

Computación Gráfica

Ing. Gabriel Ávila, MSc.

Transformaciones

Transformaciones

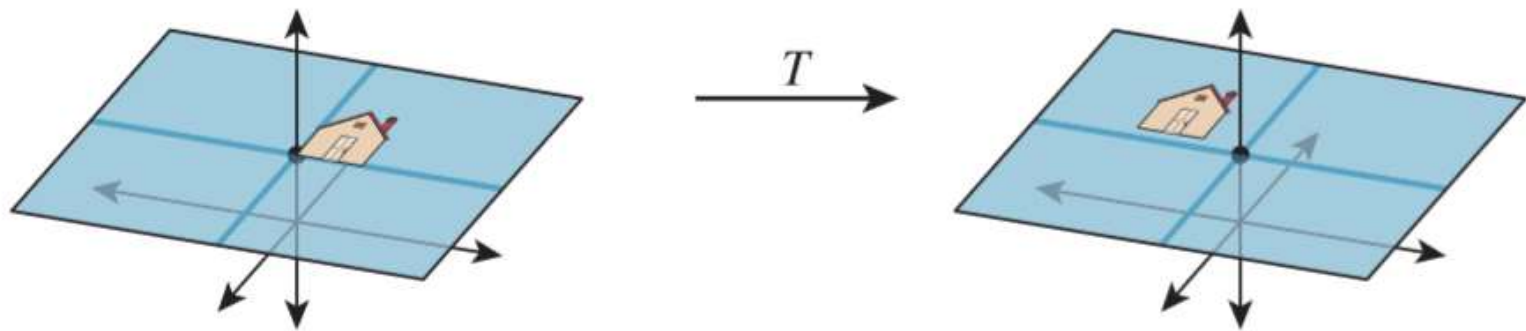
Hay que preguntarse ¿Qué se está transformando? Es posible transformar diferentes elementos:

- Vértices.
- Geometrías.
- Objetos completos.
- La cámara.
- El mundo (El sistema de coordenadas).
- etc.

Traslación 2D

Para realizar una traslación se hace uso de la **matriz de traslación**. Esta matriz permite desplazar un punto (o varios puntos de un objeto) en el espacio, sin aplicar ninguna rotación ni cambio de escala.

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} \quad \begin{array}{l} t_x : \text{traslación en } x \\ t_y : \text{traslación en } y \end{array}$$



Traslación inversa en 2D

Si se desea reversar una traslación, es necesario realizar la traslación inversa:

$$\mathbf{P} = \mathbf{T}^{-1} \cdot \mathbf{P}'$$

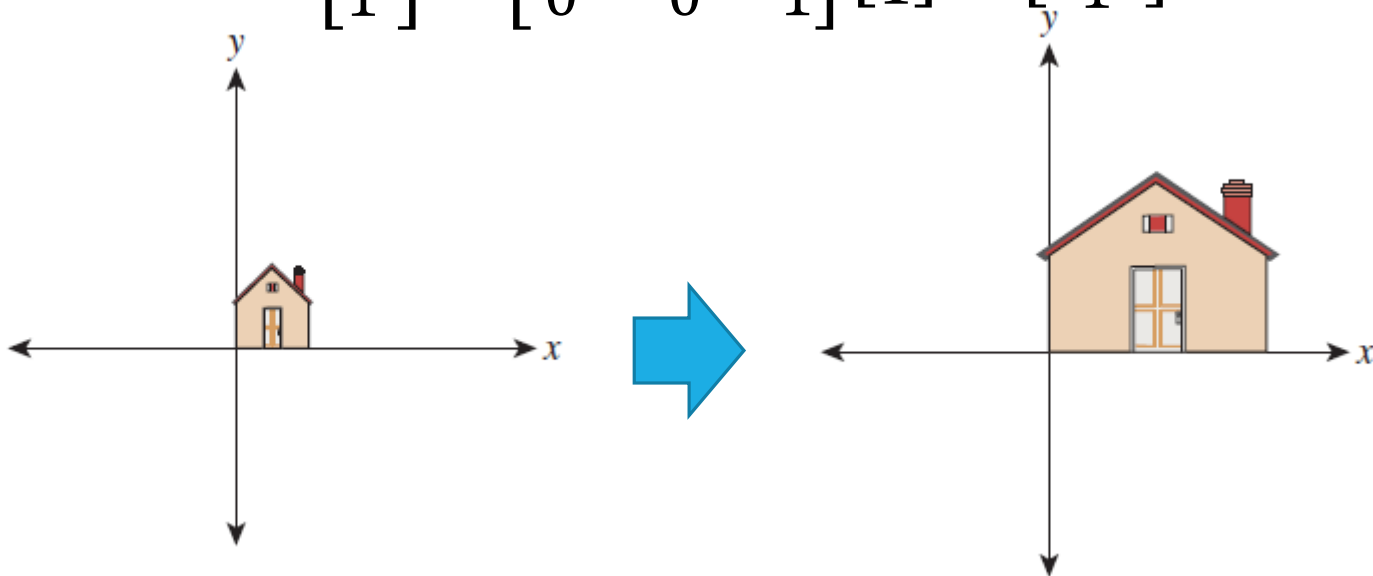
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$

De esta manera es posible regresar al punto original.

