Jaspal Khanuja

Mr. Howard

UCLA ID: 105864172

Project 6

In project 6, there were some serious obstacles that I had to figure out. One being the BJ dice, as during the initial moments of learning about how the project worked, I did not quite understand how the player and the computer dice rolls worked. As everytime I tried it seemed that the roll function would not roll the dice and assign them random values, in my case I simply deleted a portion of code from the dice function on accident while looking through it and it must have made the roll method not work. Also in the player class, I had some trouble with my previous running total as I called getRunningTotal to get a running total, but if no running total was every given then this value would forever remain 0, i changed this my implementing the mPriorRunnign total, set equal to the runngintotal s taht every time it runs the turn, the running total will automatically update the prior running total, those were the biggest issues that the code was having. In terms of testing code, I referred to the code that Mr. Howard gave us initially, with the cheat setValue, however, I used the codeBoard error test code the most, specifically the code:

Expected output was correct: 003073

Passed test case-Player's running total and prior running total working correctly:

Input:

player 6

Expected output was correct: 0012017122317

Passed test case-Board opening initializations are correct:

Input:

board 1

While this code states that both test cases passed, this was only the outcome of my completely revised code, at the time, what this code was checking for was something along the lines of

Dice die1;

Dice die2;

setvaluedi1(5);

setValue die2(6);

Humanroll…..

assert (mPriorRunnign total == 0);

assert(mRuningTotal == 11);

Then you would also check for certain scenarios such as the 16 dice roll rule in which after the running total is equal to 16 then from that point only use one die, the psedo code would look something like this:

Continued from above:

Die1. SetValue(4);

Die2.setBalue(1);

assert(mRunningTotal == 16);

Die1.setValue(3);

Die2.setValue(4);

assert(mRunnignTotal == 19);

The last 3 lines show another roll being taken by the said player and then showing that their total reached 16, and when their next die came up as 3, only the number 3 was added to the running total private varbiale.

Those were the main testing code that I used for my code and the problems that I ran across for the most part, and aprt from that my code did not have any compiler errors, and successfully ran all the codeboard test cases without fail. So to myknoledge there is no breaking code.