# « Les logiciels libres et le développement de jeux vidéo »

Toulibre

Lionel FUENTES

11 juin 2008

# Introduction

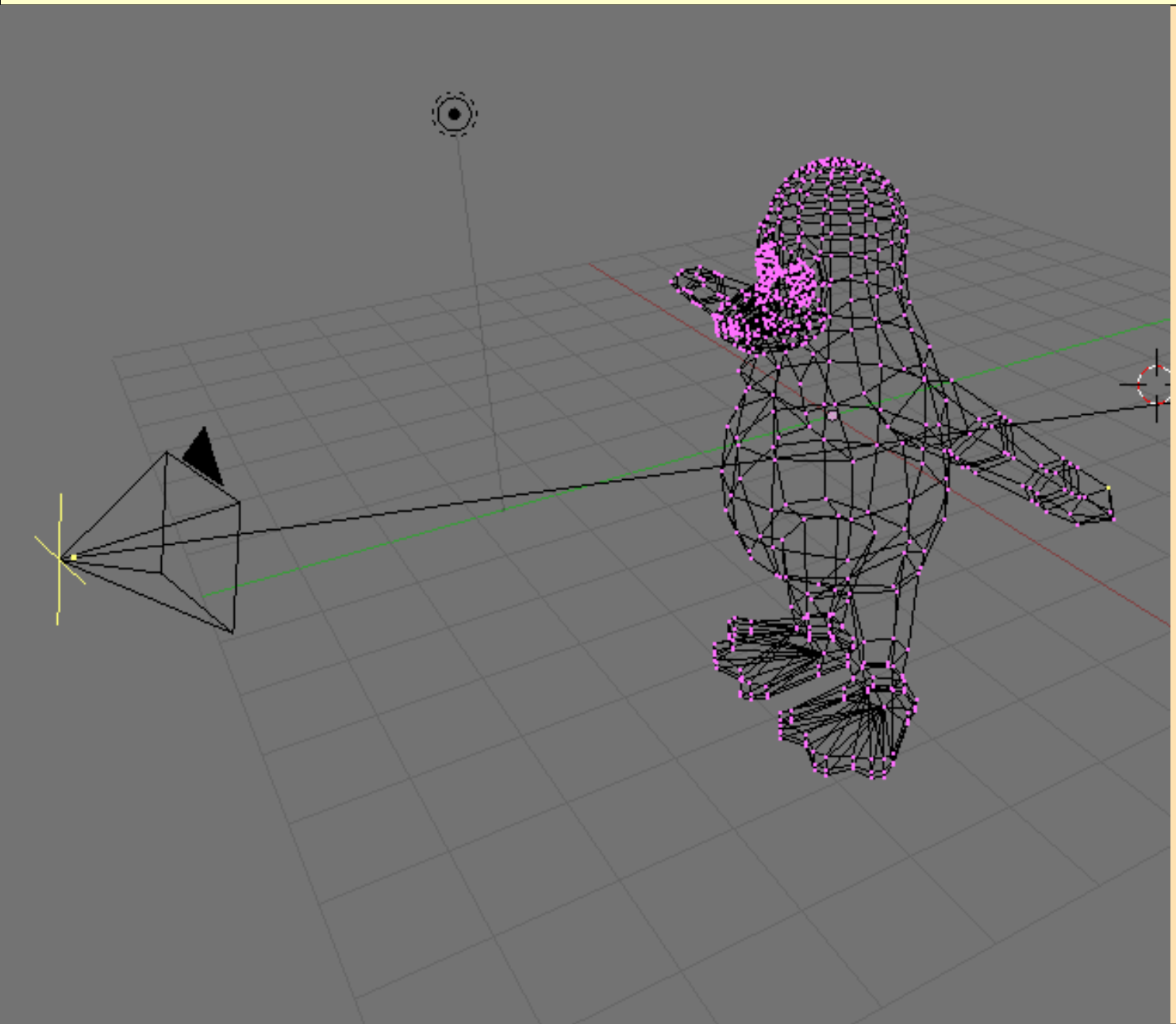
Solutions logicielles nécessaires lors de la création d'un jeu vidéo :

- Modeleur 3D : Blender, Wings3D
- Editeur d'image : GIMP, Inkscape
- Rendu 3D : Irrlicht, OGRE, Crystal Space...
- Son : OpenAL, libvorbisfile...
- Physique : Bullet, ODE, Newton...
- Lecture vidéo : FFMPEG, libtheora, Xine...
- ...etc.

# Plan

I. Notions générales
II. Irrlicht
III. OGRE
IV. Crystal Space
V. Introduction à OpenGL

# I. Notions générales



- Vertex
- Mesh
- Caméra
- Lumière

# I. Notions générales

## Texturing

+

=

# I. Notions générales

Mipmapping : pour une texture carrée donnée, on en crée des copies plus petites de côté $2^n$

# I. Notions générales

Material : combinaison de :

