Assignment 2
CSCI-4965: Three-Dimensional Computer Graphics

Due: Monday, October 16, 2000, 10:00am

1 Overview

This assignment is intended to familiarize you with Bezier splines and NURBS. You are to write a program to display a spline surface. You will animate and color the surface using multiple colors. You will also use basic lighting functions in OpenGL.

You can use the Sun Ultra10 machines in the OOT Lab (Amos Eaton 117) or in the Sparc Lab (Amos Eaton 217). Make sure that your submitted program can be compiled and run on these machines. For any further updates and information on the assignment, please see the assignment web page at `www.cs.rpi.edu/courses/fall00/graphics/assign2.html`.

2 The Spline Surface Task

You are to write an OpenGL program to draw a spline surface, and animate and color it. (Think of it as a flag waving in the wind, for example.) Your assignment consists of the following tasks:

1. Draw a Bezier spline surface using evaluator commands such as `glMap2f` and `glEval*`. Describe the Bezier surface in the README file (polynomial degree for each parameter, number of control points, and default grid spacing). Let the default projection mode be perspective projection.

   See the examples in Chapter 12 of the OpenGL book for ways to display the spline surface, including the use of rotations, lighting, and material properties to make the surface more visible. (Lighting and material properties are discussed in Chapter 5 of the OpenGL book.)

2. Create the same surface by specifying it as a NURBS surface using commands such as `gluNewNurbsRenderer`. State the order, knot vector, and other relevant information for each parameter in the README file. Let the default sampling tolerance be 25.

3. Animate the NURBS surface by specifying time varying control points. You could move the control points along sinusoidal paths or along specified interpolating splines. Describe your choice of path for the control points in the README file.

4. Color different regions of the NURBS surface with different colors. You can achieve this by specifying each colored region as a different NURBS surface, and ensuring that the appropriate boundary conditions between the regions are maintained. For example, you can create a flag of the Netherlands, which consists of horizontal red, white, and blue bands, in honor of Stephen Van Rensselaer's ancestors. (See `www.flags.net/NETH.htm` for a picture.)

5. To enable user interaction via the mouse and keyboard, write functions for the following keyboard and mouse events, with the corresponding desired behaviors:

   - `b/n` keys: Enable the Bezier surface or NURBS surface modes respectively.
   - `w` key: Toggles between wireframe and shaded polygon representations.
   - `s/S` keys: The NURBS sampling tolerance is decreased/increased in steps of 10.
   - `c` key: Toggles the display of the control points.
   - `i` key: The surface is returned to its initial configuration.
   - `q` key: The program quits.
   - Left mouse button click: Toggles the projection transformation mode between perspective projection and orthographic projection.
   - Right mouse button click: Starts/stops the animation of the NURBS surface.

You need to animate and color only the NURBS surface, and not the Bezier surface.

3 Handin

The code must be submitted no later than 10:00 am on October 16, 2000. **You are responsible for ensuring that your code can compile and run on the Sun Ultras in Amos Eaton 117 or 217.** You will hand in your code using the submit script (details on the assignment web page).

You must hand in your source code (source and header files) along with a Makefile to compile it. Also include a README file with the following information (in addition to information requested above): your name, instructions on how to compile the code and run it, known bugs or limitations, any extra credit enhancements, and any other relevant information.

4 Grading

Your assignment will be graded as follows (100 points total):

1. Display a Bezier surface using 2D evaluator commands. 20 points
2. Display surface as a NURBs surface. 20 points
3. Animate the NURBS surface. 20 points
4. Color regions of NURBS surface. 10 points
5. Keyboard and mouse functions (`b/n`, `w`, `s/S`, `c`, `i`, `q` keys; mouse clicks). 20 points
6. Code structure, clarity, and documentation. 10 points

Extra credit: You can earn up to 10 additional points for special features and creative enhancements to the assignment requirements.

Lateness policy: Late submissions will incur a penalty of 20% a day (24-hour period) after the submission deadline.