

# Projet Genie Logiciel

## OTrain

Par

Allemand Adrien, Amrani Kamil, Guidoux Vincent, Krug Loyse

Rapport

Version 8 : 01.06.2018

## Table des versions

Version	Date
1	13.04.2018
2	27.04.2018
3	30.04.2018
4	04.05.2018
5	11.05.2018
6	18.05.2018
7	25.05.2018
8	01.06.2018

## Table des matières

Introduction.....	5
Descriptif du jeu.....	5
Fonctionnement général .....	5
Déplacement de train .....	5
Extraction de ressources.....	5
Fabrication d'objets .....	5
Commerce .....	5
Un jeu sur la durée .....	6
Les acteurs.....	6
Fonctionnalités secondaires .....	6
Mockup de l'interface utilisateur .....	7
Descriptions des scénarios possibles.....	8
Se connecter .....	8
S'inscrire .....	8
Changer de gare.....	8
Placer une offre .....	8
Achat.....	8
Miner .....	8
Fabriquer .....	8
Améliorer train .....	9
Donner ressources.....	9
Bannir un joueur.....	9
Ajouter une gare.....	9
Supprimer une gare .....	9
Schema des cas d'utilisation.....	10
Client-Serveur .....	10
Responsabilités client-serveur.....	10

Responsabilité client.....	10
Responsabilité serveur .....	10
Lancement et arrêt du jeu .....	10
Protocole d'échange client serveur .....	11
Modèle de domaine .....	13
Base de données.....	13
Distribution des rôles.....	14
Backlogs de produit .....	15
Stories IceScrum .....	15
Plan d'itérations.....	16
1 <sup>er</sup> Sprint .....	16
Objectif .....	16
Stories à faire .....	16
Tasks .....	16
Tests d'acceptation.....	16
Bilan d'itération .....	17
Sprint 2.....	18
Objectifs.....	18
Stories à faire .....	18
Tasks .....	18
Tests d'acceptation.....	19
Bilan d'itération .....	20
Sprint 3.....	21
Objectifs.....	21
Stories à faire .....	21
Tasks .....	21
Tests d'acceptation.....	23
Bilan d'itération .....	25
Sprint 4.....	26
Objectifs.....	26
Stories à faire .....	26
Tasks .....	26
Test d'acceptation .....	27
Bilan d'itération .....	28
Sprint 5.....	29
Objectifs.....	29
Stories à faire .....	29

GEN-2018	Allemand Adrien, Amrani Kamil, Guidoux Vincent, Krug Loyse
Tasks .....	29
Test d'acceptation .....	29
Bilan d'itération .....	31
Sprint 6.....	31
Objectifs.....	31
Stories à faire .....	32
Tasks .....	32
Test d'acceptation .....	33
Bilan d'itération .....	35
Sprint 7.....	35
Objectifs.....	35
Stories à faire .....	35
Tasks .....	35
Test d'acceptation .....	35
Bilan d'itération .....	35
Sprints à venir .....	36
Conclusion .....	36

## 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 données, 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.

## Descriptif du jeu

OTrain consiste en un jeu de gestion de trains-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.

### Fonctionnement général

OTrain se veut un jeu à interface graphique simple (O Game like), genre tableau de bord, avec des boutons d'action, du texte et d'éventuelles images illustratives.

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.

### Déplacement de train

Un joueur peut demander le déplacement d'une gare à une autre après avoir récupéré la liste des gares alentours. Son déplacement une fois validé, un compte à rebours est lancé pour lui indiquer sa progression vers sa destination. Un joueur se déplace de gare en gare dans l'objectif d'y trouver toujours de meilleures ressources.

### Extraction de ressources

A chaque gare, une certaine quantité de ressources est disponible et les trains s'y trouvant peuvent l'extraire. Pour récupérer de la matière première destinée à alimenter les machines, fabriquer de nouveaux objets ou améliorer son train. Une ressource est minée sur une certaine durée et le joueur peut voir dans la durée sa quantité de minerais augmenter.

### Fabrication d'objets

Avec ses ressources disponibles, le joueur peut décider de fabriquer des objets plus complexes. Cette fabrication consomme des ressources et prends un certain temps pendant lequel le joueur peut lancer d'autres processus. L'objectif de fabrication d'objet est de pouvoir utiliser dans la durée des technologies de plus en plus puissantes pour faire évoluer son train.

### Commerce

Les ressources n'étant pas disponibles partout, un joueur seul pourra avoir de la peine à atteindre les technologies les plus évoluées. Pour cette raison, un espace de commerce central regroupe toutes les offres ou demandes qu'un joueur pourrait formuler. La monnaie du jeu est le Scrum

## Un jeu sur la durée

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.

Le jeu se passe dans un univers pseudo-infini et ainsi ne connaît ni niveaux, ni fin, l'« objectif » étant d'enrichir son train et l'améliorer sans cesse.

L'idée de OTrain 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.

## Fonctionnalités secondaires

Ce jeu comporte des fonctionnalités de plusieurs degré d'importance. Les fonctionnalités principales sont celles décrites dans le fonctionnement général du jeu. Il sera néanmoins possible d'ajouter des éléments pour le diversifier, une fois sa première version terminée.

- Le joueur a une vue restreinte des gares autour de lui, plus elles sont proches de lui, plus il en connaît les détails
- Ajout d'éléments aléatoires ayant lieu lors des déplacements en train
- Création d'un marché propre à chaque gare, pour inviter au déplacement
- Les gares sont des points d'intérêts proposant de temps à autre des « missions » aux joueurs
- Des fabrications d'objets en plusieurs étapes. Un objet complexe demande la fabrication de petits objets au préalable. Si un joueur lance la production d'un objet complexe, c'est tout son processus de production qui est lancé (du clou à la loco). Le joueur pourrait alors demander un arrêt de fabrication et conserve les objets déjà créés et les ressources pas encore utilisées. (Système Factorio like)
- Permettre aux serveurs de se multiplier

Login

## Cool Name

Username  Wrong Username

Password  Wrong Password

Client

Train | Station | Map | Trade | Inventory

TEXTE

## Descriptions des scénarios possibles

### Se connecter

- L'utilisateur va sur la page de login
- Il entre son nom d'utilisateur et son mot-de-passe
- Si son authentification fonctionne (il a un compte et son mot de passe est correct), il se retrouve envoyé sur son tableau de bord.

### S'inscrire

- Le joueur va sur la page de sign up
- Il entre ses données (nom d'utilisateur et mot de passe)
- Si ses informations sont acceptées, son compte est créé et il est envoyé sur la page de login pour se connecter.

