Grace Frizzell

Prof. Weiss

CMSCI 442 A

25 April 2024

Wordle 3 Writeup

This iteration of the project is written in Ruby. To run it, one must run the program and then enter a series of five letter words followed by the five integers indicating their degree of accuracy (i.e. arise01100). For more than one guess, guesses must be separated by a comma and a space (i.e. arise01100, prion01111). After this, a list of subsequent potential guesses based on the information input should be printed. The program almost exclusively uses lists to store guesses, characters, and indices as they are easy to use in Ruby. Similar to the first iteration of the project, I used a hash to identify the intersection of all guesses in order to find the words which agreed with every guess. The algorithm works via analysis of zeros, ones, and two-associated characters and their positions in order to then compare those characters and positions in the file of guess words. Additionally, I fixed the zero comparison to check if there was a double inclusion of the same character, as this was an issue in the first project. Then, each acceptable word for each guess is added to a list, then this list is iterated through to add words as keys and increment a value by one each time it is discovered in the list. Then, the keys with values equivalent to the size of the guess list are printed out as potential guesses. In a sense, Ruby was the easiest language to read and write in so far due to the fact that lists could hold multiple types, and so storing the character and its index were relatively easy. The only thing that made Ruby difficult to write in was the strict naming conventions for the variables as I continued to get error messages if I used camel-case to name variables. However, barring this error, this was the easiest implementation of the project.