

# Compte-rendu du TP4 – Programmation 3D

Université de Montpellier – Master 1 IMAGINE

Loïc Kerbaul

7 décembre 2023

## **Attention :**

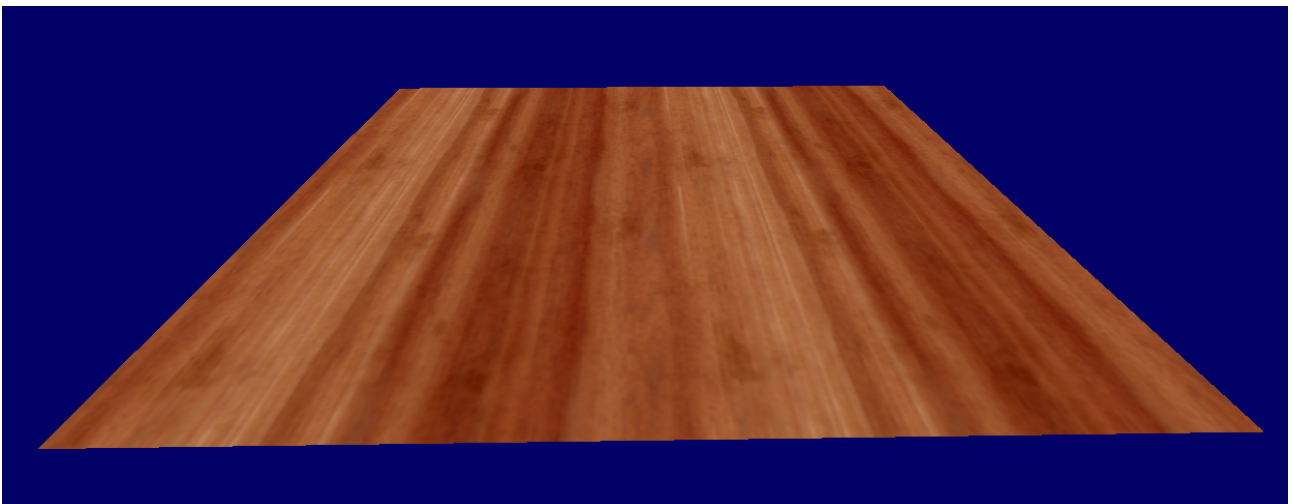
Pour les 2 premiers exercices, j'utilise les shaders suivants : *vertex.glsl*, *fragment.glsl*.  
Tandis que pour l'exercice 3, j'utilise : *vertexSkybox.glsl*, *fragmentSkybox.glsl*.

Pour exécuter le programme, commenter/décommenter le code comme indiqué dans les fonctions suivantes :

- *Material::init()*
- *draw()* (dans le fichier *main.cpp*)

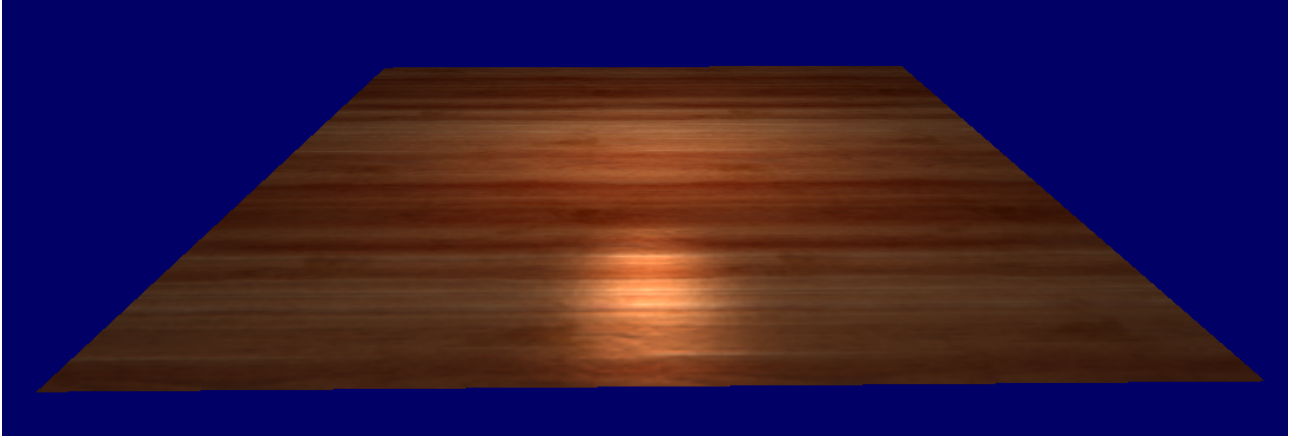
## **Exercice 1 :**

Dans ce premier exercice, le but a été de plaquer une texture 2D sur un plan 3D. J'ai chargé la texture du fichier *TwoSidedPlane\_BaseColor.png* dans la fonction *Material::init*. On obtient alors le résultat suivant :



