

Práctica 2: Iluminación en GLSL

Informática Gráfica
Grado en Diseño y Desarrollo de Videojuegos

Miguel Ángel Otaduy
Marcos García Lorenzo

Tabla de contenido

Introducción 3

Normativa..... 3

Parte guiada 3

 Paso 1: sombreado de Gouraud 3

 Paso 2: sombreado de Phong..... 4

 Paso 3: texturas..... 4

Parte obligatoria..... 4

Parte opcional 4

Introducción

El objetivo de esta práctica es que el alumno aplique las técnicas de iluminación vistas en clase en un cauce gráfico real. Se repasarán conceptos como: modelos de iluminación, modelos de sombreado, tipos luces... Cada grupo de prácticas deberá implementar la funcionalidad detallada en este guion de prácticas.

Normativa

Esta práctica se divide en tres bloques:

- Bloque guiado: este bloque se realizará de forma guiada en el aula de prácticas. La **asistencia es obligatoria**. El bloque guiado no es re-evaluable por lo que la no asistencia acarreará el suspenso de la asignatura.
- Bloque obligatorio: los grupos de prácticas deberán realizar este bloque de forma supervisada (pero no guiada) en las horas de laboratorio que se habiliten para tal efecto. El trabajo que no pueda realizarse dentro de las horas de laboratorio podrá realizarse fuera del horario de la asignatura. El trabajo realizado, tanto en el bloque guiado como en el bloque obligatorio, se evaluará mediante un examen de prácticas que podrá aprobarse tanto en la convocatoria de diciembre como en la de junio. El código de ambos bloques se entregará junto, a través de campus virtual en el plazo que indique en la página web de la asignatura. Deberá adjuntarse una pequeña memoria explicativa.
- Bloque opcional: las partes opcionales se realizarán por el alumno fuera del horario de la asignatura. Las tareas de este bloque no son obligatorias pero la nota final dependerá en gran medida de las partes opcionales realizadas. El bloque opcional se entregará junto con una memoria explicativa a través de la herramienta habilitada en campus virtual, en el plazo que se indique.

Las prácticas podrán realizarse tanto de forma individual como en grupos de dos personas.

Parte guiada

Paso 1: sombreado de Goureaux

En esta primera parte de la práctica se implementará un sombreado de Goureaux, para una única fuente de luz puntual. Las propiedades de la luz se incluirán directamente en el código de los *shaders* (*hard-coded*).

1. El profesor explicará el entorno compuesto de:
 - a. Un proyecto de Visual Studio que implementa la funcionalidad del cliente.
 - b. La librería IGlib que esconde la funcionalidad de OpenGL utilizada.
 - c. El modelo 3D de un cubo.
 - d. El esqueleto de un *shader* de vértices (*shader.v0.vert*) y de un *shader* de fragmentos (*shader.v0.frag*), sobre los que se implementará la funcionalidad requerida.
2. Copia los ficheros *shader.v0.vert* y *shader.v0.frag* a *shader.v1.vert* y *shader.v1.frag*.

3. Modifica el fichero `main.cpp` para que utilice los nuevos *shaders* de vértices y fragmentos.
4. Modifica la funcionalidad del *shader* de vértices y del *shader* de fragmentos de forma que el color final dependa de la intensidad ambiental de la fuente.
5. Modifica la funcionalidad del *shader* de vértices de forma que el color final dependa de la intensidad ambiental y de la intensidad difusa de la fuente.
6. Modifica la funcionalidad del *shader* de vértices de forma que el color final dependa de la intensidad ambiental, de la intensidad difusa y de la intensidad especular de la fuente.

Paso 2: sombreado de Phong

1. Copia los ficheros `shader.v1.vert` y `shader.v1.frag` a `shader.v2.vert` y `shader.v2.frag`.
2. Modifica el fichero `main.cpp` para que utilice los nuevos *shaders* de vértices y fragmentos.
3. Modifica la funcionalidad del *shader* de vértices y del *shader* de fragmentos de forma que el color final dependa de la intensidad ambiental de la fuente.

Paso 3: texturas

1. Copia los ficheros `shader.v2.vert` y `shader.v2.frag` a `shader.v3.vert` y `shader.v3.frag`.
2. Modifica el fichero `main.cpp` para que utilice los nuevos *shaders* de vértices y fragmentos.
3. Modifica la funcionalidad del *shader* de vértices y del *shader* de fragmentos de forma que el color del objeto se obtenga de una textura.
4. Modifica el fichero `main.cpp` para que asigne una textura emisiva al objeto (`IGlib::addEmissiveTex(objId, "../img/emissive.png")`).
5. Modifica la funcionalidad del *shader* de fragmentos de forma que al color del objeto se le añada una componente emisiva.
6. Modifica en el fichero `main.cpp` el albedo ("`color2.png`").
7. Modifica el fichero `main.cpp` para que asigne una textura especular al objeto (`IGlib::addSpecularTex(objId, "../img/specMap.jpg")`).
8. Modifica la funcionalidad del *shader* de fragmentos de forma que las propiedades especulares del material se obtengan de la textura.

Parte obligatoria

En este bloque deberá implementarse la siguiente funcionalidad:

1. **Ilumina el objeto con al menos 2 fuentes de luz.**
2. **Añade una nueva funcionalidad que atenue la intensidad lumínica en función de la distancia del objeto a la fuente lumínica.**
3. **Implementa los siguientes tipos de luz: luz direccional, luz focal.**

Parte opcional

Este apartado describe las partes que podrán realizarse de forma opcional.

1. Implementa la técnica del *Bump Mapping*. Nota: puedes utilizar la función `IGlib::addNormalTex(objId, "../img/normal.png")` para asociar un mapa de normales al modelo. El atributo `in vec3 inTangent` da acceso a las tangentes del modelo.
 - a. Mathematics for 3D Game Programming and Computer Graphics, 3rd Edition (Capítulo 7, apartado 8).
 - b. Real-Time Rendering, 3rd Edition (Capítulo 6, apartado 7).
2. Carga un nuevo modelo desde fichero. Calcula automáticamente la normal y la tangente de cada vértice. Información adicional:
 - a. <http://www.terathon.com/code/tangent.html>
 - b. Mathematics for 3D Game Programming and Computer Graphics, 3rd Edition (Capítulo 7, apartado 8).
3. Puedes sugerir nuevas partes opcionales a los profesores de prácticas.