

Université Toulouse III – Paul sabatier
L2 Informatique
Projet tuteuré

Antoine de ROQUEMAUREL
Fabrice VALLEIX
Groupe 2.2

Dossier

Projet de Boggle

Toulouse, le 8 mai 2013

Table des matières

1	But du document	3
2	Compilation	3
3	Exécution	3
3.1	Les tests fonctionnels	3
3.2	Les tests unitaires	4
4	Documentation	4
5	Gestion de projet	4
5.1	Convention de codage	4
5.2	Logiciel de gestion de projet	4
5.3	Logiciel de versionnement	4

1 But du document

C'est une description de haut niveau du produit, c'est-à-dire l'architecture générale du système, en termes de « modules », de sous modules et de leurs interactions. De plus, chaque module doit être décrit (définition des interfaces et des fonctionnalités générales). Ce document doit en premier lieu asseoir la confiance en la finalité et la faisabilité du produit, et, en second lieu, servir de base pour l'estimation des tâches à effectuer et du calendrier de leur réalisation.

Le « Dossier de Conception Préliminaire » doit également mettre en évidence le plan de tests, en termes de besoins de l'utilisateur, et montrer que l'on peut y satisfaire grâce à l'architecture proposée.

2 Compilation

La compilation du projet se fait à l'aide de l'utilitaire **Make**, ainsi la simple commande **make** à la racine du projet suffit à compiler le projet.

Cependant, afin de pouvoir compiler le projet, il est indispensable de posséder la bibliothèque *Ncurses* sur sa machine, sinon la compilation ne fonctionnera pas.

Il est possible de l'installer avec la commande **apt-get install libncurses5-dev** sur les Linux utilisant le gestionnaire de paquet de Debian.

Les tests quant à eux se compile à l'aide de la commande **make test**, cette commande va compiler puis exécuter tous les tests, cependant si vous ne possédez pas *CUnit* sur votre machine, il est également indispensable de taper la commande suivante afin de signaler au système l'emplacement de la bibliothèque.

```
| LD_LIBRARY_PATH=$LD_LIBRARY_PATH:'pwd' /Cunit/lib && export LD_LIBRARY_PATH
```

3 Exécution

Afin de tester l'application, deux types de tests sont présent dans le projet.

3.1 Les tests fonctionnels

Ces tests sont tous présents dans l'exécutable **./boggle**. Afin d'appeler les différentes fonctionnalités du programme, il est nécessaire de faire passer un paramètre, celui-ci peut prendre la forme d'une de ces trois chaînes de caractères :

-solveur Correspond à la version 1 du projet. Afin d'appeler la version 1 de l'application, l'exécutable doit être appelé à l'aide de l'argument **-solveur**

Dans cette version, une grille carrée de la taille demandée par l'utilisateur est générée, en tenant compte de la fréquence des lettres dans la langue Française. Une fois la grille générée, la position d'une case est demandée à l'utilisateur, l'utilisateur entre donc les deux coordonnées, et tous les mots commençant par cette case seront affichés à l'écran.

Attention, les coordonnées de la grille commencent à zéro.

-texte Afin d'appeler la version 2, l'exécutable doit être appelé à l'aide de l'argument **-text**

Cette version fait appel à la version 1, en effet, au lancement de l'application, il est de nouveau demandé la taille de la grille, ensuite l'intégralité de la grille générée est résolue. Une fois cette étape franchie, l'utilisateur a 3 minutes pour entrer le plus de mots possibles, l'application lui signalant si le mot est accepté ou non, une fois ce temps imparti, la solution est affichée, puis le nombre de points obtenu par le joueur.

-ncurses Afin d'appeler la version 3, l'exécutable doit être appelé à l'aide de l'argument **-ncurses**

Cette version suit le même principe que la version précédente, à la différence près qu'elle utilise la bibliothèque *Ncurses*. Ainsi, la saisie des mots se fait dorénavant avec les touches fléchées du clavier, et espace pour ajouter une lettre au mot. Pour proposer le mot surligné, la touche entrée doit être appuyée. Il est également possible de demander le nombre de mots commençant par la case sélectionnée à l'aide de la touche h.

Une fois les 3 minutes écoulées, les mots proposés par l'utilisateur et le nombre de points obtenus sont affichés, il est proposé à l'utilisateur d'afficher la solution complète.

3.2 Les tests unitaires

4 Documentation

5 Gestion de projet

5.1 Convention de codage

5.2 Logiciel de gestion de projet

5.3 Logiciel de versionnement