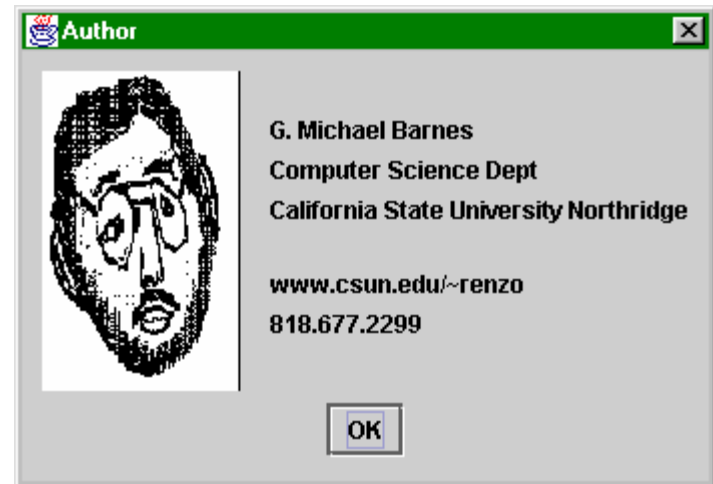
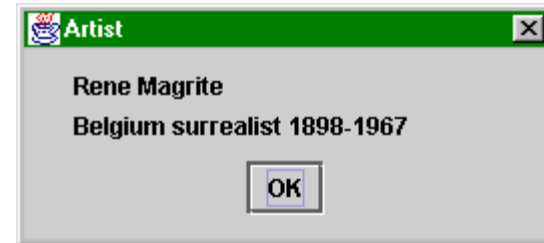
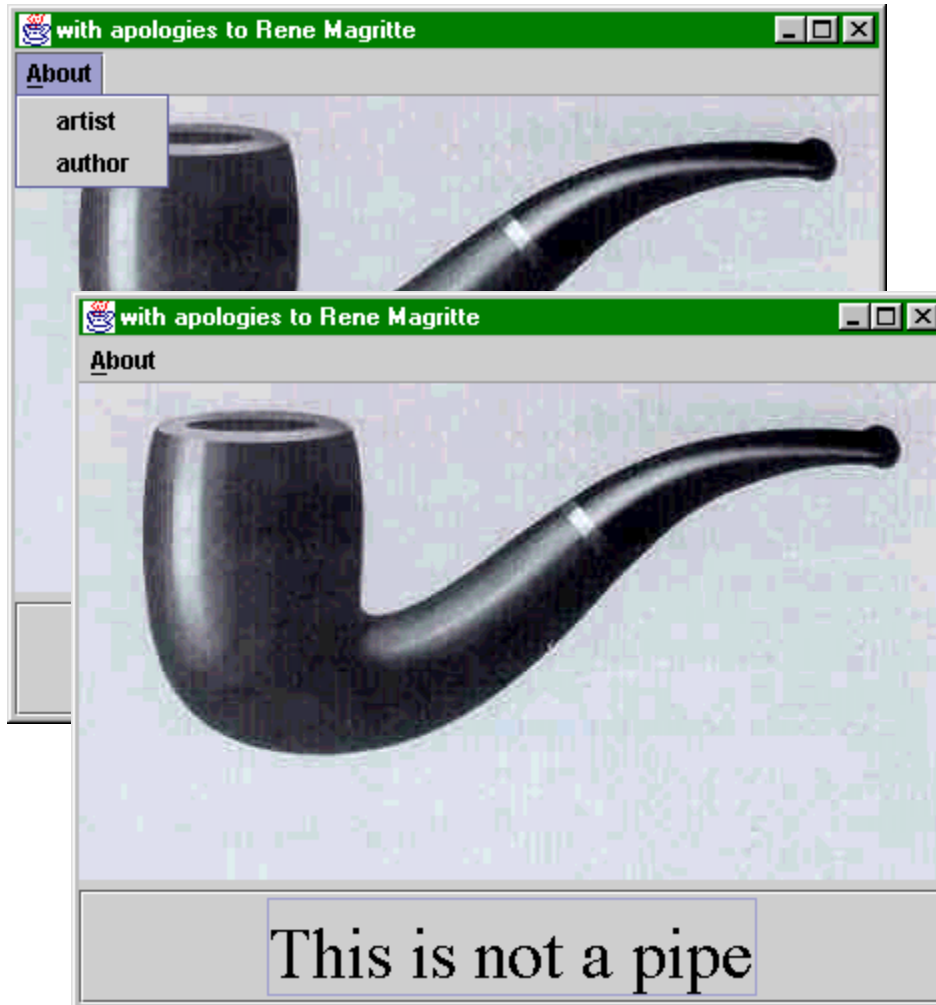


Graphical User Interfaces: GUI



Why GUIs ?

