

---

# Mini-Projet

## Exercice 1 (Puissance 4)

On se propose ici de réaliser une version du jeu « puissance 4 ». Ce jeu se joue sur une grille de six lignes et sept colonnes. Le but est d'aligner quatre jetons de la même couleur. Chaque joueur joue à tour de rôle, en ajoutant un jeton dans la colonne de son choix. Le jeton se positionne sur la case libre la plus basse.

La grille sera représentée par une matrice doublement chaînée en 4-connexité, c'est-à-dire que chaque cellule contiendra une référence vers ses 4 cellules voisines, ainsi que la couleur de l'éventuel jeton présent. On maintiendra également un index sous la forme d'un tableau contenant la référence à la prochaine case libre de chaque colonne. Tous ces éléments seront encapsulés dans une classe unique.

**Question 1 (Structure) :** Créer les différentes classes impliquées dans la structure décrite précédemment. Vous ajouterez les différentes méthodes jugées utiles. Créer une méthode d'initialisation de la grille.

On pourra utiliser une énumération pour les directions et une `EnumMap`<sup>1</sup> pour stocker les voisins.

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

**Question 3 (Vérification) :** Ajouter aux cases une méthode `check` vérifiant si elle est dans un alignement de 4 jetons de la même couleur qu'elle-même. Vous pourrez utiliser judicieusement la récursivité.

**Question 4 (Programme principal) :** Écrire un programme exécutable qui permet à deux joueurs ayant donné leur nom et choisi leur couleur de jouer alternativement. Le programme affichera l'état courant de la grille, puis lira le numéro de la colonne où le joueur courant veut positionner son jeton, tant qu'il n'y a pas un gagnant ou que la grille n'est pas remplie.

---

1. <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/EnumMap.html#>

*Question 5* (Sauvegarde et restauration) : Créer deux méthodes symétriques `save(Path)` et `load(Path)`, permettant d'enregistrer l'état courant de la grille dans un fichier texte d'une part, et de recharger un état précédemment enregistré d'autre part.

*Question 6* (Bonus) : Implémentez une intelligence artificielle de jeux, afin de pouvoir jouer contre l'ordinateur. Vous pouvez utiliser des stratégies naïves au coup par coup. Vous pouvez aussi essayer de construire l'arbre des possibilités afin de choisir le meilleur coup possible à chaque tour.