iPad横竖屏代码适配

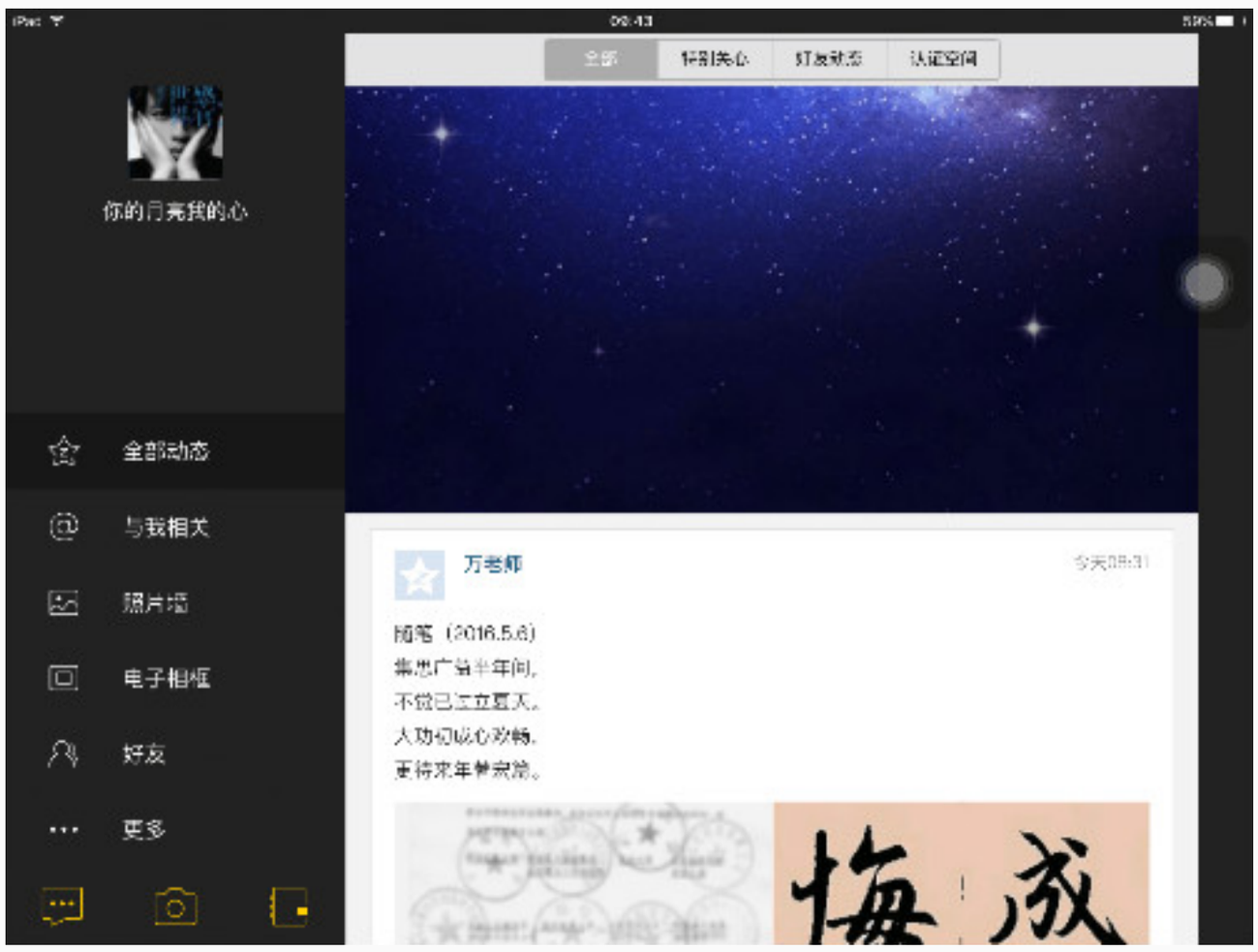
2016-05-27 08:29 编辑： cocopeng 分类： iOS开发 来源： Lemtter 0 37113711

代码屏幕适配 iPad

招聘信息：C++开发工程师（cocos2d）

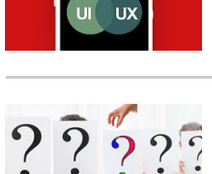
你可能非常了解用不同的方式去适配不同尺寸的iPhone屏幕,在适配iPhone屏幕时你需要考虑的只是屏幕大小变化带来的UI元素间隔的变化,但是在iPad上主要针对的是横竖屏下完全不同的UI元素的布局,在这种情况下要考虑的就不仅仅是元素之间间隔这种问题了,除了要确保UI元素在这两种模式下的正确显示还要兼顾屏幕旋转的过渡动画.下图是QQZone For iPad 在横竖屏下的布局,可以看到横竖屏下菜单栏的布局方式差别很大.

