实验编号： 6 **四川师大《IOS》实验报告 2018** 年 **10** 月 **17** 日

### **计算机科学学院** 2016 级 4 班 实验名称： Auto Layout \_

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

**实验 六** Auto Layout

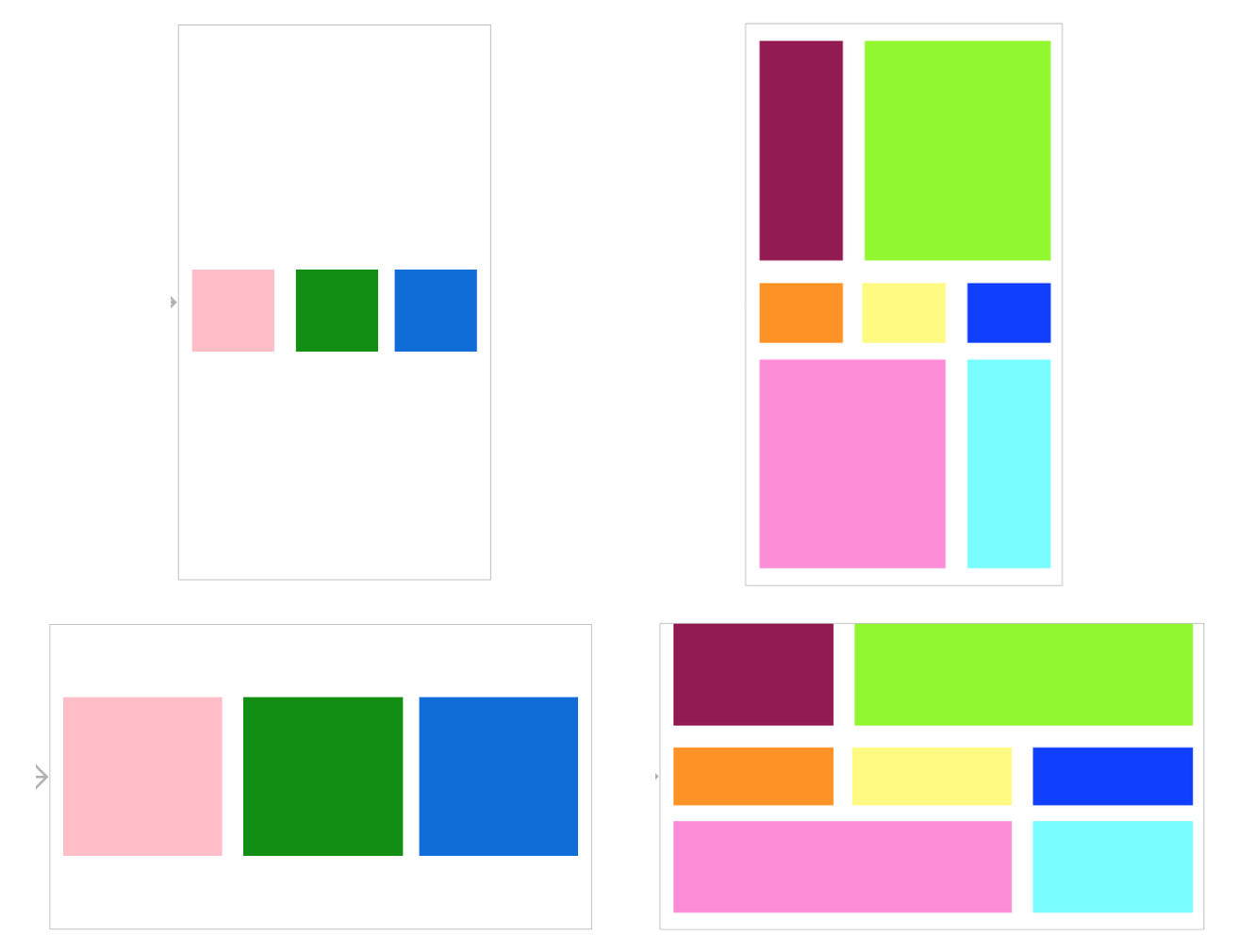
1. 实验目的及要求

### (1) 掌握UIKit基本结构

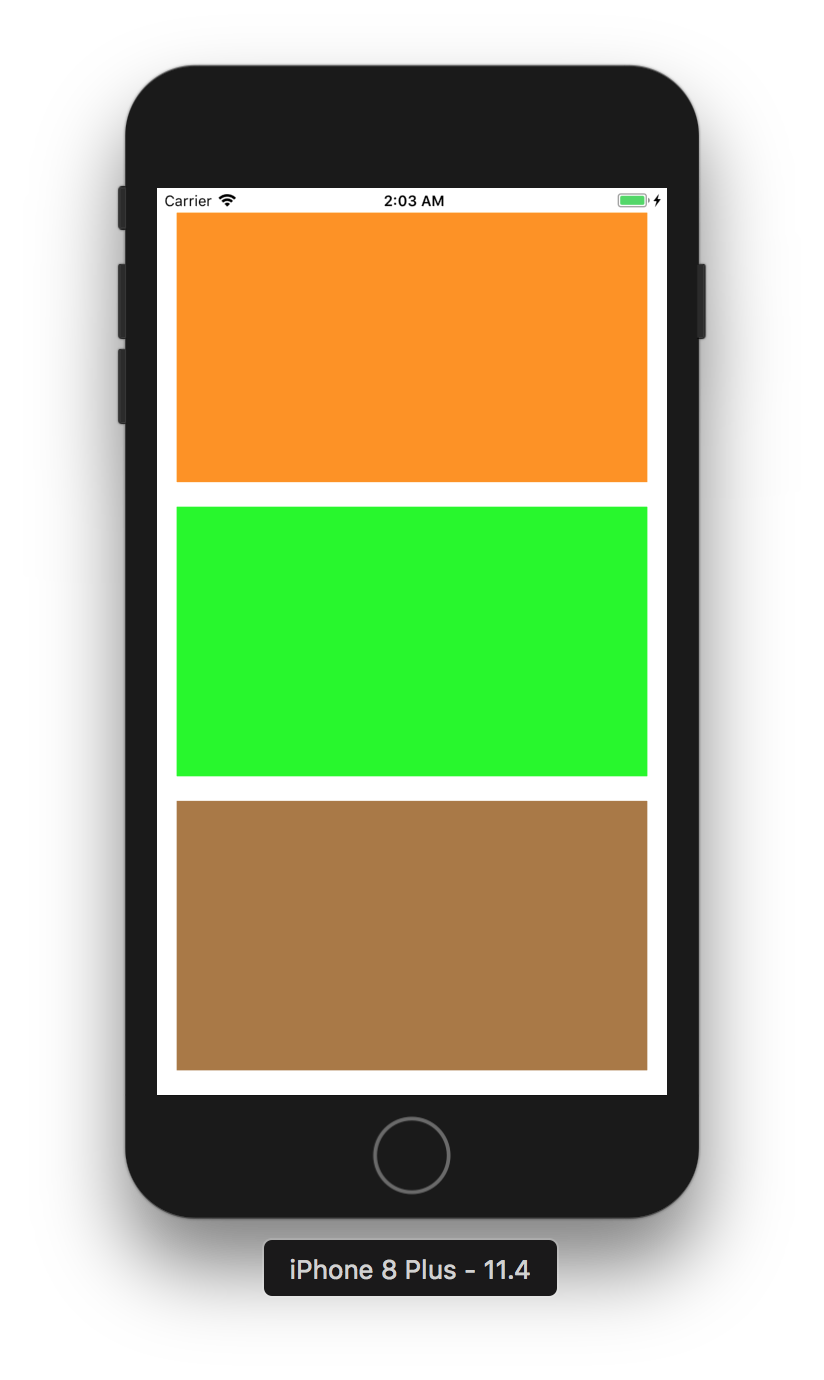
