TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

Proposal

On

Wordle in C

Submitted By:

Kushal Saud (THA081BCT016)

Suhan Man Pradhan (THA081BCT045)

Yangdi Bhote Shalaka (THA081BCT046)

Yujen Shrestha (THA081BCT048)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

February, 2025

ABSTRACT

Wordle is a word-guessing game that challenges players to identify a five-letter secret word within six attempts. The game challenges players to guess a five-letter word within six attempts, providing feedback on letter accuracy and positioning. Its simple yet strategic gameplay has made Wordle a widely popular. This project aims to implement a Wordle clone using C programming language. The project involves word validation, input handling, and result evaluation. Additionally, a dynamic word list was integrated to enhance replayability.

Keywords: C Programming, Wordle, Word game, User Input Handling, String Manipulation

Table of Contents

List of Figures

List of Abbreviations

OOP	Object Oriented Programming
CLI	Command-Line Interface
GCC	GNU Compiler Collection
GNU	Gnu’s Not Unix
VS	Visual Studio
MinGW	Minimalist GNU for Windows
ASCII	American Standard Code for Information Interchange
ANSI	American National Standards Institute
API	Application Programming Interface

1. INTRODUCTION

1.1 Background

Wordle is a popular word puzzle game where players have six chances to guess a five-letter word. Each guess provides feedback in the form of colored tiles: green for the correct letter in the correct position, yellow for a correct letter in the wrong position, and gray for a letter not in the word. [1]

The game was created by software engineer Josh Wardle during the COVID-19 pandemic. It was released to the public in October 2021 and has quickly gained popularity since then. In January 2022, The New York Times Company acquired Wordle, integrating it into their games platform while maintaining its original gameplay and daily puzzle format. The game's minimalist design and daily challenge continue to attract millions of players worldwide. [2]

1.2 Motivation

Our group members explored various project ideas and ultimately chose to venture into simple game development by recreating Wordle. We were inspired by its straightforward yet engaging gameplay. By developing Wordle in C, we aim to enhance our programming skills, particularly in logic implementation, input handling, and memory management. This project not only allows us to recreate a well-loved game but also provides a valuable learning experience in game design and algorithm development.

1.3 Problem Definition

This project focuses on developing a Wordle clone in C. It requires us to integrate an efficient word validation and input handling algorithm, in order to make sure the word is in the Wordle glossary. Then, the inputted word needs to be evaluated for all the matching letters and their placements. This is done until the player runs out of tries or guesses the word. We also plan to include a stats section to provide feedback on the player’s progress.

1.4 Objectives

The main objectives of our project are listed below:

- Recreate the core gameplay of Wordle
- Develop a dynamic word list, containing all words in the Wordle glossary
- Include a stats section to track player progress

2. LITERATURE REVIEW

2.1 History of Word Games

The history of word games is as old as language itself. As long as humans have been communicating with words, they have also been playing with them. Some of the earliest forms of word games were riddles, puns, anagrams, acrostics, palindromes, word squares, and more. Word games have been used for education, entertainment, and religious purposes.

2.2 Rise of Wordle

Josh Wardle, a software engineer in Brooklyn, knew his partner loved word games, so he created a guessing game for just the two of them. As a play on his last name, he named it Wordle. But after the couple played for months, and after it rapidly became an obsession in his family's WhatsApp group once he introduced it to relatives, Mr. Wardle thought he might be on to something and released it to the rest of the world in October, 2021. [3]

On Nov. 1, 90 people played the game daily. Just over two months later, on Jan. 2 2022, Wordle had more than 300,000 daily players. It was clear how quickly the game had captured the public's attention. This led to The New York Times Company acquiring the game for an undisclosed, low-seven figure fee later that month. [3]

This acquisition helped drive tens of millions of new players to The Times' puzzle section and app. In 2022, Wordle became the top search term on Google globally. Remarkably, seven out of the top 10 word definitions searched for that year were solutions to Wordle puzzles. Wordle's huge success has inspired various clones and adaptations, such as Quordle, Heardle, Worldle, Squirdle, etc. [3]

2.3 Game Mechanics

Wordle has a very simple game mechanic. Every day, users need to guess one single five-letter word. However, our rendition of Wordle allows for unlimited amount of plays. The players are

given six tries to guess the correct word. They are provided with feedback for each guess in the form of colored letters indicating correctness. The game uses a feedback system where:

- Green indicates the letter is correct and in the correct position.
- Yellow indicates the letter is correct but in the wrong position.
- Gray means the letter is not in the word.

