



INF2705 Infographie

Spécification des requis du système

Travail pratique 3

illumination, textures et tessellation

Département de génie informatique et génie logiciel
École polytechnique de Montréal
Hiver 2019

Table des matières

1	Introduction	2
1.1	But	2
1.2	Portée	2
1.3	Remise	2
2	Description globale	3
2.1	But	3
2.2	Travail demandé	3
3	Exigences	9
3.1	Exigences fonctionnelles	9
3.2	Exigences non fonctionnelles	9
3.3	Rapport	10
4	Liste des commandes	11
5	Figures supplémentaires	12
6	Apprentissage supplémentaire	12

1 Introduction

Ce document décrit les exigences du TP3 « *Illumination, textures et tessellation* » 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` » afin de créer l'archive « **INF2705_remise_TPn.zip** » 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 le fichier Rapport.txt et tout le code source du TP (`makefile`, `*.h`, `*.cpp`, `*.glsl`, `*.txt`).

2 Description globale

2.1 But

Le but de ce TP est de permettre à l'étudiant mettre en pratique l'illumination des objets, l'application de textures pour colorer ou créer l'illusion d'un relief en utilisant des nuanceurs en GLSL. Ce travail pratique permet aussi de se familiariser avec les nuanceurs de tessellation.

2.2 Travail demandé

Partie 1 : l'illumination des objets

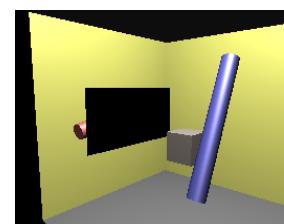
On demande de calculer l'illumination sur divers objets en utilisant deux sources de lumière (figure 1). Comme vu au cours, l'effet des deux sources de lumière est cumulatif et augmente les réflexions aux endroits doublement éclairés. Pour tous les objets, les modèles d'illumination de Gouraud et de Phong (figure 2) seront mis en œuvre et on pourra aussi choisir, pour le calcul de la réflexion spéculaire, entre le modèle de Phong ou celui de Blinn (figure 3).

Vous observerez que l'illumination de Gouraud sur le cube montre bien les limites de cette approche. Les couleurs sont calculées aux sommets de chaque face du cube et interpolées sur toute la face à partir de ces valeurs aux coins. Mais puisqu'il n'y a que deux triangles par face, la réflexion spéculaire n'est jamais très « belle », voire même inexiste : ça dépend s'il y a ou non de la réflexion spéculaire aux coins du cube. Et s'il a réflexion spéculaire au coin, alors l'intensité est alors simplement interpolée linéairement sur la face. (Modifiez le point de vue pour bien constater cet effet ... pas très souhaitable.) Ce problème de rendu dû à la faible résolution des faces sur le cube sera résolu, à la partie 3 du TP, par l'utilisation de nuanceurs de tessellation afin de subdiviser les faces du cube (figure 8). En attendant d'utiliser la tessellation, on acceptera ce défaut d'affichage sur le cube.

Pour démarrer, on pourra utiliser les nuanceurs des exemples du cours :

cours.polymtl.ca/inf2705/exemples/06-IlluminationMiroir/ et

cours.polymtl.ca/inf2705/exemples/06-Illumination/



Note : Ce TP utilise des « *Uniform Buffer Object* » (UBO) afin de transférer en bloc les variables uniformes aux nuanceurs. L'usage des UBO est très semblable aux VBO et permet surtout que le passage des valeurs des variables uniformes aux nuanceurs soit beaucoup plus efficace. Dans ce TP, on utilise ainsi quatre UBO qui correspondent aux quatre blocs de variables uniformes utilisés dans les nuanceurs : LightSource, FrontMaterial, LightModel et varsUnif. Les trois premiers blocs contiennent les variables uniformes servant à l'illumination, tandis que le dernier bloc contient les variables uniformes de l'application. (Les noms des variables sont inspirés de OpenGL 2.x.)

