

iOS Development at Cohabitat

Brent Nycum
brentnycum@gmail.com

- Protocols
 - Delegates
 - Data Sources
- Views
- View Controllers

Protocols

Protocols

- Similar to interface but without implementation
 - Have a .h file but not .m

Protocols

```
@protocol Foo  
  
- (void)method;  
  
@end
```

Protocols

- Classes must implement our protocols
 - Claim they implement it in the header file, `@interface`
- `@interface MyClass : NSObject <Foo>`
- Implement methods in our `@implementation`
 - `-(void)method;`

Protocols

- Must implement all non-optional methods
- Methods can be marked required or optional
 - `@required`
 - `@optional`

Protocols

- Top uses of protocols in Objective-C
 - Delegates
 - Data Sources

Delegates

Delegation Pattern

- A design pattern where an object delegates that task to an associated helper object

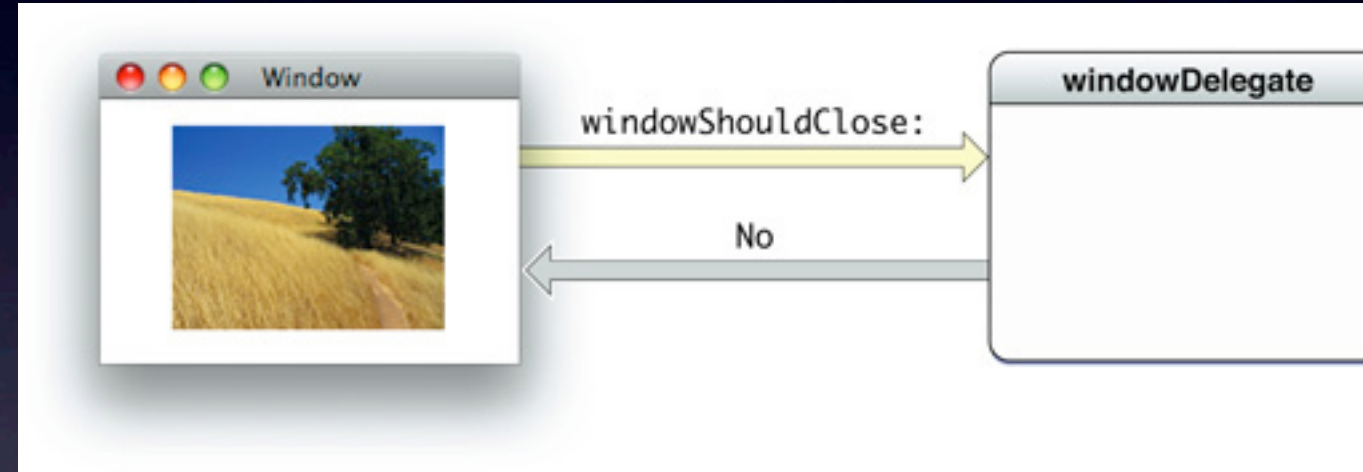
Delegates

- Allow an object to send messages to another object when an event happens
- Callbacks
- Delegation is very common in the Cocoa framework

Delegates

- Handle the should, did events

Delegates



Window asks delegate can it close

Delegates

- 3 Examples
 - UITextField
 - UIApplication
 - NSURLConnection

UITextField

- Control that displays editable text area
- Seen this in a bunch of places in iOS
 - Search bars
 - Input fields

UITextFieldDelegate

- Delegate handler for UITextField
- Some Methods
 - – (BOOL)textFieldShouldBeginEditing:(UITextField *)
textField;
 - – (void)textFieldDidBeginEditing:(UITextField *)
textField;

UITextFieldDelegate

- – (BOOL)textFieldShouldBeginEditing:(UITextField *)
textField;
- UITextField will ask if we should begin editing
 - Basically bring focus to text field and display keyboard
- Can return NO and not allow editing

UITextFieldDelegate

- – `(void)textFieldDidBeginEditing:(UITextField *)textField;`
- Sends a notice to our delegate handler that did start editing
 - Got focus and keyboard was displayed

UIApplicationDelegate

- One of the main jobs of the application delegate is to track the state transitions the application goes through while it is running
- Prior to iOS 4.0, applications were either active, inactive, or not running
- In iOS 4.0 and later, applications can also be running in the background or suspended

UIApplicationDelegate

- All of these transitions require a response from your application to ensure that it is doing the right thing
- For example, a background application would need to stop updating its user interface
- You provide the response to these transitions using the methods of the application delegate.

