

RAPPORT PROJET DEMINEUR

INFO2A

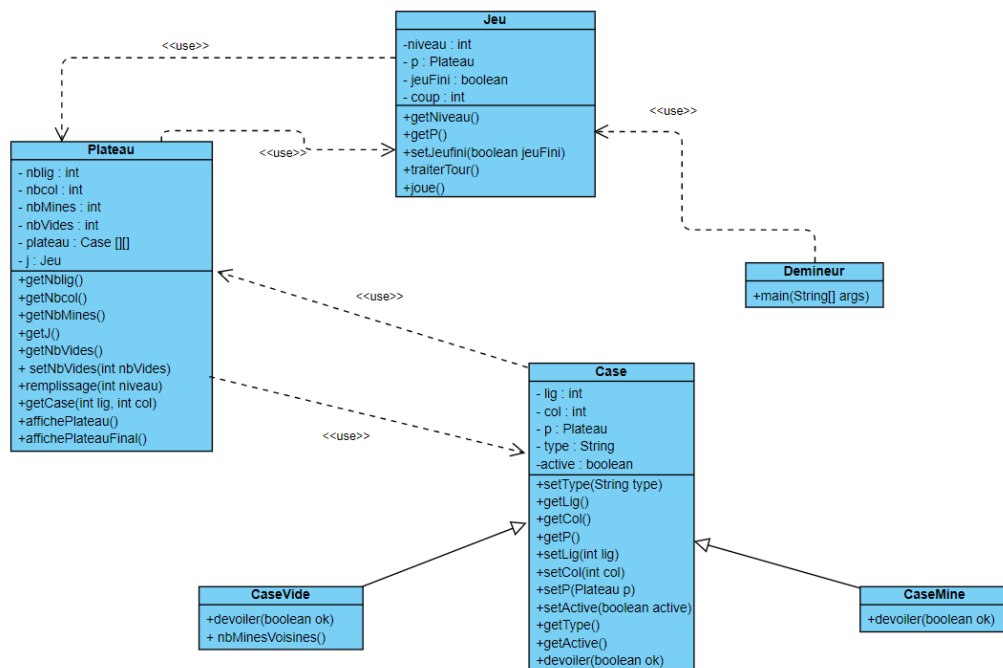
Description des fonctionnalités :

Nous avons réussi à coder toutes les fonctionnalités de base d'un jeu démineur :

- Créer un plateau carré.
- Dévoiler une case proche d'une mine, sans mines à proximité et une case mine.
- Etendre le dévoilement d'une case vide qui n'a pas de mines autour (principe de récursivité).
- Terminer le jeu si le joueur choisit une case mine.
- Faire gagner le joueur s'il a dévoilé toutes les cases vides et montrer le nombre de coups joués.
- Afficher le plateau complètement dévoilé la fin du jeu.

En plus des fonctionnalités de base nous avons réussi à créer un plateau rectangle.

Digramme de classe :



Descriptif des classes

Classe Case :

La classe case est une classe abstraite. Elle permet de représenter les cases pour la partie. C'est la classe mère des classes CaseVide et CaseMine.

Ses attributs :

- L'entier « lig » correspond à la ligne de la case.
- L'entier « col » correspond à la colonne de la case.

(Ces deux attributs sont utiles pour récupérer une case seulement avec des coordonnées)

- L'attribut « p » de type Plateau est le plateau dans lequel se trouve la case, ce qui permet de faire le lien avec l'instance Jeu de la partie.
- L'attribut « type » de type String permet de renseigner sur le type de case. Il est aussi utilisé pour l'affichage du plateau lors de la partie.
- Le booléen « active » permet de savoir si la case a déjà été dévoilée ou non.

Méthode **devoiler(boolean ok)** : méthode abstraite qui permet de dévoiler une case. Le contenu diffère selon les classes filles.

Classe CaseMine :

Cette classe hérite de la classe Case. Ses attributs sont identiques.

L'attribut « type » contient la chaîne de caractère « Mine ».

Méthode **devoiler(boolean ok)** : Lorsque la case mine est dévoilée, l'attribut « jeunfini » de l'instance Jeu est mis à true car la partie est terminée. Un message ainsi que le plateau dévoilé sont aussi affichés. Ici le booléen n'a aucun impact.

Classe CaseVide :

Cette classe hérite de la classe Case. Ses attributs sont identiques.

L'attribut « type » contient la chaîne de caractère « Vide » (ou le nombre de mines voisines si la case a été dévoilée).

Méthode **nbMinesVoisines()** : renvoie le nombre de mines à proximité

Méthode **devoiler(boolean ok)** : Cette méthode est l'une des plus compliquées car il y a plusieurs paramètres à prendre en compte.

