CSC433/533 Computer Graphics - Fall 2014
Programming Assignment #3: Solar System Simulation

Introduction

Illumination and surface reflectance models form the basis of realistic computer graphics imagery. In this assignment you will explore 3-D graphics with lighting models, surface rendering, and texture mapping.

Problem

Write a solar system simulation program that displays the Sun and the eight* planets, plus the Earth's moon and the rings of Saturn. Animate this simulation so that the planets rotate about the Sun with correct speed and trajectory relative to one another. The user may zoom in or out of the scene, pan in all directions, and speed up or slow down the simulation.

* There is some controversy about whether Pluto is truly a planet. Since Pluto's highly eccentric orbit poses difficulty in this simulation, we will take the practical view that it is *not*.

Allow the user to toggle between wireframe objects, flat shading, smooth shading, and texture mapping. Assume that illumination primarily comes from the Sun, but provide some ambient light (so that the nighttime planetary faces are not pitch black).

Program usage: `solar`

Implementation

- See *orbit.cpp* for a working simulation of the Sun, Earth and Moon as wireframe objects. Modify this code to include the other planets. Use a different color for each planet, provide text labels, and display the orbital tracks. Allow the user to change the wireframe resolution and animation speed (including a single step mode). Provide controls to toggle between wireframe objects, flat shaded surfaces, smooth shaded surfaces, and texture mapping.
- Make reasonable simplifying assumptions for your simulation: planets orbit at constant velocity in circular trajectories centered about the Sun, all orbits lie in the ecliptic, etc. Your simulation may begin with planets at any arbitrary point in their orbits (hey, they might be in that configuration someday ☺). Document your assumptions in the program header comment.
- Planetary texture maps will be provided in BMP format, along with sample code to read and display BMP images. Feel free to find better imagery for use in your program.
- Implement an easy-to-use interface for interacting with your simulation (changing the animation speed, camera viewpoint, etc.) with keyboard and mouse event handlers. Your interface should include a popup menu and help system.
- In addition to the OpenGL model-view transformations (*glTranslatef*, *glScalef*, *glRotatef*), you may use the OpenGL projection view transformations for establishing a perspective projection (*gluPerspective*) and positioning the virtual camera (*gluLookAt*).
- Extra credit features: toggle viewpoint from planet to planet, additional celestial bodies (moons, asteroid belts), etc.

Notes

- You are encouraged to find a partner and work in teams of two on this assignment.
- This assignment is intended as an exercise in OpenGL graphics programming. OpenGL/GLUT resources and code samples are provided on the course website (*http://www.mcs.sdsmt.edu/csc433*). You may use code from your instructor or the CSC433/533 textbook for programming assignments, but not other sources. Always give the author credit when using code written by others.
- When you are finished writing, testing, and debugging your program, turn it in using the MCS Web submit program. Submit only source code, not executable files or test data. If you write a multi-file program, submit it in a zip or tar archive with a Makefile.
- The MCS Web submit program is accessed via the MCS Web page (*http://www.mcs.sdsmt.edu*), by selecting the list item on the left entitled “Submit it!”. Usage is self-explanatory: enter your name, choose the instructor and click “Select Instructor”, choose the appropriate course (CSC433/533), browse to the filename you wish to submit, and click “Upload”.
- Submit your program by midnight on the due date (Tuesday November 18) in order to receive credit for this assignment. Late programs will not be accepted for partial credit unless prior arrangements have been made with the instructor. If you have any problems with the submit program, report them to your instructor and submit your program by email instead.
- To receive full credit, your code must be readable, modular, nicely formatted, and adequately documented, as well as complete and correct. Try to make it reasonably efficient in terms of both execution time and space utilization. Regardless of your development environment, your program must compile and run successfully under the GNU C++ compiler and OpenGL/GLUT libraries available on the Linux systems in the MCS Linux Lab. If your program does not run correctly, indicate why. This will make it easier to give you partial credit.

Orbital parameters

Planetary radii are given in kilometers. Average planetary distances from the Sun are given in millions of kilometers. The revolution period (year) is given in days, and the rotation period (day) in hours.

	radius	**distance**	**year**	**day**
Sun	696000	0	0	25
Mercury	2439	58	88	1416
Venus	6052	108	225	5832
Earth	6378	150	365	24
Mars	3394	228	687	24.6
Jupiter	71398	779	4332	9.8
Saturn*	60270	1424	10761	10.2
Uranus	25550	2867	30682	15.5
Neptune	24750	4492	60195	15.8
Moon	1738	0.384**	27.3	27.3

* Saturn’s brightest rings range from 90000 km to 140000 km from the planet’s center.
** The Moon’s orbital distance from the Earth is 384400 km.