

Jeu du Pendu



Ce TP a été le premier gros exercice en Python à faire chez soi.

Il nous a été donné par Mr Lefebvre Pierre,
Professeur de programmation Python, Java et C.

Enoncé :

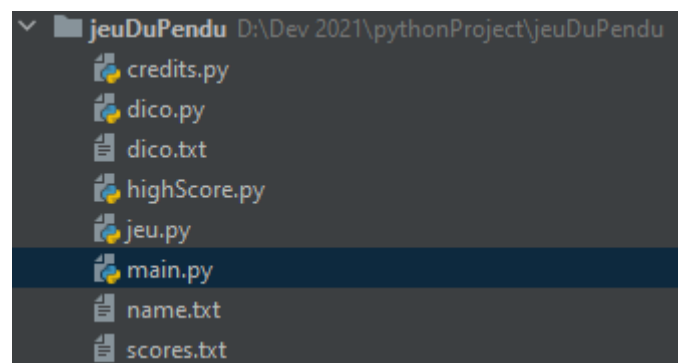
Vous devez recréer le jeu du pendu.

Pour ce faire, il faudra suivre quelques règles :

- il doit y avoir un dictionnaire de mots de 7 lettres
 - il doit y avoir un score
- il doit y avoir des underscores à la place des lettres
 - si le pendu est complet, le joueur a perdu
 - si le joueur a trouvé le mot, il gagne.

Arborescence du Projet :

Voici l'arborescence que j'ai choisis pour faire mon code.
J'ai profité de ce TP pour m'exercer au codage sur plusieurs fichiers.



Présentation d'une partie :

Le programme débute avec un sommaire :

```
Bienvenue dans le jeu du PENDU.  
Dans quelle section voulez-vous entrer ?  
1 - Dictionnaire  
2 - High Score  
3 - Jeu  
4 - Credits  
5 - Quitter
```

Sommaire du dictionnaire :

```
Que souhaitez vous faire ?  
1 - voir tous les mots  
2 - Ajouter un mot  
3 - voir un mot au hasard  
4 - Retourner au menu
```

Affichage des mots du dictionnaire :

```
ABATTRA  
  
ABEILLE  
  
ABETIES  
  
ABONDER  
  
ABORDER  
  
ABOYEUR
```

Ajout d'un mot au dictionnaire :

```
quel est le nouveau mot ?DENIERS  
nouveau mot ajouté : DENIERS
```

Afficher un mot au hasard :

```
Que souhaitez vous faire ?  
1 - voir tous les mots  
2 - Ajouter un mot  
3 - voir un mot au hasard  
4 - Retourner au menu  
3  
BALEINE
```

Revenir au menu :

```
Que souhaitez vous faire ?  
1 - voir tous les mots  
2 - Ajouter un mot  
3 - voir un mot au hasard  
4 - Retourner au menu  
4  
Dans quelle section voulez-vous entrer ?  
1 - Dictionnaire  
2 - High Score  
3 - Jeu  
4 - Credits|  
5 - Quitter
```

Affichage du High Score :

```
Mafemme ..... 10000  
Horo ..... 16  
Kaolin ..... 6  
Lau ..... 4  
Steph ..... 3  
Xavier ..... 2  
Z ..... 0
```

(ma femme gagne toujours, il faut faire avec)

Jeu :

```
Quel est votre pseudo ?  
horo  
Te revoilà Horo !  
Ton score est de : 16
```

emplacement des lettres du mot à trouver :

```
-----
```

tour N°1 :

```
tour n° 1 :  
c  
c-----
```

la réponse est bonne !

tour N°2 :

```
tour n° 2 :  
0  
  
-----  
|/  |  
|  
|  
|  
/|\-----  
  
C-----
```

Faux. La pendaison arrive !

