POLYTECHNIQUE MONTRÉAL

LE GÉNIE EN PREMIÈRE CLASSE

INF2705 Infographie

Spécification des requis du système

Travail pratique 2

Stencil, mode sélection, clôtures multiples

Département de génie informatique et génie logiciel
École polytechnique de Montréal
Hiver 2019

Table des matières

1 Introduction

Ce document décrit les exigences du TP2 « *Stencil, mode sélection, clôtures multiples* » du cours INF2705 Infographie.

1.1 But

Le but des travaux pratiques est de permettre à l'étudiant de directement appliquer les notions vues en classe.

1.2 Portée

Chaque travail pratique permet à l'étudiant d'aborder un sujet spécifique.

1.3 Remise

Faites la commande « `make remise` » afin de créer l'archive « **INF2705_remise_TPn.zip** » que vous déposerez ensuite dans Moodle. (Moodle ajoute automatiquement vos matricules ou le numéro de votre groupe au nom du fichier remis.)

Ce fichier `zip` contient le fichier Rapport.txt et tout le code source du TP (`makefile`, `*.h`, `*.cpp`, `*.glsl`, `*.txt`).

2 Description globale

2.1 But

Le but de ce TP est de permettre à l'étudiant d'utiliser les plans de coupe et de se familiariser avec les fonctions de manipulation du tampon stencil telles que `glStencilFunc()`, `glStencilOp()`.
Il permettra également de mettre en pratique l'utilisation du mode de sélection, d'un nuanceur de géométrie et l'utilisation de plusieurs clôtures.

2.2 Contexte

L'usine à poissons

Une entreprise qui produit du poisson en conserve souhaite modifier génétiquement une espèce existante de poisson afin de produire des poissons avec un corps absolument cylindrique, un squelette à fleur de peau et des yeux jaunes ressemblant à deux petits disques. Selon l'entreprise, cette forme serait idéale pour la préparation et la mise en conserve : on arrache simplement la peau et le squelette du poisson dans la même étape, on tranche sa chair sans arêtes et on le met facilement en conserve !

De plus, afin de bien récolter les poissons dans l'aquarium, une région de l'aquarium permet de montrer les poissons sous rayons X (en fil de fer), tandis que le fond de l'aquarium peut être remonté pour faciliter le dragage.

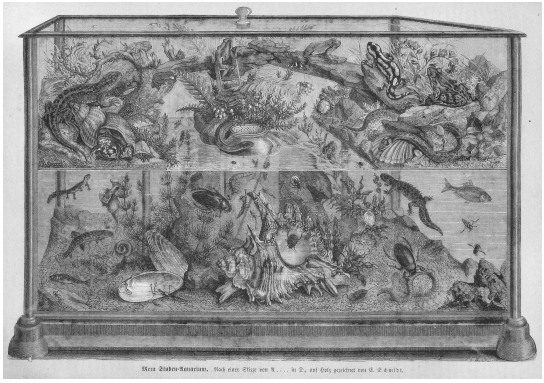

FIGURE 1 – Aquarium

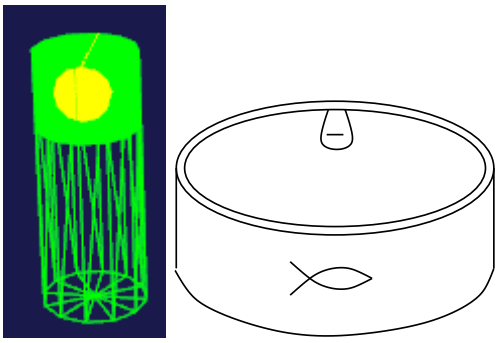

FIGURE 2 – Mise en conserve facile d'un poisson cylindrique

Cette histoire abracadabrante serait évidemment une gigantesque erreur de la nature... Convenons simplement que je « mets la table » pour ce TP en tentant d'expliquer l'existence de poissons au corps cylindrique !

Dans le même thème, à découvrir ou à réécouter : « La Maman des Poissons » de Boby Lapointe. :) https://www.youtube.com/watch?v=0EIdDTBvO3o

L'affichage de base

Le programme de base fourni affiche un aquarium contenant des poissons cylindriques. Les poissons se déplacent selon l'axe des X (figure 3). Chaque poisson est affiché à partir de deux cylindres imbriqués : un cylindre vert pour le corps et un autre cylindre jaune pour les deux yeux (figure 4).

