



**AVIGNON**  
UNIVERSITÉ

# Rapport Final

M1 ILSSEN ALT

**Yanis Labrak**

**Miora RABEFANDROANA**

**23 mai 2021**

**Licence Informatique**  
**Ingénierie Logicielle**

**UE** Interface graphique et ergonomie

**Responsables**

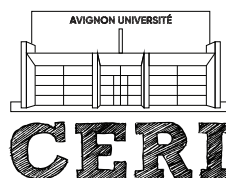
Titouan Parcollet  
Mathias Quillot

**UFR**

**SCIENCES**

**TECHNOLOGIES**

**SANTÉ**



**CENTRE**  
**D'ENSEIGNEMENT**  
**ET DE RECHERCHE**  
**EN INFORMATIQUE**  
[ceri.univ-avignon.fr](http://ceri.univ-avignon.fr)

## Sommaire

<b>Titre</b>	<b>1</b>
<b>Sommaire</b>	<b>2</b>
<b>1 Description du rendu</b>	<b>3</b>
<b>2 Manuel d'utilisation du programme</b>	<b>3</b>
2.1 Commandes	3
<b>3 Interface graphique</b>	<b>4</b>
3.1 Maquette initiale et rendu visuel	4
<b>4 Répartition des tâches</b>	<b>5</b>
4.1 Tâches individuelles	5
4.2 Issues	5
4.3 Répartition des tâches individuelles	6
4.4 Déroulement du semestre	6
4.5 Tâches communes	7
4.5.1 F01 – Conception de l'UX	7
4.5.2 F02 / F03 – Conception de l'UI	7
4.5.3 F17 – Rendre le programme responsif	7
4.6 Yanis Labrak	8
4.6.1 F04 – Lancer la partie.	8
4.6.2 F05 – Mettre en pause la partie.	8
4.6.3 F08 – Créer une carte et placer des noeuds..	9
4.6.4 F10 – Ajouter une carte par csv.	9
4.6.5 F14 – Faire varier la vitesse d'exécution du modèle.	9
4.6.6 F15 – Afficher des informations permettant de guider le joueur.	10
4.6.7 T01 – Parallélisation du programme.	10
4.6.8 F16 – Identifier clairement le chemin le plus court.	10
4.7 Miora RABEFANDROANA	11
4.7.1 F06 – Stopper la partie.	11
4.7.2 F07 – Recommencer la partie.	11
4.7.3 F09 – Charger une carte.	12
4.7.4 F11 – Supprimer une carte.	12
4.7.5 F12 – Modifier une carte.	13
4.7.6 F13 – Sauvegarder une carte.	13
4.7.7 C01 / C02 – Internationalisation / Conventions Java	14
<b>5 Problèmes rencontrés</b>	<b>15</b>

## 1 Description du rendu

Les consignes du projet nous ont été délivré le Lundi 8 Mars 2021.

Les fonctionnalités doivent être fractionné en tâches individuelles et répartie de façons équilibrés entre les 2 membres du groupes.

Le sera rendu aux enseignants le dernier jour du jalon par le biais d'un dépôt d'une archive contenant le code source, un jar ainsi qu'un rapport détaillant les différents aspects inhérents à la conduite du projet (difficultés, solutions, organisation, etc.). Ces fichiers sont à déposer par binôme sur le site de l'E-UAPV [NOM1\_PRENOM1\_NOM2\_PRENOM2\_ALT/CLA.tar].

## 2 Manuel d'utilisation du programme

### 2.1 Commandes

Afin de vous faciliter l'usage du logiciel, nous avons prit la décision de vous remettre un fichier `jar` vous permettant de tester facilement sans devoir compiler le projet.

```
1 tar -zxvf jar.tar
2 java -jar APP.jar
```

### 3 Interface graphique

#### 3.1 Maquette initiale et rendu visuel

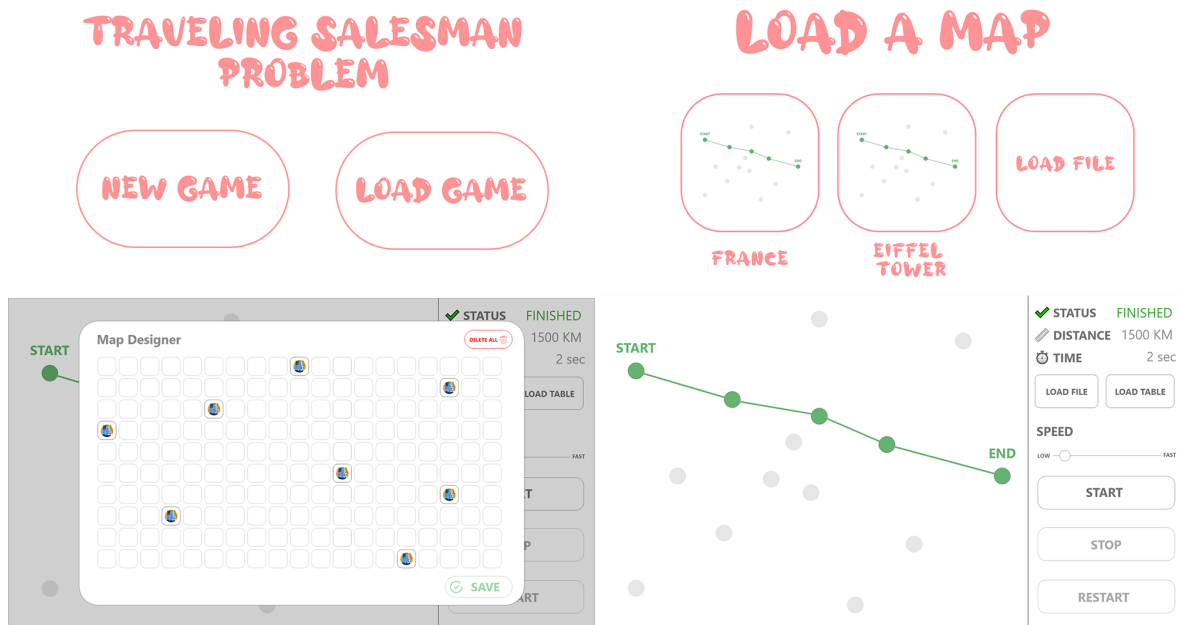
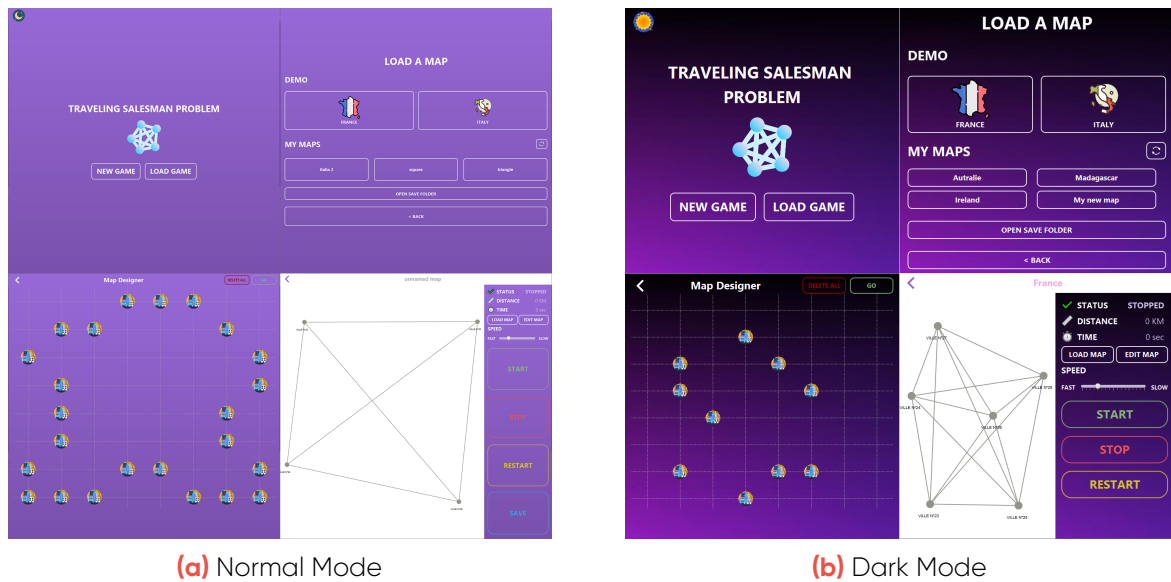


Figure 1. Maquette de l'interface graphique.



(a) Normal Mode

(b) Dark Mode

Figure 2. Rendu final de l'interface graphique.

## 4 Répartition des tâches

Nous avons décidé pour le projet d'utiliser la méthodologie de gestion de projet *SCRUM*<sup>1</sup>. Cella nous a permis d'assigner dynamiquement durant cette période difficile les tâches en fonction des envies et des compétences de chacun.

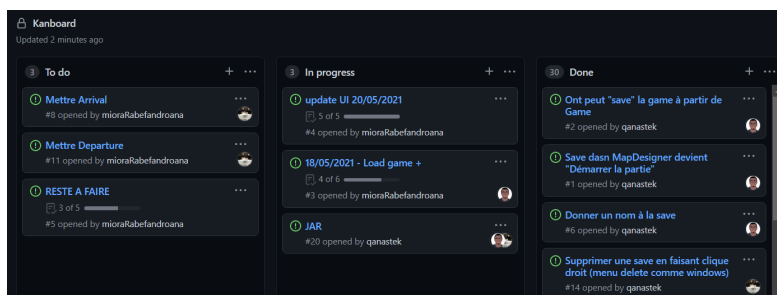
Nous avons orienté notre projet autour des sections *Issues* et *Projects* de GitHub<sup>2</sup>.

### 4.1 Tâches individuelles

Code	Nom
F01	Conception de l'UX
F02	Choix colorimétrie et charte graphique
F03	Conception de l'interface graphique
F04	Lancer une partie
F05	Mettre en pause la partie
F06	Stopper la partie
F07	Recommencer la partie
F08	Créer une carte et placer des noeuds.
F09	Charger une carte.
F10	Ajouter une carte par csv.
F11	Supprimer une carte.
F12	Modifier une carte.
F13	Sauvegarder une carte.
F14	Faire varier la vitesse d'exécution du modèle.
F15	Afficher des informations permettant de guider le joueur
F16	Identifier clairement le chemin le plus court.
F17	Rendre le programme responsif
T01	Parallélisation du programme.
C01	Internationalisation du code
C02	Respect des conventions Java

**Table 1.** Décomposition du travail en tâches individuelles.

### 4.2 Issues



**Figure 3.** Issues.

1. Méthodologie SCRUM : [https://fr.wikipedia.org/wiki/Scrum\\_\(développement\)](https://fr.wikipedia.org/wiki/Scrum_(développement))

2. Plateforme collaborative GitHub : <https://Github.com/>

### 4.3 Répartition des tâches individuelles

Tâche	Étudiant	Temps	État
F01	Miora RABEFANDROANA + Yanis Labrak	1H30	✓
F02	Miora RABEFANDROANA + Yanis Labrak	0H45	✓
F03	Miora RABEFANDROANA + Yanis Labrak	4H30	✓
F04	Yanis Labrak	1H00	✓
F05	Yanis Labrak	0H30	✓
F06	Miora RABEFANDROANA	0H45	✓
F07	Miora RABEFANDROANA	0H10	✓
F08	Yanis Labrak	1H45	✓
F09	Miora RABEFANDROANA	0H40	✓
F10	Yanis Labrak	0H15	✓
F11	Miora RABEFANDROANA	0H15	✓
F12	Miora RABEFANDROANA	1H20	✓
F13	Miora RABEFANDROANA	0H40	✓
F14	Yanis Labrak	0H15	✓
F15	Yanis Labrak	1H00	✓
F16	Yanis Labrak	1H30	✓
F17	Miora RABEFANDROANA + Yanis Labrak	2H00	✓
T01	Yanis Labrak	0H30	✓
C01	Miora RABEFANDROANA	0H30	✓
C02	Miora RABEFANDROANA	0H20	✓
TOTAL		20H15	

**Table 2.** Répartition des tâches individuelles.

### 4.4 Déroulement du semestre

Séance	Date	Objet
TP1	08/03/2021	Présentation du projet et Conception de l'UX
TP2	08/04/2021	Implémentation du graphe et TSP
TP3	15/04/2021	Implémentation de l'assistant à la création de carte
TP4	16/04/2021	Implémentation Sauvegarde, Suppression, ...
TP5	03/05/2021	Gestion des états et transitions
TP6	05/05/2021	Affichage chemin final
TP7	10/05/2021	Finalisation
R1	23/05/2021	Rendu final (sources + rapport + jar)