UIApplicationDelegate

- – (void)application:didFinishLaunchingWithOptions:(NSDictionary *)launchOptions;
- – (void)applicationWillResignActive:
- – (void)applicationDidBecomeActive:
- – (void)applicationDidEnterBackground:
- – (void)applicationWillEnterForeground:
- – (BOOL)application:handleOpenURL:(NSURL *)url;
- – (void)applicationDidReceiveMemoryWarning:
- – (void)applicationWillTerminate:

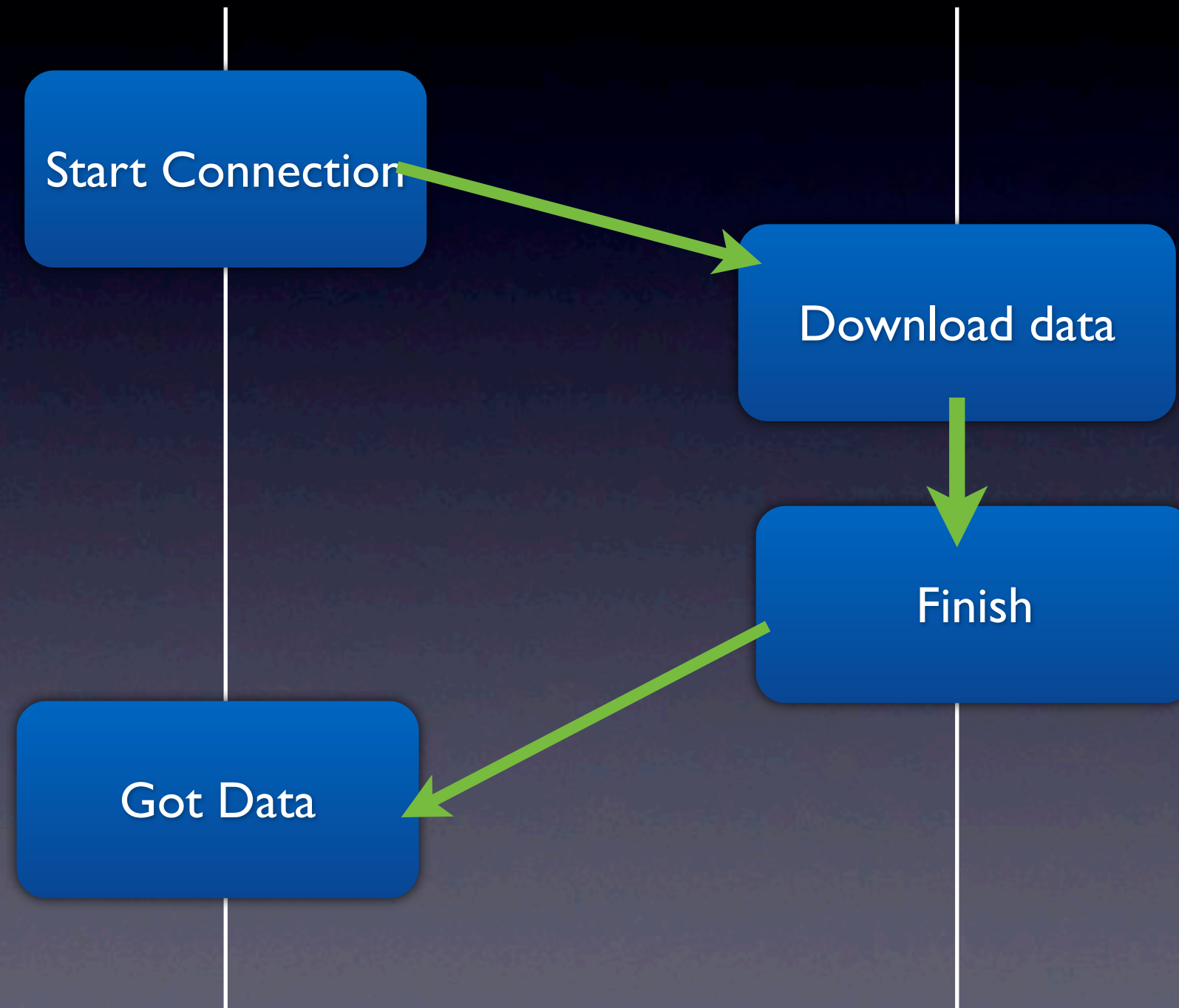
NSURLConnection

- Provides support to perform the loading of a URL request
 - Download a file
- Can be done
 - Synchronously
 - Asynchronously

NSURLConnection

- Synchronously
 - `+ (NSData *)sendSynchronousRequest:(NSURLRequest *)request
returningResponse:(NSURLResponse **)response error:(NSError **)error`

NSURLConnection



NSURLConnection

- Synchronous version is quicker to program but....
- Locks the main thread
 - Can't interact with app while locked up
 - Timeouts can be up to 60 seconds
 - Large downloads can take forever

