

Wordle Report

CS5031-Software Engineering Practice

13 February 2023

Student ID: 220026574

Wordle Implementation (How to Play)

This code is a Java program for a game called WordleApp. To play the game, a user needs to click the run button on the program. Then, a board game with six entries is presented on the terminal. The game board consists of six rows of five letters, one row for each word. The user attempts to guess a five-letter word chosen at random from a list of words stored in a text file, which the program opens to retrieve and assign to the variable *wordList*. The user must enter a five-letter word to determine if their chosen word is the correct guess. The user cannot enter a word over or under five letters, otherwise an error message will be sent to the terminal stating that the word is either more than or less than five letters long and that the user must try again to enter a word. If a user's word isn't in the word list, then the user will have to try again by entering a different word.

The user has six attempts and is given feedback for each attempt based on whether the letter chosen by the user exists in the word from the word list, and if it exists whether it is in the correct position. Feedback is displayed on the game board. The letters are displayed in either green, if the letter exists in a word and is in the correct position, yellow if the letter exists in the word but is not in the correct position, or if the letter does not exist in the word it resets to the default color of the terminal. The game ends when either the player correctly guesses the word or after the six guesses are entered and the correct answer wasn't entered, in which the user will be given the word at the end of the game.

The Test-Driven Approach

For my test WordleApp file, I programmed test cases using JUnits for Java. I tried to implement each test case to align with several methods with my WordleApp file. I wrote the failed test code first then continued by writing a test that checks for the desired action of the code.

I started with easier methods that would give a return of assert equals or assert null. This was useful for me since this would, for example, let me know if my text file was found or it was null if it didn't upload/exist. I would write small amounts of code to see if test cases would pass and commit in GIT when I needed to update my code. Each section of code was tested first then written in my main file. I did this to see if my test cases would be accepted. The test cases were used to determine/ensure that errors would not occur in the future. Once the tests passed, I reviewed the code and made any necessary changes to ensure it was clean, maintainable, and efficient. However, I had some difficulty in passing the tests for some of my board game methods, because I had too many if statements or because they were too long.

Refactoring Approach

I first designed my wordle game to make it functional. Then I broke it apart into more methods so that the code would be easier to read. I used only one main file for the program, so it would be easier to read what my methods were doing instead of using different classes. I also commented when needed for the reader to understand how my word game functioned and to explain what each section of code did. I made sure I didn't use magic numbers for making my display board game, since this isn't good practice. The board game length was used instead. Each time refactoring was done incrementally, I would review the tests results to make sure that the code still behaved as expected. I also used parameters for some of my methods. I made the code readable to other programmers who may have little knowledge of Java.

Other Software Development Practices

I would review my code several times to see if it was working properly using the debugging mode in Visual Studio. Debugging, which is typically a time-consuming process, is the process of identifying and fixing bugs in your code. By using tools like debuggers, the debugging process can be more efficient and effective.

Extra Functionality

I made the game board print to the terminal, which resulted in the better appearance of the words on the board.

References

Wordle - A daily word game. (n.d.). Wwww.nytimes.com.
<https://www.nytimes.com/games/wordle/index.html>