POLYTECHNIQUE MONTRÉAL

LE GÉNIE EN PREMIÈRE CLASSE

INF2705 Infographie

Spécification des requis du système

Travail pratique 1

Repères, transformations et VBO

Département de génie informatique et génie logiciel
École polytechnique de Montréal
Automne 2021

Table des matières

1 Introduction

Ce document décrit les exigences du TP1 « *Repères, transformations et VBO* » (Automne 2021) 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` » ou exécutez/cliquez sur « `remise.bat` » afin de créer l'archive « **INF2705_remise_TPn.zip** » (ou .7z, .rar, .tar) 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 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 pour la modification des matrices d'OpenGL : `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 : l'hélicoptère

On demande de réaliser un programme permettant d'afficher un hélicoptère, semblable à Ingenuity [1] [2] qui se promène actuellement sur Mars (figure 1).

Notre hélicoptère se déplace dans un espace limité et tourne sur lui-même pour regarder dans diverses directions (figure 2). L'hélicoptère a un corps cubique (légèrement étiré) et il possède quatre pattes attachées en haut de son corps. Les pales de l'hélicoptère, situées sur un mat au dessus du corps, tournent en sens opposées l'une de l'autre (rotor contrarotatif). Les mouvements des pales et pattes sont illustrés à la figure 3.

Dans la deuxième partie de ce TP, vous ajouterez la possibilité pour cet hélicoptère de devenir LA théière bien connue en infographie. La figure 4 montre le corps de l'hélicoptère sous ses deux formes : cube ou théière.

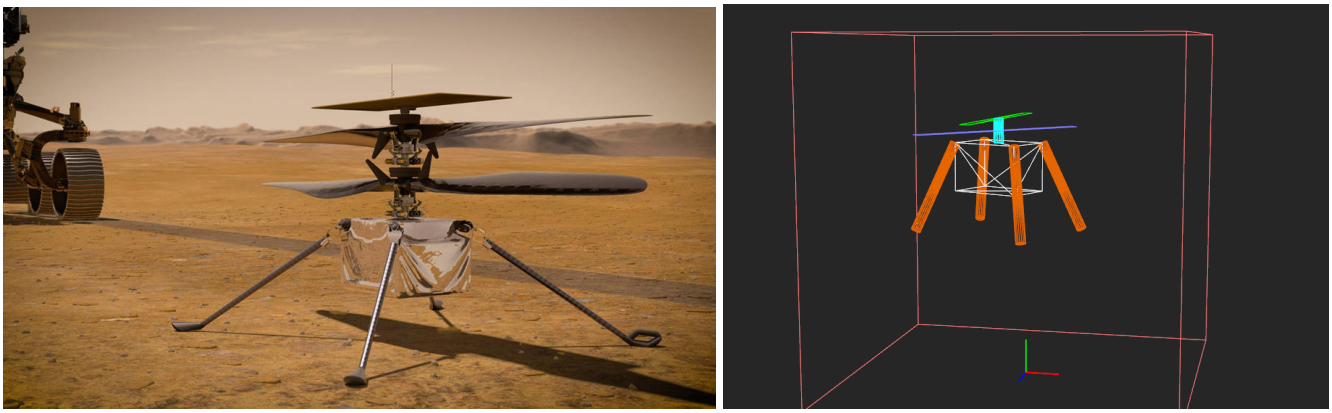

FIGURE 1 – L'original (Ingenuity) et notre modèle

1. `https://www.nasa.gov/feature/jpl/6-things-to-know-about-nasas-ingenuity-mars-helicopter`
2. `https://www.popularmechanics.com/space/moon-mars/a35353442/ingenuity-mars-helicopter/`

