结合Autolayerot 与 Animation

Autolayerot本质为相对布局，在此不累述，如何在已经布局好的View上加载动画或是自动布局与动画综合使用，而不是纯代码。

其实动画约束并不比动画属性更加困难，只是有点不同，通常只需用一个新的约束替换现有的约束；你只需要更改约束中的常量就好，加上动画就可以了。

例子中我们点击加号，头部View高度拉伸，其他布局不变，这样我们就可以在这个View上做其他的操作，实现很简单，将头部view的height Constraint拖拽到文件中：

@IBOutlet var menuHeightConstraint: NSLayoutConstraint!

NSLayoutConstraint是一个类，代表你下IB中创建的约束，就好比时按钮、label、image view 你可以创建一个约束的outlet；

在button的action中加入

isMenuOpen = !isMenuOpen menuHeightConstraint.constant = isMenuOpen ? 200.0 : 60.0 titleLabel.text = isMenuOpen ? "Select Item" : "Packing List"

运行点击，你会发现就这么实现了头部View拉高，那么加上之前的动画： UIView.animateWithDuration(1.0, delay: 0.0, usingSpringWithDamping: 0.4, initialSpringVelocity: 10.0, options: .CurveEaseIn, animations: {

self.view.layoutIfNeeded()

}, completion: nil}

当你在动画闭包中更改View的属性，它将会动画的形式达到你的期望，Auto Layout完成计算后依旧约束view的bounds和centers；

**注释：layoutIfNeeded**使用此方法强制立即进行layout,从当前view开始，此方法会遍历整个view层次(包括superviews)请求layout。因此，调用此方法会强制整个view层次布局。

旋转动画

将+选转45°

let angle = self.isMenuOpen ? CGFloat(M\_PI\_4) : 0.0 self.buttonMenu.transform = CGAffineTransformMakeRotation(angle)

动画差不多已经实现，接下来就是检查更改动画以达到预期，

接下来将lable的动画加上去，UIView类有个方法contraints()，返回当前View的所欲的约束，

for c in titleLabel.superview!.constraints() {

println(c)

}

在ismenuopen != isMenuOpen 下边加上

for constraint in titleLabel.superview!.constraints() as [NSLayoutConstraint] {

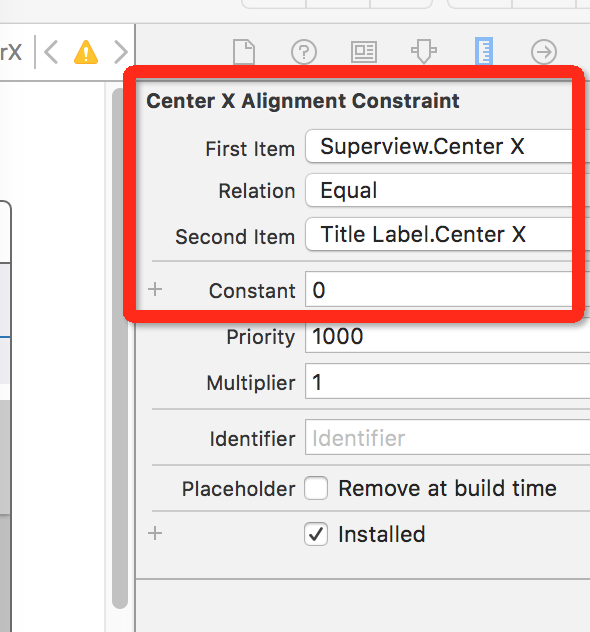
if constraint.secondItem as? NSObject == titleLabel && constraint.secondAttribute == .CenterX {

constraint.constant = isMenuOpen ? 100.0 : 0.0

continue

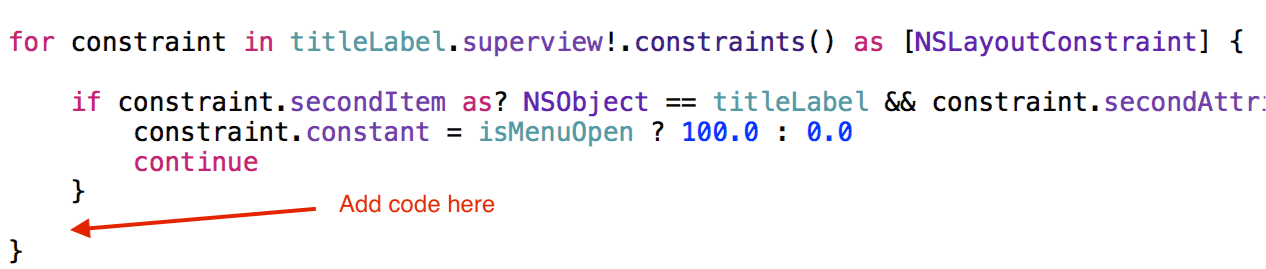
} }

这个图就是IB中点击约束出现的内容，item是Anyobject,代表内容可以在这里查看，在代码中更改First 或者 Second,以达到改变视图的位置



做到这里更改的只是约束中的常量，但是在约束中，属性值是不确定的，如果想要更改视图之间的比例、或更改约束中的其他部分，你需要remove约束，然后添加新的约束。

之前有说过可以遍历找到一个视图的所有的约束，在此不采用outlet的方式，采用遍历获得约束，



if constraint.firstItem as? NSObject == titleLabel && constraint.firstAttribute == .CenterY {

titleLabel.superview!.removeConstraint(constraint)

//add new constraint

continue

}

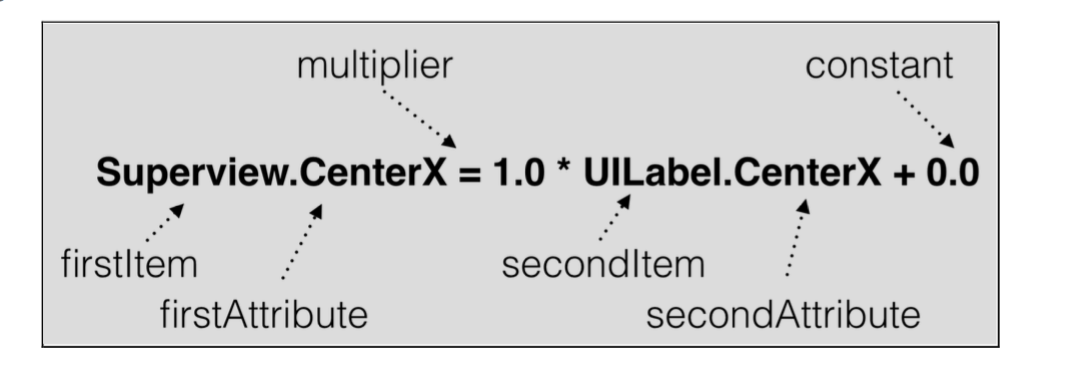
在这里的要确保firstItem、secondItem时titleLabel，

接下来添加新的约束：

let newConstraint = NSLayoutConstraint( item: titleLabel, attribute: .CenterY, relatedBy: .Equal,

toItem: titleLabel.superview!, attribute: .CenterY, multiplier: isMenuOpen ? 0.67 : 1.0, constant: 5.0)

newConstraint.active = true



NSLayoutConstraint 初始化时需要大量的属性：

item : 这个方程式中的第一个属性，在这里是titlelabel

attribute: frist item的新的约束的属性；

relatedBy：这个约束代表相等或不等，

toItem：方程式中第二个Item，在这里是superView

attribute: second item的新的约束的属性；

multiplier :Item之间的比例；

constant：方程式的常量。