

# Scrabble Algorithm

*by* Maytham Ali Kadhim

---

**Submission date:** 25-Jul-2023 12:41AM (UTC+0100)

**Submission ID:** 210643848

**File name:** 43887\_Maytham\_Ali\_Kadhim\_Scrabble\_Algorithm\_503743\_591017617.docx (158.25K)

**Word count:** 1827

**Character count:** 9670



Prifysgol Cymru  
Y Drindod Dewi Sant  
University of Wales  
Trinity Saint David

200  
1822 - 2022

|                                |                                                                                 |
|--------------------------------|---------------------------------------------------------------------------------|
| <sup>1</sup> Assignment Title: | Scrabble Algorithm                                                              |
| Module Name:                   | Analysing and Solving Problems                                                  |
| Module Code:                   | ACCA4019                                                                        |
| Program of Study:              | <sup>1</sup> Certificate of Higher Education Computing Skills for the Workplace |
| Student Name:                  | Maitham Ali Khadim                                                              |
| Student ID #:                  | 2222399                                                                         |
| Lecturer Name:                 | Dr. Abida Perveen                                                               |

## Table of Contents

|                                                                     |    |
|---------------------------------------------------------------------|----|
| <b>Task 1: Rules of Scrabble</b> .....                              | 3  |
| Game Setup:.....                                                    | 3  |
| Word Formation: .....                                               | 3  |
| Tile Placement:.....                                                | 3  |
| Scoring: .....                                                      | 3  |
| Drawing New Tiles:.....                                             | 4  |
| Challenging Words: .....                                            | 4  |
| Game End:.....                                                      | 4  |
| Priority levels: .....                                              | 4  |
| Must: .....                                                         | 4  |
| Game Board and Tiles Display:.....                                  | 4  |
| Word Validation: .....                                              | 4  |
| Turn-Based Gameplay:.....                                           | 5  |
| Should:.....                                                        | 5  |
| AI Opponent: .....                                                  | 5  |
| Multiplayer Support: .....                                          | 5  |
| Scoring and Point Calculation: .....                                | 5  |
| Could:.....                                                         | 5  |
| Customizable Rules: .....                                           | 5  |
| Hint and Suggestion System: .....                                   | 6  |
| Won't:.....                                                         | 6  |
| Language Support:.....                                              | 6  |
| In-Game Chat:.....                                                  | 6  |
| Theme Customization: .....                                          | 6  |
| Implementation Priorities for Three "Must" Options: .....           | 6  |
| <b>Task 2</b> .....                                                 | 7  |
| Variables Used in the Solution:.....                                | 7  |
| Algorithm: Finding the Sum of Two Numbers.....                      | 7  |
| Flowchart Symbols: .....                                            | 8  |
| Sub processes of your solution.....                                 | 9  |
| Sub-process: Input numbers (num1 and num2) .....                    | 9  |
| Sub-process: Calculate the sum of two numbers (num1 and num2) ..... | 9  |
| Combining these two sub-processes with the main algorithm: .....    | 10 |
| <b>References</b> .....                                             | 11 |

## Task <sup>1</sup>1: Rules of Scrabble

### Rules of Scrabble:

Scrabble is a famous word game played on a square board with a grid of 15x15 squares. The game is about making words using letter tiles that have different values, and putting them on the board to get points (Kannanat *et al.* 2016). These are the main rules of Scrabble:

### Game Setup:

- The board is placed in the center of the playing area, and each player draws seven letter tiles from a bag to start the game.
- Players decide who goes first, and that player will have the first turn (Aung *et al.* 2021).

### Word Formation:

- Players take turns forming words on the board, connecting them to existing words to create new ones.
- Words must be formed from left to right (horizontal) or top to bottom (vertically).
- All words played must be valid words from an agreed-upon word list or dictionary.

### Tile Placement:

- Each player places their word on the board in a way that all letters form a continuous line and connect with existing words if present.
- Players can form multiple words in a single turn if the placement allows.

### Scoring:

- Each letter tile has a specific point value, and the score of a word is the sum of the points of all its letters.
- Certain squares on the board multiply the word's score or the individual letter's score.
- Blank tiles can represent any letter but carry no point value (Aung and Iida, 2018).

**Drawing New Tiles:**

- After forming a word, a player replenishes their rack by drawing tiles from the bag to always have seven letters.

**Challenging Words:**

- Players have the right to challenge the validity of a word. If the word is found to be invalid, it is removed from the board, and the player loses their turn.

**Game End:**

- The game continues until all letter tiles are used by the players, or if no further appropriate words can be made (Kananat *et al.* 2016).
- Finally the winner is the one who has made the highest score.

**List of features needed for a Computer-Based Scrabble Solution:****Priority levels:**

- Must (Critical): Features that are essential for the basic functioning of the game.
- Should (Important): Features that significantly enhance the gameplay experience.
- Could (Desirable): Features that add value but are not critical.
- Won't (Optional): Features that are nice to have, but not necessary.

**Must:****Game Board and Tiles Display:**

- The computer-based solution should provide a digital game board with a 15x15 grid and display letter tiles with their point values.
- Players should be able to see their own rack of letter tiles.

