

# 实例 iPhone SDK 编程入门教程 - 第一天

## DAY ONE - MinutesToMidnight

大家一起来建立我们的第一个 iPhone app，给你的 iPhone 计算离午夜 12:00 点的剩余时间。

首先运行以安装好的 xCode

选择: File->New Project.

从 "New Project" 窗口

选择 : iPhone OS ->Applications-> View-Based Application

命名 : 我这里命名为 "MinutesToMidnight"



### (1) UIView 界面设置

- 黑色背景，红色字体，让它看起来像一个闹钟。

双击文件: "MinutesToMidnightViewController.xib" ;

然后 "Interface Builder" 会自动打开，在这里我们可以编辑改变界面

选择: Tools -> Reveal In Document Window -> View

选择: Tools -> Attributes Inspector

在色条内选择 "黑色"，可以看到背景变为黑色

### (2) 加入 UILabel 显示我们的倒数时间

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Label 到 Main View

在主显示上点击 Label; 从 Label Attributes 上选着字体的颜色和字体大小。

### **(3) 把界面文件更新到我们的代码文件上面**

在 MinutesToMidnightViewController.xib 文件窗口, 选择: File's Owner (这步很关键, 我在第一次编辑时没成功就因为忘了这一步)

选择: Tools->Identity Inspector

在 Identity inspector 中间 "Class Outlets" 按 "+";

把 myOutlet1 改为 "countdownLabel";

把 id 改为 "UILabel"; 按回车, 确定已更改。

### **(4) 写入 "MinutesToMidnightViewController.h" 和 "MinutesToMidnightViewController.m"**

确认还在 MinutesToMidnightViewController.xib 文件窗口, 选择: File's Owner

选择: File->Write Class Files

在 Save As: "MinutesToMidnightViewController"

点击: Save

点击: Merge

在右下角 "Actions" 按下选 (bottom right) "Select Left"

点击: 上方红色小圈关闭窗口

点击: Save

关闭所有 merge 窗口

打开 xCode; 选择 Build->Build

#### (5)从 Interface Builder 写入 UILabel 的 class file

再打开 SDK 工具 Interface Builder

在主视窗口或文件窗口; 点击 Label

选择: Tools -> Connection Inspector

移动鼠标在"New Referencing Outlet" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"File's Owner";

放开鼠标选择键出现 "countdownLabel"; 选上它。

选择: File -> Save then close Interface Builde

以上几步, 已成功完成界面了, 下面我们继续把程式添加我们的代码文件。

#### (6) 在 xCode 打开 MinutesToMidnightAppDelegate.h 文件

// 没修改前 //

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

```
@class MinutesToMidnightViewController;
```

```
@interface MinutesToMidnightAppDelegate : NSObject <UIApplicationDelegate> {
```

```
    UIWindow *window;
```

```
    MinutesToMidnightViewController *viewController;
```

```
}
```

```

@property (nonatomic, retain) IBOutlet UIWindow *window;

@property (nonatomic, retain) IBOutlet MinutesToMidnightViewController *viewController;

@end

// 没修改前 //

// 修改后 //

#import <UIKit/UIKit.h>

@class MinutesToMidnightViewController;

@interface MinutesToMidnightAppDelegate : NSObject <UIApplicationDelegate> {

    IBOutlet UIWindow *window;

    IBOutlet MinutesToMidnightController *viewController;

    NSTimer *timer;

}

-(void)onTimer;

@property (nonatomic, retain) IBOutlet UIWindow *window;

@property (nonatomic, retain) IBOutlet MinutesToMidnightViewController *viewController;

@end

```

#### (7) 在 xCode 打开 MinutesToMidnightAppDelegate.m 文件

```

// 没修改前 //

#import "MinutesToMidnightAppDelegate.h"

#import "MinutesToMidnightViewController.h"

@implementation MinutesToMidnightAppDelegate

@synthesize window;

@synthesize viewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {

    // Override point for customization after app launch

```

```

        [window addSubview:viewController.view];

        [window makeKeyAndVisible];
    }

- (void)dealloc {

    [viewController release];

    [window release];

    [super dealloc];
}

@end

// 没修改前 //

// 修改后 //

#import "MinutesToMidnightAppDelegate.h"

#import "MinutesToMidnightViewController.h"

@implementation MinutesToMidnightAppDelegate

@synthesize window;

@synthesize viewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {

    timer = [NSTimer scheduledTimerWithTimeInterval:(1.0) target:self selector:@selector(onTimer) userInfo:nil
repeats:YES]; // 不换行

    // Override point for customization after app launch

    [window addSubview:viewController.view];

    [window makeKeyAndVisible];
}

- (void)onTimer {

    [viewController updateLabel];
}

```

```

- (void)applicationWillTerminate:(UIApplication *)application {

    [timer invalidate];

}

- (void)dealloc {

    [timer release];

    [viewController release];

    [window release];

    [super dealloc];

}

@end

// 修改后 //

```

#### (8) 在 xCode 打开 MinutesToMidnightViewController.m 文件

```

// 修改前 //

#import "MinutesToMidnightViewController.h"

@implementation MinutesToMidnightViewController

- (void)didReceiveMemoryWarning {

    [super didReceiveMemoryWarning];

}

- (void)dealloc {

    [super dealloc];

}

@end

// 修改前 //

// 修改后 //

#import "MinutesToMidnightViewController.h"

```

@implementation **MinutesToMidnight**ViewController

```
- (void)viewDidLoad {

    [countdownLabel setFont:[UIFont fontWithName:@"DBLCDTempBlack" size:128.0]];

    countdownLabel.text = @"I0A0IN6";

    [super viewDidLoad];

}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {

    return (interfaceOrientation == UIInterfaceOrientationPortrait);

}

- (void)didReceiveMemoryWarning {

    [super didReceiveMemoryWarning]; // Releases the view if it doesn't have a superview

    // Release anything that's not essential, such as cached data

}

- (void)dealloc {

    [super dealloc];

}

-(void)updateLabel {

    NSDate* now = [NSDate date];

    int hour = 23 - [[now dateWithCalendarFormat:nil timeZone:nil] hourOfDay];

    int min = 59 - [[now dateWithCalendarFormat:nil timeZone:nil] minuteOfHour];

    int sec = 59 - [[now dateWithCalendarFormat:nil timeZone:nil] secondOfMinute];

    countdownLabel.text = [NSString stringWithFormat:@"%02d:%02d:%02d", hour, min, sec];

}
```

@end

// 修改后 //

最后在 xCode 选择 Build->Build and Go

下载今天程序文件: [MinsToMidnight.zip](#)

实例 iPhone SDK 编程入门教程, 来源于 <http://www.appsamuck.com/>, 大家有空可以去看看原文;

下面文章本人根据 appsamuck 这个网站加以翻译,修改及测试, 希望大家多多交流。一切版权归

<http://blog.sina.com.cn/iphonesdk> 所有。

## 实例 iPhone SDK 编程入门教程 - 第二天

### Day Two Bonfire

今天试试运用 UIImageView 来制作一个"营火动画"

首先运行以安装好的 xCode,和第一天一样

选择: File->New Project.

从 "New Project" 窗口

选择: iPhone OS ->Applications-> View-Based Application

命名: 我这里命名为"Bonfire"

(1)在 xcode 打开 BonfireViewController.m 文件

在-(void)viewDidLoad{ }里面添加代码, 如下

```
-(void)viewDidLoad{
```





```
// create the view that will execute our animation

UIImageView* campFireView = [[UIImageView alloc]
initWithFrame:self.view.frame];


// load all the frames of our animation
campFireView.animationImages = [NSArray arrayWithObjects:

                                [UIImage
imageNamed:@"campFire01.gif"],

                                [UIImage
imageNamed:@"campFire02.gif"],

                                [UIImage
imageNamed:@"campFire03.gif"],

                                [UIImage
imageNamed:@"campFire04.gif"],

                                [UIImage
imageNamed:@"campFire05.gif"],

                                [UIImage
imageNamed:@"campFire06.gif"],

                                [UIImage
imageNamed:@"campFire07.gif"],

                                [UIImage
imageNamed:@"campFire08.gif"],

                                [UIImage
imageNamed:@"campFire09.gif"],

                                [UIImage
imageNamed:@"campFire10.gif"],

                                [UIImage
imageNamed:@"campFire11.gif"],

                                [UIImage
imageNamed:@"campFire12.gif"],
```

```

                                [UIImage
imageName:@"campFire13.gif"],

                                [UIImage
imageName:@"campFire14.gif"],

                                [UIImage
imageName:@"campFire15.gif"],

                                [UIImage
imageName:@"campFire16.gif"],

                                [UIImage
imageName:@"campFire17.gif"], nil];

// all frames will execute in 1.75 seconds
campFireView.animationDuration = 1.75;

// repeat the annimation forever
campFireView.animationRepeatCount = 0;

// start animating
[campFireView startAnimating];

// add the animation view to the main window
[self.view addSubview:campFireView];
}

```

## (2) 下载图片文件 [images.zip](#)

把下载好的文件解压储存在与程序同一文件夹里

在 xcode - Group & Files - 在文件夹 Resource 上按鼠标右键 Add->Existing files 选择下载好的 Default.png 图片;

同上在文件夹 Resource 上按鼠标右键 Add-> New Group, 命名为 images;

在 文件夹 images 上按鼠标右键 Add->Existing files 选择下载好的在 images 里的所有图片。

最后在 xCode 选择 **Build->Build and Go**, **save all** 保存所有文件

下载今天程序文件: [Bonfire.zip](#)

## 实例 iPhone SDK 编程入门教程 - 第三天

### DAY THREE - OPEN URL

大家一起来建立一个 iPhone app, 给你的 iPhone 打开 google 地图有效地址连接。

首先运行以安装好的 xCode

选择: File->New Project.

