

Tema 2: Transformaciones Geométricas

José Ribelles

SIU020 Síntesis de Imagen y Animación

Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I

Contenido

- 1 Introducción
- 2 Transformaciones básicas
 - Traslación
 - Escalado
 - Rotación
- 3 Concatenación de Transformaciones
- 4 Matriz de Transformación de la Normal
- 5 Giro alrededor de un Eje arbitrario
- 6 Transformaciones en OpenGL

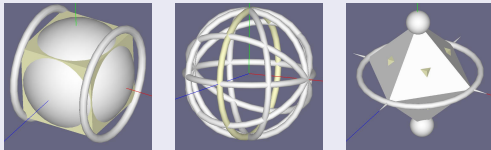
Hoy veremos...

- 1 **Introducción**
- 2 Transformaciones básicas
- 3 Concatenación de Transformaciones
- 4 Matriz de Transformación de la Normal
- 5 Giro alrededor de un Eje arbitrario
- 6 Transformaciones en OpenGL

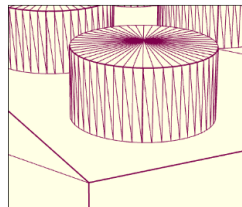
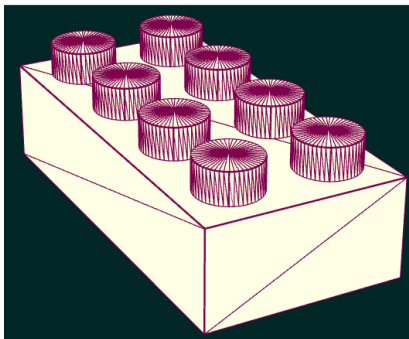
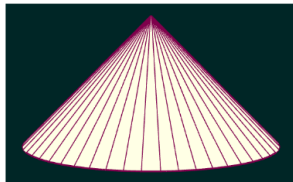
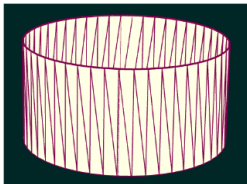
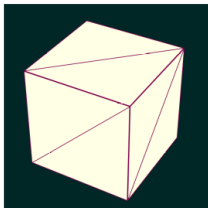
Introducción

¿Por qué las necesitamos?

- En la etapa de modelado los objetos se definen bajo un sistema de coordenadas propio.
- A la hora de crear una escena, estos objetos se incorporan bajo un nuevo sistema de coordenadas conocido como sistema de coordenadas del mundo.
- Este cambio de sistema de coordenadas es necesario y se realiza mediante transformaciones geométricas.



Introducción



Hoy veremos...

- 1 Introducción
- 2 Transformaciones básicas**
 - Traslación
 - Escalado
 - Rotación
- 3 Concatenación de Transformaciones
- 4 Matriz de Transformación de la Normal
- 5 Giro alrededor de un Eje arbitrario
- 6 Transformaciones en OpenGL

Transformaciones básicas

Traslación

- Consiste en desplazar el punto $p = (p_x, p_y, p_z)$ mediante un vector $t = (t_x, t_y, t_z)$ de manera que el nuevo punto $q = (q_x, q_y, q_z)$:

$$q_x = p_x + t_x, \quad q_y = p_y + t_y, \quad q_z = p_z + t_z \quad (1)$$

- La representación matricial con coordenadas homogéneas:

$$T(t) = T(t_x, t_y, t_z) = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

- Utilizando esta representación, el nuevo punto se obtiene así:

$$\tilde{q} = T(t) \cdot \tilde{p} \quad (3)$$

Escalado

- Consiste en multiplicar el punto $p = (p_x, p_y, p_z)$ con los factores de escala s_x , s_y y s_z de tal manera que el nuevo punto $q = (q_x, q_y, q_z)$:

$$q_x = p_x \cdot s_x, \quad q_y = p_y \cdot s_y, \quad q_z = p_z \cdot s_z \quad (4)$$

- La representación matricial con coordenadas homogéneas:

$$S(s) = S(s_x, s_y, s_z) = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

- Utilizando esta representación, el nuevo punto se obtiene así:

$$\tilde{q} = S(s) \cdot \tilde{p} \quad (6)$$

Rotación

- Gira un punto un ángulo ϕ alrededor de un eje de coordenadas:

- Alrededor de X:

$$\begin{aligned}x' &= x \\y' &= y \cos \phi - z \sin \phi \\z' &= y \sin \phi + z \cos \phi\end{aligned}\tag{7}$$

- Alrededor de Y:

$$\begin{aligned}x' &= x \cos \phi + z \sin \phi \\y' &= y \\z' &= -x \sin \phi + z \cos \phi\end{aligned}\tag{8}$$

