DM langage C

Thématique : le démineur.

Objectif du jeu : trouver les mines cachées dans une matrice de cases.

Le joueur peut soit placer un drapeau (indiquant une mine), soit mettre le pied sur la case … pour tester la présence d'une mine.

Le jeu commence par la pose d'un pied sur le terrain… si il y a une mine a cette position, fin du jeu!

Sinon la case est découverte et le jeu affiche le nombre de mines qui sont "vues" par la case, c'est-à-dire, le nombre de mines présentes sur les 8 cases adjacentes.

Si il y a aucune mines présentes sur les 8 cases adjacentes le jeu automatiquement découvre alors toutes les cases adjacentes.

Description des deux fonctions principales:**P** poser le pied quelque part (découvrir la case), **D** placer un drapeau (ou le retirer).

- P(x,y) détruira la case x,y.
- P sur une mine entraîne une défaite immédiate.
- P sur un drapeau est neutralisé (aucune action).

- P sur une case 0 (sans voisine) entraînera automatiquement la destruction des 8 cases adjacentes, évitant à l'utilisateur de cliquer de nombreuses fois pour découvrir des chiffres. OPTION: comportement récursif.
- D(x,y) sur une case non découverte placera ou enlèvera un drapeau.

Pour tout ce qui est format de fichier (des questions plus bas sur les entrées sorties). Le format de stockage du fichier est :
trois entiers qui définissent la taille du jeu: hauteur, largeur, nombre de mines, puis l'état actuel du jeu (si il n'y pas d'autre entiers dans le fichiers cela veut dire que le tirage aléatoire de la grille n'a pas été fait). Sinon il y a **hauteur** ligne de **largeur** entiers. Par exemple voici un fichier correct.

5 5 5 # trois entiers : hauteur largeur nombre de mines
0 9 9 9 1 # première ligne
0 3 -9 0 0
0 1 1 0 0
0 0 0 0 0
0 0 0 0 9 # Dernière ligne

Les entiers contenu sur la grille indiquent les informations suivantes:

Si la partie n'est pas finie :
    Si une case est pas découverte :
        9 si il y a une mine
        -9 si il y a une mine et un drapeau
        -10 si il y a un drapeau
        0 sinon
    Si une case est découverte:
        1 - 8 nombre de mines autour de la case.
        -11 pour 0 mines
Si la partie est finie (le joueur est mort en découvrant une mine …)
    Toutes les cases sont découvertes, et 10 indique une mine ayant explosé.

—----
Objectif du devoir

produire une commande "mines" qui accepte les paramètres suivants

```
mine [-a seed][-j h l m][-u][filename][-3]

option j : paramètres de création aléatoire d'un jeu avec
h = hauteur, l= largeur, m= nombre de mines.
```

```
Si (m>=l*h) le jeu signale une erreur.
option a: permet d'initialiser le générateur aléatoire avec
l'entier seed.
option filename:
Lecture du fichier filename (mines.ga en cas d'absence).
Si le fichier n'existe pas il faut utiliser les paramètres de
l'option j
ou les valeurs par défaut 10 10 10.
option u: permet d'indiquer que le jeu doit être sauvegardé à
chaque coup.
option 3: la version 3D.
```

Ensuite la fenêtre du jeu apparaît et il est possible avec la souris d'interagir et de jouer au jeu. La fenêtre doit contenir les boutons recommencer, quitter.

Structures de données: Vous pouvez utiliser cette structure de données.

```
typedef
struct _game {
int width;
int height;
int mines;
int **terrain;
int termine;
} Game;
```

Vous avez le choix du label et des noms de champs.
Mais si vous en changez, alors préparez un texte expliquant vos motivations et ajoutez au dossier que vous devez rendre: “motivation.md” .

Modalités techniques (Cahier des charges techniques):

1) en langage C avec clang comme compilateur.
2) en utilisant la librairie graphique libMLV: http://www-igm.univ-mlv.fr/~boussica/mlv/api/French/html/documentation.html
3) Livrable:
   une archive zip contenant un répertoire avec le nom de famille des deux binômes, dans ce répertoire:
   - minesweeper.c le code du programme
   - readme.md votre compte rendu de développement (difficultés, découvertes, plaisir etc)
   - (option) motivation.md
   - (option) jeu1.ga à jeu99.ga des exemple des jeux chargeables (pour montrer des trucs sympa à votre chargé de tp qui à beaucoup de projets à corriger….)
4) l’archive est à rendre sur le https://elearning.univ-eiffel.fr/ par un des deux membres du binôme.

Modalités organisationnelles (Exigences clients):
Les binômes sont constitués par les chargés de TP en présentiel ! à condition que les deux membres aient au moins la moitié des points sur les 2 feuilles d’exercices sur Premierlangage http://pl.u-pem.fr.