

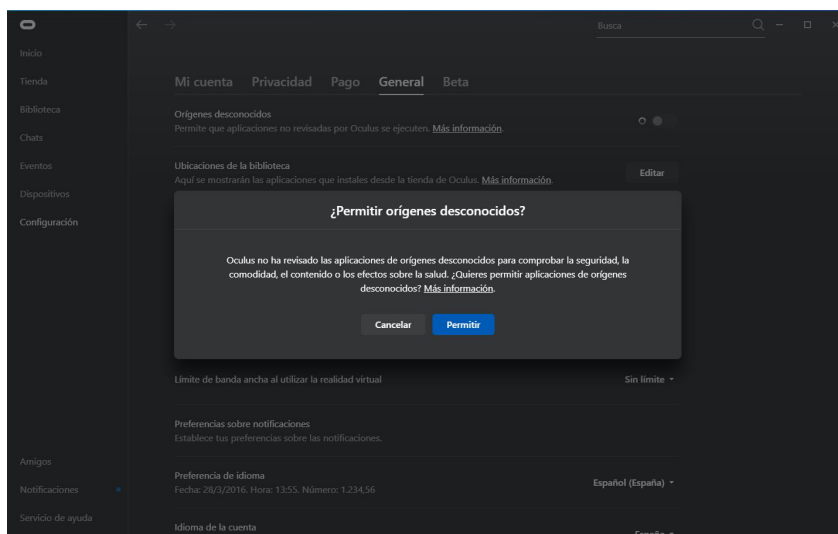
Pasos previos a la primera clase

Para asegurar una experiencia fluida y productiva durante nuestra primera clase, es crucial que completes los siguientes pasos previos, sobre todo si planeas usar tu ordenador personal. Esto nos permitirá aprovechar al máximo el tiempo en clase, enfocándonos directamente en la práctica guiada

Práctica guiada

1. Habilitar el modo de desarrollo en el software de Oculus

Para empezar hay que configurar el software de Oculus para que permita lanzar aplicaciones no registradas (como las que desarrollemos nosotros). Para eso, abrimos el software de Oculus y seleccionamos: Configuración → General → Unknown Sources = Allow.



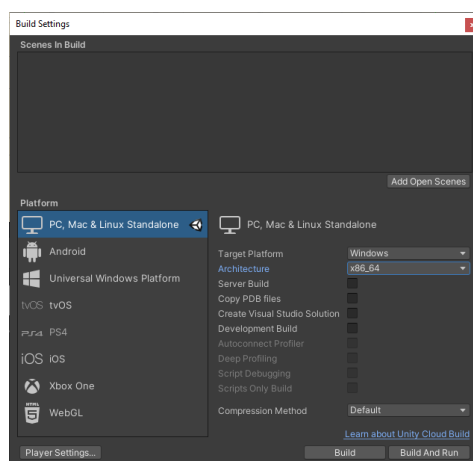
2. Crear nuevo proyecto y cargar escena de ejemplo en Unity

Para esta práctica se recomienda la versión de Unity 3D- 2021.3.1f1 LTS o superior. En los ordenadores de clase está instalada la versión 2022.3.1f1 y yo estoy utilizando 2022.3.4f1. Crea un nuevo proyecto 3D y abre la escena de ejemplo que se proporciona en Aula Virtual, así como en esta carpeta de Google Drive:

https://drive.google.com/drive/folders/1F4C1VppOTom1eJ4IVMCjD-LI-LCn4A3s?usp=drive_link

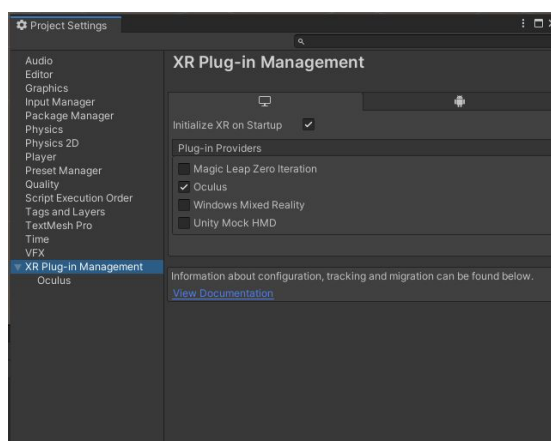
(Assets → Import Package → Custom Package y seleccionar el archivo .unitypackage.). Esta escena contiene simplemente un plano para el suelo, un cubo grande que hace de mesa y un par de cubos pequeños con los que podremos interactuar.

IMPORTANTE: Asegúrate de que el proyecto está configurado para compilar sobre una plataforma de 64 bits. En las últimas versiones de Unity esto es así por defecto (en Build Settings estará la arquitectura Intel 64-bit), pero en versiones anteriores habrá que seleccionar la arquitectura X86_64 en los Build Settings del proyecto.



