Product Design Specification (Individual Project)

Project Background

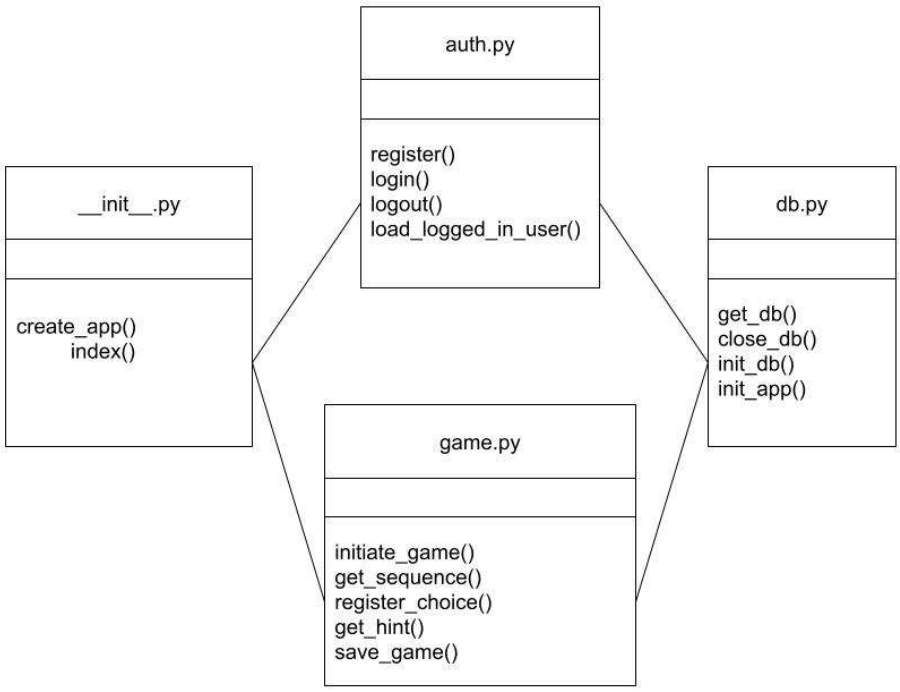
- Project title: Solitaire Scrabble
- Name: Bence Danko
- GitHub URL: https://github.com/bencejdanko/solitaire-scrabble