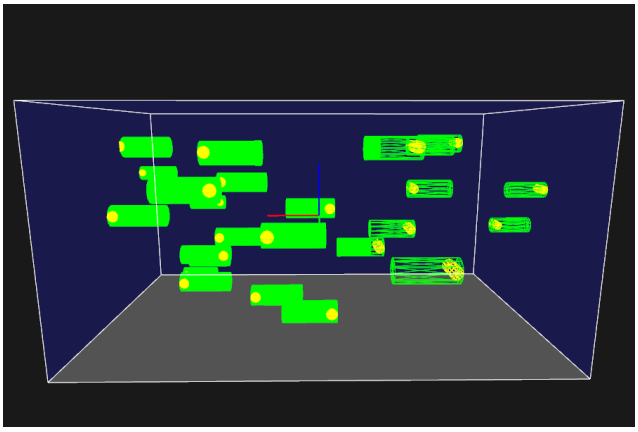

FIGURE 3 – L'aquarium et ses poissons

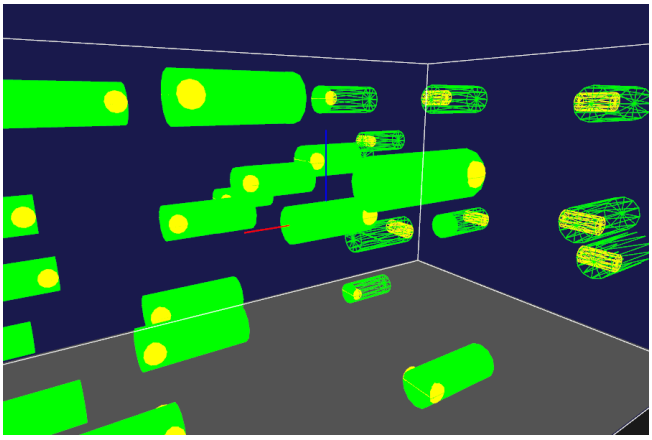

FIGURE 4 – Les yeux des poissons

À terme, les poissons seront coupés par le plan de dragage, la scène pourra être assombrie au loin et la couleur des poissons variera du vert au bleu entre la « tête » et la « queue » (Figure 5).

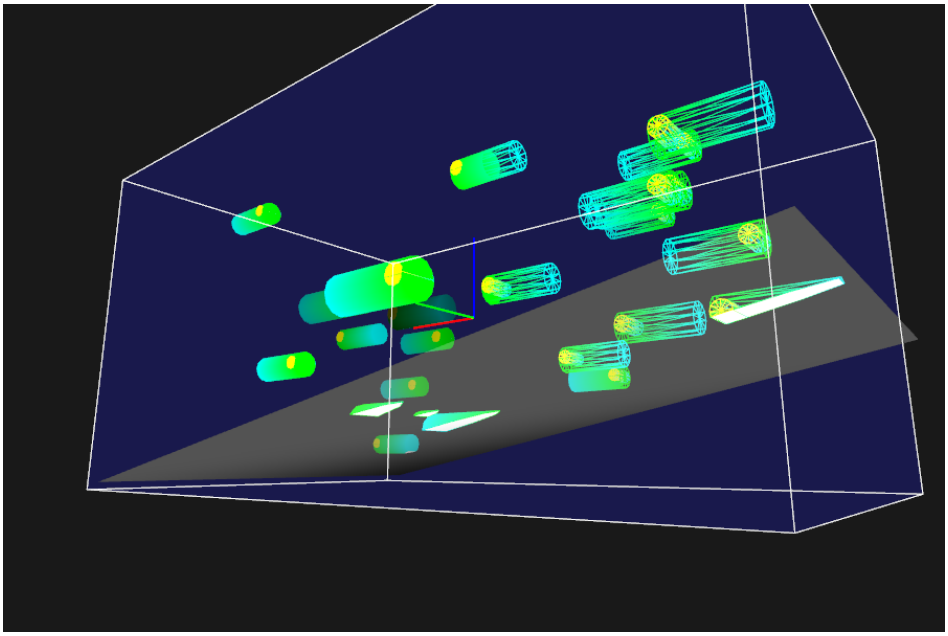

FIGURE 5 – L'aquarium et ses poissons en version finale

2.3 Travail demandé

Partie 1 : l'utilisation de plans de coupe et du tampon de stencil

Un premier plan de coupe vertical, situé à « x = -4 », sera utilisé pour délimiter deux régions dans l'aquarium : affichage normal du poisson, en plein, lorsque « x > -4 » ou affichage du squelette du poisson, en fil de fer, sinon. (Voir figure 6 ci-dessous)
Indice : la scène devra être affichée deux fois de façon différente.

