



**POLYTECHNIQUE
MONTRÉAL**

LE GÉNIE
EN PREMIÈRE CLASSE

INF2705 Infographie

Spécification des requis du système

Travail pratique 3

La caméra en orbite

Département de génie informatique et génie logiciel

École polytechnique de Montréal

Hiver 2015

Historique des modifications du document

Date	Version	Description	Auteur
	1.0	Version initiale	Benoît Ozell

Table des matières

1	Introduction	3
1.1	But	3
1.2	Portée	3
1.3	Références	3
2	Description globale	4
2.1	But	4
2.2	Travail demandé	4
3	Exigences	7
3.1	Exigences fonctionnelles	7
3.2	Exigences non fonctionnelles	8
A	Liste des commandes	9
B	Figure supplémentaire	9
C	Apprentissage supplémentaire	10

1 Introduction

Ce document décrit les exigences fonctionnelles et non fonctionnelles du TP3 « *La caméra en orbite* » du cours INF2705 Infographie.

1.1 But

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

1.2 Portée

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

1.3 Références

1. Site du cours INF2705
2. Site du cours INF2990
3. Guide de programmation C++ (INF2990).

2 Description globale

2.1 But

Le but de ce TP est de permettre à l'étudiant de se familiariser avec les fonctions de manipulation du tampon stencil telles que `glStencilFunc()`, `glStencilOp()` et à utiliser les plans de coupe (`glClipPlane()`). Il permettra également de mettre en pratique les fonctions de manipulation des matrices telles que `glRotate`, `glTranslate`, `glPushMatrix`, `glPopMatrix`, en utilisant une structure de données en arbre et aussi de se familiariser avec l'utilisation des nuanceurs.

2.2 Travail demandé

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

On demande de réaliser un programme permettant de réaliser une coupe d'un objet tridimensionnel en utilisant le tampon stencil. Le programme de base permet de choisir entre l'affichage d'une sphère, d'un cube ou d'une théière. Un plan de coupe doit être ajouté pour permettre d'effectuer une coupe de l'objet tridimensionnel affiché. De plus, il doit être possible de déplacer le plan de coupe selon la direction de sa normale pour permettre de varier l'endroit où la coupe se fait sur l'objet. La Figure 1 représente l'utilisation du plan de coupe sur les corps célestes alignés.

Pour que l'objet coupé constitue toujours un volume fermé, on demande d'afficher un quadrilatère correspondant au plan de coupe. Ce quadrilatère doit être affiché en faisant une vérification sur le stencil de manière à ce qu'il soit seulement dessiné sur la zone de coupe. Pour configurer le stencil, on peut tracer le modèle (les objets) deux fois : une fois pour en ne traçant que la face avant et une fois en ne traçant que la face arrière tout en modifiant le stencil de façon différente pour les faces avant ou arrière. Si on réfléchit un peu plus, on peut aussi tracer le modèle qu'une seule fois et voir que les pixels qui nous intéressent sont ceux qui n'ont été tracés qu'une seule fois.

