

IMN 706: Animation et rendu temps-réel

Travail pratique # 4

Projet au choix

❖ Vivien Gagliano

21 186 690

❖ Lilian Favre Garcia

21 153 421

1 / Contenu

Partie Rendu

La première partie de ce travail est la réalisation d'un moteur de rendu graphique 3D, en partant de zéro. En nous inspirant de tutoriels OpenGL (learnopengl.com et opengl-tutorial.com), nous avons mis au point un moteur possédant les fonctions basiques de rendu : contrôles caméra, importation de modèles 3D sous tous les formats standards, gestion des textures, organisation des objets dans une scène, intégration de shaders (vertex et fragment) personnalisés et système de lumières ponctuelles. Pour créer une scène, il suffit donc d'ajouter des objets en indiquant le modèle et les shaders voulus, et de programmer son comportement désiré, des fonctions utilitaires étant disponibles à cette fin. Le code est de plus architecturé selon plusieurs classes distinctes, rendant le tout très modulaire, et l'ajout de fonctionnalités extrêmement simple.

Partie Animation

En ce qui concerne la partie animation, le moteur gère les mouvements des planètes selon la gravité. Ainsi, toutes les planètes possèdent une masse et une vitesse initiale qui les font prendre certaines trajectoires calculées à chaque frame par l'attraction gravitationnelle. Pour avoir des belles orbites pour chaque planète, nous avons dû essayer plusieurs combinaisons de paramètres de masse, distance au Soleil et vitesse initiale afin d'obtenir des orbites le plus proche possible d'un cercle parfait.

Il est donc possible de modifier les valeurs des masses ou vitesses initiales ou même la distance dans le fichier *scene.cpp* dans la fonction *Populate()*. On peut également préciser que la taille visuelle des objets n'est pas proportionnelle à leur masse, c'est juste un rendu.

Dans la simulation se trouve également une soucoupe volante. Il s'agit d'un modèle 3D créé par nous-même mais qui n'est pas soumis à la gravité. Pour la faire se déplacer, nous avons donc hard-codé des points de passages dans le fichier *Interpolation.cpp* dans la fonction *CreateTable()*. Son mouvement est donc créé à partir d'interpolation comme nous avons fait dans le TP1 de ce cours. Il est donc tout à fait possible de les modifier, d'en supprimer certains ou d'en rajouter d'autres.

2 / Détails

Changer de scène

2 scènes ont déjà été créées avec des modèles 3D organisés, l'une de style cartoon et l'autre se voulant plus réaliste. Afin de passer de l'une à l'autre, il suffit de modifier la variable *cartoon* que l'on peut trouver en haut du fichier *scene.h*. Pour naviguer dans la scène, on utilise les touches WSAD ainsi que L_CTRL et SPACE pour se déplacer, et la souris pour tourner la caméra.

Lancement du projet sur Visual Studio 2019

Attention, si vous utiliser Visual Studio 2019, il y a une erreur possible du type « les outils de génération v143 sont introuvables... » lors du lancement du projet. Pour remédier à cela c'est très simple : aller dans la barre en haut « Projet » → « Propriété de IMN504 – Projet » → « Propriété de configuration » et « général » sur la gauche puis dans le panneau de droite « ensemble d'outils de plateforme » et il y aura écrit « v143 non installé ». Cliquez sur la petite flèche de la même ligne à droite puis sélectionner « Visual Studio 2019 v142 » ou la version installée sur le pc. Puis ok.

3 / Références

Librairie pour charger des modèles 3D :

- Open Asset Importer Library <https://github.com/assimp/assimp>

Modèles 3D des planètes (la licence pour chaque modèle est trouvable dans le dossier resources) :

- Rocket Orbiting Moon, Bastien Genbrugge, <https://sketchfab.com/3d-models/rocket-orbiting-moon-c613bee6fbd041e58a35d777ae87bdcb>
- Saturn, Akshat, <https://sketchfab.com/3d-models/saturn-8fb67d3defd74aaa880df3a08317e641>
- Venus, Akshat, <https://sketchfab.com/3d-models/venus-d497ce25553447f3b7b679110c85cfa1>
- Horizon World, Requinoesis, <https://sketchfab.com/3d-models/horizon-world-08963bf00816473f9eaa2a221f41bdad>
- Coruscant, Mateusz Wolinski, <https://sketchfab.com/3d-models/coruscant-802db255f49e4e1e8d398213ecc371ae>

- Sun, SebastianSosnowski, <https://sketchfab.com/3d-models/sun-9ef1c68fbb944147bcfcc891d3912645>
- Cartoon Bear, SebastianSosnowski, <https://sketchfab.com/3d-models/biological-planet-low-poly-bear-03-2-14be2940b43541168e157a7ccfb60560>
- Cartoon Clutter, NerdMath2000 , <https://sketchfab.com/3d-models/planet-clutter-05965d7aa7744f808c89adb6d0402330>
- Sun from Poly by Google, Iron Equal, <https://sketchfab.com/3d-models/sun-from-poly-by-google-67b0952528b2476c9ccd4d2dae9a3e>
- Lowild's Planet, mdsd95, <https://sketchfab.com/3d-models/lowilds-planet-90cb93056165490caabb5188dd27a5f6>
- Penguin Planet, lbelkin203, <https://sketchfab.com/3d-models/penguin-planet-787ad8cb1bc24beb8d610c6e8e3b73e2>
- Gravitation :
- <https://youtu.be/7axlmc1sxa0?t=84>
- <https://css-tricks.com/creating-your-own-gravity-and-space-simulator/>

Interpolation :

- TP1

Découpage d'une image panoramique en cubes pour la skybox :

- <https://jaxry.github.io/panorama-to-cubemap/>