### Changer de gare

- Un joueur demande la liste des gares au serveur
- Il choisit la gare à laquelle il veut se déplacer
- Si son mouvement est validé (une place dans la nouvelle gare lui est réservée) il entame son trajet.
- Au bout d'un certain temps, il arrive à destination.

### Placer une offre

- Un joueur consulte la liste de ses ressources (sur son tableau de bord)
- Il place une offre sur le marché central du jeu (un objet ou une ressource qu'il voudrait vendre)
- Si son offre est validée, le contenu de l'offre est retiré des ressources du joueur pour être placé dans une zone tampon.
- Si le joueur retire son offre avant sa vente, ses ressources lui sont rendues

### Achat

- Un joueur consulte la liste des offres sur le marché central du jeu
- Il fait une demande d'achat sur une offre
- Si sa demande est validée 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 consulte la liste des ressources disponibles dans sa gare courante
- Il lance une demande de minage d'une ressource
- Si sa demande est acceptée, il commence à extraire la matière première
- Le serveur indique combien de ressources ont été minées après un certain laps de temps (cette information est renouvelée régulièrement pendant le minage)
- Les ressources extraites sont ajoutées aux ressources du joueur
- Les ressources extraites sont retirées de la gare

### Fabriquer

- Un joueur consulte la liste des objets qu'il peut fabriquer
- Il demande la fabrication d'un objet
- Si sa demande est validée, le processus de fabrication est enclenché
- Les ressources nécessaires à la fabrication sont retirées des ressources du joueur
- Après un certain temps l'objet est terminé et est ajouté aux ressources du joueur



### Améliorer train

- Un joueur consulte la liste des améliorations possibles de son train
- Il demande une amélioration d'une partie de son train.
- Si la demande est acceptée Les ressources nécessaires à l'amélioration sont retirées des ressources du joueur
- Après un certain temps, l'amélioration est terminée et est ajoutée au train

### Donner ressources

- Un admin consulte la liste des joueurs/gares
- Il sélectionne un joueur ou une gare
- Il indique la ressource à donner et sa quantité
- Les ressources sont ajoutées au joueur/à la gare

### Bannir un joueur

- Un admin consulte la liste des joueurs
- Il sélectionne le joueur à bannir
- Le compte du joueur est supprimé

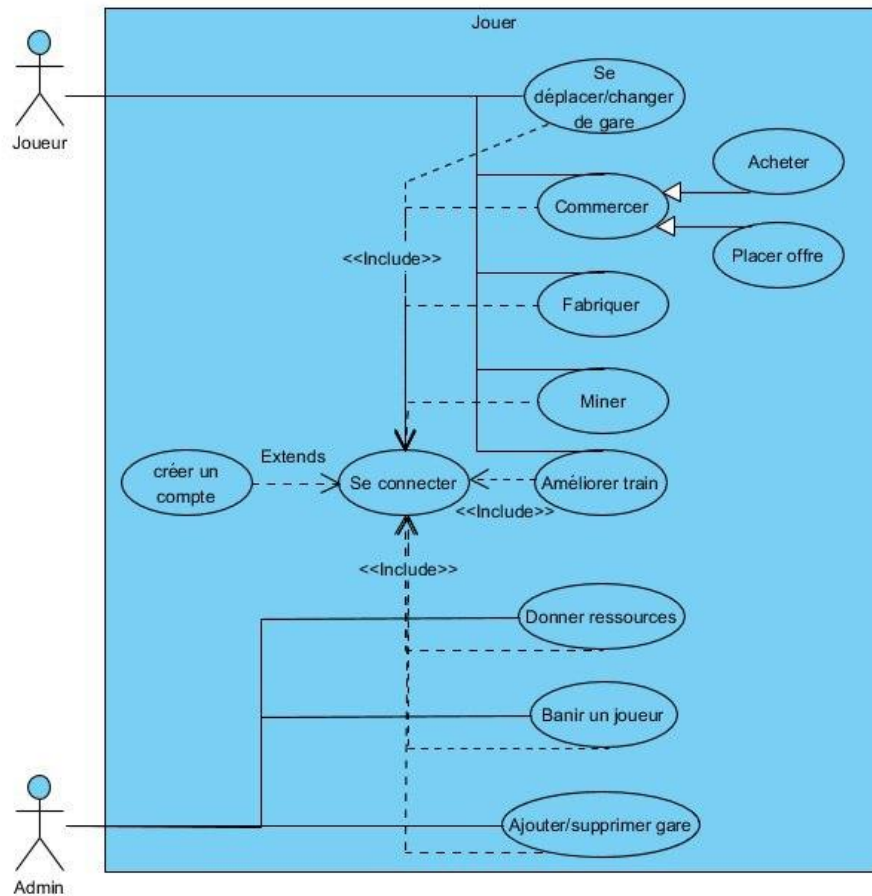
### Ajouter une gare

- Un admin va sur la page de création de gare
- Il entre les infos de la nouvelle gare
- La nouvelle gare est ajoutée à la liste des gares

### Supprimer une gare

- Un admin consulte la liste des gares du jeu
- Il sélectionne la gare à supprimer
- La gare est retirée de la liste des gares existantes

## Schema des cas d'utilisation



## Client-Serveur

### Responsabilités client-serveur

#### Responsabilité client

Le client est un observateur contrôleur du serveur. Il va récupérer des informations et envoyer des requêtes pour effectuer des actions avec son train.

#### Responsabilité serveur

Le serveur récupère et valide les requêtes du client. Il s'occupe de faire tous les calculs et contrôles. Il est le seul à avoir accès à la base de données.

