

WIDGETS

CMPT 381

Review

- Parts of a UI Toolkit:
 - Widgets and components
 - Layout management
 - Event handling
 - Model-View-Controller architecture
 - Graphics

Example UI toolkits

Toolkit	OS	Language
Tk	Multiple	Tcl, Perl, Python
JavaFX	Multiple	Java
MFC / WPF	Windows	C, C++, VB, C#
Aqua/Carbon	Mac OS X	Objective C
Gtk	Multiple	C, C++, Perl, Python
Qt	Multiple	C++

Widgets

Widget history

Widget interaction model

Simple widgets

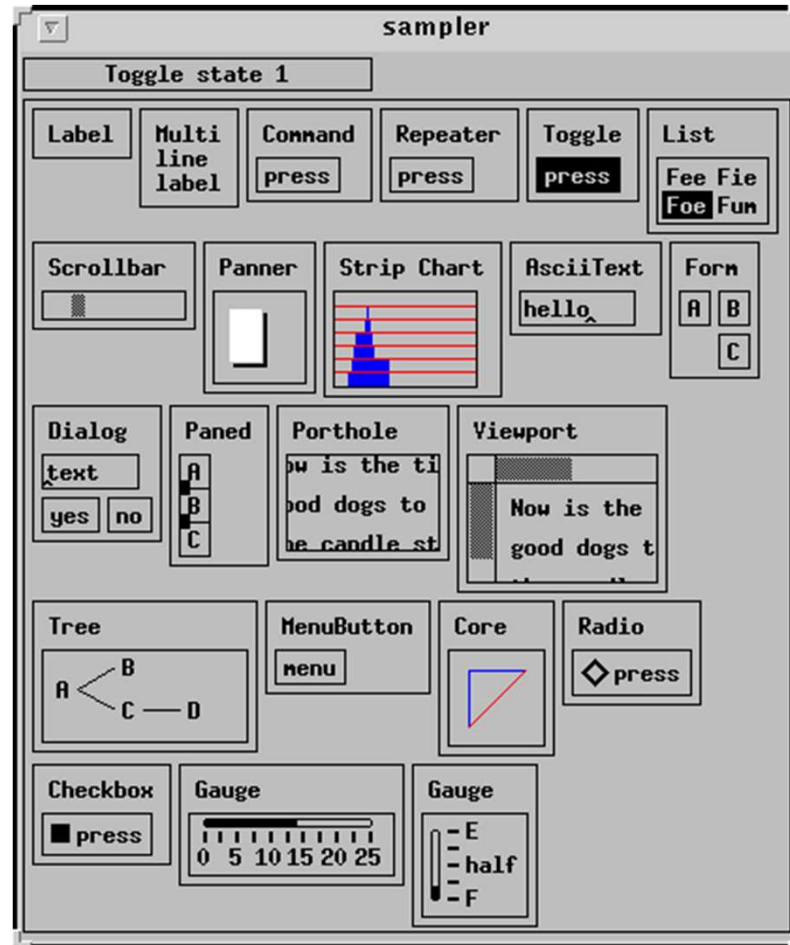
Containers

Abstract Model Widgets

Widgets History

- ‘Evolution’ of the GUI 1981 – 2009
 - <http://www.youtube.com/watch?v=TZGGUrom1Mg>
- 1981 Xerox Star:
 - <https://www.youtube.com/watch?v=Cn4vC80Pv6Q>
- 1983 Apple Lisa
- 1984 Apple Macintosh
 - <https://www.youtube.com/watch?v=2zfqw8nhUwA>
- 1984 Project Athena at MIT develops reusable components for interaction called “widgets”

Athena widgets



Widgets

- UI controls, components, and devices
 - E.g. buttons, menus, sliders, scrollbars
- Widgets are just packaged code
 - API for creation and run-time control
 - Run-time user interaction and behaviour
- Standardize look and feel
- Improve usability: learning and memory of how to use widgets, provide consistent aesthetics
- Provide more reliable implementation, easier design and more rapid development

Widgets in a typical UI toolkit

- Containers
 - Components that hold and organize other widgets
 - Top-Level Containers
 - Internal Containers
- Basic controls
 - Components to get input and show simple state
- Uneditable information displays
 - Components for providing the user with information
- Editable displays of formatted information
 - Components for editable formatted information

