IUT de Cachan LPGMP

DM : Création de mini-jeux

Projet de Workshop de la semaine du 6 janvier 2020

Table des matières

1. Obligations

- Lire entièrement le sujet
- Suivre les consignes
- Respecter les règles de rendu

2. Généralités

2.1 Questions

Questions	alexandre.th.manuel@gmail.com
Rendu Final	Dimanche 12/01 @ 23h42

Envoyez-moi un mail si vous rencontrez des difficultés.

2.2 La Triche

Les cas de triche seront sévèrement sanctionnés. Les tricheurs verront leur note multipliée par 0 et l'administration sera prévenue.

Tricher veut dire ne pas savoir expliquer votre code si on vous le demande. Vous avez en revanche le droit, et êtes même invités à collaborer avec vos camarades.

Ne prenez pas le risque de rendre un code que vous ne comprenez pas.

2.3 L'évaluation

Chaque mini jeu rapport le même nombre de points.

2.4 Prérequis

- Connaissances de Python :
  - Variables
  - Listes et séquences
  - Structures conditionnelles
  - Boucles

2.5 Compétences à acquérir

- Mise en place d'un projet
- Réalisation
- Rendu rigoureux

2.6 Rendu du projet

Vous devrez envoyer votre projet par mail **avant dimanche 12/01 à 23h42**.

Votre mail devra être constitué comme suit :

Destinataire	`alexandre.th.manuel@gmail.com`
Objet	`[IUTC] [LPGMP] [MINIGAMES] Rendu de Prénom Nom`

Contenu	`Bonjour,`
	
	`Vous trouverez en pièce jointe mon rendu pour le workshop Minigames.`
	
	`[…]`
	
	`Cordialement,`
	
	`--`
	`Prénom Nom`

Pièce jointe	`nom.prenom.zip`

Vous remplacerez :

- Prénom par votre prénom
- Nom par votre nom de famille
- […] par ce que vous voulez

Si vous en avez, remplacez les espaces dans votre prénom/nom par des tirets (-) et retirez les accents.

**/ ! \ Les deux tirets précédant la signature sont suivis d'un espace. "-- " / ! **

Votre archive nom.prenom.zip devra contenir **uniquement** ces 3 fichiers :

- plus_ou_moins.py
- mot_mystere.py
- pendu.py

Tout manquement à une de ces règles de rendu entraînera de VIOLENTES pertes de points !

Vous recevrez une confirmation de rendu de ma part. Considérez que vous n'avez PAS RENDU tant que vous n'avez pas cette confirmation !

Bon courage à tous !

3. Le projet

Vous allez devoir créer trois mini jeux :

- Plus ou moins
- Mot mystère
- Pendu

3.1 Plus ou moins

Nous arrivons maintenant dans le premier TP. Le but est de vous montrer que vous savez faire des choses avec ce que je vous ai appris. Car en effet, la théorie c'est bien, mais si on ne sait pas mettre tout cela en pratique de manière concrète… ça ne sert à rien d'avoir passé tout ce temps à apprendre.

Croyez-le ou non, vous avez déjà le niveau pour réaliser un premier programme amusant. C'est un petit jeu en mode console. Le principe du jeu est simple et le jeu est facile à programmer. C'est pour cela que j'ai choisi d'en faire le premier TP du cours.

3.1.1 Le principe

Avant toute chose, il faut que je vous explique en quoi va consister notre programme.

C'est un petit jeu que j'appelle « Plus ou moins ».

Le principe est le suivant.

1. L'ordinateur tire au sort un nombre entre 1 et 100.
2. Il vous demande de deviner le nombre. Vous entrez donc un nombre entre 1 et 100.
3. L'ordinateur compare le nombre que vous avez entré avec le nombre « mystère » qu'il a tiré au sort. Il vous dit si le nombre mystère est supérieur ou inférieur à celui que vous avez entré.
4. Puis l'ordinateur vous redemande le nombre.
5. Il vous indique si le nombre mystère est supérieur ou inférieur.
6. Et ainsi de suite, jusqu'à ce que vous trouviez le nombre mystère.

