



Compte rendu du projet Client-Serveur



AMRAOUI Ahmed

3^{ème} Année EMACS

Année scolaire 2014 /2015

Sommaire

Introduction.....	3
1. Expression du besoin :	4
2. Exigences :	6
2.1. Exigences fonctionnelles :	6
2.2. Exigences techniques :	6
3. Architecture Générale	7
3.1. Architecture du réseau	7
3.2. Architecture logicielle.....	7
4. Critères pour la qualité (structuration en couches)	10
5. Résultat obtenu :	10
6. Conclusion	13

Introduction

Ce projet rentre dans le cadre de mes études en Génie informatique à l'école des Mines d'Alès, et qui consiste en la réalisation d'un projet de développement informatique, en client-serveur. Il me permettra donc de mettre en application mes connaissances acquises en programmation, mais surtout de monter en compétences.

Ce rapport est décliné en quatre parties. La première présente une expression du besoin. La définition des exigences fonctionnelles et techniques vient en second lieu. Une troisième partie sera réservée à l'architecture générale du système ainsi que logicielle. Enfin, on détaillera la structuration en couches (critères pour la qualité).

1. Expression du besoin :

Etude de l'existant :

Le Blackjack représente un jeu de cartes dans lequel le joueur tente de battre le casino, représenté par le croupier (dealer), tout en obtenant une main d'une valeur égale ou inférieure à 21 et supérieure à la main du croupier. Bien qu'il y ait plusieurs joueurs autour d'une table de Blackjack, la relation est toujours joueur contre casino, et jamais joueur contre joueur.

Une table de blackjack est généralement composée d'un croupier (le donneur) et de un à sept joueurs. On utilise en général de un à huit jeux.

Valeurs cartes :

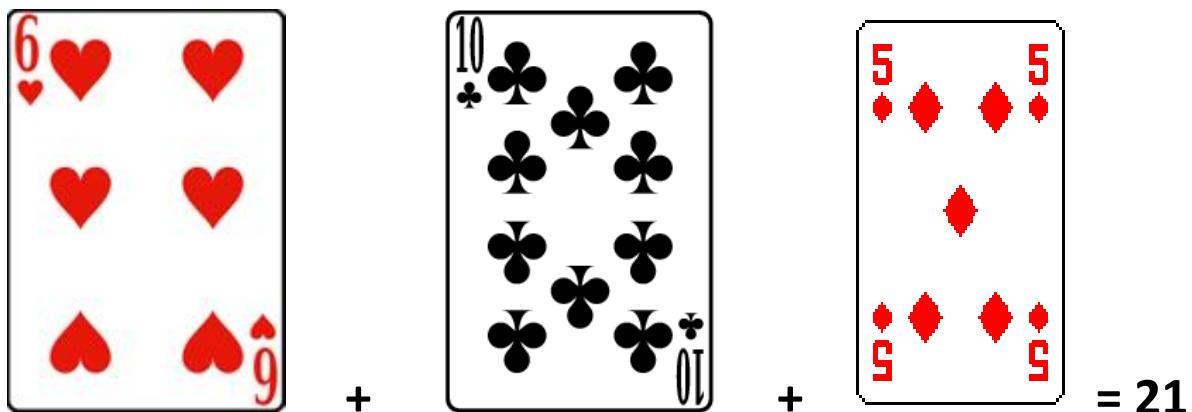
-Chaque carte numérotée de 2 à 10 a une valeur égale au numéro de la carte. Par exemple, le trois de trèfle vaut 3 points.

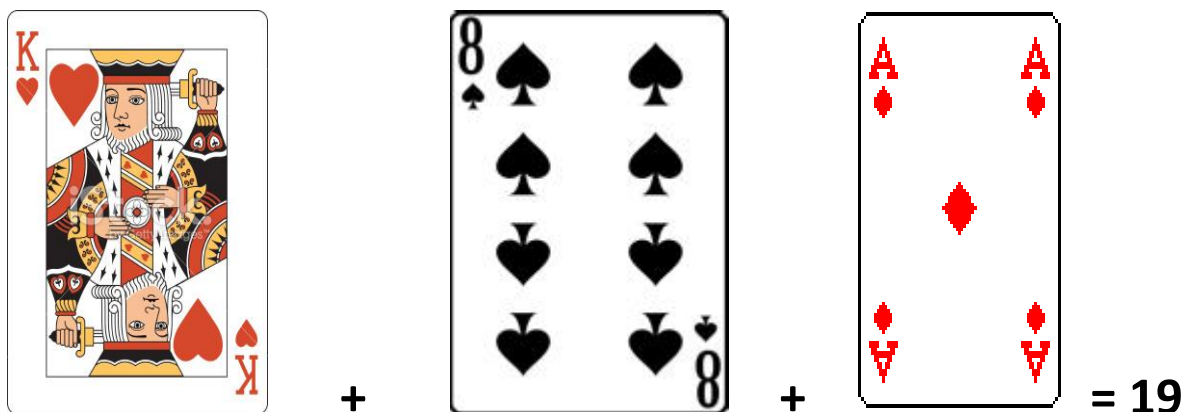
-Les figures valent 10 points (valets, dames et rois).

