



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° 03

NOMBRE COMPLETO: Hernández Vázquez Daniela

N° de Cuenta: 318092867

GRUPO DE LABORATORIO: 01

GRUPO DE TEORÍA: 02

SEMESTRE 2024-2

FECHA DE ENTREGA LÍMITE: 2 de marzo de 2024

CALIFICACIÓN: _____

REPORTE DE PRÁCTICA:

Actividad.

La actividad consista en generar una pirámide de rubik (pyraminx) de 9 pirámides por cara. Cada cara de la pyraminx debía de ser de un color diferente y que se vieran las separaciones entre instancias (las líneas oscuras son las que permiten diferenciar cada pirámide pequeña).

Utilizamos como base la practica revisada en clase apoyándonos con el video, lo primero que realizamos es modificar a la pirámide triangular para evitar escalarla con las dimensiones necesarias, para calcular cual era la ubicación de sus puntos nos apoyamos con la aplicación de GeoGebra3D.

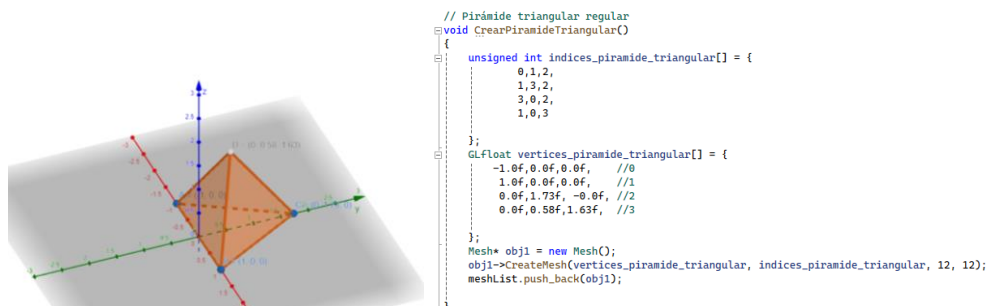


Figura 3.1 Modificación de primitiva

Posteriormente modificamos el color base de la ventana de negro a blanco y comenzamos a instanciar las figuras tomando como base una pirámide principal.

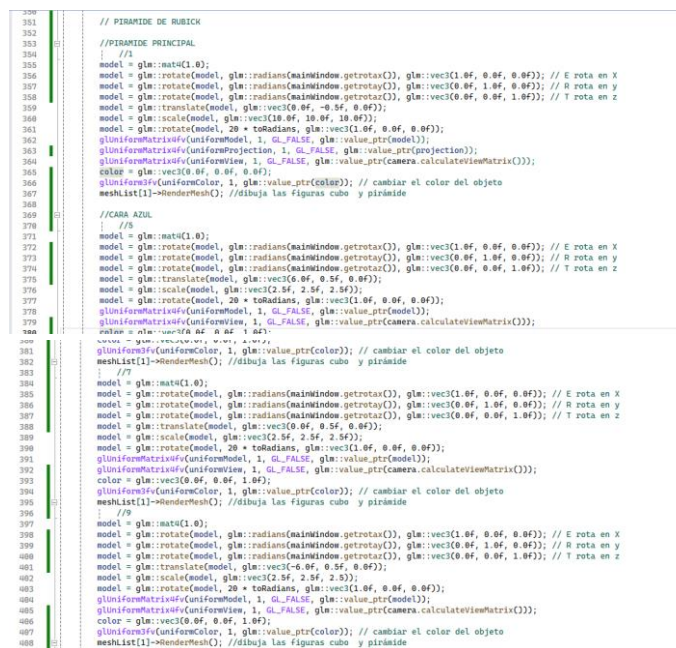


Figura 3.2 Instanciar figuras

Se instanciaron un total de 37 pirámides (incluyendo la base) cada una de estas figuras era instanciada con su respectiva traslación y rotación así como con los comandos de color e inicialización de la cámara, la cual era controlada con el movimiento del mouse y las teclas E (rotación en x), R (rotación en y), T (rotación en z), S (alejar), W (acercar), etc.

Se utiliza un solo modelo de matriz para instanciar todas las figuras inicializándola en cada interacción. Finalmente, luego de haber instanciado todas las figuras en cada cara de la pirámide obtenemos los resultados de la figura 3.3.