Widget interaction model

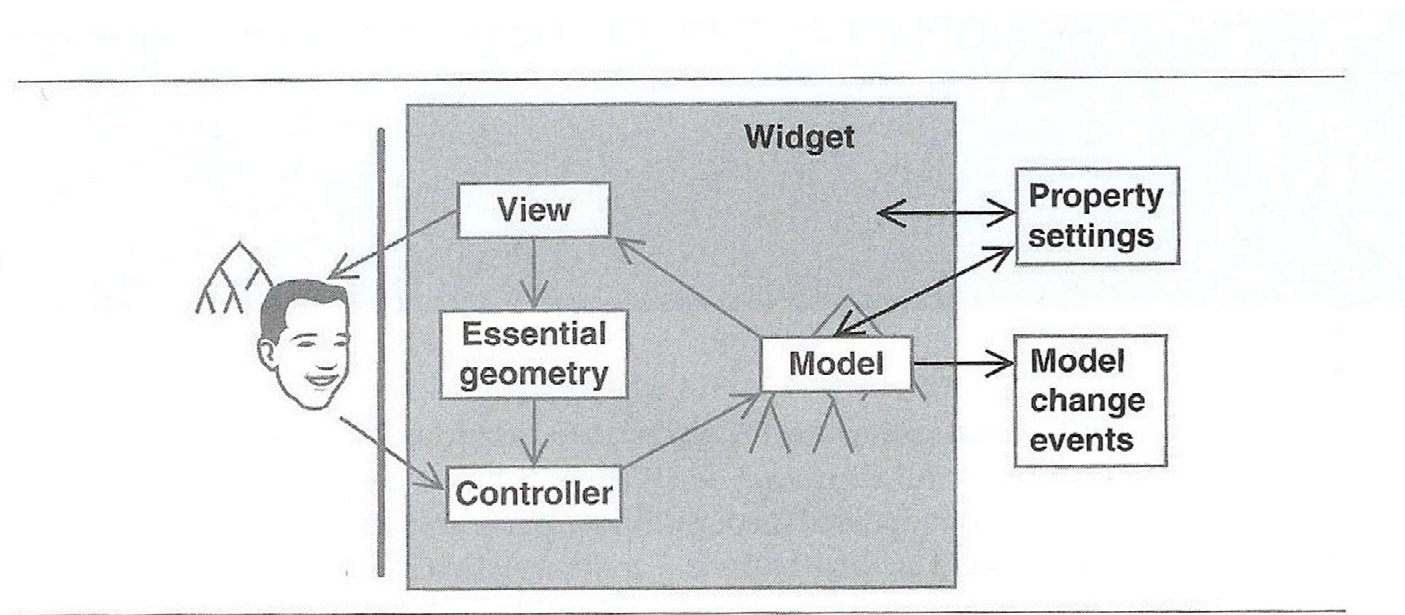


Figure 4.1 – Widget architecture

Widget interaction model

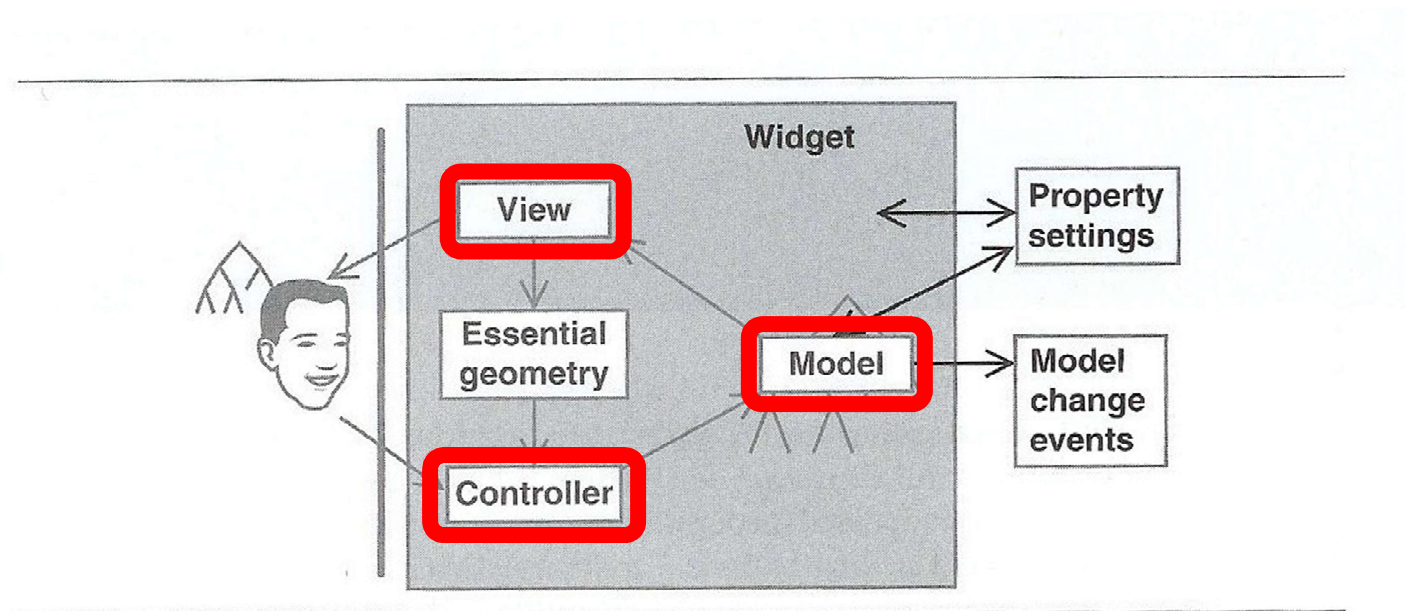


Figure 4.1 – Widget architecture

Widget overviews

- http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/ui_controls.htm
- <http://developer.android.com/guide/topics/ui/index.html>



Simple Widgets

Simple Widgets

- Virtual input devices
 - Buttons
 - No real model
 - Just generate events
 - Widgets with a minimal model
 - Model integrated with widget
 - e.g., a number or text string

Buttons



UI Toolkit interlude...

- What is a widget in terms of the UI toolkit?

JavaFX components

- JavaFX widgets are part of a class hierarchy
 - Control is the parent class of all JavaFX simple widgets
 - openjfx.io/javadoc/11/javafx.controls/javafx/scene/control/Control.html
 - Inheritance abstracts out common functionality:
 - Customizing appearance
 - Setting state
 - Handling events
 - Painting
 - Laying out interior components (if container)
 - Getting/setting size and position information

JavaFX Button API

- In Java, a button is an object and has a class
 - Run-time manipulation through method calls
 - openjfx.io/javadoc/11/javafx.controls/javafx/scene/control/Button.html
 - Sample constructors
 - Button()
 - Creates a button with an empty string for its label.
 - Button(String text)
 - Creates a button with the specified text as its label.
 - Button(String text, Node graphic)
 - Creates a button with the specified text and icon for its label.
- Example:

```
Button myButton = new Button("Click Me");
```

JavaFX Button API

- Sample methods (from several dozen)
 - (inherited from class ButtonBase):
 - `void setOnAction(EventHandler<ActionEvent> value)`
 - Adds an event handler to the button.
 - `String getText()`
 - Returns the button's text.
 - `void setText(String text)`
 - Sets the button's text.
 - `void setDisabled(boolean b)`
 - Enables/disables the button.
- **Example:**
`myButton.setText("Hello World");`

