

DRAW 7

Projet long technologie objet Rapport général de l'itération 1

Équipe 4

Mathieu Malak
Matthieu Trichard
Ethan Boswell
Nathan Albarede
Othmane Elhour
Benjamin Schlögel

Table des matières

I / Récapitulatif des programmes construits.....	2
II / Détails des sprints.....	2
III / Points à améliorer.....	3
IV /Diagramme UML.....	3

I / Récapitulatif des programmes construits

L'objectif de ce projet est de développer une variante de l'application Paint en JAVA. L'utilisateur du programme utilise majoritairement sa souris pour communiquer avec l'application. Tout d'abord, nous avons mis au point une interface graphique à l'aide de la classe "GUIHelper". Cette classe permet d'initialiser une "JFrame" et d'en afficher son contenu dans un onglet qui apparaît sur l'écran de l'utilisateur. Nous souhaitons placer des pixels de couleur sur notre interface, pour simuler un pinceau. Pour cela, nous avons créé une zone de dessin grâce à la classe "JCanvas" qui est une sous-classe de "JPanel", la JCanvas a un fond blanc et ses dimensions sont données en attribut du constructeur (cela nous permettra de laisser l'utilisateur choisir la dimension de son image). Pour le tracé à la souris, nous avons créé une interface "IDrawable" qui permet à ses implémentations de dessiner sur le JCanvas. La classe "FormDrawable" permet de créer un rectangle de couleur, qui sera considéré comme un point du trait que nous souhaitons tracer. Et "RectangleDrawable" qui est une sous-classe de FormDrawable permet de dessiner des petits rectangles de couleur aux coordonnées de la position de la souris sur le JCanvas. La classe "testCanvas" permet de tester tout ce dont nous avons parlé précédemment. Lorsque la souris est au-dessus du JCanvas et que l'utilisateur fait un clique gauche, un rectangle noir est placé sur le JCanvas à la position de la souris. De plus, l'utilisateur peut déplacer sa souris tout en continuant de dessiner. Après avoir réussi cette première étape, nous avons décidé de mettre au point des boutons pour effectuer différentes actions.

II / Détails des sprints

1. Implémentation des classes de base de l'application lui permettant d'afficher un canevas blanc sur lequel l'utilisateur peut cliquer pour dessiner.
2. Élaboration du diagramme UML et de l'architecture.
3. Implémentation de la possibilité pour l'utilisateur de dessiner en continu en cliquant puis déplaçant la souris.
4. Ajout de boutons permettant de changer la couleur du tracé.
5. Ajout d'un menu avec la possibilité de choisir la taille de la fenêtre et de créer la fenêtre de dessin.

6. Ajout de bouton pour Gommer, Colorier, Enregistrer et ouvrir un nouveau projet.
7. Ajout d'un bouton pour quitter dans le menu principal.

III / Points à améliorer

- Le tracé à la souris n'est pas optimal : lorsque l'utilisateur déplace sa souris trop vite, le trait n'est pas continu.
- L'implémentation des boutons n'est pas au point : nous devons utiliser la méthode "getBouton" pour ajouter le bouton à la barre d'outils. Il serait plus simple et évident que les différents boutons implémentés soient considérés comme des JButton pour éviter l'utilisation de cette méthode.
- L'interface graphique doit être logique pour l'utilisateur : distinguer mieux la zone de dessin des boutons et l'organisation des boutons plus claire.

IV /Diagramme UML

