

INF2705 Infographie

Spécification des requis du système

Travail pratique 2

Miroirs, caméras et nuanceurs avec les théières

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 : l'aquarium relaxant	3
2.3	Travail demandé	4
3	Exigences	9
3.1	Exigences fonctionnelles	9
3.2	Exigences non fonctionnelles	9
A	Liste des commandes	10
B	Figures supplémentaires	11
C	Apprentissage supplémentaire	12

1 Introduction

Ce document décrit les exigences du TP2 « *Miroirs, caméras et nuances avec les théières* » (Automne 2021) 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` » ou exécutez/cliquez sur « `remise.bat` » afin de créer l'archive « **INF2705_remise_TPn.zip** » (ou .7z, .rar, .tar) 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 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 en utilisant des miroirs. Il permettra également de mettre en pratique la caméra synthétique, le nuanceur de géométrie et l'illumination.

2.2 Contexte : l'aquarium relaxant

Pour relaxer et évacuer le stress, regarder un poisson rouge dans un petit aquarium simple peut suffire, mais d'autres ont besoin d'un aquarium plus ambitieux, à pièces multiples, avec pour agrémenter la vie de leurs multiples poissons (figures 1 et 2) Dans le même thème, à découvrir ou à réécouter : « *La Maman des Poissons* » de Bobby Lapointe à [//www.youtube.com/watch?v=0EldDTBvO3o](https://www.youtube.com/watch?v=0EldDTBvO3o). :)



FIGURE 1 – Aquarium simple



FIGURE 2 – Aquarium à pièces multiples

Dans ce cours d'infographie, on remplacera les poissons rouges par de magnifiques théières. Nous aurons ainsi bel aquarium rectangulaire, avec des parois réfléchissantes, contenant plusieurs théières colorées qui se déplacent lentement en regardant droit devant (figures 3 et 4).

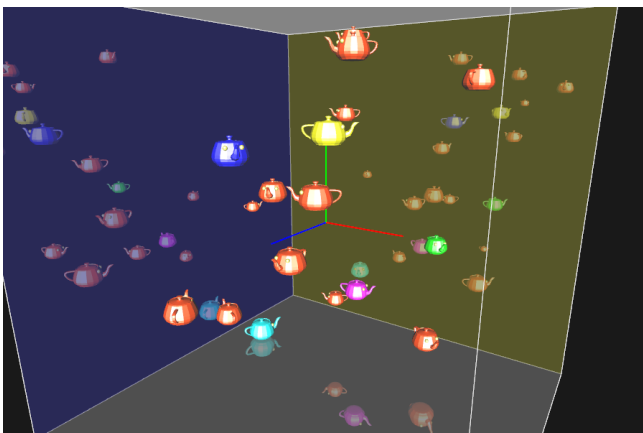


FIGURE 3 – L'aquarium et ses théières

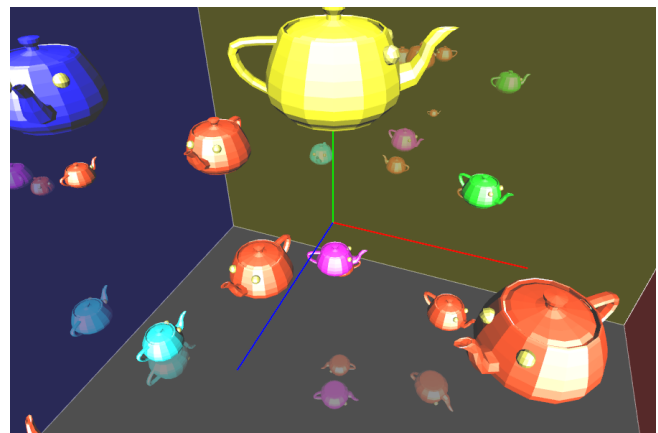


FIGURE 4 – Les yeux des théières

2.3 Travail demandé

Partie 1 : tampon de stencil

Les parois intérieures de notre aquarium sont des miroirs qui reflètent nos belles théières (figure 5). Pour avoir ce rendu de cinq miroirs sur les quatre parois et le fond de l'aquarium, il faut donc tracer la scène six fois : une fois pour le contenu et cinq autres fois pour le rendu inversé de l'autre côté de chaque miroir (figure 6, à gauche). En activant le 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). Il faudra peut-être « réfléchir » un peu à la stratégie à utiliser. :)

Notez que les parois ne se superposent jamais puisqu'on voit toujours seulement que leur face avant (celle l'intérieur de l'aquarium). Pour remplir le stencil, vous pouvez afficher chaque paroi en mettant des valeurs différentes pour chacune (ex. : 1,2,4,8,16) et, une fois le stencil rempli des valeurs adéquates, vous pouvez ensuite afficher le contenu de l'aquarium à cinq reprises en utilisant le tampon de stencil. (N'ayez pas la tentation de faire un appel à `glClearColor()` entre l'affichage de chaque miroir.)