Widgets in Android

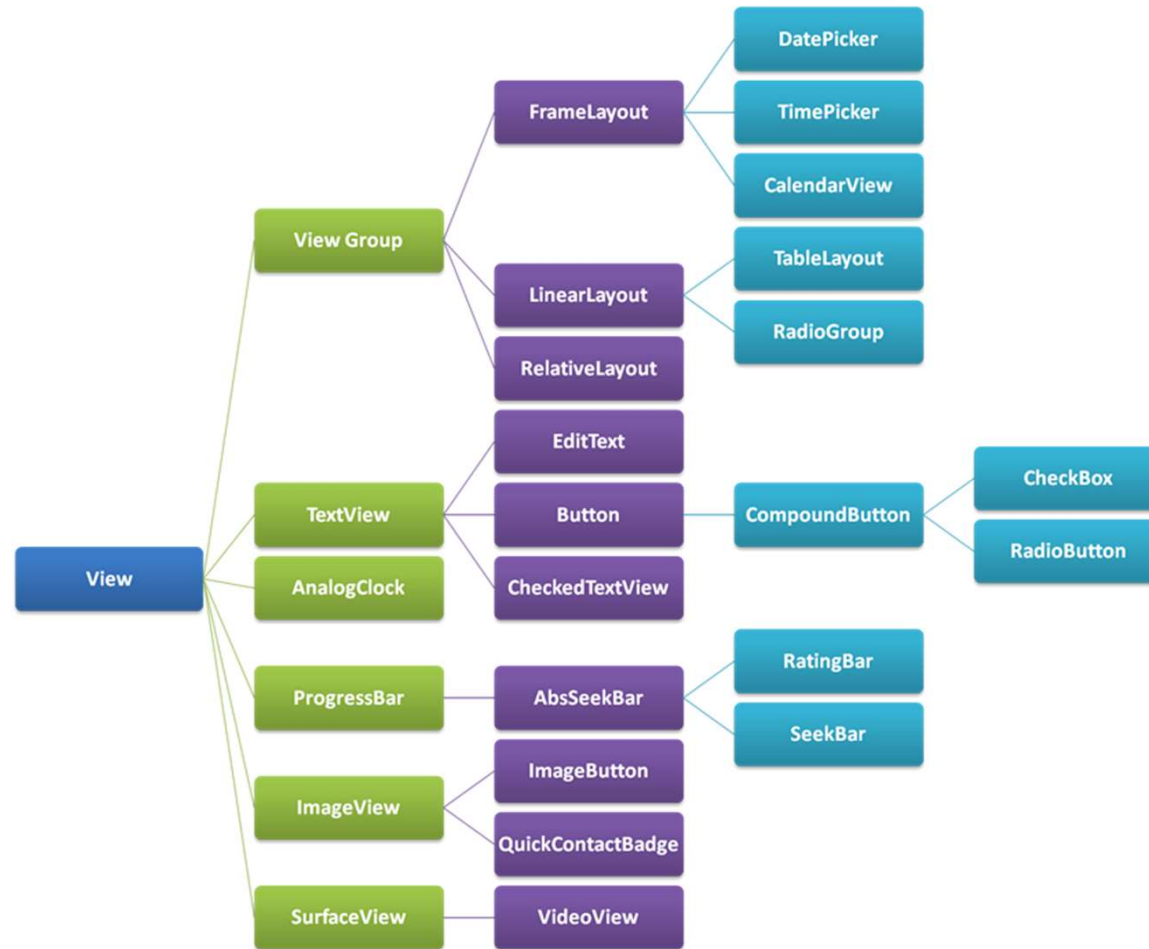
- Note: “App Widgets” are not the same as widgets
- All widgets in Android inherit from View
- Created either through code or XML

Declarative button (Android)

- Created either through code or XML
 - Attributes also specified in XML

```
<Button android:id="@+id/my_button"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:drawableLeft="@drawable/button_icon"  
        android:text="@string/button_text"  
... />
```

Buttons in Android



Buttons in Android

- Working with the button in code:
 - There are equivalent toolkit methods for each XML attribute and event

```
final Button button = (Button) findViewById(R.id.my_button);
button.setHeight(50);
button.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        // Perform action on click
    }
});
```

Buttons in Android

- Creating the button in code:

```
Button button1 = new Button(this);  
button1.setText("Hello World");  
someLayoutContainer.addView(b1);
```

Back to widgets...

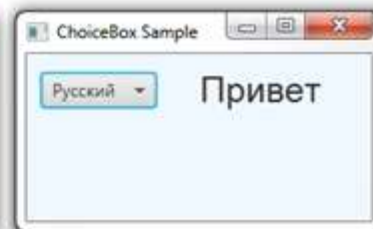
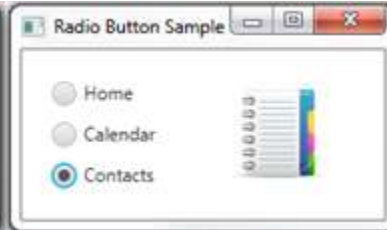
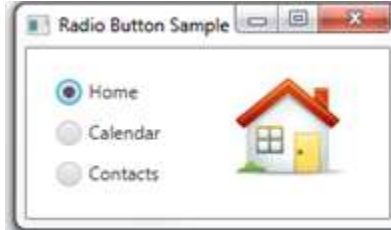
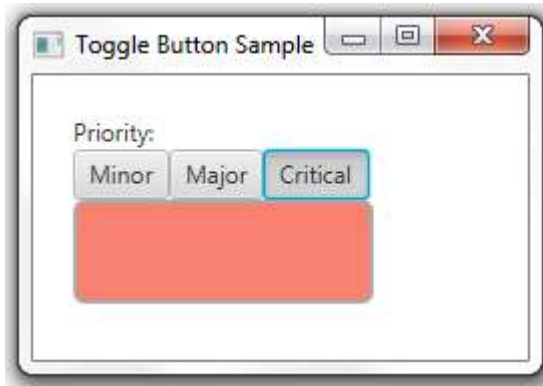
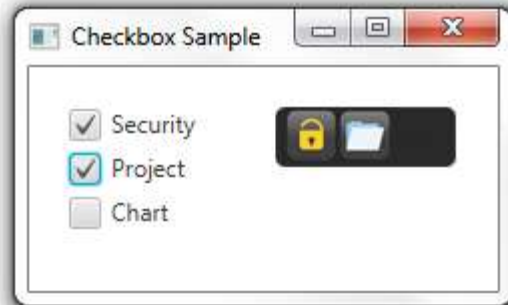
- Widgets with simple state

