Prof. Dr. Konrad Polthier
Konstantin Poelke
Thomas Gust

Scientific Visualization
Winter Semester 2015/16
Freie Universität Berlin

Exercise Sheet 1

Out: 19.10.2015

Due: 28.10.2015

You should work in groups of two. Don't forget to put your names and student ID numbers on the solution you hand in!

Exercise 1.1 (Bézier and Splines, 3+2+2+1+1+1=10 Points)**.** Answer the following questions!

1. Prove that the Bernstein polynomials $B_i^n(t)$, $i = 0, \ldots, n$ form a basis for the $\mathbb{R}$-vector space of polynomials $\mathbb{P}^n$ of degree at most n on the unit interval.

   Hint: By dimension counting it is enough to show that the $(n+1)$ Bernstein polynomials $B_i^n(t)$ are linearly independent, i.e.

   $$\sum_{i=0}^{n} \lambda_i B_i^n(t) = 0 \Rightarrow \lambda_j = 0 \text{ for all } j = 0, \ldots, n.$$

   Start with λ_n, consider $B_i^n(1)$ and use the multiplicity of $t = 1$ as a zero of the $B_i^n(t)$'s. Then examine derivatives...

2. Let $b(t)$ be the Bézier curve of degree n associated to the $(n+1)$ control points P_i, $i = 0, \ldots, n$. Show that for the second derivative the following formulas hold:

   $$b''(0) = n(n-1)(P_2 - 2P_1 + P_0) \text{ and } b''(1) = n(n-1)(P_n - 2P_{n-1} + P_{n-2})$$

3. Can you give a geometrical interpretation for the constraint that a cubic Bézier spline should be $\mathcal{C}^2$ at a junction in terms of the involved control points?

4. Given four control points $P_0, \ldots, P_3$, write the cubic Bézier curve $b(t)$ associated to these points in terms of the monomial basis $\{1, t, t^2, t^3\}$, i.e. as an expression

   $$b(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3.$$

5. For the representation just derived write $b(t)$ as a formal matrix product

   $$b(t) = (t^3 \; t^2 \; t \; 1) \cdot M \cdot (P_0 \; P_1 \; P_2 \; P_3)^T$$

   and find the entries of the matrix $M \in \mathbb{R}^{4\times 4}$.

6. What possible advantages or disadvantages of this representation compared to the usual Bézier representation in terms of Bernstein coefficients come to your mind?

Exercise 1.2 (Installing Eclipse and JavaView, 0 Points)**.** If you haven't already done so, now it's time to install the Eclipse IDE and JavaView, create a new Java project for development and import the `vgp.tutor` sources into your project. Follow the guidelines for the *Installation for Developers with JV jars* in the JavaView wiki [1] and in emergency cases ask your tutor. Create a source subdirectory directly under your JavaView project which you name "myprojects" and which will contain your own projects later on. This can be done by right-clicking the JavaView project and selecting "New→Source Folder"

Exercise 1.3 (Coloring Elements, 4 Points)**.** Understand the example project under `vgp.tutor.firstProject`. The GUI elements for the info panel are all defined in the class `MyProject_IP`. You can run it by right-clicking the `MyApplet` class and choosing "Run as→Java Application". Note that the runnable class is usually prefixed "Pa" in other JavaView projects, e.g. "PaLinear" in `vgp.tutor.linear`. When you feel comfortable with the code structure, do the following:

1. Create a new package under your "myprojects" folder (right-click it and select "New→Package") and name it "colorize". Copy the content from the example project `vgp.tutor.firstProject` to this package. You should then have 4 files in your colorize-package.
2. Now add a button to the info panel with name "Colorize" and implement a method to the project class such that every element of the animated surface with odd index becomes blue and every element with even index becomes white when you click on the button.

When your program is running correctly, export it to a runnable jar as described on the web page with the usual file naming convention `name1_name2_colorize.jar`. Make sure that the jar works as intended by running it, either via double-click or, for the terminal users, via `C:\..\ my_directory> java -jar name1_name2_colorize.jar`.

Hint 1: Find an overview of relevant methods [2] as well as instructions for exporting to a JAR [3] in the wiki.

Hint 2: You may want to disable backside colors. In JavaView go to "Method→Color" and disable "Show Backside Colors".

[1] `http://www.mi.fu-berlin.de/w/AGGeom/JavaView`
[2] `http://www.mi.fu-berlin.de/w/AGGeom/JVEssentialClasses`
[3] `http://www.mi.fu-berlin.de/w/AGGeom/JVRunnableJar`

Exercise 1.4 (Practical Bézier, 1 Bonus Point)**.** Learn and understand how to use the Bézier curve modelling tool in modern vector graphics software such as *Inkscape*, *Adobe Illustrator* etc. Draw a simple car sketch like the one shown below in Inkscape using Bézier splines only. Make sure that you draw a 21^{st} century smooth car with G^1 transitions at all junctions. Save your art work as an *.svg file under `name1_name2_beziercar.svg`.

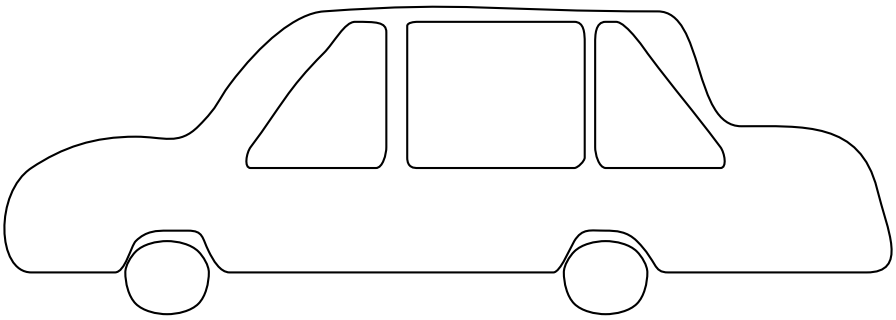

Figure 1: A modern car.