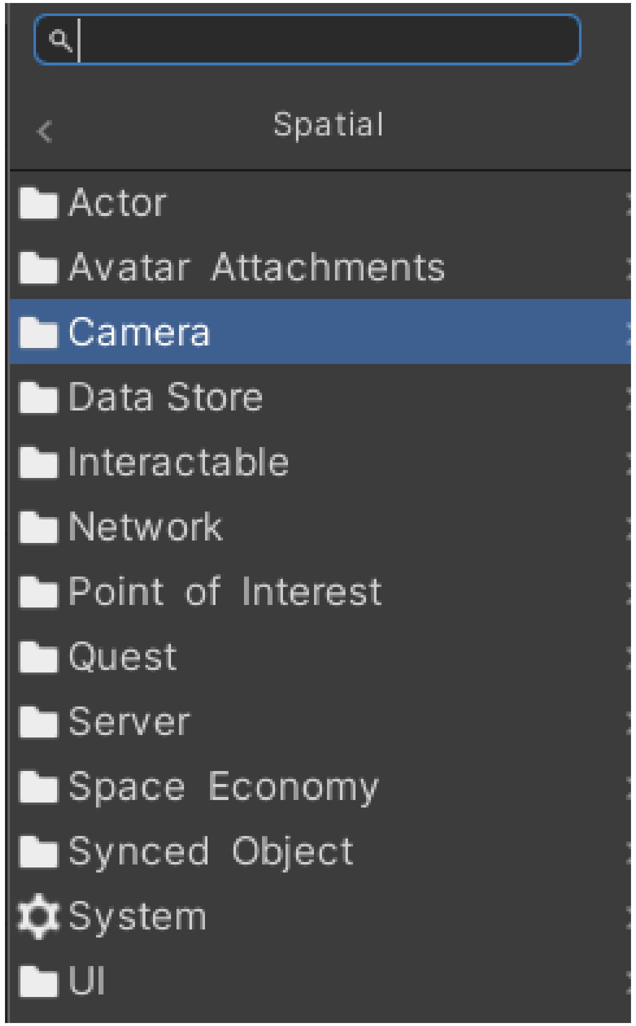
There are tons of lines saying "Hello World!" -- our script is printing this message every frame.

Creating Visual Scripts

Spatial Nodes

Now we have seen an example of Event, Function, and Object nodes. It's also important to note that there are Spatial nodes.

These are nodes that are added to visual scripting by the Spatial SDK and are therefore not native to Unity like the other nodes.



Additional Notes Below

Creating Visual Scripts

Spatial Nodes

Many of these Spatial nodes are incredibly useful, but they do not all show up when you just search "Spatial". Try searching for these:

Spatial Interactable

Spatial Input

Spatial Data Store

Spatial Sync

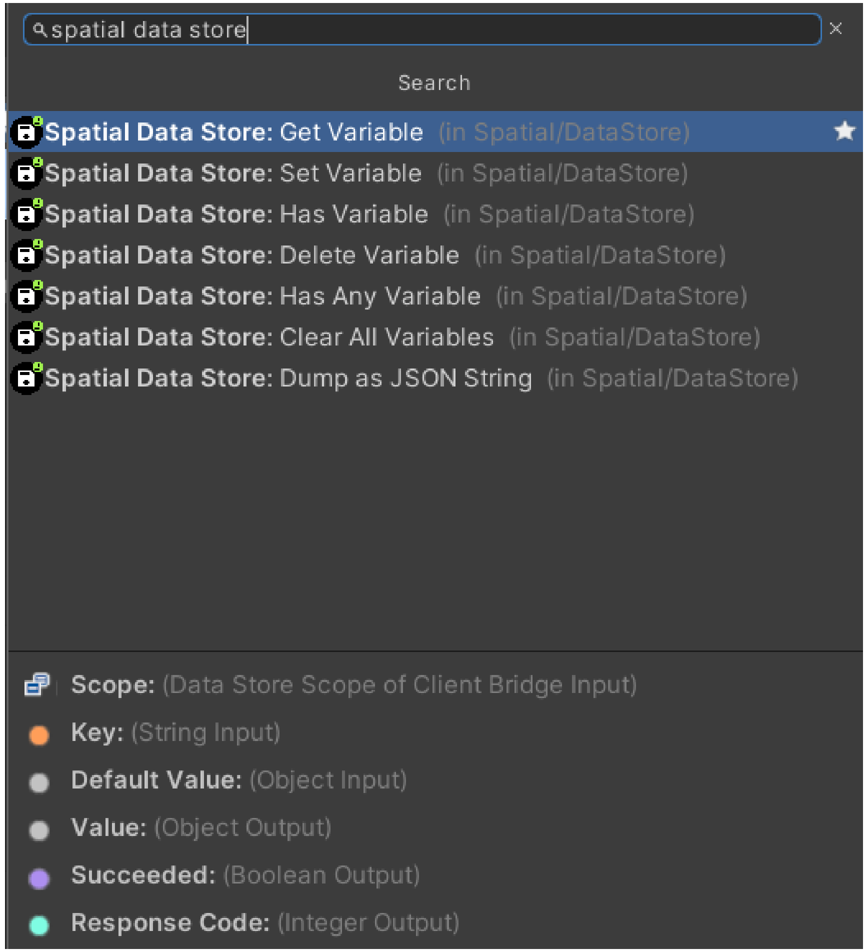
Spatial Synced Object

Spatial SFX

Spatial Quest

Spatial System

Spatial UI



spatial data store

Search

- Spatial Data Store: Get Variable (in Spatial/DataStore)
- Spatial Data Store: Set Variable (in Spatial/DataStore)
- Spatial Data Store: Has Variable (in Spatial/DataStore)
- Spatial Data Store: Delete Variable (in Spatial/DataStore)
- Spatial Data Store: Has Any Variable (in Spatial/DataStore)
- Spatial Data Store: Clear All Variables (in Spatial/DataStore)
- Spatial Data Store: Dump as JSON String (in Spatial/DataStore)

Scope: (Data Store Scope of Client Bridge Input)

● **Key:** (String Input)

● **Default Value:** (Object Input)

● **Value:** (Object Output)

● **Succeeded:** (Boolean Output)

● **Response Code:** (Integer Output)

Spatial Data Store has some incredibly useful nodes which we will look at shortly.

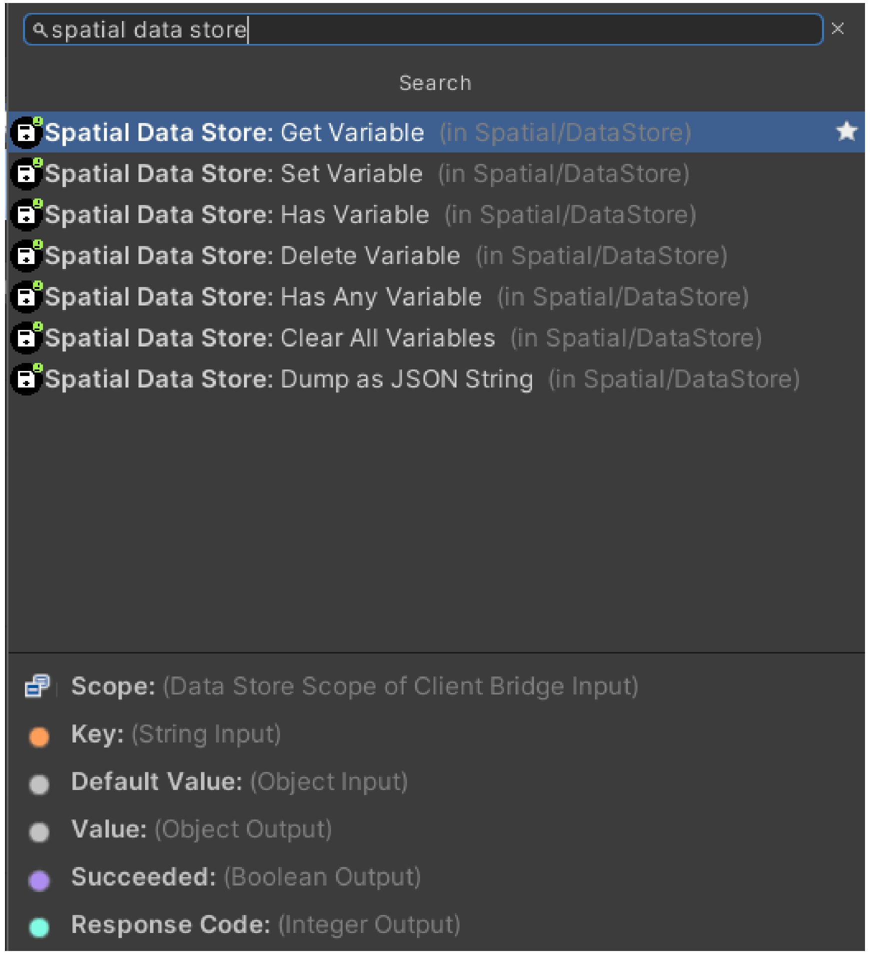
Creating Visual Scripts

Spatial Data Store

The category of Spatial nodes called Spatial Data Store can be incredibly useful.

Using these nodes, you can make global variables in your Space. Meaning, different objects in the Space can access and change these variables.

Normally, Unity has Scene Variables that serve the same purpose, but for certain cases* in Spatial these Scene Variables are unusable.



spatial data store

Search

Spatial Data Store: Get Variable (in Spatial/DataStore)
Spatial Data Store: Set Variable (in Spatial/DataStore)
Spatial Data Store: Has Variable (in Spatial/DataStore)
Spatial Data Store: Delete Variable (in Spatial/DataStore)
Spatial Data Store: Has Any Variable (in Spatial/DataStore)
Spatial Data Store: Clear All Variables (in Spatial/DataStore)
Spatial Data Store: Dump as JSON String (in Spatial/DataStore)

Scope: (Data Store Scope of Client Bridge Input)
● **Key:** (String Input)
● **Default Value:** (Object Input)
● **Value:** (Object Output)
● **Succeeded:** (Boolean Output)
● **Response Code:** (Integer Output)

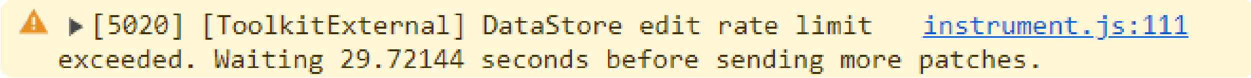
(**Important!**) Additional Notes Below

Creating Visual Scripts

Spatial Data Store

The main limitations of Spatial Data Store are its storage size and how often it can be updated.

At the moment, it can only hold 1 megabyte, and it can only be updated 10 times per minute.



```
⚠ ▶ [5020] [ToolkitExternal] DataStore edit rate limit   instrument.js:111
  exceeded. Waiting 29.72144 seconds before sending more patches.
```

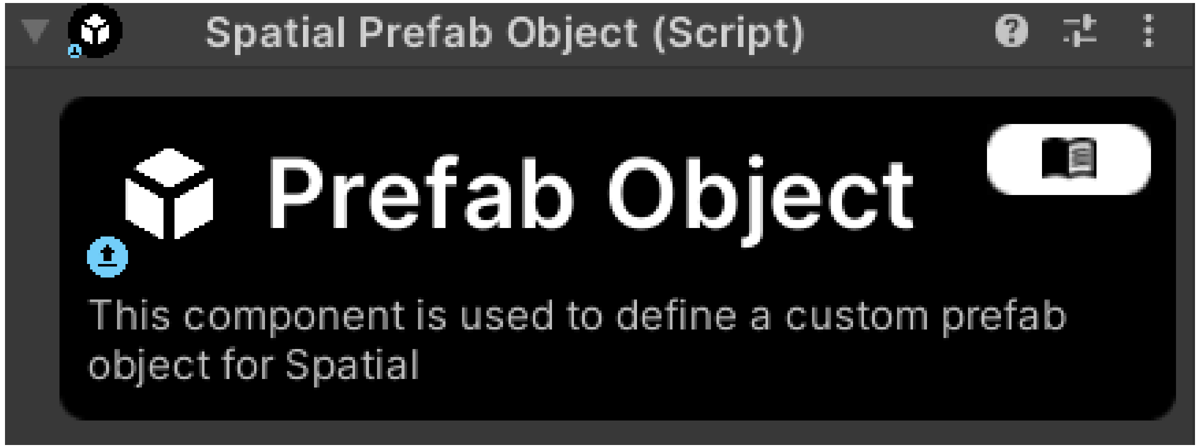
The warning that occurs when you exceed the Data Store edit rate limit

Prefab Objects

What are Spatial Prefab Objects?

Spatial offers a different package type other than a Space or a World.

Spatial Prefab Objects (not to be confused with Unity Prefabs) are a type of object that should be familiar to anyone who has used Spatial before.



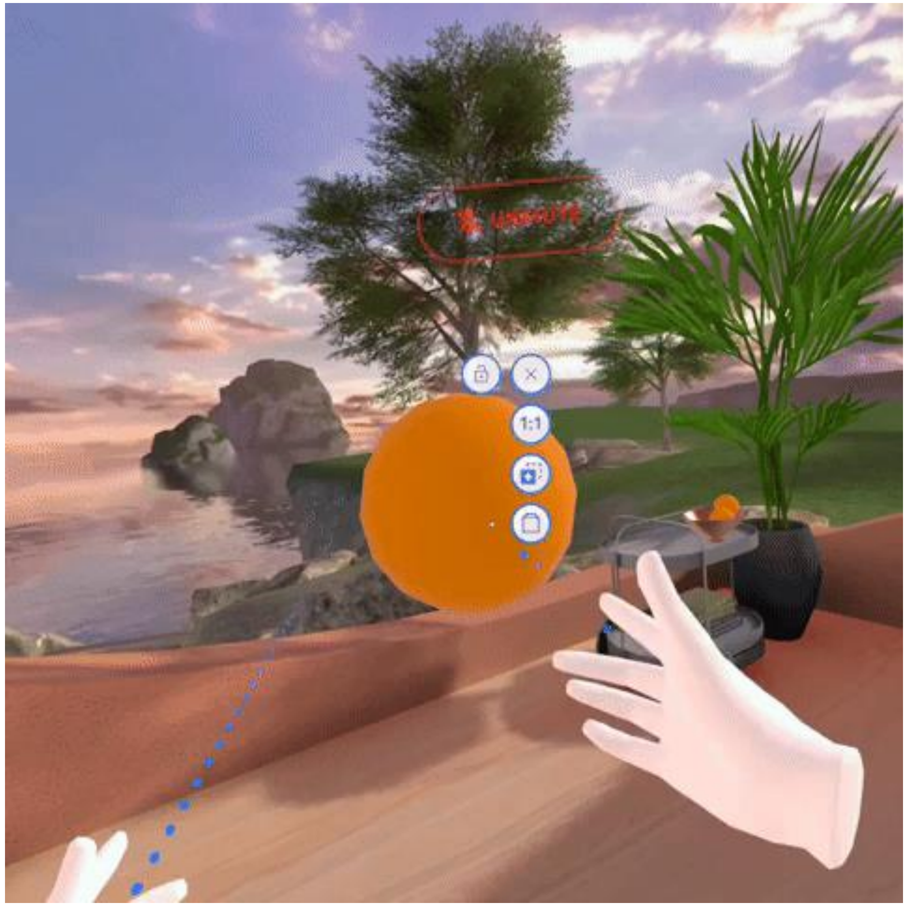
This component is used to define a custom prefab object for Spatial

Additional Notes Below

Prefab Objects

What are Spatial Prefab Objects?

If you have used Spatial before, several good examples of Prefab objects can be found right in your Home Space.



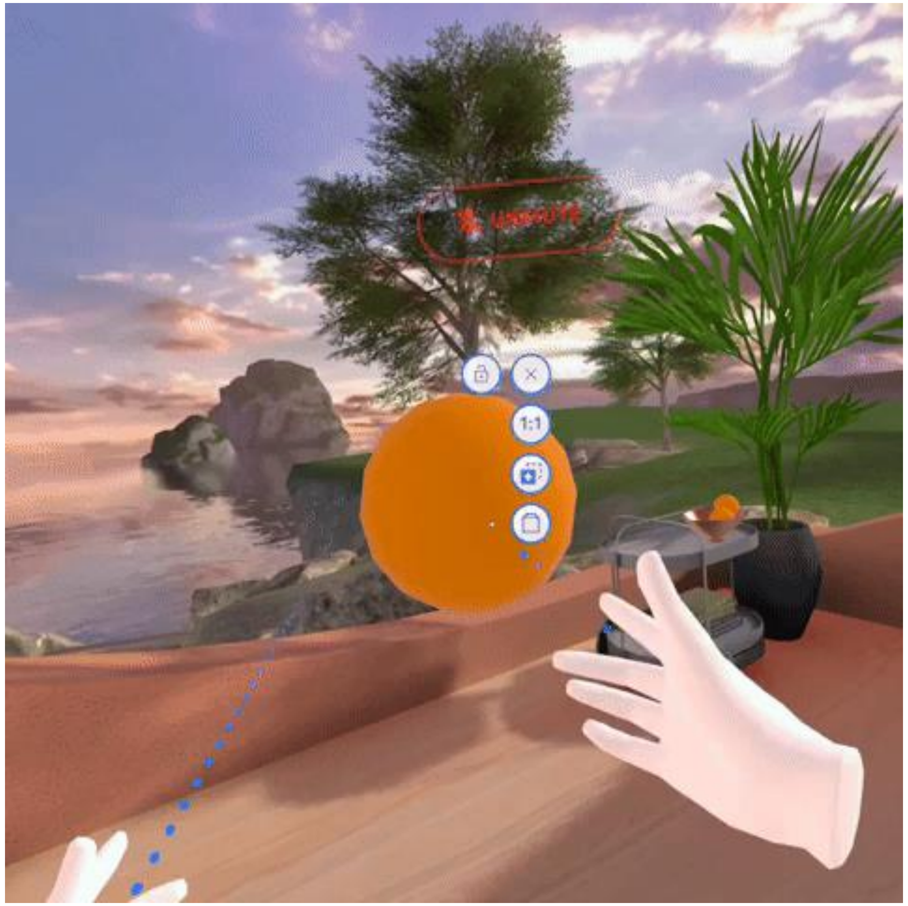
The Spatial Prefab Object of an orange in your Home Space

Prefab Objects

What are Spatial Prefab Objects?

