



# **CS 1632 Software Quality Assurance**

## **Deliverable 4**

Member 1 Name: Alexander Chlpka

Member 2 Name: Evan Moran

Team Name: Special Sauce

## 1. Introduction

### Division of Work

- Alex - Wrote pinning tests, refactored code
- Evan - Found hot spots in VisualVM, tested changes, wrote report

Evan had difficulty launching VisualVM since he had an OpenJDK installed in addition to the JRE that VisualVM requires to run. This was easily fixed by including the address of the JRE installation as an argument when launching VisualVM from the command line.

Name	Self Time (CPU)	Total Time (CPU)	Invocations
⌚ MainPanel.calculateNextIteration ()	36,171 ms (66.5%)	54,085 ms (29.9%)	76
⌚ MainPanel.iterateCell (int,int)	17,842 ms (32.8%)	17,867 ms (9.9%)	1,900
⌚ Cell.toString ()	234 ms (0.4%)	234 ms (0.1%)	25
⌚ Cell.setAlive (boolean)	46.5 ms (0.1%)	46.5 ms (0%)	3,903
⌚ MainPanel.backup ()	46.1 ms (0.1%)	45.5 ms (0%)	76
⌚ Cell.getAlive ()	15.5 ms (0%)	15.5 ms (0%)	19,050
⌚ MainPanel.getNumNeighbors (int,int)	12.7 ms (0%)	25.6 ms (0%)	1,900
⌚ StopButton\$StopButtonListener.actionPerformed ()	0.0 ms (0%)	0.0 ms (0%)	2
⌚ RunContinuousButton\$RunContinuousButtonListener.actionPerformed ()	0.0 ms (0%)	0.0 ms (0%)	2
⌚ Cell.reset ()	0.0 ms (0%)	0.0 ms (0%)	25
⌚ Cell.resetBeenAlive ()	0.0 ms (0%)	0.0 ms (0%)	53
⌚ StopButton\$StopButtonListener.actionPerformed ()	0.0 ms (0%)	0.0 ms (0%)	2

This is the visualVM hotspot report before making any changes

## 2. Optimized Feature 1

- Write the name of the feature optimized (the label on the button)  
mainPanel.java
- Write the name(s) of the method(s) you refactored to optimize this feature  
iterateCell()  
calculateNextIteration()
- Insert a screenshot of VisualVM CPU profiling **after** optimizing Feature 1. Please include only the “Hot spots” window in the screenshot.

Name	Self Time (CPU)	Total Time (CPU)	Invocations
Cell.toString ()	249 ms (70.1%)	249 ms (24.2%)	25
MainPanel.backup ()	46.0 ms (12.9%)	60.7 ms (5.9%)	97
Cell.setAlive (boolean)	30.8 ms (8.6%)	30.8 ms (3%)	4,953
FileAccess.loadFile (String)	15.6 ms (4.4%)	15.6 ms (1.5%)	1
MainPanel.iterateCell (int)	14.3 ms (4%)	6.70 ms (0.6%)	2,425
StopButton\$StopButtonListener.stop ()	0.0 ms (0%)	0.0 ms (0%)	1
RunContinuousButton\$RunContinuousButtonListener.run ()	0.0 ms (0%)	0.0 ms (0%)	1
Cell.reset ()	0.0 ms (0%)	0.0 ms (0%)	25
Cell.resetBeenAlive ()	0.0 ms (0%)	0.0 ms (0%)	53
Cell.getAlive ()	0.0 ms (0%)	0.0 ms (0%)	24,300
UndoButton\$UndoButtonListener.undo ()	0.0 ms (0%)	0.0 ms (0%)	1
RunButton\$RunButtonListener.run ()	0.0 ms (0%)	0.0 ms (0%)	1
Cell\$CellButtonListener.calculateNextIteration ()	0.0 ms (0%)	15.6 ms (1.5%)	3
ClearButton\$ClearButtonListener.clear ()	0.0 ms (0%)	0.0 ms (0%)	1

### 3. Optimized Feature 2

- Write the name of the feature optimized (the label on the button)  
Cell.java
- Write the name(s) of the method(s) you refactored to optimize this feature  
toString()
- Insert a screenshot of VisualVM CPU profiling **after** optimizing Feature 2. Please show only the profile for Feature 2 (do not invoke Feature 1 while profiling). Please include only the “Hot spots” window in the screenshot, as before.

