

Scribble Report

Introduction

- **Purpose of the Program:**

The program is a Scrabble-like board game with many of the features of the original game in java. It has a game board where 2 players can place letters adhering to certain rules.

- **Overview of Development:**

The program was created as a part of my learning to enhance my skills in java programming with emphasis on file handling, complex user input validation, and 2D arrays logic implementation.

Summary of Features

- **Core Functionalities:**

- You can add words to the game board following certain rules
- You can choose to save the current game state at any time and return to it later
- The game makes sure words fit within the game board and validates intersections.

- **Additional Features:**

- Prompts are very User-friendly.
- The Game Board and player points are displayed after each turn.
- Scores are calculated based on the characters of the word and the length.

3. Major Decisions

- **HashMap**

I chose to use a HashMap to assign the points to each letter which allowed for efficient and correct calculations of a user's points based on the word that they entered.

- **Intersection validation**

To validate the intersections, I adopted a method where I searched for the place in which the imputed word would intersect with the preexisting words on the board and checked if they are equal to each other. This is very robust as it works even if there are no words there.

- **Use of objects**

I chose to use objects to represent the players as there would be multiple players so it's easy and efficient to assign these players to objects instead of fields.

- **End condition**

I chose to make the game only last till one of the players reaches 200 points or more then said player wins. The game can also end whenever the players want by inputting an "N" when asked if they would like to continue playing.

4. Self-Evaluation

- **Strengths:**

In particular I am very proud of my intersection input validation as I believe I engineered a creative and robust solution to the problem presented. It deals with every outcome effectively and efficiently. I am also proud of my HashMap for the points as researched online to find a very efficient solution to the problem and learned how to use a HashMap effectively in my program.

- **Weaknesses:**

I did not implement all the optional tasks in my project such as a GUI or a word dictionary due to time constraints. If I had managed my time better i would have been able to implement these but this has taught me that in the future i will be able to achieve more things if I plan out my time better.

Conclusion

In conclusion I think my program turned out well and I am very happy with the finished result despite not being able to complete all that I would have liked too. This has shown me that I need to set a plan for my time to better manage multiple deadlines at once so that all of my work is set to a high standard. I believe this project has really improved my programming skills as I can now handle complex input validations, work well with 2D arrays and use objects efficiently in my code, leading to better results.