Transformations géométriques

- **transformation de modélisation courante (CMT)**
  - **matrice 4x4 maintenue par Renderman**
  - **appliquee à chaque objet**
  - **au contraire des autres attributs graphiques, les routines qui affectent la CMT ne la remplacent pas mais la modifient**

Transformations geometriques

Definition de transformations

- **Translation**

  RiTranslate(RtFloat dx, RtFloat dy, RtFloat dz)

- **Homothetie**

  RiScale(RtFloat sx, RtFloat sy, RtFloat sz)

- **Rotation**

  RiRotate(RtFloat angle, RtFloat dx, RtFloat dy, RtFloat dz)

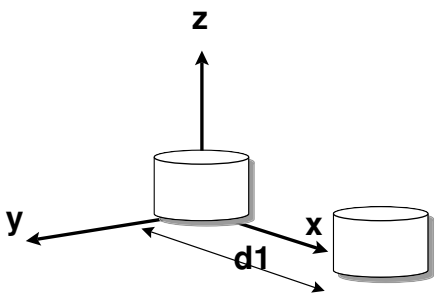
- **Transformation quelconque**

  RiTransform(RtMatrix m)

- **La transformation est concatenee a CMT**
  - **CMT = CMT x Transformation elementaire**
  - **application des tranformations dans ordre inverse de leur definition**

Transformations géométriques
Composition de transformations élémentaires

-

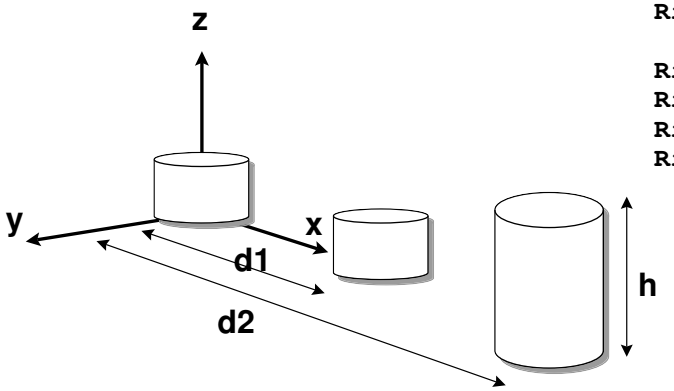



```
RiColor(Grey);
RiCylinder(1.0, 0.0,1.0, 360, RI_NULL);

RiColor(Cyan)
RiRTranslate(d1,0.0,0.0);
RiCylinder(1.0, 0.0,1.0, 360, RI_NULL);
```


Transformations géométriques
Composition de transformations élémentaires

-

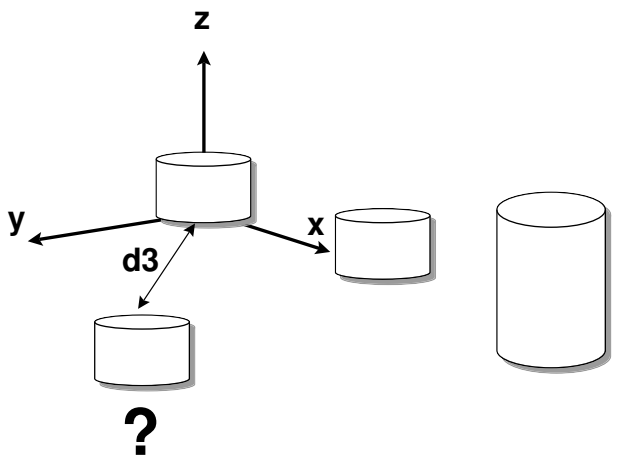



```
RiColor(Grey);
RiCylinder(1.0, 0.0,1.0, 360, RI_NULL);

RiColor(Cyan)
RiRTranslate(d1,0.0,0.0);
RiCylinder(1.0, 0.0,1.0, 360, RI_NULL);

RiColor(Yellow);
RiRTranslate(d2-d1,0.0,0.0);
RiScale(1.0,1.0,h);
RiCylinder(1.0, 0.0,1.0, 360, RI_NULL);
```

