COMP.1020-Computing II - Evil Hangman Final Project Overview

Read from the dictionary file and create the vector of vectors of my_string objects. The index of the vector corresponds to the length of the words. For example, the vector at index 8 holds all of the words of length 8.

Ask the user which word length they want to play with. Make sure to implement input validation. Only certain integers are valid and you need to account for all possible inputs from the user. Also account for the fact that certain word lengths are invalid. For example, there will be no words stored at index 0 since there are no words of length 0. That is the obvious one and there are a few others. Think about how you can figure out which word lengths don't have any words.

After getting the desired word length you can start the game. In each round, you keep track of a master word list and master word list key. The master word list is all possible words left and the master word list key is the key of all of those words. For example, if the user selected a word length of 8 then on round 1 the master word list would be all words of length 8 and the master word list key would be 8 dashes like - - - - - - - -.

At the beginning of each round get the user's guess. Once again, make sure to implement input validation. Only certain characters each round are valid so once again you need to account for all possible inputs from the user. Then, create the AVL tree. Each node in the tree will hold all of the words corresponding to a particular key. For example, if the user's guess were "b" then there would be a node with a key of b - - - b - - - for words like baseball and there would be a node with a key of b - - - - - - - for words like basement. After the tree is created, the node in the tree with the most amount of words in it is found, and the words and key in that node are used as the master word list and master word list key for the next round. This is how the game cheats. Since the node with the most amount of words in it is always chosen, the user is never going to be able to win unless they select an unfairly high amount of guesses. Let's say the node with the key of b - - - b - - - has the most amount of words in it. That means for the next round the master word list key would be b - - - b - - - and the master word list would be all words corresponding to that key like baseball. Repeat this process until the user runs out of guesses or they win. For example, if on the second round the user guessed "a" then the tree would be populated with nodes that have keys like ba - - ba - - and ba - - b - - -. The word baseball would obviously be in the node with the key of ba - - ba - -.

When the game is over, ask the user if they want to play once and handle their selection accordingly. Once again, make sure to do input validation.