In short, a Spatial Prefab Object uses a script of Spatial's design to have a certain interactability for the user when it is deployed in a Space.

All Prefab Objects have the same UI, interactability, and controls.



The Spatial Prefab Object that represents in orange in your Home Space

Prefab Objects

What are Spatial Prefab Objects?

Spatial Prefab Objects allow you to:

-Move them

-Resize them

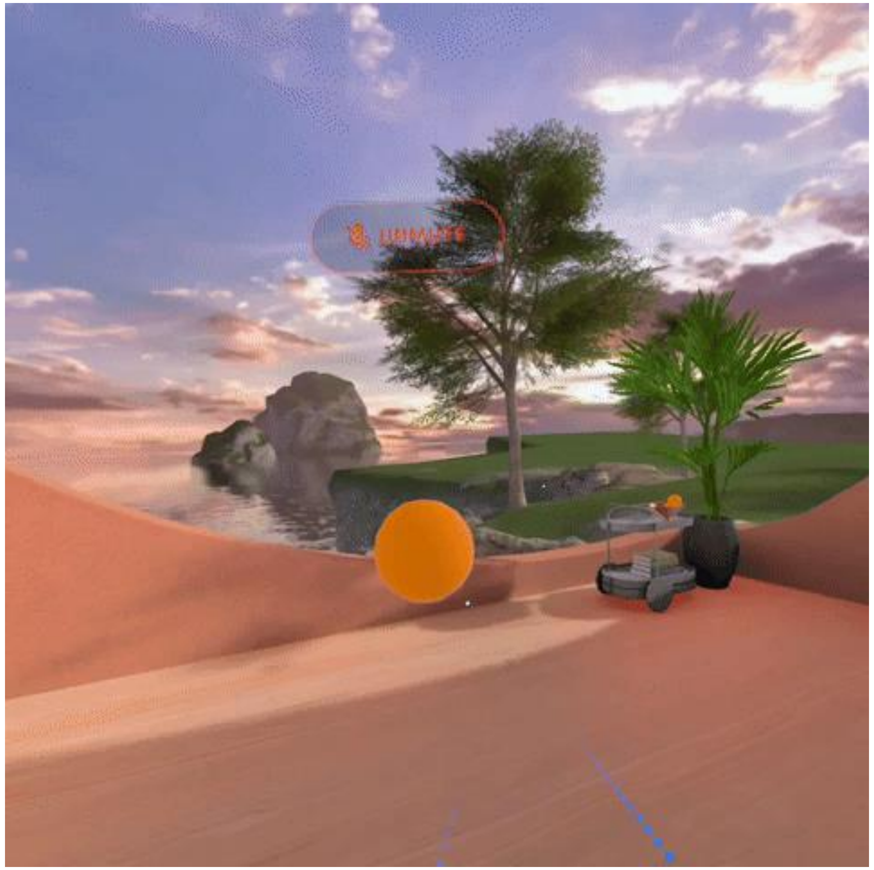
-Rotate them

-Hold them

-Duplicate and delete them

-Pin them in place

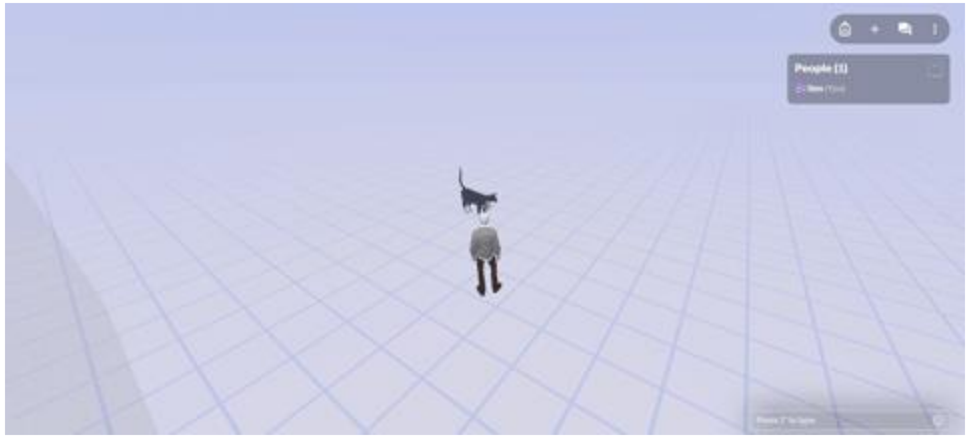
-Script them



Duplicating and resizing the orange

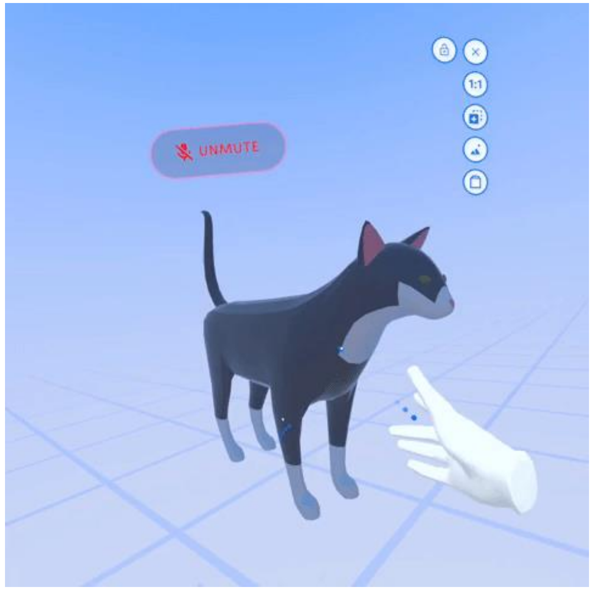
Prefab Objects

What are Spatial Prefab Objects?



Interacting with a cat Prefab on browser

Note that interacting with Spatial Prefab Objects is naturally quite different between the browser version and the Oculus version.



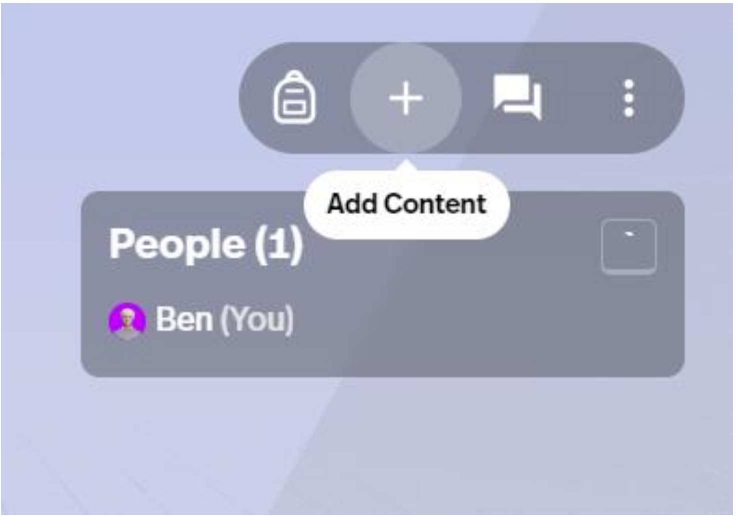
Interacting with a cat Prefab on Oculus

Prefab Objects

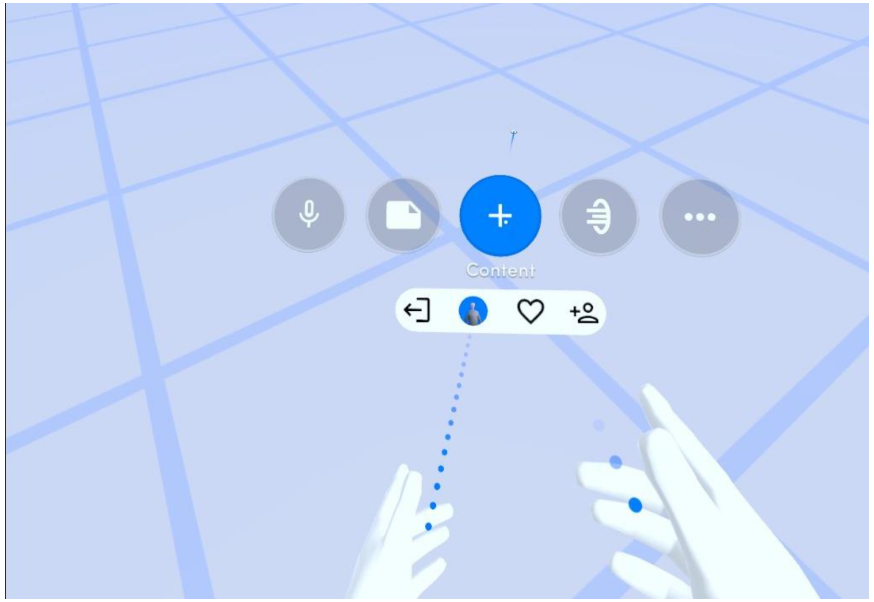
Adding Prefab Objects to your Space

To add a Spatial Prefab Object to your Space, first enter the Space in either browser or Oculus.

