

INF2705 Infographie

Spécification des requis du système

Travail pratique 2

Le bras articulé

Table des matières

1	Introduction	2
1.1	But	2
1.2	Portée	2
1.3	Références	2
2	Description globale	3
2.1	But	3
2.2	Travail demandé	3
2.3	Fichiers fournis	3
3	Exigences	6
3.1	Exigences fonctionnelles	6
3.2	Exigences non fonctionnelles	6
A	Liste des commandes	7
B	Figures supplémentaires	7
C	Apprentissage supplémentaire	9

1 Introduction

Ce document décrit les exigences fonctionnelles et non fonctionnelles du TP2 « *Le bras articulé* » 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 : `Rotate()`, `Translate()`, `Scale()`, `PushMatrix()`, `PopMatrix()` et `LookAt()`.

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 de cylindres et de sphères mis à l'échelle et tracés par des appels aux fonctions fournies (sans les modifier). Le bras possède plusieurs degrés de libertés (ddl) : il est articulé à l'épaule (2 ddl), il peut tourner au coude (1 ddl) et il peut aussi tourner au poignet (1 ddl). Les angles de rotation seront contrôlés interactivement afin de déplacer le bras. Ces mouvements permettront à l'extrémité du bras d'atteindre n'importe quel point sur le plan horizontal et aussi de s'élever au niveau de l'épaule. Une théière est affichée au bout du bras (au poignet) et la rotation du poignet permettra de verser le thé avec cette théière. Les Figures 1 et 2 montrent ce bras mécanique et sa 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 `LookAt()` pour la placer dans l'espace, b) soit 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. 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, un cylindre et une théière sur un quadrilatère. La sphère et le cylindre serviront à construire le bras. Ils seront mis à l'échelle avec les facteurs indiqués dans le code.

