Computación Gráfica - ST0275
Examen Final

Departamento de Informática y Sistemas
Universidad EAFIT

Noviembre, 2014

Notas importantes:

- Usted debe entregar el examen antes de las 18:30, por EAFIT interactiva.
- Se debe enviar un archivo zip con: código fuente, código ejecutable, proyecto para compilar el programa, screenshots de la ejecución.
- Incluir un archivo README.TXT explicando cómo compilar y ejecutar el programa.
- Incluir archivos de prueba, si su programa los utiliza.
- La respuesta se puede implementar en Java o Unity. Si va a utilizar alguna otra plataforma, hable PREVIAMENTE con el profesor. Para el punto 1 se debe utilizar Java. Para el punto 2 se puede utilizar Java o Unity. Para el punto 3 se debe utilizar Unity.
- El examen se resuelve en los mismos grupos de la práctica.
- Cada hora, o fracción de demora en la entrega, conlleva una rebaja de 0,5 en la nota.

1. Ray Tracing

El programa de Iluminación Local que se adjunta, utiliza el método de *Ray Tracing* para dibujar triángulos. Modifique el programa para que permita también dibujar esferas.

Para esto se debe hacer lo siguiente:

- Escribir el método `findIntersections(Ray ray)` en la clase `Sphere` que determina si hay intersección entre el rayo que se recibe como parámetro y la esfera.
- Completar el método `throwRays()` de la clase `Scene.java` luego de la línea 126 para invocar el método `findIntersections(Ray ray)` de la clase `Shere`.

El programa principal para correr el programa está en la clase `LocalIlluminationIncomplete`.

Para la escena que está codificada en el programa principal, la imagen que se debe dibujar es como la de la figura 1.

2. Intersección de triángulos

Crear un programa que determine si dos triángulos se intersectan. Si los triángulos se intersectan, se dibujan en rojo, de lo contrario se dibujan en verde.

Los vértices que forman los triángulos se leen desde un archivo de texto.

Para verificar si los triángulos se intersectan, se va a hacer lo siguiente: Se toma cada uno de las aristas del triángulo A como un rayo y se determina si dicho rayo se intersecta con el triángulo B. Notar que si hay intersección, el parámetro del rayo debe estar entre 0 y 1.

Para determinar si un rayo se intersecta con un triángulo, se utilizan las ecuaciones descritas en la presentación *rayTracingMath.ppt*. Notar que la intersección entre un rayo y un triángulo está codificada en

el método `findIntersections(Ray ray)` de la clase `Triangle` en el código de Iluminación Local que se adjunta, este código se puede utilizar como punto de partida.

Al enviar su solución, muestre un ejemplo con triángulos que *si* se intersectan y otro con triángulos que *no* se intersectan.

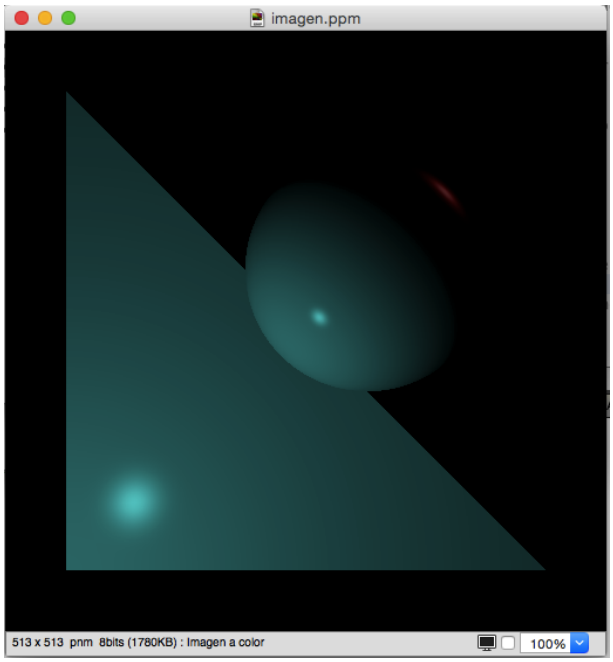

Figura 1: Imagen que debe producir el programa del punto 1

3. Unity

Crear un pequeño paquete gráfico en Unity3D. El paquete gráfico debe leer de un archiivo lo siguiente:

- Las coordenadas 3D de cada vértice que compone el objeto
- Los subíndices de cada polígono que conforma el objeto
- El material del objeto
- La posición y orientación de la cámara

A partir de dicho objeto, el programa en Unity3D debe dibujar el objeto.

A manera de ejemplo, utilice las coordenadas de la casita en 3D, que se utilizó en varios retos a lo largo del semestre.

Firma

Al firmar afirmo que lo que entregaré en este examen será fruto de mi trabajo individual. No recibiré ayuda indebida de parte de otra persona. No utilizaré código de Internet.

Punto a trabajar: ———-

_____-

Firma

_____

Fecha