



UNIVERSITÉ CLERMONT AUVERGNE

INSTITUT SUPÉRIEUR D'INFORMATIQUE, DE MODÉLISATION ET DE
LEURS APPLICATIONS

DÉPARTEMENT D'INFORMATIQUE

Programmation Orientée Objet

**Mini Projet
DEMINEUR**

AVRIL 2023

Professeur :
Yannick LOISEAU

Étudiants :

Rodrigo FERREIRA RODRIGUES & M'Hammed Salmân ABIBOU

Contents

1	Démarche	3
2	Choix architecturaux	3
2.1	Direction	5
2.2	Case	5
2.3	Grille	6
2.4	Jeu	6
2.5	JeuGraphique	6

List of Figures

1	Diagramme UML	4
2	Ajout des voisins à une case	5
3	Jeu graphique démineur	7

Programmation Orientée Objet

Mini Projet

Rodrigo FERREIRA RODRIGUES & M'Hammed Salmân ABIBOU

Avril 2023



Pour le lancement du jeu.

Il faut lancer le jeu depuis le dossier "**demineur**" et non "**build**" parce que les relatifs des fichiers sont faits par rapport au répertoire où on saisit les commandes.

1 Démarche

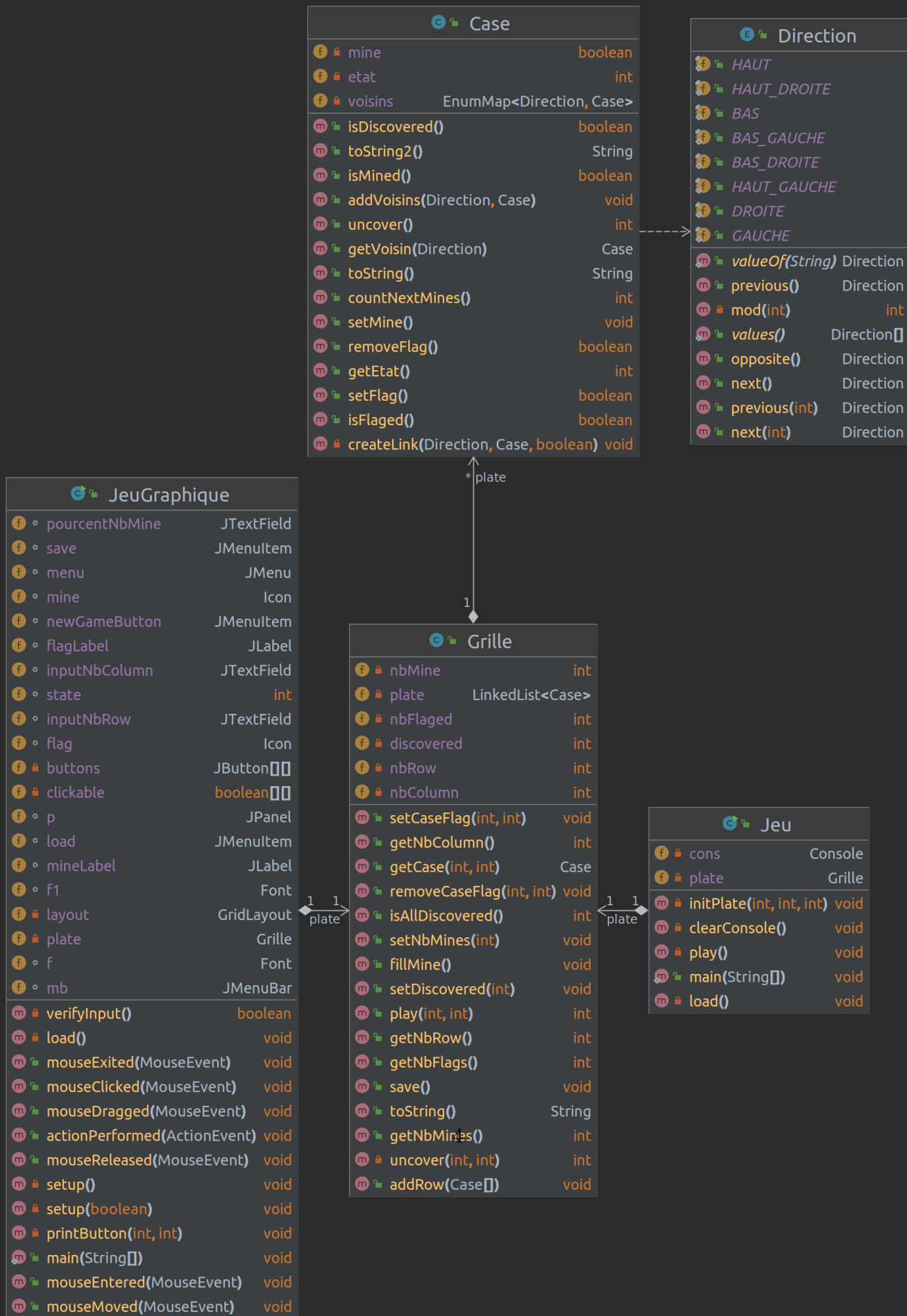
On réalise une version du jeu **démineur** en utilisant le langage de programmation orienté objet Java. Pour cela, on a suivi les principales étapes suivantes :

