

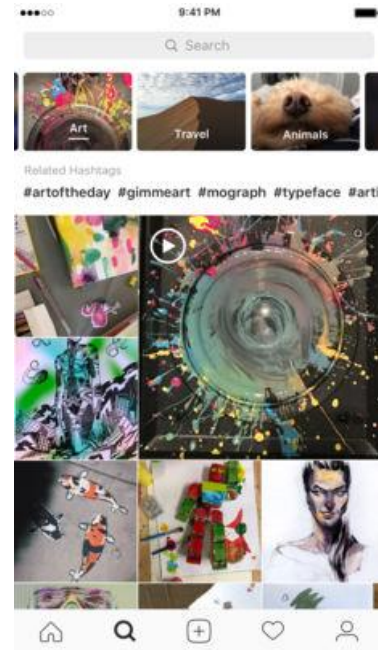
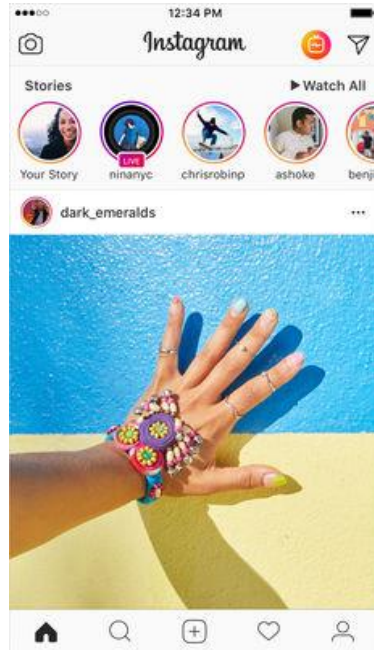


Creating a Multiview App

Ani, Colin, Elisa, Jesica

Improving over Single-View Apps

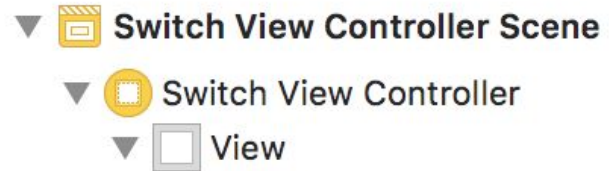
The real power of iOS emerges when you switch out views based on user input





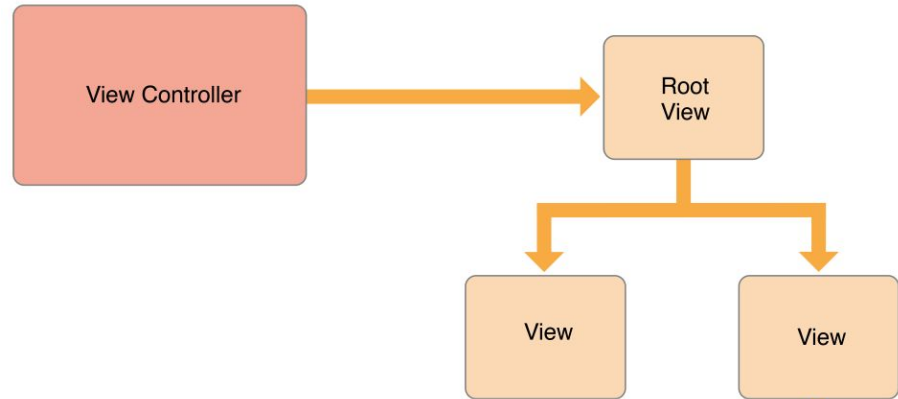
Views, Controllers and Scenes

- Views are the primary container for app content and UI
- Each view has a corresponding view controller which manages it
- The pairing of a view and a view controller is called a scene



The Root Controller

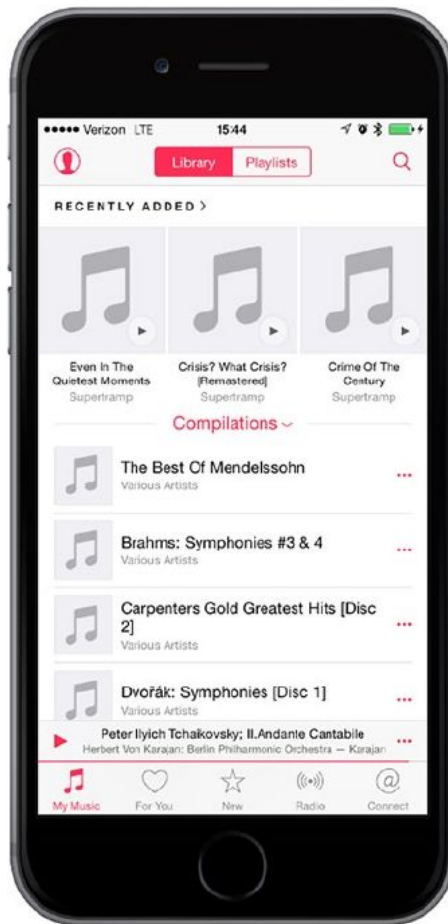
- First controller the user sees, loaded when the app loads
- Takes two or more other views and presents them to the user as appropriate based on user input

