



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e
INTERACCIÓN HUMANO COMPUTADORA



EJERCICIOS DE CLASE N° 1

NOMBRE COMPLETO: García Soto Jean Carlo

N° de Cuenta: 319226304

GRUPO DE LABORATORIO: 03

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 11/02/2025

CALIFICACIÓN: _____

EJERCICIOS DE SESIÓN:

1. **Cambiar el color de fondo de la pantalla entre rojo, verde y azul de forma cíclica y solamente mostrando esos 3 colores con un periodo de lapso adecuado para el ojo humano**

El fondo que teníamos por predeterminado era negro, el color de nuestro triangulo era rojo, pero dado que se necesita un fondo rojo y se requiere que se vean nuestras figuras, el fondo de las figuras las cambiamos a blanco para después poder observar como cambian los fondos.

```
6
7    //Dimensiones de la ventana
8    const int WIDTH = 800, HEIGHT = 800;
9    GLuint VAO, VBO, shader;
10   float rojo = 1.0f, verde = 0.0f, azul = 0.0f;
11   double lastTime = glfwGetTime();
12   int colorIndice = 1; // Inicia en el color rojo
13
```

Para esto solo se necesitó declarar variables tipo flotante (**rojo, verde y azul**), en donde van los parámetros del color del fondo, y dado que cada 2 segundos se tiene que realizar el cambio de color, declaramos más variables que nos ayudarán para cumplir los requerimientos: **lastTime** para poder registrar el tiempo de inicio, y **colorIndice** que solo nos representará el color que se debe de mostrar dependiendo el valor que tenga (1 para rojo, 2 para verde y 3 para azul), se inicializa en 1 ya que como se ve en la declaración de la variable rojo está activada, por lo que ya la estaría mostrando y si no se hace simplemente el primer cambio de color sería de 4 seg en vez de 2.

```

177
178 //Loop mientras no se cierra la ventana
179 while (!glfwWindowShouldClose(mainWindow))
180 {
181     //Recibir eventos del usuario
182     glfwPollEvents();
183
184     //Obtener el tiempo actual
185     double tiempoActual = glfwGetTime();
186
187     if (tiempoActual - lastTime >= 2.0) {
188         lastTime = tiempoActual; //Reiniciar el tiempo
189
190         //Cambiar los colores rojo->verde->azul->rojo...
191         if (colorIndice == 0) {
192             rojo = 1.0f; verde = 0.0f; azul = 0.0f; // Rojo
193         }
194         else if (colorIndice == 1) {
195             rojo = 0.0f; verde = 1.0f; azul = 0.0f; // Verde
196         }
197         else if (colorIndice == 2) {
198             rojo = 0.0f; verde = 0.0f; azul = 1.0f; // Azul
199         }
200         colorIndice = (colorIndice + 1) % 3; //Cambiar al siguiente color
201     }
202
203     //Limpiar la ventana
204     glClearColor(rojo,verde,azul,1.0f);
205     glClear(GL_COLOR_BUFFER_BIT);
206

```

Se declara dentro del ciclo while otra variable para poder obtener el tiempo y así realizar una operación con la variable lastTime y comprobar que se hayan cumplido los 2 seg, si se cumple se reinicia el valor del tiempo de la variable lastTime y así volver a repetir el proceso anterior.

Ya para saber a que color se le debe de asignar ahora, se realiza una comparación de los posibles valores, el cual dependiendo a cuál entre se le asignan valores a las variables rojo, verde y azul. Ya por ultimo se realiza una operación para incrementar nuestra variable colorIndice y el cual cuando llegue a 3 este tomará el valor de 1.

