Langage C

Programmation séparé

L2 Mathématique et Informatique

Université de Marne-la-Vallée

Programmation sur plusieurs fichiers

- Évite de manipuler de trop gros fichiers.
- Facilite la réutilisabilité.
- Facilite les tests.
- Permet la compilation séparée (utilitaire make).

Principe

Le code est réparti par thèmes entre plusieurs fichiers (`.c`).
Un fichier principal contient la fonction `main`.
Un fichier peut utiliser un type ou une fonction définie dans un autre fichier. Pour cela, le compilateur doit connaitre le type ou le prototype de la fonction.
Cette information est contenue dans les fichiers `.h` qui seront inclus partout où cette information est utilisée.
On n'inclut **jamais** de fichiers contenant du code. C'est le compilateur qui trouvera dans quel fichier est définie la fonction appelée.

Exemple : `MotsMeles`

J'ai considéré que ce programme était formé de:

- la partie principale (fichier `motMeles.c`) s'occupe de la gestion des options, de l'ouverture et fermeture des fichiers de sauvegarde, du lancement des fonctions de creation d'un problème ou de lancement du jeu;
- la partie utilisant la librairie graphique MLV (fichier `graphique.c`);
- la partie manipulant les caractères et chaine (fichier `ascii.c`);
- la partie assurant la manipulation du plateau de jeu (fichier `plateau.c`);

La structure Plateau

Elle contient toutes les informations nécessaires au jeu (taille, listes des mots,...). Elle est définie dans `motsMeles.h`. On trouve également dans ce fichier la définition des macro-constantes.

```
#define N 20
#define MAXMOTS N
#define VIDE '.'
#define ALPHA 'a' /* pour completer*/
typedef struct {
  int longf,largf;/* dimensions de la fenetre*/
  int dimxg,dimyg,dimxd,dimyd;/*coordonnees des coins ex
                                dans la fenetre graphiqu
  int taille;/* cote du plateau*/
  int nbmots;/* nb mots caches*/

  char plateau[N][N];
  char dico [MAXMOTS][N+1];
  char present[N];/*1 si pas raye*/
}Plateau;
typedef int Direction[2];
```

Ces informations sont utilisées dans chacune des 4 parties du programme.
Chaque fichier *machin*`.c` contient donc l'instruction du pré-compilateur:

```
#include "motsMeles.h"
```

pour inclure le fichier situé dans le répertoire de travail.

Les autres .h

Chaque fichier *machin*`.h` contient donc le prototype (et un court descriptif) des fonctions du fichier *machin*`.c` qui sont utilisées par d'autres parties. Le fichier `graphique.h` contient:

```
    #include <MLV/MLV_all.h>
#define TAILLE 40 /* Taille d'un carre*/
#define BORD 10 /*marge*/

/* ouverture de la fenetre graphique
   dessin d'un plateau vide*/
void initFenetre(Plateau *p);
/* dessine le plateau dans la fenetre graphique*/
void Dessine(Plateau p);
/*saisie d'un probleme*/
int remplitgraphique(Plateau *p);
```

Ce fichier est inclus dans `motsMeles.c`.
(*On peut aussi choisir d'indiquer le prototype de toutes les fonctions du fichier*)

Le fichier `graphique.c` utilise:

- les fonctions de la bibliothèque standard
- les fonctions de la bibliothèque `MLV`
- le type `Plateau`
- La constante BORD
- des fonctions de gestion du plateau de jeu
- des fonctions de manipulation de caractères et de chaînes

Il commence par:

```
#include <stdio.h>
#include <MLV/MLV_all.h>

#include "motsMeles.h"
#include "graphique.h"
#include "plateau.h"
#include "ascii.h"
```

La compilation

Dans un premier temps on compile ensembles tous les fichiers:
`gcc motsMeles.c ascii.c plateau.c graphique.c -o mots -lMLV`
Vous verrez au prochain semestre l'utilitaire `make` qui permet la compilation séparée.
En cas de changement d'un fichier, on ne recompile que ce qui a été changé, et,en cascade, ce qui en dépend.