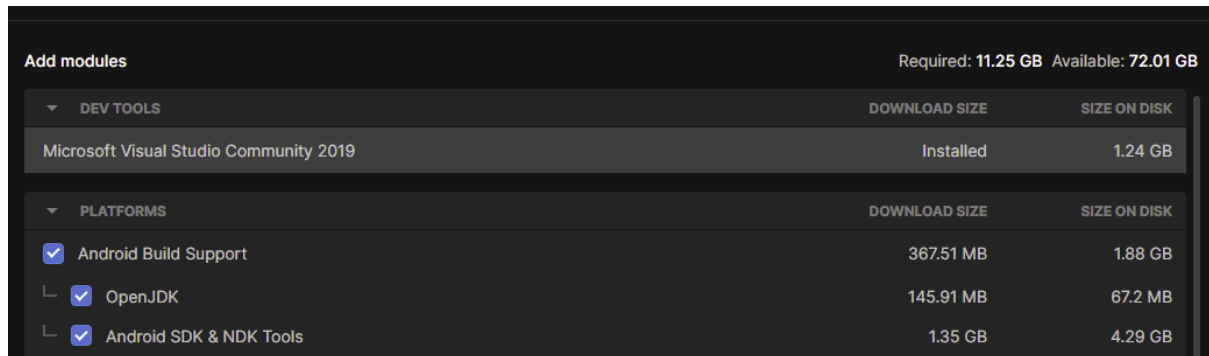
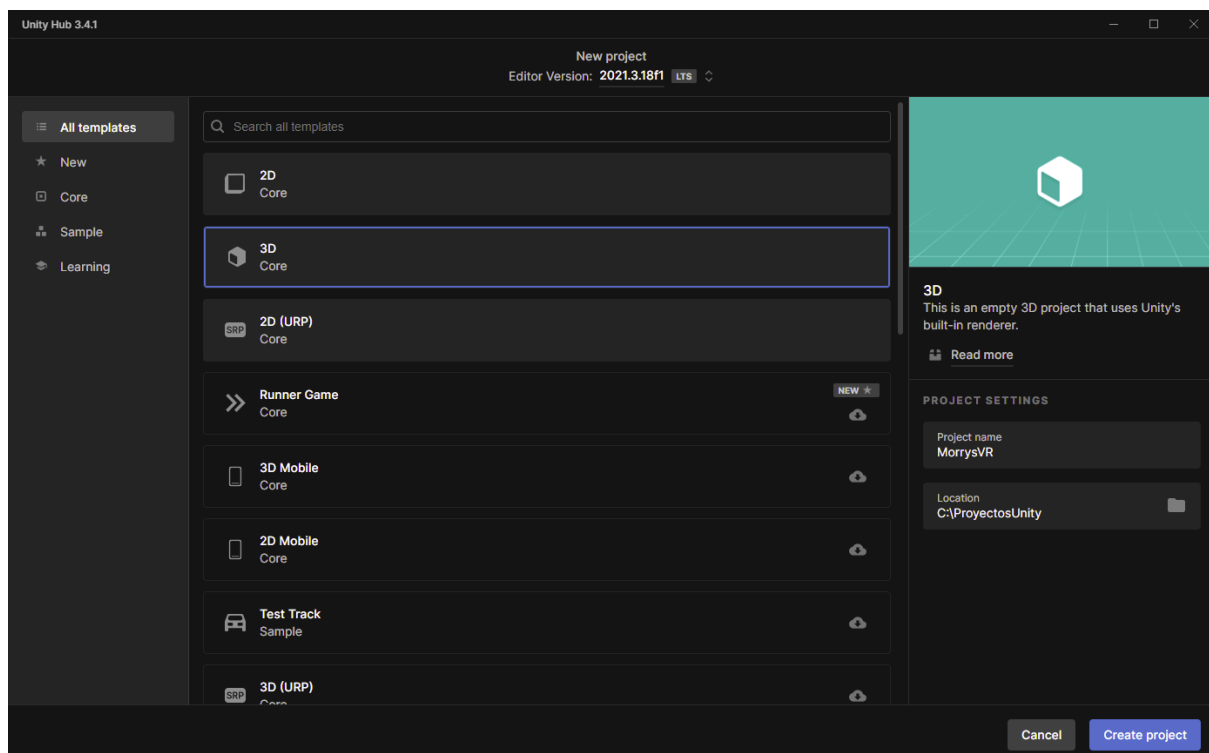


# Tutorial instalación y configuración Unity VR

En primer lugar vamos a necesitar instalar Unity Hub y dentro del mismo la versión del editor 2021.3.18f1 ya que en el caso de este proyecto, usaremos esta versión. Es muy importante que al hacer click en instalar, marquemos el editor de Visual Studio (si queremos utilizarlo en lugar de Rider como yo he hecho en este proyecto), y es imprescindible marcar las opciones de Android Build Support como se ve en la siguiente imagen ya que nuestra aplicación la construiremos para Android.



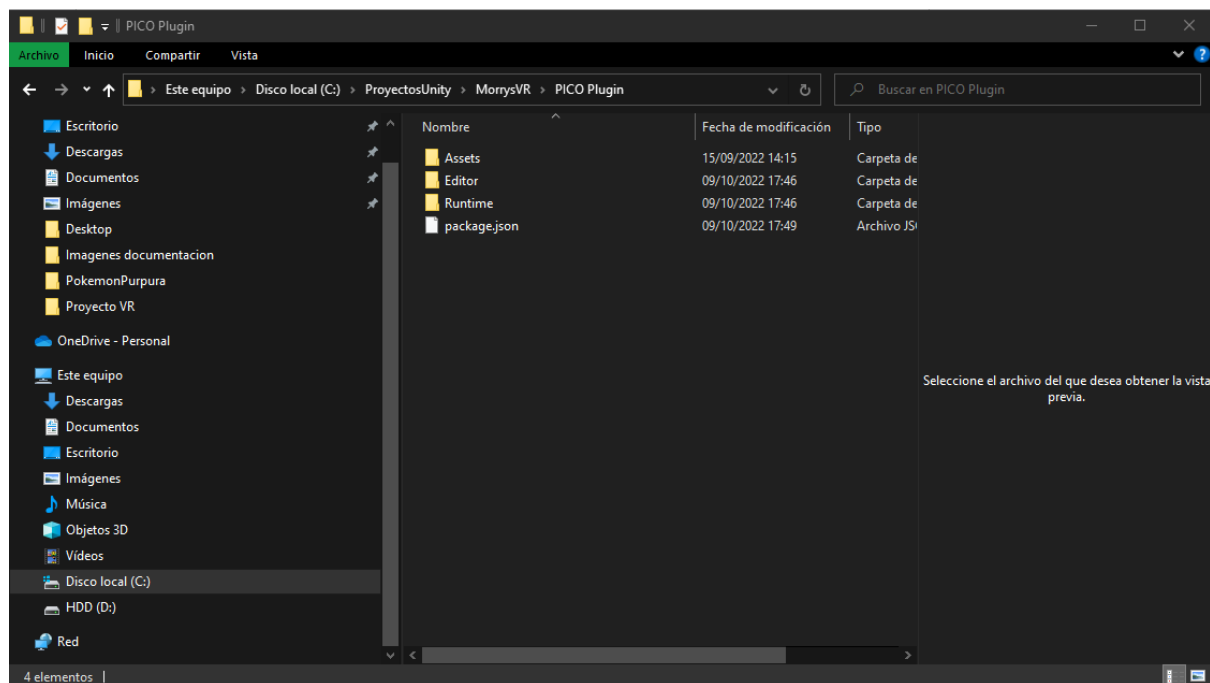
Con esto estamos preparados para comenzar con la creación del proyecto. Hacemos click en nuevo proyecto y seleccionamos la plantilla de 3D. Aunque existe la plantilla de VR, al menos en esta versión nos resultará más sencillo hacer la configuración del proyecto partiendo desde una plantilla de proyecto 3D.



