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：\_\_樊琳\_\_ 学号：\_2016110408\_ 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：

//

// AppDelegate.swift

// 10

//

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

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

//

import UIKit

@UIApplicationMain

class AppDelegate: UIResponder, UIApplicationDelegate {

var window: UIWindow?

func application(\_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

// Override point for customization after application launch.

return true

}

func applicationWillResignActive(\_ application: UIApplication) {

// Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.

// Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should use this method to pause the game.

}

func applicationDidEnterBackground(\_ application: UIApplication) {

// Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.

// If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.

}

func applicationWillEnterForeground(\_ application: UIApplication) {

// Called as part of the transition from the background to the active state; here you can undo many of the changes made on entering the background.

}

func applicationDidBecomeActive(\_ application: UIApplication) {

// Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.

}

func applicationWillTerminate(\_ application: UIApplication) {

// Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.

}

}

//

// ViewController.swift

// 10

//

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

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

//

import UIKit

class ViewController: UIViewController {

@IBOutlet weak var myview2: UIView!

@IBOutlet var myview: UIView!

var timer:Timer?

override func viewDidLoad() {

super.viewDidLoad()

// 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

// self.myview2.center.x+=20

// })

}

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

let imageview=UIImageView(frame: CGRect(x: 100, y: 100, width: 200, height: 200))

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

UIView.transition(from: myview2, to: imageview, duration: 2, options: .transitionFlipFromRight, completion: nil)

// UIView.transition(with: myview2, duration: 1, options: .transitionCurlUp, animations: {

// self.myview2.backgroundColor=UIColor.red

// }, completion: nil)

//

// timer?.invalidate()

//

// UIView.transition(with: myview2, duration: 1, options: .transitionCurlUp, animations: {

// self.myview2.backgroundColor=UIColor.red

//

// }, completion: nil)

// UIView.animate(withDuration: 2, delay: 1, options: [.curveEaseIn], animations: {

//// self.myview2.alpha=0

//// self.myview2.center.x=self.view.bounds.width

//// self.myview2.backgroundColor=UIColor.green

//// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi)).scaledBy(x: 0.5, y: 0.5)

// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi))

// self.myview2.transform=CGAffineTransform.identity

// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi))

// self.myview2.transform=CGAffineTransform.identity

// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi))

// self.myview2.transform=CGAffineTransform.identity

// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi))

// self.myview2.transform=CGAffineTransform.identity

// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi))

// self.myview2.transform=CGAffineTransform.identity

// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi))

// self.myview2.transform=CGAffineTransform.identity

//

// }) { (finished) in

// self.myview2.removeFromSuperview()

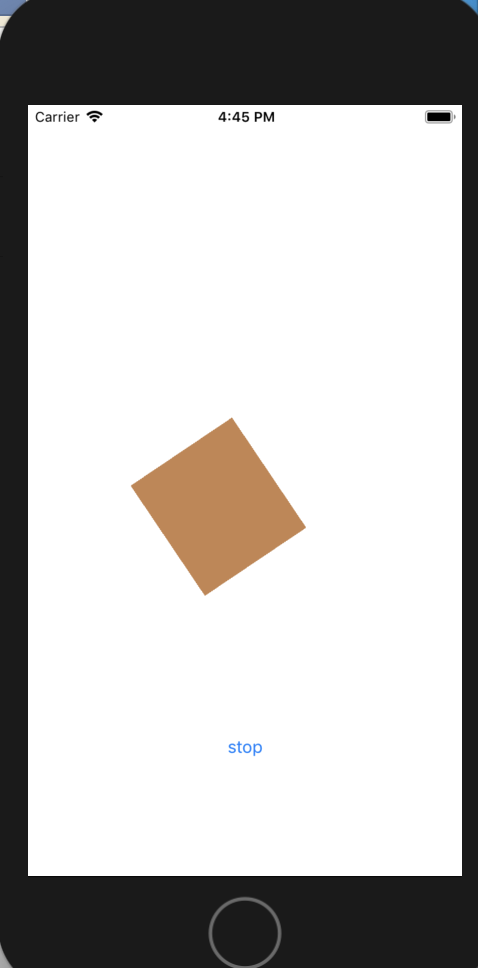
// }

// }

}

}

* + - 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：

//

// AppDelegate.swift

// 10

//

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

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

//

import UIKit

@UIApplicationMain

class AppDelegate: UIResponder, UIApplicationDelegate {

var window: UIWindow?

func application(\_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

// Override point for customization after application launch.

return true

}

func applicationWillResignActive(\_ application: UIApplication) {

// Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.

// Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should use this method to pause the game.

}

func applicationDidEnterBackground(\_ application: UIApplication) {

// Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.

// If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.

}

func applicationWillEnterForeground(\_ application: UIApplication) {

// Called as part of the transition from the background to the active state; here you can undo many of the changes made on entering the background.

}

func applicationDidBecomeActive(\_ application: UIApplication) {

// Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.

}

func applicationWillTerminate(\_ application: UIApplication) {

// Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.

}

}

//

// ViewController.swift

// 10

//

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

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

//

import UIKit

class ViewController: UIViewController {

@IBOutlet weak var myview2: UIView!

@IBOutlet var myview: UIView!

var timer:Timer?

override func viewDidLoad() {

super.viewDidLoad()

// 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

// self.myview2.center.x+=20

// })

}

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

//

//

let imageview=UIImageView(frame: CGRect(x: 100, y: 100, width: 200, height: 200))

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

UIView.transition(from: myview2, to: imageview, duration: 2, options: .transitionFlipFromRight, completion: nil)

UIView.transition(with: myview2, duration: 1, options: .transitionCurlUp, animations: {

self.myview2.backgroundColor=UIColor.red

}, completion: nil)

timer?.invalidate()

UIView.transition(with: myview2, duration: 1, options: .transitionCurlUp, animations: {

self.myview2.backgroundColor=UIColor.red

}, completion: nil)

// UIView.animate(withDuration: 2, delay: 1, options: [.curveEaseIn, .repeat], animations: {

//// self.myview2.alpha=0

//// self.myview2.center.x=self.view.bounds.width

//// self.myview2.backgroundColor=UIColor.green

//// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi)).scaledBy(x: 0.5, y: 0.5)

//// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi))

//// self.myview2.transform=CGAffineTransform.identity

//// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi))

//// self.myview2.transform=CGAffineTransform.identity

//// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi))

//// self.myview2.transform=CGAffineTransform.identity

//// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi))

//// self.myview2.transform=CGAffineTransform.identity

//// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi))

//// self.myview2.transform=CGAffineTransform.identity

//// self.myview2.transform=CGAffineTransform(rotationAngle: CGFloat(Double.pi))

//// self.myview2.transform=CGAffineTransform.identity

//

// }) { (finished) in

// self.myview2.removeFromSuperview()

