

Rapport de programmation : Générateur de Quiz.

Membres de groupe :

1. Tagne Nzeuquem Igor Laurel
2. Djieunang Noumbo nelie mabelle
3. KOUROUMA Vassy
4. AFFI ATOBLA FRANCK ISAAC
5. DJIGAL Mame Khary Falilou
6. Diallo Alpha Ousmane
7. Ismaël Takoukam Mba
8. Tambat Tresor Megane
9. Minkoulou Ngoa Joseph Rick Jordy.
10. KUELANG KENGNI BODRIN

LISTE DES IMAGES ET TABLEAUX

N°	Titres	Pages
	Introduction	
	Objectif	
1	random	3
2	tqdm	4
3	os	4
4	Fonction printList	4
5	Fonction getData	4
6	Fonction printList	4
7	Fonction processData	4

8	Fonction main	5
9	Images	6

Introduction

Dans le cadre de notre unité d'enseignement qui est PROGRAMMATION I, il nous est proposé un projet de 1 mois nous permettant de mettre en pratique nos connaissances dans le langage PYTHON acquises tout au long de notre apprentissage afin d'évaluer nos compétences dans le développement procédural.

Objectif :

Dans ce code, nous allons créer un quiz simple en utilisant les fonctions, les structures conditionnelles et les collections pour créer un programme qui génère un quiz à partir d'un ensemble de questions stockées dans un fichier.

Ce code permet à l'utilisateur d'entrer une réponse parmi les différents choix de réponse, chaque question possède quatre choix de réponse, A B C et D. À la fin du quiz le code renvoie les nombres de réponse, trouvez et échouer. Dans les lignes qui suivent, nous expliqueront de manière détaillée, les différentes parties du code.

Préalable et Exigences du code :

```
#-----fichier Initiale-----
# avant toute exécution, tapez la commande suivante "pip install tqdm"
#tqdm est le module responsable de barre de progrès dans le terminal.
#Code simple de quiz qui utilise les fonctions définies dans le code précédent pour toutes modifications
#Pour toutes modifications des questions, veuillez respecter la syntaxe des fichiers et les règles ci-dessous
#- chaque question doit avoir 4 options numérotées de a à d
#- chaque question doit se terminer par un point d'interrogation(?) suivi d'un espace ' ' et de la lettre de la réponse
```

Importations de modules :

- **random** : Ce module est utilisé pour mélanger aléatoirement l'ordre des questions dans le quiz. Cela garantit que les questions sont posées dans un ordre différent à chaque exécution du programme, ce qui ajoute de la variété et de l'imprévisibilité au quiz.

2. `import random`

- **tqdm** : Cette bibliothèque permet de créer une barre de progression interactive pendant l'exécution du quiz. Il est utilisé dans le code pour afficher une barre de progression dans le terminal pendant l'itération à travers les questions pendant le quiz. Dans l'ensemble, le rôle de **tqdm** dans le code est d'améliorer l'expérience utilisateur en fournissant une représentation visuelle de l'avancement du quiz.

3. `from tqdm import tqdm`

- **os** : Cette importation est utilisée pour appeler la fonction **clear()** du module **os** afin d'effacer l'écran entre chaque question posée. Cela permet de garder l'interface utilisateur propre en supprimant les questions précédentes de l'écran avant d'afficher la suivante.

4. `import os`

5. Fonction **printList** :

- Cette fonction prend une liste en entrée et affiche chaque élément de la liste sur une nouvelle ligne en utilisant une boucle **for**. Cela garantit une présentation propre et lisible des éléments de la liste.

6. `def printList(list:list):`

7. `for elt in list:`

8. `print(elt)`

Fonction **getData** :

- Cette fonction lit les données à partir d'un fichier texte spécifié par le paramètre **filename**.
- Le fichier doit être structuré de telle manière que chaque question soit suivie de quatre options de réponse sur des lignes consécutives.
- La fonction parcourt le fichier ligne par ligne et organise les données dans une liste de listes, où chaque sous-liste contient la question et ses quatre options de réponse.

9. `def getData(filename="Question"): #insérer le chemin du fichier question`

10. `data = []`

11. `with open(filename, "r") as file:`

12. `temp = []`

13. `for line in file:`

14. `temp.append(line.strip())`

15. `if len(temp) == 5:`

16. `data.append(temp)`

17. `temp = []`

18. `return data`

Fonction processData :

- Cette fonction prend les données brutes obtenues à partir de la fonction **getData** et les organise dans un dictionnaire pour une manipulation plus facile.
- La clé du dictionnaire est la question, et la valeur associée est un autre dictionnaire contenant la réponse correcte et les options de réponse.

```
19. def processData(data):
20.     dic = {}
21.     for elt in data:
22.         question = elt[0].split('?')[0]+'?'
23.         options = [opt.strip() for opt in elt[1:5]]
24.         answer = elt[0].split('?')[1].strip()
25.         dic[question] = {"reponse": answer, "options": options}
26.     return dic
```

