



الأكاديمية العربية للعلوم والتكنولوجيا والنقل البحري

Arab Academy for Science, Technology & Maritime Transport

COLLEGE OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER ENGINEERING

Lecturer: Prof. Dr. Mohamed Taher El-Sonni
Teaching Assistant(s): Eng. Mohamed A. Aslan

Lab #5
OpenGL 3D

C/C++ Template

Listing 1: template.c

```
1 // #include <windows.h> // Only for Windows
2 #include <GL/gl.h>
3 #include <GL/glu.h>
4 #include <GL/glut.h>
5 #include <stdio.h>
6 #include <math.h>
7
8 void DrawCube ()
9 {
10     glPushMatrix ();
11     glBegin (GL_QUADS);
12     /* 1st face */
13     glColor3f (0, 0, 0); glVertex3f (-1, -1, -1);
14     glColor3f (0, 0, 1); glVertex3f (-1, -1, 1);
15     glColor3f (0, 1, 1); glVertex3f (-1, 1, 1);
16     glColor3f (0, 1, 0); glVertex3f (-1, 1, -1);
17
18     /* 2nd face */
19     glColor3f (1, 0, 0); glVertex3f ( 1, -1, -1);
20     glColor3f (1, 0, 1); glVertex3f ( 1, -1, 1);
21     glColor3f (1, 1, 1); glVertex3f ( 1, 1, 1);
22     glColor3f (1, 1, 0); glVertex3f ( 1, 1, -1);
23
24     /* 3rd face */
25     glColor3f (0, 0, 0); glVertex3f (-1, -1, -1);
26     glColor3f (0, 0, 1); glVertex3f (-1, -1, 1);
27     glColor3f (1, 0, 1); glVertex3f ( 1, -1, 1);
28     glColor3f (1, 0, 0); glVertex3f ( 1, -1, -1);
29
30     /* 4th face */
31     glColor3f (0, 1, 0); glVertex3f (-1, 1, -1);
32     glColor3f (0, 1, 1); glVertex3f (-1, 1, 1);
33     glColor3f (1, 1, 1); glVertex3f ( 1, 1, 1);
34     glColor3f (1, 1, 0); glVertex3f ( 1, 1, -1);
```

```

35
36          /* 5th face */
37          glColor3f(0, 0, 0); glVertex3f(-1, -1, -1);
38          glColor3f(0, 1, 0); glVertex3f(-1, 1, -1);
39          glColor3f(1, 1, 0); glVertex3f(1, 1, -1);
40          glColor3f(1, 0, 0); glVertex3f(1, -1, -1);
41
42          /* 6th face*/
43          glColor3f(0, 0, 1); glVertex3f(-1, -1, 1);
44          glColor3f(0, 1, 1); glVertex3f(-1, 1, 1);
45          glColor3f(1, 1, 1); glVertex3f(1, 1, 1);
46          glColor3f(1, 0, 1); glVertex3f(1, -1, 1);
47
48          glEnd();
49          glPopMatrix();
50     }
51     void myDisplay()
52     {
53         glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
54         glMatrixMode(GL_MODELVIEW);
55         glLoadIdentity();
56
57         glTranslatef(0, 0, -10);
58         glRotatef(30, 1, 0, 0);
59         glRotatef(10, 0, 1, 0);
60
61         DrawCube();
62
63         glutSwapBuffers();
64     }
65
66     void keyboard(unsigned char ch, int x, int y)
67     {
68         if(ch == 'q')
69             exit(0);
70         glutPostRedisplay();
71     }
72
73     void reshape(int width, int height)
74     {
75         glMatrixMode(GL_PROJECTION);
76         glLoadIdentity();
77         gluPerspective(20, width / (float) height, 5, 15);
78         glViewport(0, 0, width, height);
79         glMatrixMode(GL_MODELVIEW);
80         glutPostRedisplay();
81     }
82
83
84     int main(int argc, char **argv)
85     {
86         glutInit(&argc, argv);
87         glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
88         glutInitWindowSize(600, 600);
89         glutInitWindowPosition(50, 50);
90         glutCreateWindow("OpenGL 3D Cube");
91         glClearColor(1.0, 1.0, 1.0, 0.0);
92         glEnable(GL_DEPTH_TEST); // ←
93         glutDisplayFunc(myDisplay); // ←
94         glutKeyboardFunc(keyboard);
95         glutReshapeFunc(reshape);
96         glutMainLoop();
97         return 0;
98     }

