Change request log

# Team

CSRAMS - Ahmar Aftab and Anurag Kumar

# Change Request

Change Request# 2: jEdit displayed the horizontal and vertical scroll bars, whenever the content of the opened document exceeded the size of the editor. Implement an option in the View menu that allows to hide the scrollbars.

*Change Owner:* Implemented and documented by Ahmar Aftab.

# Concept Location

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *To inspect how buttons are added on the toolbar, we inspected the jEdit.java file where the main function was being called. On analyzing the code, we figured out two properties file “jedit\_gui.props” and “jedit\_en.props” are helping to populate data in the toolbar menu.* | *We wanted to locate how the system properties are being loaded.* |
| 2 | *We inspected how an action is getting triggered from the main file and found “actions.xml” file is being used to trigger the actions we want to perform.* | *Once the toolbar is loaded with the additional option, we had to figure out how selecting that option will trigger the option we want to perform.* |
| 3 | *Used InstaSearch to find instances related to ScrollLayout.* | *Because ScrollLayout was the first thing being called to initialize the jEdit TextArea.* |
| 4 | *From the top 3 results, we clicked on the class ScrollLayout.* |  |
| 5 | *We inspected the class ScrollLayout and used dependency navigator of the IDE editor to goto the locations which are making use of that class.* | *We noticed that the TextArea class is using this class to add components like Scroll bar, gutter, and painter.* |
| 6 | *Debugged the source code using Eclipse* | *To know the flow and how the jEdit works.* |
| 7 | *While debugging, interacted with the system by using some displayed Toggle options.* | *To know which functions/ methods are invoked and how those are affecting the operations.* |
| 8 | *We found most of the toggle related options in the View class, which makes call to TextArea and subsequently ScrollLayout.* | *To know the starting point of an action related to toggle operations.* |
| 9 | *We marked the class View as “located”.* | *We confirmed this class had to be modified.* |

**Time spent (in minutes):** 150

# Impact Analysis

Since a new feature is being added, it had minimal or no impact on other related systems. However, we did the below impact analysis:

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *We looked for dependencies in class View using jRipples.* | *To track the classes that could be impacted by the change.* |
| 2 | *No classes found which would be impacted by the change.* | *As a new method is being added, which is specific to that class without having dependency on other class.* |

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

# Actualization

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *We created the method toggleScrollBars method in the View class to hide the scroll bars as needed.* | *Most of the toggle operations were inside this class, so it should be convenient to add a method in that class rather than creating a separate class for that task.* |
| 2 | *Created functions getVerticalBox and getHorizontalBox inside the TextArea class.* | *These functions return the final box configuration which are setup at the time of TextArea initialization* |
| 3 | *We created unit tests for the new class and also performed functional testing. We also ran the existing test cases.* | *To make sure everything works.* |

**Time spent (in minutes):** 75

# Postfactoring (optional)

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *We added a Hashmap data structure to store the vertical and horizontal box configurations in a key value pair.* | *This was done, because we were initializing it inside the toggleScrollBars method, which was erasing the initial data after we check and uncheck the option for the 1st time.* |
| 2 | *We tested again for multiple times for the same operation.* | *Everything worked fine.* |

**Time spent (in minutes):** 75

# Validation

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *Test case defined:*  *Inputs: Check the Toggle Scroll Bars option*  *Expected output: Scroll Bars disappeared* | *This is an expected behavior.*  *The test passed.* |
| 2 | *Test case defined:*  *Inputs: Write something on the editor to exceed the size of the editor*  *Expected output: Scroll Bars remain disappeared.* | *Scrolling possible with mouse possible even though the scroll bars are hidden. The test passed.* |
| 3 | *Test case defined:*  *Inputs: Uncheck the Toggle Scroll Bars option*  *Expected output: Scroll Bars appear again* | *This is also an expected behavior. The test passed.* |

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

# Timing

Summarize the time spent on each phase.

|  |  |
| --- | --- |
| Phase Name | Time (in minutes) |
| Concept location | 150 |
| Impact Analysis | 30 |
| Actualization | 120 |
| Postfactoring | 75 |
| Verification | 30 |
| Total | 405 |

# Reverse engineering

TextArea class

horizontalBox: Box

getVerticalBox:Box

getHorizontalBox:Box

View class

toggleScrollBars():void

The above UML shows the two classes where we did the changes. The two classes do not have any relation and are independent.

# Conclusions

*For this change, concept location was the most difficult part because the system is too large, and its architecture and code are complicated. Concept location, impact analysis, actualization (and change propagation) was done using debugging, which was very useful and Jripples was used but couldn’t get much help out of it. Testing was performed manually.*

*Challenges:*

*The code was too large to analyze and then there were no proper documentation. We had a difficult time removing the errors, even after going through the given steps. The errors existed even after the successful build and the code executed successfully, which was ambiguous.*

*Classes and methods changed/ added:*

* *org/gjt/sp/jedit/View.java/toggleScrollBars*
* *org/gjt/sp/jedit/textarea/TextArea.java*
  + *getVerticalBox()*
  + *getHorizontalBox()*
* *org/gjt/sp/jedit/jedit\_gui.props*
* *org/jedit/localization/jedit\_en.props*
* *org/gjt/sp/jedit/actions.xml*