

# **Interactable Objects by Karet**

<https://twitter.com/5xa>

Discord: Karet#3314

**This prefab Requires UdonSharp**

<https://github.com/MerlinVR/UdonSharp>

**This prefab includes VRCPlayersOnlyMirror by ascertainbluecat.**

**Go thank them for their hard work.**

<https://twitter.com/ascertainbluecat>

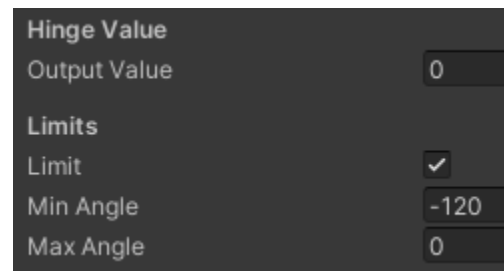
**Audio provided by ZapSplat**

## Info

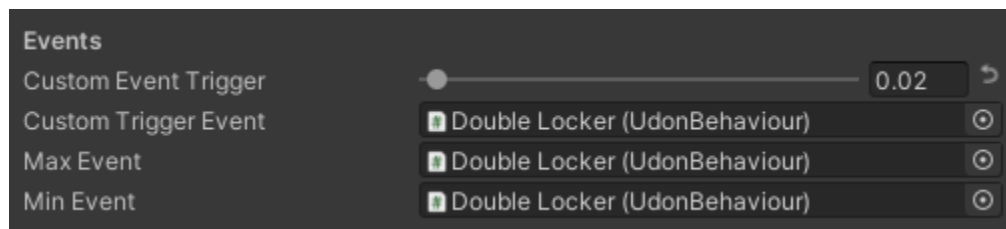
This prefab contains 5 generic methods of interaction.

1. Hinge, for doors and levers
2. Drawer, for filing cabinets and things you pull open
3. Dial, for volume knobs and such just grab and rotate
4. Pushbutton, physical buttons you can push with your finger
5. PushSlider, like a fader on a mixer you can push with your finger

Each of these interactable has a float value of 0-1 that is based off the limits you set on the udon behaviour. 0 becomes the minimum value and 1 becomes the maximum value. The output value is how far the object is between its min and max.