QQZone for iPad GitHub地址



QQZone for iPad 竖屏

热门资讯

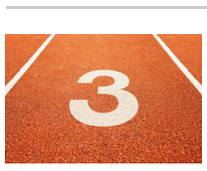

知道这20个正则表达式，能让你少写1,000 点击量 10556


苹果最刁钻的33个面试题，有种就来挑战 点击量 8052

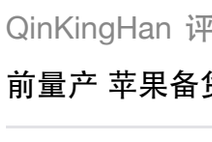

移动端数据库新王者：realm 点击量 6728

针对苹果最新审核要求为应用兼容IPv6 点击量 5747

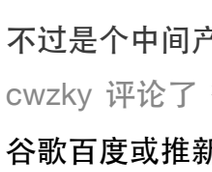
UIApplication的详细介绍 点击量 5439

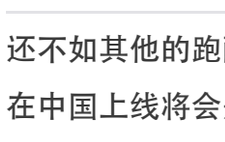

程序员这样优化简历一投制胜 点击量 5153


Runtime全方位装逼指南 点击量 5102



2016年5月TIOBE编程语言排行榜：Ruby创 点击量 4605


你真的了解load方法么? 点击量 4470



iOS 三种录制视频方式 点击量 4048

综合评论

ReSwift就是这个东东吧
hbj 评论了 Redux-OC

小伙子 我看好你！
hnny09 评论了 2017，我的iOS进阶规划...

好丑
QinKingHan 评论了 iPhone 8或六月提前量产 苹果备货900万...

如果Facebook没有国界，微信最终只不过是个中间产物。
cwzky 评论了 微信小程序出师不利，谷歌百度或推新Web应用标准抄...

还不如其他的跑酷游戏，这款游戏如果在中国上线将会是一个失败品

QQZone for iPad 横屏

屏幕适配的N种方法

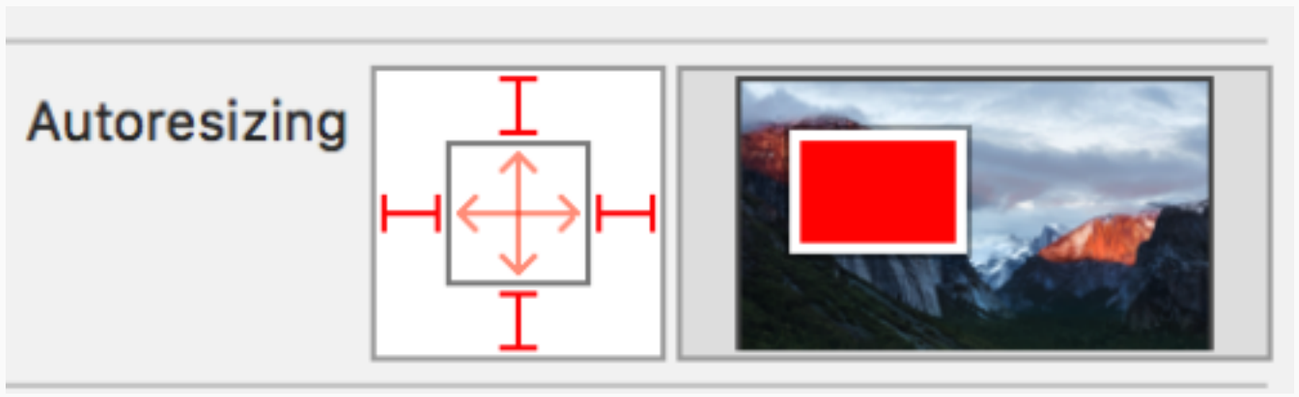
无论iPad还是iPhone适配不同屏幕(尺寸,方向)的方式都跑不出以下几种,以下会一一对不同方式做一下简单的回顾.

