

# Transformações Geométricas

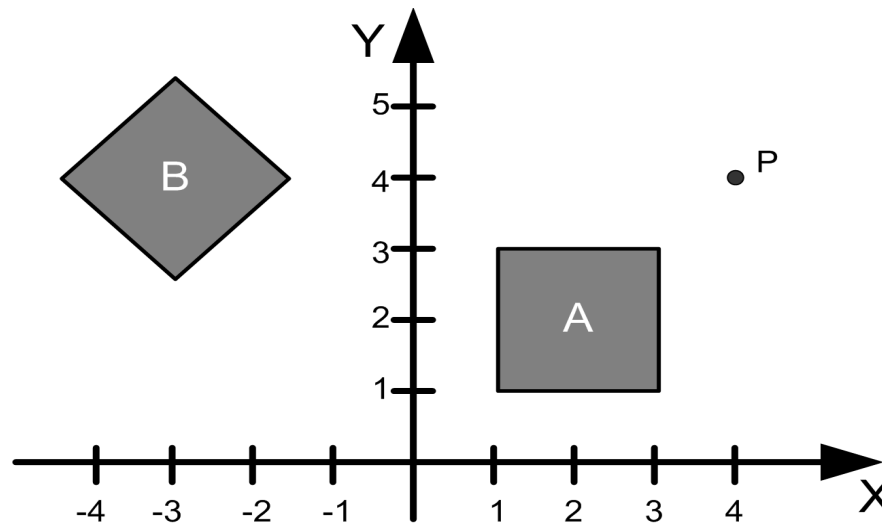
---

Composição, Mudança de Coordenadas e Grafos de Cena

# Questão 2, exame de 20/07/11

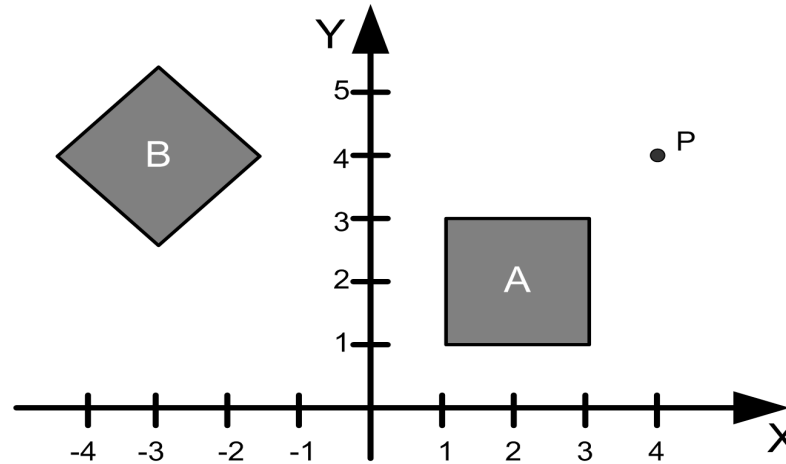
**[3.0v]** Considere os polígonos “A” e “B” e o ponto “P” representados na figura abaixo.

- a) calcule a matriz final que representa a transformação a aplicar ao polígono “A” para que este se transforme no polígono “B”.
- b) Calcule as coordenadas do ponto “Q” resultante da aplicação da transformação referida anteriormente a “P”



# Questão 2, exame de 20/07/11

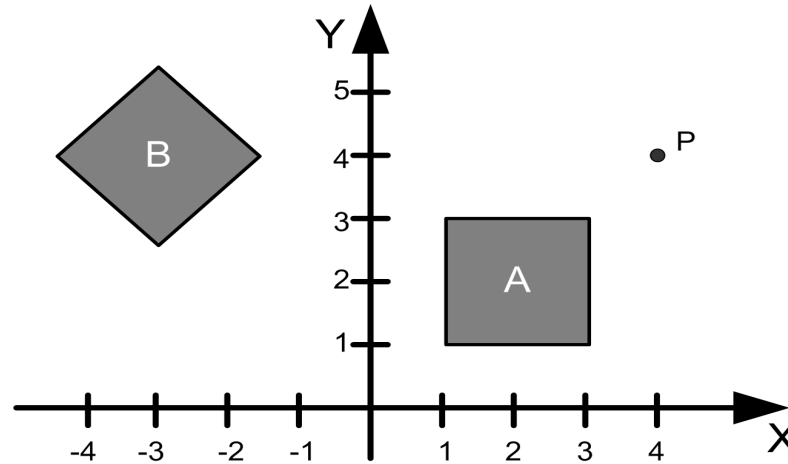
**[2.0v]** (...) calcular a matriz de transformação (...)



?

# Questão 2, exame de 20/07/11

**[1.0v]** (...) calcular as coordenadas de “Q” (...)



?

Transformações Geométricas

# Transformações em 3D

## Transformações Elementares 3D

# Translação

$$M_T = T(dx, dy, dz) = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

# Transformações Elementares 3D

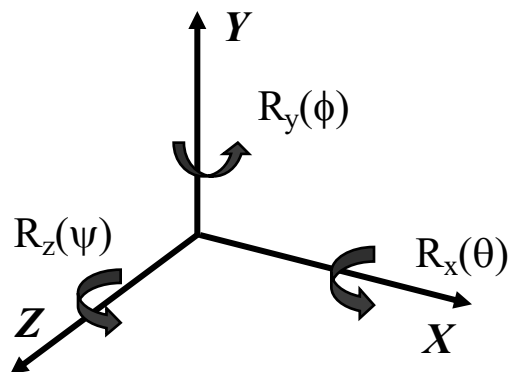
## Escala

$$M_S = S(Sx, Sy, Sz) = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

# Transformações Elementares 3D

## Rotação



$R_z(\psi)$ : em torno do eixo dos ZZ

$$\begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$R_x(\theta)$ : em torno do eixo dos XX

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

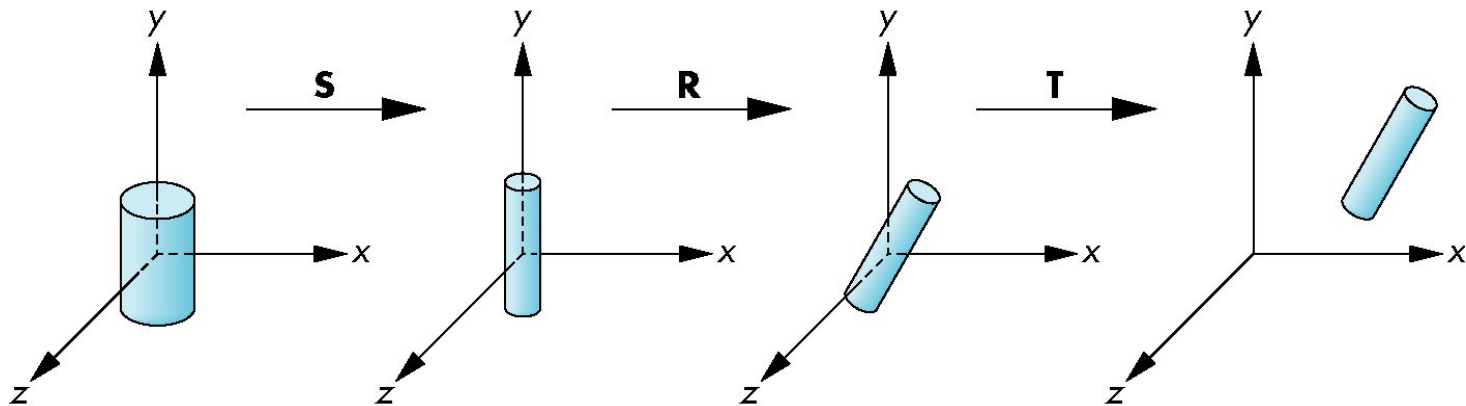
$R_y(\phi)$ : em torno do eixo dos YY

$$\begin{bmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



# Instâncias de Objectos

Cada instância de um objecto na cena  
conjunto de Transformações Geométricas



Transformações Geométricas

# Matriz de Transformação Composta

# Matriz de Transformação Composta

Qualquer sequência de  
Translações, Rotações e Escalas

Pode ser representada numa **única matriz**:

$$M = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# Composição de Transformações

Coordenadas homogêneas →

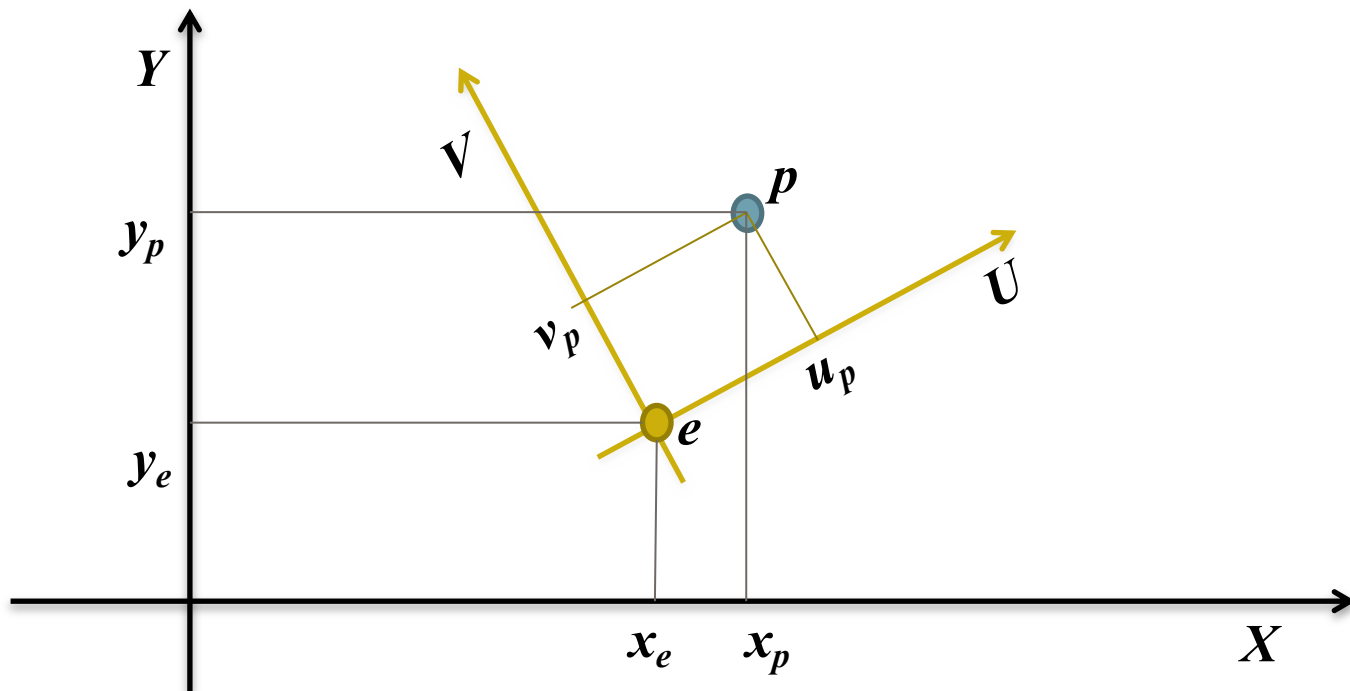
Composição de Transformações  $\Leftrightarrow$  Produto de Matrizes

$$\begin{aligned} P &\xrightarrow{T_1} P' \xrightarrow{T_2} P'' & P' &= T_1 \cdot P \quad \text{e} \quad P'' = T_2 \cdot P' \\ & & P'' &= T_2 \cdot (T_1 \cdot P) \\ & & P'' &= (T_2 \circ T_1) \cdot P = (T_2 \cdot T_1) \cdot P \end{aligned}$$

