

# Simplification

---

*Bug 1: The player is not paid out correctly*

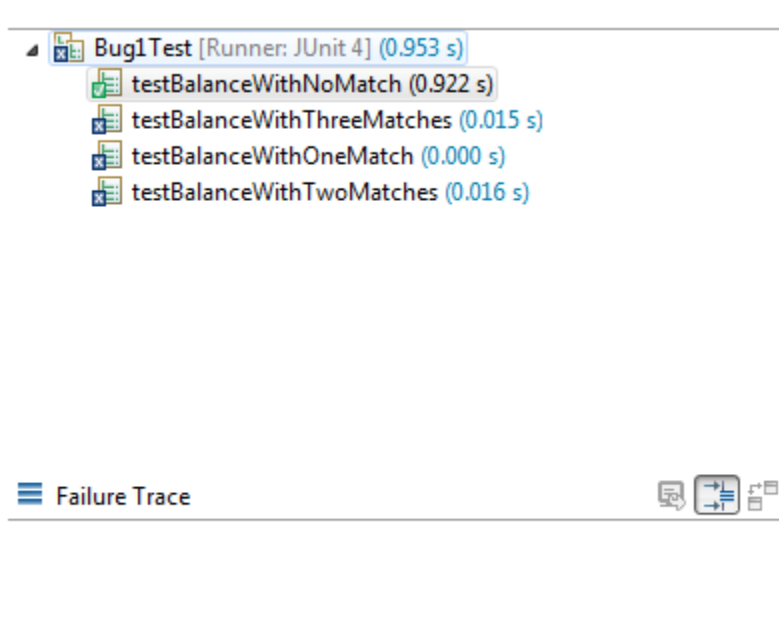
<b>Test Name</b>	Test balance increase correctly after winning
<b>Use Case Tested:</b>	Automate the testing of errors in UAT Test 1
<b>Test Description:</b>	<p>Test whether the player is paid the correct amount</p> <p>In particular:</p> <ul style="list-style-type: none"><li>- If he makes one match, he should get the initial balance and the bet.</li><li>- If he makes two matches, he should get initial balance plus two times the bet.</li><li>- If he makes three matches, he should get the initial balance plus three times the bet.</li></ul>
<b>Pre-conditions</b>	<p>Single player "October" created, balance = 100, bet = 5</p> <p>Each run to use a single value "HEART" as the player's pick.</p> <p>Run game for 4 given combination of dice values.</p>
<b>Post-conditions</b>	n/a

	TEST STEP	EXPECTED TEST RESULTS	RESULT
1.	Run Bug1Test.java	JUnit test and Console are opened.	Pass
2.	Check result of testBalanceWithNoMatch in Failure Trace	JUnit test should be no error and no failure	Pass
3.	Check result of testBalanceWithNoMatch in Console	Balance = 95 Winnings = 0	Pass
4.	Check result of testBalanceWithOneMatch in Failure Trace	JUnit test should be no error and no failure	Fail
5.	Check result of testBalanceWithOneMatch in Console	Balance = 105 Winnings = 5	Fail

	TEST STEP	EXPECTED TEST RESULTS	RESULT
6.	Check result of testBalanceWithTwoMatches in Failure Trace	JUnit test should be no error and no failure	Fail
7.	Check result of testBalanceWithTwoMatches in Console	Balance = 110 Winnings = 10	Fail
8.	Check result of testBalanceWithThreeMatches in Failure Trace	JUnit test should be no error and no failure	Fail
9.	Check result of testBalanceWithThreeMatches in Console	Balance = 115 Winnings = 15	Fail

## Result

### 1. No match: PASS



Bug1Test [Runner: JUnit 4] (0.953 s)

- testBalanceWithNoMatch (0.922 s)
- testBalanceWithThreeMatches (0.015 s)
- testBalanceWithOneMatch (0.000 s)
- testBalanceWithTwoMatches (0.016 s)

