



**SOEN 6011- Software Engineering Processes
(Summer 2016)
SmarTech-Group 10**

**Assignment 4 – Test Report for JUnit Tests
On
“Tic-Tac-Toe”**

Submitted by:

Amritpal Singh (27684878)
Amandeep Sharma (27260164)
Bhawna Sharma (27568789)
Deepinder Singh (40002787)
Hardilpreet Singh (27822200)
Jatinderpal Singh (27727267)
Manvir Singh (27680120)
Vijay Shah (27735146)

Submitted to:

Dr. Nicolangelo Piccirilli

Submission Date: May 30, 2016

Table of Contents

1. Purpose.....	3
2. Test Cases	3
2.1 Test Case 1	3
2.2 Test Case 2.....	3
2.3 Test Case 3.....	4
3. Coverage Data.....	4
3.1 Deliverable 1	4
3.2 Deliverable 2.....	5
4. References.....	6

1. Purpose

Test Cases form a really important part of the document. If a software has to be developed, it is mandatory to have the test cases written for them. In Software Engineering, the test cases serve as a document which consists of all the conditions under which a tester would determine if the developed software satisfies its initial purpose. In other words, a test cases document would comprise of a set of test data, expected results, preconditions and post conditions for a particular scenario.

2. Test Cases

Project Name: Tic-Tac-Toe

2.1 Test Case 1

Class	BoardTest.java
Test Type	Junit Test

Test Methods:	Test Results:
testGetTurnX()	PASS
testGetTurnO()	PASS
testGetSymbolX()	PASS
testGetSymbolO()	PASS

2.2 Test Case 2

Class	GameTest.java
Test Type	Junit Test

Test Methods:	Test Results:
testFlag()	PASS
testButtonSet()	PASS

2.3 Test Case 3

Class	PlayerDetailsTest.java
Test Type	Junit Test

Test Methods:	Test Results:
testIsPlayer1Null()	PASS
testIsPlayer2Null()	PASS

3. Coverage Data

EclEmma is a plug-in that provides code coverage information report and gives trace information related to test cases. The reason why we have used EclEmma is that although the existing test cases verify the correctness of our code, there is still no guarantee that the entire base code is actually tested. Hence tools such as EclEmma determine the portion of the code that was actually tested by the test cases along with the untested portion of the code.

3.1 Deliverable 1

In deliverable 1, we have used EclEmma coverage tool in order to collect coverage information. Following figure shows all the coverage information for deliverable 1 which contains coverage, covered instructions, missed instruction and total instruction.

The screenshot shows an IDE with a Java file named `Game.java` containing the following code:

```

30  setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
31  final JFrame playerDetails=new JFrame("Details");
32  playerDetails.getContentPane().setLayout(null);
33  playerDetails.setSize(400, 400);
34
35  JLabel lblName = new JLabel("Player 1");
36  lblName.setForeground(Color.BLACK);
37  lblName.setFont(new Font("Lucida Handwriting", Font.BOLD, 20));
38  lblName.setBounds(10, 47, 121, 34);
39  playerDetails.getContentPane().add(lblName);
40  setResizable(false);
41
42  /*
43   * Enter Player 1 name
44   */
45  txtPalyername1 = new JTextField();
46  txtPalyername1.setText("Player 1");
47  txtPalyername1.setBounds(189, 57, 139, 20);
48  playerDetails.getContentPane().add(txtPalyername1);
49  txtPalyername1.setColumns(10);
50

```

Below the code editor, a coverage report table is displayed:

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
tictactoe-1	23.1 %	245	817	1,062
src	23.1 %	245	817	1,062
(default package)	23.1 %	245	817	1,062
PlayerDetails.java	0.0 %	0	373	373
HomePage.java	0.0 %	0	198	198
Help.java	0.0 %	0	127	127
Game.java	63.8 %	201	114	315
TestPlayerDetails.java	0.0 %	0	5	5
TestGame.java	100.0 %	44	0	44

3.2 Deliverable 2

In deliverable 2, we have used Android Studio in order to generate coverage report. There is an inbuilt option Generate Coverage Report in analyze menu.

The screenshot shows the Android Studio interface with a coverage report for `boardTest and 1 more`. The report indicates that 3% of classes and 3% of lines are covered in all classes in scope.

Element	Class, %	Method, %	Line, %
android	0% (0/31)	0% (0/3)	0% (0/108)
anim			
anim-v21			
animator			
assets			
color			
color-v11			
color-v23			
com	7% (2/27)	14% (6/42)	4% (13/324)
dalvik			
drawable			
drawable-en-hdpi			
drawable-en-ldpi			
drawable-en-mdpi			
drawable-hdpi			
drawable-hdpi-v4			

4. References

- [1] Ableson, W. Frank., Sen, Robi.,King, Chris.,. *Android in Action* 2011.
- [2] J. Friesen . (). *Learn Java for Android development*.
- [3] B. J. MacDonald . *Programming the Finite Element Method in Java and Android* 2013.
- [4] Z. R. Mednieks . *Programming Android* 2012.
- [5] Pressman Toy Corporation. Tic tac toe. 1978.