

INF2705 Infographie

Spécification des requis du système

Travail pratique 1

le petit robot

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	Travail demandé	3
3	Exigences	7
3.1	Exigences fonctionnelles	7
3.2	Exigences non fonctionnelles	7
3.3	Rapport	8
A	Liste des commandes	8
B	Apprentissage supplémentaire	9
C	Figures supplémentaires	10

1 Introduction

Ce document décrit les exigences du TP1 « *le petit robot* » 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`, `*.gls1`, `*.txt`).

2 Description globale

2.1 But

Le but de TP est de permettre à l'étudiant de mettre en pratique les fonctions de contrôle du pipeline graphique d'OpenGL pour la modification des matrices et la manipulation de la caméra synthétique : `Rotate()`, `Translate()`, `Scale()`, `PushMatrix()` et `PopMatrix()`.

Ce travail pratique lui permettra aussi d'utiliser les fonctions liées aux *Vertex Buffer Objects (VBOs)* : `glGenBuffers()`, `glBindBuffers()`, `glBufferData()` et `glDrawElements()`.

2.2 Travail demandé

Partie 1 : le robot

On demande de réaliser un programme permettant d'afficher un robot ailé (peut-être d'origine extra-terrestre !) avec un corps (qui est aussi sa tête) sphérique de taille variable ($\text{rayon} = \text{taille}$). Des yeux jaunes ronds situés au milieu de la tête ($\text{rayon} = 0.45 * \text{taille}$) à l'avant du robot. Les yeux sont placés à 60 degrés d'écart (30 degrés vers la gauche pour un oeil et 30 degrés vers la droite pour l'autre). Dans la deuxième partie de ce TP, vous ajouterez la possibilité pour ce robot de changer sa tête en LA théière bien connue en infographie !

Il possède deux bras et jambes en parallélépipède allongé de taille fixe ($\text{longMembre} \times \text{largMembre}$) formés par des cubes étirés. Les jambes marrons sont figés en bas de la tête du robot à 60 degrés d'écart (30 degrés vers la gauche pour une jambe et 30 degrés vers la droite pour l'autre). La Figure 1 montre la tête du robot sous ses deux formes : sphère ou théière.

Le cylindre, la sphère, les parallélépipèdes et les quadrilatères sont tracés par des appels aux fonctions fournies (sans modifier ces fonctions). La tête du robot est une sphère positionnée au centre du corps. Chaque aile est composée d'un quadrilatère et articulée selon un angle (angleAile), tandis que chaque membre est représenté par un cube étiré.

Les bras cyan du robot sont un peu spéciaux, ils sont situés sur chaque côté de la tête, tournés à 80 degrés selon l'axe des X et ont un mouvement de rotation autour de la tête (axe Z) suivant l'angle de la variable angleBras en partant du côté vers l'arrière du robot (et vice-versa) : voir Figure 2.

Le robot possède une antenne composée de deux cylindres de taille différents. La première partie est un cylindre ($\text{longueur} = \text{taille}$, $\text{largeur} = (1/3) * \text{taille}$) fixe au-dessus de la tête. Une seconde partie est un cylindre ($\text{longueur} = (1/3) * \text{taille}$, $\text{largeur} = \text{taille}$) qui se trouve sur la première partie de l'antenne et tourne deux fois plus vite que angleCorps sur elle même selon l'axe Z.

Enfin, le robot tourne aussi sur lui-même (angleCorps) et se déplace (position) dans l'espace de la boîte cubique (voir Figure 3). Les valeurs de toutes les variables sont contrôlées interactivement.

