实验编号： 4 **四川师大《IOS》实验报告 2018** 年 **9** 月 **26** 日

### **计算机科学学院** 2016 级 4 班 实验名称： 自定制视图、文件、Playground可视化

姓名： 郭周倩 学号： 2016110413 指导老师：\_\_李贵洋\_\_ 实验成绩:\_\_\_\_\_

**实验\_四\_ \_**自定制视图、文件、Playground可视化**\_**

1. 实验目的及要求
2. 掌握沙盒文件的操作；
3. 掌握自定制视图的定义及使用；
4. 掌握playground下可视化开发方法；
5. 实验要求
6. 认真填写实验报告，要求附加部分运行界面和主要代码；
7. 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；
8. 实验内容
9. **文件缓存处理**
   1. 判断沙盒的Document目录下是否存在某文件夹，如果没有则新建一个该文件夹；
   2. 判断是否该文件夹下存在一个图片文件，如果存在该文件，读取该文件到一个图片对象中并进行显示，如果不存在则从网上下载一张图片并保存到该图片文件中；
10. **自定制视图**
    1. 从UIView中派生一个自定制的View；
    2. 绘制一个椭圆（或则自己喜欢的任何图形）；
    3. 新建视图对象并进行显示；
11. **代码版hello world（在视图控制器中加入代码）**
    1. 代码中生成label（outlet）和button；
    2. 将label和button加入根view中；
    3. button添加像self（target）发射action（selector）的操作；

实现clicked响应代码（selector）

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. **文件缓存处理**
   1. 判断沙盒的Document目录下是否存在某文件夹，如果没有则新建一个该文件夹；
   2. 判断是否该文件夹下存在一个图片文件，如果存在该文件，读取该文件到一个图片对象中并进行显示，如果不存在则从网上下载一张图片并保存到该图片文件中；

* 程序代码：

import PlaygroundSupport

import UIKit

//第一题：创建一个文件目录；查询是否有图片，有则加载，没有则网上下载到沙盒中

let fileManager = FileManager.default

if var docpath = fileManager.urls(for: .documentDirectory, in: .userDomainMask).first{

docpath.appendPathComponent("aaa")

if !fileManager.fileExists(atPath: docpath.path){

try fileManager.createDirectory(at: docpath, withIntermediateDirectories: true, attributes: nil)

print(docpath.path)

}

print("成功构造新的aaa文件夹")

docpath.appendPathComponent("0.jpg")

if !fileManager.fileExists(atPath: docpath.path){ //判断是否有0.jpg图片

let imageUrl = URL(string: "https://ss0.bdstatic.com/70cFuHSh\_Q1YnxGkpoWK1HF6hhy/it/u=4198824076,697431934&fm=27&gp=0.jpg")!

let imageData = try! Data(contentsOf: imageUrl)

//print(docpath.path)

print("不存在图片0.jpg,正在创建！！！")

try? imageData.write(to: docpath)

}

else{

//print(docpath.path)

print("已经存在图片0.jpg！！！")

let imageData = try! Data(contentsOf: docpath)

let image = UIImage(data:imageData)

let imageView = UIImageView(image: image)

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

let view = UIView(frame: CGRect(x: 0, y: 0, width: 200, height: 200))

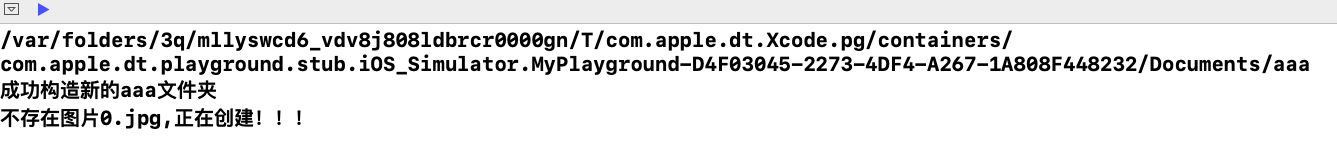
view.backgroundColor = UIColor.cyan

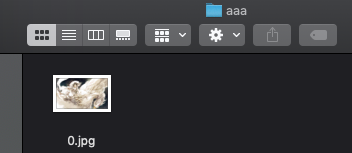
view.addSubview(imageView)

