Projet C# Quizz

# Description de l’appli

Il s’agit de développer un quizz en application console.

## Principe de base

Un quizz est composé de questions à choix multiples. Il peut être décrit par un fichier texte comme celui-ci :



Les réponses correctes sont précédées d’un astérisque (\*).

Chaque question rapporte 1 point si toutes les réponses correctes ont été données et uniquement celles-là.  
On ne retire pas de points en cas de réponse fausse.

## Déroulement du quizz

Au début du quizz, l’application demande au joueur de saisir son nom et son prénom.

Puis pour chaque question, elle :

* Affiche la question
* Affiche les propositions. Chacune d’elles est identifiée par une lettre
* Invite l’utilisateur à saisir la ou les lettres correspondant à ses réponses en majuscules et sans espaces (ex : AC). Si le format de la réponse est incorrect, le joueur est invité à la saisir de nouveau.
* Vide l’écran

A la fin du quizz, l’application enregistre les résultats du joueur dans un fichier (cf. paragraphe suivant), puis affiche ce qui suit :

* Le nombre de bonnes réponses / nombre de questions (ex : Votre résultat : 7/10)
* Les questions sur lesquelles le joueur a fait des erreurs (réafficher la question complète avec les différentes propositions), et les réponses correctes en-dessous

Après avoir vidé l’écran, l’application demande à l’utilisateur s’il veut voir les statistiques du jeu.

Puis elle affiche un message invitant l’utilisateur à appuyer sur la touche Q pour quitter l’application.

## Fichier de résultats

Il s’agit d’un fichier texte contenant les résultats de tous les joueurs, avec la structure suivante :

Date Nom Score Erreurs  
2019-12-23 LECLERC Gaëlle 8/10 3,8  
2019-12-23 FERRAND Aurélie 4/10 1,3,5,7,8,10  
2019-12-24 FORTIER Sylvain 6/10 1,3,7,9  
…

Les colonnes sont séparées par des tabulations. Elles n’ont pas obligatoirement une largeur fixe.  
La colonne erreurs contient les numéros des questions auxquelles le joueur a mal répondu.

Le fichier est complété à chaque fin de quiz

## Statistiques

Les statistiques à afficher à la fin sont les suivantes :

* Nombre total de jeux
* Score moyen
* Pourcentage de bonnes réponses sur chaque question (identifiée par son numéro), en classant les questions par pourcentage croissant.

# Spécifications techniques

Vous devez fournir un diagramme de classe décrivant la structure de l’application (ce diagramme vous aidera pour la phase d’analyse).

Les fichiers de quizz et de résultats doivent être placés dans le dossier du projet. Leurs chemins doivent être définis en relatif par rapport à l’exécutable, de façon que l’application fonctionne quel que soit le dossier dans lequel elle se trouve.  
NB/ Pour faciliter le test et le débogage de votre code, vous pouvez travailler au début sur un fichier réduit à 3 ou 4 questions.

Pour faciliter la relecture de votre code, vous devez décrire chaque méthode avec un commentaire /// (description de la méthode, de ses paramètres et de son retour). Vous devez également décrire succinctement en commentaire les choix techniques les plus importants (architecture de l’appli, pourquoi vous utilisez telle ou telle technique…)

Si vous n’avez pas le temps de tout faire, mettez en commentaire des TODO sur ce que vous aviez éventuellement prévu de faire.

# Livraison

Vous devez fournir sur GitHub :

* Le fichier du diagramme de classes
* La solution Visual Studio contenant le code de l’application
* Les fichiers textes utilisés par l’application

L’application doit pouvoir être lancée depuis Visual Studio sans modification ou paramétrage préalable.

# Critères d’évaluation

**Fonctionnel et ergonomie**

* Niveau de fiabilité de l’application
* Réponse au besoin fonctionnel et nombre de fonctionnalités implémentées
* Soin apporté à l'interface visuelle

**Architecture**

* Pertinence du diagramme UML d’activité et de l’architecture réellement mise en oeuvre
* Séparation de l’affichage et de la logique applicative, respect du principe d'encapsulation

**Maîtrise technique**

* Utilisation des techniques à bon escient
* Concision du code
* Attention portée aux performances du code

**Qualité du code**

* Qualité des commentaires et lisibilité du code
* Qualité de la sémantique (nommage des classes, méthodes, propriétés, variables…)
* Respect des conventions de codage