



Módulo 9

Sistemas Gráficos e Interação

Instituto Superior de Engenharia do Porto

Filipe Pacheco

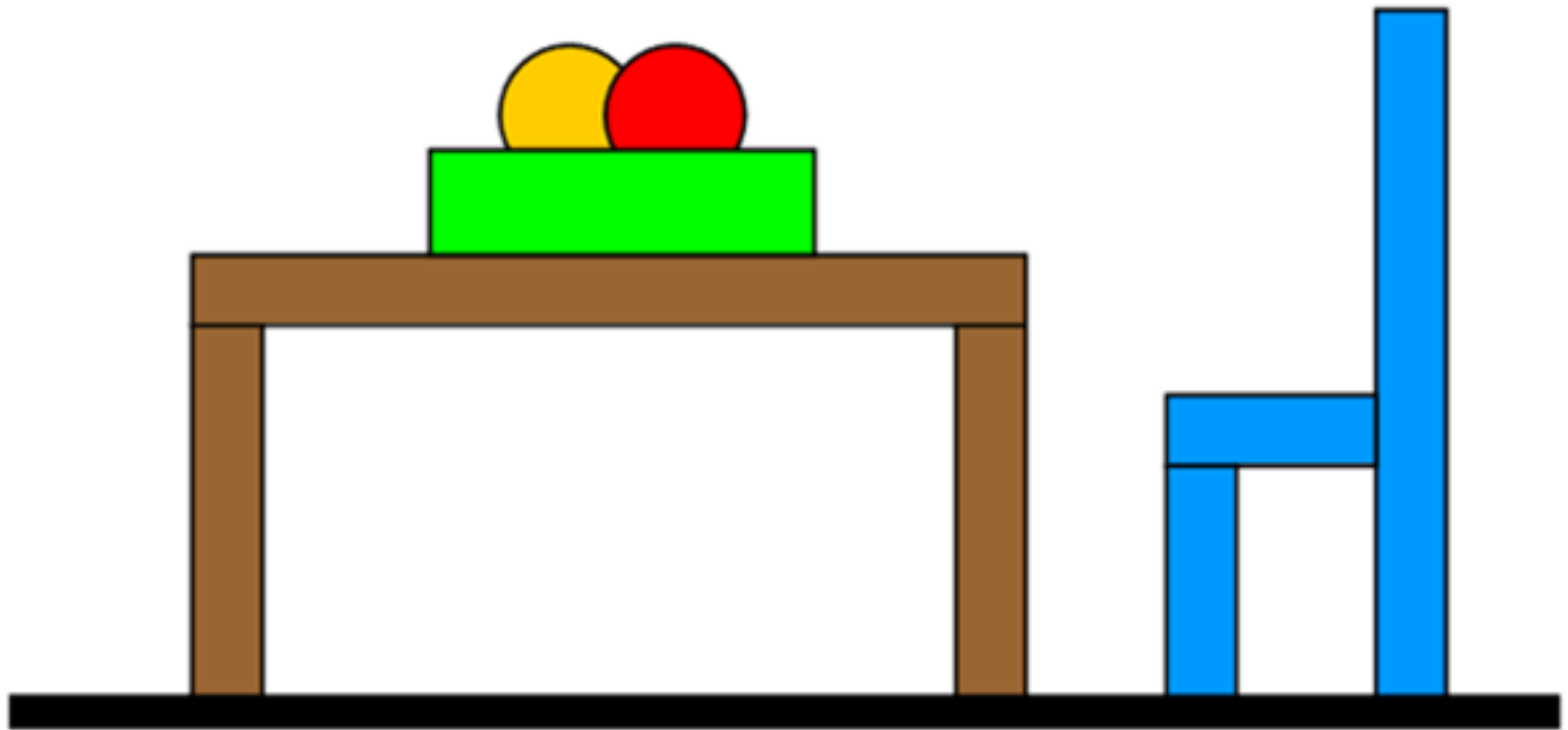
ffp@isep.ipp.pt

Modelos Hierárquicos



- ◎ Objects composed of several sub-objects
- ◎ With “joints”
- ◎ Examples:
 - ◎ robot arm
 - ◎ Solar system

Agrupamento hierárquico de objetos

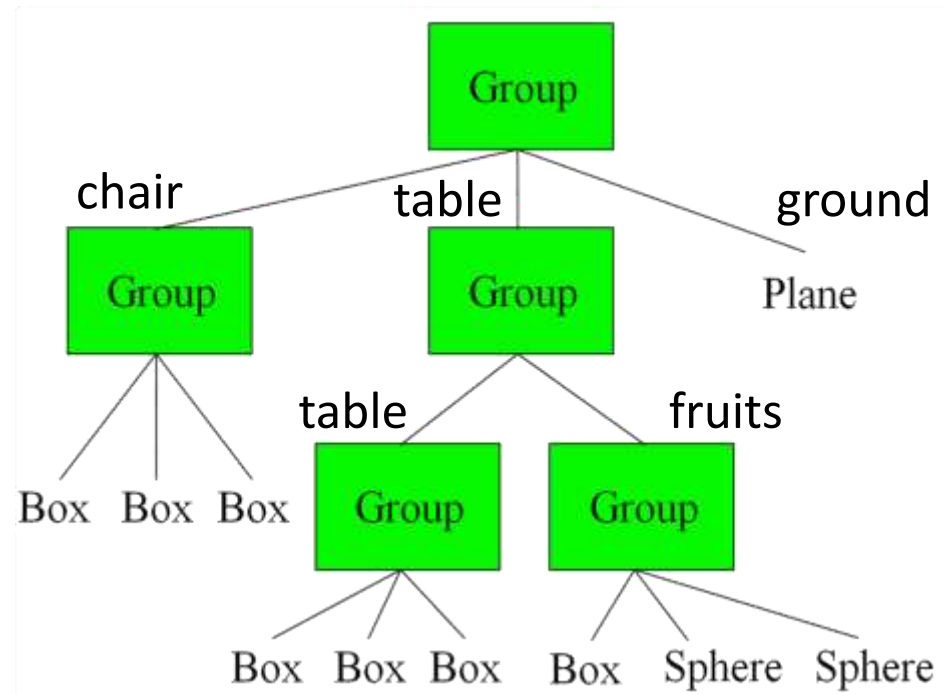
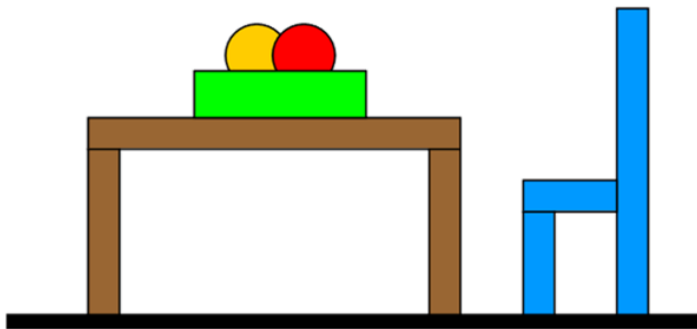


Modelos hierárquicos

- ◎ Tree structures to represent the model
- ◎ Each node is a transformation or object
- ◎ Nodes in the same branch represent accumulated transformations
- ◎ Nodes from different branches are independent transformations

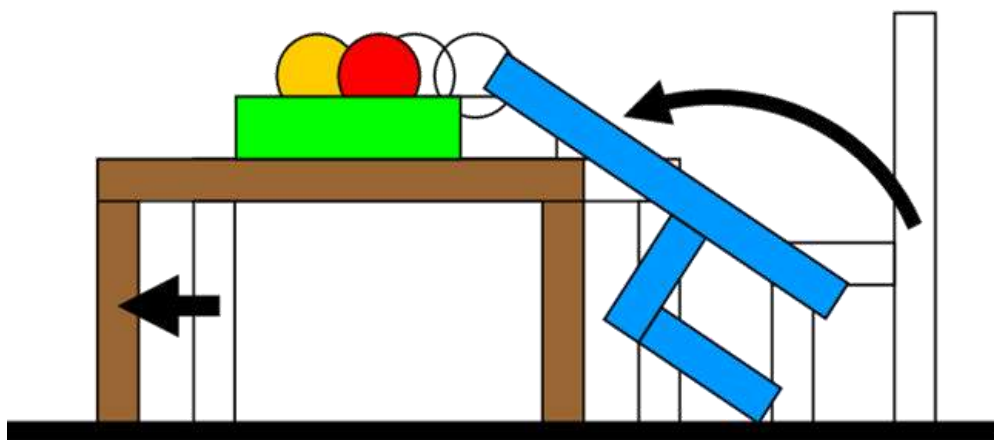
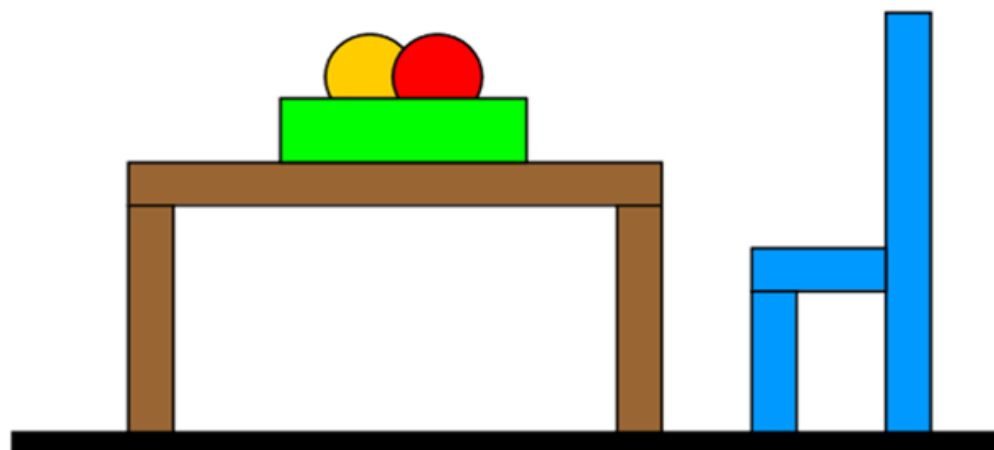
Agrupamento hierárquico de objetos

