

Transformaciones 2D y 3D

Prof. Alexandra La Cruz

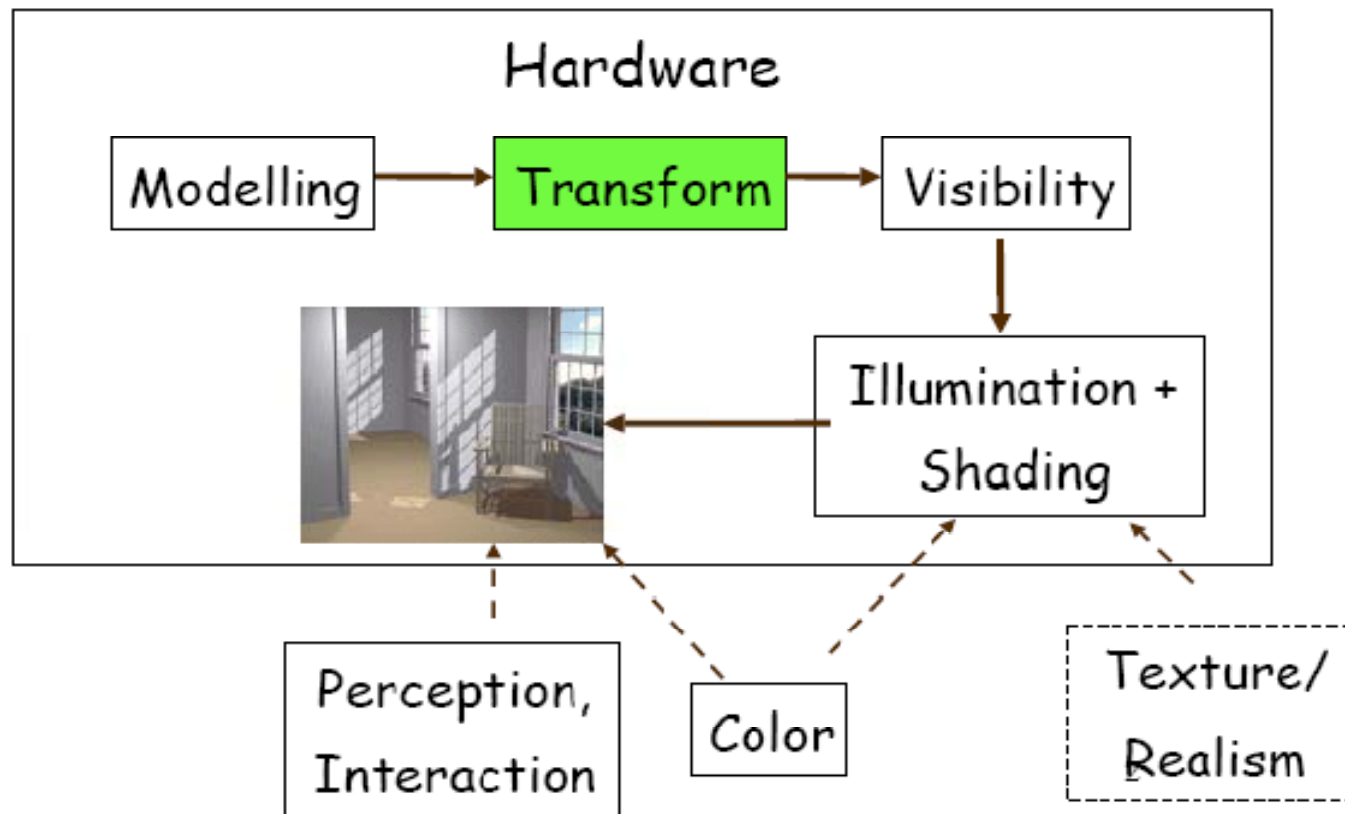
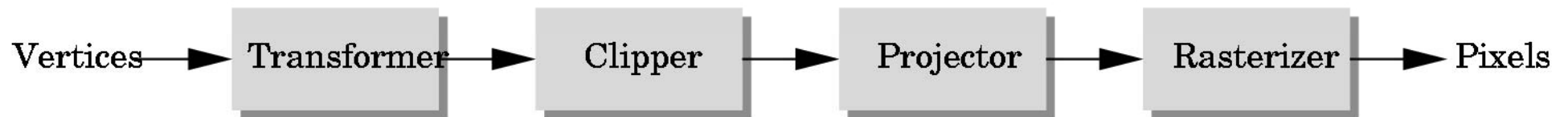
Fuentes utilizadas:

MIT

Prof. Torsten Mueller de la Universidad Simon Fraser

Prof. Eduardo Fernández de la Facultad de Ingeniería de la
Universidad de la República de Montevideo Uruguay

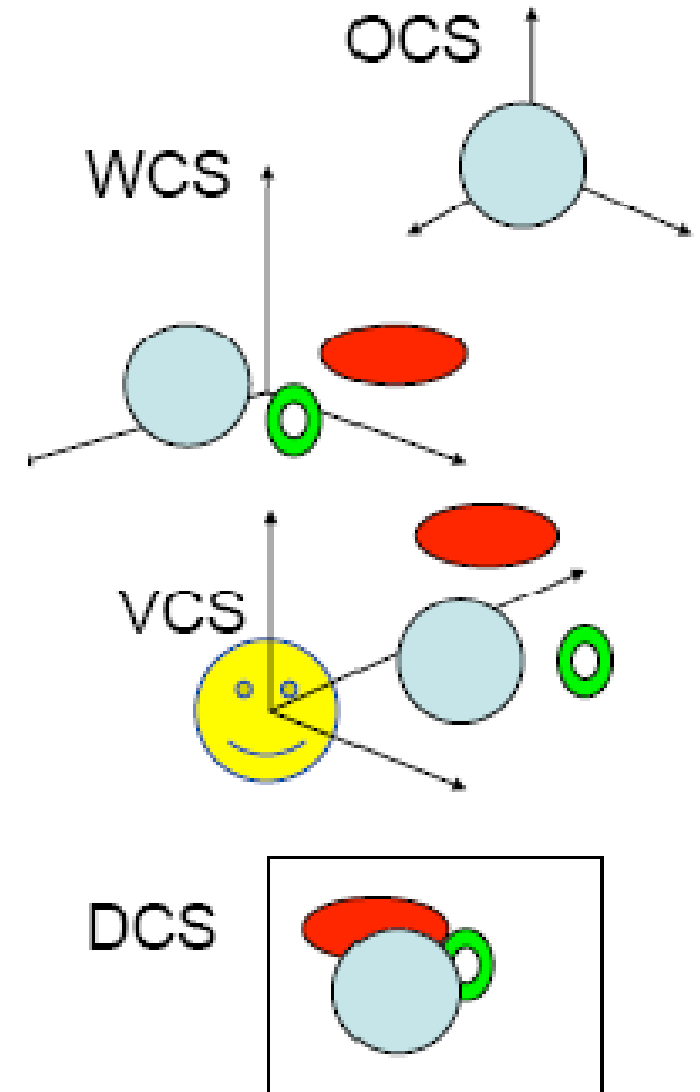
Pipeline



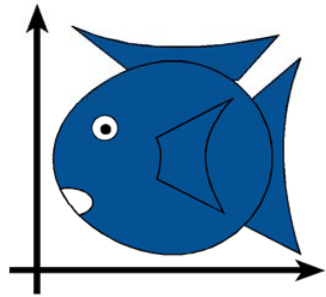
Transformaciones



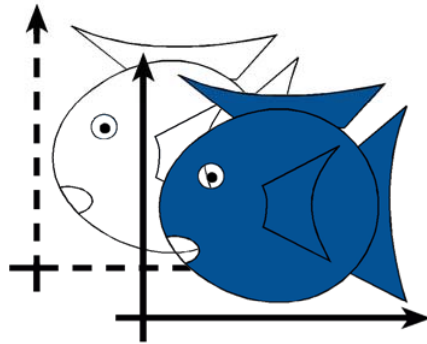
- ❑ Pasar del sistema de coordenadas del objeto al sistema de coordenadas de:
 - ❑ la escena
 - ❑ la cámara
 - ❑ la ventana de OpenGL (viewport)
- ❑ Para rotar, escalar o trasladar los objetos en la escena



