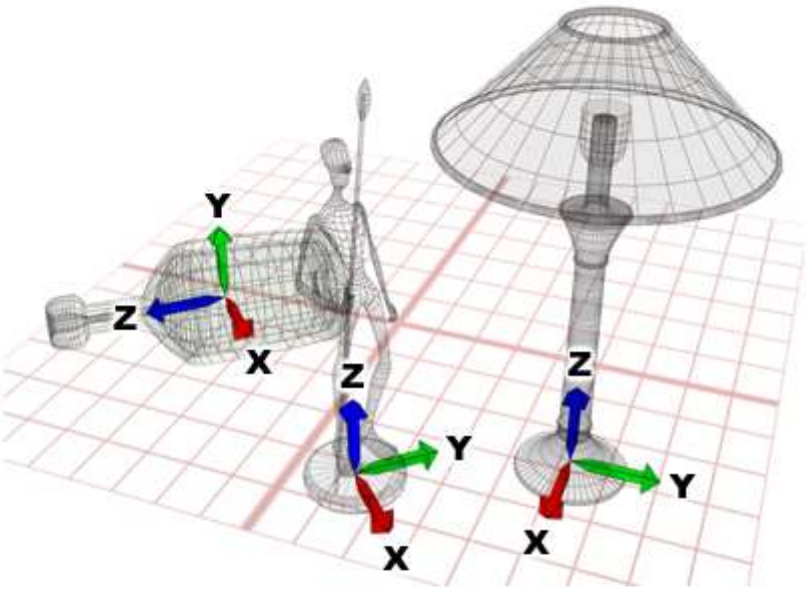
Computación Gráfica

Ing. Gabriel Ávila, MSc.

Múltiples espacios de coordenadas

¿Para qué?

Aunque todos los puntos en un espacio 3D se pueden localizar mediante un único sistema de referencia, ***algunas coordenadas están relacionadas con un marco de referencia particular***.

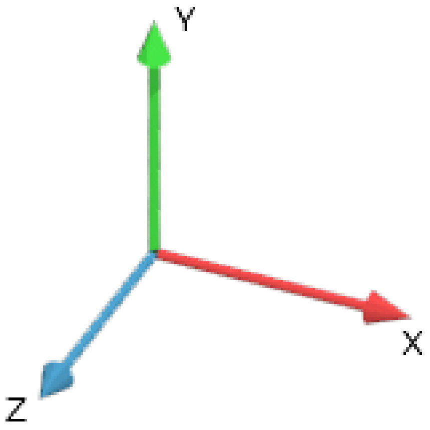


World space
Espacio de coordenadas global o universal

Representa posiciones **absolutas** en el mundo, con respecto a un origen.

Establece un marco de referencia global, mediante el cual otros sistemas de referencia pueden especificarse.

Permite ubicar objetos en el ambiente.

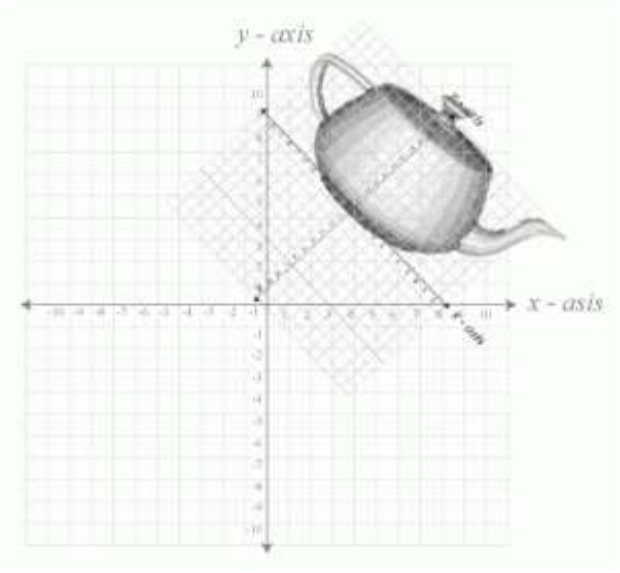


Object space
Espacio de coordenadas del objeto

Cada objeto tiene su propio espacio de coordenadas. Al trasladarse o cambiar su orientación, este espacio de coordenadas se mueve con el objeto.

En este espacio, es posible preguntarse sobre qué hay arriba, al frente o a los lados de un objeto particular.

