

Rapport de projet : Démineur

Tajeddine Slimani et Ibrahim Doucouré

Introduction :

Ce document présente le travail réalisé dans le cadre de la SAÉ 2.1 qui consiste à faire un jeu de Démineur. Le Démineur est un jeu dont le but est de révéler l'intégralité des cases d'une grille sans jamais faire exploser les mines qui y sont cachées, en s'aidant des indices de chiffres découverts au fur et à mesure.

Ce projet a été développé en langage Java, en utilisant uniquement l'API officielle fournie

Ce rapport détaille les fonctionnalités, l'architecture, ainsi que les algorithmes mis en place pour faire fonctionner le jeu.

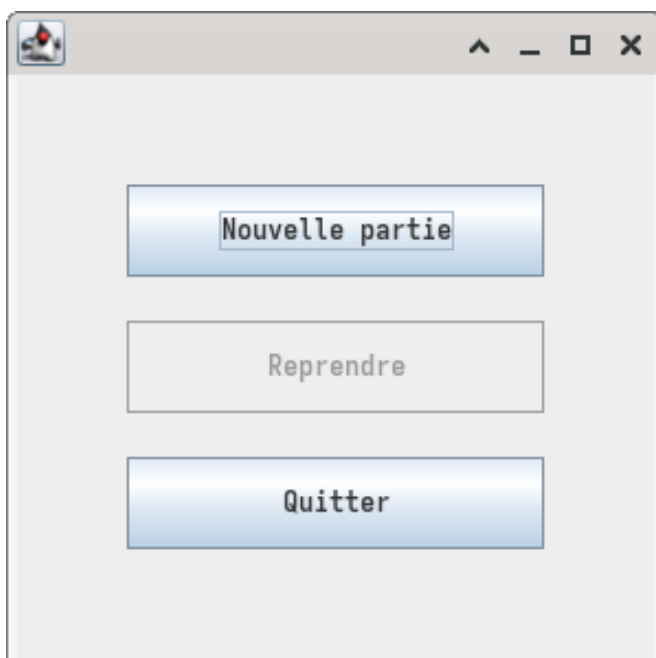
Description des fonctionnalités :

1. Le Menu Principal :

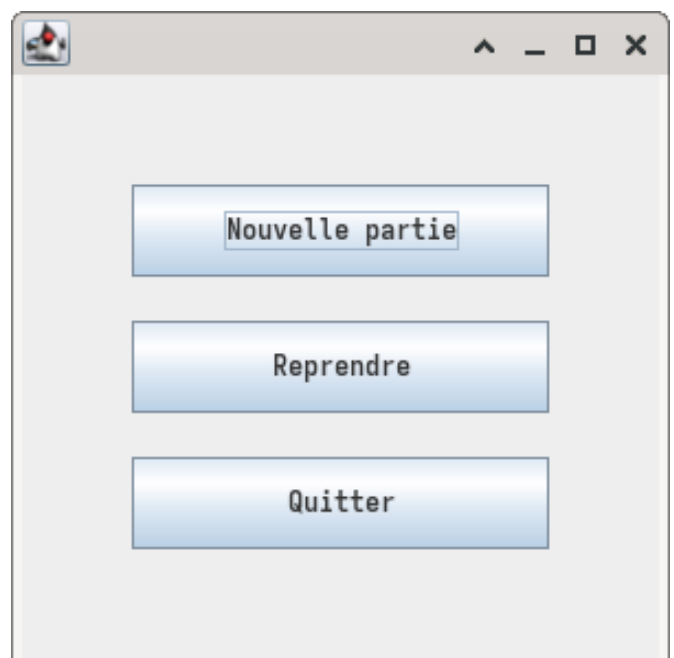
Au lancement du programme, un menu qui lui propose trois option apparaît :

- **Nouvelle partie** : Ouvre la fenêtre de configuration de la partie.
- **Reprendre** : Permet de charger la dernière partie sauvegardée. Ce bouton est grisé et inclicable si aucun fichier de sauvegarde n'est enregistré sur la machine.
- **Quitter** : Ferme l'application.

Sans Sauvegarde



Avec Sauvegarde

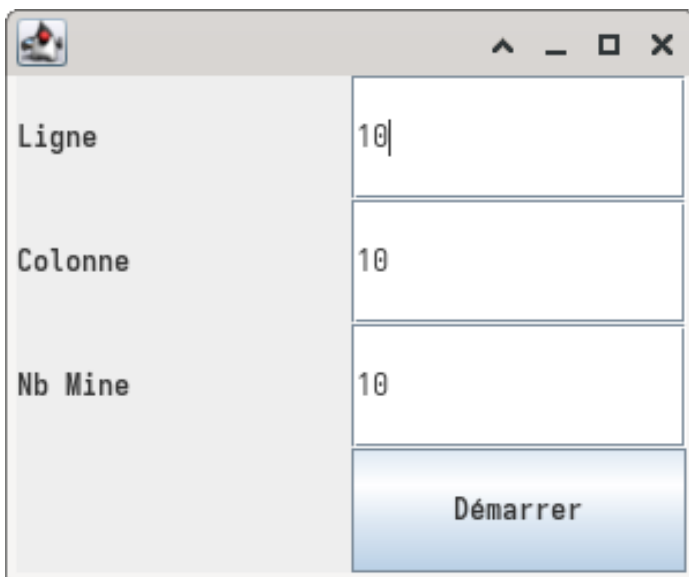


2. Configuration d'une nouvelle partie

Quand une nouvelle partie est lancée, une fenêtre permet au joueur de personnaliser entièrement sa grille. Il peut choisir :


- **Le nombre de lignes** (compris entre 4 et 30).
- **Le nombre de colonnes** (compris entre 4 et 30).
- **Le nombre total de mines** (qui doit obligatoirement être inférieur au nombre total de cases). Il y a des sécurité pour empêcher le lancement du jeu si les choix saisis ne respectent pas ces règles.

Configuration normale

A screenshot of a Java Swing window titled 'Minesweeper' (indicated by a mine icon). It contains three text input fields and a 'Démarrer' button. The first field is labeled 'Ligne' and contains '10'. The second field is labeled 'Colonne' and contains '10'. The third field is labeled 'Nb Mine' and contains '10'. The 'Démarrer' button is at the bottom.

Ligne	10
Colonne	10
Nb Mine	10
Démarrer	

Respectent pas les règles

A screenshot of the same Java Swing window, but with invalid values. The 'Ligne' field contains '32', the 'Colonne' field contains '32', and the 'Nb Mine' field contains '10000'. The 'Démarrer' button is still present at the bottom.

