

# Projet de programmation orientée objet et d'interfaces graphiques

Julien COOLEN

Yaniv BENICHOU

Année 2018 - 3<sup>e</sup> semestre L2

## 1 Introduction

Nous abordons dans ce rapport nos choix de modélisation et les difficultés rencontrées au cours de ce projet pour le cours de programmation orientée objet et d'interfaces graphiques.

Le but de ce projet consiste à implémenter plusieurs jeux de plateau en regroupant les fonctionnalités communes. Pour cela nous avons développé une plateforme logicielle<sup>1</sup> à partir de laquelle nous avons codé chaque jeu : un ensemble d'objets qui interagissent entre eux et permettent de développer des jeux de plateau dans la plus grande généralité. Ce framework devait être facilement étendu pour implémenter un nouveau jeu.

Nous avons codé tous les jeux demandés : dominos que l'on assemble pour former une chaîne, dominos gommettes que l'on assemble pour former une chaîne avec des branches, un puzzle ainsi que le jeu du saboteur.

## 2 Modélisation

Nous avons tout d'abord remarqué que chaque jeu s'articule toujours autour des mêmes objets : un plateau, des pièces, des cartes, une pioche, des joueurs et leur main. Ce sont les composants de base de notre framework. Chacun de ces objets est éventuellement étendu pour ajouter ou redéfinir des comportements.

Chaque jeu adopte la structure imposée par le framework.

---

1. En anglais *framework*.

## 2.1 Présentation du framework

## 2.2 Dominos simples et avec gommettes

Nous avons commencé par implémenter le jeu des dominos simples à l'aide d'une `ArrayList`, les pièces étaient ajoutées aux extrémités de la liste. Une méthode vérifiait si une pièce pouvait se poser à gauche ou à droite d'une autre pièce. Lorsque nous avons implémenté le placement des pièces pour le jeu du saboteur, nous avons fait évoluer notre modèle en introduisant les types `Tile<S extends Side>` pour les pièces que l'on pose sur un plateau et `Side` pour les côtés des pièces, ce dernier type servant à regrouper les variables qui définissent un côté. Plusieurs évolutions successives de notre framework ont finalement cassé le code qui devait être repris.

Nous avons en revanche constaté que le jeu des dominos simples peut être vu comme un sous-ensemble des dominos gommettes, dans le sens où les dominos simples s'assemblent pour former une seule chaîne contrairement aux dominos gommettes avec lesquels on peut en plus former des branches.

C'est pourquoi nous avons implémenté le jeu des dominos simples et dominos gommettes en étendant une classe plus générale qui permet de placer côte à côte des pièces rectangulaires sur un plateau si leurs côtés se correspondent.

## 2.3 Saboteur et puzzle

Après avoir codé le jeu des dominos simples nous avons réalisé qu'il serait plus intéressant d'implémenter le jeu du saboteur car une fois écrit nous pourrions réutiliser beaucoup de code. C'est au cours du développement de ce jeu que nous avons développé une base de code commune réutilisable et facilement extensible pour chaque jeu.

## 3 Difficultés rencontrées

Lors du développement nous avons réécrit à plusieurs reprises certaines méthodes, ce qui introduisait quelques fois de nouveaux bugs. C'est pourquoi nous avons écrit des tests unitaires à l'aide de la librairie `jUnit` que l'on lance à chaque modification pour identifier toute régression dans le code.

Nous nous sommes également servi du gestionnaire de version `git` pour sauvegarder des états du projet au fur et à mesure et collaborer à deux.

## 4 Pistes d'amélioration

## 5 Conclusion

Nous avons développé le projet itérativement en travaillant à plusieurs niveaux d'abstraction : sur chaque jeu et le framework. Le développement du framework était guidé par les besoins de chaque jeu et réciproquement le développement d'un jeu dépendait directement de l'organisation du framework. Ainsi, si une extension du code était compliquée à écrire, nous revoyions le modèle. Cet aspect du projet a été de loin le plus intéressant et enrichissant.

Il était de plus très plaisant de remarquer que beaucoup de concepts se retrouvaient dans les jeux à implémenter, ce qui nous a conduit à introduire de nouveaux types qui ont grandement facilité le développement des jeux.