

Go-Gitters Deliverable 3: Celestia

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

1. Architecture:

The testing framework for Celestia was built in the following structure:

- TestAutomation - Holds all other directories in the framework
 - ◆ Celestia - Holds the Celestia project files cloned from github
 - ◆ docs - Holds the README.md
 - ◆ oracles - Not used but required in the framework
 - ◆ reports - Not used but required in the framework
 - ◆ scripts - Holds all scripts required for running the framework (one script for the user to run and helper scripts for the execution)
 - ◆ temp - Holds the text files that are output of scripts and an html file that is created and opened automatically to display results
 - ◆ testCases - Not used but required in the framework
 - ◆ testCasesExecutables - Not used but required in the framework

2. How To Run:

This framework was designed on Ubuntu 18.4 distribution of Linux and as such has Linux specific commands nested throughout. The test framework itself does not need a full build of the project but it does check for needed packages and installs them as necessary.

From your command line type the following:

```
git clone --recursive https://github.com/csci-362-fall-2018-01/Team-Go-Gitters.git
```

(This will recursively gather Celestia as a submodule)

```
cd Team-Go-Gitters/TestAutomation (Move to correct directory)
```

```
./scripts/runAllTests.sh (Starts the automated framework and executes an html with output)
```

3. 5 Test Cases:

3.1. Test 1:

- 3.1.1. Since Celestia was such a difficult project for us to build we decided that automating the check for the correct packages needed for the build.

3.2. Test 2:

- 3.2.1. Likewise this test makes sure the makefile required to build the project is found after executing the cmake (Celestia devs changed this on 11/26 - it previously configured in a different way)
- 3.3. Test 3:
 - 3.3.1. Method Tested: degToRad()
 - 3.3.2. Input: 720.0
 - 3.3.3. Expected Output: 12.5663706144
 - 3.3.4. Actual Output: 12.5663706144
- 3.4. Test 4:
 - 3.4.1. Method Tested: degToRad()
 - 3.4.2. Input: 0.0
 - 3.4.3. Expected Output: 0.0
 - 3.4.4. Actual Output: 0.0
- 3.5. Test 5:
 - 3.5.1. Method tested: degToRad()
 - 3.5.2. Input: -180.0
 - 3.5.3. Expected Output: -3.14159265358979323846
 - 3.5.4. Actual Output: -3.14159265358979323846

4. Report:

Our initial desire with Celestia was to test some of the graphical aspects of the project using a Python library that give you control over the GUI. At this point we are unable to get a functional test using the graphics because of the complexity for interacting with a GUI using only a script. Once we determined that it was less feasible to continue trying to make the graphics work, we decided that testing math functions would be the most helpful. Given that Celestia is a space exploration program, it heavily relies on its mathematical components to be correct.

From a team perspective, we are struggling to find time to really devote to the project. Without firm instructions from an authority, we have a hard time determining how long a task should take, and we therefore spend many hours on things we should probably walk away from. Working with a group of peers is difficult when no one takes the initiative to lead the group.