Lab 3: Modeling Transformations

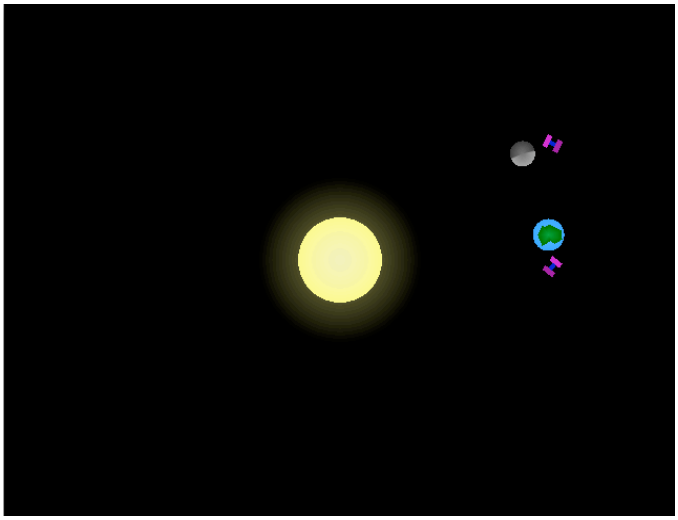

Fig 1: Screenshot of a solar body with a planet and moon, each orbited by a pesky satellite.

This lab will explore modeling transformations and transformation hierarchies. The support code provided for this lab initially displays a sun, a planet, and a moon all in a row. Fig 1 Shows the completed scene with satellites added in. When the lab is complete, watch the solar system in action!

Your task is to implement the necessary transform hierarchy to produce the desired scene above. You are provided with the functions `drawSun()`, `drawPlanet()`, `drawMoon()`, and `drawSatellite()` to use to draw the scene. Implement the `refresh()` function to update the global celestial variables (`planetRevolve`, `planetRotate`, `moonRevolve`, etc.). Also implement the display callback `disp()` to render the scene. In order to do this you will have to implement the functions `rotateZ(angle)` and `translateX(x)`. Then inside `disp()` you must use these two functions. Do *not* call `glTranslate*` or `glRotate*`.

You are free to chose the radii of orbits and the rotation and revolution rates of the celestial bodies. The only requirement on the system is that the moon orbit the planet which orbits the sun, the moon and planet each have a satellite, and the *light side* of the moon must always face (point toward) the planet (see the figure above). Each body must both rotate and revolve.

Try not to do a transformation, do another, and then *undo* the first. This is a great way to accumulate floating point errors and lose precision. The optimal solution to this lab should not call `translateX(x)` more than 4 times (try hard for this, but it is not required); in order to do this you may need to use a few OpenGL calls, other than `glTranslate*` and `glRotate*`, which you must *not* call.