**Table 3.** Répartition des séances

## 4.5 Tâches communes

### 4.5.1 F01 - Conception de l'UX

Nous avons commencer par une phase de recherche dans laquelle nous avons travailler avec des wireframes dans le but de concevoir une expérience utilisateur qui ne ferait pas biaiser par un mauvais choix de couleur par exemple. De plus, le prototypage en wireframe est un procédé rapide et permet à faible coût de faire des modifications.

Nous avons conçu notre interface graphique dans le but d'être très facilement accessible par des enfants en bas âge et donc de ne pas être trop demandante d'un point de vue cognitif.

Les actions doit être rapidement accessible et compréhensible d'un seul coup d'oeil. Pour ce faire, nous avons utiliser des icônes pour représenter les différentes fonctionnalités et états du logiciel.

### 4.5.2 F02 / F03 - Conception de l'UI

Par la suite, nous avons traiter la partie UI qui comprend donc la charte graphique (couleurs, formes, thème générale) mais aussi les différents icônes que nous allions utiliser pour représenté les fonctionnalités.

Notre choix de charte graphique s'est porté sur une charte enfantine avec une multitude de couleurs sans pour autant être trop flashy et donc injouable durant la nuit. Le choix des couleurs et des formes retranscrire au mieux l'expérience utilisateur que nous cherchions à transmettre à un publique de bas âge.

Nous avons décidé d'ajouter un thème alternatif en plus du principal afin que les utilisateurs puissent adapter l'interface à leurs utilisation et de ce fait avoir une meilleur expérience.

L'interface comporte donc 2 thème : le **Normal** et le **Dark**. Ce changement de mode peut se faire en se rendant sur l'interface d'accueil ou ce trouve un bouton ayant l'apparence d'un(e) **lune/soleil** en fonction de l'état actuel.



**Figure 4.** Changement de thème

En ce qui concernent les images, nous avons pencher notre dévolu sur des images relativement explicite et cela même pour un enfant grâce à des codes couleurs qui représente bien les émotions que nous souhaitions transmettre. Nous pouvons ici cité par exemple les images représentant les différents pays ou villes.

### 4.5.3 F17 - Rendre le programme responsif

Dû à la diversité des résolutions actuellement sur le marché (1080p, 1440p, 4K ...), il est essentiel de fournir un logiciel qui soit capable de s'adapter à cette épouvantail de possibilité.

Nous avons cependant pour des raisons esthétiques lier à l'assistant de création des cartes souhaiter contraindre le comportement du logiciel à adopter un ratio d'affichage au format 1:1. Il nous permet d'avoir un assistant de forme carré facilement compréhensible par l'utilisateur.

## 4.6 Yanis Labrak

### 4.6.1 F04 - Lancer la partie.

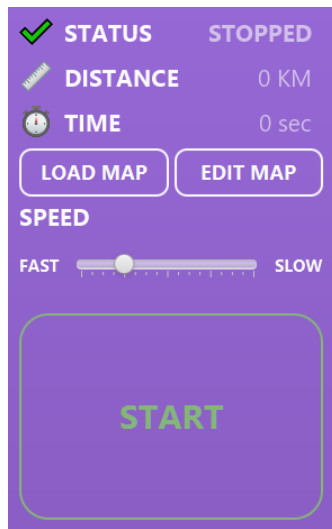


Figure 5. Bouton de démarrage

Lors du clique sur le bouton de démarrage de la partie, nous commençons par la mise à jours les éléments de l'interface graphique tels que le **Statut** ou les boutons.

Nous poursuivons ensuite sur la création d'un **Task** et d'un **Thread** dans lequel ce trouve une instance du modèle **TSPModel\_PtiDeb** qui prend en argument la vue **JavaFX** sous forme d'un **Observable**.

Avant d'initier le lancement du **Thread**, nous donnons les sommets qui représente le **Graph** au modèle.

Le démarrage du calcul est ensuite initier par la méthode **run**.