Le but du jeu, bien sûr, est de trouver le nombre mystère en un minimum de coups.

Voici une « capture d'écran » d'une partie, c'est ce que vous devez arriver à faire :

```
Quel est le nombre ? 50
C'est plus !
Quel est le nombre ? 75
C'est plus !
Quel est le nombre ? 85
C'est moins !
Quel est le nombre ? 80
C'est moins !
Quel est le nombre ? 78
C'est plus !
Quel est le nombre ? 79
Bravo, vous avez trouvé le nombre mystère !!!
```

3.1.2 Tirer un nombre au sort

Certes, nous ne savons pas générer un nombre aléatoire. Il faut dire que demander cela à l'ordinateur n'est pas simple : il sait bien faire des calculs, mais lui demander de choisir un nombre au hasard, ça, il ne sait pas faire !

En fait, pour « essayer » d'obtenir un nombre aléatoire, on doit faire faire des calculs complexes à l'ordinateur… ce qui revient au bout du compte à faire des calculs !

Bon, on a donc deux solutions.

Soit on demande à l'utilisateur d'entrer le nombre mystère via un input d'abord. Ça implique qu'il y ait deux joueurs : l'un entre un nombre au hasard et l'autre essaie de le deviner ensuite.

Soit on tente le tout pour le tout et on essaie quand même de générer un nombre aléatoire automatiquement. L'avantage est qu'on peut jouer tout seul du coup. Le défaut… est qu'il va falloir que je vous explique comment faire ! Nous allons tenter la seconde solution.

Pour générer un nombre aléatoire, on utilise la fonction randint. Cette fonction génère un nombre au hasard entre deux bornes.

Voici comment s'en servir :

```
import random
x = random.randint(1, 101)
```

Le `import random` est à mettre une seule fois au début de votre programme. Le `random.randint(min, max)` est à mettre chaque fois que vous voulez tirer un nombre au sort.

Bref, vous en savez assez. Je vous ai expliqué le principe du programme, je vous ai fait une capture d'écran du programme au cours d'une partie.

Avec tout ça, vous êtes tout à fait capables d'écrire le programme.

3.2 Mot mystère

Le sujet de ce TP n'est pas très compliqué mais promet d'être amusant : nous allons mélanger les lettres d'un mot et demander à un joueur de retrouver le mot « mystère » qui se cache derrière ces lettres (figure suivante).

```
Saisissez un mot
MYSTERE

Quel est ce mot ? YTMREES
RESTEMY
Ce n’est pas le mot !

Quel est ce mot ? YTMREES
MYSTERE
Bravo !
```

3.2.1 Le principe

Nous voulons réaliser un jeu qui se déroule de la façon suivante :

1. Le joueur 1 saisit un mot au clavier ;
2. L'ordinateur mélange les lettres du mot ;
3. Le joueur 2 essaie de deviner le mot d'origine à partir des lettres mélangées.

```
Saisissez un mot
MYSTERE

Quel est ce mot ? YTMREES
RESTEMY
Ce n’est pas le mot !

Quel est ce mot ? YTMREES
MYSTERE
Bravo !
```

1. Dans cette partie, le joueur 1 choisit « MYSTERE » comme mot à deviner.
2. L'ordinateur mélange les lettres et demande au joueur 2 de retrouver le mot qui se cache derrière « MSERETY ».
3. Le joueur 2 essaie de trouver le mot. Ici, il y parvient au bout de 3 essais :
    a. RESEMTY : on lui dit que ce n'est pas cela
    b. MYRESTE : là non plus
    c. MYSTERE : là on lui dit bravo car il a trouvé, et le programme s'arrête.

Bien sûr, en l'état, le joueur 2 peut facilement lire le mot saisi par le joueur 1. Nous verrons à la fin du TP comment nous pouvons améliorer cela.

3.2.2 Quelques conseils pour bien démarrer

Quand on lâche un débutant dans la nature la première fois, avec comme seule instruction « Allez, code-moi cela », il est en général assez désemparé.

