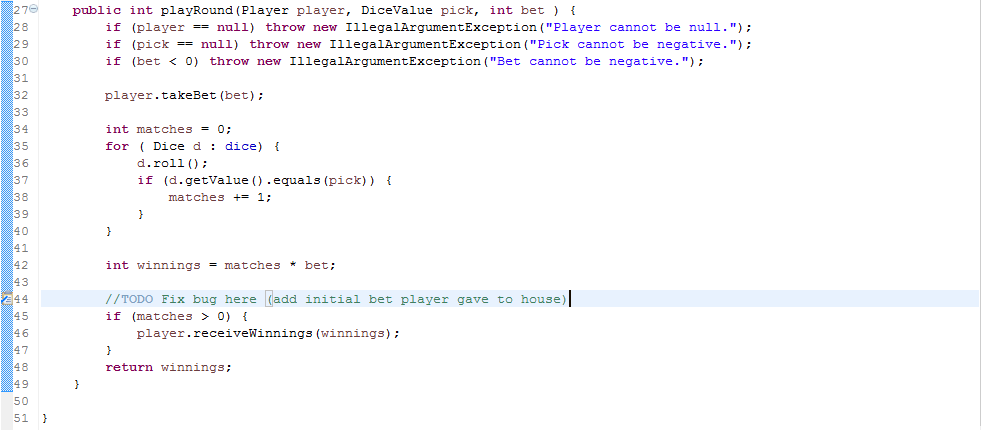
Debugging Log

Bug 1: Balance Winnings Update Bug

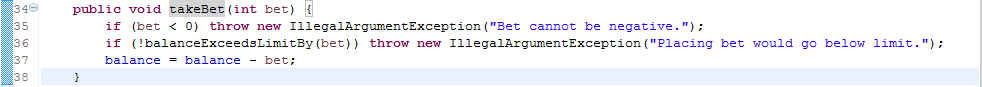
Hypothesis: Mathematical error (Logic error) where the program doesn’t give the user their initial bet back + the winnings.

Sequence:

Assume winnings calculated in Game.java and looked at the playRound method.

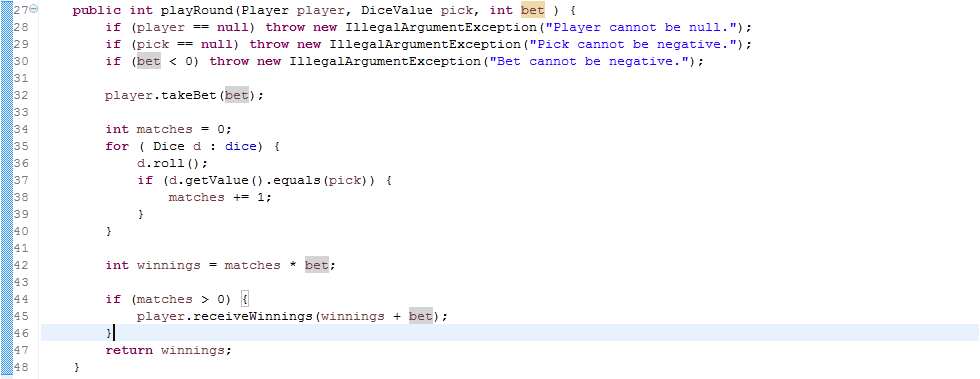


Found that the program was only giving what the user would have won without getting their bet back.



In player.java the user does indeed deposit the initial bet. So if the user wins they need to have gotten the bet back.

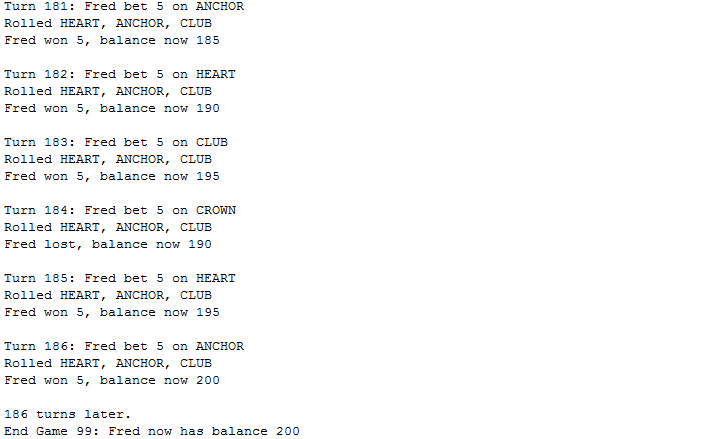
Fix:



Added the initial bet to the player.receiveWinnings method call to accommodate it.

Running a test

Shows the balance updating correctly.



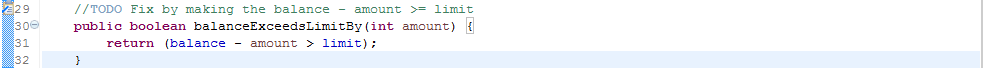
Bug 2: Limit Not Reached Bug

Hypothesis: A flag must have the wrong logical operator when checking if the limit has been reached (i.e. logic error).

Sequence:

In line 40 in the Main.java class the program uses a method called balanceExceedsLimitBy() which is in the Player.java class.

