

**B.M.S COLLEGE OF ENGINEERING BENGALURU**  
**Autonomous Institute, Affiliated to VTU**



**SPC AAT Report on**

**“WORD SCRAMBLER”**

*Submitted in partial fulfillment of the requirements for AAT*

Bachelor of Engineering  
in  
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

*Submitted by:*

**ANIRUDH BHAT H**

**1BM25A1368-T**

**ANADI JAIN**

**1BM25AI320-T**

Department of ARTIFICIAL INTELLIGENCE & MACHINE LEARNING  
B.M.S College of Engineering  
Bull Temple Road, Basavanagudi, Bangalore 560 019  
2025-2026

**B.M.S COLLEGE OF ENGINEERING**  
**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE**  
**LEARNING**



**DECLARATION**

We, ANIRUDH BHAT H AND ANADI JAIN students of 1<sup>st</sup> Semester, B.E, Department of - ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING, BMS College of Engineering, Bangalore, hereby declare that, this AAT Project entitled "WORD SCRAMBLER" has been carried out in Department of AIML, BMS College of Engineering, Bangalore during the academic semester Sep 2025 – Jan 2026. We also declare that to the best of our knowledge and belief, the AAT Project report is not from part of any other report by any other students.

**Student Name**

**Student Signature**

**1. ANIRUDH BHAT H**

**2. ANADI JAIN**

**BMS COLLEGE OF ENGINEERING**  
**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE**  
**LEARNING**



**CERTIFICATE**

This is to certify that the AAT Project titled “**WORD SCRAMBLER**” has been carried out by **ANIRUDH BHAT H (1BM25AI368-T)** and **ANADI JAIN (1BM25AI320-T)** under the mentorship of **PROF. ANITA HARISH KENCHANNAVAR** during the academic year 2025-2026.

Signature of Faculty in Charge

## TABLE OF CONTENTS

<b>Sl. No.</b>	<b>Title</b>	<b>Page no.</b>
<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Algorithm</b>	<b>6</b>
<b>3</b>	<b>Flowchart</b>	<b>8</b>
<b>4</b>	<b>Source code</b>	<b>9</b>
<b>5</b>	<b>Results (screenshots)</b>	<b>13</b>
	<b>References</b>	<b>19</b>

## 1. INTRODUCTION

The **Word Scramble Project** is an interactive console application designed to test your puzzle-solving skills while demonstrating the power of the C programming language. Created for our Alternative Assessment Task (AAT), this project takes a simple concept—unscrambling a jumbled word—and builds a complete, functional software experience around it.

At its core, the program works by selecting a random word from a built-in dictionary of fruit names (like "apple" or "mango"). It then applies a shuffling algorithm to mix up the letters completely. For instance, the word "orange" might appear on the screen as "ganreo." To achieve this, I utilized standard C libraries such as `<string.h>` to handle the text processing and `<stdlib.h>` to generate the random patterns. A key feature of the code is how it uses the computer's internal clock (`time.h`) to seed the randomizer. This ensures that every time you open the game, you get a completely new sequence of puzzles, keeping the gameplay fresh and unpredictable.

We designed the logic to be fair but competitive. The game runs in a continuous loop, allowing the user to play for as long as they wish. For every scrambled word, you are given two attempts to guess the answer. A correct guess boosts your score, while failing both attempts reveals the answer and applies a small penalty to your score. This "negative marking" adds a layer of difficulty similar to real-world exams.