从 "New Project" 窗口

选择 : iPhone OS ->Applications-> View-Based Application

命名 : 我这里命名为 "OpenURL"



**(1)** 在 xCode 打开 **OpenURLViewController.h** 文件,加入下面红色代码

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

```
@interface OpenURLViewController : UIViewController{
```

```
}
```

```
-(IBAction)openMaps;
```

```
@end
```

**(2)** 在 xCode 打开 **OpenURLViewController.m** 文件,加入下面红色代码

```
#import <UIKit/UIKit.h>#import "Day3ViewController.h"

@implementation Day3ViewController

// The designated initializer. Override to perform setup that is required before the view is loaded.

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {

    if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {

        // Custom initialization

    }

    return self;
}

- (IBAction)openMaps{

    NSString* addressText = @"1 Queen st, Auckland, NZ";

    addressText = [addressText stringByAddingPercentEscapesUsingEncoding: NSASCIIStringEncoding];

    NSString* urlText = [NSString stringWithFormat:@"http://maps.google.com/maps?q=%@",addressText];

    //NSlog(urlText);

    [[UIApplication sharedApplication] openURL:[NSURL URLWithString:urlText]];

}


// Override to allow orientations other than the default portrait orientation.

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {

    // Return YES for supported orientations

    return (interfaceOrientation == UIInterfaceOrientationPortrait);

}

- (void)didReceiveMemoryWarning {

    [super didReceiveMemoryWarning]; // Releases the view if it doesn't have a superview

    // Release anything that's not essential, such as cached data
```

```
}  
  
- (void)dealloc {  
  
    [super dealloc];  
  
}  
  
@end
```

### (3) UIView 界面设置

- 黑色背景，红色字体，让它看起来像一个闹钟。

双击文件: "OpenURLViewController.xib" ;

然后 "Interface Builder" 会自动打开，在这里我们可以编辑改变界面

选择: Tools -> Reveal In Document Window -> View

选择: Tools -> Attributes Inspector

在色条内选择 "黑色", 可以看到背景变为黑色

### (4) 加入 Button ,点击后打开连接

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主显示上点击 Button; 从 Button Attributes 里面 Title 内填上 "Open Google Map".

### (5)从 Interface Builder 写入 UIButton 的 class file

再打开 SDK 工具 Interface Builder

在主视窗口或文件窗口；点击 **Button**

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上；圆圈变为(+); 拖向直线连接到"File's Owner";

放开鼠标选择键出现 "openMaps"; 选上它。

选择: File -> Save then close Interface Builde

最后在 xCode 选择 Build->Build and Go

下载今天程序文件: [OpenURL.zip](#)

## 实例 iPhone SDK 编程入门教程 – 第四天

### DAY Four – What is my IP?

大家一起来建立一个 iPhone app，给你的 iPhone 显示 现时的 ip 地址。

首先运行以安装好的 **xCode**

选择: File->New Project.

从 "**New Project**" 窗口



选择 : iPhone OS ->Applications-> View-Based Application

命名 : 我这里命名为 "MYIP"

**(1) 在 xCode 打开 MYIPViewController.h 文件,加入下面红色代码**

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

@interface OpenURLViewController : UIViewController{

    IBOutlet UILabel *showIPLabel;

}

@end
```

**(2) 在 xCode 打开 MYIPViewController.m 文件,加入下面红色代码**

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

@implementation MYIPViewController

// The designated initializer. Override to perform setup that is required before the view is loaded.

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {

    if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {

        showIPLabel.text = @"";

    }

    return self;

}

-(NSString *)getAddress {

    char iphone_ip[255];

    strcpy(iphone_ip,"127.0.0.1"); // if everything fails

}
```

```

        NSString* myhost =[NSHost currentHost];

        if (myhost)

        {

            NSString *ad = [myhost address];

            if (ad)

                strcpy(iphone_ip,[ad cStringUsingEncoding:NSUTF8StringEncoding]);

        }

        return [NSString stringWithFormat:@"%s",iphone_ip];

    }

    - (void)viewDidLoad {

        NSString* address = [self getAddress];

        NSString* myIPAddress = [NSString stringWithFormat:@"IP Address: %@", address];

        showIPLabel.text = myIPAddress;

        [super viewDidLoad];

    }

    // Override to allow orientations other than the default portrait orientation.
    - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {

        // Return YES for supported orientations

        return (interfaceOrientation == UIInterfaceOrientationPortrait);

    }

    - (void)dealloc {

```



```
[super dealloc];  
  
}  
  
@end
```

### (3) UIView 界面设置

双击文件: "MYIPViewController.xib" ;

然后 "Interface Builder" 会自动打开, 在这里我们可以编辑改变界面

### (4) 加入 Label ,显示 iPhone 的 IP 地址

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Label 到 Main View

### (5)从 Interface Builder 写入 UILabel 的 class file

SDK 工具 Interface Builder

在主视窗口或文件窗口; 点击 Label

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"File's Owner";

放开鼠标选择键出现 "showIPLabel"; 选上它。

选择: File -> Save then close Interface Builde

以上几步,已成功完成界面了,下面我们继续把程式添加我们的代码文件。

最后在 xCode 选择 Build->Build and Go

下载今天程序文件: [WhatIsMyIp.zip](#)

## 实例 iPhone SDK 编程入门教程 – 第五天

### DAY Five – Count Me In

今天给你的 iPhone 制作一个简单的数数程序。

纲要:

—在程序显示前运行代码;

—UIButton, UILabel, UIImageView 的运用;

—把 integer 转换为 string;

首先运行以安装好的 **xCode**

选择: File->New Project.

从 "New Project" 窗口

选择 : iPhone OS ->Applications-> View-Based Application

命名 : 我这里命名为 "CountMeIn"



**(1)** 在 **xCode** 打开 **CountMeInViewController.h** 文件,加入下面代码

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

```
@interface CountMeInViewController : UIViewController {
```

```
    IBOutlet UILabel *counter;
```

```
}
```

```
-(IBAction)reset;
```

```
-(IBAction)addUnit;
```

```
-(IBAction)subtractUnit;
```

```
@end
```

**(2)** 在 **xCode** 打开 **CountMeInViewController.m** 文件,加入下面代码

```
#import "CountMeInViewController.h"
```

```
@implementation CountMeInViewController
```

```
int count = 0;
```

```
-(IBAction)reset{
```

```
    count = 0;
```

```
    counter.text = @"0";
```

```
}
```

```
- (IBAction)addUnit {
```

```
    if(count >= 999) return;
```

```
    NSString *numValue = [[NSString alloc] initWithFormat:@"%d", count++];
```

```
    counter.text = numValue;
```

```
    [numValue release];
```

```
}
```

```
- (IBAction)subtractUnit {
```

```

        if(count <= 0) return;

        NSString *numValue = [[NSString alloc] initWithFormat:@"%d", count--];

        counter.text = numValue;

        [numValue release];

    }

    // The designated initializer. Override to perform setup that is required before the view is loaded.

    - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {

        if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {

            // Custom initialization

        }

        return self;

    }

    // Implement viewDidLoad to do additional setup after loading the view, typically from a nib.

    - (void)viewDidLoad {

        counter.text = @"0";

        [super viewDidLoad];

    }

    // Override to allow orientations other than the default portrait orientation.

```

```

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {

    // Return YES for supported orientations

    return (interfaceOrientation == UIInterfaceOrientationPortrait);

}

- (void)didReceiveMemoryWarning {

    [super didReceiveMemoryWarning]; // Releases the view if it doesn't have a superview

    // Release anything that's not essential, such as cached data

}

- (void)dealloc {

    [super dealloc];

}

@end

```

### (3) 导入下面图片文件

另存下面图片，放入 **countMeIn** 文件夹内并命名为下面名称

***backgroundPattern***

***IconGreenAdd***

***IconRedSubtract***



在 xCode 下右键点击 **CountMeIn->Add->Existing Files**; 在 **countMeIn** 文件夹内,选择下载好的三个图片,按 **Add**

#### (4) UIView 界面设置

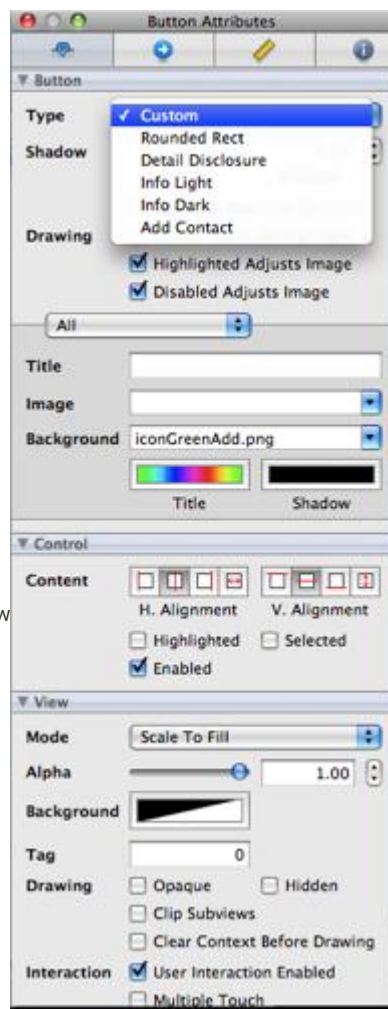
双击文件: "CountMeInViewController.xib" ;

然后 "Interface Builder" 会自动打开, 在这里我们可以编辑改变界面

#### (5) 加入 Label ,显示程序计算结果

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Label 到 Main View

在主视窗口或文件窗口; 点击 Label



选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"File's Owner";

放开鼠标选择键出现 "counter"; 选上它。

## (6) 加入 Add Button , 进行累加计算

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口: 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"File's Owner";

放开鼠标选择键出现 "addUnit"; 选上它。

选择: Tools -> Attributes Inspector

在 Type 下选择 custom; 在 Background 下选择 iconGreenAdd

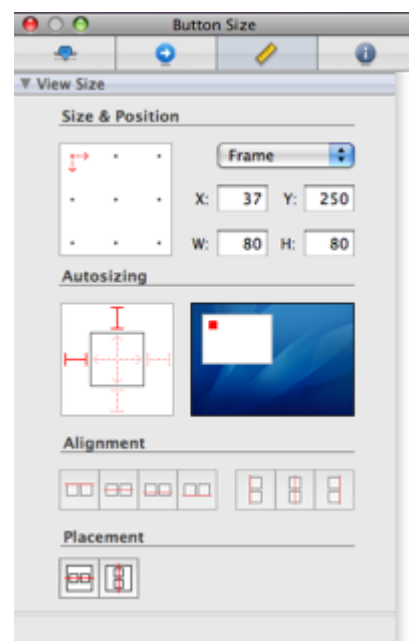
选择: Tools -> Size Inspector

调整 Size & Position

## (7) 加入 Subtract Button , 进行累减计算

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口: 点击 Button





选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"File's Owner";

