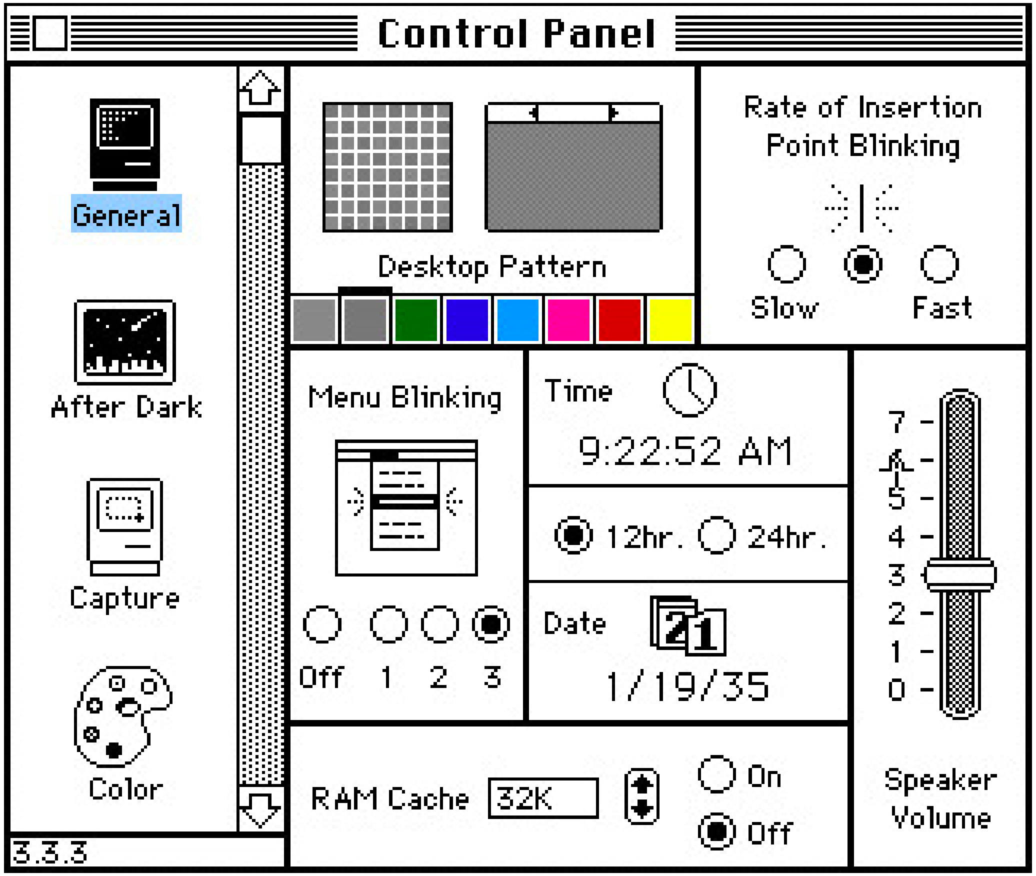
Widgets

- Abstract Widget
- Widget Types
- Widget Implementation

User Interface Widget

- A generic name for parts of an interface with their own behaviour
  - also called **components** or **controls** or **elements** or **views** …



Control Panel

General

After Dark

Capture

Color

3.3.3

Desktop Pattern

Rate of Insertion Point Blinking

○ ● ○

Slow Fast

Menu Blinking

○ Off ○ 1 ○ 2 ● 3

Time

9:22:52 AM

● 12hr. ○ 24hr.

