Travaux pratiques

Master 1 MTI 3D

POO – Programmation Orientée Objet - C++

1. Préambule

Les travaux pratiques peuvent être réalisés sous n'importe quel environnement incluant un compilateur C++ (Visual C++, CodeBlocks, gcc sous linux, etc.). Ils comportent une étape préparatoire (notée sur 5 points) et une étape de réalisation encadrée en salle machine (notée sur 15 points). Durant l'étape préparatoire (relative à la conception), il sera demandé d'étudier le problème, de définir l'ensemble des classes, les membres statiques et dynamiques de chacune d'elle ainsi que le graphe d'héritage entre les classes. Vous veillerez aussi à amorcer l'écriture de chaque algorithme demandé (en particulier l'algorithme de remplissage d'une forme et le tri des facettes).

Ce TP porte sur la mise en œuvre de techniques d'infographie de base en utilisant les possibilités que nous offre la programmation orientée objet. Nous utiliserons, en guise de fenêtre graphique pour la visualisation d'une forme, une simple image de format PPM que nous visualiserons avec l'éditeur d'images de notre choix (suggestion : gimp). Une classe de manipulation des fichiers PPM est fournie.

2. La classe point et ses membres

Un point est caractérisé par ses coordonnées (x; y; z) ainsi qu'un ensemble d'opérations géométriques de base (la translation, la rotation, l'homothétie) et de fonctions de manipulations des membres privés (permettant de fixer ou de retourner la valeur de chaque coordonnée).

1. Ecrire tous les constructeurs possibles de la classe Point : constructeur par défaut mettant à zéro chacune des trois coordonnées ; constructeur fixant les valeurs des trois coordonnées à partir des coordonnées données en paramètre d'entrée du constructeur ; constructeur de recopie à partir d'un objet point donné en paramètre. Tester chaque constructeur mis en place.

2. Surdéfinir les opérateurs + et ==. L'addition de deux points donnés en entrée donne un point en sortie. Le test d'égalité entre deux points donnés en entrée donne un booléen en sortie.

3. Ecrire les opérations géométriques : translation d'un point, rotation d'un point, homothétie d'un point. Définir correctement les paramètres d'entrée.

Rappel : matrices de rotation en x, en y et en z

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \quad R_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & & \cos\theta \end{pmatrix}$$

$$R_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

4. Ecrire les fonctions de manipulation des membres privés permettant de fixer, et de retourner la valeur de x, y ou z

5. Ecrire les 4 fonctions qui nous retournent respectivement les points voisins 4-connexité 2D d'un point `(x;y;z)` cad : `(x-1;y)`, `(x+1;y)`, `(x;y-1)` et `(x;y+1)`.

6. Tester chaque fonction programmée. Pour cela, mettre en œuvre une nouvelle fonction membre : la fonction `afficher` qui permet d'afficher le point sur une image PPM avec une couleur donnée en paramètre.

Remarque : pour tester l'affichage, déclarer un objet "Image" de la classe Ppm, donner "Image" en entrée de la procédure d'affichage, et invoquer le membre `write` de la classe Ppm afin d'écrire l'image sur le disque (voir exemple d'utilisation).

3. La classe Cercle

Nous pouvons considérer qu'un cercle est une sorte de point de rayon nul. A partir de là, il est possible de mettre en place une relation d'héritage entre le point et le cercle.

1. Ecrire la classe Cercle. Mettre en évidence l'héritage. Quels sont les nouveaux membres propres à la classe cercle ?

2. Est-il nécessaire d'écrire les fonctions de transformation géométriques (translation, rotation) propres au cercle ?

3. Ecrire la fonction afficher propre au cercle en utilisant la propriété de surdéfinition des fonctions de C++.

Rappel : équation d'un cercle $(x - x_0)^2 + (y - y_0)^2 = R^2$

4. La classe Segment

Un segment est composé de deux points.

1. Ecrire tous les constructeurs possibles d'un segment : constructeur par défaut (les deux points sont confondus, de coordonnées nulles) ; constructeur de recopie à partir d'un segment existant ; constructeur à partir de deux points donnés en paramètre.

2. Ecrire les opérations géométriques de base. Se servir de ce qui a été écrit pour le point.

3. Ecrire l'affichage d'un segment. La fonction doit utiliser le fonction membre line de la classe Ppm.

5. La classe Facette

Une facette est composée de 4 points ou de 4 segments contraints deux à deux (l'extrémité de l'un est le commencement de l'autre).

1. Ecrire les constructeurs.

2. Ecrire les opérations géométriques de base.

3. Mettre en œuvre le tracé en fil de fer en se basant sur ce qui a été écrit pour les segments.

4. Mettre en œuvre une procédure de remplissage d'une facette avec une couleur donnée. L'algorithme choisi est récursif. Il part de l'hypothèse que la facette a déjà été tracée en

fil de fer. Il démarre avec un point à l'intérieur de la facette (germe de départ) et le colore. La procédure se rappelle ensuite elle-même pour les voisins 4-connexité du point courant jusqu'à la rencontre du contour de la facette déjà coloré.

