

# **INF2705 Infographie**

## **Spécification des requis du système**

### **Travail pratique 3**

### ***La caméra en orbite***

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	But . . . . .	2
1.2	Portée . . . . .	2
1.3	Références . . . . .	2
<b>2</b>	<b>Description globale</b>	<b>3</b>
2.1	But . . . . .	3
2.2	Travail demandé . . . . .	3
2.3	Fichiers fournis . . . . .	6
<b>3</b>	<b>Exigences</b>	<b>7</b>
3.1	Exigences fonctionnelles . . . . .	7
3.2	Exigences non fonctionnelles . . . . .	7
<b>A</b>	<b>Liste des commandes</b>	<b>8</b>
<b>B</b>	<b>Figure supplémentaire</b>	<b>8</b>
<b>C</b>	<b>Apprentissage supplémentaire</b>	<b>9</b>

# 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. Il permettra également de mettre en pratique les fonctions de manipulation des matrices telles que `Rotate` et `Translate`, 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 de stencil. Le programme de base affiche un système planétaire et permet de choisir l'affichage de corps célestes sphériques ou cubiques. 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). 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. La trajectoire des planètes autour du Soleil est représentée par un tore (Figure 1).

On vous demande d'abord d'afficher le Soleil partiellement transparent afin que l'on puisse voir les corps célestes qui se passent « derrière » le Soleil (Figure 2).

On vous demande aussi d'ajouter un plan de coupe afin de permettre la coupe des objets tridimensionnels affichés. On pourra déplacer le plan de coupe selon la direction de sa normale afin de varier l'endroit où la coupe se fait. La Figure 3 représente l'utilisation du plan de coupe sur les corps célestes.

Pour que l'objet coupé constitue toujours un volume fermé, on affichera un quadrilatère correspondant au plan de coupe. Ce quadrilatère doit être affiché en faisant une vérification du stencil de manière à ce que le plan ne soit seulement dessiné sur la zone de coupe. En plus de ce quadrilatère opaque affiché conditionnellement au stencil, on affichera aussi le quadrilatère complet en transparence.

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 constater que les pixels qui nous intéressent sont ceux qui n'ont été tracés qu'une seule fois.

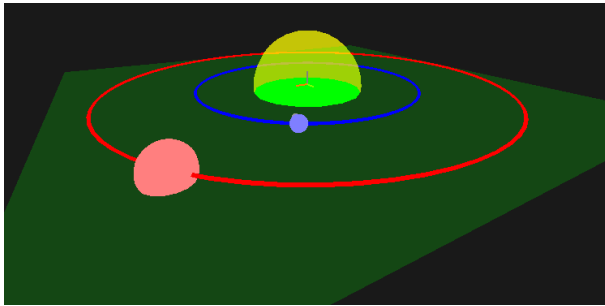


FIGURE 1 – Soleil et autres corps célestes

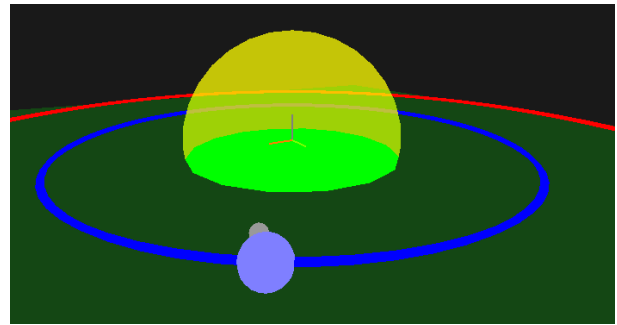


FIGURE 2 – Transparence du Soleil

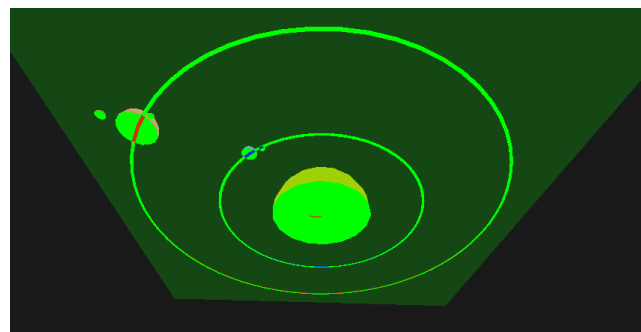
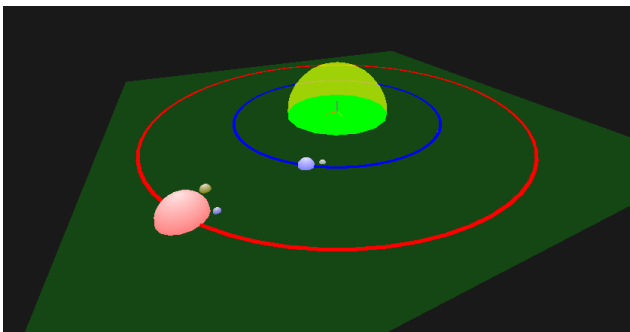