- © Logical organization of a scene



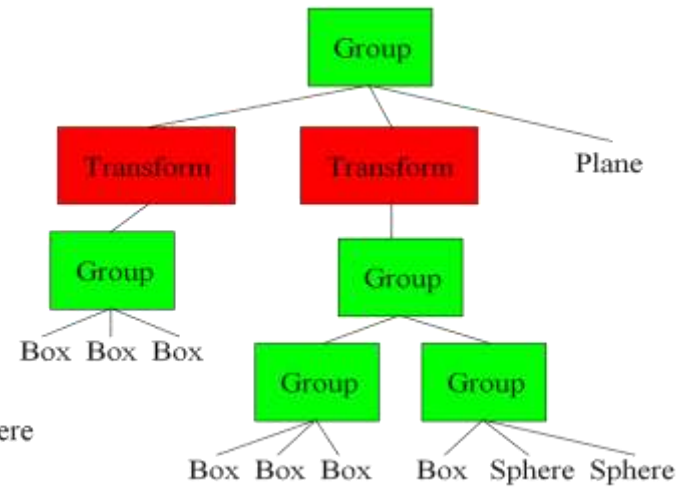
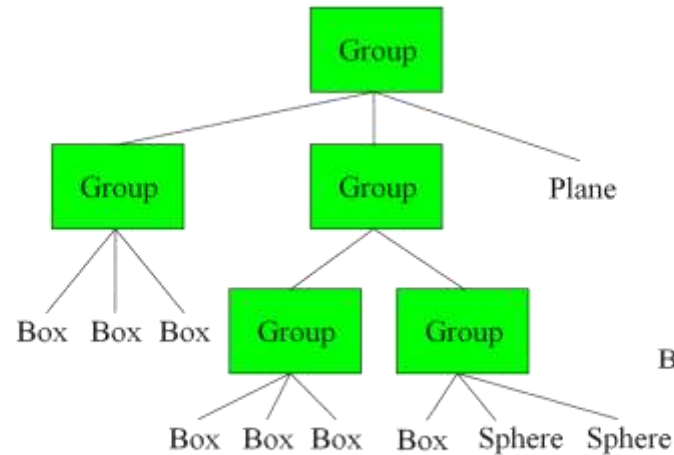
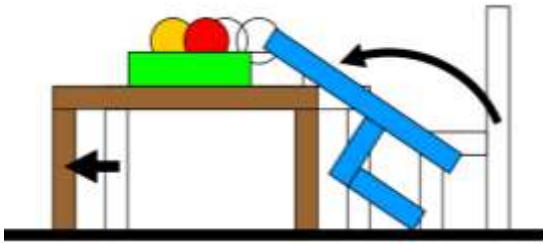
MIT Computer Graphics 6.837

Inserção de Transformações



Transformação hierárquica de objetos

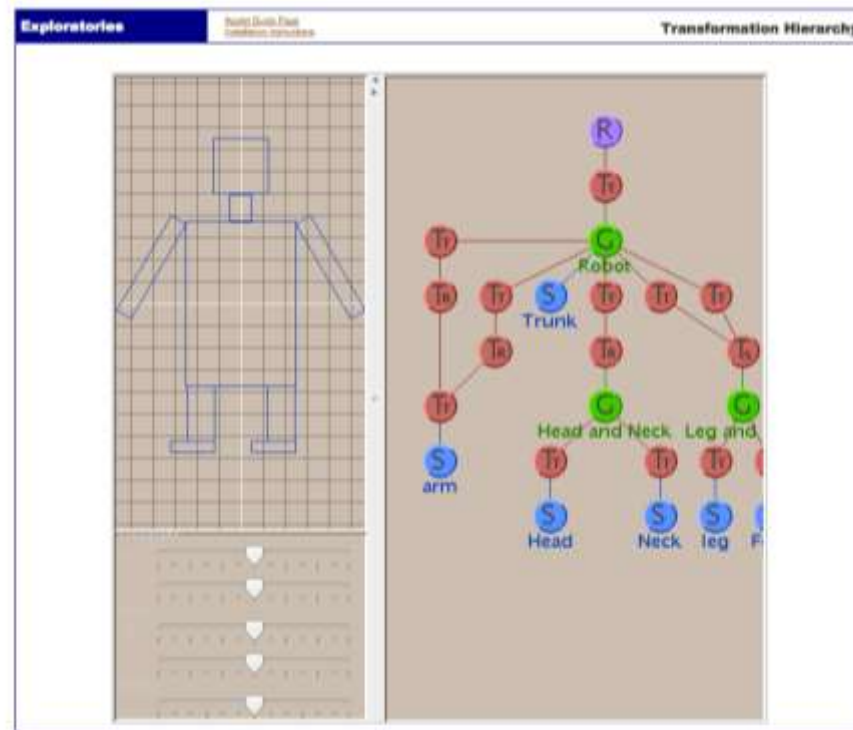
Transform the logical position of groups of objects within the scene



MIT EECS 6.837, Durand and Cutler

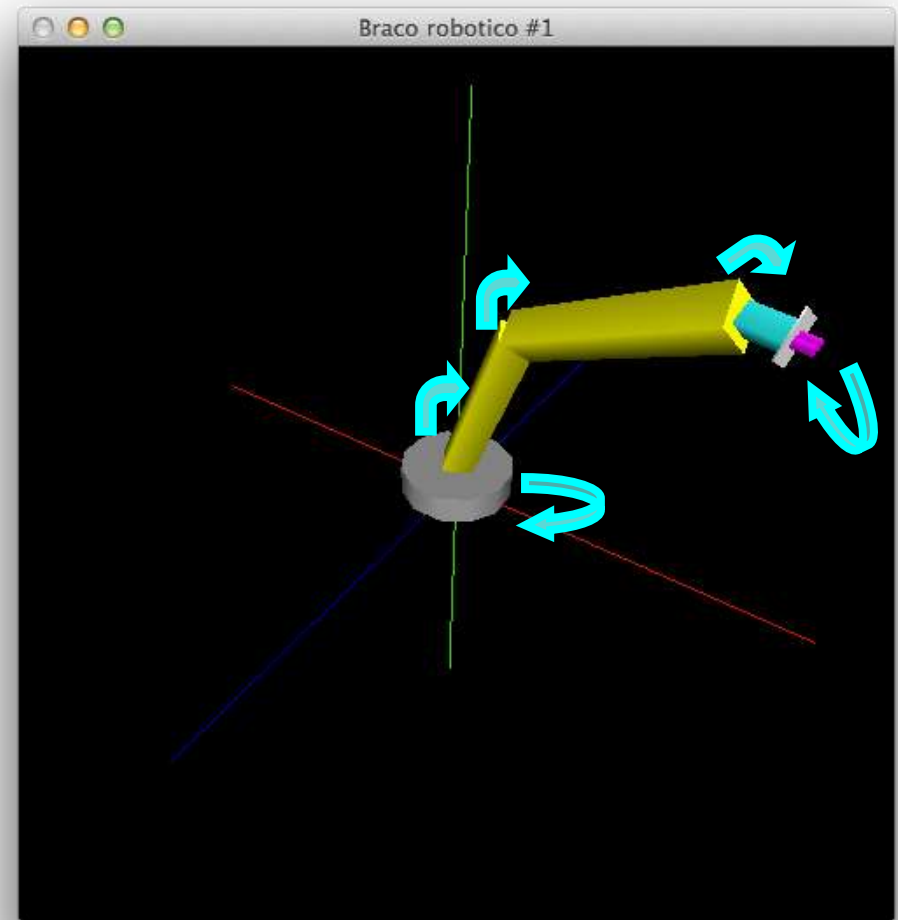
Demo

- ⊙ http://www.cs.brown.edu/exploratories/freeSoftware/repository/edu/brown/cs/exploratories/applets/transformationHierarchy/transformation_hierarchy_guide.html