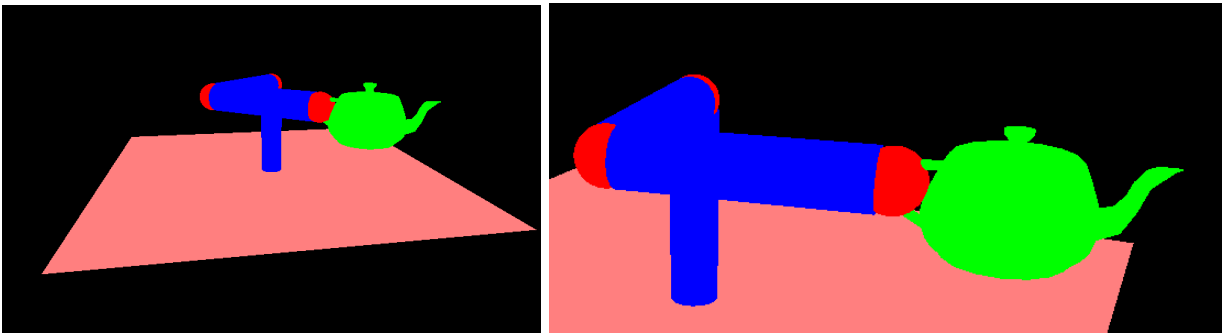


FIGURE 1 – Rotations horizontales du bras

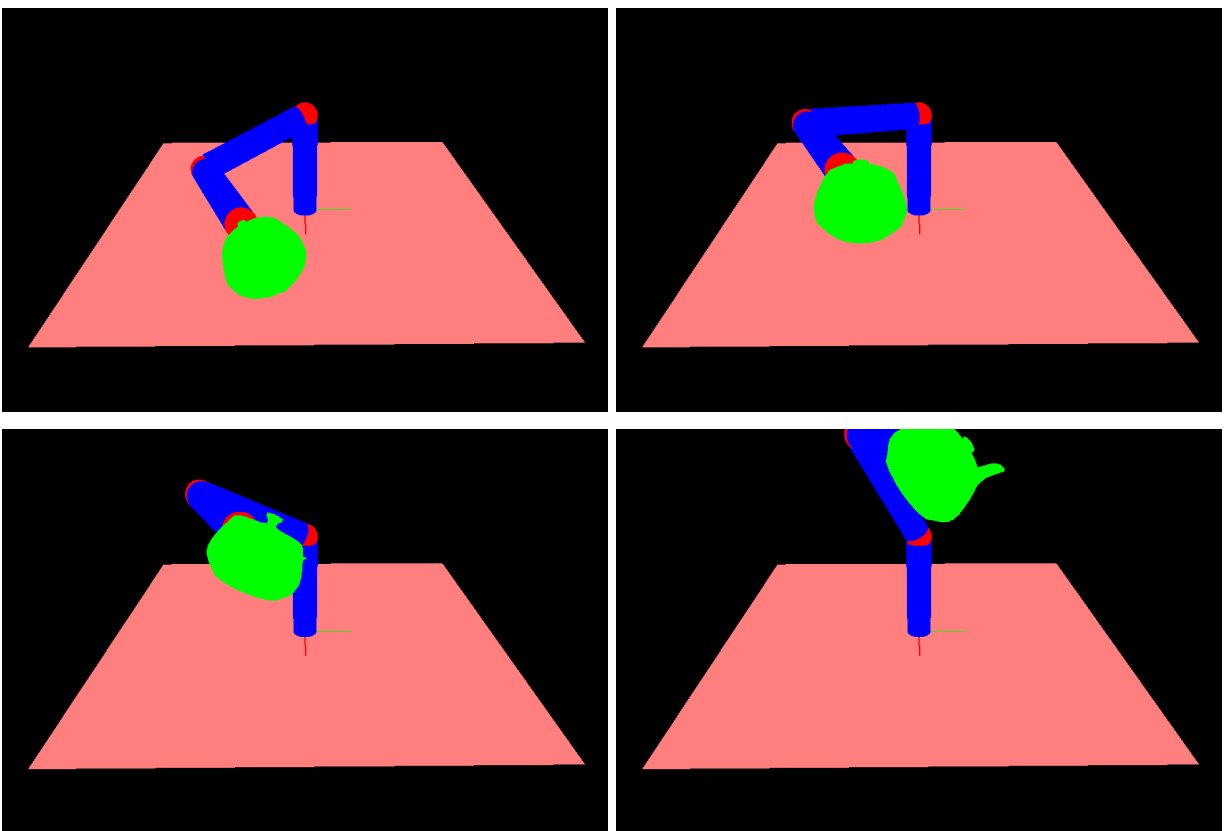


FIGURE 2 – Articulation en élévation à l'épaule

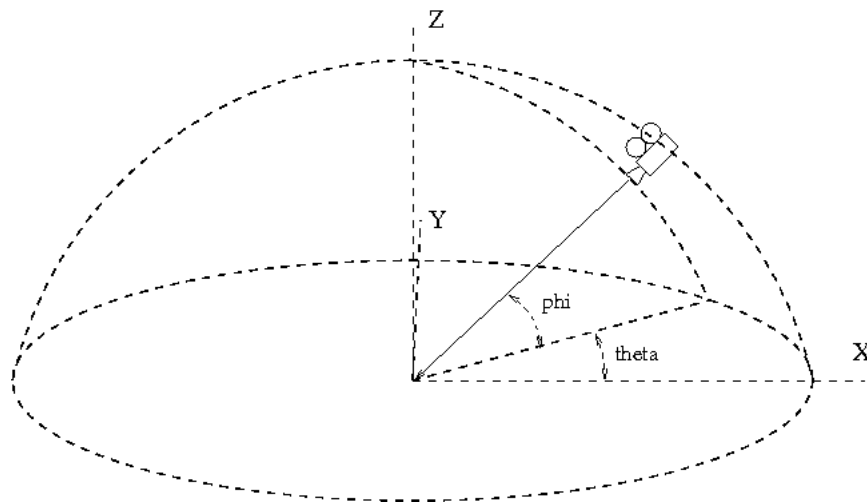


FIGURE 3 – Angles de déplacement de la caméra : θ_{Cam} et ϕ_{Cam}

3 Exigences

3.1 Exigences fonctionnelles

Partie 1 :

