

## Project 4 - Shadow Volumes (updated on 4/16)

### 1 Essential changes from the originally posted version

'slower' and 'faster' menu items (to control the speed of animation) added

### 2 Goal

Draw the same kind of scene as in Project 1 (ray tracer) but this time taking advantage of the graphics hardware. If you run your program on a machine with a good enough graphics card, you should obtain interactive frame rates.

### 3 Input File

The input file will specify the location of the light source (3 floats) and the number of primitives (spheres and triangles). Your programs should work for any number of primitives. For each of the spheres, we'll list the coordinates of the center (3 floats), the radius (one more float) and the color (specified as R, G and B components - 3 floats between 0 and 1). All this data will be preceded with an 'S' (for sphere, of course). Triangle data will be preceded by a 'T' and will contain nine floats representing the coordinates of three vertices and then three more floats representing color (RGB components, like for spheres). To simplify things, we'll only use ambient and diffuse lighting. This color for each primitive is to be used as both diffuse and ambient color. Use .2 for the ambient light intensity and .8 for diffuse light intensity. The light source location will be specified in the same coordinate system as the primitives (if you transform the scene when displaying, you should transform the light in the same manner).

We promise that all the spheres and triangles will be contained in the axis-aligned cube centered at the origin with side length of 2 (i.e. extending from  $-1$  to  $1$  in each coordinate). Similarly, the light location is not going to be 'nasty': it will make the shadow volumes extend towards the back (not towards the viewer).

### 4 Requirements

You should program a menu tied to the left mouse button. in two items The menu should contain the following items:

1. **Change mode.** It should enable the user to cycle through three modes: show shadows (which should render the spheres with shadows), hide shadows (just plain opengl rendering, no shadows) and show shadow volumes

(which would render the scene without shadows but show shadow volumes using your favourite color).

2. **Toggle animation.** The light should start moving, more precisely rotating around the view direction (which is the z-axis in the skeleton code and, I guess, you won't be changing it). Of course, this should make the shadows change. Let me stress that only light is to be moving. The spheres and triangles should stay in place.
3. **Faster.** Make the animation faster
4. **Slower.** Slow down animation

Of course, your programs should be able to convince us that you are doing what needs to be done - the scene should appear as reasonably large (take advantage of the promise at the end of Section 3, you may just replace the cube drawn in the sample code with your scene).

## 5 Remarks

There is no requirement to use display lists in this project. Use smooth shading. The show shadow volumes mode may be useful for visual debugging your shadow polygon computation routine.

## 6 New OpenGL commands needed

It's probably convenient to use quad strips for the shadow volumes. See `glBegin(GL_QUAD_STRIP);`

For setting the culling parameters: `glCullFace, glEnable/glDisable(GL_CULL_FACE)`.

For stencil buffer setup/use: `glClear` with `GL_STENCIL_BUFFER_BIT` argument, `glStencilOp, glStencilFunc, glClearStencil, glEnable/glDisable` with `GL_STENCIL_TEST` argument.

For depth buffer setup/use: `glDepthFunc, glEnable/glDisable (GL_DEPTH_TEST)`.

For enabling/disabling buffers for writing: `gl[Depth/Color]Mask`.

To set up lighting coefficients: `glLight*`.

To set up material properties: `glMaterial*`.

To add stencil buffer to your display: `glutInitDisplayMode`; include `GLUT_STENCIL` in the bitwise OR.

For drawing spheres: `gluSphere, gluNewQuadric`.

## 7 Grading

Correct handling of spheres is enough for full credit (100). Triangles are additional all-or-nothing extra credit of 30 points. If you decide to run for the extra

credit, you must add another item, 'show/hide triangles' to your menu. Hide triangles means that only spheres (and/or their shadows or shadow volumes, depending on the mode) should show up. Switching to show triangles should display both triangles and spheres.

If you are not interested in extra credit, just ignore the triangles specified in the input file and render only spheres (however, your code has to be able to deal also with input files containing triangles; they just don't need to be displayed).

1. Primitives rendered correctly (not necessarily with shadows) - 25 points
2. Shadow volumes correct (will be tested by viewing in 'show shadow volumes' mode)- 25 points
4. Shadows correct - 25 points
5. Lighting equation coefficients set up correctly (you can test this by rendering one sphere with shadow volumes turned on - if the lighting appears smooth, you are fine) - 25 points; however, no credit if shadows do not work.

## **8 Due date – NOTE THE CHANGE OF THE LATE POLICY!**

This time, there is a hard and absolute deadline: Sunday April 27 midnight. Late submissions will not be graded.