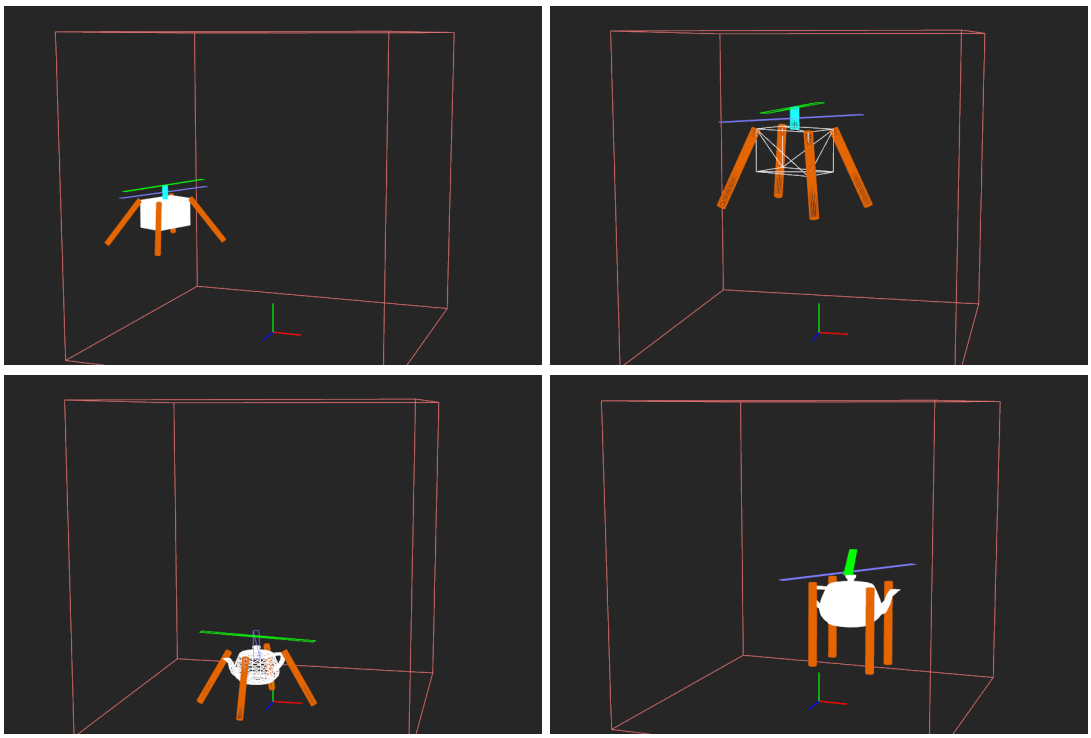

FIGURE 2 – Déplacement, rotation et articulation des pattes et des pales en mode animation

