



# Rapport Individuel Projet Long : Monopoly

Ababacar CAMARA

Avril 2020

## Table des matières

<b>1</b>	<b>Tâches effectuées</b>	<b>1</b>
1.1	Mise en place du diagramme UML de base . . . . .	1
1.2	Classes implantées . . . . .	2
1.2.1	Nouveautés Itération 2 . . . . .	2
<b>2</b>	<b>Difficultés relatives à l'implantation de l'application</b>	<b>3</b>
2.0.1	Nouveauté Itération 2 . . . . .	3

## Bilan personnel

Dans ce rapport, je présente l'ensemble des tâches personnelles réalisées pour la création de 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 réunis pour l'améliorer. Enfin, nous nous sommes réparties les taches pour 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 :

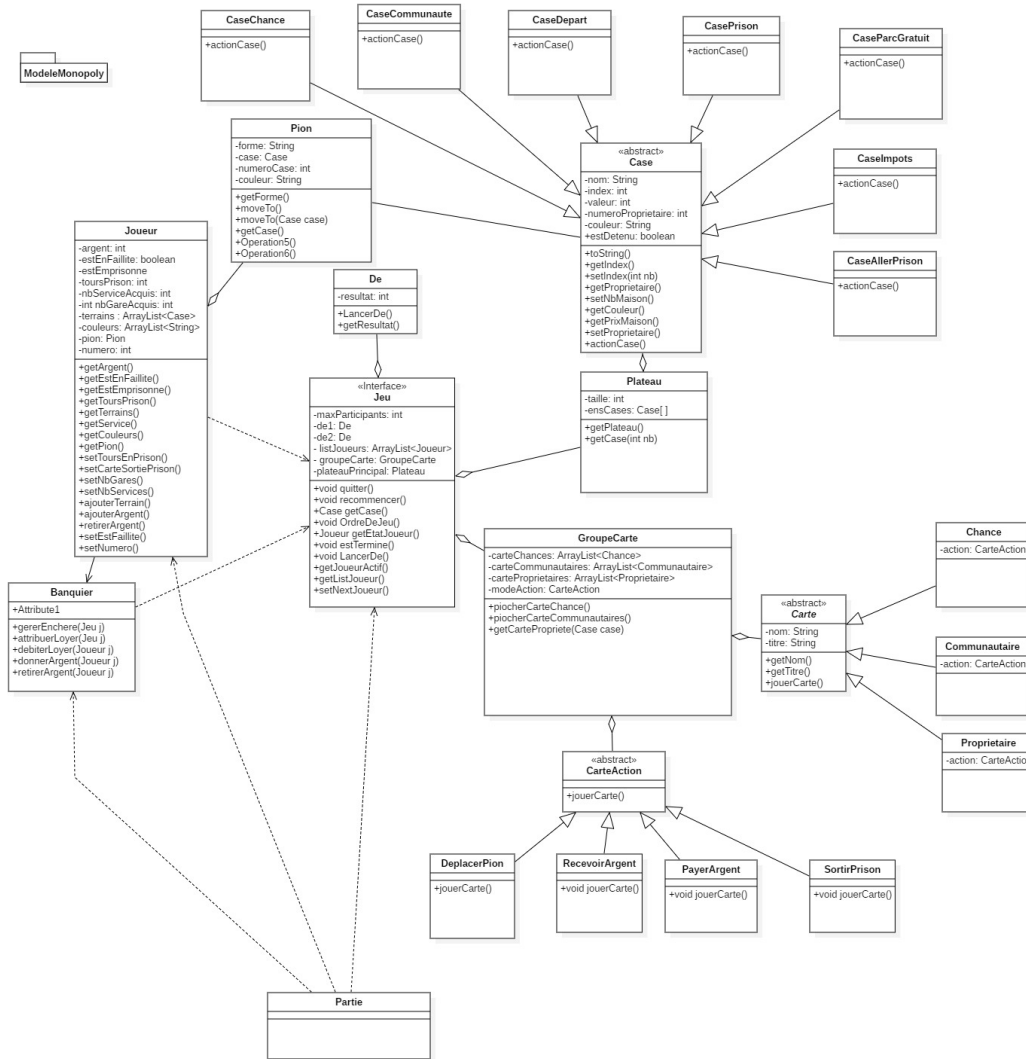


FIGURE 1: Diagramme UML

## 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

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 les listeners associés).

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

## 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.

### 2.0.1 Nouveauté Itération 2

La difficulté que j'avais pour dissocier le modèle de la vue à été résolue, notamment en rajoutant des attributs aux modèles qui permettront une meilleur gestion de cette dernière. Cependant la nouvelle difficulté à laquelle je fais fasse est de créer plusieurs classe fenêtres, et de créer des listeners externes pour éviter que la classe VueMonopoly ne soit trop longue.