

Level 1

App Anatomy



In this level we'll cover



01 Starting an iPhone project

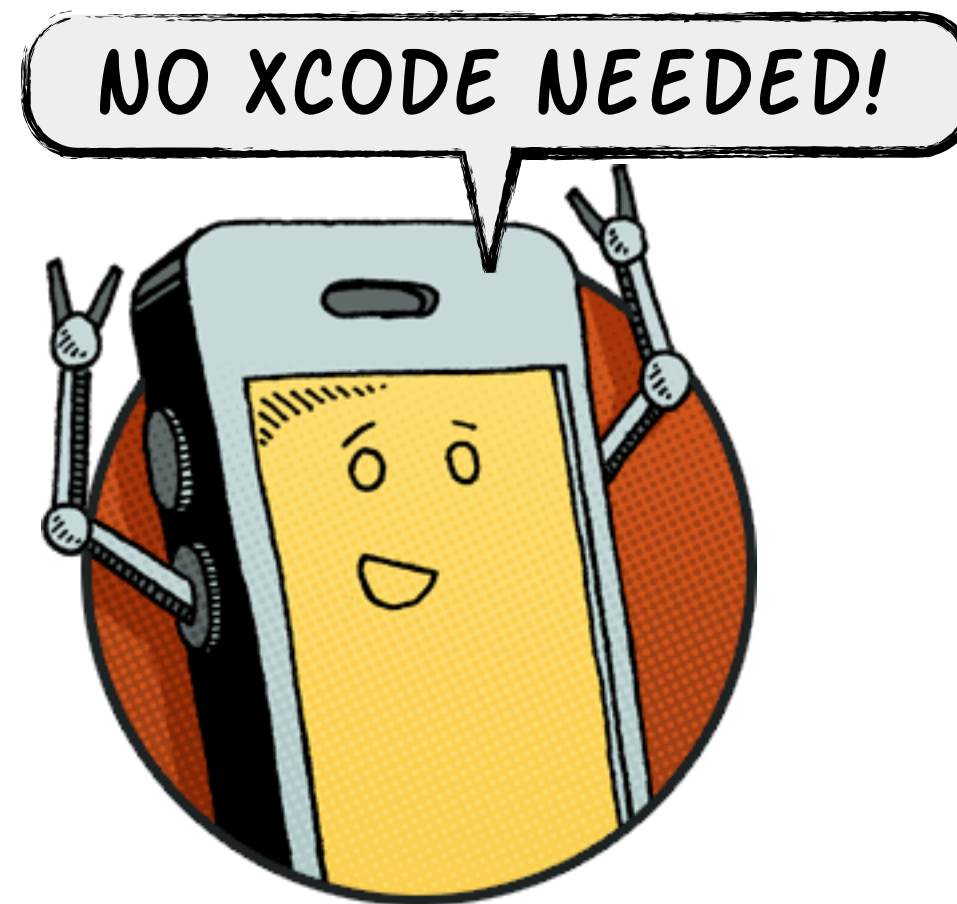
02 The Application Delegate

03 Your First iPhone App

04 How to Draw

05 Refactoring Our Code

06 Touch



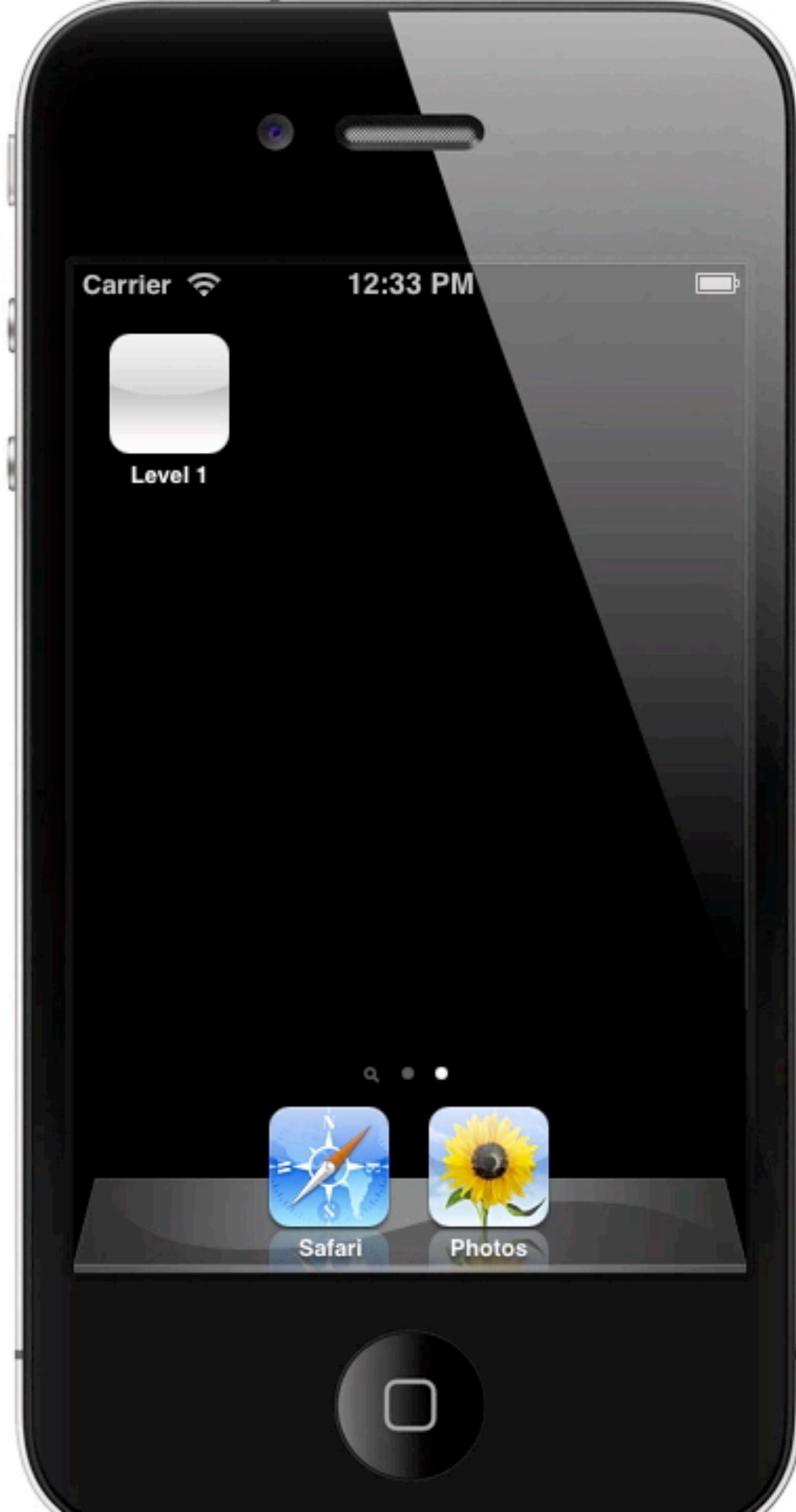
MR. HIGGLE

Your First iOS Application



Set a background color

Click to change the transparency





Choose a template for your new project



iOS

Application

Framework & Library

Other

OS X

Application

Framework & Library

Application Plug-in

System Plug-in

Other

Master-Detail Application

OpenGL Game

Page-Based Application

Single View Application

Tabbed Application

Utility Application

Empty Application

1

Single View Application

This template provides a starting point for an application that uses a single view. It provides a view controller to manage the view, and a storyboard or nib file that contains the view.

Cancel

Previous

Next

Choose options for your new project:



Product Name

Organization Name

Company Identifier

Bundle Identifier

Class Prefix

Devices

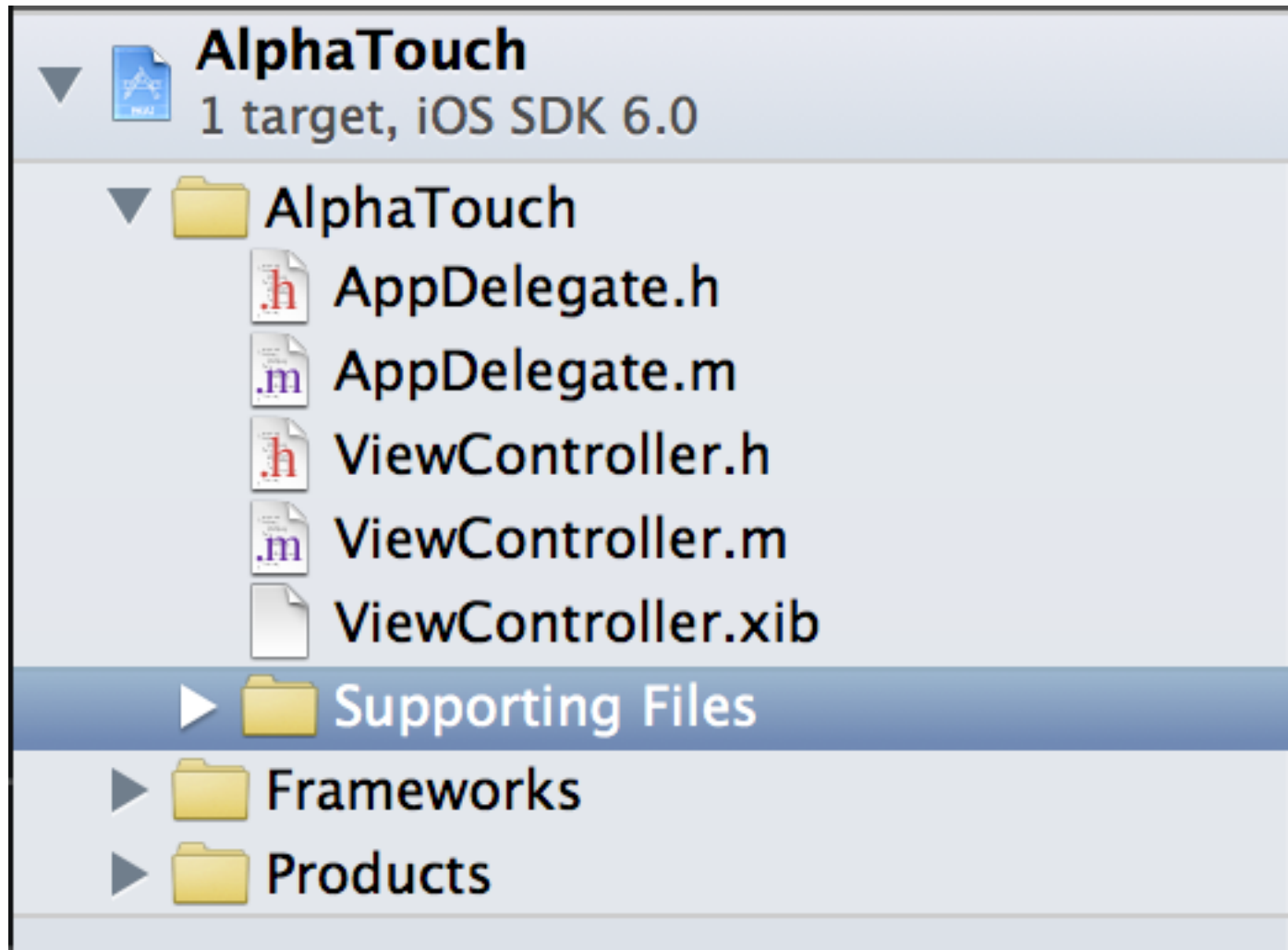
- ☐ Use Storyboards
- ☒ Use Automatic Reference Counting
- ☐ Include Unit Tests

Cancel

Previous

Next

Creates these files for you



Defines methods that will be called at different points in your applications lifecycle.

`application:didFinishLaunchingWithOptions:`

`applicationWillResignActive:`

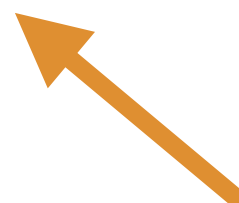
`applicationDidEnterBackground:`

`applicationWillEnterForeground:`

`applicationDidBecomeActive:`

`applicationWillTerminate:`

We only care about this





Gets called when application is launched

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
```

This is small, let's go two lines

```
- (BOOL)        application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
```

Method Name

Parameter type

Parameter names

application:didFinishLaunchingWithOptions:



Gets called when application is launched

```
- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // your code here
    return YES;
}
```

To create log messages

```
NSLog(@"Hello World");
```



String

What does an artist need to paint?



Canvas



Window

Paint Brush



ViewController

Paint



View

What does an iOS app need to draw something?

Find the dimensions of the screen



AppDelegate.m

```
- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    UIScreen *screen = [UIScreen mainScreen]; Get the main screen
    CGRect viewRect = [screen bounds]; Get the bounds of screen