Computer Graphics technology enabled GUIs and computer gaming.

GUI's were / are breakout computer technology.

Without a GUI there would be not be any, or much less:

- Computer based industry -- USA's major productivity advantage

 - Computer systems in the homes and schools

- Internet, Web, E-commerce

 - before Mosaic -- telnet, ftp, gopher, WAIS

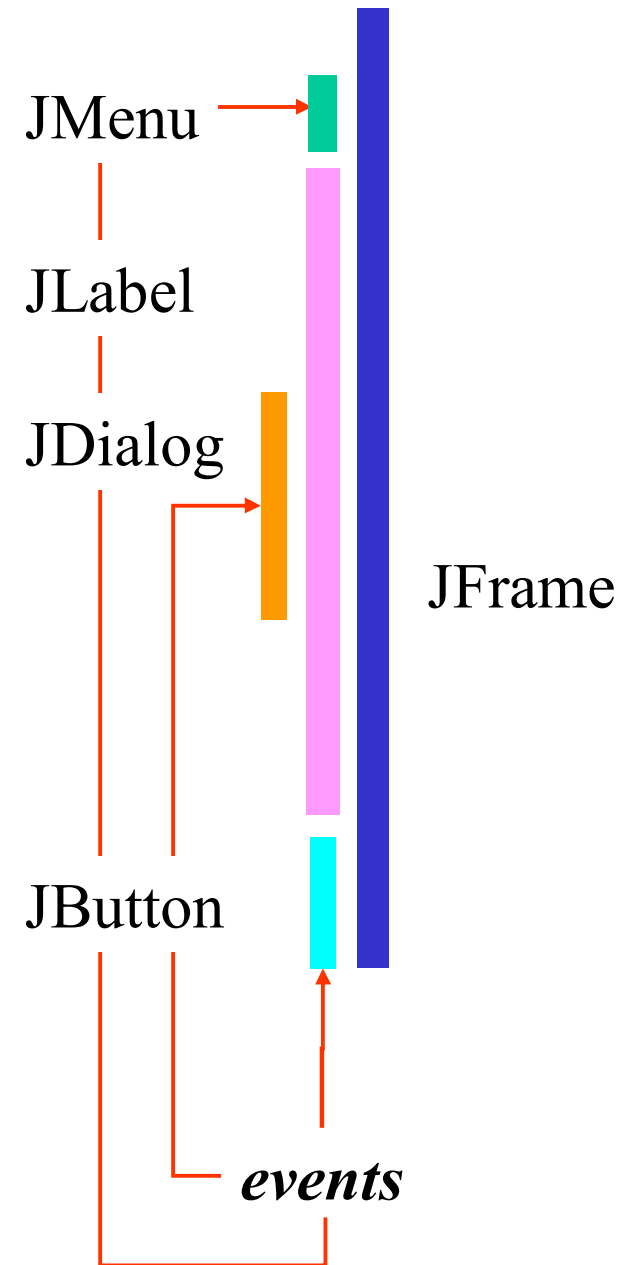
 - after Mosaic -- AOL, Amazon.com, Napster

- PDA's, cell phones

How much software would you use / buy that does not have a GUI interface?

