

## TP4 - Projet Java

### *Interface graphique*

Jérôme Buisine  
Florian Leprêtre

[jerome.buisine@univ-littoral.fr](mailto:jerome.buisine@univ-littoral.fr)  
[florian.lepretre@univ-littoral.fr](mailto:florian.lepretre@univ-littoral.fr)

19 avril 2021

Durée : 3 heures

---

La finalité de la suite des TPs proposés, est de concevoir un système permettant à des utilisateurs de jouer à des jeux cartes en réseau. Les premiers jeux de cartes développés seront le jeu de Bataille classique ainsi qu'une version simplifiée du jeu de Poker.

## 1 Travail

L'objectif de ce dernier TP est d'ajouter à notre framework de jeu une interface graphique pour le client. Ce dernier TP sera considéré comme Bonus. L'idée de ce TP est tout simplement de proposer une interface agréable et de vous faire plaisir simplement !

### 1.1 Objectifs

En exploitant la librairie [JavaFX](#), qui a pu être étudiée précédemment, il vous faudra mettre en place une interface graphique pour le jeu de la Bataille dans un premier temps pour 4 joueurs. Quelques idées

vous sera communiqué sur l'attendu, mais cela bien sûr, ne reste que propositionnel. Ce sera à vous seul de juger de la meilleure interface.

### 1.2 Environnement de travail

L'environnement de travail sera le même que les TPs précédents :

- 1. IntelliJ/Idea (version Ultimate) comme environnement de développement (ou Eclipse en fonction de votre préférence) ;
- 2. Le langage Java avec la version 15 de [JDK](#) ;
- 3. [Git](#) comme système de versionning du projet ;
- 4. [Gitlab](#) comme serveur d'hébergement de votre projet **Git** ;
- 5. Utilisation de la librairie [JavaFX](#) au sein de votre projet Java.

### 1.3 Récupération de composants graphiques

Sur la branche TP3 du projet, où la correction n'est pas encore proposée, il vous faudra récupérer les ressources graphiques proposées.

Les ressources sont les suivantes :

- components/cards, est un dossier qui vous propose un ensemble de représentations graphique de cartes. Ces éléments seront affichés au Joueur ;
- CardGameInterface.txt, qui prend en considération ces éléments graphique dans la conception du jeu.

## 2 Interface graphique

### 2.1 Configuration

Pour pouvoir intégrer JavaFX dans votre projet, il vous faudra un JDK 15. Le lien est disponible dans la section environnement 1.2 du sujet.

Pour configurer votre projet avec JavaFX, voici les différentes étapes à suivre :

- Télécharger la librairie JavaFX : <https://gluonhq.com/products/javafx/>
- Suivre la procédure de configuration de la librairie pour IntelliJ à partir de l'étape 2. **Set JDK 15** : <https://openjfx.io/openjfx-docs/#install-javafx>
- Tester la librairie à partir des exemples disponibles : <https://github.com/jjenkov/javafx-examples>

### 2.2 Jeu de bataille

Voici une proposition très minimaliste de l'interface homme machine (IHM) envisageable pour le jeu de la Bataille :

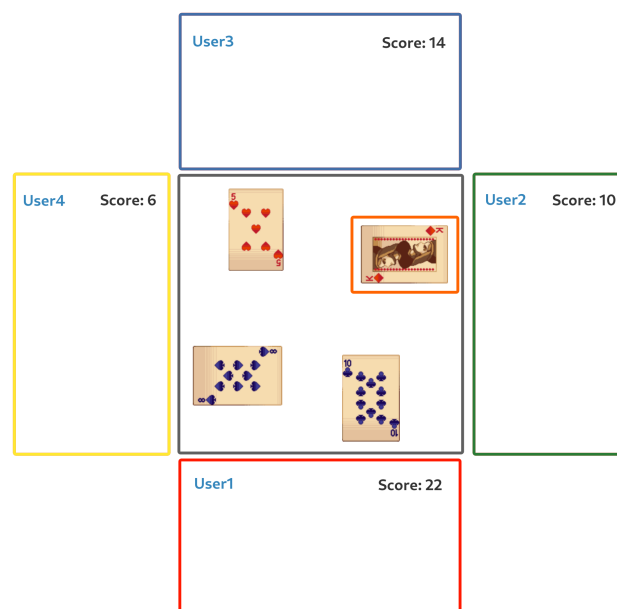


FIGURE 1 – Interface graphique d'une manche jouée pour 4 joueurs.

Dont l'étape de répartition des cartes en fin de manche, peut être représenté de la manière la suivante :

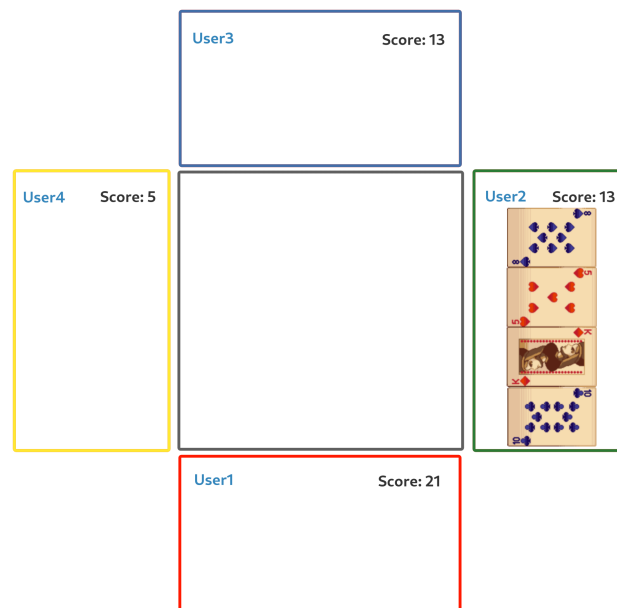


FIGURE 2 – Repartition des cartes aux gagnants. Ici, c'est le joueur 2, qui gagne l'ensemble des cartes. Les scores de chaque joueur est mis à jour.

### Travail :

À partir des ressources disponibles et de la librairie JavaFX, réaliser une interface pour le jeu de Bataille. Pour permettre une bonne visualisation du jeu, un temps de pause du jeu côté Serveur sera privilégié. Ce qui permettra une bonne visualisation de l'évolution du jeu.

### Quelques notes :

- On se limitera ici à une interface de 4 joueurs ;
- En chargeant le fichier de jeu proposé, vous pouvez stocker dans une variable `String` le chemin vers le composant. Cette variable sera un nouvel attribut de la classe `Card` ;
- Pour charger le contenu graphique d'une carte face cachée, étant donné que cette ressource sera identique pour toutes les cartes cachées, vous pouvez utiliser une variable de classe (`static final`).