

Mini-Projet Python : Développement d'un Jeu de Quiz

1. Objectif général

Créer une application Python interactive permettant de jouer à un jeu de quiz, avec un menu principal, une gestion du score, un pourcentage de réussite, et l'enregistrement des résultats.

Ce projet permet de mettre en pratique :

- Les listes
- Les chaînes de caractères
- Les fichiers texte
- Les conditions et boucles

2. Description générale du projet

Le programme doit afficher un menu principal proposant trois options :

1. Jouer une partie
2. Voir les scores enregistrés
3. Quitter le programme

a) Structure du fichier des questions (questions.txt)

Les questions sont stockées dans un fichier texte avec le format :

- Ligne 1 : la question
- Ligne 2 : choix A
- Ligne 3 : choix B
- Ligne 4 : choix C
- Ligne 5 : la bonne réponse (A, B ou C)
- Ligne 6 : ligne vide

Le programme doit lire ce fichier, interpréter son contenu, et stocker les données dans une structure adaptée (ex. liste).

b) Fonctionnement du mode "Jouer"

Lorsque le joueur choisit Jouer, le programme :

1. Charge les questions depuis le fichier
2. Affiche chaque question avec les choix A, B, C
3. Demande une réponse (input)
4. Vérifie si la réponse est correcte
5. Comptabilise le score

L'utilisateur répond une seule fois par question (aucune gestion de tentatives multiples).

c) Système de score et pourcentage de réussite

À la fin du quiz, le programme doit afficher :

- Le nombre de réponses correctes
- Le nombre total de questions
- Le pourcentage de réussite, calculé comme :

$(\text{Score obtenu} / \text{nombre total de questions}) \times 100$

Exemple :

```
===== Résultat =====  
Nom du joueur : DEV  
Réponses correctes : 7  
Réponses incorrectes : 13  
Score total : 7 / 20  
Pourcentage de réussite : 35.00%  
=====
```

d) Enregistrement des scores (scores.txt)

Chaque partie jouée doit être enregistrée dans un fichier texte, sous la forme :

Nom du joueur – Correctes : X – Incorrectes : Y – Total : X+Y - Z%

Le programme doit ajouter une nouvelle ligne après chaque partie.

e) Fonction Voir les Scores

Cette option du menu doit :

- Lire tous les scores enregistrés dans scores.txt
- Afficher chaque score sur une ligne
- Indiquer un message si aucun score n'est encore enregistré

Exemple :

```
===== SCORES =====  
Abdelkader – Correctes : 7 – Incorrectes : 13 – Total : 20 - 35.00%  
Sara – Correctes : 17 – Incorrectes : 3 – Total : 20 - 75.00%
```

f) Menu principal obligatoire

Le menu doit apparaître en boucle tant que l'utilisateur n'a pas choisi Quitter.

Exemple :

```
===== MENU PRINCIPAL =====  
1. Jouer  
2. Voir les scores  
3. Quitter  
Votre choix :
```

Le programme doit valider les choix de l'utilisateur.

3. Contraintes techniques obligatoires

Le projet doit utiliser :

❖ Les listes

- Pour stocker les questions et réponses.

❖ Les chaînes de caractères

- Pour manipuler les textes.

❖ Les fichiers texte

- questions.txt pour les questions
- scores.txt pour enregistrer les résultats

❖ Les structures de contrôle

- Boucles
- Conditions
- Input utilisateur

4. Livrables attendus

1. Le fichier quiz.py
2. Le fichier questions.txt
3. Le fichier scores.txt (généré automatiquement)
4. Une courte documentation expliquant :
 - Comment fonctionne le programme
 - Structure des fichiers
 - Améliorations possibles