

## CG Programming III – Assignment #1 (Render to texture) Due on 4/9/2007

This program must perform two render-to-texture tasks. First, the program must use render-to-texture techniques to simulate waves on the surface of water. The waves are to be stored in a texture has height data. Second, the program must use render-to-texture techniques to convert the height map to a normal map. This normal map will then be used to render a scene.

Required graphical elements:

- Apply a normal map, using any lighting model, to an object.
  - Use of a simple, generally flat object, such as a cube or a plane, is recommended.
- Animate the normal map.
  - Generate the normal map from a height map.
  - Update the height map each frame.
  - Rendering to the framebuffer, then copying to the textures is acceptable for this assignment.

Optional graphical elements:

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

- Use framebuffer objects as the render-to-texture technique.

Required inputs:

- Escape must terminate the program.
- A key sequence must be available to switch between normal mapped render and using the height map as a gray scale texture. *This will be a very helpful debugging aid while implementing the height map conversion step!*

<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 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.

<i>Criteria</i>	<i>Excellent</i>	<i>Good</i>	<i>Satisfactory</i>	<i>Marginal</i>	<i>Unacceptable</i>
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.