内蒙古师范大学《移动应用平台》实验报告

设计题目：简便计算器与羽毛球计分器

学生班级：15网路编程

学生姓名：王佳顺

学生学号：20151104710

指导教师：朝力萌

**摘要**

目前，手机是普及率非常高的电子设备，由于其便于携带，使用方便，资费适中等原因，现在手机在一定程度上开始代替固定电话的通话功能，以及原来电脑软件上的功能了。手机上的软件也随着手机的发展变得丰富起来了，时至今日，已经出现了很多专门制造手机软件应用的公司，虽然制作的多是游戏软件，但是一些辅助性的工作软件也多了很多。

虽然Android系统是现今流行的手机系统，但是iOS的受欢迎程度也很高，现在的时代是触屏手机的时代，无论是iOS还是Android系统，都是支持甚至完全基于触屏开发的系统，可以说，现在市面上的主流手机，无论品牌，都是触屏手机，她已经取代键盘手机成为新一代的宠儿，因此，只有便捷，实用性高的软件，才可以跟上飞速发展的时代。

综上所述，本学期所编写的简便计算器和羽毛球计分器都是基于触屏手机上的一种便携软件，是非常受青睐的软件。

**关键词：**手机，触屏，Android，iOS，简便计算器，羽毛球计分器

1. **引言**

**1.1课题背景与意义**

**1.1.1 课题背景**

随着21世纪的到来，我们已经步入了信息化时代，这个时代最重要的标志就是信息技术的广泛使用，而手机正是信息技术的一个重要体现，如今社会上手机的应用已经到了非常普及的程度，随处都可以看到手机的声影，智能手机的发展锐不可当。如今最受欢迎的系统当属iOS系统，毫无疑问，手机的智能化是手机的一场革命。互联网应用的发展4G

网络的成熟，4G智能手机成为最重要的终端载体，吸引了越来越多消费者利用智能手机网上冲浪、收发电子邮件、看电影、打游戏、视频对话等。智能手机让人们的生活变得越

来越便捷，这就使得智能手机成为了消费者追捧的对象。苹果ios系统作为高端机的智能系统截止至2017年11月，根据Canalys的数据显示，iOS已经占据了全球智能手机系统市场份额的30%，在美国的市场占有率为43%，并在逐渐的增加。

**第二章 设计环境**

**2.1.1 iOS简介**

苹果iOS是由苹果公司开发的手持设备操作系统。苹果公司最早于2007年1月9日的Macworld大会上公布这个系统，最初是设计给iPhone使用的，后来陆续套用到iPod touch、iPad以及Apple TV等苹果产品上。iOS与苹果的Mac OS X操作系统一样，它也是以Darwin为基础的，因此同样属于类Unix的商业操作系统。原本这个系统名为iPhone OS，直到2010年6月7日WWDC大会上宣布改名为iOS。

**2.1.2 iOS发展**

iOS最早于2007年1月9日的苹果Macworld展览会上公布，随后于同年的6月发布的第一版iOS操作系统，当初的名称为“iPhone runs OS X”。 最初，由于没人知道它的潜在价值和发展前景，导致没有一家软件公司、没有一个软件开发者给“iPhone runs OS X”开发软件或者提供软件支持。于是，苹果公司时任CEO斯蒂夫.乔布斯说服各大软件公司以及开发者可以先搭建低成本的网络应用程序（WEB APP）来使得它们能像iPhone的本地化程序来测试“iPhone runs OS X”平台。2007年10月17日，苹果公司发布了第一个本地化IPhone应用程序开发包（SDK），并且计划在2月发送到每个开发者以及开发商手中。2008年3月6日，苹果发布了第一个测试版开发包，将“iPhone runs OS X”改名为“iPhone OS”。2008年9月，苹果公司将iPod touch的系统换成“iPhone OS”。2010年2月27日，苹果公司发布iPad，iPad同样搭载了”iPhone OS”。这年，苹果公司重新设计了“iPhone OS”的系统结构和自带程序。2010年6月，苹果公司将“iPhone OS”改名为“iOS”，同时还获得了思科iOS的名称授权。