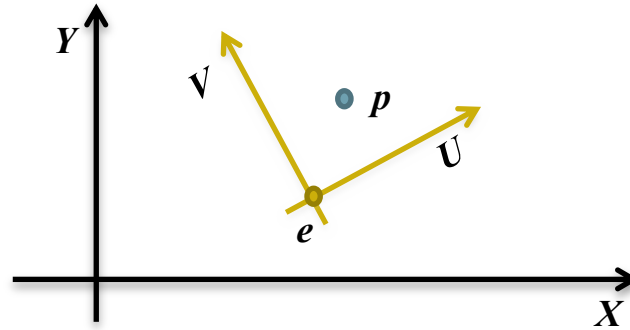
Transformações associam-se

- Da direita para a esquerda
- Pela ordem inversa de aplicação

# Mudança do Sistema de Coordenadas



# Mudança do Sistema de Coordenadas

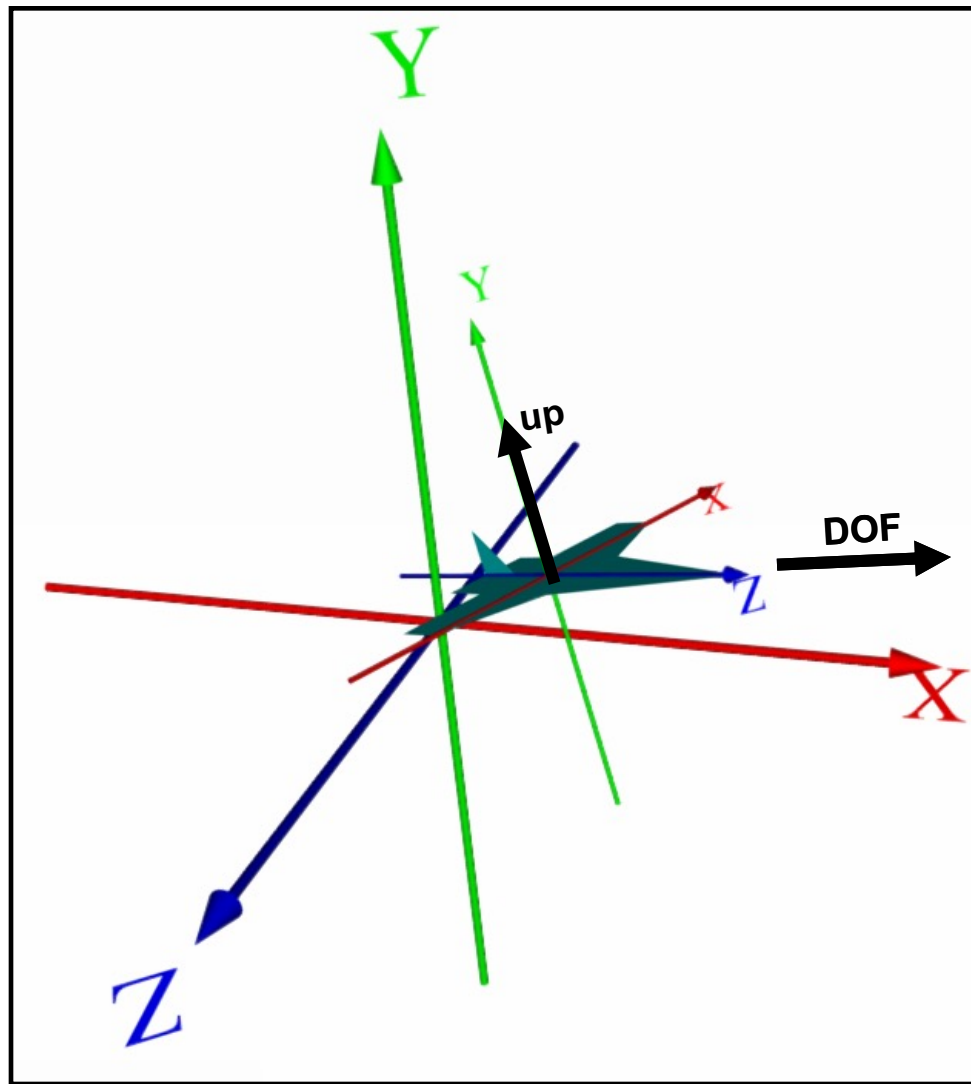
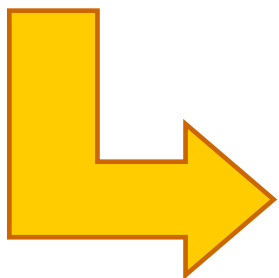
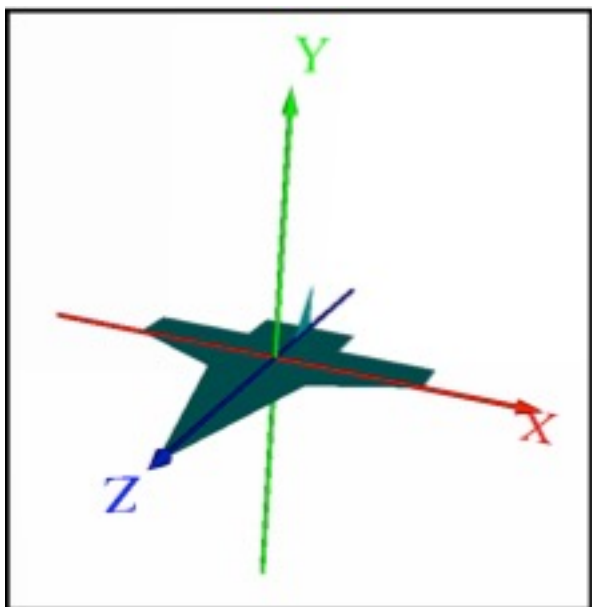


$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_e \\ 0 & 1 & y_e \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_x & v_x & 0 \\ u_y & v_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix} = \begin{bmatrix} u_x & v_x & x_e \\ u_y & v_y & y_e \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix}$$