Information widgets

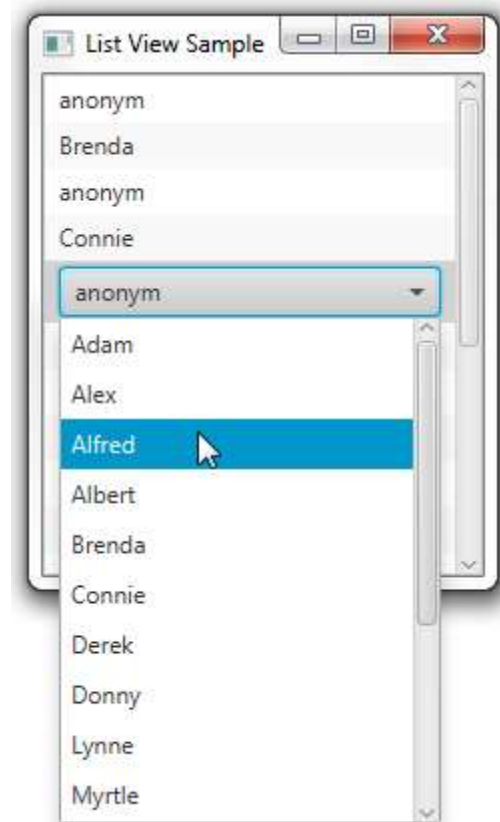
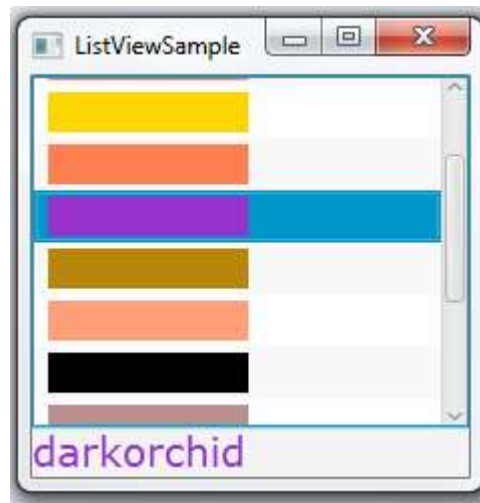
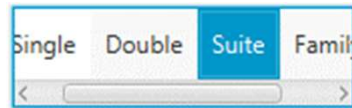
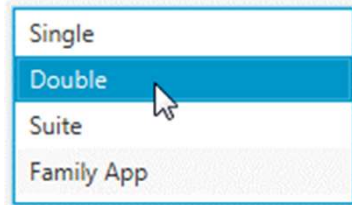
- Labels



State values



Selection(s) from a List

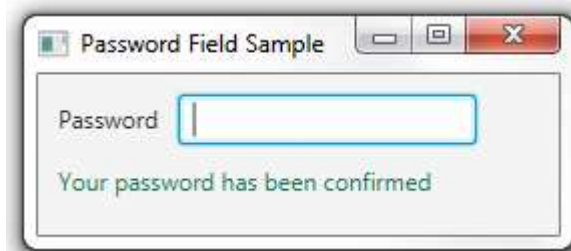


Text Widgets

Name:



A window titled "Text Field Sample" with standard window controls. It contains three text input fields. The first field contains "John", the second "Smith", and the third "Very helpfull!". To the right of the first two fields are "Submit" and "Clear" buttons. Below the third field is a text label: "John Smith, thank you for your comment!"

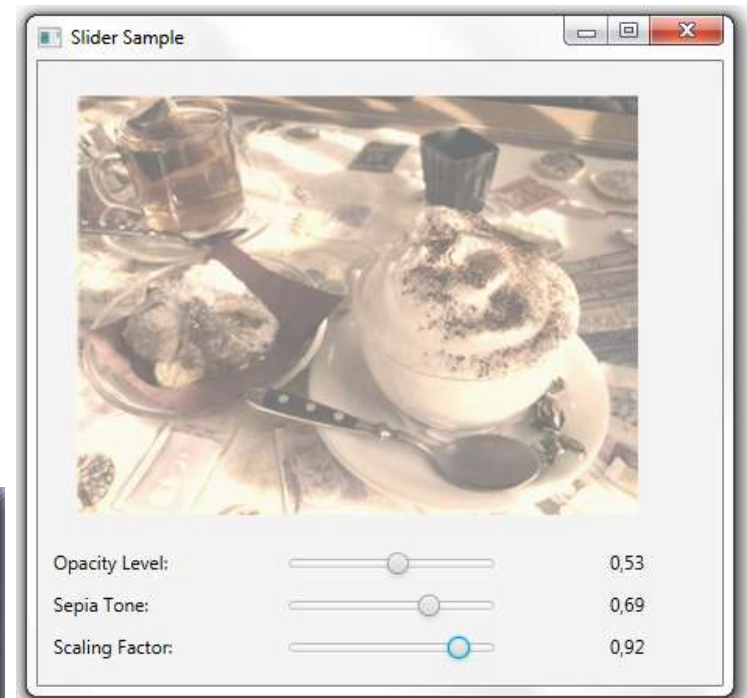
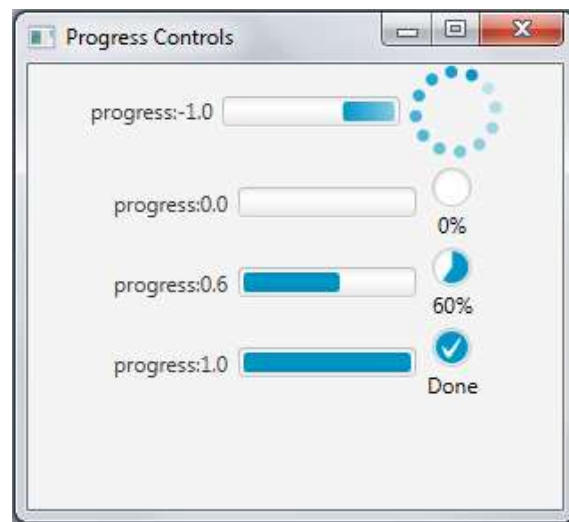
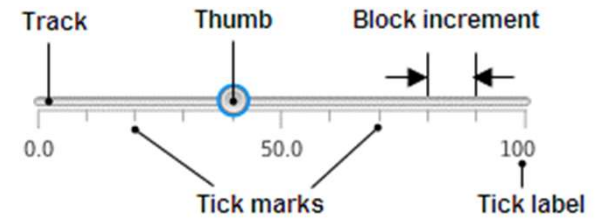
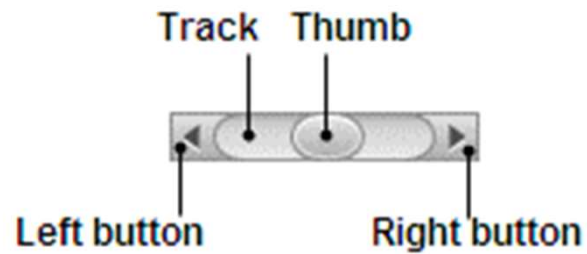


A window titled "Password Field Sample" with standard window controls. It contains a password input field with the label "Password". Below the field is a green confirmation message: "Your password has been confirmed".