    NSLog(@"Screen is %f tall and %f wide",
          viewRect.size.height, viewRect.size.width);

    return YES;
}
```

2012-10-03 21:11:59.012 AlphaTouch[44703:c07] Screen is 480.000000 tall and 320.000000 wide

DO WE REALLY NEED THE SCREEN VARIABLE?



Find the dimensions of the screen



AppDelegate.m

```
- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    CGRect viewRect = [[UIScreen mainScreen] bounds];

    NSLog(@"Screen is %f tall and %f wide",
          viewRect.size.height, viewRect.size.width);

    return YES;
}
```

2012-10-03 21:11:59.012 AlphaTouch[44703:c07] Screen is 480.000000 tall and 320.000000 wide

Create our UIWindow (canvas)

AppDelegate.m

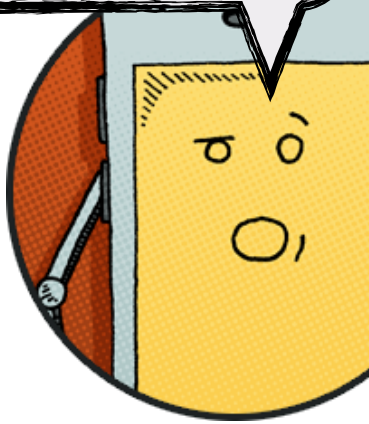
```
- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    CGRect viewRect = [[UIScreen mainScreen] bounds];

    UIWindow *window = [[UIWindow alloc] initWithFrame:viewRect];

    NSLog(@"Screen is %f tall and %f wide",
          viewRect.size.height, viewRect.size.width);

    return YES;
}
```

THIS IS ALREADY A PROPERTY!



Allocate memory for a UI Window
and initialize object with frame size to the bounds of the main screen

Create our UIWindow (canvas)




AppDelegate.h

```
@property (strong, nonatomic) UIWindow *window; ~instance variable
```

AppDelegate.m

```
- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    CGRect viewRect = [[UIScreen mainScreen] bounds];

     self.window = [[UIWindow alloc] initWithFrame:viewRect];


    NSLog(@"Screen is %f tall and %f wide",
          viewRect.size.height, viewRect.size.width);

    return YES;
}
```

Tell the UIWindow it's the Key Window & Visible



AppDelegate.m

```
- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    CGRect viewRect = [[UIScreen mainScreen] bounds];
     self.window = [[UIWindow alloc] initWithFrame:viewRect];
    [self.window makeKeyAndVisible];
    Means it should receive all keyboard & non-touch events
    NSLog(@"Screen is %f tall and %f wide",
          viewRect.size.height, viewRect.size.width);
    return YES;
}
```

2012-10-03 21:50:31.019 AlphaTouch[45090:c07] Application windows are expected to have a root view controller at the end of application launch





Try
iOS

Create the ViewController



AppDelegate.m

```
- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
     CGRect viewRect = [[UIScreen mainScreen] bounds];
    self.window = [[UIWindow alloc] initWithFrame:viewRect];

     UIViewController *colorTouchVC = [[UIViewController alloc] init];
    self.window.rootViewController = colorTouchVC;




    [self.window makeKeyAndVisible];
    NSLog(@"Screen is %f tall and %f wide",
          viewRect.size.height, viewRect.size.width);
    return YES;
}
```


This ViewController gets control of the window

Create the View



AppDelegate.m

```
 CGRect viewRect = [[UIScreen mainScreen] bounds];  
self.window = [[UIWindow alloc] initWithFrame:viewRect];  
  
 UIViewController *colorTouchVC = [[UIViewController alloc] init];  
  
 UIView *colorView = [[UIView alloc] initWithFrame:viewRect];  
colorTouchVC.view = colorView;  
  
self.window.rootViewController = colorTouchVC;  
[self.window makeKeyAndVisible];  
NSLog(@"Screen is %f tall and %f wide",  
      viewRect.size.height, viewRect.size.width);
```







Create a view the size of the whole screen

Set the background color



AppDelegate.m

```
 CGRect viewRect = [[UIScreen mainScreen] bounds];  
 self.window = [[UIWindow alloc] initWithFrame:viewRect];  
  
 UIViewController *colorTouchVC = [[UIViewController alloc] init];  
  
 UIView *colorView = [[UIView alloc] initWithFrame:viewRect];  
colorView.backgroundColor = [UIColor yellowColor];  
colorTouchVC.view = colorView;  
  
self.window.rootViewController = colorTouchVC;  
[self.window makeKeyAndVisible];  
NSLog(@"Screen is %f tall and %f wide",  
      viewRect.size.height, viewRect.size.width);
```

Look up the UIColor class to change the color

Lets Review



AppDelegate.m



```
CGRect viewRect = [[UIScreen mainScreen] bounds];  
self.window = [[UIWindow alloc] initWithFrame:viewRect];
```



```
UIViewController *colorTouchVC = [[UIViewController alloc] init];
```



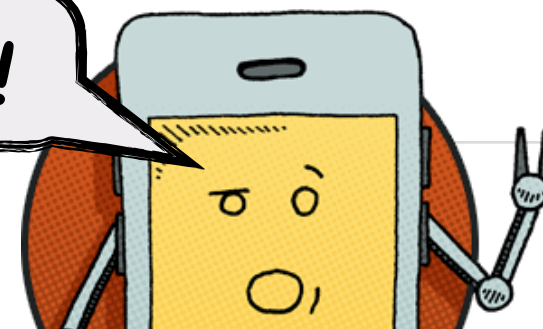
```
UIView *colorView = [[UIView alloc] initWithFrame:viewRect];  
colorView.backgroundColor = [UIColor yellowColor];  
colorTouchVC.view = colorView;
```

```
self.window.rootViewController = colorTouchVC;
```

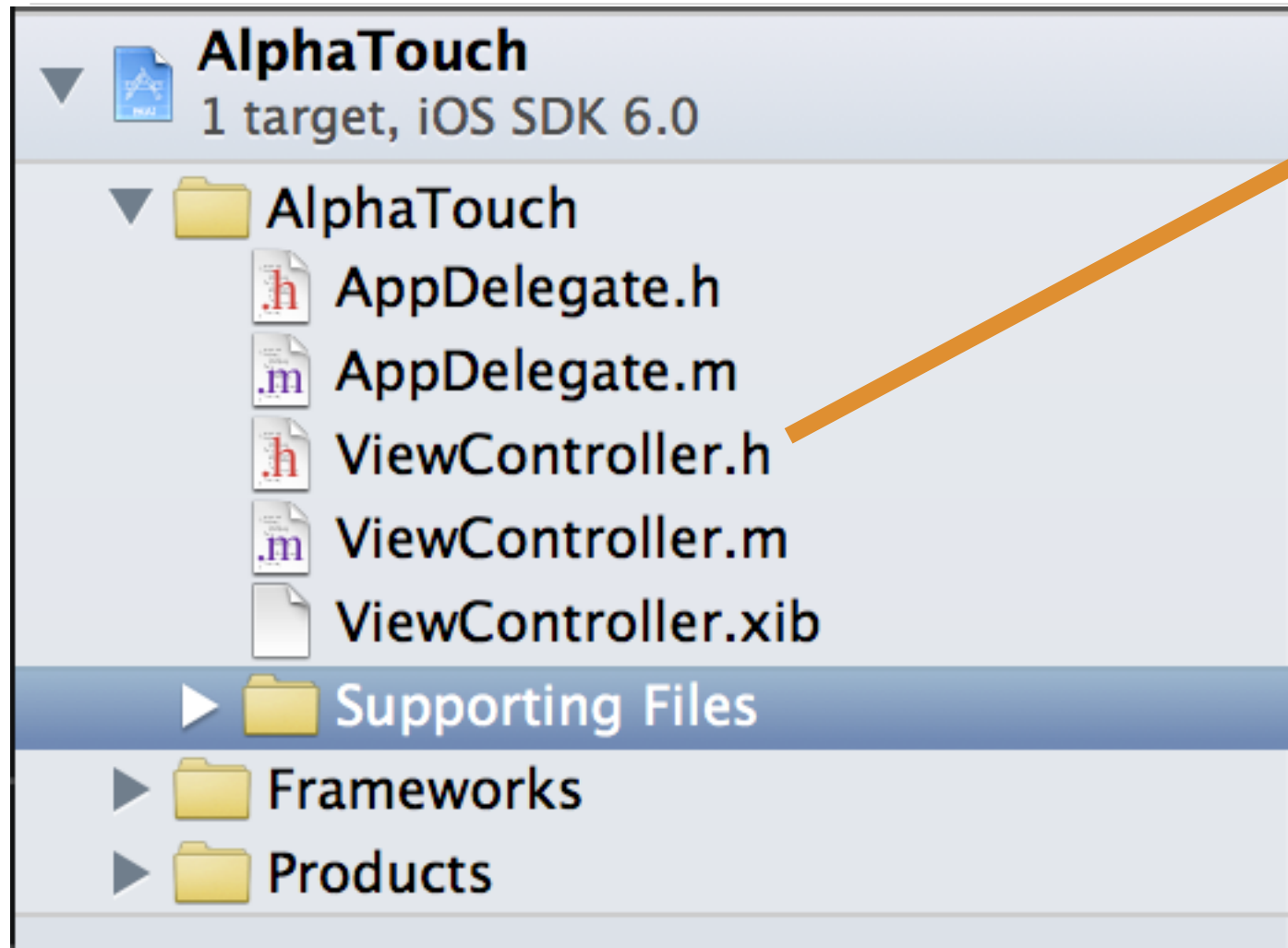
```
[self.window makeKeyAndVisible];
```