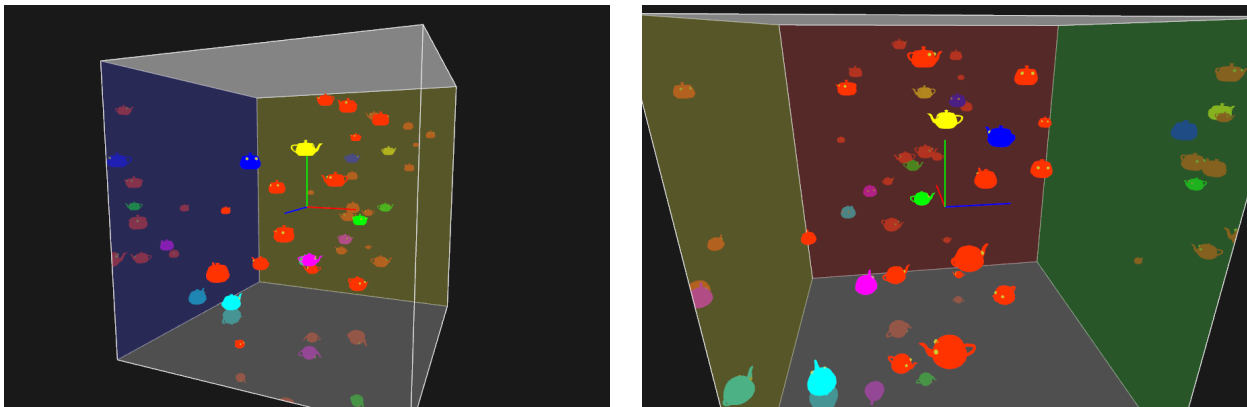


FIGURE 5 – Les miroirs de l'aquarium

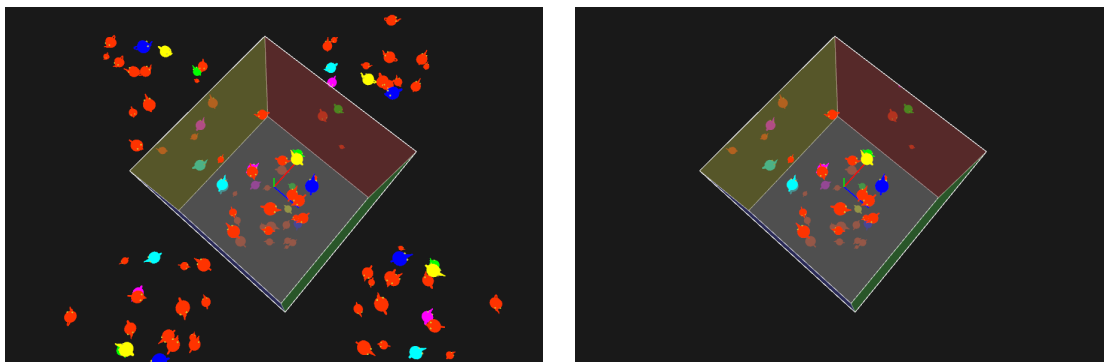


FIGURE 6 – Affichage du contenu de l'aquarium, inversé par rapport à chaque miroir (à gauche), puis en activant le stencil des miroirs pour n'afficher que dans les miroirs (à droite).

Pour s'inspirer, vous pouvez vous inspirer des exemples vus au cours :

[https://gitlab.com/ozell/inf2705-exemples/-/tree/master/04 StencilMiroir/](https://gitlab.com/ozell/inf2705-exemples/-/tree/master/04%20StencilMiroir/) et
<https://gitlab.com/ozell/inf2705-exemples/-/tree/master/04-StencilTheiere/>.

Partie 2 : point de vue à partir d'une théière choisie

Positionnement de la caméra. Pour se pratiquer à bien positionner la caméra, on affichera le point de vue selon une théière en mouvement dans l'aquarium, c'est-à-dire en positionnant la caméra sur le dessus d'une théière. Ainsi, selon le choix d'une des touches 1 à 9, on affichera le point de vue d'une des neuf premières théières de la liste, tandis que la touche 0 affichera le point de vue standard externe à l'aquarium (figure 7). À l'écran, on remarque que les cinq premières théières de la liste sont faciles à identifier par leur couleur particulière (cyan, magenta, jaune, bleu, vert) tandis que les suivantes sont simplement d'une même couleur uniforme (rouge orangé).

