2TL2 – 2017-2018

T2012 - Développement informatique avancé orienté application (Pratique)

BlackJack

Auteurs :
BLACKS Sébastien
MEYERS Humbert
GROUPE 16

Encadrants :
DEWULF Arnaud
VAN DORMAEL Louis

Version Finale du
22 décembre 2017

Table of Contents

Introduction

Le jeu du **blackjack** est un jeu de casino impliquant un ou plusieurs joueur(s) ainsi qu'un croupier. Tous les joueurs jouent indépendamment contre le croupier.

Le jeu se déroule en deux temps, le joueur peu lorsque c'est son tour soit demander une carte soit s'arrêter de jouer. Lorsque tous les joueurs ont fini de jouer, le croupier prends des cartes jusqu'à avoir une main dépassant le score de 16, au-dessus de 16, il lui est interdit de piocher une nouvelle carte.

Le but étant d'avoir une main dont le score se rapproche le plus possible de 21 sans dépasser. Le(s) joueur(s) ayant un score plus élevé que celui du croupier gagne(nt). Les valeurs des cartes sont réparties comme suit : l'as vaut 1 ou 11, du 2 au 9 valent leur valeur nominale et de 10 au roi valent tous 10.

En cas d'égalité entre le croupier et un joueur, c'est celui qui possède moins de cartes qui gagne. S'ils possèdent le même nombre de cartes, c'est le croupier qui gagne.

Liens utiles

- Repository GitHub : https://github.com/sebiba/Blackjack
- Wiki GitHub : https://github.com/sebiba/Blackjack/wiki

1. Cahier des charges

1.1. Organisation du jeu

1.1.1. Le menu

Le joueur aura la possibilité de choisir entre 4 options au démarrage de l'application :

- Lire les règles
- Jouer en solo
- Jouer en multi sur le même ordinateur
- Jouer en multi à travers le réseau

1.1.2. Choix du mode de jeu

Dans les trois cas de jeu, le(s) joueur(s) devront se nommer. Ils démarrent toujours avec somme de 1500€.

Lorsque le mode multi en réseau est sélectionné, il doit avoir un joueur qui héberge le jeu et l'autre le rejoigne grâce à son adresse IP.

1.1.3. Les actions

Les joueurs ont le choix entre miser en début de partie, de prendre une carte ou de s'arrêter.

1.1.4. Utilisation de l'application

L'application s'utilise simplement dans le mode solo et le mode multi que ce soit en console ou en GUI.

Cependant, la partie multi en réseau doit être démarrée comme suit :

- Lancement de l'hôte jusqu'à la fenêtre d'attente de connexion pour qu'il écoute sur le port 12345 prédéfini dans le code. Si la partie est lancer à travers internet, il faut au préalable faire une redirection de port sur le routeur de l'hôte.
- Lancement du client ou il doit obligatoirement préciser l'adresse IP ou le FQDN de l'hôte.

1.2. Le jeu en CLI

Le jeu se contrôle principalement avec les touches « O » et « N ».

Les cartes s'affichent en ligne de caractère. Et sont rappelées à chaque nouvelles carte piochée.

Les informations concernant le(s) joueur(s) sont quant à eux aussi rappeler

1.3. Le jeu en GUI

Le jeu se contrôle via des boutons cliquables et des entrées au clavier.

2. Informations générales

2.1. Développement

L'application est organisée suivant l'architecture MVC pour garantir une facilité de lecture et un meilleur travail en équipe.

2.2. Langage

L'application n'est disponible qu'en français. Par contre, il peut être facilement traduit dans d'autre langues.

2.3. Client / Serveur

Le jeu en multijoueur se divise en deux instances de l'application :

