

IMN 501 - Animation et rendu temps réel**Devoir 2****1 Mise en contexte**

L'animation par ordinateur trouve ses sources dans l'art, l'informatique et les mathématiques. Il est donc primordial, d'un point de vu scientifique, de maîtriser et comprendre les divers concepts théoriques qui la définisse.

Or, le développement de méthodes scientifiques complexes pour l'animation reste un processus relativement froid, suscitant peu l'intérêt du public sous cette forme. Ce n'est qu'en procédant à une mise en pratique de ces concepts qu'on obtient des résultats tangibles et intéressants à l'oeil.

Dans le cadre de ce travail pratique, vous serez donc emmenés à laisser de côté la théorie et vous pencher vers une problématique courante en animation par ordinateur, soit le chargement et l'animation de modèles 3D complexes.

2 Objectifs

- a Se familiariser avec les techniques courantes de stockage et de compression de modèles 3D animés.
- b Comprendre les processus de transfert et de préparation au rendu impliqués dans le chargement des dits modèles.
- c Savoir procéder à un rendu et une animation performante d'un modèle 3D animé en regard au format de données utilisé pour son stockage.
- d Savoir contrôler le déroulement d'une animation dans le temps en fonction de paramètres, définitions et contraintes externes.

Ce travail pratique consiste à implémenter les éléments suivants :

- a Le chargement d'un modèle MD2. (modèle et texture)
- b Le rendu d'un modèle MD2.
- c L'animation d'un modèle MD2. (avec interpolation linéaire entre les images)
- d Un contrôleur d'animation permettant :
 - (a) L'augmentation/la diminution de la vitesse d'animation
 - (b) La sélection diverses animations dans une liste.

Le format de modèle 3D utilisé pour ce travail est le format MD2. Ce format, développé vers la fin des années 1990, reste un format populaire et très répandu. Il est en effet apprécié pour sa simplicité et pour le potentiel formatif que représente son utilisation. De plus, les détails du format sont particulièrement bien documentés, ce qui permet à un programmeur de l'implémenter rapidement et simplement. Le format sera

brèvement présenté dans les paragraphes qui suivent. Des références plus précises pour son implémentation seront fournies dans la section suivante.

Le format MD2 se caractérise par l'utilisation d'un système d'animation par key-frame classique plutôt qu'un squelette d'animation. L'animation d'un tel modèle s'effectue donc à partir du rendu en séquence des différents maillages (mesh) associés aux différents frames d'animation.

D'un point de vu structurel, un fichier MD2 est constitué d'une en-tête, d'une section contenant les données du modèle et d'une section contenant des données pour le rendu du modèle de façon plus optimale.

Il existe deux modes de rendu d'un modèle MD2. Le premier, le mode de rendu par triangle, requiert l'utilisation de la primitive `TRIANGLE_LIST` d'OpenGL pour le rendu et reste simple au détriment de l'optimalité.

Une seconde méthode de rendu, soit le rendu par commande OpenGL, utilise les primitives `TRIANGLE_STRIP` et `TRIANGLE_FAN` pour accélérer le rendu. Ce mode, exprimé de façon plus complexe dans le fichier MD2, n'est pas exigé pour ce travail. Ceci étant dit, toute équipe fournissant un TP utilisant le mode de rendu par commande OpenGL sera crédité d'un bonus de 20% sur la note du TP, soit 1% de plus sur la note finale de la session.

Certaines fonctionnalités du format MD2 ne sont habituellement pas respectées ou simplement laissées de côté. Pour éviter tout conflit à ce niveau, nous utiliserons les règles suivantes dans ce travail :

- a Le format MD2 spécifie théoriquement les noms des textures qu'il est possible de charger pour le modèle courant. En pratique, très peu de personnes utilisent cette fonctionnalité. Dans le cadre du travail pratique, la texture à charger sera spécifiée séparément, via la façade qui vous sera fournie pour le développement.
- b Si le rendu d'un frame non-défini pour le modèle est demandé, le modèle ne doit simplement pas être rendu. (Par exemple, si on demande le rendu du frame 199 alors que le modèle n'en possède que 173, il ne doit pas être affiché.)
- c Certains fichiers MD2 stockent l'information de début et de fin des animations au sein même du modèle. Cette fonctionnalité n'est pas officielle et n'est donc pas respectée par tous les créateurs de modèle. Dans le cadre de notre travail pratique, une liste séparée d'animations, commune à tous nos modèles, sera fournie automatiquement via la façade au lancement de l'application (via la fonction **AddAnimationData**. Vous utiliserez ces informations d'animation.

