

3 IF 2014/2015

Andrea ACCARDO

Cyril CANETE

B3230

Compte Rendu

TP C++ n°4 : Héritage, polymorphisme

Application DrawController

Dessin vectoriel en ligne de commande

DESCRIPTION

L'application DrawController permet de gérer des formes dans un repère. L'utilisateur peut insérer, retirer et déplacer ces formes. Il peut également créer des sélections afin d'appliquer une actions à un groupe de formes. Les formes gérées sont le cercle, le rectangle, les lignes et les polygones. Une fonction d'annulation est également disponible et permet à l'utilisateur d'annuler jusqu'à 20 actions. La gestion de ces formes se fait via l'invite de commande. Chaque commande entrée par l'utilisateur est suivie d'un message de validation ou d'erreur.

COMMANDES UTILISATEUR

Après avoir démarrer l'application via la commande *tpheritage* depuis le shell, l'utilisateur peut interagir directement avec le programme.

AJOUT DE FORMES ET DE SELECTIONS

- **C Name X1 Y1 R** : *Crée un nouveau Cercle*
 - **Name** : Chaine de caractère (sans espace), Identifiant du cercle
 - **X1** : Entier, Coordonnée X du centre du cercle
 - **Y1** : Entier, Coordonnée Y du centre du cercle
 - **R** : Entier, Rayon du cercle
 - **RETOUR** : OK si l'ajout à réussi, ERR + #Type Erreur, si une erreur est survenue.
- **R Name X1 Y1 X2 Y2** : *Crée un nouveau Rectangle*
 - **Name** : Chaine de caractère (sans espace), Identifiant du rectangle
 - **X1** : Entier, Coordonnée X du premier point diagonal du rectangle
 - **Y1** : Entier, Coordonnée Y du premier point diagonal du rectangle
 - **X2** : Entier, Coordonnée X du second point diagonal du rectangle
 - **Y2** : Entier, Coordonnée Y du second point diagonal du rectangle
 - **RETOUR** : OK si l'ajout à réussi, ERR + #Type Erreur, si une erreur est survenue.
- **L Name X1 Y1 X2 Y2** : *Crée une nouvelle Ligne*
 - **Name** : Chaine de caractère (sans espace), Identifiant de la ligne
 - **X1** : Entier, Coordonnée X de la première extrémité de la ligne
 - **Y1** : Entier, Coordonnée Y de la première extrémité de la ligne
 - **X2** : Entier, Coordonnée X de la seconde extrémité de la ligne
 - **Y2** : Entier, Coordonnée Y de la seconde extrémité de la ligne
 - **RETOUR** : OK si l'ajout à réussi, ERR + #Type Erreur, si une erreur est survenue.

- **PL Name X1 Y1 Xn Yn** : Crée une nouvelle Poly-line
Remarque: n, entier strictement positif
 - **Name** : Chaine de caractère (sans espace), Identifiant de la ligne
 - **Xi** : Entier, Coordonnée X du i-ème extrémité point de la poly-ligne
 - **Yi** : Entier, Coordonnée Y du nième extrémité point de la poly-ligne
 - **RETOUR** : OK si l'ajout à réussi, ERR + #Type Erreur, si une erreur est survenue.
- **R Name X1 Y1 X2 Y2** : Crée une nouvelle Sélection
 - **Name** : Chaine de caractère (sans espace), Identifiant de la sélection
 - **X1** : Entier, Coordonnée X du premier point diagonal de la sélection
 - **Y1** : Entier, Coordonnée Y du premier point diagonal de la sélection
 - **X2** : Entier, Coordonnée X du second point diagonal de la sélection
 - **Y2** : Entier, Coordonnée Y du second point diagonal de la sélection
 - **RETOUR** : OK si l'ajout à réussi, ERR + #Type Erreur, si une erreur est survenue.

FONCTIONS DE GESTION DES FORMES ET SELECTIONS

- **DELETE Name1 Name2...NameN** : Supprime les objets « Name1 » à « NameN »
Remarque: n, entier strictement positif
 - **Name(i)** : Chaine de caractère, Identifiant de l'objet/sélection à supprimer
Remarque: Si Name(i) est une sélection, tous les objets de la sélection sont supprimés.
 - **RETOUR** : OK si l'ajout à réussi, ERR + #Type Erreur, si une erreur est survenue.
- **MOVE Name dX dY** : Déplace de dX, dY l'objet « Name »
 - **Name** : Chaine de caractère, Identifiant de l'objet/sélection à déplacer
Remarque: Si Name est une sélection, tous les objets de la sélection sont déplacés.
 - **dX** : Entier, valeur de déplacement selon d'axe X
 - **dY** : Entier, valeur de déplacement selon d'axe Y
 - **RETOUR** : OK si l'ajout à réussi, ERR + #Type Erreur, si une erreur est survenue.
- **LIST** : Enumère tous les objets du repère
 - **RETOUR** : Liste des objets présents dans le repère sous la forme de la commande nécessaire à sa création, liste par ordre alphabétique des identifiants.
- **UNDO** : Rétabli l'état du repère précédant la dernière commande
Remarque: une sélection n'est pas annulable
 - **RETOUR** : OK