Escala 2D

Al aplicar un cambio de escala, lo que se desea hacer es escalar las coordenadas de uno o varios puntos. Ej: $(x, y) \rightarrow (2x, 3y)$

$$\begin{bmatrix} x' \\ y' \\ 1' \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix}$$



Escala inversa 2D

Si se desea reversar un cambio en escala:

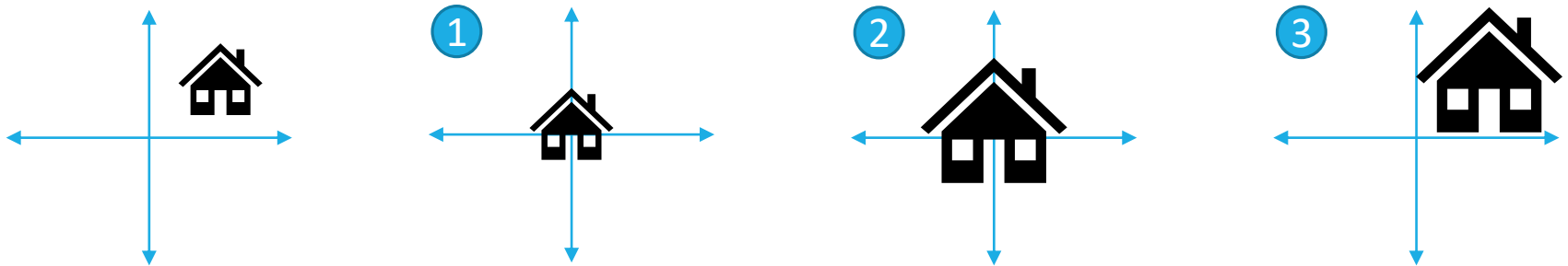
$$\mathbf{P} = \mathbf{S}^{-1} \cdot \mathbf{P}'$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1/S_x & 0 & 0 \\ 0 & 1/S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x'/S_x \\ y'/S_y \\ 1 \end{bmatrix}$$

Escala, *Con respecto al origen local*

Para lograr esto, es necesario aplicar las siguientes transformaciones:

1. Trasladar el objeto al origen.
2. Escalar.
3. Trasladar el objeto nuevamente a su posición original.



De lo contrario, el objeto se escalará con respecto al origen

Escala 2D

La matriz resultante será entonces:

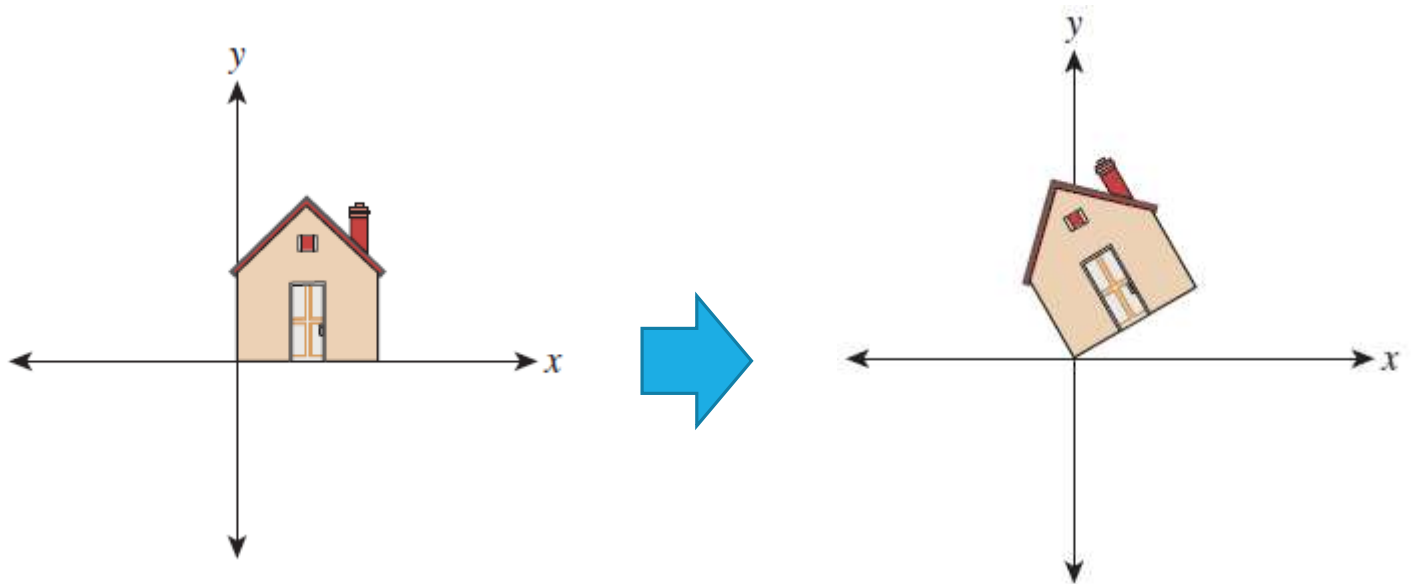
$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{T}^{-1} \cdot \mathbf{P}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & t_x(1 - S_x) \\ 0 & S_y & t_y(1 - S_y) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

De lo contrario, el objeto se escalará con respecto al origen

Rotaciones

Una rotación implica mover un punto un cierto ángulo θ , formando un arco con respecto a un punto (2D) o a un vector (3D).



