

INTERACTABLES

Kit of hand push buttons, levers, and a detonator plunger! Geo, textures, and mechanic!
The levers and detonator are (reasonably) synced, but the push buttons are not.

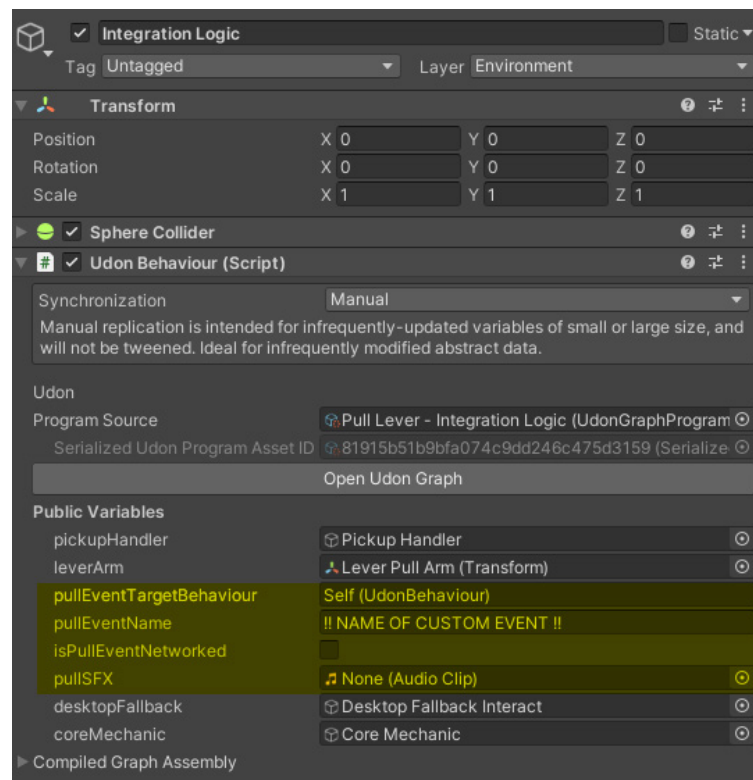
It uses Graphs, though. I am sorry.

LEVERS

Drop the lever prefab into your scene.

Click on the “Integration Logic” game object – change the ‘pullEventTargetBehaviour’ variable to the Udon Behaviour you want the lever to send events to when it’s pulled, then fill out the ‘pullEventName’ variable with the custom event in that target Udon Behaviour you want it to call.

If it needs to be a networked event, check the ‘isPullEventNetworked’ boolean.
That’s it!