### 4.6.2 F05 - Mettre en pause la partie.

Afin d'initier l'arrêt temporaire du modèle, l'utilisateur doit d'abord avoir mit en marche l'algorithme puis il pourra cliquer sur le bouton **Pause** qui va tout simplement faire appel à la méthode **setPause(True)** de l'instance du modèle.

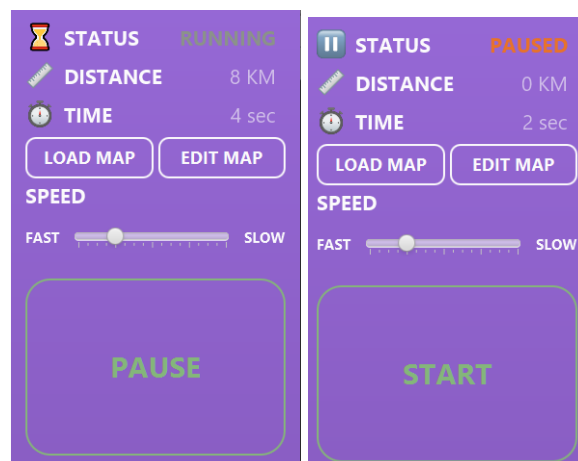


Figure 6. Pause



#### 4.6.3 F08 - Créer une carte et placer des noeuds..

La partie la plus critique pour laquelle nous voulons réduire au plus l'effort cognitif pour rendre l'utilisation du programme par un publique très jeune possible était celle de la création de la carte (graph).

Pour ce faire nous avons étudié plusieurs propositions d'UX (liste, tableau, graph avec popup) pouvant ce prêter à notre cas d'usage et nous sommes arrivés à une conclusion. Aucune d'entre elle n'est assez simple pour un enfant.

Nous avons donc fait de nouveaux essais et sommes arrivés à trouver un design plus simple à appréhender et ne nécessitant pas de tutoriel particulier.

Celui consiste à avoir un quadrillage orthonormé sur laquelle les enfants ont la possibilité de déposer des villes qui représenteront les noeuds du graph.

Une fois que l'utilisateur a fini. Il peut cliquer sur le bouton **Go** qui vérifie la présence d'au moins une ville et transfère les données vers la prochaine vue.

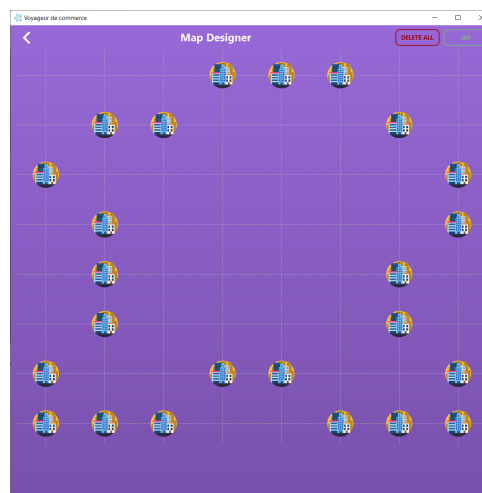


Figure 7. Assistant de création

#### 4.6.4 F10 - Ajouter une carte par csv.

Nous avons pensé qu'il serait sympa d'ajouter la possibilité aux utilisateurs de partager leurs cartes avec leurs amis.

Nous avons donc implémenté un bouton similaire au bouton "Open Pack Folder" de Minecraft permettant grâce à la fonction `openExplorerMapsFolder` d'ouvrir le dossier où se situe l'ensemble des cartes (sur windows : Documents/CERI\_TSP/Data/Maps/).

L'enfant n'a plus qu'à envoyer le **CSV** portant le nom de sa carte à ses amis qui eux les déposeront au même endroit en passant eux aussi par le même bouton.

#### 4.6.5 F14 - Faire varier la vitesse d'exécution du modèle.

Nous avons décidé de permettre à l'utilisateur de régler par lui-même le délai entre chaque déplacement grâce à la présence d'un **Slider**. Celui-ci lui permet de soit, réduire le délai en déplaçant le curseur vers **SLOW**, soit l'accélérer en déplaçant vers **FAST**.

