| **Test Name** | | | Player is unable to win betting limit | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Tested:** | | | Crown and Anchor Game | | | |
| **Test Description:** | | | The betting limit of the game is set to 0, the game will terminate when the players balance is still 5 | | | |
| **Pre-conditions** | | | Default game setup | | | |
| **Post-conditions** | | | At the end of the game the players balance should be 5 | | | |
| **Notes:** | | **Because the game terminates if the players balance is 200, the only way to see the bug is to keep playing until the game has terminated on the balance of 5. A pass for the test will indicate that the bug does actually exist** | | | | |
| **Result (Pass/Fail/Warning/Incomplete)** | |  | | | | |
|  | **TEST STEP** | | | **EXPECTED TEST RESULTS** | P | F |
|  | Run program | | |  | X |  |
|  | Check the players balance once the game has finished | | | The players balance should be 5 | x |  |
|  |  | | |  |  |  |

# Test Result

Output from Game

Turn 114: Fred bet 5 on HEART

Rolled DIAMOND, DIAMOND, DIAMOND

Fred lost, balance now 10

Turn 115: Fred bet 5 on HEART

Rolled DIAMOND, DIAMOND, DIAMOND

Fred lost, balance now 5

115 turns later.

End Game 99: Fred now has balance 5

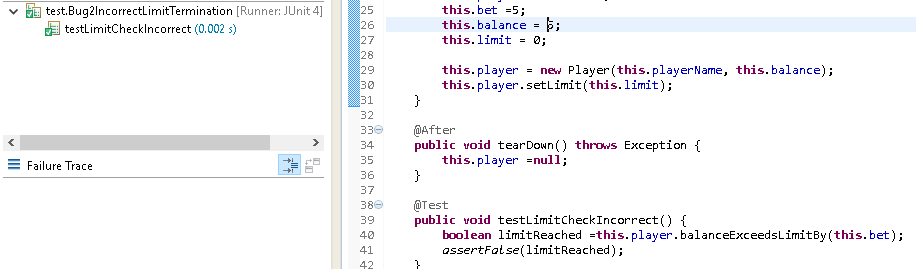
Win count = 1674, Lose Count = 6567, 0.20

Diagnose

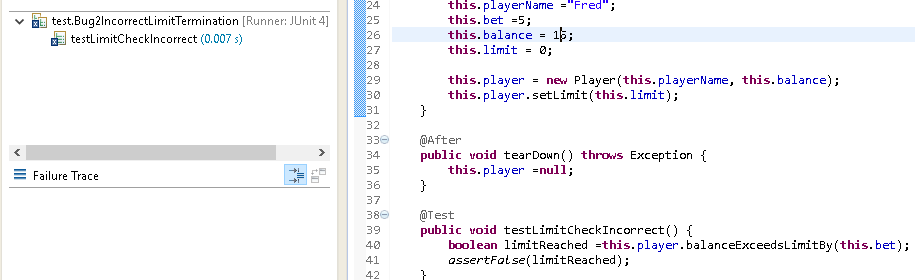
From the above output we can see that the game is terminating to early. The game appears to be stopping once the players balance is 5 and not 0. This seems to indicate that the termination code is off, and is terminating on limit +bet amount (0+5)

# Automated Test with Junit to show the bug

This bug is a little bit harder to stop with Junit testing. When you try to force the bug by dropping the balance to 5, the limit to 0 and the bet amount to 5, the balanceExceedsLimitBy check does actually return the correct result which indicates that this isn’t the bug. But the problem here is that from looking at the data in the UAT we can kind of says this is where the bug is. The issue is we are testing the function with the wrong values, which is making the first test invalid. The screenshot below shows the invalid test result which indicates that function is working correctly even though it isn’t.



Now if we change the balance to say 15 and run the same test we can see the function does actually fail. This is what we would expect from the UAT results.



The fail does indicate that the bug is due to the balanceExceedsLimitBy function. It doesn’t return false when it should. These two results we actually make it a lot easier to trace the bug down.