



OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

# OpenGL

## Introduction

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

Généralités

Syntaxe

États

Intégration

Animation

Ressources

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

Généralités

Syntaxe

États

Intégration

Animation

Ressources

- 1 Généralités
- 2 Syntaxe
- 3 OpenGL : une machine à états
- 4 Intégration avec le monde extérieur
- 5 Animation de scènes
- 6 Ressources



OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

### ■ **Les problèmes :**

- ▶ Répondre au besoin de normalisation
- ▶ Besoins différents selon les applications

### ■ **Les solutions :**

#### ▶ **Interfaces de programmation :**

- Bas niveau : OpenGL, DirectX (Direct3D)...
- Plus haut niveau : RenderMan Interface Specification (Pixar), Open Inventor...

#### ▶ **Langages de description :**

- Plus ou moins haut niveau : VRML (Virtual Reality Modelling Language)

#### ▶ **Formats spécifiques :**

- 3DSMax...

■ **Remarque :** ne pas confondre spécification et implémentation !



# Qu'est-ce qu'OpenGL ?

## Définition

OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

### ■ Techniquement :

- ▶ Interface logicielle pour le matériel sous-jacent
- ▶ API (Application Programming Interface) 2D ou 3D
- ▶ Bibliothèque de fonctions (+/-150)

### ■ Un standard ouvert :

- ▶ Maintenu par l'ARB (Architecture Review Board)
- ▶ Mécanisme d'extensions éventuellement incluses dans les versions ultérieures
- ▶ Évolutif : version 2.0
- ▶ Objectif d'efficacité et d'indépendance vis-à-vis du matériel
- ▶ Protocole spécifique de transport de requêtes



# Qu'est-ce n'est / n'a pas OpenGL ?

Anti-définition

OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

## ■ OpenGL n'est pas :

- ▶ Un langage de programmation
- ▶ Un langage de description
- ▶ Une API de haut niveau

## ■ OpenGL n'a pas :

- ▶ Un système de fenêtrage
- ▶ Un mécanisme d'interaction avec l'utilisateur



# Rendu infographique

## Les 3 composantes de la synthèse d'image

OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

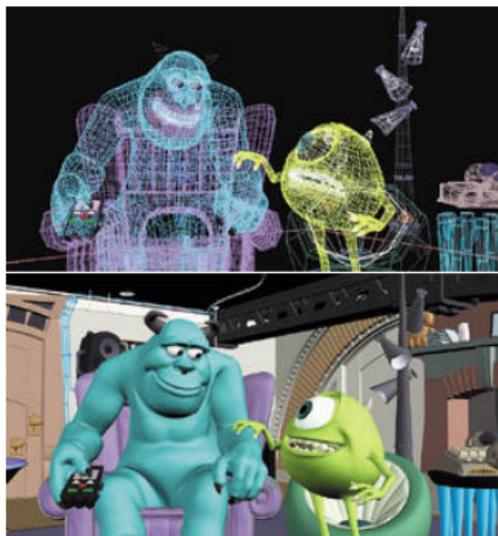
Intégration

Animation

Ressources

### ■ 3 composantes au départ...

- ▶ Géométrie
- ▶ Apparence
- ▶ Point de vue



*Monstres et Cie*

### ■ ... une image (grille de pixels) à la fin.



# Qu'est-ce que fait OpenGL ?

Concrètement

OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

- **Géométrie** : Définition d'objets (primitives : points, lignes, polygones), Positionnement spatial (translation, rotation, homothétie. . .)
- **Apparence** : Couleurs explicites, Matériaux / lumières / textures
- **Point de vue** : Volume de projection (orthographique, perspective)
- **Rasterization** : transformation de la scène en pixels à l'écran (Viewport)

Avec différents modes de tracé et d'ombrage, élimination des parties cachées, blending, antialiasing, brouillard. . .



# Un mini programme OpenGL

Hello 3D World !

OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

```
#include <stuff.h>

int main (int argc, char *argv[])
{
    gimme_a_window_please_ok_thanks ();           /* call to window system */

    glClearColor (0.0, 0.0, 0.0, 1.0);           /* set clear color to black */
    glClear (GL_COLOR_BUFFER_BIT);              /* clear the drawing area */
    glColor3f (1.0, 1.0, 1.0);                  /* set drawing color to white */
    glOrtho (-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);  /* set orthographic projection */
    glBegin (GL_POLYGON);                       /* start object description */
    {
        glVertex2f (-0.5, -0.5);                 /* define polygon by edges */
        glVertex2f (-0.5, 0.5);
        glVertex2f (0.5, 0.5);
        glVertex2f (0.5, -0.5);
    }
    glEnd ();                                   /* end object description */
    glFlush ();                                 /* ensure commands are actually
                                                executed */

    could_you_now_display_all_this_stuff ();    /* call to window system */

    return 0;
}
```



OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

### ■ **Commandes :**

- ▶ préfixées par **gl**,
- ▶ première lettre de chaque mot en majuscule

### ■ **Constantes :**

- ▶ préfixées par **GL**
- ▶ en lettres majuscules

### ■ **Suffixe ANSI C :**

- ▶ indique le nombre d'arguments
- ▶ indique leur type
- ▶ un **v** final indique un pointeur



# Suffixes ANSI C

À langage boîteux...

OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

Suffixe	Type	Type C	Type OpenGL
b	entier 8 bits	signed char	GLbyte
s	entier 16 bits	short	GLshort
i	entier 32 bits	long	GLint
f	flottant 32 bits	float	GLfloat
d	flottant 64 bits	double	GLdouble
ub	entier non signé 8 bits	unsigned char	GLubyte
us	entier non signé 16 bits	unsigned short	GLushort
ul	entier non signé 32 bits	unsigned long	GLuint



# OpenGL : une machine à états

La grande force d'OpenGL

OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

OpenGL se trouve dans différents *modes* ou *états* valables jusqu'à leur prochain changement (*i.e.* couleur et mode de tracé, projection, acquisition de formes...).

- **États booléens** : `glEnable()`, `glDisable()`
- **Requêtes d'état courant** : `glGetBooleanv()`, `glGetIntegerv()`...
- **Requêtes spécifiques** : `glGetLight*()`, `glGetError()`...
- **Pile d'états** : `glPushAttrib()`, `glPopAttrib()`, `glPushMatrix()`, `glPopMatrix()`



# OpenGL versus Direct 3D

Sans commentaire...

OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

```
glBegin (GL_TRIANGLES);  
glVertex3f (0,0,0);  
glVertex3f (1,1,0);  
glVertex3f (2,0,0);  
glEnd ();
```

```
v = &buffer.vertices[0];  
v->x = 0;  
v->y = 0;  
v->z = 0;  
  
v = &buffer.vertices[1];  
v->x = 1;  
v->y = 1;  
v->z = 0;  
  
v = &buffer.vertices[2];  
v->x = 2;  
v->y = 0;  
v->z = 0;  
  
c = &buffer.commands;  
c->operation = DRAW_TRIANGLE;  
c->vertices[0] = 0;  
c->vertices[1] = 1;  
c->vertices[2] = 2;  
  
IssueExecuteBuffer (buffer);
```



# Intégration OpenGL

OpenGL ne s'utilise pas tout seul

OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

- **OpenGL Utility Library (GLU)** : outils de plus haut niveau (transformation de coordonnées, rendu de sphères, cylindres, disques, NURBS, tessellation...)
- **Open Inventor** : toolkit orienté objet de plus haut niveau (formes complexes prédéfinies, gestion de l'interaction, import/export vers d'autres formats...)
- **Intégration aux systèmes de multifenêtrage** : GLX (**g1x**) pour X Windows, WGL (**wg1**) pour Windows, PGL (**pg1**) pour IBM OS/2 Presentation Manager
- **OpenGL Utility Library (GLUT)** : couche opaque d'abstraction indépendante des systèmes de multifenêtrage



OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

- **Le widget QGLWidget :**  
fourni trois méthodes virtuelles `initializeGL()`,  
`resizeGL()` et `paintGL()`.
- **La classe QGLFormat :**  
permet de préciser les options souhaitées pour le  
contexte d'exécution OpenGL (simple / double  
buffering, buffers requis etc).
- ...



# Widgets X11 / OpenGL et bindings

Wrappers autour de GLX et du toolkit graphique correspondant

OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

- **Xt** : `GLwDrawingArea` dans la lib `GLw`.
- **Motif** : `GLwMDrawingArea` dans la lib `GLw`. Hérite de `XmPrimitive`. Offre les callbacks standards Motif / `XmDrawingArea`
- **Gtk+** : `GtkGLArea` (et `GtkGLArea--` pour `Gtk--`) dans la lib `gtkgl`. Hérite de `GtkDrawingArea`.
- **Qt** : `QGLWidget` (en standard).
- **Tcl/Tk** : `Togl`.
- **Java** : `Canvas` du JDK, moult bindings (`jogl...`), implémentations de Java 3D (plus proche d'Open Inventor).



# Problèmes liés à l'animation

Pas vraiment du ressort d'OpenGL

OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

## ■ Caractéristiques :

- ▶ Affichage d'une séquence d'images fixes à une fréquence suffisante
- ▶ Éviter un traçage simultané à l'affichage
- ▶ Profiter du temps d'affichage pour tracer l'image suivante

## ■ Mise en œuvre :

- ▶ Buffering double
- ▶ Rendu hors écran (off-screen redering)
- ▶ Swapping des buffers



# Solutions techniques

## Pas vraiment du ressort d'OpenGL

OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

OpenGL offre le buffering double la plupart du temps (au pire, émulation logicielle), mais pas de mécanisme de swapping (indépendance vis-à-vis du matériel et du système de multifenêtrage) :

### ■ GLX :

- ▶ `glXChooseVisual()` avec `GLX_DOUBLEBUFFER`
- ▶ `glXSwapBuffers()`

### ■ Qt :

- ▶ `QGLFormat::setDoubleBuffer( TRUE );`
- ▶ Le swapping est automatique : implémenté par la classe `QGLWidget` et fait après chaque exécution de `QGLFormat::paintGL()`.



# Ressources

Liens utiles

OpenGL

Didier Verna  
ENST  
VIHM

Généralités

Syntaxe

États

Intégration

Animation

Ressources

- <http://www.sgi.com/Technology/openGL>
- <http://www.opengl.org>
- Le .plan de John Carmack du 23 Décembre 1996
- The OpenGL Programming Guide 4th Edition  
“The Red Book”.  
Addison-Wesley, ISBN 0-321-17348-1.
- The OpenGL Reference Manual 4th Edition  
“The Blue Book”.  
Addison-Wesley, ISBN 0-321-17383-X.
- Google