



Rapport Individuel Projet Long : Monopoly

Ababacar CAMARA

Avril 2020

Table des matières

1	Tâches effectuées	1
1.1	Mise en place du diagramme UML de base	1
1.2	Classes implantées	2
1.2.1	Nouveautés Itération 2 : Classe partie, Classe VueMonopoly	2
1.2.2	Nouveautés Itération 3 : Classe VueMonopoly, Classe Vue*, Modèle	3
2	Difficultés relatives à l'implantation de l'application	3
2.1	Nouveauté Itération 2 : Comment créer d'autres fenêtres sans causer un problème de thread ?	3
2.2	Nouveauté Itération 3 : Résolution du problème de la création de fenêtre	3

Bilan personnel

Dans ce rapport, je présente l'ensemble des tâches personnelles réalisées pour la création du Jeu de monopoly. Dans ce cadre, j'ai commencé par établir une vue générale de l'application en réalisant un diagramme UML de base. Ensuite, l'ensemble de l'équipe s'est réuni pour l'améliorer. Enfin, nous nous sommes répartis les tâches pour avoir une plus grande efficacité.

1 Tâches effectuées

1.1 Mise en place du diagramme UML de base

J'ai réalisé le diagramme UML de base, en essayant de respecter autant que possible les concepts vue en cours. Dans ce cadre, j'ai créé des classes abstraites qui permettrait de factoriser les éléments de la classe Carte et la classe Case. Pour l'interface jeu, nous allons utiliser le patron de conception Proxy. Et nous allons dissocier la partie Modèle de l'interface graphique. Ce qui m'a amené avoir le diagramme suivant :

