

HOMEWORK #5

A BUMP MAPPING SHADER

This assignment may be done in teams of two. **Due December 1, 2009 by 11:59PM.** The turnin name for this assignment is cs433_hw5. In this assignment you will extend Homework 4 by bump mapping the mobile plates in real-time using a shader program. In particular, your program should consider the following:

- (1) **Mobile Scene:** (20 pts) Recreate the mobile scenegraph exactly as in HW4, and use the same poly-line function to define the bump-map. All user interface capabilities outlined in HW4 should persist. You should be able to recycle the vast majority of your code from the previous assignment.
- (2) **Shader Bump Mapping:** (80 pts) Write a shader program, consisting of the appropriate vertex and fragment shaders, that applies a bump map to each plate based on the value of the poly-line function. Visually, your output should be very similar to what you observed in HW4, but we expect it will render considerably faster than a non-GPU solution. In particular, your shader program should:
 - Compute in real-time the perturbed normals to the surface of each plate, as specified by the bump map.
 - Apply the standard Phong (or Phong-Blinn) local illumination model to compute the diffuse and specular lighting terms.
 - Compute the perturbed normal and lighting on a per-pixel basis. The computations to perturb the normal and illuminate the plate should be done on each frame of the render loop, inside your shader program. (Hint: the bulk of your work should be in your fragment shader!)
- (3) **For Grads:** (20 pts) Allow the light position to be moved in the plane of the viewpoint, orthogonal to the line of sight. Recall that, in HW4, there was one point light whose position was fixed at the viewpoint. We now desire a light source that somewhat more flexible. Map the ‘A’, ‘D’, ‘W’, and ‘X’ keys to move the light position in small increments left, right, up, and down, respectively, in the plane orthogonal to the view axis. The ‘S’ key should re-center the light back to its default position at the viewpoint. (Hint: it is not necessary to use `GL_LIGHTS` to achieve this behavior.)

If you implemented HW4 using a shader program that satisfies the requirements above, you may resubmit that program as HW5. Please provide a **readme** stating such with your turn-in. Grads should further make the requested extensions to support a movable light source. As always, extra features will be considered for modest extra credit, based on effort and coolness. Finally, feel free to use any code given in the lecture slides or demo programs to complete this assignment.