UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e INTERACCIÓN HUMANO COMPUTADORA

REPORTE DE PRÁCTICA Nº 01

NOMBRE COMPLETO: Cornejo Aguilar Clara Luz

Nº de Cuenta: 316218423

GRUPO DE LABORATORIO: 01

GRUPO DE TEORÍA: 04

SEMESTRE 2024-2

FECHA DE ENTREGA LÍMITE: 17/02/2024

CALIFICACIÓN: _____

REPORTE DE PRÁCTICA:

1.- Ejecución de los ejercicios que se dejaron, comentar cada uno y capturas de pantalla de bloques de código generados y de ejecución del programa.

1.- Ventana cambia el color de fondo de forma random tomando rango de colores RGB y con una periodicidad de 2 segundos.

```
cambio += 0.009f;

if (cambio > 50.0f) {
    rojo = color_rojo();
    verde = color_verde();
    azul = color_azul();
    cambio = 0.0f;
}

//Recibir eventos del usuario
glfwPollEvents();

//Limpiar la ventana
glClearColor(rojo,verde,azul, 1.0f);
glClear(GL_COLOR_BUFFER_BIT);
```

En este código calculo los números random de cada color cada 2 segundos, lo que permite cambiar de color el fondo de pantalla aleatoriamente.

Las funciones que calculan los colores son:

Estas funciones devuelven un flotante entre 0 y 1 que será el número que se ele asignara a su valor RGB.

```
36
37  float color_verde() {
38      int numero_aleatorio = rand() % 101; // Genera un número entre 0 y 100
39      // Convertir el número a un valor entre 0 y 1
40      float color = (float)numero_aleatorio / 100.0;
41      return color;
42  }
43
44  float color_rojo() {
45      int numero_aleatorio = rand() % 101; // Genera un número entre 0 y 100
46      // Convertir el número a un valor entre 0 y 1
47      float color = (float)numero_aleatorio / 100.0;
48      return color;
49  }
50  float color_azul() {
51      int numero_aleatorio = rand() % 101; // Genera un número entre 0 y 100
52      // Convertir el número a un valor entre 0 y 1
53      float color = (float)numero_aleatorio / 100.0;
54      return color;
55  }
```

2.- 3 letras iniciales de sus nombres creadas a partir de triángulos, todas las letras son del mismo color.

Código que forma la letra C

```
GLfloat vertices[] = {
    //C
    -0.7f, 0.1f, 0.0f,//A
    -0.7f, 0.0f,0.0f,//E
    -0.8f, 0.0f, 0.0f, //F

    -0.7f, 0.1f, 0.0f,//A
    -0.9f, 0.1f, 0.0f,//B
    -0.8f, 0.0f, 0.0f, //F

    -0.9f, 0.1f, 0.0f,//B
    -0.8f, 0.0f, 0.0f, //F
    -0.8f, -0.1f,0.0f, //G

    -0.9f, 0.1f, 0.0f,//B
    -0.8f, -0.1f,0.0f, //G
    -0.9f, -0.2f, 0.0f, //C

    -0.8f, -0.1f,0.0f, //G
    -0.9f, -0.2f, 0.0f, //C
    -0.7f, -0.2f,0.0f,//D

    -0.8f, -0.1f,0.0f, //G
    -0.7f, -0.1f, 0.0f, //H
    -0.7f, -0.2f,0.0f,//D
```

Código que forma la letra L.

```
//L

-0.5f,0.1f,0.0f,//I
-0.4f, 0.1f,0.0f,//J
-0.4f, -0.1f,0.0f,//O

-0.5f,0.1f,0.0f,//I
-0.5f, -0.2f,0.0f,//K
-0.4f,-0.1f,0.0f,//O

-0.4f,-0.1f,0.0f,//O
-0.5f,-0.2f,0.0f,//K
-0.3f,-0.2f,0.0f,//N

-0.4f,-0.1f,0.0f,//O
-0.3f,-0.1f,0.0f,//L
-0.3f,-0.2f,0.0f,//N
```