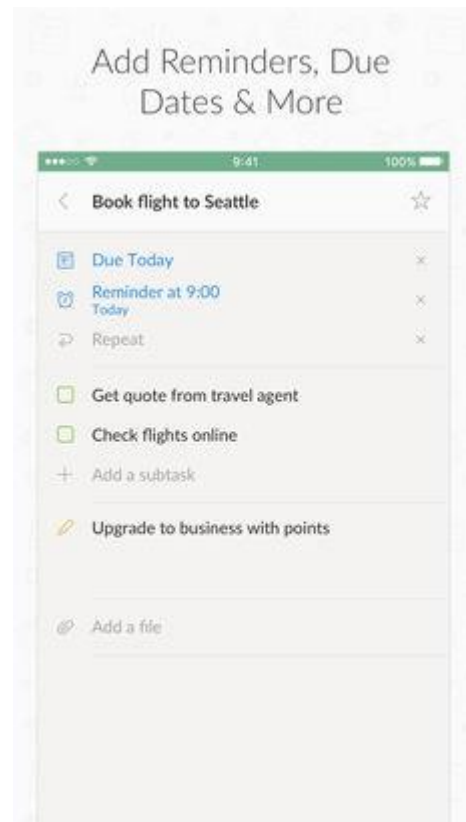
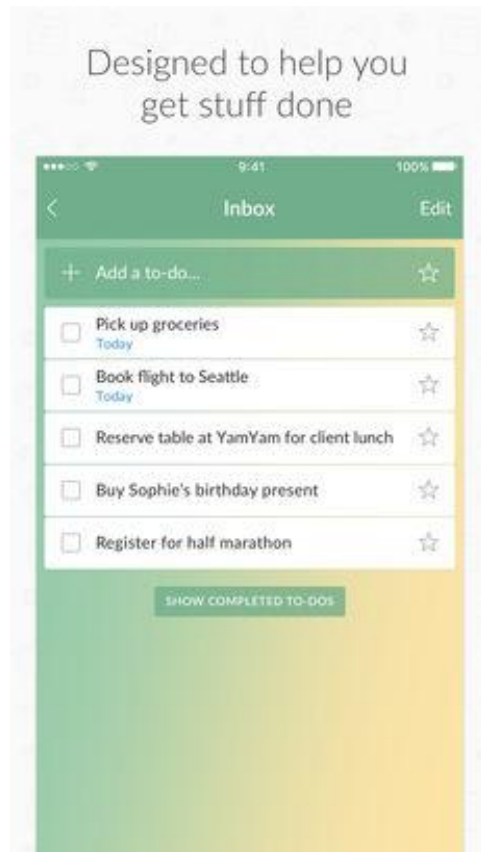




Swapping Views

- Several controllers are available for use









Let's make an app!

The link to the code and pictures for this project is below:

<https://goo.gl/EA3ucd>

Step 1

Create a new Project in Xcode.

Choose options for your new project:

Product Name:

Team:

Organization Name:

Organization Identifier:

Bundle Identifier:

Language:

☐ Use Core Data

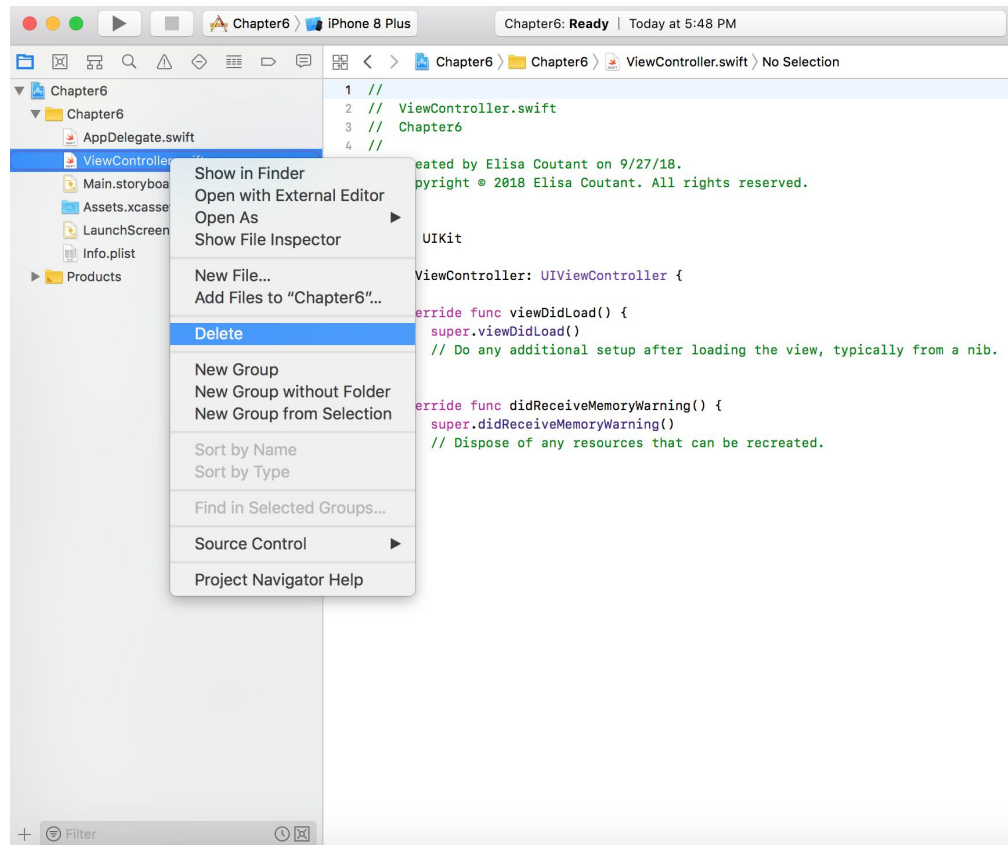
☐ Include Unit Tests

☐ Include UI Tests

Step 2

Right click and Delete the ViewController.

Note-we are deleting and creating a new view controller because 'ViewController' is referenced in multiple places in the app, and 'Refactoring' is not supported by Swift in XCode beta



Step 3

- Create a new File
- Select Cocoa Touch Class from the iOS Source section
- Name it 'SwitchViewController'
- Make it a subclass of UIViewController

Choose options for your new file:

Class:

Subclass of:

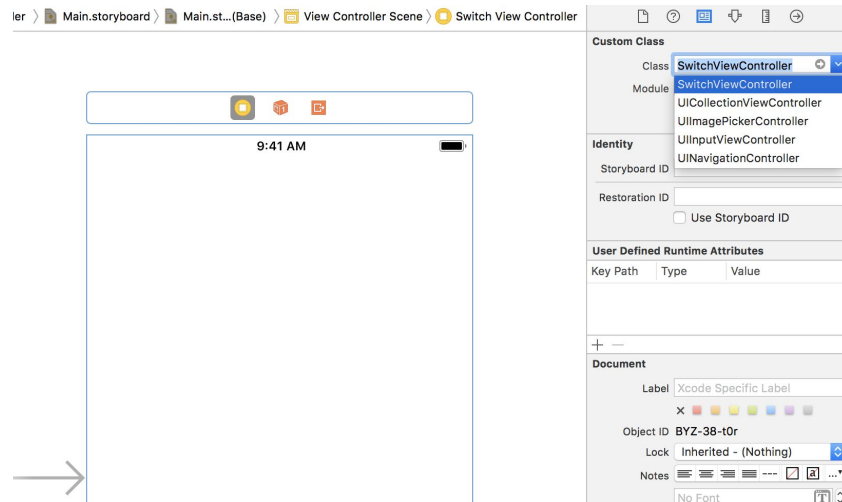
☐ Also create XIB file

Language:

Step 4

Now need to link the View with the Main.storyboard

- Select Main.storyboard from outline.
- Select the view controller from outline and open the inspector.
- Change class from UIViewController to “SwitchViewController”.





Step 5

- Add two additional view controllers
 - New -> File -> CocoaTouch Class -> subclass of UIViewController
-> Name them : OrangeViewController and GreenViewController
- Modify SwitchViewController by adding action method to change views

```
import UIKit

class SwitchViewController: UIViewController {

    → private var greenViewController: GreenViewController!
       private var orangeViewController: OrangeViewController!

    override func viewDidLoad() {
        super.viewDidLoad()

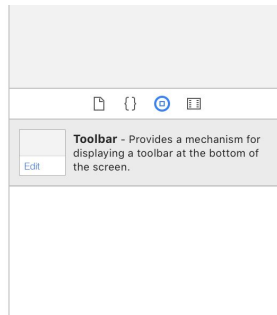
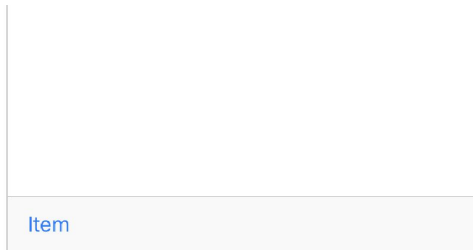
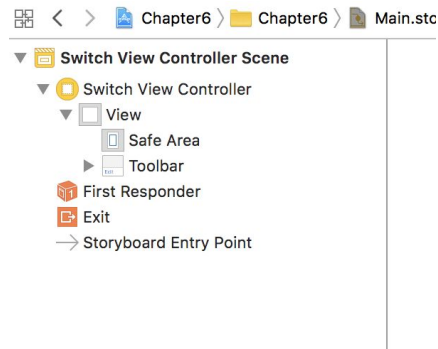
        // Do any additional setup after loading the view.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    → @IBAction func switchViews(sender: UIBarButtonItem){
       }
}
```

