

# Auxiliar 5

## **Repaso: Pipeline - Transformaciones – Grafos de Escena**

CC3501 Modelación y Computación Gráfica para Ingenieros

Primavera 2023

Profesor: Iván Sipiran

Auxiliar: Ariel Riveros

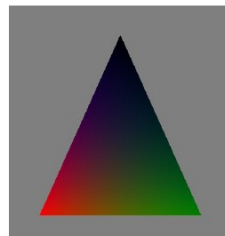
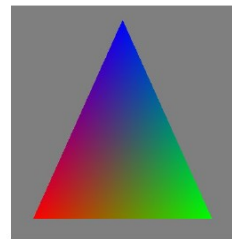
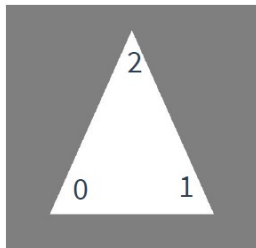
# Vértices

Un **vértice** es una estructura de datos que contiene la información necesaria para que la GPU dibuje o realice operaciones.

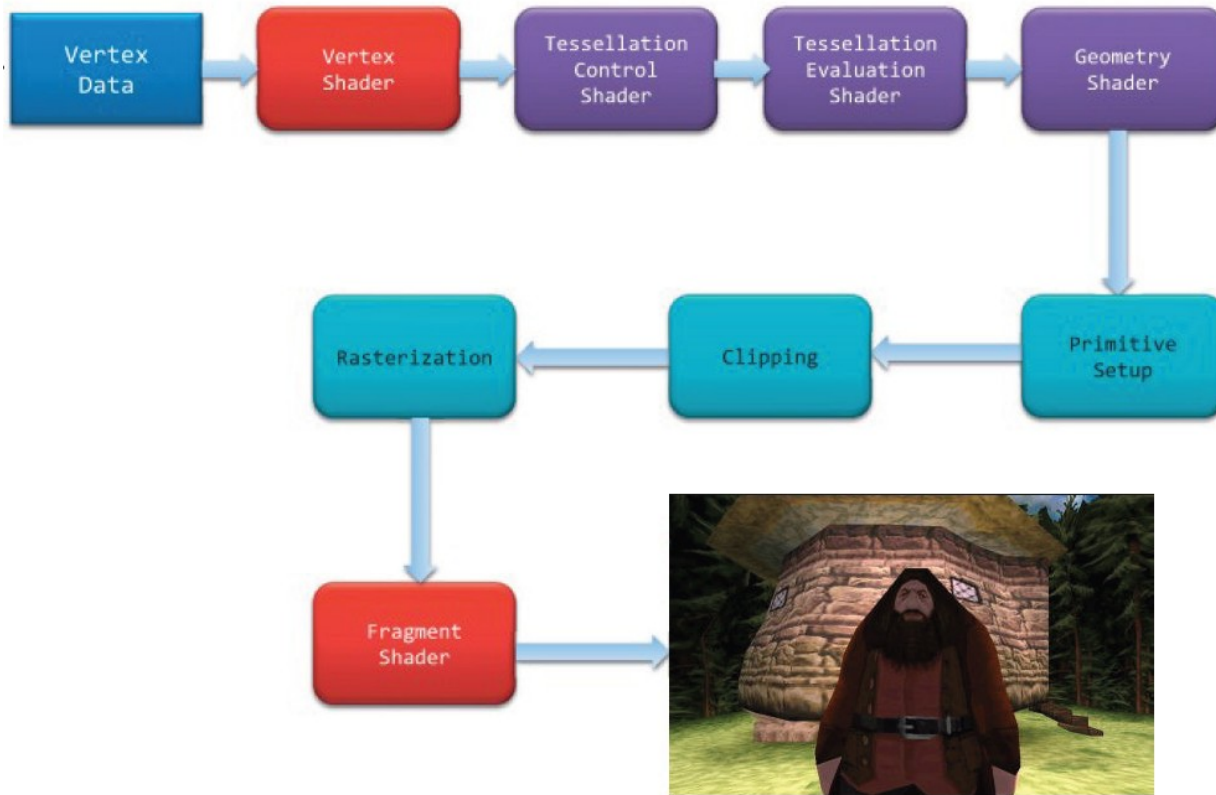
Son definidos por el usuario y están conformados por atributos que la GPU puede leer

atributo vértice	Posición (float)		Color (float)			Intensidad (float)
	x	y	r	g	b	
0	-0,5	-0,5	1,0	0,0	0,0	1,0
1	0,5	-0,5	0,0	1,0	0,0	0,5
2	0,0	0,5	0,0	0,0	1,0	0,0

Cómo se verían  
al final del pipeline:

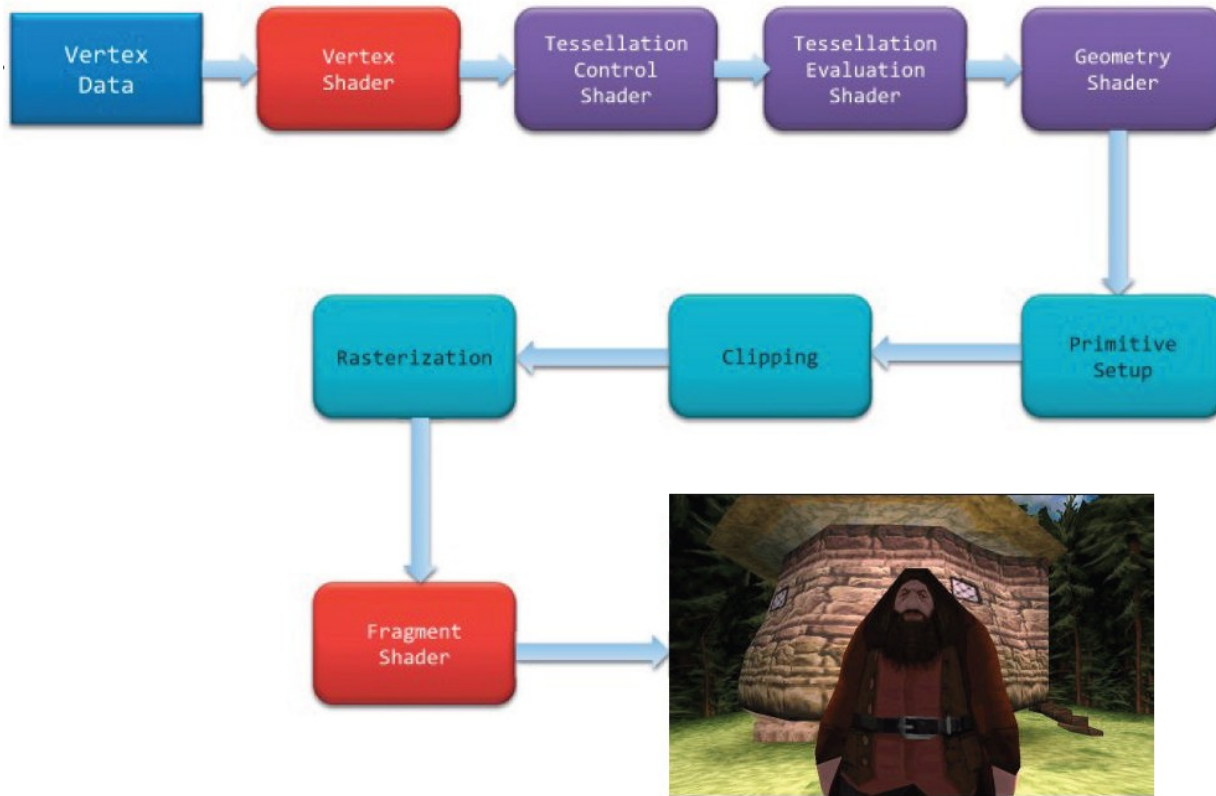


# Pipeline



Los vértices son alimentados a la GPU, donde se inicia un procedimiento llamado *rendering pipeline*: Serie de pasos que resultan en la generación de imágenes

# Shaders



Un *Shader* es un programa que se ejecuta en la GPU, los más importantes son:

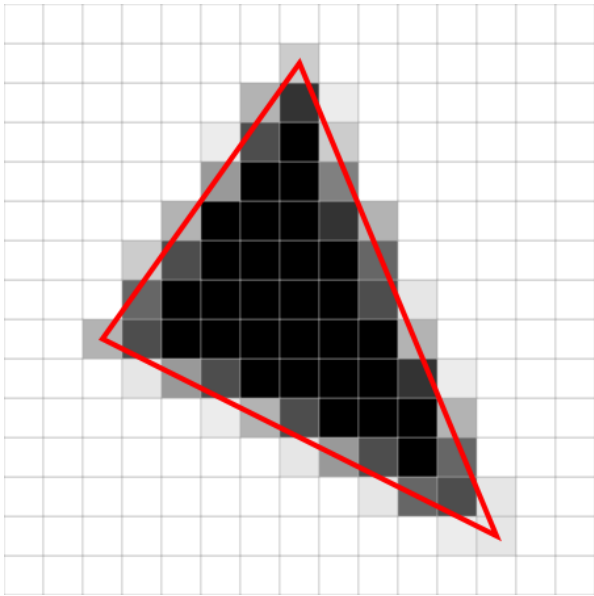
Vertex Shader:

Punto de entrada del pipeline, maneja las propiedades de cada vértice

Fragment Shader:

Luego del *rasterizado* maneja qué color se le aplica a cada pixel

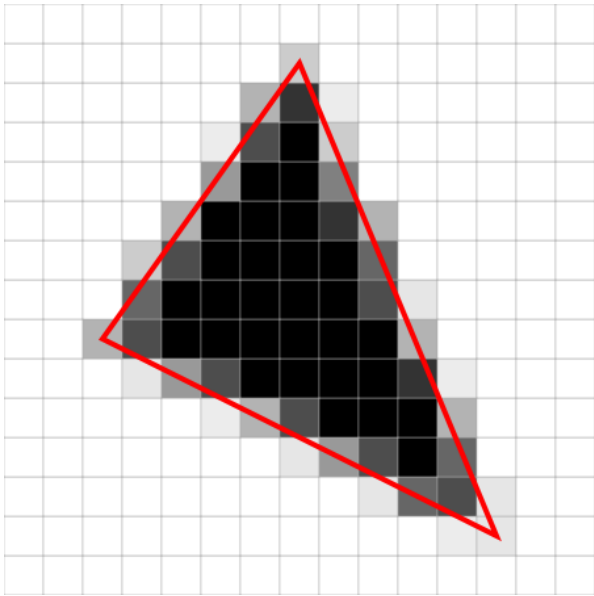
# Rasterización



Procedimiento que realiza la GPU para asignar cada porción de geometría descrita por vértices a un pixel correspondiente

Aquí es donde ocurre la interpolación

# Interpolación

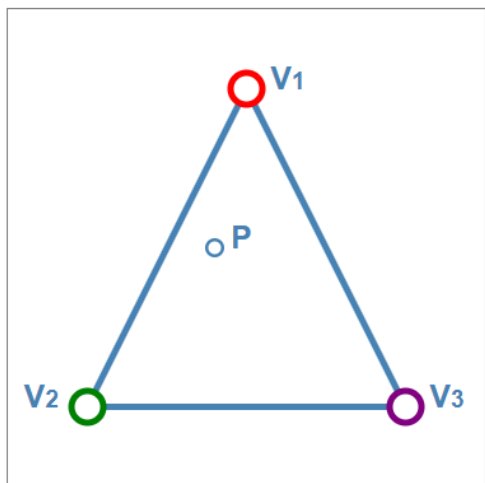


Al pipeline solo se le entregaron 3 puntos del triángulo  
La rasterización interpola todos los pixeles contenidos en él

La interpolación gráfica es la estimación de  
valores intermedios entre vértices para  
asignarlos a pixeles.

# Interpolación

Coordenadas baricéntricas:



$$w_{v_1} = \frac{(y_{v_2} - y_{v_3})(p_x - x_{v_3}) + (x_{v_3} - x_{v_2})(p_y - y_{v_3})}{(y_{v_2} - y_{v_3})(x_{v_1} - x_{v_3}) + (x_{v_3} - x_{v_2})(y_{v_1} - y_{v_3})}$$