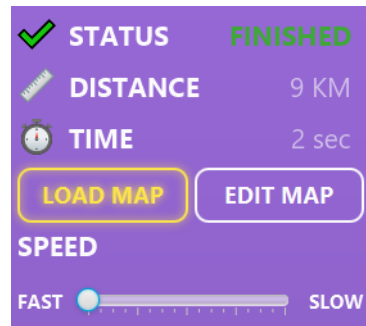


Figure 8. Slider temps de pause

#### 4.6.6 F15 - Afficher des informations permettant de guider le joueur.

Dans le but encore une fois de réduire l'effort cognitif et simplifier l'usage du programme aux plus jeunes. Nous avons décider d'appuyer au plus que possible nos intentions par la présence d'images / icônes et popup en cas d'erreurs.

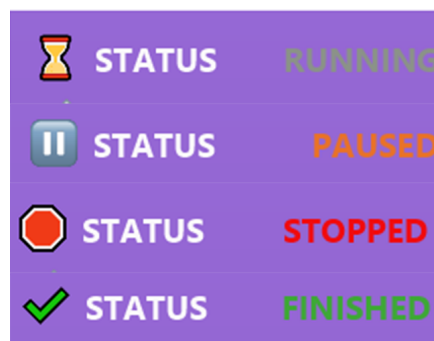


Figure 9. L'ensemble des statut

#### 4.6.7 T01 - Parallélisation du programme.

En vue de ne pas bloquer le **Thread** courant par le calcul du plus court chemin, nous devons déplacer la logique présente autour du modèle **TSPModel\_PtiDeb** dans une **Task** qui sera ensuite envelopper dans un **Thread**. Nous agissons de la sorte afin d'être en parfait accord avec les conventions de multi-threading de **JavaFX** et éviter tout risque de crash ou bug graphique.

#### 4.6.8 F16 - Identifier clairement le chemin le plus court.

Durant le déroulement du modèle, nous gardons une **ArrayList** à jours avec l'état actuelle du chemin. A chaque ajout et suppression émit de la part du modèle à l'**Observer** nous maintenons la liste à jours en répercutant les changements.

Une fois le statut **TSPModel\_PtiDeb.ActionType.Finish** de lever par le modèle. Nous entraînons une actualisation complète de l'affichage afin d'y afficher en bleu le parcours du chemin final qui vient d'être préalablement calculer.

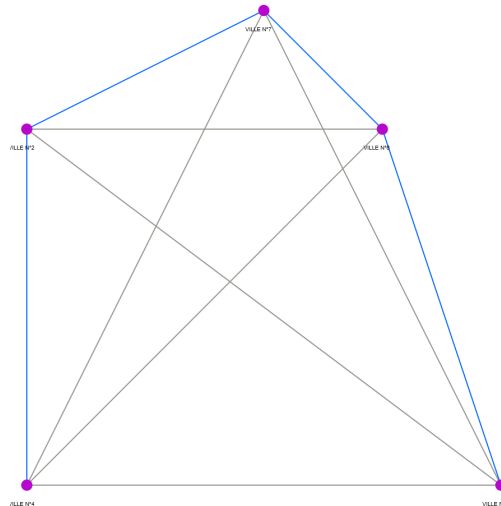


Figure 10. Chemin Final

## 4.7 Miora RABEFANDROANA

### 4.7.1 F06 - Stopper la partie.

À tout moment, au cours d'une partie, l'utilisateur a la possibilité de stopper l'exécution de l'algorithme. Pour cela il suffit de cliquer sur le bouton **STOP**.<sup>(11)</sup>

Cela entraînera le détachement de l'**Observer**, l'arrêt total du **Thread** et la destruction de l'instance du modèle.

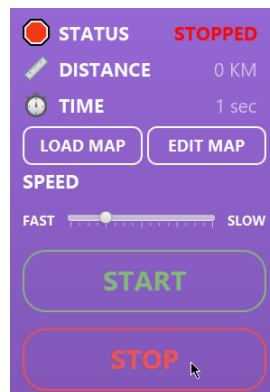


Figure 11. Stopper la partie

### 4.7.2 F07 - Recommencer la partie.

L'utilisateur a la possibilité de recommencer une partie via le bouton **RESTART**. A n'importe quel moment d'une partie, cette action aura comme effet la réinitialisation de l'exécution de l'algorithme tout comme celle du graphe.<sup>(12)</sup>