-L'As vaut **1** point ou **11** points, selon ce qui arrange au mieux le joueur. C'est une sorte de joker.

La meilleure combinaison possible est le black jack, à savoir la valeur 21 obtenue avec deux cartes (AS + FIGURE ou 10). Dans ce cas, le joueur emporte 1,5 fois sa mise.

Exemple :





Actions du joueur :

Chaque joueur reçoit deux cartes, face visible, distribuées une par une. Le donneur reçoit une carte face visible et l'autre face cachée.

Une fois la donne effectuée, l'un après l'autre, les joueurs peuvent, s'ils le souhaitent, demander une ou plusieurs cartes supplémentaires jusqu'à ce qu'il souhaite arrêter ou bien lorsqu'il saute (la somme de ses cartes faisant plus de 21).

Lorsque la main d'un joueur dépasse 21, on dit qu'il a sauté et il perd quel que soit le résultat de la main du donneur.

Si le joueur et le donneur possèdent chacun une main d'une même valeur sans avoir sauté, on dit qu'il y a égalité et le joueur ne perd pas, mais ne gagne rien non plus.

Néanmoins un Blackjack sera toujours vainqueur face à une main de 21 (ex. 10 piques + 4 cœur + 7 pique).

Actions du donneur :

Une fois que tous les joueurs ont joué et s'il reste des joueurs n'ayant pas sauté, le donneur découvre sa carte face cachée et se distribue des cartes, si nécessaire, jusqu'à ce que sa main ait atteint un total de 17 ou plus.

Après cela, si le donneur a sauté, les joueurs restants ou ayant une main plus forte que la sienne gagnent.

2. Exigences :

2.1. Exigences fonctionnelles :

Le programme doit répondre à un ensemble de fonctionnalités dans le but de permettre aux joueurs de contrarier le croupier et savoir en fin du jeu s'ils ont gagné ou pas.

Parmi ces fonctionnalités on trouve :

- Disposer d'une interface coté client, et du coté serveur.
- Permettre à un joueur ou plusieurs de se connecter au serveur (croupier), pour commencer le jeu.
- Recevoir les **deux** premières cartes distribuées par le croupier.
- Permettre au joueur de décider de continuer ou de sortir du jeu (en fonction du nombre total de ses cartes).
- Afficher le résultat du joueur ainsi que celui du croupier.
- Afficher le vainqueur de cette partie.

2.2. Exigences techniques :

BlackJack est une application desktop écrite en java en utilisant les bibliothèques incluses dans le package standard, et qui sera déployé sur le réseau, en utilisant les interfaces de connexion (les sockets).

Outils de développement :

Eclipse :

- Utilisé pour le développement et la réalisation de l'application BlackJack.



Eclipse

BoUml :