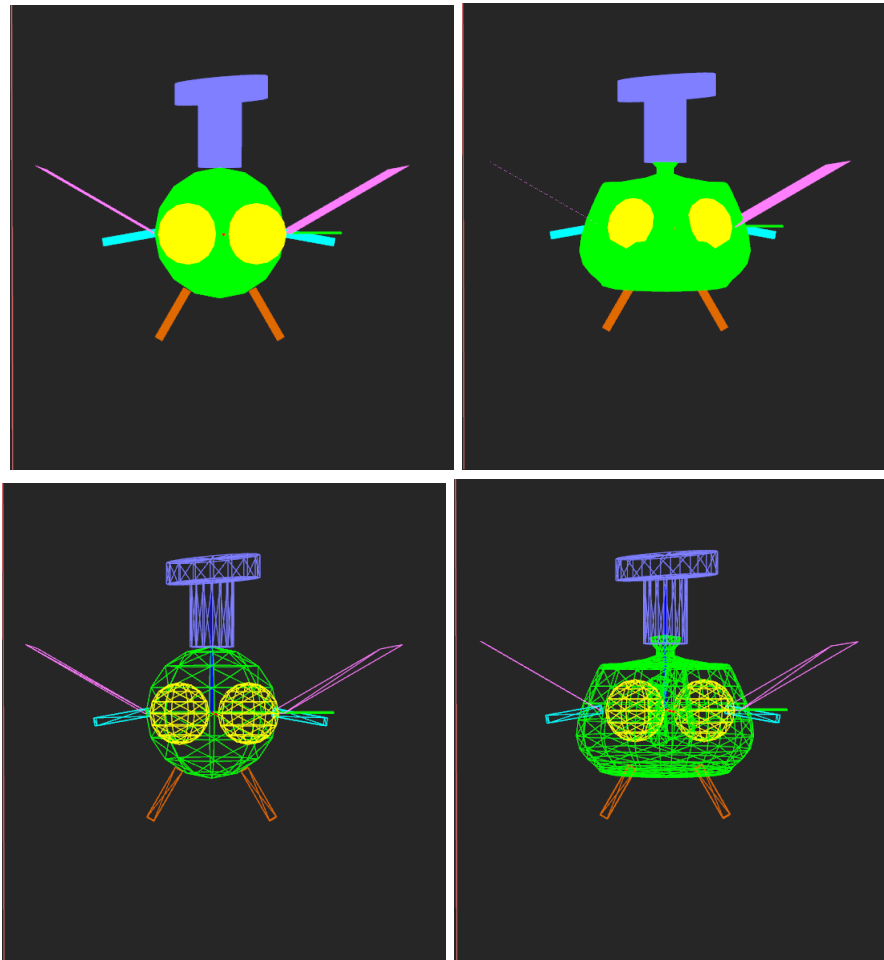


FIGURE 1 – La tête du robot sous la forme d'une sphère ou d'une théière

Partie 2 : la caméra synthétique et utilisation de *Vertex Buffer Objects (VBOs)*

On pourra représenter la tête du bonhomme par la sphère ou par la célèbre théière. La théière sera affichée en utilisant deux VBOs (sommets et indices) créés avec les deux tableaux définis dans le fichier déjà inclus « `inf2705-theiere.h` ». Ces VBOs doivent être définis une seule fois à l'initialisation et ensuite réutilisés à chaque affichage.

Pour vous inspirer, vous pouvez utiliser le triangle dans l'exemple vu au cours :

`cours.polymtl.ca/inf2705/exemples/03-Base/main.cpp`,
`cours.polymtl.ca/inf2705/exemples/03-Base/`.

