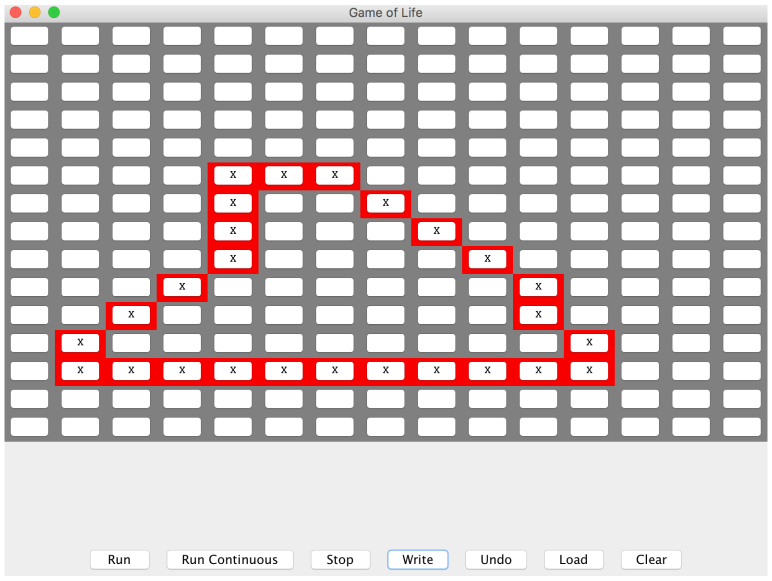
IS 2545 Deliverable 4
Conway's Game of Life Test

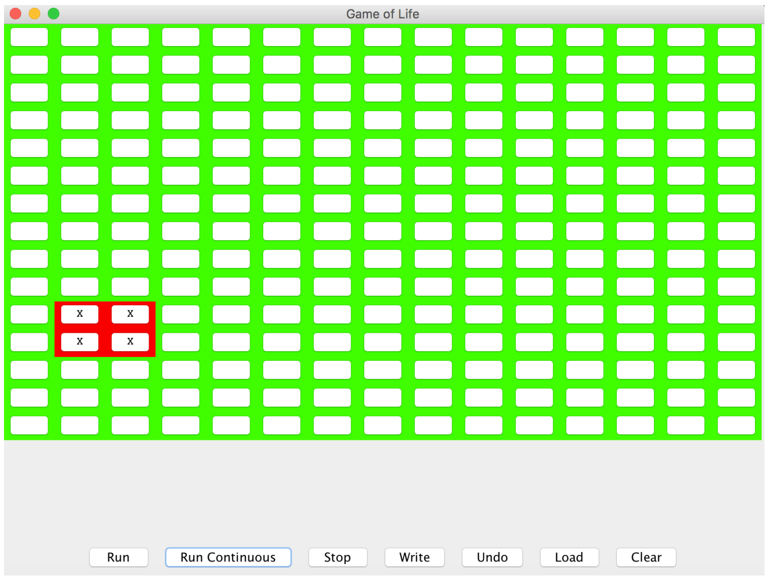
Xiaonan Huang
Xih55@pitt.edu

Summary:

My pattern is :



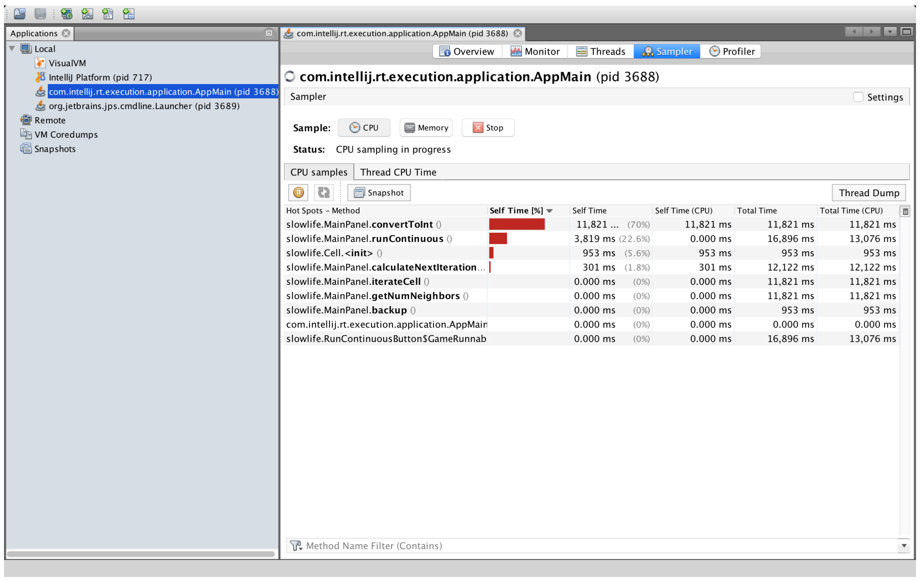
The result is:



VisualVM was used to find which functions were causing the game have poor performance.

For “Write” mode, the toString() function costs lots of times.
For “Run Continuous” mode, the coverToInt() and runContinuous() costs lost of times.

CPU Performance of “Run Continuous” mode before modified:



com.intellij.rt.execution.application.AppMain (pid 3688)

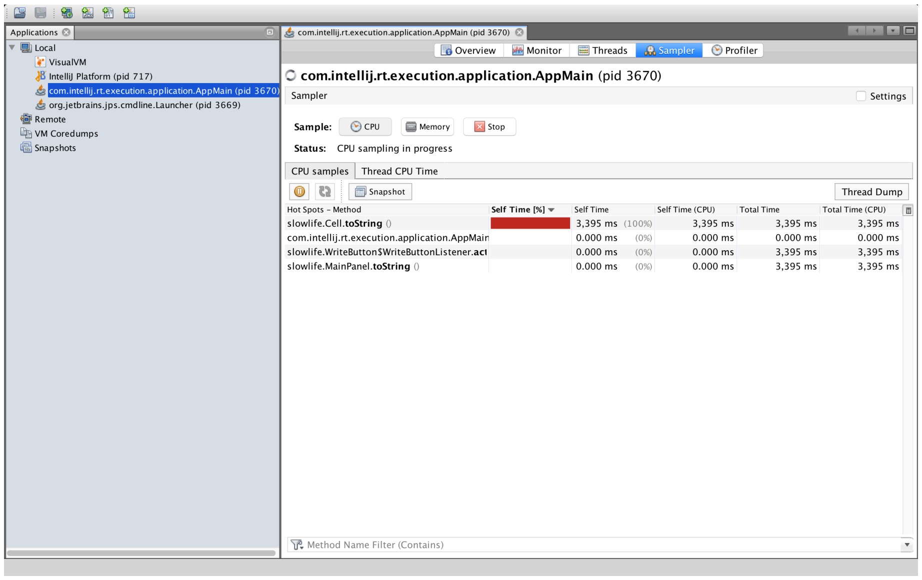
Sampler

Sample: CPU Memory Stop

Status: CPU sampling in progress

Hot Spots – Method	Self Time [%]	Self Time	Self Time (CPU)	Total Time	Total Time (CPU)
slowlife.MainPanel.**convertToInt** ()		11,821 ... (70%)	11,821 ms	11,821 ms	11,821 ms
slowlife.MainPanel.**runContinuous** ()		3,819 ms (22.6%)	0.000 ms	16,896 ms	13,076 ms
slowlife.Cell.**<init>** ()		953 ms (5.6%)	953 ms	953 ms	953 ms
slowlife.MainPanel.**calculateNextIteration**...		301 ms (1.8%)	301 ms	12,122 ms	12,122 ms
slowlife.MainPanel.**iterateCell** ()		0.000 ms (0%)	0.000 ms	11,821 ms	11,821 ms
slowlife.MainPanel.**getNumNeighbors** ()		0.000 ms (0%)	0.000 ms	11,821 ms	11,821 ms
slowlife.MainPanel.**backup** ()		0.000 ms (0%)	0.000 ms	953 ms	953 ms
com.intellij.rt.execution.application.AppMain		0.000 ms (0%)	0.000 ms	0.000 ms	0.000 ms
slowlife.RunContinuousButton$GameRunnab		0.000 ms (0%)	0.000 ms	16,896 ms	13,076 ms

CPU Performance of “Write” mode before modified:



com.intellij.rt.execution.application.AppMain (pid 3670)

Sampler

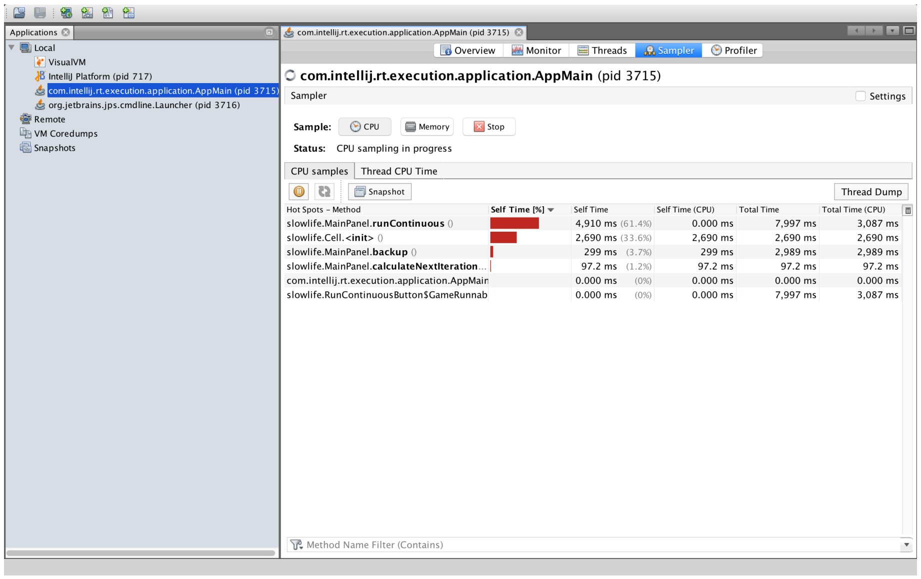
Sample: CPU Memory Stop

Status: CPU sampling in progress

Hot Spots – Method	Self Time [%]	Self Time	Self Time (CPU)	Total Time	Total Time (CPU)
slowlife.Cell.**toString** ()		3,395 ms (100%)	3,395 ms	3,395 ms	3,395 ms
com.intellij.rt.execution.application.AppMain		0.000 ms (0%)	0.000 ms	0.000 ms	0.000 ms
slowlife.WriteButton$WriteButtonListener.**act**		0.000 ms (0%)	0.000 ms	3,395 ms	3,395 ms
slowlife.MainPanel.**toString** ()		0.000 ms (0%)	0.000 ms	3,395 ms	3,395 ms

After modifying coverToInt() and runContinuous(), the performance of CPU has been significantly increased.

CPU Performance of “Run Continuous” mode After coverToInt() modified:



Applications
Local
VisualVM
IntelliJ Platform (pid 717)
com.intellij.rt.execution.application.AppMain (pid 3715)
org.jetbrains.jps.cmdline.Launcher (pid 3716)
Remote
VM Coredumps
Snapshots

com.intellij.rt.execution.application.AppMain (pid 3715)

Overview | Monitor | Threads | Sampler | Profiler

com.intellij.rt.execution.application.AppMain (pid 3715)

Sampler

Sample: CPU Memory Stop

Status: CPU sampling in progress

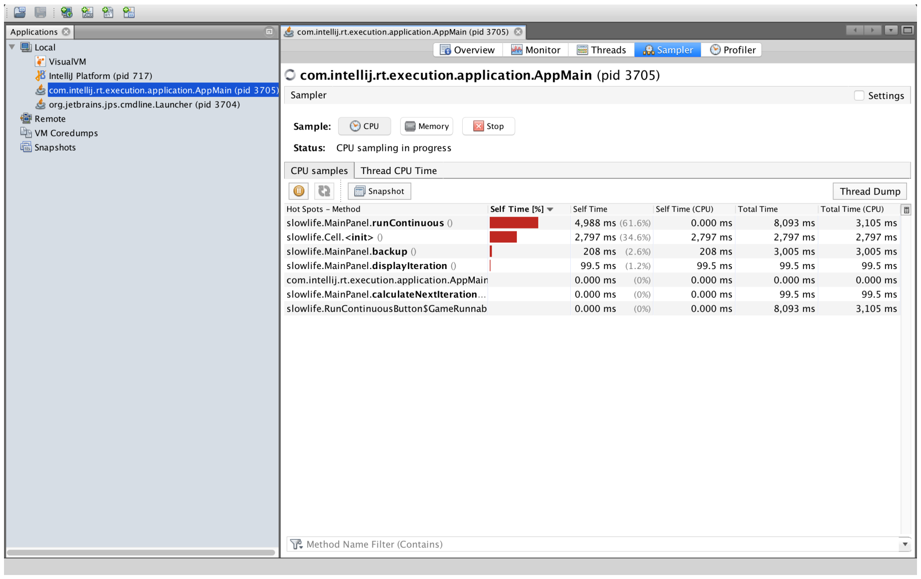
CPU samples | Thread CPU Time

Snapshot | Thread Dump

Hot Spots – Method	Self Time [%]	Self Time	Self Time (CPU)	Total Time	Total Time (CPU)
slowlife.MainPanel.**runContinuous** ()		4,910 ms (61.4%)	0.000 ms	7,997 ms	3,087 ms
slowlife.Cell.**<init>** ()		2,690 ms (33.6%)	2,690 ms	2,690 ms	2,690 ms
slowlife.MainPanel.**backup** ()		299 ms (3.7%)	299 ms	2,989 ms	2,989 ms
slowlife.MainPanel.**calculateNextIteration**...		97.2 ms (1.2%)	97.2 ms	97.2 ms	97.2 ms
com.intellij.rt.execution.application.AppMain		0.000 ms (0%)	0.000 ms	0.000 ms	0.000 ms
slowlife.RunContinuousButton$GameRunnab		0.000 ms (0%)	0.000 ms	7,997 ms	3,087 ms

Method Name Filter (Contains)

CPU Performance of “Run Continuous” mode After runContinuous() modified:



Applications
Local
VisualVM
IntelliJ Platform (pid 717)
com.intellij.rt.execution.application.AppMain (pid 3705)
org.jetbrains.jps.cmdline.Launcher (pid 3704)
Remote
VM Coredumps
Snapshots

com.intellij.rt.execution.application.AppMain (pid 3705)

Overview | Monitor | Threads | Sampler | Profiler

com.intellij.rt.execution.application.AppMain (pid 3705)

Sampler

Sample: CPU Memory Stop

Status: CPU sampling in progress

CPU samples | Thread CPU Time

Snapshot | Thread Dump

Hot Spots – Method	Self Time [%]	Self Time	Self Time (CPU)	Total Time	Total Time (CPU)
slowlife.MainPanel.**runContinuous** ()		4,988 ms (61.6%)	0.000 ms	8,093 ms	3,105 ms
slowlife.Cell.**<init>** ()		2,797 ms (34.6%)	2,797 ms	2,797 ms	2,797 ms
slowlife.MainPanel.**backup** ()		208 ms (2.6%)	208 ms	3,005 ms	3,005 ms
slowlife.MainPanel.**displayIteration** ()		99.5 ms (1.2%)	99.5 ms	99.5 ms	99.5 ms
com.intellij.rt.execution.application.AppMain		0.000 ms (0%)	0.000 ms	0.000 ms	0.000 ms
slowlife.MainPanel.**calculateNextIteration**...		0.000 ms (0%)	0.000 ms	99.5 ms	99.5 ms
slowlife.RunContinuousButton$GameRunnab		0.000 ms (0%)	0.000 ms	8,093 ms	3,105 ms

Method Name Filter (Contains)

Method which has been modified:

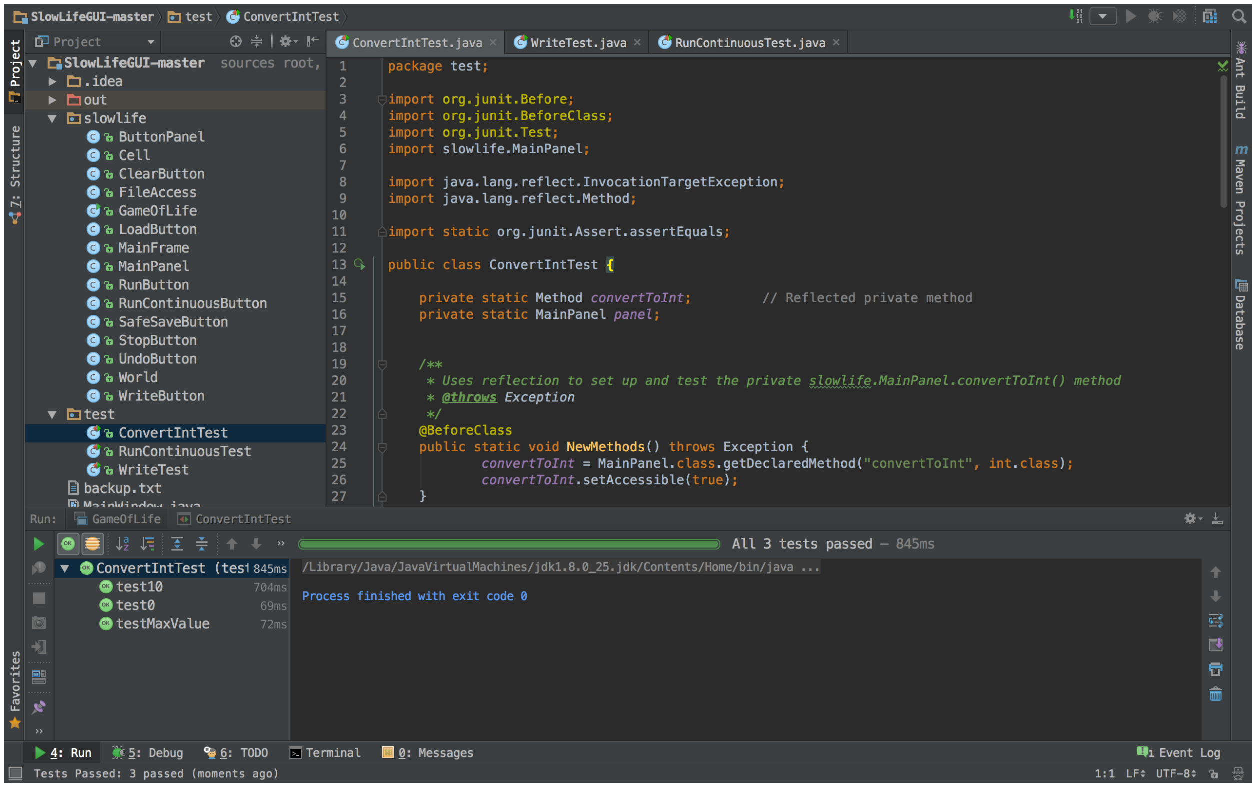
```
//Modified
private int convertToInt(int x) {

	if (x < 0) {
		throw new NumberFormatException();
	}
	return x;
}


//Modified
public void runContinuous() {
	_running = true;
	while (_running) {
	System.out.println("Running...");

	try {
		Thread.sleep(20);
	}
	catch (InterruptedException iex) {
	}

	backup();
	calculateNextIteration();
	}
```

JUnit Test:

There are 3 method contains 9 tests:

For convertToInt():

Test a random integer 10
Test the maximum int (Edge case)
Test zero (Edge case)

