

# Comp770 – Spring 2007

Programming Assignment #3

Due 3/26/07 (before 11:59:59pm)

**Objective:** In Assignment 1 you concentrated on interacting with a 3D world. In this Assignment you will use lighting, texturing, and programmable shaders to enhance the appearance of your 3D world. In addition, you will become more familiar with additional features of OpenGL that are also found in other graphics API's such as DirectX.

## Minimal requirements (worth 80%):

1. Create a sky with a sky box environment map. A sky box is a cube that surrounds the scene with texture maps on the faces that represent distant parts of the scene. The faces of any cube map texture will work for a sky box.
2. Use a heightfield as the base terrain. Terragen is a freeware program that you can use to generate terrains, or you may choose to create one yourself. There are many tutorials on the Internet for generating terrains. Alternatively you may want to load real world GIS data. The terrain must have some steep areas (to test #7).
3. Compute vertex normals for your heightfield so that you can have smooth shading.
4. Compute per-pixel lighting for the terrain with a fragment shader.
5. Apply a texture to the terrain and/or color the terrain based on elevation.
6. Add water to the terrain by rendering a single transparent quad at a given elevation level.
7. Create a cow avatar. Provide a keyboard interface for roaming across the terrain. The cow should maintain a natural orientation while moving over the terrain, i.e. it should never be oriented in a way that it would tip over. To determine when the cow will tip over project its center of gravity and the convex hull of its feet onto the plane perpendicular to the direction of gravity. The cow will fall when the center of gravity falls outside the projected convex hull. In addition the cow should not be allowed to go on terrain that is too steep or into the water. Provide a first person view for the cow.
8. Provide at least one overhead camera that is free to move about the scene
9. Turn in a brief write up of what you did and how you did it.

## Extras:

1. (5-10 pts) Use the alpha test to add billboards for grass, trees, plants, fences, or other complex objects to the scene. (Note that you may want to place the items in the scene by clicking where on the terrain you want them. You could read back the depth of the clicked pixel and use `gluUnproject()` to find the 3D point)
2. (5-10 pts) Create a procedural texture with a programmable shader. For example, you could place a marble teapot in your scene. If you want to use the noise, note that the `noise()` function in GLSL hasn't been implemented. You will have to generate your own noise from a texture. RenderMonkey has a 3D noise texture that you could use.
3. (5 pts) Animate the water by manipulating the vertices with a vertex program. One way to do this is to sum several sine waves of varying directions, amplitudes and frequencies.
4. (5 pts) Compute normals for the water ripples. The normal is just the gradient of the overall function represented by the surface.
5. (5 pts) Add reflections to the water using either a cube map or sphere map.
6. (10 pts) Use shadow maps to add shadows to the scene. Create a depth texture with render-to-texture or by copying out the frame buffer. Update the shadow map when the light position or the scene changes. Use a small amount of polygon offset to control self-shadowing artifacts.
7. (5-10 pts) Add complex materials to the scene using a shader that takes multiple texture maps as input. For example, you could have separate textures to control the diffuse and specular terms in your lighting equation. Points will be awarded depending on the sophistication of the shader.
8. (5-10 pts) Path planning. Provide a walk-to-here interface that allows the user to click on the terrain and the cow will move to that position. The cow should stop if the terrain is too steep or if there is water along the direct line path. 5 more points will be given if the cow finds an alternate route.
9. (Variable) Some idea of your own. Be sure to consult with the instructor.

**Policies:** Everyone must turn in their own assignment. You can collaborate with others, but any work that you

turn in should be your own. Turn in your work by emailing an archived and compressed version of it (source and executable) to the instructor along with instructions for running your code.