

Assignment

Enterprise Web Development

Specification

- Design a Scene in Blender, Maya or any tools that supports export using wavefront file format
- Write a C++ OpenGL application along the following lines:
 - 1. Load and render the scene
 - 2. Make the scene navigable (i.e. use keystrokes to “move around” the scene)
 - 3. Light the scene
 - 4. Add texture or bitmaps to surfaces in the scene
 - 5. Animate specific entities in the scene

1. Load and render the scene

- The wavefront file format is a standard text file, containing
 - Tables of vertices
 - Tables of normals
 - Named Groups of vertices
 - References to Material Properties in other files
- The model loader and simple abstractions developed for Assignment 1 could be a useful starting point

2. Make the scene navigable

- Will need to move from *Orthographic* to *Perspective* projection (next topic)
- Arrow keys or some combination of other keys should be employed
 - e.g. up,down,left,right,forward (space perhaps), back (tab?).
- The model should be transformed (translate/rotate) in the event handler for these keys.
- up/down/left/right will trigger a rotation of the entire scene around viewers coordinates
- Forward/back will move the scene towards or away from the viewer
- Overall effect to be encapsulated in a “Camera” abstraction

3. Light the scene

- Colour, Lighting and Materials are a fundamental part of the OpenGL programming model
- Lighting, in particular, requires the use of surface normals in order to achieve the desired calculations
- Multiple Lights may be added to the scene, of varying intensities
- We will cover these topics over the next few weeks

4. Add texture or bitmaps to surfaces in the scene

- Textures and Bitmaps can be loaded as part of the modeling package
- The appropriate data will be generated to the wavefront file, or associated materials files.
- Alternatively, you may choose to use the modeler just for vertex and normal data, and hard code your textures in your application

5. Animate specific entities in the scene

- Make some aspects of the scene (effectively selected groups) come alive.
- Examples:
 - The scene may contain some entities in rotation around some others (e.g nucleus + electron examples from lab 05/06)
 - The scene may contain some entity on a hard coded trajectory
 - The “Camera” itself may be animated, and move through the scene on some pre-programmed path
 - Aspects of the physics engine - developed in Semester 1 - can be integrated into the application to deliver more realistic animation experience

Grading Spectrum (Indicative only, to be updated...)

<i>Range</i>	<i>UML Model: 10%</i>	<i>Model: 30%</i>	<i>Rendering 30%</i>	<i>Physics: 30%</i>
Baseline	Simple class diagrams	Multiple Entities	Perspective + Multiple Orthographic Views	Determine Properties of some Physical Objects (Cube, sphere, position, orientation)
Good	+Packages	Textures & Materials	Windowed Display (shows Perspective + 3 Orthographic like Maya)	Simulations involving forces (gravity, spring)
Excellent	+Sequence	Lighting	Navigable Camera	Simulations involving forces with constrains (e.g. Newtons cradle)
Outstanding	+Activity diagrams	Aggregate Multiple Model Files	Scriptable Camera (proceed on some trajectory)	Collisions