

# **INF2705 Infographie**

## **Spécification des requis du système**

### **Travail pratique 4**

### ***Illumination et textures***

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	But . . . . .	2
1.2	Portée . . . . .	2
1.3	Références . . . . .	2
<b>2</b>	<b>Description globale</b>	<b>3</b>
2.1	But . . . . .	3
2.2	Travail demandé . . . . .	3
2.3	Fichiers fournis . . . . .	4
<b>3</b>	<b>Exigences</b>	<b>6</b>
3.1	Exigences fonctionnelles . . . . .	6
3.2	Exigences non fonctionnelles . . . . .	6
<b>A</b>	<b>Liste des commandes</b>	<b>7</b>
<b>B</b>	<b>Figures supplémentaires</b>	<b>8</b>
<b>C</b>	<b>Apprentissage supplémentaire</b>	<b>8</b>
<b>D</b>	<b>Formules utilisées</b>	<b>9</b>
D.1	Modèles de spot d'OpenGL et de Direct3D . . . . .	9
D.2	Modèles de réflexion spéculaire de Phong et de Blinn . . . . .	9

# 1 Introduction

Ce document décrit les exigences fonctionnelles et non fonctionnelles du TP4 « *Illumination et textures* » 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 ce TP est de permettre à l'étudiant mettre en pratique les notions d'illumination et d'applications de textures en utilisant des nuanceurs en GLSL.

### 2.2 Travail demandé

#### Partie 1 : l'illumination des objets

On demande d'implanter les modèles d'illumination de Gouraud et de Phong dans les nuanceurs afin d'obtenir un meilleur rendu des objets (cube, théière, etc.). De plus, dans le calcul de la réflexion spéculaire, on pourra choisir entre le modèle de Phong ou celui de Blinn.

Le programme permettra aussi d'utiliser un éclairage de type « spot » selon la définition d'OpenGL ou celle de Direct3D (voir annexe D pour les formules). Les propriétés du spot, sa position, son angle maximum (`spotCutoff`) et son exposant (`spotExponent`) sont modifiables au cours de l'exécution. En tournant l'objet sur lui-même, on pourra observer l'effet de l'illumination par rapport à l'orientation de l'objet.

#### Partie 2 : l'application de textures

Le logiciel permettra d'afficher ou non des textures sur les deux objets illuminés.

- Affichage d'un dé à jouer sur le cube. On spécifiera les coordonnées de texture afin de montrer sur le cube un dé à jouer. La texture contenant toutes les faces du cube est fournie et elle sera utilisée sans la subdiviser en 6 textures différentes. (voir Figure 3).
- Affichage d'un patron d'échiquier sur le cube. On spécifiera les coordonnées de texture afin de montrer sur le cube un échiquier centré sur chaque face ou répété selon l'une ou l'autre des directions. Il doit aussi être possible de changer l'allure de l'affichage, au cours de l'exécution, suivant les rendus de la Figure 4 sans modifier la texture.

De plus, on pourra choisir que les pixels noirs soient transparents plutôt que noirs (voir l'image de droite dans la Figure 3).

## 2.3 Fichiers fournis

Le code fourni présente un cube avec aucun éclairage. Le code pour initialiser, charger et lancer les nuanceurs est aussi fourni. Pour démarrer, on pourra aussi utiliser les nuanceurs de l'exemple du cours : [www.groupe.polymtl.ca/inf2705/exemples/nuanceurs/exempleIllumination/](http://www.groupe.polymtl.ca/inf2705/exemples/nuanceurs/exempleIllumination/)

Deux fichiers de texture (le dé et l'échiquier) sont aussi fournis (Figure 6), de même que les fonctions pour charger les textures en mémoire.

