

The background features a light blue gradient with a pattern of binary code (0s and 1s) and a diagonal film strip. The film strip contains several frames: the top frame shows a hand holding a smartphone; the middle frame shows a person sitting at a desk with a laptop; the bottom frame shows a person sitting at a desk with a laptop, looking at the screen.

iPhone Application Programming

Lecture 4: User Interface Design

*Chat Wacharamanotham
Media Computing Group
RWTH Aachen University*

Winter Semester 2013/2014

<http://hci.rwth-aachen.de/iphone>

View Concepts

- SDK provide many types of Views to show your content
- At run-time Views are organized as a tree
- Use Interface Builder to design your UI and connect it to code
- Geometry of Views are determined by constraints



Cancel

Add Event

Done

Title

Location

All-day



Starts **November 12, 2013 7:00 AM**

More dates...
Sat Nov 9

4

57

Sun Nov 10

5

58

Mon Nov 11

6

59

Today

7

00

AM

Wed Nov 13

8

01

PM

Thu Nov 14

9

02

More dates...
Fri Nov 15

10

03

Carrier 

7:31 AM



Cancel

Add Event

Done

Title

Location

All-day



Starts **November 12, 2013** **7:00 AM**

Time zone: Pacific Standard Time

Sat Nov 9	4	57	
Sun Nov 10	5	58	
Mon Nov 11	6	59	
Today	7	00	AM
Wed Nov 13	8	01	PM
Thu Nov 14	9	02	
Fri Nov 15	10	03	

Finding the Right View

- ▼ Bars
 - The Status Bar
 - Navigation Bar
 - Toolbar
 - Toolbar and Navigation Bar Buttons
 - Tab Bar
 - Tab Bar Icons
 - Search Bar
 - Scope Bar
- ▼ Content Views
 - Activity
 - Activity View Controller
 - Collection View
 - Container View Controller
 - Image View
 - Map View
 - Page View Controller
 - Popover (iPad Only)
 - Scroll View
 - Split View Controller (iPad Only)
 - Table View
 - Text View
 - Web View
- ▼ Controls
 - Activity Indicator
 - Contact Add Button
 - Date Picker
 - Detail Disclosure Button
 - Info Button
 - Label
 - Network Activity Indicator
 - Page Control
 - Picker
 - Progress View
 - Refresh Control
 - Rounded Rectangle Button
 - Segmented Control
 - Slider
 - Stepper
 - Switch
 - System Button
 - Text Field
- ▼ Temporary Views
 - Alert
 - Action Sheet
 - Modal View

Label

A **label** displays static text.

Create a stream or join one to share your best shots and enjoy friends' comments and contributions right in the iOS photos app.

API NOTE

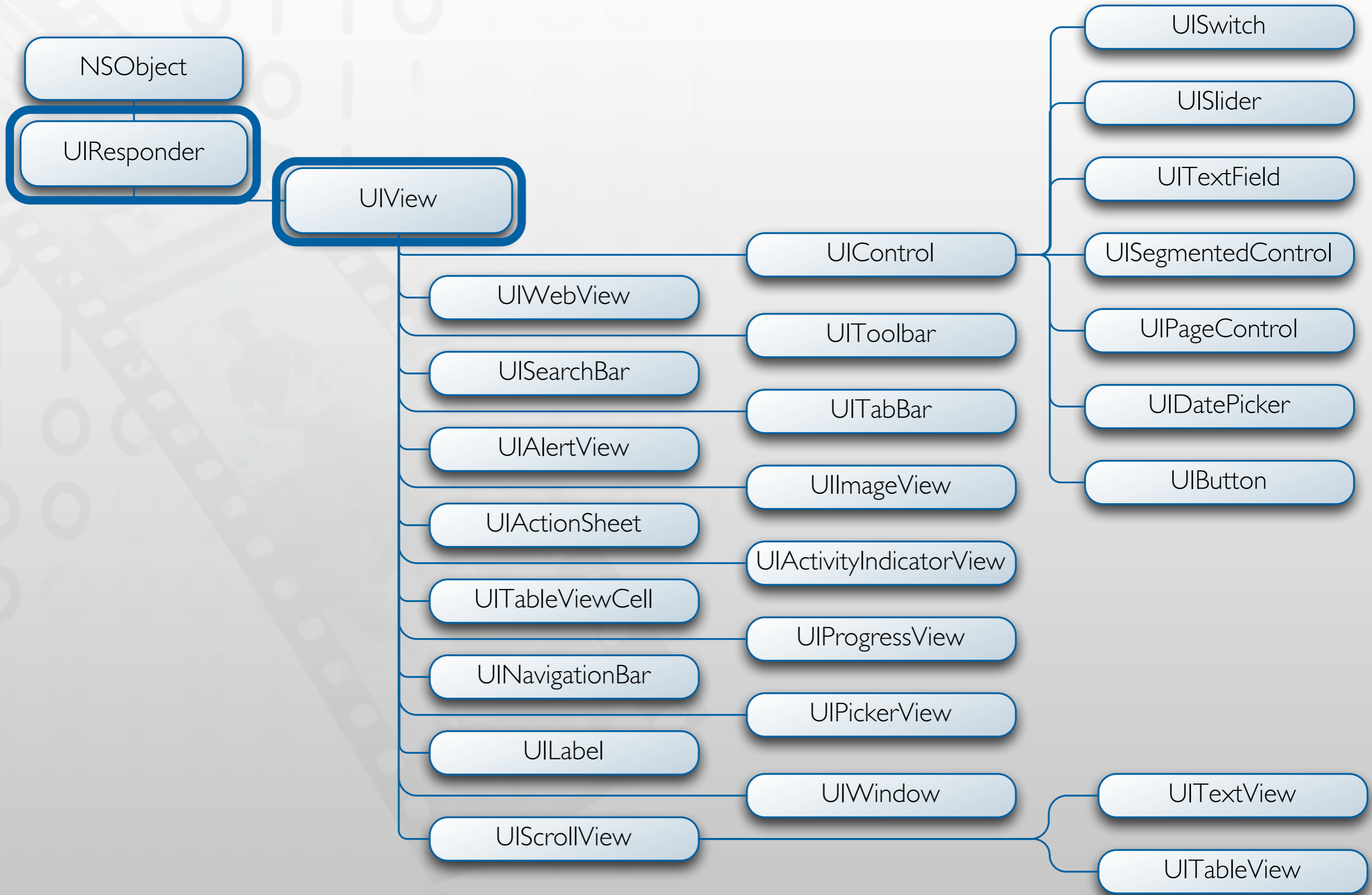
To learn more about defining labels in your code, see [UILabel Class Reference](#).

A label:

- Displays any amount of static text
- Doesn't allow user interaction except, potentially, to copy the text

Use a label to name or describe parts of your UI or to provide short messages to the user. A label is best suited for displaying a relatively small amount of text.

Take care to make your labels legible. It's best to support Dynamic Type and use the `UIFont` method `preferredFontForTextStyle` to get the text for display in a label. If you choose to use custom fonts, don't sacrifice clarity for fancy lettering or showy colors. (For guidelines about using text in an app, see [Color and Typography](#); to learn more about Dynamic Type, see "Text Styles" in [Text Programming Guide for iOS](#).)



View Concepts

- ✓ SDK provide many types of Views to show your content
 - At run-time Views are organized as a tree
 - Use Interface Builder to design your UI and connect it to code
 - Geometry of Views are determined by constraints

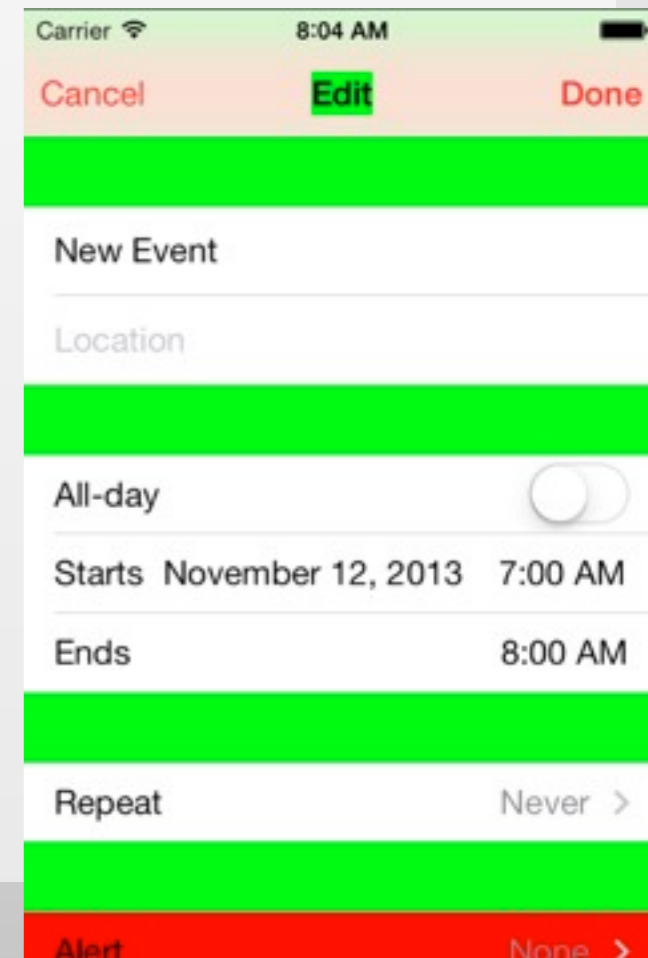
Demo: Hacking Calendar

```
(lldb) po [[UIWindow keyWindow] recursiveDescription]
```

```
<UIWindow
| <UILayoutContainerView
| | <UINavigationController
| | | <UIViewControllerWrapperView
| | | | <EKCalendarItemEditorTableView: 0xf870200; baseClass = UITableView;
| | | | | <UITableViewWrapperView
| | | | | | <UITableViewCell: 0xb19ad90
| | | ...
| | <UINavigationBar
| | | ...
| | <UINavigationControllerItemView
| | | <UILabel
| | | <UINavigationControllerItemView
| | | | ...
```