#### Lancement et arrêt du jeu

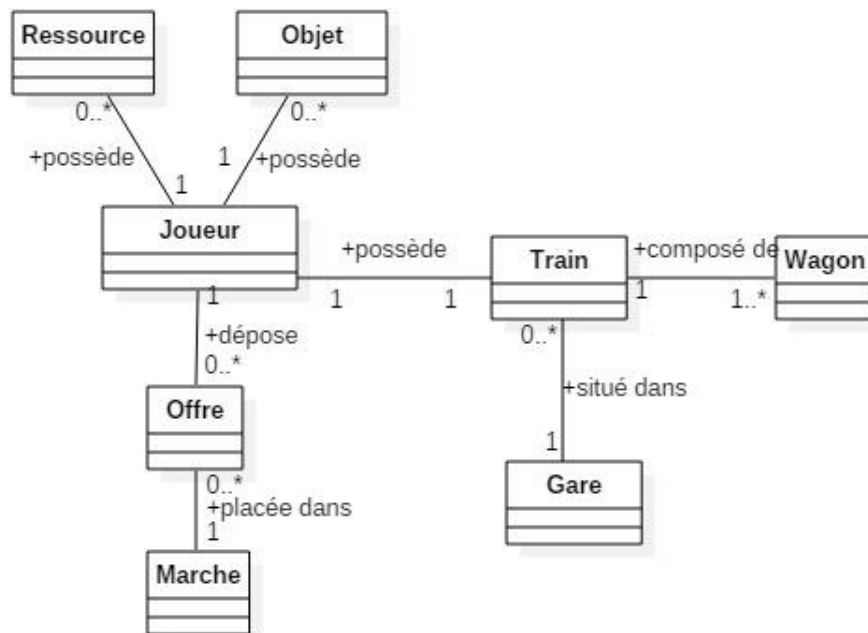
Un client démarre le jeu côté client en se connectant, et le quitte en se déconnectant. Du côté serveur le jeu tourne en continu, tant que le serveur est lancé.

## Protocole d'échange client serveur

No	Description CLI	Client Command	Server Answer	Description SRV
<b>Changer de Gare</b>				
1	Demander les gares disponibles	GET_GARES		
2			JSON<Gares>	Une liste de gare Jsonifiées
3	La gare de destination	GO_TO <id gare>		
4			SUCCESS    FAILURE	Si la demande est valide, le serveur initie le déplacement et réponds SUCCESS, sinon il réponds FAILURE
<b>Etat du train</b>				
1	Demander l'état du train	GET_TRAIN_STATUS		
2			AT_STATION	Si le train est a une gare sans être partis
			JSON <la gare actuelle>	la gare ou il est
			ON_THE_WAY	si le train est en route
			<temps restant en secondes>	le temps avant l'arrivée
			ARRIVED	Si le train est arrivé a destination après un déplacement
			JSON<Evenement>	Le résultat du voyage
<b>Placer une offre</b>				
1	Demander de placer une offre l'offre en question	SET_OFFRE JSON<Offre>		
2			SUCCESS    FAILURE	Si la demande est valide, le serveur initie le placement de l'offre et réponds SUCCESS, sinon il réponds FAILURE
<b>Acheter une ressource</b>				
1	Demander la liste des offres	GET_OFFRES		
2			JSON<Offres>	retourne la liste des offres
3	Choisir une offre a acheter	BUY_OFFRE <id offre>		

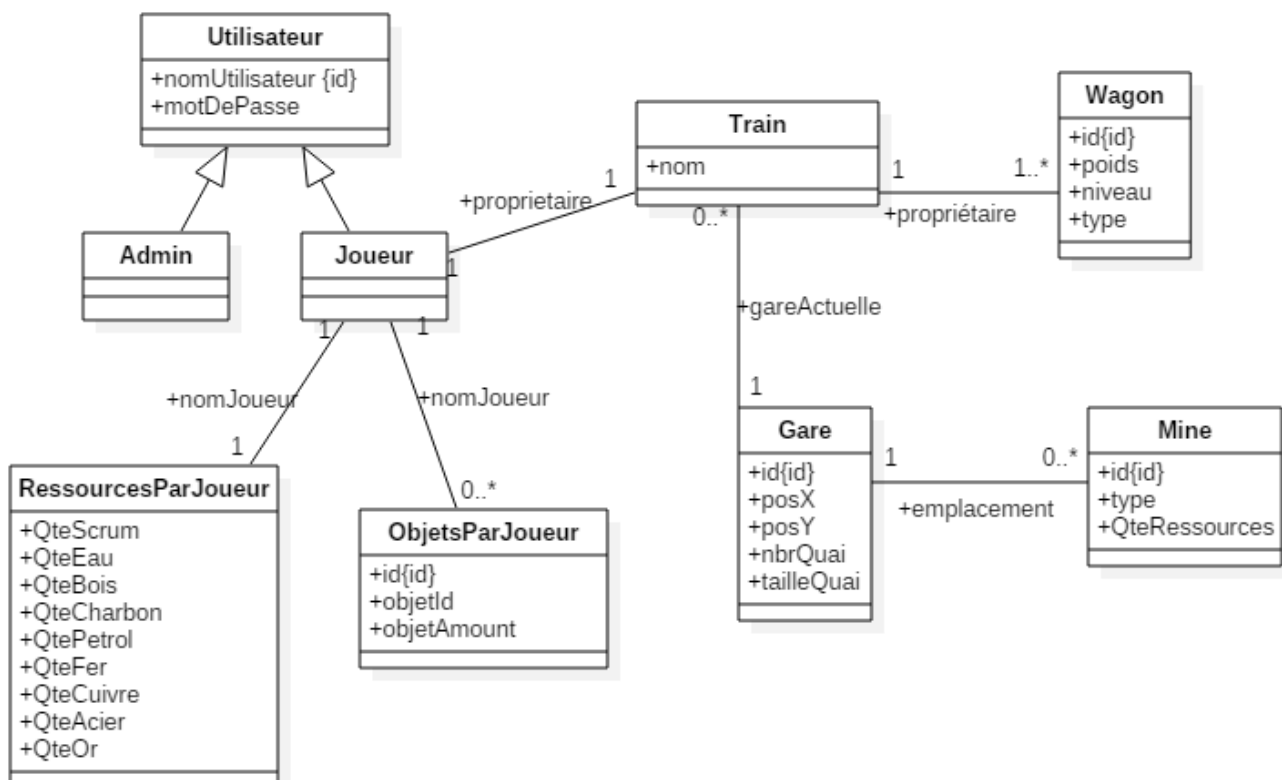
4			SUCCESS    FAILURE	Si la demande est valide, le serveur transfère les ressources et réponds SUCCESS, sinon il réponds FAILURE
<b>Consulter ses ressources</b>				
1	Demande l'état de ses ressources	GET_RESSOURCES		
2			JSON<Ressources>	retourne la liste des offres
<b>Miner</b>				
1	Demande de démarrer le minage d'une ressource	MINE		
	la ressource qu'on désire miner	<id ressource>		
2			SUCCESS    FAILURE	Réponse que le minage a pu être lancé
3	demande d'arrêter le minage	STOP_MINE		
4			SUCCESS    FAILURE	Réponse que le minage a pu être arrêté
<b>Fabrication</b>				
1	Demande l'état de la queue de production			
2			JSON<Queue de production>	Retourne la queue de production
3	Demande l'ajout d'un craft	CRAFT <id objet> <quantité>		
4			SUCCESS    FAILURE	Retourne si le craft a pu être ajouté à la queue
5	Demande l'annulation d'un craft	CANCEL_CRAFT <id groupe de prod>		
6			SUCCESS    FAILURE	Retourne si le craft a pu être annulé
<b>Connexion</b>				
1	Demande de connexion authentifiée	CONNECT <nom> <mdp>		
2			SUCCESS    FAILURE	Retourne si la connexion a fonctionné

## Modèle de domaine



## Base de données

La base de données du jeu devra contenir les informations sur les joueurs, les ressources et les objets contenus dans le jeu. Elle doit permettre au serveur d'aller vérifier l'état d'un joueur avant de lui autoriser une action. C'est également dans la base de données que seront contenu les informations destinées à l'authentification du joueur.



## Distribution des rôles

**Membres de l'équipe :** Allemand Adrien, Amrani Kamil, Guidoux Vincent, Krug Loyse

**Scrum master :** Kamil Amrani

**Project Owner :** Vincent Guidoux

**Equipe de développement :** Allemand Adrien, Amrani Kamil, Guidoux Vincent, Krug Loyse

**Testeurs :** Allemand Adrien, Amrani Kamil, Guidoux Vincent, Krug Loyse

## Backlogs de produit

### Stories IceScrum

Rank	Name	description	type
1	Se connecter	Le joueur va sur la page de login, entre son nom d'utilisateur et son mot de passe pour se connecter. Si ses identifiants sont corrects il est envoyé sur son tableau de bord	Technique
2	Changement de gare	Le joueur demande à se déplacer jusqu'à une gare. Si c'est possible, le déplacement est lancé, sinon, il reste où il est.	Utilisateur
3	Consulter ses ressources	Le joueur peut voir ses ressources sur un tableau de bord	Utilisateur
4	Miner	Le joueur déploie son matériel de minage et commence à extraire une ressource dans la gare où il se trouve. Son minage s'arrête quand la ressource est épuisée ou s'il a donné l'ordre d'arrêt	Utilisateur
5	Fabriquer	Le joueur crée de nouvelles ressources à partir de ses ressources actuelles	Utilisateur
6	Améliorer son train	Le joueur peut utiliser des ressources pour améliorer des parties de son train.	Utilisateur
7	Ajouter une gare	L'administrateur peut ajouter une gare dans la partie	Utilisateur
8	Supprimer une gare	l'admin peut supprimer un gare du jeu	Utilisateur
9	Donner des ressources	l'administrateur donne des ressources à un joueur ou à une gare	Utilisateur
10	Bannir un joueur	L'administrateur peut bannir un joueur du jeu	Utilisateur
11	Placer une offre	Le joueur pose une offre sur le marché	Utilisateur
12	Acheter des ressources	Le joueur peut voir la liste des offres et en acheter, s'il en a les moyens.	Utilisateur

# Plan d'itérations

## 1<sup>er</sup> Sprint

### Objectif

Pouvoir se connecter au serveur

### Stories à faire

Se connecter

### Tasks

VG : Vincent Guidoux, KA : Kamil Amrani, AA : Adrien Allemand, LK : Loyse Krug

30

VG

?

?

Mise en place du projet

2

KA

?

?

Création du serveur pour communiquer

1

LK

?

?

Création de la base de donnée

3

AA

?

?

Création du protocole

4

VG

?

?

Création du client

### Tests d'acceptation

1

Identifiant correctes renvoient un accès

✓ To check

No description

2

Identifiant incorrects, on redemande la connexion

✓ To check

No description

Estimation par tâche [heure/personne] :

No tâche	30	2	1	3	4
Durée[heure/personne]	0.5	4	3	3	4



## Bilan d'itération

Sprint 1 - Done  
27/04 > 03/05 8 / 8 \$ 0 M

Urgent tasks (0)

Recurrent tasks (0)

★ 1 Se connecter  
8 \$

3 AA Création du protocole	4 VG Création du client	30 VG Mise en place du projet	2 KA Création du serveur pour communiquer	1 LK Création de la base de donnée
-------------------------------	----------------------------	----------------------------------	--	---------------------------------------

*Bilan Scrum master*

## Bilan de terminaison des histoires

Toutes les histoires prévues ont été terminées

## Vélocité du sprint

8 points d'histoire

## Replanification

Pour le 2<sup>ème</sup> sprint, il faudra lier la base de données à l'application. De plus on ajoute une story pour la création d'une interface graphique permettant de s'inscrire dans la base de données et de se connecter.

La story de changement de gare est déplacée au sprint 3 pour donner le temps de finaliser la connexion avec la base de données.

## Commentaire général

Le projet avance bien.

## Autocritique

-

*Bilan personnels***Adrien Allemand**

Temps prévu: 4h pour réaliser le protocole

Temps réalisé : 1h30

Commentaire : -

**Kamil Amrani**

Temps prévu: 4h pour réaliser l'implémentation du serveur

Temps réalisé : environ 4h

Commentaire : Beaucoup de debug à la fin du sprint

Temps prévu: 4h pour réaliser l'implémentation du client - 0.5h pour la mise en place du projet

Temps réalisé : envrion 2h

Commentaire : Merci à Kamil pour l'aide

### Loyse Krug

Temps prévu: 3h pour réaliser la base de donnée

Temps réalisé : 2h30

Commentaire : -

## Sprint 2

### Objectifs

Pouvoir changer de gare et consulter ses ressources

### Stories à faire

Consulter ses ressources, lier la base de données avec le serveur, interagir avec le serveur via une interface graphique

### Tasks

VG : Vincent Guidoux, KA : Kamil Amrani, AA : Adrien Allemand, LK : Loyse Krug

Recurrent tasks (0)

<div>★ 5 Consulter ses ressources</div> <div>2 \$</div> <div>New task</div>		<div>36 LK ?</div> <div>Créer une gui pour l'affichage des ressources</div>	
<div>★ 13 Lier la base de donnée au serveur</div> <div>2 \$</div> <div>New task</div>		<div>31 KA ?</div> <div>Créer une classe de connexion à la base de données</div>	
<div>★ 14 Interagir avec le serveur via une interface graphique</div> <div>5 \$</div> <div>New task</div>		<div>32 AA ?</div> <div>Créer une gui de login</div>	<div>33 AA ?</div> <div>Créer une gui de sign up</div>
<div>☆ 15 Créer une classe qui gère les ressources</div> <div>3 \$</div> <div>New task</div>		<div>38 VG ?</div> <div>Créer classe qui gère les ressources</div>	<div>39 VG ?</div> <div>générer le JSON</div>
			<div>40 VG ?</div> <div>Lire un JSON</div>

## Tests d'acceptation

## Consulter ses ressources :

18 Modifier les ressources manuellement dans le ☒ To check



No description

45 Voir la page affichant les ressources du joueur ☒ To check



No description

## Lier la base de données au serveur :

37 le lancement de la fonction de connexion renv ☒ To check



No description

## Interagir avec le serveur via une interface graphique :

38 Voir l'interface graphique de login ☒ To check



No description

39 voir l'interface graphique de sign up ☒ To check



No description

40 Login-entrer un nom d'utilisateur absent de l ☒ To check



No description

41 Login --entrer un nom d'utilisateur et un mau ☒ To check



No description

42 Login - entrer un nom d'utilisateur et le bon r ☒ To check



**Given**  
**When**  
**Then**

43 Sign up - entrer un nom d'utilisateur deja exi ☒ To check



No description

44 Sign-up entrer un nom d'utilisateur et mot de ☒ To check



No description

Créer une classe qui gère les ressources :

46
Pouvoir ajouter/enlever des ressources
✓ To check
...

No description

47
Afficher les ressources
✓ To check
...

**Given** être connecté  
**When** Demander les ressources  
**Then** Affiche correctement les ressources et les quantités

48
générer le JSON
✓ To check
...

No description

49
Lire un JSON
✓ To check
...

No description

Estimation par tâche [heure/personne] :

No tâche	36	31	32	33	38	39	40
Durée[h/p]	3	3	2	1	1	1	1

## Bilan d'itération

Sprint 2 - Done  
04/05 > 10/05 12/93 03

DONE (7)

Recurrent tasks (0)

★ 15 Créer une classe qui gère les ressources  
3 \$

38 VG Créer classe qui gère les ressources  
40 VG Lire un JSON  
39 VG générer le JSON

★ 14 Interagir avec le serveur via une interface graphique  
5 \$

33 AA Créer une gui de sign up  
32 AA Créer une gui de login

★ 13 Lier la base de donnée au serveur  
2 \$

31 CA Créer une classe de connexion à la base de données

★ 5 Consulter ses ressources  
2 \$

36 LK Créer une gui pour l'affichage des ressources

### Bilan Scrum master

### Bilan de terminaison des histoires

Toutes les histoires prévues ont été terminées

### Vélocité du sprint

12 points d'histoire

## Replanification

Pour le 3<sup>ème</sup> sprint nous avons ajouté une story pour la finalisation de la GUI

## Commentaire général

On a bien travaillé pour ce sprint, on a fini légèrement en avance.

## Autocritique

-

## Bilan personnels

### Adrien Allemand

Temps prévu: 3h pour la gui du login et du sign up

Temps réalisé : 3

Commentaire : -

### Kamil Amrani

Temps prévu: 3h pour créer une classe de connexion à la base de données

Temps réalisé : 3h

Commentaire :

### Vincent Guidoux

Temps prévu:3h pour la classe qui gère les ressources, lire un JSON et générer un GSON

Temps réalisé : envrion 3h

Commentaire :

### Loyse Krug

Temps prévu: 3h pour la création de la GUI de l'affichage des ressources

Temps réalisé : 3h

Commentaire : -

## Sprint 3

## Objectifs

Pouvoir miner/récolter des ressources

## Stories à faire

Miner, Changement de gare, Finaliser la GUI

## Tasks

VG : Vincent Guidoux, KA : Kamil Amrani, AA : Adrien Allemand, LK : Loyse Krug

<p>★ 6 Miner 5 \$</p> <p>New task</p>	<p>20 VG 2 2 Implementer regeneration des ressources</p> <p>17 KA 2 2 Le serveur gère le protocole de minage</p> <p>41 VG 2 2 Implémenter class wagon de minage</p> <p>18 LK 1 1 Request SQL pour ajouter des mines aux qares</p> <p>19 LK 1 1 Request SQL pour ajouter wagons de minage à un train</p>
<p>★ 2 Changement de gare 5 \$</p> <p>New task</p>	<p>6 KA 2 2 définir critère de déplacement</p> <p>5 KA 1 1 Le serveur gère le protocole changement de gare</p> <p>7 LK 1 1 Request SQL pour creer des gares</p> <p>8 LK 1 1 Request SQL pour changer de gare</p> <p>9 AA 2 2 GUI pour le statut du train d'un joueur</p> <p>10 LK 1 1 Request SQL pour creer un train</p>
<p>☆ 16 Finaliser la GUI 3 \$</p> <p>New task</p>	<p>42 AA 2 2 Définir les informations permanentes</p> <p>43 AA 2 2 Définir les tabs (déplacement, miner, ...)</p> <p>44 VG 1 1 Changer le code Client</p>

## Miner

19 Les ressources sont retirées de la gare

**Given** Un joueur est dans une gare

**When** Le joueur mine

**Then** Ce qui est miné par le joueur est retiré de la gare

20 Les ressources sont ajoutées au joueur

**Given** Un joueur est dans une gare

**When** Le joueur mine

**Then** Ce qui est miné par le joueur lui est ajouté

21 Autoriser le minage

**Given** Un joueur est dans une gare

**When** Le joueur veut miner

**Then** Le serveur l'autorise

22 arrêt après capacité max atteinte

**Given** Un joueur mine dans une gare

**When** Le train du joueur n'a plus de place

**Then** Le minage est arrêté

23 arrêt si ressources épuisées

**Given** Un joueur mine dans une gare

**When** La gare n'a plus de ressources

**Then** Le minage est arrêté

3 Test récupération de la gare où on est le joueur

✓ To check

*Given* Le joueur est à dans une gare

*When* Le joueur veut savoir où il est

*Then* le serveur lui envoie la gare où il est

---

4 Test récupération des autres gares

✓ To check

*Given* Le joueur est à dans une gare

*When* Le joueur veut choisir un gare où aller

*Then* Le serveur lui envoie les informations sur tous les gares

---

5 Test validation requête de déplacement

✓ To check

*Given* Le joueur peut changer de gare

*When* Demande un changement de gare

*Then* Le serveur valide le changement

---

6 Test refus requête de déplacement

✓ To check

*Given* Le joueur ne peut pas changer de gare

*When* Demande un changement de gare

*Then* Le serveur refuse le changement

---

7 Changement a-t-il été effectué

✓ To check

No description

---

8 Demande d'info durant le trajet

✓ To check

*Given* Le joueur est en déplacement

*When* Le joueur veut savoir où il en est (il vient de se connecter)