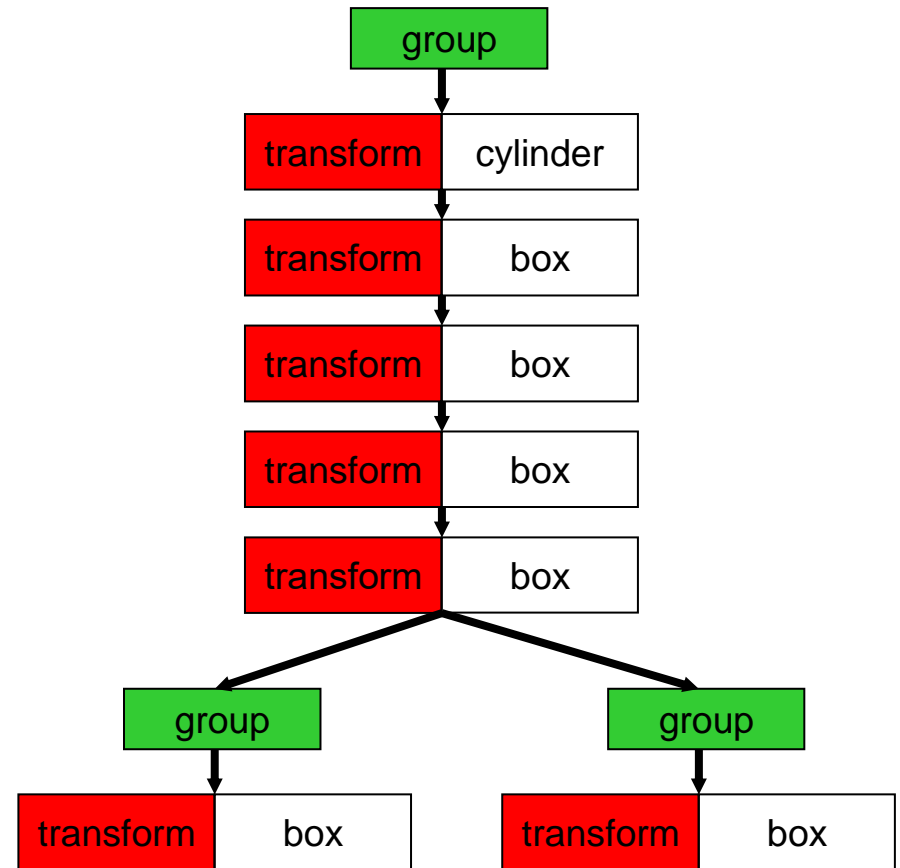
Exemplo: braço robot

- ⊙ Base
 - ⊙ Rotação no plano
- ⊙ 1º segmento
 - ⊙ Rotação no eixo
- ⊙ 2º segmento
 - ⊙ Rotação no eixo
- ⊙ Pulso
 - ⊙ Rotação no eixo
- ⊙ Garra
 - ⊙ rotação no plano
 - ⊙ Fecha/abre



Exemplo: braço robot

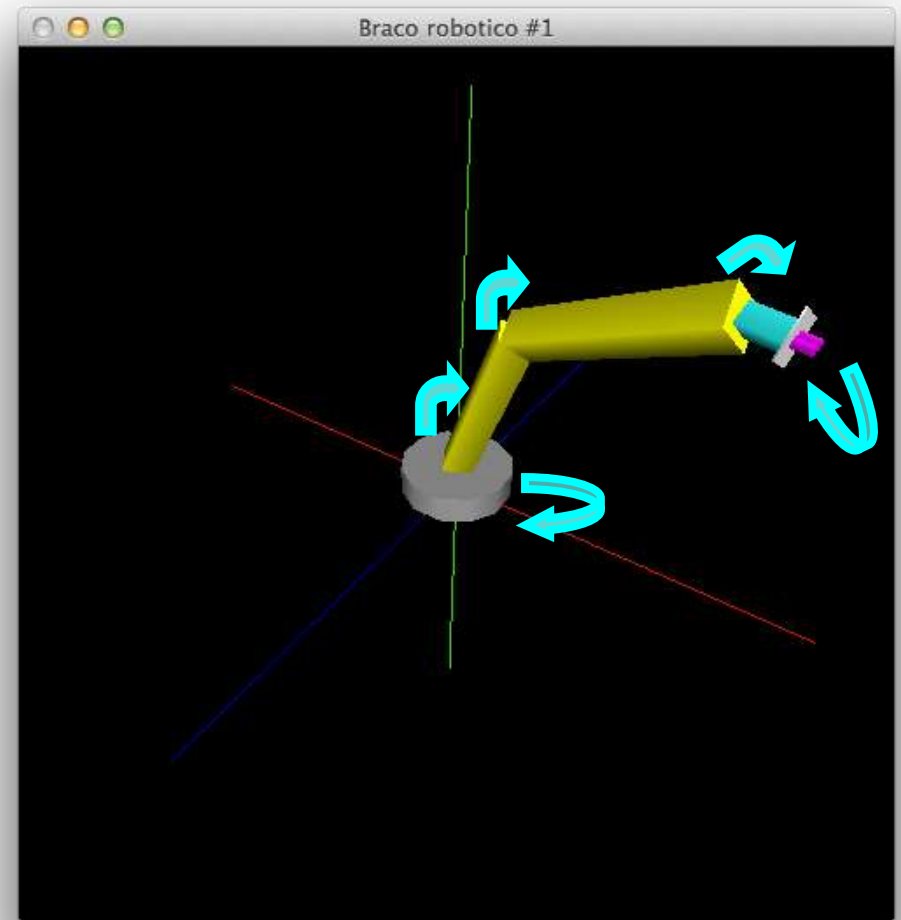
- ⊙ Base
 - ⊙ Rotação no plano
- ⊙ 1º segmento
 - ⊙ Rotação no eixo
- ⊙ 2º segmento
 - ⊙ Rotação no eixo
- ⊙ Pulso
 - ⊙ Rotação no eixo
- ⊙ Garra
 - ⊙ Rotação no plano
 - ⊙ Fecha/abre



Classe RobotArm

```
class RobotArm
{
    GLfloat rotBase;
    GLfloat rotSeg1;
    GLfloat rotSeg2;
    GLfloat rotWrist;
    GLfloat rotClaw;
    bool clawOpened;

    ...
}
```