You can then open your Add Content menu (top right UI + sign on browser, bottom UI + sign on Oculus).



Browser UI



Oculus UI

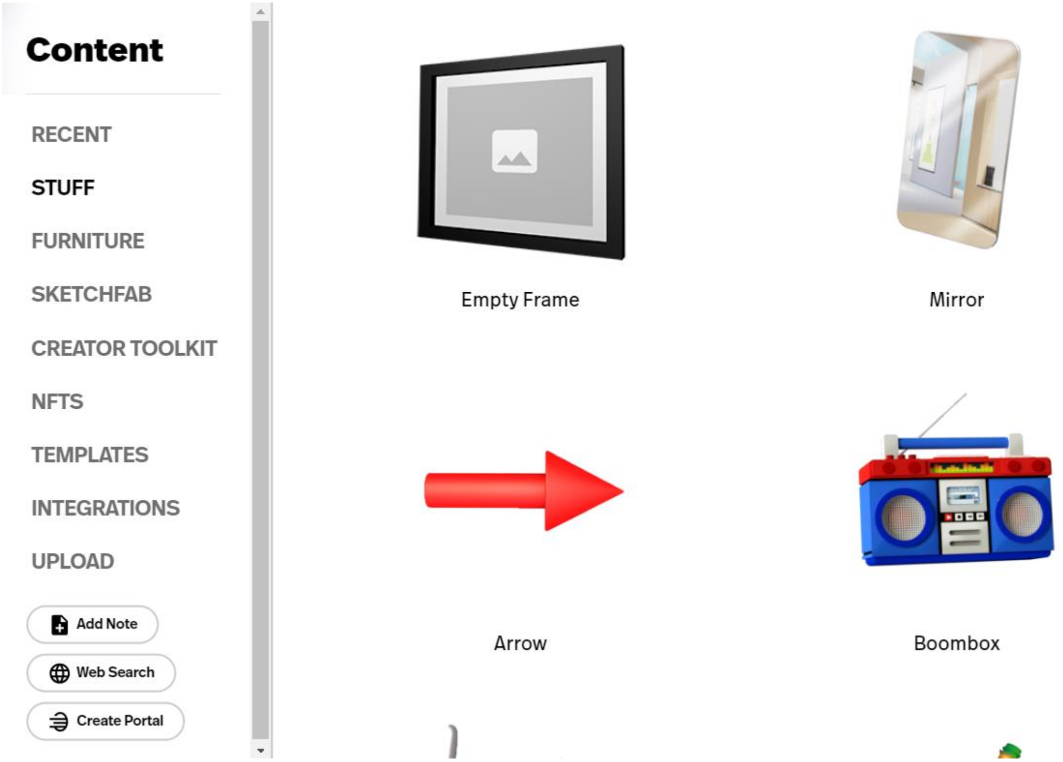
Prefab Objects

Adding Prefab Objects to your Space

From the Add Content menu, you can add Prefab Objects from various different sources.

For example, Spatial has some Prefab Objects for you under "Stuff" and "Furniture".

You can also pull Prefab Objects directly from Sketchfab.

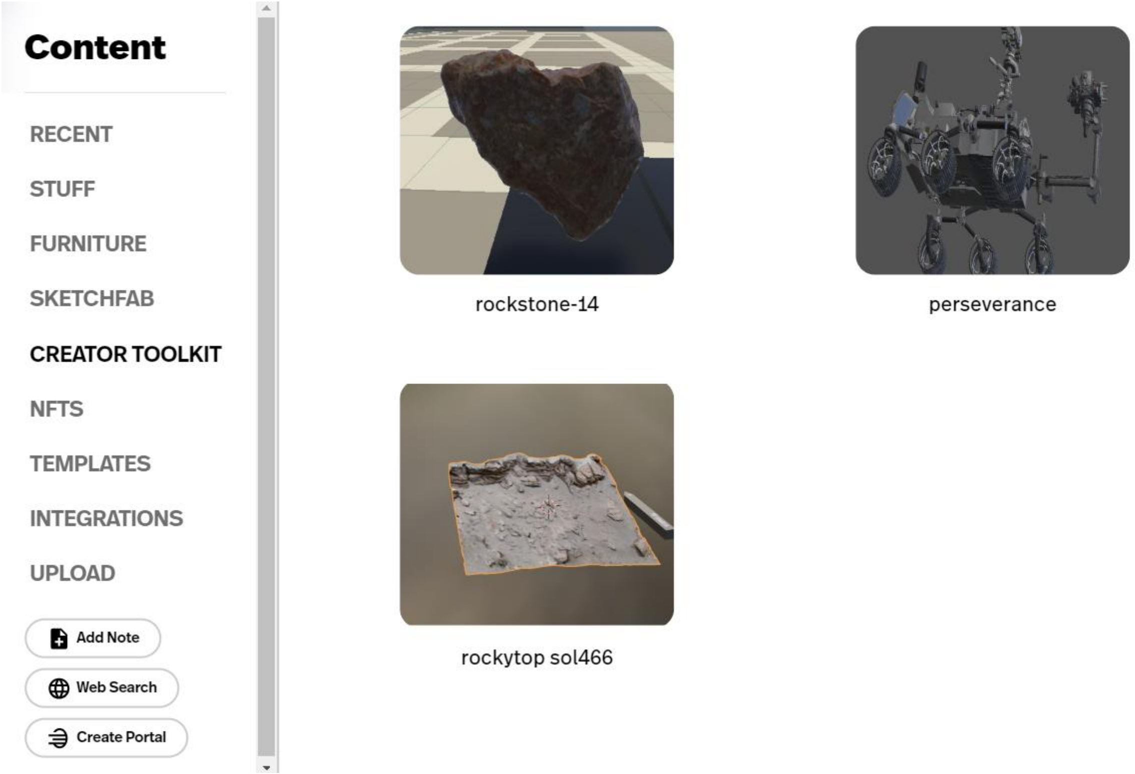


Prefab Objects

Adding Prefab Objects to your Space

Under "Creator Toolkit", you can find custom Prefab Objects that you have published.

We will cover that process next.



