**课 程 设 计 报 告**

**课程设计名称：**移动平台应用开发

**课程设计题目**：计分器

院（系）：计算机与信息工程学院

专 业：软件工程

班 级：2013软件工程

学 号：20131105786

姓 名：冯志

指导教师：朝力萌

完成日期：2016.6.22

**目 录**

[第1章 概要设计 2](#_Toc406624046)

[1.1题目的内容与要求 2](#_Toc406624047)

[1.2总体结构 2](#_Toc406624048)

[第2章 详细设计 2](#_Toc406624049)

[2.1主模块 2](#_Toc406624050)

[第3章 调试分析 3](#_Toc406624051)

[第4章 使用说明与执行结果 4](#_Toc406624052)

# 第1章 概要设计

## 1.1题目的内容与要求

**内容：编写出一个简单计分器**

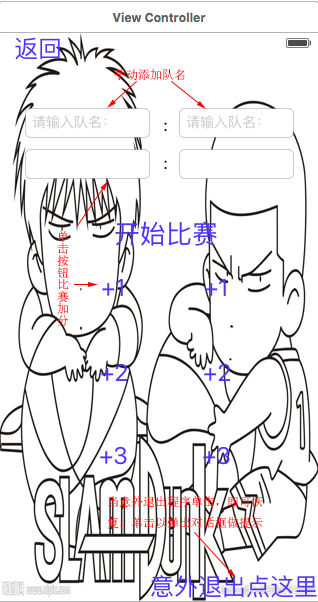
**要求： 利用swift语言编写出一个简单的计分器**

**可以实现倒计时、计分、查询历史纪录、录入队员信息等功能的计分器**

# 第2章 详细设计

## 2.1主模块

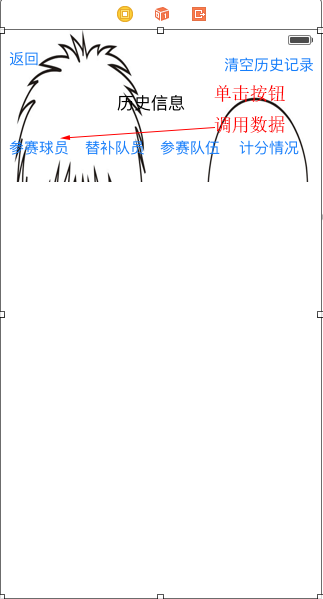
**单击计分按钮**



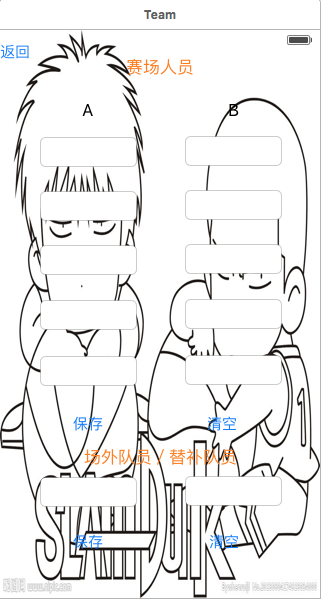
**运行程序界面：**

****

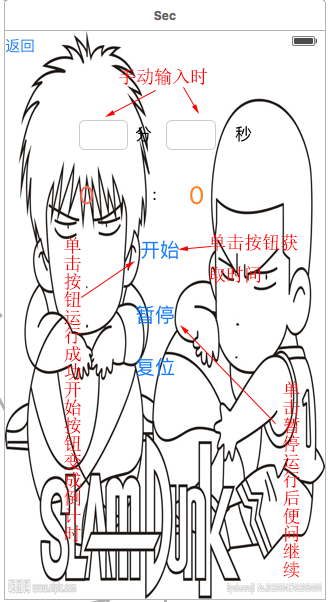
单击查询历史：



单击参赛人员按钮



单击计时按钮：



**程序主要代码：**

1. **计时：**

1）、执行倒计时功能，获取text文本框中的数据，利用计时器功能，并弹出对话框

x=60\*(min.text! as NSString).intValue  
y=(sec.text! as NSString).intValue  
leftTime=x+y  
if min.text!==""&&sec.text==""  
{  
 a("请输入时间",tile:"")  
}else if min.text=="0"&&sec.text=="0"||min.text=="0"&&sec.text==""||min.text==""&&sec.text=="0"  
{  
 a("该时间不能倒计时",tile:"")  
}  
else  
{  
 startBt.setTitle("倒计时中",forState:UIControlState.Normal)  
 start=true  
 timer = NSTimer.scheduledTimerWithTimeInterval(NSTimeInterval(1),target:self,selector:Selector("tickDown"),userInfo:nil,repeats:true)  
 a("倒计时开始，还有 \(leftTime) 秒..",tile: "倒计时开始")  
 start=true  
   
 //var emptyArray: [Int32] = []  
 save()  
}



//写暂停和继续功能定义bool值 start=false，判断其值做相应的功能

func stoporstart()  
{  
 if start==true  
 {  
 stoporcon.setTitle("继续", forState: UIControlState.Normal)  
 start=false  
 a("已经暂停，还有 \(leftTime) 秒...",tile: "")  
 timer.invalidate()  
   