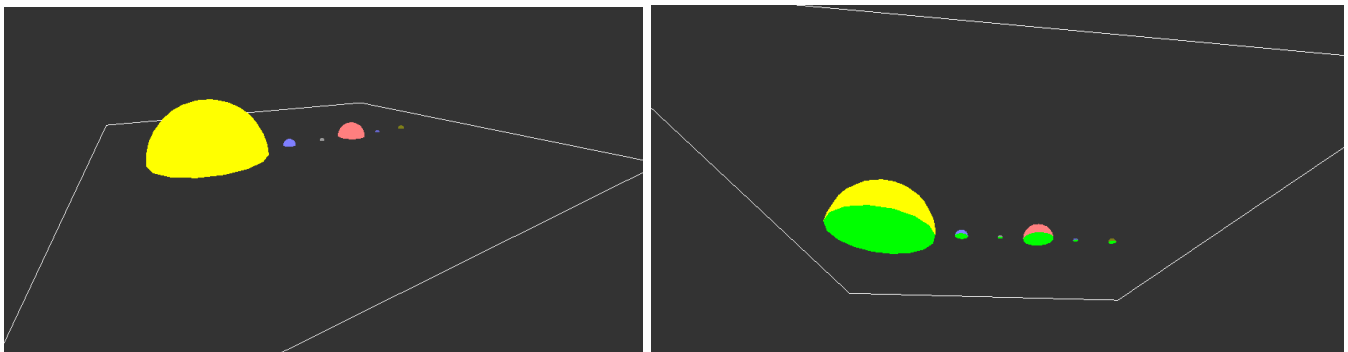


FIGURE 1 – Corps célestes coupés par le plan

Partie 2 : le système planétaire

On demande de réaliser un programme permettant l'affichage d'un système planétaire. Le système évolue en fonction du temps pour permettre aux planètes et satellites de faire des rotations sur eux-mêmes et par rapport au Soleil. L'utilisation de la fonction `glutGet(GLUT_ELAPSED_TIME)` permettra que l'évolution dans le temps soit indépendante de la fréquence d'affichage.

Le système planétaire proposé comprend le Soleil, la Terre, la Lune, Jupiter, Europa et Callisto. La position de chaque satellite (Lune, Europa, Callisto) sera donnée par rapport à sa planète respective (Terre, Jupiter) ; on utilisera des `glPushMatrix` et `glPopMatrix` au besoin. De plus, l'orbite de chaque planète autour du Soleil sera dessinée sous forme de disque (Figure 2). Enfin, le Soleil sera partiellement transparent pour que l'on puisse voir ce qui se passe « derrière » le Soleil (Figure 3).

On pourra observer le système planétaire d'un point de vue exoplanétaire où on verra tous les corps célestes (voir Figure 2). On pourra aussi observer le système planétaire à partir d'un point fixe au pôle Nord de la planète Terre (voir Figure 4). On devra alors remplacer l'appel à `gluLookAt()` par des appels aux fonctions `glTranslate()` et `glRotate()` afin de construire une matrice contenant les opérations nécessaires au positionnement de l'observateur au pôle Nord de la Terre. Cette matrice, inversée, servira ensuite à définir le changement de repère afin de dessiner le système solaire du point de vue de la planète Terre. Afin que la caméra reste fixe et en place au pôle Nord, aucun mouvement de la caméra grâce à un contrôle de l'utilisateur (clavier, souris) ne sera permis. On verra alors passer devant la caméra les planètes et leurs satellites, ainsi que le Soleil.

Partie 3 : les nuanceurs

Enfin, on le sait, la température sur Terre se réchauffe et on supposera qu'il en est de même sur tous les corps célestes de notre système (à l'exception du Soleil). On utilisera donc une paire de nuanceurs pour modifier la couleur des corps célestes selon la latitude. D'abord, on interpolera linéairement une nouvelle couleur `couleurPoleRechauffe` aux pôles selon le facteur `facteurRechauffement` pouvant varier interactivement entre 0.0 (= froid = blanc) et 1.0 (= chaud = la couleur donnée du corps céleste). Ensuite, on interpolera linéairement la couleur de chaque fragment selon la latitude : la couleur variera alors uniformément entre la couleur donnée du corps céleste à l'équateur jusqu'à la couleur `couleurPoleRechauffe` aux pôles (voir Figure 5). (La latitude aura été calculée au préalable dans le nuanceur de sommets en utilisant les valeurs de `gl_Vertex`.)

