



RAPPORT

---

# Méthode de Conception

---

BLACKJACK

Zineb Azoui 21709992  
Yacine Kerimi 21813534  
Lucie Lepetit 21203742  
Tatiana Yang 21605680

Licence Informatique  
Troisième année  
Session : 2019

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Objectif du Projet</b>	<b>3</b>
2.1	Qu'est ce qu'un Blackjack ? . . . . .	3
2.2	Ce qu'il fallait faire . . . . .	3
<b>3</b>	<b>Architecture du projet</b>	<b>4</b>
3.1	Arborescence des packages . . . . .	4
3.2	Diagramme des classes . . . . .	5
<b>4</b>	<b>Conclusion</b>	<b>7</b>

# 1 Introduction

Dans le cadre de notre premier semestre en troisième année de licence informatique, nous devons rendre un devoir pour le module **Méthode de conception** l'UE **Génie Logiciel**.

Le Sujet du devoir était le Blackjack, traduit par **Valet noir** en français, il s'agit d'un jeu de carte très connu dans les **casinos physique** et les **casinos virtuel** le blackjack fait partie donc des jeux d'argent. Il fait son apparition à la fin du XVIIIe siècle sous le nom de "21", Il est à la base inspiré de deux jeu de cartes : **chemin de fer** et la **ferme française**. Il a par la suite été introduit aux États-Unis

.<sup>1</sup>

Nous allons au travers de ce rapport vous présenter la conception de ce projet en quatre points principaux.

Tout d'abord l'objectif du projet : que fallait-il faire.

Nous décrirons ensuite la façon dont nous avons organisé le projet.

Nous parlerons dans un troisième point de l'architecture du Projet : comment nous avons réparti les différentes classes du jeu et comment nous les avons ordonnées dans différent répertoire.

Une partie conclusion à la fin du rapport permettra de récapituler les objectifs qui ont été remplis et présentera les éventuelles améliorations possibles qui pourraient être ajoutées au jeu.

---

1. source : [https://fr.wikipedia.org/wiki/Blackjack\\_\(jeu\)](https://fr.wikipedia.org/wiki/Blackjack_(jeu))

## 2 Objectif du Projet

Nous avons eu, durant notre parcours scolaire, plusieurs cours sur la Programmation en Orienté Objet. Nous avons donc, ce semestre appris différent **Design Pattern**. L'objectif de ce projet a donc été de les utilisés et de les comprendre. Notre projet repose principalement sur le MVC, pour Modèle, Vue et Controleur.

### 2.1 Qu'est ce qu'un Blackjack ?

Après avoir reçu deux cartes, le joueur pioche des cartes pour s'approcher de la valeur 21 sans la dépasser. Le but du joueur est de battre le croupier en obtenant un total de points supérieur à celui-ci ou en voyant ce dernier dépasser 21. Chaque joueur joue contre le croupier, qui représente la banque, ou le casino, et non contre les autres joueurs.

### 2.2 Ce qu'il fallait faire

Nous avons tout d'abord, créer une librairie `cartes`, nous avons donc créer des objets `Cartes` mais également des `Paquets de Cartes`. En créant cette librairie, nous pouvons créer toute sorte de jeu de carte à partir de celui-ci. Notre jeu **Blackjack** repose donc sur cette librairie, elle lui en est dépendante.

