

Rapport TP 5 GMIN 317

MORIN Matthieu 20135804

POUR FAIRE CE TP, J'AI PRÉFÉRÉ COMMENCER À PARTIR DU TP1.

NOTE : CE TP POURRAIT, COMME CERTAINS PRÉCÉDENTS TPS, MALHEUREUSEMENT, NE PAS COMPILER AVEC QBUILDER. EN EFFET, CET IDE NE RECONNAÎT PAS LES FICHIERS COMME VISUAL STUDIO (« ./MONIMAGE.PNG » NE SEMBLE PAS ÊTRE RECONNU SOUS QBUILDER).

Présentation des nouvelles fonctionnalités

La pyramide et le terrain

Le terrain et la pyramide sont rendus de la même manière. Une texture diffuse est utilisée pour « colorer » le mesh, et pour ajouter des détails, j'utilise une normal map. La normal map me permet d'ajouter des détails au niveau du calcul de lumière. En effet, la pyramide et le sol sont les deux seuls éléments à inclure des calculs de lumière diffuse, pas la sphère (ça n'était pas utile pour ce TP). Le calcul de lumière est, ici, plutôt simple. Il prend en compte trois vecteurs ; la normale au vertex, la position du vertex en cours de traitement, et la position de la lumière. Avec cela, je peux calculer l'incidence lumineuse en ce vertex, pour un point de lumière (scalaire entre les deux normales : normale du vertex, et direction normalisée de la lumière vers le vertex).

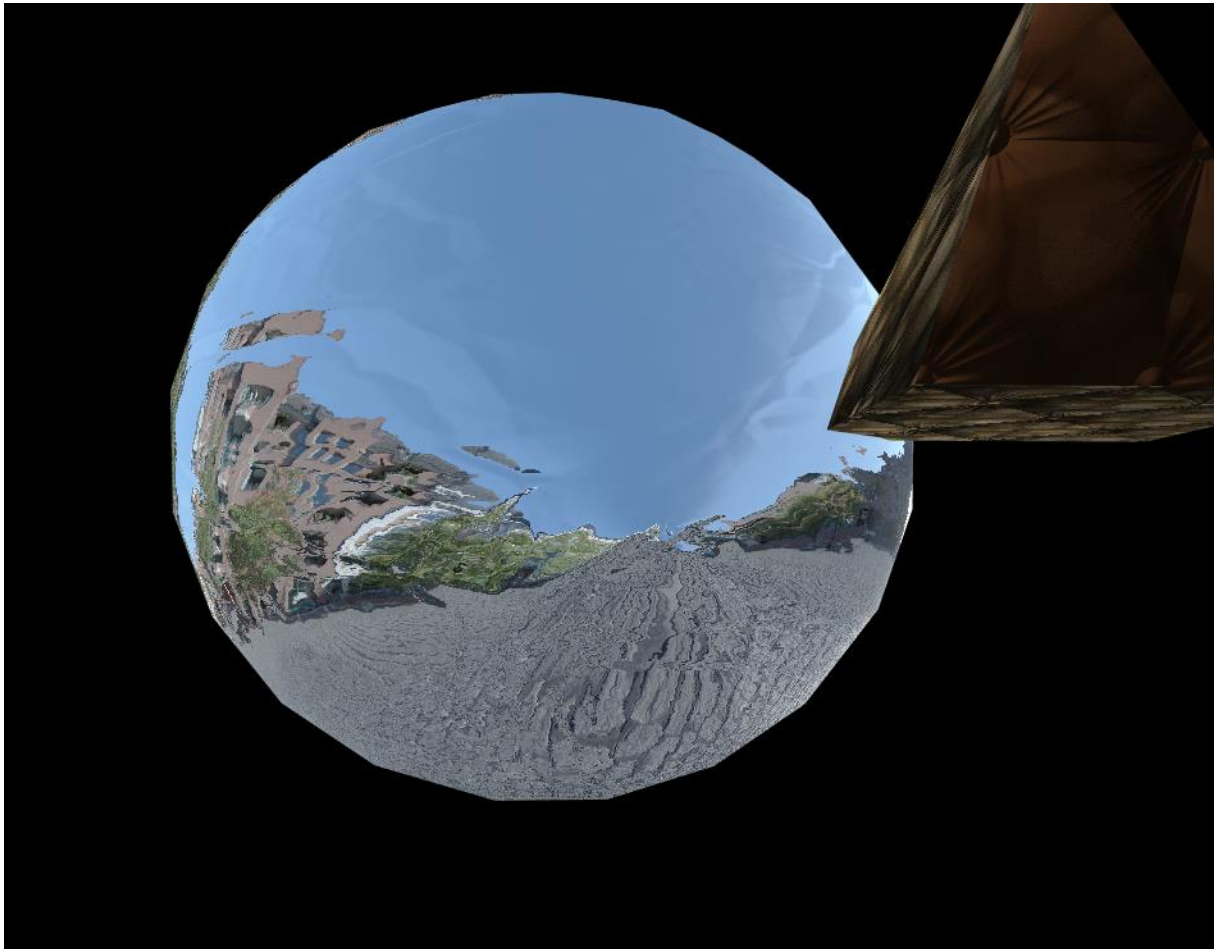
Chaque face de la pyramide affiche une texture différente, mais la même normal map est utilisée. De plus, plus un vertex est bas, plus sa couleur est sombre. On le voit mieux sur le terrain, avec les « canyons ».



La sphère

La sphère est, ici, un mesh au format PLY. Une fois chargé, je calcule automatiquement, pour chaque point, des coordonnées UV. Cependant, avec la formule, il y a un artéfact sur le dernier méridien. Ces coordonnées UV me permettent de placer une normal map, qui me sert à déformer le reflet affiché sur la sphère.

Concernant le reflet en lui-même, ses coordonnées UV sont générées par des calculs en GLSL. Ces derniers prennent en compte la position de la caméra, du vertex en cours de traitement ainsi que de la normale en ce point. Pour appliquer l'effet de déformation, je déforme « faiblement » chaque normale, à l'aide de la normal map.



Parties bonus

Mise à jour du gestionnaire de ressources pour charger les shaders, textures et objets de la scène

Pour cela, il faudrait que chaque objet de la scène soit sauvegardé avec les chemins menant à son shader et ses textures, ou que ces données soient incluses dans la sauvegarde de la scène. Il faudrait utiliser des caractères séparateurs afin de séparer les objets, mais aussi pour séparer les différentes informations liées auxdits objets.

Calculs parallèles sur le GPU

C'est faisable via les compute shaders (bibliothèque OMP), CUDA pour Nvidia, ou DirectCompute. Tout cela a déjà été réalisé lors d'un précédent TP (TP4), avec OMP.