**Word Validation:**

- The solution must include a dictionary of valid words to verify the words played by players.

- Invalid words should be rejected, and players should be able to challenge questionable words (Lentz, 1994).

**Turn-Based Gameplay:**

- The solution must support turn-based gameplay, letting players to take their turns alternately.
- Players should be able to place words on the board during their turn and draw new tiles afterward (Sheppard, 2002).

**Should:****AI Opponent:**

- The solution should offer an AI opponent with different difficulty levels.
- AI opponents should be able to form words strategically and present a challenge to players.

**Multiplayer Support:**

- The solution should allow players to play against each other online.
- Players should be able to invite friends or match with random opponents.

**Scoring and Point Calculation:**

- The solution should automatically calculate the score for each word played.
- It should consider bonus points for premium squares on the board.

**Could:****Customizable Rules:**

- The solution could provide options to customize game rules, such as time limits per turn, word list selection, etc. (Aung *et al.* 2021).

**Hint and Suggestion System:**

- It could offer a hint system to assist players in finding potential words on their racks.
- Suggestion system could recommend optimal word placements on the board (Aung and Iida, 2018).

**Won't:****Language Support:**

- While English is the standard language for Scrabble, additional language support could be considered.

**In-Game Chat:**

- An optional chat feature for players to communicate during the game.

**Theme Customization:**

- Customizable themes or skins for the game's visual appearance.

**Implementation Priorities for Three "Must" Options:**

a. Game Board and Tiles Display: Priority: High (9/10) Explanation: Displaying the game board and letter tiles is fundamental to the game. Without this feature, players cannot interact visually with the game and form words on the board. Therefore, it is critical to have a functional and clear display of the game elements.

b. Word Validation: Priority: High (9/10) Explanation: Ensuring word validity is crucial to maintaining fair gameplay. A robust word validation system using an accurate dictionary is essential to prevent players from using invalid words or exploiting the system. This feature is fundamental to maintaining the integrity of the game (Kananat *et al.* 2016)

c. Turn-Based Gameplay: Priority: High (8/10) Explanation: Implementing turn-based gameplay is essential to make sure players take their turns in an orderly fashion. Without this feature, the game loses its structure and fairness, leading to confusion and potential

disputes between players. Properly managing turns is critical for a smooth gaming experience (Lentz, 1994).

## Task 2

### Variables Used in the Solution:

- **Game Board (2D Array):** Represents the game board grid with dimensions 15x15. Each cell in the array stores information about the letter tile placed in that position or is empty.
- **Letter Tiles (Array):** Stores the set of available letter tiles in the bag, including their respective letters and point values.
- **Player Rack (Array):** Represents the letter tiles a player has in their rack. It can hold up to seven tiles.
- **Current Player (Integer):** Stores the index of the current player taking their turn.
- **Dictionary (Set/Array):** Contains a list of valid words that players can use during the game.
- **Score (Array):** Keeps track of each player's score.
- **Game Ended (Boolean):** Indicates whether the game has ended or not.
- **Player score:** An integer variable to keep track of the player's current score.
- **Computer score:** An integer variable to store the computer's current score.
- **Rounds:** An integer variable representing the total number of rounds to be played (a constant value).
- **Choices:** An array or list containing the available choices for the game (e.g., Rock, Paper, Scissors).
- **Player choice:** A string variable to store the player's chosen option for each round.
- **Computer choice:** A string variable to store the computer's randomly chosen option for each round (Sheppard, 2002).

### Algorithm: Finding the <sup>2</sup>Sum of Two Numbers

Step 1: Start

Step 2: Input the first number (num1)

Step 3: Input the second number (num2)

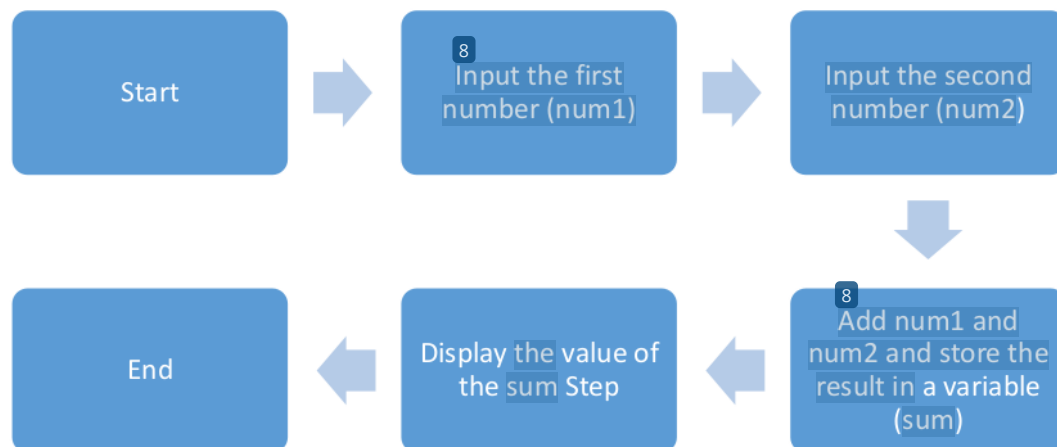
Step 4: Add num1 and num2 and store the result in a variable (sum)