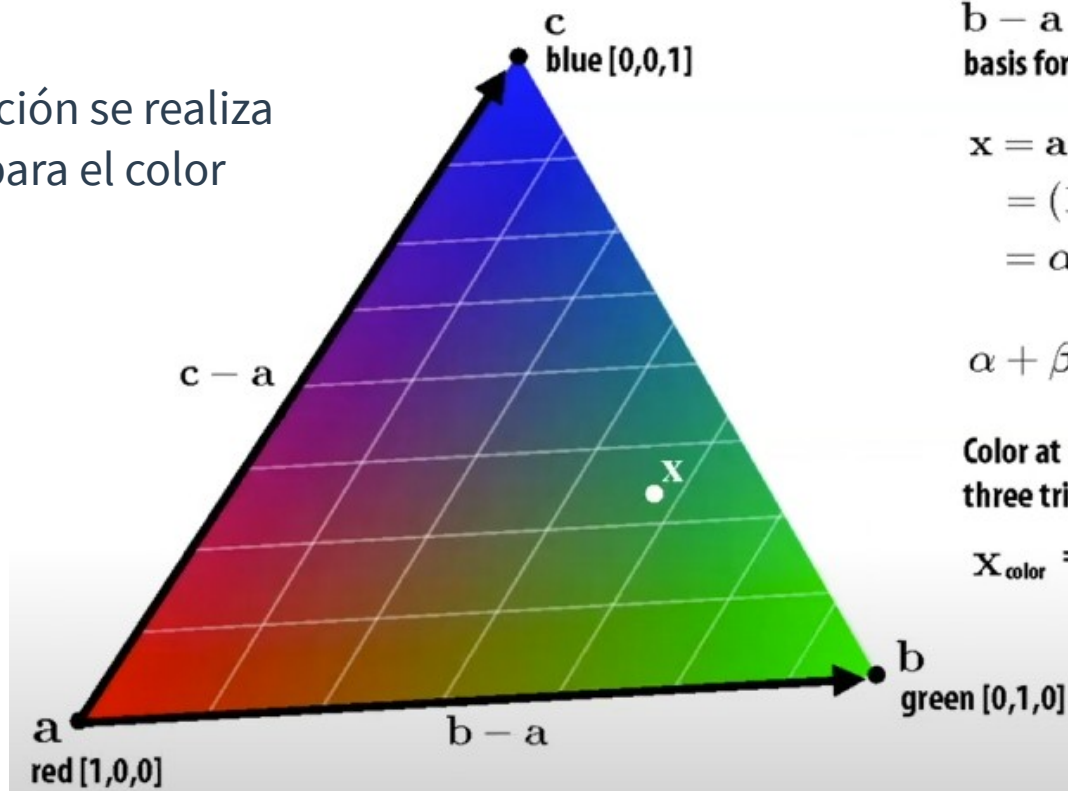
$$w_{v_2} = \frac{(y_{v_3} - y_{v_1})(p_x - x_{v_3}) + (x_{v_1} - x_{v_3})(p_y - y_{v_3})}{(y_{v_2} - y_{v_3})(x_{v_1} - x_{v_3}) + (x_{v_3} - x_{v_2})(y_{v_1} - y_{v_3})}$$

$$w_{v_3} = 1 - w_{v_1} - w_{v_2}$$

<https://www.youtube.com/watch?v=HYAgJN3x4GA>

# Interpolación

La interpolación se realiza también para el color



$b - a$  and  $c - a$  form a non-orthogonal basis for points in triangle (origin at  $a$ )

$$\begin{aligned}x &= a + \beta(b - a) + \gamma(c - a) \\&= (1 - \beta - \gamma)a + \beta b + \gamma c \\&= \alpha a + \beta b + \gamma c\end{aligned}$$

$$\alpha + \beta + \gamma = 1$$

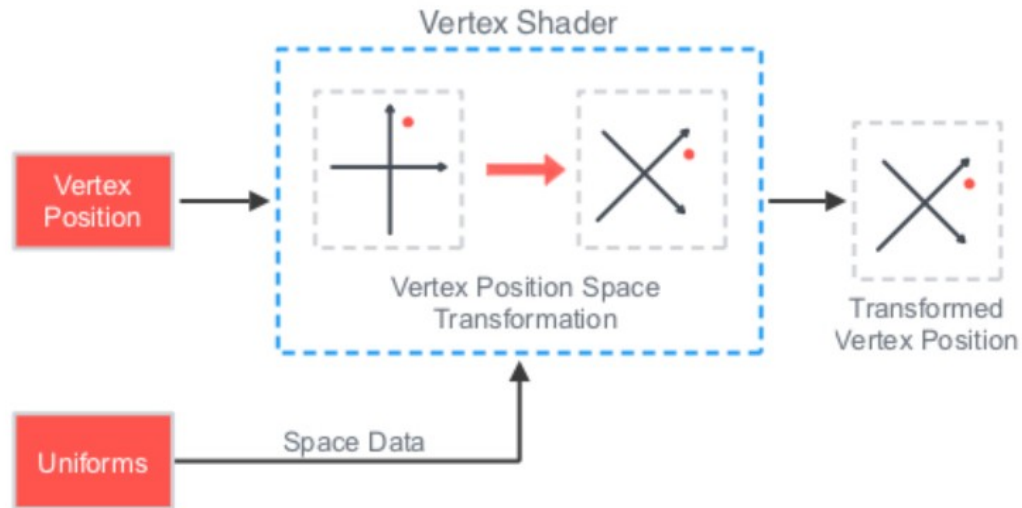
Color at  $x$  is linear combination of color at three triangle vertices.

$$x_{\text{color}} = \alpha a_{\text{color}} + \beta b_{\text{color}} + \gamma c_{\text{color}}$$



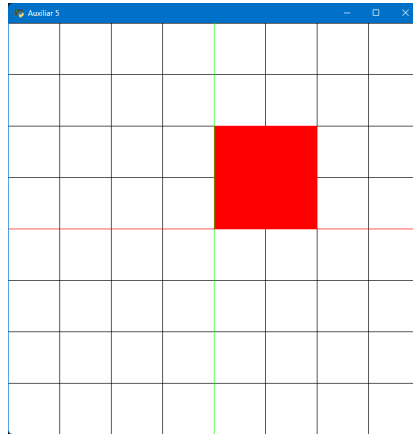
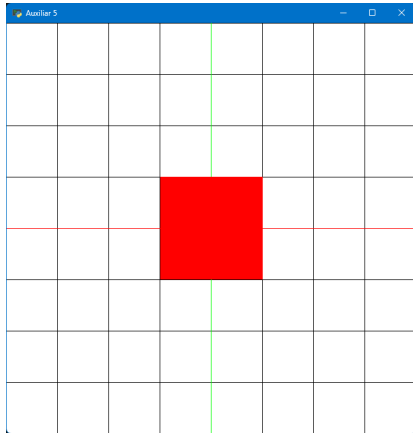
# Transformaciones

En el *pipeline*, se aplican las transformaciones a cada vértice por medio de *uniforms*, variables globales enviadas a la GPU desde la aplicación