## 3 Architecture du projet

### 3.1 Arborescence des packages

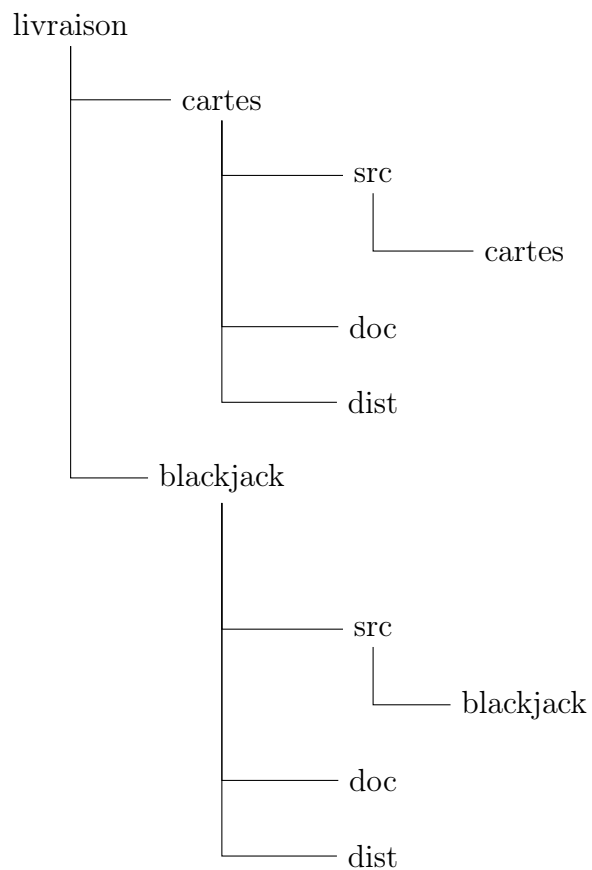


FIGURE 1 – Arborescence des packages

Sur cette figure nous pouvons voir l'arborescence de notre jeu, à la racine se trouve un répertoire *livraison* qui contient deux répertoire *carte* et *blackjack* chacun d'entre eux contient un repertoire *src* contenant le code, *doc* contenant la javadoc et *dist* contenant l'exécutable et/ou la librairie.

