

UJF - M1 Informatique 2005-2006 : Infographie

TP3 : Illumination et modèle d'éclairage

14 octobre 2005

Aujourd'hui nous allons étudier l'utilisation des lumières en openGL ainsi que l'effet des différents paramètres d'illumination utilisables.

1 Lumières

Récupérez le nouveau fichier `main.cpp`, y sont définies deux nouvelles procédures :

- `initLights` : initialise une lumière openGL en définissant sa position et ses particularités d'éclairage : une partie de la lumière est diffuse, une autre partie est spéculaire. Cette lumière est de type solaire, c'est-à-dire que tous les rayons sont parallèles et que la position définit en fait l'angle d'attaque de la lumière dans l'espace.
- `dessinerSphere` : dessine une sphère (grâce à la procédure `glutSolidSphere` de la librairie GLUT). Dans cette procédure sont définis les propriétés globales de la sphère : sa composante diffuse (ici verte), sa composante spéculaire (ici nulle) et sa brillance (*shininess*) ainsi qu'une couleur ambiante (ici nulle).

NB : la dernière composante des couleurs (car ce sont des vecteurs à 4 réels) est toujours 1, elle représente l'opacité (nous ne verrons pas cette notion aujourd'hui).

1.1 Analyse de l'éclairage

1. Compilez le programme et exécutez le avec le paramètre `"-t 6"` pour afficher la sphère.
2. Tournez autour de la sphère. Rajoutez les paramètres `"-c 100 -l 100"` pour obtenir plus de triangles sur la sphère. Qu'observez vous et pourquoi ?
3. Changez la couleur de la lumière (dans `initLights`) en `"{1,0,1,1}"`. Qu'observez vous et pourquoi ?
4. Revenez à la couleur précédente pour la lumière et donnez pour composante spéculaire du matériau le rouge : `"{1,0,0,1}"`. Qu'observez vous et pourquoi ?

5. Donnez pour composante spéculaire du matériau le blanc. Prenez moins de point pour la sphère ("c 10 -l 10"). Toujours avec un paramètre de brillance de 100, pourquoi la "tache" disparaît-elle selon le point de vue ?

Pour plus de détails sur l'éclairage en OpenGL, vous pouvez vous référer à un tutoriel (par exemple <http://www-evasion.imag.fr/Membres/Antoine.Bouthors/teaching/opengl/opengl5.html>).

1.2 Plusieurs sources lumineuses

1. Ajoutez une ou plusieurs autres lumières (changez leur position) et analysez l'effet de différents paramètres d'illumination des lumières et de la sphère.

2 Intégration du modèle d'éclairage à un maillage

Nous travaillerons maintenant avec le maillage indexé *sinus animé*.

2.1 Normales

Comme vous l'avez vu en cours, pour pouvoir évaluer la couleur en un point de la surface, vous devez connaître la normale de la surface en ce point.

1. Intégrez les normales dans le `MaillageIndexe`.
2. Ecrivez les fonctions nécessaires au calcul des normales d'un maillage.
3. Pour vérifier si vos normales sont correctes :
 - (a) Désactivez la lumière en commentant l'appel de procédure `initLights` dans le main.
 - (b) Visualisez la normale en chaque point en dessinant un vecteur à partir de chaque vertex. Pour dessiner un trait (une ligne) entre 2 points, il vous suffit d'utiliser le mode de rendu OpenGL correspondant (si vous rajoutez des vertex, OpenGL tracera une nouvelle ligne en partant du dernier point !):

```
glBegin(GL_LINES) ;
glVertex3f(p1.x,p1.y,p1.z) ;
glVertex3f(p2.x,p2.y,p2.z) ;
glEnd() ;
```

- (c) Réactivez l'éclairage

2.2 Eclairage sur un maillage

Connaissant la normale en chaque sommet du maillage vous pouvez dorénavant utiliser la procédure `glNormal3f` avant chaque `glVertex3f` pour leur associer une normale.

1. Visualisez le maillage. Il a perdu toute couleur! En effet, le fait d'activer l'éclairage fait qu'OpenGL ignore maintenant les couleurs données par `glColor3f` et ne se fie qu'au matériau de l'objet.
2. Pour donner un matériau global au maillage, il suffit d'appeler les procédures OpenGL `glMaterialfv` juste avant de dessiner le maillage (comme pour la sphère). Essayez différents paramètres (spécularité, diffusion).
3. Il est bien sûr possible d'affecter un matériau spécifique à chaque sommet du maillage de deux manières différentes :
 - (a) En conservant les `glColor3f` : il suffit d'activer le paramètre `GL_COLOR_MATERIAL` d'OpenGL (décommentez la ligne intéressée dans la procédure `initLights`). Maintenant (et par défaut), chaque appel de `glColor3f` va modifier la composante spéculaire du vertex intéressé, vous devriez donc maintenant retrouver un résultat similaire au TP précédent, avec l'aspect diffus en plus
 - (b) (désactivez `GL_COLOR_MATERIAL`) En appliquant un matériau différent pour chaque vertex en utilisant des `glMaterialfv` avant chaque vertex. Redonnez une couleur diffuse uniforme blanche pour tout le maillage et affectez comme couleur spéculaire pour chaque vertex, la couleur stockée dans la structure "Couleur" du maillage.

3 Pour aller plus loin : sources lumineuses plus complexes

Dans le fichier `main.cpp`, regardez la procédure `moveLight` et décommentez l'évolution de `x` et `y` en fonction du temps. Essayez de changer vous même les paramètres des lumières en fonction du temps.

Cherchez dans un tutoriel OpenGL comment définir une lumière de type spot et testez par exemple l'atténuation de la lumière en fonction de la distance