



**POLYTECHNIQUE  
MONTRÉAL**

LE GÉNIE  
EN PREMIÈRE CLASSE

# **INF2705 Infographie**

## **Spécification des requis du système**

### **Travail pratique 2**

### ***Le pipeline graphique***

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

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	But . . . . .	3
1.2	Portée . . . . .	3
1.3	Références . . . . .	3
<b>2</b>	<b>Description globale</b>	<b>4</b>
2.1	But . . . . .	4
2.2	Travail demandé . . . . .	4
2.3	Fichiers fournis . . . . .	4
<b>3</b>	<b>Exigences</b>	<b>6</b>
3.1	Exigences fonctionnelles . . . . .	6
3.2	Exigences non fonctionnelles . . . . .	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>10</b>

# 1 Introduction

Ce document décrit les exigences fonctionnelles et non fonctionnelles du TP2 « *Le pipeline graphique* » 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 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 : `glRotate()`, `glTranslate()`, `glScale()`, `glPushMatrix()`, `glPopMatrix()` et `gluLookAt()`.

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 bras articulé et sa théière

On demande de réaliser un programme permettant d'afficher, au centre d'un quadrilatère, un bras articulé composé en utilisant des appels aux fonctions fournies (sans les modifier) pour l'affichage d'un cylindre et d'une sphère. Le bras possédera deux articulations avec chacune un degré de liberté représentant un angle de rotation dans l'axe des Z. Les angles de rotation seront contrôlés interactivement afin de tourner ou plier le bras. Ces mouvements permettront à l'extrémité du bras de pouvoir atteindre n'importe quel point sur le plan horizontal. Une théière est affichée au bout du bras et une rotation supplémentaire permettra de verser le thé avec cette théière. La Figure 1 montre ce bras mécanique et la Figure 2 montre la théière bien connue en infographie (voir aussi annexe B).

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

Le logiciel permettra de manipuler la caméra synthétique en utilisant a) soit la fonction `gluLookAt()` pour la placer dans l'espace, b) soit une combinaison de `glTranslate()` et de `glRotate()`. L'affichage doit être le même, peu importe si on utilise l'une ou l'autre version pour définir le point de vue. La souris contrôlera les deux angles définissant la position de la caméra, tel qu'illustré à la Figure 3.

Enfin, la théière sera affichée en utilisant deux VBOs (sommets et indices). Ceux-ci doivent être créés, initialisés et utilisés lors de l'affichage.

### 2.3 Fichiers fournis

Le code fourni présente une sphère et un cylindre sur un quadrilatère. La sphère et le cylindre serviront à construire le bras. Ils devront être mis à l'échelle avec les facteurs indiqués dans le code.

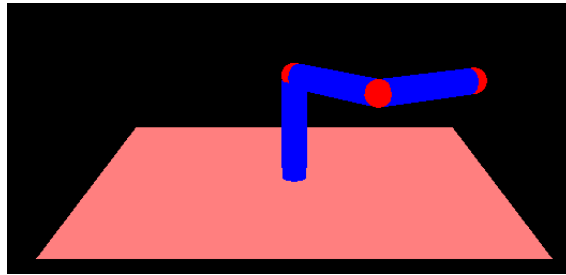


FIGURE 1 – Bras sur un plan

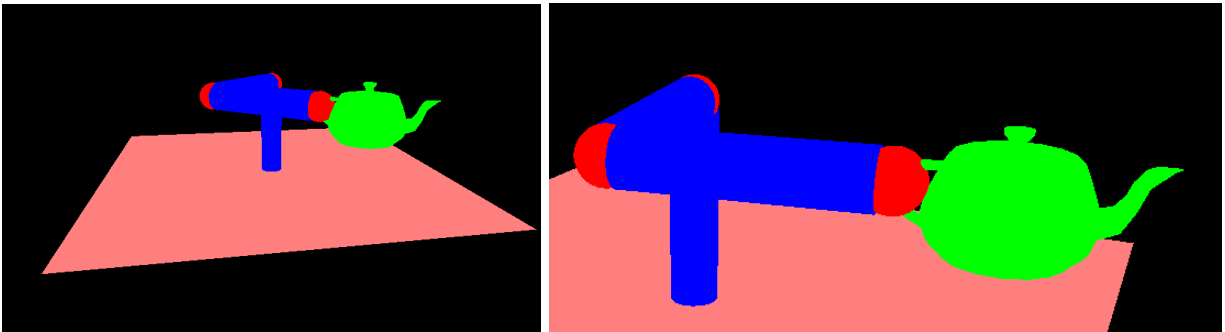
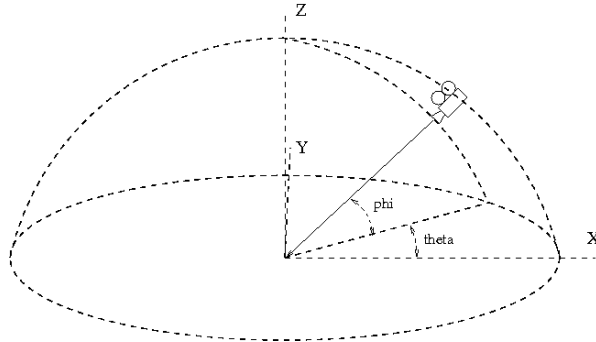


