移动平台应用开发—计分器

计算机与信息工程学院 2013级 软件工程 于丹丹 20131105783

指导教师 朝力萌 讲师

摘 要 iOS 是世界上最先进的移动平台，它重新定义了只有口袋大小的移动设备可以实现的功能。与 Xcode 集成开发环境工具集成的 iOS SDK 使得创建优秀的应用程序变得十分轻松。由于苹果 App Store 存在于每一个 iOS 操作系统设备上，并在世界各地都有本地化，因此对移动开发者们来说根本没有什么其他的平台比它更吸引人。Swift 是一种新的编程语言，用于编写 iOS 和 OS X 应用。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 Cocoa Touch 框架，他的降临将重新定义软件开发。

关键词 ios；Xcode集成开发环境；swift；

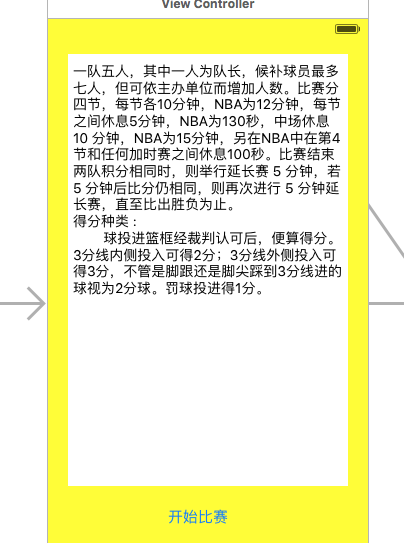
# 

# 1引言

篮球比赛在中国越来越被人们关注，同时也被更多青少年所喜爱。篮球赛记时计分器是对比赛过程中所产生的时间、比分等数据进行快速采集记录、加工处理、传递利用的工具。根据不同运动项目的不同比赛规则要求，篮球赛记时计分器包括评分类、命中类、制胜类、得分类等多种类型。 篮球比赛是根据运动队在比赛时间里得分多少来决定胜负的。因此，篮球比赛的记时计分器是一种得分类型的工具。篮球比赛的记时计分器由计时器、计分器等多种电子设备组成。同时，根据目前高水平蓝球比赛要求，完善的蓝球比赛计时计分系统设备应该能够与现场成绩处理、现场大屏幕、电视转播车等多种设备相联，以便实现提高比赛现场感、表演娱乐观众等功能。

1. 实验要求与数据
2. 按照篮球比赛规则，需要两个计分器实现双方比分的显示及比赛时间的显示（倒计时），时间精确到秒，设每节比赛的初始设置时间为12分00秒，双方初始比分显示为00；
3. 双方分别用3个按键表示加分操作，分别是1分，2分，3分；
4. 计分器应有启动、教练暂停、犯规罚分和复位开关。复位开关有效时，双方比分自动清零显示数字“00”；启动开关有效时，比赛开始计时，允许记分。教练暂停按钮有效时，停止计时，不允许双方改变比分，释放暂停开关后，继续计时比赛。犯规罚分开关有效时，停止计时，允许双方改变比分，释放该开关后，继续计时比赛。

3主窗体设计



# 功能模块

# 4.1篮球比赛窗体设计



# 实现计时器



点击“开始比赛”，计时器从12分钟开始倒计时。

* + 1. 实现清零



点击 “清零”按钮，计时器重新从12分钟考试倒计时。

* + 1. 实现暂停



点击”暂停”按钮，计时器时间暂停。

4.1.4 实现继续



点击”继续”按钮，计时器时间继续倒计。

5代码

5.1计时功能实现代码

var timer:NSTimer!

var fen=12

var miao=0

func tickDown()

    {

        min.text=String(fen)

        sec.text=String(miao)

        if(fen != 0 && miao == 0)

        {

            fen=fen-1

            miao=60

        }

        miao = miao-1

}

5.1.1 计时开始实现代码

 @IBAction func start(sender: UIButton) {

        timer = NSTimer.scheduledTimerWithTimeInterval(NSTimeInterval(1),target:self,selector:Selector("tickDown"),

            userInfo:nil,repeats:true)

}

5.1.2 计时暂停实现代码

 @IBAction func stop(sender: UIButton) {

        timer.invalidate();

    }

5.1.3计时继续实现代码

 @IBAction func con(sender: UIButton) {

        timer = NSTimer.scheduledTimerWithTimeInterval(NSTimeInterval(1),target:self,selector:Selector("tickDown"),

            userInfo:nil,repeats:true)

    }

5.1.4 计时清零实现代码

 @IBAction func reaction(sender: UIButton) {

        Ascore.text="0"

        Bscore.text="0"

        timer?.invalidate();

        min.text="12"

        sec.text="00"

    }

5.2 计分实现代码

 @IBOutlet var Ascore: UILabel!

@IBOutlet var Bscore: UILabel!