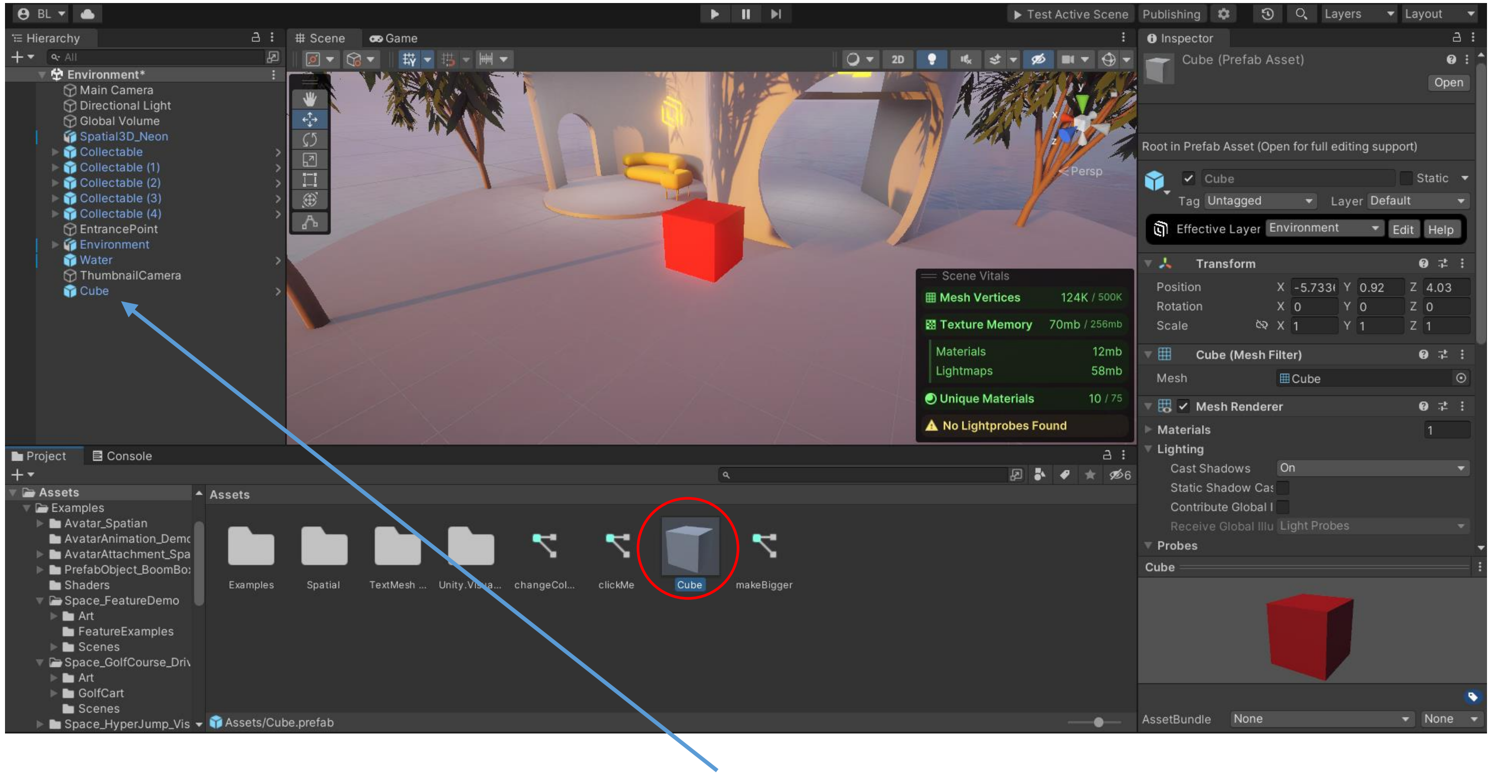
Prefab Objects

Creating custom Prefab Objects

Creating a custom Prefab Object is as easy as adding a component to your desired object in Unity Editor.

Open any Unity project set up for Spatial and create your desired object as a Unity Prefab.

To do this simply drag your object somewhere into the Assets directory at the bottom.



Scene Vitals

Mesh Vertices 124K / 500K
Texture Memory 70mb / 256mb
Materials 12mb
Lightmaps 58mb
Unique Materials 10 / 75
No Lightprobes Found

Assets/Cube.prefab

Cube (Prefab Asset)

Root in Prefab Asset (Open for full editing support)

☑ Cube □ Static
Tag Untagged Layer Default
Effective Layer Environment Edit Help

Transform
Position X -5.733(Y 0.92 Z 4.03
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Cube (Mesh Filter)
Mesh Cube

☑ Mesh Renderer
Materials 1
Lighting
Cast Shadows On
Static Shadow Cas □
Contribute Global I □
Receive Global Illu Light Probes

AssetBundle None None

Unity Prefabs are blue in the Hierarchy.

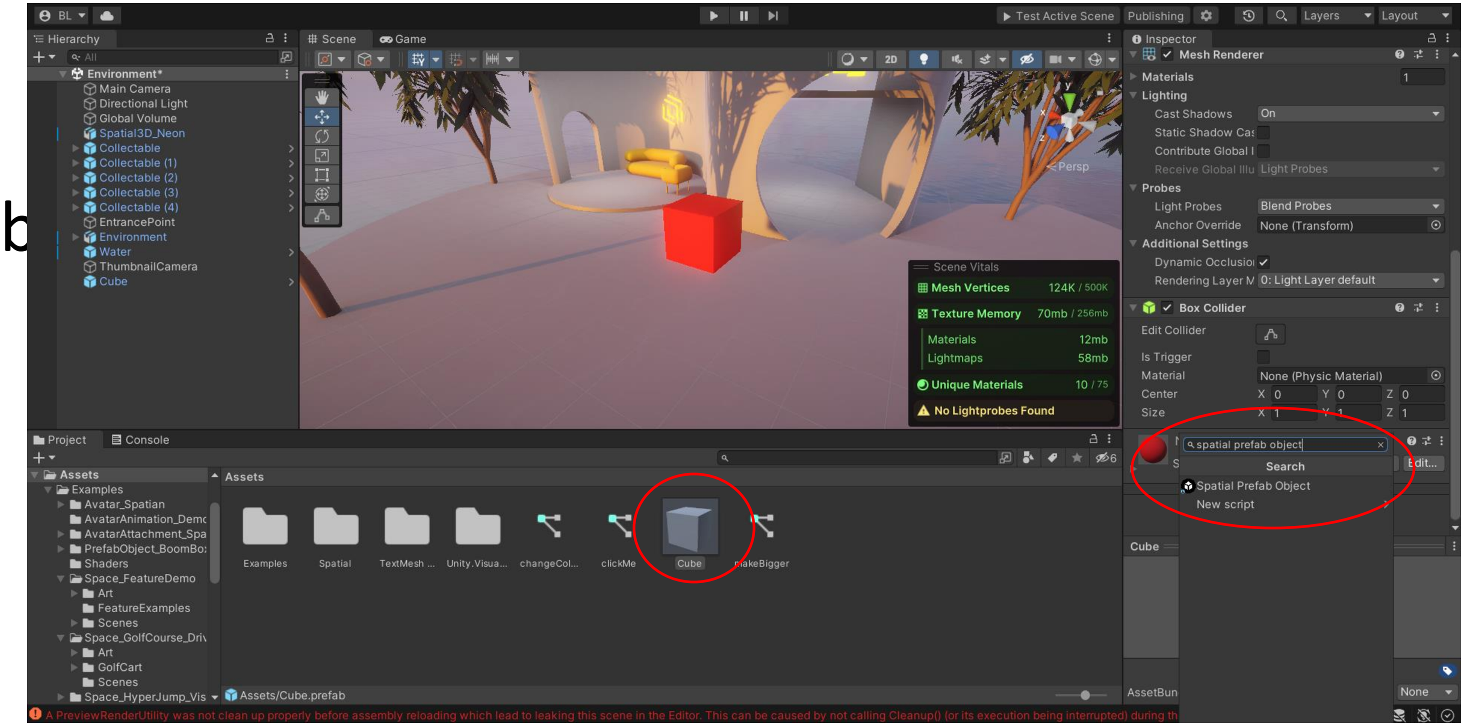
Prefab Objects

Creating custom Prefab Objects

Now, add a new component to your Unity Prefab.

Make sure to select the Prefab parent in your Assets directory (not an instance of the Prefab in your scene).

Search for "Spatial Prefab Object" and select it.



A PreviewRenderUtility was not clean up properly before assembly reloading which lead to leaking this scene in the Editor. This can be caused by not calling Cleanup() (or its execution being interrupted) during th