# Transformaciones

## Traslación



A cada vértice del cuadrado  
le sumo el vector  $[1, 1, 0]$

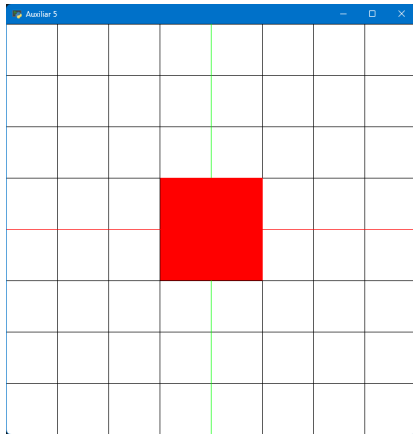
# Transformaciones

## Traslación en coordenadas homogéneas

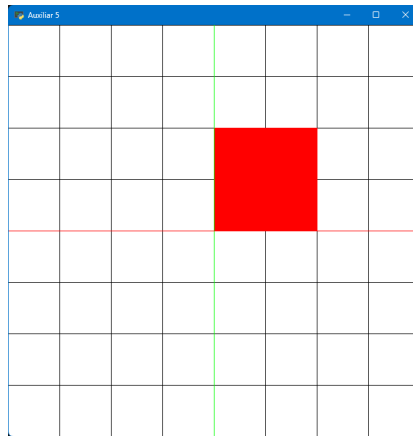
$$T(dx, dy, dz) = \begin{pmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + dx \\ y + dy \\ z + dz \\ 1 \end{pmatrix}$$

# Transformaciones

## Traslación



$T(1, 1, 0)$



A cada vértice del cuadrado  
le aplico la matriz

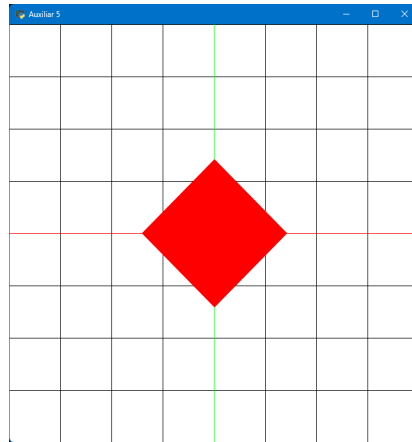
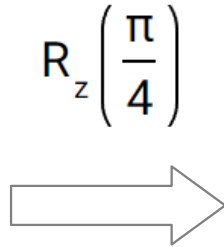
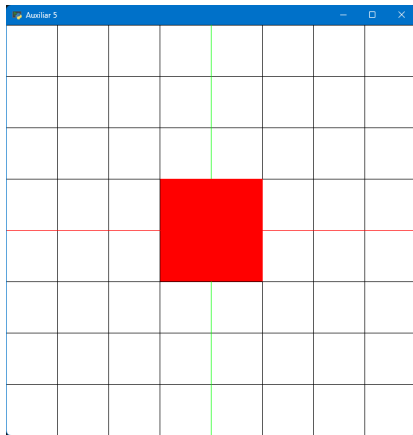
$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

**Esto permite concatenación de  
transformaciones**

# Transformaciones

## Rotación

Importante: La rotación siempre es con origen en (0, 0)



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

# Transformaciones

## Rotación en todos los ejes

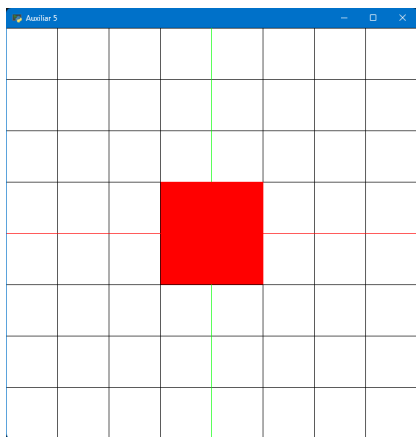
$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

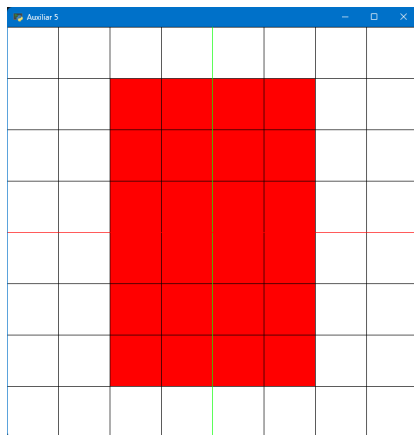
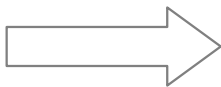
$$R_z(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

# Transformaciones

## Escala



$S(2, 3, 0)$

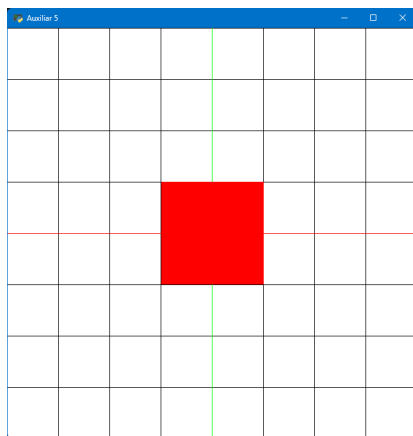


$$S(sx, sy, sz) = \begin{pmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

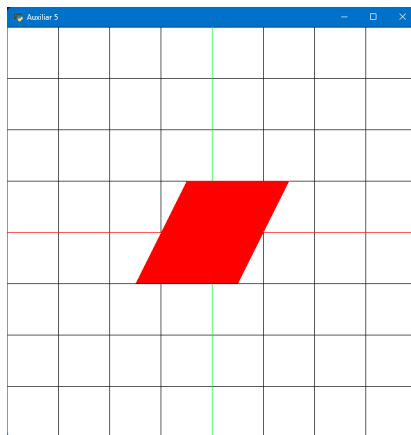
# Transformaciones

## Shear o Corte

$$\text{Sh}(xy, yx, xz, zx, yz, zy) = \begin{pmatrix} 1 & xy & xz & 0 \\ yx & 1 & yz & 0 \\ zx & zy & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



$\text{Sh}(0.5, 0, 0, 0, 0, 0)$

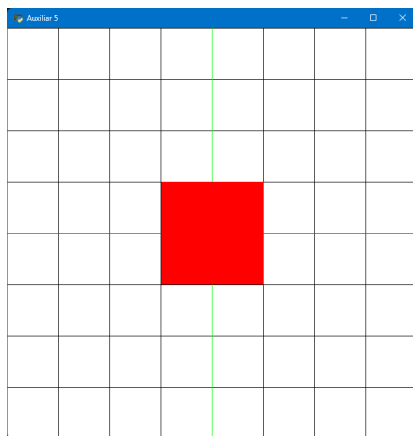




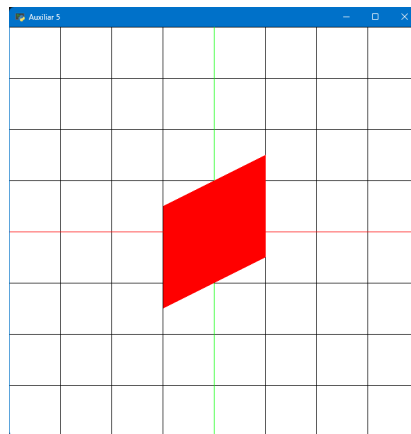
# Transformaciones

## Shear o Corte

$$\text{Sh}(xy, yx, xz, zx, yz, zy) = \begin{pmatrix} 1 & xy & xz & 0 \\ yx & 1 & yz & 0 \\ zx & zy & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