- E1. Les articulations sont dessinées avec des sphères et les bras avec des cylindres en utilisant les fonctions `afficherCylindre()` et `afficherSphere()`.
- E2. Les fonctions `Rotate()`, `Translate()` et `Scale()` sont correctement utilisées pour les transformations géométriques nécessaires au dessin de chaque partie du bras.
- E3. Les fonctions `PushMatrix()` et `PopMatrix()` sont correctement utilisées pour sauvegarder l'état des matrices pour le dessin de chaque partie du bras. La taille du bras est respectée.
- E4. La théière est placée au bout du bras tel qu'illustré à la Figure 1.
- E5. Les flèches du clavier influencent correctement les angles des articulations et l'orientation de la théière.
- E6. L'angle de rotation de l'élévation du bras, de même que l'angle de rotation du coude sont limités.
- E7. Le logiciel utilise correctement les touches listées à l'annexe A.

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.
- 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 : *thetaCam* et *phiCam*). On peut contrôler les angles de rotations et la distance de la caméra à la base du bras.
- E10. On peut choisir entre deux méthodes peuvent être utilisées pour choisir le point de vue :
 - (a) La fonction `LookAt()` est correctement utilisée.
 - (b) Les fonctions `Translate()` et `Rotate()` sont correctement utilisées.
- E11. Le déplacement de la souris avec le bouton enfoncé influence les angles.
- E12. L'angle d'élévation est limité afin d'empêcher la caméra d'être placée directement au-dessus ou directement en dessous de la théière.
- E13. Le rapprochement de la caméra à la théière est limité afin de ne pas inverser la projection.
- E14. Le logiciel utilise correctement les touches listées à l'annexe A.

3.2 Exigences non fonctionnelles

Pour la partie 1, des modifications sont principalement à faire dans les fonctions `afficherScene()` et `afficherBras()`. Pour la partie 2, le dessin peut se faire en ajoutant les fonctions `creerVBO()` et `definirCamera()`.

ANNEXES

A Liste des commandes

Touche	Description
g	Passage fil de fer ou plein
PageSuiv ou PagePrec	Le bras tourne en élévation (<code>angleElevation</code>)
Flèches clavier : gauche, droite	Le bras tourne autour de son axe (<code>angleEpaule</code>)
Flèches clavier : haut, bas	L'avant-bras se plie ou se déplie (<code>angleCoude</code>)
[ou]	La théière s'incline ou remonte pour verser le thé (<code>anglePoignet</code>)
l	(caméra) Basculer l'utilisation de <code>LookAt()</code> ou l'utilisation de <code>Translate()</code> et <code>Rotate()</code> pour placer la caméra
+ ou -	(caméra) Rapprocher ou éloigner la caméra de la base du bras
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://en.wikipedia.org/wiki/Utah_teapot.