3. Habilitar el XR Plug-in de Oculus

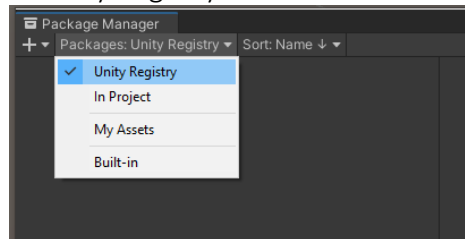
Para habilitar el desarrollo para Oculus en Unity, vamos a: Edit→Project Settings→XR Plugin Management. Instalar el Manager y marcar la casilla para **Oculus**.



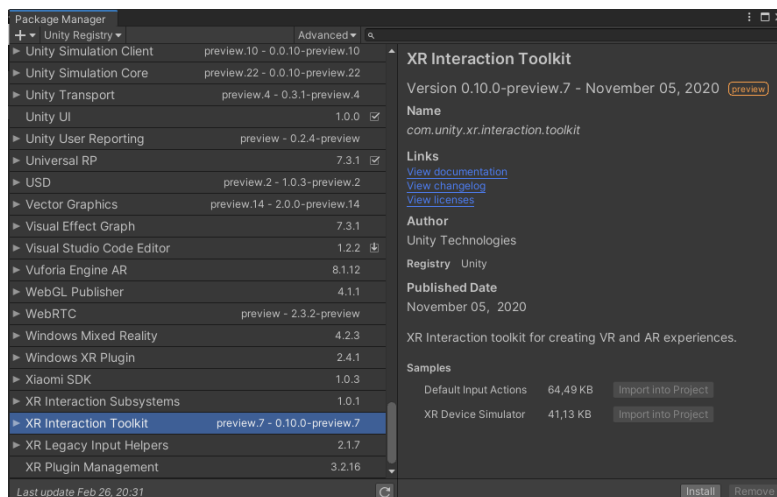
4. Instalar el paquete XR Interaction Toolkit

En esta sesión práctica, vamos a emplear el paquete de herramientas más popular y recomendado por Unity: XR Interaction Toolkit. Este conjunto de herramientas facilita la implementación de las tareas más comunes en XR y brinda una gran capacidad de adaptación, permitiéndote personalizar cualquier componente a través de la escritura de código (algo que podéis hacer si queréis pero sin mi ayuda).

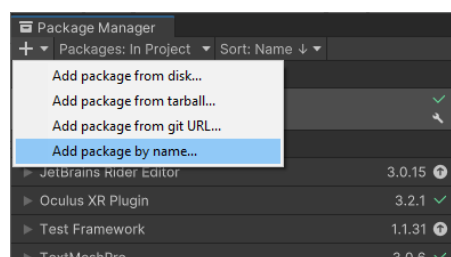
Ve a Window → Package Manager. Para ver los paquetes disponibles, tienes que seleccionar los Packages del Unity Registry:



Selecciona en la lista **XR Interaction Toolkit**, e instálalo:

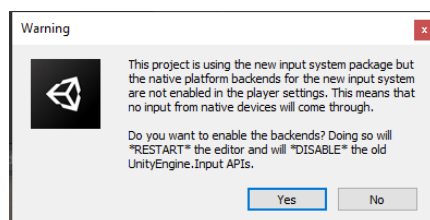


Si no aparece en la lista, puedes agregarlo directamente seleccionando la opción **Add Package by Name**:



Y escribiendo: **com.unity.xr.interaction.toolkit**

Si es la primera vez que se instala sale un aviso sobre el New Input System, que hay que aceptar y reiniciar Unity o el ordenador, si es necesario.



5. Input controllers – algunas consideraciones

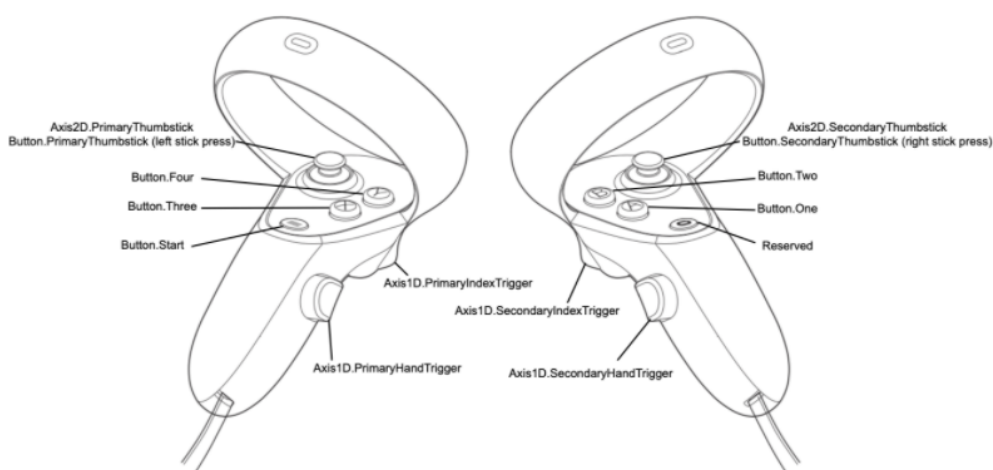
Existen dos sistemas principales para manejar los inputs en desarrollo de realidad virtual: el **Device-Based**, que asigna directamente acciones a botones específicos del hardware, que tiene su complejidad y limitaciones multiplataforma; y el **Action-Based**, que es preferido actualmente por su flexibilidad y facilidad para desarrollar aplicaciones compatibles con múltiples dispositivos al definir acciones de manera abstracta y luego mapear inputs de hardware a estas acciones.

