VORTEX-COVID PPE Setup Tutorial

Written by: Douglas Burnside

Table of Contents

Importing PPE Models

Rigid Objects

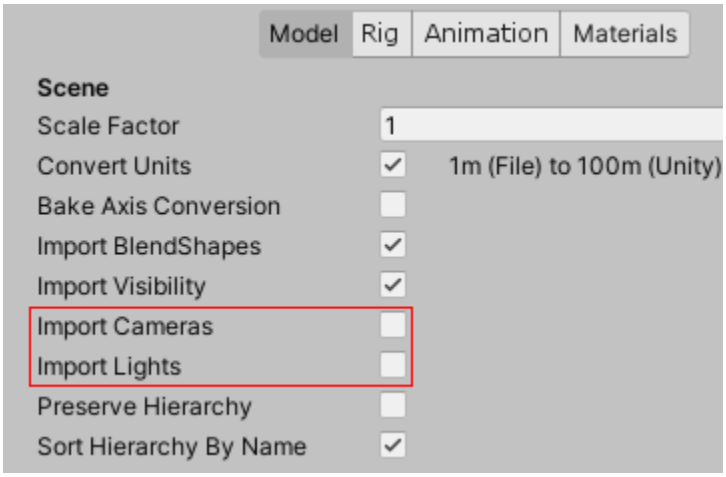
To import a PPE model into the project and allow modifications to the mesh later, the Blender save file can be added directly. Drag the .blend file directly into the asset browser in the directory where models are stored, under "*Assets* > *VORTEX_COVID* > *Models* > *Cloth"*. Blender will need to be installed on your machine for the models to load properly in the scene.

Skinned Objects

If the PPE model needs to animate with the avatar, it will need to be skin painted with the avatar as well as having a separate unskinned rigid model. The skinned cloth should be a child of the avatar's armature and will be used for animation. The unskinned model will serve as a static mesh for grab detection and collision. When importing the separate models, follow the same import process as a rigid object.

Model Import Settings

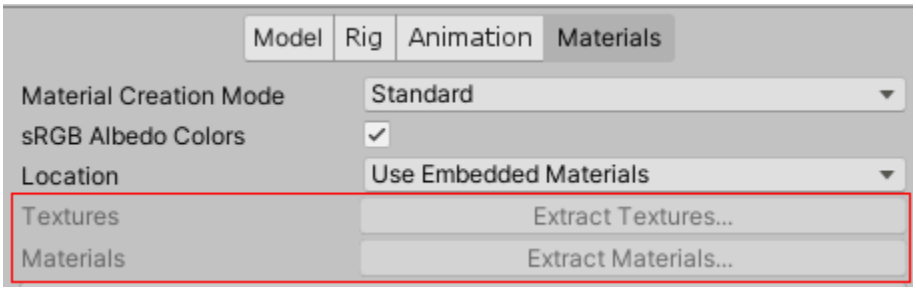
To prevent importing cameras and lights from the file, navigate to the model import settings and deselect *"Import Cameras"* and *"Import Lights"*.



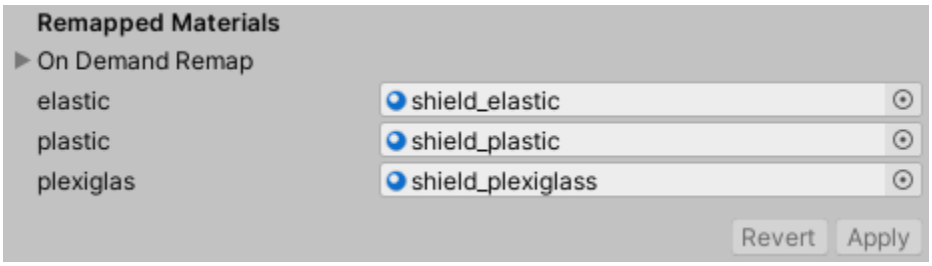
Material and Texture Extraction

If the materials on the PPE will be modified in-engine, they will need to be separated from the blend file in the project. Navigate to the material import settings and click on *"Extract Textures..*" if the textures are contained in the file and *"Extract Materials…"*. Both of these options will open a prompt asking where to save these new assets. This will create new Texture and Material

assets based on their name in the blend file. Textures are stored in "*Assets* > *VORTEX_COVID* > *Textures"* and Materials are stored in "*Assets* > *VORTEX_COVID* > *Materials"*.