- **REDO** : Rétabli l'état du repère suivant le dernier UNDO
Remarque: en cas de nouvelle action de l'utilisateur, la fonction REDO est inaccessible.
 - *RETOUR : OK*
- **LOAD nomfichier.txt** : Ajoute un ensemble d'objet contenu dans le fichier « nomfichier.txt » au repère
 - *RETOUR : OK*
- **SAVE nomfichier.txt** : Sauvegarde l'ensemble des objets dans le fichier « nomfichier.txt »
Remarque: les sélections ne sont pas sauvegardées.
 - *RETOUR : OK*
- **CLEAR** : Supprime l'ensemble des objets et sélections du repère
 - *RETOUR : OK*
- **EXIT** : Ferme l'application
 - *RETOUR : OK*

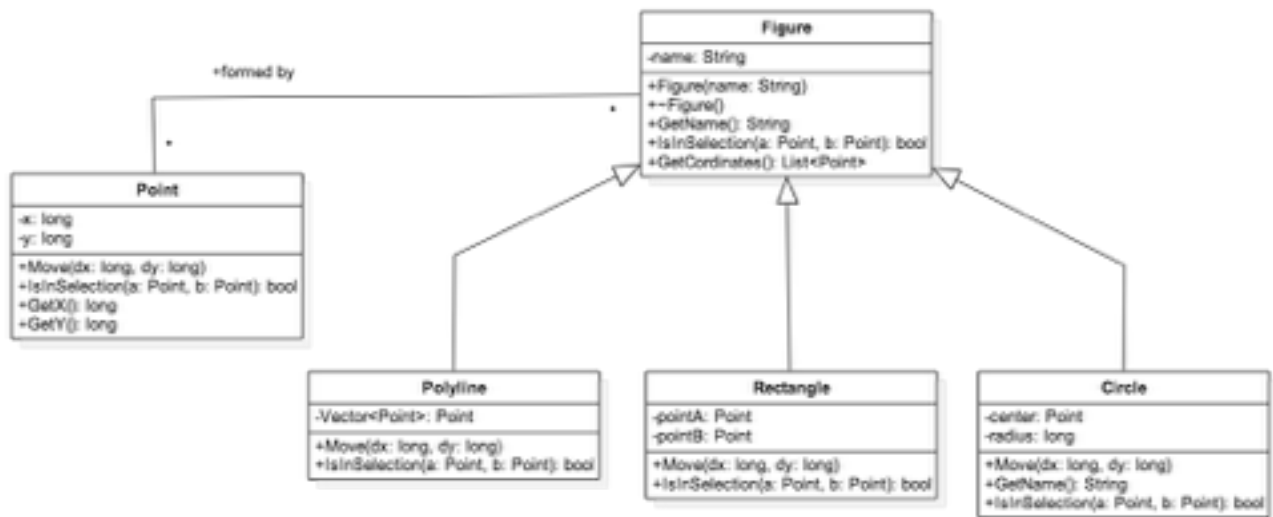
Conception

Détails de conception de l'application

INTRODUCTION

La conception de l'application DrawController est divisée en deux parties: l'ensemble *models* et l'ensemble *Comand*, conçu d'après le design-pattern Command.

