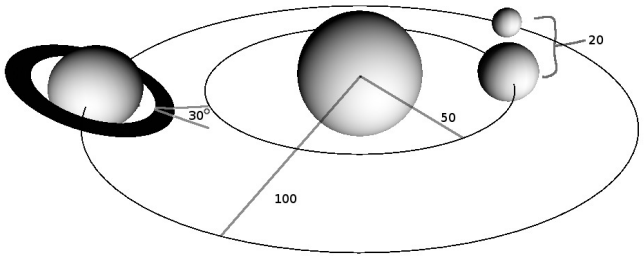
M. Stamminger, C. Weber *Erlangen,* ***Monday, 10.11.2014***

Computer Graphics - Programming Exercises

Assignment 4 [6 Points] (Transformations)

Now that you know all about transformations from the lecture, it is time to add some animated planets to the planet system in the `display` function of the main program. Note that we abandon the matrix stack completely. Instead we bind the proper matrices to our shaders. That means you have to change your lighting shaders (sun & phong) slightly so that the transformations and colorings are taken into account during lighting computations.

a) The scene to be set up is shown in the figure below.



Apart from the sun, which resides in the center of the coordinate system with a radius of 25, the scene should contain two more planets, the earth (radius 12) and the saturn (radius 16), which rotate around the sun with orbit radii of 50 and 100 respectively.
Furthermore, a moon (radius 6) should rotate around the earth on an orbit with radius 20 on a plane perpendicular to the line between earth and sun. The saturn should have his characteristic rings, which are tilted by 30° torwards the sun. Draw these rings using various line loops with slightly different radii. You can alter the color of adjacent rings to get a nicer effect. Don't forget to add normals to the ring vertices so that the rings are affected by the lighting.

For the animation, there again is a time parameter `t` which you already know from the previous assignments. The speed (i.e. the per-frame increment of `t`) can be in- and decreased by pressing '+/-'. However, the moon should rotate around the earth twice as fast as the earth rotates around the sun and the saturn should rotate around the sun half as fast as the earth does.

b) The sun can be drawn using the computations of the sun shader from the previous assignment. Yet because we use our own uniforms you will have to make some changes.
Lighting the rest of the solar system is to be done in the phong shader. However since the elements in the scene are now affected by transformation matrices, you have to perform the lighting computations in eye space instead of world space.

Additional Information

- There is no global parameter for the number of slices and stacks for the drawing of the spheres any more since the individual spheres vary in size and should thus be drawn with different (and suitable) amounts of slices and stacks.
- When using the sun shader, the sphere should be drawn using `glutSolidSphere`. When you use the drawSphere-function from assignment 1, you will see cracks in the mesh. However, the function is fine for the other planets, in fact, we need to draw them this way in a later assignment when it comes to textures.

Implementation Guidelines

The last assignments required you to set your coordinate systems in a very awkward way. Rejoice, these days are over. From now on you will manage your own M,V and P matrices and you will use the glm library. Creating and multiplying is now very intuitive and completely at your control.

Have a look at the binding of uniforms, we will extend this utility function over the course of the following exercises.

Note that future assignments will alway provide a minimalistic skeleton. If you wish to create a complete solar system, be prepared to extend this code or complete future assignments with your existing solutions.

Good Luck!

Your source code will be copied from your handin directory on:
Monday, 17.11.2014 14:00 pm
all subsequent changes cannot be taken into account!