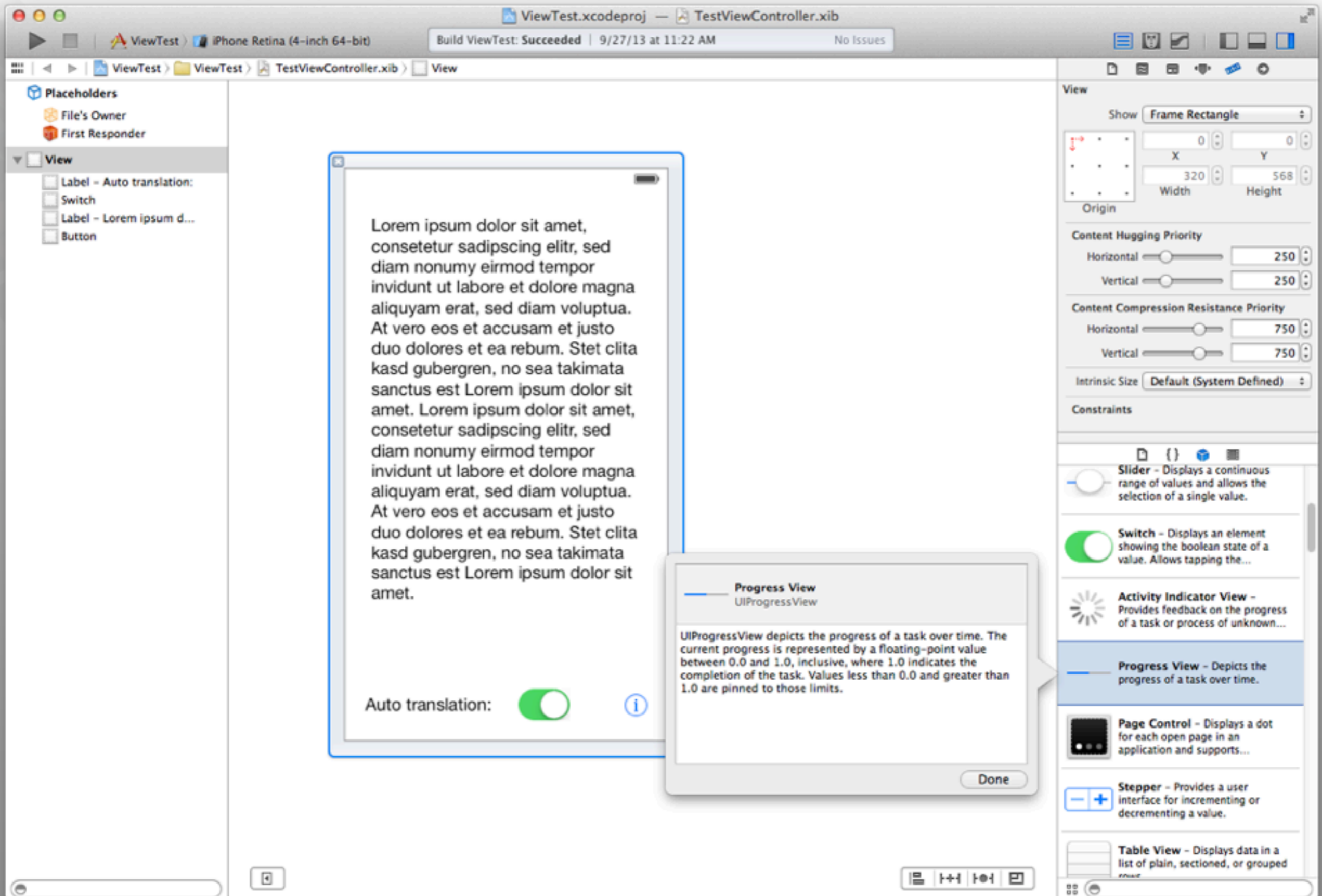
```
(lldb) expr ((UIView *)0xb19ad90).backgroundColor = [UIColor redColor]
```

```
(lldb) expr ((UIView *)0xf870200).backgroundColor = [UIColor greenColor]
```



View Concepts

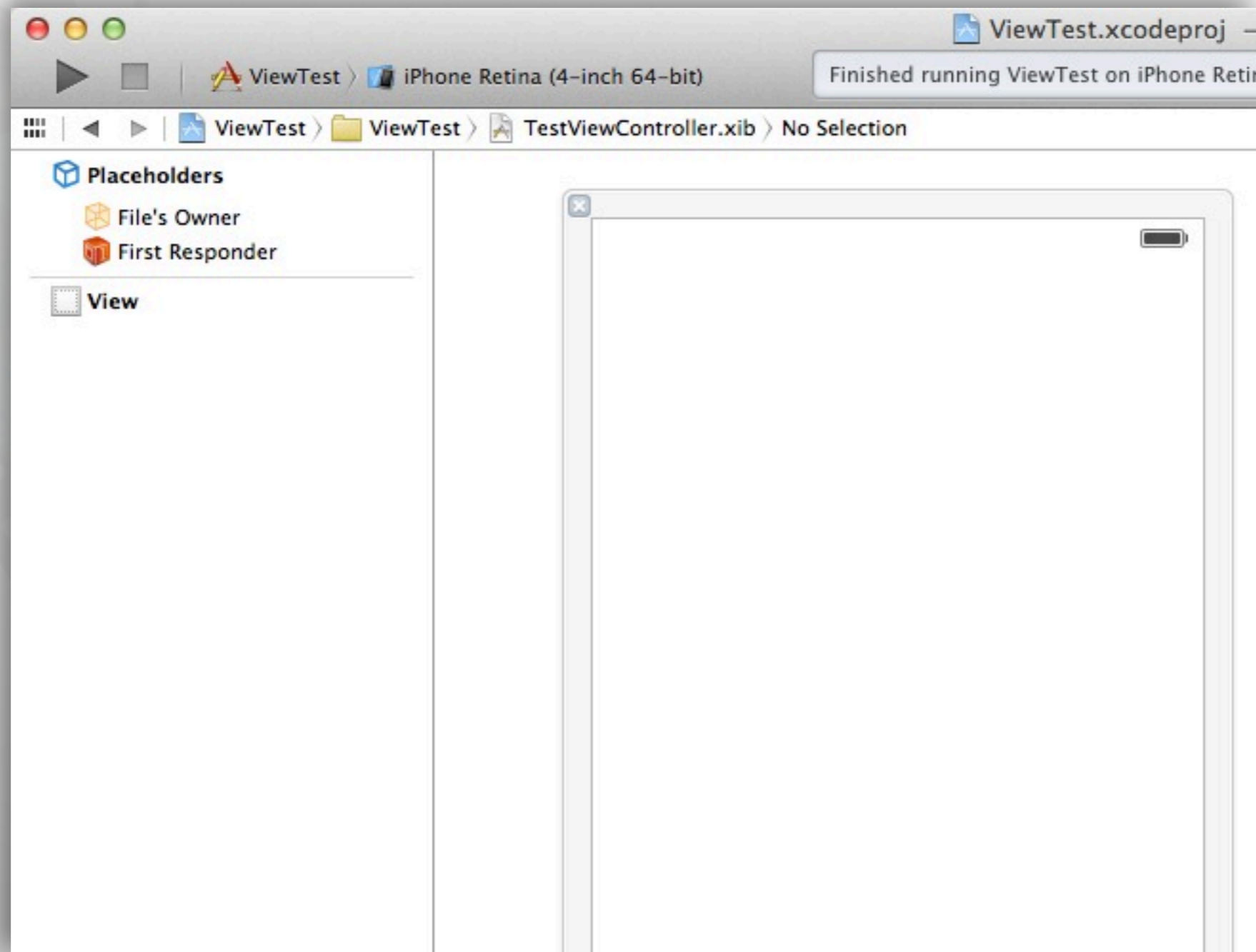
- ✓ SDK provide many types of Views to show your content
- ✓ At run-time Views are organized as a tree
 - Use Interface Builder to design your UI and connect it to code
 - Geometry of Views are determined by constraints



Interface Builder

- Graphical tool to layout user interfaces
- Create the widget hierarchy
- Set attributes of widgets
- Set up connections between the widgets
- Store these informations in nib files

