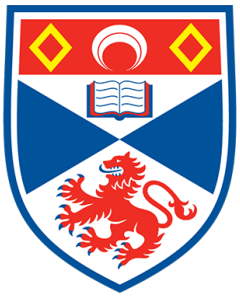
CS5031 – Software Engineering Practice

Report

220031985



University of
St Andrews

28th January 2023

Introduction:

Wordle is a **web-based word game** created and developed by Welsh software engineer Josh Wardle and owned and published by the New York Times Company since 2022 [1].

How to play [2]:

1. You have six tries to guess the five-letter Wordle of the day.

2. Type in your guess and submit your word by hitting the "enter" key on the Wordle keyboard.

3. The colour of the tiles will change after you submit your word.

* A yellow tile indicates that you picked the right letter, but it's in the wrong spot.
* The green tile indicates that you picked the right letter in the correct spot.
* The red tile indicates that the letter you picked is not included in the word at all.

4. Continue until you solve the Wordle or run out of guesses. Good luck!

Note: The game will "exit" or "restart" on the user giving the input exit or restart.

The game can be played through the command line interface. After each guess, the appropriate feedback is generated for the user, with information on the number of attempts, the number of correct letters in the correct position, and the correct letters in the wrong position.

Finally, after all, attempts, the user is provided with score results based on the number of attempts taken to guess the word.

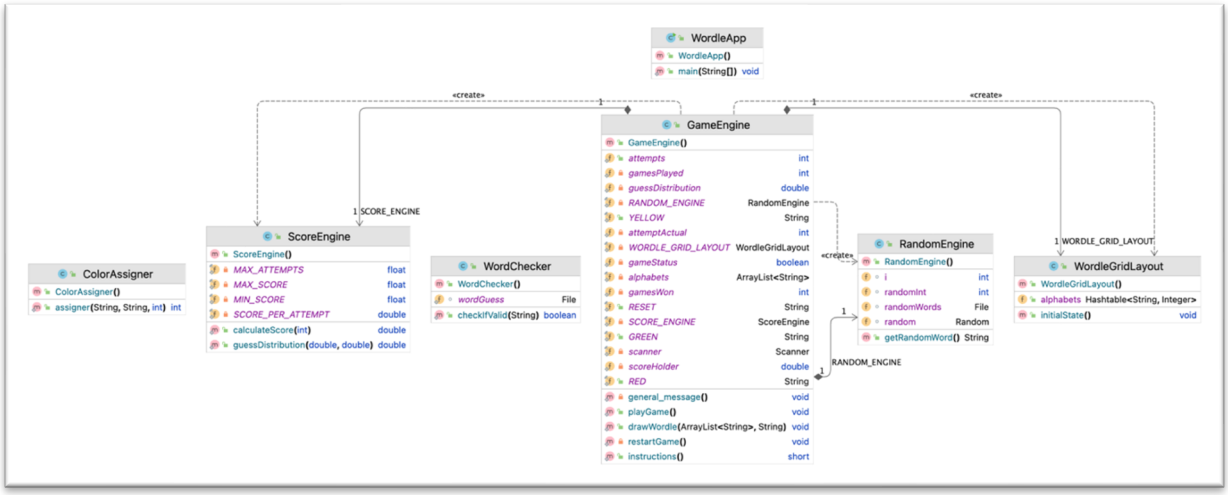


Fig: UML Class Diagram showing Wordle Game implementation

The game consists of the following classes:

WordleApp.java – *responsible for loading the wordlist and starting the game.*

GameEngine.java – *Responsible for the logic of the game.*

RandomEngine.java – *Used for generating a random word from resources.*

WordChecker.java – To *check if the guessed word is valid or not*

ScoreEngine.java – *Responsible for calculating the probability and Guess Distribution of the player.*

ColorAssigner.java – *Allocates the colour state for the characters.*

WordleGridLayout.java – *Initialises and holds the colour state for the characters.*

Test–Driven Approach:

The game consists of the following test classes:

ClassLoaderTest.java – *test case to check if all classes are loaded.*

GameEngineTest.java – *Test scenarios for GameEngine.java*

RandomEngineTest.java – *Test scenarios for RandomEngine.java*

ScoreEngineTest.java – *Test scenarios for ScoreEngineTest.java*

WordCheckerTest.java – *Test scenarios for WordChecker.java*

Wordle's development was done per the Test-driven development (TDD) approach.

With initial WordleApp.java present with a failed test case scenario, test cases were written to call all classes, following the TDD approach. JUnit 5 was used as the unit-testing framework.

Test for any class was written before implementing any code and then the code was developed to pass these tests. By following this standard, I verified the code's functionality at every stage of the development and verified that the code was working as expected.

The trickiest part was writing tests before code implementation for GameEngine.java as the class tied up all the other classes and also contained the logic of the game.

Refactoring Approach:

Throughout the development process, the code was refined and refactored were ever required. Care was taken to ensure that while following the refactoring approach, making sure that the code was clean and easy to read and followed the best coding practices. JavaDoc was added to methods that required so. Care has been taken to *Error handling* such as arithmetic expressions (divide by zero) or Arrays out of bounds.

The code has been refactored several times during development (Git log) to remove duplication code and ensure that the code was properly structured and organised.

Software Development Practices Used:

The code was version–controlled using Git, thus making it easy to revert to a previous state if anything went wrong. Branches were updated after major changes and care was taken to warnings before committing and pushing into the repository.

The code was also documented using comments or JavaDoc wherever necessary. The code could be compiled into a portable Jar file using Maven and could be run from the command line.

Extra Functionality:

Other than the use of the *'restart'* command for the user to restart the game while in the middle of it, there are no additional functionalities as the requirement was to follow the rules of the original Wordle game.

Conclusion:

The development of the Java version of Wordle followed the rules of the original game and used a test-driven development process. The code has been refactored throughout the development process to make sure it was clean, say to read and follows the best practices.

The game can be played through the command line interface by invoking WordleApp.java and the game provides appropriate feedback and score to the user.

References:

[1] Wordle, https://en.wikipedia.org/wiki/Wordle, Accessed on 30/01/2023.

[2] Best Wordle Tips, https://www.nytimes.com/2022/02/10/crosswords/best-wordle-tips.html, Accessed on 30/01/2023.

Total Words: 767