

Rapport

Projet réalisé par Saker Lucas et Cerisara Théo.

1 Fonctionnalités

Le logiciel est capable de construire des figures géométriques (quadrilatère, triangle, rectangle, cercle, ellipse), de les sélectionner après création et de les déplacer, redimensionner, ou de changer leur couleur, et de tracer des traits de différentes couleurs à main levée.

2 Mode d'emploi

2.1 Création de figures

Pour créer une nouvelle figure, sélectionnez le bouton « Nouvelle figure », cliquez sur le menu déroulant prévu à cet effet, puis sélectionnez une figure dans le menu déroulant. Ensuite, cliquez dans l'espace vide en dessous des boutons le nombre de fois nécessaire à la création d'une figure (4 fois pour le quadrilatère, 3 fois pour le triangle, 2 fois pour le rectangle, le cercle et l'ellipse). Pour sélectionner une couleur, utilisez le menu déroulant correspondant. La couleur par défaut est le noir. Il est impossible de créer une nouvelle figure lorsque le bouton « Nouvelle figure » n'est pas sélectionné.

2.2 Manipulation de figures

Pour manipuler une figure déjà créée, cliquez sur le bouton « Manipulations », et sélectionnez une figure en cliquant dessus. Une fois sélectionnée, les points de manipulation définissant la figure deviennent visibles. Vous pourrez ensuite la déplacer en cliquant dans la figure (hors points de manipulation), puis en glissant avec la souris. Une fois sélectionnée, vous pouvez aussi changer sa couleur, la faire passer au dessus ou en dessous des autres, ou encore la supprimer en utilisant les boutons dédiés. Si vous cliquez sur un point de manipulation, vous pouvez déplacer uniquement ce point, ce qui peut avoir des effets différents selon le type de figure. Par exemple, un rectangle dont un sommet est déplacé devient un quadrilatère. Un cercle dont le point sur la circonférence est déplacé changera de taille.

2.3 Tracé à main levée

Pour effectuer un tracé à main levée, cliquez sur le bouton « Trace a main levee », puis dans l'espace prévu à cet effet, déplacez la souris en gardant le bouton gauche enfoncé. Pour changer la couleur du dessin, utilisez le menu déroulant prévu à cet effet. Il n'est possible de sélectionner un dessin à main levée après sa création.

3 Ergonomie

Nous avons implémenté l'interface graphique avec *Swing*. Notre interface est prévue pour un écran d'ordinateur de bureau ou portable, mais pas pour un smartphone. Nous avons utilisé des radios-boutons pour le choix des fonctions exclusives : création de figure vs. tracé à main levée vs. manipulation de figure. Un menu déroulant pour le choix du type de figure nous permet d'ajouter facilement par la suite de nouvelles figures sans modifier l'interface.

Les boutons de suppression et de changement de niveau des figures sont directement accessibles à droite. Au final, il y a relativement peu de boutons dans l'interface graphique, ce qui rend l'utilisation du logiciel plutôt intuitive. En particulier, nous n'avons pas utilisé de barre de menus. Le *BorderLayout* alloue toute la place à la zone de dessin, et positionne les boutons de contrôle en haut.

Nous avons fait tester notre logiciel à un utilisateur extérieur, qui a créé et manipulé des figures sans problème et du premier coup, sans avoir besoin d'explications. Nous estimons donc que l'ergonomie de notre logiciel est suffisamment intuitive.

4 Architecture MVC

4.1 Architecture implémentée

Nous avons utilisé l'architecture Model-Vue-Contrôleur, qui découple au maximum ces trois composants :

- Le modèle contient la définition des figures, leurs coordonnées et leur couleur, à l'exception du tracé à main levée qui n'est pas considéré comme une figure comme les autres et qui est implémentée dans le contrôleur.
- Le contrôleur contient les commandes permettant de créer et de manipuler les figures ; le contrôleur que nous avons implémenté utilise pour cela des commandes graphiques : radio-boutons, boutons et menus déroulants.
- La vue dessine les figures dans un *Jpanel*.

Le modèle est un objet *Observable*, c'est-à-dire qu'il peut notifier tous ses *Observers* des changements qui interviennent dans le modèle. Les changements pouvant être la création d'une nouvelle figure, la suppression d'une figure, la modification d'une figure, etc.

Il n'y a qu'un seul *Observer* du modèle : la vue, et plus précisément la classe *VueDessin*. En fait, cette classe est l'unique classe composant la vue.

