Introduction à l'OpenGL

Guillaume Libersat <glibersat@linux62.org>

23 mars 2006

Qu'est ce que l'OpenGL ?

OpenGL ?

- API pour réaliser des applications 3d (+ 2d)
- Créé par SGI
- Standard industriel
- Ouvert, portable (indépendant du matériel)
- Utilisable dans de nombreux langages

Qu'est ce que l'OpenGL ?

OpenGL ?

- API pour réaliser des applications 3d (+ 2d)
- Créé par SGI
- Standard industriel
- Ouvert, portable (indépendant du matériel)
- Utilisable dans de nombreux langages

Principal concurrent

- Direct3d (DirectX)
  - Créé par Microsoft
  - Propriétaire
  - Plus proche de Windows
  - Non portable

Qu'est ce que l'OpenGL ?

OpenGL ?

- API pour réaliser des applications 3d (+ 2d)
- Créé par SGI
- Standard industriel
- Ouvert, portable (indépendant du matériel)
- Utilisable dans de nombreux langages

Principal concurrent

- Direct3d (DirectX)
  - Créé par Microsoft
  - Propriétaire
  - Plus proche de Windows
  - Non portable

Conclusion : Programmez en OpengGL !

OpenGL : : technique

Que fait l'OpenGL ?

- Fournit une zone de dessin

OpenGL : : technique

Que fait l'OpenGL ?

- Fournit une zone de dessin
- Fournit des primitives
  - points, lignes, polygones
  - images, bitmaps

OpenGL : : technique

Que fait l'OpenGL ?

- Fournit une zone de dessin
- Fournit des primitives
  - points, lignes, polygones
  - images, bitmaps
- Fournit des éclairages, transparence, ...

OpenGL : : technique

Que fait l'OpenGL ?

- Fournit une zone de dessin
- Fournit des primitives
  - points, lignes, polygones
  - images, bitmaps
- Fournit des éclairages, transparence, ...
- S'occupe du rendu (rasterisation, ...)

OpenGL : : technique

Que fait l'OpenGL ?

- Fournit une zone de dessin
- Fournit des primitives
  - points, lignes, polygones
  - images, bitmaps
- Fournit des éclairages, transparence, ...
- S'occupe du rendu (rasterisation, ...)

OpenGL ne gère pas

- Le matériel
- Le fenêtrage
- Les entrées utilisateur

OpenGL avec...

Toolkit

- Améliore la portabilité (pas d'appel système)
- Permet d'avoir un éventuel rendu imbriqué
- Permet la synchronisation avec d'autres composants

OpenGL avec...

Toolkit

- Améliore la portabilité (pas d'appel système)
- Permet d'avoir un éventuel rendu imbriqué
- Permet la synchronisation avec d'autres composants

Des toolkits

- GLUT
- SDL
- GTK, QT, ...

Donc, il faut...

Dans tous les cas

- Une implémentation d'OpenGL (Mesa, Nvidia, etc)
- Du matériel compatible OpenGL
- Un bon bouquin (le Redbook par exemple)

Donc, il faut...

Dans tous les cas

- Une implémentation d'OpenGL (Mesa, Nvidia, etc)
- Du matériel compatible OpenGL
- Un bon bouquin (le Redbook par exemple)

Et utiliser

- soit L'API OpenGL + un toolkit
- soit utiliser un moteur 3d

Donc, il faut...

Dans tous les cas

- Une implémentation d'OpenGL (Mesa, Nvidia, etc)
- Du matériel compatible OpenGL
- Un bon bouquin (le Redbook par exemple)

Et utiliser

- soit L'API OpenGL + un toolkit
- soit utiliser un moteur 3d

Nous verrons les deux.

API OGL : : Pour la démo

Nous avons besoin

- D'un langage
- D'un toolkit

API OGL : : Pour la démo

Nous avons besoin

- D'un langage
- D'un toolkit

Nous utiliserons

- Python pour le langage
  - Simple à apprendre
  - Simple à utiliser
  - Moderne

API OGL : : Pour la démo

Nous avons besoin

- D'un langage
- D'un toolkit

Nous utiliserons

- Python pour le langage
  - Simple à apprendre
  - Simple à utiliser
  - Moderne
- GLUT comme toolkit

Interlude : Python en 2 slides

- L'indentation remplace les accolades
- Pas de point virgule

Interlude : Python en 2 slides

- L'indentation remplace les accolades
- Pas de point virgule

- from xxx import yyyy : importer yyyy depuis la bibliothèque xxx
- def xxxx() : une fonction
- class xxx(yyy) : xxx hérite de yyy
- def __init__(self) : constructeur
- self.xxx : l'attribut xxx de la classe

Exemple de code python

```
1 class HttpRetriever :
2 ...def __init__(self) :
3 .....self.source = "Ultimate-guitar.com"
4
5 ...def get_name(self) :
6 .....if name == "test" :
7 .........print "booh"
```

l'API OpenGL

OpenGL dispose

- Environ 250 commandes

l'API OpenGL

OpenGL dispose

- Environ 250 commandes

Utilisées pour

- Définir des objets
- Effectuer des transformations

l'API OpenGL

OpenGL dispose

- Environ 250 commandes

Utilisées pour

- Définir des objets
- Effectuer des transformations
- Positionner des attributs
- Effectuer le rendu

l'API OpenGL

OpenGL dispose

- Environ 250 commandes

Utilisées pour

- Définir des objets
- Effectuer des transformations
- Positionner des attributs
- Effectuer le rendu

