Programmation graphique 3D
M1 Informatique I3D - S1 Automne 2022–2023

Projet de programmation graphique 3D

Introduction

Dans ce projet, vous allez proposer **la modélisation et le rendu d'un payasage en extérieur**. Il vous est demandé de créer une scène 3D qui contiendra l'ensemble des fonctionnalités listées dans le sujet ci-dessous. En particulier, on s'attardera à réaliser un rendu d'eau "réaliste". Vous êtes libre d'ajouter toute fonctionnalité que vous shouaitez, de choisir ce que contiendra votre scène (bord de mer, montagne, planète imaginaire, ...) et de son rendu (gestion de la caméra ...). De plus, **une partie de la notation sera accordée à votre force de proposition**, soyez créatifs !

Comme pour les TP, vous réaliserez votre programme avec l'API OpenGL. Vous devrez également utiliser la librairie Easycppogl qui vous a été fournie (pensez à regarder la doc dans le fichier README !). Je prêterai attention à la propreté de votre code (lisibilité, commentaires, refactoring ...).

Modalités de rendu : Ce projet est à rendre pour le **16/12/22 à 23h59 au plus tard**. Vous le déposerez sur la page moodle *GraphicsProgramming*, dans le dépôt qui a été prévu pour cela. Je vous demande de **déposer juste une archive** contenant votre/vos code(s) C++ et shaders GLSL, toutes les ressources nécessaires, ainsi qu'un **court rapport** (2 à 4 pages pdf) pour donner toutes les explications que vous jugerez nécessaire. **Vous nommerez cette archive nom_prenom.zip** (ou tar.gz comme vous voulez).

A coté de ce sujet vous trouverez :

- Un répertoire textures/ qui contient des textures qui vous serons utiles
- Un fichier *color_map.cpp* qui contient une carte de couleur.

1 Partie I : Mise en place de la scène

1.1 Modélisation du terrain

Ici vous êtes libre de modéliser le terrain comme vous le souhaitez. En partant d'une grille régulière 2D, il suffit ensuite de définir une hauteur pour chaque sommet de la grille. Vous pouvez le faire par exemple :

- en utilisant une carte de hauteur
- avec de la génération procédurale
- autre ...

Remarque : Votre terrain devra comporter au moins une zone avec de l'eau (mer, rivière, lac ...).

Aide si vous voulez faire de la génération procédurale : Vous pourrez utiliser un bruit pour générer la hauteur de chaque point de votre terrain. En TP on a vu une méthode simple pour cela mais on peut faire beaucoup mieux !
Regardez ici pour vous inspirer : https://thebookofshaders.com/11/?lan=fr

1.2 Eclairage du terrain

Vous devez ajouter de l'éclairage à votre scène en implémentant une **BRDF de Lambert** pour la diffusion de la lumière.
Pour définir la couleur de diffusion de votre terrain là encore vous avez plusieurs possibilité :

- **Carte de couleur** : stocker une "palette" de couleur dans un tableau à envoyer en uniform (on peut récupérer un tableau static en uniform dans un shader). Il suffit ensuite d'accèder à ce tableau en fonction de la hauteur d'un fragment.
- **Texture** : plaquer une unique texture à votre terrain. Soit avec une texture de grande résolution (couteux...) soit en faisant du tiling (répétition d'une texture) = il suffit d'indiquer des coordonnées de texture qui dépasse 1.0 en x et en y en multipliant les coordonnées de textures [0, 1] par un facteur, et d'indiquer à la création de la texture que celle-ci doit se répéter sur les 2 axes avec les paramètres GL_TEXTURE_WRAP_S/WRAP_T.
- **Multi-texturing** : vous pouvez même utiliser plusieurs textures en combinaison avec une "blend map". La blend map contient n couleurs qui correspondent aux n textures à utiliser. On échantillonne d'abord la blend map pour récupérer la couleur qu'elle stocke. On échantillone ensuite les n textures et on applique un facteur à chacune d'entre elles selon ce que l'on a récupérer dans la blend map.

1.3 Carte d'environnement

En arrière plan de votre scène vous devez gérer l'affichage d'une carte d'environnement (ou skybox).

2 Partie II : Rendu de l'eau

Rappel : Votre terrain devra comporter au moins une zone avec de l'eau (mer, rivière, lac ...).

Aide : Vous pouvez partir avec un quad (juste 2 triangles) qui représente un plan où se trouve la surface de l'eau. Il faudra ensuite un ou des shaders pour générer le rendu "réaliste" de l'eau. Le tout dans une/des texture(s) à plaquer sur le plan d'eau.

2.1 Réflexion et réfraction

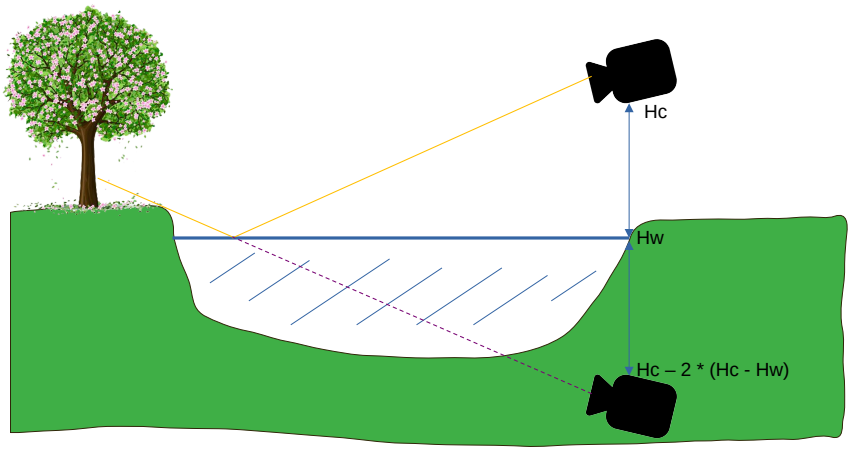
La surface de l'eau possède des propriétés physiques de **transparence** et de **réflectance**. Pour faire le rendu de la surface de votre plan d'eau en simulant ces propriétées, vous devez calculer :

1. la **réflexion** de la lumière, émis par l'environnement extérieur, sur la surface
2. la **réfraction** venant du terrain en dessous de la surface.

Pour cela, vous devez faire le **rendu de votre scène dans deux FBO** (un pour la réfraction et un pour la réflexion) stockant chacun une texture.

- Pour la réfraction, il faut faire le rendu de la scène **uniquement pour la géométrie se trouvant en dessous du plan de la surface de l'eau** et avec la position "normale" de la caméra.
- Pour la réflexion, il faut faire le rendu de la scène **uniquement pour la géométrie se trouvant au dessus du plan de la surface de l'eau** et avec une caméra se trouvant à une hauteur $Hc\_refract = Hc - 2 * (Hc - Hw)$ avec Hc la hauteur initiale de la caméra et Hw la hauteur du plan qui définit la surface de l'eau.

Aide : Utilisez l'instruction "**discard**" du fragment shader pour ne pas afficher un fragment !

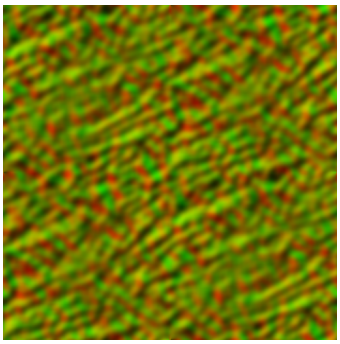


Le résultat sur le quad du plan d'eau doit être un **mix entre la réflexion et la réfraction**.

Aide : Attention aux coordonnées des textures ! Il vous faut les "normalized device coordinate" (NDC) pour échantillonner correctement ces textures provenant du rendu de la scène. Pour cela, dans le fragment shader, il faut récupérer la position dans le clip space (proj * view * model * pos) et utiliser la position 2D (xy) divisée par la composante w du vecteur. Ces cordonnées sont exprimées sur l'interval [-1; 1], il faut ensuite les ramener sur l'interval [0, 1] pour avoir les NDC. Attention également, la coordonnée de texture y pour échantillonner la texture de réfraction doit être inversée.

2.2 Distortion

Pour donner un effet de distortion de l'eau vous devez utiliser une **carte de distortion** (distortion map). On la stocke dans une **texture dont les 2 premiers cannaux de couleur stockent une information de "déplacement" en 2 dimension (vec2)**. Vous devez appliquer cette distortion aux coordonnées qui permettent d'échantillonner les textures de réflexion et de réfraction vu précédemment.



