

# 利用 iOS4 的新特性进行开发

李亮 [holly.lee@gmail.com](mailto:holly.lee@gmail.com)  
for QCon Beijing 2011

# 历史

- 2007 年 6 月 iPhone, iPhoneOS 1.0
- 2008 年 6 月 iPhone 3G, iPhoneOS 2.0, public SDK
- 2009 年 6 月 iPhone 3GS, iPhoneOS 3.0
- 2010 年 3 月 iPad, iPhoneOS 3.2
- 2010 年 6 月 iPhone 4, iOS 4

# iOS4 的新特性

- 有限的多任务支持和 Local Notification
- 开始支持不同的分辨率和不同 DPI 的设备.
- 针对多核的支持. GCD, 以及语言层面的支持.
- Game Center 的引入. 游戏社区
- iAd. Apple 的应用内嵌广告方案
- CoreTelephony, CoreVideo, etc.



# 有限的多任务

- iOS4 之前,没有多任务的日子.
- iOS4,应用不再退出,而是挂起.
- iOS4,终于给了我们一些多任务,只是很有限.

# 应用程序生命周期

What's new:

- (void) applicationWillEnterForeground:(UIApplication \*)application
- (void) applicationDidEnterBackground:(UIApplication \*)application

What's hard to get called:

- (void) applicationWillTerminate:(UIApplication \*)application

# 运行在后台

- 保持在后台运行的
  - VoIP
  - Locating
  - Audio
- 需要运行一段时间的
- Local notification



# 保持在后台运行

- 系统是否支持后台任务?  
UIDevice multitaskingSupported 属性
- 告诉系统需要后台运行  
Info.plist  
Key: UIBackgroundMode  
Value: audio, location, or voip

# 定位相关

- 只需要跟踪位置有显著变化的  
CLLocationManager  
startMonitoringSignificantLocationChanges  
stopMonitoringSignificantLocationChanges
- 持续使用系统定位服务的
- 需要在后台仍然持续跟踪位置变化的



# 后台音频播放

- Music Playing, streaming, etc...
- Keep running in background, but suspended once audio playing stopped.
- Configuration Audio Session for multiple apps coordination.

# VoIP

- 并非一直运行, 而是监视相应的 socket. 需要特别配置该 socket.
- Keep alive: UIApplication  
setKeepAliveTimeout:handler:

# 运行一段时间的应用

UIApplication

`backgroundTimeRemaining` property

`beginBackgroundTaskWithExpirationHandler:`

`endBackgroundTask:`



# Local notification

- ULocalNotification  
文字, 声音
- UIApplication  
scheduledLocalNotifications  
scheduleLocationNotification:

# 不同的设备

- iPhone, iPad, iPod Touch
- 不同的屏幕尺寸, 分辨率, 是否视网膜屏幕
- 不同的设备特性, 有无电话功能, 有无摄像头, 有无定位功能, 等等

# 声明设备能力

Info.plist 中声明需要的设备能力. iTunes 以及 AppStore 用来鉴别是否可以安装到某个设备上.

Key: **UIRequiredDeviceCapabilities**

Value: **Array or Dictionary.**

包括: 电话, Wifi, sms, 照片摄像头, 视频摄像头, 自动对焦, 闪光灯, 加速度测量仪, 陀螺仪, 罗盘, gps, 麦克风, CPU 类型, 蓝牙, 是否支持 OpenGL ES, 蓝牙支持, 等等.



# 程序中的判断

## UIDevice

UIUserInterfaceIdiom userInterfaceIdiom

## UIImagePickerController

- + (BOOL)isCameraDeviceAvailable:
- + (NSArray \*)availableCaptureModesForCameraDevice:
- + (BOOL)isFlashAvailableForCameraDevice:

## CLLocationManager

+(BOOL) headingAvailable

## CMMotionManager

BOOL gyroAvailable

# Retina Display

视网膜屏幕在不需要代码改动的情况下,带来了更高的分辨率和更清晰的图像/字体质量.

- `xxxxxx.png` to `xxxxxx@2x.png` 图像文件
- UIKit/CoreGraphics 调用完全自动处理. 大小和坐标都是一致的.

# Grand Central Dispatch

- 针对多核设计, 面向并发编程
- 操作系统的支持与程序库的结合
- 语言层面的增强: block



# Block

- Block 是在 C 语言上的增强
- 与 closure 或者 lambda 相似
- 表示方式: 以 ^ 代替函数指针的 \*
- 特点: 局部变量的 snapshot; \_\_block 变量; 与 C/Objective-C/C++ 的结合
- 作用: 将代码分割成可并行的小块, 以方便 GCD 进行调度

# Block 的语法示例

```
typedef void (^blockWithString)(char*);  
char *greeting = "hello";  
blockWithString b = ^(char* place){ printf("%s %s\n",  
greeting, place); };  
greeting = "goodbye";  
b("world"); // prints "hello world\n"
```

```
__block char *mutable_greeting = "hello";  
c = ^{ mutable_greeting = "goodbye"; };  
printf("%s", mutable_greeting); // -> "hello"  
c();  
printf("%s", mutable_greeting); // -> "goodbye"
```

# Block 的特点

- 实现: 函数指针加上一些上下文数据
- 内存: 在栈中分配. 超出作用域需要 `copy/release`
- C: `Block_copy(b)`, `Block_release(b)`
- Objective-C: `copy`, `release`. 支持引用计数
- C++: 在 `block` 中用到的C++对象需要实现 `const copy constructor`.



