

# Rapport extensions tCHu / Ennassih – Barghorn

**Afin de lancer nos extensions, il suffit de lancer la classe GameLauncher du package /src.**

Ce qui est en gras constitue le coeur de nos extensions, celles qui ont nécessité le plus de travail.

---

## Améliorations côté utilisateur :

Nos extensions s'inscrivent dans une volonté de produire un jeu utilisable par une personne extérieure au projet. Pour ce faire, nous avons implémenté un **lanceur de jeu**, qui propose différents modes de jeux, un choix du nom des deux joueurs et de **la couleur de leurs wagons** et facilitant la mise en place d'un jeu en ligne.

Un fichier java exécutable peut être facilement généré à partir de notre projet pour lancer le jeu sur n'importe quel ordinateur.

Le joueur a le choix entre les modes de jeux suivants :

- Entraînement : le jeu lance une partie de tCHu contre l'ordinateur qui effectue des actions de manière aléatoire. Ce mode permet de s'exercer et d'approfondir ses connaissances de jeu. La partie se déroule de manière classique.
- Jeu en local : le jeu lance les fenêtres des deux joueurs et offre une partie en local, c'est-à-dire sur le même ordinateur. La partie se déroule de manière classique, chaque joueur doit effectuer ses propres actions à tour de rôle sur sa fenêtre de jeu.
- Jeu en ligne : le jeu laisse le choix au joueur d'être le client ou le serveur. Dans le cas du client, le joueur peut remplir l'adresse ip et le port du serveur. Il faut bien veiller à lancer le serveur avant le client, puis la partie se lance à distance.
- **Tutoriel** : ce mode de jeu permet de découvrir toutes les règles et explications pour pouvoir jouer à tCHu. A travers une partie guidée, le jeu donne plusieurs instructions et exige des actions de la part du joueur pour poursuivre le tutoriel. Le tutoriel bloque et attend donc des actions spécifiques de la part du joueur.

---

## Implémentation du Tutoriel :

L'implémentation du tutoriel s'effectue autour d'une méthode Tutorial.play(...) qui prend en argument le nom du joueur et est appelée par le lanceur de jeu lorsque le joueur souhaite démarrer le tutoriel. Cette méthode lance une partie guidée, dont les tours et actions du joueurs sont prédéfinis au cours de son lancement. La difficulté principale résidait dans le blocage des actions du joueur et notamment l'attente des réponses attendues (le tutoriel ne continue pas tant que le joueur n'effectue pas l'action demandée) ainsi que dans la gestion du PlayerState du joueur (vider sa main de jeu ou lui ajouter des cartes données).

## **./ressources**

- > train-driver.png : image associée aux instructions du tutoriel
- > tutorial-box.css : contient les propriétés liées aux instructions et boutons du tutoriel
- > tutorial.txt : ensemble des instructions du tutoriel

## **./src/ch/epfl/tchu/gui**

- > class TutorialGraphicalPlayer
  - => similaire à GraphicalPlayer, ajoute le tutoriel sur le fenêtre
- > class TutorialViewCreator
  - => similaire aux autres ViewCreator, crée la vue du tutoriel contenant des instructions, deux boutons et une image
- > interface ActionHandlers
  - > functional interface TutorialHandler
    - onButtonClick(boolean leaves) :
      - => gestionnaire d'action lié aux boutons "Continuer" (leaves = false) et "Quitter le tutoriel" (leaves = true) du tutoriel

## **./src/ch/epfl/tchu/tutorial**

- > class Tutorial
  - static play(String playerName) : similaire à la méthode play(...) de Game, lance le tutoriel
- > class TutorialText
  - => offre une lecture ligne par ligne des instructions de tutorial.txt

=====

## **Implémentation du GameLauncher :**

La classe Launcher crée de la même manière qu'un GraphicalPlayer une interface Graphique. Toutes les Nodes nécessaires à la création de ce lanceur de jeu se trouvent dans cette seule et unique classe. Cette classe ne nécessite pas de fichier .css : toutes les valeurs JavaFX y sont stockées. Il a été particulièrement difficile de mettre en place tous les nouveaux objets graphiques que nous avons pu découvrir ainsi que de créer un interface visuellement attrayante. L'autre difficulté résidait dans la communication entre les différentes propriétés et parties de cette interface (par exemple bloquer certaines possibilités en fonction des choix du joueur).

## **./src**

- > class GameLauncher
  - => lancer une interface JavaFX représentant un menu de jeu