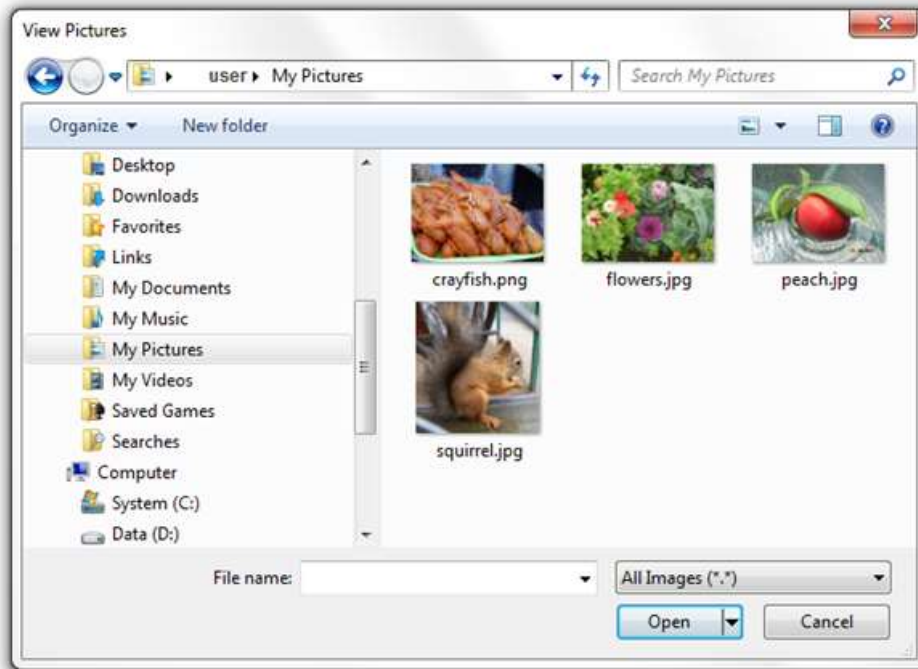


A window titled "First Pane" with standard window controls. It features a large text area containing 12 lines of text, each starting with "This is Line Number :". The lines are numbered 1 through 12. At the bottom right of the window is an "Add" button.

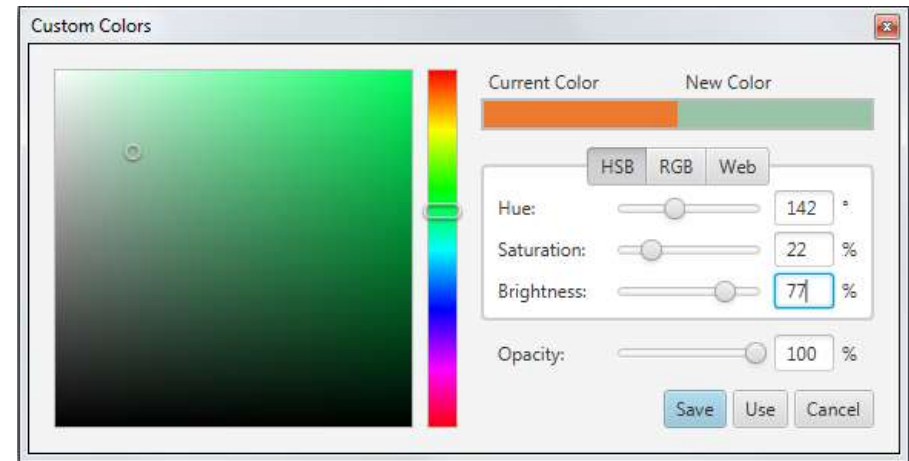
Number Widgets



Large (but still simple) widgets



File dialog



Colour picker

Other simple widgets

- http://en.wikipedia.org/wiki/GUI_widget

v · d · e	GUI widgets	[hide]
Command input	Button • Context menu • Menu • Pie menu	
Data input-output	Check box • Combo box • Drop-down list • Grid view • List box • Radio button • Scrollbar • Slider • Spinner • Text box	
Informational	Balloon help • Heads-up display • Icon • Infobar • Label • Loading screen • Progress bar • Splash screen • Status bar • Throbber • Toast • Tooltip	
Containers	Accordion • Disclosure widget • Menu bar • Panel • Ribbon • Tab • Toolbar • Window	
Navigational	Address bar • Breadcrumb • Hyperlink • Tree view	
Special windows	About box • Alert dialog box • Dialog box • File dialog • Inspector window • Modal window • Palette window	
Related concepts	Layout manager • Look and feel • Widget toolkit	

Interaction with simple widgets

- Interaction model is built into the widget
- The API specifies how you can interact
 - ...and so specifies what the user can *do*
- Where do the interaction models come from?
 - Are we stuck with what we've got?

Add-on widget toolkits

- ControlsFX
 - <http://fxexperience.com/controlsfx/features/>
- Droid UX
 - <http://droidux.com/android-ui-components-controls/>

Advanced widget interaction

- Tomer Moscovich
 - Contact Area Interaction with Sliding Widgets
 - <http://www.moscovich.net/tomer/papers/moscovich-sliding-uist09.wmv>
- Patrick Baudisch
 - Phosphor Widgets
 - <http://www.patrickbaudisch.com/publications/2006-Baudisch-UIST06-Phosphor.avi>
- Jared Cechanowicz
 - Augmented Interactions
 - http://hci.usask.ca/uploads/143-chiaugmentedwidgets_0004.wmv

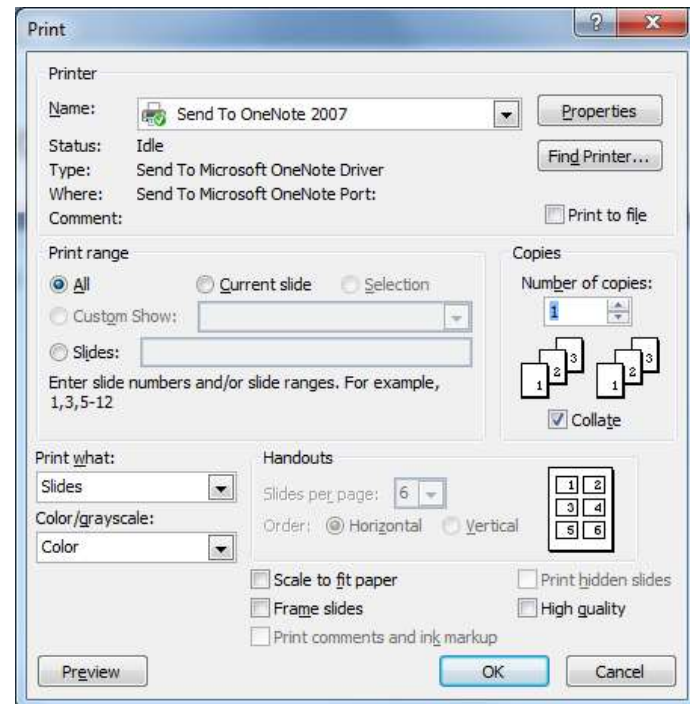


Containers

Container Widgets

- The widget's model contains a list of other widgets
- Generally do not generate events
- Can control layout for contained widgets
- Used both for top-level windows and for organization within windows