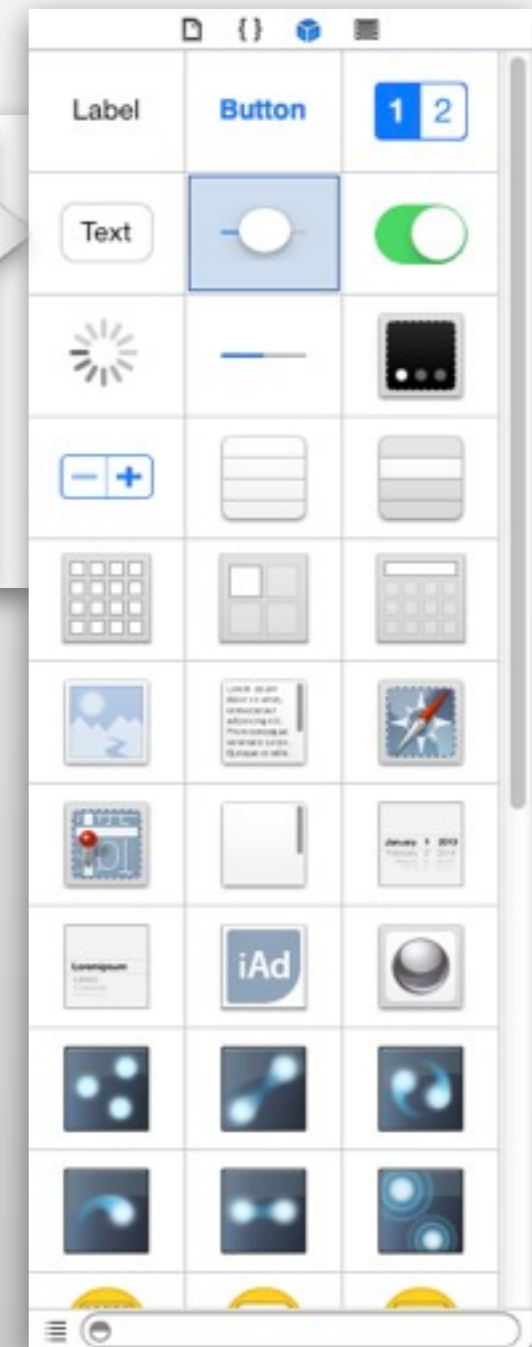
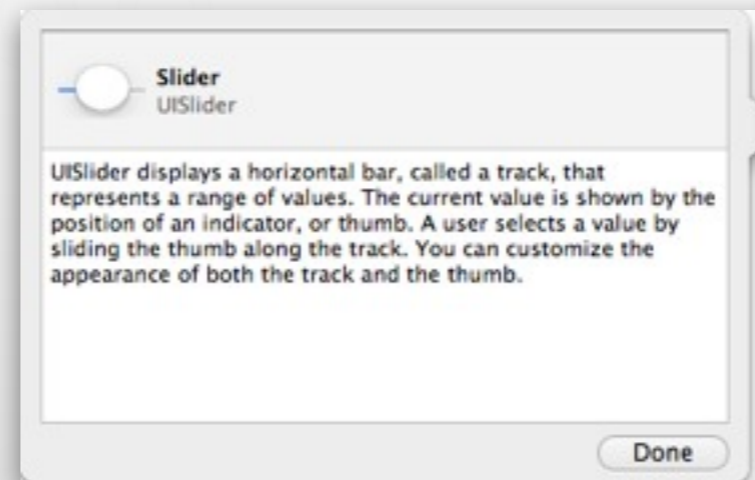
The Anatomy of a xib File



The Source of a xib

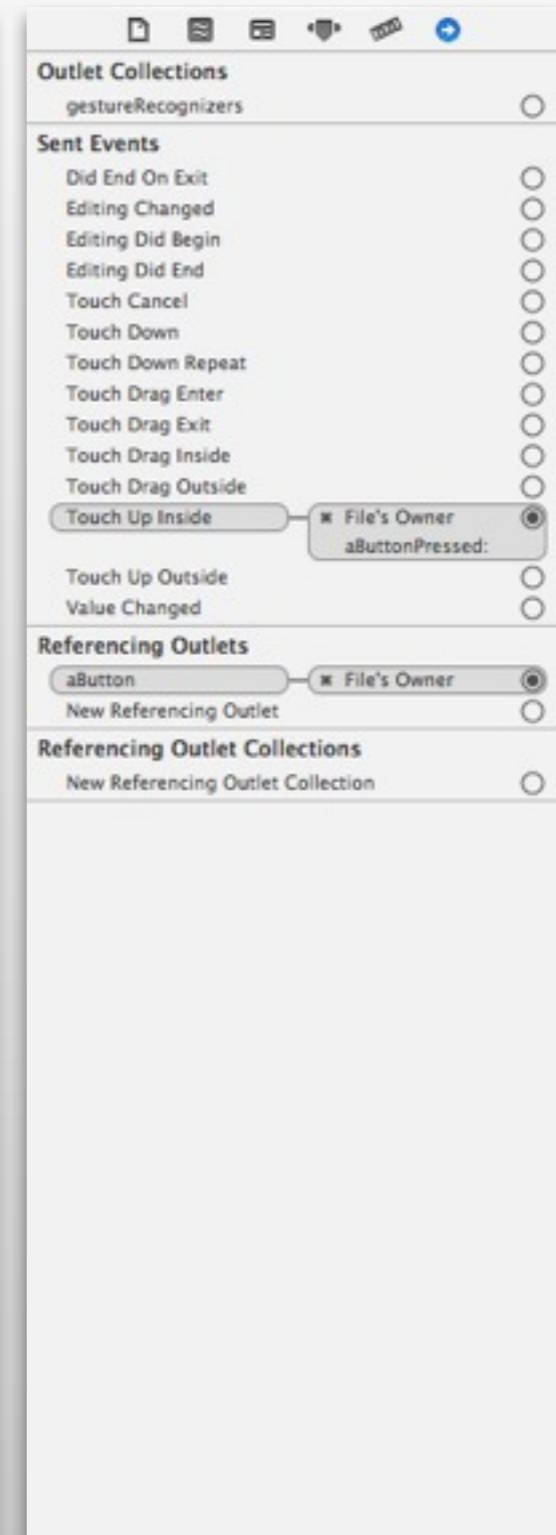
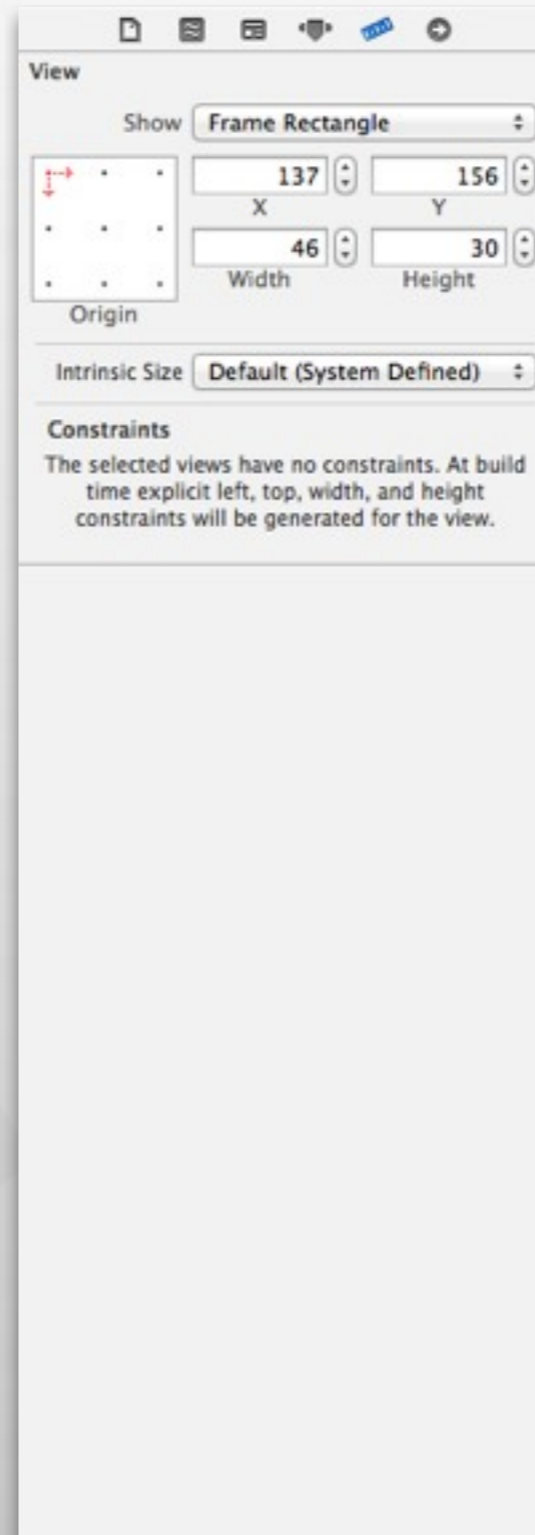
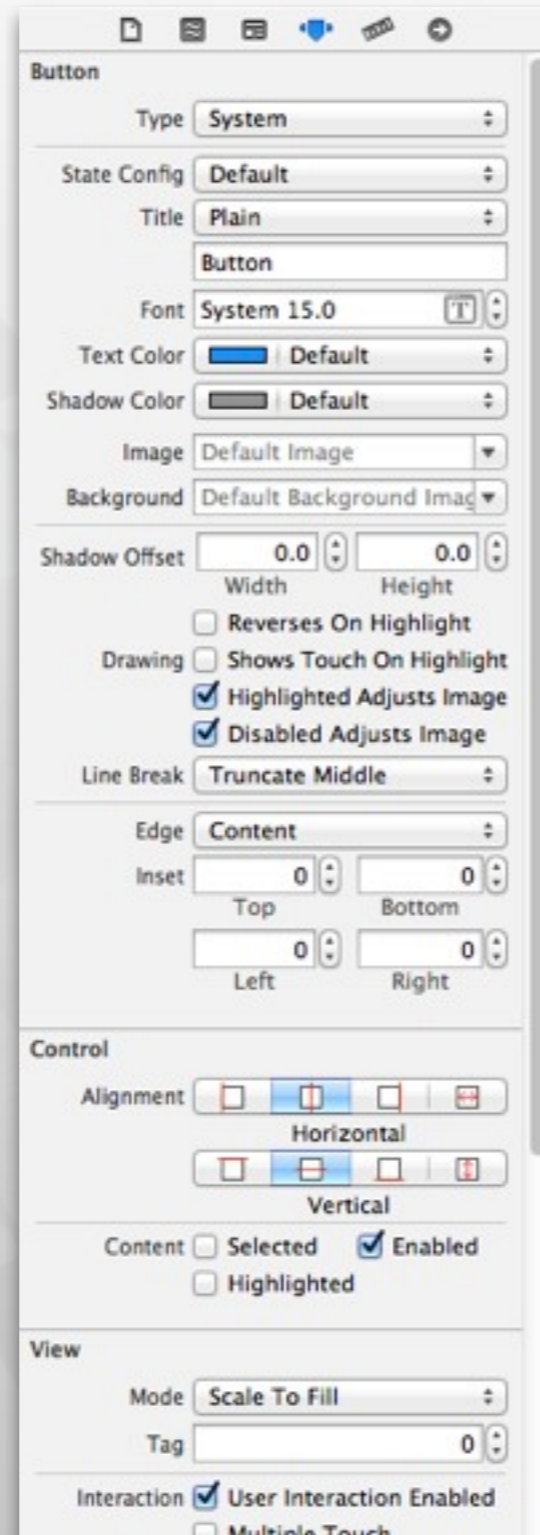
```
<?xml version="1.0" encoding="UTF-8"?>
<archive type="com.apple.InterfaceBuilder3.CocoaTouch.XIB" version="8.00">
  <data>
    <int key="IBDocument.SystemTarget">1280</int>
    <string key="IBDocument.SystemVersion">11C74</string>
    <string key="IBDocument.InterfaceBuilderVersion">1938</string>
    <string key="IBDocument.AppKitVersion">1138.23</string>
    <string key="IBDocument.HIToolboxVersion">567.00</string>
    <object class="NSMutableDictionary" key="IBDocument.PluginVersions">
      <string key="NS.key.0">com.apple.InterfaceBuilder.IBCocoaTouchPlugin</string>
      <string key="NS.object.0">933</string>
    </object>
    <array key="IBDocument.IntegratedClassDependencies">
      <string>IBUITextView</string>
      <string>IBUISwitch</string>
      <string>IBUIButton</string>
      <string>IBUIView</string>
      <string>IBUILabel</string>
      <string>IBProxyObject</string>
    </array>
    <array key="IBDocument.PluginDependencies">
      <string>com.apple.InterfaceBuilder.IBCocoaTouchPlugin</string>
    </array>
    <object class="NSMutableDictionary" key="IBDocument.Metadata">
      <string key="NS.key.0">PluginDependencyRecalculationVersion</string>
      <integer value="1" key="NS.object.0"/>
    </object>
    <array class="NSMutableArray" key="IBDocument.RootObjects" id="1000">
      <object class="IBProxyObject" id="841351856">
        <string key="IBProxiedObjectIdentifier">IBFilesOwner</string>
        <string key="targetRuntimeIdentifier">IBCocoaTouchFramework</string>
      </object>
      <object class="IBProxyObject" id="371349661">
        <string key="IBProxiedObjectIdentifier">IBFirstResponder</string>
        <string key="targetRuntimeIdentifier">IBCocoaTouchFramework</string>
      </object>
      <object class="IBUIView" id="474857037">
        <reference key="NSNextResponder"/>
        <int key="NSvFlags">292</int>
        <array class="NSMutableArray" key="NSSubviews">
          <object class="IBUITextView" id="694905917">
            <reference key="NSNextResponder" ref="474857037"/>
            <int key="NSvFlags">274</int>
            <string key="NSFrame">{{20, 20}, {280, 385}}</string>
            <reference key="NSSuperview" ref="474857037"/>
          </object>
        </array>
      </object>
    </array>
  </data>
</archive>
```