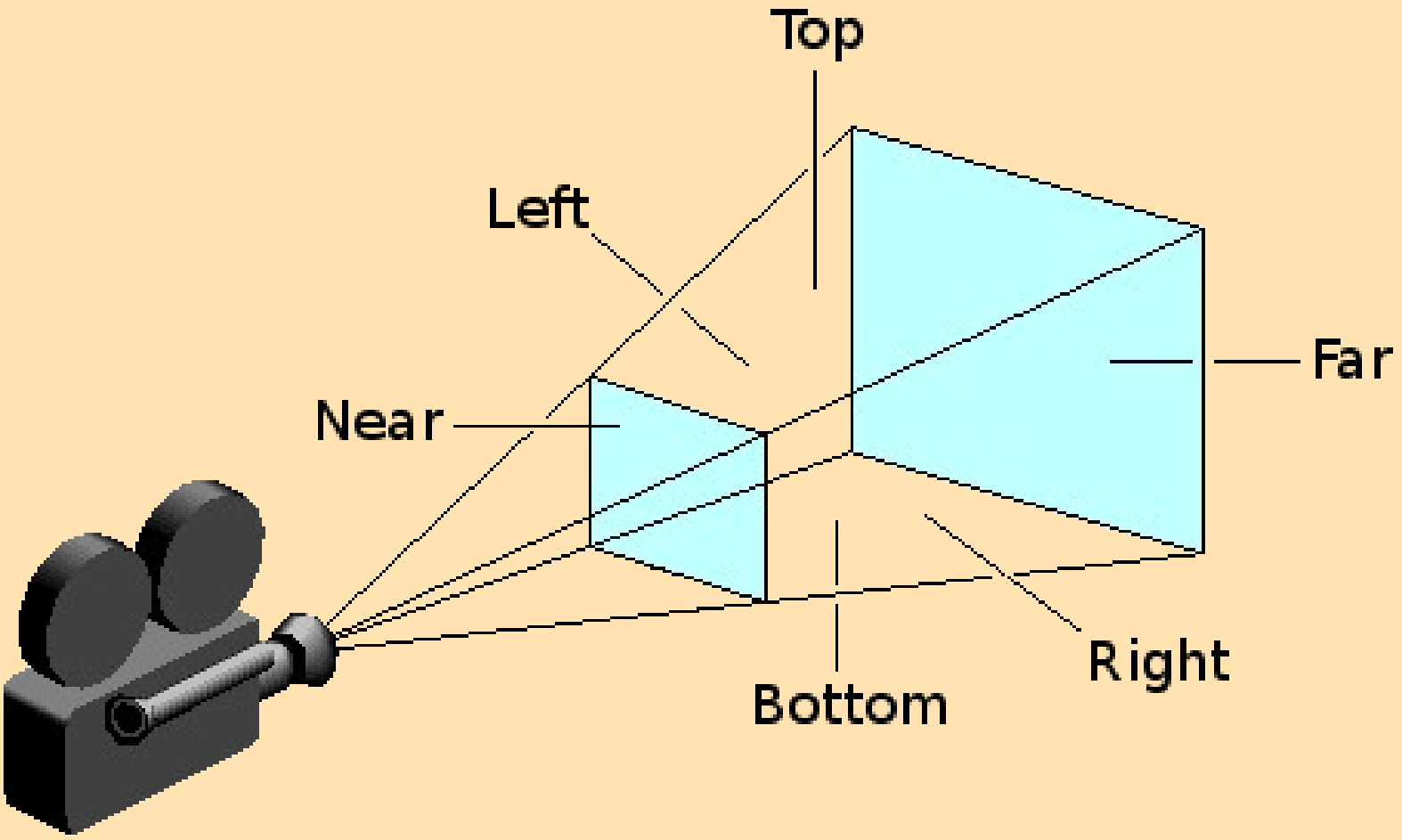
- une ou plusieurs texture(s)

- propriétés relatives au comportement vis à vis d'un certain éclairage (couleurs diffuse, ambiante, émissive et spéculaire)

- shaders (petits programmes spécifiques exécutés directement sur la carte graphique)

# I. Notions générales

## Frustum culling

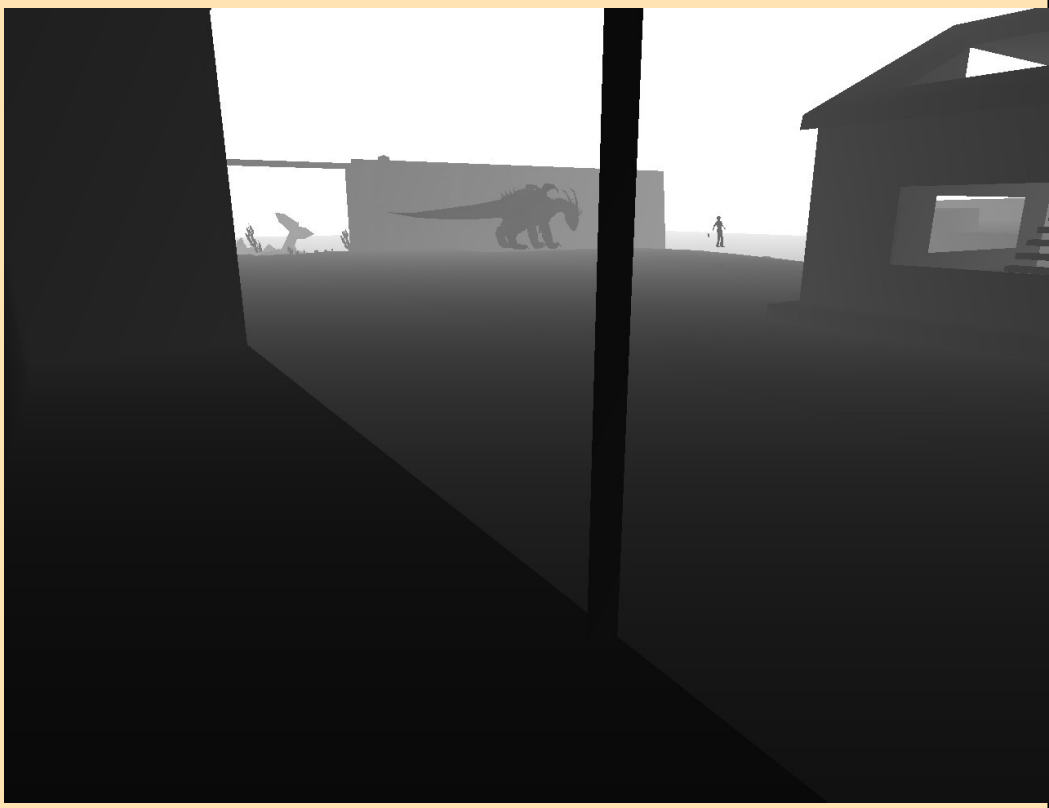



Source : http://deptinfo.unice.fr/twiki/bin/view/Minfo03/AlgosJeux3DRapportFrustum

