

# TP : Jouer à la bataille

---

## 1 Présentation du jeu

Le jeu de cartes de la bataille oppose 2 joueurs. Chaque joueur possède un tas de cartes qu'il ne voit pas. Il n'a pas le droit de les mélanger et ne peut accéder qu'à la carte située au-dessus de son tas. À chaque tour, les joueurs retournent la carte au-dessus du tas et le joueur qui a la carte la plus forte remporte la manche. Les deux cartes ramassées sont placées en-dessous du tas de cartes du joueur vainqueur.

En cas d'égalité, il y a bataille. Les 2 joueurs doivent retirer chacun 1 carte de leur tas et la poser sur la carte au tapis et retirer à nouveau une carte pour remporter la manche. Les batailles peuvent s'enchaîner!

Le premier joueur qui n'a plus de cartes a perdu la bataille.

## 2 Le paquet de cartes

Une carte est représentée par un tuple (valeur, couleur).

- Les valeurs des cartes sont le 1 pour l'as, le 2 pour le deux, etc; le valet est représenté par le nombre 11, la dame par le 12 et le roi par le 13.
- Les couleurs sont les chaînes de caractères "coeur", "carreau", "trèfle" et "pique".

Un paquet de cartes est une liste contenant 32 ou 52 cartes.

- 1) Écrire la fonction **creer\_paquet(n=32)** qui prend en paramètre le nombre **n** de cartes et renvoie la liste contenant toutes les cartes d'un jeu. Le paquet de cartes sera affecté à la variable **paquet**.
- 2) La méthode **shuffle(liste)** du module **random** permet de mélanger aléatoirement les valeurs de la liste. Attention, c'est bien la liste qui est mélangée, pas une copie!  
Créer une fonction **melanger(paquet)** pour mélanger les cartes.
- 3) Les tas de cartes de chaque joueur sont de type **file**. Créer une fonction **distribuer(paquet)** qui va enfiler les cartes du paquet dans 2 files **tasA** et **tasB** alternativement. Les deux tas seront renvoyés par la fonction.

## 3 Le jeu de la bataille

Le jeu se déroule en trois phases :

- 1) On tire (défile) une carte **a** du **tasA** et une carte **b** du **tasB**;
- 2) On compare les 2 cartes :
  - Si la carte **a** est plus forte que la carte **b**, alors on met les 2 cartes sous le **tasA**;
  - sinon, si la carte **b** est plus forte que la carte **a**, alors on met les 2 cartes sous le **tasB**;
  - sinon (les cartes sont aussi fortes) il y a bataille.
- 3) En cas de bataille, on enfile les cartes du tapis et une autre carte de chaque paquet dans une file temporaire (tapis). Ensuite on recommence avec 2 nouvelles cartes et la plus forte remporte le tout.

On donne l'algorithme du jeu :

```
On crée une file vide tapis (cartes en cas de bataille)
Tant que les tas A et B ne sont pas vides :
    on défile le tasA (a)
    on défile le tasB (b)
    si  $a > b$  :
        on enfile a et b dans le tasA
        on récupère tout le tapis dans le tasA (suite à bataille)
    sinon si  $a < b$  :
        on enfile a et b dans le tasB
        on récupère le tapis dans le tasB (suite à bataille)
    sinon : (bataille)
        on enfile a et b dans tapis
        on défile tasA et on enfile tapis
        on défile tasB et on enfile tapis
Fin de la boucle tant que (un des tas est vide)
si paquet A vide :
    afficher "A a perdu, B gagne"
sinon :
    afficher "B a perdu, A gagne"
```

Écrire une fonction **jouer()** programmant l'algorithme ci-dessus.

## 4 Programme principal

Voici l'algorithme du programme principal :

```
Créer une file vide tasA
Créer une file vide tasB
Créer un paquet de cartes paquet
Mélanger le paquet
Distribuer les cartes aux joueurs A et B
Lancer le jeu avec la fonction jouer().
```

- 1) Écrire en python le programme principal et l'exécuter.
- 2) Le programme n'affiche rien. Ajouter des affichages pour :
  - voir les cartes posées par les joueurs A et B ;
  - voir qui remporte la manche ;
  - voir lorsqu'il y a bataille et les cartes dans le tas ;
  - voir le nombre de cartes de chaque joueur après chaque manche ?