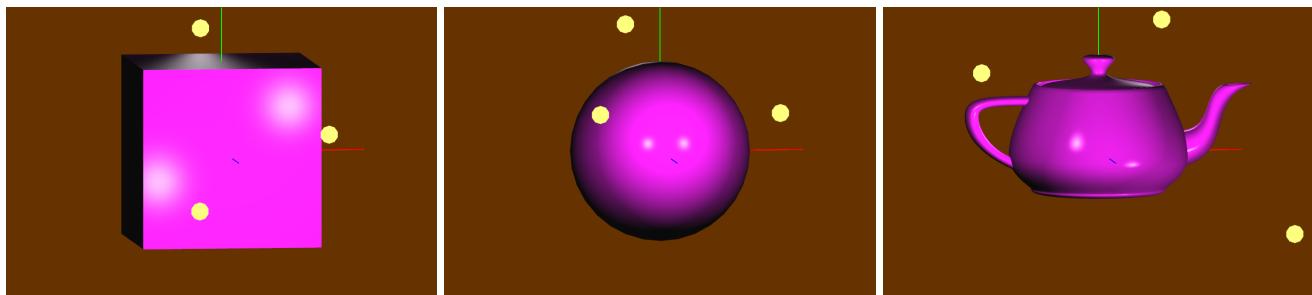


FIGURE 1 – Divers objets illuminés.

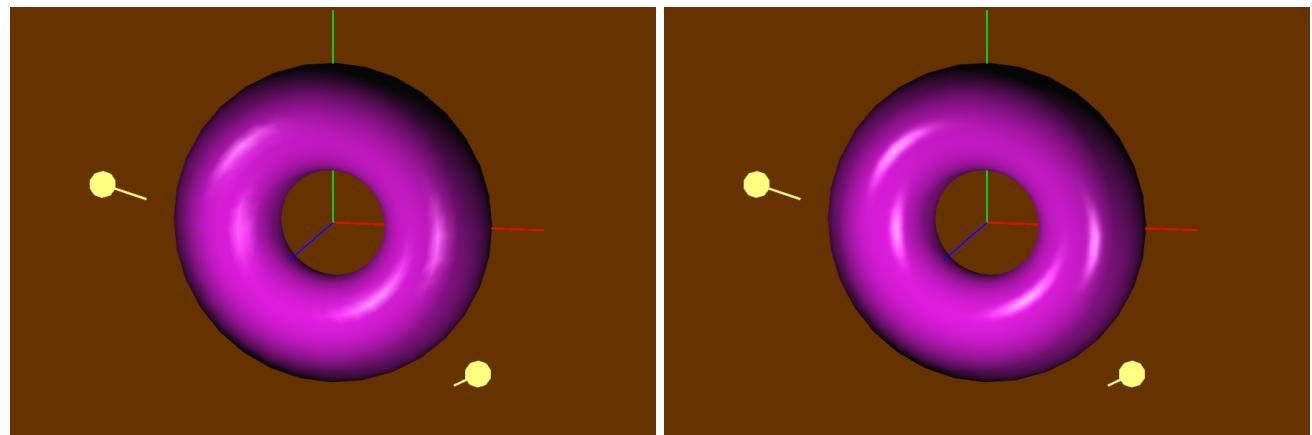


FIGURE 2 – Rendu avec réflexion spéculaire de Phong : a) illumination de Gouraud, b) illumination de Phong.