FIGURE 3 – Corps célestes coupés par le plan de coupe

## Partie 2 : des points de vue différents et le réchauffement planétaire

Dans la version de base, on peut observer le système planétaire d'un point de vue exoplanétaire où on voit tous les corps célestes (voir Figure 1). On vous demande d'ajouter un autre point de vue qui permettra d'observer le système planétaire à partir d'un point fixe au pôle Nord de la planète Terre (voir Figure 4). Vous devrez alors remplacer l'appel à `LookAt()` par des appels aux fonctions `Translate()` et `Rotate()` 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. (La librairie GLM fournit la méthode « `Minv = glm::inverse(M)` » qui sera bien utile ici.) Afin que la caméra reste fixe et en place au pôle Nord, aucun mouvement de la caméra par le clavier ou la souris ne sera permis. On verra alors passer devant la caméra les planètes et leurs satellites, ainsi que le Soleil.

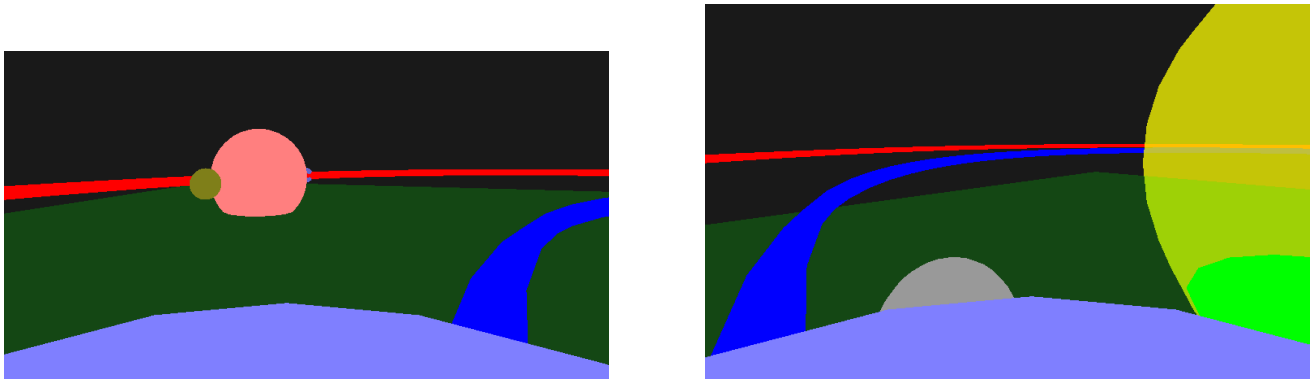


FIGURE 4 – Rotation des corps célestes vue du pôle Nord de la Terre

D'autre part, 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. Dans les nuanceurs, on modifiera donc la couleur des planètes selon la latitude (qui sera obtenue de la coordonnée *z* du *Vertex* courant). D'abord, on interpolera linéairement une nouvelle couleur `couleurPoleRechauffe` aux pôles selon le facteur `facteurRechauffement` pouvant varier 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 `couleurPoleRechauffe` aux pôles (voir Figure 5).

Pour ajouter un bel effet à l'ensemble du dessin, on atténuera les valeurs RGB de la couleur selon la distance à caméra comme illustrée à la Figure 6. Cette distance pourrait être calculée dans le nuanceur de sommets par : « `d = -( matrVisu * matrModel * Vertex ).z;` », mais on peut aussi l'obtenir dans le nuanceur de fragments par : « `d = gl_FragCoord.z / gl_FragCoord.w;` ». Dans ce TP, on appliquera l'atténuation de la couleur dans l'intervalle [20,40] : aucune atténuation (=couleur originale) lorsque  $d < 20$ , linéaire entre 20 et 40 et complète (=du noir) lorsque  $d > 40$ . (*Attention : Utilisez « `clamp()` » plutôt que « `if (...)` » !*)

Enfin, dans le simple 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 n'ajoutera aucun traitement particulier, mais vous devrez toutefois regrouper les attributs dans des structures `AttribsIn` et `AttribsOut` et utiliser celles-ci dans les trois nuanceurs :

```
in Attribs {
    float proportionLatitude;
    vec4 couleurPoleRechauffe;
    vec4 couleur;
} AttribsIn;

out Attribs {
    float proportionLatitude;
    vec4 couleurPoleRechauffe;
    vec4 couleur;
} AttribsOut;
```

