Solar System Genereation and Modelling

Cole Risch[1]

[1]crisch@pdx.edu

November 2, 2025

1 Project Topic Area

This project is meant to provide me with some experience generating 3D models in rust and animating their movements. If possible these animations would be driven by real physics. Inital rendering framework will be cribbed from the "3d_shapes" example in the Bevy Repository.

1.1 Known Problems

I will NOT be making any attempt to model n-body physics here. Minimum implementation will be "on rails" where planets and planetoids follow a pre-scripted orbit regardless of any physics involved. Maximum implementaiton would likely involve "sphere of influence" implementation of 2 body physics. This sort of implementation computes the radius at which the gravitational pull of the sun and the gravitational pull of the planet are equal (a Lagrange point). All small objects in "deep space" use the Sun as their only gravitational force, and all small objects crossing one of the Lagrange radii would then use the repective planet as a gravitational force.

2 Project Vision

This project will be implemented using the BEVY framework. This should absolutely qualify for the usage of "one or more Rust crates." The Bevy framework enforeces a modular code style, so this shouldn't be a problem. Realistically, this project *should* evolve through several stages:

1. Initial rendering of a Large Sun and one or more smaller planets such that no body intersects any other body (no planets inside the sun, no planets colliding with one another)
2. animation of the Large Sun and one or more planets rotating about their own axis at different speeds

3. animation of one or more planets rotating about the Large Sun at different speeds (bonus points for retrograde, elliptical, and orbital incliination)

4. adding controls to the simulation that allow it to be stopped, started, and reset

5. adding controls to the simulation that control the number of planets to be created, as well as the radius of the Sun

6. other controls to be implemented as possible. For instance, it would be nice to be able to control rotation, radius and orbital speed on a per-planet basis, but I am not certain that is possible.

7. attempting to implement actual physics for orbits based upon an intial mass of the solar system and then assigning masses to all bodies based on accepted astronomical distributions (e.g. solar mass being 98-99.9% of the mass of the whole solar system)

8. attempt to a random texture map to any created planet

9. attempt to render complex planetoids like Oort cloud, asteroid belts, or moons

Please note that any item in the previous list that begins with the word "attempt" implies a task that I am not certain I will be able to accomplish.

3 GIT Repo location

https://github.com/metalflow/CS_523_Solar_System_Model.git