本文来自<u>丁同舟、王博文 随手记技术团队</u>,作者 yyuuzhu

前几天,随手记团队对 iPhone X 紧张的适配工作暂时告一段落。从最初在 iPhone X 模拟器上面运行,到 最后真机测试时完全适配。在这一路过来适配的历程中,我们整理出下面这份通用的 iPhone X 适配文档供 大家参考。

Always Remember

Intro

Ú

摘自 Designing for iPhone X:为了让你的应用在 iPhone X上 面完美运行,你需要将可视元素扩展至填充 整个展示窗口(屏幕)上,同时,你也需要保证如按钮、tab bar 等可交互控件,以及一些至关重要的信息 不会因为屏幕的圆角而被裁掉,或者被手机的「刘海」和虚拟「Home键」遮住。 Updating your App to work on the new display(iPhone X) involves extending visual elements to

fill the display's view port, while also keeping controls and critical information from getting clipped in the corners or covered by the sensor housing or home indicator. **UIKit and Auto Layout** 对于原生的 UIKit 控件来说,适配 iPhone X 是一件非常轻松的事。假如你的应用使用了许多原生控件并且

没有对它们做过多的自定义的话,例如像导航条 (Navigation Bars),列表 (Tables),集合视图 (Collection

Views), 这些控件会在 iPhone X 屏幕上自动调整其布局。

11:22 10/2017 @ 本月收入 \$0.00

未设顶罩,点此设置

\$0.00

本月支出

预算



The safe area also prevents content from underlapping the status bar, navigation bar, toolbar, and tab bar.

所有的应用应该附着于安全区域和 UIKit 定义的边距中,这可以让应用在各种设备或者横竖屏情况下都有 正确的布局。同时,安全区域可以保证页面的内容不会和状态条、导航条、工具条或者底部导航重叠。 另一官方文档 Positioning Content Relative to the Safe Area 中的一张图清晰地展示了 safe area 在应

用中代表的意义。 Figure 1 The safe area of an interface

Safe Area

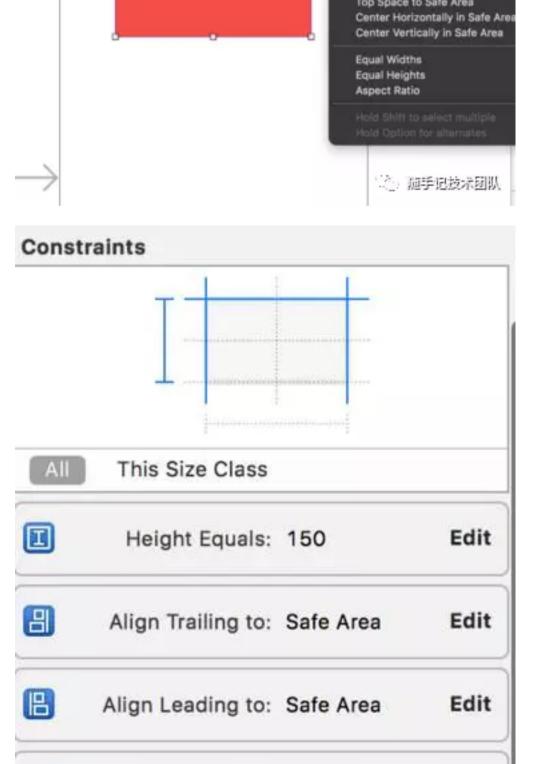
New in iOS 11, Apple is deprecating the top and bottom layout guides and replacing them with a single safe area layout guide: iOS 11中,苹果推出了 SafeAreaLayoutGuide 取代了 bottomLayoutGuide 和 topLayoutGuide,对于已

Top Space to Safe Area Center Horizontally in Safe Are Center Vertically in Safe Area Equal Heights Aspect Ratio

所以,使用 Auto Layout 布局的话, safe area 就相对于一个处于每个 view controller 的视图层级底层的

0 0 0

容器,我们的子控件只需要和它建立约束关系,就可以完美适配 iPhone X 安全区域



Align Top to: Safe Area



5.0, right: 5.0)

insets in this property are 0.

ght: 0.0)

your view controller.

其他

additionalSafeArealnsets 属性来计算。

视图层级中的时候,才能正确获取到 SafeAreaInsets,否则返回0。

例如在一个 view controller 的生命周期中追踪 self.view.safeAreaInsets 得到的结果:

来说,大概符合以下的规则:

//竖屏

programmatically.

as the navigation bar.

underneath tool palettes.

Example

Insets)")

Example:

right: 0 + additionalSafeAreaInsets.right) //横屏SafeAreaInsets = (top: heightForTopBars + additionalSafeAreaInsets.top, left: StatusBarHeight + additionalSafeAreaInsets.left, bottom: HomeIndicatorAreaHeight + additionalSafeAreaInsets.bottom, right: StatusBarHeight + additionalSafeAreaInsets.right) SafeAreaInsets 调用时机问题

If the view is not currently installed in a view hierarchy, or is not yet visible onscreen, the edge

时刻记住,SafeAreaInsets 并不是随时都能获取到的。 苹果的解释是在视图显示在屏幕上或者装载到一个

ViewController loadView() SafeAreaInsets :UIEdgeInsets(top: 0.0, left: 0.0, bottom: <mark>复制代码</mark>

bottom: heightForBottomBars + additionalSafeAreaInsets.bottom,

需要注意,在不同状态,例如竖屏和横屏下,获取到的 SafeAreaInsets 是不同的,对于 ViewController

For other views in the view hierarchy, the insets reflect only the portion of the view that is covered. For example, if a view is entirely within the safe area of its superview, the edge insets in this property are 0. 对于其他的视图,苹果的说法是只有当视图存在一部分被非安全区域遮挡的情况下,SafeAreaInsets 才会 返回相应的值。如果整个视图已经处于安全区域中,那么 SafeAreaInsets 返回 zero。

// Override to return a child view controller or nil. If non-nil, that view controller's ho me indicator auto-hiding will be used. If nil, self is used. Whenever the return value chan ges, -setNeedsHomeIndicatorAutoHiddenUpdate should be called. - (nullable UIViewController *)childViewControllerForHomeIndicatorAutoHidden API_AVAILABLE(ios(11.0)) API_UNAVAILABLE(watchos, tvos); // Controls the application's preferred home indicator auto-hiding when this view controlle r is shown. - (BOOL)prefersHomeIndicatorAutoHidden API_AVAILABLE(ios(11.0)) API_UNAVAILABLE(watchos, tv // This should be called whenever the return values for the view controller's home indicato r auto-hiding have changed. - (void)setNeedsUpdateOfHomeIndicatorAutoHidden API_AVAILABLE(ios(11.0)) API_UNAVAILABLE(wa tchos, tvos); iPhone X: Dealing with Home Indicator indicator 自动隐藏。 Discussion

UIEdgeInsets(top: 0.0, left: 0.0, bottom: 0.0, right: 0.0)

UIEdgeInsets(top: 88.0, left: 0.0, bottom: 0.0, right: 0.0)

@interface UIViewController (UIHomeIndicatorAutoHidden)

// Red View SafeAreaInsets

Home Indicator

会需要用到此特性。

88 (96 large title) 64 20 44 83

Designing for iPhone X iPhone X: Dealing with Home Indicator SafeAreaLayoutGuide Positioning Content Relative to the Safe Area Human Interface Guidelines Reference

• Designing for iPhone X

• SafeAreaLayoutGuide

https://developer.apple.com/videos/play/fall2017/801/ • iPhone X: Dealing with Home Indicator https://medium.com/the-traveled-ios-developers-guide/iphone-x-dealing-with-homeindicator-2e8e47f5647f

https://developer.apple.com/documentation/uikit/uiview/2891102-safearealayoutguide

 Positioning Content Relative to the Safe Area https://developer.apple.com/documentation/uikit/uiview/positioning_content_relative_to_the_s afe_area

• Human Interface Guidelines https://developer.apple.com/ios/human-interface-guidelines/overview/iphone-x/ 本文来自丁同舟、王博文 随手记技术团队

论坛热帖 热门文章

又上热搜! 小红书推出踩坑榜, ... 3 双十一预售遭遇服务器崩了,淘... 苹果M1 Max GPU跑分曝光: 性... 李楠评新苹果 MacBook Pro 20...

