



OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination
Lumières
Matériaux

Texturation

États

Résultat

OpenGL

Apparence et Ombrage

Didier Verna

didier@lrde.epita.fr
<http://www.lrde.epita.fr/~didier>

Version ENST – VIHM du 23 février 2009



License d'exploitation

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination
Lumières
Matériaux

Texturation

États

Résultat

Ce document est mis à votre disposition sous un contrat de license Creative Commons – Common Deed. Par le téléchargement ou la consultation de ce document, l'utilisateur accepte les conditions d'utilisation décrites par cette license et s'engage à la respecter intégralement.

La licence confère à l'utilisateur un droit d'usage sur le document consulté ou téléchargé, totalement ou en partie, dans les conditions définies ci-après et à l'exclusion expresse de toute utilisation commerciale.

Le droit d'usage défini par la licence autorise un usage à destination de tout public qui comprend:

- le droit de reproduire tout ou partie du document sur support informatique ou papier,
- le droit de diffuser tout ou partie du document au public sur support papier ou informatique, y compris mise à la disposition du public sur un réseau numérique.

Aucune modification du document dans son contenu, sa forme ou sa présentation n'est autorisée. Les mentions relatives à la source du document et/ou à son auteur doivent être conservées dans leur intégralité.

Le droit d'usage défini par la licence est personnel, non exclusif et non transmissible. Tout autre usage que ceux prévus par la licence est soumis à autorisation préalable et expresse de l'auteur.





Table des matières

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination
Lumières
Matériaux

Texturation

États

Résultat

- 1 Ombrage Simple
- 2 Ombrage avancé
 - Création de sources lumineuses
 - Spécification de matériaux
- 3 Texturation
- 4 Gestion des états OpenGL
- 5 Résultat



Ombrage simple

Caractéristiques

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

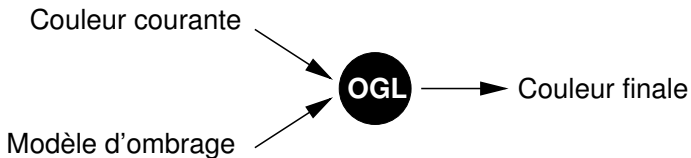
Lumières
Matériaux

Texturation

États

Résultat

- Aucune notion de lumière
- Couleur « courante » associée à chaque vertex *explicite*
- Modèles d'ombrage :
 - ▶ **Ombrage plat** : une seule couleur par primitive
 - ▶ **Ombrage de Gouraud** : interpolation entre chaque couleur explicite





Fonctions OpenGL

Couleur et modèle d'ombrage

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières
Matériaux

Texturation

États

Résultat

```
void glColor3* (TYPE red, TYPE green, TYPE blue);
```

```
void glColor4* (TYPE red, TYPE green, TYPE blue,
```

```
TYPE alpha);
```

```
void glColor{34}*v (const TYPE *v);
```

Spécifie la couleur RGB(A) courante de tracé.

```
void glShadeModel (GLenum mode);
```

Spécifie le modèle d'ombrage.

mode : GL_FLAT, GL_SMOOTH (défaut)



Sélection de couleur en ombrage plat

Quel vertex de référence ?

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières
Matériaux

Texturation

États

Résultat

Type	Vertex
lignes	2
GL_POLYGON	1
GL_TRIANGLES	3i
GL_TRIANGLE_STRIP	i + 2
GL_TRIANGLE_FAN	i + 2
GL_QUAD_STRIP	2i + 2
GL_QUADS	4i



OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières
Matériaux

Texturation

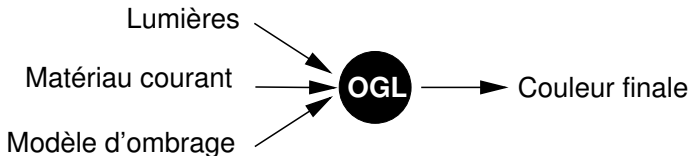
États

Résultat

■ Modélisation physique de la lumière :

- ▶ Conditions d'éclairage (sources lumineuses)
- ▶ Matériaux des objets (réactions à la lumière)

