

RAPPORT DU JEU DÉMINEUR

1 Description des fonctionnalités du jeu

2 -Description de la structure des classes à l'aide
d'un diagramme

3-Descriptif des classes

4- Jeux d'essais commentés

Choukri Imane

Fatima-Zahra Talibi

MI2-I2

• Description des fonctionnalités du jeu :

L'objectif du jeu est découvrir les n mines cachées ((on note que ces mines sont représentés par « m ») sous un plateau de jeu qui se présente sous forme d'une grille rectangulaire d'une dimension saisie par le joueur .

Le plateau est donc constitué de cases vides ou qui contiennent une mine (Plus exactement, pour une 1 difficulté de d avec l lignes et c colonnes, le nombre de mines est $(l*c*d)/10$).

Tout au long du jeu , le joueur a la possibilité de sélectionner une case du plateau (cela se fait en choisissant d'une part la ligne et de l'autre la colonne) .

Le joueur gagne si toutes les cases vides ont été dévoilées sans qu'une case contenant une mine, donc lors du clic sur une case contenant une mine le jeu est terminé, le contenu du plateau entier est dévoilé et le joueur a perdu.

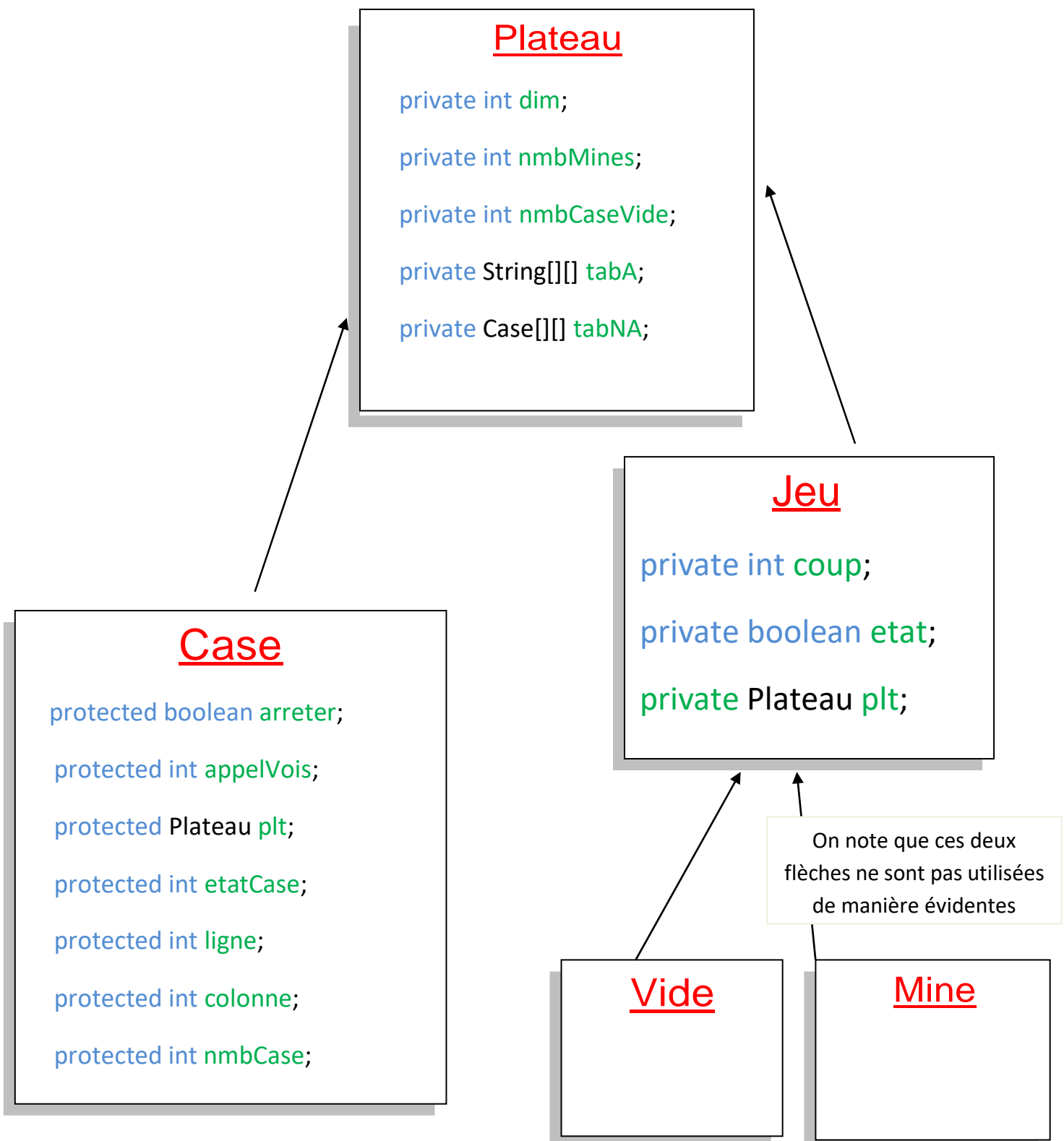
Cependant, lors du choix d'une case vide , le joueur peut continuer le jeu ,or , on se trouve devant deux cas :

