

«Talento Tech»

Desarrollo de Videojuegos

Unity 3D

Clase 15



Clase N° 15 | Mobile

Temario:

- Adaptación de controles para Mobile
- Building para Mobile

Objetivos de la clase

- Crear Joysticks en la UI.
- Usar Input System
- Configurar y Buildear para Mobile.

Introducción a los controles en Mobile

Cuando adaptamos un juego de PC a Mobile, una de las principales diferencias es la forma en que el jugador interactúa con el juego. Mientras que en PC se utilizan teclado y mouse (o un gamepad físico), en dispositivos móviles no hay botones físicos estándar, por lo que debemos implementar controles táctiles.

Métodos de Entrada en Dispositivos Móviles

Pantalla táctil (Touch Input)

- Es la forma más común de interacción en mobile.
- Se usa para detectar toques (`TouchPhase.Began`), mantener presionado (`TouchPhase.Stationary`), y soltar (`TouchPhase.Ended`).
- Puede usarse para botones virtuales, gestos de deslizamiento (swipe) o toques en la pantalla.

Acelerómetro (Gyroscope)

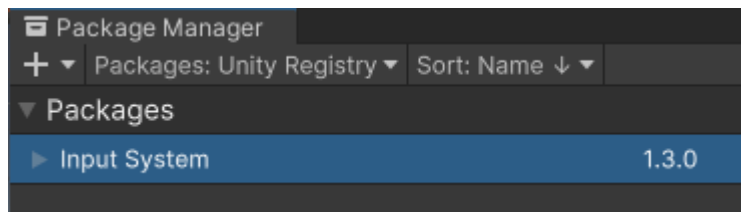
- Se usa para detectar el movimiento del dispositivo (inclinación, rotación).
- Se puede usar para juegos que requieren movimiento del dispositivo, como juegos de carreras o realidad aumentada.

Joystick Virtual

- Se implementa con elementos de UI (`Image`, `Button`, `EventTrigger`).
- Simula un joystick físico en la pantalla, permitiendo controlar el movimiento del personaje con el pulgar.
- Se usa en juegos de aventura, RPGs y shooters en tercera persona.

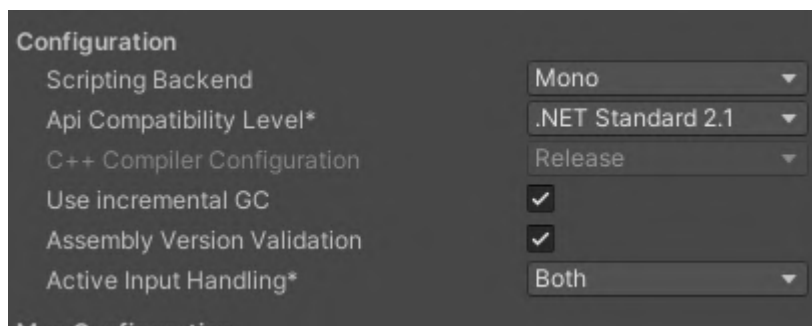
Configurar Input System

Ir **Window -> Package Manager -> Unity Registry -> Input Manager**



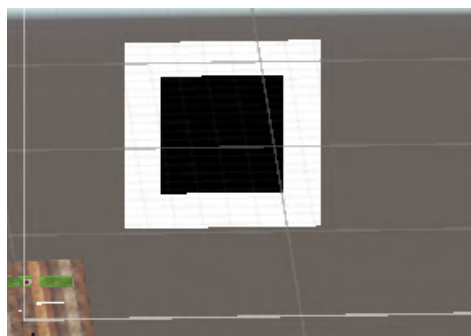
Antes de comenzar, asegúrate de que tu proyecto está usando el **Input System** en lugar del antiguo sistema de entrada.

1. Ve a **Edit -> Project Settings -> Player**.
2. En la pestaña **Other Settings**, busca la opción **Active Input Handling**.
3. Cámbiala a **Both** o **Input System Package (New)** y reinicia Unity si es necesario.