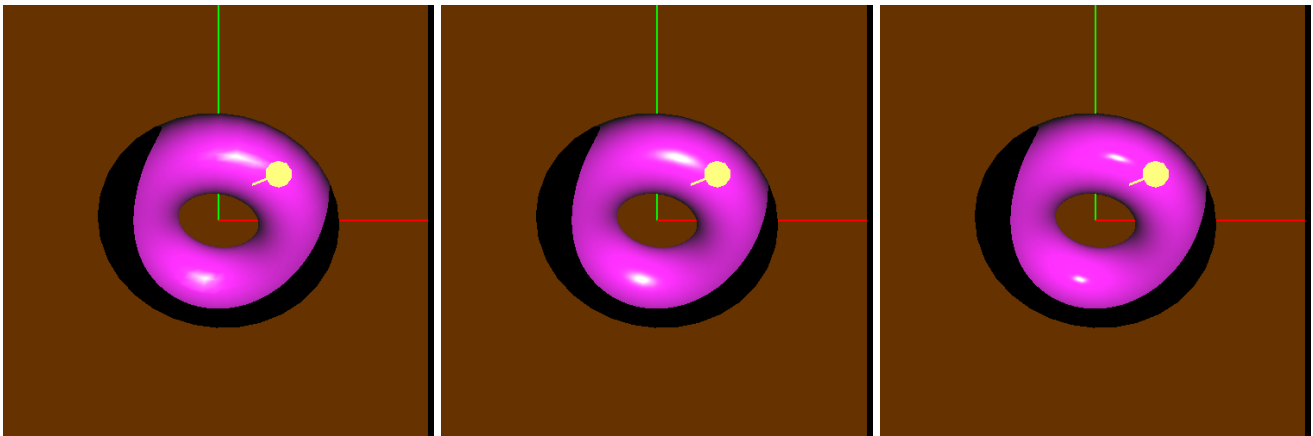


FIGURE 1 – Rendu avec i) illumination de Gouraud et réflexion spéculaire de Phong, ii) illumination de Phong et réflexion spéculaire de Blinn, ii) illumination de Phong et réflexion spéculaire de Phong

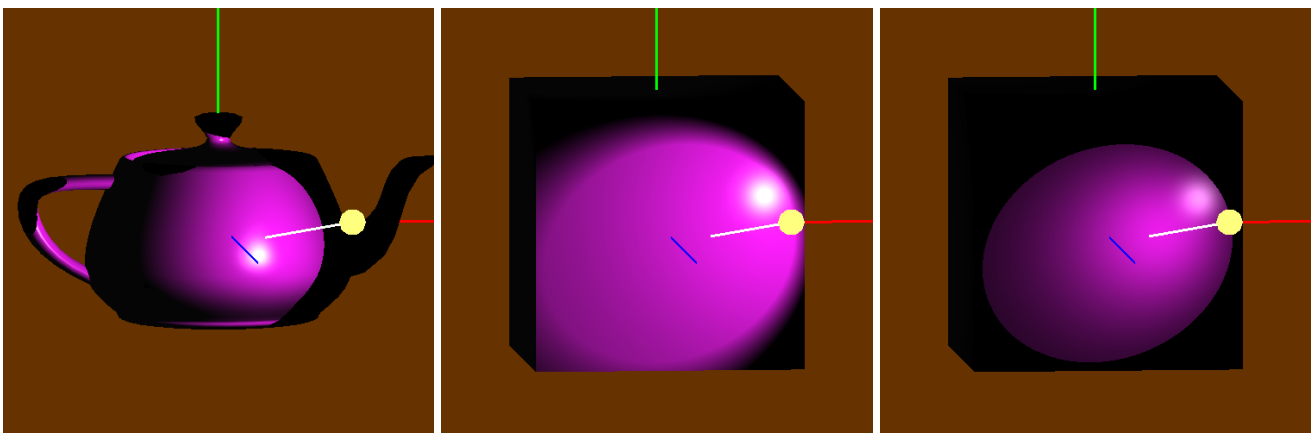


FIGURE 2 – Illumination i) de la théière, ii) avec le modèle Direct3D, ii) avec un plus grand exposant