# I. Notions générales

## Color buffer, Z-buffer, stencil buffer


Color buffer



Z-buffer, ou depth buffer

# I. Notions générales

Render to texture

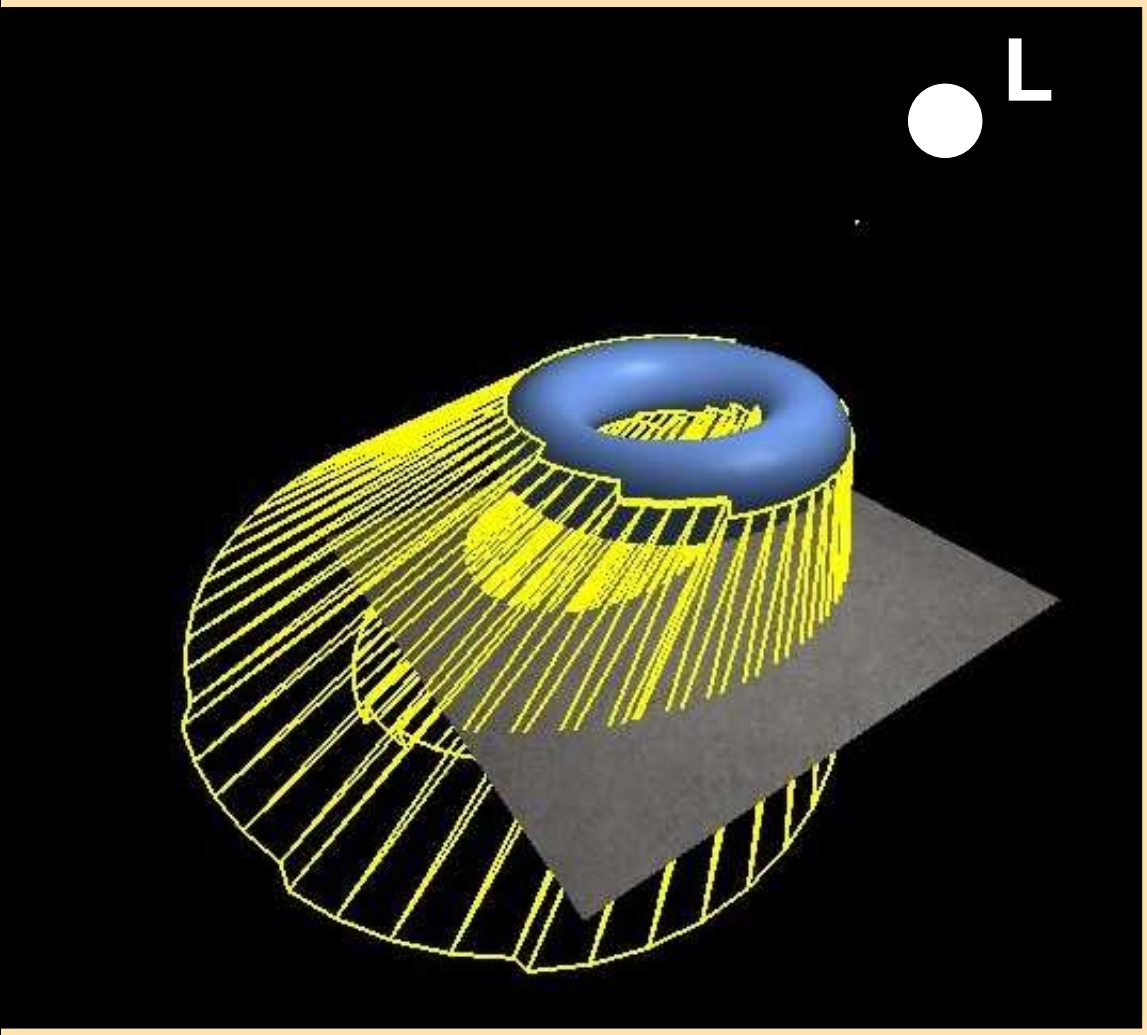
Le nom est explicite : il s'agit de rendre une scène dans une texture ^^

Utilisé pour le shadow mapping, les réflexions, les effets de post-processing (blur, glow...), le rendu de surface d'eau...

Exemple : rendu d'eau

# I. Notions générales

## Stencil shadows, ou shadow volumes




- Consistent en la création d'un «  volume d'ombre  »

- On cherche à déterminer quels points sont à l'intérieur, et lesquels à l'extérieur

# I. Notions générales

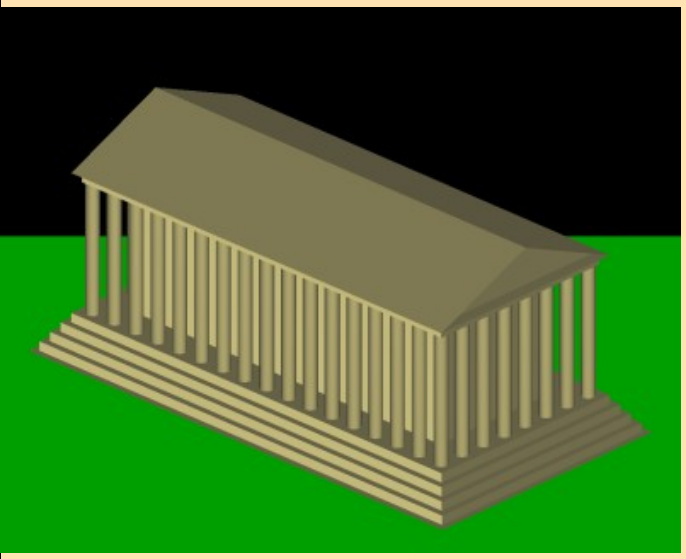
## Shadow volumes dans Doom 3

- Uniquement des ombres «  dures  »