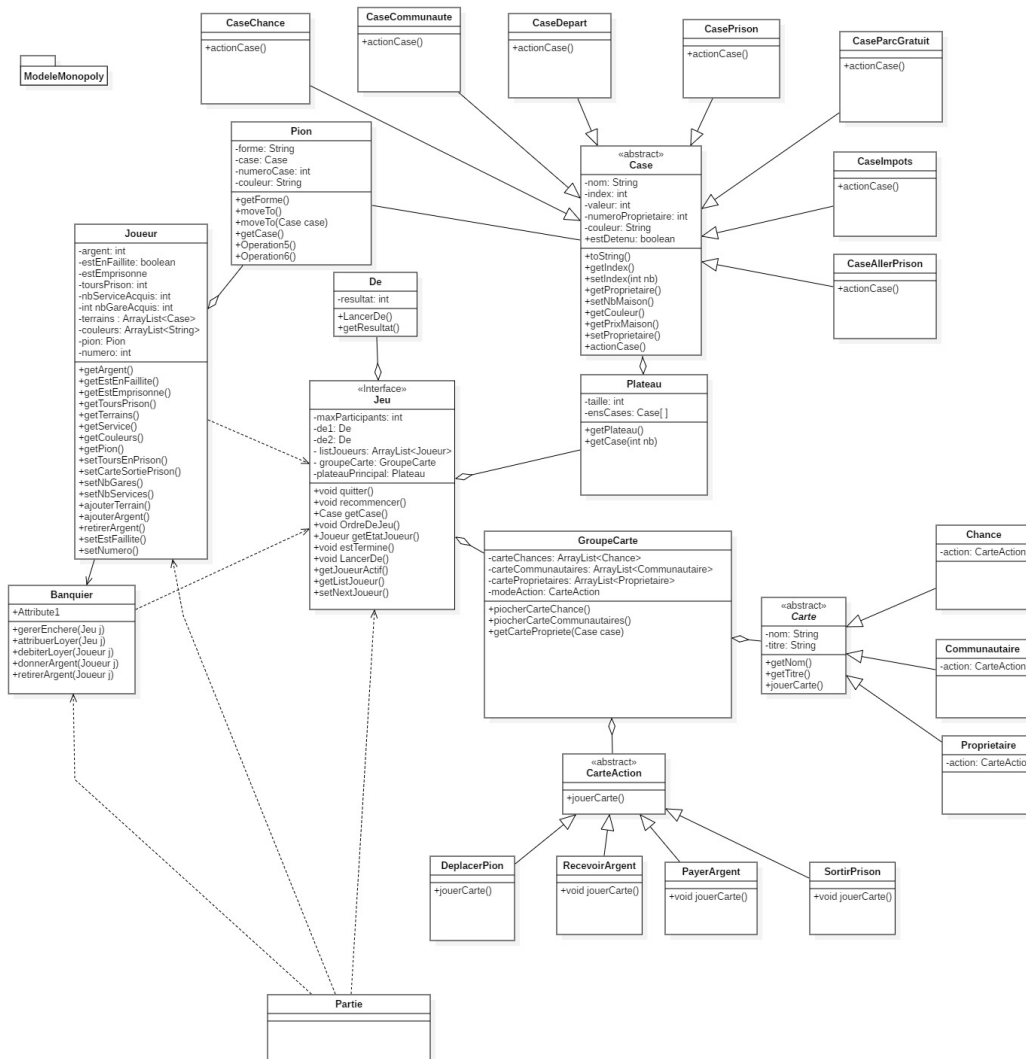


FIGURE 1: Diagramme UML

1.2 Classes implantées

J'ai fait l'implantation de la classe JeuReel qui implémente l'interface Jeu, l'implantation de la classe Dé, et je suis entrain de réaliser la classe Partie.

1.2.1 Nouveautés Itération 2 : Classe partie, Classe VueMonopoly

J'ai terminé l'implantation de la classe partie, qui en fin de compte ne servira qu'à lancer l'application. Ensuite j'ai réalisé l'implantation de la classe VueMonopoly dans laquelle se trouve aussi les contrôleurs (à travers les boutons de contrôle et leurs listeners associés).

(note : l'implantation de la classe VueMonopoly n'est pas encore terminée)

1.2.2 Nouveautés Itération 3 : Classe VueMonopoly, Classe Vue*, Modèle

Au niveau de cette itération, j'ai terminé l'implantation du Listener LancerDe et celui de TourSuivant au niveau de la, j'ai aussi créer une base de données d'images pour les pions, j'ai définis les cases , et j'ai implanté la façon dont ces pions vont être mis sur la vue Monopoly. Ensuite, j'ai créé les autres Vue(VueGare, VuePropriete,VueCommunaute,etc..)avec un code de bases. Puis j'ai entièrement implanté la VueChance, la VueCommunauté, la VueEau et la VueElectricite. J'ai aussi, corrigés certaines erreurs présentes dans l'implantation du modèle qui était pas dans ma parti et aussi j'ai ajouté de nouvelles méthodes si nécessaires.

2 Difficultés relatives à l'implantation de l'application

Pour la mise en place du modèle Swing, il est pour l'instant très compliqué de dissocier la partie modèle à la partie Swing. En effet, certaine partie du code du modèle dépend très fortement du de l'interaction avec les choix de l'utilisateur. Chose que on veut éviter le plus possible. De ce fait, je suis en train de réfléchir à comment ne plus trop être contraints par cette partie.

2.1 Nouveauté Itération 2 : Comment créer d'autres fenêtres sans causer un problème de thread ?

La difficulté que j'avais pour dissocier le modèle de la vue à été résolue, notamment en rajoutant des attributs au modèle qui permettront une meilleur gestion de ce dernier. Cependant la nouvelle difficulté à laquelle je fais face est celle de créer plusieurs classe Vue, mais aussi de créer des listeners externes pour éviter que la classe VueMonopoly ne soit trop longue.

2.2 Nouveauté Itération 3 : Résolution du problème de la création de fenêtre

La difficulté que j'avais pour créer plusieurs classe Vue a été résolue notamment en faisant hérité (extends) les autres classe que je créait de JDialog, puis de leur associer des listener internes. Ainsi, je n'ai plus eu de problème de thread. Aucune difficulté particulière n'a été révélé lors de cette itération.