



FACULTAD DE INGENIERÍA - Course 2019/ 2019

SECRETARÍA/DIVISIÓN: DIVISIÓN DE INGENIERÍA ELÉCTRICA
ÁREA/DEPARTAMENTO: INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA E INTERACCIÓN HUMANO
COMPUTADORA:

Introducción a OpenGL

Reynaldo Martell Avila

PRÁCTICA 2

Contents

1	Objetivos de aprendizaje	2
1.1	Objetivos generales:	2
1.2	Objetivos específicos:	2
2	Recursos a emplear	2
2.1	Software	2
2.2	Equipos	2
2.3	Instrumentos	2
3	Fundamento Teórico	2
3.1	Desarrollo de actividades	2
4	Observaciones y Conclusiones	3
5	Anexos	3

1 Objetivos de aprendizaje

1.1 Objetivos generales:

El alumno aprenderá los conceptos básicos de OpenGL, el paradigma de programación y las funciones que se utilizan para renderizado de OpenGL.

1.2 Objetivos específicos:

El alumno aprenderá las funciones para el dibujo de primitivas geométricas en la pantalla, el paradigma de programación de OpenGL, a crear sus primeros Shaders de vértices y fragmentos, crear y utilizar el contexto de OpenGL, los conceptos de Vertex array object (VAO) y Vertex buffer object (VBO), así como la librería para crear ventanas y manejo de eventos.

2 Recursos a emplear

2.1 Software

Sistema Operativo: Windows 7 Ambiente de Desarrollo: Visual Studio 2017.

2.2 Equipos

Los equipos de cómputo con los que cuenta el laboratorio de Computación Gráfica

2.3 Instrumentos

3 Fundamento Teórico

- **Presentación de conceptos.**

Se le da a conocer al alumno los comandos **glGenVertexArrays**, **glBindVertexArray**, **glGenBuffers**, **glBindBuffer**, **glBufferData**, **glViewport**, nomenclatura de Vertex Shader, Fragment Shader, agregar, compilar y verificar y usar shaders, e identificarlos como bloques de información, se explica cuáles son los parámetros que pueden recibir estos comandos y eso como afecta a lo dibujado. Posteriormente se utilizará los atributos de vértices usando **glVertexAttribPointer**, **glEnableVertexAttribArray**.

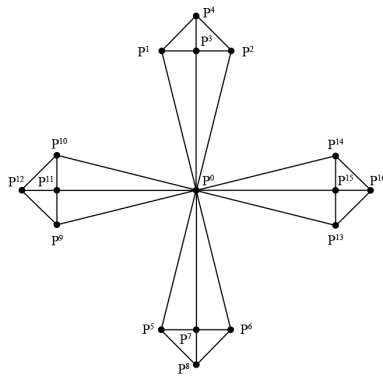
Finalmente se explican los comandos referentes a la creación de ventana

- **Datos necesarios.** Librería OpenGL 3.3, librería de creación de ventanas, IDE de desarrollo (Visual Studio 2017).

3.1 Desarrollo de actividades

1. Navegar hasta el directorio de trabajo de la práctica pasada (Donde se clono el repositorio), abrir un bash de git y teclear git pull origin master. Debe validar que la actualización esté correcta.
2. Se explica el código base para para el uso de la librería GLFW y GLEW.

3. Se explica el uso de los eventos de los dispositivos convencionales mouse y teclado (Sí es necesario).
4. Se muestra el ejemplo que crea un VAO, VBO, creación de un buffer y transferencia de memoria a la GPU, se muestra como reciben los bloques de memoria los Vertex shader y Fragment Shader.
5. Se muestra el uso los atributos de vértices, y los bloques de memoria.
6. Modificar el main.cpp para dibujar las siguientes figuras, se deben usar dos VAOs y VBOs diferentes. Finalmente, con la tecla E se muestra la estrella y con la tecla C la casa.



7. Se debe reportar y subir a github el ejercicio 6 anterior.

4 Observaciones y Conclusiones

5 Anexos

1. Cuestionario previo.
 - (a) ¿Qué es un polígono?
 - (b) ¿Qué es un polígono convexo y cóncavo?
 - (c) ¿Qué es un pixel?
 - (d) ¿Qué es OpenGL?
 - (e) ¿Qué es el contexto de OpenGL?
 - (f) ¿Cuáles son las etapas básicas del pipeline de renderizado de OpenGL?
 - (g) ¿Qué es un vertex shader?
 - (h) ¿Qué es un fragment shader?
 - (i) ¿Qué es un pixel?
 - (j) ¿Qué son las NDC (Normalized device coordinates)?

2. Actividad de investigación previa.

- (a) ¿Qué es la resolución de un dispositivo?, investigue y anote la resolución del monitor de su computadora principal.