

Informatique Graphique - Feuille de TD

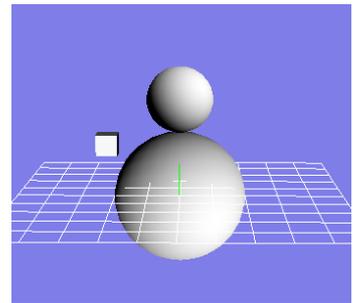
1. Affichage des formes de base

- 1) Définir un triangle, un quadrangle, une pyramide
- 2) Définir un cube avec des `GL_TRIANGLE_STRIP` et une structure indexée
- 3) Définir un cylindre et un cône avec des `GL_TRIANGLE_STRIP`
- 4) Définir une sphère avec des `GL_TRIANGLE_STRIP`

2. Affichage d'un objet complexe

Définir un bonhomme composé :

- d'un corps blanc de diamètre 4 et de centre en $(0,0,0)$
- d'une tête blanche de diamètre 2



3. Affichage des normales

Soit (n_x, n_y, n_z) la normale du point P de coordonnées (x, y, z) . Ecrivez le code OpenGL qui permet d'afficher cette normale au point P , sous la forme d'un segment. Cet affichage permet par exemple de vérifier le calcul de la normale.

4. Navigation

Soit une caméra positionnée en $pos = (2, 2, 2)$ et dirigée vers le point $C = (1, 1, 1)$.

1. Calculez l'origine O' et les vecteurs unitaires (i_1, j_1, k_1) du repère lié à cette caméra. Nous souhaitons que la direction donnée par k_1 corresponde à la direction de visée de la caméra et que l'axe des z du repère de la scène apparaisse comme vertical par projection dans le plan XOY du repère de la caméra.
2. En déduire la matrice M associée à cette caméra c'est-à-dire la matrice de passage du repère de la scène au repère de la caméra.
3. Soit le point $P = (0, 0, 1)$ (exprimé dans le repère de la scène). Calculez les coordonnées du point P dans le repère de la caméra.

5. Transformations

Vous avez écrit le code OpenGL permettant l'affichage d'un objet.

1. Ecrivez l'instruction OpenGL (mise avant l'affichage de l'objet) qui permet de traduire l'objet de $(-2, 1, 3)$. Ecrivez la matrice de transformation (en coordonnées homogènes) associée à cette instruction.
2. Ecrivez l'instruction OpenGL (mise avant l'affichage de l'objet) qui permet d'effectuer un changement d'échelle d'un facteur 2 selon X de l'objet. Ecrivez la matrice de transformation (en coordonnées homogènes) associée à cette instruction.
3. Ecrivez l'instruction OpenGL (mise avant l'affichage de l'objet) qui permet d'afficher l'objet vu de derrière.
4. Ecrivez l'instruction OpenGL (mise avant l'affichage de l'objet) qui permet d'afficher l'objet vu de dessus.

6. Projection

La projection d'une scène 3D se fait en deux étapes. Nous devons tout d'abord ramener tous les points de la scène dans le repère de la caméra. Puis, nous effectuons la projection de la scène du repère de la caméra vers le repère image.

1. La fonction `gluLookAt(ex, ey, ez, cx, cy, cz, upx, upy, upz)` de la librairie GLU permet de définir la position et l'orientation de la caméra, avec :
 - (ex, ey, ez) : position de la caméra,
 - (cx, cy, cz) : position vers laquelle la caméra pointe (aussi appelée ligne de mire),
 - (upx, upy, upz) : un vecteur qui nous informe de la direction du haut de la caméra.

Explicitement le code OpenGL permettant de faire tourner la caméra autour de l'objet affiché dans la scène.

2. La fonction `gluPerspective(fovy, aspect, zNear, zFar)` permet de définir la projection utilisée en spécifiant le volume de visualisation, avec :
 - `fovy` : angle d'ouverture vertical,
 - `aspect` : rapport largeur/hauteur,
 - `zNear, zFar` : plans de clipping proche et éloigné.

Faites un schéma du volume de visualisation en faisant apparaître les différents paramètres de la fonction `gluPerspective()`.