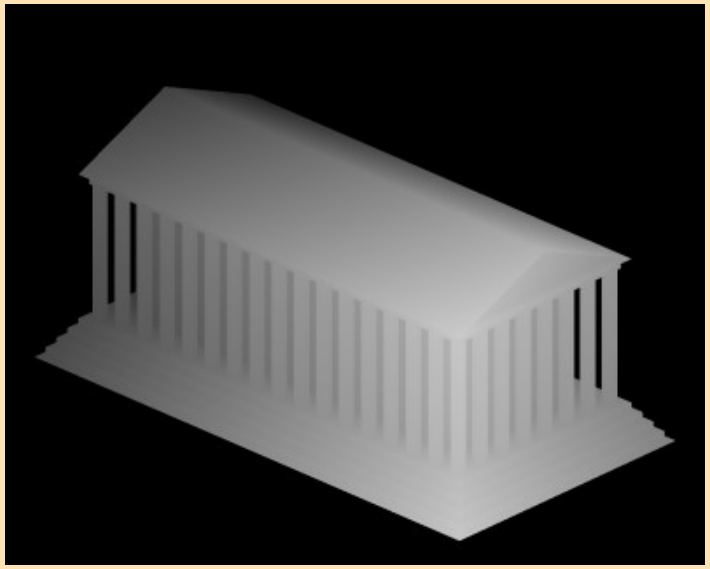
- Consommateur de ressources : beaucoup de vertex créés

# I. Notions générales

## Shadow mapping, ou projective shadowing



Scène depuis le point de vue de la lumière



Z-buffer correspondant

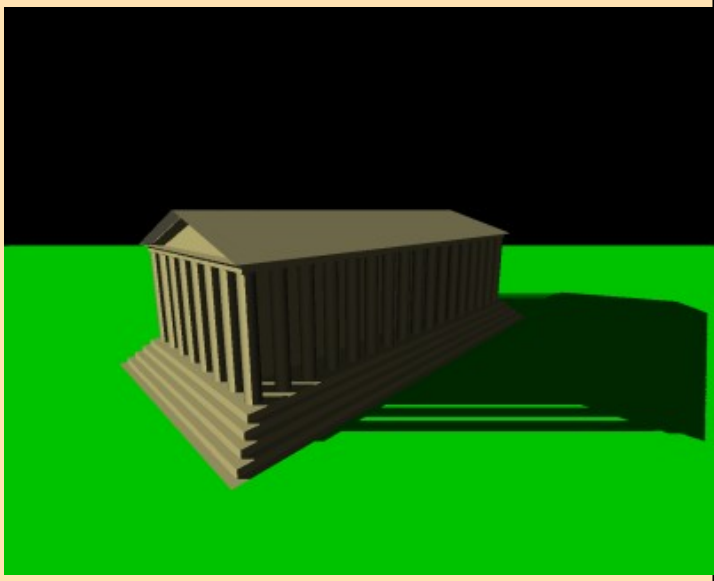
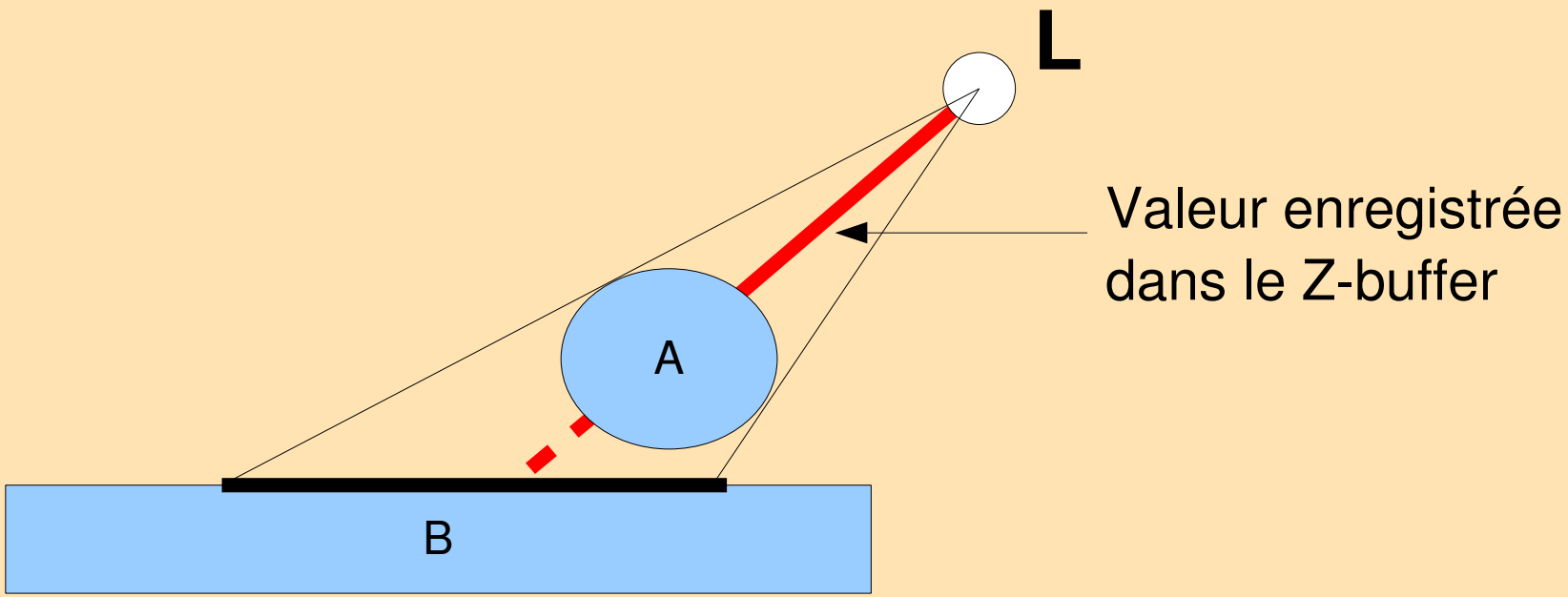


