Design Program of Project 1: Text Analysis

jonathan Lam kam cheung – S#40945909

2018

|  |  |
| --- | --- |
| **Names** | **Pseudo Code** |
| 1. Imports | 1. Import tkinter module 2. Import File Dialog library 3. Import ttk library 4. Import Message Box library 5. Import re libarary |
| Class myApp | |
| 1. Init function | 1. Initiate and position 3 frames (A, B, C) 2. Initiate and position an entry box in frame A 3. Initiate and position 3 buttons (Browse, Run and Reset All) in frame A 4. Initiate and position a progress bar in frame B 5. Initiate and position a list box in frame B 6. Initiate and position a textbox in frame C 7. Initiate and position the File Menu tab (Exit, Export and About) 8. Initiate a sub menu in Export to be named Text   ***All positions are made with the use of grids (rows and columns)*** |
| 1. Browse file path function | 1. Clear entry box for old input/file path 2. Enable entry box to allow new file path to be inserted 3. Open dialog window for user to select file 4. Disable entry box to not allow user to modify the new file path 5. Enable Run Analysis button |
| 1. Manual Enter file path function | 1. If Ticked then  * Enable entry box for user input * Disable browse file path button * Enable Run Analysis button  1. Else  * Disable entry box * Enable browse file path button * Disable Run Analysis button |
| 1. Run Analysis function | 1. Check if entry box is not empty 2. Check if the file format is text only 3. Check if file path exists 4. Set progress bar to 0 5. Prompt user to confirm analysis 6. If Yes then  * Calculate the no. of lines in text file to periodically update progress bar * Set char to 0 * Set word to 0 * Set blank space to 0 * Set dictionary to empty * For each line in the text file: * Remove useless symbols such as [,\".!?] which creates a new word * Remove useless ending character that changes the line and adds up to the character count * Calculate the percentage of current line against remaining lines to get progress bar value increase * Update status of progress bar after each line according to calculated percentage increase * Log a line in the list box indicating that the program is processing * Count the number of characters * If char then add +1 to total of characters * Split each line to get only the number of words * If word then add +1 to total of words * For each character, check to see if the character is a space * If space then add +1 to total of blanks pace * For each word * Check if word is stored in dictionary * If stored then add +1 to the value * Else create the new word in dictionary and set the value to 1  1. Else Return 2. Set progress bar to 0 3. Display the result in the textbox of frame C 4. Log a line in the list box indicating that the program process is complete |
| 1. Display Result function | 1. Check if word dictionary is empty  * If not empty then * For each item: Print word and count  1. Calculate percentage blank space count = (blank space count/char count) \*100 2. Print percentage blank space count 3. Print char count 4. Print words count 5. Print blank space count 6. Calculate non-blank space count = char count – blank space count 7. Print non-blank space count |
| 1. Reset All function | 1. Prompt user to confirm to clear all 3 frames (A, B, C) 2. If Yes then  * Enable and Clear entry box in Frame A * Disable Run Analysis button * Delete content from list box in Frame B * Delete content from textbox in Frame C * Set check box to 0 * Set Progress bar to 0  1. Else Return |
| 1. Export function | 1. Check if text box from Frame C is not empty 2. Open save dialog for client to indicate path for export.  * If the button Run has been clicked only once, then the last processed text file will be defaulted in the save file dialog * Else the save file dialog default file name will be blank when it opens  1. Set export to be .txt format by default 2. Check if client entered a file path for export 3. If file path is correct then  * Write content of text box to file * Log a line in list box indicating that the path of file exported * Log another line in list box indicating that the export is successful * Prompt user the file path |
| 1. Has File Path function | 1. Check entry box from Frame A  * If entry box length is 0 or set to None then return false * Else return true |
| 1. User Prompt Function | 1. Ask user to whether he wants to proceed  * Return user input |
| 1. Get No of Text Lines function | 1. Open file to read number of lines  * Return the number of lines + 1 |
| 1. Update Progress Bar function | 1. Gets the remaining percentage 2. Sets the progress bar = progress bar + remaining percentage 3. Updates idle tasks |
| 1. Exit function | 1. Close the application |
| Class rmFileClass | |
| 1. cleanNewLine | 1. Removes all useless new line which adds up to the total of characters in the file |
| 1. cleanChar | 1. Remove specific characters that are of no use and which adds up to a new word. For example: Name. will be a unique word when Name is already there |