■ Modèle d'ombrage : identique à l'ombrage simple





Illumination

Couleur = Matériau + Lumière

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières
Matériaux

Texturation

États

Résultat

■ **Modèle de Phong** : $I = I_a + I_d + I_s$

■ **Lumière ambiante** : $C_a = M_a I_a$

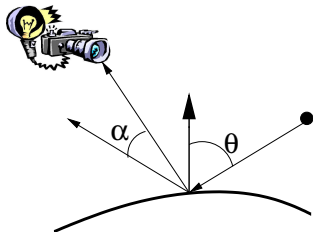
Isotropique

■ **Lumière diffuse** : $C_d = M_d I_d \cos \theta$

Unidirectionnelle, réflexion isotropique (mat)

■ **Lumière spéculaire** : $C_s = M_s I_s \cos^n \alpha$

Unidirectionnelle, réflexion préférentielle dans une direction (brillant)





Modèles d'ombrage et lumière

Plat, Gouraud, Phong

OpenGL

Didier Verna
ENST
VIHM

Colorisation

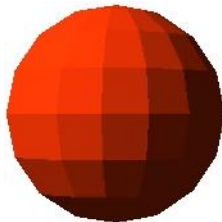
Illumination

Lumières
Matériaux

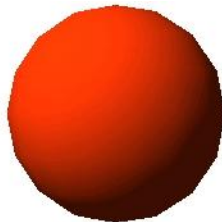
Texturation

États

Résultat



Flat



Gouraud

- **Plat** : une normale (donc une couleur) par primitive
- **Gouraud** : interpolation linéaire des intensités
- **Phong** : interpolation linéaire des normales



Gestion des normales

Essentiel pour un illumination correcte

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières
Matériaux

Texturation

États

Résultat

```
void glNormal3{bsidf} (TYPE x, TYPE y, TYPE z);
```

```
void glNormal3{bsidf}v (const TYPE *v);
```

Spécifie le vecteur normal courant.

```
glEnable / glDisable (GL_NORMALIZE);
```

Auto-normalisation des normales.



Création de sources de lumière

Fonction générale

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières

Matériaux

Texturation

États

Résultat

```
void glLight{ if }[v] (GLenum light, GLenum pname,  
TYPE pvalue);
```

Spécifie la valeur `pvalue` d'un paramètre `pname` pour la source de lumière `light`.

light : `GL_LIGHT0 ... GL_LIGHT7` au minimum



Spécification de couleur

Ambiante, diffuse, spéculaire

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières

Matériaux

Texturation

États

Résultat

pname : GL_AMBIENT, GL_DIFFUSE **et** GL_SPECULAR

pvalue : (R, G, B, A)

■ Défaut :

- ▶ Ambiante : $(0, 0, 0, 1)$
- ▶ Diffuse et spéculaire : $(1, 1, 1, 1)$ puis $(0, 0, 0, 1)$

■ Utiliser les versions vectorielles de `glLight*`



Position et atténuation

Une position à l'infini est une direction

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières

Matériaux

Texturation

États

Résultat

■ Position :

pname : GL_POSITION

pvalue : (x, y, z, w)

- ▶ Lumière positionnelle (w non nul) : émission isotropique
- ▶ Lumière directionnelle (w nul) : direction $-(x, y, z)$
- ▶ Défaut : $(0, 0, 1, 0)$
- ▶ **La position des sources dépend du mode Modelview !!**

■ Atténuation :

pname : GL_ [CONSTANT | LINEAR | QUADRATIC]
_ATTENUATION

pvalue : TYPE

- ▶ Défaut : $(1, 0, 0)$
- ▶ Lumières positionnelles uniquement



Spots (projecteurs)

Lumière positionnelle à diffusion non isotropique (conique)

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières
Matériaux

Texturation

États

Résultat

■ *pname* : GL_SPOT_DIRECTION

pvalue : (x,y,z)

▶ Défaut : (0,0,-1)

■ *pname* : GL_SPOT_CUTOFF

pvalue : TYPE

▶ Valeurs : [0,90] et 180

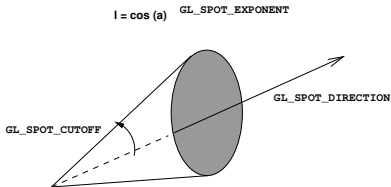
▶ Défaut : 180 (degrés)