This is done until the player guesses the word or runs out of guesses. [2]

2.4 Programming Languages and Game Development

Game development is a huge part of software engineering. Due to its complex nature, and the constant need of updating and prototyping, programming languages such as C++, Java, Python, and C# are used. Due to lack of OOP features, it is significantly harder to manage complex game mechanics in C. However, for simple games such as Tic-Tac-Toe, Snake, Tetris, and Wordle, C can be used to develop them easily. C's minimalistic style and simplicity makes C a great learning foundation for students to learn game development.

Due to these reasons, we have come to the conclusion that recreating Wordle in C is an excellent exercise in understanding core programming concepts like string manipulation, user input handling, and control flow. The simplicity of Wordle's gameplay aligns well with C's strengths, allowing developers to focus on logic and efficiency without the overhead of graphical rendering.

3. PROPOSED SYSTEM ARCHITECTURE

3.1 Block Diagram of System Architecture

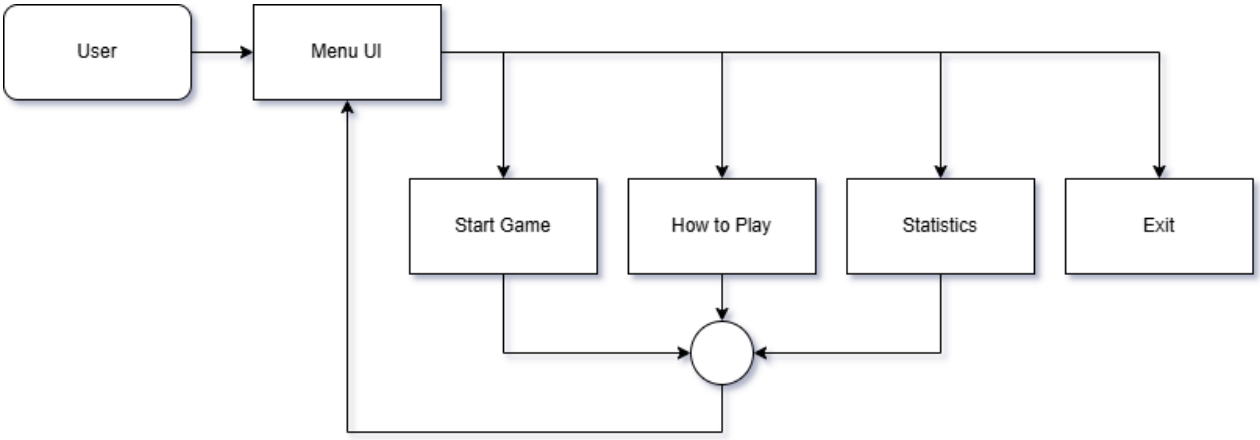

Figure 1 Block Diagram of System Architecture

3.2 Data Flow Diagram

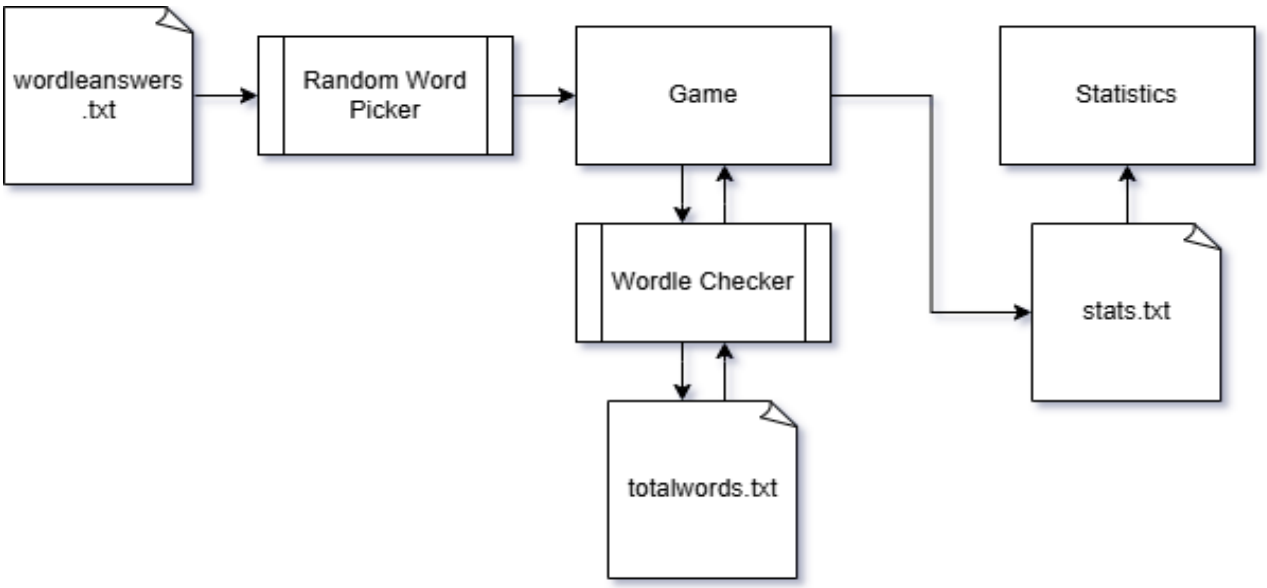

Figure 2 Data Flow Diagram

3.3 Parts of Program

3.3.1 Main Menu

The home page contains the title of the game ‘WORDLE’ and 4 options are to the user: ‘Start Game’, ‘How to Play’, ‘Statistics’, and ‘Exit Game’.

3.3.2 Start Game

This will start the game. Firstly, a random word is picked from and text file containing all Wordle solutions named “wordleanswers.txt”. This is saved by the program and the player will need to guess this word. Each time they input a word, two things are checked: whether the word has enough letters, and if the word exists in the Wordle glossary, stored in “totalwords.txt”. If both values are true, the word will then be checked with the word to guess and the result will be printed out accordingly. This will continue until the player guesses the word or guesses incorrectly six times. After a game is over, it edits the data saved in stats.txt accordingly. Then, it goes back to the main menu.