@IBAction func A3(sender: UIButton) {

        x=x+3

        Ascore.text=String(x)

        saveUser()

    }

    @IBAction func A2(sender: UIButton) {

        x=x+2

        Ascore.text=String(x)

        saveUser()

    }

    @IBAction func A1(sender: UIButton) {

        x=x+1

        Ascore.text=String(x)

        saveUser()

    }

    @IBAction func B3(sender: UIButton) {

        y=y+3

        Bscore.text=String(y)

        saveUser()

    }

    @IBAction func B2(sender: UIButton) {

        y=y+2

        Bscore.text=String(y)

        saveUser()

    }

    @IBAction func B1(sender: UIButton) {

        y=y+1

        Bscore.text=String(y)

        saveUser()

}

5.2.1 计分存储实现代码

var db:SQLiteDB!

 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（其中uid为自增主键）

        db.execute("create table if not exists score(uid integer primary key,Ascore varchar(20),Bscore varchar(20))")

        //如果有数据则加载

        initUser()

        min.text="12"

        sec.text="00"

    }//点击保存

    //从SQLite加载数据

    func initUser() {

        let data = db.query("select \* from score")

        if data.count > 0 {

            //获取最后一行数据显示

            let user = data[data.count - 1]

            Ascore.text = user["Ascore"] as? String

            Bscore.text = user["Bscore"] as? String

        }

    } //保存数据到SQLite

    func saveUser() {

        let Ascore = self.Ascore.text!

        let Bscore = self.Bscore.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 score(Ascore,Bscore) values('\(Ascore)','\(Bscore)')"

        print("sql: \(sql)")

        //通过封装的方法执行sql

        let result = db.execute(sql)

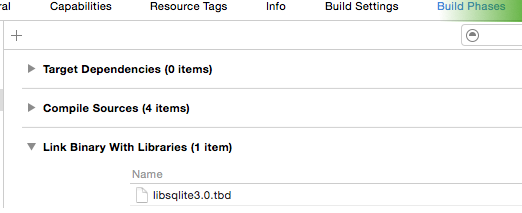
        print(result)

    }

# 6关键技术

6.1 连接数据库

# 1.在Build Phases -> Link Binary With Libraries中点击加号，添加libsqlite3.0.tdb到项目中来

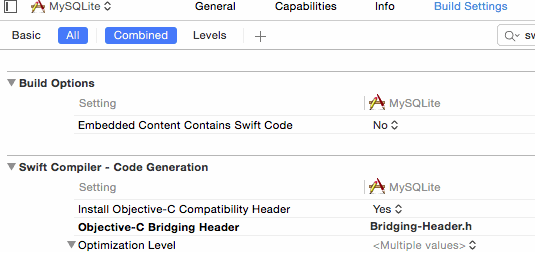


2. 创建连接头文件 Bridging-Header.h

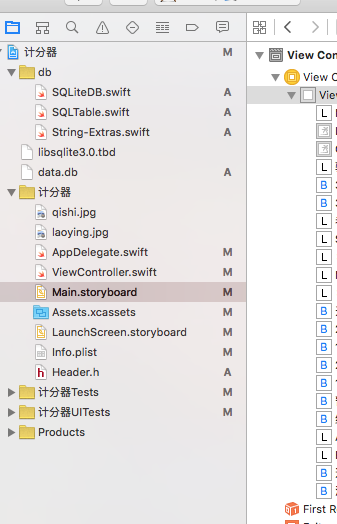
#import "sqlite3.h"

#import <time.h>

1. 在项目编译属性里引用头文件



1. 导入SQLiteDB的代码（SQLiteDB.swift、SQLTable.swift和String-Extras.swift），代码结构如下：



# 7 结束语

通过对移动平台开发的学习，基本了解了swift语言的编程思想，了解了swift语言强大的可移植性，随着时代的发展，我意识到知识的重要性。同时也让我深刻认识到了科学的严谨性，还有自己身上的很多不足之处，今后我会更加严格要求自己，以便提高自己的实操能力。本次实验真的学到了很多，计分功能、计时功能、以及数据存储功能，经过我的努力学习这些功能最终都得以实现，使我有了很大学习下去的勇气。

并且课题在开发的过程中得到了朝力萌老师的精心指导，以及同组学员的互相帮忙在此深表感谢。

Mobile platform application development - calculator

Computer and information engineering college Yu Dandan 20131105783

Directed by Chao Limeng Professor

**Abstract**  IOS is the most advanced mobile platform in the world, it re defines the function that can be realized only by pocket sized mobile devices. IOS integration with Xcode integration development environment tools to create excellent applications become very easy to create SDK. Because Apple's Store App exists on every iOS operating system device, and is localized around the world, so there is no other platform for mobile developers more attractive than it. Swift is a new programming language for the preparation of iOS and X OS applications. Swift uses a safe programming mode and adds a lot of new features, which makes programming easier, more flexible, and more fun. Swift is based on a mature and much loved Cocoa and Touch Cocoa framework, his advent will redefine the software development.

**Keywords**  Xcode; IOS integrated development environment; swift;