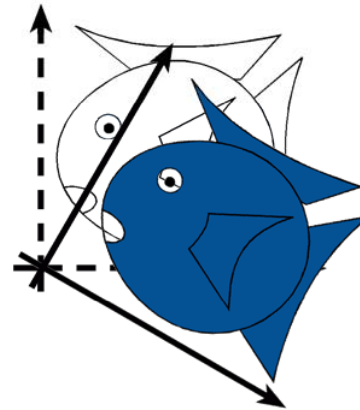
Transformaciones básicas



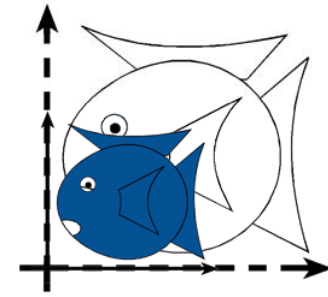
Identity



Translation



Rotation



Isotropic
(Uniform)
Scaling

- ❑ transformaciones de perspectivas o proyección
- ❑ Tipos de transformaciones
 - ❑ **Cuerpo rígido:** preservan la longitud y el ángulo
 - ❑ **Lineales:** preservan las líneas paralelas más no el ángulo o longitud
 - ❑ **Libre**

Uso de las transformaciones

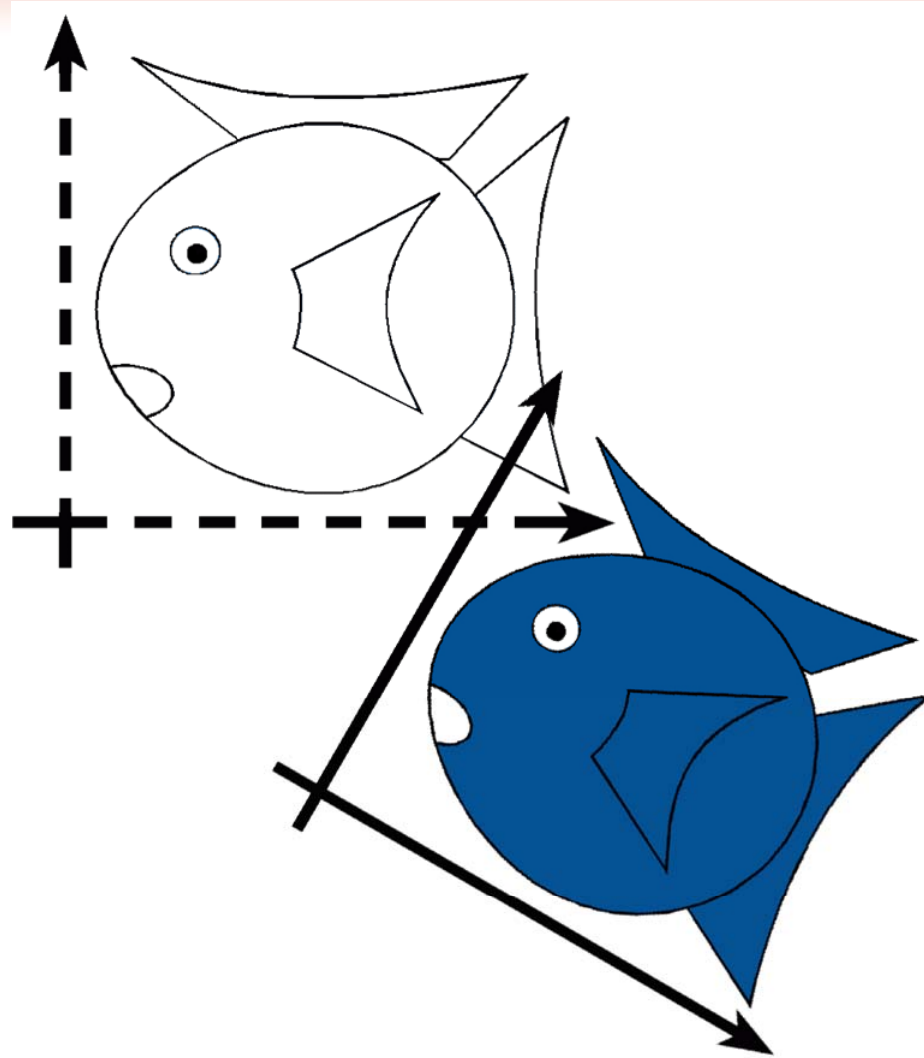


- Posicionar los objetos en la escena
- Cambiar la forma de los objetos
- Crear varias copias de los objetos
- Definir la proyección de la cámara virtual
- Animaciones

Transformaciones de cuerpo rígido



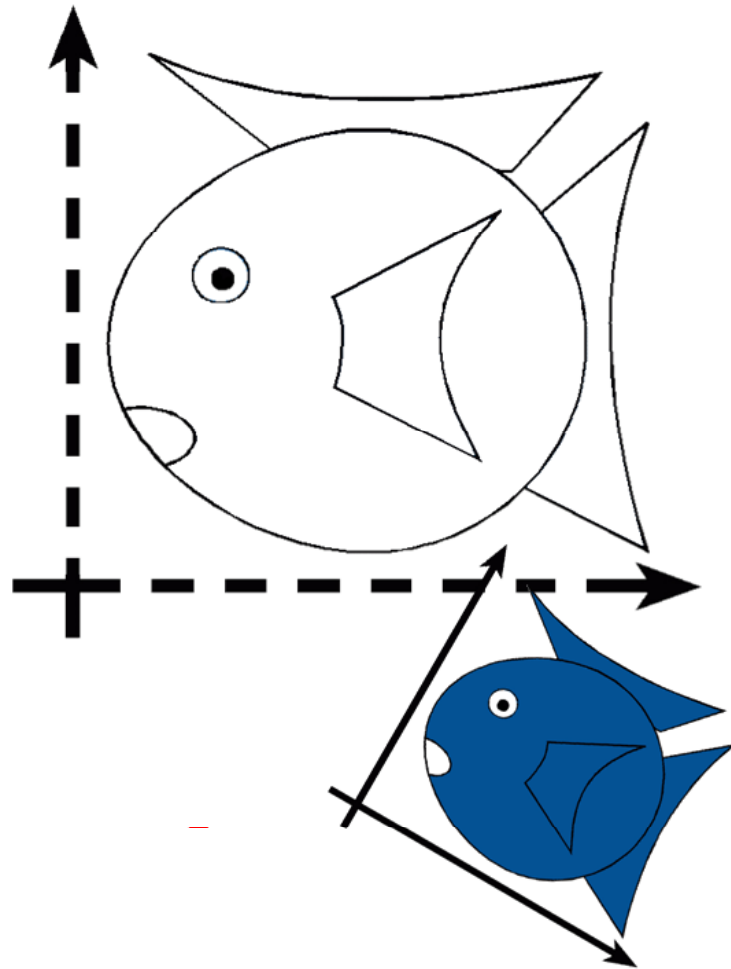
- Preservan
 - Las distancias
 - Los ángulos