Laying out the User Interface



- The library contains all UI Widgets
- Drag them to your view
- See instantly what your UI looks like
- Test your UI in the iPhone Simulator

Set Widget Attributes

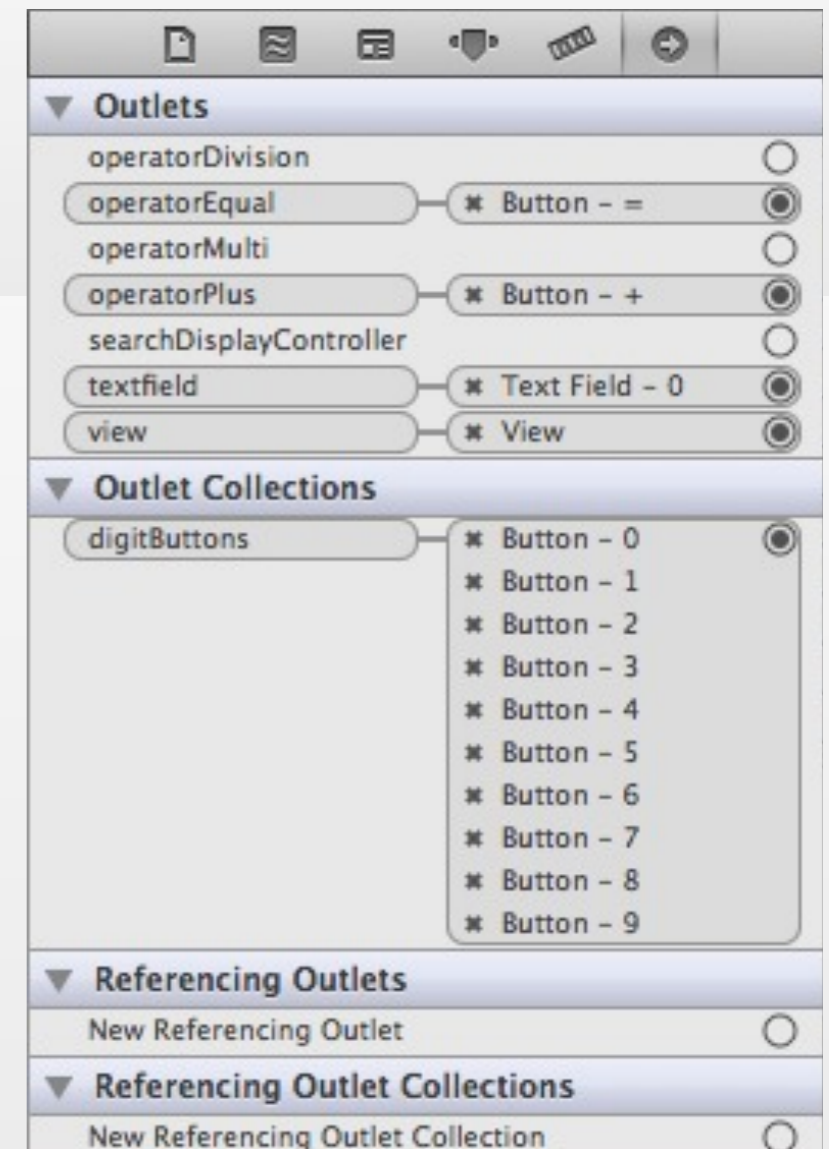


Connecting Widgets and Code

- IBActions
 - Tags a method as a target for an interface action
- IBOutlets
- IBOutletCollection
 - Variables to populate with objects from a nib file

Connecting Widgets and Code

```
@interface iCalcView: UIView {  
  
}  
  
- (IBAction)addDigit:(id)sender;  
- (IBAction)calculateResult:(id)sender;  
  
//declared properties  
//textfield  
@property (nonatomic,weak) IBOutlet UITextField *textfield;  
//operators: + =  
@property (nonatomic,weak) IBOutlet UIButton *operatorPlus  
@property (nonatomic,weak) IBOutlet UIButton *operatorEqual;  
//digits  
@property (nonatomic,weak) IBOutletCollection (UIButton) NSArray* digitButtons;
```



Interface Builder Demo

View Concepts

- ✓ SDK provide many types of Views to show your content
- ✓ At run-time Views are organized as a tree
- ✓ Use Interface Builder to design your UI and connect it to code
- Geometry of Views are determined by constraints

