实验编号： 4 **四川师大《IOS》实验报告 2019** 年 **11** 月 **20** 日

### **计算机科学学院** 2017 级 4 班 实验名称： 自定制视图、文件、Playground可视化

姓名：\_\_李晋\_ 学号：\_\_2017110517\_ 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 UIKit

////实验5

//判断文件或文件夹是否存在

//判断图片文件是否存在，不存在则从网络上下载图片到该文件里

let fileManager = FileManager.default

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

    docPath.appendPathComponent("zzp")

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

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

    }

    docPath.appendPathComponent("\_image1.jpg")

    print(docPath.path)

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

        let url = URL(string: "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1541742161361&di=ef680a1a504bb154117c20459f710ba5&imgtype=0&src=http%3A%2F%2Fpic1.win4000.com%2Fmobile%2F2017-11-15%2F5a0c2d2313c90.jpg")

        let data = try? Data(contentsOf: url!)

        try? data?.write(to: docPath)

    }

}

* 运行结果：

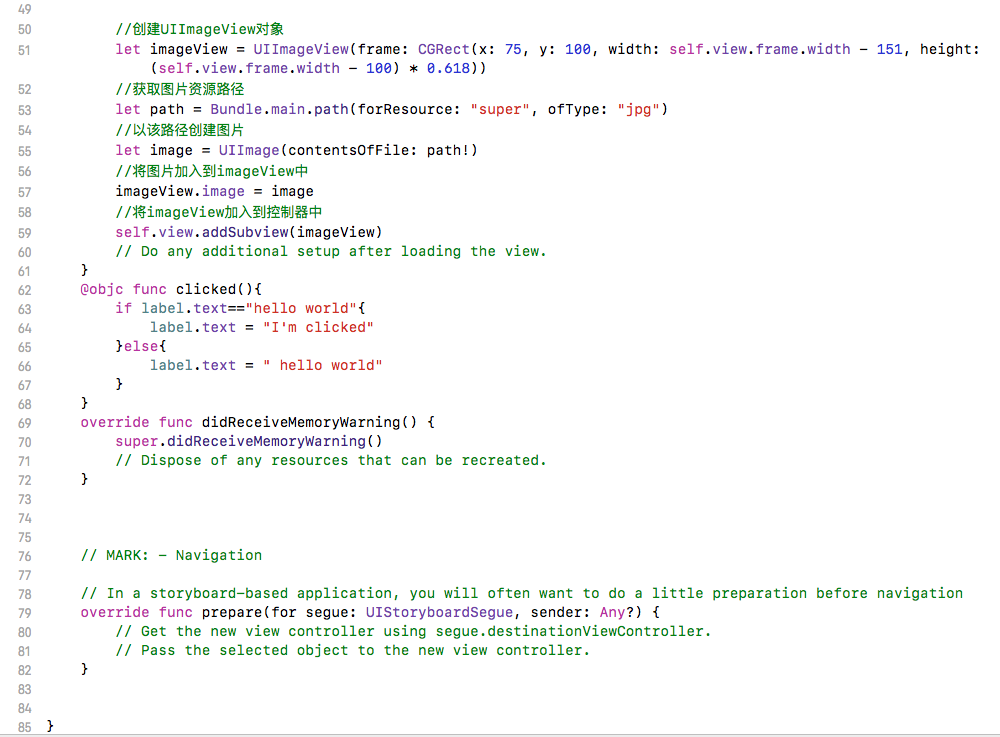


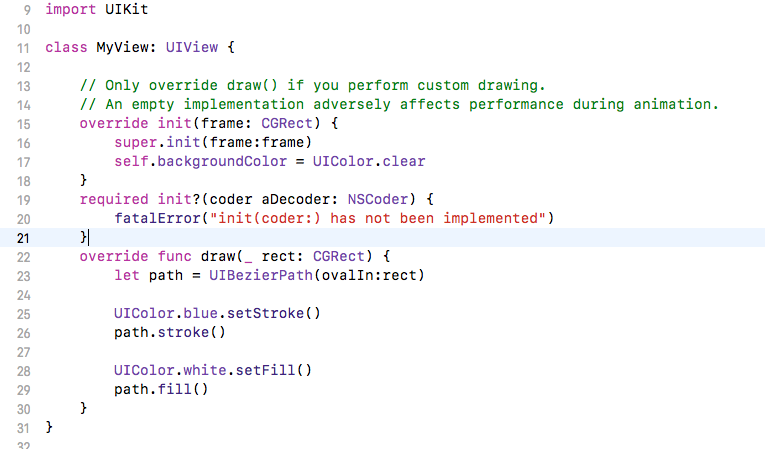
1. **自定制视图**
   1. 从UIView中派生一个自定制的View；
   2. 绘制一个椭圆（或则自己喜欢的任何图形）；
   3. 新建视图对象并进行显示；