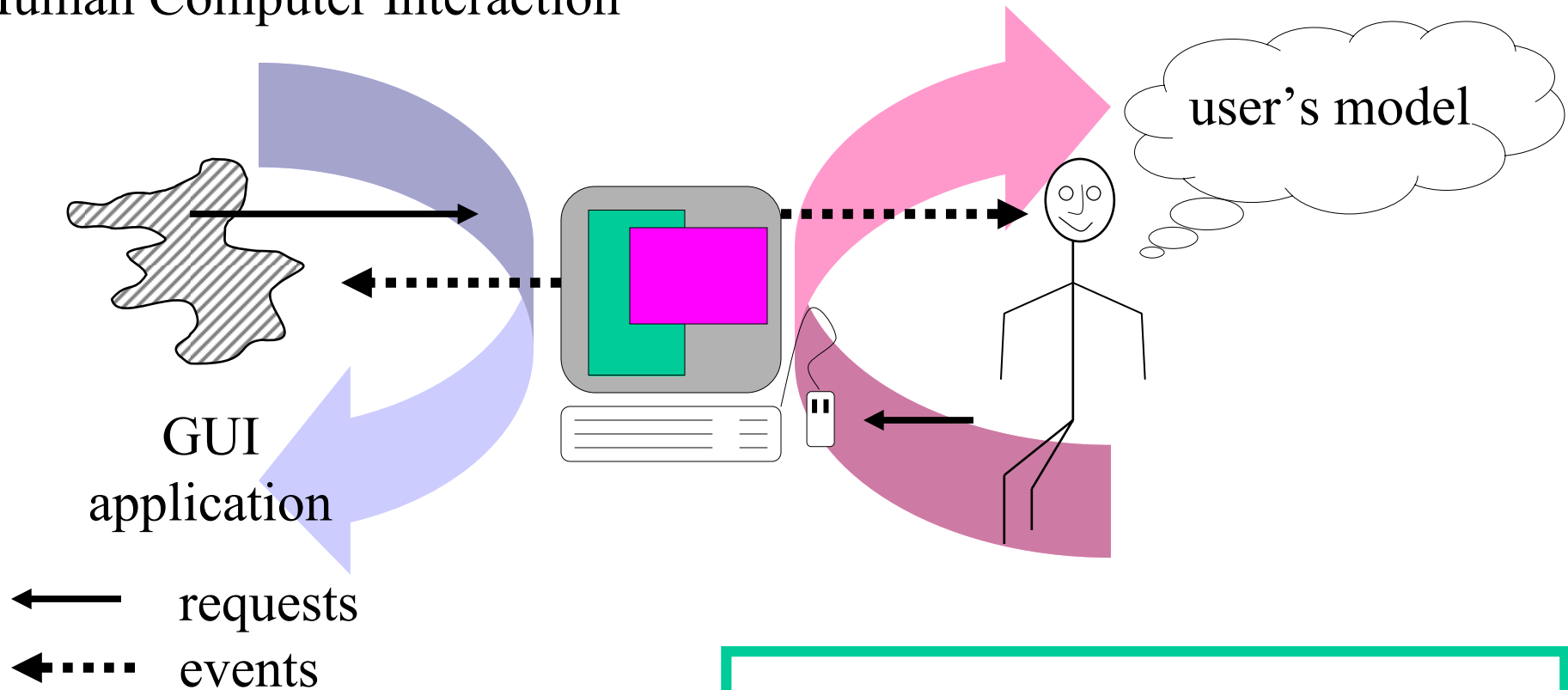
How many computer games without graphics would you play?

Where would Microsoft be without Windows? Apple w/o Macintosh?



1963	SketchPad	interactive computer graphics system
1968	NLS, augment	2D GUI, mouse, hypertext, windows, editing
1977	Apple II	color graphics personal computer
1981	Xerox Star	OOP, bit mapped GUI, ethernet, smalltalk
1984	Macintosh	personal GUI
1984	X Windows	unix / C, networked GUI, vendor independent.
1985	MS Windows	intel GUI
1989	Next	OOP GUI, interface builder
1992	3D GUIs	Xerox Parc: data wall, cone tree ...
1993	Newton	handheld PDA GUI
1994	Mosaic	Graphical Web browser
1995	Java	System independent local web GUIs
2003	.NET	Language independent GUI local web GUIs

Human Computer Interaction



Application makes display requests to system

System display is an event to the user

User responds to display with input requests (mouse, keyboard ...)

System routes user input to application as events

Application handles events and may make display requests ...