**2.2 编程环境**

**2.2.1 Xcode的介绍**

Xcode是一个款强大的IDE开发环境，就像在写Windows程序时需要VS2005一样，需要Xcode为你写Mac程序提供环境。因此，Xcode环境是我们编写简便计算器与羽毛球计分器的第一步。

**2.2.2 Swift语言的特点**

Swift是苹果于WWDC 2014发布的编程语言。

1. Swift用来写iOS和OS X程序。

 2.Swift吸取了C和Objective-C的优点，且更加强大易用。

1. Swift可以使用现有的Cocoa和Cocoa Touch框
2. Swift兼具编译语言的高性能（Performance）和脚本语言的交互性。1、Swift支持诸多高级语言特性，如泛型、闭包、多返回值、类型接口、集合。Swift通过LLVM编译时，会在编译过程中为开发者检查类型异常等错误，为优质代码提供保障。
3. Swift支持Unicode编码，从此开发者可以用喜欢的任何字符作为变量名。
4. Swift虽然是静态语言，但分号(;)不必出现在每行之后，这点相对0biective—C与Java等语言会十分方便，只需保证每条语句在不同行书写即可。当然，如果多条语句出现在同一行时，中间必须用分号(：)作为间隔。
5. Swift可以实现函数的多返回值，这一点在使用其他语言时实现比较烦琐，往往需要构建一个对象或是结构体才能完成。

**2.2.3 应用前景**

作为一项苹果独立发布的支持型开发语言，已经有了数个应用演示及合作开发公司的测试，相信将在未来得到更广泛的应用。某种意义上Swift作为苹果的新商业战略，将吸引更多的开发者入门，从而增强AppStore和MacStore本来就已经实力雄厚的应用数量基础。

**2.2.4 应用范围**

Swift是一种新的编程语言，用于编写iOS和OSX应用。Swift结合了C和Objective-C的优点并且不受C兼容性的限制。Swift采用安全的编程模式并添加了很多新特性，这将使编程更简单，更灵活，也更有趣。Swift是基于成熟而且倍受喜爱得Cocoa和CocoaTouch框架，他的降临将新定义软件开发。Swift的开发从很久之前就开始了。为了给Swift打好基础，苹果公司改进了编译器，调试器和框架结构。我们使用自动引用计（AutomaticReferenceCounting,ARC）来简化内存管理。我们在Foundation和Cocoa的基础上构建框架栈并将其标准化。Objective-C本身支持块、集合语法和模块，所以，框架可以轻松支持现代编程语言技术。正是得益于这些基础工作，我们现在才能发布这样一个用于未来苹果软件开发的新语言。Objective-C开发者对Swift并不会感到陌生。它采用了Objective-C的命名参数以及动态对象模型，可以无缝对接到现有的Cocoa框架，并且可以兼Objective-C代码。在此基础之上，Swift还有许多新特性并且支持过程式编程和面向对象编程。Swift将现代编程语言的精华和苹果工程师文化的智慧结合了起来。编译器对性能进行了优化，编程语言对开发进行了优化，两者互不干扰。

1. **简单计算器与羽毛球计分器**

**3.1.1 系统功能介绍**

（1） 计算器

1.四则运算作为计算器的最基本功能必需具备，要完整支持小数操作。

1. 加、减、乘、除四则运算，要求操作数的输入。实现指定的四则运算。结果的显示。包括输入操作数、运算结果。
2. 要求能删除或者撤销输错的数字。
3. 羽毛球计分器

