



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: Vázquez Reyes Sebastián

N° de Cuenta: 318150923

GRUPO DE LABORATORIO: 11

GRUPO DE TEORÍA: 6

SEMESTRE 2024-2

FECHA DE ENTREGA LÍMITE: sábado 17 de febrero de 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.

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

Para esta sección del código me vi en la necesidad de consultar un video en particular que explica como calcular vectores con valores al azar en C++, el video se encuentra en la bibliografía del reporte. De esta fuente, únicamente utilice la siguiente línea de código:

```
//Limpiar la ventana  
azul = (float)(1 + rand() % 100) / 100;  
rojo = (float)(1 + rand() % 100) / 100;  
verde = (float)(1 + rand() % 100) / 100;  
glClearColor (rojo, verde, azul, 1.0f);
```

En las variables azul, rojo y verde, guardamos el valor de la siguiente operación. Calculamos un valor al azar entre 1 y 100 con la funcion rand, y dividimos este valor entre 100. Como el resultado es un valor decimal, la palabra float antes de la operación nos indica que guardaremos un valor de tipo float en estas variables. Luego, son usadas en la línea de código para calcular el color de la ventana.

Para observar el cambio en 2 segundos, utilice la funcion Sleep(), que recibe en milisegundos la cantidad de tiempo que el programa esperará:

```
Sleep(2000);  
glClear(GL_C
```

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

Para esta parte me vi en la necesidad de crear un total de 78 vértices, divididos por letra, y varios repetidos, para mostrar las letras de mi nombre en la pantalla.

```

void CrearTriangulo()
{
    GLfloat vertices[] = {
        //S
        -0.8f, -0.5f, 0.0f,
        -0.8f, -0.3f, 0.0f,
        -0.5f, -0.5f, 0.0f,
        -0.8f, -0.3f, 0.0f,
        -0.5f, -0.5f, 0.0f,
        -0.5f, -0.3f, 0.0f,

        -0.5f, -0.5f, 0.0f,
        -0.5f, +0.1f, 0.0f,
        -0.65f, -0.5f, 0.0f,
        -0.5f, +0.1f, 0.0f,
        -0.65f, -0.5f, 0.0f,
        -0.65f, +0.1f, 0.0f,

        -0.8f, -0.1f, 0.0f,
        -0.8f, +0.1f, 0.0f,
        -0.5f, -0.1f, 0.0f,
        -0.8f, +0.1f, 0.0f,
        -0.5f, -0.1f, 0.0f,
        -0.5f, +0.1f, 0.0f,

        -0.8f, -0.1f, 0.0f,
        -0.8f, +0.3f, 0.0f,
        -0.65f, -0.1f, 0.0f,
        -0.8f, +0.3f, 0.0f,
        -0.65f, -0.1f, 0.0f,
        -0.65f, +0.3f, 0.0f,

        -0.8f, +0.5f, 0.0f,
        -0.8f, +0.3f, 0.0f,
        -0.5f, +0.5f, 0.0f,
        -0.8f, +0.3f, 0.0f,
        -0.5f, +0.5f, 0.0f,
        -0.5f, +0.3f, 0.0f,
    }
}

```

```

//N
-0.2f, -0.5f, 0.0f,
-0.2f, -0.3f, 0.0f,
+0.1f, -0.5f, 0.0f,
-0.2f, -0.3f, 0.0f,
+0.1f, -0.5f, 0.0f,
+0.1f, -0.3f, 0.0f,

-0.35f, +0.5f, 0.0f,
-0.2f, -0.5f, 0.0f,
-0.05f, -0.3f, 0.0f,
-0.35f, +0.5f, 0.0f,
-0.05f, -0.3f, 0.0f,
-0.25f, +0.5f, 0.0f,

+0.25f, +0.5f, 0.0f,
+0.1f, -0.5f, 0.0f,
-0.05f, -0.3f, 0.0f,
+0.25f, +0.5f, 0.0f,
-0.05f, -0.3f, 0.0f,
+0.15f, +0.5f, 0.0f,

