CS 180 Project 4

1 BlackJack Win Percent

Project assigned on: Friday Feb 17th 2012
Project due date: Thursday Feb 23rd 11:59 p.m. 2012

1.1 Description

In this project, you are to write a program that estimates the probability of winning a blackjack game between you and a dealer.

The program runs 1,000,000 rounds. For each round, you and the dealer draw two cards. For each hand, if the cards are worth 17 or more, don’t pick any more. If they are worth less than 17, draw a card. For our game, you can draw at most four (4) cards.

After both hands are finished drawing, compare the totals. If the player “busts” (has a hand greater than 21), that is a loss. If only the dealer busts, the player wins. Otherwise, if the player has a higher hand, the player wins. If the dealer has the higher hand, the player loses. If they have the same value, they get a tie (“push”).

After comparing the hands, the result is added to a win, loss, or tie count. After 1,000,000 rounds, print the percentage of winning, losing, and tieing hands with these rules.

2 Learning objectives

1. Learning how to separate code into methods to simplify code.
2. Using for loops to simulate a large number of actions.

3 Project Setup

Create a project4 folder in your cs180 folder. Save all your Java source files in this project4 folder. You will be turning in the contents of this folder when the project is completed.

Note: Use the commands below, if needed.

% cd ~/cs180
% mkdir project4
% cd project4
% pwd
/u/u9x/yourlogin/cs180/project4
% drjava

4 Project Details
The next sections describe the methods used in your program. Note that your program reads no input.

4.1 Method getHand

The method `getHand` returns the value (card sum) of a hand after you have picked all the cards according to the rules of our BlackJack game in the description. Each time you need to pick a card, you call the method `getCard`. Make sure to take into account that Aces are considered as having a value of 11, or 1, if the hand is over 21. Your getHand method may return a value over 21 if the hand represents a bust.

4.2 Method getCard

To implement method `getCard`, you draw any card from a two to an ace. Assume the deck has an infinite number of cards, meaning that drawing of a card is not affected by previous draws, nor does it affect future draws. Use a method from the java.util.Random class to figure out which card you picked, then return the value of that card. When initializing the Random variable, use a seed of 0.

Use the following values as a reference for the value of cards:

1. Two: value of 2
2. Three: value of 3
3. Four: value of 4
4. Five: value of 5
5. Six: value of 6
6. Seven: value of 7
7. Eight: value of 8
8. Nine: value of 9
9. Ten: value of 10
10. Jack: value of 10
11. Queen: value of 10
12. King: value of 10
13. Ace: value of 11, or 1 if the hand would be over 21

4.3 Method decideGame

In your main method, call the method `decideGame` to decide who has won the current round.

1. If the dealer wins, return -1 to signal a loss.
2. If the player wins, return 1 to signal a win.
3. If the dealer and player tie, return 0 to signal a tie.

4.4 Displaying Output
Your program should output three values as percentages: the chance of winning, the chance
of losing, and the chance of tying. Format these values to 5 digits of precision. Use either the
DecimalFormat class from the java.text package, or the printf() method from System.out.

Example Output:

Wins: 45.000%
Losses: 51.000%
Ties: 6.000%

5 Skeleton Code

/**
 * Project 4 -- BlackJack Simulator
 * @author your name
 * @recitation section number and recitation instructor's name
 * @date date of completion
 */
public class BlackJack {

    /**
     * Determine if the dealer or player has won this game (round).
     * @param dealerValue value of dealer’s hand
     * @param playerValue value of player’s hand
     * @return 1, 0, -1, indicated player wins, ties, or loses
     */
    public static int decideGame(int dealerValue, int playerValue) {
        return 0;
    }

    /**
     * Draw cards into the “hand”, calculating its value. Stop when the total
     * is at least 17 (handling Aces correctly to avoid “busting” if possible).
     * @return integer value of the hand.
     */
    public static int getHand() {
        return 0;
    }

    /**
     * Use a random number generator to select a card from the deck.
     * @return an integer representing the value of the card
     */
private static int getCard()
{
    // uses the random variable to decide which card was picked
    return 0;
}

public static void main(String[] args)
{
    // Your Main Program
}

6 Coding standards

Make sure you follow the coding standards specified in the course website at
http://web.ics.purdue.edu/~cs180/Spring2012Web/java_programming_standards.html

7 Grading

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct Card Draw</td>
<td>10</td>
</tr>
<tr>
<td>Correct Hand Draw</td>
<td>30</td>
</tr>
<tr>
<td>Correct Decision for Who Wins</td>
<td>10</td>
</tr>
<tr>
<td>Calculations</td>
<td>20</td>
</tr>
<tr>
<td>Correct Output</td>
<td>20</td>
</tr>
<tr>
<td>Coding Standard</td>
<td>10</td>
</tr>
</tbody>
</table>

8 Turnin

Make sure you have class level comments that clearly specify your name, cs login, recitation section number, and date.

/*
 * name:
 * cs login:
 * recitation section
 * date
 */

To turn in your project, first remove all the .class files and other backup files (e.g., files with names ending in ~) from your project4 folder.

% pwd
/u/u9x/yourlogin/cs180/project4
% rm -i *.class *~
Change your current folder to your cs180 folder and only then run the turnin command.

% cd ..
% pwd
/u/u9x/yourlogin/cs180
% turnin -v -c cs180=XXX -p project4 project4

In the table below, find the recitation section in which you are enrolled. Substitute the XXX above with the corresponding value in column XXX.

<table>
<thead>
<tr>
<th>Section Day</th>
<th>Time</th>
<th>Rec. TA</th>
<th>XXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>R01 F</td>
<td>11:30:00 AM</td>
<td>Gnanasekaran, Bala</td>
<td>REC1</td>
</tr>
<tr>
<td>R02 F</td>
<td>02:30:00 PM</td>
<td>Zage, John</td>
<td>REC2</td>
</tr>
<tr>
<td>R03 F</td>
<td>09:30:00 AM</td>
<td>Stephen, Julian</td>
<td>REC3</td>
</tr>
<tr>
<td>R04 F</td>
<td>08:30:00 AM</td>
<td>Sriran, Dinesh</td>
<td>REC4</td>
</tr>
<tr>
<td>RM1 F</td>
<td>03:30:00 PM</td>
<td>Zage, John</td>
<td>REC2</td>
</tr>
<tr>
<td>RM2 F</td>
<td>10:30:00 AM</td>
<td>Stephen, Julian</td>
<td>REC3</td>
</tr>
<tr>
<td>RM3 F</td>
<td>04:30:00 PM</td>
<td>Sriram, Dinesh</td>
<td>REC4</td>
</tr>
</tbody>
</table>