

TP Moteur de Jeux

DUPRAT Thomas

LOUX Guillaume

https://github.com/L0ux/Projet_Moteur lien vers le projet

Nous avons fait ce TP à deux car nous pensons c'est une partie importante de notre moteur de jeux et nous voulions réfléchir ensemble à comment serait géré le graphe de scène ainsi que les fonctions de transformation.

Nous n'avons pas réussi à faire fonctionner le fichier Basic.IO pour charger un modèle .off, nous avons donc utilisé les cubes du premier TP pour représenter les différents astres.

GameObject et Transform

Nous avons donc un type GameObject qui contient une liste des GameObjects fils, un nom ainsi qu'un type Transform.

Le type transform sauvegarde la matrice de l'objet et permet l'application des transformations sur notre objet, la rotation, la translation et la mise à l'échelle. Ces opérations fonctionnent.

Nous ne sommes pas encore totalement d'accord sur comment le type Transform fonctionne. Actuellement il sauvegarde la dernière rotation, mise à l'échelle et translation effectuée et retourne au besoin la matrice modifier il est donc nécessaire à chaque opération de prendre en compte les opérations précédentes si on veut les garder.

La lune est par exemple le GameObject fils de la terre qui est lui le GameObject fils du soleil. La position donc de la lune se fait dans par rapport à la terre.

Graphe de scène

le graphe de scène est géré par un GameObject « World » créé au lancement de l'application. Tout GameObject de la scène sera donc un fils de cet GameObject.

Pour le moment nous n'avons pas encore de boucle efficace dans pour parcourir le graphe de scène de manière efficace pour mettre à jour les GameObject.

Garder un rendu temps réel

Pour garder un rendu en temps réel nous voulons tester une autre façon de calculer et de stocker le type Transform pour éviter le plus de calcul possible dans la boucle de rendu. Nous voulons tester aussi d'autres structure GamObject pour prendre en compte plusieurs types d'objets différents car pour le moment tous les objets sont identiques.