```
NSLog(@"Screen is %f tall and %f wide",  
      viewRect.size.height, viewRect.size.width);
```

THIS CODE NEEDS TO BE REFACTORED!



We already had a ViewController



ViewController.h

```
#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

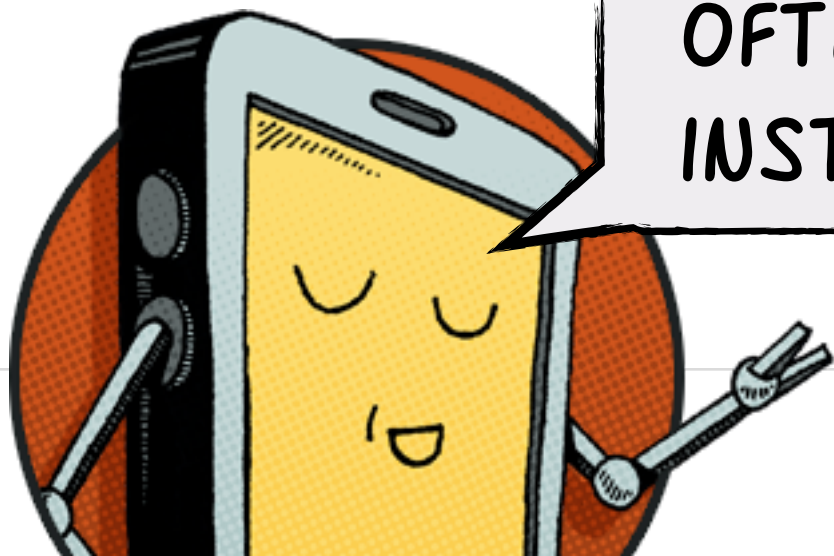
@end
```



Subclass name

Parent Class

OFTEN WE SAY SUPERCLASS
INSTEAD OF PARENT CLASS



Need to reference the ViewController



ViewController.h

```
#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

@end
```



AppDelegate.h

```
@class ViewController; Forward Declaration
@property (strong, nonatomic) ViewController *viewController;
```



AppDelegate.m

```
#import "ViewController.h" Includes the class
```

Need to reference the ViewController





AppDelegate.m

old code

```
 CGRect viewRect = [[UIScreen mainScreen] bounds];  
self.window = [[UIWindow alloc] initWithFrame:viewRect];
```



```
 UIViewController *viewController = [[UIViewController alloc] init];
```

```
 UIView *view = [[UIView alloc] initWithFrame:viewRect];  
view.backgroundColor = [UIColor yellowColor];  
viewController.view = view;  
self.window.rootViewController = viewController;
```

```
...
```


Need to reference the ViewController



AppDelegate.m

```
 CGRect viewRect = [[UIScreen mainScreen] bounds];  
self.window = [[UIWindow alloc] initWithFrame:viewRect];
```



```
 self.viewController = [[ViewController alloc] init];
```

```
 UIView *view = [[UIView alloc] initWithFrame:viewRect];  
view.backgroundColor = [UIColor yellowColor];
```