La stratégie pour positionner la caméra à une théière est de déplacer le repère de la façon habituelle jusqu'à la position voulue, de récupérer la matrice de modélisation courante, de l'inverser et de s'en servir comme matrice de visualisation. (C'est un simple changement de repère.)

La méthode `Theiere::obtenirMatriceCourante()` devra retourner la matrice de modélisation courante `mtc` représentant la position de la théière choisie, c'est-à-dire les déplacements pour amener le repère à la position voulue¹. La méthode `Aquarium::positionnerCamera()` devra calculer l'inverse de cette matrice de modélisation courante et s'en servir ensuite pour initialiser la matrice de visualisation : `matrVisu.setMatr(glm::inverse(glm::mat4(mtc)));`.

Notez bien que le déterminant de la matrice de visualisation doit toujours être égal à 1 afin de ne pas introduire de changement d'échelle.²

Pour bien amener le repère en haut de la théière (plutôt que le laisser à sa base) et retourner une matrice de modélisation courante appropriée, il faut inclure une translation correspondant à la hauteur de la théière choisie. On sait que la théière de base a une hauteur en Y d'environ 5.0 unités, mais aussi que chaque théière affichée a une taille variable qui est modulée selon son facteur `taille`. La translation nécessaire fera donc intervenir ces deux valeurs.³

Enfin, pour que chaque théière regarde droit devant elle (vers « l'avant »), c'est-à-dire vers le bec de la théière plutôt que vers son anse, il faudra aussi tenir compte que certaines théières tournent dans un sens et d'autres dans l'autre sens, selon si leur `vitesse` est positive ou négative. Le plus simple est donc d'inclure une rotation de 180 degrés dans la matrice de modélisation avant de l'inverser.

Nuanceur de géométrie. Cette semaine, prenez aussi le temps d'introduire dans votre pipeline graphique, un nuanceur de géométrie qui ne fait que recopier les valeurs reçues du nuanceur de sommets vers le nuanceur de fragments. Ce nouveau nuanceur vous servira dans la suite de ce TP la semaine prochaine.

1. Pourquoi ne peut-on simplement pas sauvegarder et utiliser une copie de la matrice de modélisation produite lors de l'affichage de cette théière ? À réfléchir...

2. Que se passe-t-il si le déterminant de la matrice de visualisation n'est pas unitaire ?

3. Pourquoi ne pas simplement faire une mise à l'échelle de `taille` et une translation fixe de 5 unités ? À réfléchir...

