En el motor tenemos una función para pasar de JSON a STRUCT.

Texto

Descripción generada automáticamente

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

Volvemos a Shaders

Difusa = N\*L

Especular = E,L,N (el ojo es el que determina si hay brillo)

Final = Difuse \* LightColor + Especular. Aquí no está el color del material. Rojo, absorbe todo menos el rojo. Si ilumino con azul oscuro, debe haber un coeficiente de material difuso que multiplique por la luz.

El especular quedaba mejor blanco. Coeficiente Especular (de material) para saber cuanto absorbe.

La ecuación es lineal (fisica). Un jpg de una textura, el color es el q se ve el monitor, pero queremos los cálculos en calculo lineal.

Dibujo de un pizarrón blanco

Descripción generada automáticamente con confianza media

* Valores a espacio lineal y El resultado se tienen que pasar a gama correcta
* Si tienes mucho especular, tienes poco color difuso, no puedo tener las dos cosas.

Se ideó el PBR, ecuaciones para que la luz sea realista (el especular cuando más tangencial mas se ve, un coeficiente que dice lo metálico q es el objeto, …).

Se montaron CubeMaps. Sirve para saber qué luz llega en cada dirección y se accede con 3 coordenadas (u,v,w)

Un dibujo de un pizarrón blanco

Descripción generada automáticamente con confianza baja

Hoy vamos a hacer unas pruebas de texturas y luego otras pruebas de entorno

En el píxel hacemos lo que queremos con lo que nos llega del Vertex. Hay unos valores que son las UVs (coord. De textura) normalmente vienen del objeto. Normalmente son 2 coordenadas. Necesitamos las UVs, un objeto y el sampler (es una preconfiguración: dice como accedo a la textura, que pasa si las uvs están fuera de 0 y 1, si quiero usar mipmaps, si quiero que interpole…).

Tenemos los mipmaps

Un dibujo de un pizarrón blanco

Descripción generada automáticamente con confianza media

El resultado es un float4 (color) de la textura.

Creamos una variable Uv en el shader

Texto

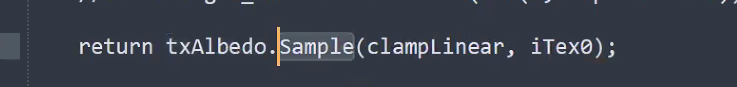
Descripción generada automáticamente

Y devolvemos return float4(input.Uv, 0, 1);

El sampler

Texto

Descripción generada automáticamente



Texto

Descripción generada automáticamente

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

Conv lineal:

Hacemos

* ( u.x + offset ) \* frequency

O bien

* Uv \* frequency
* Uv.x + offset

Ahora vamos a poner una segunda textura y combinar las dos.

Texto

Descripción generada automáticamente

Para el alfa:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteMapeo:

Texto

Descripción generada automáticamente

El ultimo

Texto

Descripción generada automáticamente

Para leer el texel de al lado, tenemos que hacer 1/resolución textura (hay una constante que se pasa).

Deberes: podríamos hacer que un vertice estuviera vibrando