Formation OpenGL

12 – 16 novembre 2012

1 Jour 1 – introduction

1.1 Introduction au rendu graphique

- rendu 3D et pipeline graphique
- techniques de rendu
  - transformations et projection
  - rasterisation
  - shading
  - texture mapping
  - blending
- notions de géométrie dans l'espace
  - systèmes de coordonnées
  - vecteurs et points
  - matrices de transformation
  - projections

1.2 Présentation d'OpenGL

- le standard OpenGL
  - standardisation
  - versions d'OpenGL
  - OpenGL ES
- l'API OpenGL
  - types de base
  - gestion d'erreur
  - GLU, GLUT, GLEW, GLFW, SFML

1.3 Premier projet

- système de build
  - installation
  - chemin des bibliothèques
  - détection automatique
  - compilation et link
  - système
- interfaçage
  - ouverture de fenêtre
  - clavier
  - souris
- boucle principale

  - rafraîchissement
  - méthodes push ou pull
- rendu
  - géométrie
  - attributs
  - affichage d'un triangle
  - dépréciation du mode immédiat
- blending
  - équation de blending
  - alpha test

2 Jour 2 – shaders

2.1 Shaders

- langage GLSL
  - syntaxe
  - exemples
  - vertex shaders
  - fragment shaders
- architecture client/serveur
  - uniforms
  - attributes
  - varyings
- création de shader
  - compilation
  - linkage
  - gestion des erreurs

2.2 Géométrie

- primitives
  - triangles
  - triangle strips, triangle fans
  - points
- primitives indexées
  - pre-processing
  - problème du multi-indexage
  - génération de strips

2.3 Animation

- transformations
  - modèle, vue, projection
  - normales
- rotations

  - angles d'Euler
  - quaternions
  - conversions
- gestion de scène
  - pile de matrices
  - scenegraph
- GLSL avancé
  - shuffling
  - fonctions trigonométriques
  - fonctions rationnelles

2.4 Illumination

- équation de la lumière
  - puissance et flux lumineux
  - modèle de vision humaine
- lumière diffuse
  - éclairage ponctuel
  - éclairage directionel
  - spotlights
- modèles d'illumination
  - « flat shading »
  - modèle ADS
  - shading de Phong
- illumination avancée
  - introduction aux BRDF
  - subsurface scattering

3 Jour 3 – textures

3.1 Formats de pixel

- espaces de couleur
  - RGB, YUV, XYZ, HSL...
  - YCoCg
- quantification
  - int8, int16, int32
  - float16 et float32
- transfert
  - conversion client-side
  - conversion GPU-side

3.2 Texture mapping

- coordonnées de texture
  - 2D

  - correction 3D
- interpolation
  - interpolation 2D
  - mipmaps
  - interpolation brilinéaire

3.3 Exercices

- normal mapping
- post-processing
- rendu dans une texture
- synthèse de texture

4 Jour 4 – GLSL avancé

4.1 Retour sur le langage

- fonctions avancées
  - min, max
  - mix, lerp
- type de données
  - encombrement
  - packing
- génération de bruit
  - bruit blanc
  - bruit rose
  - bruit de Perlin
  - utilisation de textures

4.2 Exercices

- post-processing avancé
- dithering
- antialiasing
- soustraction d'éclairage

5 Jour 5 – performance

5.1 Débogage

- conseils et pratiques courantes
  - modularisation
  - séparation draw calls / calcul
  - glError
  - ARB_debug_output

- gDEBugger

5.2 Performance OpenGL

- mesures
  - calcul de latence
  - chiffres type
  - performance et embarqué
- bande passante GPU
  - architectures IMR et tiled
  - rendu front to back
  - depth prepass
  - compression de texture
  - chargement de textures différé
  - PBO (OpenGL seulement)
- optimisation des appels au driver
  - return to default state
  - state tracking
  - réduction du nombre d'appels (batching)
  - instanciation de géométrie (OpenGL seulement)

5.3 Performance GLSL

- astuces GLSL
  - exploitation du swizzling
  - linéarisation
- optimisation du fragment shader
  - précalcul sur le vertex shader
  - correction de perspective

5.4 Organisation des données

- structures pour le cache
  - AoS
  - SoA
  - AoSoA
- optimisation de géométrie
  - géométrie implicite (points)
  - séparation xy/z
  - compression de vertex

5.5 Multithreading

- stratégies
  - global lock
  - render thread

  - worker threads
  - contextes partagés
- astuces
  - OpenGL et Windows
  - spinlocks
  - transferts asynchrones