En algunos contextos se conoce como ***espacio de modelado*** o ***"body space"***

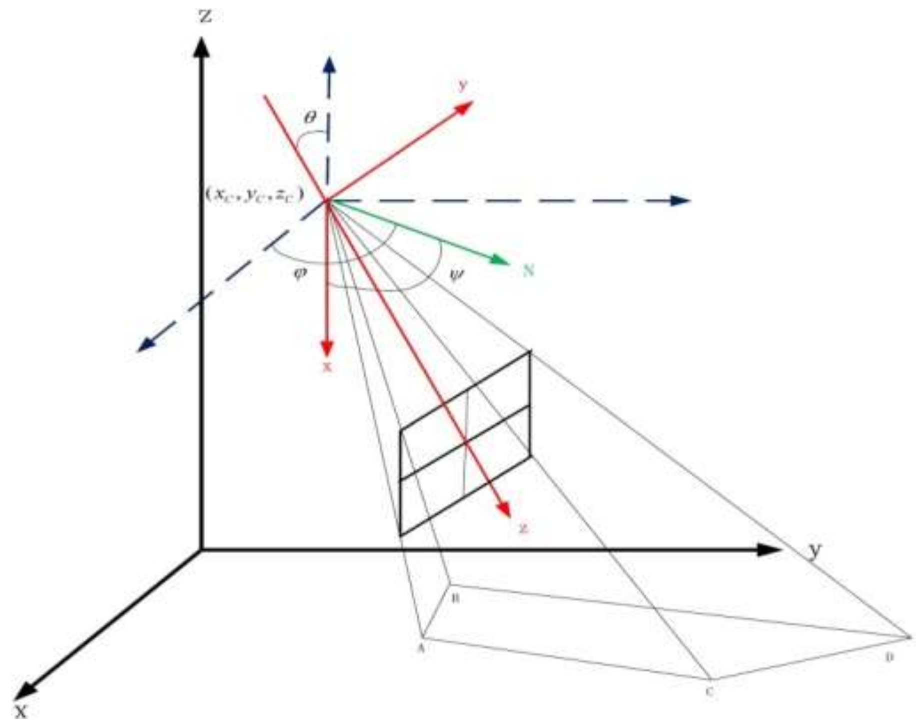


Camera space
Espacio de coordenadas de la cámara

Se trata de un espacio de coordenadas en 3D, asociado con el observador.

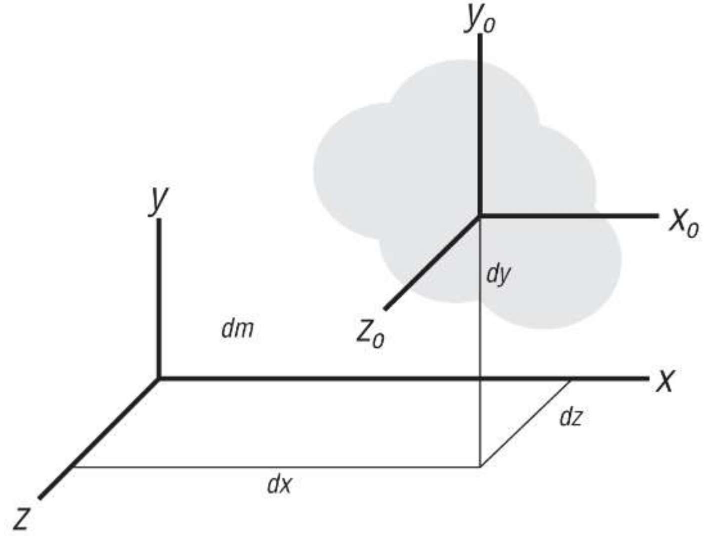
El objeto que define este espacio es la cámara, que a su vez define el punto de vista de la escena.

Permite evaluar si un objeto es visto o no por la cámara, así como la distancia de los objetos hacia ésta.



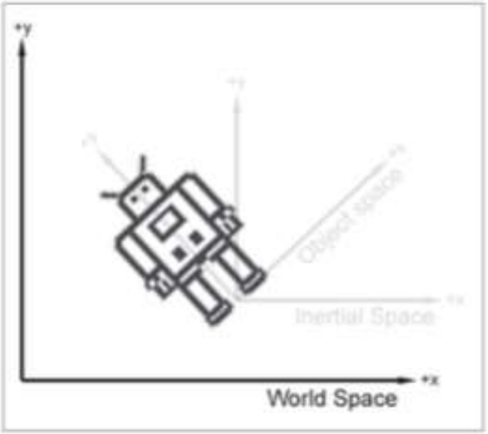
Inertial space
Espacio de coordenadas inercial

Es un espacio intermedio, entre el espacio global y el del objeto. Su origen es el mismo del espacio de objeto, pero los ejes se mantienen paralelos a los del espacio global.

