Programming assignment no. 1

(15pts)

Topic: Tensor-product Bézier patch

Deadline: 3 March 2020 – 22 March 2020

Goal: The goal of the first assignment is to:

1. draw four types of surfaces using the tensor-product Bézier patch of the bidegree (m, n),
2. draw isoparametric curves of the patch for given parameters s and t.

Instructions:

5pts Implement procedures which draw the listed types of surfaces in the reserved space in the given template:

Surface interpolating the given set of points – let $\mathcal{I}$ be the set containing $(m+1)(n+1)$ points, which are distributed along a planar grid. Then the tensor-product Bézier patch of the bidegree (m, n) is constructed, so the set $\mathcal{I}$ is **interpolated**.

After modifying the bidegree or poisition of an interpolated point, the new patch is generated automatically, so it interpolates $\mathcal{I}$.

Surface approximating the spherical triangle – consider a sphere with center $(-1, -1, -1)^\top$ and radius 2. The triangle is obtained by restriction to the octant given by equations $x > -1$, $y > -1$ and $z > -1$. The task is to construct the tensor-product Bézier patch of the fixed bidegree (m_s, n_s), so the spherical triangle is approximated.

Surface approximating the cylindrical sector – in this case, tensor-product Bézier patch of the fixed bidegree (m_v, n_v) is constructed, so it approximates the of a cylinder given by the circle lying in the plane $z = 0$, having center $(-1, -1, -1)^\top$ and radius 2, restricted to the quadrant given by $x > -1$ and $y > -1$. The height of the cylinder is 2.

Surface approximating the conical sector – the sector is constructed for the cone with the same base as in the previous case. The task is to construct the tensor-product Bézier patch of the fixed bidegree (m_s, n_s), so the conic sector is approximated.

3pts In all cases, the tensor-product Bézier patch is rendered as a quadrilateral mesh having $(k+1)\times(k+1)$ points, while each point is computed using **de Casteljau algorithm**. For each point of the mesh, the **unit normal vector** needs to be computed as well. Also, the **control net** of the patch needs to be drawn, i.e. the position of the control vertices needs to be computed.

2pts Also it is possible to depict the **isoparametric curves** of the patch for the given parameters s and t. Each of the curves is represented as a polyline having $k+1$ points. Again, apart from the curves, also their control polygons need to be drawn, too.

After modification of any parameter in the user interface, the surface is redrawn automatically.

marcel.makovnik@fmph.uniba.sk

Notes on the code: comments `TODO` are included in the following lines:

```
MainWindow.xaml.cs - 101, 111, 121, see also the line 76.
Patch.cs - 205, 216, 227, 238, 247, 258.
```

Output:

5pts Apart from the code implementing the patches (this part needs to be easily identifiable and created by the author, i.e. created explicitly for the purposes of this assignment), there is a discussion regarding the submitted solution. Details of the discussion may be found at `https://mkvnk.sk/GM1/#grading`.

Usage or integration of solely external libraries is strictly forbidden!

Apart from these, the code needs to be **richly** commented and **neatly** formatted. Poor commenting and ugly formatting might be penalized by loss up to *3 pts*.

Sample application and template is available on the webpage, together with this .pdf file.

The sample application lacks the control nets except for the case of the interpolating surface. However, your solution is supposed to render control nets for all four types of surfaces.