

Projet Python en binôme / Jeu de Pendu

Instructions

- Nom du fichier à créer : `pendu_nom1_nom2.py`
- Déposez le fichier sur Pronote **avant le lundi 14 novembre à minuit**. Un retard entraînera un retrait de points (2 points par jour de retard).
- Programmez le jeu en respectant les instructions (fonctions, etc). Pensez à écrire les signatures (ou spécifications) des fonctions.
- La réussite complète du niveau 1 assure la note de 16/20.
- Pensez à bien commenter votre code si nécessaire et à écrire les spécifications des fonctions.

Mieux vaut un code incomplet qu'un code recopié (nous utilisons un détecteur de plagiat)

Principe

Le jeu du pendu consiste à choisir un mot et à le faire deviner à son adversaire en un nombre limité de propositions de lettres.

Dans le jeu que vous allez programmer, l'ordinateur devra choisir un mot au hasard et demander des lettres à l'utilisateur, en affichant après chaque proposition l'état du pendu et les lettres du mot déjà devinées (en utilisant des tirets pour les lettres inconnues).

Niveau 1 : votre mission

Pour programmer le pendu, vous avez à votre disposition dans le Drive partagé

- un fichier `mots.txt` qui contient un peu plus de 800 mots
- un fichier `pendu.py` contenant une fonction Python `dessinPendu` qui permet de dessiner le pendu. Dans ce fichier se trouve également quelques lignes de code permettant d'ouvrir et de lire le fichier `mots.txt`.

Travail demandé dans le fichier `pendu.py` :

Écrire une fonction `jeu_pendu()` qui permet de jouer au pendu et renvoie le mot 'victoire' ou 'defaite' selon le résultat.

Dans cette fonction :

1. tirer au hasard un mot appelé ensuite `mot_mystere` dans le fichier `mots.txt`

2. définir `mot_trouve` comme une liste de caractères (et non comme une chaîne de caractères) de la longueur du mot à deviner
3. demander au joueur une lettre
4. appeler une fonction `miseajour_mot(mot_mystere, mot_trouve, l)` écrite en dehors de `jeu_pendu()` qui met à jour `mot_trouve` si la lettre `l` est contenue dans le mot mystère et qui renvoie un booléen indiquant si la lettre est dans `mot_mystere`
5. afficher l'état d'avancement de `mot_trouve`, afficher le pendu si le mot n'a pas été trouvé après le nombre d'essais permis par les dessins ou terminer le jeu si toutes les lettres du mot ont été trouvées.

Indications :

- les triples guillemets permettent de délimiter une chaîne de caractères située sur plusieurs lignes.
- `mot = mot.rstrip()` permet de supprimer du mot tous les caractères de fin (espaces, retour à la ligne etc.)
- la fonction `''.join(liste)` renvoie une chaîne de caractères correspondant à la concaténation des éléments de liste donnée en paramètre. Elle peut vous être utile pour transformer la liste `mot_trouve` en chaîne de caractères.
- la fonction `list(chaine)` peut vous être utile (à tester pour comprendre son utilité)

Niveau 2 : Améliorations

Ce niveau prolonge le niveau 1 et son code figure donc dans le même fichier.

1. En dehors de la fonction `jeu_pendu()`, proposer au joueur de rejouer après une partie et afficher à la fin le pourcentage de victoires et celui de défaites.
2. Dans la fonction `jeu_pendu()`, détecter si le joueur propose plusieurs fois la même lettre. Dans ce cas, demander une nouvelle lettre sans mettre à jour le pendu.
3. Permettre au joueur de proposer indifféremment des lettres minuscules ou majuscules.

Si vous avez d'autres idées d'améliorations, n'hésitez pas ! Bon courage !