Finalement, prenez note que le nombre d'images par seconde dans les animations d'un modèle MD2 est de beaucoup inférieur à la fréquence minimale requise pour percevoir une animation. Votre tâche sera donc, entre autre, d'interpoler linéairement entre les frames fournis afin de garantir une animation fluide.

3 Information sur le format

La référence principale suggérée pour l'implémentation du format MD2 se trouve à l'adresse suivante :

<http://tfcduke.developpez.com/tutoriel/format/md2/>

Des exemples de code pour l'implémentation sont fournis sur cette page. Notez cependant que le code contient quelques erreurs, reste peu optimal et ne tient pas compte des directives particulières énoncées dans la section précédente. Il est conseillé de vous y référer mais déconseillé de le copier/coller dans votre travail pratique. D'un même ordre d'idée, vous devrez utiliser les structures mathématiques déjà disponibles avec le code du TP pour le chargement et l'animation de votre modèle MD2. (Les classes contenues dans le dossier "Foundation" du code.)

D'autres références sur les modèles MD2 sont disponibles sur internet. Elles n'ont cependant pas été utilisées/consultées pour la mise en place de ce travail. L'utilisation de références alternatives se fait donc à vos risques et périls.

Vous remarquerez en consultant la documentation que le format MD2 utilise une table de vecteurs normaux pré-calculés. La table en question est fournie avec le TP et peut être trouvée dans le fichier `MD2Normals.h`, contenu dans le répertoire "Source/MD2".

Pour le chargement des textures, le format utilisé est le format PCX. Une fonction permettant le chargement de ces images est fournie dans le fichier `PCXHandling.h` situé dans le répertoire "Source/Foundation". Pour vous rafraîchir la mémoire, il est suggéré de jeter un coup d'oeil aux fichiers "Cube.h/.cpp" du premier travail pour un exemple de chargement de texture. (La seule différence étant qu'on utilise ici des images PCX plutôt que Bitmap.)

4 Implémentation

Afin de séparer votre implémentation du code fourni avec le TP, l'exposition des fonctionnalités du TP se fait par le biais d'une classe façade définie dans le fichier `MD2Model.h`. Votre tâche sera de produire une architecture efficace derrière cette façade, s'occupant du chargement, du rendu et du contrôle de l'animation du modèle MD2.

La totalité des fonctions contenues dans la façade doivent être implémentées. Leur comportement et la signification des paramètres sont expliqués dans les commentaires associés à la définition de la classe.

5 Barème

La répartition des points est la suivante :

Chargement du modèle 3D	15%
Rendu du modèle 3D	15%
Animation du modèle 3D	30%
Fonctionnalités périphériques	10%
Qualité de la programmation	15%
Respect des modalités de remise	5%

Les fonctionnalités périphériques incluent toute fonctionnalité ajoutée à notre visionneur d'animation. (Changement de la vitesse, changement des animations, changement du modèle.)

La qualité de la programmation est déterminée notamment par la qualité de sa présentation (indentation, noms de variable, commentaires) et par son efficacité (flexibilité, performance, réutilisabilité, robustesse).

6 Modalités de remise

Pour la remise, vous devrez remettre la totalité des fichiers contenus dans le répertoire **"Source/MD2"** (et aucun autre fichier). Lors de la correction, ils seront testés avec le code originalement fourni avec le TP. Vous devez donc vous assurer que le tout compile avec le code d'origine **non-modifié**. Le code doit compiler sur un ordinateur du laboratoire (sur Windows ou Linux).

Vous avez jusqu'au **28 mars 2016 à 22h** pour remettre ce travail pratique. La remise se fera à l'aide du système de demande de fichiers de *Dropbox*, à l'adresse

<https://www.dropbox.com/request/1iuSnAHT5wXxDl1DtNAw>.

N'oubliez pas d'inscrire votre nom et celui de vos coéquipiers dans les fichiers envoyés. Le travail s'effectue en équipe de deux ou trois personnes.

Advenant un retard, une pénalité de 20% par jour de retard sera appliquée sur la note du TP. Ce travail compte pour 5% de la note finale, un bonus de 1% sur la note finale de session sera alloué si les conditions du bonus sont remplies.