# Hay un Sistema de reflection de los shaders:

yo le puedo preguntar al shader dónde está la cámara. Yo le digo cámara. Y el me dice 1. (slot 1).

Un dibujo de un pizarrón blanco

Descripción generada automáticamente con confianza baja

**El shader traduce Nombre de estructura (cámara) a un número.**

En el lenguaje, podemos ponerle uno **fijo** (lo forzamos) al número 1, por ejemplo. Si lo forzamos, nos ahorramos el tener que preguntarle.

Aquí

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

En el constants.h

Texto

Descripción generada automáticamente

En el shader\_platform.h

Texto

Descripción generada automáticamente

Problemas: de saber dónde está y se traduce y tam de estructuras

Un hombre con un pizarrón blanco en la mano

Descripción generada automáticamente con confianza media

Tenemos HLSL y C++, yo me creo un .h de donde lo leen ambos.

Hay tipos float3 que hay en el HLSL que no están en C++. Entonces en mi .h le hago la traducción float3 a VEC3.

Para forzar un slot: nombre estructura, register

Texto

Descripción generada automáticamente

Macro que define structura y el slot. C++ pasa del slot y al HSLS le ponemos la estructura que quiere.

Los registros de la HSLS, son para 4 componentes (rgba o xyzw). Para pasar un int y VEC4 :

Dibujo de un pizarrón blanco

Descripción generada automáticamente con confianza media

Tamaño de esto: 4float \* 4 + 4 = 20 bytes

Pero si le pregunto al HLSL me dice que ocupa 32 bytes. Pq lo va a hacer alineado a 16.

Pizarrón blanco con letras negras

Descripción generada automáticamente con confianza media

El vulkan le podemos decir que alinea a otra cosa, pero aquí es así a 16.

La matriz nuestra es de 4x4 en nuestro shader\_platform.h

Yo le puedo preguntar por el camera\_fwd y el me devuelve el offset en el que se encuentra.

Habrá veces que necesite dummys, pongo algo para que luego

Nos recomienda no meter bool.

Float3 nom[2] da problemas. Así que le pongo un float4 n [2] y

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Assert si le pongo un bool y no está alineado.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Para iluminar tenemos los siguientes elementos

* P, Punto que ilumino
* Normal
* Luz Position
* Vector L: vector desde el punto P a LuzPosition

La superficie absorbe X luz dependiendo del angulo N,L

Un dibujo de un pizarrón blanco

Descripción generada automáticamente con confianza baja

Producto escalar entre N,L (normalizados) para saber el coseno del angulo. Me devuelve entre -1 y 1.

Dibujo de un pizarrón blanco

Descripción generada automáticamente con confianza media

Ligh pos la tenemos. En ese vector ahora tenemos el 0,0,0. Tenemos que hacer algo ahí. Pongo los valores y lo enchufo, pongo una entidad. Transform

Le podemos cambiar la mesh

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Satúrate. Si tengo una luz negativa y tengo que

ESPECULAR: Brilla mas si el Ojo lo tengo justo donde rebota la luz en la superficie (refleja).

Imagen que contiene interior, colgando, cuarto, reloj

Descripción generada automáticamente

Si los vectores (REFLECT y PuntoOjo) se parecen muchos, fogonazo. Esos vectores generan un angulo B

cosB()^n

Un dibujo de una persona

Descripción generada automáticamente con confianza baja

Esto va a ser casi 0 casi siempre. Pq 0.5\*0.5\*0.5\*… n

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente