Introducción Unreal

Motor de epic Games y viene de un juego que se llamaba Unreal y usaron este motor. Los señores que lo crearon descubrieron que a la hora de hacer un juego había 2 partes

* Tecnica: cabelado interno
* Creativa: el juego concreto que quería yo hacer

Vieron que la parte técnica era muy similar siempre (colisiones, animaciones, …). Era como si Van Gogh tuviera que hacer sus pinturas, no, las tiene ya hechas. Entonces la idea es esa: centrarnos en lo creativo.

Pongamos que vamos a hacer un proyecto en 6 meses, para 6 hombres. 36 meses hombre.

Si empiezo en el 0, llego al nivel 36, pero si llega un señor que ya nos pone en el nivel 50 con su código pues llego mas lejos en menos tiempo. Es como un taburete.

**No resuelvas problemas que no tienes**.

Hay varias capas al hacer juegos:

* wizard (viene hecho ya, lo puede hacer cualquiera)
* **Blueprints** (**diseñador**): es una metáfora visual del código C++ que hay por detrás. Esto está **interpretado** (es más **lento**).
* **C++** (**coder**): para customizar el engine. Está **compilado** (es más **rápido**).

Podemos publicar el juego perfectamente si no ganamos dinero y si ganamos dinero hay que dar a Epic un 5% o así.

Vamos a ver también donde guarda Unreal los proyectos.

Unreal 4 acabo en la 4.27. Cuando hay una nueva versión hay una migración. Con blueprints migración y ya, con C++ es más jodido.

Lo abrimos con EpicGamesLauncher. Componentes:

* Instalador
* Engine
* Marketplace: es como una asset store de Unity

Quality Preset: lo dejamos en máximo y se puede cambiar después

Starter content: le damos siempre marcado, viene con unas cuantas mallas, sillas y tal. Eso no es suficiente, son 4 cositas.

Raytracing: sale a partir de la 4.27 🡪 en la 4.20 ya empezaban a salir cosas chulas que no soportaba Unreal (disersion rayleigh, hace que la luz atmosférica sea más bonica).

Una imagen editada de un árbol

Descripción generada automáticamente con confianza baja

Es como aire volumétrico.

El raytracing calcula las reflexiones mucho mejor. Le mete mucha caña a los PCs así que mejor que no. Si tenemos a partir de 3000 la activamos, si no, no.

* **Lumen:** usa radiosity, es un algoritmo de iluminación muy potente: puede poner luz dinámicamente en tiempo real. Antes o la luz no era dinámica, o si era dinámica era una basura.
* **Nanite**: algortitmo de teselación al alta en tiempo real (al alta porq añade detalle): su camisa, si fuera una malla no tendría arrugas, estaría en un Normal Map. Si el motor es tan potente podría hacerlo en tiempo real. Fabrica geometría. Vemos mas detalle del que realmente hay.

Un hombre parado en una montaña

Descripción generada automáticamente con confianza media

Normalmente funciona con **1 solo Viewport** (aunque tamb se pueden poner 4).

* **Cubo+** para añadir objetos
* **Outliner**: jerarquía y abajo los detalles
* **Content** **Drawer**: navegador de carpetas

Se puede jugar al juego desde distintos formas. Hacer una build es MUY LENTO. 20 minutos el proyecto vacío. No hacer build si no es imprescindible.

* Selected Viewport: este es el que usamos
* Mobile Editor
* Standaolone Game
* Simulate

Para hacer un binario (ya lo explicará, que lo han cambiado en el 5)

Cuando se hace el nivel se hacen WildBoxes

* **Level Designer**: una persona hace el Blocking del nivel. Representación fea pero precisa del nivel o juego. La gameplay está ya definida, tiene que ocupar lo que diga el level designer.
* **Artist**: La segunda persona, sustituye los objetos del Blocking por arte, arboles, puertas, etc.