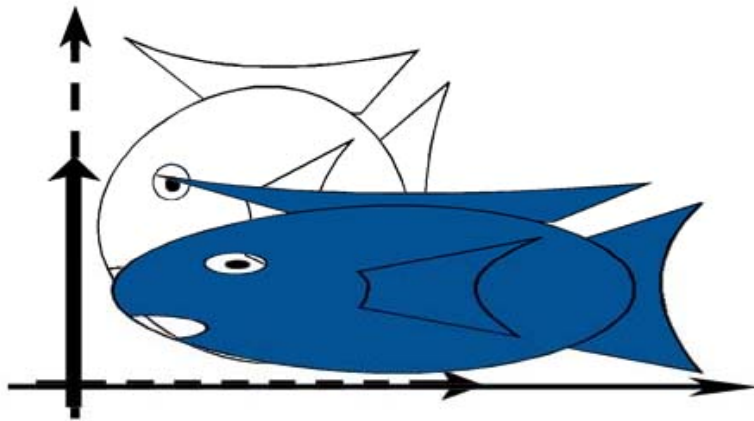
Transformaciones de cuerpo rígido



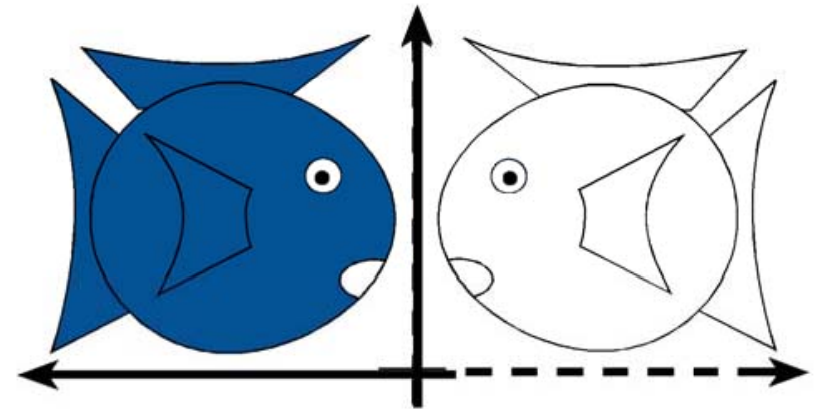
- Preserva los ángulos



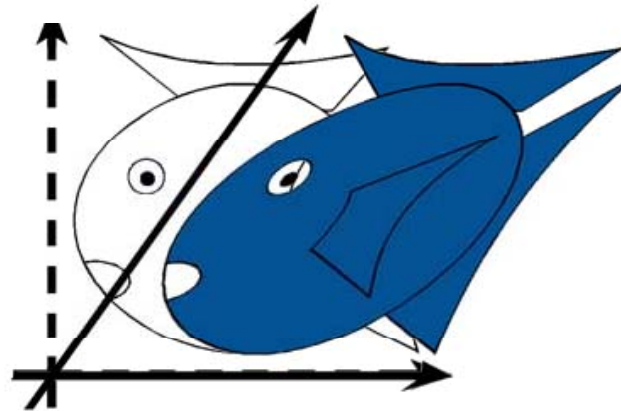
Transformaciones lineales



Scaling



Reflection

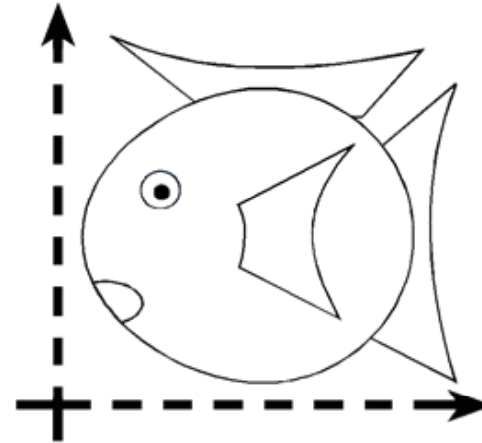
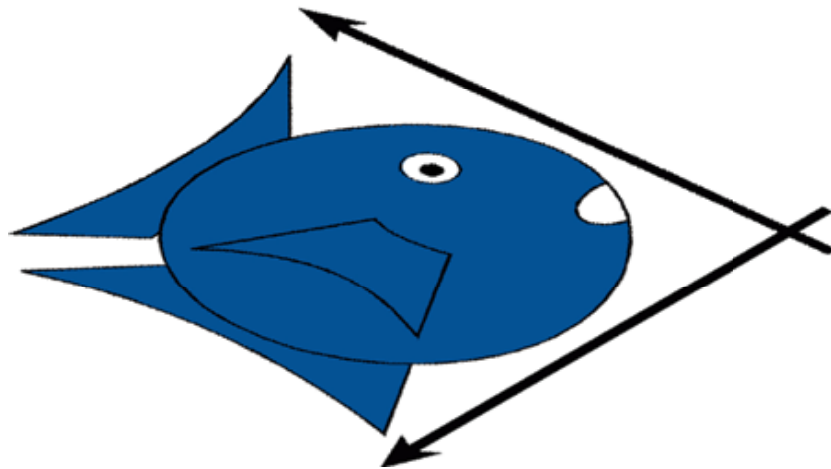


Shear

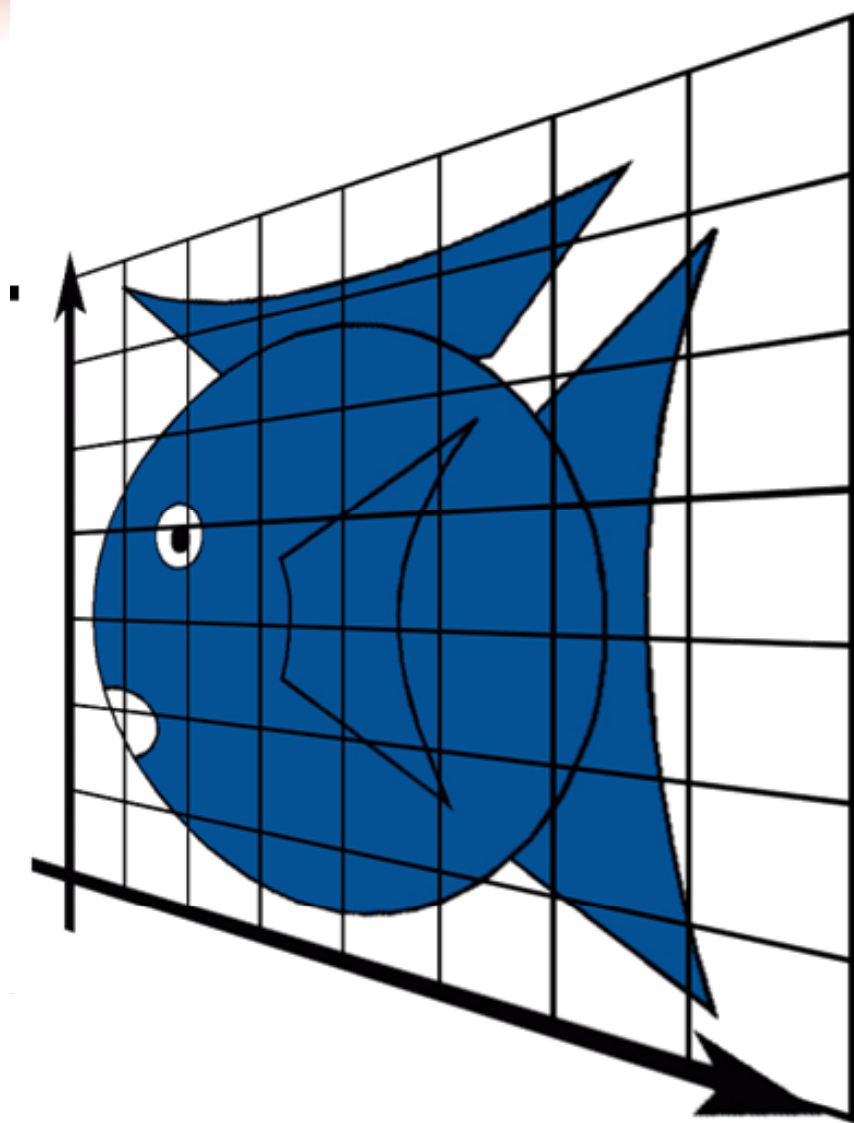
Transformaciones afín



- Preserva las líneas paralelas



Transformaciones de perspectiva



Transformaciones libres



- No se preservan las líneas



Fig 1. Undeformed Plastic

Fig 2. Deformed Plastic

From Sederberg and Parry, Siggraph 1986

Matriz de Transformación



$$x' = ax + by + c$$

$$y' = dx + ey + f$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ d & e \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} c \\ f \end{pmatrix}$$

$$p' = Mp + t$$

Coordenadas homogeneas



□ Se agrega una dimensión extra

□ en 2D, se usan 3 x 3 matrices

□ en 3D, se usan 4 x 4 matrices

$$p' = M p$$

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$$

Traslación



$$x' = ax + by + c$$

$$y' = dx + ey + f$$

Formulación afín

Formulación homogénea

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ d & e \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} c \\ f \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$p' = Mp + t$$