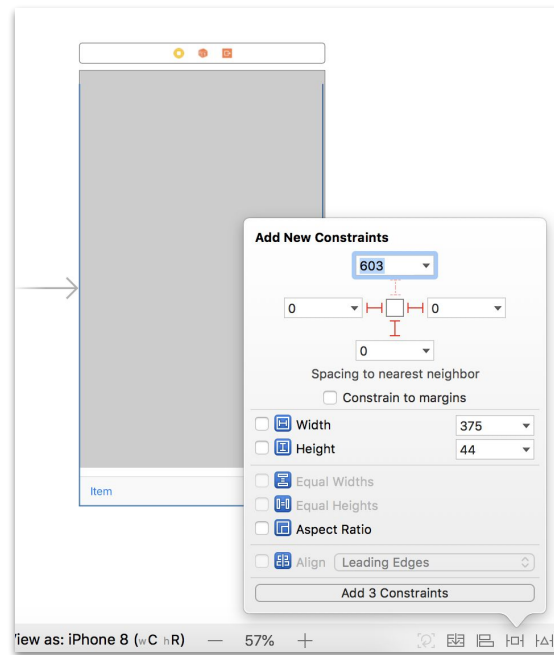
Step 6

- Building the SwitchViewController
 - Select Main.storyboard, we'll start building our GUI here.
 - Search for “toolbar” on the bottom right, and add it at the bottom of your view.



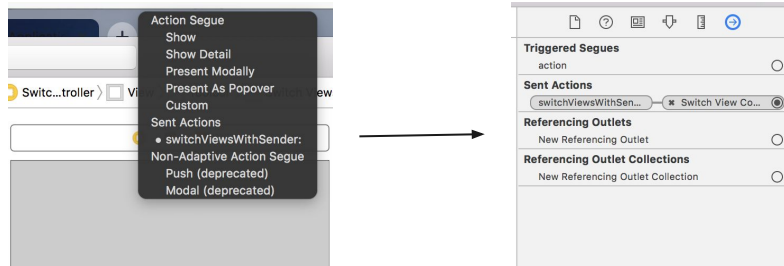
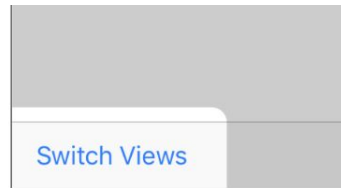
Step 7

- Add constraints to keep the toolbar at the bottom and centered.
 - Make sure “Constrain to margins” is not clicked.
 - Set distances to left, right and bottom to zero.
 - Click the three dashed red lines that link the distance boxes to the center. (they should turn into solid lines)
 - Change the update frame to “Items of new constraints” and add 3 constraints.



Step 8

- Link the ViewController and the Toolbar button.
 - Change the title of the button to “Switch View”
 - Link toolbar button to action method in SwitchViewController
 - Click SwitchView button and control drag it over the yellow switching viewController icon
 - Select the switchViewsWithSender: action.





Step 9

- Write the root view controller implementation.
 - Navigate to the SwitchViewController.swift file
 - Add the following code to viewDidLoad()

```
override func viewDidLoad() {  
    super.viewDidLoad()  
    // Do any additional setup after loading the view.  
    greenViewController =  
        storyboard?.instantiateViewController(withIdentifier: "Green")  
        as! GreenViewController  
    greenViewController.view.frame = view.frame  
    switchViewController(from: nil, to: greenViewController) // helper method  
}
```

*Note the compiler will give you an error because we have not yet written the helper method “switchViewController”.



Step 10

- Fill in the switchViews() method with the following code:

```
@IBAction func switchViews(sender: UIBarButtonItem) {  
    // Create the new view controller, if required  
    if orangeViewController?.view.superview == nil {  
        if orangeViewController == nil {  
            orangeViewController =  
                storyboard?.instantiateViewController(withIdentifier: "Orange")  
                as! OrangeViewController  
        }  
    } else if greenViewController?.view.superview == nil {  
        if greenViewController == nil {  
            greenViewController =  
                storyboard?.instantiateViewController(withIdentifier: "Green")  
                as! GreenViewController  
        }  
    }  
    // Switch view controllers  
    if greenViewController != nil  
        && greenViewController!.view.superview != nil {  
        orangeViewController.view.frame = view.frame  
        switchViewController(from: greenViewController,  
                             to: orangeViewController)  
    } else {  
        greenViewController.view.frame = view.frame  
        switchViewController(from: orangeViewController,  
                             to: greenViewController)  
    }  
}
```

Use of unresolved identifier 'switchViewController'

