

IUT Nice Côte d'Azur - Département Informatique - Année 2008/2009

Charles FOUCO - Léo SOUQUET - Gaël BOOS - Cédric HULIN

PROJET TUTEURÉ

BOOGLE

Rapport

Tutrice : Rosa LECAT

Table des matières

I – Déroulement	3
1. Objectifs	3
2. Le jeu du Boogle	
3. Fonctionnalités supplémentaires	3
II – Réalisation	4
1. Programme principal	4
2. Fonctions	4
2.1 Fonction « regle »	4
2.2 Fonction « jouer »	4
2.2.1 Fonction « matrice »	5
2.2.2 Fonctions « recherche »	6
2.3 Fonction « statistique »	8
2.4 Fonction « erreur »	8
III – Conclusion	9

Première partie

Déroulement du projet

1 Objectifs

Afin de mettre en pratique les connaissances acquises durant le semestre, nous avons pour objectif de réaliser un projet tuteuré.

Ce projet consiste à réaliser un jeu de lettres : le Boogle.

Le langage utilisé pour ce projet est le langage C.

2 Le jeu du Boogle

Le joueur dispose d'une grille de jeu au format 6x6. Cette grille de jeu étant remplie aléatoirement par des lettres.

Le but du jeu est de trouver le maximum de mots dans cette grille.

Pour qu'un mot soit valide il faut qu'il remplisse les conditions suivantes :

- Le mot doit être formé d'au moins 3 lettres
- Le mot doit être constitué de lettres qui se touchent
- Chaque lettre ne peut être utilisée qu'une seule fois pour un même mot
- Le mot doit être à l'horizontale, verticale ou se trouver sur les diagonales

3 Fonctionnalités supplémentaires

Nous avons dans un premier temps choisi de stocker les informations sur une partie jouée. En découle les fonctionnalités suivantes :

- Phase d'inscription : demande du nom et du prénom
- Nouvel onglet dans le menu : Statistique

Dans un second temps, à la demande de notre tutrice nous devons implémenter un dictionnaire.

Nous avons donc choisi de laisser au joueur le choix de jouer avec ou sans dictionnaire lors de son inscription.

Deuxième partie

Réalisation

1 Programme principal

Afin de rendre le code le plus lisible possible nous avons opté pour un programme principal simple et clair.

En effet, le programme principal se résume en 3 points :

- Appel d'une fonction nous affichant un menu
- Appel de la fonction correspondante au choix du joueur
- Gestion d'un choix invalide

2 Fonctions

Comme nous l'avons vu le programme principal appelle tout d'abord une fonction permettant d'afficher un menu.

En fonction du choix du joueur nous allons appeler la fonction correspondante à son choix.

2.1 Fonction « regle »

La fonction prend en paramètre un entier correspondant au choix que l'utilisateur va entrer.

Voici les éléments du menu :

- 1) Règles du jeu
- 2) Jouer
- 3) Statistiques
- 4) Quitter

2.2 Fonction « jouer »

Cette fonction comme son nom l'indique nous permettra de jouer une partie de Boogle.

Avant de lancer la partie nous allons procéder à l'inscription du joueur puis nous allons lui proposer de jouer avec un dictionnaire.

L'enregistrement du joueur étant fini, nous allons générer une grille de jeu grâce à une fonction « *matrice* » puis nous l'affichons au joueur.

Nous proposons alors au joueur de saisir un mot.

Plusieurs vérifications sont faites sur le mot entré :

- Longueur minimale du mot
- Longueur maximale du mot
- Vérification que le mot n'a pas déjà été entré

Si le joueur a décidé de jouer avec un dictionnaire nous allons vérifier que le mot saisi fait bien parti de celui-ci.

Passé toutes les conditions nécessaires à un mot valide nous allons procéder à la recherche du mot dans la grille de jeu grâce aux *fonctions de recherche* que nous avons implémentées.

Un affichage soigné des informations relatives à la partie a été mis au point comme :

- Le nombre de mots trouvés
- Le nombre de points

Puis la fonction enregistrera les informations de la partie dans un fichier texte nommé « *enregistrement.txt* ».

2.2.1 Fonction « *matrice* »

Le but de cette fonction est de générer une matrice carrée de 8x8.

« Pourtant le jeu nous propose une grille de 6x6 ? ».

Oui en effet, cependant pour une question de facilité de recherche du mot dans la grille de jeu nous avons opté de remplir les contours de la matrice avec le caractère spécial '&'.

Afin de remplir la grille de jeu avec des lettres choisies aléatoirement nous utilisons la fonction c « *rand()* »

2.2.2 Fonctions de recherche

Afin de vérifier si le mot saisi par le joueur se trouve dans la grille de jeu nous avons implémenté différentes fonctions de recherche :

- 2 fonctions de recherche verticale
- 2 fonctions de recherche horizontale
- 4 fonctions de recherche en diagonale

« Comment savoir qu'elle fonction appeler ? »

Pour pouvoir appeler la bonne fonction nous devons connaître la trajectoire du mot saisie dans la grille.

Par exemple avec la grille suivante nous savons que le mot « **ARBRE** » se trouve à la verticale (descendante) :

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

Avant d'appeler notre fonction nous allons parcourir la matrice, puis si la lettre rencontrée correspond à la première lettre du mot que le joueur a entré nous allons regarder autour de cette lettre si nous avons une lettre correspondante à notre seconde lettre de notre mot.

Pourquoi me direz-vous ?

Pour connaître la direction ! En effet, il nous suffit de vérifier que 2 lettres qui se touchent dans la grille correspondent aux 2 premières lettres du mot saisi par le joueur pour connaître la direction du mot.

Reprenons l'exemple précédant. Le joueur a saisi le mot « arbre ». Pour savoir que ce mot se trouve dans la grille de jeu nous avons donc parcouru la grille et nous sommes tombés sur la première case du tableau qui correspond à notre première lettre du mot à savoir le A.

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

À partir de là nous allons parcourir les lettres qui se trouvent autour du 'A'.

Un problème se pose me direz-vous ? En effet nous voulons parcourir les 8 cases qui se trouvent autour du 'A', cependant nous n'en avons que 3 !

Si vous avez bien suivi plus haut, nous avons généré une matrice non pas de 6x6 mais de 8x8 en délimitant les contours par le caractère spécial '&'. Et là notre problème est résolu :

&	&	&	&	&	&	&	&
&	A	F	M	A	P	D	&
&	N	C	E	R	S	Z	&
&	U	Y	N	B	C	W	&
&	X	O	P	R	H	G	&
&	D	D	G	E	Z	P	&
&	K	I	Y	T	L	Q	&
&	&	&	&	&	&	&	&

Nous allons donc regarder autour de notre 'A' si nous trouvons un 'R'. On voit sur cet exemple que ce n'est pas le cas.

Nous allons donc continuer à parcourir notre grille jusqu'au prochain 'A' et effectuer le même traitement.

On se retrouve dans le cas en rouge, nous avons nos deux lettres qui correspondent aux 2 premières lettres de notre mot, et nous pouvons donc appeler notre méthode de recherche en verticale pour vérifier si les prochaines lettres ('B' 'R' et 'E') correspondent bien à la suite du mot.

2.3 Fonction « statistique »

Cette fonction permet de visualiser toutes les parties jouées en lisant le contenu du fichier « enregistrement.txt ».

Les informations affichées sont :

- Le nom
- Le prénom
- Le score
- Le nombre de parties jouées

2.4 Fonction « erreur »

Cette fonction a pour but de gérer deux types d'erreur :

- Choix invalide dans le menu
- Ouverture impossible d'un fichier

Lorsque l'on rencontre une de ces erreurs le joueur sera averti par un message puis redirigé vers le menu principal.

Troisième partie

Conclusion

Au final ce projet ne fût pas réellement compliqué niveau programmation, toutes les compétences nécessaires étant acquises.

On notera le fait que les erreurs possibles sont gérées entièrement et que le nombre de voyelles dans la grille de jeu est optimal pour pouvoir trouver un minimum de mots.