Step 5: Display the value of the sum

Step 6: End (Aung *et al.* 2021)

### Flowchart Symbols:

- Terminal/Start or End: Represents the start or end of the algorithm. It is a rounded rectangle with the word "Start" or "End" inside.
- Input/Output: Represents inputting or outputting data. It is a parallelogram with the label of the data being read or displayed.
- Process: Represents a process or action to be taken. It is a rectangle with the description of the process.
- Connector: Represents a connection between different parts of the flowchart. It is a small, labeled circle.
- Decision: Represents a decision point in the algorithm. It is a diamond shape with a question inside and arrows leading to different paths based on the condition's result (Aung and Iida, 2018).



## Sub processes of your solution

Let's detail two sub-processes of the algorithm for finding the sum of two numbers using pseudocode:

### Sub-process: Input numbers (num1 and num2)

**Pseudo code:**

FUNCTION InputNumbers():

    4    DISPLAY "Enter the first number (num1):"

    INPUT num1

    DISPLAY "Enter the second number (num2):"

    INPUT num2

END FUNCTION

**Description:** This sub-process is responsible for taking input from the user for two numbers, num1 and num2. It displays prompts to the user asking them to enter each number and then reads the input from the user. The input numbers are later used to calculate their sum (Kananat *et al.* 2016).

### Sub-process: Calculate the sum of two numbers (num1 and num2)

6

**Pseudo code:**

FUNCTION CalculateSum(num1, num2):

    sum = num1 + num2

    RETURN sum

END FUNCTION

**Description:** This sub-process takes two numbers, num1 and num2, as input parameters and calculates their sum. The sum is stored in a variable called "sum" using the addition operation. Finally, the calculated sum is returned to the calling function (Lentz, 1994).

### Combining these two sub-processes with the main algorithm:

#### Pseudo code:

FUNCTION InputNumbers():

    DISPLAY "Enter the first number (num1):"

    INPUT num1

    DISPLAY "Enter the second number (num2):"

    INPUT num2

END FUNCTION

FUNCTION CalculateSum(num1, num2):

    sum = num1 + num2

    RETURN sum

END FUNCTION

FUNCTION Main():

    CALL InputNumbers() // Sub-process to input the numbers

    result = CALL CalculateSum(num1, num2) // Sub-process to calculate the sum

    DISPLAY "The sum of ", num1, " and ", num2, " is ", result

END FUNCTION

CALL Main() // Start the execution of the main algorithm

In this way, the algorithm divides the task into sub-processes, making it more modular and easier to manage. The "InputNumbers" sub-process handles the input of the two numbers, and the "CalculateSum" sub-process performs the addition and returns the result, which is then displayed in the main algorithm. This approach improves code reusability and readability (Sheppard, 2002).

## References

- <sup>1</sup> Aung, H.P.P., Khalid, M.N.A. and Iida, H., 2021. What Constitutes Fairness in Games? A Case Study with Scrabble. *Information*, [online] 12(9), p.352.
- <sup>7</sup> Aung, H.P.P., and Iida, H., (2018). 'Advantage of Initiative Revisited: A case study using Scrabble AI'. *2<sup>nd</sup> International Conference on Advanced Information Technologies (ICAIT) Yangon, Myanmar*.
- <sup>3</sup> Kananat, S., Terrillon, J.C., and Iida, H. (2016). 'Gamification and Scrabble. In: Bottino, R., Jeuring, J., Veltkamp, R. (eds) Games and Learning Alliance. GALA 2016'. *Lecture Notes in Computer Science()*, vol 10056. Cham: Springer.
- <sup>9</sup> Lentz, V. A., and Stanford, J. (1994). 'Scrabble® - A Collaboration Training Tool'. *Insight Into System Engineering*, 4(1), pp. 349-356.
- Sheppard, B., (2002). 'World-championship-caliber Scrabble'. *Artificial Intelligence*, 34(2), pp. 241-275.

# Scrabble Algorithm

## ORIGINALITY REPORT

16%

SIMILARITY INDEX

8%

INTERNET SOURCES

4%

PUBLICATIONS

16%

STUDENT PAPERS

## PRIMARY SOURCES

|   |                                                                              |    |
|---|------------------------------------------------------------------------------|----|
| 1 | Submitted to University of Wales, Lampeter<br>Student Paper                  | 6% |
| 2 | kz.co-aol.com<br>Internet Source                                             | 2% |
| 3 | link.springer.com<br>Internet Source                                         | 2% |
| 4 | Submitted to LSU, Alexandria<br>Student Paper                                | 1% |
| 5 | Submitted to GNA University<br>Student Paper                                 | 1% |
| 6 | Submitted to Western International College<br>(WINC London)<br>Student Paper | 1% |
| 7 | hrmars.com<br>Internet Source                                                | 1% |
| 8 | Submitted to Midlands State University<br>Student Paper                      | 1% |
| 9 | Submitted to University of Birmingham<br>Student Paper                       | 1% |

---

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off