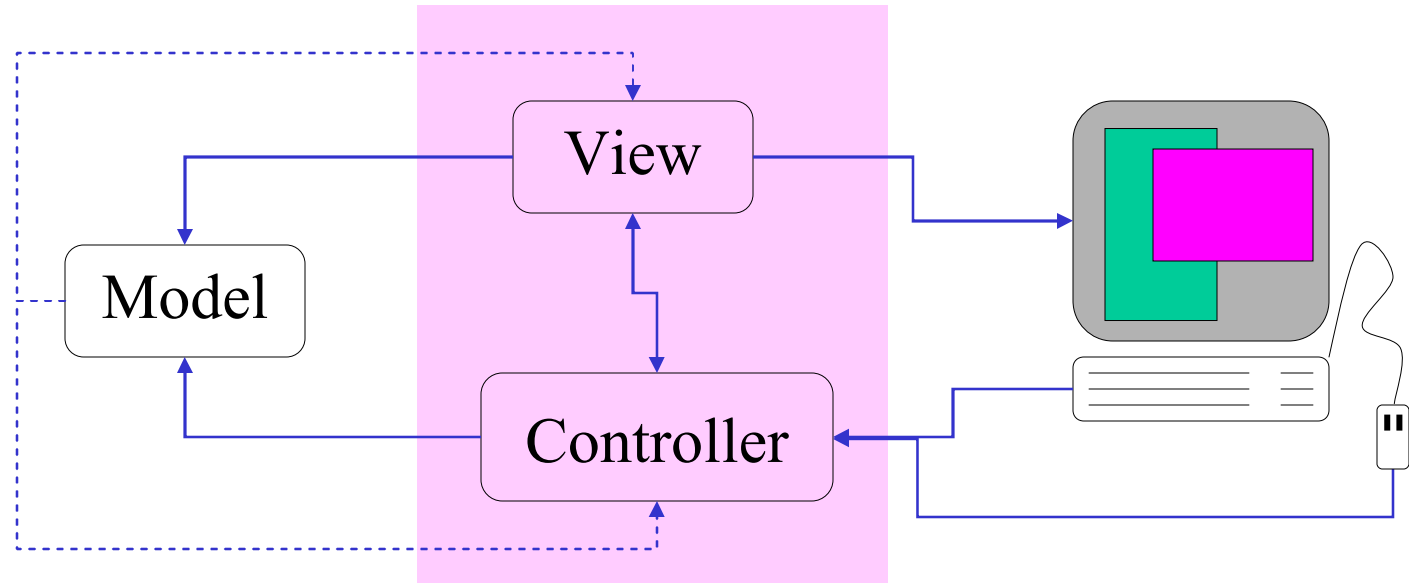
MVC architecture

SmallTalk's window concept had 3 object components

Model: object to be viewed or modified

View: object that determines how the model is to be displayed
“draw to display requests”

Controller: object that handles user interaction with "window"
“handle events”

