实验编号： 10 **四川师大《IOS》实验报告 2018** 年 **11** 月 **14** 日

### **计算机科学学院** 2016 级 4 班 实验名称： 视图动画 \_

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

**实验 十 \_\_\_**视图动画**\_\_\_\_\_**

1. 实验目的及要求
2. 掌握视图基本动画的原理和使用；
3. 掌握动力学动画的原理和使用；
4. 实验要求
5. 认真填写实验报告，要求附加部分运行界面和主要代码；
6. 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；
7. 实验内容
8. 基于UIView.animation实现如下动画效果：
   1. 改变视图的位置
   2. 改变视图的大小
   3. 改变视图的transform
9. 基于UIView.transition实现如下动画效果：
   1. 改变视图的背景颜色；
   2. 切换两个子视图，观察切换后视图层次的变化情况；
10. 基于UIDynamicAnimtor实现简单的力学动画：
    1. 可以掉落方块；
    2. 有碰撞；
    3. 可以反弹；
11. 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）
12. 基于UIView.animation实现如下动画效果：
    1. 改变视图的位置
    2. 改变视图的大小
    3. 改变视图的transform
       * 程序代码：

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var myView: UIView!

    override func viewDidLoad() {

        super.viewDidLoad()

    }

    @IBAction func btnClicked(\_ sender: Any) {

        UIView.animate(withDuration:2,delay:1,options:[.curveEaseInOut,.autoreverse,.repeat],animations:{

            self.myView.alpha = 0

            self.myView.center.x = self.view.bounds.width

            self.myView.backgroundColor =  colorLiteral(red: 0.6080191446, green: 0.9232503658, blue: 1, alpha: 1)

            self.myView.transform = CGAffineTransform(rotationAngle: CGFloat(Double.pi)).scaledBy(x: 0.5, y: 0.5)

        }) {(finished) in

            if finished {

                self.myView.removeFromSuperview()

            }

        }

    }

    override func didReceiveMemoryWarning() {

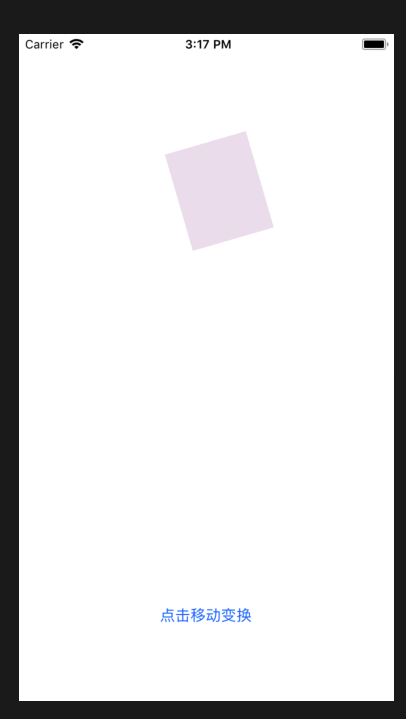
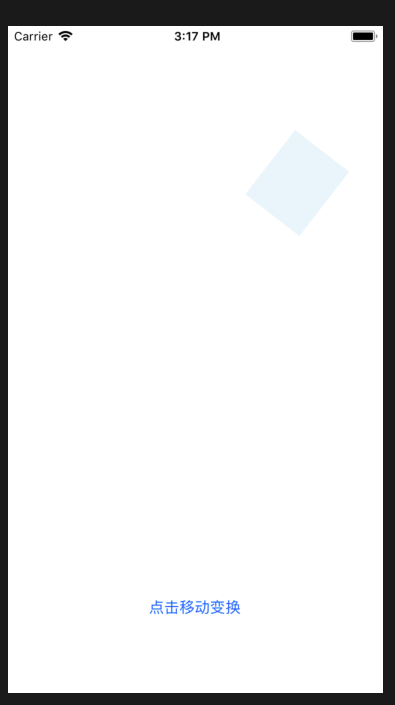
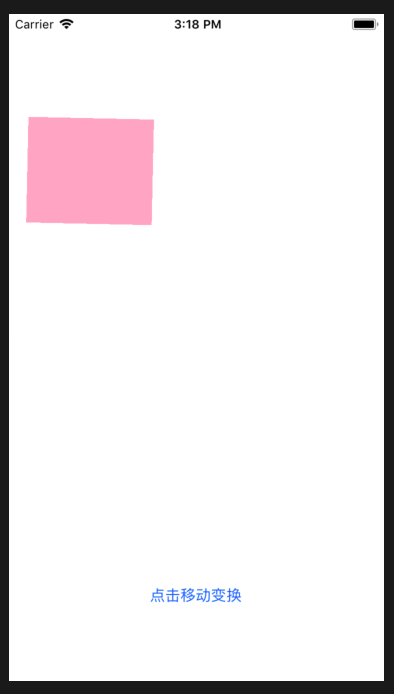
        super.didReceiveMemoryWarning()

        // Dispose of any resources that can be recreated.

