

# **INF2705 Infographie**

## **Spécification des requis du système**

### **Travail pratique 1**

### ***Utilisation du pipeline graphique et des VBO***

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	But . . . . .	2
1.2	Portée . . . . .	2
1.3	Remise . . . . .	2
<b>2</b>	<b>Description globale</b>	<b>3</b>
2.1	But . . . . .	3
2.2	Travail demandé . . . . .	3
<b>3</b>	<b>Exigences</b>	<b>7</b>
3.1	Exigences fonctionnelles . . . . .	7
3.2	Exigences non fonctionnelles . . . . .	7
3.3	Rapport . . . . .	7
<b>A</b>	<b>Liste des commandes</b>	<b>8</b>
<b>B</b>	<b>Figures supplémentaires</b>	<b>8</b>
<b>C</b>	<b>Apprentissage supplémentaire</b>	<b>11</b>

# 1 Introduction

Ce document décrit les exigences fonctionnelles et non fonctionnelles du TP1 « *Utilisation du pipeline graphique et des VBO* » 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 Remise

Faites la commande « `make remise` » afin de créer l'archive « **INF2705\_remise\_TPn.zip** » que vous déposerez ensuite dans Moodle.

Ce fichier zip contient le fichier Rapport et tout le code source du TP (`makefile`, `*.h`, `*.cpp`, `*.glsl`, `*.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 : la bestiole

On demande de réaliser un programme permettant d'afficher une bestiole un peu bizarre (probablement d'origine extraterrestre !) avec une tête sphérique de rayon fixe, un corps cylindrique de taille variable (`bestiole.taille`), deux ailes carrées (même taille que le corps) et quatre pattes cylindriques de taille fixe (`bestiole.longPatte` x `bestiole.rayonPatte`) formées par des cylindres étirés. Cette bestiole pourra aussi transformer son corps en LA théière bien connue en infographie ! La Figure 1 montre cette bestiole sous ses deux formes : cylindre et théière.

Les cylindres, la sphère et les quadrilatères sont tracés par des appels aux fonctions fournies (sans modifier ces fonctions). La tête de la bestiole est une sphère positionnée sur l'arête supérieure du corps (en X+ et Z+). Chaque aile est composée d'un quadrilatère et articulé selon un angle (`bestiole.angleAile`), tandis que chaque patte est composée d'un cylindre étiré et articulé selon un angle (`bestiole.anglePatte`) (voir Figure 2). Enfin, la bestiole peut aussi tourner sur elle-même (`bestiole.angleCorps`) et se déplacer (`bestiole.position`) dans l'espace de la boîte (voir Figure 3). Les valeurs de toutes les variables sont contrôlés interactivement.

*Note* : Ce TP utilise OpenGL 4.x et certains appels à OpenGL sont un peu différents des appels vus au TP0. En classe, nous avons parlé de l'utilisation de la librairie `glm` et de `std::stack` qui sont utilisés dans la classe `MatricePipeline` du fichier `inf2705-matrice.h`. Consultez les notes de cours et explorez la classe `MatricePipeline` afin de bien comprendre son fonctionnement.

