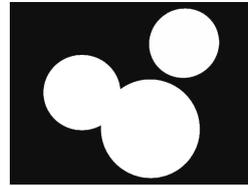


Illumination

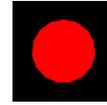
Alexandre Meyer
 Florence Zara
<http://licence-info.univ-lyon1.fr/LIFO41>

1

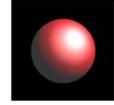
Couleur / éclairage / illumination



Objets sans couleur
ni éclairage



Objet rouge sans
éclairage



Objet rouge avec
éclairage

Comment calculer la couleur et l'éclairage de
chaque partie des objets ?

Cours de synthèse d'images 2

Gestion de la couleur en OpenGL

- Mode RGBA (Rouge, Vert, Bleu, Alpha / transparence)
 - Valeur comprise entre 0.0 et 1.0
- Tampon chromatique :
 - contient la dernière image dessinée
 - On doit le vider en lui attribuant une couleur de fond (couleur de vidage spécifiée une fois)
 - `glClearColor()` : permet de spécifier la couleur de vidage
 - `glClear()` : permet de vider le tampon spécifié en attribuant leur couleur de vidage
 - GL_COLOR_BUFFER_BIT : tampon chromatique
 - GL_DEPTH_BUFFER_BIT : tampon de profondeur
 - GL_STENCIL_BUFFER_BIT : tampon stencil

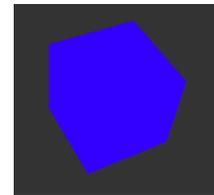
Cours de synthèse d'images 3

Ajout de la couleur en OpenGL

- `glColor()` définit la couleur active
 - Valeur comprise entre 0.0 et 1.1
- Exemple avec l'ajout de la couleur bleu à un cube


```
glColor3f(0, 0, 1);
Draw_Cube();
```

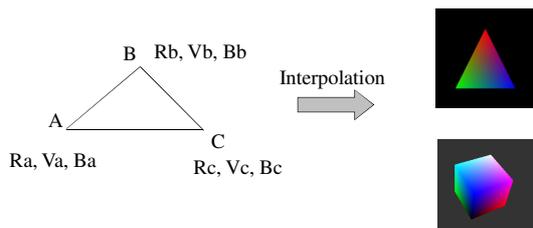
La **couleur est constante** pour
l'ensemble du cube (spécifiée
avant la création du cube)



Cours de synthèse d'images 4

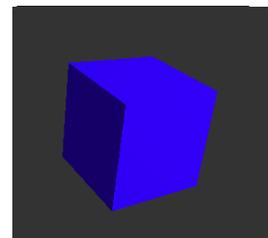
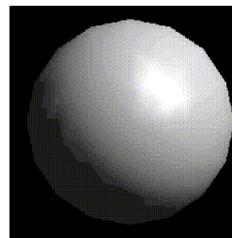
1 couleur par sommet

- `glColor()` est spécifié pour chacun des sommets
- Il est mis avant le `glVertex()`



Cours de synthèse d'images 5

Gestion de la lumière



On aimerait ajouter de
l'éclairage (illumination)

Cours de synthèse d'images 6

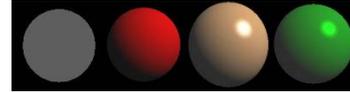
Apport de la lumière

- Selon la lumière, les objets réfléchissent ou n'absorbent pas la même couleur
 - Ils apparaissent ainsi différents (la mer n'a pas la même couleur qd il fait beau ou qd il pleut)
- Couleur de la lumière déterminée par composantes rouge, vert, bleu émises (3 longueurs d'ondes)
- Propriétés du matériel permettent de déterminer les composantes réfléchies et ainsi la couleur de l'objet
 - OpenGL : éclairage dépend des caractéristiques de la lumière et des matériaux

Cours de synthèse d'images 7

Composantes de la lumière

- Ambiante : colore les pixels d'un objet par la même couleur qq soit environnement lumineux
- Diffuse (différente pour chaque sommet) : donne effet 3D et lissé aux objets
- Spéculaire : correspond au reflet de la lumière sur les bords des objets
- Emissive : lumière émise par un objet