Transformations géométriques
Composition de transformations élémentaires

- 


```
RiColor(Grey);
RiCylinder(1.0, 0.0,1.0, 360, RI_NULL);

RiColor(Cyan)
RiRTranslate(d1,0.0,0.0);
RiCylinder(1.0, 0.0,1.0, 360, RI_NULL);

RiColor(Yellow);
RiRTranslate(d2-d1,0.0,0.0);
RiScale(1.0,1.0,h);
RiCylinder(1.0, 0.0,1.0, 360, RI_NULL);
```


Transformations géométriques
Effacement de la transformation courante

- **"Remise à zero"**

  RiIdentity()

- **Remplacement par une matrice quelconque**

  RiTransform(RtMatrix t)

- **Sauvegarde / Restitution**

  RiTransformBegin()

  RiTransformEnd()

  ⇨ **modélisation hierarchique**

Modélisation hiérarchique

- **structure hierarchique modèle géométrique induite par :**

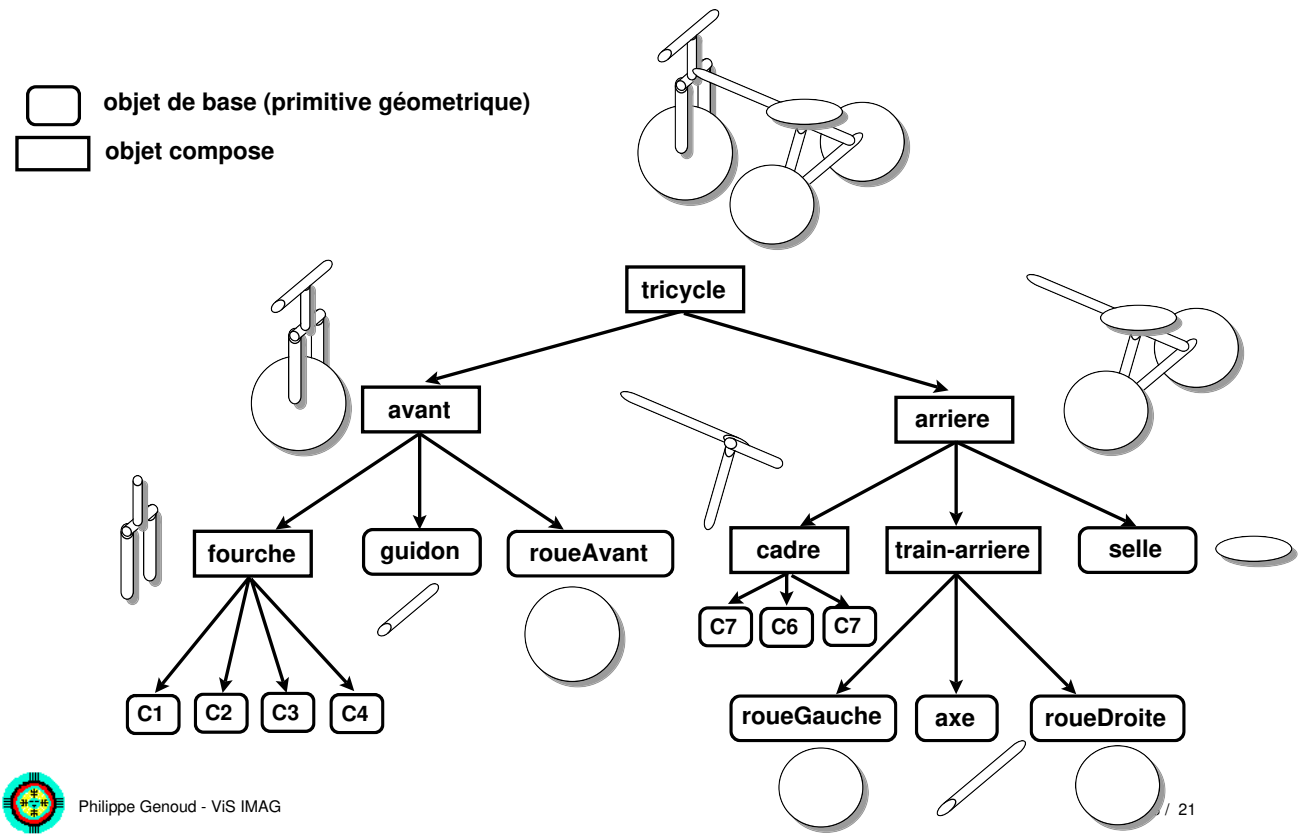
  processus de construction ascendant ("bottom-up")