Antes de abrir el proyecto es necesario descargar un plugin necesario para que funcione el dispositivo VR en el cual estamos ejecutando la aplicación: PICO 4 VR. Para ello accedemos al siguiente link:

<https://developer-global.pico-interactive.com/sdk?deviceId=1&platformId=1&itemId=7153484514247917573>

Pulsamos download y se nos descargará un archivo comprimido el cual podemos guardar en cualquier carpeta de nuestro equipo aunque para mayor comodidad, dentro de la carpeta de mi proyecto, crearé una carpeta llamada PICO Plugin y extraeré el contenido descargado en la carpeta. De esta forma si necesito abrir el proyecto en otro equipo, ya tendré el plugin descargado. La carpeta creada debería verse así:

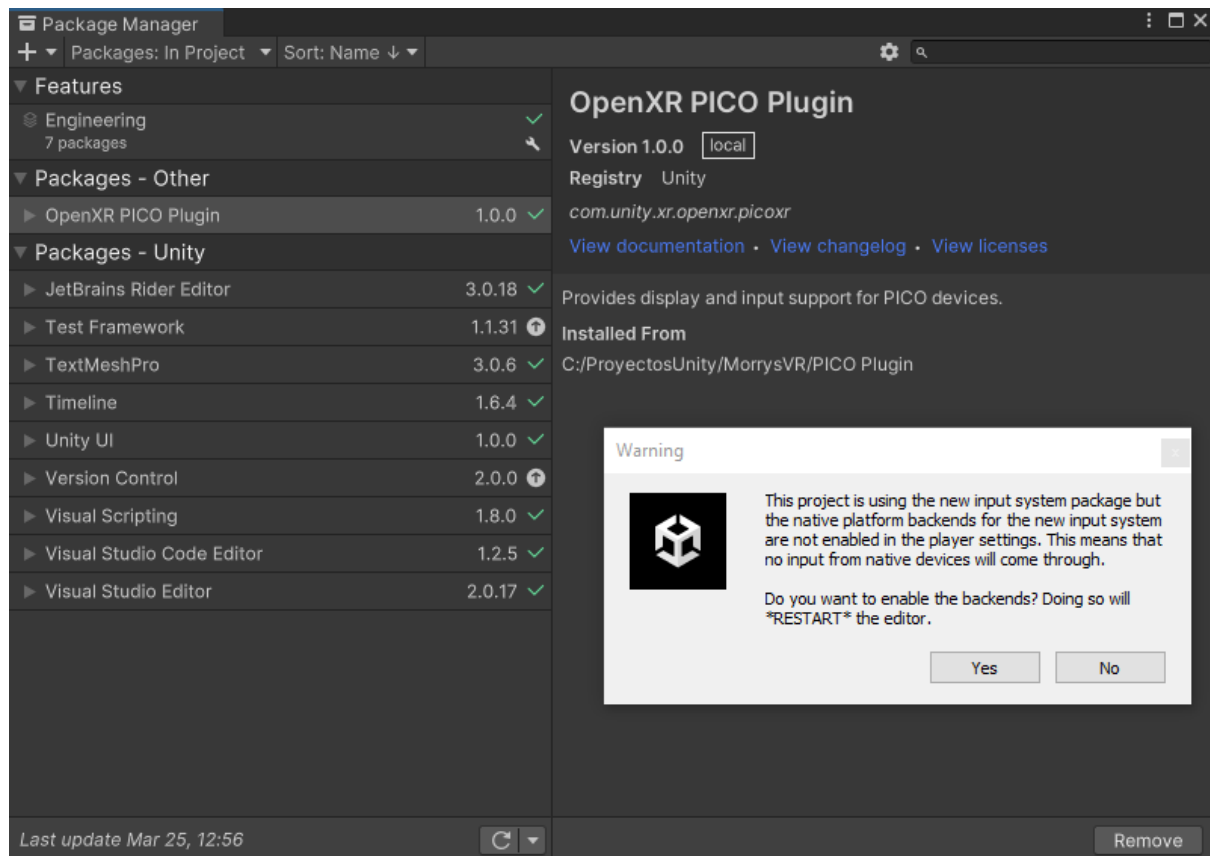


Ya podemos acceder al proyecto. Si queremos usar Rider o cualquier otro editor de código vamos a modificarlo antes de empezar. Para ello vamos a Edit->Preferences ->External Tools y en External Script Editor nos aparecerán los que tenemos instalados. Marcamos el que queramos usar. En mi caso Rider.

Ahora sí podremos empezar la configuración. La dividiremos en pasos para que resulte más sencillo.

1. Window->Package Manager. Donde pone Packages: In project, pulsamos y cambiamos a Unity Registry. Hecho esto, en la lista vamos abajo del todo e instalamos XR Plugin Management. Ahora pulsamos “+” arriba a la derecha de la

ventana y seleccionamos “Add package from disk”. Ahora vamos a nuestra carpeta del proyecto y dentro de PICO Plugin, hacemos doble click en package.json.



