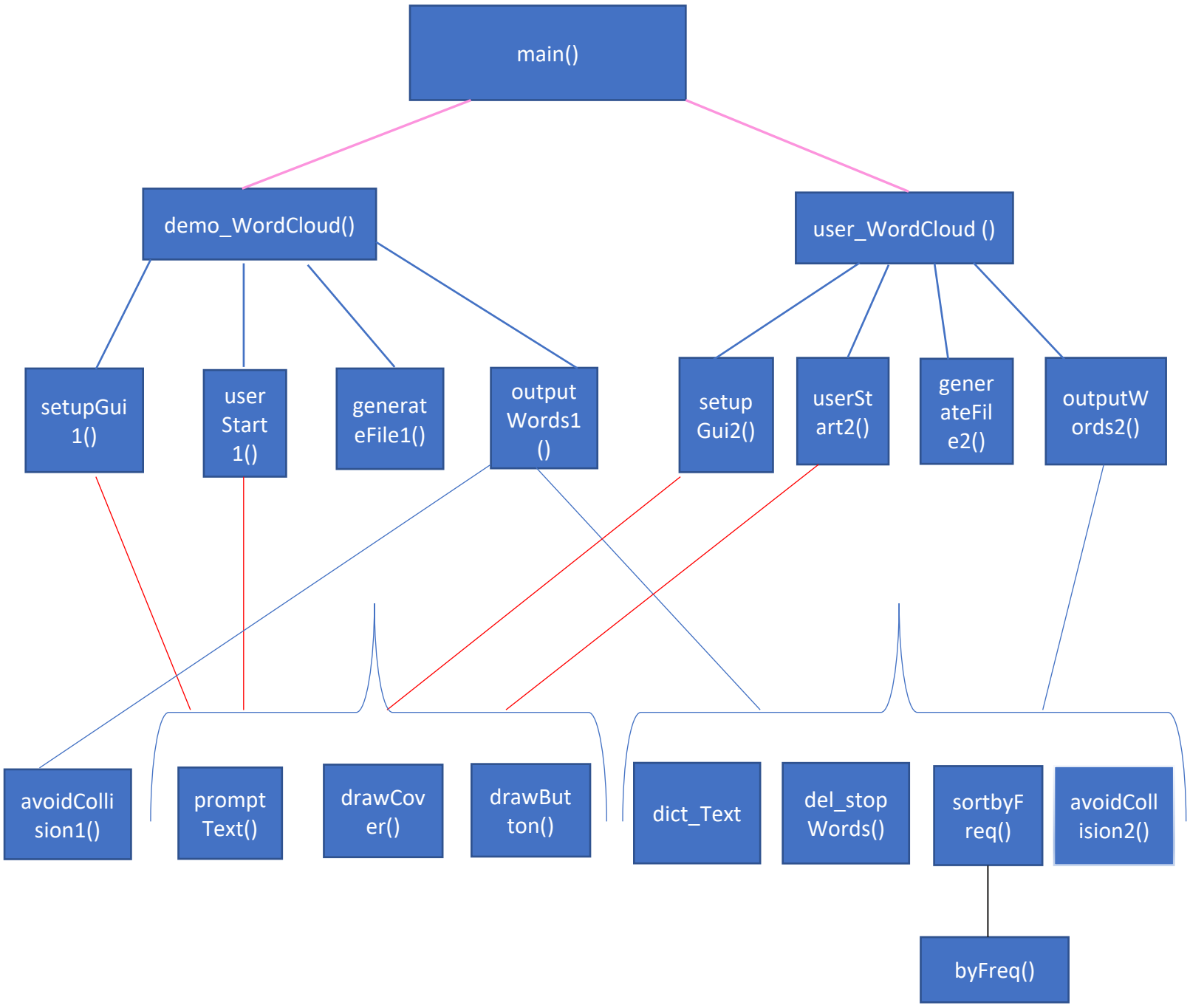
Programming Assignment 4 – Top-Down Documentation – Linh Chi Pham



I. First-Level Design

The program has 2 main parts: A demo word cloud and a customized word cloud generator for users.