Défaite :

```
tour n° 8 :  
d  
  
-----  
|/  |  
|  0  
|  /|\  
|  / \  
/|\-----  
  
C-----  
  
Dommage ! Potasse un peu et reviens ;p
```

Victoire :

```
tour n° 6 :  
n  
BALATON  
  
Comme dirait le gars dans Code Quantum : Oh bravo !
```

Nouvelle partie :

```
Voulez-vous continuer ?(y/n)y
```

Crédit :

```
L'origine du jeu du pendu est inconnue. En Grande-Bretagne, il semble avoir fait son apparition pendant l'époque victorienne, vers la fin du XIXe siècle.  
  
Pour ce qui est du code, remerciez Léo avec un café au lait ;p
```

Le code source et leur action

main.py

```
8  #blabla d'acceuil
9  print("Bienvenue dans le jeu du PENDU.")
10 #déclaration
11 menuBreak = False
12 #boucle d'appels
13 while menuBreak == False:
14     #blabla
15     print("Dans quelle section voulez-vous entrer ?\n1 - Dictionnaire\n2 - High Score\n3 - Jeu\n4 - Credits\n5 - Quitter")
16     #demande où le joueur veut aller
17     section = int(input(""))
18     if section == 1:
19         dico()                # dictionnaire de mots
20     elif section == 2:
21         highScore()           # présentation des scores
22     elif section == 3:
23         # nouvelle partie
24         print("Quel est votre pseudo ?")
25         pseudo = input("")
26         pseudo = pseudo.capitalize()
27         isInNameTxt(pseudo)    # vérifie si le pseudo existe
28         scoreJoueur = joueur(pseudo) # obtien le score du joueur
29         strtParty(scoreJoueur, pseudo) # démarre la partie
30     elif section == 4:
31         credit()               # petite histoire et copyright
32     elif section == 5:
33         menuBreak = True       # extinction du programme
34     print("à la prochaine !")
```

Il y a dedans le menu principal du programme avec les appels aux autres parties du jeu.

HighScore.py

Fonction profil :

```
1  # ouverture, lecture et fermeture
2  def profil():
3      with open("scores.txt", 'r') as scoresFile: # score
4          scores = scoresFile.readlines()
5          for i in range(len(scores)):
6              scores[i] = int(scores[i].replace("\n", ""))
7          scoresFile.close()
8
9      with open("name.txt", 'r') as nameFile: # pseudo
10         name = nameFile.readlines()
11         for i in range(len(name)):
12             name[i] = name[i].replace("\n", "")
13         nameFile.close()
14     return scores, name
```

Récupère les pseudo et scores

Fonction joueur :

```
16  # vérifie le pseudo puis obtient son score
17  def joueur(pseudo):
18      scores, name = profil()
19      for i in range(len(pseudo)):
20          if name[i] == pseudo:
21              scoreJoueur = scores[i]
22              i = len(pseudo)
23      return scoreJoueur
```

Récupère le score du joueur.

Fonction ajout :

```
25     # ouverture, écriture tout à la fin et fermeture
26     def ajout(pseudo):
27         pseudo = "\n"+pseudo
28         with open("scores.txt", 'a') as scoresFile: # score
29             scoresFile.write("\n0")
30             scoresFile.close()
31         with open("name.txt", 'a') as nameFile: # pseudo
32             nameFile.write(pseudo)
33             nameFile.close()
```

Ajoute un pseudo au fichier name et créer un score dans le fichier score.

Fonction modif :

```
35     # ouverture, écriture et fermeture
36     def modif(names, scores):
37         with open("scores.txt", 'w') as scoresFile:
38             scoresFile.write("\n".join(scores))
39             scoresFile.close()
40         with open("name.txt", 'w') as nameFile:
41             nameFile.write("\n".join(names))
42             nameFile.close()
```

modifie le score lié au pseudo du joueur.

Fonction tri_bulle :

```
44 # tri des scores du meilleur au moins bon
45 def tri_bulle():
46     scores, names = profil()
47     n = len(scores)
48     # Traverser tous les éléments du tableau
49     for i in range(n):
50         for j in range(0, n-i-1):
51             # échanger si l'élément trouvé est plus grand que le suivant
52             if scores[j] < scores[j+1]:
53                 scores[j], scores[j+1] = scores[j+1], scores[j]
54                 names[j], names[j+1] = names[j+1], names[j]
55     return scores, names
```

Tri les scores du meilleur au moins bon en gardant la liaison pseudo <=> score.

Fonction isInNameTxt :

```
57 # vérifie si le pseudo existe
58 def isInNameTxt(pseudo):
59     scores, name = profil()
60     count = 0
61     for i in range(len(name)):
62         if name[i] == pseudo:
63             count += 1
64         if count >= 1:
65             print("Te revoilà " + pseudo + " !")
66         else:
67             print("Oh ! Un nouveau.")
68             ajout(pseudo)
69             print("Pseudo : " + pseudo + "\nScore : 0")
```

Vérifie si fichier name contient le pseudo et renvoi une phrase d'introduction.

