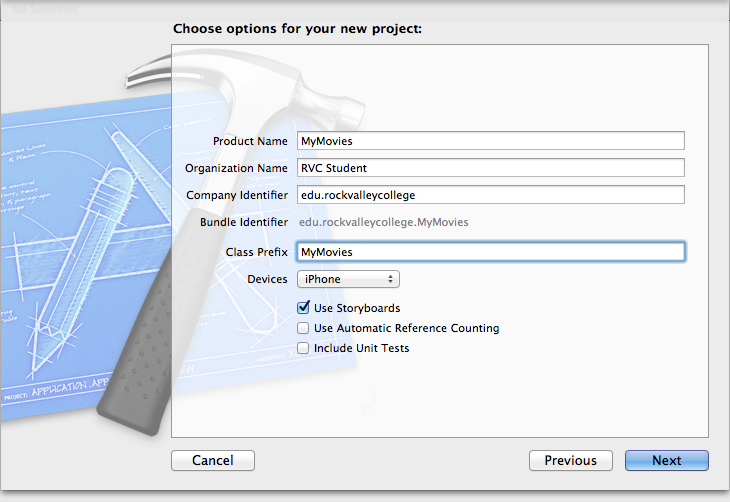
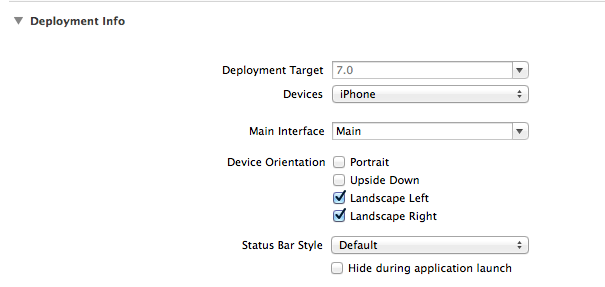
My Movie App – In Class Exercise

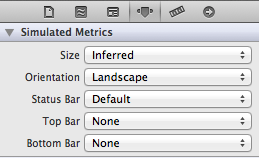
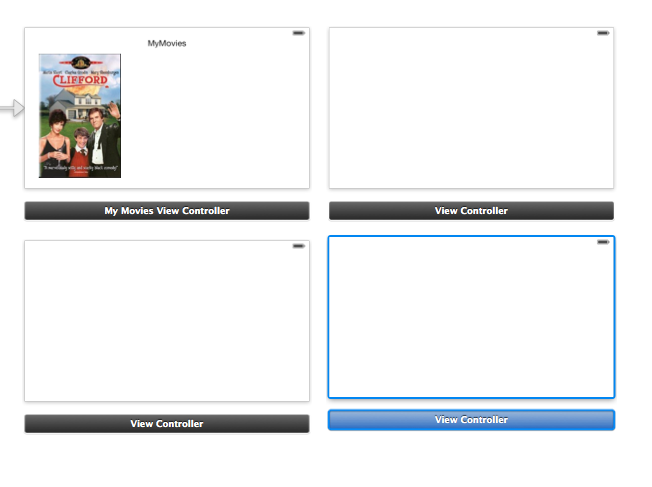
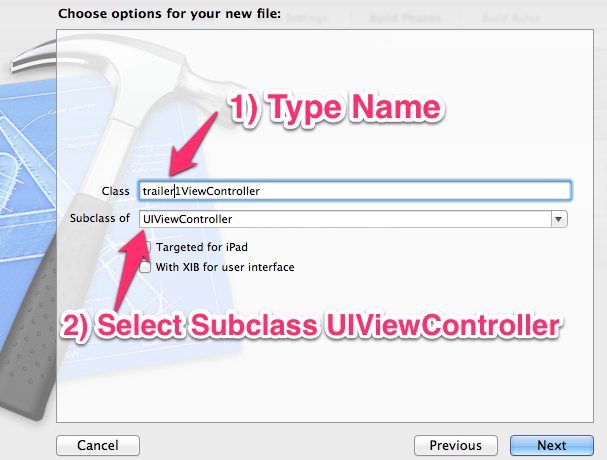
This app shows how to load local movies into your app. You will find 3 of your favorite movie trailers, download the movie and embed in app to play. The app will have a main title screen with options to launch the movie. The app will only run in landscape mode.