*Then* Le serveur lui dit de où à où il va et quand il arrivera

9 Réserver un slot dans une gare

✓ To check

*Given* Le joueur peut changer de gare

*When* Demande un changement de gare

*Then* Le serveur réserve un slot (quai) dans la gare pour le joueur

## Finaliser la GUI

50 changer d'affichage sans recharger la fenêtre

✓ To check

*Given* Le joueur est connecté

*When* Le joueur change de tab

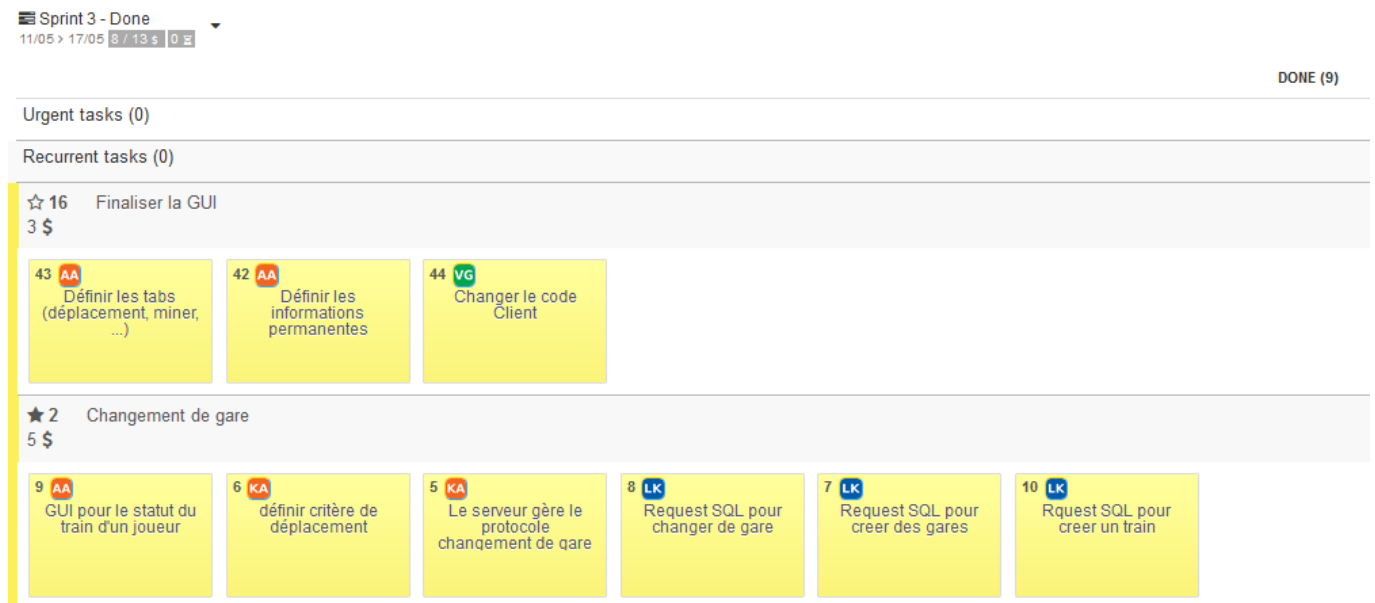
*Then* La fenêtre n'est pas rechargée



## Estimation par tâche [heure/personne] + bilan personnel

Numéro de tâche	Personne assignée	Durée prévue [heure/homme]	Durée effective [heure/homme]	Commentaire
17	Kamil	2	0	Pas effectuée
18	Loyse	1	1	Requête écrite, mais pas encore de tests
19	Loyse	1	1	Requête écrite, mais pas encore de test
20	Vincent	2	0	Pas effectué
41	Vincent	2	0	Pas effectué
5	Kamil	1	3	Gestion du JSON en plus
6	Kamil	2	2	Ajout d'un contrôleur
7	Loyse	1	1	Modification de la DB pour ajouter les gares
8	Loyse	1	1	Passé comme prévu
9	Adrien	2	1	Moins de travail que prévu
10	Loyse	1	1	Passé comme prévu
42	Adrien	2	1	-
43	Adrien	2	1	-
44	Vincent	1	3	Plus de temps que prévu (apprentissage swing)

## Bilan d'itération



## Bilan de terminaison des histoires

## Histoires terminées :

- 16 : Finaliser GUI
- 2 : Changement de gare

## En cours :

- 6 : Miner

## Vélocité du sprint

8 points d'histoire

## Replanification

L'histoire 6 (Miner) est déplacée au sprint suivant. L'histoire 7 (Fabriquer) est maintenue.

[Commentaire général](#)

A cause du travail des autres branches, nous n'avons pas pu terminer toutes les histoires. Nous avons enrichi la structure de la base de données. Nous n'avons pas implémenté suffisamment de tests.

[Autocritique](#)

Même si nous avons eu du temps supplémentaire, le planning était un peu ambitieux. Nous avons chacun pu travailler sur nos parties et avons réussi à les intégrer au projet sans trop de problèmes.

[Sprint 4](#)[Objectifs](#)

Implémenter le minage et la fabrication d'objets

[Stories à faire](#)

6 :Miner , 7 : Fabriquer

[Tasks](#)