1.要求能显示参赛选手的头像，对参赛选手的图片能截取并显示在界面上。

2.运用按钮功能，点击按钮功能能连接本地相册，选取运动员头像。

3.将比赛场地实时显示在界面上，能将参赛选手的发球状况显示在场地上。在一局比赛开始时（比分 0 ： 0）或发球方得分为偶数时，发球方在右半场进行发球。当发球方得分为奇数时，在左半场进行发球 。如果发球方取得一分，那么下一回合其继续发球。 如果接发球方取得一分，那么下一回合其成为发球方。

4.能运用循环，实现参赛选手的加分功能，.总分为21 分制，3局2胜。每回合中，取胜的一方加 1 分 ，当双方均为 20 分时，领先对方 2 分的一方赢得该局比赛 ，当双方均为 29 分时，先取得 30 分的一方赢得该局比赛 ，一局比赛的获胜方在下一局率先发球。

5.将总的比分显示在界面上。

6.能有撤销功能，即在输入比分错误的情况下，能撤销所加错的分数。

7.结束比赛功能，在参赛选手分出胜负时，点击结束比赛按钮，能显示出胜利一方，并结束比赛。

8.开始比赛功能，点击开始比赛按钮，才可以为参赛选手加分。

9.只有先将参赛选手头像上传到界面，才能开始比赛，否则不能开始比赛。

10.将比赛结果等数据存储到数据库中，这就要求我们连接数据库，在手机模拟器中下载一个小型数据库SQLite，与本地连接起来，存储数据。

**第三章 代码分析**

**3.1 计算器代码**

|  |
| --- |
|  |
|  | importUIKit |
|  | class ViewController: UIViewController { |
|  | var flg = 0 |
|  | var temp = "" |
|  | var s :String="" |
|  |  |
|  | @IBOutlet weak var x: UITextField! |
|  |  |
|  | @IBAction func one(\_ sender: Any) { |
|  | x.text = x.text! + "1" |
|  | } |
|  | @IBAction func two(\_ sender: Any) { |
|  | x.text = x.text! + "2" |
|  | } |
|  | @IBAction func three(\_ sender: Any) { |
|  | x.text = x.text! + "3" |
|  | } |
|  | @IBAction func fore(\_ sender: Any) { |
|  | x.text = x.text! + "4" |
|  | } |
|  | @IBAction func five(\_ sender: Any) { |
|  | x.text = x.text! + "5" |
|  | } |
|  | @IBAction func six(\_ sender: Any) { |
|  | x.text = x.text! + "6" |
|  | } |
|  | @IBAction func seven(\_ sender: Any) { |
|  | x.text = x.text! + "7" |
|  | } |
|  | @IBAction func eigth(\_ sender: Any) { |
|  | x.text = x.text! + "8" |
|  | } |
|  | @IBAction func nine(\_ sender: Any) { |
|  | x.text = x.text! + "9" |
|  | } |
|  | @IBAction func zero(\_ sender: Any) { |
|  | x.text = x.text! + "0" |
|  | } |
|  | @IBAction func del(\_ sender: UIButton) { |
|  | x.text = "" |
|  | } |
|  | @IBAction func point(\_ sender: UIButton) { |
|  | x.text = x.text!+"." |
|  | } |
|  | @IBAction func jia(\_ sender: UIButton) { |
|  | s = x.text! |
|  | x.text="" |
|  | flg=1 |
|  | } |
|  | @IBAction func jian(\_ sender: UIButton) { |
|  | s = x.text! |
|  | x.text="" |
|  | flg=2 |
|  |  |
|  | } |
|  | @IBAction func cheng(\_ sender: UIButton) { |
|  | s = x.text! |
|  | x.text="" |
|  | flg=3 |
|  |  |
|  | } |
|  | @IBAction func chu(\_ sender: UIButton) { |
|  | s = x.text! |
|  | x.text="" |
|  | flg=4 |
|  |  |
|  | } |
|  | @IBAction func deng(\_ sender: UIButton) { |
|  | switch flg { |
|  | case 1: |
|  | var temp:Double |
|  | temp = Double(s)!+Double(x.text!)! |
|  | x.text="\(temp )" |
|  | case 2: |
|  | var temp:Double |
|  | temp = Double(s)!-Double(x.text!)! |
|  | x.text="\(temp )" |
|  |  |
|  | case 3: |
|  | var temp:Double |
|  | temp = Double(s)!\*Double(x.text!)! |
|  | x.text="\(temp )" |
|  |  |
|  | case 4: |
|  | var temp:Double |
|  | temp = Double(s)!/Double(x.text!)! |
|  | x.text="\(temp )" |
|  | default : |
|  | x.text=x.text |
|  |  |
|  | } |
|  | } |
|  | override func viewDidLoad() { |
|  | super.viewDidLoad() |
|  | // 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. |
|  | } |
|  |  |
|  |  |
|  | } |
|  |  |
|  |  |

