

INF2705 Infographie

Spécification des requis du système

Travail pratique 2

Miroirs, mode sélection et nuanceur de géométrie

Table des matières

1	Introduction	2
1.1	But	2
1.2	Portée	2
1.3	Remise	2
2	Description globale	3
2.1	But	3
2.2	Contexte	3
2.3	Travail demandé	4
3	Exigences	8
3.1	Exigences fonctionnelles	8
3.2	Exigences non fonctionnelles	8
3.3	Rapport	8
A	Liste des commandes	9
B	Figures supplémentaires	9
C	Apprentissage supplémentaire	10
D	Fonctions pertinentes	11
D.1	Utilisation d'un plan de coupe	11
D.2	Utilisation du stencil	11
D.3	Utilisation des masques d'écriture	11

1 Introduction

Ce document décrit les exigences du TP2 « *Miroirs, mode sélection et nuanceur de géométrie* » 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()` et `glStencilOp()` par l'utilisation de miroirs. 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 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 la peau et le squelette du poisson en une seule étape, on tranche sa chair maintenant sans arêtes et on le met facilement en conserve !

De plus, afin de bien visualiser la santé de ces poissons dans l'aquarium, une région de l'aquarium permet de voir les poissons sous rayons X (en fil de fer).

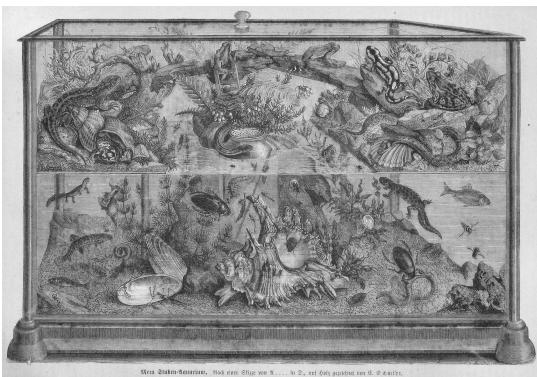


FIGURE 1 – Aquarium

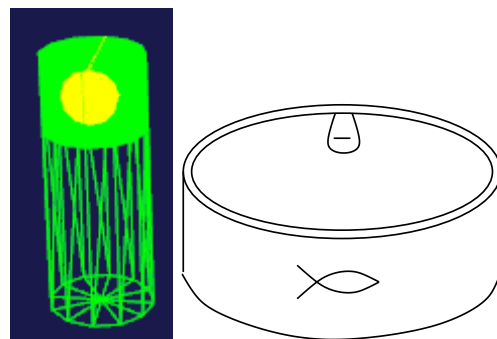


FIGURE 2 – Mise en conserve facile d'un poisson cylindrique

Cette histoire abracadabrante serait évidemment une gigantesque erreur humaine ... 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 Bobby Lapointe. :)
<https://www.youtube.com/watch?v=0EldDTBvO3o>

L'affichage de base

Le programme de base fourni affiche un aquarium contenant des poissons cylindriques où les poissons tournent en rond (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).