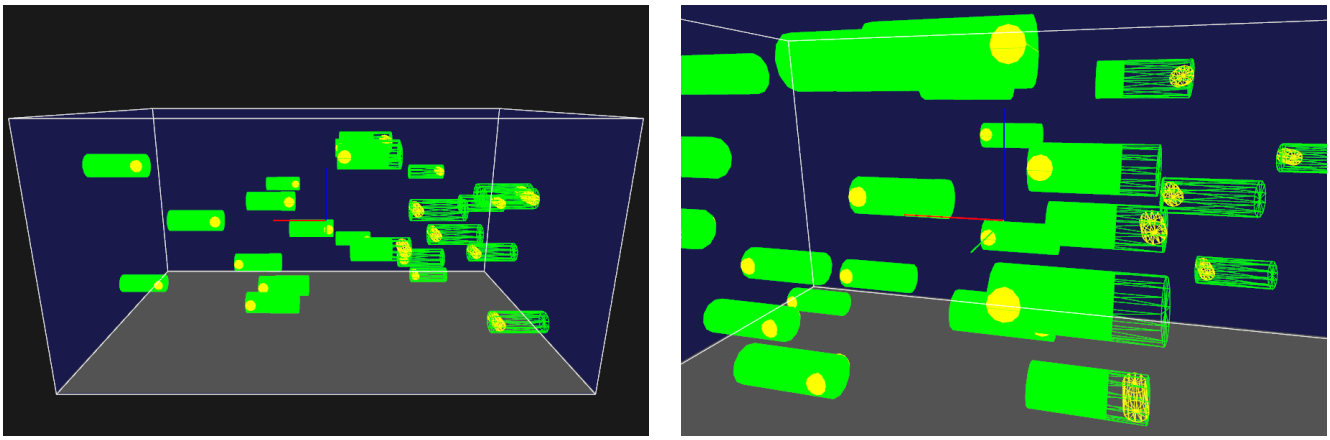

FIGURE 6 – Poissons normaux et sous rayons X

Un second plan de coupe, pour le dragage, découpera les poissons par le fond. On peut déplacer ce plan de dragage selon la direction de sa normale ou l'incliner afin de varier l'endroit où la coupe se fait (figure 7).

Le quadrilatère représentant ce plan de dragage sera créé avec deux VBO (sommets, connectivité) et sera affiché en transparence (alpha = 0.25) et à la bonne taille dans l'aquarium en utilisant la fonction `glDrawElements()`. Notez que l'utilisation de cette fonction est avantageuse pour éviter la duplication des informations à un sommet, surtout si on y spécifie des coordonnées, une couleur, une normale pour l'illumination ou des coordonnées de texture comme nous le ferons plus tard.

D'autre part, pour que le poisson coupé constitue toujours un volume fermé, on tracera aussi un quadrilatère opaque blanc, correspondant au plan de dragage qui sera affiché conditionnellement, en faisant une vérification du stencil de manière à ce que le quadrilatère ne soit dessiné que sur la zone de coupe. Les figures 7 et 8 représentent l'effet du plan de coupe sur la scène.

Pour ce faire, on peut afficher les poissons, une fois en plein et une autre fois sous rayons X, tout en découpant l'affichage à chaque fois par le plan de dragage. L'affichage à l'écran sera alors quasiment complété : il ne manquera que le plan opaque blanc fermant les poissons coupés.

Ensuite, pour remplir et utiliser le stencil pour compléter le dessin, on peut tracer la scène deux fois tout en désactivant son rendu à l'écran : une première fois en ne traçant que les faces avant et une seconde fois en ne traçant que les faces arrière tout en modifiant le stencil de façon différente à chaque fois. Une possible stratégie pour remplir le tampon de stencil est montrée à la figure 9 et les fonctions pertinentes sont présentées à l'annexe D.

