Criterion A

Scenario

One of my friends organizes a learning and brain-boosting club in order to improve critical thinking and problem-solving skills for children. The club uses puzzles and logic-based games to make learning more fun and engaging. One of the most popular games among the children is "Sedecordle," a word game where players must guess 16 five-letter words simultaneously within 20 attempts. While this activity is beneficial for cognitive development, my friend has identified a recurring issue: manually solving these puzzles is time-consuming, and verifying multiple students' answers quickly is difficult. (Appendix A, Entry 1)

To address this problem, my friend requires a Sedecordle Solver, a program that can efficiently determine all 16 words from a given game state. This will allow my friend to verify solutions instantly, generate answers for discussion, and analyse strategies to help students improve their performance. I realised that this was the perfect opportunity for my IA, so I proposed developing the Sedecordle Solver, which my friend accepted, and we scheduled a meeting to discuss the project requirements.

Rationale for solution

The Sedecordle Solver will consist of a base game available to all users and an additional solver function only accessible to teachers. Users will be able to register and log in to play the game, and teachers can access the solver feature by entering a dev key during registration. The solver will solve and display the correct 16 words based on the entered game seed, allowing teachers to validate solutions quickly. Additionally, a timer and unique game seed will allow students to compete fairly, solving the same set of words within a trackable time frame.

The project will be developed using Python with a Tkinter-based graphical user interface (GUI). Python and Tkinter are chosen due to their extensive libraries and ease of handling file management and cross platform capabilities across Windows, Linux, and Mac OS. Each game session will generate a unique seed linked to a specific sequence of words, enabling teachers to retrieve a student's game state by entering the corresponding seed into the solver. This eliminates the need for manual checking and ensures consistency in game validation.

For storing the accepted words library (5_letter_words.json), user credentials (users.json), game seeds and their corresponding sequence of words (sedecordle_seeds.json), JSON will be used as the database format. JSON is an excellent choice because it is lightweight, and human readable. Additionally, JSON allows for easy retrieval and manipulation of stored data, ensuring that the game and solver can be loaded and run efficiently. This format is particularly beneficial for a project of this scale, as it reduces overhead while maintaining flexibility and performance.

Unlike existing word games like Wordle, my project includes a built-in solver and time tracking, making it useful for fun and education. Unlike generic word solvers, mine is integrated within the game, allowing for seed-based word generation and instant answer validation, which existing games lack.

Success Criteria

1. Login for users, with an additional dev key for teacher access to the solver.
2. The GUI should be simple and user friendly.
3. Unique seed for every game, for fair competing with the same 16 words.
4. Timer to track progress and competing.
5. The solver must be able to generate all 16 words within a few seconds.
6. The solver should be able to solve all 16 words in 16-17 attempts.
7. The game and solver should function consistently without crashing or producing errors.

Words: 477