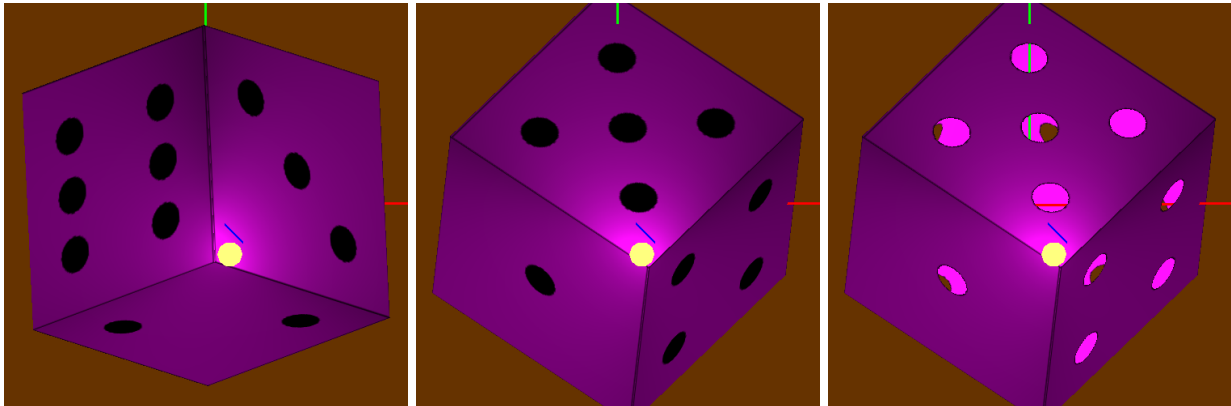


FIGURE 3 – Texture appliquée sur le dé en 3D (noir opaque ou noir transparent)

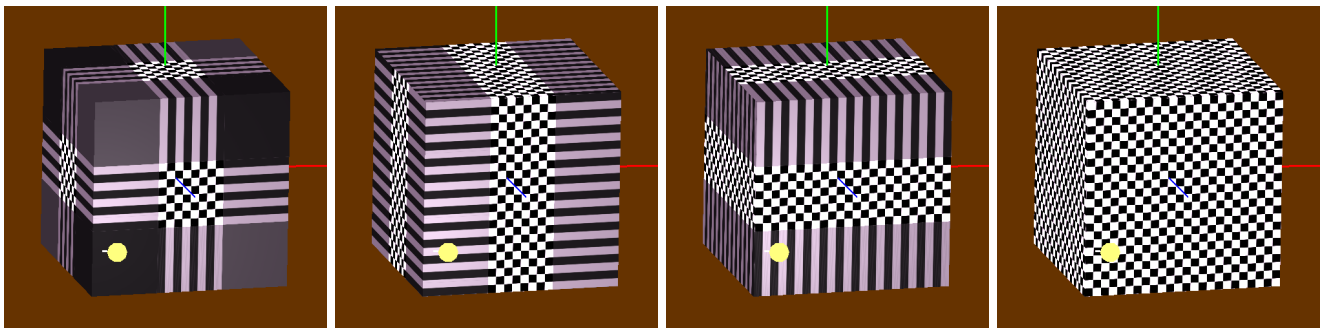


FIGURE 4 – Texture échiquier appliquée sur le cube avec différents modes de répétition

