

# Computación Gráfica

## Segundo Trabajo Obligatorio

### 1. Problemas obligatorios

El problema a resolver consiste en dibujar una escena simple, compuesta únicamente por esferas, utilizando la técnica de *RayTracing*.

El programa comienza desplegando su interfaz de usuario (una ventana con el título “*RayTracer*”) e inmediatamente comienza el renderizado de la escena.

El usuario únicamente puede cerrar la aplicación, sin mayor interacción que esa (se recomienda permitir la interrupción en cualquier momento, se haya completado o no el renderizado).

El trabajo se deberá implementar tomando como base el *Framework\_Raytracer*, disponible en webasignatura, en el Lenguaje de Programación C.

Los problemas obligatorios a resolver son:

- Carga del archivo de configuración de la escena (puede utilizar el parser provisto por la Cátedra<sup>1</sup>).
- Renderizado de la escena, compuesta por esferas, utilizando la técnica de *Raytracing*.
- Sombreado de las esferas con iluminación difusa y especular, soportando cualquier cantidad de luces puntuales.
- Sombras proyectadas por las esferas.
- Reflejo de la escena sobre la superficie de objetos reflejantes.

### 2. Problemas opcionales

La Para complementar la nota de los otros trabajos (parcial y primer obligatorio) puede implementar hasta 15 puntos de entre los trabajos complementarios que se detallan a continuación:

- Refracción de rayos al pasar por esferas de diferentes materiales. Tenga en cuenta que debe modificar los archivos de entrada y el parser para definir el índice y coeficientes de refracción de los materiales. [10 pt]
- Intersección de rayos con otras primitivas. Estas pueden ser elipses centradas en el origen de coordenadas, triángulos o cualquier otra figura geométrica para la cual la intersección pueda ser calculada. [10 pt]
- Optimización utilizando técnicas de *HPC* (threading)<sup>2</sup>. [5 pt]
- Optimización por subdivisión espacial<sup>3</sup> [15 pt]

---

1 [http://webasignatura.ucu.edu.uy/file.php?file=%2F58%2FMaterial\\_del\\_curso%2FArchivos%2FEscenas\\_y\\_Datos%2FParser\\_Raytracer.zip](http://webasignatura.ucu.edu.uy/file.php?file=%2F58%2FMaterial_del_curso%2FArchivos%2FEscenas_y_Datos%2FParser_Raytracer.zip)

2 Linux: POSIX threads <http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html>

Windows: pthreads for win32: <https://www.sourceware.org/pthreads-win32/>

Mac: <https://developer.apple.com/library/mac/documentation/Darwin/Reference/ManPages/man3/pthread.3.html>

3 [http://webasignatura.ucu.edu.uy/pluginfile.php?file=%2F15542%2Fmod\\_folder%2Fcontent%2F0%2FReferencia%2FPaper-Subdivision-Uniforme.pdf&forcedownload=1](http://webasignatura.ucu.edu.uy/pluginfile.php?file=%2F15542%2Fmod_folder%2Fcontent%2F0%2FReferencia%2FPaper-Subdivision-Uniforme.pdf&forcedownload=1)

### 3. Entrega

La entrega consistirá en el envío del código fuente de un programa C mediante *WebAsignatura* el día Domingo 22 de Noviembre de 2015. Su entrega deberá incluir todos los archivos necesarios para compilar la aplicación.

Su aplicación deberá ser “autocontenida”, y deberá restringirse de intentar acceder a archivos de configuración, modelos, imágenes, etc. en rutas predefinidas como “C:\”, “C:\Documents and Settings” u otros. Todas las rutas utilizadas por su aplicación deberán ser relativas y estar formadas por el carácter “/” (no “\”). Esta acotación incluye tanto las rutas a los recursos mencionados, como las instrucciones `#include` en los archivos fuente y la declaración de los fuentes a compilar en el archivo *Makefile*.

Los trabajos son en grupos de hasta 2 personas y deberán ser de su propia autoría (Trabajos Originales). Es suficiente con que uno de los integrantes del grupo suba la entrega.

Esta entrega es de carácter eliminatorio y debe consistir de un archivo comprimido con el nombre “*Raytracer\_<Apellidos>.zip*” conteniendo todos los archivos necesarios para compilar y ejecutar estas aplicaciones, además de un archivo *README* especificando los integrantes del grupo, las características extras que fueron implementadas, las teclas configuradas para interactuar con su aplicación, librerías externas que haya utilizado y cualquier otra información que considere relevante para poder evaluar su trabajo.

### 4. Calificación

Los trabajos se calificarán según los siguientes criterios: El conjunto de Características Obligatorias tiene un valor de 30% de la nota total del curso.

Se penalizará hasta con un 15% de la nota a los trabajos que aborten por errores o por falta de memoria (debido a Fugas). Si su trabajo tiene Fugas de Memoria, pero no llega a abortar, de todos modos repercutirá en su calificación, pero en menor medida.

La nota de cada característica extra se suma directamente a la nota final del curso, con un máximo de 15% en total.

Le sugerimos utilizar herramientas como *valgrind* o equivalentes para asegurarse de no hacer un uso indebido de la memoria y, en caso contrario, poder detectar y reparar las posibles Fugas y Errores de Acceso de su programa.

No se aceptarán trabajos desarrollados en lenguajes de programación distintos de C.

Sus fuentes deberán poder ser compilados sin modificaciones por el ambiente de desarrollo provisto por la Cátedra. No se aceptarán trabajos que requieran utilizar algún IDE o editor específico.

**Los trabajos no originales serán considerados Plagio Académico y causarán la pérdida instantánea del curso y la sanción académica correspondiente.**