

**12/04/2019**

RAPPORT COO

Présenté par :

NITEKA Lys Ciella, JALAL Réda et NABIL Inès.

Projet de développement – Encadré par **Elena CABRIO**

**SOMMAIRE**

Table des matières

[INTRODUCTION 2](#_Toc5991394)

[I. Diagramme d’Activité 3](#_Toc5991395)

[II. USE CASE 5](#_Toc5991396)

[a) Use Case côté serveur 5](#_Toc5991397)

[b) Use Case côté client 6](#_Toc5991398)

[III. Diagramme séquence 7](#_Toc5991399)

# INTRODUCTION

"Seven Wonders" est un jeu dont le thème est les sept merveilles du monde. C’est un jeu évolutif ayant pour but la construction des bâtiments des septs merveilles. Le jeu se joue jusqu'à sept joueurs. La main de départ contient sept cartes. Les joueurs développent leur civilisation autour de sept types de cartes.

Le principe est qu’à un tour, chaque joueur choisit une carte de sa main à jouer et passe le reste à son voisin. Ces derniers vont répéter l’action jusqu’à ce que 6 cartes des 7 en mains aient été jouées.

Nous avons alors pour but de modéliser le jeu de Seven Wonder et de le développer en java.

Nous allons exposer nos différents diagrammes de conception afin de montrer la façon dont nous allons procéder.

# Diagramme d’Activité

🡪 Tout d’abord, le serveur lance le jeu et en même temps les clients se connectent en donnant les noms qu’ils vont utiliser en tant que joueur.

🡪 S’il y’a 3 joueurs (sinon la partie n’est pas lancée),le serveur les enregistrent en ajoutant chacun par la suite dans la partie

🡪 Une fois les joueurs ajoutés, le serveur commence la partie en distribuant les cartes (les cartes merveilles et les cartes ressources). Avant le positionnement des joueurs (gauche et à leur droite), le serveur lance la partie

🡪 Après le lancement de la partie, les joueurs choisissent la carte qu’ils désirent poser, ou défausser en fonction de leurs ressources et de leurs golds.

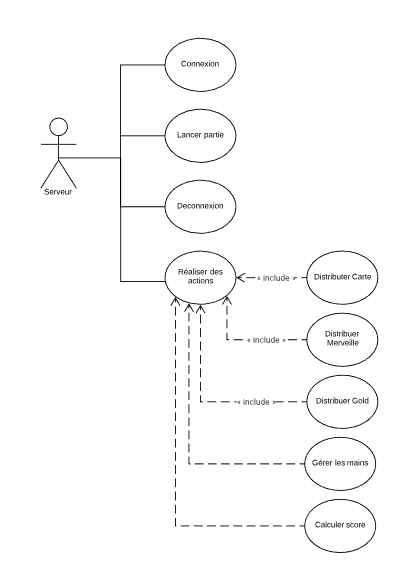
🡪 Le serveur va attendre de recevoir un signal lui communiquant la fin des actions des joueurs pour pouvoir choisir s'il enclenche un nouveau tour ou si l’âge est terminé (s'il y a eu plus de 6 tours).

🡪 Lorsque l’âge se termine, le serveur a le choix entre lancer l’âge suivant (si l’âge actuelle n’est pas 3) ou justement déclarer la fin de la partie à l’âge 3.

🡪 A la fin de la partie, il affiche les scores et termine ses actions.

