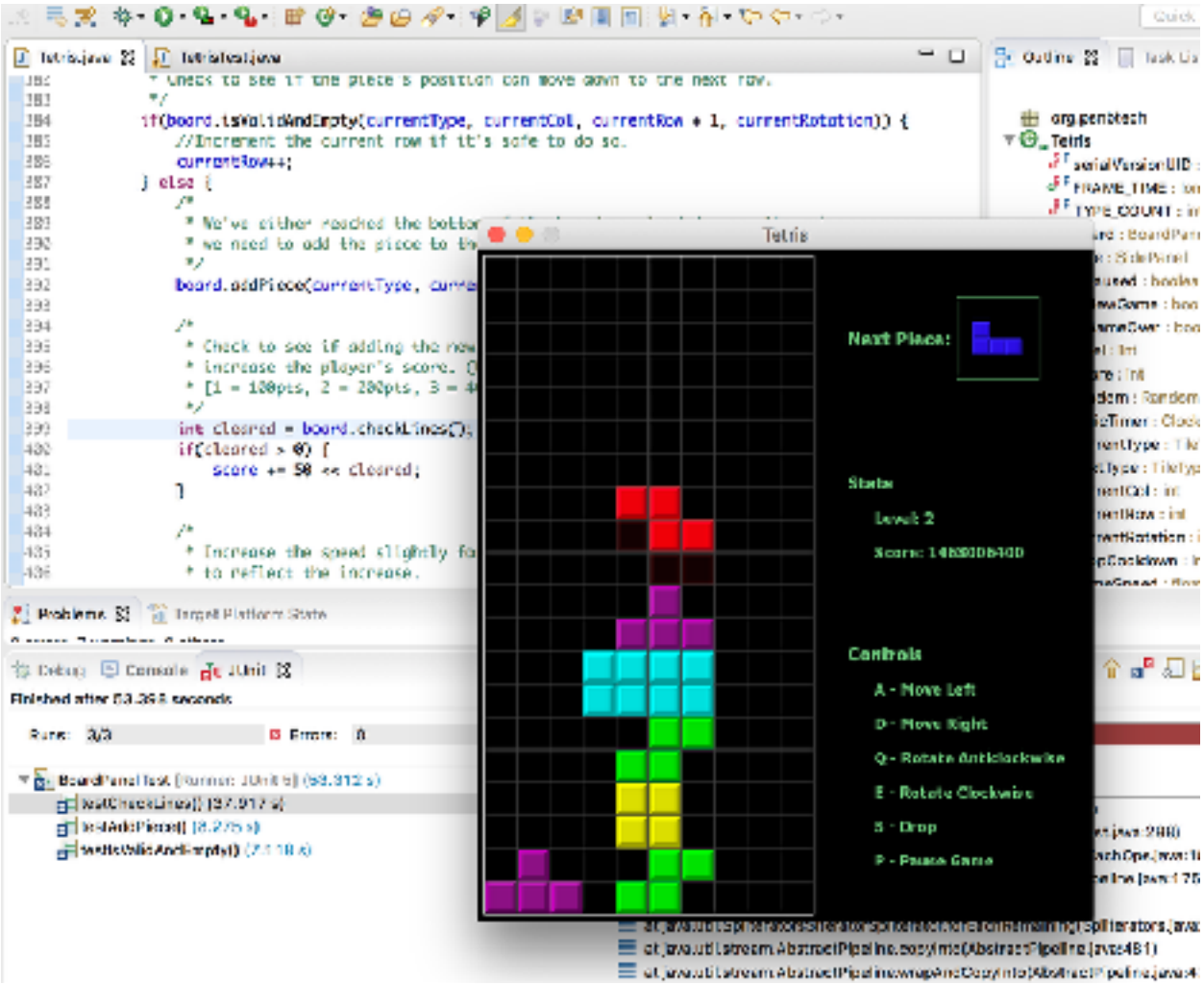
Software Methods and Tools
Spring 2018 Assignment 6 – JUnit Testing

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

While testing this application, I found that the score is negative and in random. I played the game and saw that the score increases with any tile drop, didn't wait for the line to complete. This meant that the code had bugs. I found out more while testing the *updateGame()* method of *Tetris* class and *checkLines()* method of *BoardPanel* class.

Here we can see that the score changes abnormally, not proportional to the game or not based on winning or putting the tiles together.



TetrisTest.java

This file tests the *updateGame()* method of *Tetris* class.

BoardPanelTest.java

This file tests *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?

