



Institut Supérieur d'Informatique, de
Modélisation et de leurs Applications
Université Blaise Pascal
Année universitaire 2016-2017

Synthèse d'images 3D avec OpenGL

TP n° 1

Premiers pas avec *OpenGL* et *ASSIMP*

Objectifs :

Le but de ce *TP* est de se familiariser avec leurs paramètres de la caméra (`gluPerspective` et `gluLookAt`). On réalisera aussi le chargement de fichier de scènes 3D avec la librairie *ASSIMP* 3.

1 On s'amuse (un peu) avec la caméra

On commencera par télécharger sur <http://malgouyres.org/opengl> la petite architecture de base pour une application *SDL2/OpenGL* et on testera (après le *checksum...*) la compilation et l'exécution du programme.

Exercice 1 Faire une fonction de redimensionnement avec réglage de la transformation 2D par `glViewport` pour que le centre de la fenêtre ait pour coordonnées (0, 0), sans changement d'échelle. Régler la caméra avec un angle d'ouverture en γ égal à 50 degrés.

Exercice 2 Modifier la fonction d'affichage pour qu'elle affiche une théière de taille 5. On fera un modèle de caméra fondé sur l'opération *Look At* de manière que la théière soit visible et apparaisse verticale dans le champs de vision. Modifier les paramètres du *Look At* en essayant d'imaginer la position de la caméra et son angle d'ouverture.

Exercice 3 Définir dans le modèle de caméra des membres `mPos` pour la position de la caméra, et `mCentre` pour le point de la direction de visée. Permettre la modification de ces attributs.

Exercice 4 Créer un mouvement (en positif et en négatif) des coordonnées x , y et z de la position de la caméra lors de la pression des touches x , X , y , Y , et z , Z .

Exercice 5 Créer un mouvement des coordonnées x et z du centre de la vue lors d'un mouvement de la souris avec le bouton droit enfoncé.

Exercice 6 Créer un mouvement de la coordonnée y de la position de la caméra lors d'un enfoncement des touches `haut` et `bas` du clavier.

Exercice 7 Créer un mouvement de la coordonnée y du centre de la vue lors d'un enfoncement des touches `gauche` et `droit` du clavier.

Exercice 8 Créer un mouvement d'éloignement ou de rapprochement de la position de la caméra lorsque l'utilisateur enfonce les touches `e` et `r`. On fera (coordonnée par coordonnée) :
$$\text{nouveu_pos} = \text{centre} + 1.1 * (\text{ancien_pos} - \text{centre})$$
pour l'éloignement, et la transformation inverse pour le rapprochement.

Exercice 9 Augmenter l'angle d'ouverture de la caméra lorsque l'utilisateur presse la touche `A`. Diminuer l'angle d'ouverture de la caméra lorsque l'utilisateur presse la touche `a`.

2 Chargement d'une scène 3D avec *ASSIMP*

La librairie `assimp` est une librairie Open Source (disponible sur SourceForge.net) qui permet de charger en mémoire dans une structure de donnée les données d'une scène 3D dans différents formats. Les objets y sont représentés sous forme de maillage. Pour compiler, on utilisera (en plus des *CFLAGS* habituels, les options suivantes :

```
g++ exemple.cpp -o exemple -g -lSDL2 -lGL -lGLU -lassimp
```

Voir la documentation, ainsi qu'un exemple simple d'utilisation sur

<http://assimp.sourceforge.net/>

Vous trouverez sur l'index de cette même URL la documentation (*doxygen*) de la librairie.

Téléchargez les fichiers de scène 3D sur <http://www.malgouyres.org/opengl> (qui datent un peu) ou sur toute autre source de scènes 3D avec la license qui va bien.

Observez les structures de données dans les classes `aiScene` et `aiMesh`, et cherchez à comprendre la structure de maillage, avec ses sommets, ses faces, et les attributs des sommets. Voyez comment parcourir les maillages de la scène.

Observez la documentation (et l'exemple simple d'utilisation) de la fonction `Importer.ReadFile`. Utilisez cette fonction pour charger une scène au format. On pourra par exemple utiliser les flags suivants :

```
aiProcess_FindDegenerates |  
aiProcess_Triangulate |  
aiProcess_SortByPType |
```

Voir la documentation de tous les flags dans `postprocess.h`.

Exercice 10 Réalisez l'affichage *WireFrame* d'une scène précédemment chargée en parcourant le maillage.

Exercice 11 Structurer le code précédent en le concevant comme un *wrapper* pour *ASSIMP*, permettant de charger une scène en créant votre propre interface d'accès aux données de la scène.