实验编号：4 **四川师大《IOS高级开发技术》实验报告 2018** 年**10**月**8**日

**计算机科学学院**2016 级 4班 实验名称： 作业四 \_

姓名：\_邹琳\_ 学号：\_2016110458\_ 指导老师：\_李贵洋\_ 实验成绩:\_\_\_\_\_

**实验\_四\_ \_\_\_\_\_作业四\_\_\_\_\_\_**

一．实验目的

1、实验目的

1. 掌握沙盒文件的操作；
2. 掌握自定制视图的定义及使用；
3. 掌握playground下可视化开发方法；

2、实验要求

1. 认真填写实验报告，要求附加部分运行界面和主要代码；
2. 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；

二. 实验内容

**作业1（文件、自定制视图、代码版helloworld）**

1. 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。
2. 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；
3. 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；
   4. 实现clicked响应代码（selector）

三．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）

**作业1（文件、自定制视图、代码版helloworld）**

1. 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：

//: Playground - noun: a place where people can play

import UIKit

import PlaygroundSupport

let manager = FileManager.default

let document = manager.urls(for:.documentDirectory,in: .userDomainMask).first?.path

let file = document?.appending("/img")

print(document!)

if manager.fileExists(atPath: file!) { //若该文件夹存在

let image = file?.appending("/\*\*.png") //图片文件

if manager.fileExists(atPath: image!) { //若该文件存在，则显示到界面上

let data = try Data(contentsOf: URL(fileURLWithPath: image!))

let img = UIImage(data: data)

let imageView = UIImageView(image: img)

imageView.frame = CGRect(x: 0, y: 100, width: 300, height: 200)

// PlaygroundPage.current.liveView(imageView)

} else { //若不存在，则从网络下载一个图片并保存为图片文件

let url = URL(string: "https://ss3.bdstatic.com/70cFv8Sh\_Q1YnxGkpoWK1HF6hhy/it/u=280692610,4239256719&fm=26&gp=0.jpg")

let data = try Data(contentsOf: url!)

try data.write(to: URL(fileURLWithPath: image!), options: .atomicWrite)

print("图片不存在，成功下载了新图片")

}

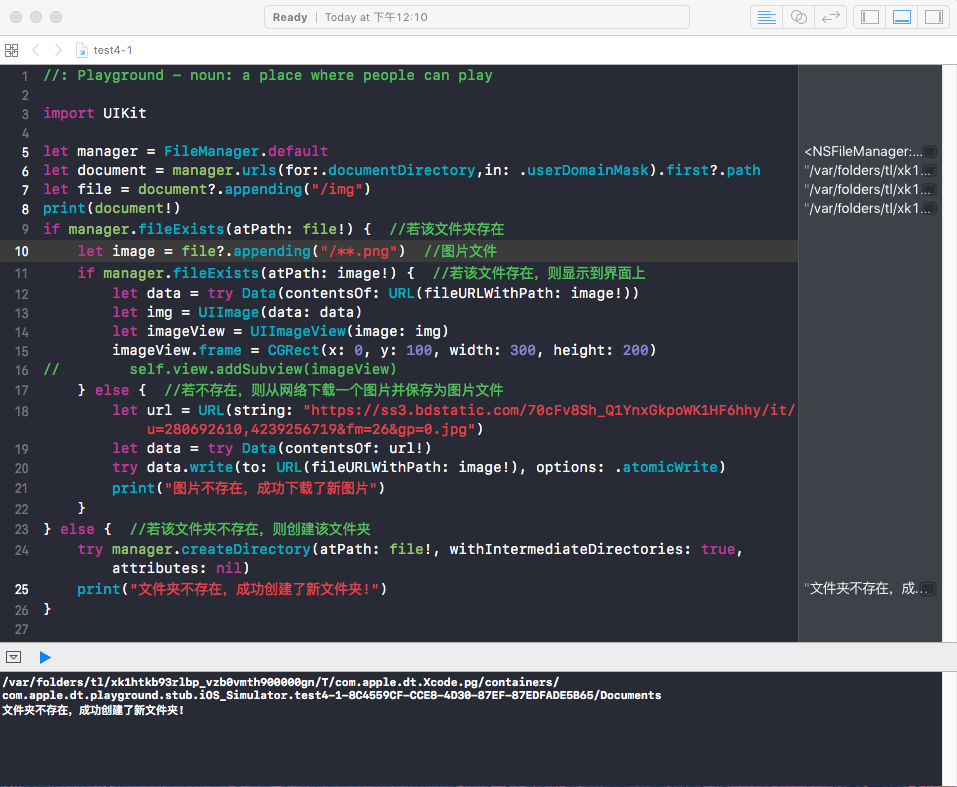
} else { //若该文件夹不存在，则创建该文件夹

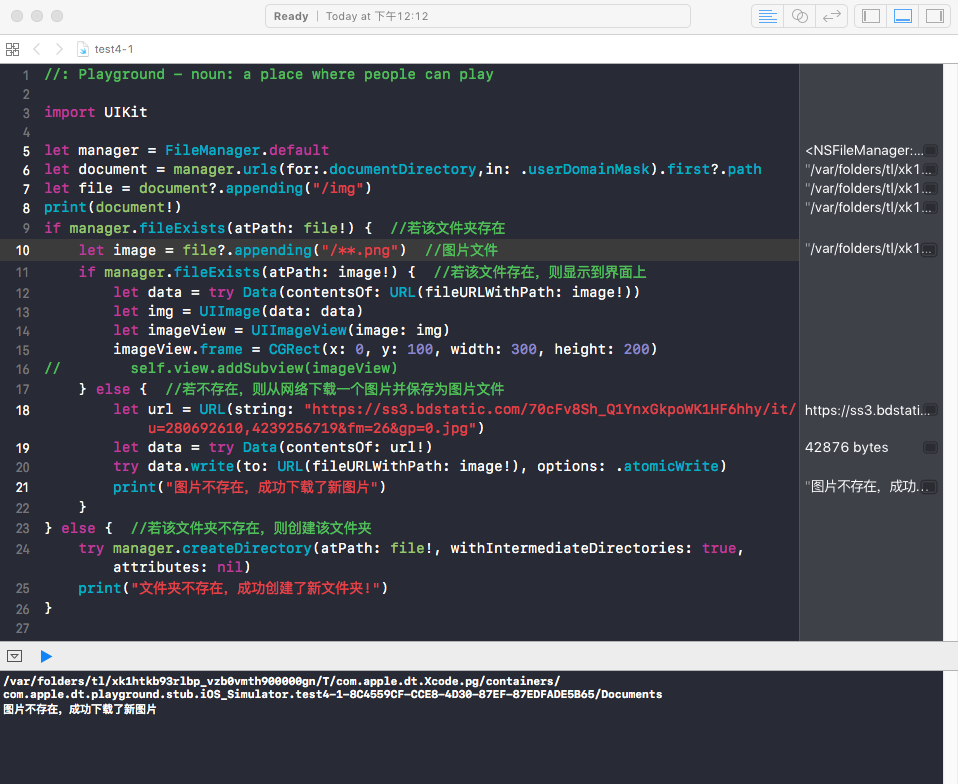
try manager.createDirectory(atPath: file!, withIntermediateDirectories: true, attributes: nil)

print("文件夹不存在，成功创建了新文件夹!")

