ANDROID

CH - 5

Introduction To iPhone

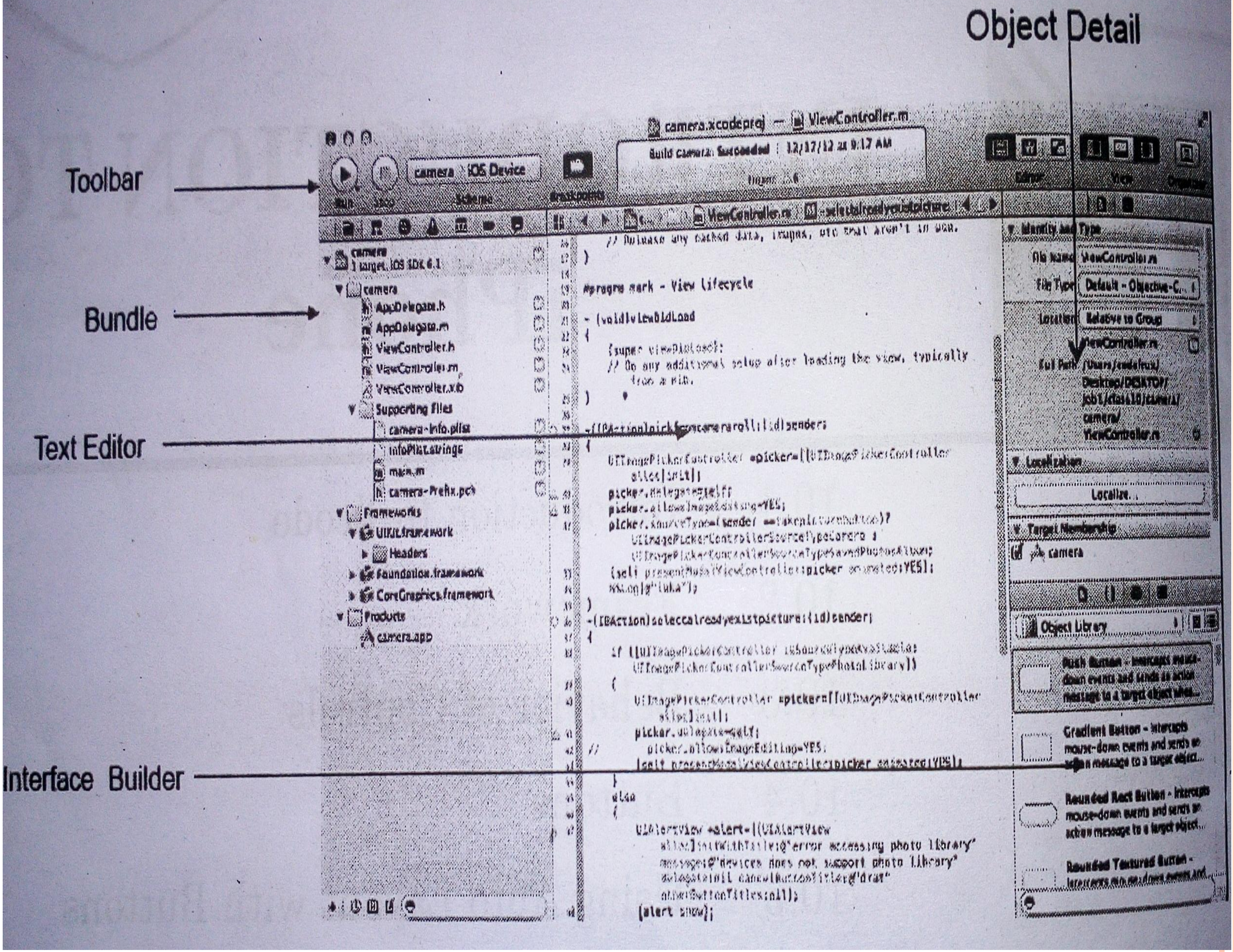
Prepared By :

Ms. Kakadiya Jainam A.

10.1 INTRODUCTION TO XCODE

- Xcode is an Integrated Development Environment(IDE) containing a suite of software development tools developed by Apple for developing soft for OS X and iOS first released in 2003.

- **1) Let’s check how it’s editor looks :**
  - The Xcode interface integrates code editing, user interface design, asset management, testing, and debugging within a single workspace window.
  - For example, select a file in one area, and an appropriate editor opens in another area.
  - You can customize your environment by opening multiple windows and multiple tabs per window.



Object Detail

Toolbar

Bundle

Text Editor

Interface Builder

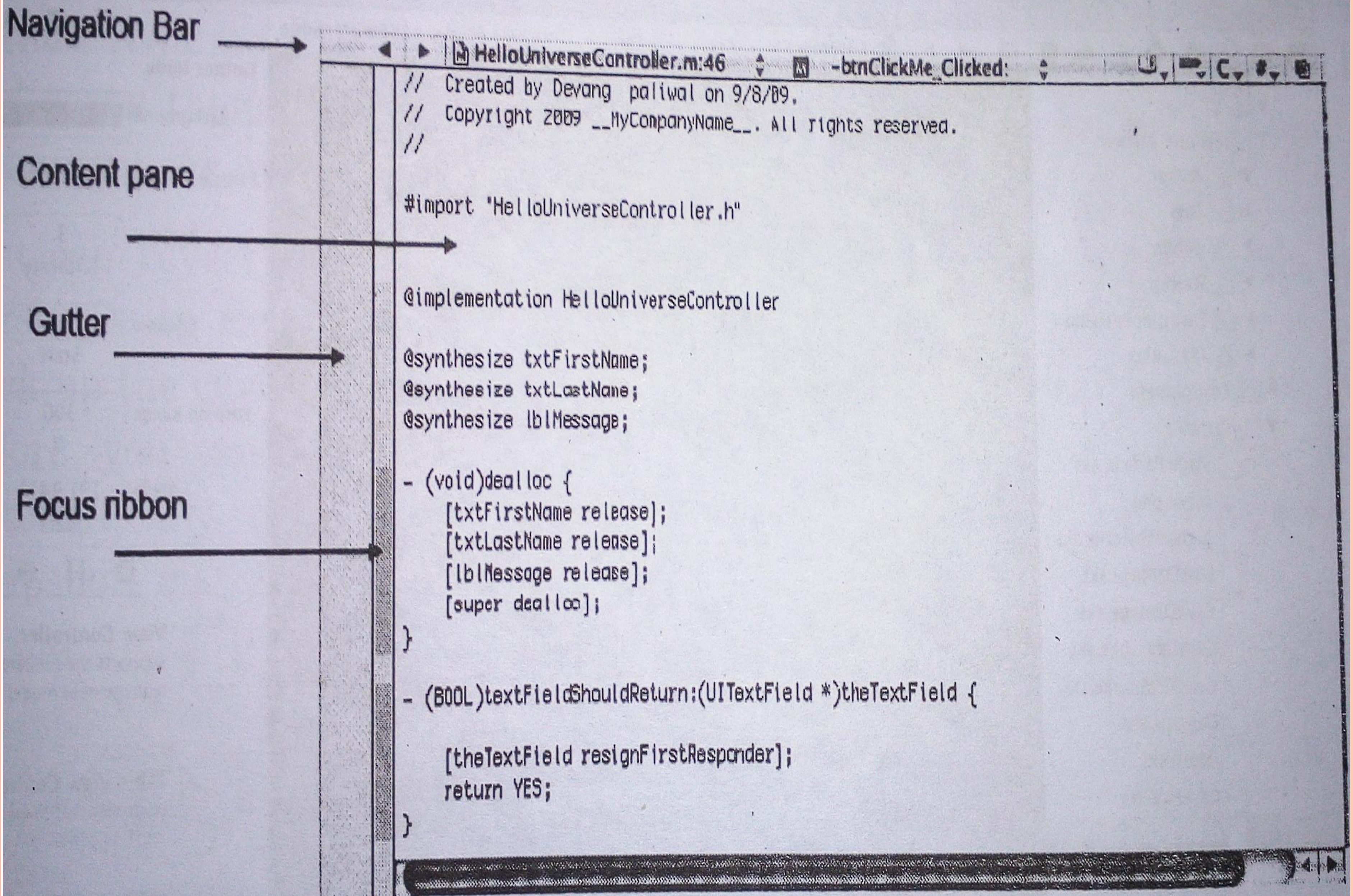
camera.xcodeproj — ViewController.m
Build camera: Succeeded | 12/17/12 at 9:17 AM
Run
Stop
camera iOS Device
Scheme
Breakpoints
Editor
View
Organizer
ViewController.m
-selectalreadyexistpicture
camera
1 target, iOS SDK 6.1
camera
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
camera-Info.plist
InfoPlist.strings
main.m
camera-Prefix.pch
Frameworks
UIKit.framework
Headers
Foundation.framework
CoreGraphics.framework
Products
camera.app
Identity and Type
File Name ViewController.m
File Type Default - Objective-C...
Location Relative to Group
ViewController.m
Full Path /Users/.../Desktop/DESKTOP/job1/class10/camera/camera/ViewController.m
Localization
Localize...
Target Membership
camera
Object Library
Push Button - Intercepts mouse-down events and sends an action message to a target object when...
Gradient Button - Intercepts mouse-down events and sends an action message to a target object...
Rounded Rect Button - Intercepts mouse-down events and sends an action message to a target object...
Rounded Textured Button - Intercepts mouse-down events and...

```
    // Release any cached data, images, etc that aren't in use.
}

#pragma mark - View lifecycle

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically
        from a nib.
}

-(IBAction)pickfromcameraroll:(id)sender;
{
    UIImagePickerController *picker=[[UIImagePickerController
        alloc]init]];
    picker.delegate=self;
    picker.allowsImageEditing=YES;
    picker.sourceType=(sender ==takepicturebutton)?
        UIImagePickerControllerSourceTypeCamera :
        UIImagePickerControllerSourceTypeSavedPhotosAlbum;
    [self presentModalViewController:picker animated:YES];
    NSLog(@"luka");
}
-(IBAction)seleccalreadyexistpicture:(id)sender;
{
    if ([UIImagePickerController isSourceTypeAvailable:
        UIImagePickerControllerSourceTypePhotoLibrary])
    {
        UIImagePickerController *picker=[[UIImagePickerController
            alloc]init];
        picker.delegate=self;
//        picker.allowsImageEditing=YES;
        [self presentModalViewController:picker animated:YES];
    }
    else
    {
        UIAlertView *alert=[[UIAlertView
            alloc]initWithTitle:@"error accessing photo library"
            message:@"devices does not support photo library"
            delegate:nil cancelButtonTitle:@"drat"
            otherButtonTitles:nil];
        [alert show];
```

