

OpenGL - TP 4

Animation de personnages

Dans ce TP, nous allons animer notre personnage.

Récupérez le code fourni. Copiez le contenu de `a_recopier.c` dans votre programme. Appelez la fonction `updateAnimationParameters()` dans `renderScene()`.

1 Boucle d'animation et pas de temps

1.1 Principe

Comme `updateAnimationParameters()` est appelée dans `renderScene()`, elle est appelée à chaque pas de temps. A chaque appel de la fonction `updateAnimationParameters()`, les paramètres de contrôle (ou paramètres d'animation) sont mis à jour. Prenons d'abord le temps de comprendre ce que fait cette fonction.

Mesure du temps : Tout d'abord, un paramètre `t` simule le temps passé depuis le dernier appel à la fonction. Ce temps est normalisé par un pas de temps appelé `deltaT` qui permet de contrôler la vitesse de l'animation.

Paramètres d'animation : La seconde partie de `updateAnimationParameters()` s'occupe de mettre à jour les paramètres de contrôle (ici, un angle de rotation : `animRotation`) qui sont déterminés à partir de `t`. En synchronisant de la sorte tous les paramètres de contrôle sur `t`, on peut synchroniser tous les mouvements correspondants.

1.2 Application

Si l'on compile et exécute le programme, aucune animation n'est visible. Modifiez le programme afin que le personnage réponde au paramètre de contrôle. Par exemple, faites-le tourner sur lui même.

2 Personnalisation du mouvement

Vous avez maintenant compris le principe d'animation d'un personnage :

1. mesurer le temps et décider d'un pas de temps,
2. synchroniser et mettre à jour les paramètres d'animation,
3. paramétrer les fonctions de dessin.

Ajoutez d'autres types de paramètres (rotation, simple translation, trajectoire grâce à une interpolation d'Hermite (cf : `hermite.h`), échelle, couleur, ...) pour ajouter de la complexité au mouvement de votre personnage.

Conseil : Créez une fonction `draw()` pour chaque membre du personnage et paramétrez-la par les paramètres de contrôle impliqués dans l'animation de ce membre.