

# Rapport

Réalisé par : Celina CHABANE



**L3 ISEI**

**Université Vincennes Saint-Denis Paris 8**



**2023/2024**

## Introduction:

L'objectif du projet "Swift Quiz Game" est de développer un jeu-questionnaire interactif en se servant du langage de programmation Swift. Le jeu offre aux joueurs la possibilité de répondre à des questions à divers niveaux de complexité et de consulter leur score final et leur position parmi les autres participants. Le jeu est élaboré de manière à être facile à comprendre et amusant, donnant aux utilisateurs une expérience plaisante.

### 1. Détaillant les choix de conception:

Dans cette réalisation, nous avons fait usage de deux structures primordiales : **Question** et **Player**. La question est représentée par la structure Question dans le jeu, tandis que la structure Player enregistre les joueurs avec leur nom, niveau de difficulté et score.

Le jeu montre des instructions à l'utilisateur pour qu'il saisisse son nom et sélectionne le niveau de difficulté lors de l'interaction. Ensuite, les questions sont affichées avec des choix de réponse, ce qui permet à l'utilisateur de choisir sa réponse en utilisant le clavier.

Les informations des joueurs sont stockées dans un fichier JSON appelé "player.json". Cela garantit la continuité de l'expérience de jeu en sauvegardant les scores des joueurs entre les sessions.

Finalement, à la clôture de chaque partie, le classement des joueurs est montré. Les performances des joueurs sont comparées les unes aux autres en classant les joueurs par score dans l'ordre décroissant pour chaque niveau de difficulté.

### 2. Les choix de types de données:

Nous avons principalement opté pour l'utilisation de structures (struct) pour représenter les éléments clés du jeu : les questions et les joueurs.

**Question:** La classe Question sert à stocker chaque question du jeu. Elle inclut des caractéristiques comme le contenu de la question, les options de réponses, la réponse correcte, le niveau de complexité et la catégorie de la question. Les informations sont stockées de manière organisée, ce qui rend plus facile la manipulation et l'accès aux données de chaque question.

```
// structure pour les question
struct Question: Codable {
    let question: String
    let propositions: [String]
    let reponse: Int
    let niveau: Int
    let categorie: String
}
```

**Player :** La classe Player sert à stocker les données des joueurs telles que leur nom, leur niveau de difficulté sélectionné et leur score. Une fois de plus, la structuration des données permet de les stocker de manière ordonnée et de les manipuler aisément.

```
// structure du gamer
struct Player: Codable{
    var name: String
    var difficulty : Int
    var score : Int
}
```

En employant des structures pour symboliser les questions et les joueurs, nous profitons des éléments favorables de la programmation orientée objet, comprenant l'encapsulation des informations et des caractéristiques, ce qui simplifie le code, le rend plus fragmenté et plus accessible à entretenir. En outre, en employant les types de données adéquats tels que des tableaux pour enregistrer les joueurs et les questions, nous pouvons aisément naviguer et modifier ces données en fonction des exigences du jeu.

## **Conclusion:**

L'application Swift Quiz Game est un projet interactif qui fournit une expérience de jeu divertissante et cognitive à ses utilisateurs. En raison de décisions de conception judicieuses et d'une gestion efficace des difficultés rencontrées, le jeu parvient à offrir une expérience utilisateur sans accroc et plaisante. En améliorant constamment et en optimisant le jeu, il sera possible de proposer une expérience de jeu améliorée à l'avenir.