Use of unresolved identifier 'switchViewController'



Step 11

- Add the following code to safely release unneeded controllers during low memory conditions

```
override func didReceiveMemoryWarning() {  
    super.didReceiveMemoryWarning()  
    // Dispose of any resources that can be recreated.  
    if greenViewController != nil  
        && greenViewController!.view.superview == nil {  
        greenViewController = nil  
    }  
    if orangeViewController != nil  
        && orangeViewController!.view.superview == nil {  
        orangeViewController = nil  
    }  
}
```



Step 12

- Add the switchViewController() helper method.
 - Add the following code to your SwitchViewController.swift file

```
private func switchViewController(from fromVC:UIViewController?, to toVC:UIViewController?) {  
    if fromVC != nil {  
        fromVC!.willMove(toParentViewController: nil)  
        fromVC!.view.removeFromSuperview()  
        fromVC!.removeFromParentViewController()  
    }  
    if toVC != nil {  
        self.addChildViewController(toVC!)  
        self.view.insertSubview(toVC!.view, at: 0)  
        toVC!.didMove(toParentViewController: self)  
    }  
}
```



Step 13

- We are now implementing the Content Views for the green VC and the orange VC.
 - Add the following code to GreenViewController.swift:

```
@IBAction func greenButtonPressed(sender: UIButton) {  
    let alert = UIAlertController(title: "Green View Button Pressed",  
                                message: "You pressed the button on the green view",  
                                preferredStyle: .alert)  
  
    let action = UIAlertAction(title: "Yes, I did", style: .default,  
                              handler: nil)  
    alert.addAction(action)  
    present(alert, animated: true, completion: nil)  
}
```



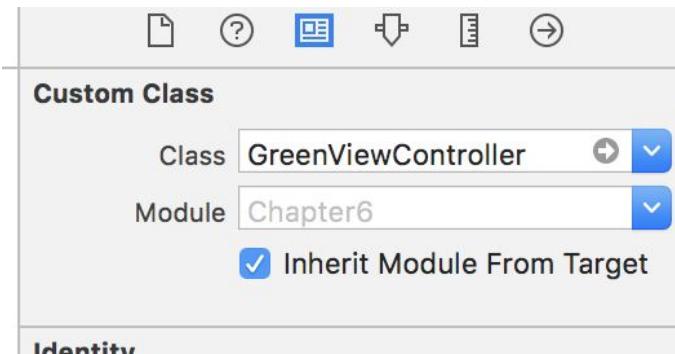
Step 13 (continuation)

- Add the following code to the OrangeViewController.swift

```
@IBAction func orangeButtonPressed(sender: UIButton) {  
    let alert = UIAlertController(title: "Orange View Button Pressed",  
                                message: "You pressed the button on the orange view",  
                                preferredStyle: .alert)  
    let action = UIAlertAction(title: "Yes, I did", style: .default,  
                              handler: nil)  
    alert.addAction(action)  
    present(alert, animated: true, completion: nil)  
}
```

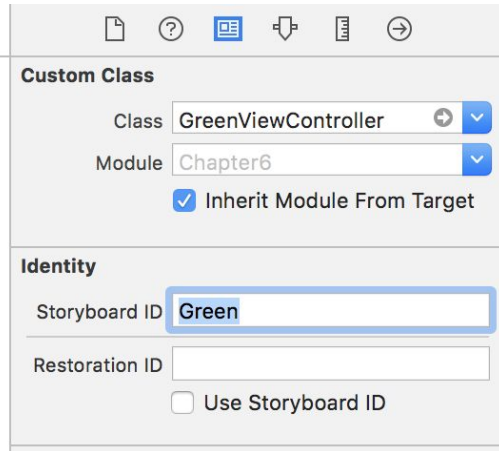
Step 14

- Open Main.storyboard.
- Add new scene for GreenViewController
 - Search for “ViewController” on the bottom right
 - Drag a new view controller from the object library to the editing area.
 - Click on the new VC and pick the identity inspector icon from the top right menu.
 - In the custom class section change the class menu to GreenViewController.