$$p' = Mp$$

Coordenadas homogeneas



□ $w = 1$

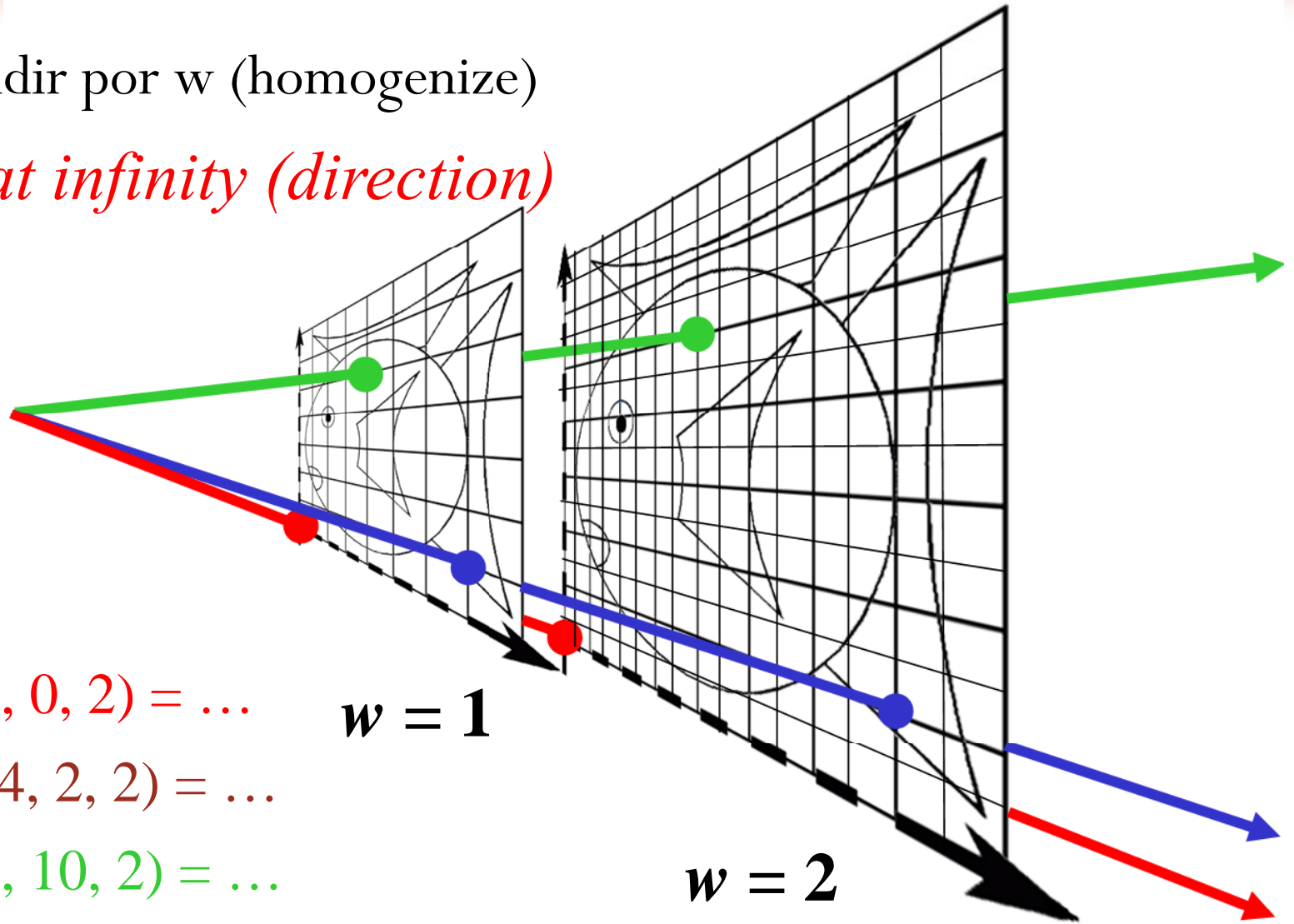
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Visualización homogénea



- Al dividir por w (homogenize)

Point at infinity (direction)



$$(0, 0, 1) = (0, 0, 2) = \dots$$

$w = 1$

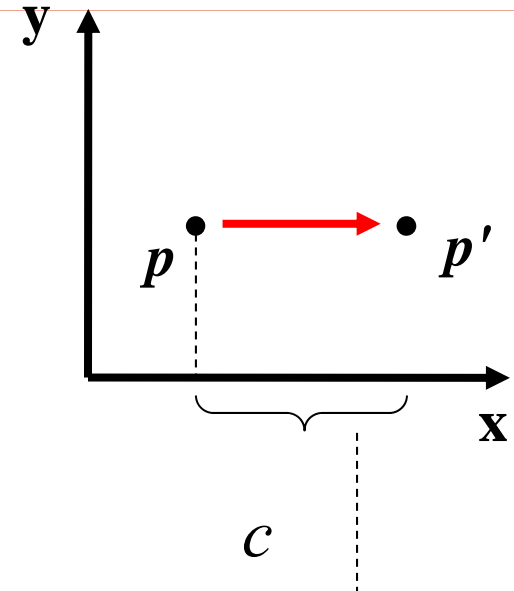
$$(7, 1, 1) = (14, 2, 2) = \dots$$

$$(4, 5, 1) = (8, 10, 2) = \dots$$

$w = 2$

Trasladar (t_x, t_y, t_z)

Translate($c, 0, 0$)

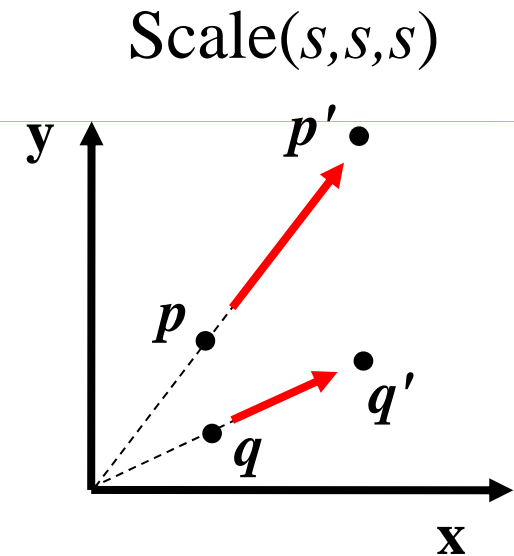


$$\begin{pmatrix} x' \\ y' \\ z' \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Escalar (s_x, s_y, s_z)

□ Uniforme

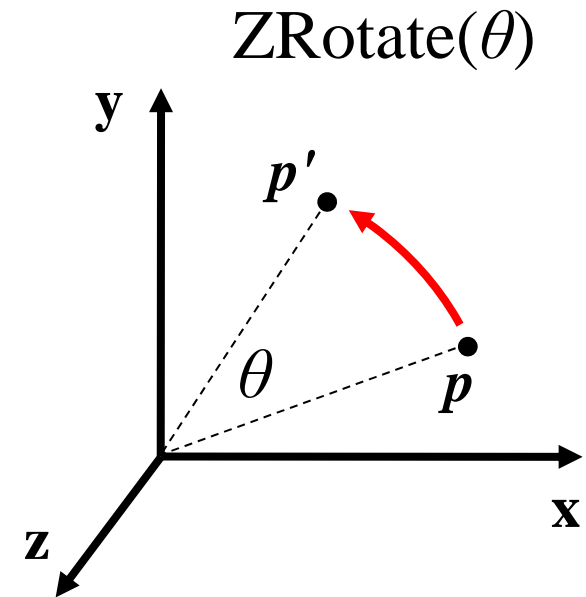
scaling: $s_x = s_y = s_z$



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

Rotation

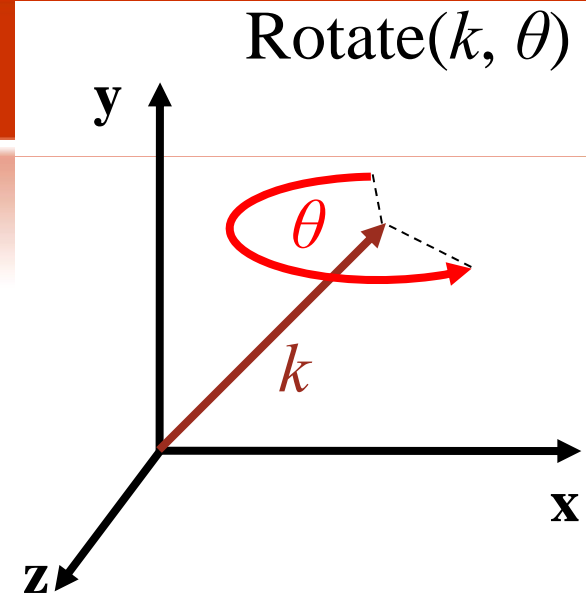
□ Sobre el eje z



$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Rotación

- sobre cualquier eje dado por el vector unitario (k_x, k_y, k_z)