Autoresizing

Autoresizing可以说是Autolayout始祖,Autoresizing的是一项比较有历史的技术了,其在iOS2的时代就推出了.当设置UIView实例对象的autoresizesSubviews属性为true(默认值为true),那么其子view会根据自已的autoresizingMask属性值自动调整与superview的位置和大小关系.autoresizingMask有六种可组合的使用的值,默认值是.None.这六种有效枚举值的意思如下:

- FlexibleLeftMargin 按比例跟随父控件变化的左间距
- FlexibleWidth 按比例跟随交控件变化的宽度
- FlexibleRightMargin 按比例跟随父控件变化的右间距
- FlexibleTopMargin 按比例跟随父控件变化的顶部间距
- FlexibleHeight 按比例跟随控件变化的高度
- FlexibleBottomMargin 按比例跟随父控件变化的底部间距

另外在xib,storyboard取消Autolayout时(Autoresizing与Autolayout相互冲突)可以在Size inspector可以更加直观地按需求进行组和使用.Autoresizing技术在一定应用场景下可以勉强使用但应对更为精细的布局就无能为力了,你可以在使用Autoresizing同时重写layoutSubviews方法去做更为精细的布局,尽管如此但还是不推荐这么做,因为同时得写layoutSubviews和使用Autoresizing去布局会让你的布局逻辑变得不清晰,这将给后期的维护带来麻烦.



Autoresizing in storyboard

支付宝支付 只能调用支付宝钱包支付 不能跳网页支付

女生零基础学ui好学吗?

【新手求助】ios程序如何内嵌h5页面?