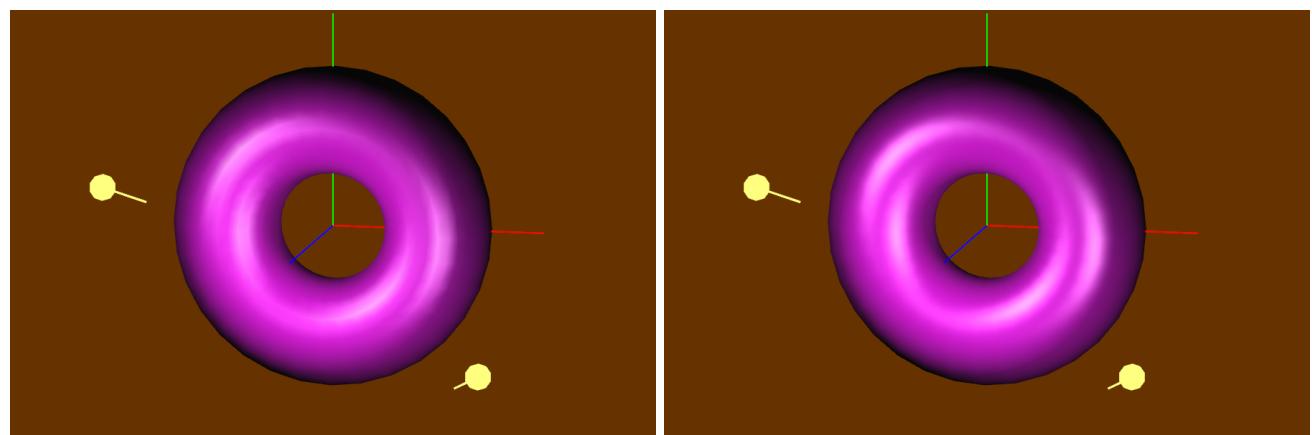


FIGURE 3 – Rendu avec réflexion spéculaire de Blinn : a) illumination de Gouraud, b) illumination de Phong.

Partie 2 : l'application de textures

Le logiciel permettra d'afficher des textures (figure 12) sur les objets illuminés.

- Le 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 (figure 5).
- Le patron d'échiquier sur le cube : on spécifiera les coordonnées de texture afin de montrer l'échiquier répété 3 fois dans les deux directions sur chaque face (figure 4.).
- Les autres textures sur le cube : on spécifiera les coordonnées de texture afin de montrer la texture directement sur chaque face (figure 4.).
- Les autres objets (tore, sphère, théière, cube, cylindre, cône) définissent eux-mêmes leurs coordonnées de texture.

L'utilisateur pourra choisir son type de rendu en tenant compte de la couleur de base de l'objet ou d'une couleur grise uniforme grisUniforme = $\text{vec4}(0.7, 0.7, 0.7, 1.0)$ (semblable à la figure 4). Enfin, on pourra choisir que les texels foncés de la texture ($\text{rgb} < 0.5$) soient affichés normalement (foncés), colorés (la moyenne de la couleur de la texture et de la couleur de l'objet) ou complètement transparents (figures 5 et 6).