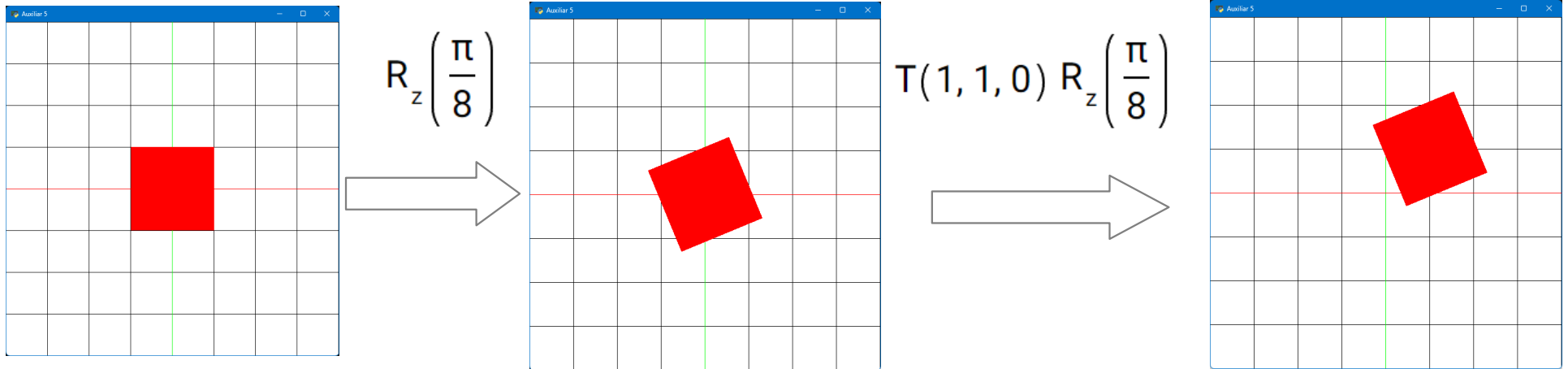
$\text{Sh}(0, 0.5, 0, 0, 0, 0)$



# Transformaciones

## Concatenación de Transformaciones

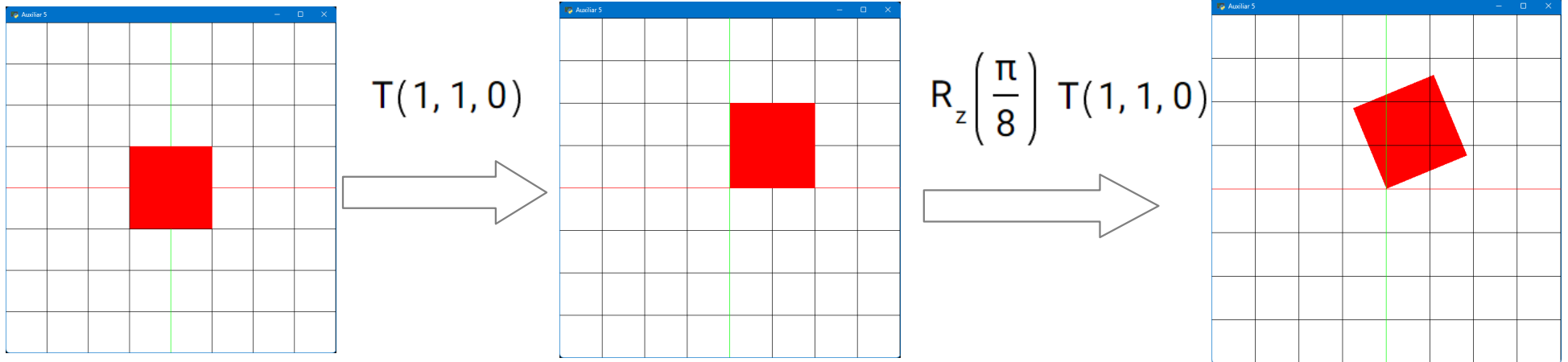
Como las transformaciones son matrices de igual dimensión, pueden multiplicarse



# Transformaciones

## Concatenación de Transformaciones

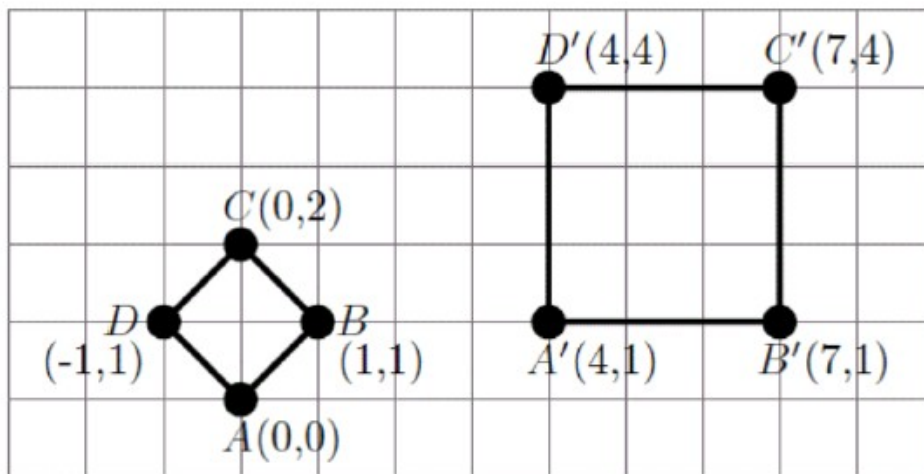
No es una operación conmutativa



# Transformaciones

## P3: Transformaciones (1.5 ptos)

Considere la siguiente figura:

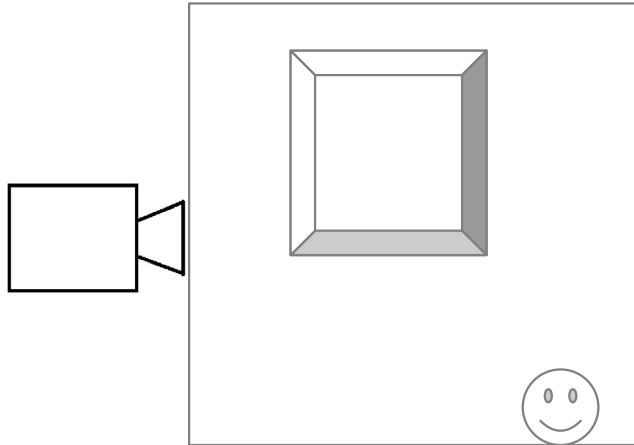


Cuál es la matriz de transformación (o producto de matrices) que transforma el cuadrilátero ABCD en el cuadrilátero A'B'C'D'?

