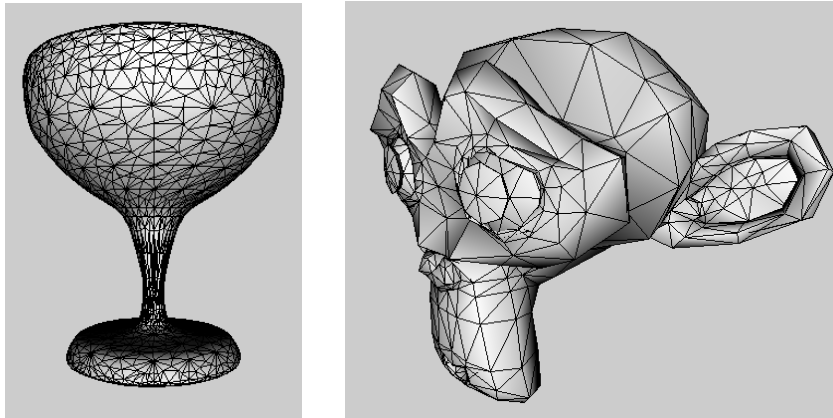

Wire (Wire.*)

Usee GLarenaPL de la vostra instal·lació local del visualitzador

Escriu un **render plugin** que, utilitzant dues passades de rendering, permeti visualitzar el model amb els triangles omplerts de color, juntament amb el seu contorn de color negre:



El plugin caldrà que implementi les funcions

```
void onPluginLoad();
bool paintGL();
```

En el primer pas de renderització (**6 punts**), caldrà pintar l'escena amb VS+FS que apliquin il·luminació bàsica: el color final del fragment tindrà per components RGB la component Z de la normal en *eye space*. El codi dels shaders ha d'estar al mateix fitxer cpp (com a l'exemple effectCRT).

En el segon pas de renderització (**3 punts**), caldrà tornar a pintar l'escena, amb els mateixos shaders, però enviant-li un uniform addicional per tal que el color del fragment sigui negre. Per tal de dibuixar només el contorn dels triangles, haureu d'usar **glPolygonMode** (més avall en teniu la descripció).

Feu servir **glPolygonOffset** (**1 punt**) per tal de donar prioritat de visibilitat als contorns negres dels triangles, per evitar que quedin parcialment ocultats pels fragments del primer pas.

Identificadors obligatoris:

Wire.cpp, Wire.h, Wire.pro (segur que has escrit **Wire** correctament?)

glPolygonMode — select a polygon rasterization mode

```
void glPolygonMode(GLenum face, GLenum mode);
```

face - Specifies the polygons that mode applies to. Must be `GL_FRONT_AND_BACK` for front- and back-facing polygons.

mode - Specifies how polygons will be rasterized. Accepted values are `GL_LINE`, and `GL_FILL`. `GL_LINE` - Boundary edges of the polygon are drawn as line segments. `GL_FILL` - The interior of the polygon is filled.

glPolygonOffset — set the scale and units used to calculate depth values

```
void glPolygonOffset(GLfloat factor, GLfloat units);
```

factor - Specifies a scale factor that is used to create a variable depth offset for each polygon. The initial value is 0.

Units - Is multiplied by an implementation-specific value to create a constant depth offset. The initial value is 0.

When `GL_POLYGON_OFFSET_FILL` or `GL_POLYGON_OFFSET_LINE` is enabled, each fragment's depth value will be offset after it is interpolated from the depth values of the appropriate vertices. The value of the offset is $\text{factor} \times \text{DZ} + r \times \text{units}$, where DZ is a measurement of the change in depth relative to the screen area of the polygon, and r is the smallest value that is guaranteed to produce a resolvable offset for a given implementation. The offset is added before the depth test is performed and before the value is written into the depth buffer.