

# **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>6</b>  |
| 3.1      | Exigences fonctionnelles . . . . .     | 6         |
| 3.2      | Exigences non fonctionnelles . . . . . | 6         |
| 3.3      | Rapport . . . . .                      | 6         |
| <b>A</b> | <b>Liste des commandes</b>             | <b>7</b>  |
| <b>B</b> | <b>Figures supplémentaires</b>         | <b>7</b>  |
| <b>C</b> | <b>Apprentissage supplémentaire</b>    | <b>10</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

Vous remettrez un fichier zip contenant tout le code source du TP et le rapport (\*.cpp, \*.h, \*.glsl, makefile, \*.txt).

Faites « `make remise` » pour créer l'archive « `remise.zip` ».  
Vous déposerez ensuite ce fichier dans Moodle.

## 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 cubique de taille variable (`tailleCorps`) et quatre pattes de taille fixe (`longPatte` x `largPatte`) formées par des cubes 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 : cube et théière.

La sphère et les cubes sont tracés par des appels aux fonctions fournies (sans modifier ces fonctions). La tête de la bestiole est une sphère positionnée au milieu de l'arête supérieure (en  $X+$  et  $Z+$ ). Chaque patte est composé d'un cube étiré et articulé selon un angle (`anglePatte`) (Figure 2). Enfin, la bestiole peut aussi tourner sur elle-même (`angleBestiole`) et se déplacer (`positionBestiole`) dans l'espace de la boîte (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.h`. Consultez les notes de cours et explorez la classe `MatricePipeline` afin de bien comprendre son fonctionnement.

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

Le corps de la bestiole pourra être représenté par un cube 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 « `teapot_data.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.

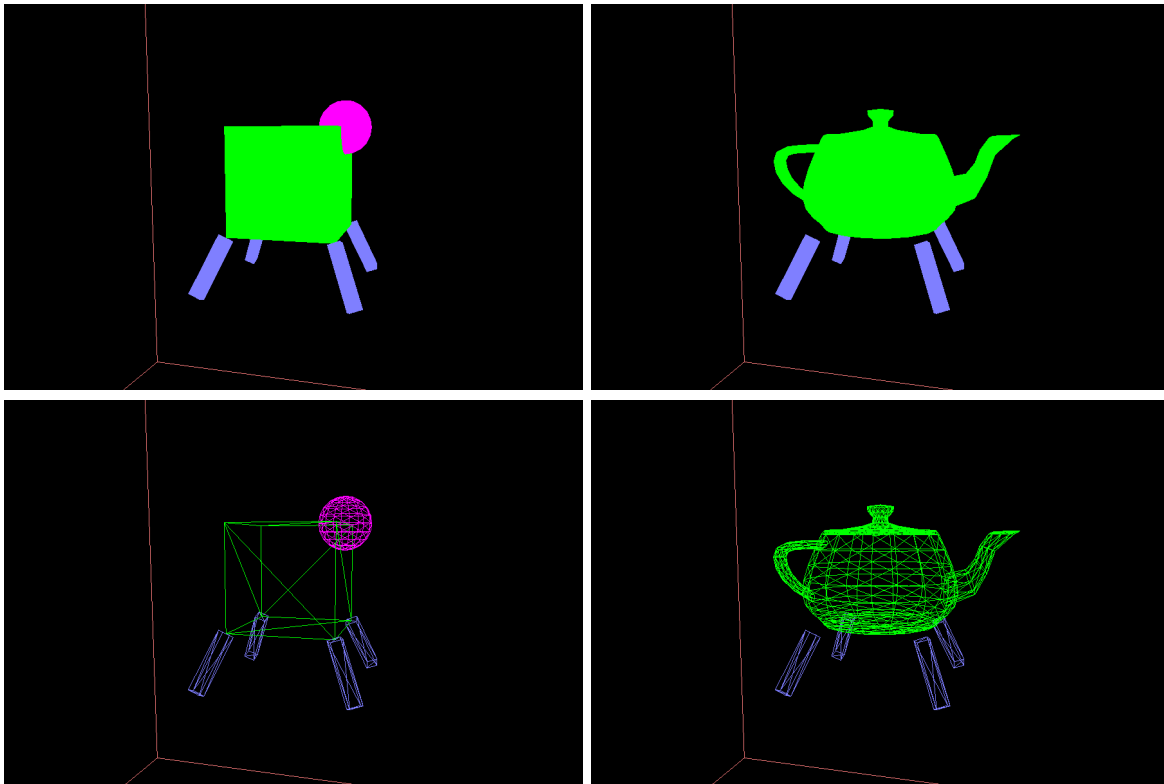


FIGURE 1 – Bestiole sous la forme d'un cube ou d'une théière

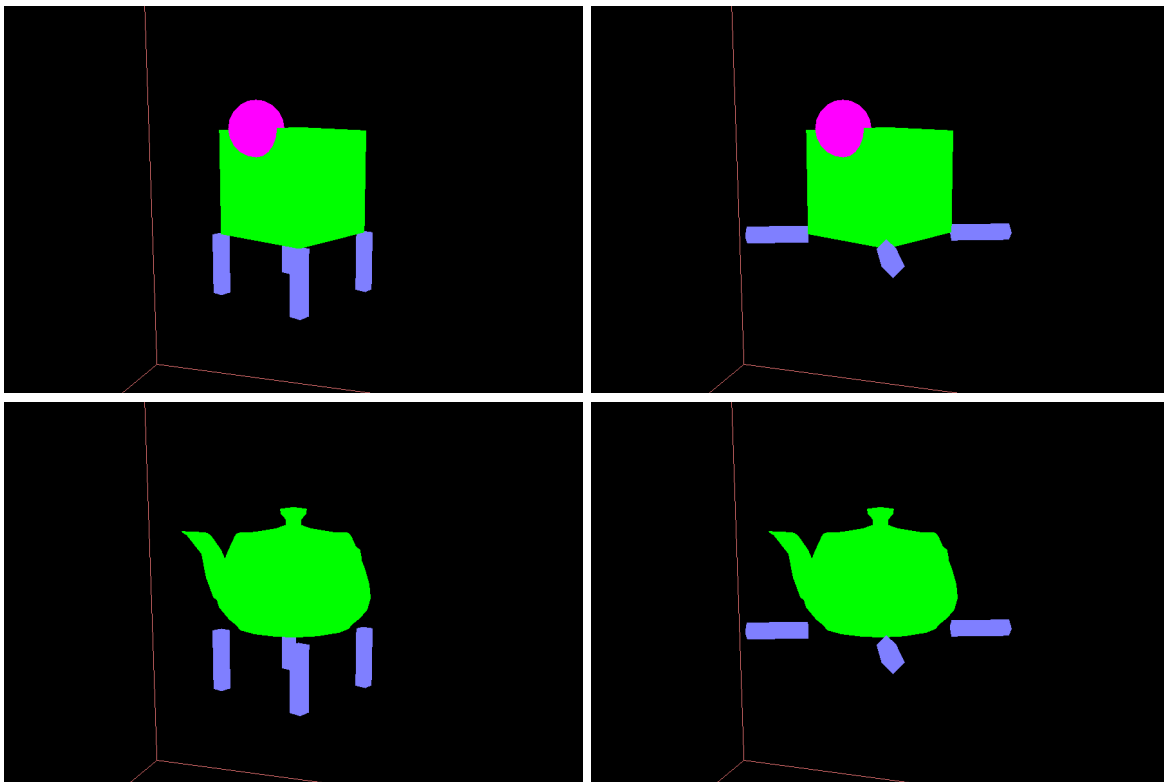


FIGURE 2 – Articulation des pattes

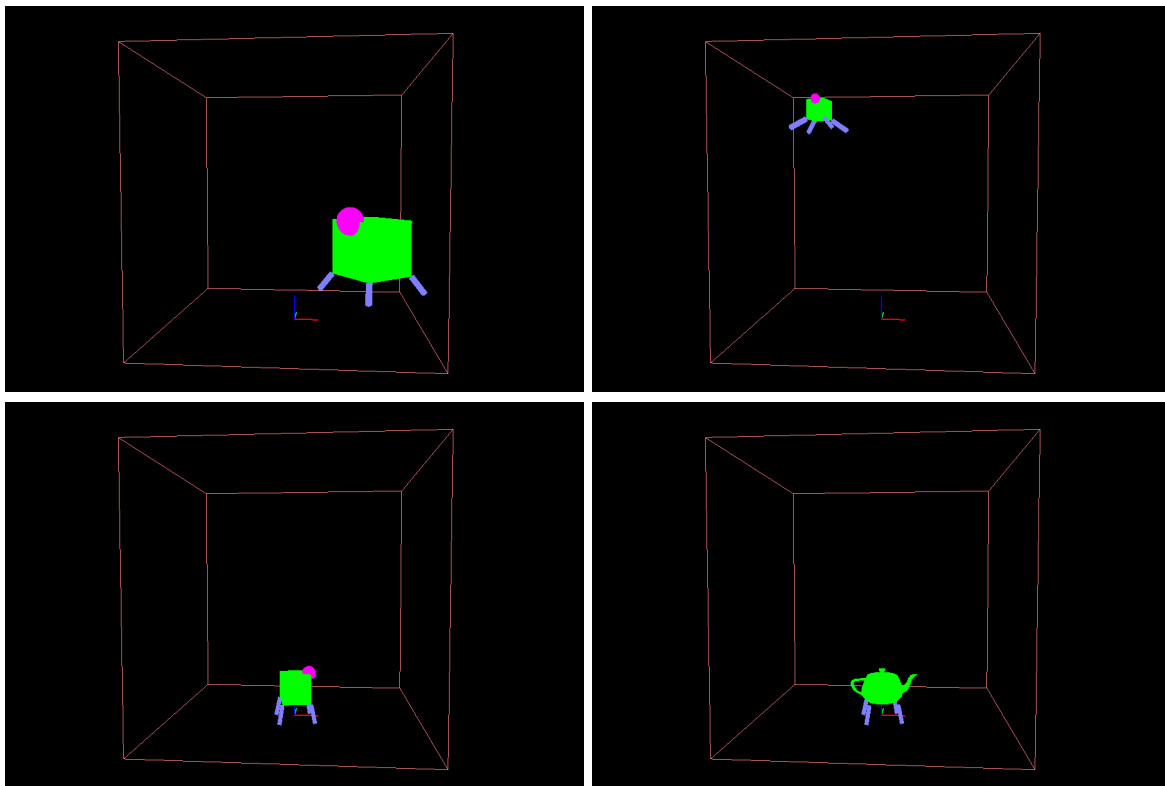


FIGURE 3 – Déplacement, rotation et articulation des pattes en mode animation

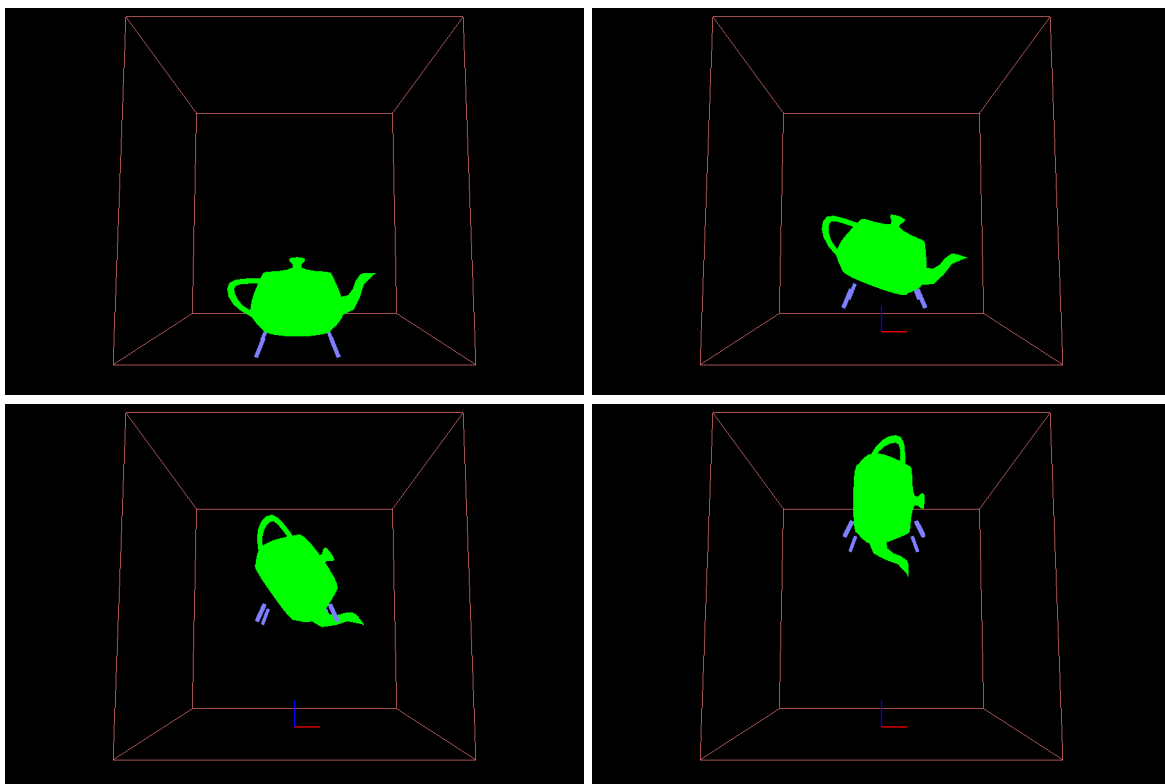


FIGURE 4 – Variation de l'orientation de la théière en fonction de sa hauteur

### 3 Exigences

#### 3.1 Exigences fonctionnelles

Partie 1 :

- E1. Le corps, la tête et la pattes sont dessinés en utilisant les fonctions `afficherCube()` et `afficherSphere()`.
- 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 `positionBestiole` et la taille de son corps est donnée par `tailleBestiole`.
- 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 attachées aux coins du cube et sont de taille `largPatte x largPatte x 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.
- E10. L'orientation de la théière varie linéairement en fonction de sa hauteur tel qu'illustré à la Figure 4.
- E11. (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` et l'inclure 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                      |
| v             | Recharger les fichiers des nuanceurs et recréer le programme |
| i             | Réinitialiser le point de vue                                |
| g             | Permuter l'affichage en fil de fer ou plein                  |
| 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                                 |
| FIN           | Diminuer la taille du corps                                  |
| DEBUT         | Augmenter la taille du corps                                 |
| VIRGULE       | Tourner la bestiole dans le sens anti-horaire                |
| POINT         | Tourner la bestiole dans le sens horaire                     |
| CROCHETGAUCHE | Diminuer l'angle des pattes                                  |
| CROCHETDROIT  | Augmenter l'angle des pattes                                 |
| 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. 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.
2. Quel est le nombre minimal de `PushMatrix()/PopMatrix()` à utiliser ? Pourquoi faut-il éviter d'en ajouter inutilement ?
3. Allonger les pattes selon la taille du corps.
4. Ajouter des ailes à la bestiole et faites-la voler automatiquement.
5. Utiliser un octaèdre régulier au lieu d'un cube.
6. Utiliser une sphère au lieu d'un cube.