- ❖ Si la case est vide et n'a aucune mine dans son entourage, elle est dévoilée ainsi que ses cases voisines.
- ❖ Si la case est vide et que certaines (Au moins une) de ses cases adjacentes contiennent des mines, le nombre de ces mines est affiché dans la case .

Pour finir , en cas de victoire (Si toutes les cases qui restent masquées contiennent des mines, le joueur gagne) le nombre de coups joués est affiché (nombre de sélections de cases effectuées par le joueur).

- Description de la structure des classes à l'aide d'un diagramme :

Le jeu doit communiquer au plateau son niveau ou le nombre de mines correspondant. Lors de son dévoilement, une « case mine » doit pouvoir signaler au jeu que le joueur a perdu. Le plateau doit pouvoir constater que toutes les cases vides ont été dévoilées et communiquer au jeu que le joueur a gagné. Pour calculer le nombre de mines qui l'entoure ou pour pouvoir propager le dévoilement de cases, une case vide doit pouvoir demander au plateau quelles sont ses cases voisines...



• Descriptif des classes :

- ❖ **Classe Plateau** : cette classe contient plusieurs attributs qui jouent un rôle essentiel afin d'assurer le fonctionnement de cette dernière :
 - ❖ **dim** : cet attribut est de type int (entier) et sert à gérer la taille du tableau , celle saisie par l'utilisateur .
 - ❖ **nmbMines** : cet attribut est de type int (entier) et sert à gérer le nombre de mines (qui seront calculés en fonction de la difficulté du jeu et le nombre totale des cases).
 - ❖ **nmbCaseVide** : cet attribut est de type int (entier) et sert à gérer le nombre de cases vide tout au long du déroulement du jeu .
 - ❖ **tabA** : cet attribut est de type String , c'est le tableau à deux dimensions qui va être afficher
 - ❖ **tabNA** : cet attribut est un tableau à deux dimensions contenant des éléments de type Case , il sert à stocker les mines et cases vides, contrairement au précédent, celui-ci ne s'affiche pas .

Cette classe contient aussi un constructeur avec paramètres qui initialisent les attributs avec des valeurs donné par l'utilisateur et des getters et setters des attributs déclarés : `getNmbMines()`, `getCase()`, `getTabNA()`, `setTabNA()`, `getDim()`, `setCase()`, `getNmbCase()`, et `setNmbCase()`. Elle fonctionne à l'aide de plusieurs méthodes :

- ❖ **public void RemplirPlateau()** : est une méthode qui sert à remplir le tableau à deux dimensionné nommé `tabA` , se remplissage se fait en fonction de la ligne et colonne de la case en question .
- ❖ **public void initPlateauCase()** : est une méthode qui remplit le tableau `tabNA` avec des cases vides .
- ❖ **Public void ajoutMine()** : Cette méthode ne retourne rien mais elle permet d'ajouter des mines dans le tableau `tab`, l'indice des cases autour de la mine sera augmenté (+1) en créant une nouvelle instance de la Classe vide et en ajoutant +1 a l'indice déjà présent .
- ❖ **public String runPlateau()** : Cette méthode créer le tableau non affiché avec ces mines si l'attribut `EtatCase` est 9 cela veut dire que c'est une mine donc on la case contient « m » si l'`EtatCase` est un 12 on remplace la case par une étoile sinon la case sera emplie par l'`EtatCase` lui-même .
- ❖ **public String affichagePlateau()** : Cette méthode sert tout simplement à afficher le plateau qui a été rempli grâce à la méthode `RemplirPlateau()`.

• **Classe Case** : Cette classe contient un constructeur avec paramètres qui initialise les attributs avec des valeurs donné par l'utilisateur , des accesseurs qui assurent le fonctionnement de la classe comme : `getAppelVois()` et `setAppelVois ()` , enfin plusieurs attributs :

- ❖ **boolean arreter**: cet attribut est de type Boolean il sert à arrêter le jeu il renvoie donc true pour mine et false pour une case vide .

