

TD d'infographie n°3

Cours d'infographie
DUT Informatique S4

Modélisation lumineuse, matériaux et textures

*Modélisation et exploitation des sources de lumière.
Paramétrage des matériaux sous OpenGL. Application
de texture.*

Le but de ce travail est de représenter de manière réaliste (d'un point de vue de l'éclairage) une scène virtuelle déjà définie. Le but est de savoir appliquer les couleurs et le modèle d'illumination à une scène préétablie. Enfin, le TD vous montrera comment plaquée une image sur les objets de la scène. Pour ce faire, commencez déjà par récupérer l'archive `td3_texture.tgz` située sur le site internet <http://igm.univ-mlv.fr/~biri/>. Ce code source comprend un fichier principal (main) nommé `eclairage.c` ainsi que les couples de fichiers `ioPPM` (.h et .c) servant à lire et écrire des images ppm, `texture` (.c et .h) servant à stocker et gérer des textures en OpenGL, et enfin `gladd` (.c et .h) offrant des fonctions de dessin de forme (quad, cylindre et cône).



Exercice 1 : Propriété des matériaux & éclairage

Pour avoir une « illumination » réaliste d'une scène virtuelle, il faut effectuer 3 tâches. La première est de créer des sources de lumière. La seconde sera de paramétrer tous les objets de la scène pour qu'ils aient des propriétés de couleur. La troisième sera d'indiquer à la carte graphique que l'on souhaite un rendu « réaliste » avec des sources de lumière. Pour notre scène nous souhaitons utiliser 2 sources de lumière **de type spot** : un plafonnier, et un halogène « tournant ».

1. Vous commencerez par créer les sources de lumière. Cela se fait en utilisant la fonction `glLightfv` et `glLightfv`. Regardez dans le fichier `eclairage.c` dans la fonction `init`. Vous y trouverez un exemple d'utilisation de ces fonctions. Inspirez vous en pour fixer les coefficients de diffusion `GL_DIFFUSE` (couleur blanche), et de spéculaire `GL_SPECULAR` (couleur noire) de chacune des deux lampes. Vous fixerez également l'angle du spot (propriété `GL_SPOT_CUTOFF`) et le coefficient d'atténuation de ce spot (`GL_SPOT_EXPONENT`) à mettre à 8. Il reste deux propriétés que nous n'avons pas définies : la position et l'orientation du spot (respectivement `GL_POSITION` et `GL_SPOT_DIRECTION`). Celles-ci doivent être définies lors du rendu, c'est à dire dans la fonction `drawFunc`. En effet, la position et l'orientation subissent les transformations définies dans la matrice `MODELVIEW`. Positionnez (position et direction), après l'appel à la fonction `glutLookAt` le plafonnier. Vous utiliserez les variables globales `pos_light_plaf` et `dir_light_plaf` situées au début du fichier `eclairage.c`. **Notez que pour le moment, vous ne percevez aucun changement.**
2. Il nous faut maintenant définir les propriétés des objets. Cela se fait grâce à la fonction `glMaterialf` qui prend trois arguments. Le premier indique quelles faces recevront la propriété (face avant, arrière ou les deux). Vous mettrez systématiquement `GL_FRONT_AND_BACK` (les deux faces donc) pour ce premier argument. Pour les deux autres arguments, il s'agit du mot clé de la propriété et de la valeur associée. Regardez dans la documentation de référence d'OpenGL afin de bien renseigner ces valeurs. Renseignez les propriétés `GL_AMBIENT_AND_DIFFUSE` pour toutes les surfaces. Pour la propriété `GL_SPECULAR`, mettez du noir pour toutes les surfaces. **Notez que pour le moment, vous ne percevez aucun changement.**
3. Enfin nous allons activer le rendu lumineux. Cela se fait en activant simplement l'illumination par un `glEnable(GL_LIGHTING)` dans la fonction `init`. Vous devez également activer, une par une, les sources de lumière grâce à la fonction `glEnable(GL_LIGHTi)` où *i* doit être remplacé par un entier entre 0 (première lumière) et 7 (8ème lumière). Activez la lumière 0 (le plafonnier). Vous devriez maintenant voir l'illumination « réaliste » de la scène. Modifiez la position et l'orientation du plafonnier. Faites quelques tests de changement de couleur pour vos matériaux (propriétés des objets). Essayez également de modifier l'angle du spot ou de changer son atténuation.
4. Essayez maintenant de positionner la lumière dans l'halogène tournant. N'oubliez pas de l'activer également dans la fonction `init`.

Exercice 2: Texture !

Dans cette partie nous allons exploiter les textures. Pour apposer une texture à une surface, il y a deux étapes à suivre. La première consiste à charger la texture en mémoire de la carte graphique. La seconde consiste à l'application de cette texture sur un maillage. Nous allons, dans cet exercice, mettre une texture de bois au plancher et une image de paysage sur la « fenêtre du fond ».

1. Pour la première étape (et la suivante d'ailleurs) il faut commencer par « activer » les textures. Cela se fait grâce à l'instruction `glEnable(GL_TEXTURE_2D)` à insérer dans la fonction `init`. Puis créez, dans la fonction `init`, les deux textures (remplir les variables globales `tex_plancher` et `tex_tableau`). Pour ce faire, appuyez vous sur les fonctions offertes dans le fichier `texture.h`. Deux textures sont à votre dispositions dans le répertoire `texture` mais vous pouvez aller en chercher d'autres sur internet (et les convertir en ppm).
2. Vous devez également (toujours dans la même fonction), initialiser ces textures. Regardez bien le code fourni dans `texture.c` et essayer de décrypter les appels de fonction. La première étape est faite.
3. Pour le seconde étape, il faut dire à la carte graphique quelle texture utiliser. Cela se fait grâce à la fonction `glBindTexture` mais vous, vous pourrez utiliser la fonction précodée `chargeTexture()`. De plus, il faut affecter, à chaque sommet, une coordonnée de texture grâce à la fonction `glTexCoord2f`. Cela est déjà fait dans les fonctions de dessin données dans `gladd.c`. Regardez bien comment sont fournies ces coordonnées de texture. Activez la texture de bois au moment du dessin du plancher et désactivez la ensuite.
4. Plaquez une texture sur le « tableau » ou la « fenêtre » du fond.
5. Vous pouvez constater que la texture de bois est très étirée. Modifiez la fonction `glDrawQuad` afin de permettre à une texture de se répéter n fois (n étant un paramètre à rajouter à la fonction) « horizontalement » et « verticalement ». Il suffit pour cela de jouer sur les coordonnées de texture.
6. On souhaite donner un aspect de fenêtre au quad du fond. Pour ce faire, il ne faut pas que la lumière des sources lumineuses se mêle avec la couleur de la texture. Ajouter, juste avant le chargement de texture, l'appel à la fonction `glTexEnv(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, XXX)` avec le bon argument `XXX` permettant de remplacer complètement la couleur calculée avec la couleur de la texture. N'oubliez pas de revenir au comportement « normal » de modulation après (même appel avec un autre argument `XXX`).

Exercice 3: Fenêtre sur la plage

Afin d'augmenter le réalisme, vous pouvez associer d'autres textures aux murs. Modifiez également le mur du fond afin de créer véritablement une fenêtre donnant sur un paysage. Pour ce faire, faites un « trou » dans le mur (en fait il faut maintenant que le mur soit constitué de 4 quads), puis positionnez un grand quad texturé derrière.

Rajoutez du mobilier.