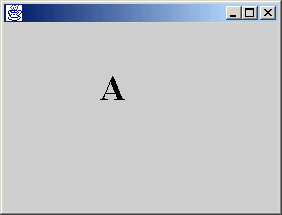
**KEY EVENTS**

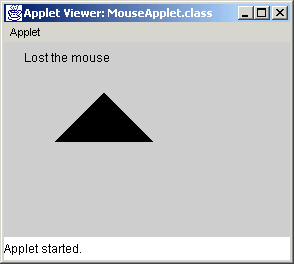
/\* Displays a key pressed ny the user. Moves the  
\* character to the right if the user presses the right arrow  
\* key and to the left if the user presses the left arrow  
\* key. Moves ten pixels if the user hold down the  
\* control key and two pixels otherwise.  
\*/



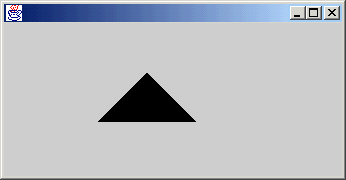
import java.awt.\*;  
import java.awt.event.\*;   
import javax.swing.\*;  
  
public class Key extends JFrame  
implements KeyListener {  
public static int SLOW = 2;  
public static int FAST = 10;  
private int x=100, y=100;  
private char theKey = 'A';   
private Font f = new Font("Serif",Font.BOLD,36);  
private int deltaX = SLOW;  
  
public Key() {  
setFont(f);   
addKeyListener(this);  
requestFocus();  
}  
public void paint(Graphics g) {  
super.paint(g);  
g.drawString(String.valueOf(theKey),x,y);  
}  
public void keyPressed(KeyEvent event){  
int code = event.getKeyCode();   
if (code == event.VK\_CONTROL) deltaX = FAST;   
else   
if (code == event.VK\_RIGHT){x += deltaX; repaint();}  
else if (code == event.VK\_LEFT){ x -= deltaX; repaint();}  
}  
public void keyReleased(KeyEvent event) {  
if (event.getKeyCode() == event.VK\_CONTROL) deltaX = SLOW;  
}  
public void keyTyped(KeyEvent event) {  
theKey = event.getKeyChar();  
repaint();   
}  
public static void main(String [] args)  
{Key f = new Key();  
f.setSize(400,400);  
f.setVisible(true);  
}  
  
}

MOUSE EVENTS

/\* Fills a red triangle when the user presses the mouse inside it.  
\* Fills the triangle in blue when the user releases the mouse inside it.  
\* Draw "Got the mouse" when the user enters an applet, and draws "Lost the mouse"  
\* when the user exits the applet.  
\*/

  
  
  
import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*;  
  
public class MouseFrame extends JFrame  
  
{ private int [] x = {50,100,150};  
private int [] y = {100,50,100};  
private Polygon p = new Polygon(x,y,3);  
private int oldx;  
private int oldy;  
private String mouse="";  
  
MouseFrame() {  
addMouseListener(new MousePressListener());  
}  
public void paint(Graphics g) {  
super.paint(g);  
g.drawString(mouse,200,200);  
g.fillPolygon(p);  
}  
  
class MousePressListener extends MouseAdapter {  
public void mousePressed(MouseEvent event) {  
int x = event.getX();  
int y = event.getY();  
if (p.contains(x,y)){  
setForeground(Color.red);  
repaint();  
}  
}  
public void mouseReleased(MouseEvent event) {  
int x = event.getX();  
int y = event.getY();  
if (p.contains(x,y)){  
setForeground(Color.blue);  
repaint();  
}  
}  
public void mouseEntered(MouseEvent event)  
{mouse = "Got the mouse";  
repaint();}  
public void mouseExited(MouseEvent event)  
{mouse = "Lost the mouse";  
repaint();}  
  
}  
public static void main(String [] args)  
{MouseFrame f = new MouseFrame();  
f.setSize(440,400);  
f.setVisible(true);  
}  
  
}

/\* Drags a triangle to a new location  
\* using the position of the mouse to  
\* determine how to move the polygon.  
\*/



import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*;  
  
public class Mouse extends JFrame  
implements MouseMotionListener {  
private int [] x = {50,100,150};  
private int [] y = {100,50,100};  
private Polygon p = new Polygon(x,y,3);  
private int oldx;  
private int oldy;  
  
public Mouse() {  
addMouseListener(new MousePressListener());  
addMouseMotionListener(this);  
}  
public void paint(Graphics g) {  
super.paint(g);  
g.fillPolygon(p);  
}  
public void mouseMoved(MouseEvent event) { }  
public void mouseDragged(MouseEvent event) {  
int x = event.getX();  
int y = event.getY();   
if (p.contains(x,y)){  
p.translate(x-oldx,y-oldy);  
oldx=x;  
oldy=y;   
repaint();  
}  
}  
class MousePressListener extends MouseAdapter {  
public void mousePressed(MouseEvent event) {  
int x = event.getX();  
int y = event.getY();   
if (p.contains(x,y)){  
oldx = x;  
oldy = y;  
}   
}  
}  
public static void main(String [] args)  
{Mouse f = new Mouse();  
f.setSize(440,200);  
f.setVisible(true);  
}  
  
}