Top-level windows



Internal containers

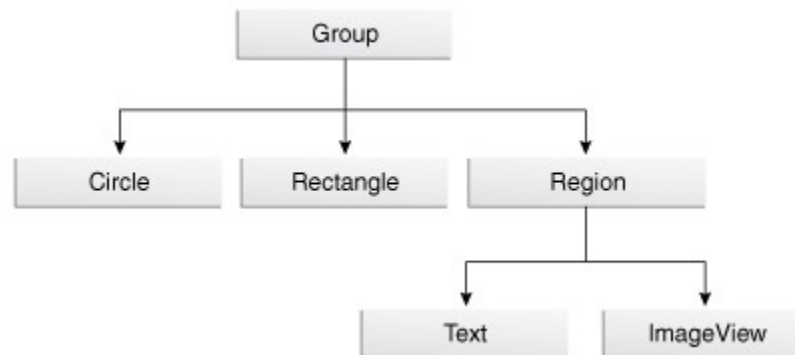


USB debugging connected

Example

one	two
three	four
five	six
seven	eight
nine	ten

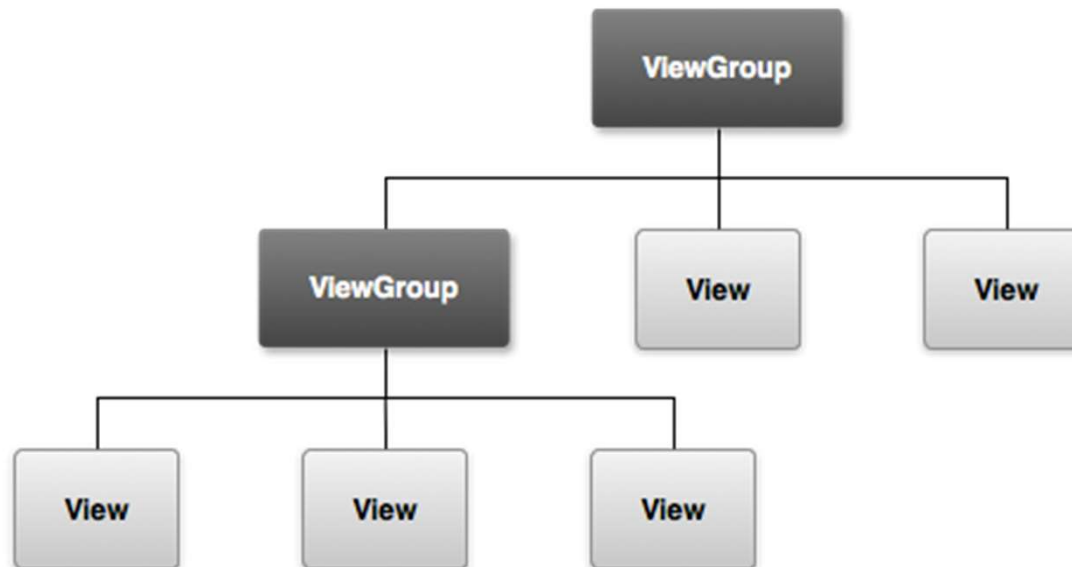
Containment hierarchy in JavaFX



The “scene graph”

<http://docs.oracle.com/javase/8/javafx/scene-graph-tutorial/scenegraph.htm>

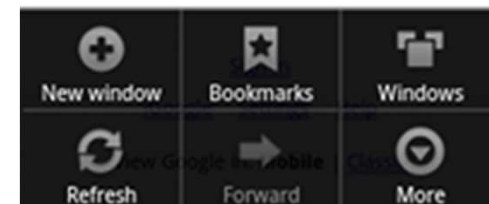
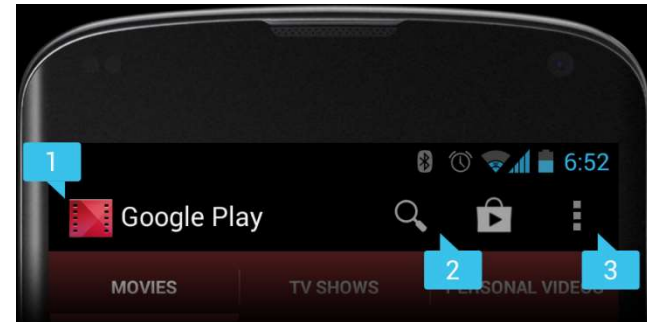
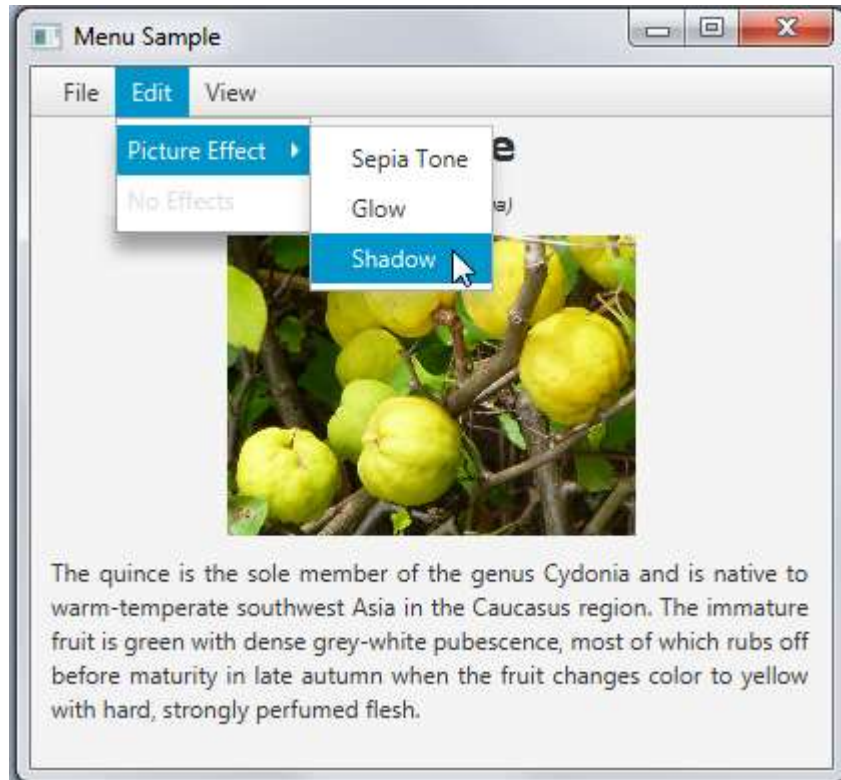
Containment hierarchy in Android



