Debugging Log

**Assignment 2**

ITC515 – Professional Programming Practice

Justin McKay - Student ID: 11530610

# Debugging Log

## BUG001 - Game does not pay out at correct level

### Steps taken to trace error

* Check *Player.receiveWinnings* method. Method appears to add winnings to balance correctly.
* Check *Game.playRound* method. Note: playRound method calls the takeBet method prior to testing the rolled dice against the value selected.
* This does not appear to be a bug. There could be confusion with the game only the winnings, so you don’t actually see the bet being taken out, and then the winnings put back onto the balance.
* The following 4 screenshots show the progression from taking the bet (balance before and after) and receiving the winnings (balance before and after)

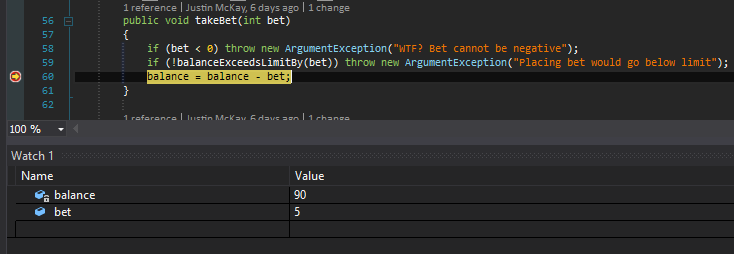


Figure - Showing balance before the bet has been taken

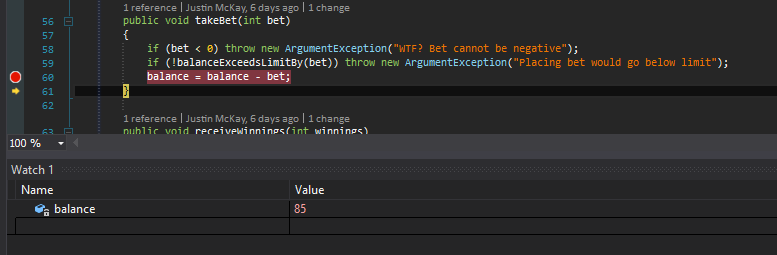


Figure - Showing the balance after the bet has been taken

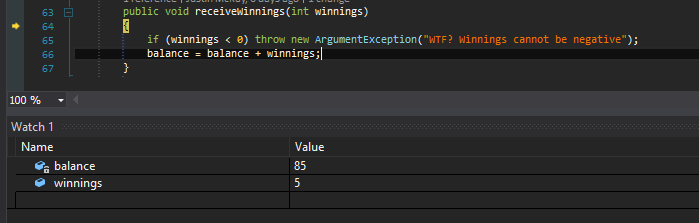


Figure - Showing the balance before the winnings have been received

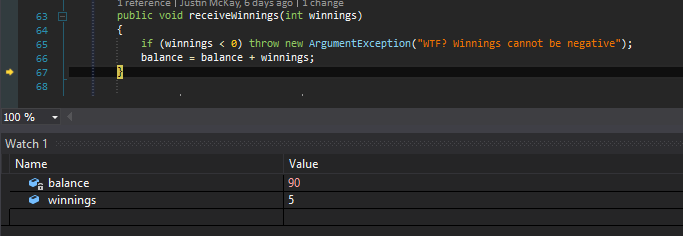


Figure - Showing the balance after the winnings have been added back to the balance

* Will write a test that confirms the bet has been placed (bet deducted from balance) and when a single dice wins, shows the winnings added to the balance.

### Bug elimination steps

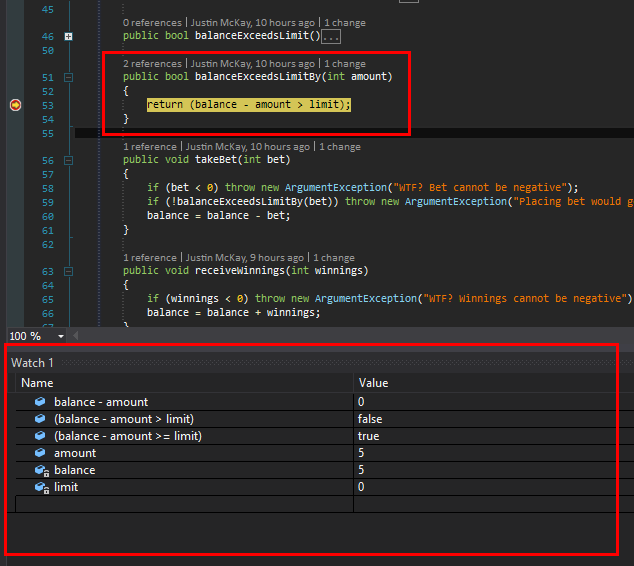
As this is not actually a bug, no steps are required for corrective action.

### Possible Improvements

Change the UI to reflect both the bet being taken off the balance and the winnings being added to the balance. This should correct the issue.

## BUG002 - Player cannot reach betting limit

### Steps taken to trace error

* Check *Player.takeBet* method. Noticed *Player.balanceExceedsLimitBy method* will prevent the bet from being taken if false is returned.
* Step into *Player.balanceExceedsLimitBy* method. Notices that the method will only return true (and allow the player to take a bet) if the balance – bet is greater than 0 (so that you can’t end up with a negative limit). However, as the limit is 0, this method should be checking for balance – bet is greater than *or equal to* the betting limit.
* This behaviour is confirmed with the following debugger screenshot:  
  

### Bug Elimination Steps

To fix the bug, the following code needs to be updated:

**Player.cs:**Change line 53 to be amount >= limit

### Possible Code Refactoring Improvements

Consider renaming method from *balanceExceedsLimit* to *balanceReachedLimit* so that it’s slightly clearer to the developer that the balance can reach (equal to) or exceed (greater than) the limit.

## BUG003 - Odds in game do not appear to be correct

### Steps taken to trace error

* When running the game in debug mode from Visual Studio, the loss count is always 2000, which I suspect is causing the results to be skewed.
* Running the game outside of Visual Studio (i.e. from command line) the win:(win+lose) ratio is closer to the correct ratio of 0.42.