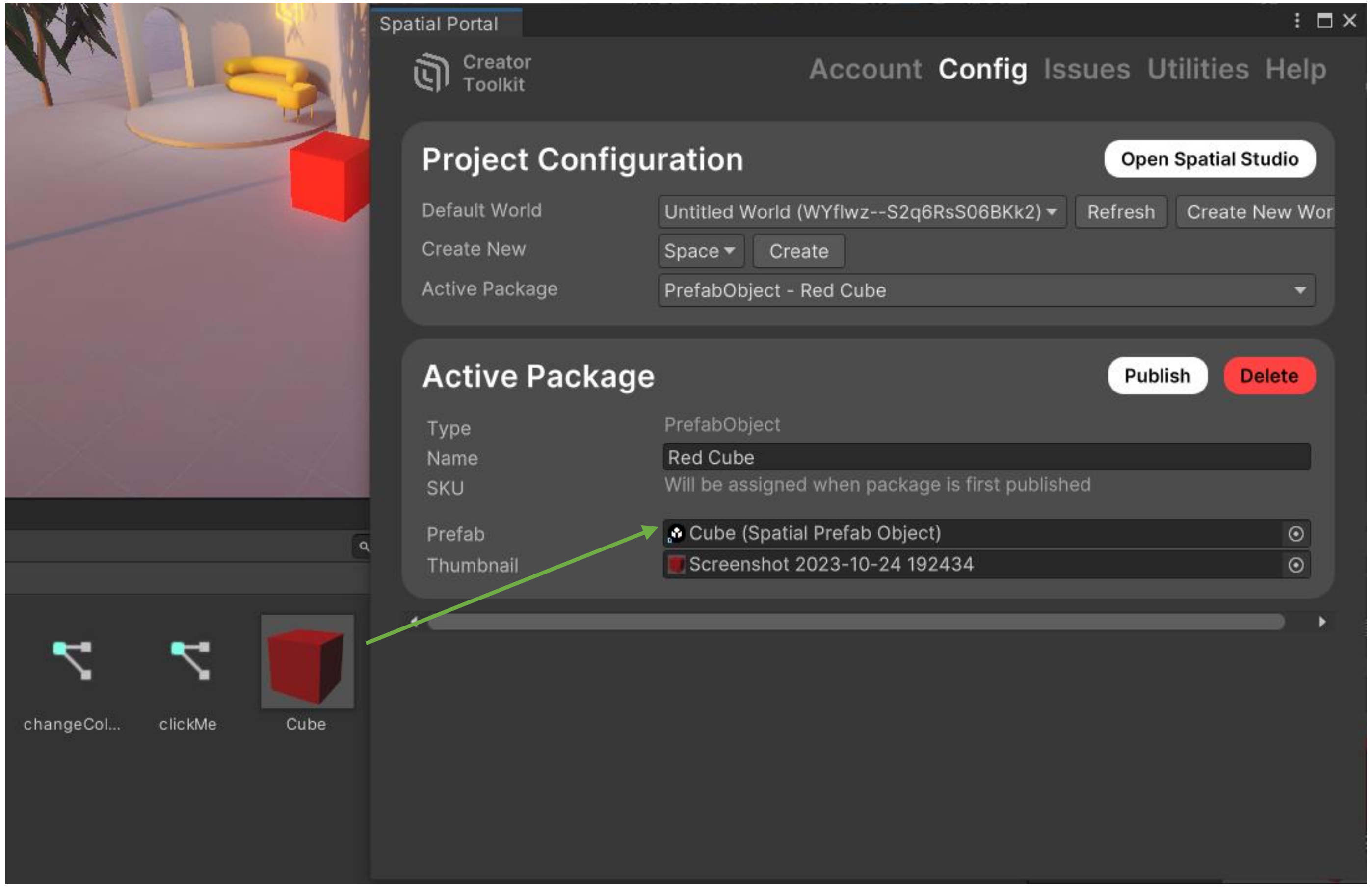
Prefab Objects

Creating custom Prefab Objects

Drag the Spatial Prefab Object from your Assets directory to the Prefab input.

Name your Prefab Object accordingly, and create a (512x512) thumbnail for it.

Then, click "Publish".



Spatial Portal

Creator Toolkit

Account Config Issues Utilities Help

Project Configuration

Open Spatial Studio

Default World: Untitled World (WYflwz--S2q6RsS06BKk2) Refresh Create New Wor

Create New: Space Create

Active Package: PrefabObject - Red Cube

Active Package

Publish Delete

Type: PrefabObject

Name: Red Cube

SKU: Will be assigned when package is first published

Prefab: Cube (Spatial Prefab Object)

Thumbnail: Screenshot 2023-10-24 192434

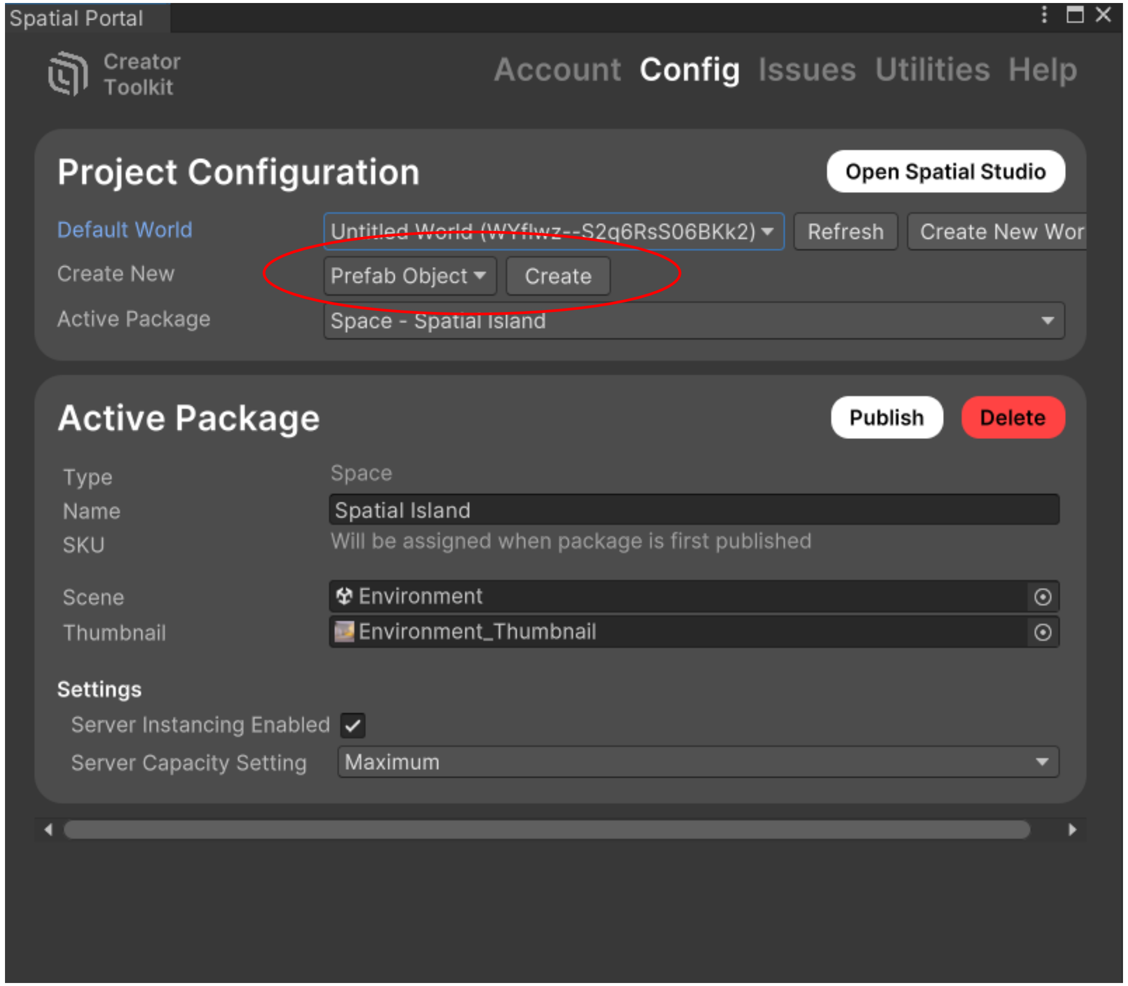
Prefab Objects

Creating custom Prefab Objects

The last step is to publish your Prefab Object package to Spatial.

In the top menu, click "Publishing". Select any World.

Next to "Create New", select "Prefab Object" and click "Create".



Project Configuration

Default World: Untitled World (WYflwz--S2q6RsS06BKk2)

Create New: Prefab Object

Active Package: Space - Spatial Island

Active Package

Type: Space

Name: Spatial Island

SKU: Will be assigned when package is first published

Scene: Environment

Thumbnail: Environment_Thumbnail

Settings

Server Instancing Enabled ✓

Server Capacity Setting: Maximum

Prefab Objects

Getting your Prefab Objects in a Space

Go ahead and delete any instances of the Spatial Prefab Object in your scene.

When you bounce your scene to Spatial, any instances of Prefab Objects already in the scene will not instantiate as Prefab Objects.