Rotaciones 2D

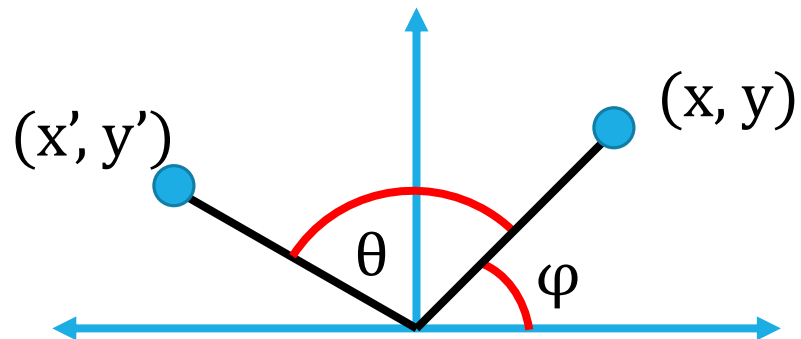
Se desea rotar un punto (x, y) , un ángulo θ con respecto al origen:

$$x = r \cos \varphi$$

$$y = r \sin \varphi$$

$$x' = r \cos(\theta + \varphi)$$

$$y' = r \sin(\theta + \varphi)$$



Utilizando las identidades trigonométricas de la suma de ángulos:

$$x' = r \cos(\theta) \cos(\varphi) - r \sin(\theta) \sin(\varphi)$$

$$\mathbf{x' = x \cos(\theta) - y \sin(\theta)}$$

$$y' = r \sin(\theta) \cos(\varphi) + r \cos(\theta) \sin(\varphi)$$

$$\mathbf{y' = x \sin(\theta) + y \cos(\theta)}$$

Rotaciones 2D

A partir de las dos ecuaciones obtenidas:

$$x' = x \cos(\theta) - y \sin(\theta)$$

$$y' = x \sin(\theta) + y \cos(\theta)$$

Se puede reformular, utilizando matrices:

$$\begin{bmatrix} x' \\ y' \\ 1' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \cos(\theta) & -y \sin(\theta) \\ x \sin(\theta) & +y \cos(\theta) \\ 1 \end{bmatrix}$$

Obteniendo la matriz de rotación por un ángulo θ en el origen:

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

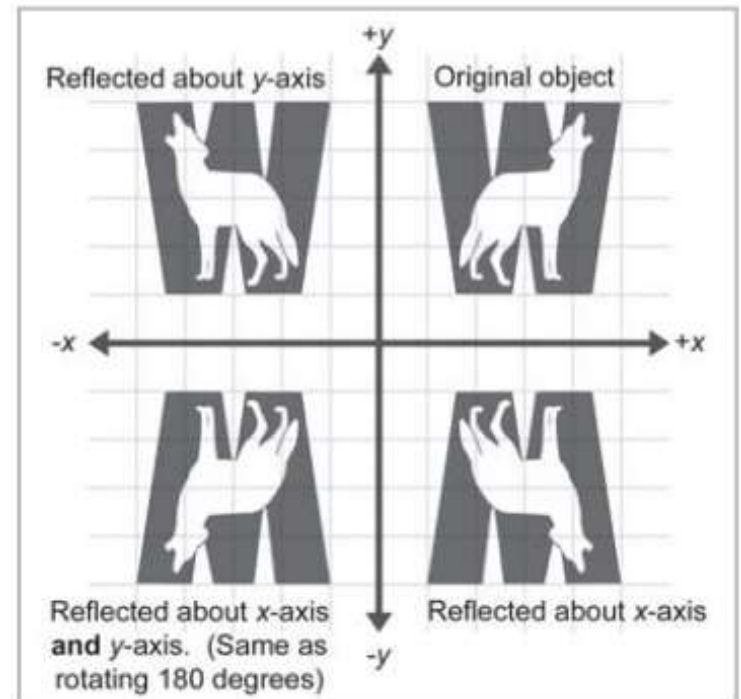
Reflection

Transformación de espejo

Refleja el elemento con respecto a una línea (en 2D).