# Más Transformaciones

## View Matrix

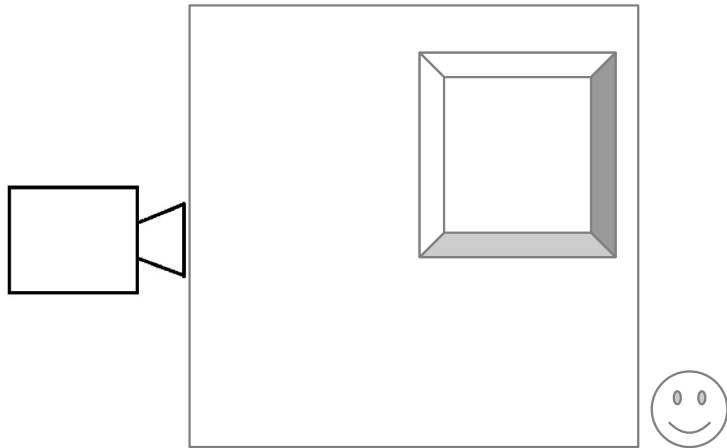
Transforma todo el mundo a la transformación inversa de la cámara



# Más Transformaciones

## View Matrix

Transforma todo el mundo a la transformación inversa de la cámara

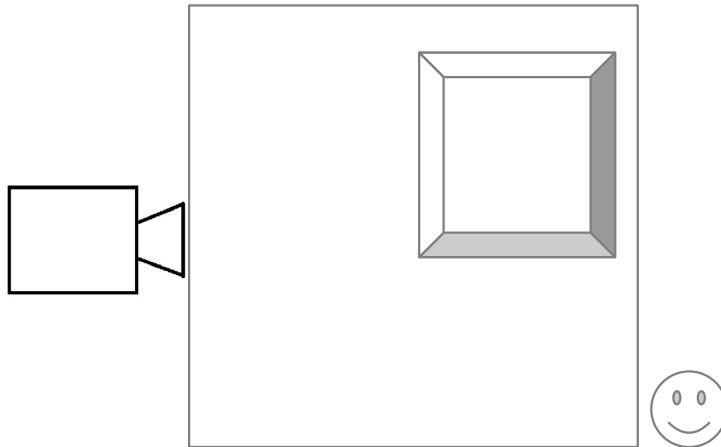


Si muevo la cámara a la izquierda

# Más Transformaciones

## View Matrix

Transforma todo el mundo a la transformación inversa de la cámara

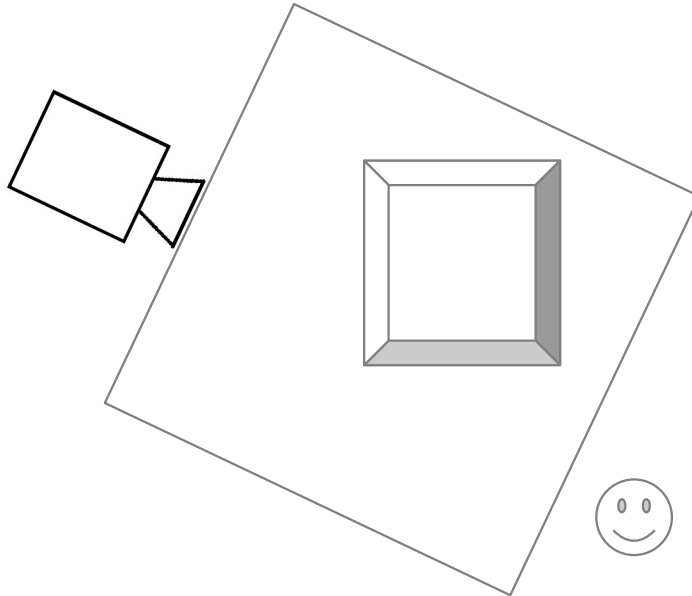


Si muevo la cámara a la izquierda  
En realidad muevo los objetos a la derecha

# Más Transformaciones

## View Matrix

Transforma todo el mundo a la transformación inversa de la cámara



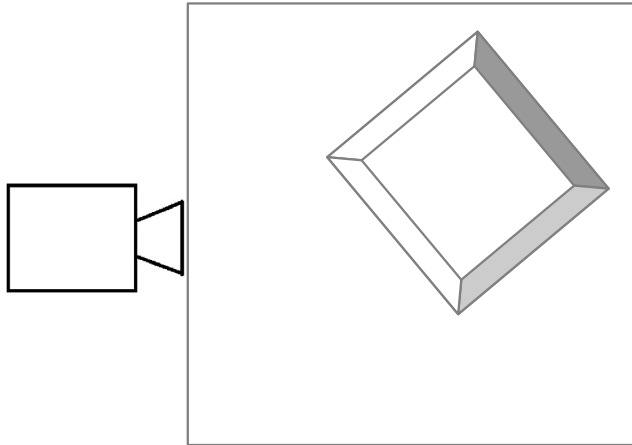
Si muevo la cámara a la izquierda  
En realidad muevo los objetos a la derecha  
Lo mismo con la rotación



