

# TP4 Moteur

Mateusz Birembaut

April 2025

**Code du TP 4 (Release v4.0.0) car plus de 50 Mo**

## Contents

<b>1</b>	<b>Détection du mesh sous l'objet</b>	<b>2</b>
1.1	Mouvement de l'objet . . . . .	2
1.2	Detection du mesh en dessous de l'objet . . . . .	2
<b>2</b>	<b>LODs</b>	<b>3</b>

# 1 Détection du mesh sous l'objet

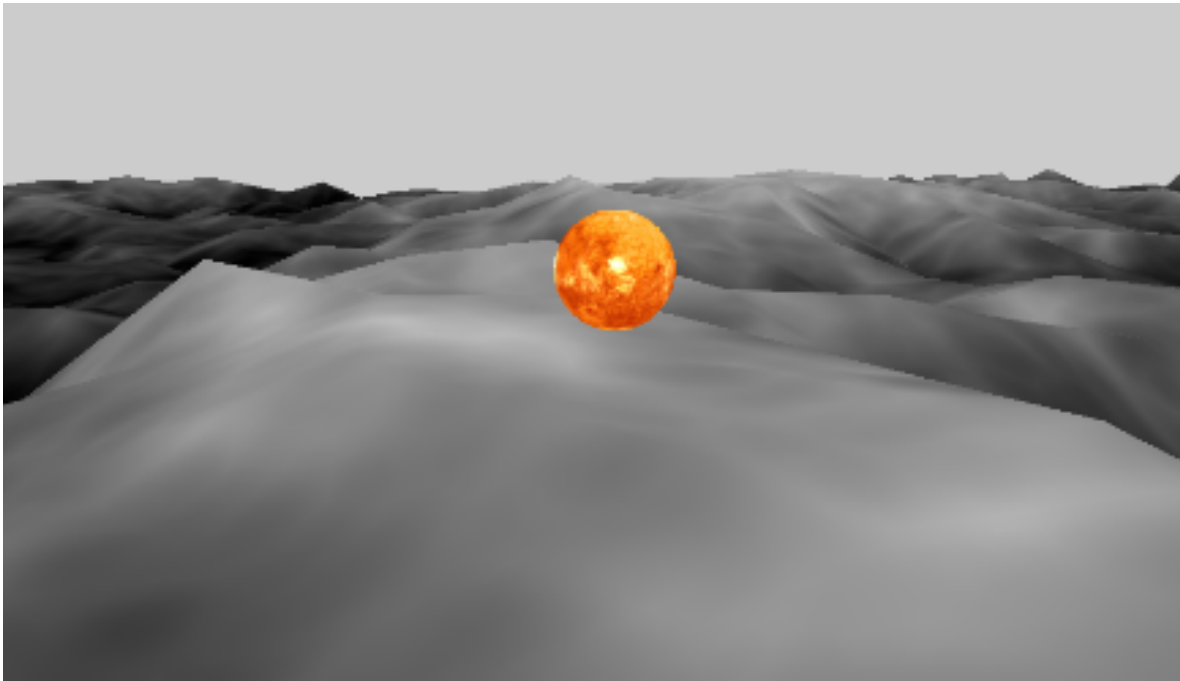
## 1.1 Mouvement de l'objet

Ajout d'une classe "Controller" qui permet de bouger l'objet avec les flèches directionnelles et la touche 1 pour descendre et 2 pour monter.

## 1.2 Detection du mesh en dessous de l'objet

Pour faire la détection du mesh sous l'objet :

