Ryan Romano
CS II
Professor Sommers
4/26/15

Set Project

The experiment is to play 10,000 games of Set to find the average number of sets from a random collection of 12 cards from a deck, and the average number of cards leftover to estimate the actual values. In order to do this, the program written for the experiment first sets int games and double totalSets, totalCards, avgNumSets, and avgNumCards to 0. After starting a for loop that repeats 10,000 times, it creates an object of Game which generates 81 cards in Deck and adds 12 cards from Deck to Table. The variable games is increased by one and method game.numSets() searches for sets in Table and records the number it finds to average. Afterwards, the game is played until there are no more cards left in Deck. The number of cards remaining in Table is recorded and a new average is created. After 10,000 games, the final averages come out to be 2.7873 sets in Table per game and 7.7022 cards left per game. I ran the experiment four more times to get the values of 2.7873, 2.7883, 2.7981, 2.8012, 2.7876 sets and 7.7022, 7.7022, 7.6791, 7.6866, 7.6848 cards remaining. With the exception of one pair of cards, none of the values were the same past the digit. averaging the five values, I get 2.7925 sets and 7.69098 cards. I could not get even the 10ths digit to be consistent so I decided to try 100,000 games fives times too. I got the values of 2.78221, 2.77845, 2.7921, 2.78785, 2.78283 sets and 7.68045, 7.69215, 7.6851, 7.69092, 7.69068 with an average of averages of 2.784688 sets and 7.68786 cards. This time, I got a consistent 2.7 sets and 7.6 cards to one decimal place. The hundredths decimal place is almost consistent which brings the estimated values of 2.78 sets and 7.69 cards. That means I can guarantee the tens and the tenths place to be the same each time and almost the hundredths place. In other words, it's accurate up to one significant digit and almost two. I could not get a more accurate estimate because the number of games required to get a more accurate estimate takes far too much time to to execute.

If I were to do this project again, one thing I would do differently is write more readable code. I stuck with the same coding style throughout the project to stay consistent for anyone who would read my code in the future but frankly, it was a tough decision. I like the structure of the project and how each Java class could be independant from one another. This made the program easier to create and test, and I would make a project this way again. As for the IDE, I would be happy to switch to anything besides DrJava because using DrJava was the hardest part of the project. Sometimes the debugger would stop working and DrJava could no longer see files. Other times DrJava would not compile a file because it was a different version or even not open at all. The most annoying thing was that I couldn't view two files at once. That made managing multiple files incredibly difficult. My best decision was to ask for

help when I needed it instead of staring at the screen for hours dazed and confused. I won't allow stubbornness to get the better of me.

I learned a lot while making a project of this magnitude. I saw how organized a program could be by separating each part into a class with its own file instead of trying to program one big file. My understanding of ArrayLists, Linked Lists, and its applications was solidified. One of the most important things I learned was to create and use tests for everything. For the Set project, tests were necessary to make sure the program worked the way the professor wanted. I needed to make sure every test passed in order to reach that goal. Not only were tests used to fulfil the requirements, it also helped me in a lot of ways. There are countless times I thought my program was working when something I changed broke another use case. I could abstract the process by making tests and tell myself exactly what I want to achieve from the program instead of coding and hoping it matches the specifications. I also learned how to better manage my time. Splitting up the project into separate parts with different due dates gave me a better understanding of how a professional would complete a big project. Writing a program like this for the first time would not be fun. I aim to gain the skills and knowledge of a professional for my future academic and career endeavors.