

原文及示例代码来自raywenderlich store中的iOS Apprentice 系列3教程，经过翻译和改编。

版权归原作者所有，本系列教程仅供学习参考使用，感兴趣的朋友建议购买原英文教程教程(The iOS Apprentice Second Edition: Learn iPhone and iPad Programming via Tutorials!)

购买链接：

<http://www.raywenderlich.com/store>

最近玩炉石有点迷，顺便看了下比较火的几个无节操（非贬义）游戏解说视频，比如夏一可，星际少帮主，毕游侠神马的，推荐大家假期可以好好看看，玩玩。

神马？你竟然问我天梯排名多少？这不是拍脸至死的节奏吗？！那啥，我们还是开始学习本课的内容吧，看我怎么用各种SY（术语）虐死你。



首先还是回顾下上期节目内容。恩，怎么觉得哪里有点不对。

在上一期节目中，我们学习了如何利用Core Location框架来获取一个CLLocation对象，而且了解了如何处理不同的错误信息。接下来该咋办？

好吧，如果你够细心，会看到Core Location一直在努力寻找新的位置信息对象，一直在努力，一直在努力.....即便最终的坐标已经不再发生任何变化了，还是一直在努力，一直在努力...这是因为用户可能会时不时走两步，然后他们的位置信息也的确发生了变化了。不过这里我们不打算搞一个导航应用，因此当我们觉得位置信息已经足够准确了以后，就需要通知location manager停止发送位置信息更新。

这一点很重要，因为获取位置信息更新很耗电，因为在这个过程中需要保持GPS/Wi-Fi/蜂窝网络信号的开启。而我们这款应用并不需要随时请求GPS信息，因此当位置信息差不多了时候就该消停一会儿了。

问题在于，怎么来判断这个位置信息已经足够精确了呢？

这个的确要动一下脑子。这里给出的解决办法是：如果最后获得的几个坐标信息没有明显的变化，那么就说明位置信息基本上就差不多准确了，此时就可以让定位的硬件元件休息一下省省电了。

在CLLocationViewController.m中更改didUpdateLocations方法如下：

```
-(void)locationManager:(CLLocationManager *)manager didUpdateLocations:(NSArray *)locations{  
    CLLocation *newLocation = [locations lastObject];  
    NSLog(@"已更新坐标，当前位置：%@",newLocation);  
  
    if([newLocation.timestamp timeIntervalSinceNow] < -5.0){
```

```

        return;
    }

    if(newLocation.horizontalAccuracy <0){
        return;
    }

    if(_location == nil || _location.horizontalAccuracy > newLocation.horizontalAccuracy){

        _lastLocationError = nil;
        _location = newLocation;
        [self updateLabels];

        if(newLocation.horizontalAccuracy <= _locationManager.desiredAccuracy){
            NSLog(@"***目标诺森德! 成功完成定位");
            [self stopLocationManager];
        }

    }
}

```

上面这段代码有点长，我们还是一步步来看吧：

```

if([newLocation.timestamp timeIntervalSinceNow] < -5.0){
    return;
}

```

如果获取最近一个位置信息对象的时间太长（这里设置的阈值是5秒），那么它就是一个所谓的 **cached**（缓存）结果。**location manager**并非一直向你提供最新的位置信息，而是在最开始的时候提供最近的位置信息，当然，它这样做的原因是假定你从上次定位开始并没有过多移动。（额，如果你是超人、火箭飞人之类的生物，这个假设显然是毫无意义的）。对于这种旧的缓存位置信息，显然对我们是没用的。注意这里的判断是在当前时刻前的5秒前。

接下来看这个语句：

```

    if(newLocation.horizontalAccuracy <0){
        return;
    }

```

这里我们使用位置信息对象的 **horizontalAccuracy** 属性来判断新的结果是否比之前的结果更为准确。如果碰到这个属性变量的值比0小，就说明这些测量结果是无效的，我们应该将其忽略。

继续看下面的这行语句：

```

    if(_location == nil || _location.horizontalAccuracy > newLocation.horizontalAccuracy){

```

这个语句的作用是判断新的读数信息是否比之前的更有用。一般来说，**Core Location**会先提供一个不太精确的数据信息，然后随着时间推移不断将其优化。不过我们无法确保任何时候新的读数信息都比之前的数据要准确。

需要注意的是，更大的 **accuracy** 数值意味着更低的精度-想想看，精度为100米肯定比精度10米要低。这就是为什么我们要检查前一个读数 **\_location.horizontalAccuracy** 是否比新的读数 **newLocation.horizontalAccuracy** 要大。

当然，我们还检查了 **\_location == nil**。**\_location**是我们之前所定义的一个实例变量，它会保存我们在前一次调用 **didUpdateLocations** 方法中所获取的 **CLLocation** 对象。如果 **\_location** 是 **nil**，说明这是

第一次获取位置信息。这里的||操作符（又叫逻辑或）会测试条件语句中是否有一个是真。如果是，那么就继续执行下面的语句：

```
_lastLocationError = nil;  
_location = newLocation;  
[self updateLabels];
```

上面的三行语句之前见过，其作用就是清除之前的任何错误，然后将新的位置信息对象保存到\_location变量中。

接下来还会做一次判断：

```
if(newLocation.horizontalAccuracy <= _locationManager.desiredAccuracy){  
    NSLog(@"****目标诺森德！ 成功完成定位");  
    [self stopLocationManager];  
}
```

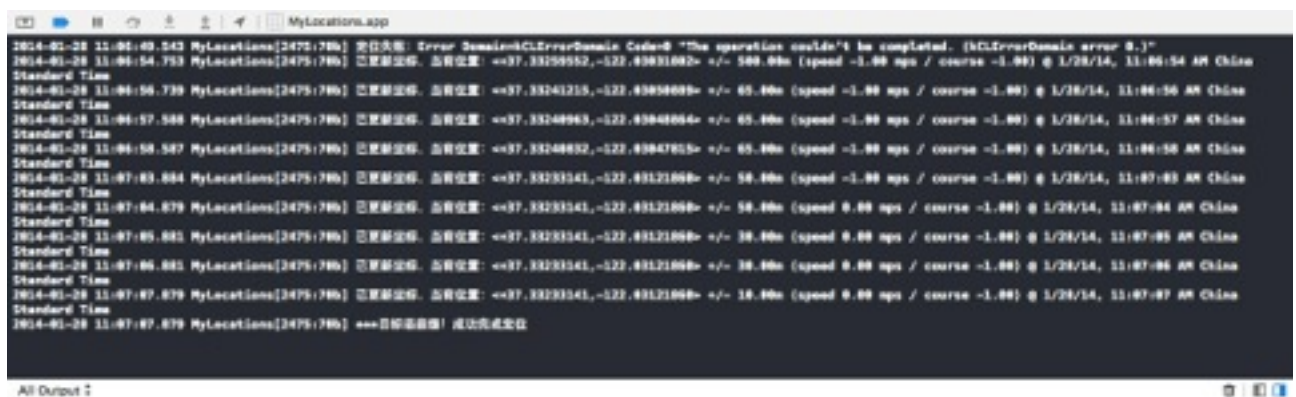
如果新位置信息对象的精度大于或等于我们期望的精度，然后我们就输出一条信息宣布任务搞定，同时停止location manager。

当我们在startLocationManager中启动location manager的时候，将期望精度设置为10米(kCLLocationAccuracyNearestTenMeters)，对我们这个应用来说已经足够了。

编译运行应用。

首先将Simulator的位置设置为None（不知道怎么设置？回头看上一课内容！谁让你BS我天梯排名的？），然后触碰Get My Location按钮。

此时界面上会显示“Searching...”。然后将地址切换成Apple。过一会儿之后，界面上就会显示定位到的GPS坐标信息。如果我们仔细看debug调试区的信息，就会看到类似下面的：



额，很不错的吗。不过目标诺森德是啥意思？

你不玩魔兽的吗？你不玩炉石的吗？那我不跟你废话了，继续下面的学习吧。

作为一个开发者，我们很容易就知道location manager停止工作了，因为可以在debug区里面看到相关信息。不过用户咋办？当第一个位置信息出现的时候界面上就出现了”标记位置“这个按钮。但此时的位置信息可能不是那么准确。所以最好当应用已经获取了最准确的位置后再向用户显示这个按钮。

好吧，为了让这个过程对用户显得更友好，我们需要在搜寻位置信息的时候把Get My Location按钮更改为Stop，而当搜索完成后重新设置为Get My Location。这样用户就会知道究竟发生了什么。当然，在本教程后面的课里面，我们还会使用一个动画一样的活动标志（苹果标志性的菊花？你太邪恶了，这里



不提供肥皂）让这个过程更加可视化。

为了更改按钮的状态，首先需要在代码中添加一个configureGetButton方法。

在CurrentLocationViewController.m中添加一个新的方法如下：

```
-(void)configureGetButton{  
    if(!_updatingLocation){  
        [self.getButton setTitle:@"停停停停停" forState:UIControlStateNormal];  
    }else{  
        [self.getButton setTitle:@"获取当前所在位置" forState:UIControlStateNormal];  
    }  
}
```

上面方法中的代码还是比较容易理解的：

如果应用当前在更新位置信息，那么将按钮的显示文字设置为Stop，否则就还是Get My Location。请原谅我偷懒了，烦着这些文字是中文是英文都随你。

接下来当然就是在合适的地方调用configureGetButton方法了：

```
-(void)viewDidLoad  
{  
    [super viewDidLoad];  
    [self updateLabels];  
    [self configureGetButton];  
}  
  
-(IBAction)getLocation:(id)sender{  
    [self startLocationManager];  
    [self updateLabels];  
    [self configureGetButton];  
}  
  
-(void)locationManager:(CLLocationManager *)manager didFailWithError:(NSError *)error{  
    NSLog(@"定位失败：%@",error);  
    if(error.code ==kCLErrorLocationUnknown){  
        return;  
    }  
    [self stopLocationManager];  
    _lastLocationError = error;  
    [self updateLabels];  
    [self configureGetButton];  
}  
  
-(void)locationManager:(CLLocationManager *)manager didUpdateLocations:(NSArray *)locations{  
    CLLocation *newLocation = [locations lastObject];  
    NSLog(@"已更新坐标，当前位置：%@",newLocation);
```



```

if([newLocation.timestamp timeIntervalSinceNow] < -5.0){
    return;
}

if(newLocation.horizontalAccuracy < 0){
    return;
}

if(_location == nil || _location.horizontalAccuracy > newLocation.horizontalAccuracy){

    _lastLocationError = nil;
    _location = newLocation;
    [self updateLabels];

    if(newLocation.horizontalAccuracy <= _locationManager.desiredAccuracy){
        NSLog(@"***目标诺森德! 成功完成定位");
        [self stopLocationManager];
        [self configureGetButton];
    }
}
}
}

```

基本上只要我们调用updateLabels方法的地方都需要调用configureGetButton方法。不过在didUpdateLocations中我们只会在目标完成的时候才会调用它~

好吧，现在可以再次编译运行应用了，让我们来进行同样的测试吧（先把Simulator的位置设置成none,再改成apple）。当我们触碰底部那个“获取位置”按钮的时候它会变身为Stop。当定位完成的时候它再变身回来，实在是七十二般变化样样精通啊。哦，说到七十二般变化，最近有一部名为“大闹天宫”的电影貌似还不错啊，好像招行信用卡的积分还可以免费换票啊。等等，怎么有点打广告的嫌疑？就我这节目收视率，还打个毛的广告啊？就是为了图大家一乐，正所谓一笑解千愁，无码定乾坤~



看到上面这个无节操的图片，你是不是以为这一课已经结束了？

那可不行，还没有虐够你呢！首先看看我们截取的simulator截图吧，当然结果肯定是我们预料的那样，除非你犯了某些低级错误。



这还没完。

从用户的角度来说，当他们看到”太上老君急急如律令-停停停停停“的时候，希望可以通过触碰这个按钮来扮演神的角色停止这一切。可惜的是目前这么做什么也不会改变。因此，我们需要更改getLocation这个动作方法：

```
-(IBAction)getLocation:(id)sender{
    if(_updatingLocation){
        [self stopLocationManager];
    }else{
        _location = nil;
        _lastLocationError = nil;

        [self startLocationManager];
    }

    [self updateLabels];
    [self configureGetButton];
}
```

这里我们再次利用\_updatingLocation这个状态标记变量来判断应用所处的状态。

当用户触碰这个按钮时，如果应用正在努力获取位置信息，就停止location manager。当然同时我们要注意的，在尝试获取一个新的位置前，还需要清除旧的位置和错误对象。

好了，再次编译运行应用。

现在触碰Stop按钮会终止位置更新。当然debug区也看不到新的位置信息出现了。

恭喜你修得老君派人阶初级定身法术的第一层，不过貌似离雷劫还早着呢，切莫害怕

好了。这一课的内容算是到此结束。

不过我们还是要提供两个消息。

第一个是关于苹果的。

我五星神厨船长今天公布了2014年第一季度财报，收入576亿美元（注意是美刀，不是天币，更不是bitcoins），纯利润131亿美元。实在是当今一大土豪啊。

苹果Q1给开发者开了20亿美元的支票，这其中有没有你的一份呢？

当然，这个数字虽然惊人，但iphone5100万部的销量还是低于分析师的预计，因此苹果股票大跌5%。

第二个吗，还是关于苹果的。

有谣言称苹果2014年将推出新的apple tv，当然这不是重点。重点的是新的apple tv将配有体感装置，还有专为apple tv提供内容的appstore。

当然，一般的土豪们更关心iPhone6。不过作为开发者，还是要关注下这个具有强大潜力的新生态市场-智能电视。



好了，最后是福利时间。  
快过年了，在家看美女很危险的说，所以还是送红包吧。

注意，不是给你红包图哦，是真的红包哦！！

从今天起（2014年1月28日，农历腊月二十八）到2014年1月31日（正月初一），凡是加我微信好友而且给我送微信红包的朋友都会得到我回赠的红包，哈哈~开玩笑，只要加了我微信，在微信上拜个年就都有，虽然很穷，但送个几元小红包还是比较喜庆的~

我的微信红包已经准备好了，你的呢？