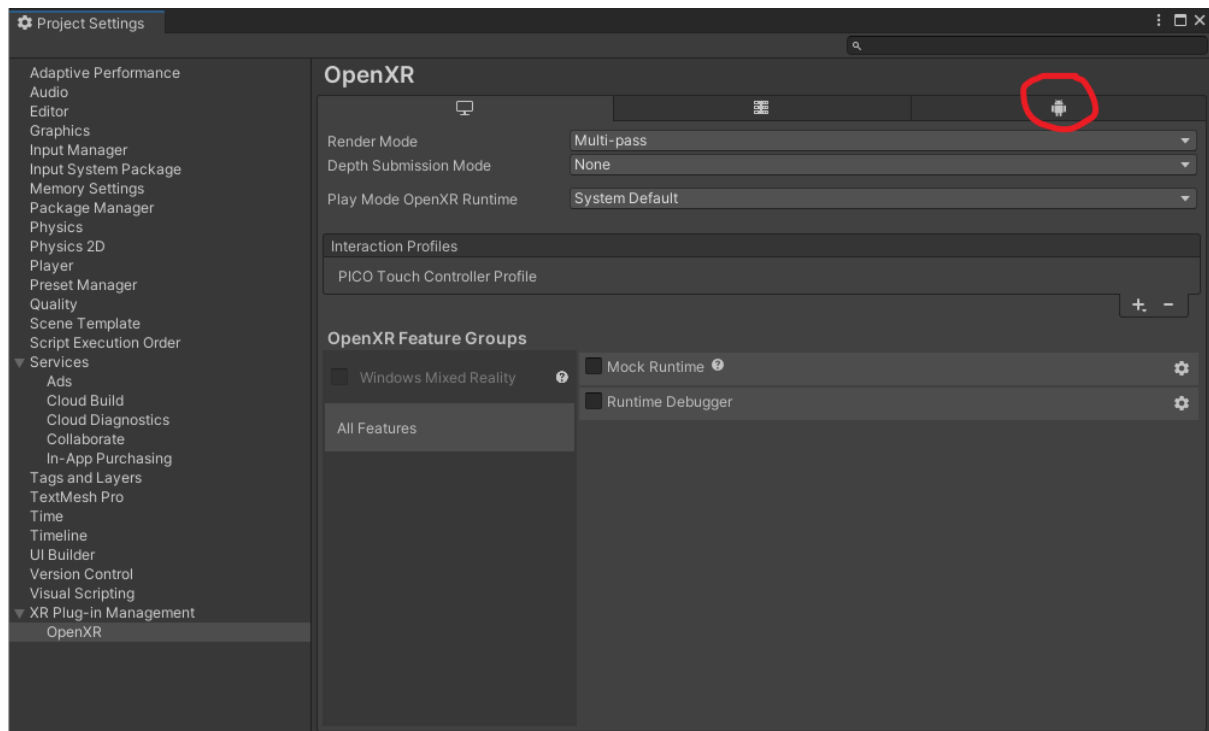
Si todo ha ido bien, Unity reconocerá el package y nos pedirá reiniciarse. Aceptamos y esperamos a que se vuelva a abrir.

2. Edit->Project Settings->XR Plugin Management. Pulsamos en Install XR Plugin Management si aparece. En caso contrario, puedes omitir este paso.

3. Dentro de XR Plugin Management, marcamos la opción OpenXR y aparecerá un simbolo amarillo de alerta junto a la opción que acabamos de marcar. Hacemos click en ella. Nos aparecerán 2 avisos. Uno nos avisa de que una de las opciones que estamos usando será deprecada en versiones futuras. Es FUNDAMENTAL que ignoremos este aviso. De lo contrario, la aplicación no funcionará al instalarla en el dispositivo VR. En la otra nos avisará de que necesitamos añadir un perfil de interacción. Hacemos click en Edit.

Dentro de esta ventana cambiamos el Render mode a Multi-pass y en la lista de Interaction Profiles añadimos PICO Touch Controller Profile.

Ahora hacemos click en la pestaña de Android que solo aparecerá si al instalar la versión del editor, marcamos la instalación de las herramientas de Android.