En 2D, con respecto a un eje, que pasa por el origen y es perpendicular a n :

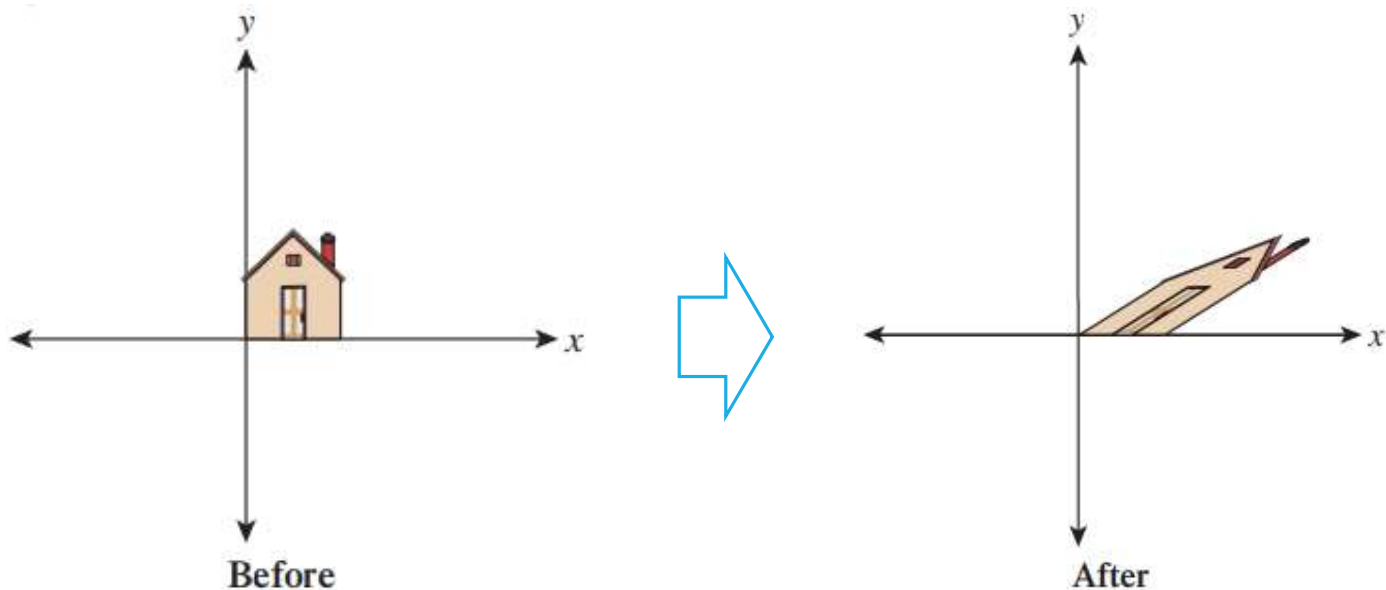
$$R(n) = \begin{bmatrix} 1 - 2n_x^2 & -2n_xn_y \\ -2n_xn_y & 1 - 2n_y^2 \end{bmatrix}$$



Shear - Skew

Transformación de corte

Permite deformar el elemento en 3D de manera no uniforme. No se preservan los ángulos, sin embargo, los volúmenes y áreas sí.



Shear - Skew

Transformación de corte

La idea básica es agregar un múltiplo de una coordenada a la otra. Por ejemplo, en 2D, se podría tomar un múltiplo de y para agregárselo a x , de tal forma que:

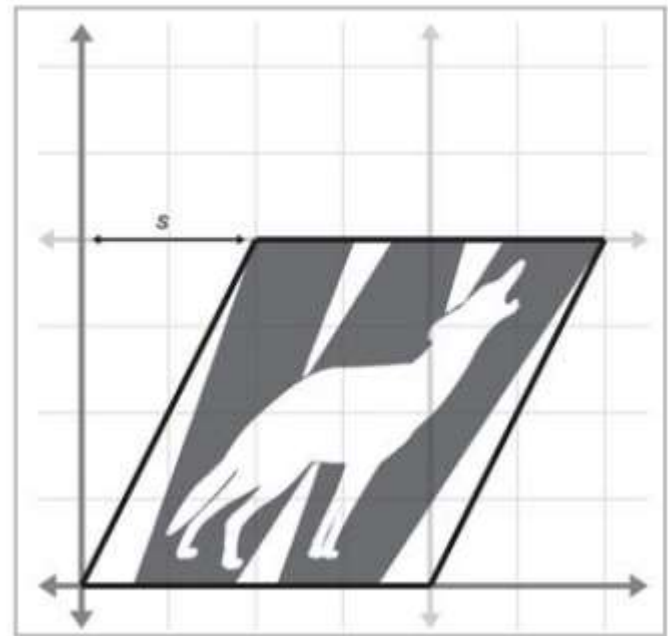
$$x' = x + sy$$

La matriz que permite hacer esto es:

$$H_x(s) = \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix}$$

En la coordenada y , la matriz sería:

$$H_y(s) = \begin{bmatrix} 1 & 0 \\ s & 1 \end{bmatrix}$$



Transformaciones en 3D

Traslación 3D

En un espacio 3D, dado el vector de traslación $[t_x \ t_y \ t_z]$, existe una matriz 4x4 que representa la traslación deseada:

$$T_{x,y,z} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiplicando la matriz por el punto $P(x, y, z)$ que se quiere trasladar:

$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{P} \quad \Rightarrow \quad \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{bmatrix}$$

Si el vector de traslación es $[0 \ 0 \ 0]$, el objeto se mantiene igual.