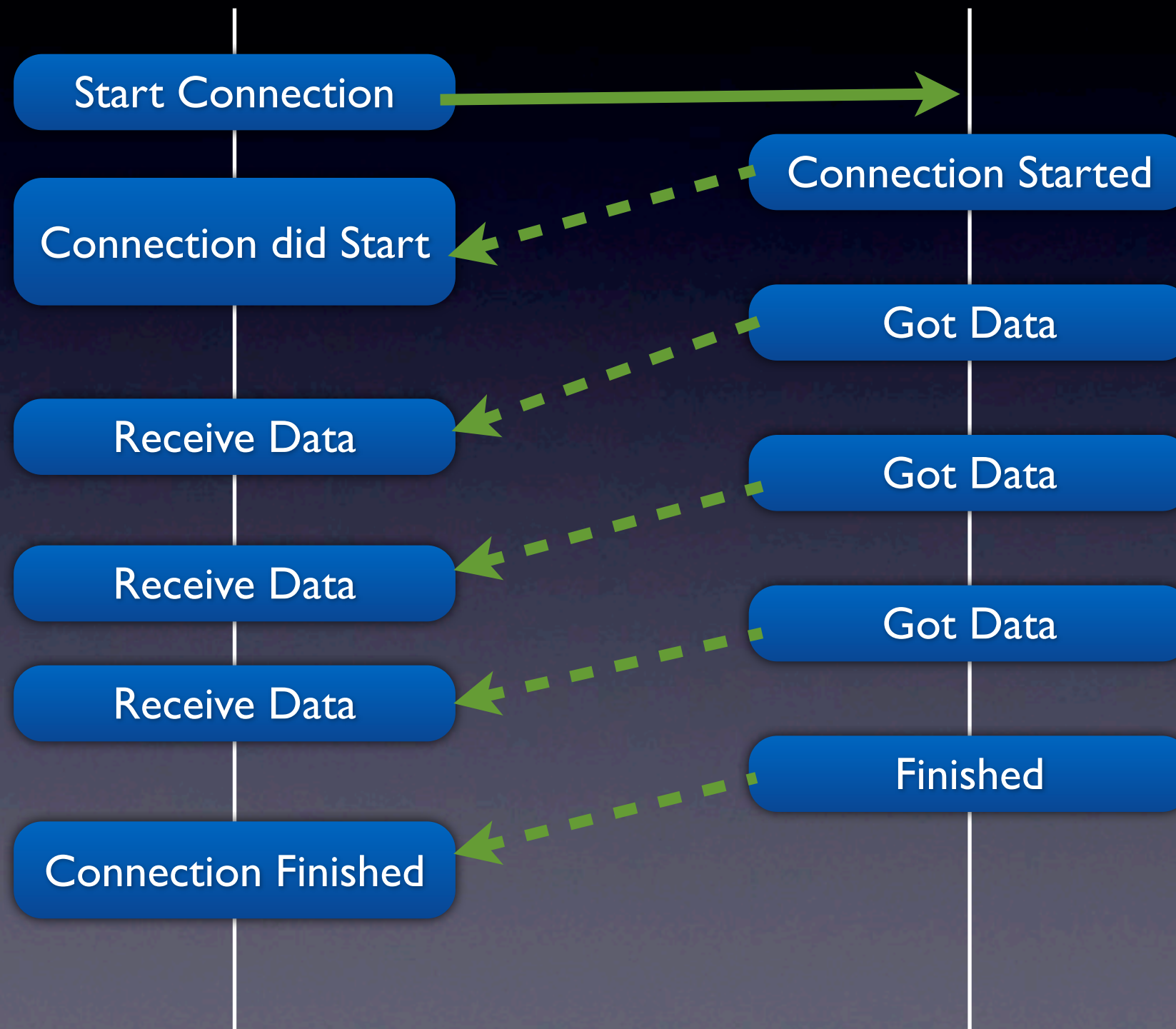
NSURLConnection

- Asynchronous version
 - Works on separate thread, runs in background
 - We get better updates of our connection
 - Have to handle more in code, but worth it

NSURLConnection

- – (void)connection:(NSURLConnection *)
connection didReceiveResponse:(NSURLResponse *)
response
- – (void)connection:(NSURLConnection *)
connection didReceiveData:(NSData *)data
- – (void)connectionDidFinishLoading:
(NSURLConnection *)connection

NSURLConnection



Data Sources

Data Sources

- Popular use UITableView / UITableViewDataSource
- Provides the table-view with the information it needs to construct and modify the table view
- Doesn't handle when the table was tapped, that's the job of the UITableViewDelegate

UITableViewDataSource