放开鼠标选择键出现 "SubtractUnit"; 选上它。

选择: Tools -> Attributes Inspector

在 Type 下选择 custom; 在 Background 下选择 iconRedSubtract

选择: Tools -> Size Inspector

调整 Size & Position

## (7) 加入 Reset Button , 进行累减计算

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

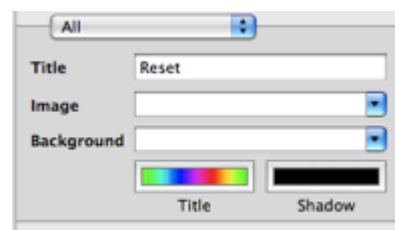
移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到

"File's Owner";

放开鼠标选择键出现 "Reset"; 选上它。

选择: Tools -> Attributes Inspector

在 Title 下填上 Reset



#### (8) 加入 UIImageView , 背景图案

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 imageView 到 Main View; 调整到全屏

在主视窗口或文件窗口; 点击 imageView

选择: Tools -> Inspector; 在 **image** 下选择 backgroundImage.png

选择: Tools -> Layout -> Send To Back; 图片设为背景

最后在 xCode 选择 **Build->Build and Go; Save All.**

下载今天程序文件: [CountMeIn.zip](#)

## 实例 iPhone SDK 编程入门教程 - 第六天

## 考反映小游戏

今天大家一起来建立一个 iPhone app，给你的 iPhone 制作一个简单的考反应游戏。

纲要：

- 在程序显示前运行代码
- UIButton, UILabel, UIImageView 的运用；
- 利用 random 增加游戏可玩性；
- 关于 iPhone 的“Utility Application”。

首先运行以安装好的 xCode

选择: File->New Project.

从 "New Project" 窗口

选择 : iPhone OS

->Applications-> Utility Application





命名：我这里命名为“ReactionTime”

指示灯的反应程序编写

在按下开始游戏后，分为三盏的指示灯按照“黄”、“红”、“绿”的先后顺序各自读取相应的指示灯颜色图片文件，当黄灯亮后游戏使用随机变量产生没有规律的红灯持续时间，因为这样的红灯使玩家没法预计绿灯什么时候亮，所以按下油门的时间要看玩家的反应速度，油门按下后，游戏会把从绿灯亮起直到玩家按下油门之间所使用的时间显示新窗口，并且提示玩家时候挑战更好的成绩，下面我们来看看怎么用代码来实现这样的游戏。

(1) 在 xCode 打开 **MainView.h** 文件,加入下面代码

创建时间对象(**NSDate**)和信号灯图像对象(**IBOutlet UIImageView**)

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

#import <Foundation/Foundation.h>

@interface MainView : UIView {

    NSDate *startDate;

    //NSDate: 单一点的时间数值类的对象

    IBOutlet UIImageView *stopLight;

    //IBOutlet UIImageView: 把信号灯转换的图片对象告诉 Interface Builder

}

//-----

@property (nonatomic, copy) NSDate *startDate;

    // 日子选取器 (@property(nonatomic, copy) NSDate *date)

    // nonatomic: 使用单线程机制减少系统资源使用

    // copy: 建立时间开始数值为文字类型(NSString)的对象

- (IBAction)gasPedalPressed;

    //油门按钮动作对象(IBAction); IBAction: 连接 Interface Builder 中把方法

(method)作为行为(Action)

@end
```

(2) 在 xCode 打开 **MainView.m** 文件,加入下面代码

```
#import "MainView.h"
```

```
@implementation MainView
```

```
@synthesize startDate;
```

```
// 本 Synthesize 指令为程序自动生成 startDate 设置函数(setters)和获得函数(getter)
```

```
int greenLightOn = 0;
```

```
-(void)awakeFromNib {
```

```
    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Reaction Time: Ready to Play"
```

```
    message:@"当绿灯亮时以最快的速度按下脚踏板."
```

```
    delegate:self cancelButtonTitle:@"游戏开始" otherButtonTitles: nil];
```

```
    [alert show];
```

```
}
```

```
//当 Interface Builder 的档案读取后,准备好所有的 项目 (Object),
```

```
//它使游戏刚运行就可以读取显示消息框
```

```
// (UIAlertView) 内定义好的游戏开始的信息和按钮。
```

```
//UIAlertView *alert: 定义消息框内容
```

```
//UIAlertView alloc : 配置消息框内容
```

```
//initWithTitle : 消息框为“准备开始游
```

```
戏”
```

```
//message : 消息框内详细消息为“当绿灯亮时以最快的速度按下脚踏板”

//delegate:self; 代表自己

//cancelButtonTitle: 取消消息框按钮为“游戏开始”

//otherButtonTitles: nil; 没有其他按钮

//[alert show]; 显示消息框

//-----

- (void)alertView:(UIAlertView *)alertView didDismissWithButtonIndex:(NSInteger)buttonIndex
//在点击消息框(UITableView)内的按钮，消息框将消失并运行此程序里的内容，
{
    stopLight.image = [UIImage imageNamed:@"yellowLightSmall.png"];//指示灯的图片定制为黄灯图片

    greenLightOn = 0;//数值为 0，绿灯关闭

    [NSTimer scheduledTimerWithTimeInterval:(3.0) target:self
    selector:@selector(onYellowLightTimer) userInfo:nil repeats:NO];
}

//stopLight.image:

//NSTimer 黄灯的时间定制

//scheduledTimerWithTimeInterval:(3.0) 预定的时间为 3，时间间隔为 0

//target:self 对象为自己

//selector:@selector(onYellowLightTimer) 选择者为黄灯时间

//userInfo:nil 没有用户信息
```

```

        //repeat:no 不重复运行

//-----

- (void)onYellowLightTimer //游戏中红灯的时间长度定制随机数(random)运用
{
    stopLight.image = [UIImage imageNamed:@"redLightSmall.png"];

    int delay = ((int) (random() % 7) + 1);

    [NSTimer scheduledTimerWithTimeInterval:(3.0 + delay) target:self
    selector:@selector(onRedLightTimer) userInfo:nil repeats:NO];
}

    //int delay = ((int) (random() % 7) + 1); 定制数字变量为随机数字除以 7 加 1

    //NSTimer

    //scheduledTimerWithTimeInterval:(3.0+delay) 预定的时间为 3，时间间隔
为随机

    //target:self 对象为自己

    //selector:@selector(onRedLightTimer) 选择者为红灯时间

    //userInfo:nil 没有用户信息

    //repeat:no 不重复运行

//-----

- (void)onRedLightTimer //游戏中绿灯的时间长度定制
{
    stopLight.image = [UIImage imageNamed:@"greenLightSmall.png"];

    // stopLight.image: 指示灯的图片定制为绿灯图片

    greenLightOn = 1;//绿灯打开

```



```

    self.startDate = [NSDate date]; //反应时间导入，开始计算时间
}

//-----

- (IBAction)gasPedalPressed
{
    double noSeconds = (double) [self.startDate timeIntervalSinceNow] * -1000;

    NSString *reactionTime= [[NSString alloc] initWithFormat:@"好样的! 你的响应速度
是 %1.0f 毫秒. 再来一次, 创造更好的成绩...", noSeconds];

    if(greenLightOn == 0)
        reactionTime = @"请不要急. 要等到绿灯亮时才按脚踏板, 再来一次";

    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Reaction Time"
message:reactionTime
                                delegate:self cancelButtonTitle:@"确定"
                                otherButtonTitles: nil];
    [alert show];
}

```

//double noSeconds : 定制带小数点的数值

//self.startDate timeIntervalSinceNow : 时间开始数值为按下按钮后

//\*-1000 : 以毫秒显示

//NSString \*reactionTime : 定制文字字符串值

//[NSString alloc] : 向文字字符串值配置(alloc)文字

//initWithFormat : 初始化格式文字导入

```
//if(greenLightOn == 0) : 如果在绿灯关闭的时候按下按钮
```

```
//-----
```

```
@end
```

### (3) 导入下面图片文件

下载下面图片，放入 ReactionTime 文件夹内并命名为下面名称

gasPedalSmall.png



greenLightSmall.png



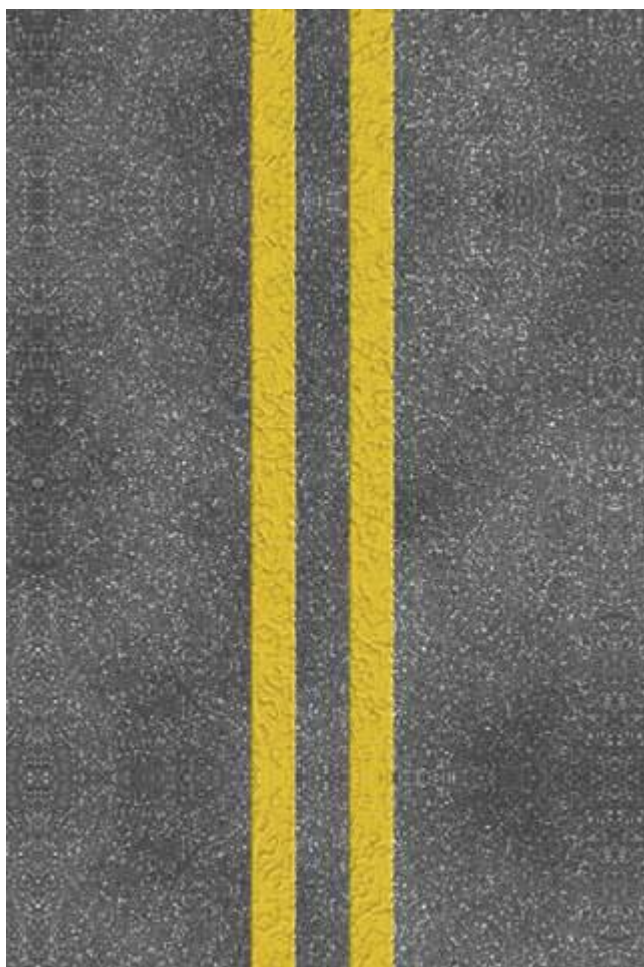
yellowLightSmall.png



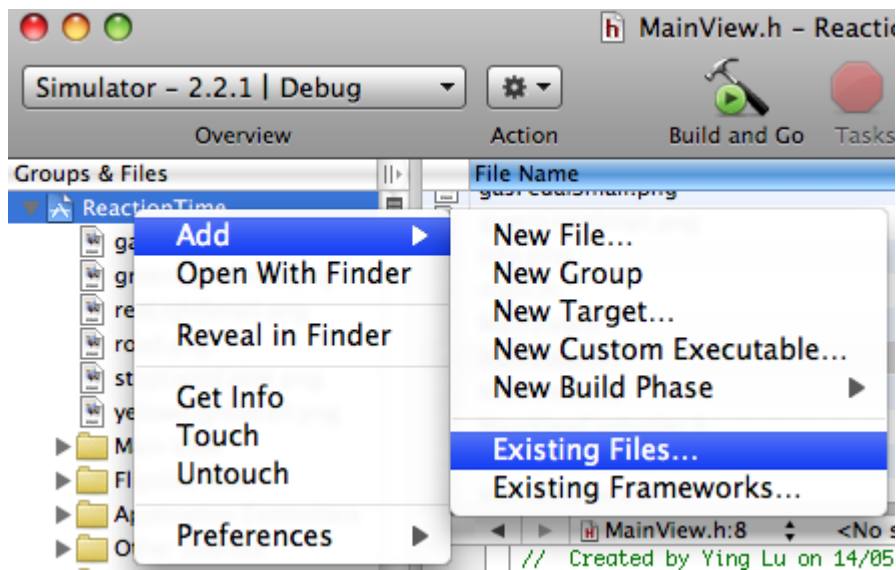
redLightSmall.png



road.png



在 xCode 下右键点击 ReactionTime->Add->Existing Files; 在 Reaction 文件夹内, 选择下载好的图片,  
按 Add



#### (4) UIView 界面设置

双击文件: "main.xib" ;

然后 "Interface Builder" 会自动打开, 在这里我们可以编辑改变界面

#### (5) 加入 Add Button

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"File's Owner";

放开鼠标选择键出现 "gasPedalPressed"; 选上它。

选择: Tools -> Attributes Inspector

在 Type 下选择 custom; 在 Background 下选择 gasPedalSmall.png

选择: Tools -> Size Inspector

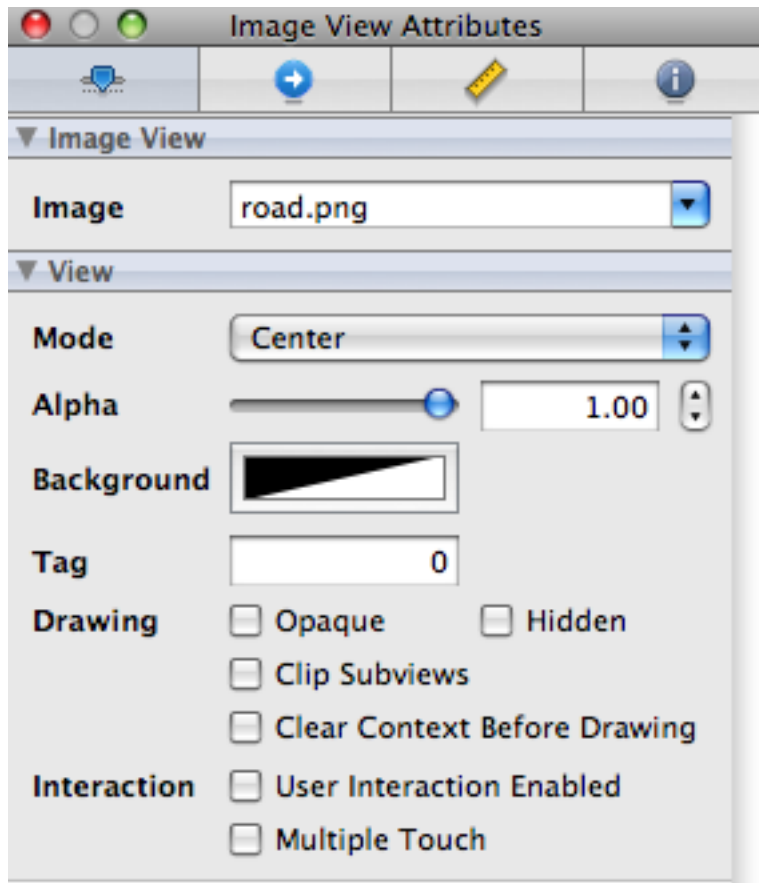
调整 Size & Position

#### (6) 加入 UIImageView , 交通灯图片

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 imageView 到 Main View;

在主视窗口或文件窗口; 点击 imageView

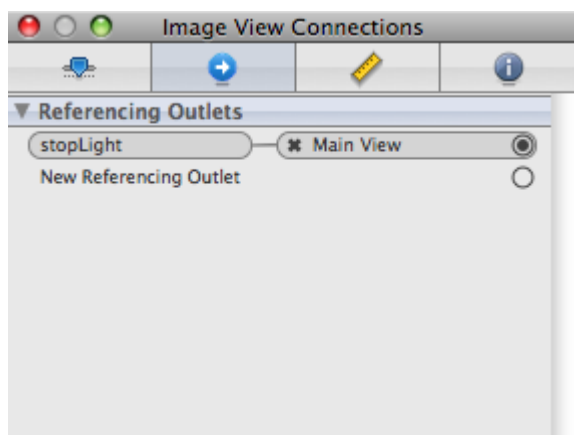
选择: Tools -> Inspector; 在 image 下选择 road.png



选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"File's Owner";

放开鼠标选择键出现 "stopLight"; 选上它。



### (7) 加入 UIImageView , 背景图案

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 imageView 到 Main View; 调整到全屏

在主视窗口或文件窗口; 点击 imageView

选择: Tools -> Inspector; 在 image 下选择 road.png

选择: Tools -> Layout -> Send To Back; 图片设为背景

最后在 xCode 选择 Build->Build and Go; Save All.

下载今天程序文件: [ReactionTime.zip](#)

## 实例 iPhone SDK 编程入门教程 - 第七天

### DAY SEVEN – Settings Bundle

今天建立一个 iPhone app, 运用 Settings Bundle 设置运行程序

纲要:

- 在程序显示前运行代码;
- UILabel 的运用;
- 把 Settings Bundle 加入程序;



首先运行以安装好的 xCode

选择: File->New Project. 从 "New Project" 窗口

选择 : iPhone OS ->Applications-> View-Based Application

命名 : 这里命名为 "SettingsBundle"

(1) 在 xCode 打开 File -> New File.. ;

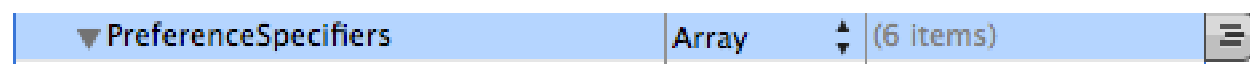
选择 Settings Bundle; 命名为 Settings.bundle;



(2) 在 Setting.bundle 双击 Root.plist



把 PreferenceSpecifiers 前的三角向下, 按下 , 加入:



Item1,

▼ Item 1	Dictionary ▴ ▾	(8 items)
Type	String	PSTextFieldSpecifier
Title	String	Text Entry
Key	String	textEntry_key
DefaultValue	String	Simple Text
IsSecure	Boolean	<input type="checkbox"/>
KeyboardType	String	Alphabet
AutocapitalizationType	String	Sentences
AutocorrectionType	String	Default

Item2,

▼ Item 2	Dictionary ▴ ▾	(4 items)
Type	String	PSTitleValueSpecifier
Title	String	Read Only
Key	String	readOnly_key
DefaultValue	String	Read Only Value

Item3,

▼ Item 3	Dictionary ▴ ▾	(6 items)
Type	String	PSToggleSwitchSpecifie
Title	String	Toogle
Key	String	toogle_key
DefaultValue	String	YES
TrueValue	String	YES
FalseValue	String	YES

Item4,

▼ Item 4	Dictionary ▴ ▾	(5 items)
Type	String	PSSliderSpecifier
Key	String	slider_key
DefaultValue	Number	1
MinimumValue	Number	1
MaximumValue	Number	100

Item5,



▼ Item 5	Dictionary ▴ ▾	(2 items)
Type	String	PSGroupSpecifier
Title	String	Group2

Item6,

▼ Item 6	Dictionary ▴ ▾	(6 items)
Type	String	PSMultiValueSpecifier
Title	String	Colors
Key	String	colors_key
DefaultValue	String	1
▼ Values	Array	(3 items)
Item 1	String	1
Item 2	String	2
Item 3	String	3
▼ Titles	Array	(3 items)
Item 1	String	Blue
Item 2	String	Red
Item 3	String	Green

在 xCode 打开 File -> Save;

(3) 在 xCode 打开 SettingBundleViewController.h 文件,加入代码

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

```
@interface SettingsBundleViewController : UIViewController {
```

```
IBOutlet UILabel *lblText;
```

```
IBOutlet UILabel *lblReadOnly;
```

```
IBOutlet UILabel *lblSlider;
```

```

IBOutlet UILabel *lblColor;

IBOutlet UILabel *lblToggle;

}

@property (nonatomic, retain) UILabel *lblText;
@property (nonatomic, retain) UILabel *lblReadOnly;
@property (nonatomic, retain) UILabel *lblSlider;
@property (nonatomic, retain) UILabel *lblColor;
@property (nonatomic, retain) UILabel *lblToggle;

@end

```

**(4) 在 xCode 打开 SettingsBundleViewController.m 文件,加入下面代码**

```

#import "SettingsBundleViewController.h"

@implementation SettingsBundleViewController

// The designated initializer. Override to perform setup that is required before the view is loaded.

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {

    if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {

        // Custom initialization

    }

    return self;
}

// Implement viewDidLoad to do additional setup after loading the view, typically from a nib.