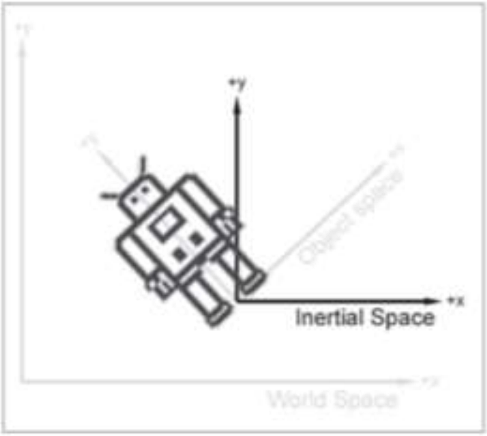


Transformación entre espacios de coordenadas

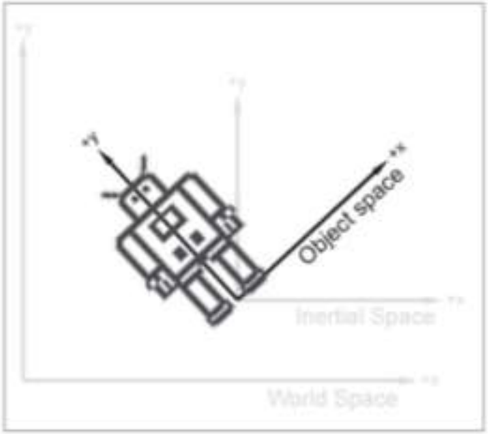
Espacio global



Espacio inercial



Espacio local

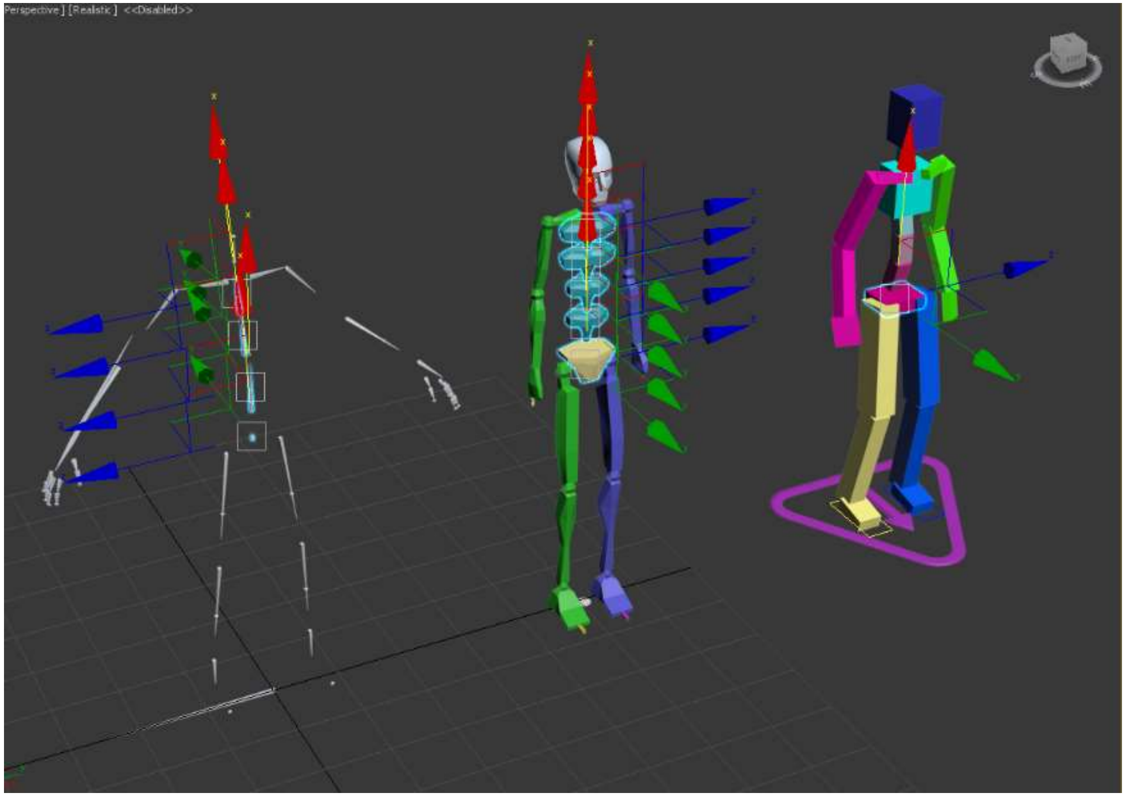


Traslación

Rotación

En ocasiones, es necesario realizar transformaciones entre espacio de objeto y espacio global. En estos casos se utilizan el ***espacio inercial*** como intermediario.