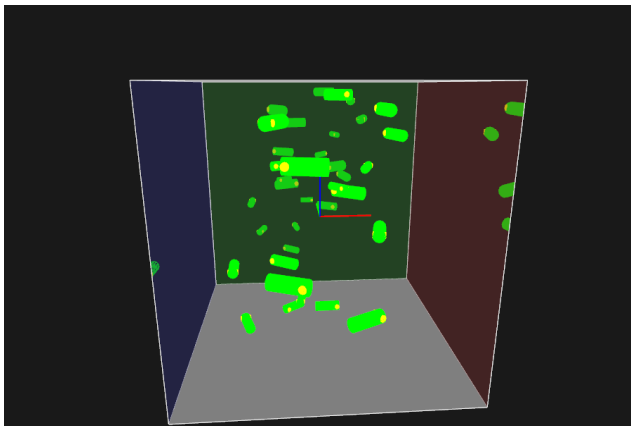


FIGURE 3 – L'aquarium et ses poissons

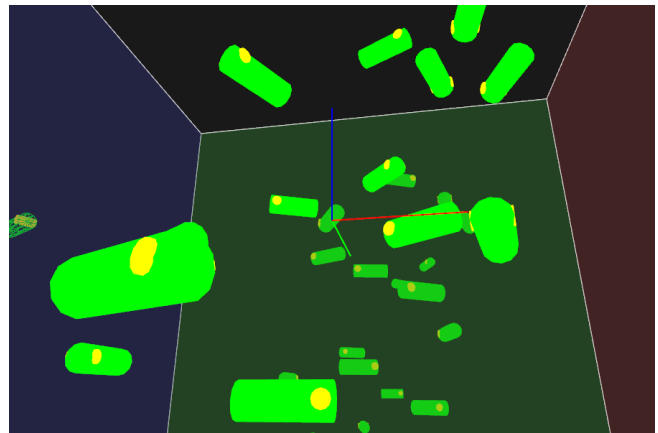


FIGURE 4 – Les yeux des poissons

À terme, les poissons passeront sous les rayons X, on verra le contenu de l'aquarium sur les quatre parois et la couleur des poissons variera du vert au bleu entre la « tête » et la « queue » (Figure 5).

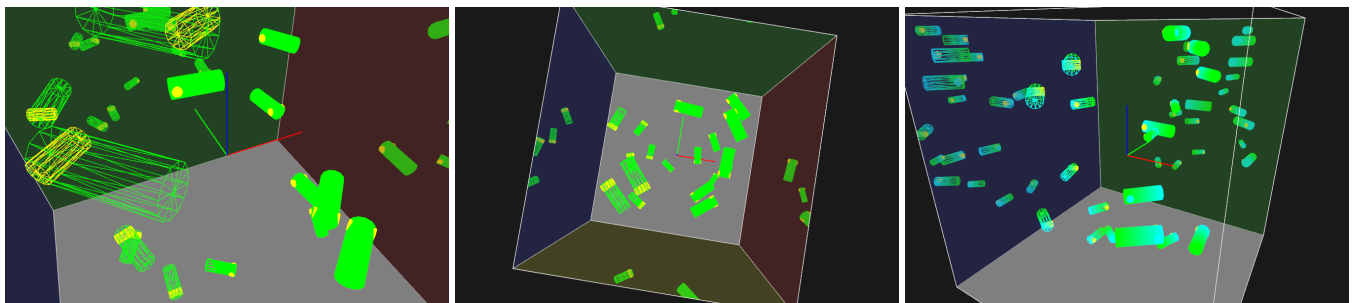


FIGURE 5 – L'aquarium et ses poissons en version finale avec ses 4 miroirs

2.3 Travail demandé

Partie 1 : tampon de stencil et utilisation d'un plan de coupe

On demande de faire le rendu de quatre miroirs sur les parois de l'aquarium. Il faudra donc tracer la scène cinq fois : une fois pour le contenu et quatre autres fois pour le rendu inversé de l'autre côté de chaque miroir (figure 6, à gauche). En activant le mode stencil et en faisant en sorte que chaque paroi affichée ait des valeurs différentes, il est possible de n'afficher que le contenu réfléchi dans le miroir correspondant (figure 6, à droite).

Notez que les parois ne se superposent jamais puisqu'on voit toujours seulement que leur face avant (celle l'intérieur de l'aquarium). Vous pouvez d'abord afficher chaque paroi afin de mettre des valeurs différentes dans le stencil et, une fois le stencil rempli des valeurs adéquates, vous pouvez ensuite afficher le contenu de l'aquarium à quatre reprises en utilisant le tampon de stencil. (N'ayez pas la tentation de faire un appel à `glClear()` entre l'affichage de chaque miroir !)

