

## **Enhancing a Simple Java Text Editor with Statistics, Bracket check, Undo and Redo Features**

**Overview:** The objective of this project was to improve a simple Java text editor by adding bracket check, character count, word count, line count, commonly used code-snippets, as well as undo and redo functionalities to the features already present in the application. These features allow users to view statistics vital to the project updated in real time and ease the process of writing and editing text.

### **Features Added:**

1. **Undo:** Users can undo their previous actions one step at a time, in reverse chronological order. This feature is particularly useful when users make unintended changes or want to revert to an earlier version of their text.
2. **Redo:** Users can redo actions that were previously undone, restoring changes that were reverted using the undo feature.
3. **Bracket Check:** Users can check to see if there is an unclosed bracket within the text area. If there is an unclosed bracket, the bracket checker will return a message in red indicating the line of the unclosed opening bracket. Otherwise, it will return a message in green indicating the brackets are balanced.
4. **Character Count, Word Count, Line Count:** Users can see the respective character, word, and line counts of their texts updated in real-time at a panel at the bottom of the main GUI.
5. **Code-Snippets:** Through utilizing the pulldown menu, users can easily implement commonly used methods in Java, such as Java's main method, a method to read a file into I/O, and a method to write to a file.

### **Technologies Used:**

- Java programming language
- Swing GUI toolkit for Java
- IntelliJ IDE
- GitHub repository

### **Implementation Details:**

- **Undo Stack:** Java Swing's UndoManager framework stores snapshots of the text editor's content at various points in time. Each new edit pushes a snapshot onto the undo stack.
- **Redo Stack:** Like the previous example, a stack is used to store the reversed states (undo actions) that become available after an undo operation.
- **Bracket – Check:** A stack is utilized in which a bracket is pushed onto the stack when it is typed into the text area, and popped off the stack when its coinciding bracket is typed into the text area.
- **Character Count, Word Count, Line Count:** Character Count returns the length of the text area as a string. Word count iterates over the text area, utilizing regex and trim() to differentiate between strings separated by an empty space. Line Count iterates over the

text area and returns the number of lines. Each of these classes implements Java's `DocumentListener` to update the variables in real-time. Through utilizing Java Swing's framework, `JLabels` indicating "Word Count: ", "Character Count: ", and "Line Count: " were added to a `JPanel`, which was then added to the main GUI.

- **Code Snippets:** The code snippets return a string to the text area.

### **Usage Instructions:**

1. Download the source code files or clone the repository to your IDE.
2. In the project display navigate to the class `SimpleJavaTextEditor.java`.
3. Execute the main method. (NOTE: steps 2 and 3 can also be done at by compiling the code and running it on the command line.)
4. Start typing or editing text in the text editor.
5. To undo a previous action, select "Undo" from the Edit menu.
6. To redo an action that was undone, select "Redo" from the Edit menu.
7. To use the bracket check, navigate to the "Tools" pulldown menu and select "Bracket Check."
8. To implement the code-snippets, navigate to "Code Snippets" and select which code snippet you would like to enter into the text area.
9. The statistics for word count, character count, and line count are visible at the bottom of the GUI.

**Contributing:** Contributions to this project are welcome. If you have suggestions for improvements or new features, please open an issue or submit a pull request on GitHub.

### **Acknowledgments:**

- Special thanks to Professor Peter Dordal, who guided us throughout this project and was immensely helpful to us with his suggestions from the beginning to the end of the project.

### **Future Work**

- Set a maximum number of undo/redo steps to prevent redundancy and excessive memory use.
- Provide visual indicators within the text editor to highlight recently undone or redone sections.
- Implement keyboard shortcuts for undo, redo, and bracket check. This can be done easily in the `UI` class.
- Provide a bracket check by hovering the cursor over an opening bracket, perhaps by utilizing `DocumentListener`.

## **Conclusion**

The addition of a bracket check, character count, word count, line count, undo and redo features to the Java text editor enhances its functionality and improves the user experience and enhances the text editor's efficiency. Users can now easily check for missing brackets, undo and redo actions taken in the text area, and view updates to the word count, character count, and line count in real time. Commonly used code-snippets are available in the pulldown menu. As an open-source project, it encourages further development and contributions from willing participants in the community.