3D

Core Animation 帮助你从二维的世界自由浮动;尽管它不是一个真正的三维框架, Core Animation 来帮助你设置二维物体在三维空间位置。换句话说,图层和动画仍然发生在两个维度,但你可以旋转和每个元素的2 d平面位置在三维空间一样:

CATransform3D中将使用一个特殊的性质:即转换层。CATransform3D 和CGAffineTransform类似,但除了缩放、倾斜和翻转在x和y方向,它还带来了第三维度:z。

简单的3D动画

原来你可以在Layer层做动画，现在你可以在三维扩展层中旋转、缩放、翻转等。

结合骨架、3D动画、MVC、单例、Storyboard综合

1、侧边栏同样是APP的一种常规的布局方式：优点：留出更多的空间给用户，将不常用的交互隐藏起来；

缺点：对于有些操作不能够直观的操作。

2、将对常用的APP设计进行模块化拆分、整合、总结。

3、拆分：

第一：一个tableview列表就可以实现侧边栏的选择，首先创建一个tableview，storyboard中拉入一个UItableview，关联一个SideMenuViewController.swift类，来进行交互；其中将MVC拆分开，这个因为业务简单，就将View的展示放在控制层，模型负责形成模型，采用单例的形式保存，保证初始化一次，不过多使用内存；

**全局常量**

let \_SingletonSharedInstance = Singleton()

class Singleton  {

    class var sharedInstance : Singleton {

        return \_SingletonSharedInstance

    }

}

嵌入结构体

structs do support static constants. By using a nested struct we can leverage its static constant as a class constant.

class Singleton {

    class var sharedInstance : Singleton {

        struct Static {

            static let instance : Singleton = Singleton()

        }

        return Static.instance

    }

}

多线程方式（OC 的方式）

**dispatch\_once**

class Singleton {

    class var sharedInstance : Singleton {

        struct Static {

            static var onceToken : dispatch\_once\_t = 0

            static var instance : Singleton? = nil

        }

        dispatch\_once(&Static.onceToken) {

            Static.instance = Singleton()

        }

        return Static.instance!

    }

}

此时侧边栏已经独立形成，采用MVC的形式；

func menuTransformForPercent(percent: CGFloat) -> CATransform3D {

var identity = CATransform3DIdentity

identity.m34 = -1.0/1000

}

为什么这个属性被称为m34 ?视图和层变换数学表示为二维矩阵。在一层的情况下变换矩阵,第三行第四列中的元素集你的z轴的角度。你可以设置这个元素直接应用所需的透视变换。

现在创建了一个CATransform3D的结构体，并设置他的m34为1.0/1000;

但是你为什么要使用1000camera距离?距离是用点表示和前面的场景,但使用什么值,事实是,你需要尝试不同的值,看看有什么特别的动画看起来不错。

|  |  |
| --- | --- |
| 距离 |  |
| 0.1...500 | 远景距离，效果明显，拉伸的感觉 |
| 750...2,000 | 内容清晰可见 |
| 2,000 and up | 几乎没什么效果的距离 |

let remainingPercent = 1.0 - percent let angle = remainingPercent \* CGFloat(-M\_PI\_2)

计算当前视图与菜单之间的角度

let rotationTransform = CATransform3DRotate(identity, angle, 0.0, 1.0, 0.0)

let translationTransform = CATransform3DMakeTranslation(menuWidth \* percent, 0, 0)

return CATransform3DConcat(rotationTransform, translationTransform)

你使用rotationTransform旋转层远离你y -轴。从左边的菜单移动,所以你也创建一个翻译转换沿着x轴移动它,最终menuWidth为100%。最后,将两个转换和返回结果。

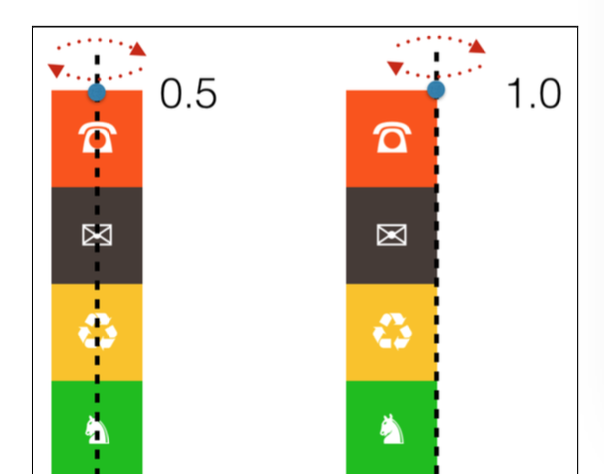
~~接下来就可以替换~~

~~menuViewController.view.frame.origin.x = menuWidth \* CGFloat(percent) – menuWidth~~

menuViewController.view.layer.transform = menuTransformForPercent(percent)

更改Layer层的锚点

原来的x锚点为0.5，在这个点旋转，会产生一些间距，将锚点设置为菜单与控制器的中间点—1.0;左右的旋转动画都围绕着锚点计算，当你设置Layer的位置时，你需要以这个锚点作为参考；意味着在设置位置之前最好改变Layer的anchorPoint，即锚点；

****

在viewDidLoad() 中设置menu的Layer，在Frame设置之前

menuViewController.view.layer.anchorPoint.x = 1.0

通过阴影创建透视的效果

阴影会使3D动画更加逼真，接下来设置左侧菜单阴影：

setToPercent():

menuViewController.view.alpha = CGFloat(max(0.2, percent))

设置View的透明度随着位移改变，效果还是不错的；

但是仔细观察会发现，cell之间的那条线会有白色的曲线，这是因为在第一次触发这个开关后，依旧没有设置3D 动画的参数和Layer层转换；

viewDidLoad():

setToPercent(0.0)