Date

1/19/35

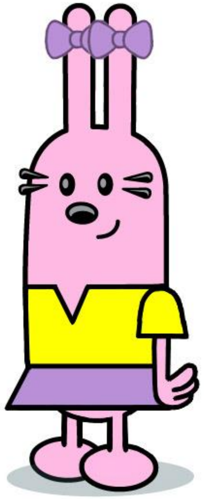
RAM Cache 32K

○ On

● Off

Speaker Volume

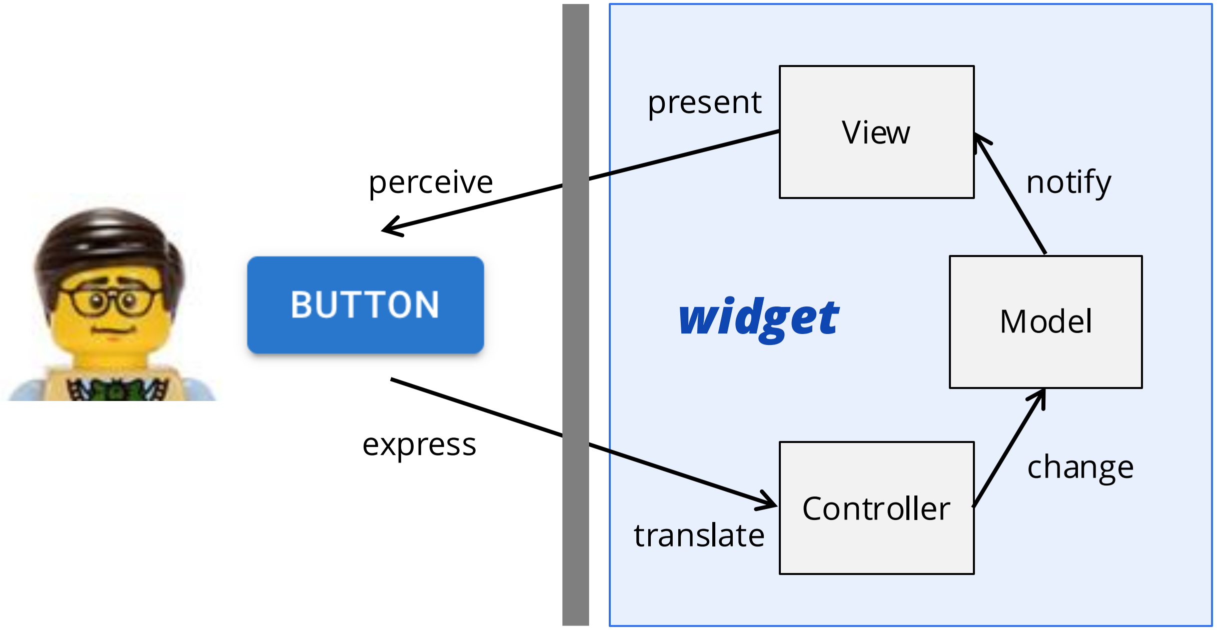
7 - 6 - 5 - 4 - 3 - 2 - 1 - 0 -



https://aleclikesmacintosh.tumblr.com/

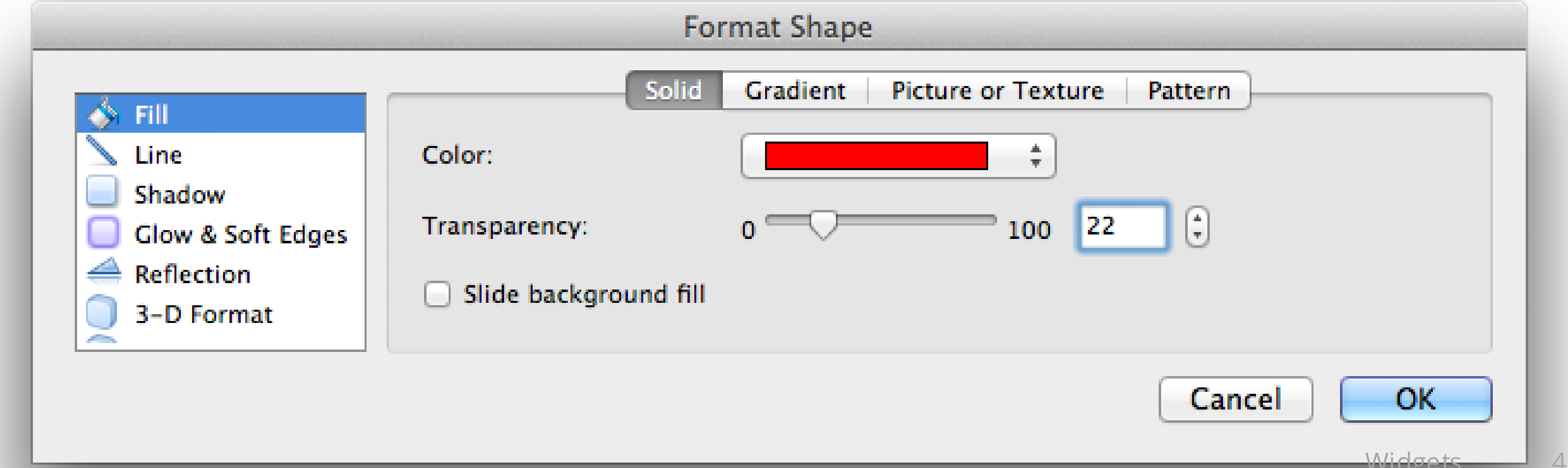
Widget Functions

1. **Display information** like a MVC View
   - Convey current state, e.g. of data, of action
   - e.g. numeric value, cursor is over button, button was clicked
2. **Handle user input** like a MVC Controller
   - Capture user input and generate events
   - e.g. detect when button was clicked on, send ”action” event



Characterizing Widgets

- What essential function does it enable?
  - display only? trigger a command? pick a colour?
- Can it contain other widgets?
  - simple widget vs. container widget
- What is its behaviour?
  - the events it generates, how it changes due to state
- What does it look like?
  - look & feel and rendering options



