

User Interfaces

Lecture 17

Cocoa: Windows & Views

Hamza Bennani
hamza@hamzabennani.com

September 11, 2018



Windows

- ▶ In the AppKit Framework, all windows are `NSWindow` (or subclasses)
- ▶ A window is associated with a corresponding XIB/NIB file
 - ▶ Recall: the XIB/NIB file stores all the graphical elements contained in the window
 - ▶ MacOS will set up lots of GUI state for you

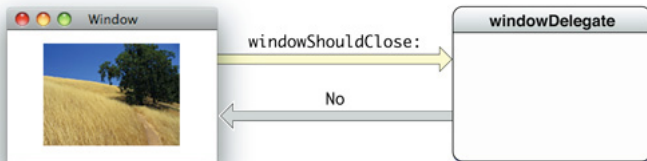
NSWindow

Methods for window?

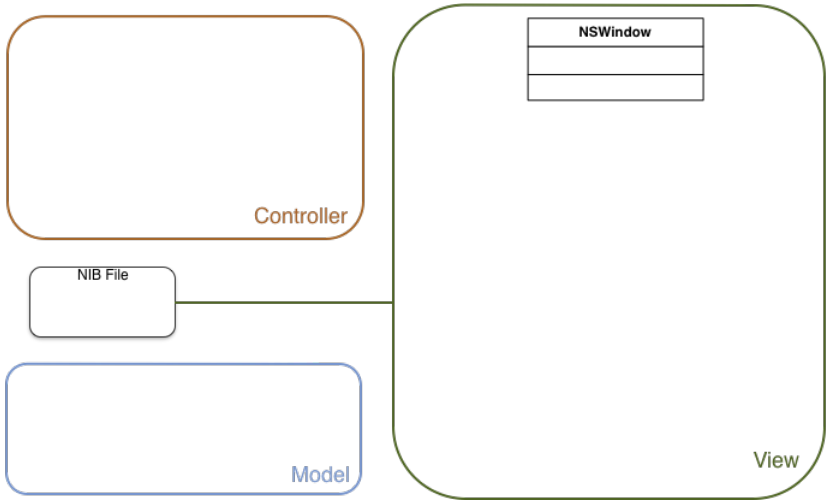
- ▶ Configuration
- ▶ Sizing
- ▶ Moving
- ▶ Closing
- ▶ Minimising
- ▶ Drawing
- ▶ Handling events
- ▶ ...

Window Delegate

- ▶ Window object has a delegate
- ▶ You can set a delegate with `setDelegate:`
 - ▶ Delegate has to conform to `NSWindowDelegate` protocol
 - ▶ The protocol has an array of methods that are invoked when events occur such as: **`windowDidResize:`**, **`windowWillResize:`**, **`windowDidBecomeKey:`**, etc.



Application (Window)



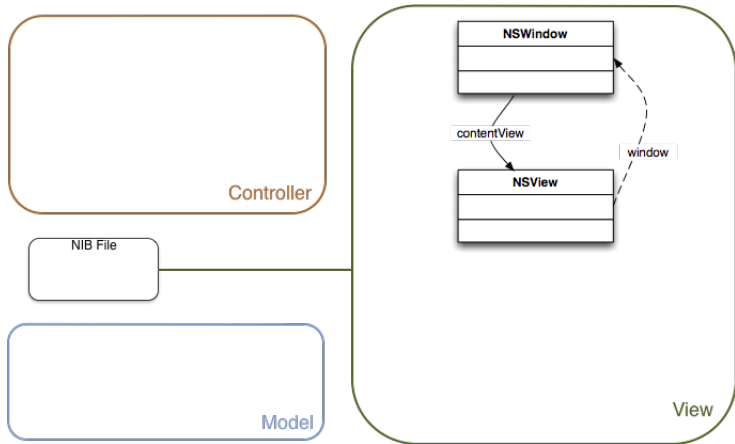
Views

- ▶ Within each **NSWindow** object there are multiple **NSView** objects
 - ▶ **NSControl** objects are subclassed from **NSView**
 - ▶ Examples are **NSButton**, **NSSlider**, **NSTextField**, etc.
- ▶ We are **not** talking here about the View from the MVC design pattern!
 - ▶ These views are types of objects placed in an application window

Views

- ▶ A view is a structure for drawing, printing and handling events in a window
- ▶ Everything inside a window is a view:
 - ▶ Buttons, sliders, text-boxes, images,? are all views
- ▶ A window gets created with a content view? a view spanning the entire window frame
- ▶ Usually, you would use Interface Builder to make other views . . . but can be done at runtime

Application (window, content view)



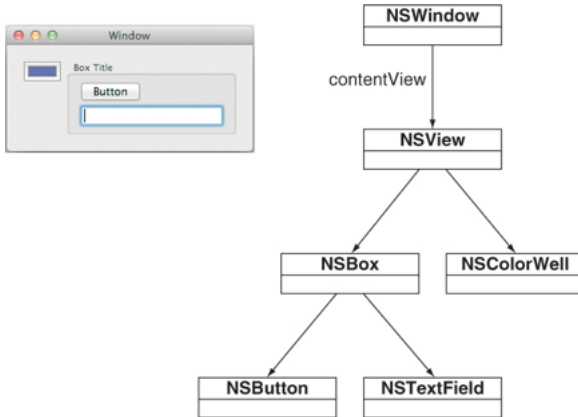
NSView

Methods are provided in **NSView** for?

