

Software Methods and Tools
Fall 2015 Assignment 7 – JUnit Testing

Sri Harsha Chennavajjala
sc9v9@mail.umkc.edu

1. What problems did you find in the code? For each problem, further explain how you found it (e.g. using which test case).

When I test the application, I found that the score is increasing abnormally. I've found this by testing the *updateGame()* method of *Tetris* class and *checkLines()* method of *BoardPanel* class.

TetrisTest.java

This JUnit test file has the logic to test the *updateGame()* method of *Tetris* class.

BoardPanelTest.java

This JUnit test file has the logic to test *isValidAndEmpty()*, *addPiece()* and *checkLines()* methods of *BoardPanel* class.

2. Specifically explain the test case that you have created for the *updateGame* method of Class *Tetris*. What is your input, and what is your expected output? What is your logic of testing this method?

This method doesn't have any return type. So, in order to test this method, I've temporarily changed the return type of this method. Based on the return value of this method, I am validating its correctness.

Once we successfully cleared the row, then score should be updated. Score assignment is as below...

1 row cleared – 100 points

2 rows cleared – 200 points

3 rows cleared – 400 points

4 rows cleared – 800 points

At any point of time, we can clear a maximum of 4 rows in a single go. This is because the maximum length of all types of tiles is 4 (for I shaped tile).

This information will come from the method *checkLine()* method of *BoardPanel* class. As I did the unit testing on *updateGame()* method, the number of rows cleared each time is 22. This is the reason for the abnormal scores.

The general equation for score is:

```
score += 50 << cleared
```

```
score = score + 50*2[0-4]
```

Here the range of *cleared* variable should be:

`0 <= cleared <= 4`

But, here as we are always getting 22 cleared rows (which we should not get), the score is becoming abnormal.

New equation for score is:

`score = score + 50 * 222`

To identify this bug, I've created a test case for *updateGame()* method which check the validity of number of cleared rows. Here, once the tile reached the bottom of the board, even if the row is not full, we get 22 (which is total height of board) as the cleared rows. My test case will check for count and returns fail status. Below is the output of the test case.

The screenshot displays the Eclipse IDE interface during a debug session. The top toolbar includes standard development tools. The 'Debug' console shows the execution flow of the Tetris application, with the main thread suspended at line 212 of BoardPanel.java. The 'Variables' window on the right shows the current state of variables: 'this' points to the BoardPanel object, 'completedLines' is 22, and 'row' is 21. The 'BoardPanel.java' file is open, showing a loop that checks for empty rows. Handwritten red text 'should be 2 P y O ->' and 'should be false' are visible. A Tetris game window is also shown in the foreground, displaying the game board and controls.

Debug - Tetris-master/src/org/psnbtech/BoardPanel.java - Eclipse

File Edit Source Refactor Navigate Search Project Games Run Window Help

Quick Access Java EE Debug

Debug Servers

Tetris [Java Application]

org.psnbtech.Tetris at localhost:49941

Thread [main] (Suspended)

BoardPanel.checkLines() line: 212

Tetris.updateGame() line: 341

Tetris.startGame() line: 294

Tetris.main(String[]) line: 564

Thread [AWT-EventQueue-0] (Running)

C:\Program Files\Java\jre1.8.0_60\bin\javaw.exe (Nov 29, 2015, 3:03:49 PM)

BoardPanel.java Tetris.java

```
224 */
225 private boolean checkLine(int line) {
226     /*
227      * Iterate through every column in this row. If an
228      * empty, then the row is not full.
229      */
230     for(int col = 0; col < COL_COUNT; col++) {
231         if(!isOccupied(col, line)) {
232             return true;
233         }
234     }
235 }
```

should be 2 P y O ->

should be false

Variables

Name Value

this BoardPanel (id=44)

completedLines 22

row 21

Tetris

Next Piece: [Yellow Tetris Piece]