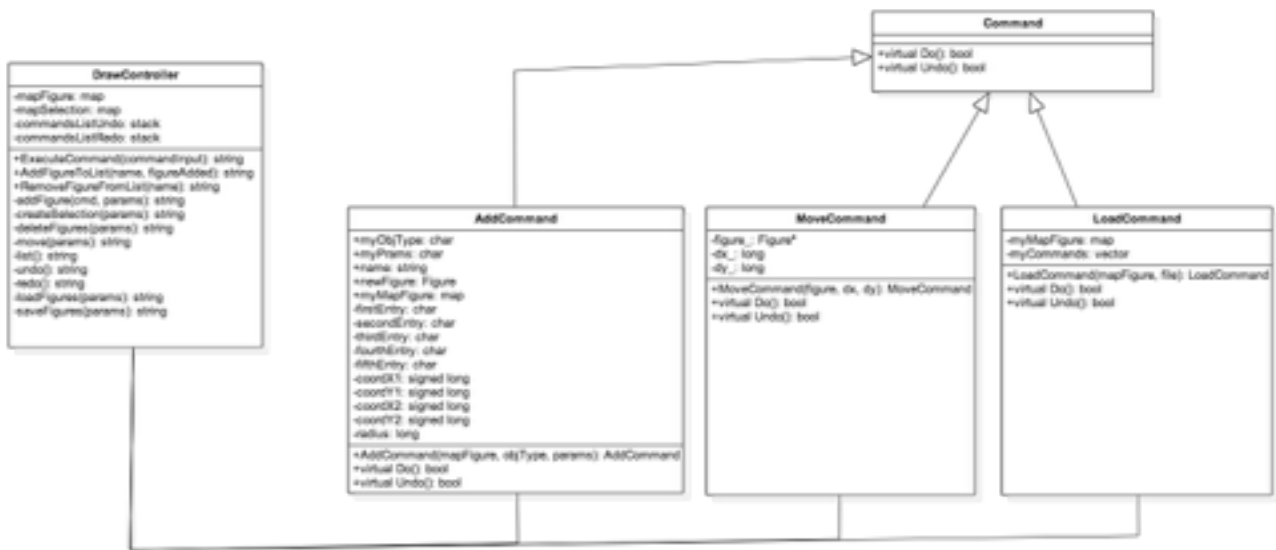
MODEL VIEW



Chaque objet est hérité de la classe abstraite *Figure* imposant la présence d'une fonction *Move(x, y)*, *IsInSelection()* et d'un nom quel que soit le type de la figure.

Le *Point* est une classe simple contenant des coordonnées (x,y).

CONTROLLER VIEW



La vue contrôleur s'articule autour de DrawControleur qui effectue la gestion des commandes utilisateur.

DRAWCONTROLLER

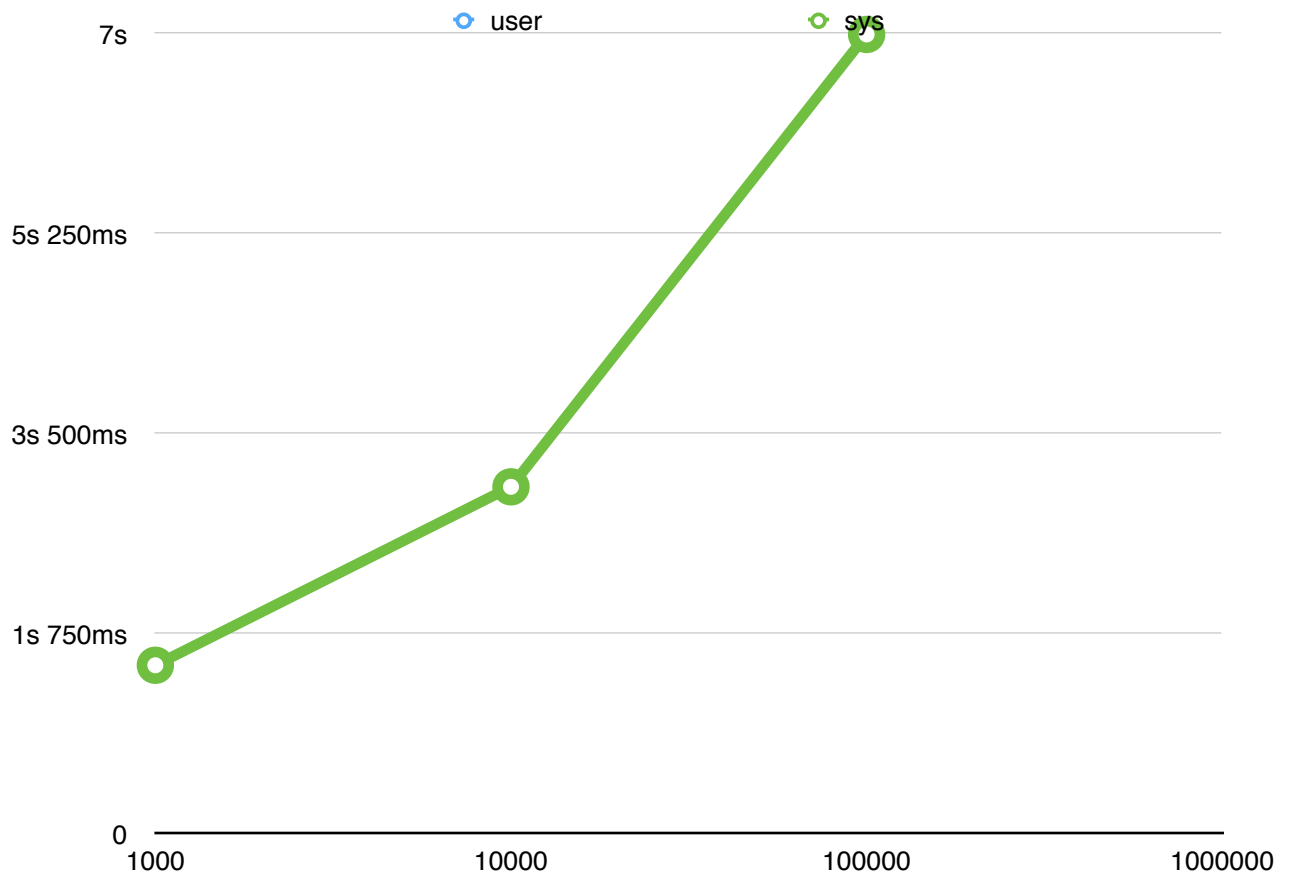
- **ExecuteCommand** : Analyse la commande entrée par l'utilisateur et appelle la méthode correspondante
- **addFigure**, **deleteFigure**, **move**, **list**, **undo**, **redo**, **loadFigures**, **saveFigures** : vérifie la syntaxe de la commande, crée l'objet *Command* correspondant et l'exécute.