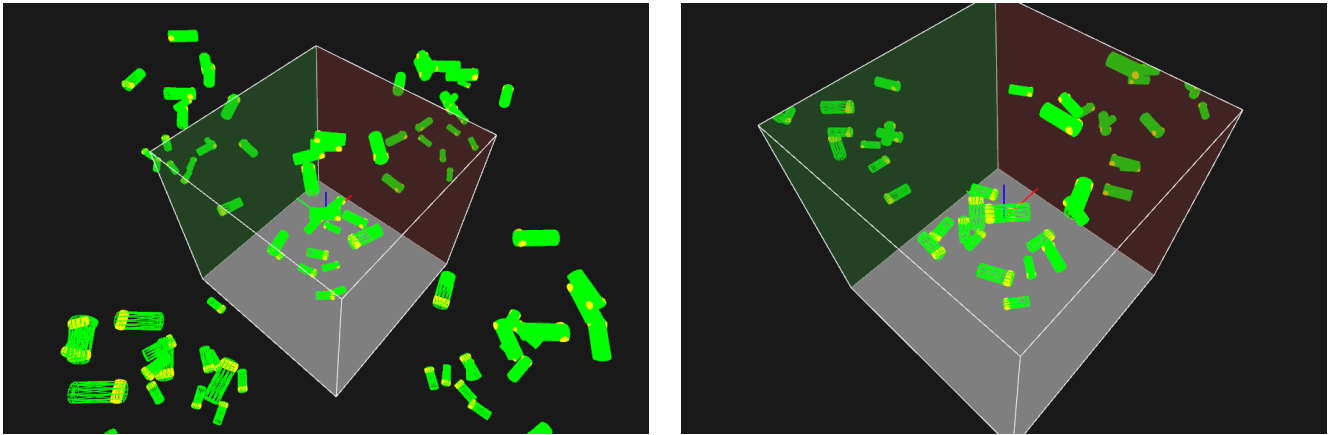


FIGURE 6 – Affichage des contenus inversés (à gauche) et des miroirs (à droite).

Pour le passage aux rayons X, un plan de coupe vertical, situé à une distance variable, $X = d$, sera utilisé pour délimiter deux régions dans l'aquarium : affichage normal du poisson, en plein, lorsque « $x > d$ » ou affichage du squelette du poisson, en fil de fer, sinon. (Voir figure 7 ci-dessous.)

Indice : la scène est affichée une première fois avec les poissons en plein et une seconde fois, en changeant l'orientation du plan de coupe, avec les poissons sous rayons X, en fil de fer.

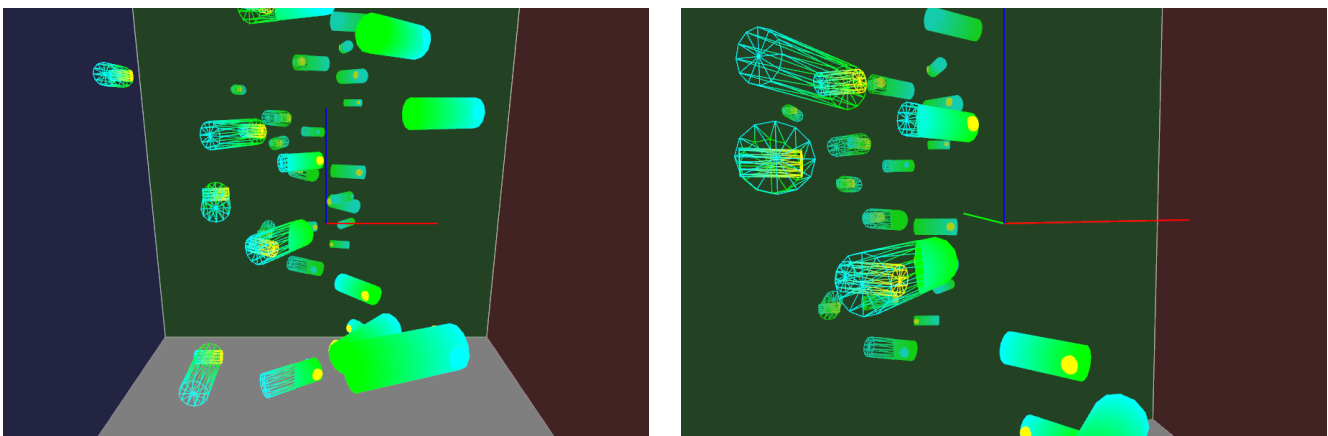


FIGURE 7 – Poissons normaux et sous rayons X

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 `Poisson::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 chacun des poissons.

Note : En mode sélection, vous devrez aussi éviter de faire le `swap()` dans le `main()` tel que montré dans les exemples du cours. 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 8).

