

Using the hands

Just drag

Kandooz -> Hands -> Sci-Fi Hand->Prefabs->KVR Rig To your scene.

For interaction add the `interactable` and `throwable` components to the object you want to interact grab, and edit the settings on the `interactable` components, more info on the rest of the documentation.

What is KVR?

KVR is an VR Hand controller and Interaction System that is designed to be simple and rely only on unity APIs, This is done to make sure that everything is cross platform and done within unity without requiring external tools or changing config files

In it's current state KVR have two systems:

- The Hand Pose control
- VR Input controls
- Interaction System

The Hand Pose Control

The hand controller is designed from the ground up using the **Playables API** to control the hand and finger poses it is designed to give control over individual fingers, and lerp between different poses seamlessly.

To use the system you need to either ¹:

- A hand model with at least two poses
 - o closed fist
 - o relaxed opened hand
 - o Extra poses if needed
- Download any of our KVR models from the asset store (they come pre-configured)

If you are using your own model you need to create a **Hand data manager** :

Project ->create ->Kandooz -> KVR -> Hand data

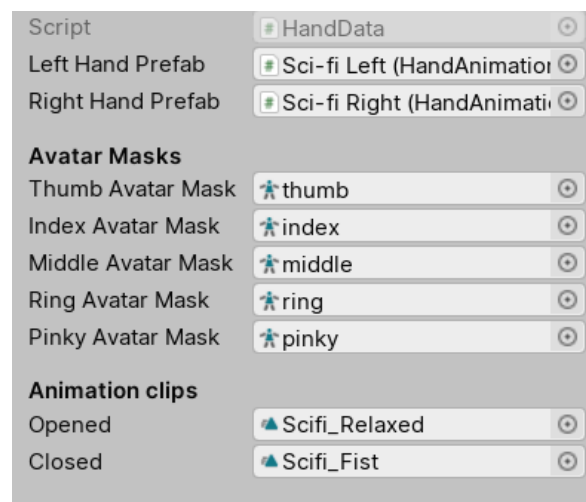
If you are just using the Hand control system there is no need to set the Prefabs for the right and left hands.

What is required is the Avatar Masks for each finger.

They are normal unity avatar mask overrides

Where you must enable only the corresponding finger.

And finally you have to put the animation clips for the opened and closed hands poses

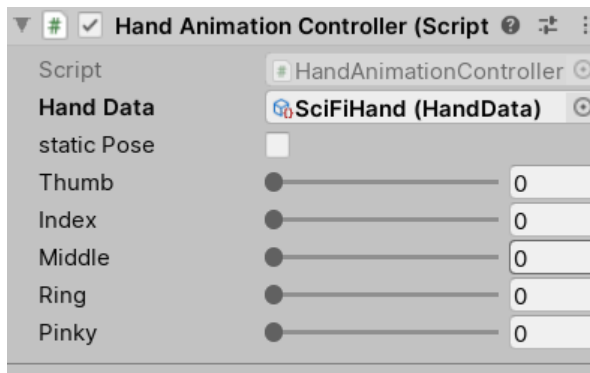


Once you have the Hand data set up need to drag and drop your hand model to the scene and put each hand inside a container.



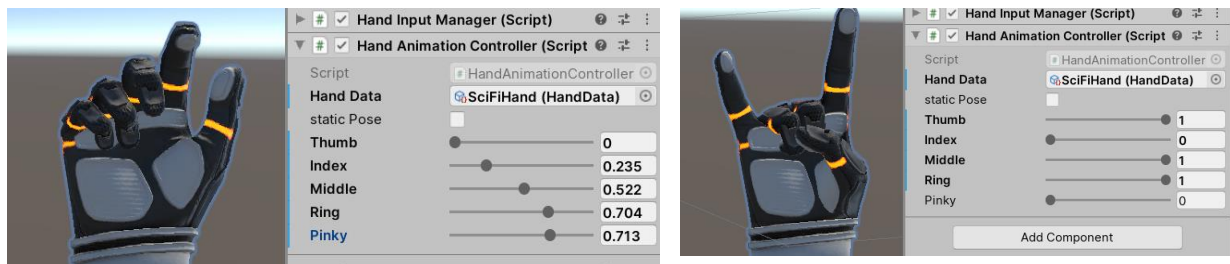
You don't have to put them in a container but this will save your life later.