Traslación inversa en 3D

Si se desea reversar una traslación, es necesario realizar la traslación inversa:

$$\mathbf{P} = \mathbf{T}^{-1} \cdot \mathbf{P}'$$

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{bmatrix}$$

De esta manera es posible regresar al punto original.

Traslación en Three.js

```
var t = new THREE.Matrix4();  
t.set( 1, 0, 0, tx,  
      0, 1, 0, ty,  
      0, 0, 1, tz,  
      0, 0, 0, 1 );  
object.applyMatrix(t);  
object.elementsNeedUpdate = true;
```

Traslación inversa en Three.js

```
positionMatrix = new THREE.Matrix4();  
positionMatrix.copyPosition( object.matrix );  
inverseMatrix = new THREE.Matrix4();  
inverseMatrix.getInverse( positionMatrix );  
object.applyMatrix( inverseMatrix );
```

Escala 3D

En un espacio 3D, si se desea aplicar un cambio de escala $[s_x \quad s_y \quad s_z]$, esto puede ser representado por la matriz 4x4 :

$$S_{a,b,c} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiplicando la matriz por el punto (x, y, z) que se quiere trasladar:

$$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \\ s_z z \\ 1 \end{bmatrix}$$

Escala inversa 3D

Si se desea reversar un cambio en escala:

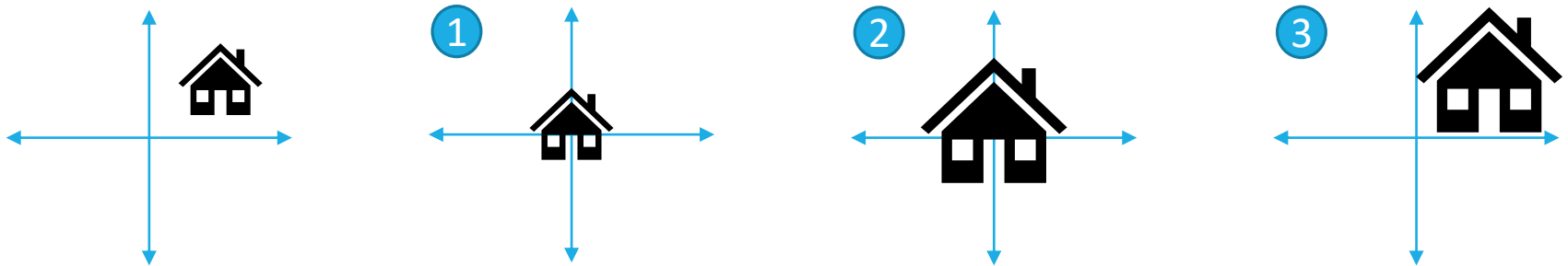
$$\mathbf{P} = \mathbf{S}^{-1} \cdot \mathbf{P}'$$

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1/s_x & 0 & 0 & 0 \\ 0 & 1/s_y & 0 & 0 \\ 0 & 0 & 1/s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} x'/s_x \\ y'/s_y \\ z'/s_z \\ 1 \end{bmatrix}$$

Escala 3D, *Con respecto al origen local*

Para lograr esto, es necesario aplicar las siguientes transformaciones:

1. Trasladar el objeto al origen.
2. Escalar.
3. Trasladar el objeto nuevamente a su posición original.



De lo contrario, el objeto se escalará con respecto al origen

Escala 3D

La matriz resultante será entonces:

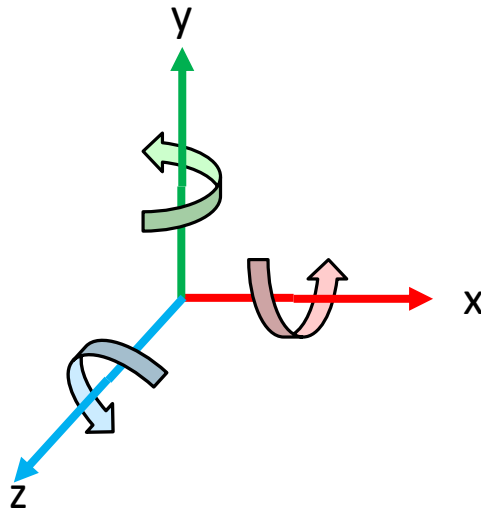
$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{T}^{-1} \cdot \mathbf{P}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & t_x(1 - s_x) \\ 0 & s_y & 0 & t_y(1 - s_y) \\ 0 & 0 & s_z & t_z(1 - s_z) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

De lo contrario, el objeto se escalará con respecto al origen

Rotaciones 3D

En 3D, la rotación se hace alrededor de un vector, tomando el ángulo positivo como aquel que genera una rotación en el sentido contrario a las manecillas del reloj (regla de la mano derecha).