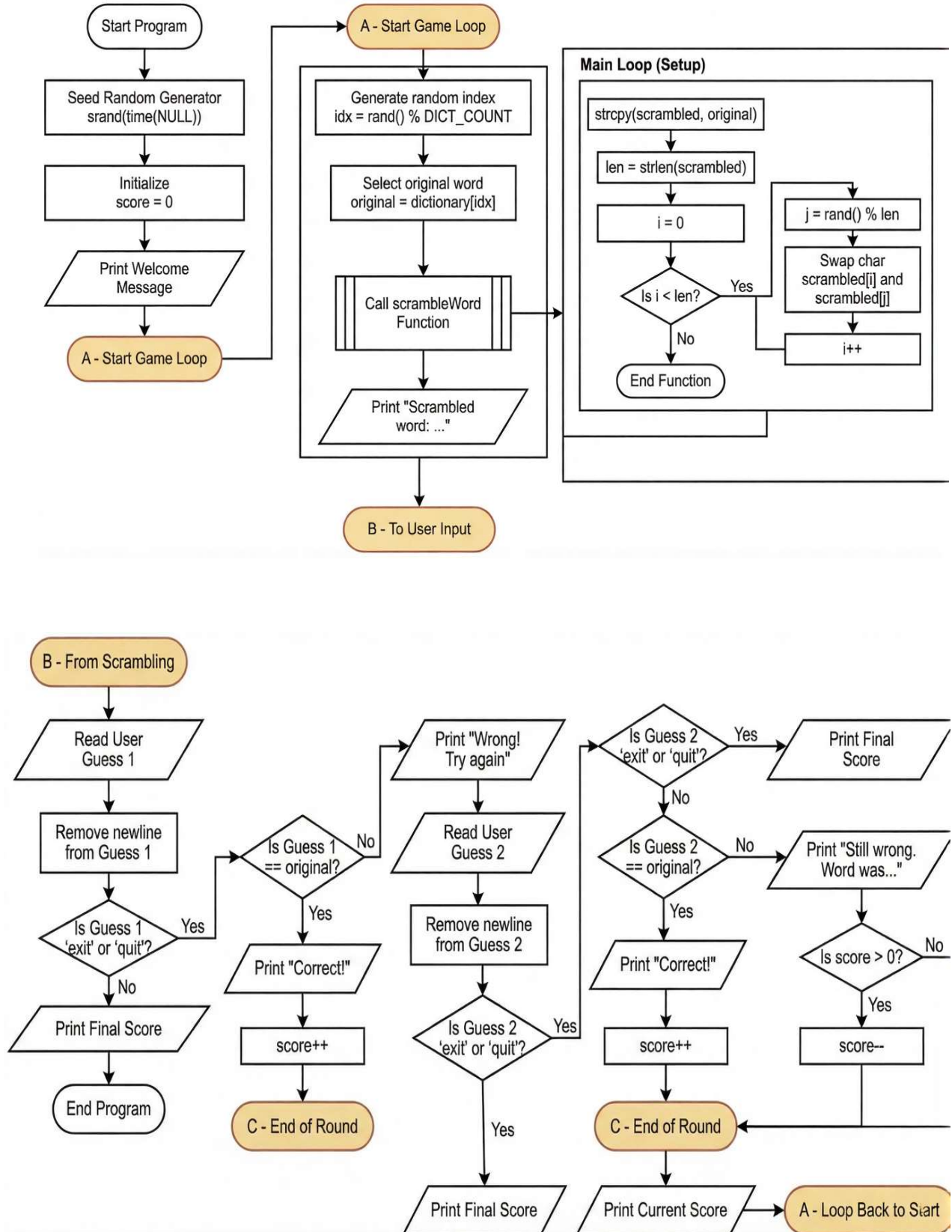
Beyond just the game mechanics, this project focuses on robust programming practices. We prioritized safe user input (using `fgets` instead of `scanf`) to ensure that typing errors or unexpected characters don't crash the program. Overall, this project is a practical summary of our learning this semester, showing how arrays, loops, and pointers come together to create something fun and interactive.

## 2. ALGORITHM

1. **Step 1:** Start
  - 1.1. Initialize score = 0.
  - 1.2. Display welcome message and instructions.
2. **Step 2:** Initialize Dictionary
  - 2.1. Store 'n' predefined words in an array called dictionary.
3. **Step 3:** Seed Random Generator
  - 3.1. Use current system time to seed the random number generator.
4. **Step 4:** Main Game Loop
  - 4.1. Repeat until the user chooses to quit:
    - 4.1.1. 1: Select Word
  - 4.2. Pick a random word from dictionary.
    - 4.2.1. 2: Scramble Word
  - 4.3. Copy the chosen word into a buffer.
  - 4.4. For each character in the word:
  - 4.5. Generate a random index.
  - 4.6. Swap the current character with the character at the random index.
    - 4.6.1. 3: Display Scrambled Word
  - 4.7. Show the scrambled word to the user.
5. **Step 5:** First Guess
  - 5.1. Prompt user for input.
  - 5.2. Remove newline character from input.
  - 5.3. If input is "exit" or "quit":
    - 5.3.1. Display final score.
    - 5.3.2. End program.
  - 5.4. Else if guess matches original word:
    - 5.4.1. Display "Correct!".
    - 5.4.2. Increment score by 1.
  - 5.5. Else:
    - 5.5.1. Display "Wrong! Try again."
6. **Step 6:** Second Guess (if first was wrong)
  - 6.1. Prompt user again.
  - 6.2. Remove newline character from input.

- 6.3. If input is "exit" or "quit":
  - 6.3.1. Display final score.
  - 6.3.2. End program.
- 6.4. Else if guess matches original word:
  - 6.4.1. Display "Correct!".
  - 6.4.2. Increment score by 1.
  - 6.4.3. Else:
    - 6.4.4. Display "Still wrong. The word was: <original>".
    - 6.4.5. Decrement score by 1 (but not below 0).
- 7. **Step 7:** Update Score
  - 7.1. Display "Current score: <score>".
- 8. **Step 8:** Repeat
  - 8.1. Return to Step 4 for the next round.
- 9. **Step 9:** End
  - 9.1. When user exits, display final score and "Goodbye!".
  - 9.2. Terminate program

