



OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

# OpenGL

## Effets Spéciaux

Didier Verna

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

Version ENST – VIHM du 3 septembre 2008



# License d'exploitation

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

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

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

1 Fog

2 Blending

3 Ombres portées

4 Réflexions

5 Le stencil buffer

6 Résultat



# Fog

Un effet qui ne coûte pas cher

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

- **Utilisation** : Simulation d'effets atmosphériques (brouillard, pollution *etc.*)
- **Caractéristiques** :
  - ▶ Uniformisation de l'apparence avec l'éloignement
  - ▶ Limitation effective de la visibilité (d'où optimisations)
- **Fonctionnement** :
  - ▶ Contrôle de la densité
  - ▶ Contrôle de la couleur
  - ▶ Application après transformations, lumières, textures
- **(Dés) Activation** :  
`glEnable / glDisable (GL_FOG);`



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

Spécifie tous les paramètres définissant le fog.

### ■ Couleur :

*pname* : GL\_FOG\_COLOR

*pvalue* : (R, G, B, A)

### ■ Équation :

*pname* : GL\_FOG\_MODE

*pvalue* : GL\_LINEAR, GL\_EXP, GL\_EXP2

### ■ Paramétrage :

*pname* : GL\_FOG\_DENSITY, GL\_FOG\_START,  
GL\_FOG\_END

*pvalue* : valeur correspondante pour l'équation de fog



# Équations de fog

## Sémantique des paramètres

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

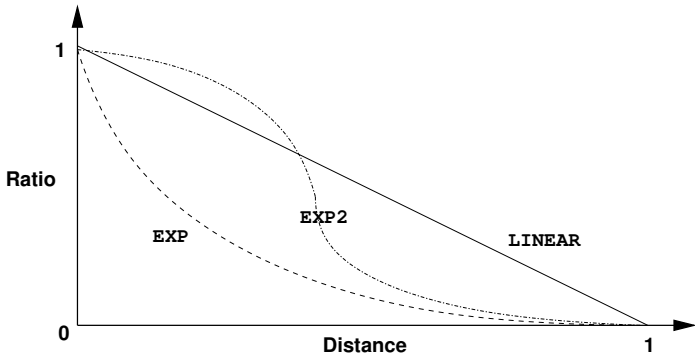
Stencil

Résultat

$$\text{GL\_LINEAR} : f = \frac{\text{end}-z}{\text{end}-\text{start}}$$

$$\text{GL\_EXP} : f = e^{-(\text{density} \cdot z)}$$

$$\text{GL\_EXP2} : f = e^{-(\text{density} \cdot z)^2}$$





# Blending

La vérité sur  $\alpha$  enfin dévoilée !

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

## ■ Utilisation :

- ▶ Simulation de transparence / translucidité
- ▶ Rendu non rectangulaire de texture (« billboarding »)

## ■ Caractéristiques :

- ▶ Combinaison entre source (fragment) et destination (frame buffer)
- ▶ Fonction de combinaison paramétrable par les composantes  $\alpha$
- ▶ Application *après* rasterisation

## ■ Fonctionnement :

- ▶  $S_r + D_r \Rightarrow B_{sr}.S_r + B_{dr}.D_r$   
 $S_g + D_g \Rightarrow B_{sg}.S_g + B_{dg}.D_g$   
 $S_b + D_b \Rightarrow B_{sb}.S_b + B_{db}.D_b$
- ▶ La composante alpha est un facteur d'*opacité*



# Fonction OpenGL

Exemples de paramètres

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

## ■ Paramétrisation :

```
void glBlendFunc (GLenum src, GLenum dst);
```

Spécifie les paramètres définissant le blending.

Constante	Facteur B
GL_ZERO	$(0, 0, 0, 0)$
GL_ONE	$(1, 1, 1, 1)$
GL_SRC_ALPHA	$(S_{ar}, S_{ag}, S_{ab}, S_{aa})$
GL_DST_ALPHA	$(D_{ar}, D_{ag}, D_{ab}, D_{aa})$
GL_ONE_MINUS_SRC_ALPHA	$(1 - S_{ar}, 1 - S_{ag}, 1 - S_{ab}, 1 - S_{aa})$
GL_ONE_MINUS_DST_ALPHA	$(1 - D_{ar}, 1 - D_{ag}, 1 - D_{ab}, 1 - D_{aa})$

