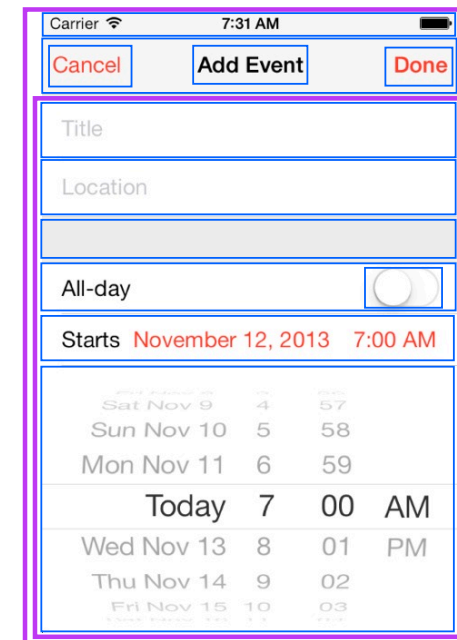
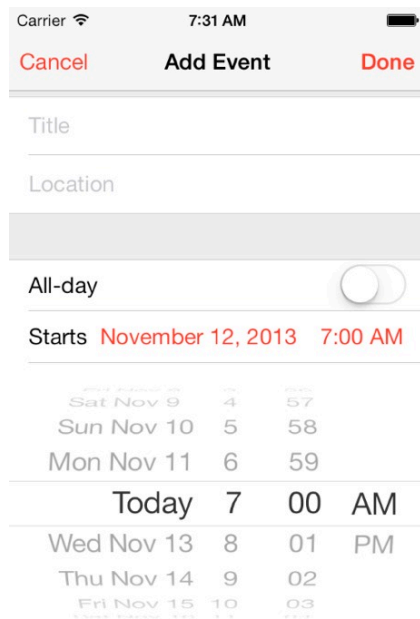


View Concepts

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>

- 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



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
- Containers Views
 - Activity
 - Activity View Controller
 - CollectionView
 - Container View Controller
 - Image View
 - Map View
 - Page View Controller
 - Popover (iPad Only)
 - ScrollView
 - Split View Controller (iPad Only)
 - TableView
 - WebView
- Controls
 - ActivityIndicator
 - 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
 - TextField
- 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

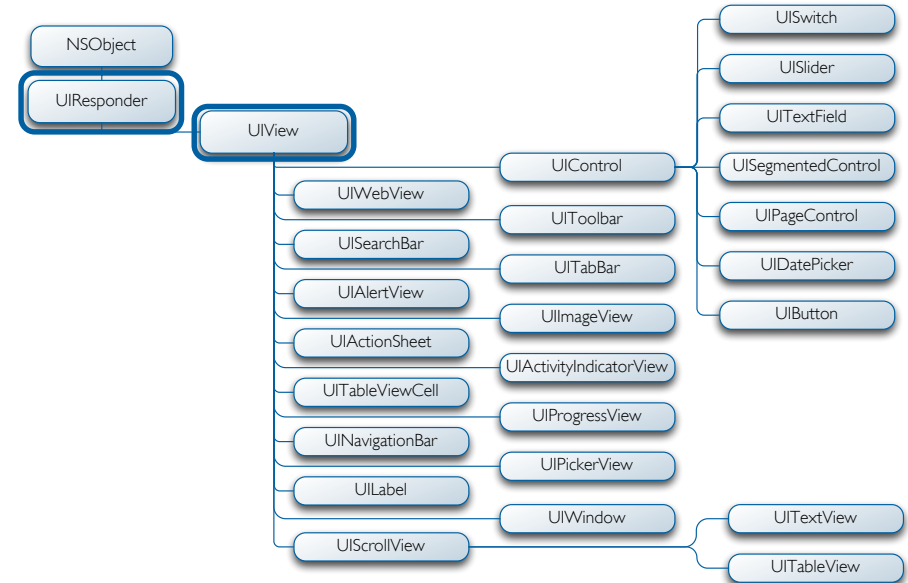
To learn more about defining labels in your code, see [UILabel Class Reference](#).

