



CS-108  
Printemps 2021

---

## Rapport de l'étape 12

### tCHu

---

Mallory HENRIET — 311258  
Louis HERVÉ — 312937

19 juillet 2021



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Améliorations de l'expérience utilisateur</b>	<b>2</b>
2.1	Description . . . . .	2
2.2	Mise en œuvre Java . . . . .	2
<b>3</b>	<b>Barre de menus</b>	<b>3</b>
3.1	Description . . . . .	3
3.2	Mise en œuvre Java . . . . .	4
<b>4</b>	<b>Mode sombre</b>	<b>5</b>
4.1	Description . . . . .	5
4.2	Mise en œuvre Java . . . . .	5
<b>5</b>	<b>Serveur</b>	<b>6</b>
5.1	Description . . . . .	6
5.2	Mise en œuvre Java . . . . .	6

# 1 Introduction

Pour commencer, merci de télécharger le fichier **resources.zip**, et utiliser le dossier **resources** qu'elle contient en lieu et place de celui fourni par défaut.

Cela est nécessaire car nous avons ajouté plusieurs fichiers de ressources. Nous avons aussi restructuré le dossier en créant des dossiers distincts. L'archive finale de rendu dépassait les 2 Mo maximum requis, et n'était donc pas acceptée par le serveur.

De plus, il faut ajouter à la configuration le module **javafx.media**, afin que les sons puissent être joués.

## 2 Améliorations de l'expérience utilisateur

### 2.1 Description

Pour améliorer l'utilisation du jeu en cours de partie, nous avons fait quelques ajouts.

Premièrement, les jauges des pioches de cartes et billets sont désormais de couleur verte lorsque leur pourcentage dépasse les 45%, de couleur orange entre 25% et 45%, et de couleur rouge lorsqu'elles arrivent à moins de 25%.

Deuxièmement, nous avons ajouté la possibilité de jouer des sons lorsque le joueur effectue différentes actions :

- piocher ou tirer une carte ou des billets, ainsi que choisir des billets, lance un bruitage de carte, ou plusieurs ;
- prendre possession d'une route lance un bruit de construction, pour référer à la pose de wagons sur des rails ;
- prendre possession d'une route qui fait passer son nombre de wagons à un nombre inférieur ou égale à 2 lance un bruit de klaxon de train.

Tous les différents sons peuvent être activés ou désactivés depuis la barre de menus, voir ci-dessous.

Troisièmement, nous avons ajouté une information dans la liste des billets indiquant s'il a été rempli par l'utilisateur.

### 2.2 Mise en œuvre Java

Pour mettre en œuvre le changement de couleur des jauges, nous avons adapté le fichier `decks.css` en changeant la couleur de la propriété `-fx-fill` de classe `.foreground`. Puis nous avons créé les nouvelles classes `Button.gauged .is-medium` et `Button.gauged .is-low` qui prennent pour `-fx-fill` du orange et du rouge respectivement.

Enfin, grâce à des `Listener`, nous avons pu changer les feuilles de styles des boutons dans la méthode `bindPercentage()` de la classe `DecksViewCreator` en observant la propriété passée en argument.

Pour les sons, nous avons créé une fonction `playSound()` dans notre classe `GuiUtils` (anciennement `ConstantsGui` avant le bonus). Elle instancie des `MediaPlayer` et les lance à l'aide de leur méthode `play()`. La méthode est appelée grâce à des `Listener` dans les cas voulus. Par exemple, lorsque que l'on appelle `onDrawCard()` dans `GraphicalPlayer`.

## 3 Barre de menus

### 3.1 Description

Nous avons ajouté à la partie supérieure de l'écran une barre de menus (voir Figure 3.1).



FIGURE 3.1 – Barre de menus

Cette barre est composée de 3 menus :

- un menu tCHu ;
- un menu d'affichage ;
- un menu sons.

Le premier menu (voir Figure 3.2) permet d'afficher les règles du jeu, faire siffler le train du joueur, ou quitter la partie. L'icône de train dans ce menu a été réalisée par nous-mêmes dans un logiciel de dessin vectoriel.



FIGURE 3.2 – Menu tCHu

Le deuxième menu (voir Figure 3.3) permet de passer le jeu en mode clair ou sombre, inverser la vue des statistiques et des pioches, ou bien encore régler la taille de la carte. Nous avons voulu permettre aux joueurs de changer cette taille, car nous avons rencontré des problèmes lorsque nous avons joué sur de petits écrans : par exemple, on ne pouvait pas prendre les routes en Autriche, qui se situaient alors sous la vue des pioches.

