

Stage C++ - TP1

Master informatique 2ème année - Université Bordeaux I
Constantes, références, surcharge de fonctions.

11 octobre 2007

1 Hello World

- Écrivez et compilez un programme qui affiche “Bonjour le monde”.
- Écrivez et compilez un programme qui demande votre âge et qui affiche “ok” sur la sortie standard si vous avez un âge positif et “erreur” sur la sortie d’erreur dans le cas contraire.

2 Fenêtre X11

- En utilisant l’objet X11 fourni, écrivez “Hello World” au centre d’une fenêtre X11.
- Écrivez un programme qui dessine un carré au centre de la fenêtre et qui affiche 1000 points au hasard à l’intérieur de celle-ci. Pour la génération aléatoire on utilisera la fonction rand() de la bibliothèque “cstdlib” qui renvoie un entier pseudo-aléatoire entre 0 et RAND_MAX.
- Écrivez un programme qui dessine une grille de 20x20 cases au centre de la fenêtre.

3 Grille 3D

Écrivez un programme qui dessine une grille en 3D. En utilisant les matrices de rotations et la projection en perspective faites tourner cette grille autour de l’axe x, puis de l’axe z. On utilisera uniquement des fonctions pour effectuer ce programme. Les fonctions nécessaires sont les suivantes :

- void affiche(const double matrix[4][4]) // affiche sur la sortie standard la matrice matrix
- void affiche(const double vect[4]) // affiche sur la sortie standard le vecteur vect
- void ident(double matrix[4][4]) // initialise matrix avec la matrice identité
- void getRotateXMatrix(const double alpha, double matrix[4][4]) // construit la matrice de rotation d’angle alpha autour de Ox
- void getRotateZMatrix(const double alpha, double matrix[4][4]) // construit la matrice de rotation d’angle alpha autour de Oz
- void mulVector(double vect[4], const double matrix[4][4]) // calcule le produit du vecteur par la matrice
- void getProjectionMatrix(const double f, const double d, double matrix[4][4]) // construit la matrice de projection de focale f et de distance d ;

Une approximation consiste à n’effectuer que le calcul des quatre coins en 3D. Les autres points seront déterminés par une interpolation en 2D.

4 Particule (optionnel)

- Écrivez un programme qui affiche en 2D le déplacement d’une particule lancée du coin haut gauche de la fenêtre avec une vitesse initiale dirigée vers la droite. Lorsque la particule touche le sol faites la rebondir en diminuant sa vitesse sur y en fonction d’un coefficient d’amortissement. On ne prendra en compte que l’effet de la pesanteur.

- Réécrivez le même programme mais cette fois-ci en trois dimensions. On considérera que la particule touche le sol lorsque sa coordonnée y devient négative et on la lancera du point $[10, 10, 10]$ avec un vecteur vitesse $\vec{v} = [1, 1, 1]$.
- Faites un programme qui affiche la grille et plusieurs particules en mouvement en même temps.