Figure 12. Recommencer une partie

#### 4.7.3 F09 - Charger une carte.

Le chargement d'une carte se fait via l'interface "Load a Map". Accessible via l'accueil avec le bouton "LOAD GAME" ou encore le bouton "LOAD MAP" de l'interface de jeu. Cette interface offre la possibilité de charger l'une des cartes de démonstration fournies avec le programme (France, Italie, ...) mais aussi de charger les cartes sauvegardées préalablement par l'utilisateur via la fonctionnalité **SAVE**.<sup>(13)</sup>

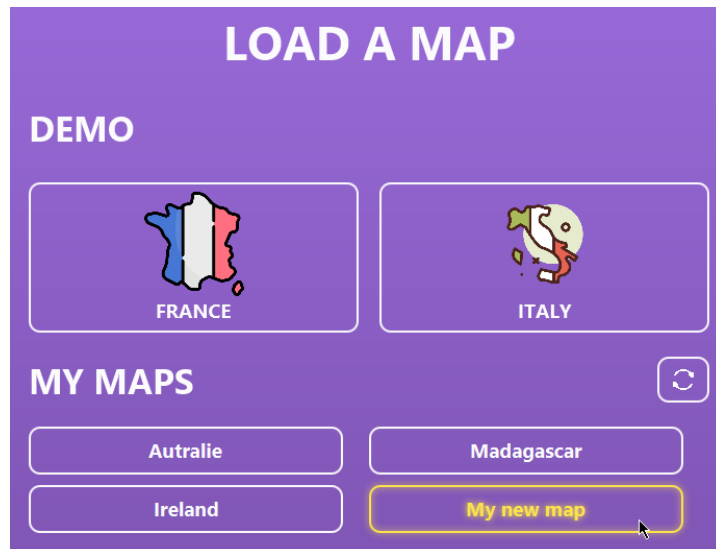


Figure 13. Chargement d'une carte

#### 4.7.4 F11 - Supprimer une carte.

Dans l'interface "Load a Map", les cartes enregistrées peuvent être supprimées facilement par le biais d'un clic droit sur la carte que nous souhaitons supprimer puis cliquer sur "DELETE".<sup>(14)</sup>

NB : Les cartes de démonstration ne peuvent pas être supprimées.

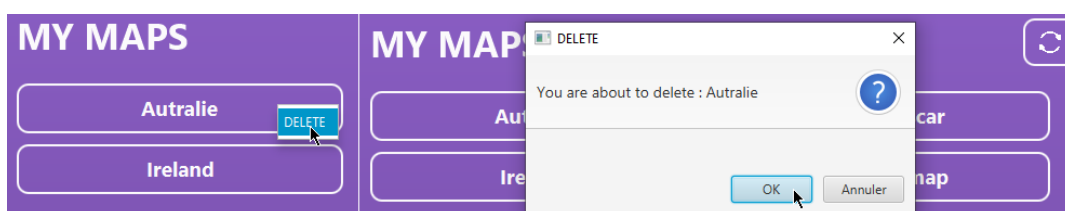


Figure 14. Suppression d'une carte

#### 4.7.5 F12 - Modifier une carte.

Chacune des cartes enregistrer peuvent être modifier. Pour cela, il faut charger la carte depuis l'interface **Load a Map** puis cliquer sur le bouton **"Edit Map"**. Une fois la carte chargée, nous accédons à l'interface d'édition **"Map Designer"**.<sup>(15)</sup>

Pour que les modifications soit pris en compte, l'utilisateur doit les enregistrer grâce au bouton **SAVE**.

NB : Les cartes de démonstration ne sont pas modifiable

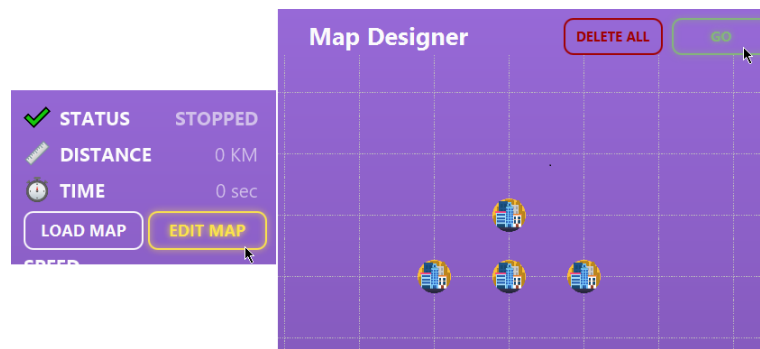


Figure 15. Modification d'une carte

#### 4.7.6 F13 - Sauvegarder une carte.

L'enregistrement d'une carte se fait via le bouton **"SAVE"**. Ce dernier permet à la fois de sauvegarder une nouvelle carte et d'appliquer les changements à une carte préalablement chargée.<sup>(16)</sup>