Afterwards, the materials will need to be remapped in the model's material import settings. Under the *"Remapped Materials"* section, the newly created materials from the previous step can be assigned. After they have been remapped, click the *"Apply"* button to save the changes.

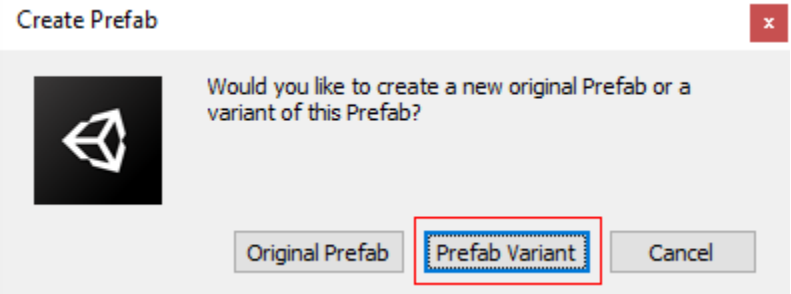


Setting Up Prefabs for PPE

Creating the Prefabs

When creating a prefab for a piece of PPE, it will need to contain the model as a child of an empty GameObject. This allows the mesh and hierarchy to update when the file is updated and prevents the need to recreate or manually update the prefab for every modification. Once it is made a child of the object, it can be named according to which PPE it is. This object can then be dragged into the asset browser to create a prefab. This is what will be placed in the scene.

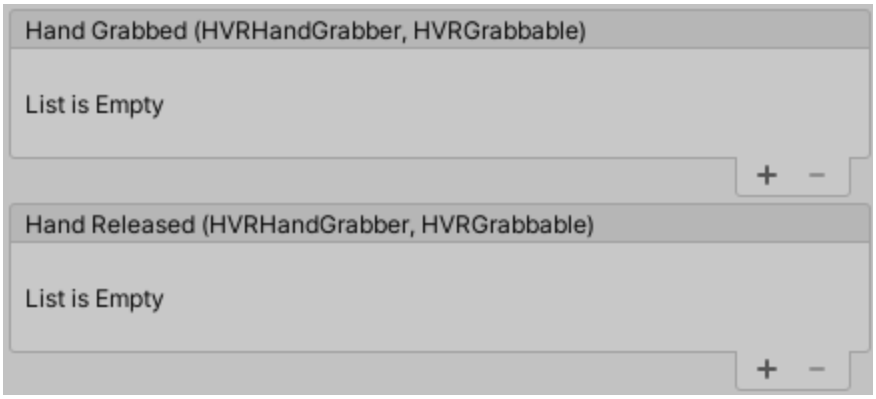
After the base prefab is created, a second duplicate prefab will need to be created as a prefab variant. Drag the original prefab into a scene and drag that instantiated object back into the asset browser, then click *"Prefab Variant"*. This duplicate will be used to place on the avatar with the same mesh and hierarchy but will have different scripts. It should be named differently from the original prefab to help discern which prefab will be placed in the scene or on the avatar. Any changes made to the original prefab will also be made in the variant, so for some future steps some components will have to be removed from the variant.



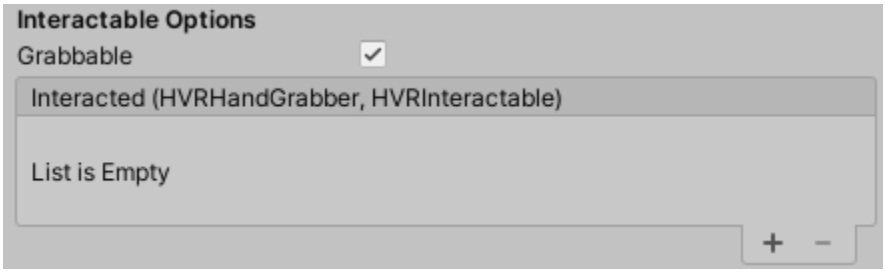
Adding Cloth Interaction Scripts and Components

HVR Interactable

Starting in the base prefab and on the empty GameObject, the *"HVRInteractable.cs"* script will need to be added to allow HVR hand grabbing or interacting. The *"Hand Grabbed"* and *"Hand Released"* events can be used for triggering any functions that need to occur when the object is either grabbed or released by a hand. Note that both of these events will only be invoked if the interactable is marked as grabbable in the Interactable Options.