Image finale

Source : Wikipedia

# I. Notions générales

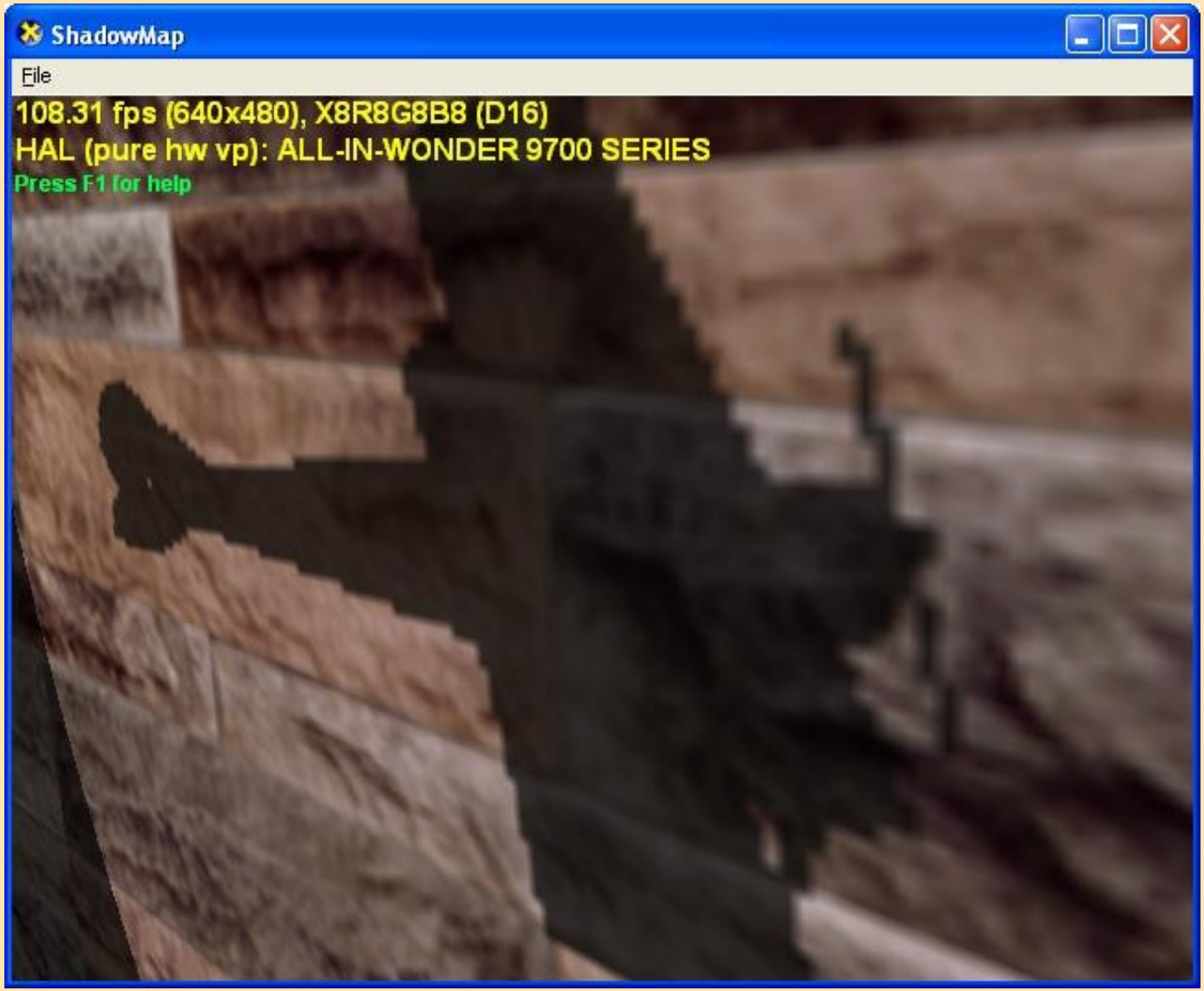
## Shadow mapping




On récupère la distance entre un point et la lumière, et on la compare à la valeur enregistrée dans le Z-buffer

# I. Notions générales

## Problème du shadow mapping : crénelage



# II. Irrlicht


- Le plus jeune des 3 et le moins avancé

- Le plus léger (presque aucune dépendance !)

