Project 2 Phase 1 Report

Tracy Karol

4/16/2018

Instructor: Mark Baldwin

**Description**

This program is a Blockhead Poker game that features an AI player to play against. It is based on input and output from the terminal and allows users to make betting decisions throughout the game. The logic for the AI player is quite simple, but in the next deliverable, a Beta AI will be created that can make better decisions and will be superior to the Alpha AI.

To code this game, I followed the template of classes given in the project description, which included an abstract Player class. I added some variables and methods to the classes to make further use of them. I split up the different kinds of rounds into methods within the Game object. These included methods that executed a single hand round (there are twenty in the game total) and a method to execute a single betting round (there are three in each hand round). This made the code more organized and minimized the tasks that the game methods need to take care of.

**Challenges**

Something I struggled with for this phase was working through the flow of the game-play as I have never played poker before. As I was fleshing out each class, I found that most of them needed more methods and variables to facilitate the game, but they weren’t very difficult to add. Another problem I ran into was guarding against a user entering a letter rather than a number when prompted for a raise amount. Since I was already safeguarding for the user entering a number out of the raising bounds (1-10), I wasn’t sure how to also check for the entry being a letter in the same loop.

**Testing**

Test 1: When prompted for a letter input (R, F, or C), user should not be able to enter an invalid option

Expected: An error message will show and prompt for a new entry

Actual: Got expected result

Test 2: After raising three times, player shouldn’t be able to raise anymore

Expected: When prompted, user can only call or fold

Actual: Got expected result

Test 3: Deck should contain correct cards

Expected: Each suit has 13 cards

Actual: Got expected result (printed out the deck to test)

Test 4: Player’s chip amount should be consistent

Expected: Player’s chip amount decreases after adding to the pot

Actual: Got expected result

Test 5: Player’s hand should be correctly evaluated

Expected: At end of hand, player with larger valued hand should win

Actual: Got expected result

Test 6: Player should be able to exit the game

Expected: Game exits if ‘Q’ is entered

Actual: Got expected result