Department of Computer Science
Northern Kentucky University
NKU Summer Programming
Workshop 2015

Project 9: Text Twist

You have probably played TextTwist or TextTwist2 online. In this game, you are given 6 (or 7) letters and you are to create as many words out of the letters as possible. For instance, if the letters are 'a', 'e', 'm', 'r', 's', 't', some of the words you can create are "are", "era", "ram", "same", "meat", "team", "teams", "steam", "master" and "stream". TextTwist is a graphical game where you either type letters or click on the letters to move them. Here, we will create a text-only version which has other limitations. A skeleton of the game is given on the website. Start from the skeleton by filling in the various methods. If you are successful, then you can work on enhancements.

The game requires an English dictionary. Your program will load this into an array (the dictionary consists only of 3, 4, 5 and 6-letter words with no repeated letters, for instance you will find "stream" but not "street" in this dictionary). Code to input the dictionary from text file into the array is given to you. With the dictionary, you will need to write a method to select a random 6-letter word (the method is selectWord), display the letters (getLetters, displayLetters) and then input a guess from the user. With each guess, you will need to test to see if the word is legal (isLegal), has not been guessed yet (notTried) and is a dictionary word (trueWord). If all of these are true, add the user's guess to the tried list and add the word's value to the user's score. The user will quit by entering "quit" as their word guess.

You will need several arrays:

- String[] words – all of the words in the dictionary
- String[] tried – initially an empty list, but for every word that the user guesses correctly, add it to this list so that the user cannot get credit for trying the same word twice
- char[] letters – the 6 letters that make up the word, this will be presented to the user before each guess and shown in sorted order (a sort method is given to you)

You will also need two String variables, the selected word (word) and the user's current guess (guess).

The methods are as follows:

1. loadWords – given to you
2. selectWord – get the random 6-letter word from the dictionary
3. getLetters – obtain the letters of the 6-letter word and sort them
4. printLetters – output the 6 letters for the user to see
5. isLegal – test to make sure the word contains 3-6 letters
6. notTried – the user's guess is not part of the tried list
7. trueWord – the user's guess is in the dictionary
8. getWorth – return the value of this word to add to the user's score. The scoring works as follows: 3-letter word = 3 points, 4-letter word = 5 points, 5-letter word = 9 points, 6-letter word = 12 points.

Enhancements:

1. add to isLegal code to make sure that the letters that make up the entered word are actually letters in the selected word (for instance, if the word is "stream", we want to prevent the user from entering the word "dumb" since "dumb" is in the dictionary and would give the user points, but the letters in "dumb" are not part of the letters in "stream").
2. keep track of whether the user was able to guess a 6-letter word correctly and if so, the user can optionally play another round. Add an outer while loop to repeat the main portion of the game if the user gets a 6-letter word and does not want to exit. The word "quit" will be used to exit the inner loop (quit playing the current 6-letter word) and "exit" to exit the game entirely.
3. expand the scoring to use Scrabble-like values for the letters instead of the points shown above in #8.