## ■ (Dés) Activation :

```
glEnable / glDisable (GL_BLEND);
```





# Exemples

Pour fixer les idées

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

## ■ Mélange de deux images à 50% :

- ▶ Tracer l'image de départ sans blending
- ▶ Tracer la deuxième image avec `src` et `dst` à `GL_SRC_ALPHA`, et  $\alpha = 0.5$

## ■ Mélange de deux images à 75/25% :

- ▶ Tracer l'image de départ sans blending
- ▶ Tracer la deuxième image avec `src` à `GL_SRC_ALPHA`, `dst` à `GL_ONE_MINUS_SRC_ALPHA`, et  $\alpha = 0.25$

## ■ Application aux ombres portées, reflets *etc.*



# Problèmes / Précautions

C'était trop simple...

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

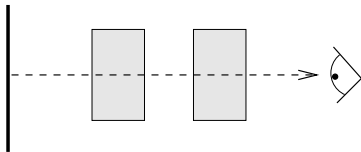
Ombres

Réflexions

Stencil

Résultat

- **Utilisation des paramètres `*_DST_*` :**
  - ▶ Nécessite un color buffer de type RGBA
  - ▶ Pas disponible partout (jamais en émulation !)
- **Ordre de tracé :**
  - ▶ La couleur finale en dépend
- **Depth Buffer :**
  - ▶ Certains objets cachés doivent être tracés
  - ▶ Tracer les objets translucides *après* les autres
  - ▶ Utiliser un z-buffer en lecture seule





# Projection d'ombres

Ombres portées  $\neq$  Ombrage

OpenGL

Didier Verna  
ENST  
VIHM

Fog

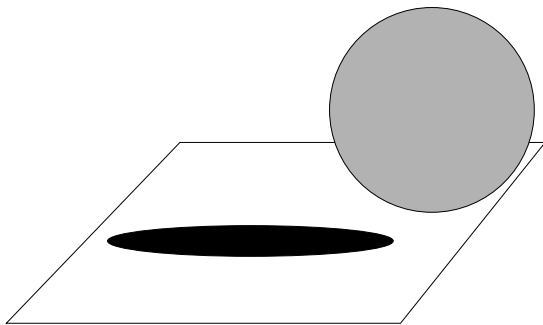
Blending

Ombres

Réflexions

Stencil

Résultat



- Une ombre portée est la trace d'un objet à travers une *projection*.
- Idée : tracer deux fois le même objet, dont une à travers une matrice de projection, mais en mode `Modelview!!`



# Projection directionnelle sur un plan

Un peu de maths, pour une fois...

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

■ **Équation d'un plan** :  $a.x + b.y + c.z + d = 0$

■ **Direction de projection** :  $(d_x, d_y, d_z)$

■ **Principe** : pour tout point  $(S_x, S_y, S_z)$  de l'espace Euclidien, trouver l'intersection de la droite  $S + \alpha \vec{D}$  avec le plan concerné.

$$P = \begin{bmatrix} b.d_y + c.d_z & -b.d_x & -c.d_x & -d.d_x \\ -a.d_y & a.d_x + c.d_z & -c.d_y & -d.d_y \\ -a.d_z & -b.d_z & a.d_x + b.d_y & -d.d_z \\ 0 & 0 & 0 & a.d_x + b.d_y + c.d_z \end{bmatrix}$$

■ **Coordonnées homogènes** : OpenGL n'apprécie pas les  $w < 0$



OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

## ■ Apparence des ombres portées :

- ▶ Devrait dépendre des conditions d'éclairage
- ▶ **Objets opaques** : faire du blending avec  
`src = GL_ZERO` et `dst = GL_SRC_ALPHA`
- ▶ **Objets translucides** : blending assombri entre le plan et l'objet projeté

## ■ Collision ombres / plan :

- ▶ Surélévation
- ▶ Offset

## ■ Débordement des ombres :

- ▶ Limitation du tracé
- ▶ Stencil Buffer



# Réflexions d'objets

Un effet qui demande réflexion

OpenGL

Didier Verna  
ENST  
VIHM

Fog

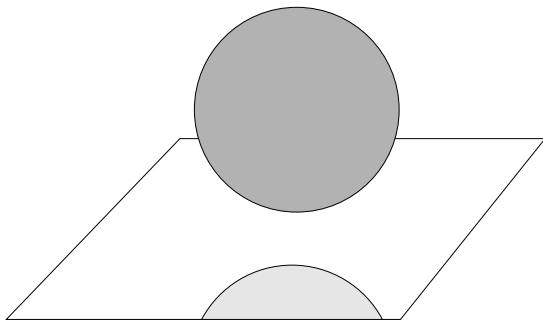
Blending

Ombres

Réflexions

Stencil

Résultat



- Une réflexion est la trace d'un objet à travers une *symétrie*.
- Idée : tracer deux fois le même objet, dont une à travers une matrice symétrisant l'objet par rapport au plan.



