Compte-rendu TP Textures – LAPORTE Laëtitia – 7 décembre 2023
Programmation 3D
Université de Montpellier, Master 1 IMAGINE

1. Introduction

Pour ce TP, j'ai réalisé un plaquage de textures 2D sur un plan 3D, un normal mapping ainsi qu'un environnement mapping. Je n'ai malheureusement pas eu le temps de faire l'effet d'émission, l'occlusion ambiante, le modèle PBR, l'albedo, le metalness et la roughness.
Bonne lecture.

Dépôt git de mon TP : https://github.com/Laeti016934/prog3D-TPsbeugnon-2023

2. Texture Mapping

Tout d'abord, j'ai appliqué une seule texture au modèle plan.
Pour faire cela, il suffit de créer un GLint (m_texture), de l'initialiser en chargeant une texture, d'activer et de lier cette texture pour l'envoyer au shader.
Enfin, j'ai créé une seconde texture de la même manière que la première.

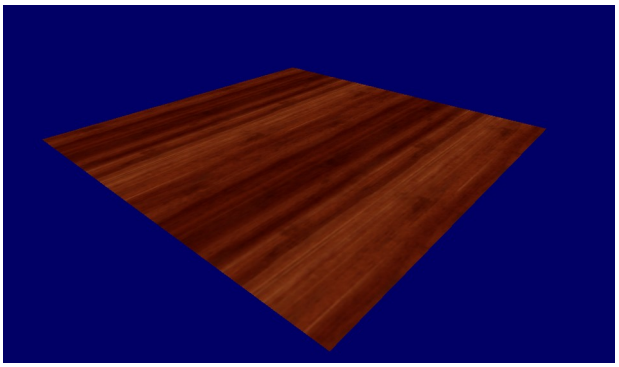

Figure 1: Rendu avec une seule texture (texture bois)

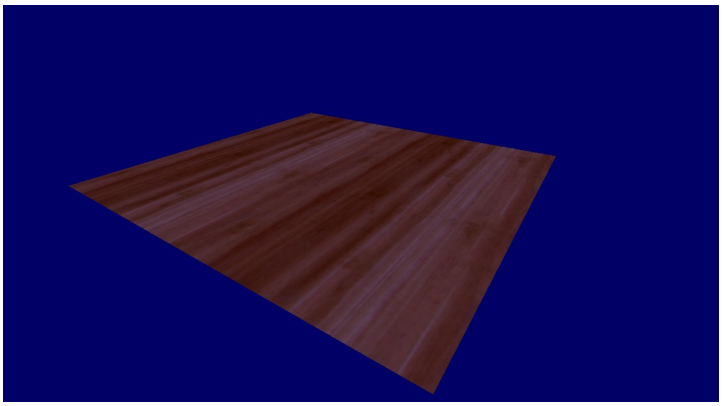

Figure 2: Rendu avec deux textures (bois et normale bois, sans normal mapping)

3. Normal Mapping

Pour effectuer le normal mapping, j'ai repris le modèle de Phong (qui prend en compte la lumière ambiante, diffuse et spéculaire ainsi que la couleur de l'objet) et lui ai transmis la nouvelle normale du monde (par rapport au modèle choisi) que j'ai calculé grâce à la matrice TBN et le vecteur ntangent.

J'ai eu quelques problèmes avec le modèle plan. J'ai consulté mes collègues et il s'avère qu'ils ont eu les mêmes problèmes avec ce modèle.

