Visualisation de scènes 3d

Eric Maisel

1 Architecture du programme

Le programme est distribué sur plusieurs processus (exécutés éventuellement sur plusieurs machines). Ces processus coopèrent entre eux en échangeant des messages. Pour l'instant :

- Les messages sont des chaînes de caractères (une ligne) ;
- Le protocole de transport des messages utilisé est TCP/IP .

Nous appellerons par la suite **composants** ces processus. Pour l'instant il existe deux composants :

- `diffuseur` : il met en oeuvre un service de diffusion. Les autres composants peuvent s'y abonner. Tous les messages qu'un abonné envoie au diffuseur est répercuté aux autres abonnés. Les messages transitent via des sockets. Pour l'instant le port est codé en dur (50000).
- `visu3d` : ce composant permet de visualiser une scène décrite dans un format décrit ci-dessous.

D'autres composants de visualisation sont envisageables. Mettant en oeuvre d'autres fonctionnalités, écrits dans d'autres langages, ... La seule contrainte : assurer une compatibilité ascendante avec les composants existants déjà.

2 Utilisation du programme

2.1 Ligne de commandes

```
diffuseur &
visu3d nomUtilisateur &
```

2.2 Interface graphique

Il s'agit ici de l'interface du composant `visu3d`. Les commandes sont simples :

- La souris pour contrôler la direction (drag horizontal)
- Les flèches haut/bas pour contrôler la vitesse

- Les flèches gauche/droit pour contrôler le sens de déplacement (avant/arrière)
- La touche espace pour arréter le dééplacement de la caméra
- La touche ESC pour quitter le programme de visualisation

3 Description des scènes

3.1 Description des objets élémentaires

3.1.1 Sol

```
<Sol texture="../data/textures/sable.png" taille="100.0"/>
```

3.1.2 Ciel

```
<SkyBox texture="../data/skyboxes/miramar_large.jpg" size="1000"/>
```

3.1.3 Tableaux

```
<Tableau texture="../data/tableaux/Vinci-Joconde.jpg"
         hauteur="0.5" largeur="0.608" />
```

3.1.4 Objets au format obj

```
<Obj url=nomFichierObj />
```

3.2 Description des regroupements d'objets

3.2.1 Groupes d'objets avec application de transformations géométriques

Les transformations géométriques élémentaires utilisées sont :

- une translation, spécifiée par les 3 coordonnées du vecteurs translation (**par défaut** : le vecteur nul)
- une rotation, spécifiée par 3 angles de rotation (autours des axes du repère associé au noeud `Transform` (**par défaut** : la rotation d'angles 0.0, 0.0, 0.0)
- une mise à l'échelle, spécifiée par 3 facteurs d'échelle (par rapport aux axes du repère associé au noeud `Transform` (**par défaut** : la mise à l'échelle de facteurs 1.0, 1.0, 1.0)

On applique d'abord la mise à l'échelle puis les rotations puis la translation.

```
<Transform translation="0.0 0.0 0.0"
           rotation="0.0 0.0 0.0"
           scale="1.0 1.0 1.0" >
   ...
</Transform>
```

3.2.2 Listes d'affichage

Les listes d'affichage (display-list pour openGL) permettent de compiler un ensemble d'instructions d'affichage. On peut ainsi obtenir un gain de temps lors de l'affichage dun ensemble d'objets. Par contre il devient impossible de modifier le code openGL exécuté, donc d'animer ces objets. On reprend ici cette fonctionnalité en spécifiant qu'un ensemble de graphes de scène peut être compilé.

```
<DisplayList>
  ...
</DisplayList>
```

4 Protocole de communication entre les composants