$$p_{xy} = \begin{bmatrix} u & v & e \\ 0 & 0 & 1 \end{bmatrix} p_{uv}$$

# Composição de Transformações em 3D

## Exemplo Prático

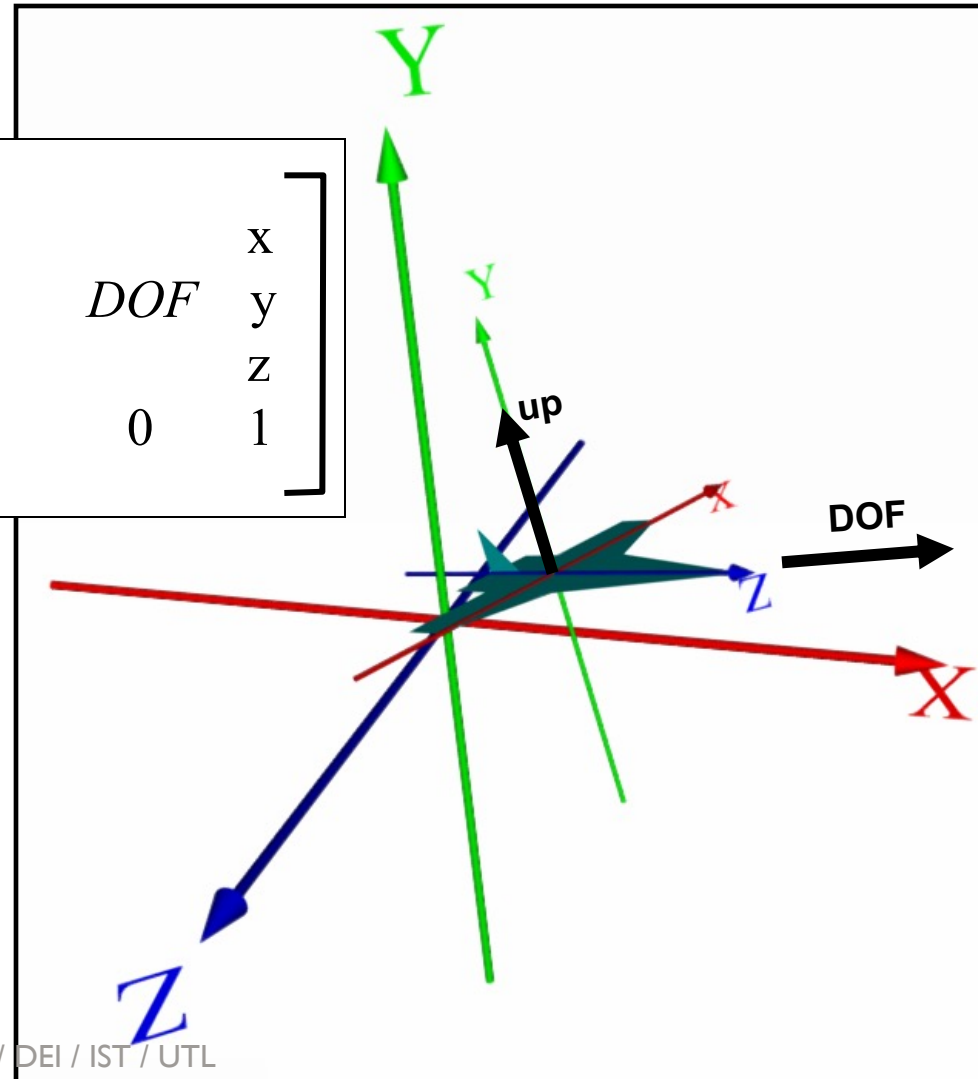


# Composição de Transformações em 3D

## Exemplo Prático

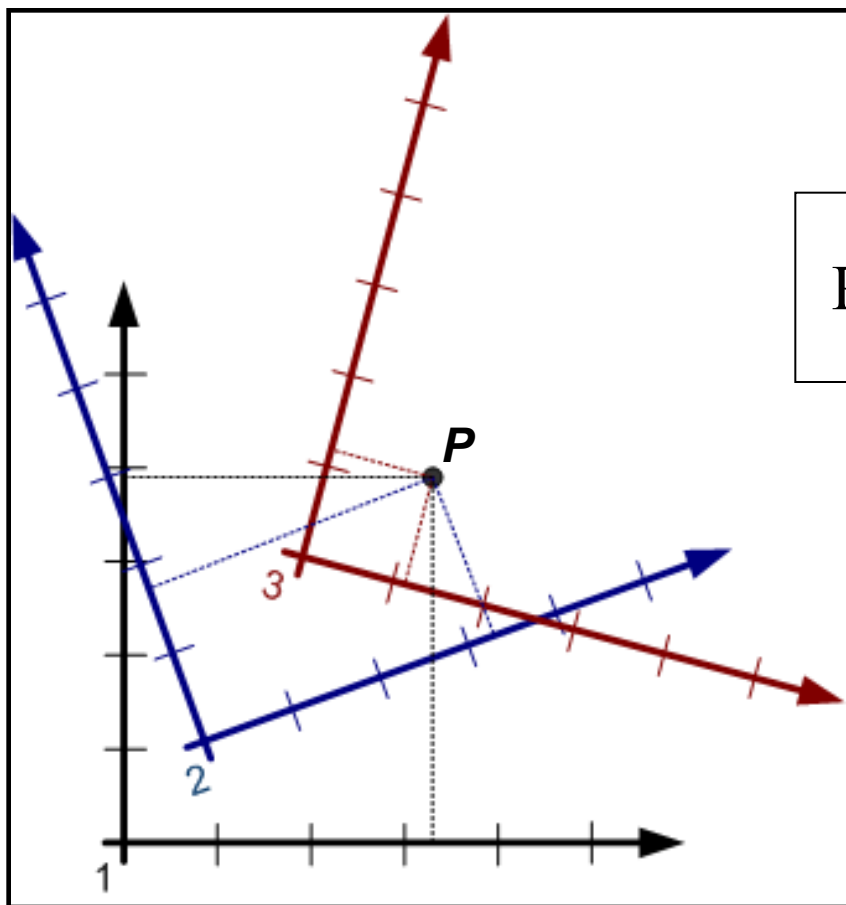
Como calcular *MTC*?

$$MTC = \begin{bmatrix} up \times DOF & DOF \times (up \times DOF) & DOF \\ 0 & 0 & 0 \end{bmatrix} \begin{matrix} x \\ y \\ z \\ 1 \end{matrix}$$





# Mudança de Sistema de Coordenadas



$$P^{(2)} = M_{2 \leftarrow 3} \cdot P^{(3)} \quad P^{(1)} = M_{1 \leftarrow 2} \cdot P^{(2)}$$

$$M_{1 \leftarrow 3} = M_{1 \leftarrow 2} \cdot M_{2 \leftarrow 3}$$

# Transformações Geométricas

## Mudança de Sistema de Coordenadas

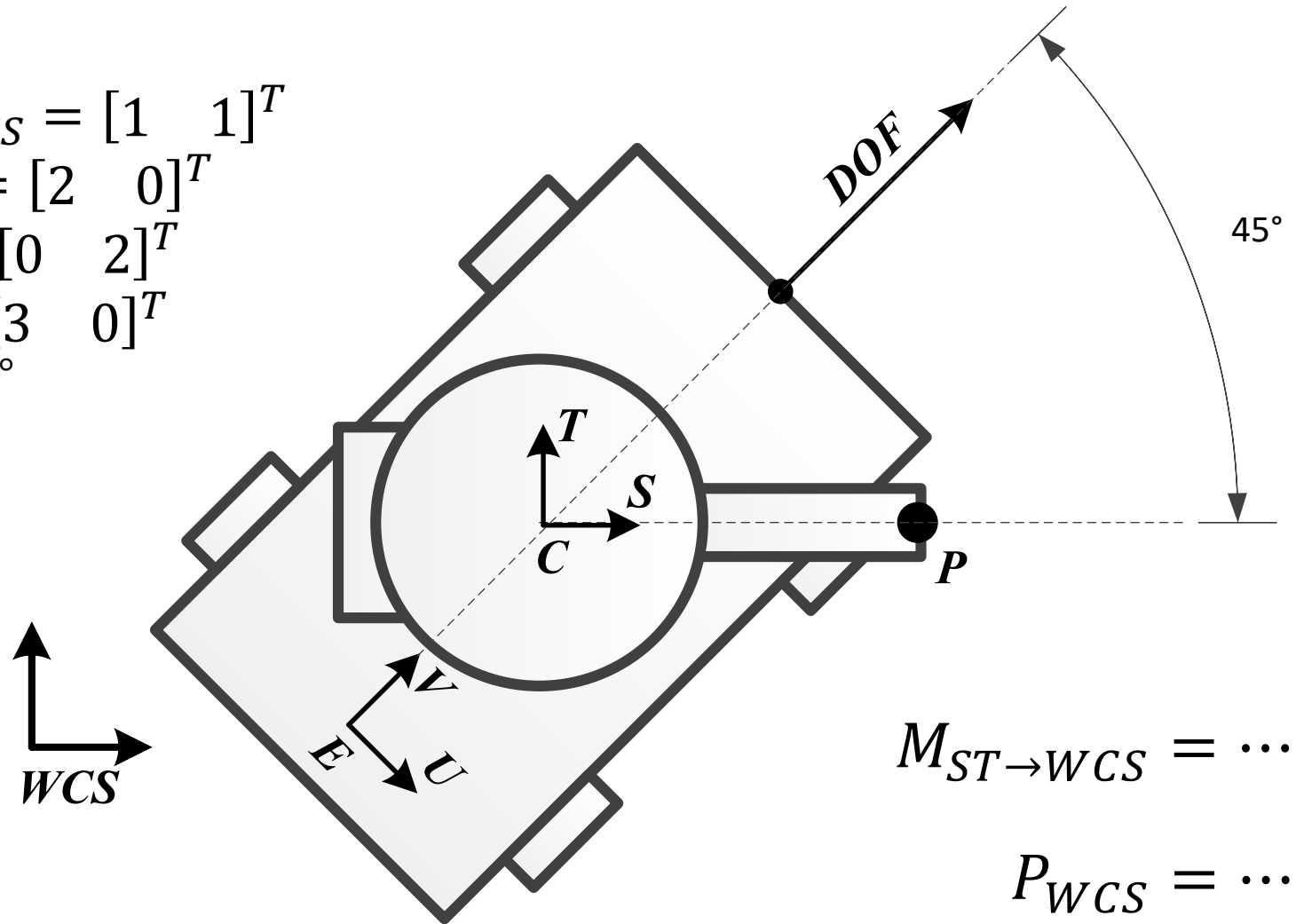
$$DOF_{WCS} = [1 \quad 1]^T$$

$$E_{WCS} = [2 \quad 0]^T$$

$$C_{UV} = [0 \quad 2]^T$$

$$P_{ST} = [3 \quad 0]^T$$

$$\theta = 45^\circ$$



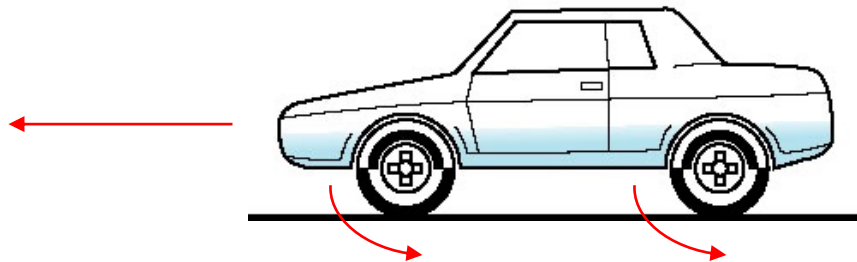
# Grafo de Cena

Motivação:

# Modelo de um carro

Chassis e quatro rodas

Movimento do carro e rotação das rodas relacionados



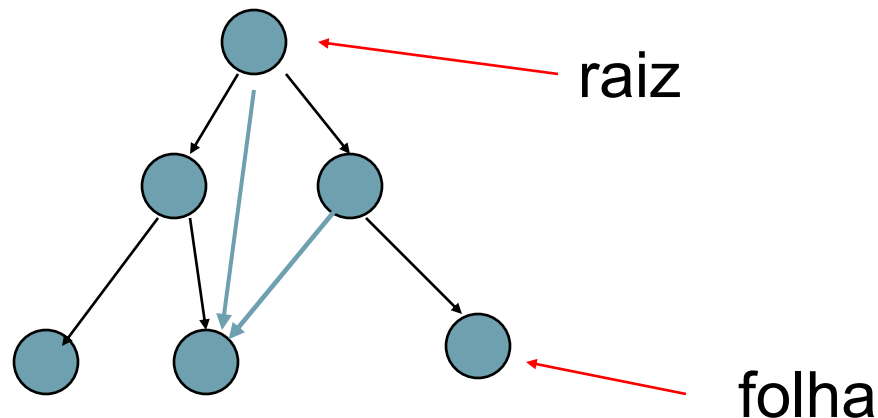
# Grafo de Cena

## Grafo Acíclico Orientado

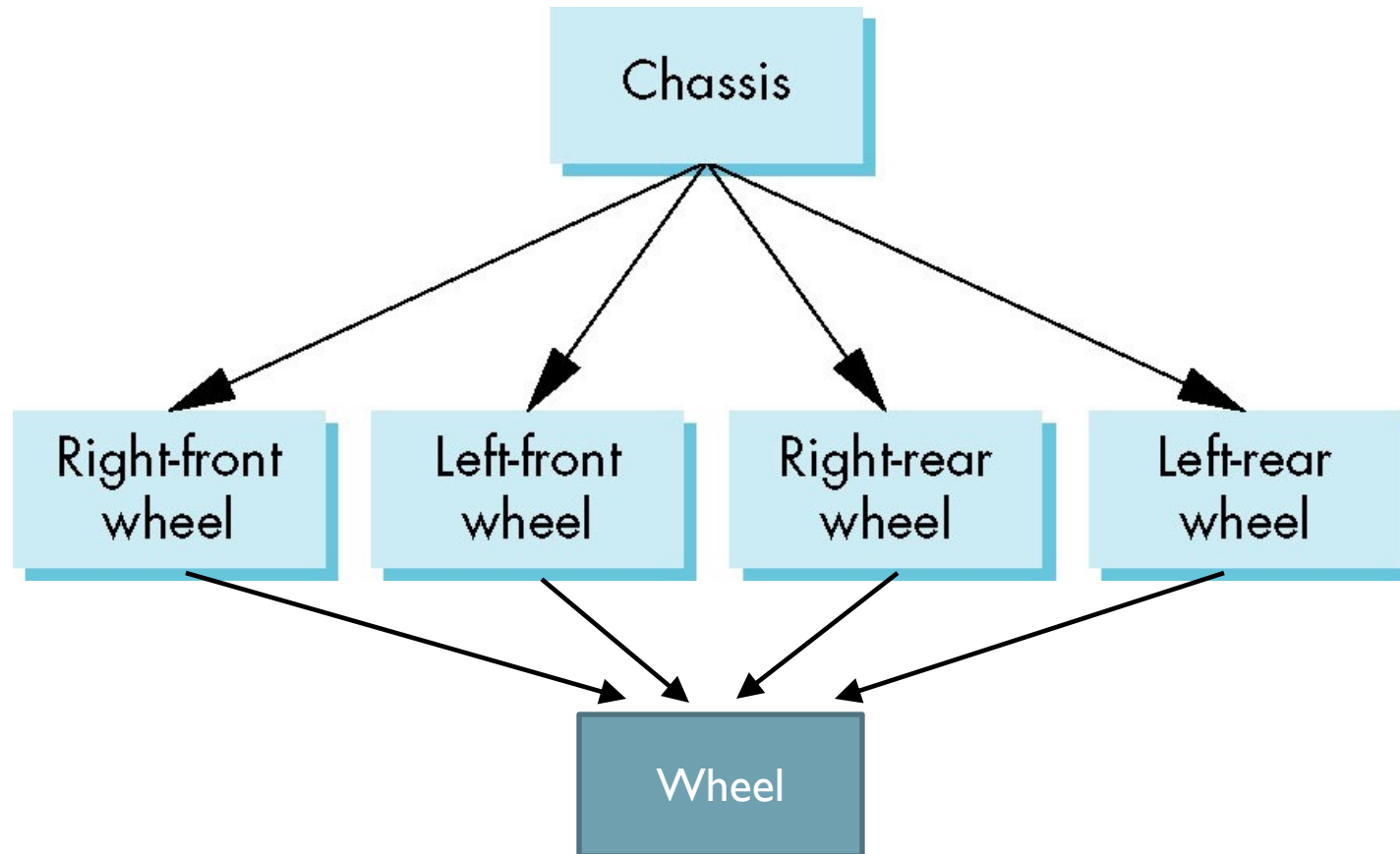
Cada nó (excepto raiz) **pode ter um ou mais *parents***

Pode ter inúmeros filhos

Folha não tem filhos (usualmente é uma primitiva)



# Exemplo: Grafo do Carro

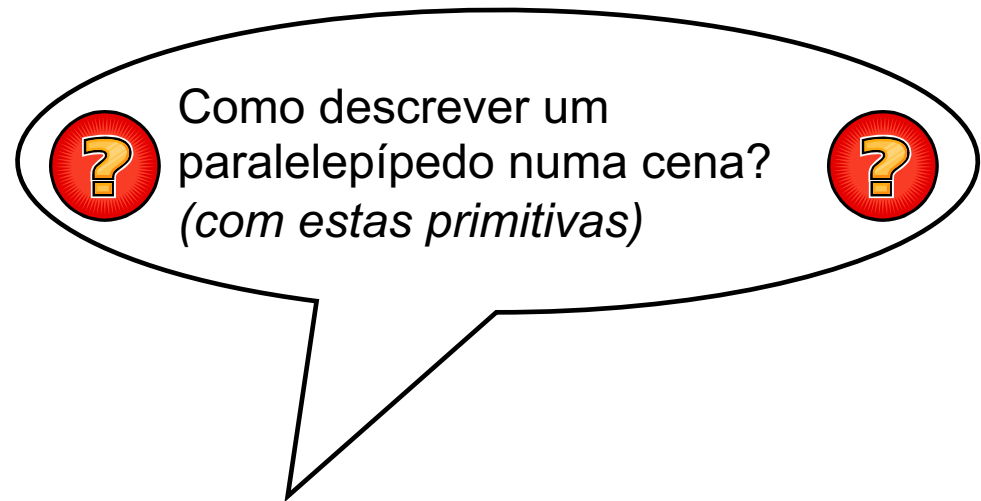


# Grafo de Cena

- Cenas 3D armazenadas em DAG

- Grafo de Cena

- Java3D
    - VRML
    - OpenSceneGraph
    - OpenSG



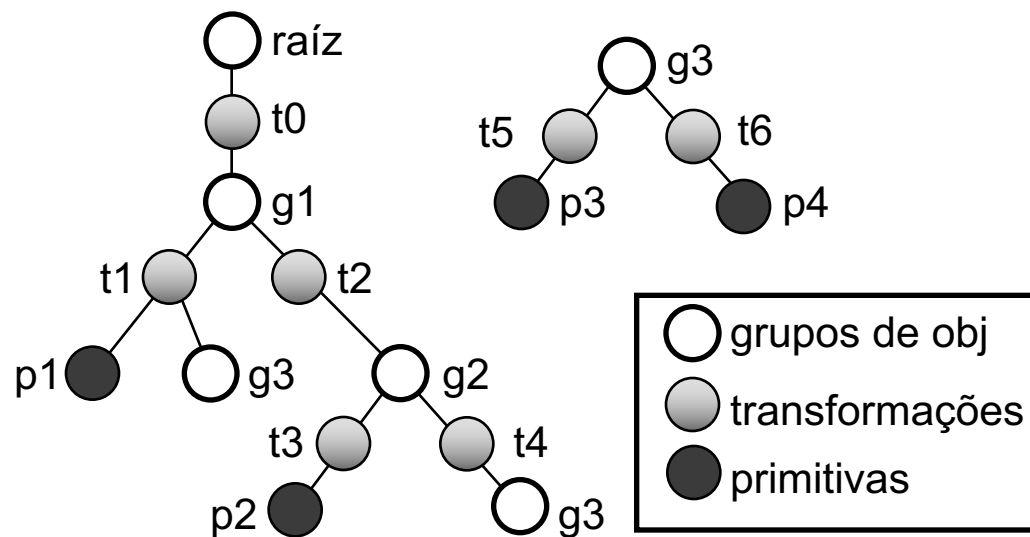
- Grafo de Cena contém

- Objectos (primitivas gráficas)

- Cubos, esferas, cones, superfícies,....
    - Atributos e Transformações

# Transformações em Grafos de Cena

## Exemplo de um Grafo de Cena

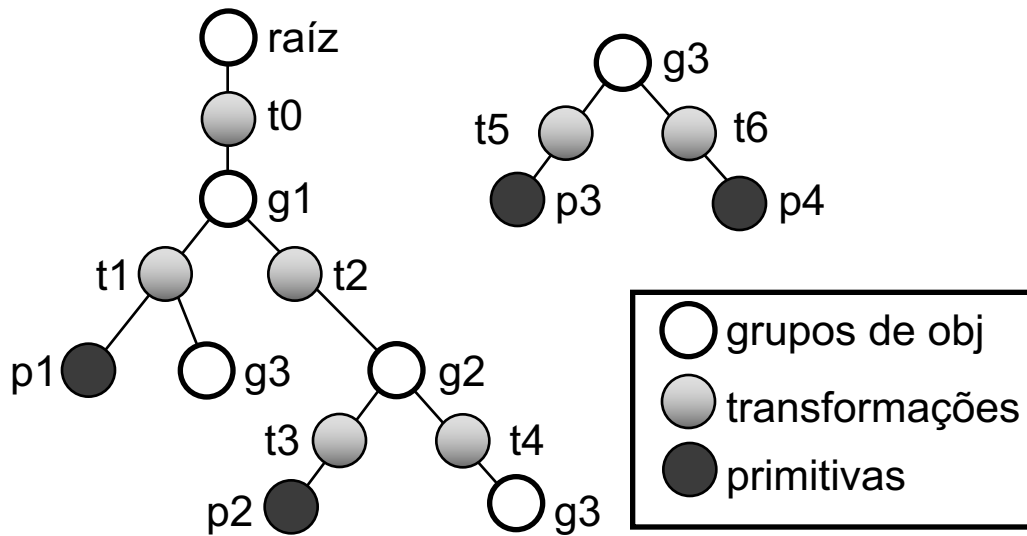


- Neste grafo de cena
  - A transformação **t0** afecta **todos** os objectos
  - Enquanto **t2** só afecta **p2** e uma instância do grupo **g3**
    - **t2** não afecta **p1** e a outra instância de **g3**



# Múltipla Instanciação

## Exemplo de um Grafo de Cena

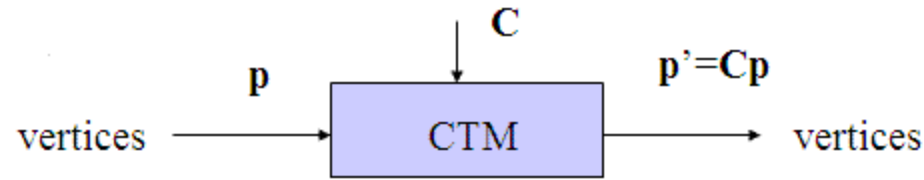


- Múltiplas instâncias de uma *sub-árvore*
  - Podem utilizar-se várias
  - É necessário *definir* antes de *instanciar*
    - Mais simples de concretizar