4.2 Limites de l'implantation de l'architecture MVC

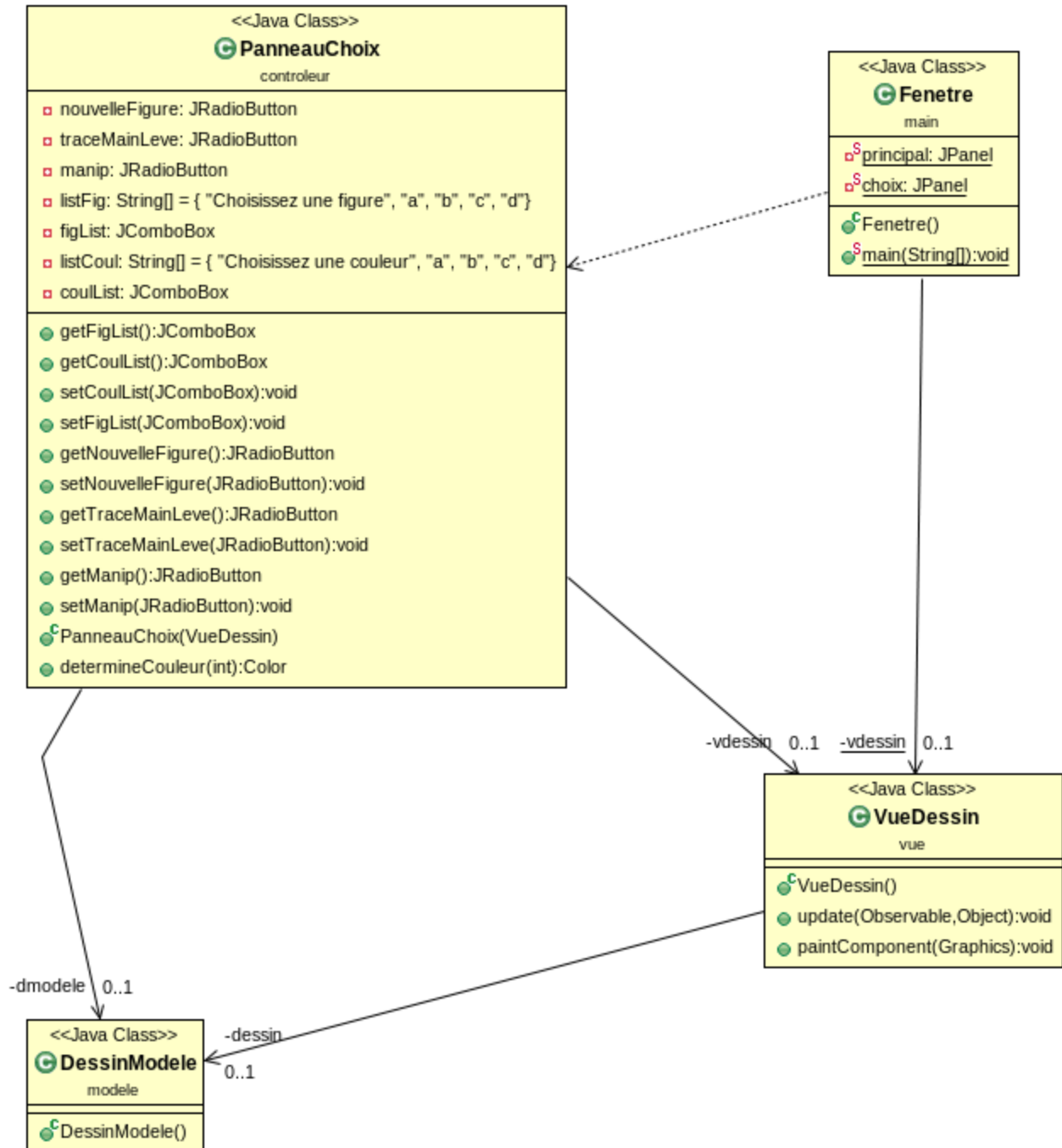
Aucune classe du modèle n'a de lien vers la vue ou le contrôleur : il y a un bon découplage entre les modèles d'une part, et la vue et le contrôleur d'autre part. De plus, la classe *DessinModele* est observable et est observée uniquement par la classe *VueDessin*. On respecte donc bien le design pattern MVC.