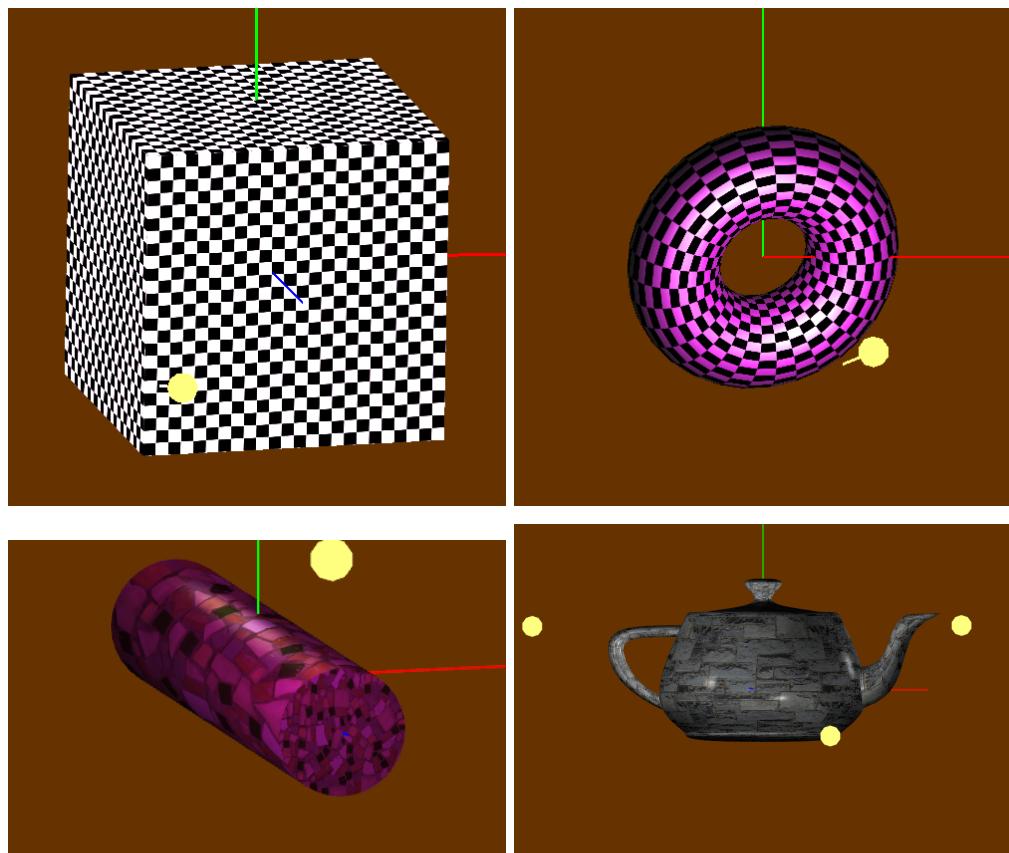


FIGURE 4 – Textures sur divers objets

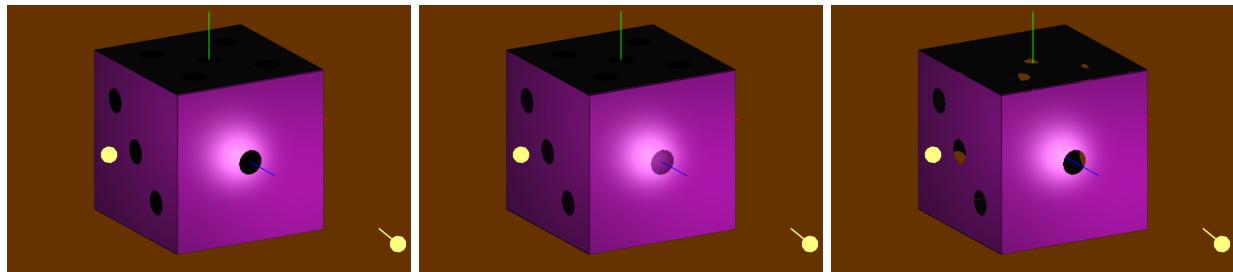


FIGURE 5 – Texture appliquée sur le dé en 3D (texels foncés opaques, colorés ou transparents)