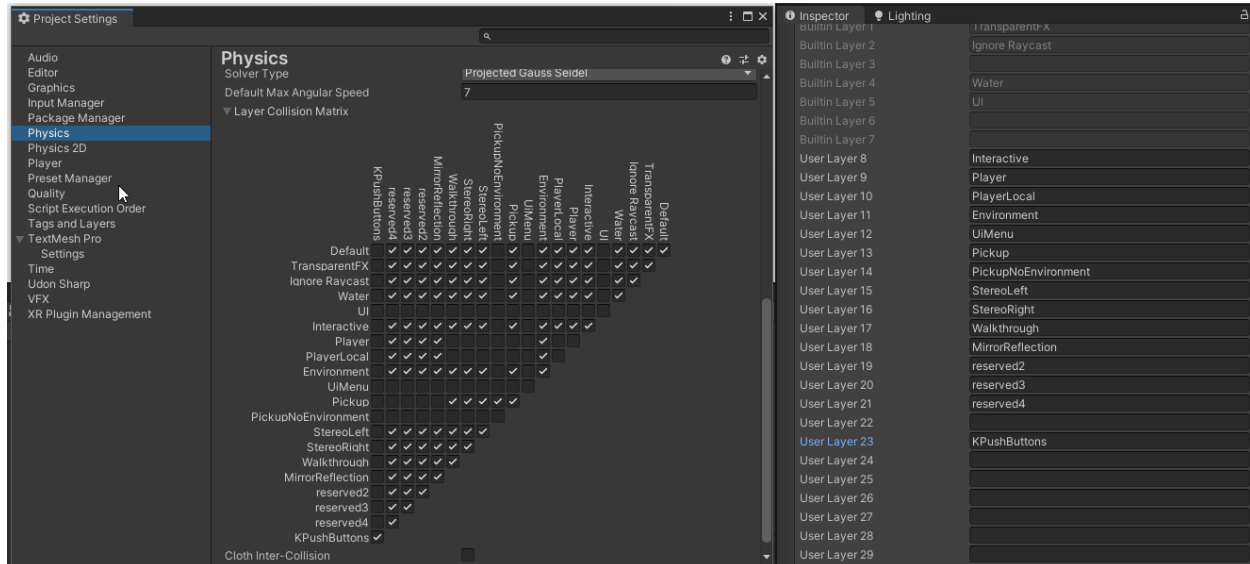


Each interactable can be configured to send events when they reach their maximum, minimum, or custom set values. The value of each interactable is also exposed so other scripts can read it directly.

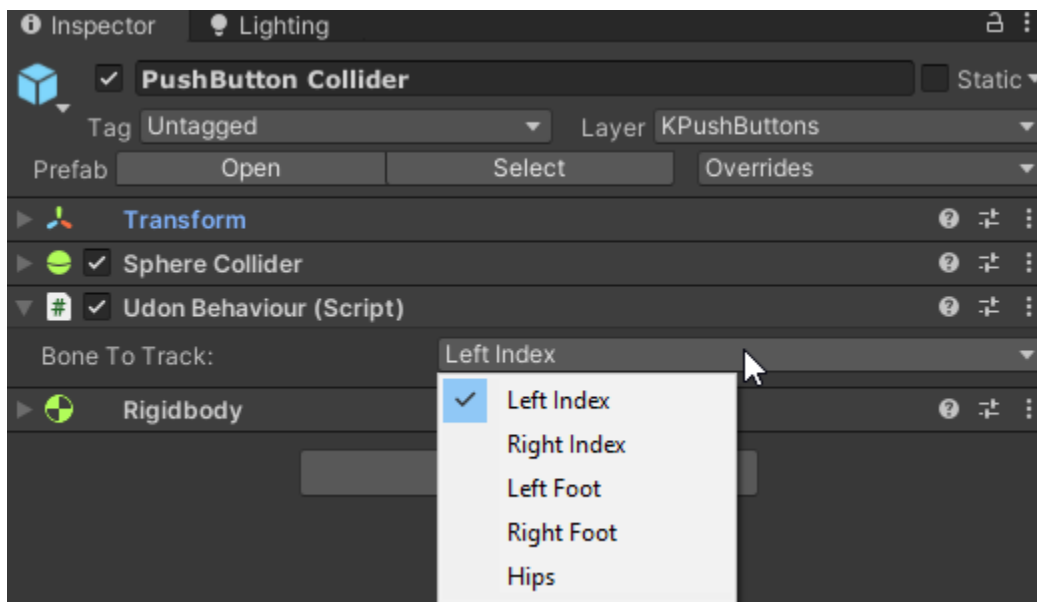


## PushButton/Slider Setup

The PushButton and PushSliders require a bit of setup. By default they use physics Layer 23 but they can use any layer that only interacts with itself.



