Prof. Dr. Hans-Peter Seidel
Dr. Michael Wand
Nils Hasler, Jens Kerber, Thomas Schultz,
Carsten Stoll, Martin Sunkel

Geometric Modeling

Assignment sheet 8 (Spline Surfaces, due June 27th 2008)

Some of the material covered in this exercise was not covered by the lecture yet. However, the corresponding slides are already online (Lecture #11).

(1) Bezier Triangles [1+5 points]

a. Add a button that adds a triangle to the new 3D viewer (download on homepage). Uniformly tessellate it using a user definable number of subdivision levels.

b. Devise a method that tessellates the triangle non-uniformly such that the number of triangles is minimized but curvature is represented faithfully. Hint: One solution would be to add all triangles to a list and split triangles with the highest errors first.

(2) Surface of Revolution [1+3+3 points]

a. Add a button that creates a number user specifiable of connected points in the x,y-plane. These points act as the "generatrix" for a surface of revolution. Allow dragging them and display the G1 continuous cubic B-Spline. Hint: Resetting the camera allows you to edit the curve in the x,y-plane.

b. Add a button to rotate the curve around the y-axis. Create quadratic rational Bezier curves that show the rotation of the points fully around the axis. You will need either three or four segments to describe a full revolution (implement just one variant). Add copies of the "generatrix" at the angles you terminate the rotation segments.

c. Triangulate the surface.

(3) Tensor Product Surfaces [1+2+4 points]

a. Add a button that creates the points for a 3x3 grid of connected Bezier patches (see slide 23 of Lecture #11). Allow dragging the points and display the grid of curves starting and ending at the control points.

b. Also display the triangulated surface.

c. Add buttons that ensure C0 and C1 continuity moving the points as little as possible.

Software Package Remarks:

The new package contains a class ***GLGeometryViewer3D*** which you should use throughout this assignment.

Its vertices (for points, lines and triangles) have to be of type ***Point3D***.

Navigation:	Left-Mouse-Button: camera rotation
	Middle-Mouse-Button: camera translation in the plane
	Right-Mouse-Button: zoom in and out (or camera translation in depth)
Reset-Button:	Sets the camera to the initial view
Select-Button:	If no control point is picked, then one can and will rotate the camera.
Light-Button:	Toggles lighting on and off. A triangle contains a variable for its diffuse color, which can be manipulated.
Fill-Mode-Button:	Allows toggling between a solid triangle and a wireframe representation.
Axes-Button:	Lets you decide if the axes are drawn or not.
Handles-Button:	Explicitly draws all control points in yellow.

Control points are all those for which the *canBeModified* flag is set to true. All other inner vertices should be marked with *canBeModified* = false. Only control points can be translated! The flag is true by default.

Remember that whenever a control point is translated, the *update()* function of your experiment will be called (which is empty by default) .