- De loin le plus simple à utiliser

- Très bonne doc

# II. Irrlicht

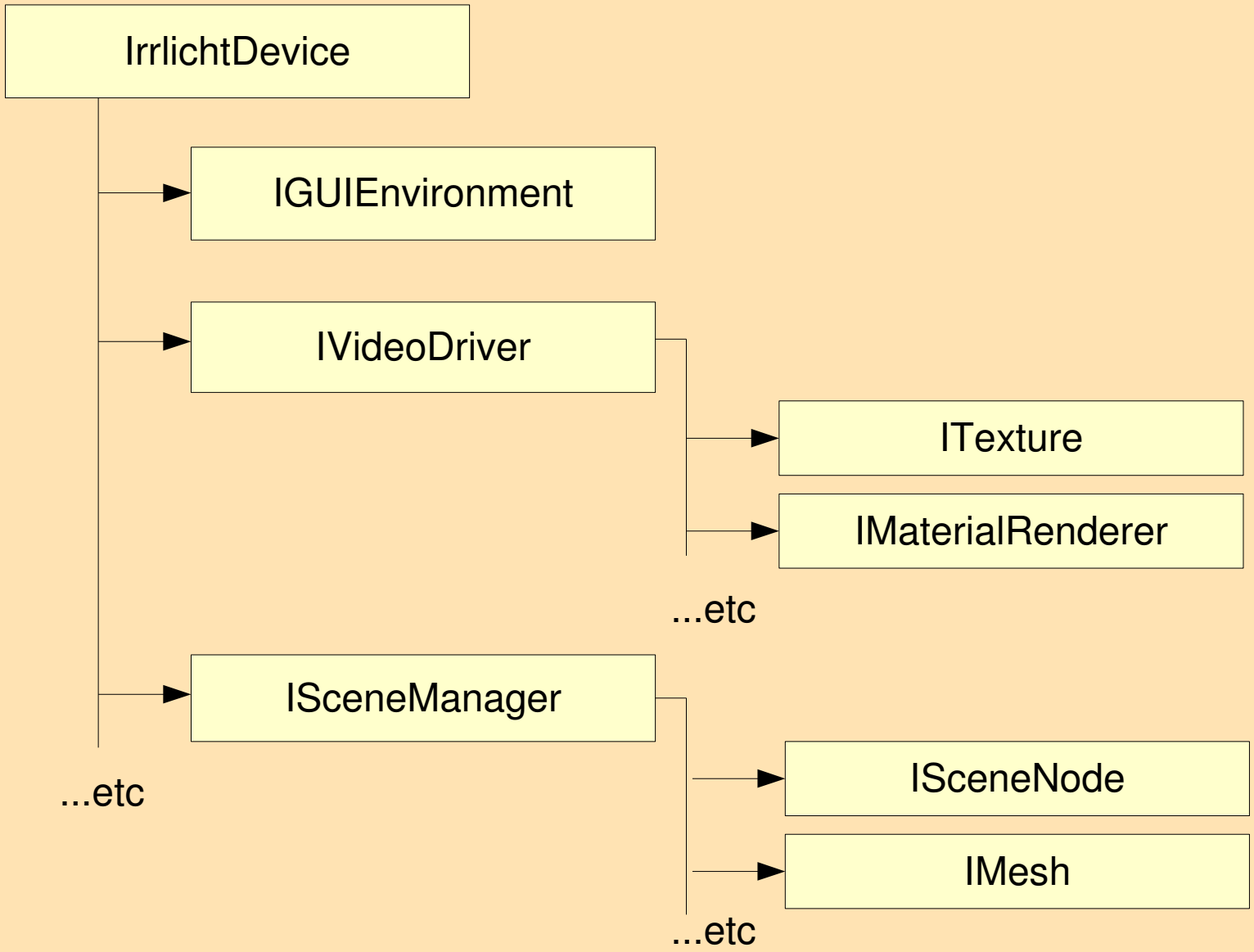
- Support OpenGL et Direct3D 9

- Outils supplémentaires :
  - IrrEdit (uniquement sous Windows, non libre) : world editor
  -IrrKlang : son (non libre)
  - IrrXML : traitement des fichiers XML

- Support de nombreux formats 3D : .x, Milkshape3D, 3DS, MD2...

- Ombres : uniquement shadow volumes

# Irrlicht


IrrlichtDevice
IGUIEnvironment
IVideoDriver
ITexture
IMaterialRenderer
...etc
ISceneManager
ISceneNode
IMesh
...etc
...etc

# II. Irrlicht

- Démos officielles

- Création d'une petite démo !

# III. OGRE


- Le plus avancé des 3

- Object-oriented Graphics Rendering Engine

- **Uniquement un moteur graphique** : pas de gestion de la physique, du son, ni même des entrées/sorties clavier/souris !

- Support OpenGL et Direct3D (version 10 en cours de développement)

# III. OGRE


- Possède son propre format 3D : le .mesh

- Les materials, shaders, systèmes de particules et effets de compositing sont scriptés.

# III. OGRE


- Utilisé dans plusieurs jeux commerciaux

- Nombreuses librairies complémentaires : physique, lecture vidéo, terrain...

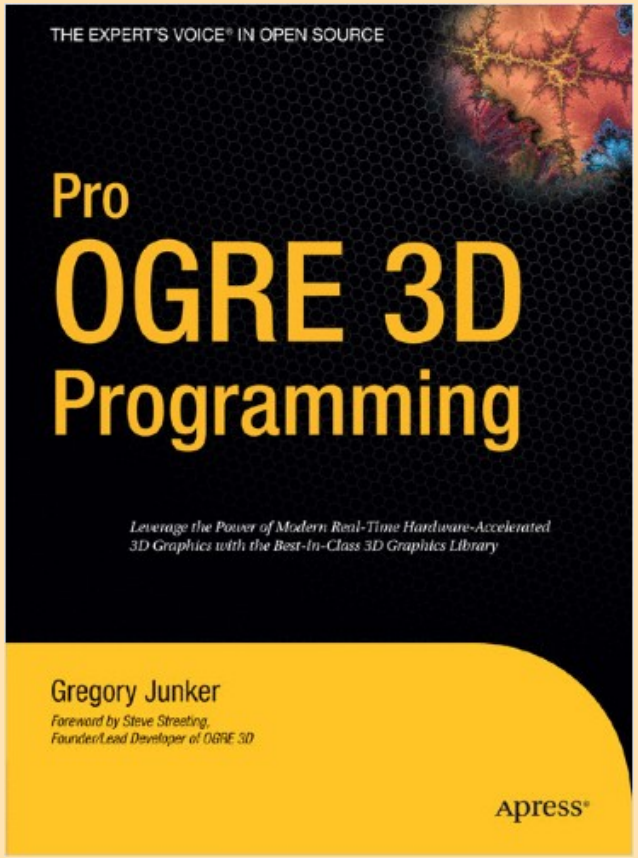
- Architecture « pluggable »