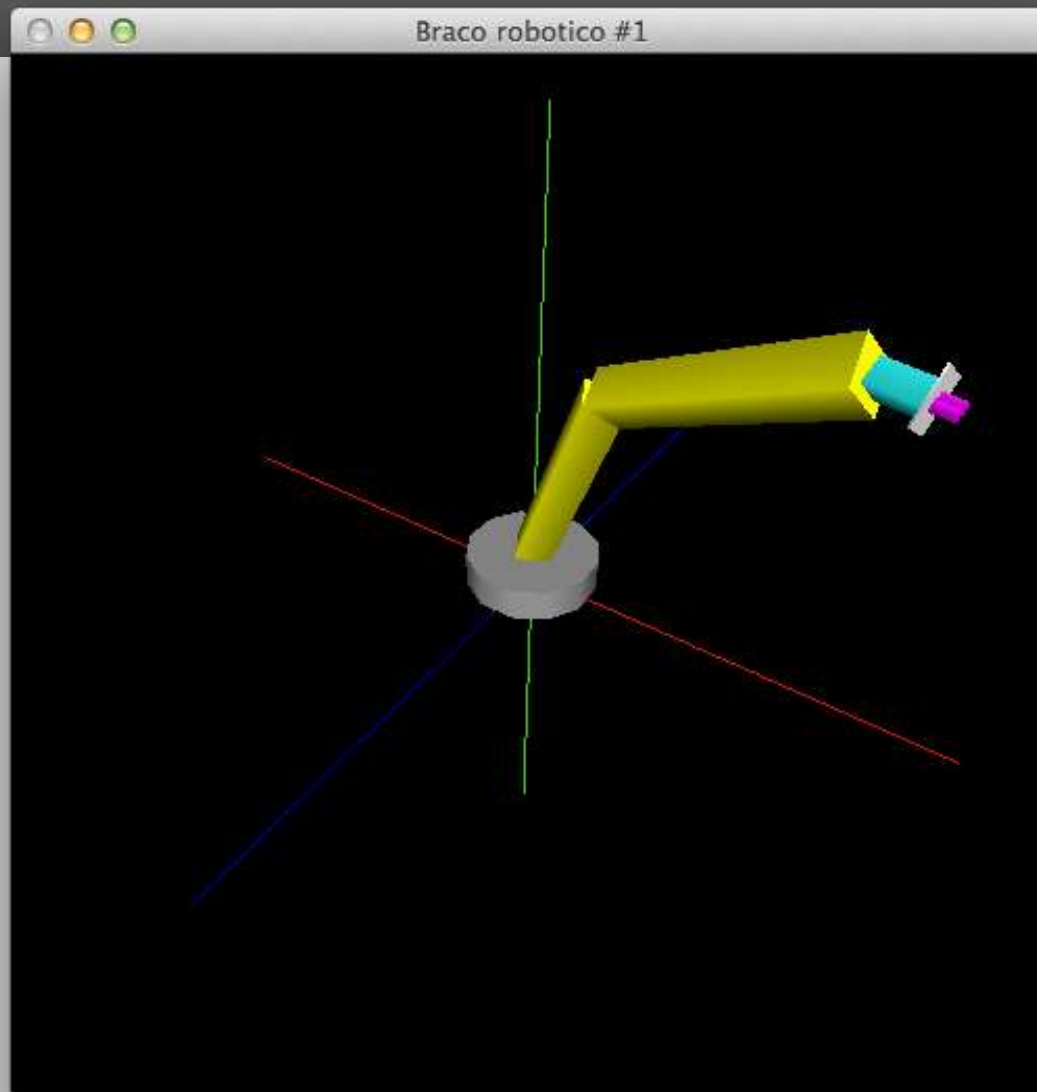


```

glPushMatrix();
    //base
R   glRotatef(rotBase, 0, 0, 1);
    cylinderWithTopAndBottom(mode, BASE_RADIUS, BASE_HEIGHT, 12, 2);
    //segmento 1
T   glTranslatef(0, 0, BASE_HEIGHT);
R   glRotatef(rotSeg1, 1, 0, 0);
    box(mode, SEG1_WIDTH, SEG1_LENGTH);
    //segmento 2
T   glTranslatef(0, 0, SEG1_LENGTH);
R   glRotatef(rotSeg2, 1, 0, 0);
    box(mode, SEG2_WIDTH, SEG2_LENGTH);
    //pulso
T   glTranslatef(0, 0, SEG2_LENGTH);
R   glRotatef(rotWrist, 1, 0, 0);
    box(mode, WRIST_WIDTH, WRIST_LENGTH);
    //garra
T   glTranslatef(0, 0, WRIST_LENGTH);
R   glRotatef(rotClaw, 0, 0, 1);
    box(mode, CLAW_BASE_WIDTH, CLAW_BASE_LENGTH);
    // pinças
T   glTranslatef(0, 0, CLAW_BASE_LENGTH);
    float d = (clawOpened ? CLAW_BASE_WIDTH/2 : CLAW_WIDTH/2);
    // pinça "direita"
    glPushMatrix();
        glTranslatef(-d, 0, 0);
        box(mode, CLAW_WIDTH, CLAW_LENGTH);
    glPopMatrix();
    // pinça "esquerda"
    glPushMatrix();
        glTranslatef(+d, 0, 0);
        box(mode, CLAW_WIDTH, CLAW_LENGTH);
    glPopMatrix();
glPopMatrix();

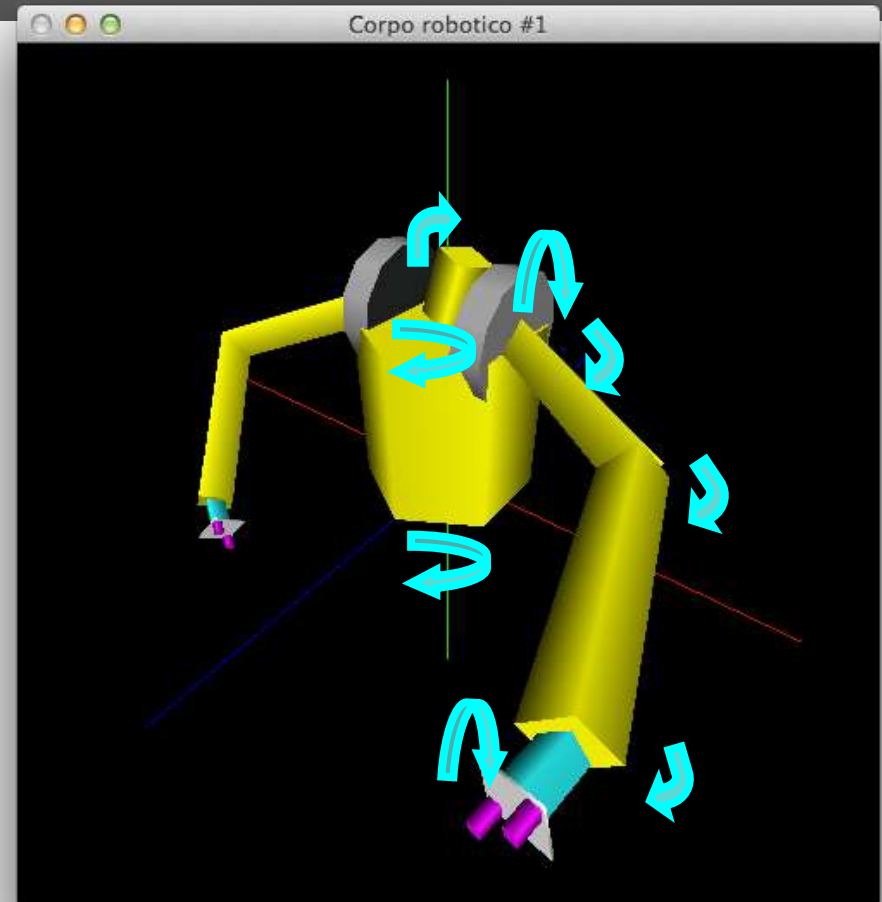
```

Demo



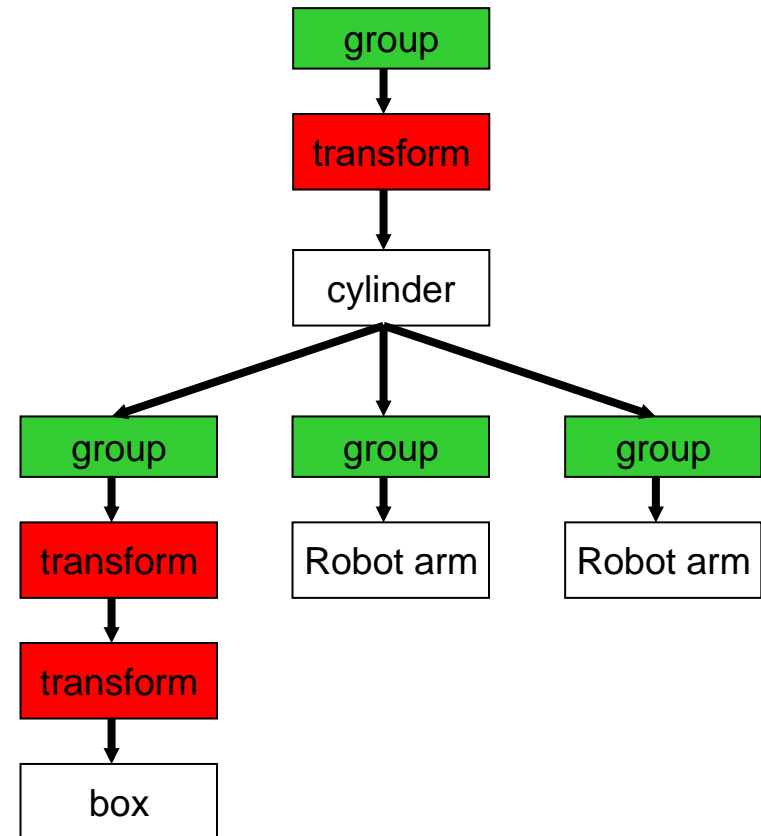
Exemplo: corpo robot

- ⊙ Tronco
 - ⊙ Rotação no plano
- ⊙ Pescoço
 - ⊙ Rotação no plano
 - ⊙ Rotação no eixo
- ⊙ Dois braços
 - ⊙ Rotação no plano
 - ⊙ Rotação no eixo



Exemplo: corpo robot

- ⊙ Tronco
 - ⊙ Rotação no plano
- ⊙ Pescoço
 - ⊙ Rotação no plano
 - ⊙ Rotação no eixo
- ⊙ Dois braços

