CG Programming III – Assignmnet #4 (Shadow volumes)
Due on 5/14/2007

For this assignment, you will need to reimplement assignment #2 using shadow volumes. A simple scene with multiple objects will be rendered. Any sort of lighting model can be used. All of the objects must cast shadows, as appropriate for the position of the light, on the other objects. Rather than using simple planar projection shadows, shadow volumes are to be used.

- Implement hard shadows using stencil buffer shadow volumes.
- Shadowed scene must include at least one moving light source.
  - Light source may be either a point light or a spot light.
  - If a spot light is used, draw a wireframe outline of the light's view frustum or the polygons used to enclose the spot light.
- Shadowed scene must include at least two objects.
  - One object may be a ground plane.
  - One object must be non-convex object (i.e., an object that can show self-shadowing.

Additional points can be earned by implmenting one or more of the following items.

- Implement the "simple" version of ZP+.
  - The implementation need not include the crack avoidance algorithm described in section 5 of the ZP+ paper.

The following inputs must be implemented. In addition, the program must, in some way, communicate to the user how to use it.

- Escape must terminate the program.
- An input must be implemented to display the shadow volume polygons alpha blended onto the rest of the scene.
  - Front facing shadow volume polygons should be a different color than back facing shadow volume polygons.
  - This is a *good* debugging aid. I strongly recommend that you implement this early on!

Criteria	**Excellent**	**Good**	**Satisfactory**	**Marginal**	**Unacceptable**
Code Function	Program correctly implements all required graphical elements in a manner that is readily apparent when the program is executed. Appropriate algorithms and data structures are used.	Program implements all required graphical elements, but the operation of some elements may not be obvious. Appropriate algorithms and data structures are used.	Program implements all required graphical elements in some fashion. Algorithms and data structures are used that perform the required function, but may be less than ideal.	Program implements most required graphical elements in some fashion.	Most or all of the required graphical elements are missing or do not function correctly.
Code Mechanics	Program code is formatted in a consistent manner, variables and data structures are named in a consistent, logical manner. Code is commented adequately.	Program code is mostly consistent, but contains some occasional inconsistencies.	Program code is readable. Some functions or code blocks show consistent formatting, but that formatting does not carry through the entire program.	Program code is not consistently formatted, but is still somewhat readable.	Program code is a mess and may be more suitable as an entry to the International Obfuscated C Coding Competition.
User Interface	The program is responsive to input. All required inputs are implemented, and the user is informed, by the program, what the inputs are. The program can be terminated by the user.	The program is responsive to input. All required inputs are implemented. Some of the inputs are documented by the program.	The program is unresponsive under some circumstances. All required inputs are implemented. Some of the inputs are documented by the program.	The program is unresponsive under some circumstances. Some required inputs are either not implemented or are not implemented correctly. Some inputs are documented by the program.	Many of the required inputs are either not implemented or are not implemented correctly. The program lacks documentation for the inputs.