### 3. FLOWCHART





#### 4. SOURCE CODE

```
#include <stdio.h>                                // Standard input/output functions (printf, fgets, etc.)

#include <stdlib.h>                                // Standard library (rand, srand, etc.)

#include <string.h>                                // String handling functions (strcmp, strcpy, strlen, strcpyn)

#include <time.h>                                  // Time functions (time for random seed)

#define MAX_WORD_LEN 100                          // Maximum length allowed for a word

#define DICT_COUNT 16                             // Number of words in the dictionary

// Built-in dictionary of words

static const char *dictionary[DICT_COUNT] = {

    "apple", "orange", "banana", "grape", "pear",

    "melon", "peach", "plum", "mango", "berry",

    "cherry", "lemon", "lime", "kiwi", "papaya", "ram"

};

// Function to scramble a word

// Takes the original word and randomly swaps letters to create a scrambled version

void scrambleWord(const char *word, char *scrambled)

{

    strcpy(scrambled, word);                      // Copy original word into scrambled buffer

    int len = strlen(scrambled);                  // Find length of the word

    for (int i = 0; i < len; i++) {                // Loop through each character

        int j = rand() % len;                      // Pick a random position

        char temp = scrambled[i];                  // Swap current character with random one

        scrambled[i] = scrambled[j];

        scrambled[j] = temp;

    }

}
```

```
}  
  
}
```

```
int main(void) {  
  
    srand((unsigned)time(NULL));           // Seed random number generator with current time  
  
    int score = 0; // Initialize score to 0  
  
    printf("Welcome to the Word Scramble Game!\n");  
  
    printf("Type your guess (or 'exit' to quit).\n");  
  
  
    // Main game loop (runs until user quits)  
  
    while (1)  
    {  
  
        // Pick a random word from dictionary  
  
        const char *original = dictionary[rand() % DICT_COUNT];  
  
        char scrambled[MAX_WORD_LEN];  
  
        scrambleWord(original, scrambled);  
  
        // Scramble the chosen word  
  
        // Show scrambled word to user  
  
        printf("\nScrambled word: %s\n", scrambled);  
  
  
  
        // First attempt  
  
        char guess[MAX_WORD_LEN];  
  
        printf("Your guess: ");  
  
        if (!fgets(guess, sizeof(guess), stdin)) break;           // Read user input
```

```

guess[strcspn(guess, "\n")] = '\0'; // Remove newline character

// Check if user wants to quit

if (strcmp(guess, "exit") == 0 || strcmp(guess, "quit") == 0)
{
    printf("Final score: %d\n", score);

    printf("Goodbye!\n");

    break; // Exit the loop
}

// If first guess is correct

if (strcmp(guess, original) == 0)
{
    printf("Correct!\n");

    score++; // Increase score
} else
{
    // Give another chance if first guess is wrong

    printf("Wrong! Try again.\n");

    printf("Your second guess: ");

    if (!fgets(guess, sizeof(guess), stdin)) break;

    guess[strcspn(guess, "\n")] = '\0';

    // Check again if user wants to quit

    if (strcmp(guess, "exit") == 0 || strcmp(guess, "quit") == 0) {

        printf("Final score: %d\n", score);
    }
}

```

```

printf("Goodbye!\n");

break;

}

// If second guess is correct
if (strcmp(guess, original) == 0)
{
printf("Correct!\n");

score++; // Increase score

} else
{

// If both guesses are wrong
printf("Still wrong. The word was: %s\n", original);

if (score > 0) { // Negative marking, but score cannot go below 0
score--;

}

}

}

// Show current score after each round
printf("Current score: %d\n", score);

}

return 0; // End of program

}

```

## 5. RESULTS

- THE IMAGES BELOW ARE THE SCREEESHOTS OF THE OUTPUT -

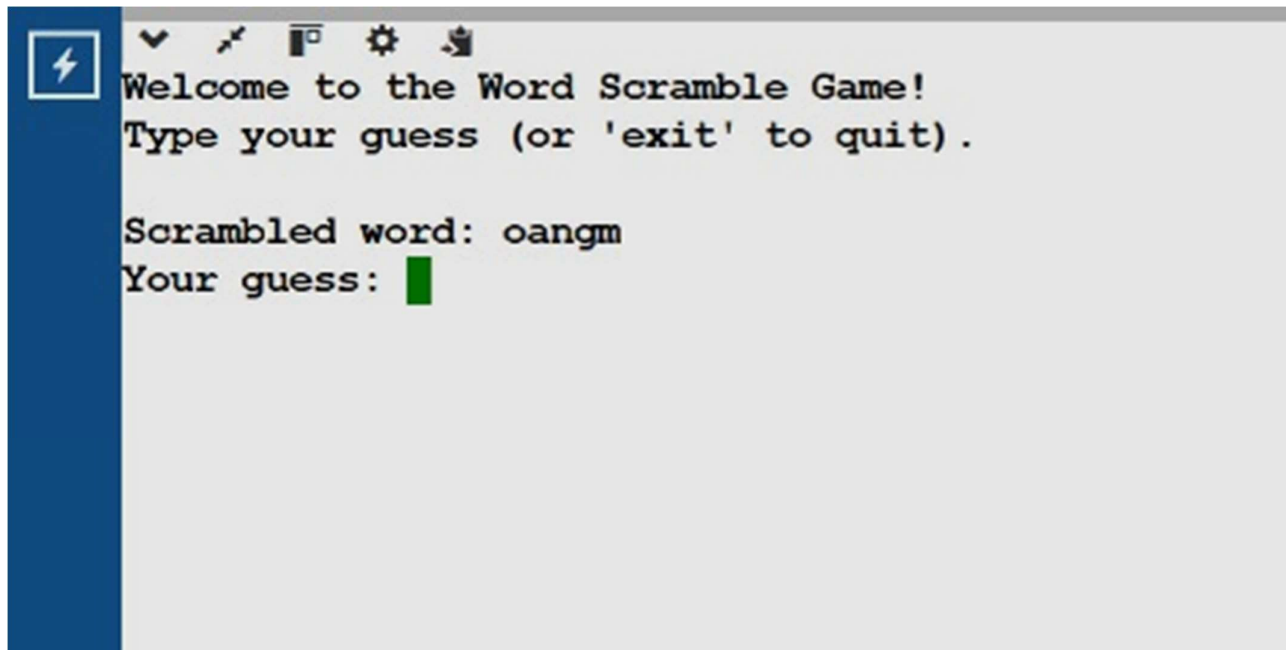


Fig. 5.1. Initial interface showing welcome message and first scramble.

