



L'objectif de ce DM est de programmer un jeu de démineur.

Le démineur est un jeu vidéo de réflexion né dans les années 60 et popularisé par le système d'exploitation Windows dont le but est de localiser des mines cachées dans une grille représentant un champ de mines virtuel, avec pour seule indication le nombre de mines dans les zones adjacentes.

Au début du jeu, le programme affiche une aire de jeu rectangulaire constitué de cases inconnues. Les cases de l'aire de jeu peuvent être vide ou contenir une mine. Le programme a déterminé au hasard la localisation d'un nombre fixé de mine dans la grille. Cette localisation n'est pas connue du joueur : une case vide et une case contenant une mine ont la même apparence tant que la case n'a pas été découverte.

Le joueur peut alors désigner une case qu'il souhaite découvrir :

- si cette case contenait une mine, le jeu est fini et le joueur a perdu. Le programme affiche alors la carte complète avec l'emplacement des mines ;
- si cette case est vide, le programme détermine le nombre de mines dans les 8 cases adjacentes à celle-ci et l'inscrit dans la case. Si ce nombre est nul (aucune mine dans les cases immédiatement adjacentes), le programme découvre automatiquement toutes les cases adjacentes récursivement ce qui fait qu'on découvre une région entière bordées de cases portant un numéro (nombre de mines adjacentes).

Le jeu s'arrête et le joueur a gagné s'il reste un nombre de cases non découvertes égal au nombre de mines en jeu.

Vous devez programmer en python un jeu de démineur qui respecte le cahier des charges minimal suivant :

- ✓ le programme doit permettre de jouer sur une grille carrée de 10×10 cases contenant 10 mines. Celle-ci est générée aléatoirement par le programme à chaque partie ;
- ✓ le programme fonctionne en mode texte en tour par tour ;
- ✓ à chaque tour, le programme affiche la grille d'exploration puis demande au joueur la case à découvrir. La case choisie est découverte selon les règles du jeu vues plus haut ;
- ✓ la case choisie pourra être désignée par le joueur sous une forme compacte, comme « C6 » ;
- ✓ le programme détecte et affiche la fin de de partie aussi bien pour la défaite (case contenant une mine) que la victoire (il reste 10 cases non découvertes) ;
- ✓ quand la partie est finie, le programme affiche la carte d'exploration sur laquelle il a ajouté les mines non découvertes et marqué la mine ayant explosé en cas de défaite. Il est alors proposé au joueur de faire une autre partie.

```

Quelle case voulez-vous
découvrir ? D5
      ABCDEFGHIJ
0  .....  0
1  12.....  1
2  1.....  2
3  1111111..  3
4  .....1..  4
5  .....111  5
6  111111    6
7  2....1    7
8  13....2   8
9  .....1    9
      ABCDEFGHIJ
. : case non découverte
. : case découverte et vide

```

Ceci est le travail minimal, mais libre à vous d'améliorer le programme en prenant en charge par exemple :

- la pose de drapeaux : le joueur peut poser des drapeaux sur les cases où il pense que se trouve une mine. Jouer sur une case contenant un drapeau enlève celui-ci sans découvrir la case (sécurité). Les drapeaux comptent comme des cases vides pour la condition de victoire ;
- différentes tailles de grille et un nombre de mines réglable ;
- un chronomètre qui permet d'afficher le temps nécessaire à la résolution de la grille ;
- une interface graphique avec tkinter ou avec pyxel pour un jeu avec la souris.