

Fall 2009
22C:151 Introduction to Computer Graphics
Assignment 9

Due: Friday November 20th at 11:59 pm

*(Note: November 20th is the Friday prior to the Thanksgiving break. The assignment is due then. However, for this one assignment **only**, you may submit assignments after the deadline until Friday November 27th with no penalty, and with no use of late days. You may not use late days in addition to this extension. Assignments submitted November 28th will receive **no credit**.)*

Goal: Add mipmapping while texturing and observe the difference between various texture interpolation schemes. Add projective shadows and get a basic idea how to incorporate shadow maps to your program.

Problem 1 (5 points): Change Homework 7 to add mipmapping on the “infinite” plane. You should only need to change (literally) 2 lines of code. Also add a keyboard callback (or menu entries) which allows you to change between the various minification interpolation constants (`GL_NEAREST_MIPMAP_NEAREST`, `GL_NEAREST_MIPMAP_LINEAR`, `GL_LINEAR_MIPMAP_NEAREST`, and `GL_LINEAR_MIPMAP_LINEAR`).

Problem 2 (10 points): Replace the textured cone and cube from Homework 7 with one of the complex illuminated objects from Homework 6. This object should cast a shadow on the floor using projective shadowing. Make sure you can still move the light via the trackball and have the shadow behave appropriately. Keep the texture on the floor (at least in the unshadowed regions).

Hint: You should not project your shadow exactly onto the floor plane, but rather a slightly offset plane, so that there is no z-tearing (conflicts between the floor and the shadow in the z-buffer).

Problem 3 (10 points): Use a *projective texture* to texture your object with one of the PPMs from the class web page. Make sure you modulate the texture (`GL_MODULATE` in `glTexEnvf`) so you see both the lighting and the texture.

Note: as you rotate your object, the texture should remain fixed on the object. (e.g., `GL_OBJECT_LINEAR` plane for automatic texture generation).

Problem 4 (5 points): Render a *depth map* (a.k.a *shadow map*) from the light’s point of view. Copy this into a texture (use `glCopyTexImage2D()`) and render it onto a square in the lower right hand corner of the viewport. Make sure the depth map updates as you rotate your light or object.

Note: code for rendering a texture in the lower right corner of the screen is online.

Extra Credit (8 points): Automatically generate texture coordinates (i.e., a projective texture) to add shadows via the shadow map. With these shadows enabled you should see shadows on *both* the floor and the object! Add a keyboard toggle or menu option to allow switching between projective shadows and shadow maps. Make sure you still display the shadow map in the lower right corner of the window as per Problem 4. See p. 459 for more information. You may disable the checker texture on your floor for this (alternatively, use multitexturing).

NOTE: A “README” file is required in order to get full credit! It is worth the 2 or 3 minutes it takes to write such a file, as it guarantees we know how to compile and run your program, and you can make note of any odd behavior or strange bugs (which may mask required functionality).