### (2) 掌握UIKit界面构造技术

### (3) 掌握自动布局方法

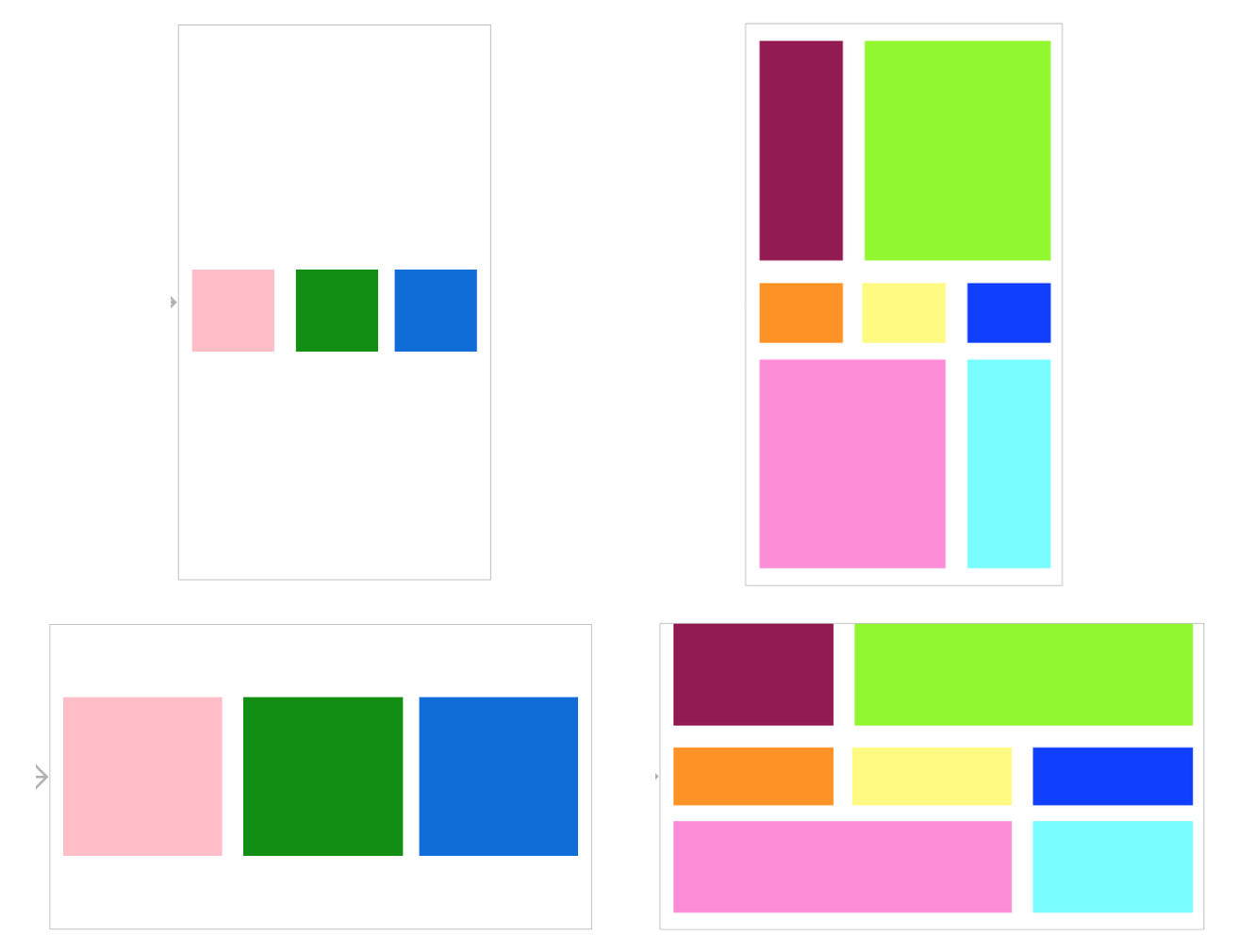
1. 实验内容
2. 采用Autolayout 中的 stackview 和约束实现以下布局；



