# 记分器实验报告

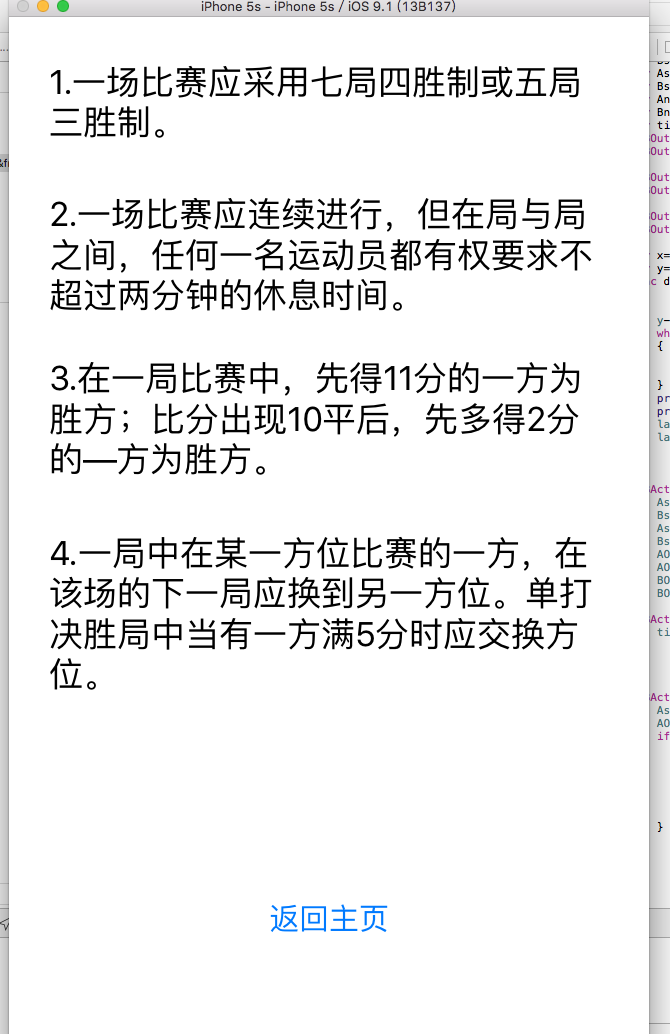
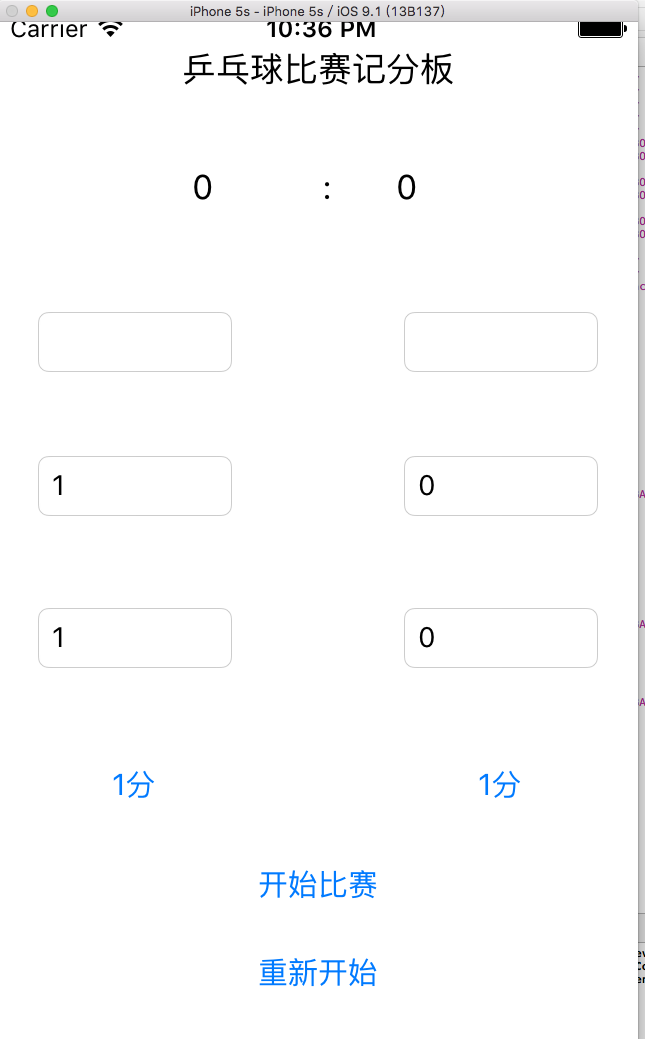
## 实验目的