- Depuis la version 1.4 : framework pour le « compositing »

# III. OGRE


- Très bonne documentation : manuel + API reference + exemples

- Un livre de référence :



# III. OGRE


Ankh : jeu commercial utilisant OGRE

# III. OGRE

# III. OGRE

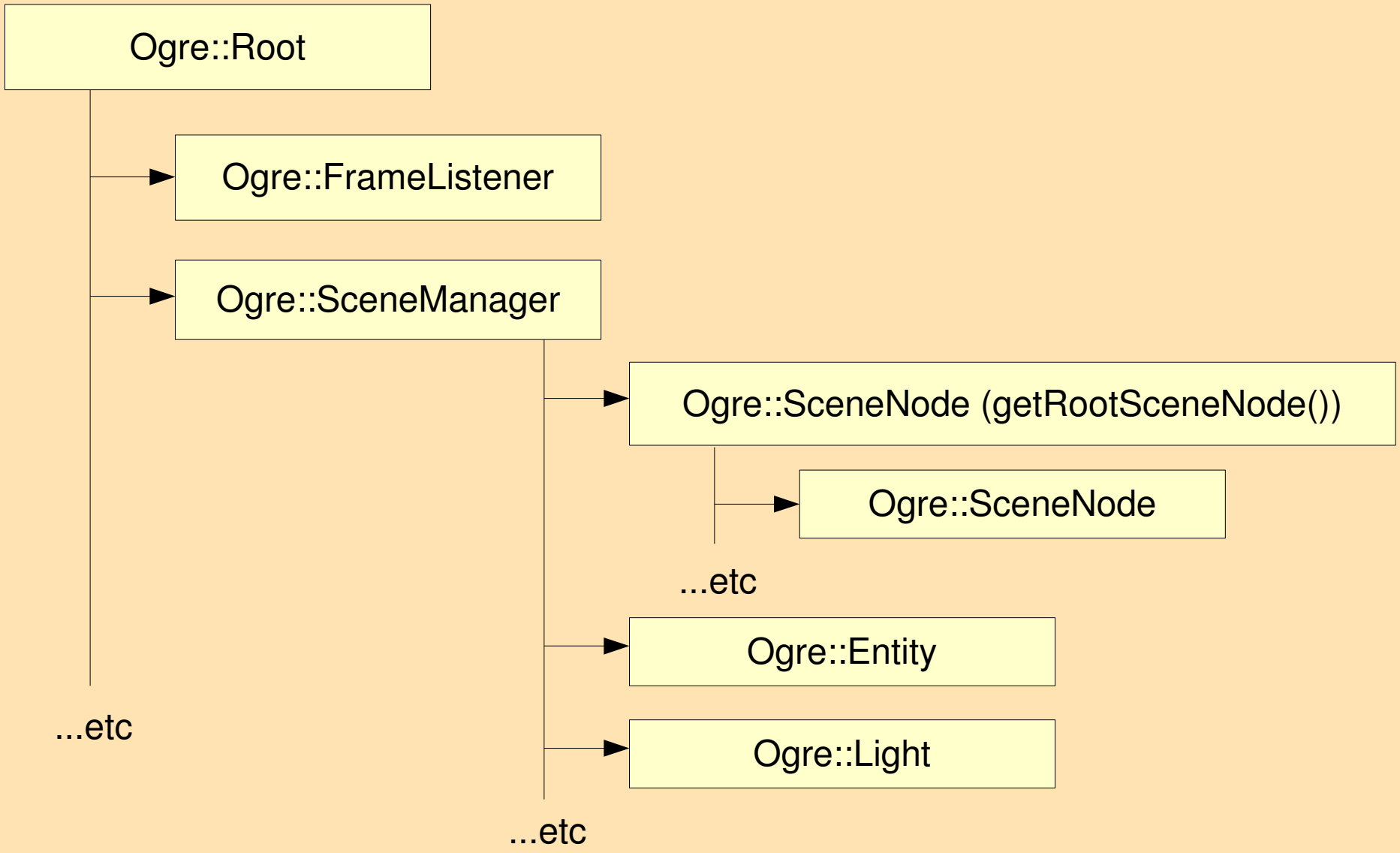
# III. OGRE


Pacific Storm

# III. OGRE


MOTORM4X

# III. OGRE


Ogre::Root
Ogre::FrameListener
Ogre::SceneManager
Ogre::SceneNode (getRootSceneNode())
Ogre::SceneNode
...etc
Ogre::Entity
Ogre::Light
...etc
...etc

# III. OGRE

- Démos officielles

- Création d'une petite démo !

# IV. Crystal Space


- Le plus ancien des 3

- Construit sur un système de plugins

- Gère à peu près tout ce qui est nécessaire à un jeu vidéo (3D, physique, son, scripting...)

- S'utilise habituellement avec CEL : Crystal Entity Layer

# IV. Crystal Space


- Point fort de Crystal Space : Blender2Crystal

- Crystal Space s'utilise souvent avec Blender et Python

# IV. Crystal Space


Moteur du projet de MMORPG Planeshift

# IV. Crystal Space

# IV. Crystal Space


- Apricot : mené par les communautés Blender et Crystal Space
- Vise à créer un jeu 3D libre et à améliorer les 2 projets

# IV. Crystal Space


Apricot...

# IV. Crystal Space

- Démos

- Présentation de Blender2Crystal

# IV. Crystal Space

TODO
Démos :

- Démos officielles
- Exemple :
-> terrain
-> lumière
-> etc...

# V. Introduction à OpenGL


- «  Standard  » : implémenté par les drivers des cartes graphiques

- Portable

- Concurrent de Direct3D

- Utilisé assez rarement dans les jeux vidéo commerciaux, à l'exception de quelques gros titres (Doom 3...)