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

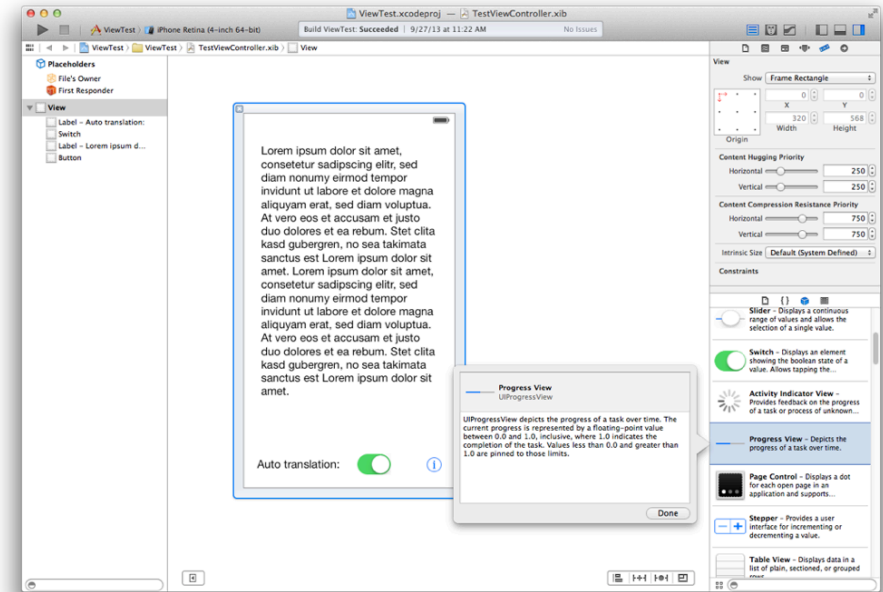
```
(lldb) po [[UIWindow keyWindow] recursiveDescription]
<UIWindow
  |<UILayoutContainerView
    |<UINavigationController
      |<UIViewControllerWrapperView
        |<EKCalendarItemEditorTableView: 0xf870200; baseClass = UITableView;
          |<UITableViewControllerWrapperView
            |<UITableViewCell: 0xb19ad90
              ...
            ...
          ...
        ...
      |<UINavigationBar
        ...
        |<UINavigationController
          |<UILabel
            |<UINavigationController
              ...
            ...
          ...
        ...
      ...
    ...
  ...

```

```
(lldb) expr ((UIView *)0xb19ad90).backgroundColor = [UIColor redColor]
(lldb) expr ((UIView *)0xf870200).backgroundColor = [UIColor greenColor]
```

View Concepts

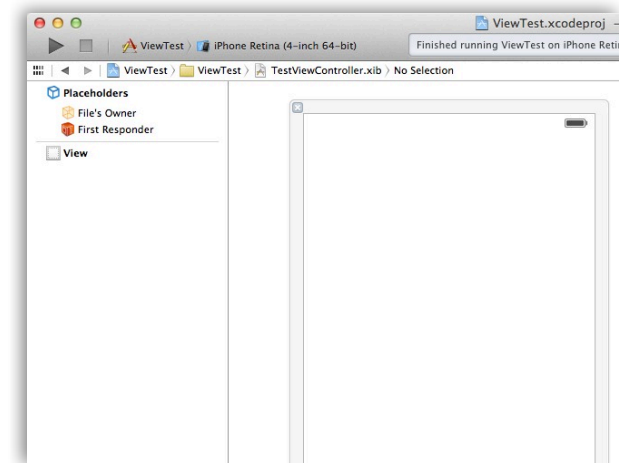
- ✓ SDK provide many types of Views to show your content
- ✓ At run-time Views are organized as a tree
- Use Interface Builder to design your UI and connect it to code
- Geometry of Views are determined by constraints