Rotaciones 3D, *Sobre los ejes cartesianos*

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

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

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

Rotación 3D, *Sobre un eje paralelo al eje de coord.*

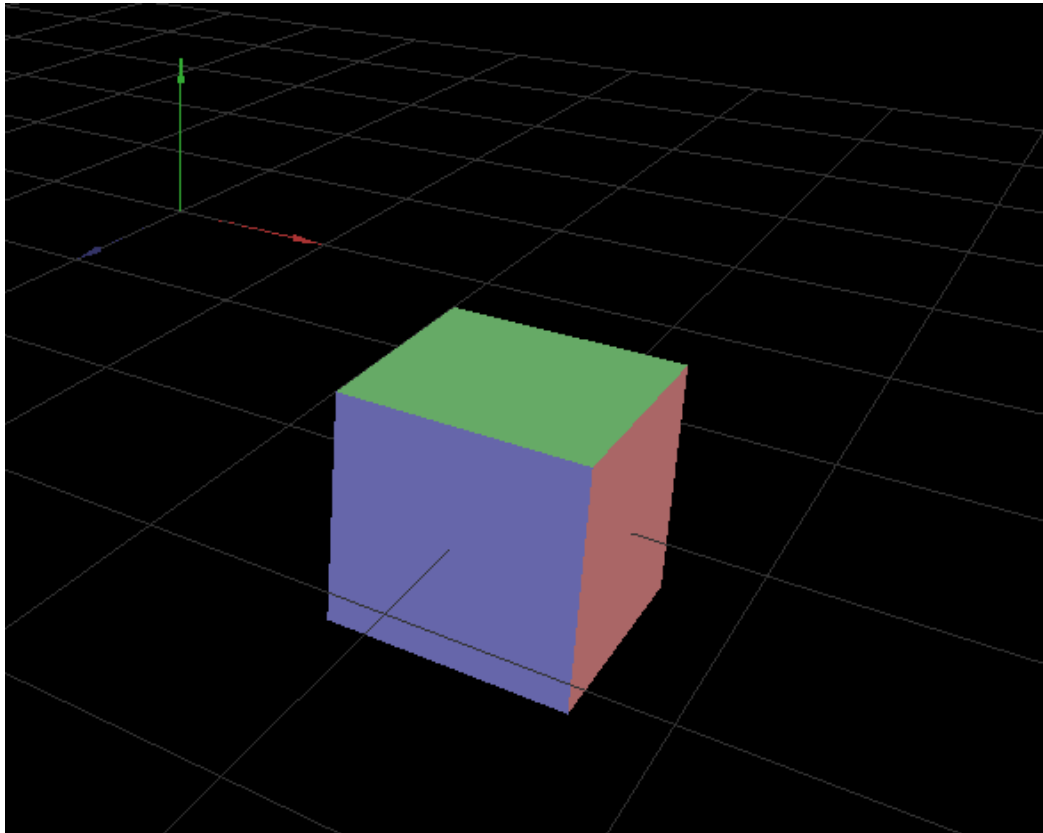
Es necesario aplicar las siguientes transformaciones:

1. Trasladar el objeto, para que los dos ejes de coordenadas coincidan.
2. Rotar sobre el eje de coordenadas.
3. Trasladar el objeto nuevamente a su posición original.

$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{R}(\theta) \cdot \mathbf{T}^{-1} \cdot \mathbf{P}$$

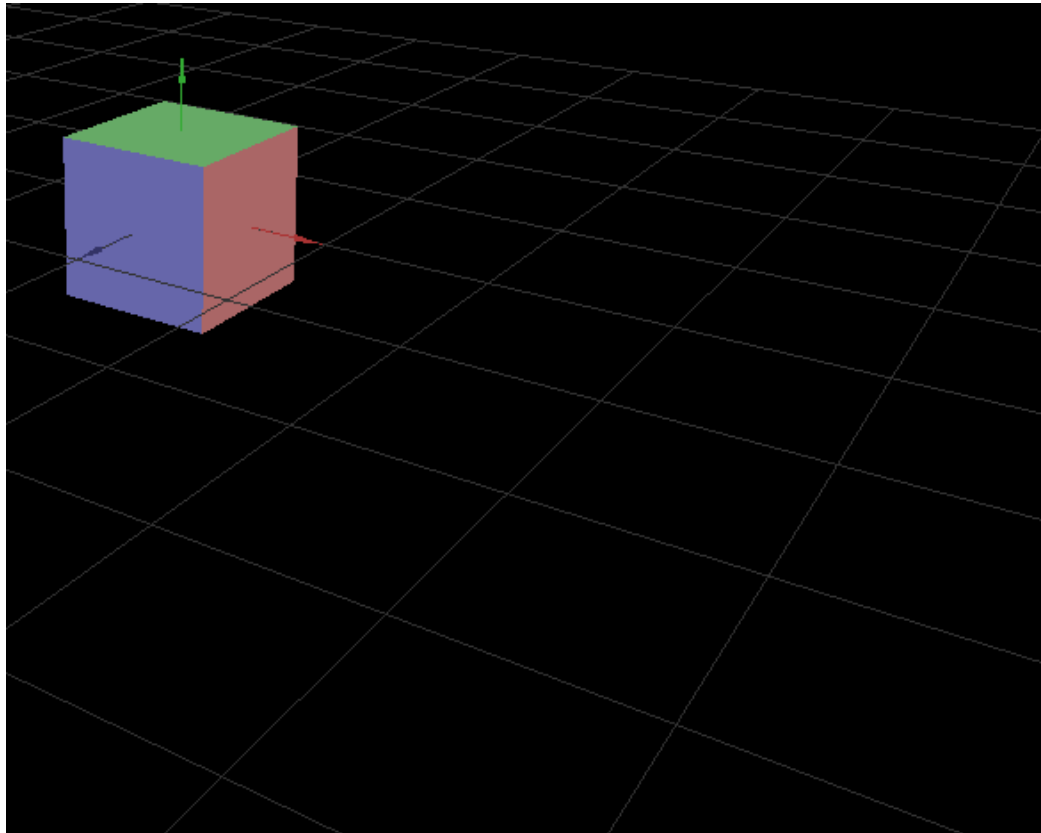
Rotación 3D, *Sobre un eje paralelo al eje de coord.*