}

}

* 运行结果：





1. **自定制视图**
   1. 从UIView中派生一个自定制的View；
   2. 绘制一个椭圆（或则自己喜欢的任何图形）；
   3. 新建视图对象并进行显示；

* 程序代码：

class MyView: UIView{

override func draw(\_ rect: CGRect) {

let path = UIBezierPath()

path.move(to: CGPoint(x: rect.size.width/2, y: 0))

path.addLine(to: CGPoint(x: rect.size.width, y: rect.size.height))

path.addLine(to: CGPoint(x: 0, y: rect.size.height))

UIColor.green.setFill()

path.fill()

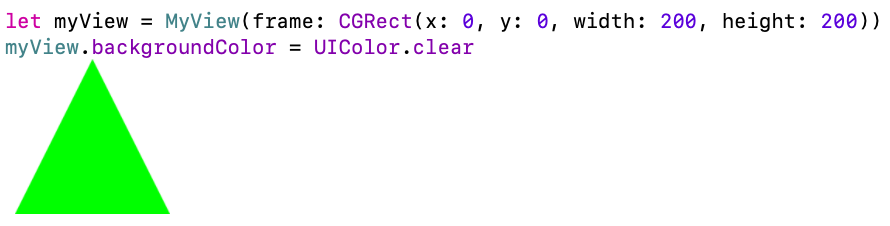
}

}

let myView = MyView(frame: CGRect(x: 0, y: 0, width: 200, height: 200))

myView.backgroundColor = UIColor.clear

* 运行结果：



1. **代码版hello world（在视图控制器中加入代码）**
   1. 代码中生成label（outlet）和button；
   2. 将label和button加入根view中；
   3. button添加像self（target）发射action（selector）的操作；

实现clicked响应代码（selector）

* 程序代码：

class TriangleView: UIView{

override func draw(\_ rect: CGRect) {

let path = UIBezierPath()

path.move(to: CGPoint(x: rect.size.width/2, y: 0))

path.addLine(to: CGPoint(x: rect.size.width, y: rect.size.height))

path.addLine(to: CGPoint(x: 0, y: rect.size.height))

UIColor.green.setFill()

path.fill()

}

}

class MyViewController : UIViewController {

var label: UILabel!

var triangle:TriangleView!

@IBAction func clicked(){

print("I am Clicked")

label?.text = "I am Clickd!!!"

UIView.animate(withDuration: 1) {

self.triangle.center = CGPoint(x: 200, y: 0)

}

}

override func loadView() {

view = UIView(frame: CGRect(x: 0, y: 0, width: 400, height: 800))

view.backgroundColor = UIColor.red

label = UILabel(frame: CGRect(x: 100, y: 20, width: 200, height: 50))

label.text = "Hello World!"

label.textColor = UIColor.white

label.backgroundColor = UIColor.blue

label.textAlignment = .center

view.addSubview(label)

let url = URL(string: "https://ss0.bdstatic.com/70cFuHSh\_Q1YnxGkpoWK1HF6hhy/it/u=4198824076,697431934&fm=27&gp=0.jpg")!

let data = try! Data(contentsOf: url)

let image = UIImage(data: data)

let imageView = UIImageView(frame: CGRect(x: 0, y: 100, width: 400, height: 400))

imageView.image = image

view.addSubview(imageView)

let button = UIButton(frame: CGRect(x: 100, y: 600, width: 200, height: 50))

button.setTitle("Click Me", for: .normal)

button.addTarget(self, action: #selector(MyViewController.clicked), for: UIControlEvents.touchUpInside)

view.addSubview(button)

triangle = TriangleView(frame: CGRect(x: 100, y: 400, width: 200, height: 200))

triangle.backgroundColor = UIColor.clear

view.addSubview(triangle)

self.view = view

}

}

// Present the view controller in the Live View window

PlaygroundPage.current.liveView = MyViewController()

* 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：[git@github.com:Gzqqqqq/FourthExperimentalReport.git](mailto:git@github.com:Gzqqqqq/FourthExperimentalReport.git)

GitHub链接只有word文档，因为实验是最开始做的，没有找到原来的项目文件，所以只放了一个文档在GitHub链接上面。

正式进入iOS布局View的开发，前面都是在学习swift语言，后面就开始学习如何如开发一个iOS的app。初步接触了如何构建一个自定一view，以及如何将一个view放到我们的页面布局当中，为后边学习图形化界面打下基础。

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。