- Provides the table view with things like:
 - Number of Sections
 - Number of Rows in each Section
 - Titles for each Section
 - Footer for each Section

Views

Views

- Base class UIView
- Sliders, buttons, labels, etc are all Views
- Normally constructed in Interface Builder
 - Even custom views

Views

- Can also be done in code
 - `-(void)addSubview:(UIView *)aView;`
 - `-(void)removeFromSubview;`
- Managing order of subviews
 - `-(void)insertSubview:(UIView*) aView atIndex:(int) index;`
 - Also `belowSubview:` and `aboveSubview:`

Coordinates

- CGPoint
- Struct with 2 CGFloats: X and Y
 - CGFloat typedef for float or double
- CGPoint p = CGPointMake(72.5, 34.0);
- p.x += 20;

Size

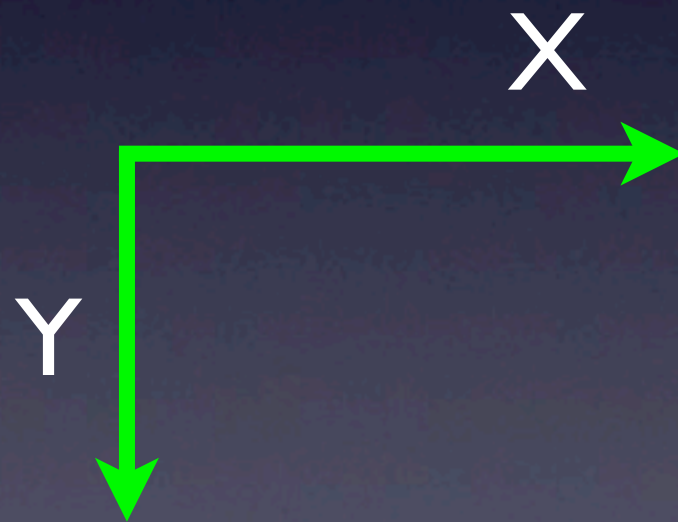
- CGSize
- CGSize s = CGSizeMake(300.0, 240.0);
- s.width += 20;