Color: 🟥

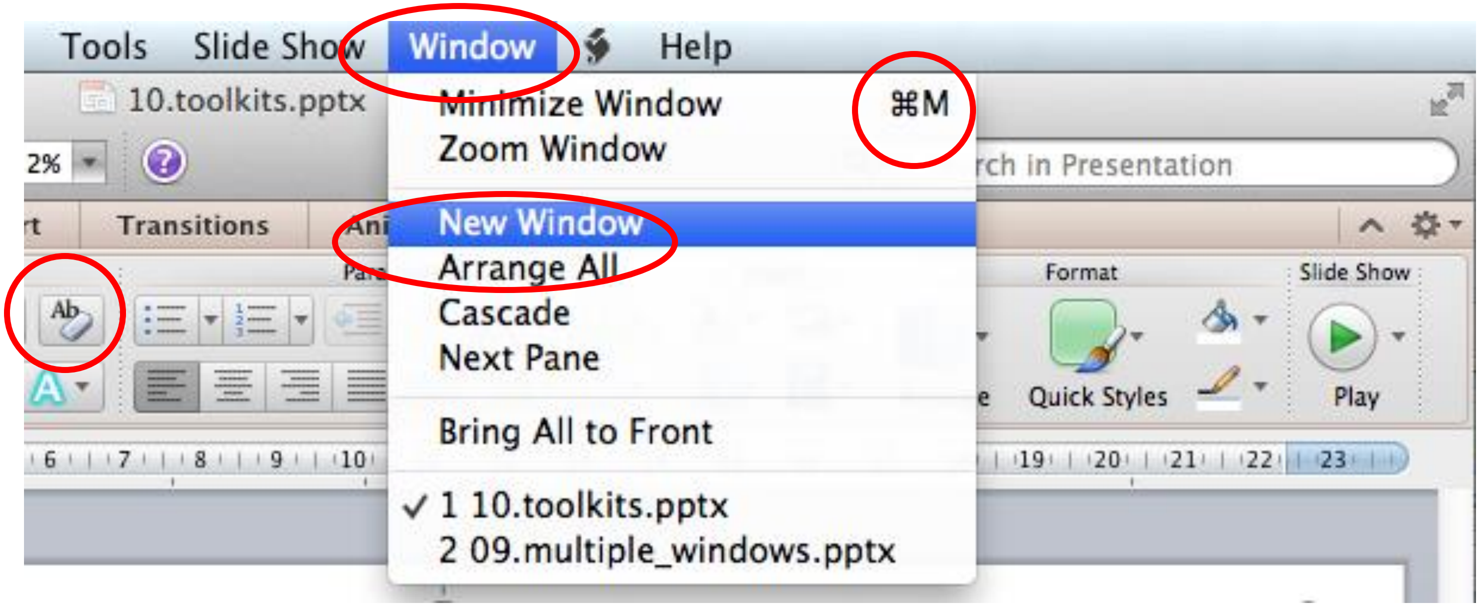
Transparency: 0 100 22

□ Slide background fill

Abstract Widgets

Describes a widget based on **function** (not appearance)

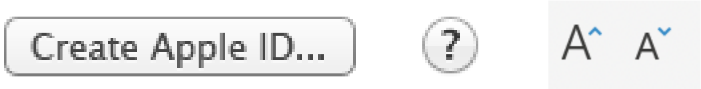
- a *category* of widgets based on common *functionality*
  - e.g. a *action abstract widget* generates and “action” event
- A **widget** is a *specific instance* of an abstract widget
  - e.g. a *Button* widget is an instance of a *action abstract widget*



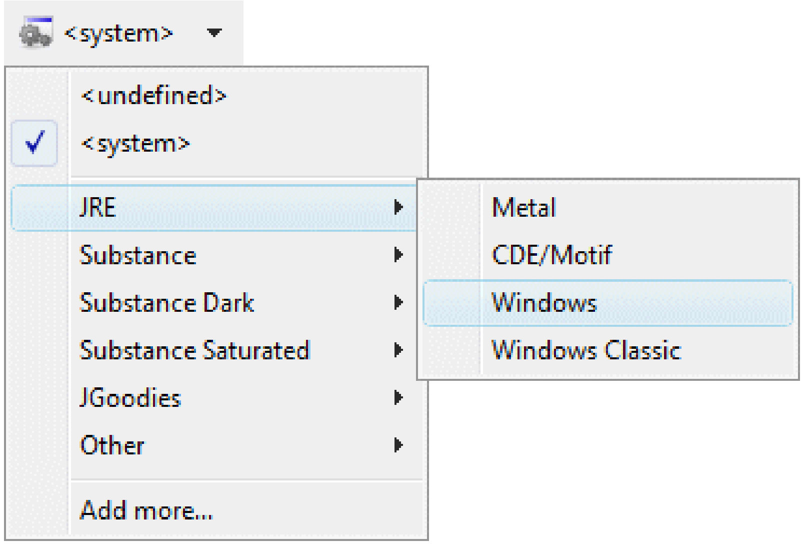
Action Abstract Widget

- trigger a single action
  - sends ”execute action" event

- Example widget instances:



Button



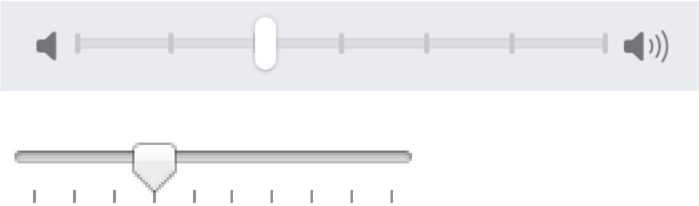
Menu

hierarchical list of buttons

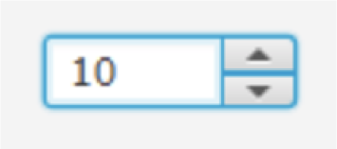
Number Abstract Widget

- Input and display a numeric value
  - usually in certain range
  - may support validation
  - sends "value changed" event

