Computación Gráfica
Primer Trabajo Obligatorio

1. Problemas obligatorios

Tomando como punto de partida los trabajos prácticos desarrollados hasta la fecha, implemente una escena 3D interactiva simple.

Los problemas obligatorios a resolver son:

- Rasterización de modelos *OBJ,* dibujando triángulos sólidos o una "malla de alambre".
- Determinación de superficies ocultas por medio de *Z-Buffer* y optimización por *Backface culling*.
- Soporte para al menos 8 luces, *puntuales* y/o *direccionales*.
- Sombreado por medio de la técnica *Gouraud Shading*.
- Mapeo de texturas, con filtrado bilineal.
- Implementación de una gestor de escena en forma de árbol (*Scene Graph*). Esta estructura debe tener las siguientes características:
  - Cada nodo del árbol debe tener los atributos *posición*, *escala*, y *rotaciones*, de los cuales se deducirá la matriz modelo del objeto contenido en dicho nodo.
  - Cada objeto debe heredar las tranformaciones de su objeto padre.
  - Cada nodo del arbol puede alojar una geometría dibujable (*OBJ*) o una luz.
- Implementar una cámara movil controlable por entradas de teclado y/o mouse.

2. Problemas complementarios

- Crear primitiva *Plano*, que cree la geometría de un plano horizontal (sobre los ejes *XZ*), dado el tamaño y la cantidad de segmentos que lo componen. Por ejemplo, la primitiva *plano(1000.0f, 10, 10)* crearía un plano cuadrado, de 1000 unidades de lado, formado por una grilla de 100 segmentos (10 de largo y 10 de ancho) de 100 uidades de lado cada uno. ***Nota***: Se recomienda estudiar el tipo de renderizado *GL_TRIANGLE_STRIP*. (+10 pts).
- Cargar la escena desde un archivo de configuración. Debe definir su propio formato de archivo y parser. (+10 pts).
- Sombreado por medio de *Phong Shading* u otra técnica, utilizando shaders *glsl.* (+20 pts).

- Animaciones en función del tiempo. (+10 pts).
- Mapas de iluminación (*Light Maps*) (+10 pts).
- Cualquier otra característica que, a consideración de los docentes, aporte valor extra a su trabajo.

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.

Importante: Su aplicación deberá ser "autocontenida", y deberá restringirse de intentar a acceder a archivos de conguración, 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 *"Raster_<Apellidos>.zip"* conteniendo todos los archivos necesarios para compilar y ejecutar estas aplicaciones, ademas de un archivo README especificando los integrantes del grupo, las características extras que fueron implementadas, las teclas configuradas para iteractuar 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:

1. El conjunto de Características Obligatorias tiene un valor de 80 puntos, sobre los 100 de la nota total del trabajo.
2. Cada Característica Adicional implementada otorga un valor diferente, el cual se detalla al final de la descripcion de cada una de ellas en este documento.
3. El valor final del trabajo será truncado a 100 puntos, en caso de que se sobrepase ese límite, y tendrá un peso del 30% sobre la nota final del curso,

Se penalizará hasta con 20 puntos a los trabajos que aborten por errores o por falta de memoria (debido a Fugas). Si su trabajo tiene Fugas de Memoria graves, pero no llega a abortar, de todos modos repercutirá en su calificación, pero en menor medida.

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 modicaciones 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.