POGL
TP3 : OpengGL 4 - *Shaders*

Jonathan Fabrizio
LRDE - EPITA

Objectif

L'objectif de ce T.P. est de programmer des *Shaders*.
Note : Veillez à bien conserver les différentes versions de vos *Shaders*.

1 Travail préliminaire

Téléchargez le squelette du programme. Regardez les sources et essayez de comprendre ce qu'il fait et comment il fonctionne.

Les *shaders* sont dans les fichiers `vertex.shd` et `fragment.shd`. Vous devez compléter ces fichiers pour répondre aux questions posées.

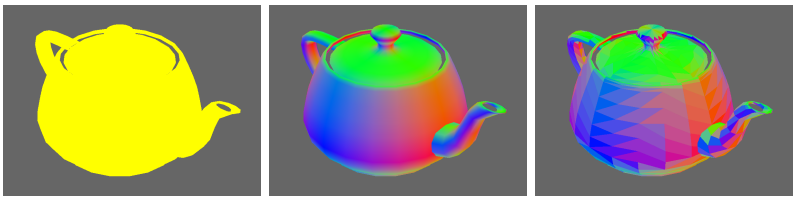
2 La couleur

Dans cette partie vous utiliserez seulement les champs `position` et `color`. `position` contient la position du sommet dans l'espace, `color` donne la couleur associée.

Question 1 : Complétez les *shaders* afin d'afficher l'objet en jaune.

Question 2 : Modifiez les *shaders* afin d'afficher la couleur de l'objet qui est passée par le *VBO* `color`. D'après vous, que représente cette couleur ?

Question 3 : Modifiez les *shaders* afin de désactiver l'interpolation de la couleur entre les *vertex shader* et *fragment shader*. Vous devriez voir apparaître les triangles qui compose l'objet.

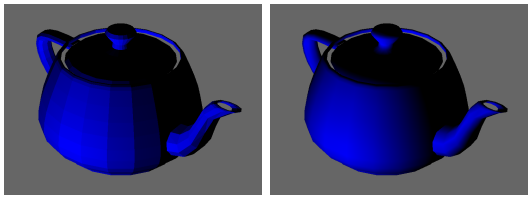


3 Les normales

Dans cette partie vous utiliserez encore le champs `position` mais plus le champs `color`. En revanche, vous utiliserez les deux champs `normal`. `normalFlat` donne, pour chaque sommet, la normale au triangle. `normalSoomth` donne, pour chaque sommet, une interpolation des normales des triangles adjacents.

Question 1 : Affichez maintenant l'objet en bleu mais en tenant compte de la lumière. Chaque face doit avoir une couleur unis.

Question 2 : Lissez maintenant l'objet (toujours en bleu).



4 Les textures

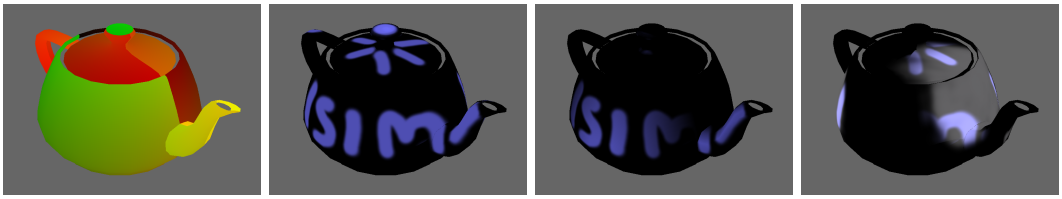
Dans cette partie vous utiliserez encore le champs `position` et `normalSoomth`. Vous utiliserez aussi le champs `uv` qui donne pour chaque sommet ses coordonnées dans le repère texture.

Question 1 : Affichez comme couleur : sur le canal rouge le u et sur le canal vert le v.

Question 2 : Affichez la texture sur l'objet (et uniquement la texture) sans tenir compte de la lumière.

Question 3 : Affichez la texture sur l'objet en tenant compte de la lumière.

Question 4 : Pour simuler l'effet d'un spot, ne tenez plus compte de la lumière mais combinez la texture (que vous pouvez un peu éclaircir pour un résultat plus joli) avec l'image du spot.



5 Des effets

Reprenez simplement votre objet avec la couleur d'origine.

Question 1 : Affichez l'image en niveau de gris

Question 2 : Affichez l'image en vert (style *night vision*. Pour faire plus style essayez de changer l'intensité en fonction des lignes - c.f. `gl_FragCoord`).

Question 3 : Affichez l'objet en vert et modulez l'intensité en fonction de la profondeur.

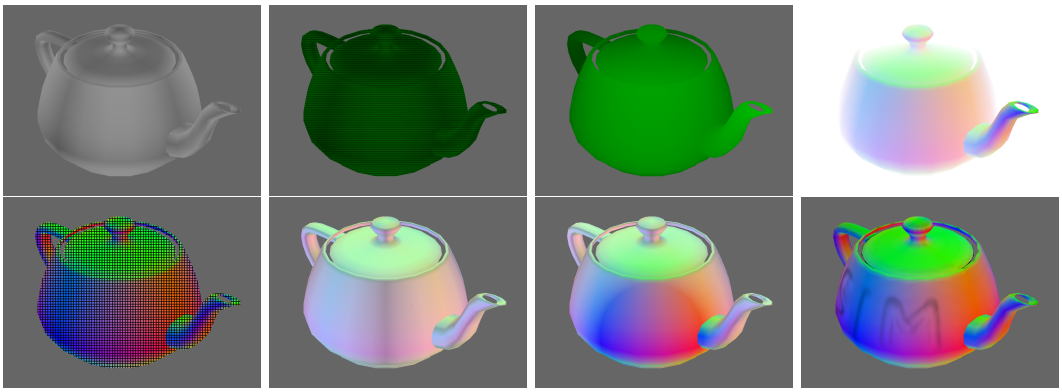
Question 4 : Ajoutez du brouillard (pensez à changer la couleur de fond).

Question 5 : Affichez l'objet sur un damier seulement (sur des carrés de 6×6 séparés par 2 pixels. Entre les carrés il y aura la couleur de fond ou du noir.

Question 6 : Affichez une image plus terne.

Question 7 : Affichez un dégradé "couleur vers niveaux de gris" en fonction de la distance au centre de l'image.

Question 8 : A l'aide de *normalmap*, faites apparaître des bosses sur la théière. Les normales sont codées dans le repère tangent. Chaque composante est codée entre 0 et 1 et doivent être ramenée entre -1 et 1.



Bonus

Parmi toutes les images, une est fausse. Laquelle ?