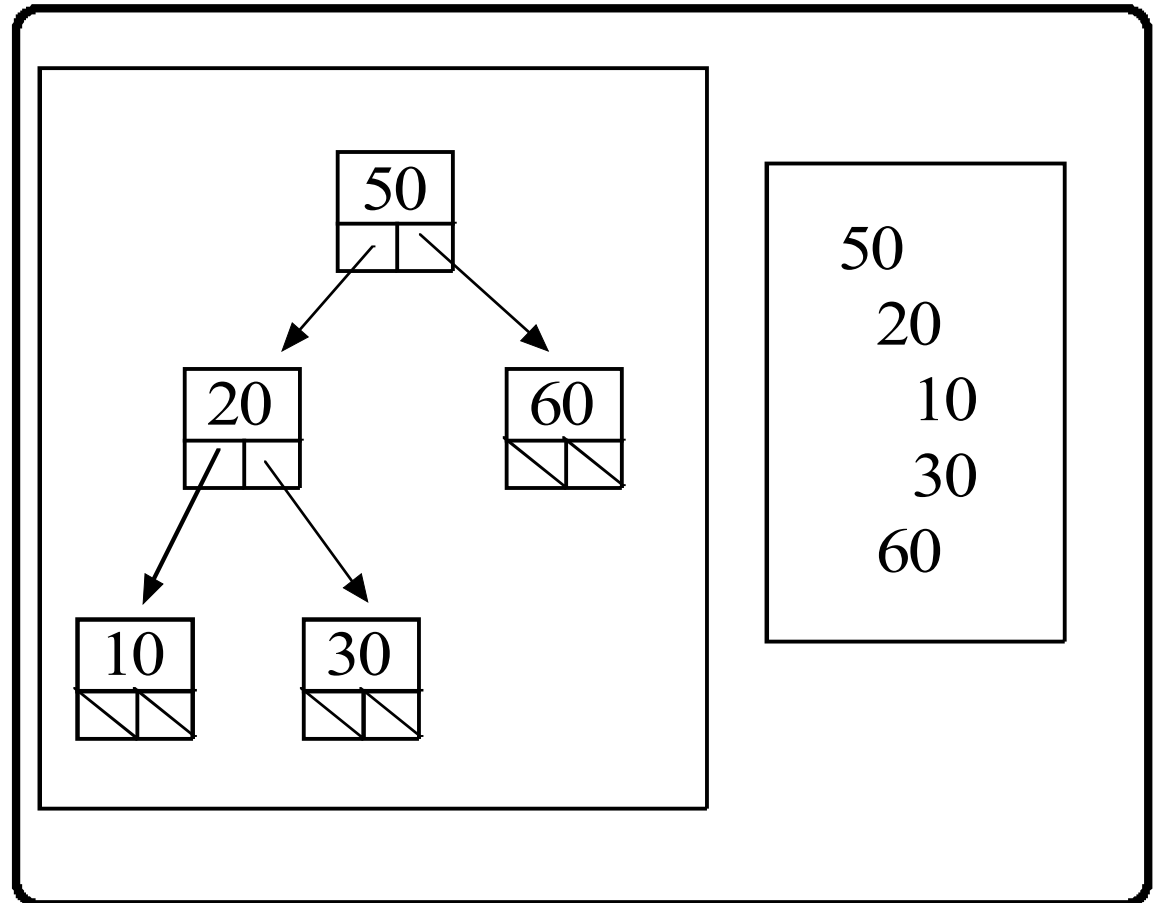


Consider a binary
tree visual
simulation:

1 model
internal binary
tree algorithms

2 views
graphic
indented text

3 controllers
keyboard
voice input



\$ add 30
\$ delete 15
\$

“add 30”
“delete 15”

Model View architecture

MVC is not used in modern GUIs

Java UI-delegate combines MVC view-controller aspects

UI objects know how to : draw themselves, handle events, determine their size ...

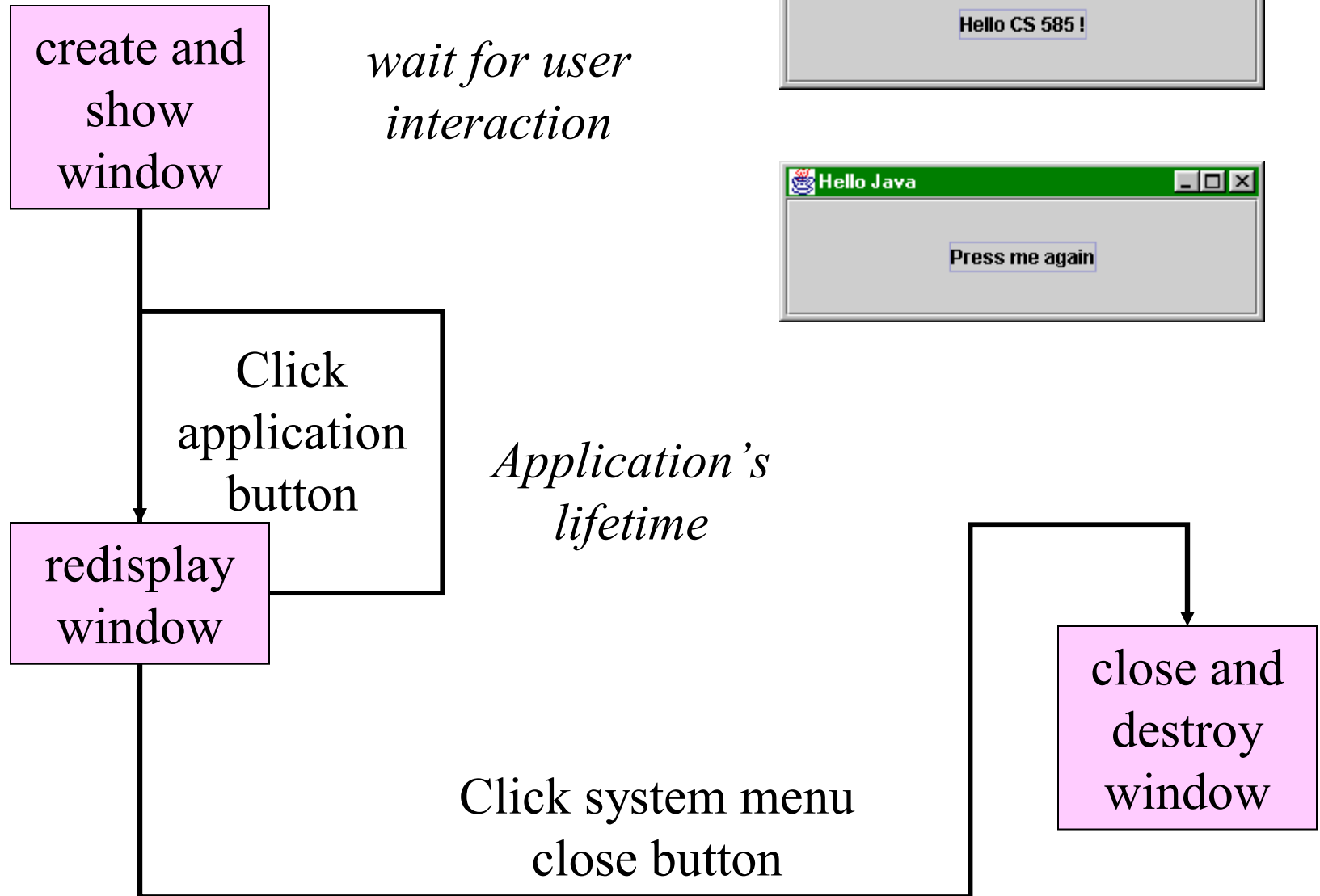
Java UI classes can have associated model objects