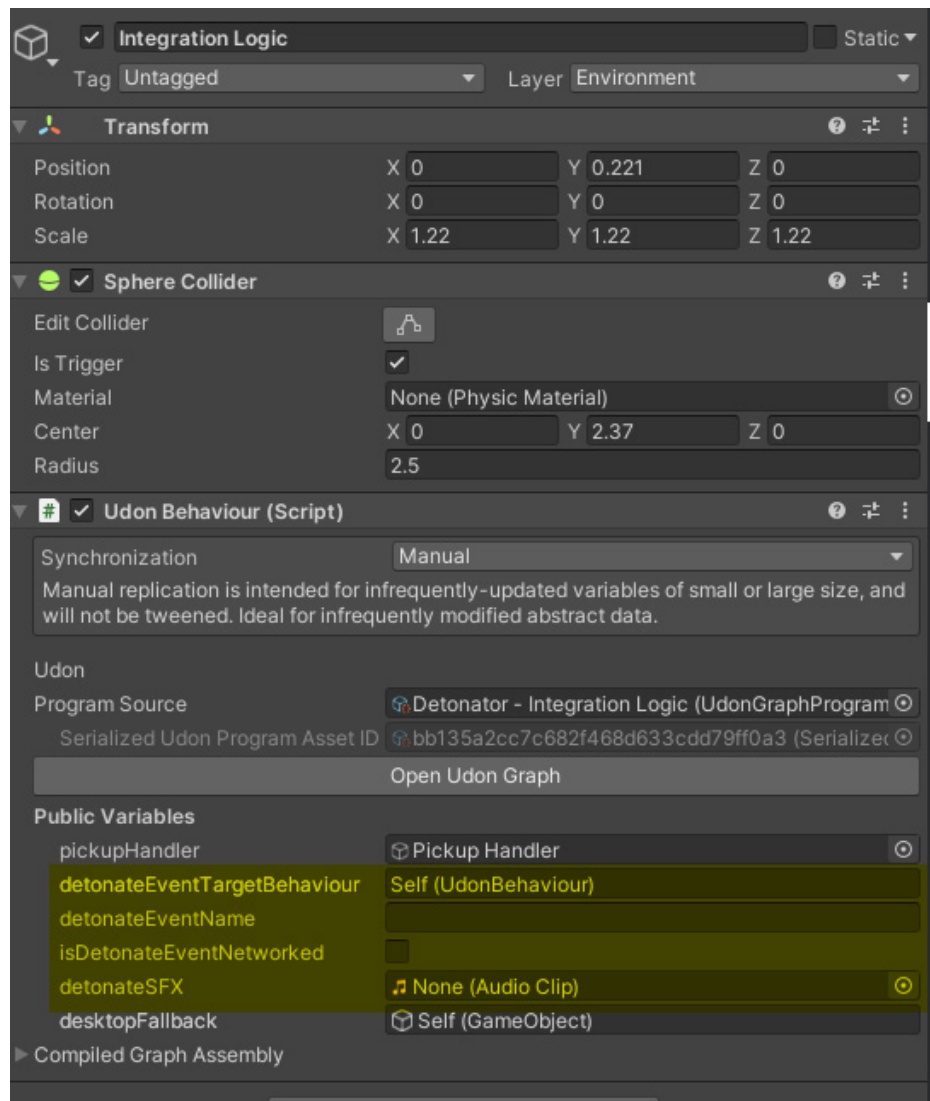
Make sure the ‘integration logic’ game object is on a layer that collides with the player. Also, drag a sound effect to the pullSFX public variable/slot or you’ll get bullied

DETONATOR PLUNGER

Drop the detonator prefab into your scene.

Click on the “Integration Logic” game object - change the ‘detonateEventTargetBehaviour’ to the Udon Behaviour you want the detonator to send events to when it’s pushed down, then fill out the ‘detonateEventName’ variable with the custom event name in that target udon behaviour you want it to call.

If it needs to be a networked event, check the ‘isDetonateEventNetworked’ boolean.
That’s it!



Make sure the ‘integration logic’ game object is on a layer that collides with the player. Also, drag a sound effect to the ‘detonateSFX’ public variable/slot or I will eat a shoe