Código que forma la letra A

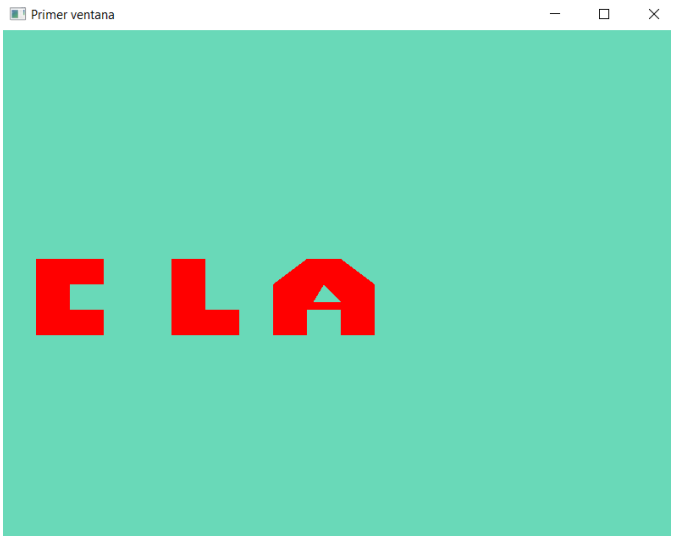
```
//A
-0.2f,-0.2f,0.0f,//M
-0.1f,-0.2f,0.0f,//U
-0.1f,-0.1f,0.0f,//V

-0.2f,-0.2f,0.0f,//M
-0.2f,0.0f,0.0f,//P
-0.1f,-0.1f,0.0f,//V

-0.2f,0.0f,0.0f,//P
-0.1f,-0.1f,0.0f,//V
-0.09f,-0.09,0.0f,//A

-0.1f,-0.1f,0.0f,//V
-0.09f,-0.07,0.0f,//A
0.0f,-0.1f,0.0f,//W

0.0f,-0.07f,0.0f,//b1
-0.09f,-0.07,0.0f,//A1
0.0f,-0.1f,0.0f,//W

-0.2f,0.0f,0.0f,//P
-0.05f,0.0f,0.0f,//C
-0.09f,-0.09,0.0f,//A

-0.2f,0.0f,0.0f,//P
-0.05f,0.0f,0.0f,//C
-0.1f,0.1f,0.0f,//Q

-0.05f,0.0f,0.0f,//C
-0.1f,0.1f,0.0f,//Q
0.0f,0.1f,0.0f,//R

-0.05f,0.0f,0.0f,//C
0.1f,0.0f,0.0f,//s
0.0f,0.1f,0.0f,//R

-0.05f,0.0f,0.0f,//C
0.1f,0.0f,0.0f,//s
0.0f,-0.07f,0.0f,//B

0.0f,-0.07f,0.0f,//B
0.0f,-0.1f,0.0f,//W
0.1f,0.0f,0.0f,//s

0.0f,-0.1f,0.0f,//W
0.1f,0.0f,0.0f,//s
0.1f,-0.2f,0.0f,//T

0.0f,-0.1f,0.0f,//W
0.0f,-0.2f,0.0f,//z
0.1f,-0.2f,0.0f,//T
```

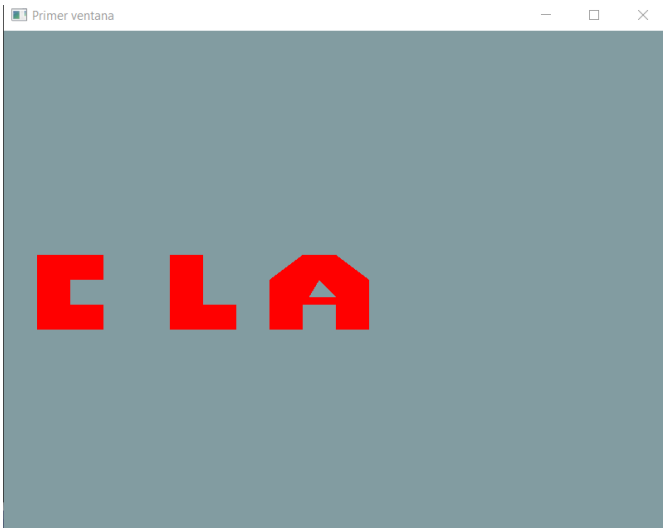
Una vez declarados los vértices de cada letra a la función que dibuja los triángulos se le agrega el número de vértices.

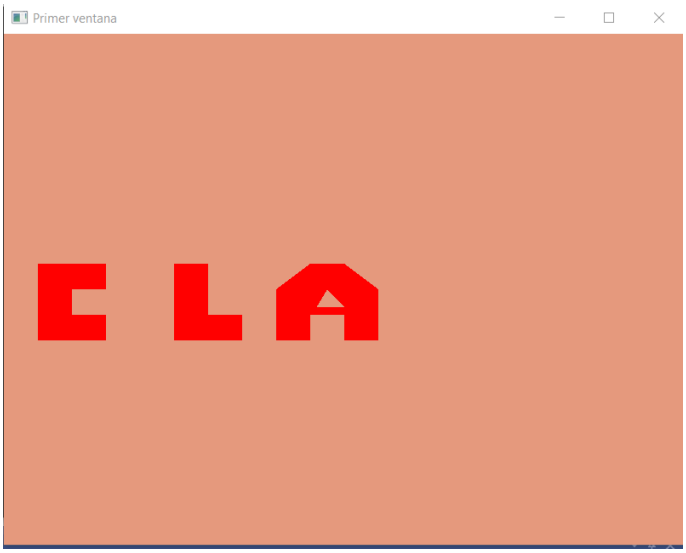
```
//se agregan nuevos vertices, se toman de 3 en tres vertices
glBindVertexArray(VAO);
glDrawArrays(GL_TRIANGLES, 0, 69);//LINES, POINT, TRIANGLES
glBindVertexArray(0);
```

Los dos ejercicios se muestran de forma simultanea y están en el mismo main.

Resultados







2.- Liste los problemas que tuvo a la hora de hacer estos ejercicios y si los resolvió explicar cómo fue, en caso de error adjuntar captura de pantalla

No presenté ningún problema al realizar la práctica sin embargo, al escribir los vértices de la letra A tuve confusiones pero lo resolví.

3.- Conclusión:

La complejidad del ejercicio fue buena ya que va acorde con lo visto en clase.

Lo único laborioso fueron los vértices de cada letra.

Open GL nos permite dibujar figuras a partir de triángulos, lo cuál nos permitirá realizar cosas interesantes en posteriores prácticas.

Bibliografía en formato APA

Tutorial 2 : El primer triángulo. (s. f.). https://www.opengl-tutorial.org/es/beginners-tutorials/tutorial-2-the-first-triangle/