CMSC 23700 Introduction to Computer Graphics Project 6
Autumn 2015 November 21, 2015

Terrain rendering (part 2)

Due: December 8, 2015 (12 noon)

1 Summary

Project 6 is the second part of the final project. In this part, you will add visual embellishments, of your own choosing, to your Project 4 code. For purposes of this project, you may choose to develop your features for a specific map, such as the Grand Canyon, but your code should support the Project-4 features for all of the test maps.

2 Completing Project 5

The first part of the project is to make sure that your Project 5 code is working. The Project 5 features will be evaluated when grading Project 6, so take the time to address any outstanding issues.

3 Embellishments

In this section, we list some possible ideas for visual effects that you might add to your viewer, but you should not feel constrained by this list. We have annotated these suggestions with an estimate as to their difficulty of implementation.

You may also find `http://vterrain.org` as a useful source of information.[1]

3.1 Detail textures [Easy]

To make the surface of the terrain look more realistic, you can blend in a *detail texture*, which is essentially a noise texture. The detail texture is used to modulate the surface color close to the camera (say out to 80 to 100 meters for a ground viewer). A detail texture will be added to the `final-project/data` directories in your repositories. The texture is allocated as an RGB image, you may want to copy it to an RGBA representation with a 40-50% alpha channel, which will mute its effect somewhat. You can also use alpha blending to get a smooth transition from textured polygons to untextured ones.

[1] It appears that their domain registry just expired on the 18th. I hope that they will get back online, but you may have to use the wayback machine to get access to it.

3.2 Skybox [Easy]

Outdoor rendering engines use skyboxes (or skydomes) to provide a backdrop for the terrain. The basic idea is define a cube with the edges of the terrain mesh as sides and to render it with a texture of distant mountains, clouds, etc. Fancier skyboxes will animate the clouds. If you add a skybox, you will need to adjust the way that you render fog so that the skybox is not obscured by the fog.

3.3 Procedural detail [Medium]

For most of the map data sets, the geometric detail is fairly crude (especially for a ground viewer). For example, there are 60m between height field samples in the Grand Canyon map. To make the terrain look more realistic, you can use tessellation shaders and fractal noise to add geometric detail close to the viewer.

3.4 Shadow texture [Medium]

The map format includes the direction of the sun (specifically, the vector for a directional light). Once can precompute a shadowmap from the height field. For each grid cell in the height field, you should compute a 2×2 texture sample (*i.e.*, the shadow texture will have size $2^{n+1} \times 2^{n+1}$ for a 2^n wide height field. Use one byte per texel, with 0 meaning shadowed and 255 meaning lit. You can determine if a texel is in shadow by casting a ray from its center back towards the direction of the light (see Figure 1). Using this information, you precompute the lighting information and

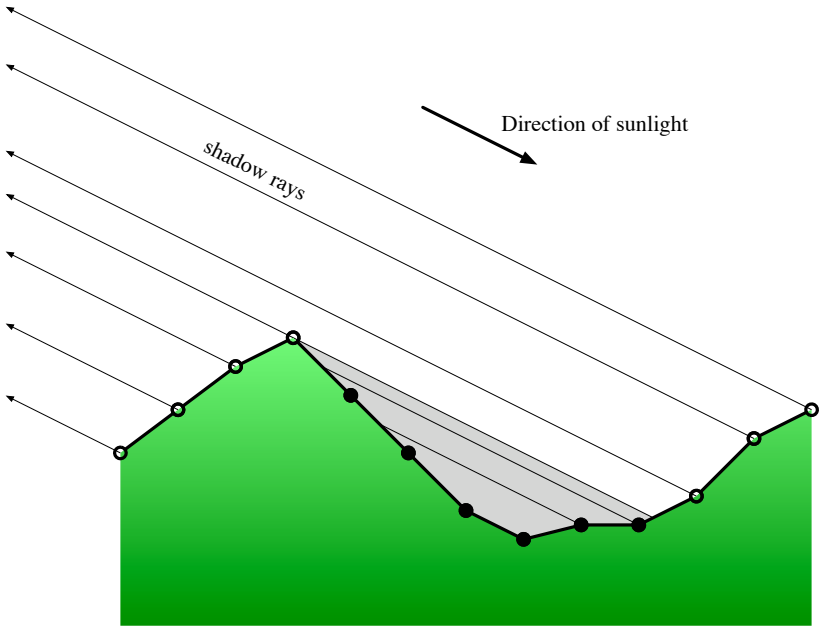

Figure 1: Computing shadows by ray casting

luminance map for the terrain. A better result can be obtained by computing several samples per texel and averaging their values.

3.5 Shadows [Hard]

As we have discussed, shadows play an important rôle in making scenes look more realistic. Implementing shadows for a large-scale terrain, however, is challenging. Stencil shadows may work well for the terrain, since you can compute the shadow volumes at load time. It is also possible to use shadow mapping, but you will probably need multiple maps.

3.6 Vegetation [Varying]

Terrain usually has trees, shrubs, grass etc. Trees and shrubs might be represented by models that are placed on the terrain, while grass can be rendered using a particle system (each blade of grass is modeled as a particle). To handle large numbers of trees, you may want to use impostors.

3.7 Particle effects [Medium]

Particle systems are low-cost physics simulations that are used to render *fuzzy phenomena*, such as smoke, fire, liquids, explosions, *etc.*. In the context of this project, particle effects could be used to implement weather effects (rain or snow); erupting volcanoes and lava; or flocking birds.

3.8 Water animation [Hard]

Some of the maps have major bodies of water. The surface of the lakes is flat and static. We can make it more interesting by adding waves to the water surface. For such cells, there is a file called `water.png`, which is a black-and-white image that you can use as a mask to determine which vertices in the cell are in water and which are not.

4 Generating maps

Please contact Prof. Reppy if you need access to the tools for generating the map representation.

5 Submission

Project 6 is due on December 8, 2015 at 12 noon. We will be having project demos starting at 1:30pm on the 8th.

History

2015-11-21 Original version.