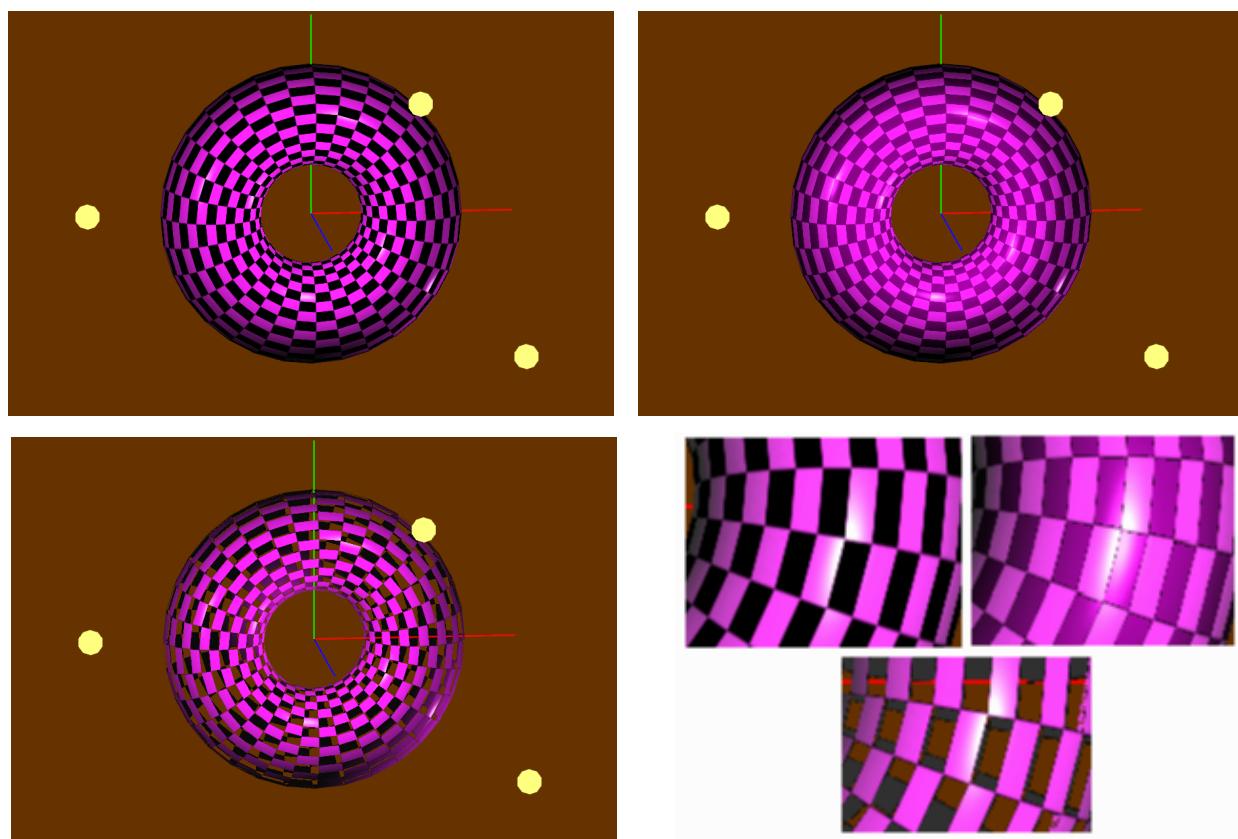


FIGURE 6 – Les texels foncés peuvent être affichés normalement, colorés ou transparents

Partie 3 : l'utilisation des nuanceurs de tessellation

On aura remarqué que le modèle de Gouraud n'offre pas un très beau rendu graphique, en comparaison de celui de Phong, lorsque la surface du cube n'est composée que seulement de deux triangles (figure 7). Afin de corriger cet effet, on utilisera des nuanceurs de tessellation pour subdiviser chaque face du cube et fournir un plus beau rendu lorsque l'algorithme de Gouraud est utilisé. Il faudra d'abord activer les nuanceurs de tessellation dans le programme principal et ensuite utiliser des GL_PATCHES afin d'afficher les faces du cube qui seront alors subdivisées en un nombre variable de sous-triangles (figures 8 et 9).

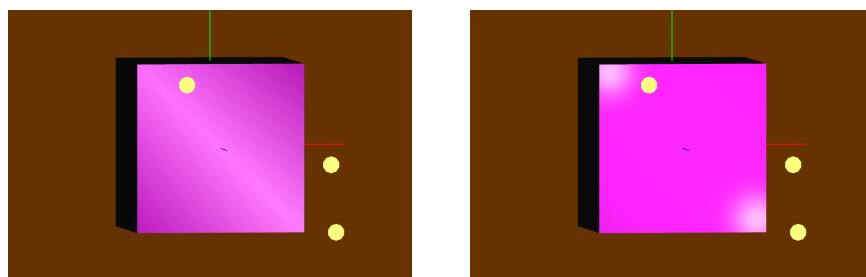


FIGURE 7 – Comparaison entre les rendus du modèle de Gouraud et du modèle de Phong

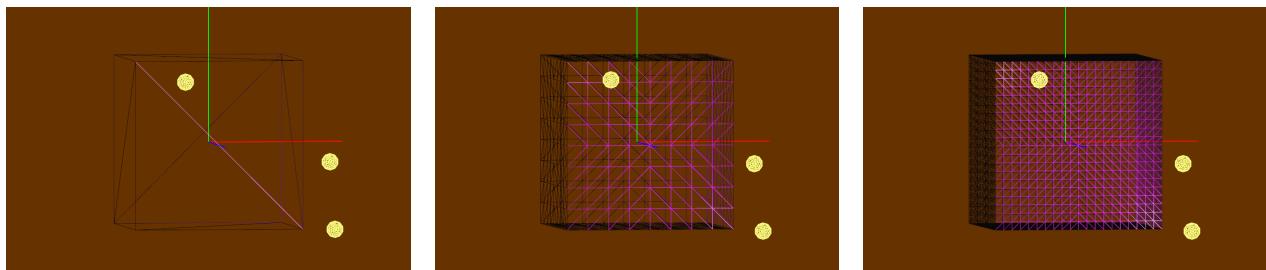


FIGURE 8 – La face du cube avec des niveaux de tessellation de 1, 8 ou 20 (fil de fer)