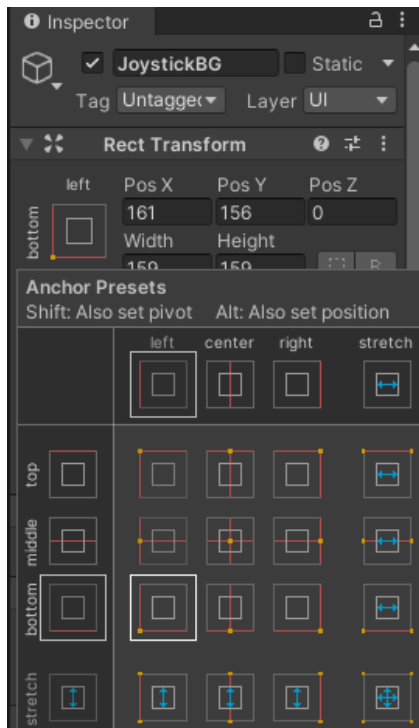
Crear la estructura en UI:

- Añade un **Canvas** con **Screen Space - Overlay**.
- Dentro del Canvas, crea un **Image** (para la base del joystick).
- Dentro de la base, crea otro **Image** (para el "handle" o palanca).



Recordá que podes elegir distintas imágenes, esta es cuadrada a base de ejemplo.

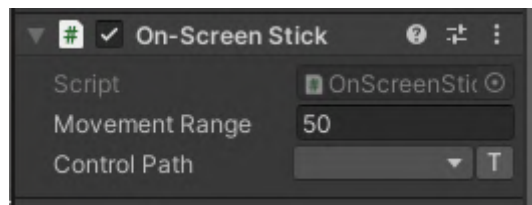
Configurar RectTransform:



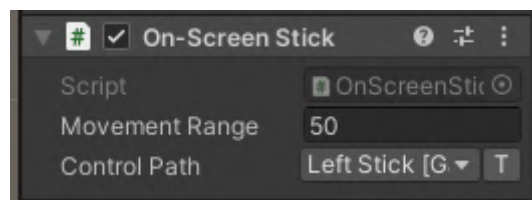
- Coloca el joystick en la esquina inferior izquierda o donde prefieras.
- Ajusta el tamaño y anclajes según el diseño de tu juego.

Agregar el component On-Screen Stick

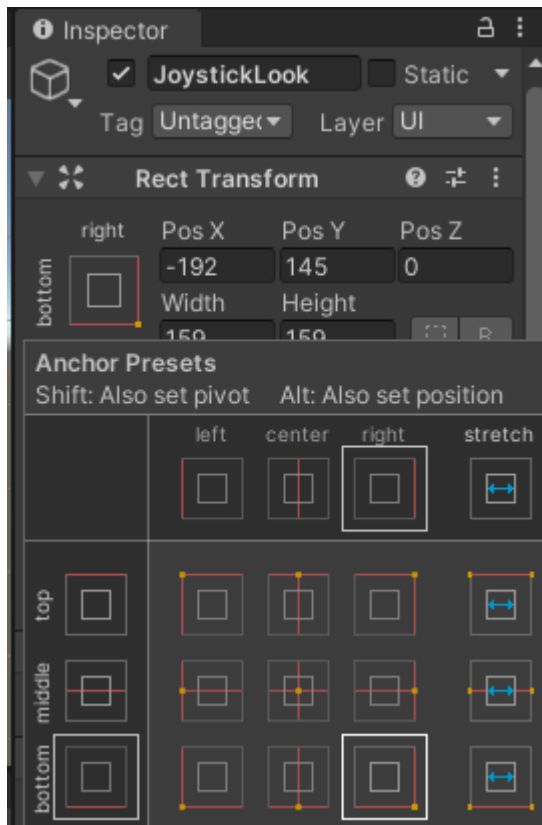
Al objeto que creaste agregale el componente On-Screen Stick. Esto permitirá que puedas moverlo como un joystick al darle play. La variable “Movement Range” nos dejará modificar el rango de movimiento del Joystick.



