GRÀFICS I VISUALITZACIÓ DE DADES

Práctica 1
(Esfera)

Víctor Fernández
David García
12/04/2016

Sumario

Objetivos de la práctica.

En esta práctica aprendemos a usar un material, las tres tipos de luces, calcular las normales para vértices y juntar todo con la formula de blinn-phong para que salga como resultado un objeto iluminado según su iluminación y material.

Además también aprendemos a implementar una textura a una esfera.

Explicación de lo que hemos hecho.

Aplicar Material

En este apartado tuvimos que implementar en material.h y material.cpp una serie de variables y funciones. Las variables que generamos en el material.h tres componentes de tipo vec3, una para la kd, ks y ka, y un float para guardar el shininess.

En material.gpp hemos tenido que crear un constructor donde no se le pasa nada y generamos nosotros a mano el material y otro constructor sobrecargado donde ya nos pasan el kd, ks, ka y shininess. Así como un método llamado toGPU(QGLShaderProgram *program) donde enviamos el material a la GPU en una Struc de forma Uniform, que eso quiere decir que para cada GPU es igual.

Aplicar Luces

En este punto para nosotros fue el más complicado, ya que nos costo entender, por un lado, bien la teoría y por otro lado se nos complico un poco el código al implementar las luces con herencia.

Cuando nos descargamos el proyecto ya venia una clase llamada llum.h y llum.cpp. Nosotros creamos 3 clases más que heredan de esta, una para cada tipo de luz (Puntual, Direccional y SpotLight).

La forma de pasar las luces es en un vector de luces, que este vector de 3 posiciones, en

forma de Struc Uniform con todos los componentes de la luz que necesitaremos en la GPU. Nosotros hemos determinado que la posición 0 es para la puntual, 1 para la direccional y 2 para la spotlight.

Aplicar los Shader con blinn-phong

Aquí primero tuvimos que calcular las normales para los vértices, para ello hay que calcular la normal de la cara y a continuación, para cada vértice de esa cara, se le suma la normal acumulada a ese vértice. Para hacer eso creamos una tabla de adyacencia en el que se asigna para cada vértice su normal acumulada.

Posteriormente pasamos las normales a la GPU a través del *Buffer* que comparten la CPU y GPU.

Una vez nos funciono las normales y se veían bien, nos pusimos a implementar la formula de blinn-phong, aquí tuvimos algún problema ya que al pasar las coordenadas de la luz y la dirección le pasábamos un vec3 en lugar de un vec4. Cuando conseguimos arreglar este error, creamos los programas necesarios, es decir, para un vshader y fshader para Gouraud, otro para Phong y Toon.

Para Phong y Gouraud se aplica de la misma forma, solo que uno es en vertex shader (Gouread) y Phong se calcula en el fragment.

En el caso de Toon, es aplicar una formula que esta en el PDF, nosotros usamos los valores que hay ahí para la interpolación por normales, pero según la intensidad multiplicamos el kd por el valor que queremos que salga.

Para seleccionar el tipo de programa que se esta usando en cada momento, creamos en GLWidget un enum con los tres tipos de programas. También creamos una variable de tipo del enum, donde guardamos cual es el último programa cargado. Así como necesitamos crear un array de programas, para poder seleccionar entre ellos.

Aplicar las texturas

Para aplicar las texturas trabajamos sobre la clase Objetecte y lo que hacemos primero es

cargar la textura en memoria. Una vez hemos calculado las normales que le vamos a pasar a la GPU, transformamos las coordenadas xyz a uv, eso se hace con el arco tangente y arco sino de la normal.

Para pasar las texturas a la GPU hacemos uso del *Buffer* compartido entre CPU y GPU.

Opcionales y puntos implementados.

Opcional

Como opcional hemos implementado poder mover la esfera con los cursores y las teclas A, S, D, W. Para implementar esto nos hemos basado en parte del código de cup.

Puntos implementados

Los puntos implementados son:

- Poder añadir una nueva luz en el menú, funciona con los 3 Shaders.
- Encender y apagar la luz seleccionada, funciona con los 3 Shaders.
- Poder cambiar la intensidad de la luz seleccionada y si tiene coordenadas también se pueden modificar, funciona con los 3 Shaders.
- Seleccionar el Shader a mostrar.
- Selecciona el Shader y activar/desactivar las texturas para Phong y Gouraud.
- El opcional mencionado anteriormente, mover la esfera.