Tout d'abord, il faut que la case n'ait pas déjà été dévoilée. Pour cela, nous vérifions si l'attribut « active » de la case est à true et que c'est le joueur qui souhaite dévoiler cette case (« ok » est alors à true lors de l'appel de la méthode). Si ces deux conditions sont respectées, le joueur ne peut pas dévoiler à nouveau la case.

Ensuite, il faut déterminer si la case a des mines à proximité ou non (d'où la création de la méthode **nbMinesVoisines()**).

Si des mines sont à proximité, seule la case vide est dévoilée. On change le type de la case avec le nombre de mines voisines, ce qui facilite l'affichage du plateau.

Si aucune mine n'est à proximité, il faut dévoiler toutes les cases adjacentes (et répéter l'opération par récursivité sur ces dernières). Notre plateau est volontairement agrandi d'une ligne ou colonne (que nous appelons cases « invisibles ») sur les extrémités afin de faciliter les tests pour dévoiler des cases vides sur les bords du plateau. Grâce à cela, nos boucles **for** sont les mêmes pour toutes les cases et nous ne sortons jamais des limites du tableau.

Cependant, cela signifie que nous devons tester si ces cases adjacentes ne sont pas des cases invisibles ou des cases déjà activées. Si ces deux conditions sont respectées, alors les cases sont révélées et leur nombre de mines voisines assigné à leur type.

Si la case adjacente dévoilée n'a aucune mine à proximité, il faut répéter cette opération sur son voisinage, d'où le rappel de la méthode dans la méthode elle-même avec le booléen false en paramètre, ce qui évite les erreurs pour dévoiler des cases à proximité de cases déjà révélées. Toutes les cases devant être dévoilées le sont par récursivité.

Classe Plateau :

Cette classe permet de représenter le plateau pour la partie. Elle fait le lien entre l'instance Jeu et les cases.

Ses attributs :

- Les entiers « nblig » et « nbcol » contiennent respectivement le nombre de colonnes et de lignes.
- Les entiers « nbMines » et « nbVides » contiennent respectivement le nombre de mines et de cases vides.
- Le tableau à deux dimensions « plateau » contient des éléments de type Case, il représente le plateau de jeu.
- L'attribut « j » de type Jeu correspond à l'instance de Jeu à laquelle est rattachée le plateau pour la partie.

Méthode **remplissage()** : Cette méthode permet de remplir le plateau suivant la difficulté du jeu.

Tout d'abord, tous les éléments du plateau sont transformés en case vide. Le type des cases « invisibles » est mis à -1 afin de pouvoir les reconnaître.

Ensuite, le nombre de mines est calculé ((ligne x colonne x difficulté) / 10). Des cases sont alors sélectionnées aléatoirement dans le tableau et sont transformées en case mine.

Méthode **getCase(lig,col)** : permet de récupérer une case dans le plateau avec ses coordonnées.

Méthode **affichePlateau()** : afficher le plateau durant la partie.

Méthode **affichePlateauFinal()** : afficher le plateau à la fin de la partie, comme si toutes les cases vides avaient été dévoilées.

Méthode **affichePlateauDevoile()** : permet durant la partie d'afficher le plateau dévoilé.

Classe Jeu :

Cette classe gère une partie de démineur.

Ses attributs :

- L'entier « niveau » contient le niveau choisi par le joueur pour la partie.
- L'attribut « p » de type Plateau contient le plateau utilisé pour la partie.
- Le booléen « jeu fini » permet de savoir si le jeu est terminé ou non.
- L'entier « coup » contient le nombre de coups réalisé par le joueur durant la partie.

Méthode **traiterTour()** : traite un tour durant la partie.

Méthode **joue()** : lance la partie et la continue tant qu'elle n'est pas terminée.

Jeux d'essais :

Exemple : Tableau 3x3 niveau 2

Veillez choisir la taille du plateau :

Nombre de lignes (compris entre 1 et 99) : 3

Nombre de colonnes (compris entre 1 et 99) : 3

Veillez choisir le niveau de jeu (Compris entre 1 et 9) : 2

Il y a 2 mines

[1 : 1] [1 : 2] [1 : 3]

[2 : 1] [2 : 2] [2 : 3]

[3 : 1] [3 : 2] [3 : 3]

Choisissez la case à dévoiler pour le coup 1

Choisissez la ligne : 2

Choisissez la colonne : 1

[1 : 1] [1 : 2] [1 : 3]

[2 : 1] [2 : 2] [2 : 3]

[3 : 1] [3 : 2] [3 : 3]

Choisissez la case à dévoiler pour le coup 2

Choisissez la ligne : 2

Choisissez la colonne : 3

[1 : 1] [1 : 2] [1 : 3]

[2 : 1] [2 : 2] [2 : 3]

[3 : 1] [3 : 2] [3 : 3]

Choisissez la case à dévoiler pour le coup 3

Choisissez la ligne : 1

Choisissez la colonne : 3

[1 : 1] [1 : 2] [1 : 3]

[2 : 1] [2 : 2] [2 : 3]