Fonction highScore :

```
71 # highScore
72 def highScore():
73     scores, names = tri_bulle()
74     for i in range(len(scores)):
75         print(names[i], ".....", scores[i])
```

Affiche les scores triés dans la fonction tri_bulle.

Fonction win :

```
77 # condition de victoire
78 def win(scoreJoueur, tour):
79     scoreJoueur += 7+13-tour
80     print("Comme dirait le gars dans Code Quantum : Oh bravo !")
81     return scoreJoueur
```

Conditions de victoire.

Fonction loose :

```
83 # condition de défaite
84 def loose(scoreJoueur, tour):
85     scoreJoueur -= 7+13-tour
86     print("Dommage ! Potasse un peu et reviens ;p")
87     return scoreJoueur
```

Conditions de défaite.

Fonction endGame :

```
89     # enregistrement du score par pseudo
90     def endGame(pseudo, score):
91         scores, name = profil()
92         for i in range(len(name)):
93             if name[i] == pseudo:
94                 scores[i] = str(score)
95             else:
96                 scores[i] = str(scores[i])
97         modif(name, scores)
```

Vérifie la ligne à modifier et écrit le score du joueur sur le fichier scores.

dico.py

Fonction dico (sommaire):

```
47     # sommaire
48     def dico():
49         #déclarations
50         ShouldBreak = False
51         direction = 0
52         #boucle d'appels
53         while ShouldBreak == False:
54             #blabla
55             print("Que souhaitez vous faire ?\n1 - voir tous les mots\n2 - ajouter un mot\n3 - mot aléatoire\n4 - exit")
56             #demande où le joueur veut aller
57             direction = int(input(""))
58             if direction == 1: # afficher tout le dico
59                 dicoLire()
60             elif direction == 2: # ajout de mot dans le dico
61                 newMot = input("quel est le nouveau mot ?")
62                 newMot = newMot.upper()
63                 dicoAjout(newMot)
64             elif direction == 3: # mot aléatoire
65                 print(prendreMotRandom())
66             elif direction == 4: # exit
67                 ShouldBreak = True
```

Fonction prendreMotRandom :

```
3      # ouverture, lecture et fermeture
4      def prendreMotRandom():
5          with open("dico.txt", 'r') as dicoFile:
6              dico = dicoFile.readlines()
7              mot = dico[randint(0, len(dico)-1)]
8              dicoFile.close()
9          return mot
```

Lit le fichier dico et prend un mot au hasard.

Fonction isWordInAlphabet :

```
11     # vérifie si le mot est composé de caractères alphabétique
12     def isWordInAlphabet(word):
13         alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
14         for i in range(len(word)):
15             if word[i] not in alphabet:
16                 return False
17         return True
```

Vérifie si les lettres du mot en paramètre sont bien alphabetiques.

Fonction dicoAjout :

```
19 # ajout d'un mot dans le dico
20 def dicoAjout(newMot):
21     with open("dico.txt", 'r+') as dicoFile: # lecture +
22         dico = dicoFile.readlines()
23         for i in range(len(dico)):
24             dico[i] = dico[i].replace("\n", "")
25         if newMot in dico:
26             print("Le mot existe déjà.")
27         elif len(newMot) != 7:
28             print("Le mot n'est pas à la bonne taille. \nIl faut uniquement des mots de 7 lettres.\n")
29         elif isWordInAlphabet(newMot)==False:
30             print("Les caractères spéciaux ne sont pas acceptés.\n")
31         else:
32             dicoFile.seek(0,0)
33             dico.append(newMot)
34             dico.sort()
35             dicoFile.write("\n".join(dico))
36             print("nouveau mot ajouté : " + newMot)
37     dicoFile.close() # fermeture
```

Ouvre le fichier dico en lecture +, vérifie si le mot est conforme et peut l'ajouter si tout va bien.

