Homework 7

Software Design **Allen B. Downey**
Spring 2006

Due: Wednesday 15 March

7.1 Poker!

The following are the possible hands in poker, in increasing order of value (and decreasing order of probability):

pair: two cards with the same rank

two pair: two pairs of cards with the same rank

three of a kind: three cards with the same rank

straight: five cards with ranks in sequence (aces can be high or low, but other than that, there is no wraparound).

flush: five cards with the same suit

full house: three cards with one rank, two cards with another

four of a kind: four cards with the same rank

straight flush: five cards in sequence and with the same suit

The goal of this homework is to use Monte Carlo simulation to estimate the probability of drawing these various hands.

1. Type the following commands in your `sd` directory to download and unpack a tar file:

   ```
   wget http://wb/sd/code/poker.tgz
   tar -xzf poker.tgz
   ```

   It should create a directory named `poker` that contains the following files:

   `Card.py` : A complete version of the `Card` and `Deck` classes in the book.

   `Poker.py` : An example program that shows a graphical representation of playing cards on a "table". You don't have to do anything with this program, but you might find it useful for debugging.

   `cardsets` : A directory containing GIF files for playing cards in a variety of styles.

   `World.py, Gui.py, Lumpy.py` : The same files you should already have.

   As always, you should take some time to familiarize yourself with the code I give you, especially `Card.py`.

2. `Card.py` includes a class named `PokerHand` that provides a method named `has_flush` that determines whether or not a hand has a flush. Add similar methods named `has_pair`, `has_twopair`, etc. that return True or False according to whether or not the hand meets the relevant criteria. Your code should work correctly for "hands" that contain any number of cards (although 5 and 7 are the most common sizes).
3. Write a method named `classify` that figures out the highest-value classification for a hand and sets the `label` attribute accordingly. For example, a 7-card hand might contain a flush and a pair; it should be labeled "flush".
4. Write a loop that shuffles a deck of cards, divides it into hands, classifies the hands, and counts the number of times various classifications appear.
5. Print a table of the classifications and their probabilities. Run your program with larger and larger numbers of hands until the output values converge to a reasonable degree of accuracy.
6. As a JFFE, you can investigate the claim that certain hands tend to occur in clusters. For example, if you are at a table with four other players, and you have a flush, is it more likely that someone else at the table also has a flush? What about a full house?

What to turn in

1. As always, take some time to improve your code, and add comments that explain any non-obvious features.
2. Please print a copy of the complete program, and also a copy of the probability table that it outputs.