* 程序代码：

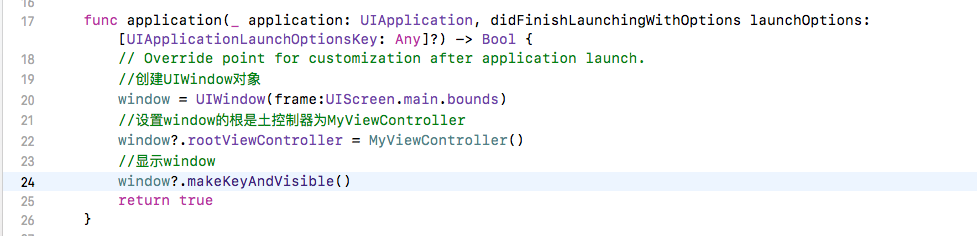
MyViewController代码：





MyView代码：  


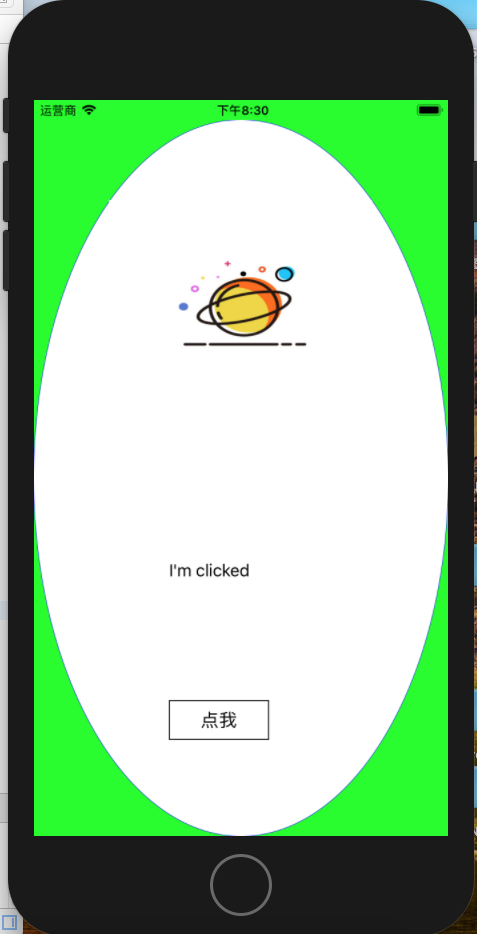
AppDelegate代码：



* 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：

import UIKit

////实验5

//判断文件或文件夹是否存在

//判断图片文件是否存在，不存在则从网络上下载图片到该文件里

let fileManager = FileManager.default

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

    docPath.appendPathComponent("zzp")

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

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

    }

    docPath.appendPathComponent("\_image1.jpg")

    print(docPath.path)

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

        let url = URL(string: "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1541742161361&di=ef680a1a504bb154117c20459f710ba5&imgtype=0&src=http%3A%2F%2Fpic1.win4000.com%2Fmobile%2F2017-11-15%2F5a0c2d2313c90.jpg")

        let data = try? Data(contentsOf: url!)

        try? data?.write(to: docPath)

    }

}

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

import UIKit

////实验5

//判断文件或文件夹是否存在

//判断图片文件是否存在，不存在则从网络上下载图片到该文件里

let fileManager = FileManager.default

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

    docPath.appendPathComponent("zzp")

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

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

    }

    docPath.appendPathComponent("\_image1.jpg")

    print(docPath.path)

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

        let url = URL(string: "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1541742161361&di=ef680a1a504bb154117c20459f710ba5&imgtype=0&src=http%3A%2F%2Fpic1.win4000.com%2Fmobile%2F2017-11-15%2F5a0c2d2313c90.jpg")

        let data = try? Data(contentsOf: url!)

        try? data?.write(to: docPath)

