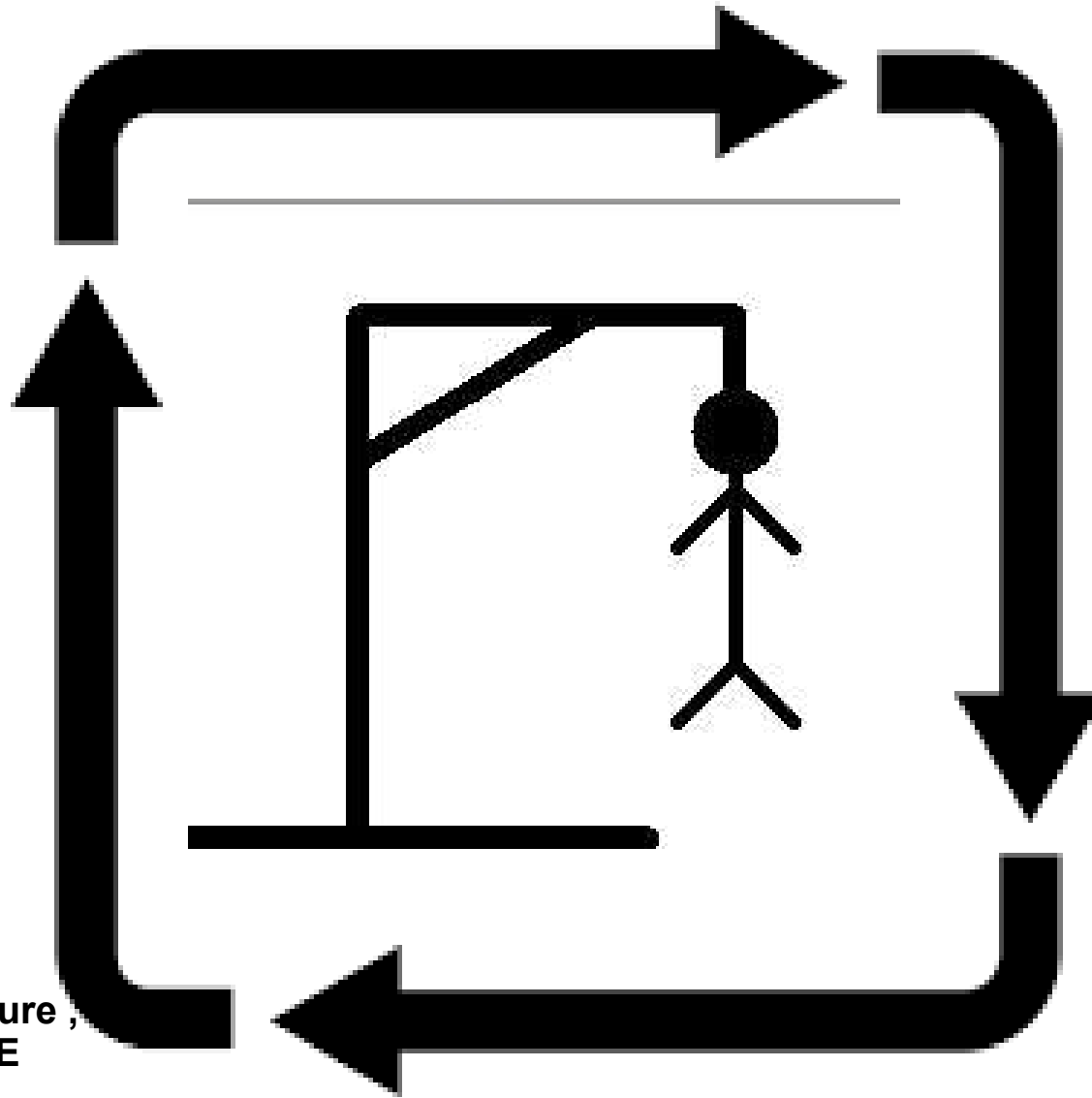


Projet : Autour du jeu « le pendu »



