

Scrabble report

by Daniel Leone

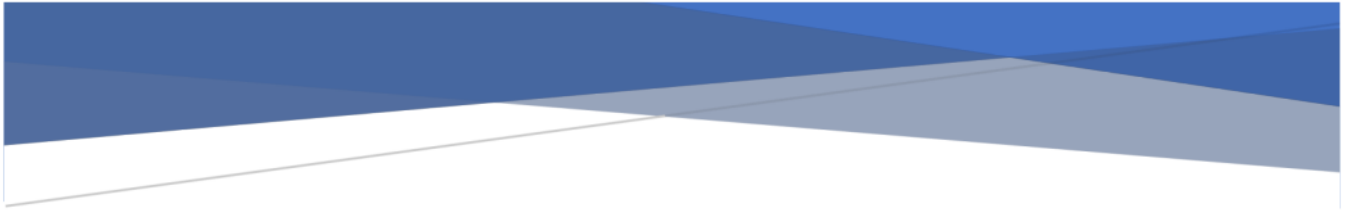
Submission date: 21-Nov-2023 09:49AM (UTC+0000)

Submission ID: 216396038

File name: 49307_Daniel_Leone_Scrabble_report_543034_1672094690.docx (218.76K)

Word count: 1488

Character count: 7762



SCRABBLE ALGORITHM REPORT 21/11/23

Analysing & Solving Problems
ACCA4019

Daniel Leone (2317483)
2317483@student.uwtsd.ac.uk

Table of Contents

1. Introduction.....	2
2. Analysis	2
2.1 Rules and Requirements.....	2
2.2 MoSCoW: Must haves, Should haves, Could haves, Won't/Would haves	3
3. Creating a solution	4
3.1 Variables and constants.....	4
3.2 Algorithm proposal	4
3.2.1 About	4
3.2.2 Flowchart	4
3.2.3 Pseudocode	6
4.Summary.....	7
5. References	8

1. Introduction

The aim of this report is to propose and present an abstract solution which could be implemented for creating an application of the game 'Scrabble'. The author will analyse and document the boardgame, deconstructing its' key elements and rules of gameplay. Through this analysis, an implementable solution will then be conceptualised, and presented to readers, through the use of a flowchart and pseudocode.

2. Analysis

2.1 Rules and Requirements

As described in the official Hasbro Scrabble game guide [1], Scrabble is a two to four player board game which requires players to think of, and form words, utilizing a random set of tiles which are distributed at the start, and also throughout the game. Each player will be dealt seven random lettered tiles from 'a' to 'z', and as they play, their tiles will be replenished back to a total of seven after each turn, until there are no tiles left to be dealt. Some tiles are blank and do not have a set letter on them, these blank tiles will allow the player to choose and play it as any letter they see fit. The objective of the game is to gain as many points as possible; the player who finishes with the most points, once all tiles are used up, will be the winner.

The first word placed, by the first player, must cover the center square. Words must be placed in a crossword style on the board, always coordinated from up to down or left to right, never diagonally. Each tile used within the words are worth a different number of points, with higher difficulty words such as those containing the letter 'z' would gain the player substantially more points in comparison to a more basic and shorter word. As an example of this, the word 'zebra' would gain a player more points than playing the word 'cat'.

Other rules include; Apart from the first turn, all words must join onto existing tiles which have already been placed on the board. All touching tiles on the board must create full and real words which would appear in a dictionary. Words that have already been placed on the board can be played again; there is no limit to how many times a single word can be played. Each turn will be scored based on the word played and also any existing words that may have been adapted or modified because of the play, or, any additional words that have been created alongside of the played word. Although words can be adapted, Existing tiles on the board can never be moved or replaced once they have been played.

2.2 MoSCoW: Must haves, Should haves, Could haves, Won't/Would haves

There are many nuances to the game of scrabble which aren't necessarily vital to the core gameplay. Through deconstructing and categorizing elements into a 'MoSCoW' table, we are able to identify and prioritise the key functionalities when considering a solution.