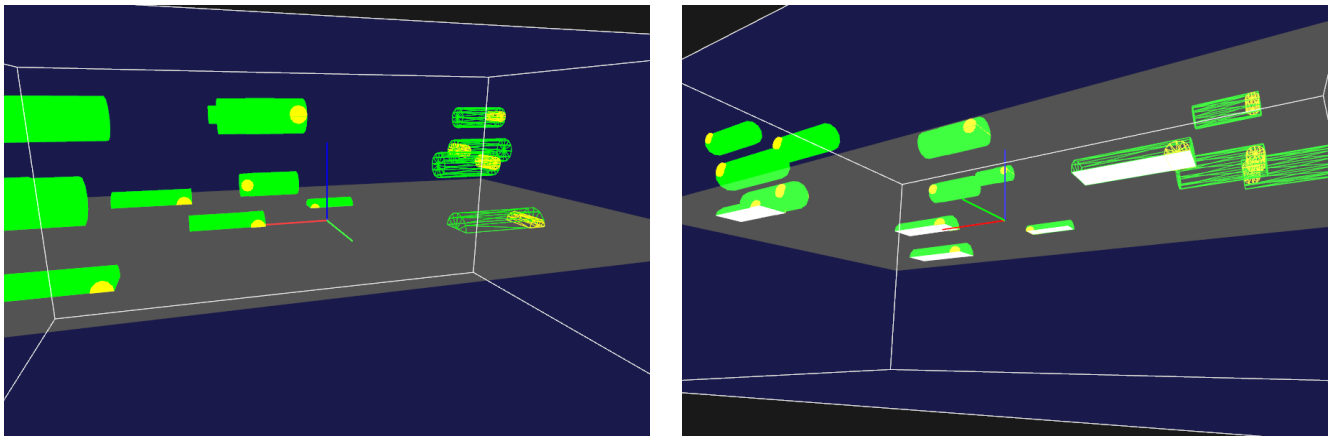

FIGURE 7 – Poissons coupés par le plan de dragage

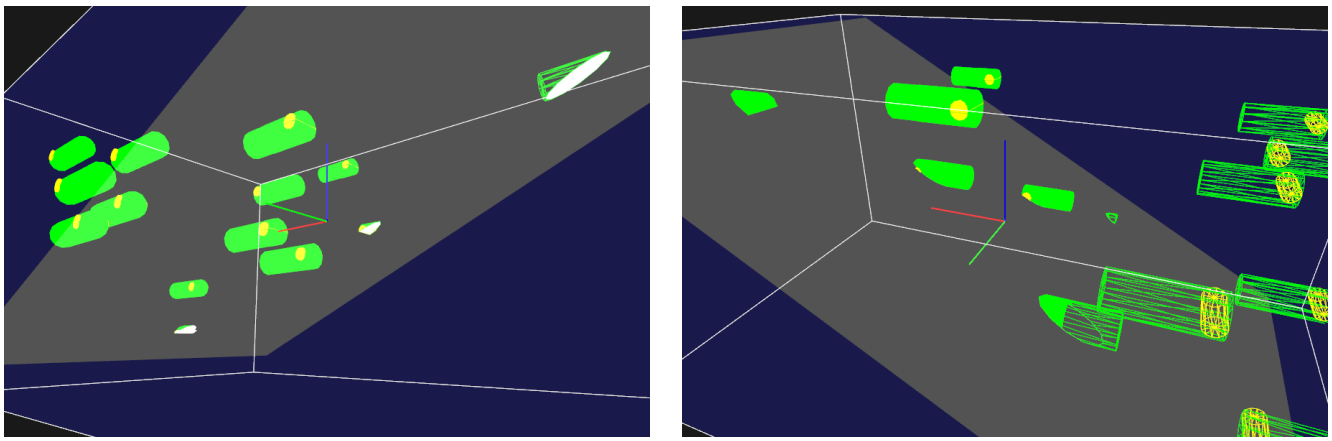

FIGURE 8 – Poissons coupés par le plan de dragage incliné

— activer le stencil ;
  — désactiver l'écriture dans le tampon de couleur (modifier le masque d'écriture) ;
  — désactiver le test de profondeur ;
  — activer le culling ;
  — indiquer que le test de stencil passe toujours ;
— afficher seulement les faces arrière de la scène en incrémentant les valeurs dans le stencil
— afficher seulement les faces avant de la scène en décrémentant les valeurs dans le stencil
⇒ à ce point, le tampon de stencil contient des 0 aux endroits qui ont vu ~~un nombre pair de faces~~ autant de faces avant que de faces arrière.
  — réactiver l'écriture dans le tampon de couleur (modifier le masque d'écriture) ;
  — réactiver le test de profondeur ;
  — désactiver le culling ;
— tracer le quadrilatère fermant les solides seulement aux endroits où les bits ne sont pas à 0 ;
— désactiver le stencil qui n'est plus nécessaire

FIGURE 9 – Une possible stratégie pour remplir le tampon de stencil

Partie 2 : mode sélection, nuanceur de géométrie

« Ô temps! suspends ton vol, ... » (Alphonse de Lamartine, 1790-1869)

L'utilisation du mode de sélection permettra de sélectionner un poisson et de suspendre son déplacement. À cette fin, vous utiliserez un booléen « `estSelectionne` » dans la classe `Poisson` qui permettra de savoir si le poisson est sélectionné. Les poissons seront en arrêt et leur position ne sera pas modifiée dans la méthode `avancerPhysique()`.

