Alex Maass

Am838

CS 2024: Final Project Report

For my final project I am building a small word unscrambling game called Unscramble. The core concept of Unscramble is that the program will present the user with scrambled words and the user has to unscramble them. The dictionary that the words are drawn from is the same dictionary that is used in the popular Words with Friends mobile game.

To start, the game needs to read in all of the words from the dictionary. I originally tried using fseek and reading words in a fixed sized manner, but unfortunately I could never get it to work properly. (The failed attempt is also included as a separate folder in the final project archive)Therefore, I reverted back to simply reading through the entire document at start up and placing all of the words into a static vector variable. While this has a high start-up cost of around 25 seconds to load the file, it allows me to do extremely quick look ups. Once the file has been parsed and the dictionary has been created, a menu is presented to the user. There are 5 options presented in the menu: start game, change settings, view high scores, view instructions, and quit. Input is done using the cin operator and is then passed through a stringstream to return an int when so desired. When the user inputs the numeric option for view instructions, the instructions will print out in the terminal. When the user views the high scores, the high scores will be printed out from the current session. High scores are also stored in a vector and the vector is sorted using the sort algorithm provided in C++. To keep track of scores and their respective players, I created a small Score class that contains the 3 letter initials of the current player and their score. Then, when sort is called to sort the high scores vector, I have a small comparator function that looks specifically at the score values inside the Score class to rank them from highest to lowest. Therefore, when the high scores are printed out for viewing, the top 10 scores are printed out in sequential order. Quit is self-evident in what it does: it quits the game.

The actual game play is done by randomly looking up a word from the dictionary vector, scrambling the word and presenting the word to the user to unscramble. All 3 of these tasks are handled by 3 different methods, with scrambling and retrieving words being separate methods and the presenting and input handling being done in the main method. When the user begins a round, the total number of correctly guess words is reset, and the first scrambled word is presented. The user then has to type in their best guess for the word and if it is not correct, the user will be prompted for the word again. If at any point the user gets frustrated and wants to skip the word, typing “giveup” will force the game to present a new word. The objective of the game is to try and unscramble as many words as possible in the time limit. Once the time limit has been reached, the game will end and will prompt the user for their initials to place in the high scores vector.

There are also a few settings that the user can change. These can be accessed by the change settings option in the main menu. In the change settings options, the user can select 1 of 4 things. The user can return to the main menu, change the maximum word length, change the time limit, or reset the high scores. To change the word length, an integer number of letters for the max word length must be typed in. This number must be greater than 2 otherwise no word can be displayed. The user can also change the time limit to allow them more time to play each round in the game. Finally, selecting the reset high scores option will wipe all of the current high scores.