  composants de base (primitives géométriques) utilisées comme blocs de construction et assemblées pour créer des entités de niveau plus élevé et ainsi de suite

  processus de construction descendant ("top-down")

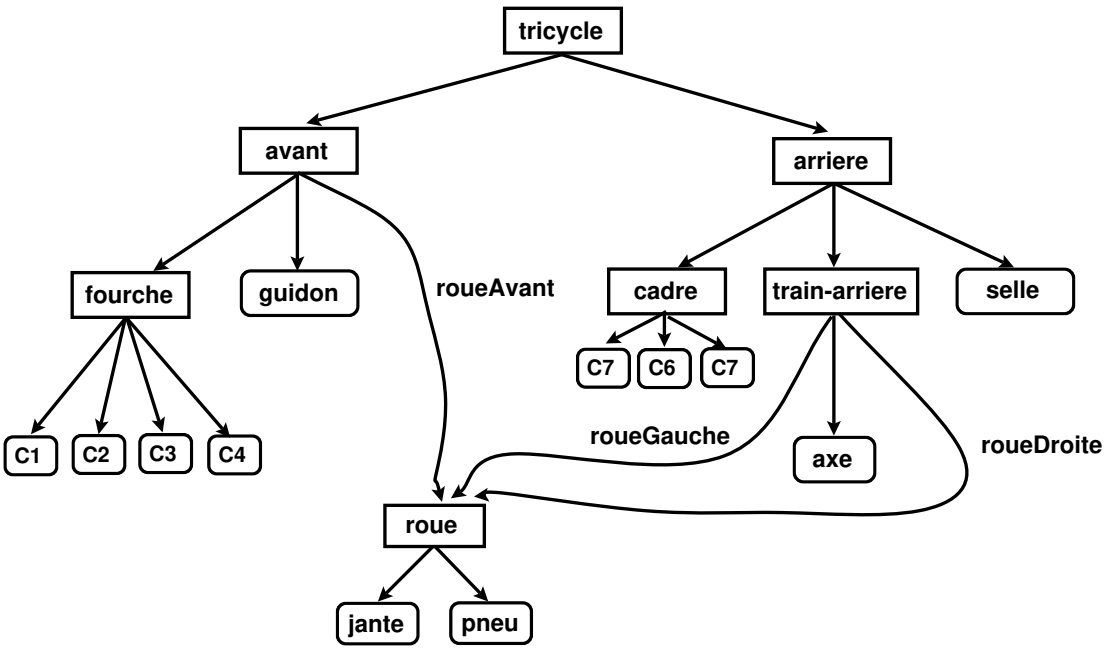
  décomposition recursive d'un modèle géométrique en objets plus simples jusqu'à aboutir à des objets élémentaires (primitives géométriques)

Modélisation hiérarchique
représentation arborescente



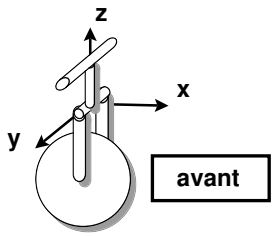
Modélisation hiérarchique
représentation arborescente

Graphe Orienté Acyclique (DAG) : un même objet peut être inclus plusieurs fois dans la hiérarchie



