

1^{er} Obligatorio de Introducción a la Computación Gráfica

Videojuego sobre OpenGL y SDL

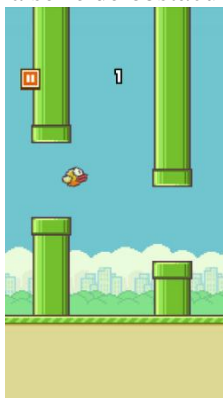
Introducción

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 conceptos).

Este es un obligatorio de Computación Gráfica, no de diseño de juegos. 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.

Ejercicio (*Flappy Bird*)

Este ejercicio consiste en desarrollar una variante en 3D del juego *Flappy Bird*, desarrollado por *Dong Nguyen* y publicado por *.GEARS Studio* en el año 2013. Es un juego de estilo *desplazamiento-horizontal* en donde un ave debe sortear una serie de obstáculos, para lograr la mayor cantidad de puntos posible.



<http://flappybird.io/>

Videos demostrativos del juego y sus variantes: <http://www.youtube.com/watch?v=wmnpxUMMfu0>,
<http://www.youtube.com/watch?v=yKudFB6wNl4>, <http://www.youtube.com/watch?v=FoGpynco2IM>,
<http://www.youtube.com/watch?v=NdXnvmuB23Y>

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 obligatorio. La ausencia de alguno de los puntos especificados sin la correcta justificación significará la no aprobación del obligatorio (esto es, la pérdida del curso).

- El programa entregado debe ejecutar sobre el sistema operativo Windows, en las máquinas de los laboratorios (salón 115).
- El juego está gobernado por las flechas del teclado (o teclas direccionales).
- Es 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 puede detener el juego (tecla P), y salir (tecla Q).

- **Ajustes (settings):** Se debe disponer de una interfaz para ajustar los siguientes parámetros:
 - Velocidad del juego. El juego debe poder variar su velocidad manteniendo aproximadamente constante la cantidad de imágenes generadas por segundo. Para esto se debe multiplicar al tiempo transcurrido entre frames por un valor acorde a la velocidad del juego.
 - Estados posibles: wireframe (on/off); facetado/interpolado; texturas (on/off). Cada estado tiene dos valores y se los puede ajustar de forma independiente. Se debe poder cambiar de estado durante la ejecución.
 - Dirección y color de por lo menos una luz. Las direcciones y colores pueden tener valores predeterminados o se puede implementar una interfaz para poder ajustar sus valores.
- Sobre cargado y renderizado de texturas y modelos 3D (por lo menos una textura y un modelo). No es mandatorio que los modelos cargados formen parte de la simulación del juego (pueden formar parte del fondo o de algún efecto visual).
- El nivel (disposición de los obstáculos) tiene que ser definido en un archivo XML. Se recomienda utilizar alguna de las siguientes librerías:
 - TinyXML (<http://sourceforge.net/projects/tinyxml/>)
 - PugiXml (<http://pugixml.org/>)
- Game HUD básico (puntaje, nivel, etc.) dibujados con una proyección ortogonal. Se recomienda que tenga elementos 3D.
- El peso total de la aplicación (ejecutable + modelos + texturas) no debe exceder los 10 MBytes. Se recomienda almacenar las imágenes en formatos con compresión (jpg, png, gif, etc.). Notar que no se considera los dlls para calcular el peso total.

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.

- Implementar un editor de niveles.
 - Como resultado se espera que el editor pueda guardar un archivo XML que luego será utilizado como uno de los niveles del juego.
- Implementar efectos gráficos (por ejemplo: sistema de partículas) para marcar los diferentes eventos del juego (por ejemplo: sumar un nuevo punto, choque del ave contra una columna, etc.).
- Implementar un efecto de estela para indicar el camino que está recorriendo el ave.
- Implementación del render de los modelos 3D.

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 6 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. Se puede utilizar SDL 1.2 o 2.0 indistintamente. <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 deben 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 así poder armar los grupos.

Defensa del obligatorio (lunes 26 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 presentan los grupos (con todos sus integrantes) y entregan el ejecutable y la documentación. Los grupos realizan una breve demostración de sus aplicaciones, que deben 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, 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, se hará entrega al docente la documentación que consiste 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. Explicar cómo resolvieron el modelado de objetos, los temas de iluminación, textura, y otros aspectos que hacen a los temas gráficos de la aplicación.
 4. 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.
 5. Versión utilizada de SDL.
- 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 influye 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 de forma que nadie sepa quiénes son los autores (a menos, claro, que Uds. hagan aparecer sus nombres en la salida gráfica). Luego, aquellos estudiantes que presenten el 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 del juego para extraer ideas que aporten a lo gráfico.
- Utilizar el EVA para intercambiar opiniones sobre problemas, dudas, en lo que respecta a la letra, a la programación en OpenGL, etc.