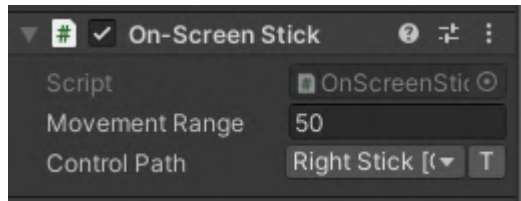
Configura el “**Control Path**” haciendo click y yendo a “**GamePad**” -> “**Left-Stick**”. Esto le permitirá reconocer el tipo de movimiento.



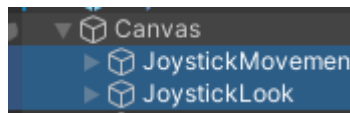
Joystick de cámara



Duplicaremos el Joystick que hicimos y lo situaremos a la derecha, modificando el **Anchor** y el **Control Path** a **Right Stick**. Este sera el control que usaremos para mover la cámara.



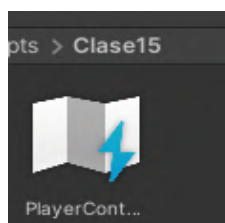
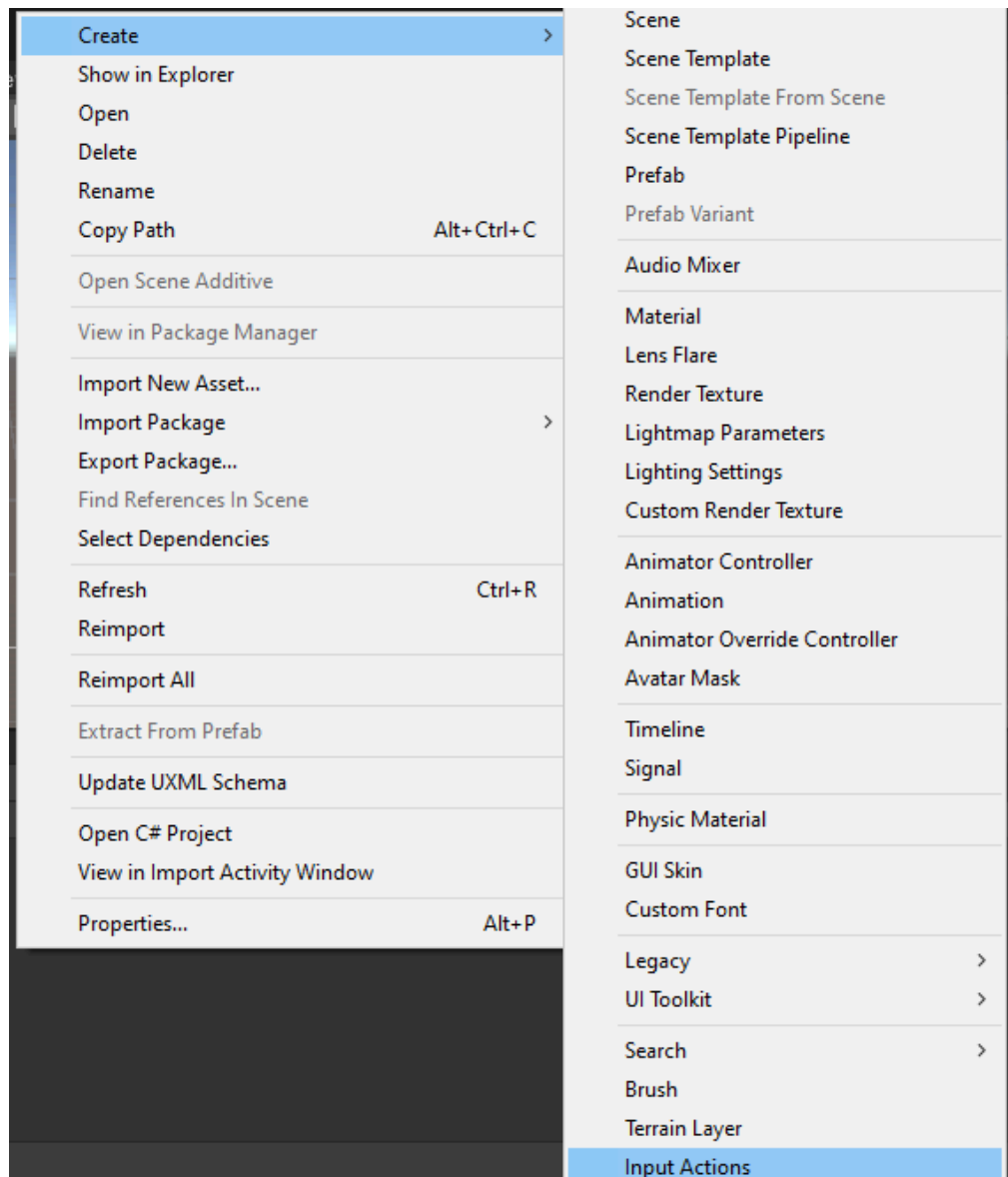
Así tendremos los 2 Joystick en nuestro Canvas.