- Example widget instances:



Slider



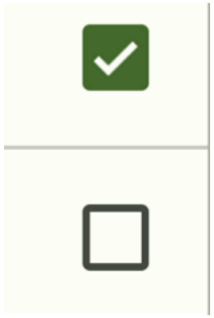
10

Spinner

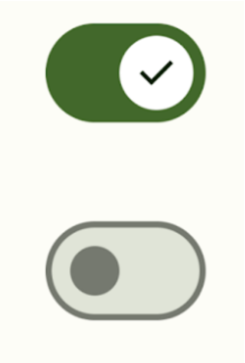
Boolean Abstract Widget

- Input and display a Boolean value
  - sends "value changed" event

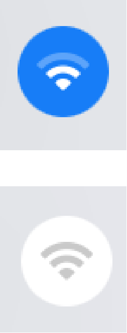
- Example widget instances:



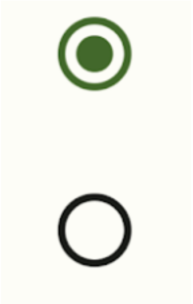
Checkbox



Toggle



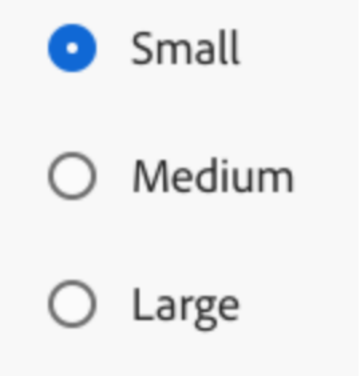
Toggle Button



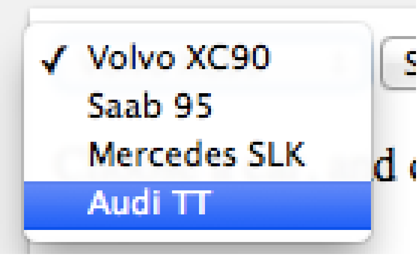
Radio Button

Choice Abstract Widget

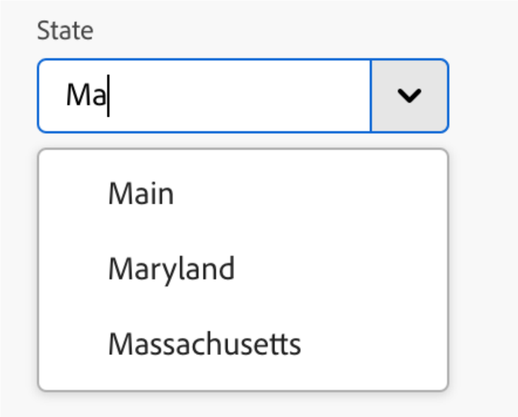
- Input and display one choice among a list
  - sends "selection changed" event



Radio Button Group
fixed set of options, must choose one



Dropdown
fixed set of options, must choose one



Combo Box
choose from set of options
text box to add new options

Why “Radio Button”

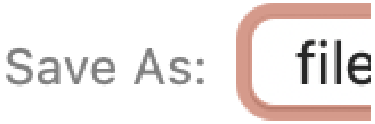
Only one button can be pressed at any time

○ ○ ◉ ○ ○

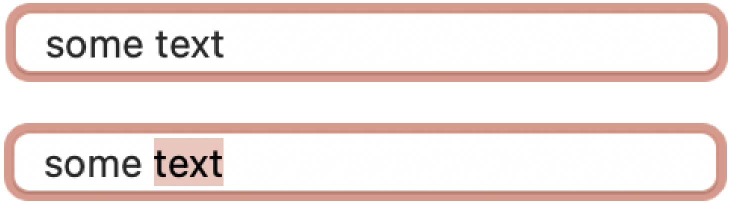
Text Abstract Widget

- Display a text value
- Edit a text value
  - sends "value changed" event

- Example widget instances:



Save As: file



some text

some text

Label
Display only

TextField
Single line of text
can be disabled
may use a formatter (numeric, phone number, ...)