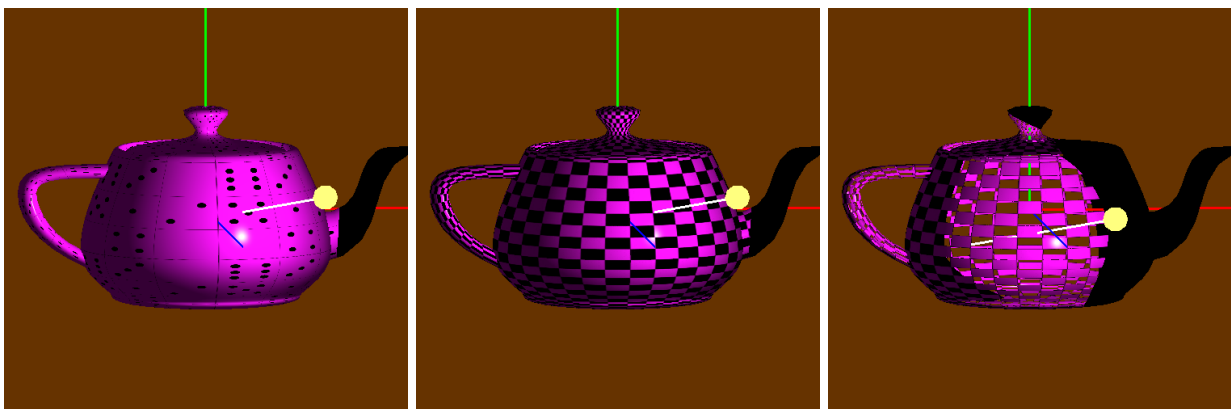


FIGURE 5 – La théière texturée (noir opaque ou noir transparent)

## 3 Exigences

### 3.1 Exigences fonctionnelles

Partie 1 :

- E1. Le modèle d'illumination de Gouraud est correctement implanté dans le nuanceur de sommets.
- E2. Le modèle d'illumination de Phong est correctement implanté dans le nuanceur de fragments.
- E3. Les modèles de réflexion de Phong et de Blinn sont correctement implémentés (voir Figure 1).
- E4. Les modèles de spot d'OpenGL et de Direct3D sont correctement implémentés (voir Figure 2).
- E5. Les modifications des propriétés du spot sont visibles.
- E6. Le logiciel utilise correctement les touches listées à l'annexe A.

Partie 2 :

- E7. Les paramètres des textures sont bien initialisés.
- E8. Le cube est affichée correctement avec la texture du dé (voir Figure 3).
- E9. Le cube est affichée correctement avec la texture de l'échiquier (voir Figure 4).
- E10. L'affichage de l'échiquier peut varier selon les quatre rendus illustrés à la Figure 4.
- E11. L'utilisateur peut changer de texture (entre dé et échiquier) ainsi que de mode de répétition de la texture.
- E12. Permettre l'affichage de l'objet texturé avec les pixels noirs devenus transparents lorsqu'on utilise les nuanceurs (ex. : Figures 3 et 5).

### 3.2 Exigences non fonctionnelles

La plupart des modifications seront faites dans les nuanceurs. La totalité du dessin peut se faire dans la fonction `afficherScene()` et dans les nuanceurs. 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

<b>Touche</b>	<b>Description</b>
Mouvements souris et bouton 1 enfoncé	Tourner l'objet
Mouvements souris et bouton 2 enfoncé	Modifier l'orientation du spot
Mouvements souris et bouton 3 enfoncé	Déplacer la lumière
g	Passage fil de fer ou plein
p	Projection perspective ou orthographique
v	Recharger les fichiers des nuanceurs et recréer le programme
;	Choisir le modèle : cube, théière, tore, sphère, dodécaèdre, icosaèdre
i	Alterner entre le modèle d'illumination : Gouraud ou Phong
r	Alterner entre le modèle de réflexion spéculaire : Phong ou Blinn
s	Alterner entre le modèle de spot : OpenGL ou Direct3D
+ ou -	Incrémenter ou décrémenter l'angle maximal
. ou /	Incrémenter ou décrémenter l'exposant
t	Change la texture utilisée (aucune, dé, échiquier)
w	Change le mode de répétition de la texture
c	Changer l'affichage de l'objet texturé avec couleurs ou sans couleur
n	Lorsqu'on utilise les nuanceurs, le noir est transparent
ESC	Arrêt de l'application