# Más Transformaciones

## View Matrix

Transforma todo el mundo a la transformación inversa de la cámara



Si muevo la cámara a la izquierda

En realidad muevo los objetos a la derecha

Lo mismo con la rotación



# Más Transformaciones

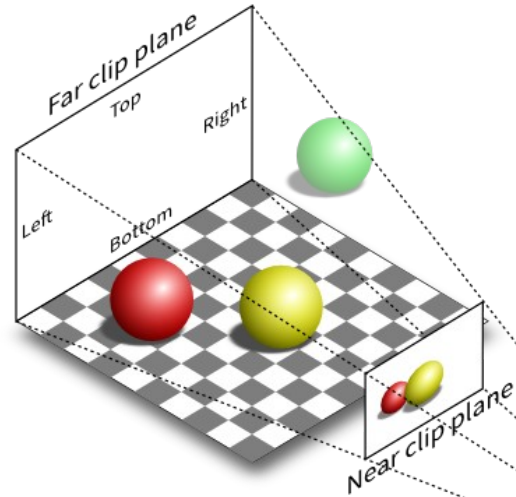
## View Matrix

<https://jsantell.com/model-view-projection/mvp.webm>

# Más Transformaciones

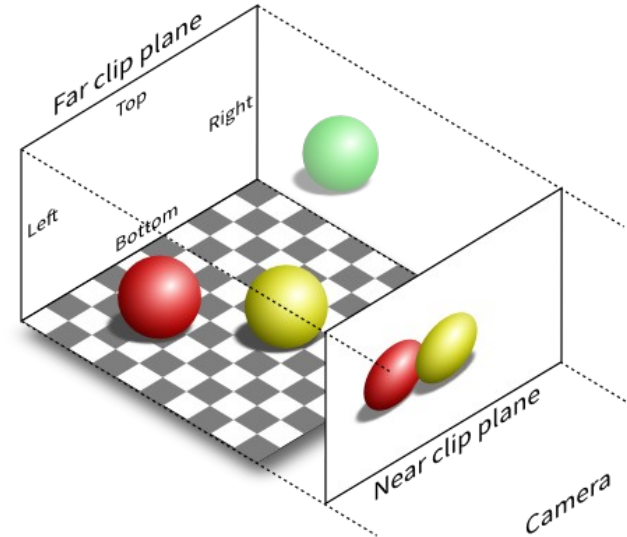
## Proyecciones

En perspectiva,  
se entrega la percepción de profundidad



Perspective projection (P)

En ortográfica,  
Se mantiene paralelismo



Orthographic projection (O)

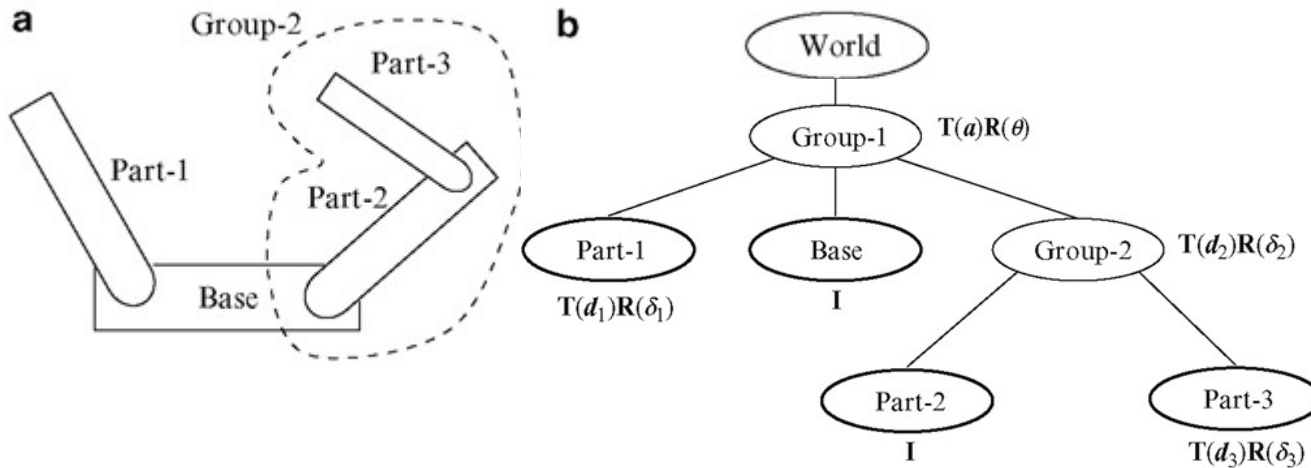
# Más Transformaciones

## Proyecciones

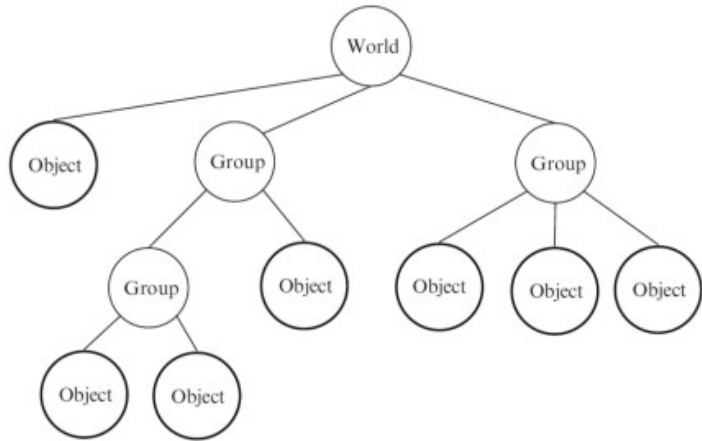
<https://jsantell.com/3d-projection/world-to-ndc.webm>

# Grafos de Escena

Son estructuras y relaciones de objetos en una escena.  
Ayudan a organizar, describir y optimizar la renderización



# Grafos de Escena

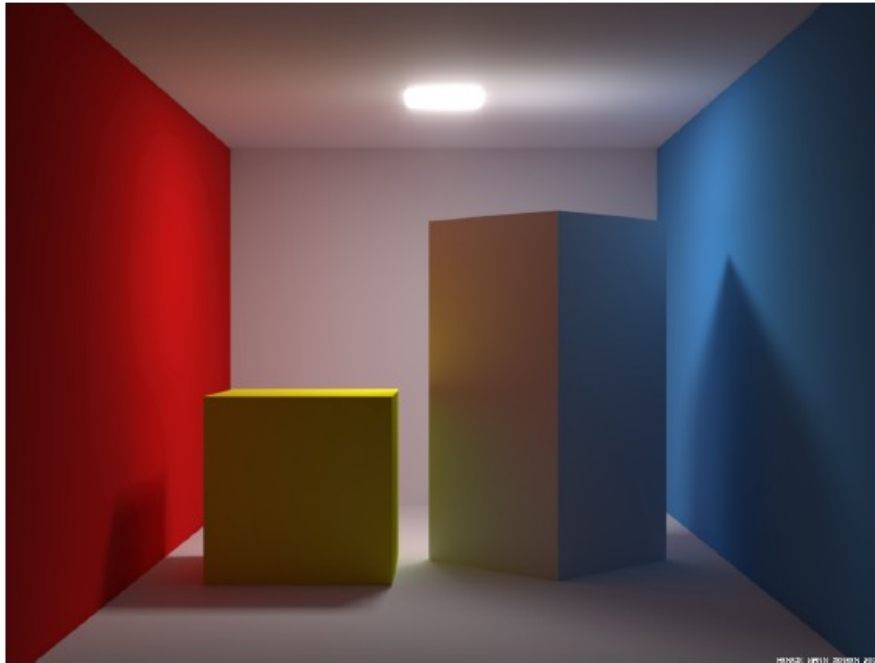


- Cada hoja del árbol es un objeto básico.
- Cada nodo interno representa un grupo de objetos.
- Cada arco representa una transformación.
- La transformación final de un objeto es la composición de todas las transformaciones desde la raíz hasta la hoja.

