

Examen Final

Mercredi 27 Juin 2012

Coefficient : 40 %

Aucun document autorisé. Calculatrice et smartphones non autorisés.

Remarques et conseils :

- Chaque partie devra être rédigée sur une feuille séparée.
- Lisez attentivement chaque question avant d'y répondre.
- Justifiez autant que possible vos réponses.

Partie I (10 points)

Exercice 1 – Questions de cours (6 points)

1. Que représente une *Gloss Map* ? Comment l'utiliser ? (1 point)
2. Que représente un *framebuffer* ? Pourquoi l'utiliser ? (1 point)
3. Qu'est-ce que le *Stencil Buffer* ? Donnez au moins deux exemples d'utilisation. (1 point)
4. Décrivez et explicitez clairement chacune des étapes du pipeline de rendu OpenGL 3.
Vous pourrez illustrer vos propos à l'aide d'un schéma. (2 points)
5. Quel est le principe de fonctionnement de la technique du *deferred shading*. (1 point)

Exercice 2 – Conception du rendu d'une scène 3D (4 points)

Vous êtes en charge de réaliser le rendu de la scène ci-dessous :



Décrivez aussi précisément que possible les étapes nécessaires pour obtenir ce rendu (en précisant notamment quels modèles 3D utiliser pour chaque objet de la scène, les lumières et les textures utilisés, les composantes des lumières et des matériaux, les techniques et shaders utilisées, ainsi que les informations utilisées dans les *Vertex Shaders* et *Fragment Shaders*...).

Partie II (6 points)

Exercice 3 – Programmation Shader (6 points)

On souhaite afficher un ensemble d'objets translucides éclairés par un nombre variable de lumières omnidirectionnelles. On considère le modèle d'illumination de Blinn-Phong dans son intégralité. En supposant que les objets sont affichés du plus éloigné au plus proche de la caméra, écrivez en

GLSL ou en pseudo code le ou les shaders nécessaires au calcul de l'éclairage.

Vous préciserez comment doivent être configurés les états du pipeline OpenGL avant l'invocation de chaque shader et l'ordre d'exécution s'il y en a un.

Vous expliquerez entre autres à quelles informations correspondent chacune des entrées (attributs et variables uniforms) de votre ou vos shaders.

Partie III (4 points)

Exercice 5 – Structures de données (4 points)

1. Décrivez en pseudo-code une structure de données de type octree permettant de stocker des objets représentés par une boîte englobante alignée sur les axes du monde.
2. Écrivez une fonction permettant de récupérer la liste de tous les objets de l'octree en intersection avec une boîte alignée sur les axes du monde passée en paramètre.