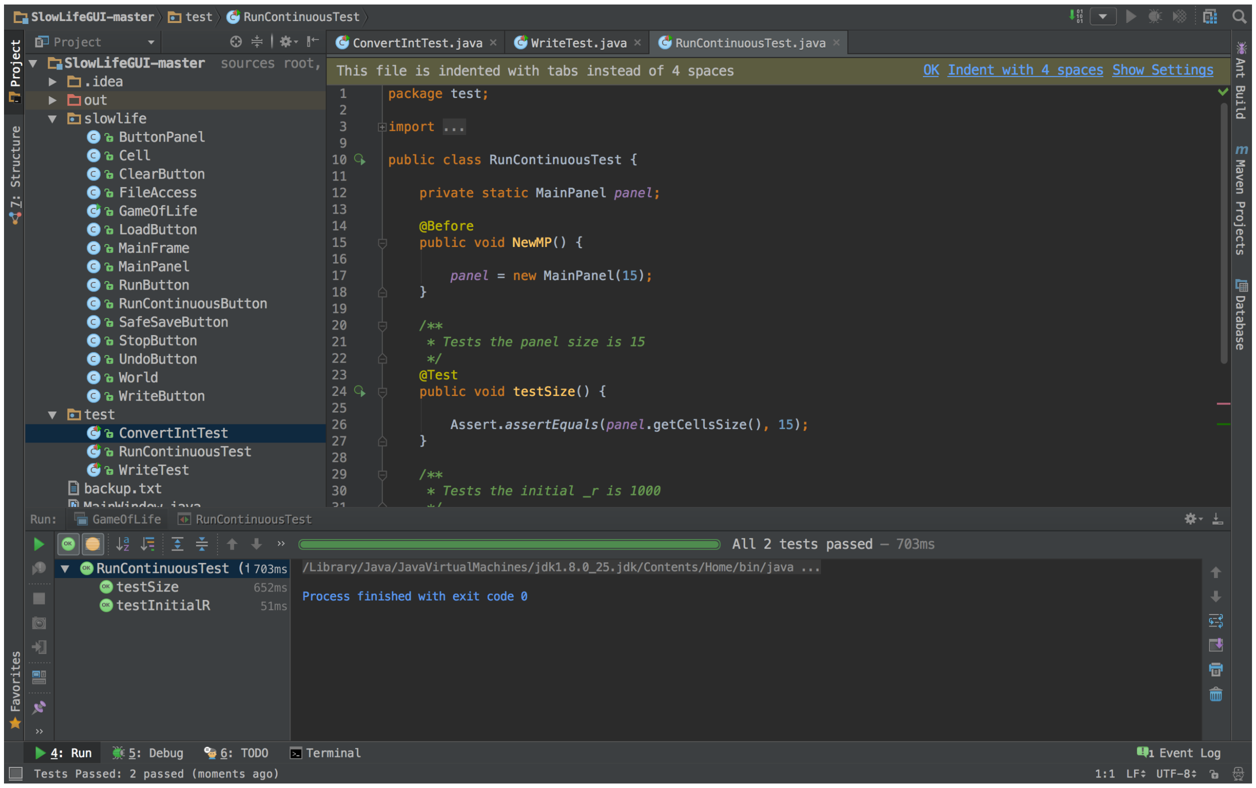


```java
package test;

import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import slowlife.MainPanel;

import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;

import static org.junit.Assert.assertEquals;

public class ConvertIntTest {

    private static Method convertToInt;         // Reflected private method
    private static MainPanel panel;


    /**
     * Uses reflection to set up and test the private slowlife.MainPanel.convertToInt() method
     * @throws Exception
     */
    @BeforeClass
    public static void NewMethods() throws Exception {
            convertToInt = MainPanel.class.getDeclaredMethod("convertToInt", int.class);
            convertToInt.setAccessible(true);
    }
```

```
All 3 tests passed – 845ms
ConvertIntTest (tesi 845ms
    test10         704ms
    test0           69ms
    testMaxValue    72ms

/Library/Java/JavaVirtualMachines/jdk1.8.0_25.jdk/Contents/Home/bin/java ...

Process finished with exit code 0

Tests Passed: 3 passed (moments ago)
```

For RunContinuous:

Test the panel size is 15
Test initial _r is 1000



SlowLifeGUI-master 〉 test 〉 RunContinuousTest

ConvertIntTest.java × | WriteTest.java × | RunContinuousTest.java ×

This file is indented with tabs instead of 4 spaces — OK Indent with 4 spaces Show Settings

```
SlowLifeGUI-master  sources root,
    .idea
    out
    slowlife
        ButtonPanel
        Cell
        ClearButton
        FileAccess
        GameOfLife
        LoadButton
        MainFrame
        MainPanel
        RunButton
        RunContinuousButton
        SafeSaveButton
        StopButton
        UndoButton
        World
        WriteButton
    test
        ConvertIntTest
        RunContinuousTest
        WriteTest
    backup.txt
```

```java
package test;

import ...

public class RunContinuousTest {

	private static MainPanel panel;

	@Before
	public void NewMP() {

		panel = new MainPanel(15);
	}

	/**
	 * Tests the panel size is 15
	 */
	@Test
	public void testSize() {

		Assert.assertEquals(panel.getCellsSize(), 15);
	}

	/**
	 * Tests the initial _r is 1000
```