Cours de synthèse d'images 8

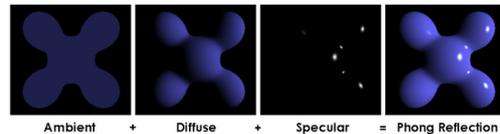
Illumination

- Objectif : déterminer la couleur d'un point de l'objet
- Solution : calculer comment ce point est illuminé

Cours de synthèse d'images 9

Modèle d'illumination de Phong

- Modèle de Phong
 - Modèle local : calcul effectué en chaque point
 - Empirique : plausible mais loin d'une simulation physique
 - Calcule la **lumière réfléchie** par le point étudié
 - Combine 3 éléments : ambiant + diffus + spéculaire



Cours de synthèse d'images 10

Modèle d'illumination de Phong

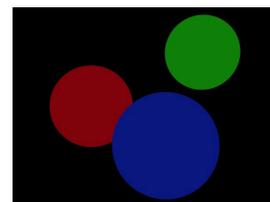
$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \underbrace{\begin{bmatrix} K_a \times \begin{pmatrix} R_a \\ G_a \\ B_a \end{pmatrix} \\ \uparrow \\ \text{couleur de la lumière} \end{bmatrix}}_{\text{Ambiant}} + \underbrace{\begin{bmatrix} \begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \\ \uparrow \\ \text{couleur de l'objet} \end{bmatrix}}_{\text{Diffus}} \times \underbrace{\left(K_d (N \cdot L) + K_s (H \cdot N)^m \right)}_{\text{Spéculaire}} \times \begin{pmatrix} R_o \\ G_o \\ B_o \end{pmatrix}$$

- K : proportion de lumière réfléchie vs absorbée par le matériel

Cours de synthèse d'images 11

Lumière ambiante (Ambient Light)

- Approximation d'illumination globale
 - Même sans lumière directe, un objet est éclairé
 - Constant sur tout l'objet
 - Se traduit par un niveau de gris
 - On ne voit que la silhouette des objets



Cours de synthèse d'images 12

Lumière ambiante (*Ambient Light*)

- Lumière ambiante est définie pour toute la scène
 - Intensité de la lumière ambiante : $I_a = (R_a, G_a, B_a)$
 - Par exemple la couleur du ciel
- Chaque objet réfléchit seulement une portion de cette lumière
 - K_a : proportion de la lumière ambiante réfléchie (valeur entre 0 et 1)
 - Intensité de la lumière ambiante renvoyée par l'objet : $I = I_a K_a$

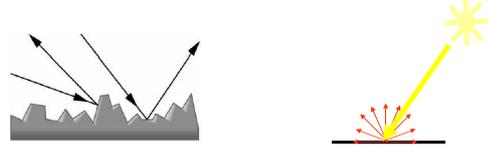
Ambiant

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \left[K_a \times \begin{pmatrix} R_a \\ G_a \\ B_a \end{pmatrix} + \begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \times (K_d (N \cdot L) + K_s (H \cdot N)^m) \right] \times \begin{pmatrix} R_o \\ G_o \\ B_o \end{pmatrix}$$

Cours de synthèse d'images 13

Diffusion de la lumière

- Loi de Lambert



- Lumière réfléchi de manière diffuse (réflexion diffuse)
- Diffusion uniforme dans toutes les directions
- Diffusion dépend de l'angle d'arrivée de la lumière
- Surface de ce type = surface Lambertienne

Cours de synthèse d'images 14

Loi de Lambert

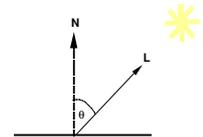
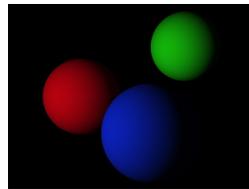


- L = direction vers la lumière
- N = vecteur normal à la surface
- Lumière incidente (arrivant sur la surface) est proportionnelle à d
- d dépend de l'angle d'arrivée de la lumière
 - d est proportionnel à $\cos \theta = N \cdot L$
- ⇒ Intensité réfléchi est proportionnelle à $\cos \theta$

Cours de synthèse d'images 15

Lumière diffuse (*Diffuse Light*)

- La lumière est réfléchi selon la loi de Lambert
- I_i : intensité de la lumière
 - énergie lumineuse par unité de surface
- K_d : coefficient de diffusion de la surface
- Intensité de la lumière diffusée : $I_d = I_i K_d N \cdot L = I_i K_d \cos \theta$



Cours de synthèse d'images 16

Lumière diffuse (*Diffuse Light*)

Diffus

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \left[K_a \times \begin{pmatrix} R_a \\ G_a \\ B_a \end{pmatrix} + \begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \times (K_d (N \cdot L) + K_s (H \cdot N)^m) \right] \times \begin{pmatrix} R_o \\ G_o \\ B_o \end{pmatrix}$$

↑
intensité
de la
lumière

Cours de synthèse d'images 17

Si on a plusieurs lumières ?