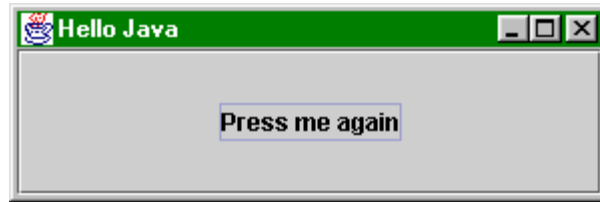
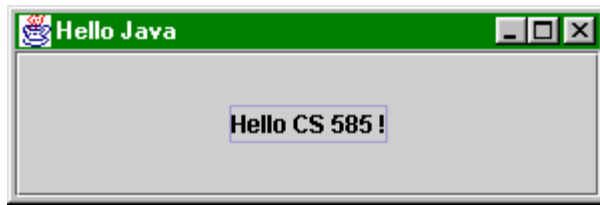
MS MFC uses a document - view architecture where view classes combine aspects of the MVC view-controller and document classes represent models

MS .NET uses "controls", "views", and models. Views are complex controls.

I will use model and view as references to the model and the view-controller concepts.

HelloSwing example





Trace of HelloSwing

```
HelloSwing :: enter main()
HelloSwing :: enter HelloSwing()
HelloSwing :: leave HelloSwing()
HelloSwing :: leave main()
HelloSwing :: actionPerformed()
HelloSwing :: actionPerformed()
```

```
/**
 * HelloSwing Java example program for CS 585.
 * Frame window with a button.
 * Click button and label changes.
 * Trace statements are added to illustrate order of method calls.
 *
 * Mike Barnes 11/9/99
 */

// need to import awt components and events and swing components.
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
```

Look at HelloSwing.html generated with javadoc: inheritance, documentation.

```
/**
 * HelloSwing is an application - subclass Frame for window
 */
public class HelloSwing extends JFrame {
    private JButton aButton;
    private boolean toggle = true;

    //All applications must have a main method.
    public static void main(String args[]) {
        System.out.println("HelloSwing :: enter main()");
        HelloSwing app = new HelloSwing("Hello Java");
        app.setSize(300,100);
        app.show (); // or, app.setVisible(true);
        // need to be able to close the window. 2 approaches:
        // app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        app.addWindowListener( new WindowAdapter() {
            public void windowClosing(WindowEvent e)
                {System.exit(0);} });
        System.out.println("HelloSwing :: leave main()");
    }
}
```

```
// the constructor of the frame window
public HelloSwing(String frameTitle) {
    super(frameTitle);
    System.out.println("HelloSwing :: enter HelloSwing()");
    Container contentPane = getContentPane();
    aButton = new JButton("Hello CS 585 !");
    // listen for JButton events
    aButton.addActionListener(new JButtonListener());
    contentPane.add(aButton, "Center");
    System.out.println("HelloSwing :: leave HelloSwing()");
}

// the "callback function" to handle button press events
class JButtonListener implements ActionListener {
    public void actionPerformed(ActionEvent event) {
        System.out.println("HelloSwing :: actionPerformed()");
        if (toggle) {
            aButton.setText("Press me again");
            toggle = false; }
        else {
            aButton.setText("Hello CS 585 !");
            toggle = true; }
    }
}
```