```

Compile:

- `cc template.c -o template -lglut -lGLU`

Java

Listing 2: Template.java

```
1 import javax.swing.*;
2 import javax.media.opengl.*;
3 import com.sun.opengl.util.Animator;
4 import java.awt.event.*;
5 import javax.media.opengl.glu.*;           // ←
6
7 public class Template extends JFrame implements GLEventListener, KeyListener{
8     private GLU glu = new GLU();           // ←
9     public static void main(String[] args){
10         Template window = new Template();
11         window.setVisible(true);
12     }
13     public Template(){
14         setSize(600, 600);
15         setTitle("OpenGL 3D Cube");
16         setDefaultCloseOperation(EXIT_ON_CLOSE);
17
18         GLCapabilities capabilities = new GLCapabilities();
19         capabilities.setDoubleBuffered(true);
20         GLCanvas canvas = new GLCanvas(capabilities);
21         canvas.addGLEventListener(this);
22         canvas.addKeyListener(this);
23         getContentPane().add(canvas);
24         Animator animator = new Animator(canvas);
25         animator.start();
26     }
27     // Called by the drawable immediately after the OpenGL context is initialized
28     public void init(GLAutoDrawable drawable){
29         GL gl = drawable.getGL();
30         gl.glClearColor(1.0f, 1.0f, 1.0f, 0.0f);
31         gl.glViewport(0, 0, drawable.getWidth(), drawable.getHeight());
32         gl.glMatrixMode(gl.GL_PROJECTION);
33         gl.glLoadIdentity();
34         glu.gluPerspective(20, drawable.getWidth() / (float) drawable.getHeight(), 5, 15);
35         gl.glEnable(GL.GL_DEPTH_TEST);           // ←
36         gl.glMatrixMode(gl.GL_MODELVIEW);
37         gl.glLoadIdentity();
38     }
39     // Called by the drawable to initiate OpenGL rendering by the client
40     public void display(GLAutoDrawable drawable){
41         GL gl = drawable.getGL();
42         gl.glClear(GL.GL_COLOR_BUFFER_BIT | GL.GL_DEPTH_BUFFER_BIT);
43         gl.glMatrixMode(gl.GL_MODELVIEW);
44         gl.glLoadIdentity();
45
46         // Draw a 3D cube
47         gl.glTranslatef(0.0f, 0.0f, -10.0f);
48         gl.glRotatef(30.0f, 1.0f, 0.0f, 0.0f);
49         gl.glRotatef(10.0f, 0.0f, 1.0f, 0.0f);
50         drawCube(gl);
51
52         drawable.swapBuffers();
53     }
54     // Called by the drawable during the first repaint after the component has been resized
55     public void reshape(GLAutoDrawable drawable, int x, int y, int w, int h){
56         GL gl = drawable.getGL();
57         gl.glViewport(0, 0, w, h);
58         gl.glMatrixMode(gl.GL_PROJECTION);
59         gl.glLoadIdentity();
60         glu.gluPerspective(20, w / (float) h, 5, 15);