Add the **HandAnimationController** component to the container and then drag the hand Data you created to the Hand data property.



Now you can change the value of each finger to define how closed it is, 0 means the hand is open and 1 means its fully closed.

It is basically lerping between the animations defined in step 1



The VR Input Control System

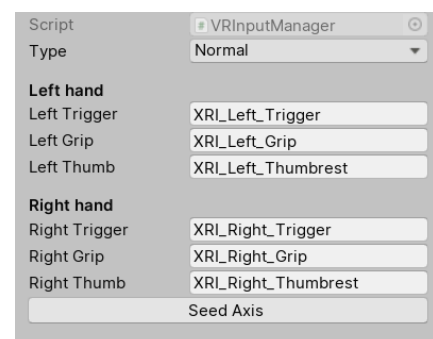
This system uses the unity Input manager not the input system but it can easily be changed to use the new input system (The option will be added soon).

This system has two parts an input manager scriptable object and a Hand input Manager script

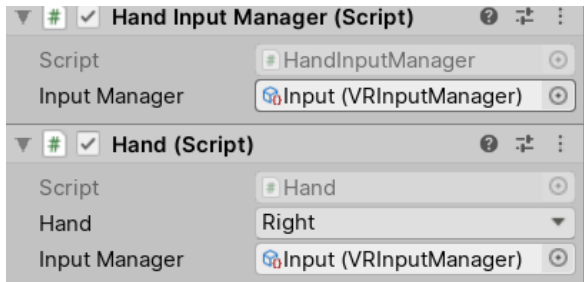
To create an input manager script you just need to

Project ->create ->Kandooz -> KVR -> Input Manager

This will create the input manager scriptable object and populate it with some default values, those values represent the Axe names that the system will create in the unity Input manager, if you don't want to give them specific names just press **Seed Axis** , if you are using a Knuckles controller change the type to Knuckles and then press Seed Axis Again.



The second part of the Input control System is the **HandInputManager** component, you just need to drag it to the same container as the Hand animation Controller, this will add another component Called **Hand** to your game object and then drag the input manager you just created to both of them .



And Make sure to change the Hand property in the Hand component to be right or left

This will make the hand Track the position and will animate the fingers depending on the buttons pressed on the controllers.

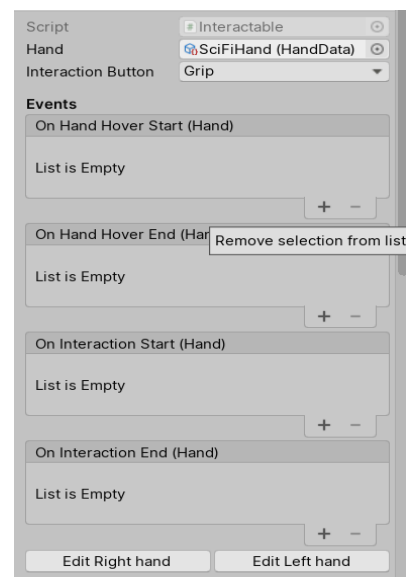
The Interaction System

For the interaction system you just need to add two components to the object you want to interact with, the first one is the interactable component.

The first thing you need to set is the Hand Data property, and which button will start the interaction with the object.

You can then start adding Events to what will happen when the hand start Hovering, Interact, end the interaction and end hovering.

And then you need to specify how both hands will interact with the object by either pressing edit right hand or left hand



You can change how closed each finger will be and also if a finger can move while holding the object you define the limits of the finger for example in gun we defined the position and rotation of the gun relative to the hand and then locked all the finger except for the Index which can move but with the limits specified.

