实验编号： 9 **四川师大《IOS》实验报告 2018** 年 **11** 月 **7** 日

### **计算机科学学院** 2016 级 4 班 实验名称： Gesture、UIAlertController、 ScrollView

姓名：\_蒋宇童\_\_\_\_\_\_\_\_ 学号：\_2016110419\_\_\_\_\_\_\_\_\_ 指导老师：\_\_李贵洋\_\_ 实验成绩:\_\_\_\_\_

**实验 九 \_\_\_\_\_** Gesture、UIAlertController、 ScrollView **\_\_\_\_\_\_\_**

1. 实验目的及要求
2. 理解并掌握iOS多点手势识别的相关技术；
3. 实验内容
4. Gesture
   1. 分别采用代码随机位置大小生成和直接拖拽的方式产生多个视图；
   2. 采用简单的动画进行移动；
   3. 给视图加上阴影(layer)；
   4. 可全部清空子视图；
   5. 视图支持手势（pan移动、tap删除、pinch缩放、rotation旋转）；

提示：Pinch的scale属性可用于调整frame

rotation需要用transform属性实现

1. 实现UIAlertController交互
   1. 显示ActionSheet并进行交互；
   2. 显示Login Alert并进行交互；
2. 一个界面使用两个scrollView
   1. 在一个scrollView中可进行多张图片横屏滚动浏览(相册)，需要有pagecontrol进行提示；
   2. 在另一个scrollView中可放大缩小；

提示：需用delegate

1. 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）
2. Gesture
   1. 分别采用代码随机位置大小生成和直接拖拽的方式产生多个视图；
   2. 采用简单的动画进行移动；
   3. 给视图加上阴影(layer)；
   4. 可全部清空子视图；
   5. 视图支持手势（pan移动、tap删除、pinch缩放、rotation旋转）；

提示：Pinch的scale属性可用于调整frame

rotation需要用transform属性实现

* 程序代码：

//

// ViewController.swift

// gesture

//

// Created by jiang on 2018/11/19.

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

//

import UIKit

class ViewController: UIViewController {

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

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

let y=Int(arc4random())%Int(view.bounds.height)

let label = UILabel(frame: CGRect(x: x, y: y, width: 30, height: 30))

label.text="jyt"

label.backgroundColor=UIColor.red

label.textAlignment = .center

label.layer.shadowColor=UIColor.gray.cgColor

label.layer.shadowOffset=CGSize(width: 5, height: 5)

label.layer.shadowOpacity=1

let panRecognizer=UIPanGestureRecognizer(target: self, action: #selector(pan(recognizer:)))

label.addGestureRecognizer(panRecognizer)

let tagRecognizer=UITapGestureRecognizer(target: self, action: #selector(tag(recognizer:)))

label.addGestureRecognizer(tagRecognizer)

tagRecognizer.numberOfTapsRequired=1

tagRecognizer.numberOfTouchesRequired=2

let pinchRecognizer=UIPinchGestureRecognizer(target: self, action: #selector(pinch(recognizer:)))

label.addGestureRecognizer(pinchRecognizer)

let rotationRecoginzer=UIRotationGestureRecognizer(target: self, action: #selector(rotation(recoginzer:)))

label.addGestureRecognizer(rotationRecoginzer)

label.isUserInteractionEnabled=true

view.addSubview(label)

}

@objc func pan(recognizer:UIPanGestureRecognizer)

{

switch recognizer.state {

case .changed:

fallthrough

case .ended:

let translation=recognizer.translation(in: self.view)

recognizer.view?.center.x+=translation.x

recognizer.view?.center.y+=translation.y

recognizer.setTranslation(.zero, in: self.view)

default:

break

}

}

@objc func tag(recognizer:UITapGestureRecognizer)

{

switch recognizer.state {

case .recognized:

recognizer.view?.removeFromSuperview()

default:

break

}

}

@objc func pinch(recognizer:UIPinchGestureRecognizer)

{

switch recognizer.state {

case .changed:

fallthrough

case .ended:

recognizer.view?.bounds=CGRect(x: 0, y: 0, width: (recognizer.view?.bounds.width)!\*recognizer.scale, height: (recognizer.view?.bounds.height)!\*recognizer.scale)

recognizer.scale=1

default:

break

}

}

@objc func rotation(recoginzer:UIRotationGestureRecognizer)

{

let rotation=recoginzer.rotation

recoginzer.view?.transform=(recoginzer.view?.transform.rotated(by:rotation))!

recoginzer.rotation=0

}

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

for lable in view.subviews

{

if lable is UILabel

{

UIView.animate(withDuration:1)

{

let x=Int(arc4random())%Int(self.view.bounds.width)

let y=Int(arc4random())%Int(self.view.bounds.height)

lable.center.x=CGFloat(x)

lable.center.y=CGFloat(y)

}

}

}

}

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

for lable in view.subviews

{

if lable is UILabel

{

lable.removeFromSuperview()