61         gl.glMatrixMode(gl.GL_MODELVIEW);
62         display(drawable);
63     }
64 }
```

```

64 // Called by the drawable when the display mode or the display device associated with ←
    the GLAutoDrawable has changed
65 public void displayChanged(GLAutoDrawable drawable, boolean modeChanged, boolean ←
    deviceChanged){
66 }
67 public void keyPressed(KeyEvent e){
68 }
69 public void keyReleased(KeyEvent e){
70 }
71 public void keyTyped(KeyEvent e){
72     char key = e.getKeyChar();
73     if(key == 'q')
74         System.exit(0);
75 }
76 // Called by display() to draw a 3D cube
77 void drawCube(GL gl)
78 {
79     gl.glPushMatrix();
80     gl.glBegin(gl.GL_QUADS);
81     /* 1st face */
82     gl.glColor3f(0.0f, 0.0f, 0.0f); gl.glVertex3f(-1.0f, -1.0f, -1.0f);
83     gl.glColor3f(0.0f, 0.0f, 1.0f); gl.glVertex3f(-1.0f, -1.0f, 1.0f);
84     gl.glColor3f(0.0f, 1.0f, 1.0f); gl.glVertex3f(-1.0f, 1.0f, 1.0f);
85     gl.glColor3f(0.0f, 1.0f, 0.0f); gl.glVertex3f(-1.0f, 1.0f, -1.0f);
86
87     /* 2nd face */
88     gl.glColor3f(1.0f, 0.0f, 0.0f); gl.glVertex3f(1.0f, -1.0f, -1.0f);
89     gl.glColor3f(1.0f, 0.0f, 1.0f); gl.glVertex3f(1.0f, -1.0f, 1.0f);
90     gl.glColor3f(1.0f, 1.0f, 1.0f); gl.glVertex3f(1.0f, 1.0f, 1.0f);
91     gl.glColor3f(1.0f, 1.0f, 0.0f); gl.glVertex3f(1.0f, 1.0f, -1.0f);
92
93     /* 3rd face */
94     gl.glColor3f(0.0f, 0.0f, 0.0f); gl.glVertex3f(-1.0f, -1.0f, -1.0f);
95     gl.glColor3f(0.0f, 0.0f, 1.0f); gl.glVertex3f(-1.0f, -1.0f, 1.0f);
96     gl.glColor3f(1.0f, 0.0f, 1.0f); gl.glVertex3f( 1.0f, -1.0f, 1.0f);
97     gl.glColor3f(1.0f, 0.0f, 0.0f); gl.glVertex3f( 1.0f, -1.0f, -1.0f);
98
99     /* 4th face */
100    gl.glColor3f(0.0f, 1.0f, 0.0f); gl.glVertex3f(-1.0f, 1.0f, -1.0f);
101    gl.glColor3f(0.0f, 1.0f, 1.0f); gl.glVertex3f(-1.0f, 1.0f, 1.0f);
102    gl.glColor3f(1.0f, 1.0f, 1.0f); gl.glVertex3f( 1.0f, 1.0f, 1.0f);
103    gl.glColor3f(1.0f, 1.0f, 0.0f); gl.glVertex3f( 1.0f, 1.0f, -1.0f);
104
105    /* 5th face */
106    gl.glColor3f(0.0f, 0.0f, 0.0f); gl.glVertex3f(-1.0f, -1.0f, -1.0f);
107    gl.glColor3f(0.0f, 1.0f, 0.0f); gl.glVertex3f(-1.0f, 1.0f, -1.0f);
108    gl.glColor3f(1.0f, 1.0f, 0.0f); gl.glVertex3f( 1.0f, 1.0f, -1.0f);
109    gl.glColor3f(1.0f, 0.0f, 0.0f); gl.glVertex3f( 1.0f, -1.0f, -1.0f);
110
111    /* 6th face*/
112    gl.glColor3f(0.0f, 0.0f, 1.0f); gl.glVertex3f(-1.0f, -1.0f, 1.0f);
113    gl.glColor3f(0.0f, 1.0f, 1.0f); gl.glVertex3f(-1.0f, 1.0f, 1.0f);
114    gl.glColor3f(1.0f, 1.0f, 1.0f); gl.glVertex3f( 1.0f, 1.0f, 1.0f);
115    gl.glColor3f(1.0f, 0.0f, 1.0f); gl.glVertex3f( 1.0f, -1.0f, 1.0f);
116    gl.glEnd();
117    gl.glPopMatrix();
118 }
119 }

```