Toute carte à sauvegarder doit avoir un nom. Celui peut être spécifié dans le champ titre correspondant en haut de l'interface.

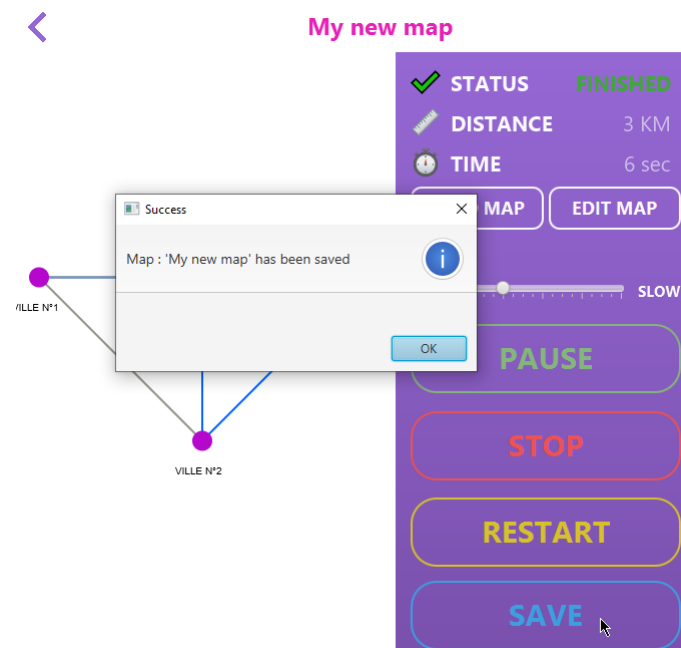


Figure 16. Sauvegarde d'une carte

#### **4.7.7 C01 / C02 - Internationalisation / Conventions Java**

Nous avons remis de l'ordre dans notre code de sorte à respecter scrupuleusement les conventions Java. J'ai pu remarquer par exemple qu'une partie des variables static ne respectent pas les conventions. Cela m'a permis en même temps de m'apercevoir que certaines parties du code n'étaient plus utilisées et que nous pouvions donc les supprimer. J'ai aussi remarqué que le code était dans certains cas redondant et j'en ai profité pour factoriser tout cela.

## 5 Problèmes rencontrés

Pour ce projet nous n'avons pas rencontré de problématique particulière hormis quelques difficultés avec le modèle que nous avons reçu et plus particulièrement concernant la mise en pause qui était difficile à mettre en oeuvre.

Nous avons appris qu'il pouvait parfois avoir des différences entre l'interface conçu lors de la phase de réflexion sur l'UX et la phase d'implémentation. Dans notre cas nous avons conçus une interface qui ce porter bien à un usage par un publique adulte mais qui ne peut malheureusement pas être transposer à un usage par un public enfant car peu attractif et n'étant pas forcément très explicite.

Nous avons aussi rencontrer des difficultés avec l'affichage harmonieux de l'écran de création de la carte. Celui-ci n'était pas adapter à un affichage en 16:9 et nous avons donc prit l'initiative de forcer un affichage au format 1:1 dans toute l'application.

Au cours de ce rendu, nous avons porter un soin tout particulier à l'expérience utilisateur. Nous voulions faire de notre possible pour réduire le nombre de clic et de pop-up qui s'avèrent au final trop intuitives et cassent la dynamique de navigation.

Nous avons aussi travailler grandement sur l'effort cognitif nécessaire à l'utilisation de l'application. Nous voulions que le jeu soit abordable pour un public très jeune. Pour ce faire, nous avons abondamment utiliser des termes court et des images d'illustration.