**3.2羽毛球记分器**

|  |  |
| --- | --- |
|  |  |
|  | import UIKit |
|  |  |
|  | class ViewController: UIViewController,UIImagePickerControllerDelegate,UINavigationControllerDelegate{ |
|  |  |
|  |  |
|  |  |
|  |  |
|  | var xa:Int = 0 |
|  | var xb:Int = 0 |
|  | var xaa:Int = 0 |
|  | var xbb:Int = 0 |
|  | var dlg=0 |
|  | var flagA=0 |
|  | var flagB=0 |
|  | var del=0 |
|  | var aa = 0 |
|  | var bb = 0 |
|  |  |
|  |  |
|  | @IBOutlet weak var bi: UITextField! |
|  |  |
|  |  |
|  | @IBOutlet weak var vs: UITextField! |
|  |  |
|  | @IBOutlet weak var A: UITextField! |
|  |  |
|  | @IBOutlet weak var B: UITextField! |
|  |  |
|  | @IBOutlet weak var Aimage: UIImageView! |
|  | @IBOutlet weak var Bimage: UIImageView! |
|  |  |
|  |  |
|  |  |
|  | @IBOutlet weak var game: UITextField! |
|  |  |
|  |  |
|  | @IBOutlet weak var Aleft: UITextField! |
|  |  |
|  | @IBOutlet weak var Aright: UITextField! |
|  |  |
|  | @IBOutlet weak var Bleft: UITextField! |
|  |  |
|  | @IBOutlet weak var Bright: UITextField! |
|  |  |
|  | @IBOutlet weak var Aall: UITextField! |
|  |  |
|  | @IBOutlet weak var Ball: UITextField! |
|  |  |
|  | @IBAction func Aphoto(\_ sender: Any) { |
|  |  |
|  |  |
|  | if dlg==0{ |
|  | flagA=1 |
|  | flagB=0 |
|  | if UIImagePickerController.isSourceTypeAvailable(.photoLibrary)//初始化图片控制器 |
|  | { |
|  | let picker = UIImagePickerController()//设置代理 |
|  | picker.delegate = self //设定图片控制器类型 |
|  | picker.sourceType = UIImagePickerControllerSourceType.photoLibrary //弹出控制器，显示界面 |
|  | self.present(picker, animated: true, completion: {() ->Void in}) |
|  | }else{ |
|  | print("读取相册错误") |
|  | } |
|  | } |
|  |  |
|  | } |
|  |  |
|  |  |
|  |  |
|  | @IBAction func Bphoto(\_ sender: Any) { |
|  | if dlg==0{ |
|  | flagA=0 |
|  | flagB=1 |
|  | if UIImagePickerController.isSourceTypeAvailable(.photoLibrary)//初始化图片控制器 |
|  | { |
|  | let picker = UIImagePickerController()//设置代理 |
|  | picker.delegate = self //设定图片控制器类型 |
|  | picker.sourceType = UIImagePickerControllerSourceType.photoLibrary //弹出控制器，显示界面 |
|  | self.present(picker, animated: true, completion: {() ->Void in}) |
|  | }else{ |
|  | print("读取相册错误") |
|  | } |
|  | } |
|  |  |
|  | } |
|  | func imagePickerController(\_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [String : Any]) { |
|  | //查看info对象 |
|  | print(info)//显示的照片 |
|  | let image:UIImage! |
|  | //获取选择的原图 |
|  | image = info[UIImagePickerControllerOriginalImage] as!UIImage |
|  | if(flagA==1){ |
|  | Aimage.image = image |
|  | }else if(flagB==1){ |
|  | Bimage.image = image |
|  | }// |
|  | picker.dismiss(animated: true, completion: {() ->Void in}) |
|  | } |
|  | /\*从相册中选取图片，将info中的程序添加一个相册处理功能的应用\*/ |
|  |  |
|  | @IBAction func Aget(\_ sender: Any) { |
|  | if dlg==1{ |
|  | del=1 |
|  | if game.text != "Game Over"{ |
|  | A.textAlignment = .center |
|  | xa=xa+1 |
|  | if xa>=21{ |
|  | let xaa=xa-2 |
|  | if xaa >= xb{ |
|  | A.text="胜利" |
|  | B.text="失败" |
|  | aa = aa + 1 |
|  | Aall.text = "\(aa)" |
|  | xa = 0 |
|  | xb = 0 |
|  | Bleft.text="" |
|  | Bright.text="" |
|  | Aleft.text="" |
|  | Aright.text="" |
|  | if aa == 2{ |
|  | game.text="Game Over" |
|  | xa = 0 |
|  | A.text = "" |
|  | xb = 0 |
|  | B.text = "" |
|  |  |
|  |  |
|  | } |
|  |  |
|  |  |
|  | } |
|  | else{ |
|  | if xa>xb{ |
|  | if xa==30{ |
|  | A.text="胜利" |
|  | B.text="失败" |
|  | aa = aa + 1 |
|  | Aall.text = "\(aa)" |
|  | Bleft.text="" |
|  | Bright.text="" |
|  | Aleft.text="" |
|  | Aright.text="" |
|  | if aa == 2{ |
|  | game.text="Game Over" |
|  | xa = 0 |
|  | A.text = "" |
|  | xb = 0 |
|  | B.text = "" |
|  | aa = 0 |
|  | bb = 0 |
|  |  |
|  | } |
|  |  |
|  | } |
|  | else{ |
|  | A.text="\(xa)" |
|  | if xa%2 != 0{ |
|  | Aleft.text="\("发球")" |
|  | Aright.text="" |
|  | Bleft.text="" |
|  | Bright.text="" |
|  | } |
|  | else{ |
|  | Aleft.text="" |
|  | Aright.text="\("发球")" |
|  | Bleft.text="" |
|  | Bright.text="" |
|  |  |
|  | } |
|  | } |
|  | } |
|  | else{ |
|  | A.text="\(xa)" |
|  | if xa%2 != 0{ |
|  | Aleft.text="\("发球")" |
|  | Aright.text="" |
|  | Bleft.text="" |
|  | Bright.text="" |
|  | } |
|  | else{ |
|  | Aleft.text="" |
|  | Aright.text="\("发球")" |
|  | Bleft.text="" |
|  | Bright.text="" |
|  |  |
|  | } |
|  |  |
|  |  |
|  |  |
|  | } |
|  |  |
|  | } |
|  | } |
|  | else { |
|  | A.text="\(xa)" |
|  | if xa%2 != 0{ |
|  | Aleft.text="\("发球")" |
|  | Aright.text="" |
|  | Bleft.text="" |
|  | Bright.text="" |
|  | } |
|  | else{ |
|  | Aleft.text="" |
|  | Aright.text="\("发球")" |
|  | Bleft.text="" |
|  | Bright.text="" |
|  |  |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |
|  | /\*选手A的分数如果先到了21分，则A胜利，这里涉及到了发球选手的代码，如果A得分，则A发球，如果A胜利一局，则A发球\*/ |
|  |  |