3.3.3 How to play

This section simply tells the player how the game works and how to play the game.

3.3.4 Statistics

This section tells the user about the total number of games played, the number of games won, along with how many games were won with what number of guesses. This will also show the win streak. All these data are stored in the “stats.txt” file. This section will help users keep track of their progress. The user may also reset the stats if they wish.

3.3.5 Exit Game

This exits the game.

4. METHODOLOGY

4.1 Tools and Technology

VS Code is used as the main code editor due to its lightweight and cross-platform capacity along with useful features like extensions, and debugging tools.

GCC is used as the compiler as it is highly reliable and offers excellent optimization capabilities.

4.2 Word Selection and Dictionary Management

There are 2309 possible answers to a Wordle. The project implements all 2309 answers into the game by storing them in a text file “wordleanswers.txt”. Then, a random integer in the range from 0 to 2308 is taken using functions rand() and srand(). A function goes to that line to scan a word which is selected as the word to be guessed.

4.3 Input Handling

A user will try to guess the correct word. They may input only alphabetical letters. Any other characters won’t be taken as input. Even though there are only 2309 possible answers, there are 12,947 words that can be inputted. These all are stored in another text file “totalwords.txt”. For the inputted word to be accepted, it must have 5 letters and it must be in the list of words. Since 12,947 is a very large number, we can’t run a basic loop that many times without hampering the execution speed of the program. So, we use binary search in order to search for the word. Binary search is an efficient algorithm for finding an item from a sorted list of items. It works by repeatedly dividing in half the portion of the list that could contain the item, until you've narrowed down the possible locations to just one. After the word passes both criteria, it moves on to the next step, which is letter validation.

