Project Approval: Team Trifecta (214)

Kanakabha Choudri choudrk, Lucia Cristiano cristial, Raymond Tu tur1

January 15, 2020

1 Original Project: Scrabble

URL for project: https://github.com/fayrose/Scrabble
Number of lines of code: 493
Can source code be compiled? Yes

1.1 Software Purpose

The purpose of the software is to manifest a text based version of the board game version Scrabble.

1.2 Software Scope

The scope of this project is to ultimately improve upon it in a meaningful way that can be seen by the user. These improvements will be seen in the form of additional features, better documentation of existing code, and modularization to make future changes and improvements easier..

1.3 Specialized Hardware Requirements

There are no specific hardware requirements of the chosen project.

1.4 Software Licensing

License type: The MIT License(MIT)
License allows for public redevelopment. Yes it allows for commercial use, modification, distribution, private use
Any software license that McMaster does not own? Yes, the MIT License is owned by Massachusetts Institute of Technology.

1.5 Programming Language

The programming language used in this project is Python. This language is feasible for our team because all of the members have a fair amount of experience with python and are most comfortable with it. This will make it easier to implement additional functionality and focus on improving the original project, rather than learning a new language and parsing the code.

1.6 Domain Knowledge

The game of scrabble is a household board game name that all of us have played at some point in our lives. Combining real life experience with a fairly good understanding of the game rules makes us well equipped with knowledge of the domain.

1.7 Test Cases

Some examples of test cases for the existing software would be to test the major functionalities of the game. This would include functions in the bag class, whether letters are truly random and shuffled correctly, and it keeps track of all current tiles on the board. Additionally, test cases would check whether score is tallied correctly and whether the win state is checked when the bag runs out of tiles. Furthermore, test cases would be created to test the functionality of any new features added during this project, for example different board sizes, as well as the difficulty of the computer/AI and whether this can be improved.

1.8 Adding Features

Given the current state of the project these are the following changes we plan on implementing:

1. Improve on existing features that require improvement for better user experience. For example the program currently only takes in numbers for player names when it should be strings. Further inspection of the code can possibly expose more such faults.

2. Develop a much more graphical view of the program using either pygames or TikZ libraries and additionally make it so that that the user interface remains on one screen and doesn't keep old copies of the board game visible to the players in addition to the new one created by playing a game move. This will make the game both more aesthetically pleasing and practical, vs an ASCII character based interface.

3. Potentially develop a computer/AI opponent that allows the game for singleplayer gamemodes, compared to the solely human vs human gamemodes between 2-4 players currently. This would have potential for different difficulty levels, players vs the computer, as well easier automation for testing if only 1 player's input needs to be taken into account.