



INSTITUT NATIONAL DES SCIENCES APPLIQUÉES
DE RENNES

Génie Logiciel - Projet JavaFX Fonctionnalités additionnelles

William MOCAËR
Augustin RIALAN

Décembre 2017 - Janvier 2018
INSA de Rennes

1 Fonctionnalités additionnelles

1.1 Réinitialisation du dessin

En cliquant sur le bouton « clear », le dessin se réinitialise.

1.2 Agrandissement et réduction des formes

On peut agrandir les formes (rectangle et ellipse) et sélectionnant le bouton « taille ». Le clic gauche de la souris permet d'agrandir la forme pointée, le clic droit permet de la réduire jusqu'à un certain point.

1.3 Scroll en haut et à gauche

En déplaçant un objet sur le haut et la droite, le pane se met à jour.

1.4 Les points et les lignes

En sélectionnant le bouton « point », on peut tracer des points reliés par une ligne. On peut arrêter la génération de la ligne avec le clic droit.

1.5 Sauvegarde et chargement de dessin

1.5.1 La fonctionnalité

En sélectionnant le bouton « save », un nouveau panneau s'ouvre sur la droite. On peut spécifier un nom dans le champ texte (ou sélectionner un fichier déjà existant dans la liste), en cliquant sur « valid », le dessin s'enregistre, on peut le voir dans la liste des fichiers juste en dessous. Pour charger un dessin il suffit de cliquer sur le bouton « load » et de sélectionner le dessin à charger.

1.5.2 Difficulté technique rencontrée

Pour sauvegarder un dessin, nous avons choisi de sérialiser l'objet « Dessin » afin de le sauvegarder dans un fichier. Cependant, les éléments JavaFx « Observable » ne sont pas sérialisable (*NotSerializableException*), nous sommes donc passés par des méthodes externes retournant des listes classiques (*ArrayList*) à partir des listes observables. Pour les attributs (*DoubleProperty*, *Paint...*) nous avons également stocké les valeurs dans une *ArrayList* afin de les récupérer et les réinjecter dans les objets observable lors de la désérialisation.

On aurait pu également sauvegarder le dessin dans un format lisible, par exemple en XML, pour qu'on puisse éventuellement le modifier à la main.

2 Documentation

2.1 JavaDoc

La JavaDoc est générée et disponible dans le dossier « doc ». Dans lequel il y a également le diagramme de classe de l'application en format *Astah*¹, ainsi qu'en format *png*.

2.2 Extensibilité

Pour rajouter une nouvelle forme, il faut quatre étapes :

- Rajouter un bouton dans le *FXML*.
- Dans le code du contrôleur ; le rajouter dans la liste au début de la méthode *initialize*.
- Modifier la méthode *getModelInstance* (rajouter un case qui correspond à l'id du bouton fxml dans le switch, avec le model correspondant).
- Modifier la méthode *getShapeInstance* de cette même classe pour retourner la *Shape* javaFx correspondante et « binder » avec le model.

1. <http://astah.net/editions/community>

On aurait pu améliorer en automatisant l'ajout de forme en la généralisant (lister dans un fichier de configuration qu'on exploiterait dans le code), mais cela devient compliqué lorsqu'on doit binder le model et la vue.

2.3 Structure

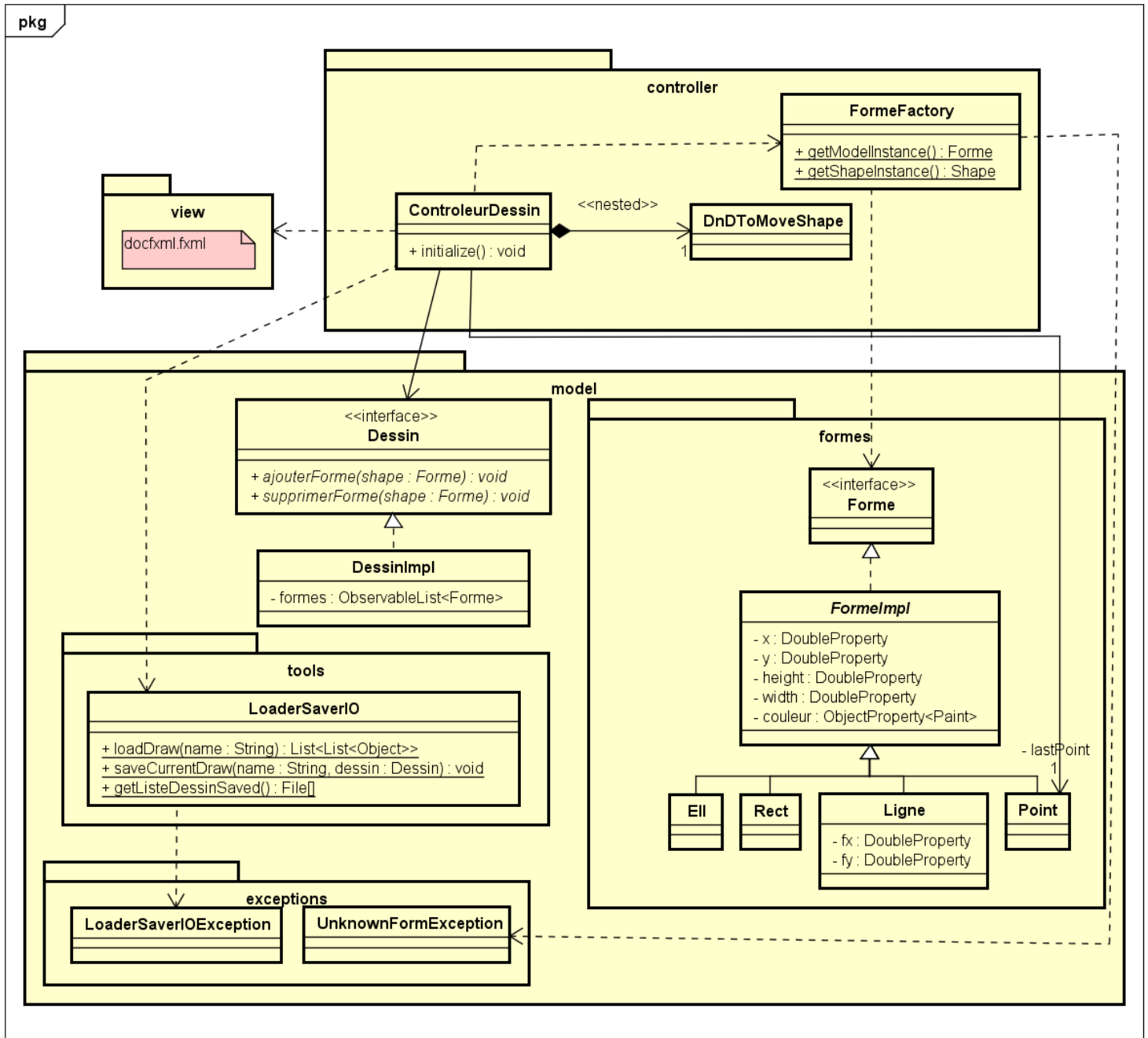


FIGURE 1 – Diagramme de classe de l'application