Modélisation hiérarchique
Transformations géométriques

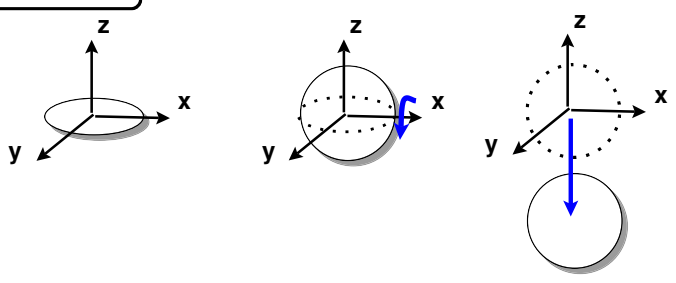
- **A chaque noeud est associé un repère**
- **Repère associé à la racine : repère de la scène**
- **A chaque arc est associée une transformation géométrique (transformation de modélisation) qui positionne l'objet fils dans le repère de son père**



tricycle

Rotation 90 axe des x puis Translation de –h en z

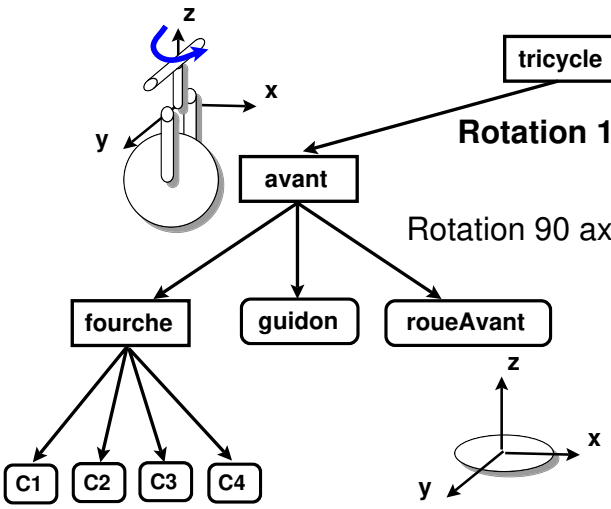
fourche guidon roueAvant



C1 C2 C3 C4

Modélisation hiérarchique
Transformations géométriques

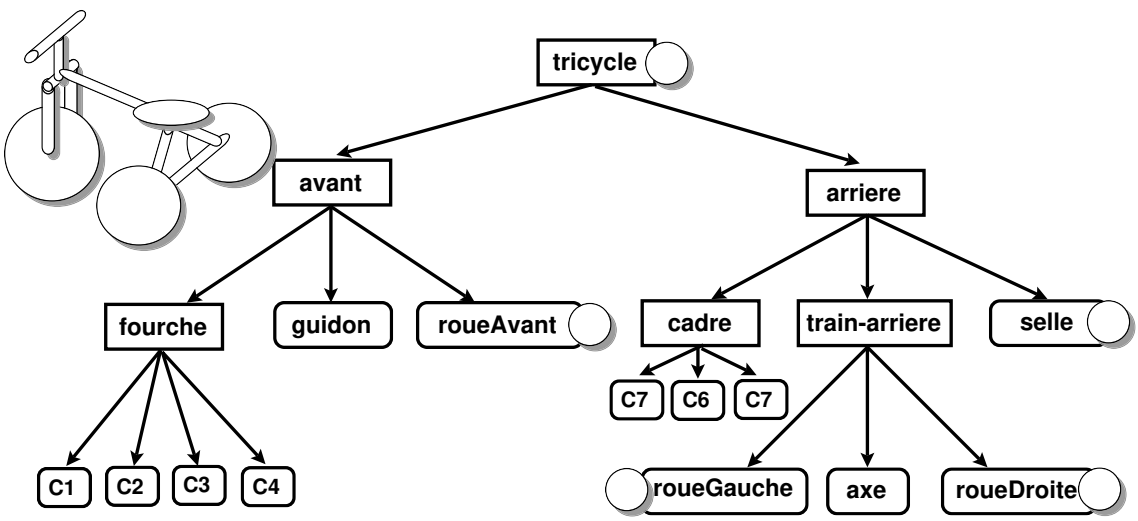
- **Lors de la visualisation les transformations de modélisation sont combinées dans la hierarchie.**
- **Primitive géometrique (feuille) affichée en transformant ses coordonnées avec la transformation qui est le produit de toutes les transformations de modélisation associées aux arcs reliant la primitive à la racine**



