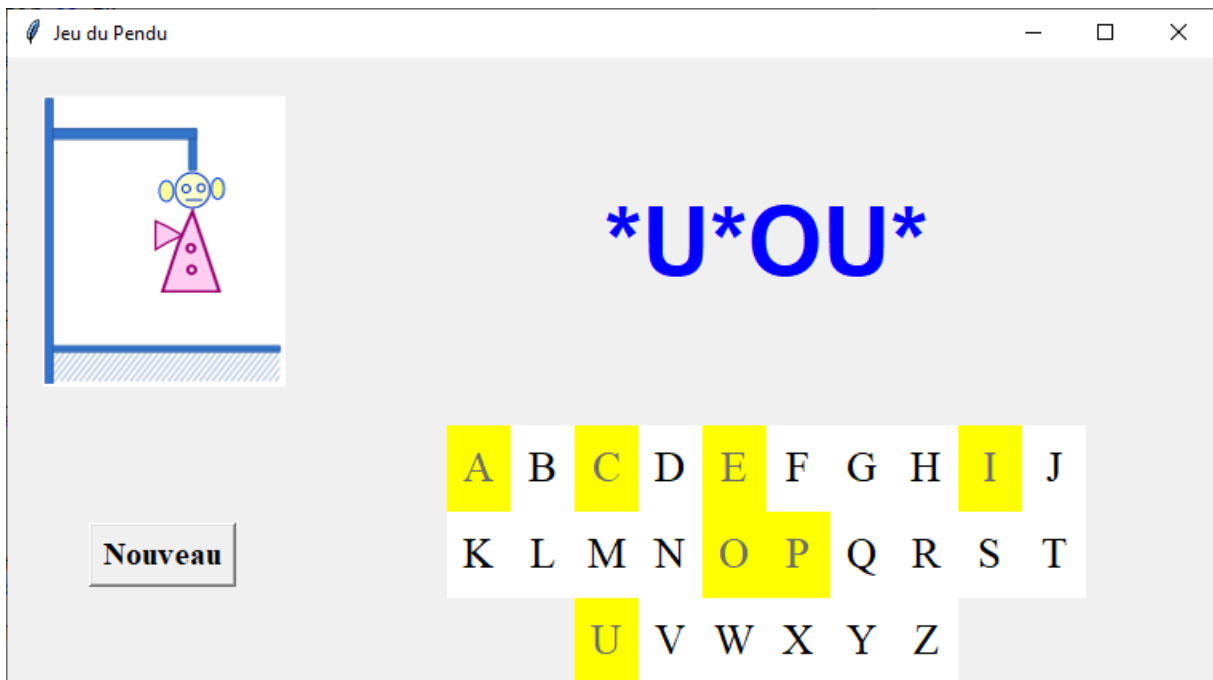


Pendu : Console

Le jeu



Dans ce jeu, l'ordinateur choisit un mot et le joueur doit trouver le mot en proposant différentes lettres.

Si la lettre proposée appartient au mot, l'ordinateur affiche la lettre dans le mot à tous les emplacements où elle apparaît.

Sinon, un nouveau morceau du pendu est affiché.

Lorsque le pendu est complet, le joueur a perdu et l'ordinateur affiche le mot complet.

Si toutes les lettres du mot sont trouvées, le joueur gagne.

Analyse du programme à réaliser

Le programme se déroulera en plusieurs étapes :

1. Choix d'un mot secret dans une liste de mots
2. Affichage d'autant d'étoiles qu'il y a de lettres dans le mot
3. Demander au joueur d'entrer une lettre
4. Vérifier que la lettre fournie, n'a pas déjà été utilisée
5. Si la lettre est dans le mot secret, l'afficher à toutes ses positions
6. Si la lettre n'est pas dans le mot secret, afficher une nouvelle image du pendu
7. Tester si le joueur a gagné

Les images du pendu

Ces images sont constituées de caractères imprimables, répartis sur plusieurs lignes.

Nous placerons ces images dans une liste.

Pour déclarer une chaîne de caractères contenant des sauts de lignes, nous devons utiliser les triples apostrophes. Ainsi la première image du pendu sera :

```
'''
+---+
  |
  |
  |
==='''
```

Et la dernière :

```
'''
+---+
O |
/ \|
/\ |
==='''
```

Nous pouvons construire ainsi 7 images que l'on placera dans une liste IMAGES_PENDU.

IMAGES_PENDU=["première image", "deuxième image", etc.]

La liste des mots secrets et le choix d'un mot

Pour l'instant nous construisons simplement une liste de mot.

```
liste_mots = ['GERANIUM', 'URANUS', 'ORDINATEUR', 'CLAVIER', 'POMME', 'WAHHABITE',
'HORLOGE', 'PSCHENT', 'DICTIONNAIRE', 'ECRAN', 'WHISKY', 'ARMOIRE']
```

Et pour choisir un mot dans cette liste, il suffit d'utiliser la fonction choice du module random.

```
secret = random.choice(liste_mots)
```

Le mot en cours de progression

Au départ ce mot est une liste constituée d'autant d'étoiles que le mot secret comporte de lettres.

```
motProgression = list("." * len(secret))
```

Pour l'afficher sans que les éléments de la liste soit entouré de crochets, nous transformerons cette liste en une chaîne de caractères par :

```
print("".join(motProgression))
```

Ensuite lorsque l'utilisateur va proposer une lettre contenue dans le mot secret, il faut remplacer une ou plusieurs étoiles par la lettre.