- (void)viewDidLoad {

    NSString *textValue = [[NSUserDefaults standardUserDefaults] valueForKey:@"textEntry_key"];

    NSString *readOnlyValue = [[NSUserDefaults standardUserDefaults] valueForKey:@"readOnly_key"];

    NSString *sliderValue = [[NSUserDefaults standardUserDefaults] valueForKey:@"slider_key"];

```

```

NSString *colorValue = [[NSUserDefaults standardUserDefaults] valueForKey:@"colors_key"];

NSString *toggleValue = [[NSUserDefaults standardUserDefaults] valueForKey:@"toggle_key"];


lblText.text = [NSString stringWithFormat:@"Text Value: %@", textValue];

lblReadOnly.text = [NSString stringWithFormat:@"Read Only Value: %@", readOnlyValue];

lblSlider.text = [NSString stringWithFormat:@"Slider Value: %@", sliderValue];

lblColor.text = [NSString stringWithFormat:@"Selected color value: %@", colorValue];

lblToggle.text = [NSString stringWithFormat:@"Toggle Control Value: %@", toggleValue];


}

// Override to allow orientations other than the default portrait orientation.

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {

    // Return YES for supported orientations

    return (interfaceOrientation == UIInterfaceOrientationPortrait);

}


- (void)didReceiveMemoryWarning {

    [super didReceiveMemoryWarning]; // Releases the view if it doesn't have a superview

    // Release anything that's not essential, such as cached data

}


- (void)dealloc {

    [lblToggle release];

    [lblText release];

    [lblReadOnly release];

    [lblSlider release];

    [lblColor release];

    [super dealloc];

}

```

@end

### (5) UIView 界面设置

双击文件: "SettingsBundleViewController.xib" ;

然后 "Interface Builder" 会自动打开, 在这里我们可以编辑改变界面

选择: Tools -> Reveal In Document Window -> View

选择: Tools -> Attributes Inspector

在色条内选择 "白色", 可以看到背景变为黑色

### (3) 加入五个 Label

选择: Tools -> Label ; 从 Library 显示菜单中拖拉一个 Label 到

Main View

在 Interface Builder

在主视窗口或文件窗口; 点击 Label

选择: Tools -> Connection Inspector



移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"File's Owner";

放开鼠标选择键出现 "lblText"; 选上它。

其余四个 label 步骤同上分别选上 "lblReadOnly", "lblSlider", "lblToggle", "lblColor"

选择: File -> Save then close Interface Builde

最后在 xCode 选择 Build -> Build and Go

下载今天教程文件: [SettingsBundle.zip](#)

三十一天手把手 **iPhone app 基础**, 来源于 [www.appsamuck.com](http://www.appsamuck.com), 大家有空可以去看看原文;

文章本人根据 appsamuck 这个网站加以翻译,修改及测试,希望大家多多交流。一切版权归 <http://blog.sina.com.cn/iphonesdk> 所有。

## 实例 iPhone SDK 编程入门教程 - 第八天

### DAY EIGHT – Flick Map

今天建立一个 iPhone app, 给你的 iPhone 打开 Flick Map。

纲要:

- 在程序显示前运行代码;
- UIButton, UIImageView 的运用;
- 打开 Flick 连接;
- 关于 iPhone 的“Utility Application”。

首先运行以安装好的 xCode

选择: File->New Project.



从 "New Project" 窗口

选择 : iPhone OS ->Applications-> Utility Application

命名 : 我这里命名为 "FlickrMap"

(1) 在 xCode 打开 **MainView.h** 文件,加入下面代码

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

@interface MainView : UIView {

}

-(IBAction)loadFlickrKML;

@end
```

(2) 在 xCode 打开 **MainView.m** 文件,加入下面代码

```
#import "MainView.h"

@implementation MainView

- (IBAction)loadFlickrKML {

    NSString* kmlURL = @"http://www.flickr.com/map/china";

    [[UIApplication sharedApplication] openURL:[NSURL URLWithString: kmlURL]];

}

@end
```

### (3) 导入下面图片文件

下载下面图片，放入 ReactionTime 文件夹内并命名为下面名称

Pic.png



在 xCode 下右键点击 Flickr Map->Add->Existing Files; 在 Flickr Map 文件夹内,选择下载好的图片,按 Add

### (4) UIView 界面设置

双击文件: "main.xib" ;

然后 "Interface Builder" 会自动打开, 在这里我们可以编辑改变界面

### (5) 加入 Add Button

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+);

拖向直线连接到"Main View";



放开鼠标选择键出现 "loadFlickKML"; 选上它。

#### (6) 加入 UIImageView

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 imageView 到 Main View; 调整到全屏

在主视窗口或文件窗口; 点击 imageView

选择: Tools -> Inspector; 在 image 下选择 Pic.png

最后在 xCode 选择 Build->Build and Go; Save All.

下载今天教程文件: [FlickrMap.zip](#)

## 实例 iPhone SDK 编程入门教程 - 第九天

### DAY NINE - Snow Fall

今天来建立一个 iPhone app, 给你的 iPhone 制作一场雪景。

纲要:

- 在程序显示前运行代码 -
- UIImageView 的运用 -
- 关于 iPhone 的“Utility Application”运用 -
- onTimer 代码运用 -
- onAnimation 代码运用 -





首先运行以安装好的 xCode

选择: File->New Project.

从 "New Project" 窗口

选择 : iPhone OS ->Applications-> Utility Application

命名 : 我这里命名为 “SnowFall”

(1) 在 xCode 打开 **MainView.h** 文件,加入下面代码

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

@interface MainViewController : UIViewController {

    UIImage* flakeImage;

}

@property (nonatomic, retain) UIImage* flakeImage;

- (void)onTimer;

- (void)onAnimationComplete:(NSString *)animationID finished:(NSNumber *)finished context:(void *)context;

@end
```

(2) 在 xCode 打开 **MainView.m** 文件,加入下面代码

```
#import "MainViewController.h"

#import "MainView.h"

@implementation MainViewController

@synthesize flakeImage;
```

```
- (void)viewDidLoad {

    [super viewDidLoad];

    // 把背景颜色设置为冷色

    self.view.backgroundColor = [UIColor colorWithRed:0.5 green:0.5 blue:1.0 alpha:1.0];

    // 把雪花图片文件不停导出

    flakeImage = [UIImage imageNamed:@"flake.png"];

    // 在 onTimer 设置开始时间每秒二十次

    [NSTimer scheduledTimerWithTimeInterval:(0.05) target:self selector:@selector(onTimer) userInfo:nil
    repeats:YES];

}

// Timer event is called whenever the timer fires

- (void)onTimer

{

    //建立一个雪花图片 flake image

    UIImageView* flakeView = [[UIImageView alloc] initWithImage:flakeImage];

    //use the random() function to randomize up our flake attributes

    int startX = round(random() % 320);

    int endX = round(random() % 320);

    double scale = 1 / round(random() % 100) + 1.0;

    double speed = 1 / round(random() % 100) + 1.0;

    // set the flake start position

    flakeView.frame = CGRectMake(startX, -100.0, 25.0 * scale, 25.0 * scale);

    flakeView.alpha = 0.25;
```

```

// put the flake in our main view

[self.view addSubview:flakeView];

[UIView beginAnimations:nil context:flakeView];

// set up how fast the flake will fall

[UIView setAnimationDuration:5 * speed];

// set the position where flake will move to

flakeView.frame = CGRectMake(endX, 500.0, 25.0 * scale, 25.0 * scale);

// set a stop callback so we can cleanup the flake when it reaches the

// end of its animation

[UIView setAnimationDidStopSelector:@selector(onAnimationComplete:finished:context:)];

[UIView setAnimationDelegate:self];

[UIView commitAnimations];

}

- (void)onAnimationComplete:(NSString *)animationID finished:(NSNumber *)finished context:(void *)context {

    UIImageView *flakeView = context;

    [flakeView removeFromSuperview];

    // open the debug log and you will see that all flakes have a retain count

    // of 1 at this point so we know the release below will keep our memory

    // usage in check

    NSLog([NSString stringWithFormat:@"[flakeView retainCount] = %d", [flakeView retainCount]]);

    [flakeView release];

}

```

```

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {

    // Return YES for supported orientations

    return (interfaceOrientation == UIInterfaceOrientationPortrait);

}


- (void)didReceiveMemoryWarning {

    [super didReceiveMemoryWarning]; // Releases the view if it doesn't have a superview

    // Release anything that's not essential, such as cached data

}


- (void)dealloc {

    [flakeImage release];

    [super dealloc];

}

@end

```

### (3) 导入下面图片文件

下载下面图片，放入 SnowFall 文件夹内并命名为下面名称

flake.png



在 xCode 下右键点击 SnowFall->Add->Existing Files; 在 SnowFall 文件夹内，选择下载好的图片，按 Add

**最后在 xCode 选择 Build->Build and Go; Save All.**

# 实例 iPhone SDK 编程入门教程 – 第十天

(2009-05-31 14:53:45)

[转载](#)

标签: 分类: [实例 iPhoneSDK 入门](#)

[iphone](#)

[sdk](#)

[编程](#)

[iphone 游戏](#)

[iphone 软件编程](#)

[framework](#)

[添加](#)

[it](#)

## DAY TEN – WhereAmI

今天来建立一个 iPhone app 软件，用你的 iPhone 显示你现在的位置。

纲要:

- 在程序显示前运行代码 -
- 加入 CoreLocation Frameworks -
- 关于 iPhone 的“Utility Application” 运用 -
- CLLocationManager 代码运用 -

首先运行以安装好的 xCode

选择: File->New Project.

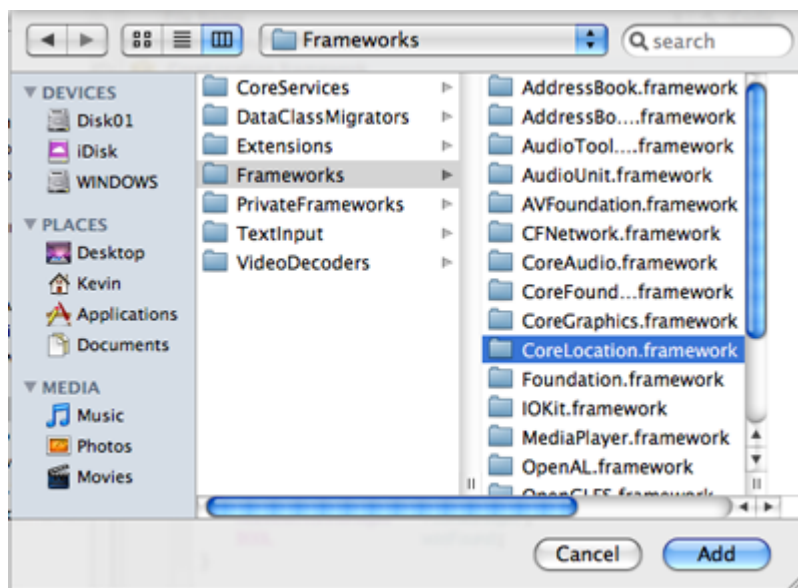
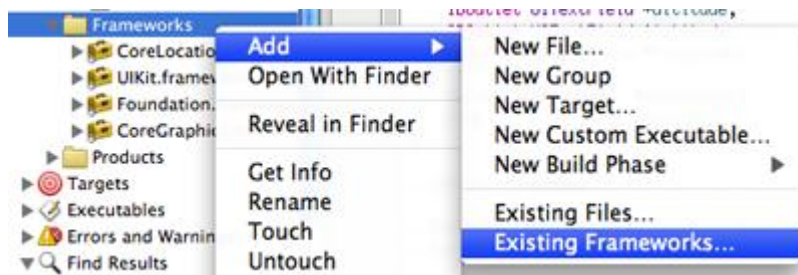


从 "New Project" 窗口

选择 : iPhone OS ->Applications-> Utility Application

命名 : 我这里命名为 "WhereAmI"

(1) 在 xCode 右键点击 Frameworks ->Add->Existing Framework;在 Frameworks 文件夹下选择 CoreLocation.framework, 按 Add



(2) 在 xCode 打开 MainView.h 文件,加入下面代码