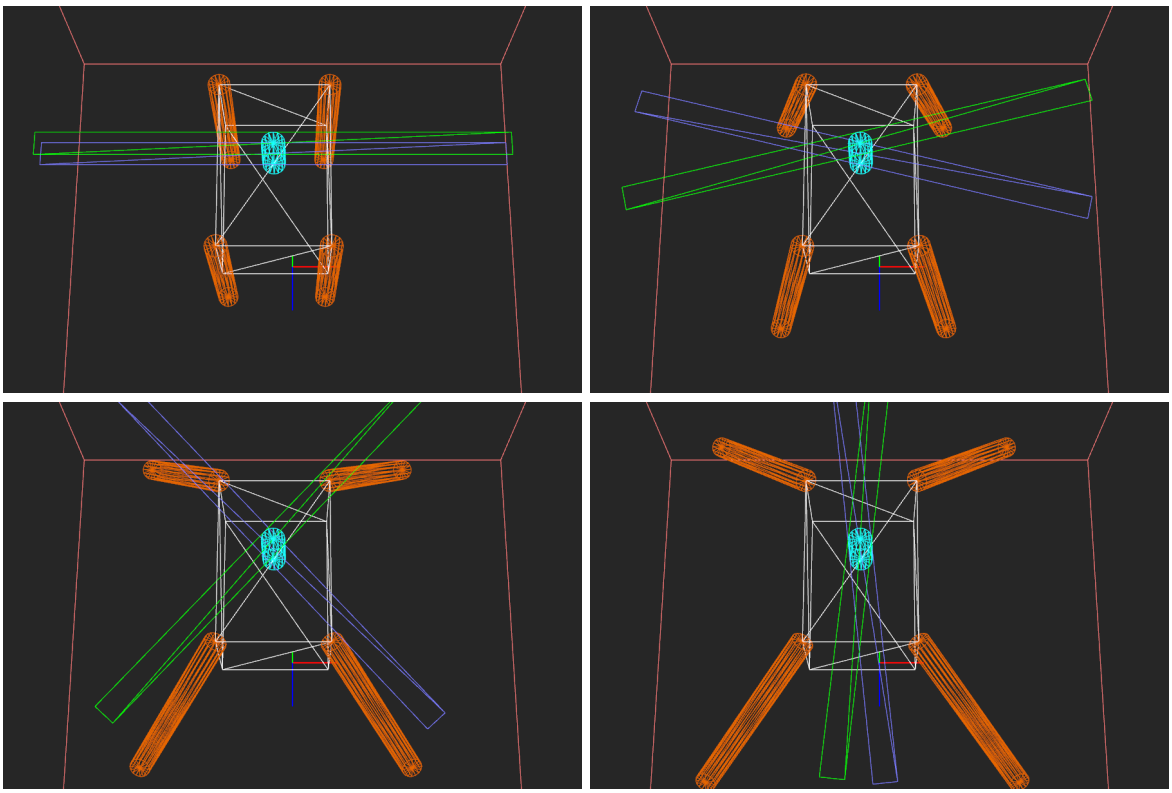

FIGURE 3 – Mouvements des pales et des pattes

L'hélicoptère est construit ainsi :

- est placé à la position courante « `position[]` »,
- est orienté selon l'angle « `angleCorps` » et
- est de taille : (taille, taille, taille),

- Son corps est un cube légèrement étiré en z (avant-arrière) : facteurs d'échelle = (1.0,1.0,1.5). Pour la partie 2, la taille de la théière sera simplement modulée par (0.25,0.25,0.25).
- Le bas du cube corps est à l'origine.
- Au-dessus du corps, un mat vertical est représenté par un cylindre vertical de taille (0.1,0.1,0.5)

- Une pale est un quadrilatère de longueur « 4.0 » et de largeur « 0.2 ».
  - La première pale est à hauteur de 0.25 au-dessus de la face supérieure du corps et tourne dans le sens horaire lorsque vue du dessus.
  - La seconde pale est à hauteur de 0.50 au-dessus de la face supérieure du corps et tourne dans le sens antihoraire.
- L'angle de rotation des pales est donné par la variable « `anglePale` ».

- Les pattes sont des cylindres de largeur « `largMembre` » et longueur « `longMembre` ».
- Les pattes sont attachées en haut de la boite du corps de l'hélicoptère à « `anglePatte` » degrés d'écart.

Le cylindre, la sphère, les parallélépipèdes et les quadrilatères sont tracés par des appels aux fonctions fournies (sans modifier ces fonctions). Les valeurs des variables sont contrôlées interactivement.

La couleur de chaque partie de l'hélicoptère est donnée dans le tableau ci-dessous :

Corps	blanc	(1.0, 1.0, 1.0)
Mat	cyan	(0.0, 1.0, 1.0)
Pale 1	violet	(0.5, 0.5, 1.0)
Pale 2	vert	(0.0, 1.0, 0.0)
Pattes	marron	(0.9, 0.4, 0.0)

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

On pourra représenter le corps de l'hélicoptère 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 « `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 :
`https://gitlab.com/ozell/inf2705-exemples/-/tree/master/03-Base/src/main.cpp`,
`https://gitlab.com/ozell/inf2705-exemples/-/tree/master/03-Base/`.

