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：\_\_\_\_蒋宇童\_\_\_\_\_ 学号：\_2016110419\_\_\_\_\_\_\_\_\_ 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. 实验内容
5. 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
6. 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；
7. 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；
8. 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）
9. 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：

//

// ViewController.swift

// UIView.animation

//

// Created by jiang on 2018/11/21.

// Copyright © 2018年 jiang. All rights reserved.

//

import UIKit

class ViewController: UIViewController {

@IBOutlet weak var myView: UIView!

let imageView = UIImageView(frame: CGRect(x: 20, y: 20, width: 100, height: 100))

override func viewDidLoad() {

super.viewDidLoad()

imageView.image=UIImage(named: "1")

//myView2=myView

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

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

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

if myView == nil

{

}

else

{

UIView.animate(withDuration: 4, delay: 1, options: [.curveLinear,.autoreverse], animations: {

self.myView.backgroundColor= colorLiteral(red: 0.4867877364, green: 0.7669275999, blue: 0.932189405, alpha: 1)

self.myView.center.x=self.view.bounds.width/2

self.myView.transform = CGAffineTransform(scaleX: 5, y:5)

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

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

self.myView.transform = CGAffineTransform.identity

},completion: nil)

}

}

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

if myView == nil

{

}

else

{

UIView.transition(with: myView, duration: 2, options: .transitionCurlUp, animations: {

self.myView.backgroundColor= colorLiteral(red: 0.9610558152, green: 0.5509537458, blue: 0.01276976243, alpha: 1)

self.myView.transform = CGAffineTransform(scaleX: 4, y: 4)

}, completion: nil)

}

}

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

if myView == nil

{

}

else

{

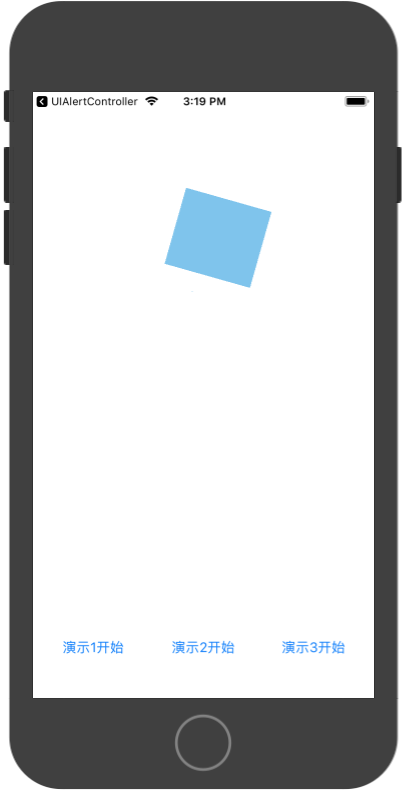
UIView.transition(from: myView, to: imageView, duration: 2, options: .transitionFlipFromRight, completion: nil)

}

}

}

* + - 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：

//

// ViewController.swift

// UIView.animation

//

// Created by jiang on 2018/11/21.

// Copyright © 2018年 jiang. All rights reserved.

//

import UIKit

class ViewController: UIViewController {

@IBOutlet weak var myView: UIView!

let imageView = UIImageView(frame: CGRect(x: 20, y: 20, width: 100, height: 100))

override func viewDidLoad() {

super.viewDidLoad()

imageView.image=UIImage(named: "1")

//myView2=myView

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

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

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

if myView == nil

{

}

else

{

UIView.animate(withDuration: 4, delay: 1, options: [.curveLinear,.autoreverse], animations: {

self.myView.backgroundColor= colorLiteral(red: 0.4867877364, green: 0.7669275999, blue: 0.932189405, alpha: 1)

self.myView.center.x=self.view.bounds.width/2

self.myView.transform = CGAffineTransform(scaleX: 5, y:5)

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

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

self.myView.transform = CGAffineTransform.identity

},completion: nil)

}

}

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

if myView == nil

{

}

else

{

UIView.transition(with: myView, duration: 2, options: .transitionCurlUp, animations: {

self.myView.backgroundColor= colorLiteral(red: 0.9610558152, green: 0.5509537458, blue: 0.01276976243, alpha: 1)

self.myView.transform = CGAffineTransform(scaleX: 4, y: 4)

}, completion: nil)

