

Desarrollo de videojuegos con Unity

Más que un motor.

Unity es un ecosistema completo de herramientas y servicios diseñados para personas que desean desarrollar un negocio exitoso a través de la creación de juegos multiplataforma y contenido interactivo.

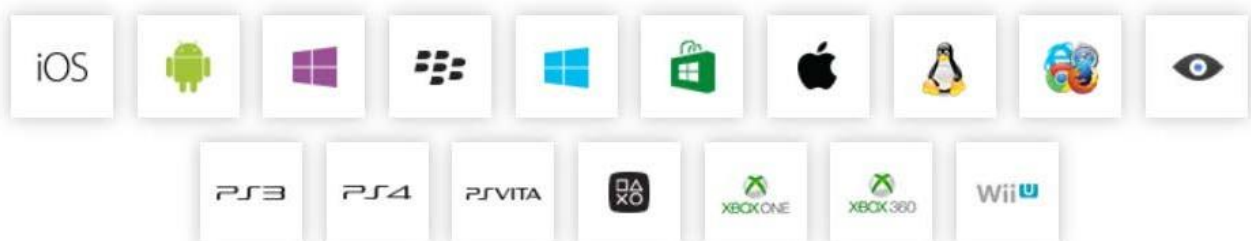
El motor **Unity** integra, en una plataforma sin paralelos, las herramientas que se necesitan para crear contenido interactivo 2D y 3D; soluciones de colaboración, despliegue rápido multiplataforma, así como servicios de retención, publicidad y análisis que harán crecer tu negocio.

Crea lo que imagines, trabaja como te guste.

Desarrolla tu contenido con herramientas y flujos de trabajo altamente optimizados en el motor **Unity**. Experimenta inclusive mayor eficiencia personalizando y ampliando la funcionalidad de la interfaz del editor de la forma en que desees.

Despliegue en más de 17 plataformas

Publica en todas las plataformas móviles, Mac, PC y Linux para escritorio, web o consolas. Utiliza las poderosas herramientas multiplataforma para lograr que tu contenido interactivo se ejecute en forma fluida en cualquier dispositivo.



Competencias

Al final de la formación el alumno alumnos deberá ser capaz de:

- Conocer y utilizar el motor de desarrollo de videojuegos Unity y elaborar la estructura y lógica de un videojuego.
- Conocer los principios esenciales de diseño de videojuegos profesionales. Conocer la programación en JavaScript para poder controlar toda la mecánica de cualquier situación que pueda requerir nuestro videojuego.
- Gestionar proyectos de desarrollo de videojuegos en Unity.

A quién va dirigido

El curso va dirigido a todas aquellas que quieran desarrollar videojuegos multiplataforma de gran calidad como artistas, diseñadores, programadores, animadores y en general cualquier persona con inquietudes en la creación de videojuegos.

No son necesarios conocimientos previos de programación

Plan de Estudios Detallado.

Instalar Unity y crear un proyecto

Interface

Entorno

ventana Scene

ventana Juego

ventana jerarquía

ventana Project

ventana inspector

Asset Store

Gestión

importar Assets

Prefabs

layers

tags

Packages

Asset Store

Importar Exportar Packages

Publicar

Plataformas

Project Settings / Build Settings

Project Settings / Player Settings

Motor Gráfico

Diseñar meshes xa videojuegos

Importar desde 3DMAX (unidades/normales/smoothingGroups/sistema de coordenadas)

MeshFilter+MeshRenderer = visibleObject

settings de Texturas

settings de meshes

Materiales y shaders

Crear shaders propios

Camaras

Luces

SkyBox

Project Settings / Render Settings

Project Settings/Quality settings

Manipular gráficos con Scripting

luces/ materiales / etc...

Motor Físico

- Unidades y convenciones
- Colliders / Triggers
- RigidBody
- Add Force Component
- Materiales Físicos
- Joints
- Project Settings/ Physics
- Ragdoll
- Manipular Físicas por Script

Motor Grafico 2, animación

- ventana animation = crear animaciones desde unity
- importar meshes animadas (legacy)
- importar meshes animadas (humanoid Avatar)
- importar meshes animadas (morphing)
- Animator Controller / ventana Animator (maquina de estados)
- Gestión via Script

Motor de Audio,

- Import Audio Settings
- Audio Source
- Audio Listener

Motor Gráfico 3, UI (User Interface)

- Canvas UI
- Objetos UI (imágenes, botones, scrollBars...)
- Eventos UI
- Transiciones UI

Unity como motor 2D

- Motor Gráfico2D
- Motor Físico 2D

Scripting

- Scripts en unity (lenguajes/integración con unity)
- Programación Estructurada
- MonoBehaviours
 - Awake
 - Start
 - Update
 - LateUpdate
 - FixedUpdate
 - OnTriggerEnter
 - OnCollision
 - OnMouseDown

Debug Class

- Log
- Breck
- DrawRay

Script Execution Order Settings

Input class

- Key / Button / Axis
- Project Settings / Input

Physics Class

- Raycasting
- SphereCast
- OverlapSphere

Mathf Class

- Clamp
- Sin/Cos
- Lerp
- long etc...

Screen Class

Time Class

- Time
- deltaTime
- timeScale

Crear Clases

Transformaciones a través de script

- Transform.Translate()
- El problema de rotar un objeto: Quaternions
- Transform.Rotate()
- LookAt

Enabling / Disabling components

Activating / DeActivating GO

Instantiate / Destroy