Thus, the main() function would have 2 main functions: demo_WordCloud() and user_WordCloud() and 1 function setupGUI1() at the very beginning to return main window(win).

A, demo_WordCloud()
- While loop checking if user click's in button, if not, try again
- Start program 1 once user clicks in the right button
- Another while loop checking if user's click is not in Exit button(that leads to part 3); 3 if conditions for 3 display modes; 1 else statement for when user clicks no button.
- Output 35 words with the highest frequency in the second window (win2)
- After each display mode, user clicks anywhere on the second window to return to main window (win)

B, user_WordCloud()
- If condition to check if user clicks in the right button
- While loop that checks checks if user click's in button, if not, try again
- Output 35 most frequent words into main window (win)

II. Second-Level Design

A, PART 1

- **setupGUI1():**

- A 1000x800 GUI window is set up:
  + Title and subs create using Text objects
  + Import image for visualization
  + Button to start the demo is created

- **userStart1():**

- A cover is drawn to hide previous objects
- Sleep call so that program loads at a natural speed
- (Sub)Titles, visual image are drawn
- 3 buttons/ 3 modes for displaying the word cloud are created
- Another button added to continue to the last part: user's customized word cloud

- **generateFile1():**

- Generate file: story of sun and moon.txt

- **outputWords1():**

- Output 35 words with the highest frequency

B, PART 2

- **setupGUI2():**

- A 1000x800 GUI window is set up:
  + Titles and subtitles are added using text objects
  + Entry boxes for user's inputs (text file, preferred font, text and background colors) created using Entry object
  + Prompt texts for entry boxes (guidance for users to input correctly)

+ Draw button to generate word cloud

- **userStart2():**

- Input boxes are moved away using move()
- A cover is drawn to hide previous objects
- Sleep call so that program loads at a natural speed

- **generateFile2():**

- Generate file: word cloud.txt (default input in entry box)

- **outputWords2():**

- Output 35 words with the highest frequency

III. Third-Level Design

A, avoidCollision1() – for part 1

- Blank list of points pt_list created, into which random points(x,y) in specific range are added (for loop):
  + Loop through items list for 35 words
  + A random point in pt_list is created (text_pt)
  + Blank list chosen_list is created
  + A while loop checks if the random point is already in chosen_list, if it's not, then add it into chosen_list -> no position point(x,y) overlaps with one another.
  + Unique points are then added into chosen_list
  + Unique points (that do not clash) are stored into Text objects (word_output)
  + word_ouput's font and colors are set according to the display mode chosen by user, then drawn into the second window (win2)
  + User clicks anywhere to close the second window and return to main window (win)

B, avoidCollision2() – for part 1

- Blank list of points pt_list created, into which random points(x,y) in specific range are added (for loop):
  + Loop through items list for 35 words
  + A random point in pt_list is created (text_pt)
  + Blank list chosen_list is created
  + A while loop checks if the random point is already in chosen_list, if it's not, then add it into chosen_list -> no position point(x,y) overlaps with one another.
  + Unique points are then added into chosen_list
  + Unique points (that do not clash) are stored into Text objects (word_output)
  + word_ouput's font and colors are set according to user's inputs (using getText), then drawn into the main window (win)
  + User clicks anywhere to close the main window

C, sortbyFreq()

- Sort list items by frequency

D, del_stopWords()
- Read stop words from a text file, then split the stop words file into a list.
- Replace punctuations with blank space-> store into text
- Split the newly modified text into a list
- Create.a for loop, if a word is not in stop word list, append it into a blank list. (words1)

E, dictText()
- Create a blank dictionary
- For every item in words1, if it is already in the dict, return its value in counter, otherwise set to default value 0.

F, prompText()
- Text object formula is created

G, drawButton()
- Rectangle object formula created
- Midpoint of rectangle object is defined, position calculated by deviding the sum of the two points x/y by 2
- Label is put on button at midpoint position

H, drawCover()
- A Rectangle object of size 1000x800 is drawn to cover previous objects

IV. Fourth-Level Design

byFreq()
- For each pair of item in the dictionary, return only the frequency