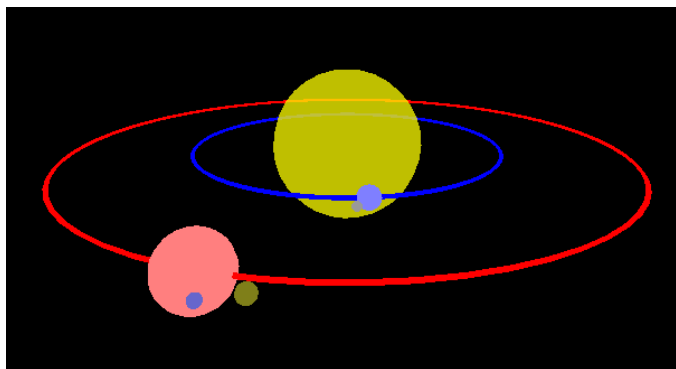


FIGURE 2 – Soleil et autres corps célestes

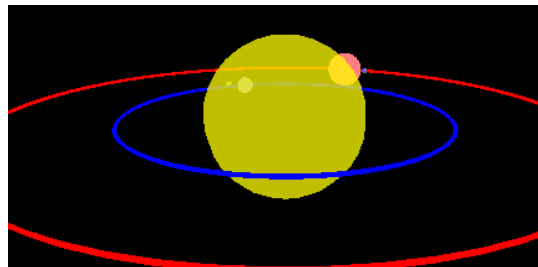


FIGURE 3 – Transparence du Soleil

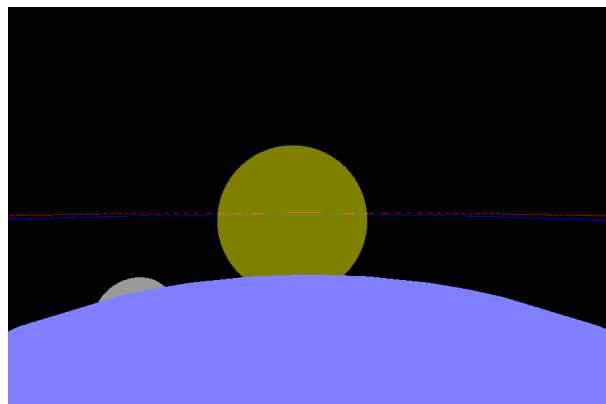
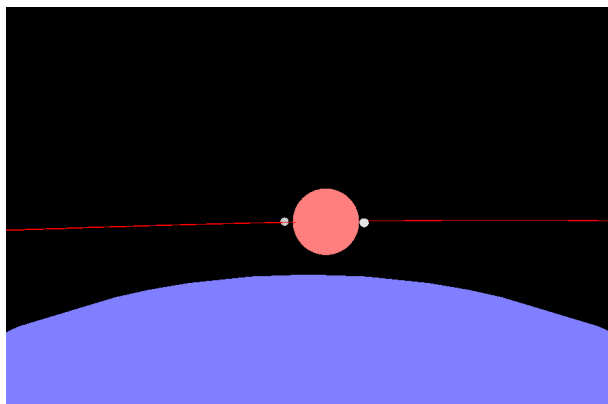


FIGURE 4 – Rotation des corps célestes vue du pôle Nord de la Terre

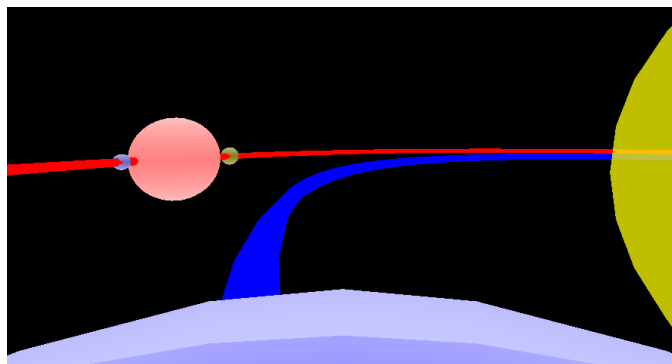
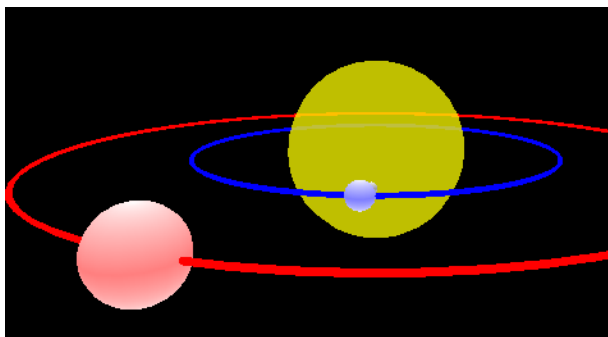


FIGURE 5 – Variation des couleurs selon le réchauffement (froid et chaud)

3 Exigences

3.1 Exigences fonctionnelles

Partie 1 :

- E1. L'utilisation d'un plan de coupe (`glClipPlane()`) permet d'effectuer une coupe verticale pour chacun des solides tridimensionnels.
- E2. Le solide coupé est fermé en utilisant le tampon stencil pour dessiner un quadrilatère dont l'affichage est limité aux pixels de la coupe.
- E3. On peut déplacer le plan de coupe selon la direction de sa normale.
- E4. On peut afficher le contour d'un quadrilatère représentant le plan de coupe (`GL_LINE_LOOP`).
- E5. Le logiciel utilise correctement les touches listées à l'annexe A.

