

## TP 4 d'OpenGL

### Animation de personnage

Dans ce TP, nous allons enfin faire bouger notre personnage.

## 1. Boucle d'animation et pas de temps

Recuperez le code source fourni (<http://artis.imag.fr/~Pascal.Barla/opengl/tp4.zip>) et decompressez-le. Commencez par copier la nouvelle fonction d'animation et les variables associees donnees dans *a\_recopier.cpp* dans votre programme. Puis n'oubliez pas de faire un appel a la fonction *updateAnimationParameters()* dans le corps de *renderScene()*. Cela permettra de faire evoluer les parametres d'animation de notre personnage.

Prenons d'abord le temps de comprendre ce que fait la fonction *updateAnimationParameters()*. Elle calcule tout d'abord un parametre  $t$  qui va simuler le temps qui s'ecoule dans notre programme. Pour cela, il faut tout d'abord mesurer le temps passe depuis le dernier appel a la fonction (taper man clock pour plus d'info a ce sujet). Ce temps est normalise par un pas de temps appele  $\delta t$  qui permet de controler la vitesse de l'animation. La seconde partie de *updateAnimationParameters()* s'occupe ensuite de calculer un parametre d'animation (un angle de rotation) qui est determine a partir de  $t$ . En synchronisant de la sorte tous les parametres d'animation sur  $t$ , on peut synchroniser tous les mouvements correspondants.

## 2. Personnalisation du mouvement

Si l'on compile le programme, aucune animation n'est visible. Pourquoi ?

Modifier le programme afin que le personnage reponde aux parametres d'animation.

Conseil: creer une fonction *draw()* pour chaque membre du personnage que vous desirez animer, parametree par l'angle de rotation *animRotation*.

Vous avez maintenant compris le principe d'animation de personnage: decider d'un pas de temps, synchroniser les valeurs d'animation et parametrer les fonctions de dessin. Vous pouvez maintenant ajouter d'autres types de parametres (rotation, mais aussi translation ou echelle), et ainsi ajouter de la complexite au mouvement de notre personnage.