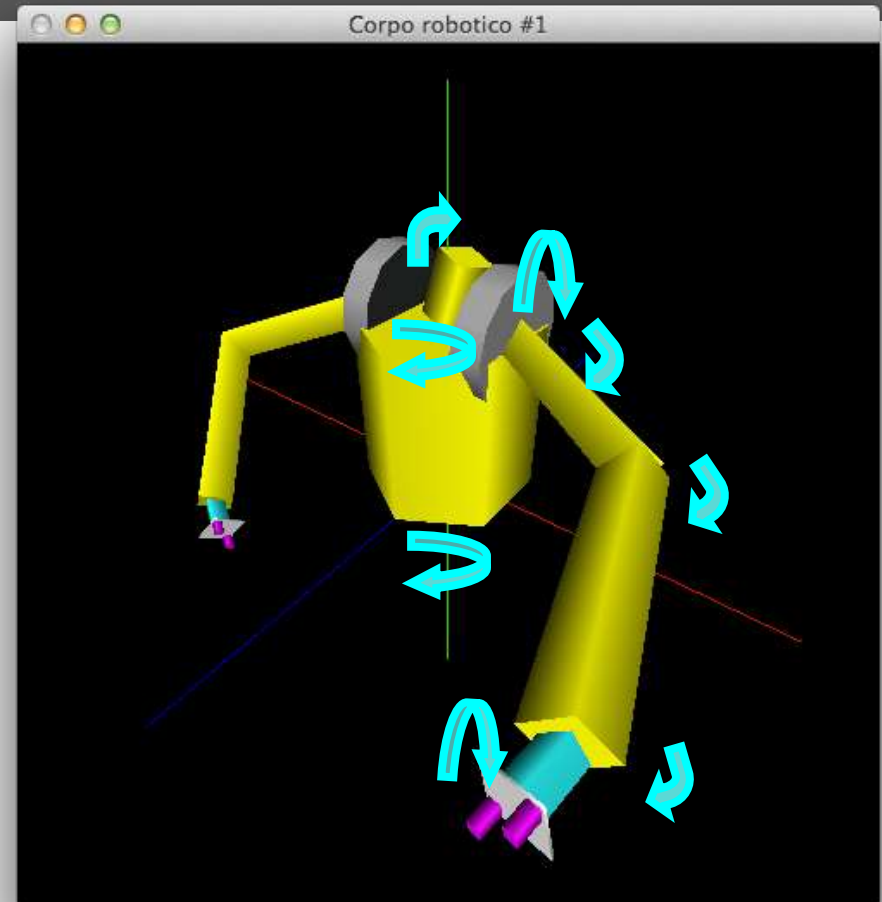


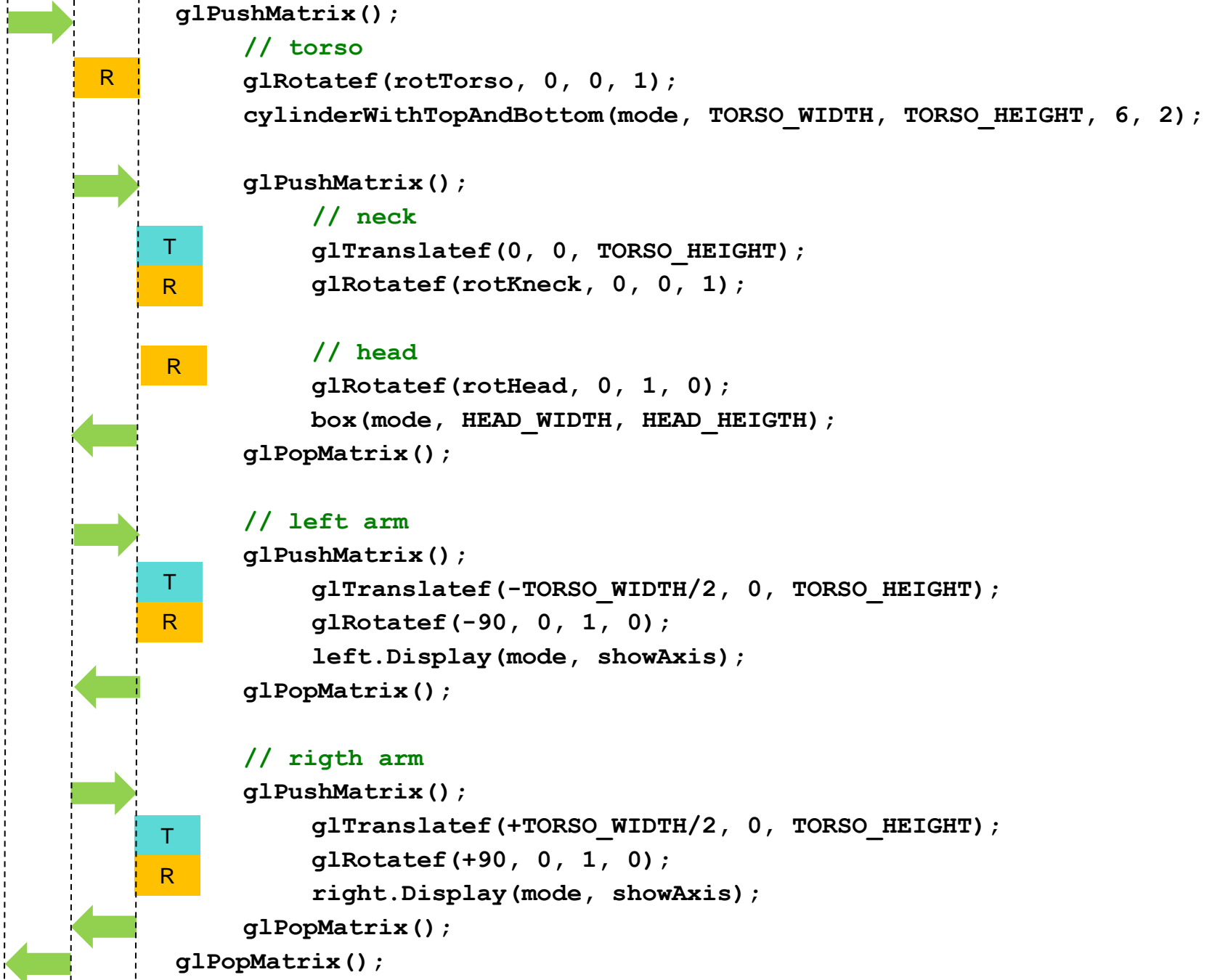
Classe Robot

```
class Robot
{
    GLfloat rotTorso;
    GLfloat rotKneck;
    GLfloat rotHead;

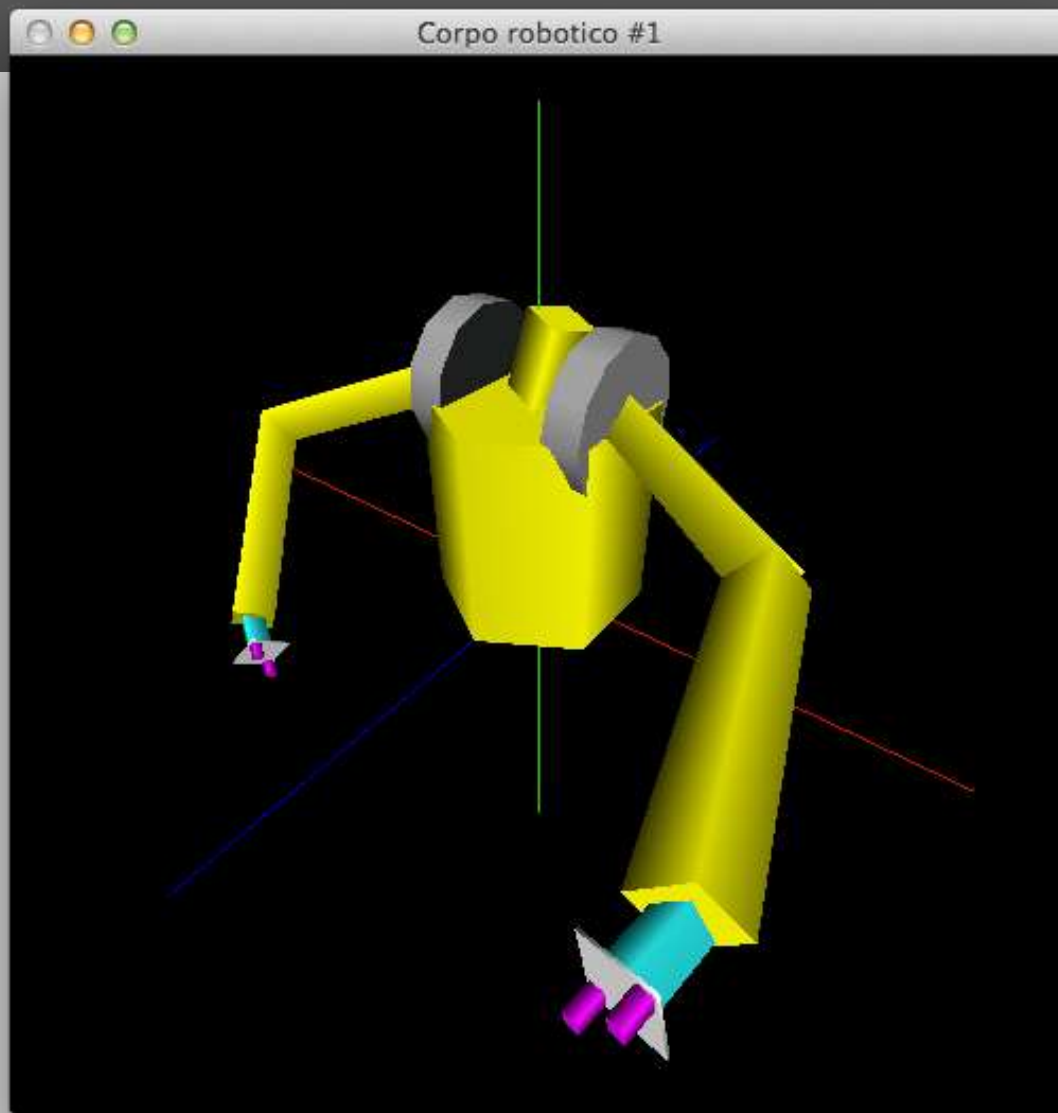
    RobotArm left;
    RobotArm right;

    ...
}
```