- El sistema **Device-Based** se enfoca en los controles específicos del hardware, como los botones y sensores de los dispositivos de realidad virtual. Por ejemplo, programarías directamente qué sucede cuando se presiona el botón A en un mando de Quest. Esto significa que los scripts y configuraciones están estrechamente ligados a los dispositivos con los que estás trabajando, requiriendo que manejes cada controlador o dispositivo de entrada de manera individual. Aunque este método te da un control preciso sobre cómo se utilizan los dispositivos específicos en tu juego o aplicación, puede ser más complicado de manejar cuando quieres que tu proyecto funcione a través de diferentes tipos de hardware. Este enfoque es útil para experiencias altamente personalizadas que dependen de las características únicas de un dispositivo específico.

Mapeos genérico que facilita el desarrollo de soluciones multiplataforma al asignar un alias a cada control:

InputFeatureUsage	Feature type	Legacy input index (left controller/right controller)	WMR	Oculus	GearVR	Daydream	OpenVR (Full)	Vive	Oculus via OpenVR	WMR via OpenVR	Magic Leap
primary2DAxis	2D axis	[[1,2]/[4,5]]	Touchpad	Joystick	Touchpad	Touchpad	Trackpad/Joystick	Trackpad	Joystick	Joystick	Touchpad
trigger	Axis	[9/10]	Trigger	Trigger	Trigger	Trigger	Trigger	Trigger	Trigger	Trigger	Trigger
grip	Axis	[11/12]	Grip	Grip		Bumper	Grip	Grip	Grip	Grip	Bumper
secondary2DAxis	2D axis	[[17,18]/[19,20]]	Joystick								
secondary2DAxisClick	Button	[18/19]	Joystick - Click								
primaryButton	Button	[2/0]		[X/A] - Press		App	Primary	Primary (sandwich button)(1)	Primary (Y/B)	Menu	Menu
primaryTouch	Button	[12/10]		[X/A] - Touch							
secondaryButton	Button	[3/1]		[Y/B] - Press			Alternate		Alternate (X/A)		
secondaryTouch	Button	[13/11]		[Y/B] - Touch							
gripButton	Button	[4/5]	Grip - Press	Grip - Press			Grip - Press	Grip - Press	Grip - Press	Grip - Press	Bumper - Press
triggerButton	Button	[14/15]	Trigger - Press	Trigger - Press	Trigger - Press	Trigger - Press	Trigger - Press	Trigger - Press	Trigger - Touch	Trigger - Press	Trigger - Press
menuButton	Button	[6/7]	Menu	Start (left controller only)							
primary2DAxisClick	Button	[8/9]	Touchpad - Click	Thumbstick - Press	Touchpad - Press	Touchpad - Press	Trackpad/Joystick - Press	Trackpad - Press	Joystick - Press	Touchpad - Press	
primary2DAxisTouch	Button	[16/17]	Touchpad - Touch	Thumbstick - Touch	Touchpad - Touch	Touchpad - Touch	Trackpad/Joystick - Touch	Trackpad - Touch	Joystick - Touch	Touchpad - Touch	Touchpad - Touch
batteryLevel	Axis		Battery level								
userPresence	Button		User presence	User presence							

La siguiente figura muestra el mapeo para los mandos Oculus Touch:



El sistema **Action-Based** permite pensar en términos de acciones en vez de botones específicos. Imagina que tienes una acción llamada "Disparar"; puedes hacer que esta acción se active con un botón en un control o mediante un gesto con la mano. Al desarrollar tu juego o aplicación, te centras en detectar cuando la acción "Disparar" ocurre, sin preocuparte por el dispositivo específico que la activa. Además, Unity ofrece herramientas como un editor gráfico para facilitar la conexión entre acciones y controles físicos, aunque también puedes hacerlo por código. Vamos a usar este sistema en la práctica guiada porque nos permite simular interacciones sin necesidad de tener las Oculus conectadas físicamente, facilitando el desarrollo y la prueba de nuestras aplicaciones.

6. Descargas adicionales

Descargar la carpeta **Materiales** desde este enlace de Google Drive:
https://drive.google.com/drive/folders/1F4C1VppOTOm1eJ4IVMcjD-LI-LCn4A3s?usp=drive_link

Importar su contenido en el proyecto de Unity, arrastrando: **Audio, Modelos 3D, Texturas y Valencia Fountain** a la carpeta de Assets en la pestaña del proyecto o en la carpeta Assests. Las carpetas incluyen modelos 3D y texturas que se usarán durante el taller.

Se pueden utilizar modelos 3D (.fbx u .obj), imágenes de texturas (.jpg) y audio (.mp3 o .wav) propios.

Importar archivos: Busca, descarga e importa en el proyecto el Skybox **Free Night Sky** (Window > Asset Store > Search online > **Free Night Sky**). Importante tener cuenta e identificarse.