

Votre mission, si vous l'acceptez (et même si vous ne l'acceptez pas), est de réaliser un jeu de bataille en utilisant une structure de donnée adaptée, et avec une interface graphique. Vous aurez pour cela 6 heures (les trois séances jusqu'aux vacances de Noël).

Plus précisément, on attend au minimum :

- Une classe `Carte` ayant
  - deux attributs `valeur` et `couleur`
  - une méthode `compare` qui prend en paramètre une autre carte `other`, et qui renvoie
    - 1 si la valeur de `self` est supérieure à la valeur de `other` ;
    - 1 si la valeur de `other` est supérieure à la valeur de `self` ;
    - 0 si les deux valeurs sont les mêmes.
- Une classe `Jeux` ayant
  - Un attribut `paquet` contenant toutes les cartes du jeu. Le nombre de cartes (32 ou 52) sera un paramètre du constructeur. Ce paquet sera créé puis mélangé dans le constructeur.
  - Une méthode `distribue` qui renvoie deux `Files` contenant la moitié des cartes du paquet chacune. On utilisera la classe `File` du cours (disponible sur Moodle).

À chaque tour de jeu, chaque joueur joue la carte supérieure de son paquet. Le joueur ayant la carte de plus grande valeur récupère les deux cartes et les remet sous son paquet.

En cas d'égalité, chaque joueur ajoute une carte à l'envers, puis une autre carte à l'endroit, et on compare à nouveau les valeurs. En cas de nouvelle égalité, on réitère le processus. Le gagnant remporte l'ensemble des cartes jouées.

Si un joueur ne peut plus jouer car il n'a plus de carte, il a perdu la partie.

Concernant l'interface graphique, elle sera réalisée en `Tkinter`, et permettra de faire une partie contre l'ordinateur. Il n'est pas nécessaire d'afficher les cartes à un format graphique (même si ce n'est bien sûr pas interdit) : on pourra se contenter de textes de la forme « 3 de carreau » ou « roi de pique ».

Le chef de chaque équipe se chargera de la répartition des tâches, en accord avec les membres de son équipe évidemment.