Rect

- Point & Size
- CGRect
 - C Struct with CGPoint (origin) and CGSize (size)
 - CGRect aRect = CGRectMake(15, 25, 320, 240);
 - X,Y,Width, Height
 - aRect.size.height += 15;
 - aRect.origin.x += 12;

Views

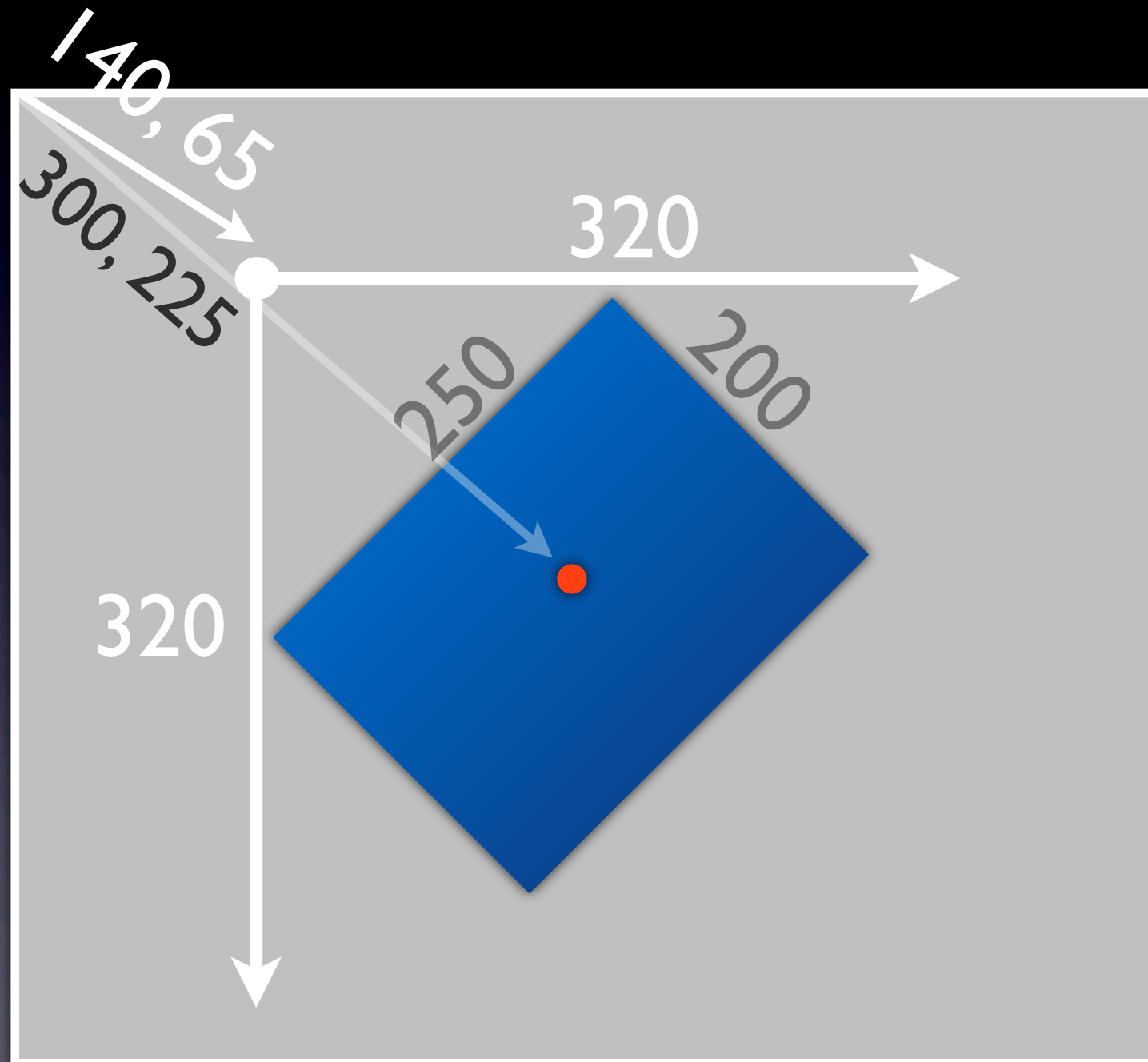
- Origin of a view's coordinate system is upper left
- Units are points, not pixels



