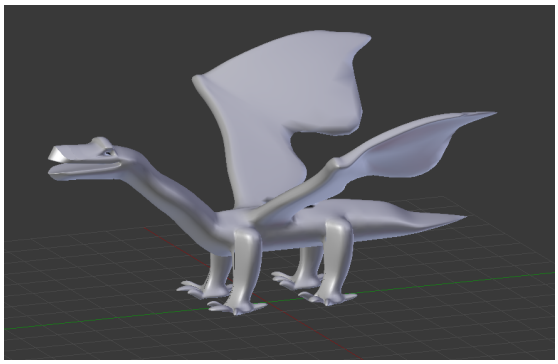
Université de Technologie de Belfort Montbéliard

IN55

Animation d'un personnage 3D avec OpenGL

Florent Jacquet
Romain Thibaud
Antonin Waltz
Superviseur : Fabrice Lauri



Printemps 2016

Table des matières

Table des figures

Chapitre 1

Présentation du projet

Durant ce semestre en IN55, nous avons choisi le projet *Animation d'un personnage 3D* parmi tout ceux proposés. Le personnage que nous avons choisi de modéliser et d'animer est un dragon. En effet, effectuer un rendu naturel de plusieurs mouvements (voler, marcher, s'asseoir) nous a sembler être un challenge intéressant sur une créature de ce type.

Le projet implique d'effectuer le rendu des animations avec OpenGL et d'implémenter une caméra libre dans la scène.

Chapitre 2

Modélisation et armature

Nous avons effectué la modélisation du dragon sous Blender. Son armature se découpe en plusieurs parties indépendantes les unes des autres. Il y a :

— La tête toute entière
— La machoire
— Le cou
— Le corps allant de la base du coup jusqu'à la queue
— Les ailes, indépendantes
— Les pattes, indépendantes également

Chaque partie est composée de plusieurs os reliés entre eux. Chaque os dispose d'une position et d'une orientation qui dépends de celle de son parent. Sous Blender nous avons utilisé des Inverse Kinematics Bones pour les pattes et les ailes en particulier. En procédant ainsi, un mouvement fluide est obtenu.

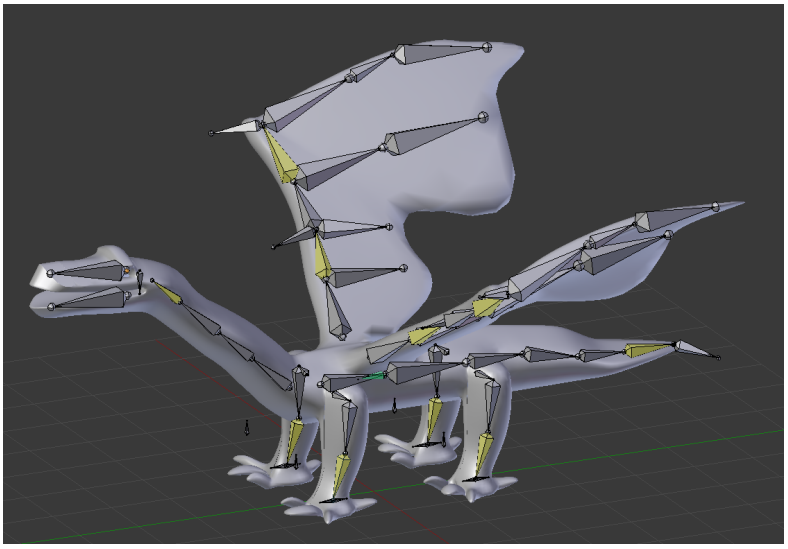


Figure 2.1 – Armature

Chapitre 3

Diagramme de classe

Chapitre 4

Architecture du projet

4.1 Choix technologiques

OpenGl étant une librairie que nous ne connaissions pas, nous avions envie de faire les choses par nous même afin d'avoir une bonne compréhension de la programmation graphique. Initialement, il nous fallait rapidement une interface afin de visualiser le résultat des données que nous parsions. La librairie s'est finalement révélé très pratique pour le chargement des modèles 3D.

4.2 La librairie Asset Import

Asset Import est une librairie libre sous licence BSD permettant d'importer dans une structure de données un très grand nombre de fichiers 3D.

