Math 155B – Topics in Computer Graphics – Spring 2017

Project #1 - Catmull-Rom, Overhauser, and Centripetal interpolating splines

Overview: For this assignment, you will write a program that accepts points via mouse clicks, and draws a spline curve that interpolates them. Your program will generate three kinds of curves: **Catmull-Rom curves** and **Overhauser curves with chord length parameterization** and with **centripetal parameterization..**

Due date: Friday, April 13, 9:00pm.

Your program should support the following:

1. Left mouse clicks place points in the window. The curve is updated on the screen interactively with mouse clicks.
2. The curve interpolates all the points, including the first and last points. This was not true of the Catmull-Rom and Overhauser splines as defined in class. You may implement this in several ways. The most likely ways are either: (a) by acting as if the first and last point are placed twice or (b) by setting the first derivative to zero at the beginning and the end of the curve.
3. The **0** command selects Catmull Rom curves (unit length parameterization),
4. The **1** command selects Overhauser curves with chord length parameterization,
5. The **2** command selects Overhauser curves with centripetal parameterization.
6. The **f** command removes the first point on the curve. The **l** command removes the last point on the curve ("f" for "first" and "l" for "last"). If more than 64 points are placed, then the first point is removed.

A skeletal version of the program is available from the textbook's web page, or directly from http://math.ucsd.edu/~sbuss/MathCG/OpenGLsoft/ConnectDots/ConnectDots.html**.** This program supports catching mouse clicks, the **f** and **l** commands, and clamping to at most 64 points. It draws only straight-line segments, not curves: your job is to add to the program the ability to draw interpolating curves.

The sample programs draw big, black points and thick, colored lines and curves.

A sample PC executable is available as **CatmullRomDraw.exe.** This program also supports a "B" option for B-spline interpolation. This is NOT part of Project 1. To avoid virus blockers, you can download the executable via the following zip file:

http://www.math.ucsd.edu/~sbuss/CourseWeb/Math155B_2017Spring/Project1/Project1_Math155B_Spring2017.zip

Helpful hints:

1. Start with the provided sample code.
2. For each segment of the interpolating spline, draw it with a one dimensional Bezier curves (with **glMap1f, glMapGrid1f, glEvalMesh1**, etc). Use the **GL_MAP1_VERTEX_3** parameter for glMapGrid1f, and set **z**-coordinates equal to zero.

3. Watch out for repeated points (obtained by clicking the same point twice in a row), so you do not get a divide-by-zero exception. Handle repeated points as you think best. Be prepared to explain how you handled this.

Hand in procedure: As usual, create a directory called **Project1** 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.

Grading: Grading is an individual session with Srivastava or Professor Buss. 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.