

---

# Mini-Projet

## Exercice 1 (Démineur)

On se propose ici de réaliser une version du jeu « démineur ».

Le jeu se compose d'une grille, dont les cases peuvent contenir des mines. Le but est d'identifier les cases minées et celles libres. Pour cela, lorsqu'une case vide est découverte, le nombre de cases 8-connexes contenant des mines est indiquée (voir figure 1).

2	M	1	
M	2	1	
2	2		
M	1		

FIGURE 1 – Exemple de grille complètement découverte

Si ce nombre est 0, toutes les cases vides adjacentes sont automatiquement découvertes. Il est possible de marquer une case comme minée sans avoir à la découvrir. Si une case minée est découverte, la partie est perdue. Les mines non découvertes, et les éventuelles erreurs de marquage sont alors affichées.

La figure 2 page suivante montre l'état de la grille de la figure 1 après avoir découvert la case en bas à droite, et marqué la mine de la première ligne.

La grille sera représentée par une matrice doublement chaînée en 8-connexité, c'est-à-dire que chaque case contiendra une référence vers ses 8 cases voisines. Cette structure est illustrée figure 4 page suivante. Une case maintient aussi la présence éventuelle d'une mine, ainsi que son état : marquée, cachée ou découverte. On ne peut pas découvrir une case marquée, ni marquer une case découverte.

?	M	1	
?	2	1	
?	2		
?	1		

FIGURE 2 – Exemple de grille en cours de partie

?	?	?	?	?	?	?	?	?	?
1	2	2	1	1	?	?	?	?	?
1	1		1	3	?	?	?	?	?
M	1		1	M	M	M	5	?	?
1	1		1	2	3	3	M	M	3
		1	1	1		1	2	2	1
1	2	2	M	1			1	1	1
M	4	M	3	1			1	M	1
M	4	M	2				1	1	1
1	2	1	1						

FIGURE 3 – Exemple de grille plus complexe

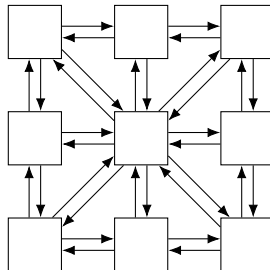


FIGURE 4 – Illustration des liens entre les cases d'une grille

**Question 1 (Structure)** : Créer les différentes classes impliquées dans la structure décrite précédemment. Vous ajouterez les différentes méthodes jugées utiles. Veillez notamment à ce qu'une case maintienne la structure de liens avec ses voisines en cas d'ajout ou changement. 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 unicodes ainsi que les codes d'échappement ANSI pour les couleurs.

**Question 3 (Vérification)** : Ajouter aux cases une méthode `uncover` déclenchant la découverte de la cases, et éventuellement de toutes les cases vides adjacentes. Vous pourrez utiliser judicieusement la récursivité.

**Question 4 (Programme principal)** : Écrire un programme exécutable qui permet de choisir la dimension de la grille et le pourcentage de cases minées, puis de jouer. Le programme affichera l'état courant de la grille, puis lira le numéro de la case que le joueur veut soit marquer, soit découvrir, tant que toutes les cases libres ne sont pas découvertes, et qu'une case minée n'est pas découverte.

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

---

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