- Je lance un rayon qui est envoyé depuis la position de l'objet dans le monde vers le bas (0, -1, 0). Pour ne pas tester avec tous les objets, j'utilise mon octree de la scène qui me servait pour le partitionnement de l'espace et le frustum culling.
- On obtient un point d'intersection dans le repère monde.
  - Si l'objet a une heightmap : J'applique l'inverse de la matrice modèle de l'objet le plus proche au point pour l'avoir dans le repère de l'objet.
  - Je transforme le point en un u, v pour sampler la heightmap puis je fais une interpolation bilinéaire entre les 4 points les plus proches pour avoir la hauteur au point d'intersection.
  - (le problème avec cette méthode, est que si je n'ai pas assez de triangles, je peux trouver une hauteur différente de celle d'où sont affichés les triangles, je n'ai pas eu le temps de modifier, mais je ferais plutôt en sorte de récupérer la hauteur de mes trois sommets en samplant la texture au niveau des sommets et faire une interpolation entre ces trois valeurs pour avoir la hauteur réellement affichée).
- Si la différence de hauteur entre le point d'intersection et celle de l'objet est inférieur à la taille de l'objet en y, on fait : `go.transformation.translation.y = intersectionPoint.y + sizeY;`

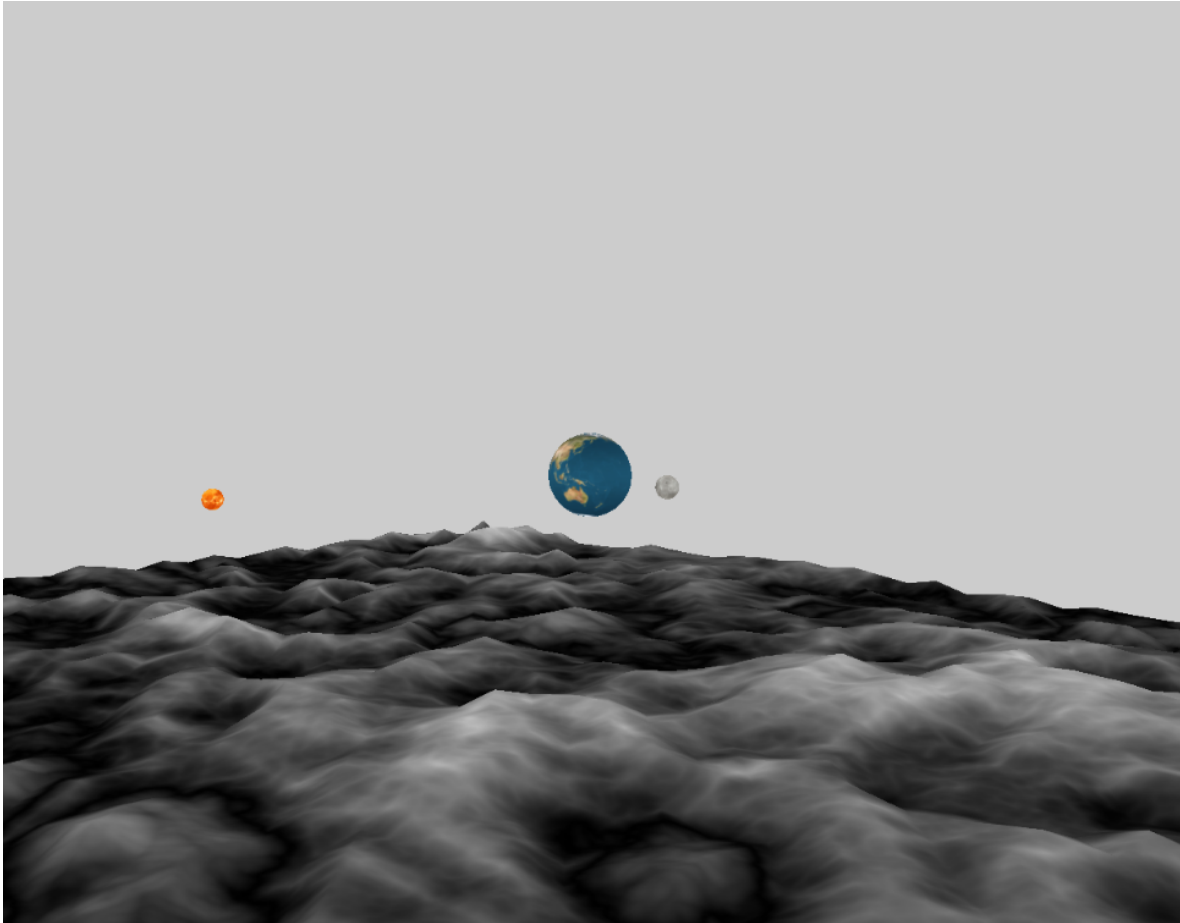


L'objet ne peux pas descendre plus bas

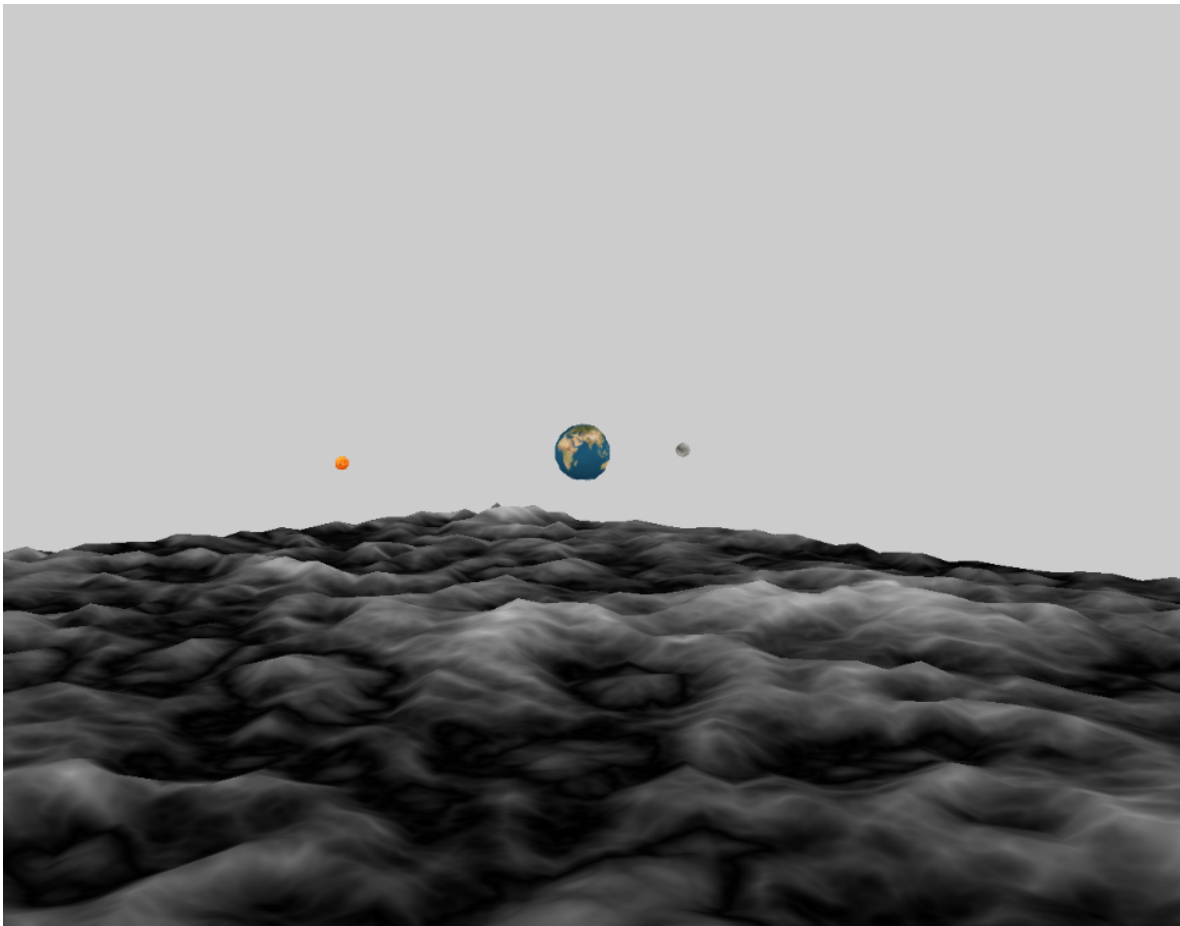
## 2 LODs

```
//earth.mesh = sphereMesh;  
earth.lods->addMesh(20, sphereMesh);  
earth.lods->addMesh(30, sphereLOD1);  
earth.lods->addMesh(50, sphereLOD2);
```

