Projet IF2B P2022

L'objectif de ce projet est de réaliser un programme permettant de jouer à un jeu de lettres inspiré de Boggle :

- Des lettres sont disposées aléatoirement dans une grille
- Puis, dans un temps imparti, le joueur doit composer des mots à partir de cette grille ; les mots sont composés en reliant des lettres adjacentes (soit horizontalement, soit verticalement, soit diagonalement)
- A la fin de la partie, on reprend la liste des mots réalisés pour compter le score

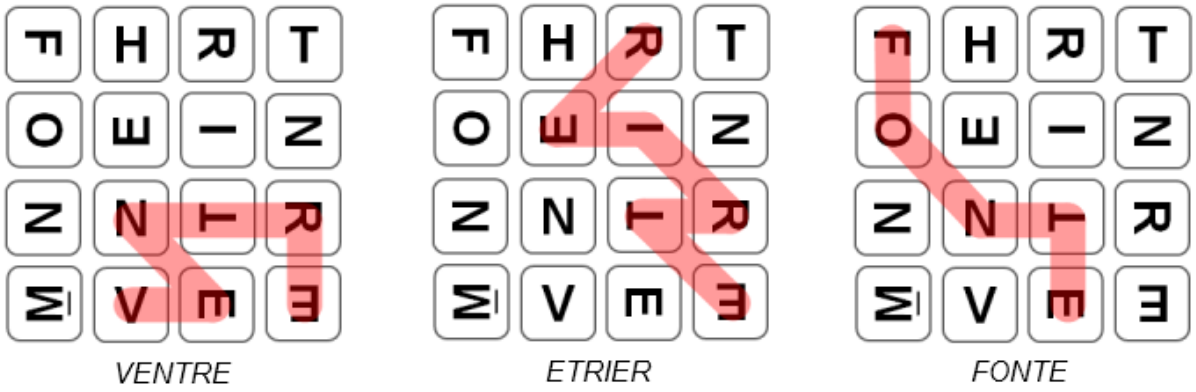

Figure 1 - Différents mots tracés à partir de lettres, contrairement à cette figure, dans le programme à produire les lettres seront affichées verticalement

Réalisation du projet

Le projet sera intégralement réalité en C par **groupes de 2 ou 3 étudiants** du même groupe (mais pas forcément dans les mêmes groupes présentiel/distanciel).

Les livrables attendus sont :

- Le code source (fichiers .c et .h), dûment commenté et documenté, accompagné des instructions de compilation (fichier CMakeLists.txt ou fichier Makefile)
- D'un rapport d'une dizaine de pages présentant le travail réalisé, notamment la structure générale de votre code, les choix réalisés pour le développement du jeu, et le résultat final (en faisant le bilan de ce qui a été réussi et des points d'amélioration éventuels)

Le projet devra être rendu au plus tard le **12/06 à 18h** dans l'espace de dépôt dédié sur Moodle « Dépôt projet P22 ». Tous les fichiers (rapport et sources) seront contenus dans une archive nommée **NOM-BINOME1_NOM-BINOME2_NOM-BINOME3.zip**

Une soutenance de projet sera organisée pendant les séances de TD pour vous permettre de présenter votre réalisation (un PPT et une démonstration du projet seront attendues).

Menu initial

Au démarrage on propose à l'utilisateur un menu affichant trois options :

- Démarrer une partie
- Afficher les meilleurs scores
- Quitter

Selon le choix de l'utilisateur, les actions correspondantes seront réalisées.

Démarrage d'une partie

Avant de démarrer une partie, on demande au joueur plusieurs informations :

- Quelle est la dimension souhaitée pour la grille (de 4x4 jusqu'à 8x8)
- En combien de temps souhaite-t-il réaliser la partie (de 60s à 180s)

Puis une grille composée de lettres aléatoires est générée (voir algorithme en annexe).

Cette grille est affichée à l'utilisateur.

Déroulement d'une partie

Tant que le temps imparti n'est pas écoulé, on demande à l'utilisateur de saisir des mots.

A chaque mot saisi par l'utilisateur, on vérifie que celui-ci est un mot français (en le comparant à une liste exhaustive de tous les mots français, chargée à partir d'un fichier), puis que celui-ci peut être écrit en connectant les lettres affichées dans la grille, la connexion pouvant se faire dans n'importe quel sens.

Si le mot est valide, on l'enregistre dans une liste des mots saisis par l'utilisateur. Sinon, on lui affiche un message d'erreur pour indiquer que le mot est invalide.

Fin de partie

La partie se termine quand le temps imparti est écoulé.

Soit N le nombre de mots trouvés, et soit L_i le nombre de lettres du mot i, alors le score est calculé selon la relation suivante :

$$score = \sum_{i=0}^{N-1} {L_i}^{4/3}$$

On demande à l'utilisateur de saisir son nom puis on enregistre son nom et son score dans un fichier, ainsi que ses paramètres de partie.

Une fois le score sauvegardé, on revient au menu initial.

Affichage des meilleurs scores

L'affichage des meilleurs scores permet de visualiser les utilisateurs ayant réalisé les meilleurs scores à partir du fichier des scores. La liste est triée par score décroissant.

Dans l'affichage des scores, un menu est proposé à l'utilisateur pour :

- Rechercher un utilisateur et afficher son score
- Filtrer et afficher les résultats selon la taille de la grille (on demande alors la taille désirée)
- Revenir au menu principal

Instructions supplémentaires

Le programme devra être réalisé en C et devra être programmé de façon modulaire (division du code en plusieurs modules – fichiers C et fichier H). Le code devra être dûment commenté et documenté.

Des fonctions devront être créées afin de structurer le programme.

Annexe 1 – Génération du tableau de lettres

Afin de simplifier la vérification de l'existence des mots sur le plateau contenant les lettres, nous proposons de poser quelques contraintes concernant la génération aléatoires des lettres dans ce tableau.

L'objectif est que, dans chaque sous-carré de 3 x 3 du tableau, il n'y ait jamais deux fois la même lettre.

Pour cela, commencer à générer 9 lettres aléatoires, toutes différentes, en fonction de la distribution donnée en annexe 2, puis placer ces lettres en haut à gauche du tableau final :

R	**A**	**J**		
F	**I**	**E**		
Z	**B**	**G**		

Ensuite, décaler cette « fenêtre » de 3 x 3 d'une colonne vers la droite, puis générer trois nouvelles lettres en vous assurant que les lettres soient toutes différentes (les trois nouvelles doivent être différentes des six déjà générées).

R	**A**	**J**	**F**	
F	**I**	**E**	**R**	
Z	**B**	**G**	**U**	

Répéter jusqu'à atteindre la droite du tableau, puis recommencer en décalant d'une ligne vers le bas.

R	A	**J**	**F**	**A**
F	I	**E**	**R**	**I**
Z	B	**G**	**U**	**P**

R	A	J	F	A
F	**I**	**E**	R	I
Z	**B**	**G**	U	P
A	**C**	**K**		

Itérer jusqu'à ce que le tableau soit complet.

Annexe 2 – Probabilité des lettres

La probabilité d'apparition des lettres est basée sur la fréquence d'occurrence dans la langue française :

https://fr.wikipedia.org/wiki/Fr%C3%A9quence_d%27apparition_des_lettres_en_fran%C3%A7ais

Les caractères spéciaux et accentués ne sont pas considérés en tant que tel.

Ainsi :

Lettre	Français
a	8,182%
b	0,902%
c	3,349%
d	3,673%
e	16,726%
f	1,067%
g	0,867%
h	0,738%
i	7,587%
j	0,614%
k	0,074%
l	5,462%
m	2,971%
n	7,103%
o	5,802%
p	2,524%
q	1,363%
r	6,700%
s	7,957%
t	7,252%
u	6,318%
v	1,840%
w	0,049%
x	0,427%
y	0,128%
z	0,326%