Bien sûr, pour appliquer ainsi le mode de sélection par couleurs, vous devrez définir et utiliser une couleur alternative pour identifier le poisson tracé. À vous de choisir votre patron de couleurs pour identifier uniquement les 25 poissons.

Note : En mode sélection, vous devrez aussi éviter de faire le `swap()` dans le `main()` tel que montré dans les exemples du cours. De plus, en mode sélection, il est préférable de ne pas afficher le quadrilatère représentant le plan de dragage puisqu'on ne désire pas qu'il puisse être sélectionné ! Enfin, pour tester et « voir » ce qui est affiché en mode sélection, vous pouvez prétendre toujours être en mode sélection (laisser `modeSelection` à `true` et laisser le `swap()` actif dans le `main()`.)

D'autre part, dans le but de mettre en pratique l'utilisation d'un nuanceur de géométrie, vous ajouterez un nuanceur de géométrie dans le pipeline GLSL utilisé. Ce nuanceur permettra d'utiliser deux clôtures différentes pour voir simultanément deux vues sur la scène (voir figure 10).

Dans la vue du haut, le plan de dragage éliminera tout ce qui est *au-dessous* du plan de dragage, comme vous avez fait jusqu'à maintenant. À l'inverse, dans la vue du bas, le plan de dragage éliminera plutôt tout ce qui est *au-dessus* du plan de dragage. (*Indice* : comment se comparent les valeurs de `gl_ClipDistance[0]` dans les vues du haut et du bas ?)

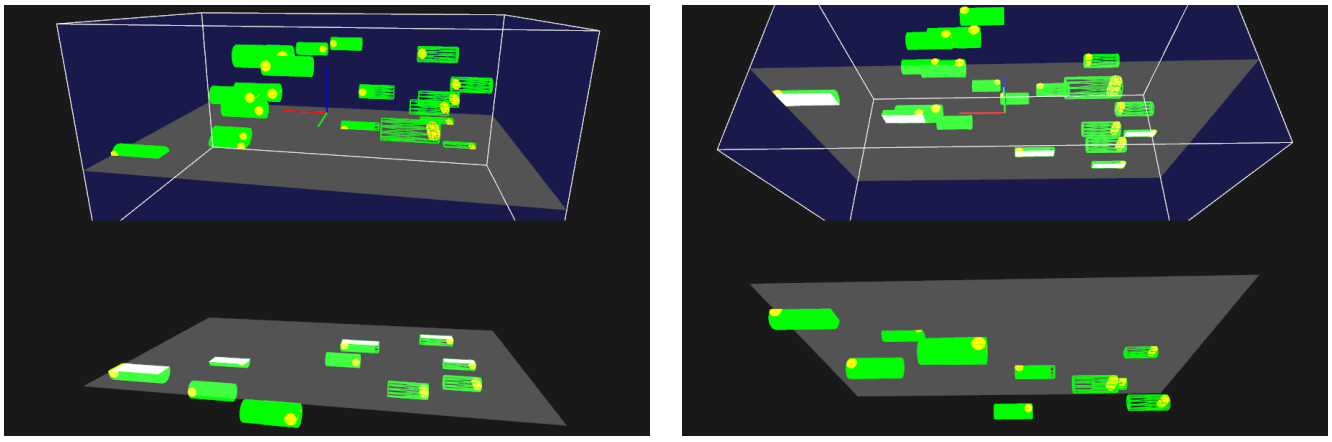

FIGURE 10 – Utilisation de deux clôtures avec le plan de coupe à différentes positions

Enfin, pour bien faire les choses, il vaut mieux alors attendre dans le nuanceur de géométrie avant d'affecter une valeur à `gl_ClipDistance[]`. On peut déplacer le calcul de la valeur du nuanceur de sommets au nuanceur de géométrie ou simplement passer dans `AttribsIn` le résultat à affecter.

Partie 3 : modification aux couleurs en GLSL

Pour ajouter un bel effet à l'ensemble du dessin, on atténuera les valeurs RGB de la couleur selon la distance à la caméra comme illustrée à la figure 11. Cette distance pourrait être calculée dans le nuanceur de sommets par : « `dist = -( matrVisu * matrModel * Vertex ).z;` », mais on peut aussi l'obtenir dans le nuanceur de fragments par : « `dist = gl_FragCoord.z / gl_FragCoord.w;` ».