The updateGame method's return type is void. So, for testing this method, I had to change 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:

$$score = score + 50*2^{22}$$

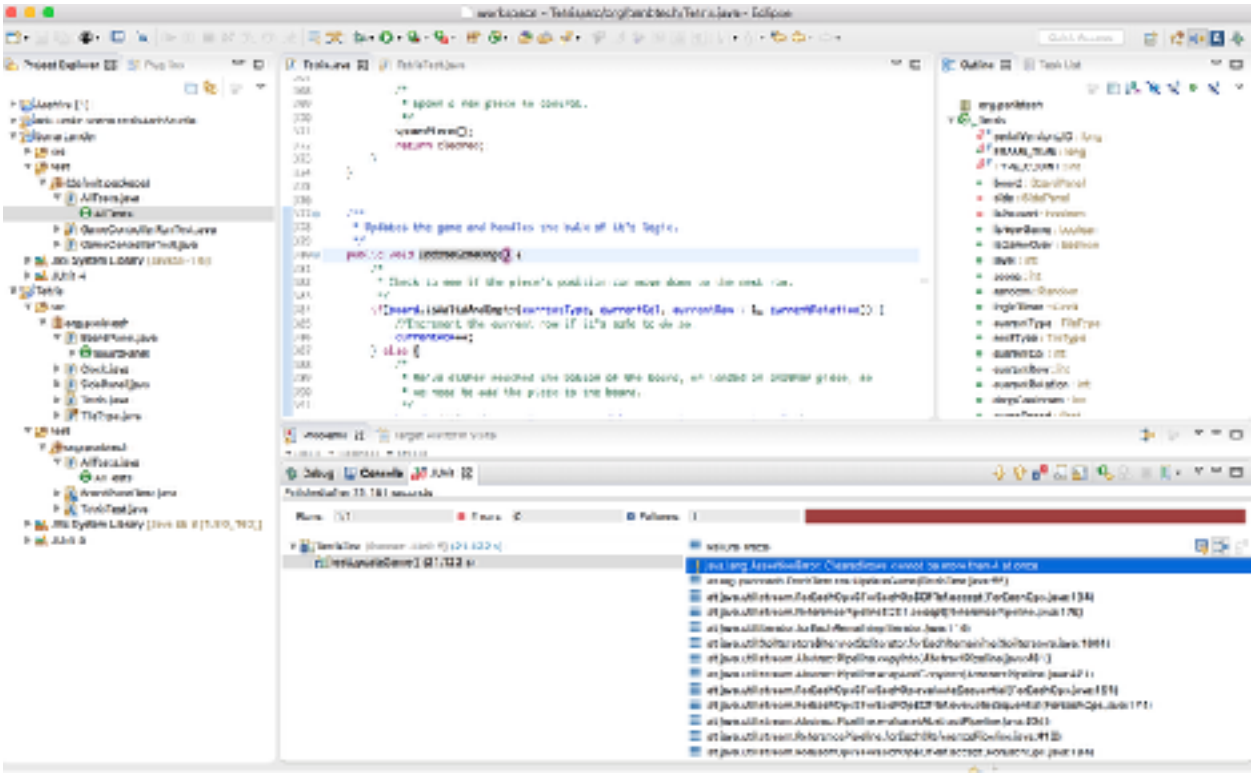
To identify this bug, I've created a test case for *updateGame()* method which check the validity of number of cleared rows. Here, once the tile reached the bottom of the board, even if the row is not full, we get 22 (which is total height of board) as the cleared rows. My test case checks for count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

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

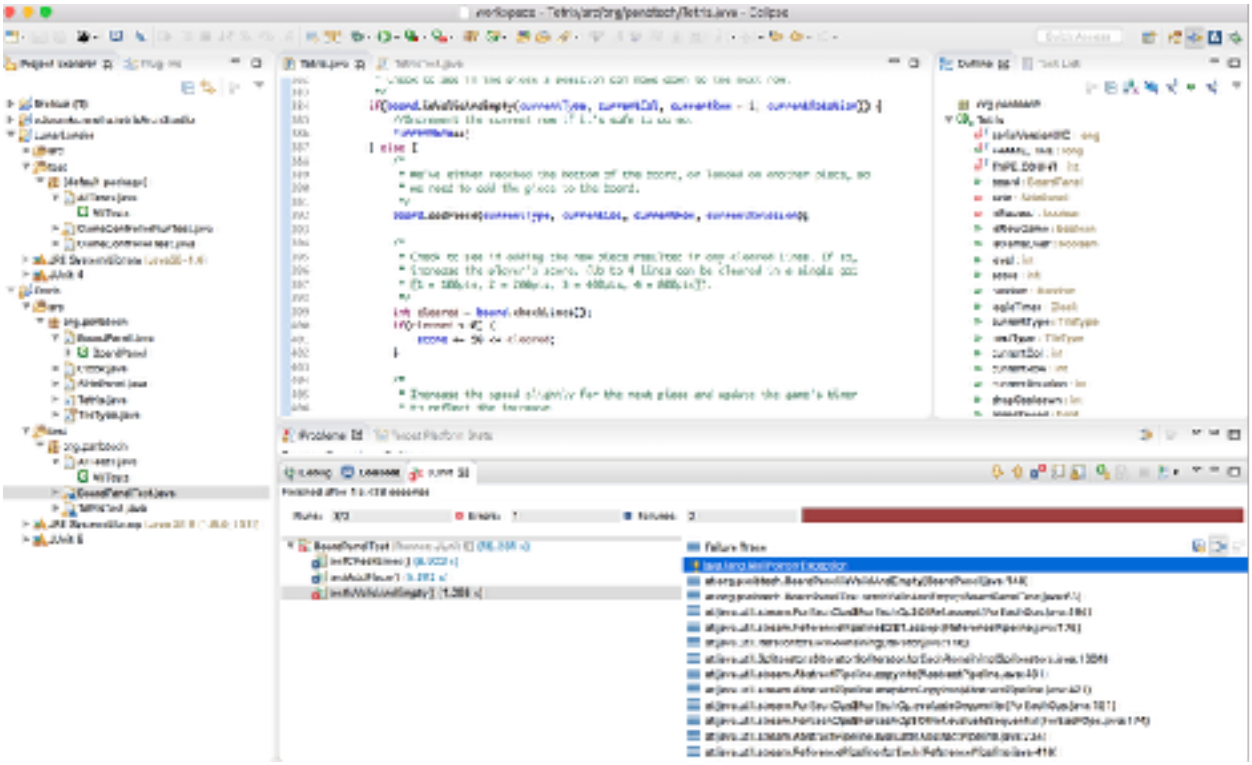
3. Include a screenshot of the result of running your test suite

