CS 450/550 Fall 2020

Solar System

Zan Zhang

zhangza@oregonstate.edu

A solar system, do this:

- Scale everything suitably.
- Therefore you have to exaggerate both of:
    1. Planet diameter
    2. Planet orbital radius

       By using these data https://www.bobthealien.co.uk/solarsystem/table.htm

       And find a good rate
- When you exaggerate, keep everything to scale, except, possibly the Sun's diameter. If Planet A's diameter is twice as big as Planet B's in real life, make sure it is still twice as big in your scaled-life. And so on.
- You need to follow Kepler's Third law of planetary motion:
  Orbital Period is proportional to OrbitalRadius$^{3/2}$ = pow(OrbitalRadius, 3./2.)
  Get this right!
- But, you will also need to scale the orbital periods, or the outer planets will not appear to move at all. Again, keep the orbital periods proportional. If Planet A's orbital period is twice as long as Planet B's in Kepler's Third Law, make sure it is still twice as long in your scaled-life.
- Treat the planets' orbits as circular.
- Be careful of Temporal Aliasing (the wagon-wheel-spokes-rotating-backwards effect) on the fast-orbiting inner planets. This is part of what makes this project interesting. Pick an orbital period scale factor where the inner planets do not move too quickly, but the outer planets do not move too slowly.
- Find good textures or good coloring for the sun and all the planets. NASA has some at: https://nasa3d.arc.nasa.gov/images.
- Have a good outside viewing option where we can see everything.
- Do point-light lighting from the Sun. Throw a little ambient in there too.
- Do not call the Sphere( ) function directly from inside of your Display( ) callback. You would end up paying for all those sine and cosine calls each display update, which would ruin your graphics performance. Instead, put each planet into its own Display List and then glCallList( ) it from within Display( ).