# Transformações Hierárquicas (1/2)

## Matriz de Transformação Corrente (CTM)

Matriz a aplicar a todos os vértices



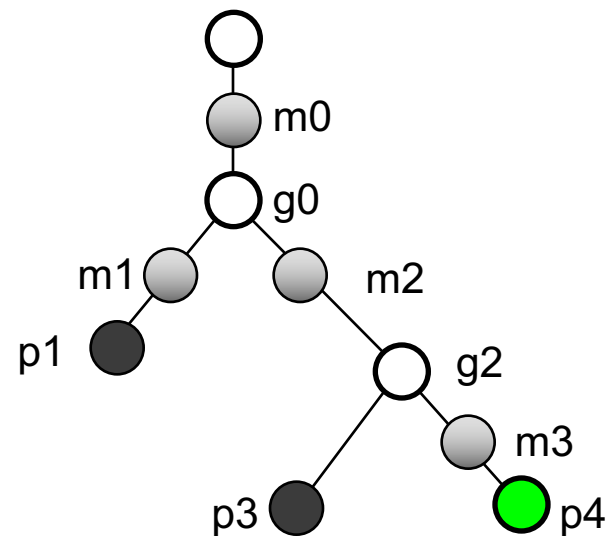
## Cálculo da CTM

Concatenação de **todas** as transformações em nós superiores no caminho

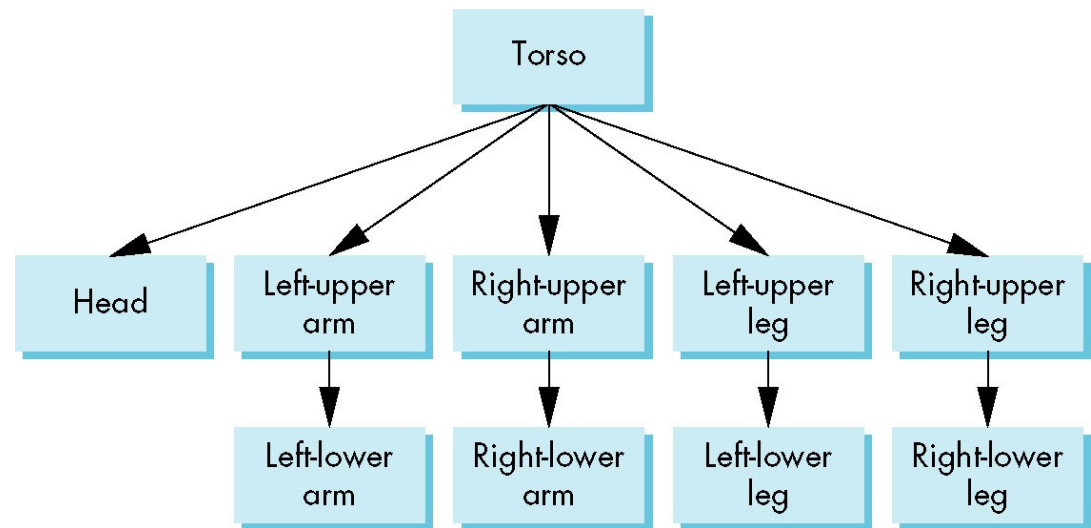
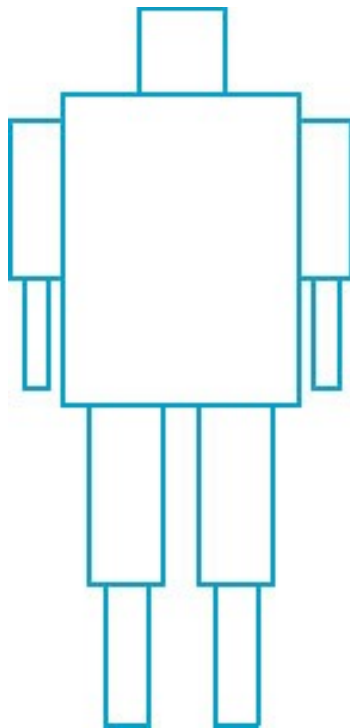
# Transformações Hierárquicas (2/2)

## Exemplo

- No grafo
  - para  $g_0$ , temos  $MTC = m_0$
  - para  $p_1$ ,  $MTC = m_0 * m_1$
  - para  $p_3$ ,  $MTC = m_0 * m_2$
  - para  $p_4$ ,  $MTC = ?$
- Em que
  - $m_i$  - matriz de transformação
  - $p_i$  - primitiva associada ao nó  $i$
  - $g_i$  - sub-árvore com raiz em  $i$



# Grafo para um Robot

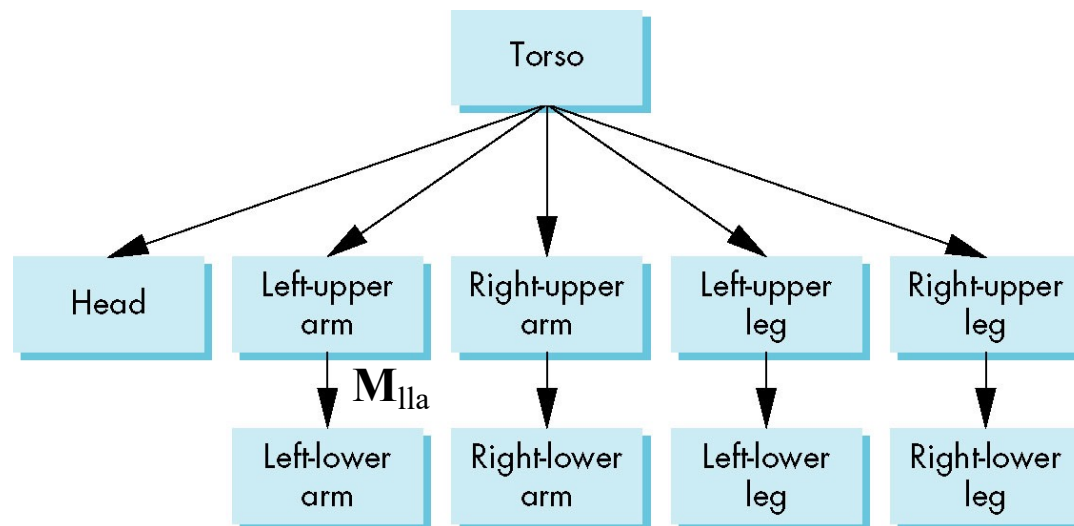


# Realizar o Robot

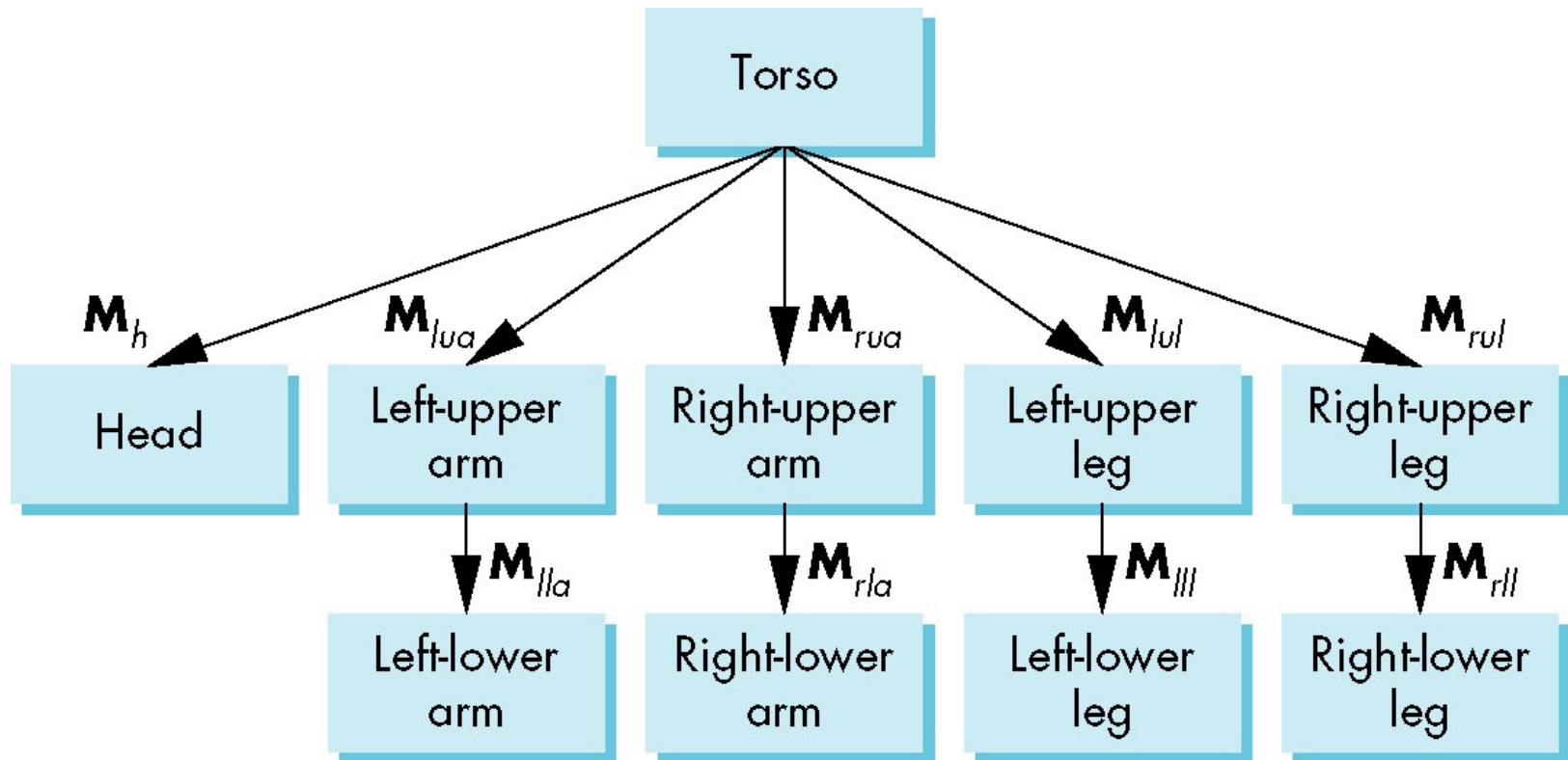
Primitivas Geométricas Simples (e.g. *boxes*)