Dans ce TP, on appliquera l'atténuation de la couleur dans l'intervalle [30,50] : aucune atténuation (= couleur originale) lorsque `d` < 30, atténuation variable entre 30 et 50 et atténuation complète (= noir) lorsque `d` > 50. Utilisez la fonction GLSL prédéfinie « `smoothstep` » qui calcule une transition lisse (par une interpolation d'Hermite) entre les valeurs données pour produire une valeur entre 0 et 1 :

`http://www.khronos.org/registry/OpenGL-Refpages/gl4/html/smoothstep.xhtml`

```
float smoothstep( float x1, float x2, float x )
{ float t = clamp((x-x1)/(x2-x1), 0.0, 1.0 ); return t*t*(3.0-2.0*t); }
```

Le cylindre de base qui sert à afficher le corps des poissons est, avant les transformations de modélisation, de longueur 1 et orienté dans l'axe des Z. Il sera donc assez facile de modifier la couleur des poissons afin qu'elle varie du vert au bleu entre la « tête » et la « queue ». Utilisez la fonction GLSL « `mix` » pour interpoler la nouvelle couleur de chaque sommet.

`http://www.khronos.org/registry/OpenGL-Refpages/gl4/html/mix.xhtml`

```
vec4 mix( vec4 c1, vec4 c2, float fact ) { return c1*(1-fact) + c2*fact); }
```

