CS 324E Assignment 3
Novel Visualization Project Description

For this project, our group selected A Doll's House: A Play by Henrik Ibsen from Project Gutenberg and built a complete pipeline to extract, analyze, and visualize word data from the text. The project consists of a Python preprocessing script and a MonoGame visualization application that together transform raw literary text into structured data and interactive graphical representations.

The Python script extract_words.py reads the cleaned novel text file with Project Gutenberg metadata removed, converts all text to lowercase, and extracts only alphabetic characters using regular expressions. From this processed corpus, the script generates three output files. allwords.txt contains every word in the play including duplicates. uniquewords.txt contains only words that appear exactly once in the text. wordfrequency.txt maps each frequency value to the number of words that occur at that frequency and is sorted in increasing order. Word occurrences were tracked using a dictionary based counting approach to ensure accuracy and efficiency.

The MonoGame project A3_NovelVisualization consumes these generated files and implements two interactive visualizations. By default, the program displays a word cloud style representation built from a randomized selection of unique words. Words are rendered on a 700 by 600 canvas with dynamic wrapping logic to prevent overflow beyond screen boundaries. Each word's rendered dimensions are measured prior to placement to ensure consistent spacing and legibility. Three custom colors are used to create visual variation, and clicking the canvas generates a new randomized selection of unique words.

Pressing the Enter key toggles to a second visualization that represents the relationship between word frequency and the number of words with that frequency. Horizontal bars are drawn to reflect how many words appear at each frequency level, visually illustrating the characteristic distribution pattern in literary text where many words appear only once and progressively fewer appear at higher frequencies. Input handling uses edge detection logic to ensure that holding the Enter key does not repeatedly toggle between views.

The project was completed collaboratively. Steven and Advaiith were primarily responsible for implementing the Python script and MonoGame functionality, while Harris reviewed the code, refined functionality, and ensured smooth integration between preprocessing and visualization components. All members contributed to debugging, testing, and overall design decisions.

This project demonstrates a full data workflow that begins with extracting structured information from raw text, computing statistical summaries, and translating those summaries into interactive visual representations. The final system satisfies all assignment requirements and provides a clear visualization of word usage patterns within A Doll's House.