1. 采用纯代码添加试图以及相应约束的方式实现下述布局；

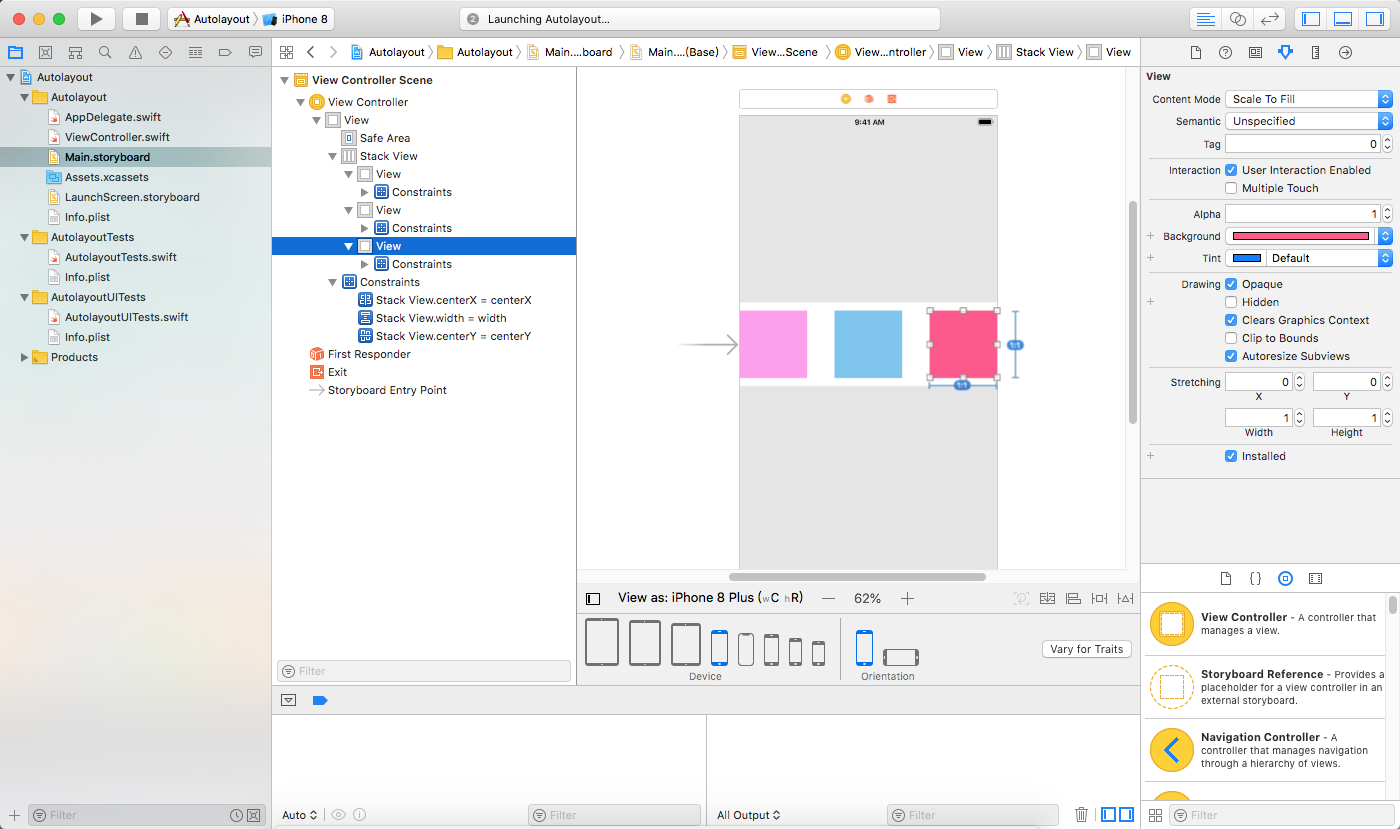
 

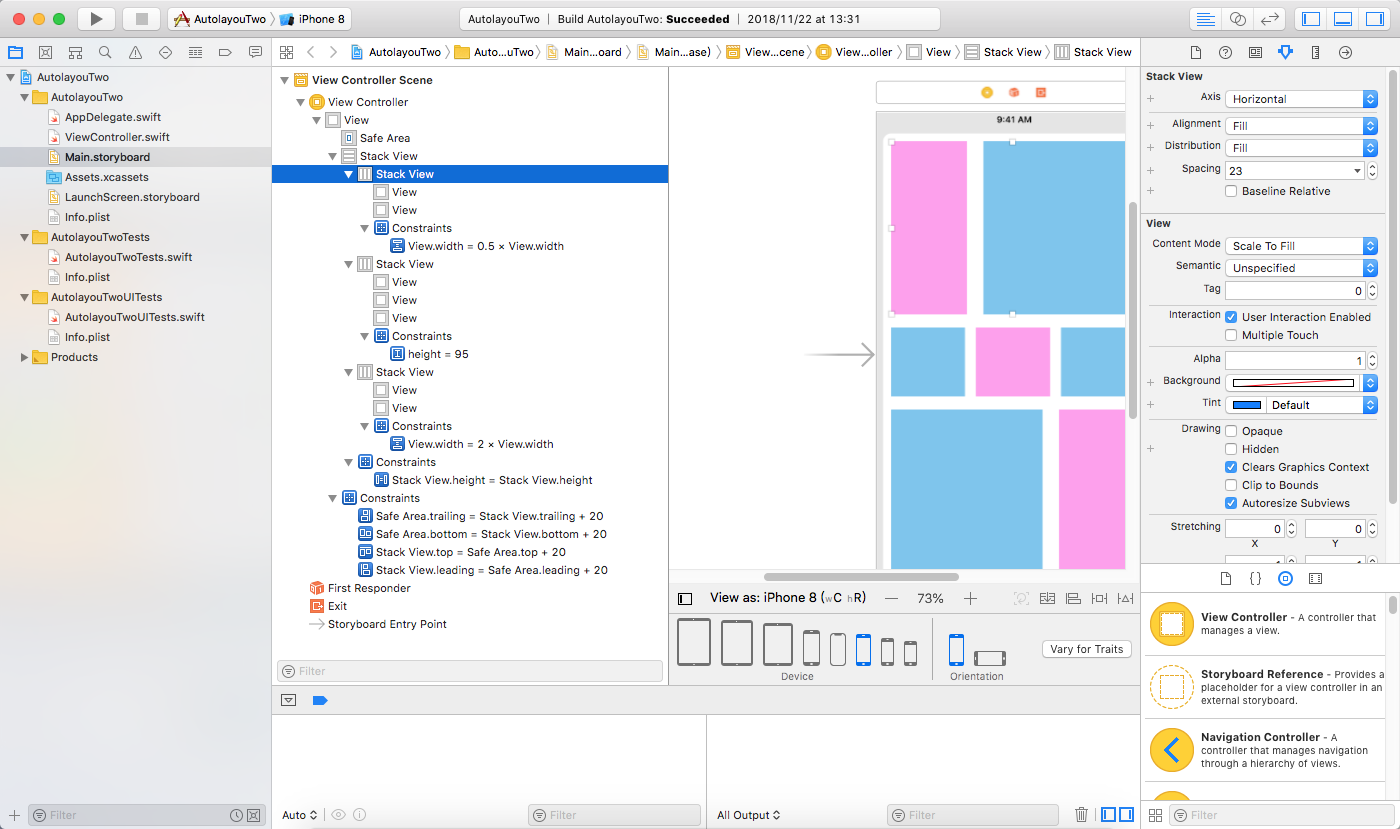
1. 实验主要流程、基本操作或核心代码、算法片段（该部分如不够填写，请另加附页）
2. 采用Autolayout 中的 stackview 和约束实现以下布局；