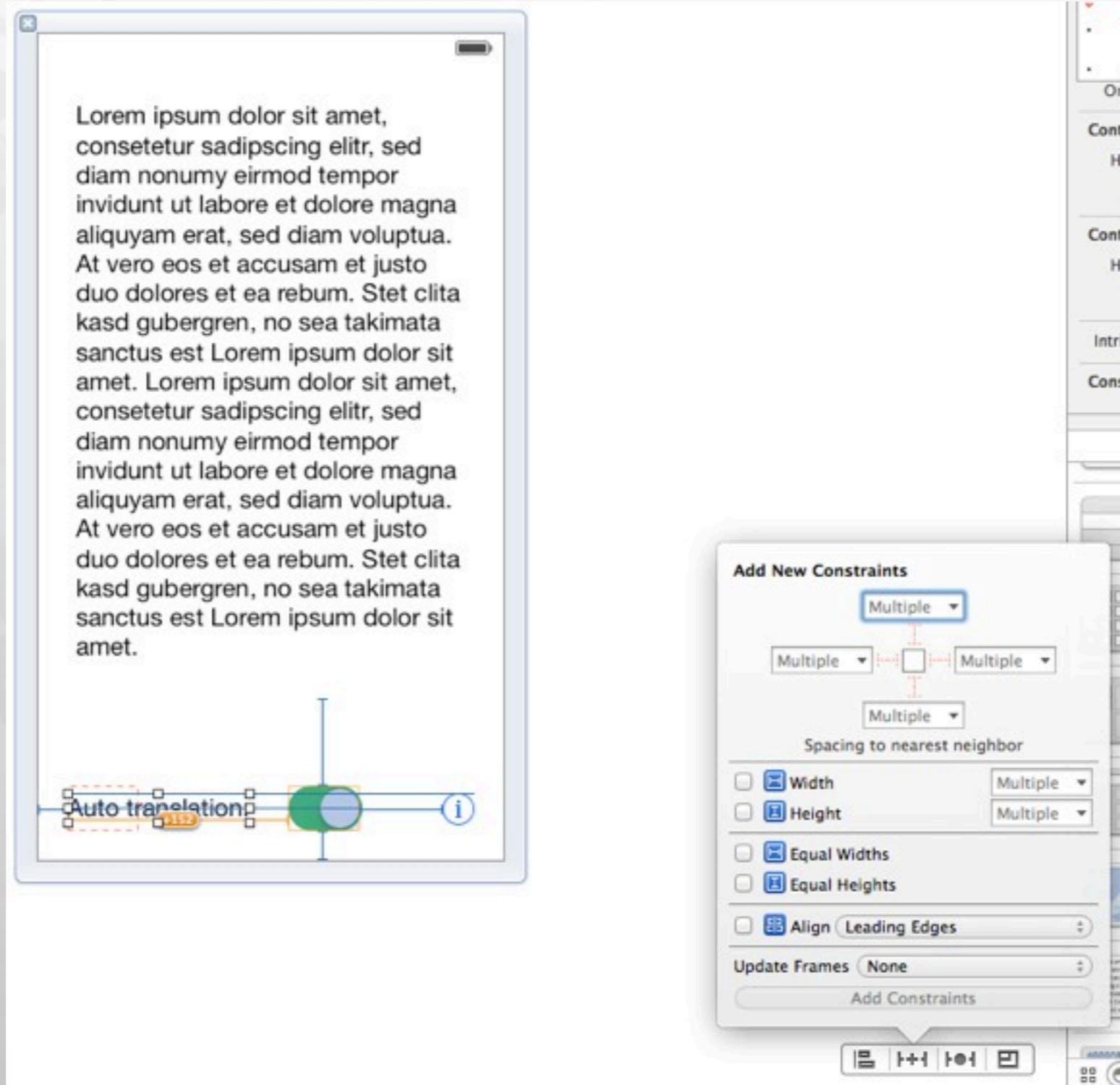
Auto Layout

- Preferred layout management
- Allows you to create views that work both in portrait and landscape mode
- Available in iOS 6 and higher
- Spatial relationships expressed by constraints

Auto Layout Constraints

- Constraints are mathematical expressions
 - \leq , $=$, \geq
- Constraints have a priority level
- The runtime tries to solve the system of equations

Adding Constraints



View Concepts

- ✓ SDK provide many types of Views to show your content
- ✓ At run-time Views are organized as a tree
- ✓ Geometry of Views are determined by constraints
- ✓ Use Interface Builder to design your UI and connect it to code

An alle die programmieren können!!!

Meldet euch zum bonding Hackathon an & stellt eure Programmierfähigkeiten unter Beweis.

Wir sorgen für

- eine **Einarbeitungsstunde** in die **Systemsprache**
- kostenlose **Verpflegung**
- **Shuttle-Service**
- attraktive **Siegerpreise**
- Siegerehrungs**party** am 26.11

bonding Hackathon

Implementiere die Zukunft in der Nacht vom 22. auf den 23. November 2013!

Anmeldung

www.bonding.de/ac_hackathon

KOSTENLOS
von Studenten
für Studenten

bonding – erlebe, was du werden kannst.

RWTH AACHEN

TAG DER 06.12. INFORMATIK

Informatikzentrum | Ahornstr. 55 | Aachen

NEUER
PROFESSOREN 14⁰⁰ ONE MINUTE MADNESS 15⁰⁰ HAUPT- VORTRAG

13⁰⁰ BEGRÜSSUNG 13⁴⁵ VORSTELLUNG

12⁰⁰ AUSSTELLUNG

FIRMENKONTAKTMESSE &

16⁰⁰ FIRMEN: ONE MINUTE MADNESS WWW.TDI.AC

17⁰⁰ ABSOLVENTENEHRUNG 19⁰⁰ SEKTEMPFANG 19³⁰ DINNER

21⁰⁰ PARTY
MIT DJ SHALIEN

Popcorn by IVU
Watterott iWall
Cocktailbar by Think Cell
Livemusik von 5th Edition by Inform & Party mit DJ Shalien

IVU TRAFFIC TECHNOLOGIES AG
INFORM
think-cell
wetterott
msg systems
ascom
FERCHAU ENGINEERING
GENERALI Informatik Services
itestra Software Productivity
SENACOR
Ubisense
NUANCE
traveltainment the amadeus leisure group
ERICSSON
aixigo
aixness
dSPACE
Galileo Press
LAMBERTZ
Sparkasse für die Hochschulen
Springer Vieweg

Review

- A calendar app reminds user of a scheduled event
- A news reader app loads updates in background and updates the UI upon completion
- An app determines a path for an image that is shipped with the app itself
- Base class for objects that respond to UI events
- Base class for managing screen content

View Concepts

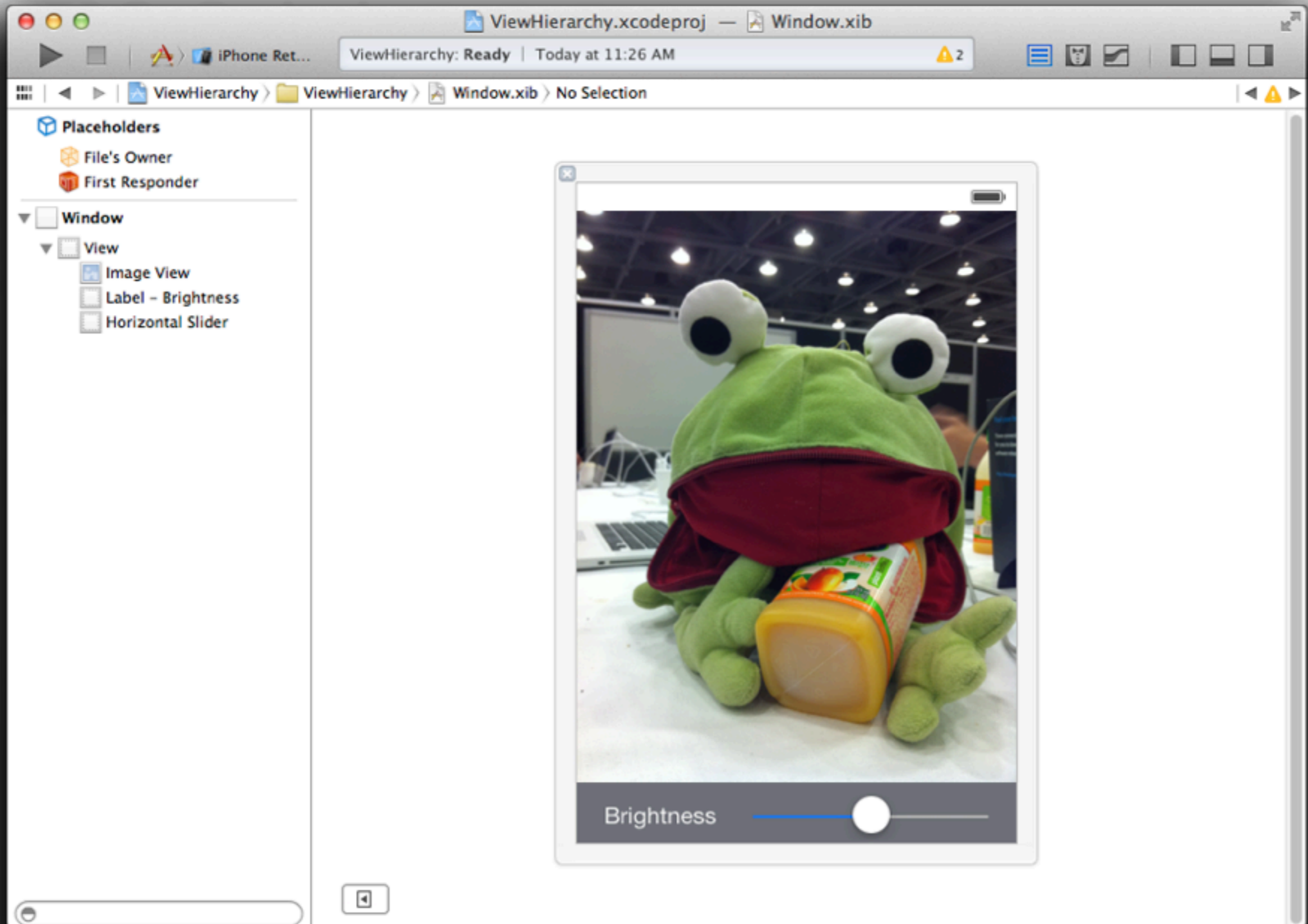
- ✓ SDK provide many types of Views to show your content
- ✓ At run-time Views are organized as a tree
- ✓ Geometry of Views are determined by constraints
- ✓ Use Interface Builder to design your UI and connect it to code

