**Code Review Grading Rubric**

Total Points: 500 / 500 points

Week: 26 Oct - 2 Nov

Team Name: Raytheon

Preternship Software Engineers: Justin Pajak, Patrick Creaven, Carter Goldman

Preternship Project Manager: Ryan Farrow

Proposed Change List:

* Fix to Project Requirements section
* Completion of goals listed
  + Conduct research on satellite technical details
  + Start implementation of classes
  + Schedule meeting

Deliverables (250 / 250 points) :

* Basic structure for classes present (benchmark passed)
* Be careful how you use add\_edge. Figure out a way to keep track of satellite indexes in graph
* Consider adding a .gitignore file to suppress exe and obj folders being pushed to repository
* Code tested with a sample Satellite object created, functions called and printed to console for accuracy
* Also tested with two Satellite objects created and constellation functions tested on those

Design ( 150 / 150 points) :

* Code Clarity-
  + Logical level of abstraction with classes, not overdeveloped
* Code Intention -
  + Code is readable and easily accessible through classes defined in include folder
* Code Integration -
  + Good integration powered by the adaptability of the Graph class. One point that needs work is enumerating which satellite is which in the graph, but this is work that can be done next week.

Spiral Software Development (100 / 100 points):

* Objectives -
  + Get satellite data file and figure out how to parse it
  + Write code for the conversion from spherical orbit data to xyz relative to the center of the earth
  + Enumerate satellite indexes to make code more readable
* Risks and Alternatives -
  + Satellite data file ends up being very challenging to use in our main program.
    - Alternative: potentially find a different data set or convert the file to a different format to fix.
  + Which satellites can communicate with each other, which can’t
    - Alternative: which satellite can see another satellite (line of sight)
  + May have difficulty defining line of sight
    - Alternative: account for curvature of Earth and use an equation to solve ambiguity
* Plan for Product Development -
  + Utilizing wget linux command to acquire the data on the satellites
    - Utilize function that parses required data based on some type of delimiter or value in the file (column number, etc.)
  + Using equations from research and data from parsed orbital datafile, perform conversions
  + Create an enum at the top of our project file that assigns each satellite variable’s name to a number starting at 0. This is used when creating edges in our graph to simulate distances between the satellites.