

## Hand Tracking

Antes de nada, es necesario activar el HandTracking en las gafas.

En Configuración → Manos y.... → Seguimiento de manos ON.

Descargar e importar el paquete Oculus Integration.

Edit → Project Settings → XR Plug-in Management

Debe estar oculus seleccionado tanto en PC como Android

Edit → Project Settings → XR Plug-in Management → Oculus

StereoRenderingMode Multi Pass

TargetDevices Quest y Quest2 marcados

### **Copiar a la jerarquía el Prefab OVRCameraRig**

En el script OVRManager establecer TrackingOriginType a Floor Level

Añadir al OVRCameraRig, dentro de TrackingSpace → LeftHandAnchor y TrackingSpace → RightHandAnchor, el prefab OVRHandPrefab.

En el caso de la mano derecha ajustar los valores de OVRHand → HandType y OVRSkeleton → SkeletonType

Añadir al OVRCameraRig, dentro de TrackingSpace → LeftHandAnchor → LeftControllerAnchor y TrackingSpace → RightHandAnchor → RightControllerAnchor, el prefab OVRControllerPrefab.

En ambas manos, ajustar el valor de OVRControllerHelper → Controller a L Touch o R Touch, según corresponda.

### **Copiar a la jerarquía el Prefab InputOVR.**

Para hacer unas primeras pruebas ajustamos a 0 los siguientes valores, que originalmente están a 3. Estos valores no es necesario cambiarlos como parte del proceso de configuración del proyecto para trabajar con Oculus Integration. Lo hacemos solo para poder hacer pruebas antes de completar la configuración sin que aparezcan errores. Más adelante los volveremos poner a su valor original.

InputOVR → Hands → LeftHand → HandInteractorsLeft → MaxInterationsPerFrame

InputOVR → Hands → RightHand → HandInteractorsRight → MaxInterationsPerFrame

InputOVR → Controllers → LeftController → ControllerInteractors → MaxInterationsPerFrame

InputOVR → Controllers → RightController → ControllerInteractors → MaxInterationsPerFrame

En OVRCameraRig añadir un componente InteractionOVRCameraRig

En InputOVR → TrackingToWorldTrasformerOVR → CameraRigRef poner una referencia a OVRCameraRig.

En OVRCameraRig → LeftHandAnchor → OVRHandPrefab y OVRCameraRig → RightHandAnchor → OVRHandPrefab eliminar los siguientes scripts OVRSkeletonRenderer, OVRMesh, OVRMeshRenderer y SkinnedMeshRenderer.

[Solo quedan OVRHand y OVRSkeleton]

En InputOVR → Controllers → LeftController y RightController añadir el Prefab ControllerGrabInteractor.

En InputOVR → Controllers → LeftController → ControllerInteractors → InteractorGroup → Interactors y

InputOVR → Controllers → RightController → ControllerInteractors → InteractorGroup → Interactors ponemos referencias a cada uno de los ControllerGrabInteractor que acabamos de añadir en el paso anterior.

A su vez en cada uno de los ControllerGrabInteractor → ControllerRef ponemos la referencia a su objeto padre (InputOVR → Controllers → LeftController o InputOVR → Controllers → RightController, según a que mano corresponda)

Ya podemos poner otra vez a 3 los valores de

InputOVR → Controllers → LeftController → ControllerInteractors → MaxIterationsPerFrame  
InputOVR → Controllers → RightController → ControllerInteractors → MaxIterationsPerFrame

Si ahora hacemos una prueba y cogemos el mando, veremos que aparece una bola verde.

En InputOVR → Hands → LeftHand y InputOVR → Hands → RightHand añadimos el prefab HandGrabInteractor

En InputOVR → Hands → LeftHand → HandInteractorsLeft → InteractorGroup → Interactors y InputOVR → Hands → RightHand → HandInteractorsRight → InteractorGroup → Interactors añadimos referencias a los correspondientes HandGrabInteractor que acabamos de añadir en el paso anterior.

En InputOVR → Hands → LeftHand → HandGrabInteractor → HandRef y InputOVR → Hands → RightHand → HandGrabInteractor → HandRef ponemos referencias a sus objetos padre InputOVR → Hands → LeftHand y InputOVR → Hands → RightHand

Ya podemos poner otra vez a 3 los valores de

InputOVR → Hands → LeftHand → HandInteractorsLeft → MaxIterationsPerFrame  
InputOVR → Hands → RightHand → HandInteractorsRight → MaxIterationsPerFrame

Esto finaliza la configuración de las manos para ser trackeables e interactuar con objetos.

Lo siguiente es crear un objeto cualquiera, un cubo por ejemplo y añadirle los componentes necesarios para que se puedan agarrar. Ojo, hay varias referencias al propio objeto o sus componentes que hay que establecer, se mencionan algunas pero faltan otras.

## Un Rigidbody

### Grabbable (antes se llamaba Transformable)

#### GrabInteractable

Con esto ya se puede coger con los mandos, pero tenemos que arreglar el *script* Grabbable para que se pueda coger correctamente el objeto, ya que esta versión de Oculus Integration tiene lo que parece ser a todas luces un bug. Nuestra solución no es completa, pero nos permite hacer unas pruebas básicas.

#### HandGrabInteractable

Con esto ya debería de ser agarrable con la mano por trackeo.

El comportamiento por defecto de los objetos al soltarlos es “teletransportarse” al suelo. Esto no está funcionando así en esta versión de Oculus Integration, pero si lo arreglan, la forma de cambiar este comportamiento por otro más físico es añadir el componente PhysicsGrabbable

Otros componentes interesantes:

OneGrabTranslateTransformer. Sirve para limitar los movimientos en los ejes deseados (para un cajón o una puerta corredera, por ejemplo). Hay que establecer una referencia a él en el script Grabbable

TwoGrabFreeTransformer. Sirve para agarrar un objeto con las dos manos (si se agarra con solo una no hay ningún efecto) y, en teoría, para escalarlo según la distancia entre las manos, pero esto último no funciona. También hay que establecer una referencia a él en el script Grabbable

OneGrabRotateTransformer. Sirve para hacer rotar un elemento respecto a un punto (un Transform) y en un eje determinado. También hay que establecer una referencia a él en el script Grabbable

Reconocimiento de gestos.

ShapeRecognizerActiveState, HandRef, ActiveStateSelector y SelectorUnityEventWrapper. En conjunto permiten reconocer gestos de la mano y asociar acciones a ellos.

En ShapeRecognizerActiveState se establece una referencia al objeto que contiene la pose y otra al objeto InputOVR → Hands → LeftHand o InputOVR → Hands → RightHand según cual sea la mano que queramos que efectúe el gesto. Si queremos hacerlo con las dos tenemos que duplicar el objeto completo con todos sus componentes. La pose vale la misma.

En HandRef ponemos la misma referencia al objeto InputOVR → Hands → LeftHand o En InputOVR → Hands → RightHand que en ShapeRecognizerActiveState.

En ActiveStateSelector ponemos una referencia si mismo (!)

En SelectorUnityEventWrapper ponemos las acciones asociadas a los eventos de Selection y Unselected asociados al gesto, a la manera ya conocida de Unity.

Para crear una pose que represente un gesto a reconocer hay que crear un objeto Oculus → Interaction → SDK → PoseDetection → Shape.