



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 #4**  
*OpenGL Timer*

---

**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 int v=0, h=0;
9 int ceta = 0;
10 float width=1.0, height=1.0;
11
12 void myStyleInit ()
13 {
14     glClearColor(1.0, 1.0, 1.0, 0.0);
15     glMatrixMode(GL_PROJECTION);
16     glLoadIdentity();
17     glOrtho(-250, 250, -250, 250, 0, -1);
18 }
19
20 void myDisplay ()
21 {
22     glClear(GL_COLOR_BUFFER_BIT);
23     glMatrixMode(GL_MODELVIEW);
24     glLoadIdentity();
25     glPointSize(4.0);
26
27     glPushMatrix();
28     glColor3f(1.0, 0.0, 0.0);
29     glTranslatef(h, v, 0);
30     glScalef(width, height, 1);
31     glRotatef(ceta, 0, 0, 1);
32     // Draw a 100x100 square
33     glBegin(GL_POLYGON);
34         glVertex2i(-50, -50);
```

```

35     glVertex2i(-50, 50);
36     glVertex2i(50, 50);
37     glVertex2i(50, -50);
38     glEnd();
39     glPopMatrix();
40
41     glutSwapBuffers(); // ← replaces glFlush() in Double Buffering
42 }
43
44 // This function will be called on every key stroke
45 void myKeyboard(unsigned char ch, int x, int y)
46 {
47     if(ch == 'w')
48         v += 10;
49     else if(ch == 's')
50         v -= 10;
51     else if(ch == 'd')
52         h += 10;
53     else if(ch == 'a')
54         h -= 10;
55     else if(ch == 'i')
56         width *= 2;
57     else if(ch == 'o')
58         width /= 2;
59     else if(ch == 'j')
60         height *= 2;
61     else if(ch == 'k')
62         height /= 2;
63     else if(ch == 'r')
64         ceta += 5;
65     else if(ch == 'e')
66         ceta -= 5;
67     else if(ch == 'q')
68         exit(0);
69     glutPostRedisplay(); // from now on, we will use glutPostRedisplay() instead of ←
70     // calling myDisplay() explicitly
71 }
72 void myTimer(int val)
73 {
74     h += 5;
75     glutPostRedisplay();
76     glutTimerFunc(500, myTimer, 0); // reset the timer to 1000 once more
77 }
78
79 int main(int argc, char **argv)
80 {
81     glutInit(&argc, argv);
82     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB); // ← uses Double Buffering
83     glutInitWindowSize(600, 600);
84     glutInitWindowPosition(50, 50);
85     glutCreateWindow("OpenGL Template");
86     myStyleInit();
87     glutDisplayFunc(myDisplay);
88     glutKeyboardFunc(myKeyboard);
89     glutTimerFunc(500, myTimer, 0); // ← call myTimer() after 1000 msec elapses
90     glutMainLoop();
91     return 0;
92 }

```

## 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
6 public class Template extends JFrame implements GLEventListener, KeyListener, ←
    ActionListener{
7     private int moveX = 0, moveY = 0;
8     private int ceta = 0;
9     private float width=1.0f, height=1.0f;
10    private Timer t;
11    public static void main(String[] args){
12        Template window = new Template();
13        window.setVisible(true);
14    }
15    public Template(){
16        setSize(600, 600);
17        setTitle("OpenGL Template");
18        setDefaultCloseOperation(EXIT_ON_CLOSE);
19
20        GLCapabilities capabilities = new GLCapabilities();
21        capabilities.setDoubleBuffered(true); // uses Double Buffering
22        GLCanvas canvas = new GLCanvas(capabilities);
23        canvas.addGLEventListener(this);
24        canvas.addKeyListener(this);
25        getContentPane().add(canvas);
26        Animator animator = new Animator(canvas);
27        animator.start();
28
29        t = new Timer(500, this);
30        t.addActionListener(this); // timer events
31        t.start(); // Runs forever till t.stop() is called
32    }
33    // Called by the drawable immediately after the OpenGL context is initialized
34    public void init(GLAutoDrawable drawable){
35        GL gl = drawable.getGL();
36        gl.glClearColor(1.0f, 1.0f, 1.0f, 0.0f);
37        gl.glMatrixMode(gl.GL_PROJECTION);
38        gl.glLoadIdentity();
39        gl.glOrtho(-250, 250, -250, 250, 0, -1);
40        gl.glMatrixMode(gl.GL_MODELVIEW);
41        gl.glLoadIdentity();
42    }
43    // Called by the drawable to initiate OpenGL rendering by the client
44    public void display(GLAutoDrawable drawable){
45        GL gl = drawable.getGL();
46        gl.glClear(GL.GL_COLOR_BUFFER_BIT | GL.GL_DEPTH_BUFFER_BIT);
47
48        gl.glPushMatrix();
49        gl.glColor3f(1.0f, 0.0f, 0.0f);
50        gl.glTranslatef(moveX, moveY, 0);
51        gl.glScalef(width, height, 1);
52        gl.glRotatef(ceta, 0, 0, 1);
53        // Draw a 100x100 square
54        gl.glBegin(gl.GL_POLYGON);
55        gl.glVertex2i(-50, -50);
56        gl.glVertex2i(-50, 50);
57        gl.glVertex2i(50, 50);
58        gl.glVertex2i(50, -50);
59        gl.glEnd();
60        gl.glPopMatrix();
61
62        drawable.swapBuffers(); // ← replaces gl.glFlush() in Double Buffering
63    }
64    // Called by the drawable during the first repaint after the component has been resized
65    public void reshape(GLAutoDrawable drawable, int x, int y, int w, int h){
66    }
67    // Called by the drawable when the display mode or the display device associated with ←
    the GLAutoDrawable has changed
68    public void displayChanged(GLAutoDrawable drawable, boolean modeChanged, boolean ←
        deviceChanged){

```

```

69     }
70     public void keyPressed(KeyEvent e){
71     }
72     public void keyReleased(KeyEvent e){
73     }
74     public void keyTyped(KeyEvent e){
75         char key = e.getKeyChar();
76         if(key == 'w')
77             moveY += 10;
78         else if(key == 's')
79             moveY -= 10;
80         else if(key == 'd')
81             moveX += 10;
82         else if(key == 'a')
83             moveX -= 10;
84         else if(key == 'i')
85             width *= 2;
86         else if(key == 'o')
87             width /= 2;
88         else if(key == 'j')
89             height *= 2;
90         else if(key == 'k')
91             height /= 2;
92         else if(key == 'r')
93             ceta += 5;
94         else if(key == 'e')
95             ceta -= 5;
96         else if(key == 'q')
97             System.exit(0);
98     }
99     public void actionPerformed(ActionEvent ev){
100         if(ev.getSource() == t){
101             moveX += 5;
102         }
103     }
104 }

```