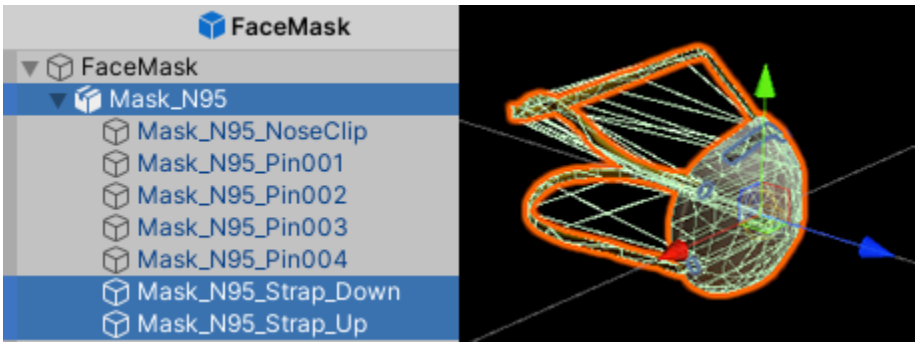


The *"Grabbable"* boolean toggles the object between its "grabbable" and "interactable" states. Grabbable objects will be pulled to and grabbed by the hand, moving with it. Interactable objects will only trigger the *"Interacted"* event when a hand grabs it and remains stationary. The original PPE prefab should have this variable checked and the variant should have it unchecked.



HVR Grabbable Child

Additionally, under the model object for both prefabs, the *"HVRGrabbableChild.cs"* script will need to be added on each child object that will contain a collider. This ensures that the collision events will be sent to the parent object and allow it to be grabbed using that collider.



Rigidbody and Colliders

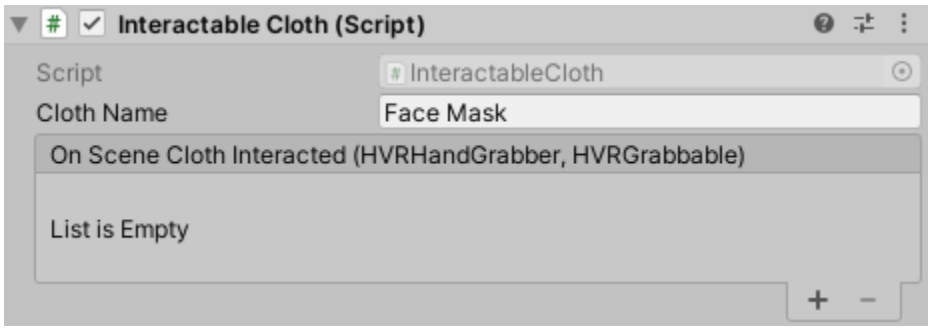
On the original prefab, a rigidbody should be added to the base object so it can receive collision events and be affected by any rigidbody interactions. This should be removed from the variant prefab after being added since this object will not move or use rigidbody interactions.

On both prefabs, colliders should be added to any part of the PPE that needs to collide with anything else in the scene or if it needs to be grabbable from that position. Convex mesh colliders are recommended for complex objects.

Interactable Cloth

On the original prefab, the *"InteractableCloth.cs"* script will be added on the base object. This will mark the object as part of an interactable cloth pair and allow the controller to find it in the scene. The cloth name should be the same as what the object name is, with any additional information to differentiate it from other PPE. This name will be used in other components and it *MUST* have the exact same name to pair correctly.

The *“On Scene Cloth Interacted”* event will trigger if this object is grabbed or interacted with by a hand. This is mainly used for communication with the controller, but it may also be used for triggering functions that use both the *Hand Grabbed* and *Interacted* events.

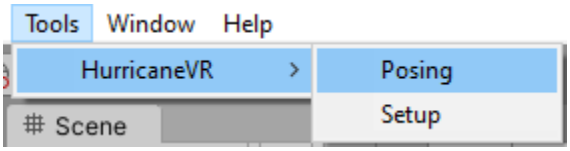


Worn Cloth

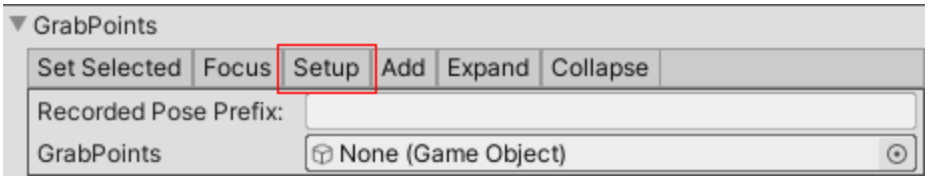
On the variant prefab, the *“WornCloth.cs”* script will replace the *Interactable Cloth* component, which should be removed. The *“Object With Skinned Mesh”* should be set to the GameObject that contains the meshes that will be displayed on the model. If it is a rigid object, this field can be left empty. If the avatar it is placed on has this PPE as a skinned object, that will fill this field when it is placed on the avatar prefab.

