

Grille de rendu du projet S1102

NOM 1 : Thomas Auffroy

NOM 2 : Alexandre Laurac

1. Rendu de la version finale

Le projet doit être rendu avant le **vendredi 11 juin 2021 minuit** sur le site Chamilo dans le module S1102 à la rubrique "Travaux". Vous pouvez toujours modifier votre dépôt jusqu'à la date buttoir.

Le rendu doit comporter :

- Ce document correctement rempli **au format PDF**.
- Le code source complet documenté.

Tout le projet doit être soumis sous la forme **d'un seul fichier compressé** au format ZIP ou GZIP comportant le nom des deux participants (ex : *dupont_durand.zip*).

Vérifiez après votre dépôt que le fichier déposé est correct en le téléchargeant, en le décompressant dans /tmp et en l'exécutant à cet endroit.

1. Informations générales (utiliser la place nécessaire)

Indiquez ici les renseignements pour l'utilisateur : que permet globalement de faire votre logiciel.

Notre logiciel permet de réaliser les actions suivantes :

- tracé de formes avec couleurs

Il y a 3 types de formes – rectangle, ovale, ligne (on a décidé de ne pas en implémenter plus pour consacrer du temps à des fonctionnalités de nature différente. Mais une forme de type triangle, notamment, était envisagée). Pour dessiner une forme, il s'agit d'abord de cliquer sur le bouton comportant le nom correspondant à celle-ci dans le panneau de contrôle à gauche de l'interface.

Il est possible de choisir la couleur de la bordure et celle de l'intérieur de la forme.

- sélection d'une forme (la couleur de sa bordure est modifiée)

La suppression de la forme correspondante (par double clic) a été implémentée mais n'est pas fonctionnelle

- déplacement d'une forme après sélection de celle-ci.

- sauvegarde du dessin réalisé dans l'interface au format svg (le fichier sauvegardé est visualisable dans un logiciel adapté).

La lecture d'un fichier SVG a été implémentée mais n'est pas fonctionnelle.

A partir de ces quelques fonctionnalités, on peut réaliser des dessins tout à fait honnêtes.

2. Informations techniques (utiliser la place nécessaire)

Indiquez ici quelle structure de données est utilisée pour chaque type d'"objet".

Indiquez les particularités éventuelles de votre code, de votre approche, ...

Le code a été écrit selon le paradigme MVC.

Pour la partie vue, différentes classes ont été écrites :

- une classe MyFrame dérivant de wxFrame → implémente une fenêtre
- une classe MyControlPanel dérivant de wxPanel → implémente le panneau de contrôle à gauche de l'interface
- une classe MyDrawingPanel dérivant de wxPanel → implémente le cadre de dessin de l'interface

L'objet de type MyFrame instancié au lancement de l'application contient deux pointeurs, l'un sur un objet de type MyControlPanel et l'autre sur un objet de type MyDrawingPanel (ces deux objets contiennent en retour un pointeur sur l'instance MyFrame). La fenêtre contient également un pointeur vers l'objet de type Controler représentant le contrôleur de l'application.

Pour la partie modèle, une classe Controler a été écrite.

Elle contient l'état de l'interface à travers 3 attributs de type int : type de bouton sélectionné, forme sélectionnée, état de la souris. (Un fichier .h de constantes contient une énumération permettant de définir cet état).

La classe Controler est reliée à la partie vue par un attribut pointeur sur un objet de type MyFrame et reliée à la partie modèle par un attribut de type Dessin.

La partie modèle de notre application implémente les classes suivantes :

- une classe Forme parente de toutes les formes géométriques et embarquant un ensemble de méthodes communes à celles-ci. Elle possède un attribut couleur de type Couleur, attribut label de type String.
- une classe Point utilisée par toutes les classes représentant une forme géométrique. Elle contient les méthodes et attributs de base attendues par une classe de ce type (deux entiers pour les coordonnées).
- des classes représentant les différents formes géométriques : Rectangle, Oval, Ligne.

Les classes Rectangle et Oval sont très proches, un ovale étant dessiné grâce à un rectangle. Ces deux classes contiennent un point haut gauche ainsi qu'une largeur et une hauteur. La classe Ligne contient deux points.

Le paradigme MVC n'a pas été complètement respecté sur un point : chaque classe géométrique possède une méthode permettant à l'objet correspondant de se dessiner par une méthode de la bibliothèque wxWidgets.

- une classe utilitaire Couleur : elle contient quatre entiers destinés à contenir les valeurs d'une couleur dans le format RGBA (le dernier entier n'a finalement pas servi mais il est implémenté).
- une classe Dessin modélisant le concept de dessin dans un cadre. Cette classe possède comme attribut un objet vector contenant des pointeurs sur des formes géométriques et destiné à contenir l'ensemble des objets des différentes classes géométriques dessinés dans le cadre. La classe possède également un attribut pour conserver la forme sélectionnée.

3. Détail des fonctionnalités (utiliser le nombre de lignes nécessaire)

Vous devez indiquer ici l'état de votre développement en terme de fonctionnalités. Une fonctionnalité peut être *conçue mais non réalisée*, *partiellement réalisée* (non complètement fonctionnelle) ou *réalisée* (totalement fonctionnelle). Vous pouvez indiquer ce qui fonctionne ou ce qui pose problème. La première ligne est un exemple.

Fonction	Conçu mais non réalisée	Réalisation Partielle	Réalisée	État de fonctionnement
Choix des figures à dessiner parmi : lignes, ovales et rectangle, et dessin à la souris de ces figures.			X	Fonctionne complètement. Les figures s'affichent dans l'ordre de création, c'est-à-dire que les cercles et rectangles dessinés en dernier cachent les figures dessinées précédemment.
Choix de couleurs			X	Fonctionne complètement.

Sélection d'une forme		X		Fonctionne partiellement : la sélection d'une ligne n'est pas implémentée
Suppression d'une forme	X			Le code complet a été écrit, il semble cohérent mais ne fonctionne pas.
Déplacement d'une forme		X		Fonctionne partiellement : le déplacement d'une ligne n'est implémenté. De plus, le déplacement d'une forme repositionne le point haut gauche de celle-ci au point du clic.
Sauvegarde au format SVG			X	Fonctionne complètement
Lecture d'un format SVG		X		Implémenté seulement pour le rectangle mais non fonctionnel
Texte d'information sur la forme en cours de création			X	Fonctionne complètement