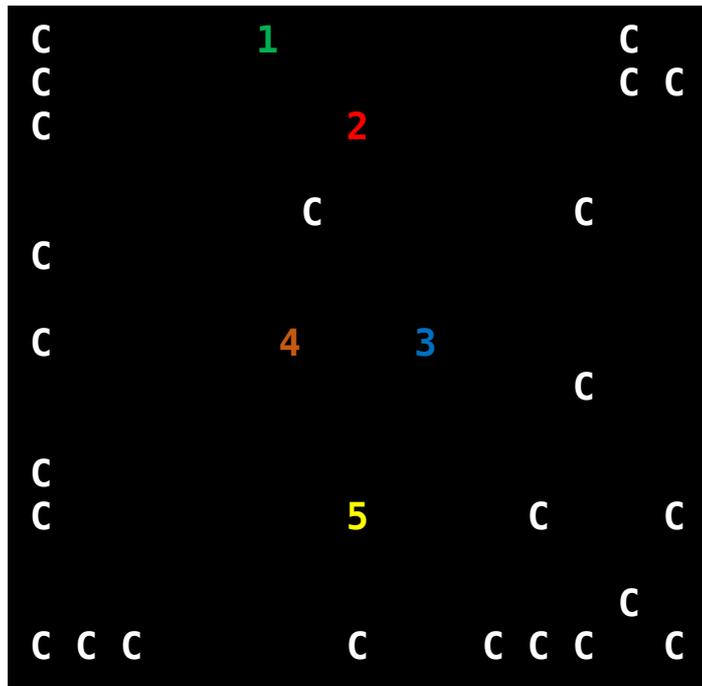


Gare aux chats

Présentation du jeu

Le but du jeu est, en tant que souris, de survivre le plus longtemps possible dans un monde infesté de chats.



Les chiffres de 1 à 5 représentent les souris.

Votre objectif

Réaliser le code d'une souris. L'interface du monde est fournie !

Prérequis

- Savoir programmer en C (fonctions et macro-constante)

Règles du jeu

Le jeu se joue tour par tour.

- L'ajout aléatoire de chat (dans les coins)

A chaque tour, il y a une chance sur 4 pour qu'un chat apparaisse dans un coin. Les probabilités sont indépendantes pour chaque coin. Il peut donc il y avoir entre 0 et 4 chats en plus par tour.

- Le mouvement des chats

Les chats se déplacent de manière aléatoire. Ils peuvent aller en haut, en bas, à droite, à gauche ou rester sur place.

- Le déplacement des souris

En commençant par la souris1, chaque souris donne par appel à la fonction codée par le joueur une direction (UP, DOWN, RIGHT, LEFT, SPOT). Si la fonction codée par le joueur renvoie autre chose ou si le déplacement n'est pas possible (sortie du monde) alors la souris ne se déplacera pas.

Si un chat se déplace sur une souris alors la souris est mangée et le joueur est éliminé (ajout aléatoire compris).

PS : Les chats ne peuvent en aucun cas être détruits. Si le joueur dans la fonction codée modifie le monde (tableau 2D) alors la souris correspondante sera automatiquement disqualifiée.

Ce que vous avez besoin de savoir

Le monde correspond à un tableau 2D de hauteur **HEIGHT** et de largeur **WIDTH**.

Dans ce monde :

- les souris sont représentées par **SOURIS1, SOURIS2, SOURIS3 ...**
- les cases vides sont représentées par **CASE_VIDE**
- les chats sont représentés par **CHAT**

L'exécution du jeu se réalise avec **5 souris** dans le monde.

***macro constante**

Votre code

Choisissez un numéro de souris entre 1 et 5 !

Votre fonction doit avoir comme prototype : `int sourisX(int** monde);`

Le retour de la fonction est une macro constante qui correspond au mouvement que doit réaliser la souris. Vous pouvez retourner **RIGHT, LEFT, UP, DOWN, SPOT** (pour rester sur place). Si le retour est différent, la souris ne bougera pas.

Exemple de fonction :

Je choisis le numéro 3. Je code donc le programme pour la souris 3.

Pierrick Delrieu

```
int souris3(int** monde)
{
    int mouvement = 1 + (rand() % 5);

    if (mouvement == 1){
        return RIGHT;
    }
    else if (mouvement == 2){
        return LEFT;
    }
    else if (mouvement == 3){
        return UP;
    }
    else if (mouvement == 4){
        return DOWN;
    }
    else{
        return SPOT;
    }
}
```

Vous devez aussi réfléchir à la position initiale que devra avoir votre souris sur le monde. Vous pouvez coder une sous fonction si et seulement si le prototype est de la forme : sourisX_nomfonction();

Vous avez à votre disposition la fonction : int case_valide_souris(int i, int j, int** monde); qui renvoie 1 si la case de coordonnées (i,j) est dans le monde, elle renvoie 0 sinon.

PS : Si votre fonction fait crasher le programme, vous serez bien évidemment disqualifié.

Petite "aide" (info)

L'ajout et le déplacement des souris dans le tour se fait avant le déplacement des chats.

Si vous êtes motivé

Si vous voulez tester votre souris, coder vous-même l'interface du monde (en mode console).

UNE FOIS VOTRE FONCTION SOURIS TERMINÉ

(Envoyez-la moi pour que je la teste avec les autres)

Je vous transmets les résultats à chaque test.

Envoyez-la moi à l'adresse suivante : pierrick.delrieu@efrei.net

N'oubliez pas de me transmettre la position initiale de votre souris en plus de la fonction.