Tutorial 9
Computer Graphics, Spring'23

Description

This tutorial is about implementing a curve editor using OpenGL. You are provided with C++ source code which uses OpenGL to draw points on the screen by mouse click. You have to write the code to draw Bézier curve, B-spline and Beta-spline curves using these points.

Unzip the provided files, compile and run. At this point you will see a drawing screen where you can click mouse and draw points. This is a simple user interface for manipulating curves. You should be able to create and move the control points by clicking and dragging with the mouse. This functionality is provided by the *PointCanvas* class. Right-clicking brings up a menu comprising of different types of curves that you have to implement: basic Bézier curve, cubic B-spline curve and Beta-spline curve. Last 2 menu items are for clearing the screen and exiting from the program, which are already implemented. You will need to write code at the specified places in *curves.cpp* (and *curves.h* if you add any new function).

Following are the details of the tasks for the assignment:

- **Bézier curve evaluation:** Implement simple evaluation and rendering of Bézier curve. You are provided with an incomplete *curveOperations* class in *curves.cpp*. Implement *curveOperations::evaluateBezier* to return the point along the Bézier curve at the given t-value. A function *curveOperations::binomialCoefficient()* is provided to compute $\binom{m}{i} = \frac{m!}{i!(m-i)!}$.

  Next, implement *curveOperations::drawBezier*. Draw the curve by evaluating it at uniformly-spaced t-values, and then drawing lines between those points. We have provided a function *curveOperations::drawLine()* to do the line-drawing for you. It is up to you to choose how many different points to evaluate in order to produce a reasonable-looking curve ($\Delta t = 0.01$ produces pretty good result).

- **Draw cubic B-spline curve:** Implement *curveOperations::drawCubicBspline* to draw a cubic B-spline curve from the clicked control points. Follow the course notes to get the details of cubic B-spline. There are different notations that are used, feel free to use any idea. However, the notation used in slide 43, 44 are pretty straightforward:

$$P(u) = \sum_{i=nu-2}^{nu+2} P_i B(nu - i).$$

Use the range of $(nu-2)$ from $floor(nu-2)$ to $ceil(nu+2)$. This notation will help you to implement rest of the parts of the assignment without any ambiguities from the course notes. Use *curveOperations::drawLine()* to draw the curve after generating the points.

- **Draw Beta-spline curve:** Implement *curveOperations::drawBetaspline* to draw a Beta-spline curve from the clicked control points. Follow the course notes, or read shared material from the scanned book for detailed description of Beta-spline. Use skew (s) = 1.0 and tension (t) = 10.0.

You should not need to modify any files except for *curves.cpp*. However, feel free if you want to add any function to *curveOperations* class (say a separate method for blending function), and obviously you will need to modify *curves.h*.