UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE

IFT 159
Devoir # 6
Analyse, conception et programmation

Attention :	Cette spécification sert premièrement à faire l'analyse/conception et ENSUITE la programmation de VOTRE analyse/conception.

Le but de ce travail pratique est de faire l'analyse d'un problème de difficulté moyenne, de faire la conception de sa solution en utilisant la notation UML et la méthodologie orientée-objet, puis de la programmer. L'objectif est de vous familiariser avec les notions vues dans le cours et principalement la notion de classe.

Ce travail pratique se divise en deux phases :

1. La phase 1 consiste à faire l'analyse et la conception complète du problème selon la méthodologie orientée-objet avec la notation UML. **Le document d'analyse est à remettre au plus tard le mardi 1*er* décembre 2009 à 8h30 dans les casiers situés au sous-sol du D4.**
2. La phase 2 consiste à programmer la solution que vous avez élaborée lors de la phase 1. **Le programme est à remettre au plus tard le lundi 7 décembre 2009 à 23h59. La remise devra être faite grâce à la commande *turnin*.**

Sujet : Le logiciel de dessin.

On vous demande d'écrire un programme qui implante un logiciel de dessin. Ce logiciel permet de dessiner et de manipuler des figures géométriques (cercle, rectangle et triangle). Les manipulations sont :

- le déplacement ;
- la rotation ;
- le changement de taille.

Votre programme demande de façon répétitive le type de manipulation à effectuer. Les manipulations requises sont :

- «a» pour ajouter une figure ;
  Lorsque cette commande est donnée, on demande le type de figure (cercle (1), rectangle (2) ou triangle (3)) et les coordonnées nécessaires pour la dessiner. On doit donc spécifier le point central du cercle et son rayon, deux points pour le rectangle et trois points pour le triangle.
- «l» pour obtenir la liste des figures ;

Lorsque cette commande est donnée, la liste de toutes les figures est affichée à l'écran. Pour chaque figure, on affiche son numéro dans la liste, son type et la valeur de ses propriétés. Le numéro de la liste et le type sont requis pour les autres commandes.

- «e» pour effacer une figure ;
  Lorsque cette commande est donnée, on doit déterminer la figure que l'on veut effacer (son type et son numéro).
- «r» pour la rotation ;
  Lorsque cette commande est donnée, on doit déterminer la figure sur laquelle on l'applique (son type et son numéro), le point central de rotation et l'angle de rotation.
- «d» pour déplacement ;
  Lorsque cette commande est donnée, on doit déterminer la figure sur laquelle on l'applique (son type et son numéro) ainsi que la distance de déplacement en x et en y.
- «t» pour la modification de la taille ;
  Lorsque cette commande est donnée, on doit déterminer la figure sur laquelle on l'applique (son type et son numéro), le point central de la modification et la proportion.

Pour implanter votre programme, vous disposez d'une classe « `Canevas` » (`canevas.h` et `canevas.cpp`) qui se trouve sur le site Web du cours. Cette classe fournit un canevas de dessin, une fonction pour effacer le canevas et les opérations (méthodes) pour aller chercher les valeurs des différentes propriétés du canevas.

Le diagramme de classe de ce canevas est :

Canevas
-tailleX: entier -tailleY: entier -couleurDeFond: entier -identificateurFenetre: entier
+Canevas(tailleX:entier,tailleY:entier,couleurFond:entier) +Canevas(tailleX:entier,tailleY:entier) +Canevas(couleurFond:entier) +Canevas() +effacer(): vide +valX(): entier +valY(): entier +valIdentificateur(): entier +valCouleur(): entier

De plus, vous disposer d'une classe « `Ligne` » (`ligne.h` et `ligne.cpp`) qui se trouve sur le site Web du cours. Cette classe fournit une ligne qui possède deux points et une couleur, une fonction pour afficher la ligne sur un canevas, des fonctions pour changer les points et la couleur de la ligne et les fonctions pour aller chercher les valeurs des points et de la couleur.