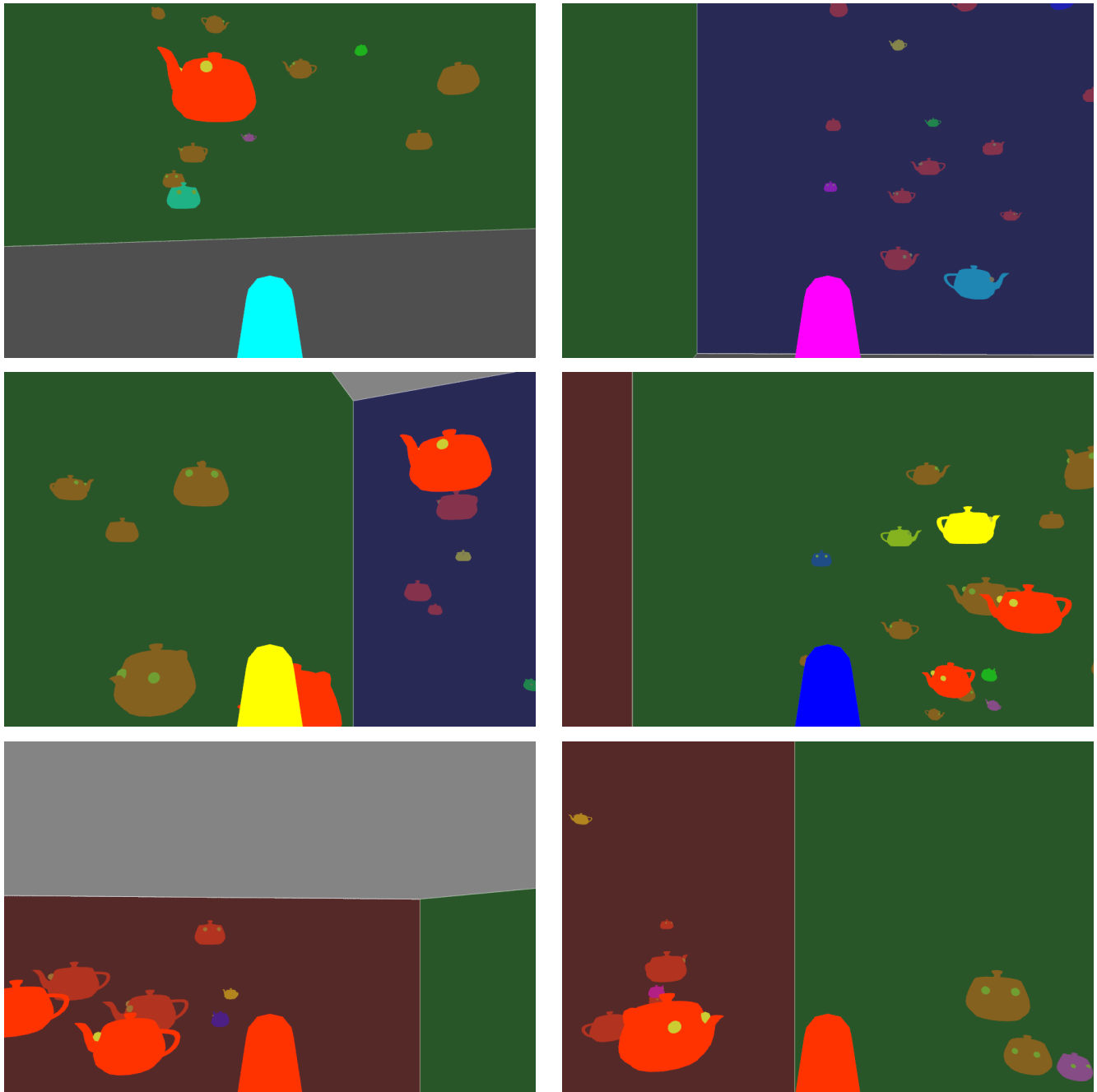


FIGURE 7 – Points de vue selon diverses théières

Partie 3 : illumination et nuanceur de géométrie

Enfin, afin de produire un rendu intéressant (pensez « petits poissons rouges »), vous ajouterez à la scène l'illumination de Phong dans le nuanceur de fragments. Le nuanceur de géométrie vous permettra de calculer la normale à chaque face en faisant le produit vectoriel de deux arêtes du triangle.

Ainsi, dans le nuanceur de géométrie que vous ajouterez, vous recevrez les trois sommets de chaque triangle dans `gl_in[i].gl_Position`, $i=0,1,2$. Il est donc facile de calculer le produit vectoriel entre deux arêtes du triangle pour obtenir un vecteur perpendiculaire à ces deux vecteurs (et à la surface du triangle), c'est-à-dire la normale (N) recherchée :

```
vec3 arete1 = ( Sommet1 - Sommet0 );
vec3 arete2 = ( Sommet2 - Sommet0 );
normale = cross( arete1, arete2 );
```

De plus, la source de lumière sera positionnée au loin dans la direction de la caméra ce qui nous permet de simplifier et d'utiliser un vecteur constant :

```
lumiDir = vec3( 0, 0, 1 ).
```

Enfin, on considérera que l'observateur est aussi toujours placé au plan avant et nous utiliserons :

```
obsVec = vec3( 0, 0, 1 ).
```

Ces simplifications pour la position de la lumière et de l'observateur sont souvent utilisées pour que les calculs d'illumination soient plus simples. Dans ce TP, nous utiliserons le nuanceur de géométrie pour imposer ces valeurs à chaque face, mais nous utiliserons quand même l'illumination complète de Phong dans le nuanceur de fragments, sans y chercher à simplifier les calculs.

Pour s'inspirer, vous pouvez utiliser les exemples d'illumination vus au cours :

<https://gitlab.com/ozell/inf2705-exemples/-/tree/master/06-IlluminationMiroir/> et
<https://gitlab.com/ozell/inf2705-exemples/-/tree/master/06-IlluminationTheieres/>.

Vous pouvez aussi utiliser l'exemple du nuanceur de géométrie vu au cours :

<https://gitlab.com/ozell/inf2705-exemples/-/tree/master/05-NuanceurGeometrie/>.

