Assignment in Computer Graphics II

– Assignment 5 –

Computer Graphics and
Multimedia Systems Group
David Bulczak, Christoph Schikora

Assignment 1 [1 Point] Evaluation of TP-Bézier-surfaces

Given the bi-quadratic Bézier-surface with control points

$$\mathbf{C}_{00} = \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix}, \quad \mathbf{C}_{10} = \begin{pmatrix} 4 \\ 2 \\ 6 \end{pmatrix}, \quad \mathbf{C}_{20} = \begin{pmatrix} 12 \\ 2 \\ 2 \end{pmatrix}, \quad \mathbf{C}_{01} = \begin{pmatrix} 0 \\ 6 \\ 6 \end{pmatrix}, \quad \mathbf{C}_{11} = \begin{pmatrix} 4 \\ 6 \\ 10 \end{pmatrix}$$

$$\mathbf{C}_{21} = \begin{pmatrix} 12 \\ 6 \\ 6 \end{pmatrix}, \quad \mathbf{C}_{02} = \begin{pmatrix} 0 \\ 14 \\ 2 \end{pmatrix}, \quad \mathbf{C}_{12} = \begin{pmatrix} 4 \\ 14 \\ 6 \end{pmatrix}, \quad \mathbf{C}_{22} = \begin{pmatrix} 12 \\ 14 \\ 2 \end{pmatrix}$$

1. Evaluate twice the surface using the two-stage de Casteljau algorithm for the parameters (0.5,0.5):
   - First in u and then in v direction
   - Then first in v and then in u direction.
2. Determine the surface normal $\hat{n}$ for the parameters (0.0) and (1.1).
3. Determine the surface normal $\hat{n}$ for the parameters (0.5,0.5).

Assignment 2 [1 Point] Program Frenet-Frame

Write a program for visualization of the Frenet frames of a 3D-Bézier-curve.

The available program has the following functionality:

1. Data read by argument (eg. data_01): degree n and the corresponding Bézier-points, number of sample points for the display.
2. Evaluating the 3D-Béziercurve for given parameters.
3. Curve and Frenet Frame visualize.

The provided program consisting of the following classes:

`P3D, V3D`: A 3D-point and 3D-vector class including various operators to manipulate points and vectors.

`ParamCurve3D`: The already defined class for 3D-curves and additional methods for the determination of the first and second derivative.

`BezierCurve3D`: The concrete instance of a 3D-parametric curve

`FrenetFrame`: Class for computing the Frenet-frame for a given `ParamCurve3D`.

`DisplayOGL`: Class to display the curve and the frame for a fixed parameter value.

The class hierarchy is complete, the following methods and operators have not yet been implemented:

1. cross product `V3D::operator^(const V3D& vec) const`.
2. Calculation of the first and second derivative in
   `BezierCurve3D::eval(double u, bool preserveDerivatives)`.
   Clue:
   The derivatives are calculated based on the calculated points in the de Casteljau algorithm.
   $$\mathbf{C}'(u) = n(\mathbf{C}_1^{n-1}(u) - \mathbf{C}_0^{n-1}(u)), \quad \mathbf{C}''(u) = n(n-1)((\mathbf{C}_2^{n-2}(u) - \mathbf{C}_1^{n-2}(u)) - (\mathbf{C}_1^{n-2}(u) - \mathbf{C}_0^{n-2}(u)))$$
3. Calculation of the vectors of the Frenet frames
   `bool FrenetFrame::computeFrame(double u, P3D& position, V3D& tangent, V3D& normal, V3D& binormal)` .

Submission: 13.11.2014, before /at the beginning of the exercise.

Submit task 1 on paper and send for task 2 an email with the modified files.
→ Email to: david.bulczak@uni-siegen.de, christoph.schikora@uni-siegen.de

The **deadline** is the same for all tasks, e.g. emails will only be accepted till Thursday 12:00 clock. If we receive your mail, we will send you as soon as possible a confirmation.