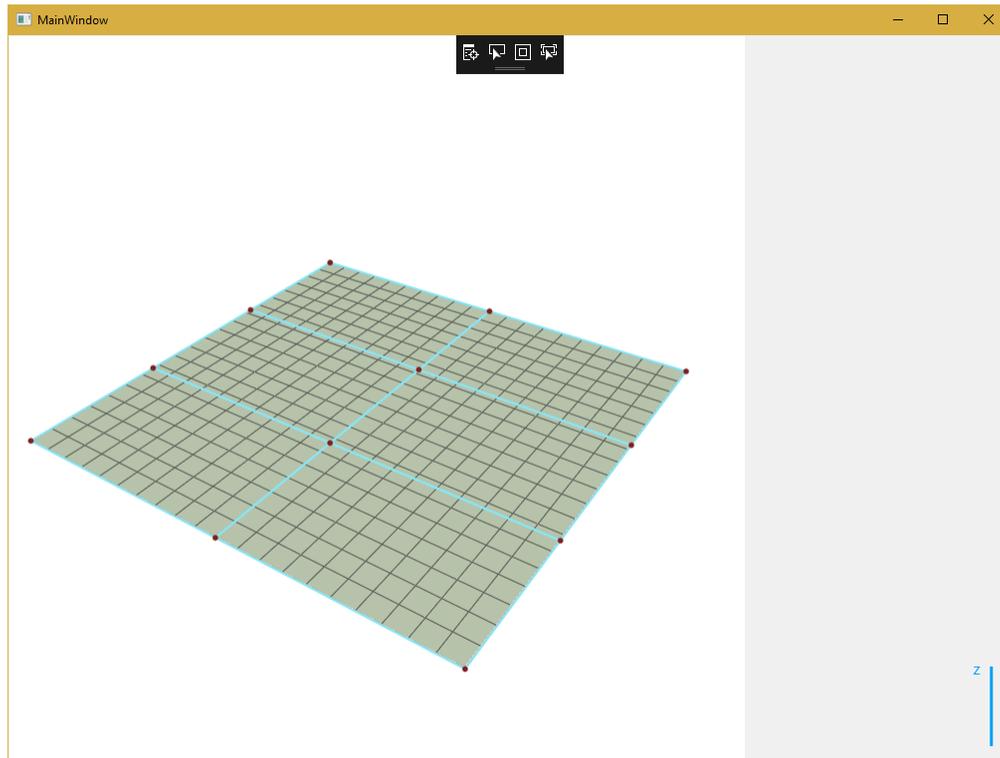


Using the interface



Obr. 1: left – the drawn patch, right – the user interface, bottom right – the edit mode.

Using the camera

- **rotation** – by dragging the RMB
- **scaling** – using the mousewheel

Moving the control vertices - by dragging the LMB.

- you can use the three edit modes:
 1. by pushing the "**Z**" key, you move the control vertex along the z-axis (the x- and y-axes are locked in this case, so the vertex does not "runaway due to the perspective).
 2. by pushing the "**X**" key or "**Y**" key, the view changes in such a way, that the control vertices can be moved in the x and y direction.
- you can use only one mode at time, however you can switch between the modes using the keys mentioned above,
- you can switch off the current mode by pushing the same key,
- active mode is depicted in the bottom right corner of the window,
- if none of the modes is active, the editing of the control vertices is disabled.

The additional user controls (buttons, text boxes etc.) are inserted from the toolbox menu in the file `MainWindow.xaml`.

Implementation and code

Template can be compiled and run in the Visual Studio 2015 and 2017 (in the older versions there might compatibility issues with the version of OpenTK) without installing the OpenTK libraries. **If you have issues running the template, please, contact me!**

When debugging, use the Debug mode. However, when the final version of the application is created, use the Release mode. This increases the performance of the application. The .exe files can be found in bin folder, together with all the necessary libraries.

You edit mainly the procedures in the file Patch.cs. Some of them are prescribed, however you may need to add your own procedures. In the file MainWindow.xaml.cs you do not have to edit procedures starting with the prefix GLcontrol_ (drawing with OpenTK) – these are placed after the comment line DRAWING PROCEDURES.

All procedures which need to be implemented are introduced by the massive comment !!! TODO !!!.
All the remaining technical notes are written directly in the code.

Do not forget to write floats in the correct notation, i.e. 1.0f.

If you need to work with matrices of arbitrary dimension, the library ALGLIB is imported in the project. For example if you want to compute the inverse matrix to the matrix M, you may use the following piece of code:

```
double[,] M = { {1, 2, 3},
                {4, 5, 6},
                {7, 8, 9} };
int Info = 0;
alglib.matinvreport Report;
alglib.rmatrixinverse(ref M, out Info, out Report);
```

After this procedure, the inverse of M is imported to the matrix M (i.e. M is replaced by its inverse).
For complete documentation, follow the pages of ALGLIB – <http://www.alglib.net/>

Useful links

Official site of OpenTK – <https://github.com/opentk>

Everything about C# and Visual studio – <https://msdn.microsoft.com>

When you are hopeless ☺ – <http://stackoverflow.com/>

If there are any questions, please contact me at marcel.makovnik@fmph.uniba.sk.