

## Dragon vole

Vous débutez en programmation, et vous voulez commencer avec Python. Vous êtes un fan du Seigneur des anneaux et nostalgique des parties de « dungeons and dragons » de votre lycée ? Vous aimeriez encore faire des parties endiablées, mais le tirage des personnages vous rebute, car il est trop complexe ? Heureusement, une fois de plus, FREELOG est là...

Le hasard...

Hé oui, pour jouer à ce type de jeux, il faut maîtriser le hasard ;-). Sous certaines conditions et avec Python, c'est plus facile qu'il n'y paraît. En effet, pour ceux qui ne le connaissent pas, ce type de jeu se joue avec des dés comportant 4,6,8,10,12, et 20 faces. Plusieurs jets de dés caractérisent un personnage de façon aléatoire (Voir plus bas les caractéristiques...). Puis ce personnage virtuel joue son propre rôle dans une aventure imaginaire. D'autres jets de dés permettront de façonner le destin de cet aventurier virtuel, en indiquant la réussite ou l'échec de ses actions en fonctions des potentialités des personnages et de leurs adversaires ou des événements. Pour faire plus « vrai », nous allons doter notre personnage de diverses manies, ou d'un passé tumultueux (ou rangé), de telle sorte qu'un véritable personnage soit créé.

Fait bien les choses.

Et nous aussi. Au cours de notre réalisation, nous allons voir la gestion de chiffres aléatoires et les bases élémentaires de la gestion de fichier. Cette réalisation étant particulière, nous ne créerons qu'un squelette d'application que vous enrichirez à votre guise. Nous allons créer 2 parties distinctes : la génération proprement dite du personnage avec des tirages aléatoires, et dans un second temps, nous allons le doter d'une vie sentimentale (pour le moins assez bizarre, mais c'est juste pour l'exemple). Un « perso » comporte 6 caractéristiques de base (force, sagesse, intelligence, dextérité, endurance, ). Chacune de ces qualité est déterminée par le lancer de 3 dés à 6 faces. Nous vous faisons confiance pour le doter d'un passé, de signes distinctifs (signe zodiacal, cicatrices, métier, goûts culinaires... qui correspondront tous à des fichiers différents et spécifiques).

On attaque

Comme à chaque fois, nous commençons par les importations. Python dispose d'un module spécifique de gestion des nombres aléatoires. Notre première ligne est donc :

```
from whrandom import *
```

A partir de là, nous allons définir les fonctions qui correspondront à un lancer de dés. Pour un dé à 6 faces, nous aurons :

```
def d6():  
    resultd6=int(random()*6)+1  
    return resultd6
```

Inutile de dire que nous devons générer des dés à 10 faces (utiles pour les pourcentages) et d'autres dés spécifiques (nous en avons déjà parlé). Tous seront bâtis sur le même modèle. La fonction random génère une « graine » comprise au hasard entre 0 et 1. On la multiplie par 6 et on garde la partie entière. Donc, on récupère un chiffre entre 0 et 5. On ajoute donc 1 pour se situer entre 1 et 6, et on retourne cette valeur à l'appelant. Aussi simple que ça. Pour utiliser ce dé, on code la séquence suivante :

```
strenght=d6()+d6()+d6()  
print 'force de ', strenght
```

Voilà la force définie par le lancer de 3 dés à 6 faces. Nous avons généré la première caractéristique par un jet de dés exactement comme dans le jeu réel. Les puristes critiqueront en disant que la génération aléatoire par procédé informatique n'est pas tout à fait aléatoire, (elle est générée au moyen d'une suite mathématique). Mais pour notre usage, cela suffit. On l'affiche par print. Print est une instruction simple mais essentielle. Profitons de l'occasion qui nous est donnée pour voir une « séquence d'échappement ». Il s'agit tout bêtement d'un caractère particulier placé après un antislash (\). Dans notre cas, la lettre n provoque un retour à la ligne et un saut de ligne, une fois avant et une fois après l'affichage de l'alignement (votre caractère, qui est bien sûr loyal et bon ;-)). On aurait pu mettre la séquence \t pour obtenir une tabulation, bien utile pour les présentations alignées à gauche. Dernière précision pour les vrais débutants, si on fait suivre un print xxx par une virgule, on empêche le retour à la ligne (ce que nous ne faisons pas ici).

```
print '\n alignement lawfull good \n '
```

