

Curso: *Programación de gráficos y animaciones con GPU usando OpenGL*

Docente: Wílmer Efrén
Pereira González

e-mail: wilmer.pereira@itam.mx
wpereira@ucab.edu.ve

I. Objetivo general del curso

El uso de tarjetas gráfica se ha popularizado enormemente debido a la disponibilidad tanto localmente como por la oferta de uso de GPU de los proveedores en la nube. En última instancia, la mayoría de las computadoras ofrecen al menos un procesador gráfico que emula una GPU para ejecutar aplicaciones gráficas. Por ende, los programadores tenemos a nuestra disposición las librerías gráficas especializadas, sin necesidad de invertir en una tarjeta gráfica ni acceder a la nube.

Por otro lado, del cine y los video juegos es un área de aplicación rentable que utiliza masivamente las tarjetas gráficas, mayormente con IDE de desarrollo especializados pero montados frecuentemente sobre la librería OpenGL.

Dado este escenario, el objetivo del curso es comprender y ejercitarse en los fundamentos de la programación de tarjetas gráficas para la generación de gráficos y animaciones en OpenGL

II. Conocimientos previos necesarios

En primer lugar, es un curso práctico de programación de gráficos y animaciones por lo que se parte de la premisa que los participantes saben programar. Aunque la librería para programar en OpenGL está en varios lenguajes de programación, el curso se dictará en C por lo que es requerido un nivel mínimo de conocimiento.

Aún cuando el curso depende de los conocimientos previos en C, se hará una tarea de nivelación para asentar las bases de lo mínimo que deben conocer los participantes sobre dicho lenguaje.

III. Competencias que desarrolla el curso

a) Generales:

- Diseñar y programar una imagen sencilla en dos dimensiones con OpenGL usando los principios de programación de tarjetas gráficas (*shaders*)

- Diseñar y programar una animación en dos dimensiones o tres dimensiones utilizando *shaders*.
- b) Especificas:**
- Comprender los principios y programar la creación y manejo de ventanas en OpenGL
 - Aplicar texturas para reducir la complejidad en el diseño de gráficos y animaciones
 - Componer gráficos y animaciones mediante el uso de transformaciones lineales homogéneas
 - Estudiar y aplicar los fundamentos de iluminación a una escena en tres dimensiones

IV. Resultado de aprendizaje del curso

Al final del curso, el estudiante estará en capacidad de modelar, diseñar e implementar gráficos y animaciones dominando algunos principios básicos de manejo de texturas e iluminación. El desarrollo de las aplicaciones en tres dimensiones dependerá de la rapidez con que los estudiantes asimilen los conocimientos y el tiempo en el que se desarrollará el curso.

V. Contenidos detallados

1. Introducción a la computación gráfica

- Dispositivos de entrada y salida y almacenamiento
- Arquitectura de tarjetas gráficas y principios de paralelismo
- Principios y estándares de audio y video

2. Principios de programación en lenguaje C

- Tipos de datos compuestos
- Apunadores
- Funciones, procedimientos y pase de parámetros

3. Manejo de ventanas en OpenGL

- Arquitectura OpenGL
- Gestores de ventanas (GLFW, SDL, freeglut, ...)
- Algoritmo de trazado de líneas
- Algoritmos de relleno de polígonos

4. Utilización de GPU para eficiencia de OpenGL
 - Pipeline gráfico
 - Programación de *shaders* con GLSL
 - Memoria CPU vs memoria GPU
5. Manejo de texturas 1D, 2D y 3D
 - Bitmap vs pixelmap
 - Buffers de visualización
 - Formato de imágenes
 - Carga y modo de aplicación de texturas con *shaders*
6. Transformaciones lineales: traslación, rotación y escalamiento en 2D y 3D
 - Traslación, rotación, escalado, reflexión e inclinación
 - Matrices homogéneas
 - Combinación lineal de matrices de transformación
7. Animaciones
 - Diseño de animaciones
 - Parámetros de reproducción
 - Animación por marcos y animación por tiempo
 - Técnicas de animación con velocidad y aceleración
 - Captura de movimientos
8. Fundamentos de la visualización en 3D
 - Perspectivas axométricas y puntos de fuga
 - Proyección ortogonal y en perspectiva para OpenGL
 - Transformaciones de cámara y transformaciones de escena
 - Vertex shader y Fragment shader
 - Uso de ratón y teclado para desplazamiento en la escena
9. Características de la iluminación: fuentes de luz, reflexión, refracción y fenómenos atmosféricos
 - Reflexión especular y difusa
 - Atenuación angular y radial
 - Mapas de iluminación por materiales
 - Superficies transparentes
 - Iluminación de múltiples fuentes