### 3.2 Diagramme des classes

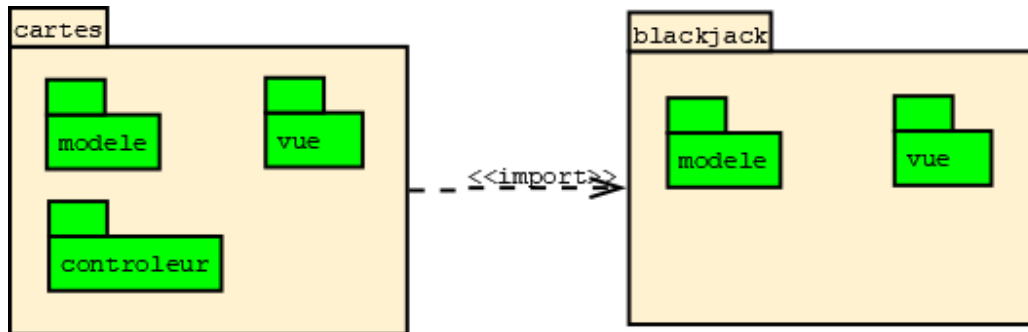


FIGURE 2 – Diagramme de package cartes et blackjack

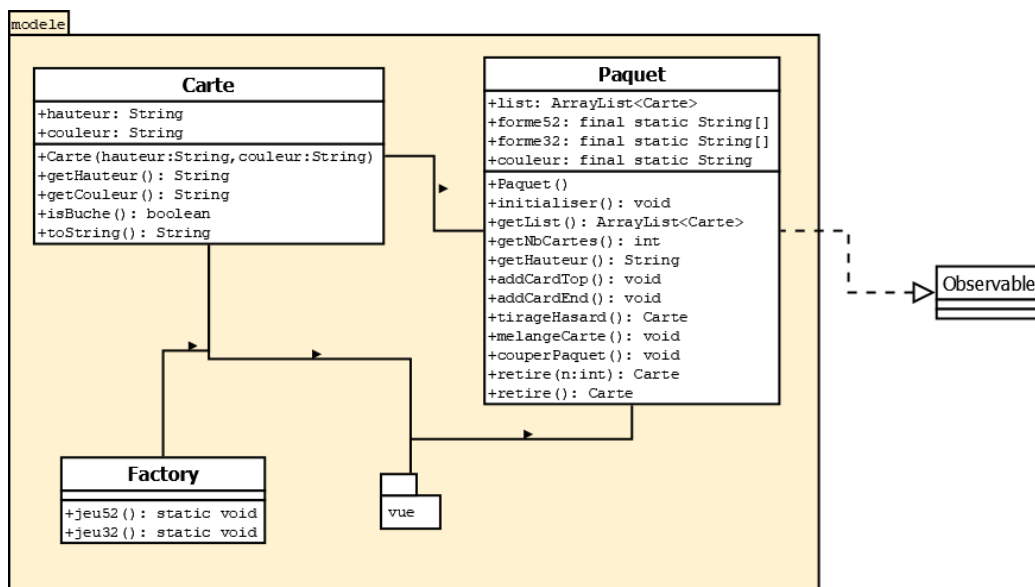


FIGURE 3 – Diagramme de package du modèle dans le package cartes

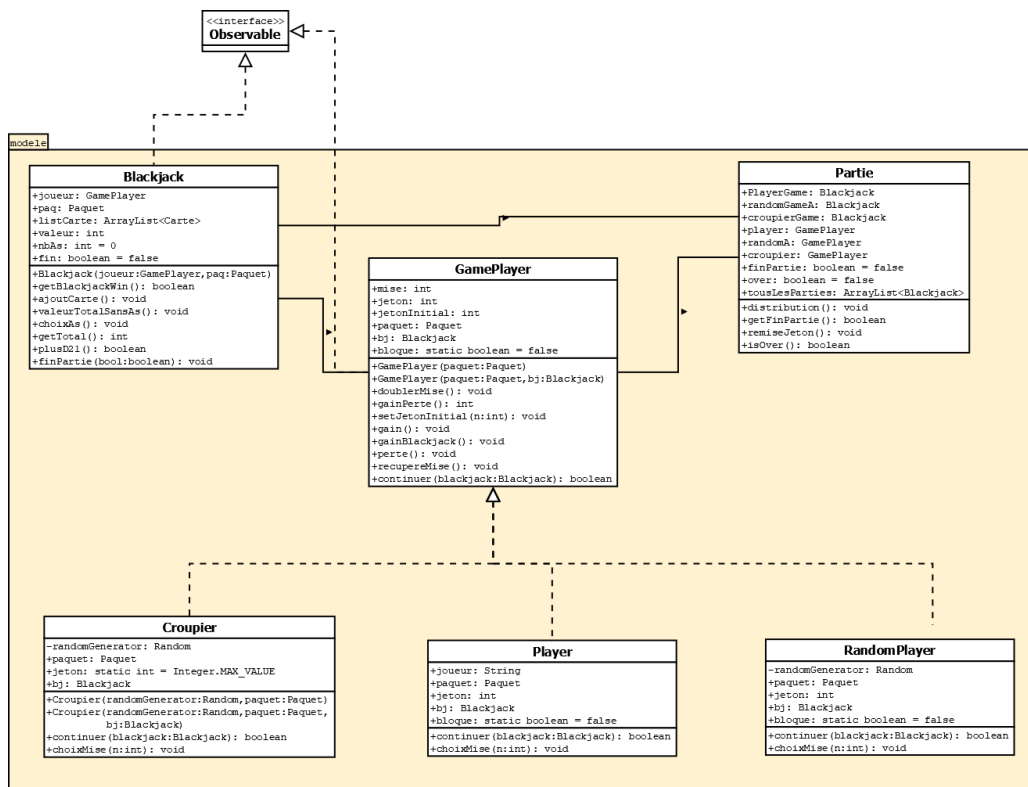


FIGURE 4 – Diagramme de classe du modele dans le package blackjack

### Le déroulement du jeu :

#### Librairie cartes :

Notre jeu fonctionne donc sous un pattern MVC (Modele, Vue, Controleur).

On trouve tout d'abord une librairie cartes, qui permet pour tout type de jeu de carte, d'afficher une pioche et une main de joueur. Ici, tout ceci est regit via l'executeur Main de cartes. Si l'utilisateur clique sur la pioche, une carte est piochée. Le controleur actualise la vue de la pioche en enlevant une carte a chaque clic, celle-ci, apparait dans la vue de la main VueCarteJoueur. Ces deux vues sont intégrées à l'interface GUI de carte , et s'actualisent donc grace au controleur. Notre jeu du blackjack utilise cette librairie de cartes qui affiche la vue de la main et la vue de la pioche. Le jeu est lancé via l'executeur Main du blackjack.

#### Modèle :

### Blackjack

Le modèle de notre projet est la classe Blackjack qui est observable, et est observé par la classe BlackjackGUI présent dans le package vue. Le Blackjack crée un tableau d'objets d'instances de Carte.

### Partie

La classe Partie possède pour chaque joueur une partie qui est une instance de la classe Blackjack parmi les joueur figure un joueur humain qui est une instance de Player, un joueur aléatoire qui est une instance de RandomPlayer et le croupier qui est un joueur aléatoire aussi. Quand la partie des joueur est lancé tout les joueur se voit être distribué deux carte chacun et la partie commence.

## 4 Conclusion

Grace à ce projet, nous avons pu apprendre à utiliser quelques **Design Pattern** parmi l'ensemble de ceux, dont nous avons appris durant notre semestre. L'utilisation de ces Design Pattern ont permis à une meilleure compréhension de notre code, et une efficacité lors de la réalisation de ce projet.