HAND PUSH BUTTON

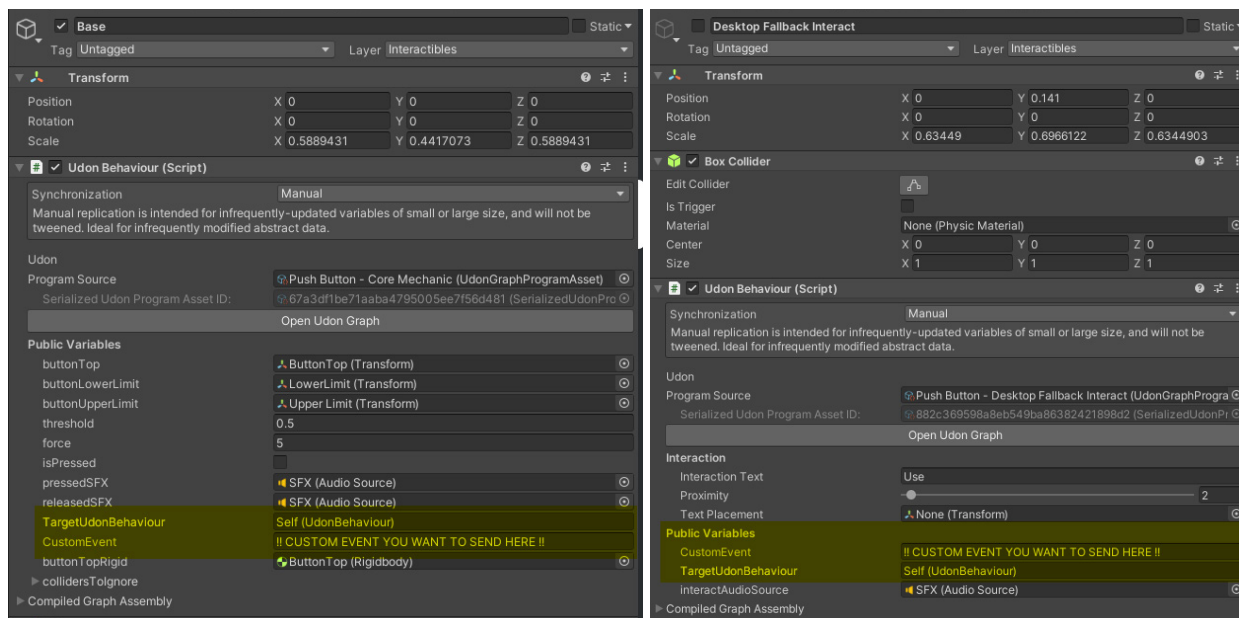
Not the most user-friendly bcs I didn't think we were gonna be prefab-ing it but we're doing that so HERE WE GO:

Drop push button prefab into scene.

Change the 'targetUdonBehaviour' and 'CustomEvent' public variable on the "Base" AND "Desktop Fallback Interact" game objects.

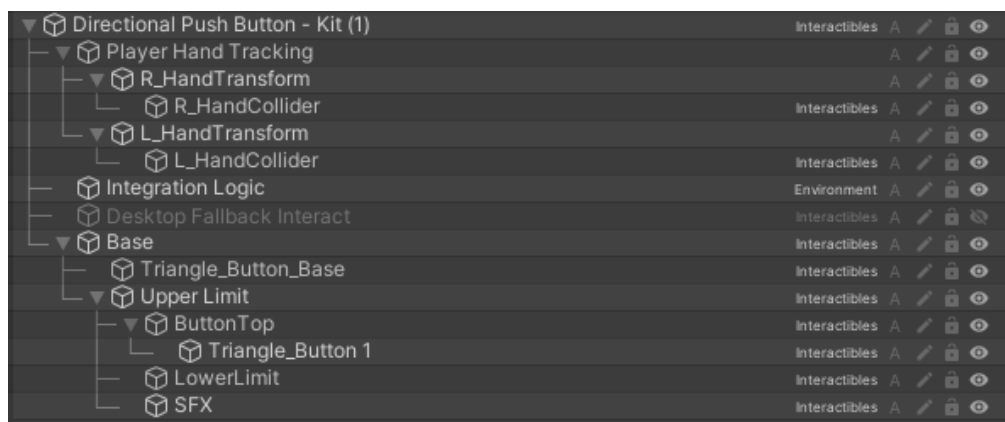
This will send out whatever custom event you type in to the "target" Udon Behaviour on press.

Also, add a sound to the "SFX" game object! Sounds!



Not sure if the prefab will carry layers over -- so the 'Integration Logic' game object needs to be on a layer that collides with the player. You can setup a new layer for the hand collider and the button to collide with each other if you run into issues with that.

Here's my setup for reference:



PUSH BUTTON WARNING:

There's player hand tracking included in *both* button prefabs.

You should move it out of the prefab's hierarchy / only use one instance of the hand tracking object/behaviour if you use both buttons... but you'll need to update the 'handTracking' public game object variable for one of the buttons (on the 'Integration Logic' game object).

You may not need the hand tracking if you're already getting the location of the player's hand or fingers elsewhere in what you're making...BUT the button mechanic does a range check with a trigger volume to enable/disable the hand tracking for performance. I did not test what happens if you delete the hand tracking game object so you might crash the behavior if you try that.

Beware! Just rip it to shreds and do what you need with it, wu tang energy.