Creating Grab Points

To allow the PPE to be grabbed or interacted with, there must be at least one point where the hand can attach to. To begin this up, open the *“HVR Posing”* editor window.

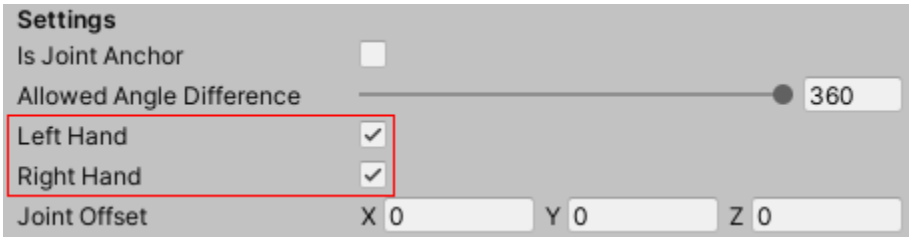


Open the original prefab, select the base GameObject which has the HVR Interactable, and click on *“Setup”* under *“GrabPoints”*.



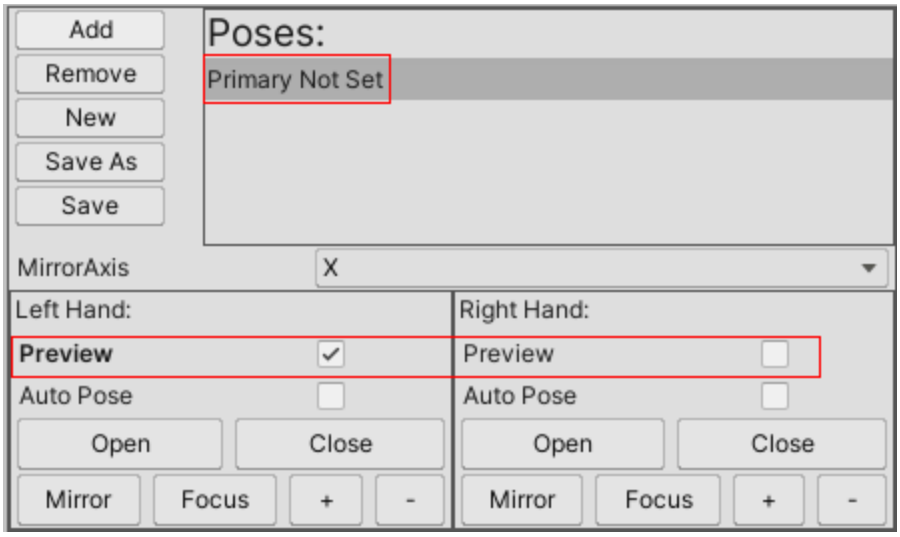
This will create an object in the prefab called *"GrabPoints"*, which is a container for which all grab points should be under. To make a grab point, drag the previously created *GrabPoints* object into the "*HVR Posing"* window and click *"Add"*.

The two scripts that are added to every grab point are *"HVRPosableGrabPoint.cs"* and "*HVRHandPoser.cs"*. The Posable Grab Point component doesn't need to be changed, but if only one hand should grab the object, it can be specified in the *"Settings"* section. The Hand Poser will be used to create hand poses for that specific grab point.

