



## IFCD57. Programación para videojuegos en unity.

**Sku:** PS\_IFCD57

**Horas:** 300

**Formato:** HTML

### OBJETIVOS

- Realizar las operaciones principales para la creación de un proyecto de videojuego con Unity.
- Desarrollar videojuegos en 2D utilizando gameobjects, recursos y scripts específicos para escritorio y dispositivos Android.
- Desarrollar videojuegos en 3D utilizando gameobjects, recursos y scripts específicos para escritorio y dispositivos Android.

### CONTENIDOS

- **Unidad 1. Programación con Unity.**
  - Unity como motor de videojuegos.
    - Crear, abrir, guardar proyectos y escenas
    - Estructura de carpetas de un proyecto en Unity.
    - El editor de Unity
    - Unity en 2D vs 3D.
    - Carpetas especiales (Librerías ...)
    - Importación de objetos a un proyecto (Formatos admitidos)
    - GameObjects
    - Componentes
    - Prefabs.
  - Diseño de un videojuego
    - Flujo de un juego.
    - Creación de escenas (splash, menú principal, juego...)
    - Creación de elementos y ventanas emergentes
    - Adaptación del diseño a cualquier tamaño de pantalla
    - Exportación a diferentes plataformas
  - Scripting en Unity.

- Ciclo de vida y orden de ejecución del MonoBehaviour
  - Variables: Definición, asignación y tipos de variables
  - Clases y Atributos (visibilidad)
  - Static vars
  - Definición y llamada a métodos
  - Buenas prácticas en la definición de métodos, clases, atributos y variables.
  - Referencias de objetos
  - Acceso y modificación de componentes
- **Unidad 2. Videojuegos 2D en Unity**
  - La interfaz de usuario (UI)
    - Interfaz adaptativa (“responsive”)
    - Tipos de elementos de la UI (Text, Image, Slider, Button)
    - Interacción con los elementos de la UI
    - Incorporación de sonidos (sonido ambiental y efectos).
    - Cambio de escenas del juego con elementos de la UI
    - Menús (principal, pausa)
  - Cámaras
    - Tipos y configuración de cámaras.
    - Definición de diversas cámaras en el juego
    - Modificación de los atributos de una cámara
    - Cambios entre cámaras
  - Sprites.
    - El editor de sprites
    - Ordenación de capas
    - Atributos de un sprite
    - Animación de sprites
  - Física 2D.
    - Leyes físicas en un videojuego.
    - El componente Rigidbody 2D.
    - El componente Collider 2D (Tipo Trigger)
    - Tabla de colisión en Unity.
    - Detección de colisiones mediante programación
    - Materiales físicos
    - Rigidbody2D (Fuerzas)
    - Configuración y programación de la física de un videojuego
  - Subprocesos.
    - Invoke.
    - InvokeRepeating
    - Coroutines
  - Sistemas de partículas.
    - Configuración de un sistema de partículas básicas
    - Detección de colisiones con partículas
    - Sistemas de partículas en tiempo de juego, en tiempo de ejecución.
  - Animación.
    - Importación de un modelo con animaciones en FBX
  - Tipos de animaciones

- Cortado de animaciones
  - Creación o modificación de animaciones mediante programación.
  - Creación de un Animator Controller
  - Diagrama de estados de animaciones
  - Activación y desactivación de animaciones mediante programación
- Layers y tags.
  - Tags.
  - Asignación de tags
  - Layers
  - Asignación de layers
  - Configuración del renderizado de layers mediante la cámara
- Un juego de plataformas.
  - Menú principal.
  - Cambio de escenas
  - Diseño del nivel del juego
  - Pausa del juego
  - La interfaz de usuario
  - Otras escenas (victoria, derrota ...)
- Build en Android.
  - Instalación de Android Studio
  - Configuración de Android Studio
  - Configuración del móvil en modo desarrollador
  - Instalación de Unity Remote
  - Exportación a Android (generar un Apk)
  - Depuración en Android - Interacción con un dispositivo móvil.
  - Interacción táctil básica. (Touches)
  - Programación de la interfaz táctil básica
  - Gestores táctiles, definición y programación (Swipe, ...)
- Adaptación del juego de plataformas en un dispositivo móvil
  - Modificación de la interacción
  - Modificación de los controles
- **Unidad 3. Videojuegos 3D en Unity.**
  - Terrenos.
    - El editor de terrenos
    - Subida y bajada del terreno
    - Suavizado de alturas
    - Texturizado del terreno
    - Añadido de árboles
    - Configuración del terreno.
    - Importación de un terreno en FBX
    - Zonas de viento
  - Cámaras 3D.
    - Cámara en Perspectiva.
    - Propiedades generales
    - Pantallas divididas para 2 y 4 jugadores
    - Minimapa del juego (radar)

- First Person Controller.
    - Paquete Standard Assets
    - Definición de las mecánicas del FPS
    - Configuración y explicación de cada uno de los atributos
    - Detección de colisiones
    - Definición del HUD
    - Definición de enemigos
    - Definición de los diferentes estados de los enemigos
  - Zonas de Spawn
    - Creación de objetos en tiempo de juego
    - Creación de objetos de forma aleatoria
    - Definición de la política de creación de objetos
    - Creación de enemigos en tiempo de juego
    - Definición de la política de creación de enemigos
    - Características generales de la instanciación
    - Raycast
  - Animator y diagramas de estados de la animación
    - Definición de las animaciones existentes en el juego (enemigos, cofres, puertas ...)
    - Creación de animaciones con Animation
    - Asignación y definición de estados (Animator controller)
    - Parámetros del Animator Controller por el cambio de estado.
    - Programación del cambio de estados en el Animator Controller.
  - Inteligencia artificial de enemigos (NavMesh)
    - Inteligencia artificial de enemigos básica
    - Inteligencia artificial de enemigos compleja mediante NavMesh
    - El NavMesh Agent.
    - Propiedades del Bake
    - Programación de la inteligencia artificial de enemigos.
  - Sonido
    - Audio Clip
    - Audio Source
    - Efectos y filtros
    - Banda sonora
    - Sonido permanente entre cambios de escena (NonDestroy)
    - Activación de un sonido mediante programación
  - Iluminación.
    - Tipos de luz.
    - Iluminación Global.
    - Animación de luces
    - Programación de las luces
  - Vehículos.
    - Físicas del vehículo.
    - Wheel Collider.
    - Configuración de los parámetros de un WheelCollider.
    - Programación de un vehículo funcional en el videojuego.
-