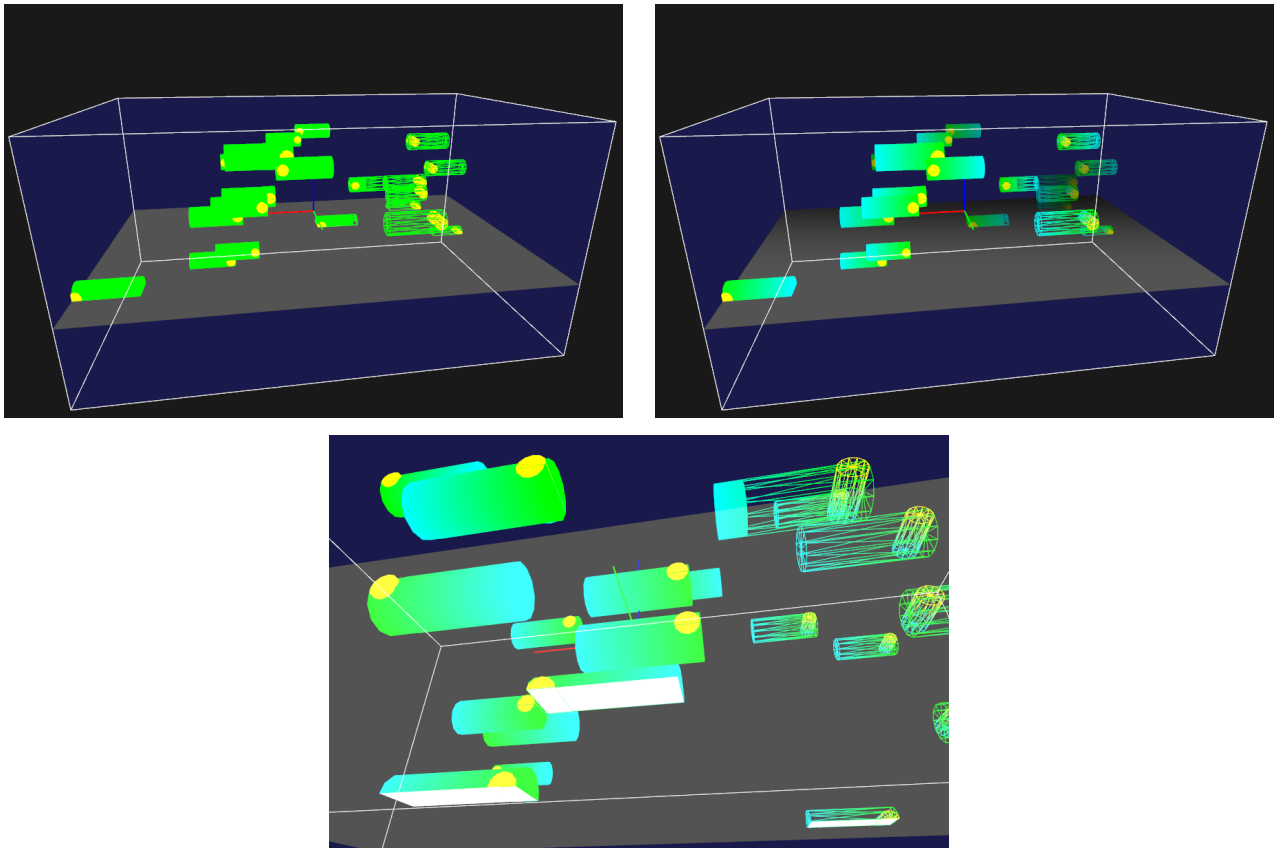

FIGURE 11 – Affichage sans atténuation et avec atténuation en fonction de la profondeur

3 Exigences

3.1 Exigences fonctionnelles

Partie 1 :

- E1. Un premier plan de coupe vertical sépare la région montre le squelette des poissons (en fil de fer pour un effet rayons X).
- E2. Un second plan de coupe de dragage coupe les poissons par le fond.
- E3. Le quadrilatère représentant le plan de dragage est créé avec deux VBO et affiché en transparence.
- E4. La taille du plan est correcte lorsque le plan de coupe de dragage est déplacé ou incliné.
- E5. Les poissons coupés sont « fermés » en utilisant le tampon stencil pour dessiner un quadrilatère dont l'affichage est limité aux pixels de la coupe.

Partie 2 :

- E6. Un nuanceur de géométrie est utilisé pour afficher deux vues découpées différemment par le même plan de dragage.
- E7. Un poisson sélectionné ne bouge plus. (On sélectionne les poissons dans la vue du haut seulement.)

Partie 3 :

- E8. La couleur peut être atténuée selon la profondeur.
- E9. La couleur du poisson varie linéairement entre sa tête (verte) et sa queue (cyan=vert+bleu). La couleur est modifiée dans le nuanceur.
- E10. (Le logiciel utilise correctement les touches listées à l'annexe A pour faire varier les divers paramètres.)

3.2 Rapport

Vous devez répondre aux questions dans le fichier `Rapport.txt` qui sera inclus dans la remise. Vos réponses doivent être complètes et suffisamment détaillées. (Quelqu'un pourrait suivre les instructions que vous avez écrites sans avoir à ajouter quoi que ce soit.)

ANNEXES

A Liste des commandes

Touche	Description
`q`	Quitter l'application
`x`	Activer/désactiver l'affichage des axes
`v`	Recharger les fichiers des nuanceurs et recréer le programme
`ESPACE`	Mettre en pause ou reprendre l'animation
`c`	Déplacer le plan de rayonsX vers la droite (-X)
`z`	Déplacer le plan de rayonsX vers la gauche (+X)
`HAUT`	Déplacer le plan de dragage vers le haut
`BAS`	Déplacer le plan de dragage vers le bas
`DROITE`	Augmenter l'angle du plan de dragage
`GAUCHE`	Diminuer l'angle du plan de dragage
`PLUS`	Incrémenter la distance de la caméra
`MOINS`	Décrémenter la distance de la caméra
`a`	Atténuer ou non la couleur selon l'éloignement
`BOUTON GAUCHE`	Modifier le point de vue
`BOUTON DROIT`	Sélectionner des objets
`Molette`	Déplacer le plan de dragage