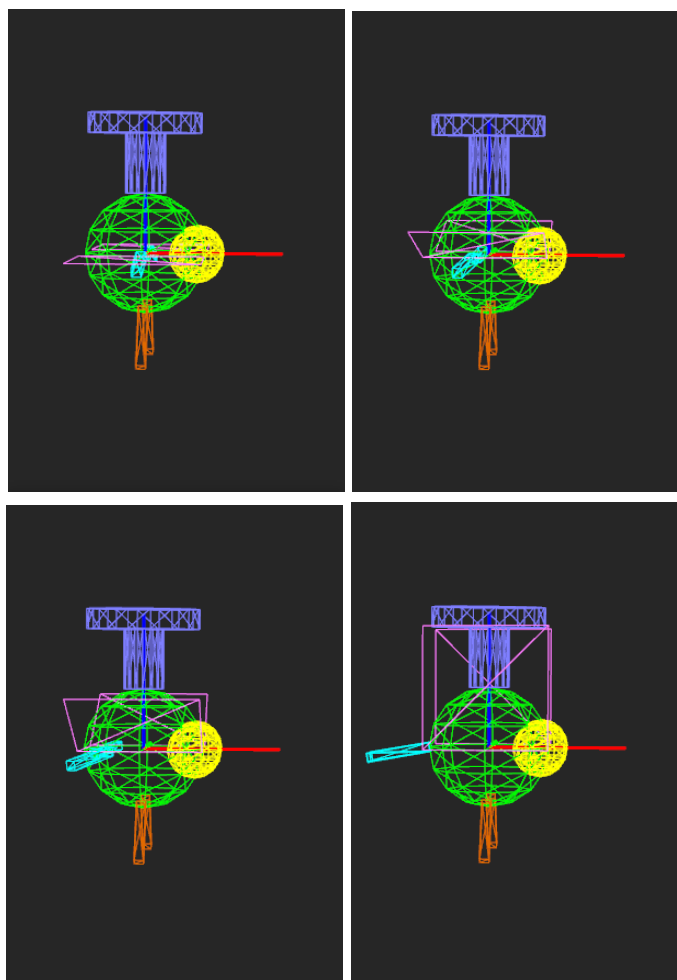


FIGURE 2 – Articulation des jambes et des ailes

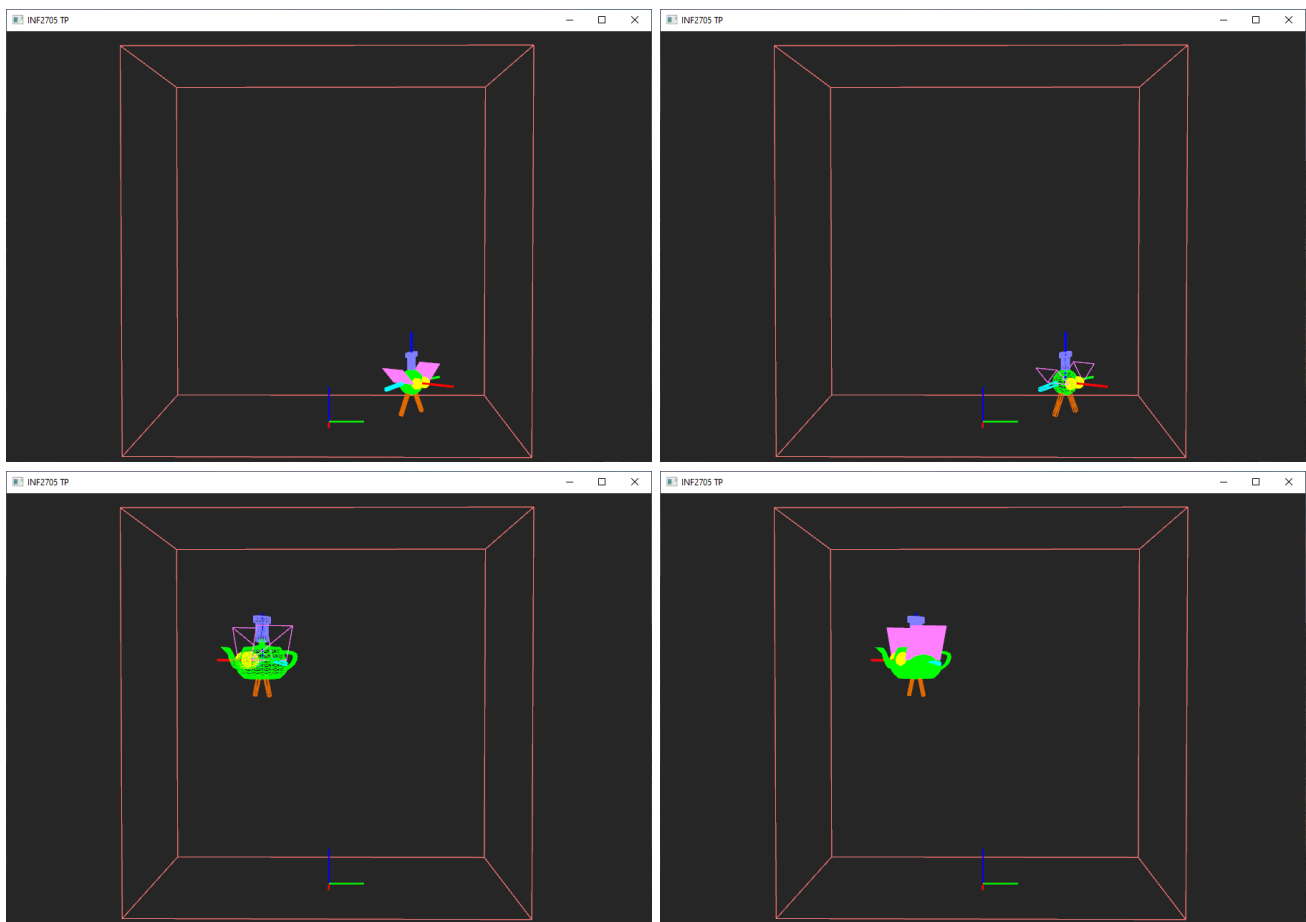


FIGURE 3 – Déplacement, rotation et articulation des bras et des ailes en mode animation

