

# **Jeu du démineur**

## Sommaire

---

I. Présentation du sujet.....	3
a. Les règles du jeu.....	3
b. Les fonctionnalités du jeu.....	3
II. Présentation technique.....	6
a. Présentation du code.....	6
b. Explication de la sauvegarde.....	6
c. Explication pour révéler plusieurs cases.....	6
III. Conclusion.....	7
IV. Annexe.....	8

# **I. Présentation du sujet**

## **a. Les règles du jeu**

Le démineur est un jeu de réflexion qui consiste à trouver un nombre de bombes cachées dans une grille.

Pour cela le joueur clique sur une case ce qui permet de révéler son contenu. Une fois la case découverte, il est indiqué si elle contient une bombe, alors la partie est perdue.

Par contre, si il n'y a pas de bombe un chiffre renseigne sur le nombre de bombes adjacentes à cette case, si le nombre est 0 alors le programme révèle automatiquement les cases vides qui se trouvent autour.

Le joueur a également la possibilité de marquer la case avec une ★ ou un ? afin de se souvenir de ses raisonnements (★ si le joueur pense qu'une bombe est située sur cette case, ? si le joueur a un soupçon).

## **b. Les fonctionnalités du jeu**

Un premier écran titre propose au joueur de créer une partie ou de quitter le jeu, ensuite si il crée une partie il aura les choix suivants :

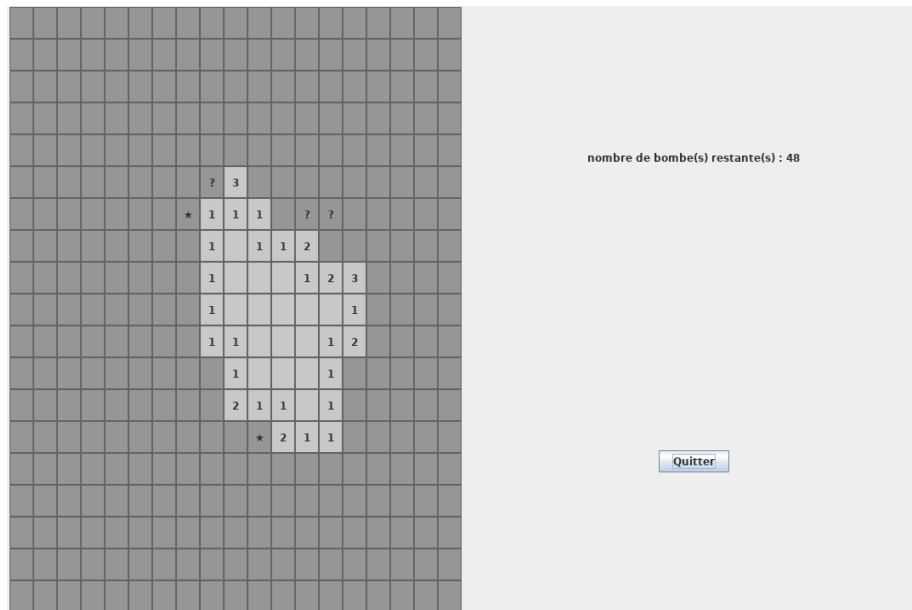
- faire une nouvelle partie
- continuer une partie précédente commencée et sauvegardée
- quitter le jeu



Quand le joueur crée une nouvelle partie il peut choisir le nombre de lignes, de colonnes et de bombes.

The image shows a configuration screen for the Minesweeper game. It has a light gray background. In the center, there are three lines of text, each followed by a text input field. The first line is 'nombre de lignes (min 4, max 30) : 19', the second is 'nombre de colonnes (min 4, max 30) : 19', and the third is 'nombre de bombes : 50'. Below these input fields are two blue buttons with white text: 'annuler' on the left and 'valider' on the right.

Ensuite, l'écran de jeu apparaît si le joueur souhaite quitter la partie, celle-ci sera automatiquement sauvegardée.



Une fois la partie terminée, la sauvegarde sera automatiquement supprimée et une fenêtre apparaîtra pour signaler au joueur si il a gagné ou perdu. Après cela il peut décider de refaire une partie ou de quitter le jeu.



## **II. Présentation technique**

### **a. Présentation du code**

Le code est séparé en plusieurs parties ([voir annexe](#)) et il contient quatre «class» principales qui représentent les trois phases du jeu (PanneauChoix, PanneauJeu, MenuJeu) et la dernière qui représente le corps du jeu (Main).

### **b. Explication de la sauvegarde**

La sauvegarde est enregistrée au format machine. Les deux premières données correspondent à la largeur et la hauteur de la grille, la troisième correspond au nombre de bombes situées dans la grille.

Les données suivantes sont regroupées par paire et correspondent aux données de la grille, la première permet de savoir ce qu'il y a dans la case et la deuxième permet de connaître le statut d'affichage

### **c. Explication pour révéler plusieurs cases**

Une fonction récursive est utilisée afin pouvoir révéler les cases vides adjacentes à une case elle-même vide.

La fonction consiste à tourner jusqu'à ce qu'elle rencontre une case avec une ou des bombes à côté. Alors la fonction s'arrête et toutes les cases qui ont été testées par la fonction sont révélées.

### **III. Conclusion**

Après avoir réalisé un premier projet en binôme, le codage d'un programme en solo m'a permis de me tester afin de connaître mes capacités de réalisation de projet en totale autonomie. Ainsi j'ai pu, pour la première fois, utiliser une interface graphique ainsi que «git».

## IV. Annexe