Must Have

- Ability to have at least 2 players
- A set number of total tiles which can be dealt from the bag
- A limit of 7 tiles held per player
- Set points awarded for each different letter
- A gameboard with defined positions where tiles can be set
- A scorekeeper

Should Have

- First word to be placed covering center tile
- Ability to skip a turn
- Building off existing played words
- First letter must be at least two letters long

Could Have

- Blank tiles which can be played as any letter
- Premium tile squares on the board
- 'Bingo' for an additional 50 points on top of the score of the word (using all seven tiles)

Won't Have / Would Have

- Alternative game modes such as 'Finish Line Scrabble'
- Ability to shuffle players tiles
- 3-4 players

3. Creating a solution

3.1 Variables and constants

Identifying elements known as 'variables' and 'constants' allows us to better understand specific value data that a computer program will need to refer to, throughout its operation. As claimed on a web page by Kenneth Leroy Busbee and Dave Braunschweig [2] A Variables' value can change during a program's execution, whereas a Constants' value, cannot change.

The identified variables for a game of Scrabble are as followed

- CurrentPlayerTiles - Tiles on Player(s) rack
- PlayedWord - Played words on board
- PlayedScore - Score for played word
- RemainingBagTiles - Number of tiles remaining in bag
- CurrentPlayerScore - Player(s) total score (separate value linked to Player1 & Player2)

The identified Constants for a game of Scrabble are as followed

- Dictionary - Playable/Allowed words
- Points - Points value per letter tile & word

3.2 Algorithm proposal

3.2.1 About

In order to propose a solution, a clear and easy to understand method in which to communicate the abstract ideas needs to be used. This will be achieved through the use of a flowchart and pseudocode. As described in an article written by Sagarika Ghosh, Sanjoy Pratihar, Sanjay Chatterji & Anupam Basu [3], For computer novices, it can be challenging to understand algorithms, therefore it is important to illustrate and write in a way that facilitates a thorough understanding.

3.2.2 Flowchart

A flowchart has been created to visually present the created solution. The flowchart consists of a main process, and three sub-processes. The sub-processes within the flowchart for the proposed solution are as follows:

1. TileDistribution

This sub-process is implemented to check and distribute the tiles to each player at the start of the turn loop.

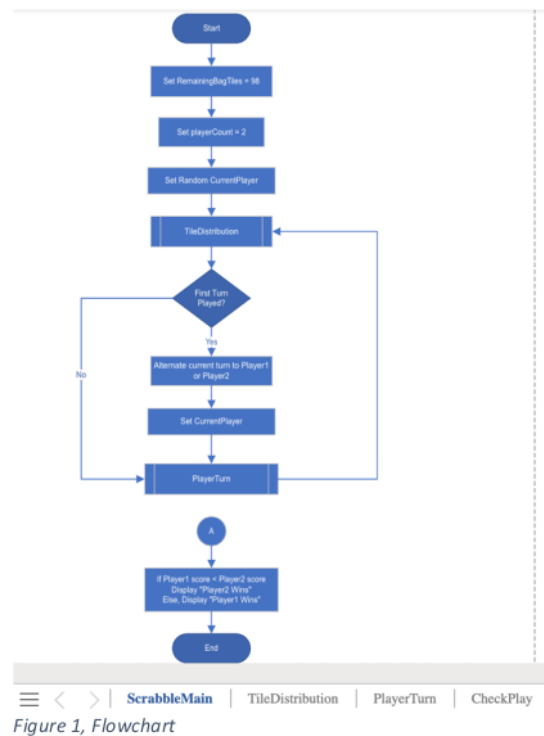
2. PlayerTurn

This sub-process gives players the option to either play or skip their turn. If the player chooses to play a word, it first checks whether the played word is acceptable, and then moves to the next sub process (CheckPlay)

3. CheckPlay

This sub-process scores the current turn played, adds points, and then also checks whether or not to move towards ending the game based on whether 2 specified conditions have been met.