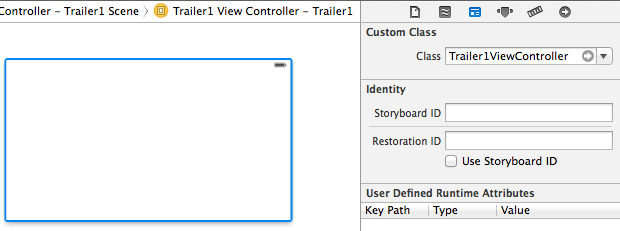
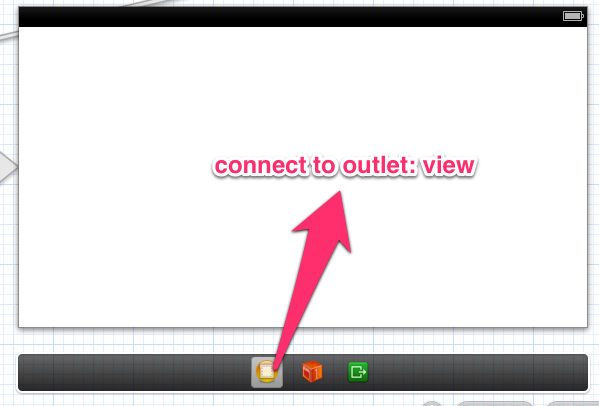
1. Start new storyboard iPhone single view app named “MyMovies”

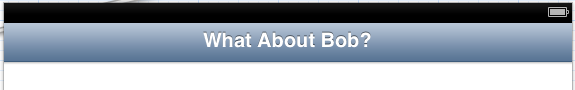


1. Change supported interface to Landscape

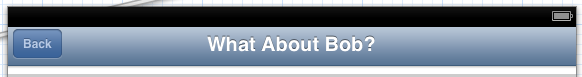


1. Change View Orientation to Landscape. Add label with “**My Movie**” title along top of main view
2. Download 3 movies from youtube.com and save as .mp4 files (Use FireFox addon to download youtube videos)  
   <https://addons.mozilla.org/en-us/firefox/addon/easy-youtube-video-downl-10137/>
   1. Make sure it play using Quicktime
3. Add movie into supporting files (rename to **trailer1.mp4,trailer2.mp4,trailer3.mp4)**
4. Create Group: **Images** and add images from movie (use http://imdb.com) into supporting files. You might have to [shift]+[command]+4 to capture image. (rename to **trailer1, trailer2,trailer3)**
5. Add 3 buttons to storyboad and add **trailer1, trailer2, trailer3** image to “image” inspector property. Resize to fit on left side of view.
6. Run App [Command] + R
7. Add 3 new View Controller (to load movie trailer)
   1. Change View Controller Attributes to below
   2. 
   3. 
   4. Change **Title** property to Title1, Title2, Title3
8. Create a new group named **Trailers** in project navigator
9. Add 3 Classes
   1. Right Click on Trailers > New File > Objective C Class and
      1. Name trailer1ViewController, trailer2ViewController, trailer3ViewController
      2. SubClass: UiViewController
      3. 
      4. Next > Create
   2. There should be an. h and .m create

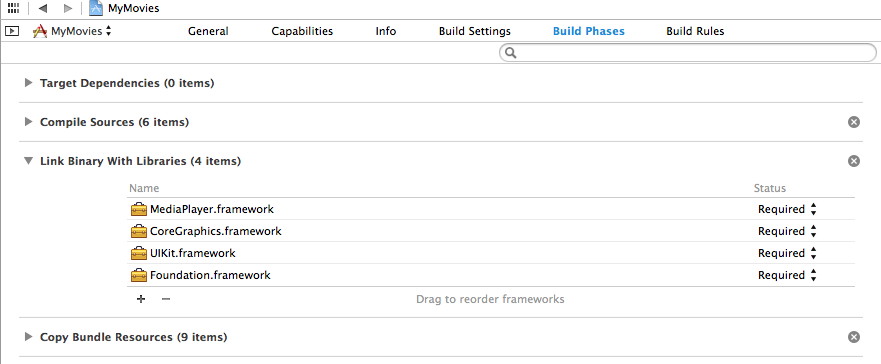
1. Bind new View Controllers to new classes
   1. Storyboard > Select Each ViewController with mouse > Click the **Show Identity Inspector** > Select Class
   2. 
2. Connect Viewcontrollers via Segue
   1. [CNTRL] drag buttons from main view to correct view (ie; Trailer1 Button > Trailer1 View)
   2. Select Segue action **modal**
3. Connect viewcontrol to view on each new viewcontrol
   1. 
4. Add **Navigation Bar** to top of new ViewController and change title to title of movie (place below battery icon)



