



YZV 102E/104E
Introduction to Programming for Data Science
Term Project

Tevhidenur Serdar
150210335

June 4, 2023
Istanbul Technical University

Summary

The term project is a Python-based word game that challenges players to answer questions within a time limit. The game incorporates functionalities such as reading questions from a file, generating randomized letters as options, and calculating scores based on the players' answers. The game begins by reading questions and answers from a text file, allowing for easy customization and expansion of the question bank. Randomized letter options are generated for each question to provide assistance to players. During gameplay, players can either request a letter or submit their answer. The scoring system rewards correct answers and deducts points for requesting letters. Player information, including scores and remaining time, is stored in a database, and a leaderboard feature allows players to view top scores.

Overall, the provided report further discusses the game's methodology, including the functions and modules involved in the gameplay. It serves as a comprehensive introduction to the implemented word game. It presents the game's features, implementation details and potential for enhancement.

Word Game Project

Tevhidenur Serdar

Department of Artificial Intelligence and Data Engineering

Istanbul Technical University

Istanbul, Turkey

Serdar21@itu.edu.tr

Abstract—This term project implements a game program in Python that challenges users to answer questions within a time limit. The program includes functionalities for reading questions from a file, generating random letters, and calculating scores based on the users' answers. Player information, including scores and remaining time, is stored in a database. The program also provides options to view a leaderboard and offers the choice to play again or exit the game. Through this project, students can explore concepts such as file handling, randomization, user input validation, and time management. The code demonstrates the application of these concepts in a fun and interactive manner. Overall, this project provides an engaging experience for users while showcasing essential programming skills.

Index Terms—Word game, Python, file handling, randomization, user input validation, time management.

CONTENTS

I	Introduction	2
I-A	Project Overview.....	2
II	Methods	2
II-A	Reading Question Data	2
II-B	Randomized Letter Generation	3
II-C	Gameplay and Scoring.....	3
II-D	Database Storage and Leaderboard.....	3
II-E	Gameplay Options.....	3
III	Discussion	3
IV	Conclusion	3
V	Recommendations	3

I. INTRODUCTION

The aim of this term project is to develop a Python program that offers an interactive and challenging game experience for users. The program revolves around a question-and-answer format, where players are presented with a series of questions and are required to provide answers within a specified time limit. The code incorporates various functionalities to ensure an engaging and dynamic gameplay.

A. Project Overview

The main features of the program include:

- 1) **Question Data:** The program reads questions and corresponding answers from a text file, allowing for easy customization and expansion of the question bank.

- 2) **Randomized Letter Generation:** To enhance the gameplay, the program generates a set of randomized letters for each question. These letters are presented to the player as options to aid in answering the question.
- 3) **Score Calculation:** The program dynamically calculates scores based on the correctness of the answers and the time taken to provide them. The scoring system rewards players for accurate and timely responses.
- 4) **Database Storage:** Player information, including usernames, scores, and remaining time, is stored in a database file. This enables the program to maintain a record of players' performance and allows for the generation of leaderboards.
- 5) **Leaderboard Display:** The program offers the functionality to view a leaderboard showcasing the top players' scores and remaining time. This adds a competitive element and encourages players to improve their performance.
- 6) **Play Again Option:** Upon completing a game, players are given the choice to play again or exit the program. This allows for multiple gameplay sessions and provides a seamless and enjoyable user experience.

By implementing this project, students can explore various concepts in Python programming, such as file handling, randomization, user input validation, score calculation, and database management. Additionally, the project promotes critical thinking, time management, and decision-making skills as players aim to answer questions accurately and quickly.

II. METHODS

The implementation of the game involves several key functions and modules, which collectively provide the desired interactive and challenging gameplay experience. The following sections outline the methodology and describe each component's role in the program.

A. Reading Question Data

The `read_questions()` function is responsible for reading question and answer data from a text file. The function takes the filename as a parameter and opens the file in read mode. It iterates through the file, randomly selecting questions and their corresponding answers from different categories. The selected questions are stored in a dictionary for later use. This approach ensures that a diverse set of questions is presented to the players, enhancing the gameplay experience.

