

Table of Contents

| | |
|---|----------|
| Bug 1: Incorrect Payout on win | 2 |
| BUG 2: Player cannot reach betting limit | 2 |
| Bug 3: Game overall odds | 2 |
| Bug 4: Dice seem to always roll the same after first roll..... | 3 |
| Bug 5: "SPADE" doesn't appear in roll. | 3 |

Bug 1: Incorrect Payout on win

Following is the screenshot of the output of the automated test (unit test) that has passed after fixing the bug. The test was check against all the zero, one , two and three matched condition.

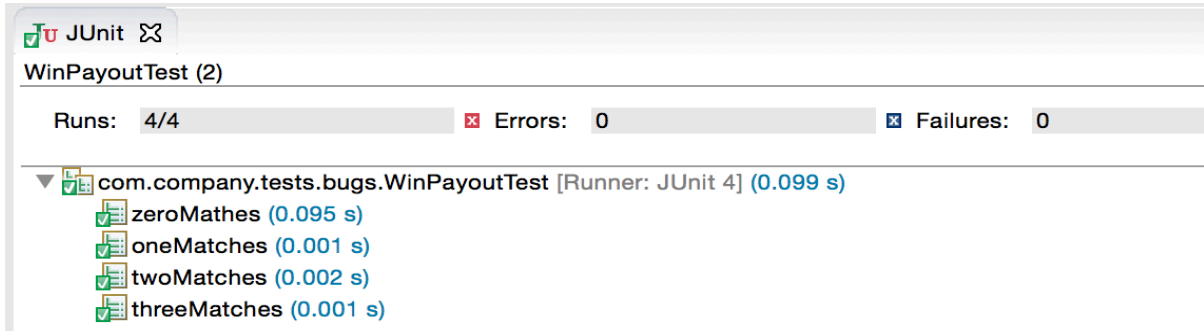


Fig: Screenshot showing the automated test has passed

All the four tests have passed after fixing the bug. There is 0 failure.

BUG 2: Player cannot reach betting limit

Following is the screenshot of the output of the automated test (unit test) that has passed after fixing the bug. The test was carried out in the test pass condition that the player reaches 0 to end the game.

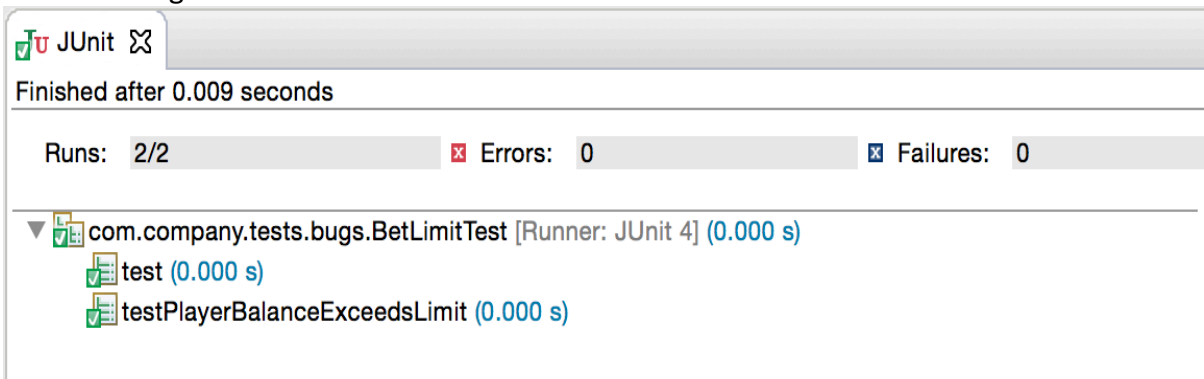


Fig: Screenshot showing the automated test has passed

Both the tests have passed after fixing the bug. There is 0 failure in the test.

Bug 3: Game overall odds

Following is the screenshot shows the automated test for overall game odds has passed after resolving the bug. The test was carried out to test the overall win ratio was between 0.42 to 0.45.

