
Mini-Projet

Exercice 1 (Master Mind)

On se propose ici de réaliser un jeu du type « Master Mind ». Le but du jeu est de deviner une combinaison de pions de couleur choisie par l'adversaire en un minimum de tentative. Le nombre de pions (4 ou 5) et de couleurs (6 à 8), de même que la possibilité d'utiliser plusieurs fois la même couleur et le nombre de tentatives autorisées (10 ou 12), varient selon les versions et le niveau de jeu. Ces éléments seront ici paramétrables.

À chaque tentative, on donne le nombre de pions ayant la bonne couleur mais étant à la mauvaise position et le nombre de pions de la bonne couleur à la bonne position. Selon le niveau de jeu, on peut soit indiquer de quels pions il s'agit, soit ne donner que leur nombre respectif.

Question 1 (Structure) : Créez les différentes classes métier. Vous pourrez notamment créer des classes pour les pions, pour une combinaison de pions (tentative et secret à découvrir), pour représenter le résultat d'une tentative (selon le mode choisi), pour le plateau, ...

Question 2 (Affichage) : Définir une méthode d'affichage du plateau dans un terminal. Vous pourrez utiliser judicieusement les symboles unicode ainsi que les codes d'échappement ANSI pour les couleurs.

Question 3 (Validation) : Définir une méthode sur le secret permettant de tester la validité d'une tentative, et retournant un objet représentant le résultat.

Question 4 (Programme principal) : Écrire un programme exécutable qui permet de choisir le nombre de tentatives, de pions, de couleurs, le type d'indications, puis de jouer. Le programme affichera l'état courant du plateau, puis lira la combinaison de couleurs (représentées par un nombre) de la tentative suivante, tant que la combinaison n'est pas découverte et qu'il reste des tentatives. La combinaison à deviner sera définie de manière aléatoire.

Question 5 (Mode multi-joueur) : Ajoutez la possibilité de jouer à plusieurs, à tour de rôle. Le système comptabilise les points de chaque personne sous la forme du

nombre de tentatives restantes. Un nombre de parties paramétrable sera joué avant de terminer le jeu et afficher les scores totaux.

Question 6 (Sauvegarde et restauration) : Créer deux méthodes symétriques `save(Path)` et `load(Path)`, permettant d'enregistrer l'état courant de du jeu dans un fichier texte d'une part, et de recharger un état précédemment enregistré d'autre part.

Question 7 (Bonus 1) : Implémentez un joueur automatique, contre qui il sera possible de jouer.

Question 8 (Bonus 2) : Implémentez une interface mode texte (TUI), en utilisant par exemple la bibliothèque *lanterna*, ou même une interface graphique avec les éléments de la bibliothèque standard Swing par exemple.