Toutes les commandes sont préfixées par “gl”

Quelques règles

Pour construire un objet

- Déclarer le début de construction
- Indiquer le type de construction

Quelques règles

Pour construire un objet

- Déclarer le début de construction
- Indiquer le type de construction
- Construire dans le sens anti-horaire
- La forme créée doit être convexe

Quelques règles

Pour construire un objet

- Déclarer le début de construction
- Indiquer le type de construction
- Construire dans le sens anti-horaire
- La forme créée doit être convexe
- Indiquer la fin de construction

Quelques règles

Pour construire un objet

- Déclarer le début de construction
- Indiquer le type de construction
- Construire dans le sens anti-horaire
- La forme créée doit être convexe
- Indiquer la fin de construction

Les types de construction

- Points : affiche des points détachés
- Lignes : trace des lignes entre les points
- Polygone : surface pleine entre les points
- ...

Un exemple en pseudo code

Créer un rectangle

```
CreerUnContexte()
CouleurBlanche()
DebutPolygone()
Point(0.25, 0.25, 0.0)
Point(0.75, 0.25, 0.0)
Point(0.75, 0.75, 0.0)
Point(0.25, 0.75, 0.0)
FinPolygone()
MettreAJourAffichage()
```

Un peu de pratique

Construction pas à pas

1. Rectangle
2. Couleur
3. Animation
4. Volume

Un peu de pratique

Construction pas à pas

1. Rectangle
2. Couleur
3. Animation
4. Volume

Un exemple plus complet

Pour résumer

Assez fastidieux

- Il faut créer point par point
- Il faut gérer toutes les transformations

Et encore, nous n'avons presque rien vu...

Pour résumer

Assez fastidieux

- Il faut créer point par point
- Il faut gérer toutes les transformations

Et encore, nous n'avons presque rien vu...
En pratique il faut gérer

- Les caméras
- Les lumières
- Les déformations
- ...

Pour résumer

Assez fastidieux

- Il faut créer point par point
- Il faut gérer toutes les transformations

Et encore, nous n'avons presque rien vu...
En pratique il faut gérer

- Les caméras
- Les lumières
- Les déformations
- ...

Conclusion : un peu d'aide serait bienvenue !

Les moteurs 3d

De quoi s'agit-il ?

- Un ensemble d'outils
- Dédié à une tâche spécifique
- Souvent utilisé pour les jeux

Les moteurs 3d

De quoi s'agit-il ?

- Un ensemble d'outils
- Dédié à une tâche spécifique
- Souvent utilisé pour les jeux

Avantages

- Évite de recréer la roue
- Fournit des routines optimisées
- Permet de développer rapidement
- ...

Les moteurs 3d

De quoi s'agit-il ?

- Un ensemble d'outils
- Dédié à une tâche spécifique
- Souvent utilisé pour les jeux

Avantages

- Évite de recréer la roue
- Fournit des routines optimisées
- Permet de développer rapidement
- ...

Inconvénients

- Souvent spécifiés
- Disponibilité dans les langages

Des moteurs 3d

Ogre3d (http ://www.ogre3d.org/)

- C++

Crystal Space (http ://www.crystalspace3d.org/)

- C++

Soya3d (http ://home.gna.org/oomadness/fr/soya/index.html)

- Python

Soya 3d

Démos

1. Importation d'objets
2. Clonage, animations (caterpillar)
3. Interaction clavier (caterpillar2)
4. Interaction souris (caterpillar3)
5. Génération de terrain
6. Création d'effets (cellshading)
7. Génération de particules
8. Effets de persistence (ray)
9. Raypicking
10. Caméras
11. Menus

Soya3d (2)

Soya3d + Son + Réseau = Nous avons tout ce qui est nécessaire pour un jeu !

Créons un mini-jeu

But : créer un mini jeu de plate-formes

Créons un mini-jeu

But : créer un mini jeu de plate-formes

Étape 1

1. Mettre en place un décor
2. Ajouter un objet cible
3. Poser une caméra

Créons un mini-jeu

But : créer un mini jeu de plate-formes

Étape 1

1. Mettre en place un décor
2. Ajouter un objet cible
3. Poser une caméra

Étape 2

1. Créer un raypicking

Créons un mini-jeu (2)

Étape 3

1. Modéliser un personnage (blender)
2. Animer le personnage (blender)
3. Importer le personnage dans le jeu

Créons un mini-jeu (2)

Étape 3

1. Modéliser un personnage (blender)
2. Animer le personnage (blender)
3. Importer le personnage dans le jeu

Étape 4

1. Lier les animations à des actions (cal3d)
2. Ajouter de l'interaction utilisateur

Pour finir...

Conclusion

- L'OpenGL est abordable
- Être bon en maths aide mais n'est pas nécessaire
- Il est facile d'avoir un résultat en peu de temps

Pour finir...

Conclusion

- L'OpenGL est abordable
- Être bon en maths aide mais n'est pas nécessaire
- Il est facile d'avoir un résultat en peu de temps

Références

1. Excellent didacticiel : http ://nehe.gamedev.net/
2. The OpenGL Programming guide (The Redbook)

Questions ?