    }

}

* + - 运行结果：

1. 基于UIView.transition实现如下动画效果：
   1. 改变视图的背景颜色；
   2. 切换两个子视图，观察切换后视图层次的变化情况；
      * 程序代码：

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var myView1: UIView!

    @IBOutlet weak var myView2: UIView!

    override func viewDidLoad() {

        super.viewDidLoad()

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

    }

    @IBAction func btnClicked1(\_ sender: Any) {

        UIView.transition(with: myView1, duration: 2, options: .transitionCurlUp, animations: {

            self.myView1.backgroundColor =  colorLiteral(red: 1, green: 0.694814277, blue: 0.7656143615, alpha: 1)

        }, completion: nil)

    }

    @IBAction func btnClicked2(\_ sender: Any) {

        let imageView = UIImageView(frame: CGRect(x: 70, y: 300, width: 235, height: 100))

        imageView.image = UIImage(named: "iamge")

        UIView.transition(from: myView2, to: imageView, duration: 2, options: .transitionFlipFromRight, completion: nil)

    }

        override func didReceiveMemoryWarning() {

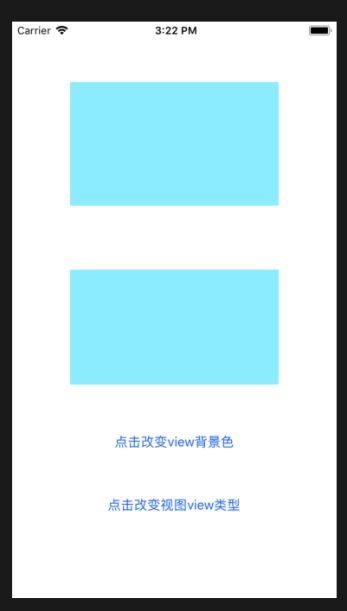
        super.didReceiveMemoryWarning()

        // Dispose of any resources that can be recreated.

    }

}

* + - 运行结果：

1. 基于UIDynamicAnimtor实现简单的力学动画：
   1. 可以掉落方块；
   2. 有碰撞；
   3. 可以反弹；
      * 程序代码：

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var backView: UIView!

    lazy var animator = UIDynamicAnimator(referenceView: self.backView)

    let gravity = UIGravityBehavior()

    let collision = UICollisionBehavior()

    override func viewDidLoad() {

        super.viewDidLoad()

        animator.addBehavior(gravity)

        animator.addBehavior(collision)

        collision.translatesReferenceBoundsIntoBoundary = true

    }

    @IBAction func addView(\_ sender: Any) {

        let width = Int(backView.bounds.width / 10)

        let randX = Int(arc4random() % 10) \* width

        let label = UILabel(frame: CGRect(x: randX, y: 0, width: width, height: width))

         label.backgroundColor =  colorLiteral(red: 0.6103215218, green: 0.9613727927, blue: 0.9741045833, alpha: 1)

        label.text = "A"

        label.textAlignment = .center

        backView.addSubview(label)

        gravity.addItem(label)

        collision.addItem(label)

    }

    @IBAction func deleteall(\_ sender: Any) {

        for item in self.backView.subviews {

            if item is UILabel {

                item.removeFromSuperview()

                gravity.removeItem(item)

                collision.removeItem(item)

            }

        }

    }

    @IBAction func up(\_ sender: Any) {

        gravity.gravityDirection = CGVector(dx: 0, dy: -1)

    }

    @IBAction func down(\_ sender: Any) {

        gravity.gravityDirection = CGVector(dx: 0, dy: 1)

    }

    @IBAction func left(\_ sender: Any) {

        gravity.gravityDirection = CGVector(dx: -1, dy: 0)

    }

    @IBAction func right(\_ sender: Any) {

        gravity.gravityDirection = CGVector(dx: 1, dy: 0)