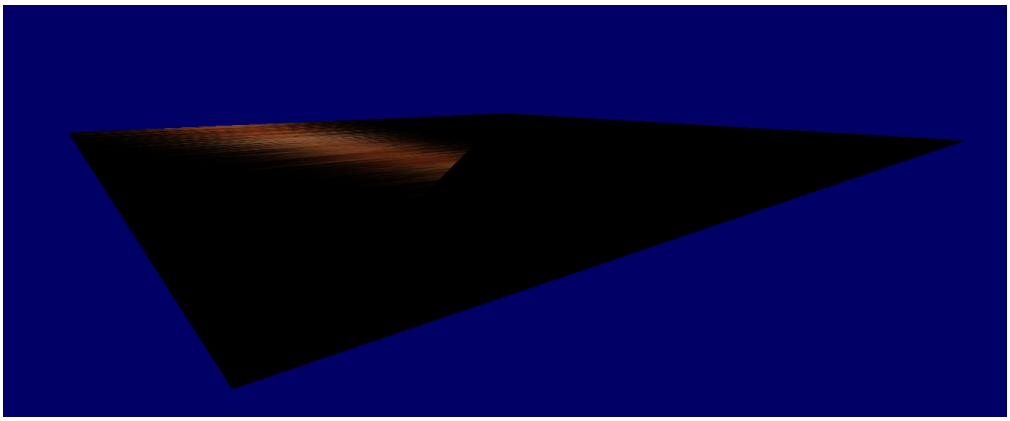

Figure 3: Modèle du Plan où Phong ne fonctionne pas correctement

J'ai donc changé de modèle et essayé avec un modèle de bouteille et un modèle de radio (boombox). Avec ces modèles, le modèle de Phong ainsi que le normal mapping fonctionnent.

Figure 4: Phong et normal mapping Bouteille

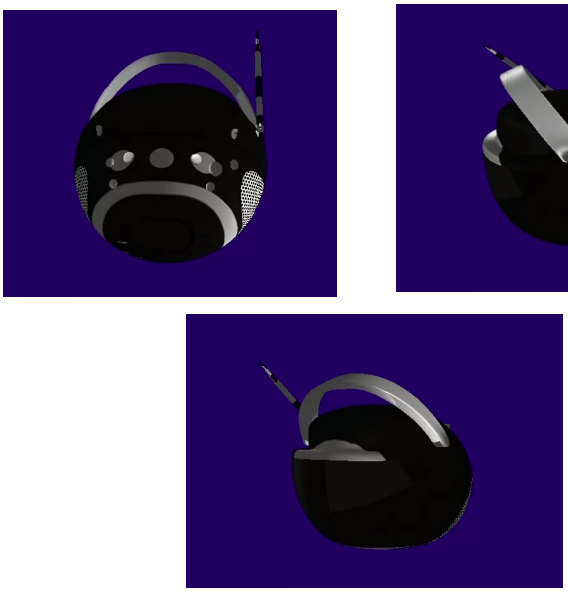

Figure 5: Phong et normal mapping BoomBox

4. Environnement Mapping

Pour faire l'environnement mapping, dans la fonction **initSkybox**, j'ai commencé par créer un cube 3D, *skyboxVertices[]*, qui est composé de la position des sommets de la skybox que l'on veut créer. Je génère et lie ensuite les objets de la skybox (VAO et VBO) et copie les données du tableau de sommets que je viens de réaliser dans des buffers.
Puis, j'ai créé la fonction **loadCubemap** qui charge la texture de la skybox composée de 6 images en format jpg. Je me sers de ces deux fonctions pour initialiser la skybox dans **init**.
Pour afficher cette skybox j'ai créé un shader faisant le rendu de cette skybox (il se trouve dans shaders/skybox).

Figure 6: Modèle Boombox représenté dans la Skybox

Enfin, pour avoir un rendu réflectif d'un objet, il faut aussi réaliser un shader (shaders/reflective) et envoyer la texture de la skybox à ce shader.

Figure 7: Rendu réflectif du modèle BoomBox

5. Conclusion

J'ai eu beaucoup de mal à commencer ce TP. La compréhension du code et le fait de passer du temps sur ce qui n'allait pas dans le rendu du normal mapping intégré à Phong du modèle Plan m'ont pris du temps. Malgré ce contre-temps, j'ai appris et appliqué de nouvelles notions telles que le plaquage d'une texture 2D sur un plan 3D, le normal mapping ainsi que l'environnement mapping.

Je n'ai malheureusement pas eu le temps de me pencher sur la deuxième partie du TP qui nous fait apprendre l'effet d'émission, l'occlusion ambiante, le modèle PBR, l'albedo, le metalness et la roughness.
Ce sujet m'intéresse beaucoup, je vais donc essayer de continuer ce TP après ce rendu (Je créerai sûrement une autre branche sur mon dépôt git pour différencier ce que j'ai fait de ce que je vais faire).

Source des modèles utilisés : https://github.com/KhronosGroup/gltf-Sample-Assets
Source de la texture de la skybox : https://opengameart.org/content/field-skyboxes