- Coordonnées de texture : il faudra passer de l'interval [-1, 1] de la position du sommet à l'interval [0, 1] de la texture.
- Application d'un **tiling** : répétition d'une texture pour éviter d'utiliser une texture de très grande résolution. (voir 1.2)
- Echantillonnage dans le fragment shader. Dans la texture de distortion on récupère une information stockée entre 0 et 1 (normalement une couleur), donc ici on ramenera cela dans l'interval [-1, 1] pour représenter une coordonnées 2D à l'écran. Il sera aussi surement nécessaire de diviser cette valeur pour ne pas avoir une trop grande distortion.
- Animation : faites bouger la distortion en fonction du temps (aide : pour un effet plus réaliste vous pouvez aussi échantillonner 2 fois la carte de distortion avec des coordonnées différentes et additionner les 2).

2.3 Effet de Fresnel

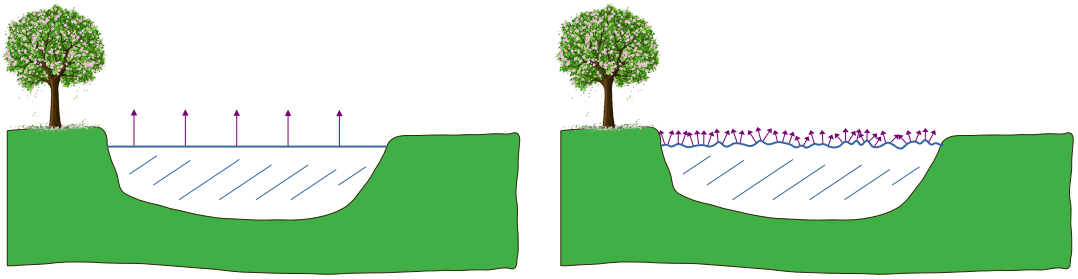
L'effet de Fresnel est un comportement physique de la surface d'un matériau. Plus on regarde la surface avec un angle rasant, plus l'indice de réflectance de celle-ci est élevé. Avec un angle perpendiculaire à la surface, on ne voit aucune réflexion.

Ici vous allez seulement simuler cet effet avec un calcul simplifié du **coefficient de réflexion** qui sera égale à l'angle entre la direction vers l'observateur (caméra) et la normale à la surface.

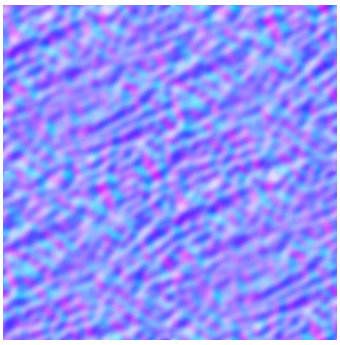
Servez-vous de ce coefficient pour appliquer plus ou moins de réflectance à la surface de l'eau.

2.4 Surface spéculaire

Pour finir, vous devez ajouter un effet de **spécularité sur la surface de l'eau**. Comme vous le savez, il faut prendre en compte la **normale** pour calculer cet effet, mais le plan de l'eau possède une normale = (0, 1, 0) en tout points de la surface ...



Vous allez simuler une surface moins régulière à l'aide d'une **carte de normale** (normal map). Celle-ci se stocke dans une texture qu'il faut échantillonner pour récupérer la normale (vec3) en un point de la surface. En échantillonnant (voir aide), vous récupérez un vec3 (R, G, B) et vous obtiendrez la normale vec3 (X, Y, Z) avec (X = R, Y = B, Z = G). De plus, pour le Y, on veut que la normale soit positive (pointe vers le haut) mais pour un rendu plus réaliste, on veut que le X et le Z puisse être négatif, il faudra donc faire un changement d'interval [0, 1] vers [-1, 1] pour X et Z.



Aide : La carte de normale qui est fournie pour ce projet coincide avec la carte de distortion. Prenez donc le vec2 récupéré en échantillonnant la carte de distortion comme coordonnées de texture pour échantillonner la carte de normale.

Avec cette normale, **calculez la spécularité**. Il vous faudra connaitre la **direction de la lumière** (à partir de la position d'une source lumineuse, le soleil) et la **direction de la vue** (à partir de la caméra). **Attention :** cette normale est exprimée dans l'espace monde !

3 Partie III : D'autres idées ?

Vous êtes libre d'ajouter les éléments que vous voulez et les fonctionnalitées que vous souhaitez ! Un personnage qui se déplace ? des arbres ? un effet jour/nuit ? du brouillard ? des ombres ? un soleil ? une atmosphère ? un effet de lentilles avec le soleil (halos) ? ...