

TP 2 – Jeu du Motus en Python

Classes : TIC-Z

Enseignante : Mabrouk Asma

Créez un nouveau répertoire de travail pour les scripts de ce TP.

1. Règles du jeu

Le jeu « Motus » repose sur la recherche de mots d'un nombre fixé de lettres (entre 7 et 10). Un candidat propose un mot qui doit contenir le bon nombre de lettres et être correctement orthographié, sinon il est refusé. Le mot apparaît alors sur une grille : les lettres bien placées sont colorées en rouge, les lettres présentes mais mal placées sont en jaune.



C	A	S	T	O	R
C	I	N	E	M	A
C	Y	P	R	E	S
C	I	T	R	O	N

Une lettre ne peut être colorée au maximum que le nombre de fois que cette lettre apparaît dans le mot. Par exemple, si le mot cherché est CABLES et que l'on propose CABANE, on aura la coloration :

C	A	B	A	N	E
---	---	---	---	---	---

et non pas

C	A	B	A	N	E
---	---	---	---	---	---

car il n'y a qu'un « A » dans CABLES.

2. Implémentation du jeu de motus en console

Mots non valables

- les noms propres ;
- les mots mal orthographiés ;
- les mots composés (chewing-gum, week-end, ...) ;
- les verbes conjugués (seuls les infinitifs et les participes passés et présents sont acceptés) ;
- les mots qui ne commencent pas par la première lettre indiquée ;
- les mots qui ne respectent pas le nombre de lettres donné.

Votre programme devra :

- Le joueur doit choisir un nombre de lettres prédéfini par le système. Ce nombre devra être entre 7 et 10 ;
- prendre un mot au hasard dans le fichier « dico.txt ».
- indiquer la première lettre au joueur ;
- écrire le numéro de l'essai ;

- écrire sous la proposition du joueur :
 - une lettre majuscule si elle est bien placée ;
 - une lettre minuscule si elle est mal placée ;
 - un point si la lettre n'est pas dans le mot, ou si elle a moins d'occurrences ;
 - un message d'erreur si le mot est non valide. Un mot est non valide s'il n'a pas la même longueur que le mot à deviner ou il ne commence pas par la lettre du mot à deviner où il n'existe pas dans le dictionnaire.
 - Le joueur gagne s'il parvient à trouver le mot juste avant le sixième essai. Si non il perd la partie.
- a. Ecrire la fonction **dictionnaire_ordonne** qui lit le fichier f et range les mots alphabétiquement selon leur longueur (entre 7 et 10) dans un dictionnaire. La fonction retourne un dictionnaire dont pour chaque longueur L (clé) est associée une liste des mots de longueur L (valeur).
 - b. Ecrire la fonction **valide** qui, étant donné le mot à deviner *prop*, teste si le mot proposé par le joueur est valide ou non.
 - c. Vous allez ensuite coder le programme principal qui effectue les étapes suivantes :
 - i. Choix du nombre de lettres du mot à deviner ;
 - ii. Création du dictionnaire ordonné dic à partir du fichier dico.txt ;
 - iii. Choix aléatoire du mot secret dans le dictionnaire dic ;
 - iv. Entrée de l'essai du joueur ;
 - v. Affichage des lettres bien placées en majuscule et celles mal placées en minuscule et le nombre d'essai ;
 - vi. Suivant le résultat, retour à l'étape « iv » ou passage à l'étape suivante ;
 - vii. Affichage si le joueur a gagné ou non. En cas de défaite le mot secret sera dévoilé.

3. Exemple de partie :

```

Vous cherchez un mot de 7 lettres. Vous avez 6 essais.
O.....
1 orageux
OrA.e..
2 opaline
O.A.I.e
3 otaries
OTARIES
Bravo !

Vous cherchez un mot de 8 lettres. Vous avez 6 essais.
F.....
1 fouinant
FOUi.a.t
2 febriles
Fe..i..s
3 fouettes
FOUet..s
4 foutasse
Pas dans le dictionnaire
5 foutaise
FOUTAISE
Bravo !

Vous cherchez un mot de 7 lettres. Vous avez 6 essais.
T.....
1 trousse
T.o.s..
2 passeur
Première lettre non valide
3 traceur
T.aC...
4 tamisees
Longueur non valide
5 tapisse
TA..s..
6 tapoter
TA.o...
Désolé, la réponse était TANCONS

```