Debugging Journal

ITC205 ASSESSMENT ITEM 4

Dissanayake, Shane

11516600

16-oCT-2015

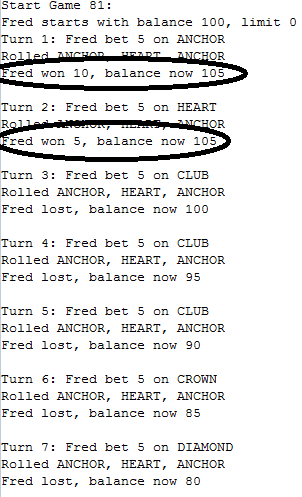
2015

# **Bug One**

## Tracing

The first bug of the assessment shows that the player is receiving the incorrect amount when the player has won a bet.

**Output of Bug 01**



Each time the player wins a bet the balance does not increase, it stays the same balance from the previous game. Therefore their bet was not being returned correctly and neither is their balance.

## Hypothesis:

The game is not returning the correct balance each time the player wins.

## Prediction:

The player’s balance is not being returned correctly after winning a bet or the console is not displaying the right numbers.

## Test:

testTheory() class is used to test Bug01. 1 \* the bet is returned as the game rules stated.

## Resolution:

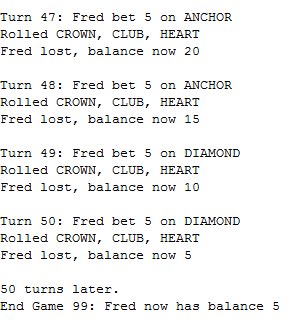
The game is now displaying the correct balance of the player after a winning bet. If matches are greater than 0 and the match is won, it will return the original bet and the receiving winnings.



*Balance now showing an increase when players wins a bet.*

# Bug 02

## Tracing



This output is not showing the end game limit of 0 when the game ends, instead it display a balance of 5. As this bug was found in the player balancer, it was assumed that the bug was within Player.java. balanceExceedsLimitBy() is the method returning the end balance – the amount was greater then the limit therefore it will always stop before the balance was equal to the limit of 0. An = sign was placed after the > sign.

## Hypothesis:

End game limit is not returning correctly. It has the balance being greater than the limit rather than greater than or equal to.

## Predictions:

The balance being greater than limit, not equals to.

## Test:

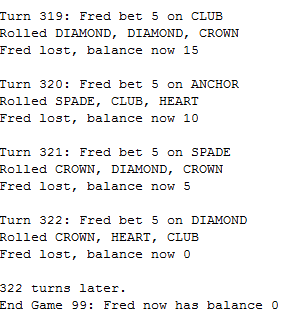
testTheory() has sets the player limit to 0, bet value to 5 and balance to 15.

It will loop until balance is at the limit or an exception

Then the final result will be displayed.

## Resolution:

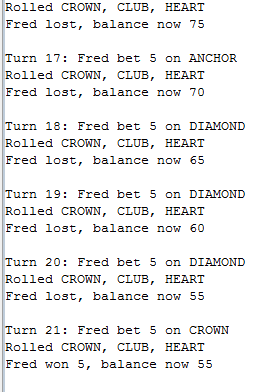
Balance ExceedsLimitBy() method was incorrectly comparing the limit and the balance. Therefore Hypothesis true.



*Output now shows end game balance at zero.*

# Bug 03

## Tracing



The game output is showing that every game the odds are changed because SPADE face is never showing at any point. As well as the rolls does not changed from CROWN, CLUB AND HEART.

To locate this bug it was assumed that the bug was something to do with the diceValue.java and dice.java since they were an obvious factor. While searching through diceValue.java It was learnt that the diceValue.getrandom method was not including the spade face. This is the cause of why spade is not inputting.

## Hypothesis;

diceValue.getrandom method is not returning a random result which includes all faces

## Predications:

The method is set although Spade is not include and randomisation is not happening.

## Test:

testTheory() creates a new game with 30 rolls. These rolls will have random outputs. A counter method is used to find any spade faces, upon finding a spade face it will print out the number of spades that are found in 30 games.

## Resolution:

DiceValuegetRandom() changed RANDOM.nextint(DiceValue.SPADE.ordinal()) to (VALUE\_REPR\_MAP.size()); This makes the odds around 0.41