Fonction dicoLire :

```
39 # lecture du dico
40 def dicoLire():
41     with open("dico.txt", 'r') as dicoFile:
42         dico = dicoFile.readlines()
43         dicoFile.close()
44         for i in range(len(dico)):
45             print(dico[i])
```

Lit le contenu du dictionnaire.

credits.py

```
1 # histoire et copyright
2 def credit():
3     print("L'origine du jeu du pendu est inconnue. "
4           "En Grande-Bretagne, "
5           "il semble avoir fait son apparition pendant l'époque victorienne, "
6           "vers la fin du XIXe siècle."
7           "\n\nPour ce qui est du code, remerciez Léo avec un café au lait ;p\n")
8
```

Ici, j'ai essayé de placer un peu d'histoire et d'humour.

jeu.py

Animation du pendu :

```
3 # schema du pendu
4 imgPendou = [
5     " _____\n |/\n | \n | \n | \n |/\n _____\n\n",
6     " _____\n |/\n | \n | \n | \n |/\n _____\n\n",
7     " _____\n |/\n | \n 0\n | \n | \n |/\n _____\n\n",
8     " _____\n |/\n | \n 0\n | \n | \n |/\n _____\n\n",
9     " _____\n |/\n | \n 0\n | \n / \n | \n |/\n _____\n\n",
10    " _____\n |/\n | \n 0\n | \n / \n | \n |/\n _____\n\n",
11    " _____\n |/\n | \n 0\n | \n / \n | \n |/\n _____\n\n",
12    " _____\n |/\n | \n 0\n | \n / \n | \n |/\n _____\n\n"]
```

ce qui donne ce que l'on peut voir plus haut dans la présentation.

Fonction getLettre :

```
14 # obtention d'une lettre chaque tour
15 def getLettre(tour):
16     print("tour n°", tour, ":")
17     lettre = input("")
18     lettre = lettre.upper()
19     return lettre
```

Que la lettre soit en minuscule ou majuscule, la lettre pourra être lue.

Fonction LetterFound :

```
21 # compare la lettre obtenue à celles du mot
22 def LetterFound(lettre, mot, motAtrouver, lettresTrouvees):
23     for i in range(7):
24         if lettre == mot[i]:
25             motAtrouver[i] = mot[i]
26             lettresTrouvees += 1
27     return lettresTrouvees, motAtrouver
```

compare et échange le "_" par les bonnes lettres.

Fonction strtParty :

```
29 # déroulement de la partie
30 def strtParty (scoreJoueur,pseudo):
31     print("Ton score est de : ",scoreJoueur,"\n")
32     playerWantPlay = True
33     while playerWantPlay == True: # boucle nouvelle partie
34         up = 0
35         lettresTrouvees = 0
36         tour = 1
37         answer = 'o'
38         mot = prendreMotRandom() # obtention d'un mot
39         motAtrouver = ["_", "_", "_", "_", "_", "_", "_"] # emplacement du mot
40         print(mot) # à supprimer
41         for i in range(7):
42             print(motAtrouver[i], end="")
43         print("\n")
44         while up < 7 and lettresTrouvees < 7:
45             lettre = getLettre(tour)
46             if lettre in mot:
47                 lettresTrouvees, motAtrouver = LetterFound(lettre,mot,motAtrouver,lettresTrouvees)
48             else:
49                 up += 1
50                 print(imgPendu[up])
51                 for i in range(7):
52                     print(motAtrouver[i], end="")
53                 print("\n")
54                 tour +=1
55             if up >= 7:
56                 scoreJoueur = loose(scoreJoueur, tour)
57             else:
58                 scoreJoueur = win(scoreJoueur, tour)
59                 # nouvelle partie ?
60                 while answer != 'y' and answer != 'n':
61                     answer = input("Voulez-vous continuer ?(y/n)")
62                     answer = answer.lower()
63                 if answer == 'y':
64                     playerWantPlay = True
65                 else:
66                     playerWantPlay = False
67                     endGame(pseudo, scoreJoueur)
```

Déroulement d'une partie par tour et vérification des conditions de victoire ou défaite.

Ce que ce projet m'a apporté :

Python est le langage de programmation que je connaissais déjà avant le début du BTS, cet exercice, bien que plus long que ce que j'ai pu faire avant, a été très intéressant et amusant.

En plus des consignes, j'ai voulu essayer la lecture, modification, ajout de fichiers ".txt" et l'écriture de code sur plusieurs fichiers.

Ce n'est pas encore de la POO mais j'ai pu mettre au clair mes idées et je pense que ça a pu m'aider dans les autres langages de programmation par la suite.

Ce qui pourrait être amélioré :

- Refactoriser
- Mettre le code en POO
- utiliser tkinter