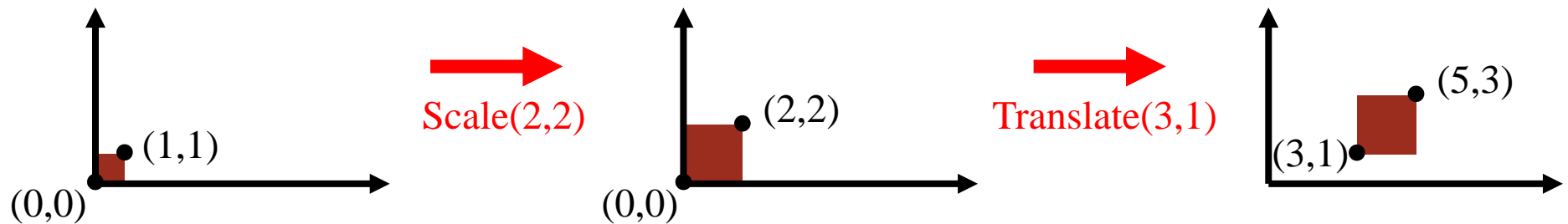


$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} k_x k_x (1-c) + c & k_z k_x (1-c) - k_z s & k_x k_z (1-c) + k_y s & 0 \\ k_y k_x (1-c) + k_z s & k_z k_x (1-c) + c & k_y k_z (1-c) - k_x s & 0 \\ k_z k_x (1-c) - k_y s & k_z k_x (1-c) - k_x s & k_z k_z (1-c) + c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

where $c = \cos \theta$ & $s = \sin \theta$



Escalar y luego trasladar



Multiplicación de matrices: $p' = T (S p) = TS p$

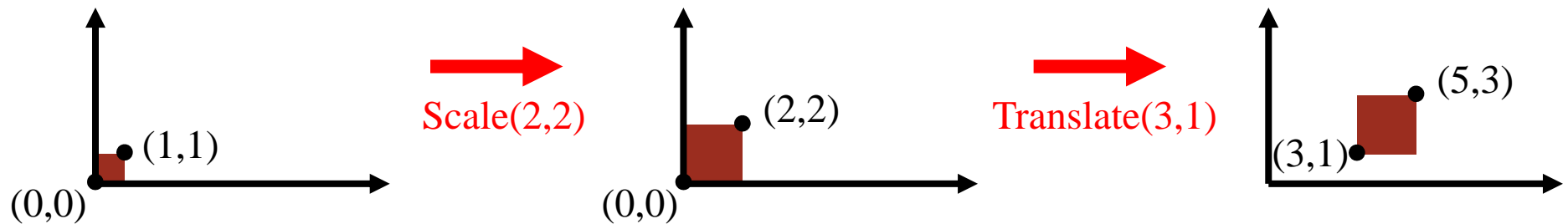
$$TS = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Atención: La multiplicación de matrices no es conmutativa

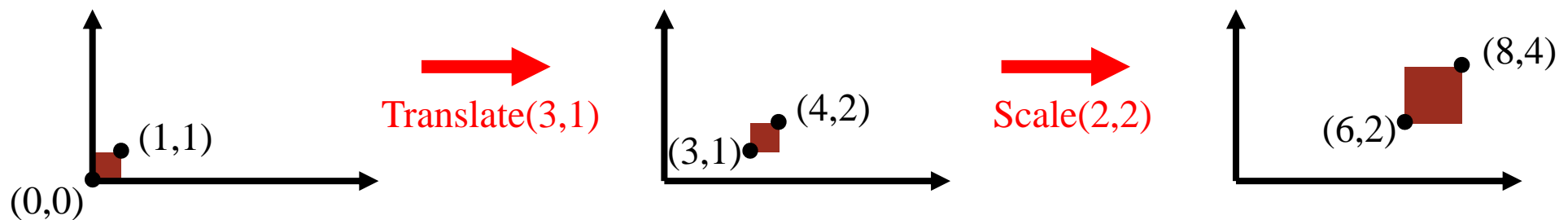
Composición no conmutativa



Escalar luego trasladar: $p' = T (S p) = TS p$



Trasladar luego escalar: $p' = S (T p) = ST p$



Matricialmente



Escalar y luego trasladar: $p' = T (S p) = TS p$

$$TS = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Trasladar y luego escalar: $p' = S (T p) = ST p$

$$ST = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 6 \\ 0 & 2 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

Matriz de transformación



□ En general

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$p' = M p$$

Matrices de transformación en OpenGL



- ❑ Las matrices de transformación en OpenGL (M) están escondidas, son parte del estado de OpenGL
- ❑ La matriz utilizada para dibujar es la que está activa en ese momento hasta que sea cambiada
- ❑ Hay tres matrices de transformación:
 - ❑ GL_MODELVIEW: 2D/3D transformaciones
 - ❑ GL_PROJECTION: proyección
 - ❑ GL_TEXTURE: texturas

Ejemplo

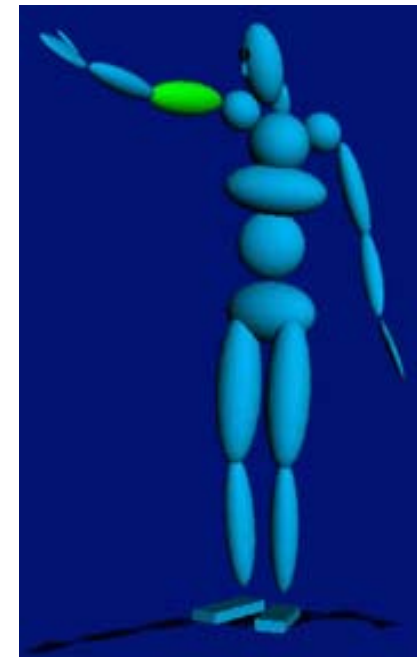
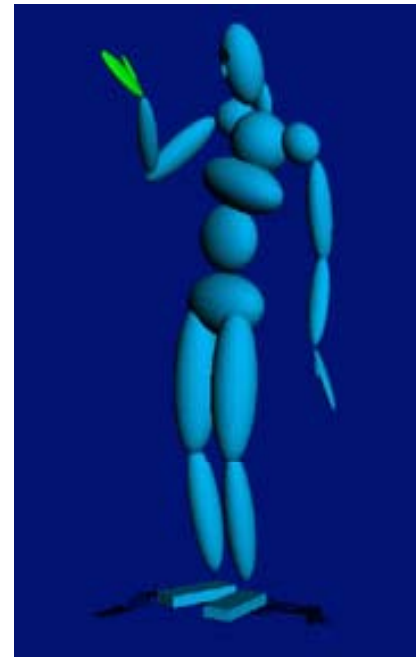
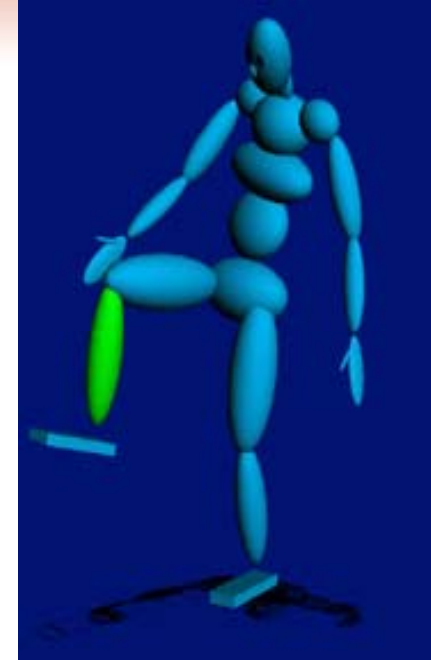


```
glMatrixMode(GL_MODELVIEW);  
glLoadIdentity();  
glTranslatef(dx, dy, dz);  
glRotatef(angle, vx, vy, vz);  
glScalef(sx, sy, sz);
```

Transformaciones en objetos modelados



- ❑ Posición del objeto en la escena
- ❑ Cambiar la forma de los objetos
- ❑ Crear múltiples copias de los objetos
- ❑ Definir la proyección de la cámara virtual
- ❑ Animaciones