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

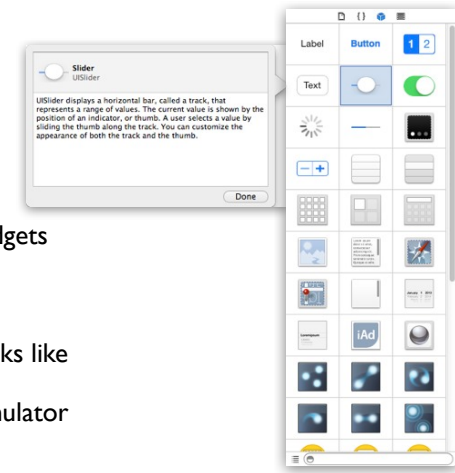
The Anatomy of a xib File



The Source of a xib

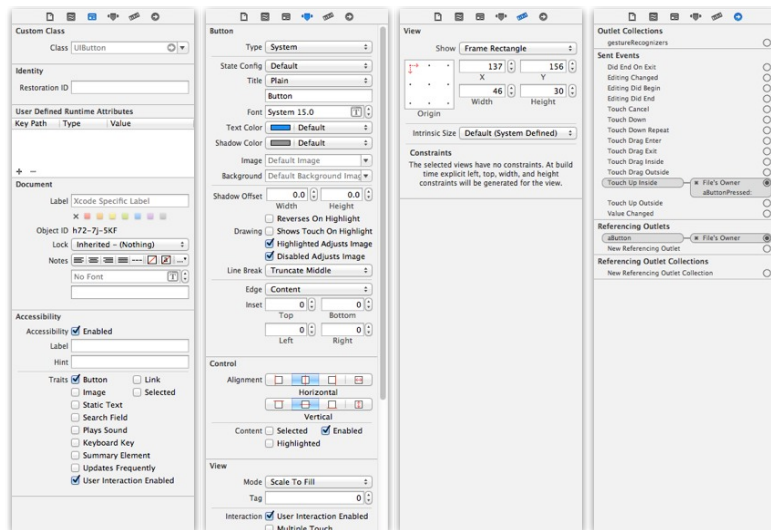
```
<?xml version="1.0" encoding="UTF-8"?>
<archive type="com.apple.InterfaceBuilder3.CocoaTouch.XIB" version="8.00">
  <data>
    <int key="IBDocument.SystemTarget">1280</int>
    <string key="IBDocument.SystemVersion">11.74</string>
    <string key="IBDocument.InterfaceBuilderVersion">1938</string>
    <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">0</integer>
    </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="474857837">
        <reference key="NSNextResponder" ref="371349661"/>
        <int key="NSVFlags">292</int>
        <array class="NSMutableArray" key="NSSubviews">
          <object class="IBUITextView" ref="694905917">
            <reference key="NSNextResponder" ref="474857837"/>
            <int key="NSVFlags">274</int>
            <string key="NSFrame">{{20, 20}, {280, 385}}</string>
            <reference key="NSSuperview" ref="474857837"/>