Le diagramme de classe de cette ligne est :

Ligne
+point1: Point +point2: Point +couleur: entier
+Ligne(debut:Point,fin:Point,couleur:entier) +Ligne(debut:Point,fin:Point) +Ligne(couleur:entier) +Ligne() +afficher(ecran:Canevas): void +valPoint1(): Point +valPoint2(): Point +valCouleur(): entier +changePoint1(nouveauPoint:Point): void +changePoint2(nouveauPoint:Point): void +changeCouleur(couleur:entier): void +init(debut:Point,fin:Point,couleur:entier): void +init(debut:Point,fin:Point): void

Vous devez aussi implanter un type abstrait « `Point` » dont la spécification est donné dans `point.h` et est exprimée par le diagramme de classe suivant :

Point
-coordonneeX: entier -coordonneeY: entier
+Point() +Point(x:entier,y:entier) +init(in x:entier,in y:entier): vide +lire(): vide +afficher(): vide +rotation(centre:point,angleRadian:réel): vide +deplacer(in deplX:entier,in delpY:entier): vide +deplacerProp(ancre:point,proportion:réel): vide +valX(): entier +valY(): entier

Compilation

Pour compiler vous devez utiliser la commande suivante :

```
g++ -otp6 tp6.cpp canevas.cpp ligne.cpp point.cpp ... -lm -lX11 -lg2
```

Vous devez obligatoirement travailler sur Rigel. Vous pouvez aussi travailler sur un ordinateur Linux ou Windows à condition d'y installer et de configurer la bibliothèque graphique `g2`. Vous trouverez la bibliothèque `g2` à l'adresse `http://g2.sourceforge.net/`.

Remise de la partie analyse et conception

Remettre, sur des feuilles séparées mais brochées ensemble et clairement identifiées (noms, matricules **et codes d'usager**), votre analyse et conception orientée objet. On devrait y retrouver l'analyse globale, le diagramme d'utilisation, les diagrammes de classes, les diagrammes de séquences de messages et l'analyse/conception détaillée des classes et de leurs méthodes.

Soumission du programme

Vous devez me soumettre avant la date limite, à l'aide de la commande

turnin -cift159 -ptp6 tp6

un seul **répertoire** portant OBLIGATOIREMENT le nom **tp6** qui contient le programme demandé (tous les fichiers ".h" et ".cpp"), le fichier **modifications.txt** qui contient les modifications apportées à votre analyse/conception originale, et trois fichiers tests (**test1, test2, test3**). Étant donné que la correction est partiellement automatisée, une mauvaise soumission entraîne une mauvaise note (possiblement 0). Il n'y a pas de recorrection.

Pour vous éviter de mauvaises surprises, faites un test d'essai dans les mêmes conditions que le train de travaux. Vous devez faire imprimer votre fichier source (ou vos fichiers sources) et exécuter votre programme comme suit :

```
tp6 < test1
```

où "test1" est un de vos fichiers de données. Évidemment "tp6" est le fichier qui contient votre programme exécutable.

Pour vérifier si le programme a bien été soumis, vous pouvez faire la commande :

turnin -v -cift159 -ptp6

Quelques notions d'infographie

1. Dessiner un cercle ;

   Il est possible de dessiner un cercle en dessinant des lignes comme illustré à la figure 1. On sépare donc notre cercle en un certain nombre d'angles. Si on débute à l'angle $\alpha = 0$ et que l'on progresse par incrément de ϵ, on calcule les deux points (P1 et P2) de l'extrémité du triangle de la façon suivante :
   - P1(x) = C(x) + (rayon $\times cos(\alpha)$)
   - P1(y) = C(y) + (rayon $\times sin(\alpha)$)
   - P2(x) = C(x) + (rayon $\times cos(\alpha + \epsilon)$)
   - P2(y) = C(y) + (rayon $\times sin(\alpha + \epsilon)$)

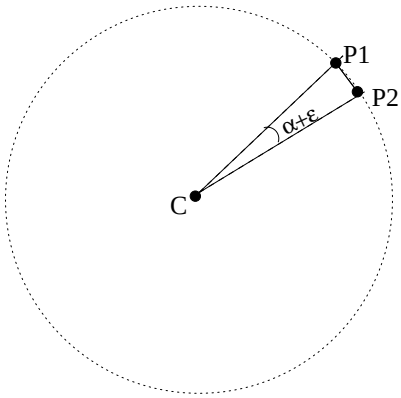

FIG. 1 – technique pour dessiner un cercle

2. Effectuer une rotation ;

   Pour faire tourner une figure (pas le cercle) autour d'un centre C d'un angle α, on utilise tous les points qui la composent et on les déplace de la façon suivante :
   - on déplace la figure de façon à ce que le point C soit à (0,0) ;
   - on calcule la rotation de chaque point grâce aux formules suivantes
     $$x' = x \times sin(\alpha) + y \times cos(\alpha)$$
     $$y' = x \times cos(\alpha) - y \times sin(\alpha)$$
   - on ramène la figure à sa position de départ.

3. Effectuer un déplacement proportionnel relatif à un point C (changement de taille).

   Pour déplacer un point P par rapport à un autre point C à partir d'une proportion δ, on fait :
   $$x' = (P(x) - C(x)) \times \delta + C(x)$$
   $$y' = (P(y) - C(y)) \times \delta + C(y)$$