

CURSO DE

PROGRAMACIÓN DE VIDEOJUEGOS CON UNITY Y UNREAL

180 H. (2-3 MESES)

El contenido de este programa es orientativo, pudiendo variar el mismo por razones de actualización o modificación de los contenidos.

CEI.

OBJE- TIVOS

La programación es, junto con el diseño, uno de los elementos esenciales en el desarrollo de videojuegos, y con el *Curso de Programación de Videojuegos con Unity y Unreal* puedes adquirir los conocimientos necesarios para adentrarte en esta profesión.

En este curso introducimos a los alumnos en las aplicaciones y los lenguajes necesarios para desarrollar videojuegos. Profundizaremos en lenguajes como C# en Unity y el manejo de Blueprints en Unreal para crear videojuegos y proyectos interactivos.

INFOR- MACIÓN

Unity
Unreal

DURACIÓN

180 H. Total

Presencial u *Online*

140 H. de Clases

Lunes a jueves
(+algún viernes)

40 H. Proyecto final

HORARIOS

MAÑANAS

09.00 - 12.00 H.
12.00 - 15.00 H.

TARDES

16.00 - 19.00 H.
19.00 - 22.00 H.

PRECIO

2.050 €

Consulta ofertas y
promociones ac-
tuales en

info@cei.es
www.cei.es

OFI- CIAL



AUTHORIZED
Training Center

Nuestra escuela es uno de los cinco centros **Adobe Authorized Training Center** de España. Este hecho demuestra que **CEI** ha cumplido con todas las exigencias que solicita **Adobe** y, asimismo, constituye una garantía de calidad para todos nuestros alumnos.



Authorized
Training Center

CEI es un **Centro de Formación Autorizado por Apple**. Por ello, nuestros alumnos podrán obtener la certificación profesional de **Apple** e impulsar su carrera en tecnologías del diseño y de la información, entornos empresariales y mercados B2B.



IBM confía en nosotros y por ello nos ha otorgado la insignia **IBM Education Partner**. Este sello premia a centros de enseñanza que mantienen y fomentan una formación de calidad.



Nuestra escuela se consolida como Centro de Formación de referencia de **Maxon** en España y nos convertimos en la 1ª escuela de Europa en ser **Maxon Training Provider for R18**.



CURSO

**PROGRAMACIÓN
DE VIDEOJUEGOS
CON UNITY Y UNREAL**

UNITY

01- HOLA UNITY

- Instalación de Unity
- Unity HUD
- Versiones y actualizaciones
- Proyectos 2D y 3D
- Guardar proyecto y escena
- Importación de *Assets*
- Ventanas: Proyectos, Escena, Juego, Jerarquía e Inspector

02- FUNDAMENTOS DE C#

- Fundamentos de tecnología y computación
- Variables y constantes
- Tipos de datos primitivos
- *Overflowing*
- *Scope*
- Declaración e impresión de variables
- Conversión de datos, operadores y comentarios
- Trabajar con operadores aritméticos
- Operadores lógicos y de comparación
- Tipos de datos no primitivos.
- Lógica y Control de flujo
- Trabajar con *loops*
- Crear y llamar funciones
- *Arrays* y listas

03- C# CON UNITY

- *Scripts* como componente
- *Translate*
- *Rotate*
- *Inputs*
- *Prefabs*
- *GameObjects*
- *Instantiate*
- *Random*
- *Destroy*
- Vectores y tiempo
- *Box Collider*
- *RigidBody*
- *OnTriggerEnter*
- *OnTriggerExit*
- Cámara
- Físicas en 2D
- Máscaras de colisión
- *Joins*
- *Effectors*
- Materiales físicos
- Físicas en 2D

04- PROGRAMACIÓN ORIENTADA A OBJETOS (POO)

- Clase *GameObject*
- Acceder a los componentes
- Entender las transformaciones
- *Vector 3*
- Transformando objetos con código
- *Time*
- *DeltaTime*
- *Transform*
- *Translate*
- Mover objetos de un punto A a un punto B
- *MoveTowards* y *Lerp*
- Mover objetos de ida y retorno
- La función *Update*
- *Transform.forward*
- *Transform.right*
- *Transform.up*
- Rotar objetos con *transform.rotation* y rotar objetos con *transform*
- *Rotate*
- Escalar objetos
- *GetComponent*
- *SerializeField*
- Movimiento con física

05- CREACIÓN DE UN *PLAYER* EN C#

- *Character controller*
- Movimiento del *Character Controller*
- Movimientos en local y global
- *Input Manager*
- *Input.GetAxis*
- *Input.GetKey*
- *Input.GetKey(KeyCode)*
- Mover y rotar nuestro *Character Controller*
- Saltar obstáculos
- Motor de físicas de Unity
- *PhysicsManager*
- Movimiento con *Rigidbody*
- Seguimiento de cámara
- Destruir objetos con colisiones
- Tele-transformación con *Triggers*

