Dawson College

Scrabble Game

Complete, slight program drawback: AI plays first word it finds instead of finding the best word possible, but overall works properly.

Serban Alin Caia

Daniel Simon Chin

Asli Zeybek

Data Communications & Networking

430-518-DW Section 00001

Carlton Davis

November 27, 2020

Program Narrative

\*Important Note when playing the game: When you successfully make a play, pass or swap, the AI will go right after. If you feel like it is taking a while, please wait for up to 15 seconds. A popup box will appear telling you that the AI could not find any words. Do not repeatedly click the play, swap, clear or pass button if the results for the play you are trying to make does not give instant feedback \*

To run the game, the server must be run using the run button for the server (scrabble\_server). Once the server is run and listening for socket connections, we run the client to which will open the SetupSession window which asks the user for server and port imports. You cannot proceed to the GameMenu unless two valid inputs are given. The GameMenu allows the user to select if the AI starts, continue to the Game, go back to the SetupSession or Exit the application. The board interface/game is opened if the user’s inputs can create a connection to the server. It is important that whenever you are closing the window for the client/interface, to also close the server project which is running since both boards on the server and client mirror each other. If the user selected the checkbox for the AI to start, the board will open with the first word already placed, if possible. The player can drag and drop from his/her rack anywhere on the board that does not already contain a letter. If a player tries to play without any letters placed on the board, a message prompt will alert the user. This will repeat until the user plays a valid word, passes their turn, or swaps. The clear button removes all placed letters that were not part of a play and returns it to the player rack. The pass button will make the AI make the first move if no words have been played or make a subsequent move if the board has words on it. The swap button brings up a window with checkboxes to select which letters are to be swapped from the rack. If no checkboxes are selected, the program will alert the user that nothing was swapped (the turn still belongs to the player since nothing was swapped). It is important to note that the player should not press buttons repeatedly when the AI is searching for a valid word to place. The AI is limited to 15 seconds in which it can find a valid word. Once the time is up, the program will either display the word and points made by the AI or a dialog box indicating that the AI did not find any valid words (The AI’s rack is swapped completely when nothing can be found). The text area on the top right represents all the words that were played along with its description(s) and hyponyms(s) if applicable. The scores of the interface are refreshed whenever a good play is made along with the number of tiles remaining on the board. The server project contains the same data structures such as the board, letter, tiles. All the word validation using the dictionary is done on the server. The client’s board mimics the server’s board. Once the game ends, the rack and buttons are disabled so the human player cannot do anything. A dialog message displays the winner with the points of both the player and AI.

**Multithreading (Part 2):**

The program now includes multithreading using a thread-per-client. Every thread/client game is independent from the other. Simply run the server, and then run as many clients as desired. A human player will be restricted from placing a letter from a board onto another client’s board.

The git project contains two sub-projects inside of it (1 for client, 1 for server). Ensure that both are being opened in the IDE. Run the server project, followed by the client project. The server’s console will display the IP Address at which it is running once the Play Game button is clicked and the board is displayed. The server is “localhost” or 0.0.0.0 and the server port is 50,000.

Scrabble Game before multithreading: <https://gitlab.com/AlinCaia/scrabbleproject.git>

Scrabble Game with multithreading: <https://gitlab.com/DanielSimonChin/scrabbleproject.git>