- Alrededor de Z:

$$\begin{aligned}x' &= x \cos \phi - y \sin \phi \\y' &= x \sin \phi + y \cos \phi \\z' &= z\end{aligned}\tag{9}$$

Rotación

- Gira un punto un ángulo ϕ alrededor de un eje.
- La representación matricial con coordenadas homogéneas:

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (10)$$

$$R_y(\phi) = \begin{pmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (11)$$

$$R_z(\phi) = \begin{pmatrix} \cos \phi & -\sin \phi & 0 & 0 \\ \sin \phi & \cos \phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (12)$$

- Utilizando esta representación, el nuevo punto se obtiene así:

$$\tilde{\mathbf{q}} = R(\phi) \cdot \tilde{\mathbf{p}} \quad (13)$$

Hoy veremos...

- 1 Introducción
- 2 Transformaciones básicas
- 3 Concatenación de Transformaciones**
- 4 Matriz de Transformación de la Normal
- 5 Giro alrededor de un Eje arbitrario
- 6 Transformaciones en OpenGL

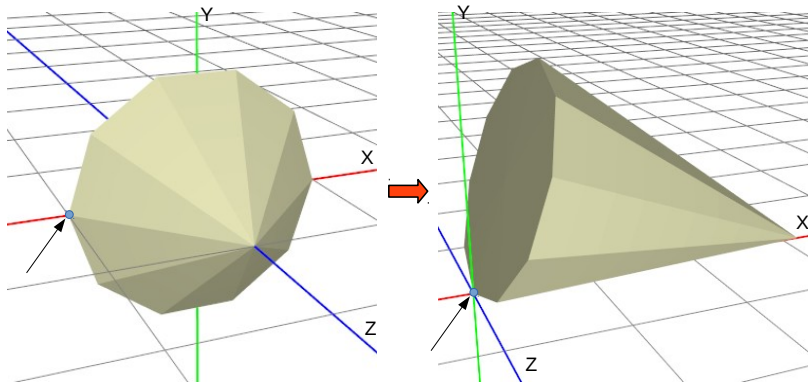
Concatenación de Transformaciones

Descripción

- Las transformaciones geométricas en su forma matricial con coordenadas homogéneas se pueden concatenar.
- Una sola matriz puede representar a toda una secuencia de matrices de transformación.
- Es muy importante operar la secuencia de transformaciones en el orden correcto ya que el producto de matrices no posee la propiedad conmutativa.
 - Dibuja en el papel como quedaría una esfera de radio unidad centrada en el origen dadas las dos siguientes matrices de transformación $T(5, 0, 0)$ y $S(5, 5, 5)$ si las matrices se multiplican de las dos formas posibles, es decir, $M = T \cdot S$ y $M = S \cdot T$.
- El producto de matrices sí posee la propiedad asociativa:
 - $M = R \cdot T \cdot S = R \cdot (T \cdot S) = (R \cdot T) \cdot S$

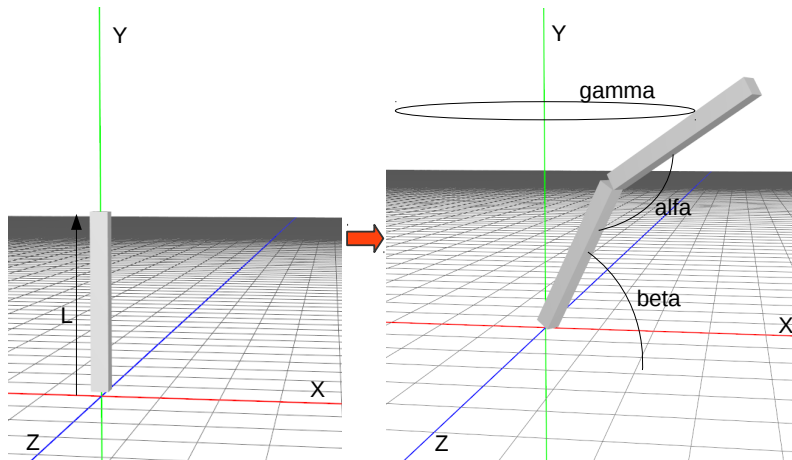
Ejercicio

Determina las transformaciones que sitúan el objeto que se muestra a la izquierda de la siguiente figura (radio de la base y altura uno) en la posición que se muestra en la derecha (radio de la base uno y altura tres).



Ejercicio

Determina las transformaciones que sitúan el objeto que se muestra a la izquierda de la siguiente figura en las posiciones que se muestran en la derecha.



Hoy veremos...

