**CS 349 - User Interfaces** **Winter 2018**

Lecture 7: January 18, 2018

*Lecturer: Keiko Katsuragawa* *Notes By: Harsh Mistry*

## 7.1 Widgets

- Widget is a generic name for parts of an interface that have their own behavior: buttons, drop-down menus, spinners, file dialog boxes, progress bars, sliders, ...
- widgets also called components, or controls
- They provide user feedback and capture user input
- They have a defined appearance
- They send and receive events

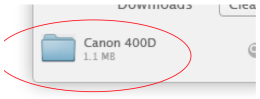
### 7.1.1 Logical Device

- A logical device is the essence of what a widget does. Its function
- E.g logical button device. The function is to generate a pushed event.
- A widget is a logical device with an appearance.

### 7.1.2 Categorizing and Characterizing Widgets

- Logical Device (Button, number, text, choice, etc)
- Event the widget generates (action, change, etc)
- Properties to change behaviour and appearance (colour, size, icon, allowable values)

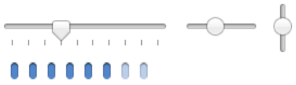
### 7.1.3 Simple Widgets



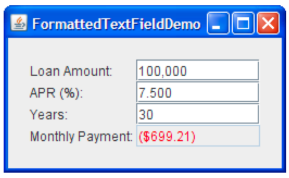

- Labels and Images
  - (usually) no model or events
  - e.g. label, icon, spacer,


- Button
  - no model, pushed event
  - properties: label, size
  - e.g. button

- Boolean
  - true/false model, changed event
  - e.g. radio button, checkbox, toggle button

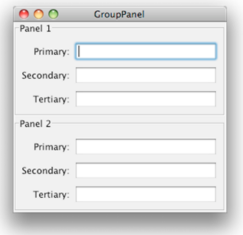


- Number
  - model: real number, changed event
  - properties: range, step
  - e.g. slider, progress bar, scrollbar

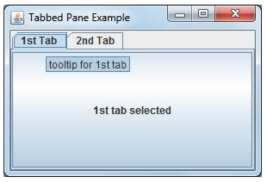



- Text
  - model: string; changed, selection, insertion events
  - properties: formatters (numeric, phone number, ...)

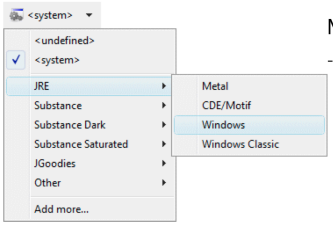
### 7.1.4 Container Widgets



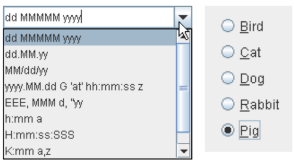

- Panel (Pane, Form, Toolbar)
  - arrangement of widgets
  - e.g. JPanel, toolbar


- Tab
  - choice between arrangements of widgets




Menu
- hierarchical list of (usually) buttons

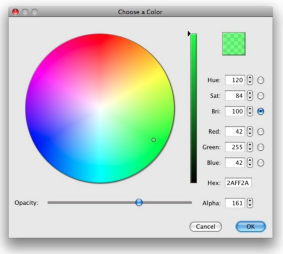


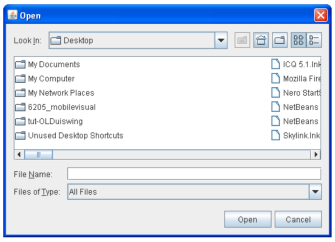




- Choice from a List
  - list of boolean widgets
  - e.g. drop-down, combo-box, radio button group, split button

### 7.1.5 Special Value Widgets




### 7.1.6 Widget toolkits

- Also called widget libraries or GUI toolkits or GUI APIs
- Software bundled with a window manager, operating system, development language, hardware platform
- Defines a set of GUI components for programmers
- Examples: buttons, drop-down menus, sliders, progress bars, lists, scrollbars, tab panes, file selection dialogs, etc.
- Programmers access these GUI components via an application programming interface (API)

### 7.1.7 Event-driven programming

- Widget toolkits use event-driven programming model
- Reactive systems
  - User Action → program response
  - Most of the time the program sits around doing nothing
- Widget toolkit supports a mechanism for mapping user action on widget to appropriate application code to handle that action

**Widget Toolkit Design Goals:**

- Complete
  - GUI designers have everything they need
- Consistent
  - Behaviour is consistent across components
- Customizable
  - Developer can reasonably extend functionality to meet particular needs of application

- Meeting these requirements encourages reuse

### 7.1.8 Completeness

- All you really need are
  - Button
  - Slider
  - Pulldown menu
  - Check box
  - Radio button
  - Text field

## 7.2 Heavyweight Widgets

- OS provides widgets and hierarchical "windowing" system
- Widget toolkit wraps OS widgets for programming language
- BWS can dispatch events to a specific widget
- Examples: nested X Windows, Java's AWT, OSX Cocoa, standard HTML form widgets, Windows MFC

Advantages

- Events generated by user are passed directly to components by BWS/OS
- Preserves OS look and feel

Disadvantages

- OS-specific programming
- Multi-platform toolkits tend to be defined as the "lowest-common set" of components

## 7.3 Lightweight Widgets

- OS provides a top level window
- Widget toolkit draws its own widgets in the window.
- Toolkit is responsible for mapping events to their corresponding widgets
- Examples: Java Swing, JQuery UI, Windows WPF

Advantages

- Can guarantee identical look-and-feel across platforms.
- Can guarantee consistent widget set on all platforms.
  (see SwingThemeDemo.java lecture code)
- Can implement very light/optimized widgets.

Disadvantages

- Concerns that they appear "non-native".
- Concerns about performance with extra layer of abstraction.