



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 #2
OpenGL Keyboard

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
10 void myStyleInit ()
11 {
12     glClearColor(1.0, 1.0, 1.0, 0.0);
13     glPointSize(4.0);
14     glMatrixMode(GL_PROJECTION);
15     glLoadIdentity();
16     glOrtho(-250, 250, -250, 250, 0, -1);
17 }
18
19 void myDisplay ()
20 {
21     glClear(GL_COLOR_BUFFER_BIT);
22     glColor3f(0.0,0.0,0.0);
23     glMatrixMode(GL_MODELVIEW);
24     glLoadIdentity();
25     glPointSize(4.0);
26
27     // Draw a 100x100 square
28     glBegin(GL_POLYGON);
29         glVertex2i(-50 + h, -50 + v);    // <— Add v & h to simulate motion
30         glVertex2i(-50 + h, 50 + v);
31         glVertex2i(50 + h, 50 + v);
32         glVertex2i(50 + h, -50 + v);
33     glEnd();
34
```

```

35     glFlush();
36 }
37
38 // This function will be called on every key stroke
39 void keyboard(unsigned char ch, int x, int y)
40 {
41     if(ch == 'w')
42         v += 10;
43     else if(ch == 's')
44         v -= 10;
45     else if(ch == 'd')
46         h += 10;
47     else if(ch == 'a')
48         h -= 10;
49     else if(ch == 'q')
50         exit(0);
51     myDisplay();           // ← remember to call the display function
52 }
53
54 int main(int argc, char **argv)
55 {
56     glutInit(&argc, argv);
57     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
58     glutInitWindowSize(600, 600);
59     glutInitWindowPosition(50, 50);
60     glutCreateWindow("OpenGL Template");
61     myStyleInit();
62     glutDisplayFunc(myDisplay);
63     glutKeyboardFunc(keyboard); // ← call keyboard() on every key stroke
64     glutMainLoop();
65     return 0;
66 }

```

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{
7      private int v=0, h=0;
8      public static void main(String[] args){
9          Template window = new Template();
10         window.setVisible(true);
11     }
12     public Template(){
13         setSize(600, 600);
14         setTitle("OpenGL Template");
15         setDefaultCloseOperation(EXIT_ON_CLOSE);
16         GLCanvas canvas = new GLCanvas(new GLCapabilities());
17         canvas.addGLEventListener(this);
18         canvas.addKeyListener(this);
19         getContentPane().add(canvas);
20         Animator animator = new Animator(canvas);
21         animator.start();
22     }
23     // Called by the drawable immediately after the OpenGL context is initialized
24     public void init(GLAutoDrawable drawable){
25         GL gl = drawable.getGL();
26         gl.glClearColor(1.0f, 1.0f, 1.0f, 0.0f);
27         gl.glMatrixMode(gl.GL_PROJECTION);
28         gl.glLoadIdentity();

```

```

29     gl.glOrtho(-250, 250, -250, 250, 0, -1);
30     gl.glMatrixMode(gl.GL_MODELVIEW);
31     gl.glLoadIdentity();
32 }
33 // Called by the drawable to initiate OpenGL rendering by the client
34 public void display(GLAutoDrawable drawable){
35     GL gl = drawable.getGL();
36     gl.glClear(GL.GL_COLOR_BUFFER_BIT | GL.GL_DEPTH_BUFFER_BIT);
37     gl.glColor3f(0.0f, 0.0f, 0.0f);
38     // Draw a 100x100 square
39     gl.glBegin(gl.GL_POLYGON);
40         gl.glVertex2i(-50 + h, -50 + v);    // <— Add v & h to simulate motion
41         gl.glVertex2i(-50 + h, 50 + v);
42         gl.glVertex2i(50 + h, 50 + v);
43         gl.glVertex2i(50 + h, -50 + v);
44     gl.glEnd();
45     gl.glFlush();
46 }
47 // Called by the drawable during the first repaint after the component has been resized
48 public void reshape(GLAutoDrawable drawable, int x, int y, int w, int h){
49 }
50 // Called by the drawable when the display mode or the display device associated with ←
51 // the GLAutoDrawable has changed
52 public void displayChanged(GLAutoDrawable drawable, boolean modeChanged, boolean ←
53 // deviceChanged){
54 }
55 public void keyPressed(KeyEvent e){
56 }
57 public void keyReleased(KeyEvent e){
58 }
59 public void keyTyped(KeyEvent e){
60     char key = e.getKeyChar();
61     if(key == 'w')
62         v += 10;
63     else if(key == 's')
64         v -= 10;
65     else if(key == 'd')
66         h += 10;
67     else if(key == 'a')
68         h -= 10;
69     else if(key == 'q')
70         System.exit(0);
71 }
72 }

```