# Problèmes / Précautions

C'était trop simple...

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

## ■ Symétrisation horizontale :

- ▶ `glScalef (1.0, -1.0, 1.0) ;`
- ▶ Inversion des faces des polygones (attention au culling)

## ■ Apparence des reflets :

- ▶ Renversement des lumières
- ▶ Blending entre les reflets et le plan

## ■ Débordement des reflets :

- ▶ Limitation du tracé
- ▶ Stencil Buffer



# Le Stencil Buffer

Un troisième buffer d'OpenGL

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

## ■ Utilisation typique :

- ▶ limitation du tracé (masquage de régions)

## ■ Caractéristiques :

- ▶ Conditionnement du tracé par le « stencil test »

## ■ Fonctionnement :

- ▶ Buffer contenant des *entiers* (nombre de bits variable)
- ▶ Une *fonction* paramétrable pour le test
- ▶ une *opération* paramétrable pour la modification du buffer





# Fonction et Opération de stencil

Les deux mamelles du dessin

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

**void** glStencilFunc (GLenum func, GLint ref, GLuint mask);

Spécifie la fonction de test, qui compare la valeur de référence avec la valeur actuelle dans le stencil buffer, mais seulement sur les bits du mask.

*func* : GL\_NEVER, GL\_ALWAYS, GL\_EQUAL, GL\_LESS, GL\_LEQUAL *etc.*

**void** glStencilOp (GLenum fail, GLenum zfail,

GLenum zpass);

Spécifie les opérations de modification du buffer, quand le test échoue, quand le z-test échoue, et quand le z-test réussit.

*fail, zfail, zpass* : GL\_KEEP, GL\_ZERO, GL\_REPLACE *etc.*



# (Dés) Activation et nettoyage du stencil buffer

Toujours le même refrain

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

## ■ Requête :

- ▶ **GLX** : `glXChooseVisual()` avec `GLX_STENCIL_SIZE`
- ▶ **GLUT** : `glutInitDisplayMode()` avec `GLUT_STENCIL`

## ■ (Dés) Activation :

`glEnable / glDisable (GL_STENCIL_TEST);`

## ■ Nettoyage :

`void glClearStencil (GLint val);`

Spécifie la valeur de nettoyage du stencil buffer.

`glClear (GL_STENCIL_BUFFER_BIT);`

Nettoie le stencil buffer.



# Application (ombres et réflexions)

Un scénario possible

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

## 1 Tracer le plan dans le stencil buffer :

- ▶ Valeur de référence à 1
- ▶ Le test de stencil doit **toujours** réussir : fonction `GL_ALWAYS`
- ▶ La valeur de stencil doit être remplacée : opération `GL_REPLACE`

## 2 Tracer les reflets :

- ▶ **Test de stencil** : est-on sur le plan ? `GL_EQUAL`
- ▶ **Opération de stencil** : conservation : `GL_KEEP`

## 3 Tracer le plan normalement

## 4 Tracer les ombres (même combat que pour les réflexions)

## 5 Tracer les objets réels



# Dernières considérations

La touche finale

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

- **Problème** : *ne pas* tracer le plan dans le color buffer quand on le trace dans le stencil buffer

- ▶ **Solution** :

```
void glDrawBuffer (GLenum mode);
```

Spécifie le mode de tracé dans le(s) color buffer(s).

*mode* : GL\_NONE, GL\_BACK, GL\_FRONT, *etc.*

- **Améliorations** : pendant le tracé du plan dans le stencil buffer, pas besoin de depth test, de lumière, d'ombrage *etc.*



# Résultat du TP

OpenGL

Didier Verna  
ENST  
VIHM

Fog

Blending

Ombres

Réflexions

Stencil

Résultat