- **2) Assisted Source Code Editing**

  - Xcode checks your source code as you it, and when Xcode notice a mistake, the source code editor highlights the error. The source code editor then offers to fix it.

- Text Editor



Navigation Bar

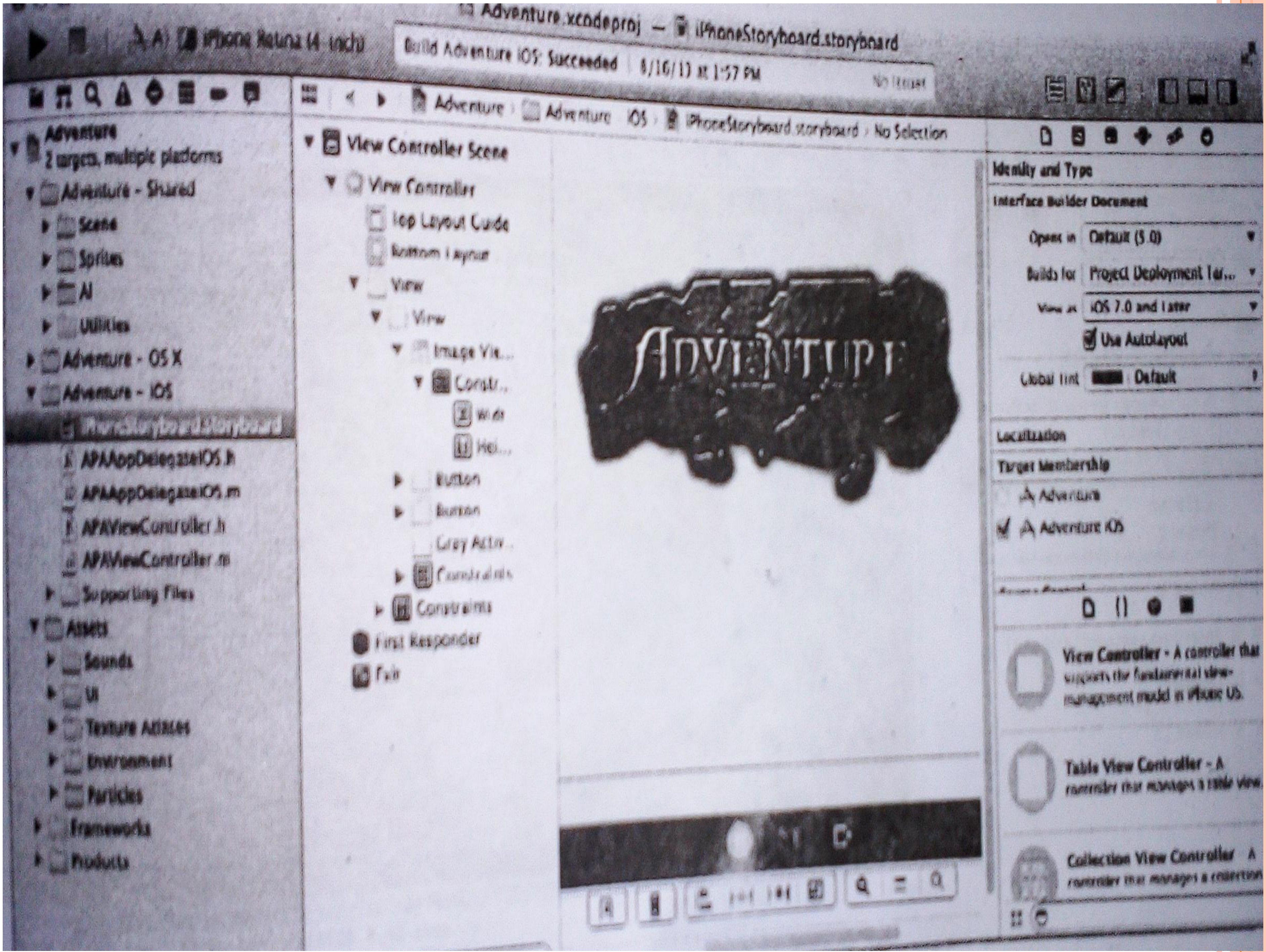
Content pane

Gutter

Focus ribbon

```
//  Created by Devang paliwal on 9/8/09.
//  Copyright 2009 __MyCompanyName__. All rights reserved.
//

#import "HelloUniverseController.h"


@implementation HelloUniverseController

@synthesize txtFirstName;
@synthesize txtLastName;
@synthesize lblMessage;

- (void)dealloc {
    [txtFirstName release];
    [txtLastName release];
    [lblMessage release];
    [super dealloc];
}

- (BOOL)textFieldShouldReturn:(UITextField *)theTextField {

    [theTextField resignFirstResponder];
    return YES;
}
```

- **3) Graphical UI Design :**
  - Interface Builder is a visual design editor that’s integrated into Xcode. Use Interface Builder to create the user interface of your aapps by assembling windows, views, controls, menus, and other elements from a library of configurable objects.



Build Adventure iOS: Succeeded | 8/16/13 at 1:57 PM

No Issues

Opens in: Default (5.0)

Builds for: Project Deployment Tar...

