Reporte de Práctica de Laboratorio de Computación Gráfica e Interacción Humano-Computadora

Información de la práctica

Nombre del alumno:	Martínez Pérez Brian Erik		
Número de Cuenta:	319049792	**Práctica número:**	1
Título de la práctica:	Introducción a OpenGL		
Fecha de entrega:	24 de agosto de 2024		

Introducción: (descripción inicial sobre la práctica)

Esta práctica tuvo como objetivo familiarizarse con el entorno de desarrollo visual studio 2022 y los conceptos básicos de OpenGL, una API ampliamente utilizada para el renderizado de gráficos 2D y 3D. Se busca comprender la estructura básica de un programa en OpenGL, desde la inicialización de la ventana y uso de las primitivas de open GL hasta la renderización de primitivas geométricas simples.

Resumen Teórico: (introducción sobre los fundamentos teóricos en los que se basa la práctica y cuáles son los objetivos teóricos que persigue.)

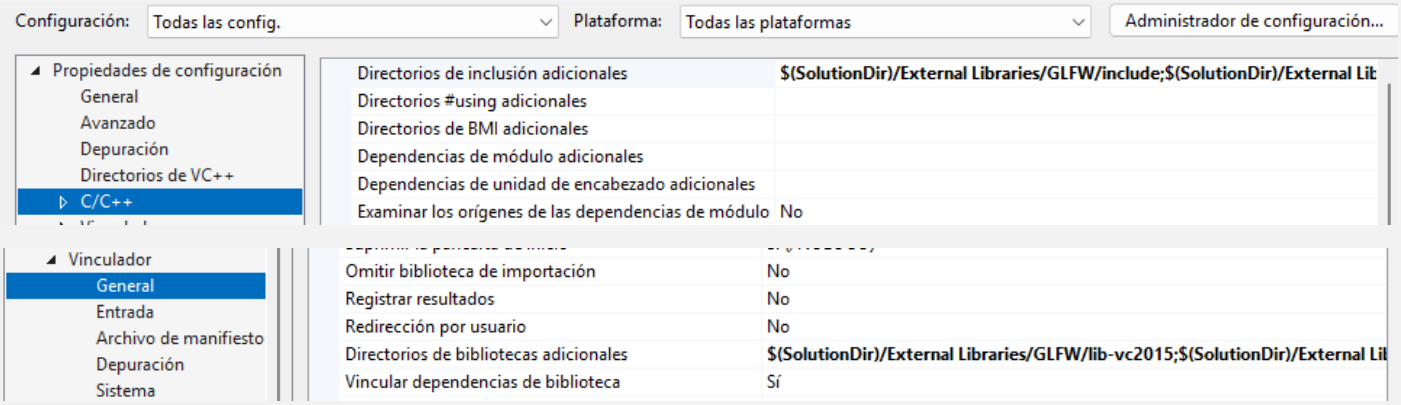
OpenGL es una interfaz de programación de aplicaciones (API) que permite la creación de gráficos por computadora de alto rendimiento. Entre los fundamentos teóricos clave se encuentran el pipeline de renderizado, que transforma vértices en píxeles en la pantalla, el uso de shaders para controlar el proceso de renderizado

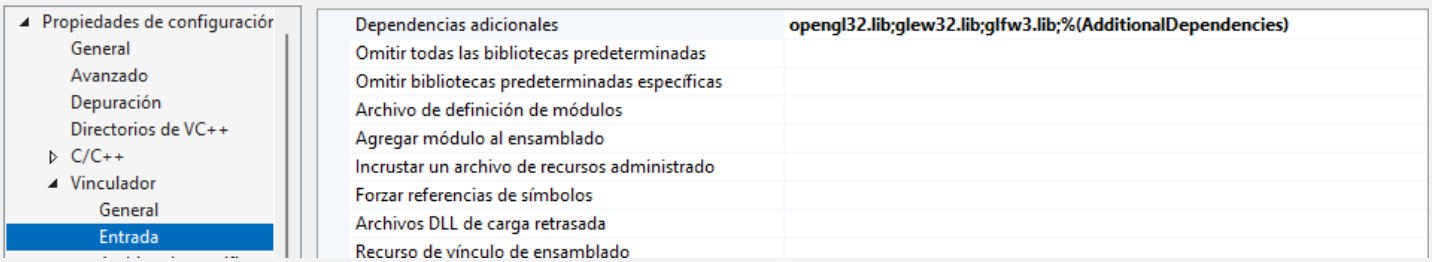
Configuración inicial de un proyecto en VS 2022 para usar las bibliotecas GLEW, GLFW y la API Open GL.