An Image of the flowcharts' main process, as well as tabs for the sub-processes can be seen in Figure 1, below.



3.2.3 Pseudocode

The pseudocode below represents the programmed blocks that a real application might follow based on the created solution.

Main

- Set RemainingBagTiles = 100
- Set PlayerCount = 2
- Set CurrentPlayer (random), Player1 or Player2
- Go to TileDistribution Sub-process
- Check, First Turn Played? (True /False)
 - If False, Go to PlayerTurn Sub-process
- Alternate CurrentPlayer to Player1 or Player2
- Set CurrentPlayer
- Go to PlayerTurn Sub-process
- [Looped] Go to TileDistribution Sub-process

- **A (End Game)**
- If Player1 score < Player2 score Display "Player2 Wins"
 - Else, Display "Player1 Wins"

TileDistribution

- Check RemainingBagTiles = 0? (True/False)
 - If True, Return to main process
- Check CurrentPlayer tiles =<7 (True/ False)
 - If True, Return to main Process
- Distribute tiles from ReamaingBagTiles into CurrentPlayerTiles until CurrentPlayerTiles = 7, or, until RemainingBagTiles = 0
- Set RemainingBagTiles = RemaningBagTiles – DistributedTiles
- Return to Main Process

PlayerTurn

- Input: input word, Or, Skip Turn
 - If input = Skip Turn, Return to Main process
- Check Acceptable, Real Word (True/False)
 - If False, Display "Try Again"

- [Looped] Start PlayerTurn_Sub-process
- Go to CheckPlay Sub-process
- Return to Main Process

CheckPlay

- Calculate Points of played word
- Set word to table
- Add Points to CurrentPlayer Score
- CurrentPlayerTiles = 0? AND RemainingBagTiles = 0? (True/False)
 - If True, Go **to A (End Game)** in main process
 - If False, Return to PlayerTurn_Sub-process

4.Summary

Through analysing the game of Scrabble, it was discovered that some elements and features were not crucial to the core gameplay, therefore the created solution foregoes some aspects of the games' features. If required, these missing features can simply be integrated at a later date. This is common practice in development, especially where deadlines need to be met. The use a 'MoSCoW' table enabled the author to narrow down which aspects were important for the initial solution, and which were not so important. A flow chart was used to visually demonstrate the movement, and the direction of the steps that a real application would follow. Pseudocode has been written to represent what those processes might look like in a real algorithm. In conclusion, this report should provide readers with a clear presentation of how the authors' abstract scrabble solution was created, and also demonstrate how these abstract concepts can be clearly communicated through the use of Flowcharts, and pseudocode.

5. References

- [1] Hasbro. [Online]. Available:
] <https://www.hasbro.com/common/documents/dad2884b1c4311ddb0b0800200c9a66/0606acdbbc9d4b55bfd2df956f61a723.pdf>.
- [2] K. L. B. a. D. Braunschweig, "PROGRAMMING FUNDAMENTALS," 2018. [Online]. Available:
] <https://press.rebus.community/programmingfundamentals/chapter/constants-and-variables/>.
[Accessed 19 11 2023].
- [3] S. P. S. C. & A. B. Sagarika Ghosh, "Matching of hand-drawn flowchart, pseudocode, and english
] description using transfer learning," *Multimedia Tools and Applications*, vol. 82, no. 17, p. 27027–27055, 1 2 2023.

Scrabble report

GRADEMARK REPORT

FINAL GRADE

80/100

GENERAL COMMENTS

There is a clear fulfilment of brief with clear application of subject knowledge applied. Logical decisions have been made regarding prioritisation.

Flowcharts and document provided are laid out professionally and presented well.

More relevant imagery could have elevated overall document giving further context to some areas.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9