Input Action

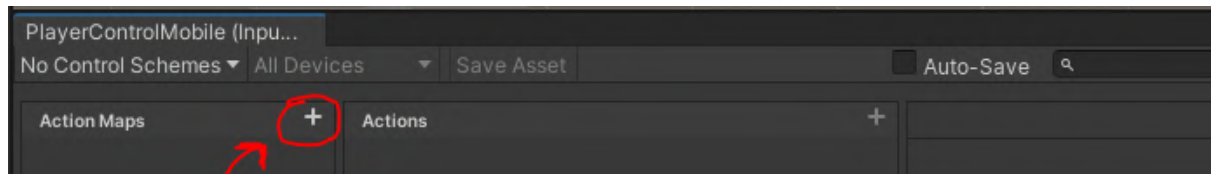
Crearemos la conexión para que estos elementos sean reconocido como Inputs.

En Project haremos **Click Derecho -> Create -> Input Action**

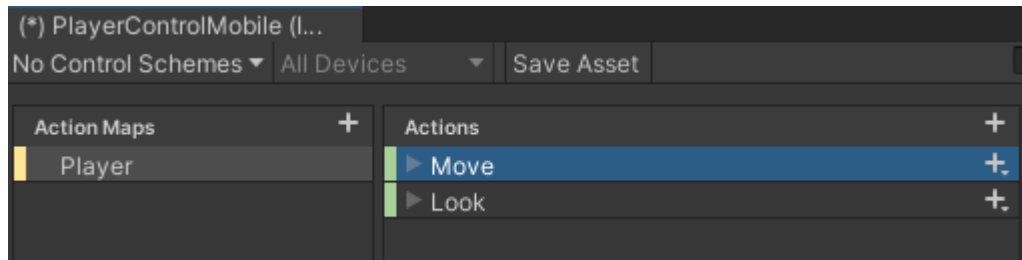


Y le pondremos de nombre algo como **“PlayerControlMobile”**

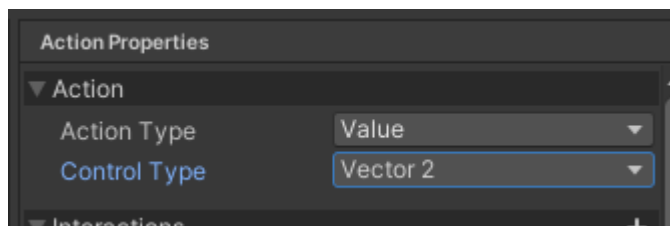
Lo abriremos y luego buscaremos la opción para agregar un nuevo **Action Map**



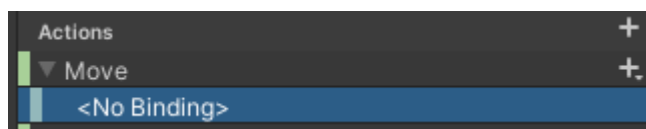
Crearemos uno y le agregaremos 2 acciones, "**Move**" y "**Look**"