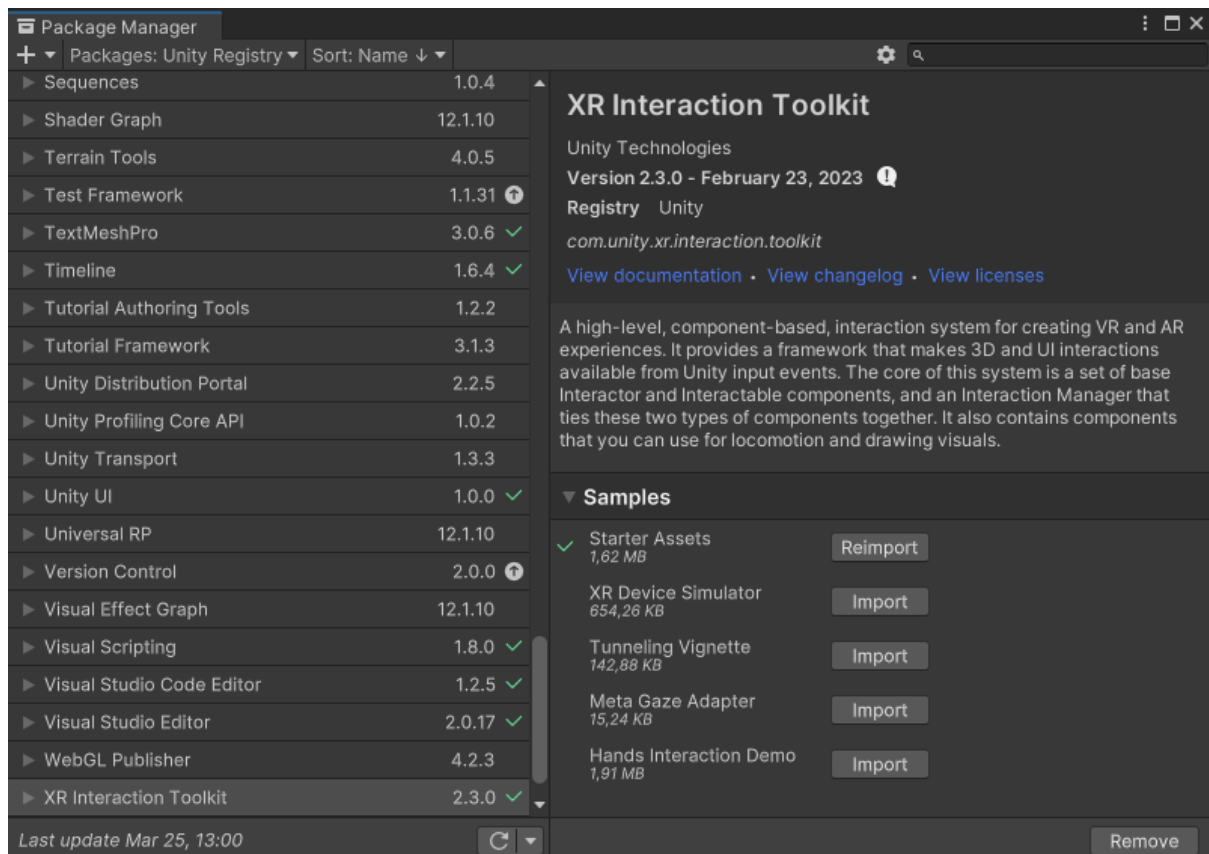


Dentro de esta pestaña, cambiamos el Render mode a Multi-pass, en interaction profiles, añadimos PICO Touch Controller Profile y abajo, marcamos la opción de PICO Support.

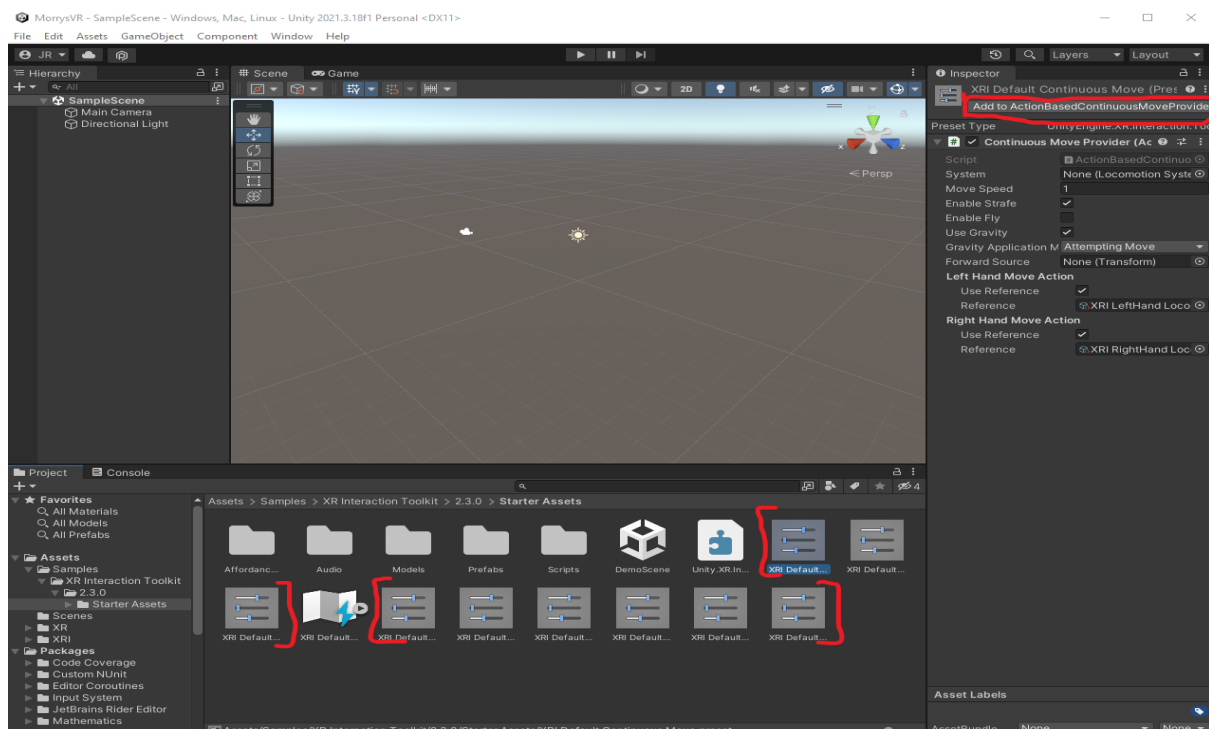
Por último volvemos a pulsar en XR Plug-in Management y dentro de la pestaña de Android que se encuentra en el mismo sitio que en la imagen anterior, marcamos la opción de OpenXR si no está marcada.

4. Window->Package Manager. Pulsamos “+” arriba a la derecha de la ventana y seleccionamos “Add package from git URL”, escribimos `com.unity.xr.interaction.toolkit` y pulsamos intro. Aquí nos aparecerá un aviso pidiendo que hagamos un backup pero no será necesario. Aceptamos y continuamos.

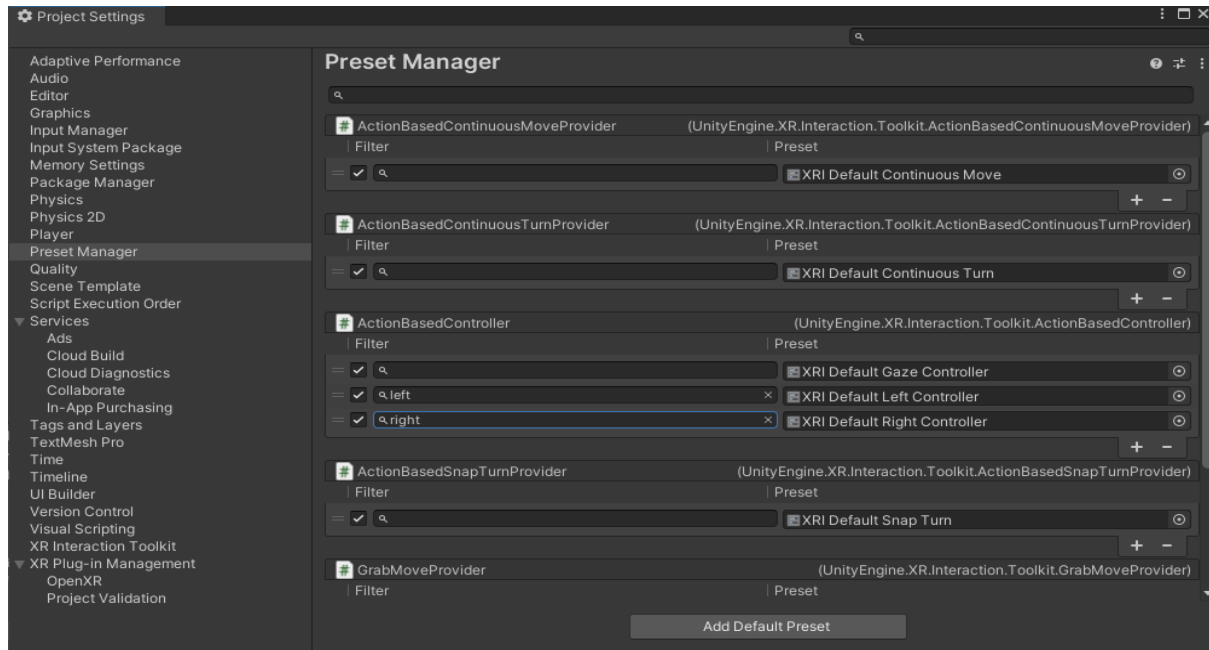
En la lista de paquetes a la izquierda, clickamos XR Interaction Toolkit y en el detalle nos aparecerá “Samples”. Pulsamos el desplegable y de todas las opciones solo importamos Starter Assets. Gracias a esto, se añadirán a nuestro proyecto varias “Plantillas” con gran parte de la lógica que usaremos en nuestra aplicación.