- ❖ Plateau plt :cet attribut désigne tout simplement le plateau dont aura besoin los du jeu.
- ❖ int appelVois: Cet attribut est un entier qui est initialisé au début à une valeur de 0 puis 1 lors de l'utilisation de cette méthode .
- ❖ int etatCase: cet attribut est un entier qui nous aide à vérifier si la case est vide ou contient une mine,si elle est vide il vaut 0, si elle contient une mine il vaut 9.
- ❖ int ligne : cet attribut est un entier qui désigne la ligne de la case courante du tableau .
- ❖ int colonne : cet attribut est un entier qui désigne la colonne de la case courante du tableau .
- ❖ int nmbCase : Cet attribut désigne le nombre de case il sera utile dans des méthodes des classes filles de la classe case .

•Classe Jeu : Cette classe contient un constructeur par défaut et trois attributs :

- ❖ int coup : Cet attribut est un entier qui gère le nombre de coups fait par l'utilisateur,il est initialisé à une valeur de 0 .
- ❖ boolean etat : Cet attribut permet de gérer l'état du tableau ,il est utilisé dans la méthode joue().
- ❖ Plateau plt : cet attribut désigne tout simplement le plateau dont aura besoin los du jeu.

Cette classe contient aussi la méthode **joue()**,elle propose au joueur de saisir le niveau de difficulté du jeu ainsi que sa dimension afin d'initialiser le plateau et le nombre de mine ,pour cela on fait appel à des méthodes comme initPlateauCase() , remplirPlateau() et ajoutMine() .Après la création du tableau on affiche le nombre de mines existant puis grace à une boucle on répète la même action (proposer au joueur de saisir une case donc une ligne et une colonne puis passer à l'étape suivante tant que le nombre de case est différent de zero et this.etat == false .

•Classe Vide : Cette classe comporte un constructeur avec paramètres qui initialise les attributs avec des valeurs donné par l'utilisateur (cet constructeur fait appel à celui de la super classe) , plusieurs méthodes dont : getappelVois() , setappelVois(),getEtatCase() et getArreter().

De plus, elle contient :

- la méthode active() qui fait appel à la méthode Voisinage() puis vérifie si le nombre de case n'est pas nul pour afficher le tableau avec toutes les cases vidées suivi d'un message qui montre que l'utilisateur a gagné. Sinon on affiche le nouveau tableau avec les cases vidées grâce à la méthode voisinage.

- La méthode Voisinage() qui fait appel à son tour à des méthodes externes comme getCase() et getappelVoi(). Elle sert à stocker dans la variable « avant », l'entier créé en fonction de la ligne, la colonne et le tableau saisi par l'utilisateur, et qui nous permettra par la suite de faire appel à la méthode (Remp ())
- La méthode Voisinage() est une méthode récursive qui utilise sa propre méthode au sein de son code, elle nous permet de mettre en paramètre le tableau, la ligne, la colonne et deux entiers « avant » et « nouveau ». Elle permet de changer la case courante d'indice (l,c) du tableau affiché par l'indice de la case en position (l,c) de tableau non affiché cela se fait suivant le code suivant : `tab.setCase(l,c,tab.getCase(l,c).getIndCase());`. Finalement une case est changée si ce n'est pas une mine, et l et c sont dans le tableau et si l'attribut isAf = 0, une fois la case changée l'attribut isAf = 1 avec la méthode `tab.getCase(l,c).setIsAf();`, elle lance l'appel récursif sur les cases à côté (haut, bas, gauche, droite)

Classe Mine : Cette classe comporte un constructeur avec paramètres qui initialise les attributs avec des valeurs données par l'utilisateur, des méthodes comme : `getappelVoi()`, `setappelVoi()`, `getEtatCase()`, `getArreter()`.

De plus, une méthode spécifique nommée `active()` qui sert à initialiser l'attribut `etatCase` à une valeur de 9 puis à afficher le tableau suivi d'un message « Vous avez Malheureusement Perdu ».