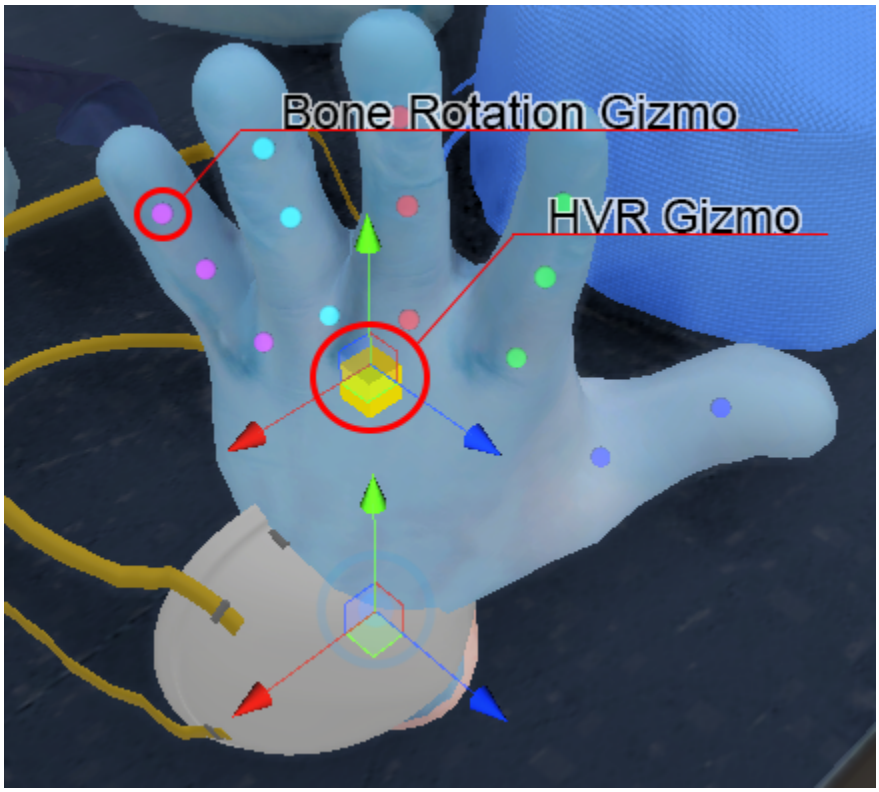


Creating Hand Poses

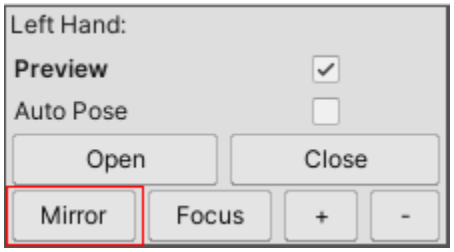
When creating a grab point, at least one hand pose will need to be added to the Hand Poser to allow the hand to animate and visually grab the object properly. When first created, the primary hand pose will be empty. Enter a scene with the cloth in it, navigate to the grab point, click on *"Primary Not Set"* to select the pose that will be edited, then click *"Preview"* under either hand to begin editing the pose. If you want to let HVR automatically pose the hand based on collision, select *"Auto Pose"*.



The avatar poser prefab will appear and the hand will be targeted to the origin of the object by default. The hand position should be primarily modified by the GrabPoint object, since this will be the anchor position. The yellow HVR gizmo can be used for small translation adjustments and rotation. Clicking on the yellow cube toggles between translation and rotation of the hand. To modify the finger bone rotations, click on one of the bone rotation gizmos.

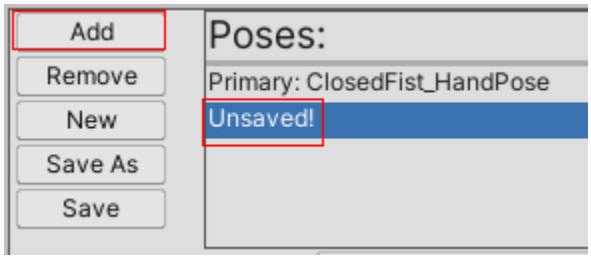


After modifying the bone rotations and hand position, the hand pose can be mirrored to the opposite hand by clicking *“Mirror”* on the hand that was modified. This will mirror the pose based on the HVR HandMirrorSettings on the avatar.



The pose will also need to be saved. To do so, click either “Save” or “Save As” and choose the location for the asset to be saved to and its name. Hand poses are stored under *“Assets > VORTEX_COVID > HandPoses”*.

To add an additional pose to the grab point, click *“Add”* and then click *“Unsaved!”* to start posing.



After adding any hand poses, the changes will need to be applied to the prefab for it to be saved on the object for future use.

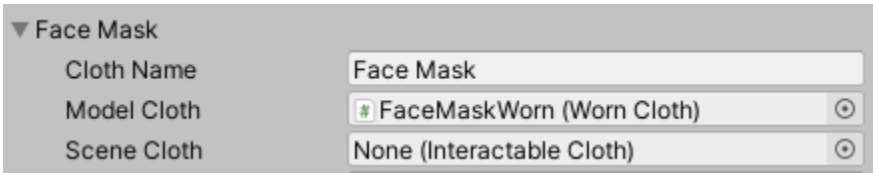
Adding PPE to an Avatar

Binding Objects to a Bone

To add a PPE object to an avatar, it needs to be added as a child to one of the avatar’s bones. The object that needs to be added is the variant prefab created previously. When placing the PPE on a body part, use the bone that the mesh is skinned to in that location.