VI. Estrategias Didácticas

Los estudiantes se ejercitarán con códigos de ejemplo que se encuentran en github y, durante la clase, los ejecutarán y harán pequeñas verificaciones para comprobar su comprensión del código en C. Los ejemplos serán usando OpenGL con el gestor de ventana GLFW y, por supuesto, usando *shaders*.

Además, se implementará en clases una dinámica participativa, por grupos, donde cada estudiante compila individualmente el programa pero los estudiantes, por grupo, trabajan en las modificaciones.

La propuesta para este curso es para 35 horas, intercalando las sesiones de teoría con las experiencias prácticas. Lo ideal es que se realice una sesión inicial para fijar las condiciones de realización del curso y evaluar los conocimientos previos de los asistentes en C. En ese momento se asignarán las tareas de nivelación y el estudiante asume el compromiso de configurar el ambiente en Visual Studio que le permitirá la realización de las practicas.

Se recomienda que las 35 horas del curso se cubran en al menos dos semanas dada la densidad del contenido teórico y práctico. Para una mejor distribución podría también ser en tres semanas para facilitar la asimilación de los conceptos y consolidar la información adquirida.

VII. Evaluación

El curso es teórico-práctico por lo que tendrán exámenes teóricos y proyectos de desarrollo de aplicaciones. Además se realizará una tarea inicial en C para ajustar los diferentes niveles de dominio del lenguaje.

Uno de los proyectos será el diseño e implementación de un gráfico o figura abstracta en 2D y el segundo una animación, probablemente inspirada en la figura que montaron para el primer proyecto.

Los porcentajes designados para cada actividad son:

- Tarea de nivelación (10%)
- Proyectos (25% c/u)
- Dos exámenes parciales (20% c/u)

VIII. Bibliografía

Básica:

- J. de Vries, Learn OpenGL - Graphics Programming, editorial Kendall&Welling, 2020.
<https://learnopengl.com/> (Enlaces a un sitio externo.)
- Guía de referencia del lenguaje ANSI C,
https://www.mhe.es/universidad/informatica/8448145127/archivos/apendice_genera_l_2.pdf (Enlaces a un sitio externo.)
- T. Jensen, Tutorial sobre apuntadores y arreglos en C, 2000.
<https://www.cimat.mx/~alram/cpa/pointersC.pdf> (Enlaces a un sitio externo.)
- Universidad Malacitana, Estructuras dinámicas en C.
<http://www.lcc.uma.es/~illeivao/algoritmos/t4.pdf>

Complementaria:

- Hearn, Baker and Carithers, Computer Graphics in OpenGL, Pearson, 2014
- Hughes J. Van Dam A. Foley J. et al, Computer Graphics: Principles and Practice, Addison-Wesley, 2014

IX. Cronograma del desarrollo del curso

Es posible realizar el curso en dos semanas aunque sería recomendable con más holgura y tiempo de asimilar conocimientos. Por ello se sugiere realizarlo en tres semanas.

En caso de ser de dos semanas, la primera sería introducción a la computación gráfica y lenguaje C, gestores de ventanas, *shaders* y manejo de texturas. Al término de esa semana tendrían el primer examen parcial (de preferencia sobre una plataforma como canvas o Moodle) y se les asignaría el primer proyecto.

La segunda semana se cubrirían los temas de: transformaciones lineales, animaciones, visualización en 3D e iluminación. Además, se explicaría el segundo proyecto y se aplica el segundo examen parcial al final de la parte teórica.

La entrega de los proyectos será, como máximo, dos semanas después de finalizadas las clases y en ese momento se entregará la calificación definitiva a los alumnos y a servicios escolares.