Matrizes descrevem posição e orientação do nó relativamente ao seu *pai*

$M_{lla}$  posiciona left lower arm relativamente ao left upper arm



# Grafo com Matrizes

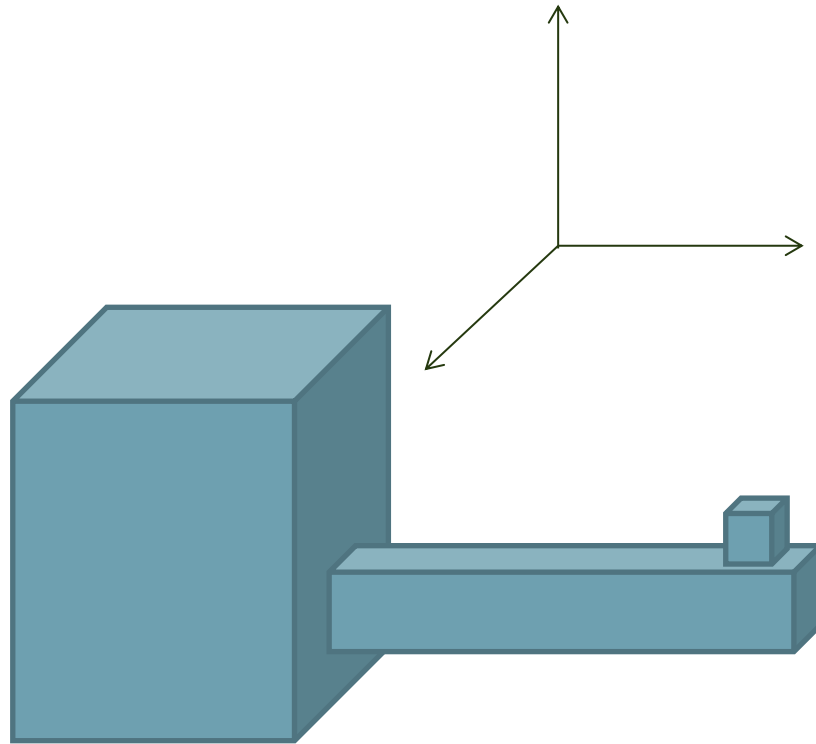


Transformações Geométricas

# Construção de uma cena em Three.js

# Modelação

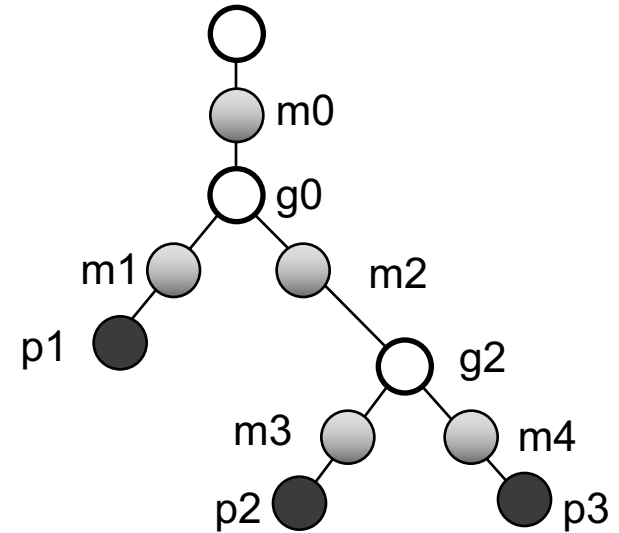
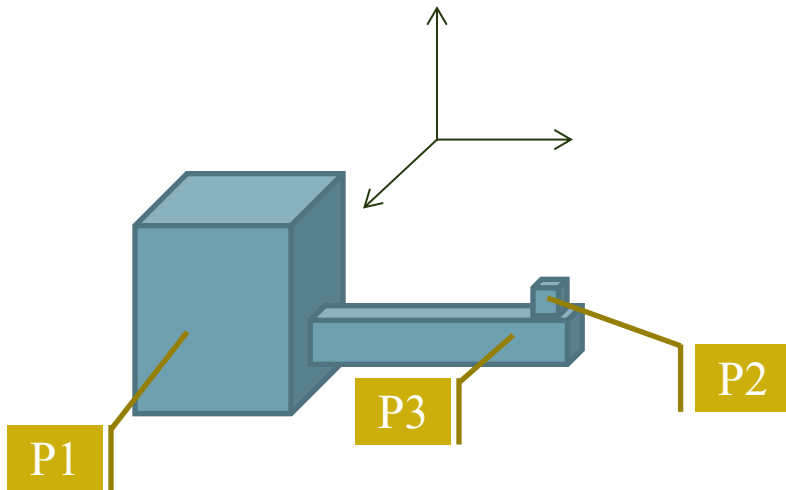
Um exemplo





# Modelação

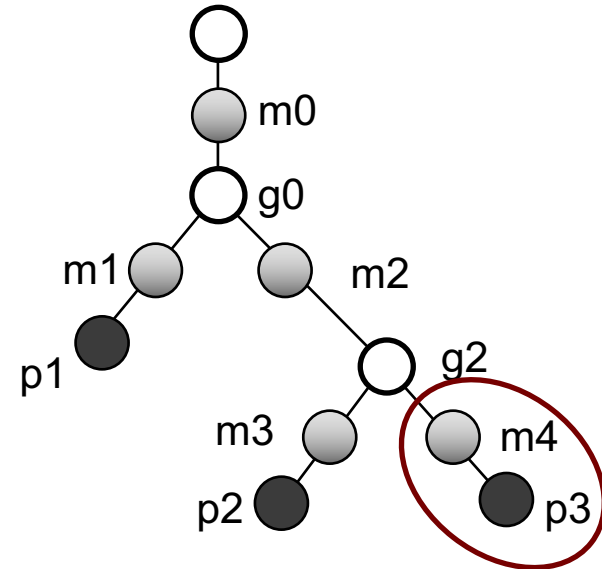
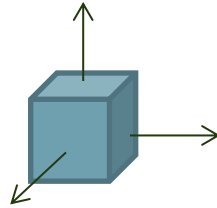
Um exemplo



- Primitivas (*P1*, *P2* e *P3*)
  - Cubo Unitário
- Transformações
  - Translação e Escala

# Modelação

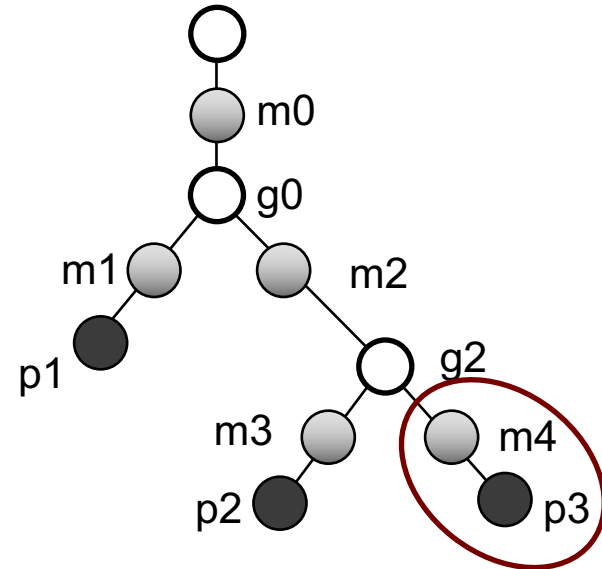
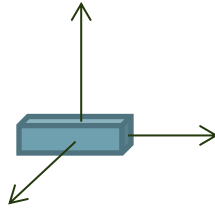
Um exemplo



- P3 = Cubo Unitário
- M4 = Escala

# Modelação

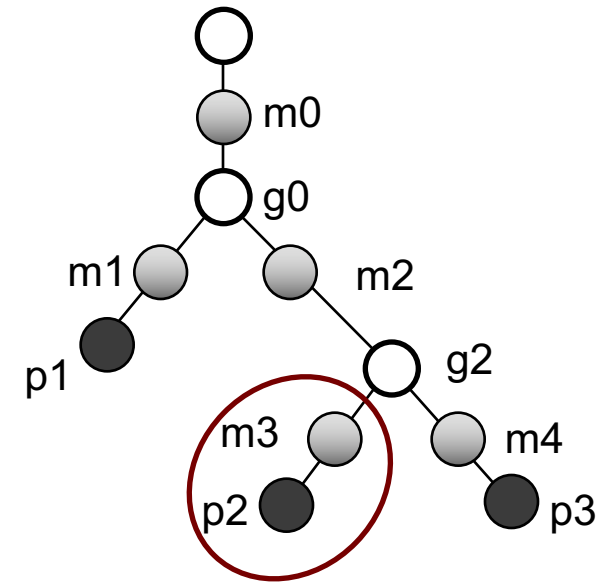
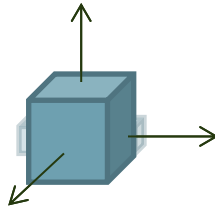
Um exemplo



- P3 = Cubo Unitário
- M4 = Escala

# Modelação

Um exemplo

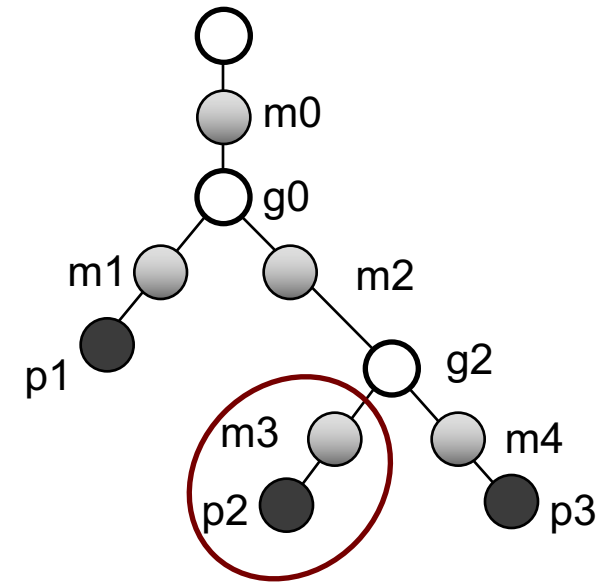
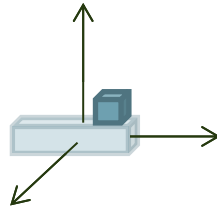