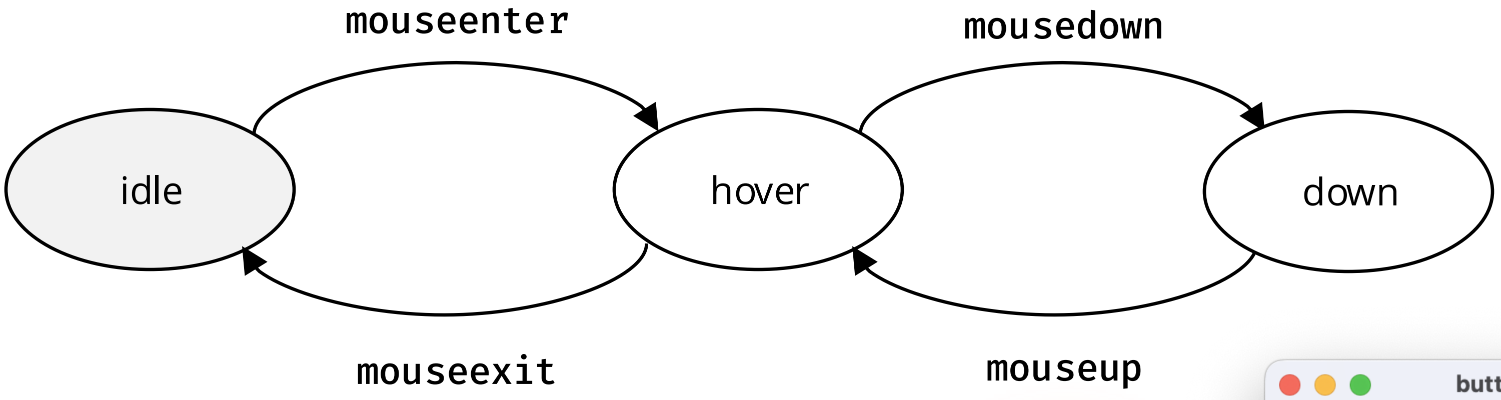
element / element.ts

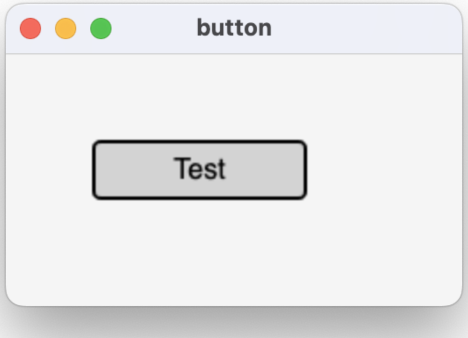
- The base widget class
  - location and size
  - Inside rectangle hitTest()
  - abstract draw()
- Using “props” method for constructor arguments
  - flexible options with named argument
  - (see next slide and “props” demo)

NOTE: The demos in this lecture are ***simplified*** versions of the widgets in SimpleKit

button

- Button state and drawing
- Uses standard widget Style object (in style.ts module)
  - standard dimensions, font, colours, etc.
  - DEMO: change the style
- main.ts simulates dispatch and focus events

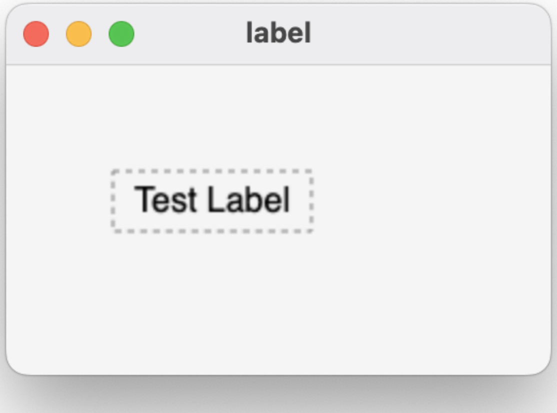




- How would you implement a `disabled` state?

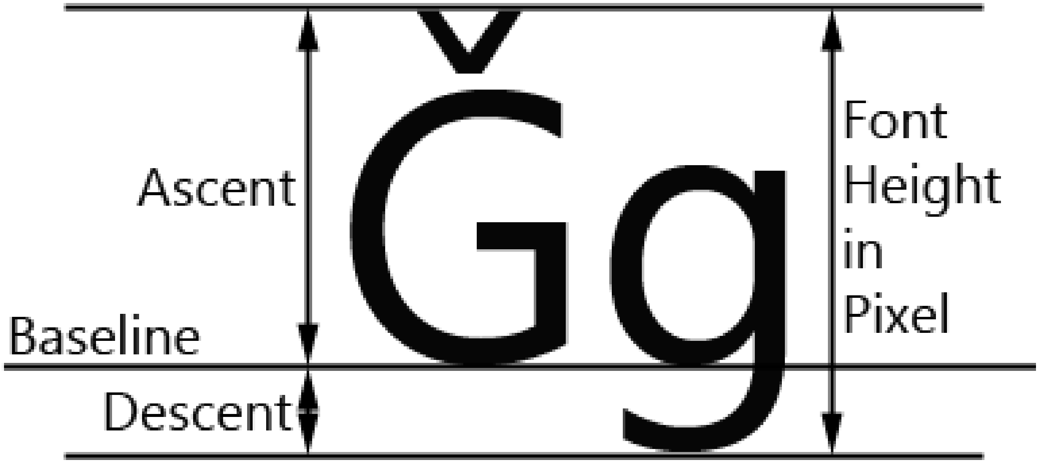
label

- Simple text display widget
- MeasureText utility function in SimpleKit
  - (see next slide)
- Alignment (applies when width greater than text)
- debug flag to see bounds



simplekit/utility/measuretext.ts

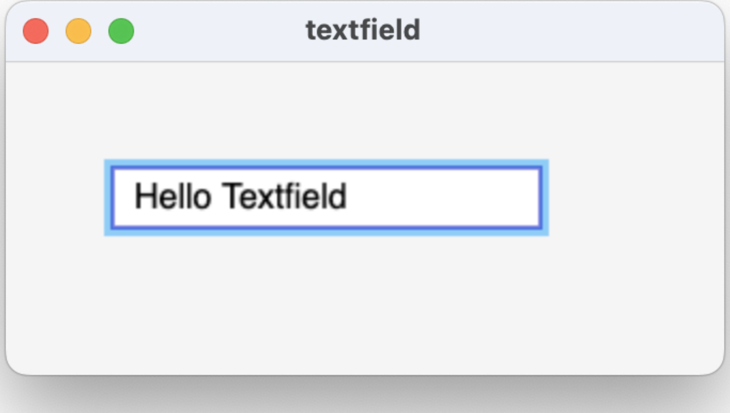
- Uses gc.measureText()
- Measures rendered text using a canvas buffer
- Caches canvas buffer
- height is ascent + descent