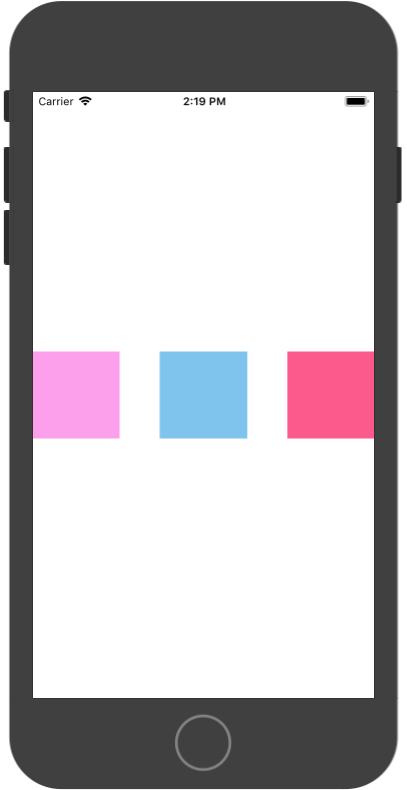
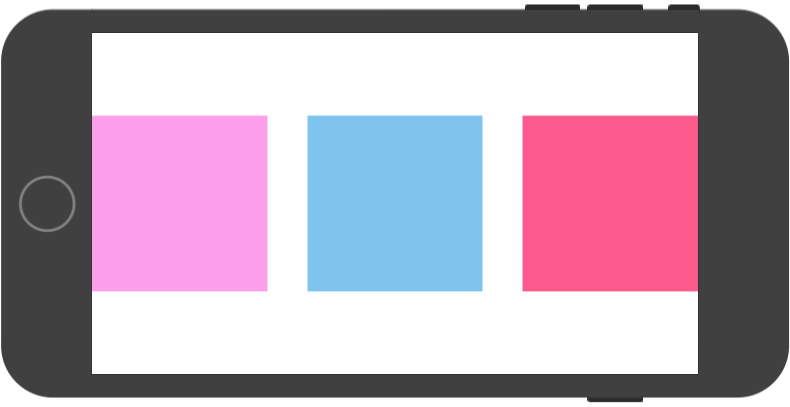
* 程序代码：

