|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bug Number 1** | | Game does not pay out at correct level. | | |
| **Bug Description** | | When player wins on 1 match, balance does not increase. | | |
| **Reproduction Test Description:** | | Run simulation, find instance where player wins on one match and compare balance to previous balance. | | |
| **Pre-conditions** | | Use ‘baseline’ version of Main class. | | |
| **Post-conditions** | | N/A | | |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | |
|  | Run the simulation (execute Main.main()). | | Simulation starts & prints output of simulation (100 games and stats) to console. |  |
|  | Scroll to top (start of simulation) of output in console. | | Output starts with:  Start CrownAndAnchorGame.Game 0:  Fred starts with balance 100, limit 0  … |  |
|  | Read down through the output until you reach the first (or any) instance where Fred wins with one match | | Identify instance by finding a case where the suite bet on is rolled ONLY once (but in no particular order):  Turn 3: Fred bet 5 on **ANCHOR**  Rolled DIAMOND, **ANCHOR**, HEART |  |
|  | Compare the balance in the previous turn to the balance in the turn where Fred wins with one match \*  \* if the first instance is the first round of the game, you need to compare the balance to the balance in the preceding statement ‘Fred starts with balance 100, limit 0.’ | | The balances match (although the new balance should be greater due to the win) e.g.:  Turn 2: Fred bet 5 on ANCHOR  Rolled CROWN, HEART, HEART  Fred lost, balance now **90**  Turn 3: Fred bet 5 on CROWN  Rolled CROWN, HEART, HEART  Fred won 5, balance now **90** |  |

|  |  |
| --- | --- |
| **Automated Test** | TestBug1 |
| **Bug Description** | Given game with 1 die that matches player’s roll, final balance should equal starting balance + winnings |
| **Test Output** | Final balance remains as at starting balance (see failure below) and does not increase by winnings amount |