5. En el navegador de archivos de nuestro proyecto, vamos a Assets->Samples->XR Interaction Toolkit->2.3.0->Starter Assets y dentro de esta carpeta encontraremos varios elementos. Hacemos click en cada uno de los presets y clickamos en Añadir como se muestra en la siguiente imagen:



6. Por último vamos a Edit->Project Settings->Preset Manager. En actionBasedController aplicamos los filtros left y right a las respectivas manos como vemos en la imagen.

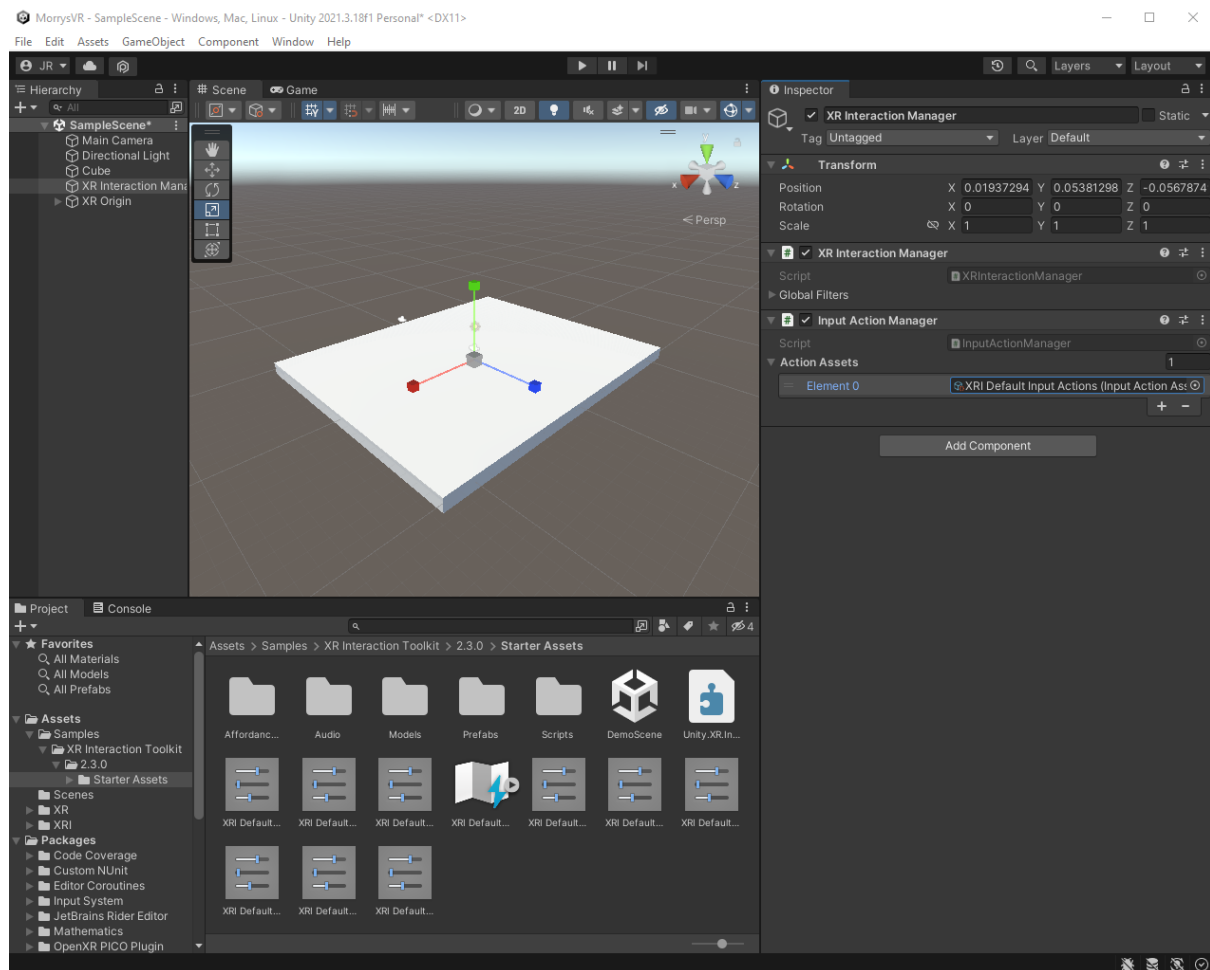


Con esto acabaríamos la configuración del proyecto. Ahora vamos a crear un proyecto básico que sirva como punto de partida, añadiendo elementos a nuestra escena.

