Examen Médian

Mardi 6 Mai 2008, de 10h15 à 12h15

Coefficient : 25 % Aucun document autorisé. Calculatrice autorisée.

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 : Questions de cours (6 points)

Exercice 1 – Questions de cours (4 * 1 point + 2 points)

1. Qu'est-ce qu'un SceneGraph ? Pourquoi est-il nécessaire d'en utiliser un lors du rendu d'une scène ? Illustrez vos propos par un exemple.
2. Pourquoi les librairies graphiques comme Java3D ou OpenGL utilisent-elles des vecteurs et des matrices homogènes ?
3. Rappeler ce qu'est une transformation d'Euler.
   Quels sont les avantages d'utiliser des quaternions plutôt que des transformations d'Euler pour orienter des objets 3D ?
4. Expliquer aussi précisément que possible la méthode qui permet à un Quad de toujours faire face à l'observateur.
5. Démontrez que la projection d'un vecteur $\vec{U}$ sur un vecteur $\vec{V}$ peut-être calculée par l'équation $proj_{\vec{V}} \vec{U} = \frac{\vec{U}.\vec{V}}{\|\vec{V}\|^2} \vec{V}$. **(2 points)**

Partie II : Intersections et transformations géométriques (7 points)

Exercice 2 – Calculs d'intersection (3 points)

Soient les points $P_1 = (1\ 2\ 0)^T$, $P_2 = (2\ 0\ -1)^T$ et $P_3 = (0\ -2\ 1)^T$ et la droite définie par $A(t) = S + tV$ avec $S = (1\ 1\ 6)^T$ et $V = (0\ 0\ 1)$.

1. Déterminer, s'il existe, le point d'intersection entre la droite A(t) et le plan (P) formé des points P_1, P_2 et P_3.
2. Si ce point d'intersection existe, déterminer s'il est à l'intérieur du triangle formé par les trois points P_1, P_2 et P_3.

Exercice 3 – Transformations (4 points)

Dans cet exercice, les vecteurs et matrices sont exprimés en coordonnées homogènes.

Soient deux points $P_1 = (10\ 5\ 1\ 1)^T$ et $P_2 = (-3\ 10\ -5\ 1)^T$ définis dans l'espace affine homogène ayant pour origine $O = (0\ 0\ 0\ 1)^T$ et comme base les vecteurs orthonormés $e_1 = (1\ 0\ 0\ 0)^T$, $e_2 = (0\ 1\ 0\ 0)^T$ et $e_3 = (0\ 0\ 1\ 0)^T$.

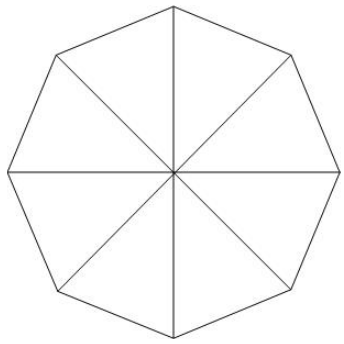
1. Donnez les coordonnées du point P_2, image du point P_1 ayant subi tout d'abord une rotation

de 45° autour de l'axe de vecteur $(5\ 3\ 2\ 0)^T$ passant par le point $P = (4\ 4\ 5\ 1)^T$ puis une translation de vecteur $(1\ 3\ 5\ 0)^T$.

2. En considérant que le point P_1 résulte de l'application de la transformation $M = \begin{pmatrix} \cos 45 & -\sin 45 & 0 & 1 \\ \sin 45 & \cos 45 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ sur le point P_0, exprimer P_1 en fonction de P_0, P_0 en fonction de P_1 et donnez alors les coordonnées de P_0.

Partie III : Java3D (7 points)

Exercice 4 - Création d'une forme (3 + 2 = 5 points)



1. On désire créer de façon automatique la forme géométrique ci-contre en utilisant Java3D. Complétez la méthode « createShape » qui prend en paramètre le nombre de rayons et leurs longueurs et qui retourne un LineArray qui va contenir tous les segments de notre forme.

```
public LineArray createShape(int rayCount, float rayLength) {

    LineArray lines = new LineArray(... , GeometryArray.COORDINATES);
    ...
    return lines;
}
```

Note 1: *Le LineArray dispose de la méthode setCoordinate(int index, Point3f point) qui permet d'ajouter des vertex à la forme aux indices passés en paramètre.*

Note 2: *l'équation paramétrique du cercle est:*

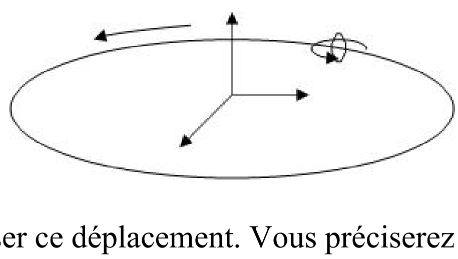
$$x = a + r\cos(angle)$$
$$y = b + r\sin(angle)$$

où (a, b) est le centre du cercle, r le rayon.

2. L'utilisation d'un LineArray présente le principal défaut de dupliquer les vertices. Modifiez votre méthode afin d'utiliser un IndexedLineArray en remplacement du LineArray.

Note: *L'IndexedLineArray dispose, en plus de la méthode « setCordinate » du LineArray, la méthode setCoordinateIndex(int index, int coordinateIndex) qui permet de passer les indices des vertex à utiliser.*

Exercice 5 - SceneGraph (2 points)



Nous souhaitons faire tourner une Shape3D sur elle-même en même temps que sur une trajectoire en forme de cercle, comme le montre le schéma ci-contre.

Pour cela, nous utiliserons des « RotationInterpolator ».

Dessinez le SceneGraph Java3D à mettre en place pour réaliser ce déplacement. Vous préciserez les liens hiérarchiques entre les différents composants, les liens avec les RotationInterpolator et pour chaque TransformGroup, les transformations qui y seront appliquées.