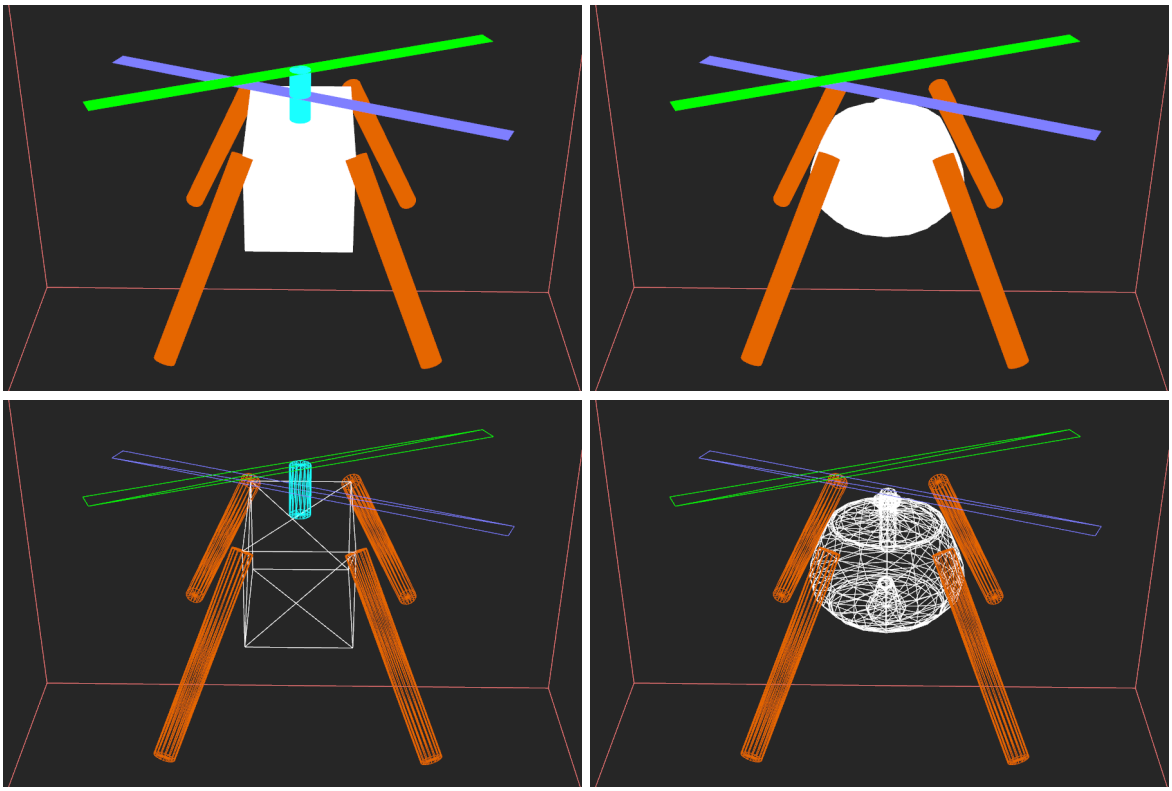

FIGURE 4 – Le corps de l'hélicoptère sous la forme d'un cube ou d'une théière

3 Exigences

3.1 Exigences fonctionnelles

Partie 1 :

E1. Toutes les parties de l'hélicoptère sont dessinées en utilisant les fonctions déjà définies : `afficherCylindre()`, `afficherQuad()` et `afficherCube()`. [2 pts]

E2. Les fonctions `Rotate()`, `Translate()` et `Scale()` sont correctement utilisées pour les transformations géométriques nécessaires au dessin de chaque partie de l'hélicoptère. [1 pt]

E3. L'hélicoptère est positionné selon `position`. La taille du modèle est donné par `taille`. [2 pts]

E4. L'orientation de l'hélicoptère suit bien `angleCorps`. [2 pts]

E5. Les pattes de l'hélicoptère sont bien positionnés aux coins supérieurs du corps et chaque membre est de rayon `largMembre` et de longueur `longMembre`, et orientées selon `anglePatte`. [2 pts]

E6. Les pales de l'hélicoptère sont bien positionnées au-dessus du corps et tournent selon `anglePale`. [2 pts]

E7. Les fonctions `PushMatrix()` et `PopMatrix()` sont correctement utilisées pour sauvegarder l'état des matrices. [2 pts]

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 une seule fois à l'initialisation et réutilisés à chaque affichage. [3 pts]

E9. Le corps de l'hélicoptère peut être affichée en utilisant cette théière telle qu'illustrée à la figure 4. [2 pts]

E10. (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 pts]

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()`.

