CS 1632 Software Quality Assurance

Deliverable 4

Member 1 Name: Caleb Beichner

Team Name: caleb_d4

1. **Introduction**

   Write your retrospective here. Please describe the division of work between group members. Also, please describe any difficulties that you faced while using VisualVM.

I decided to do this deliverable myself. I ran into issues with VisualVM where the snapshot would only display the threads, but I think that was more user error. Eventually figured it out and got it working.

2. Optimized Feature 1

a. Write the name of the feature optimized (the label on the button)
Run Continuous

b. Write the name(s) of the method(s) you refactored to optimize this feature
iterateCell, calculateNextIteration

c. 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.**<init>** ()	1,738 ms	(31.4%)	1,981 ms	(8.1%)	64,001
Cell.**getAlive** ()	982 ms	(17.7%)	982 ms	(4%)	640,000
MainPanel.**getNumNeighbors** (int, int)	868 ms	(15.7%)	1,629 ms	(6.7%)	64,000
Cell.**setAlive** (boolean)	556 ms	(10.1%)	556 ms	(2.3%)	128,000
MainPanel.**backup** ()	455 ms	(8.2%)	2,749 ms	(11.3%)	2,561
MainPanel.**iterateCell** (int, int)	287 ms	(5.2%)	2,016 ms	(8.3%)	64,000
Cell$CellButtonListener.**<init>** (Cell)	242 ms	(4.4%)	242 ms	(1%)	64,000
MainPanel.**runContinuous** ()	163 ms	(3%)	5,536 ms	(22.7%)	1
MainPanel.**displayIteration** (boolean[][])	121 ms	(2.2%)	487 ms	(2%)	2,560
MainPanel.**calculateNextIteration** ()	119 ms	(2.2%)	2,623 ms	(10.8%)	2,560
RunContinuousButton$RunContinuousButtonListener.**actionPerformed** (java.awt.event.ActionEvent)	0.238 ms	(0%)	0.258 ms	(0%)	1
RunContinuousButton$GameRunnable.**run** ()	0.152 ms	(0%)	5,537 ms	(22.7%)	1
RunContinuousButton$GameRunnable.**<init>** (RunContinuousButton)	0.020 ms	(0%)	0.020 ms	(0%)	1

3. **Optimized Feature 2**
    a. Write the name of the feature optimized (the label on the button)
    Write

    b. Write the name(s) of the method(s) you refactored to optimize this feature
    toString

    c. 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
FileAccess.**saveFile** (String, String)	35.0 ms	(73%)	35.0 ms	(36%)	93
MainPanel.**toString** ()	6.25 ms	(13%)	10.3 ms	(10.6%)	93
WriteButton$WriteButtonListener.**actionPerformed** (java.awt.event.ActionEvent)	2.63 ms	(5.5%)	48.0 ms	(49.3%)	93
Cell.**getAlive** ()	2.22 ms	(4.6%)	2.22 ms	(2.3%)	2,325
Cell.**toString** ()	1.84 ms	(3.9%)	1.84 ms	(1.9%)	2,325