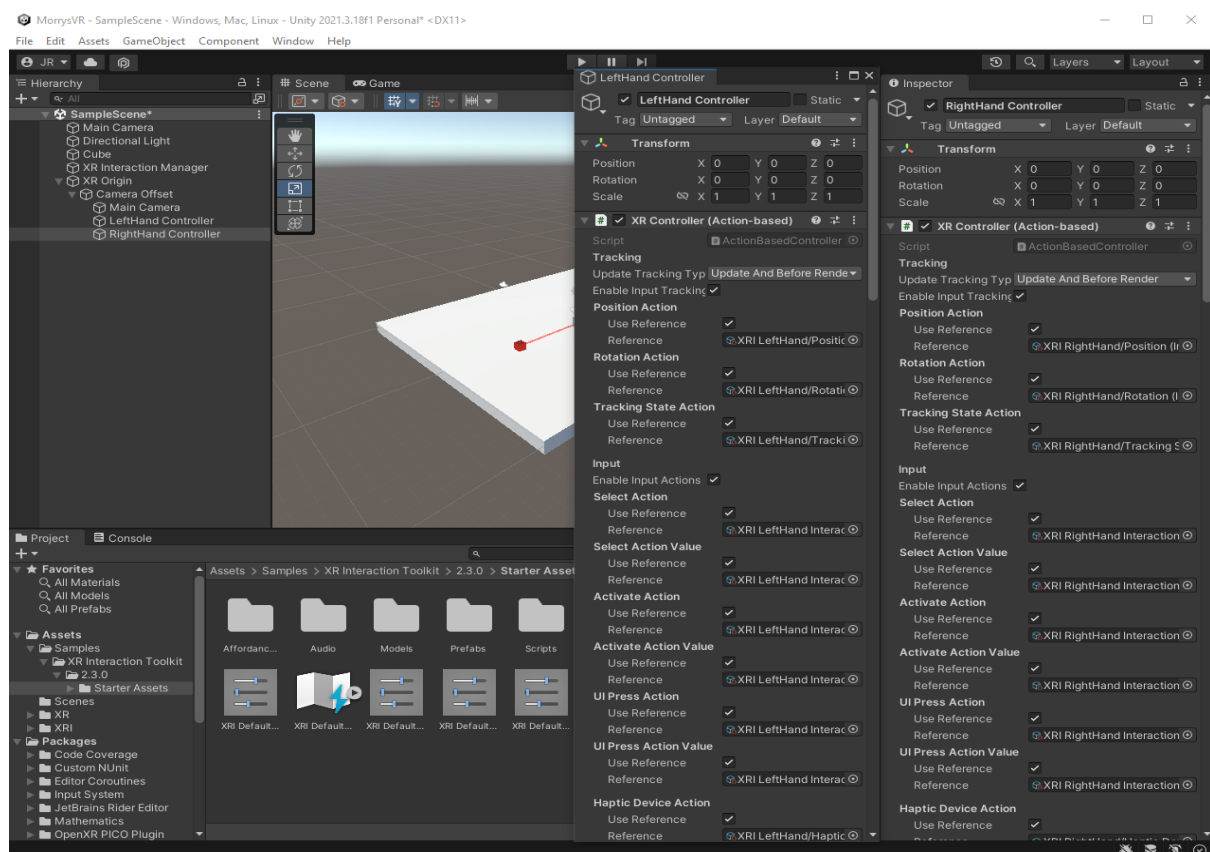
En primer lugar crearemos un cubo 3D que estiraremos a conveniencia y usaremos como suelo.

Una vez creado, crearemos un XR Origin que es el punto donde “apareceremos” como jugador al entrar en la aplicación.

Click derecho en la lista de elementos, XR->XR Origin (VR). Veremos que en la lista se han creado 2 componentes. XR Origin y XR Interaction Manager. Dentro del segundo, añadimos un componente Input Action Manager y en la lista añadimos el XRI Default Input Actions



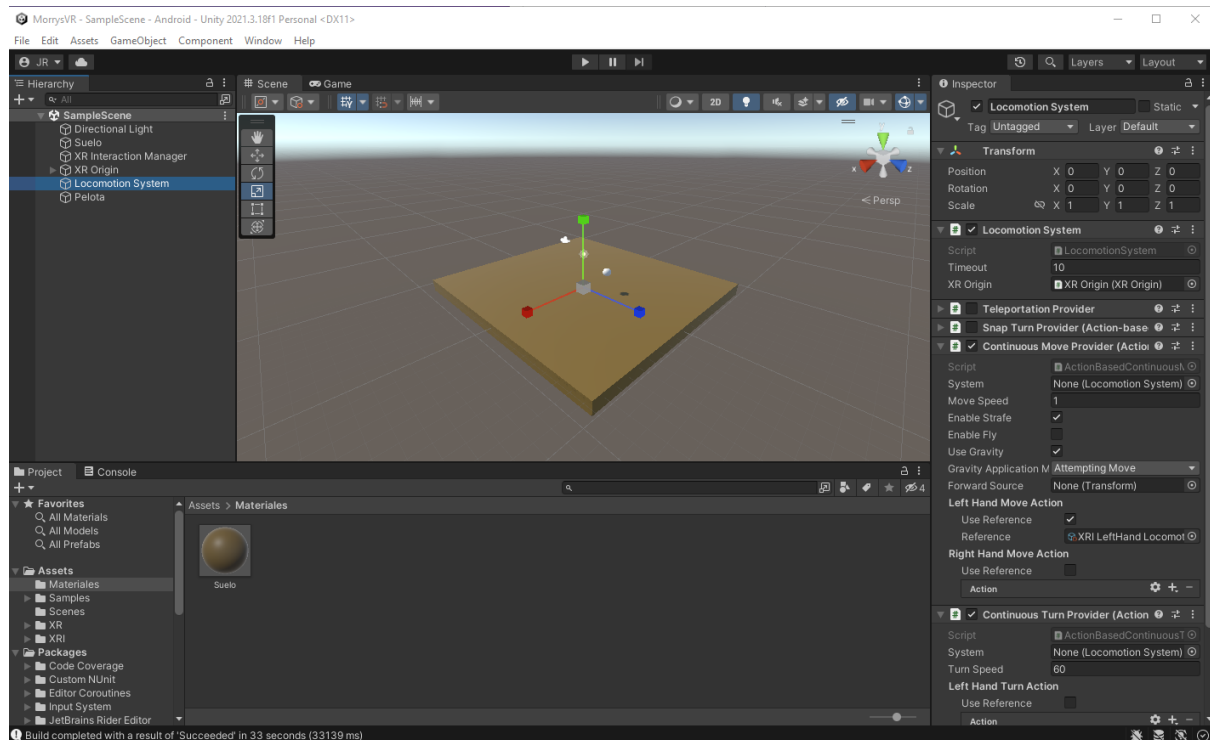
En los elementos hijos de XR Origin acabaremos encontrando LeftHandController y RightHandController. Nos tenemos que asegurar de que en cada una de sus acciones tenga asignados los valores de mano derecha para la derecha y de izquierda para la izquierda como se muestra en la imagen.