View as: iOS 7.0 and Later

☑ Use Autolayout

Global Tint: Default

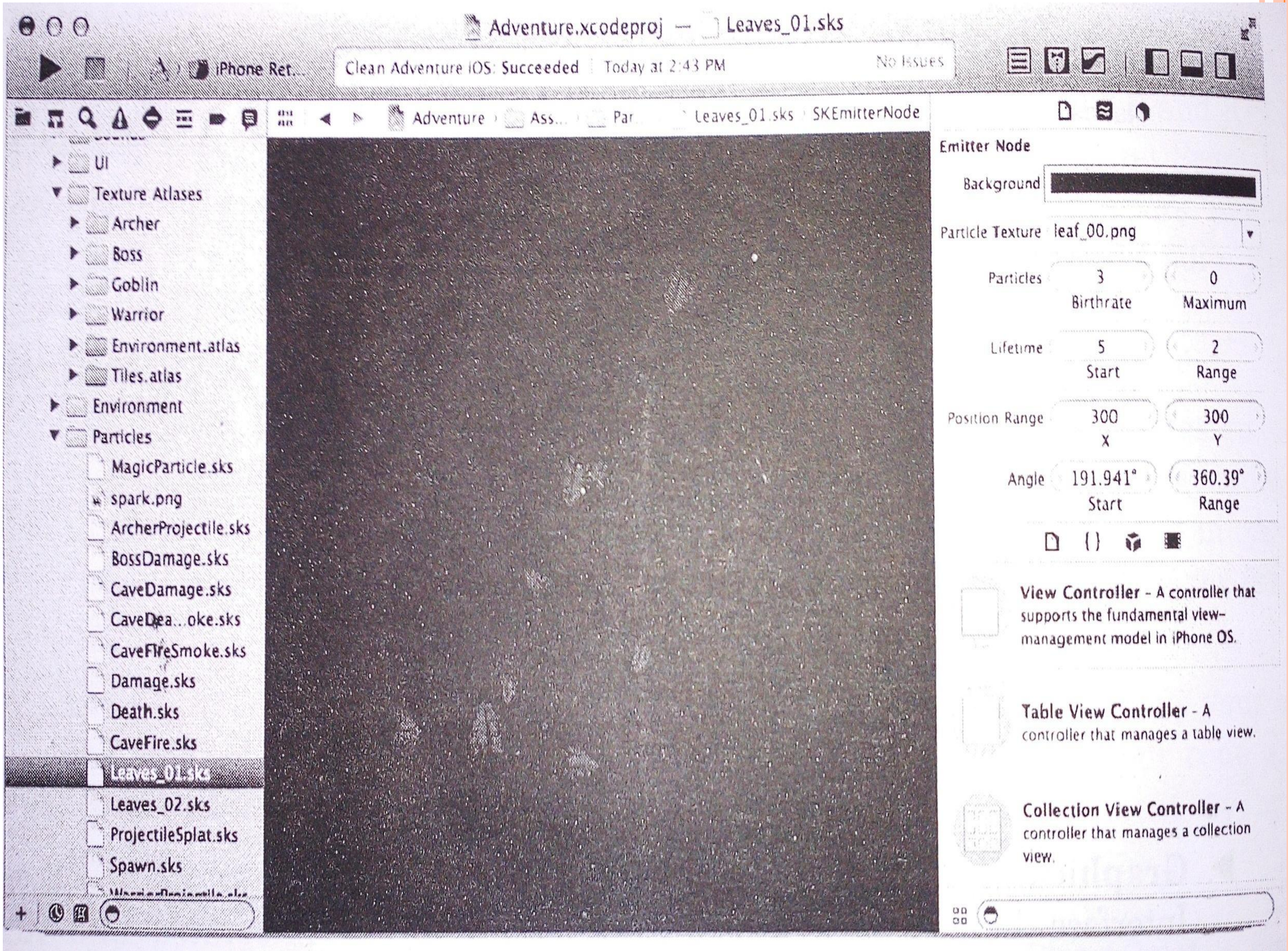
□ Adventure

☑ Adventure iOS

View Controller - A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller - A controller that manages a table view.

Collection View Controller - A controller that manages a collection



Emitter Node

Background

Particle Texture: leaf_00.png

Particles: 3 Birthrate, 0 Maximum

Lifetime: 5 Start, 2 Range

Position Range: 300 X, 300 Y

Angle: 191.941° Start, 360.39° Range

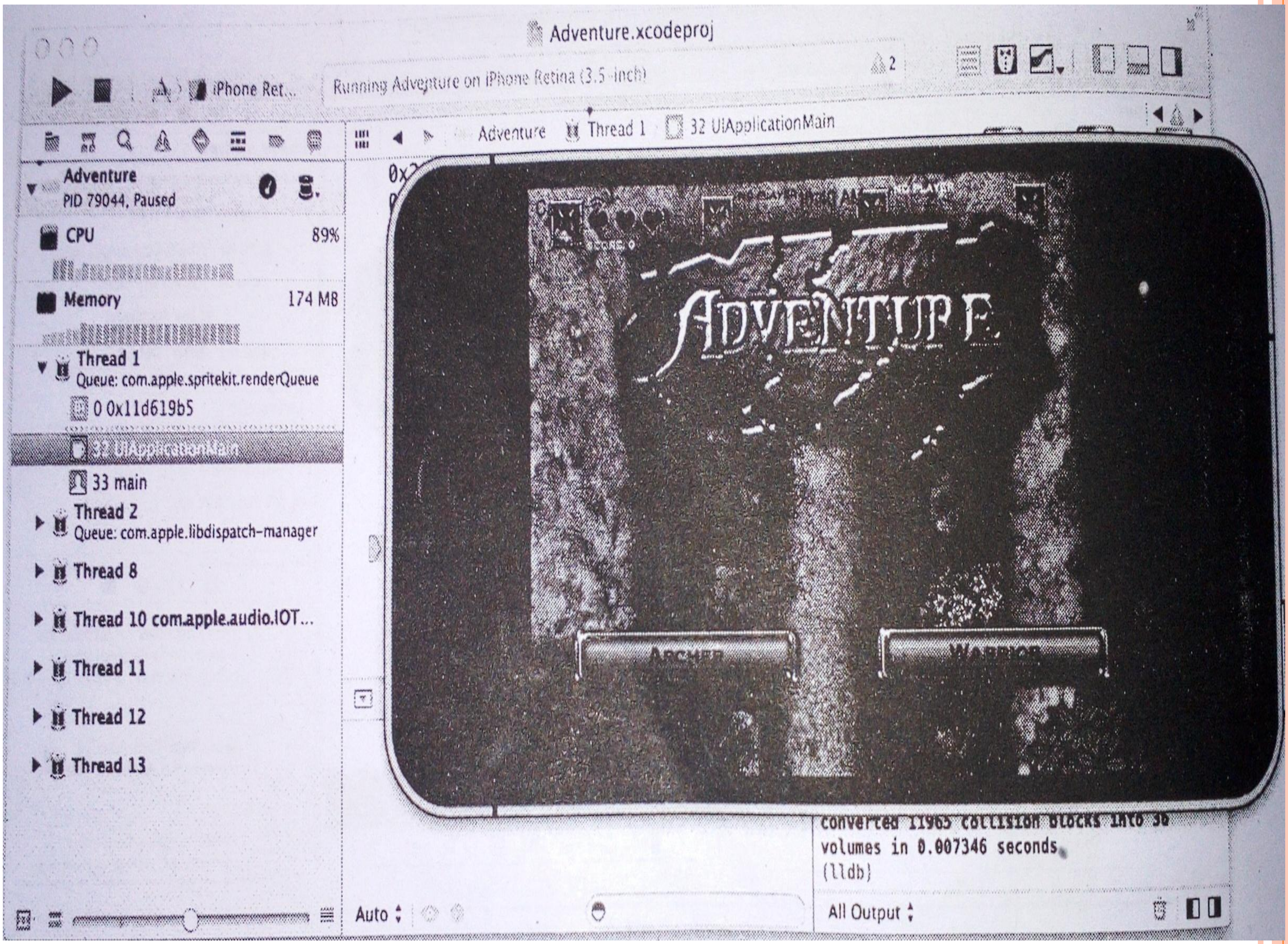
View Controller - A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller - A controller that manages a table view.

Collection View Controller - A controller that manages a collection view.

4) Integrated Debugging :

- Builder your app, and Xcode launches it and immediately starts a debugging session. If you are running an iOS app, Xcode launches it either in iOS Simulator or on an iOS device connected to your Mac.



Running Adventure on iPhone Retina (3.5-inch)

Adventure
PID 79044, Paused

CPU 89%

Memory 174 MB

Thread 1
Queue: com.apple.spritekit.renderQueue
0 0x11d619b5
32 UIApplicationMain
33 main

Thread 2
Queue: com.apple.libdispatch-manager

Thread 8

Thread 10 com.apple.audio.IOT...

Thread 11

Thread 12

Thread 13

```
converted 11965 collision blocks into 36
volumes in 0.007346 seconds
(lldb)
```

- **5) Automatic Saves, Projects Snapshots, and Source Control Management :**

  - While you work, Xcode automatically saves changes to source and project files. This feature required so configure.

10.2 FRAMEWORK

- A framework is a hierarchical directory that encapsulates shared resources, such as a dynamic shared library, images files, header files etc.

- A framework is also a bundle and its contents can be accessed using Core Foundation Bundle Service or the Cocoa NSBundle class.

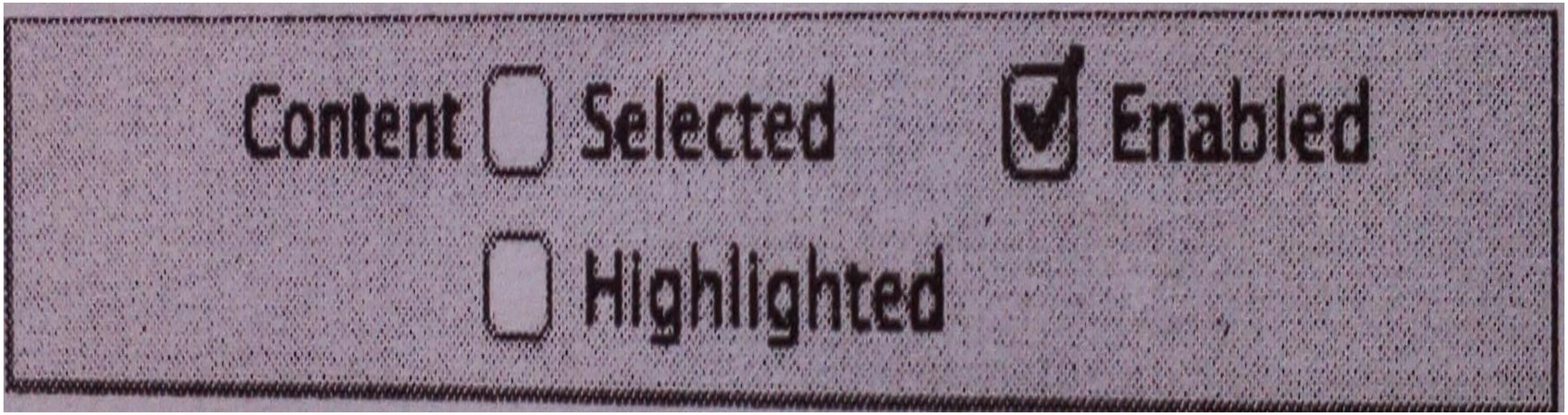
- Framework serve the same purpose as static and dynamic shared libraries.

- Frameworks offer the following advantages over static-linked libraries and other types of dynamic shared libraries :
  - Frameworks group related resources together.
  - Framework can include a wider variety of resource types than libraries. E.g framework can include any header file, documents etc.
  - Multiple variety of a framework can be include in the same bundle.

10.3 Behavior Of Controls

- **Control States :** A Control State describes the current interactive state of a control : normal, selected, enabled, or highlighted.

- The fast way to configure the initial state of s controls is by using the Attributes Inspector :



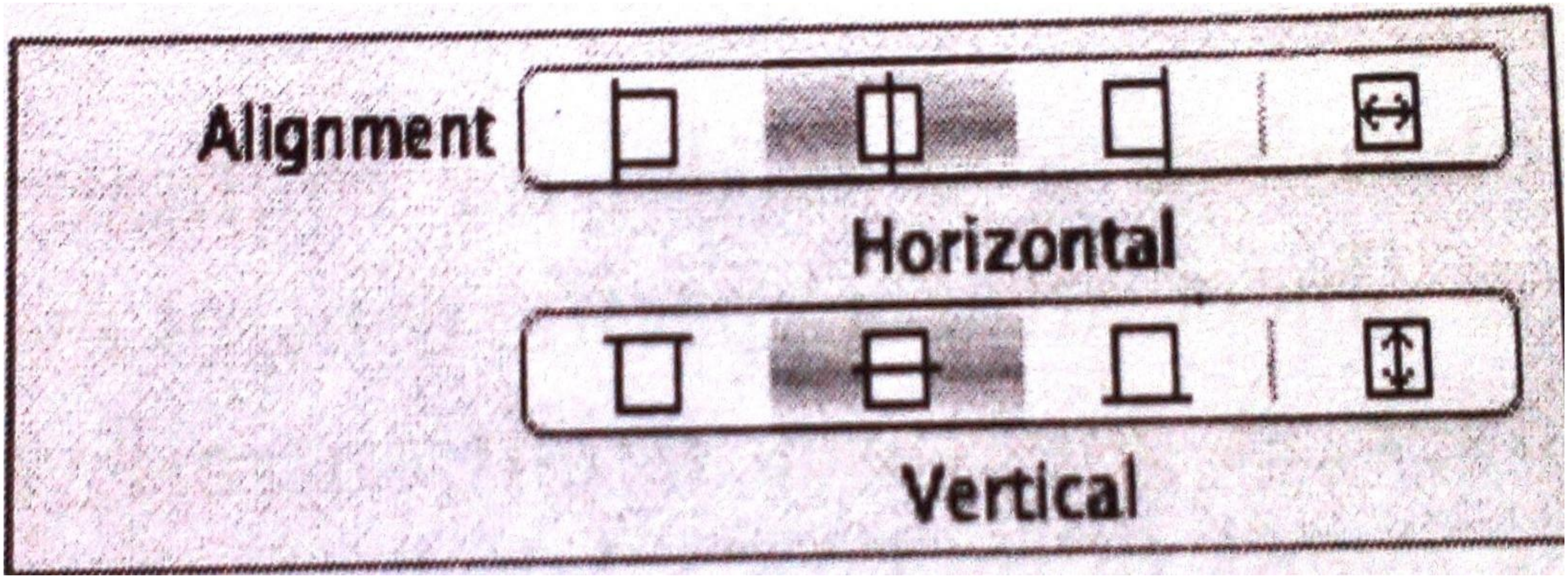
Content □ Selected ☑ Enabled

□ Highlighted

- **Control Events :** A Control Event represents various physical gestures that uses can make on controls, such as lifting a finger from a control, dragging a finger into a control, and touching down within a text field.

- **Target-Action Mechanism :** The target-action mechanism is a model for configuring a control to send an action message to s target object after a specific control event. For example, when a user interacts with the slider, it generates a UIControlEventValueChanged control event.

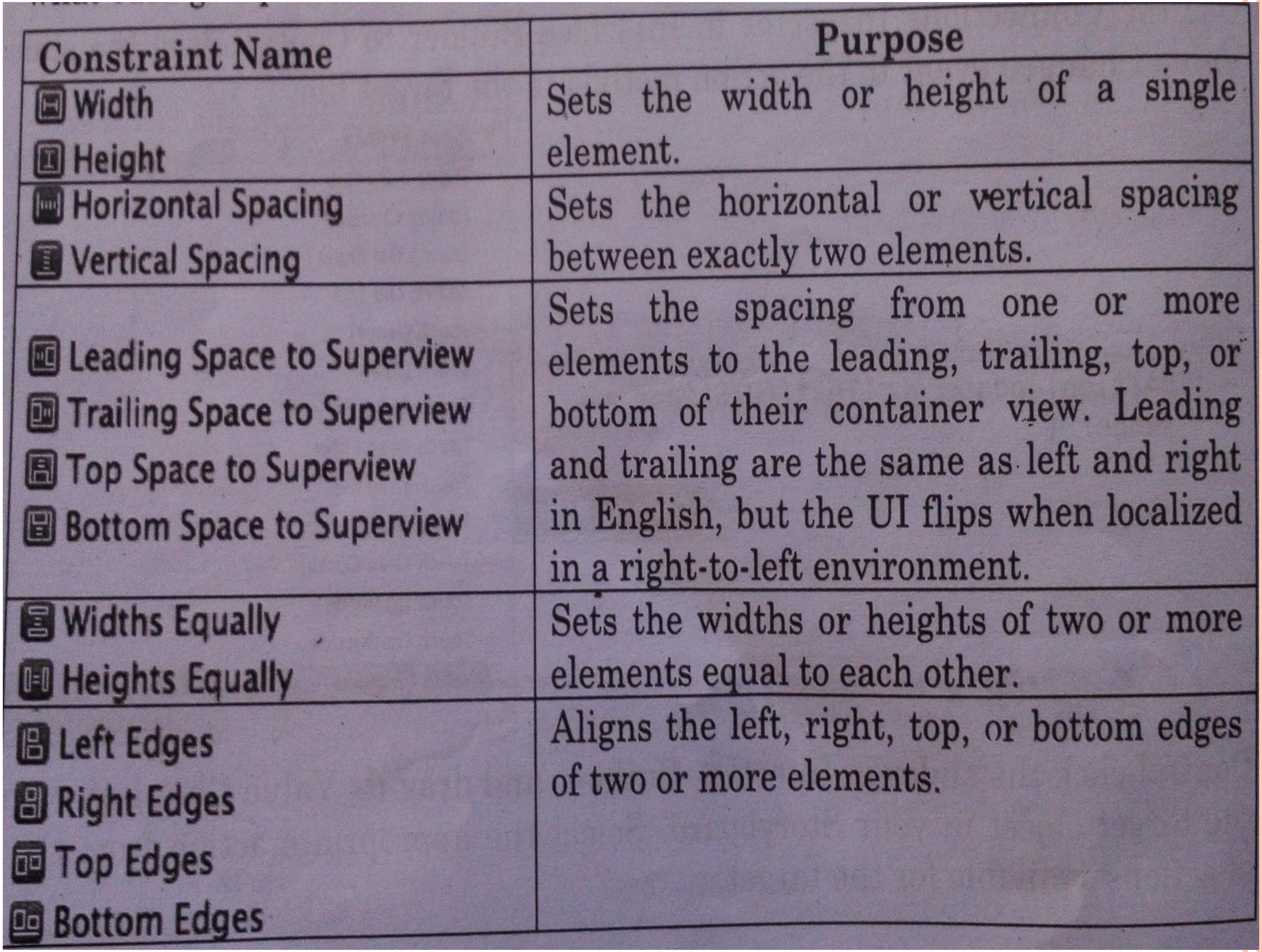
- **Content Alignment :** Certain controls- such as buttons and text fields can contain custom images or text. For this controls, you can specify the alignment of that content by using the “Horizontal Alignment” and “Vertical Alignment” option in Attributes Inspectors.



- **Using Auto Layout with Controls :** The auto layout system allows you to define layout constraints for user interface elements, such as views and controls.

- constraints represent relationship between user interface elements.

- The following tables describe what each group of constraints in the auto layout menu accomplishes :



Constraint Name	Purpose
Width Height	Sets the width or height of a single element.
Horizontal Spacing Vertical Spacing	Sets the horizontal or vertical spacing between exactly two elements.
Leading Space to Superview Trailing Space to Superview Top Space to Superview Bottom Space to Superview	Sets the spacing from one or more elements to the leading, trailing, top, or bottom of their container view. Leading and trailing are the same as left and right in English, but the UI flips when localized in a right-to-left environment.
Widths Equally Heights Equally	Sets the widths or heights of two or more elements equal to each other.
Left Edges Right Edges Top Edges Bottom Edges	Aligns the left, right, top, or bottom edges of two or more elements.

- **Making Controls Accessible :** Controls are accessible by default. To be useful, an accessible user interface element must provide accurate and helpful information about its screen position, name, behaviour, value, and type.

- The UI Accessibility programming interface defines the following attributes :

  - **Label :** A short, localized word or phase that briefly describes the control or view, but does not identify the element’s type. Example are “Add” or “Play”.

- **Hint :** A brief, localized pharse that describes the rules of an action on an element. Examples are “Adds a title” or “Opens the shopping list”.

- **Frame :** The frame of the element in screen coordinates, which is given by the structure that specifies an element’s screen location and size.

- **Value :** The current value of an element, when the value is not represented by the label. For example, the label for a slider might be “Speed” , but its current value might be “50%”

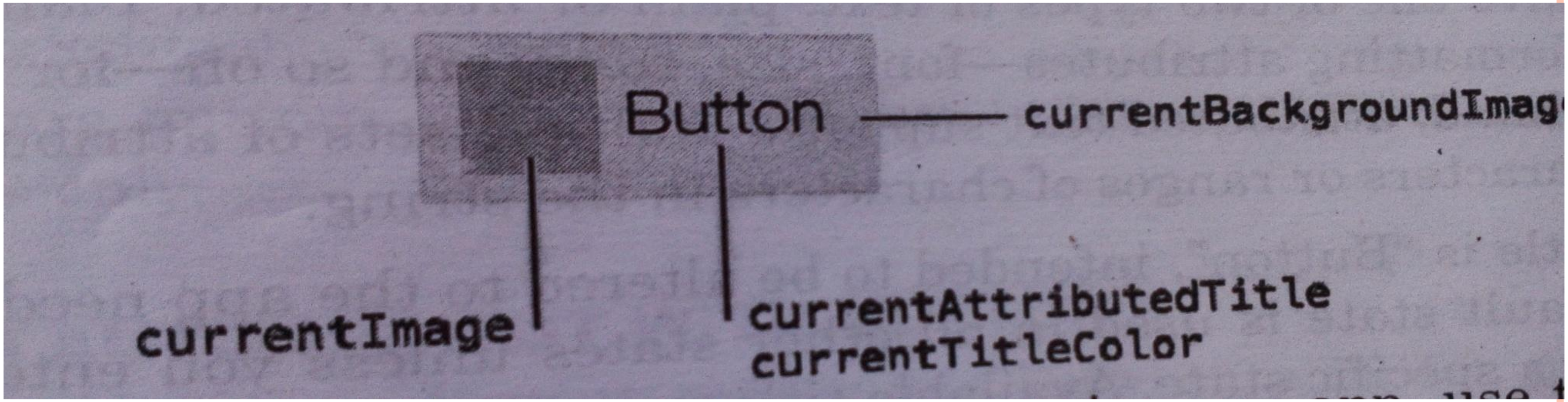
10.4 BUTTON

- You app changes button appearance based upon user touch interactions, using highlighting, changes in the label or image, color, and state to indicate the button action dynamically.

  - **Content of Buttons :** Set a button’s content using the Type field in the Attributes Inspector. Button objects can be specified as one of five standard types – system, detail disclosure, info light, info dark, and add contact.

- **Behavior of Buttons :** Buttons do not need a delegate to function property a view controller can define their behavior and functionality without implementing any delegate protocols.

- **Appearance of Buttons :** You can customize the appearance of a button by setting the properties show below.



10.5 Using Auto Layout With Button

- You can create Auto Layout constraints between a button and other user interface elements. You can create any type of constraint for a button.

  - **Elements Similar to a Button :** The following element provides similar functionality to a toolbar or for navigation in a navigation bar.

  - **1) Text Fields :** Text fields allow the user to inout a single line of text into an app.

- **2) Content of Text Fields:** Sets the content of the text field using the Text field.

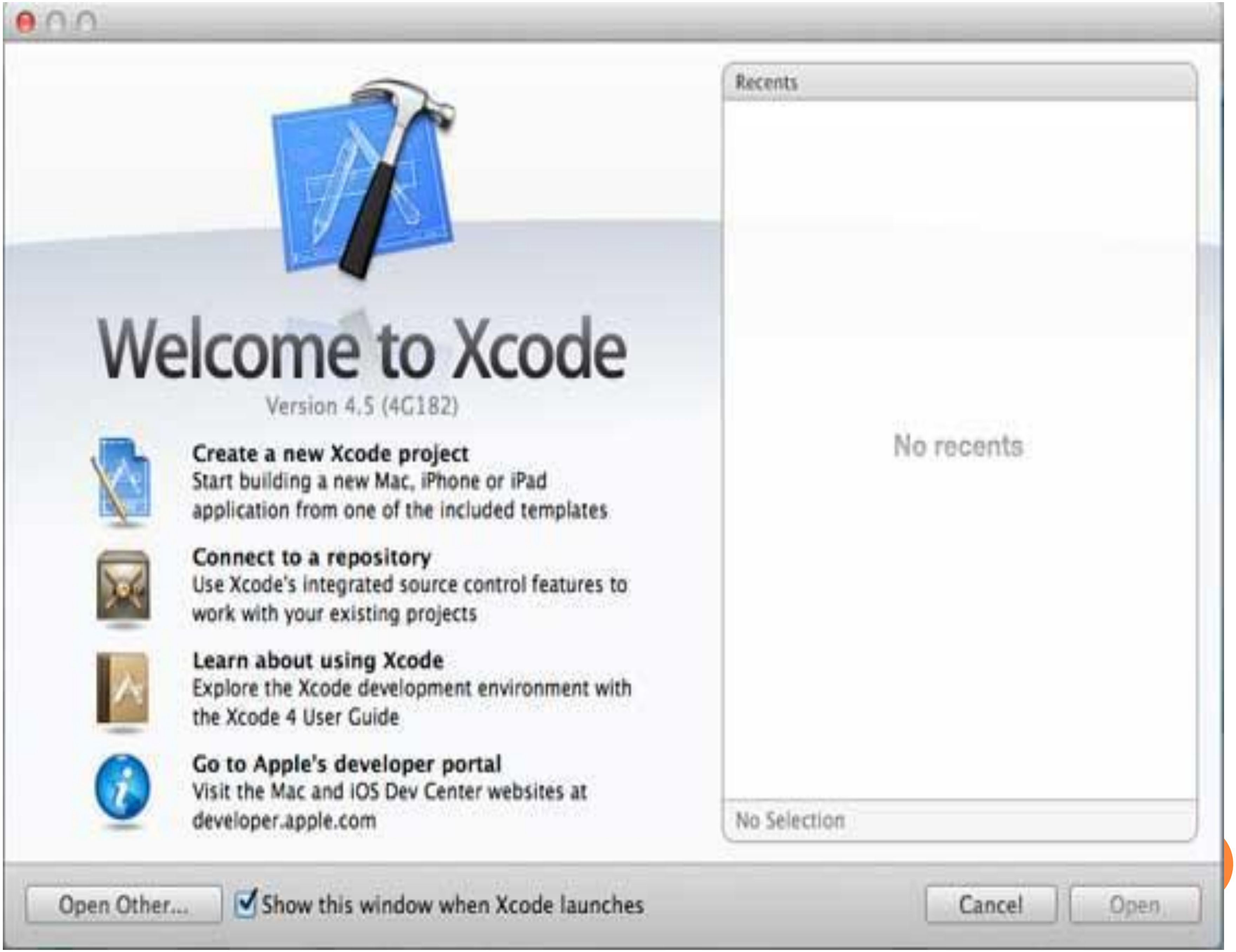
- **3) Behavior of Text Fields :** Text fields need a delegate to handle any custom behaviours, such as displaying additional overlay views when a user begins editing it.

Creating our First App

- Now we are just going to create a simple single view application (a blank app) that just runs on the iOS simulator.

- The steps are as follows :

  - 1. Open Xcode and select create a new Xcode project.



Welcome to Xcode

Version 4.5 (4G182)

Create a new Xcode project
Start building a new Mac, iPhone or iPad application from one of the included templates

Connect to a repository
Use Xcode's integrated source control features to work with your existing projects

Learn about using Xcode
Explore the Xcode development environment with the Xcode 4 User Guide

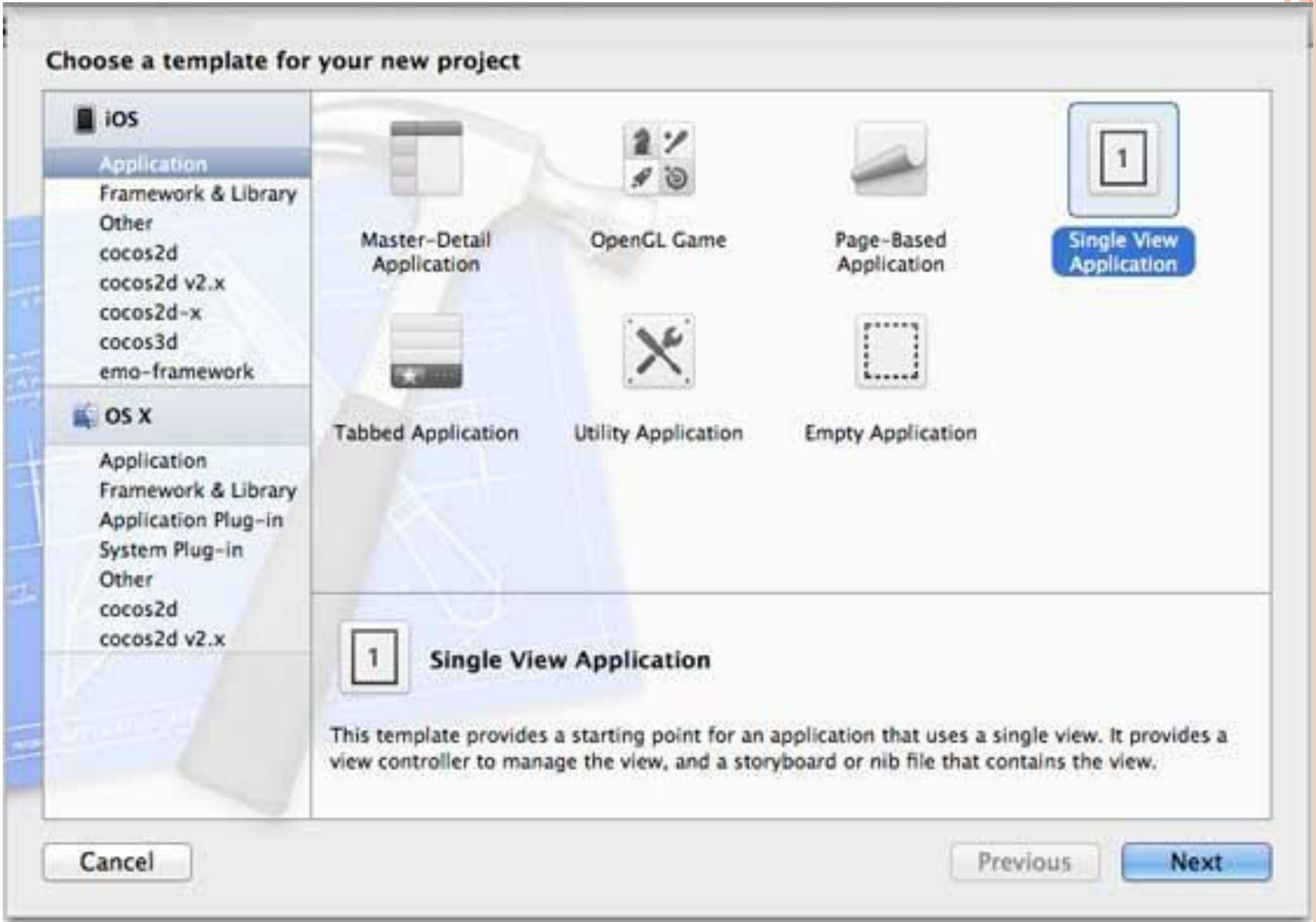
Go to Apple's developer portal
Visit the Mac and iOS Dev Center websites at developer.apple.com

Recents

No recents

No Selection

- **2. Then select single view application**

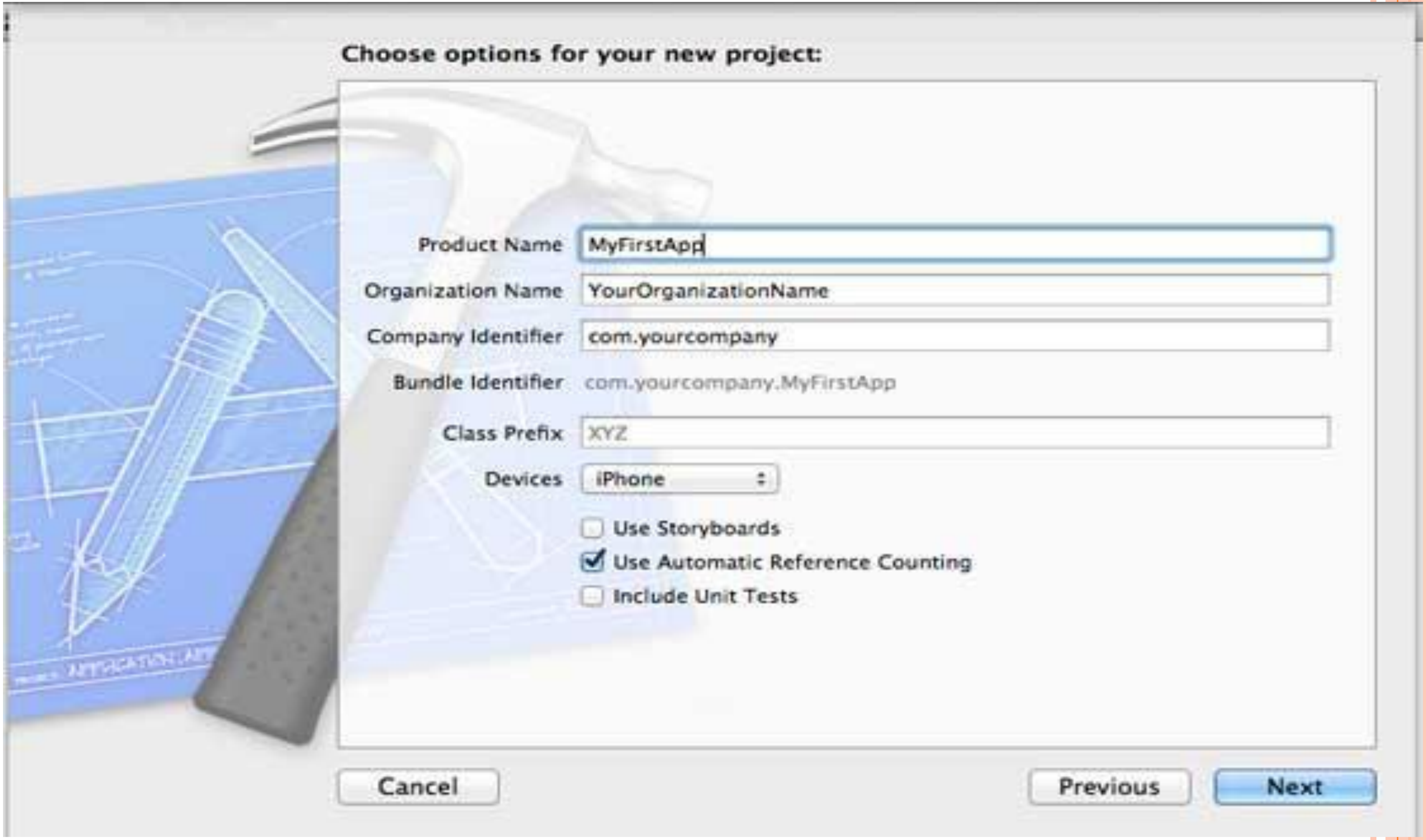


Choose a template for your new project

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.

- 3. Then enter product name i.e. the name of the application, organization name and then the company identifier.



