Hypothesis & Testing – Bug 04

# Bug Reported

*“Every game the die are rolled, every round the die are the same as the first roll. The game requires that every round has a new roll in order to be fair”*

Contents

[Bug Reported 1](#_Toc432782249)

[Analysis 2](#_Toc432782250)

[Localising the bug 3](#_Toc432782251)

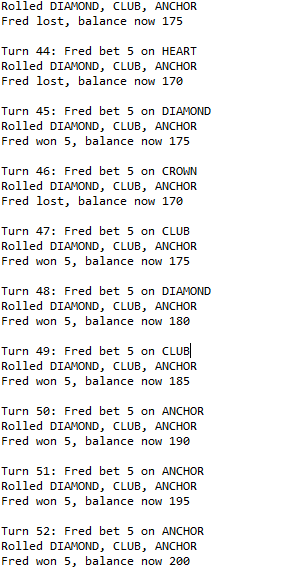
[Hypothesis 5](#_Toc432782252)

[Results 6](#_Toc432782253)

[Conclusion 7](#_Toc432782254)

# Analysis

Initial inspection of program reveals similar behaviour as to what was described in the bug report. The rolls were the same for every turn.



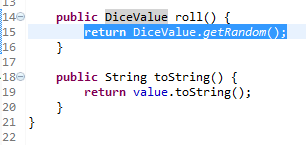
Rolls are the same for every turn

Additional runs reveal the same results, the bug is consistent throughout all runs.

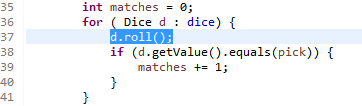
# Localising the bug

The bug is related to calculating the dice rolls and determining die-roll values. Closer inspection reveals that only Dice, DiceValue and Game.java should be inspected.

Code analysis reveals that the bug is likely within Game.java. Further code analysis reveals that I was incorrect, the responsible method was being called from Game.java. The responsible method in Dice.java is likely to produce the bug:

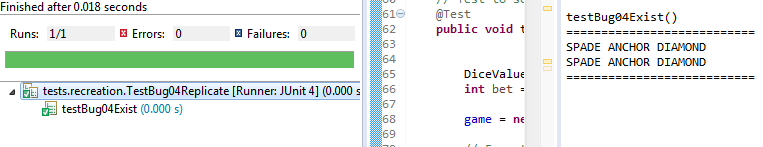


The above code should be responsible for creating the bug.



The code above is how this method is called upon from the Game.java

Automated test reveals that the bug is created in this method. See Bug04Replicate.java for test.



The above image is testing for if the bug exists, the bugs exist, so the tests pass. The above example is also proof of localising and recreating the bugs.

When testing if the code is fixed (separate test), if a bug exists, the tests will not pass.

# Hypothesis

An error in the code is creating the bug. In Game.java, the process is as follows for roll method:

1. Method calls upon DiceValue.getRandom()
2. **Method returns the DiceValue obtained from call**

Step 2 is the suspected cause of the bug, the value of the dice should not be ‘returned’, it should instead be ‘set’. Changing it to set, should re-roll the die every time a round is player (every time it is called from round method).

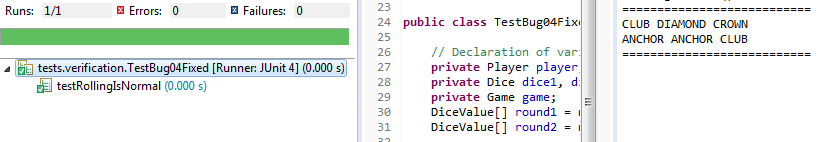
Replication has revealed that when the value is ‘returned’ the die is not updated and nothing will handle the returned DiceValue. If it updated each dies value, which is what SHOULD happen, each die will re-roll every turn.

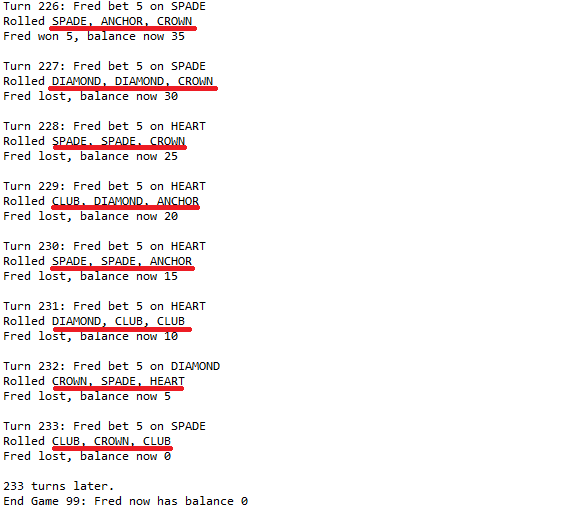
The proposed solution is to alter Dice.java, changing the ‘return’ to a ‘value =’ once this solution has been implemented, the dice should re-roll properly.

# Results

My hypothesis was correct. The cause of the bug was that the return statement, as it did not update the dies value, and nothing was implemented to handle the return of a DiceValue. The die should re-roll every round.

Testing had confirmed both that the bug exists and can be replicated, and that it is now removed from the fixed code. Test results below:





# Conclusion

* My hypothesis was correct.
* The cause of the bug was that the return statement, as it did not update the dies value, and nothing was implemented to handle the return of a DiceValue
* Bug has been tested and fixed.