GUI Spring 2020

Assignment – Interface-2 (I2)

Due on June 18, 2019.

Submission Instructions:

1. Please carefully follow I1 instructions.
2. This assignment has up to 5 extra credit points for going beyond the instructions set below.

Assignment Instructions:

This assignment is an extension of assignment I2, it includes the following tasks:

1) Add a drawing (graphics) window along with a viewport to the interface constructed in assignment I1. Use parallel projection and place the camera so that it "sees" the scene.
    a. Draw at least 8 different objects, including points, lines, triangles, quads, and other polygons inside the window. You can use a modified version of the program snippet for drawing an N-gon attached below.
    b. Draw a Circle in the window. You can use the plotting example presented in class or an n-gon with n > 29.
    c. Draw a cube in the window. Note that since you use parallel projection, the cube might appear as a square, so make sure that the cube is positioned in a way that three of its faces are visible.
    d. Each object (from (a)) and each face of the cube (b) should have a different color. Each point should be 2x2 pixels and line width should be 4 pixels.

In order to implement this assignment, you may want to look at chapter 8, in the book, the examples on OpenGL in the QT documentation and the QT demo launcher.

/********Author: Daniel Palmer **********/**

//Draw triangle	//Draw square	//Draw hexagon	//Draw circle
glColor3f(1, 1, 0); ngon(3);	glColor3f(1, 0, 1); ngon(4);	glColor3f(1, .5, .5); ngon(6);	glColor3f(0, 0, 1); ngon(100);

```
//Function to draw a N-sided object that is upright (defined to have the bottom edge horizontal)
//at the center of the current model identity, with diameter = 1

void GLobj::ngon(int n){
        float degree, vertx, verty, rotate, degToRad;
        rotate = 360.0/n;//Vertex rotation increment
        degree = rotate/2 + 180;//Current degree of vertex (starts rotated to make object upright)
        degToRad = 180/3.14159; //Conversion factor from degrees to radians

  glBegin(GL_POLYGON);
   for(int i = 0; i < n; i++, degree += rotate) {
                vertx = 0.5 * sin(degree/degToRad);       //next vertex's x
                coordinate  verty = 0.5 * sin((90 - degree)/degToRad); //next
                vertex's y coordinate  glVertex3f(vertx, verty, 0); }
   glEnd();}
```