

Projet en python™ : le Yahtzee

Le principe du jeu

Le *Yahtzee* est un jeu de dés qui se joue avec 5 dés, et qui peut se jouer à plusieurs joueurs. Le but du jeu est de remplir une grille de treize figures (ou « contrats ») constituées par des combinaisons de 5 dés (ou « mains »), chaque contrat réussi rapportant un certain nombre de points. Pour cela, les joueurs ont le droit à 3 essais par tour de jeu. Le joueur qui cumule le meilleur total de points sur la grille gagne la partie. Les 13 contrats à effectuer sont les suivants :

Contrat	Points
Les « 1 »	somme des 1
Les « 2 »	somme des 2
Les « 3 »	somme des 3
Les « 4 »	somme des 4
Les « 5 »	somme des 5
Les « 6 »	somme des 6
Brelan (au moins 3 dés identiques)	somme des 5 dés
Carré (au moins 4 dés identiques)	somme des 5 dés
Full (3 dés identiques + 2 autres dés identiques)	25
Petite suite (4 dés qui se suivent)	30
Grande suite (5 dés qui se suivent)	40
Yahtzee (5 dés identiques)	50
Chance (n'importe quelle main)	somme des 5 dés

Déroulement d'une partie

Les joueurs jouent chacun leur tour. Chaque joueur a le droit au maximum à trois lancers de dés par tour de jeu. La première fois il lance les cinq dés. Après chaque lancer, le joueur choisit les dés qu'il veut conserver et ceux qu'il veut relancer en fonction de la combinaison qu'il souhaite réaliser. Après le dernier lancer, le joueur doit choisir le contrat qu'il valide. Plusieurs cas de figures se présentent alors :

- la combinaison de dés correspond à un contrat qu'il reste à faire, le joueur peut alors valider ce contrat
- la combinaison de dés ne correspond pas à un contrat restant, le joueur doit sacrifier un contrat de son choix, c'est-à-dire inscrire 0 point pour ce contrat (il peut aussi utiliser le contrat « chance » si celui-ci est encore disponible).

Au bout de treize tours, chaque joueur a alors accompli avec plus ou moins de succès tous ses contrats. De plus, une prime de 35 points est offerte au joueur si la somme de ses contrats « 1 à 6 » est supérieure ou égale à 63. Le vainqueur est celui qui a le plus de points au total.

Travail demandé

Le projet consiste, en implantant les fonctions imposées dans les instructions ci-dessous, à réaliser un programme Python permettant de jouer une partie de *Yahtzee* à un ou deux joueurs selon les règles définies ci-dessus. Après avoir lancer les dés, ils seront affichés en ordre croissant pour faciliter la lecture des combinaisons.

Instruction 0. Faire quelques parties réelles (avec 5 dés réels et un ou des joueurs humains) ou virtuelles (c.f. e.g. <https://yahtzee.game/> ou <https://cardgames.io/yahtzee/>) pour bien comprendre les règles du jeu.

Instruction 1. Implanter une procédure `afficher_des` permettant d'afficher une main de 5 dés.

Instruction 2. Implanter une fonction `generer_lancer`, qui retourne la valeur d'un lancer aléatoire de dé (*indication : utiliser le module `random` de Python*).

Instruction 3. Implanter une procédure `trier_des` permettant de trier une main par valeur croissante (*indication : on pourra reprendre un des tris vus en TP*).

Instruction 4. Implanter une procédure `choisir_des` permettant au joueur de choisir les dés à garder.

Instruction 5. Implanter une procédure `lancer_des`, qui prend deux paramètres, une main et la liste des dés à relancer, et effectue les lancers correspondants aux dés à relancer. Cette procédure devra être utilisée également pour le lancer initial (*indication : utiliser le paramètre de la liste des dés à relancer comme paramètre optionnel, avec une valeur par défaut correspondant à relancer tous les dés*).

Instruction 6. Implanter une procédure `jouer_tour` pour jouer un tour complet (c'est-à-dire 3 lancers de suite avec à chaque tour, possibilité pour le joueur de choisir des dés à garder).