Partie 2 :

- E6. Les positions des satellites dépendent de celles des planètes et chaque corps céleste utilise des périodes de rotation et de révolution qui lui sont propres.
- E7. Les fonctions `glRotate` et `glTranslate` sont correctement utilisées pour positionner et faire la rotation des différents corps.
- E8. Les fonctions `glPushMatrix` / `glPopMatrix` sont correctement utilisées pour sauvegarder l'état des matrices pour le dessin de chacune des planètes.
- E9. Le temps est correctement utilisé pour gérer l'évolution du système. (Les angles sont incrémentés en fonction du temps écoulé.)
- E10. L'orbite des planètes est dessinée sous forme de disque.
- E11. Le Soleil est partiellement transparent.
- E12. Le mouvement des planètes peut être observé de deux façons : globalement d'un point de vue externe au système (déjà implanté) ou d'un point d'observation situé au pôle Nord de la planète Terre.
- E13. Le calcul du positionnement de l'observateur au pôle est fait en utilisant les fonctions matricielles d'OpenGL et non en calculant explicitement la position des autres corps par rapport à la Terre.
- E14. Du point de vue situé au pôle, on peut voir passer tous les autres corps (planètes, satellites, Soleil) et voir aussi une partie de la Terre.

Partie 3 :

- E15. Lorsque les nuanceurs sont utilisés, la couleur des pôles varie selon le facteur de réchauffement.
- E16. Lorsque les nuanceurs sont utilisés, la couleur des planètes varie linéairement entre l'équateur et les pôles.

3.2 Exigences non fonctionnelles

La totalité du dessin peut se faire dans les fonctions `afficherScene()` et `dessinerModele()` (et quelques autres identifiées), de même que dans les fichiers de nuanceurs. Vous pouvez cependant ajouter des fonctions supplémentaires ainsi que d'autres classes et/ou fichiers si vous le jugez nécessaire.

ANNEXES

A Liste des commandes

Touche	Description
Mouvements souris avec bouton 1 enfoncé	Déplacer (angles) le point de vue externe
;	Choix du modèle : 1-sphère, 2-cube, 3-théière (déjà implanté)
< / >	Déplacement du plan de coupe
+ / -	Incrémenter / décrémenter la distance de l'observateur
g	Passage fil de fer ou plein
0	Affichage d'un point de vue globale/externe
1	Affichage d'un point de vue situé au pôle Nord
f, c	Incrémenter / décrémenter le facteur de réchauffement
x	Passage entre le mode pipeline fixe et utilisation des nuanceurs
v	Recharge les fichiers des nuanceurs et recrée le programme
ESC	Arrêt de l'application

B Figure supplémentaire



FIGURE 6 – Demi-théière avec réflexion sur la lame du couteau

C Apprentissage supplémentaire

Partie 1 :

1. Changer l'orientation du plan de coupe.
2. Utiliser deux ou trois plans de coupe.
3. Utiliser des objets non complètement fermés.
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 6.

Partie 2 :

5. Donner une orbite elliptique aux planètes et satellites. Changer l'orientation du plan de chaque orbite.
6. Utiliser le clavier pour changer la vitesse de révolution des planètes, contrôler l'écoulement du temps, s'approcher ou s'éloigner du système.
7. Notez que les images montrées à la Figure ?? présentent un point de vue qui n'est pas exactement au pôle Nord, mais plutôt un peu en retrait. Déplacez le point de vue afin d'obtenir ces images.
8. Afficher le point de vue à partir d'autres endroits sur la Terre.
9. Afficher le point de vue qu'un astronaute aurait à partir navette spatiale en orbite autour de la Terre.

Notez qu'il n'est pas souhaitable dans ce TP de respecter les dimensions exactes de notre système solaire. En effet, les quatre premières planètes apparaîtraient alors très (trop !) petites par rapport au Soleil et seraient aussi situées trop près du Soleil pour produire un affichage où on voit facilement leurs mouvements. Il serait alors difficile de bien évaluer votre utilisation des énoncés OpenGL. Pour les intéressés, les paramètres réels des planètes de notre système solaire sont donnés, entre autres, à <http://www.astro-rennes.com/planetes/intro.php>.