

# **SOFTWARE ENGINEERING**

## **TASK 3 : EVOLUTION - GROUP H**

### **Project description :**

EduTriviam is a software system that aims to enhance students' learning experience through gamified quizzes. Quizzes that are generated are based on predefined curriculum standards and questions are dynamically selected from a comprehensive database. This comprehensive database can be tailored to match the difficulty levels and objectives. Some of the key features of this system are the multiple subjects, consisting of questions from different difficulty levels, which are present in the comprehensive question database, assistant features, such as '50/50' lifeline, 'eliminate one answer' lifeline, tracking the number of lives available to the player during gameplay, along with newly implemented features, which is described in this document.

### **Features implemented in 1st implementation :**

1. Question Database: A robust database (`QuestionDatabase` class) containing general questions of various difficulty levels was implemented.
2. Quiz Controller: The `QuizController` class to manage the quiz game's flow, interact with the user, and calculate scores based on responses was developed.
3. Quiz View: Created the `QuizView` class to handle user interaction via the command-line interface, displaying questions, options, and feedback.
4. Question Model: The `Question` class to represent individual quiz questions, encapsulating text, options, and correct answers was created.
5. 50-50 Lifeline Feature: A lifeline feature allowing users to remove two incorrect options during the quiz, improving their chance of selecting the correct answer was implemented.
6. Remove One Wrong Option Lifeline: A lifeline feature enabling users to remove one incorrect option, enhancing their likelihood of choosing the correct answer was added.

7. **Number of Lives:** Introduced a "Number of Lives" feature, deducting lives for incorrect answers and ending the quiz when lives reach zero, providing a strategic element to the gameplay.

### **Task 1 : Improvement of at least one feature implemented in 1st implementation**

1. **Creation of a comprehensive database :** The database that was handed over had general questions (a total of 18) of varying difficulties. In order to improve this feature, these topics were categorized based on subjects - General Knowledge, Geography, Literature, Maths and Science. In addition to the existing questions, new questions were also added to the question database. Apart from this, difficulty levels for each selected subject was also set as Easy, Medium and Hard. There are currently 8 questions in each difficulty level of a subject, which is a total of 120 questions in the question database.
2. **Review of incorrectly answered questions :** This was a feature that was part of the requirements that we were able to implement. At the end of the quiz, if the user had incorrectly answered any of the questions, the incorrectly answered question along with the correct answer is displayed.
3. **Reattempt quiz :** A reattempt quiz was implemented, which was also part of the key features to be included in the requirements. Once the user has reviewed the incorrectly answered questions, the user will have the choice of whether to take a reattempt quiz or not. If yes, the user will be quizzed on the incorrectly answered questions. If not, the game ends.
- 4.

### **Task 2 : Implementation of new features**

1. **Implementation of 'Hint' lifeline :** A new assistance feature, called the Hint lifeline was implemented. The hint feature provides the user a clue, for questions in the difficulty level 'Hard'. The `QuestionDatabase` class was updated with hints for each 'Hard' question in each subject.
2. **Implementation of 'Back to Main Menu' Option :** The user can choose to leave the game mid-way when selecting the 'Back to Main Menu' option. The user then, can choose another subject and/or difficulty level to learn.

## Maintenance

1. **Repetition of option letters fixed** : When a user selects the 50-50 lifeline, two options are eliminated and the remaining two options had a repetition of option letters. For instance:  
a. a. Berlin  
instead of a. Berlin. This error was rectified.
2. **Error message for foreign input values as answers to questions** : When a user enters an invalid option for a question, it was recognised as an incorrect answer and the user loses a point. This problem was rectified such that if the user enters a value apart from the options given, an error message stating that an invalid value was entered is printed and asks the user to enter a valid answer.