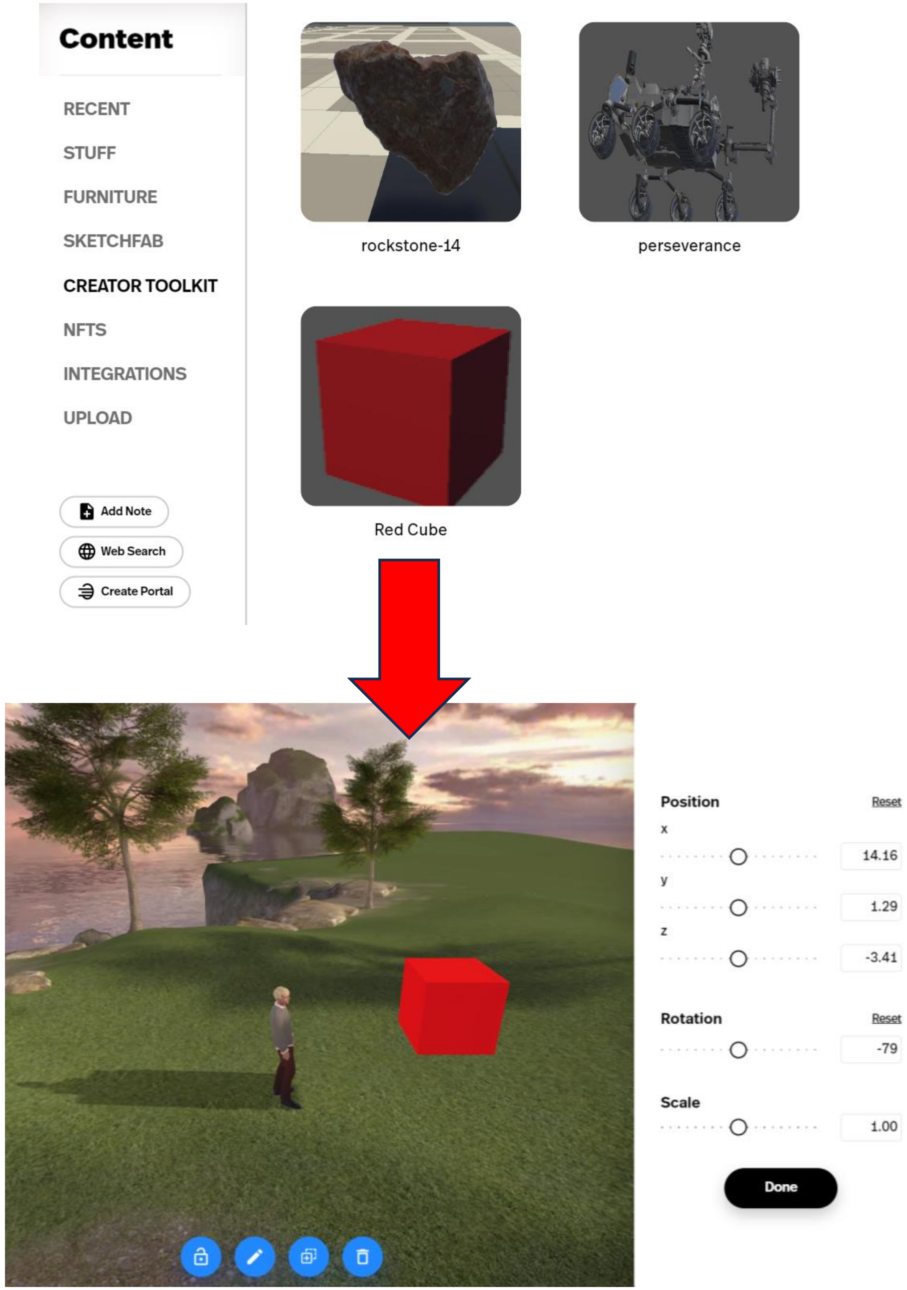
A Prefab Object that you naively put in your scene before publishing will not have any of the functionality of a Prefab Object.

Prefab Objects

Getting your Prefab Objects in a Space

To see your custom Prefab Object in a Space, you have two options.

1: For any pre-existing Space you can add content to, you can add your custom Prefab from the Add Content menu. Look for it in the "Creator Toolkit" tab.



Position
x 14.16
y 1.29
z -3.41

Rotation -79

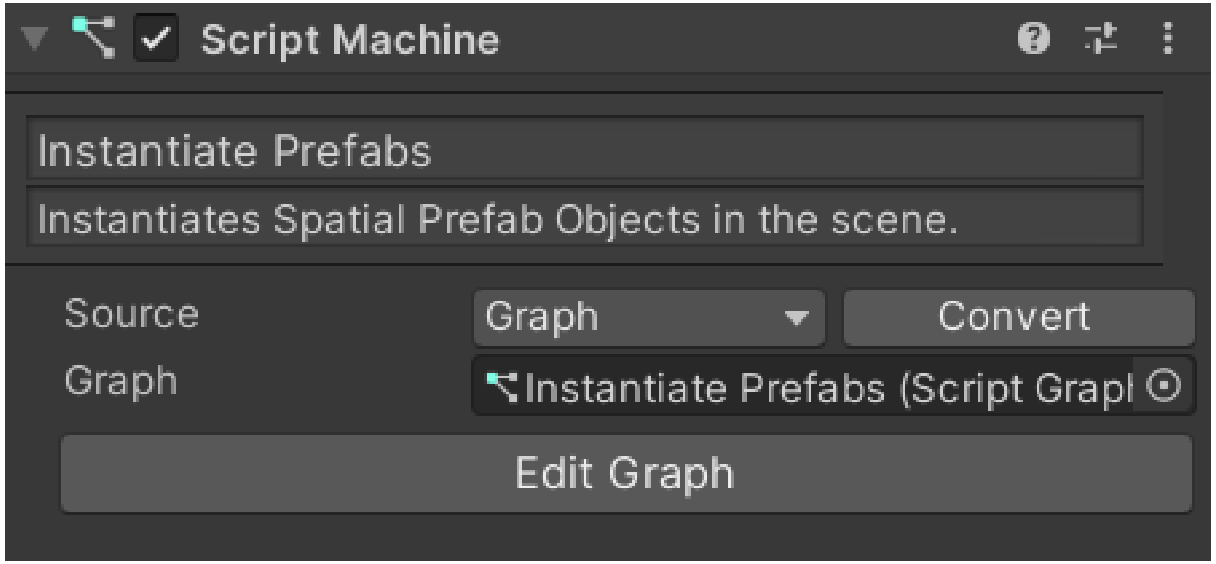
Scale 1.00

Prefab Objects

Getting your Prefab Objects in a Space

2: In any existing Spatial project, you can instantiate Prefab Objects using a visual script in Unity.

This is more complicated, but it is the only way to have a Space loaded with Prefab Objects automatically.



Instantiate Prefabs

Instantiates Spatial Prefab Objects in the scene.

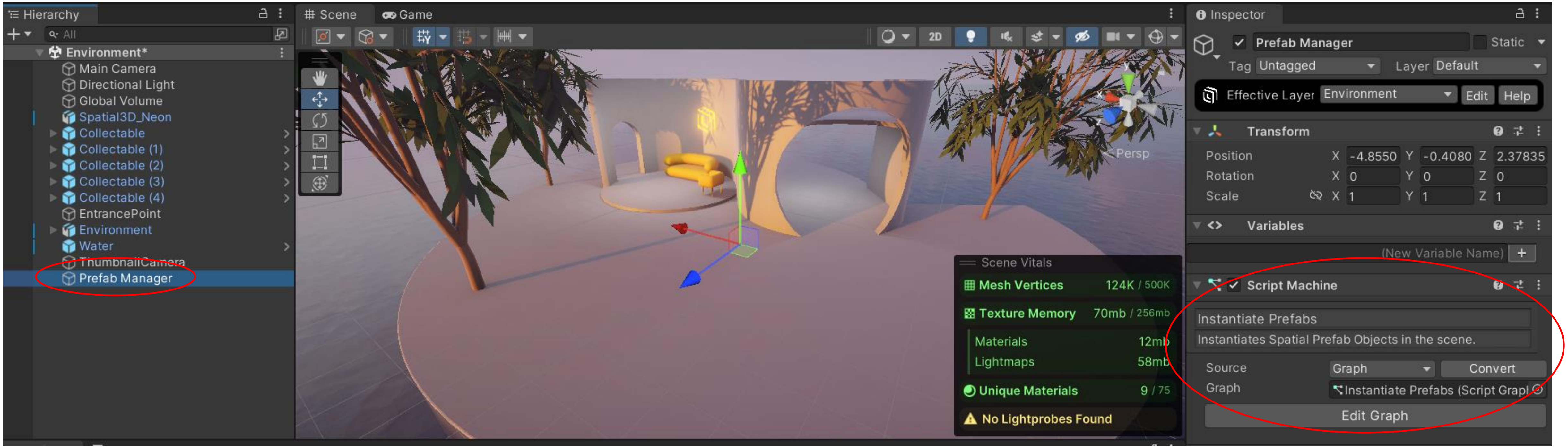
Source: Graph

Graph: Instantiate Prefabs (Script Grapl

Prefab Objects

Getting your Prefab Objects in a Space

For a script that instantiates Spatial Prefab Objects, it is best practice to apply it to an empty object in the scene, called something like "Prefab Manager". This way, we can better organize our scripts.