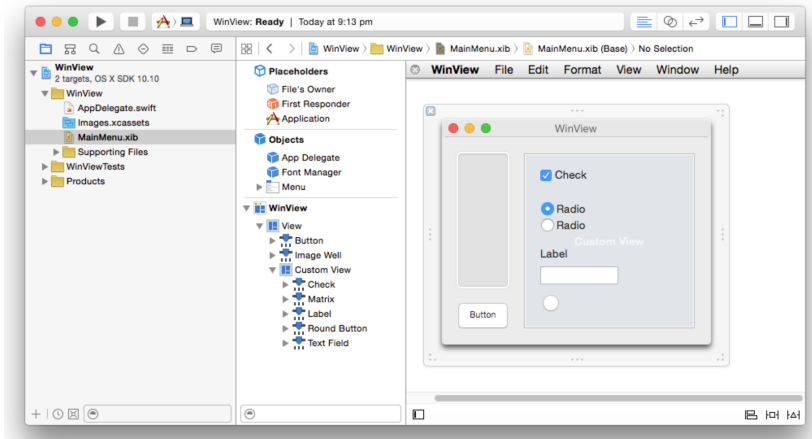
- ▶ Drawing
- ▶ Coordinates handling
- ▶ Bounds handling
- ▶ Hiding
- ▶ Scrolling
- ▶ Event handling
- ▶ ...

The View hierarchy

A window has a content view and the content view has subviews.

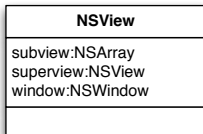


The view hierarchy

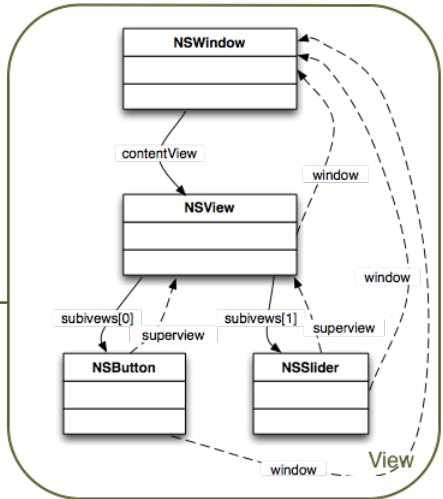
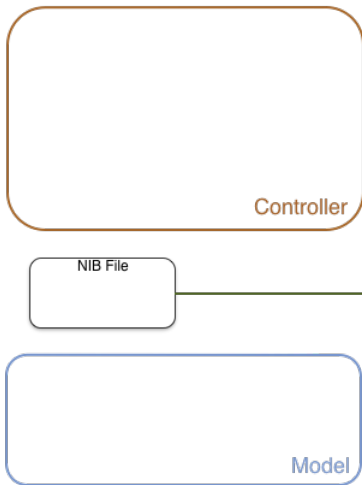


The view hierarchy

- ▶ Every view knows its superview, its subviews, and its window (NB: view hierarchy is not a class inheritance hierarchy!)
- ▶ Most ready-made views have no subviews, except:
 - ▶ **Content of window**
 - ▶ **NSBox** - a box around other views
 - ▶ **NSScrollView** - scroll bars
 - ▶ **NSSplitView** - adjacent views
 - ▶ **NSTabView** - tabs that swap between views



Application (window, view hierarchy)



Custom Views

- ▶ To make a custom view, you subclass `NSView` (within Interface Builder: a "Custom View")
 - ▶ Override `drawRect:` and potentially the constructors
- ▶ Create a new class that extends `NSView`
- ▶ In Interface Builder position a Custom View object in appropriate location in your window
- ▶ Set the class name of the Custom View object to the name of your class
 - ▶ Thus corresponding custom view object to your view
- ▶ Your class: add / call / override `NSView` methods

View Geometry

A view has a **frame** and **bounds**. How you specify the frame and bounds will affect how the view is scaled.

Superview



Frame rectangle at (5.0, 5.0), size (73.0, 88.0)



Bounds rectangle at (0.0, 0.0), size (73.0, 88.0)

Superview



Frame rectangle at (5.0, 5.0), size (146.0, 88.0)



Bounds rectangle at (0.0, 0.0), size (73.0, 88.0)

View Geometry

Views can be rotated. . .

Superview



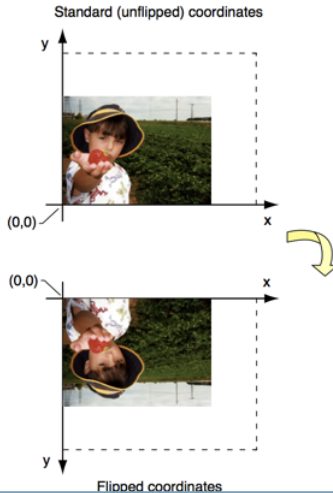
Frame rectangle at (10.0, 10.0), size (108.0, 75.0)



Bounds rectangle at (0.0, 40.0), size (108.0, 75.0)
Bounds rotation 20 degrees

View Geometry

... and flipped



Summary

- ▶ Window
- ▶ NSWindow
- ▶ Window Delegate
- ▶ Views, content view, custom views
- ▶ NSView
- ▶ View hierarchy
- ▶ View geometry