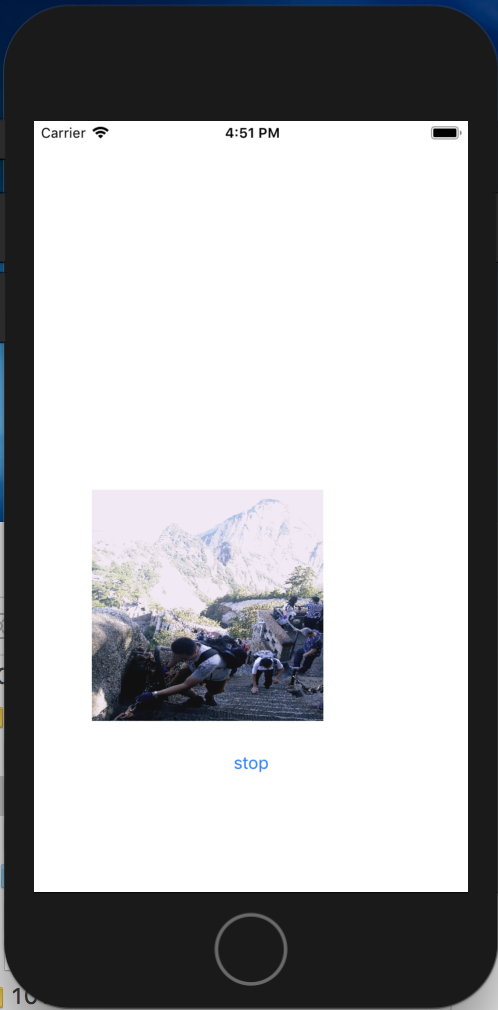
// }

// }

}

}

* + - 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：

//

// AppDelegate.swift

// 10.2

//

// Created by student on 2018/12/5.

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

//

import UIKit

@UIApplicationMain

class AppDelegate: UIResponder, UIApplicationDelegate {

var window: UIWindow?

func application(\_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

// Override point for customization after application launch.

return true

}

func applicationWillResignActive(\_ application: UIApplication) {

// Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.

// Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should use this method to pause the game.

}

func applicationDidEnterBackground(\_ application: UIApplication) {

// Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.

// If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.

}

func applicationWillEnterForeground(\_ application: UIApplication) {

// Called as part of the transition from the background to the active state; here you can undo many of the changes made on entering the background.

}

func applicationDidBecomeActive(\_ application: UIApplication) {

// Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.

}

func applicationWillTerminate(\_ application: UIApplication) {

// Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.

}

}

//

// ViewController.swift

// 10.2

//

// Created by student on 2018/12/5.

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

//

import UIKit

class ViewController: UIViewController {

@IBOutlet weak var backview: UIView!

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

let gravety=UIGravityBehavior()

let collsion=UICollisionBehavior()

override func viewDidLoad() {

super.viewDidLoad()

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

animator.addBehavior(gravety)

animator.addBehavior(collsion)

collsion.translatesReferenceBoundsIntoBoundary=true

}

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

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

let randx=Int(arc4random())%Int(view.bounds.width)

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

lable.backgroundColor=UIColor.red

lable.text="a"

lable.textAlignment = .center

backview.addSubview(lable)

gravety.addItem(lable)

collsion.addItem(lable)

}

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

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

}

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

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

}

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

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

}

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

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

}

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

for item in backview.subviews{

if item is UILabel{

item.removeFromSuperview()

gravety.removeItem(item)

collsion.removeItem(item)