```

#import <UIKit/UIKit.h>

#import <CoreLocation/CoreLocation.h>

#import <CoreLocation/CLLocationManagerDelegate.h>

@interface MainView : UIView {

    IBOutlet UITextField *altitude;

    IBOutlet UITextField *latitude;

    IBOutlet UITextField *longitude;

    CLLocationManager *locmanager;

    BOOL wasFound;

}

- (IBAction)update;

- (void)locationManager:(CLLocationManager *)manager didUpdateToLocation:(CLLocation *)newLocation
fromLocation:(CLLocation *) oldLocation ;

- (void)locationManager:(CLLocationManager *)manager didFailWithError:(NSError *) error;

@end

```

(3) 在 xCode 打开 MainView.m 文件,加入下面代码

```

#import "MainView.h"

@implementation MainView

- (IBAction)update {

    locmanager = [[CLLocationManager alloc] init];

    [locmanager setDelegate:self];

    [locmanager setDesiredAccuracy:kCLLocationAccuracyBest];

```

```

[locmanager startUpdatingLocation];

}

CLLocationManager* locmanager;

-(void)awakeFromNib {

[self update];

}

- (void)locationManager:(CLLocationManager *)manager didUpdateToLocation:(CLLocation *)newLocation
fromLocation:(CLLocation *)oldLocation
{

if (wasFound) return;

wasFound = YES;

CLLocationCoordinate2D loc = [newLocation coordinate];

latitude.text = [NSString stringWithFormat:@"%f", loc.latitude];

longitude.text = [NSString stringWithFormat:@"%f", loc.longitude];

altitude.text = [NSString stringWithFormat:@"%f", newLocation.altitude];

}

- (void)locationManager:(CLLocationManager *)manager didFailWithError:(NSError *)error
{

}

- (id)initWithFrame:(CGRect)frame {

if (self = [super initWithFrame:frame]) {

```



```
// Initialization code

}

return self;

}


- (void)drawRect:(CGRect)rect {

// Drawing code

}


- (void)dealloc {

[super dealloc];

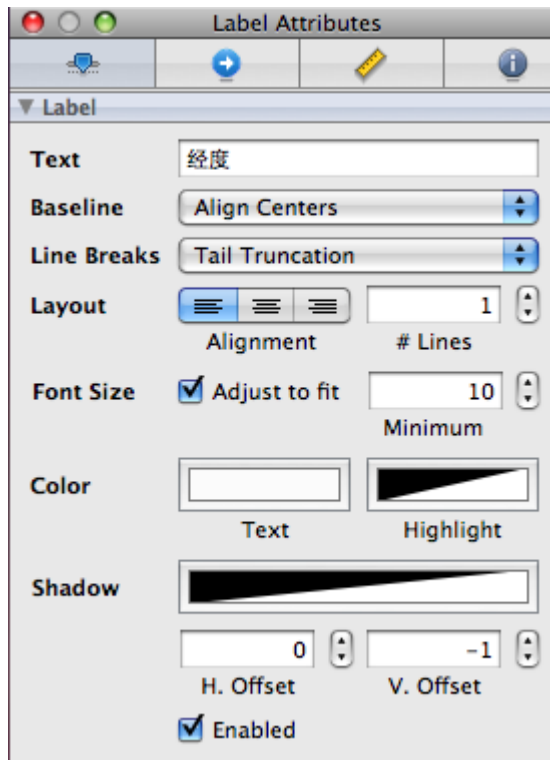
}
```

#### **(4) UIView 界面设置**

双击文件: "main.xib" ;

然后 "Interface Builder" 会自动打开, 在这里我们可以编辑改变界面

#### **(5) 加入 Label 在 Attributes 下, Text 内填上"经度"**



(6) 加入 Text Field ; 显示:经度

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Text Field 到 Main View

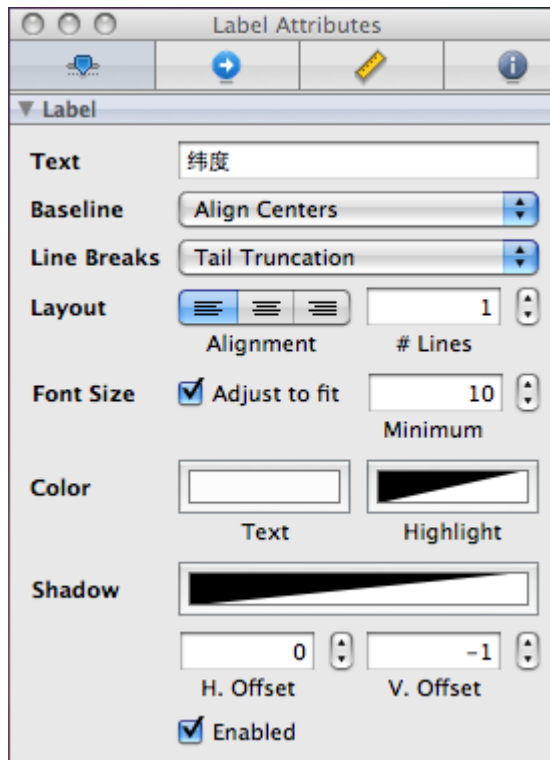
在主视窗口或文件窗口; 点击 Text Field

选择: Tools -> Connection Inspector

移动鼠标在"Referencing Outlets" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "longitude"; 选上它。

(7) 加入 Label 在 Attributes 下,Text 内填上"纬度"



(8) 加入 Text Field ; 显示:纬度

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Text Field 到 Main View

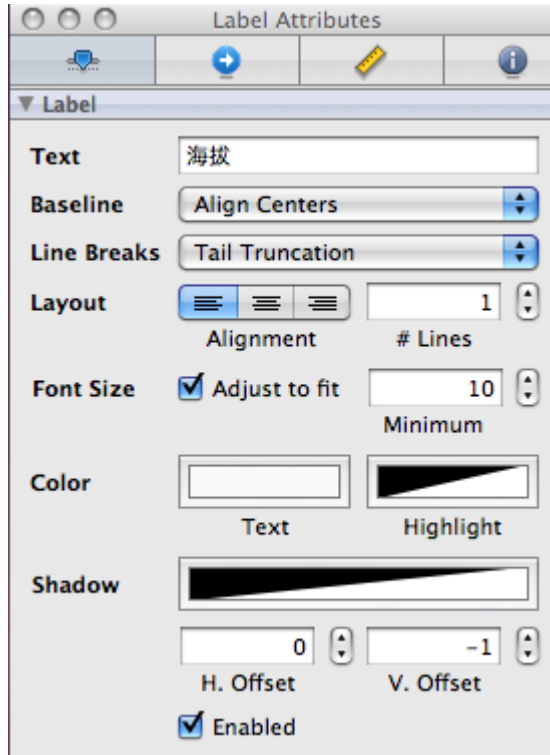
在主视窗口或文件窗口; 点击 Text Field

选择: Tools -> Connection Inspector

移动鼠标在"Referencing Outlets" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "latitude"; 选上它。

(9) 加入 Label 在 Attributes 下,Text 内填上"海拔".



**(10) 加入 Text Field ；显示:海拔**

选择: Tools -> Library ； 从 Library 显示菜单中拖拉一个 Text Field 到 Main View

在主视窗口或文件窗口； 点击 Text Field

选择: Tools -> Connection Inspector

移动鼠标在"Referencing Outlets" 后面圆圈上；圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "altitude"; 选上它。

**(11) 加入 UpDate Button 在 Attributes 下,Text 内填上"Update".**

选择: Tools -> Library ； 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口； 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "Update"; 选上它。

最后在 xCode 选择 **Build->Build and Go; Save All.**

下载今天教程文件: [WhereAmI.zip](#)

三十一天手把手 iPhone app 基础, 来源于 [www.appsamuck.com](http://www.appsamuck.com), 大家有空可以去看看原文;

下面文章本人根据 appsamuck 这个网站加以翻译,修改及测试, 希望大家多多交流。一切版权归

<http://blog.sina.com.cn/iphonesdk> 所有

## 实例 iPhone SDK 编程入门教程 - 第十一天

(2009-06-09 19:46:18)

转载

标签: 分类: [实例 iPhoneSDK 入门](#)

[iphone](#)

[sdk](#)

[编程](#)

[iphone 工具](#)

[iphone 软件编程](#)

[framework](#)

[添加](#)

[it](#)

[水平尺](#)

### DAY ELEVEN – PlumbBob



今天来建立一个 iPhone app 软件，用你的 iPhone 当作测量水平器。

纲要：

- 在程序显示前运行代码 -
- 加入 QuartzCore Frameworks -
- 关于 iPhone 的“Utility Application” 运用 -
- UIAccelerometer (加速度检波器) 代码运用 -

首先运行以安装好的 xCode

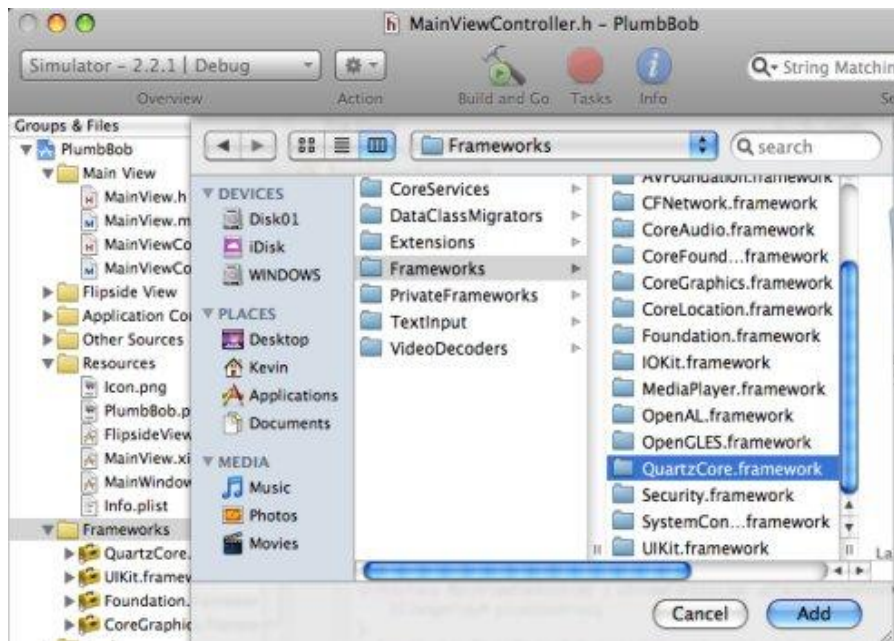
选择: File->New Project.

从 "New Project" 窗口

选择 : iPhone OS ->Applications-> Utility Application

命名 : 我这里命名为 “PlumbBob”

(1) 在 xCode 右键点击 Frameworks ->Add->Existing Framework;在 Frameworks 文件夹下选择 QuartzCore.framework, 按 Add



## (2) 导入下面图片文件

下载下面图片，放入 PlumbBob 文件夹内并命名为下面名称

PlumbBob.png



在 xCode 下右键点击 PlumbBob->Add->Existing Files; 在 PlumbBob 文件夹内,选择下载好的图片,按 Add

## (3) 在 xCode 打开 MainView.h 文件,加入下面代码

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