    }

}

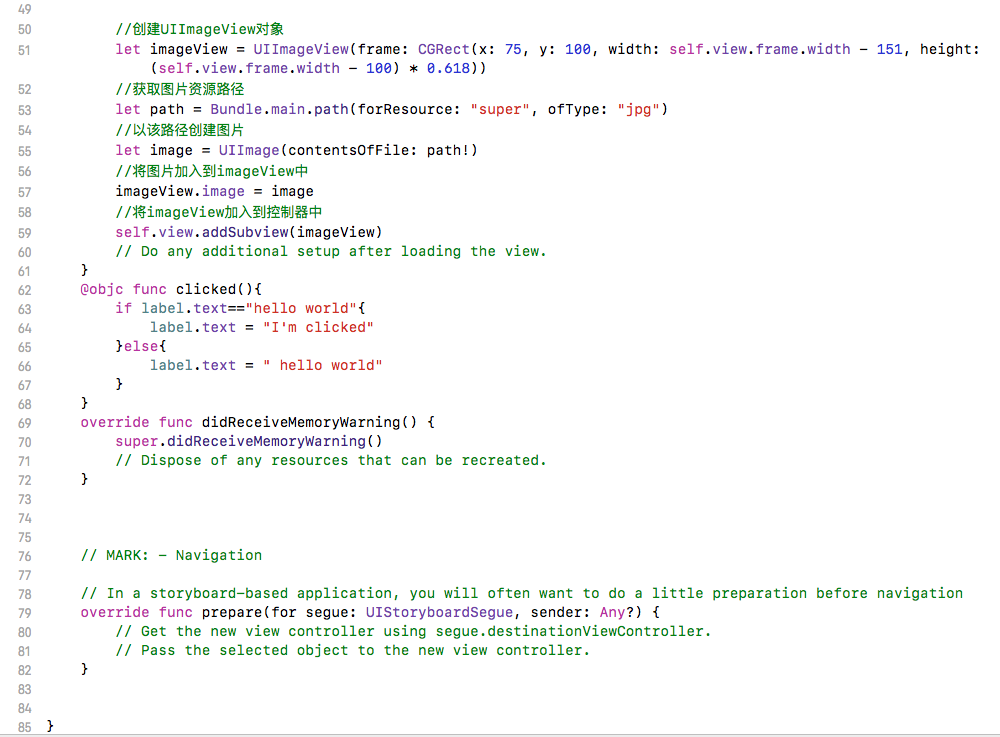
* 运行结果：

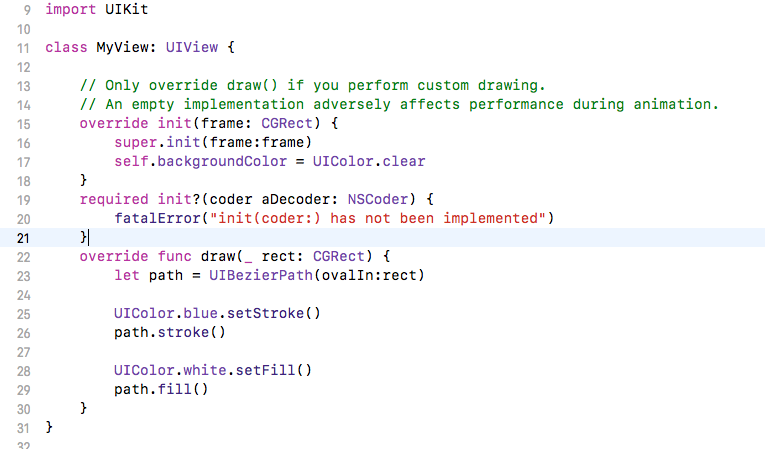
1. **自定制视图**
   1. 从UIView中派生一个自定制的View；
   2. 绘制一个椭圆（或则自己喜欢的任何图形）；
   3. 新建视图对象并进行显示；

* 程序代码：

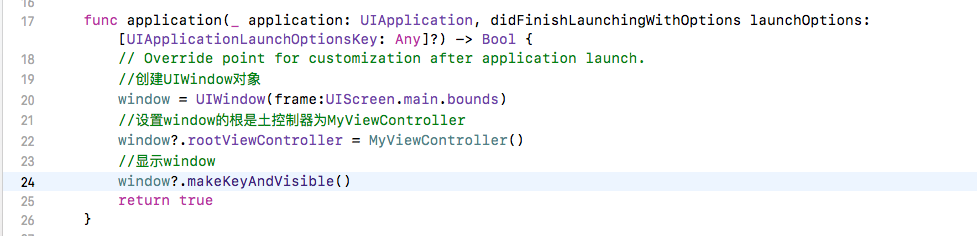
MyViewController代码：





MyView代码：  


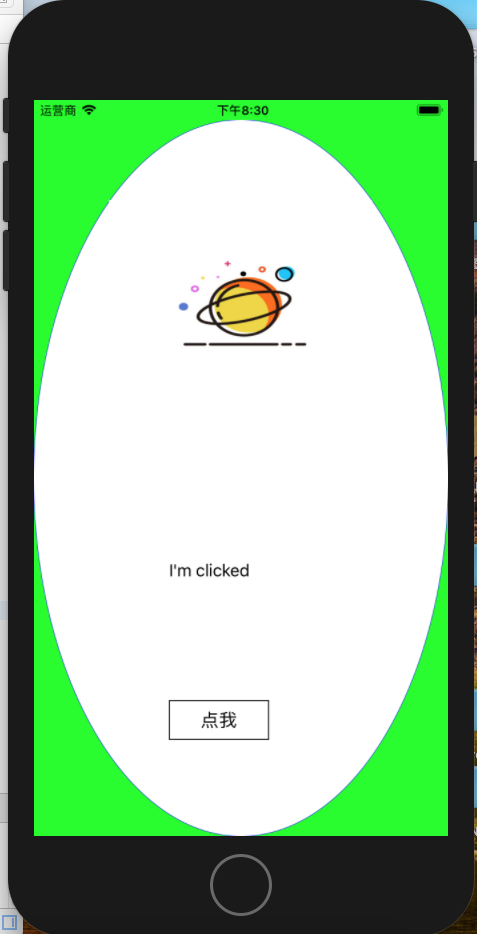
AppDelegate代码：



* 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）
   1. 了解视图UIView的实例，自定义绘制一个图形，改变其颜色、位置等属性。
   2. 对代理文件AppDelegate.swift有了更佳深刻的了解。
   3. 掌握UIButton和UILabel控件的使用

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。