**Frame animation**

学习如何创建一个非常特殊类型的动画Frame animation ；

创建项目，采取横屏的方式，选中storyboard中viewcontroller-》orientation(Landscape);与Deployment info-🡪仅选中Device Orientation 的Landscape left

在loadWalkAnimation() 中添加

penguin.animationImages = walkFrames penguin.animationDuration = animationDuration / 3 penguin.animationRepeatCount = 3

1、animationImages  ：存储所有帧图像的帧动画。walkFrames声明包含用户界面图像对象的数组所以你只需要将它分配给企鹅。animationImages设置动画帧。

2、animationDuration  ：这告诉UIKit动画的一个迭代应该持续多久,因为你将有动画重复三次(见下文),设置这个总animationDuration的长度的三分之一。

3、animationRepeatCount:  这个控制动画的重复计数在这种情况下动画会重复三次。

在viewdidload中调用

var isLookingRight: Bool = true { didSet {

let xScale: CGFloat = isLookingRight ? 1 : -1 penguin.transform = CGAffineTransformMakeScale(xScale, 1) slideButton.transform = penguin.transform

} }

这段代码设置x轴的企鹅，滑动按钮（1、-1）根据isLookingRight的value而转变方向，保证企鹅可以面对正确的方向:

@IBAction func leftButton(sender: AnyObject) {

isLookingRight = false

penguin.startAnimating()

}

此时点击左侧按钮，企鹅会转过头，重复走3次；因为调用iamgeView的startAnimating() 。

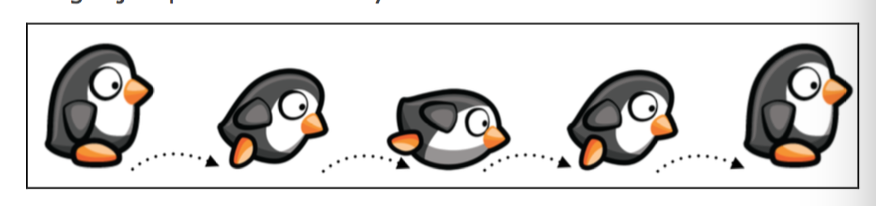
@IBAction func leftButton(sender: AnyObject) {

UIView.animateWithDuration(animationDuration, delay: 0.0, options: .CurveEaseOut, animations: {

self.penguin.center.x -= self.walkSize.width }, completion: nil)

向左偏移

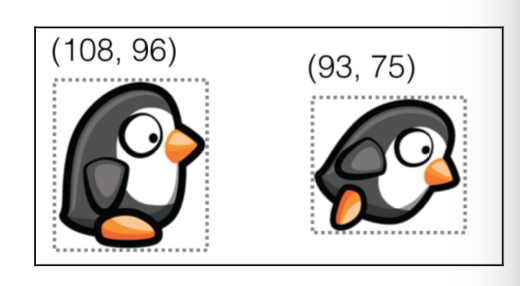
现在企鹅可以四处走动,是时候让他跳和滑行了。



添加滑行动画

penguin.animationImages = slideFrames penguin.animationDuration = animationDuration penguin.animationRepeatCount = 1

在此初始化动画数组，执行时间以及重复次数；



动画执行不同阶段的Frame；行走的时候时108\*96，当滑行的时候时93\*75，如果他们有一致的尺寸，那么需要大量空白的区域在彼此图片之间；

解决这个问题很简单,那就是把内容模式的图像视图的默认值方面填补中心或左上角，

接下来手动调整和重新定位图像视图创建一个美丽的动画；

首先, 你做任何动画之前需要设置所需的图像视图Frame,这样可以确保Frame正确的大小的当的在屏幕上可见。

penguin.frame = CGRect( x: penguin.frame.origin.x, y: penguinY + (walkSize.height - slideSize.height), width: slideSize.width, height: slideSize.height)

这段代码将企鹅移动图像视图下一点来弥补的短帧幻灯片动画和调整图像视图,slideSize相匹配。

UIView.animateWithDuration(animationDuration - 0.02, delay: 0.0, options: .CurveEaseOut, animations: {

self.penguin.center.x += self.isLookingRight ? self.slideSize.width : -self.slideSize.width

}, completion: {\_ in // animation is complete

})

运行中会将企鹅的高度缩小了一点，

在comlpletion中添加

self.penguin.frame = CGRect( x: self.penguin.frame.origin.x, y: self.penguinY, width: self.walkSize.width, height: self.walkSize.height)

self.loadWalkAnimation()

来恢复位置