4.4 Letter Validation Logic

Firstly, the secret word is copied onto another string, so that we can work on that one. We make an array which stores the color value for each letter (0 for gray, 1 for yellow and 2 for green). By

default, all values in the array is stored as a zero. The inputted word is then checked for any green colors (i.e. correct letter in correct placement). If it is a match, the corresponding letters in the secret word is changed to a zero, or any other non-alphabetical character and the corresponding value in the array is changed to a 2. Then, it checks for yellow colors (i.e. the letter exists in the main word, but isn't in the correct position). This all is done in order to remove any kind of error that may come due to repeated letters.

4.5 Feedback Generation

Next step is to print the characters with the correct feedback for each of them. We opted to go for color feedback, as it is the closest to the original Wordle. This can be done in two ways. Firstly, using ANSI escape codes. However this can be done only on newer versions of PowerShell. Since we wanted this project to be able to run in most computers, we chose to go for the second option which is using windows API. Here we use the SetConsoleTextAttribute( ) function from the windows.h header file. This allows us to change the foreground as well as the background text color. Using different intensities and mixtures of red, green and blue, most colors can be displayed on the console.

Figure 3 Color Feedback

This does require writing a slightly longer code and can get tedious for multiple colors. There are many alternatives which may not be as visually intuitive, but can work quite effectively. Some of them include:

i. Symbol-Based Feedback
This method utilizes different symbols to convey the status of each letter. We could use '*' for correct letter in the correct position, '?' for correct letter in incorrect position, and '-' for incorrect letter. It would look something like this:
Guess: HORSE
Feedback: ? ? - - *

ii. ASCII Art Feedback
This method utilizes ASCII characters such as brackets to convey the status. We could use '[ ]' for correct placements, '{ }' for incorrect placements, and '- -'for incorrect letters. For example:
Guess: HORSE
Feedback: {H} {O} -R- -S- [E]

iii. Number Indicators

Here, Numbers are used to signify status. 2 for correct placements, 1 for incorrect placements and 0 for incorrect letter. For example:

Guess: HORSE

Feedback: 1 1 0 0 2

iv. Sound Alerts

We can also use sound alerts to show the status. Using the Beep( ) function in windows.h header file, varying pitch of sound can be created. For example, we can assign high pitch to correct placements, medium pitch to incorrect placements and low pitch to incorrect letters.

4.6 Statistics Tracking

After a game is over, the essential data is written to "stats.txt" which will keep track of the statistics of the user. This can be accessed via the Statistics section in the main menu. We only need to store total number of games played, and the number of games won within each number of guesses. The remaining statistics such as number of games won, the percentage of games won and the best try can be calculated. We can also integrate an option to reset stats, which will work by editing all values to 0.