```
self.viewController.view = view;  
self.window.rootViewController = self.viewController;
```

```
...
```

Our app delegate shouldn't know about view!




AppDelegate.m

 `CGRect viewRect = [[UIScreen mainScreen] bounds];
self.window = [[UIWindow alloc] initWithFrame:viewRect];`



 `self.viewController = [[ViewController alloc] init];`

 `UIView *view = [[UIView alloc] initWithFrame:viewRect];
view.backgroundColor = [UIColor yellowColor];
self.viewController.view = view;
self.window.rootViewController = self.viewController;`

...

Need to move this into ViewController.m

Getting familiar with ViewController



ViewController.m



```
- (void)viewDidLoad After loadView, typically where labels/buttons go
{
    [super viewDidLoad];
    self.view.backgroundColor = [UIColor yellowColor];
}
- (void)loadView Called first time view property accessed
{
    CGRect viewRect = [[UIScreen mainScreen] bounds];
    UIImageView *view = [[UIImageView alloc] initWithFrame:viewRect];
    self.view = view;
}
```



With the View removed from the ViewController



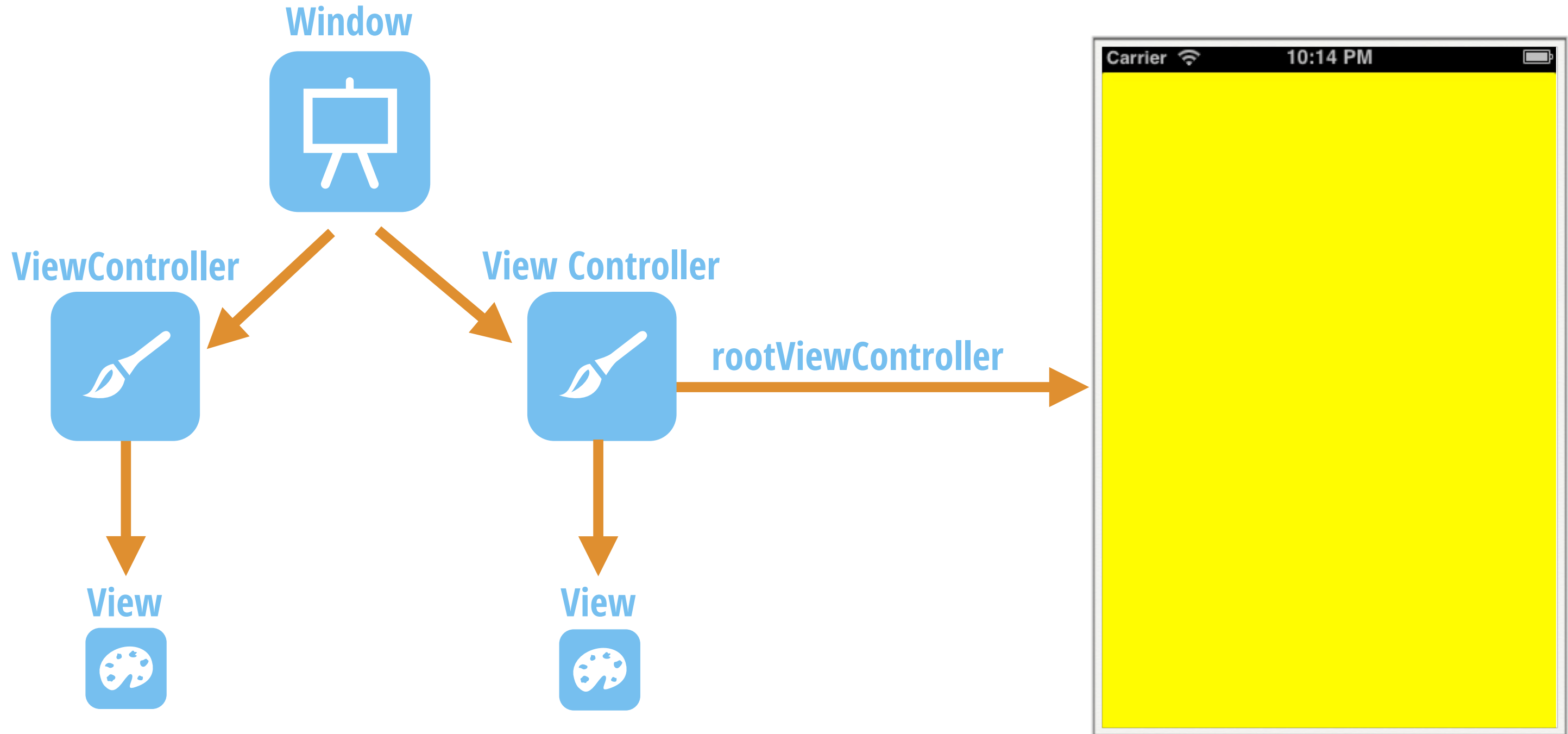
AppDelegate.m

