**Compte Rendu TP4 C++ : Héritage, polymorphisme**

**Document de spécification et de conception**

**I\ Spécifications**

Le but de ce TP est de manipuler des figures géométriques différentes. L’utilisateur utilisera l’application à travers des lignes de commandes. Il pourra ainsi créer, modifier et supprimer des figures. Il lui sera aussi possible de sauvegarder son travail ou de le charger.

**Définitions**

**Cercle :** Figure géométrique caractérisé par son origine et son rayon

**Rectangle :** Figure géométrique caractérisé par deux points : son coin supérieur gauche et son coin inférieur droit.

**Ligne :** Figure géométrique caractérisé par deux points.

**Polyligne :** Figure géométrique caractérisé par n points : Chaque doublet de points peut être assimilé à une ligne.

**Figure :** Ensemble regroupant les quatre formes énoncées ci-dessus.

**Sélection :** Objet capable d’englober plusieurs figures.

**Container :** Espace dans lequel toutes nos figures et sélections sont créées.

**Graphics :** Ensemble regroupant les figures et les sélections

**Commande :**

**Choix généraux**

Chaque figure possède des attributs qui lui sont propres : Le cercle possède un rayon et un centre, le rectangle possède deux points…

De plus, chaque figure est aussi caractérisée par un rectangle dans lequel elle est inscrite. Cela permet de savoir si une figure se trouve entièrement incluse dans une sélection.

**II\ Conception**

Nous avons choisi une structure composée de … classes

**La classe main:** Gère la saisie des commandes saisies par l’utilisateur et fait appel aux méthodes de la classe container en fonction de celles-ci.

**La classe Container:**

**La classe Graphics:**

**La classe Figure:**

**La classe Selection:**

**La classe Circle:**

**La classe Rectangle:**

**La classe Line:**

**La classe Polyline:**

**La classe Command:**

**La classe ComposedCommand:**

**La classe SimpleCommand:**

**La classe DeleteCommand:**

**La classe DeleteElementCommand:**

**La classe MoveCommand:**

**La classe LoadCommand:**

**La classe Tools:**

**Structures de données et algorithmes**

**Structure de donnée**: On a choisi d’utiliser la structure arbre pour stocker les données de la classe graphe en raison de la complexité algorithme de recherche d’un élément, la non-redondance.

**Diagramme UML :**

**Spécification des tests fonctionnels**

Index des tests :

0\_x=Test de création

1\_x=Test de déplacement

2\_x=Test de suppression

3\_x=Test UNDO/REDO

4\_x=Test LOAD/SAVE

5\_x=Cas limites/ cas d’erreur

**Test 0\_1 :** Création d’un cercle

On vérifie que le cercle à bien été créé avec les bons paramètres

**Test 0\_2 :** Création rectangle

On vérifie que le rectangle à bien été créé avec les bons paramètres

**Test 0\_3 :** Création ligne

On vérifie que la ligne à bien été créée avec les bons paramètres

**Test 0\_4 :** Création polyligne

On vérifie que le polyligne à bien été créé avec les bons paramètres

**Test 0\_5 :** Création sélection

On vérifie que la sélection a bien été créée et qu’elle n’apparait pas si on LIST

**Test 0\_6 :** Création sélection vide

On vérifie que la création d’une sélection vide ne pose aucun problème

**Test 1\_1 :** Déplacement figure

On vérifie que les coordonnées de la figure passée en paramètre ont été incrémentées de dx et dy

**Test 1\_ 2:** Déplacement sélection

On vérifie que les coordonnées de toutes les figures présentes dans la sélection passée en paramètre ont été incrémentées de dx et dy

**Test 1\_3 :** Déplacement d’une figure qui se trouve dans une sélection

On vérifie que les coordonnées de la figure passée en paramètre ont été incrémentées de dx et dy mais que les autres figures de la sélection n’ont pas bougé

**Test 1\_4 :** Déplacement d’une sélection qui englobe une autre sélection

On vérifie que les coordonnées de toutes les figures présentes dans les deux sélections ont été incrémentées de dx et dy