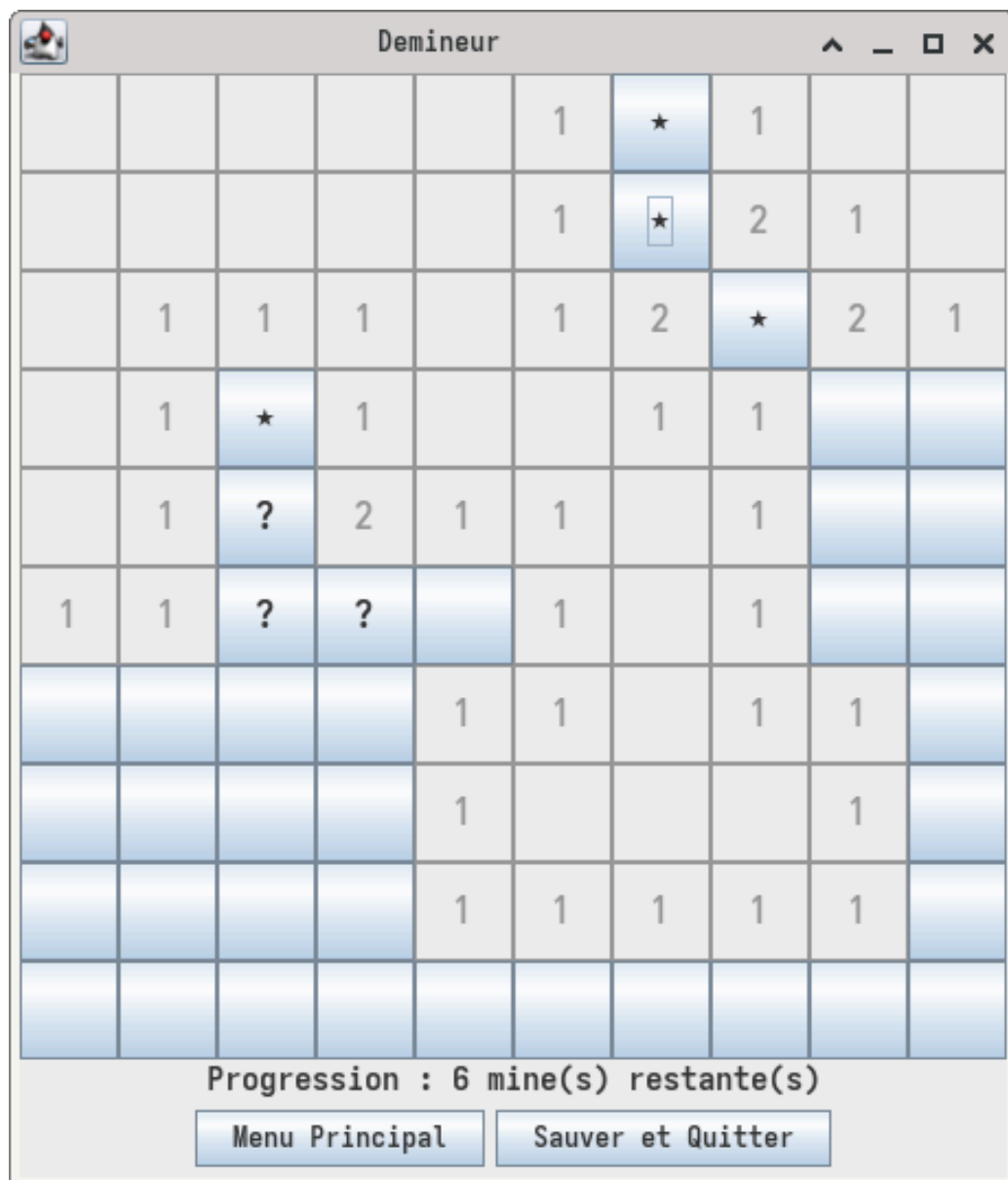
Ligne	32
Colonne	32
Nb Mine	10000
Démarrer	

```
[doucoure@archlinux SAE21_2025]$ make run
java Demineur
Erreur : Les dimensions doivent être comprises entre 4 et 30
Erreur : Le nombre de mines doit être inférieur au nombre de cases !
```

3. L'interface de jeu et les contrôles

Une fois la partie lancée, le joueur interagit avec la grille à l'aide de sa souris :

- **Clic gauche** : Permet de révéler une case cachée. Si la case ne possède aucune mine adjacente, on révèle automatiquement toutes les cases voisines en cascade. Le clic gauche est désactivé sur les cases avec un marqueur.
- **Clic droit** : Permet de faire défiler les marqueurs sur une case cachée pour aider le joueur à réfléchir (Vide -> Étoile ★ -> Doute ? -> Vide).
- **Compteur de progression** : Un compteur en bas de l'écran affiche le nombre de mines restantes à trouver (calculé en avec le nombre de marqueurs moins le nombre total de mines).