Project Description

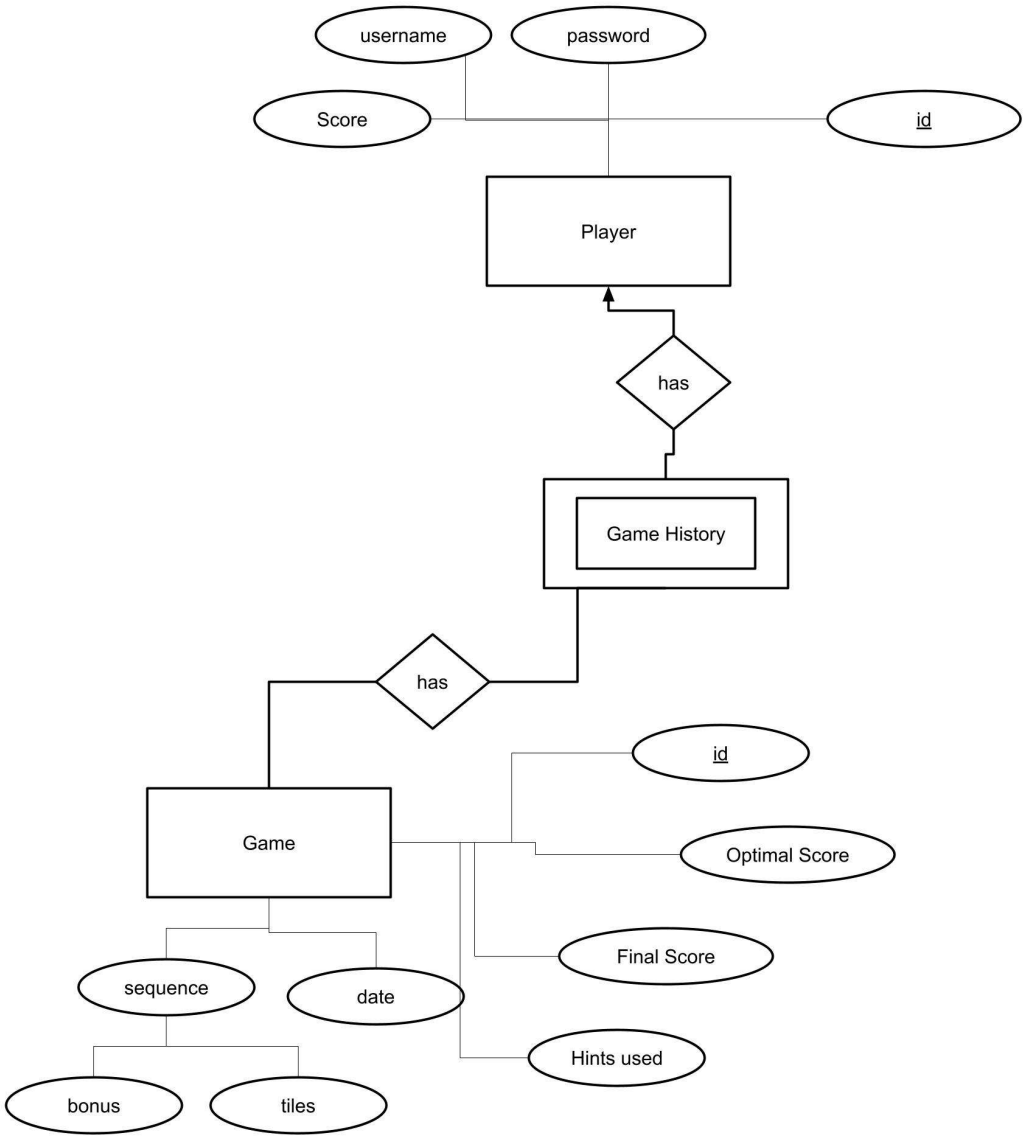
Players begin are given a sequence of tiles, each with a score and a character. The player starts with 7 of the sequence. In each turn, the player must form a word and receive a score, then receive the next characters in the sequence. The game ends upon the sequence being finished, or having no more characters to pick up. Users have the opportunities for doubles, triples and bingos. They can repeat the puzzles and try for higher scores.

Diagrams

- UML Diagram



- ER Diagram


username
password
Score
id
Player
has
Game History
has
id
Game
Optimal Score
Final Score
sequence
date
Hints used
bonus
tiles

Method used to solve the problem

We will have an array of characters[] to represent the sequence, scores[] for each character, and bonus[] to measure the placement board bonuses.

We will have an isWord() method to check if a sequence is a word.

Brute Force:

We can design a recursive brute-force algorithm, findMaxSequence(characters[i…n], bonus[i…n]) to find the highest-scoring word placement in the optimal solution, which we could reveal to the user. We would iteratively pass the current subsequence and bonus-placement, and generate as many recursive calls for the next iteration as there are possible words. We would then return the maximum scoring sequence and then compare it against the bonus to find the highest word placement.

Market space and selling points

Casual word puzzle games are a popular genre, including games like Scrabble or Wordle. This application would be similarly appealing, where a player can easily test their vocabulary knowledge while trying to get as many competitive points as possible.

Features

- User Registration and Authentication
    - Allow players to create new accounts
        - Save username, password, score
    - Admin users
        - Can view all accounts, select to remove, or to reset their password
- User Ranking Visualizations
    - Users ranked against pre-created games
    - Visual score comparisons per game and in a global ranking
- Shared, competitive games and solo randomly-generated games
- Hint feature to find the highest-scoring word placement in the optimal game, or the highest scoring word in the next turn

- Possible multiplayer:
    - A game with players on the same board and a little larger sequence, except with a harsh time limit for each of them (think of a high-scoring word in <10-20 seconds)

    - Would require up-to-date synchronization of the game sequence and decisions of other player
    - Winner is whoever scores the most

Deployment

- Create python .toml file
- Install to virtual environment
- Port-forward on router
- Start application on relevant port
- Expose to internet

Milestones

- Milestone 1 (2/5 - 2/16): Initialize technology frontends, backend development environments
- Milestone 2 (2/19 - 3/22): Develop CRUD API, UI for login forms and users
- Milestone 3 (3/25 - 4/12): Implement game UI and algorithm
- Milestone 4 (4/15 - 4/26): Assemble puzzle level prototype, add admin users
- Milestone 5 (4/29 - 5/13): Add hints