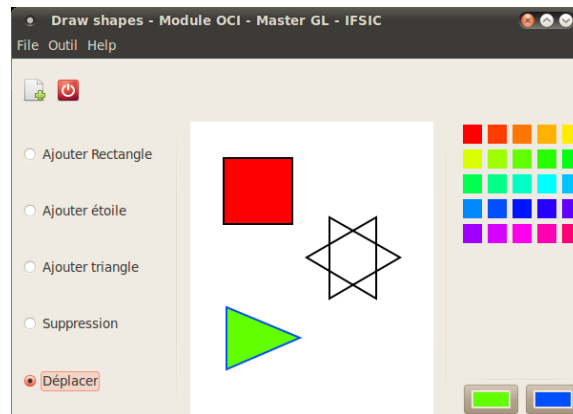


Cahier des charges



1 Introduction

Les TP précédents ont permis de réaliser des versions intermédiaires fonctionnelles de l'application de dessin. Néanmoins, ces versions ne sont absolument pas réfléchies en terme de conception logiciel. Ceci est donc intellectuellement très peu satisfaisant. En particulier, la gestion effective des outils de dessin et des couleurs restent complètement à faire.

Il est donc important de structurer le fonctionnement du logiciel. En particulier, une séparation entre la couche d'abstraction et la couche présentation (je reprends ici le formalisme PAC ou MVC sans en demander une application stricte) est fortement conseillée. Une telle approche ne peut être que bénéfique dans la situation de création d'un système avec plugins.

Dans la suite de ce document, on donne quelques descriptions (un peu trop vagues pour être appelées spécifications) de ce qui est attendu de l'application pour ce projet du module OCI.

2 Fonctionnalités générales de l'application

L'application est un éditeur interactif de dessin vectoriel. Les dessins possibles dans cette interface sont mis à disposition sous la forme d'« outils ». Dans l'exemple, on a les outils de création de rectangle, d'étoile et de triangle. Ces formes sont, par défaut, pleines ou creuses (donc il faut deux outils différents pour gérer, par exemple, un rectangle plein et un rectangle creux). En plus des outils de création de forme, il faut mettre à disposition les outils « génériques » suivants :

- ▶ la suppression de forme,
- ▶ le déplacement de forme

Cette application doit proposer une possibilité de développer des *plugins* d'outils.

Dans la conception de votre logiciel, il faudra distinguer au mieux ce qui est de l'ordre du modèle et qui est de l'ordre de la présentation.

Il est tout à fait possible de compléter votre application avec de nouvelles fonctionnalités :

- ▶ enregistrement et chargement d'un fichier de dessin,
- ▶ implémentation de formes de différents types (sous forme de *plugins*)

- ▶ implémentation d'outils génériques (sous formes de *plugins*): par exemple, le changement de position suivant Z.
- ▶ ...

L'utilisation d'optimisations sur la gestion de l'affichage de la zone de dessin serait apprécié.

3 Description de l'IHM

A Description de l'apparence de l'application

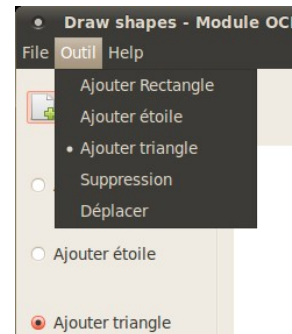
L'apparence proposée est indicative. L'organisation générale de la fenêtre peut être améliorée. Les éléments suivants doivent apparaître dans l'interface :

- ▶ Une zone pour la sélection de l'outil courant,
- ▶ Une zone pour le dessin effectif, en fonction de l'outil utilisé, l'icône de la souris pourra changer (mettre un + lors de l'ajout d'une forme, une X lors de la suppression et une main pour le déplacement),
- ▶ Une zone pour le choix de la couleur avec la possibilité de conserver quelques couleurs utilisables facilement. Les ColorButton (en bas) donne les couleurs avec lesquels dessiner une nouvelle forme.
- ▶ Un menu et une barre d'outil.

Le menu devra être composé d'une partie `File` avec au moins un sous-menu pour quitter l'application et un sous-menu pour effacer le contenu actuel d'un dessin, et un menu `Help` pourra contenir un sous-menu pour accéder à une fenêtre `APropos`.

Le menu `Outils` devra contenir la liste des outils disponibles dans l'interface. Les sous-menus seront « sélectionnables » et seront synchronisés avec la sélection de l'outil visualisé sur l'interface principale.

La barre d'outil devra contenir quelques fonctionnalités, et pourra également contenir le choix de l'outil courant (synchronisé avec les autres affichages).



Il est tout à fait possible de compléter votre interface avec plus d'éléments facilitant l'interaction :

- ▶ mettre des *tool tips*,
- ▶ mettre des touches de raccourcis,
- ▶ ...

B Description des interactions

- ▶ Il faut absolument donner une représentation continue du rectangle lors de l'interaction, c'est-à-dire le dessiner lors du déplacement de la souris bouton appuyé et non pas seulement lors du relâchement du bouton.

- ▶ Il n'est pas demandé d'avoir la possibilité d'interrompre la création d'une forme alors qu'un clic a déjà été fait ... l'utilisateur devra finir de créer sa forme et la supprimer ensuite.
- ▶ Pour les outils « générique » agissant sur les formes en place, l'interaction avec une forme ne sera possible que si on clique exactement au dessus d'une forme visible. En particulier, pour les formes autres que le rectangle qui ne couvrent pas leur emprise, il faut que l'interaction se fasse uniquement sur la forme dessinée. Et si deux formes se superposent, uniquement celle du premier plan doit être modifiée.
- ▶ Il doit être possible de faire des DnD de couleurs entre :
 - ▶ les couleurs par défaut et les deux `ColorButton` (et réciproquement),
 - ▶ les couleurs par défaut ou les `ColorButton` et les formes,
 - ▶ une couleur extérieure à l'application et les couleurs par défaut et les `ColorButton`.

L'effet d'un DnD sur une forme déjà en place sera de modifier la couleur de remplissage d'une forme (pas de modification de la couleur de trait). Si une forme est creuse, ca n'aura aucun effet sur elle.

- ▶ Pour les interactions du `ColorSelector`, se référer au TP1
- ▶ Le sous-menu `APropos` fera apparaître une fenêtre modale indiquant les noms du groupe et les références du projet (Module OCI, M2 GL - ...)

4 *Format de retour et documentation*

Votre travail doit être retourné par mail dans les délais impartis (délais à préciser). Il est impératif que votre code soit fonctionnel et propre :

- ▶ Le contenu de votre archive (`zip`, `tar.gz`, ou `rar`, évitez de `7z`) doit être propre : sans fichiers objet et prêts à compiler (joindre les fichiers `Cmake`, `Makefile`, `Eclipse` ou `Code::Block`)
- ▶ Le code doit compiler,
- ▶ La réalisation doit être fonctionnelle.

Il est préférable de se limiter à quelques fonctionnalités plutôt que de fournir un code non-fonctionnel même s'il aborde pleins de points du sujet.

En plus du code lui-même, il vous est demandé de fournir les éléments de documentation suivants :

- ▶ Le code sera indenté, propre et commenté,
- ▶ Un document (1 à 2 pages pour la partie `Gtkmm`) présentera les solutions retenues d'utilisation des signaux (et argumentera les choix). Certains points spécifiques de votre implémentation pourront également être détaillés.