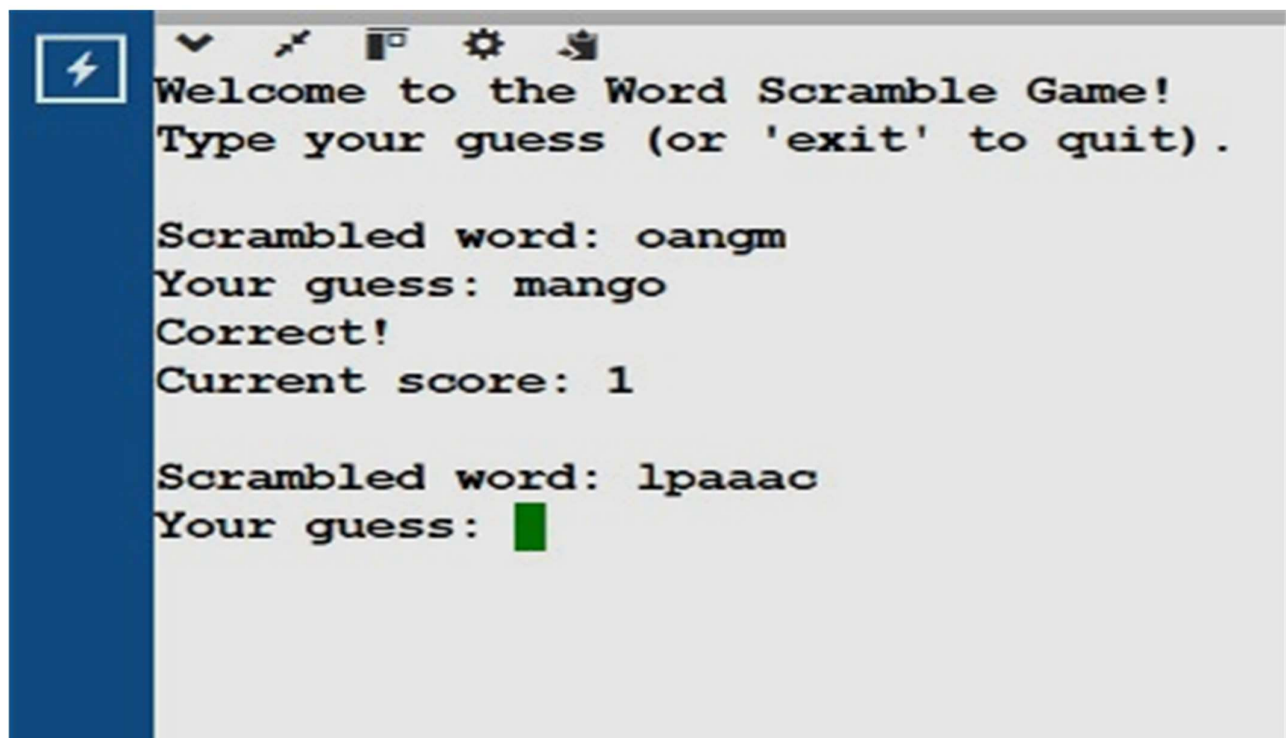


Fig. 5.2. Successful word guess updating the score.