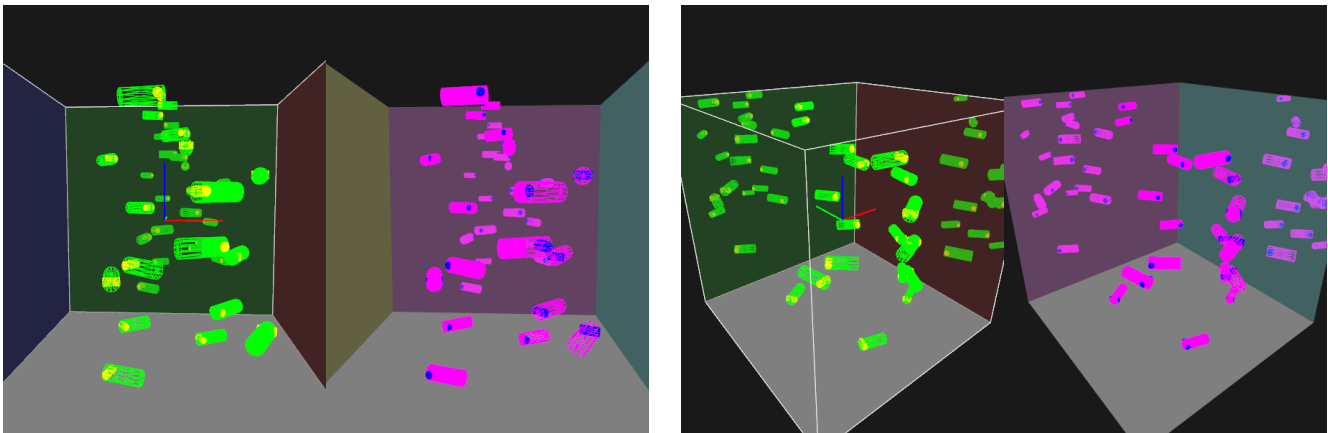


FIGURE 8 – Utilisation de deux clôtures avec affichage en négatif

La clôture de gauche montrera le contenu de façon habituelle, tandis que la clôture de droite montrera le contenu avec un affichage en négatif (*reverse video*).

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

Avec autant de poissons, l'eau de l'aquarium peut facilement devenir sale et moins transparente. Pour donner un tel effet « réaliste » à cet aquarium, on atténuera les valeurs RGB de la couleur selon la hauteur en Z en les mélangeant avec une couleur « sale » d'atténuation comme illustrée à la figure 9. Dans ce TP, on appliquera une atténuation de la couleur en Z dans l'intervalle $[-12, +5]$: aucune atténuation (= couleur originale) lorsque $z < +5$, atténuation variable entre -12 et +5 et atténuation maximale (couleur = couleurAtténuation) lorsque $z > -12$. 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. Vous pourrez ensuite utiliser cette valeur comme argument pour le mélange avec la fonction GLSL « mix » pour interpoler la nouvelle couleur des sommets lorsqu'une atténuation est activée.

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 est donc 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/smoothstep.xhtml>

