

Rabisa Report

by Rabisa Ali

Submission date: 21-Nov-2025 03:30PM (UTC+0500)

Submission ID: 2822984221

File name: Final_Project_Report_11.docx (565.75K)

Word count: 807

Character count: 4398

1

Final Project Report — Programming Fundamentals

University Name: FAST (National University of Computer & Emerging Sciences

Karachi Campus)

Department: Department of Computer Science

Course: Programming Fundamentals

Project Title: Rewordling Wordle

Submitted By: Rabisa Ali (25K-0876) Syeda Marium Fatima (25K-0970)

Submitted To: Ms.Kinza Mushtaq

Semester: Fall 2025

Date: November 21, 2025

Abstract

Rewordling wordle is a console-based game developed in C language to implement online wordle game efficiently. A player will be given ten consecutive Wordle puzzles. However, a single failed attempt to guess the word will lead to the termination of program. At the end, the program will calculate the average number of guesses, best guess, and a percentile ranking of pre-hard coded player performance data collected from google. This project applies fundamental programming concepts such as loops, arrays, and functions. This takes Wordle beyond a casual guessing game to something much more intense.

1. Introduction

Wordle is an online word puzzle game where a player has six attempts to guess a five-letter word. Feedback is given through colored tiles: green for a correct letter in the right spot, yellow for a correct letter in the wrong spot, and red for a letter not in the word at all. The goal is to guess the hidden word using this feedback within the limited attempts. Although there are numerous Wordle implementations online, most of them offer instant win or loss results without any in-depth analytical insights.

2. Objectives

- To find the hidden word using the feedback within the attempts.
- To find out the average number of tries it took the user to judge the word
- To find out the best guess
- To provide a percentile ranking
- To reinforce programming concepts like loops, arrays, and functions.
- To provide a simple, user-friendly interface.

3. System Design

System Overview

Flow of the program:

Start → Load the words into memory → Display instructions → Ask the user to enter the word
→ Compare the word with the original word and change the font color accordingly → If the user guessed the word correctly within the set number of tries, ask him if he wants to continue. If yes then continue, else end the game → End the game by displaying average number of guesses in which the user was able to guess the word correctly, best guess, and his percentile ranking → Exit.

Algorithm

1. Start the program
2. Load the words into memory
3. Display instructions
4. Ask the user to enter the word
5. Compare the word with the original word and change the font color accordingly
6. If the user guessed the word correctly within the set number of tries, ask him if he wants to continue. If yes then continue, else end the game.
7. End the game by displaying average number of guesses in which the user was able to guess the word correctly, best guess, and his percentile ranking.
8. End

Input & Output

Input: Guess of the user.

Output: Display instructions, feedback on each word, best guess, average number of tries and the percentile ranking.

1

4. Implementation

Language: C

Compiler/IDE: Code::Blocks / Dev C++ / GCC

Key Features

- Sequential **gameplay** of up to 10 word puzzles in a single session
- Immediate session termination upon first failed attempt
- Per-letter feedback system highlighting correct, misplaced, and incorrect letters
- Tracking and averaging of total guesses across solved puzzles
- Hardcoded percentile evaluation based on statistical benchmarks

Code Snippet

```
int processGuess(char *answer, char *guess){  
    int correct = 0;  
    3  
    for (int i = 0; i < 5; i++){  
        if (guess[i] == answer[i]){  
            printf("\033[92m%c\033[0m", guess[i]);  
            correct++;  
        } else {  
            int found = 0;  
            for (int j = 0; j < 5; j++){  
                if (guess[i] == answer[j] && i != j){  
                    found = 1;  
                    break;  
                }  
            }  
            if (found)  
                printf("\033[93m%c\033[0m", guess[i]);  
            else
```

```
    printf("\033[91m%c\033[0m", guess[i]);

}

}

printf("\n");

return correct;
}

int game(char* answer, char* guess){

int guesses = 0;

while (guesses < 6){

printf("\033[94mEnter a 5-letter word: \033[0m");

scanf("%5s", guess);

guesses++;

if (processGuess(answer, guess) == 5){

return guesses;

}

}

return 7;
}
```

Sample Output:

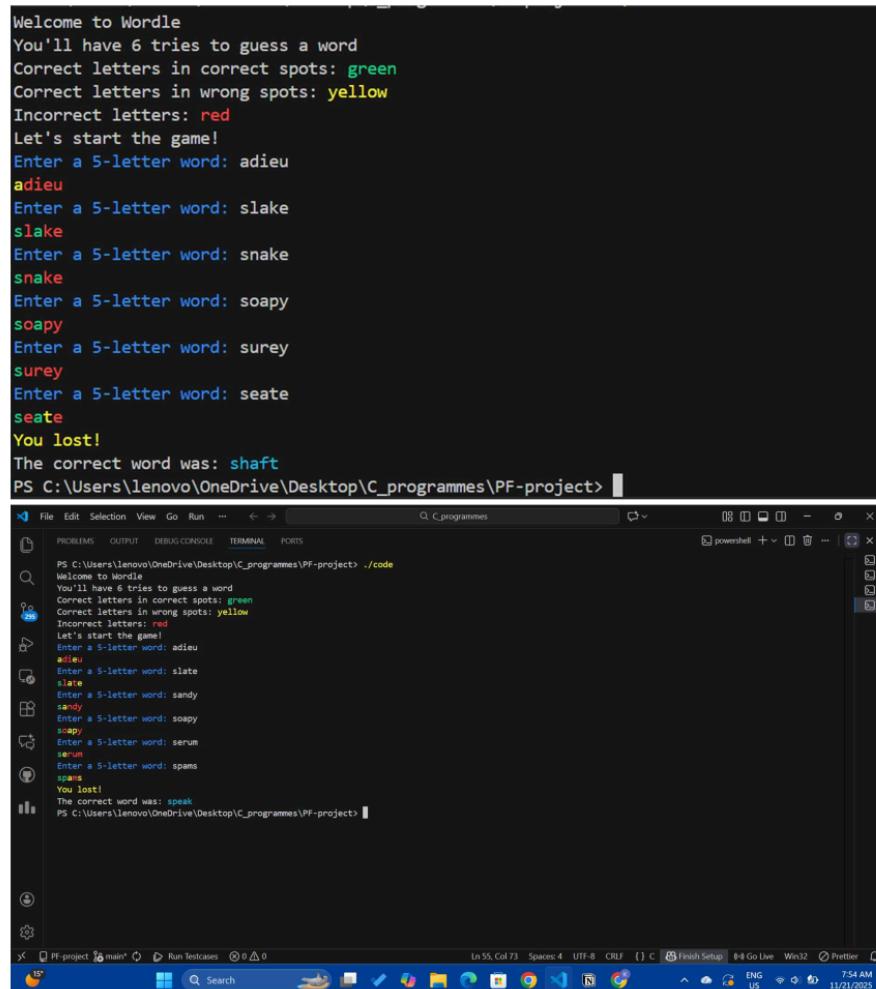
The screenshot shows a terminal window titled "C_programmes" with the following text output:

```
Welcome to Wordle
You'll have 6 tries to guess a word
Correct letters in correct spots: green
Incorrect letters: red
Let's start the game!
Enter a 5-letter word: adieu
adieu
Enter a 5-letter word: slate
slate
Enter a 5-letter word: plai
plai
Enter a 5-letter word: plaint
plaint
Enter a 5-letter word: plant
plant

You Won!
Do you wish to play again? Enter Yes or No: yes
Enter a 5-letter word: adieu
adieu
Enter a 5-letter word: slate
slate
Enter a 5-letter word: nerdy
nerdy
Enter a 5-letter word: ranter
ranter
Enter a 5-letter word: blare
blare
Enter a 5-letter word: transe
transe
You lost!
The correct word was: frame
```

The terminal window is part of a larger interface with tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", and "TERMINAL". The status bar at the bottom shows "In 55, Col 73" and "11/21/2025".

