

Crossword

Nathan Mbock

January 29th

1 Introduction

Le but de ce projet est de coder un générateur de mots-croisés en C ou l'utilisateur rentre lui-même les mots. Après avoir affiché le mot croisé, une solution est proposée à l'utilisateur s'il le souhaite.

Pour ce faire, 5 étapes sont à distinguer :

1. La création d'une grille remplie de lettre aléatoire, ainsi que d'une matrice booléenne pour la solution
2. La saisie des mots par l'utilisateur
3. L'insertion dans la grille de chaque nouveau mot
4. L'affiche de la grille
5. L'affichage de la solution si l'utilisateur le souhaite

Le programme sera composé de 3 fichiers :

- main.c, programme principal comportant la suite logique des événements
- crossword.h, librairie des fonctions utilisés dans main.c
- crossword.c, code des fonctions de la librairie crossword.h

Pour compiler l'ensemble la commande utilisée est :

```
Gcc -o crossword crossword.c main.c
```

2 main.c

Le main contient toute l'architecture du programme, il est donc essentiel de le comprendre pour bien comprendre le fonctionnement de l'algorithme. La première étape est de créer les 2 matrices, cela se fait avec la commande

```
char GRILLE[SIZE][SIZE];  
char BOOL[SIZE][SIZE];
```

où SIZE est une valeur entière définie dans crossword.h, ici sa valeur est 20. Maintenant que les 2 variables sont créées, il faut les initialiser. Le module creation() s'en occupe.

Par la suite, l'utilisateur doit indiquer combien de mots il veut rentrer. La variable correspondante est nb et elle est passée par le module nb_en adresse pour modifier sa valeur.

Maintenant que l'on a le nombre de mots que l'on veut insérer une boucle for composé de nb tours de boucles permet de faire les insertion successives des mots dans les matrices.

Il ne reste plus qu'à afficher la GRILLE, puis de demander à l'utilisateur s'il veut la solution et lui afficher si c'est le cas.

3 Conclusion

```

File Actions Edit View Help
~$ gcc -Wall -o ok crossword.c main.c
~(kali@kali)-[~/crossword]
~$ ./ok
Veuillez rentrer le nombre de mots que vous souhaitez
5
Veuillez rentrer un mot
hello
Veuillez rentrer un mot
world
Veuillez rentrer un mot
github
Veuillez rentrer un mot
overleaf
Veuillez rentrer un mot
crossword

T N V W O R L D O C
N G I T H U B V W R
M X I D Q U E J E O
L A J B V R A K A S
X F E V L H B V X S
A L O E E X M U A W
J L A L M W J F I O
W F L C N A D H N R
W O G F F V E S J D
D S H E F M C G J D

Voulez-vous la solution [y/n]?y

0 0 0 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 0 1
0 0 0 0 0 0 1 0 0 1
0 0 0 0 0 1 0 0 0 1
0 0 0 0 1 1 0 0 0 1
0 0 0 1 1 0 0 0 0 1
0 0 1 1 0 0 0 0 0 1
0 1 1 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0

```

FIGURE 1 – SIZE = 10

Le programme fonctionne parfaitement, cependant il est perfectible, je n'ai traité aucun cas de saisie tel qu'un mot de taille supérieure à SIZE. De plus je n'ai pas

réussi à faire une approche dynamique où l'utilisateur choisi les dimensions de la grille. Cela à cause de problème de stockage de mémoire lors des malloc pour créer les matrices.