



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e
INTERACCIÓN HUMANO COMPUTADORA



PREVIO N° 03

NOMBRE COMPLETO: Aguilar Pérez José Ramón

N° de Cuenta: 317515048

GRUPO DE LABORATORIO: 02

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 24/febrero/2025

CALIFICACIÓN: _____

1. Contenido requerido

1. ¿Cómo funciona la cámara sintética glm::lookAt?

Esta función crea una matriz de vista la cual simula una cámara. Se utiliza para transformar coordenadas del mundo en coordenadas de cámara para una vista 3D. Recibe tres parámetros, todos vectores `vec3`:

- `position`: Especifica la posición de la cámara en las coordenadas mundiales.
- `target`: Especifica la posición/dirección del objetivo hacia la que debe mirar la cámara.
- `up`: Especifica un vector que apunta en la dirección Y positiva utilizada para crear el vector correcto. Normalmente se establece en $(0.0f, 1.0f, 0.0f)$.

2. ¿Cómo funciona la matriz de vista en el shader? (Versión moderna de OpenGL)

La matriz de vista transforma las coordenadas de los objetos desde el espacio del mundo al espacio de la cámara. En el espacio de la cámara, la cámara está situada en el origen $(0,0,0)$ y mira hacia el eje negativo de z . La matriz se pasa como uniforme al vertex shader para que este aplique la matriz a los vértices para transformarlos en espacio de recorte.

Para usar la matriz se puede emplear la función `glm::lookAt`, se obtiene la ubicación del uniforme del shader con `glGetUniformLocation` y con estos dos parámetros se crea una matriz de tipo `glUniformMatrix4fv`.

3. ¿Cuáles son las funciones basadas en senos y cosenos para generar por medio de triángulos: esfera, cilindro y cono?

Las funciones principales para la generación de este tipo de figuras son `GL_TRIANGLE_STRIP` y `GL_TRIANGLE_FAN`, pero los cálculos de los vértices que reciben estas funciones se deben hacer manualmente con ayuda de senos y cosenos:

- Esfera $\rightarrow x = r \cos(\theta) \sin(\phi)$, $y = r \sin(\theta) \sin(\phi)$, $z = r \cos(\phi)$
- Cilindro $\rightarrow x = r \cos(\theta)$, $y = r \sin(\theta)$, $z = h$
- Cono $\rightarrow x = r(1 - h/H) \cos(\theta)$, $y = r(1 - h/H) \sin(\theta)$, $z = h$

Donde: $\theta \rightarrow$ rotación horizontal, $\phi \rightarrow$ rotación vertical, $r \rightarrow$ radio del círculo, $h \rightarrow$ altura, $H \rightarrow$ altura total y en el cono: $h \rightarrow$ variable de 0 a H .

2. Conclusión

Gracias a la investigación realizada para este previo se comprendió que la función `glm::lookAt` simula la función de una cámara sintética por medio de una matriz de vista, esta cámara está ubicada en una posición específica y mirando a un punto/objeto determinado. Los datos de esta matriz se pasan a los shaders para transformar las coordenadas del objeto al espacio de la cámara y al espacio de recorte.

Finalmente, se entendió que para la generación de figuras más complejas como la esfera, cilindro y cono se pueden generar por medio de triángulos de manera poligonal. Además, el cálculo de los vértices necesarios para estas figuras se hacen por medio de operaciones que involucren senos y cosenos.

Bibliografía

- Instructor, T. (2023, junio 22). *extern*. <https://www.makigas.es/series/tutorial-de-c/extern>
- *LearnOpenGL - camera*. (s/f). Learnopengl.com. Recuperado el 21 de febrero de 2025, de <https://learnopengl.com/Getting-started/Camera>
- *Práctica 1. Introducción a OpenGL en Delphi*. (s/f). Cimat.mx. Recuperado el 25 de febrero de 2025, de https://www.cimat.mx/~fory/geocomp/practica2/_history/index.html.~11~
- *Práctica 5 Transformada de vista y transformada de proyección*. (s/f). Informatica.uv.es. Recuperado el 21 de febrero de 2025, de <http://informatica.uv.es/iiguia/IG/prac5.pdf>