

CG Programming II – Assignment #5 (Term Project)

This is the final assignment of the term. It will be due on the day of the March 19th, 2007, the day of the final. This assignment will be presented to the class. You will show your program running and put it through its paces. You will also explain which lighting models you implemented and defend your design choices.

You will be required to implement six materials using no less than four lighting models. At least one of the implemented lighting models must include anisotropy. These lighting models may include *any* that we have studied this term.

- Lafortune
- Cook-Torrance (using either Ward or Beckmann distributions). Implementing both distributions only counts as one lighting model.
- Ashikhmin
- fakefur
- Banks
- Phong / Blinn

In addition a single point light, for one light model reflection map lighting.

Required graphical elements:

- Two or more user selectable objects. Only one object need be displayed at a time, but the user must be able to switch between the two at run-time. At least one of the objects must have more than 8 surfaces.
- Two different light source methods:
 - Animated point light source.
 - Light source moves (e.g., orbits the object).
 - Light source has constant color.
 - Light source is represented on the screen using a point. Setting the point size > 1.0 would be helpful, but is not required.
 - Environment map lighting.
 - Environment is stationary.
 - Environment map is rendered around object as a “sky box”.
- GLSL implementations of six materials using no less than four different lighting models. It is strongly recommended to implement one material using each lighting model, then go back and implement additional materials using one or more of the implemented lighting models.
 - Example:
 1. Skin (using Lafortune)
 2. Steel (using Laforutne)
 3. Steel (using Phong)
 4. Nickel (using Lafortune)
 5. Dog hair (using fakefur)
 6. Plastic (using Cook-Torrance)

Optional graphical elements:

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

- Draw two objects side-by-side. One object should be drawn with one lighting model, and the other object should be drawn with a different lighting model.
- Implement more than 4 lighting models.
- Implement multiple lighting models on a single object. Lafortune for skin with fakefur for hair is a good combination.
- Implement reflection map lighting for an additional lighting model.
- Implement bump mapping.
- Implement tangent mapping.
 - This is really only applicable to the anisotropic lighting models.

Required inputs:

- Escape must terminate the program.
- Key sequences must be available modify the various parameters to the BRDF. *All standard parameters must be modifiable at run-time.* I strongly recommend that the inputs to modify the parameters for a particular BRDF be printed each time a material using that BRDF is selected.
 - This includes the ability to modify the tangent direction and anisotropy parameters for anisotropic lighting models.
- A key sequence must be available to switch between point lighting and reflection map lighting. If reflection map lighting is not implemented for a particular lighting model, the program should print a message to that effect.
- A key sequence must be available to select between materials (e.g., the keys 1 through 6).
- A key sequence must be available to pause the animation of the object.
- A key sequence must be available to pause the animation of the light source.

<i>Criteria</i>	<i>Excellent</i>	<i>Good</i>	<i>Satisfactory</i>	<i>Marginal</i>	<i>Unacceptable</i>
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	Program code is mostly consistent, but contains some occasion inconsistencies.	Program code is readable. Individual functions or code blocks show consistent	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

<i>Criteria</i>	<i>Excellent</i>	<i>Good</i>	<i>Satisfactory</i>	<i>Marginal</i>	<i>Unacceptable</i>
	a consistent, logical manner. Code is commented adequately.		formatting, but that formatting does not carry through the entire program.		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 of the required inputs are either not implemented or are not implemented correctly. Some of the 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.