

Documents autorisés : polycopiés de cours

Durée : 2h

Barème à titre indicatif.

« Mots mêlés »

Objectif du jeu : MOTS MÊLES est un jeu de mots cachés classique.

Un thème, une liste de mots, dont un mystère, à retrouver dans une grille de lettres. Les mots peuvent être inscrits horizontalement, verticalement et en diagonale, de gauche à droite et de droite à gauche.

Le travail demandé consiste à écrire :

- dans un fichier ***mots_meles.h***, les déclarations nécessaires pour utiliser cette bibliothèque,
- dans un fichier ***mots_meles.c***, les fonctions spécifiées ci-dessous,
- un programme qui permet le déroulement du jeu.



Question 1 :

(3 points)

Ecriture du fichier « ***mots_meles.h*** » composé de :

1. Définir les macros ***TAILLEGRILLE*** et ***LGDISCO*** pour qu'ils correspondent à l'initialisation faite dans le programme principal (cf. page jointe).
2. On disposera d'une fonction ***saisirCoup()***, qui permet de saisir le coup à jouer, et qui demande à l'utilisateur la direction dans laquelle il sélectionne un mot en utilisant les points cardinaux.
Ecrire le type énuméré ***DIRECTION*** dont les valeurs sont E=0, W=1, N=2, S=3, NE=4, SE=5, NW=6, SW=7.
3. Définir le type ***typeMot*** basé sur une structure composée de 3 champs : un tableau de capacité ***TAILLEGRILLE*** de caractères, un entier pour la longueur du mot, un entier qui servira de booléen pour savoir si le mot a été trouvé.

Question 2 :

(12 points)

Ecriture, dans le fichier « ***mots_meles.c*** », de la définition des fonctions suivantes :

1. ***int verifCoord(int x, int y, int tailleGrille)*** qui vérifie que les coordonnées x et y passées en paramètre sont valides.
2. La fonction ***typeCoord calculeDirection(DIRECTION dir)*** qui selon la ***DIRECTION dir*** (cf. type énuméré ci-dessus) choisie retourne une variable de type ***typeCoord*** correspondant à un vecteur (E=0,1 - W=0,-1 - N=-1,0 - S=1,0, NE=-1,1 - SE=1,1 NW=-1,-1 - SW=1,-1).

3. *int compareMots(char mot1[TAILLEGRILLE], int lgMot1, char mot2[TAILLEGRILLE], int lgMot2)* qui retourne 1 si *mot1* et *mot2* sont identiques et 0 sinon.
4. La fonction *int rechercheMot(typeMot motsATrouver[LGDICO], char mot[TAILLEGRILLE], int lgMot)* qui cherche l'indice du mot *mot* dans le tableau des mots à trouver. S'il est présent cet indice est retourné, sinon c'est la valeur **LGDICO** qui est retournée.
NB : chaque case du tableau *motsATrouver* contient à la fois un mot et sa taille.
5. La fonction *int extraireMot(char grille[TAILLEGRILLE][TAILLEGRILLE], int x, int y, int dx, int dy, char mot[TAILLEGRILLE], int lgMot)* qui copie dans *mot* les *lgMot* lettres se trouvant à partir de l'indice *x,y* dans la direction *dx, dy* de la matrice *grille*.
NB : *dx* et *dy* prennent leurs valeurs parmi E=0,1 - W=0,-1 - N=-1,0 - S=1,0, NE=-1,1 - SE=1,1 NW=-1,-1 - SW=1,-1
6. La fonction d'affichage de la grille et de la liste de mots à trouver *void afficheGrille(char grille[TAILLEGRILLE][TAILLEGRILLE], int grilleTrouve[TAILLEGRILLE][TAILLEGRILLE], typeMot motsATrouver[LGDICO])*. La matrice *grilleTrouve* contient des booléens permettant de savoir si les lettres correspondantes de la matrice *grille* font partie de mots déjà trouvés, si c'est le cas, elles seront affichées entre parenthèses. A côté de la grille sera également affiché les mots restants à trouver.
7. La fonction *int toutTrouve(typeMot motsATrouver[LGDICO])* qui retourne 1 si tous les mots ont été trouvés, 0 sinon.

Question 3 :

(5 points)

Dans le fichier de la page suivante, qui permet de jouer au jeu des mots mêlés, **corriger le programme** quand c'est nécessaire.

Exemple de déroulement d'une partie :

```
mots mêlés :
X 0 1 2 3 4
0 A I N T J      IF
1 R F H S E      INT
2 K A H N U      CHAR
3 P A H O T      CASE
4 E S A C Y      CONST
Début du mot (x,y) : 0,1
direction (E=0, W=1, N=2, S=3, NE=4, SE=5,
NW=6, SW=7) : 0
longueur : 3

Bravo, INT trouvé !!!

mots mêlés :
X 0 1 2 3 4
0 A (I) (N) (T) J      IF
1 R F H S E
2 K A H N U      CHAR
3 P A H O T      CASE
4 E S A C Y      CONST
continue ? (oui 1/ non 0) : 1
Début du mot (x,y) : 4,3
direction (E=0, W=1, N=2, S=3, NE=4, SE=5,
NW=6, SW=7) : 2
longueur : 5
```

Bravo, CONST trouvé !!!

```
mots mêlés :
X 0 1 2 3 4
0 A (I) (N) (T) J      IF
1 R F H (S) E
2 K A H (N) U      CHAR
3 P A H (O) T      CASE
4 E S A (C) Y
continue ? (oui 1/ non 0) : 1
Début du mot (x,y) : 4,3
direction (E=0, W=1, N=2, S=3, NE=4, SE=5,
NW=6, SW=7) : 6
longueur : 4
```

Bravo, CHAR trouvé !!!

```
mots mêlés :
X 0 1 2 3 4
0 A (I) (N) (T) J      IF
1 (R) F H (S) E
2 K (A) H (N) U
3 P A (H) (O) T      CASE
4 E S A (C) Y
continue ? (oui 1/ non 0) : 0
```

NOM :
Prenom :

IS de langage C – 2013-2014
GM3

Fichier « mots_meles.h »

```
#include <stdio.h>

/* macro pour la taille de la grille */
_____

/* macro pour le nombre de mots à trouver */
_____

/* type énuméré pour choisir la direction : E=0, W=1, N=2, S=3, NE=4, SE=5, NW=6, SW=7 */
_____

/* type coordonnées */
typedef struct
{ int x, y; } typeCoord;

/* type pour définir les coups à jouer */
/* x et y sont l'origine du mot sélectionné dans la grille */
/* dx et dy sont la direction dans laquelle on sélectionne le mot : 0, -1 ou 1 pour chacun */
/* lg est la longueur du mot sélectionné */
typedef struct
{ int x, y, dx, dy, lg; } typeCoup;

/* typeMot pour définir les mots à trouver ainsi que leur longueur et s'ils ont été trouvés ou non */
_____
_____
_____
_____

int verifCoord(int x, int y, int tailleGrille);
/* retourne vrai (1) si les coordonnées x, y sont dans la grille, faux (0) sinon */

typeCoord calculeDirection(DIRECTION dir);
/* direction (E=0,1 - W=0,-1 - N=-1,0 - S=1,0, NE=-1,1 - SE=1,1 NW=-1,-1 - SW=1,-1) */

int compareMots(char mot1[TAILLEGRILLE], int lgMot1, char mot2[TAILLEGRILLE], int lgMot2);
/* retourne 1 si les mots sont identiques, 0 sinon */

int rechercheMot(typeMot motsATrouver[LGDICO], char mot[TAILLEGRILLE], int lgMot);
/* retourne l'indice du mot s'il est trouve, LGDICO sinon */

int extraireMot(char grille[TAILLEGRILLE][TAILLEGRILLE], int x, int y, int dx, int dy, char mot[TAILLEGRILLE], int lgMot);
/* extrait et stocke dans mot la suite de lgMot caractères commençant à l'indice x,y dans la direction dx,dy
retourne la longueur du mot extrait (au plus lgMot caractère) */

void motTrouve(int grilleTrouve[TAILLEGRILLE][TAILLEGRILLE], int x, int y, int dx, int dy, int lg);
/* met à jour les booléens du mot trouvé dans la grille */

void afficheMot(char mot[TAILLEGRILLE], int lgMot);
/* affiche le mot */

void afficheGrille(char grille[TAILLEGRILLE][TAILLEGRILLE], int grilleTrouve[TAILLEGRILLE][TAILLEGRILLE],
typeMot motsATrouver[LGDICO]);
/* affiche une grille dont les mots trouvés sont entourés, ainsi que la liste des mots restants à trouver */

int toutTrouve(typeMot motsATrouver[LGDICO]);
/* retourne 1 si tous les mots ont été trouvés, 0 sinon */

typeCoup saisirCoup();
```

Fichier « mots_melesMain.c »

```
#include <stdio.h>
#include "IS_MotsMeles.h"

int main () {

    int i, j;
    int lg;
    int encore;
    typeCoup coup;

    char grilleJeu[TAILLEGRILLE][TAILLEGRILLE]={ {'A', 'T', 'N', 'T', 'J'},
                                                    {'R', 'F', 'H', 'S', 'E'},
                                                    {'K', 'A', 'H', 'N', 'U'},
                                                    {'P', 'A', 'H', 'O', 'T'},
                                                    {'E', 'S', 'A', 'C', 'Y'} };

    typeMot motsATrouver[LGDICO]={ {2, {'T', 'F'}, 0},
                                     {3, {'T', 'N', 'T'}, 0},
                                     {4, {'C', 'H', 'A', 'R'}, 0},
                                     {4, {'C', 'A', 'S', 'E'}, 0},
                                     {5, {'C', 'O', 'N', 'S', 'T'}, 0} };

    int grilleTrouve[TAILLEGRILLE..TAILLEGRILLE];
    char motATrouver[TAILLEGRILLE];

    for (i=0, i<TAILLEGRILLE, i++)
        for (j=0; j<TAILLEGRILLE; j++) ;
        grilleTrouve[i][j]=0;

    afficheGrille(grilleJeu, grilleTrouve, motsATrouver);

    do {
        coup=saisirCoup();

        lg=extraireMot(grilleJeu, coup.x, coup.y, coup.dx, coup.dy, motATrouver, coup.lg);

        i=rechercheMot(motsATrouver, motATrouver, lg);
        if (i<LGDICO)
        {
            printf("\nBravo, ");
            afficheMot(motATrouver, lg);
            printf(" trouvé !!!\n\n");

            motTrouve(grilleTrouve, coup.x, coup.y, coup.dx, coup.dy, lg);
            motsATrouver[i].trouve=1;
        }
        else afficheMot(motATrouver, lg);
        printf(" pas dans la grille\n");

        afficheGrille(grilleJeu, grilleTrouve, motsATrouver);

        if (toutTrouve(motsATrouver)) {
            printf("BRAVO !!! Vous avez trouvé tous les mots\n");
            encore=0;
        }
        else { printf("continue ? (oui 1/ non 0) : ");
               scanf("%f", &encore);
            }
    } while (encore!=0);

}
```