Département d'informatique et de recherche opérationnelle
Université de Montréal

Pierre Poulin
Bureau : 2389
poulin@iro.umontreal.ca

IFT3350/IFT6350 : Infographie
Hiver 2004
Examen Intra : 20%

Règlements :
- documentation et calculatrice permises
- répondre sur les pages de droite du cahier réponse (utilisez les pages de gauche comme brouillon)
- durée de deux heures

Question 1 (2 points)

Modifiez l'algorithme de remplissage à germe (***flood fill***) pour qu'il remplisse les pixels avec une couleur linéairement décroissante à partir du germe (***seed***), dans un cercle de rayon maximal de n pixels.

Question 2 (2 points)

Souvent lors de la modélisation, les artistes créent la moitié du modèle géométrique, en font une copie, et appliquent une matrice de changement d'échelle ($Scale(-1, 1, 1)$ par exemple) pour créer l'autre moitié miroir du modèle. Si les normales ne sont pas explicitement assignées, ces modèles sont problématiques par rapport à l'illumination et le *backface culling*. Expliquez pourquoi.

Question 3 (2 points)

Décrivez un algorithme qui détermine si un point est à l'intérieur d'un polygone convexe en exploitant les propriétés du produit vectoriel et le fait que les sommets soient tous ordonnés dans le même sens, que ce soit en sens horaire ou anti-horaire.

Question 4 (2 points)

Expliquez comment une opération de *clipping* de la pyramide de vue peut rendre incohérentes les valeurs du compteur d'ombre de chaque pixel lors de la *scan-conversion* des polygones d'ombre (*shadow volume polygons*).

Question 5 (1 point)

Qu'advient-il dans l'algorithme de visibilité par lignes de balayage (*scanline*) lorsqu'un segment 3D d'un polygone se projette entre deux lignes de balayage ? Note : le polygone lui-même se projette sur plusieurs lignes de balayage.

Question 6 (3 points)

Les fameux facteurs $1/r^2$ et $\cos\theta$ en illumination directe d'une source de lumière polygonale portent souvent à confusion. Ici, $\cos\theta$ représente l'angle entre la normale à la source de lumière et la direction vers le point à illuminer. Pour un point à illuminer avec sa normale, on peut calculer l'illumination directe en
(a) choisissant aléatoirement des points sur la source de lumière ;
(b) choisissant aléatoirement des directions dans l'angle solide formé par la source de lumière au point à illuminer ;
(c) choisissant aléatoirement des directions dans l'hémisphère du côté de la normale du point à illuminer.
Dans chacun de ces trois cas, dites si le facteur $1/r^2$ et/ou $\cos\theta$ sont nécessaires, et expliquez pourquoi.

Question 7 (2 points)

Expliquez comment générer automatiquement un *shadow map* qui sera valide pour toute une scène polygonale éclairée par une source de lumière directionnelle.

Question 8 (3 points)

(a) Décrivez un algorithme utilisant le z-buffer qui permettrait d'afficher le contenu d'une scène normalement, mais qui afficherait par-dessus une zone zoom centrée sous le curseur. Dans la zone zoom, la distance entre deux points image est plus petite que dans l'image normale.

(b) On déforme l'image dans la fenêtre de zoom de la même façon que le serait une image due à une lentille placée devant l'image formée sur le plan de projection. Existe-t-il des différences entre ceci et simuler une lentille en 3D devant la caméra virtuelle ? Si oui, donnez un exemple.

Question 9 (1 point)

Pour certaines surfaces, il n'existe pas de paramétrisation de surface qui permette d'associer (*mapping*) une texture rectangulaire sur la surface sans déformations critiques des texels. Une solution consiste à "découper" la texture en morceaux et de les disposer dans une seule texture. Expliquez une difficulté causée par ce découpage dans le filtrage de ce type de textures.

Question 10 (2 points)

Dans l'exemple de la figure 1, un rectangle est filtré par *mip-mapping*. Cependant lorsque le rectangle est tourné et projeté à angle plus rasant, la texture dans le pixel au centre devient beaucoup trop floue, plus qu'on ne s'y attendrait. Qu'est-ce qui dans le *mip-mapping* expliquerait cette situation ?

FIG. 1 – Un rectangle texturé (gauche) est tourné en 3D autour de sa diagonale pour finir dans la configuration de droite. Le carré gris au centre de chaque projection perspective de ce rectangle représente le pixel qui nous intéresse.