« Par quoi dois-je commencer ? », « Qu'est-ce que je dois faire, qu'est-ce que je dois utiliser ? ». Bref, il ne sait pas du tout comment s'y prendre et c'est bien normal vu qu'il n'a jamais fait cela.

Mais moi, je n'ai pas envie que vous vous perdiez ! Je vais donc vous donner une série de conseils pour que vous soyez préparés au mieux. Bien entendu, ce sont juste des conseils, vous en faites ce que vous voulez.

3.2.3 Repérer les étapes du programme

Je vous ai décrit un peu plus tôt les 3 étapes du programme :

1. Saisie du mot à deviner ;
2. Mélange des lettres ;
3. Boucle qui se répète tant que le mot mystère n'a pas été trouvé.

Ces étapes sont en fait assez indépendantes. Plutôt que d'essayer de réaliser tout le programme d'un coup, pourquoi n'essayeriez-vous pas de faire chaque étape indépendamment des autres ?

1. étape 1 : l'utilisateur doit saisir un mot qu'on va stocker en mémoire. Si vous connaissez input, vous ne mettrez pas plus de quelques minutes à écrire le code correspondant.
2. étape 2 : vous avez une string qui contient un mot comme « MYSTERE » et vous voulez aléatoirement mélanger les lettres pour obtenir quelque chose comme « MSERETY ». Comment faire ? Google est votre ami :) « shuffle string in python » devrait faire l'affaire.
3. étape 3 : vous devez créer une boucle qui demande de saisir un mot et qui le compare au mot mystère. La boucle s'arrête dès que le mot saisi est identique au mot mystère.

3.2.4 Bonus : Piochez le mot dans un fichier-dictionnaire

Pour pouvoir jouer seul, vous pourriez créer un fichier contenant une série de mots (un par ligne) dans lequel le programme va piocher aléatoirement à chaque fois. Voici un exemple de fichier-dictionnaire :

```
MYSTERE
XYLOPHONE
ABEILLE
PLUTON
MAGIQUE
AVERTISSEMENT
```

Au lieu de demander le mot à deviner (étape 1) on va chercher dans un fichier comme celui-ci un mot aléatoire. À vous d'utiliser les fonctionnalités de lecture de fichiers que vous avez apprises ! … Allez, puisque vous m'êtes sympathiques, je vous fournis même un fichier-dictionnaire tout prêt avec des dizaines de milliers de mots !

3.3 Jeu du pendu

Je vous propose de réaliser un Pendu. C'est un grand classique des jeux de lettres dans lequel il faut deviner un mot caché lettre par lettre.

3.3.1 Le principe

Je tiens à ce qu'on se mette bien d'accord sur les règles du Pendu à réaliser. Je vais donc vous donner ici les consignes, c'est-à-dire vous expliquer comment doit fonctionner précisément le jeu que vous allez créer.

Tout le monde connaît le Pendu, n'est-ce pas ? Allez, un petit rappel ne peut pas faire de mal : le but du Pendu est de retrouver un mot caché en moins de 10 essais (mais vous pouvez changer ce nombre maximal pour corser la difficulté, bien sûr !).

3.3.2 Déroulement d'une partie

Supposons que le mot caché soit ROUGE.

Vous proposez une lettre à l'ordinateur, par exemple la lettre A. L'ordinateur vérifie si cette lettre se trouve dans le mot caché.

> *Rappelez-vous :* Pour vérifier si un caractère est présent dans une chaîne de caractères, vous pouvez utiliser le mot clé « in » : `if 'o' in 'mot'`

À partir de là, deux possibilités :

- la lettre se trouve effectivement dans le mot : dans ce cas, on dévoile le mot avec les lettres qu'on a déjà trouvées ;
- la lettre ne se trouve pas dans le mot (c'est le cas ici, car A n'est pas dans ROUGE) : on indique au joueur que la lettre ne s'y trouve pas et on diminue le nombre de coups restants. Quand il ne nous reste plus de coups (0 coup), le jeu est terminé et on a perdu.

