



**Rapport intermédiaire 2**  
**Conception Orientée**  
**Objet**

**Groupe PD7WA**



## 1. Point de vue général de l'architecture

### a. Glossaire

**Client** : membre qui communique à travers un réseau entre plusieurs réseaux. Celui qualifié de client a pour rôle d'envoyer des requêtes, des demandes.

**Serveur** : membre qui communique à travers un réseau entre plusieurs réseaux. Le serveur, quant à lui, attend les requêtes du client afin de pouvoir y répondre.

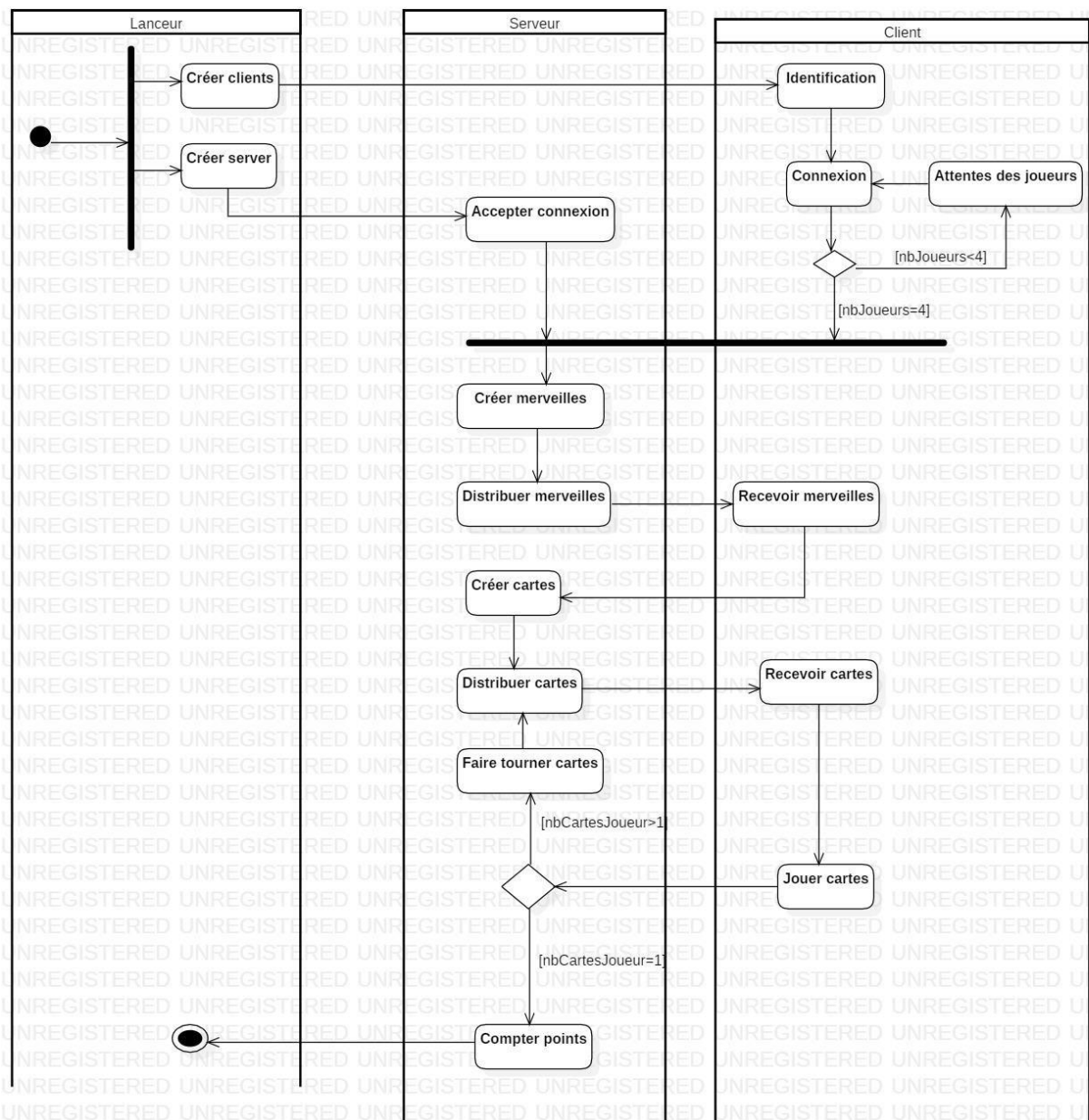
**Diagramme de cas d'utilisation** : le diagramme de cas d'utilisation permet une vision globale du comportement fonctionnel d'un logiciel. Ils mettent en scène les acteurs d'un programme et les actions qui leur correspondent.

