Examen Médian

Mercredi 9 Mai 2012

Coefficient : 30 % Aucun document autorisé.

Remarques et conseils :
- Lisez attentivement chaque question avant d'y répondre.
- Justifiez autant que possible vos réponses.

Exercice 1 – Questions de cours (4 points)

1. Expliquez et démontrez que la matrice à appliquer à chaque normale d'une surface d'un objet est l'inverse de la transposée de la matrice Modèle. **(2 points)**
2. Rappelez la séquence des transformations appliquées à un objet 3D jusqu'à sa visualisation dans une fenêtre 2D. Pour chaque transformation, précisez le nombre de coordonnées du vecteur sur laquelle elle est appliquée et dans quel espace ce vecteur s'exprime. **(2 points)**

Exercice 2 – Modélisation procédurale d'une pyramide (5 points)

Soient :
- V un tableau de N *vertices* 3D,
- C un tableau de M couleurs RGB,
- (I_1,P_1), …, (I_n,P_n) des couples de tableaux d'indices et de primitives associées permettant d'effectuer le rendu d'une pyramide.

On considère que l'origine de l'espace local de la pyramide est au centre du carré représentant sa base. L'objectif est de construire par primitive graphique une pyramide dont la base est de couleur rouge et les quatre pans de la pyramide de couleur bleue.

Questions :
- En numérotant sur une figure les vertices de la pyramide, renseigner chacun des tableaux de vertices et de couleurs.
- Définissez l'ensemble des couples de tableaux d'indices et de primitives associées permettant d'effectuer ce rendu. Seules les primitives TRIANGLES, TRIANGLE_FAN et TRIANGLE_STRIP seront à considérer.

Exercice 3 – Programme *Shader* (3 points)

Soit le carré défini par les tableaux de vertices et de couleurs suivants :

Vertices 3D	0,0,0	1,0,0	0,1,0	1,1,0
Couleurs RGB	0,1,0	1,0,0	1,0,0	1,0,0

Quel rendu obtient-on si le programme *Shader* ci-dessous est utilisé avec les informations précédentes (tableau de coordonnées de vertices et tableau de couleurs) :

Vertex Shader	Fragment Shader

uniform mat4 mvp;
 	
in vec3 position;
in vec3 color;
 	
out vec3 fColor;
out vec2 fCoords;
 	
void main()
{
  fColor = color;
  fCoords = position.xy;
 	
  position.x *= 2.0f ;
  gl_Position = mvp*vec4(position, 1.0f);
}
	const float border = 0.05;
	
	in vec3 fColor;
	in vec2 fCoords;
	
	out vec4 fragColor;
	
	void main()
	{
	  if (fCoords.x > border && fCoords.x < (1-border) && fCoords.y > border && fCoords.y < (1-border))
	  {
	   fragColor = vec4(fColor, 1.0f);
	  }
	  else
	  {
	   fragColor = vec4(0.0f, 0.0f, 0.0f, 1.0f);
	  }
	}

Justifiez votre réponse.

Exercice 4 – Transformations (4 points)

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

Soit un point $P_1 = (6\ 4\ 2\ 1)^T$ défini 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 $e_3 = (0\ 0\ 1\ 0)^T$.

Exprimez puis 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 $(1\ 3\ 5\ 0)^T$.
L'utilisation d'un quaternion vous permettra d'obtenir P_2 plus rapidement.

Exercice 5 – Intersections (4 points)

1. Déterminer les paramètres (normale N et distance signée D) du plan P passant par les points $P_1 = [1\ 2\ 0]^T$, $P_2 = [2\ 0\ -1]^T$ et $P_3 = [3\ -2\ 1]^T$.
2. Soient deux droites $A(t_1) = S_1 + t_1 V_1$ et $B(t_2) = S_2 + t_2 V_2$, avec $S_1 = (8\ 7\ 10)^T$, $V_1 = (1\ 2\ 0)^T$, $S_2 = (3\ 3\ 8)^T$ et $V_2 = (4\ 4\ -4)^T$.
   Pour chacune de ces droites, déterminer si elle intersecte le plan P et le(s) point(s) d'intersection.