

Assignment 2

CMSC 740, Spring 2006

Due: 2:00pm Tuesday, February 28, 2006

This assignment involves familiarization with OpenGL texturing and lighting, as well as simple animation. The assignment web-page is at www.cs.umd.edu/class/spring2006/cmcs740/assg2/. For this assignment, you should make the following extensions to your program from assignment 1.

- **Keyboard control:** Allow the user to move the viewer's position with the keyboard. Keys `w/s` should move the viewer forward/backward, `a/d` should move left/right, and `e/c` should move up/down.
(2 points)
- **Texturing:** We are providing you two bmp files, `Grass.bmp` and `Sky.bmp`. You may use these as textures, or generate your own. Map the grass texture to your terrain, with texture coordinates `u` and `v` each ranging from 0...1 from one end of the terrain to the other. For the sky, generate a large cube that encloses the entire scene (viewer and geometry) and map the sky texture to each face of the cube.
(3 points)
- **Lighting:** Light your scene using ambient lighting and a directional light. This will require you to specify normals for the vertices of your terrain. You can approximate the normal at each vertex by computing two approximate tangent vectors to the vertex and taking their cross product. The tangent vectors can be computed from the heights of the four adjacent vertices (you will need to special case the edge and corner vertices).

The directional lighting should not affect the sky box. You should `glDisable` the directional light before rendering the sky box, and `glEnable` it again before rendering the terrain.

(4 points)

- **Animation:** Make the sun rise and set. Render the sun as a yellow `gluSphere` floating over your terrain. To make the sun continuously rotate around the terrain, use `glutIdleFunc` to register an idle callback function – OpenGL will repeatedly call this function when it does not need to perform any other tasks. Each time this function is called, it should compute the amount of time since it was last called (for example, by using `glutGet(GLUT_ELAPSED_TIME)`), and increment a variable representing the sun's angle of rotation by a corresponding amount. This angle should be passed into a `glRotate` call in your draw function.

As the sun rotates, you should also change the direction of the directional light so that it appears to be coming from the sun. As the sun gets close to setting, the light intensity should gradually dim out, and gradually fade back in again as the sun rises. You should make the sky box change brightness as well, either by using an ambient light, or by directly altering its vertex colors.

(6 points)

(Total: 15 points)

[Assignment tips are on the next page ...]

Tips:

- Lighting and texturing each require multiple OpenGL parameters to be set up. You might want to start by copying the parameter initialization function calls from a working example.
- OpenGL requires normal vectors to be unit length. If you don't want to have to worry about this, you can `glEnable(GL_NORMALIZE)`, which will make OpenGL keep all normal vectors unit length for you.
- The color of lit geometry is determined by the color of the light and the geometry's material color, which can be set by `glMaterialfv`. You can also use the `glColorMaterial` and `glEnable(GL_COLOR_MATERIAL)` functions to allow the material color to be changed by `glColor` commands.
- When using texturing with lighting, you should set `glTexEnv` to `GL_MODULATE` rather than `GL_REPLACE`. This will cause the color of the texture to be influenced by the interpolated color from the vertices, which is what the OpenGL lighting is affecting. In this case I would recommend setting the material color of the vertices to white, allowing the geometry color to be determined by just the texture color and light color.