**Diagramme d'activité** : cette représentation est qualifiée de comportementale, il permet de modéliser un processus interactif, global ou partiel pour un système donné.

**Diagramme de classe** : ce schéma représente les classes et les interfaces des systèmes ainsi que les relations qui les lient.

**Diagramme de séquences** : il s'agit de la représentation graphique qui présente les interactions entre les acteurs et le système selon un ordre chronologique.

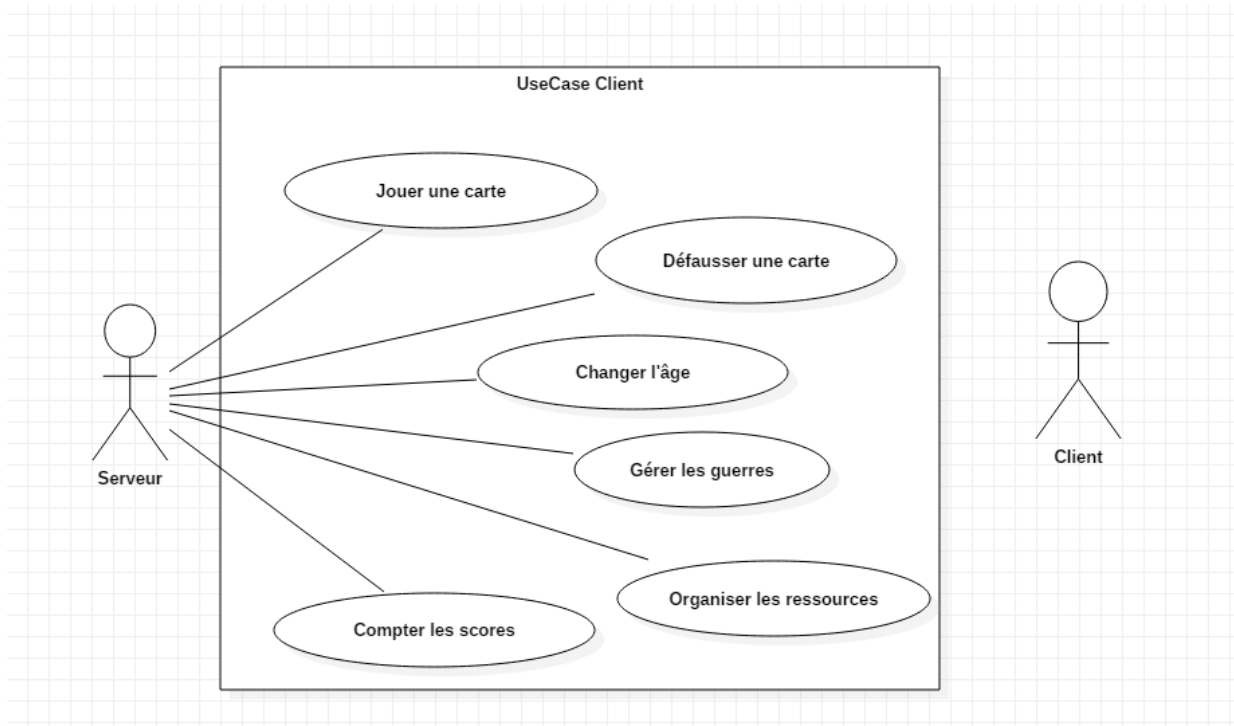
## b. Représentation générale (diagramme d'activité)



Sur le diagramme d'activité nous avons rajouté l'action d'identification après la création des clients et avant la phase de connexion. De plus nous avons modifié le IF pour voir si le nombre de cartes du joueurs est égal à 1 et passer directement à la fin de partie et comptage des points. Le IF après la phase de connexion permet d'attendre les N clients, dans notre cas 4 joueurs pour lancer la partie.