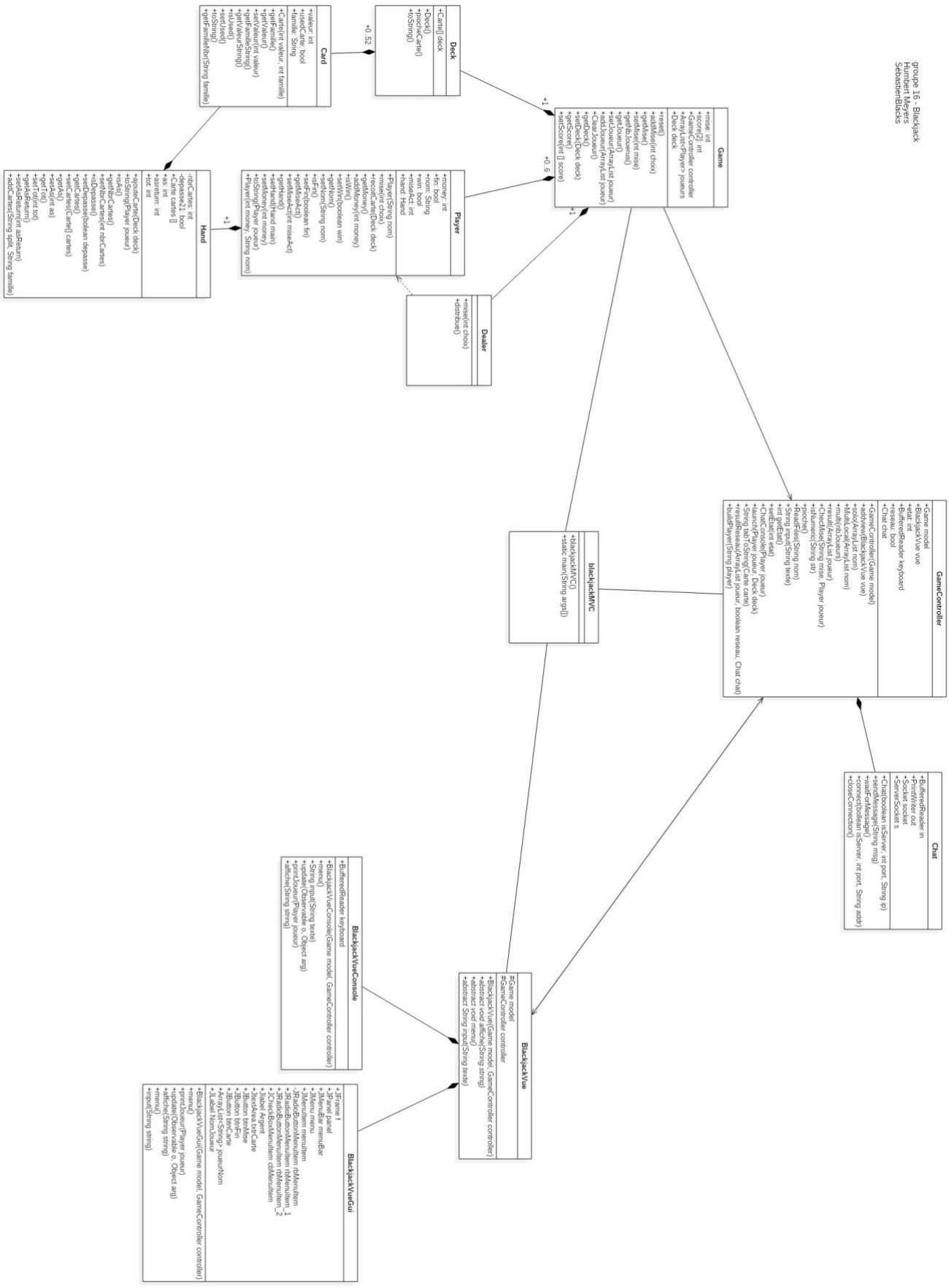
- L'hôte :

  Il reçoit les différentes informations du client lorsque celui-ci a finis de jouer la partie, pour ensuite finir la partie, calculer les résultats et les renvoyer au client.

- Le client :

  Après avoir finis la partie, l'hôte envoi son nom de joueur ainsi que son score final au serveur pour ensuite laisser l'hôte jouer à son tour et renvoyer les résultats.

3. UML


groupe 16 - Blackjack
Humbert Meyers
SebastienBlacks
Game
Deck
Card
Player
Dealer
Hand
GameController
Chat
blackjackMVC
BlackjackVue
BlackjackVueConsole
BlackjackVueGui

4. Difficultés rencontrées

4.1. Idées mise de cotées

Dû à l'échéance nous avons dû mettre plusieurs idées de côté. Tels que :

- Une GUI affichant des images de cartes au lieu d'afficher les cartes en textes.
- Un serveur indépendant permettant des parties multijoueur en réseaux a plus de deux joueurs.
- Implémentation de la classe dealer qui est restée vide.

4.2. Difficulté

Lors de ce projet nous avons eu plusieurs difficultés. Les principales difficultés ont été :

- le délai relativement cours pour terminer le projet
- MVC qui n'était pas assez maitrisé par aucun d'entre nous…

5. Conclusion

Ce projet a été une bonne expérience malgré le fait qu'on soit pas arrivé à nos attentes dû à une mauvaise gestion de temps de notre part. Nous avons que très peu modifier le projet dans l'ensemble.

Nous avons cependant des idées d'améliorations, telle que :

- Faire un serveur centralisé avec plusieurs clients.
- Pouvoir jouer plusieurs parties d'affiler sans devoir redémarrer l'application.
- Un mode de sauvegarde d'une partie.
- Idées mises de coté.

5.1. Points de vue personnels

5.1.1. Sébastien

Faire ce projet m'a montré l'importance de prévoir plus large dans les délais que prévus car en programmation il y a toujours ce petit truc qui coince quand il ne faut pas…

Durant ce projet j'ai aussi compris que je n'aurais pas dû remettre l'implémentation de MVC à plus tard comme je l'ai fais en pensant prendre de l'avance dans la logique du blackjack, cela n'a rendu les choses que plus compliqué. Et que même si on travaille peut-être plus vite proche de l'échéance la « beauté » du code en prend un méchant coup et c'est regrettable.

5.1.2. Humbert

Avoir fait ce projet m'a apporté des choses que l'on n'a qu'en groupe tel qu'un travail d'équipe ainsi qu'une amélioration dans la programmation d'une application mutuelle avec différent point de vue.

Au début du projet, j'avais une motivation hâtée par des faiblesses en programmation qui n'est pas mon domaine de prédilection. Avoir Sébastien comme binôme m'a permis de mieux comprendre le langage Java.

Nous avons pu avoir l'aide de certaines personnes telle que Benoit ou Adrien qui ont pu nous apporter d'autres point de vue sur nos problèmes.