

1er Obligatorio de Introducción a la Computación Gráfica

Videojuego sobre OpenGL y SDL

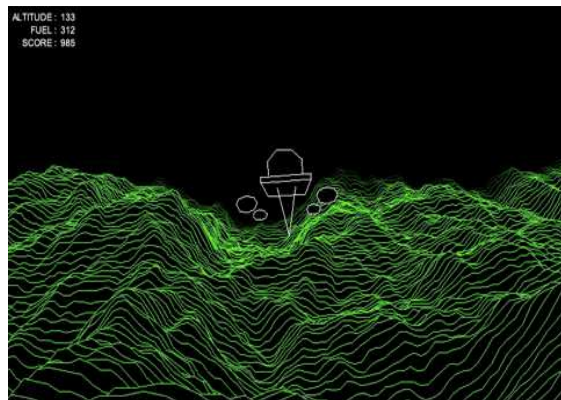
Introducción

El obligatorio se centra en conceptos vistos en el curso así como en las librerías OpenGL y GLU (para efectuar el render), y SDL (para manejo de dispositivos de entrada y creación de ventanas, entre otros).

Este es un obligatorio de Computación Gráfica, no de diseño de videojuegos. Se le dará mayor prioridad a todos aquellos aspectos que tengan que ver con el logro de efectos gráficos interesantes y asuntos relativos al "tiempo real", por encima de los aspectos relacionados con la estrategia, las reglas del juego o conceptos no afines a la computación gráfica.

Obligatorio (Lunar Lander)

Este obligatorio consiste en desarrollar una variante en 3D del "popular" juego *Lunar Lander*, publicado por *Atari Inc.* en el año 1979 (aunque la primera versión es de diez años antes). Es un juego estilo *ARCADE* en donde el jugador debe guiar una nave por el espacio, activando sus cohetes, con el objetivo de que alunice con cuidado sobre una superficie horizontal. El "combustible" del que se dispone es la restricción importante para el logro del objetivo.



Videos y aplicaciones del Lunar Lander se encuentran en:

<http://seb.ly/2009/04/lunar-lander-3d-in-5k/>

No se evaluará el grado de reproducción o copia de los detalles estéticos del juego. Se evaluará que la aplicación cumpla con los objetivos y características básicas del juego original, así como la innovación resultante del uso de elementos de computación gráfica. **La creatividad, especialmente en los efectos gráficos, influirá positivamente en el puntaje.**

Requerimientos del videojuego

El conjunto de requerimientos que se detallan a continuación, representan los mínimos necesarios para aprobar este trabajo. La ausencia de alguno de los puntos especificados, sin una correcta justificación, significará la no aprobación del obligatorio (y la consiguiente pérdida del curso).

- El juego estará gobernado “principalmente” por las flechas del teclado (o teclas direccionales).
- Será posible rotar el juego para cambiar el punto de vista de observación a través del uso del mouse, mientras se presiona el botón derecho.
- Se podrá detener el juego (tecla P), y salir (tecla Q).
- **Ajustes (settings):** Se deberá disponer de una interfaz para ajustar los siguientes parámetros:
 - Velocidad del juego: que será independiente de las imágenes por segundo generadas por la aplicación, y será independiente de la computadora en la que sea ejecutada la aplicación.
 - Estados posibles: wireframe (on/off); facetado/interpolado; texturas (on/off). Se deberá poder cambiar de estado durante la ejecución.
 - Ubicación y color de por lo menos una luz.
- Se cargarán y renderizarán texturas y modelos 3D (por lo menos una textura y un modelo). No es mandatorio que los modelos cargados formen parte del juego (pueden formar parte del fondo o de algún efecto visual). **La programación del “render” del modelo 3D debe ser realizada por los estudiantes.**
- Se deberá generar el terreno a partir de una malla regular donde la altura y ubicación de cada vértice corresponda con un píxel de un mapa de altura. El mapa de alturas puede generarse utilizando algún software realizado por terceros, como por ejemplo “Hme” (Height map editor), descargable desde <http://hme.sourceforge.net/>.
- La cámara podrá ubicarse en lugares predeterminados:
 - Vista aérea.
 - Cámara que acompaña al Lunar Lander desde cerca.
- Se implementará al menos un tipo de “objeto enemigo” (por ejemplo, otra nave que se mueva aleatoriamente en la escena, donde cada choque implique una penalización).
- Se contemplará que la nave se encuentra bajo los efectos del campo gravitatorio lunar, por tanto deberá “simular” un comportamiento físicamente realista.
- Cuando ocurra un encuentro de la nave con la superficie lunar, deberá controlarse a qué velocidad ocurre y qué orientación tiene la nave y el terreno. Si la velocidad de la nave es inferior a cierto umbral, y la nave y el terreno están en posición aproximadamente horizontal, entonces el alunizaje se considerará exitoso, en caso contrario el jugador y la nave serán perjudicados.
- El game HUD básico (puntaje, nivel, etc.) será dibujado con una proyección ortogonal a la ventana. Se recomienda que tenga elementos 3D.
- El peso total de la aplicación (ejecutable + modelos + texturas) no deberá exceder los 10 MBytes. Se deberán almacenar las imágenes en formatos con compresión.

