

Langages et programmation

Mini projet sur le jeu du pendu

Objectif

Réaliser une implémentation du jeu du pendu en mode ligne de commande (i.e. aucune interface graphique n'est demandée).

Deux versions du jeu sont envisagées :

- une version classique conforme aux règles du jeu ;
- une version on l'on modifie en cours de partie le mot à trouver.

Consignes générales et critères d'évaluation.

La production finale attendue est constituée des seuls fichiers contenant les codes sources des fonctions écrites pour répondre aux problématiques posées.

La validation de la correction de vos fonctions sera réalisée à partir de tests unitaires. Il est par conséquent impératif que vous respectiez le prototype des fonctions, c'est à dire le nombre des arguments d'appel à fournir, leur ordre, leur type, et la nature de la valeur renvoyée.

Le travail est individuel et à rendre *via l'espace élève de ProNote* au plus tard

- le *mardi 14 décembre 2021* pour la classe de *1NSI-2*
- le *vendredi 17 décembre 2021* pour la classe de *1NSI-1*

sous la forme d'une archive (.zip) à votre nom constituée de l'intégralité des fichiers suivants :

- le fichier texte `mots.txt` contenant une liste de mots courants de 7 à 10 lettres ;
- le fichier Python `dessin_pendu.py` contenant le code source de la fonction utilisée pour afficher en console une représentation graphique de l'état du jeu ;
- le fichier Python `jeu_pendu_v1.py` correspondant à l'implémentation classique du jeu suivant les règles de ce dernier ;
- le fichier `jeu_pendu_v2.py` correspondant à l'implémentation du jeu avec modification du mot à trouver en cours de partie.

Les critères d'évaluation sont les suivants :

- qualité de la réponse aux problématiques ;
- pertinence des choix algorithmiques ;
- maîtrise des éléments de langage Python ;
- documentation du code (i.e. docstring des fonctions et commentaires) ;
- lisibilité du code (i.e. choix des noms de variables).

I. Présentation

Le but du jeu est de trouver un mot dont on ne connaît initialement que le nombre de lettres. Ce mot est représenté en console par des traits soulignés ('_'), pour chaque lettre manquante, séparés par des espaces (' ').

A titre d'exemple, en début de partie, l'affichage souhaité en console pour un mot de 7 lettres est le suivant :

```
mot à trouver : _ _ _ _ _ _ _
```

A chaque tour de jeu, le joueur propose une nouvelle lettre :

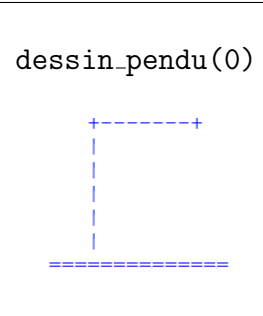
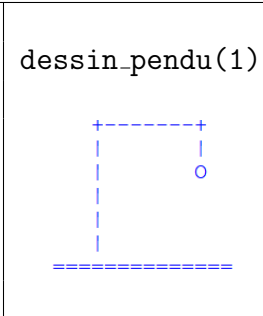
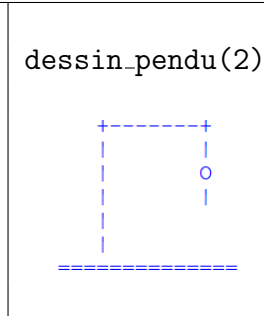
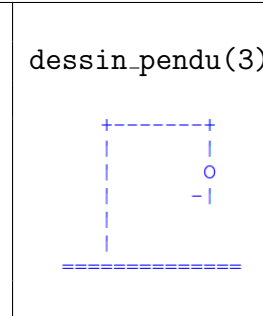
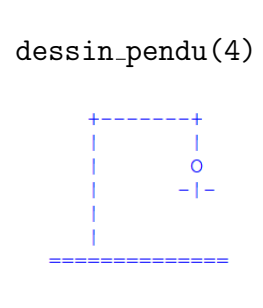
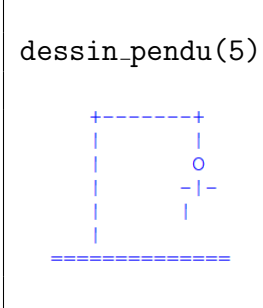
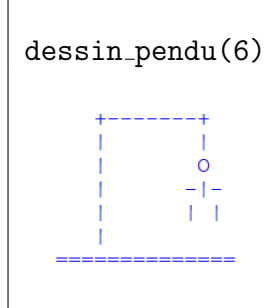
- Si la lettre appartient au mot choisi alors elle est affichée aux bonnes positions en complétant le mot à trous.