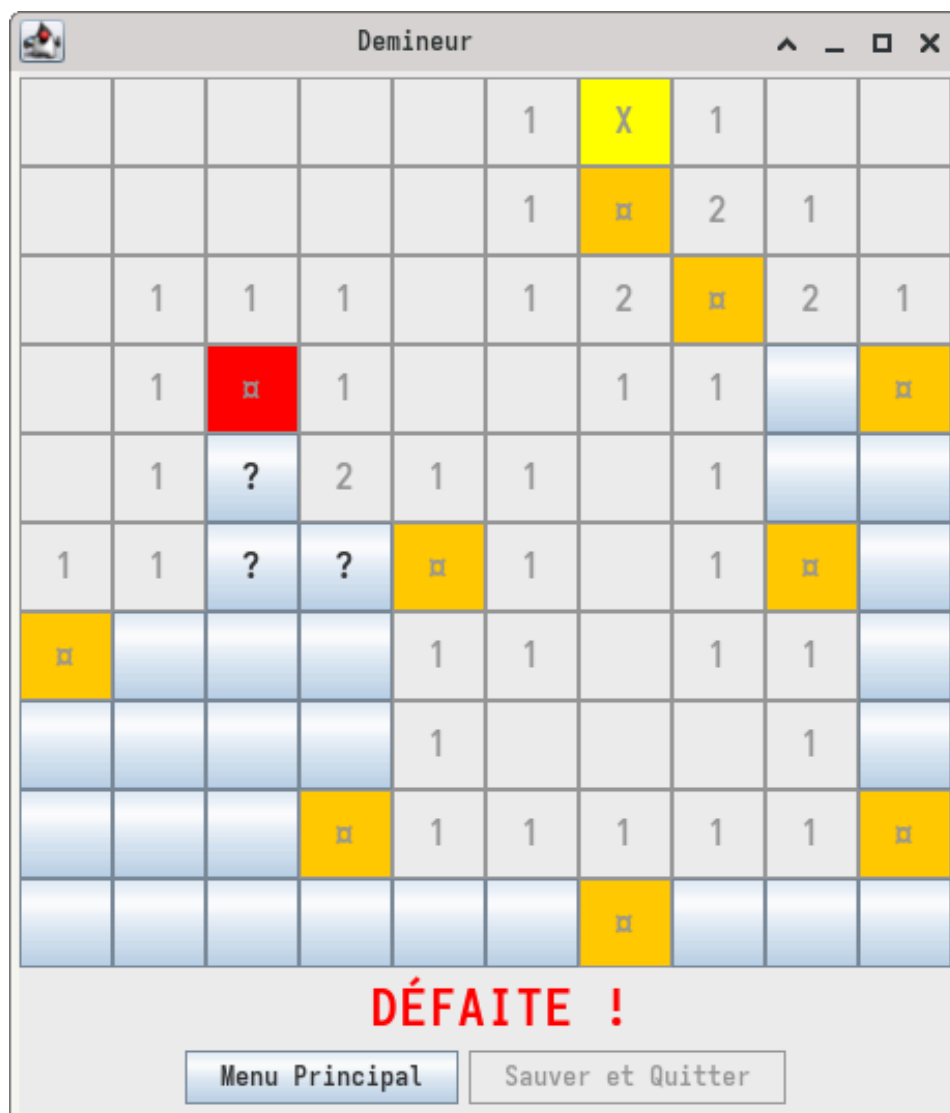


4. Fin de partie et Sauvegarde

L'application gère de manière visuelle la fin de la partie :

- **Défaite** : Si le joueur clique sur une mine, la partie s'arrête. La mine fatale s'affiche en **rouge**, les mines oubliées sont révélées en **orange**, et les cases où le joueur s'était trompé (fausse étoile) s'affichent en **jaune** avec une croix ("X"). Le joueur peut comme ça voir ses erreurs.
- **Victoire** : Révéler toutes les cases sans les bombes déclenche un message de victoire.
- **Sauvegarde** : Un bouton "Sauver et Quitter" (et la croix de fermeture de la fenêtre) permet de sauvegarder la progression. Pour éviter toute triche, la sauvegarde est impossible (et l'ancien fichier est supprimé) si la partie est déjà terminée.

En cas de défaite :



En cas de Victoire:

 Demineur ^ _ □ ×

	1	1	1				1	★	1
	1	★	1			1	2	2	1
	1	1	1			1	★	1	
					1	2	2	1	
					1	★	1		
			1	1	2	2	2	1	
			1	★	1	2	★	2	
			1	2	2	3	★	3	1
				1	★	3	3	★	1
				1	1	2	★	2	1

VICTOIRE ! Bien joué !