Rz 10° . T(0,0,-h) . Rx 90° . $\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

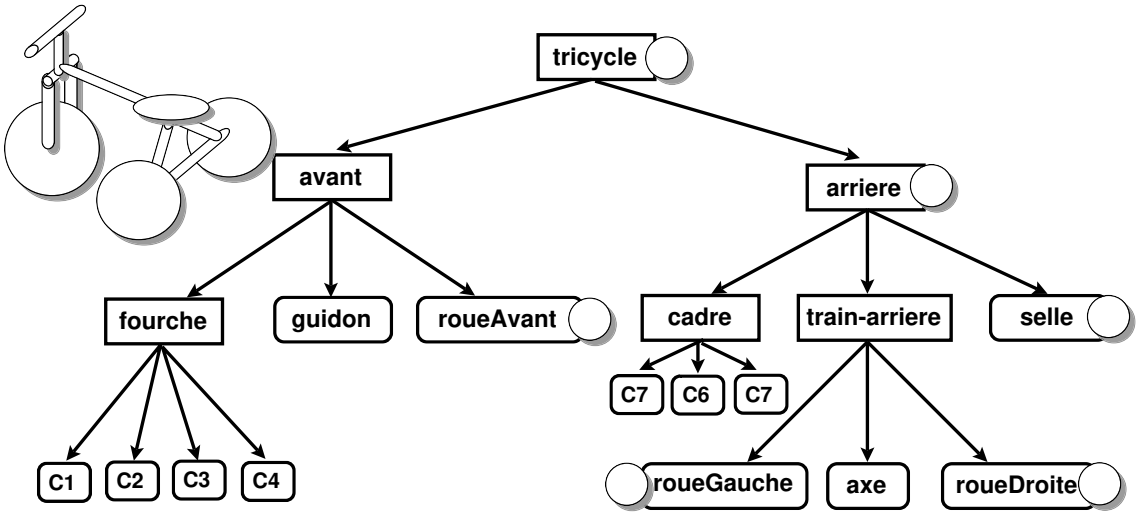
Modelisation hierarchique
composition des attributs graphiques

- **un attribut graphique associe a un noeud de la hierarchie s'applique a tous les objets de la sous arborescence de ce noeud**



Modelisation hierarchique
composition des attributs graphiques

- **un attribut graphique associe a un noeud de la hierarchie s'applique a tous les objets de la sous arborescence de ce noeud**



Modelisation hierarchique
integration dans les systemes graphiques

- **Structures de donnees**

  PHIGS (Programmer's Hierarchical Interactive Graphic System)
- **Hierarchie d'appels de procedures**

  openGL (SGI)

  Renderman
- **Hiérarchie d'objets (programmation objet)**

  openINVENTOR (SGI, C++)

  3DKIT (NeXT, Renderman, Objective C)

Modélisation hiérarchique systèmes graphiques
gérant une base de données hiérarchique

- **ex PHIGS**

  attributs et primitives géométriques definis par le programme memorisés par le système graphique dans des "structures"

  une structure peut faire réference à d'autres structures

  à la visualisation le système graphique parcourt la hiérarchie des structures et compose attributs et transformations géométriques

Modélisation hiérarchique systèmes graphiques
avec hiérarchie de procédures

- **ex openGL - Renderman**

  système mémorise la valeur des attributs graphiques (aspect, transformation géométrique) : contexte graphique

  définition d'un attribut graphique modifie le contexte courant

  affichage des primitives géométriques au fur et à mesure de leur définition en leur appliquant les attributs définis dans le contexte courant

  mécanisme de sauvegarde explicite du contexte courant (Pile)

Modélisation hiérarchique
Renderman

- **Sauvegarde du contexte graphique**

RiAttributeBegin();

....