}

}

}

override func viewDidLoad() {

super.viewDidLoad()

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

}

override func didReceiveMemoryWarning() {

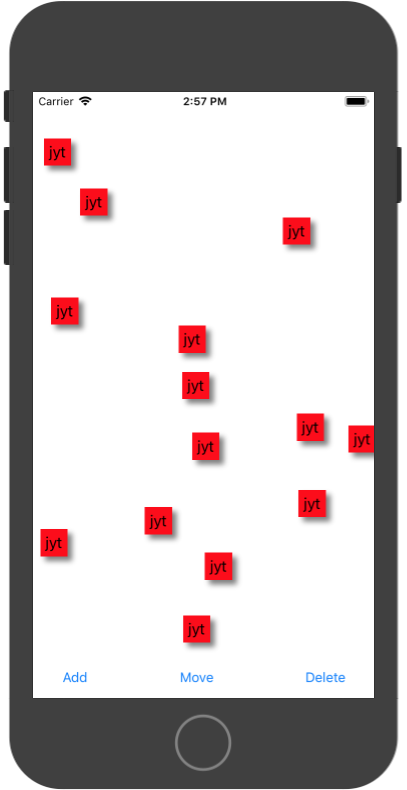
super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

}

* 运行结果：



1. 实现UIAlertController交互
   1. 显示ActionSheet并进行交互；
   2. 显示Login Alert并进行交互；

程序代码：

//

// ViewController.swift

// UIAlertController

//

// Created by jiang on 2018/11/19.

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

//

import UIKit

class ViewController: UIViewController,UITableViewDelegate,UITableViewDataSource {

@IBOutlet weak var myView: UITableView!

var users=[User]()

let loginAlert=UIAlertController(title: "登录", message: nil, preferredStyle:.alert)

let registerAlert=UIAlertController(title: "注册", message: nil, preferredStyle: .alert)

let sureAlert=UIAlertController(title: nil , message:"登录成功", preferredStyle:.alert)

let warningAlert=UIAlertController(title: nil, message:"该用户不存在", preferredStyle: .alert)

let errorAlert=UIAlertController(title: nil, message: "该用户已经存在", preferredStyle: .alert)

var images=["baiyun","chengxi","furong","jinping","nangang","qilihe","shangjie","wuhou"]

func tableView(\_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

return users.count

}

func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

let userTableViewCell=tableView.dequeueReusableCell(withIdentifier: "userTableViewCell") as! UserTableViewCell

userTableViewCell.userName.text=users[indexPath.row].userName

userTableViewCell.passWord.text=users[indexPath.row].passWord

userTableViewCell.images.image=UIImage(named: images[indexPath.row%7])

userTableViewCell.images.layer.cornerRadius=userTableViewCell.images.frame.size.width/2

userTableViewCell.images.clipsToBounds=true

return userTableViewCell

}

override func viewDidLoad() {

super.viewDidLoad()

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

let user1=User(userName: "11", passWord: "22")

users.append(user1)

sureAlert.addAction(UIAlertAction(title: "确定", style: .default, handler: nil))

warningAlert.addAction(UIAlertAction(title: "确定", style: .default, handler: nil))

errorAlert.addAction(UIAlertAction(title: "确定", style: .default, handler: nil))

loginAlert.addTextField{(textField) in textField.placeholder="请输入的userName"}

loginAlert.addTextField{(textField) in textField.placeholder="请输入的passWord"}

loginAlert.addAction(UIAlertAction(title: "确定", style: .default, handler: {(action) in self.judgeUser()}))

loginAlert.addAction(UIAlertAction(title: "取消", style: .default, handler: nil))

registerAlert.addTextField{(textField) in textField.placeholder="请输入的userName"}

registerAlert.addTextField{(textField) in textField.placeholder="请输入的passWord"}

registerAlert.addAction(UIAlertAction(title: "确定", style: .default, handler: {(action) in self.addUser()}))

registerAlert.addAction(UIAlertAction(title: "取消", style: .default, handler: nil))

}

func judgeUser()

{

guard let username=loginAlert.textFields![0].text,let password=loginAlert.textFields![1].text

else {

return

}

var flage=false

for user in users

{

if user.passWord==password&&user.userName==username

{

flage=true

break

}

}

if flage==true

{

self.present(sureAlert, animated: true, completion: nil)

}

else

{

self.present(warningAlert, animated: true, completion: nil)

}

}

func addUser()

{

guard let username=registerAlert.textFields![0].text,let password=registerAlert.textFields![1].text

else {

return

}

var flage=false

for user in users

{

if user.passWord==password&&user.userName==username

{

flage=true

break

}

}

if flage==true

{

self.present(errorAlert, animated: true, completion: nil)

}

else

{

let user=User(userName: username, passWord: password)

users.append(user)

}

myView.reloadData()