# Grafos de Escena

## P1: Scene Graph y Transformaciones (1.5 pts)

Dada la siguiente imagen de la Caja de Cornell:

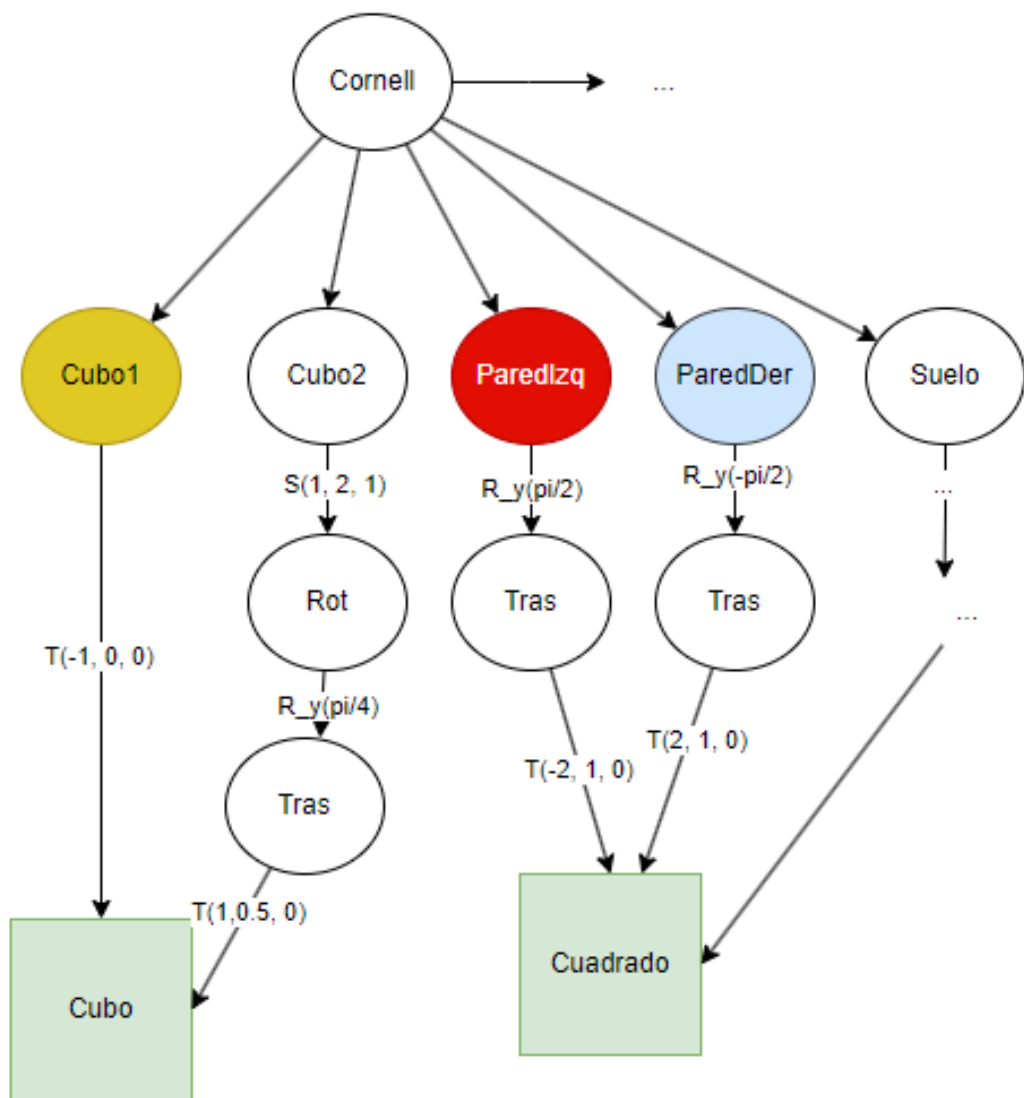
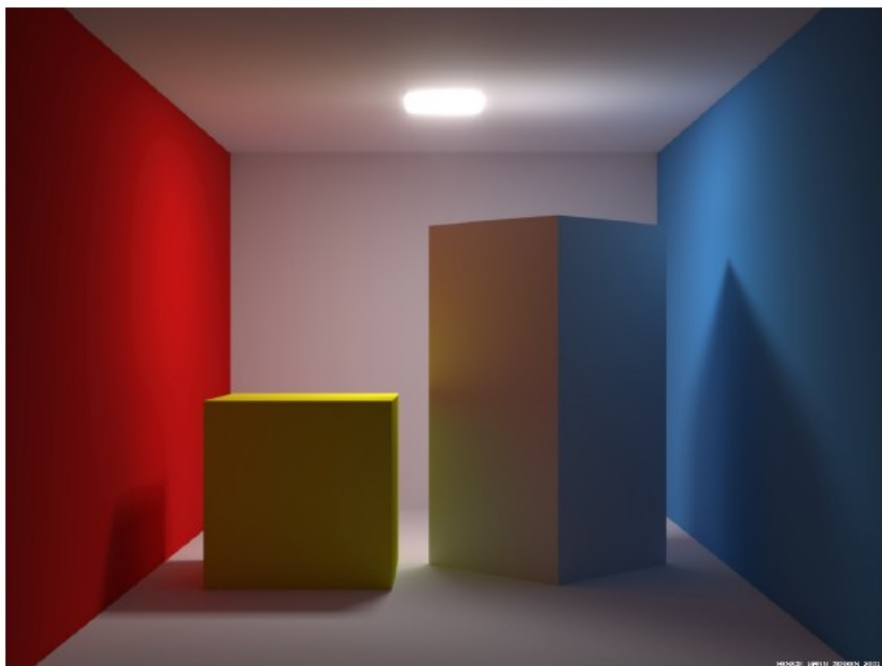


Describa los objetos 3D (vértices e índices) y el grafo de escena necesario para generarla de manera eficiente.

No considere los efectos de iluminación ni sombras, sin embargo, la fuente de luz sí existe como objeto (asuma que es una luz rectangular).

Los tamaños y ángulos de rotación quedan a su discreción mientras mantengan la coherencia visual de la imagen.

# Grafos de Escena





# Auxiliar 5

## **Repaso: Pipeline - Transformaciones – Grafos de Escena**

CC3501 Modelación y Computación Gráfica para Ingenieros

Primavera 2023

Profesor: Iván Sipiran

Auxiliar: Ariel Riveros