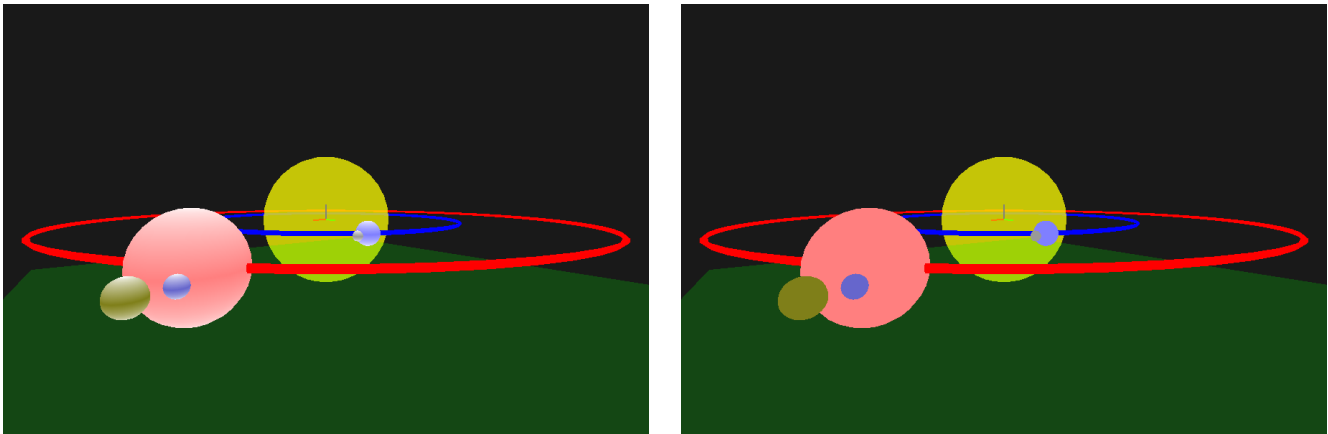


FIGURE 5 – Variation des couleurs selon le réchauffement (froid et chaud)

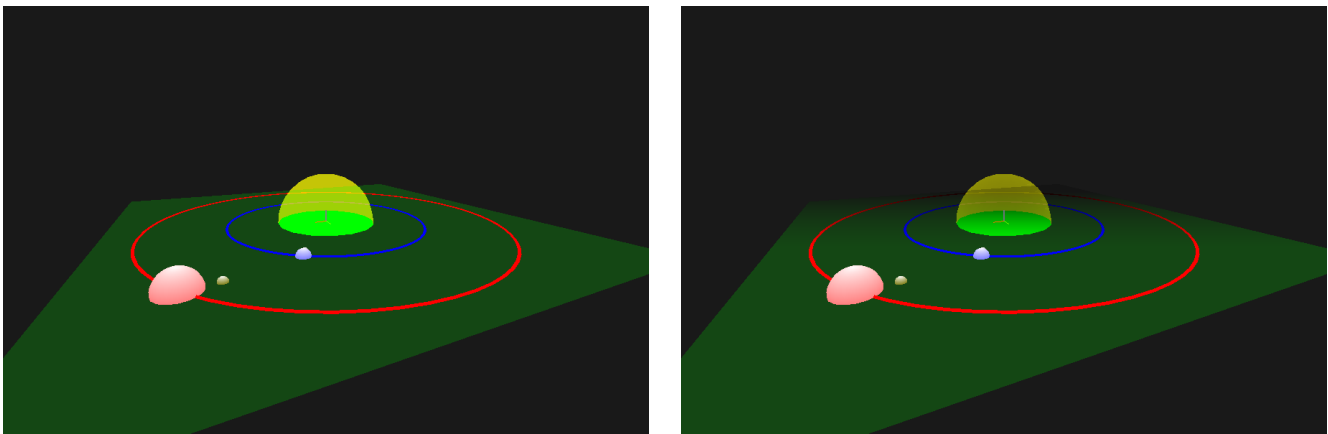


FIGURE 6 – Atténuation en fonction de la profondeur

## 2.3 Fichiers fournis

Le code fourni présente le Soleil, la planète Terre et la planète Jupiter, ainsi que leurs satellites.

## 3 Exigences

### 3.1 Exigences fonctionnelles

Partie 1 :

- E1. L'utilisation d'un plan de coupe permet d'effectuer une coupe horizontale 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 et l'affichage est correct.
- E4. Le quadrilatère représentant le plan de coupe est affiché en transparence ( $\alpha = 0.2$ ).
- E5. Le logiciel utilise correctement les touches listées à l'annexe A.

Partie 2 :

- E6. Le Soleil est transparent ( $\alpha = 0.5$ ).
- E7. Le mouvement des planètes peut être observé de deux façons : d'un point de vue externe au système (déjà implanté) ou d'un point d'observation situé au pôle Nord de la planète Terre.
- E8. Le calcul du positionnement de l'observateur au pôle est fait en utilisant les matrices de `glm` et non en calculant explicitement la position des autres corps par rapport à la Terre.
- E9. 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.
- E10. La couleur des pôles varie selon le facteur de réchauffement.
- E11. La couleur des planètes varie linéairement entre l'équateur et les pôles.
- E12. La couleur peut être atténuée selon la profondeur.
- E13. Un nuanceur de géométrie est utilisé.

### 3.2 Exigences non fonctionnelles

La totalité du dessin peut se faire dans les fonctions `afficherScene()` et `afficherModele()` (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)
[ ou ]	Déplacement du plan de coupe
+ ou -	Incrémenter ou 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 ou c	Incrémenter ou décrémenter le facteur de réchauffement
p	Atténuer ou non la couleur selon la profondeur
v	Recharge les fichiers des nuances et recrée le programme
ESC	Arrêt de l'application

### B Figure supplémentaire



FIGURE 7 – 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 (plutôt que des sphères).
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 7.

### 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 4 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 d'une 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>.