

# Projet : Swift Quiz Game (Jeu de quiz Swift)

---

Date du rendu : 2 avril 2024 (Avant Minuit)

## Description du projet :

---

Développer une application de jeu de quiz sur console en utilisant le langage de programmation Swift. Ce jeu testera les joueurs sur divers sujets (par exemple, culture générale, science, histoire, programmation) au moyen de questions à choix multiples.

## Fonctionnalités et exigences de base :

---

### 1. Modèle de données de quiz

- Concevoir un modèle de données simple pour représenter les questions du quiz, les réponses possibles, la bonne réponse, le niveau de difficulté et la catégorie des questions.
- Les questions doivent être définies dans un fichier JSON et lu dans le programme à l'aide du framework Foundation.
- Chaque question doit être représentée sous forme de **class** ou de **struct** une fois chargée dans le programme Swift.
- Concevoir un modèle de données pour suivre les classements des scores des utilisateurs au quiz.
- À la fin d'une session de jeu, le classement mis à jour des scores du quiz doit être écrit dans un fichier json.
- Utiliser ce fichier json de classement des scores de quiz pour suivre les classements des scores des utilisateurs entre les sessions d'application.

### 2. Mécaniques de jeu

- Au début du jeu, l'utilisateur doit saisir son nom et sélectionner un niveau de difficulté. Le niveau de difficulté indique les questions les plus difficiles pouvant être posées.
- Mettre en place un système permettant de présenter les questions à l'utilisateur une à la fois dans un ordre aléatoire, en respectant le niveau de difficulté.
- Permettre à l'utilisateur de sélectionner une réponse et de fournir un retour immédiat indiquant si elle était correcte ou incorrecte.
- Garder une trace du score de l'utilisateur tout au long de la session de jeu.

- Une fois la dernière question répondue, l'utilisateur verra son score et son classement par rapport aux autres utilisateurs, les classements doivent être classés par niveau de difficulté.

### 3. Interface utilisateur

- Concevoir une interface de ligne de commande claire et concise pour le jeu. Inclure des instructions, l'affichage des questions actuelles, des choix de réponses et un moyen de sélectionner les réponses.
- Mettre en œuvre la validation des entrées pour garantir que l'utilisateur sélectionne un choix de réponse valide.

### 4. Système de scores et de remontée d'informations

- Développer un système de notation basé sur l'exactitude des réponses et la difficulté des questions.
- Offrir des commentaires pour chaque question une fois que l'utilisateur a répondu.
- Si l'utilisateur saisit une entrée invalide, fournir des instructions claires sur la manière de saisir une entrée valide.

### 5. Règles

- Les questions ne doivent pas apparaître plus d'une fois par session de jeu.
- Vous êtes libre de choisir n'importe quel sujet ou catégorie pour vos questions.
- Le projet doit être développé en utilisant Git.
- Le code doit avoir des commentaires. Il n'est pas nécessaire de commenter chaque ligne mais les fonctions, composants / éléments clés et tests conditionnels doivent avoir des commentaires.

## Fonctionnalités avancées :

---

2 points supplémentaires seront attribués pour la mise en œuvre d'une fonctionnalité avancée.  
5 points supplémentaires seront attribués si vous mettez en œuvre les deux.

- **Réponses chronométrées** Accorder à l'utilisateur un point bonus si la réponse correcte est saisie en moins de 5 secondes. Définir un délai d'attente de 30 secondes par question. Si un utilisateur prend plus de 30 secondes pour répondre, la question sera marquée comme incorrecte. Assurez-vous de fournir les commentaires des utilisateurs appropriés.
  - **Éditeur de banque de questions** : Créer une interface simple pour ajouter, modifier ou supprimer des questions de la banque de questions.
-

# Livrables du projet

---

- Code source de l'application Swift Quiz Game (jeu de quiz Swift) avec soit un lien Github, soit un projet avec un répertoire .git.
- Un fichier README.md qui explique comment configurer, démarrer le jeu et comment jouer.
- Un bref rapport (~ 2 pages) détaillant les choix de conception, les choix de types de données, les défis rencontrés et la manière dont ils ont été résolus.