

# Sujet du problème Mastermind

Aurélie Lemaitre

## 1 Présentation du sujet

Le Mastermind est un jeu qui se joue à deux joueurs : l'un des joueurs fabrique une combinaison secrète de pions de couleurs pour le second joueur, qui doit essayer de retrouver la combinaison. Pour cela, le second joueur effectue plusieurs tentatives de combinaison. A chaque tentative, le premier joueur indique la validité de la réponse avec un pion noir pour chaque pion de couleur correcte bien placée, et un pion blanc pour chaque pion de couleur correcte mais mal placée.

On souhaite réaliser une implémentation qui respecte les contraintes suivantes :

- le joueur qui fait deviner la combinaison secrète sera joué par l'ordinateur.  
Le joueur qui devine sera joué par un humain.
- on fabriquera des combinaisons de 4 pions de couleurs différentes, parmi 6 couleurs représentées par leur initiale.
- on souhaite obtenir les traces d'exécution suivantes :

```
# Les couleurs disponibles sont : ('B', 'R', 'J', 'V', 'O', 'M')
# Entrer la valeur de la case 0 O
# Entrer la valeur de la case 1 M
# Entrer la valeur de la case 2 R
# Entrer la valeur de la case 3 J
# Tentative 1 | ['O', 'M', 'R', 'J'] | ['Blanc', 'Noir']
# Entrer la valeur de la case 0 B
# Entrer la valeur de la case 1 J
# Entrer la valeur de la case 2 R
# Entrer la valeur de la case 3 V
# Tentative 2 | ['B', 'J', 'R', 'V'] | ['Blanc', 'Blanc', 'Noir', 'Noir']
# Entrer la valeur de la case 0 V
# Entrer la valeur de la case 1 J
# Entrer la valeur de la case 2 R
# Entrer la valeur de la case 3 B
# Tentative 3 | ['V', 'J', 'R', 'B'] | ['Noir', 'Noir', 'Noir', 'Noir']
# Bravo, vous avez réussi en 3 essais
```

## 2 Représentation des données

On représentera la combinaison mystère, ainsi que les tentatives du joueur sous la forme d'une liste de caractères (les initiales des couleurs).

## 3 Codage des fonctions

1. Ecrire une fonction `genereCombinaison` qui prend en entrée une liste de couleurs disponibles et qui renvoie une liste contenant les 4 couleurs de la combinaison mystère.

Remarque : la fonction `random.sample(lst,nb)` génère une liste en piochant aléatoirement `nb` éléments dans la liste `lst`.

2. Ecrire une fonction `entreeValeurs` qui ne prend pas de paramètre, qui demande à l'utilisateur de saisir consécutivement les 4 valeurs de sa combinaison, et qui renvoie la liste contenant les 4 couleurs jouées par l'utilisateur.
3. Ecrire une fonction `calculeResultat` qui prend en paramètre la liste contenant la combinaison mystère et la liste contenant la proposition du joueur. Cette fonction doit comparer ces deux listes pour générer le résultat : pions noirs et pions blancs produits. La fonction renverra une liste résultat contenant de 0 à 4 valeurs {"Blanc"/"Noir"}. Note : on s'assurera de trier la liste pour regrouper les occurrences de "Blanc" et les occurrences de "Noir".
4. Ecrire une fonction `jeuTermine` qui prend en paramètre une liste résultat (contenant des valeurs "Noir" / "Blanc") et qui renvoie vrai si le jeu est terminé, faux sinon.
5. Ecrire une fonction `afficheResultat` qui prend en entrée un entier numéro de tentative, une liste tentative de l'utilisateur et une liste résultat et qui réalise un affichage tel que présenté dans les traces d'exécution :

```
# Tentative 2 | ['B', 'J', 'R', 'V'] | ['Blanc', 'Blanc', 'Noir', 'Noir']
```

6. Ecrire une fonction `masterMind` qui ne prend pas de paramètre. C'est la fonction principale qui sera appelée pour jouer au jeu. Cette fonction doit initialiser la combinaison, afficher les couleurs disponibles, puis faire jouer l'utilisateur tant qu'il n'a pas trouvé la bonne solution, tout en comptant son nombre d'essais. A la fin du jeu, elle affichera le nombre de coups joués. Cette fonction ne doit rien renvoyer.