

Projet C : Mot le plus long

Avant de commencer : la qualité des commentaires, avec notamment la présence des antécédents, des conséquents, des invariants de boucle, les rôles de chacune des fonctions, ainsi que les noms donnés aux variables, l'emploi à bon escient des majuscules et la bonne indentation rentrent pour une part importante dans l'appréciation du travail. Ce projet doit permettre de montrer votre autonomie et votre compréhension tant dans la conception du programme que dans sa réalisation. Enfin, si les codes de plusieurs projets se trouvent être identiques, ou être copiés depuis le web, tous les projets concernés seront immédiatement sanctionnés par un zéro.

1 Le Jeu

L'objectif de ce projet est de doter d'une interface graphique votre programme *mot le plus long* que vous avez programmé au premier semestre.

On rappelle que le jeu du *mot le plus long* oppose deux joueurs qui tirent à tour de rôle et au hasard jusqu'à 9 lettres (voyelles ou consonnes). Chaque joueur essaye de former un mot avec les 9 lettres. Le joueur qui a trouvé le mot le plus long a gagné.

2 Travail à réaliser

Vous devrez développer un programme en langage C qui propose à deux utilisateurs de jouer l'un contre l'autre. Votre programme commence par effectuer le tirage des 9 lettres. Chaque joueur tire alternativement une voyelle ou une consonne.

Les voyelles seront tirées aléatoirement dans un ensemble de 45 voyelles donné par la table de fréquence suivante :

A	E	I	O	U	Y
9	15	8	6	6	1

Les consonnes seront, quant à elles, tirées aléatoirement dans un ensemble de 55 consonnes donné par la table de fréquence suivante :

B	C	D	F	G	H	J	K	L	M
2	2	3	2	2	2	1	1	5	3
N	P	Q	R	S	T	V	W	X	Z
6	2	1	6	6	6	2	1	1	1

Ensuite, chaque joueur propose son mot le plus long. Votre programme devra vérifier que le mot proposé est *valide*, c'est-à-dire qu'il est bien formé à partir des lettres du tirage précédent, qu'il appartient bien au dictionnaire français. Votre dictionnaire devra être placé dans un fichier (ou plusieurs fichiers) qui sera consulté par votre programme. Si le mot n'est pas valide le joueur a perdu.

Si les deux joueurs ont donné deux mots valides, le gagnant est celui qui aura proposé le mot le plus long. Si les mots ont la même taille, il y a égalité.

Ci-dessous, un exemple de déroulement de partie :

Le 1er joueur tire une lettre : voyelle ou consonne (v/c) ? : c

m _ _ _ _ _

```

-----
Le 2ème joueur tire une lettre :  voyelle ou consonne (v/c) ? : v
-----
m o _ _ _ _ _
-----
Le 1er joueur tire une lettre :  voyelle ou consonne (v/c) ? : c
-----
m o p _ _ _ _ _
-----
Le 2ème joueur tire une lettre :  voyelle ou consonne (v/c) ? : v
-----
m o p a _ _ _ _ _
-----
Le 1er joueur tire une lettre :  voyelle ou consonne (v/c) ? : v
-----
m o p a i _ _ _ _ _
-----
Le 2ème joueur tire une lettre :  voyelle ou consonne (v/c) ? : c
-----
m o p a i j _ _ _ _ _
-----
Le 1er joueur tire une lettre :  voyelle ou consonne (v/c) ? : v
-----
m o p a i j e _ _ _ _ _
-----
Le 2ème joueur tire une lettre :  voyelle ou consonne (v/c) ? : c
-----
m o p a i j e l _ _ _ _ _
-----
Le 1er joueur tire une lettre :  voyelle ou consonne (v/c) ? : v
-----
m o p a i j e l i _ _ _ _ _
-----

```

Joueur 1 : proposez un mot : pale

Joueur 2 : proposez un mot : ami

Le joueur 2 a gagné, son mot est le plus long

Les deux mots **pale** et **ami** sont deux mots valides. Ils sont bien formés à partir des lettres du tirage (m o p a i j e l i) et ils appartiennent tous les deux au dictionnaire.

Votre programme devra être structuré à l'aide de procédures ou de fonctions. En particulier, il devra comporter *au moins* les routines :

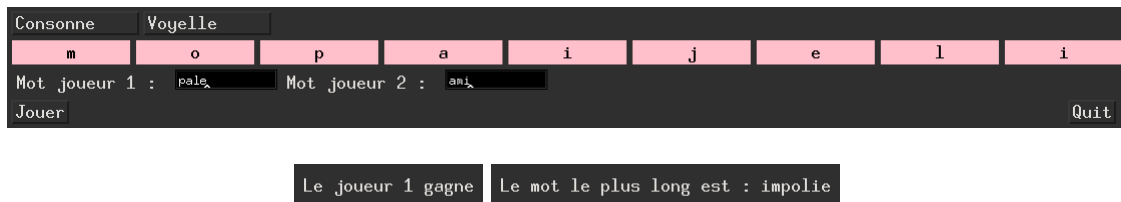
- **tirerVoyelle** qui renvoie une voyelle tirée au hasard dans l'ensemble des voyelles disponibles;
- **tirerConsonne** qui renvoie une consonne tirée au hasard dans l'ensemble des consonnes disponibles;
- **motValide** qui teste si le mot proposé par un joueur est valide ou pas (*i.e.* lettres valides et mot appartenant au dictionnaire);
- **jouer** qui gère la partie entre les deux joueurs.

Cette liste de routines n'est pas exhaustive, et certainement vous devrez en définir d'autres.

Interface utilisateur

À l'aide de la bibliothèque graphique *libsx*, votre programme devra proposer une interface graphique ergonomique à l'utilisateur. Notez que vous pouvez commencer votre projet par une interface textuelle, puis la remplacer par un interface graphique. La programmation du jeu proprement dit **doit être indépendante** de l'interface graphique.

Ci-dessous, un exemple d'interface graphique possible :



3 Bonus

À partir du tirage choisi, votre programme devra donner un plus long mot possible. Par exemple, avec les lettres précédentes m o p a i j e l i, un mot le plus long possible est **impolie**.

4 Remise du projet

Votre projet est à faire en *binôme*. Il est à rendre au plus tard :

Le samedi 4 juillet 2020, 23h

1. vous enverrez à vos enseignants `dmei2006@xidian.edu.cn`, `jyliu@xidian.edu.cn` une archive `pluslongmot-n1-n2.tar.gz` avec `n1` et `n2` sont les deux derniers chiffres de numéro d'étudiant
2. vous enverrez à `vg@unice.fr` un rapport en français de 2 pages (pas moins, pas plus), *UNI-QUEMENT* au format pdf, qui décrit votre projet.

L'archive devra obligatoirement contenir :

- vos fichiers sources `.c` et `.h` correctement documentés (chaque fonction doit avoir un commentaire, les invariants de boucle doivent être marqués), indenté, et codé (les noms de variables explicites, éviter les trop longues fonctions) ;
- le fichier `Makefile` qui permet la compilation de toute votre programme.
- un fichier `Documentation` au format pdf et décrivant le fonctionnement général du programme, les algorithmes, ainsi que les choix de programmation ;
- la compilation avec les options `-Wall` ne doit pas donner de *warning*.

Bon travail et bon courage