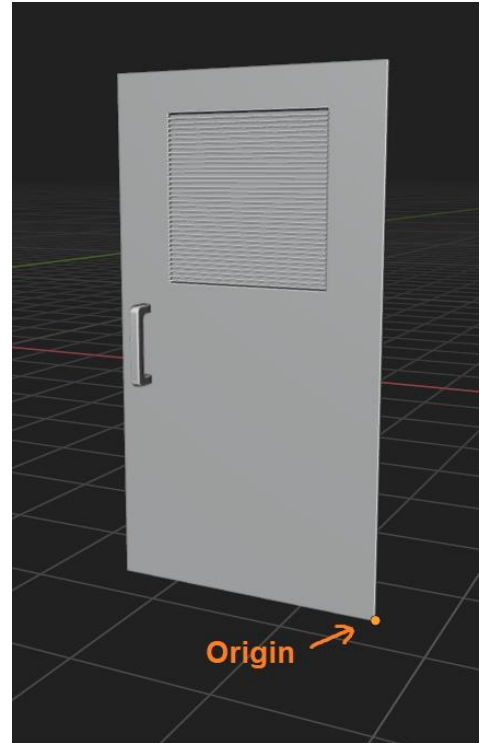
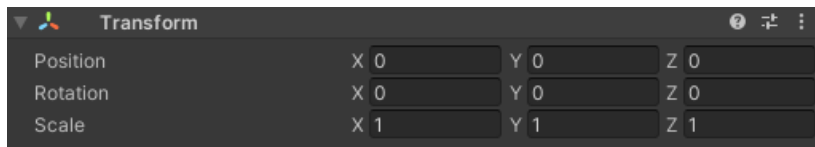
After Setting up the physics layer you will also need to drag some of the PushButton Collider prefabs into the scene and choose what bone you want them to follow. These are the colliders that allow you to interact with the PushButtons and Push Sliders.



## Mesh Setup

The Hinge objects require the origin point of the mesh be at the point you want the object to pivot around. The orientation of the mesh also matters as it will be rotated around the local Z axis in unity. For the Drawer, PushSlider, and PushButton, they will be moved back and forth along the Z axis and the Dial will be rotated around the Z axis..

The mesh GameObjects transform should be 0,0,0 for the position and rotation however its parent objects orientation doesn't matter so if you need to rotate something then you can set an empty GameObject as the parent of the interactable.