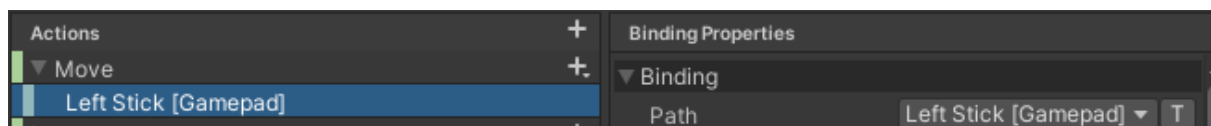
En Move iremos a Action Properties y cambiaremos las variable **Action Type = Value** y **Control Type = Vector 2**.



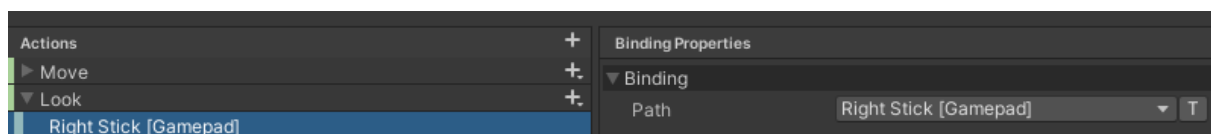
Expandimos el menu de Move ue nos dira "**No Binding**"



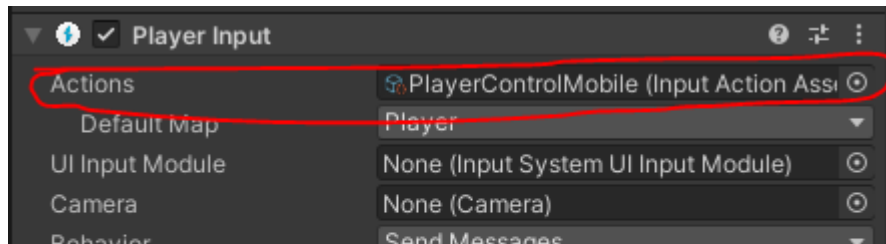
Iremos a **Binding Properties** -> **Path** -> **GamePad: LeftStick**



Repetiremos el proceso con "**Look**" pero le pondremos "**Right Stick**"



En nuestro Player pondremos el Component **“Player Input”** y le arrastraremos nuestra Action Map a la variable Actions.



Código

Para el código, modificaremos los **Scripts** de movimiento y de la camara para que acepte los Inputs nuevos.

Movimiento

Tendremos que colocar la librería **using UnityEngine.InputSystem;** y crearemos una variable de referencia de nuestro sistema de inputs y le asignaremos el componente en el **Start()**

```
PlayerInput pi;  
Mensaje de Unity | 0 referencias  
void Start()  
{  
    pi = GetComponent<PlayerInput>();  
}
```

Reemplazamos los valores de los inputs

```
void FixedUpdate() {  
    Vector2 input = pi.actions["Move"].ReadValue<Vector2>();  
    Vector3 movimientoMobile = new Vector3(input.x, 0, input.y);  
    movimientoMobile = movimientoMobile.normalized;  
    rb.velocity = new Vector3(movimientoMobile.x * moveSpeed,  
rb.velocity.y, movimientoMobile.z * moveSpeed); }
```

En la línea **“Vector2 input = pi.actions["Move"].ReadValue<Vector2>();”** Accedemos al Action **“Move”** que creamos y leemos el valor del Vector2 que se genera constantemente.

Look

Accedemos al código de la cámara, repetimos el proceso de la variable y adaptamos los Inputs:

```
void Update() {  
    Vector2 input = pi.actions["Look"].ReadValue<Vector2>();  
    Vector3 rotateMobile = new Vector3((input.y), (input.x * -1), 0);  
    transform.eulerAngles = transform.eulerAngles - rotateMobile;}  
}
```

Misma idea que con el movimiento, solo que en vez de acceder a Move, buscamos por el que tiene el Joystick derecho, “**Look**”