https://developer.mozilla.org/en-US/docs/Web/API/TextMetrics

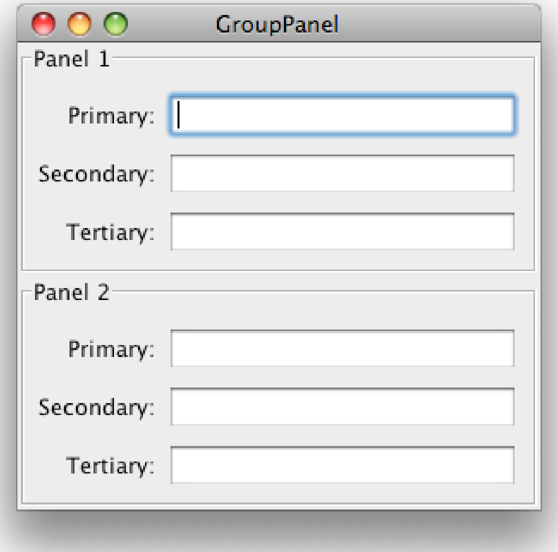
textfield

- Short text entry widget
  - focus state
  - text property get/set
  - textWidth and method to draw cursor when focused
  - applyEdit method
- testing in main.ts (simulate dispatch and focus events)
- TODO:
  - insert text in middle of string
  - support cursor keys
  - highlight text to replace or delete



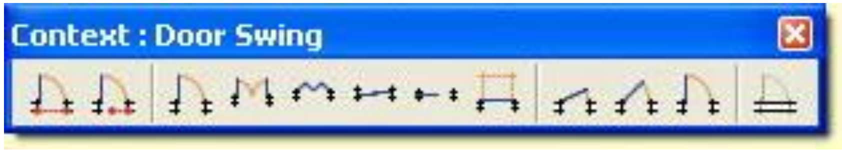
Hello Textfield

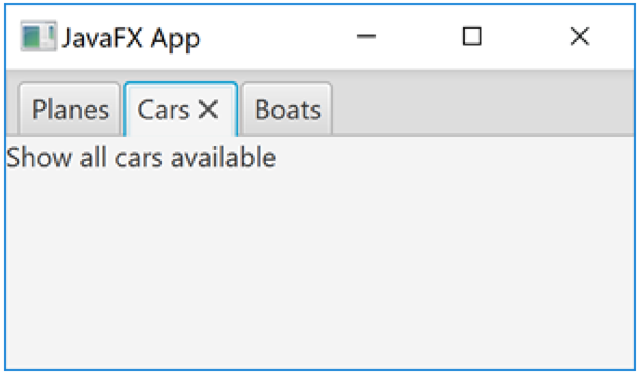
Container Abstract Widget



Panel

- set of widgets in layout



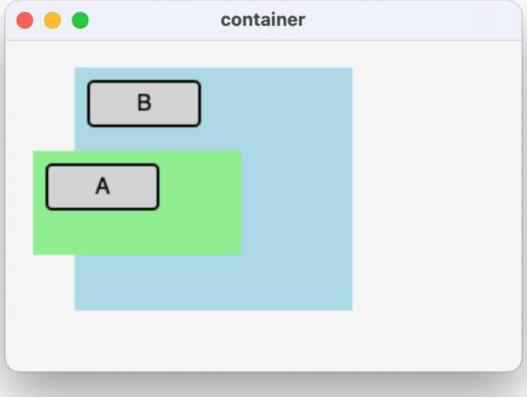


Tabs

- choice between set of widget

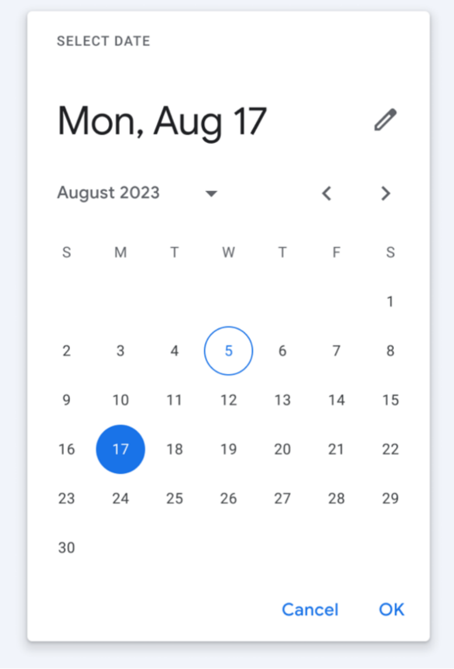
container

- `SKContainer` to hold “child” widgets (i.e. SKElements)
  - children are drawn *relative* to container position
  - containers create **widget tree**
- Demos
  - Add green container to blue container children
  - Move green container so it overlaps button B
  - Change order the green container and button B are added