Menu Principal

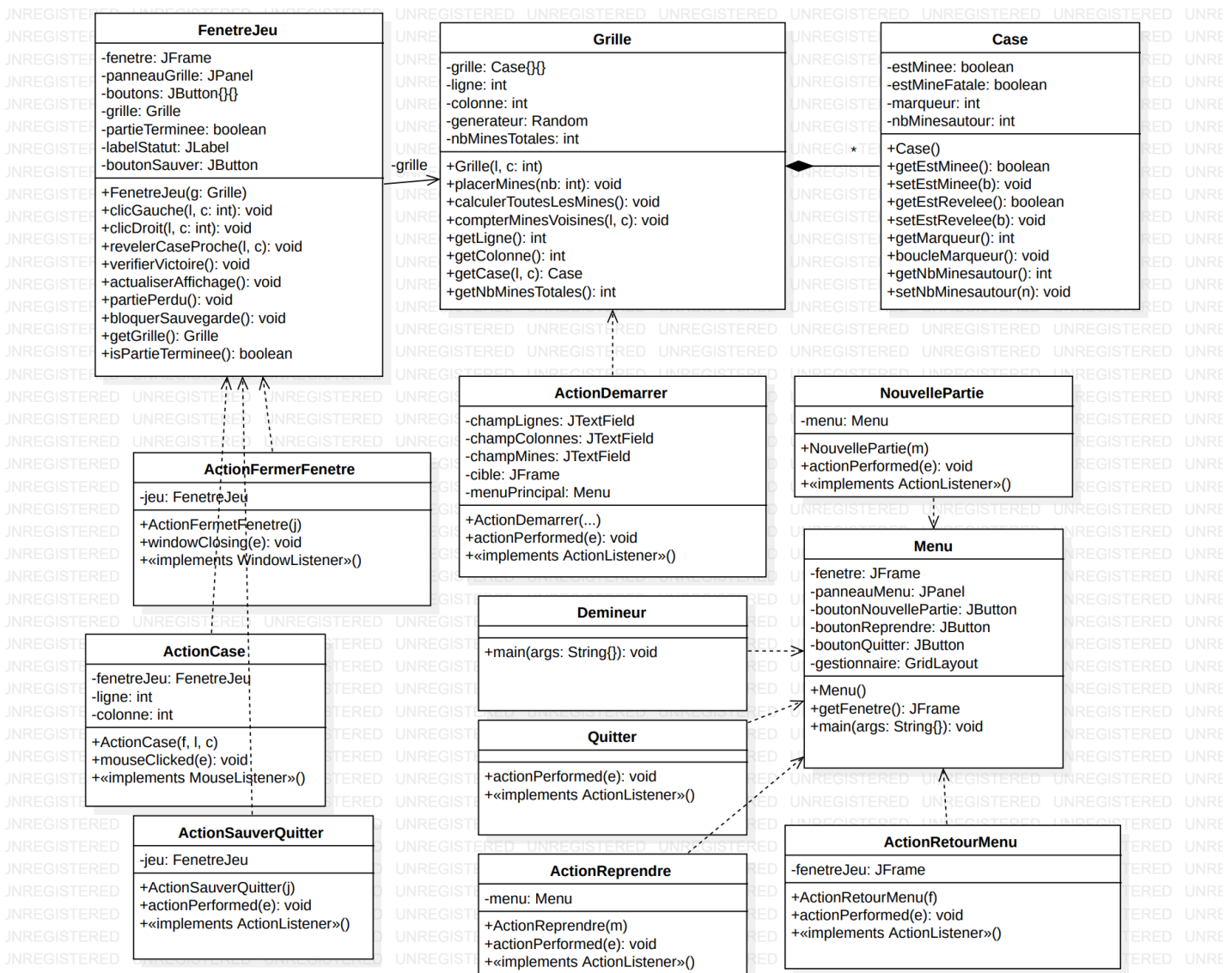
Sauver et Quitter

Structure du programme :

