

Projet de programmation orientée objet et d'interfaces graphiques

Julien COOLEN

Yaniv BENICHO

Année 2018 - 3^e 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 (en anglais *framework*) 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.

3 Difficultés rencontrées

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.

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