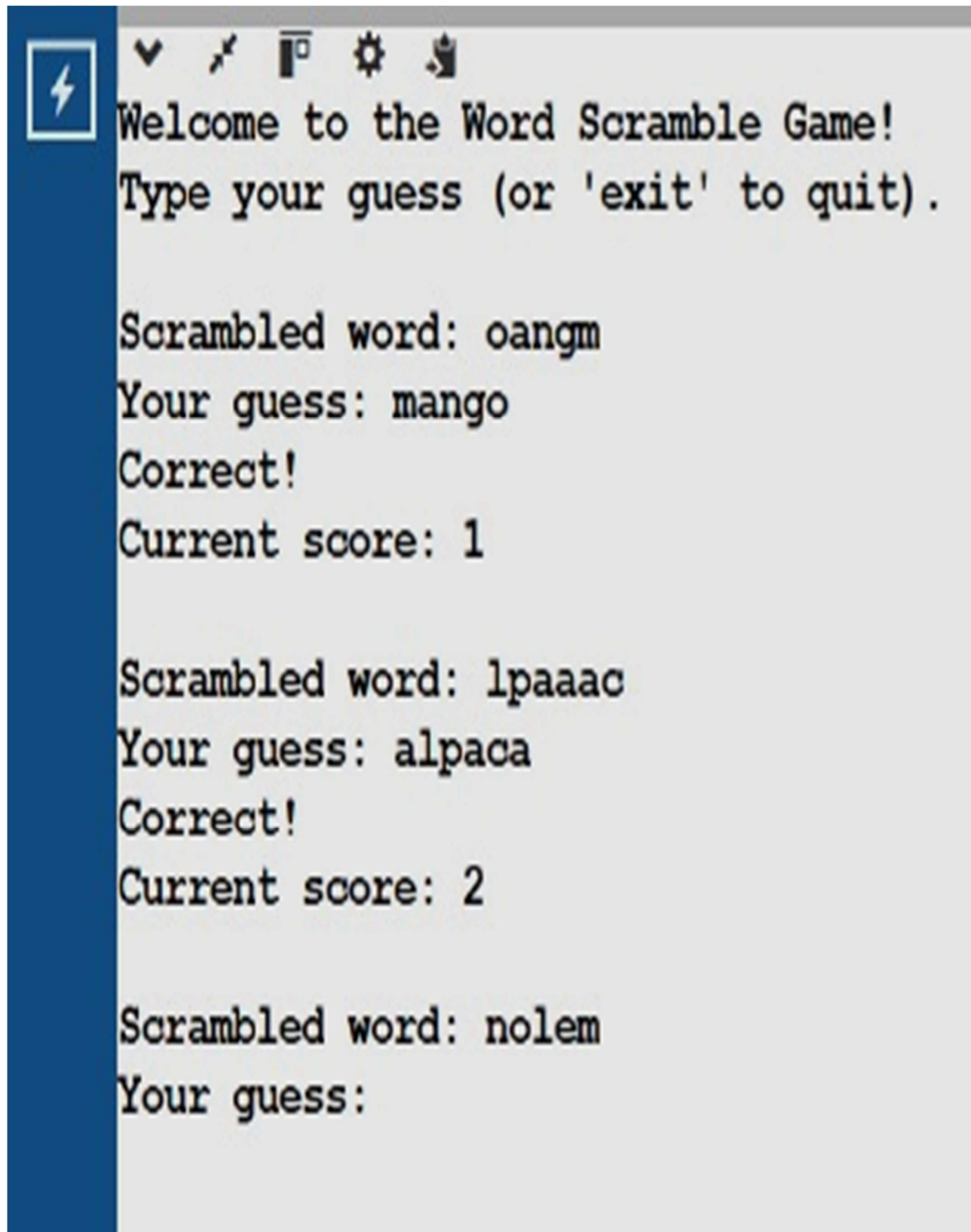


Fig. 5.3. Successful attempt updating the cumulative score

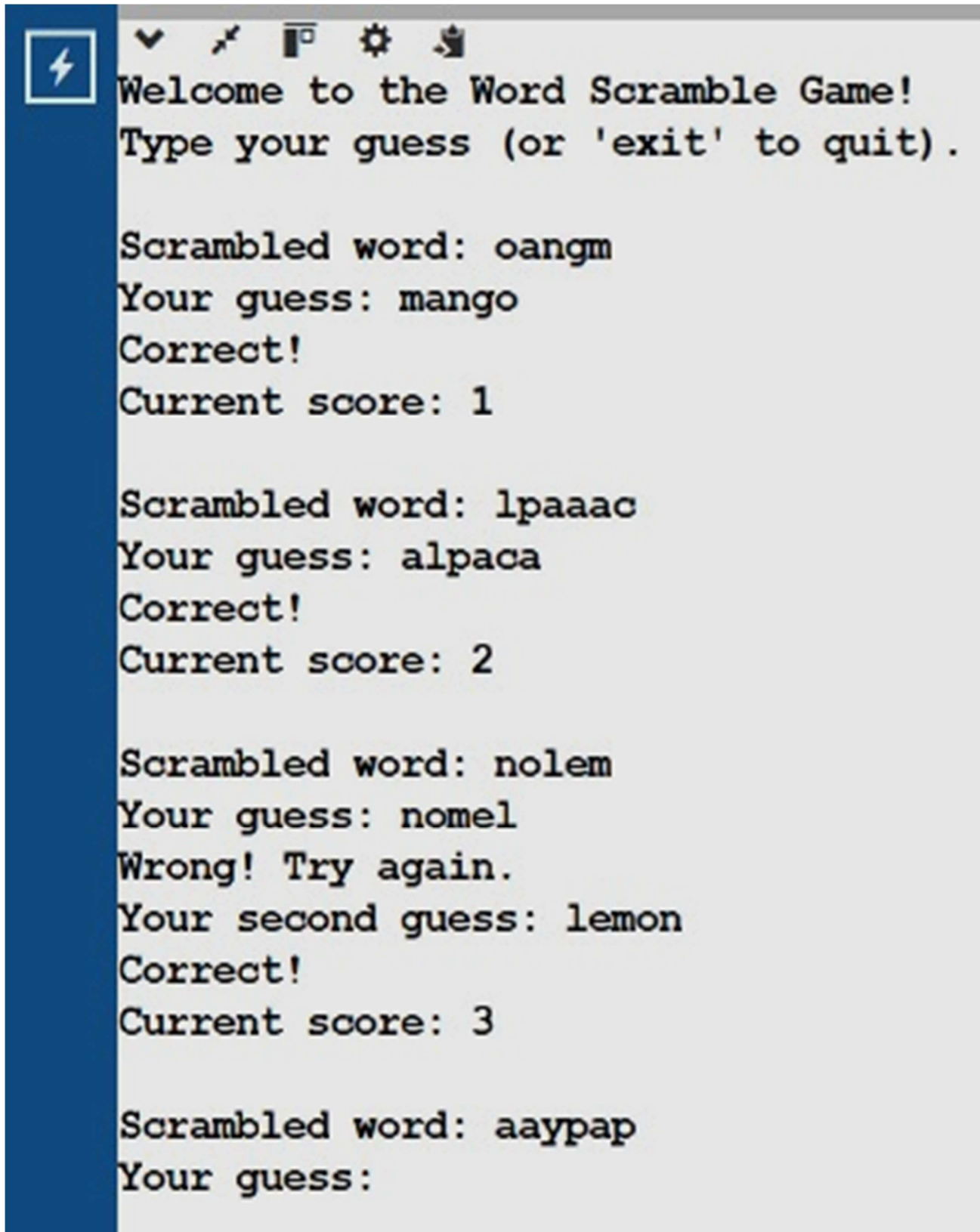