La importancia de las primitivas gráficas (puntos, líneas, triángulos) en la construcción de objetos visuales.

Descripción de la práctica: (En esta sección se deben describir todos los pasos ejecutados durante la sesión práctica realizada en el laboratorio.)

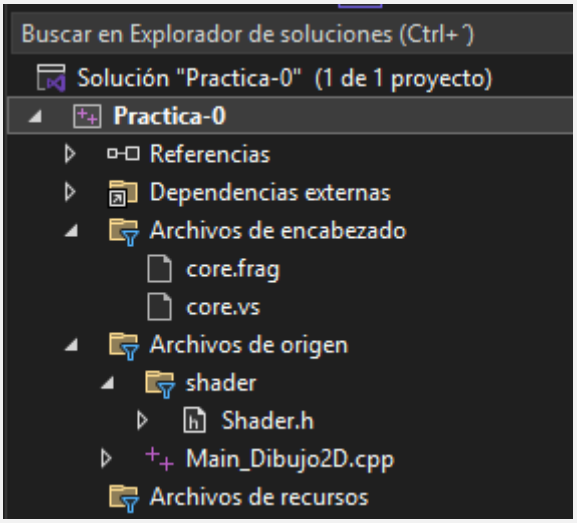
Se configuró el entorno de desarrollo usando Visual Studio Community y las bibliotecas GLEW y GLFW.





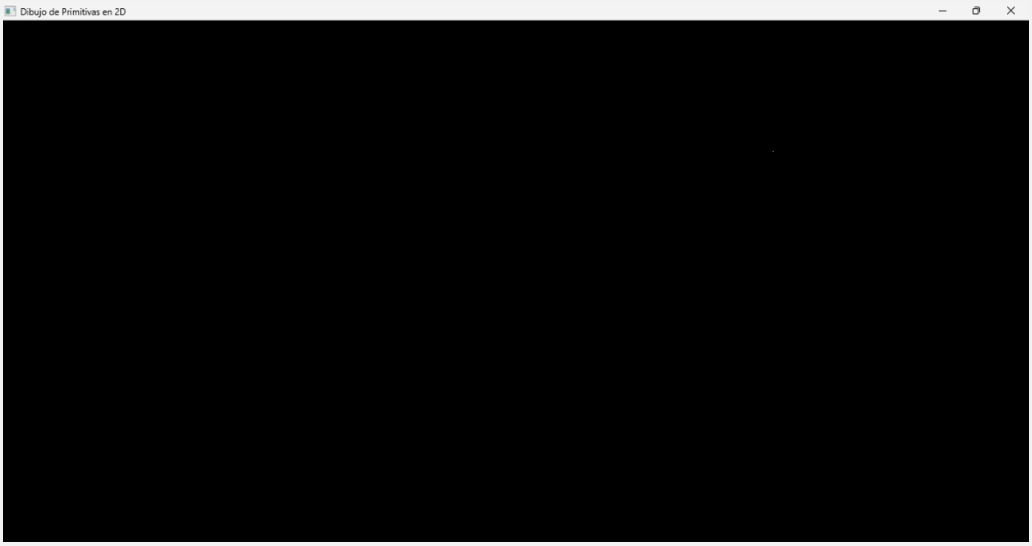
Agregamos los core.frag (definicion del color de los píxeles) y core.vs (transformación de vertices) en la sección de “Archivos de encabezado”.

Agregamos sheader.h (carga, compilación, enlace) y Main_Dibujo2D.cpp (código con la implementación de las primitivas) en la sección de “Archivos de Origen”.



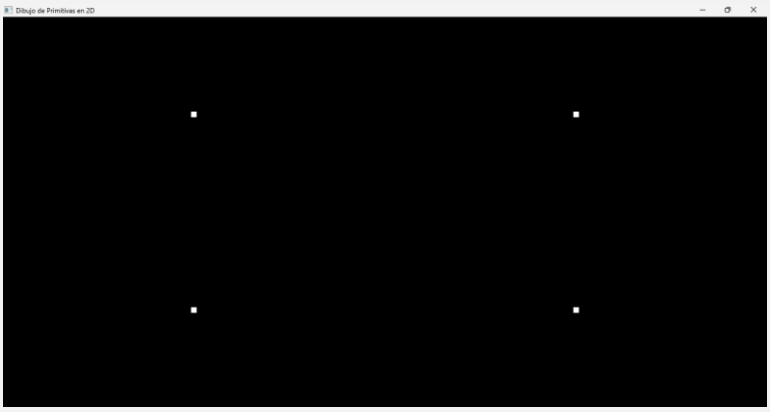
Primera ejecución del programa Main_Dibujo2D.cpp donde generamos un pequeño punto en la sección de superior derecha.

```
glPointSize(1); //tamaño del punto
glDrawArrays(GL_POINTS,0,1);
```



