实验编号： 10 **四川师大《IOS》实验报告 2018** 年 **11** 月 **14** 日

### **计算机科学学院** 2016 级 4 班 实验名称： 视图动画 \_

姓名：\_\_蒋星\_\_\_ 学号：2016110418 指导老师：\_\_李贵洋\_\_ 实验成绩:\_\_\_\_\_

**实验 十 \_\_\_**视图动画**\_\_\_\_\_**

1. 实验目的及要求
2. 掌握视图基本动画的原理和使用；
3. 掌握动力学动画的原理和使用；
4. 实验要求
5. 认真填写实验报告，要求附加部分运行界面和主要代码；
6. 对设计好的程序，检查输出是否符合预期，如有错请分析错误原因并解决；
7. 实验内容
8. 基于UIView.animation实现如下动画效果：
   1. 改变视图的位置
   2. 改变视图的大小
   3. 改变视图的transform
9. 基于UIView.transition实现如下动画效果：
   1. 改变视图的背景颜色；
   2. 切换两个子视图，观察切换后视图层次的变化情况；
10. 基于UIDynamicAnimtor实现简单的力学动画：
    1. 可以掉落方块；
    2. 有碰撞；
    3. 可以反弹；
11. 实验主要流程、基本操作或核心代码、算法片段（该部分如不够填写，请另加附页）
12. 基于UIView.animation实现如下动画效果：
    1. 改变视图的位置
    2. 改变视图的大小
    3. 改变视图的transform
       * 程序代码：

@IBAction func StopMoveClicke(\_ sender: UIButton) {

timer?.invalidate()

//在几秒内 透明度变成0

UIView.animate(withDuration: 4, delay: 0, options: [.autoreverse,.curveLinear], animations: {

self.myview.alpha = 0

self.myview.center.x += self.myview.bounds.width

self.myview.backgroundColor = UIColor.red

//旋转

self.myview.transform = CGAffineTransform(rotationAngle:CGFloat(Double.pi))

//伸缩

//.scaledBy(x: 0.5, y: 0.5)

self.myview.transform = CGAffineTransform.identity

self.myview.transform = CGAffineTransform(rotationAngle:CGFloat(Double.pi))

self.myview.transform = CGAffineTransform.identity

self.myview.transform = CGAffineTransform(rotationAngle:CGFloat(Double.pi))

self.myview.transform = CGAffineTransform.identity

self.myview.transform = CGAffineTransform(rotationAngle:CGFloat(Double.pi))

self.myview.transform = CGAffineTransform.identity

self.myview.transform = CGAffineTransform(rotationAngle:CGFloat(Double.pi))

self.myview.transform = CGAffineTransform.identity

self.myview.transform = CGAffineTransform(rotationAngle:CGFloat(Double.pi))