The method is:

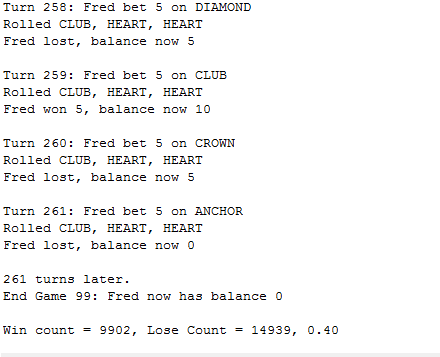


The problem here is that it only checks if the balance is above the limit and not when the balance is equal to the limit (which is what a limit is). A quick fix to this would be to make the operator >=

Fix:



Running a test:



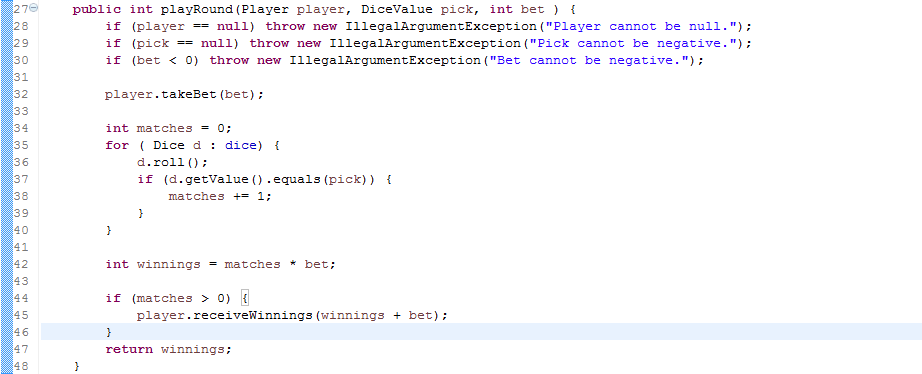
This shows the limit is being reached before exiting the game. Fixed.

Bug 3: Win Ratio Odds Incorrect Bug

Hypothesis: The luck mechanic in the game is not correct, so the bug is either in the Game.java class or the Dice.java class

Sequence:

Started by looking at the game.java class for anything that would denote dice rolls etc. Looked at the playRound() method.



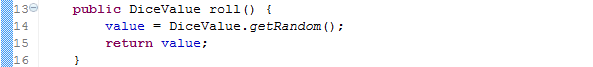
The roll() method for the dice is where the basis of luck is done.

The roll() method in Dice.java



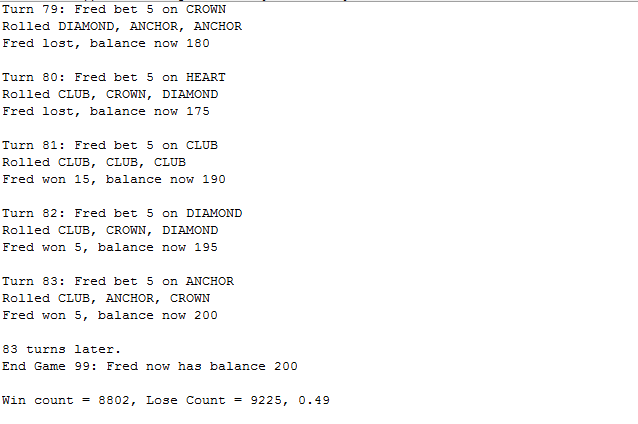
The Dice’s value isn’t actually being updated, so every game the rolls are the same, which is not how the game works.

Fix:



This will overwrite the previous dice value.

Running the tests



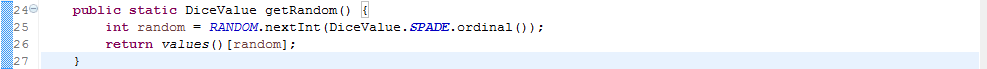
Instead of a consistent 0.42 it returns a consistent 0.49. Which is not what we want. Upon further Investigation, another bug was found. Detailed Under.

Bug 4: Spade Never Rolls Bug

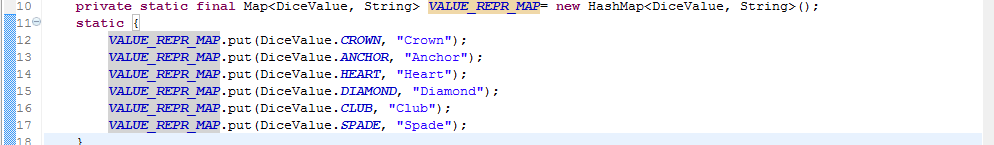
Hypothesis: Logic error when picking dice values at random (DiceValue.getRandom());.

Sequence:

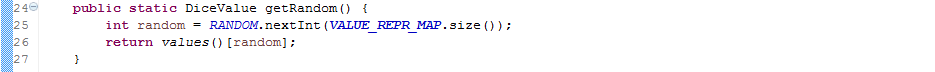
Checked the values and found that with the way the code works it skips SPADE because it increments from it.



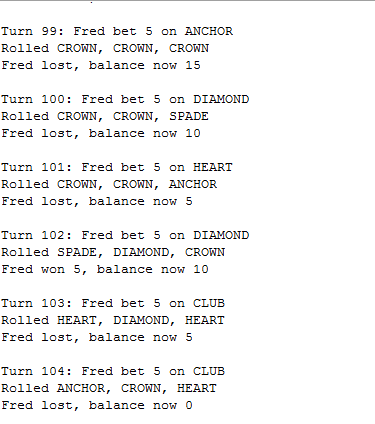
A way to fix this would be to use a value that goes through all the dice values rather than starting from after SPADE. A good solution would be to use the size of the collection of values, which can be taken from the VALUE\_REPR\_MAP.



Fix:



Running the tests:



This also fixed the third bug (Win ratio of 0.42). (Output shown below).