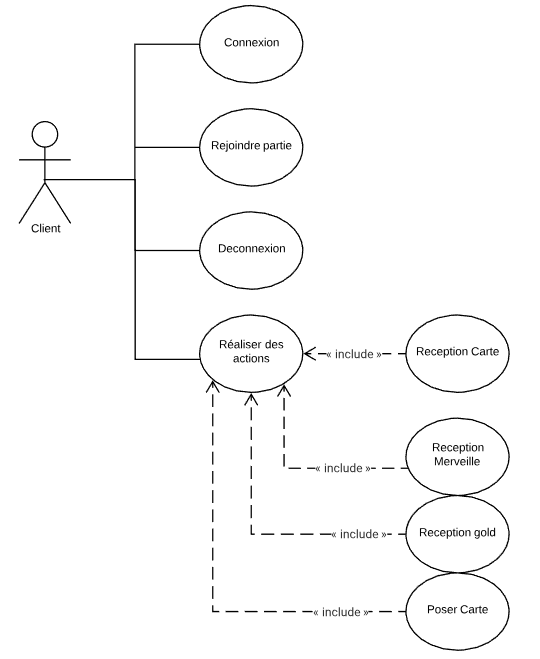
# USE CASE

## Use Case côté serveur



* Lorsque le serveur est lancé, il va attendre la connexion de joueurs, lorsque c’est fait, celui-ci lancera la partie.
* Les joueurs vont s’identifier puis se verront attribués cartes, merveilles et pièces.
* Durant la partie, le serveur réalisera des actions qui permettent l’organisation d’une partie telles que distribuer des cartes, gérer les mains lorsque les cartes sont jouées, calculer le score….
* Lorsque toutes les actions ont été réalisées, les joueurs se déconnecteront puis le serveur se déconnectera.

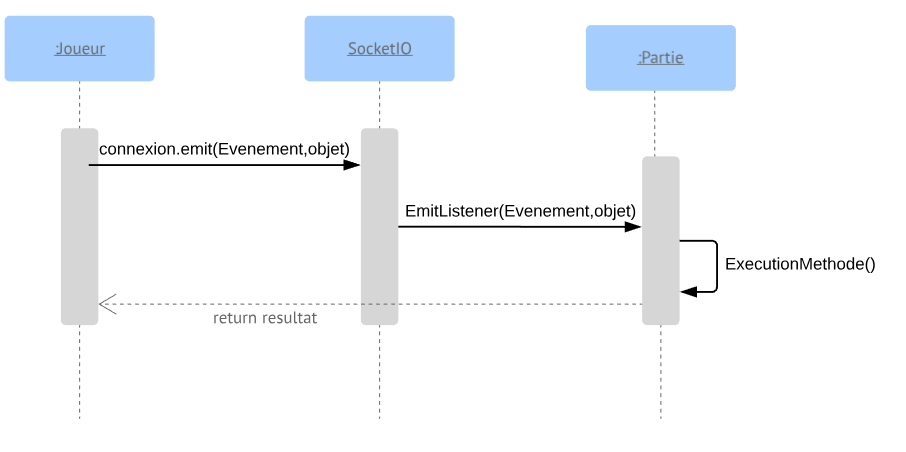
## Use Case côté client



* Les clients vont d’abord se connecter au serveur ensuite émettre un message, en indiquant qu’ils sont bien connectés ainsi que leurs noms.
* Les clients devront attendre l’arrivée de 3 joueurs avant le début de la partie.
* Ils vont recevoir une merveille, 7 cartes et 3 pièces. Les clients vont ensuite attendre un autre signal du serveur leur permettant de commencer à poser leurs cartes.
* Par la suite le serveur enverra des signaux qui permettront aux clients d’effectuer leurs action, par exemple poser une carte. Une fois la carte pose ils transmettront un signal de fin de tour et le serveur fera faire tourner les cartes des joueurs.
* Toutes les actions des clients sont donc gérés par le serveur qui permet le bon déroulement de la partie.
* Une fois les actions des clients terminées, ceux-ci se déconnectent du serveur. La partie est terminée.

# Diagramme séquence

Diagramme de séquence du joueur



# CONCLUSION

On a montré la conception globale faite pour notre projet, où on a démontré en détails les différents diagrammes, qui permettent de bien comprendre le fonctionnement de chaque élément du projet.

On a mis en avant la mise en œuvre d’un système client-serveur dès le début avec la capacité de son utilisation afin de pouvoir manipuler le jeu.

On consacrera plus de temps à mieux améliorer les différents tests unitaires.