CIS 211
Spring 2014

Project 5: Blackjack

Due Monday May 5
Upload via Blackboard by 11:00 P.M.

Reading

Introduction to Computing Using Python, Ch 9, plus resources from the class web site.

Modules

You will need to use the Card class from Project 3 and the Deck class from Project 4. If you want you can download the instructors' code for these classes from the web. We will provide precompiled binaries in files named Card.pyc and Deck.pyc that will work for Python 3.3.3 (let us know if you want a file for a different version).

You will also need the CardLabel class and card images you used in Project 4.

Programming Project

Write a program named `blackjack.py` that will play a game of Blackjack. The top level window should have room for two rows of cards, with six cards in each row. Below the cards display three buttons, named "deal", "hit", and "pass".

The deal button should start a new game. Shuffle a deck of cards, then display two cards for the dealer in the top row, with one card face down, and display two cards for the player, face up, in the bottom row. The remaining four places in each row should be blank.

The hit button should turn over the next card in the bottom row, and it should update the player's score using a function named `total` (decscribed below). If the score is over 21 the player loses the game, and your program should display an alert box with a consoling message.

If the user clicks the pass button, turn over the dealer's hidden card and compute the total points for the cards in the dealer's row. Then while the dealer's total score is less than 17 turn over a the next card in the dealer's hand and add the points for that card to the dealer's total. When the dealer's total is 17 or higher compare the dealer't total with the player's total and display a message that shows the result of the game (dealer wins, player wins, or tie game).

The Number of Points in a Hand

Page 201 of the textbook has the Python code for a function named `total` that shows how to compute the total number of points for a hand in Blackjack. You should adapt this algorithm to work with the points method in the BlackjackCard class you wrote for the previous assignment.

The idea is to use a loop that computes an initial total, using 11 points for each ace and counts the number of aces in the hand. Then a second loop changes aces to 1 point if necessary: while the number of aces is greater than 0 and the total is greater than 21 subtract 10 from the total and subtract one from the ace counter.

Extra Credit Ideas

- Have your program keep track of the number of games the player wins and loses, and display the current total somewhere in the window.
- Allow the player to specify an account balance and an amount they want to bet, and update the balance after every game.
- When showing the dealer's cards have the program pause for ½ second between each card.
- Check to see if either player has "blackjack" (a total of 21 using only two cards), and if so, display the game results right away, without waiting for the hit or pass button.
- In some casinos, the dealer or the player wins if they have 5 cards that total 21 or less, no matter what the other player's total is (unless the other person has a blackjack).
- The window in your Blackjack program has room for 6 cards for each player. Is this enough space? What is the probability of a player needing more than 6 cards? One way to figure this out is to use a technique called, appropriately enough, *Monte Carlo integration*.

  Create a histogram with 10 bins, labeled 1 through 10, all initially set to 0. Then deal a series of hands, keeping track of how many cards you need to deal before the total is 22 or higher. For each hand update the bin that corresponds to the number of cards dealt; for example, if a hand reaches 25 on the 3rd card add one to the count in bin 3. If you repeat this process several million times you will start to get a pretty good idea of the shape of the probability distribution.

  How many times did you get a hand that would have won a game with 5 cards, i.e. how many hands took 6 or more cards to go past 21? From this count can you calculate the probability of winning with 5 cards? Will your program ever see a hand that requires 7 or more images in a single row?

What to Turn In

Documentation Write a short description (two or three paragraphs total) of what you did for this project. The documentation should be in a file named `writeup` with an extension that identifies the file format (.doc for Microsoft Word, .pdf for Adobe PDF, .txt for plain text, .rtf for rich text format).

If you did anything extra for this project describe it in a separate paragraph at the end of your writeup.

Create a package (tar or zip format) that includes your writeup and your blackjack.py program and upload the package via Blackboard.