> Dans un « vrai » Pendu, il y aurait normalement le dessin d'un bonhomme qui se fait pendre au fur et à mesure que l'on fait des erreurs. En console, ce serait un peu trop difficile de dessiner un bonhomme qui se fait pendre rien qu'avec du texte, on va donc se contenter d'afficher une simple phrase comme « Il vous reste X coups avant une mort certaine ».

Supposons maintenant que le joueur tape la lettre G. Celle-ci se trouve dans le mot caché, donc on ne diminue pas le nombre de coups restants au joueur. On affiche le mot secret avec les lettres qu'on a déjà découvertes, c'est-à-dire quelque chose comme ça :

```
Mot secret : ***G*
```

Si ensuite on tape un R, comme la lettre s'y trouve, on l'ajoute à la liste des lettres trouvées et on affiche à nouveau le mot avec les lettres déjà découvertes :

```
Mot secret : R**G*
```

3.3.3 Le cas des lettres multiples

Dans certains mots, une même lettre peut apparaître deux ou trois fois, voire plus !

Par exemple, il y a deux Z dans PUZZLE ; de même, il y a trois E dans ELEMENT.

Que fait-on dans un cas comme ça ? Les règles du Pendu sont claires : si le joueur tape la lettre E, toutes les lettres E du mot ELEMENT doivent être découvertes d'un seul coup :

```
Mot secret : E*E*E**
```

Il ne faut donc pas avoir à taper trois fois la lettre E pour que tous les E soient découverts.

3.3.4 Exemple d'une partie complète

Voici à quoi devrait ressembler une partie complète en console lorsque votre programme sera terminé :

```
Bienvenue dans le Pendu !

Il vous reste 10 coups à jouer
Quel est le mot secret ? ******
Proposez une lettre : E

Il vous reste 9 coups à jouer
Quel est le mot secret ? ******
Proposez une lettre : A

Il vous reste 9 coups à jouer
Quel est le mot secret ? *A****
Proposez une lettre : O

Il vous reste 9 coups à jouer
Quel est le mot secret ? *A**O*
Proposez une lettre :
```

Et ainsi de suite jusqu'à ce que le joueur ait découvert toutes les lettres du mot (ou bien qu'il ne lui reste plus de coups à jouer) :

```
Il vous reste 8 coups à jouer
Quel est le mot secret ? MA**ON
Proposez une lettre : R

Gagne ! Le mot secret etait bien : MARRON
```

3.3.5 Liste de mots

Dans un premier temps pour vos tests, je vais vous demander de fixer le mot secret directement dans votre code. Vous écrirez donc par exemple :

```
mot_secret = 'MARRON'
```

Alors oui, bien sûr, le mot secret sera toujours le même si on laisse ça comme ça, ce qui n'est pas très rigolo. Je vous demande de faire comme ça dans un premier temps pour ne pas mélanger les problèmes. En effet, une fois que votre jeu de Pendu fonctionnera correctement (et seulement à partir de ce moment-là), vous attaquerez la seconde phase : la création de la liste de mots.

La liste de mots, c'est comme son nom l'indique, une liste qui contient des mots. Une fois votre liste initialisée, vous devrez en tirer un au sort en début de partie, grâce à randint qu'on a découvert dans le jeu du plus ou moins.

3.3.6 Bonus

Vous pouvez, en guise de bonus, construire votre liste de mots à partir du fichier dictionnaire du mot mystère.

3.3.7 Conclusion

Je vous laisse un peu réfléchir à tout cela, je ne vais pas trop vous aider quand même, sinon ça ne serait plus un TP ! Sachez que vous avez acquis toutes les connaissances qu'il faut dans les chapitres précédents, vous êtes donc parfaitement capables de réaliser ce jeu. Ça va prendre plus ou moins de temps et c'est moins facile qu'il n'y paraît, mais en vous organisant correctement (et en créant suffisamment de fonctions), vous y arriverez.

Bon courage, et surtout : per-sé-vé-rez !

À vous de jouer ! Bonne chance !