



**Linnæus University**  
Sweden

## Report

# Assignment 3

*[Subtitle]*



Author: Xingrong Zong  
Supervisor: [Supervisor]  
Semester: VT21  
Course code: 2DV608



## Table of contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Task 1 – Codebase Analysis .....</b>      | <b>3</b> |
| <b>2</b> | <b>Task 2 – Re-engineering Plan .....</b>    | <b>8</b> |
| 2.1      | <i>Task 2.1 – Resolve Cycle Groups .....</i> | 8        |
| 2.2      | <i>Task 2.2 – Re-architect.....</i>          | 9        |
| <b>3</b> | <b>Task 3 – Re-engineering .....</b>         | <b>9</b> |



## 1 Task 1 – Codebase Analysis

| Info                         |                                      | Issues  |       |
|------------------------------|--------------------------------------|---|-------|
| Name:                        | JMines                               | Total:  | 25    |
| Language(s):                 | Java                                 | Configuration:                                    | 0     |
| State:                       | Model loaded                         | Workspace:  | 0     |
| Analyzers:                   | Not running (execution level 'Full') | Threshold violations:                             | 23    |
|                              |                                      | Cycle groups:                                     | 2     |
| Size                         |                                      | Structure   |       |
| Total lines:                 | 21,210                               | System Maintainability Level:                     | 20.00 |
| All comment lines:           | 5,762                                | System ACD:                                       | 76.11 |
| Code comment lines:          | 4,797                                | Highest module ACD:                               | 76.11 |
| Lines of code:               | 9,467                                | Cyclic Java packages:                             | 14    |
| Source element count:        | 7,492                                | Structural debt index (Java packages):            | 1,390 |
| Java packages:               | 14                                   | Component dependencies to remove (Java packages): | 94    |
| Java byte code instructions: | 47,342                               | Parser dependencies to remove (Java packages):    | 450   |
| Components:                  | 81                                   | Cyclic components:                                | 76    |
| Components ignoring issues:  | 0                                    | Structural debt index (components):               | 540   |
|                              |                                      | Component dependencies to remove (components):    | 37    |
|                              |                                      | Parser dependencies to remove (components):       | 170   |

*Figure 1: System Analysis*

As shown in Figure 1, in “Size” section, there are a total of 21210 lines among which 5762 lines are comment lines and 9467 lines are code. Also, there are 14 packages and 81 components which are classes.

In “Issues” section, there are in total 25 issues detected. 23 of them are threshold violations and the rest are cycle groups.

In “Structure” section, it shows the system maintainability level is quite low as 20. The system average component dependency (ACD) is 76.11 which means each class depend on average component with 76.11 other classes, and it is very high. There are 76 cyclic components which results in a structural debt of 540 components. To remove all the cyclic dependencies among these four cyclic components there are three dependencies to remove.



| System                     | Metrics           | Workspace | Workspace Dependencies | Issues | Cycle Groups | Exploration | Exploration |
|----------------------------|-------------------|-----------|------------------------|--------|--------------|-------------|-------------|
| Element                    | Affected Elements | Error     | War...                 | Info   |              |             |             |
| ▼ Main                     | 104               | 2         | 23                     | 0      |              |             |             |
| ▼ jmines                   | 103               | 2         | 22                     | 0      |              |             |             |
| > actions                  | 48                | 2         | 0                      | 0      |              |             |             |
| > dialogs                  | 9                 | 2         | 0                      | 0      |              |             |             |
| > events                   | 3                 | 2         | 0                      | 0      |              |             |             |
| > listeners                | 11                | 2         | 0                      | 0      |              |             |             |
| > panels                   | 13                | 2         | 8                      | 0      |              |             |             |
| > persistence              | 6                 | 2         | 2                      | 0      |              |             |             |
| > AudioPlayer.java         | 1                 | 1         | 0                      | 0      |              |             |             |
| > GameBoard.java           | 6                 | 1         | 9                      | 0      |              |             |             |
| > Robot.java               | 3                 | 1         | 3                      | 0      |              |             |             |
| > TilesShapeUnsupportedExo | 1                 | 1         | 0                      | 0      |              |             |             |