FIGURE 2 – Théière attachée au bras

FIGURE 3 – Angles de déplacement :  $\theta$  et  $\phi$

## 3 Exigences

### 3.1 Exigences fonctionnelles

Partie 1 :

- E1. Les fonctions `glPushMatrix()` / `glPopMatrix()` sont correctement utilisées pour sauvegarder l'état des matrices pour le dessin de chacune des parties du bras. La taille du bras est respectée.
- E2. Les fonctions `glRotate()`, `glTranslate()` et `glScale()` sont correctement utilisées pour les transformations géométriques nécessaires au dessin de chaque partie du bras.
- E3. Les articulations sont dessinées avec des sphères et les bras avec des cylindres en utilisant les fonctions `dessinerCylindre()` et `dessinerSphere()`.
- E4. Les flèches du clavier influencent les angles des articulations et l'orientation de la théière.
- E5. L'angle de rotation de la seconde articulation est limité.
- E6. Le logiciel utilise correctement les touches listées à l'annexe A.

Partie 2 :

- E7. Les fonctions `glGenBuffers()`, `glBindBuffers()`, `glBufferData()` et `glDrawElements()` sont correctement utilisées afin d'utiliser deux VBOs (sommets et indices) pour afficher la théière.
- E8. La théière est placée au bout du bras tel qu'illustré à la Figure 2.
- E9. La caméra est placée de façon à regarder à partir d'un point d'observation  $P_{obs}$  vers la base du bras. Cette position d'observation peut être déplacée autour de la base du bras, selon un système de coordonnées sphériques (azimuth et élévation, ou simplement  $\theta$  et  $\phi$ ). On peut contrôler les angles de rotations et la distance du point  $P_{obs}$  à la base du bras.
- E10. On peut choisir entre deux méthodes peuvent être utilisées pour choisir le point de vue :
  - (a) La fonction `gluLookAt()` est correctement utilisée.
  - (b) Les fonctions `glTranslate()` et `glRotate()` sont correctement utilisées.
- E11. Le mode perspective est correctement utilisé et le rapport d'aspect est respecté lors d'un redimensionnement de la fenêtre.
- E12. Le déplacement de la souris avec le bouton enfoncé influence les angles.
- E13. L'angle d'élévation est limité afin d'empêcher la caméra d'être placée directement au-dessus ou directement en-dessous de la théière.
- E14. Le rapprochement de la caméra à la théière est limité afin de ne pas inverser la projection.
- E15. Le logiciel utilise correctement les touches listées à l'annexe A.

### 3.2 Exigences non fonctionnelles

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

Le dessin peut se faire en ajoutant les fonctions `creerVB0()` et `definirCamera()`. 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
g	Passage fil de fer ou plein
Flèches clavier : gauche, droite	Le bras tourne autour de son axe ( $\phi$ )
Flèches clavier : haut, bas	L'avant-bras se plie ou se déplie ( $\theta$ )
[ / ]	La théière s'incline / remonte (pour verser le thé)
l	(caméra) Basculer l'utilisation de <code>gluLookAt()</code> ou l'utilisation de <code>glTranslate()</code> et <code>glRotate()</code> pour placer la caméra
+ / -	(caméra) Rapprocher / éloigner la caméra de la théière
r	(caméra) Remise à zéro de la caméra
Mouvements souris en X avec bouton enfoncé	(caméra) Rotation de la caméra autour de l'axe des Z
Mouvements souris en Y avec bouton enfoncé	(caméra) Changement de l'angle d'élévation de la caméra
ESC	Arrêt de l'application

### B Figures supplémentaires

Allez voir la théière bien connue en infographie sur Internet :

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

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

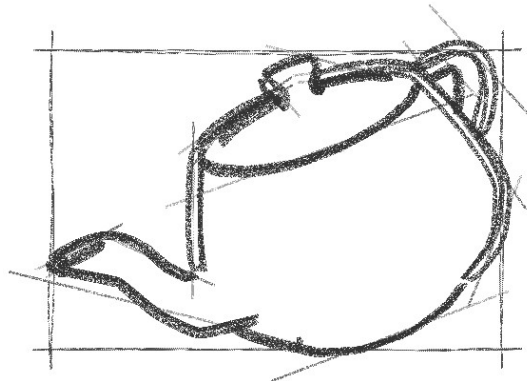


FIGURE 4 – Esquisse



FIGURE 5 – La théière utilisée dans d'autres situations

## C Apprentissage supplémentaire

1. Quel est le nombre minimal de `glPushMatrix()/glPopMatrix()` à utiliser ? Pourquoi faut-il éviter d'en ajouter inutilement ?
2. Ajoutez une rotation pour manipuler le bras en 3D afin d'atteindre des positions en hauteur.
3. Ajoutez un mode automatique pour animer le bras. Utiliser diverses touches du clavier pour contrôler sa vitesse ou son déplacement.
4. Si vous voulez que le bout du bras se rende à une certaine position au-dessus du plan (p.e. pour y saisir quelque chose), comment allez-vous vous y prendre pour déterminer les valeurs appropriées des angles afin de positionner le bras à cet endroit ?
5. Affichez la théière sans utiliser aucun VBO.
6. Une fois la théière affichée, modifiez les VBOs pour afficher autre chose (ex. : un cube).