|  | @IBAction func Bget(\_ sender: Any) { |
|  | if dlg==1{ |
|  | del=2 |
|  | if game.text != "Game Over"{ |
|  | B.textAlignment = .center |
|  | xb=xb+1 |
|  | if xb>=21{ |
|  | let xbb=xb-2 |
|  | if xbb >= xa{ |
|  | B.text="胜利" |
|  | A.text="失败" |
|  | bb = bb + 1 |
|  | Ball.text = "\(bb)" |
|  | xa = 0 |
|  | xb = 0 |
|  | Bleft.text="" |
|  | Bright.text="" |
|  | Aleft.text="" |
|  | Aright.text="" |
|  | if bb == 2{ |
|  | game.text="Game Over" |
|  | xb = 0 |
|  | A.text = "" |
|  | xb = 0 |
|  | B.text = "" |
|  |  |
|  |  |
|  | } |
|  |  |
|  |  |
|  | } |
|  | else{ |
|  | if xb>xa{ |
|  | if xb==30{ |
|  | B.text="胜利" |
|  | A.text="失败" |
|  | bb = bb + 1 |
|  | Ball.text = "\(bb)" |
|  | Bleft.text="" |
|  | Bright.text="" |
|  | Aleft.text="" |
|  | Aright.text="" |
|  | if bb == 2{ |
|  | game.text="Game Over" |
|  | xb = 0 |
|  | A.text = "" |
|  | xb = 0 |
|  | B.text = "" |
|  | aa = 0 |
|  | bb = 0 |
|  |  |
|  | } |
|  |  |
|  | } |
|  | else{ |
|  | B.text="\(xb)" |
|  | if xb%2 != 0{ |
|  | Aleft.text="" |
|  | Aright.text="" |
|  | Bleft.text="\("发球")" |
|  | Bright.text="" |
|  | } |
|  | else{ |
|  | Aleft.text="" |
|  | Aright.text="" |
|  | Bleft.text="" |
|  | Bright.text="\("发球")" |
|  |  |
|  | } |
|  | } |
|  | } |
|  | else{ |
|  | B.text="\(xb)" |
|  | if xb%2 != 0{ |
|  | Aleft.text="" |
|  | Aright.text="" |
|  | Bleft.text="\("发球")" |
|  | Bright.text="" |
|  | } |
|  | else{ |
|  | Aleft.text="" |
|  | Aright.text="" |
|  | Bleft.text="" |
|  | Bright.text="\("发球")" |
|  |  |
|  | } |
|  |  |
|  |  |
|  |  |
|  | } |
|  |  |
|  | } |
|  | } |
|  | else { |
|  | B.text="\(xb)" |
|  | if xb%2 != 0{ |
|  | Aleft.text="" |
|  | Aright.text="" |
|  | Bleft.text="\("发球")" |
|  | Bright.text="" |
|  | } |
|  | else{ |
|  | Aleft.text="" |
|  | Aright.text="" |
|  | Bleft.text="" |
|  | Bright.text="\("发球")" |
|  |  |
|  | } |
|  | } |
|  | } |
|  | } |
|  |  |
|  | } |
|  | /\*选手B的分数如果先到了21分，则B胜利，这里涉及到了发球选手的代码，如果B得分，则B发球，如果胜利一局，则B发球\*/ |
|  | @IBAction func start(\_ sender: Any) { |
|  | xb = 0 |
|  | A.text = "" |
|  | xb = 0 |
|  | B.text = "" |
|  |  |
|  |  |
|  | if Bimage.image != nil && Aimage.image != nil { |
|  | if dlg==0{ |
|  | dlg=1 |
|  | if xa==0{ |
|  | if xb==0{ |
|  | A.textAlignment = .center |
|  | B.textAlignment = .center |
|  | bi.textAlignment = .center |
|  | vs.textAlignment = .center |