Par contre, il y a des liens bidirectionnels entre la vue et le contrôleur, ce qui n'est pas idéal pour du MVC. De plus, le tracé à main levée est pour le moment stocké dans le contrôleur et non pas dans le modèle.

5 Diagrammes de classes

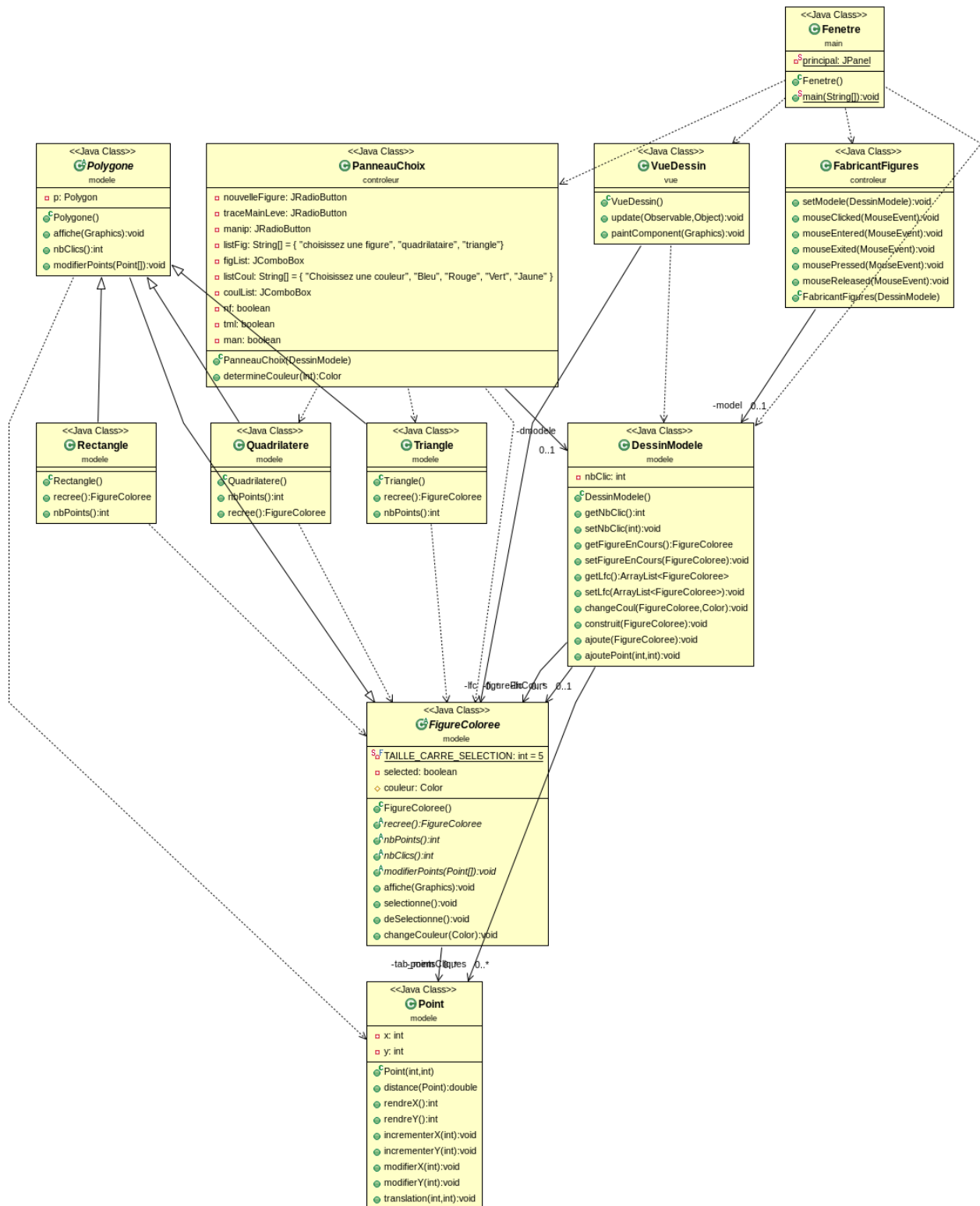
5.1 Version 1

La version 1 n'implémente aucune figure, mais mets en place la classe principale du modèle et l'interface graphique pour la vue.