Containment hierarchy in Android

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android=http://schemas.android.com/apk/res/android
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
<TextView
    android:id="@+id/text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="I am a TextView" />
<Button android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="I am a Button" />
</LinearLayout>
```

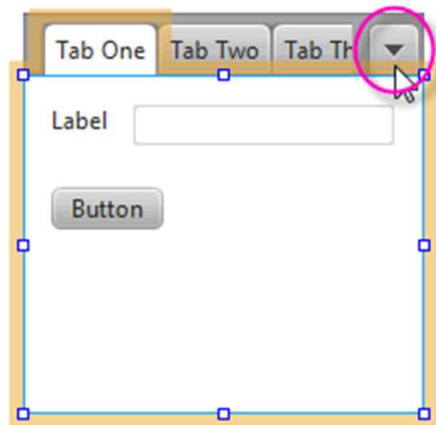
Menus



Menus

- ...and if you thought there was nothing more you could do with menus:
- Pourang Irani
 - Improving Selection in Cascading Menus
 - hci.cs.umanitoba.ca/assets/publication_files/2011-journal-erum-menu.pdf

Tabs



Advanced container widgets

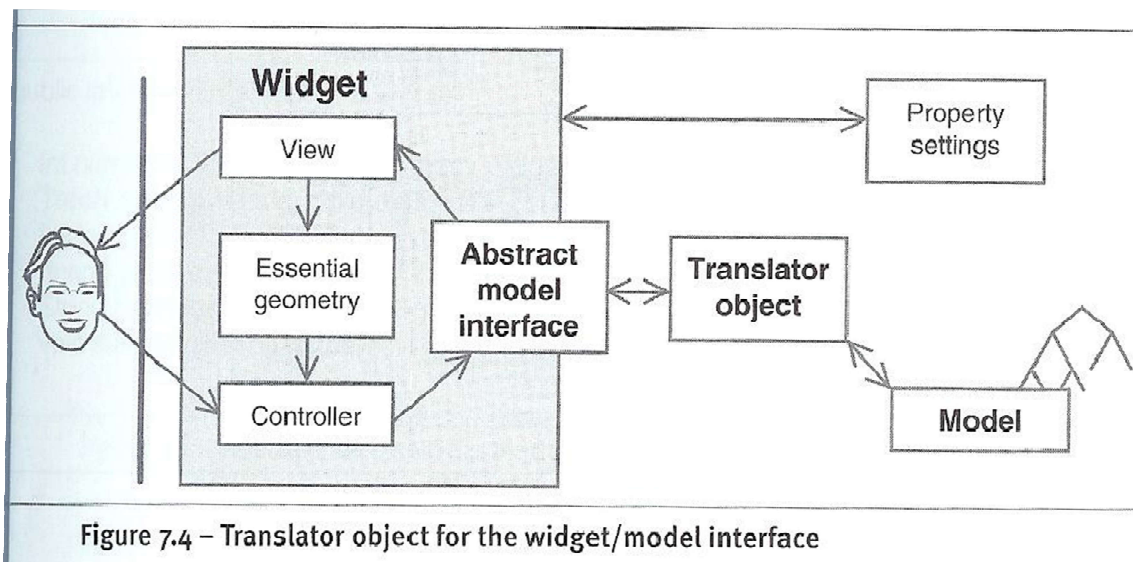
- Problem: what if the toolbar contains only half of the tools you need for a specific task?
 - How to support user customization of the UI?
 - Current customization mechanisms are tedious
- Wolfgang Stuerzlinger and Olivier Chapuis
 - User interface facades
 - <https://www.youtube.com/watch?v=kUUtiOWTzz4>
- Florian Block
 - VoodooSketch
 - <https://www.youtube.com/watch?v=9kXdQw23ubY>



Abstract Model Widgets

Abstract Model Widgets

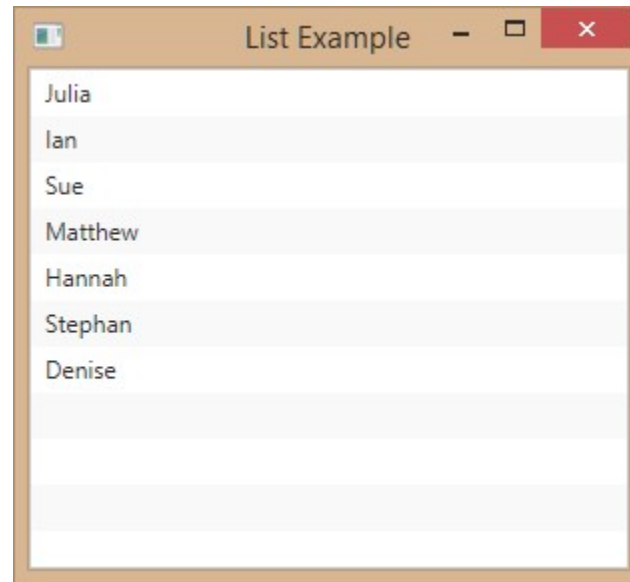
- More complex widgets require a detailed model of what is to be displayed
 - Interaction is still defined in the widget, but model can be manipulated and populated separately



More Complex Text/Web Widgets

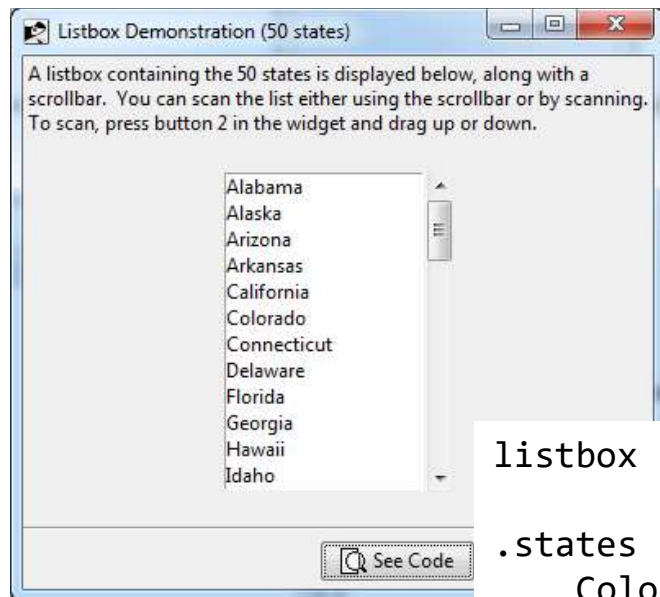


List widgets



