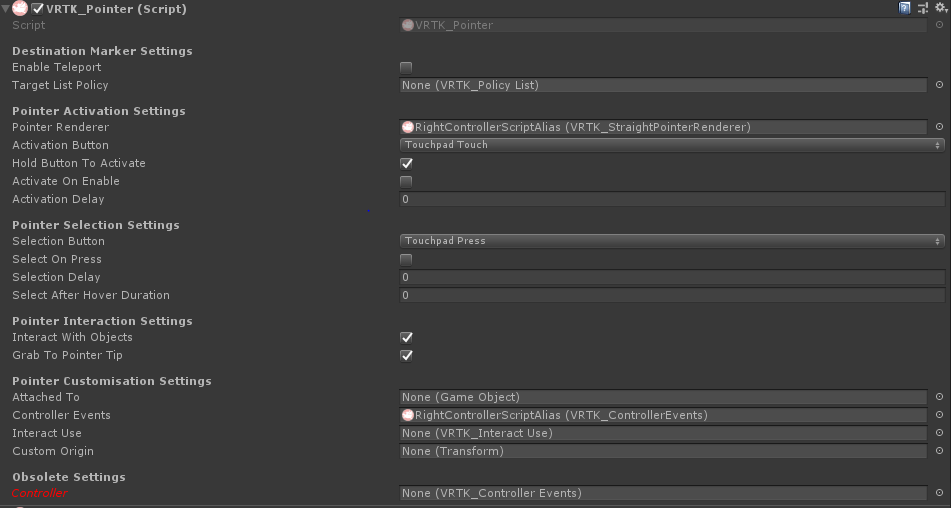
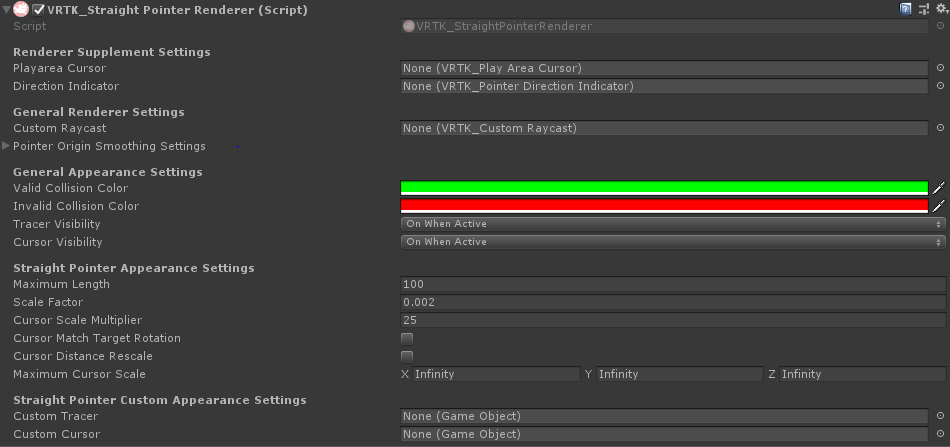
1. Open Unity project 3D (Unity 2018.2)
2. Download VRTK from GitHub, drag and drop into assets
3. Get oculus integration from unity asset store
4. Delete main camera
5. Create empty gameObject call it [VRTK\_SDKManager]
6. Underneath [VRTK\_SDKManager] create [VRTK\_SDKSetups]
7. Under [VRTK\_SDKSetups] add VRTK\_SDK Manager (Script)
8. Under [VRTK\_SDKSetups] create empty gameObject Oculus
9. Add VRTK\_SDKSetup script to oculus game object
10. Select GearVR (Android: Oculus) in sdk selection
11. Add OVRCameraRig and LocalAvatar as children of Oculus
12. On Oculus, under the VRTK\_SDK Setup **script** (Note this is under the Oculus gameObject) object references, deselect auto populate and add game objects as seen below (this can be done by clicking and dragging from the hierarchy to the slots)

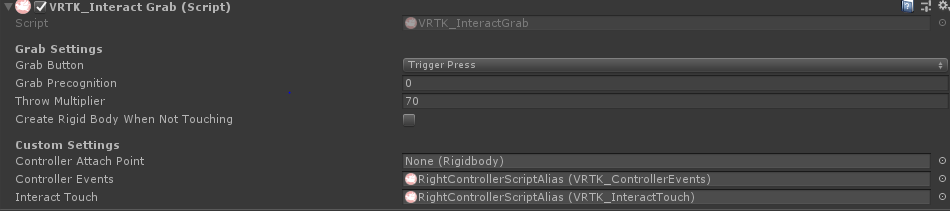


1. Create an empty gameObject that is child only to your scene. Call it the [VRTK\_Scripts]
2. Underneath [VRTK\_Scripts] create two children called LeftControllerScriptAlias and RightControllerScriptAlias respectively
3. Highlight both children in the hierarchy and add (this will add to both) VRTK\_ControllerEvents, VRTK\_Pointer, VRTK\_StraightPointerRenderer, VRTK\_InteractGrab, VRTK\_InteractTouch and Rigidbodies to the ControllerScriptAliases
4. Select RightControllerScriptAlias and follow images below:

PLEASE NOTE THAT YOU MUST TURN ON “INTERACT WITH OBJECTS” AND “GRAB TO POINTER TIP”

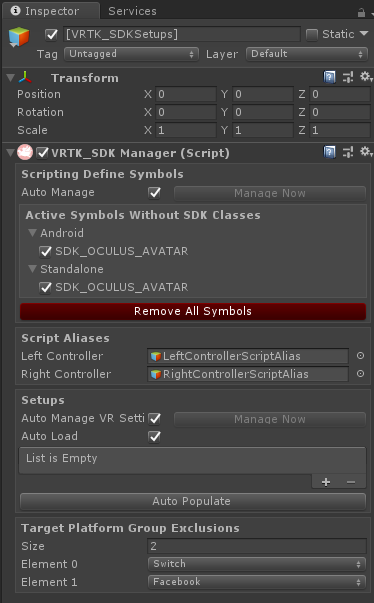
“Hold button to activate” can be turned on as well, but is not absolutely necessary



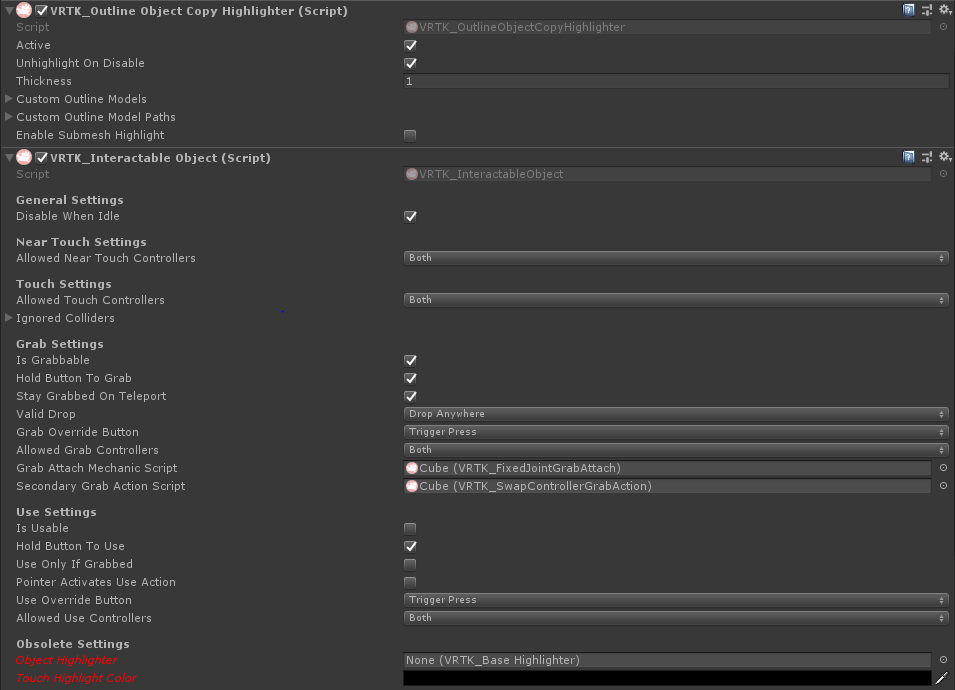


Drag the RightControllerScriptAlias onto the slots that are filled for all scripts. (ControllerEvents and Interact Touch are not changed). NOTE! Turn off ‘use gravity’ on the rigidbody.

1. Do the same for the LeftControllerScriptAlias
2. At this point, return back to [VRTK\_SDKSetups] and look under Script Aliases (can be found under VRTK\_SDK Manager (Script)) drag and drop LeftControllerScriptAlias and RightControllerScriptAlias under their respective slots as shown.



1. For organizational purposes, create an empty gameObject and name it ‘SceneObjects’. This is where you will put all of your objects.
2. Create an empty game object and name it EventSystem. Add the EventSystem and StandaloneInputModule scripts.
3. For the purpose of a random example. Create a cube (3D- gameObject - cube) (note these scripts can be applied to any objects so long as scripts do not collide).
4. Attach to the object VRTK\_Outline Object Copy Highlighter (Script), VRTK\_Interactable Object (Script), VRTK\_Fixed Joint Grab Attach (Script), VRTK\_Child Of Controller Grab Attach (Script),VRTK\_Swap Controller Grab Action (Script), Rigidbody



1. If all goes well this should produce a working grabbable cube using the Oculus Go. There is a strong chance it will not go well.

\*\*\*\*\*\*\*\*\*\*\*\* TO TELEPORT AS WELL, JUST ADD THE BASIC TELEPORT SCRIPT TO THE POINTER AND ENABLE TELEPORT ON THE POINTER SCRIPT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*