# GCD 的调度队列

libdispatch 提供了 GCD 的类型和函数.

```
dispatch_queue_t a_queue;  
dispatch_async(a_queue, ^{ b; });
```

- 全局并发队列  
dispatch\_get\_global\_queue();
- 私有串行队列  
dispatch\_queue\_create();
- 主队列  
dispatch\_get\_main\_queue();

# GCD 的同步机制

- 同步调度

`dispatch_sync(a_queue, ^{ wait_for_me(); });`

- 回调: 将回调放到 block 的末尾

- Group 机制

`dispatch_group_notify, dispatch_group_wait`

- 信号量

`dispatch_semaphore_signal, dispatch_semaphore_wait`

# GCD 的事件支持

- GCD 支持以时钟, POSIX signal, 文件描述符/socket, 进程状态, Mach port 等作为事件源, 还支持应用自定事件源.
- 事件支持提供了一种不阻塞的处理方式



# Game Center

- 社会化潮流在游戏中的体现
- Game Center 是设备, app 与云结合起来工作的.
- Game Center 提供了认证, 排行榜, 成就, 和为多人网络游戏寻找合适玩家的能力.
- Game Center 在 GameKit framework 中提供.

# Game Center

身份认证. 本地玩家, Apple ID.





# Game Center: Player

- GKPlayer 类用来代表一个玩家. 对任一设备都有一个本地玩家 (GKLocalPlayer), 别的是远程玩家
- `playerID`, `alias`, `isFriend` properties
- 认证本地玩家: GKLocalPlayer 的 `authenticateWithCompletionHandler`: 方法. 检查 `isAuthenticated` property
- 获得远程玩家的信息: GKPlayer 的 `loadPlayersForIdentifiers:withCompletionHnadler`:



# Game Center: Friends

- 取得朋友信息: 认证并成功调用了 `loadFriendsWithCompletionHandler:` 后 `GKLocalPlayer` 的 `friends` property 包含了 friend 信息.
- 利用 `GKFriendRequestComposeViewController` 向别的玩家发出朋友邀请.

# Game Center: 成就

- 设计成就. Id, 标题, 描述, 点数, 图片, 是否可见. iTunes Connect 配置.
- GKAchievement. 报告成就的完成度.  
`percentComplete` property 和  
`reportAchievementWithCompletionHandler`: 方法.
- `loadAchievementsWithCompletionHandler`: 方法来装入目前的成就.
- `resetAchievementWithCompletionHandler`: 重置.
- 用 GKArchivementViewController 显示成就

# GameCenter: 排行榜

- 分数以 64位整数表示. 格式, 分类等通过 iTunes Connect 配置.
- **GKScore** 类. `initWithCategory:`
- 报告分数: `value` property,  
`reportScoreWithCompletionHandler:` 方法.
- **GKLeaderboardViewController** 显示排行榜. `category` 和 `timeScope` property
- **GKLeaderboard** 类可以获取分数以及分类标题



# Game Center: 多人竞技

- 比赛请求 **GKMatchRequest**, 指明人数等
- 显示比赛组成界面

## **GKMatchmakerViewController**

**initWithMatchRequest:**

- 设置 **GKMatchMaker** 的 **inviteHandler** property 来处理来自别人的邀请.
- 比赛组成了或者取消时会调用 **GKMatchmakerViewControllerDelegate** 的相应方法.
- 组成时会返回一个 **GKMatch** 对象. 它包含了参与的玩家的信息以及提供方法在玩家之间传送数据

# iAd

- Apple 的应用程序内广告解决方案
- 两种形式的 View. Banner 和全屏
- 需要先加入到 iAd network 中, 并在 iTunes Connect 中作配置

# iAd: Banner

- **AdBannerView**
- iPhone: 320x50 & 480x32  
iPad: 768x66 & 1024x66
- **AdBannerViewDelegate:**
  - bannerViewActionShouldBegin:willLeaveApplication:
  - bannerViewDidLoadAd:
  - bannerView:didFailToReceiveAdWithError:
  - bannerViewActionDidFinish:



# iAd: 全屏

- 仅适用于 iPad
- **ADInterstitialAd**
- 显示: 独占显示或者作为View的一部分  
`presentFromViewController:`  
`presentInView:`
- **ADInterstitialAdDelegate**  
`interstitialAdActionShouldBegin:willLeaveApplication:`  
`interstitialAdActionDidFinish:`  
`interstitialAdDidUnload:`

# 谢谢

李亮 [holly.lee@gmail.com](mailto:holly.lee@gmail.com)



The QCon logo features a large, stylized letter 'Q' in a vibrant green color, followed by the letters 'Con' in a bold, blue sans-serif font. The background of the entire slide is a photograph of a traditional Chinese stone pagoda with multiple tiers and circular openings, situated in a body of water with a hazy cityscape in the distance.

# QCon

杭州站 · 2011年10月20日~22日

[www.qconhangzhou.com](http://www.qconhangzhou.com) ( 6月启动 )

QCon北京站官方网站和资料下载

[www.qconbeijing.com](http://www.qconbeijing.com)

全球企业开发大会

THE ANNUAL  
INTERNATIONAL  
SOFTWARE DEVELOPMENT  
CONFERENCE