

# Informática Gráfica

## Grafos de escena

Juan Carlos Torres  
**Grupos C y D**

Dpt. Lenguajes y Sistemas Informáticos  
ETSI Informática y de Telecomunicación  
Universidad de Granada

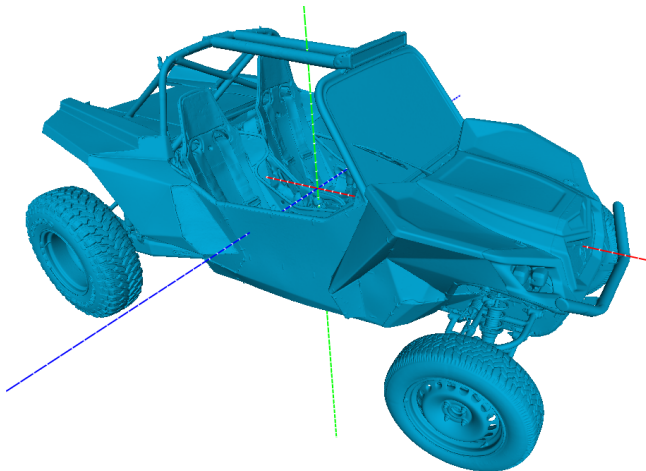
---

Curso 2024-25

# Grafos de escena

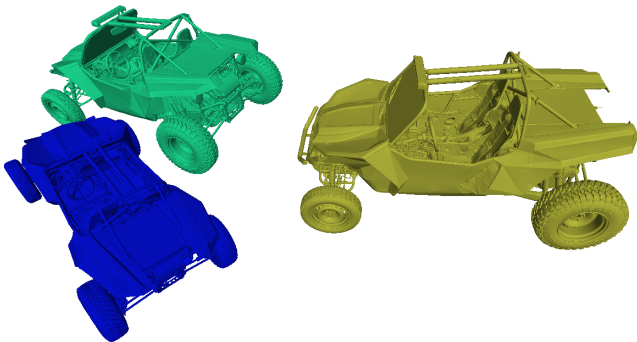
# Estructuración de los modelos

Este modelo tiene 200.000 vértices. ¿Que hacemos si queremos crear una escena con varios?



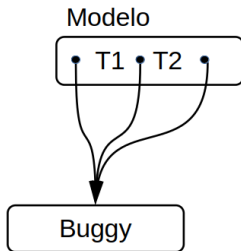
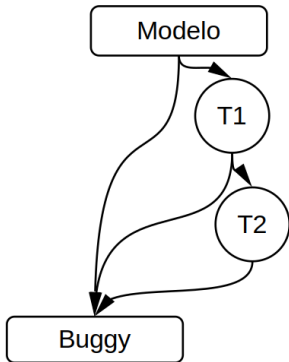
# Instanciación

```
buggy();  
Transformacion(T1);  
buggy();  
Transformacion(T2);  
buggy();
```



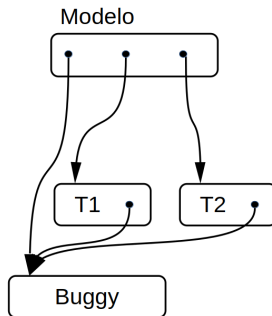
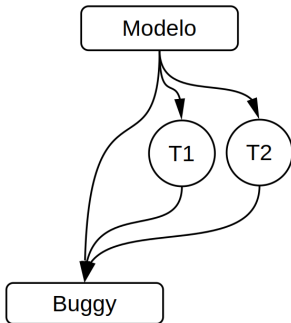
# Grafo de escena

```
buggy();  
Transformacion(T1);  
buggy();  
Transformacion(T2);  
buggy();
```

