

Fiche de TP numéro 4 - Le jeu du Pendu – Groupe 4 – Bonus

Aujourd'hui, vous allez programmer le jeu du pendu. Voici un exemple d'exécution du jeu :

```
LE PENDU

+---+
|   |
|   |
|   |
|   |
=====

Erreurs :

- - - - -

Proposez une lettre : a

+---+
|   |
|   |
|   |
|   |
=====

Erreurs :

_ a _ _ a _ _ a

Proposez une lettre : f

...

+---+
|   |
O   |
/|\  |
|   |
|   |
=====

Erreurs :
f e i p

b a r r a c u _ a

Proposez une lettre : d
Vous avez gagné !
Le mot secret était : barracuda
```

Nous allons décomposer le problème en plusieurs sous-problèmes, où chacun correspond à une des fonctions suivantes :

- choisit_mot : pour choisir le mot secret.
- input_lettre : pour demander une lettre au joueur.
- dessine_ecran : pour dessiner l'écran du jeu.
- gagne : pour tester si le joueur a gagné
- main : la fonction principale.

La fonction dessine_ecran sera elle-même décomposée en plusieurs fonctions.

N'oubliez pas de documenter et tester vos fonctions !

Préambule

Vous avez accès à deux fichiers `mots.py` et `figures_pendu.py` sur Moodle. Ces deux fichiers contiennent, respectivement :

- une liste de mots qui va servir de base pour jouer
- les figures du pendu pour dessiner l'état du joueur

Pour pouvoir les utiliser, téléchargez ces deux fichiers dans le même répertoire que le fichier `tp5_pendu.py` que vous allez créer.

Puis, commencez votre fichier `tp5_pendu.py` par les lignes suivantes ;

```
# Nom Prénom
# groupe X

#####
# importations
#####

from mots import MOTS
from figures_pendu import FIGURES_PENDU
import random

#####
# fonctions
#####
```

Vous pourrez maintenant utiliser les constantes `MOTS` et `FIGURES_PENDU` comme si elles avaient été définies dans votre fichier.

Choisir le mot secret

Spécifiez puis écrivez la fonction `choisit_mot` qui reçoit une liste de chaînes de caractères en argument et qui **retourne** une chaîne aléatoirement choisie dans la liste.

Exemples :

```
>>> choisit_mot(['alligator', 'escargot'])
'alligator'
>>> choisit_mot(['alligator', 'escargot'])
'escargot'
```

Demander une lettre au joueur

Spécifiez puis écrivez la fonction `est_dans` qui teste si un caractère apparaît dans une chaîne de caractères.

Spécifiez puis écrivez la fonction `input_lettre` qui reçoit une chaîne de caractères `props` des lettres déjà proposées en paramètre et qui retourne la nouvelle lettre choisie par le joueur. La fonction doit demander à l'utilisateur une lettre. Si l'utilisateur fait une erreur, i.e. saisir plus d'une lettre, saisir autre chose qu'une lettre, ou saisir une lettre qui appartient à `props` ; la fonction doit afficher un message d'erreur adapté à chaque situation et redemander une nouvelle lettre.

Exemple :

```
>>> input_lettre('abc')
Proposez une lettre : dd
Proposez une seule lettre, s'il vous plaît.
Proposez une lettre : a
Vous avez déjà proposé cette lettre.
Entrez une lettre : 0
0 n'est pas une lettre.
proposez une lettre : e
'e'
>>>
```

Dessiner l'écran du jeu

La fonction qui dessine l'écran du jeu est un peu plus compliquée. Nous allons donc la décomposer en plusieurs fonctions :

- `dessine_pendu` : affiche un dessin du pendu.
- `affiche_erreurs` : affiche les erreurs du joueur.
- `affiche_correctes` : affiche les lettres correctes.

1 Dessiner le pendu

Spécifiez puis écrivez la fonction `dessine_pendu` qui reçoit un entier `n` en argument et qui ne retourne rien. La fonction doit afficher la n -ième figure du pendu.

La meilleure façon de dessiner le pendu est d'utiliser une liste de dessins. Dans notre cas, nous pouvons utiliser une liste de chaînes de caractères où chaque élément correspond à un des dessins du pendu. Par exemple, la liste ci-dessous contient deux figures du pendu :

```
FIGURES_PENDU = [ '''
+---+
|   |
|   |
|   |
|   |
=====
+---+
|   |
O   |
/|\  |
/ \  |
=====
''' ]
```

L'instruction `print (FIGURES_PENDU[0])` affichera la première figure du pendu. Vous pouvez récupérer un fichier avec les figures du pendu sur Moodle.

2 Afficher les erreurs

Spécifiez puis écrivez la fonction `affiche_erreurs` qui reçoit une chaîne de caractères `erreurs` en argument et qui ne retourne rien. La fonction doit afficher les éléments de la chaîne comme les erreurs du joueur.

Exemples :

```
>>> affiche_erreurs('abc')
Erreurs :
a b c
>>> affiche_erreurs('efgdx')
Erreurs :
e f g d x
```

3 Afficher les lettres correctes

Spécifiez puis écrivez la fonction `affiche_correctes` qui reçoit deux chaînes de caractères `correctes` et `mot_secret` et qui affiche les lettres correctes entrées par le joueur.

Exemples :

```
>>> affiche_correctes('elc', 'ecurreil')
e c _ _ _ e _ l
>>> affiche_correctes('lia', 'alligator')
a l l i _ a _ _ _
```

4 Tester la fin de partie

Spécifiez puis écrivez la fonction `gagne` qui prend en paramètre le mot secret et les propositions faites par le joueur et retourne vrai si le joueur a trouvé le mot secret (faux sinon).

Exemples :

```
>>> gagne("fgar", "fanfaronner")
False
>>> gagne("ncoepklfar", "fanfaronner")
True
```

La fonction principale

Spécifiez puis écrivez la fonction principale du programme (`main`) qui ne reçoit pas d'arguments et qui ne retourne rien. La fonction doit utiliser les fonctions décrites auparavant pour :

1. Afficher l'entrée du jeu.
2. Choisir un mot secret.
3. Tant que le nombre max d'erreurs n'est pas atteint et que le joueur n'a pas gagné :
 - (a) Dessine le pendu.
 - (b) Affiche les erreurs.
 - (c) Affiche les lettres correctes.
 - (d) Demande au joueur de proposer une lettre.
 - (e) Met à jour les propositions et les erreurs en fonction de cette lettre.
4. Affiche si le joueur a gagné ou pas et affiche le mot secret.