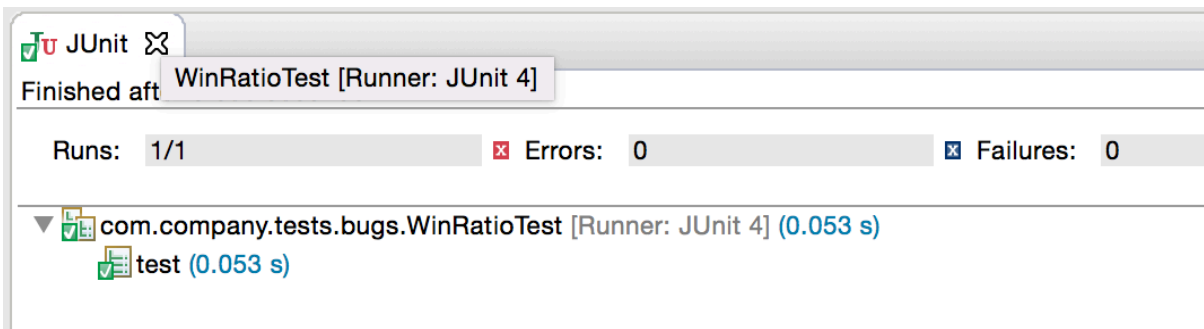
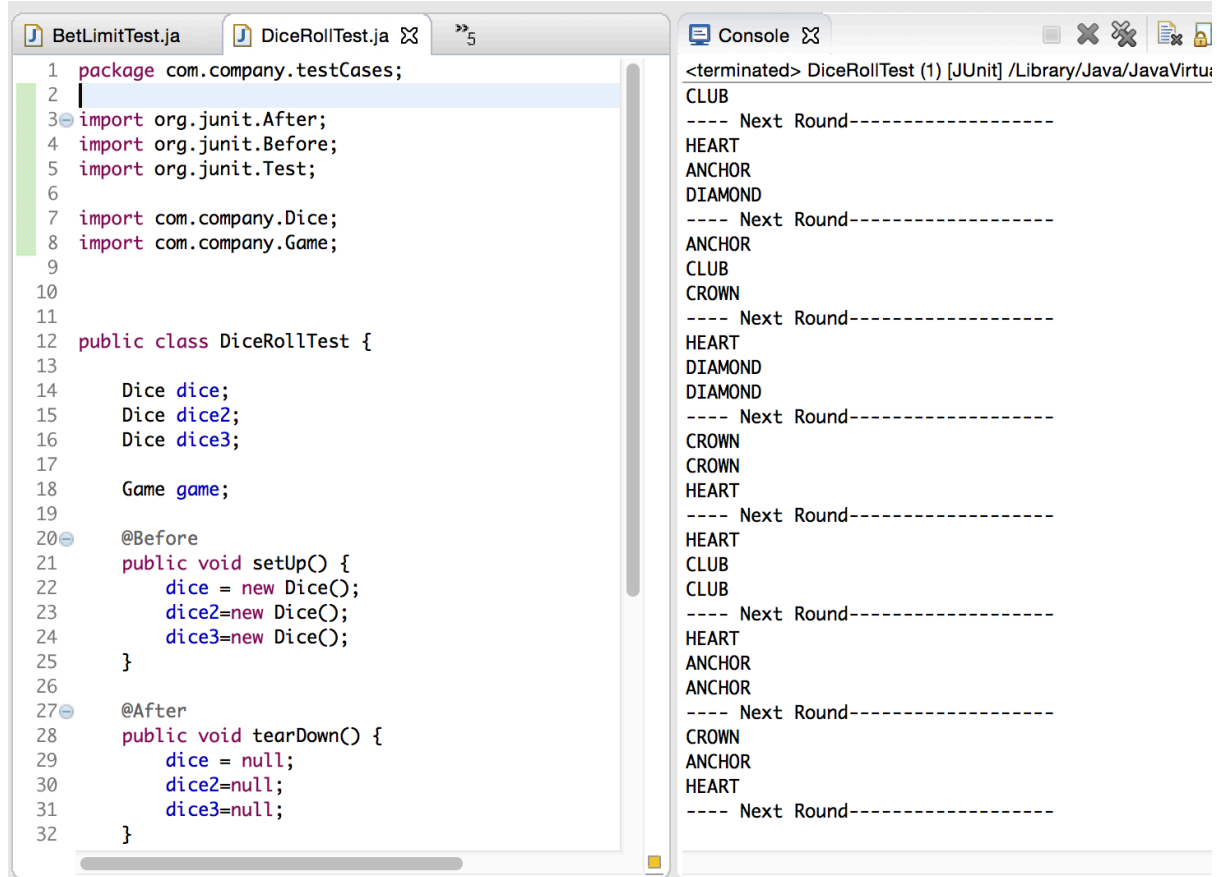


Fig: Screenshot showing the automated test has passed

Bug 4: Dice seem to always roll the same after first roll.