DELROT Sylvain
HOUNKPATIN Bonaventure ,
MORARD – LAVALLETTE
Laurence
SAUVERGEAT Magali

BLOC 1 – DIU

Pré-requis - notions

Pré-requis :

- Bases de Python (boucles, listes, dictionnaires)
- Bases HTML et CSS (générer un tableau)

Notions abordées :

- les représentations de données
- le traitement des données en tables
(recherche dans une table)
- Indexation de tables
- les modalités de l'interaction entre l'homme et la machine (code HTML)

Position et durée

Début Décembre environ

Durée approximative :- > 5 semaines

- Séance Cours / TD : 2h TP + 2h de Correction
- Séance Projet : 6H
- Séance Finale : de 2H à 4H

Projet de 3 élèves

Séance 1 : « le pendu minimum »

Séance 1 : 2 Heures de TP/TD guidé

Objectif : Pendu en mode texte et mot caché fixe.

- `motcache="octogone"`
- Les tableaux pas des « string »=> liste de lettres.

Fonctions à coder :

- `genere_tirets / lettre_presente,`
- `devoiler_lettre / teste_gagne`
- `prog_principal`

Solution donnée en cours

Les 3 séances de projet

Des groupes de 3 élèves

- Chaque élève a en charge une partie différente
- Le groupe doit s'entraider.
- L'évaluation sera la moyenne d'une note personnelle et d'une note d'équipe.

Point technique

Utilisation d'un dictionnaire des fréquences des lettres de l'alphabet

```
frequency = {'a' : 0.083944, 'b' : 0.007669,  
'c' : 0.033297, 'd' : 0.040699, 'e' :  
0.145037, 'f' : 0.012109, 'g' : 0.009495, 'h' :  
0.007973, 'i' : 0.081828, 'j' : 0.006377,  
'k' : 0.000638, 'l' : 0.058405, 'm' : 0.029355,  
'n' : 0.075570, 'o' : 0.053669, 'p' :  
0.032087, 'q' : 0.012613, 'r' : 0.070209,  
's' : 0.080091, 't' : 0.074775, 'u' : 0.059808,  
'v' : 0.015791, 'w' : 0.000067, 'x' :  
0.004098, 'y' : 0.003155, 'z' : 0.001240}
```

Utilisation : cryptographie, reconnaissance de la langue utilisée dans un texte

Fréquence \neq Probabilité

La fréquence d'une lettre dans un texte

- La probabilité d'apparition d'une lettre dans un mot
- Exemple 5 mots de 10 lettres:
 - Qualifiait / nombreuses / cataplasme
 - Obtiennent / supporteur

Fréquence du a : 5 / 50 probabilité de a : 2 / 5

- Fréquence du e : 6 / 50 probabilité de e : 4 / 5
- Fréquence du t : 5 / 50 probabilité de t : 4 / 5
- Ordre décroissant :
 - Fréquence : e-a-t-i-n-s
 - Probabilité : e-t-o-s-u-a

Cahier des Charges: Élève 1

- 4 fonctions
- choisit_mot : lit un fichier de mots
 - retourne un mot tiré au hasard
- normalise : reçoit mot avec accents
 - retourne un mot sans accent
- demo_v1 : reçoit les lettres déjà proposées
 - retourne une nouvelle lettre aléatoire
- demo_v2 : reçoit les lettres déjà proposées
 - retourne la lettre la plus fréquente suivante