Choose options for your new project:

Product Name: MyFirstApp

Organization Name: YourOrganizationName

Company Identifier: com.yourcompany

Bundle Identifier: com.yourcompany.MyFirstApp

Class Prefix: XYZ

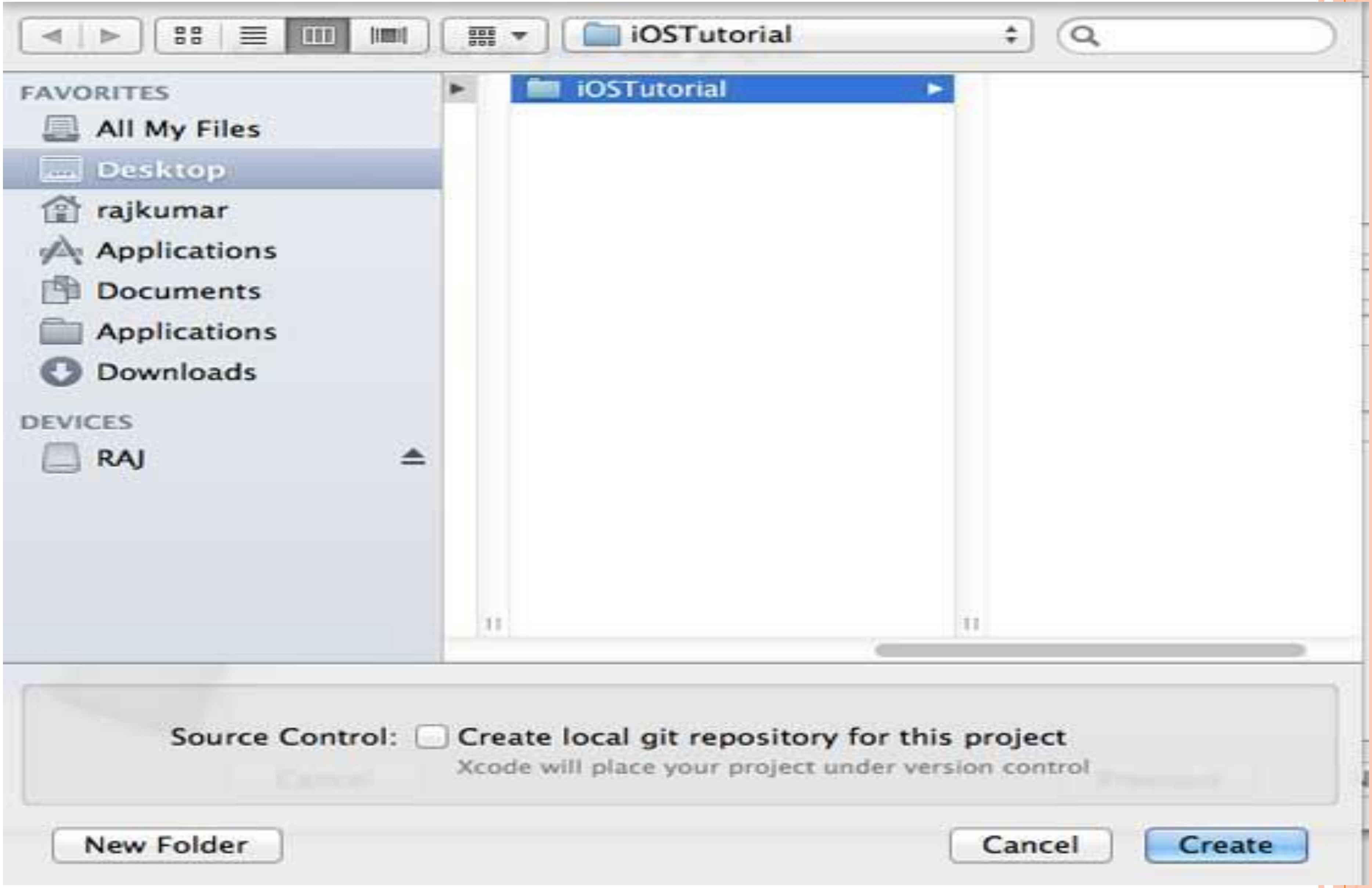
Devices: iPhone

□ Use Storyboards

☑ Use Automatic Reference Counting

□ Include Unit Tests

- 5. Select the directory for the project and select create.



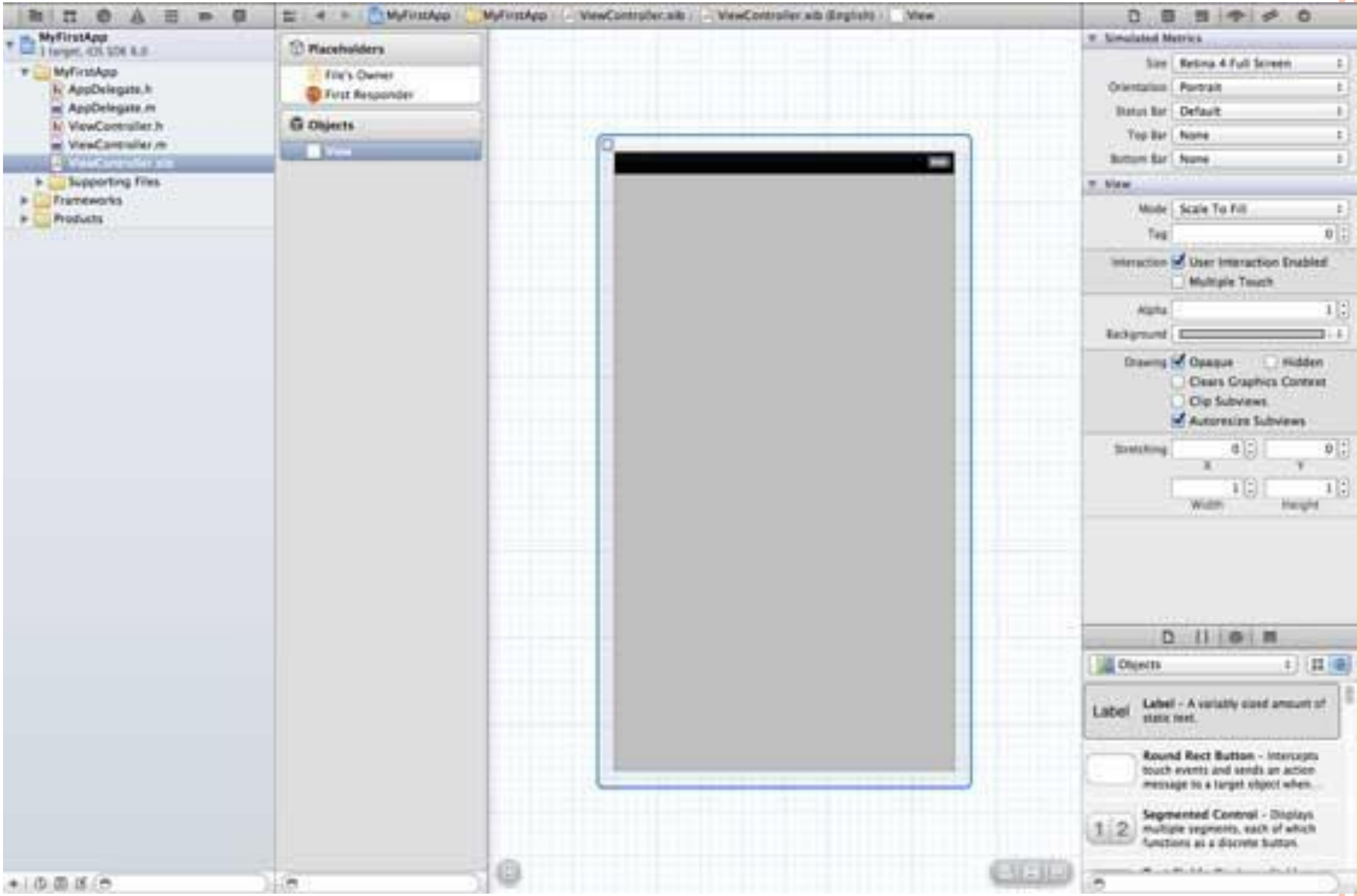
Source Control: □ Create local git repository for this project
Xcode will place your project under version control

Creating the iOS App User Interface

- Simply by the very nature of the environment in which they run, iPhone apps are typically visually oriented.

- As such, a key component of just about any app involves a user interface through which the user will interact with the application and, in turn , receive feedback.

Creating the iOS App User Interface



▼ Simulated Metrics

Size: Retina 4 Full Screen

Orientation: Portrait

Status Bar: Default

Top Bar: None

Bottom Bar: None

▼ View

Mode: Scale To Fill

Tag: 0

Interaction: ☑ User Interaction Enabled
□ Multiple Touch

Alpha: 1

Background:

Drawing: ☑ Opaque □ Hidden
□ Clears Graphics Context
□ Clip Subviews
☑ Autoresize Subviews