Espacios de coordenadas anidados (jerarquías)



Transformaciones

Transformaciones

Hay que preguntarse ¿Qué se está transformando? En un sistema 3D, es posible transformar diferentes elementos:

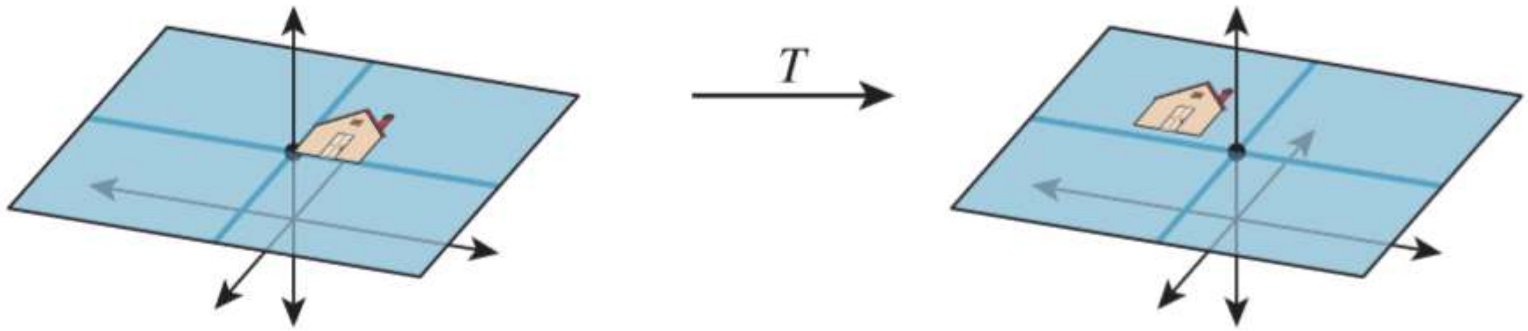
- Vértices.
- Geometrías.
- Objetos completos.
- La cámara.
- El mundo (El sistema de coordenadas).
- etc.

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.

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

t_x : traslación en x
t_y : traslación en y



Traslación 3D

En un espacio 3D, dado el vector de traslación $[t_x \quad t_y \quad t_z]$, 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:

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

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

Traslación inversa en 3D

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

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

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

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 );
```

Ejercicio

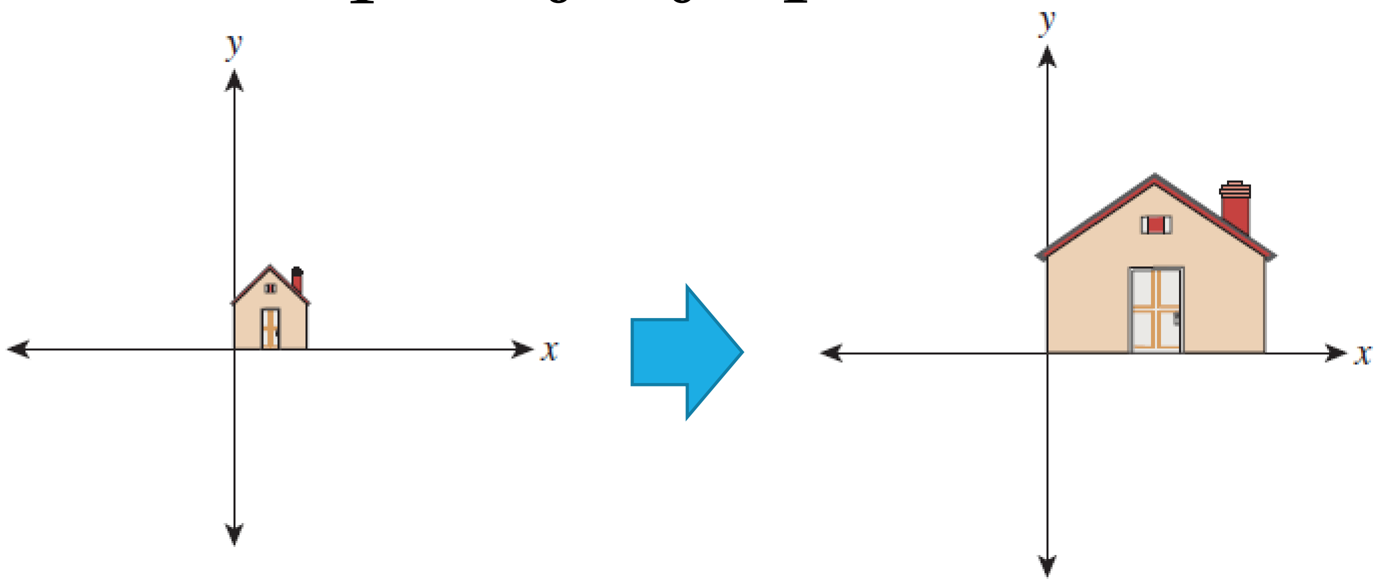
Descargar el archivo ***03-Traslación.***

Modificarlo de tal forma que permita obtener traslaciones del cubo, en cualquier dirección x, y, z.

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) -> (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 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:

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

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

Escala inversa 3D

Si se desea reversar un cambio en escala:

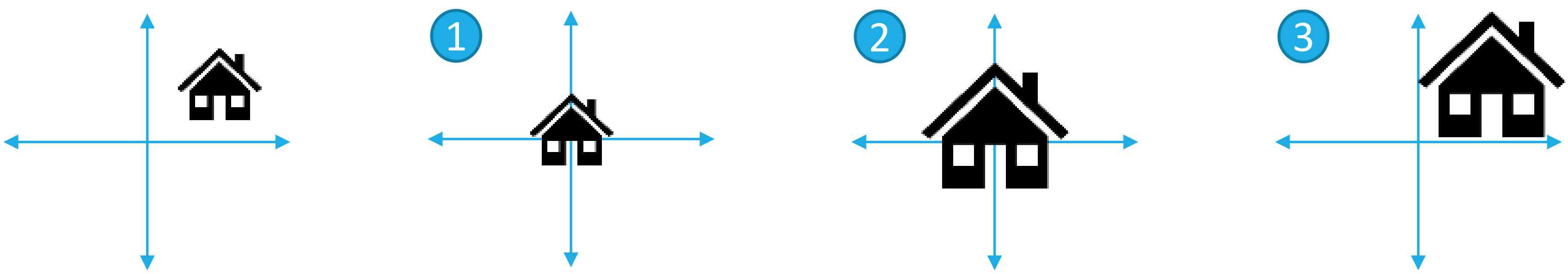
