CG Programming III – Assignment #2 (Planar soft shadows) Due on 4/23/2007

General description of what the program needs to do.

Required graphical elements:

- Implement soft shadows using ***one*** of the three following algorithms:
  1. Heckbert and Herf's method.
  2. Gooch's method. This can be done by either projecting the caster multiple times or by creating a shadow texture and applying the shadow texture multiple times.
  3. Single pass shadow textures. If this method is used, the shadow *must* be cast on non-planar objects using projective texturing.

Optional graphical elements:

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

- Using Gooch's method to create soft shadows and casting those shadows on non-planar objects using projective texturing (combining #2 and #3 above).

Required inputs:

- Escape must terminate the program.
- If either Heckbert and Herf's method or Gooch's method are used, a key sequence must be available to increase or decrease the number of shadow samples used. If the number decreases below *three* samples, the program must switch to hard shadows.

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 in the implementation.	Program implements all required graphical elements, but the operation of some elements may not be obvious. Appropriate algorithms and data structures are used in the implementation.	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 occasion inconsistencies.	Program code is readable. Individual 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	The program is	The program is	The program is	The program is	Many of the

Criteria	***Excellent***	***Good***	***Satisfactory***	***Marginal***	***Unacceptable***
Interface	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.	responsive to input. All required inputs are implemented. Some of the inputs are documented by the program.	unresponsive under some circumstances. All required inputs are implemented. Some of the inputs are documented by the program.	unresponsive under some circumstances. Some of the required inputs are either not implemented or are not implemented correctly. Some of the inputs are documented by the program.	required inputs are either not implemented or are not implemented correctly. The program lacks documentation for the inputs.