

HAI819I – Moteurs de jeux

Fabien Caballero

April 13, 2023

Contents

1	Plan et Graphe de scène	2
2	Object non traversant	2
3	LOD	2

1 Plan et Graphe de scène

Afin d'utiliser les plans que l'on a faits dans un tp précédent, j'ai créé un objet `ObjectPlan` qui hérite d'`Object3D` et qui a des options en plus comme des paramètres de taille, de résolution, et de `heightMap`.

Afin que tout fonctionne, j'ai fait comme pour `ObjectLoaded`, le constructeur ou la fonction `makePlan` créé les vertices, les indices et les textures puis met à jour le vector correspondant de la classe parent dont il hérite. De plus, si le plan a une `heightMap` on change la valeur d'un mode qui lors du draw sera envoyé au vertex shader. Si mode est à 0 qui est sa valeur par défaut alors la position du vertice courant et celle reçue si c'est 1 alors on récupère la `heightMap` et on fait les manipulations pour mettre à jour la position y.

Lors de l'appel à la fonction draw si l'objet est un `ObjectPlan` étant donné que la fonction draw est surchargée dans `ObjectPlan` celle-ci est appelée à la place de celle de `Object3D`. Dans la fonction draw d'`ObjectPlan` on envoie la `heightMap` au vertex shader puis on appelle la fonction draw de `Object3D` qui réalisait tout le reste.

Ainsi, on peut obtenir un plan dans notre Graphe de scène.

2 Object non traversant

Pour cette partie, dans un premier temps, j'ai grâce aux coordonnées du centre de mon objet, j'ai déterminé le carré du plan au-dessus duquel le point se trouve. Ensuite, je détermine dans quel triangle du carré, il se trouve en utilisant les coordonnées barycentriques.

Puis une fois cela obtenu, j'ai déterminé le point projeté ainsi que ses coordonnées de textures grâce à celle des 3 sommets du triangle.

Et enfin, on récupère la valeur de la `heightMap` associée à ces coordonnées textures et on s'en sert pour faire une transformation selon Y.

Problème : mon objet bouge parfois, mais pas fait pas forcément les bons mouvements.

Je n'ai pas trouvé d'où ça vient, et j'ai remarqué que certaines fois mes coordonnées barycentriques sont au-delà de 1 ou en dessous de 0, il s'agit sûrement d'erreurs de calcul ou d'algorithme.

3 LOD

Pour avoir un objet de différentes résolutions selon la distance j'ai utilisé 3 mesh de bunny de différentes résolutions. À l'initialisation, je charge celui avec une haute résolution. Puis dans mon do while je vérifie si ma caméra est à une certaine distance de mon objet, je change mon vector de vertices et d'indices avec le nouveau mesh avec une résolution plus basse.

De plus, si la caméra est à une autre distance plus grande, on en charge un autre ainsi de suite.

J'ai utilisé des valeurs brutes et le calcul de distance seulement selon l'axe Z avec le z de la caméra, mais cela peut être facilement adaptable à un centre autre que le vecteur `vec3(0,0,0)` en utilisant camera-centre à la place de prendre juste la position de la caméra, et utiliser des variables pour les bornes de distance.

Au début, je faisais seulement ça, puis en le mettant en pratique, je me suis aperçu que cela faisait fortement chuter les fps, car le modèle était constamment reload j'ai donc ajouté un int d'état, et dans condition de distance, j'ai ajouté le fait que si il faut charger le mesh loin et que l'int d'état à déjà la valeur correspondant au loin alors on le recharge pas. Il faut donc penser à changer la valeur de l'état lorsque change de mesh chargé.