          </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

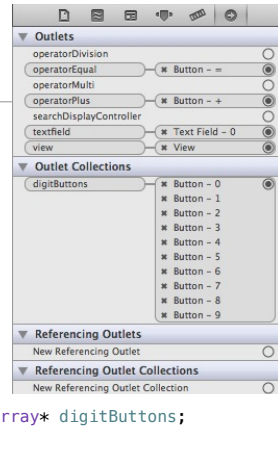


Connecting Widgets and Code

- 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

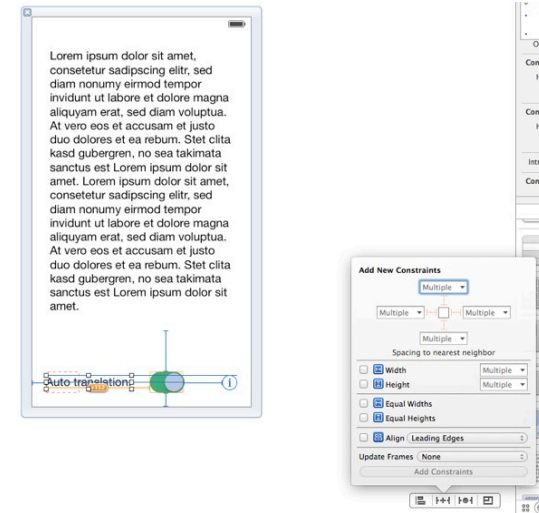
Auto Layout

- 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
 - $<=$, $=$, $=>$
- 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

bonding
STUDENTENINITIATIVE E.V.

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
bonding – erlebe, was du werden kannst.

KOSTENLOS von Studenten für Studenten

Tag der Informatik
06.12.
Informatikzentrum | Altonstr. 35 | Aachen

NEUER 17th BEGRÜßUNG 15th VORSTELLUNG
PROFESSOREN 14th ONE MINUTE MADNESS 15th HAUT-
VORTRAG 16th FIRMEN: ONE MINUTE MADNESS
WWW.TDI.AC

21st PARTY
17th ABSCHLUSSEMPFANG 18th DINNER
KITTSPIELN
Popcorn ist im
Wichtig!
Wall
Geduldlos
ist immer da!
Lernen ist die
4th Edition
des
Party mit
01 Stunden

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

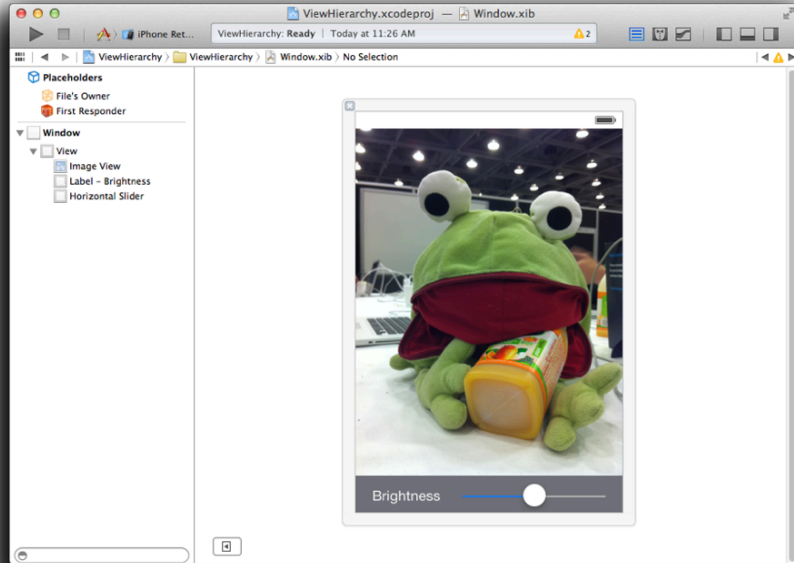
- ✓ SDK provide many types of Views to show your content
- ✓ At run-time Views are organized as a tree
