INF3320 - OBLIGATORY EXERCISE 4

VISUALIZATION OF TRIANGLE MESHES & SUBDIVISION

CHRISTOPHER DYKEN, BART SIWEK, REVISED BY ANDRÉ BRODTKORB

Revision: November 5, 2014

In this exercise we will extend the viewer made in previous exercises. Instead of rendering a sphere or a cube, we will render an arbitrary triangle mesh. Furthermore we will allow the user to refine the mesh using wither Loop or $\sqrt{3}$ subdivision. That is, we will implement and use one of

```
TriMesh* subdivideLoop();
TriMesh* subdivideSqrt3();
```

functions of `TriMesh` class. The functions return the `TriMesh` itself refined one step using either Loop or $\sqrt{3}$-subdivision rules respectively.

THE APPLICATION

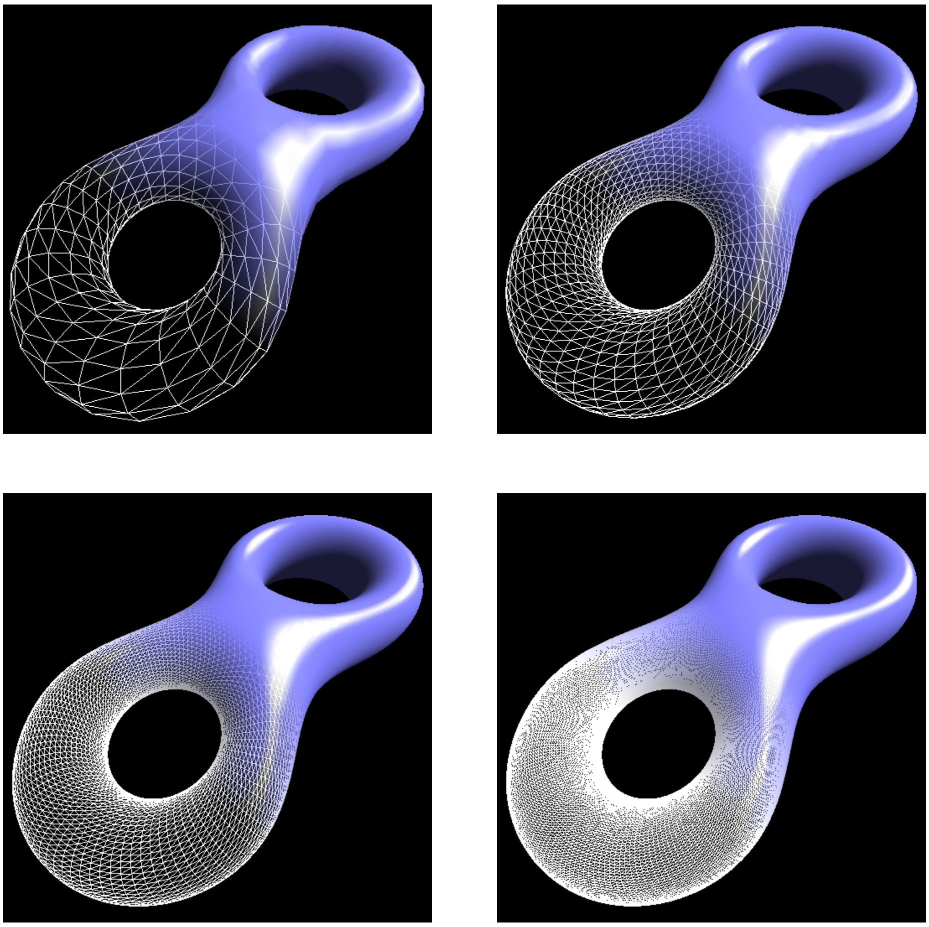
Extend your program from previous compulsory exercises to include functionality for reading and displaying `*.msh`-files. The mesh should be repositioned and scaled such that it is positioned in the center of the screen. Furthermore assume that the initial refinement level is 0 (i.e. no refinement at all). Each time the '+'-key is pressed, the mesh is refined one step, and the '−'-key reduces the subdivision level by one. Thus, it is a good idea to store the sequence of refinements of the mesh.

Details on the refinement schemes can be found in both the lecture notes and in "Real-time rendering". Start by programming function that builds the connectivity of the mesh. Next using that add computation of normals. Then start working on subdivision scheme. Make the splitting work first, retain the position of the old vertices and let the new vertices be simple averages. When the splitting scheme works, add the full calculations.

Finally, the application should handle *both open and closed meshes*.

DETAILS AND HINTS

The files `TriMesh.hpp` and `TriMesh.cpp` contains the code for a half-edge based datastructure for triangle meshes. It also contains code for reading `*.msh`-files. You will need to suplement the `TriMesh.cpp` with the code that builds mesh connectivity. Then in the file `TriMeshGL.cpp` you will need to add the code responsible for computing the normals. A suitable normal vector for each vertex is the average of the triangle normal vectors of triangles adjacent to the vertex. A nice extension is to include functionality handling sharp edges. From this point on the meshes should be display on the screen with full shading.



Final step is to choose one of the subdivion schemes (and make that choice visible in the body of the `subdivide` function in `TriMesh.cpp`), then implement the scheme in the `TriMeshLoop.cpp` or `TriMeshSqrt3.cpp` denending on your choice.

HANDING IN THE EXERCISE

The assignment is individual and everyone shall create their own program. If you choose to use code or derivations of code that is not your own, its source and author shall be explicitly cited. We will normally accept some degree of unoriginal work, but we may require that parts of the unoriginal code must be rewritten.

We reserve the right to do an oral examination of the student for each submitted exercise. The failing or passing of this examination will decide whether you pass this exercise, and this decision is final.

The program should preferably be a C++ program, and the visualisation shall be done using OpenGL. The choice of windowing toolkit is the student's. Document your efforts by commenting the code.

All source code as well as a Makefile shall be included. The Makefile should have at least a target to build the program as well as a clean-target that deletes all object-files as well as the

executable. Assume that your user name is `foo` (exchange “foo” with your user name). Put all files in a directory called `foo-4`.
Handing in the assignment is performed by creating a private repository at either GitHub (sign up as student for five private repositories on `https://github.com/edu`) or Bitbucket, and then sending an email to `mariuek@ifi.uio.no` with the clone URL and a short description detailing the files involved in your hand-in. Make sure the program compiles and runs cleanly on the computers at IFI, and that it fulfils the requirements outlined in the evaluation guidelines.

Good luck!