Keyboard/Mouse Input

SI@UCF Java/Java GUI Recitation

Keyboard and mouse input are some of the most vital aspects of GUI programs in Java. It's hard to make many useful programs without being able to get input from the user, and that's exactly what we're going to learn how to do.

Documentation:

http://docs.oracle.com/javase/7/docs/api/java/awt/event/MouseListener.html

http://docs.oracle.com/javase/7/docs/api/java/awt/event/KeyListener.html

http://docs.oracle.com/javase/7/docs/api/java/awt/event/KeyEvent.html

Common and Fun Methods:

keyPressed(KeyEvent e)
Invoked when a key has been pressed.

keyReleased(KeyEvent e)
Invoked when a key has been released.

keyTyped(KeyEvent e)
Invoked when a key has been typed.

mouseClicked(MouseEvent e)
Invoked when the mouse button has been clicked (pressed and released) on a component.

mouseEntered(MouseEvent e)
Invoked when the mouse enters a component.

mouseExited(MouseEvent e)
Invoked when the mouse exits a component.

mousePressed(MouseEvent e)
Invoked when a mouse button has been pressed on a component.

mouseReleased(MouseEvent e)
Invoked when a mouse button has been released on a component.

getKeyChar()

Returns the character associated with the key in this event.

Below is an example of an application using a KeyListener to take input from the user. A MouseListener works almost exactly the same way, it listens to a component and does actions based on events that occur.

```
class StudentWorkSolution extends JComponent {
	Color outlineColor, backgroundColor, highlightColor;

	int xCoordinate = -50, yCoordinate = -50;

	public void initialize() { }

	public void draw(Graphics g) { g.setColor(this.backgroundColor); g.fillRect(0,
		0, 300, 200); g.setColor(this.outlineColor);

		for (int i = 0; i < 3; i++) {
			g.drawRect(i*50 + 50, 50, 50, 50);

			if (i != 9) {
				g.drawString("" + (i+1), i*50 + 75, 75);
			}
			else {
				g.drawString("" + 0, i*50 + 75, 75);
			}
		}

		g.setColor(this.highlightColor); g.drawRect(xCoordinate, yCoordinate, 50, 50);
	}

	public void highlightSquare(int x, int y) { xCoordinate = x;
		yCoordinate = y; this.repaint();
	}

	public StudentWorkSolution() { this.initialize(); outlineColor =
		Color.black;
		backgroundColor = Color.white; highlightColor = Color.red;
	}

	public void paintComponent(Graphics g) { this.draw(g);
	}
}
```

```
public class keyListenerPractice { static JFrame frame;
	static StudentWorkSolution student;

	public static void main(String[] args) { frame = new JFrame("Logotech");
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); frame.setSize(300, 200);

		student = new StudentWorkSolution();
		student.addKeyListener(new KeyboardListener());
		student.setFocusable(true);

		frame.add(student);
		frame.setResizable(false);
		frame.setVisible(true);
	}

	static class KeyboardListener implements KeyListener { public void
		keyTyped(KeyEvent e) { }

		public void keyPressed(KeyEvent e) { if (e.getKeyChar() == '1') {
				student.highlightSquare(50, 50);
			}
			else if (e.getKeyChar() == '2') { student.highlightSquare(100, 50);
			}
			else if (e.getKeyChar() == '3') { student.highlightSquare(150, 50);
			}
		}

		public void keyReleased(KeyEvent e) { student.highlightSquare(-50, -50);
		}
	}
}
```