$$\boldsymbol{P} = \boldsymbol{S}^{-1} \cdot \boldsymbol{P}'$$

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

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:

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

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

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

Ejercicio

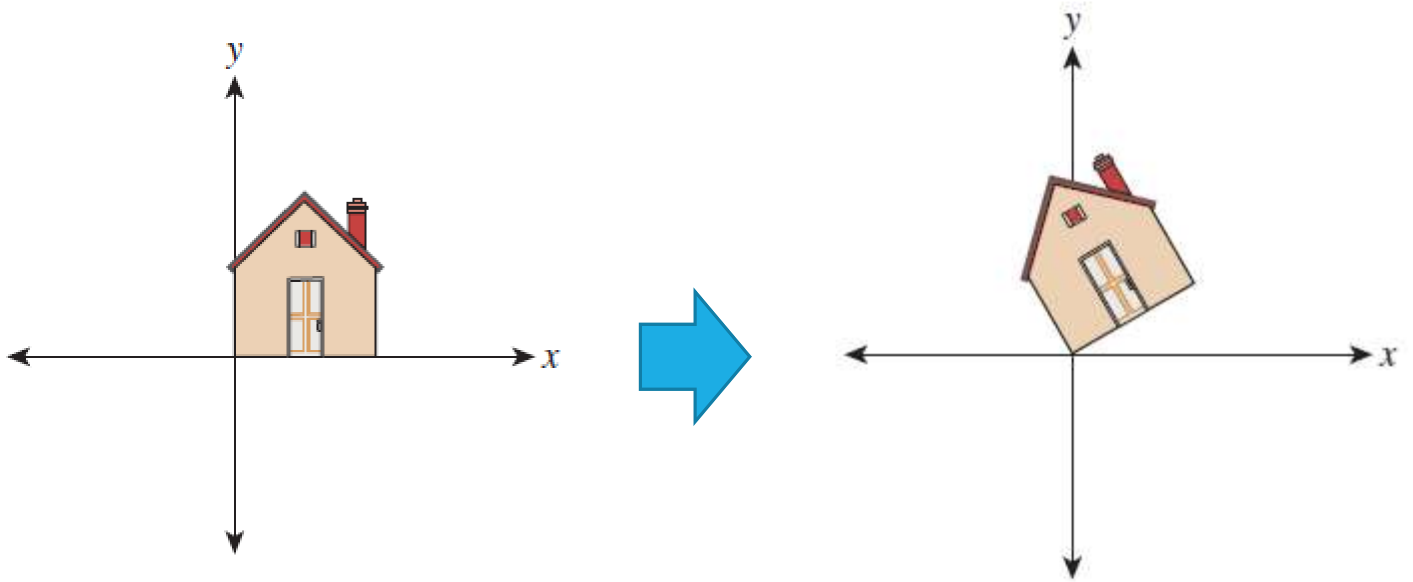
Aplicar la matriz de escala en el archivo descargado, de dos maneras:

- Escalando el objeto con respecto al origen.

- Trasladando el objeto al origen, escalando y regresando a su posición original.

Rotaciones

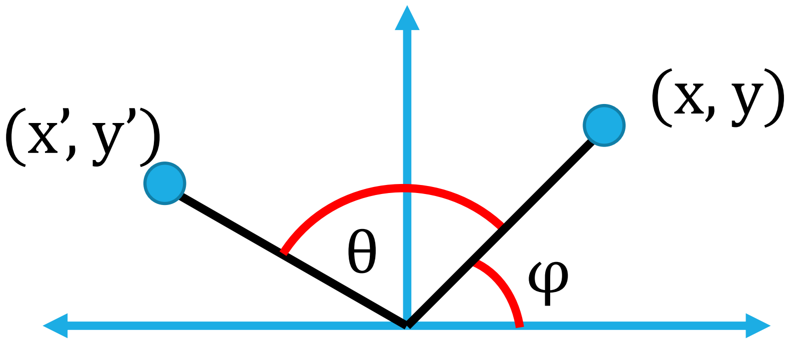
Una rotación implica mover un punto un cierto ángulo θ, formando un arco con respecto a un punto (2D) o a un vector (3D).



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:

$$x' = r\cos(\theta)\cos(\varphi) - r\sin(\theta)\sin(\varphi)$$