|  | Bleft.textAlignment = .center |
|  | Bright.textAlignment = .center |
|  | Aleft.textAlignment = .center |
|  | Aright.textAlignment = .center |
|  | game.textAlignment = .center |
|  | bi.text = ":" |
|  | A.text = "0" |
|  | B.text = "0" |
|  | vs.text = "VS" |
|  | game.text = "" |
|  | let temp = Int(arc4random()%100)+1 |
|  | if temp%2==0{ |
|  | Bright.text="\("发球")" |
|  | } |
|  | else{ |
|  | Aright.text="\("发球")" |
|  | } |
|  |  |
|  | } |
|  | } |
|  | } |
|  | } |
|  | else{ |
|  | if Aimage.image == nil && Bimage.image != nil { |
|  | let alertController = UIAlertController(title:"系统提示", |
|  | message:"请先上传选手A照片：\n", |
|  | preferredStyle:. alert) |
|  |  |
|  |  |
|  | //显示显示框 |
|  | self.present(alertController,animated: true,completion: nil) |
|  | let okAction = UIAlertAction(title: "",style: .default,handler:{ |
|  | action in |
|  | print("") |
|  | }) |
|  | alertController.addAction(okAction) |
|  | //5秒钟后自动消失 |
|  | DispatchQueue.main.asyncAfter(deadline:DispatchTime.now() + 5) { |
|  | self.presentedViewController?.dismiss(animated: false, completion: nil) |
|  | } |
|  | } |
|  | } |
|  | } |
|  | /\*这里需要先上传照片才能开始比赛，如果未上传照片，则点击比赛按钮无效\*/ |
|  | @IBAction func dele(\_ sender: Any) { |
|  | if dlg==1{ |
|  | if game.text==""{ |
|  | if del==1{ |
|  | if xa>=1{ |
|  | xa=xa-1 |
|  | A.text="\(xa)" |
|  | del=0 |
|  | } |
|  | } |
|  | else if del==2{ |
|  | if xb>=1{ |
|  | xb=xb-1 |
|  | B.text="\(xb)" |
|  | del=0 |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |
|  | /\*这里的撤销按钮的功能是，可以撤销加错的分数\*/ |
|  |  |
|  | @IBAction func close(\_ sender: Any) { |
|  | if dlg==1{ |
|  | if game.text=="Game Over"{ |
|  | A.text="" |
|  | B.text="" |
|  | bi.text = "" |
|  | vs.text = "" |
|  | dlg=0 |
|  | Bleft.text="" |
|  | Bright.text="" |
|  | Aleft.text="" |
|  | Aright.text="" |
|  | game.text="" |
|  | Bimage.image=nil |
|  | Aimage.image=nil |
|  | xa=0 |
|  | xb=0 |
|  | flagA=0 |
|  | flagB=0 |
|  | Aall.text="" |
|  | Ball.text="" |
|  | } |
|  | } |
|  | } |
|  | /\*这里的关闭按钮是结束比赛的意思，就是结束当前俩位选手的比赛，以此可以进行下一场比赛\*/ |
|  |  |
|  |  |
|  |  |
|  | } |

1. **总结**

经过这一学期的ios移动应用开发实践，我终于体验到，开发一个软件，不仅需要扎实的理论依据，还需要对一款软件的了解，更要不断了解所编程序的功能和要求，最重要的一点是，要有缜密的逻辑思维，还要有一颗不断学习的心，在这一学期里，我懂得了很多，也成熟了很多，我能够在遇到问题时，可以不断找办法解决，而不是轻易放弃，这是我这一学期最大的收获，当今世界，信息技术飞速发展，各种各样的编程语言层出不穷，我们只有不断学习，不断接受新的事物，才能在飞速变换的编程世界里有立足之地，只有不断学习，不放弃，才是我这一学期最大的收获。