

Master 1 Informatique

Examen Infographie

Mercredi 8 décembre 2009
13h30-15h30

Les documents papier sont autorisés.

Les ordinateurs sont interdits. Toute connexion à Internet est interdite.

Le cas échéant, on n'oubliera pas de fournir quelques commentaires sur les techniques employées.

Question 1: Modélisation OpenGL

On souhaite créer un jeu de simulation du billard français.

a) Ecrire une fonction void billard(void) ; modélisant en OpenGL un table de billard composée d'une surface et de 4 bords. La surface est modélisée par un polygone rectangulaire placé dans le plan xz et centré sur le point de coordonnées (0.0, 75.0, 0.0). Les bords sont constitués de 4 parallépipèdes rectangles accolés aux cotés de la surface rectangulaire, modélisés avec une section de 10.0x5.0 (5.0 est la hauteur) et définissant une surface intérieure de 220.0x160.0.

b) Ajouter un matériel permettant d'afficher votre table de billard comme recouverte de feutrine verte relativement foncée. Les bords seront bleu foncé.

c) Ajouter une lumière ponctuelle blanche située au dessus du centre du billard à une distance de 80.0.

d) Une fenêtre d'affichage de ratio 16/9èmes est utilisée. Quelle fonction reshape définiriez-vous pour effectuer une visualisation en projection parallèle orthographique en vue plongeante « du dessus » orientée vers le centre de la table de manière que celle-ci soit intégralement visible?

e) Toujours pour une fenêtre d'affichage de ratio 16/9èmes, quelle fonction reshape définiriez-vous pour effectuer une visualisation en projection en perspective orientée vers le centre de votre billard depuis le point de vue de position (250.0, 180.0, 150.0) de manière que le billard soit entièrement visible?

Question 2: Clipping

L'algorithme de clipping de Cohen-Sutherland permet le clipping d'un segment de droite dans un rectangle à bords horizontaux et verticaux. Spécifier une version modifiée de cet algorithme pour le clipping d'un segment de droite quelconque dans un triangle quelconque. On précisera les difficultés spécifiques à cette généralisation par rapport à la version classique et quelles solutions y sont apportées.

Question 3: Mathématiques

a) Prouver mathématiquement que les zooms 3D non uniformes (rapports différents selon les axes x, y et z) conservent les parallélismes.

b) Prouver mathématiquement que les zooms 3D non uniformes ne conservent pas les angles droits.

c) Décrire et implanter une méthode de test du parallélisme de 2 facettes triangulaires.

Question 4: Algorithmique de tracé

L'algorithme ci-contre réalise la rasterisation 2D d'un cercle de rayon r centré sur l'origine O en limitant le dessin aux pixels du second octant :

Adapter cet algorithme pour qu'il réalise le **remplissage** d'un disque entier en s'assurant, par souci d'optimisation, qu'aucun pixel ne sera « allumé » deux fois.

```
void arcDeCercle(int r) {
    int x,y,d ;
    x = 0 ;
    y = r ;
    d = 1 - r ;
    allume_pixel(x,y) ;
    while ( y > x ) {
        if ( d < 0 )
            d += 2 * x + 3 ;
        else {
            d += 2 * (x - y) + 5 ;
            y-- ; }
        x++ ;
        allume_pixel(x,y) ; }
}
```