}

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

let alert=UIAlertController(title: nil, message: nil, preferredStyle: .actionSheet)

alert.addAction(UIAlertAction(title: "登录", style: .default, handler: {(action) in self.present(self.loginAlert, animated: true, completion: nil)}))

alert.addAction(UIAlertAction(title: "注册", style: .default, handler: {(action) in self.present(self.registerAlert, animated: true, completion: nil)}))

self.present(alert, animated: true, completion: nil)

}

override func didReceiveMemoryWarning() {

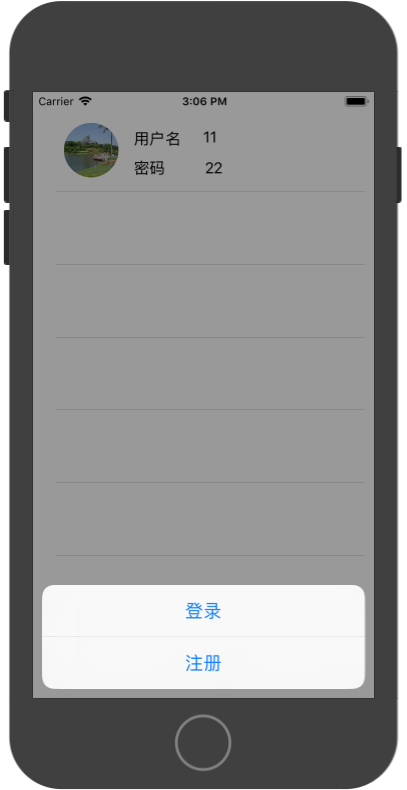
super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

}

运行结果：

1. 一个界面使用两个scrollView
   1. 在一个scrollView中可进行多张图片横屏滚动浏览(相册)，需要有pagecontrol进行提示；
   2. 在另一个scrollView中可放大缩小；

提示：需用delegate

* 程序代码：

//

// ViewController.swift

// scrcollView

//

// Created by jiang on 2018/11/20.

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

//

import UIKit

class ViewController: UIViewController,UIScrollViewDelegate{

@IBOutlet weak var pagControll: UIPageControl!

@IBOutlet weak var scrollView: UIScrollView!

override func viewDidLoad() {

super.viewDidLoad()

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

for i in 1...5

{

let imageView=UIImageView(image: UIImage(named: "\(i)"))

imageView.frame=CGRect(x: CGFloat (i-1) \* scrollView.bounds.width, y: 0, width: scrollView.bounds.width, height: scrollView.bounds.height)

imageView.contentMode = .scaleAspectFit

scrollView.addSubview(imageView)

scrollView.isPagingEnabled=true

}

scrollView.contentSize=CGSize(width: scrollView.bounds.width\*5, height: scrollView.bounds.height)

pagControll.numberOfPages=5

pagControll.currentPage=0

pagControll.isUserInteractionEnabled=true

// scrollView.minimumZoomScale=0.2

// scrollView.maximumZoomScale=5

scrollView.delegate=self

scrollView.showsHorizontalScrollIndicator=false

}

// func viewForZooming(in scrollView: UIScrollView) -> UIView? {

// return scrollView.subviews.first

// }

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

func scrollViewDidEndDecelerating(\_ scrollView: UIScrollView) {

pagControll.currentPage=Int(scrollView.contentOffset.x/scrollView.bounds.width)

}

@IBAction func pageControlAction(\_ sender: UIPageControl) {

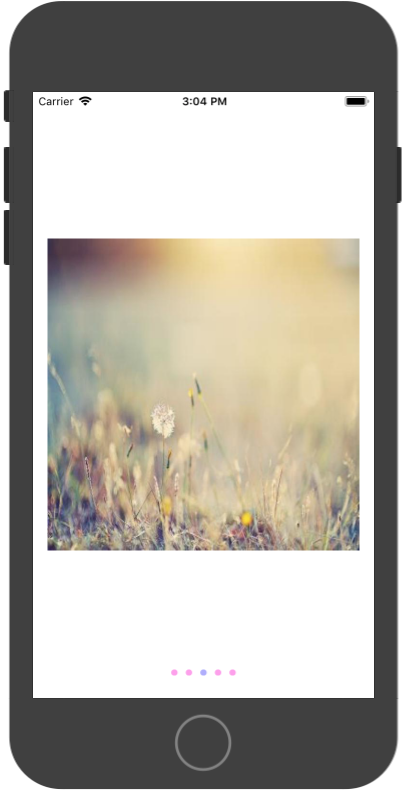
let rec=CGRect(x: CGFloat(pagControll.currentPage) \* scrollView.bounds.width, y: 0, width: scrollView.bounds.width, height: scrollView.bounds.height)

scrollView.scrollRectToVisible(rec, animated: true)

}

}

* 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)

这次的实验是做手势和UIAlertController交互和scrollView。通过这次的实验我学会了iOS中怎么用手势。我觉得手势是很有趣的。UIAlertController是应用当中很常见的。我感觉这次实验是很有趣，让我更喜欢iOS。

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。