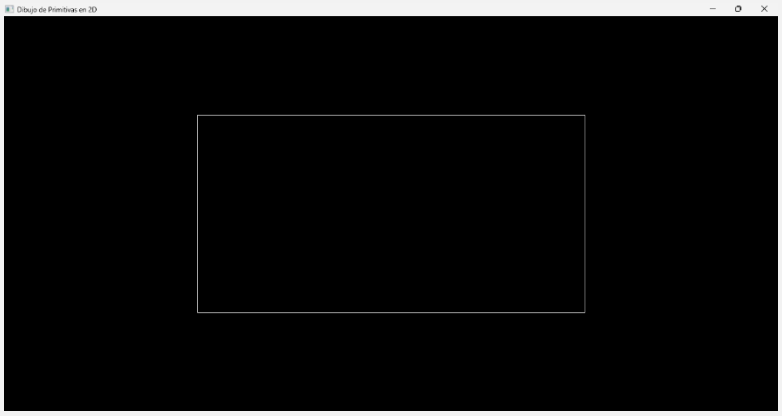
En la segunda ejecución modificamos las dos líneas del código para generar cuatro puntos y aumentar su tamaño

```
glPointSize(10); //tamaño del punto
glDrawArrays(GL_POINTS,0,4);
```



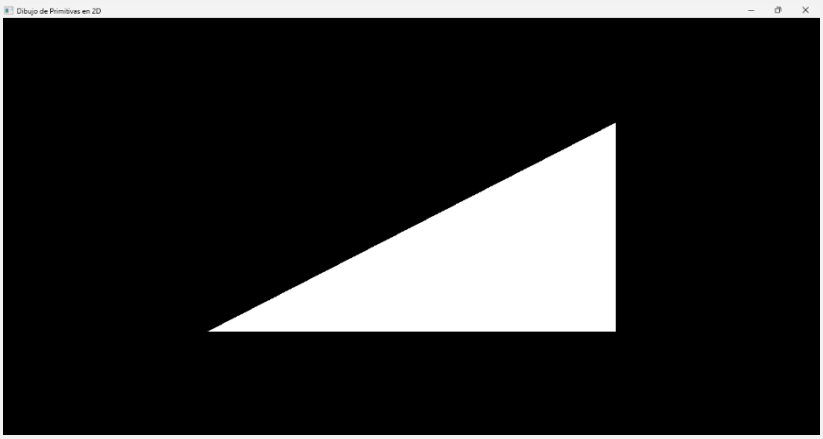
En la tercera ejecución generamos un rectángulo a partir de las primitivas de líneas

```
glDrawArrays(GL_LINES,0,2);
glDrawArrays(GL_LINE_LOOP,0,4);
```



En la cuarta ejecución utilizamos la última primitiva generamos un triángulo de color de relleno blanco.

```
glDrawArrays(GL_TRIANGLES,0,3);
glDrawElements(GL_TRIANGLES, 1,GL_UNSIGNED_INT,0);
```



Resultados: (explicar lo que se logró o aquello que no lograron realizar o comprender)

En esta practica logramos hacer la configuración de VS 2022, además de conocer la primitivas para el diseño de modelos. No solo conocimos las primitivas, sino que también las aplicamos cada una en archivo cpp para poder visualizarlas. Además, modificamos la información de los vértices ya sea su ubicación en la pantalla o su color, en mi caso no pude cambiar el color de los vértices.

Conclusiones:

La práctica permitió entender la estructura fundamental de un programa en OpenGL y la importancia de los shaders en el pipeline de renderizado. Se identificó la importancia de integrar correctamente los shaders en el proceso de renderizado para controlar aspectos visuales como el color y las posiciones de las primitivas generadas durante la clase de laboratorio.

Bibliografía consultada:

OpenGL - the industry's foundation for high performance graphics. (2011, julio 19). The Khronos Group. https://www.khronos.org/opengl/

Learn OpenGL, extensive tutorial resource for learning Modern OpenGL. (s/f). Learnopengl.com. Recuperado el 23 de agosto de 2025, de https://learnopengl.com/