RiAttributeEnd();

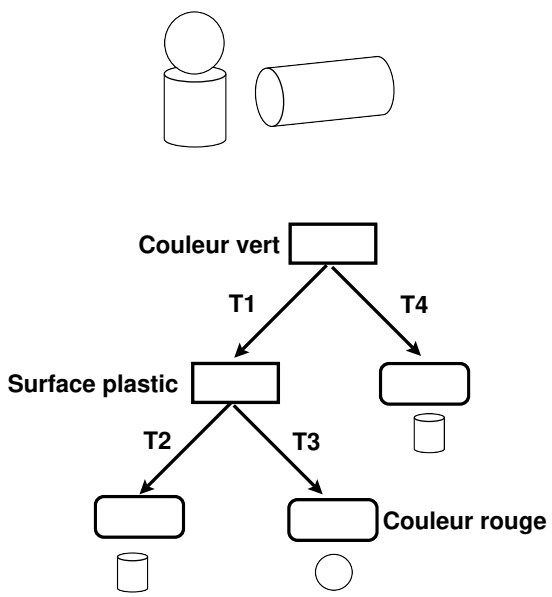
- **Sauvegarde de la CMT**

RiTransformBegin();

....

RiTransformEnd();

Modélisation hiérarchique
Renderman




```
RiColor(Green);
RiAttributeBegin();
  RiTransform(T1);
  RiSurface("plastic");
  RiAttributeBegin()
    RiTransform(T2);
    RiCylinder(...);
  RiAttributeEnd();
  RiAttributeBegin();
    RiColor(Red);
    RiTransform(T3);
    RiSphere(...);
  RiAttributeEnd();
RiAttributeEnd();
RiTransformBegin();
  RiTransform(T4);
  RiCylinder(....);
RiTransformEnd();
```

Transformation de visualisation

- **Transformation de modelisation (CMT)**

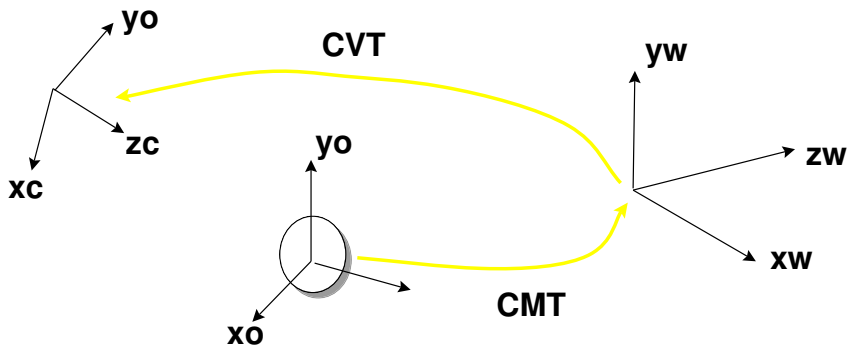
  positionne objets dans repere de la scene

  Object Coordinate Space ----> World Coordinate Space

- **Transformation de visualisation (CVT)**

  positionne objets de la scene dans le repere de la camera

  World Coordinate Space ----> Camera Coordinate Space



Transformation de visualisation (suite)

- **transfo. appliquee aux coordonnees des objets**

  CVT . CMT . (x,y,z,1)

- **specification de CVT**

```
RiTranslate(....);
...                          <-- CVT
RiRotate(...);
RiWorldBegin();
        RiTranslate(...);
        ....
        RiScale(...);        <-- CMT
        RiSphere(...);
        ....
RiWorldEnd();
```

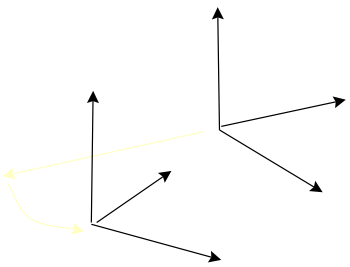
Transformation de visualisation (suite)

- **definition de CVT**

  camera un objet place dans la scene

  T tranformation placant la camera dans WCS

  T = Tn (Tn-1 (.... T1 (cam) ...))

  CVT = tranformation placant la scene dans repere de la camera

  CVT = T' (inverse de T)
  = T1' (.... (Tn-1' (Tn' (world))...)