# V. Introduction à OpenGL


- Supporte un système d' «  extensions  » qui rend l'API facilement évolutive

- Version actuelle : 2.0 : bientôt obsolète avec la sortie de 2 nouvelles APIs :
  - OpenGL Longs Peak
  - OpenGL Mount Evans

# V. Introduction à OpenGL


Nombreuses librairies utilitaires :

- (Free)GLUT, SDL, GLFW : fenêtrage, clavier, joystick...
- DevIL : support du chargement d'images
- Cal3D : chargement et animation de personnages
- GLEW : gestion des extensions
- COLLADA

# V. Introduction à OpenGL


Un peu de pratique :)

# V. Introduction à OpenGL

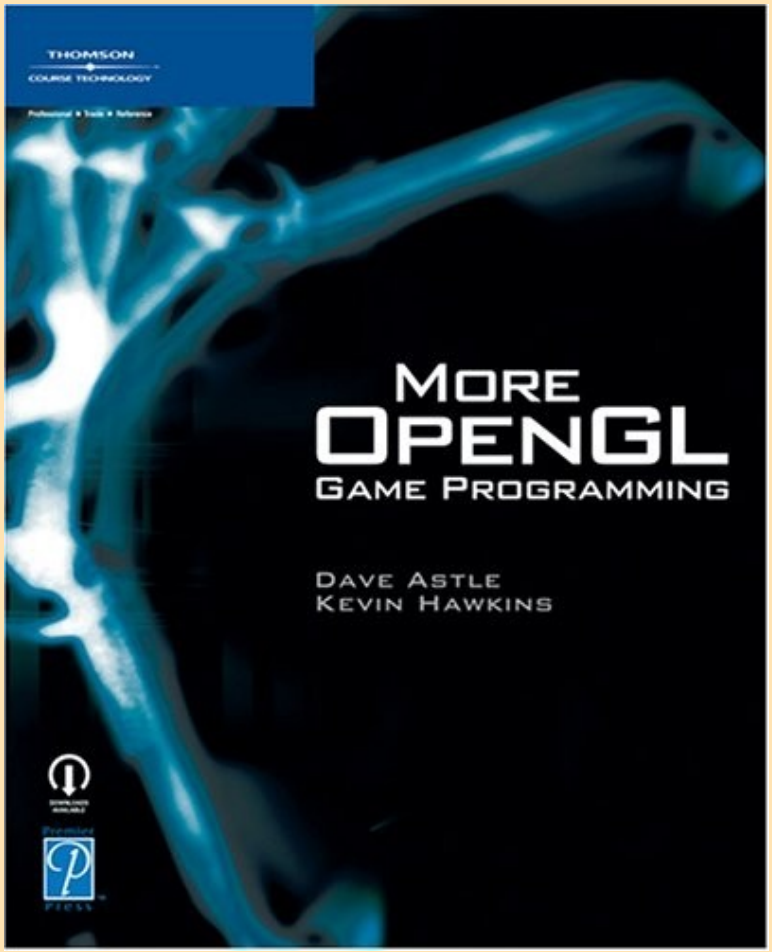
Documentation :

Français :
- GLInFrench : http://glinfrench.apinc.org
- GCN : http://www.games-creators.org
- LinuxGraphic : http://www.linuxgraphic.org
- Coder-Studio : http://www.coder-studio.com

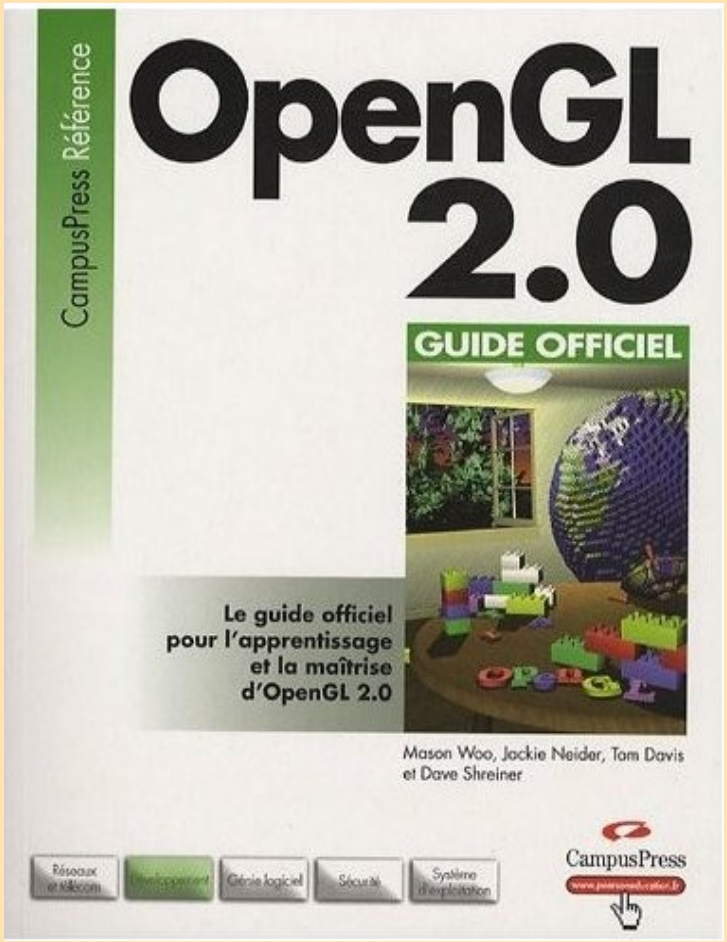
Anglais :
- NeHe : http://nehe.gamedev.net
- GameDev : http://www.gamedev.net

# V. Introduction à OpenGL

## Documentation




Techniques avancées avec OpenGL




Le « red book »

# Fin

Des questions ?