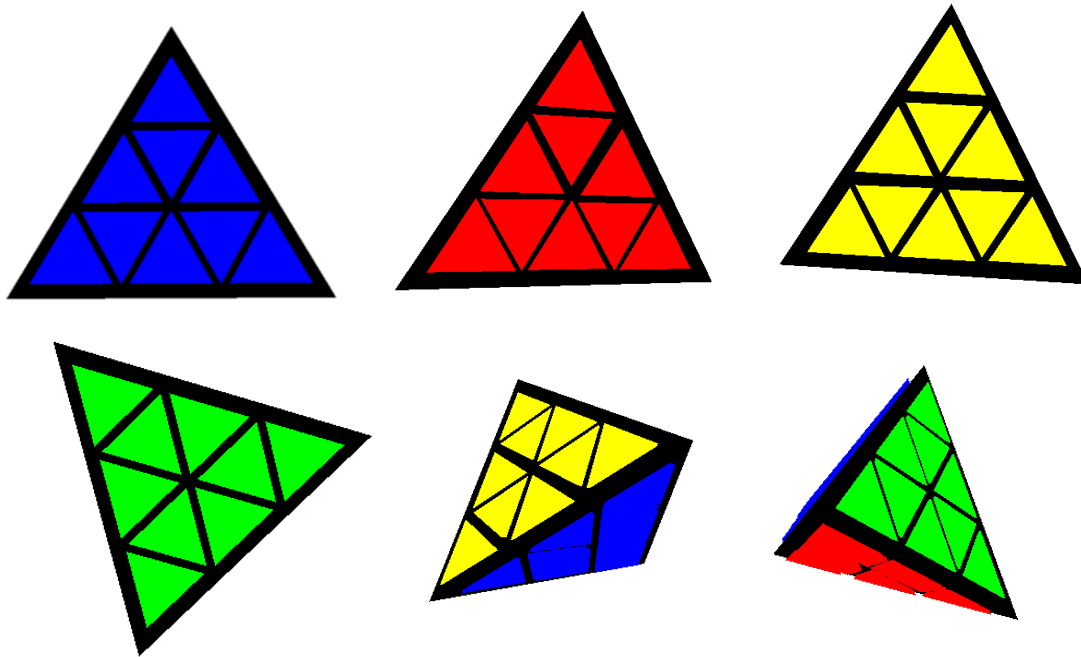


Figura 3.3 Pirámide creada

Problemas:

En esta práctica se tuvieron muchos más problemas que con las prácticas anteriores. La creación de este código fue mucho más complicada de lo que esperaba en un inicio.

En primer lugar, al momento de instanciar observé que si los comandos de la cámara para cada elemento se colocan después de haber realizado la rotación y traslación al mismo provocaba que funcionaran independientemente y no como un conjunto, por ejemplo, al rotar con R.

Dado a que la figura tenía ángulos fue posible solo trabajar bien con las coordenadas de 2 caras que eran paralelas a alguno de los planos, en este caso la cara azul y verde. Por su parte fue más complicado realizar las caras con ángulo ya

que las figuras debían de tener una rotación adicional para colocar los triángulos invertidos, así que se fue realizado en estas 2 caras una acomodación por tanteo de las pirámides ejecutando el programa varias veces para realizar las modificaciones, aun con esto estas caras no queda totalmente simétricas a comparación de las otras.

Como en la clase se había mencionado se intentó optimizar el número de elementos de la figura pero no pude comprender como hacerlo, al principio pensé en que sería más fácil colocarle una textura, posteriormente pensé en introducir otras figuras geométricas como una esfera que no tuviera una gran cantidad de lados y que solo sobresalieran algunas de sus partes desde el centro o inclusive mover la figura por bloques pero esto no se podía hacer, así que forzosamente se modificó elemento a elemento.

Conclusión:

Esta actividad fue mucho más compleja que las anteriores prácticas, ya que se necesitaba tener un buen análisis espacial. Fue especialmente difícil manejar al rotación y traslación sin tener un sistema de referencia en la ventana y más aún porque no es muy cómodo que se mueva la figura al colocar el mouse o apretar las teclas. La necesidad de coordinar la cámara con los movimientos del mouse y las teclas, junto con la rotación y traslación de las pirámides, resultó algo complicado.

En cuanto a la explicación de clase me pareció buena para poder tener los elementos para modificar el programa, aunque me hubiera gustado que se profundizara más con el uso de la cámara y el cómo afectan el orden de las instrucciones el programa en cuanto a la vista.

Referencias:

Kessenich, J. M., Sellers, G., & Shreiner, D. (2016). *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V*. OpenGL.

Méndez Servín, M. (2022). *NOTAS PARA EL CURSO DE GRAFICACIÓN POR COMPUTADORA* [Libro electrónico].
https://prometeo.matem.unam.mx/recursos/VariosNiveles/iCartesiLibri/recursos/Notas_Graficacion_por_Computadora/index.html

RGB 0-1 Color Picker. (s. f.). <https://rgbcolorpicker.com/0-1>

Geogebra. Calculadora 3D (s. f.). <https://www.geogebra.org/3d?lang=es>