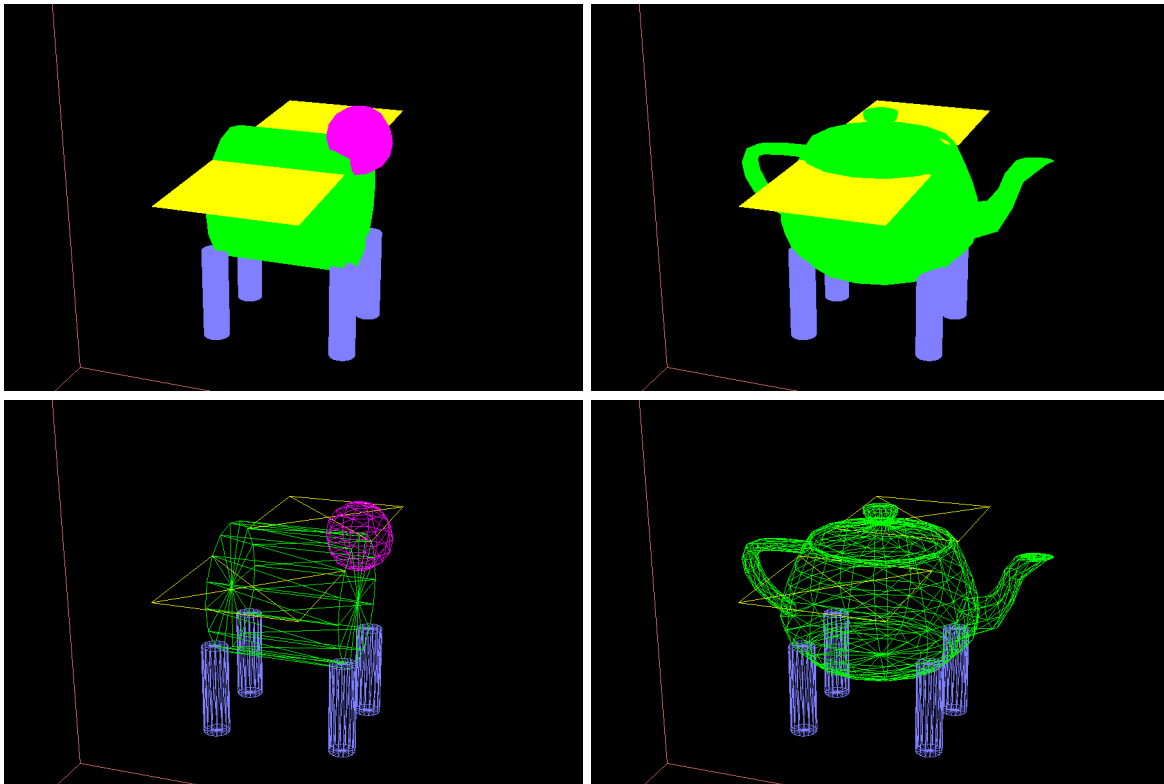


FIGURE 1 – Bestiole sous la forme d'un cylindre ou d'une théière

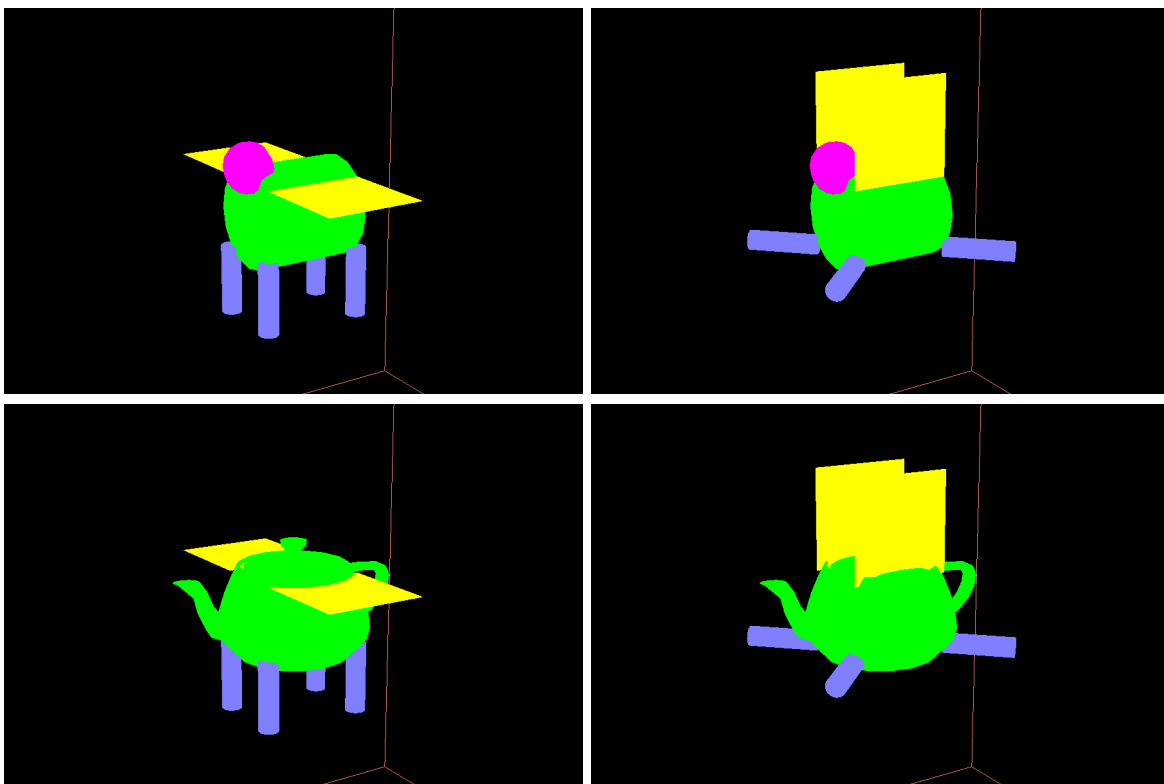


FIGURE 2 – Articulation des pattes

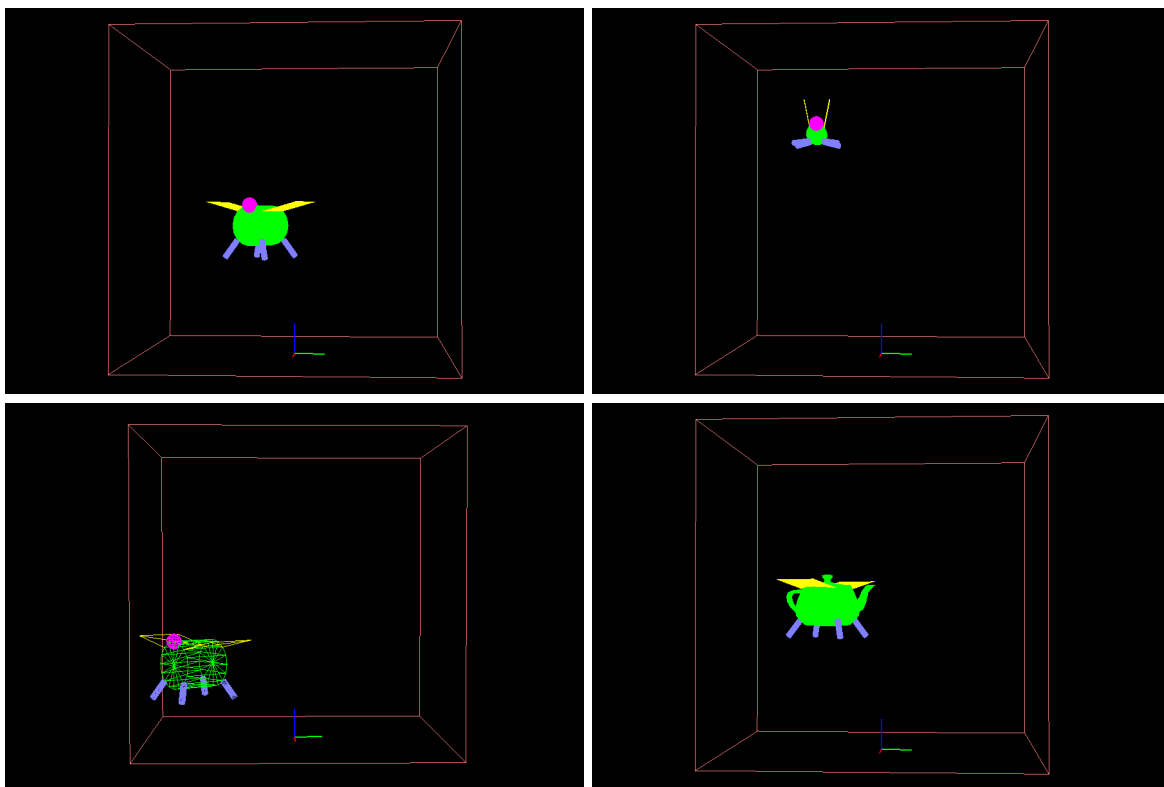


FIGURE 3 – Déplacement, rotation et articulation des pattes en mode animation

## Partie 2 : utilisation de *Vertex Buffer Objects (VBOs)*

Le corps de la bestiole pourra être représenté par un cylindre 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.