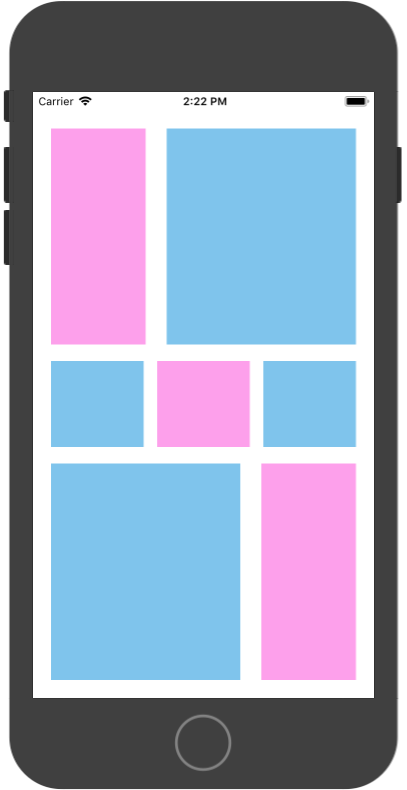
没有代码主要是在页面中设置了约束

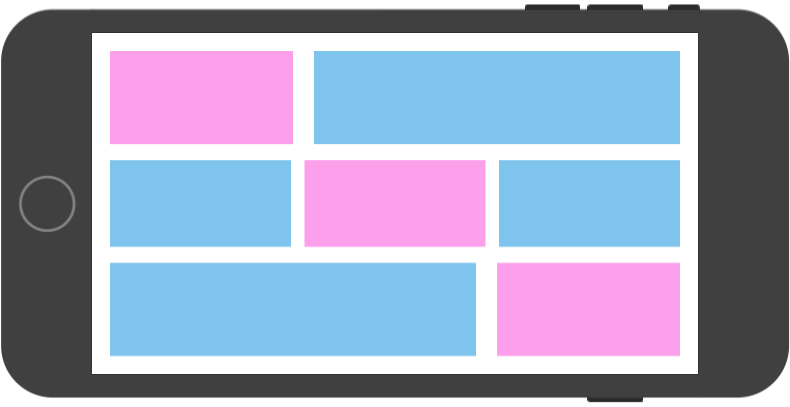




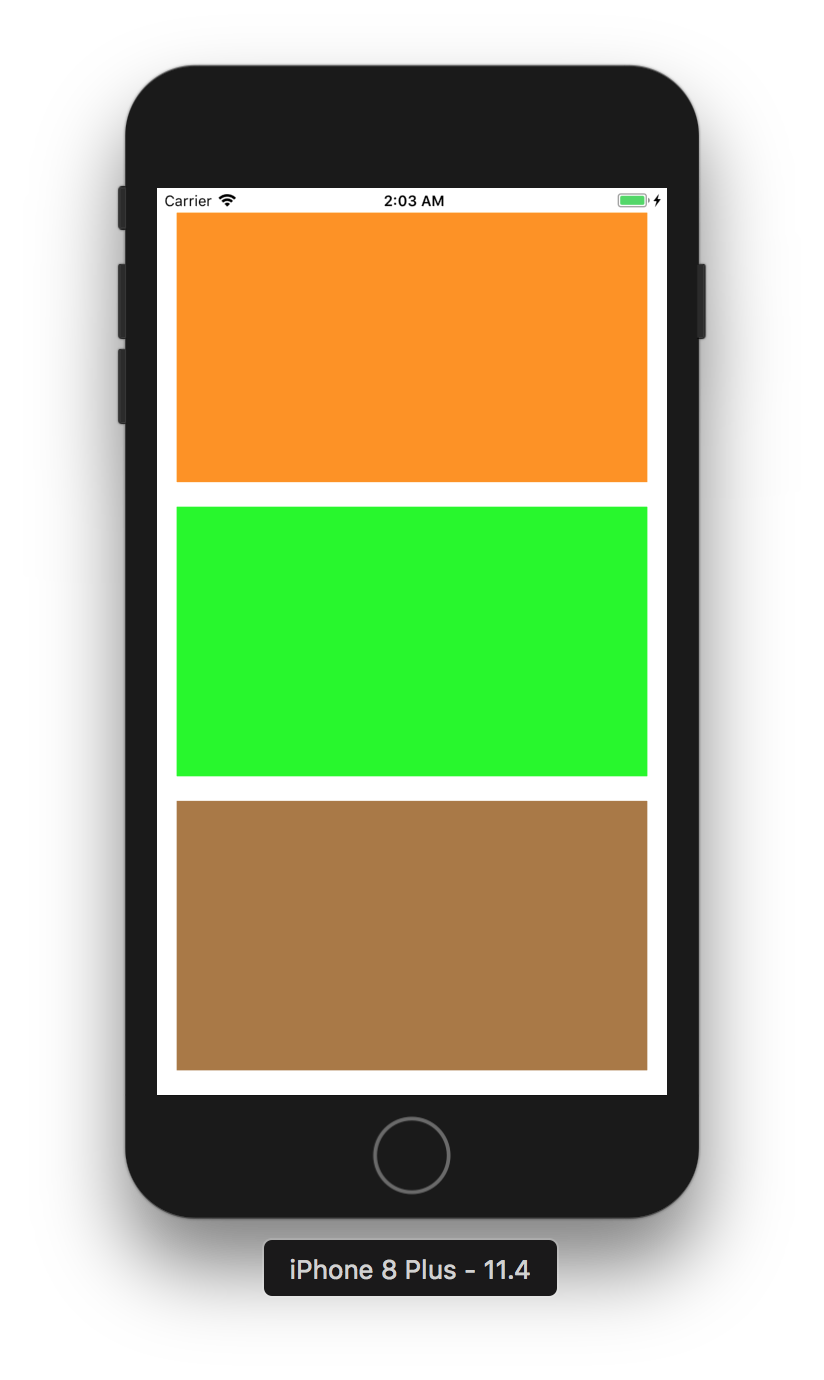
* 运行结果：





1. 采用纯代码添加试图以及相应约束的方式实现下述布局；

* 程序代码：

//

// ViewController.swift

// AutolayoutThree

//

// Created by jiang on 2018/11/10.