Requerimientos opcionales

Los requerimientos que se detallan a continuación representan elementos que tienen un valor agregado pero no influyen en la aprobación del trabajo obligatorio. La realización de requerimientos opcionales no exime la realización de algún punto especificado como obligatorio.

- Implementación de un generador automático de mapas de altura que generará mapas aleatorios.
- Posibilitar el almacenamiento en disco del mapa generado.
- Implementar una lluvia de meteoritos que cae continuamente sobre la luna. Tanto el Lunar Lander como la superficie lunar se verán afectados por esta. La colisión de cada meteorito con la superficie lunar generará un cráter, que modificará el mapa de alturas.

Recursos disponibles

En los equipos PC Windows de las salas de laboratorio de la Facultad de Ingeniería, los alumnos pueden ingresar con el número de cédula de identidad como usuario, y la contraseña de bedelía como password. En la sala 115 se han instalado unas 30 computadoras con hardware gráfico NVIDIA GeForce 480. Sobre ese hardware se instalaron bibliotecas gráficas y herramientas de desarrollo adecuadas.

Allí tienen instalado OpenGL y GLU.

El paquete SDL de desarrollo puede ser descargado desde la página oficial de la librería.

<http://www.libsdl.org/>

También disponen del IDE de desarrollo Visual Studio 2008 y 2010 (su utilización **NO** es obligatoria).

Opcionalmente se pueden utilizar otros compiladores/IDEs C++:

- Mingw dentro del IDE Code::Block

<http://www.codeblocks.org/>

- Mingw dentro del IDE Bloodshed Dev-C++

<http://www.bloodshed.net/devcpp.html>

¿Trabajo individual o colectivo?

Se deberán formar grupos de 3 alumnos (en algún caso 2 alumnos, pero con la debida justificación). La formación de los grupos queda por cuenta de los estudiantes. Si fuese necesario, pueden utilizar el newsgroup para buscar compañeros y armar los grupos.

Defensa del obligatorio (jueves 24 de Mayo)

La defensa consta de 2 partes, una presencial y otra en base a la documentación presentada.

- **Defensa presencial:**

El día de la defensa se presentarán los grupos (con todos sus integrantes) y entregarán el ejecutable y la documentación. Los grupos realizarán una breve demostración de sus aplicaciones, que deberán ejecutar sobre el sistema operativo Windows, en las máquinas de los laboratorios (salón 115). Es importante que prueben sus aplicaciones antes de la defensa en las máquinas con las tarjetas GeForce 480, para evitar contratiempos innecesarios. También podrán comentar oralmente algunos aspectos de la aplicación y de la documentación. Los docentes harán preguntas sobre el trabajo realizado.

- **Documentación:**

Durante la defensa presencial los estudiantes entregarán al docente una documentación que consistirá en:

- Una documentación impresa en la que se explica:
 1. Arquitectura de la aplicación.
 2. Estructura de datos interna desarrollada para manejar los diferentes estados de la aplicación.
 3. Cómo se utilizó OpenGL, en especial cómo se utilizaron funcionalidades de Display List y comandos relacionados con los Vertex Arrays.
 4. Cómo resolvieron el modelado de objetos, los temas de iluminación, textura, y otros aspectos que hacen a los temas gráficos de la aplicación.
 5. Agregar un capítulo de referencias donde se mencione la documentación utilizada. Si se utilizaron partes de código fuente extraído de otras aplicaciones, explicar cuáles fueron y de dónde se extrajeron. Lo mismo para ideas extraídas de otras aplicaciones.
- Un CD con el código fuente, el ejecutable y el archivo de la documentación impresa.

No está permitido entregar la documentación por mail, ni en otro momento distinto al de la defensa presencial.

En base a la evaluación de ambas instancias se definirá la nota final del obligatorio.

Parece obvio, pero no está permitido utilizar códigos que resuelvan el obligatorio de forma trivial.

¡Concurso!

Como en años anteriores, este año se realizará un concurso entre los trabajos presentados. El concurso no influirá en la nota y no es de participación obligatoria.

La mecánica del concurso consiste en que se pondrán los ejecutables en el sitio web de la asignatura, y los nombraremos con un código identificador de forma que nadie sepa quiénes son los autores (a menos, claro, que Uds. hagan aparecer sus nombres en la salida gráfica). Luego, en el momento de la entrega del 2º obligatorio, tendrán la posibilidad de votar según una metodología que les presentaremos a su debido tiempo.

Sugerencias

- Utilizar la documentación provista en www.opengl.org
- Examinar videos y jugar ejemplos del juego para extraer ideas que aporten a lo gráfico.
- Asistir a las clases de consulta, los martes a las 19h.
- Utilizar el news para intercambiar opiniones sobre problemas, dudas, en lo que respecta a la letra, a la programación en OpenGL, etc.