ANNEXES

A Liste des commandes

Touche	Description
`q`	Quitter l'application
`x`	Activer/désactiver l'affichage des axes
`i`	Réinitiliaser le point de vue et l'hélicoptère
`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 l'hélicoptère vers +X
`GAUCHE`	Déplacer l'hélicoptère vers -X
`HAUT`	Déplacer l'hélicoptère vers +Y
`BAS`	Déplacer l'hélicoptère vers -Y
`PAGEPREC`	Déplacer l'hélicoptère vers +Z
`PAGESUIV`	Déplacer l'hélicoptère vers -Z
`f`	Diminuer la taille de la tête
`r`	Augmenter la taille de la tête
`VIRGULE`	Diminuer l'angle de rotation de l'hélicoptère
`POINT`	Augmenter l'angle de rotation de l'hélicoptère
`o`	Diminuer l'angle des pattes
`p`	Augmenter l'angle des pattes
`j`	Diminuer l'angle des pales
`u`	Augmenter l'angle des pales
`b`	Incrémenter la dimension de la boite
`h`	Décrémenter la dimension de la boite
`ESPACE`	Mettre en pause ou reprendre l'animation
`s`	Sauvegarder une copie de la fenêtre dans un fichier
`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 pattes différemment de la taille du corps.
3. Utiliser une sphère au lieu d'un cube pour le corps.
4. Ajouter des pales supplémentaires à l'hélicoptère.
5. Utiliser un cube au lieu d'un cylindre pour les pattes.
6. 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()`.
7. 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.
8. Si vous voulez que l'hélicoptère 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 le corps 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 l'hélicoptère ne sorte pas de sa terrain de jeu !)

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 5 – LA théière dans un écran de veille (Windows 95/98) `http://www.eeggs.com/items/2422.html`