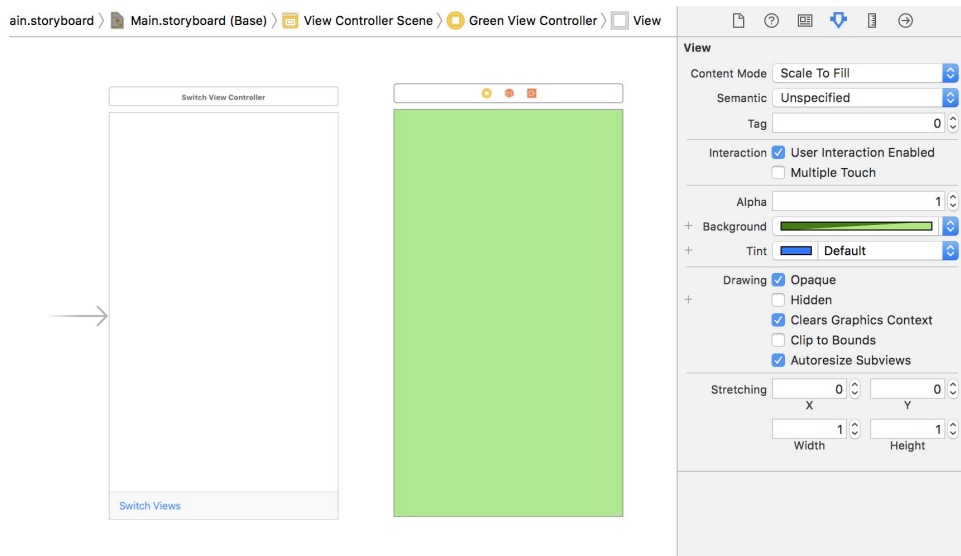
Step 15

- Create identifier for new VC.
 - In the identify inspector, change the Storyboard ID to “Green” to match your code.



Step 16

- Add background color to your new view
 - Click on your view and select the Attributes Inspector icon.
 - Pick green as your new background color.



Step 17

- Drag a button from the library onto your view.
- Use the guidelines to center it in the view.
- Add necessary constraints to make sure it stays centered no matter what.
- Change the color of the text to white and the text to “Press me”.
- Select the “Press me” button and control click on it and drag from it to icon at the top.
- Connect to the greenButtonPressedWithSender: action.





Step 18

- Repeat the steps for the orange view controller
- Change the button text for the orange VC to “Press Me, Too”





Step 19

We are now going to animate the transition between views

- Open SwitchViewController.swift
- Add the following changes to switchViews() method
- Build and run

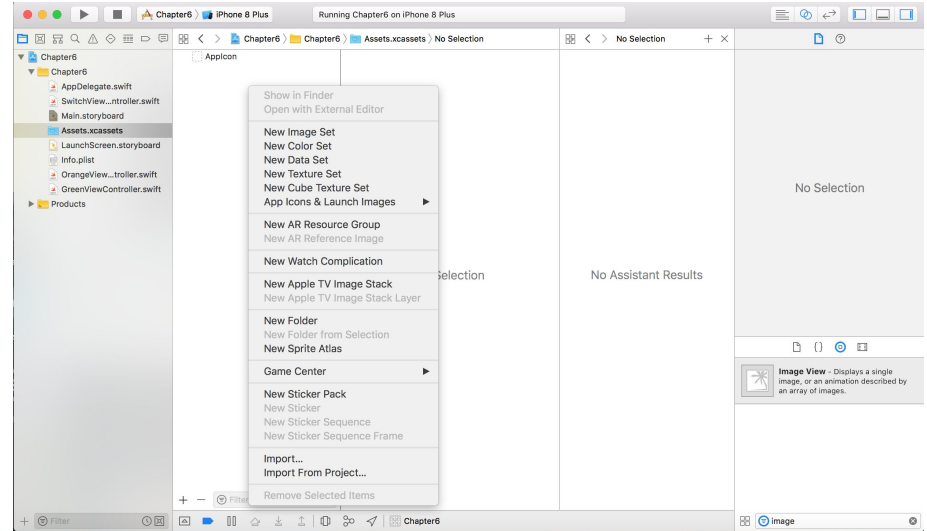


```
UIView.beginAnimations("View Flip", context: nil)
UIView.setAnimationDuration(0.4)
UIView.setAnimationCurve(.easeInOut)
// Switch view controllers
if greenViewController != nil
    && greenViewController!.view.superview != nil {
    orangeViewController.view.frame = view.frame
    switchViewController(from: greenViewController,
                        to: orangeViewController)
} else {
    greenViewController.view.frame = view.frame
    switchViewController(from: orangeViewController,
                        to: greenViewController)
}
UIView.commitAnimations()
}
```

Step 20:

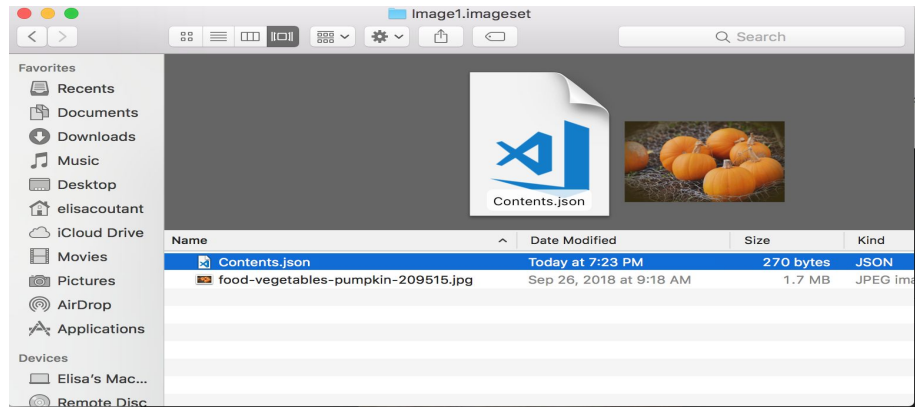
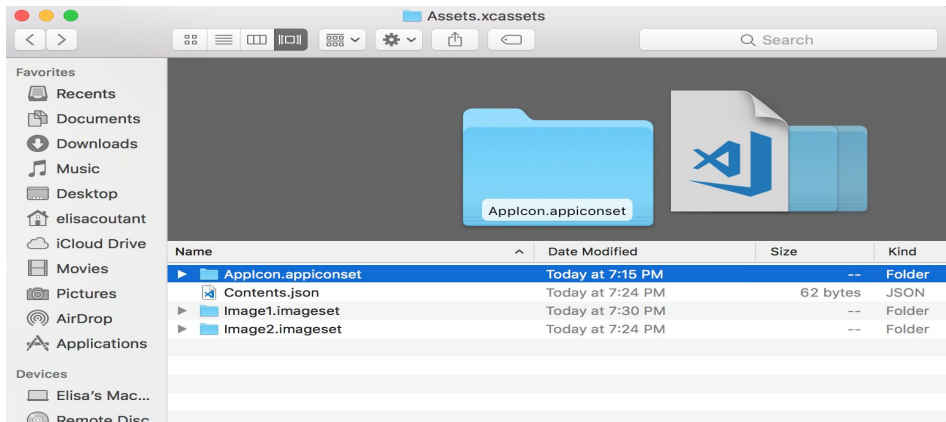
We are now going to add pictures to the background

- Select the Assets folder from the Document View.
- Right click to create an image set.
- Create two new Image sets, and name them Image1 and Image2



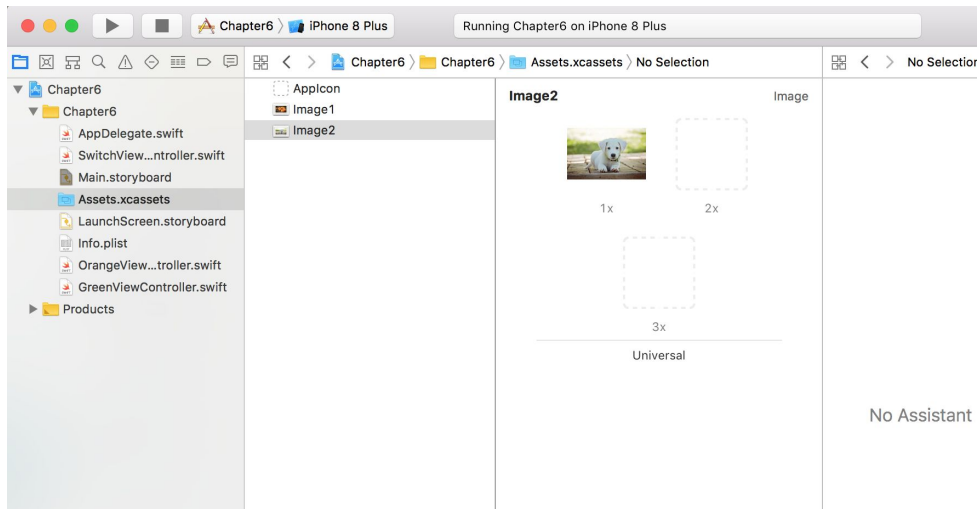
Step 21

- Use any two images of your choice
- Drag and drop one image in each Image folder