Describir una escena



Scene

Camera

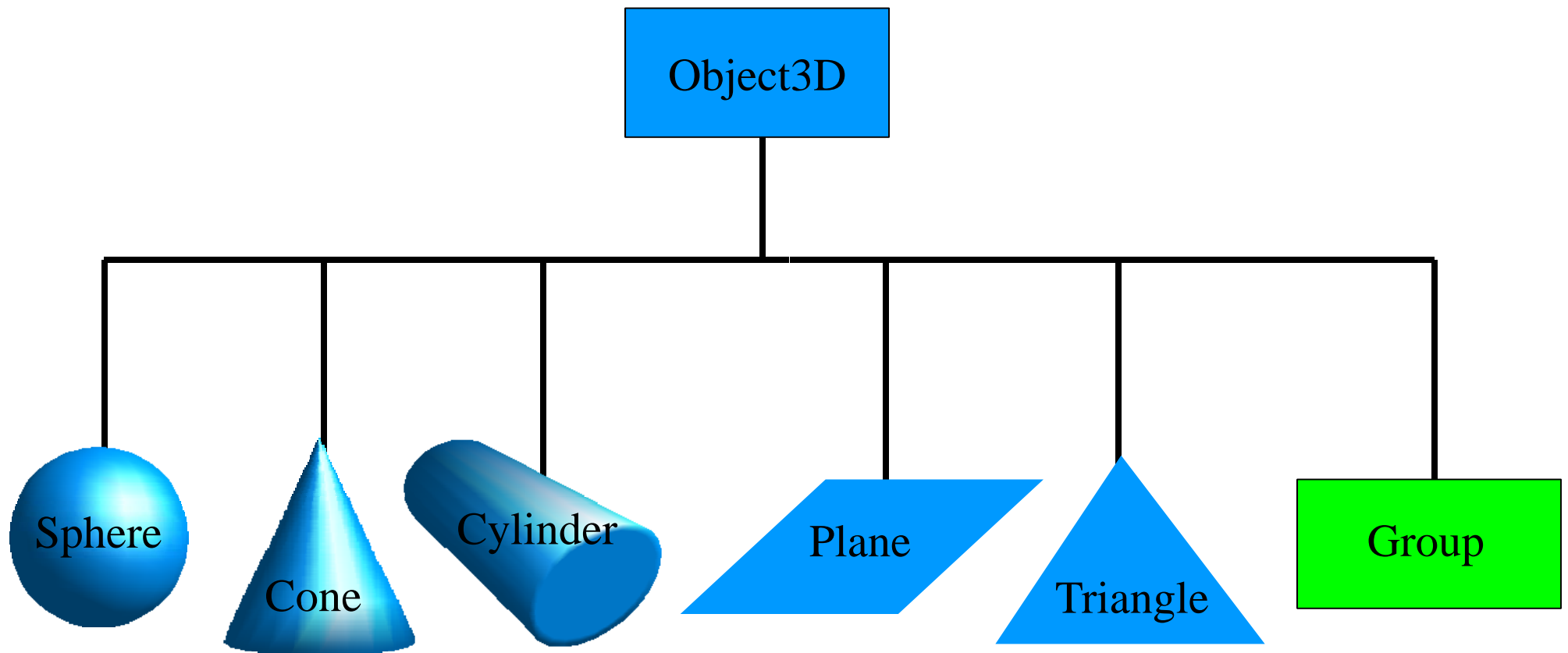
Lights

Background

Materials
*(much more
next week)*

Objects

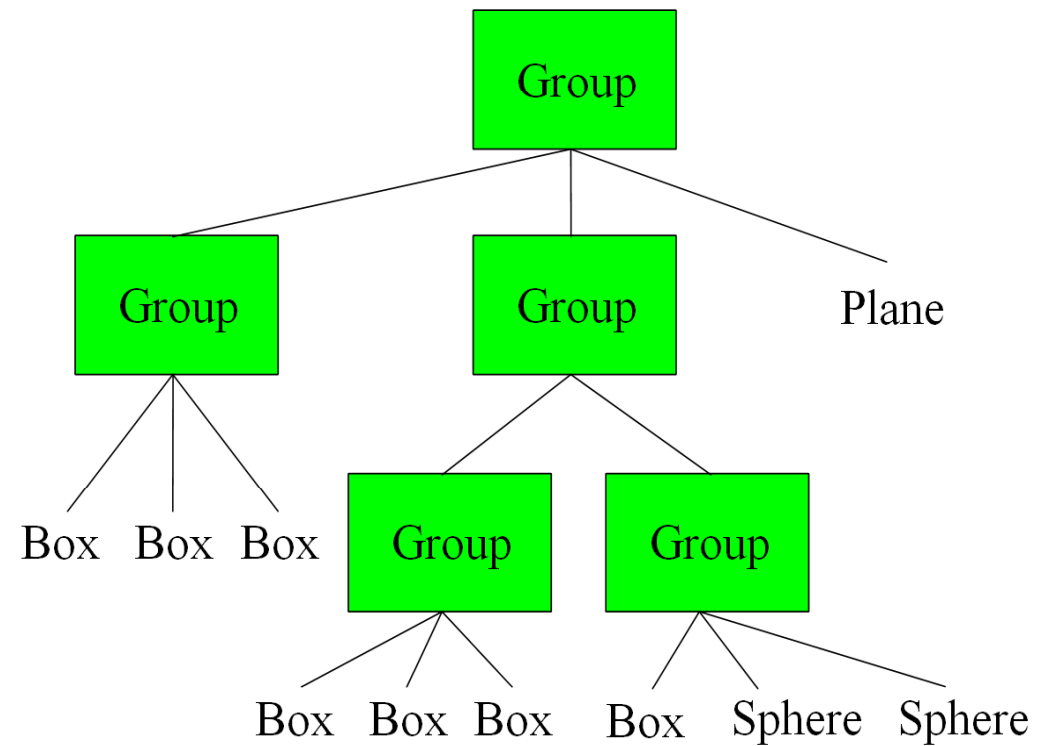
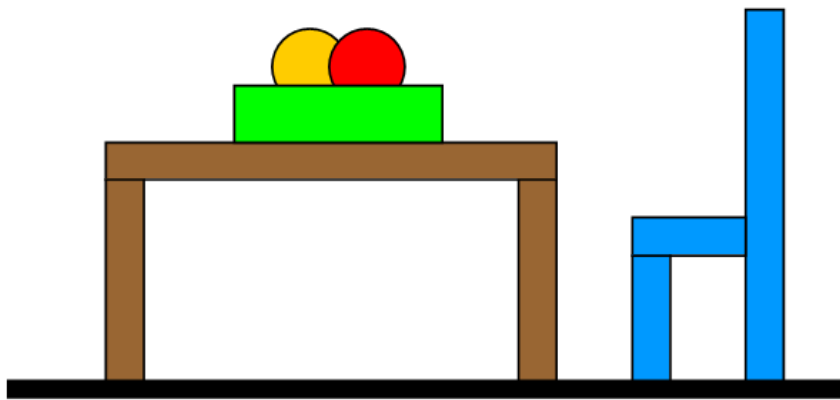
Jerarquia de los objetos



