INF2110 - Introduction à l'Informatique Graphique **M1 Informatique**
TP6 **Fall 2018**
Prof. Caroline Larboulette

Polygonal Surfaces

What you need to turn in at the end of the lab session
At the end of the lab session, you should upload a .zip file containing the source code .cpp (+ eventually other classes you have used) of each exercice as well as a .pdf shortly describing what you have done for each question with screenshots to illustrate the results you have obtained. Remember the assignment may be graded.

The aim of this lab is to write an .obj parser, store the model in memory and display the model in point, wireframe, flat shaded and smooth shaded (switching between modes using keyboard).
First, create an object of your choice in Autodesk Maya, with material and texture, OR, download a free Maya or .obj model (if possible containing a texture) from the Internet. Program using the *libqglviewer* library.

1. .obj format specification
https://en.wikipedia.org/wiki/Wavefront_.obj_file

```
# List of geometric vertices, with (x,y,z[,w]) coordinates, w is optional
and defaults to 1.0.
  v 0.123 0.234 0.345 1.0
  v ...
  …
# Polygonal face element (see wikipedia page for details)
  f 1 2 3
  f 3/1 4/2 5/3
  f 6/4/1 3/5/3 7/6/5
  f 7//1 8//2 9//3
  f ...
```

2. Loading geometry
First, write a parser to read the vertices positions, store them, and display them using `glBegin(GL_POINTS)`. Then, read polygonal face elements and draw the faces in wireframe (only the edges) with `glBegin(GL_TRIANGLES)`.
`Use glPolygonMode(GL_FRONT_AND_BACK, GL_LINE)` or `glPolygonMode(GL_FRONT, GL_FILL)` to switch from wireframe to shaded.

3. Flat shading
Compute the normal per triangle and use `glNormal*()` to display flat shaded geometry.

```
void glNormal3{bsidf} (TYPE nx, TYPE ny, TYPE nz);
void glNormal3{bsidf}v (const TYPE *v);
```
Sets the current normal vector as specified by the arguments.

4. Smooth shading
Compute the normal per vertex and use `glNormal*()` to display smooth shaded geometry.

To compare, read the normals from the .obj file and use them for display.

```
# List of vertex normals in (x,y,z) form; normals might not be unit
vectors.
  vn 0.707 0.000 0.707
  vn ...
  ...
```