# Debugging the Crown and Anchor Game

***Bug 1: Game doesn’t pay out correct level***

**Hypothesis 1**: May be wining is not calculated properly.

**Experiment:** Inspecting through the classes, I have come to understand the Game class have playRound method which returns the winning, according to matches against user pick. May be the method is not returning right winning amount.

**Test Output:**

There are four conditions to test, in order to ensure that the method doesn’t have bug.

Winning is calculated through bet multiplied by matched numbers.

If there is no match between user picked symbols and system selected symbols then bet multiplied zero which will be zero

If there is one match then bet multiplied by 1

If there is two match then bet multiplied by 2

If all, three match then bet multiplied by 3

Below is the test screenshot from JUnit test, testing these conditions

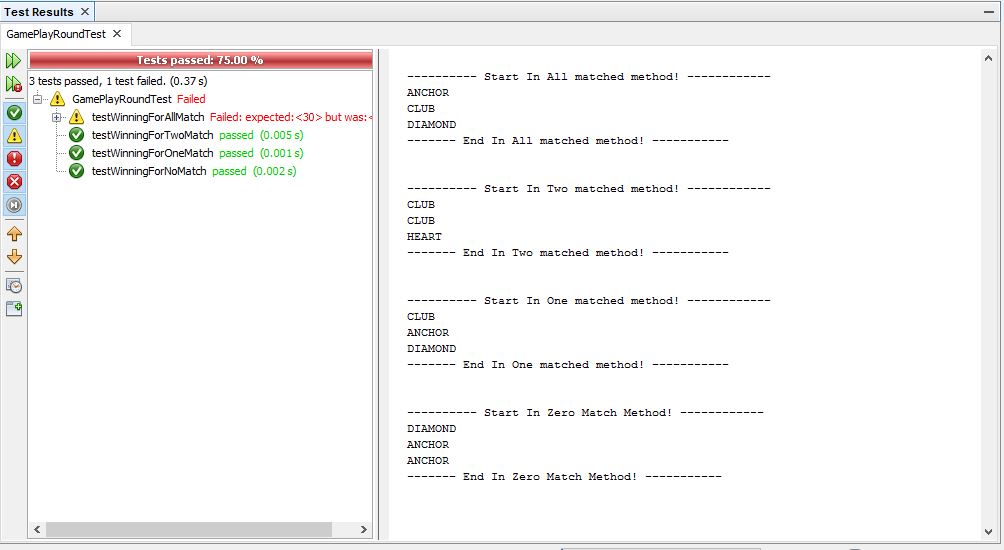


Figure : Game class, playRound method test screenshot

As we can see, I had picked and hard coded Club as a dice value, and we can see that, in two matches the winning is same as the expected result, which results in test being passed, and similar cases in one match and zero match, but in all match since there is only one match, which is only one club, that’s why test failed.

**Conclusion:** Since methods are returning correct value, there is no error in this method, thus the hypothesis is wrong.

**Hypothesis 2:** Maybe bet taken from user is not added back to user balance

**Experiment:** While inspecting the playRound method, in Game class, I have noticed that at first bet is being deducted from the user balance and then user picked symbol is matched against the system selected symbols, but if the user wins the match the bet amount is not added to the user balance only returning winning amount. May be there is the error.

**Test Output:**

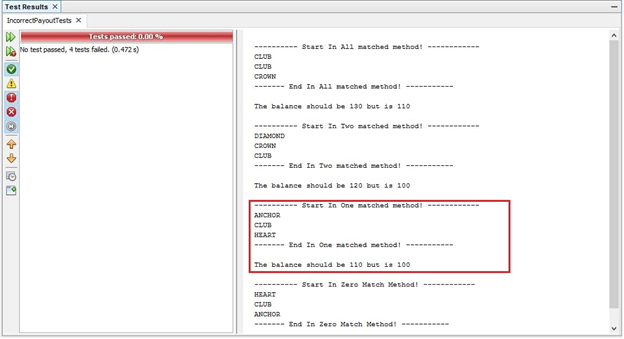


Figure : Check player balance after payout test

As we can see in the one match, being matched club exactly one times, the user initial balance was 100 which means after winning the match, user should have 110 but instead it is 100, which means bet taken from the user is not returned to the user.

**Conclusion:**

Hence, I can conclude that the error is in Game class, playRound method, in which user bet is taken from the user balance but upon winning the bet is the added to the balance, and this is the state where the sane to infected transition occurs.

**Bug 2: Player cannot reach betting limit: Limit set to 0, but game ends with player still with 5 (dollars) remaining.**

**Hypothesis:** Maybe betting limit is incorrectly compared against the player balance.

**Experiment:** Balance should have 0 at the end of the match but it exits early, while still having left 5, which is incorrect. While inspecting the Main class and Player class, balanceExceedsLimitBy, which deducts the bet amount from the balance and checks with limit and ensure that the player balance isn’t negative. May be the bug is incorrect use of comparison operator between player balance and limit.

**Test Output:**

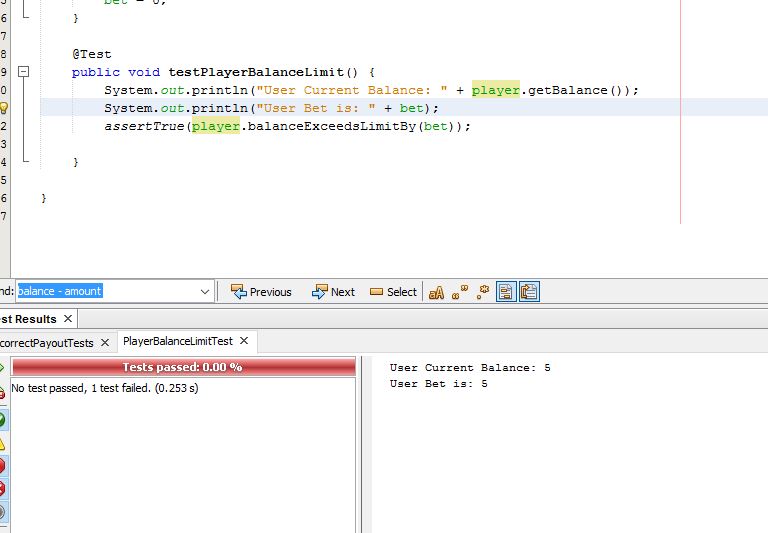


Figure : Player balance limit test

As we can see, even though the user balance is 5 and the user bet is 5, player should have got the last chance to play the game, but instead the method returns false, which means user cannot play game and user balance is 0 or less than that, which is not the case.

**Conclusion:**

Hence, the hypothesis is correct and the program transition from sane to infectious state, with incorrect comparison between balance and limit.