- 1 Introducción
- 2 Transformaciones básicas
- 3 Concatenación de Transformaciones
- 4 Matriz de Transformación de la Normal**
- 5 Giro alrededor de un Eje arbitrario
- 6 Transformaciones en OpenGL

Matriz de Transformación de la Normal

Descripción

- La matriz de transformación no siempre es válida para los vectores normales a la superficie, ¿en qué casos ocurre?
- Lo habitual es que la matriz de transformación de la normal N sea la traspuesta de la inversa de la matriz de transformación.
 - Como la normal es un vector y la traslación no le afecta, y el escalado y la rotación son transformaciones afines, solo hay que trabajar con los 3x3 componentes superior izquierda.
- Las longitudes de las normales no se preservan si hay una transformación de escalado, por lo que es necesario normalizarlas, ¿y si es uniforme y conoces el factor de escala?

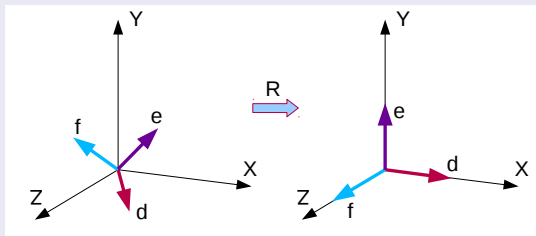
Hoy veremos...

- 1 Introducción
- 2 Transformaciones básicas
- 3 Concatenación de Transformaciones
- 4 Matriz de Transformación de la Normal
- 5 Giro alrededor de un Eje arbitrario**
- 6 Transformaciones en OpenGL

Giro alrededor de un Eje arbitrario

Descripción

- Sean d y ϕ el vector unitario del eje de giro y el ángulo de giro.
- Hay que calcular una base ortogonal que contenga a d .
- La idea es hacer un cambio de base entre la base que forman los ejes de coordenadas y la nueva base.



Cálculo de la matriz R

- La matriz que representa al cambio de base es esta:

$$R = \begin{pmatrix} d_x & d_y & d_z \\ e_x & e_y & e_z \\ f_x & f_y & f_z \end{pmatrix} \quad (14)$$

- e es un vector unitario normal a d .
- f es el producto vectorial de los otros dos vectores $f = d \times e$
- El vector e se puede obtener de la siguiente manera:
 - Partiendo del vector d haz cero su componente de menor magnitud.
 - Intercambia los otros dos componentes y niega uno de ellos.
 - Normalízalo.
- Teniendo en cuenta que R es ortogonal, la matriz de rotación final:

$$R_d(\phi) = R^T R_x(\phi) R \quad (15)$$

Hoy veremos...

- 1 Introducción
- 2 Transformaciones básicas
- 3 Concatenación de Transformaciones
- 4 Matriz de Transformación de la Normal
- 5 Giro alrededor de un Eje arbitrario
- 6 Transformaciones en OpenGL**

Transformaciones en OpenGL

La biblioteca glmMatrix

Proporciona funciones tanto para la construcción de las matrices de transformación como para operar con ellas:

- `mat4.fromTranslation (out, v)`
- `mat4.fromScaling (out, v)`
- `mat4.fromRotation (out, rad, axis)`

```
// Para obtener la matriz de transformacion del modelo
var modelMatrix = mat4.create();
var T = mat4.create();
var R = mat4.create();
mat4.fromTranslation (T, [0,0.2,0]);
mat4.fromRotation (R, Math.PI/2, [1,0,0]);
mat4.multiply (modelMatrix, T, R);

// Para obtener la matriz de transformacion de la normal
var normalMatrix = mat3.create();
mat3.normalFromMat4 (normalMatrix, modelMatrix);
```

¿Dónde calcular las matrices?

La construcción de ambas matrices, la matriz de transformación del modelo y la matriz de transformación de la normal, es conveniente que tenga lugar en la aplicación y que ambas se suministren al procesador de vértices.



Listado 1: Shader para transformar la posición y la normal de cada vértice

```
uniform mat4 modelMatrix; // matriz de transformacion del modelo
uniform mat3 normalMatrix; // matriz de transformacion de la normal

in vec3 VertexPosition, VertexNormal;
out vec3 VertexNormalT;

void main ()
{
    VertexNormalT = normalize (normalMatrix * VertexNormal);
    gl_Position = modelMatrix * vec4(VertexPosition, 1.0);
}
```

Ejercicios

- Dada una matriz de transformación que al operarla con un modelo produce su simétrico, ¿qué le ocurren a las normales si las operamos con esa misma matriz?
- ¿Qué transformaciones debería sufrir el cubo según el siguiente fragmento de código?

Listado 2: Ejemplo de transformación geométrica de un cubo

```
var M = mat4.create();
var S = mat4.create();
var T = mat4.create();

var iM = gl.getUniformLocation(program, "M");

mat4.fromScaling(S, [2, 2, 2]);
mat4.fromTranslation(T, [1, 1, 1]);

mat4.multiply(M, T, S);

gl.uniformMatrix4fv(iM, false, M);

draw(exampleCube);
```