

## CECS277 – Lecture 16 – GUI Applications – Event Listeners

**Mouse Event Handlers** – Allows the user to use the mouse as input to the program. Event Handlers allow your program to react whenever the user moves or clicks the mouse. All five of these methods must be added if any of them are to be used.

mouseClicked() – called when the mouse is clicked (pressed and released).

mousePressed() – called when the mouse button is pressed down but not released.

mouseReleased() – called when the mouse button has been released after being pressed.

mouseEntered() – called when the mouse pointer enters a specific area.

mouseExited() – called when the mouse pointer leaves a specific area.

```
public class MApp extends JPanel implements MouseListener {
    private Rectangle r = new Rectangle( 10, 10, 25, 25 );
    public MApp() {
        addMouseListener( this );
    }
    public void mouseClicked ( MouseEvent e ) {
        Point click = new Point( e.getX(), e.getY() );
        System.out.println( click );
        if( r.contains( click ) ) {
            System.out.println("Clicked in Rectangle");
        }
    }
    public void mousePressed (MouseEvent e) {}
    public void mouseReleased (MouseEvent e) {}
    public void mouseEntered (MouseEvent e) {}
    public void mouseExited (MouseEvent e) {}
}
```

**MouseMotionListener** – allows you to track where the mouse is and performs events when the pointer moves over certain locations. Also allows you to perform click and drag actions. Both of these methods must be created if either of them are to be used.

mouseMoved() – called whenever the mouse is moved.

mouseDragged() – called whenever the mouse is moved while the button is pressed.

```
public class MApp extends JPanel implements MouseMotionListener {
    public MApp() {
        addMouseMotionListener( this );
    }
    public void mouseMoved( MouseEvent e ) {
        Point move = new Point( e.getX(), e.getY() );
        System.out.println( move );
    }
    public void mouseDragged( MouseEvent e ) {}
}
```

**KeyListener** – allows you to know if a specific key was pressed. The key listener sends an event every time a key is pressed on the keyboard and executes the code in the functions.

Just like with mouse events, you can test the KeyEvent parameter to find out information about the event. If you are listening for a particular key to be pressed, use an if statement to see if the KeyEvent matches the key you are looking for (type KeyEvent and then a dot to see a list of possible keys (or look at the Java API page for KeyEvent: <https://docs.oracle.com/javase/8/docs/api/java/awt/event/KeyEvent.html>)).

keyPressed() – called whenever a key is pressed down.

keyReleased() – called whenever a key comes up.

keyTyped() – called when a printable character is sent to the system input (repeatable).

```
public class KApp extends JPanel implements KeyListener {
    public KApp() {
        addKeyListener( this );
        setFocusable( true );
    }
    public void keyPressed( KeyEvent e ) {
        if( e.getKeyCode() == KeyEvent.VK_A ) {
            System.out.println( "You pressed 'A'" );
        }
        if( e.getKeyCode() == KeyEvent.VK_ENTER ) {
            System.out.println("You pressed Enter");
        }
        if( e.getKeyCode() == KeyEvent.VK_UP ) {
            System.out.println("You pressed Up Arrow");
        }
    }
    public void keyReleased( KeyEvent e ) {}
    public void keyTyped( KeyEvent e ) {}
}
```