L'architecture est la suivante :

- **Le point d'entrée** : La classe **Démineur** contient la méthode principale (main) et lance le Menu.
- **L'interface utilisateur (Vue)** : Les classes **Menu**, **NouvellePartie** et **FenetreJeu** gèrent l'affichage (fenêtres, boutons, panneaux).
- **La logique** : La classe **Grille** gère les dimensions du plateau et le calcul des mines, la classe **Case** représente l'état de chaque case (minée, révélée, marqueur). Une **Grille** possède un tableau à deux dimensions d'objets **Case**.
- **Les contrôleurs (Actions)** : Toutes les interactions de l'utilisateur sont gérées par des classes qui implémentent les interfaces **ActionListener**, **MouseListener** ou **WindowListener** (ex: ActionCase, ActionDemarrer, ActionSauverQuitter). Ces classes font le lien entre les clics sur l'interface et les modifications dans la grille.

Diagramme de classe :



Mécanisme de sauvegarde et reprise:

Pour faire la sauvegarde de la partie , on a utiliser la **Sérialisation**

Plutôt que d'enregistrer manuellement l'état de chaque case (ses coordonnées, son état, etc.) dans un fichier texte brut, la sérialisation permet de "congeler" l'état d'un objet (la Grille et toutes les Case qu'elle contient) et de les mettre en octets. Ces octets sont ensuite mis dans un fichier nommé **sauvegarde.ser**.

La sauvegarde est déclenchée (uniquement si la partie n'est pas terminée) soit par le bouton "**Sauver et Quitter**", soit via la croix en haut à droite de la fenêtre. Quand le joueur veut reprendre sa partie depuis le menu principal, le programme effectue une Désérialisation. Il lit les octets du fichier pour re faire l'objet Grille avec toutes ses données en mémoire, ce qui permet à l'interface graphique de s'actualiser et de reprendre exactement là où le joueur s'était arrêté.

Algorithme de révélation en cascade

L'une des mécaniques du Démineur est la réaction en chaîne qui permet de révéler de grandes zones vides en un seul clic. Pour réaliser cela, nous avons implémenté une **méthode récursive** (une méthode qui s'appelle elle-même).

Le fonctionnement est le suivant :

1. Lorsqu'un joueur fait un clic gauche sur une case cachée, on vérifie d'abord que cette case ne contient pas de mine et n'a pas de marqueur.
2. La case est alors révélée. Le programme regarde le nombre de mines présentes dans les 8 cases adjacentes.
3. La condition de récursivité : Si, et seulement si, ce nombre est égal à 0 (la case est vide), le programme déclenche une boucle qui explore ses 8 cases voisines (haut, bas, gauche, droite et diagonales).
4. Pour chacune de ces cases voisines valides (situées dans les limites de la grille et non encore révélées), l'algorithme s'exécute à nouveau depuis l'étape 1.
5. La cascade s'arrête lorsqu'elle rencontre des cases adjacentes à au moins une mine (chiffre supérieur à 0).

Conclusion Personnelle

Conclusion de Ibrahim : Personnellement j'ai trouvé ce projet assez amusant ,hormis le fait que j'avais du retard dans les tp donc j'ai du sauter certain exercice pour apprendre comment fonctionne les flux d'octet ou exception, J'ai pu voir que le Java est beaucoup plus intuitif que le langage C , ça demande moins de ligne et c'est plus facile de retranscrire ce qu'on a en tête.

Conclusion de Tajeddine : De mon côté, ce projet SAE 2.1 a été une expérience d'apprentissage enrichissante, en particulier en ce qui concerne l'application concrète des concepts de programmation orientée objet en Java. Ce projet m'a permis de solidifier mes compétences en Java, notamment en manipulant les interfaces et les listeners et je suis satisfait du résultat final même si le jeu n'est pas très beau esthétiquement, il est complet et fonctionnel.