## Exercice 2 :

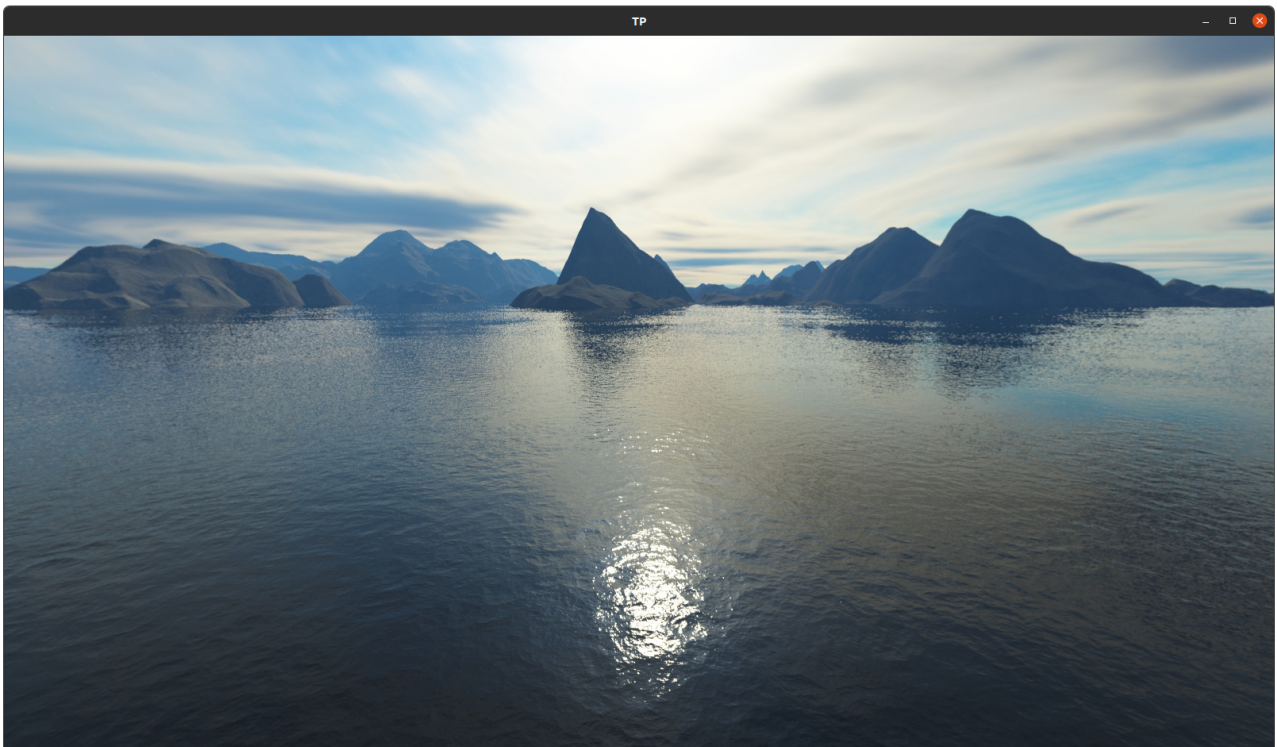
J'ai utilisé une normal map afin de modifier les normales du modèle. Pour voir ces modifications, j'ai calculé la lumière selon le modèle d'illumination de Phong dans le fragment shader (la lumière utilisée est paramétrée dans la fonction *Material::internalBind*). Avec la normal map du fichier *TwoSidedPlane\_Normal.png*, on obtient le rendu suivant :



Sur l'image, on ne voit pas très bien les reliefs sur le plan, mais ils sont bien présents.

## Exercice 3 :

Les 6 textures, chacune pour une face de la skybox (voir dans *template/data/skybox*), sont d'abord listées dans le `std::vector liste_textures` (voir dans *Material::init*), puis elles sont chargées dans *Material::loadCubemap*. Le rendu obtenu est le suivant :



**Avis sur ce TP :**

J'ai trouvé ce sujet de TP très intéressant. Le plus gros problème auquel j'ai dû faire face est le fait que la base de code fournie était assez complexe à prendre en main. Je pense qu'il aurait été plus productif d'avoir un peu plus de temps pour le travailler, car personnellement je n'ai pas pu m'y consacrer autant que je l'aurais voulu.