Opening various image formats

Software Maintance Autumn 2023

Magnus Christian Larsen

September 14, 2023

GitHub Username: Autowinto
Student Mail: magla21@student.sdu.dk
Project: https://github.com/Autowinto/JHotDraw
Date: mm/dd/yyyy

Contents

1 Change Request **3**
1.1 User Story 3
1.2 Acceptance Criteria 3

2 Concept Location **4**

3 Impact Analysis **5**

4 Refactoring Patterns and Code smells **6**

5 Refactoring Implementation **7**

6 Verification **8**

7 Continuous Integration **9**

8 Conclusion **10**

9 Source Code **11**

1 Change Request

From the list of features that were available to be implemented, in the group, it was decided, that I would be implementing the "open various formats" feature. This includes png, jpg, gif, pct and text.

1.1 User Story

The change request, defined as a user story, is as follows:

"As a user, I want to be able to open images of the formats png, jpg, gif, pct, and text, so that I can work with many different formats without having to convert myself"

1.2 Acceptance Criteria

The following subtasks as identified serve as acceptance criteria:

- **When I click on the 'Open' option, the software should allow me to browse and select files in png, jpg, gif, pct, and text formats.**
- **Upon selecting a valid file of the mentioned formats, the software should display the image or content correctly within the workspace.**
- **If I attempt to open a corrupted or unsupported file type, the software should provide a clear error message.**
- **The software should maintain the original quality of the image when opening it.**
- **For text files, the software should provide an option to convert the text into an editable shape or object in the drawing workspace.**

2 Concept Location

Explain the methodology that you have used to locate each concept that was part of your change request. Using Table 1, list all the files in the order that you have visited them (2nd column). Explain how you have found each file (3rd column). You can simply read the source code, but we encourage you to use the features provided by Netbeans and Featureous: "Quick Find", "Find Symbol", "Go To Definition", "Call Browser", "Find all References", "Class View", "Insert Breakpoint", or any other software tools that you want to use. Furthermore, Featureous Inspector is a good starting point for localizing domain classes based on their features. In the fourth column, write what you have learned about the class.

Table 1: The list of all the domain classes visited during concept location.

#	**Domain classes**	**Tool used**	**Comments**

Use Featureous Feature call-tree to provide a tree-based visualization of the runtime call graphs of methods implementing the features of your change request. This view provides an execution-based alternative to the hierarchical fashion of browsing features supported by the mentioned feature inspector view.

3 Impact Analysis

Use Featureous Feature-Code Characterization View to illustrate Scattering and Tangling of each feature that is involved in your Change Request. Write your reflections about how these measurements relate to your feature and future refactorings. For example, text about the purpose of Impact Analysis and ripple effects.

Use Featureous Feature Relations Characterization view to relate your change request features to each other with respect to how they depend on the objects created by other features and how they exchange data by sharing these objects with one another.

Use Feature-code correlation graph and feature-code correlation grid to illustrate detailed investigations of the correspondences between source code units and related features to your change request.

Using table 2 list the packages and their number of classes that you visited after you located the concept. Write short comments explaining what you have learned about each package and how they contribute to your feature?

In the fourth column, mention if the class is related to the concept. Use one of the following terms:

- Use "Unchanged" if the class has no relation to the concept but you have visited it.
- Use "Propagating" if you read the source code of the class and it guides you to the location of the concept, but you will not change it.
- Use "Changed" if the class will be changed.

Table 2: The list of all the packages visited during impact analysis.

Package name	**# of classes**	**Tool used**	**Comments**

4 Refactoring Patterns and Code smells

Describe the code smell that triggered your refactoring, see Chapter 4 in [Ker05]. Describe what you plan to change by refactoring. Why do you think it's good to do the refactoring? Describe the strategy of the refactorings. Remember there is not one way to implement a pattern. Which of the refactorings from [Ker05] did you apply and what was the reasoning behind it?

5 Refactoring Implementation

Explain where and why you made changes in the source code. That is, explain the difference between the actual change set compared to the estimated impacted set from the impact analysis. Which classes or methods did you create or change? Furthermore, describe used design patterns and reflect about improved design to improve future software maintenance.

6 Verification

At class level document unit tests of important business functionality. Document how you have verified your implemented change. Document the results of your acceptance test that test your feature from your change request.

7 Continuous Integration

Reflect on the following questions:

- What is Continuous Integration and how could you apply it?
- How to use version control systems (git) based on your portfolio?

8 Conclusion

Explain your experience with creating the final baseline of JHotDraw. How did you manage to merge in your changes to the baseline ? How was the system tested? Describe your reflection on what went well and what went not so well. Did you have to cut the scope of your change request and did you have to put some issues back into the backlog. What can be done to avoid future problems, i.e. what have you learned from the iteration.

9 Source Code

Submit all your code changes to your local Git repository and finally the remote GitHub repository. Please provide links to your source code / Feature Branch.

References