Cours de programmation séquentielle

Démineur

1 Buts

- Utilisation de types énumérés, de tableaux et de fonctions
- Affichage en mode texte
- Modularisation du code
- Récursivité

2 Énoncé

Le but de ce travail est d'implémenter le jeu *démineur* en `C` (vous pouvez jouer à une version en ligne sur https://minesweeperonline.com par exemple)

Figure 1: Un exemple du jeu démineur.

3 Règles du jeu

Le jeu du *démineur* se joue seul. Un nombre fixé de bombes cachées se trouvent sur une aire de jeu rectangulaire. A chaque étape, le joueur peut soit ouvrir une

case, soit placer un drapeau là où il pense que se trouve une bombe. Lorsqu'une case est ouverte, le nombre de bombes dans le voisinage de cette case est affiché. Si ce nombre est 0, alors les 8 voisins de la case sont également ouverts. Le processus est répété pour tout voisin n'ayant aucune bombe dans ses 8 cases adjacentes, puis ceci est propagé aux voisins, etc. Si une case contenant une bombe est ouverte, le jeu se termine sur une défaite et l'aire de jeu est affichée avec toutes les cases ouvertes. Si le joueur réussit à ouvrir toutes les cases sans bombe, alors il gagne et la partie se termine. Le temps écoulé doit s'afficher à la fin de la partie. A noter qu'il ne doit pas être possible de placer plus de drapeaux que de bombes et que le joueur peut toujours décider d'enlever un drapeau d'une case s'il pense s'être trompé sur la présence d'une bombe.

4 Utilisation

Votre programme doit se lancer à la ligne de commande en deux modes. Si votre exécutable se nomme `demineur`, on le lancera comme suit:

```
$ ./demineur C L M
```

où `C` et `L` sont le nombre de colonnes et de lignes respectivement, et `M` le nombre de mines. Dans ce mode de jeu les mines sont placées aléatoirement. La seconde version permet de placer les mines à des endroits prédéterminés

```
$ ./demineur C L c1 l1 c2 l2 ...
```

où `C` et `L` sont le nombre de colonnes et de lignes respectivement, et (`c1`, `l1`), (`c2`, `l2`), … sont les positions (ligne, colonne) des mines.

Le jeu ensuite demande à l'utilisateur trois actions différentes:

- `D`: on place un drapeau,
- `O`: on ouvre une case,
- `Q`: on quitte le jeu et on affiche le temps de jeu.

Dans le cas où l'utilisatrice a joué `D` ou `O`, on demande encore la position du drapeau à déposer ou de la case à ouvrir sous la forme de deux entiers `l`et `m`.

L'affichage se fait uniquement en mode textuel dans le terminal. L'espace vide est un (espace), la bombe un `B`, le drapeau un `D`, et le nombre de bombes voisines le nombre correspondant.

5 Structure de données

Chaque case de votre jeu est d'un type énuméré nommé `square_t` qui consiste en les variantes `EMPTY`, `BOMB` ou `FLAG`. Ainsi votre tableau de jeu est un tableau à deux dimensions dont chaque case est du type `square_t`. On vous propose d'utiliser encore deux autres tableaux. Les cases du premier tableau contiendront le nombre de bombes voisines à une case. Le second tableau stockera si une case du jeu a été ouverte ou non.

Afin de simplifier, nous utiliserons les `VLA` (ou variable length array) et le tableau de jeu sera un tableau de taille `L x C` (attention à ne pas choisir des `C` ou `L` trop

grands, car cela pourrait avoir des effets désastreux lors de l'exécution de votre programme). Ainsi un tableau de tableau de type `square_t` se déclare comme

```
square_t board[L][C];
```

6 Remarques

6.1 Le libre-service

Vous êtes fortement encouragés à aller au libre service pour obtenir des conseils et de l'aide pour avancer votre TP.

6.2 Fonctions

Il est indispensable que vous définissiez des fonctions. Réfléchissez bien à l'organisation de votre programme avant de commencer à coder.

6.3 Mesure du temps d'exécution

La mesure du temps d'exécution de votre programme se fait grâce à la librairie `<time.h>`. Un exemple d'utilisation est donné par les lignes suivantes:

```
struct timespec start, finish;
clock_gettime(CLOCK_MONOTONIC, &start);
// Code dont on veut mesurer le temps d'exécution
clock_gettime(CLOCK_MONOTONIC, &finish);
float elapsed = (finish.tv_sec - start.tv_sec);
printf("elapsed seconds = %f\n", elapsed);
```