Universidad Nacional

Autónoma de México

Practica No. 1

Alumno: Arroyo Llanes Miguel Alejandro.

No. Cuenta: 319218460

Materia: Lab. Computación Grafica e Interacción Humano-Computadora

Grupo Laboratorio: 3

Grupo teoría: 6

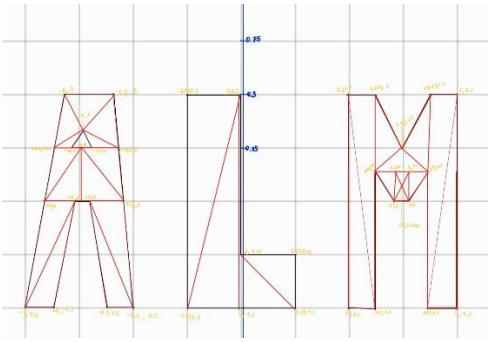
Fecha de Entrega: 15/02/2025

Nombre profesor: Ing. Jose Roque Roman Guadarrama

Comentario:

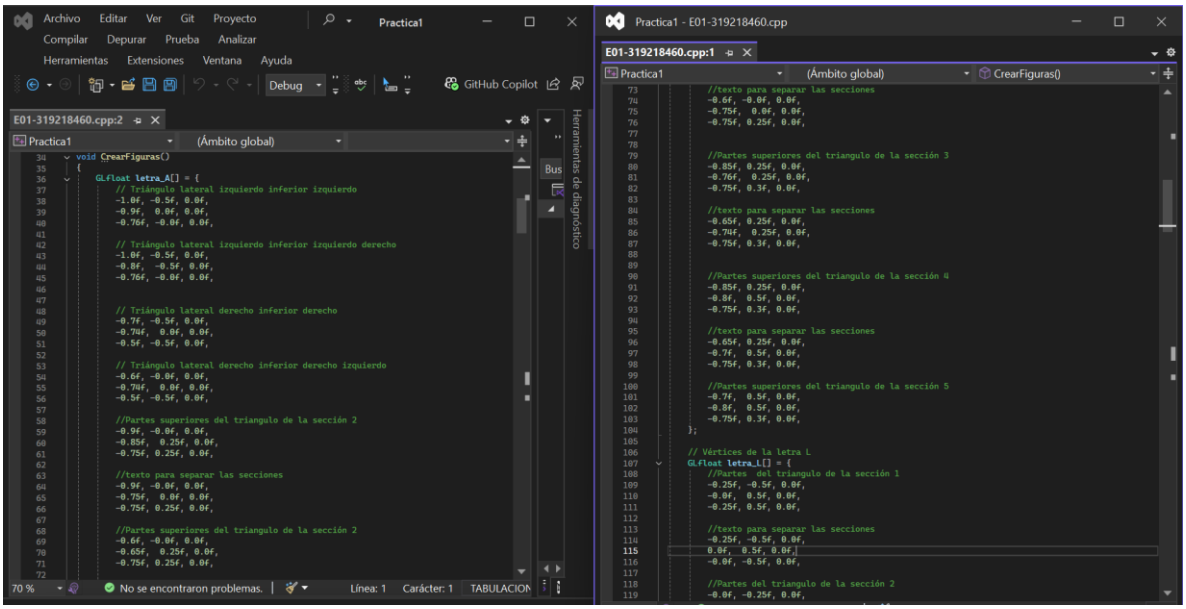
Primeramente, se buscó hacer el cambio de fondo en la ejecución por el RGB de forma aleatoria, se implementaron las bibliotecas stdlib.h y time.h para cambiar de forma aleatoria, también se implemento el cambio de color de forma aleatoria cada 2 segundos.

