

C'est du brutal - Compte rendu de l'application

Dans le cadre de l'enseignement de la programmation orientée-objet, il nous a été demandé de développer une application Java permettant de mettre en pratique les notions acquises en TD et en cours. Le cahier des charges de l'application que l'on a développé définit un jeu de stratégie tour par tour, dans lequel deux joueurs peuvent s'affronter. Dans ce compte rendu, nous allons présenter l'état actuel de l'application en présentant ses fonctionnalités ainsi que les choix que nous avons fait pour son développement.

1. État actuel de l'application

L'application que nous livrons aujourd'hui intègre toutes les fonctionnalités « core » spécifiées dans le cahier des charges. Nous entendons par « core » les fonctionnalités qui permettent le bon déroulement d'une partie. Néanmoins, certaines fonctionnalités relatives à l'interface graphique du jeu comme « Cette étape (Paramétrage des troupes) est bien entendue cachée pour l'adversaire. », ont été omises de la version que nous proposons. Effectivement, nous estimons que beaucoup d'améliorations peuvent être apportés quant à l'interface graphique de notre application ainsi que sur le choix des contrôleurs. En effet, pour jouer, le joueur doit saisir des commandes dans une boîte de dialogue. Cet aspect peut être amélioré pour rendre l'utilisation du jeu plus intuitive.

2. Choix de conception

Pour le développement du jeu, nous avons opté pour des choix de conception qui garantissent la flexibilité du projet et facilitent l'implémentation de nouvelles fonctionnalités. En effet, nous avons favorisé l'utilisation d'interfaces qui offrent une meilleure couche d'abstraction. Pour le changement d'état du jeu, nous avons implémenté le patron de conception « State » qui permet de définir les règles de changement d'état du jeu indépendamment de la classe Game. Pour l'accès à la partie en cours, nous avons choisi d'implémenter le patron de conception « Singleton » qui offre un accès unique et sécurisé à la partie en cours. Le développement du jeu a permis de nous familiariser au Test Driven Development (TDD) et à la librairie JUnit.

3. Évolution du diagramme de classes

Les diagrammes de classes de l'application présents dans le dossier « uml » permettent de se rendre compte de l'évolution de notre vision au cours du développement du projet. En effet, le dernier diagramme est plus complet car il inclut l'implémentation du patron de conception MVC ainsi que d'autres classes essentielles au bon fonctionnement du jeu. Ensuite, nous avons supprimé des méthodes que nous jugions inutiles et déplacer certaines dans d'autres classes afin de déléguer certaines opérations pour plus de flexibilité.

4. Structure du projet

Le projet est structuré en plusieurs sous-répertoires. Voici une liste descriptive des répertoires et fichiers qui composent le projet :

- **/build/** : contient un exécutable du projet
- **/javadoc/** : contient la documentation sous forme de javadoc du projet
- **/src/** : contient les classes et interfaces du projet
- **/tests/** : contient une liste de commandes permettant de jouer une partie
- **/uml/** : contient les diagrammes de classes du projet
- **README.txt** : contient le manuel du jeu