3 Exigences

3.1 Exigences fonctionnelles

Partie 1 :

- E1. Le corps, la tête, les ailes, les bras et les jambes sont dessinés en utilisant les fonctions définies `afficherCylindre()`, `afficherSphere()`, `afficherQuad()` et `afficherCube()`. [2 points]
- E2. Les fonctions `Rotate()`, `Translate()` et `Scale()` sont correctement utilisées pour les transformations géométriques nécessaires au dessin de chaque partie du bonhomme. [3 points]
- E3. Le corps du bonhomme est positionné selon `position`. Le rayon de son corps/tête est donné par `taille`. Les yeux sont bien positionnés. [2 points]
- E4. La tête du bonhomme est bien positionnée au centre du corps. L'antenne est bien positionnée. [1 point]
- E5. La rotation du corps du bonhomme suit `angleCorps`. [1 point]
- E6. Les bras et les jambes du bonhomme sont positionnés aux arêtes de la sphère et chaque membre est de rayon `largMembre` et de longueur `longMembre`. [1 point]
- E7. Les ailes du bonhomme sont positionnées aux arêtes du cylindre et chaque aile est carrée, d'arête `taille`. [1 point]
- E8. Les fonctions `PushMatrix()` et `PopMatrix()` sont correctement utilisées pour sauvegarder l'état des matrices. [2 points]

Partie 2 :

- E9. Les fonctions `glGenBuffers()`, `glBindBuffers()`, `glBufferData()` et `glDrawElements()` sont correctement utilisées afin d'utiliser deux VBOs (sommets et indices) pour afficher la théière. Les VBOs sont définis à l'initialisation et réutilisés à chaque affichage. [3 points]
- E10. La tête du bonhomme peut être affichée en utilisant cette théière telle qu'illustrée à la Figure 1. [2 points]
- E11. (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]

Pour la partie 1, des modifications sont principalement à faire dans les fonctions `afficher*()`. Pour la partie 2, des modifications sont principalement à faire dans les fonctions `initialiserGraphique()` et `afficherTheiere()`.

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
i	Réinitialiser le point de vue et le bonhomme
g	Permuter l'affichage en fil de fer ou plein
c	Permuter l'affichage des faces arrières
l	Utiliser LookAt ou Translate+Rotate pour placer la caméra
m	Choisir le modèle affiché : cube, théière
MOINS	Reculer la caméra
PLUS	Avancer la caméra
DROITE	Déplacer le bonhomme vers +X
GAUCHE	Déplacer le bonhomme vers -X
PAGEPREC	Déplacer le bonhomme vers +Y
PAGESUIV	Déplacer le bonhomme vers -Y
HAUT	Déplacer le bonhomme vers +Z
BAS	Déplacer le bonhomme vers -Z
f	Diminuer la taille du corps
r	Augmenter la taille du corps
VIRGULE	Diminuer l'angle de rotation du bonhomme
POINT	Augmenter l'angle de rotation du bonhomme
o	Diminuer l'angle des bras
p	Augmenter l'angle des bras
j	Diminuer l'angle des ailes
u	Augmenter l'angle des ailes
b	Incrémenter la dimension de la boîte
h	Décrémenter la dimension de la boîte
ESPACE	Mettre en pause ou reprendre l'animation
BOUTON GAUCHE	Déplacer (modifier angles) la caméra

