

Rapport:

Introduction:

Ce projet consiste en la création d'une application de quiz en Java. L'objectif est de proposer à l'utilisateur une série de questions à choix multiples, accompagnées d'un chronomètre de 10 secondes pour chaque réponse. À la fin du quiz, les réponses correctes sont comptabilisées et un résumé des résultats est affiché.

L'application offre une navigation fluide entre les questions et affiche le score final dans une boîte de dialogue. Elle intègre une logique claire pour la gestion des questions, des réponses et du chronomètre.

Logique du Quiz :

- **Questions et options** : Les questions et leurs options sont gérées sous forme de tableau. Chaque question est associée à quatre options possibles.

- **Réponses correctes** : La réponse donnée par l'utilisateur est comparée à un tableau contenant les réponses correctes pour déterminer si elle est valide.

- **Comptabilisation des réponses correctes** : Chaque réponse correcte augmente le score. Le score final est affiché à la fin du quiz.

- **Chronomètre** : Un décompte automatique de 10 secondes par question est activé. Si le temps est écoulé avant qu'une réponse ne soit donnée, la question est automatiquement sautée.

- **Score** : Correspond au nombre de réponses correctes données par l'utilisateur au cours d'une session.

- **High Leader Score** : Représente le meilleur score jamais atteint par un utilisateur depuis le lancement de l'application.

Redémarrage du quiz (restartQuiz) :

À la fin du quiz, l'utilisateur est invité à rejouer.

- Si "Oui" : Toutes les variables (score, numéro de question, etc.) sont réinitialisées, et le quiz redémarre à la première question.

- Si "Non" : L'application se termine.

Les membres de groupes avec la tâches de chacune :

- **Merzouk Siham** : Gestion du timer et des résultats.

- **Merzouk Fatima** : Gestion des questions et des réponses.

- **Kerrouche Dina** : Interface graphique, gestion des boutons et du layout.

Les autres tâches ont été réalisées ensemble.

Fonctionnalités:

1.Interface graphique (GUI) : L'application utilise Swing pour afficher les questions, les options, le chronomètre et les boutons, offrant une interface intuitive et conviviale.

2.Questions à choix multiples: Les questions et réponses sont stockées dans des tableaux, et la méthode `setQuestion()` met à jour l'interface.

3.Chronomètre : Un `Timer` Swing décrémente automatiquement le temps restant pour répondre. Si le chronomètre atteint zéro, la question est automatiquement considérée comme non répondue.

4.Vérification des réponses: La méthode `checkAnswer()` compare la réponse sélectionnée par l'utilisateur avec la réponse correcte et met à jour le score.

5.Navigation: Le bouton "Next" permet de passer à la question suivante, et le bouton "Show Result" affiche un résumé final du score.

6. Affichage des résultats : La méthode `displayResults()` montre le score final ainsi qu'un récapitulatif des réponses correctes et incorrectes dans une boîte de dialogue.

7. Gestion des événements utilisateur : L'interface `ActionListener` gère les clics sur les boutons pour permettre l'interaction avec le quiz.

8. Conception modulaire : Les différentes fonctionnalités sont organisées dans des méthodes distinctes, facilitant la maintenance et les extensions éventuelles.

Défis rencontrés:

- Mauvaise organisation des indices des tableaux : Au début, un manque de maîtrise dans la manipulation des tableaux a conduit à des erreurs dans l'organisation des indices.

- Apprentissage du chronomètre : Nous avons étudié la création d'un timer de A à Z pour l'intégrer correctement dans l'application.

- Problème de mise à jour du chronomètre : Initialement, le chronomètre ne se réinitialisait pas à chaque nouvelle question, ce qui a nécessité des ajustements dans le code.

- Structure de la fenêtre : La fenêtre graphique était mal organisée au départ, mais a été améliorée après plusieurs itérations et ajustements.

Nous avons surmonté ces difficultés en recherchant des solutions sur différentes chaînes YouTube, ainsi qu'avec l'aide de ChatGPT et Perplexity.

Conclusion:

Ce projet a permis d'explorer et de consolider plusieurs concepts essentiels, notamment :

- Gestion des données : Organisation efficace des questions, options et réponses dans des tableaux.

- Manipulation des tableaux : Simplification des opérations liées à la gestion des données en Java.

- Création et gestion d'un timer : Mise en place d'un chronomètre fonctionnel et sa réinitialisation à chaque nouvelle question.

Cette expérience a également mobilisé une grande partie des notions étudiées lors des TP. Elle nous a permis de développer des compétences précieuses qui seront particulièrement utiles pour notre projet de fin d'études.