Leah Kazenmayer

Professor Li

CSC 230

24 February 2020

Pseudocode Project 1

algorithm main **is**

Input:

- Int Argc, which is the number of arguments from command lines
- Char *argv[], which is the char array which holds the arguments

Output:

- A two-d vector matrix will be formed based off the matrix's dimensions and the contents of the word search that follow in the .txt file
- An identical copy of that two-d vector will be created that'll hold the highlights of the found matches; it'll hold boolean values rather than letters

- Check method will be called to check each index in the matrix in all directions to see if each word from the argv[] will be found or not
- The print matrix method will then be called to print the matrix highlighted in the terminal

return 0

algorithm check **is**

Input:

- Matrix (with the characters)
- Matrix (with the booleans)
- argc
- *argv[]

Output:

- for each word in argv from command line do

  for each row do

  for every column do

  Check for that specific word in every direction at that specific index (forward, backward, downward, upward, diagonal down right, diagonal down left, diagonal up right, diagonal up left)

algorithm printHighlightedMatrix **is**

Input:

- Matrix (with the characters)
- Matrix (with the booleans)

Output:

- Link code from colormod.h

  for each row do

for each column do

If the element at that specific index in the boolean matrix is false

Print out char at the same index of matrix the default color

Else

Print out char at the same index of matrix red

There are 8 Separate Checks: forward, backward, downward, upward, diagonal right, diagonal left, diagonal up right, diagonal up left. Each one is a method.

All will have same format as below:

algorithm oneDirectionCheck **is**

Input:

- Matrix (with the characters)
- Matrix (with the booleans)
- Int j (j represents the row index)
- Int k (k represents the column index)
- string currentElement (This string holds the specific word that's going to be checked in every direction)

Output:

- If the word at that index goes outside the matrix in that specific direction

return;

Else

Starting from that index and looping based on the length of the word being searched, append each char/letter to a string called wordFound

If wordFound equals currentElement

All letters that form the found word will be “highlighted” (their boolean values will go from false to true, hence highlighting)