Views

- bounds
 - Views internal drawing space
- center
 - center of your view in superview's coordinate space
- frame
 - Rectangle which contains your view's bounds.size
 - frame.size != bounds.size;

Coordinates



Bounds = (0, 0, 200, 250)
Frame = (140, 65, 320, 320)
Center = (300, 225)

Views are rarely rotated

View Controllers

View Controller

- Base class: UIViewController
- Controller of MVC
- Have a lifecycle from creation to destruction

View Controller

- Life Cycle starts with alloc and init
- – `(id)initWithNibName:(NSString *)nibName bundle:(NSBundle *)aBundle;`
- Designated Initializer
- Loads view from .xib file called nibName
- If nibName is nil, attempts to use the name of the class
- NSBundle allows you to specify for localization

View Controller

- `-(void)viewDidLoad;`
 - After view controller is initialized
 - Good place for setup code
 - Geometry of your view isn't set

View Controller

- `-(void)viewWillAppear:(BOOL)animated;`
- View Geometry has been set here
- View will load once, but appear and disappear multiple times

View Controller

- `-(void)viewWillAppear:(BOOL)animated;`
- Good place to save data
 - Nothing too time-consuming or app becomes sluggish
- Maybe remember scroll location

View Controller

- Also “did” versions
 - `-(void)viewDidAppear:(BOOL)animated;`
 - `-(void)viewDidDisappear:(BOOL)animated;`

View Controller

- Handles device rotation
 - - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)anOrientation
- Default is only portrait

View Controller

```
- (BOOL)shouldAutorotateToInterfaceOrientation:  
(UIInterfaceOrientation)anOrientation {  
  
    return (anOrientation == UIInterfaceOrientationPortrait) ||  
        (anOrientation == UIInterfaceOrientationPortraitUpsideDown);  
  
}
```

Also `UIInterfaceOrientationLandscapeLeft` and `Right`

View Controller

- UINavigationController
 - Hierarchical flow

View Controller

- UINavigationController
 - Group of independent views selected using tabs at the bottom of the screen

View Controller

- UISplitViewController
 - Side by side or master>detail
 - iPad only

Xcode Template

- What's going on?
- What are all these files?

Xcode Template

- This example is based on us using the standard “View-based Application” Template
 - Named: FirstApp

What Files are Created?

- main.m
- FirstAppViewController.[h|m]
- FirstAppViewController.xib
- MainWindow.xib
- FirstApp-Info.plist
- FirstAppAppDelegate.[h|m]

main.m

- Basic c entry point function
 - `int main(int argc, char *argv[])`
- Starts the run loop
- Calls UIApplicationMain and creates UIApplication object
- Creates autorelease pool
- Normally never modify this

MainWindow.xib

- Contains our UIWindow
 - Top of the view hierarchy for our apps
- Customizable per platform (iPhone / iPad)
- Contains the Application Delegate

FirstAppAppDelegate

- Handles applicationDid and applicationWill
 - Launched, Active, Background, Phone Calls, Text Messages
- application: didFinishLaunchingWithOptions:
 - Where view is added to the window
 - Window made visible

FirstApp-Info.plist

- Application Configuration
 - Icon Locations
 - Supported Interface Orientations
 - Version Number

FirstAppViewController.xib

- Interface Builder file
- Contains default UIView
 - grey screen
- The screen we see when running the application
- Where we drag our controls

FirstAppViewController.[h|m]

- Interface and Implementation files
- Where we wire up our connections
 - Targets and Actions

Interface Builder

Interface Builder

- Need to know about 2 things
 - IBOutlet
 - IBActions
- Both are macros to indicate these variables or methods can be referred to by Interface Builder

IBOutlet

- References an object
- Our outlets would be our sliders, buttons, etc

IBAction

- Special method that is triggered by a UI object
 - Tapping a button
 - Sliding a slider

Questions

Next Week

- More on Controllers
 - UINavigationController
 - UITabViewController
- Table Views