1. Création des différentes classes nécessaires (Direction, Case, Grille et Jeu) ;
2. Création du programme principal exécutable ;
3. Implémentation d'une interface graphique (JeuGraphique).

L'implémentation des différentes méthodes utilisées dans les classes ci-dessus seront explicitées dans la section suivante.

2 Choix architecturaux

Dans cette section, on détaille les différentes méthodes de chaque classe, pourquoi et comment elles ont été implémentées.



2.1 Direction

Direction contient l'énumération des 8 directions possibles `public enum Direction{GAUCHE, HAUT_GAUCHE, HAUT, HAUT_DROITE, DROITE, BAS_DROITE, BAS, BAS_GAUCHE}`; autour d'une case (les 8 cases voisines d'une case). Elle contient des méthodes pour aller dans les directions opposée, suivante et précédente.

2.2 Case

La classe **Case** gère la création d'une case et des liens référençant ses 8 voisins, l'état de la case, les mines et les drapeaux.

La méthode **addVoisins** (`public void addVoisins(Direction d, Case c);`) permet d'ajouter une case à la grille à partir d'une autre case déjà présente dans cette grille. Cette fonction permet de garantir qu'avec l'ajout d'une nouvelle case, les liens entre les cases ne soient pas brisés. Pour mieux comprendre, utilisons un exemple. Ici, on ajoute la case *c* à partir de la case *this* et la direction *droite*. À noter que les directions appartiennent à une énumération. Les fonctions `next()` et `previous()` permettent de passer d'une direction à une autre dans le sens des aiguilles d'une montre et inversement respectivement. Tout d'abord, on va créer un lien entre ces deux cases. Donc, on ajoute à *this* *c* comme voisin droit et à *c* *this* comme voisin gauche (la fonction `createLink` se charge de ça). Or, si on veut maintenir la cohésion de la grille, toutes les cases qui sont voisines à *this* et à *c* doivent être liées à *c* aussi. Ce sont les **étapes 1 et 2**, en **rouge** et en **vert** respectivement. Pour créer ces liens, on part maintenant de *c*, car on sait qu'il n'est pas nul contrairement aux autres cases, et on manipule la direction *d* passée en paramètre pour qu'elle soit bonne. Or, *c* et les cases en vert ont aussi d'autres cases voisines en commun. On doit alors créer des liens si possible avec elles. Ce sont les **étapes 3 et 4**, en **magenta** et **bleu** respectivement. Le procédé utilisé est le même que précédemment.

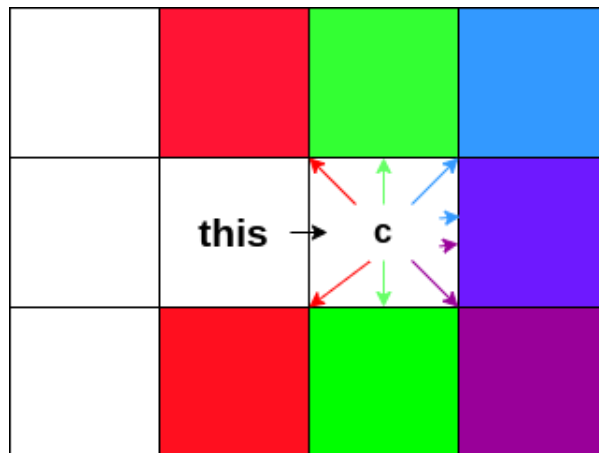


Figure 2: Ajout des voisins à une case

2.3 Grille

La classe **Grille** a plusieurs constructeurs pour créer une grille avec différentes configurations, ainsi que des méthodes pour manipuler et gérer la grille.