Note : Ce TP utilise des « *Uniform Buffer Object* » (UBO) afin de transférer en bloc les variables uniformes aux nuanceurs. L'usage des UBO est très semblable aux VBO et permet surtout que le passage des valeurs des variables uniformes aux nuanceurs soit beaucoup plus efficace. Dans ce TP, on utilise ainsi trois UBO qui correspondent aux trois blocs de variables uniformes utilisés dans les nuanceurs : `LightSource`, `FrontMaterial` et `LightModel`. Les trois blocs contiennent les variables uniformes servant à l'illumination. (Les noms de variable sont inspirés de OpenGL 2.x.)

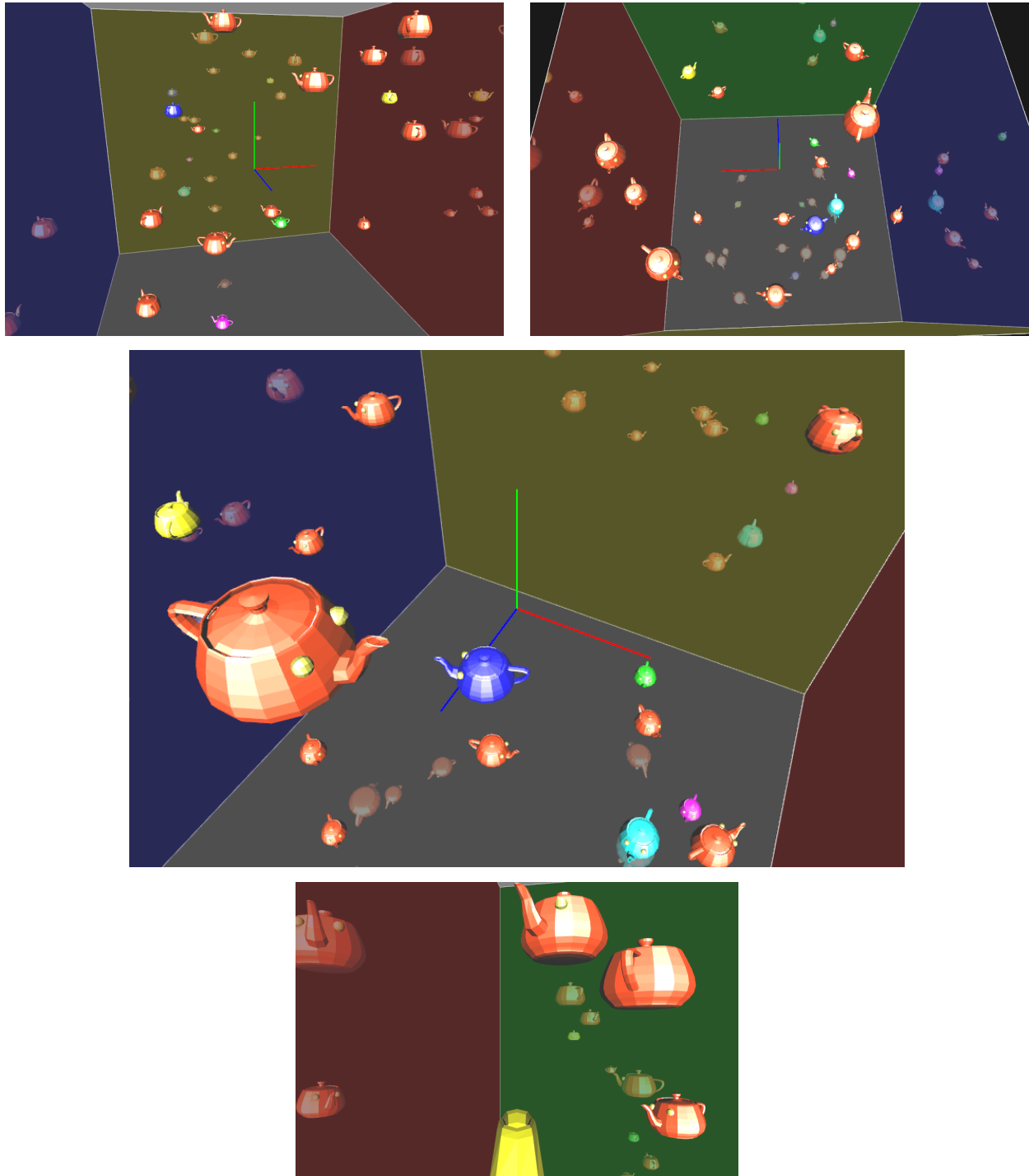


FIGURE 8 – Illumination avec des vecteurs normaux constants par face.