Failure Trace

```

Automated Test for Bug 1: The player is not paid out correctly
Start game
October starts with balance 100, limit 0

Rolled ANCHOR, ANCHOR, DIAMOND
October bet 5 on HEART
balance now 95

Winnings: 0

```

## 2. One match: FAIL - Winning is correct but balance is wrong

Bug1Test [Runner: JUnit 4] (0.953 s)  
testBalanceWithNoMatch (0.922 s)  
testBalanceWithThreeMatches (0.015 s)  
testBalanceWithOneMatch (0.000 s)  
testBalanceWithTwoMatches (0.016 s)

### Failure Trace

java.lang.AssertionError: expected:<105> but was:<100>  
at Bug1Test.testBalanceWithOneMatch(Bug1Test.java:83)

Automated Test for Bug 1: The player is not paid out correctly  
Start game  
October starts with balance 100, limit 0

Rolled HEART, ANCHOR, DIAMOND  
October bet 5 on HEART  
balance now 100

Winnings: 5

## 3. Two matches: FAIL - Winning is correct but balance is wrong

Bug1Test [Runner: JUnit 4] (0.953 s)  
testBalanceWithNoMatch (0.922 s)  
testBalanceWithThreeMatches (0.015 s)  
testBalanceWithOneMatch (0.000 s)  
testBalanceWithTwoMatches (0.016 s)

### Failure Trace

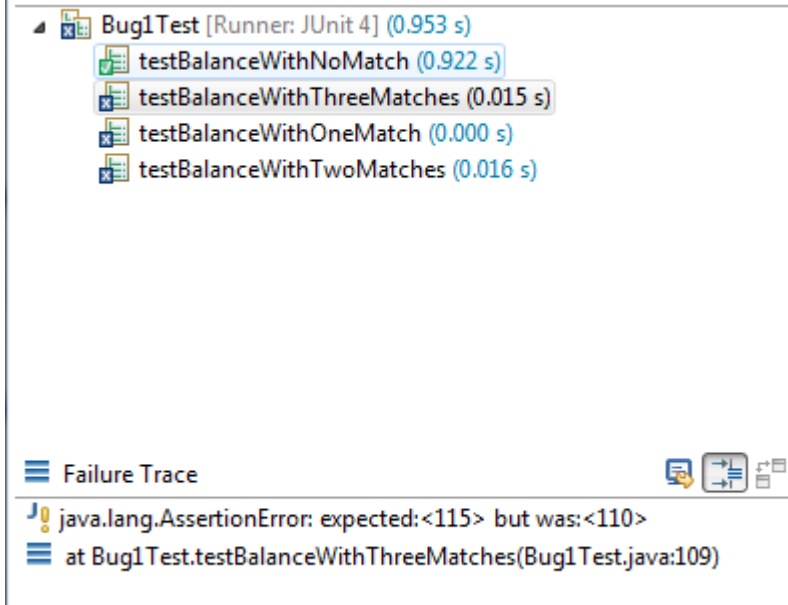
java.lang.AssertionError: expected:<110> but was:<105>  
at Bug1Test.testBalanceWithTwoMatches(Bug1Test.java:96)

Automated Test for Bug 1: The player is not paid out correctly  
Start game  
October starts with balance 100, limit 0

Rolled HEART, HEART, DIAMOND  
October bet 5 on HEART  
balance now 105

Winnings: 10

#### 4. Three matches: FAIL - Winning is correct but balance is wrong



Bug1Test [Runner: JUnit 4] (0.953 s)

- testBalanceWithNoMatch (0.922 s)
- testBalanceWithThreeMatches (0.015 s)
- testBalanceWithOneMatch (0.000 s)
- testBalanceWithTwoMatches (0.016 s)

Failure Trace

java.lang.AssertionError: expected:<115> but was:<110>  
at Bug1Test.testBalanceWithThreeMatches(Bug1Test.java:109)

Automated Test for Bug 1: The player is not paid out correctly  
Start game  
October starts with balance 100, limit 0

Rolled HEART, HEART, HEART  
October bet 5 on HEART  
balance now 110

Winnings: 15