Rapport Projet GameLoader

CHEVALIER Simon
LACROIX Florent

Introduction :

Ce projet a pour but de proposer une interface permettant à l'utilisateur de centraliser différents jeux en Java et de les lancer directement à partir de cette interface. Nous avons opté pour une programmation en JavaFX 2. Le but principal était de pouvoir rajouter facilement des jeux à l'interface, sans toucher au code source. L'utilisateur rajoute simplement dans un dossier son jeu ainsi qu'un fichier XML le décrivant succinctement, et celui-ci est directement affiché dans l'interface de notre **GameLoader**.

Ce projet nous a également permis de découvrir la programmation avec JavaFX 2, qui était une première pour nous.

Ce que nous avons fait :

Nous avons d'abord crée les fichiers et classes permettant de trouver les jeux et les XML, ainsi que de charger les informations contenues dans les XML afin de les stocker en mémoire. Pour cela, nous avons utilisé trois classes : *Game*, *GameLister*, et *XMLReader*. *Game* permet de stocker dans une variable l'intégralité des informations contenues dans le XML, ainsi que le lien permettant de démarrer le jeu. Nous voulions au départ utiliser la classe *ClassLoader*, mais cela entraînait des complications quand nous mêlions Java et JavaFX. Nous avons donc abandonné cette idée et opté pour la méthode suivante, plus simple :

```
try {
   Process proc = Runtime.getRuntime().exec("java -jar "+game.getJARFile().getPath());
} catch (IOException e) {
}
```

Cela nous permet en plus de lancer le jeu dans une nouvelle fenêtre, ce qui nous apparaît au final plus judicieux.

GameLister permet de lister l'ensemble des jeux et de retourner une *LinkedList* de *Game*. Elle utilise la classe *XMLReader*, qui permet de lire les XML et de récupérer les informations qu'ils contiennent.

Une fois ces informations récupérées, le reste n'est que la création de l'interface permettant de dispenser ces informations à l'utilisateur (classe *GameLoader*). Le but du projet étant de faire non pas que du fonctionnel, mais aussi une production esthétiquement agréable, c'est la partie du projet

qui nous a pris le plus de temps, pour aboutir au résultat voulu. Voilà donc un aperçu de l'application (les jeux listés étant pour la plupart des jeux fictifs, servant plus les tests) :

A gauche, nous avons la liste des jeux disponibles. En cliquant sur l'un d'eux, nous arrivons à une page le décrivant rapidement, et permettant de le lancer à l'aide du bouton présent. Les informations sont celles récupérées dans le XML. Si le titre du jeu est trop long, il est coupé et réécrit dans le résumé du jeu, en plus petit. Si le texte de description est trop long, on fait appel à un *ScrollPane*, permettant de faire défiler la description.

Des effets ont été implantés : un fondu à chaque fois que l'on change de jeu, le fond de l'application est animé (effet présenté dans la documentation officielle, et adapté dans le jeu), … Aussi, l'intégralité de l'application est stylisée à l'aide d'un fichier CSS (couleur, boutons, barres de défilement, certains effets également, …). Les polices utilisées dans l'application, si elles ne sont pas standards (comme celle utilisée ici : Freestyle Script) sont chargées par l'application, afin d'être sûr que l'utilisateur puisse en profiter.

Pour rajouter des jeux, il suffit de créer un dossier dans le dossier 'game' présent à côté de l'application, contenant l'application ainsi qu'un fichier XML de la forme suivante :

```
<?xml version="1.0"?>
<game>
	<informations>
		<name>Snake</name>
		<description>Le classique jeu du Snake !</description>
		<author>Florent</author>
		<version>1.0</version>
		<date>20/11/2013</date>
		<age>8+</age>
		<category>Divers</category>
	</informations>
</game>
```

En redémarrant l'application, il sera automatiquement pris en compte et affiché dans le menu de gauche.

Conclusion et améliorations possibles :

Nous aurions pu améliorer quelques points : en effet, quand le XML contient une erreur, l'application plante avant même de démarrer. Ce point aurait pu être corrigé. Ensuite, la structure du programme en elle-même n'est pas parfaite : cela vient du fait que nous débutions en JavaFX, et que nous avons construit l'application en même temps que nous testions les capacités de JavaFX. Maintenant que nous avons un peu plus de recul, nous ferions cela à l'aide d'un fichier FXML, permettant de coder correctement et facilement un modèle MVC, et surtout de faciliter la création de l'interface à l'aide de JavaFX Scene Builder. Enfin, nous aurions pu intégrer directement les librairies permettant de parser les XML dans l'application, ce qui nous éviterait d'avoir un fichier contenant les librairies en permanence à côté de notre application. Cela a en revanche été fait pour les différentes autres ressources (images, CSS, polices).

Néanmoins, ce projet remplit le cahier des charges demandé : il propose une interface permettant de centraliser des jeux et de les lancer, ainsi que de pouvoir facilement rajouter de nouveaux jeux à l'interface. Il est aussi, esthétiquement parlant, agréable et simple d'utilisation, ce qui était également un des points du cahier des charges.

Ce rapport succinct ne présente pas tout ce qui a été fait dans l'application, mais les aspects principaux, à savoir comment nous avons procédé, ainsi que quelques points de l'interface graphique qui nous ont paru intéressants.