P2 = Cubo Unitário

M3 = Escala seguida de Translação

# Modelação

Um exemplo

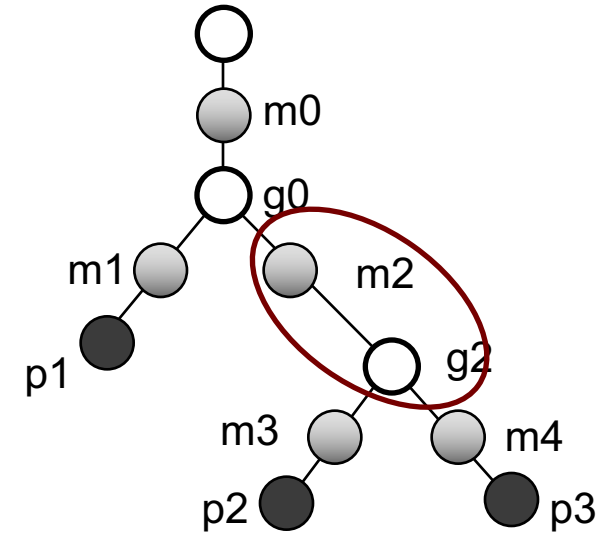
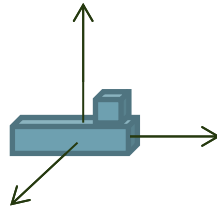


P2 = Cubo Unitário

M3 = Escala seguida de Translação

# Modelação

Um exemplo

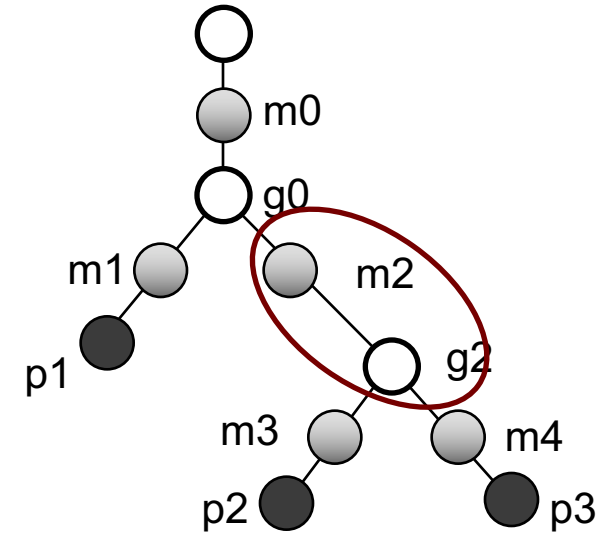
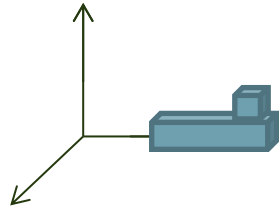


$G2$  = composição de duas primitivas

$M2$  = Translação

# Modelação

Um exemplo

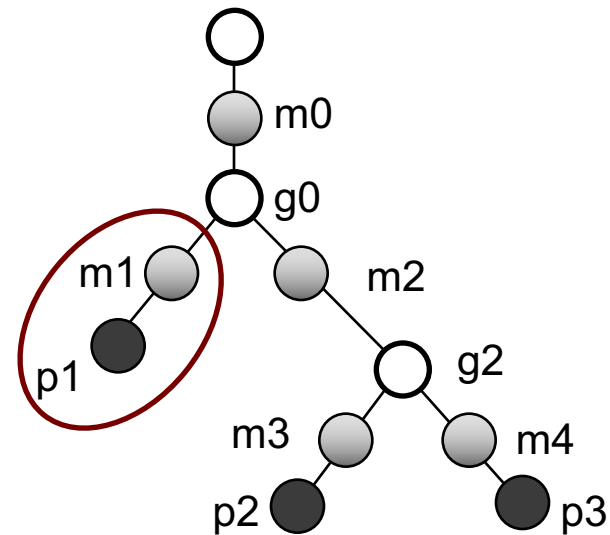
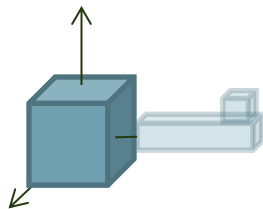


G2 = composição de duas primitivas

M2 = Translação

# Modelação

Um exemplo

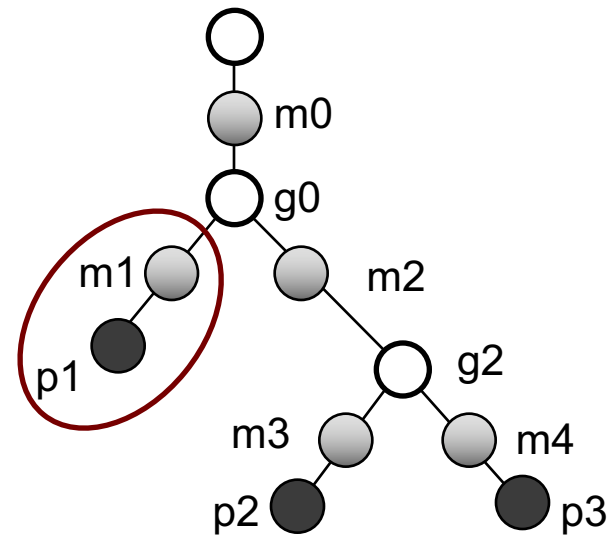
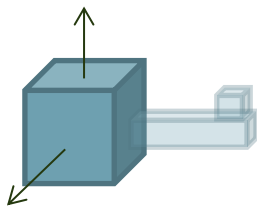


- PI = Cubo Unitário
- MI = Escala



# Modelação

Um exemplo

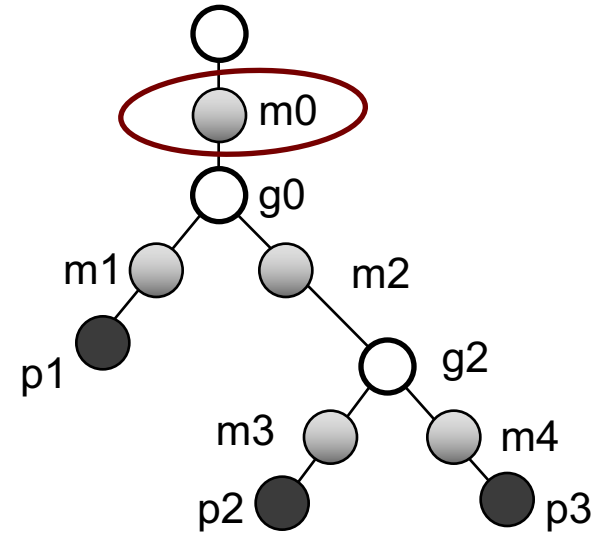
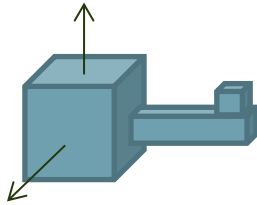


PI = Cubo Unitário

MI = Escala

# Modelação

Um exemplo

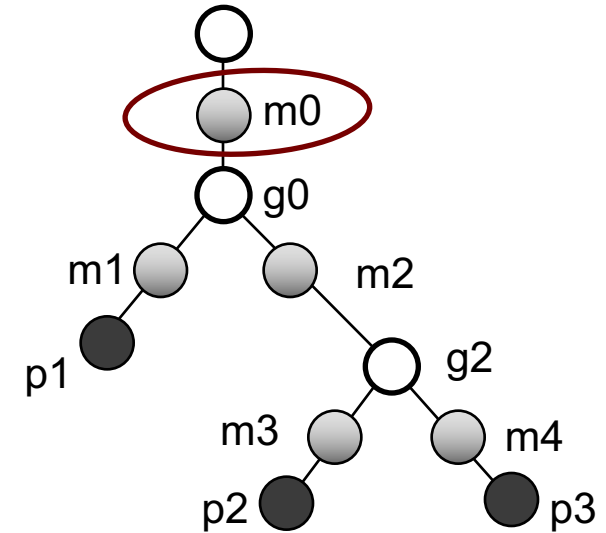
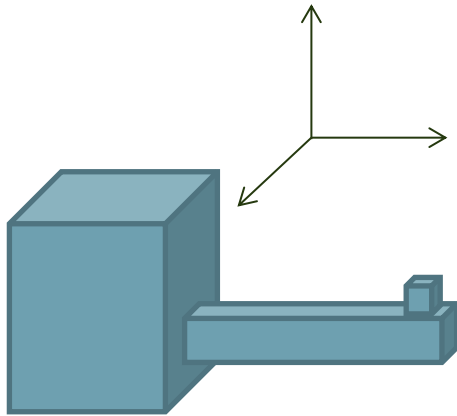


G0 = Objecto Composto

M0 = Translação

# Modelação

Um exemplo

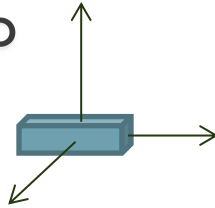


G0 = Objecto Composto

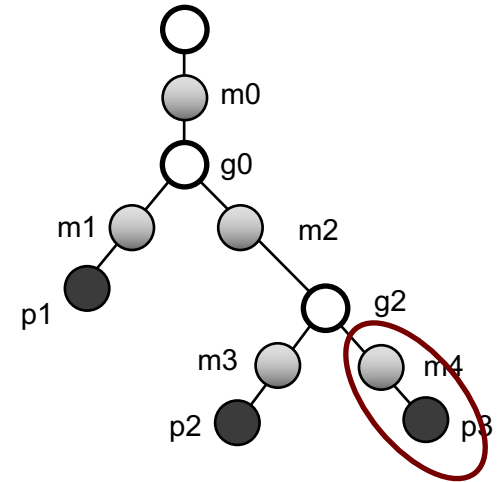
M0 = Translação

# Modelação

Um exemplo

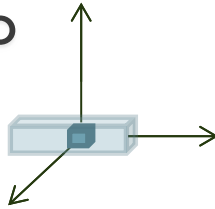


```
var g0, g2, p1, p2, p3;  
p3 = new THREE.Mesh(new THREE.BoxGeometry(1.2, 0.2, 0.2), mat);
```

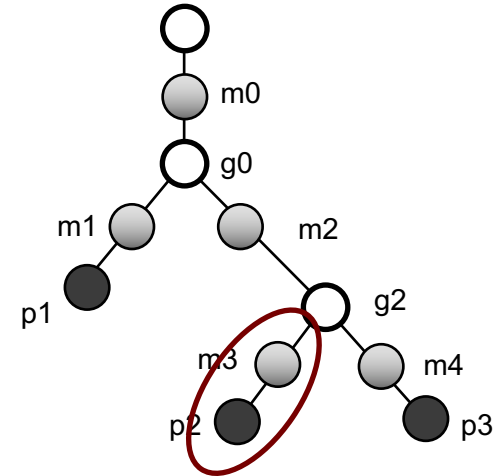


# Modelação

Um exemplo

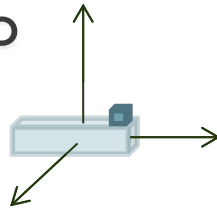


```
var g0, g2, p1, p2, p3;  
p3 = new THREE.Mesh(new THREE.BoxGeometry(1.2, 0.2, 0.2), mat);  
p2 = new THREE.Mesh(new THREE.BoxGeometry(0.1, 0.1, 0.1), mat);
```