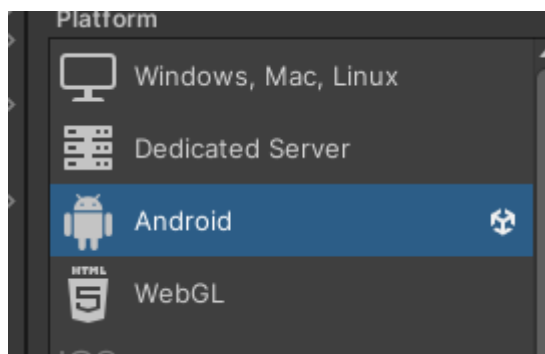
Build para Mobile

Para hacer un build destinado a mobile primero tenemos que estar seguros de tener el módulo de **Android instalado en Unity**.

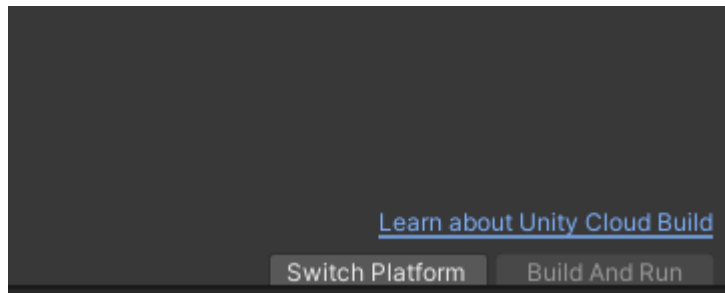
▼ PLATFORMS	DOWNLOAD SIZE	SIZE ON DISK
Android Build Support	Installed	1.86 GB
└─ Android SDK & NDK Tools	Installed	165.94 MB
└─ OpenJDK	Installed	145.91 MB

Luego iremos a iremos a **file->Build Setting**

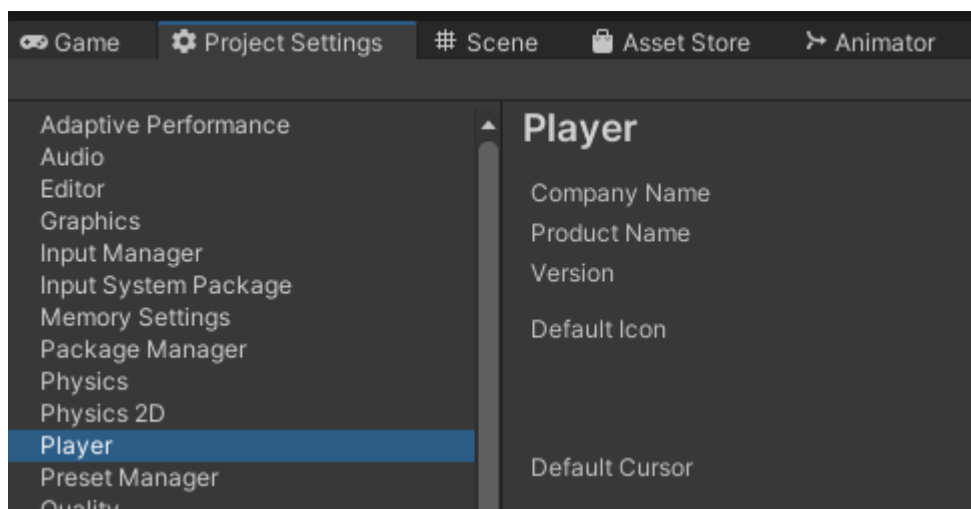
Seleccionaremos la Opción de **Android**