Agrupar los objetos 3D



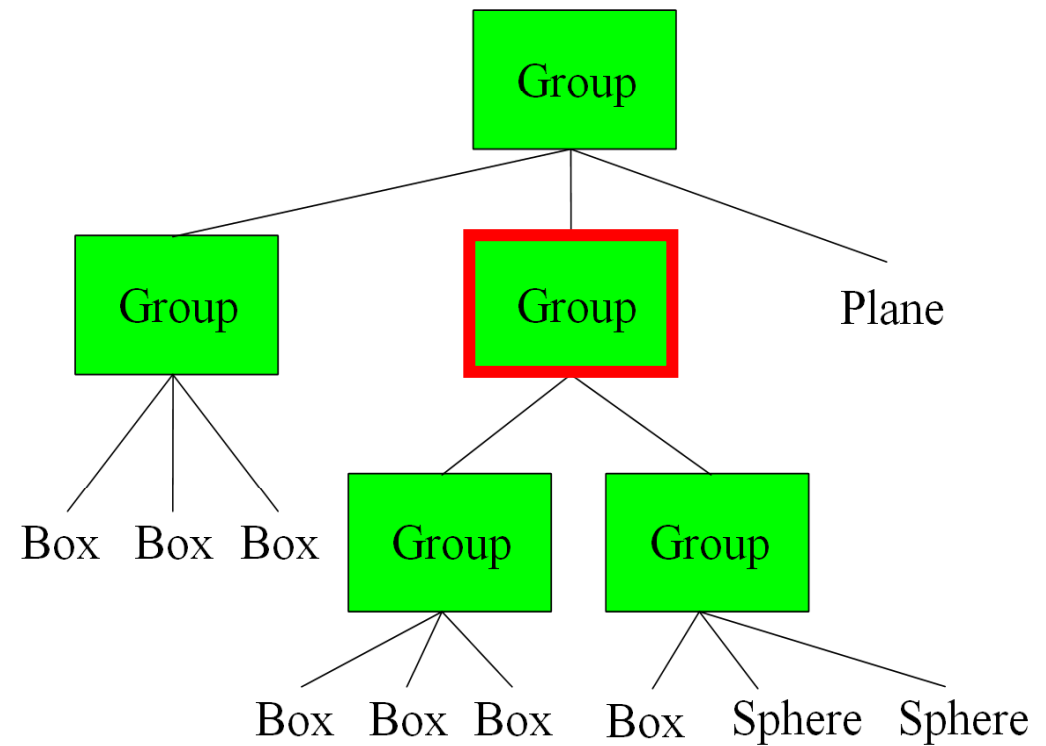
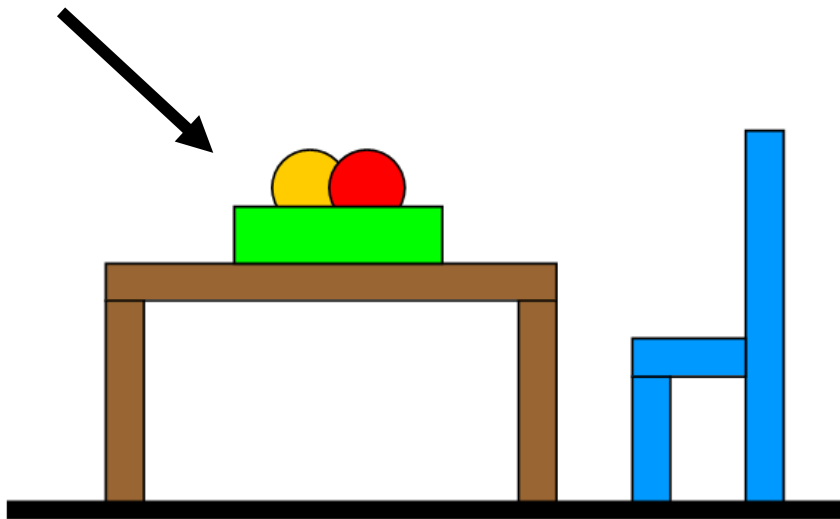
Organización de una escena



Intersección de un rayo a un grupo



- Recursividad a los sub-grupos



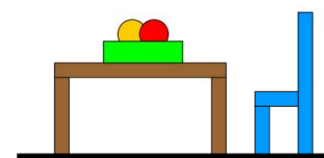
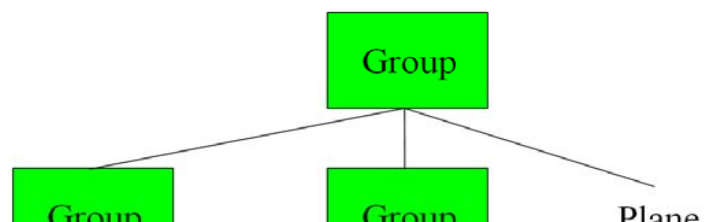
Ejemplo



```
Group {  
  numObjects 3  
  Group {  
    numObjects 3  
    Box { <BOX PARAMS> }  
    Box { <BOX PARAMS> }  
    Box { <BOX PARAMS> } } } 1
```

```
Group {  
  numObjects 2  
  Group {  
    Box { <BOX PARAMS> }  
    Box { <BOX PARAMS> }  
    Box { <BOX PARAMS> } }
```

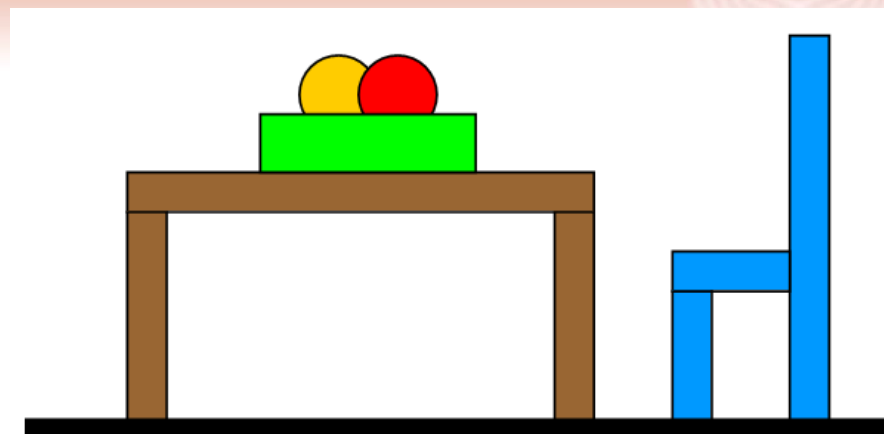
```
Group {  
  Box { <BOX PARAMS> } }  
  Sphere { <SPHERE PARAMS> }  
  Sphere { <SPHERE PARAMS> } } }  
Plane { <PLANE PARAMS> } }
```



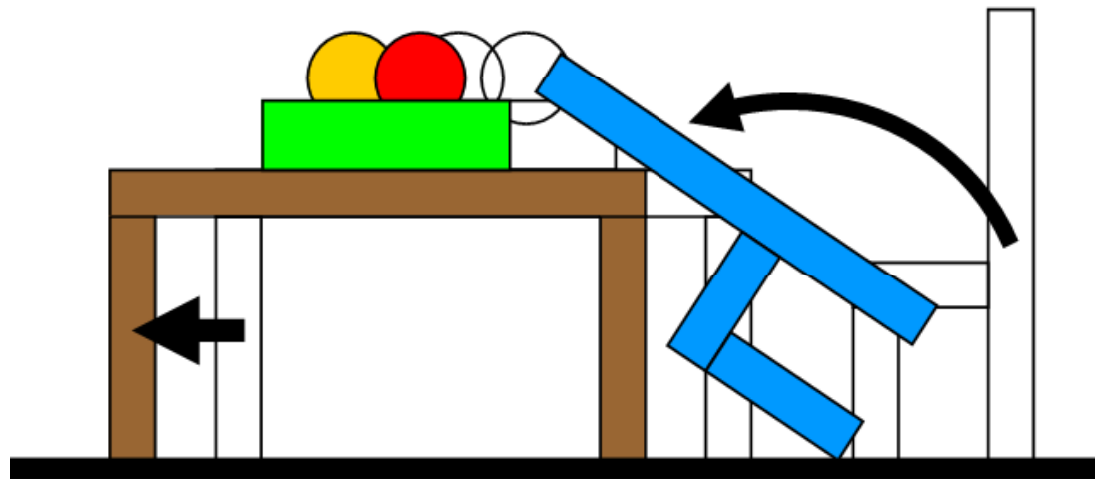
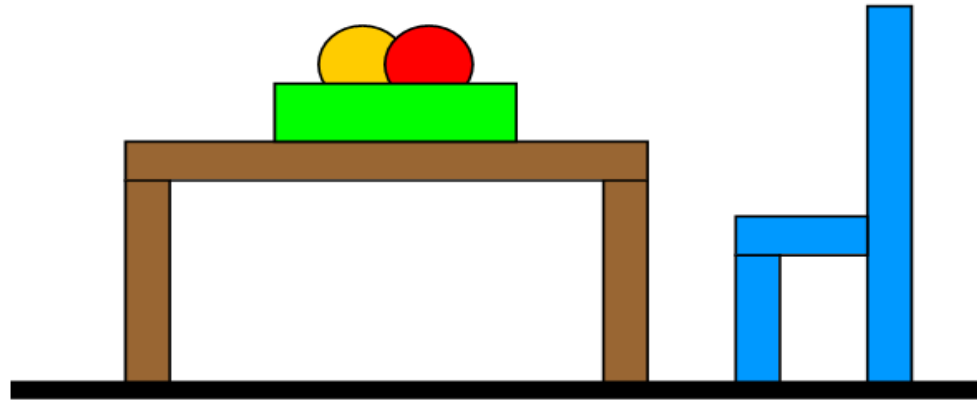
Material