3 Exigences

3.1 Exigences fonctionnelles

Partie 1 :

- E1. Les parois et le fond agissent 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. [7 pts]
- E2. Lorsque les parois de l'aquarium sont déplacées (en utilisant les flèches sur le clavier) afin de faire varier la taille de l'aquarium, les miroirs continuent d'être bien positionnés sur les parois et la réflexion est correcte. [3 pts]

Partie 2 :

- E3. La caméra peut être positionnée correctement au-dessus des théières 1 à 9 et on voit les autres théières, de même que toutes les réflexions dans les miroirs. [5 pts]
- E4. La caméra est positionnée correctement au-dessus de théière, peu importe la taille de la théière. [1 pt]
- E5. La caméra est positionnée correctement afin de toujours voir vers l'avant, c'est-à-dire vers le bec de théière. [1 pt]

Partie 3 :

- E6. Un nuanceur de géométrie est utilisé pour calculer le vecteur normal de chaque triangle et pour donner des valeurs correctes aux variables servant à l'illumination. [5 pts]
- E7. Le nuanceur de fragments calcule correctement l'illumination de Phong en utilisant les valeurs obtenues du nuanceur de géométrie. [5 pts]
- E8. Le nuanceur de fragments consulte la variable uniforme `illumination` pour appliquer ou non l'illumination de Phong. [1 pt]
- E9. (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 pts]

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
CROCHETGAUCHE	Inverser l'avancement du temps (il recule)
CROCHETDROIT	Remettre l'avancement du temps normalement
g	Permuter l'affichage en fil de fer ou plein
c	Permuter l'affichage des faces arrières
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 Z
BAS	Diminuer la dimension de la boîte en Z
PLUS	Incrémenter la distance de la caméra
MOINS	Décrémenter la distance de la caméra
i	Calculer ou non l'illumination de Phong
0	Afficher le point de vue « normal » (extérieur à l'aquarium)
1	Afficher le point de vue de la théière 1
2	Afficher le point de vue de la théière 2
3	Afficher le point de vue de la théière 3
4	Afficher le point de vue de la théière 4
5	Afficher le point de vue de la théière 5
6	Afficher le point de vue de la théière 6
7	Afficher le point de vue de la théière 7
8	Afficher le point de vue de la théière 8
9	Afficher le point de vue de la théière 9
d	Mode debug : (ne pas utiliser le stencil)
Molette	Changer la distance de la caméra

B Figures supplémentaires

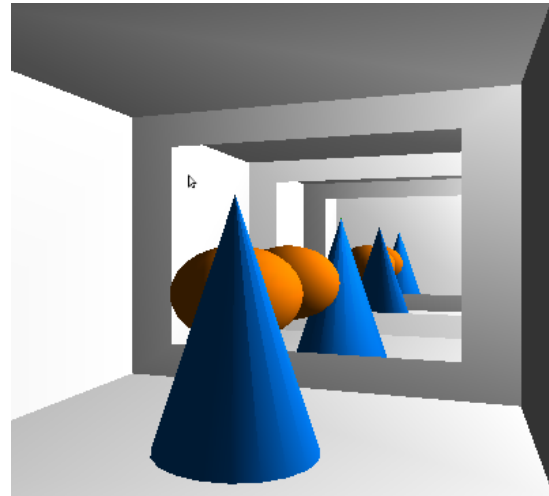


FIGURE 9 – Demi-théière avec réflexion sur la lame du couteau et réflexions multiples



FIGURE 10 – LA théière si célèbre. :)

C Apprentissage supplémentaire

Partie 1 :

1. Utiliser d'autres touches du clavier pour changer la vitesse de déplacement des théières ou contrôler l'écoulement du temps.
2. Utiliser un plan de coupe à l'intérieur de la scène pour ne pas afficher certaines régions.
3. Utiliser d'autres objets simples au lieu des théières, par exemple des cubes ou des sphères.
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 9.
5. *[Un peu plus compliqué]* Ajouter des cônes représentant la pénombre causée par chaque corps.

Partie 2 :

6. Utiliser la souris pour changer l'orientation du point de vue, tout en restant à la même position au-dessus de la théière.
7. *[Un peu plus compliqué, mais pas trop]* 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.)

Partie 3 :

8. Faites varier la couleur de l'eau dans l'aquarium en fonction de la distance à la caméra.