```
ObservableList<String> names = FXCollections.observableArrayList(  
    "Julia", "Ian", "Sue", "Matthew", "Hannah", "Stephan", "Denise");  
ListView<String> listView = new ListView<String>(names);
```

...compare with Tcl/Tk:



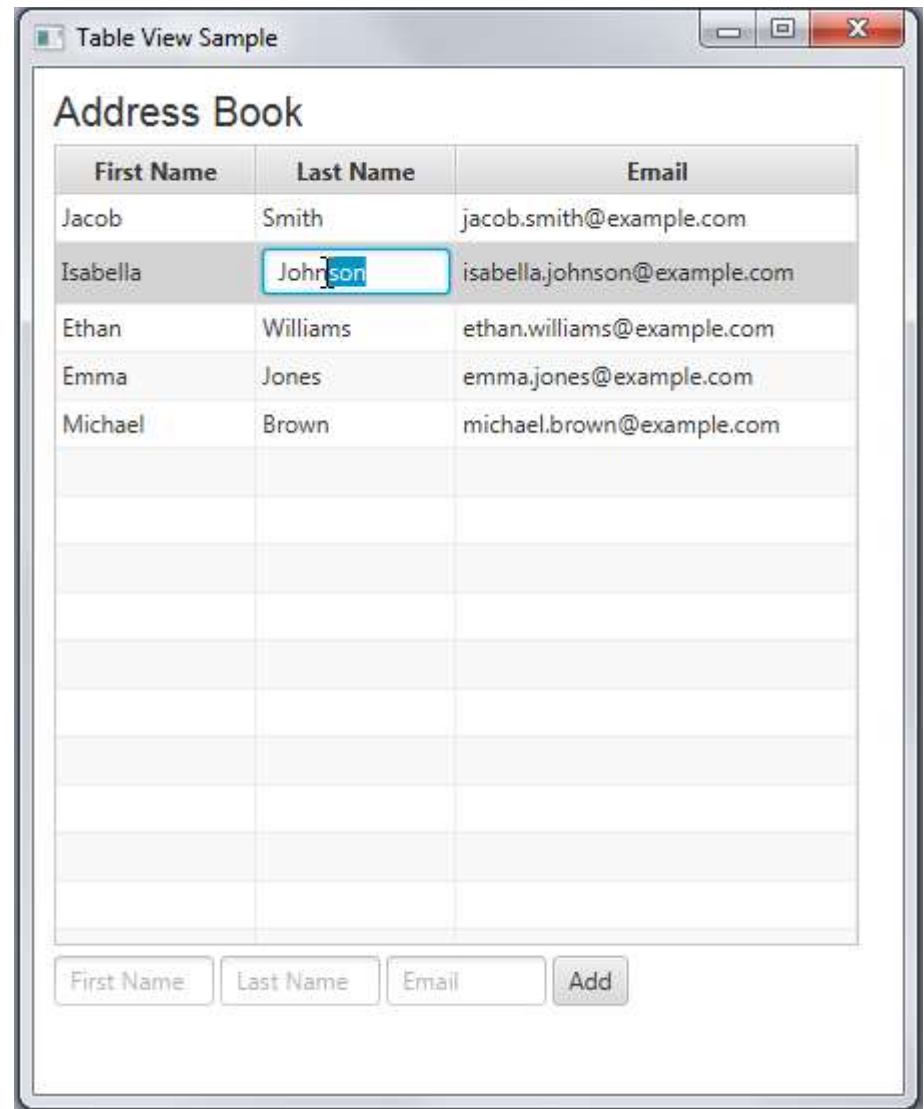
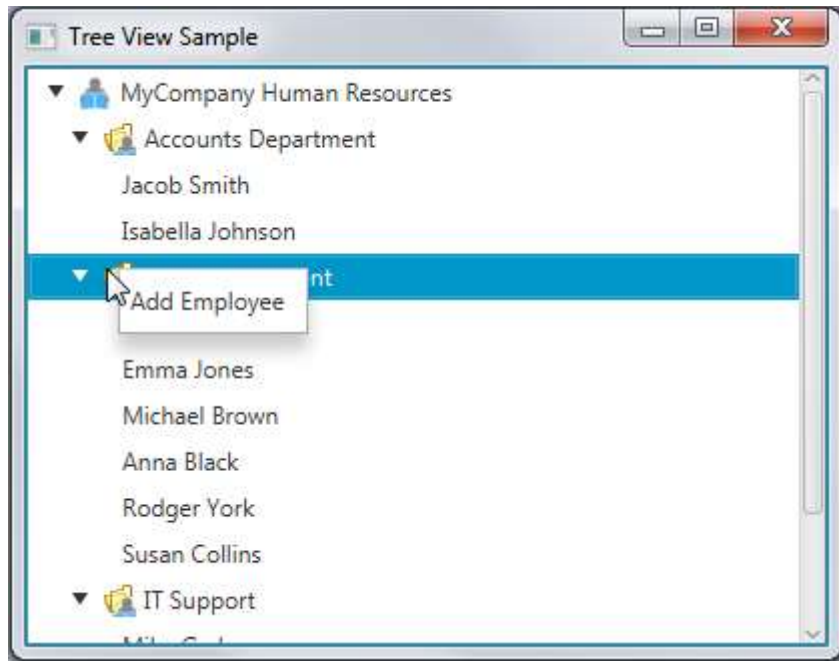
```
listbox .states -height 12
```

```
.states insert 0 Alabama Alaska Arizona Arkansas California \  
Colorado Connecticut Delaware Florida Georgia Hawaii Idaho \  
Indiana Iowa Kansas Kentucky Louisiana Maine Maryland \  
Massachusetts Michigan Minnesota Mississippi Missouri \  
Montana Nebraska Nevada "New Hampshire" "New Jersey" \  
"New York" "North Carolina" "North Dakota" \  
Ohio Oklahoma Oregon Pennsylvania "Rhode Island" \  
"South Carolina" "South Dakota" \  
Tennessee Texas Utah Vermont Virginia Washington \  
"West Virginia" Wisconsin Wyoming
```

Interaction with lists

- Carl Gutwin and Andy Cockburn
 - Improving List Revisitation with ListMaps
 - <http://hci.usask.ca/publications/2006/listmap.mov>

Trees and Tables



For more information:

- Chapter 7 in textbook

Next...

-
- Events: chapter 3 in the text