5. Testing and Results



```
Welcome to Wordle
You'll have 6 tries to guess a word
Correct letters in correct spots: green
Correct letters in wrong spots: yellow
Incorrect letters: red
Let's start the game!
Enter a 5-letter word: adieu
adieu
Enter a 5-letter word: slake
slake
Enter a 5-letter word: snake
snake
Enter a 5-letter word: soapy
soapy
Enter a 5-letter word: surey
surey
Enter a 5-letter word: seate
seate
You lost!
The correct word was: shaft
PS C:\Users\lenovo\OneDrive\Desktop\C_programmes\PF-project>
```

The screenshot shows a terminal window in a dark-themed code editor. The terminal is running a Wordle-like game. It starts by welcoming the user and explaining the rules: 6 tries, green for correct letters in correct spots, yellow for correct letters in wrong spots, and red for incorrect letters. The user is prompted to "Let's start the game!" and asked to "Enter a 5-letter word:". The user enters "adieu", which is highlighted in green. The next few attempts ("slake", "snake", "soapy") are also highlighted in green. Subsequent attempts ("surey", "seate") are highlighted in yellow. Finally, the user enters "spans", which is highlighted in red. The message "You lost!" is displayed, followed by the correct word "shaft". The terminal prompt "PS C:\Users\lenovo\OneDrive\Desktop\C_programmes\PF-project>" is shown at the bottom.

```
File Edit Selection View Go Run ... C programmes TERMINAL PORTS
PROBLEMS OUTPUT DEBUG CONSOLE powerhell + x
Enter a 5-letter word: adieu
adieu
Enter a 5-letter word: slate
slate
Enter a 5-letter word: plai
plai
Enter a 5-letter word: plaint
plaint
Enter a 5-letter word: plant
plant

You Won!
Do you wish to play again? Enter Yes or No: yes
Enter a 5-letter word: adieu
adieu
Enter a 5-letter word: slate
slate
Enter a 5-letter word: nerdy
nerdy
Enter a 5-letter word: rante
rante
Enter a 5-letter word: blare
blare
Enter a 5-letter word: trane
trane
You lost!
The correct word was: frame
You played Wordle 1 times and guessed the word(s) in an average of 12.00 steps
Your best attempt was 5 guesses
Your best attempt ranks in the 35.0% of Wordle players worldwide
PS C:\Users\lenovo\OneDrive\Desktop_C_programmes\PF-project>

File Edit Selection View Go Run ... C programmes TERMINAL PORTS
PROBLEMS OUTPUT DEBUG CONSOLE powerhell + x
Welcome to Wordle
You'll have 6 tries to guess a word
Correct letters in correct spots: green
Correct letters in wrong spots: yellow
Incorrect letters: red
Let's start the game
Enter a 5-letter word: adieu
adieu
Enter a 5-letter word: slate
slate
Enter a 5-letter word: plai
plai
Enter a 5-letter word: plaint
plaint
Enter a 5-letter word: plant
plant

You Won!
Do you wish to play again? Enter Yes or No: yes
Enter a 5-letter word: adieu
adieu
Enter a 5-letter word: slate
slate
Enter a 5-letter word: nerdy
nerdy
Enter a 5-letter word: rante
rante
Enter a 5-letter word: blare
blare
Enter a 5-letter word: trane
trane
You lost!
The correct word was: frame
```

The program performed successfully for all test cases. It handled both correct and incorrect guesses efficiently, produced accurate best guess and percentile. Execution speed was near-instant, and the program required minimal system resources.

6. Conclusion, Limitations & References

Conclusion

The Rewording Wordle successfully demonstrates the application of basic programming principles. It takes Wordle beyond a casual guessing game to measurable performance evaluation. The project strengthened understanding of arrays, loops, conditional statements, and functions.

Limitations

- Data of average number of guesses and best guess is lost when the program closes (no file handling yet).
- No graphical interface, purely console-based.

Future Enhancements

- Add file handling to store records permanently.
- Implement a graphical or web interface.

References

- Let Us C by Yashavant P. Kanetkar
- <https://www.geeksforgeeks.org/c-programming-language/>
- <https://www.sportskeeda.com>
- <https://www.nytimes.com › wordle-bot-year-in-review>

Rabisa Report

ORIGINALITY REPORT



PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	4%
2	www.thetablereadmagazine.co.uk Internet Source	4%
3	www.coursehero.com Internet Source	3%
4	wordlegame.org Internet Source	3%

Exclude quotes

On

Exclude matches

< 3 words

Exclude bibliography

On