Name	Self Time (CPU)	Total Time (CPU)	Invocations
MainPanel.calculateNextIteration ()	9,031 ms (68%)	13,216 ms (30%)	19
MainPanel.iterateCell (int, int)	4,187 ms (31.5%)	4,185 ms (9.5%)	475
MainPanel.backup ()	31.0 ms (0.2%)	30.9 ms (0.1%)	19
RunContinuousButton\$RunContinuousButtonListener.actionPerformed (java.awt.event.ActionEvent)	15.6 ms (0.1%)	15.6 ms (0%)	2
Cell.setAlive (boolean)	15.6 ms (0.1%)	15.6 ms (0%)	1,128
StopButton\$StopButtonListener.actionPerformed (java.awt.event.ActionEvent)	0.0 ms (0%)	0.0 ms (0%)	2
Cell.toString ()	0.0 ms (0%)	0.0 ms (0%)	50
Cell.reset ()	0.0 ms (0%)	0.0 ms (0%)	50
Cell.resetBeenAlive ()	0.0 ms (0%)	0.0 ms (0%)	103
Cell.getAlive ()	0.0 ms (0%)	0.0 ms (0%)	4,825
UndoButton\$UndoButtonListener.actionPerformed (java.awt.event.ActionEvent)	0.0 ms (0%)	0.0 ms (0%)	1
RunButton\$RunButtonListener.actionPerformed (java.awt.event.ActionEvent)	0.0 ms (0%)	687 ms (1.6%)	1
Cell\$CellButtonListener.actionPerformed (java.awt.event.ActionEvent)	0.0 ms (0%)	0.0 ms (0%)	3
ClearButton\$ClearButtonListener.actionPerformed (java.awt.event.ActionEvent)	0.0 ms (0%)	0.0 ms (0%)	2
LoadButton\$LoadButtonListener.actionPerformed (java.awt.event.ActionEvent)	0.0 ms (0%)	15.5 ms (0%)	2
WriteButton\$WriteButtonListener.actionPerformed (java.awt.event.ActionEvent)	0.0 ms (0%)	0.0 ms (0%)	2

This is the final result after applying all changes

Name	Self Time (CPU)	Total Time (CPU)	Invocations
MainPanel.backup ()	75.6 ms (52.9%)	73.3 ms (19.8%)	269
MainPanel.getNumNeighbors (int, int)	36.1 ms (25.3%)	26.6 ms (7.2%)	6,725
RunContinuousButton\$RunContinuousButtonListener.actionPerformed (java.awt.event.ActionEvent)	15.6 ms (10.9%)	15.6 ms (4.2%)	2
FileAccess.loadFile (String)	15.6 ms (10.9%)	15.6 ms (4.2%)	4
StopButton\$StopButtonListener.actionPerformed (java.awt.event.ActionEvent)	0.0 ms (0%)	0.0 ms (0%)	3
Cell.toString ()	0.0 ms (0%)	0.0 ms (0%)	75
Cell.reset ()	0.0 ms (0%)	0.0 ms (0%)	150
Cell.resetBeenAlive ()	0.0 ms (0%)	0.0 ms (0%)	276
Cell.getAlive ()	0.0 ms (0%)	0.0 ms (0%)	67,350
Cell.setAlive (boolean)	0.0 ms (0%)	0.0 ms (0%)	13,851
UndoButton\$UndoButtonListener.actionPerformed (java.awt.event.ActionEvent)	0.0 ms (0%)	0.0 ms (0%)	1
Cell\$CellButtonListener.actionPerformed (java.awt.event.ActionEvent)	0.0 ms (0%)	0.0 ms (0%)	26
ClearButton\$ClearButtonListener.actionPerformed (java.awt.event.ActionEvent)	0.0 ms (0%)	0.0 ms (0%)	6
LoadButton\$LoadButtonListener.actionPerformed (java.awt.event.ActionEvent)	0.0 ms (0%)	15.5 ms (4.2%)	4
WriteButton\$WriteButtonListener.actionPerformed (java.awt.event.ActionEvent)	0.0 ms (0%)	0.0 ms (0%)	3
MainPanel.toString ()	0.0 ms (0%)	0.0 ms (0%)	3
MainPanel.clear ()	0.0 ms (0%)	0.0 ms (0%)	6
MainPanel.load (java.util.ArrayList)	0.0 ms (0%)	0.0 ms (0%)	4
MainPanel.stop ()	0.0 ms (0%)	0.0 ms (0%)	3