Posición original del objeto:



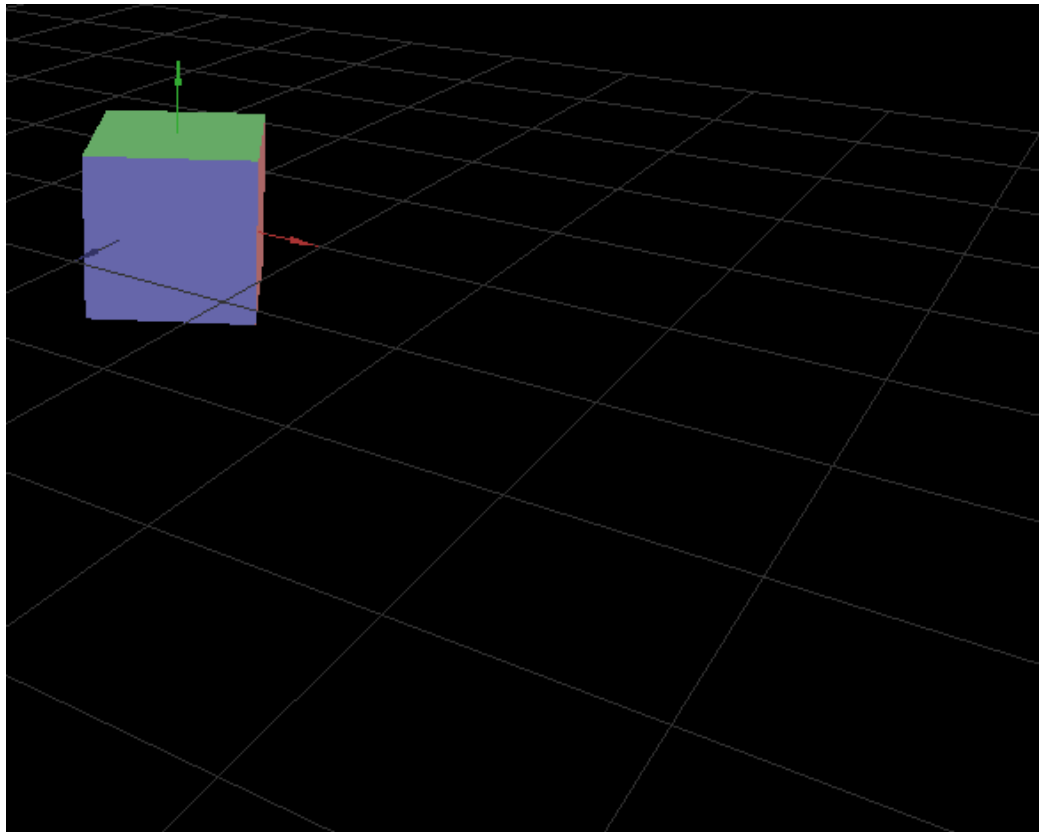
Rotación 3D, *Sobre un eje paralelo al eje de coord.*

Traslación al origen:



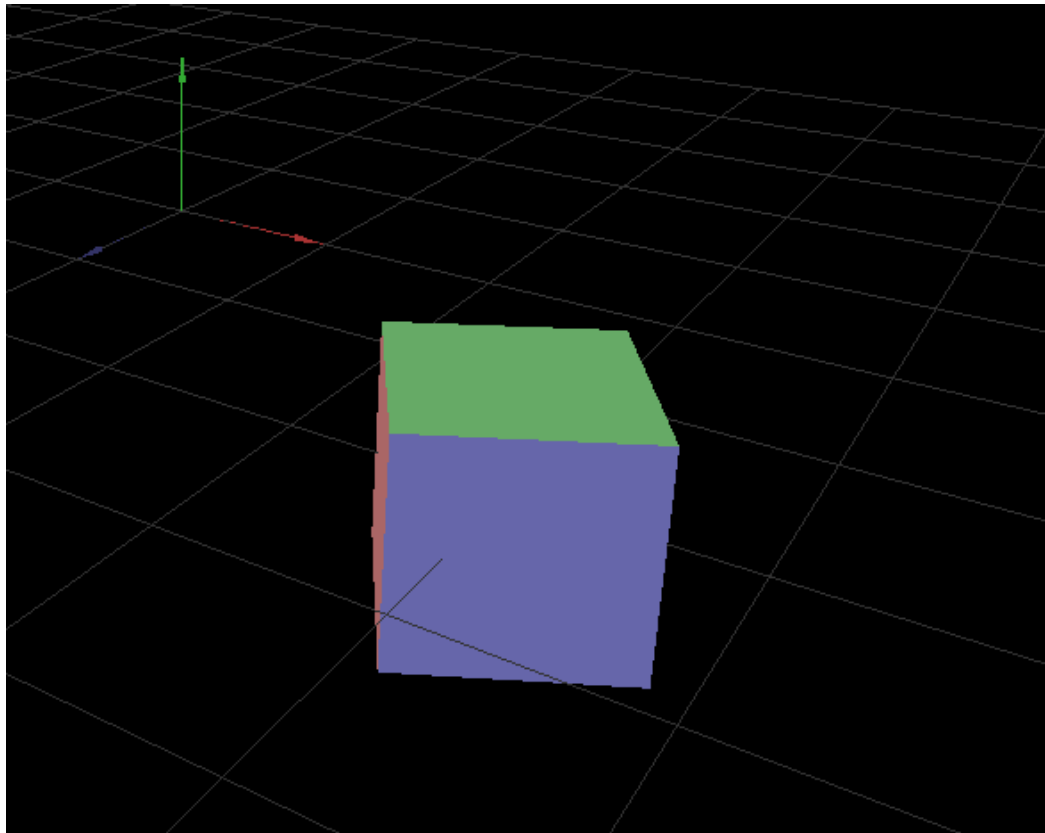
Rotación 3D, *Sobre un eje paralelo al eje de coord.*

Rotación de 30° en el origen, con respecto al eje y:



Rotación 3D, *Sobre un eje paralelo al eje de coord.*

Traslación a la posición original:



Rotación 3D, *Sobre un eje arbitrario.*

1. Trasladar el objeto para que el eje de rotación pase a través del origen de coordenadas.
2. Rotar el objeto para que el eje de rotación se alinee con un eje de coordenadas.
3. Rotar sobre el eje de coordenadas.
4. Aplicar la rotación inversa para devolver el eje de rotación a su orientación original.
5. Trasladar el objeto para que el eje de rotación retorne a su posición original.

$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{R}_z(\theta) \cdot \mathbf{R}_y(\varphi) \cdot \mathbf{R}_x(\alpha) \cdot \mathbf{R}_y^{-1}(\varphi) \cdot \mathbf{R}_z^{-1}(\theta) \cdot \mathbf{T}^{-1} \cdot \mathbf{P}$$

Acumulación de transformaciones

La combinación de transformaciones se realiza multiplicando las diferentes matrices, en un orden predefinido:

$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{R}(\theta) \cdot \mathbf{S} \cdot \mathbf{P}$$

Se debe tener en cuenta que primero se realiza el escalamiento, luego la rotación y finalmente la traslación (la multiplicación de matrices se realiza de derecha a izquierda).

Hacerlo de otra forma dará resultados diferentes.

Transformaciones en Three.js

Es posible utilizar funciones preestablecidas, para realizar cambios a un Objeto3D:

```
object.position.z = 5;  
object.position.copy(start_position);  
object.translateX(distance);  
object.rotateX(angle_in_radians);  
object.rotateOnAxis(axis, angle_in_radians);  
object.rotateOnWorldAxis(axis, angle_in_radians);  
object.quaternion.copy(quaternion);
```

Si se desea controlar la actualización de la matriz se puede aplicar:

```
object.matrixAutoUpdate = false;  
object.updateMatrix();
```

Transformaciones en Three.js

También es posible aplicar directamente una transformación a la matriz de ese objeto:

```
object.matrix.setPosition(start_position);  
object.matrixAutoUpdate = false;
```

En este caso, no se usa `updateMatrix()` pues se perderían los cambios.

Bibliografía

Tutorial 3: Matrices. <http://www.opengl-tutorial.org/beginners-tutorials/tutorial-3-matrices/>

The Matrix and Quaternions FAQ. (2002). http://www.opengl-tutorial.org/assets/faq_quaternions/index.html

Rueda, A. (2015). *Transformaciones en 3D*. Universidad Javeriana.