```
 CGRect viewRect = [[UIScreen mainScreen] bounds];  
self.window = [[UIWindow alloc] initWithFrame:viewRect];  
  
 self.viewController = [[ViewController alloc] init];  
self.window.rootViewController = self.viewController;  
[self.window makeKeyAndVisible];
```



MUCH BETTER!
IT'S NOT GOOD TO HAVE
VIEWS IN YOUR APPDELEGATE.

App Anatomy

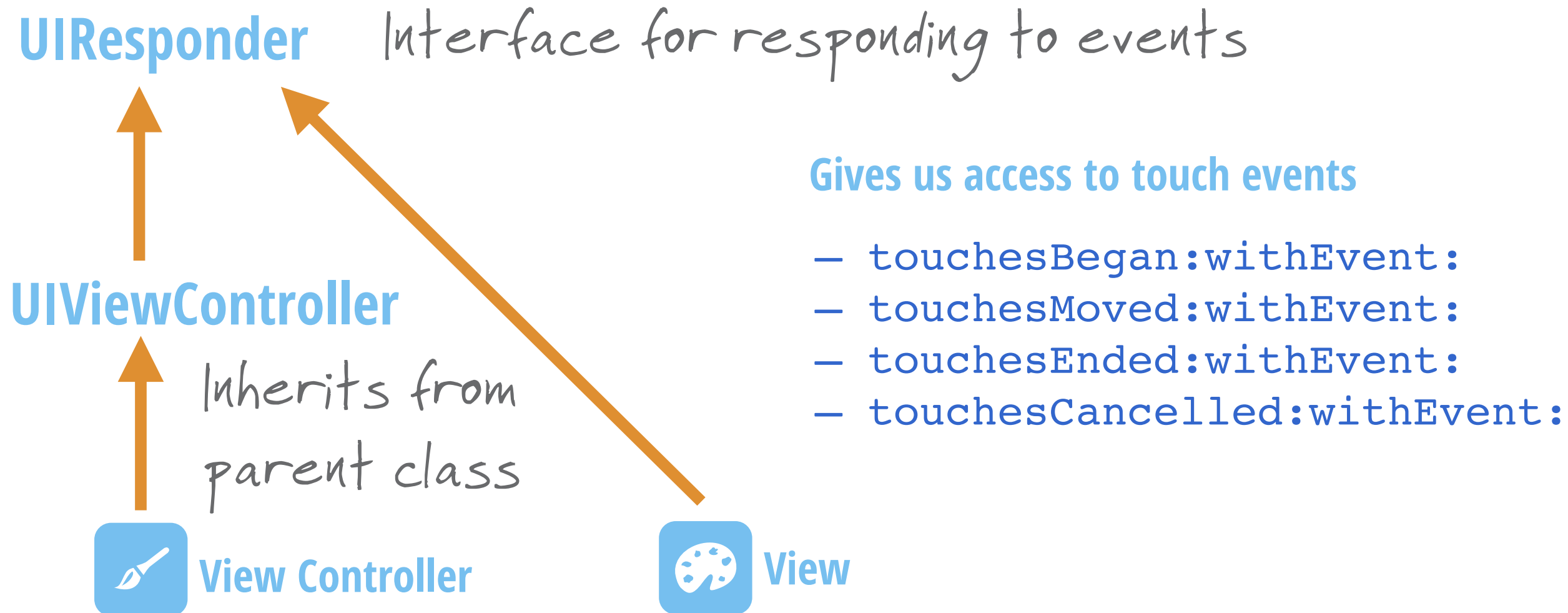


Next objective



Print log message whenever touched

```
NSLog(@"Started touching the screen");
```



Print log message whenever touched



ViewController.m

```
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
{
    NSLog(@"Started touching the screen");
}
```