1. 只要继承了UIResponder的对象都可以作为事件的响应者，实质上UIControl继承了UIView,而UIView又继承UIResponder，因此所有的UI控件都可以作为事件的响应者。(UIViewController继承UIResponder，UIResponder继承NSObject)
2. 事件响应者链的典型传播路线：First Responder(第一响应者，用户触摸屏幕的控件)-> First Responder的视图控制器->父控件->父控件的视图控制器->直到顶层视图对应的视图控制器->应用程序窗口(UIWindow,继承UIView)->UIAplication(继承UIResponder)->应用程序委托对象->事件被丢弃(停止传播)
3. 手势处理器(UIGestrueRecognizer)：

UITapGestureRecognizer:点击手势

numberOfTapsRequired:点击几次触发，默认为1

numberOfTouchesRequired:几个手指点击触发，默认1

UIPinchGestureRecognizer：捏合手势

scale:捏合的比例

velocity:捏合的速度

state:捏合的状态 （开始/结束）

UIRotationGestureRecognizer 选择手势

rotation:旋转角度

velocity:旋转速度

UISwipeGestureRecognizer 轻扫手势 --- 贪食蛇

direction: 方向 right/left/up/down

numberOfTouchesRequired: 几个手指触碰

UIPanGestureRecognizer 拖动手势

maximunNumberOfTouches: 最多支持几个手指拖动

minmunNumberOfTouches:至少几个手指拖动

-translationInView 获取拖动位移，返回CGPoint,

-velocityInView 获取拖动速度，返回CGPoint

UILongPressGestureRecognizer 长按手势

minmumPressDuration:至少长按多少秒触，默认0.5

numberOfTouchesRequire:几个手指长按触发

allowableMovement:允许移动最大距离

自定义手势处理器：

继承UIGestureRecognizer,

实现touchesBegan[获取触碰的任意点 [touches anyObject],获取触碰点在view中的坐标 locationInView]

touchesMove/touchesEnd/touchesCancelled

4.国际化（I18N）

设置国际化：

---PROJECT---Info---Localizations----添加语言

设置方式：1.界面不同，设置不同stroyboard，

2.界面相同，设置不同key

4.1 app名称国际化:

新建“Strings”文件，命名为InfoPlist.strings,选中该文件，查看右边工具栏的Localization,选中语言，在对应的string文件设置CFBundleDisplayName的值

4.2 文本国际化

新建“Strings”文件，命名为Localizable.strings,….(同上)

方法： NSLocalizedString(key,comment)

key:string文件对应的key，comment：该消息的注释，可为nil

zh-Hans:简体中文，zh-Hant:繁体中文

5.数据存储

5.1 沙盒：iOS将所有的app保存在application目录下，每个app都保存在一个Globally Unique Identifier(GUID)文件夹中，就是应用都沙盒

保证每个应用程序只能访问本沙盒的数据，避免与其他应用冲突

5.2 沙盒文件结构：

Documents:应用程序都数据、文件

NSString \*docDiectory =

NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,NSUserDomainMask,true);

Library:基于NSUserDefaults

tmp:临时文件，iTunes不会备份

NSString \*tempPath = NSTemporaryDirectory();

5.3 文件保存策略

5.4 少量数据存储（Library/Preferences）

5.4.1 Settings Bundle

5.4.2 NSUserDefaults[XxxForKey: setXxx: forKey:]

注意：key必须是字符串，value可以任意类型

set值后，都要调用 【defaults synchronize】保存

5.4.3 属性列表（NSArray,NSDictionary）

writeToFile:atomically / contentsOfFile

5.4.4 对象归档

-encodeWithCoder / initWithCoder

5.4.5 使用SQLite3数据库

iOS的SQLite使用了原生、面向过程的C函数库，

在项目中配置步骤：TARGET – Build Phases – Link Binary – 添加（搜索sqlite,选择libsqlite3.tbd）-导入头文件（xcode后可省略）

#import <sqlite3.h>

可以使用SQLite Manager管理数据库

5.4.6 使用Core Data框架

CoreData是一个纯粹的面向对象框架，实际上大部分情况下，底层都采用SQLite数据库

6.多媒体应用

6.1使用AVAudioPlayer播放音乐

AVAudioPlayer属于AVFoundation.framework的类，

6.1.1 初始化 initwithContentsOfURL: / initwithData

6.1.2 属性：duration-获取总时长；currentTime-获取当前播放点;

numberOfLoops-循环播放次数，为负数表示无限循环;playing-是否正在播放；volume-设置音量0-1；

6.1.3 代理方法：play-开始或恢复播放；playAtTime-指定时间后播放；pause-暂停；stop-停止；

6.2 使用MPMediaPickerController和MPMusicPlayerController选择播放系统音乐，使用MediaPlayer框架

6.3 使用AVPlayer播放视频

AVPlayer类似于AVAudioPlayer，也需要AVFoundation框架，由于AVPlayer播放视频，需要输出视频播放的界面，所以通过AVPlayerLayer来包装AVPlayer,且AVPlayerLayer继承CALayer,因此用CALayer管理AVPlayerLayer。

6.3.1 初始化-playerWithURL