



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 N° 01

NOMBRE COMPLETO: Rosas Martinez Francisco Javier

N° de Cuenta: 320109766

GRUPO DE LABORATORIO: 03

GRUPO DE TEORÍA: 05

SEMESTRE 2026-1

FECHA DE ENTREGA LÍMITE: 24/Agosto/2025

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.

Ejercicio 1:

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

Para lograr el cambio de colores de manera aleatoria use las librerías *time.h* y *stdlib.h*, debido a que estas librerías usan el ciclo propio de la computadora tuve que utilizar la función *srand(time(NULL))* afuera del ciclo principal para determinar los números aleatorios evitando que se repitiera algún patrón, además utilice un vector con dimensión 3 para almacenar cada uno de los números aleatorios, por último con la librería *Windows.h* pause el programa 2 segundos en cada iteración.

Bloques de código generado

```
5      //Para reposar el programa //  
6      #include <Windows.h>  
7  
8      //Para el uso de números randoms //  
9      #include <stdlib.h> // Para rand() y srand()  
10     #include <time.h>   // Para time()  
11
```

```
301     // Inicializar la semilla con el tiempo actual  
302     srand(time(NULL));  
303  
304  
305     //Loop mientras no se cierra la ventana  
306     while (!glfwWindowShouldClose(mainWindow))  
307     {  
308  
309         for (int i = 0; i < 3; ++i) {  
310             colores[i] = ((rand() % 10) + 1); // rand() % 10 genera 0-9, +1 → 1-10  
311             colores[i] = colores[i] / 10; // Valores de 0.0 a 1.0 //  
312         }  
313  
314  
315         //Recibir eventos del usuario  
316         glfwPollEvents();  
317  
318         //Limpiar la ventana  
319         // Llamamos a la función con los índices deseados //  
320         glClearColor(colores[0], colores[1], colores[2], 1.0f);  
321
```

```
334  
335 // Función para reposar el programa 2000 milisegundos en cada iteración //  
336 Sleep(2000);
```

Código en ejecución



Ejercicio 2:

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

Con este ejercicio tuve que dividir cada una de mis iniciales en pequeños triángulos para lograr que en conjunto formarán la forma de la letra, además utilice una constante simbólica donde colocaba el numero de triangulos que iba a dibujar para que a la hora de pasarlo a la función *IDrawArrays()* el atributo se multiplica por 9 que es el número de puntos que ocupa cada triángulo.

Bloques de código generado

```
38 //Constante simbólica para contar el número de triángulos //  
39 #define NUMERO 24
```

```

45 // Coordenadas de cada triangulo //
46 //Triángulos para la R //
47 -0.65f,-0.3f,0.0f,
48 -0.65f,0.3f,0.0f,
49 -0.55f,0.3f,0.0f,
50
51
52 -0.65f,-0.3f,0.0f,
53 -0.55f,0.3f,0.0f,
54 -0.55f,-0.3f,0.0f,
55
56
57 -0.55f,-0.1f,0.0f,
58 -0.35f,-0.3f,0.0f,
59 -0.35f,-0.2f,0.0f,
60
61
62 -0.35f,-0.2f,0.0f,
63 -0.55f,-0.1f,0.0f,
64 -0.55f,-0.0f,0.0f,
65
66
67 -0.55f,-0.0f,0.0f,
68 -0.35f,-0.0f,0.0f,
69 -0.55f,0.1f,0.0f,
70
71
72 -0.35f,-0.0f,0.0f,
73 -0.55f,0.1f,0.0f,
74 -0.35f,0.1f,0.0f,
75
76
77 -0.55f,0.2f,0.0f,
78 -0.35f,0.2f,0.0f,
79 -0.55f,0.3f,0.0f,
80
81
82 -0.35f,0.2f,0.0f,
83 -0.55f,0.3f,0.0f,
84 -0.35f,0.3f,0.0f,
85

```

```

86
87 -0.35f,0.1f,0.0f,
88 -0.45f,0.1f,0.0f,
89 -0.45f,0.2f,0.0f,
90
91
92 -0.35f,0.1f,0.0f,
93 -0.35f,0.2f,0.0f,
94 -0.45f,0.2f,0.0f,
95
96 //Triángulos para la M //
97
98 -0.2f,-0.3f,0.0f,
99 -0.1f,-0.3f,0.0f,
100 -0.2f,0.3f,0.0f,
101
102
103 -0.1f,-0.3f,0.0f,
104 -0.2f,0.3f,0.0f,
105 -0.1f,0.3f,0.0f,
106
107
108 -0.1f,0.3f,0.0f,
109 0.0f,0.2f,0.0f,
110 -0.1f,0.15f,0.0f,
111
112
113 0.0f,0.2f,0.0f,
114 -0.1f,0.15f,0.0f,
115 0.0f,0.05f,0.0f,
116
117
118 0.2f,-0.3f,0.0f,
119 0.1f,-0.3f,0.0f,
120 0.2f,0.3f,0.0f,
121
122
123 0.1f,-0.3f,0.0f,
124 0.2f,0.3f,0.0f,
125 0.1f,0.3f,0.0f,
126

```

```

127
128 0.1f,0.3f,0.0f,
129 0.0f,0.2f,0.0f,
130 0.1f,0.15f,0.0f,
131
132
133 0.0f,0.2f,0.0f,
134 0.1f,0.15f,0.0f,
135 0.0f,0.05f,0.0f,
136
137
138 //Triángulos para la F //
139
140 0.35f,-0.3f,0.0f,
141 0.35f,0.3f,0.0f,
142 0.45f,0.3f,0.0f,
143
144
145 0.35f,-0.3f,0.0f,
146 0.45f,0.3f,0.0f,
147 0.45f,-0.3f,0.0f,
148
149
150 0.45f,-0.0f,0.0f,
151 0.65f,-0.0f,0.0f,
152 0.45f,0.1f,0.0f,
153
154
155 0.65f,-0.0f,0.0f,
156 0.45f,0.1f,0.0f,
157 0.65f,0.1f,0.0f,
158
159
160 0.45f,0.2f,0.0f,
161 0.65f,0.2f,0.0f,
162 0.45f,0.3f,0.0f,
163
164
165 0.65f,0.2f,0.0f,
166 0.45f,0.3f,0.0f,
167 0.65f,0.3f,0.0f,
168
169 }

```

```
328 // Cambiamos el número de valores aceptados //  
329 // Número de puntos por triángulo //  
330 glDrawArrays(GL_TRIANGLES,0,NUMERO*9);  
331 glBindVertexArray(0);
```

Código en ejecución



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

A la hora de ir resolviendo el primer ejercicio tuve ligera complicación a la hora de mapear cada letra en la ventana para que tuviera la forma más legible de la letra, además había veces que me confundía a la hora de escribir todos los puntos de cada triángulo, con el segundo ejercicio tuve complicación para determinar cómo iba a usar los números aleatorios para que no siguieran un patrón en cada iteración.

Conclusión:

Tras el término de la práctica logré conocer funciones básicas de OpenGL, con esto pude generar diversas figuras combinando distintos triángulos, me pareció una práctica muy entretenida que además me ayudó a recordar la sintaxis de los lenguajes de programación C y C++. Por último, gracias a la explicación del profesor en el laboratorio, pude entender cómo abordar cada ejercicio y así lograr resolverlos correctamente.

Bibliografia :

Shreiner, D., Sellers, G., Kessenich, J., & Licea-Kane, B. (2013). OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3 (8th ed.). Addison-Wesley Professional.