## B Figures supplémentaires

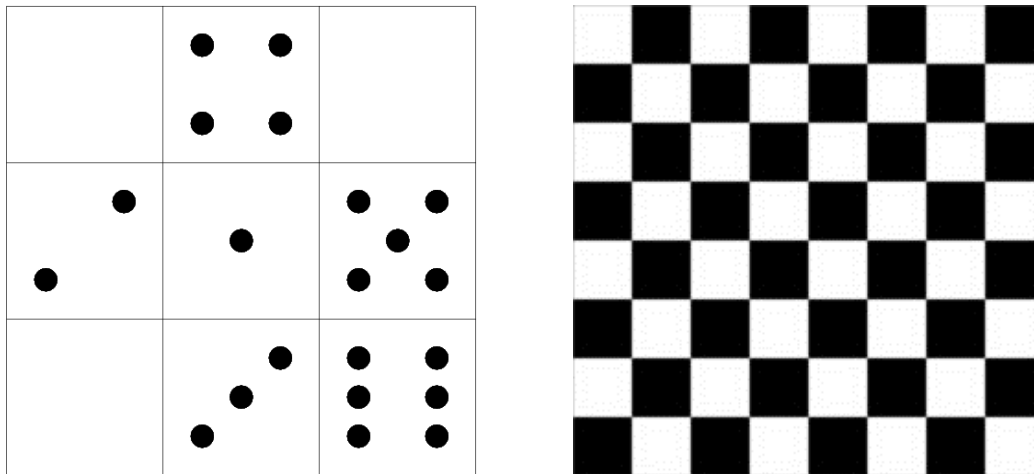


FIGURE 6 – Les textures fournies

## C Apprentissage supplémentaire

Partie 1 :

1. Valider les réflexions individuelles de chaque composante de la lumière (ambiante, diffuse, spéculaire) en modifiant le nuanceur.
2. Comparer visuellement les différences entre les modèles de Phong et Blinn.
3. Comment implanteriez-vous le modèle d'illumination de Lambert ?

Partie 2 :

4. Modifier les coordonnées de texture pour constater l'effet sur le résultat visuel.
5. Définir les bonnes coordonnées de texture pour un objet plus complexe composé de triangles.

## D Formules utilisées

### D.1 Modèles de spot d'OpenGL et de Direct3D

Un spot n'éclaire qu'à l'intérieur d'un cône, c'est-à-dire a une influence seulement si l'angle  $\gamma$  entre la direction du spot et la direction vers le point à éclairer est plus petit que l'angle d'ouverture  $\delta$  du spot, c'est-à-dire si  $\cos(\gamma) > \cos(\delta)$ . La différence entre les modèles d'OpenGL et de Direct3D réside dans la formule pour calculer le facteur qui multiplie l'intensité lumineuse du spot à l'intérieur du cône :

- OpenGL utilise le facteur :  $(\cos(\gamma))^c$

- Direct3D utilise le facteur :  $(\cos(\gamma) - \cos(\theta_{outer})) / (\cos(\theta_{inner}) - \cos(\theta_{outer}))$

où :

$\cos(\delta)$ :	cosinus de l'angle d'ouverture	= $\cos(\text{LightSource}[0].\text{spotCutoff})$
$\vec{L}_n$ :	direction du spot	= $\text{LightSource}[0].\text{spotDirection}$
$c$ :	exposant du spot	= $\text{LightSource}[0].\text{spotExponent}$
$\cos(\gamma)$ :	est obtenue par	$(\vec{L} \cdot \vec{L}_n)$
$\cos(\theta_{inner})$ :	est remplacé <i>dans ce TP</i> par	$\cos(\delta)$
$\cos(\theta_{outer})$ :	est remplacé <i>dans ce TP</i> par	$(\cos(\delta))^{1.01+c/2}$

### D.2 Modèles de réflexion spéculaire de Phong et de Blinn

Le calcul de la réflexion spéculaire fait intervenir un produit scalaire entre deux vecteurs. La différence entre les modèles de Phong et de Blinn réside dans le choix des deux vecteurs utilisés :

- Phong utilise :  $\vec{R} \cdot \vec{V} = \text{reflect}(\vec{L}, \vec{N}) \cdot \vec{V}$

- Blinn utilise :  $\vec{B} \cdot \vec{N} = \text{bissectrice}(\vec{L}, \vec{V}) \cdot \vec{N}$

où :

$\vec{N}$ :	normale à la surface	
$\vec{L}$ :	direction du point vers la source lumineuse	
$\vec{R}$ :	direction du rayon réfléchi	= $\text{reflect}(\vec{L}, \vec{N})$
$\vec{V}$ :	direction du point vers l'observateur	
$\vec{B}$ :	bissectrice entre les vecteurs $\vec{L}$ et $\vec{V}$	= $\text{normalise}(\vec{L} + \vec{V})$