Environment*
Main Camera
Directional Light
Global Volume
Spatial3D_Neon
Collectable
Collectable (1)
Collectable (2)
Collectable (3)
Collectable (4)
EntrancePoint
Environment
Water
ThumbnailCamera
Prefab Manager

Scene Vitals
Mesh Vertices 124K / 500K
Texture Memory 70mb / 256mb
Materials 12mb
Lightmaps 58mb
Unique Materials 9 / 75
No Lightprobes Found

☑ Prefab Manager □ Static
Tag Untagged Layer Default
Effective Layer Environment

Transform
Position X -4.8550 Y -0.4080 Z 2.37835
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Variables
(New Variable Name) +

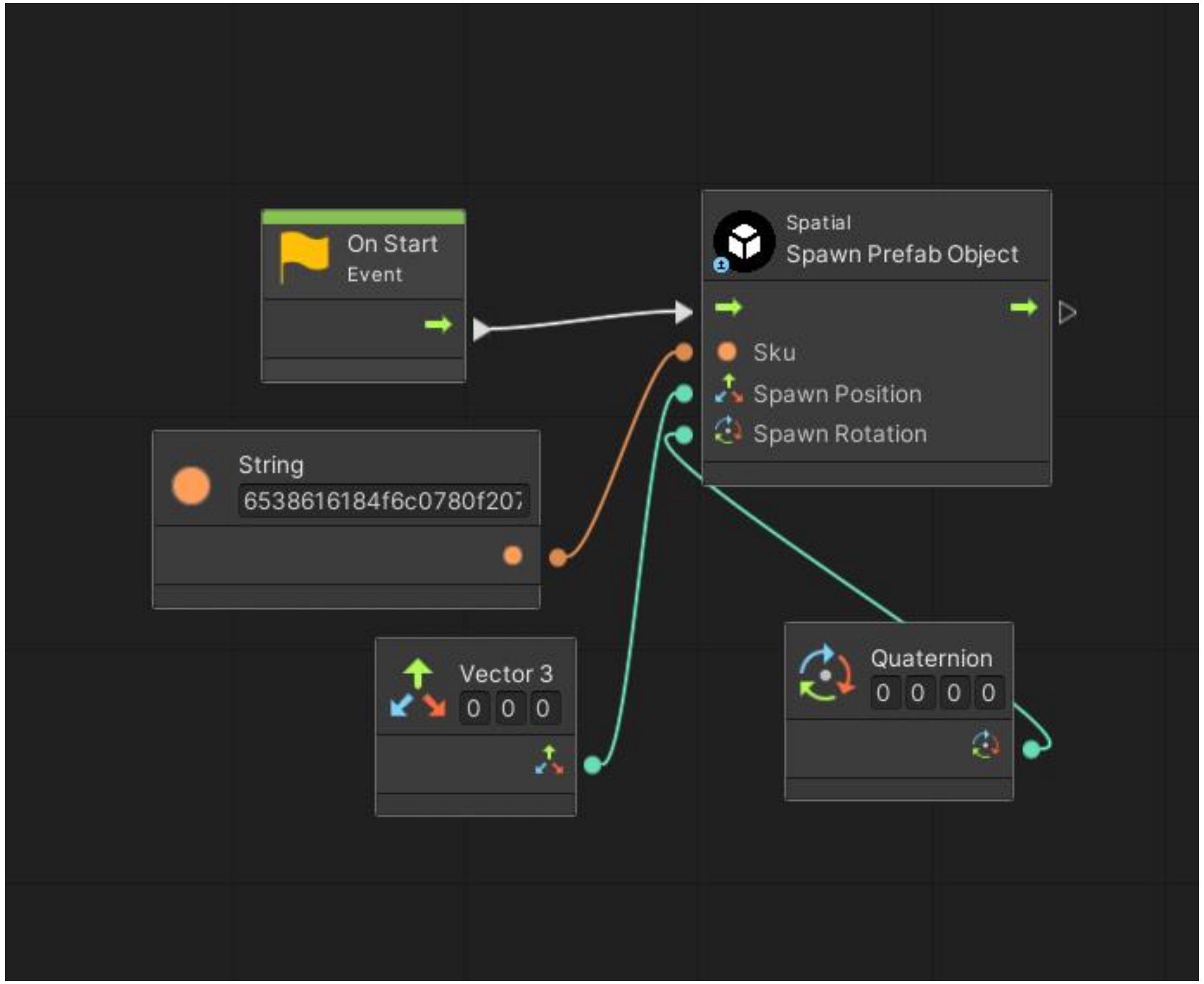
☑ Script Machine
Instantiate Prefabs
Instantiates Spatial Prefab Objects in the scene.
Source Graph
Graph Instantiate Prefabs (Script Grapl

Prefab Objects

Getting your Prefab Objects in a Space

Inside the graph, you can add a node called "Spatial: Spawn Prefab Object". It asks for a SKU code, which is generated for each Prefab Object when it is published for the first time.

Find this code by clicking "Publishing" and then selecting your Prefab Object, or you can find it here.



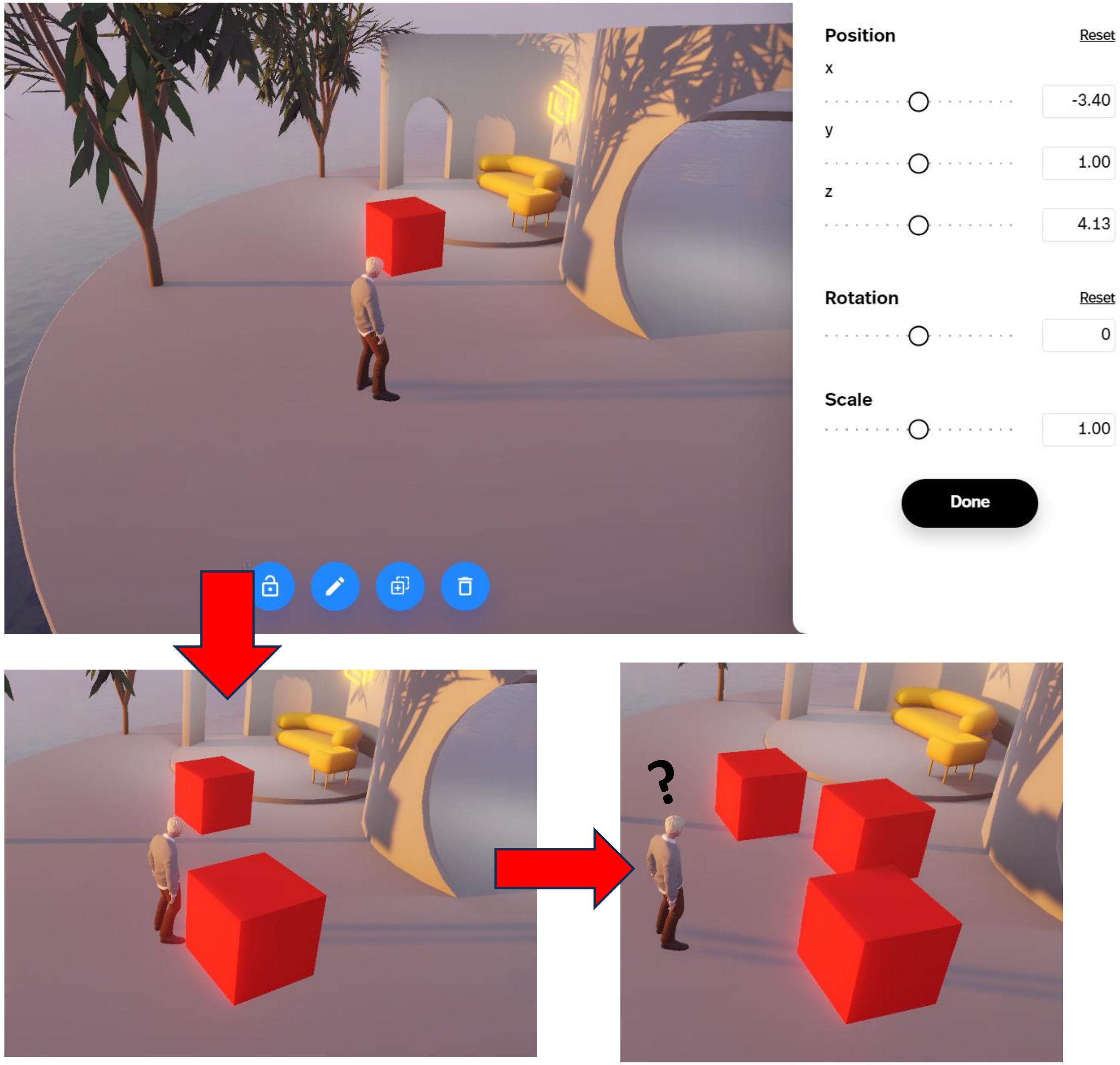
Prefab Objects

Getting your Prefab Objects in a Space

Now our Spatial Prefab Object instantiates in our Space every time it is started.

Starting your room doesn't delete all the Prefabs, so restarting multiple times can result in multiple copies of the Prefab.

Deal with this as you see fit (perhaps your script deletes all Prefab Objects before it instantiates any).



Position Reset

x -3.40

y 1.00

z 4.13

Rotation Reset

0

Scale

1.00

Done

Clicking "Test Active Scene" will trigger the On Start event. Therefore, testing 3 times will result in 3 On Start triggers.

Prefab Objects

Prefab Objects Limitations

Prefab Objects cannot use Unity's scene variables.

Therefore, in order to make variables that are global for Prefab objects, we need to make use of the Spatial Data Store script nodes.

For more information about Spatial Data Store, go back to slide 17 or visit Spatial's documentation.