Noah Jaffe
CMSC 331 – Dr. Bryan Wilkinson
12/7/2017

Semester Project

I chose Java to create the game because of my experience with the language, and because object oriented programming should be a simple way to accomplish part one. I chose Lua to do the analyzation because of my experience with the language, and a scripting language should be the best way to accomplish part two. My first programming language was Java. Most of my computer science courses have been taught with Java, so it has been my favorite language so far. I felt that it would take me longer to write the main part of the program with Lua or Scheme because I am rather inexperienced with them. I also figured that because I am writing a simulation of objects, using with an object-oriented language would be a simple and logical design. If part one was done with a functional or a scripting programming language it would be completely different, yet possibly shorter. However, with Java the code would be simple to read and write. I originally tried to use Scheme for part two but was unable to analyze the data as it was read in the way I wanted to, so I switched to Lua. I was able to keep it simple for Lua because I didn't have to do any recursion and it was still fast for parsing large files.

Java is missing the ability to give multiple returns. For example, when I was figuring out how to get the next move from a player. One method would be to return two card objects then find the two cards on the board, but that would make the algorithm have a time of O(n). If I just returned two locations on the board, then the algorithm would have a time of O(1). An ideal way to get this would be to return four

numbers as the first and second card row and column positions. However, Java does not support multiple returns and assignments, so rather than writing a Location class, I just returned an array of integers. The code wasn't pretty, but it was still easily readable and fast.

I am certain Lua has an easier way to create struct like datatypes as I was trying to create a struct that could store each statistic in its own variable. I had issues figuring that out so I just created a two dimensional table which might make the code a bit hard to follow, but I documented the code well enough so it should not be an issue.

I believe that having the project split into two coding sections is best. Having the game simulation in one language keeps it simple and neat. If it were possible I would have written the whole project with one language. I could have created a simple class that held game statistics, a class to hold an array of those, and a few lines to output the results in Driver.java.