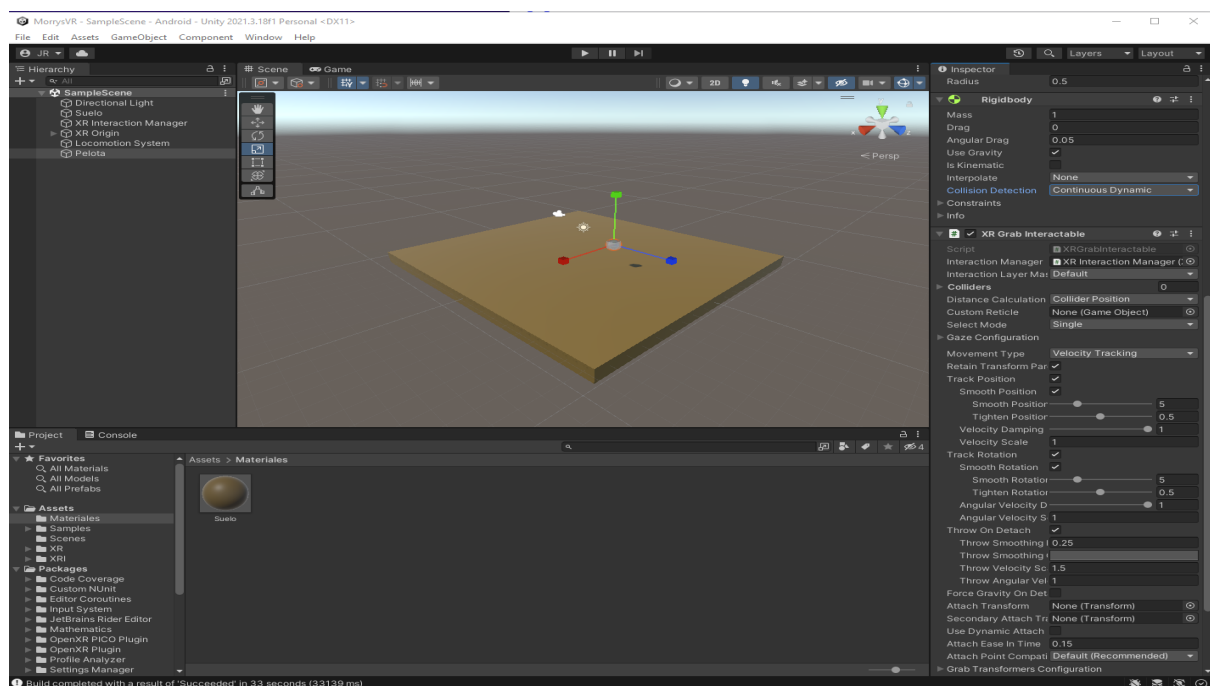
Hecho esto solo nos faltan dos pasos: poder movernos y poder interactuar con objetos.

Para el primero haremos click derecho en la lista de elementos y seleccionaremos XR->Locomotion System (Action Based). Tendrá los elementos Teleportation provider y Snap Turn provider. Desactivamos ambos y añadimos los elementos Continuous Move Provider y Continuous Turn Provider. En ellos, asignaremos el movimiento a la mano izquierda y el giro a la derecha ajustándolos a los controles habituales de movimiento. Para ello en cada uno debemos desactivar "Use reference" de la Mano contraria como se ve en la imagen. Además, dentro de cada mano desactivamos la opción de Anchor control.





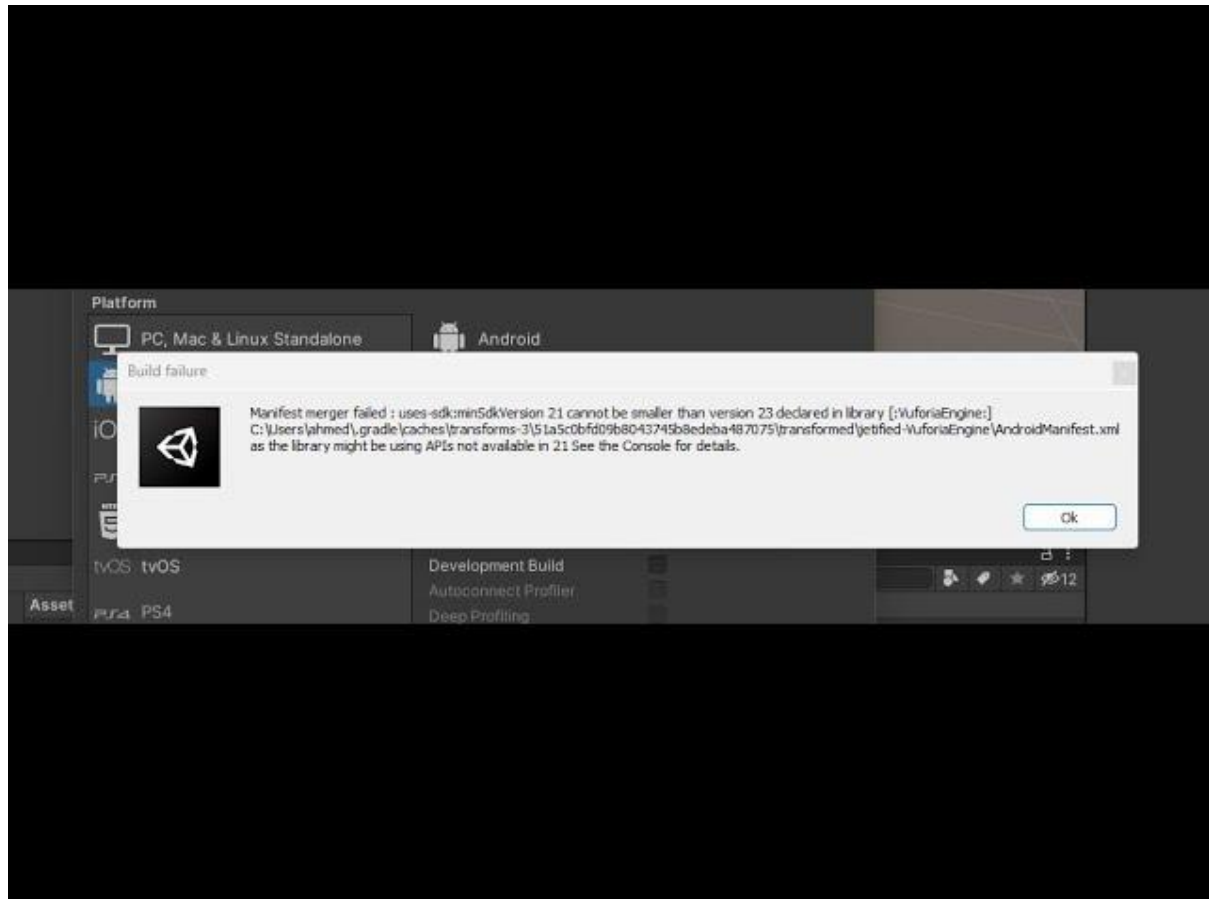
Por último vamos a añadir un objeto con el que podamos interactuar. En este caso crearemos una esfera en 3D objects. Para poder interactuar con ella, le añadiremos el componente XR Grab Interactable. Como veremos se crea un collider y un rigidbody para darle físicas al objeto. En el rigidbody cambiamos Collision Detection a Continuous Dynamic. En el XR Grab Interactable, cambiamos movement type a Velocity tracking para que las físicas sean realistas al agarrar el objeto, y marcamos Smooth position y Smooth rotation para suavizar el movimiento de estos objetos.