这是一个关于内购打款的问题

所有虚函数都不承认了

免费的轻快pdf阅读器适用于Windows系统

需求贴

高德地图自定义标记 不响应点击事件

(求助) 招商银行visa全币卡 无法购买开发者账号，提示你的支付授权失败

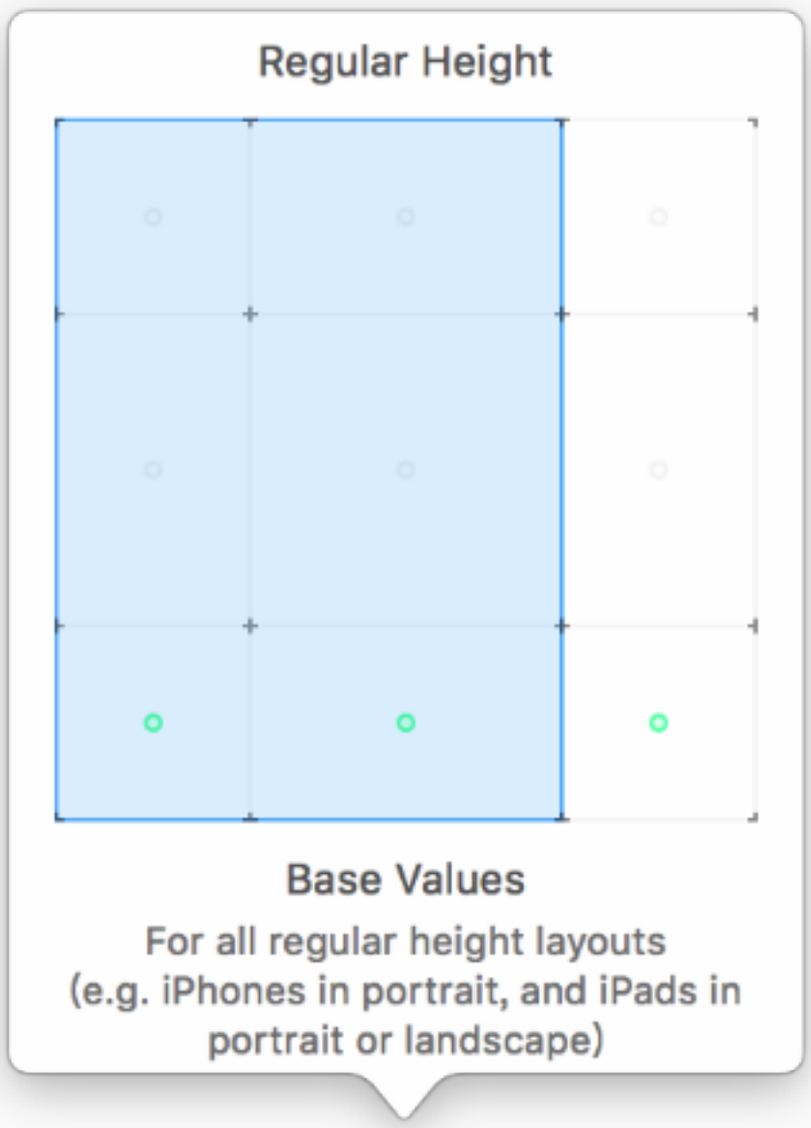
Autolayout

Autolayout是iOS6时代引入的技术,专门用来处理不同屏幕尺寸下的UI布局.从Xcode6开始Autolayout配合xib,storyboard极大的提高屏幕的适配工作效率.在一定程度上甚至可以完全摆脱设置frame布局的方式.由于storyboard,xib在多人合作开发冲突不断的尴尬境地,在实际的开发中多使用第三方框架用代码进行Autolayout布局.这样既避免了解决冲突麻烦又享受到了Autolayout带来的宏利.比较受欢迎的Autolayout每三方框架有Masonry还有GSD_iOS大神的SDAutoLayout

尽管Autolayout有很多好处但还是很多代码党不愿使用,究其原因还是约束.约束的问题大至可以分约束冲突和约束不满足两大类,当在storyboard中对一个复杂的界面进行Autolayout约束,一但出现问题将很难排查,用代码行约束往往程序运行起来才能确认约束是否满足条件,同样排查起来也不是那么方便.关于Autolayout这不再占用过多的篇幅,网上有相当多的资料可供参考.

SizeClass

SizeClass是要配合Autolayout使用的,SizeClass实际上是对屏幕尺寸的抽象,把屏幕宽高分成Compact:紧凑、Regular:宽松、Any:任意三种类型这样就可以组合出九种不同的屏幕类型.在storyboard,xib编辑界面下最下方可以选择某一约束在只在某一类屏幕下生效.这样可以在不同屏幕下得到不同的UI布局效果.关于SizeClass的使用可以参考raywenderlich系例文章.



SizeClass

代码计算坐标

在所有的布局方法中这种可能是最费体力的一项,因为所有的UI元素都需要一个一个明确的计算或者指定出来.尽管如此正因为每个元素的frame是手动计算因此灵活性也非常大你可以随心所欲的计算每个控件的frame,出现问题时也非常好排查.如果需要动态的改变view的frame就需要重写父控件的layoutSubviews方法,在重写的layoutSubviews明确计算出frame. 如果view是固定的则只需要在添加到父控件时指定view的frame.一般常见的代码布局形式如下:

```
override func layoutSubviews() {
super.layoutSubviews()
let x: CGFloat = 0
let y: CGFloat = frame.height * 0.7 // 根据父控件高度按比例确定y座标
let w: CGFloat = frame.width
let h: CGFloat = frame.height - y  // 根据父控件高度,子控件按比例调整高度
```

```
7  subView.frame = CGRectMake(x, y, w, h)
8  }
```

上面是常见的根据父控件动态调整子控件frame的形式,真实开发中可能还需要考虑横竖屏下动态的布局(下面将提到的),以上基本形式可以根据需求进行扩展.当然你也可以不用重写layoutSubviews方法而在需要改变frame的时机显式直接改变frame,但在这种方式并不符合苹果的逻辑.在view的层次结构中某一view肯定是有superview的,而子view是否变化,以及什么时候变化应是由superview来决定的,在一个多层次结构的view视图中如果显示的设定子view的frame那你不得不根据view的层次结构一级一级的设置子view的frame.superview和subView之前会出现较强的关联性.理想情况下一个superview应该只关注自身的subView的布局,无论这个superview的frame或层次结构怎么变化其subviews并不需要知道.因此我觉得比较好的做法是有所关于subviews的frame的设定都应该重写layoutSubviews,在layoutSubviews中去做.这样做的另一个好处是屏幕旋转时你并不需要显示的去写UIView动画.

layoutsubviews的调用时机

用代码在layoutSubviews中布局你必需要知道系统会在哪些时机去调用layoutsubviews函数.关于这个问题的结论可以参看stackoverflow的讨论.关于这个问题我个人比较赞同第二个回答者对第一个回答的纠正.基本上layoutsubviews会在以下几种情况下调用:

- 当view的bounds发生改变时
- 当view的直接subView的bounds发生改变时
- 当subView添加或移除时
- 调用setNeedsLayout方法会在下一个显示周期主动调用layoutsubviews