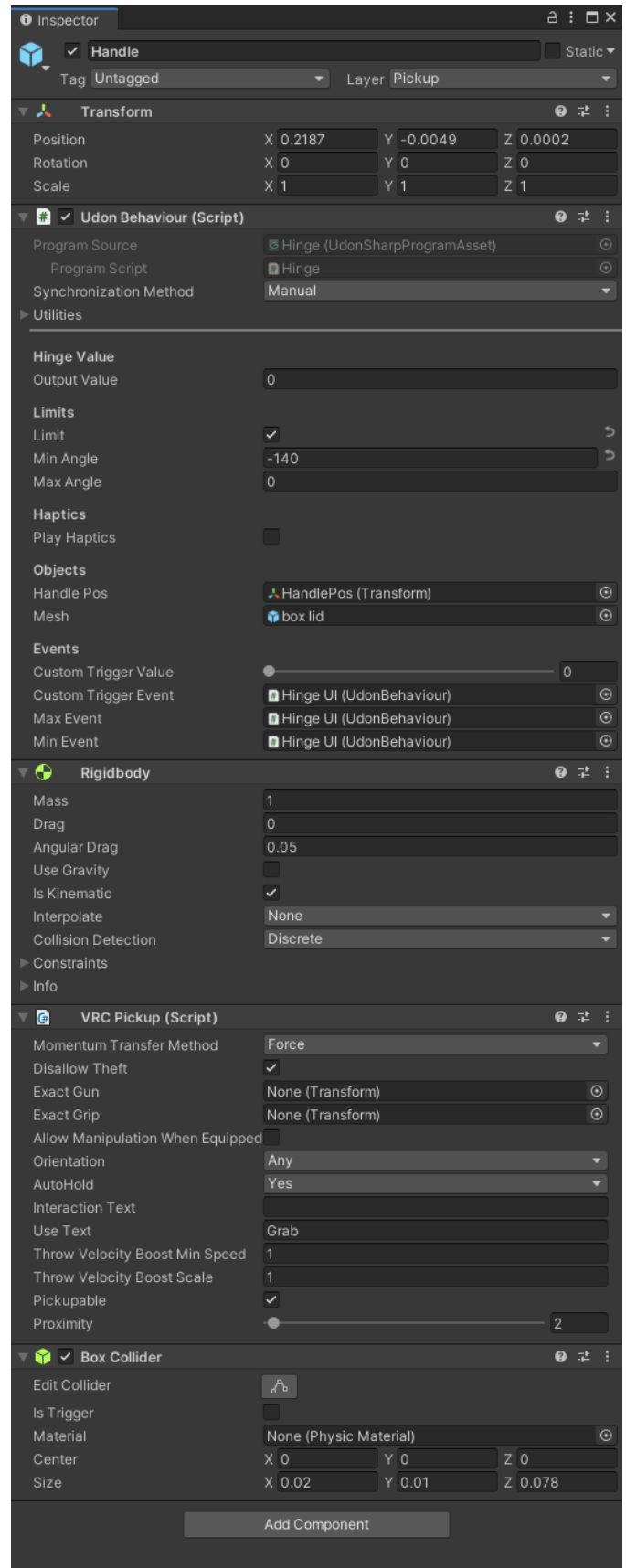


## UdonBehaviour Setup

To create a new Hinge, Dial, or Drawer we are going to abuse VRC Pickups a bit. Create a new Gameobject and give it a VRC Pickup script, some kind of collider, and an udonbehaviour of whatever interactable you want to make.

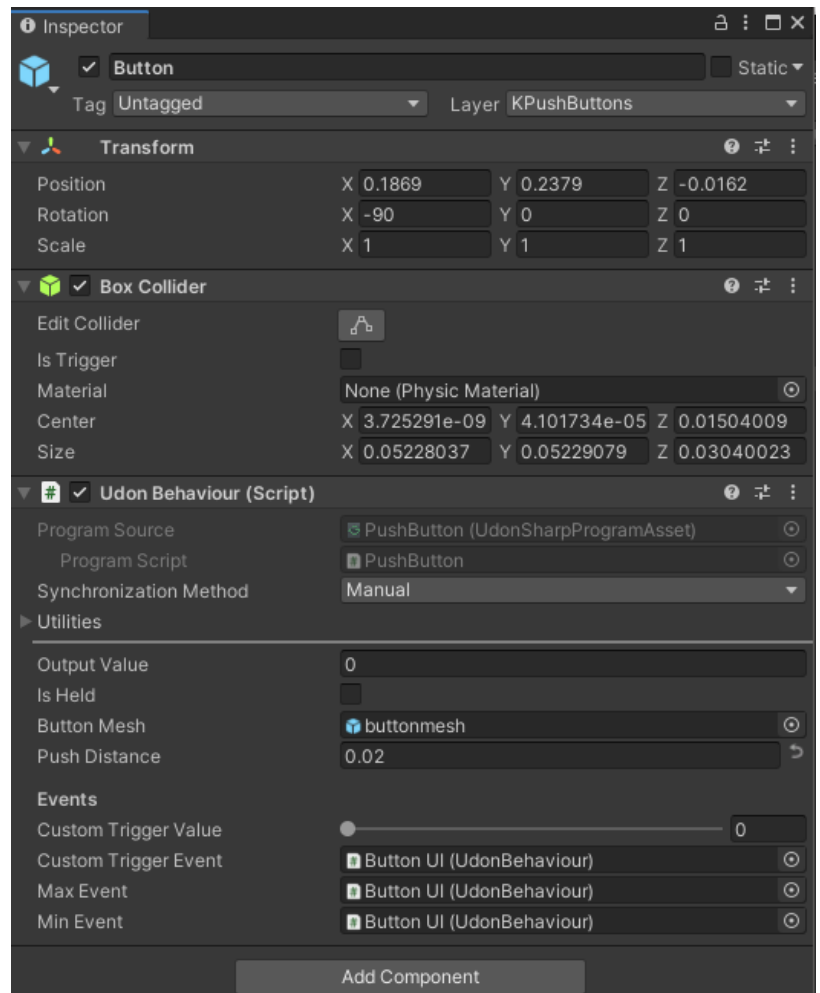
For the Hinge, Dial, and Drawer there is a serialized field for the Mesh that is going to be manipulated and a Transform called Handle Pos.

