



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:

**Proyecciones y puertos de vista. Transformaciones Geométricas**

Reynaldo Martell Avila

**PRÁCTICA 3**

## Contents

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

# 1 Objetivos de aprendizaje

## 1.1 Objetivos generales:

El alumno repasará como crear buffers de OpenGL, leer archivos, comprenderá los diferentes tipos de proyección y las funciones de la librería glm para crear éstas, así como comprenderá los diferentes sistemas de referencia de OpenGL. Del mismo modo practicará como colocar en la escena diferentes geometrías.

## 1.2 Objetivos específicos:

El alumno practicará crear geometrías con índices, revisará los sistemas de referencia que se aplican en OpenGL, comprenderá la utilización de la matriz de modelo, vista, proyección y la zona de dibujo.

# 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.**

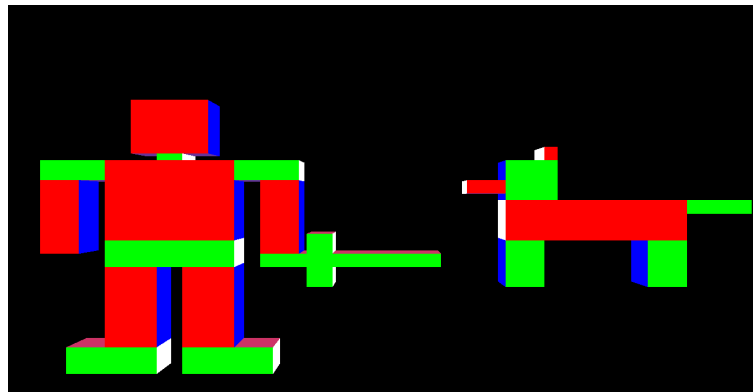
Se mostrará la utilización de índices, se revisará como crear los shaders de fragmento y vértices desde un archivo, se presentará el funcionamiento de variables globales que son tomadas de los shaders, utilizará los diferentes tipos de proyecciones y funciones para crearlas, así como colocar diferentes figuras en pantalla.

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

## 3.1 Desarrollo de actividades

1. Se explica el código para crear una clase que maneje el programa y los shaders.
2. Se explica el código base para crear un cubo utilizando indices.
3. Se muestra el sistema de referencia de dibujo `glViewoport()`
4. Se utiliza el concepto de variables uniform y su utilidad.
5. Se explica y se cambian los parámetros que definen los diferentes tipos de proyecciones.

6. Ejercicio: Los siguientes ejercicios se deben crear con un cubo con un color por cada lado. Se debe crear otro VAO y VBO
7. Utilizando un cubo unitario con centro en el origen como primitiva, y la transformación de translación y escalamiento, se creará una escena con las siglas CG 2019.
8. Se procede a crear un par de figuras instanciando el cubo y aplicando transformaciones básicas a cada una de las instancias.



9. **Ejercicio:** Crear la misma forma de la estrella de la práctica 2 con índices.
10. Deben subir sus ejercicios en Github y colocar la liga en su reporte.

## 4 Observaciones y Conclusiones

## 5 Anexos

1. Cuestionario previo.
  - (a) ¿Qué es una clase en c++?
  - (b) ¿Qué es un constructor y destructor de la clase, y cómo se declara en c++?
  - (c) ¿Cómo se instancia un objeto en c++?
  - (d) Investigar como abrir un archivo en c++.
  - (e) Investigue para qué sirve la función **glViewport** y que parámetros recibe.
  - (f) Investigue que es la matriz de Modelo, Vista, Proyección.
  - (g) ¿Qué es una proyección e investigue los tipos de proyecciones, en el ámbito de gráficos?
  - (h) Que utilidad tiene las funciones **glm::ortho**, **glm::frustum** y **glm::perspective**, y que son los parámetros que reciben.
  - (i) Para qué sirve la función **glfwSetWindowPos** y que parámetros recibe.

- (j) ¿Cuáles son las transformaciones geométricas básicas en tres dimensiones y sus matrices asociadas?
  - (k) Investigué para sirve la función **glm::scale**, **glm::translate**, **glm::rotate**, y que parámetros reciben.
2. Actividad de investigación previa.
- (a) Realizar un **git pull origin master** y un **git pull myRepo master**, antes de comenzar la práctica.