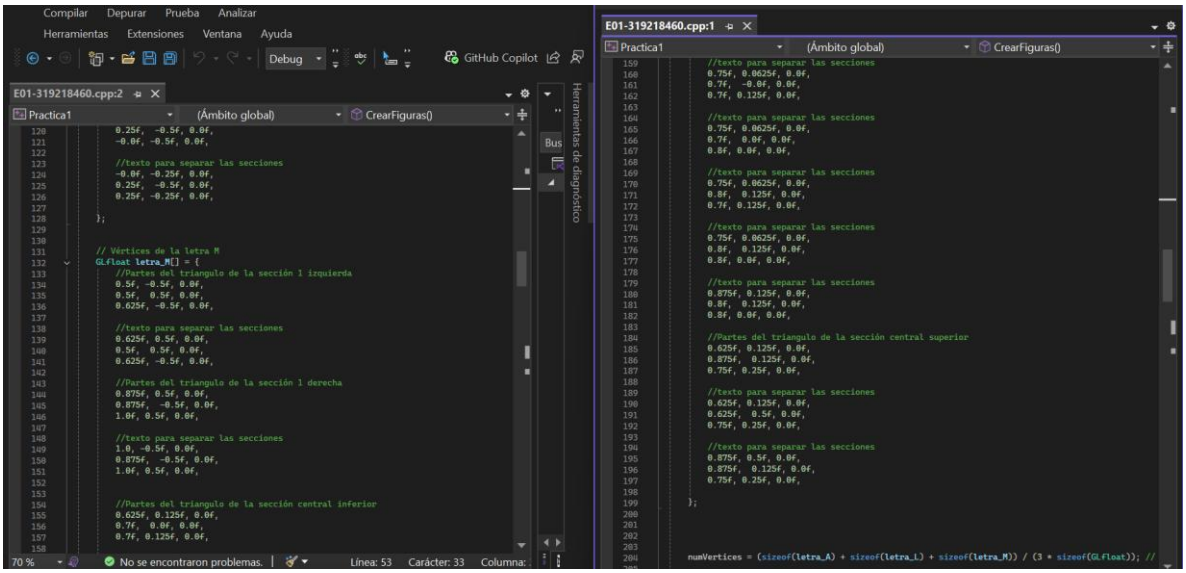
Posteriormente se dibujaron las letras con triángulos, primeramente, se dibujaron de manera digital utilizando una Tablet para poder visualizar cuantos triángulos se necesitarían por letra y en que coordenadas se debían colocar los puntos para formar el triángulo. Una vez hechas las letras a mano y se dibujaran las coordenadas necesarias se escribió en el código las líneas suficientes para dibujar cada triángulo y formar así las letras.



Bloque de código:

Líneas de código que muestran el dibujo de los vértices (triángulos) de las letras “A”, “L” y “M”.






```
numVertices = (sizeof(letra_A) + sizeof(letra_L) + sizeof(letra_M)) / (3 * sizeof(GLfloat)); // Calcular la cantidad correcta de vértices



glGenVertexArrays(1, VAO); // Generar 1 VAO
glGenBuffers(1, VBO); // Generar 1 VBO

glBindVertexArray(VAO[0]); // Asignar VAO
glBindBuffer(GL_ARRAY_BUFFER, VBO[0]);
glBufferData(GL_ARRAY_BUFFER, sizeof(letra_A) + sizeof(letra_L) + sizeof(letra_M), NULL, GL_STATIC_DRAW);
glBufferSubData(GL_ARRAY_BUFFER, 0, sizeof(letra_A), letra_A);
glBufferSubData(GL_ARRAY_BUFFER, sizeof(letra_A), sizeof(letra_L), letra_L);
glBufferSubData(GL_ARRAY_BUFFER, sizeof(letra_A) + sizeof(letra_L), sizeof(letra_M), letra_M);


glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 3 * sizeof(GL_FLOAT), (GLvoid*)0);
glEnableVertexAttribArray(0);

glBindBuffer(GL_ARRAY_BUFFER, 0);
glBindVertexArray(0);
```

Cambio de color de fondo de la pantalla de la ejecución en colores R (rojo), G (verde) y B (azul), los cuales cambiaran aleatoriamente cada 2 segundos.

```
        //Limpiar la ventana
        //glClearColor(0.0f,0.0f,0.0f,1.0f);
        glClear(GL_COLOR_BUFFER_BIT);

        double currentTime = glfwGetTime();
        if (currentTime - lastTime >= 2.0) { // Cambiar color cada 2 segundos
            int colorIndex = rand() % 3; // Elegir un número aleatorio entre 0 y 2

            if (colorIndex == 0) {
                glClearColor(1.0f, 0.0f, 0.0f, 1.0f); // Rojo
            }
            else if (colorIndex == 1) {
                glClearColor(0.0f, 1.0f, 0.0f, 1.0f); // Verde
            }
            else {
                glClearColor(0.0f, 0.0f, 1.0f, 1.0f); // Azul
            }

            lastTime = currentTime;
        }
```

Ejecución:

Conclusión:

Con lo aprendido en la clase y en el previo he aprendido el como hacer dibujos con vértices al menos de momento en un espacio de 2 dimensiones, así como aprendí a optimizarlo de la mejor manera cada letra para evitar desperdiciar recursos, aunque aun se puede mejorar y aun es una introducción al tema, se ha logrado absorber el conocimiento suficiente y reafirmar el que se tenia anteriormente con cursos pasados.

Bibliografía:

1. En esta practica no se hizo uso de fuentes alternas a las que ya se habían aprendido en el laboratorio y previo.