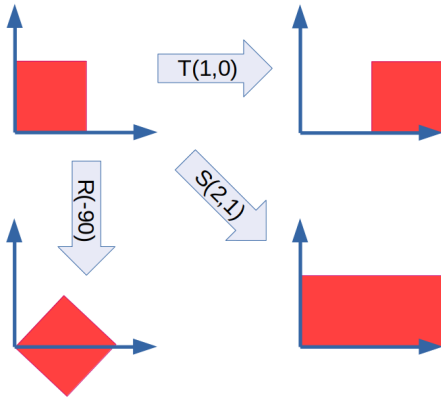


# Independizar transformaciones

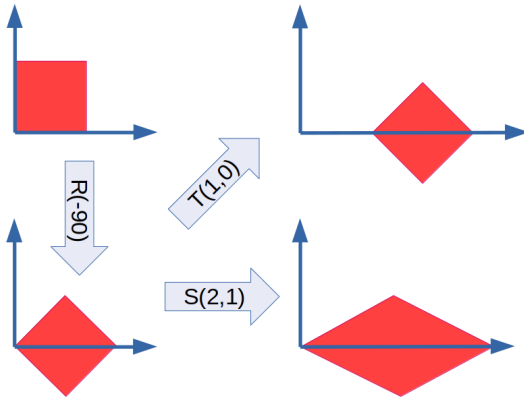
```
buggy();  
glPushMatrix();  
  Transformacion(T1);  
  buggy();  
glPopMatrix();  
Transformacion(T2);  
buggy();
```



# Orden de aplicación de transformaciones

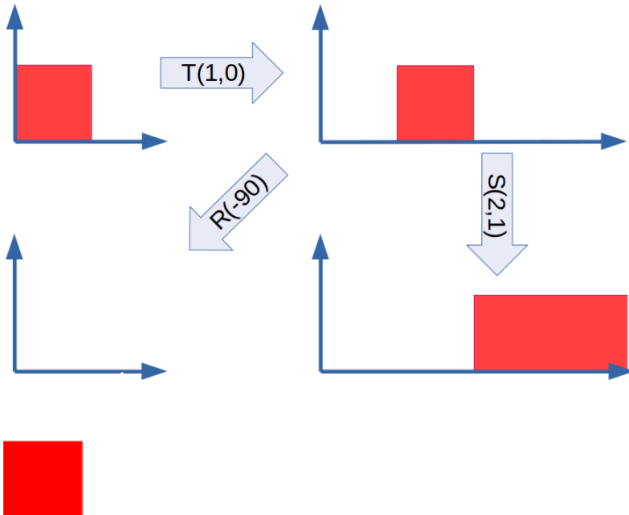


# Orden de aplicación de transformaciones

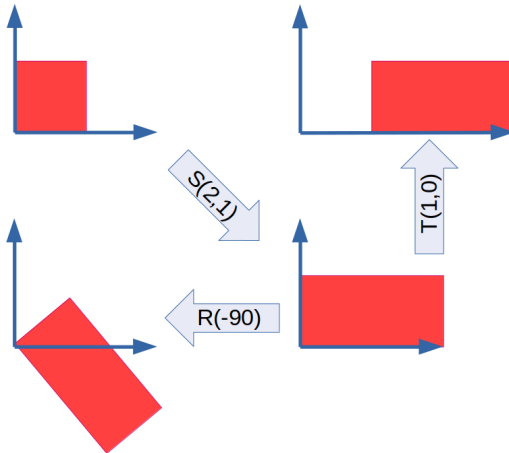




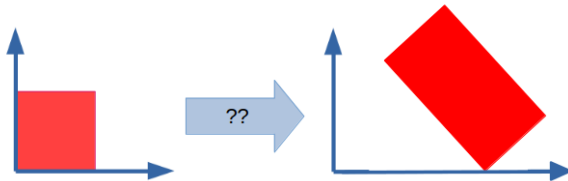
# Orden de aplicación de transformaciones



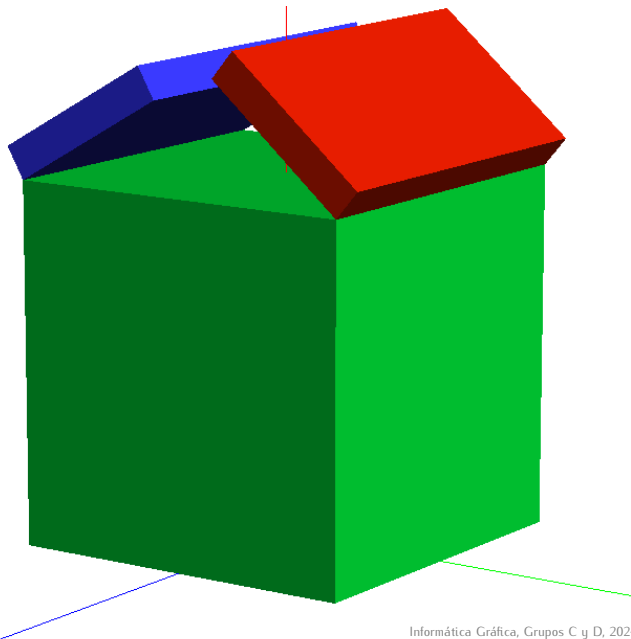
# Orden de aplicación de transformaciones