Con esto, la aplicación sería totalmente funcional y es una buena base desde la que partir para este proyecto.

### **Posibles errores frecuentes.**

P1. Al construir la APK nos aparece un mensaje de error:



R. En este caso simplemente cambia la versión mínima del SDK a la segunda que indica el error en los player settings.

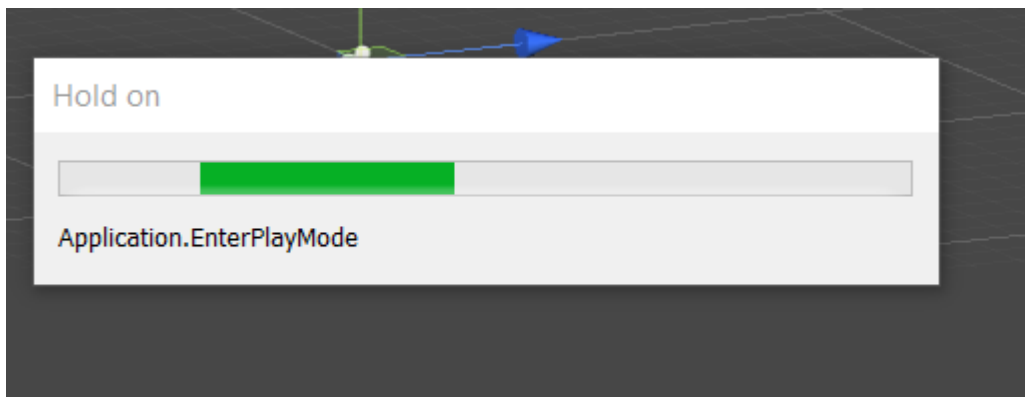
P2. Al instalar la APK en el dispositivo VR, la aplicación aparece con el icono de Unity (y no el de android) y al ejecutarla aparece una imagen estática y no “dentro” del escenario.

R. El PICO Plugin no está correctamente instalado. Repasa el procedimiento anterior.

P3. Al instalar la APK en las gafas, no puedo mover ni las manos ni mi personaje. Solo puedo mirar con la cabeza a mi alrededor.

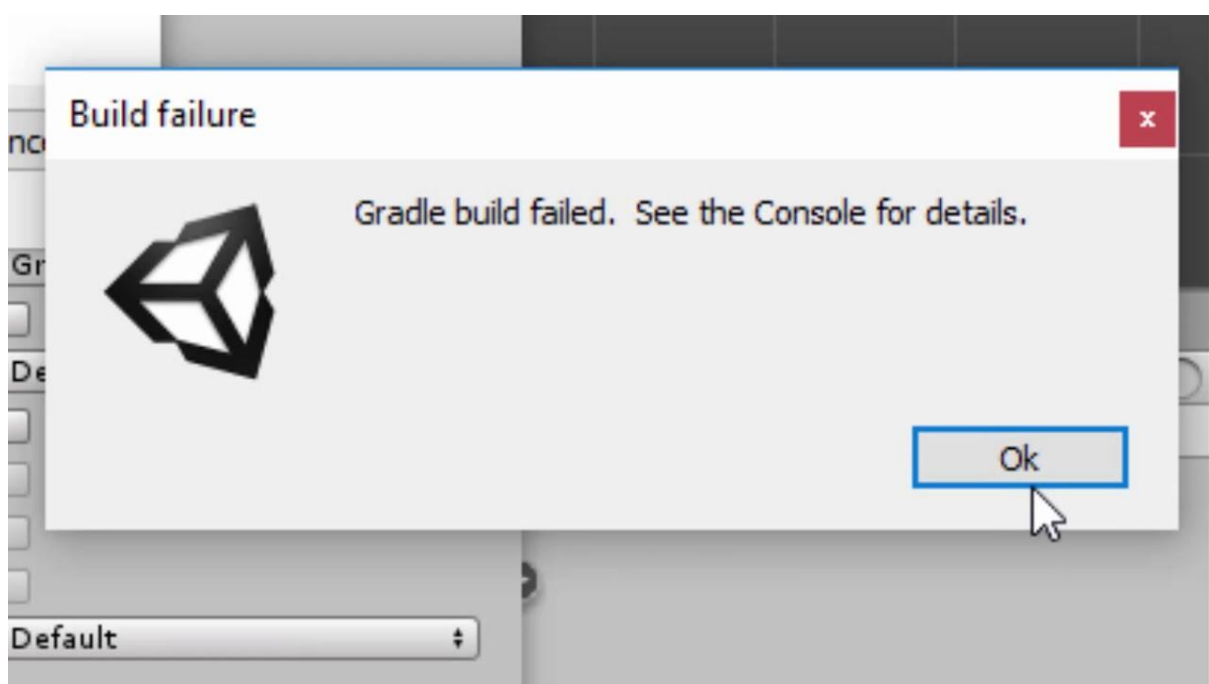
R. Has “Solucionado” el error que te marcaba como opcional en el paso 3. A día de hoy no sé cómo revertirlo y en principio tendrías que volver a una versión anterior del proyecto o crear uno nuevo.

P4. Al ejecutar el modo prueba de Unity con las gafas conectadas al ordenador, no se cargan en las gafas y en el editor de Unity aparece una barra cargando como esta que nunca termina.



R. Actualmente parece tratarse de un error de Unity sin resolver y aparentemente la única solución es esperar. Al cabo de unos días parece solucionarse solo. Este error no solo afecta a tu proyecto si no que te impide el uso del modo de pruebas en cualquier proyecto en tu equipo. Igualmente esto no impide la construcción de la APK y ejecutarlo directamente en las gafas.

P5. Al intentar construir la aplicación me aparece el siguiente error:



R. En este caso, la solución depende completamente de la información que nos proporcione la consola.