B Figures supplémentaires

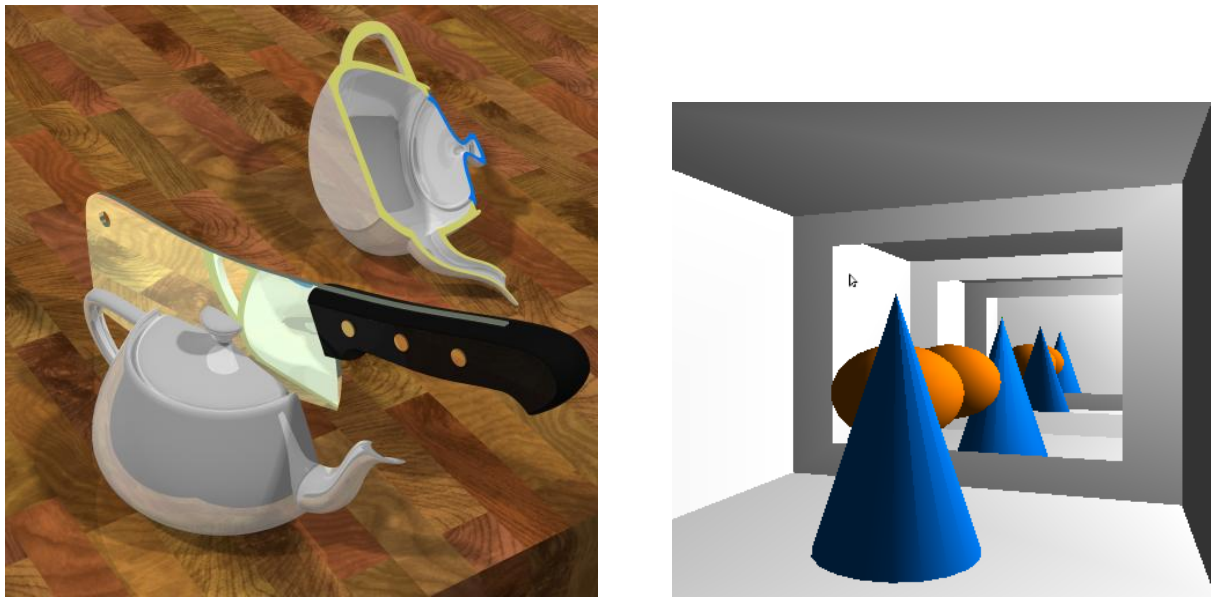

FIGURE 12 – Demi-théière avec réflexion sur la lame du couteau et réflexions multiples

C Apprentissage supplémentaire

Partie 1 :

1. Utiliser d'autres touches du clavier pour changer la vitesse de déplacement des poissons ou contrôler l'écoulement du temps.
2. Utiliser d'autres plans de coupe.
3. Utiliser des objets non complètement fermés plutôt que des cylindres, tel que peut-être des théières ! (Ça causera probablement quelques problèmes à l'affichage.)
4. *[Un peu plus compliqué]*. Ajouter un miroir à la scène. Notez la réflexion de la demi-théière sur la lame du couteau à la figure 12.

Partie 2 :

5. Utiliser quatre clôtures différentes pour montrer différents points de vue sur la scène. (Il vaut mieux alors appliquer les matrices de projection et de visualisation dans le nuanceur de géométrie.)
6. Faites varier la trajectoire des poissons selon une ellipse. Changer l'orientation du plan de chaque trajectoire.
7. Afficher le point de vue à partir d'un poisson sélectionné (à l'arrêt).
8. Afficher le point de vue à partir d'un poisson non sélectionné (en mouvement).

Partie 3 :

9. Faites varier la couleur de l'eau en fonction de la hauteur dans l'aquarium.

D Fonctions pertinentes

D.1 Utilisation d'un plan de coupe

La valeur w des variables `planRayonsX` et `planDragage` est la position du plan en X ou en Z, c'est-à-dire X = w ou Z = w. On inversera donc le signe de w dans l'équation du plan de coupe à transmettre au nuanceur afin d'avoir Ax + By + Cz - w = 0. Le signe du produit scalaire `dot(plan,pos)`, évalué dans le nuanceur, permettra alors de savoir si le point est d'un côté ou de l'autre du plan.

La fonction qui l'activation d'un plan de coupe est :

```
void glEnable( GL_CLIP_PLANEi ); // i=0,1,2,3,...
ex.: glEnable( GL_CLIP_PLANE0 );
     glEnable( GL_CLIP_PLANE1 );
```

et, dans les nuanceurs :

```
...
gl_ClipDistance[0] = dot( planDragage, pos );
gl_ClipDistance[1] = dot( planRayonsX, pos );
```

D.2 Utilisation du stencil

Les fonctions qui contrôlent le comportement du tampon de stencil sont :

```
void glStencilFunc( GLenum func, GLint ref, GLuint mask );
void glStencilOp( GLenum sfail, GLenum zfail, GLenum pass );
```

D.3 Utilisation des masques d'écriture

La fonction qui contrôle si les valeurs de R,G,B,A du fragment courant peuvent être écrites ou non dans le tampon de couleur est :

```
void glColorMask( GLboolean red, GLboolean green, GLboolean blue,
                  GLboolean alpha );

ex.: glColorMask( GL_TRUE, GL_TRUE, GL_TRUE, GL_TRUE );
     glColorMask( GL_FALSE, GL_FALSE, GL_FALSE, GL_FALSE );
```

La fonction qui contrôle si les valeurs de Z du fragment courant peuvent être écrites ou non dans le tampon de profondeur est :

```
void glDepthMask( GLboolean flag );

ex.: glDepthMask( GL_TRUE );
     glDepthMask( GL_FALSE );
```