• Jeux d'essais commentés :

- ❖ Cas d'un jeu gagnant :

```

run:
Binevenue dans le jeu du démineur!!

Choisissez le niveau de difficulté
1
Saisir la dimension
10
{0;0} {0;1} {0;2} {0;3} {0;4} {0;5} {0;6} {0;7} {0;8} {0;9}
{1;0} {1;1} {1;2} {1;3} {1;4} {1;5} {1;6} {1;7} {1;8} {1;9}
{2;0} {2;1} {2;2} {2;3} {2;4} {2;5} {2;6} {2;7} {2;8} {2;9}
{3;0} {3;1} {3;2} {3;3} {3;4} {3;5} {3;6} {3;7} {3;8} {3;9}
{4;0} {4;1} {4;2} {4;3} {4;4} {4;5} {4;6} {4;7} {4;8} {4;9}
{5;0} {5;1} {5;2} {5;3} {5;4} {5;5} {5;6} {5;7} {5;8} {5;9}
{6;0} {6;1} {6;2} {6;3} {6;4} {6;5} {6;6} {6;7} {6;8} {6;9}
{7;0} {7;1} {7;2} {7;3} {7;4} {7;5} {7;6} {7;7} {7;8} {7;9}
{8;0} {8;1} {8;2} {8;3} {8;4} {8;5} {8;6} {8;7} {8;8} {8;9}
{9;0} {9;1} {9;2} {9;3} {9;4} {9;5} {9;6} {9;7} {9;8} {9;9}

Le nombre de mine trouvée est:10
Le nombre de case vide trouvée est:90
Le nombre de coup est: 0
veuillez sélectionner une case dans la grille
Ligne est:
5
Colonne est:
4
{0;0} { 1 } {   } { 1 } {0;4} { 1 } {   } {0;7} {   } {0;9}
{1;0} { 1 } {   } { 1 } {1;4} { 1 } { 1 } { 1 } { 2 } { 1 }
{ 1 } { 1 } {   } { 1 } { 1 } { 1 } { 1 } { 1 } {2;7} { 2 } {2;9}
{   } {   } {   } {   } {   } { 1 } { 2 } { 2 } { 3 } { 2 }
{   } {   } {   } {   } {   } { 1 } {4;6} { 2 } { 2 } {4;9}
{   } {   } {   } {   } {   } { 1 } { 2 } {5;7} { 2 } { 1 }
{   } {   } {   } {   } {   } {   } { 1 } { 1 } { 1 } {   }
{   } {   } {   } {   } {   } { 1 } { 1 } { 1 } {   } {   }
{   } {   } {   } { 1 } { 1 } { 2 } {8;6} { 1 } { 1 } { 1 }
{   } {   } {   } { 1 } {9;4} { 2 } {9;6} { 1 } { 1 } {9;9}

Le nombre de case vide trouvée est:5
Le nombre de coup est: 1

```

```

Ligne est:
0
Colonne est:
0
{ 1 } { 1 } {   } { 1 } {0;4} { 1 } {   } {0;7} {   } {0;9}
{1;0} { 1 } {   } { 1 } {1;4} { 1 } { 1 } { 1 } { 2 } { 1 }
{ 1 } { 1 } {   } { 1 } { 1 } { 1 } { 1 } { 1 } {2;7} { 2 } {2;9}
{   } {   } {   } {   } {   } { 1 } { 2 } { 2 } { 3 } { 2 }
{   } {   } {   } {   } {   } { 1 } {4;6} { 2 } { 2 } {4;9}
{   } {   } {   } {   } {   } { 1 } { 2 } {5;7} { 2 } { 1 }
{   } {   } {   } {   } {   } {   } { 1 } { 1 } { 1 } {   }
{   } {   } {   } {   } {   } { 1 } { 1 } { 1 } {   } {   }
{   } {   } {   } { 1 } { 1 } { 2 } {8;6} { 1 } { 1 } { 1 }
{   } {   } {   } { 1 } {9;4} { 2 } {9;6} { 1 } { 1 } {9;9}

Le nombre de case vide trouvée est:4
Le nombre de coup est: 2
veuillez sélectionner une case dans la grille
Ligne est:
5
Colonne est:
7
{ 1 } { 1 } { 0 } { 1 } { 1 } { 1 } { 0 } { 0 } { 0 } { 0 }
{ m } { 1 } { 0 } { 1 } { m } { 1 } { 1 } { 1 } { 2 } { 1 }
{ 1 } { 1 } { 0 } { 1 } { 1 } { 1 } { 1 } { m } { 2 } { m }
{ 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 1 } { 2 } { 2 } { 3 } {2;9}
{ 0 } { 0 } { 0 } { 0 } { 0 } { 1 } { m } { 2 } { 2 } { m }
{ 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 1 } { 2 } { * } { 2 } { 1 }
{ 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 1 } { 1 } { 1 } { 0 }
{ 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 1 } { 1 } { 1 } { 0 } { 0 }
{ 0 } { 0 } { 0 } { 1 } { 1 } { 2 } { m } { 1 } { 1 } { 1 }
{ 0 } { 0 } { 0 } { 1 } { m } { 2 } { 1 } { 1 } { 1 } { m }

Vous avez Malheureusement Perdu
{ 1 } { 1 } { 0 } { 1 } { 1 } { 1 } { 0 } { 0 } { 0 } { 0 }
{ m } { 1 } { 0 } { 1 } { m } { 1 } { 1 } { 1 } { 2 } { 1 }
{ 1 } { 1 } { 0 } { 1 } { 1 } { 1 } { 1 } { m } { 2 } { m }
{ 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 1 } { 2 } { 2 } { 3 } {2;9}
{ 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 1 } { m } { 2 } { m }
{ 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 1 } { 2 } { * } { 2 } { 1 }
{ 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 1 } { 1 } { 1 } { 0 }
{ 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 1 } { 1 } { 1 } { 0 } { 0 }
{ 0 } { 0 } { 0 } { 1 } { 1 } { 2 } { m } { 1 } { 1 } { 1 }
{ 0 } { 0 } { 0 } { 1 } { m } { 2 } { 1 } { 1 } { 1 } { m }

BUILD SUCCESSFUL (total time: 17 seconds)