Cela a permis de s'affranchir des opérations bas-niveau de parsage de fichier, ainsi que de la vérification syntaxique du fichier 3D, tout en permettant de faire des tests avec un très grand nombre possible de fichiers comme le Collada, le FBX, le format 3DS, ou encore simplement un fichier OBJ.

Ces fichiers supportant différentes fonctionnalités, il n'est toutefois pas toujours possible de visualiser les animations.

4.3 Nos structures de données

Si la librairie assimp permet de charger le fichier en mémoire dans une structure de données, elle ne permet cependant pas de faire de l'animation squelettale directement. Il faut donc soit faire une fonction capable de parcourir la structure rapidement, afin d'extraire les bonnes informations pour ensuite les afficher, mais cette solution est très couteuse en terme de temps de calcul, puisqu'en plus de faire les calculs d'interpolation, il faut aussi gérer le parcours de structure complexe.

Nous avons donc décider de faire nos propres structures de données, plus simples, et donc moins exhaustives, mais suffisantes pour les besoins du projet.

Des fonctions de chargement sont donc appellées au lancement, récupérant les informations requises grâce à assimp, pour pouvoir ensuite y accéder rapidement et simplement.

4.3.1 Les structures principales

On trouve généralement dans un fichier une scène contenant plusieurs objets.

— *Mesh* : Un *Mesh* est une structure qui contient principalement une liste de *vertex*, de *bones* et de *faces*. L'orientation de ces faces est définie par leur normale, comprise également dans le *mesh*.
— *Vertice* : Un *Vertices* contient un *vertex* avec deux tableaux de taille 4. Le premier détaille les *Bones* qui influent sur lui. Le second mesure le poids d'influence.
— *Bones* : Un *Bones* contient un *bone* et la liste des *vertex* sur lesquels il influe.
— *Face* : Une *Face* contient une liste d'indice qui forment une face.
— *Animation* : Une *Animation* contient une liste de *BoneAnim*. Il y a autant de liste que de *Bone*.
— *BoneAnim* : Une *BoneAnim* contient 3 listes qui correspondent aux clés pour la translation, la rotation et la mise à l'échelle.
— *BoneState* : Une *BoneAnim* contient 3 listes qui correspondent aux clés pour la translation, la rotation et la mise à l'échelle.

4.3.2 Caméra libre

La caméra libre s'utilise avec la souris. Le déplacement sur les axes X et Y s'effectue à la l'aide respectivement du bouton droit et gauche en maintenant le clic. Le déplacement en profondeur sur l'axe Z se fait grâce à la molette.

Chapitre 5

Bilan

5.1 Difficultés rencontrées

Durant ce projet, nous avons été confronté à plusieurs difficultés :
— La prise en main des Inverse Kinematics pour avoir un rendu acceptable des animations dans Blender a été laborieuse.
— Il nous a fallut un temps d'adaptation pour prendre en main et comprendre comment utiliser la librairie Assimp.
— Comprendre comment parcourir de grandes quantités de données à travers des structures complexes pleines de références croisées a également été un frein à l'avancé du projet.
— De manière générale, la gestion de la mémoire en C++.

5.2 Améliorations possibles

Nous aurions pu améliorer notre projet de la façon suivante :
— Nous pourrions ajouter des textures sur le modèle.
— Nous pourrions implémenter un système des gestion de la lumière avec des shaders.
— Nous pourrions améliorer les animations, qui restent rudes malgré tout.
— Nous pourrions optimiser les interpolations et applications des matrices de transformation via des shaders.
— Nous pourrions augmenter la fluidité de la caméra libre et simplifier son maniement par l'utilisateur.

5.3 Conclusion

Pour conclure, ce projet a été l'occasion de découvrir la programmation graphique avec OpenGL et l'utilisation de librairies gravitant autour de la 3D. Travailler sur un dragon et le voir s'animer au fil du temps a rendu le projet particulièrement ludique. Nous avons pu appréhender le processus de modélisation et d'animation d'un personnage pour arriver à la représentation du rendu dans une scène. Nous avons pu constater comment appliquer les concepts vus en cours. Explorer les différentes possibilités offertes par des librairies libres nous a

aussi forcer à aller chercher ce qu'il est possible de faire et comment le réaliser avec ces outils mis à notre disposition.

Chapitre 6

Annexes