## 2. Client

### a. Analyse des besoins (cas d'utilisation)

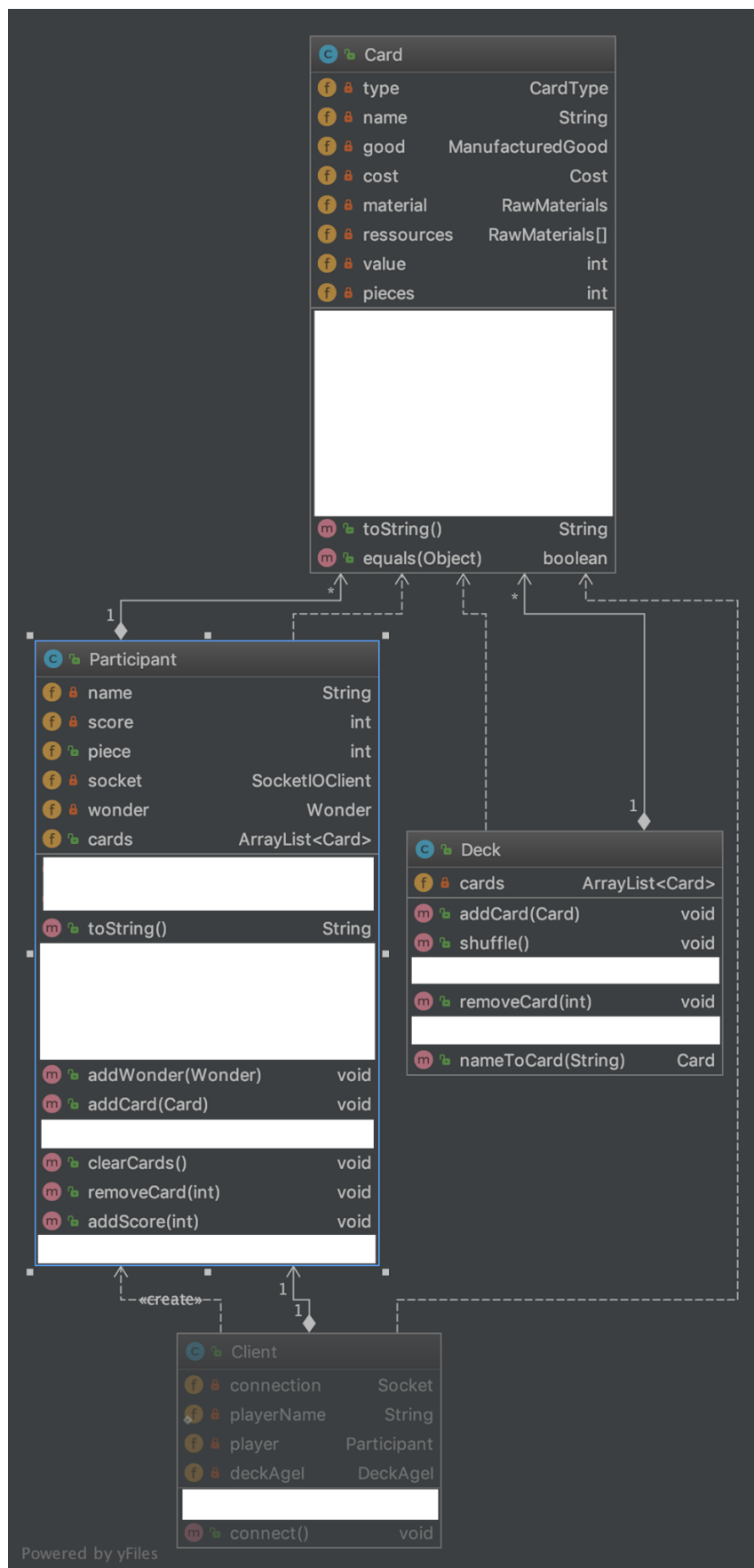


Sur le use case client, nous pouvons voir l'acteur serveur qui peut réaliser plusieurs actions :

