# Projet: Jeu du Pendu

## 🛕 princ

#### principe du projet

- Vous travaillerez par binomes sur ce projet (ou exceptionnellement par trinome).
- Vous rendrez le code complet + votre dossier personnel pour le 09 novembre 2022 au plus tard.
- Vous aurez un temps en classe pour réaliser le projet, mais ce temps ne sera pas suffisant! Vous devrez vous coordonner pour arriver à vos fins!

# 1. Description du projet

## 0

#### jeu du Pendu

Le principe retenu pour le jeu est le suivant :

- 1. Un mot français est tiré aléatoirement depuis un fichier externe.
- 2. Ce mot est *nettoyé* des accents et autres signes diacritiques français (ç, æ, æ, ...) puis converti en majuscules. **Les tirets des** mots composés sont conservés!
- 3. On affiche une série de tirets correspondant aux emplacements des lettres ainsi que la potence de départ du pendu.
- 4. On demande à l'utilisateur-trice une lettre saisie au clavier, et on vérifie que :
  - · c'est bien une unique lettre;
  - elle n'a pas déjà été proposée.
- 5. Si la lettre fait partie du mot cherché, les tirets correspondants sont remplacés par la lettre. Sinon on ajoute un élément au dessin du pendu.
- 6. On reprend à l'étape 4 tant que :
  - soit il reste des lettres à trouver;
  - soit le pendu est complètement dessiné (6 échecs).
- 7. Une fois la partie terminée, le joueur/la joueuse" peut alors choisir de recommencer ou non une nouvelle partie.

Le rendu final devrait ressembler si possible à celui de la vidéo suivante :



## 2. Les fichiers nécessaires



Pour mener à bien le projet, vous aurez d'abord besoin du fichier des mots français.

Vous téléchargerez celui-ci et le copierez dans votre dossier de travail, à côté du fichier python utilisé.

J'ai retiré de celui-ci tous les mots contenant des signes diacritiques autres que :

- · les lettres accentuées;
- · les tirets;
- · les cédilles ;
- les lettres combinées (e dans l'o,...)

Les mots peuvent être en majuscule, en minuscule, ou toute autre combinaison de casse.



#### Base de code Python

Je vous donne ici le code qui doit être votre base de travail.



fonction choix\_mot(adresseFichier)

Cette fonction ne doit pas être modifiée!

Elle prend en argument l'adresse **relative** ou **absolue** d'un fichier texte, et renvoie la chaine de caractères correspondant à une ligne de ce fichier, où :

- les éventuels espaces de début et de fin de ligne sont supprimés ;
- les caractères retour chariot (sauts de lignes) \n sont supprimés.

Dans le cadre de ce projet, le fichier liste\_francais\_modifiee.txt doit normalement être situé dans le même répertoire que ProjetPendu.py. Donc vous pouvez utiliser cette fonction de la manière suivante :

```
mot = choix_mot("liste_francais_modifiee.txt")
```

Ainsi la variable mot contiendra un mot extrait aléatoirement du fichier.

### 3. Plan de travail

- 1. Vous commencerez par compléter la fonction formate\_mot(mot), afin qu'elle renvoie une chaîne de caractères en majuscule dans laquelle tous les signes diacritiques ont été supprimés (à par les tirets des mots composés). Des tests unitaires sont donnés à titre d'exemple Vous pouvez éventuellement rajouter les votres.
- 2. Vous complèterez ensuite la fonction genere\_tirets(mot\_a\_trouver, lettres\_utilisees), qui renvoie une chaîne de caractères correspondant à celle passée en premier argument mot\_a\_trouver, pour laquelle les lettres non présentes dans la chaîne de caractère lettres\_utilisees. De plus, chaque caractère de la chaîne finale devra être suivi d'un espace. Des tests unitaires sont donnés à titre d'exemple Vous pouvez éventuellement rajouter les votres.
- 3. Vous complèterez la fonction compte\_restantes(mot\_a\_trouver,lettres\_utilisees) qui renvoie un entier correspondant au nombre de lettres restant à trouver dans mot\_a\_trouver sachant la chaîne de lettres déjà utilisées lettres\_utilisees.
- 4. Vous complèterez ensuite la fonction demande\_joueur\_lettre() et la rendrez dumbproof : cette fonction doit continuer à redemander au joueur de saisir une lettre tant que celle-ci n'est pas compatible avec les règles du jeu.
- 5. Vous complèterez ensuite la fonction affiche\_pendu(mot\_a\_trouver, lettres\_utilisees, nb\_echecs) qui affiche non seulement la potence, amis aussi le mot à trouver sous sa forme de tirets. Pour construire cette fonction, vous utiliserez une *f-string* multi-lignes telle que :



- 6. A partir de toutes les fonctions précédentes, vous finaliserez le jeu en complétant la fonction une\_manche().
  - 1. Une fois le jeu complété, vous devrez en outre compléter un dossier personnel d'une ou deux pages présentant :
  - 2. ce que vous avez réalisé individuellement dans ce projet;
  - 3. les difficultés rencontrées et/ou les problèmes que vous n'avez pas pu résoudre ;
  - 4. les aides qui vous on été apportées.
  - 5. Vous pourrez enfin apporter des modifications et/ou améliorations au code, par exemple en :
    - ajoutant un compteur de score qui donne le nombre de réussites par rapport au nombre de parties jouées.
    - ajouter un niveau de difficulté, en changeant le nombre d'erreurs possibles ;
    - etc...

## 4. Grille de notation

intitulé	barême	Détails
fonction formate_mot	3 pt	passage de tous les tests unitaires
fonction genere_tirets	2 pt	passage de tous les tests unitaires
fonction compte_restantes	2 pt	passage de tous les tests unitaires
fonction demande_joueur_lettre	1 pt	dumbproof
fonction affiche_pendu	3 pt	Affichage correct
fonction une_manche	3 pts	On attend un jeu a minima fonctionnel
Noms des variables clairs	2 pts	On proscrira les noms de variable d'un seul caractère, sauf compteurs précis
Code commenté et clair	3 pts	Des explications minimales doivent être écrites pour expliquer votre code
Améliorations, qualité du code, etc	2 pts	