}

运行截图：



****

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：

//: Playground - noun: a place where people can play

import UIKit

import PlaygroundSupport

class TrangleView:UIView{ //从view中派生出自定义的View

override func draw(\_ rect:CGRect){ //重写父类的draw方法

// let firstpath = UIBezierPath(rect: rect)

//创建一个UIBezierPath变量，用于画三角形

let path = UIBezierPath()

path.move(to: CGPoint(x: rect.size.width/2, y: 0)) //设置线段的起始点

path.addLine(to: CGPoint(x: rect.size.width, y: rect.size.height))//添加一条线段

path.addLine(to: CGPoint(x: 0, y: rect.size.height))//添加一条线段

UIColor.black.setStroke()

path.stroke()

UIColor.green.setFill()

path.fill()

//画椭圆和圆

let path3 = UIBezierPath(ovalIn:rect)

UIColor.blue.setStroke()

path3.stroke()

UIColor.yellow.setFill()

path3.fill()

}

}

class Controller:UIViewController{

override func viewDidLoad() {

super.viewDidLoad()

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

view.backgroundColor = UIColor.gray

//画椭圆

let ovel = TrangleView(frame:CGRect(x:50,y:100,width:60,height:130))

ovel.backgroundColor = UIColor.clear

self.view.addSubview(ovel)

//画椭圆

let circle = TrangleView(frame:CGRect(x:210,y:220,width:150,height:80))

circle.backgroundColor = UIColor.clear

self.view.addSubview(circle)

//画圆

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

triangle.backgroundColor = UIColor.clear

view.addSubview(triangle)