Demo



Implementação genérica

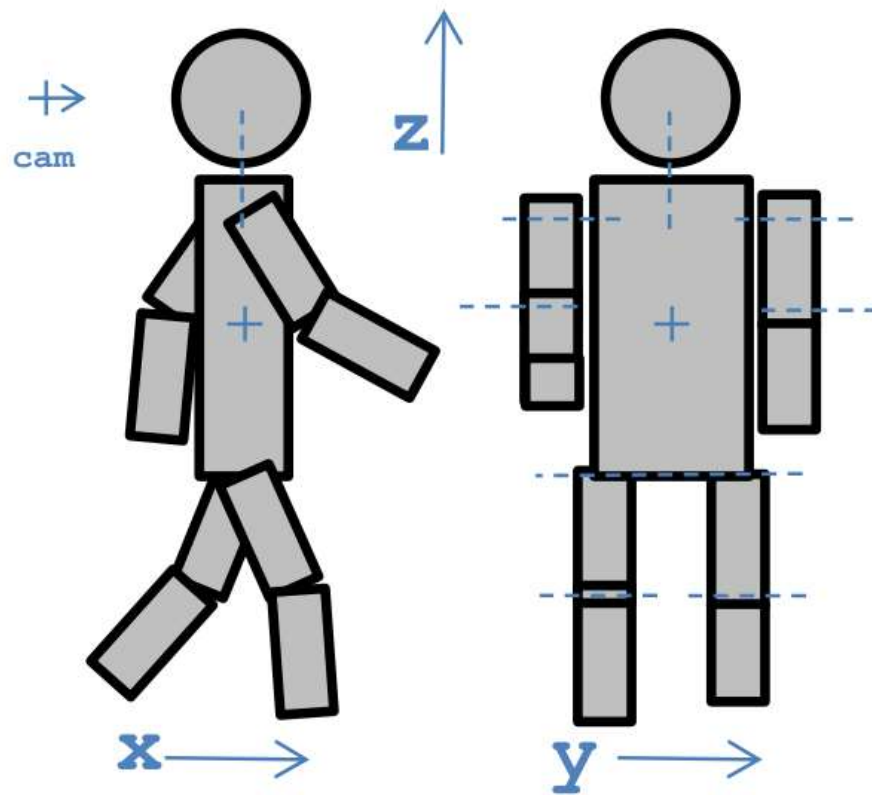
- ◎ Use a tree structure to describe the models
- ◎ Node types: Transformation or Element
- ◎ OO polymorphism
- ◎ Generic routine for going through the tree and drawing the objects
- ◎ Read model from file

Exemplo

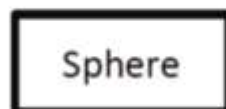
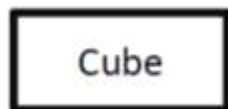
```
class No {
    List<No> descendentes;
    void virtual Draw();
}

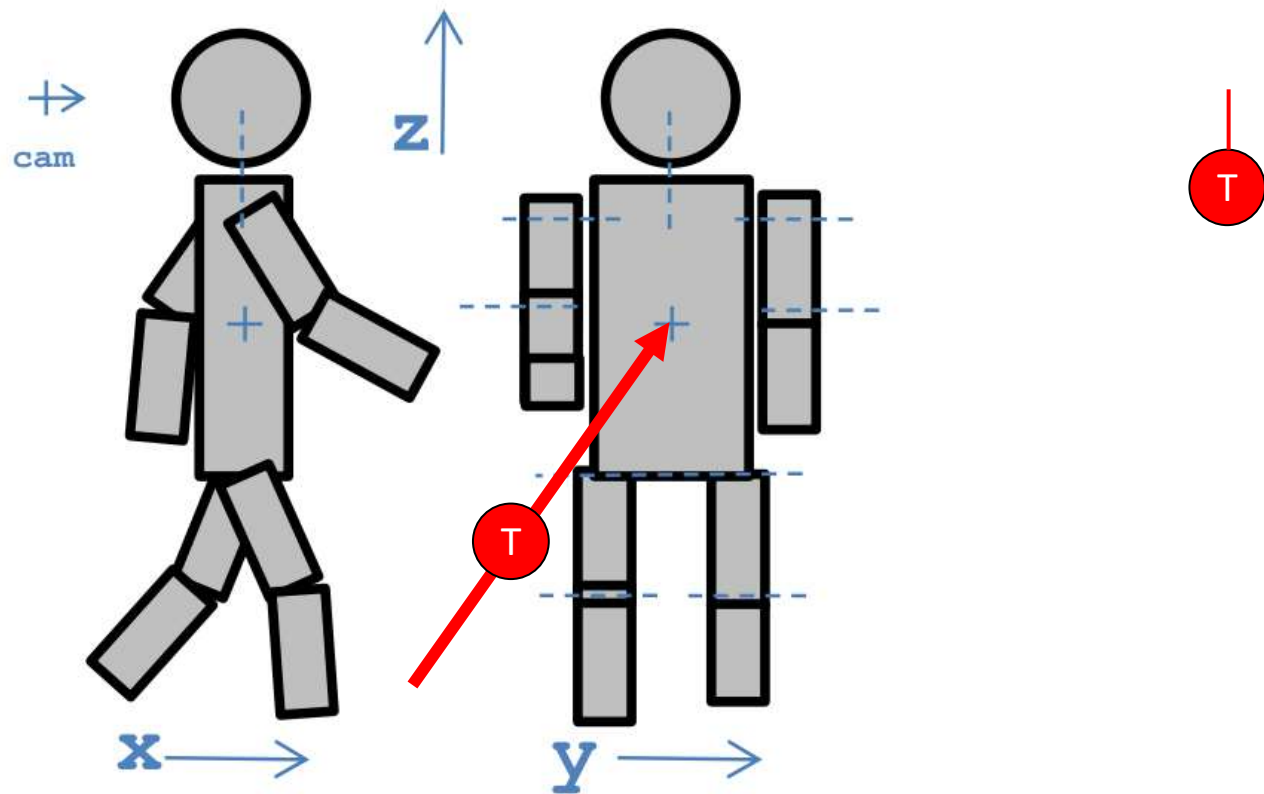
class Transformacao : No { ... }
class Translacao : Transformacao { ... }
class Rotacao : Transformacao { ... }
class Escala : Transformacao { ... }

class Elemento : No { ... }
class Esfera : Elemento { ... }
class Cubo : Elemento { ... }
class Paralelepipedo : Elemento { ... }
```



- e. [3.0] Pretende-se implementar a árvore de desenho do boneco anterior, para o corpo, cabeça e **apenas um dos braços**. Use as funções `glutSolidCube(1.0)` para desenhar cada um dos paralelepípedos e `glutSolidSphere(modelo.corpo.raio_cabeca,10,10)` para desenhar a cabeça. Ambas as instruções desenham objetos centrados na origem atual. Construa a árvore de cena recorrendo a um ou mais dos blocos a seguir indicados:





- e. [3.0] Pretende-se implementar a árvore de desenho do boneco anterior, para o corpo, cabeça e **apenas um dos braços**. Use as funções `glutSolidCube(1.0)` para desenhar cada um dos paralelepípedos e `glutSolidSphere(modelo.corpo.raio_cabeca, 10, 10)` para desenhar a cabeça. Ambas as instruções desenham objetos centrados na origem atual. Construa a árvore de cena recorrendo a um ou mais dos blocos a seguir indicados:

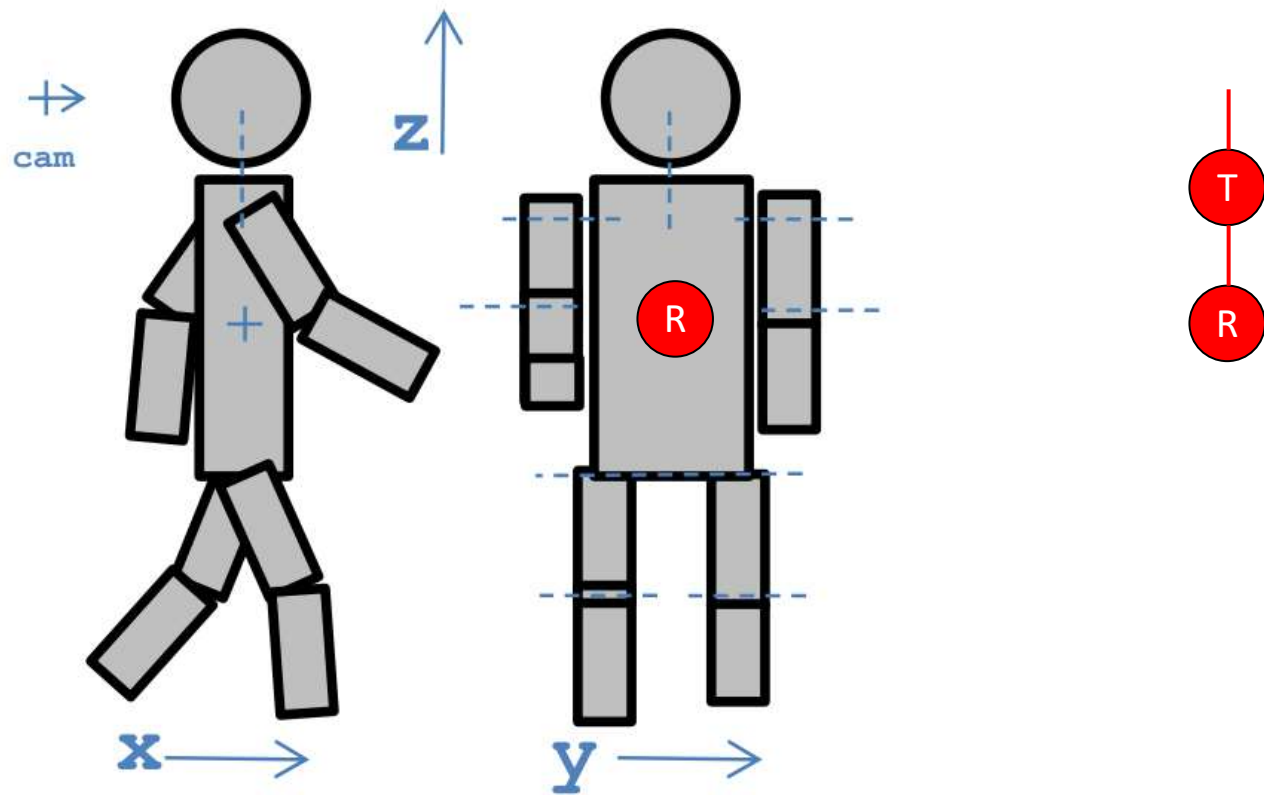
Cube

Sphere

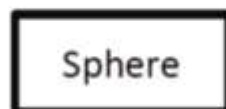
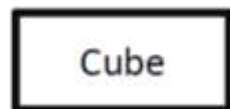
S

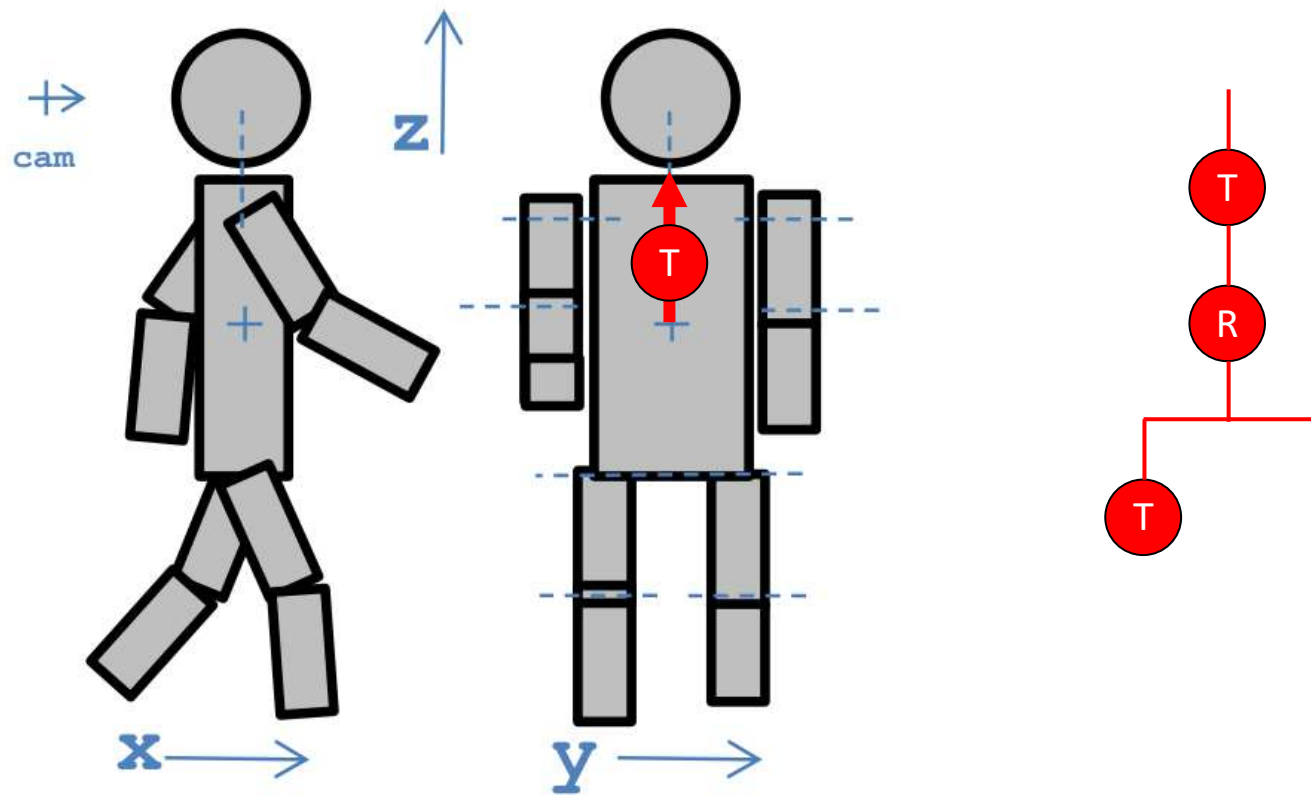
R

T



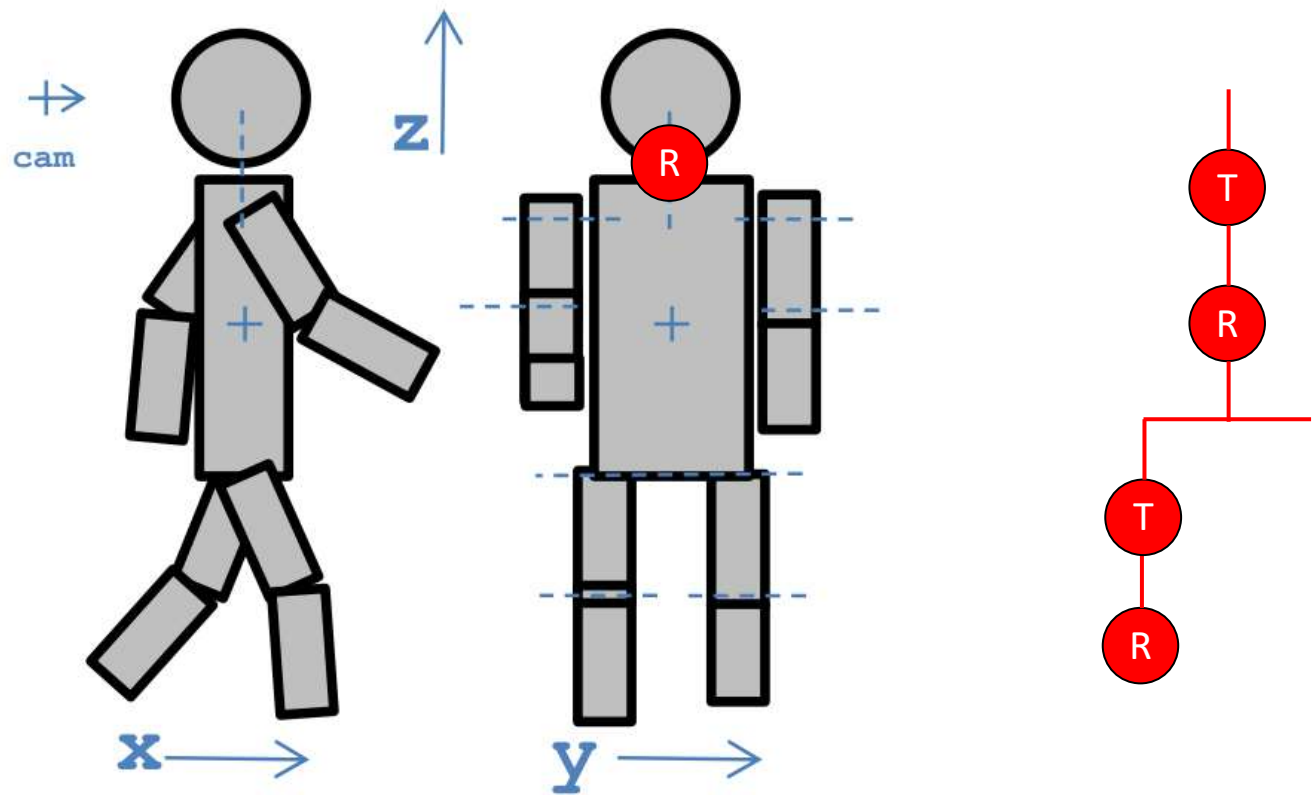
- e. [3.0] Pretende-se implementar a árvore de desenho do boneco anterior, para o corpo, cabeça e **apenas um dos braços**. Use as funções `glutSolidCube(1.0)` para desenhar cada um dos paralelepípedos e `glutSolidSphere(modelo.corpo.raio_cabeca, 10, 10)` para desenhar a cabeça. Ambas as instruções desenham objetos centrados na origem atual. Construa a árvore de cena recorrendo a um ou mais dos blocos a seguir indicados:





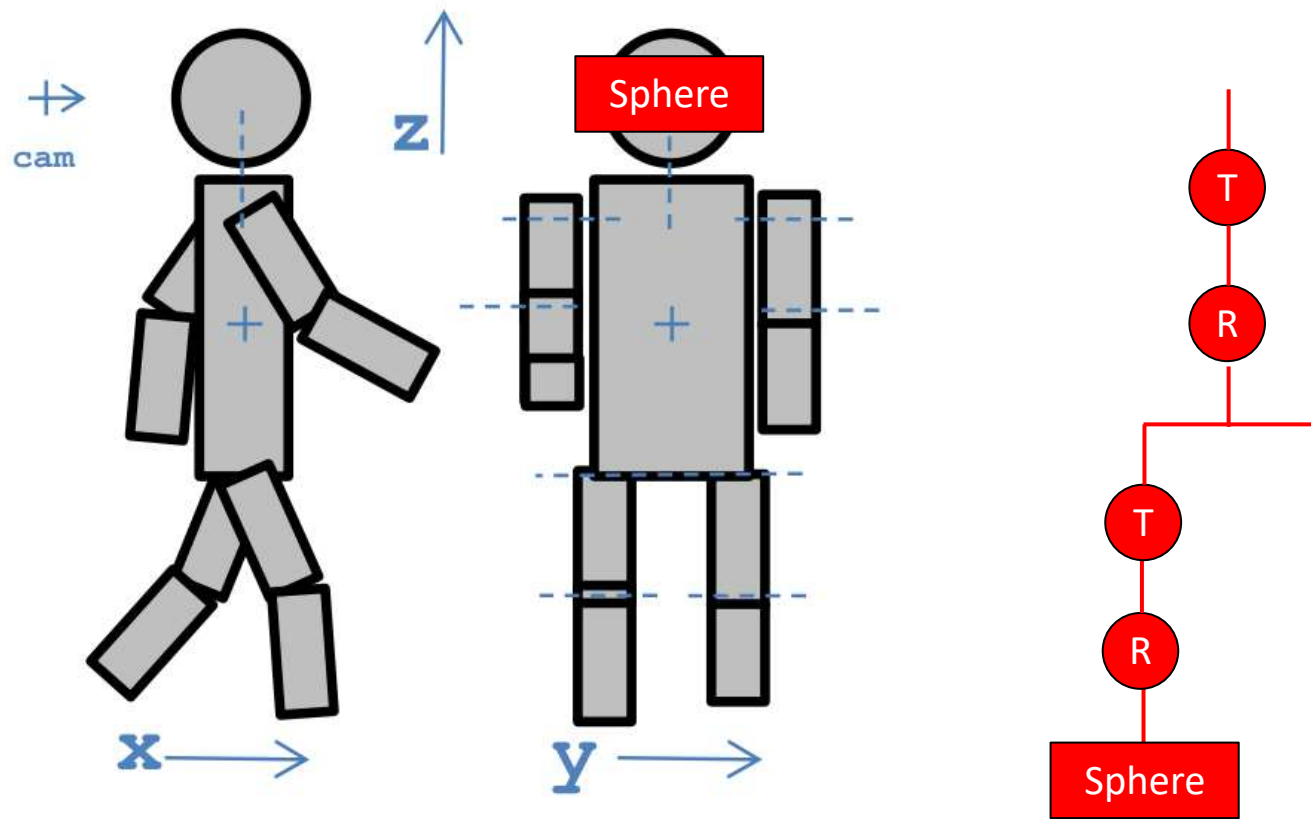
- e. [3.0] Pretende-se implementar a árvore de desenho do boneco anterior, para o corpo, cabeça e **apenas um dos braços**. Use as funções `glutSolidCube(1.0)` para desenhar cada um dos paralelepípedos e `glutSolidSphere(modelo.corpo.raio_cabeca, 10, 10)` para desenhar a cabeça. Ambas as instruções desenham objetos centrados na origem atual. Construa a árvore de cena recorrendo a um ou mais dos blocos a seguir indicados:





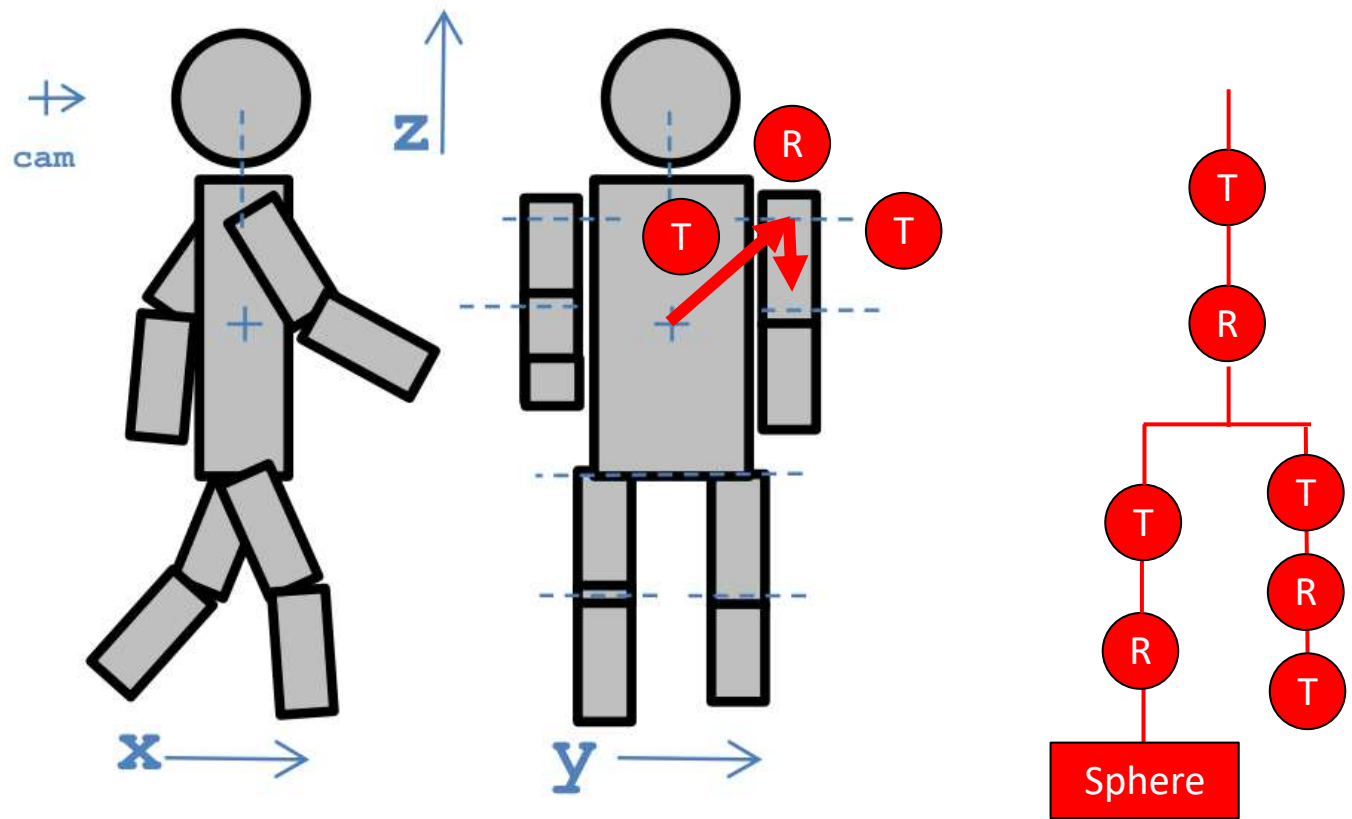
- e. [3.0] Pretende-se implementar a árvore de desenho do boneco anterior, para o corpo, cabeça e **apenas um dos braços**. Use as funções `glutSolidCube(1.0)` para desenhar cada um dos paralelepípedos e `glutSolidSphere(modelo.corpo.raio_cabeca, 10, 10)` para desenhar a cabeça. Ambas as instruções desenham objetos centrados na origem atual. Construa a árvore de cena recorrendo a um ou mais dos blocos a seguir indicados:





- e. [3.0] Pretende-se implementar a árvore de desenho do boneco anterior, para o corpo, cabeça e **apenas um dos braços**. Use as funções `glutSolidCube(1.0)` para desenhar cada um dos paralelepípedos e `glutSolidSphere(modelo.corpo.raio_cabeca,10,10)` para desenhar a cabeça. Ambas as instruções desenharam objetos centrados na origem atual. Construa a árvore de cena recorrendo a um ou mais dos blocos a seguir indicados:





- e. [3.0] Pretende-se implementar a árvore de desenho do boneco anterior, para o corpo, cabeça e **apenas um dos braços**. Use as funções `glutSolidCube(1.0)` para desenhar cada um dos paralelepípedos e `glutSolidSphere(modelo.corpo.raio_cabeca, 10, 10)` para desenhar a cabeça. Ambas as instruções desenham objetos centrados na origem atual. Construa a árvore de cena recorrendo a um ou mais dos blocos a seguir indicados:



