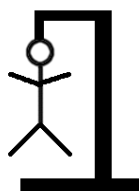




Coder un Pendu en Python

Niveau Débutant • 13-16 ans • 1h30

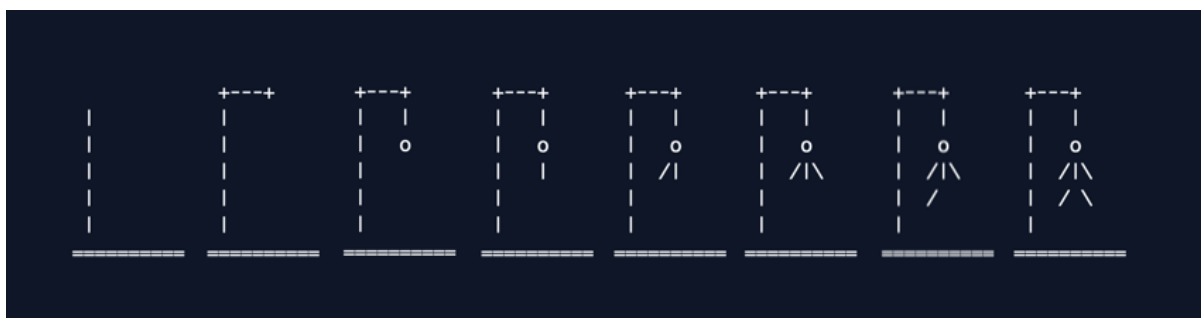
1. Présentation du jeu



Le **Pendu** est un jeu consistant à trouver un mot en devinant les lettres qui le composent. Bien que traditionnellement joué à 2 sur papier, nous allons aujourd'hui programmer une version digitale du jeu en **Python**.

Certains éléments du code du **Pendu** ont été perdus... À vous de compléter le code du jeu pour qu'il fonctionne de nouveau !

Dans ce sujet, vous apprendrez à programmer des boucles, des conditions en **Python** et le fonctionnement global d'un jeu.



2. Outils & Ressources

2.1 Ressources

Pour commencer, rendez-vous sur <https://repl.it/@EmmaEpitech/PythonPendu>.

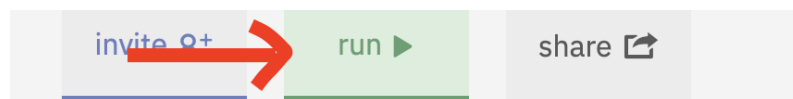
L'écran est découpé en plusieurs parties : la partie fichier, à gauche : c'est ici que vous coderez le jeu. Et la partie rendu, à droite : c'est ici que vous verrez votre progression et pourrez tester votre jeu.

2.2 Prise en main de la plateforme

Sur la partie gauche de l'écran, tu peux voir une liste de fichiers **Python**.

Aujourd'hui, tu n'auras besoin de modifier que le fichier "**main.py**". Tu peux toutefois aller voir ce qu'il se passe dans les autres fichiers si tu le souhaites.

Pour lancer ton code appuie sur la flèche verte "**RUN**" en haut de l'écran.



3. Découverte du jeu

3.1 Initialiser le Pendu



La première étape pour ce jeu du **Pendu** est l'initialisation. Lors de cette étape, un mot aléatoire sera déterminé, et les variables du jeu seront initialisées, comme la variable contenant la liste des lettres utilisées, par exemple.

Pour cela, il suffit d'ajouter la ligne de code suivante au début de ton fichier. Ajoute ensuite la seconde ligne pour afficher l'état du jeu.

```
my_pendu = Pendu()  
my_pendu.show_game_status()
```



Lance le jeu pour voir ce qu'il se passe !

3.2 La boucle de jeu



À présent, nous allons créer la boucle de jeu. À chaque tour, nous devons vérifier si le mot aléatoire a été trouvé.

La variable **word_to_guess**, correspond au mot à deviner, c'est la solution.

Et la variable **guessed_letters** contient elle les lettres déjà devinées à l'emplacement où elles se trouvent dans la solution.



Par exemple, si le mot est “**PENDU**” et que j’ai trouvé la lettre “**D**”, alors **guessed_letters** sera “**___D_**”.

Ajoute la ligne suivante pour créer la boucle de jeu :

```
while my_pendu.guessed_letters != my_pendu.word_to_guess:
```

3.3 Tour de jeu : saisir une lettre



Nous allons maintenant demander à l'utilisateur d'entrer un caractère, puis vérifier si celui-ci est déjà utilisé. (Sans tenir compte de s'il est valide ou non, nous vérifions cela à l'étape suivante).

Ajoute la ligne suivante à l'intérieur de la boucle pour demander à l'utilisateur d'écrire un caractère puis le récupérer :

```
char = get_char("> ")
```

Puis ajoute la condition suivante pour vérifier que c'est bien la première fois que le caractère est écrit par l'utilisateur :

```
if char in my_pendu.used_letters:  
    continue
```



Le mot-clé **continue** permet d'arrêter ce tour de boucle, sans exécuter les instructions suivantes, puis d'en commencer un nouveau.

Finalement, ajoute la ligne suivante pour enregistrer la lettre dans la liste des lettres utilisées :

```
my_pendu.used_letters += char
```

3.4 Tour de jeu : vérifier si la lettre est valide



À présent, nous allons vérifier si la lettre entrée par l'utilisateur fait partie du mot à deviner. Si ce n'est pas le cas, nous ajouterons un point d'erreur au compteur pour faire avancer le **Pendu**.

Appelle la fonction **update_guessed_letters** pour mettre à jour la variable **guessed_letters** selon la lettre saisie par l'utilisateur.



Cette fonction retourne le nombre de lettres qui aura été mise à jour.

```
nb = my_pendu.update_guessed_letters(char)
```

Si le nombre de lettres mises à jour est égal à 0, alors la lettre n'est pas valide. Incrémente de 1 le compteur d'erreur dans ce cas.

```
if nb == 0:  
    my_pendu.error_count += 1
```

Puis ajoute la ligne suivante pour afficher l'état du jeu :

```
my_pendu.show_game_status()
```



Lance le jeu pour voir ce qu'il se passe !

3.5 Fin de la partie



Dernière étape, détecter le cas de perte si l'on a fait trop d'erreur, et si l'on a gagné.

Le joueur perd si le nombre d'erreurs est au moins égal au nombre maximum d'erreurs autorisées. Si le joueur perd, on quitte le jeu immédiatement.

Ajoute la condition suivante :

```
if my_pendu.error_count >= const.MAX_ERROR:  
    print("Perdu :(")  
    break
```

Le joueur gagne donc lorsqu'on sort de la boucle et que nous n'avons pas dépasser le nombre maximal d'erreurs.

Rappelle-toi, au début de l'atelier nous avons défini dans la condition de la boucle while que celle-ci continuait jusqu'à ce que la solution soit trouvée.

Il suffit donc d'ajouter la ligne suivante tout à la fin du programme pour indiquer à l'utilisateur que la partie est gagnée, en n'oubliant pas de tester le nombre d'erreurs.

```
if my_pendu.error_count < const.MAX_ERROR:  
    print("gagné !")
```

 Appuie sur **RUN** et essaye de jouer !

Bonne chance à toi !

4. Bonus

Si tu as fini toutes les étapes précédentes, tout d'abord **Bravo !**

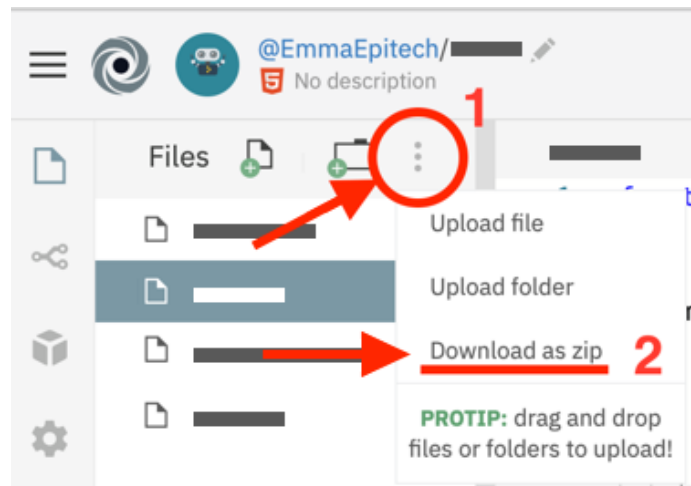
Nous avons préparé quelques étapes bonus que tu peux faire pour améliorer le jeu ou le rendre plus difficile.

- Ajouter de nouveaux mots dans le dictionnaire.
- Afficher le mot complet lorsque le joueur a perdu.
- Afficher la première lettre du mot dès le début.
- Proposer à l'utilisateur de rejouer une nouvelle partie.
- Permettre à l'utilisateur d'écrire le mot en entier en plus d'écrire les lettres une par une.
- Personnaliser l'Ascii Art du Pendu.

5. Sauvegarde ton programme



Tu peux enregistrer ton jeu sur ton ordinateur pour le modifier ou y jouer plus tard. Pour cela, clique sur “**Download as zip**”.



6. Quelques liens utiles

Pour apprendre le Python :

→ <https://www.w3schools.com/python>

Pour refaire l'exercice :

→ <https://repl.it/@EmmaEpitech/PythonPendule>

Pour voir nos autres exercices :

→ <https://repl.it/@EmmaEpitech>

Pour plus d'informations sur nos activités :

→ <https://www.e-mma.org>