Special Value Widgets

- date, colour, file, etc. pickers

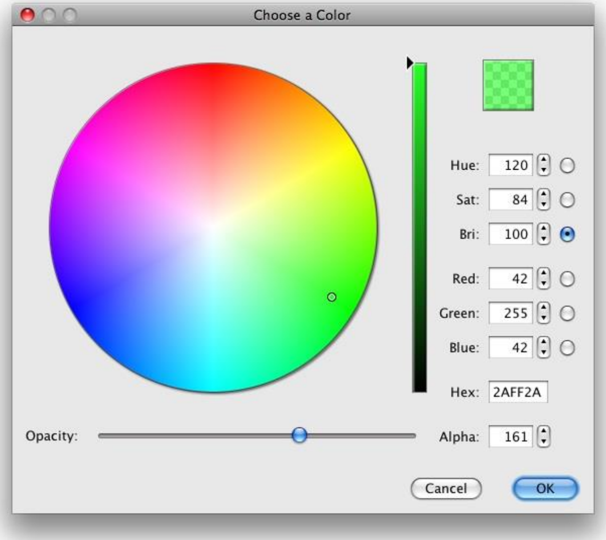


SELECT DATE

Mon, Aug 17

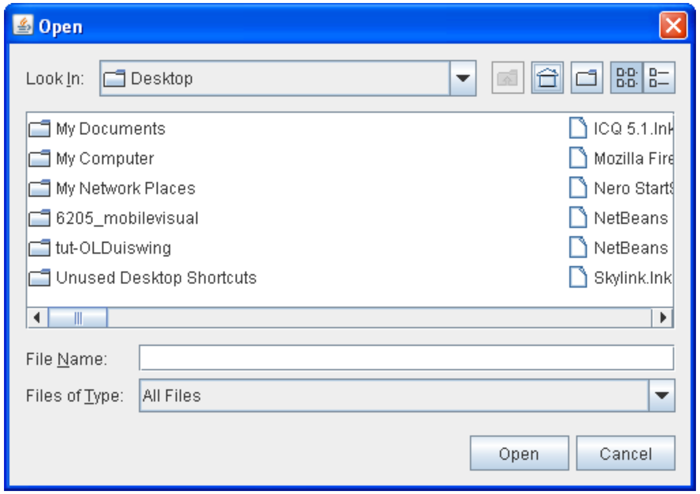
August 2023

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						



Choose a Color

Hue: 120 ○
Sat: 84 ○
Bri: 100 ●
Red: 42 ○
Green: 255 ○
Blue: 42 ○
Hex: 2AFF2A
Opacity:
Alpha: 161



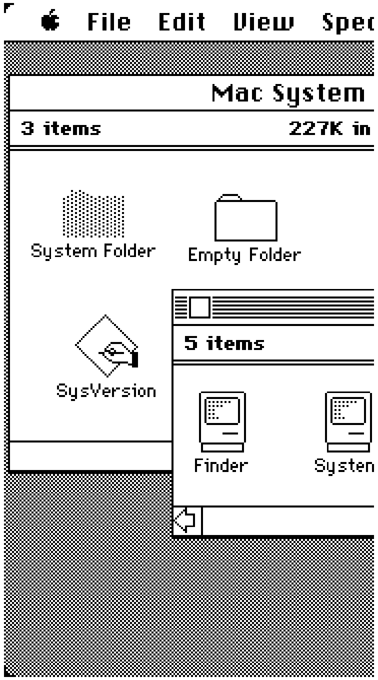
Open

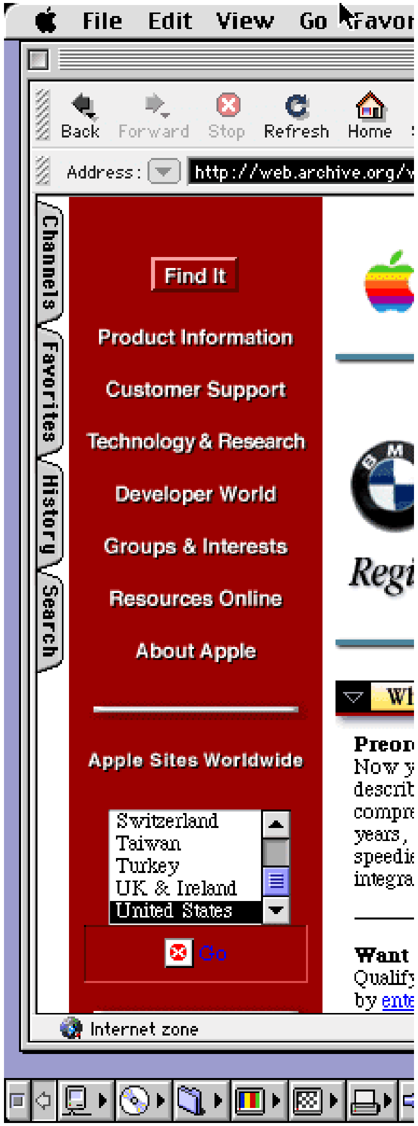
Look In: Desktop

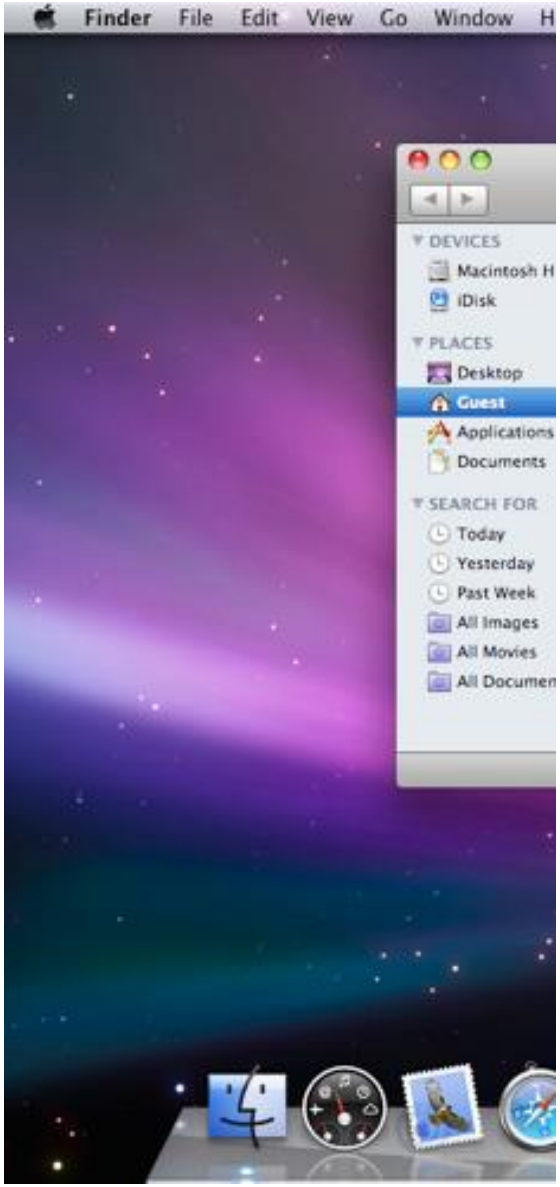
My Documents
My Computer
My Network Places
6205_mobilevisual
tut-OLDuiswing
Unused Desktop Shortcuts
ICQ 5.1.lnk
Mozilla Fire
Nero Start
NetBeans
NetBeans
Skylink.lnk

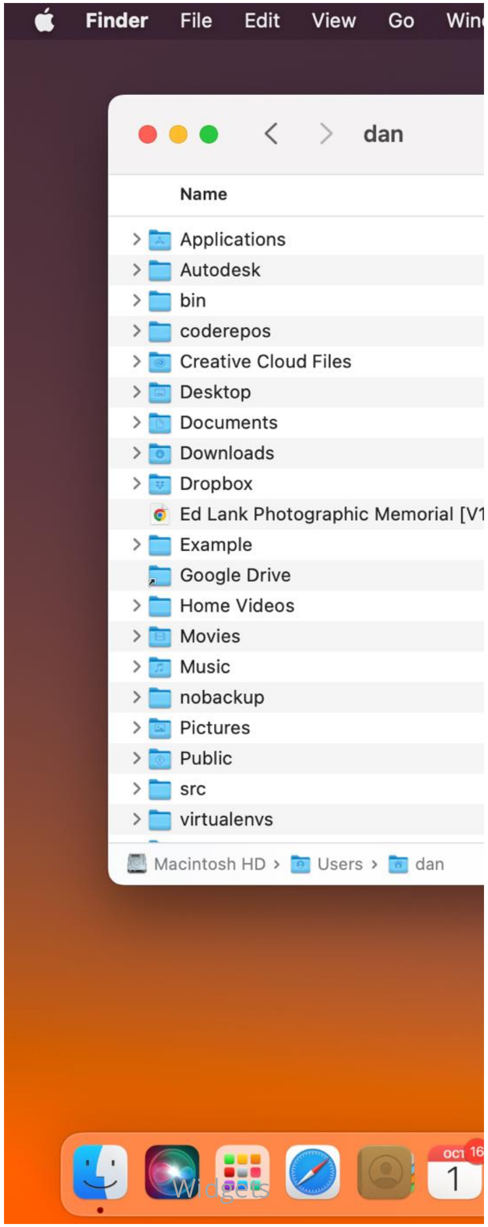
File Name:
Files of Type: All Files

UI Look and Feel









Customization

Support changing some widget behaviour and appearance

- Expose key appearance or behaviour properties
  - e.g. colours, font, orientation, …
- Factor out key behaviour
  - LayoutManagers
  - Formatters
- Expose class hierarchy that promotes customization

  ```
  Button extends ActionWidget
  ```

  - e.g. a base class implements an action abstract widget

Exercise

EXERCISE

Create a SKCheckbox widget

- It’s a boolean abstract widget
- Add files `element.ts` and `style.ts` to your project (without modification)
- Follow general structure of `button.ts` to define a new SKCheckbox widget class
- Default size for your checkbox should be `Style.widgetHeight - 8`
- Test it using SimpleKit canvas-mode with a SKEventListener similar to the button demo