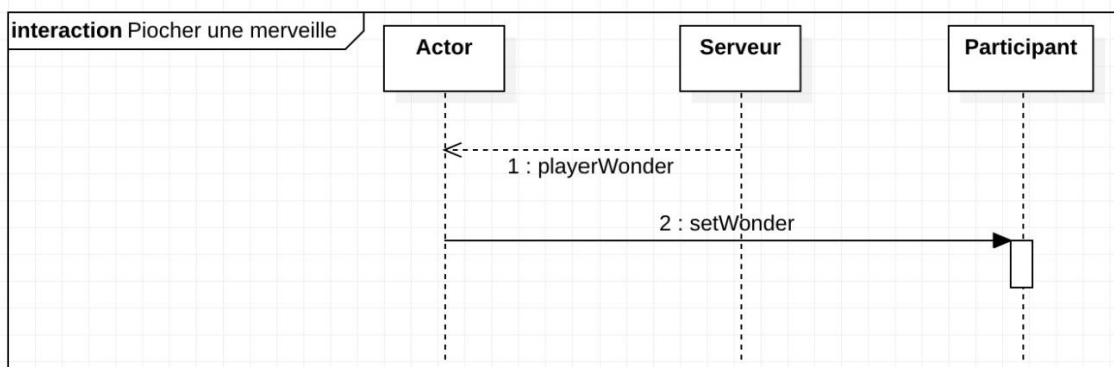
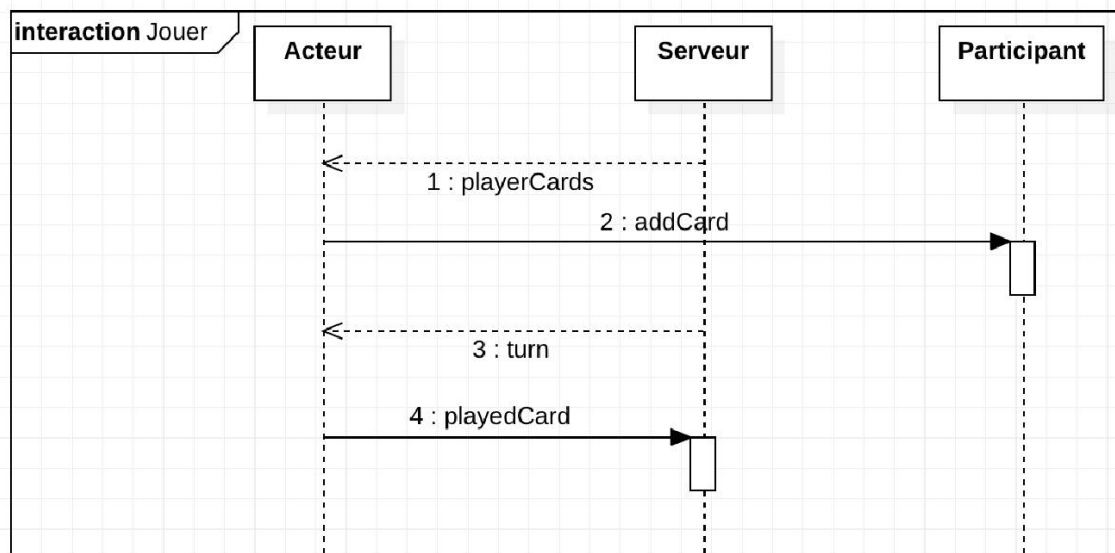
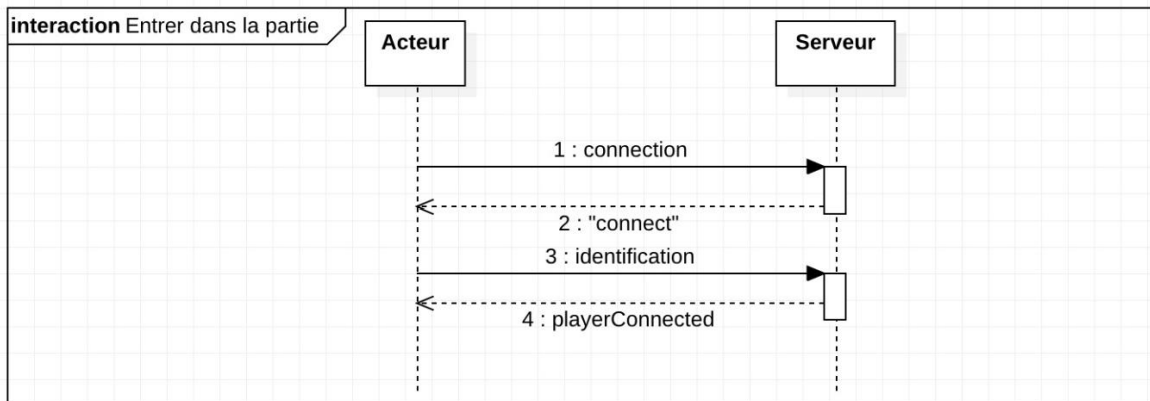
- Jouer une carte
- Défausser une carte
- Changer l'âge
- Gérer les guerres
- Organiser les ressources
- Compter les scores

