

RAPPORT DU PROJET 'BLACKJACK'

Plan :

- 1. Introduction**
- 2. Principes du jeu**
- 3. Designs pattern utilisés**
- 4. Diagramme de classes**
- 5- Expériences obtenues**

1. Introduction :

Dans le cadre du module 'Méthodes de conception' on a réalisé une application en langage Java qui consiste à un jeu de carte : Blackjack. On a d'abord implémenté une librairie 'carte' qu'on a manipulée dans le projet de blackjack.

Pour mettre en pratique les connaissances acquises de ce module on a implémenté dans le projet certains patterns : Modèle-Vue-Contrôleur(MVC), le Pattern Observer et le pattern Factory.

2. Principes du jeu :

Le jeu de Blackjack consiste à faire jouer un croupier (qui représente la banque) contre plusieurs joueurs. Notre application fait jouer un humain et deux robots contre le croupier qui lui-même est un robot.

Au début, l'utilisateur appuie sur le bouton 'commencer' pour démarrer le jeu. En suite le croupier distribue 2 cartes à chaque joueur puis il prend lui aussi 2 cartes.

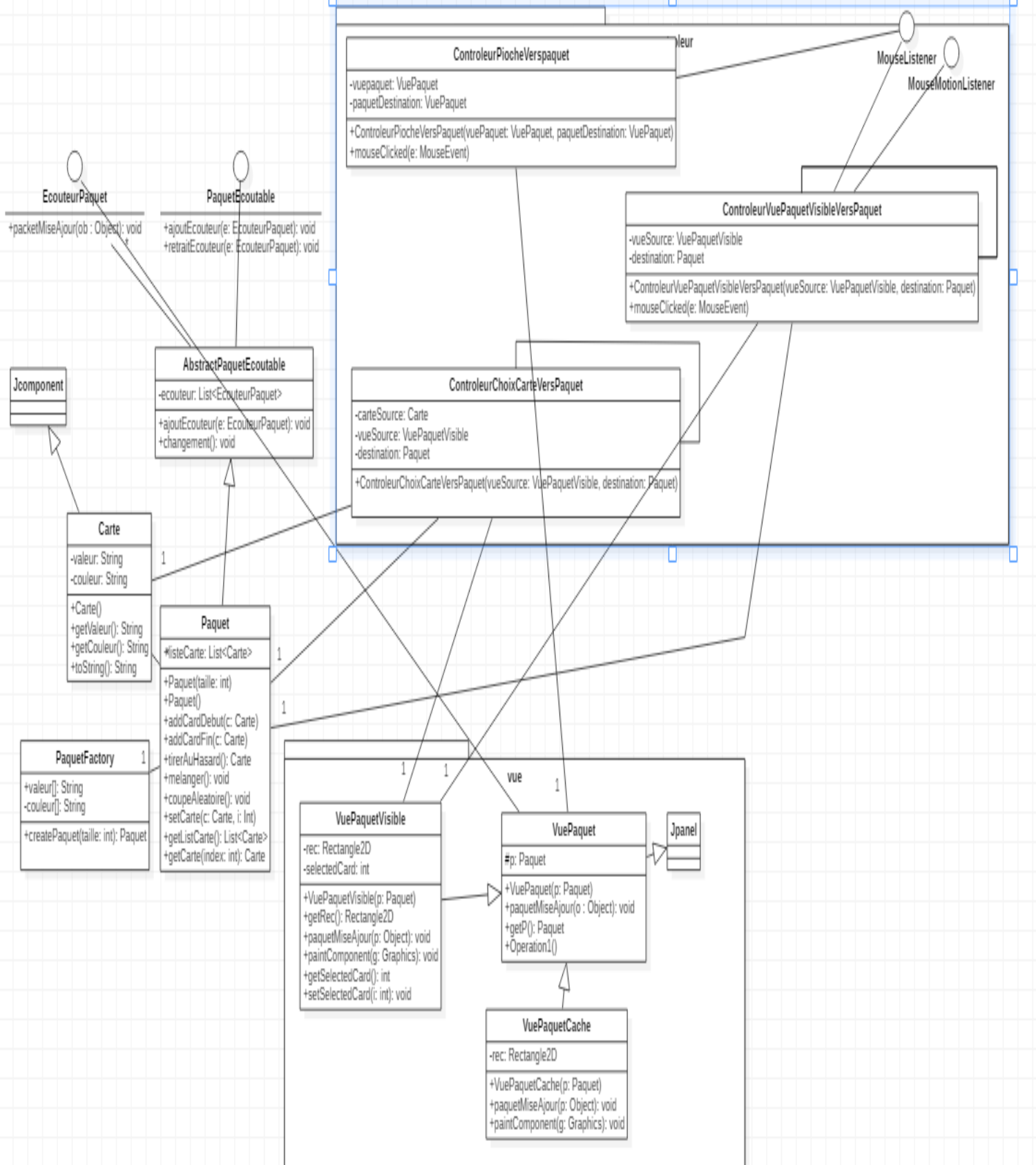
Le joueur humain est le premier à jouer il a la possibilité de demander une carte, doubler sa mise ou passer son tour. Le joueur ne peut doubler sa mise seulement après la distribution initiale des cartes. S'il passe son tour, alors c'est le tour des robots qui eux aussi peuvent décider de demander une carte ou de passer leurs tours de manière automatique.