**Test 1\_5 :** Déplacement d’une sélection qui ne contient pas entièrement une figure

On vérifie que les coordonnées de la figure ne figurant pas entièrement dans la sélection n’ont pas été incrémentées

**Test 1\_6 :** Déplacement d’une figure qui se trouve dans une sélection

On vérifie que les coordonnées de la figure passée en paramètre ont été incrémentées de dx et dy mais que les autres figures de la sélection n’ont pas bougé

**Test 1\_7 :** Déplacement d’une figure contenue dans une sélection qui vient à dépasser les limites de la sélection

On vérifie que les coordonnées de la figure sont correctes, que seule cette figure s’est déplacée et qu’elle est toujours présente dans la sélection

**Test 2\_1 :** Suppression figure

On vérifie que la figure passée en paramètre ne fait plus partie de la liste des figures présentes dans le container

**Test 2\_2 :** Suppression plusieurs figures

On vérifie que les figures passées en paramètre ne font plus partie de la liste des figures présentes dans le container

**Test 2\_3 :** Suppression sélection

On vérifie que les figures présentes dans la sélection passée en paramètre ne font plus partie de la liste des figures présentes dans le container

**Test 2\_4:** Suppression figure faisant partie d’une sélection

On vérifie que la figure passée en paramètre ne fait plus partie de la liste des figures présentes dans le container mais que la sélection existe toujours

**Test 2\_5 :** Suppression d’une sélection suivie de la suppression d’une figure qui faisait partie de cette sélection dans la même commande ?

On vérifie que toutes les figures présentes dans la sélection ont été retirées de la liste des figures présentes dans le container comme si la sélection seule avait été supprimée

**Test 3\_1 :** UNDO création figure

On vérifie que la figure créée précédemment ne fait plus partie de la liste des figures présentes dans le container

**Test 3\_2 :** UNDO déplacement figure

On vérifie que la figure a récupéré les mêmes coordonnées qu’elle possédait avant son déplacement

**Test 3\_3 :** UNDO suppression figure

On vérifie que la figure fait a nouveau partie de la liste des figures présentes dans le container

**Test 3\_4 :** UNDO CLEAR

On vérifie que toutes les figures qui ont été supprimées par le CLEAR font à nouveau partie de la liste des figures présentes dans le container

**Test 3\_5 :** UNDO LOAD

On vérifie que toutes les figures introduites dans le container par le LOAD ont disparues

**Test 3\_6 :** UNDO SAVE

On vérifie que le fichier dans lequel la sauvegarde a été effectuée a disparu

**Test 3\_7 :** UNDO création sélection

On vérifie que le UNDO s’applique à la commande précédant la création de la sélection car une sélection n’est pas undo-able

**Test 3\_8 :** UNDO suppression figure qui faisait partie d’une sélection

On vérifie que la figure est bien revenue mais qu’elle ne fait plus partie de la sélection

**Test 3\_9 :** UNDO suivi d’une commande puis d’un REDO

On vérifie que l’on ne peut pas REDO

**Test 3\_10 :** 21 UNDO

On vérifie que l’on ne peut pas faire le 21ème UNDO

**Test 4\_1 :** SAVE

On vérifie qu’un nouveau fichier a été créé et qu’il contient les commandes associées aux Graphics présents dans le container

**Test 4\_2 :** LOAD

On vérifie que les commandes contenues dans le fichier sont exécutées dans le container

**Test 5\_1 :** Pas le bon nombre de paramètres

On obtient un message d’erreur et un commentaire explicatif

**Test 5\_2 :** Rayon négatif

On obtient un message d’erreur et un commentaire explicatif

**Test 5\_3 :** SAVE avec nom de fichier déjà existant

On obtient un message d’erreur et un commentaire explicatif

**Test 5\_4 :** LOAD avec un nom de fichier inexistant

On obtient un message d’erreur et un commentaire explicatif

**Test 5\_5 :** Nom déjà pris

On obtient un message d’erreur et un commentaire explicatif

**Test 5\_6 :** DELETE dont un des noms n’existe pas

On vérifie qu’aucun des noms valides de la commande n’a été supprimé. On obtient un message d’erreur et un commentaire explicatif

Faire un tests corner rectangle + 21 REDO