



LEVELUP

GAME DEV HUB

ENTORNOS 3D

GAME ENGINES

Game Engines

- Motores de juego **estándar** en la industria
 - Unreal Engine - Epic Games
 - Unity - Unity Technologies
 - CryEngine (en menor medida) - Crytek
 - Game Maker (2D)
- Motores de grandes compañías
 - Lumberyard, UbiArt Framework, ...
- Otros motores
 - Ogre 3D, Godot3D, Panda3D, ...

Diferencias entre Unreal Engine y Unity

¿Cuál es mejor?



UNREAL
ENGINE

VS



unity

Diferencias entre Unreal Engine y Unity

DEPENDE

Diferencias entre Unreal Engine y Unity

- Plataformas destino
 - Unreal especializado en **PC y Consolas**, siempre en entornos 3D
 - Aprovecha mejor las capacidades de las máquinas, tiene herramientas de análisis de rendimiento muy detalladas, tiene nativamente soporte multijugador.
 - Requiere más recursos para trabajar y funcionar pero el resultado visual es mucho más potente con menos esfuerzo (según el acabado que se quiera)
 - Unity orientado a **Móvil, PC y consolas 2D y 3D**
 - Unity dispone de herramientas más pensadas para integración y desarrollo de tipos de juego móvil, pensando mucho en integración de publicidad, micropagos, recolección de métricas, ...
 - Tiene desde hace unos años herramientas especializadas para trabajar en 2D (dentro de un entorno 3D)

Diferencias entre Unreal Engine y Unity

- Curva de aprendizaje
 - Unity
 - Es el sistema estándar elegido por los estudios indies ya que es muy accesible, muy visual y con las opciones necesarias para crear proyectos. Tiene una escalabilidad limitada ya que no está pensado para ello, pero para hacer proyectos más pequeños es ideal.
 - Unreal, dispone de muchas herramientas con una complejidad mayor. La curva de aprendizaje es mayor que en Unity, está pensada para grandes proyectos muy escalables, pero para aprovechar toda esta potencia es necesario un conocimiento alto del motor.

Diferencias entre Unreal Engine y Unity

- Implementación
 - Unity
 - C#, simplificación de C++ con añadidos para Unity
 - Unreal Engine
 - C++, más complejo, se puede programar a más bajo nivel
 - Blueprints, programar sin tocar código

Diferencias entre Unreal Engine y Unity

- Flexibilidad
 - Unity
 - Muy abierto para crear entornos con la estética que se quiera, iluminación y shading
 - Unreal Engine
 - Más encorsetado, si quieres hacer algo alejado de las herramientas que te ofrecen tienes que tocar el motor y darle soporte a lo largo de las versiones. Mantenimiento significativamente más complejo.

Cómo organizar nuestros proyectos

- Requisitos para el desarrollo
 - Unity
 - Bajo coste de producción, necesita menos máquina para desarrollar los proyectos, a cambio los tiempos de carga pueden llegar a ser muy bloqueantes
 - Sistema por GUID, habilidad de eliminar y cambiar archivos de carpeta libremente
 - Unreal Engine
 - Requiere máquinas más potentes y significativamente más memoria para funcionar, tiempos de compilación muy inferiores, sobre todo para probar el juego.

Cómo organizar nuestros proyectos

- Sistema de archivos
 - Unity
 - Uso directo de los archivos de trabajo en algunos casos (.blend) y los archivos exportados (fbx, obj, tga, png,...)
 - Unreal Engine
 - Uso de archivos propios referenciados
 - Sistema muy “hardcoded” en los que mover archivos, cambiar nombres o borrar archivos puede ser un infierno.

Diferencias entre Unreal Engine y Unity

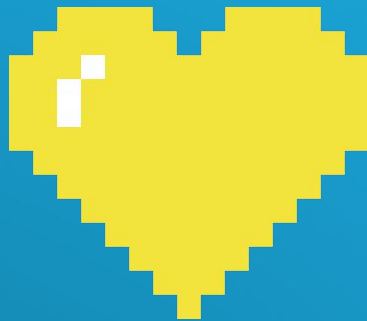
- Control de versiones
 - Unity
 - Tiene buena implementación con Plastic SCM, sistema accesible y mantenimiento relativamente sencillo. Se pueden usar la mayoría de sistemas de Control de versiones. En ocasiones tiene problemas de hacer check out en archivos que no se han tocado.
 - Unreal Engine
 - Buena implementación con Perforce, sistema robusto pero de coste muy elevado y requiere mantenimiento especializado, las grandes empresas usan casi todas estas. Se pueden usar otros sistemas pero la integración no es tan buena.

Diferencias entre Unreal Engine y Unity

- Tecnologías nuevas
 - Unreal Engine
 - **Nanite** es un sistema de optimización virtualizado de meshes con millones de polígonos para conseguir un realismo extremo. Tecnología recién creada en Unreal Engine así que tiene algunas limitaciones que van puliendo.
 - **One file per actor** Sistema que permite trabajar en una misma escena mucha gente a la vez
 - **Lumen** Nuevo sistema de iluminación que busca un resultado realista en tiempo real, de modo que no se necesitan lightmaps. Queda mucho desarrollo por delante, sobre todo en temas de optimización.



LEVELUP
GAME DEV HUB



¡Muchas gracias!