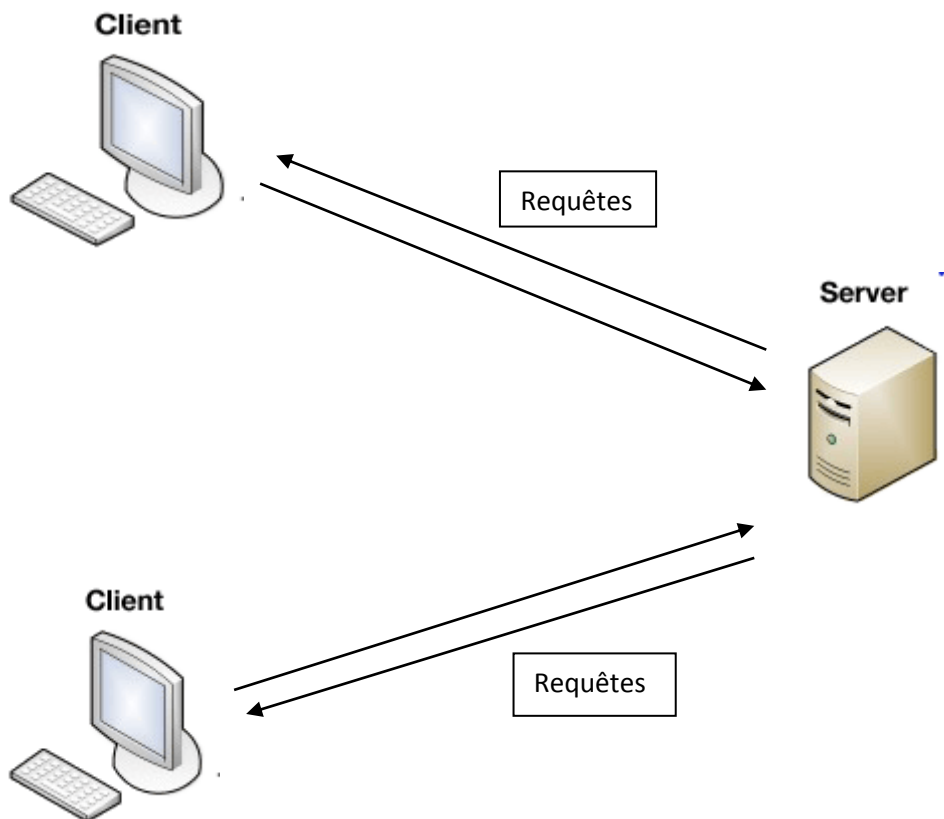
- Utilisé pour dessiner des diagrammes UML



3. Architecture Générale

3.1. Architecture du réseau

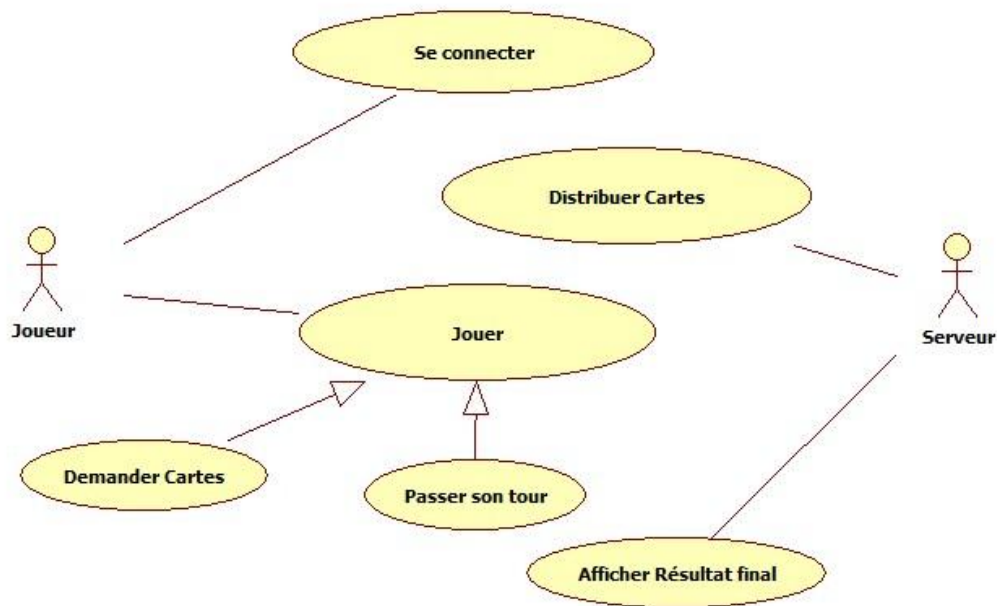
La structure du réseau est simple. Deux clients (ou plus) se connectent au serveur et échangent des requêtes. La structure du réseau est la suivante pour le cas de deux clients connectés au serveur :



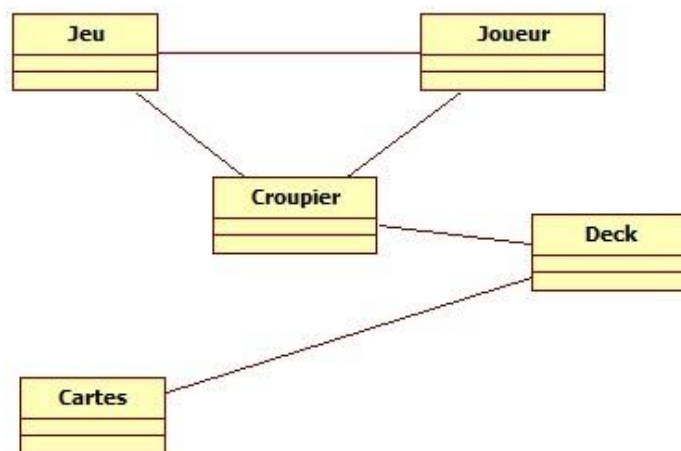
3.2. Architecture logicielle

Pour décrire l'architecture logicielle nous proposons un diagramme cas d'utilisation, un diagramme de classes et un diagramme de séquences.

Pour avoir une vision globale du comportement fonctionnel de notre programme et pour mieux visualiser les acteurs et leurs rôles nous avons eu recours à un diagramme de cas d'utilisations. Le joueur se connecte et joue. Le serveur distribue les cartes et affiche le résultat final :

