Assignment # 15 Project – Playing Chess with Java

Report II

Beifang Yi

4/28/Thursday

The following is based on A#16 Project not A#15!!!

- Algorithms used in the Project
    - String—split: to split a paragraph/line of texts (a long string) into an array of strings. We may use Java’s StringTokenizer or String.split() for this.
    - Word-sorting: for a collection of words (text-strings), we need to sort it so these words will be arranged in alphabetical order.
        - Solution: since Java’s String has embedded comparator, we may represent the words in String and then add them into Java’s TreeMap which can put these words in “order”.
- APIs/Datastructures used in the Project
    - Java’s TreeMap<K, V>
        - The Key will be the inputted words
        - The Value will be the number of times (frequency) of the words (Keys) that occur in the text file.
        - How to use TreeMap<K, V>
            - If the word to be inserted is not in TreeMap, we set its V as 1, otherwise, we need read out its V, increment it by 1, and save it back.
            - After we have put all the words from the file in the TreeMap, we always get the first item on TreeMap (the Key and its associated Value), output them to a file, and then remove this current first item, and read the first item of the updated TreeMap.

- **Pseudocode**

```
1.Open the text file and create a TreeMap<String, Integer> instance, tMap;
2. Read each line of the text in the file and for each line, do the following:
        2.1. Split it into an array of strings
        2.2 For each item in the array, store the item (word) and its associated
            frequency to tMap—the frequency must updated at first
3. Create an output file, oFile, to save the outputs
4. Read the content of TreeMap and save it to oFile according to the following:
                While (tMap is not empty)
                        Read the first key and save this key and its associated V to oFile
                        Remove the first key.
```