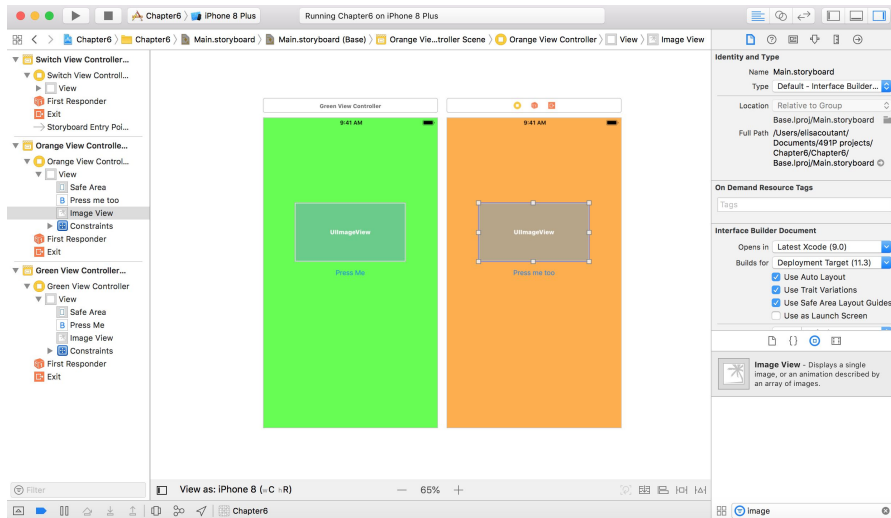
Step 22

- Drag and drop the image into the icon



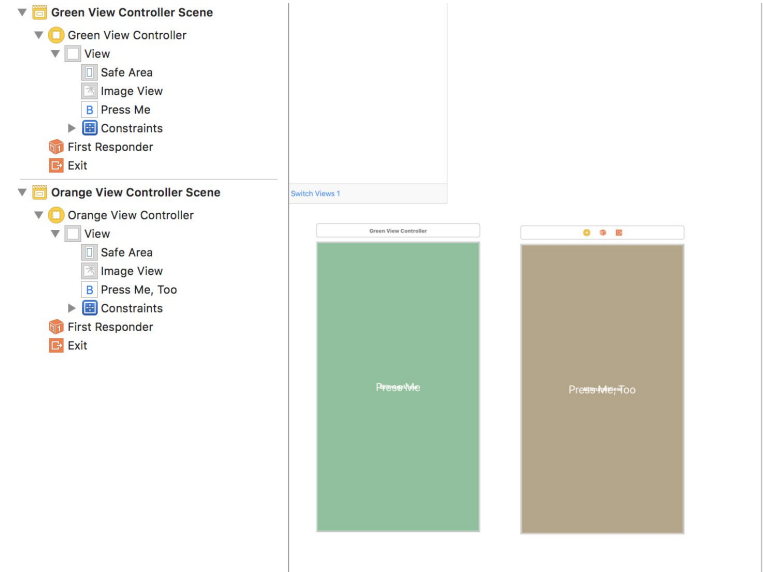
Step 23

- Drag and drop an Image View into the green and orange views.



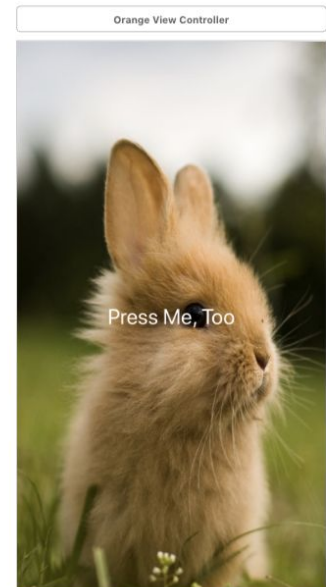
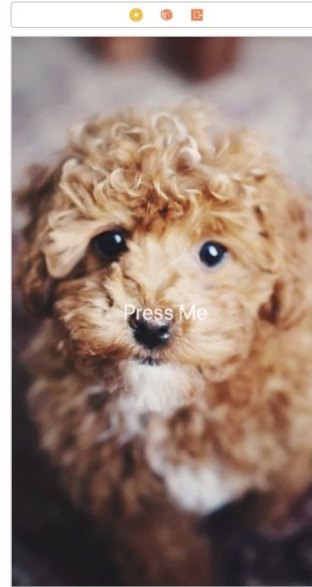
Step 24

- Set image views to be the size of the orange and green view controllers.
- Make sure that in the order of the scene objects on the left navigator, the image views are above the Press Me and the Press Me, Too buttons.
- Set the necessary constraints on the image views so they are centered vertically and horizontally.



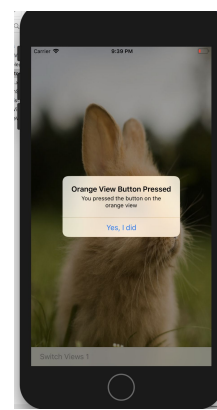
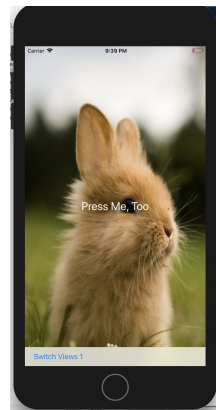
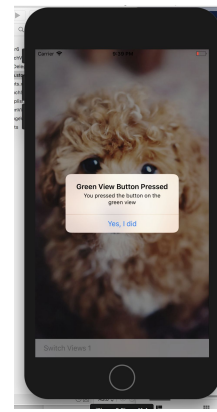
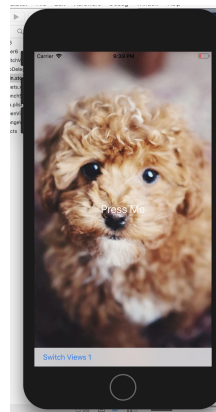
Step 25

- Select each image view
 - In the attributes inspector, select the image to be shown.



Step 26

- Run the simulator to make sure that everything works correctly.





Link to these slides

<https://goo.gl/zuzKtV>