**Change request log**

**Team**

Shravan Konduru  
Manish Kanagala

**Change Request**

jEdit change request #1 – Modify the Status Bar to also show the word offset of the caret and the number of words in the file.

**Concept Location**

Time spent: 40 minutes

|  |  |
| --- | --- |
| We ran the system using ant |  |
| We interacted with the text area. | By interacting with the text area, we were able to see the actions reflected on the status bar, getting familiar with what values it should show and how it updates. |
| We went to the file org.git.sp.jedit.jEdit. | Because this was the location the program was running from. |
| We searched for the word caret. Many of the methods related to the word came from View.java, including getTextArea() | This is what the numbers on the status bar are based of off, so it is important to find where the caret is updated. |
| We went to the file View.java and found the code for caret listening and updating. | This lead us to what we were looking for, the caret being updated, but we realized that this wasn’t were the status bar was being updated. |
| We searched the names of files for status bar | This could be another avenue to search for. |
| We went to gui.StatusBar.java |  |
| We searched the file for the word caret and found the function updateCaretStatus() | Since there were other parts to the status bar, we were only interested in the part dealing with the line and offset of the caret |
| We found the section containing the same syntax that is found in the status bar and studied the code | Inside if statements were lines containing buf.append, in the order of #,# (#/#), which is similar to the part of interest in the status bar. |
| We added one to each of the values individually and ran the program | To confirm whether the integer variables in the code corresponded to the correct numbers on the status bar. |
| We marked the method updateCaretStatus() as “located” | We confirmed this class had to be modified. |

**Impact Analysis**

Time spent: 30 minutes

|  |  |
| --- | --- |
| We observed what other methods or classes relied on the updateCaretStatus method. | Interacts with the classes TextArea, Standard Utilies, jEdit, |
| We inspected how a change in updateCaretStatus would change the other classes. | updateCaretStatus only gets information from other classes about things like the caret position and the status of the textArea. |
| We inspected the if statements surrounding many of the caret status updates | The status bar only updates if the caret position and offset is valid, so including the new changes in the same if statement would also make it so that the word count is only updated if the caret position is valid. |
| We concluded that adding new code inside the updateCaretStatus would not impact other classes | As long as we continued to only use get methods for the information about the words in the textArea. |

**Actualization**

Time spent: 60 minutes

|  |  |
| --- | --- |
| We added code under the if statement that checked if caret offset and caret bufferlength were true. | This is the part of the code that updates the first set of parenthesis containing the character offset of the caret and the total number of characters. Placing the new code here guarantees that every new character typed or caret moved results in an updated word count. |
| We made a string to hold the text of the textArea using textArea.getText(), an inbuilt function from the textAreaclass. | This would allow us to count the number of words. |
| We added a for loop that goes through the string for each character. | If the current character is a space or newline, and the character after is not a space or newline, then the total wordcount goes up by 1.  When the character number of the string equals the caret offset-1, the caret wordcount is set to the current value of the total wordcount. |
| We added extra if statements for edge cases such as no characters or the file starting with spaces and newlines. | For the previous part to work smoothly, both counts were initially set to one. If bufferLength is 0, then both are set to 0. If the first character is a space or newline, both are decremented by 1 at the end. |
| We appended the values to buf | In the format consistent with the character offset and total characters. |
| We performed functional testing and also ran the existing test cases. | To make sure it works properly |

**Validation**

Time spent: 15 minutes

|  |  |
| --- | --- |
| Test cases defined:  Inputs: |, a|, a df |s d, \n|, a \n |d  Expected outputs: 1,1 (0/0)(0/0), 1,2 (1/1)(1/1), 1,6 (5/8)(3/4), 2,1 (1/1)(0/0), 2/2 (4/5)(2/2) | This is the regular and only expected behavior  The test passed |

**Timing**

Summarize the time spent on each phase.

|  |  |
| --- | --- |
| **Phase Name** | **Time (in minutes)** |
| Concept location | 40 mins |
| Impact Analysis | 30 mins |
| Actualization | 60 mins |
| Verification | 15 mins |
| **Total** | 145 mins |

**Reverse engineering**

Diagram

Description automatically generated

Chart, box and whisker chart

Description automatically generated

**Conclusions**

For this change the concept location was slightly easier than anticipated, as all that was needed was the StatusBar class, and its relationships between other classes were mostly limited to get requests from other classes and displaying the formated requests on a gui. Concept location, impact analysis, and actualization was done manually in the pair programming process. Testing was done on the text area of jEdit, with the expected outputs shown on the status bar.

Classes and methods changed

* org/gjt/sp/jedit/gui/StatusBar.java
  + updateCaretStatus()