Vous êtes tous fichés

Ici, nous allons passer à la gestion des caractères par fichier. Le principe comme toujours est simple. Nous allons utiliser des fichiers texte, d'extension .txt que nous pouvons créer avec le notepad. L'intérêt d'un fichier est de stocker une bonne fois pour toutes des données (on peut toujours les modifier, avec le notepad), et de pouvoir les récupérer ultérieurement en totalité. Ce format .txt signifie que seuls les caractères constituent le fichier à l'exclusion de toute autre donnée concernant le format. On stocke du texte brut. Avec le notepad, créons notre fichier « sentim.txt » et dotons le de lignes de textes correspondants aux critères tels que : célibataire; marié 2 enfants; etc... Dans notre exemple, nous avons créé une table de 10 lignes, chacune correspondant à un état sentimental précis. Vous pouvez être plus prolixes que nous, mais à condition de rester sur une seule ligne (et dans les bonnes moeurs ;-)). Attention : tout retour à la ligne est considéré comme une autre ligne de donnée. Pour tirer une ligne au hasard, jetons un dé 10 faces (il est défini comme expliqué plus haut. A vous de travailler) et stockons sa valeur dans la variable sentiment. Il va de soi que si vous mettez 15 lignes de possibilités dans votre fichier, il vous faudra générer un tirage aléatoire entre 1 et 15.

```
sentiment=d10()
```

Opération tonnerre

Et maintenant, on passe aux choses sérieuses : on ouvre le fichier « sentim.txt » pour lire les données contenues. On définit une variable d'affectation de fichier toto. On crée ainsi un objet qui est le fichier (sans le savoir vous faites de la programmation objet). Son nom n'a aucune importance, choisissez le donc explicite (évitons toto ;-)). Reste à ouvrir le fichier en donnant son arborescence complète (modifiez l'arborescence pour correspondre avec votre configuration), et un paramètre (r dans notre cas qui correspond à read; lire).

```
toto=open('c:\Art_Freelog\Perso AD&D\sentim.txt','r')
```

Ce paramètre indique le mode d'ouverture du fichier : r pour la lecture; w pour l'écriture, a pour l'ajout de données. D'autres options sont possibles avec rb, wb, etc... pour ceux qui veulent travailler directement avec les valeurs hexadécimales. Attention, on accède ainsi à l'intérieur d'un fichier, et une corruption des données peut être fatale à certains de vos programmes.

Bouclons la boucle

Une structure classique de boucle va nous permettre de faire une sélection dans la liste en affichant la ligne correspondant au tirage évoqué plus haut. Pour plus de détails sur la structure des programmes, lisez les articles de Gilles Baillou, ils sont excellents.

```
for i in range(1,sentiment):  
    gazpacho=toto.readline()  
vie_sent=toto.readline()
```

Nous savons que sentiment est un chiffre compris entre 1 et 10. Nous allons donc lire les « sentiment » premiers enregistrements. Vous remarquerez que l'on utilise une valeur inutile appelée gazpacho pour stocker les lignes lues, mais qui ne nous intéressent pas. Regardez bien l'indentation. A la fin de la boucle, il ne s'est rien passé de constructif. On lit alors la ligne suivante que l'on affecte à vie\_sent. Vous vous rappelez que nous expliquons toujours pourquoi on rajoute 1 à la borne supérieure des boucles. Hé bien, c'est inutile dans ce cas. Nous lisons tout simplement une ligne en supplément à la fin de la boucle. Il nous suffit alors d'afficher le résultat par un simple print.

```
print vie_sent
```

Rideau

Voilà notre programme terminé ou presque. Pour faire les choses proprement, il convient de fermer le fichier, ce qui est fait par la méthode close() dont vous trouverez la syntaxe ci dessous. Remarquez de quelle façon on indique que c'est toto que l'on ferme, car vous pourriez avoir ouvert plusieurs fichiers à la fois.

```
toto.close()
```

A la prochaine.

Cette fois, nous avons créé juste un embryon de squelette de programme. Complétez le à votre guise et créez d'autres fichiers textes spécifiques pour définir un passé, une marotte ou un tic à votre personnage. Soyez créatif, ce genre de programme nécessite de l'imagination pour être intéressant. Sur le plan technique, vous disposez de tous les éléments nécessaires pour mener à bien votre réalisation. Vous disposez également des notions de base d'utilisation des fichiers. N'hésitez pas à nous envoyer vos réalisations et jouez bien entre copains. C'est bien connu, l'informatique favorise les rapports humains.