B Apprentissage supplémentaire

1. Quel est le nombre minimal de `PushMatrix()`/`PopMatrix()` à utiliser ? Pourquoi faut-il éviter d'en ajouter inutilement ?
2. Allonger les bras ou les jambes selon la taille du corps.
3. Ajouter des ailes supplémentaires au robot.
4. Utiliser un octaèdre régulier au lieu d'une sphère.
5. Utiliser un cube au lieu d'une sphère pour le corps.
6. Utiliser un cylindre au lieu d'un cube pour les jambes.
7. Afin de mettre en pratique l'utilisation des transformations élémentaires, remplacer l'utilisation de la fonction `LookAt()`, en gardant le même point de vue, par une série de transformations utilisant des appels à `Translate()` et `Rotate()`.
8. Remplacer la fonction `LookAt()` qui positionne la caméra en orbite autour de la scène. Le `VecteurUp` devra être modifié en fonction de la position de l'observateur afin d'avoir l'impression que l'observateur est en orbite.
9. Si vous voulez que la tête du bonhomme se rende à une certaine position au-dessus du plan (p.e. pour y regarder quelque chose), comment allez-vous vous y prendre pour déterminer les valeurs appropriées des angles afin de positionner la tête à cet endroit ?
(C'est une question qui relève de la « cinématique inverse ».)
Y a-t-il plusieurs solutions possibles ?
Y a-t-il des solutions qu'il faut éviter ?
(Il faut aussi faire attention à ce que le bonhomme ne sorte pas de sa cage !)

C Figures supplémentaires

Allez voir la théière bien connue en infographie sur Internet :

https://fr.wikipedia.org/wiki/Théière_de_l'Utah

https://www.sjbaker.org/wiki/?title=The_History_of_The_Teapot.



FIGURE 4 – La théière dans un écran de veille (Windows 95/98) <http://www.eeggs.com/items/2422.html>

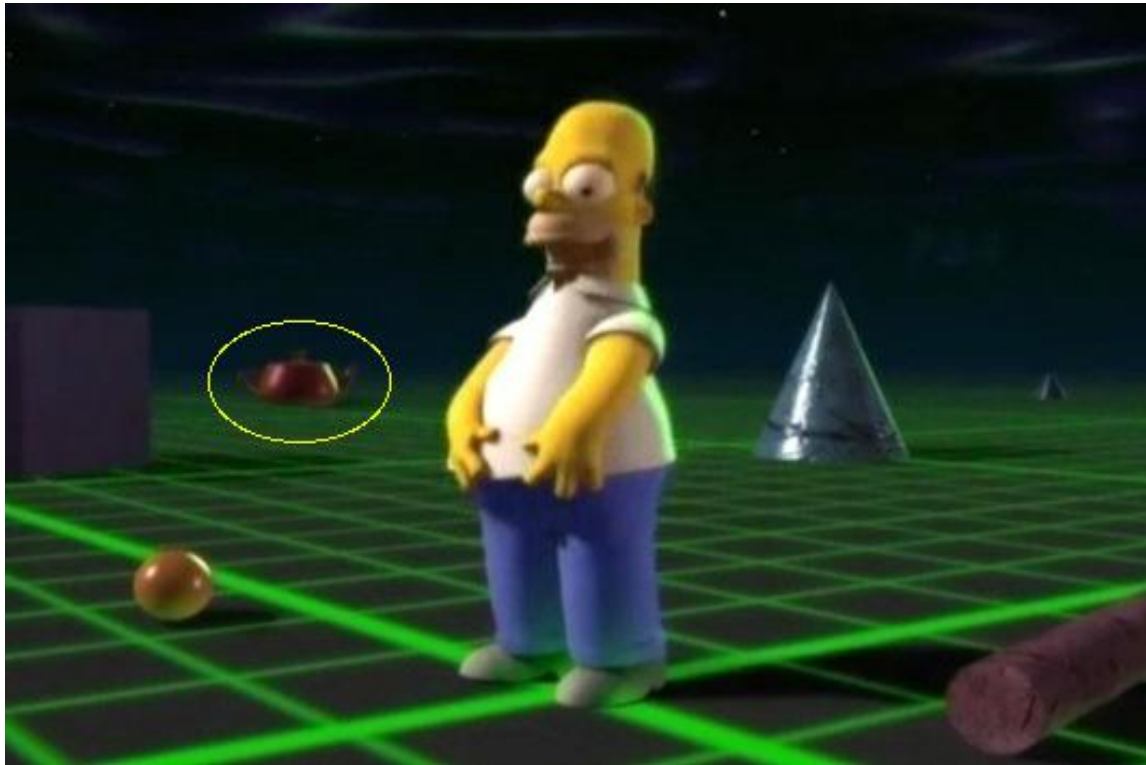


FIGURE 5 – Homer Simpson a vu la théière dans *Treehouse of Horror VI*



FIGURE 6 – La théière utilisée dans *Toy Story*