$$\boldsymbol{x' = x\cos(\theta) - y\sin(\theta)}$$
$$y' = r\sin(\theta)\cos(\varphi) + r\cos(\theta)\sin(\varphi)$$
$$\boldsymbol{y' = x\sin(\theta) + y\cos(\theta)}$$

Rotaciones 2D

A partir de las dos ecuaciones obtenidas:

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

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

Se puede reformular, utilizando matrices:

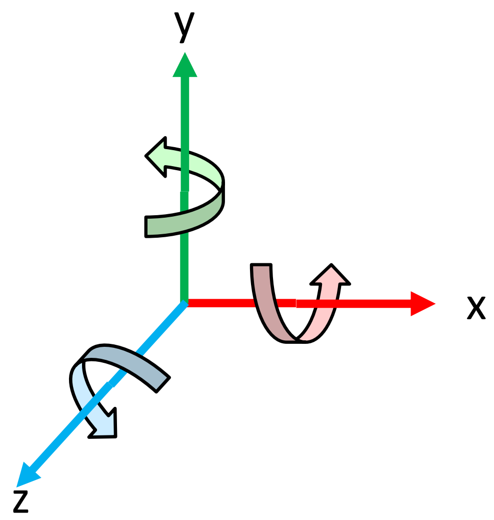
$$\begin{bmatrix} x' \\ y' \\ 1' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x\,cos(\theta) & -\,y\,sin(\theta) \\ x\sin(\theta) & +\,y\cos(\theta) \\ & 1 \end{bmatrix}$$

Obteniendo la matriz de rotación por un ángulo $\boldsymbol{\theta}$ en el origen:

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

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.*

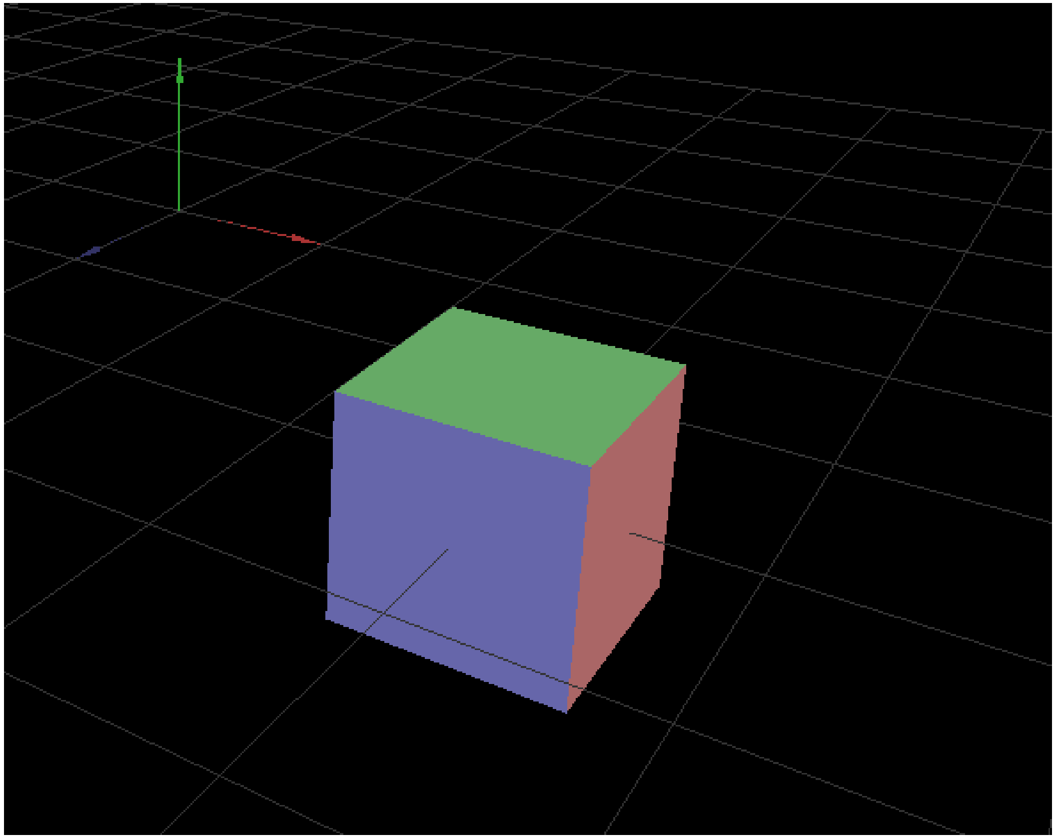
Es necesario aplicar las siguientes transformaciones:

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.