```

```

+0.15f, +0.5f, 0.0f,
//R
+0.8f, +0.3f, 0.0f,
+0.8f, +0.5f, 0.0f,
+0.5f, +0.3f, 0.0f,
+0.8f, +0.5f, 0.0f,
+0.5f, +0.3f, 0.0f,
+0.5f, +0.5f, 0.0f,

+0.4f, +0.5f, 0.0f,
+0.4f, -0.5f, 0.0f,
+0.55f, -0.5f, 0.0f,
+0.4f, +0.5f, 0.0f,
+0.55f, -0.5f, 0.0f,
+0.55f, +0.5f, 0.0f,

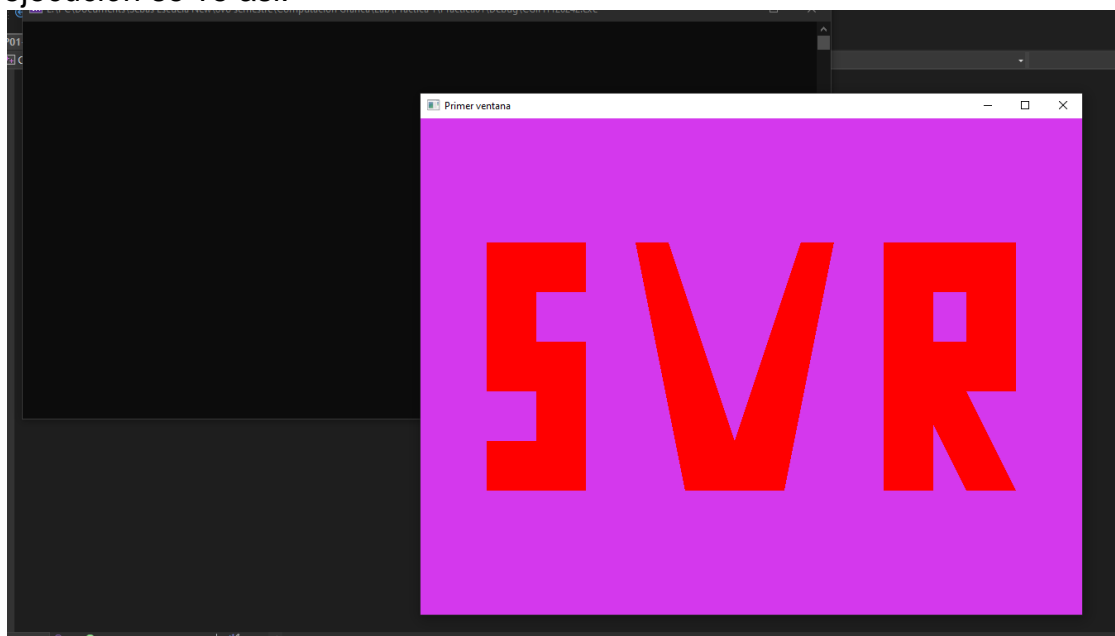
+0.8f, +0.1f, 0.0f,
+0.8f, -0.1f, 0.0f,
+0.5f, +0.1f, 0.0f,
+0.8f, -0.1f, 0.0f,
+0.5f, +0.1f, 0.0f,
+0.5f, -0.1f, 0.0f,

+0.8f, +0.5f, 0.0f,
+0.8f, -0.1f, 0.0f,
+0.65f, -0.1f, 0.0f,
+0.8f, +0.5f, 0.0f,
+0.65f, -0.1f, 0.0f,
+0.65f, +0.5f, 0.0f,

+0.8f, -0.5f, 0.0f,
+0.65f, -0.5f, 0.0f,
+0.5f, -0.1f, 0.0f,
+0.8f, -0.5f, 0.0f,
+0.65f, -0.1f, 0.0f,
+0.5f, -0.1f, 0.0f

```

Y la ejecución se ve así:





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

Para el primer problema, tarde un poco en encontrar una forma de calcular los números aleatorios. Investigué acerca de las librerías en C++ que ofrecieran funciones así, aunque

algunas bastantes complicadas, pero no fue hasta que encontré el video que me decidí por implementar esas líneas de código, por su sencillez. Además, como la función Sleep() detiene el programa durante 2 segundos, tan pronto se cambia el color de la ventana el programa se detiene, lo que causa que ni siquiera te deje mover la ventana o cerrarla hasta que pasen los 2 segundos y el programa se active de nuevo, tal vez haya una forma mas efectiva de lograr este efecto sin detener el programa así.

Para el segundo problema, fue únicamente prueba y error hasta conseguir la forma de las letras, realmente no fue un problema si no mas bien tedioso, aunque no me haya tomado mucho tiempo.

3.- Conclusión:

a. Los ejercicios del reporte: Complejidad, Explicación.

En un inicio los ejercicios se veían complicados, pero fue mas prueba y error que otra cosa. Afortunadamente entendí todo lo necesario en el laboratorio para poder hacer el reporte.

b. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias para mejorar desarrollo de la práctica

Explicar todo con detalle es muy bueno, ayuda a entender como hacer las cosas con detalle sin quebrarse tanto la cabeza.

c. Conclusión

Los ejercicios me ayudaron mucho a entender la forma en que funciona la construcción del color en los entornos gráficos con los que he interactuado. Además, también aprendí a crear triángulos y como crear diversas figuras a partir de triángulos.

Bibliografía

1. InformativaC., "Números decimales aleatorios en un vector C++". Recuperado el 15 de febrero de 2024 de <https://www.youtube.com/watch?v=p8my4WOjxB4>