self.myview.transform = CGAffineTransform.identity

}, completion: { (finish) in

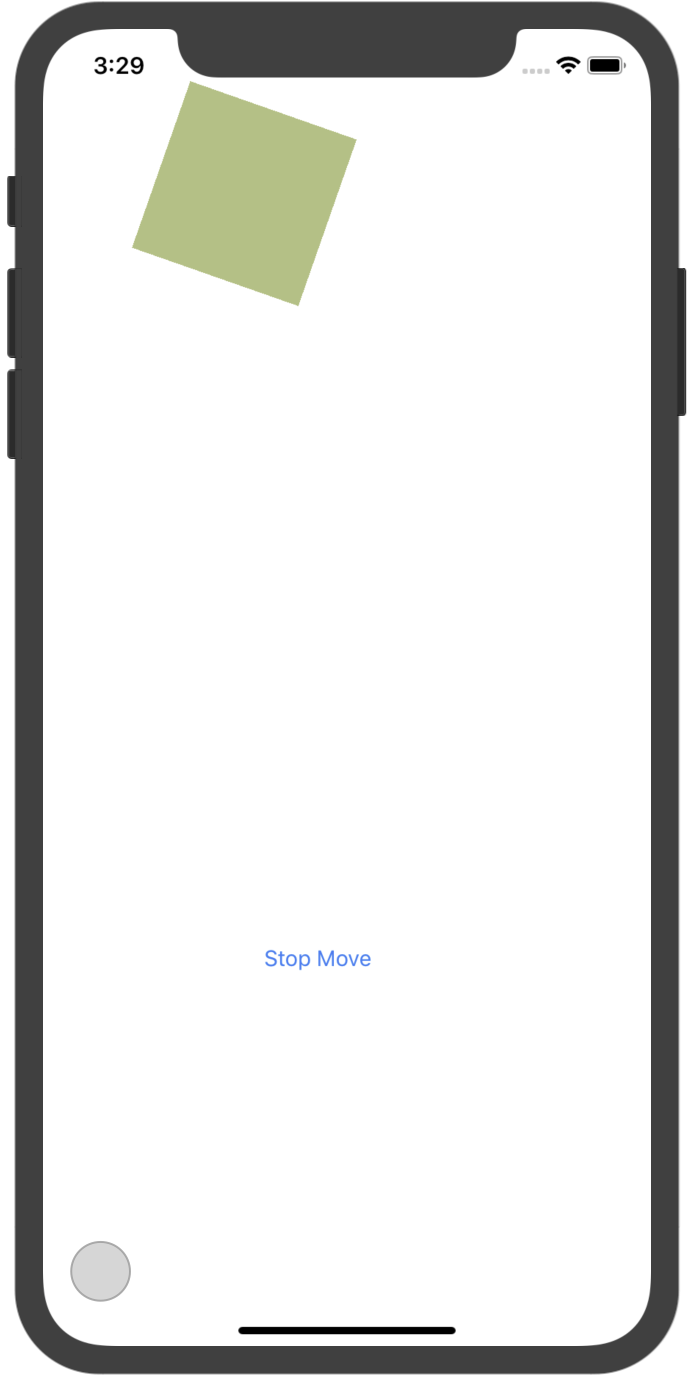
if finish {

self.myview.removeFromSuperview()

}

})

* + - 运行结果：



1. 基于UIView.transition实现如下动画效果：
   1. 改变视图的背景颜色；
   2. 切换两个子视图，观察切换后视图层次的变化情况；
      * 程序代码：

@IBAction func StopMoveClicke(\_ sender: UIButton) {

//翻书动画

UIView.transition(with: myview, duration: 2, options: .transitionCurlUp, animations: { self.myview.backgroundColor = UIColor.green }, completion: nil)

//从上向下翻转

UIView.transition(with: myview2, duration: 2, options: .transitionFlipFromTop, animations: { self.myview2.backgroundColor = UIColor.green }, completion: nil)

//从左向右边翻转

UIView.transition(with: myview3, duration: 2, options: .transitionFlipFromLeft, animations: { self.myview3.backgroundColor = UIColor.green }, completion: nil)

/\*

let imageView = UIImageView(frame:CGRect(x:50,y:50,width:100,height:100))

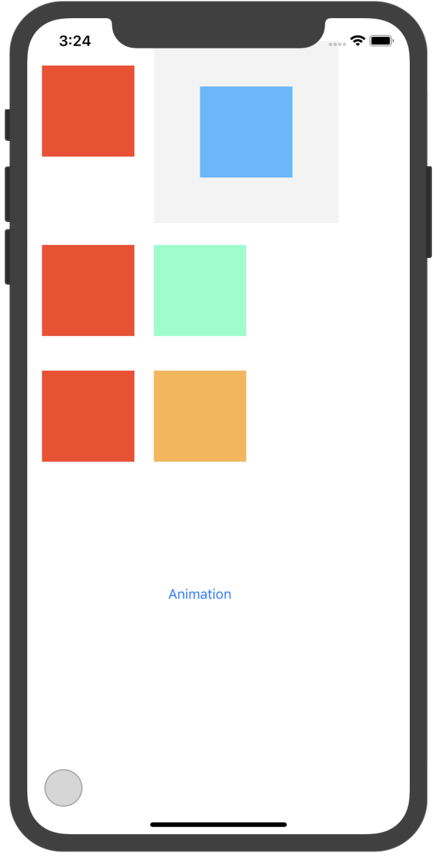
imageView.image = UIImage(named:"xjp")

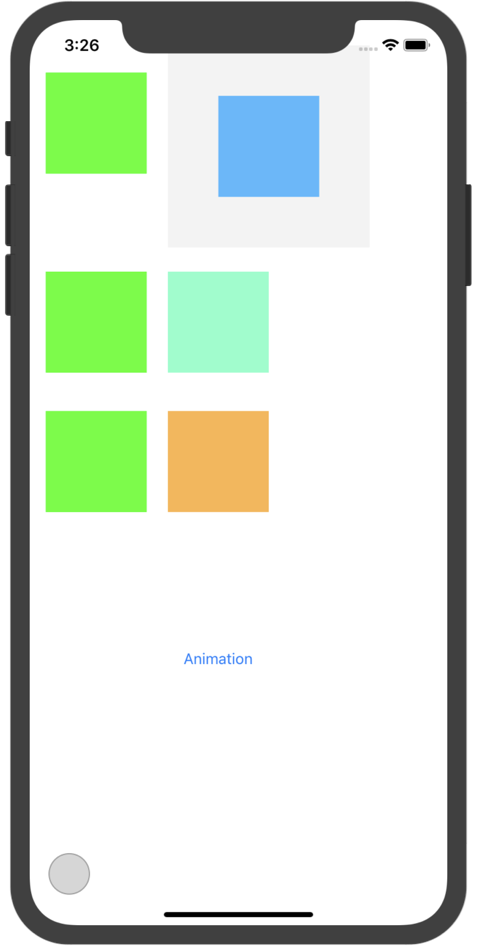
UIView.transition(from: myview4, to: imageView, duration: 2, options: .transitionFlipFromLeft, completion: nil)

\*/

}

* + - 运行结果：





1. 基于UIDynamicAnimtor实现简单的力学动画：
   1. 可以掉落方块；
   2. 有碰撞；
   3. 可以反弹；
      * 程序代码：

class ViewController: UIViewController {

lazy var animator = UIDynamicAnimator(referenceView:self.backView)

@IBOutlet weak var backView: UIView!

