Dayna Dunninger

Software Development I

Prof. Pablo Rivas

8 November 2016

Final Project Write-up

Abstract:

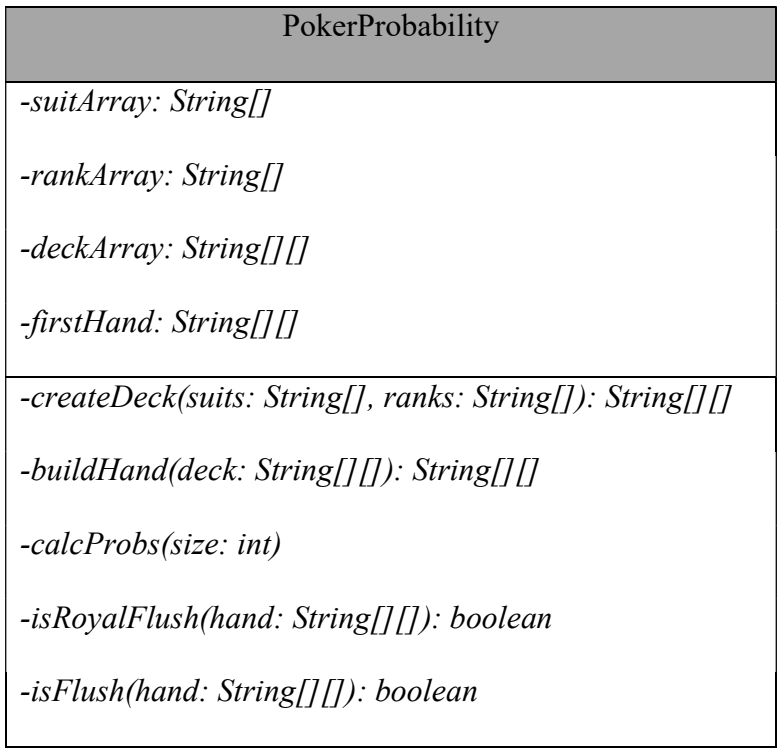
From the time I submitted my final project proposal to now, the premise of my project has changed. I will still be using probability as the topic of my project and paper, but I will not be doing combinations and permutations in order to find the probability of getting certain poker hands. Instead, for my final project I am using a double array in order to build a deck of cards with four suits and thirteen ranks. From this deck of cards, more arrays are used in order to develop a five card poker hand that will be analyzed. The five card hand is then analyzed and determined whether it contains a flush or a royal flush. Once it is determined what kind of a hand it is, royal flush, regular flush, or neither, there is a count for the two flush categories that are incremented each time one of these hands occur. The counts for the royal flush and flush categories end after all of the hands have been gone through. These counts represent the number of royal flushes and regular flushes out of all of the hands. The counts are then divided by the total number of hands to give us a calculated probability for receiving a royal flush and a flush. The number of hands, the calculated probabilities, as well as the actual probabilities from the Monte Carlo experiment are then printed to the screen. This allows the user to compare and notice the difference in the calculated probabilities from the program and the actual probabilities of receiving those particular hands.

Introduction:

When choosing my final project idea I wanted to choose something that applied to my major as well as what I want to pursue as a career. I am currently an applied mathematics major with an actuarial track in the process of taking my first actuarial exam. Software Development I, although required for my major, was a class I chose to take to continue in the science section of my actuarial track. Software Development seemed like the best choice compared to something like Chemistry or Biology because I could potentially be working with java and programming in my future career. I'm very glad I did choose this class because I am able to apply what I learn in certain math classes and create a program that I enjoyed working on.

One of the other classes I am currently taking is probability and statistics, which happens to also be the subject of the first actuarial exam. This project allows me to see the calculated probabilities from my program and apply what I learn about probabilities to the outcomes of the program. In doing so I also wanted the user to have a hands on experience with the program. Allowing the user to enter the number of hands which are generated gives insight to how the probabilities work. In entering in a variety of numbers, the higher the number is the closer you will be to the actual probability from the Monte Carlo experiment.

Detailed System Description:



Requirements:

There are only a certain amount of things you need in order to run this program. The user would need a laptop or computer in which to run the program. On the device that the person is using they would need to have the latest version of Java installed. This person would also have to have a basic understanding of the way java works and how to run a program from the computer that they are utilizing. It would be very beneficial for the user to understand the composition of a deck of cards as well as the hierarchy of all of the poker hands. With this the user should also have a knowledge of the way poker works in the area of knowing what the requirements for each possible hand is, in particular the royal flush and flush.

Literature Survey:

Many different projects have been done in programming that have to do with poker hands and the concepts that go along with generating these hands and the results. One article I found that takes a part of creating the poker hands in a different way uses C++ programming and uses algorithms in order to implement the shuffling of a deck of cards through the use of arrays. As noted in the article this could also be used in order to shuffle any number of other categories that would be necessary. The difference between this and my program, aside from the type of code it is, is that instead of shuffling the deck of cards, I go through the deck of cards and randomly choose a certain amount of cards to create a poker hand. (Rolfe 1)

In a more generalized sense there are many other ways in which someone can implement this program. One change, for example, could be that instead of using arrays, the programmer could implement card objects. There is also the amount of cards in the poker hand. The programmer could decide to only have four hands or to make it a seven card hand. Another thing the programmer could do is include different instances of what poker hands the user could receive from the five card hand: high card, one pair, two pair, three-of-a-kind, four-of-a-kind, full house, straight flush, or bust.

User Manual:

For this particular project, the user is not required to do much with using the program. The program automatically has a certain amount of cards that the deck should have along with the number and types of both the suit and the rank: a 52 card deck with 4 suits (Spades, Hearts, Clubs, and Diamonds) and 13 ranks (numbers 2-10 as well as Jack, Queen, King and Ace). It is also already decided in the code how many cards the poker hand will contain, which is five each time. The first and only thing that the user must input is how many hands they wish to pick

consecutively. When this is done the user then waits a certain amount of time depending on the number that they entered and then the amount in order for the program to count how many royal and regular flushes there are. Each number plus their calculated probabilities verses the actual probabilities are the printed to the screen. From here, any user can then calculate things such as margin of error or simply just compare and contrast the two probabilities for each category.

Conclusion:

As stated in the Literature Survey section, other things that could be applied to this project are calculating and analyzing more probabilities in any of the other categories in poker hands such as full house or three-of-a-kind. Errors in the code will occur if the user enters a number that is too big to be stored as an integer. There will also be an inaccurate representation of the calculated probabilities if the user enters a number that is too small because in a smaller amount of hands there may not be any amount of royal flushes or regular flushes giving us zero hands and a probability of zero. Mathematicians could potentially use this code in order to calculate the percent error between the calculated and actual probabilities as well as compare it to other aspects in the subject or probability and statistics.

References / Bibliography:

Rolfe, Timothy. "Randomized Shuffling." *Dr. Dobb's Journal* 25.1 (2000): 113-

4. *ProQuest.* Web. 12 Dec. 2016.