

Graphe de Scène Java3D

Patrick Reignier

March 12, 2001

1 Introduction

L'objectif de ce TP est de se familiariser avec le graphe de scène Java3D, les principaux objets et les principales méthodes permettant de le construire.

Le but est de construire un robot composé de deux bras et avant-bras que l'on peut commander à l'aide de 4 `scrollbars`

2 Fichiers initiaux

Récupérez les fichiers sur le site <http://www-prima/reignier/lp/tp.tgz>. Décompressez les fichiers à l'aide de la commande `tar xzvf tp.tgz`. Vous obtenez alors l'arborescence de la figure 1.

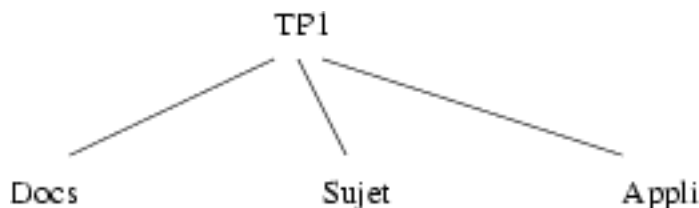


Figure 1: Arborescence du TP

Le répertoire **Sujet** contient la version PostScript et pdf de ce texte. Le répertoire **Appli** contient le squelette de l'application à réaliser. Le répertoire **Docs** contient la documentation des classes fournies (générée par `javadoc`).

3 Classes fournies

3.1 Classe Appli

La classe **Appli** est la classe principale de l'application. Elle est responsable de l'ouverture d'une fenêtre de visualisation, d'une fenêtre de contrôle (les `scrollbars`) et la mise en place des différents éléments associés à l'univers 3D :

- Un `SimpleUniverse` (attribut `universe` de la classe **Appli**).
- Un noeud `Background` (fond de l'écran).
- Deux lumières pour la prise en charge de l'éclairage de l'univers.
- Un robot.
- Un sol.

Le graphe de scène est mis en place dans le constructeur de la classe. Il est représenté figure 2.

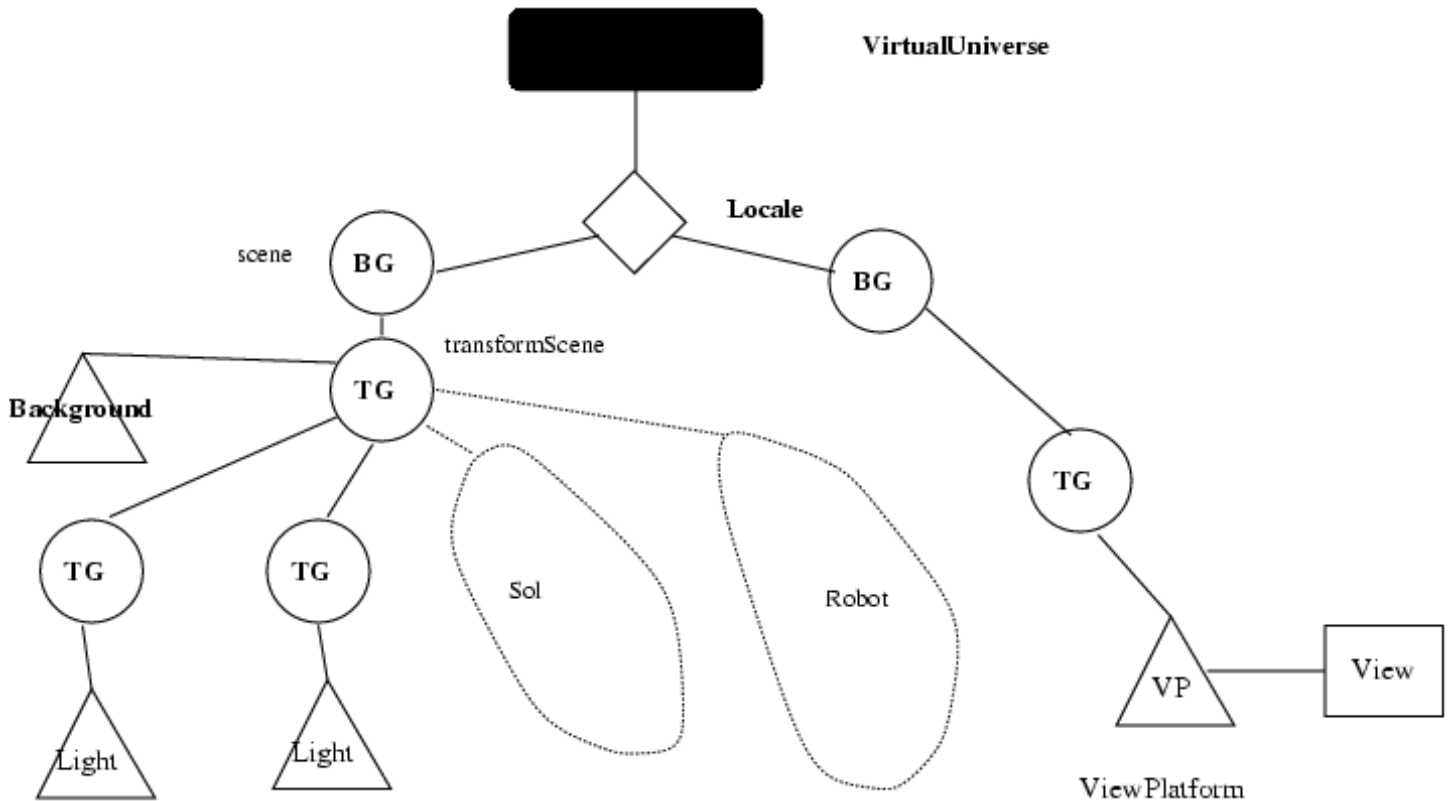


Figure 2: Graphe de scène mis en place dans le constructeur de la classe Appli

3.2 Sol

La classe Sol a pour objectif de créer un sol (plan recouvert d'une texture). Le sous-graphe de scène est construit dans le constructeur de la classe. On peut y accéder par la méthode `public Node getSol();`

3.3 Robot

La classe Robot a pour objectif de créer un robot. Le robot est constitué de *Boîtes* (classes Java3D Box). Ces boîtes sont regroupées selon le schéma. figure 3.

les différentes boîtes sont construites dans les méthodes suivantes :

```
public Node creerJambe() ;
public Node creerCorps() ;
public Node creerTete() ;
public Node creerAvantBras() ;
public Node creerBras() ;
```

Elles sont centrées autour de (0,0,0).

4 Travail demandé

1. Modifier le constructeur de la classe Appli afin de :

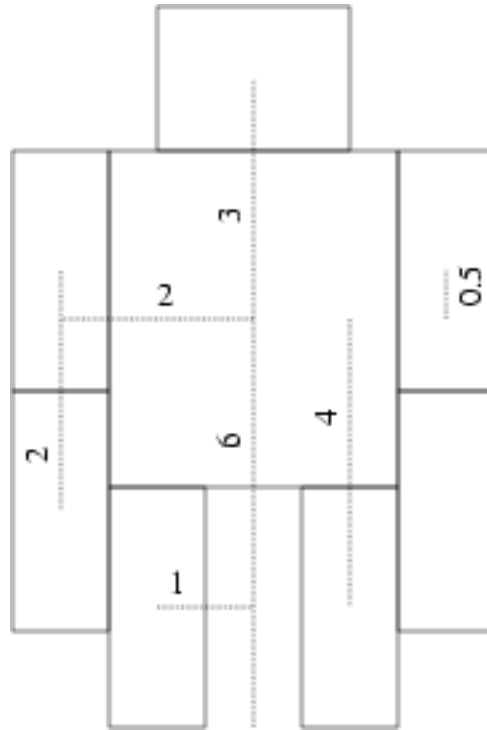


Figure 3: Plan du robot

- Créer une instance de la classe `Sol`.
- Ajouter le sous graphe associé à cette instance au graphe de l'univers virtuel. Le sol doit maintenant apparaître dans votre fenêtre de visualisation.
- Complétez le constructeur de la classe `Robot` afin de construire le sous graphe de scène associé au robot. On respectera les positions des différents éléments données figure 3.
- Modifiez le constructeur de la classe `Appli` afin de :
 - Créer une instance de la classe `Robot`.
 - Ajouter le sous graphe associé à cette instance au graphe de l'univers virtuel. Le robot doit maintenant apparaître dans votre fenêtre de visualisation (mais les bras sont immobiles).
- Complétez les méthodes suivantes :
 - `public void setBrasGauche(float theta)`
 - `public void setBrasDroit(float theta)`
 - `public void setAvantBrasGauche(float theta)`
 - `public void setAvantBrasDroit(float theta)`

Rappel : ces méthodes sont appelées chaque fois que vous bougez la `scrollbar` correspondante. L'angle passé en argument est l'angle que doit avoir le bras ou l'avant-bras correspondant. Le robot doit maintenant avoir des bras et des avant-bras contrôlables.

5 Exemples de code

5.1 Positionnement d'une box en (10,0,0)

```
TransformGroup transGroup = new TransformGroup() ; // noeud groupe
t = new Transform3D() ; // transformation 3D
t.setTranslation(new Vector3f(10.0f,0.0f,0.0f)) ;
transGroup.setTransform(t) ; // association de la transformation
                             // au noeud group
transGroup.addChild(new Box(1.0f,0.0f,0.0f,app)) ;
```

5.2 Spécification d'une rotation le long de l'axe des x

```
Transform3D trans = new Transform3D() ;
trans.setRotation(new AxisAngle4f(1.0f,0.0f,0.0f,theta)) ;
```