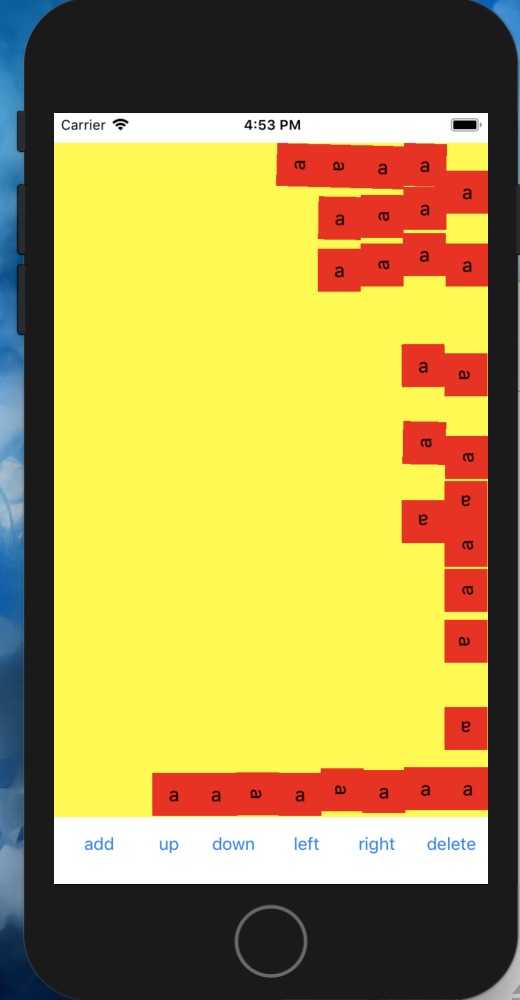
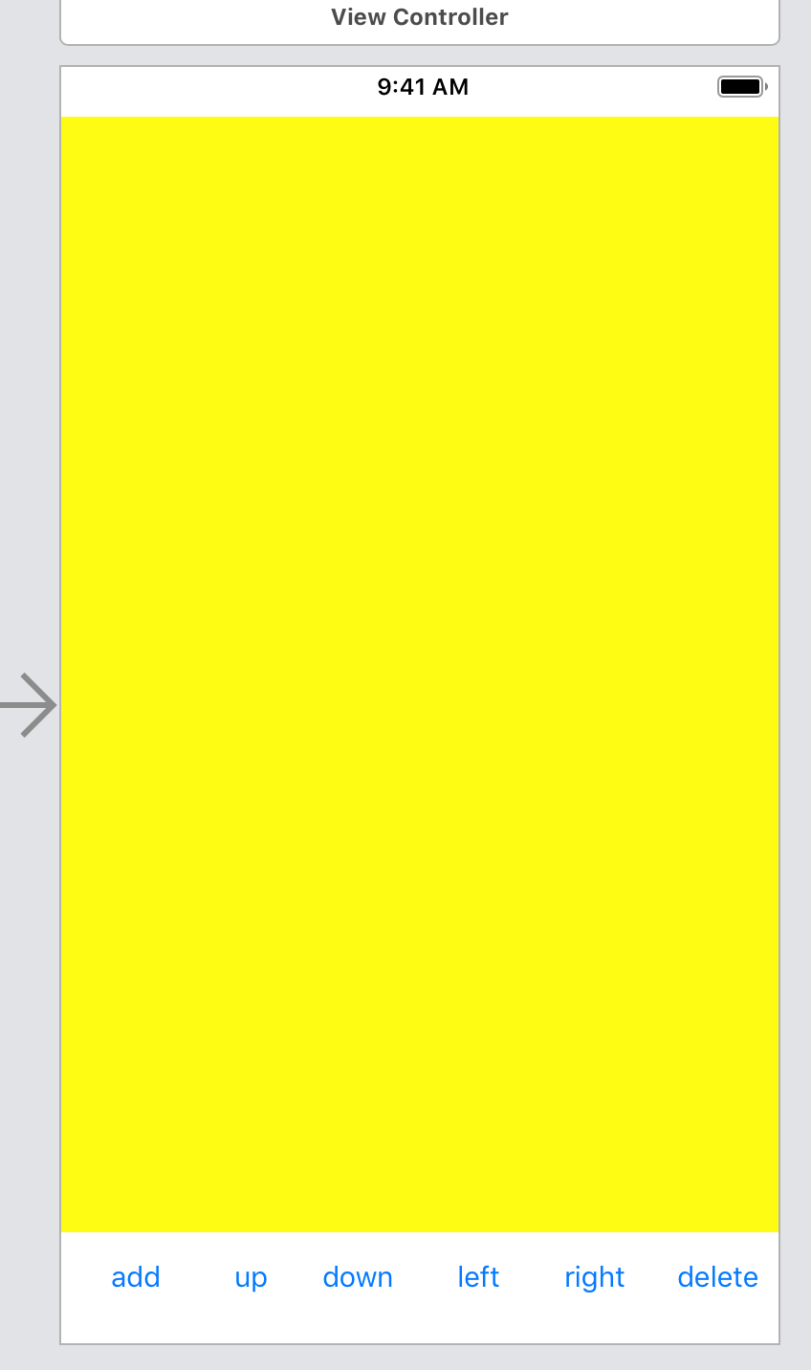
}

}

}

}

* + - 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）

动画是我们做项目必不可少的东西。在swift语言中运用动画并不是很难，都已经封装好了，调用就可以了。最后一个是实验关于重力和碰撞的运动也不是很难，也可以通过直接调用函数来实现。

Github地址：

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。