Change request log

# Team

Samuel Armstrong

Ashwin Somasundaram

# Change Request

6.3 : Modify the status bar to show: the word offset of the caret from the beginning of the file and the number of words in the file.

# Concept Location

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | We went into the file list and started looking for anything having to do with the status bar. | We can go through all of these and determine where we need to make our changes, |
| 2 | We messed around in the TextArea.java class. | This is where we found space to make methods for enabling and disabling. |
| 3 | We went to actions.xml | We thought we might need to add actions in here to make the code work. |
| 4 | We went to jedit\_en.props. | We will need to add stuff to make sure the user knows how to enable and disable scroll bars. |
| 5 | We went to jedit\_gui.props | We thought we’d need to do something similar to what we’d need to do in jedit\_gui.props. |
| 6 | We went to textarea.actions.xml | We thought we’d need to do the same thing we did in action.xml |

**Time spent (in minutes):** 45

# Impact Analysis

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | We went through all the files that we thought might need to be changed and checked what we needed to change in them | To help make sure we didn’t miss any file changes we wanted to be as thorough as possible. |
| 2 | We inspected the class TextArea.java | We determined this is where we would make the methods and functionality to disable and enable the scroll bars. |
| 3 | We inspected the actions.xml and textarea.actions.xml | We that these two files would need to have actions for disabling and enabling the scroll bars. |
| 4 | We inspected the jedit\_gui.props and jedit\_en.props files. | We determined this is where we would add the stuff that displayed to the user. |

**Time spent (in minutes):** 30

# Actualization

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | We created two methods: disableScrollBar and enableScrollBar in TextArea.java | The methods were made to make the scroll bars appear and work, in the case of enableScrollBar, and to make them disappear, in the case of disableScrollBar. |
| 2 | We changed actions.xml and textarea.xml | We added the actions to enable and disable scroll bars, along with their labels. |
| 3 | We changed jedit\_en.props. | We just specified what pops up on the user interface when we enable and disable the scroll bars, basically make it more user friendly. |
| 3 | We changed jedit\_gui.props. | We just specified what pops up on the user interface when we enable and disable the scroll bars, basically make it more user friendly. |

**Time spent (in minutes):** 90

# Validation

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | Test Case:  Enable the scroll bars, when there is enough words for them to appear.  Expected output: working scroll bars.  Actual output: working scroll bars. | This is the regular expected behavior.  The test passed. |
| 2 | Test case defined:  Disable the scroll bars, when they should be appearing.  Expected output: no scroll bars  Actual output: no scroll bars | This is the regular expected behavior.  The test passed. |
| 3 | Test case defined:  Disable the scroll bars, when there is not enough text for them to appear.  Expected output: no scroll bars  Actual output: no scroll bars | This is the regular expected behavior.  The test passed. |
| 4 | Test case defined:  Disable the scroll bars, when there is not enough text for them to appear.  Expected output: no scroll bars  Actual output: no scroll bars | This is the regular expected behavior.  The test passed. |

**Time spent (in minutes):** 60

# Timing

Summarize the time spent on each phase.

|  |  |
| --- | --- |
| Phase Name | Time (in minutes) |
| Concept location | 45 |
| Impact Analysis | 30 |
| Actualization | 90 |
| Verification | 60 |
| Total | 225 (3 ¾ hrs) |

# Reverse engineering

Create a UML sequence diagram (or more if needed) corresponding to the main object interactions affected by your change.

Create a partial UML class diagram of the classes visited while navigating through the code. Include the associations between classes (e.g., inheritance, aggregations, compositions, etc.), as well as the important fields and methods of each class that you learn about. The diagram may have disconnected components. Use the UML tool of your preference. When a significant fact about a class or method is learned, indicate it via annotations on the diagram. **For each change request, start with the diagram produced in the previous change request. For the first, you will start from scratch.**

# Conclusions

For this change, concept location was a little difficult as there were a lot more files to be changed for this request as opposed to change request 1, which we did first. Impact Analysis was pretty easy, although it did take a while, we kept getting distracted, but we found all the classes and files we would make an impact on when we were doing concept location, so it was fairly simple.

Classes and methods changed:

* org/jedit/localization/jedit\_en.props
  + added functionality defining what is displayed when User attempts to disable and enable scroll bars.
* org/gjt/sp/jedit/textarea/textarea.actions.xml
  + added actions for enabling and disabling the scroll bars.
* org/gjt/sp/jedit/textarea/TextArea.java
  + added methods enableScrollBar and disableScrollBar
* org/gjt/sp/jedit/jedit\_gui.props
  + added functionality defining what is displayed when User attempts to disable and enable scroll bars.
* org/gjt/sp/jedit/actions.xml
  + added actions for enabling and disabling the scroll bars.

Testing was extremely simple, we just needed to test the disable and enable functions with a few words, then again with a lot fo words to test all possibilities to make sure our code functions properly.