```
@interface MainViewController : UIViewController <UIAccelerometerDelegate> {
```

```
    UIImageView* plumbBobView;
```

```
}
```

```
- (void)rotatePlumbStringToDegree:(CGFloat)positionInDegrees;
```

```
@end
```

```
CGFloat DegreesToRadians(CGFloat degrees);
```

```
CGFloat RadiansToDegrees(CGFloat radians);
```

**(4) 在 xCode 打开 MainView.m 文件,加入下面代码**

```
#import "MainViewController.h"
```

```
#import "MainView.h"
```

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

```
// Constant for the number of times per second (Hertz) to sample acceleration.
```

```
#define kAccelerometerFrequency 40
```

```
@implementation MainViewController
```

```
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {
```

```
    if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {
```

```
        // Custom initialization
```

```
    }
```

```
    return self;
```

```
}
```

```
// Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
```

```
- (void)viewDidLoad {
```

```
    // 设置背景色为黑色
```

```
    self.view.backgroundColor = [UIColor blackColor];
```

```
    // 设置 "测锤" 图案
```



```

UIImage* image = [UIImage imageNamed:@"PlumbBob.png"];

plumbBobView = [[UIImageView alloc] initWithImage:image];

// 移动锚点到底部节拍器区域中间

plumbBobView.layer.anchorPoint = CGPointMake(0.5, 0.0);

//确定锚点后设置帧，显示的将会在正确启动位置.

plumbBobView.frame = CGRectMake(self.view.frame.size.width/2 - 20, 0, 40, 450);

[self.view addSubview:plumbBobView];

[plumbBobView release];

// 配置和启动加速度检波器

[[UIAccelerometer sharedAccelerometer] setUpdateInterval:(1.0 / kAccelerometerFrequency)];

[[UIAccelerometer sharedAccelerometer] setDelegate:self];

}

// Override to allow orientations other than the default portrait orientation.

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {

    // Return YES for supported orientations

    return (interfaceOrientation == UIInterfaceOrientationPortrait);

}

- (void)didReceiveMemoryWarning {

    [super didReceiveMemoryWarning]; // Releases the view if it doesn't have a superview

    // Release anything that's not essential, such as cached data

}

```

```

- (void)dealloc {

    [plumbBobView release];

    [super dealloc];
}


#pragma mark -

#pragma mark === Accelerometer delegate ===

#pragma mark -

// UIAccelerometerDelegate method, 当设备加速时呼出.

- (void)accelerometer:(UIAccelerometer *)accelerometer didAccelerate:(UIAcceleration *)acceleration {

    [self rotatePlumbStringToDegree:-acceleration.x* 30];

}


#pragma mark -

#pragma mark === Swing the plumb and string ===

#pragma mark -

- (void)rotatePlumbStringToDegree:(CGFloat)positionInDegrees {

    [plumbBobView.layer removeAllAnimations];

    CATransform3D rotationTransform = CATransform3DIdentity;

    rotationTransform = CATransform3DRotate(rotationTransform, DegreesToRadians(positionInDegrees), 0.0,
0.0, 1.0);

    plumbBobView.layer.transform = rotationTransform;

}

@end

```

```
CGFloat DegreesToRadians(CGFloat degrees) {return degrees * M_PI / 180;};
```

```
CGFloat RadiansToDegrees(CGFloat radians) {return radians * 180/M_PI;};
```

最后在 xCode 选择 **Build->Build and Go; Save All.**

下载今天教程文件: [PlumbBob.zip](#)

三十一天手把手 iPhone app 基础, 来源于 [www.appsamuck.com](http://www.appsamuck.com), 大家有空可以去看看原文;

下面文章本人根据 appsamuck 这个网站加以翻译,修改及测试, 希望大家多多交流。一切版权归

<http://blog.sina.com.cn/iphonesdk> 所有。

## 实例 iPhone SDK 编程入门教程 - 第十二天

(2009-06-26 12:14:59)

转载

标签: 分类: [实例 iPhoneSDK 入门](#)

[iphone](#)

[sdk](#)

[编程](#)

[iphone 工具](#)

[iphone 软件编程](#)

[分歧终端机](#)

[it](#)

### DAY TWELVE – Decide



今天来建立一个 iPhone app 软件，用你的 iPhone 分歧解决器。

纲要：

- 在程序显示前运行代码 -
- 关于 iPhone 的“Utility Application” 运用 -
- Rand, Label, Button 代码运用 -

首先运行以安装好的 xCode

选择: File->New Project.

从 "New Project" 窗口

选择 : iPhone OS ->Applications-> Utility Application

命名 : 我这里命名为 “UDecide”

**(1) 在 xCode 打开 MainView.h 文件,加入下面代码**

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

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

```
@interface MainView : UIView {
```

```
    IBOutlet UILabel *result;
```

```
}
```

```
- (IBAction)abcd;
```

```
- (IBAction)agreeDisagree;
```

```
- (IBAction)headsTails;
```

```
- (IBAction)leftCenterRight;
```

```
- (IBAction)lottery;
```

```
- (IBAction)oneToHundred;
```

```
- (IBAction)positiveNegative;  
  
- (IBAction)russianRoulette;  
  
- (IBAction>trueFalse;  
  
- (IBAction)yesNo;  
  
@end
```

(2) 在 xCode 打开 **MainView.m** 文件,加入下面代码

```
#import "MainView.h"  
  
@implementation MainView  
  
- (IBAction)abcd {  
  
    int rNumber = rand() % 4; //计算几率: 四分之一  
  
    switch (rNumber) {  
  
        case 0:  
  
            result.text = @"A";  
  
            break;  
  
        case 1:  
  
            result.text = @"B";  
  
            break;  
  
        case 2:  
  
            result.text = @"C";  
  
            break;  
  
        case 3:  
  
            result.text = @"D";  
  
            break;  
  
        default:  
  
            break;  
  
    }  
}
```

```
}
```

```
}
```

```
- (IBAction)agreeDisagree {
```

```
    int rNumber = rand() % 2;
```

```
    switch (rNumber) {
```

```
        case 0:
```

```
            result.text = @"Agree";
```

```
            break;
```

```
        case 1:
```

```
            result.text = @"Disagree";
```

```
            break;
```

```
        default:
```

```
            break;
```

```
    }
```

```
}
```

```
- (IBAction)headsTails {
```

```
    int rNumber = rand() % 4;
```

```
    switch (rNumber) {
```

```
        case 0:
```

```
            result.text = @"Heads";
```

```
            break;
```

```
        case 1:
```

```
            result.text = @"Tails";
```

```
            break;
```

```
        default:
```

```
            break;
```

```
}
```

```
}
```

```
- (IBAction)leftCenterRight {
```

```
    int rNumber = rand() % 3;
```

```
    switch (rNumber) {
```

```
    case 0:
```

```
        result.text = @"Left";
```

```
        break;
```

```
    case 1:
```

```
        result.text = @"Center";
```

```
        break;
```

```
    case 2:
```

```
        result.text = @"Right";
```

```
        break;
```

```
    default:
```

```
        break;
```

```
    }
```

```
}
```

```
- (IBAction)lottery {
```

```
    int rNumber = rand() % 3;
```

```
    switch (rNumber) {
```

```
    case 0:
```

```
        result.text = @"Buy";
```

```
        break;
```

```
    case 1:
```

```
        result.text = @"Sell";
```

```

        break;

    case 2:

        result.text = @"Hold";

        break;

    default:

        break;

    }

}

- (IBAction)oneToHundred {

    int rNumber = rand() % 100;

    result.text = [[NSString alloc] initWithFormat:@"%d", rNumber];

}

- (IBAction)positiveNegative {

    int rNumber = rand() % 2;

    switch (rNumber) {

        case 0:

            result.text = @"Positive";

            break;

        case 1:

            result.text = @"Negative";

            break;

        default:

            break;

    }

}

- (IBAction)russianRoulette {

    int rNumber = rand() % 6;

```



```
switch (rNumber) {  
  
    case 0:  
  
        result.text = @"BANG!!!";  
  
        break;  
  
    default:  
  
        result.text = @"Click...";  
  
        break;  
  
}  
}
```

```
- (IBAction>trueFalse {  
  
    int rNumber = rand() % 2;
```

```
switch (rNumber) {  
  
    case 0:  
  
        result.text = @"True";  
  
        break;  
  
    case 1:  
  
        result.text = @"False";  
  
        break;  
  
    default:  
  
        break;  
  
}  
}
```

```
- (IBAction)yesNo {  
  
    int rNumber = rand() % 2;
```

```
switch (rNumber) {
```

```
case 0:

    result.text = @"Yes";

    break;

case 1:

    result.text = @"No";

    break;

default:

    break;

}

}

@end
```

### (3) UIView 界面设置

双击文件: "main.xib" ;

然后 "Interface Builder" 会自动打开, 在这里我们可以编辑改变界面

### (4) 加入 Label 在 Attributes 下, 调整合适的大小

选择: Tools -> Connection Inspector

移动鼠标在"Referencing Outlets" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "result"; 选上它。

### (5) 加入 Button ; 显示:Yes/No

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "yesNo"; 选上它。

选择: Tools -> Attributes Inspector

在 Title 里填上 Yes/No

#### **(6) 加入 Button ; 显示:A/B/C/D**

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "abcd"; 选上它。

选择: Tools -> Attributes Inspector

在 Title 里填上 A/B/C/D

#### **(7) 加入 Button ; 显示:1-100**

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "oneTohundred"; 选上它。

选择: Tools -> Attributes Inspector

在 Title 里填上 1-100

#### **(8) 加入 Button ; 显示:左/中/右**

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "leftCenterRight"; 选上它。

选择: Tools -> Attributes Inspector

在 Title 里填上 左/中/右

#### **(9) 加入 Button ; 显示:买/卖/保留**

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "lottery"; 选上它。

选择: Tools -> Attributes Inspector

在 Title 里填上 买/卖/保留

#### **(10) 加入 Button ; 显示:同意/反对**

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "agreeDisagree"; 选上它。

选择: Tools -> Attributes Inspector

在 Title 里填上 同意/反对

#### **(10) 加入 Button ; 显示:同意/反对**

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "agreeDisagree"; 选上它。

选择: Tools -> Attributes Inspector

在 Title 里填上 同意/反对

#### **(11) 加入 Button ; 显示: 公/字**

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "headsTails"; 选上它。

选择: Tools -> Attributes Inspector

在 Title 里填上 公/字

#### **(12) 加入 Button ; 显示: 俄罗斯轮盘**

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "russianRoulette"; 选上它。

选择: Tools -> Attributes Inspector

在 Title 里填上 俄罗斯轮盘

### **(13) 加入 Button ; 显示: 正/负**

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "possitiveNegative"; 选上它。

选择: Tools -> Attributes Inspector

在 Title 里填上 正/负

### **(14) 加入 Button ; 显示: 对/错**

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Button 到 Main View

在主视窗口或文件窗口; 点击 Button

选择: Tools -> Connection Inspector

移动鼠标在"Touch Up Inside" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"Main View";

放开鼠标选择键出现 "true/flase"; 选上它。

选择: Tools -> Attributes Inspector

在 Title 里填上 对/错

最后在 xCode 选择 Build->Build and Go; Save All.

下载今天教程文件: [Decide.zip](#)

三十一天手把手 iPhone app 基础, 来源于 [www.appsamuck.com](http://www.appsamuck.com), 大家有空可以去看看原文;

下面文章本人根据 appsamuck 这个网站加以翻译,修改及测试, 希望大家多多交流。一切版权归

<http://blog.sina.com.cn/iphonesdk> 所有。

## 实例 iPhone app 编程入门教程 - 第十三天

(2009-07-29 19:14:17)

转载

标签: 分类: [实例 iPhoneSDK 入门](#)

[iphone](#)

[sdk](#)

[编程](#)

[iphone 工具](#)

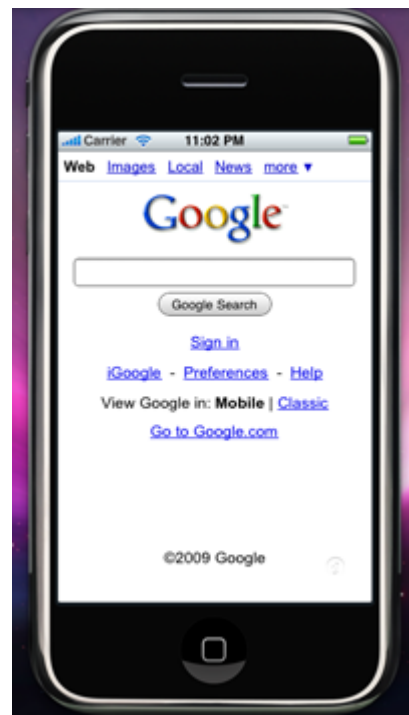
[iphone 软件编程](#)

[打开网页](#)

[网页编程](#)

[it](#)

**DAY thirteen – My Google**



今天来建立一个 iPhone app 软件，用你的 iPhone 直接打开谷歌网页。

纲要：

- 在程序显示前运行代码 -
- 关于 iPhone 的“UIWebView”运用 -

首先运行以安装好的 xCode

选择: File->New Project.

从 "New Project" 窗口

选择 : iPhone OS ->Applications-> Utility Application

命名 : 我这里命名为 “Mygoogle”

(1) 在 xCode 打开 MainViewController.h 文件,加入下面代码

```
@interface MainViewController : UIViewController {  
  
    IBOutlet UIWebView *webView;  
  
}  
  
@end
```

(2) 在 xCode 打开 MainViewController.m 文件,加入下面代码

```
#import "MainViewController.h"  
  
#import "MainView.h"  
  
@implementation MainViewController
```



```
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {  
  
    if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {  
  
        // Custom initialization  
  
    }  
  
    return self;  
  
}
```

// Implement viewDidLoad to do additional setup after loading the view, typically from a nib.

```
- (void)viewDidLoad {  
  
    // set userInteractionEnabled to true so the user can click links  
  
    webView.userInteractionEnabled = true;  
  
    // navigate to google  
  
    [webView loadRequest:[NSURLRequest alloc] initWithURL:[NSURL alloc]  
initWithString:@"http://www.google.com"]];  
  
}
```

// Override to allow orientations other than the default portrait orientation.

```
- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {  
  
    // Return YES for supported orientations  
  
    return (interfaceOrientation == UIInterfaceOrientationPortrait);  
  
}
```

```
- (void)didReceiveMemoryWarning {  
  
    [super didReceiveMemoryWarning]; // Releases the view if it doesn't have a superview
```

```
// Release anything that's not essential, such as cached data  
}
```

```
- (void)dealloc {  
    [super dealloc];  
}
```

**(3) 在 xCode 打开 FlipsideViewController.h 文件,加入下面代码**

```
#import <UIKit/UIKit.h>  
  
@interface FlipsideViewController : UIViewController {  
    IBOutlet UIWebView *webView;  
}  
  
@end
```

**(4) 在 xCode 打开 FlipsideViewController.m 文件,加入下面代码**

```
#import "FlipsideViewController.h"  
  
@implementation FlipsideViewController  
  
- (void)viewDidLoad {  
    webView.userInteractionEnabled = true;  
    // load our about page from the resource bundle
```

```
NSString *filePath = [[NSBundle mainBundle] pathForResource:@"about" ofType:@"html"];

NSData* htmlData = [NSData dataWithContentsOfFile:filePath];

[webView loadData:htmlData MIMEType:@"text/html" textEncodingName:@"UTF-8" baseURL:[NSURL
URLWithString:@"http://www.mywebsite.com/about.html"]];

}
```

```
// Override to allow orientations other than the default portrait orientation.

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {

    // Return YES for supported orientations

    return (interfaceOrientation == UIInterfaceOrientationPortrait);

}
```

```
- (void)didReceiveMemoryWarning {

    [super didReceiveMemoryWarning]; // Releases the view if it doesn't have a superview

    // Release anything that's not essential, such as cached data

}
```

```
- (void)dealloc {

    [super dealloc];

}
```

@end

(5) 导入下面 html 文件

下载下面 html，放入 Mygoogle 文件夹内并命名为下面名称 about.html

文件: [about.html](#)

在 xCode 下右键点击 Mygoogle->Add->Existing Files; 在 Mygoogle 文件夹内，选择下载好的图片，按 Add

#### **(6) MainView 界面设置**

双击文件: "mainView.xib" ;

然后 "Interface Builder" 会自动打开，在这里我们可以编辑改变界面

#### **(7) 加入 UIWebView ; 显示:www.google.com**

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Webview 到 Main View

在主视窗口或文件窗口; 点击 UIWebView

选择: Tools -> Connection Inspector

移动鼠标在 "New Referencing Outlets" 后面圆圈上; 圆圈变为(+); 拖向直线连接到 "File's Owner";

放开鼠标选择键出现 "webView"; 选上它。

#### **(8) FlipsideView 界面设置**

双击文件: "mainView.xib" ;

然后 "Interface Builder" 会自动打开，在这里我们可以编辑改变界面

#### **(9) 加入 UIWebView ; 显示:about.html**

选择: Tools -> Library ; 从 Library 显示菜单中拖拉一个 Webview 到 Main View

在主视窗口或文件窗口; 点击 UIWebView

选择: Tools -> Connection Inspector

移动鼠标在"New Referencing Outlets" 后面圆圈上; 圆圈变为(+); 拖向直线连接到"File's Owner";

放开鼠标选择键出现 "webView"; 选上它。

**最后在 xCode 选择 Build->Build and Go; Save All.**

下载今天教程文件: [Mygoogle.zip](#)

三十一天手把手 iPhone app 基础, 来源于 [www.appsamuck.com](http://www.appsamuck.com), 大家有空可以去看看原文;

下面文章本人根据 appsamuck 这个网站加以翻译,修改及测试, 希望大家多多交流。一切版权归

<http://blog.sina.com.cn/iphonesdk> 所有。