A titre d'exemple, si le mot à trouver est 'BONJOUR' et que le joueur propose la lettre 'O' alors l'affichage en console est modifié comme suit :

```
mot à trouver : _ O _ _ O _ _
```

- Si la lettre n'appartient pas au mot choisi alors le dessin du pendu affiché en console est complété d'un segment. Lorsque celui est complet cela signifie que la partie est terminée et que le joueur a perdu.

A titre d'exemple, le dessin du pendu affiché en console, pour un nombre de propositions erronées de 0 à 6, est donné ci-dessous :

dessin_pendu(0)	dessin_pendu(1)	dessin_pendu(2)	dessin_pendu(3)
			
dessin_pendu(4)	dessin_pendu(5)	dessin_pendu(6)	
			

Remarque. L'affichage en console de l'état du jeu présenté précédemment constitue l'interface minimale du jeu du pendu.

Libre à vous de l'enrichir si vous le souhaitez, en particulier :

- pour rappeler au joueur les consignes en début de partie ;
- pour notifier au joueur la fin de la partie que ce soit en cas de succès ou d'échec.

II. Travail demandé

II. 1. Implémentation selon les règles du jeu

L'intégralité des codes demandés est à implémenter dans le fichier `jeu_pendu_v1.py`.

- 1) Écrire une fonction `importer_mots` qui :
 - prend en paramètre un nom de fichier (i.e une chaîne de caractères) ;
 - renvoie sous la forme d'une liste de mots le contenu du fichier texte passé en argument de la fonction.

A titre d'exemple, l'exécution en console de l'instruction

```
liste_mots = importer_mots('mots.txt')
```

renvoie une liste de 11716 mots de 7 à 10 lettres.

- 2) Écrire une fonction `choisir_mot` qui :
 - prend en paramètre une liste de mots ;
 - renvoie un mot choisis au hasard dans la liste de mots.

Remarque. Pour choisir un élément au hasard dans une liste, vous pouvez, après l'avoir importé, utiliser la fonction `choice` du module `random` de Python.

<https://docs.python.org/fr/3/library/random.html>

A titre d'exemple, après exécution en console de l'instruction

```
mot_choisi = choisir_mot(liste_mots)
```

la variable `mot_choisi` = BONJOUR.

- 3) Écrire une fonction `construire_mot_partiel` qui :
 - prend en paramètre le mot choisi ;
 - renvoie une chaîne de caractères de longueur double de celle du mot choisi.

A titre d'exemple, après exécution en console de l'instruction

```
mot_partiel = construire_mot_partiel(mot_choisi)
```

la variable `mot_partiel` = _ _ _ _ _ .

- 4) Écrire une fonction `ajouter_lettre` qui :
 - prend en paramètre une lettre, le mot choisi et le mot partiellement trouvé ;
 - complète le mot partiellement trouvé avec la lettre partout où elle apparaît dans le mot choisi ;
 - renvoie le mot partiellement trouvé actualisé si la lettre est présente dans le mot choisi ou inchangé dans le cas contraire.

A titre d'exemple, après exécution en console de l'instruction

```
mot_partiel = ajouter_lettre('O', mot_choisi, mot_partiel)
```

la variable `mot_partiel` = _ O _ _ O _ _ .

- 5) Écrire les instructions nécessaires à la gestion du déroulement d'une partie à partir des fonctions précédentes.

II. 2. Implémentation avec modification du mot à trouver

L'intégralité des codes demandés est à implémenter dans le fichier `jeu_pendu_v2.py`.

Dans cette seconde partie, on souhaite modifier le mot à trouver à chaque fois que le joueur propose une lettre appartenant à ce dernier. Bien entendu, le choix du nouveau mot doit prendre en compte les coups passés du jeu, c'est à dire être cohérent avec l'ensemble des lettres déjà proposées par le joueur.

6) Écrire une fonction `modifier_liste_mots` qui :

- prend en paramètre la liste de mots, le mot choisi ainsi que la liste des lettres déjà proposées par le joueur ;
- renvoie une nouvelle liste ne contenant que les mots composés de lettres n'appartenant pas à la liste de celles déjà proposées par le joueur, et dont le nombre total de lettres est identique à celui du mot choisi.

7) En utilisant les fonctions implémentées dans la partie II.1, écrire les instructions nécessaires à la gestion du déroulement d'une partie du jeu.