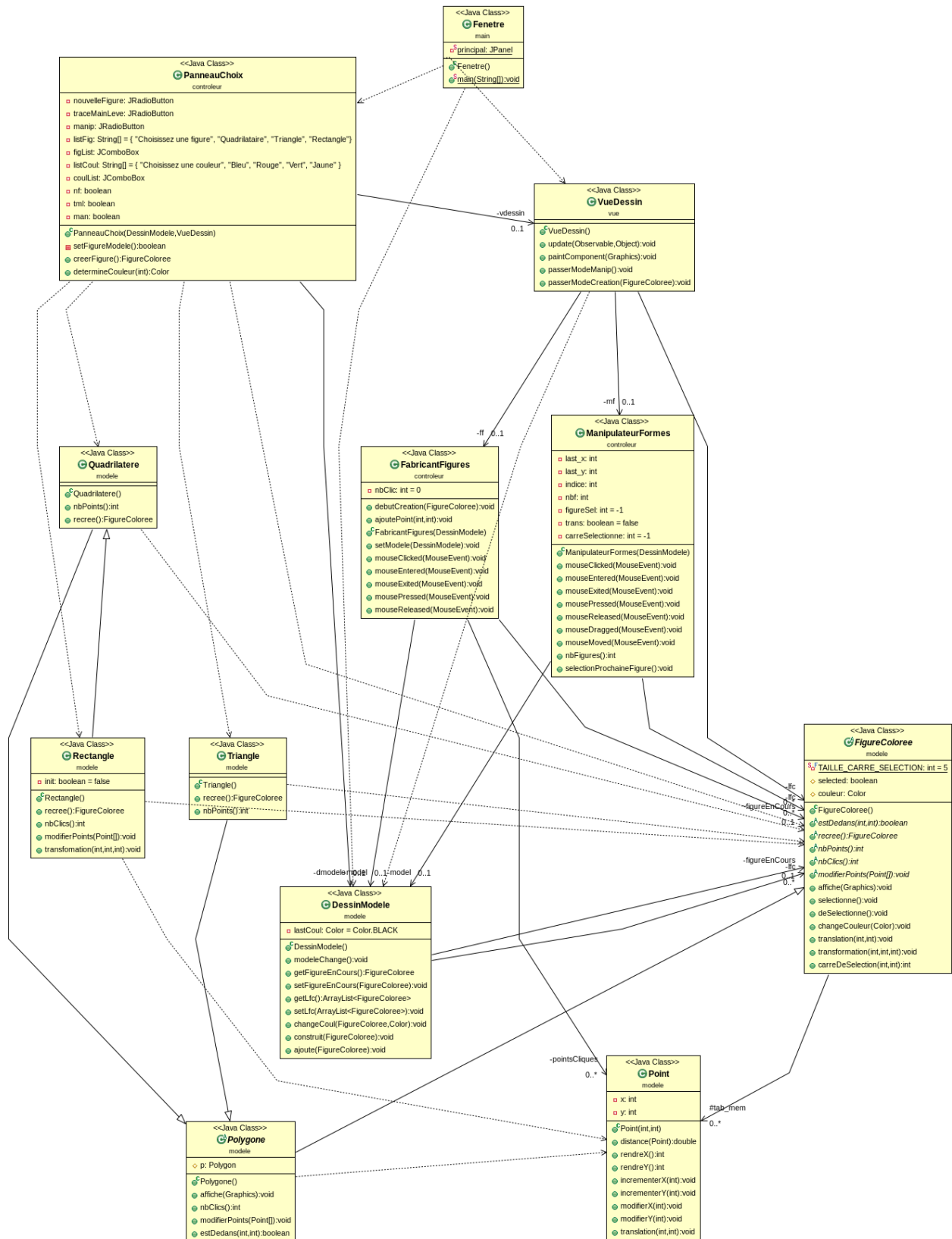
5.2 Version 2

La version 2 ajoute les figures Rectangle, Quadrilatère et Triangle, qui héritent toutes de Polygone, lui-même héritant de FigureColorée. Nous ajoutons deux classes au contrôleur : la gestion des boutons de création ou de manipulation (PanneauChoix) et FabricantFigure qui permet de construire des figures.



5.3 Version 3

La version 3 ajoute au contrôleur le manipulateur de formes et la gestion des points de manipulation des figures.



5.4 Version 4

La version 4 ajoute le tracé à main levée et quelques figures : ellipse et cercle. Nous avons également ajouté le tracé à main levée, et de nouvelles opérations de manipulation : supprimer une figure, la faire passer devant (up) les autres ou derrière (*down*).