VG : Vincent Guidoux, KA : Kamil Amrani, AA : Adrien Allemand, LK : Loyse Krug

★ 6 Miner

5 \$

New task

19 LK 1 \$

Request SQL pour ajouter wagons de minage à un train

41 VG 2 \$

Implémenter class wagon de minage

17 KA 2 \$

Le serveur gère le protocole de minage

20 VG 2 \$

Implémenter regeneration des ressources

18 LK 1 \$

Request SQL pour ajouter des mines aux gares

46 AA 1 \$

Creer GUI minage

★ 7 Fabriquer

8 \$

New task

22 KA 1 \$

Le serveur gère le protocole de fabrication

49 KA 2 \$

Créer une classe recette

48 LK 1 \$

Mettre à jour la base de donnée

21 AA 2 \$

Créer le diagramme du craft des ressources

23 VG 2 \$

ajout de wagon de craft

45 LK 2 \$

Ajouter un contrôleur pour le craft

47 AA 1 \$

Ajouter GUI pour le craft

## Test d'acceptation

## Miner :

19 Les ressources sont retirées de la gare ✓ To check

*Given* Un joueur est dans une gare

*When* Le joueur mine

*Then* Ce qui est miné par le joueur est retiré de la gare

20 Les ressources sont ajoutées au joueur ✓ To check

*Given* Un joueur est dans une gare

*When* Le joueur mine

*Then* Ce qui est miné par le joueur lui est ajouté

21 Autoriser le minage ✓ To check

*Given* Un joueur est dans une gare

*When* Le joueur veut miner

*Then* Le serveur l'autorise

22 arrêt après capacité max atteinte ✓ To check

*Given* Un joueur mine dans une gare

*When* Le train du joueur n'a plus de place

*Then* Le minage est arrêté

23 arrêt si ressources épuisées ✓ To check

*Given* Un joueur mine dans une gare

*When* La gare n'a plus de ressources

*Then* Le minage est arrêté

51 regeneration des ressources ✓ To check

*Given* mines n'est pas pleine

*When* attendre un moment et personne ne mine

*Then* la mine contient plus de ressources

## Fabriquer :

24 Retrait des ressources utilisées ✓ To check

*Given* Le joueur a de quoi fabriquer un objet

*When* Le joueur lance la fabrication d'un objet

*Then* Les ressources sont retiré du joueur

26 Ajout de la ressource créée ✓ To check

*Given* Le joueur à lancé la création d'un objet

*When* La fabrication est terminée

*Then* L'objet est ajouté au joueur

52 Liste des recettes disponibles ✓ To check

*Given* un joueur avec des ressources

*When* il ouvre la table de craft

*Then* il voit toutes les recettes qu'il peut crafter

Numéro de tâche	Personne assignée	Durée prévue [heure/homme]	Durée effective [heure/homme]	Commentaire
17	Kamil	2	1	On a l'habitude de faire évoluer la gestion du protocole par le serveur
18	Loyse	1	1	Mise en place des tests pour valider a mis du temps
19	Loyse	1	1	Mise en place des tests pour valider a mis du temps
20	Vincent	2	2	Les contrôleurs utilisent maintenant des timers
21	Adrien	2	2	Fait deux fois à cause d'un programme inadapté
22	Kamil	1	1	-
23	Vincent	2	0	Pas commencé, remplacé par des tests
41	Vincent	2	4	On n'utilise pas l'héritage pour les wagon, mais on stocke les infos dans WagonStats (nouvelle classe)
45	Loyse	2	1	Commencé mais pas eu le temps de terminer
46	Adrien	1	1	-
47	Adrien	1	1	-
48	Loyse	1	0	Pas eu le temps de commencer la tâche
49	Kamil	2	1	Plus facile que prévu

## Bilan d'itération

Sprint 4 - Done

18/05 > 24/05 5 / 13 \$ 0

Urgent tasks (0)

Recurrent tasks (0)

★ 6 Miner  
5 \$

19 LK Request SQL pour ajouter wagons de minage à un train	46 AA Créer GUI minage	41 VG Implémenter class wagon de minage	18 LK Request SQL pour ajouter des mines aux gares	20 VG Implementer regeneration des ressources	17 KA Le serveur gère le protocole de minage
---	---------------------------	--	---	--	---

★ 7 Fabriquer  
8 \$

49 KA Créer une classe recette	21 AA Créer le diagramme du craft des ressources
-----------------------------------	---

## Bilan de terminaison des histoires

Histoire terminée :

- 6 : Miner

Histoire en cours :

- 7 : fabriquer : faite à 20%

## Vélocité du sprint

5 points d'histoire

*Replanification*

Nous avons dû déplacer l'histoire 7 (Fabriquer) au sprint suivant. L'histoire 5 : Améliorer son train est maintenue pour le sprint 5, l'histoire 2 ajouter une gare, elle, est repoussée au sprint 6.

*Commentaire général*

Dû au rendu de PRO nous n'avons pas pu consacrer autant de temps que voulu au projet. La sérialisation globale dans le projet a été revue.

*Autocritique*

Chacun a travaillé dans son domaine (GUI, protocole, contrôleur, base de données) ce qui nous permet de bien paralléliser le travail.

*Sprint 5**Objectifs*

L'utilisateur doit pouvoir fabriquer des objets et améliorer les wagons de son train.

*Stories à faire*

7 : Fabriquer, 8 : Améliorer son train

*Tasks*

VG : Vincent Guidoux, KA : Kamil Amrani, AA : Adrien Allemand, LK : Loyse Krug

<p>★ 8 Améliorer son train 5 \$</p> <p>New task</p>	<div>53 LK 2 \$ Requêtes SQL</div> <div>52 VG 3 \$ Contrôleur pour les améliorations</div> <div>24 KA 2 \$ Le serveur gère le protocole d'amélioration</div> <div>51 KA 2 \$ Le client gère le protocole d'amélioration</div> <div>50 AA 1 \$ GUI amélioration</div> <div>25 AA 1 \$ Diagramme et cout d'amélioration</div>
<p>★ 7 Fabriquer 8 \$</p> <p>New task</p>	<div>22 KA 1 \$ Le serveur gère le protocole de fabrication</div> <div>23 VG 2 \$ ajout de wagon de craft</div> <div>45 LK 2 \$ Ajouter un contrôleur pour le craft</div> <div>47 AA ? \$ Ajouter GUI pour le craft</div> <div>48 LK 1 \$ Mettre à jour la base de donnée</div>

*Test d'acceptation*

Fabriquer :

## 24 Retrait des ressources utilisées

✓ To check

*Given* Le joueur a de quoi fabriquer un objet*When* Le joueur lance la fabrication d'un objet*Then* Les ressources sont retiré du joueur

## 26 Ajout de la ressource créée

✓ To check

*Given* Le joueur à lancé la création d'un objet*When* La fabrication est terminée*Then* L'objet est ajouté au joueur

## 52 Liste des recettes disponibles

✓ To check

*Given* un joueur avec des ressources*When* il ouvre la table de craft*Then* il voit toutes les recettes qu'il peut crafter

## Améliorer son train :

## 28 Validation d'une amélioration

✓ To check

*Given* le joueur a suffisamment de ressource*When* il demande l'amélioration d'un wagon*Then* le serveur renvoie un message d'acceptation

## 29 Amélioration effectuée

✓ To check

*Given* une amélioration est lancée sur un wagon*When* l'amélioration est terminée*Then* le wagon a augmenté de niveau

## 30 ressources retirées

✓ To check

*Given* le joueur lance une amélioration*When* le serveur l'accepte*Then* les ressources nécessaires sont retirées du joueur

## Estimation par tâche [heure/personne] + bilan personnel

Numéro de tâche	Personne assignée	Durée prévue [heure/homme]	Durée effective [heure/homme]	Commentaire
22	Kamil	1	1	-
23	Vincent	2	1	Plus simple que prévu
24	Kamil	2	2	-
25	Adrien	1	3	Equilibrage des ressources a pris beaucoup de temps
45	Loyse	2	2	-