}

}

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

if myView == nil

{

}

else

{

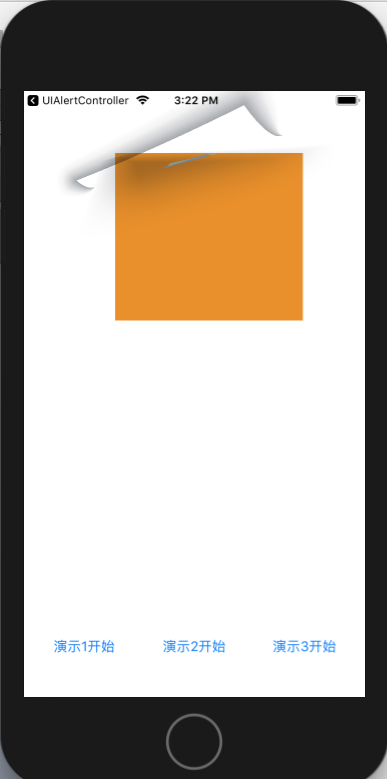
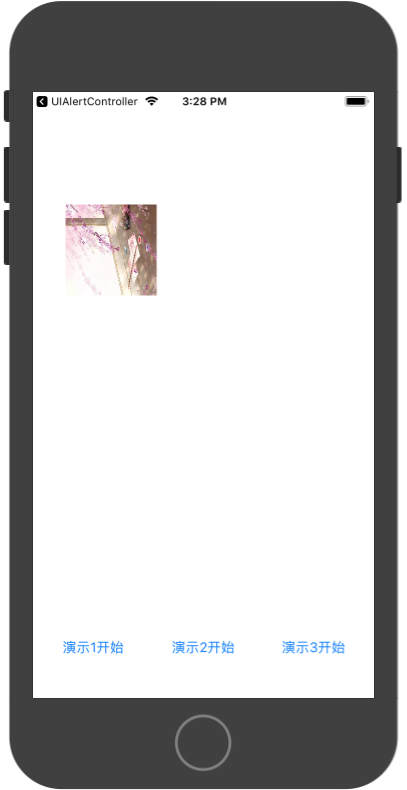
UIView.transition(from: myView, to: imageView, duration: 2, options: .transitionFlipFromRight, completion: nil)

}

}

}

* + - 运行结果：



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：

//

// ViewController.swift

// UIDynamicAnimtor

//

// Created by jiang on 2018/11/21.

// Copyright © 2018年 jiang. All rights reserved.

//

import UIKit

class ViewController: UIViewController {

var timer :Timer?

lazy var animation = UIDynamicAnimator(referenceView: self.myView)

let gravity = UIGravityBehavior()

let collision = UICollisionBehavior()

@IBOutlet weak var myView: UIView!

override func viewDidLoad() {

super.viewDidLoad()

animation.addBehavior(gravity)

animation.addBehavior(collision)

collision.translatesReferenceBoundsIntoBoundary=true

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

timer=Timer.scheduledTimer(withTimeInterval: 1, repeats: true, block: { [weak weakSelf=self](myTimer) in

weakSelf?.myView.center.x+=20

})

//timer?.invalidate()

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

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

let width = Int(myView.bounds.width/10)

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

let lable = UILabel(frame: CGRect(x: randx, y: 0, width: width, height: width))

lable.backgroundColor= colorLiteral(red: 1, green: 0.6215450618, blue: 0.9306755428, alpha: 1)

lable.text="jyt"

lable.textAlignment = .center

myView.addSubview(lable)

gravity.addItem(lable)

collision.addItem(lable)

}

@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 mydelete(\_ sender: Any) {

for lable in myView.subviews

{

if lable is UILabel

{

gravity.removeItem(lable)

collision.removeItem(lable)

lable.removeFromSuperview()

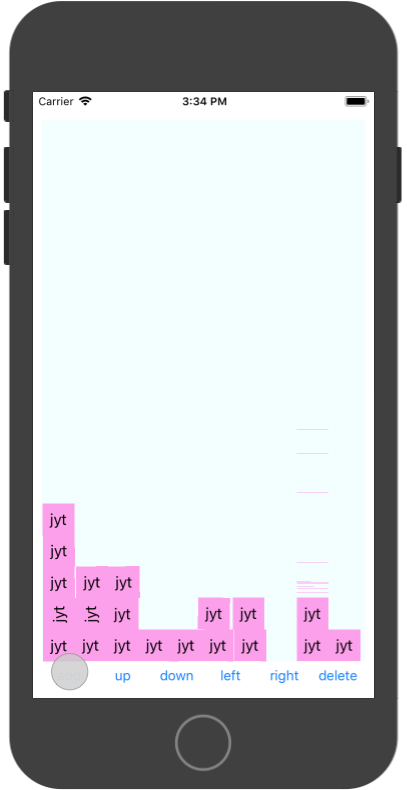
}

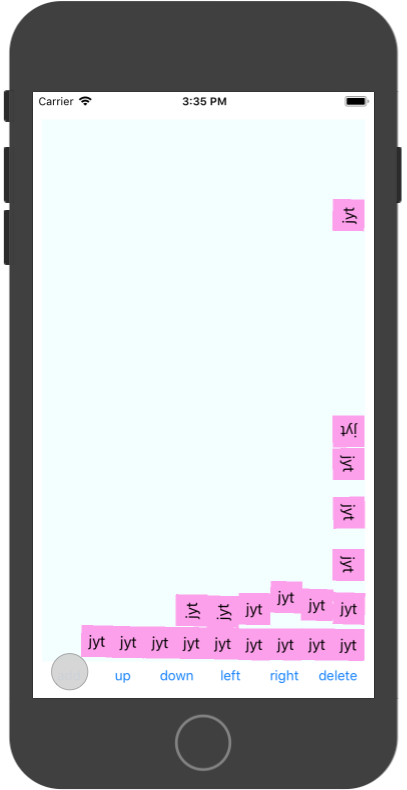
}

}

}

* + - 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/jiangyutong/swiftWork/tree/master/代码**](https://github.com/jiangyutong/swiftWork/tree/master/代码)

这次实验是关于动画的。真心觉得iOS比较强大，他的这些都已经封装好了，你相当于欧就是掉一个一个函数就可以了，并且他可以模拟重量让元素仿佛有了重力一样。

感觉做出来都应用很真实。

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。