The following screenshot is the “system out” testing method. In this method the results are printed through the unit test and analyzed to validate the bug fix.



The screenshot shows an IDE with two tabs: `BetLimitTest.java` and `DiceRollTest.java`. The `DiceRollTest.java` tab is active, displaying the following code:

```
1 package com.company.testCases;
2
3 import org.junit.After;
4 import org.junit.Before;
5 import org.junit.Test;
6
7 import com.company.Dice;
8 import com.company.Game;
9
10
11
12 public class DiceRollTest {
13
14     Dice dice;
15     Dice dice2;
16     Dice dice3;
17
18     Game game;
19
20     @Before
21     public void setUp() {
22         dice = new Dice();
23         dice2 = new Dice();
24         dice3 = new Dice();
25     }
26
27     @After
28     public void tearDown() {
29         dice = null;
30         dice2 = null;
31         dice3 = null;
32     }
33 }
```

The console output on the right shows the results of the test:

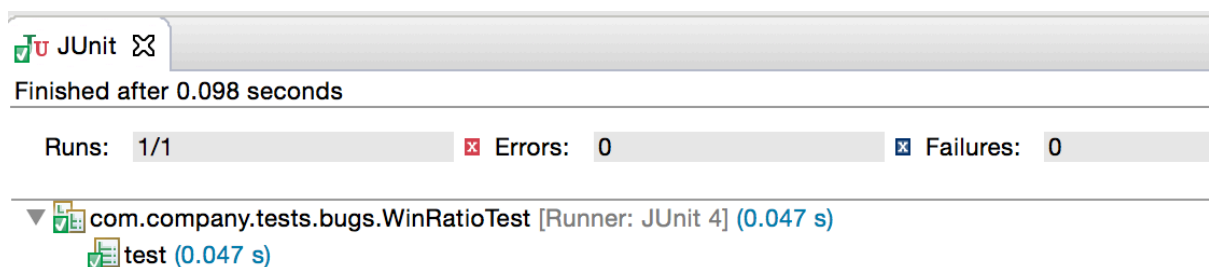
```
<terminated> DiceRollTest (1) [JUnit] /Library/Java/JavaVirtu:
CLUB
---- Next Round-----
HEART
ANCHOR
DIAMOND
---- Next Round-----
ANCHOR
CLUB
CROWN
---- Next Round-----
HEART
DIAMOND
DIAMOND
---- Next Round-----
CROWN
CROWN
HEART
---- Next Round-----
HEART
CLUB
CLUB
---- Next Round-----
HEART
ANCHOR
ANCHOR
---- Next Round-----
CROWN
ANCHOR
HEART
---- Next Round-----
```

Fig: Screenshot showing the automated test has passed

In the above screenshot, the result shows the different value of dice roll for different rounds of play which means the test has passed.

Bug 5: “SPADE” doesn’t appear in roll.

Following is the screenshot of the automated test that was testing the “spade” not showing in the dice roll. The screenshot shows that the automated test has passed after the bug was fixed, hence no failure occurs in the test. The test was carried out to test if the “spade” comes out in any of the round while rolling the dice.



The screenshot shows the JUnit test results for the `WinRatioTest` class. The test finished after 0.098 seconds. The results are as follows:

| Runs | Errors | Failures |
|------|--------|----------|
| 1/1 | 0 | 0 |

The test results are displayed in a tree view:

- com.company.tests.bugs.WinRatioTest [Runner: JUnit 4] (0.047 s)
 - test (0.047 s)

Fig: Screenshot showing the automated test has passed

Automated test demonstrating correct operation after resolution of each bug
Bijaya Raj Basnet