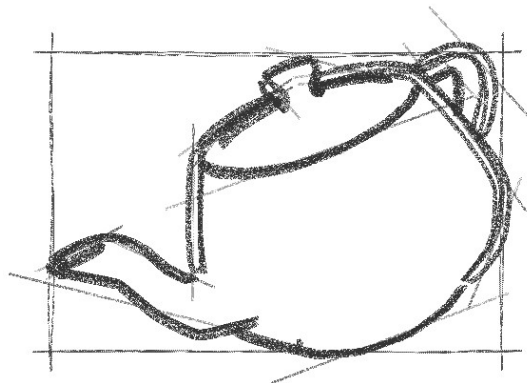


FIGURE 4 – Esquisse

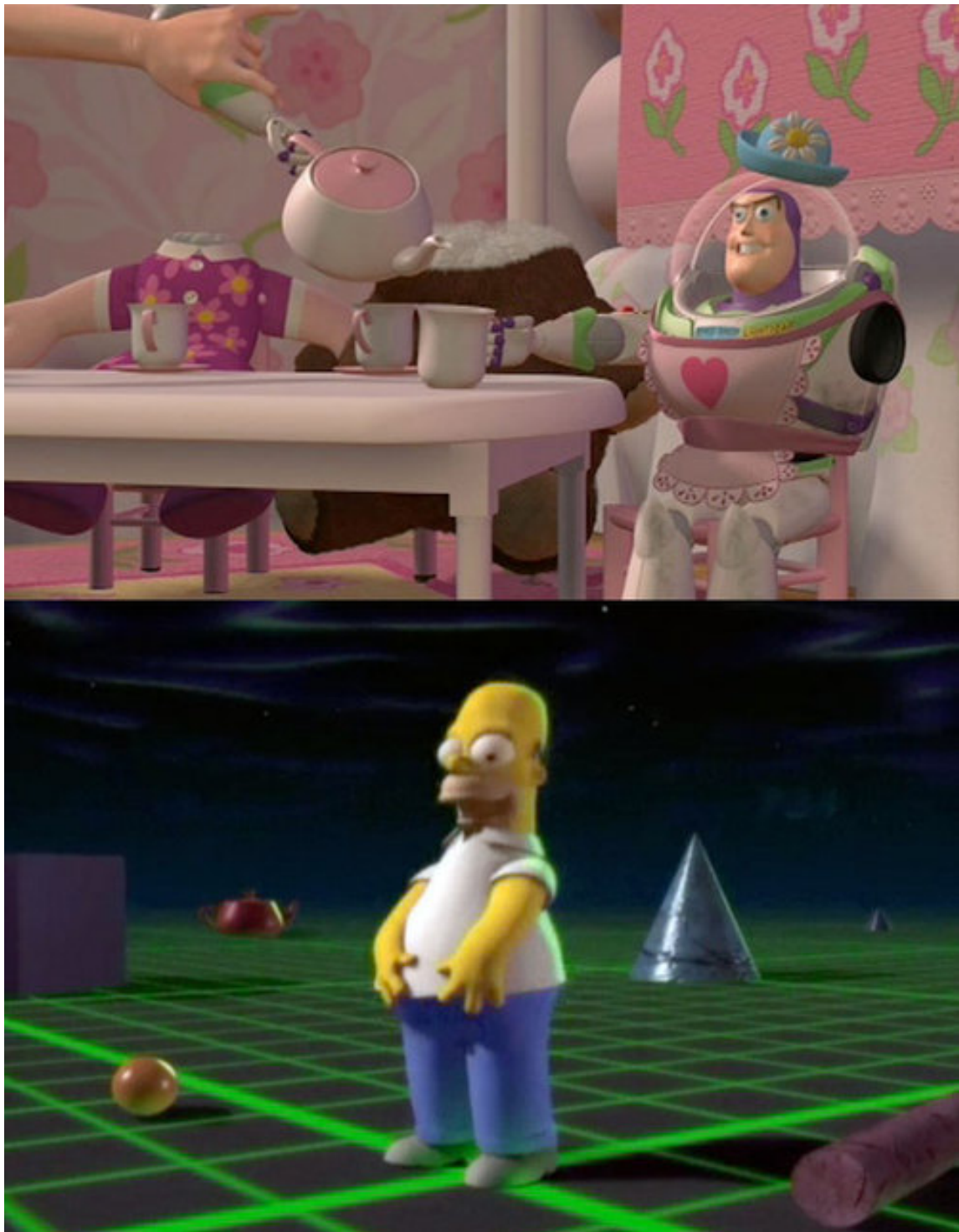


FIGURE 5 – La théière utilisée dans d'autres situations

C Apprentissage supplémentaire

1. 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 ? (C'est ce qu'on nomme la « cinématique inverse ».)
2. Ajoutez un mode automatique pour animer le bras. Utiliser diverses touches du clavier pour contrôler sa vitesse ou son déplacement.
3. Affichez la théière sans utiliser aucun VBO.
4. Une fois la théière affichée, ajoutez ou modifiez les VBOs existants pour afficher autre chose (ex. : un cube, une sphère).
5. Quel est le nombre minimal de `PushMatrix()`/`PopMatrix()` à utiliser ? Pourquoi faut-il éviter d'en ajouter inutilement ?