   
 }else  
 {  
 if min.text==""&&sec.text==""||min.text=="0"&&sec.text=="0"||min.text==""&&sec.text=="0"||min.text=="0"&&sec.text==""  
 {  
 a("",tile: "还没有开始")  
 }else  
 {  
 stoporcon.setTitle("暂停", forState: UIControlState.Normal)  
 timer = NSTimer.scheduledTimerWithTimeInterval(NSTimeInterval(1),target:self,selector:Selector("tickDown"),  
 userInfo:nil,repeats:true)  
 a("已经暂停，还有 \(leftTime) 秒...",tile: "到计时开始")  
 start=true  
 }  
 }  
   
}

//写弹出对话框

func a(msg:String,tile:String)  
{  
   
 alertView.message = msg  
 alertView.title = tile  
 alertView.addButtonWithTitle("确定")  
 alertView.show()  
}

**2.计分**

**1）、//拖控件text文本上显示所加分数**

@IBAction func one(sender: UIButton) {  
 a=a+1  
 text1.text=("\(a)")  
 saveUser()  
}

**2）、当程序意外退出获取文本框中的数据以便于继续比赛，并弹出提示对话框**

@IBAction func exitt(sender: UIButton) {  
 var x:Int32=0  
 var y:Int32=0  
 x=(text1.text! as NSString).intValue  
 y=(text2.text! as NSString).intValue  
 a=x  
 b=y  
 alertView = UIAlertView()  
 alertView.message = "非常抱歉，由于系统原因意外退出，现在您可继续使用"  
 alertView.title = "Sorry"  
 alertView.addButtonWithTitle("确定")  
 alertView.show()  
}



**3）、//将数据存入数据库并写入按钮中**

func saveUser() {  
 let Ascore = self.text1.text!  
 let Bscore = self.text2.text!  
 let Aname = self.name1.text!  
 let Bname = self.name2.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 name(Aname,Bname) values('\(Aname)','\(Bname)')"  
 let sql2 = "insert into score(Ascore,Bscore) values('\(Ascore)','\(Bscore)')"  
 print("sql: \(sql2)")  
 print("sql: \(sql)")  
 //通过封装的方法执行sql  
 let result = db.execute(sql)  
 \_ = db.execute(sql2)  
 print(result)  
}

**3参赛人员**

**//添加相对应按钮**

@IBAction func save1(sender: UIButton) {  
 save1()  
}

**//写入方法**

func save1(){  
 let Bname6 = self.B6.text!  
 let Aname6 = self.A6.text!  
   
 let sql2 = "insert into turnpeople(Aname6,Bname6) values('\(Aname6)','\(Bname6)')"  
 print("sql: \(sql2)")  
 let result = db.execute(sql2)  
 print(result)  
   
}

**4．查询历史**

**//添加对应按钮**

@IBAction func score(sender: UIButton) {  
 initscore()  
}

**//填写查询数据库内容**

func initscore()  
{  
 text.text=""  
 let score=db.query("select \* from score")  
 text.text! += "历史记录:"+String(score.count)+"\n"  
 for var a=0;a<score.count;a++  
 {  
 let score1 = score[a]  
 text.text! += String(score1["uid"]!)+" "+"A队得分："+String(score1["Ascore"]!)+" "+"B队得分："+String(score1["Bscore"]!)+"\n"  
 }  
   
}

**//删除表中数据**

@IBAction func del(sender: UIButton) {  
 alertView.title="确定要历史记录吗"  
 alertView.addButtonWithTitle("删除")  
 alertView.addButtonWithTitle("确定")  
 alertView.show()  
   
 // if alertView.clipsToBounds  
 //{  
 let x=db.query("delete from score")  
 db.query("delete from name")  
 db.query("delete from nomalpeople12")  
 db.query("delete from turnpeople")  
 text.text = "历史记录:"+String(x.count)+"\n"  
   
 //}  
}

**5. 数据库表**

**数据库所建的表 score name team turn四个表**

# 第3章 调试分析

**对程序分别进行以下操作：**

1. **分别单击主界面四个按钮可以实现跳转**
2. **在计时中可以正确获取文本框中的内容，并且可以实现计时暂停等功能**
3. **计分可以实现基本加分功能，当意外退出可以实现继续使用，并可以存入相对应数据库表**
4. **参赛人员可以实现添加数据库，清空功能**
5. **历史纪录可以实现查询四个表中的数据，并且可以清空历史纪录并弹出对话框**

# 第4章 使用说明与执行结果

运行操作及结果:

通过调试分析可以实现连接数据库，并实现增删改查，基本实现计时器功能

**二．课程设计小结**

**在实验中发现——分析——总结的问题有：**

实现了计分器的基本功能但是还有很多功能没有实现如不能实现：

1. 无法跨界面运用
2. 弹出对话框后确定和取消无法执行
3. 没有实现罚球，罚下场功能