

Application de jeu de quiz en Swift

- Introduction

Ce rapport détaille le processus de développement de l'application de jeu de quiz en Swift. Nous aborderons les choix de conception, les types de données utilisés, les défis rencontrés et les solutions apportées tout au long du développement de ce projet.

- ❖ Choix de conception

- Modèle de données

Pour représenter les questions du quiz, les réponses possibles, la bonne réponse, le niveau de difficulté et la catégorie des questions, nous avons opté pour un modèle de données simple utilisant des structures en Swift. Chaque question est représentée par une structure `Question` contenant des propriétés telles que `question`, `options`, `correctAnswer`, `infos`, `difficulty` et `category`.

- Mécaniques de jeu

Nous avons mis en place un système permettant de présenter les questions à l'utilisateur une à une dans un ordre aléatoire tout en respectant le niveau de difficulté sélectionné. L'utilisateur peut sélectionner une réponse et recevoir un retour immédiat sur sa réponse. Le score de l'utilisateur est stocké tout au long de la session de jeu.

- Interface utilisateur

Nous avons conçu une interface de ligne de commande claire et concise pour le jeu. Cela inclut des instructions, l'affichage des questions actuelles, des choix de réponses et un moyen de sélectionner les réponses. Nous avons également implémenté une validation des entrées pour garantir que l'utilisateur sélectionne un choix de réponse valide.

- Système de scores et de remontée d'informations

Nous avons développé un système de notation basé sur l'exactitude des réponses et la difficulté des questions. Des commentaires sont fournis pour chaque question une fois que l'utilisateur a répondu. En cas d'entrée invalide, des instructions claires sont fournies à l'utilisateur sur la manière de saisir une entrée valide.

❖ Choix de types de données

▪ Utilisation de structures

Nous avons choisi d'utiliser des structures en Swift pour représenter les questions du quiz et les données associées. Les structures offrent une manière claire et concise de définir les propriétés d'une question et de les regrouper de manière logique.

▪ Utilisation de fichiers JSON

Pour stocker les questions du quiz et les scores des utilisateurs, nous avons décidé d'utiliser des fichiers JSON. Cela permet une gestion facile des données et une lecture simple dans le programme à l'aide du framework Foundation.

❖ Défis rencontrés et solutions

▪ Gestion des scores des utilisateurs

Le défi était de suivre les scores des utilisateurs tout au long de la session de jeu et de les enregistrer pour une utilisation future. Nous avons résolu ce problème en concevant un modèle de données pour suivre les classements des scores des utilisateurs et en écrivant les classements mis à jour dans un fichier JSON à la fin de chaque session de jeu.

▪ Traitement des scores

Le problème rencontré était le fait que dans le classement final, le rang était seulement défini par le nombre de score. Ainsi le code ne fait pas de différence entre les niveaux.

❖ Conclusion

Le développement de l'application de jeu de quiz en Swift a été une expérience enrichissante. En utilisant des structures en Swift, des fichiers JSON et les fonctionnalités du framework Foundation, nous avons réussi à concevoir une application de jeu de quiz fonctionnelle avec une interface utilisateur claire et des mécaniques de jeu engageantes. Les défis rencontrés tout au long du processus de développement ont été résolus avec succès, ce qui a abouti à un produit final robuste et satisfaisant.