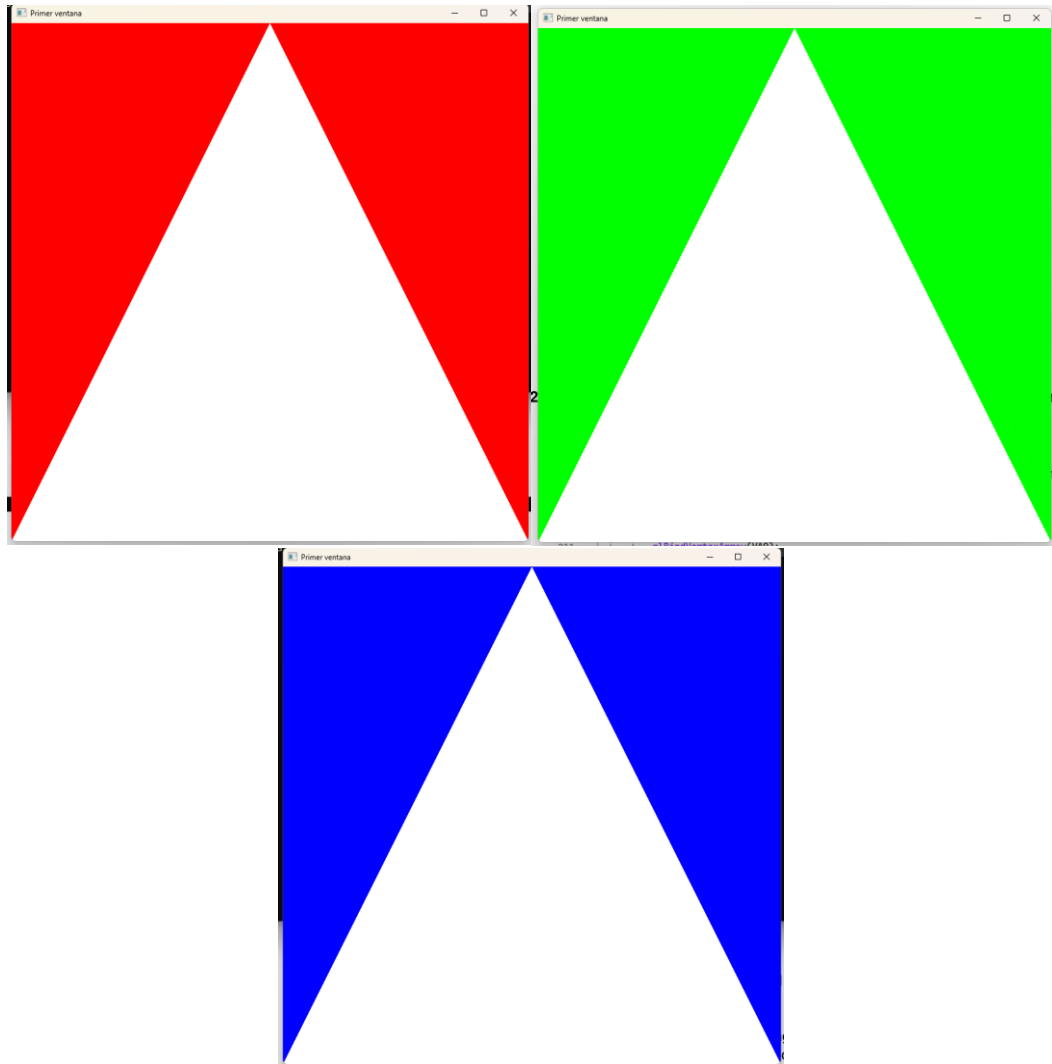
Y finalmente le pasamos las variables rojo, verde y azul para que cambien el color de fondo.

```

33
34 void CrearTriangulo()
35 {
36     GLfloat vertices[] = {
37         -1.0f, -1.0f, 0.0f, //Vertices del triangulo
38         0.0f, 1.0f, 0.0f,
39         1.0f, -1.0f, 0.0f
40
41
42     };

```

Vértices del triangulo



2. Dibujar de forma simultánea en la ventana 1 cuadrado y 1 rombo separados.

Para este ejercicio se nos enseñó primero a como generar un triángulo, los vértices que necesitaba, así como el rango de valores en los que podíamos trabajar. Como nuestras figuras están hechas a base de triángulos se concluyó que necesitábamos 12 vértices.

```

211 | | | glBindVertexArray(VA0);
212 | | | glDrawArrays(GL_TRIANGLES,0,12); // Cambiar de acuerdo a la cantida de vertices
213 | | | glBindVertexArray(0);
214 | | |

