**Memorandum 4**

**From:** Justin Pajak, Patrick Creaven, Carter Goldman, Raytheon Preternship Team

**To:** Dr. Matthew Morrison, Assistant Teaching Professor, Department of Computer Science and Engineering, University of Notre Dame

**CC:** John Mallinger, Deputy Chief Engineer, Next Gen GPS Ground System, Raytheon Technologies

Below is an overview of what we have done so far with the project, and our plans for the future.

**What were the goals for this week?**

1. Get satellite data file and figure out how to parse and implement it in our project
2. Write code for the conversion from spherical orbit data to xyz relative to the center of the earth
3. Utilize a clock library in our program in order the display a satellite’s changing location over time, change the position of the satellites in the graph

**What was accomplished this week?**

We accomplished goal three, which was utilizing a clock library to display the satellite locations changing over the course of time. At the moment, it updates the satellite locations every second based upon their orbital paths.

Goal one proved to be too difficult to implement within both the scale and scope of our project. All of the C++ libraries available for parsing and calculating orbits utilizing TLE satellite data (the international standard) proved to be too difficult to compile and make use of on the student machines, as none of us have superuser privileges, so we cannot install libraries. Furthermore, any programs we attempted to use ran into that same issue. While we could have opted to write the code to calculate the positions ourselves, we instead just created some hypothetical satellites and gave them “ideal” orbit patterns. While it does not simulate real life orbital patterns as closely as a library utilizing algorithms written by aerospace engineers would, the orbital behavior we programmed is realistic enough to still show off the main functionality of our data structures.

Since goal two was contingent upon us completing goal one, we did not complete it either. Instead, we seed each satellite with an initial xyz position relative to the center of the earth, then we calculate subsequent positions based on a simple circle equation.

**Goals for next week**

1. Make the Graph\_Sat class more concise by eliminating redundant functions
   1. Change graph\_sat to graph\_structure so we can add ground stations as well
2. Fix the Dijkstra’s algorithm function in the Graph\_Sat class so it doesn’t segfault when there are more than 2 satellites in the constellation
3. Create a constellation of satellites with a semi-realistic position relative to the center of the earth and determine their orbits
4. Create a user interface for the project

**Critical dependencies, open problems, or other things to be aware of for next week?**

1. We are nearing the end of the semester with the Final Project Presentation next week, so it is important to prioritize the main goals of the project, and determine what still is and is not within scope for this final week.
2. We may have to completely rewrite the Dijsktra’s algorithm code if we can’t get Professor Morrison’s code to integrate with our Graph\_Sat class

**How many hours were spent on each goal noted above?**

1. Justin Pajak - 8 hours
2. Patrick Creaven - 8 hours
3. Carter Goldman - 8 hours

Very Respectfully,

Justin Pajak

Patrick Creaven

Carter Goldman

Raytheon Preternship Team

University of Notre Dame

Department of Computer Science and Engineering