- ✓ Geometry of Views are determined by constraints
- ✓ Use Interface Builder to design your UI and connect it to code

View Programming

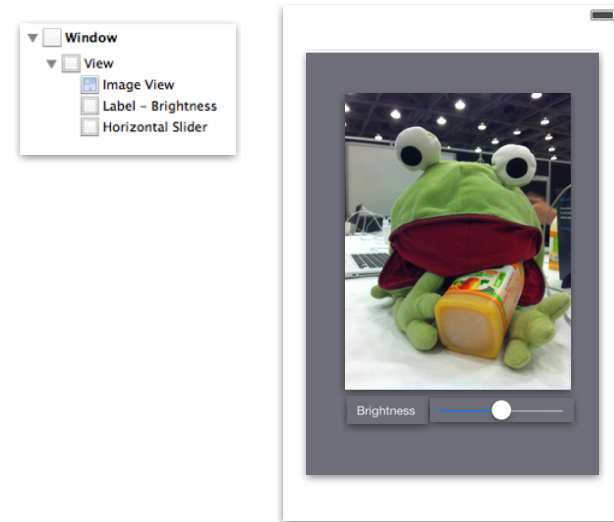
- Defines a rectangular area on the screen
- Two responsibilities
 - Render content
 - React to user input
 - Manage subviews
- Layout as view hierarchy

UIView

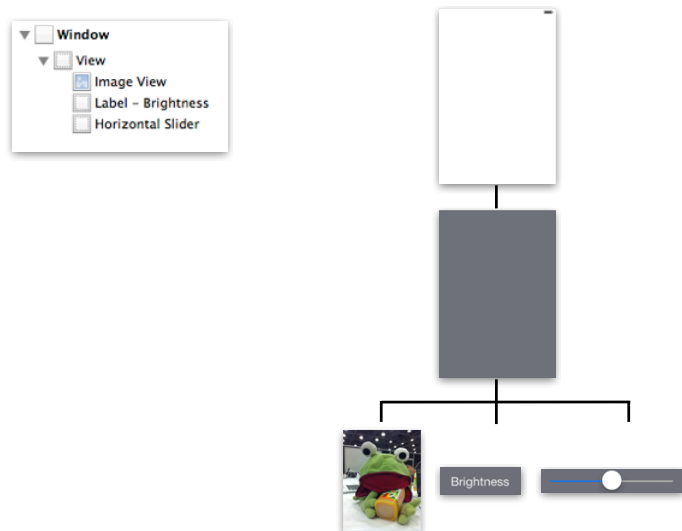
View Hierarchies



View Hierarchies

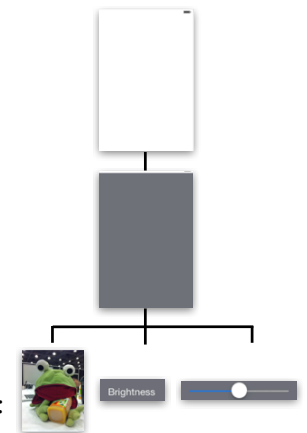


View Hierarchies



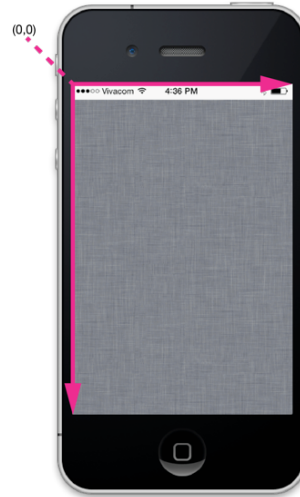
Managing Views

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



View Coordinate System

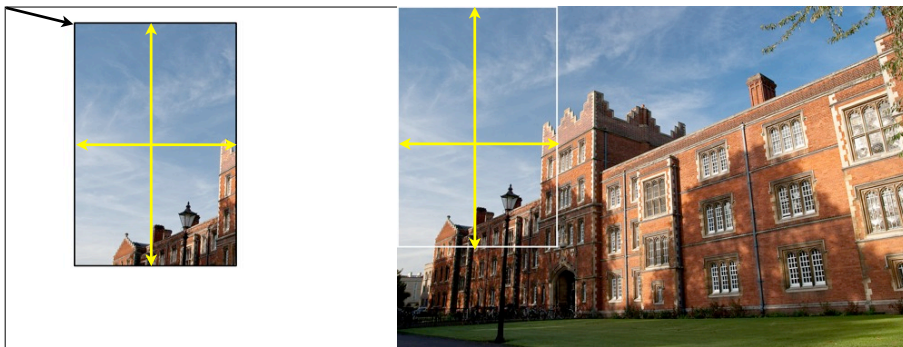
- Origin at top-left corner
- Coordinates use points as unit
 - iPhone: 320x480
 - iPhone 5: 320x568
 - iPad: 768x1024



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)

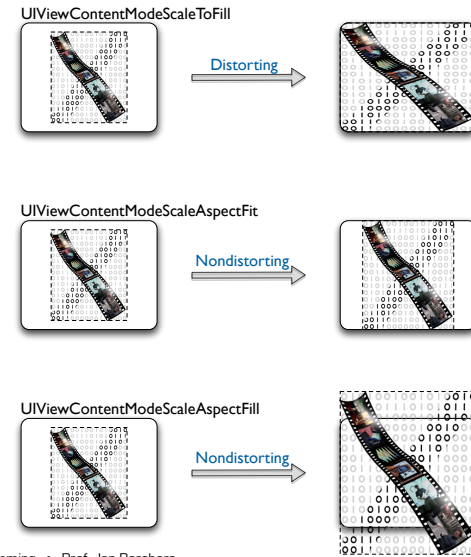
Setting Values

- 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



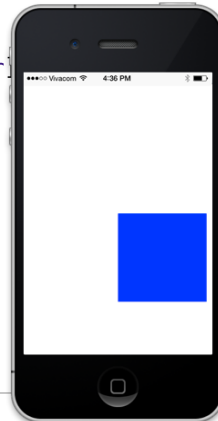
Content Mode



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



- 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