Juan Abadía

[juan.abadia@gmail.com](mailto:juan.abadia@gmail.com)

El objetivo es crear un juego. Generar un binario. El ejecutable está acompañado de una serie de datos (donde está el player, la malla, etc.).

AVISO: el rendimiento no puede ser un problema. Debemos conocer todas las partes.

Un dibujo de un pizarrón blanco

Descripción generada automáticamente con confianza baja

**Cooking**: coge los datos raw y los adapta para que vayan lo más rápido en la plataforma destino (IOS / PS4 / PC / SWITCH / XBOX).

El objetivo es que el arte que llega tenga calidad.

**Asset**: cualquier recurso que necesita el juego.

Cosas necesarias para el videojuego:

* Art/**Personajes**
  + Meshes
  + Skeleton Mesh
  + Animaciones
  + Materiales
* **IA** **Controller**
* **Player** **Controller** ()
* **Iluminación**
* **Sounds** (SFX, MP3)
* **Partículas**: cuidado con el rendimiento.
* **UI**:
* **Props**
* **Weapons**: mesh + material + lógica
* **Meshes**
  + **Vértices**
  + **Faces**
  + **Normales** (por cada vértice)
  + **Material ID** (por cara)
  + **Puntos de interés** (anclajes)
  + **UV: coordenadas de textura** (puede haber varios)
  + **Malla de colisión** (triángulos y lista de geometrías{2 cubos por ejemplo})
  + **LOT: leve lof detail**
  + **Tangent Space:**

**Es muy importante que el ciclo de iteración sea corto**. Si arranco y tengo que esperar mucho para probar algo, pierdo mucho tiempo.

Un pizarrón con un texto en blanco

Descripción generada automáticamente

Imagen en blanco y negro

Descripción generada automáticamente con confianza baja

Aquí el unreal no nos dice que hay 8 vertices. Hay como 24.

Max dice que hay 6 caras, pero unreal te dice que hay 12 triangulos.

Cuando pinto un triangulo para una cara, tendrá distinta normal (aunque tenga distinta normal).

Imagen que contiene nieve, firmar, alambre, colgando

Descripción generada automáticamente

Sin embargo, será un mismo vertice para las caras de la izquierda F1 y F2. Un vertice viene definido por:

* Posición
* UV
* Normal

Vertice = 3 floats

1 float = 4 bytes. Con un float no puedo expresar tanto como con un entero

**Textura**

Grid (rejillas) 🡪 sus unidades de la rejilla son **texels**. En una textura de 8x8 tengo 64 texels

4 canales (gris, RGB, RG, RGBA). Asumimos que hay 4

4 bytes por texel. Para cada canal de un texel 1 byte (1 para R, 1 para G, 1 para B y otro para A)

Cuadradas (width = height), potencias de 2