**IS2545 - DELIVERABLE 5: Performance Testing Conway’s Game of Life**

**Xin Lai (xil154)**

**Name of project: Deliverable5**

**Summary:**

The first step that I did is run the SlowLifeGUI at the size of 16 to see if there is anything wrong. That is, doing the exploratory testing first to see how this application works and how those different functions work as they should be.

While doing the exploratory testing, there is a very noticeable thing. That is, when you try to click the Wrtite button, it will be an unusual delay for this Write function. And other function seems work fine.

Next, I started to profile the application, using VisualVM. It tells that, in the MainPanel class, there are two methods, one is convertToInt and another is runContinuous. Especially the convertToInt method, cost most time when running the application, which looks like is not a good performance. So I trace the original code MainPanel.java to figure it out. Then I found the converToInt method is useless.

Then I look at the runContinuous method, and see that there is for loop about \_r, and it is useless because in the end it changes nothing. So I deleted the for loop.

Backing about the the issue I met in the exploratory, there seems something wrong to the Write Function. Combined the Cell and Cell.toString costing unusual time, I found there might be something wrong in Cell class. So I look at the Cell.toString method and find a useless loop in this method and I modified those.

In the end, I started on the pinning test and also make sure that the methods I modified above should not changed by my refactor.

One more thing that I use manual test to test the runContinuous method issue, because I think it is an easiest way to test this issue.

Git: <https://github.com/vagarii/IS2545>

Modified code can be found in the MainPanel.java file and Cell.java file.

Pinning tests can be found in the PinningTests.java files.

**Manual Test:**

Testing: MainPanel.runContinuous()

1.

**Identifier:** Run and Run Continuous

**Test Case:** The SlowLifeGUI application will run without error when the Run Continuous button is clicked.

**Pre-conditions:**

1. Comment out the useless for loop;
2. Execute the GameOfLife.java by typing “java GameOfLife 16” on command line;
3. Click four alive cells which all in a vertical line.

**Execution Steps:**

1. Click the Run button;
2. Click the Run Continuous button;
3. Watch the output and look for any errors that might have occurred during the application execution.

**Post-conditions:** The messages “Calculating…” and “Displaying…” are repeated shown in the terminal, indicating that the program is running continuously.

2.

**Identifier:** Run Continuous and Stop

**Test Case:** The SlowLifeGUI application will run without error when the Run Continuous button is clicked, and will continue to do so until clicking the Stop button.

**Pre-conditions:**

1. Comment out the useless for loop;
2. Execute the GameOfLife.java by typing “java GameOfLife 16” on command line;
3. Click five alive cells which all in a vertical line.

**Execution Steps:**

1. Click the Run button;
2. Click the Run Continuous button;
3. Click the Stop button.
4. Watch the output and look for any errors that might have occurred during the application execution.

**Post-conditions:** The messages “Calculating…” and “Displaying…” are repeated shown in the terminal, indicating that the program is running continuously. And then stop the repetitions after click the Stop button.

3.

**Identifier:** Run Continuous many times

**Test Case:** The SlowLifeGUI application will run without error when the Run Continuous button is clicked, and then click the Run Continuous button four more times.

**Pre-conditions:**

1. Comment out the useless for loop;
2. Execute the GameOfLife.java by typing “java GameOfLife 16” on command line;
3. Click four alive cells which all in a vertical line.

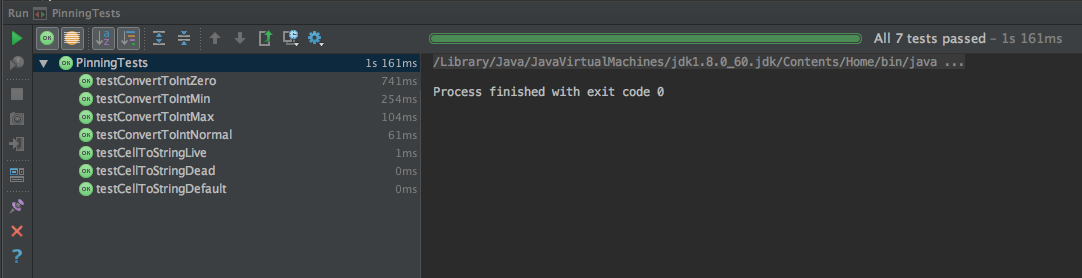
**Execution Steps:**

1. Click the Run button;
2. Click the Run Continuous button;
3. Click the Run Continuous button;
4. Click the Run Continuous button;
5. Click the Run Continuous button;
6. Click the Run Continuous button;
7. Watch the output and look for any errors that might have occurred during the application execution.

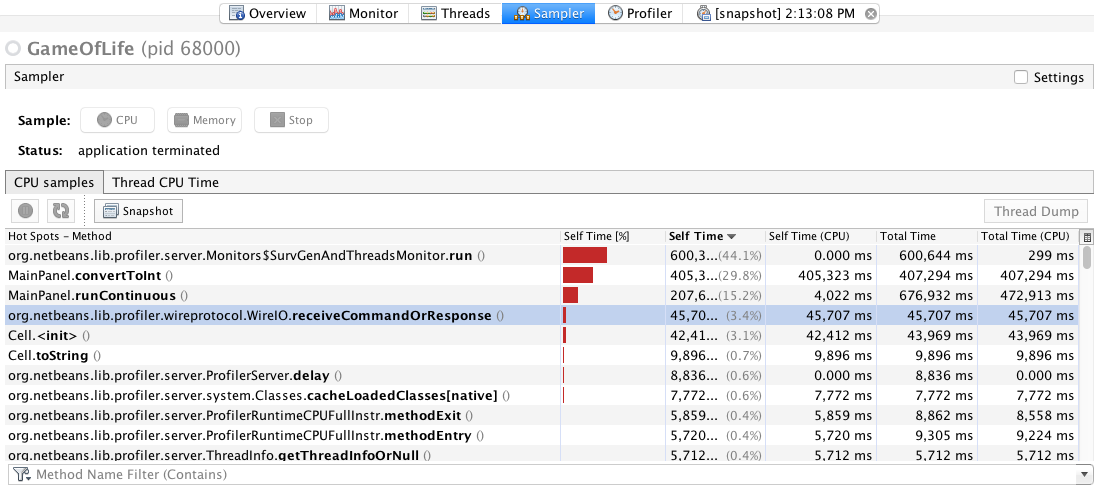
**Post-conditions:** The messages “Calculating…” and “Displaying…” are repeated shown in the terminal, indicating that the program is running continuously.

**Screen Shot:**

* Pinning Tests:



* Visual VM:
* Before modification:



* After modification:

