Ex02 - Show of Your Curves

Info:
Create a simple application to compute a Bezier curve using De Casteljau algorithm. As usual implement a similar functionality as in the sample application. Use a template or implement in your own way. The implementation should be a De Casteljau algorithm by definition.
Application should be able to add points and move or delete selected once.

De Casteljau's Algorithm:

- PSEUDOCODE

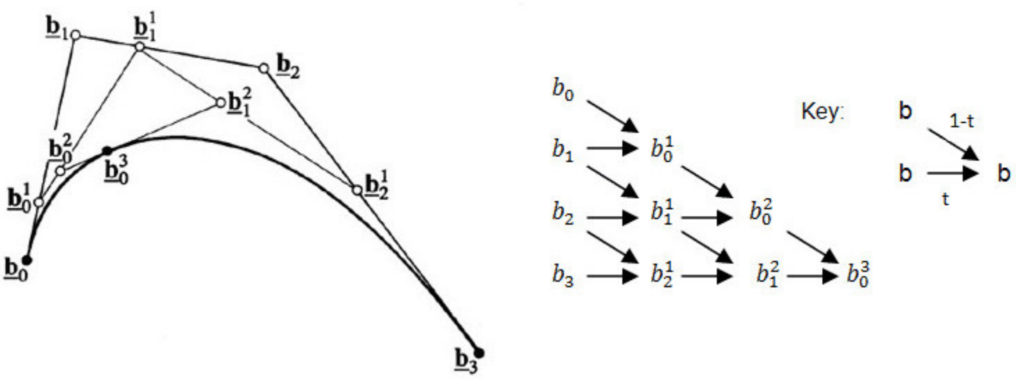

Figure 1: *Left:Representative visualization of De Casteljau algorithm. Right: Pseudocode.*

- IN
  Input are control points $b_0 ... b_n$, showed in Figure 1 and a parameter $t \in [0, 1]$. Sampling quality changes number of curve points computed by De Casteljau algorithm.

- OUT
  Output are partial points $b_0^1 ... b_1^2$ given by

$$b_0^1 = (1-t)b_0 + tb_1, \tag{1}$$

and a final curve points b_0^3 computed for parameter $t \in [0, 1]$.

- EXTREME

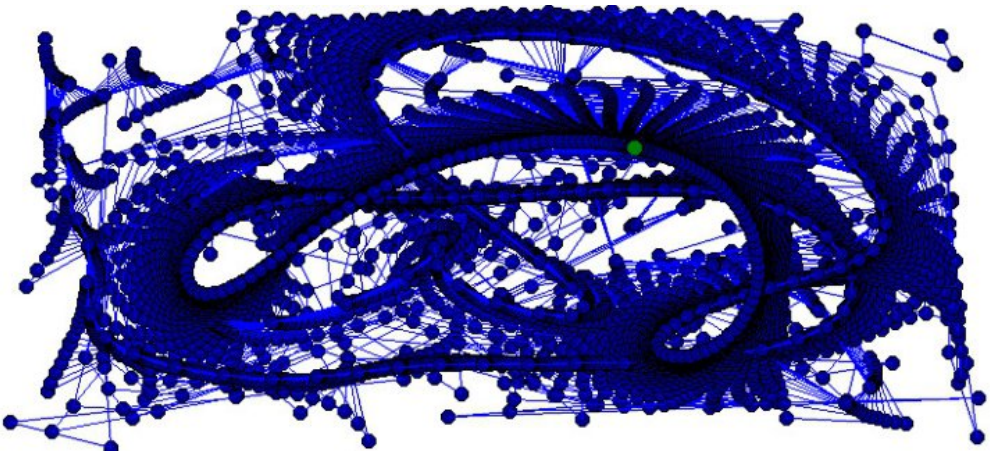

Figure 2: *Visualisation bonus example. Blue points are partial computations. Green point is the result curve.*

Bonus Points:

1. Implement representative visualisation of De Casteljau algorithm - **1 point**
2. Implement one other curve computation - Lagrange interpolation, piecewise interpolation or other - **1 point**

More information during the seminar and / or mail and consultation

Materials:
Bezier Curve - `http://en.wikipedia.org/wiki/B%C3%A9zier_curve`

De Casteljau algorithm - `http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/spline/Bezier/de-casteljau.html`

Piecewise Bezier curve - `http://www.cs.sunysb.edu/~qin/courses/geometry/5.pdf`

Lagrange interpolation - `http://www.sirver.net/blog/2011/09/06/the-bridge-between-lagrange-an`