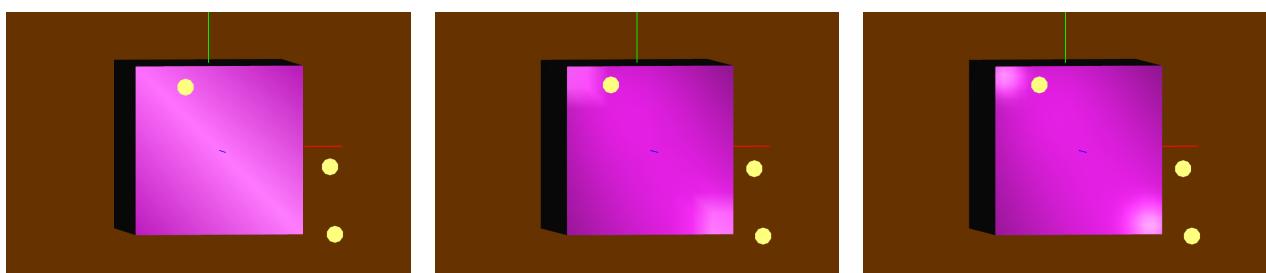


FIGURE 9 – La face du cube avec des niveaux de tessellation de 1, 8 ou 20 (plein)

Partie 4 : le placage de relief

Le placage de relief permet de simuler la variation des normales à la surface en utilisant une texture spéciale qui contient des variations à appliquer aux normales calculées. Cet effet permet ainsi de montrer un relief (simulé) à la surface de l'objet (figure 10).

En utilisant deux textures, on peut colorer un objet avec une texture pour les couleurs, puis simuler l'effet ajoutée d'un placage de relief avec une texture supplémentaire pour les normales (figure 11).

