Math 155B – Topics in Computer Graphics – Spring 2017

Project #1 – Create a textured model with Rational Bezier patches

Due date: Friday, April 28, 9:00pm.

Overview: For this assignment you will program by hand an object using rational Bezier patches. Your program should support the following:

1. Design an object similar to the famous Teapot, for instance, you might design a coffee pot, or a vase, or an urn, or a lantern, or etc. There must be circular or elliptical cross-sections somewhere in your design.
2. Apply a texture map to at least part of your object. The texture map should be able to toggled off and on so we can see the object without the texture map.
3. Your program should support keyboard controls for changing the direction of view, so that the model can be viewed from any direction.
4. All surfaces should have normals, and show specular highlights so that we can tell the normals are correct.
5. Your program should allow interactively changing mesh resolutions, and interactively toggling between wireframe and solid mode. Mesh resolution can be done with commands "**U**", "**u**", "**V**", "**v**" as in the new sample code for **SimpleNurbs**, or with "**M**" and "**m**" as was done in last programming projects from last quarter.
6. Your program should support toggling between wireframe and non-wireframe mode, by a "**p**" command for instance.
7. Sample projects from an old 155B class for a similar project can be seen online at http://math.ucsd.edu/~sbuss/CourseWeb/Math155B_2004Winter/BezierProjectsView.html.

Suggestions:

1. There is a sample program **SimpleNurbs,** available from the textbook's web page. This shows how to form a Bezier surface for a spherical shape. The suggestion is to write your program by starting with this program and bring over code from your old Project 5 code from 155A to handle texture maps. (If you need a working version of the Project 5, I can provide you with source code for the demo version.)
2. An upgraded version of **RgbImage** is now available at http://www.math.ucsd.edu/~sbuss/MathCG/OpenGLsoft/. This handles more kinds of .bmp files than the old version, so please use this new code instead of the code from last quarter.
3. If you are having problems picking control points for 1D Bezier curves (say for silhouettes of surfaces of rotation), you may use an online curve designer such as the one at https://www.desmos.com/calculator/cahqdxeshd.
4. See **SimpleNurbs.cpp** for examples of how to draw Bezier patches with the OpenGL commands (with **glMap2f, glMapGrid2f, glEvalMesh2**, etc). Use the **GL_MAP2_VERTEX_3** parameter for **glMap2f** to specify control points describing the shape of a Bezier surface. Use the **GL_MAP2_TEXTURE_COORD** parameter for **glMap2f** to specify control points describing the shape of a Bezier surface. (You will probably want to use a degree 1 Bezier patch for the texture coordinates.) Use **glEnable** for both of these parameters too. The **glMapGrid2f, glEvalMesh2** commands will generate both the surface and the texture coordinates. The easiest

way, and usually the best way, to have normals specified automatically is to use **glEnable(GL_AUTO_NORMAL).**

Hand in procedure: One-one grading PLUS submit a PDF file.

(1) As usual, create a directory called **Project2** in the **Documents** folder your networked file system on the PC computers. (not the desktop). Place there your source files and your Visual C++ solution and project files. Do not modify these after turning in the programs, so as to leave the file dates unmodified. Your program should compile and run in your work directory with the version of Visual C++ installed on the computer lab computers. If you use another version of C++ at home for development, it is your responsibility to convert your project files to work on the computer lab computers.

(2) Create a one or two page PDF file (preferably one page, **at most** two pages!) describing your project. For this, include a short description of your project, and at least one image. This will be uploaded for Professor Buss, so he can see all projects from the class. By default, the PDF files will be made available for other students to see in this class and possibly future classes, so include your name, but not your PID. (Please let me know if you prefer that your report not be included.)

There will be further instructions on how to submit the PDF files.

Grading: Grading is an individual session with Srivastava or Professor Buss, and on evaluation of your PDF submission. Please do not modify your files after the due date. You should be prepared to explain how your program works, and to show examples of the relative advantages and disadvantages of the Catmull-Rom splines and the two types of Overhauser splines.