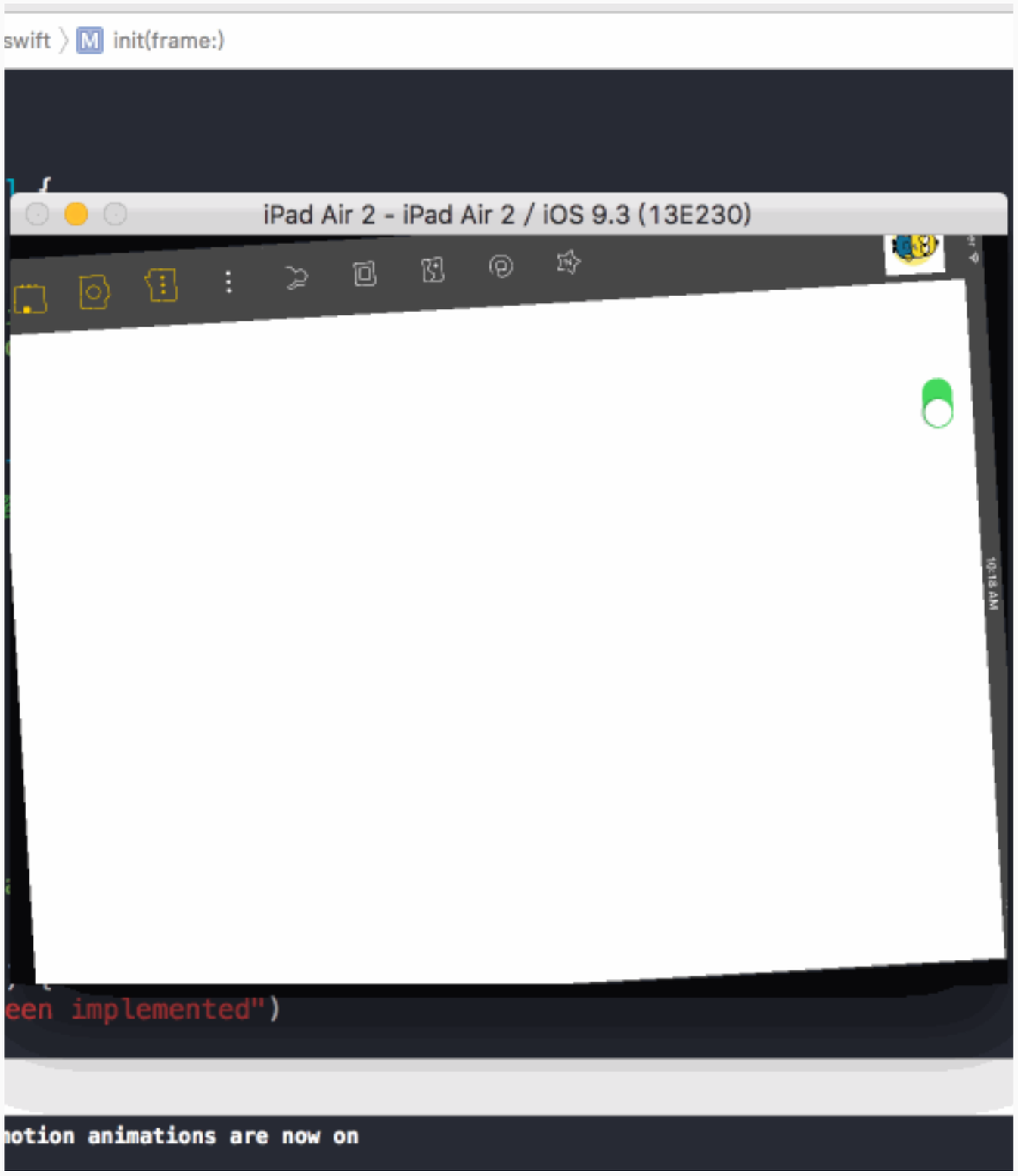
如何获取当前屏幕方向

关于获取当前屏幕方向我所知道的方法仅包括以下几种:

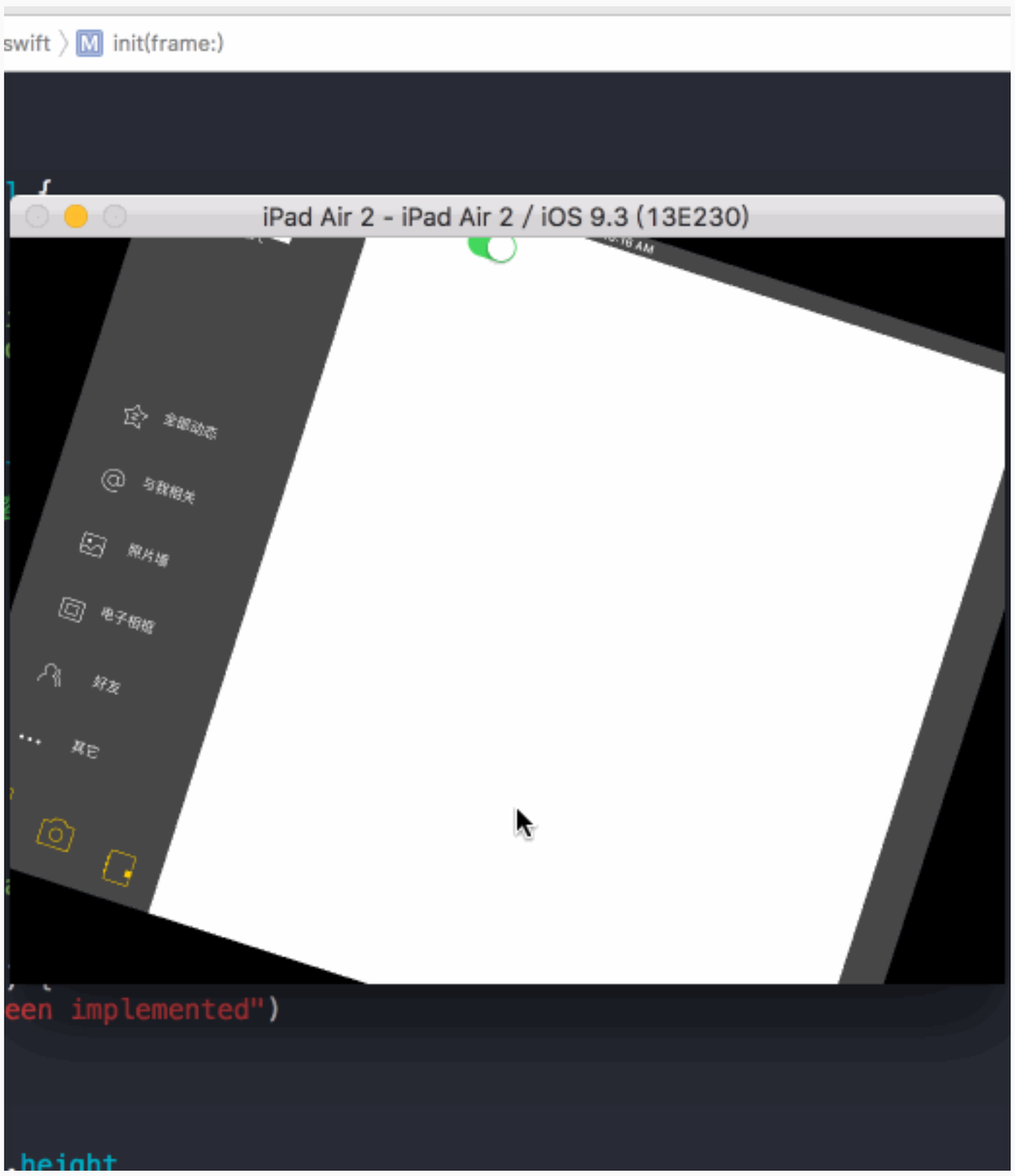
- 通过控制器的interfaceOrientation只读属性获取,iOS8后过期
- 能过状态栏的方向间接获取,UIApplication的只读属性statusBarOrientation,iOS9后过期
- 能过UIDevice只读属性orientation获取.需主动调用beginGeneratingDeviceOrientationNotifications开启通知
- 通过根控制器的view宽高推导获取,当高>宽为竖屏否则为横屏

Demo预览

下面是这个Demo的最终效果,我将通个下面的例子记录我认为合理的代码适配横竖屏的方式.



