Go-Gitters Deliverable 5: Celestia

By: Lara Brooksbank, Jacob Mattox, Kyle Cooper, and Alexander Swanson

Fault Injection Overview:

We have an a single include file with multiple functions to make testing easier. The changes we made to the code are as follows:

1. Changed the circleArea function by removing the parenthesis surrounding r * r. This causes the code to return a number when a negative value is passed as the radius.

2. Changed the sphereArea function by removing the parenthesis surrounding r * r. This causes the code to return a number when a negative value is passed as the radius.

3. Changed the pfmod function to check for a zero value in the denominator of the input. This causes zero to be returned instead of nan.

4. Changed the clamp function to return any value under 1 as 0 and any value over 1 as 1. No longer returns a range and instead only produces 0 or 1.

5. Changed the sign function to return the inputted value when it is negative, instead of returning 1.

Report:

We had our easiest time of the entire project on this deliverable. Most of the code that we used is math based and is relatively easy to inject faults into a math function without ruining the entire test framework. There was something oddly satisfying about adding failure points to the code we spent so long trying make perfect.

Once again, time and planning were the most important barriers for this part of the project. Without an adequate test plan we mostly just stumbled through the creation of the fault cases. In our case, it was fortunate we chose a project that is built using a large portion of math. Had we been unable to understand more portions of the code we used then creating fault injections would have been considerably more difficult.