B. Randomized Letter Generation

To add an additional layer of challenge and assistance to the players, the `random_letter()` function generates randomized sets of letters for each question. The function takes the question dictionary as input and creates a 2D list to store the letter options for each question. It selects random indices representing letters in the answer and constructs a set of boxes with the chosen letters filled in. The resulting 2D list provides the players with letter options that can aid them in solving the questions.

C. Gameplay and Scoring

The core functionality of the game is implemented in the `main()` function. This function takes the question dictionary, letter options list, and remaining time as input parameters. It iterates through the questions, presenting each question to the player one by one. The player is provided with empty boxes representing the answer length and the corresponding question. During the gameplay, the function allows the player to either request a letter or submit their answer. If the player requests a letter, the function displays a randomized letter option from the generated list and updates the score accordingly. The player can continue requesting letters until they have all the letters or choose to submit their answer. If the player submits an answer, it is compared against the correct answer, and the score is updated accordingly. The function also tracks the time taken for each question and deducts it from the remaining time.

D. Database Storage and Leaderboard

The `user_and_skor()` function is responsible for storing the player's username, score, and remaining time in a dictionary. This information is later used to update a database file using the `database()` function. The database file maintains a record of all players' performance, allowing for the generation of a leaderboard.

The `top_10()` function reads the database file, sorts the player data based on scores and remaining time, and selects the top 10 players. It then creates a new CSV file named `top_users.csv` and writes the top players' information into it. This file serves as the leaderboard, showcasing the top performers in the game.

E. Gameplay Options

The `play_again()` function provides options to the player after completing a game session. It allows the player to choose whether to play again, exit the game, or view the leaderboard. Depending on the chosen option, the function either returns a Boolean value to continue or end the game loop or displays the leaderboard by reading the `top_users.csv` file.

By implementing the aforementioned methodology, the program achieves its goal of providing an interactive and challenging game-like experience. It combines various functionalities such as reading question data, generating randomized letter options, calculating scores, storing player information, and displaying leaderboards. The modular design and careful implementation ensure an enjoyable and engaging gameplay experience for the users.

III. DISCUSSION

The implemented game successfully provides an interactive and challenging experience for the players. By randomly selecting questions from different categories and offering randomized letter options, the game ensures variety and unpredictability, keeping the players engaged. The scoring system rewards players for correct answers and deducts points for requesting letters, adding an element of strategy to the gameplay.

The time limit adds a sense of urgency, increasing the difficulty and excitement. Players must manage their time effectively to answer as many questions as possible within the given duration. The game's modular design allows for easy modification and expansion of the question bank, ensuring a fresh experience with each playthrough.

One potential area for improvement is the user interface. Enhancing the visual presentation with graphical elements, such as colored boxes for letters and a more appealing layout, could further enhance the gameplay experience. Additionally, implementing sound effects or background music could add to the game's immersion and enjoyment.

IV. CONCLUSION

In conclusion, the developed game provides an entertaining and challenging experience for players. By combining randomized questions, letter options, and a time limit, the game tests players' knowledge, strategy, and time management skills. The modular design allows for easy scalability and customization, enabling the addition of more questions and categories.

The implementation of a database and leaderboard feature further enhances the game's replay value, fostering competition among players. The ability to store player information and display top scores and remaining times adds a competitive element and motivates players to improve their performance.

Overall, the game achieves its intended goal of engaging players in an educational and entertaining manner, promoting learning and problem-solving skills.

V. RECOMMENDATIONS

Based on the implementation and user feedback, the following recommendations are suggested for further enhancing the game:

- Improve the user interface by incorporating visually appealing elements, such as colored boxes for letters and a well-designed layout.
- Consider implementing sound effects or background music to create a more immersive and enjoyable gaming experience.
- Expand the question bank by adding more diverse categories and a larger pool of questions, ensuring a wider range of challenges for the players.
- Introduce difficulty levels or customizable settings to cater to different skill levels and preferences.
- Explore the possibility of multiplayer functionality, allowing players to compete against each other in real-time.

By implementing these recommendations, the game can be further enhanced to provide an even more engaging and enjoyable experience for players of all ages and skill levels.