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

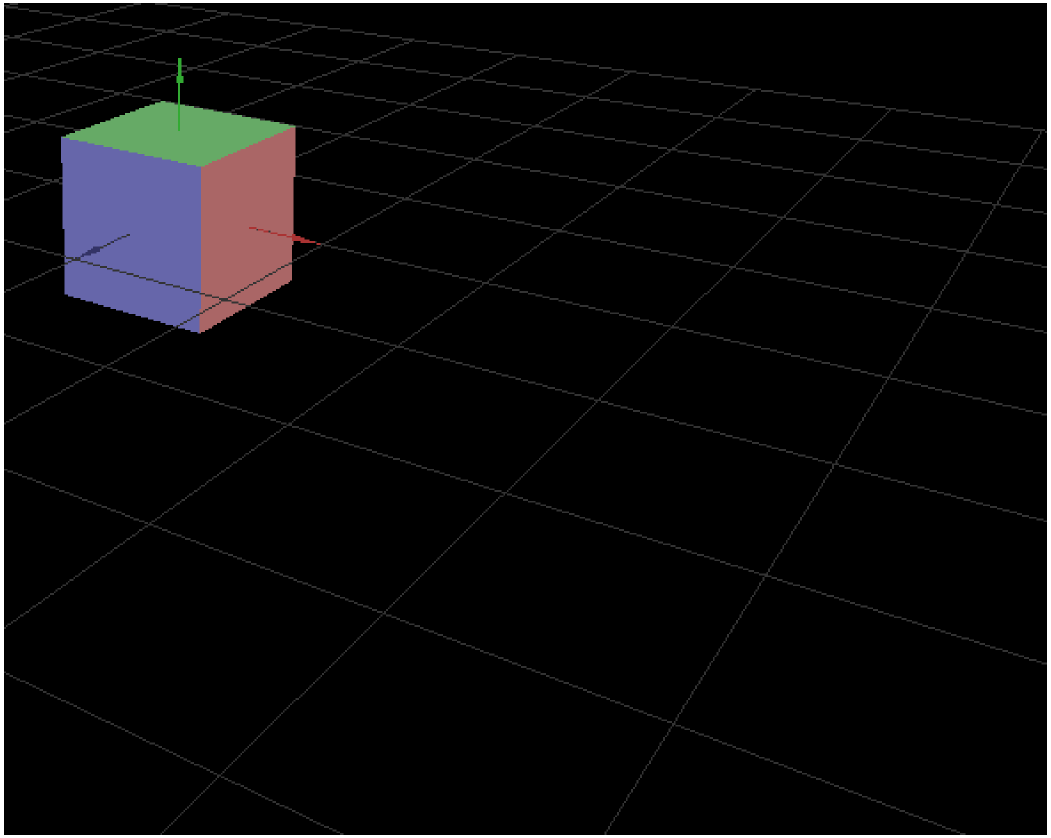
Rotación 3D,
Sobre un eje paralelo al eje de coord.

Posición original del objeto:



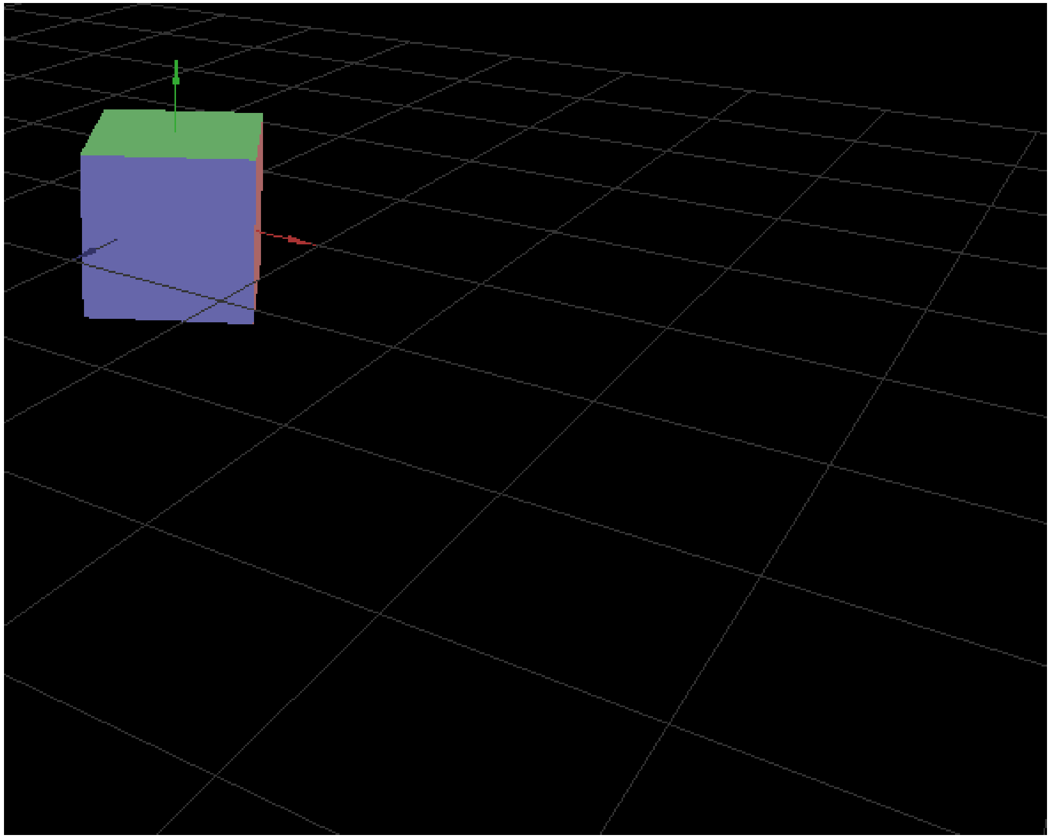
Rotación 3D,
Sobre un eje paralelo al eje de coord.

Traslación al origen:



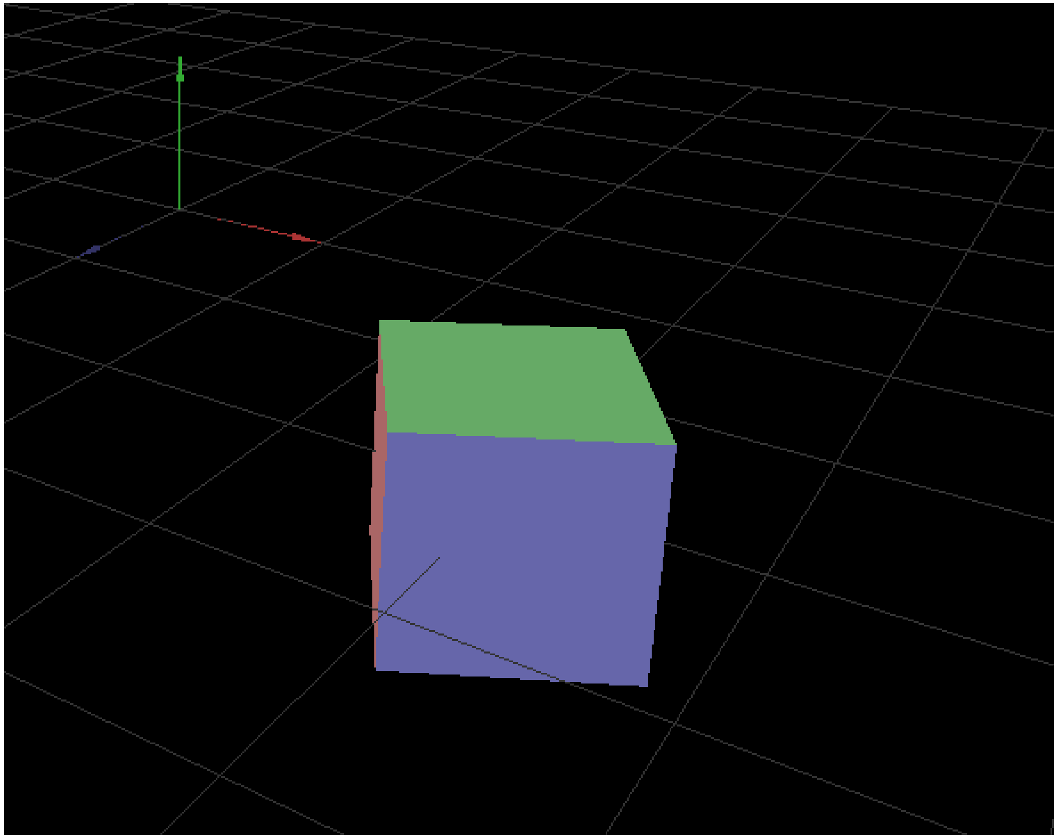
Rotación 3D,
Sobre un eje paralelo al eje de coord.

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.

$$\boldsymbol{P'} = \boldsymbol{T} \cdot \boldsymbol{R_z}(\theta) \cdot \boldsymbol{R_y}(\varphi) \cdot \boldsymbol{R_x}(\alpha) \cdot \boldsymbol{R_y^{-1}}(\varphi) \cdot \boldsymbol{R_z^{-1}}(\theta) \cdot \boldsymbol{T}^{-1} \cdot \boldsymbol{P}$$

Acumulación de transformaciones

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

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

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.