Instruction 7. Utiliser une structure de données pour sauvegarder la liste des contrats avec leur valeur (*indication : on remplira les cases soit avec la valeur du contrat, soit avec `-1` si le contrat n'a pas encore été choisi*), et implanter une procédure `creer_grille` d'initialisation / création de grille.

Instruction 8. Implanter une procédure `afficher_grille`.

Instruction 9. Implanter la procédure `choisir_contrat` pour choisir et valider la combinaison à sauvegarder (*indication : commencer par vérifier que la combinaison choisie est bien disponible*).

Instruction 10. Implanter les fonctions (prenant une main en paramètre) :

- `somme_totale` : retourne la somme totale de tous les dés
- `somme_valeur` : retourne le total des dés ayant comme valeur un entier passé en paramètre
- `est_brelan`, `est_carre`, `est_full`, `est_petite_suite`, `est_grande_suite`, `est_yahtzee` : retournent `True` si la main est respectivement un brelan, un carré, un full, une petite suite, une grande suite et un Yahtzee (*indication : on pourra s'aider d'une fonction auxiliaire renvoyant le nombre de répétition de la valeur qui apparaît le plus dans la main*)

Instruction 11. Implanter une fonction `valider_contrat`, qui remplit la grille en fonction du contrat choisi par le joueur et de sa combinaison obtenue.

Instruction 12. Implanter la boucle principale permettant de jouer une partie à un ou deux joueurs.

De plus, les contraintes suivantes devront impérativement être respectées.

Contrainte sur les structures de données. Une main (ensemble des cinq dés) devra obligatoirement être implanté par une variable de type liste, et une grille devra obligatoirement être implanté par une variable de type dictionnaire (dont les clés seront `"1"`, `"2"`, ..., `"brelan"`, ..., `"chance"`). Il est également imposé de regrouper les grilles des joueurs dans une seule et unique structure (donc dans une liste de grilles, couple ou n -uplets de grilles ou un dictionnaire de grilles), ce qui permettra d'étendre facilement le jeu à plus de deux joueurs.

Contraintes sur la structures des fichiers. Le programme devra être découpé en quatre fichiers Python, chacun ayant un rôle bien défini et regroupant les fonctions correspondantes :

- `affichage.py` : module de gestion des sorties (affichage) du programme ; il contient la (ou les si nécessaires) procédure(s) d'affichage

- `saisie.py` : module de gestion des entrées (saisie au clavier); il contient les fonctions de saisie (saisie qui doit être protégée, *c.f.* ci-dessous)
- `noyau.py` : module contenant toutes les fonctions et procédures nécessaires (excepté celles des entrées/-sorties) au déroulement d'une partie
- `yahtzee.py` : fichier principal, à exécuter pour jouer une partie; les autres modules y sont importés, et la boucle principale de jeu est ensuite implantée

L'import de la totalité des fonctions dans un autre module se fait par l'instruction :

```
from module_a_importer import *
```

Contrainte sur la saisie. Il est de plus demandé de gérer la *protection de la saisie au clavier* : le programme ne devra pas s'interrompre si le joueur ne saisit pas les bons types de données demandés. Par exemple, si on saisit « `!!?3@nimp&#%` » comme numéro de dé, le programme ne devra pas s'arrêter avec un message d'erreur mais demander de ressaisir un numéro de dé.

Consignes et évaluation

Le code source de votre projet est à envoyer par mail intitulé [NSI] Prénom 1 Prénom 2 - projet Yahtzee à l'adresse `senotprof@gmail.com` **avant le mardi 24/01/2023 à 23h59**. Le code source sera joint au mail dans un format de dossier compressé de type `prenom.zip`, ou d'archive compressée au format `.tar.gz` (aucun format propriétaire, type `.rar`, ne sera accepté), le dossier devant être nommé au format `prenom_1_prenom_2` (sans accent ni majuscule). Aucun délai supplémentaire ne sera accordé, la date butoir est immuable, et tout retard sera pénalisé : il convient donc de commencer le projet au plus tôt.