- Additionne les termes diffus

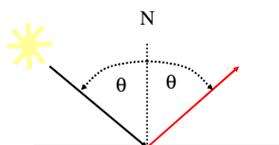
$$I_T = K_a I_a + \sum_{j=1}^m I_j K_d (N \cdot L_j)$$

- m : nombre de sources lumineuses
- I_j : intensité de la lumière incidente j
- L_j : vecteur indiquant la direction vers la lumière j

Cours de synthèse d'images 18

Réflexion spéculaire

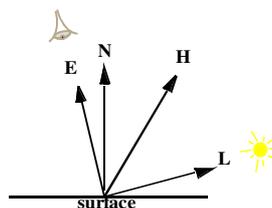
- Prise en compte du fait qu'il y a plus de lumière renvoyée dans la direction de réflexion



- Apparaît sur les surfaces brillantes
- Cas réflexion parfaite
 - On ne verrait pratiquement jamais le reflet spéculaire !

Cours de synthèse d'images 19

Spéculaire « imparfait » (Phong)

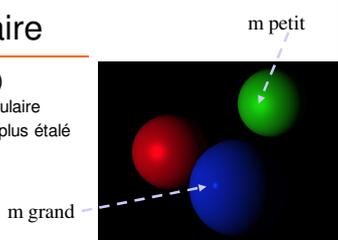
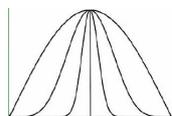


- E = direction de l'œil (Eye)
- N = normal
- L = direction de la lumière
- $H = (E+L) / 2$

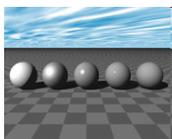
Cours de synthèse d'images 20

Lumière spéculaire

- m = brillance (*shininess*)
 - m grand : petit reflet spéculaire
 - m petit : reflet spéculaire plus étalé



- Intensité de la lumière spéculaire : $I_s = I_i K_s (H \cdot N)^m$



Cours de synthèse d'images 21

Modèle de Phong : ambiant, spéculaire et diffus

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \underbrace{\left[K_a \times \begin{pmatrix} R_a \\ G_a \\ B_a \end{pmatrix} \right]}_{\text{Ambiant}} + \underbrace{\left[G_i \times \frac{I}{\text{Diffus}} + K_s \times (H \cdot N)^m \right]}_{\text{Spéculaire}} \times \begin{pmatrix} R_o \\ G_o \\ B_o \end{pmatrix}$$

Couleur de l'objet

- Si on a plusieurs sources lumineuses :
 - somme les termes diffus et spéculaires

Cours de synthèse d'images 22

Comment on fait en pratique avec des polygones ?

23

Flat shading

- Eclairage d'une face proportionnel à l'angle entre la normale et la direction de la lumière
 - 1 normale par face
 - Valeur de l'intensité uniforme pour toute la face
- Discontinuité d'ombrage aux bords de chaque face



Problème : on voit les facettes !!

Cours de synthèse d'images 24

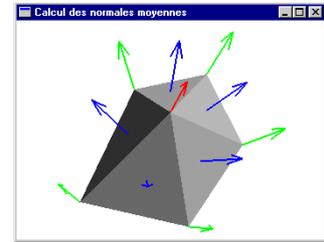
Lissage de Gouraud (*Gouraud shading*)

- Gouraud calcule la couleur
 - $I = \text{ambiant} + \text{diffus}$ (sans terme spéculaire)
 - Pour chaque sommet du polygone
 - Au moment du remplissage, **interpolation bilinéaire des couleurs**
 - fait une interpolation des couleurs le long des arêtes du polygone
 - fait une seconde interpolation lors du remplissage du polygone
- Coûte moins cher que de calculer I pour chaque pixel du polygone