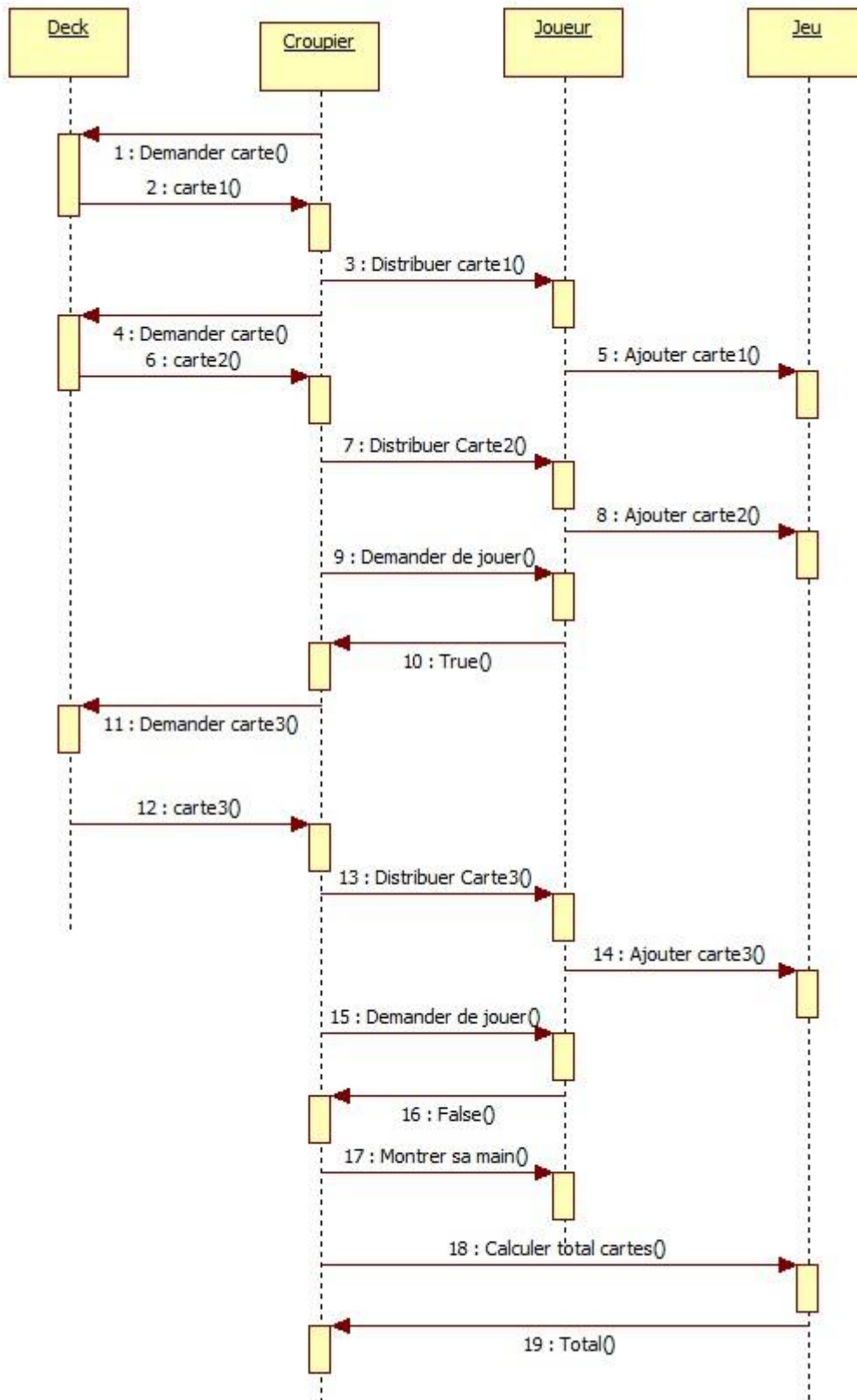


Pour présenter les différentes classes et interfaces du système ainsi que les relations entre celles-ci nous avons eu recours aux diagrammes de classes présenté ci-dessous:



Afin de mieux visualiser les différentes interactions entre les acteurs et le système selon un ordre chronologique nous procédons à la réalisation d'un diagramme de séquences décrivant le scénario ci-dessous et faisant intervenir les classes suivantes : Deck, Croupier, Joueur, Jeu.

Le Croupier demande des cartes au Deck, le Croupier distribue les cartes aux joueurs, les joueurs ajoutent les cartes à leur Jeu. Le Croupier demande au joueur de jouer. Le joueur joue. Le Croupier demande au Deck de nouvelles cartes que le joueur ajoute à son jeu. Le joueur refuse de jouer pour montrer sa main pour qu'enfin le Croupier calcule le total des cartes.

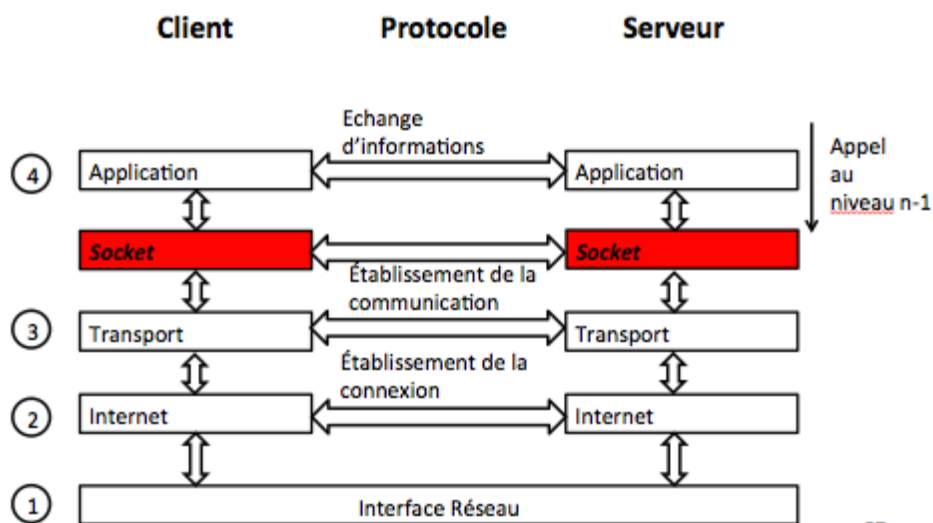


4. Critères pour la qualité (structuration en couches)

Le client et le serveur implémentent des protocoles pour générer et analyser les données échangées.

Les protocoles s'empilent par dessus les protocoles internes de TCP/IP.