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

//

import UIKit

class ViewController: UIViewController {

override func viewDidLoad() {

super.viewDidLoad()

let redView=UIView()

redView.backgroundColor = UIColor.red

let greenView=UIView()

greenView.backgroundColor = UIColor.green

let yellowView=UIView()

yellowView.backgroundColor = UIColor.yellow

let stackView=UIStackView(arrangedSubviews: [redView,greenView,yellowView])

stackView.frame=CGRect(x: 0, y: 0, width: 200, height: 400)

stackView.axis = .vertical

stackView.alignment = .fill

stackView.distribution = .fillEqually

stackView.spacing=20

view.addSubview(stackView)

stackView.translatesAutoresizingMaskIntoConstraints=false

stackView.leadingAnchor.constraint(equalTo: view.leadingAnchor, constant: 20).isActive=true

stackView.topAnchor.constraint(equalTo: view.topAnchor, constant: 20).isActive=true

stackView.trailingAnchor.constraint(equalTo: view.trailingAnchor, constant: -20).isActive=true

stackView.bottomAnchor.constraint(equalTo: view.bottomAnchor, constant: 20).isActive=true

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

}

override func didReceiveMemoryWarning() {

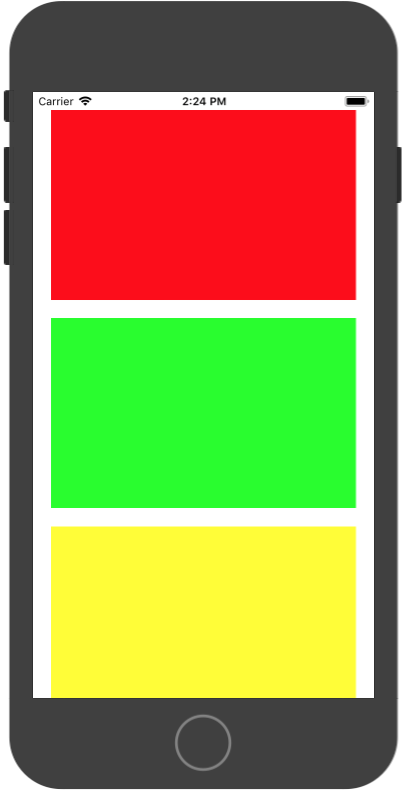
super.didReceiveMemoryWarning()

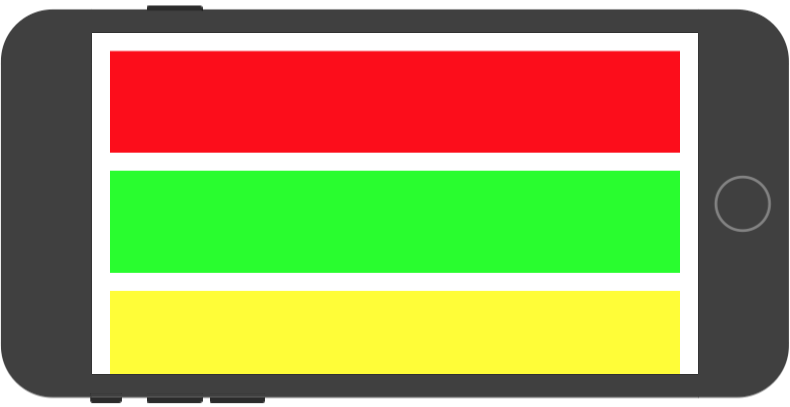
// Dispose of any resources that can be recreated.

}

}

* 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)

这次的实验是自动布局，一个是直接在storyboard上面写约束，一个是用代码来写约束。

因为有自动布局所以不管应用是横屏还是竖屏它上面的元素都不会乱的。有时候需要在横屏竖屏上面的单独设置不同的约束。对于一个应用来说的话约束是很重要的。

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。