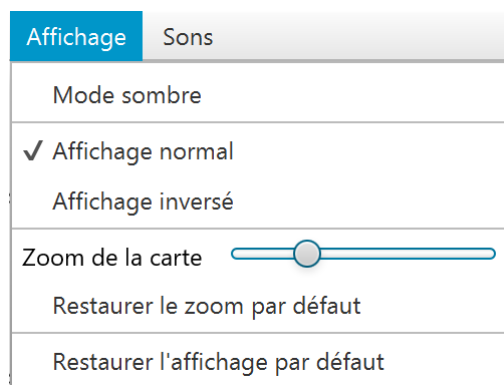


FIGURE 3.3 – Menu Affichage

Le troisième menu (voir Figure 3.4) permet d'activer ou désactiver les différents sons susceptibles d'être joués durant la partie.

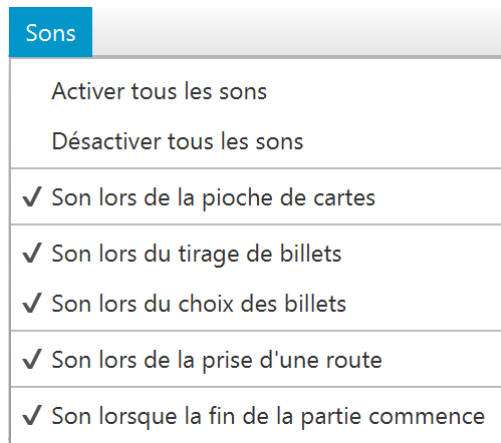


FIGURE 3.4 – Menu Sons

## 3.2 Mise en œuvre Java

Pour mettre en œuvre cette barre, nous avons créé une instance de **MenuBar** que nous avons placée en haut du **BorderPane** principal. Le mode sombre sera détaillé plus bas. Pour le mode inversé, nous avons simplement utilisé un **Listener** qui ajuste le contenu des vues associées du **BorderPane**. Pour la gestion des sons, nous avons créé une propriété booléenne **BooleanProperty** pour chaque son, indiquant si l'utilisateur désire les activer ou non.

## 4 Mode sombre

### 4.1 Description

Désormais, le jeu se lance par défaut en mode “sombre” (voir Figure 4.1). Dans ce mode, l’interface a des couleurs plus foncées. Les couleurs de certaines cartes, du contour des routes, des fenêtres de choix ou encore de la carte (voir Figure 4.2) en elle-même ont donc été modifiées.

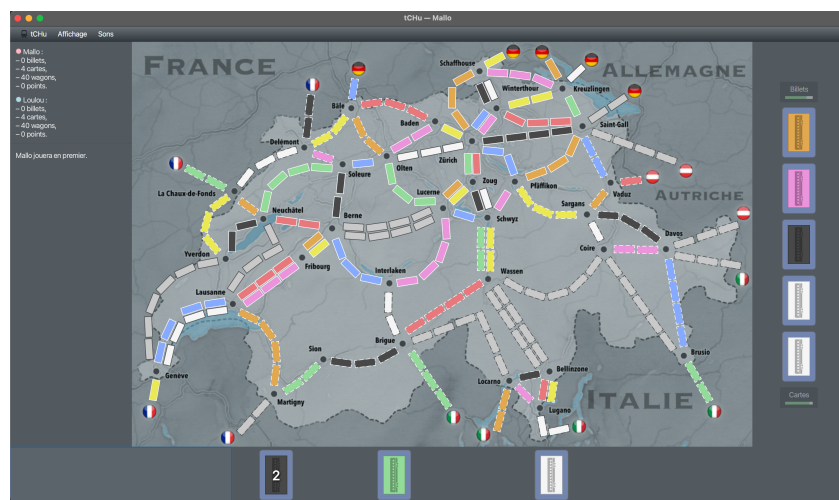


FIGURE 4.1 – Le jeu en mode sombre



(a) Ici en mode clair



(b) Ici en mode sombre

FIGURE 4.2 – Les deux types de cartes

### 4.2 Mise en œuvre Java

Pour implémenter le mode sombre, nous avons d’abord créé des nouvelles feuilles de style ressemblant aux anciennes mais avec le suffixe *-dark*, comme par exemple `map-dark.css`. Elles changent les valeurs des couleurs des objets qui nous intéressent. Grâce à la propriété booléenne `darkModeP`, nous changeons les feuilles de style des différents `Node`.

La carte a été recolorée pour l’occasion.



## 5 Serveur

### 5.1 Description

Nous avons ajouté une classe `ServerMainCli`, et créé une archive JAR qui peut être lancée depuis un serveur privé virtuel, chez OVH dans notre cas (`vps.dousherve.fr`).

### 5.2 Mise en œuvre Java

Il a suffi d'adapter `ServerMain` afin de prendre différents arguments, et de créer les deux joueurs en tant que `RemotePlayerProxy`.