Demo预览慢速

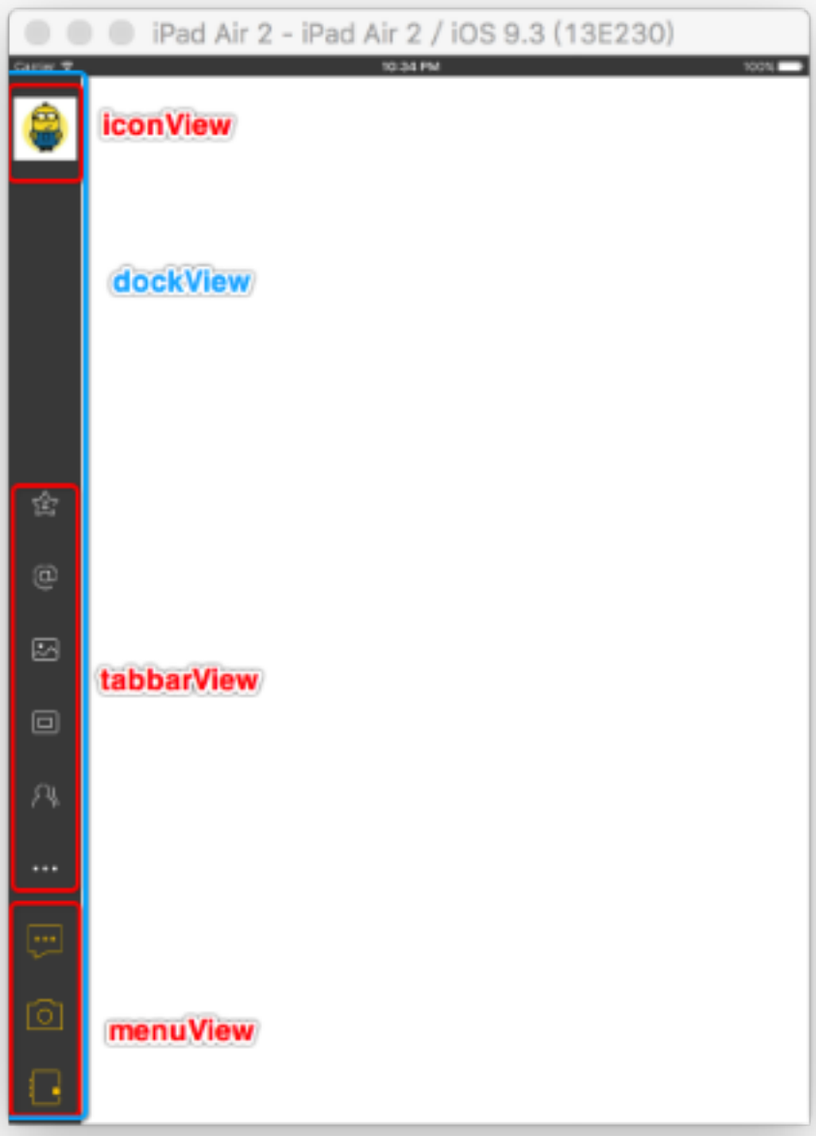


Demo预览正常

Demo分析

上面一些基础的知识将有助于理解Demo的做法,所以尽管有一点废话连篇的感觉好在也并不是一无是处.在讲解Demo的实现思路之前你可以在GitHub下载这个Demo,以便更方便的查看我讲到的代码.

Demo中最复杂的,横竖屏布局变化最大的部分就是左侧的菜单栏可以称它为Dock栏,通过旋转屏幕可以看到原生QQZone HD的Dock栏的变化.可以根据变化的特征将整个Dock栏分为三部分.一是顶部的头像 二是中间的类TabBar,我称它为TabBar 三是 底部的快捷导航菜单.因此DockView的subview包含iconButton,tabBarView,menuBar三个,而这三个subview又可以分另包含各自的子控件.



view层次结构

实际上实现QQZone for iPad屏幕的横竖屏的布局并不复杂.一个view要知道怎样在layoutSubviews中去布局其子view只需知道当前其superview的状态(横竖屏).在这里我声明了一个协议,这个协议只包含一个获取当前view是否是竖屏的方法.让每一个需要根据横竖屏动态变化的view都实现这个协议的方法,这样在layoutSubviews方法就可以询问当前应该怎么样布局子控件而当前状态是由父控件状态决定的.由此形成了屏幕状态的传递链,使得每个veiw只关心自身直接subview的布局.

```
UIViewisPortrait协议
protocol UIViewisPortrait: NSObjectProtocol {
func isPortrait() -> Bool
}
```

根控制器view的任意subview可都可以通过如下代码获取当前是否是竖屏

```
subview 如何获取superview状态
func isPortrait() -> Bool {
guard let superview = superview else { // 如果不存大superview默认返回竖屏
return true
}
return ((superview as? UIViewisPortrait)?.isPortrait())!
}
```

而根控制器view则直接通过宽高获取屏幕状态

```
func isPortrait() -> Bool {
return frame.width < frame.height
}
```

当前view知道是横竖屏后就可以直接在layoutsubviews布局子控件了,以menuBar为例

```
override func layoutSubviews() {
```

```
super.layoutSubviews()
guard subviews.count > 0 else {
return
}
var x, y, w, h:CGFloat
for (index, view) in subviews.enumerate() {
if isPortrait() == true {
w = frame.width
h = kDockItemHeight
x = 0
y = CGFloat(index) * h
view.frame = CGRect(x: x, y: y, width: w, height: h)
} else {
w = frame.width / CGFloat(subviews.count)
h = kDockItemHeight
x = CGFloat(index) * w
y = frame.height - h
view.frame = CGRect(x: x, y: y, width: w, height: h)
}
}
}
```

其它类型的所有子控件都可以用类似的方法进行布局,如此你只需要定义一个view并始之成为某个View的subview,在你定义的view中你可以随意的获取屏幕状态布局子控件了.

为了在根控制器View的layoutSubviewsr的方法中布局DockView,需要重写控制器的loadView方法,让控制器加载自定义的View. 如果你注意到原生QQZone for iPad的内容显示区域在横竖屏下的变化会发现在横竖屏下内容显示区域宽都是一样的,所以还需要在根控制器View中添加一个容器View以显示内容.

```
class HomePageView: UIView, UIViewisPortrait {
private lazy var dockView:DockView = DockView()
lazy var contentView: UIView = {
let contentView = UIView()
contentView.backgroundColor = UIColor.whiteColor()
// 作为容器View,子控制器的view将添加到容器view上
self.addSubview(contentView)
return contentView
}()
func isPortrait() -> Bool {
return frame.width < frame.height
}
override init(frame: CGRect) {
super.init(frame: frame)
addSubview(dockView)
dockView.backgroundColor = globalBackgroudColor
}
required init?(coder aDecoder: NSCoder) {
fatalError("init(coder:) has not been implemented")
}
override func layoutSubviews() {
super.layoutSubviews()
dockView.frame.size.height = frame.height
dockView.frame.size.width = isPortrait() ? kDockProtraitWidth : kDockLandscapeW
let x: CGFloat = dockView.frame.width
let y: CGFloat = 20
// 无论横竖屏,内容视图的宽都一样
let w: CGFloat = min(frame.width, frame.height) - kDockProtraitWidth
let h: CGFloat = frame.height - y
contentView.frame = CGRectMake(x, y, w, h)
}
}
```

最后由于内容区域不用区分横竖屏,因此内容区域的子视图可以只考虑竖屏的情况,以上可以说得不是很清楚,如果感觉有兴趣可下载原码参阅.

总结

以上重点仅仅是用代码进行iPad横竖屏适配方法的探讨,这里只是记录了我认为较为合理的方法,当然这种方法可能并不适用所有的布局,毕竟每个App都有自己独特的UI部分.如果觉得这种方法不好欢迎指出,我将虚心请教.如果这个方法对你的业务提供了一点点的灵感希望点个赞,以上完.

微信扫一扫

订阅每日移动开发及APP推广热点资讯

公众号：CocoaChina

我要投稿　收藏文章

分享到：

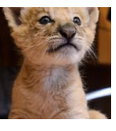
66

上一篇：如何写好一个UITableView（完整版）

相关资讯

- 前苹果员工透露苹果正在测试新iPad键盘配件
- iOS 7代码暗示新一代iPad Mini未配备Retina显示屏
- iOS 7优化Non-Retina iPad运行iPhone应用显示效果
- 开始你的iPhone和iPad应用开发之旅
- 美国社交网络Path发布iPad应用
- 分析师称Retina iPad mini将在下周发布 但会遭遇产能尴
- 分析师称iPhone 5S本月开始生产 新一代iPad mini今年推
- OS X 10.8.3再出测试版 2013年iPad mini有望赶超iPad
- Inside Mobile Apps本周免费榜分析
- 分析人士预测苹果iPadmini将蚕食iPad市场

我来说两句



你怎么看？快来评论一下吧！

发表评论

所有评论（0）