    }

    override func didReceiveMemoryWarning() {

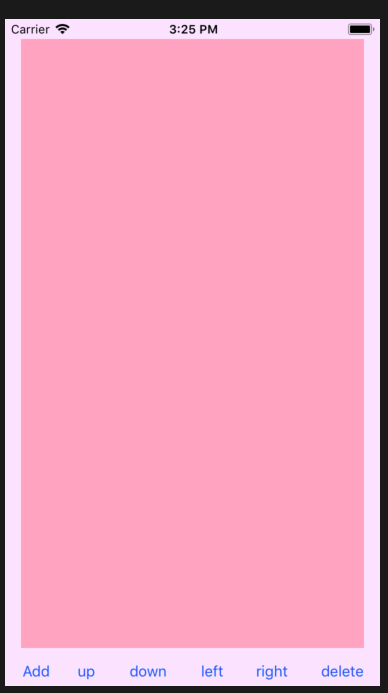
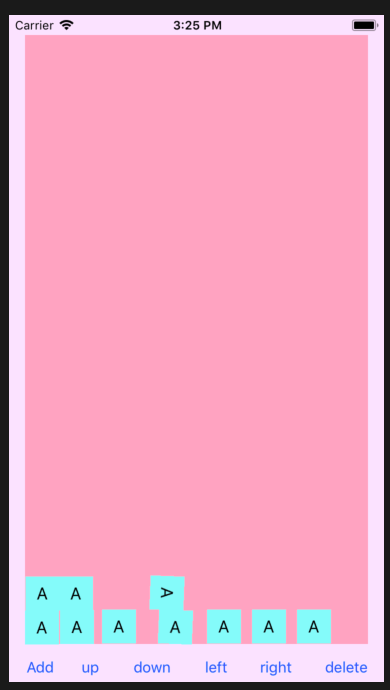
        super.didReceiveMemoryWarning()

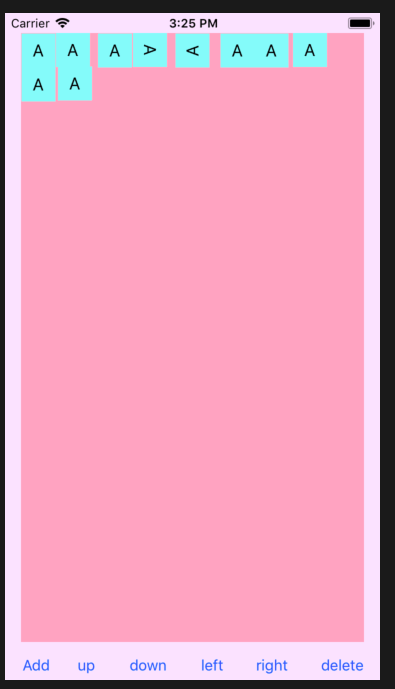
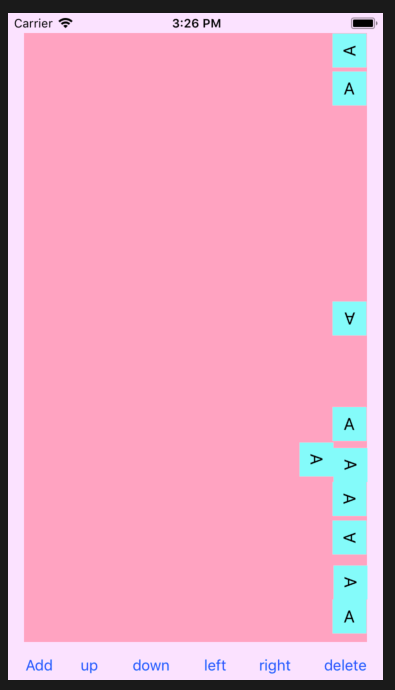
        // Dispose of any resources that can be recreated.

    }

}

* + - 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：

本次实验主要学习了三个方面的知识点：动画（包括移动、缩放、变换颜色）、翻转（改变颜色、改变view类型），以及 基于UIDynamicAnimtor的简单力学动画（实现下落、碰撞、反弹等），收获就是学习了解了，什么是动画，以及如何去使用动画，给我们的界面带来不一样的效果，也简单的了解了什么是力学动画，通过实验制作的小作业，也让我们感受了一下力学动画的实现过程。

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。