```

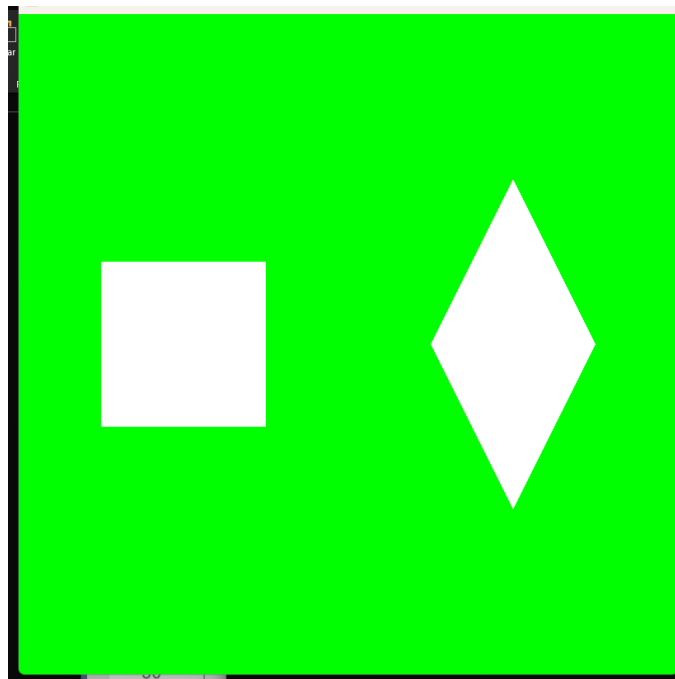
El cual en la línea 212 marcamos el número de vértices que queremos dibujar. Y después pasamos a poder o marcar nuestras coordenadas de cada uno de los vértices de nuestros 4 triángulos.

```

34  void CrearTriangulo()
35  {
36      GLfloat vertices[] = {
37          -0.75f, 0.25f, 0.0f,    //primer triangulo
38          -0.25f, 0.25f, 0.0f,
39          -0.75f, -0.25f, 0.0f,
40
41          -0.25f, 0.25f, 0.0f,    //segundo triangulo
42          -0.25f, -0.25f, 0.0f,
43          -0.75f, -0.25f, 0.0f,
44
45          0.25f, 0.0f, 0.0f,      //tercer triangulo
46          0.5f, 0.5f, 0.0f,
47          0.5f, -0.5f, 0.0f,
48
49          0.75f, 0.0f, 0.0f,      //cuarto triangulo
50          0.5f, 0.5f, 0.0f,
51          0.5f, -0.5f, 0.0f
52      };
53  };
54

```

Los primeros dos triángulos son para el cuadrado y los últimos dos para el rombo.



Problemas presentados. Listar si surgieron problemas a la hora de ejecutar el código

No hubo como tal un problema, sino una falta a lo mejor de conocimiento sobre las palabras reservadas o funciones que tiene el lenguaje, ya que gracias a que tengo conocimientos en el lenguaje C, no sabía que no estaba la palabra reservada `sleep()`, el cual iba a usarlo para poder estar retrasando o hacer que el programa se detuviera para que cambiara de color, pero dado que no existe en C++, tuve que buscar sobre las funciones que te ayudan a obtener el tiempo del sistema. El cual una vez supo que palabras reservadas puedo usar me dí una idea de como iba a realizar ese código.

De ahí en fuera me tuve que apoyar con un dibujo en un cuaderno, para poder visualizar un poco mejor o más fácil las coordenadas en donde iba a poner cada vértice de cada triángulo.

Conclusión:

Los ejercicios de la clase en mi opinión no estuvieron difíciles, el cual, si se puede complicar un poco por el motivo que no estoy familiarizado con el lenguaje C++, pero con un poco de investigación se pudo resolver los ejercicios solicitados.

Si bien considero que estuvo o se abordó bien la práctica, creo que podría ayudar demasiado unos ejemplos de las cosas que cambian del lenguaje C a C++, o de mostrar las funciones que se estarían usando. Pero en general considero que estos dos ejercicios fueron de acuerdo con el nivel esperado en el primer contacto.