

IFT3355: Infographie

Travail Pratique 3: Ombrage, éclairage et textures

Remise: Mercredi 11 mars 2020 23:59

1 Introduction

L'objectif principal de ce TP est d'explorer l'éclairage et l'ombrage des modèles 3D. Vous simulerez quatre styles d'ombrage différents : Gouraud, Phong, Blinn-Phong et les textures. Dans ce TP, vous allez *ajouter* de nouvelles paires de shaders (vertex et fragment shaders) pour effectuer ces tâches. Vous écrirez un peu plus de code, mais les shaders seront très similaires aux devoirs précédents.

2 Template

Le modèle de code se trouve dans le répertoire principal du TP. Il est très similaire au modèle précédent du TP 1. Pour ce devoir, vous devrez modifier le code et créer de nouveaux shaders. Nous vous fournissons la géométrie par défaut pour tous les objets que vous devrez ombrer, et nous avons déjà défini les propriétés d'une lumière directionnelle de base dans la scène (définie par les variables `lightColor`, `lightDirection` et `ambientColor`). Nous avons également défini les propriétés de matériau suggérées pour les objets que vous allez ombrer (définies dans `kSpecular`, `kDiffuse` et `kAmbient`). Vous serez responsable de la mise à jour des différents shaders pour mettre en œuvre les schémas d'éclairage souhaités.

3 Règles importantes

Les modèles d'éclairage que vous allez implémenter dans cette mission sont très courants dans les applications graphiques. Ainsi, `three.js` les a déjà implémentés dans un certain nombre de matériaux par défaut fournis avec `three.js`. Ceux-ci peuvent être utilisés pour tester si vous obtenez des résultats raisonnables, mais **vous devez implémenter ces modèles d'ombrage vous-même dans vos propres shaders personnalisés**. Nous avons créé des modèles de shaders similaires aux devoirs précédents à partir desquels vous pourrez travailler. Pour la question de texturation, nous avons utilisé la fonction `three.js THREE.ImageUtils.loadTexture` pour charger la texture, mais vous devez la transmettre aux shaders et rechercher la couleur de texture vous-même.

4 Travail à réaliser (100 pts)

(15 pts) Ombrage de Gouraud Dans cette partie, vous commencerez avec un modèle d'éclairage simple et rapide. L'ombrage de Gouraud calcule l'éclairage d'un objet à chaque sommet (par exemple dans le vertex shader). Le pipeline graphique standard interpolera ensuite la couleur de l'objet entre les sommets pour obtenir les couleurs des fragments individuels. Un exemple du résultat (tiré de wikipedia) est présenté ci-dessous, notez à quel point les modèles ont tendance à avoir un aspect polyédrique ou pixelisé s'ils n'ont pas un nombre élevé de polygones. Le calcul de l'éclairage à un sommet devrait être basé sur le modèle de réflexion de Phong standard discuté en classe. Vous pouvez jouer avec les différents composants (diffus, spéculaire et ambiant) pour produire différents effets et matériaux.

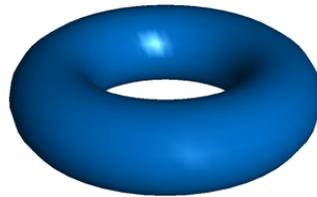


Figure 1: Tore ombré par Gouraud avec modèle de réflexion de Phong.

(15 pts) Réflexion et Ombrage de Phong Dans la section précédente, un modèle d'éclairage simplifié a été utilisé pour calculer la couleur de l'objet à un sommet donné. Ceci est rapide et efficace, mais n'a pas l'air très attrayant ou lisse. Dans cette section, vous devez implémenter le modèle de *réflexion* de Phong complet en transmettant les paramètres d'éclairage (tels que la normale du triangle et la position du sommet) en tant que *varying variables* au fragment shader. Ensuite, vous devez calculer l'éclairage du maillage à chaque fragment en utilisant les paramètres d'éclairage interpolés. Cela permet au modèle d'ombrage de mieux se rapprocher des surfaces courbes et produit un modèle d'ombrage très lisse. L'image suivante, extraite de l'article Wikipédia sur le modèle de réflexion de Phong, montre comment les différents composants se présentent individuellement et se résument ensemble:

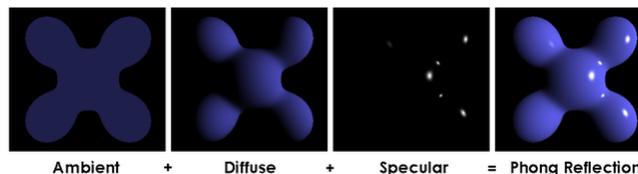


Figure 2: Modèle de réflexion de Phong

Les calculs principaux doivent tous être effectués dans le fragment shader (dans `phong.fs.gsl`). Notez que vous aurez toujours besoin d'un vertex shader (dans `phong.vs.gsl`), pour transmettre les informations appropriées à votre fragment shader.