La grille est constituée de cases (représentées par la classe "Case") (section 2.2) disposées en lignes et colonnes. Les cases peuvent être soit découvertes, soit cachées et peuvent contenir une mine ou non. Les cases sont organisées sous forme d'une liste chaînée bidimensionnelle, avec des liens entre les cases voisines.

2.4 Jeu

La classe **jeu** permet d'effectuer une partie avec les caractéristiques de la grille que l'utilisateur aura paramétré.

2.5 JeuGraphique

Ici, il s'agira de créer une interface graphique du jeu basé sur les classes "Grille" et "Case". On a créé un programme exécutable **demineur.jar** qui permet de lancer le jeu¹.

¹java -jar demineur.jar

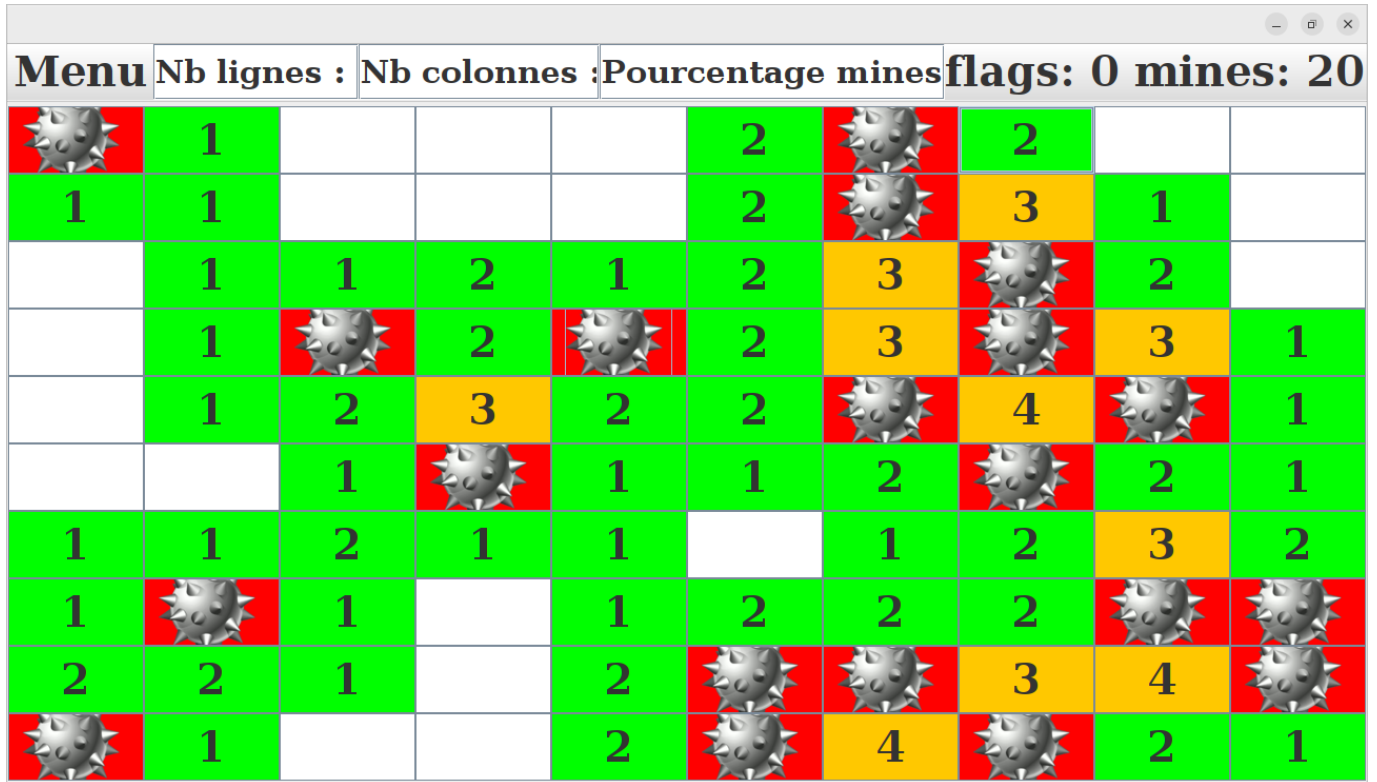


Figure 3: Jeu graphique démineur