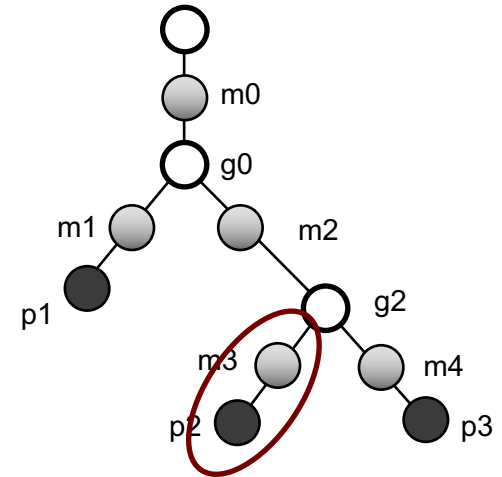


# Modelação

Um exemplo

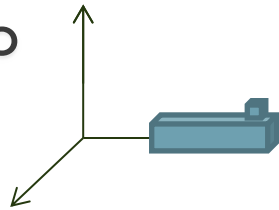


```
var g0, g2, p1, p2, p3;  
  
p3 = new THREE.Mesh(new THREE.BoxGeometry(1.2, 0.2, 0.2), mat);  
  
p2 = new THREE.Mesh(new THREE.BoxGeometry(0.1, 0.1, 0.1), mat);  
p2.position.set(0.5, 0.1, 0.0);
```

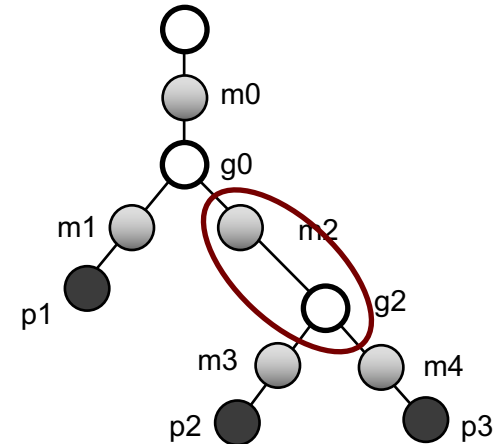


# Modelação

Um exemplo

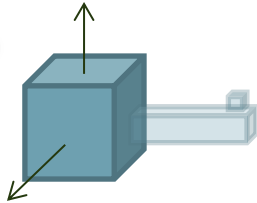


```
var g0, g2, p1, p2, p3;  
  
p3 = new THREE.Mesh(new THREE.BoxGeometry(1.2, 0.2, 0.2), mat);  
  
p2 = new THREE.Mesh(new THREE.BoxGeometry(0.1, 0.1, 0.1), mat);  
p2.position.set(0.5, 0.1, 0.0);  
  
g2 = new THREE.Object3D();  
g2.add(p3);  
g2.add(p2);  
g2.position.set(1.2, 0.0, 0.0);
```

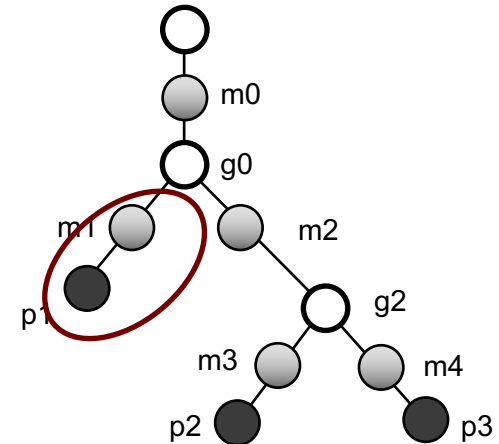


# Modelação

Um exemplo



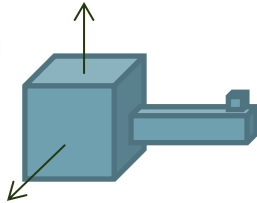
```
var g0, g2, p1, p2, p3;  
  
p3 = new THREE.Mesh(new THREE.BoxGeometry(1.2, 0.2, 0.2), mat);  
  
p2 = new THREE.Mesh(new THREE.BoxGeometry(0.1, 0.1, 0.1), mat);  
p2.position.set(0.5, 0.1, 0.0);  
  
g2 = new THREE.Object3D();  
g2.add(p3);  
g2.add(p2);  
g2.position.set(1.2, 0.0, 0.0);  
  
p1 = new THREE.Mesh(new THREE.BoxGeometry(1.5, 1.5, 1.5), mat);
```





# Modelação

Um exemplo



```
var g0, g2, p1, p2, p3;

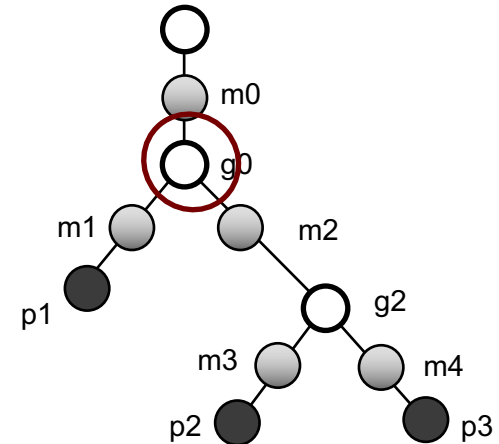
p3 = new THREE.Mesh(new THREE.BoxGeometry(1.2, 0.2, 0.2), mat);

p2 = new THREE.Mesh(new THREE.BoxGeometry(0.1, 0.1, 0.1), mat);
p2.position.set(0.5, 0.1, 0.0);

g2 = new THREE.Object3D();
g2.add(p3);
g2.add(p2);
g2.position.set(1.2, 0.0, 0.0);

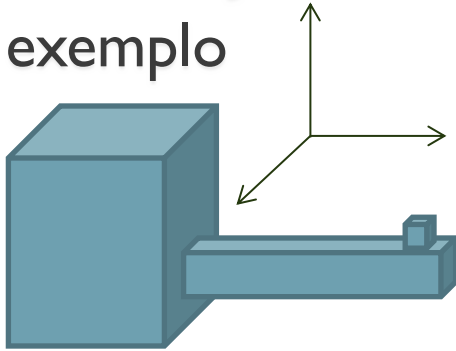
p1 = new THREE.Mesh(new THREE.BoxGeometry(1.5, 1.5, 1.5), mat);

g0 = new THREE.Object3D();
g0.add(p1);
g0.add(g2);
```



# Modelação

Um exemplo



```
var g0, g2, p1, p2, p3;

p3 = new THREE.Mesh(new THREE.BoxGeometry(1.2, 0.2, 0.2), mat);

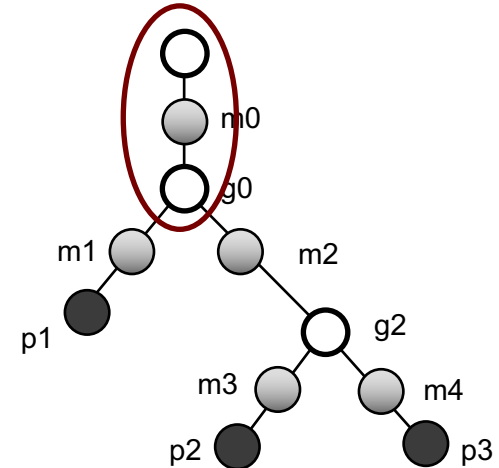
p2 = new THREE.Mesh(new THREE.BoxGeometry(0.1, 0.1, 0.1), mat);
p2.position.set(0.5, 0.1, 0.0);

g2 = new THREE.Object3D();
g2.add(p3);
g2.add(p2);
g2.position.set(1.2, 0.0, 0.0);

p1 = new THREE.Mesh(new THREE.BoxGeometry(1.5, 1.5, 1.5), mat);

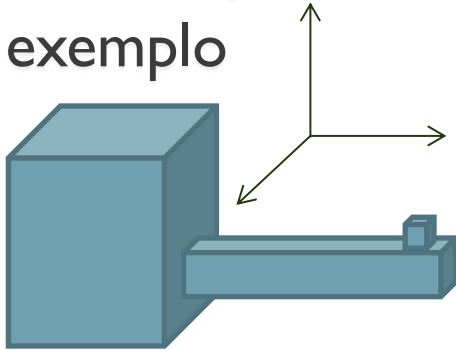
g0 = new THREE.Object3D();
g0.add(p1);
g0.add(g2);

g0.position.set(0.0, 0.0, 2.0);
scene.add(g0);
```



# Modelação

Um exemplo



```
var g0, g2, p1, p2, p3;

p3 = new THREE.Mesh(new THREE.BoxGeometry(1.2, 0.2, 0.2), mat);

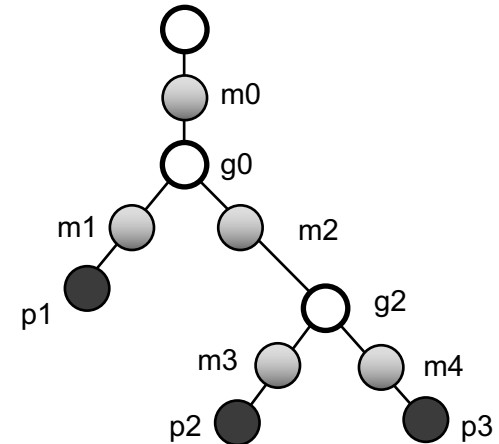
p2 = new THREE.Mesh(new THREE.BoxGeometry(0.1, 0.1, 0.1), mat);
p2.position.set(0.5, 0.1, 0.0);

g2 = new THREE.Object3D();
g2.add(p3);
g2.add(p2);
g2.position.set(1.2, 0.0, 0.0);

p1 = new THREE.Mesh(new THREE.BoxGeometry(1.5, 1.5, 1.5), mat);

g0 = new THREE.Object3D();
g0.add(p1);
g0.add(g2);

g0.position.set(0.0, 0.0, 2.0);
scene.add(g0);
```



# Hierarquia:

## Modular Swinging Robot vs Hover Robot

