Bug 1

When player wins on 1 match, balance does not increase.

Sample Buggy Output

```
Turn 76: Fred bet 5 on ANCHOR
Rolled CROWN, CROWN, CLUB
Fred lost, balance now 20

Turn 77: Fred bet 5 on CLUB
Rolled CROWN, CROWN, CLUB
Fred won 5, balance now 20
```

```
Turn 7: Fred bet 5 on CLUB
Rolled CLUB, ANCHOR, CLUB
Fred won 10, balance now 90

Turn 8: Fred bet 5 on ANCHOR
Rolled CLUB, ANCHOR, CLUB
Fred won 5, balance now 90
```

H1. The bug is in the `Game.playRound()` function.

Test	Check the winnings amount is passed correctly. Place a breakpoint on line 46 of Game.java `player.receiveWinnings(winnings);`
Prediction	winnings will have a value of 0 when passed to `Player.receiveWinnings()`
Result	The player isn't getting payed out correctly, if they bet 5 and win, they should receive 10. The original 5 bet plus 5 winnings.
Notes	• Adjusted the winnings assignment statement, multiplied bet by 2 before multiplying it by the number of matches won. `int winnings = matches * (bet * 2);`

H2. The bug should be resolved.

Test	Run the game, and check that the balance is increasing when the player wins.
Prediction	The balance will increase correctly when the player wins.
Result	The balance increases correctly.

	`Turn 93: Fred bet 5 on CROWN`
	`Rolled CLUB, ANCHOR, CROWN`
	`Fred won 10, balance now 195`
	
	`Turn 94: Fred bet 5 on ANCHOR`
	`Rolled CLUB, ANCHOR, CROWN`
	`Fred won 10, balance now 200`

Notes	My first guess was correct. ☺

Sample Fixed Output

Turn 141: Fred bet 5 on ANCHOR Rolled CLUB, ANCHOR, CROWN Fred won 10, balance now 145 Turn 142: Fred bet 5 on ANCHOR Rolled CLUB, ANCHOR, CROWN Fred won 10, balance now 150	Turn 87: Fred bet 5 on CLUB Rolled CROWN, ANCHOR, CLUB Fred won 10, balance now 195 Turn 88: Fred bet 5 on CROWN Rolled CROWN, ANCHOR, CLUB Fred won 10, balance now 200

Before and after screen shot of the bug.

```java
public int playRound(Player player, DiceValue pick, int bet ) {
    if (player == null) throw new IllegalArgumentException("Player cannot be null.");
    if (pick == null) throw new IllegalArgumentException("Pick cannot be negative.");
    if (bet < 0) throw new IllegalArgumentException("Bet cannot be negative.");

    player.takeBet(bet);

    int matches = 0;
    for ( Dice d : dice) {
        d.roll();
        if (d.getValue().equals(pick)) {
            matches += 1;
        }
    }

    int winnings = matches * bet;

    if (matches > 0) {
        player.receiveWinnings(winnings);
    }

    return winnings;
}
```

```java
public int playRound(Player player, DiceValue pick, int bet ) {
    if (player == null) throw new IllegalArgumentException("Player cannot be null.");
    if (pick == null) throw new IllegalArgumentException("Pick cannot be negative.");
    if (bet < 0) throw new IllegalArgumentException("Bet cannot be negative.");

    player.takeBet(bet);

    int matches = 0;
    for ( Dice d : dice) {
        d.roll();
        if (d.getValue().equals(pick)) {
            matches += 1;
        }
    }

    int winnings = matches * (bet * 2);

    if (matches > 0) {
        player.receiveWinnings(winnings);
    }

    return winnings;
}
```