**Test Plan**

|  |  |
| --- | --- |
| # | Plan |
| Bug1 | 1. Make a new Punter object by passing name, 100 and 20 as arguments.  2. place a bet of 10  3. set punter's state to betting  4. call lose bet from the punter  5. Punter's balance must be 90 |
| Bug2 | 1. Make a new Punter object by passing name, 100 and 20 as arguments.  2. place a bet of 10  3. return bet from user  4. call receive winnings with call 10  5. Punter's balance must be 110 |
| Bug3 | 1. Make a new Punter object by passing name, 100 and 20 as arguments.  2. place a bet of 80, must not throw any exception |
| Bug4 | 1. Set win as 0  2. call loop from 1 to 10000 and repeat steps 3 and 4  3. Initialize three dices and for each dice roll it, If its face value matches face with index 0, increment matches by 1  4. if matches are greater than 0, increment win by 1  5. win/10000 must be equal to 0.42 with 0.01 error |

**Bug 1: Player loses double their bet when they lose**

Point of Failure: At line 104 in **Punter**.java, **balance** is incorrect.

CurrentBet is already deducted from the balance. So, it must not be subtracted again.

H0: var **state** is reported incorrectly

T0: set a breakpoint at line 103, visually inspect **state** value before output.

Result0: H0 false: **state** is reported correctly.

H1: var **balance** is reported incorrectly

T1: set a breakpoint at line 104, visually inspect **balance** value before output.

Result0: H1 true: **balance** is incorrect.

**Solution:** the current bet is already subtracted from balance, remove line 104 of punter.java.

**Bug 2: Player doesn’t receive any winnings**

Point of Failure: At line 86 in **Punter**.java, the **state** is incorrect.

It must be set to RECEIVE\_WINNINGS instead of NOT\_BETTING.

H0: var **state** is reported incorrectly

T0: set a breakpoint at line 92, visually inspect **state** value before output.

Result0: H0 true: **state** is incorrect.

H1: var **state** is changed incorrectly

T1: set a breakpoint at line 83, visually inspect **state** value before output and check how it changes in the function.

Result0: H1 true: the **state** is changed incorrectly.

**Solution:** the state must be set to RECEIVE\_WINNINGS in line 86 of punter.java.

**Bug 3: Player cannot reach the betting limit**

Point of Failure: At line 61 in **Punter**.java, comparision is incorrect.

H0: var **limit** is set incorrectly

T0: set a breakpoint at line 46, visually inspect **limit** value and its value in this method.

Result0: H0 false: the **limit** is incorrect.

H1: comparison is incorrect in method balanceExceedsByLimit in lie 61.

T1: set a breakpoint at line 61, visually inspect the **comparison.**

Result0: H1 true: **comparison** is made incorrectly.

**Solution:** instead of checking just the greater than condition, equal to the condition must also be checked.

**Bug 4: Odds in the game are incorrect.**

Point of Failure: At line 34 in **Face**.java, **len** variable is incorrect.

H0: var **len** is set incorrectly

T0: set a breakpoint at line 34, visually inspect **len** value and its value in this method.

Result0: H0 true: **len** is incorrect.

**Solution:** random.nextInt(value), returns a random value from 0 to value-1. When we have 5 faces, len must be 6 to get a value from 0 to 5 but the len var was set to 5 and we were only getting the random values from 0 to 4.