1. Add **Bar Button** item to Navigation Bar and change text to **Back.** [Cntrl] + drag to Main home view control and select **modal** action

****

1. Add to link library MediaPlayer.Framework



1. Code for .h file (Yellow highlighted text needs added)

#import <UIKit/UIKit.h>

#import <MediaPlayer/MediaPlayer.h>

@interface Trailer1ViewController : UIViewController

{

MPMoviePlayerController \*player;

}

@end

1. Add code to .m file (Yellow highlighted text needs added)

- (void)viewDidLoad

{

//Change to match you video filename in supporting files

[super viewDidLoad];

// Do any additional setup after loading the view.

if([self checkinternet] == NO)

{

// Not connected to the internet

UIAlertView \*message = [[UIAlertView alloc] initWithTitle:@"Internet Connection Required"

message:@"Close app and return when internet connection available."

delegate:nil

cancelButtonTitle:@"OK"

otherButtonTitles:nil];

[message show];

}

else

{

//Change to match you video filename in supporting files

NSString \*url = [[NSBundle mainBundle]

pathForResource:@"trailer1"

ofType:@"mp4"];

player = [[MPMoviePlayerController alloc]

initWithContentURL:[NSURL fileURLWithPath:url]];

[[NSNotificationCenter defaultCenter]

addObserver:self

selector:@selector(movieFinishedCallback:)

name:MPMoviePlayerPlaybackDidFinishNotification

object:player];

//—set the size of the movie view and then add it to the View window—

//get screensize

CGSize size = [self getScreenSize];

player.view.frame = CGRectMake(0, 50, size.width,size.height);

[self.view addSubview:player.view];

//—play movie—

[player play];

}

}

//copy paste right below viewdidload

- (void) movieFinishedCallback:(NSNotification\*) aNotification {

MPMoviePlayerController \*moviePlayer = [aNotification object];

[[NSNotificationCenter defaultCenter]

removeObserver:self

name:MPMoviePlayerPlaybackDidFinishNotification

object:moviePlayer];

[moviePlayer.view removeFromSuperview];

}

-(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation

{

// Return YES for supported orientations

return (interfaceOrientation != UIInterfaceOrientationPortraitUpsideDown);

}

- (void)viewDidUnload

{

[super viewDidUnload];

}

- (void)viewWillAppear:(BOOL)animated

{

[super viewWillAppear:animated];

}

- (void)viewDidAppear:(BOOL)animated

{

[super viewDidAppear:animated];

}

- (void)viewWillDisappear:(BOOL)animated

{

[super viewWillDisappear:animated];

}

- (void)viewDidDisappear:(BOOL)animated

{

//Stop Player when Back button caused view to disappear

[player stop];

[super viewDidDisappear:animated];

}

- (CGSize)getScreenSize

{

//Get Screen size

CGSize size;

if (UIInterfaceOrientationIsLandscape([UIApplication sharedApplication].statusBarOrientation) && [[UIScreen mainScreen] bounds].size.height > [[UIScreen mainScreen] bounds].size.width) {

// in Landscape mode, width always higher than height

size.width = [[UIScreen mainScreen] bounds].size.height;

size.height = [[UIScreen mainScreen] bounds].size.width;

} else if (UIInterfaceOrientationIsPortrait([UIApplication sharedApplication].statusBarOrientation) && [[UIScreen mainScreen] bounds].size.height < [[UIScreen mainScreen] bounds].size.width) {

// in Portrait mode, height always higher than width

size.width = [[UIScreen mainScreen] bounds].size.height;

size.height = [[UIScreen mainScreen] bounds].size.width;

} else {

// otherwise it is normal

size.height = [[UIScreen mainScreen] bounds].size.height;

size.width = [[UIScreen mainScreen] bounds].size.width;

}

return size;

}

- (BOOL) checkinternet

{

//check internet connection

NSURL \*scriptUrl = [NSURL URLWithString:@"http://www.google.com/m"];

NSData \*data = [NSData dataWithContentsOfURL:scriptUrl];

if (data)

{

NSLog(@"Device is connected to the internet");

return YES;

}

else

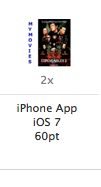
{

NSLog(@"Device is not connected to the internet");

return NO;

}

}

1. Create App icon (120x120) PNG file. Remember <http://pixlr.com>
   1. Click Arrow to right of AppiCon
   2. Drag and Drop 120x120 png onto **iPhone App iOS 7**
   3. 
2. Test, Done