Stats

Level: 1

Score: 209715200

Controls

A - Move Left

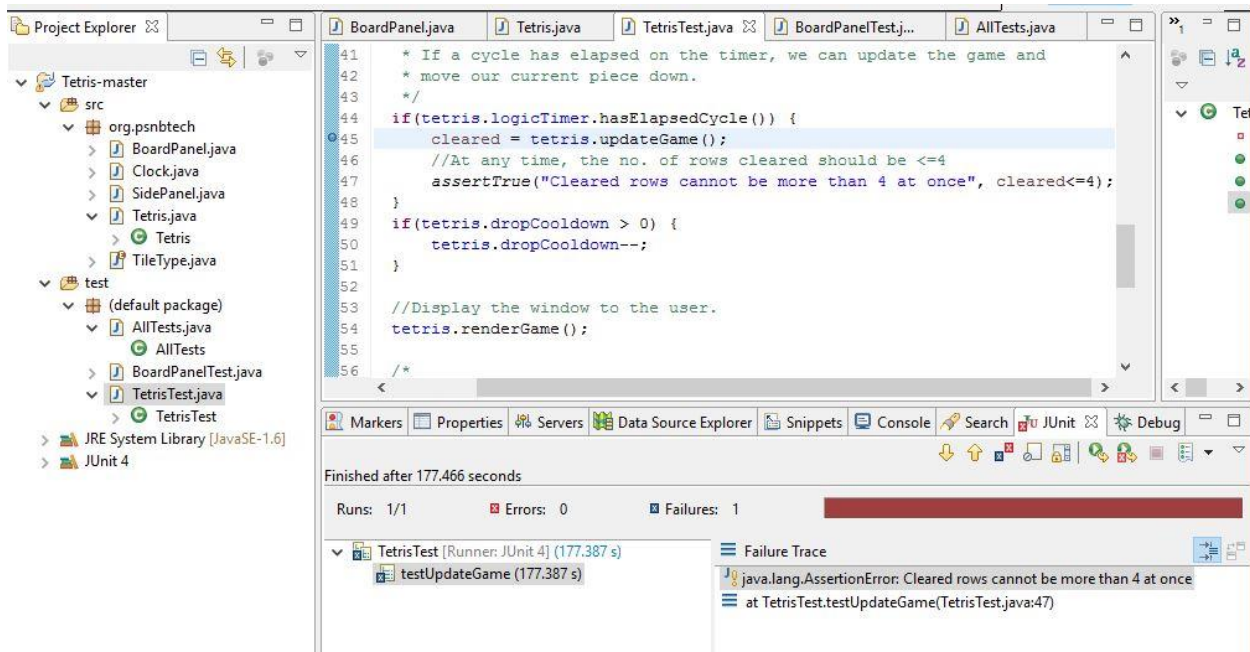
D - Move Right

Q - Rotate Anticlockwise

E - Rotate Clockwise

S - Drop

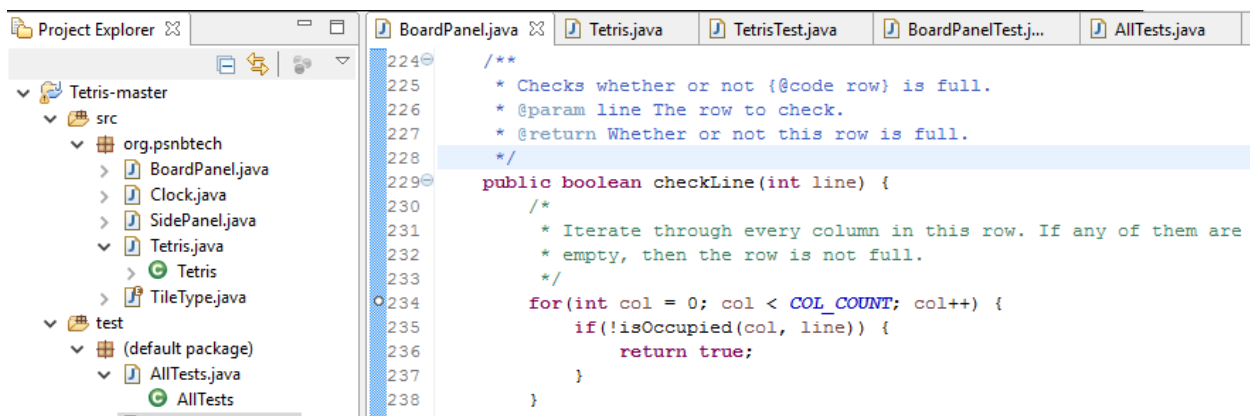
P - Pause Game



Here, the test case is failed (which is success scenario for us) because the method *updateGame()* received invalid number of cleared rows.