View Programming

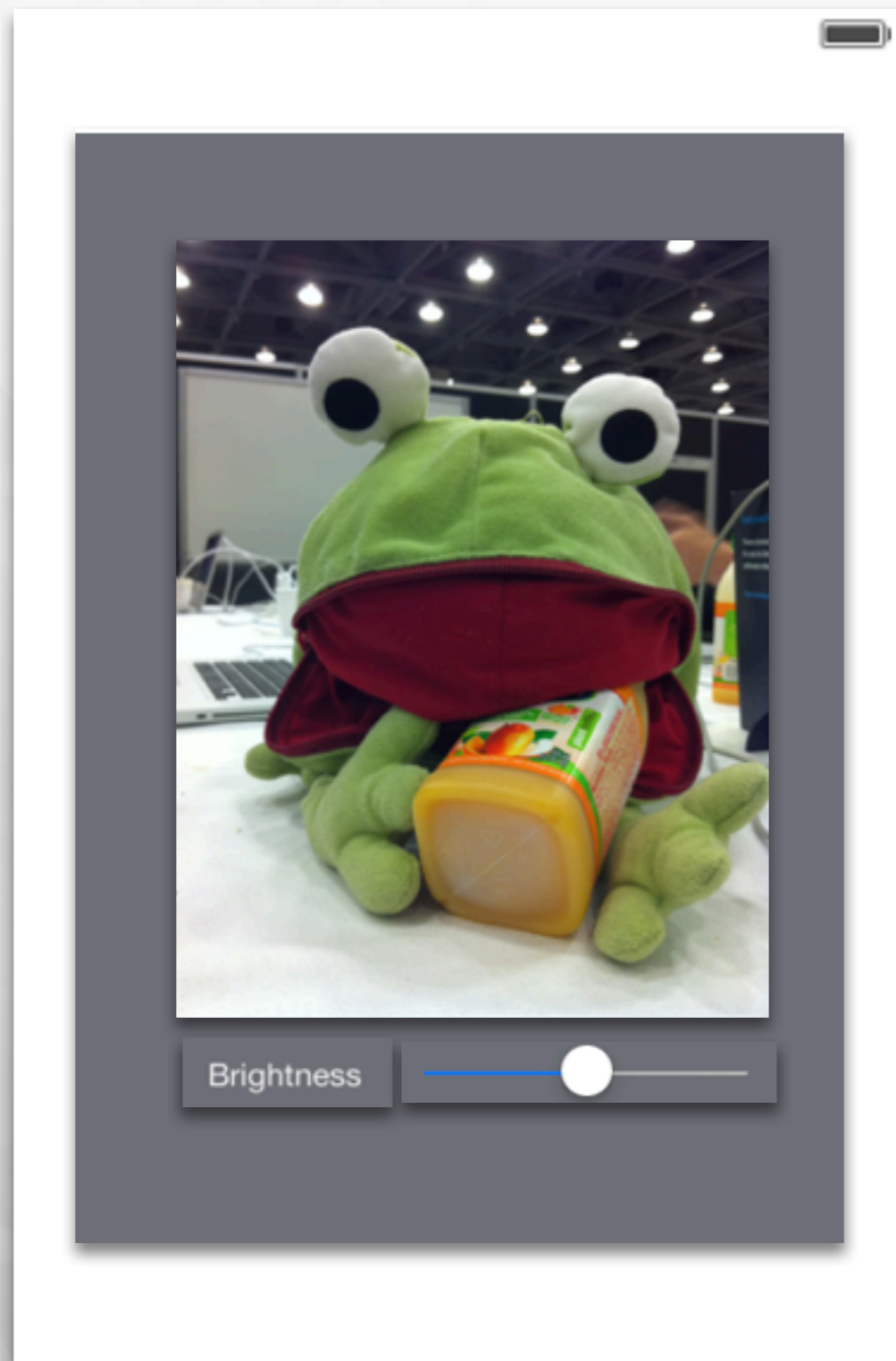
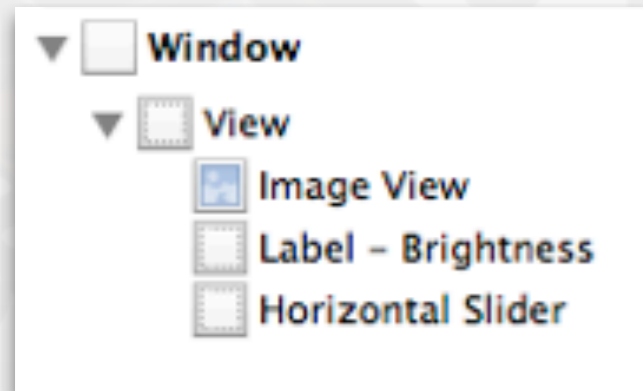
UIView

- Defines a rectangular area on the screen
- Two responsibilities
 - Render content
 - React to user input
 - Manage subviews
- Layout as view hierarchy

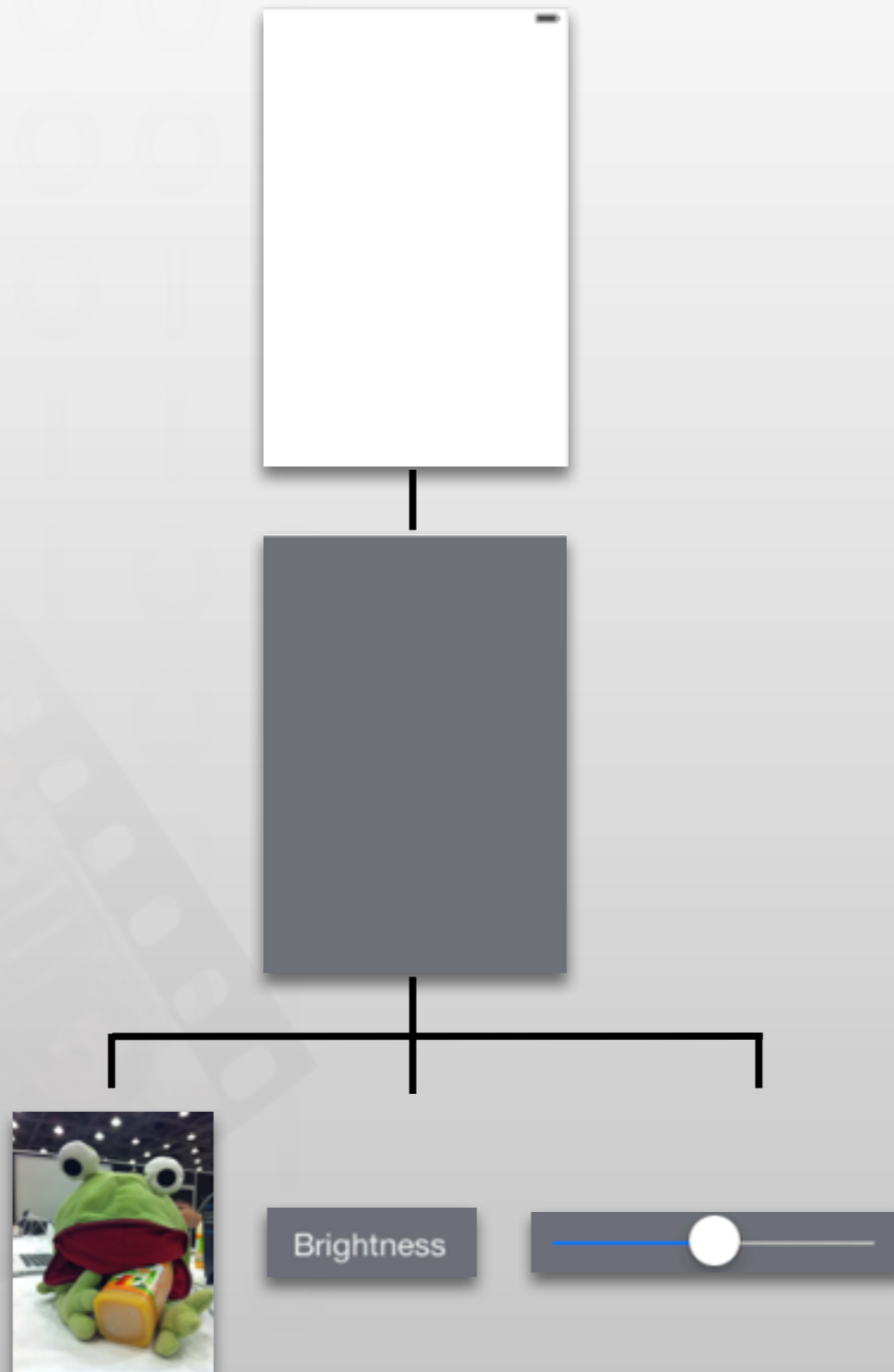
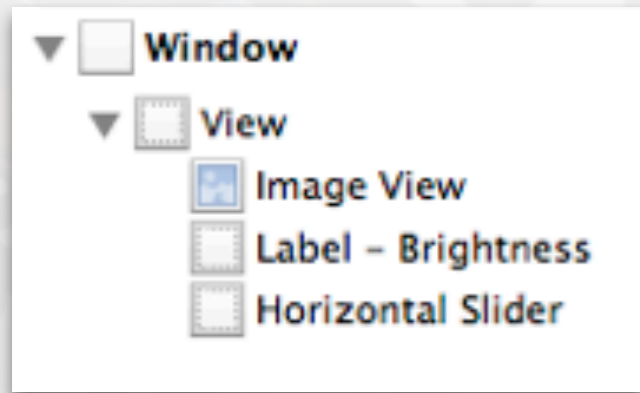
View Hierarchies