Adding Cloth Control Scripts

On the avatar prefab, the *“InteractableClothController.cs”* script will need to be added. For each PPE object that is on the model, a clothing pair will need to be created in the array *“Clothing Pairs”*. The pair name will need to be the exact same as the name in the *Interactable Cloth* script added to the original prefab earlier. The cloth that is added to the model can then be inserted under the *“Model Cloth”* field. The *“Scene Cloth”* field can be left empty if the *Interactable Cloth* needs to be found when the avatar is instantiated in the scene at runtime.

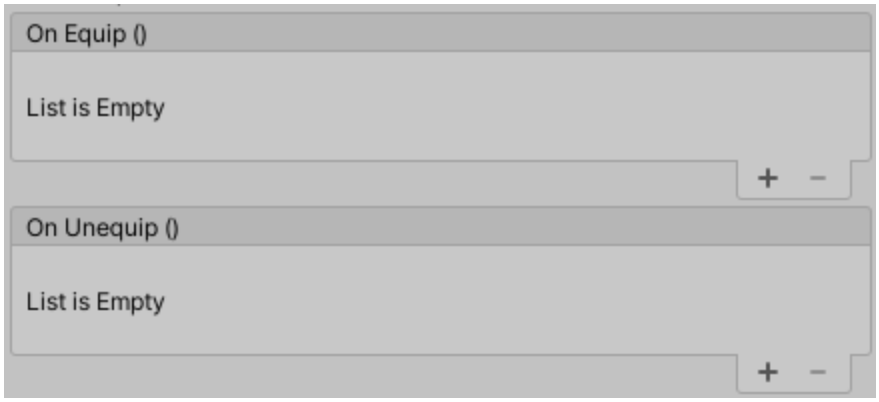


The distance and angle threshold will need to be modified to allow the user to easily and comfortably equip the PPE when grabbing is enabled. The distance threshold is the distance from the worn cloth that the scene cloth will need to be to equip, and it may change based on the scale of the avatar.

The *“Equip At Start”* variable determines whether the avatar will spawn with the worn cloth already equipped or not. If it is active, the scene cloth will deactivate and the worn cloth will activate when the avatar is instantiated. If it is disabled, the worn cloth will deactivate and the scene cloth will remain active.

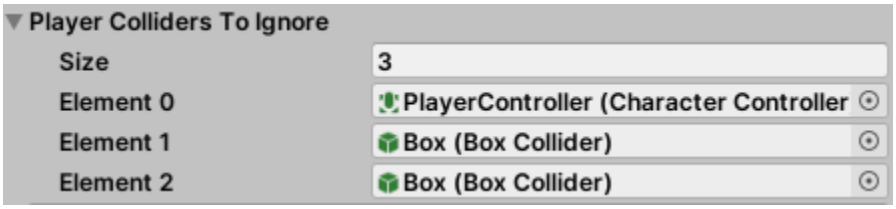
The object’s grabbable state can also be toggled using the *“Snap On Grab”* variable. If this is disabled, normal grabbing behavior will occur. If it is enabled, the object will become interactable instead of grabbable. When the user grabs the object, it will trigger the *Interacted* event and the cloth will equip instantly. When unequipped, the scene cloth will reappear where it was last located.

There are two pairs of events that trigger on this script when a cloth is equipped or unequipped. The first pair, *“OnEquip”* and *“OnUnequip”*, is object-specific and can be used if only one specific object needs to invoke a function.



The second pair of events trigger when any of the cloth objects are equipped or unequipped. The cloth pair is passed as a parameter to whatever function is called from these events. This is mainly used as a single point of communication to other scripts.

Player colliders may also prevent the cloth from equipping properly, so their collisions need to be ignored by the physics engine. The “Player Colliders To Ignore” array stores any colliders that are located on the HVR rig. Every collider contained within every cloth pair will ignore collisions with these colliders. The most important collider to ignore is located on the “PlayerController” HVR rig object. In the example below, two additional colliders are ignored since they follow the hands.



Preventing Cloth from Being Grabbed by a Hand while Equipped

If the PPE object needs to be equipped near a hand, it may be interacted with by the hand it is equipped to. To prevent this, add the *"IgnoreHandgrab.cs"* script to the worn variant prefab. In the *"Grab Colliders"* array, several colliders will need to be added from the HVR rig hand that will be ignored. These colliders are shown in the image below.