| Issue [24]              | Description                            | Severity  | Category            | Element   | To Element | Provider |
|-------------------------|--|-----------|---------------------|---|------------|----------|
| ■ Namespace Cycle Group | Java Module 'Main' contains 14 cycl... | ● Error   | Cycle Group         | ■ [Critical] Java Package cycle group 1.1       | n/a        | Core     |
| ■ Component Cycle Group | Java Module 'Main' contains 76 cycl... | ● Error   | Cycle Group         | ■ [Critical] Component cycle group 1.1          | n/a        | Core     |
| ■ Threshold Violation   | Modified Cyclomatic Complexity = ...   | ▲ Warning | Threshold Violation | ▲ playVideo(byte, byte, byte, byte[])           | n/a        | Core     |
| ■ Threshold Violation   | Modified Cyclomatic Complexity = ...   | ▲ Warning | Threshold Violation | ▲ playGrid(byte, byte, byte, byte[])            | n/a        | Core     |
| ■ Threshold Violation   | Modified Cyclomatic Complexity = ...   | ▲ Warning | Threshold Violation | ▲ fillDisplayChangeButtonsColorMenu(Ma...       | n/a        | Core     |
| ■ Threshold Violation   | Modified Cyclomatic Complexity = ...   | ▲ Warning | Threshold Violation | ▲ fillDisplayChangeBgColorMenu/MainFra...       | n/a        | Core     |
| ■ Threshold Violation   | Number of Statements = 103 (allow...   | ▲ Warning | Threshold Violation | ▲ fillDisplayChangeBgColorMenu/MainFra...       | n/a        | Core     |
| ■ Threshold Violation   | Modified Cyclomatic Complexity = ...   | ▲ Warning | Threshold Violation | ▲ fillDisplayChangeBgColorMenu/MainFra...       | n/a        | Core     |
| ■ Threshold Violation   | Number of Statements = 103 (allow...   | ▲ Warning | Threshold Violation | ▲ fillDisplayChangeBgColorMenu/MainFra...       | n/a        | Core     |
| ■ Threshold Violation   | Modified Cyclomatic Complexity = ...   | ▲ Warning | Threshold Violation | ▲ fillDisplayChangeBgColorMenu/MainFra...       | n/a        | Core     |
| ■ Threshold Violation   | Number of Statements = 103 (allow...   | ▲ Warning | Threshold Violation | ▲ fillDisplayChangeBgColorMenu/MainFra...       | n/a        | Core     |
| ■ Threshold Violation   | Number of Statements = 103 (allow...   | ▲ Warning | Threshold Violation | ▲ fillDisplayChangeBgColorMenu/MainFra...       | n/a        | Core     |
| ■ Threshold Violation   | Lines of Code = 1,591 (allowed rang... | ▲ Warning | Threshold Violation | ■ GameBoard.java                                | n/a        | Core     |
| ■ Threshold Violation   | Modified Cyclomatic Complexity = ...   | ▲ Warning | Threshold Violation | ▲ getTriangular14Neighborhood(int, int) : ...   | n/a        | Core     |
| ■ Threshold Violation   | Number of Statements = 375 (allow...   | ▲ Warning | Threshold Violation | ▲ getTriangular14Neighborhood(int, int) : ...   | n/a        | Core     |
| ■ Threshold Violation   | Modified Cyclomatic Complexity = ...   | ▲ Warning | Threshold Violation | ▲ getTriangularNeighborhood(int, int) : Co...   | n/a        | Core     |
| ■ Threshold Violation   | Modified Cyclomatic Complexity = ...   | ▲ Warning | Threshold Violation | ▲ getPentagonalNeighborhood(int, int) : C...    | n/a        | Core     |
| ■ Threshold Violation   | Number of Statements = 196 (allow...   | ▲ Warning | Threshold Violation | ▲ getPentagonalNeighborhood(int, int) : C...    | n/a        | Core     |
| ■ Threshold Violation   | Number of Statements = 119 (allow...   | ▲ Warning | Threshold Violation | ▲ initializePentagonalPolygons(int, int) : void | n/a        | Core     |
| ■ Threshold Violation   | Modified Cyclomatic Complexity = ...   | ▲ Warning | Threshold Violation | ▲ getParquetNeighborhood(int, int) : Colle...   | n/a        | Core     |
| ■ Threshold Violation   | Number of Statements = 195 (allow...   | ▲ Warning | Threshold Violation | ▲ getParquetNeighborhood(int, int) : Colle...   | n/a        | Core     |
| ■ Threshold Violation   | Modified Cyclomatic Complexity = ...   | ▲ Warning | Threshold Violation | ▲ containsSchemas() : boolean                   | n/a        | Core     |
| ■ Threshold Violation   | Modified Cyclomatic Complexity = ...   | ▲ Warning | Threshold Violation | ▲ searchSchemas() : boolean                     | n/a        | Core     |
| ■ Threshold Violation   | Number of Statements = 104 (allow...   | ▲ Warning | Threshold Violation | ▲ searchSchemas() : boolean                     | n/a        | Core     |
| ■ Threshold Violation   | Lines of Code = 951 (allowed range...  | ▲ Warning | Threshold Violation | ■ Configuration.java                            | n/a        | Core     |
| ■ Threshold Violation   | Modified Cyclomatic Complexity = ...   | ▲ Warning | Threshold Violation | ▲ manageLocales() : void                        | n/a        | Core     |