为了熟悉swift编程，将所学的理论知识付诸于实践，我们将在mac操作系统下开发一个简易的记分器。

## 实验要求

记分器可以连接数据库，使之在意外退出的时候再重新进入可以记录之前的比分。

1. **程序主界面**

****

1. **功能实现代码**

**4.1程序介绍**

此计分器为乒乓球比赛计分器，共有三个界面。主界面包括两个按钮分别是开始比赛和比赛规则。点击开始比赛可以跳转到功能界面。

在功能界面内点击开始比赛，界面上方计时器开始计时，比赛开始。从上至下的三个文本框分别是比赛参赛者名字，场分和局分。点击一分按钮可以加分。点击重置比赛按钮，比赛得分和参赛人员即清空，比赛可以重新开始。点击返回主页即可返回主界面。

用户在主界面点击比赛规则可以跳转到比赛规则介绍界面，了解乒乓球比赛规则，点击其内返回主页按钮即可返回主界面。

**4.2 主要代码**

界面之间跳转由Xcode提供控件直接生成并未使用到代码。图片使用了Xcode中提供的控件imageview。

4.2.1 计时器代码

var x=15  
var y=1  
func dec()  
{  
   
 y-=1  
 while y==0  
 {  
 x=x-1  
 y=60  
 }  
 print(x)  
 print(y)  
 la1.text=String(x)  
 la2.text=String(y)  
}

此部分代码定义了分，秒两个变量和一个方法。其中x为分，y为秒，当y等于0时，x-1.y重置为60.

@IBOutlet weak var la1: UILabel!  
@IBOutlet weak var la2: UILabel!

@IBAction func start(sender: UIButton) {  
 time=NSTimer.scheduledTimerWithTimeInterval(1, target: self, selector: Selector("dec"), userInfo: nil, repeats: true)  
}

此部分代码绑定了点击比赛开始按钮的事件以及两个用来显示时间x,y的两个label控件，点击按钮即可调用之前定义的方法dec()。

4.2.2 计分代码

var Ascore = 0  
var Bscore = 0  
var Ascore\_2 = 0  
var Bscore\_2 = 0  
var Aname = ""  
var Bname = ""  
@IBOutlet weak var AContestantName: UITextField!  
@IBOutlet weak var BContestantName: UITextField!  
  
@IBOutlet weak var AOutResult\_2: UITextField!  
@IBOutlet weak var AOutResult: UITextField!  
  
@IBOutlet weak var BOutResult\_2: UITextField!  
@IBOutlet weak var BOutResult: UITextField!

此处代码定义了从上至下的三个文本框分比赛参赛者名字，场分和局分。并进行了初始化。

@IBAction func AscoreAdd(sender: AnyObject) {  
 Ascore = Ascore + 1  
 AOutResult.text = ("\(Ascore)")  
 if(Ascore == 11 && Bscore < 10){  
 Ascore\_2 = Ascore\_2 + 1  
 AOutResult\_2.text = ("\(Ascore\_2)")  
 Ascore = 0  
 AOutResult.text = ("\(Ascore)")  
 Bscore = 0  
 BOutResult.text = ("\(Bscore)")  
 } else if(Ascore >= 10 && Bscore >= 10 && Ascore - Bscore >= 2){  
 Ascore\_2 = Ascore\_2 + 1  
 AOutResult\_2.text = ("\(Ascore\_2)")  
 Ascore = 0  
 AOutResult.text = ("\(Ascore)")  
 Bscore = 0  
 BOutResult.text = ("\(Bscore)")  
 }  
 saveScore()  
}  
  