[3 : 1] [3 : 2] [3 : 3]

Choisissez la case à dévoiler pour le coup 4

Choisissez la ligne : 1

Choisissez la colonne : 1

[1 : 1] [1 : 2] [1 : 3]

[2 : 1] [2 : 2] [2 : 3]

[3 : 1] [3 : 2] [3 : 3]

Choisissez la case à dévoiler pour le coup 5
Choisissez la ligne : 3
Choisissez la colonne : 2

[2]	[1 : 2]	[2]
[2]	[2 : 2]	[2]
[3 : 1]	[1]	[3 : 3]

Choisissez la case à dévoiler pour le coup 6
Choisissez la ligne : 3
Choisissez la colonne : 1

[2]	[1 : 2]	[2]
[2]	[2 : 2]	[2]
[1]	[1]	[3 : 3]

Choisissez la case à dévoiler pour le coup 7
Choisissez la ligne : 3
Choisissez la colonne : 3
Vous avez gagné en 7 coups !

[2]	[Mine]	[2]
[2]	[Mine]	[2]
[1]	[1]	[1]

Exemple : Tableau 4x8 niveau 3

Veillez choisir la taille du plateau :
Nombre de lignes (compris entre 1 et 99) : 4
Nombre de colonnes (compris entre 1 et 99) : 8
Veillez choisir le niveau de jeu (Compris entre 1 et 9) : 3
Il y a 10 mines

[1 : 1]	[1 : 2]	[1 : 3]	[1 : 4]	[1 : 5]	[1 : 6]	[1 : 7]	[1 : 8]
[2 : 1]	[2 : 2]	[2 : 3]	[2 : 4]	[2 : 5]	[2 : 6]	[2 : 7]	[2 : 8]
[3 : 1]	[3 : 2]	[3 : 3]	[3 : 4]	[3 : 5]	[3 : 6]	[3 : 7]	[3 : 8]
[4 : 1]	[4 : 2]	[4 : 3]	[4 : 4]	[4 : 5]	[4 : 6]	[4 : 7]	[4 : 8]

Choisissez la case à dévoiler pour le coup 1
Choisissez la ligne : 2
Choisissez la colonne : 3

[1 : 1]	[1 : 2]	[1 : 3]	[1 : 4]	[1 : 5]	[1 : 6]	[1 : 7]	[1 : 8]
[2 : 1]	[2 : 2]	[3]	[2 : 4]	[2 : 5]	[2 : 6]	[2 : 7]	[2 : 8]
[3 : 1]	[3 : 2]	[3 : 3]	[3 : 4]	[3 : 5]	[3 : 6]	[3 : 7]	[3 : 8]
[4 : 1]	[4 : 2]	[4 : 3]	[4 : 4]	[4 : 5]	[4 : 6]	[4 : 7]	[4 : 8]

Choisissez la case à dévoiler pour le coup 2
 Choisissez la ligne : 2
 Choisissez la colonne : 4

[1 : 1]	[1 : 2]	[1 : 3]	[1 : 4]	[1 : 5]	[1 : 6]	[1 : 7]	[1 : 8]
[2 : 1]	[2 : 2]	[3]	[3]	[2 : 5]	[2 : 6]	[2 : 7]	[2 : 8]
[3 : 1]	[3 : 2]	[3 : 3]	[3 : 4]	[3 : 5]	[3 : 6]	[3 : 7]	[3 : 8]
[4 : 1]	[4 : 2]	[4 : 3]	[4 : 4]	[4 : 5]	[4 : 6]	[4 : 7]	[4 : 8]

Choisissez la case à dévoiler pour le coup 3
 Choisissez la ligne : 1
 Choisissez la colonne : 4
 Vous avez perdu !

[]	[1]	[Mine]	[Mine]	[1]	[1]	[2]	[Mine]
[]	[1]	[3]	[3]	[2]	[1]	[Mine]	[3]
[]	[1]	[2]	[Mine]	[3]	[4]	[4]	[Mine]
[]	[1]	[Mine]	[3]	[Mine]	[Mine]	[Mine]	[2]

Exemple : Tableau 3x3 niveau 1

Veuillez choisir la taille du plateau :
 Nombre de lignes (compris entre 1 et 99) : 3
 Nombre de colonnes (compris entre 1 et 99) : 3
 Veuillez choisir le niveau de jeu (Compris entre 1 et 9) : 1
 Il y a 1 mines

[1 : 1]	[1 : 2]	[1 : 3]
[2 : 1]	[2 : 2]	[2 : 3]
[3 : 1]	[3 : 2]	[3 : 3]

Choisissez la case à dévoiler pour le coup 1
 Choisissez la ligne : 1
 Choisissez la colonne : 3
 Vous avez gagné en 1 coups !

[]	[]	[]
[1]	[1]	[]
[Mine]	[1]	[]

Fin