//物理引擎

let gravity = UIGravityBehavior() //重力效果

let collision = UICollisionBehavior() // 碰撞效果

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view, typically from a nib.

//物理效果添加到视图中

animator.addBehavior(gravity)

animator.addBehavior(collision)

//防止跌落出边框 越界

collision.translatesReferenceBoundsIntoBoundary = true

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

@IBAction func add(\_ sender: UIButton) {

let width = Int(self.backView.bounds.width / 10)

let randx = Int(arc4random() % 10) \* width

let label = UILabel(frame:CGRect(x:randx,y:20,width:width,height:width))

label.backgroundColor = UIColor.purple

label.text = "王"

label.textColor = UIColor.white

label.textAlignment = .center

self.backView.addSubview(label)

//给视图施加重力影响

gravity.addItem(label)

collision.addItem(label)

}

@IBAction func up(\_ sender: UIButton) {

gravity.gravityDirection = CGVector(dx:0,dy:-1)

}

@IBAction func down(\_ sender: UIButton) {

gravity.gravityDirection = CGVector(dx:0,dy:1)

}

@IBAction func left(\_ sender: UIButton) {

gravity.gravityDirection = CGVector(dx:-1,dy:0)

}

@IBAction func right(\_ sender: UIButton) {

gravity.gravityDirection = CGVector(dx:1,dy:0)

}

@IBAction func deleteAll(\_ sender: Any) {

for item in backView.subviews{

if item is UILabel {

item.removeFromSuperview()

gravity.removeItem(item)

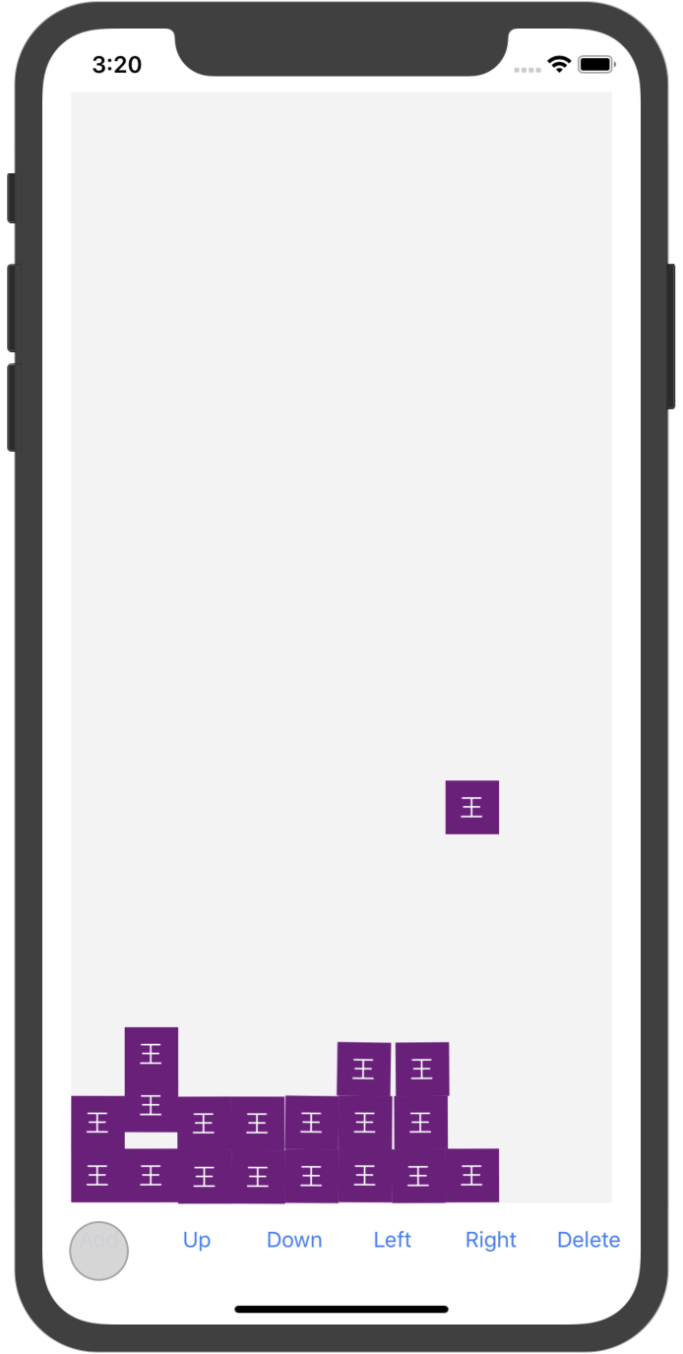
collision.removeItem(item)

}

}

}

* + - 运行结果：



1. 实验结果的分析与评价（该部分如不够填写，请另加附页）

Github地址：https://github.com/KikcerGoer/study-swift/tree/master/homework

顶呱呱

注：实验成绩等级分为（90－100分）优，（80－89分）良，(70-79分)中，（60－69分）及格，（59分）不及格。