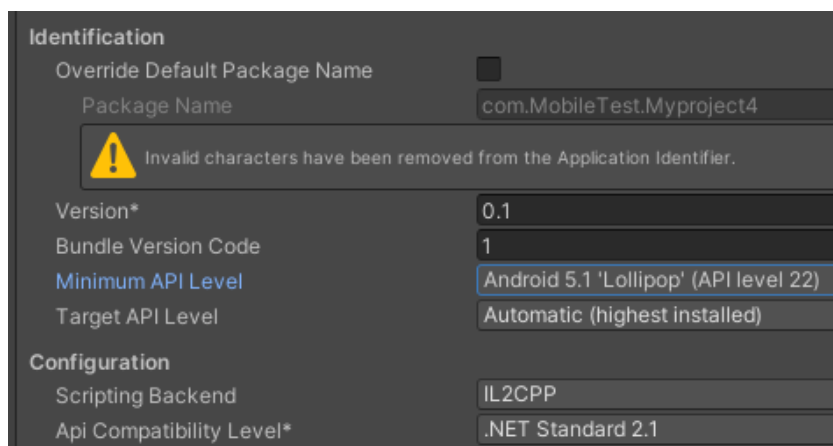
Y haremos click en **Switch Platform** abajo a la derecha, donde mayormente està la opción **“Build”**



Cuando terminemos iremos a **Edit -> Project Setting -> Player**



En **Identification** modificaremos el **Minimum API**, para el mas bajo posible y el **Target API** en **Automatic**. En **Configuration** cambiaremos el **Scripting Backend** de **Mono** a **IL2CPP**



Y con esto iremos a **Build** y ya tendremos nuestra APK.

Bienvenido al mundo Mobile:



A medida que **Nexus** avanza, el equipo de **TalentoLab** se enfrenta a un nuevo desafío: expandir el juego a un público más amplio. Con la creciente demanda de juegos móviles, ha llegado el momento de **adaptar el proyecto para plataformas móviles**. Esto implica rediseñar algunas mecánicas y ajustar los controles para ofrecer una experiencia fluida y atractiva en dispositivos táctiles.

"Nuestro juego está tomando forma. Ahora necesitamos que todo el mundo, sin importar la plataforma, pueda disfrutar de **Nexus**. ¡Es hora de llevarlo a las manos de los jugadores!"

Este es un paso clave en el desarrollo del proyecto, y el equipo debe aprender a manejar los **inputs para mobile**, usando el **Input System** de Unity para que el control sea intuitivo y cómodo.

Ejercicios prácticos:

Con el juego **Nexus** tomando forma en las plataformas móviles, es el momento de asegurarse de que los jugadores puedan interactuar de manera cómoda y fluida. Para ello, debemos adaptar las acciones del juego a botones táctiles, algo clave para brindar una experiencia completa en dispositivos móviles.



"Recuerda, en dispositivos móviles no contamos con el mismo tipo de controles que en una PC o consola. Necesitamos ofrecer controles intuitivos, accesibles, y con una respuesta inmediata para mantener la experiencia envolvente."

🎯 **Objetivo:** Implementar un botón en la interfaz móvil que permita ejecutar una acción específica dentro del juego.

✓ **Diseñar un botón** en la interfaz móvil que realice una acción que elijas dentro del juego (por ejemplo, un ataque, salto, o interacción).

✓ **Configurar el Input System** para que interactúe correctamente con las mecánicas del juego, activando la acción seleccionada al tocarlo.

✓ **Asegurarte de que sea visible y fácil de usar**, ubicándolo de forma que no interfiera con otros controles, y testeando su funcionamiento en un dispositivo móvil real.

✓ **Personalizarlo** para que tenga una estética adecuada al estilo de **Nexus**, manteniendo la coherencia visual con el resto del juego.

Materiales y recursos adicionales.

Input System

<https://docs.unity3d.com/Packages/com.unity.inputsystem@1.13/manual/index.html>

Desarrollando con Android

<https://docs.unity3d.com/es/530/Manual/android-GettingStarted.html>

Preguntas para reflexionar.

1. ¿Qué diferencias se plantean a la hora de hacer un juego para PC y para Mobile?
Algunos temas centrales: Controles, Flow, curvas de interacción y optimización.

Próximos pasos.

En la próxima clase reflexionaremos sobre el juego creado, compartiendo ideas, experiencias y opiniones para enriquecer el proceso de diseño y desarrollo. Será una oportunidad para identificar fortalezas, áreas de mejora y aprender de los enfoques de cada uno.



Buenos Aires
aprende
Agencia de Habilidades para el Futuro

BA Buenos
Aires
Ciudad