### b. Conception logicielle

#### i. Point de vue statique (diagramme de classes)

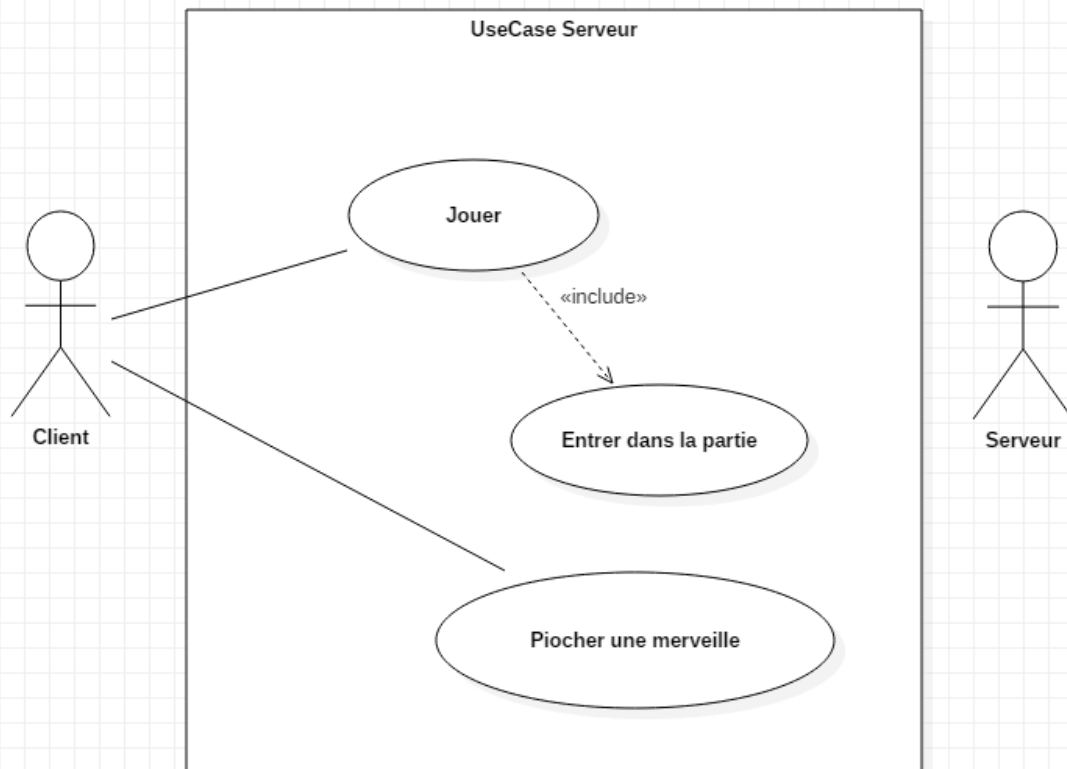


ii. Point de vue dynamique (diagramme de séquence)



### 3. Serveur

#### a. Analyse des besoins (cas d'utilisation)



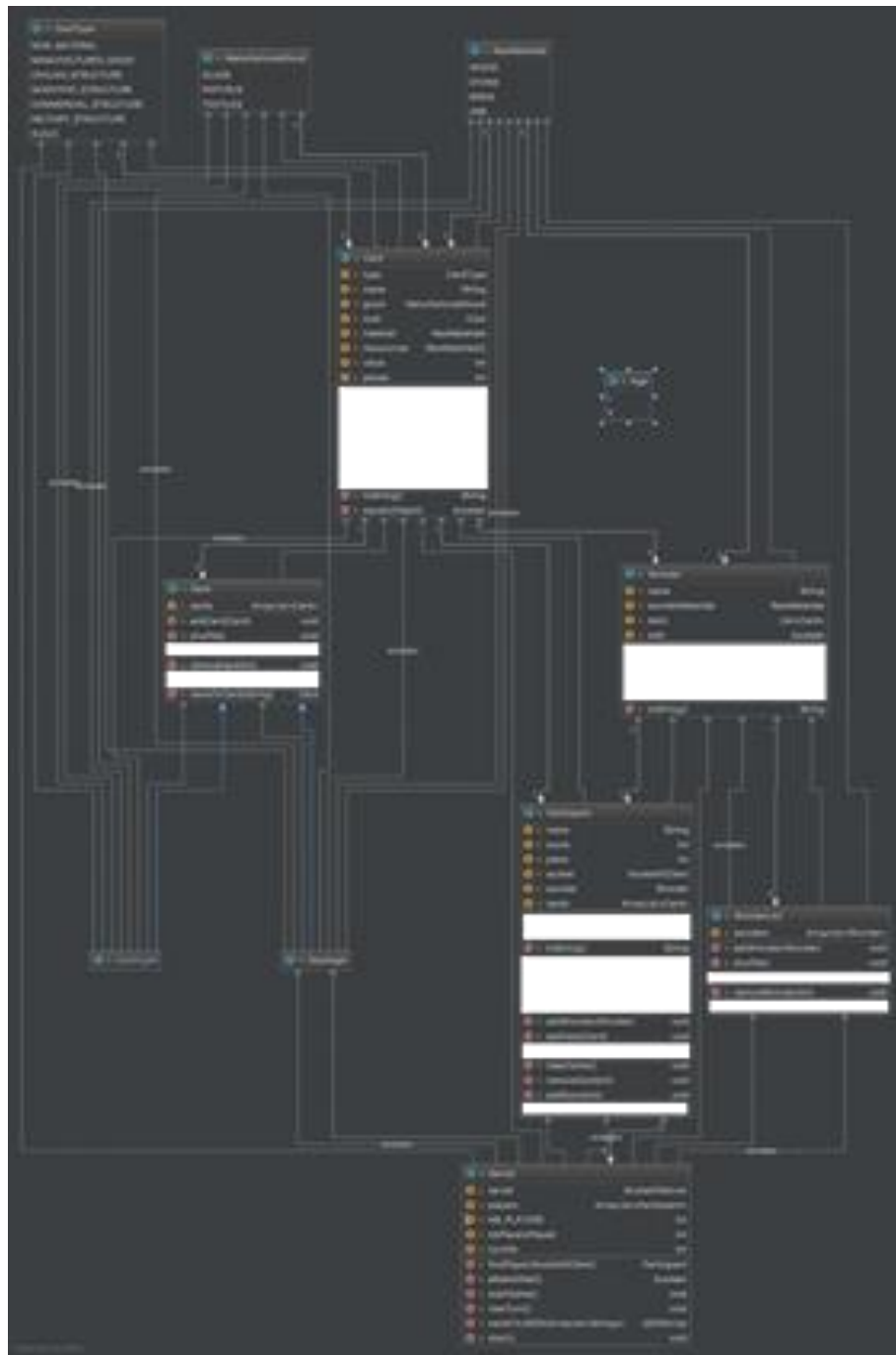
Sur le use case serveur, nous pouvons voir l'acteur client qui peut réaliser l'action de Jouer.

Pour jouer, le client doit être entré dans la partie ce qui implique la phase d'identification et la phase de connexion.

De plus, le client peut piocher une merveille.

#### b. Conception logicielle

i. Point de vue statique (diagramme de classes)



ii. Point de vue dynamique (diagramme de séquence)

4. Interactions entre les clients et les serveurs

5. Conclusion

a. Analyse de notre solution (points forts et points faibles)



Points Forts	Points Faibles
Nous persévérons en prenant en compte les remarques qui nous sont faites afin d'améliorer au maximum notre projet.	Nous avons rencontré des difficultés qui restent d'itérations en itérations et qui nous ralentissent dans notre développement.

b. Evolution prévue

Nous prévoyons d'améliorer notre projet pour arriver à une satisfaction quasi-totale du client. Nous allons nous concentrer un maximum sur l'âge 1 afin qu'il fonctionne de la meilleure façon possible.

Nous sommes actuellement en train de refactoriser le code afin de permettre une meilleure gestion des âges et des scores également.