3. Testing *BoardPanel* methods:

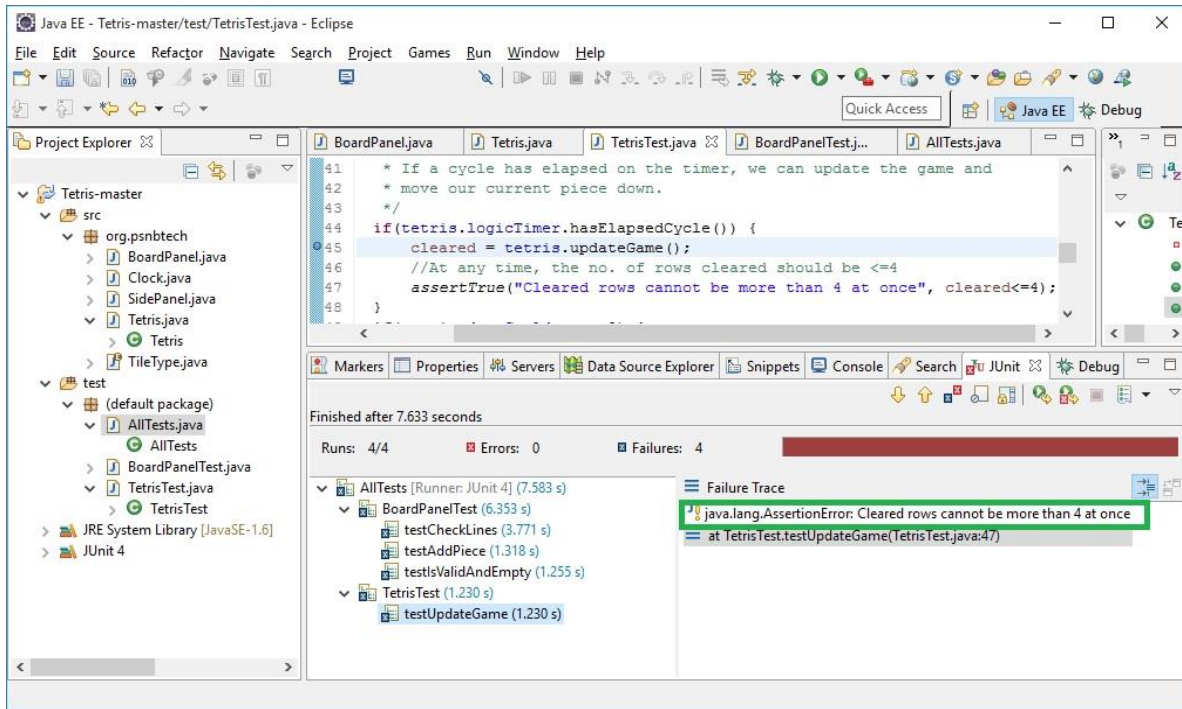
The methods *isValidAndEmpty* and *addPiece* don't have any bugs in them. But the method *checkLines()* is receiving invalid input from *checkLine()* method. *checkLine()* method should return *false* if the position on the board is occupied. But it is returning *true*. So, *checkLines()* method yielding invalid results. Here, *addPiece()* method doesn't have a return type. So, I've temporarily modified its code such that it returns a value based on its side effect on the application.



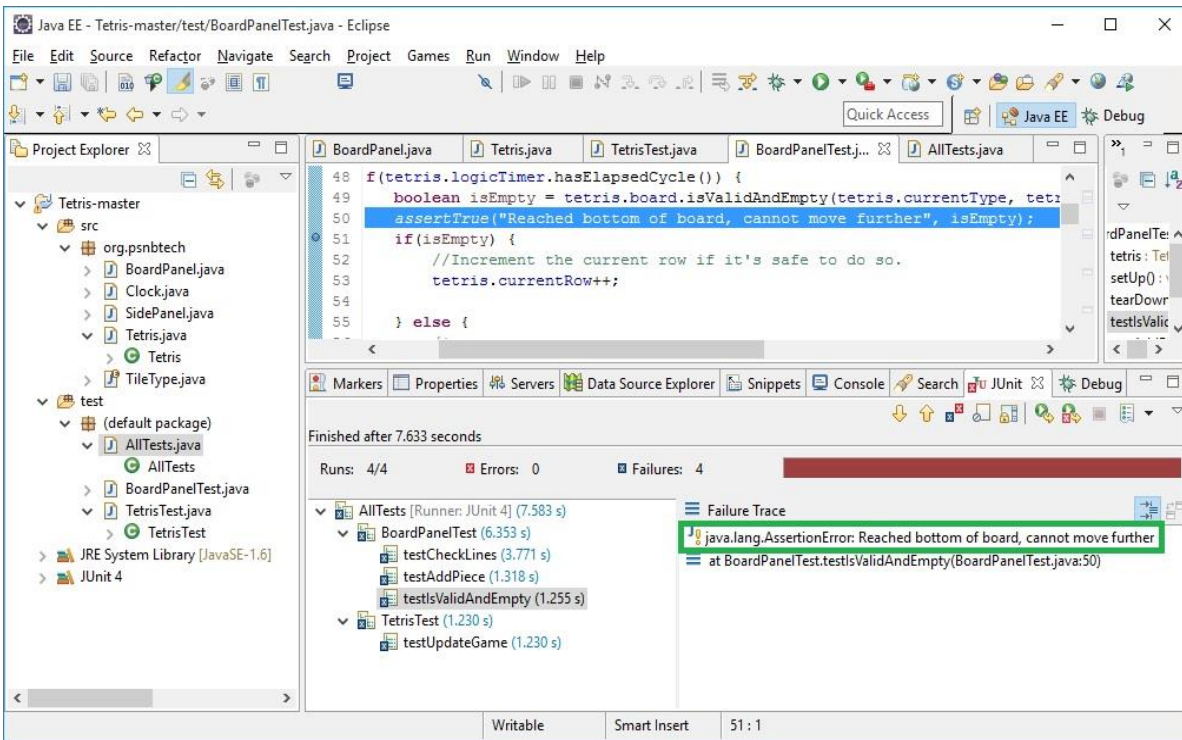
4. Include a screenshot of the result of running your test suite