■ *pname* : GL_SPOT_EXPONENT

pvalue : TYPE

▶ Valeurs : [0,128]

▶ Défaut : 0





Paramètres généraux

Non spécifiques à chaque source

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières

Matériaux

Texturation

États

Résultat

```
void glLightModel{if}[v] (GLenum pname, TYPE pvalue);
```

■ Lumière ambiante globale :

pname : GL_LIGHT_MODEL_AMBIENT

pvalue : composantes RGBA

- ▶ Défaut : (0.2,0.2,0.2,1.0)

■ Gestion des faces arrières :

pname : GL_LIGHT_MODEL_TWO_SIDE

pvalue : GL_TRUE, GL_FALSE

- ▶ Défaut : GL_FALSE

■ Calcul des reflets spéculaires :

pname : GL_LIGHT_MODEL_LOCAL_VIEWER

pvalue : GL_TRUE, GL_FALSE

- ▶ Défaut : GL_FALSE



(Dés) Activation des lumières

Ne pas oublier !

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières

Matériaux

Texturation

États

Résultat

```
glEnable / glDisable (GL_LIGHTING);
```

Sélection entre ombrage simple et avancé.

```
glEnable / glDisable (GL_LIGHT0 | ... | GL_LIGHT7);
```

Sélection individuelle de source lumineuse.



Spécification de matériaux

Fonction générale

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières

Matériaux

Texturation

États

Résultat

```
void glMaterial{ if }[v] (GLenum face, GLenum pname,  
TYPE pvalue);
```

Spécifie la valeur `pvalue` d'un paramètre `pname` pour la face `face`.

face : GL_FRONT, GL_BACK, GL_FRONT_AND_BACK



Réaction aux lumières

Ambiante, diffuse, spéculaire

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières

Matériaux

Texturation

États

Résultat

■ Réaction aux composantes :

pname : GL_AMBIENT, GL_DIFFUSE,
GL_AMBIENT_AND_DIFFUSE, GL_SPECULAR

pvalue : (R, G, B, A)

- ▶ Ambiante : $(0.2, 0.2, 0.2, 1.0)$
- ▶ Diffuse : $(0.8, 0.8, 0.8, 1.0)$
- ▶ Spéculaire : $(0, 0, 0, 1)$

■ Intensité des reflets spéculaires :

pname : GL_SHININESS

pvalue : $[0, 128]$

- ▶ Défaut : 0



Lumière « émissive »

Lumière virtuelle, locale à un objet

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières

Matériaux

Texturation

États

Résultat

pname : GL_EMISSION

pvalue : (R, G, B, A)

- Défaut : $(0, 0, 0, 1)$



Technique alternative

Pistage de couleur

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières

Matériaux

Texturation

États

Résultat

void glColorMaterial (GLenum face, GLenum mode);

Piste les valeurs spécifiées par glColor* et met à jour automatiquement le(s) composante(s) correspondante(s) du matériau.

face : GL_FRONT, GL_BACK, GL_FRONT_AND_BACK

mode : GL_AMBIENT, GL_DIFFUSE,

GL_AMBIENT_AND_DIFFUSE, GL_SPECULAR,

GL_EMISSION

glEnable / glDisable (GL_COLOR_MATERIAL);

(Dés) Activation du pistage.



Plaquage de texture

Application d'images sur des surfaces

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination
Lumières
Matériaux

Texturation

États

Résultat

■ Utilisation :

- ▶ Plus de réalisme : textures photographiques marbre, bois *etc.*
- ▶ Moins de polygones : relief simulé murs de briques *etc.*

■ Caractéristiques :

- ▶ Plaquage sur points, lignes, polygones
- ▶ Textures uni, bi ou tridimensionnelles
- ▶ Apparence directe ou combinée

■ Fonctionnement :

- ▶ Association « texel » / vertex
- ▶ Distorsion géométrique

■ (Dés) Activation :

```
glEnable / glDisable (GL_TEXTURE_2D);
```



Avec ou sans texture

Vous préférez quoi ?

