For the Python file, we implemented regular expressions to get all alphabetic words from the text and convert them to lowercase for uniformity. We used defaultdict to count the occurrences of each word and to calculate how many words appear a certain number of times in the text file. We implemented with open file handling to read the original text and write the results into three separate files: "allwords.txt" for all words in the text file, "uniquewords.txt" for words that appear only once, and "wordfrequency.txt" for displaying a summary of how many words show up the same number of times. We used loops to iterate through lists and dictionary items, and also used the sorted function to organize the frequency data in ascending order.

MonoGame File
Implemented Functionalities:
- DisplayUniqueWords()
  - Reads a list of unique words from uniquewords.txt.
  - Randomly assigns each word a color and position within the display window.
  - Scales words based on frequency data to emphasize relative importance.
  - Implements automatic wrapping so words fit neatly within a bounded area.
  - This function is called when the user presses Enter to toggle into word cloud mode.

- DisplayWordFrequency()
  - Reads data from wordfrequency.txt.
  - Aggregates word counts into frequency buckets (freq, count).
  - Renders a simple bar chart showing the number of words at each frequency.
  - Includes labels (“WORD FREQUENCIES”) and an outlined histogram for clarity.
  - This function is called when toggling into frequency visualization mode.

Other features
- Scalable Bars: Bar heights are normalized relative to the maximum frequency so they always fit on screen.

Spencer: - python file
- readme file
- wordcloud.png
Anunay: - python file
- report
Katie: - Monogame functions
- report