5. Mettre en œuvre un moyen efficace de choisir le germe de départ.

   Indication : procédure de calcul de centre de gravité d'un polygone. Statut de la procédure : amitié ? encapsulation ? proposez un moyen de la protéger.

6. La classe Cube

Un cube est composé de 6 facettes.

1. Ecrire les constructeurs.
2. Ecrire les opérations géométriques.
3. Ecrire le tracé en fil de fer.
4. Réaliser le remplissage du cube : on procédera facette par facette. Obtenez-vous les résultats escomptés ? Expliquer.
5. Tester vos fonctions sur le cube suivant :

   - Les plans correspondants aux faces du cube ont pour équations : `x = 80 ; y = 80; z = 80 ; x =0 ; y = 0 ; z = 0`
   - Les facettes sont construites comme suit :

     Facette 1 : points 1 2 3 4 (dessous)

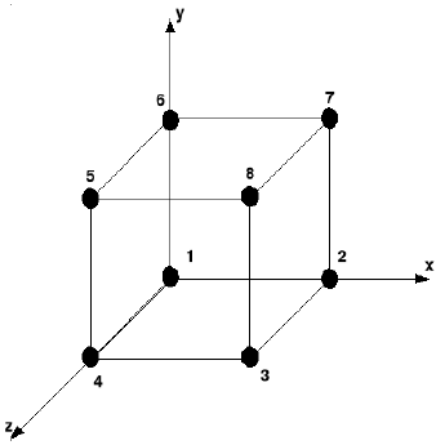
     Facette 2 : points 8 5 4 3 (face)

     Facette 3 : points 5 6 1 4 (côté gauche)

     Facette 4 : points 6 7 2 1 (arrière)

     Facette 5 : points 7 8 3 2 (côté droit)

     Facette 6 : points 5 8 7 6 (dessus)



   - Le cube sera présenté légèrement tourné. Vous effectuerez par exemple une rotation de 10° selon l'axe des x et de -35° selon l'axe des y. On testera aussi d'autres types de rotations par exemple : $\frac{\pi}{8}$ selon x et $\frac{\pi}{4}$ selon y.
   - Le cube sera translaté de 100 suivant x et 100 suivant y.

7. Algorithme du peintre

Le principe de l'algorithme est d'afficher les facettes des plus lointaines aux plus proches. En affichant les plus lointaines pour commencer, on les cache successivement en dessinant pardessus les facettes les plus proches. Nous utiliserons la procédure de remplissage précédemment écrite afin de "dessiner" chaque facette en veillant à ce qu'elle soit conçue de telle manière qu'elle puisse autoriser la croissance même si les pixels analysés ont déjà été

colorés. Pour trier nos facettes, nous devons rajouter une information supplémentaire : la distance de cette facette à l'observateur, qui sera dans notre cas la distance z moyenne de la facette. On utilisera, afin d'effectuer le tri, un tableau `tabProfondeur` où chaque élément aura deux champs : un champ contenant le numéro de la facette et un champs relatif à la profondeur moyenne de la facette. Pour tester la procédure, on reprendra l'orientation et la position du cube du départ puis on testera pour d'autres configurations au choix. Afin d'effectuer le tri des éléments du tableau, on proposera une procédure de tri.

8. Résultats escomptés

Testez vos fonctions sur le programme main suivant :

```
#include "formes.h"
#define T 80

int main()
{
    Ppm im(256,256);

    Point pt1(0,0,0,ROUGE), pt2(T,0,0,ROUGE), pt3(T,0,T,ROUGE), pt4(0,0,T,ROUGE);
    Point pt5(0,T,T,ROUGE), pt6(0,T,0,ROUGE), pt7(T,T,0,ROUGE), pt8(T,T,T,ROUGE);

    Facette fdessous(pt1,pt2,pt3,pt4,BLEU);
    Facette fface(pt8,pt5,pt4,pt3,ROUGE);
    Facette fgauche(pt5,pt6,pt1,pt4,VERT);
    Facette farriere(pt6,pt7,pt2,pt1,JAUNE);
    Facette fdroit(pt7,pt8,pt3,pt2,CYAN);
    Facette fdessus(pt5,pt8,pt7,pt6,ROSE);

    Cube C(fdessous,fface,fgauche,farriere,fdroit,fdessus);

    C.translation(100,100,0);
    C.rotation(20.,-30.,0.);
    C.afficherFilDeFer(im);

    C.remplissage(im);

    Cercle cer(10);
    cer.translation(50,50,0);
    cer.afficher(im);

    im.write("res.ppm");

    return 0;
}
```

Résultat attendu :