

# Dojo - Le jeu de Poker

## (extension)

Octobre 2022

V0.4

Philippe Collet

## Validation PREALABLE de la main de Poker

Pour réaliser ces nouvelles fonctionnalités, un chargé de TD doit d'abord valider que votre version de la main de Poker est bien conçue et bien testée :

- Le code est organisé autour des bonnes abstractions (exemple : le résultat d'une comparaison ne peut pas être une String ou un effet de bord qui écrit sur la sortie standard le résultat).
- L'ensemble est testé unitairement et couvre bien la comparaison des 2 mains par rapport aux règles, ainsi que la robustesse du programme vis-à-vis des entrées saisies.

## Description du problème – Texas Hold'em sans mise

Réaliser un jeu à 4 joueurs de Poker en ligne de commandes selon les règles du Texas Hold'em :

- Le moteur de jeu interactif tire maintenant au sort une main de 2 cartes pour chaque joueur et l'affiche avec la même syntaxe que pour la main de Poker (les cartes sont censées être cachées mais l'idée est de les voir sur la console).
- Le moteur affiche progressivement (on demande à passer à l'étape suivante sur la ligne de commande) les 5 cartes ouvertes (3 cartes au flop, 1 carte au turn, 1 carte à la rivière), visibles de tous.
- A chaque étape, le moteur affiche pour chaque joueur la meilleure main potentielle entre ses cartes et les cartes ouvertes.
- A la fin, le moteur affiche le gagnant avec une explication de comment il gagne (avec quelles cartes de son jeu et quelles cartes ouvertes).

Il n'est pas demandé de système de mises, ni d'interactions. En revanche, cette version du jeu prépare une suite complète, (que vous ne réaliserez pas), donc il ne faut surtout pas que cette version se contente d'une série de grosses méthodes dans la classe principale. Il faut aussi construire les bonnes abstractions (gérant par exemple les tours de jeu.) pour préparer la suite.

Références : [https://fr.wikipedia.org/wiki/Texas\\_hold%27em](https://fr.wikipedia.org/wiki/Texas_hold%27em)

## Travail à réaliser

Ensuite il vous faudra les découper (verticalement) en tâches, les répartir et avoir un code testé pour chaque nouvelle fonctionnalité réalisée.