Tony Diep
3-11-17
CS 2420
Design Document

Assignment 8: Stochastic Processes - Poker

The purpose of this project is to explore a different approach to solving problems as well as working with other aspects of what is defined to be “random”. When using Java’s utilization and algorithm of random generators, we wonder whether if the numbers generated are truly random and how computing random numbers actually work behind the scenes. More extensively, we will apply this concept of random through a real life application: Poker. This project’s objective is to create a program that calculates probabilities of obtaining certain types of hands in Poker (i.e. royal flush, straight flush, high card). We will incorporate a lot more timing analysis and more practice of OOP applied to real life objects.

This project requires a lot of classes to be implemented. Initially, we are to create the basic classes representing a Poker game. These include creating a Card class (representing each playing card), a Deck class (a 52 card deck), and a Hand class (representing a player holding usually 5 or 7 cards in a Poker game). We must also create supplementary classes such as the Odds class and the Main class. The Odds class is a class containing static methods that have statistical calculations on the odds of winning a game based on a certain hand as well as comparing two different ways of the average percentage: 1) exhaustive method, and 2) stochastic method. These static methods will be useful for performing timing experiments and analyses when done implementing the Card, Hand, and Deck classes.

In addition to the assignment, we must also implement the different classes representing each distinctive random generator: Java’s Random generator, our “Worst” Random generator, and our “Best” random generator. For the last 10% of the portion of the assignment, we were to analyze another casino game similar to Poker: Texas Hold’em.

The key to getting this assignment working is ensuring that each and every method in the Card, Deck, and Hand works correctly and efficiently. We must also ensure that the different random generators are working correctly and efficiently. Theoretically, when comparing the different random generators, our “Best” Random generator should produce truly random numbers and thus no bias. Java’s Random generator would then come next followed by our “Worst” Random generator.

We believe that me and my partner will enjoy this assignment despite the amount of workload. The program could be useful for future references, such as that if we ever plan to play a Poker game and obtain better odds of winning Poker games in a casino.