Le projet est à **coder obligatoirement en binôme** (sauf dérogation pour un unique trinôme, si l'effectif de classe est impair). La note sera commune au binôme, sauf cas de travail déséquilibré flagrant. Il n'est pas interdit (et c'est même conseillé) de discuter du projet entre les différents groupes et de s'échanger des idées ou de demander un peu d'aide (à commencer par celle l'enseignant), mais l'intégralité du code devra être produite par les membres du groupe. Une mini-séance de questions individuelles pourra avoir lieu après la remise des projets, et sera l'occasion de s'assurer que le programme a bien été conçu et implanté en Python par les membres du groupe. **Tout plagiat fera l'objet de sanctions avec une tolérance strictement négative.**

L'évaluation prendra en compte :

- les erreurs de syntaxe : les fichiers `.py` doivent être directement interprétés sans erreur par la console Python
- **l'originalité du code** (dans le sens de « code personnel et individuel »)
- l'implantation des fonctions imposées et le **respect des consignes** (règles du jeu, cahier des charges, consignes sur le code, les structures de données et les spécifications des fonctions, mais aussi les consignes annexes, comme l'intitulé du mail et le format de fichier, ainsi que bien évidemment le respect des délais)
- la correction et la performance du code
- la propreté et la lisibilité du code ainsi que tout ce qui facilitera sa compréhension par le correcteur (commentaires du code, docstrings, noms des variables, commentaires du code, noms des fonctions, commentaires du code, « aération » du code, commentaires du code, *etc*)
- la facilité d'utilisation et la jouabilité

Un exemple

Un exemple d'interface et de présentation du déroulement d'un tour de jeu est donné sur la page suivante. Cet exemple a pour vocation à donner des idées et servir d'inspiration, et non à être recopié à l'identique.

BIENVENU AU YAHTZEE

C'est au joueur 1 de jouer :

Dé n°1 Dé n°2 Dé n°3 Dé n°4 Dé n°5

```
-----
|  1  |  3  |  6  |  6  |  6  |
-----
```

Un dé à relancer (non pour finir) ? 1

Un autre dé à relancer ? `!!?3@nimp&#%`

Entrée non valide. Ressaisir.

Un autre dé à relancer ? 2

Un autre dé à relancer ? non

Dé n°1 Dé n°2 Dé n°3 Dé n°4 Dé n°5

```
-----
|  1  |  6  |  6  |  6  |  6  |
-----
```

Un dé à relancer (non pour finir) ? 1

Un autre dé à relancer ? non

Dé n°1 Dé n°2 Dé n°3 Dé n°4 Dé n°5

```
-----
|  4  |  6  |  6  |  6  |  6  |
-----
```

Votre grille :

Contrat n° 1 (les 1) :

Contrat n° 2 (les 2) :

Contrat n° 3 (les 3) :

Contrat n° 4 (les 4) :

Contrat n° 5 (les 5) :

Contrat n° 6 (les 6) :

Contrat n° 7 (brelan) :

Contrat n° 8 (carré) :

Contrat n° 9 (full) :

Contrat n° 10 (petite suite) :

Contrat n° 11 (grande suite) :

Contrat n° 12 (Yahtzee) :

Contrat n° 13 (chance) :

Quel contrat souhaitez-vous valider ? Le numero machin -426455 truc dodo&~~*! zzzzzzzzzzzzzzzzzzz

Entrée non valide. Ressaisir.

Quel contrat souhaitez-vous valider ? 8

Votre grille :

Contrat n° 1 (les 1) :

Contrat n° 2 (les 2) :

Contrat n° 3 (les 3) :

Contrat n° 4 (les 4) :

Contrat n° 5 (les 5) :

Contrat n° 6 (les 6) :

Contrat n° 7 (brelan) :

Contrat n° 8 (carré) : 28

Contrat n° 9 (full) :

Contrat n° 10 (petite suite) :

Contrat n° 11 (grande suite) :

Contrat n° 12 (Yahtzee) :

Contrat n° 13 (chance) :