Lissage de Gouraud

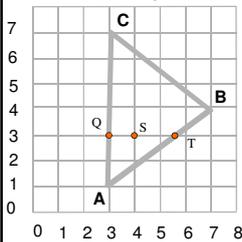
- On calcule une normale par sommet
 - Moyenne des normales des faces adjacentes

Normale rouge = moyenne des normales bleues



Exemple du lissage de Gouraud

- Gouraud calcule la couleur des sommets A, B, C
- Puis la couleur du pixel S est calculée par interpolation bilinéaire.
- Pour composante rouge R_S :



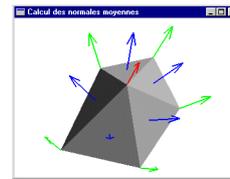
$$R_Q = \frac{y_Q - y_A}{y_C - y_A} R_C + \frac{y_C - y_Q}{y_C - y_A} R_A = \frac{2}{6} R_C + \frac{4}{6} R_A$$

$$R_T = \frac{y_T - y_A}{y_B - y_A} R_B + \frac{y_B - y_T}{y_B - y_A} R_A = \frac{1}{3} R_B + \frac{2}{3} R_A$$

$$R_S = \frac{x_S - x_Q}{x_T - x_Q} R_T + \frac{x_T - x_S}{x_T - x_Q} R_Q = \frac{1}{2.6} R_T + \frac{1.6}{2.6} R_Q$$

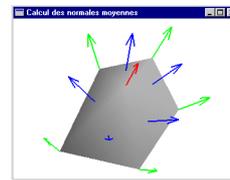
Idem pour G_S et B_S

Flat vs Gouraud shading



Flat shading :

- 1 normale par face
- Couleur uniforme sur la face
- Discontinuité aux bords des faces



Lissage de Gouraud :

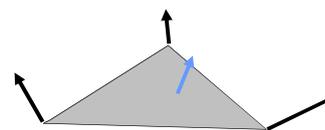
- 1 normale par sommet
- Calcul de la couleur aux sommets
- Interpolation de la couleur pendant le remplissage de la face

Lissage de Gouraud

- Problème
 - Ne tient pas compte de la réflexion spéculaire
 - Pas de tâche spéculaire
 - Couleur non constante avec la rotation des points

Phong Shading

- Phong calcule la couleur en chaque pixel
 - $I = \text{ambiant} + \text{diffus} + \text{spéculaire}$
 - Pour trouver la normale en chaque pixel, **interpolation des normales** au moment du remplissage



La normale bleue est une interpolation des normales noires

- Recapture la tâche spéculaire au centre du polygone
- Inclus la composante spéculaire
- Possible en hard avec les Pixel shader (depuis ~2000)

Flat vs Gouraud vs Phong Shading



Gauche : pas d'interpolation (1 normale par surface) Droite : Phong (interpolation des normales)

Milieu : Gouraud (Interpolation des couleurs) Cours de synthèse d'images 31

Conclusion

- Couleur d'un pixel
 - Ambient
 - Diffuse
 - Spéculaire
 - Il peut y avoir plusieurs sources de lumière
- Besoin
 - Ombres
 - Modèle plus réaliste de réflexion (BRDF)
 - Illumination globale

Cours de synthèse d'images 32

Eclairage en OpenGL

- Définir une source de lumière

```
glEnable(GL_LIGHTING); // Active éclairage
glEnable(GL_LIGHT0); // 8 lumières possibles : 0, ..., 8
```

```
// Définition de la position de la lumière 0
GLfloat light_position[] = { 1.0, 1.0, 1.0, 0.0 };
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
```

Rq : La couleur de la source vaut 1,1,1 par défaut

Cours de synthèse d'images 33

Eclairage en OpenGL

- Donner une couleur diffuse et spéculaire à un objet

```
// Définition des propriétés du matériel
GLfloat mat_diffus[] = { 1.0, 1.0, 1.0, 1.0 };
GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0 };
GLfloat mat_shininess[] = { 50.0 };
```

```
glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUS, mat_diffus);
glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, mat_specular);
glMaterialfv(GL_FRONT_AND_BACK, GL_SHININESS, mat_shininess);
```

// GL_FRONT_AND_BACK : indique faces où la matière doit être appliquée

```
Draw_Cube();
```

Cours de synthèse d'images 34