Debugging Log:

Bug 1: Game not paying out at correct level.

Code Run Through with Debugger 1:

|  |  |  |
| --- | --- | --- |
| File | Line | Possible Problem |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Hypothesis 1:

Test 1:

Test 2:

Test 3:

Issue Resolved: Y/N

Code Run Through with Debugger 2:

|  |  |  |
| --- | --- | --- |
| File | Line | Reason |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Hypothesis 2:

Test 1:

Test 2:

Test 3:

Issue Resolved: Y/N

Bug 2: Player never reaches betting limit.

Code Run Through with Debugger 1:

|  |  |  |
| --- | --- | --- |
| File | Line | Reason |
| Game.java | 40 | Uncertain of why matches +=1; shouldnt it be matches++;? Is there a difference? |
| Player.java | 27 | return (balance > limit); The balance should be allowed to be >= limit? This would make sense with the reported bug. |
| Player.java | 31 | return (balance - amount > limit); As above. E.g. amount = 5, balance = 5, limit = 0 |
|  |  |  |

Hypothesis 1:

Test 1:

Test 2:

Test 3:

Issue Resolved: Y/N

Code Run Through with Debugger 2:

|  |  |  |
| --- | --- | --- |
| File | Line | Reason |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Hypothesis 2:

Test 1:

Test 2:

Test 3:

Issue Resolved: Y/N

Bug 3: Application odds incorrect.

Code Run Through with Debugger 1:

|  |  |  |
| --- | --- | --- |
| File | Line | Reason |
| Dice.java | 16 | The roll() method doesn’t update DiceValue value variable in class Dice. This means a call to value doesn’t return the last value of the dice when it was rolled. |
| Game.java | 21-27 | The code at this location shows that the game.getDiceValues() method simply calls dice.getValue() on each dice to return their present values. |
| Game.java | 37-41 | The number of matches the user makes on the die is done based on the dice.getValue() method which is not updated by a call to roll, which is done in line 38.  Seen with debugger. The value of d.getValue() doesn’t appear to change after a d.roll().  e.g. Before Roll:  After Roll:    The value doesn’t change.  Variable “value” in dice.java is sane before roll() and is infected after roll(). |

Hypothesis 1: The method dice.roll() should update the variable value to ensure that the dice values change.

Test 1: Set d.roll() method to update the variable:  
  
Added to line 16 of Dice.java:

value = DiceValue.getRandom();

return value;

Ran debugger:

Test 2:

Test 3:

Issue Resolved: Y/N

Template:

Code Run Through with Debugger X:

|  |  |  |
| --- | --- | --- |
| File | Line | Reason |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Hypothesis X:

Test 1:

Test 2:

Test 3:

Issue Resolved: Y/N