- (15 pts) **Ombage de Blinn-Phong** Au lieu de calculer le produit scalaire entre le vecteur de lumière réfléchi et celui d'observation, un produit scalaire entre le vecteur à mi-chemin entre la lumière et la direction d'observation, et la normale de surface peut être utilisé. C'est l'essence du modèle d'ombage de Blinn-Phong. Implémentez un shader qui simule le modèle d'ombage de Blinn-Phong. Le manuel ou les diapositives du cours contiennent plus de détails sur ce sujet.
- (25 pts) **Texturation de base** Un objet modélisé a souvent une surface à très petits détails et facettes (par exemple, le grain de bois sur une boîte, les rayures sur le métal, les taches de rousseur sur la peau). Ceux-ci sont très difficiles, voire impossibles, à modéliser comme un seul matériau éclairé par un modèle semblable à un Phong. Afin de simuler efficacement ces matériaux, nous avons généralement recours à la texturation. Dans la texturation de base, les coordonnées UV sont prises à partir du vertex buffer (three.js les fournit gratuitement sur les objets par défaut tels que la sphère dans notre devoir). Les coordonnées UV sont un index 2D dans un fichier image dont les valeurs RVB peuvent être utilisées pour aider à colorer le maillage.

Bien que vous puissiez mélanger le Phong et la texturation dans la pratique, dans cette tâche, nous vous demanderons d'utiliser les données de texture comme couleur finale du fragment. Nous vous fournissons un fichier de texture standard appelé "gravel-rocks-texture.jpg" que vous devez appliquer à la quatrième sphère à l'aide d'un shader personnalisé. Il est chargé dans la variable "sphereTexture" dans a3.js Cela devrait être fait dans le fragment shader. **Vous êtes encouragé mais pas obligé de tenir compte de la distorsion de perspective dans la texture.**

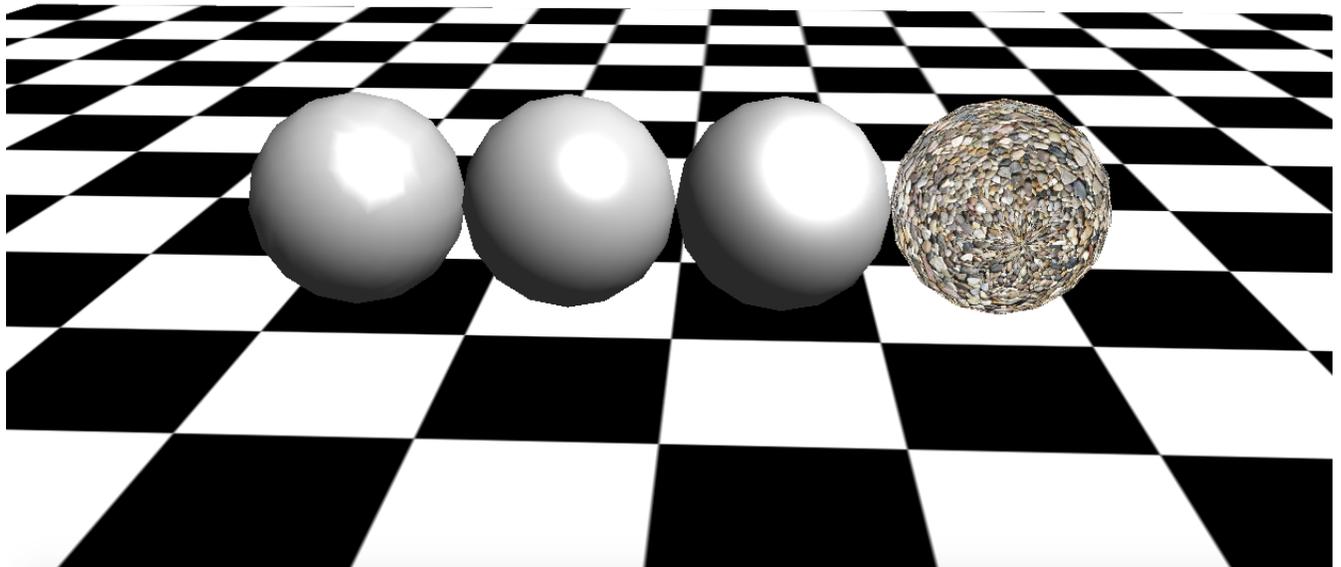


Figure 3: Exemples de résultats pour Gouraud, Phong, Blinn-Phong et texturation, affichés de gauche à droite.

(30 pts) License créative. Cette partie dépend de vous. Vous devez créer un nouveau shader ou modifier l'un des shaders ci-dessus pour implémenter des effets d'éclairage, d'ombrage ou de texturation intéressants similaires à la complexité de ceux ci-dessus. Quelques recommandations :

- Mettre en œuvre un matériau anisotrope, tel que du métal brossé.
- Ajouter des spots et des lumières directionnelles, peut-être les animer.
- Simuler le brouillard.
- Implémenter une carte de normales
- Mettre en œuvre un schéma pour calculer les coordonnées UV pour les objets qui n'en ont pas (par exemple le tatou)
- Mettre en œuvre une carte cubique pour créer un objet réfléchissant

Des points bonus seront attribués pour récompenser des pistes d'exploration particulièrement remarquables.

Remise : Créez un répertoire `VOTRENOM_TP3` et placez-y les fichiers sources du projet en respectant la hiérarchie du template. Placez-y également un fichier `README.txt` contenant votre nom ainsi que toute information que vous aimeriez fournir au correcteur. Comprimez ce répertoire avant de le rendre sur StudiUM.