GEN-2018

Allemand Adrien, Amrani Kamil, Guidoux Vincent, Krug Loyse

47	Adrien	1	1	-
48	Loyse	1	1	-
50	Adrien	1	1	-
51	Kamil	2	2	-
52	Vincent	3	2	On a déjà beaucoup de contrôleurs, on commence à avoir l'habitude
53	Loyse	2	1	Beaucoup de requêtes déjà faites, on commence à avoir l'habitude

## Bilan d'itération

**Sprint 5 - Done**  
25/05 > 31/05 13 / 13 \$ 0

**DONE (11)**

Urgent tasks (0)

Recurrent tasks (0)

★ 8 Améliorer son train  
5 \$

- 50 AA GUI amélioration
- 52 VG Contrôleur pour les améliorations
- 51 KA Le client gère le protocole d'amélioration
- 25 AA Diagramme et cout d'amélioration
- 24 KA Le serveur gère le protocole d'amélioration
- 53 LK Requêtes SQL

★ 7 Fabriquer  
8 \$

- 45 LK Ajouter un contrôleur pour le craft
- 22 KA Le serveur gère le protocole de fabrication
- 47 AA Ajouter GUI pour le craft
- 48 LK Mettre à jour la base de donnée
- 23 VG ajout de wagon de craft

## Bilan de terminaison des histoires

Histoires terminées :

- 7 : Fabriquer
- 8 : Améliorer son train

## Vélocité du sprint

13 points d'histoire

## Replanification

Pas de replanification prévue

## Commentaire général

On a rattrapé le retard. On commence à pouvoir jouer au jeu. Les lambda expression c'est la vie.

## Autocritique

Chacun se sent de plus en plus investi, et les choses avance plus facilement.

## Sprint 6

### Objectifs

Ajout du pan admin au jeu. Un admin doit pouvoir ajouter et supprimer des éléments du serveur

## Stories à faire

9 : Ajout/Suppression des ressources

10 : Bannir un joueur

11 : Ajouter/Supprimer une mine

12 : Ajouter/Supprimer une gare

## Tasks

VG : Vincent Guidoux, KA : Kamil Amrani, AA : Adrien Allemand, LK : Loyse Krug

## Ajout/Suppression des ressources

62 <b>LK</b> 1	26 <b>KA</b> 0.5	63 <b>LK</b> 0.5	59 <b>KA</b> 0.5
Requête SQL d'ajout/suppression de ressources à un	Protocole d'ajout/suppression de ressources à un	Requête SQL d'ajout/suppression de ressources à une	Protocole d'ajout/suppression de ressources à une
70 <b>VG</b> 1	60 <b>AA</b> 1	61 <b>AA</b> 1	72 <b>VG</b> 1
Contrôleur ajout/suppression de ressources à un	GUI d'ajout/suppression de ressources à un	GUI d'ajout/suppression de ressources à une	Contrôleur suppression d'un joueur
69 <b>VG</b> 1			
Contrôleur ajout/suppression de ressources à une			

## Bannir un joueur

65 <b>LK</b> 0.5	64 <b>AA</b> 1	71 <b>KA</b> 0.5
Requête SQL suppression joueur	GUI suppression joueur	Protocole suppression de joueur

## Ajouter/Supprimer une mine

74 <b>VG</b> 1	57 <b>LK</b> 1	73 <b>KA</b> 0.5	58 <b>AA</b> 1
Contrôleur ajout/suppression de mine	Requête SQL ajout/suppression de mines	Protocole ajout/suppression de mines	GUI ajout/suppression mine



## Ajouter/Supprimer une gare



## Test d'acceptation

## Ajout/Suppression des ressources

32 Le joueur voit ses ressources modifiée

*Given* Le joueur a une quantité de ressources

*When* L'admin change la quantité de ressources du joueur

*Then* Le joueur voit ses ressources modifiées

53 Changer la quantité de ressources dans une mine

*Given* Une mine a une quantité de ressources

*When* L'admin change la quantité de ressources dans la gare

*Then* La mine a sa quantité de ressources modifiées

## Bannir un joueur

33 vérifie que le compte du joueur n'existe plus

*Given* Un joueur est dans la base de données

*When* L'admin banni le joueur

*Then* Le joueur n'est plus dans la base de données

## Ajouter/Supprimer une mine

34 Les joueurs voient la gare

*Given* Les joueurs voient qu'une gare est sans mine

*When* L'admin ajoute une mine dans cette gare

*Then* Les joueurs voient qu'une gare a une mine

54 Les joueurs ne voient plus la gare

*Given* Les joueurs voient qu'une gare a une mine

*When* L'admin supprime une mine dans cette gare

*Then* Les joueurs voient qu'une gare est sans mine

## Ajouter/Supprimer une gare

35 le joueur voit la gare

*Given* La gare n'est pas présente dans le jeu

*When* L'admin ajoute la gare

*Then* La gare est dans le jeu

36 gare n'existe plus sur la carte

*Given* La gare est dans le jeu

*When* l'admin supprime la gare

*Then* La gare n'est plus dans le jeu

## Estimation par tâche [heure/personne] + bilan personnel

Numéro de tâche	Personne assignée	Durée prévue [heure/homme]	Durée effective [heure/homme]	Commentaire
60	Adrien	1		
61	Adrien	1		
62	Loyse	1		
63	Loyse	0.5		
26	Kamil	0.5		
59	Kamil	0.5		
69	Vincent	1		
70	Vincent	1		
64	Adrien	1		
65	Loyse	0.5		
71	Kamil	0.5		
72	Vincent	1		
58	Adrien	1		
57	Loyse	1		
73	Kamil	0.5		
74	Vincent	1		
54	Adrien	1		
55	Loyse	1		
68	Kamil	1		
75	Vincent	0.5		
56	Vincent	1		
76	Vincent	0.75		

GEN-2018

Allemand Adrien, Amrani Kamil, Guidoux Vincent, Krug Loyse

Bilan d'itération

*Bilan de terminaison des histoires*

*Vélocité du sprint*

*Replanification*

*Commentaire général*

*Autocritique*

Sprint 7

Objectifs

Stories à faire

Tasks

VG : Vincent Guidoux, KA : Kamil Amrani, AA : Adrien Allemand, LK : Loyse Krug

Test d'acceptation

Estimation par tâche [heure/personne] + bilan personnel

Bilan d'itération

*Bilan de terminaison des histoires*

*Vélocité du sprint*

*Replanification*

*Commentaire général*

*Autocritique*



## 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 avons 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 connaissons 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.