Pour le croupier, tant qu'il n'a pas atteint 17 points il doit tirer une carte à chaque tour de jeu. S'il dépasse 17 points alors le jeu se termine, si un joueur dépasse 21 points il est éliminé d'office. En fin de jeu, on a une liste des gagnants suivants leurs points en comparaison avec le croupier qui lui aussi peut gagner ou perdre.

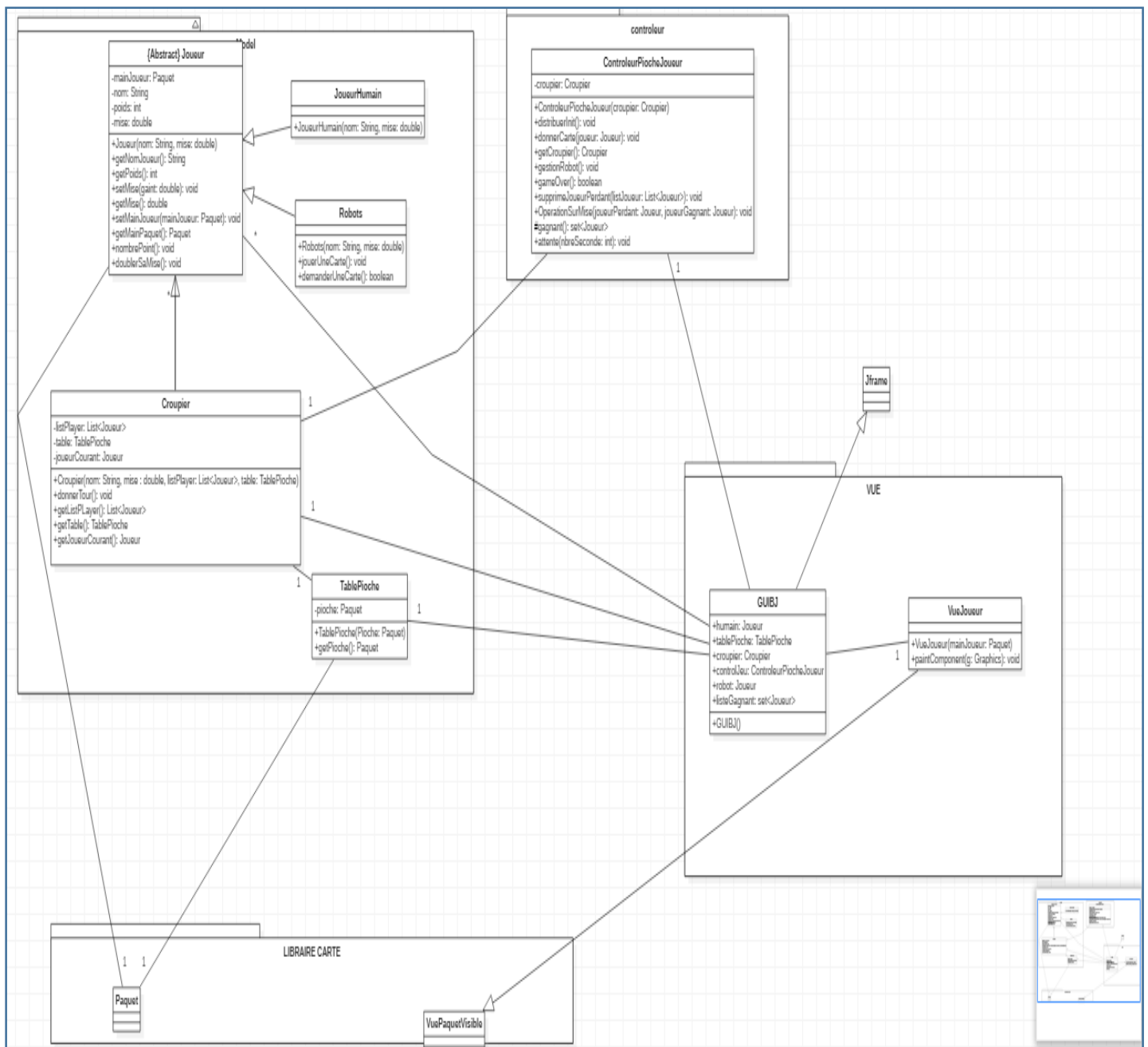
Le croupier gagne si au moins un joueur a perdu et perd si tous les joueurs ont gagné.

4. Diagrammes de classe

4.1 - Pour la librairie Carte :



4.2 - Pour BlackJack :



5- **Expériences obtenues** : Dans le cadre de ce projet, on a appris plusieurs patterns, certes qu'on n'a pas implémenté certains comme Adapter. Ces patterns nous ont permis de concevoir une application bien structurée, facile à maintenir, et à faire évoluer.

On a aussi appris à conduire un projet de la modélisation jusqu'au déploiement, notamment de l'organisation.