Fonction main :

- Cette fonction gère le déroulement principal du quiz.
- Elle commence par initialiser le score du joueur à zéro et mélange aléatoirement l'ordre des questions pour chaque nouvelle session de jeu.
- Ensuite, elle boucle à travers les questions, posant chacune à l'utilisateur.
- L'utilisateur saisit sa réponse, qui est ensuite vérifiée par rapport à la réponse correcte.
- Le score est mis à jour en conséquence.
- Une fois que toutes les questions ont été posées, le score final est affiché et l'utilisateur a la possibilité de choisir de continuer à jouer ou de quitter le quiz.

```
27. def main():
28.     m_score = 0
29.     fileData = getData()
30.     m_dic = processData(fileData)
31.     keys = list(m_dic.keys())
32.     random.shuffle(keys)
33.     taille = len(m_dic)
34.
35.     while True:
36.         for key in tqdm(keys, bar_format="{l_bar}{bar}{r_bar}", colour="green"):
37.             print()
38.             print(key)
39.             printList(m_dic[key]['options'])
40.             b = input("Entrez votre réponse (a, b, c ou d) : ")
41.             while b not in ['a', 'b', 'c', 'd']:
42.                 print("Veuillez entrer une réponse valide (a, b, c ou d)")
```

```

43.         b = input("Entrez votre réponse (a, b, c ou d) : ")
44.         if b == m_dic[key]["reponse"]:
45.             m_score += 1
46.
47.         os.system('cls')
48.
49.     print("Votre score est de : ", m_score, "/", taille)
50.     c = input("Voulez-vous continuer ? (O/N) ")
51.     if c.upper() == 'N':
52.         break
53.     else:
54.         random.shuffle(keys)
55.         m_score = 0
56.         os.system('cls')
57.
58. if __name__ == "__main__":
59.     main()

```

C'est ainsi que fonctionne ce script de quiz interactif en détail.

Image :

Situation 1 : Lorsque l'utilisateur est en cours du jeu

The screenshot shows a Python IDE terminal window with the 'TERMINAL' tab selected. The terminal displays a quiz question: '17% | Quel est le principal objectif de l'Arduino ?'. Below the question are four multiple-choice options: 'a- Il permet de souder du matériel informatique', 'b- Il permet de distraire des enfants', 'c- Il permet, à partir d'événements détectés par des capteurs, de programmer et commander des actionneurs ;', and 'd- Il permet de détruire des IA'. The prompt 'Entrez votre réponse (a, b, c ou d) : ' is shown at the bottom of the terminal, with a cursor waiting for input. The top of the terminal window shows tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The status bar at the bottom right indicates 'Python', a zoom level of '+', and a file icon. The bottom right corner of the terminal shows the progress '2/12 [00:46<04:20, 26.02s/it]'.

Situation 2 : Lorsque l'utilisateur termine le Quiz

The screenshot shows a Python IDE terminal window with the 'TERMINAL' tab selected. The terminal displays the end of a quiz. A green progress bar is shown at the top, followed by the text '100% | Votre score est de : 8 / 12'. Below this, the prompt 'Voulez-vous continuer ? (O/N) ' is shown, with a cursor waiting for input. The top of the terminal window shows tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The status bar at the bottom right indicates 'Python', a zoom level of '+', and a file icon. The bottom right corner of the terminal shows the progress '12/12 [04:53<00:00, 24.47s/it]'.

Situation 3 : Quand l'utilisateur entre une valeur incohérente

The screenshot shows a Python IDE terminal window with the 'TERMINAL' tab selected. The terminal displays a quiz question: '8% | Quel est le plus grand pays du monde par superficie ?'. Below the question are four multiple-choice options: 'a-La Russie', 'b-La Chine', 'c-Les États-Unis', and 'd-Le Canada'. The prompt 'Entrez votre réponse (a, b, c ou d) : ' is shown, followed by the user's input 'g'. Below this, the prompt 'Veuillez entrer une réponse valide (a, b, c ou d) ' is shown, followed by the user's input 'g'. Below this, the prompt 'Entrez votre réponse (a, b, c ou d) : ' is shown, with a cursor waiting for input. The top of the terminal window shows tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The status bar at the bottom right indicates 'Python', a zoom level of '+', and a file icon. The bottom right corner of the terminal shows the progress '1/12 [00:04<00:50, 4.63s/it]'.

En résumé, ce code fournit une application de quiz simple qui lit les questions à partir d'un fichier, les présente à l'utilisateur une par une et suit leur score. Il permet également une personnalisation facile des questions en modifiant le fichier d'entrée.