Stretching: X 0, Y 0, Width 1, Height 1

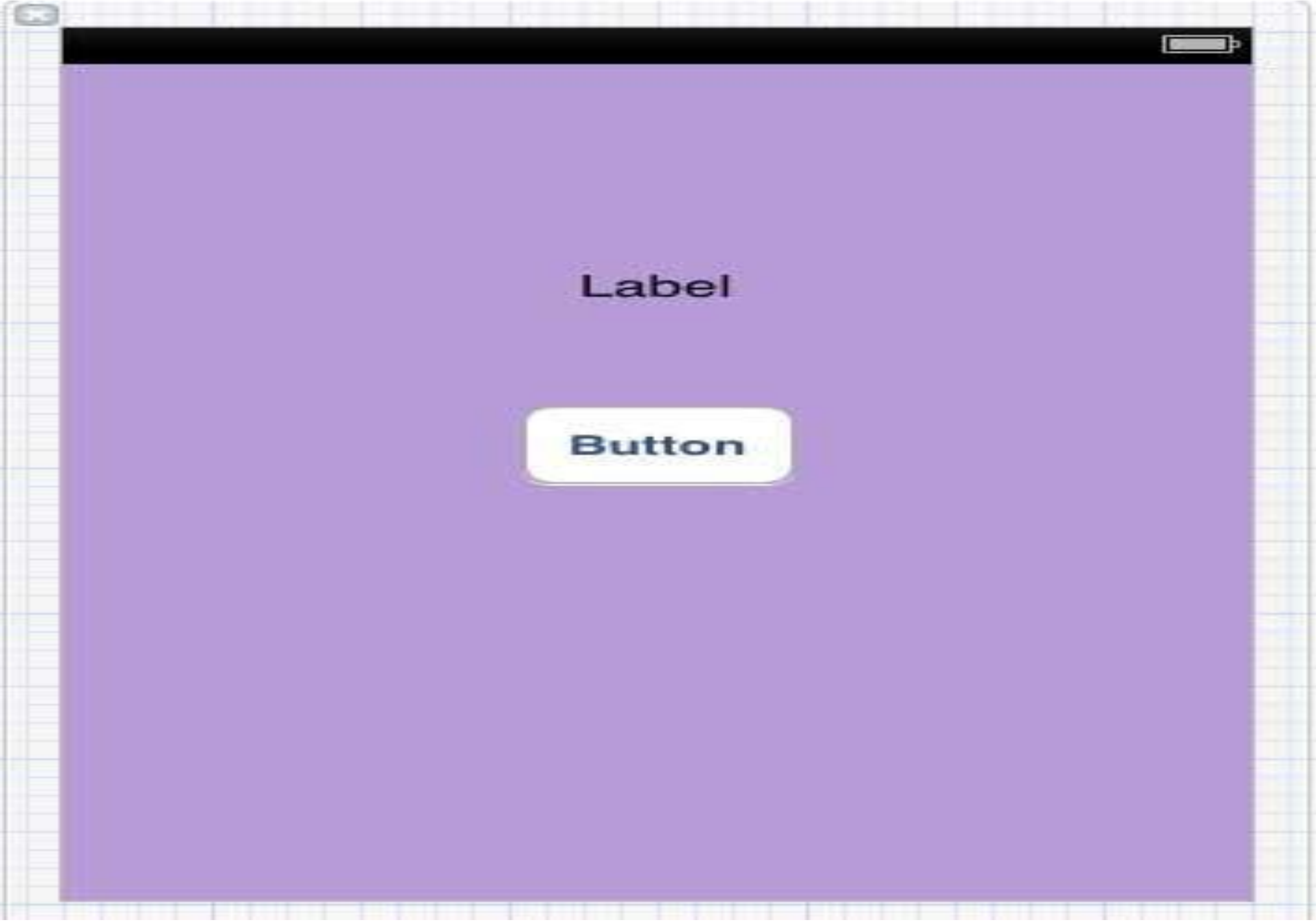
Label – A variably sized amount of static text.

Round Rect Button – Intercepts touch events and sends an action message to a target object when...

Segmented Control – Displays multiple segments, each of which functions as a discrete button.

Creating the iOS App User Interface

- 2. Adding Objects to the User Interface (Let add a Label and Button to our view) .



Cocoa touch

- Cocoa touch is a user interface framework provided by Apple for building software applications for products like iPhone, iPad, and iPod Touch.
- It is primarily written in Objective C language and is based on MAC OS x.
- Cocoa Touch was developed based on model view controller software architecture.

COCOA TOUCH

- The high-level application programming interface available in Cocoa Touch help to make animation, networking, and adding the appearance and behaviour of the native platform to the developed applications possible with less code development.

- **The main features of Cocoa Touch include :**

  - **1) Core Animation :** Helps to create rich user experiences by allowing for the smooth movement of visual elements.

Cocoa touch

- **2) Core Data :** Provides an object-oriented data management solution and aids in defining an application's data model in a logical and graphical way.
- **3) Code Audio**

- **Cocoa Touch is made up of several frameworks, but the key ones are :**

  - **Audio and Video :**
    - Core Audio
    - OpenAL

- Media Library

- **Data Management :**
  - Core Data
  - SQLite
- **Data Management :**
  - Core Animation
  - Quartz 2D
- **Networking and Internet :**
  - Socket
  - WebKit
- **User Applications:**
  - Address Book
  - Core Location

MVC INTRODUCTION

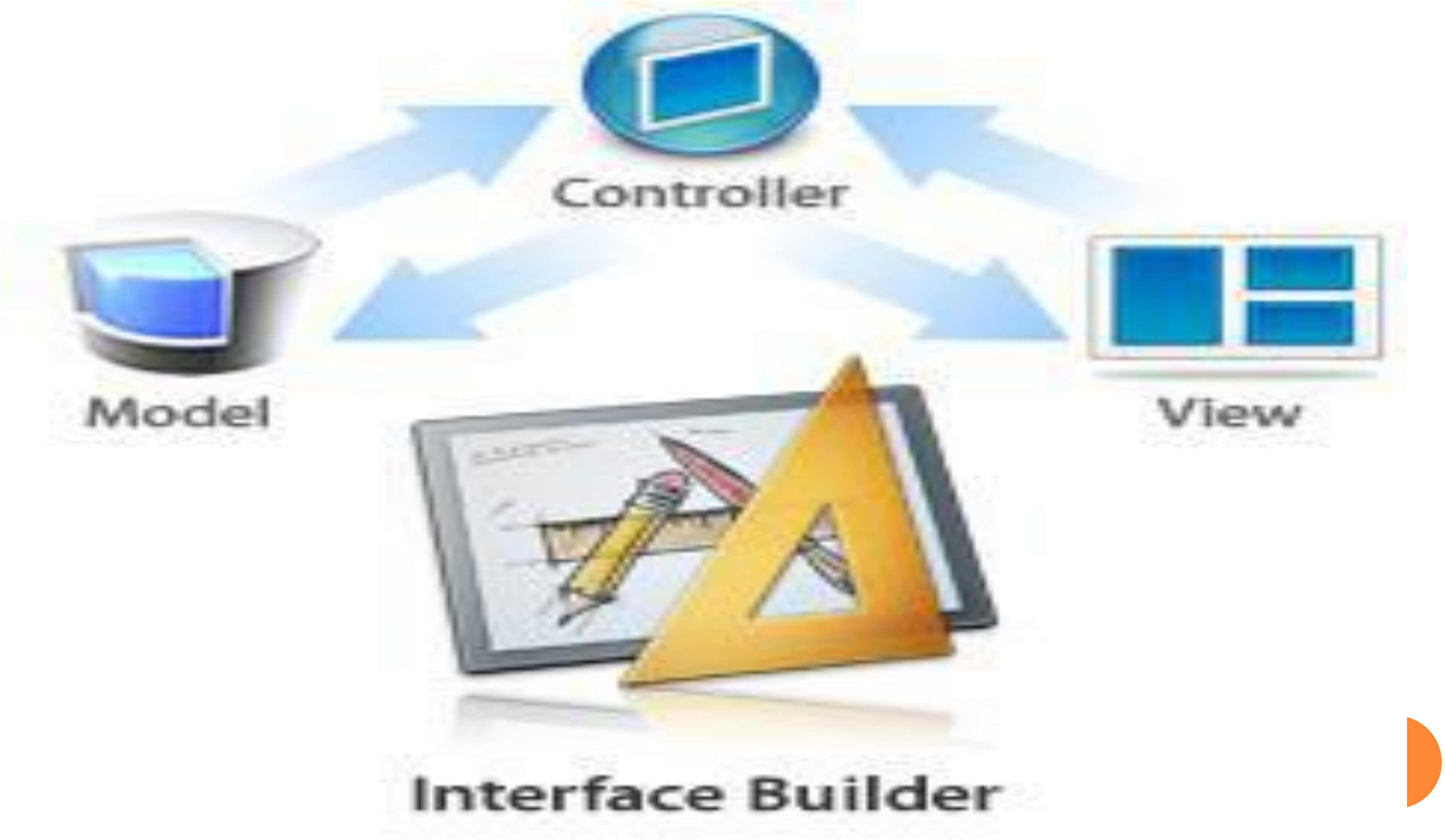
- The MVC software design pattern refers to three seprate role : **i)** Model , **ii)** View and, **iii)** Controller

- **i) The Model :** The Model represents the data in your application. It's responsible for sorting, validating and organizing your data.

- **ii) The View :** The View is the user interface. You can create views programmatically through code using Apple classes such as UIView or you can create a XIB file to represent view and visually layout your elements through Interface Builder.

- **iii) The Controller :** The Controller manages the communication between the view and the model. It takes the data from the model and communicates it to the view for display.
- The benefit of this modular architecture is that the separation of roles allows us to make modifications easily with less bugs.
- For example, if in the future, you need to make a change to the way the data is fetched or organized, all you need to do is switch out model. As long as you keep the interface of the model the same, then the views and controller will be none the wiser.

- Here’s a diagram illustrating what we’ve talked about


Controller
Model
View
Interface Builder