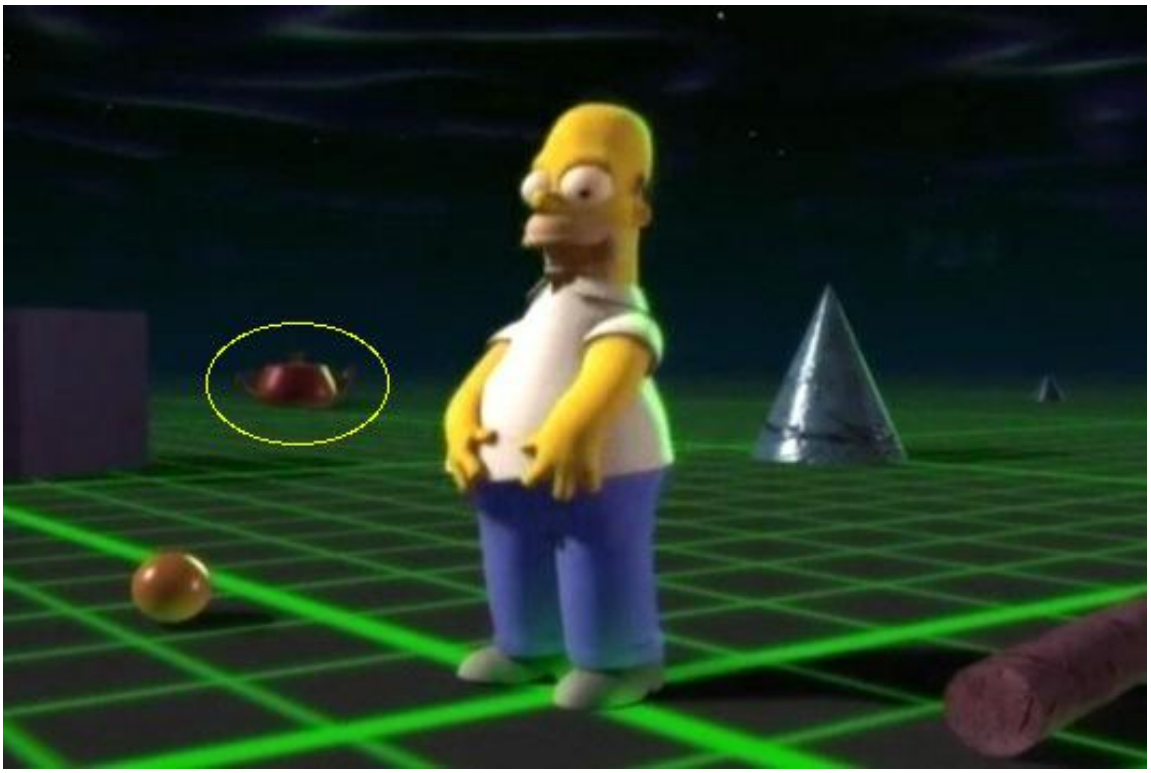

FIGURE 6 – Homer Simpson a vu LA théière dans *Treehouse of Horror VI*

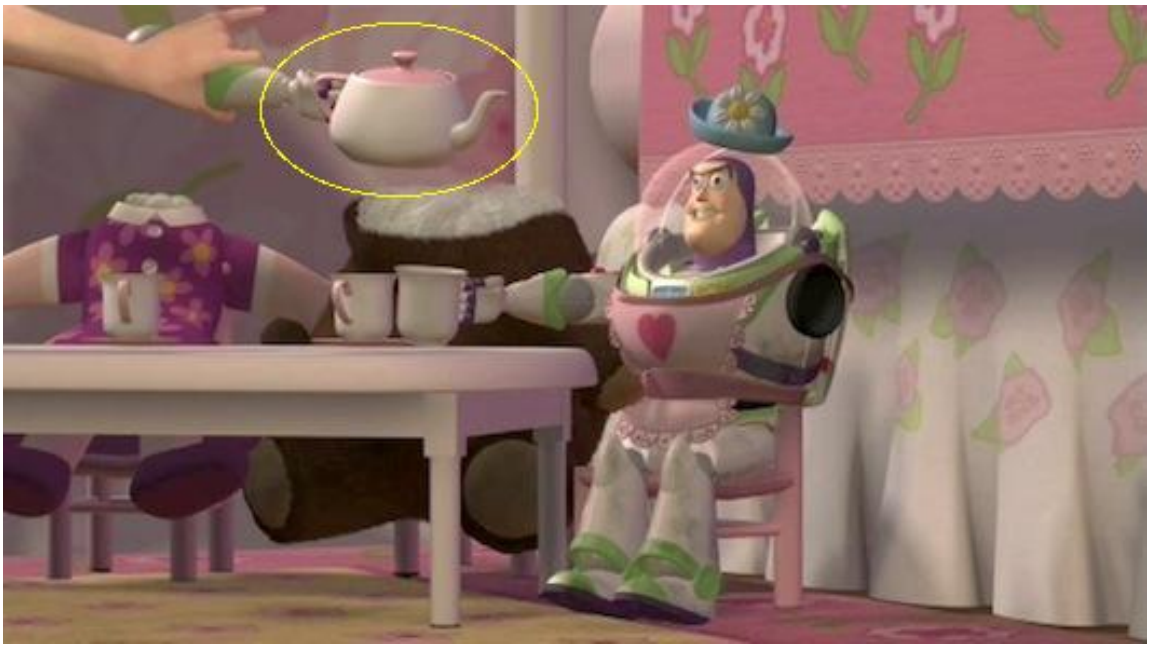

FIGURE 7 – LA théière utilisée dans *Toy Story*

FIGURE 8 – LA théière chez moi ! :)