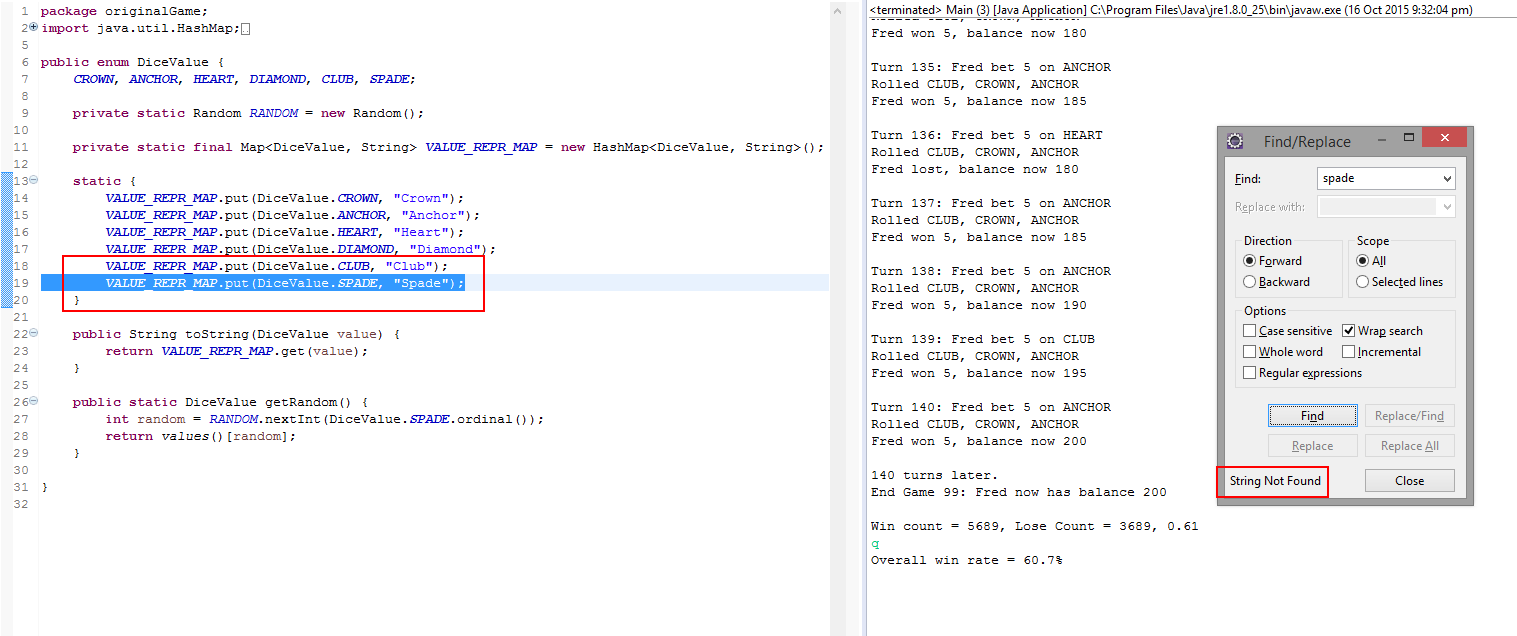
# Bug 3 Hypothesis:

Hypothesis 1:   
The Spade diceValue is not being chosen within the game run-throughs and this is negatively effecting the probability of the game.

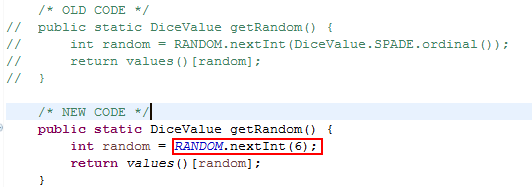
### Problem 1: The diceValue spade is never chosen when the game is ran. This problem greatly effects the odds of the game.



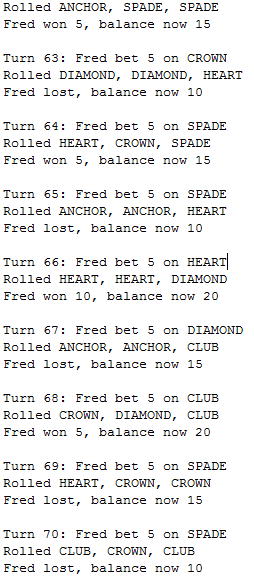
The “Spade” diceValue was not selected by the game, or chosen by the user in 9378 games. That is enough evidence to prove that the Spade diceValue is not being chosen.

Suggested Fix:

Including the SPADE index within the DiceValue getRandom() method. This involves entering the integer “6” into the nextInt command, to ensure the program is retrieving all diceValues up to, and including the index 6.



Evidence after Hypothesis 1:



The results are coming back with the inclusion of SPADE, however the winning rates are still fluctuating and not matching our desired win/lose ratio, mentioned in the Bug Report.