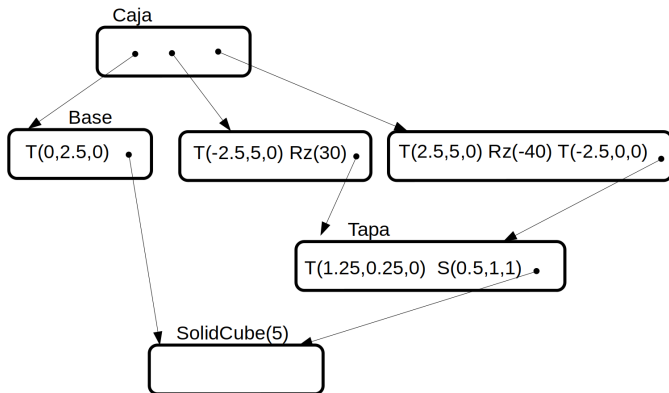
# Orden de aplicación de transformaciones



## Ejemplo de grafo de escena



# Ejemplo de grafo de escena



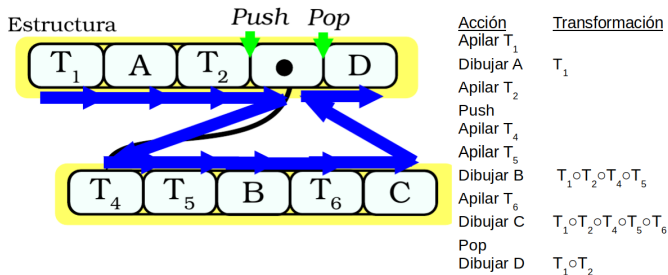
# Ejemplo de grafo de escena I

```
void Base() {  
    glMaterialfv( GL_FRONT, GL_AMBIENT_AND_DIFFUSE, green );  
    glPushMatrix();  
        glTranslatef(0.0,2.5,0.0);  
        glutSolidCube(5);  
    glPopMatrix();  
}  
  
void Tapa() {  
    glPushMatrix();  
        glTranslatef(1.25,0.25,0);  
        glScalef(0.5,0.1,1);  
        glutSolidCube(5);  
    glPopMatrix();  
}
```

# Ejemplo de grafo de escena I

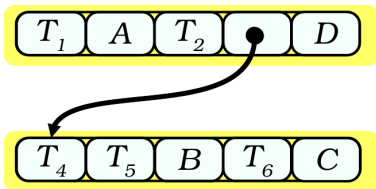
```
void Caja() {  
    Base();  
    glPushMatrix();  
        glMaterialfv( GL_FRONT, GL_AMBIENT_AND_DIFFUSE, blue );  
        glTranslatef(-2.5, 5, 0);  
        glRotatef(30, 0, 0, 1);  
    Tapa();  
    glPopMatrix();  
    glPushMatrix();  
        glMaterialfv( GL_FRONT, GL_AMBIENT_AND_DIFFUSE, red);  
        glTranslatef(2.5, 5, 0);  
        glRotatef(-40, 0, 0, 1);  
        glTranslatef(-2.5, 0, 0);  
    Tapa();  
    glPopMatrix();  
}
```

# Implementación en OpenGL





# Implementación en OpenGL



El código es secuencial =>  
a OpenGL le pasamos el  
recorrido del grafo

Transformación  $T_1$

Dibujar A

Transformación  $T_2$

**glPushMatrix()**

Transformación  $T_4$

Transformación  $T_5$

Dibujar B

Transformación  $T_6$

Dibujar C

**glPopMatrix()**

Dibujar D

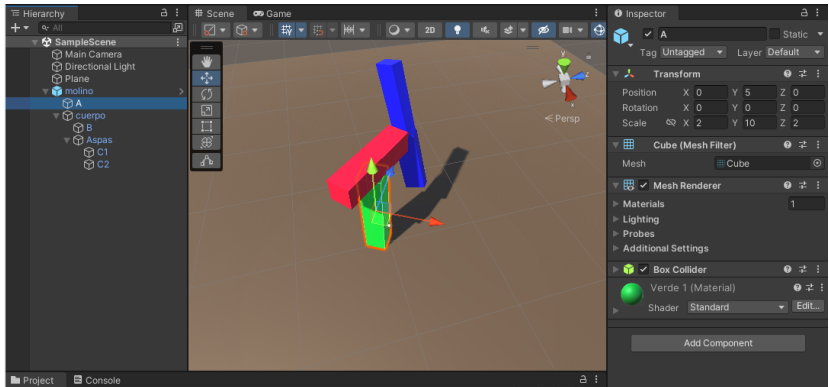
# Grafos de escena en Unity

Los nodos contienen:

- Un objeto
- Su transformación geométrica ( $S, R, T$ , en este orden)
- Opcionalmente pueden tener una lista de nodos hijos.

Para concatenar transformaciones se utilizan nodos sin geometría (*empty*).

# Grafos de escena en Unity



# Diseño de grafos de escena

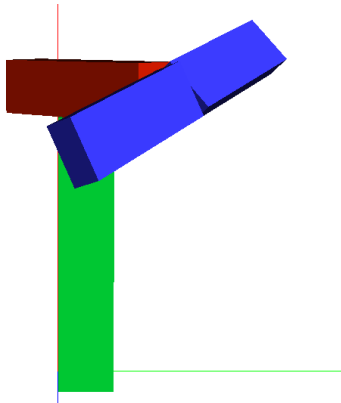
# Diseño de grafos de escena

- 1 Descomponer la escena recursivamente en componentes mas simples. Este proceso genera el esqueleto del grafo de escena.
- 2 Decidir las primitivas y transformaciones geométricas que se deben incluir en cada nodo.
- 3 Implementación del modelo.

# Descomposición de la escena

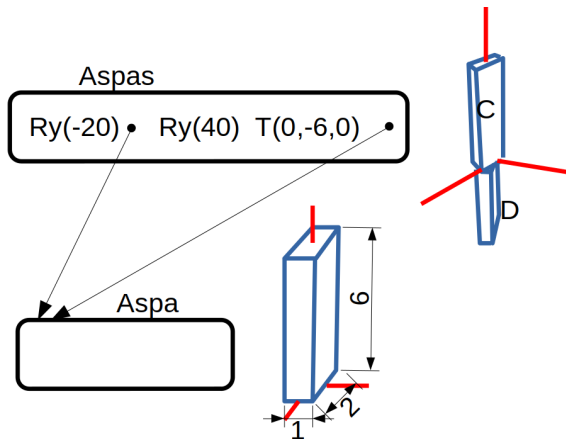
- Reutilización de componentes.
- Las ramas del grafo comparten transformaciones geométricas.

Hacer un boceto de cada nodo



# Diseño de los nodos

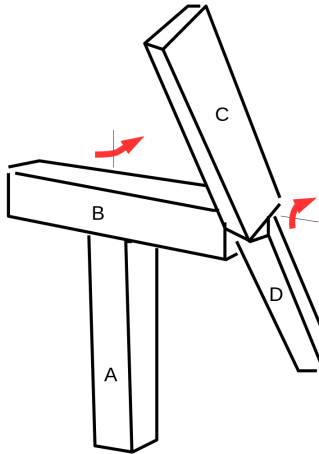
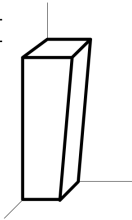
Diseñamos cada nodo mirando solo su boceto y el de sus hijos



# Ejemplo de diseño: Molino

Las piezas  
están creadas  
con un vértice  
en el origen y  
las siguientes  
dimensiones