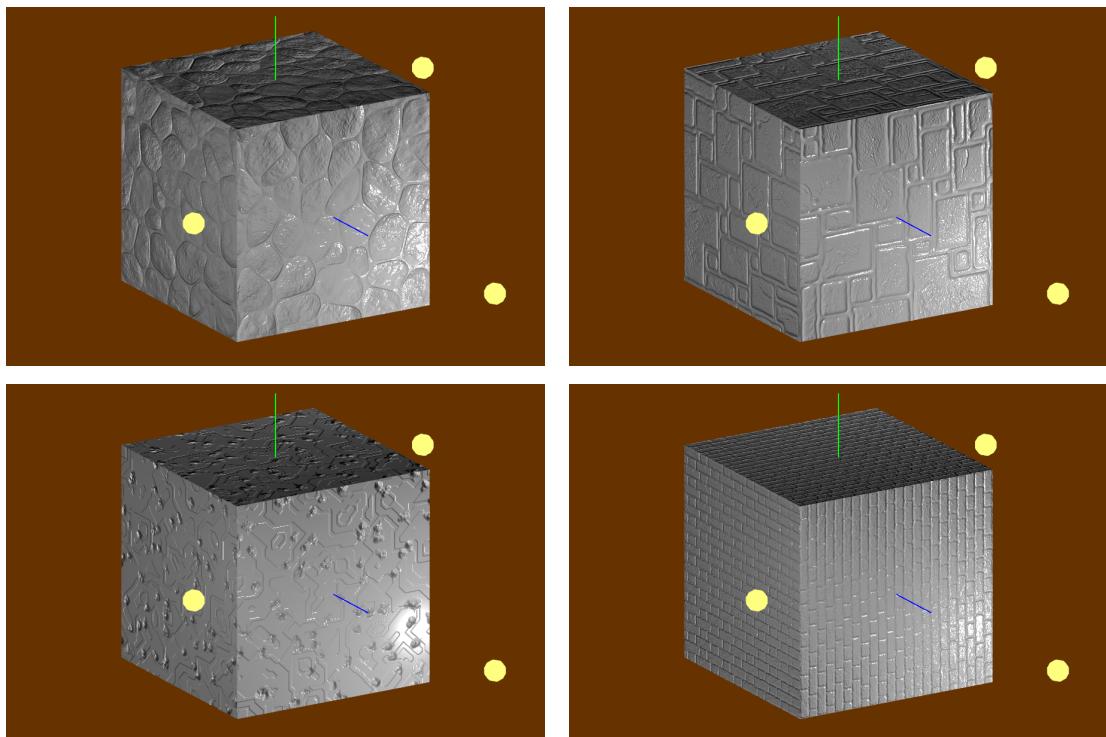


FIGURE 10 – Placage de relief seulement

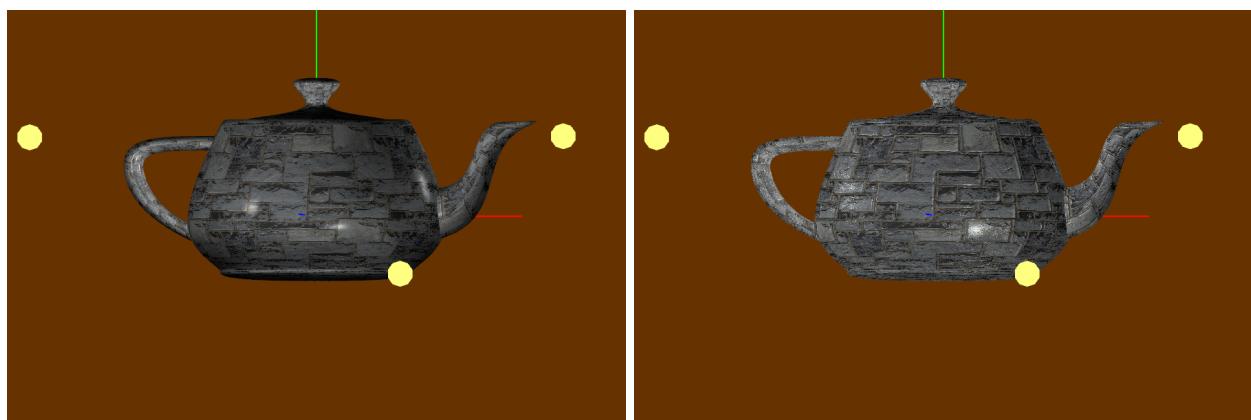


FIGURE 11 – Placage de texture, puis de texture et de relief

3 Exigences

3.1 Exigences fonctionnelles

Partie 1 :

- E1. Le modèle d'illumination de Phong est implanté comme demandé.
- E2. Le modèle d'illumination de Gouraud est implanté comme demandé.
- E3. Les modèles de réflexion de Phong et de Blinn sont implémentés comme demandé.

Partie 2 :

- E4. Les paramètres des textures sont bien initialisés et elles sont correctement affichées sur les objets.
- E5. Tous les objets texturés sont correctement illuminés.
- E6. Le cube est affiché correctement avec la texture du dé (figure 5).
- E7. Le cube et les autres objets sont affichés correctement avec la texture de l'échiquier (figure 4).
- E8. Les texels foncés sur l'objet sont affichés normalement, mi-colorés ou transparents (figure 6).
- E9. L'objet texturé peut être affiché avec ou sans sa couleur (ex. : figures 4).

Partie 3 :

- E10. Des nuanceurs de tessellation sont utilisés pour corriger le rendu illuminé avec le modèle de Gouraud.

Partie 4 :

- E11. Une texture de relief peut être appliquée sur les objets (figure 10).
- E12. Deux textures peuvent être appliquées conjointement pour modifier couleurs et normales (figure 11).

3.2 Exigences non fonctionnelles

Notez bien que, normalement, on chargerait des nuanceurs différents pour chaque cas d'utilisation afin d'augmenter la performance en évitant les énoncés conditionnels à la valeur d'une variable uniforme.

Toutefois, dans le contexte de TP, on utilisera sciemment de tels énoncés conditionnels aux valeurs des variables uniformes (modifiables interactivement) : le modèle d'illumination, si on veut des pixels transparents, etc. Ceci permettra de plus facilement contrôler le type de rendu, en plus de faciliter votre développement... et la correction !

Dans vos nuanceurs, on pourra donc voir des énoncés semblables à ceux-ci :

```
if ( variableUniforme == ... ) ... else ... ;
```

ou, mieux encore :

```
( variableUniforme == ... ) ? ... : ...
```

3.3 Rapport

Vous devez répondre aux questions dans le fichier Rapport.txt qui sera inclus 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.)

