

TP 3

Le but de ce TP est de réaliser une version texte du jeu Mastermind. Voici une description des règles du jeu : <https://fr.wikipedia.org/wiki/Mastermind>. Nous avons choisi de coder dans le programme `mastermind.py` la version avec une combinaison secrète à 5 pions. Le joueur dispose de 12 tentatives pour découvrir la solution.

Exercice 1

En reprenant, ce que nous avons vu en cours, réaliser la fonction `creer_combinaison ()` qui retourne une liste de chaînes de caractères (nom des couleurs) qui représente la combinaison secrète à 5 pions à partir des six couleurs à disposition dans le jeu. Cette combinaison est générée de manière aléatoire.

Exercice 2

Réaliser la fonction `lire_proposition ()` qui lit une proposition de combinaison saisie au clavier par le joueur et retourne la liste de chaînes de caractères contenant cette proposition.

Exercice 3

Réaliser la fonction `tester_proposition (proposition, combinaison_secrete)` qui compare une proposition de combinaison avec la combinaison secrète et qui retourne une paire contenant le nombre de couleurs bien placées (pions noirs dans le jeu original) et le nombre de couleurs mal situées (pions blancs).

Exercice 4

Réaliser fonction `mastermind ()` qui contient la boucle du jeu et propose à l'utilisateur de saisir une nouvelle combinaison tant que celui-ci n'a pas découvert la bonne solution ou atteint le nombre limite de tentatives. Si le joueur a gagné, félicitez le et dites-lui combien d'essais il lui a fallu pour gagner, sinon dites lui quelle était la combinaison à deviner.

Exercice 5

On voudrait qu'il soit possible d'enchaîner plusieurs parties. En dehors de la fonction, on veut que le programme demande au joueur s'il souhaite faire une nouvelle partie. Si celui-ci répond non, on quitte le jeu, sinon on appelle la fonction qui lance le Mastermind (i.e. la fonction `mastermind ()`). Modifier `mastermind.py` pour répondre à ce souhait.

Exercice 6

À la sortie du programme, on voudrait afficher un résumé des parties quand le jeu est quitté : pour chaque partie jouée, on veut afficher la combinaison à deviner et combien d'essais (en cas de défaite, nous inscrirons 13) ont été nécessaires pour trouver la solution. En utilisant une liste (un élément par partie) de dictionnaires, implémenter le comportement souhaité dans `mastermind.py`.