A: 2x10x2  
B: 2X6X2  
C: 2x6x1  
D: 2x6x1

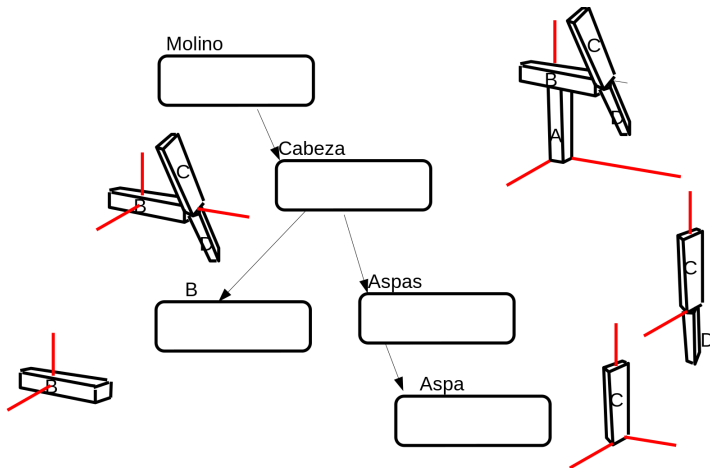


B Gira sobre el eje  
vertical

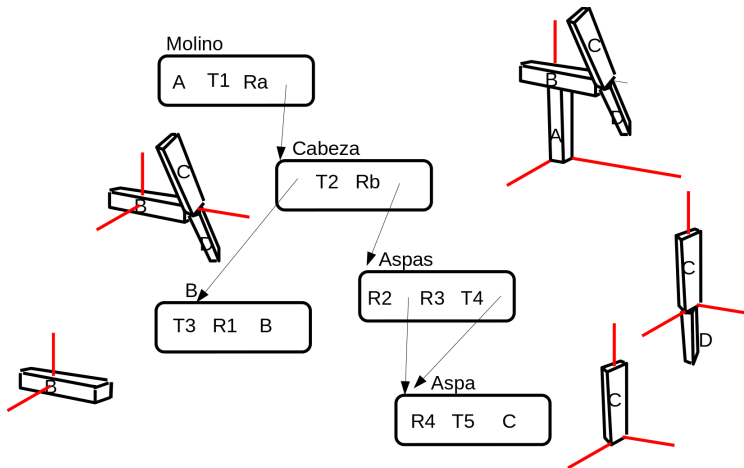
C y D giran sobre el  
eje horizontal de B



# Ejemplo de diseño: Molino



# Ejemplo de diseño: Molino



# Ejemplo de diseño: Molino

T1: 1,11,1

Ra: Giro eje y

T2: 3.5,0,0

Rb: Giro eje x

T3: -3, 1, -1

R1: -90 eje z

R2: -20 eje y

R3: 40 eje y

T4: 0,-6,0

R4: y, 90°

T5: -0.5, 0, -1

Molino

A T1 Ra

Cabeza

T2 Rb

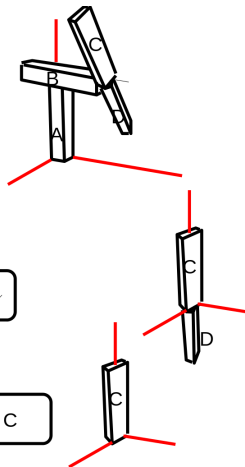
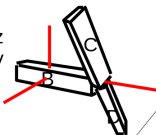
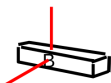
Aspas

R2 R3 T4

Aspa

R4 T5 C

B  
T3 R1 B



# Ejemplo de diseño: Molino

```
void A() {  
    glMaterialfv( GL_FRONT, GL_AMBIENT_AND_DIFFUSE, green );  
    box(2,10,2);  
}  
  
void B() {  
    glMaterialfv( GL_FRONT, GL_AMBIENT_AND_DIFFUSE, red );  
    glTranslatef(-3,1,-1);  
    glRotatef(-90,0,0,1);  
    box(2,6,2);  
}  
  
void C() {  
    glMaterialfv( GL_FRONT, GL_AMBIENT_AND_DIFFUSE, blue );  
    box(2,6,1);  
}  
  
void Aspa() {  
    glRotatef(90,0,1,0);  
    glTranslatef(-1,0,-0.5);  
    C();  
}
```

# Ejemplo de diseño: Molino

```
void Aspas() {  
    glRotatef(-40, 0, 1, 0);  
    glPushMatrix();  
        Aspa();  
    glPopMatrix();  
    glRotatef(20, 0, 1, 0);  
    glTranslatef(0, -6, 0);  
    Aspa();  
}  
  
void Cabeza() {  
    glPushMatrix();  
        B();  
    glPopMatrix();  
    glTranslatef(4, 0, 0);  
    glRotatef(angle2, 1, 0, 0);  
    glPushMatrix();  
        Aspas();  
    glPopMatrix();  
}
```

# Ejemplo de diseño: Molino

```
void Molino() {  
    glPushMatrix();  
        A();  
    glPopMatrix();  
    glTranslatef(1,11,1);  
    glRotatef(angle1,0,1,0);  
    glPushMatrix();  
        Cabeza();  
    glPopMatrix();  
}
```