OpenGL

Didier Verna
ENST
VIHM

Colorisation

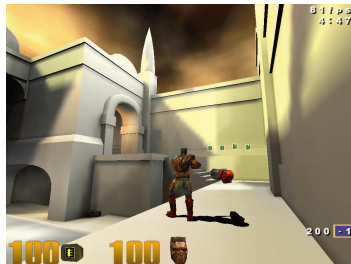
Illumination

Lumières
Matériaux

Texturation

États

Résultat



Quake III Arena



Spécification de texture

Fonction générale

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination
Lumières
Matériaux

Texturation

États

Résultat

```
void glTexImage2D (GLenum target, GLint level,  
GLint components, GLsizei width, GLsizei height,  
GLint border, GLenum format, GLenum type,  
const GLvoid *pixels);
```

Défini une texture bidimensionnelle.

target : GL_TEXTURE_2D

level : 0 (niveau de détail)

components : 0 ... 4, GL_RGBA *etc.*

width height : 2^n , $n_{max} \geq 6$

border : 0

format : GL_RGB, GL_RGBA *etc.*

type : GL_BYTE, GL_UNSIGNED_BYTE *etc.*



Alignement du pack de données

Restons portables

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination
Lumières
Matériaux

Texturation

États

Résultat

```
void glPixelStore{ if } (GLenum pname, TYPE pvalue);
```

pname : GL_UNPACK_ALIGNMENT

pvalue : 1,2,4,8 (défaut : 4)



OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination
Lumières
Matériaux

Texturation

États

Résultat

■ Association texel / vertex

```
void glTexCoord{1234}{sifd}{v} (TYPE coords);
```

Spécifie les coordonnées courantes de texture dans $[0, 1]$.

■ Extension de texture : (hors $[0, 1]$)

```
void glTexParameteri (GLenum target,
```

```
GLenum pname, TYPE pvalue);
```

Spécifie le mode d'extension de texture.

target : GL_TEXTURE_2D

pname : GL_TEXTURE_WRAP_T,
GL_TEXTURE_WRAP_S

pvalue : GL_CLAMP, GL_REPEAT



Filtrage de textures

Quand il y a distortion

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

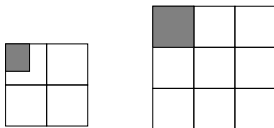
Lumières
Matériaux

Texturation

États

Résultat

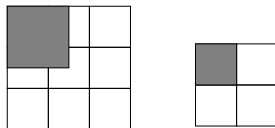
Magnification



Texture

Polygone

Minification



Texture

Polygone

- **Mimapping** : niveaux de détail (LOD)
- **Point sampling** : texel le plus proche
- **Bilinéaire** : interpolation sur 4 texels
- **Trilinéaire** : interpolation inter-LOD
- **Anisotropique** : prise en compte des effets d'angle (32 texels)



Filtrage de textures

Exemples

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières
Matériaux

Texturation

États

Résultat





Filtrage de texture

Le cas d'OpenGL

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination
Lumières
Matériaux

Texturation

États

Résultat

```
void glTexParameteri (GLenum target, GLenum pname,  
TYPE pvalue);
```

Spécifie le mode de filtrage de texture.

target : GL_TEXTURE_2D

pname : GL_TEXTURE_MAG_FILTER,

GL_TEXTURE_MIN_FILTER

pvalue : GL_NEAREST, GL_LINEAR



OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination
Lumières
Matériaux

Texturation

États

Résultat

```
void glTexEnv{if}{v} (GLenum target, GLenum pname,  
TYPE pvalue);
```

Spécifie le mode de calcul d'apparence avec texture.

target : GL_TEXTURE_ENV

pname : GL_TEXTURE_ENV_MODE

pvalue : GL_REPLACE, GL_MODULATE



Gestion des états OpenGL

Analogie avec les piles de matrices

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination
Lumières
Matériaux

Texturation

États

Résultat

```
void glPushAttrib ( GLbitfield mask);
```

```
void glPopAttrib (void);
```

Sauvegarde / restauration d'états.

- *mask* : GL_ENABLE_BIT, GL_LIGHTING_BIT *etc.*



Résultat du TP

OpenGL

Didier Verna
ENST
VIHM

Colorisation

Illumination

Lumières
Matériaux

Texturation

États

Résultat