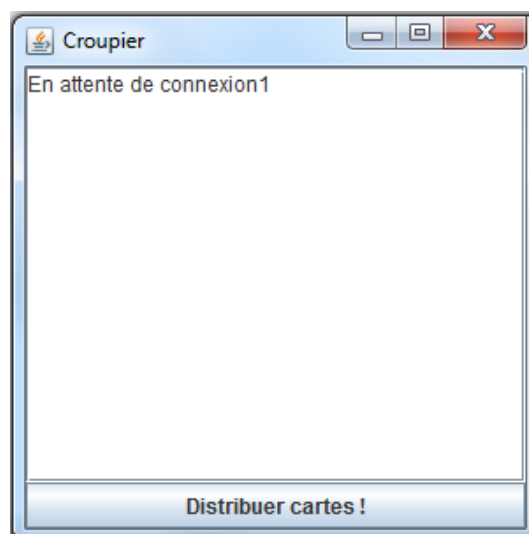
Structuration en couches



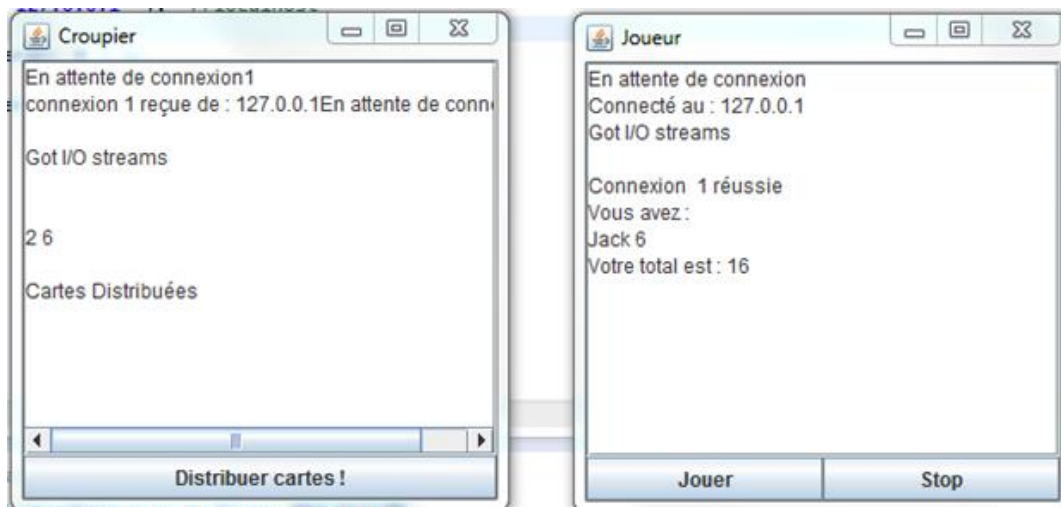
5. Résultat obtenu :

Nous présentons dans cette partie les résultats obtenus.

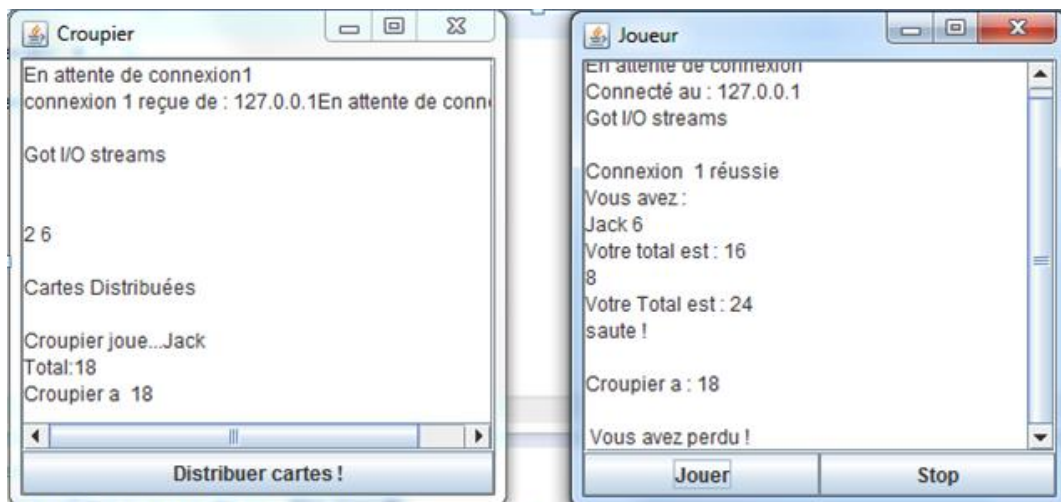
La figure suivante présente le Croupier en attente de connexion.



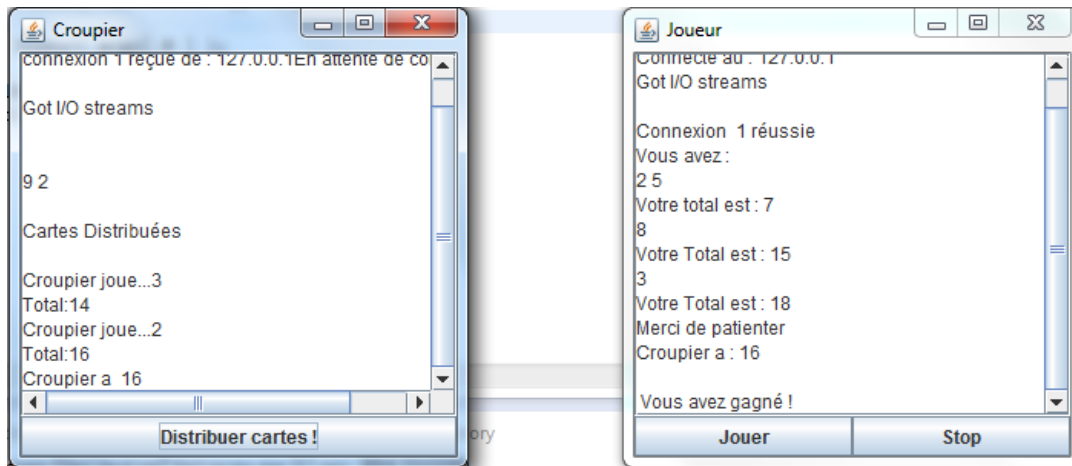
Le joueur se connecte. Le Croupier distribue les cartes.



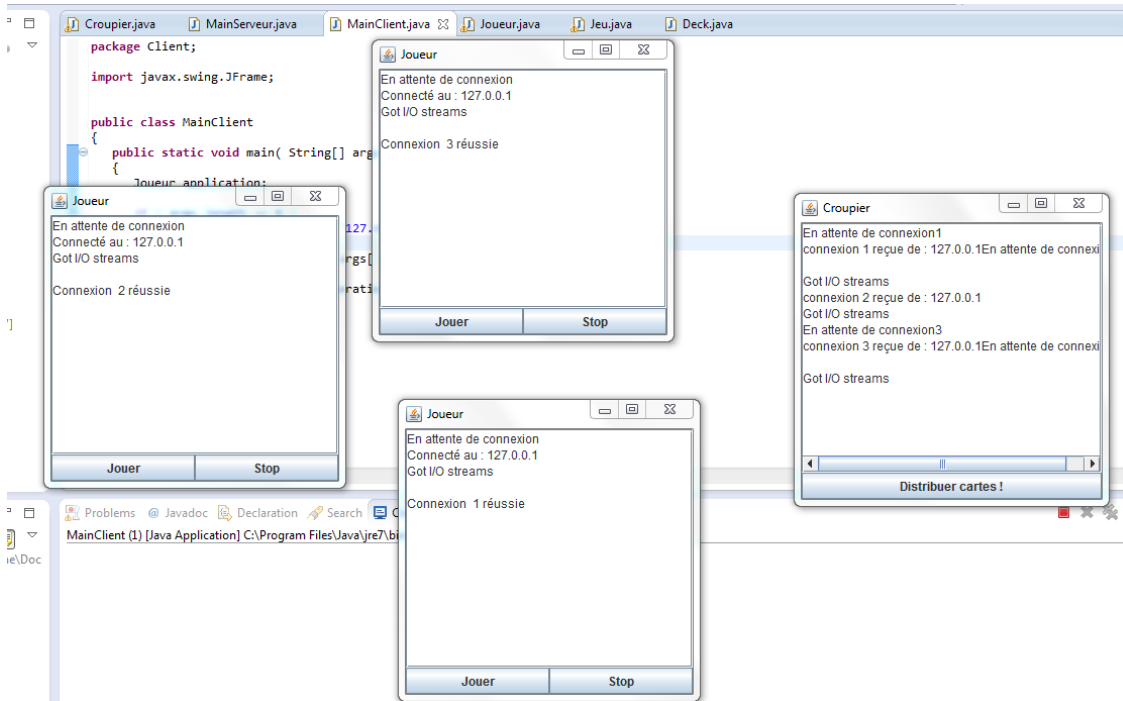
Le Joueur décide de jouer, son total dépasse 21 il perd comme ci-dessous.



Ci-dessous un autre scénario. Le total du joueur est de 18 qui est supérieur au score du Croupier et inférieur à 21. Le joueur gagne.



Ce jeu peut également se jouer à plusieurs.



6. Conclusion

Ce projet m'a permis d'aborder, de comprendre et d'utiliser les technologies client-serveur, précisément d'utiliser les interfaces de connexion. Il m'a permis d'implémenter un jeu, le black Jack et de connaître les rouages de la programmation de ce jeu.

Ce projet m'a permis de capitaliser mes connaissances dans le domaine de la programmation client serveur et m'a permis d'ouvrir les champs d'utilisation de cette technologie. En effet, nous pouvons facilement utiliser cette technologie dans divers domaines (la communication avec l'implémentation d'un chat,...).