Le design-pattern *command* permet la gestion de l'historique des commandes, sauvegardées dans deux structures de données de type piles.

Chaque commande « réversible » est une classe issue de la classe abstraite *Command*, possède un constructeur et un destructeur ainsi qu'une méthode `Do()` et `Undo()` qui fait et défait l'action demandée.

Performances

Détails des performances de l'application



Les résultats des tests de performances système sont presque linéaires.

Document de conception

Spécification complètes des classes

Ici on peut trouver tous les tests effectués et leurs relatives adresses :

I. <i>Description</i> : Création d'un cercle	Test_Cercle : 1. run : Syntaxe correcte ; 2. Test_InvalideCenter/run : Valeur X et/ou Y non valides ; 3. Test_InvalideRadius/run : Valeur R non valide ; 4. Test_NameMissing/run : Nom manquante.
II. <i>Description</i> : Création d'une ligne	Test_Line : 1. run : Syntaxe correcte ; 2. Test_InvalidePoints/run : Valeurs X ₁ , Y ₁ , X ₂ et/ou Y ₂ non valides ; 3. Test_NameMissing/run : Nom manquante.
III. <i>Description</i> : Création d'un rectangle	Test_Rectangle: 4. run : Syntaxe correcte ; 5. Test_InvalidePoints/run : Valeurs X ₁ , Y ₁ , X ₂ et/ou Y ₂ non valides ; 6. Test_NameMissing/run : Nom manquante.
IV. <i>Description</i> : Création d'un poly-ligne	Test_Polyline: 7. run : Syntaxe correcte ; 8. Test_InvalidePoints/run : Valeurs X ₁ , Y ₁ , X ₂ , Y ₂ , ... X _n et/ou Y _n non valides ; 9. Test_NameMissing/run : Nom manquante.
V. <i>Description</i> : Création d'une sélection	Test_Selection: 1. run : Syntaxe correcte ; 2. Test_InvalidePoints/run : Valeurs X ₁ , Y ₁ , X ₂ et/ou Y ₂ non valides ; 3. Test_NameMissing/run : Nom manquante.
VI. <i>Description</i> : Affichage des objets existants	Test_List: run : Affichage correcte (ignore les sélections).
VII. <i>Description</i> : Suppression de plusieurs objets	Test_Delete: 4. run : Syntaxe 5.
VIII. <i>Description</i> : Déplacement d'un objet.	Test_Move: 1. run : Syntaxe correcte ; 2. Test_InvalidePoints/run : Valeurs δX et/ou δY non valides ; 3. Test_NameMissing/run : Nom manquante.
IX. <i>Description</i> : Annulation de la dernière commande	Test_Undo: run : annule correctement toute commande sauf celles qui concernant les sélection.
X. <i>Description</i> : Rétablir la dernière commande annulée	Test_Redo: run : rétablit correctement la dernière commande sauf si après le UNDO on a exécuté un autre commande.

XI. <i>Description</i> : Suppression de tous les objets	Test_Clear: run : suppression de tous les objets (remise à zéro).
XII. <i>Description</i> : Sauvegarde d'un fichier	Test_Save: 1. run : Syntaxe correcte ; 2. Test_NameMissing/run : Nom manquante ; 3. Test_ExistentFile/run : Fichier déjà existant.
XIII. <i>Description</i> : Chargement d'un fichier	Test_Load: 1. run : Syntaxe correcte ; 2. Test_NameMissing/run : Nom manquante ; 3. Test_FileMissing/run : Fichier inexistant.