4 Liste des commandes

Touche	Description
q	Quitter l'application
x	Activer/désactiver l'affichage des axes
9	Permuter l'utilisation des nuanceurs de tessellation
v	Recharger les fichiers des nuanceurs et recréer le programme
i	Augmenter le niveau de tessellation interne
k	Diminuer le niveau de tessellation interne
o	Augmenter le niveau de tessellation externe
l	Diminuer le niveau de tessellation externe
u	Augmenter les deux niveaux de tessellation
j	Diminuer les deux niveaux de tessellation
w	Alterner entre le modèle d'illumination : Gouraud, Phong
r	Alterner entre le modèle de réflexion spéculaire : Phong, Blinn
s	Alterner entre le modèle de spot : OpenGL, Direct3D
p	Permuter lumière positionnelle ou directionnelle
a	Incrémenter l'angle d'ouverture du cône du spot
z	Décrémenter l'angle d'ouverture du cône du spot
y	Incrémenter le coefficient de brillance
h	Décrémenter le coefficient de brillance
m	Choisir le modèle affiché : cube, tore, sphère, théière, cylindre, cône
0	Replacer Caméra et Lumière afin d'avoir une belle vue
t	Choisir la texture de couleurs utilisée : aucune, dé, échiquier, mur, métal, mosaique
e	Choisir la texture de normales utilisée : aucune, pierre, bulles, mur, brique, circuit
c	Changer l'affichage de l'objet texturé avec couleur ou sans couleur
f	Changer l'affichage des texels foncés (normal, mi-coloré, transparent)
POINT	Augmenter l'effet du déplacement
VIRGULE	Diminuer l'effet du déplacement
BARREOBLIQUE	Permuter la projection : perspective ou orthogonale
g	Permuter l'affichage en fil de fer ou plein
n	Utiliser ou non les normales calculées comme couleur (pour le débogage)
ESPACE	Permuter la rotation automatique du modèle
BOUTON GAUCHE	Tourner l'objet
BOUTON MILIEU	Modifier l'orientation du spot
BOUTON DROIT	Déplacer la lumière
Molette	Changer la taille du spot

5 Figures supplémentaires



FIGURE 12 – Les textures fournies

6 Apprentissage supplémentaire

Partie 1 :

1. Valider les réflexions individuelles de chaque composante de la lumière (ambiante, diffuse, spéculaire) en commentant sélectivement les lignes qui calculent la contribution de chaque composante dans le nuanceur.
2. Comparer visuellement les différences entre les modèles de Phong et Blinn, en particulier pour une lumière rasante.
3. Comment feriez-vous pour implanter le modèle d'illumination de Lambert ?
4. Dans un nuanceur, utilisez des couleurs de lumière diffuse différentes pour chacun des sources lumineuses. Comment se mélangent les couleurs à l'écran ?
5. Afficher une sphère au lieu du cube. (Vous devrez alors calculer et spécifier ses normales.)

Partie 2 :

6. Modifier les coordonnées de texture pour constater l'effet sur le résultat visuel.
7. Définir les bonnes coordonnées de texture pour un objet plus complexe composé de triangles.
8. Modifier les coordonnées de texture afin de déplacer la texture en fonction du temps.

Partie 3 :

9. Rapetisser légèrement chaque triangle afin de voir à l'intérieur du cube entre les triangles.
10. Déformer la surface du cube différemment, par exemple pour la convertir en ellipsoïde.
11. Utiliser le facteur de déformation `facteurDeform` pour mettre en œuvre une déformation variable.

Partie 4 :

12. Que se passe-t-il lorsque vous déplacez la source lumineuse à l'intérieur de l'objet ?
13. Utiliser une atténuation en fonction de la distance comme prévu dans le modèle d'illumination.
14. Plutôt que de mettre en noir les fragments non directement éclairés par la source de lumière, diminuez simplement leur intensité par un facteur de 2.
15. Transformer les sources de lumière en spot en utilisant les paramètres appropriés, selon le modèle inspiré de OpenGL ou de Direct3D.
16. Définissez vos propres modèles de spot (autres paramètres et fonction de calcul) qui vous semblent intéressant.