Debug Log

## Bug 1:

The reported bug is that balance cannot increase.   
  
The following methods in MainTest.java check the 1to1, 2to1, and 3to1 payouts:

checkGamePaysCorrectWinnings()

checkPlayerBalanceIncreasesWinning2to1()

checkPlayerBalanceIncreasesWinning3to1()

These methods check the playround method, and then get the balance and check it was updated with the correct values.

**Use Case Tested:** Balance will increase successfully when the user wins a game.

**Test Description:** Dice results are mocked to provide 1to1, 2to1 and 3to1 wins and the balance is then checked.

**Pre-conditions:** Balance of the user is 6.

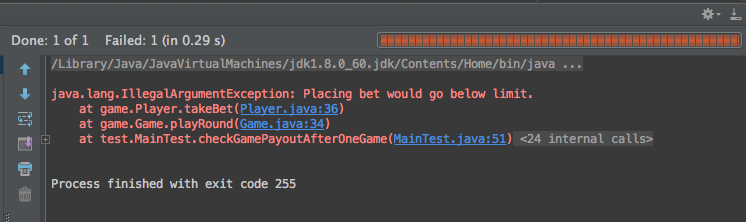
**Post conditions:** Balance of the user is 6, 11, or 16.

**Steps:**

1. Dice mocks are initialised. Values are set depending on the test.
2. Player is initialised with a balance of 6.
3. A new game is created, providing the dice.
4. Game.PlayRound method is called providing the player object, dice value and bet value of 5.
5. Check for correct return values depending on dice.
6. Expected balance is asserted, test should pass.

## Bug 2:

Making a bet of 5 with a a balance of 5, will cause the program to give an exception:



The issue is the operator used in the Player class, checking the amount betted is greater than the limit. It should be greater than or equal to, or the minimum balance will always be 1.

**Use Case Tested:** Player cannot reach betting limit.

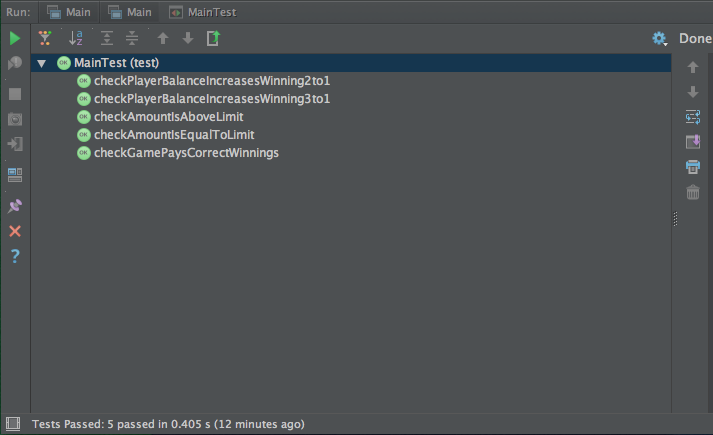
**Test Description:** Check if the player can reach the betting limit.

**Pre-conditions:** Balance of the user is 6

**Post conditions:** Balance of the user is 0.

**Steps:**

1. Dice mocks are initialised. Values are set.
2. Player is initialised with a balance of 6.
3. A new game is created, providing the dice.
4. Game.PlayRound method is called providing the player object, dice value, and bet value of 6.
5. Player will bet the wrong dice value and lose the game.
6. Check balance is zero.

Unit Test Summary:  
  


All tests pass successfully.