```

❖ Cas d'un jeu perdant :


```

run:
Binevenue dans le jeu du démineur!!

Choisissez le niveau de difficulté
1
Saisir la dimension
10
{0;0} {0;1} {0;2} {0;3} {0;4} {0;5} {0;6} {0;7} {0;8} {0;9}
{1;0} {1;1} {1;2} {1;3} {1;4} {1;5} {1;6} {1;7} {1;8} {1;9}
{2;0} {2;1} {2;2} {2;3} {2;4} {2;5} {2;6} {2;7} {2;8} {2;9}
{3;0} {3;1} {3;2} {3;3} {3;4} {3;5} {3;6} {3;7} {3;8} {3;9}
{4;0} {4;1} {4;2} {4;3} {4;4} {4;5} {4;6} {4;7} {4;8} {4;9}
{5;0} {5;1} {5;2} {5;3} {5;4} {5;5} {5;6} {5;7} {5;8} {5;9}
{6;0} {6;1} {6;2} {6;3} {6;4} {6;5} {6;6} {6;7} {6;8} {6;9}
{7;0} {7;1} {7;2} {7;3} {7;4} {7;5} {7;6} {7;7} {7;8} {7;9}
{8;0} {8;1} {8;2} {8;3} {8;4} {8;5} {8;6} {8;7} {8;8} {8;9}
{9;0} {9;1} {9;2} {9;3} {9;4} {9;5} {9;6} {9;7} {9;8} {9;9}

Le nombre de mine trouvée est:10
Le nombre de case vide trouvée est:90
Le nombre de coup est: 0
veuillez sélectionner une case dans la grille
Ligne est:
6
Colonne est:
4
Bravo,Vous avez gagné!!!
{ 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 }
{ 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 }
{ 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 }
{ 1 } { 1 } { 0 } { 1 } { 1 } { 1 } { 0 } { 0 } { 1 } { 1 }
{ m } { 1 } { 0 } { 1 } { m } { 1 } { 0 } { 0 } { 1 } { m }
{ 1 } { 1 } { 1 } { 2 } { 2 } { 1 } { 1 } { 1 } { 3 } { 2 }
{ 0 } { 0 } { 2 } { m } { 2 } { 1 } { 2 } { m } { 3 } { m }
{ 0 } { 0 } { 2 } { m } { 2 } { 1 } { m } { 2 } { 3 } { m }
{ 0 } { 1 } { 2 } { 2 } { 1 } { 1 } { 1 } { 1 } { 1 } { 1 }
{ 0 } { 1 } { m } { 1 } { 0 } { 0 } { 0 } { 0 } { 0 } { 0 }

```

- Le jeu du démineur est un jeu prévu pour un seul joueur, et son principe consiste au début à demander au joueur de choisir le niveau de difficulté (1,2,3...) à condition qu'il ne doit pas dépasser 9.
- Ainsi que de Saisir la dimension du plateau qui influe sur le nombre de mines, après cette étape
- On affiche à l'utilisateur le nombre de mines et de cases vides trouvées, puis en lui demande de choisir une case (une ligne puis une colonne)
- Alors, si une case contient une mine le joueur dans ce cas a perdu ce qui provoque la fin du jeu,
- Sinon si la case est vide et n'a aucune mine dans son entourage, toutes ses cases vides adjacentes Sont aussi dévoilées.
- Cependant, si la case est vide et possède une mine dans son entourage, la case saisie affiche Le nombre de mines trouvées dans son voisinage.
- Finalement, on peut gagner dans le cas où on a dévoilé toutes les cases vides.

MERCI