<http://www.khronos.org/registry/OpenGL-Refpages/gl4/html/mix.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); }
vec4 mix( vec4 c1, vec4 c2, float fact ) { return c1*(1-fact) + c2*fact); }
```

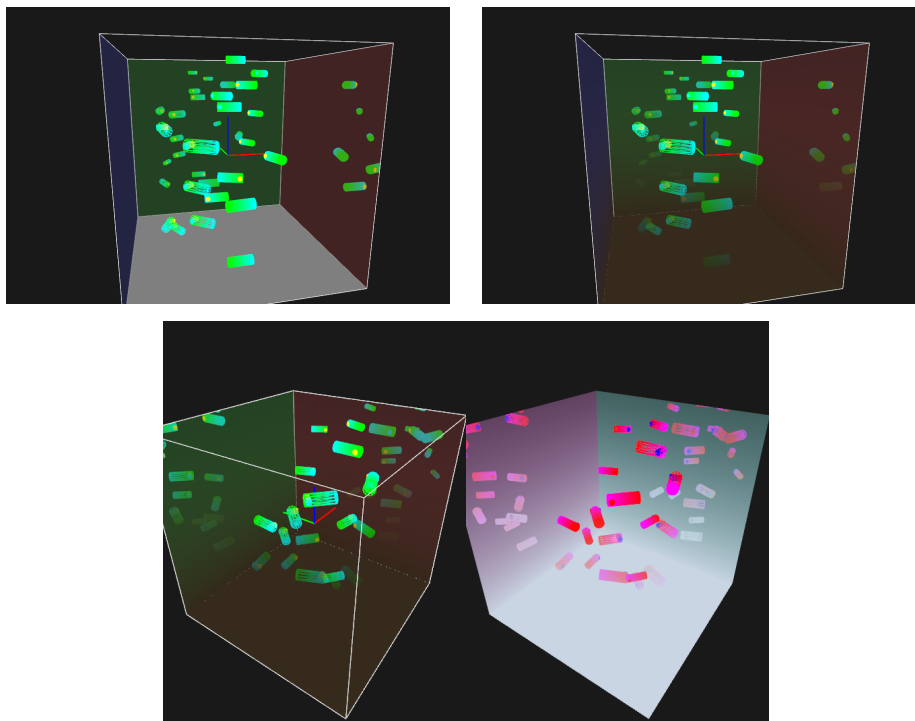


FIGURE 9 – Affichage, avec la couleur des poissons variant du bleu au vert, sans atténuation et avec atténuation en fonction de la profondeur de l'aquarium.

3 Exigences

3.1 Exigences fonctionnelles

Partie 1 :

- E1. Les quatre parois agiront comme des miroirs qui montrent le contenu de l'aquarium en réflexion sur chacune. Un seul appel à `glClear()` est utilisé au début de l'affichage. [6 points]
- E2. Lorsque les parois de l'aquarium sont déplacées afin de faire varier la taille de l'aquarium, les miroirs continuent d'être bien positionnés sur les parois. [2 points]
- E3. Un plan de coupe vertical est utilisé pour séparer la région où on montre le squelette des poissons (en fil de fer pour un effet rayons X). [4 points]

Partie 2 :

- E4. Un poisson sélectionné ne bouge plus. (On sélectionne les poissons dans la vue de gauche seulement.) [5 points]
- E5. Un nuanceur de géométrie est utilisé pour afficher deux vues sur le contenu de l'aquarium : habituelle et en couleurs négatives. [4 points]

Partie 3 :

- E6. La couleur peut être atténuée selon la profondeur. [4 points]
- E7. La couleur du poisson varie linéairement entre sa tête (verte) et sa queue (cyan=vert+bleu). La couleur est modifiée dans le nuanceur. [3 points]
- E8. (Le logiciel utilise correctement les touches listées à l'annexe A pour faire varier les divers paramètres.)

3.2 Exigences non fonctionnelles

De façon générale, le code que vous ajouterez sera de bonne qualité. Évitez les énoncés superflus (qui montrent que vous ne comprenez pas bien ce que vous faites!), les commentaires erronés ou simplement absents, les mauvaises indentations, etc. [2 points]

3.3 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.) [2 points]

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)
DROITE	Augmenter la dimension de la boîte en X
GAUCHE	Diminuer la dimension de la boîte en X
HAUT	Augmenter la dimension de la boîte en Y
BAS	Diminuer la dimension de la boîte en Y
PLUS	Incrémenter la distance de la caméra
MOINS	Décrémenter la distance de la caméra
a	Atténuer ou non la couleur selon la profondeur
BOUTON GAUCHE	Modifier le point de vue
BOUTON DROIT	Sélectionner des objets
Molette	Changer la distance de la caméra

B Figures supplémentaires

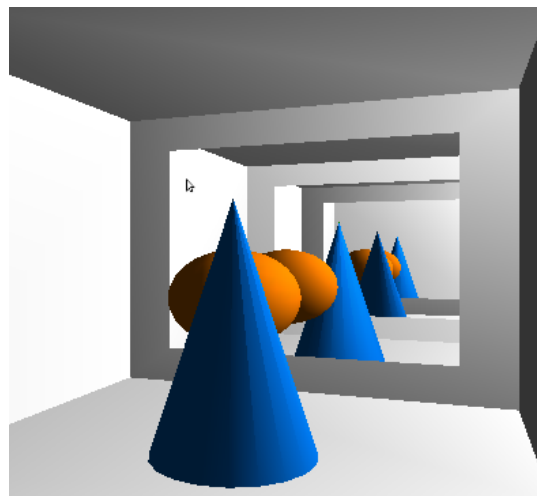


FIGURE 10 – Demi-théière avec réflexion sur la lame du couteau et réflexions multiples

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 d'autres objets pour représenter les poissons, par exemple des théières.
4. *[Un peu plus compliqué]*. Sur chaque miroir, ajouter les réflexions des autres miroirs dans la scène (double réflexion). Notez les réflexions multiples à la figure 10.

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 distance à la caméra.

D Fonctions pertinentes

D.1 Utilisation d'un plan de coupe

La valeur w de la variable `planRayonsX` est la position du plan, soit $X = w$. 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 active 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( 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 );
```