Fig. 5.4. Gameplay log illustrating multiple game rounds

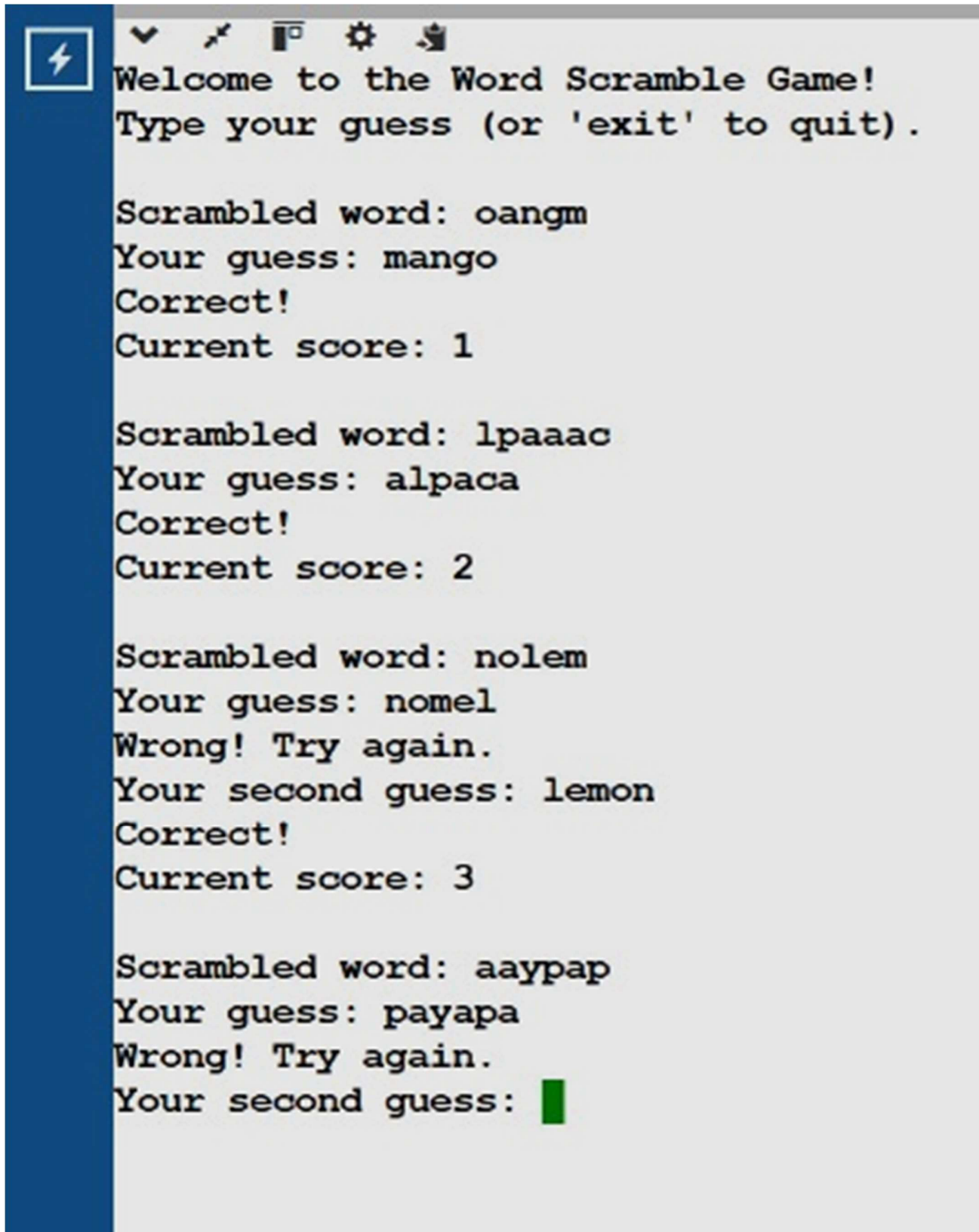


Fig. 5.5. Interface requesting a second attempt.



