| **Test Name** | | | Payout Balance Test | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Tested:** | | | User wins and balance is updated with earnings | | | |
| **Test Description:** | | | This test shows how the game doesn’t update the user’s balance when the user wins a game | | | |
| **Pre-conditions** | | | Fred must win. Initial balance is 100. | | | |
| **Post-conditions** | | | Fred’s balance should have gone up after a win | | | |
| **Notes:** | |  | | | | |
| **Result (Pass/Fail/Warning/Incomplete)** | | **Fail** | | | | |
|  | **TEST STEP** | | | **EXPECTED TEST RESULTS** | P | F |
|  | Set the value of pick to DiceValue.ANCHOR | | | This will make the value of the user’s pick ANCHOR | P |  |
|  | Remove all elements in cdv and add DiceValue.ANCHOR and 2 random values of choosing. | | | This will make the returned dice values ANCHOR and any 2 values i.e. CLUB, HEART | P |  |
|  | Run | | | Fred starts with balance 100, limit 0  Turn 1: Fred bet 5 on ANCHOR  Rolled ANCHOR, CLUB, HEART  Fred won 5, balance now 105 |  | F |

|  | **TEST STEP** | **EXPECTED TEST RESULTS** | P | F |
| --- | --- | --- | --- | --- |
|  | Set the value of pick to DiceValue.HEART | This will make the value of the user’s pick HEART | P |  |
|  | Remove all elements in cdv and add DiceValue.HEART and DiceValue.HEART and 1 other value. | This will make the returned dice values of 2 HEARTs and one other random value | P |  |
|  | Run | Fred starts with balance 100, limit 0  Turn 1: Fred bet 5 on HEART  Rolled HEART, HEART, CLUB  Fred won 10, balance now 110 |  | F |
|  | Set the value of random in DiceValue.java to 1 to make every value returned to ANCHOR | This will make every value returned ANCHOR | P |  |
|  | Run | Fred starts with balance 100, limit 0  Turn 1: Fred bet 5 on ANCHOR  Rolled ANCHOR, ANCHOR, ANCHOR  Fred won 15, balance now 115 |  | F |