

# Projet Genie Logiciel

## Jeu de gestion de train

Par

Allemand Adrien, Amrani Kamil, Guidoux Vincent, Krug Loyse

## Table des matières

Introduction.....	2
Descriptif général du jeu .....	3
Les acteurs.....	3
Mockup de l'interface utilisateur .....	3
Descriptions des cas d'utilisation .....	4
Changer de gare.....	4
Placer une offre .....	4
Achat.....	4
Miner .....	4
Fabriquer .....	4
Améliorer train .....	4
Donner ressources.....	4
Bannir un joueur.....	5
Ajouter une gare.....	5
Supprimer une gare .....	5
Schema des cas d'utilisation.....	5
Conclusion .....	6

## Introduction

Dans le cadre du cours de Génie Logiciel, il nous est demandé de réaliser un projet de programmation par groupe de 4 personnes. Il doit proposer un moyen de se connecter et aussi être assez important pour pouvoir utiliser les techniques de gestion de groupe et de travail vues en cours. Le logiciel doit avoir une base de donnée, deux programmes, un serveur et un client communiquant par Internet – sockets. Nous devons utiliser nos connaissances GIT et JUnit comme ciment à la maison qu'est notre projet. Nous avons décidé de faire un jeu de gestion avec une interface graphique textuelle.

Le jeu consiste en la gestion d'un train-usine dans un univers post-apocalyptique où la collaboration est de mise pour survivre et atteindre l'apogée des technologies de manufacture et d'extraction de matière première.

A l'aide d'une interface simple (O Game like) le joueur pourra lancer des commandes qui auront une influence sur ses ressources et son usine. Ses actions pourront faire partie d'une des catégories suivantes : déplacement du train vers une autre gare, extraction de ressource, fabrication d'objets, commerce avec d'autres joueurs.

Au cours du temps et avec l'évolution des technologies, les actions du joueur voient leur durée augmenter drastiquement. Ainsi, il s'agit d'un jeu se déroulant sur la durée, demandant une connexion pour lancer une commande ou observer l'état de ses ressources. Une fois le joueur déconnecté, le serveur continue à traiter les requêtes qui ont été lancées, jusqu'à leur terminaison.

L'idée du jeu est de favoriser la collaboration, ainsi un joueur avancera plus vite s'il se concentre principalement sur une activité (miner ou manufacturer) et qu'il utilise les mécanismes de commerce pour recevoir des ressources qu'il n'extrait ou ne fabrique pas lui-même.

## Les acteurs

**Joueur**

Un joueur est un utilisateur de l'application pour y avoir un train-usine à gérer. Ses actions sont limitées par les règles du jeu. Pour joueur, il doit posséder un compte. Il a la possibilité s'inscrire en tout temps.

## Admin

Un admin doit se connecter au compte administrateur du serveur. Il peut gérer les différents éléments du jeu : les gares, les ressources et les joueurs.

## Serveur

Un serveur tourne en arrière-plan pour gérer les requêtes des joueurs et de l'admin. Il est actif en continu et est relié à une base de donnée stockant les informations sur tous les joueurs et sur le monde.

## Mockup de l'interface utilisateur

Login

Cool Name

Username

Wrong Username

Password

Wrong Password

Register

Login

The diagram illustrates a client interface. At the top, a horizontal menu bar contains five items: "Train", "Station", "Map", "Trade", and "Inventory", each separated by a vertical line. Below this menu bar, on the left side, is a large rectangular area filled with many thin, horizontal lines, representing a list or a data table. To the right of this area, there are four rectangular buttons stacked vertically, each labeled "Button".

## Descriptions des cas d'utilisation

### Changer de gare

- Un joueur se connecte à l'application
- Il demande la liste des gares au serveur
- Il choisit la gare à laquelle il veut se déplacer
- Pendant le déplacement, un événement aléatoire peut être appliqué au joueur

### Placer une offre

- Un joueur se connecte à l'application
- Il consulte la liste de ses ressources
- Il place une offre sur le marché
- Le serveur valide l'offre du joueur
- Le contenu de l'offre est retirée des ressources du joueur

### Achat

- Un joueur se connecte à l'application
- Il consulte la liste des offres
- Il fait une demande d'achat sur une offre
- Le serveur confirme son achat
- Le contenu de l'offre est ajouté aux ressources du joueur
- Le prix de l'offre est retiré du budget du joueur
- Le prix de l'offre est ajouté au joueur qui a placé l'offre

### Miner

- Un joueur se connecte à l'application
- Il lance le minage
- Le serveur valide le minage
- Le serveur indique combien de ressources ont été minées après un certain laps de temps
- Les ressources extraites sont ajoutées aux ressources du joueur
- Les ressources extraites sont retirées de la gare

### Fabriquer

- Un joueur se connecte à l'application
- Il lance la fabrication d'un objet
- Le serveur valide le lancement de fabrication
- Les ressources nécessaires à la fabrication sont retirées des ressources du joueur
- Après un certain temps le serveur indique que l'objet est construit
- L'objet est ajouté aux ressources du joueur

### Améliorer train

- Un joueur se connecte à l'application
- Il lance une amélioration du train
- Les ressources nécessaires à l'amélioration sont retirées des ressources du joueur
- Après un certain temps le serveur indique que l'amélioration est terminée
- L'amélioration est appliquée au train

### Donner ressources

- Un admin se connecte à l'interface administrateur
- Il sélectionne un joueur ou une gare

- Le serveur valide la sélection
- L'admin indique la ressource à donner et sa quantité
- Le serveur valide le choix
- Les ressources sont ajoutées à la gare/au joueur

### Bannir un joueur

- Un admin se connecte à l'interface administrateur
- Il sélectionne le joueur à bannir
- Le serveur valide le choix
- Le compte du joueur est supprimé

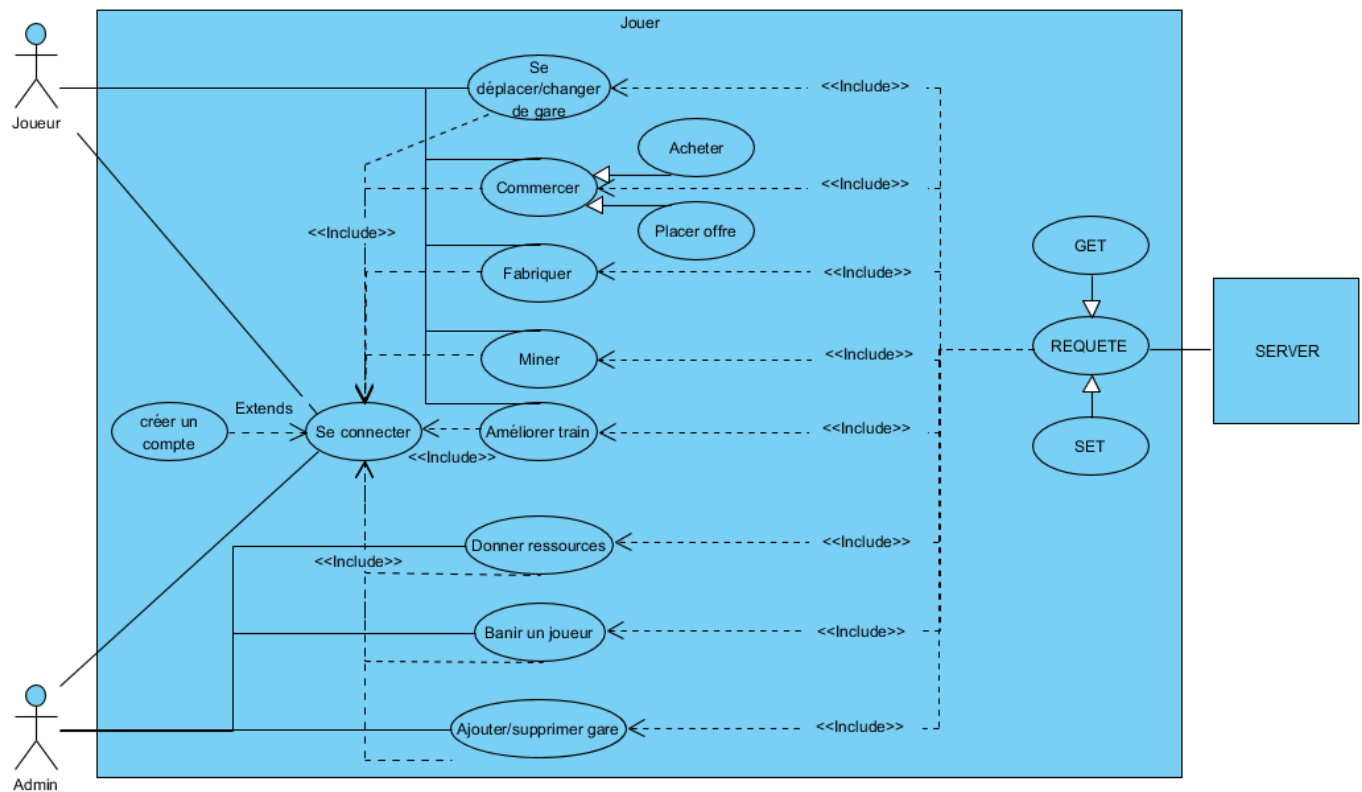
### Ajouter une gare

- Un admin se connecte à l'interface administrateur
- Il entre les infos de la nouvelle gare
- Le serveur valide l'entrée
- La nouvelle gare est ajoutée à la liste des gares

### Supprimer une gare

- Un admin se connecte à l'interface administrateur
- Il sélectionne la gare à supprimer
- Le serveur valide son choix
- La gare est retirée de la liste des gares existantes

## Schema des cas d'utilisation



## Conclusion

Cette étape nous a permis de mettre en place les fondations de notre petite maison que sera le logiciel final, mettre d'accord tout le monde sur le fil rouge principal et surtout, effacer le mirage que nous avions en tête pour la fin du projet par une image plus claire mais qui va sûrement changer en cours de route. Les rôles se forment gentiment dans le groupe, nous nous connaissions déjà alors la moitié du travail de collaboration est déjà fait. Nous avons fait cette étape assez efficacement en utilisant les techniques agiles vues en cours. Tout c'est bien déroulé et dans un temps raisonnable.