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

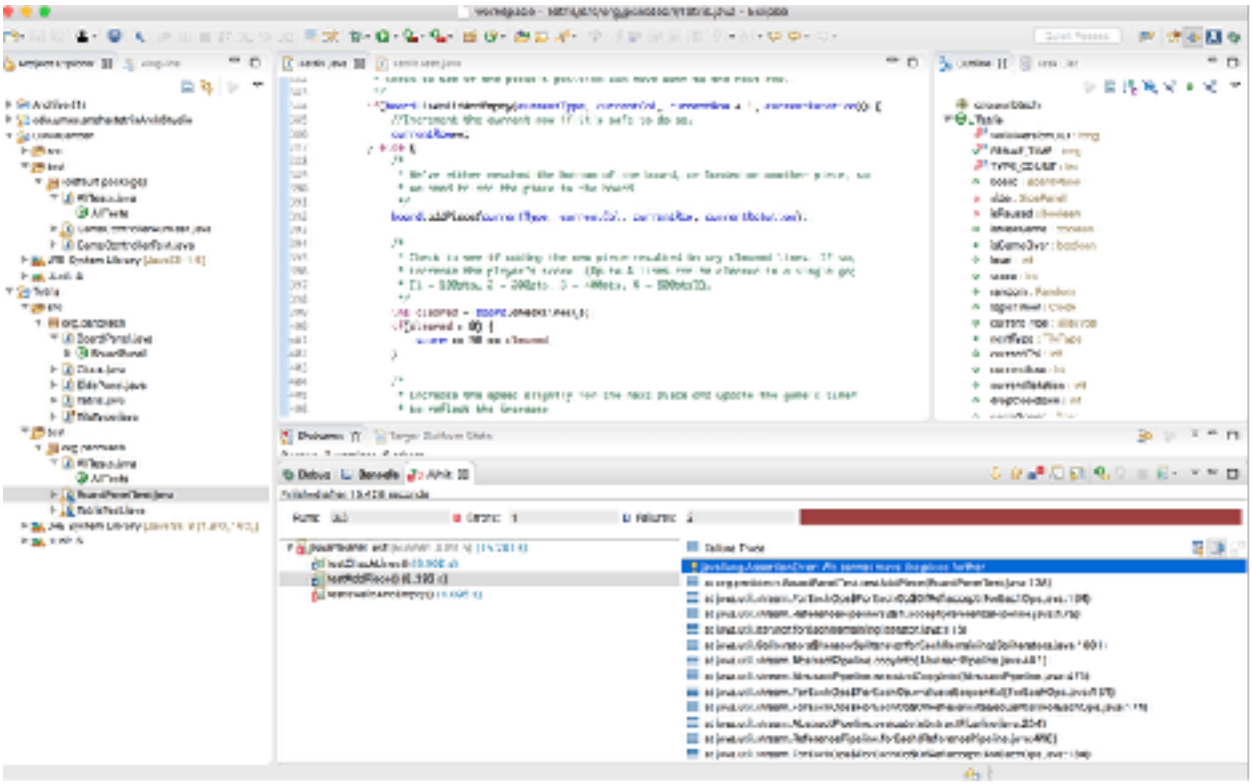
3.1 Test Case for *updateGame()* method:



3.2 Test case for *isValidAndEmpty()* method:



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



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