Figure 2: Issues detected by Sonargraph, 23 Warnings, 2 Errors

In Figure 2, the details of Issues section are shown. In total 25 issues as mentioned before, 23 of them are warnings as threshold violations and the rest 2 are errors as cycle group.

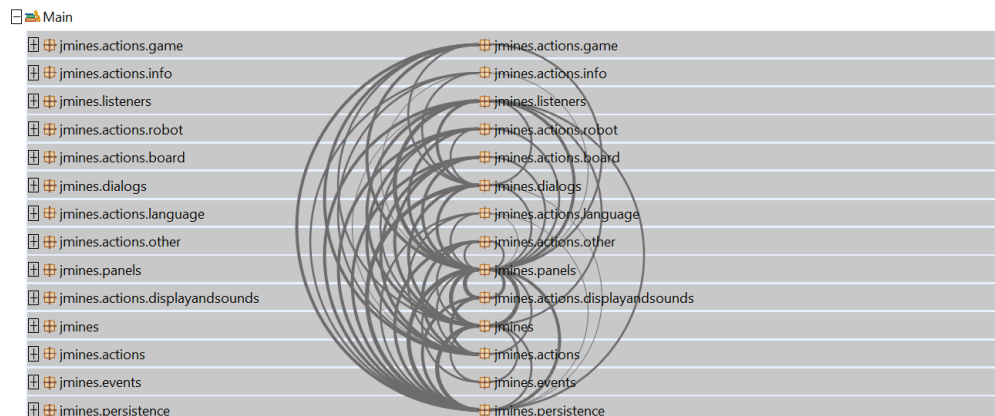
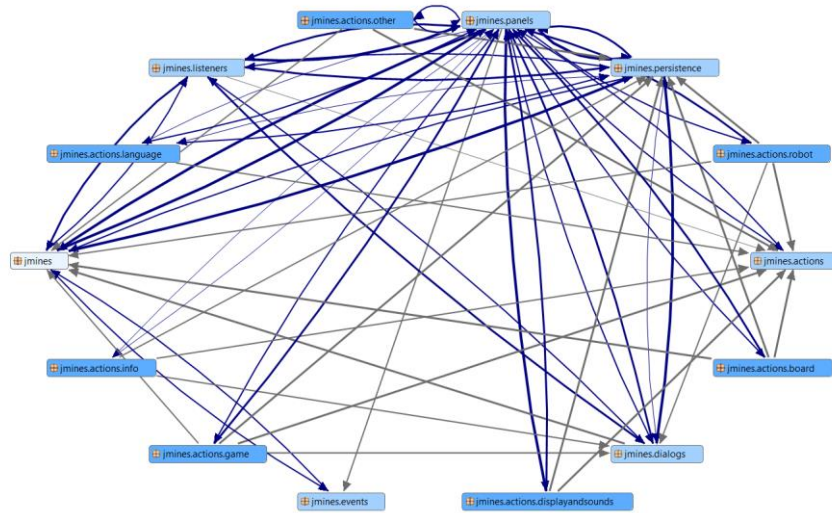


Figure 3: Java Package Cycle Group in Exploration View, 1 Cycle Groups, 14 Cyclic Elements



*Figure 4: Java Package Cycle Group in Cycle View, 14 Cyclic elements*

In Figure 3 and 4, the cycle groups among Java packages are shown in the exploration view and cycle view of Sonargraph. Each java package has dependencies with one or several packages as the above figures show.

The issue of Java Package Cycle Group is at the severity of error. The problem is due to the code of components that creates many dependencies in between and it leads to the problem of cyclic dependency in java packages. To resolve this problem, I need to remove the dependency between java packages by merging smaller packages into a bigger package, however, that would only solve this problem ostensibly. Because later, I will discuss about the Component Cycle Group.

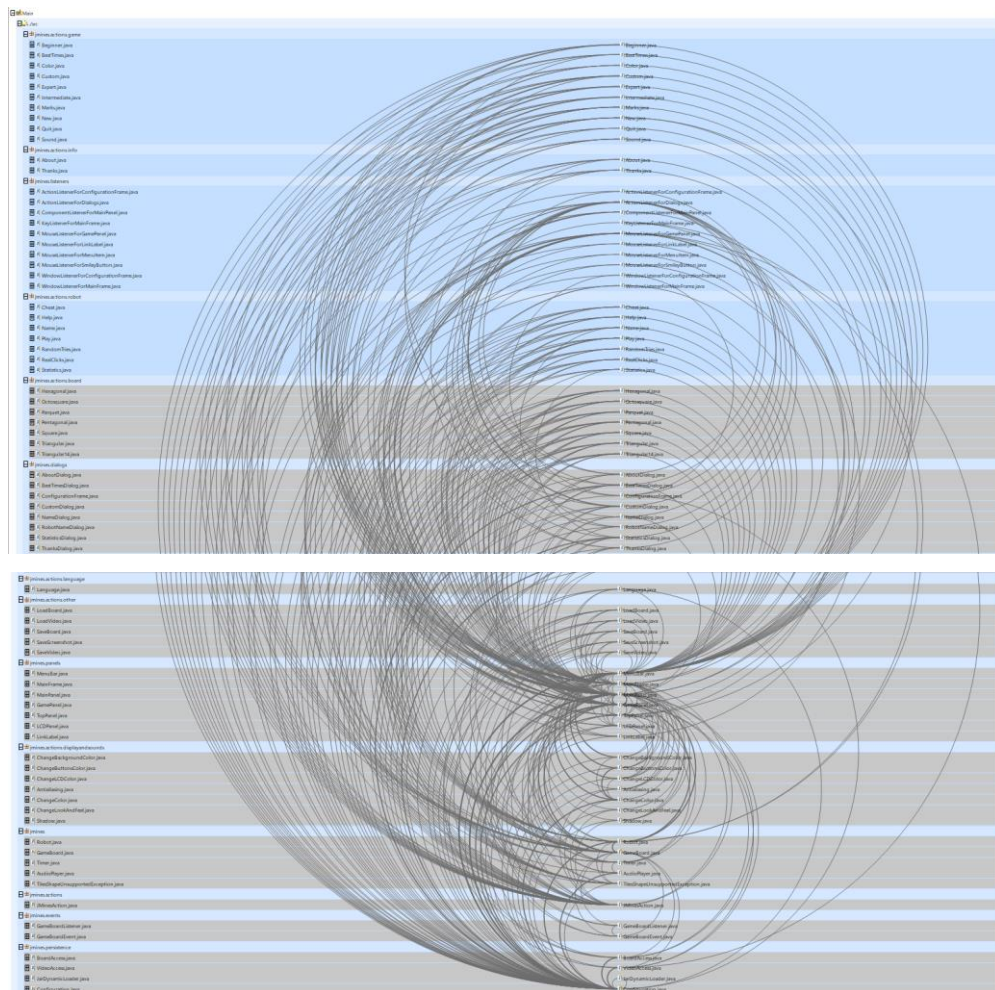
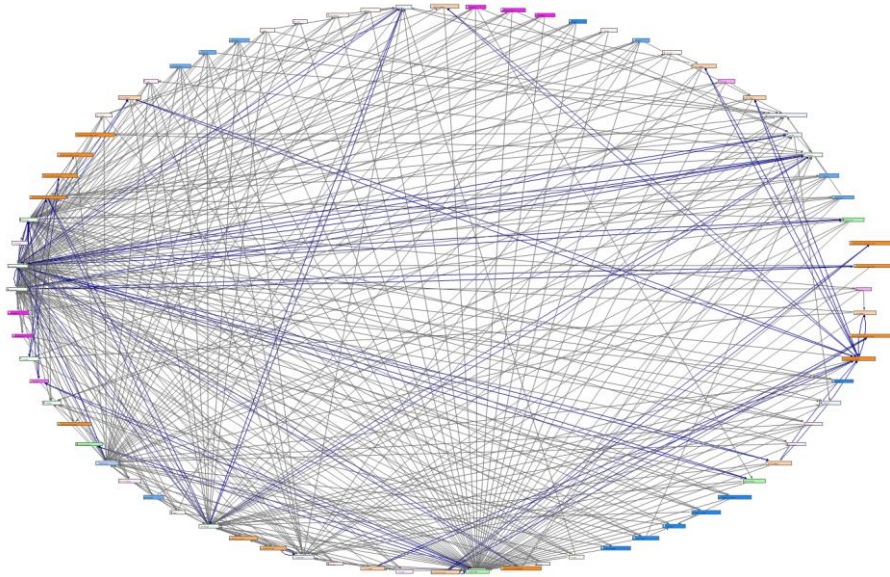


Figure 5: Component Cycle Group in Exploration View, 1 Cycle Group, 76 Cyclic Elements





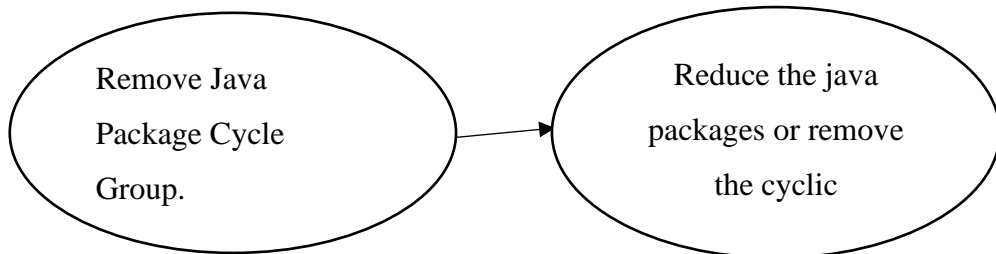
*Figure 6: Component Cycle Group in Cycle View, 76 Cyclic Elements*

In Figure 5 and 6, the cycle groups among Components are shown in the exploration view and cycle view of Sonargraph. Each component has dependencies with one or several packages as the above figures show. As mentioned in the last paragraph, it shows that the reason of having the problem of Java Package Cycle Group is from this Component Cycle Group, because these components are distributed into those java packages. Therefore, the solution for cycle groups is to remove the cyclic dependencies in the classes as much as possible as well as keeping the functionality.

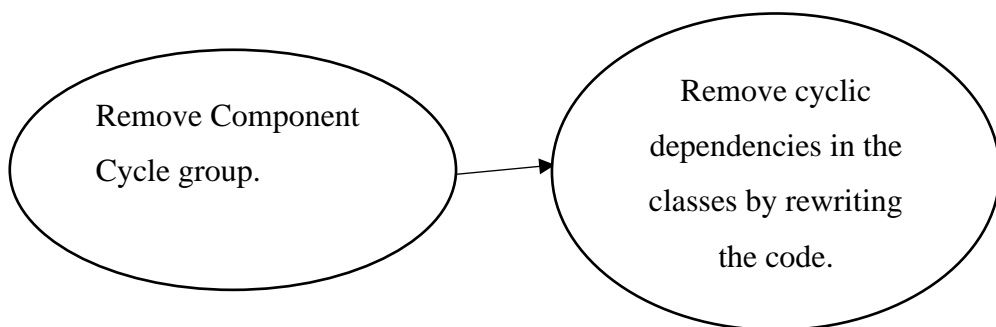


## 2 Task 2 – Re-engineering Plan

### 2.1 Task 2.1 – Resolve Cycle Groups



In order to remove Java Package cycle group, I chose to divide classes by using MVC pattern. Therefore, I created three new packages – “Model”, “View”, “Controller”. Then I moved classes from their original package into these packages and deleted the original packages, it helps to reduce the severity of Java package cycle group from error to warning. The number of cyclic elements should reduce from 14 to 3.



To remove Component cycle group, I decided to go through each class to see with the help of Sonalgraph to check each cyclic dependency, however, there are in total of 76 of them, due to time-consuming and the goal of keep the functionality, this task did not manage to finish in time, but it can be set as a plan with more time given.





## 2.2 Task 2.2 – Re-architect

I went through all the classes to separate them into controller, view and model packages.

- Create a new package “View”, move all classes from package “dialogs” and “panels” into package “View”.
- Create a new package “Controller”, move all classes from package “actions”, “listeners”, “persistence into package “Controller”.
- Create a new package “Model”, move all classes from package “events” and “listeners” into package “Model”.

## 3 Task 3 – Re-engineering

Due to time consuming, the quickest way to solve cycle group is to merge the smaller java packages into a bigger java package. If there is more time, the best solution is to remove the cyclic dependencies in the classes instead. The new codebase Sonagraph analysis could not be shown, because of during the adjustment of java packages, it somehow creates class root destination problem that the package or the class is not recognized or found and therefore, the system could not be compiled, which lead to impossible to import into Sonagraph for new analysis.