This transform should be a child of the mesh GameObject and positioned where “handle” on the mesh is. The pickup object that the players grab to manipulate the interactable will teleport back to this position when the player drops it.



To create a Pushbutton we need a GameObject with a collider that has the button's mesh as a child at 0,0,0. Simply put this GameObject on the physics layer we setup earlier and specify how far back from 0 you want the button to be pushed.

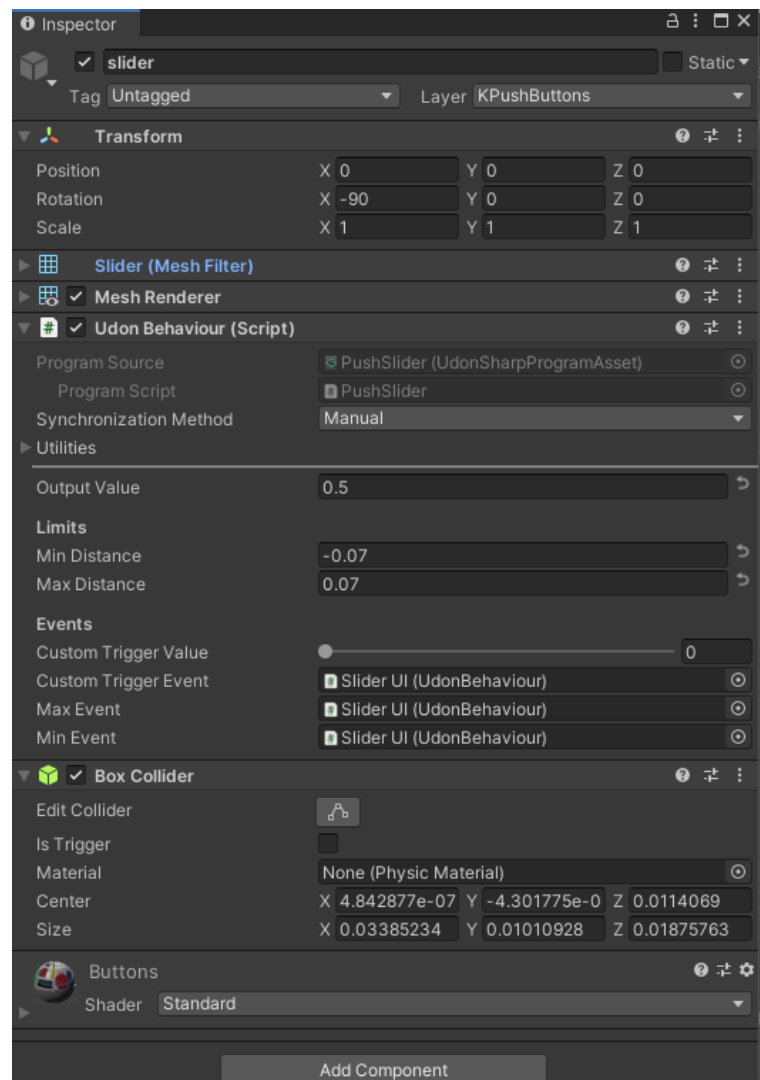
When a players finger collider enters the collider on this GameObject the button will begin to move based on how far they have moved into the collider. This looks best if the collider is the same size as the button mesh.



Buttons also uniquely have a public bool, “Is Held”, that returns true if the player is holding the button down all the way.

Lastly the PushSlider is much like the PushButton except the UdonBehaviour and collider go onto the same GameObject as the Mesh.

You also have to specify limits much like the Drawer. Think of it like a drawer you push rather than grab.



Hopefully that all works out for you. If you have any issues, you can always reach out to me on discord or in game. If you would like to see some examples of these interactable objects in game, drop by the LPD Station world.