



Práctica 3.- Estudio de las Transformaciones Afines

Objetivos:

Estudiar la implementación y acción de los diferentes tipos de transformaciones afines y de sólido rígido sobre una geometría en 3D.

Requerimientos:

Transformaciones afines.
Concepto de Puerto de Vista y Ventana del Mundo
Transformación de sistemas de coordenadas.

Introducción:

NOTAS.

La máquina de estados de la open Gl aplicará la matriz de transformación que hay acumulada en la pila de MODELVIEW a toda la geometría.

Vamos a considerar la pila de matrices que utiliza la OpenGL para aplicar transformaciones a la geometría. Esta pila se denomina MODELVIEW, y se activa con la instrucción

```
glMatrixMode (MODELVIEW);
```

La orden

```
glMultMatrix(float matriz[16]);
```

multiplica la matriz que le pasamos como argumento al contenido de la pila de matrices de transformación.

La matriz a multiplicar se pasa como un vector de 16 componentes (para geometrías en 3D) con el siguiente significado:

```
matriz[16] = {n0,n1,n2,....,n15} ->  n0  n4  n8  n12  
                                     n1  n5  n9  n13  
                                     n2  n6  n10 n14  
                                     n3  n7  n11 n15
```

Para limpiar inicialmente la pila de matrices se llama a la función

```
glLoadIdentity();
```

que carga la pila con la matriz identidad.

En esta práctica usaremos estas dos instrucciones para manejar la pila. Sin embargo hay que ser conscientes de que la OpenGL nos da otras facilidades como son:

`PushMatrix()`; Establece una marca en la pila. A partir de ésta, todas las matrices de transformación

que se introduzcan en la pila, podrán ser desapiladas de un golpe con la instrucción

```
PopMatrix();
```

Aunque las transformaciones las vamos a aplicar directamente construyendo la matriz nosotros mismos, existen una serie de transformaciones que OpenGL puede realizar por nosotros, simplemente indicando los datos

```
glTranslatef(float x, float y, float z); //x,y,z son los valores de traslación en cada eje
```

```
glRotatef(float angulo, float x, float y, float z); // Se gira en el sentido positivo (en contra de las agujas del reloj, un ángulo expresado en GRADOS sobre un eje que pasa por el origen y el punto (x,y,z).
```

```
glScalef(float x, float y, float z); //Escala con factores Sx, Sy, Sz
```



Enunciado:

Cada una de las opciones que se piden, debe ser un procedimiento, para poder corregirlas adecuadamente.

1. Operaciones de Sólido rígido

Para situar los objetos en la escena, dibuja los tres ejes de coordenadas que vayan desde el origen hasta una distancia de 3.0. Puedes pintarlos de diferente color para distinguirlos.

Visualiza el objeto “tetera” Este objeto tiene sus coordenadas de objeto centradas en el origen de coordenadas, con lo que aparecerá en el centro del sistema de coordenadas que has trazado.

```
glutWireTeapot(2.0);
```

El parámetro indica la dimensión de la tetera. Para un cubo de visualización (ventana del mundo) de extremos $-5,5$ en cada eje, este factor es adecuado.

Define dos variables globales

```
trasX que almacena la traslación sobre el eje X  
angulo que almacena la rotación.
```

a)

1. Usa el evento del ratón izdo. para provocar sucesivas traslaciones en el objeto sobre el eje x de una magnitud de 0.5 positivo en las unidades de la escala.

Para ello tendrás que actualizar la matriz de MODELVIEW e incrementar la variable trasX cada vez que sea pinchado el botón izdo.

```
trasX += 0.5;
```

La matriz de MODELVIEW es actualizada en la propia función que aplica la transformación. La última instrucción del caso de haber pulsado el botón izdo. debe ser

```
glutPostRedisplay();
```

que fuerza a que se redibuje el objeto.

2. Igualmente usa el botón dcho. del ratón para crear una traslación de -0.5 unidades en el eje x

```
trasX -= 0.5;
```

Piensa que el efecto de la traslación dependerá de la escala que utilicemos en la visualización. Esta escala la proporciona el tamaño de la ventana del mundo (en 3D el prisma de visión) que se configura con la orden

```
glOrtho(left, right, bottom, top, near, far);
```

b) Usa el evento de pulsar la tecla ‘+’ para provocar una rotación positiva de $\pi/7$ respecto del eje X

Usa el evento de pulsar la tecla ‘-’ para provocar una rotación negativa de $-\pi/7$ respecto del eje X

c) Dibuja el objeto trasladado a una distancia (6.3, 7.8, 2.5) y girado en $\pi/10$ respecto del eje X y en $\pi/7$ respecto del eje Z.

d) Calcula la matriz de sólido rígido que lleva la dirección del eje z de la tetera a la dirección (3, 2, -4). Constrúyela y aplícala a la tetera.

2. Operaciones de transformaciones afines

a) Utiliza la interacción con las teclas ‘e’ y ‘E’ para realizar escalados uniformes incrementales. Cada pulsación a la tecla ‘E’ debe incrementar el tamaño del objeto en 1.3 respecto de los tres ejes. Por el contrario con la tecla ‘e’ realizaremos escalados uniformes de factor 0.7

b) Aplica una composición de transformaciones para conseguir que el objeto se encuentre en la posición $T(3, 5.5, 6, 1)$, girado un ángulo $\pi/4$ respecto del eje x y $\pi/3$ respecto del eje z y escalado con $S_y = 0.5$.



3. Creación de escenas

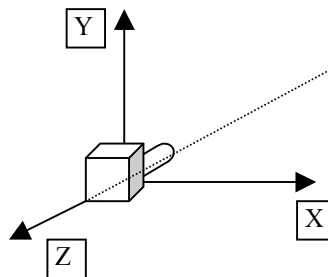
a) Sitúa una tetera a una distancia $(0,1,0)$. Sitúa un cubo encima de la tetera a una distancia $y=1$ de ésta. Sitúa una esfera encima del cubo a la misma distancia.

b) Dibuja una composición de 12 teteras dispuestas en círculo alrededor del punto $(1,1,1)$ como si fuera la esfera de un reloj y cada tetera estuviera situada en una hora.

4. Generación de una vista de la escena

Considera la escena que has diseñado en el apartado 3b). Vamos a observarla desde un determinado punto de vista.

Por defecto en OpenGL nosotros observamos la escena desde el origen de coordenadas mirando hacia la dirección negativa del eje Z.



Vamos a hacer como si estuviéramos observando la escena desde la posición $(5,5,5)$ y centráramos nuestros ojos en el centro del corro de teteras (punto $(1,1,1)$). Para crear este efecto, esto es equivalente a cambiar nuestro sistema de referencia a otro sistema de referencia y observarlo todo desde allí. Por lo que deberemos orientar el eje de coordenadas Z en la dirección de la recta que forman los puntos $(1,1,1)$ y $(5,5,5)$ y en sentido contrario (como huyendo del punto $(1,1,1)$).

Multiplica la matriz adecuada a la escena de las teteras para observarla desde ese punto de vista.