Il suffit de faire une boucle qui teste chaque lettre du mot secret.

```
for i in range(len(secret)):
    if secret[i]==lettre:
        motProgression[i]= lettre
```

Les lettres déjà proposées

Il faut empêcher que le joueur propose plusieurs fois la même lettre.

Nous placerons donc dans une liste, toutes les lettres qui ont déjà été proposées.

Cette liste permettra de tester la lettre entrée par le joueur et elle sera affichée à chaque nouveau tour.

Elle sera mise à jour par :

```
dejaProposee.append(lettre)
```

et elle sera affichée par :

```
print("".join(dejaProposee))
```

Le programme

Vous trouverez sur le site, un squelette du programme, vous évitant de saisir la liste des images.

```
# -----boucle principale
```

```
#le mot à trouver
```

```
secret = ""
```

#le mot affiché au fur et à mesure que des lettres sont bien placée

motProgression=[]

#liste des lettres déjà proposées

dejaProposee=[]

#définit le numéro de l'image du pendu

numEtape=-1

#la partie n'est pas finie

fin = False

while True:

#initialisation de toutes les données

#choix du mot secret, création de la liste motProgression

init()

while not fin:

#affiche l'état du jeu

afficher_jeu()

#demander une lettre au joueur

uneLettre = choisir_lettre()

#mettre à jour la liste motProgression

maj_motProgres(uneLettre)

#tester cette étape : le jour a t'il gagné, perdu ?

#Si le jeu n'est pas fini, passer à l'étape suivante

test_etape(uneLettre)

print('Une autre partie? (oui / non)')

if input().lower().startswith('o'):

init()

else:

break

Les fonctions

Initialisation : cette fonction doit réinitialiser toutes les données pour entamer une nouvelle partie.

#initialisation avant une nouvelle partie

```
def init():
    global secret,motProgression,fin,numEtape
    # prend au hasard un mot dans la liste
    secret = rn.choice(liste_mots)
    #constitue le mot en cours de progression
    motProgression = list(''.join("*"*len(secret)))
    #nettoie la liste des lettres proposées
    dejaProposee.clear()
    #on repart sur une nouvelle partie
    fin=False
    numEtape = -1
```

Affichage de l'état du jeu : éventuellement une image du pendu et les lettres déjà proposées et le mot en progression

#affiche l'état du jeu

```
def afficher_jeu():
    #si le joueur a déjà proposé une lettre qui n'est pas dans le mot secret
    if numEtape >=0:
        print(IMAGES_PENDU[numEtape])
        print()

    print('Lettres déjà jouées:', end=' ')
    print(''.join(dejaProposee))
    print()
    print(''.join(motProgression))
    print()
```

L'entrée d'une lettre par le joueur

```
def choisir_lettre():
    global fin
    # Renvoie la lettre entrée par le joueur
    # Cette fonction s'assure que le joueur n'a pas entré une lettre déjà jouée
    # Elle s'assure aussi que le joueur entre soit une lettre, soit 0
    # Si 0, cela indique que le joueur veut arrêter avant la fin du jeu
    while True:
        print('Donnez une lettre.')
        lettre = input()
        lettre = lettre.upper()
        if len(lettre) != 1:
            print("'Ne donnez qu'une seule lettre'")
        elif lettre in dejaProposee:
            print('Vous avez déjà proposé cette lettre')
        elif lettre == '0':
            fin = True
            return lettre
        elif lettre not in 'ABCDEFGHIJKLMNOPQRSTUVWXYZ':
            print('Entrez une lettre')
        else:
            return lettre
```

Mise à jour du mot en progression : si besoin on remplace une * par la lettre proposée

Met à jour le motProgression: remplace éventuellement une * par la lettre

```
def maj_motProgres(lettre):
    for i in range(len(secret)):
        if secret[i] == '*':
            motProgression[i] = lettre
```

Tester la proposition de lettre

#test si la lettre jouée fait gagner ou perdre le joueur

def test_etape(lettre):

 global numEtape, fin

#met à jour la liste des lettres déjà jouées

 dejaProposee.append(lettre)

#si la lettre n'est pas dans le mot, on passe à l'étape suivante

 if lettre not in secret:

 numEtape+=1

 return

#tester si la lettre fait gagner le joueur

 j=0

 for i in range (len(motProgression)):

 if motProgression[i]!='*':

 j+=1

 if j == len(motProgression):

 fin = True

 afficher_jeu()

 print("Vous avez gagné")

 return

#on en était à la dernière étape

 if numEtape==7:

 fin = True

 print("Vous avez perdu")

 print("Le mot a trouvé était : ",secret)