5. TIME ESTIMATION

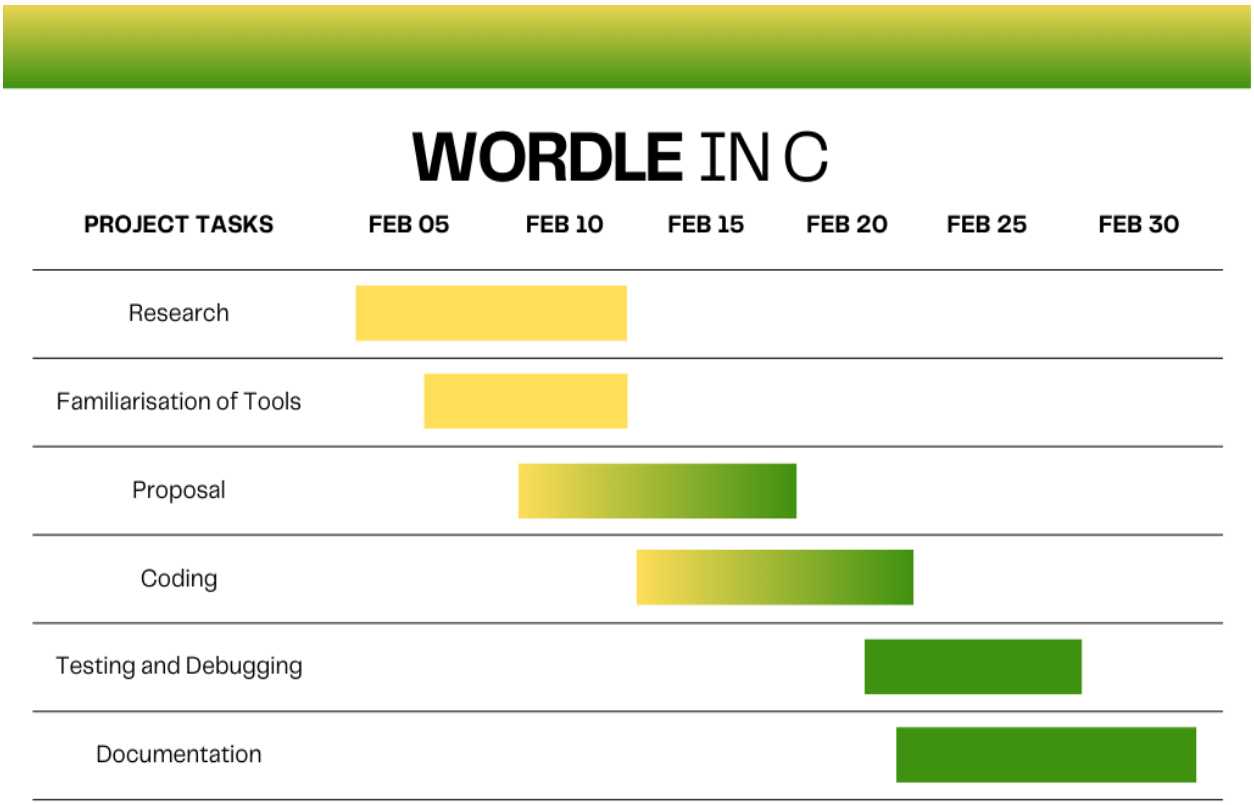

Figure 4 Time Estimation Gantt Chart

6. FEASIBILITY ANALYSIS

6.1 Technical Feasibility

Developing a basic version of the popular word-guessing game Wordle using the C programming language is highly achievable from a technical standpoint. C is well-suited for text-based applications due to its efficiency, flexibility, and wide range of built-in functions for handling strings and user input. Since Wordle primarily relies on logic, string manipulation, and simple user interactions rather than complex graphics or animations, it can be effectively implemented using fundamental C programming concepts. Additionally, C provides libraries such as stdio.h and string.h, which make handling words and user input straightforward. The simplicity of the game logic makes it an ideal project for both beginners and experienced programmers looking to practice algorithm development and logical problem-solving.

6.2 Economic Feasibility

From an economic perspective, developing Wordle in C is highly cost-effective. A C compiler, such as GCC or MinGW, is available for free, and there are numerous online C compilers that allow coding without any software installation. Since the game does not require any advanced graphical components, no additional expenses are needed for game engines or external libraries. This makes it an excellent choice for developers with limited financial resources who want to build a functional and engaging project with minimal investment.

6.3 Operational Feasibility

Running a Wordle game in C is straightforward, as it only requires a system with a C compiler. The game operates through a simple text-based interface, making it compatible with any operating system that supports C programming. To ensure ease of maintenance and future updates, the project will follow structured programming principles, including modular functions and clear documentation. This will allow developers to easily modify the word list, improve the game logic, or add features such as difficulty levels and score tracking. Given these factors, developing a text-

based Wordle game in C is a practical and feasible project that can be executed efficiently with minimal resources.

6.4 Schedule Feasibility

Developing a basic Wordle game in C can be completed in a short time frame, typically within a few days to a couple of weeks, depending on experience level. Since the game is not graphically demanding, testing and debugging can be done quickly without specialized tools, making the project realistic and achievable within a short timeline.

References

[1] The New York Times. (2022). *Wordle - The Daily Word Game.* The New York Times. https://www.nytimes.com/games/wordle

[2] Rosenberg, A. (2024, December 3). *What is 'Wordle'? Here's everything you need to know.* Mashable https://mashable.com/article/wordle-word-game-what-is-it-explained

[3] Victor, D. (2022, January 3). *Wordle Is a Love Story*. The New York Times. https://www.nytimes.com/2022/01/03/technology/wordle-word-game-creator.html