Run: GameOfLife | RunContinuousTest

All 2 tests passed – 703ms

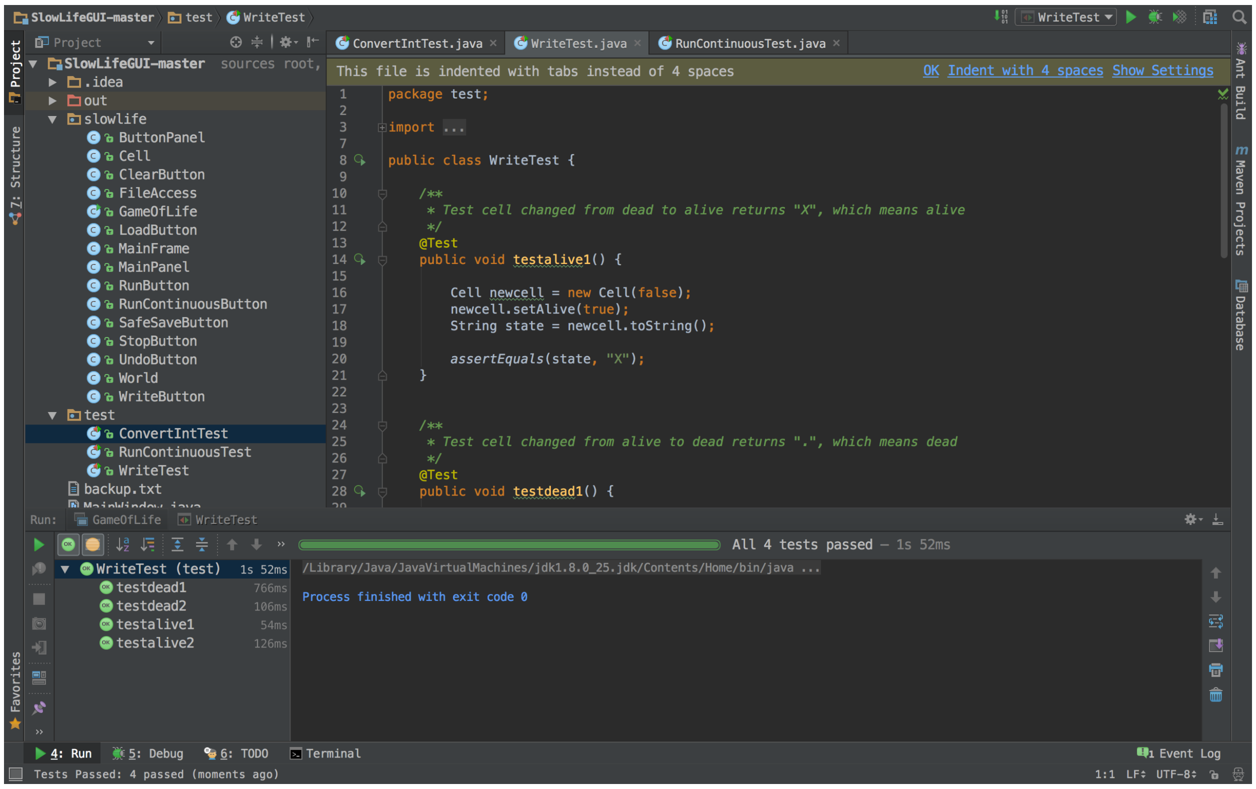
```
RunContinuousTest  703ms
    testSize       652ms
    testInitialR    51ms
```

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_25.jdk/Contents/Home/bin/java ...

Process finished with exit code 0
```

Tests Passed: 2 passed (moments ago)

For Write:

Test cell changed from dead to alive returns "X", which means alive
Test cell changed from alive to dead returns ".", which means dead
Test cell with the parameter true returns "X", which means alive
Test cell with the parameter false returns ".", which means dead



This file is indented with tabs instead of 4 spaces

```java
package test;

import ...

public class WriteTest {

	/**
	 * Test cell changed from dead to alive returns "X", which means alive
	 */
	@Test
	public void testalive1() {

		Cell newcell = new Cell(false);
		newcell.setAlive(true);
		String state = newcell.toString();

		assertEquals(state, "X");
	}


	/**
	 * Test cell changed from alive to dead returns ".", which means dead
	 */
	@Test
	public void testdead1() {
```

Run: GameOfLife WriteTest

All 4 tests passed – 1s 52ms

```
WriteTest (test)   1s 52ms
  testdead1         766ms
  testdead2         106ms
  testalive1         54ms
  testalive2        126ms
```

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_25.jdk/Contents/Home/bin/java ...

Process finished with exit code 0
```

Tests Passed: 4 passed (moments ago)   1:1 LF÷ UTF-8÷