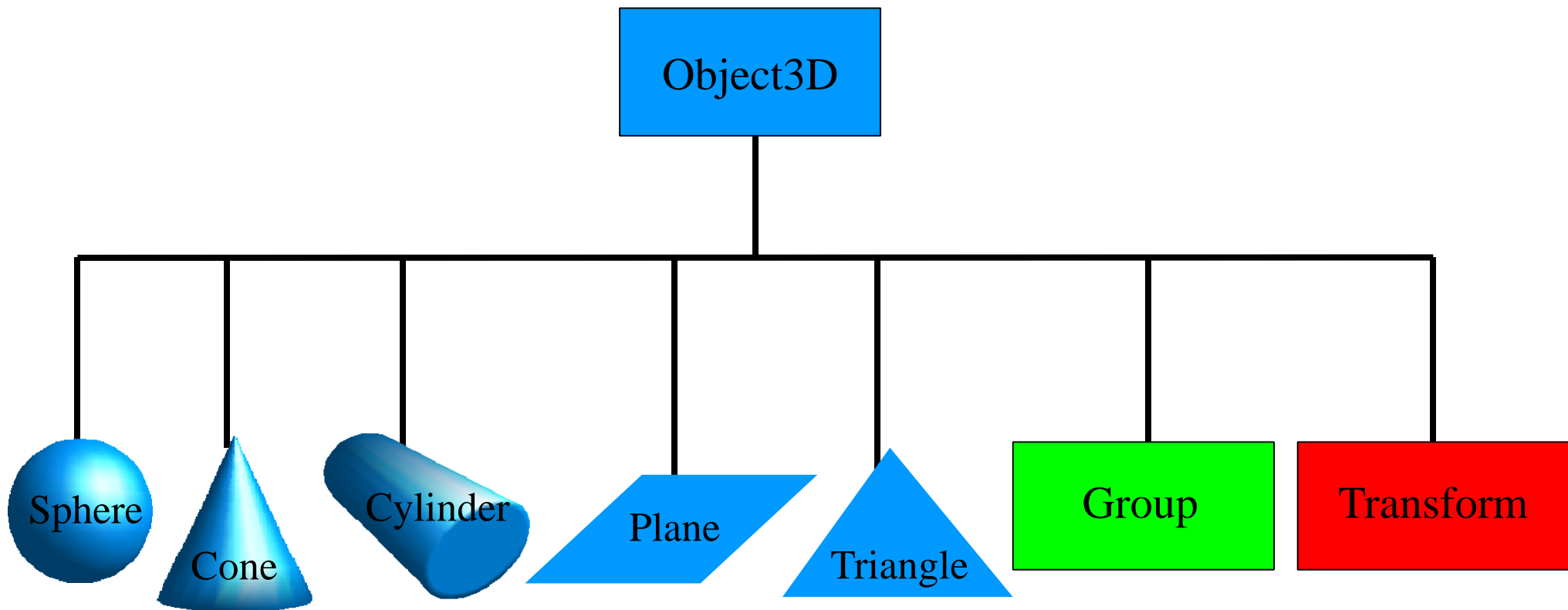
```
Group {
  numObjects 3
  Material { <BLUE> }
  Group {
    numObjects 3
    Box { <BOX PARAMS> }
    Box { <BOX PARAMS> }
    Box { <BOX PARAMS> } }
  Group {
    numObjects 2
    Material { <BROWN> }
    Group {
      Box { <BOX PARAMS> }
      Box { <BOX PARAMS> }
      Box { <BOX PARAMS> } }
    Group {
      Material { <GREEN> }
      Box { <BOX PARAMS> }
      Material { <RED> }
      Sphere { <SPHERE PARAMS> }
      Material { <ORANGE> }
      Sphere { <SPHERE PARAMS> } } } }
  Material { <BLACK> }
  Plane { <PLANE PARAMS> } }
```



Transformaciones a grupos



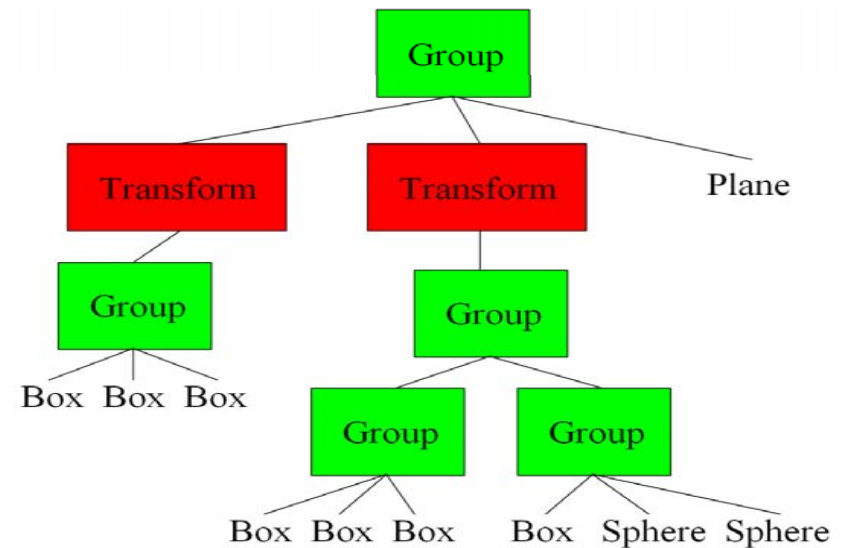
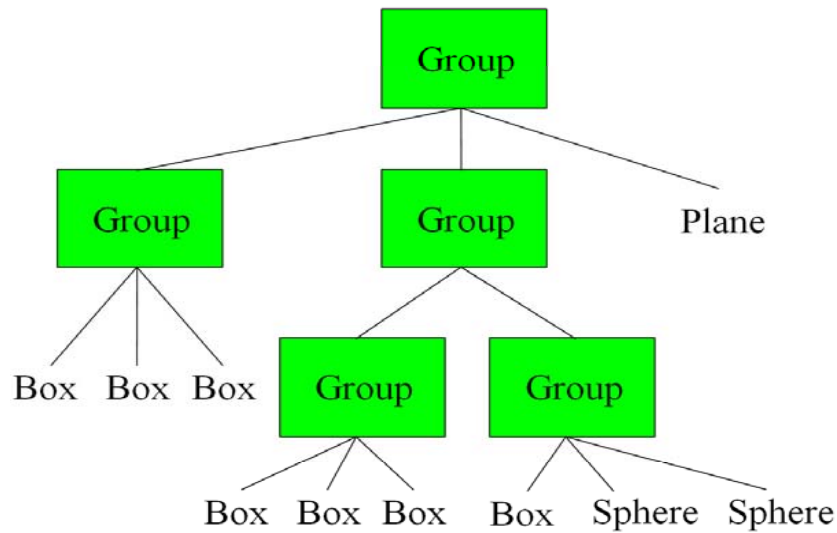
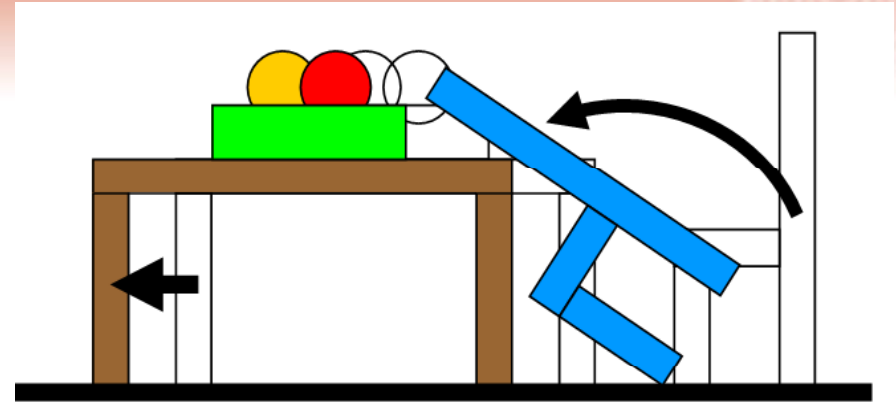
Clases jerárquicas



Transformaciones



- Posicionar grupos de objetos dentro de la escena



Ejemplo



```
Group {
  numObjects 3
  Transform {
    ZRotate { 45 }
    Group {
      numObjects 3
      Box { <BOX PARAMS> }
      Box { <BOX PARAMS> }
      Box { <BOX PARAMS> } }
    Transform {
      Translate { -2 0 0 }
      Group {
        numObjects 2
        Group {
          Box { <BOX PARAMS> }
          Box { <BOX PARAMS> }
          Box { <BOX PARAMS> } }
        Group {
          Box { <BOX PARAMS> }
          Sphere { <SPHERE PARAMS> }
          Sphere { <SPHERE PARAMS> } } } }
  Plane { <PLANE PARAMS> } }
```