L'orientation de cette théière variera linéairement en fonction de sa hauteur (afin de verser le thé !), tel qu'illustré à la Figure 4 : à l'horizontale (rotation de  $0^\circ$ ) lorsque la théière est au sol ; à la verticale (rotation de  $90^\circ$ ) lorsque la théière est tout en haut.

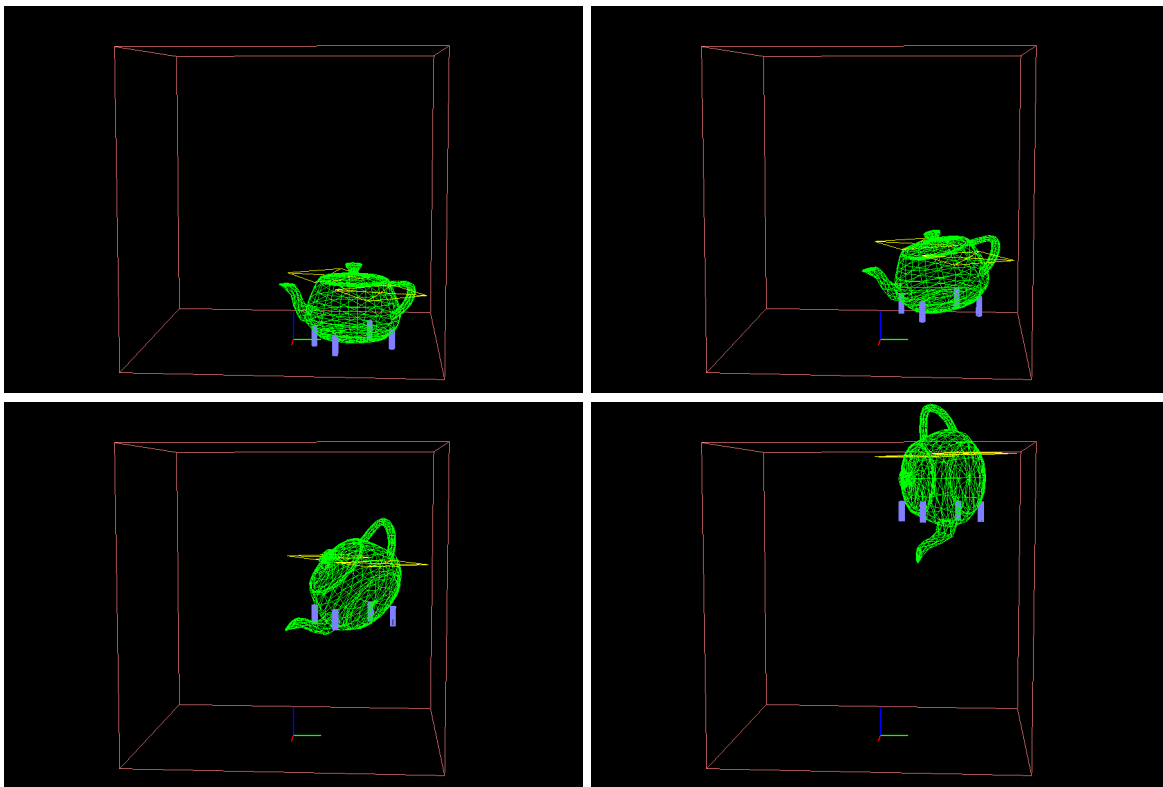


FIGURE 4 – Variation de l'orientation de la théière en fonction de sa hauteur

## 3 Exigences

### 3.1 Exigences fonctionnelles

Partie 1 :