@IBAction func BscoreAdd(sender: AnyObject) {  
 Bscore = Bscore + 1  
 BOutResult.text = ("\(Bscore)")  
 if(Bscore == 11 && Ascore < 10){  
 Bscore\_2 = Bscore\_2 + 1  
 BOutResult\_2.text = ("\(Bscore\_2)")  
 Ascore = 0  
 AOutResult.text = ("\(Ascore)")  
 Bscore = 0  
 BOutResult.text = ("\(Bscore)")  
 } else if(Ascore >= 10 && Bscore >= 10 && Bscore - Ascore >= 2){  
 Bscore\_2 = Bscore\_2 + 1  
 BOutResult\_2.text = ("\(Bscore\_2)")  
 Ascore = 0  
 AOutResult.text = ("\(Ascore)")  
 Bscore = 0  
 BOutResult.text = ("\(Bscore)")  
 }  
   
 saveScore()  
}

此处代码为计分功能代码。每点击一次1分按钮，其对应参赛人员局分加1，当某一一个参赛人员获得本局胜利后其对应场分加1，并将自己及对手的局分置为0，开始进行下一局比赛。

@IBAction func Restart(sender: AnyObject) {  
 Ascore = 0  
 Bscore = 0  
 Ascore\_2 = 0  
 Bscore\_2 = 0  
 AOutResult.text = ("\(Ascore)")  
 AOutResult\_2.text = ("\(Ascore\_2)")  
 BOutResult.text = ("\(Bscore)")  
 BOutResult\_2.text = ("\(Bscore\_2)")  
}

此处代码为重置比赛功能。点击之后将参赛者的局分和场分都置为0。

4.2.3数据库

override func viewDidLoad() {  
 super.viewDidLoad()  
   
 //获取数据库实例  
 db = SQLiteDB.sharedInstance()  
 //如果表还不存在则创建表  
 db.execute("create table if not exists p\_contestant(acontestantname varchar(20),ascore varchar(10), ascore\_2 vachar(10))")  
 db.execute("create table if not exists p\_contestant02(bcontestantname varchar(20),bscore varchar(10), bscore\_2 vachar(10))")  
 //如果有数据则加载  
 initUser()  
}  
  
//从SQLite加载数据  
func initUser() {  
 let data = db.query("select \* from p\_contestant")  
 let data\_2 = db.query("select \* from p\_contestant02")  
 if data.count > 0 {  
 //获取最后一行数据显示  
 let acontestant = data[data.count - 1]  
 AContestantName?.text = acontestant["acontestname"] as? String  
 AOutResult?.text = acontestant["ascore"] as? String  
 AOutResult\_2?.text = acontestant["ascore\_2"] as? String  
   
   
 }  
 if data\_2.count > 0 {  
 //获取最后一行数据显示  
 let bcontestant = data\_2[data\_2.count - 1]  
 BContestantName?.text = bcontestant["bcontestname"] as? String  
 BOutResult?.text = bcontestant["bscore"] as? String  
 BOutResult\_2?.text = bcontestant["bscore\_2"] as? String  
   
 }  
}  
  
//保存数据到SQLite  
func saveScore() {  
 let Aname = self.AContestantName.text!  
 let Bname = self.BContestantName.text!  
 let Ascore = self.AOutResult.text!  
 let Ascore\_2 = self.AOutResult\_2.text!  
 let Bscore = self.BOutResult.text!  
 let Bscore\_2 = self.BOutResult\_2.text!  
 //插入数据库，这里用到了esc字符编码函数，其实是调用bridge.m实现的  
 let sql = "insert into p\_contestant(acontestantname,ascore,ascore\_2) values('\(Aname)','\(Ascore)','\(Ascore\_2)')"  
 let sql02 = "insert into p\_contestant02(bcontestantname,bscore,bscore\_2) values('\(Bname)','\(Bscore)','\(Bscore\_2)')"  
 print("sql: \(sql)")  
 print("sql:\(sql02)")  
 //通过封装的方法执行sql  
 let result = db.execute(sql)  
 let result02 = db.execute(sql02)  
 print(result)  
 print(result02)  
}

此处代码进行了数据库的操作，使程序即使意外关闭，也可以保存比赛数据。

1. **总结**

本次实验是在计算器之后进行的，通过此次实验，我加深了对swift这门编程语言的了解。成功的运用所学的知识解决了一个相对实际的问题。在本次实验中我学会了如何在Xcode中配置数据库等一些新的知识。但是本程序还有一些未能解决的Bug，例如计时器秒只能从60开始等。