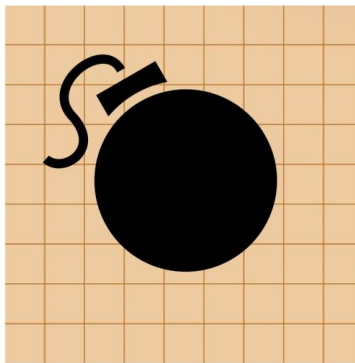


Filière Première Année Cycle d'Ingénieur : Sciences des données, Big Data & IA



Jeu-Démineur

Projet réalisé par :

Hicham Mimouni

Ahmed Gadi

Sous la supervision de :

Monsieur Mohammed Kasri

Explication des parties de code

Fonction 1: Initialisation de la grille

```
1 void initialiserGrille() {
2     for (int i = 0; i < TAILLE; i++) {
3         for (int j = 0; j < TAILLE; j++) {
4             grille[i][j] = ' ';
5             grilleVisible[i][j] = '-';
6         }
7     }
8     minesPlacees = 0;
9 }
```

Cette fonction initialise la grille de jeu. Elle parcourt chaque cellule de la grille et la met à l'état initial : pas de mine (' ') et non visible ('-'). Elle réinitialise également le compteur de mines placées.

Fonction 2: Placement des mines

```
1 void placerMines(int difficulte) {
2     int MINES = 10 * difficulte;
3     srand(time(NULL));
4     while (minesPlacees < MINES) {
5         int x = rand() % TAILLE;
6         int y = rand() % TAILLE;
7         if (grille[x][y] != '*') {
8             grille[x][y] = '*';
9             minesPlacees++;
10        }
11    }
12 }
```

Cette fonction place un nombre de mines sur la grille en fonction de la difficulté choisie par l'utilisateur. Elle génère des coordonnées aléatoires pour chaque mine et s'assure qu'il n'y a pas déjà une mine à cet endroit.

Fonction 3: Affichage de la grille

```
1 void afficherGrille() {
2     printf(" ");
3     for (int j = 0; j < TAILLE; j++) {
4         printf("[%d]", j);
5     }
6     printf("\n");
7
8     for (int i = 0; i < TAILLE; i++) {
9         printf("[%d]", i);
10        for (int j = 0; j < TAILLE; j++) {
11            printf("[%c]", grilleVisible[i][j]);
12        }
13        printf("\n");
14    }
15 }
```

Cette fonction affiche la grille de jeu à l'utilisateur. Elle parcourt chaque cellule de la grille et affiche son état actuel.

Fonction 4: Basculer le drapeau

```
1 void basculerDrapeau(int x, int y) {
2     if (grilleVisible[x][y] == '-') {
3         grilleVisible[x][y] = 'd';
4     } else if (grilleVisible[x][y] == 'd') {
5         grilleVisible[x][y] = '-';
6     }
7 }
```

Cette fonction permet à l'utilisateur de placer ou d'enlever un drapeau sur une cellule de la grille. Si la cellule est actuellement non visible ('-'), elle devient un drapeau ('d'). Si elle est un drapeau, elle redevient non visible.

Fonction 5: Compter les mines adjacentes

```
1 int compterMinesAdjacentes(int x, int y) {
2     int count = 0;
3     for (int i = x - 1; i <= x + 1; i++) {
4         for (int j = y - 1; j <= y + 1; j++) {
5             if (i >= 0 && i < TAILLE && j >= 0 && j < TAILLE &&
6                 grille[i][j] == '*') {
7                 count++;
8             }
9         }
10    }
11    return count;
12 }
```

Cette fonction compte le nombre de mines dans les cellules adjacentes à une cellule donnée. Elle parcourt chaque cellule autour de la cellule donnée et incrémente un compteur chaque fois qu'elle trouve une mine.

Fonction 6: Révéler une case

```
1 void revelerCase(int x, int y) {
2     if (x < 0 || x >= TAILLE || y < 0 || y >= TAILLE) {
3         return;
4     }
5     if (grilleVisible[x][y] != '-') {
6         return;
7     }
8     if (grille[x][y] == '*') {
9         grilleVisible[x][y] = '*';
10        return;
11    }
12    int minesAdjacentes = compterMinesAdjacentes(x, y);
13    grilleVisible[x][y] = (minesAdjacentes == 0) ? ' ' : (
14        minesAdjacentes + '0');
15    if (minesAdjacentes == 0) {
16        for (int i = x - 1; i <= x + 1; i++) {
17            for (int j = y - 1; j <= y + 1; j++) {
18                revelerCase(i, j);
19            }
20        }
21    }
22 }
```

Cette fonction révèle l'état d'une cellule à l'utilisateur. Si la cellule contient une mine, elle devient visible. Sinon, elle affiche le nombre de mines dans les cellules adjacentes. Si elle n'a pas de mines adjacentes, elle révèle également les cellules autour d'elle.

Fonction 7: Vérifier la victoire

```

1 int verifierVictoire() {
2     for (int i = 0; i < TAILLE; i++) {
3         for (int j = 0; j < TAILLE; j++) {
4             if (grille[i][j] != '*' && grilleVisible[i][j] == '-')
5                 return 0; // Le jeu continue
6         }
7     }
8     return 1; // Le joueur a gagné
9 }
10

```

Cette fonction vérifie si l'utilisateur a gagné la partie. Elle parcourt chaque cellule de la grille et vérifie si toutes les cellules sans mine ont été révélées. Si c'est le cas, l'utilisateur a gagné.

Fonction 8: Fonction principale

```

1 int main() {
2     int finDuJeu = 0;
3     int n;
4     int difficulte;
5
6     printf("Choose the difficulty (1-3): ");
7     scanf("%d", &difficulte);
8
9     do {
10         initialiserGrille();
11         placerMines(difficulte);
12         finDuJeu = 0;
13
14         while (!finDuJeu) {
15             afficherGrille();
16             int x, y;
17             printf("Enter the coordinates (x y) of the cell to
18 reveal/flag: ");
19             if (scanf("%d %d", &x, &y) != 2 || x < 0 || x >= TAILLE
20 || y < 0 || y >= TAILLE) {
21                 printf("Invalid coordinates.\n");
22                 continue;
23             }
24
25             char action;
26             printf("Reveal (r) or place/remove a flag (d)? ");
27             scanf(" %c", &action);
28
29             if (action == 'r') {
30                 if (grille[x][y] == '*') {
31                     printf("Game Over\n");
32                     finDuJeu = 1;
33                 }
34             }
35         }
36     } while (finDuJeu == 0);
37 }

```

```

31         } else {
32             revelerCase(x, y);
33             if (verifierVictoire()) {
34                 printf("Congratulations, you won!\n");
35                 finDuJeu = 1;
36             }
37         }
38     } else if (action == 'd') {
39         basculerDrapeau(x, y);
40     } else {
41         printf("Unrecognized action.\n");
42     }
43 }
44
45 printf("\t1- Play Again.\n");
46 printf("\t2- Exit.\n");
47 printf("Choice: ");
48 }

```

La fonction principale 'main' est le point d'entrée du programme. Elle initialise la grille, place les mines, puis entre dans une boucle de jeu où elle affiche la grille, demande à l'utilisateur de révéler ou de marquer une cellule, et vérifie si l'utilisateur a gagné. Si l'utilisateur révèle une mine, le jeu se termine. Sinon, le jeu continue jusqu'à ce que toutes les cellules sans mine soient révélées.