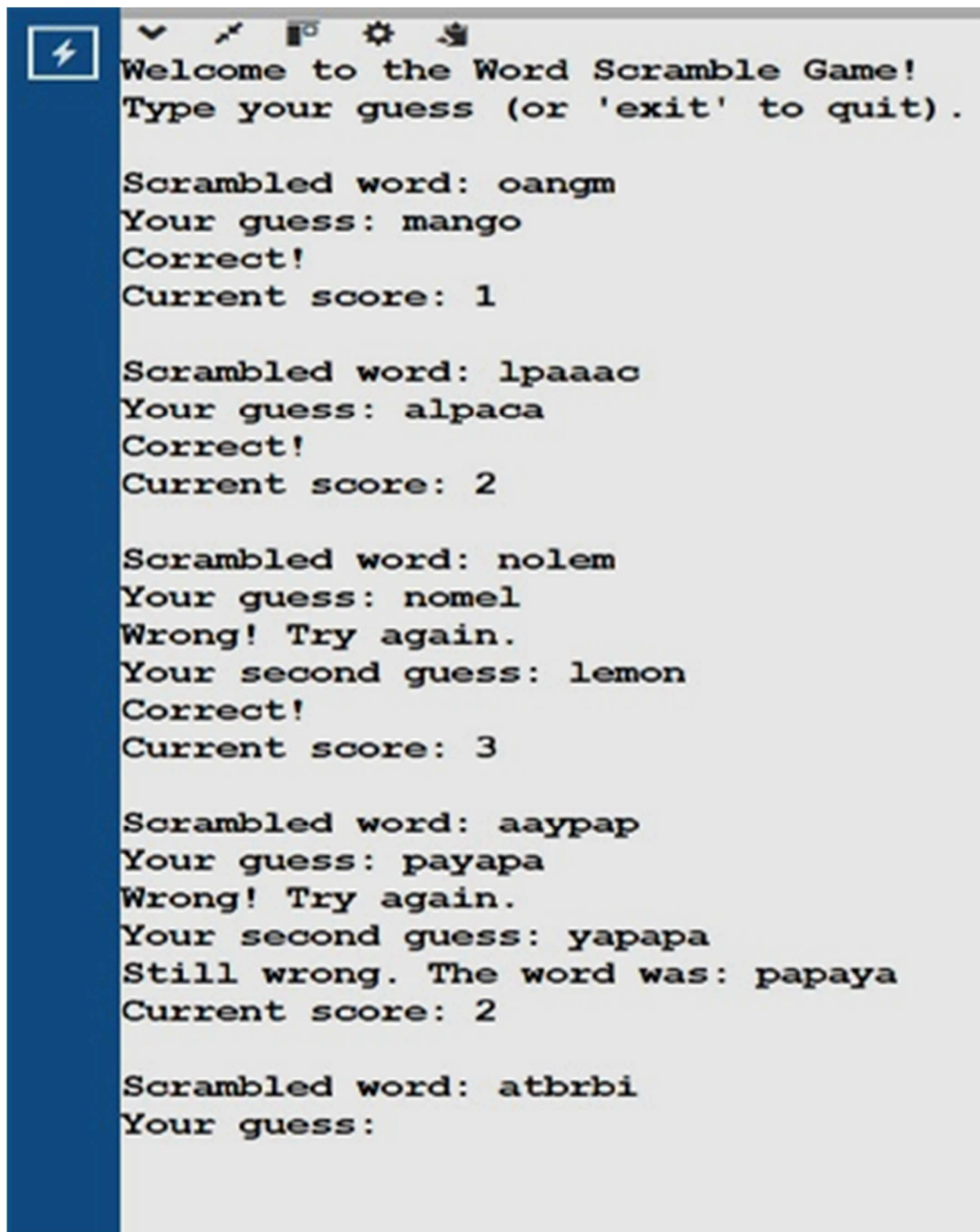
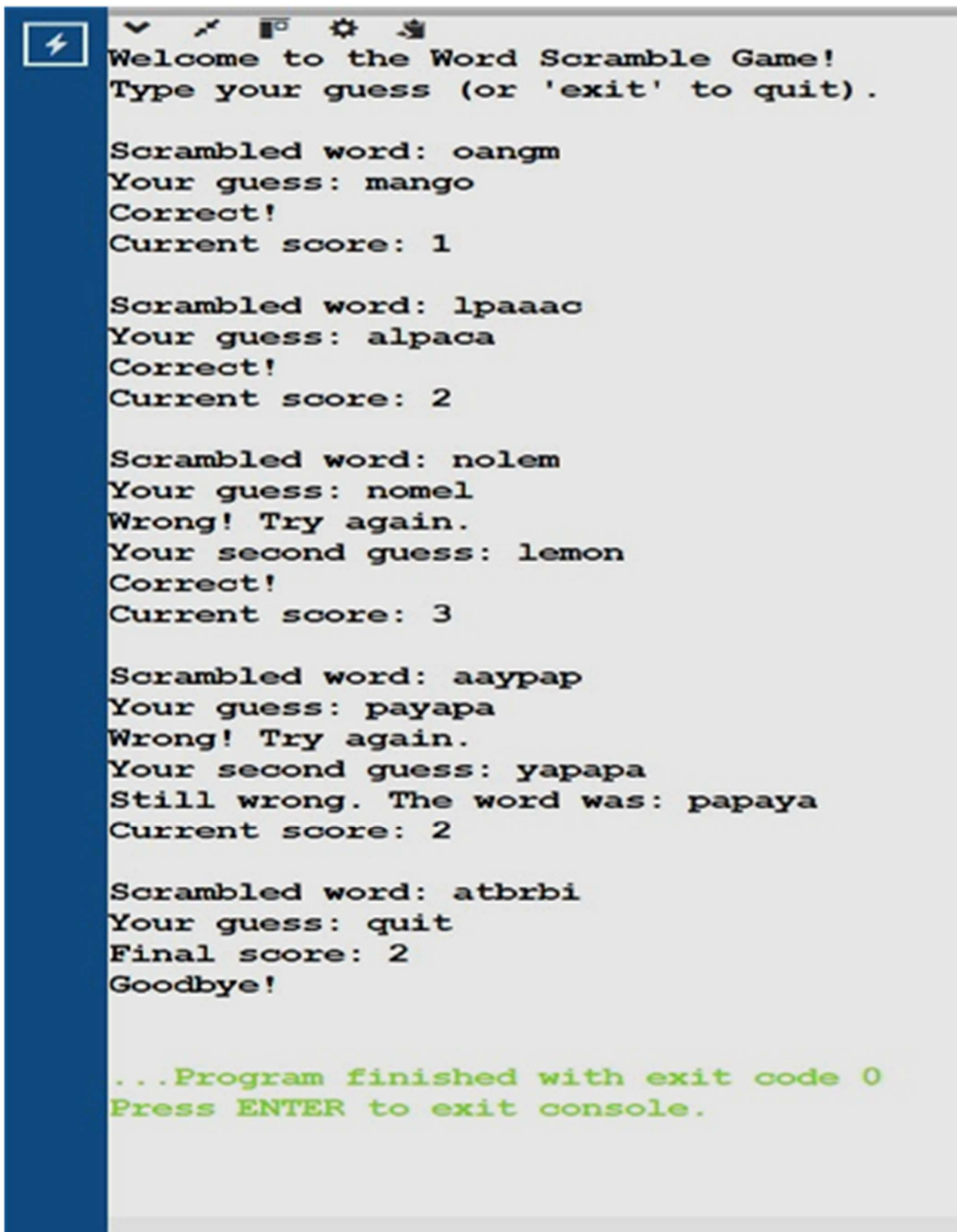


