

Animation Programming

Procédure du rendu

Start	Fin
Lundi 09/12/2021 -13h	Lundi 02/01/2022 -09h

Nombre d'étudiants par groupe : 2

Rendu :

- Sources : tag GOLD sur git (sur la branche master)
- Sur l'exercice, un zip NOM_NOM.zip
 - Exécutable
 - Screenshots: Au nombre de 5, dans un dossier Screens

4 points seront retirés si les consignes ne sont pas appliquées

Pénalités de retard:

- 5 minutes de retard = -1 points
- 15 minutes de retard = -2 points
- 30 minutes de retard = -4 points
- 1 heure de retard = -10 points
- Plus de retard = 0

Faites donc attention au jour du rendu 😊

Objectifs

Le but du projet consiste à appréhender les bases de l'animation aux travers du modèle de base de unreal.

Ce projet comptera pour **100%** de la note du module.

Compétences

Le projet sera évalué sur les compétences suivantes:

Hard Skills:

- Savoir animer un modèle 3D à partir de données pré-établies
- Connaître les structures de données des animations

Sujet

Vous devez reprendre la base de code existante et, en utilisant les fonctions présentes dans Engine.h, vous devez essayer d'obtenir le résultat de la vidéo Goal.

Ce n'est pas grave si vous n'arrivez pas au résultat final, essayez d'avoir les éléments suivants dans l'ordre

- Dessiner le squelette avec des lignes en utilisant DrawLine (un peu devant le perso afin de voir les lignes)
- Afficher la première keyframe de l'animation walk sur le squelette
- Jouer l'animation de walk en passant en dur de Keyframe à Keyframe
- Coder le hardware skinning pour que les vertices suivent le bones (envoyer les transformations des bones via la fonction SetSkinningPose dans Engine.h)
- Interpoler les keyframes pour que l'animation soit smooth
- Blender les animations de walk et run, en contrôlant les vitesses de play des 2 anims pour qu'elles soient toujours synchronisées (qu'elles aient le même temps normalisé)
- Déplacer un bone manuellement avec des animations hardcodés.

Contraintes

- **Aucune des fonctions dans Engine.h ne doit être appelée durant l'update à l'exception de SetSkinningPose. Les autres fonctions doivent être appelées depuis l'Init uniquement**
- Vous devez donc récupérer toutes les données des animations et du squelette pour les stocker dans des structures adéquates, en faisant un effort particulier à avoir une architecture la plus propre possible (notamment vis à vis de la POO)
- Vous devez par conséquent également coder vos structures Vector3 et Quaternion (pour ces 2 structures, mettez les membres en public)
- **Il faut impérativement que votre code soit le plus lisible et correct possible** (pas de pointeur non gardé, fonctions et classes petites et bien réparties par fichiers, pas de membres non initialisés, etc)
- **Il ne doit y avoir aucune approximation mathématique, tous vos calculs doivent être parfaitement justes** (pas de hack)
- Vous ne devez évidemment utiliser aucune librairie externe, ni copier du code de vos camarades, sous peine d'un 0

Tips

- Les animations sont échantillonnées à 30 images par secondes
- Il faudra modifier le fichier skinning.vs, source code du vertex shader, situé dans le dossier Data/Ressources pour coder le matériel skinning, les données des indices et weights sont valides
- Si vous commettez une erreur dans le vertex shader, l'erreur devrait s'afficher dans la console lorsque vous lancez le programme
- Vous pouvez vous déplacer dans la scène avec les touches ZQSD et la souris
- Pensez à rendre votre rendu agréable (rewind, blinding, pause etc...)
- Il se peut que le programme crash de temps en temps à l'exécution, vous avez eu le cours de Multi-Threading vous devriez savoir pourquoi 😊