



Projet long de Technologie Objet

Rapport général

Élèves :

ARRIX-POUGET Baptiste
DEMAZURE Clement
DREUMONT Evann
GIULIANI Astrid
GUTIERREZ Tom
LAGRANGE Angel
SABLAYROLLES Guillaume
THEVENET Louis

27 Avril 2024

1. Itération 1

1.1. Formes

On appelle Forme les objets que l'utilisateur peut ajouter à son projet dans Pinte (rectangle, cercle, etc...)

1.1.1. User Stories réalisées

Titre	Créer les structures de données représentant les formes dans l'application
Priorité	Haute
Estimation	3
En tant que	Programmeur
Je souhaite	Avoir accès à des formes
afin de	Pouvoir développer les outils de création et édition de formes durant la prochaine itération

Titre	Permettre la conversion entre le format SVG et les formes de l'application
Priorité	Haute
Estimation	3
En tant que	Programmeur
Je souhaite	Pouvoir convertir les formes au et depuis le format SVG
afin de	Pouvoir développer la sauvegarde d'un projet utilisateur vers un fichier durant la prochaine itération

1.1.1.1. Formes

Classes ajoutées :

- Classe `CanvasColor` (voir Fig. 1)
- Classe `CanvasObject` abstraite sur laquelle les `CanvasObject` s'appuient (voir graphe d'héritage Fig. 2)
- Classe `CanvasObjectEllipse`
- Classe `CanvasObjectRectangle`
- Classe `CanvasObjectPolygon`

1.1.1.2. Conversion SVG

Classes et méthodes ajoutées :

- Classe `CanvasObjectParser` qui permet de récupérer la valeur associée à un mot-clé dans une chaîne SVG
- Méthodes `toSVG` et `createFromSVG` ajoutées aux formes

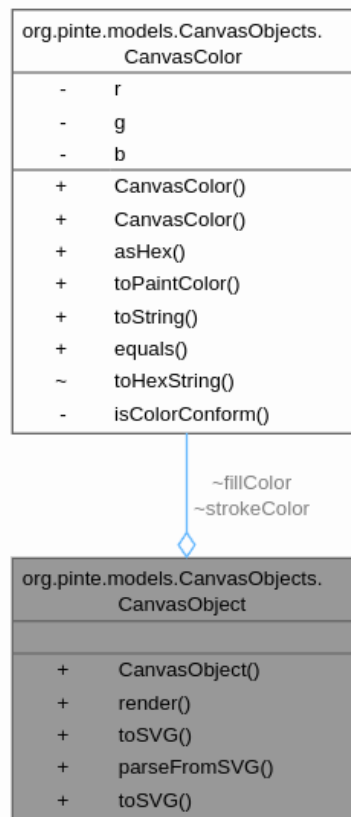


Fig. 1. – Graphe de collaboration de CanvasColor

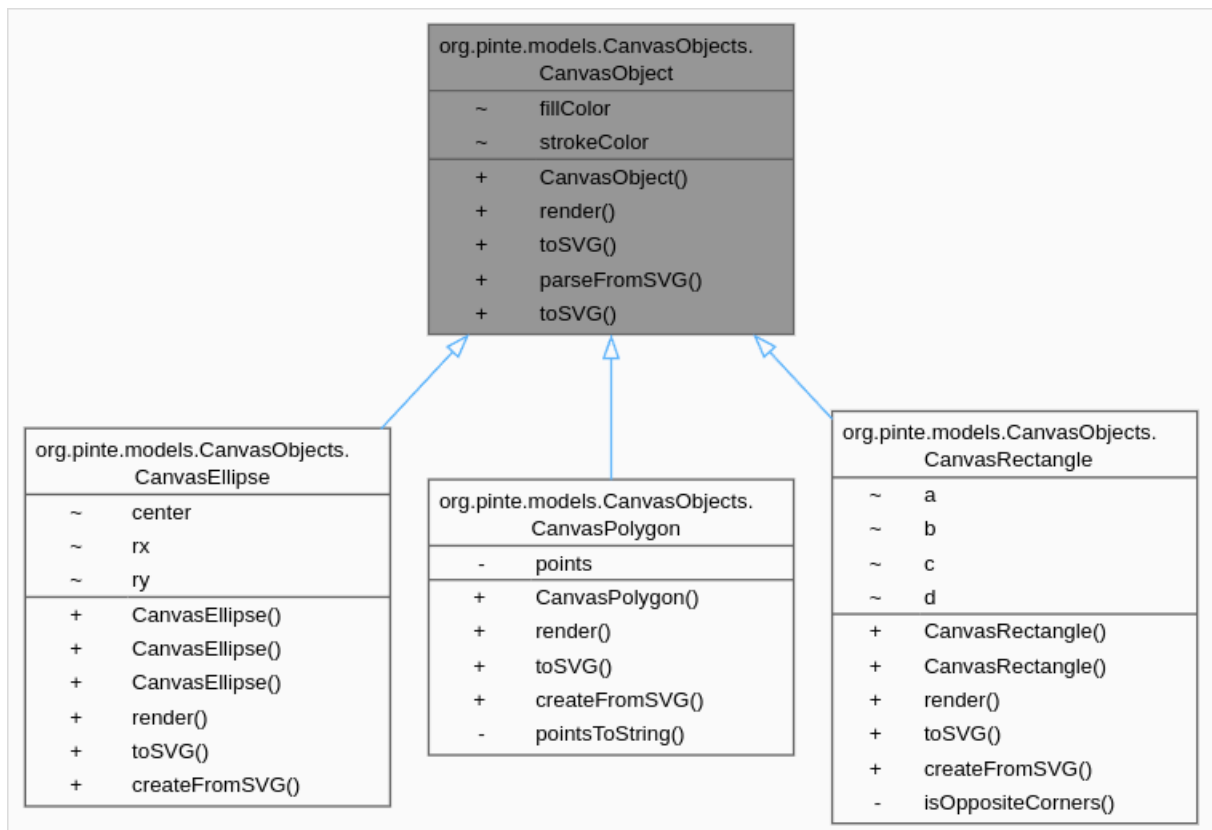


Fig. 2. – Graphe d'héritage de CanvasObject

Les méthodes `render` permettent d'obtenir un objet `Shape` affichable par `javafx`.

1.1.1.3. Tests

Classes de tests ajoutées, ces tests correspondent aux classes des formes pour tester leur comportement nominaux.

- La classe `SVGTest` a été ajouté mais ne pourra être réellement testée que lors des premiers render de l'application.

1.2. Interface