Examples:





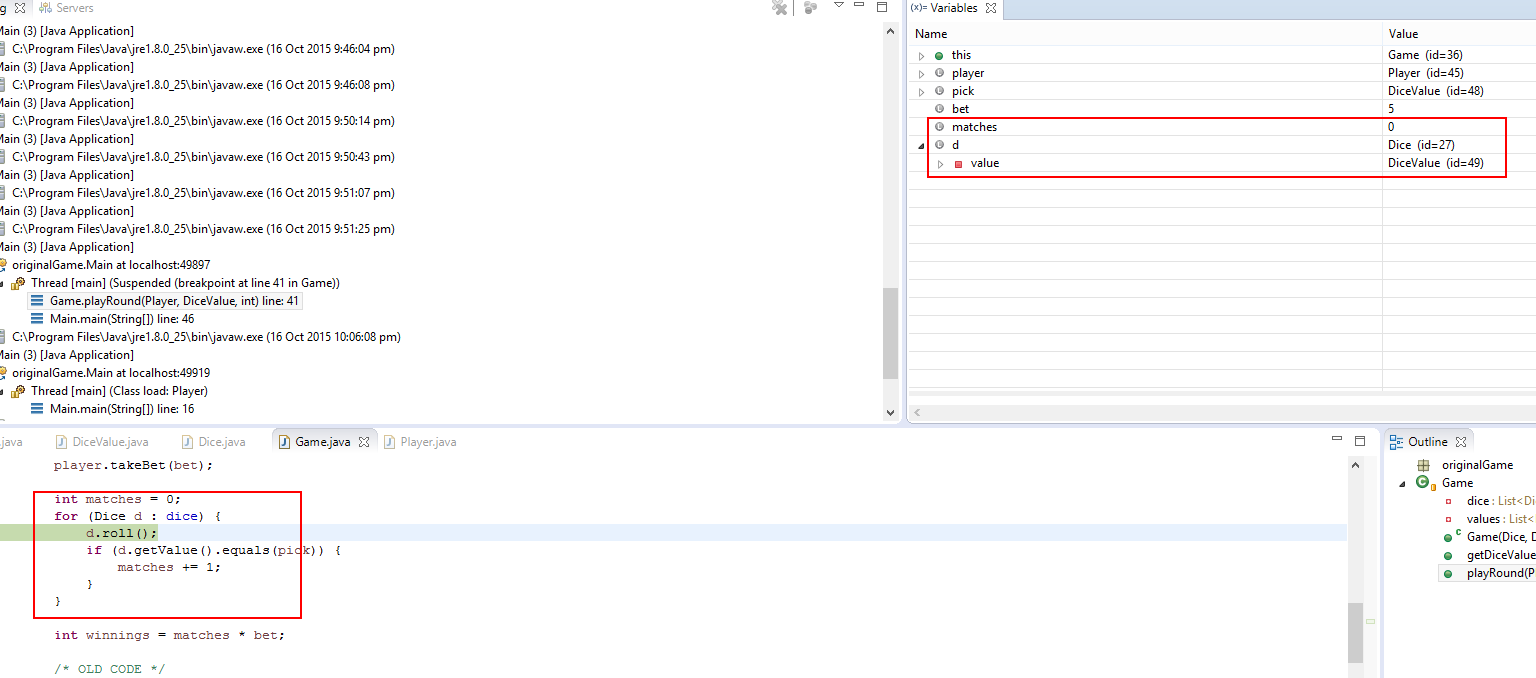




### Hypothesis 2:

Even though, the SPADE issue has been fixed, there is still a problem with the win/lose ratio with the game. The dice rolls of the game are directly relational to the probability of the game, therefore the bug regarding the odds are relational to the rolling of the dice.

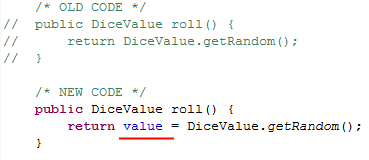
The most suspicious method for the origin of this value lies in lines 40-44 of Game.java. While placing a breakpoint at line 41, we can effectively detect what is happening to the d.Roll value.



From this evidence, we can see the diceValue being returned, however we cannot see the roll value being returned.

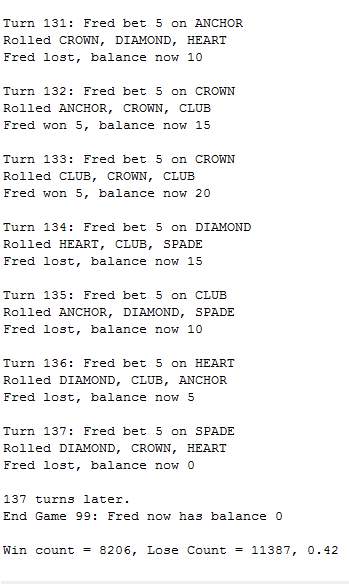
Quick Fix:

Adding a return value to line 15 in Dice.java, like so:



# Bug 3 Resolution:

### Evidence of Resolved Bug



We are now getting a constant win/lose ratio of 42%.

| **Test Name** | | Odds Bug 3 |
| --- | --- | --- |
| **Use Case Tested:** | | Betting Limit |
| **Test Description:** | | User may now observe the odds have been fixed and calculating at a constant 42% each game. |
| **Pre-conditions** | | * Console Exists * Game Exists * 3 Dice Exist * Player Exists Player has declared their bet Player’s Balance exceeds their bet declaration * totalWins, totalLosses, winCount, loseCount and turn are initialized to Zero |
| **Post-conditions** | | * Game has ended * The win/lose ratio should be at, or around 0.42. Every run-through of the program should achieve a similar result. |
| **Notes:** | **Checking if bug 3 has been resolved.** | |
| **Result (Pass/Fail/Warning/Incomplete)** | **Pass – bug 3 resolved** | |

|  | **TEST STEP** | **EXPECTED TEST RESULTS** | P | F |
| --- | --- | --- | --- | --- |
|  | Run the Game. | The program executes and can be seen in the console. | X |  |
|  | User observes the win ratio after program completion. | At the bottom of the output, the player can see the win/lose ratio of 42%. | X |  |