I've created a test suite which includes all the test cases that I've created. The results of the test cases are as below...

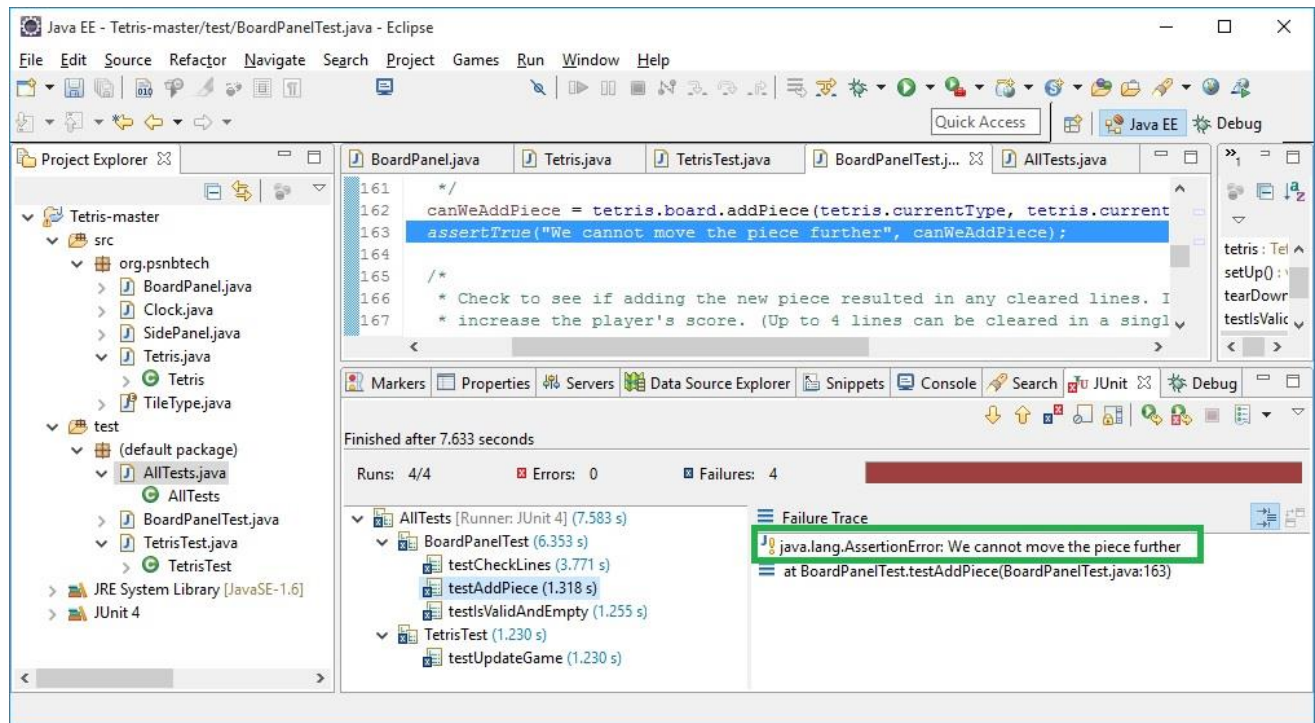
4.1 Test Case result of *updateGame()* method:



4.2 Test case result of *isValidAndEmpty()* method:



4.3 Test case result of *addPiece()* method:



4.4 Test case result of *checkLines()* method:

