DS GM3

Informatique : Langage C

Kotowicz Jean-Philippe
(jean-philippe.kotowicz@insa-rouen.fr)

Documents autorisés : polycopiés de cours
Durée : 3h
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)**

Dans le fichier « ***mots_meles.h*** » (cf. page jointe) compléter :

1. la macro ***max(a, b)***.
2. les types des champs du type ***typeJeu***.

Question 2 : **(8 points)**

Ecrire la définition des fonctions suivantes (cf. déclaration et commentaires dans ***mots_meles.h***) :

1. ***coupValide*** qui vérifie que le coup passé en paramètre est valide (les coordonnées initiales et finales sont dans la grille et sur la même ligne, la même colonne ou la même diagonale).
2. ***saisirCoup*** qui demande à l'utilisateur les coordonnées initiale et finale de son coup et retourne le coup dès qu'il est valide.
3. ***extraireMot*** qui construit et retourne la *chaîne de caractère* du mot se trouvant entre les coordonnées initiale et finale du coup joué.
4. ***toutTrouve*** qui retourne 1 si tous les mots ont été trouvés, 0 sinon.

Tournez la page SVP →→→

Question 3 : allocation dynamique **(2 points)**

Ecrire la fonction ***char ** allolueGrilleChar(int tailleGrille)*** qui alloue et retourne une grille de *tailleGrille* par *tailleGrille* caractères. Cette fonction retourne NULL en cas de problème d'allocation.

Question 4 : affichage du jeu **(3 points)**

Ecrire la fonction ***void afficheJeu(typeJeu unJeu)***. Cet affichage doit être tel qu'il apparaît dans l'exemple de déroulement de partie ci-dessous.

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
                             JEU

Début du mot (x,y) : 0,1
fin du mot (x,y) : 0,3

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
                             JEU

continue ? (oui 1/ non 0) : 1
Début du mot (x,y) : 4,3
fin du mot (x,y) : 4,0

Bravo, CASE 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
  4(E)(S)(A)(C) Y            CONST
                             JEU

continue ? (oui 1/ non 0) : 1

Début du mot (x,y) : 4,3
fin du mot (x,y) : 1,0

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
  4(E)(S)(A)(C) Y            CONST
                             JEU

continue ? (oui 1/ non 0) : 0
```

Question 5 : sauvegarde et chargement d'une partie **(7 points)**

Une grille de mots mêlés sera chargée à partir d'un fichier ***« GRILLE.DATA »*** pour commencer à la jouer ou pour continuer une partie commencée précédemment. Le fichier de données initial correspondant à l'exemple est donné en fin de ce document.

1. Ecrire la fonction ***chargeJeu*** qui ouvre le fichier ***GRILLE.DATA*** et charge successivement la grille de lettres, la grille de booléens et la liste des mots à trouver.
2. Ecrire la fonction ***lireMotsATrouver*** qui charge, à partir du fichier, et retourne la liste des mots à trouver. Attention, il s'agit d'un tableau dynamique de ***typeMot***, chaque mot de cette structure étant une chaîne de caractères.
3. Ecrire la fonction ***sauveJeu*** qui sauvegarde dans le fichier ***GRILLE.DATA*** l'ensemble des éléments de la partie en cours.

Fichier « mots_meles.h »

```
/* macro pour le maximum de a et b */
_____________________________________________________

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

/* type pour définir les coups à jouer */
/* cinit est la coordonnée initiale du mot sélectionné dans la grille */
/* cfinal est la coordonnée final du mot sélectionné dans la grille */
typedef struct { typeCoord cinit, cfinal; } typeCoup;

/* type pour définir les mots à trouver et s'ils ont été trouvés ou non */
typedef struct {
char *mot;     // chaine de caractère pour le mot
int trouve;    // booléen pour savoir s'il a été trouvé ou non
} typeMot;

/* type pour définir un jeu de mots mêlés */
typedef struct {
       __________ tailleGrille;            // taille de la grille
       __________ grille;                  // matrice dynamique de caractères
       __________ grilleTrouve;            // matrice dynamique de booléens
       __________ tailleDico;              // nombre de mots à trouver
       __________ motsATrouver;            // tableau dynamique des mots à trouver
} typeJeu;

/* retourne 1 si un coup est valide,
   c'est-à-dire que les 2 coordonnées sont alignées et dans la grille */
int coupValide(typeCoup coup, int tailleGrille);

/* retourne un coup valide saisi par l'utilisateur */
typeCoup saisirCoup(int tailleGrille);

/*  retourne l'indice du mot s'il est trouve, nbMots sinon */
int rechercheMot(typeMot *motsATrouver, int nbMots, char *mot);

/*  extrait et retourne le mot entre les indices cinit et cfinal du coup
    On suppose le coup valide */
char * extraireMot(char **grille, typeCoup coup);

/*  met à jour les booléens du mot trouvé dans la grille
    On suppose le coup valide */
void motTrouve(int **grilleTrouve, typeCoup coup);

/* retourne 1 si tous les mots sont trouvés, 0 sinon */
int toutTrouve(typeMot *motsATrouver, int nbMots);

/*  affiche une grille dont les mots trouvés sont entourés, ainsi que la liste des mots
restants à trouver */
void afficheJeu(typeJeu unJeu);

/* alloue et retourne une grille de tailleGrille x tailleGrille caractères */
char ** alloueGrilleChar(int tailleGrille);

/* alloue et retourne une grille de tailleGrille x tailleGrille entiers */
int ** alloueGrilleInt(int tailleGrille);

/* lit dans un fichier une grille de mots mélés */
char ** lireGrille(FILE *fic, int *tailleGrille);

/* lit dans un fichier les mots à trouver dans la grille de mots mélés */
typeMot * lireMotsATrouver(FILE *fic, int tailleGrille, int *tailleDico);

/* charge à partir d'un fichier une grille de mots mélés
   ainsi que la liste des mots à trouver */
int chargeJeu(typeJeu *unJeu);

/* sauvegarde une partie dans un fichier */
int sauveJeu(typeJeu unJeu);
```

Tournez la page SVP →→→

Fichier « mots_melesMain.c »

```
#include <stdio.h>
#include "Exam_MotsMeles.h"
int main () {
    int i, j;
    int lg;
    int encore=1;
    typeJeu jeu;
    typeCoup coup;
    char *motATrouver;

    if (chargeJeu(&jeu)==0)
        printf("Pas de grille\n");
    else
        {
            afficheJeu(jeu);

            if (toutTrouve(jeu.motsATrouver, jeu.tailleDico))
                    printf("BRAVO !!! Vous avez trouvé tous les mots\n");
            else
               { do {
                    coup=saisirCoup(jeu.tailleGrille);

                    motATrouver=extraireMot(jeu.grille, coup);
                    i=rechercheMot(jeu.motsATrouver, jeu.tailleDico, motATrouver);
                    if (i<jeu.tailleDico)
                       {
                        printf("\nBravo, %s trouvé !!!\n\n", motATrouver);

                        motTrouve(jeu.grilleTrouve, coup);
                        jeu.motsATrouver[i].trouve=1;
                       }
                    else printf("%s pas un mot à trouver\n", motATrouver);

                    afficheJeu(jeu);

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

                sauveJeu(jeu);
               }
        }
}
```

Fichier « GRILLE.DATA » initial

```
[TAILLEGRILLE=5]
AINTJ
RFHSE
KAHNU
PAHOT
ESACY
[TROUVE=0]
[TAILLEDICO=6]
IF 0
INT 0
CHAR 0
CASE 0
CONST 0
JEU 0
```

Fichier « GRILLE.DATA » d'une partie en cours

```
[TAILLEGRILLE=5]
AINTJ
RFHSE
KAHNU
PAHOT
ESACY
[TROUVE=1]
0 1 1 1 0
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
1 1 1 1 0
[TAILLEDICO=6]
IF 0
INT 1
CHAR 1
CASE 1
CONST 0
JEU 0
```