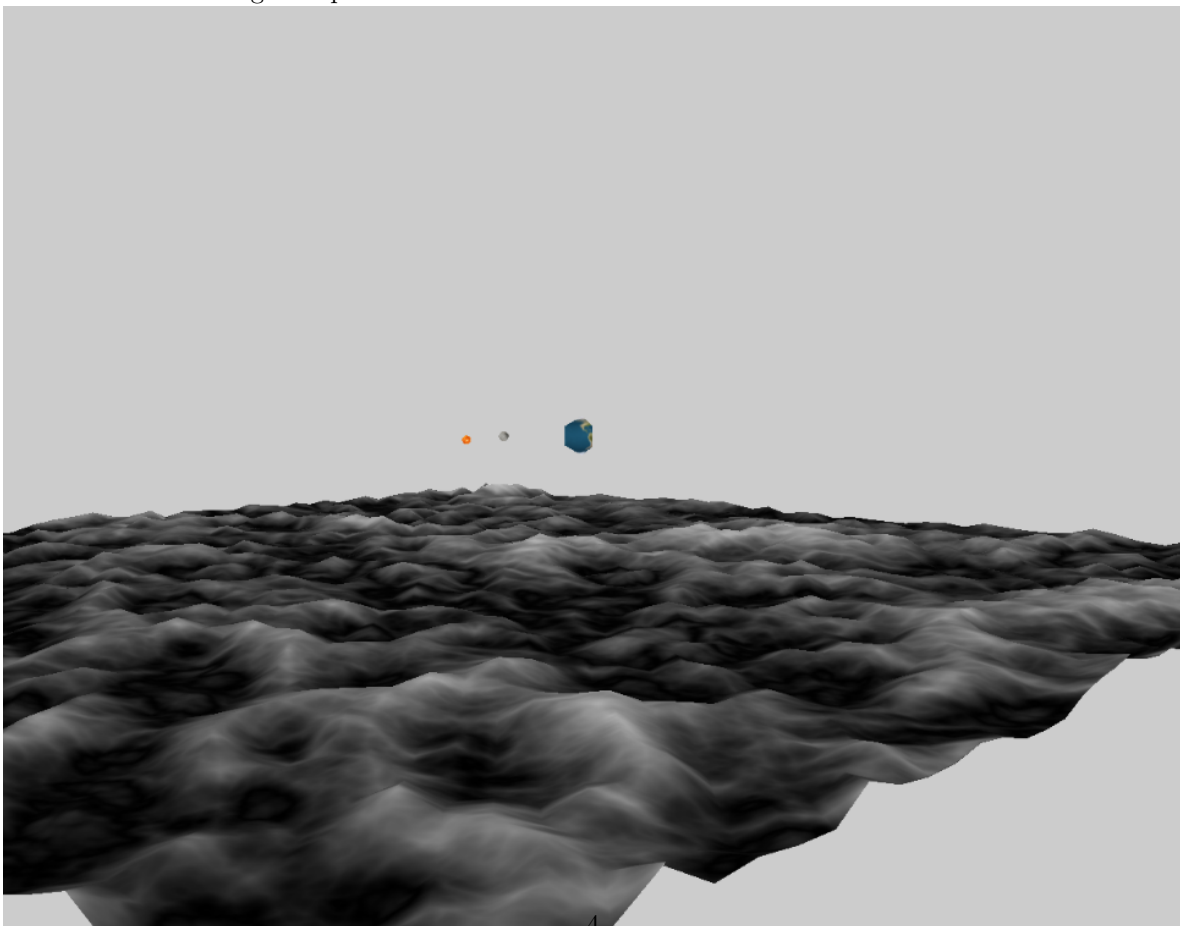
Ajout des lods, paramètre 1 : la distance jusqu'à laquelle le mesh est affiché, paramètre 2 : pointeur vers le mesh.



Affichage de sphereMesh si la distance avec la caméra est inférieure à 20



Affichage de sphereLOD1 si la distance avec la caméra est inférieur à 30



Affichage de sphereLOD2 sinon