06- RAYCAST

- Abrir puertas con *triggers*
- *Raycast*
- Cómo visualizar el *Raycast*
- Entendiendo *Raycast* de físicas
- La clase *Ray*
- Cómo obtener información con *RaycastHit*

- Raycast y la posición del ratón
ScreenPointToRay
- Comunicación con SendMessage

07- DECALS

- Preparar nuestro *Prefab Decal*
- Instanciar los *Decals* con Raycast
- Definir el tiempo de disparo
- Rotación de nuestros *Decals*
- Selección de objetos para disparar
- Crear un *Array* de *Decals*
- Emparentar los *Decals*
- Destruir los objetos

08- PROGRAMANDO NUESTRA UI (INTERFAZ DE USUARIO)

- Entendiendo el *Canvas* (lienzo)
- Componentes del *Canvas*
- *Canvas Scaler*
- Entendiendo el *Rect Transform*
- *Image*
- *Image Raw*
- *Text*
- *Button*
- *Slider*
- Creando una mirilla
- Creación de un contador de *coins*
- Creación de una barra de vida
- Hacer daño a nuestro *FPSController*
- Cómo curar nuestro *FPSController*
- Cómo limitar el número de munición
- Contador de balas
- Recargar munición
- Pantalla de fallecimiento

09- ANIMACIÓN EN UNITY

- Características principales
- Control de pesos de animación en tiempo de ejecución
- Llamadas a eventos desde animación
- Jerarquías
- *Animation Clips*
- Ventana Animación
- *Animation Controller*
- Ventana *Animator*
- Máquina de estados
- Transiciones
- Configuración de la importación del Modelo
- *Option Rig*
- *Option Animations*
- *Character Controller* en nuestro personaje

- *Animator Controller*
- Interacción de un estado a otro
- Estado *Walk* a *Running* y de *Running* a *Walk*
- Transiciones

10- NAVIGATION Y PATHFINDING

- Cómo funciona el sistema de navegación
- Encontrar Caminos (*Finding Paths*)
- Seguimiento de recorrido
- Evitando Obstáculos
- El movimiento de un Agente Global y Local
- Construir un *NavMesh*
- Crear un *NavMesh Agent*
- Crear un *NavMesh Obstacle*
- Crear un *Off-mesh Link*

11- AGREGAR AUDIO EN UNITY

- *Audio Source* y *Audio Listener*
- Música del juego
- Efectos de sonido
- Múltiples canales de audio
- Co-rutinas
- Creación de ejecutables

UNREAL

12- HOLA UNREAL

- ¿Qué es *Unreal Engine*?
- Descarga del *Epic Games Launcher* y familiarización
- *Overview* del *Unreal Editor*
- Creando un proyecto
- Blueprints vs. C++
- Descargas de *Assets*
- Explorando Unreal
- Mover
- Escalar
- Rotar
- Coordenadas locales y globales

13- DENTRO DE UNREAL

- Viewport en profundidad
- Navegando en Unreal
- Modificación de objetos
- Los visores
- *Play*
- *ToolBox*

- *World Outliner*
- *Details Panel*
- *Modes Panel*
- *Content Browser*
- Tipos de archivos y *Assets* válidos para Unreal 4

14- PROGRAMACIÓN EN BLUEPRINTS

- *Level BluePrint* y *Blueprint Class*
- Eventos
- Evento *BeginPlay*
- Evento *Tick*
- Variables
- *Boolean*
- *Byte*
- *Integer*
- *Float*
- *String*
- *Name*
- *Text*
- *Append*
- *Structure*
- *Transform*
- *Interface*
- *Objects Types*
- *Enums*
- Funciones y Macros
- Entradas y Salidas
- Clases de Blueprints
- Colisiones
- Colisiones-Sistema con Láseres

15- PLAYER

- *Characters*
- Movimientos
- Rotación y saltos
- Animación con Blueprint
- *Animation Blend Space*
- *Animation Montage*
- Eventos
- Variables
- Funciones y Macros
- Clases de Blueprints

16- MECÁNICAS DE JUEGO

- Colisiones
- Sistema con láseres
- *Event Dispatchers*
- Creando trampas
- Creando *Power-ups*
- Creando interruptores
- Inteligencia artificial

- Patrullar
- Seguimiento
- Sistema de dificultad

17- INTERFACES GRÁFICAS

- *Widgets*
- Eventos en UI
- *Widget Interaction Component*
- Textos en el juego
- Materiales y texturas
- Menú principal
- Menú de Pausa

18- INTELIGENCIA ARTIFICIAL

- *Behavior trees* y *Blackboards*
- *Composite nodes*
- *Decorators* y *Services*
- *AI controller*
- *AI Perception*
- *Polish Attack*
- *Enemy Chase*
- *Enemy Attack*

19- COMPILACIÓN