

Proyecto VII - Opengl - Shader

El proyecto consiste en programar unos shader para usarlos en el escenario que se les pasa junto a este documento.

Deberán de hacer una iluminación (point light) por pixel, como se vio en clase, en el proyecto adjunto los .vert y .frag que se vio en la última clase.

En la hoja de excel en donde se encuentran las notas, hay una columna que señala los shaders que van a tener que programar.

Bloque A.

- Difuso: Biased diffuse (01.jpg)
- Specular: Glossy sharp (04.jpg)

Bloque B:

- Difuso: Seeliger diffuse (02.jpg)
- Specular: Minnaert shading (07.jpg)

Todos los equipos deberán hacer:

- Specular: Cook - Torrence (05.jpg y 06.jpg)
- Fresnel (03.jpg).

En el documento tache unas lineas de código que pueden ser un poco confusa, ya que este código es de Renderman Shading Language y maneja un poco distinto ciertas cosas, sin embargo, la aritmetica y logica de los shaders pueden ser llevado sin problema a Opengl shading Language.

Para el algoritmo de Cook - Torrence, necesitan una funcion fresnel, usen la siguiente implementación:

```
float fresnel(vec3 normal, vec3 light, float indexR)
{
    // Note: compute R0 on the CPU and provide as a
    // constant; it is more efficient than computing R0 in
    // the vertex shader. R0 is:
    // float const R0 = pow(1.0-refractionIndexRatio, 2.0)
    // / pow(1.0+refractionIndexRatio, 2.0); Fresnel Reflection
    //WP-01401-001_v01 7
    // light and normal are assumed to be normalized

    float R0 = pow(1.0-indexR, 2.0)/pow(1.0+indexR, 2.0);
```

```
return R0 + (1.0-R0) * pow(1.0-dot(light, normal), 5.0);  
  
}
```

Cabe mencionar que en el código lo que llaman la variable I es lo equivalente a la variable C que vimos en clase, es decir, es el vector de la cámara.

Voy a colocar a continuación las teclas para cada variable de los shaders, cada equipo solo programará las teclas de los shaders asignado.

Teclas:

- 1: activa el specular asignado para cada equipo
- 2: activa el Cook Torrance specular.
- 3: activa el efecto fresnel, este efecto debe verse solo, es decir, no se combina con el efecto diffuse ni especular.
- 4. desactiva el efecto fresnel.
- q: aumenta en 0.15 el índice de refracción (cook -torrence)
- w: disminuye en 0.15 el índice de refracción (cook -torrence), este valor no puede ser menor que 1.
- a: aumenta en 0.01 la variable m (cook -torrence)
- m: disminuye en 0.01 la variable m (cook -torrence) este valor no puede ser menor que 0.
- e: aumenta en 0.02 la variable sharpness (glossy), no puede ser mayor que 1
- r: disminuye en 0.02 la variable sharpness (glossy) este valor no puede ser menor que 0.
- d: aumenta en 0.02 la variable roughness(glossy)
- f: disminuye en 0.02 la variable roughness(glossy) este valor no puede ser menor que 0.1.
- t: aumenta en 0.1 la variable k (Minnaert), no puede ser mayor que 1.
- y: disminuye en 0.1 la variable k (Minnaert).
- u: aumenta en 0.02 la variable η (fresnel).
- i: disminuye en 0.02 la variable η (fresnel) este valor no puede ser menor que 0.
- j: aumenta en 0.1 la variable Kfr (fresnel).
- k: disminuye en 0.1 la variable Kfr (fresnel) este valor no puede ser menor que 0.
- z: aumenta en 0.1 la variable bias (bias y fresnel).
- x: disminuye en 0.1 la variable bias (bias y fresnel).
- c: aumenta en 0.1 la intensidad especular.
- v: disminuye en 0.1 la intensidad especular, este valor no puede ser menor que 0.
- b: aumenta en 0.1 la intensidad especular.
- n: disminuye en 0.1 la intensidad difusa, este valor no puede ser menor que 0.

Especificaciones del proyecto:

1. El proyecto se puede crear en **parejas**.
2. El proyecto tiene un **15%** de nota.
3. La entrega es el **sábado 28/02**.
4. Deberán enviarme por correo (comprimido en .rar) el código fuente (solo los archivos .cpp y el .vert y .frag, no es necesario comprimir todo el proyecto) y el ejecutable.

IMPORTANTE: el asunto del correo debe estar escrito con el siguiente formato: **“Graficas I 2015 - Proyecto VI - Opengl - NombresEstudiantes”**.

NOTA: es importante que toda variable que pasen del CPU al GPU usando las variables del tipo uniform sean utilizadas en el código, les va a suceder mucho que cuando compilen no les corra el programa, si eso pasa verifiquen:

1. Que los nombres de las variables uniform sean iguales a como se establece en el .cpp
2. Que la variable sea utilizada en el código.
3. Tener una variable no declarada correctamente en el .frag o .vert.

En clase voy a detallar un poco mejor esto.

Imágenes de Ejemplo:





