



Rémy Malgouyres,  
Algorithmes pour la synthèse d'images  
et l'animation 3D,  
DUNOD, 2002

## TP n° 13 : Navigation

### Objectifs :

Le but de ce TP est de permettre la navigation dans la scène 3D créée et affichée au cours des TP précédents. On implémentera les translations par un vecteur défini dans le repère de la caméra, et les rotations autour d'axes définis dans le repère de la caméra et passant par le centre de celle-ci.

On écrira le code de la fonction suivante :

```
void Scene3D : :translatecam(double dx, double dy, double dz)
```

qui réalise sur la scène une translation de vecteur  $(dx, dy, dz)$ . Ce vecteur est défini dans le repère de la caméra (la matrice de ce repère, ainsi que la position et le centre de la caméra dans le repère de la scène sont accessibles dans l'objet de la classe la classe `CameraPerspect` pointé par la donnée `tabcam[nocamselect]` dans la classe `Scene3D`).

On utilisera pour cela les techniques vues en cours, sachant que les coordonnées des sommets des polyèdres de la scène, ainsi que les coordonnées des sources lumineuses, sont toujours maintenues dans le repère de la caméra. Les coordonnées des sommets doivent être modifiées en parcourant la liste chaînée `objets` des objets de la scène (voir la classe `Scene3D`).

On écrira ensuite le code de la fonction

```
void Scene3D : :translatecamcentral(double dx, double dy, double dz)
```

qui calcule un vecteur `dci = (dx,dy,dz/(maxix-minix)*distance)`, où `distance` représente la distance entre la position et le centre de la caméra, et `maxix-minix` représente la largeur de la scène (voir les données membres de la classe `Scene3D`, puis appelle la fonction `translatecam` pour effectuer la translation de vecteur `dci`.

La fonction `translatecamcentral` doit laisser le centre de la caméra inchangé (on pourra par exemple mémoriser l'ancien centre puis réaffecter la valeur mémorisée au centre de la caméra).

On écrira enfin le code de la fonction

```
void Scene3D : :rotatecamcentral(double teta, char noaxe)
```

Le paramètre `teta` représente l'angle de la rotation (en degrés, à convertir en radians), et le paramètre `noaxe` représente le numéro (0, 1 ou 2) de l'axe. On commencera par traiter le cas où `noaxe` vaut 2, pour lequel on fait une rotation autour de l'axe des  $z$  du repère de la caméra (voir cours).

On traitera ensuite le cas où `noaxe` vaut 0 (rotation autour d'un axe passant par le centre et parallèle à l'axe des  $x$  du repère de la caméra, voir cours).

On implémentera enfin le cas où `noaxe` vaut 1, qui correspond à un axe parallèle à l'axe des  $z$  du repère de la scène (voir cours).