Fig. 5.6. Solution reveal after two failed attempts.

A terminal window with a blue sidebar on the left containing a lightning bolt icon. The terminal displays the text of a word scramble game. The game starts with a welcome message and a prompt to enter a guess. It then proceeds through several rounds of word scrambling and guessing. The first round shows the scrambled word 'oangm' and the guess 'mango', which is correct, resulting in a score of 1. The second round shows 'lpaaac' and the guess 'alpaca', also correct, resulting in a score of 2. The third round shows 'nolem' and the guess 'nomel', which is wrong. The second guess 'lemon' is correct, resulting in a score of 3. The fourth round shows 'aaypap' and the guess 'payapa', which is wrong. The second guess 'yapapa' is also wrong, and the program reveals the word was 'papaya', resulting in a score of 2. The fifth round shows 'atbrbi' and the guess 'quit', which is wrong. The final score is 2, and the program says 'Goodbye!'. At the bottom, green text indicates the program finished with exit code 0 and prompts the user to press ENTER to exit the console.

```
Welcome to the Word Scramble Game!  
Type your guess (or 'exit' to quit).  
  
Scrambled word: oangm  
Your guess: mango  
Correct!  
Current score: 1  
  
Scrambled word: lpaaac  
Your guess: alpaca  
Correct!  
Current score: 2  
  
Scrambled word: nolem  
Your guess: nomel  
Wrong! Try again.  
Your second guess: lemon  
Correct!  
Current score: 3  
  
Scrambled word: aaypap  
Your guess: payapa  
Wrong! Try again.  
Your second guess: yapapa  
Still wrong. The word was: papaya  
Current score: 2  
  
Scrambled word: atbrbi  
Your guess: quit  
Final score: 2  
Goodbye!  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Fig. 5.7. Exit command execution and final score display.

## REFERENCES

[1] H. Schildt, C: The Complete Reference, 4th ed. New York, NY, USA: McGraw-Hill/Osborne, 2000.

[2] E. Balagurusamy, Programming in ANSI C, 8th ed. New Delhi, India: McGraw Hill Education, 2019.

[3] Code With Harry, "C Language Tutorials In Hindi," YouTube. [Online]. Available: [https://youtube.com/playlist?list=PLu0W\\_9lII9aiXlHcLx-mDH1Qul38wD3aR&si=EYHUTs-4nA5kJaRL](https://youtube.com/playlist?list=PLu0W_9lII9aiXlHcLx-mDH1Qul38wD3aR&si=EYHUTs-4nA5kJaRL).

[4] Apna College, "C Language Tutorial for Beginners (with Notes & Practice Questions)," YouTube, Mar. 6, 2022. [Online]. Available: <https://www.youtube.com/watch?v=irqbmMNs2Bo>.

[5] Portfolio Courses, "Word Scramble Game | C Programming Example," YouTube, Oct. 29, 2021. [Online]. Available: <https://youtu.be/loa5eBDONHw>.