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

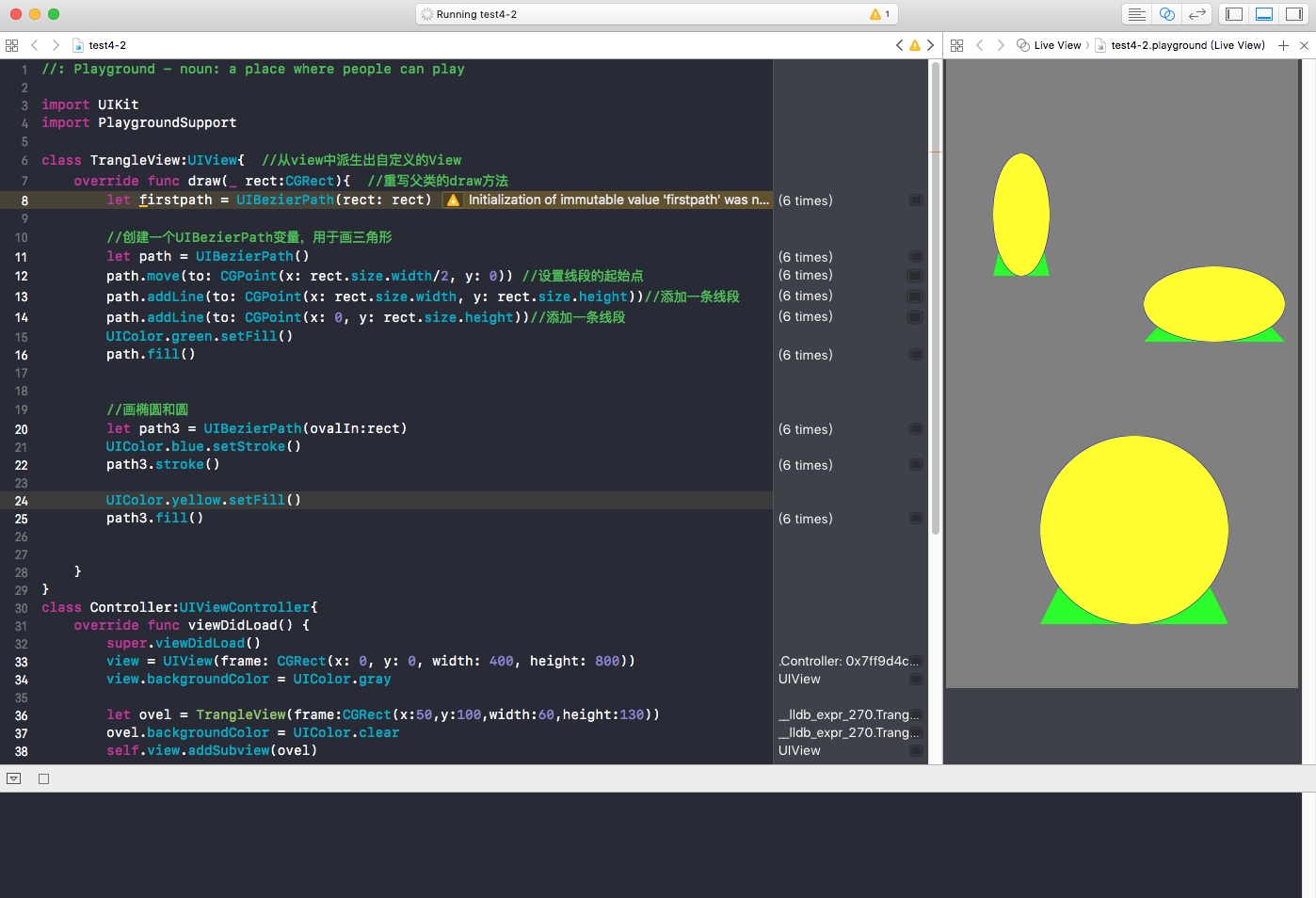
}

}

let ct = Controller()

PlaygroundPage.current.liveView = ct

运行截图：



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；
   4. 实现clicked响应代码（selector）

程序代码：

//: Playground - noun: a place where people can play

import UIKit

import PlaygroundSupport

class TriangleView: UIView{

override func draw(\_ rect: CGRect) {

let path = UIBezierPath()

let firstpath = UIBezierPath(rect: rect)

//创建一个UIBezierPath变量，UIBezierPath可创建基于矢量的路径，常用来绘图

UIColor.red.setStroke() //红色边界

firstpath.stroke() //将视图边界弄成红色

UIColor.yellow.setFill()//黄色填充

firstpath.fill()//将TrangleView将视图填充为黄色

//画椭圆和圆

let secondpath = UIBezierPath(ovalIn:rect)

UIColor.blue.setStroke()

secondpath.stroke()

UIColor.green.setFill()

secondpath.fill()

//画三角形

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))

path.close()

UIColor.black.setStroke()

path.stroke()

UIColor.cyan.setFill()

path.fill()

}

}

class Controller:UIViewController{

var label:UILabel!

var triangle:TriangleView!

@IBAction func clicked(){ //生成点击事件

print("i am clicked")

label?.text = "i am clicked!!"

UIView.animate(withDuration: 1){

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

}

}

override func loadView() { //加载视图

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

view.backgroundColor = UIColor.lightGray

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

label.backgroundColor = UIColor.blue

label.textColor = UIColor.white

label.textAlignment = .center

label.text = "hello,word"

view.addSubview(label)//将从网上下载的图片作为背景图片

let url = URL(string: "https://ss3.bdstatic.com/70cFv8Sh\_Q1YnxGkpoWK1HF6hhy/it/u=280692610,4239256719&fm=26&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)

//添加按钮

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

button.backgroundColor = UIColor.brown

button.setTitle("clicked me", for: .normal)

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

view.addSubview(button)

//画图

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

triangle.backgroundColor = UIColor.clear

view.addSubview(triangle)