Cahier des Charges: Élève 2

Dessiner le pendu via le module Turtle

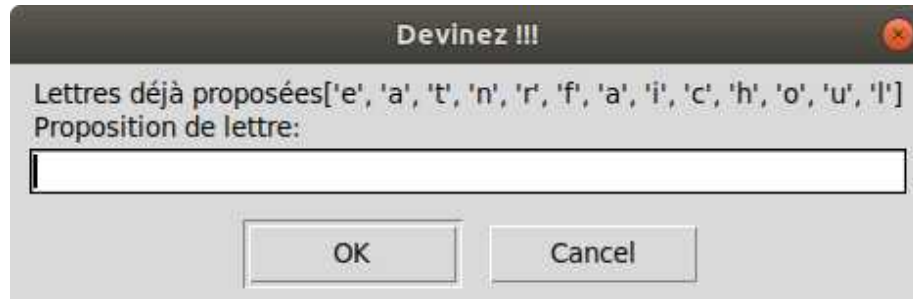
Création d'une listes de coordonnées :

```
Placement = [ (0,-70), (0,-100), # texte  
              (-150,-30), (50,-30), #plancher  
              (-50,-30), (-50,150), #barre verticale  
              (-50,150), (80,150), #barre horiz  
              (... ) ]
```

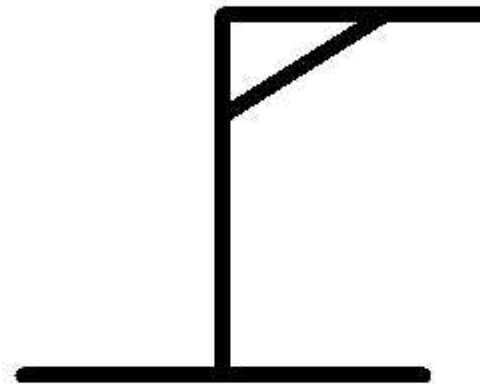
2 fonctions :

- **dessin_pendu** : prend en paramètre l'étape
dessine la partie du pendu
- **dessin_texte** : prend le texte à écrire

Cahier des Charges: Élève 2



Le Pendu DIU-Bloc1 Orsay



su-anne

lettres proposées : eanrlmsou

Cahier des Charges: Élève 3

Générer une page HTML et CSS
tableau du dictionnaire de données

Table des fréquences de l'alphabet en français

Lettre	% de Fréquence	Lettre	% de Fréquence
a	7.80	n	7.52
b	1.56	o	5.91
c	4.18	p	2.90
d	2.41	q	0.62
e	16.31	r	8.79
f	1.40	s	6.86
g	1.72	t	7.37
h	1.22	u	4.50
i	8.83	v	1.39
j	0.24	w	0.04
k	0.11	x	0.44

DII-Bloc1

Cahier des Charges: Élève 3

- 3 fonctions
- `gen_html` :
 - reçoit le dictionnaire des fréquences
 - Génère la page HTML
- `Calcul_fréquence` : lit un fichier de mots
 - Compte le nombre de chaque lettre
 - Calcule la fréquence, construit le dictionnaire
- `Calcul_probabilité` : lit un fichier de mots
 - Compte le nombre d'apparition d'une lettre
 - Calcule la probabilité, construit le dictionnaire

Séance finale

- Intégration des 3 modules
 - Ajout du choix : mode démo / mode joueur
- Réalisation d'un compte rendu.
 - Chaque élève donne le listing de son code
 - Commentaires
 - Montre les tests de ses fonctions
- Partie commune :
 - Code du programme principal
 - Tests des différents fonctionnalités

Évaluation

- Élève :
 - Degré d'autonomie
 - La clarté du code
 - Parcours des listes
 - Les tests expliqués dans le CR
- Groupe :
 - Dynamique et cohésion du groupe
 - Intégration valide
 - Les tests expliqués dans le CR

Élève 1 et 3 – partie avancée

- Réfléchir à des fonctions pour améliorer le mode démo.
 - Gestion des voyelles
 - Gestions des associations de lettres
- création d'un dictionnaire de dictionnaire d'association

Élève 2 – partie avancée

- Réfléchir à l'utilisation du gestionnaire d'évènements :

```
turtle.listen()
```

```
turtle.onscreenclick(quitte, 3)
```

```
turtle.onscreenclick(reinit, 2)
```

```
turtle.onkeypress(gestion_evea, "a")
```

```
turtle.onkeypress(gestion_eveb, "b")
```


Remarques

- Les solutions proposées : pas optimum
- Merci de faire des propositions de modification
- Ne pas fournir les corrigés aux élèves
- Sinon les solutions se trouveront sur Internet

Questions