View Hierarchies

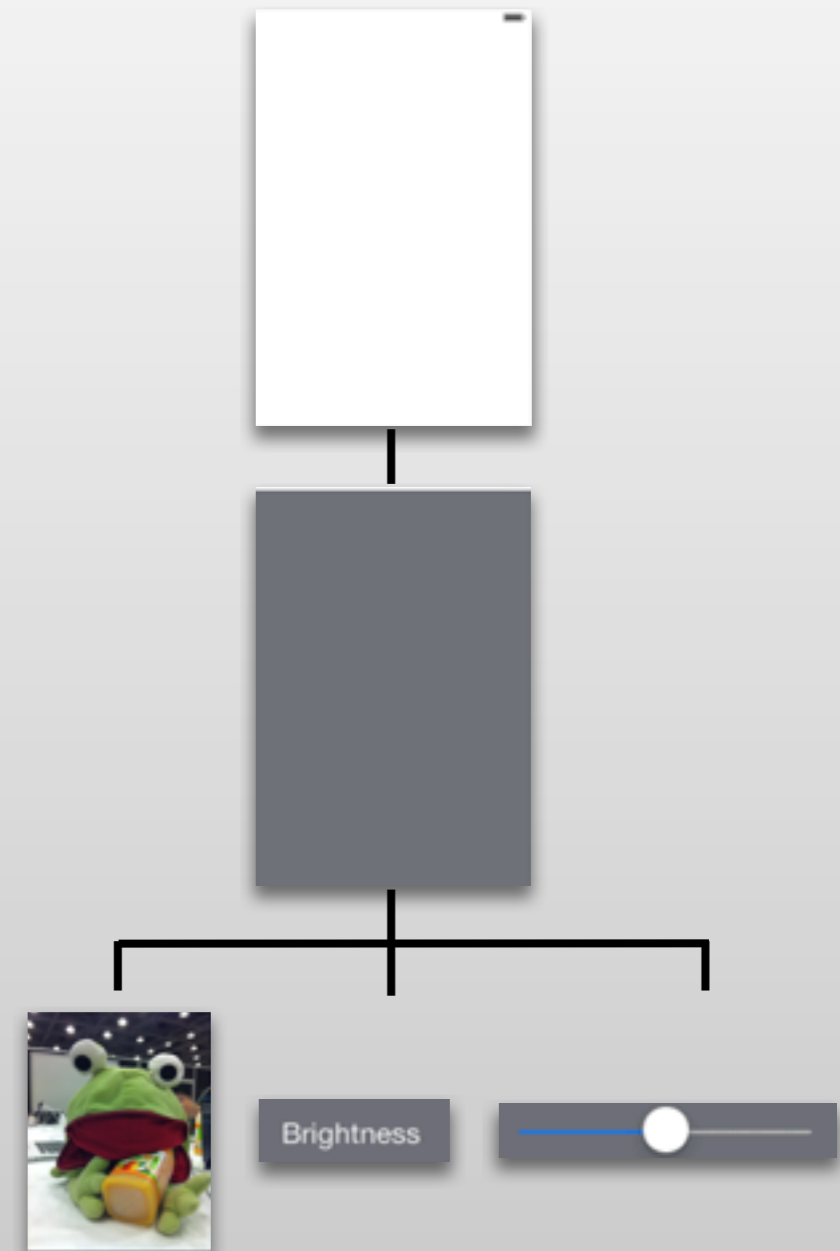


View Hierarchies



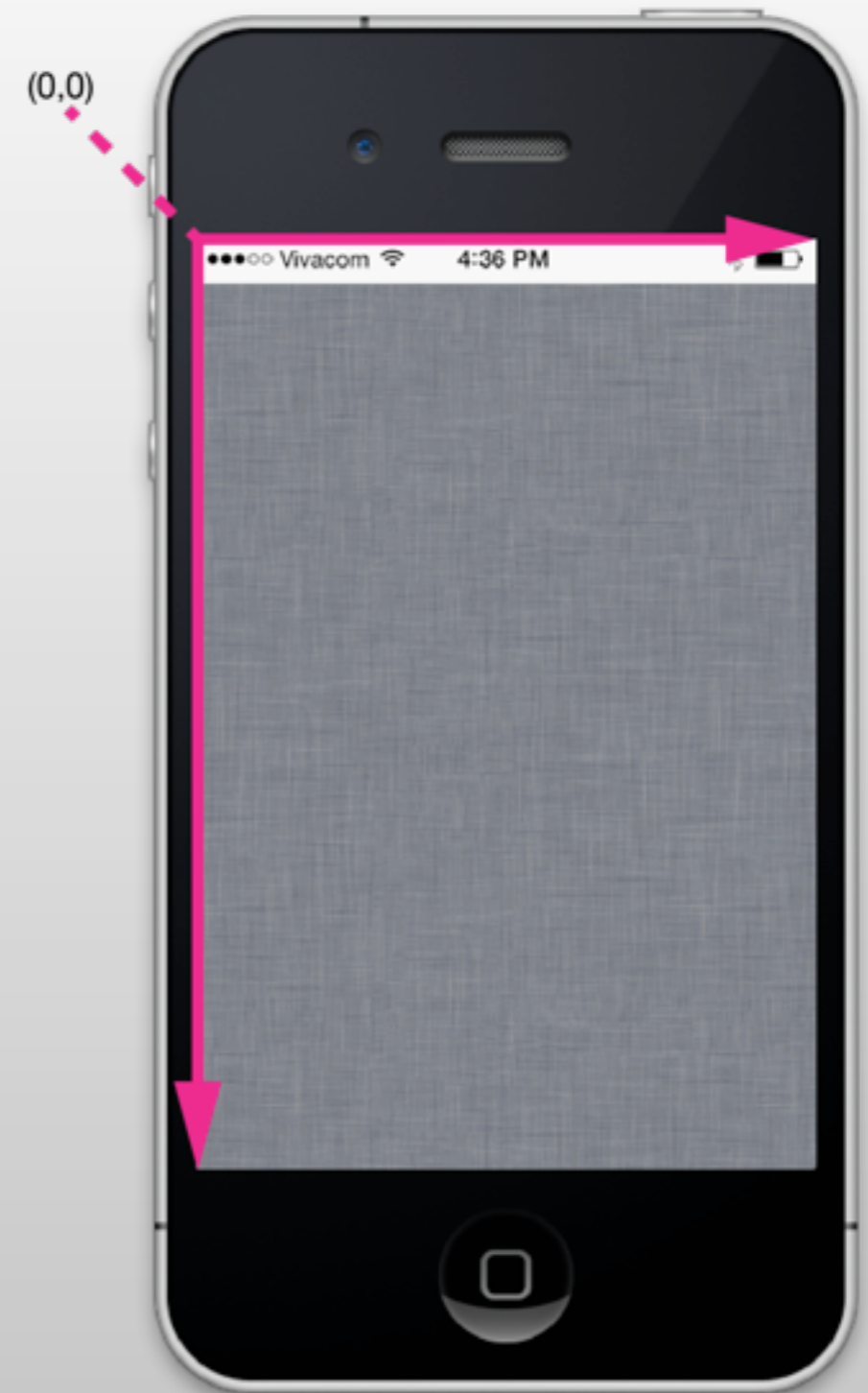
Managing Views

- addSubview
- addSubview:atIndex:
- addSubview:aboveSubview:
- removeFromSuperview
- bringSubviewToFront:
- sendSubviewToBack:
- exchangeSubviewAtIndex:withSubviewAtIndex:



View Coordinate System

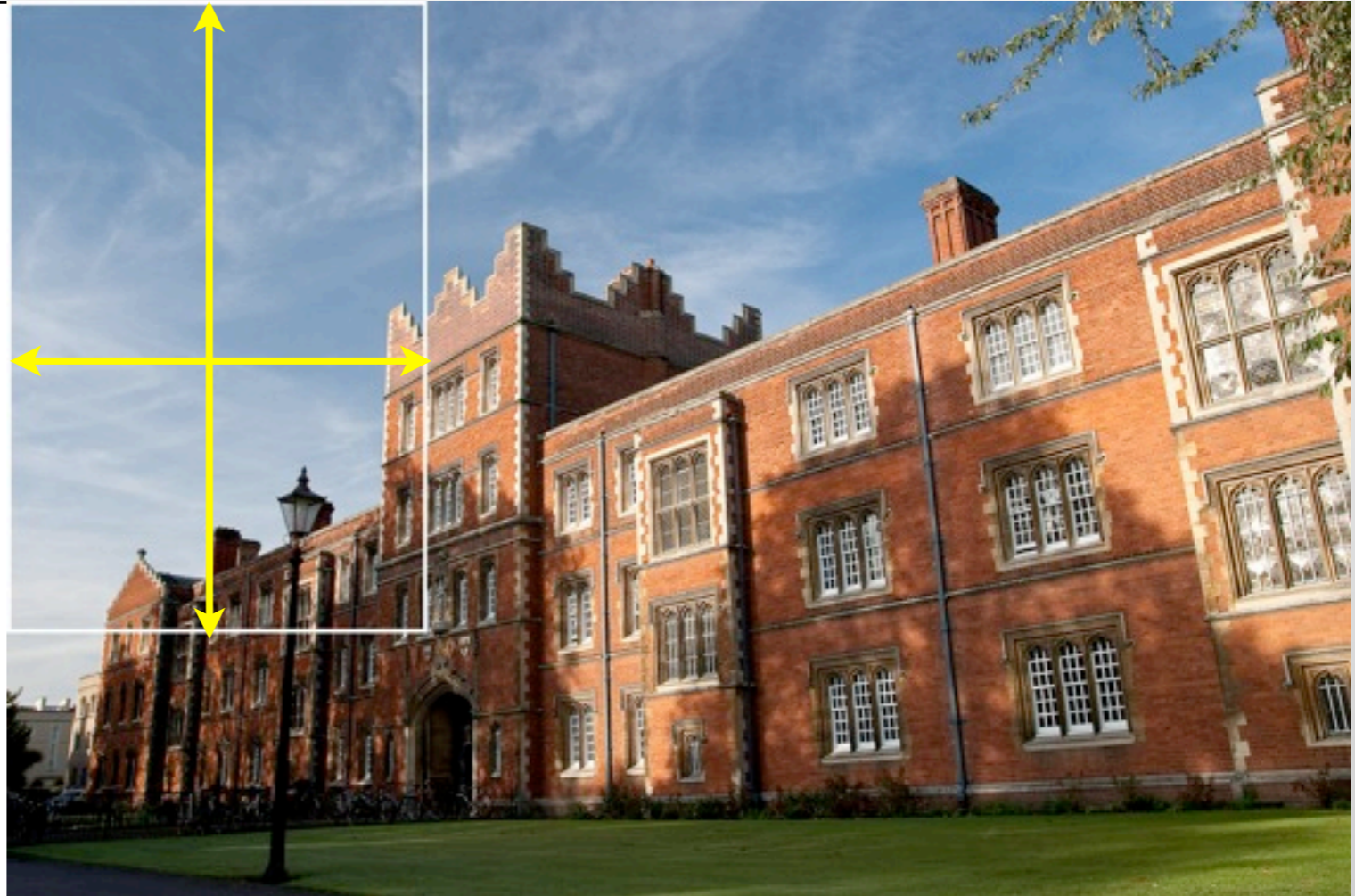
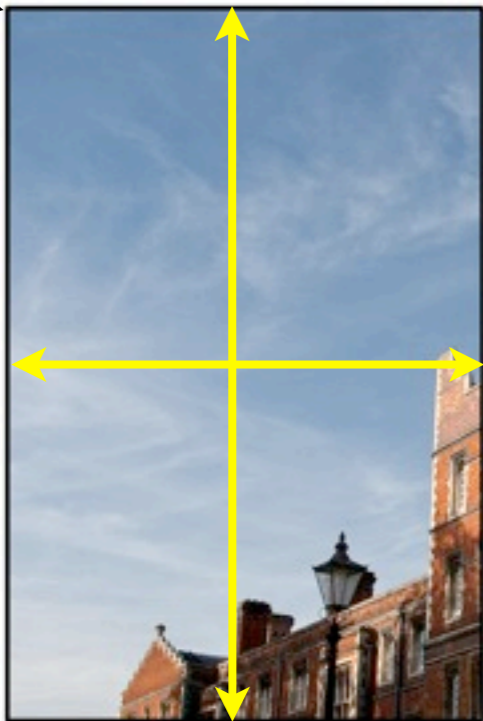
- Origin at top-left corner
- Coordinates use points as unit
 - iPhone: 320x480
 - iPhone 5: 320x568
 - iPad: 768x1024



View Geometry

- **Frame**
 - A rectangle with origin and size **relative to the superview**
- **Bounds**
 - A rectangle with origin (0,0) and the size of the view
- **Center**
 - The center point of the frame rectangle

Frame vs. Bounds



Frame (looking from outside)

Origin: (140.0,35.0)

Size: (320.0,480.0)

Bound (looking from the view itself)

Origin: (0.0,0.0)

Size: (320.0,480.0)

Setting Values

- Setting `frame`:
 - `bounds` matches the size
 - `center` is adjusted
- Setting `center`:
 - The origin of `frame` is set accordingly
- Setting the size of `bounds`
 - The size of `frame` is set accordingly

Content Modes & Scaling



Content Mode

UIViewContentModeScaleToFill



Distorting



UIViewContentModeScaleAspectFit



Nondistorting



UIViewContentModeScaleAspectFill



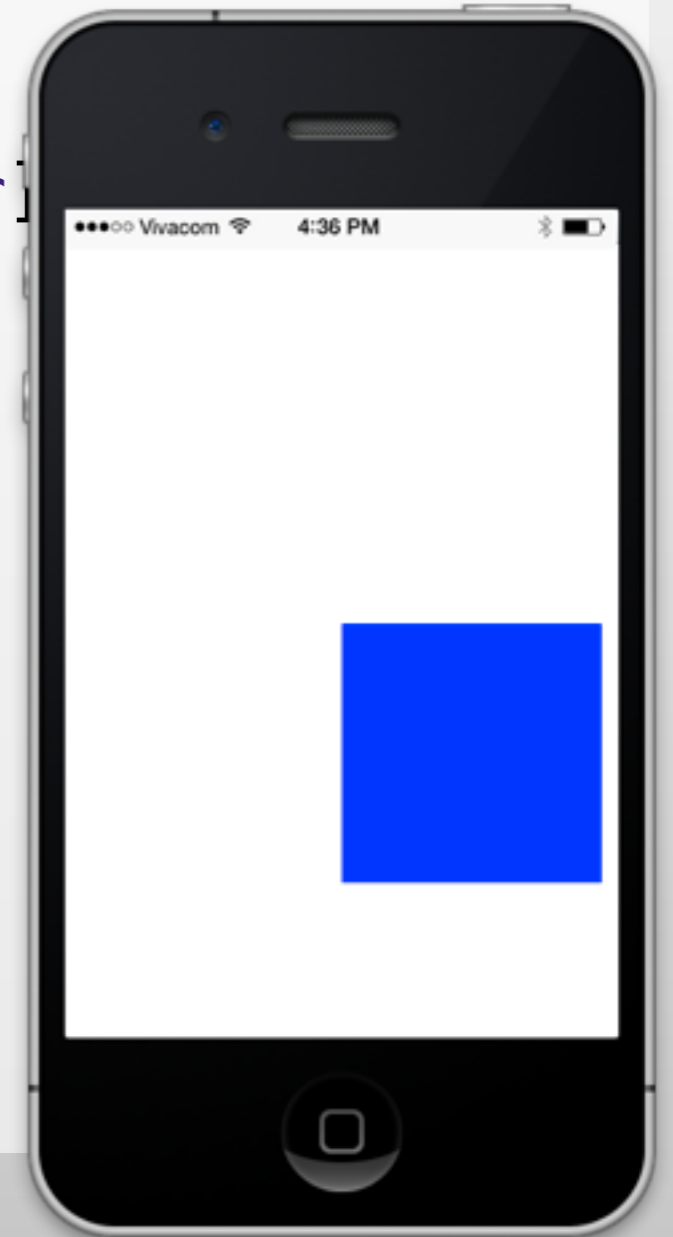
Nondistorting



Creating Views

```
// Create a view with blue background
CGRect viewFrame = CGRectMake(160, 240, 150, 150);
UIView *blueView = [[UIView alloc]
initWithFrame:viewFrame];
blueView.backgroundColor = [UIColor blueColor];

// add it to the main window
[window addSubview:blueView];
```



Subclassing UIView

```
// Only override drawRect: if you perform custom drawing.  
// An empty implementation adversely affects performance during animation.  
- (void)drawRect:(CGRect)rect {}  
  
[view setNeedsDisplay];  
  
// Overridden by subclasses to layout subviews when layoutIfNeeded is  
invoked.  
// The default implementation of this method does nothing  
- (void)layoutSubviews{}  
  
[view setNeedsLayout];
```

Lecture 6

Reaction to Events

Lecture 7

- Adjust properties of the view and its subviews
- Mark the view as needing a change in its layout
- Mark the view as needing to be redrawn
- Notify a controller that data has changed

Scroll Views

- Container view
- Displays content larger than the app window
- Support for scrolling
- Support for zooming

UIScrollView

```
UIScrollView *scrollView = [[UIScrollView alloc]
initWithFrame:window.bounds];
[window addSubview:scrollView];
// window retains its subview, thus we can release the view here
[scrollView release];

CGRect contentFrame = CGRectMake(0.,
                                0.,
                                window.bounds.size.width*2.,
                                window.bounds.size.height*2.);

scrollView.contentSize = contentFrame.size;
scrollView.contentOffset = window.center;

// add the content view
[scrollView addSubview:aView];
```

UIScrollView: Zooming

```
// Enable zooming
scrollView.minimumZoomScale = 0.5;
scrollView.maximumZoomScale = 2.5;
scrollView.delegate = self;




#pragma mark UIScrollView delegate methods

- (UIView*)viewForZoomingInScrollView:(UIScrollView *)scrollView;
{
    return [window viewWithTag:1];
}
```

Summary

- Interface Builder
- UIView
- View hierarchies

- Reading Assignment:

-  View Programming Guide
-  Interface Builder User Guide
-  UIView Class Reference

- Taking control of Auto Layout in Xcode 5 **WWDC2013**



UIViewController

- Manages typically one screen
- Flushes the view on low-memory situations
- Resizes the view on orientation change
- Creates modal views on top of the current view