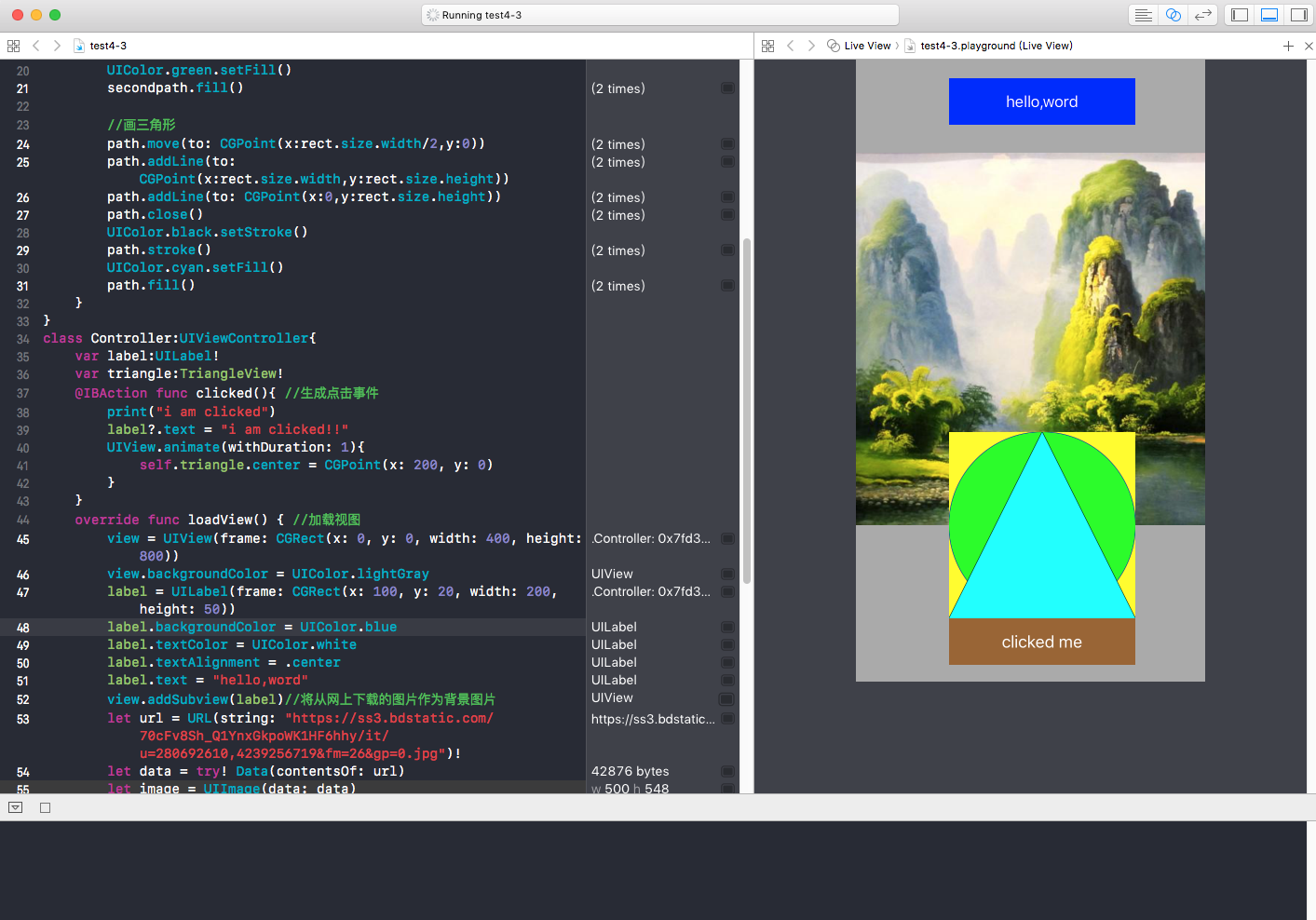
}

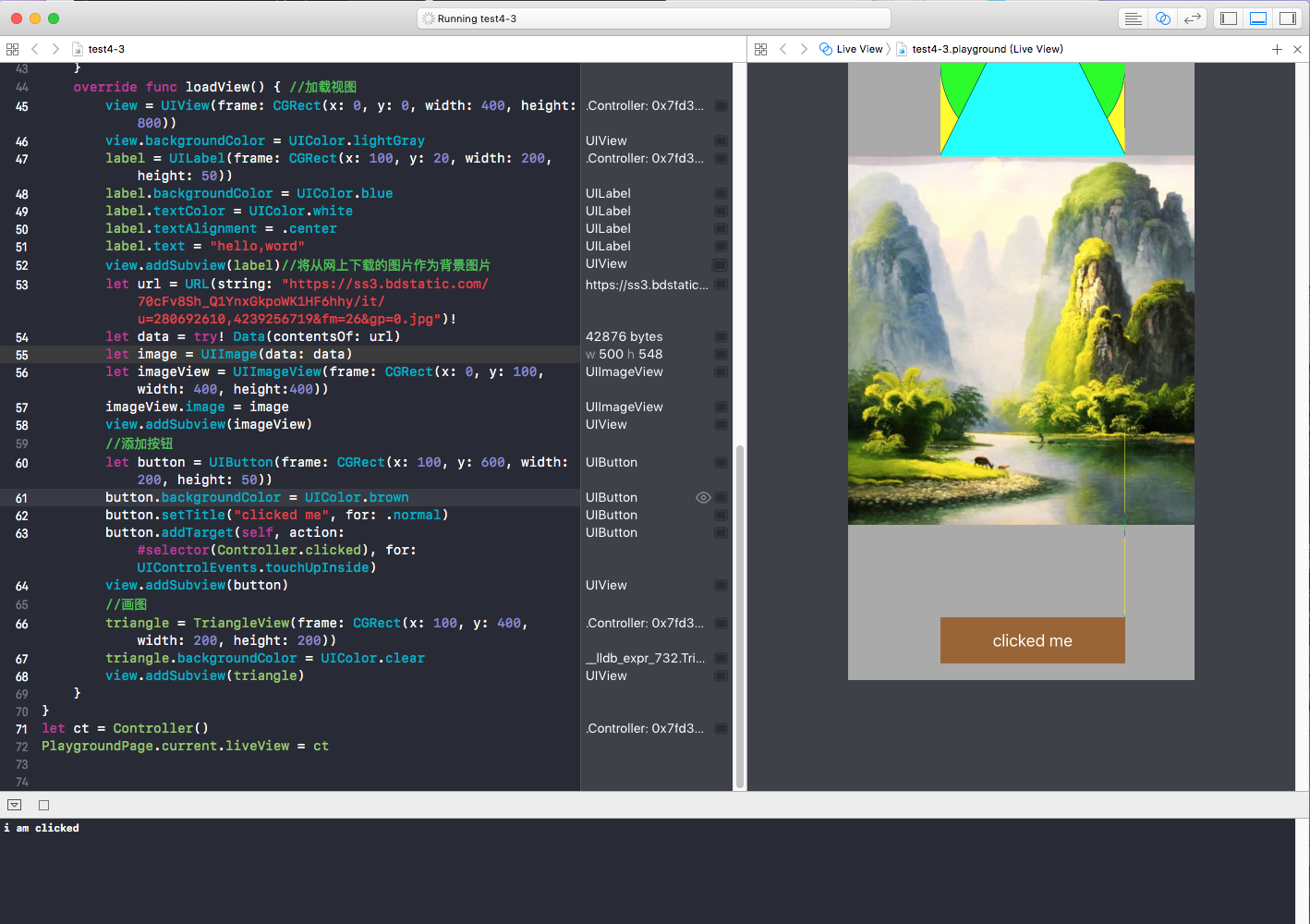
}

let ct = Controller()

PlaygroundPage.current.liveView = ct

运行截图：





四．实验结果的分析与评价（该部分如不够填写，请另加附页）

掌握了沙盒系统下对文件夹及其目录下的文件的新建，访问与读写，学会了用视图画基本的几何图形，以及通过视图控制器实现对事件的响应。在实验过程中，一开始对视图进行操作时不能得到结果，调试后发现忘记了导入import PlaygroundSupport所致。

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。