Imagen que contiene edificio, tablero, hombre, banqueta

Descripción generada automáticamenteVaciamos el escenario

En Unreal 4 se quedan las sombras, en el 5 se hacen dinámicamente. En Unreal 4 hay que reconstruir la luz cada vez que mueva la geometría. Build Lighting no se qué.

Pero ojo en el 5 si es dinámico tengo menos FPS, a lo mejor renta en algunos casos cambiarlo a objetos estáticos, para que vaya mas rápido.

Vamos a hacer un partenon.

En los detalles de los objetos dice:



* Static: no se puede mover (rendering mas rápido y hace Baked Lighting)
* Stationary: este objeto se puede cambiar pero no movido (se puede borrar, sombra sí o no), sombras dinámicas cacheadas.
* Movable: totalmente dinámico, render mas lento y casts a Dynamic shadow,

Vistas

Interfaz de usuario gráfica

Descripción generada automáticamente

Lit, unlit, wireframe las típicas

Pero tamb hay para programadores-

* Shader complexity, para que nos enseñe cuanto cuesta pintar cada parte de la pantalla para la tarjeta gráfica.
* Y un montón de otras formas de Debug.

Ejemplo: al tirar un tiro se reduce los FPS, entramos aquí y vemos que es por el sistema de partículas.

Window > Place Actors

Para hacer la Build: Platforms > Windows > Package Project

Un compilado tiene un binario (.exe) (lógica) y unos datos (contenido). El package Project lo empaqueta todo. El cook content solo compila el binario, lo que hace que vaya más rápida.

y ahí tenemos plataformas por defecto, no se necesita ningún SDK para estas. Para las otras sí. Si quieres Mac, PlayStation, Xbox. Tu hablas con ellos y te dan el sdk para unreal y te hace un patch.

A veces quiero un techo, que es una forma mas compleja. Para eso podemos usar Operaciones Booleanas. Para mallas con 3DSMax prohibido.

Nos vamos al menú de **Geometry**, ya que aquí tenemos un cubo paramétrico, que se puede manipular. Podemos cambiar por ejemplo los sides. (menú Shapes no podemos)

Imagen que contiene interior, con baldosas, azulejo, verde

Descripción generada automáticamente

Operaciones booleanas

* **Union**: suma objetos
* **Resta**: eso

Imagen que contiene con baldosas, azulejo, tabla, grande

Descripción generada automáticamente

Cojo la caja, y en los detalles a **Brush Type**. Substrractive

LUCES

En el outliner hay una carpeta lighting.

* DirectionalLight: es el sol

Vamos a poner la lampara. En el panel de la izq hay luces:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* **Directional light**: llegan rayos paralelos.
* **Point light**: bombilla, omni direccional
* **Spot light**: es cara, es un cono de luz (flexo)
* **Rect light**: es un plafón de luz, un rectángulo largo que emite luz
* **Sky light**: simulación de la atmosfera

Unidad del SI de la luz 🡪 candela

Efectos: niebla, brillos…

Todos los niveles de unreal hay niebla, Eso es porq si vemos una montaña de lejos la ves un poco azulada, siempre hay.

Hay otro objeto PlayerStart, es donde empieza el jugador.

**Texturas**:

Cojo el objeto y sale un Static Mesh y un Material

**Import desde 3DSMax**

File > Export > Export a FBX

Verde: material

Cyan: static mesh

Interfaz de usuario gráfica

Descripción generada automáticamente

**BLUEPRINTS**

Ahora vamos a hacer que cuando coja el mystery Cube, lo coja y explote. Y esto lo hacemos con BluePrints

Crea un objeto blueprints

Interfaz de usuario gráfica

Descripción generada automáticamente

Para llevarme el proyecto a otro PC:

Save All y la carpeta donde esta el uproject le hago un zip y y ya

Bazar de Epic para Añadir al proyecto, y seleccionamos cual queremos. Al añadirlo ocupará unos GB más.

**Deberes**: masticar lo que hemos hecho, pero NO blueprints. Hacer una build.