Assignment in Computer Graphics II

– Assignment 3 –

Computer Graphics and
Multimedia Systems Group
Markus Kluge, Dmitri Presnov

Assignment 1 [2 Points] Polynomial curves & Bezier curves

In this task we introduce a curve framework that supports several types of curves presented in the lecture. You will begin with an initial curve type and extend this tool during the next weeks. Download the framework `curve-framework.zip` and take an initial look on the code.

All relevant files for this and future programming tasks, related to curves, can be found in the `Curves` folder. In `Curve/Curve.hpp` you can find the abstract base class for all further curve classes. It provides three abstract member functions `eval, evalCurve, evalConstruct` which you will have to implement for all derived classes at least. Please study this class, read the comments and try to understand it.

Additionally take a look into the `Curve/PolynomialCurve.hpp` and `Curve/PolynomialCurve.cpp` files. They provide the implementation of a class for curves based on the monomial basis. Take this as an example how to access important C++ containers and how to store the corresponding result values.

To build the project in your preferred development environment use the included CMake project ("CMakeLists.txt"). CMake can be downloaded from the following website: http://www.cmake.org/. Use the instructions on the page
http://www.cmake.org/cmake/help/runningcmake.html and the tutorial page to create the project.

1. Implement Bézier-Curves in `Curve/BezierCurve.hpp` and `Curve/BezierCurve.cpp`. This curve should be evaluated by using Bernstein Basis polynomials and by using the de Casteljau algorithm.
   - `calculateBernstein`: Evaluates a Bernstein basis polynomial.
   - `evalBernstein`: Computes curve value for a given parameter by evaluating Bernstein polynomials.
   - `evalNextLevel`: For a given array of points, that represent an iteration level of the de Casteljau algorithm, it computes the next level. This function should be used to compute the affince combinations presented in the lecture.
   - `eval`: Computes curve value for given parameter by evaluating the de Casteljau algorithm.
   - `evalConstruct`: Evaluates curve by the de Casteljau algorithm and stores intermediate steps.
   - `evalCurve`: In this function the curve has to be evaluated for the whole interval $[0, 1]$.

Further explanations can be found in the comments of the code.

Assignment 2 [2 Points] A-Frame Construction

Given a cubic Bezier curve C with control points

$$\mathbf{C}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad , \quad \mathbf{C}_1 = \begin{pmatrix} 3 \\ 4 \end{pmatrix} \quad , \quad \mathbf{C}_2 = \begin{pmatrix} 7 \\ 5 \end{pmatrix} \quad , \quad \mathbf{C}_3 = \begin{pmatrix} 8 \\ 2 \end{pmatrix}.$$

Continue with another Bezier curve D from control point $\mathbf{C}_3$ to control point $\mathbf{D}_3 = \begin{pmatrix} 12 \\ 1 \end{pmatrix}$ so that the resulting curve is C^2 continuous.

1. Determine control points $\mathbf{D}_0$, $\mathbf{D}_1$, $\mathbf{D}_2$ for the new curve.
2. Specify a piecewise defined formula for the new curve $\mathbf{G}(v)$ with $v \in [0,1]$ that passes through $\mathbf{C}_0, \mathbf{C}_3$ and $\mathbf{D}_3$. Thus connect curves C and D in $v = \frac{1}{2}$.
3. Prove by calculation that the transition between C and D is C^2 continuous.

Total points after sheet 3: 15 of 70.

Hand in: Until 3.05.2018 12:00 o'clock in mailbox of our chair (next to room 7115) and the programming assignment via e-mail (`jan.mussmann@student.uni-siegen.de`**).**