- E1. Le corps, la tête, les ailes et la pattes sont dessinés en utilisant les fonctions déjà présentes `afficherCylindre()`, `afficherSphere()` et `afficherQuad()`.
- E2. Les fonctions `Rotate()`, `Translate()` et `Scale()` sont correctement utilisées pour les transformations géométriques nécessaires au dessin de chaque partie de la bestiole.
- E3. Les fonctions `PushMatrix()` et `PopMatrix()` sont correctement utilisées pour sauvegarder l'état des matrices pour le dessin de chaque patte.
- E4. La bestiole est positionnée selon `bestiole.position` et la taille de son corps est donnée par `bestiole.taille`.
- E5. La tête de la bestiole est bien positionnée au milieu de l'arête.
- E6. La rotation du corps de la bestiole suit `angleBestiole`.
- E7. Les pattes de la bestiole sont positionnées aux arêtes du cylindre et chaque patte est de rayon `bestiole.rayonPatte` et de longueur `bestiole.longPatte`.

Partie 2 :

- E8. 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.
- E9. Le corps de la bestiole peut être affiché en utilisant cette théière tel qu'illustré à la Figure 1. L'orientation de la théière varie linéairement en fonction de sa hauteur tel qu'illustré à la Figure 4.
- E10. (Le logiciel utilise correctement les touches listées à l'annexe A pour faire varier les divers paramètres.)

### 3.2 Exigences non fonctionnelles

Pour la partie 1, des modifications sont principalement à faire dans la fonction `afficherBestiole()`. Pour la partie 2, des modifications sont principalement à faire dans les fonctions `initiliaser()` 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.)



## ANNEXES

### A Liste des commandes

Touche	Description
q	Quitter l'application
x	Activer/désactiver l'affichage des axes
i	Réinitialiser le point de vue
l	Utiliser LookAt ou Translate+Rotate pour placer la caméra (n'est pas un requis)
g	Permuter l'affichage en fil de fer ou plein
c	Permuter l'affichage des faces arrières
m	Choisir le modèle affiché : cube, théière
MOINS	Reculer la caméra
PLUS	Avancer la caméra
DROITE	Déplacer la bestiole vers +X
GAUCHE	Déplacer la bestiole vers -X
PAGEPREC	Déplacer la bestiole vers +Y
PAGESUIV	Déplacer la bestiole vers -Y
BAS	Déplacer la bestiole vers +Z
HAUT	Déplacer la bestiole vers -Z
f	Diminuer la taille du corps
r	Augmenter la taille du corps
VIRGULE	Tourner la bestiole dans le sens anti-horaire
POINT	Tourner la bestiole dans le sens horaire
o	Diminuer l'angle des pattes
p	Augmenter l'angle des pattes
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 Figures supplémentaires

Allez voir la théière bien connue en infographie sur Internet :

[http://www.sjbaker.org/wiki/?title=The\\_History\\_of\\_The\\_Teapot](http://www.sjbaker.org/wiki/?title=The_History_of_The_Teapot)

[http://en.wikipedia.org/wiki/Utah\\_teapot](http://en.wikipedia.org/wiki/Utah_teapot).

FIGURE 5 – La théière utilisée dans *Toy Story*FIGURE 6 – La théière utilisée dans l'épisode *Treehouse of Horror VI*



FIGURE 7 – La théière utilisée dans un écran de veille (Windows)

## C Apprentissage supplémentaire

1. Quel est le nombre minimal de `PushMatrix()`/`PopMatrix()` à utiliser ? Pourquoi faut-il éviter d'en ajouter inutilement ?
2. Allonger les pattes selon la taille du corps.
3. Ajouter des ailes supplémentaires à la bestiole.
4. Utiliser un octaèdre régulier au lieu d'un cylindre.
5. Utiliser un cube au lieu d'un cylindre.
6. Remplacer la fonction `LookAt()` qui positionne la caméra par une combinaison de `Translate()` et de `Rotate()`. L'affichage doit être le même, peu importe si on utilise l'une ou l'autre version pour définir le point de vue.