∠ 写文章

昨晚,苹果解释了什么叫"吓人"

注册 | 登录

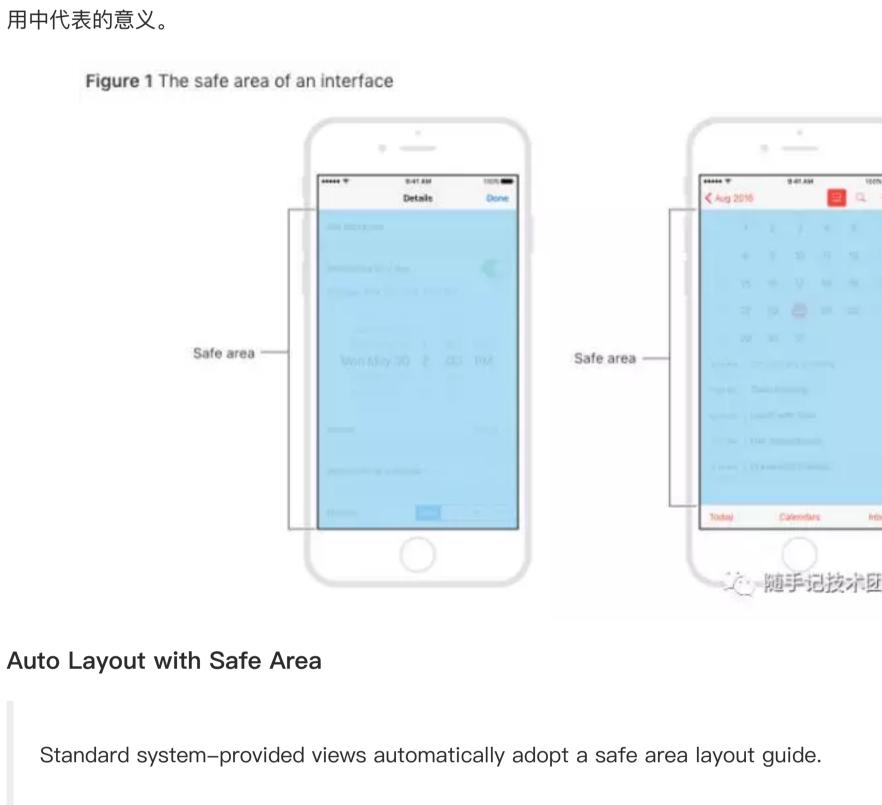
本周最热 | 本月最热

本周最热 | 本月最热

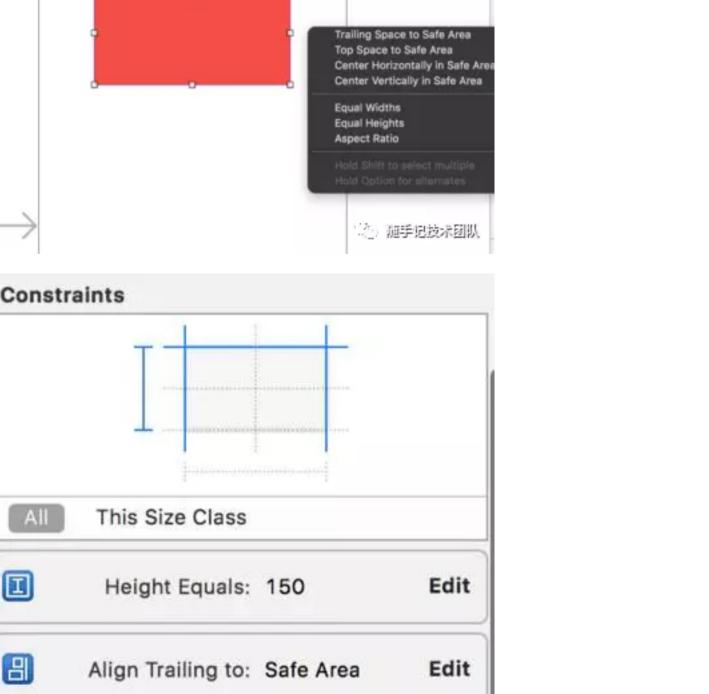
元宇宙 Cocoa社区 用户协议・隐私协议・权利声明・商务合作 京ICP备 11006519号-1 京ICP证 100954号

京公安网备 1101502037140号 京网文[2018]6281-491号 北京触控科技有限公司 版权所有@2020 Chukong Technologes,Inc 友情链接









Edit

加手见技术团队



另一个新的 API additionalSafeAreaInsets 则提供给开发人员能够自主扩展 safe area 区域的能力。顾名

Use this property to adjust the safe area insets of this view controller's views by the specified

guaranteed to be unobscured by the system status bar or by an ancestor-provided view such

amount. The safe area defines the portion of your view controller's visible area that is

You might use this property to extend the safe area to include custom content in your

interface. For example, a drawing app might use this property to avoid displaying content

print("view controller's additionalSafeAreaInsets set to " + "\(self.additionalSafeArea

复制代码

复制代码

思义,如果对这个属性主动赋值,那么整个视图的 safe area 区域便会发生变化。

var additionalSafeAreaInsets: UIEdgeInsets { get set }

override func viewDidAppear(_ animated: Bool) {

super.viewDidAppear(animated)

控制台输出如下 Origin SafeAreaInsets:UIEdgeInsets(top: 88.0, left: 0.0, bottom: 34.0, right: 0.0) 复制代码 view controller's additionalSafeAreaInsets set to UIEdgeInsets(top: 5.0, left: 5.0, bottom: Adjusted SafeAreaInsets: UIEdgeInsets(top: 93.0, left: 5.0, bottom: 39.0, right: 5.0)

SafeAreaInsets = (top: heightForTopBars + additionalSafeAreaInsets.top,

left: 0 + additionalSafeAreaInsets.left,

print("Origin SafeAreaInsets :" + "\(self.view.safeAreaInsets)")

print("Adjusted SafeAreaInsets:" + "\(self.view.safeAreaInsets)")

self.additionalSafeAreaInsets = UIEdgeInsetsMake(5, 5, 5, 5)

ViewController viewDidLoad() SafeAreaInsets: UIEdgeInsets(top: 0.0, left: 0.0, bottom: 0.0, right: 0.0) ViewController viewWillAppear() SafeAreaInsets :UIEdgeInsets(top: 0.0, left: 0.0, bottom: 0 .0, right: 0.0) ViewController viewDidLayoutSubviews() SafeAreaInsets: UIEdgeInsets(top: 44.0, left: 0.0, b ottom: 34.0, right: 0.0) ViewController viewDidAppear() SafeAreaInsets :UIEdgeInsets(top: 44.0, left: 0.0, bottom: 3 4.0, right: 0.0) 控制器根视图

For the view controller's root view, the insets account for the status bar, other visible bars,

and any additional insets that you specified using the additionalSafeAreaInsets property of

对于 view controller 的根视图, SafeAreaInsets 会根据各种 bar 的高度,以及开发者自己设置的

iPhone X - IOS 11.1 (15B5078b) 复制代码 // Blue View SafeAreaInsets

Home indicator 的设置类似于 prefersStatusBarStyle, iOS 11 增加了 UIViewController 的一个

UlHomeIndicatorAutoHidden 分类来控制 home 键的自动隐藏。通常在全屏播放视频,全屏游戏等场景下

需要注意 prefersHomeIndicatorAutoHidden,苹果官方并不确保这个重写这个方法一定可以让 home Override this method to signal your preference for displaying the visual indicator. The system takes your preference into account, but returning true is no guarantee that the indicator will be hidden. 值得一提的是,苹果并没有提供可以手动改变 home indicator 颜色的接口给开发者。因为苹果并不希望我 们手动更改 home indicator 的颜色,而应该遵循其自动切换颜色的特性(Home indicator 会根据底色的 不同自动切换黑色或白色)。

其他 Some Height Normal Height iPhone X Height Item UINavigationBar UIStatusBar UlTabBar Reference