



---

# Benemérita Universidad Autónoma de Puebla.

---

Tarea #9.



GRAFICACIÓN  
Pedro Vargas Arenas  
19/Marzo/2019

## **INTRODUCCIÓN.**

De manera introductoria al entorno 3D con OpenGL, se realizó esta práctica que consistía en rotar una figura en 3D en función de los ejes X, Y, Z. Para esto se aplica el concepto de rotación en 3D que es distinto a la rotación en 2D debido a que la matriz que se va a operar con los puntos, depende del eje de referencia a rotar. Sin embargo, en este caso se usaron las tres matrices ya que se pedían la rotación sobre los tres ejes y además, esta se iniciara presionando las teclas respectivas.

## CONCEPTOS DESARROLLADOS.

Para la realización de la práctica se estudió el nuevo concepto de rotación en 3D, para esto se utilizan matrices dependiendo del eje de referencia para rotar.

Se comenzó dibujando los ejes x, y, z. Después se asignaron los puntos para dibujar una pirámide cuadrangular, esta es la figura que se rotará sobre los ejes. Cada punto es almacenado en un arreglo de tipo flotante, al ser una figura con cinco puntos, se tienen cinco arreglos con los puntos y otros cinco arreglos que sirven como auxiliares, en estos se almacenarán los puntos ya operados.

```
float p1[4] = {-1, -1, -1, 1};
float p2[4] = {1, -1, -1, 1};
float p3[4] = {1, -1, 1, 1};
float p4[4] = {-1, -1, 1, 1};
float p5[4] = {0, 1, 0, 1};
float aux1[4], aux2[4], aux3[4], aux4[4], aux5[4];
```

Para las matrices, se tiene una clase *Matriz*, la cual contiene funciones para inicializar las tres matrices respectivamente y una función para operar cada matriz con los puntos que se reciban. Las matrices son:

Para el eje X:

```
void Matriz::setMatrizX(float grados){
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para el eje Y:

```
void Matriz::setMatrizY(float grados){
```

$$\begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para el eje Z:

```
void Matriz::setMatrizZ(float grados){
```

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

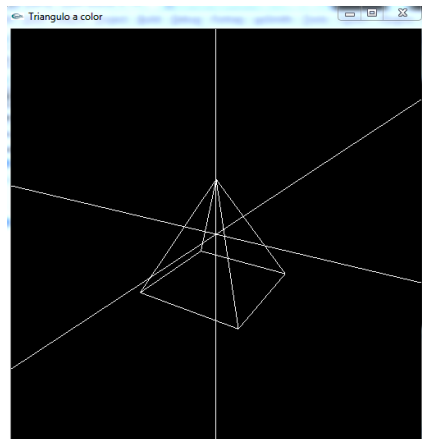
Para operar las matrices con los puntos, se tiene una función para operar con la matriz de X, Y y Z respectivamente. Estas funciones reciben el arreglo que tiene las coordenadas del punto y regresa el punto ya operado.

Las coordenadas del punto ya operado se guardan en los arreglos auxiliares, para esto, dentro de la clase *main* se tiene cinco funciones (una para cada punto de la figura) que reciben las coordenadas operadas. Para iniciar la rotación de la figura, se tiene una función *keys* la cual recibe una salida del teclado y realiza una acción dependiendo de lo captado. Usando esto, se asigna las teclas "x" y "X" para aumentar el grado de rotación y se asigna un 1 a una variable que se usa para saber qué eje se seleccionó, en este caso, el eje x. Las teclas "y" y "Y" realizan la misma acción para el eje y y a la variable del eje se iguala a 2. Las teclas "z" y "Z" realizan la misma acción para el eje z y a la variable del eje se iguala a 3. Por último, cualquier tecla exceptuando las anteriores reinicia los puntos de la figura.

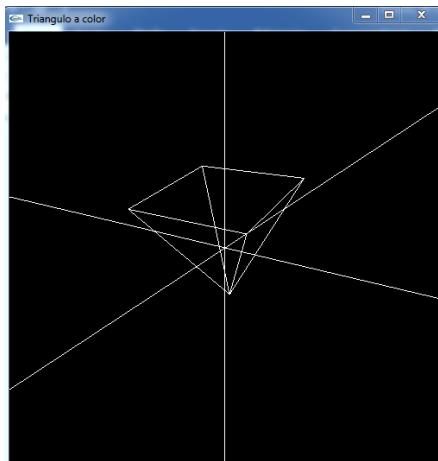
Adentro de la función *display* está un switch-case que ve cuál es el eje seleccionado y aplica la rotación. Este mismo proceso que realiza una y otra vez

## PRUEBAS REALIZADAS.

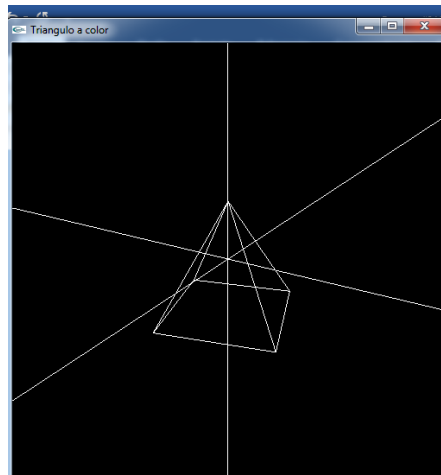
Figura en estado inicial y con los ejes dibujados.



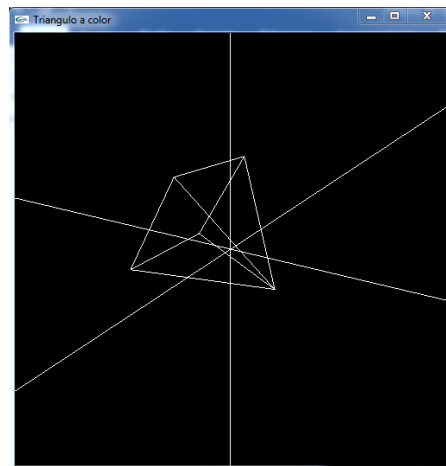
Rotación sobre el eje x.



Rotación sobre el eje y.



Rotación sobre el eje z.



## **CONCLUSIONES.**

La práctica tuvo cierta dificultad debido a la asignación de puntos, ya que las asignaciones no se realizaban correctamente y esto resultaba en una rotación mal ejecutada. Sin embargo, resultó útil para poner en práctica el dibujo de figuras en 3D y el concepto de rotación en esta dimensión se reforzó.