

HAI819I – Moteurs de jeux

Fabien Caballero

March 15, 2023

Contents

1	Graph de scène	2
1.1	Classes	2
1.2	Textures	2
1.3	Plan infini	2

1 Graph de scène

1.1 Classes

Pour réaliser un graphe de scène j'ai créé une classe Object3D qui est un objet lambda. Ensuite, héritant de cette classe deux sous types existent, le premier un chargeable (Object3DLoaded) qui pourra être un objet load à partir d'un fichier off. Le deuxième est un objet plan (ObjectPlan) qui réalise un plan. Un objet 3D possède un vector et un id de buffer pour les vertices et pour les indices. De plus, un Object3D possède un objet Transform qui est composé d'une matrice locale et d'une matrice monde (Matrice de taille 4x4). Pour appliquer des transformations à notre objet, on modifie la matrice locale.

Ensuite pour avoir une hiérarchie dans notre graphe on ajoute dans notre class Object3D un Object3D parent et une liste de d'Object3D fils, qu'on ajoute grâce à la fonction addChild.

Ensuite une fois tout cela fait, pour pouvoir dessiner nos objets on appelle la fonction updateMeAndChilds de l'Object3D racine, celle-ci actualise la matrice monde en affectant à celle-ci la matrice locale si l'objet n'a pas de parent et la matrice monde de son parent si sa matrice locale s'il a un parent. Puis une fois cela fait, il appelle la fonction updateMeAndChilds de chacun de ses enfants pour actualiser toutes les matrices world des objets du graphe de scène.

Une fois l'actualisation des matrices world faite, on appelle la fonction draw de l'Object3D racine qui va envoyer sa matrice monde, ses vertices et ses indices au vertex shader. Puis on dessine notre objet avec glDrawElements, et une fois cela fini on appelle la même fonction mais sur tout nos enfants, afin de tous les dessiner récursivement.

En faisant cela on obtient un graphe de scène avec une dépendance parent/fils.

1.2 Textures

On fait le même principe pour les textures, chaque Object3D a son buffer texture, celle-ci est load et envoyée au shader dans le draw, et est remplacée par celle de l'objet suivant a dessiné lorsque le précédent a été dessiné et que l'on a appelé récursivement la fonction draw sur les enfants.

Voir vidéo Systemesolaire.mp4

1.3 Plan infini

J'ai réussi à réaliser un plan infini selon un axe, ce qui est facilement adaptable à 2 axes. J'ai réalisé un plan infini selon l'axe Z négatif, pour cela à chaque frame on regarde si la position de la caméra en Z est inférieure au Z maximal des points de notre plan + distance de rendu. Si elle n'est pas inférieure alors on ne fait rien. Si elle l'est on génère un nouveau plan à la suite du précédent et on met à jour la position en Z maximale courverte par notre plan infini.

Voir vidéo planinfini.mp4