```
1.触摸事件
                                         拖动
                                         滑动等等
              事件分类
                                         摇晃
                         2.加速计事件
                                         旋转,陀螺仪
                         3.远程控制事件
                               支持事件的对象必须是继承
                                UIResponder的对象,称为响应者对象
                                               触摸事件
                                               - (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event;
                                               – (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event;
                                               – (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event;
                                               – (void)touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)event;
                               处理事件的方法
                                               加速计事件
                                               - (void)motionBegan:(UIEventSubtype)motion withEvent:(UIEvent *)event;
                                               – (void)motionEnded:(UIEventSubtype)motion withEvent:(UIEvent *)event;
                                               – (void)motionCancelled:(UIEventSubtype)motion withEvent:(UIEvent *)event;
                                               远程控制事件
                                               – (void)remoteControlReceivedWithEvent:(UIEvent *)event;
                                             开始:
              触摸响应者对象
                                             UIButton
              (UIResponder)
                                             UIView
                               响应链-流程
                                                    --- UIViewController
                                             UIView
                                             UIWindow
                                             UIApplication - - - ▶ UIApplicationDelegate - - - ▶ nil 结束
                                           响应者链从最外层操作对象(比如一个Button)一直延伸到AppDelegate,一个响应者的
                                           规则是如果有父视图则nextResponder指向父视图,如果是控制器根视图则指向控制
                                          ·器,控制器如果在导航控制器中则指向导航控制器的相关显示视图最后指向导航控制
                                响应者链 ---▶
                                           器,如果是根控制器则指向UIWindow,UIWindow的nexResponder指向UIApplication最
                                           后指向AppDelegate,而他们实现这一套指向都是靠重写nextReponder实现的。
                                加速度计、陀螺仪、磁强计都不属于响应者链。而是由CoreMotion
                                传递事件给你指定的对象。Core Motion
                                                       2.系统从底层开始寻找到该按钮
                                                                                 3.找到按钮后,处理操作事件方
                                                                                 法,然后判断父视图是否需要处
                         响应事件流程简述
                                                       (UIApplication开始遍历所有子
                                          1点击按钮
                                                       视图,直到最外层Button)
                                                                                 理事件,直到底层
                                                 只有继承了UIResponder的对象才能接受
                                                 并处理事件,我们称之为"响应者对象"
                                                 UIApplication, UIWindow, UIViewController, UIView
                         首先是:响应者(Responder)
                                                 等等(大部分UIKit类)都是UIResponder的子类
                                                 是响应者,不代表一定能接收事件,比如
响应链机制
                                                 UllmageView默认是不能接收事件的
                                                        userInteractionEnabled = NO
                                                         隐藏: isHidden = YES
                                                         透明度<0.01
                                          不能接收事件
                         其次:能接收事件
                                                                          如果在frame之外
                                                                          了,hitTest:withEvent方法将忽
                                                         位置超过视图边界
                                                                          略这个视图和它的所有子视图
                                          其他情况的响应者一般都可以
                                          接收事件
                                                点击屏幕后,从APP的底层
                                                                          从底层到顶层寻找触发事件的对象
                                        传递 ◀--- UlApplication 一层层 到最顶层 ◀--▶
                                                                          UIApplication -> .. -> UIButton
                                                找到按钮Button的过程
                                                传递到顶层按钮后, 先处理按钮的
                                                click事件, 再判断其父类是否需
                                                                           . 找到触发对象后,开始处理事件,直到底层
                                        响应 ---▶
                                                要处理事件,如不不需要就停止,
                                                                            响应者:UIButton -> .. ->UIApplication
              触摸事件
                                                否则一层一层到UIApplication
                                                                               -(UIView *)hitTest:(CGPoint)point withEvent:(UIEvent *)event
                         事件的传递和
                         响应的区别?
                                                                  能否接收事件
                                                                               视图hitTest返回nil,则不会向其
                                                                               子视图继续传递了
                                                         传递
                                                                                -(BOOL)pointInside:(CGPoint)point withEvent:(UIEvent *)event
                                        会调用哪些方法?
                                                                 坐标是否在视
                                                                                视图pointInside返回NO,则不会
                                                                  图范围之内
                                                                                向其子视图继续传递了
                                                                 touche事件
                                                                              比如touchesBegan等等
                                                         响应
                                                                  其他创建的点击事件
                                                                                    比如Button的addTarget添加click:方法
                                                                                                                   等等其他
                                                                   添加Button
                                                                                 - -(viod)click:(id) sener { }
                                    1. 主界面: ViewController - - - ▶ view
                                                                   添加ImageView
                                                                     由UIApplication -> UIButton 逐个调用hitTest:和pointInside:判断是否继续
                         示例分析
                                    2. 点击 Button按钮 ---▶ 事件开始传递 ---▶
                                                                     (会把视图的所有子视图都遍历一遍)
                                                                                如果有[super ..]方法,则继续处
                                                                                                         响应断开,或者直到
                                    直到 Button按钮 --- 开始响应, 处理按钮的click:事件
                                                                                理父视图的响应事件
                                                                                                         UIApplication为止
                                                                当事件传递到某个控件的时候,系统就会调用该控件的此方法
                                                                遍历子控件有就返回,没有就返回self,返回nil则表示需要父视图处理,其
                                                                他子视图都不能处,hit:withEvent:方法底层会调用pointInside:withEvent:方
                                            hitTest: withEvent:
                                                                法判断点在不在方法调用者的坐标系上,不在返回nil
                         事件传递涉及的方法
                                                                如果重新该方法,可以指定某个视图来处理事件
                                                                判断点在不在当前view坐标系上,返回NO说
                                            pointInside:withEvent:
                                                                明当前对象不能处理事件
                            加速度计、陀螺仪、磁强计都不属于响应者链,
                            是由CoreMotion传递事件给你指定的对象
                            控件直接与它相关的target对象使用action消息通信。当用户与控件交互时,控件
              运动事件相关
                            调用target对象的action方法,换句话说,控件发送action消息到目标对象。Action
                            消息不是事件,但是它仍然可以利用响应链。当控件的target对象为nil,UlKit从
                            target对象和响应链走,直到找到一个对象实现了合适的action方法
```

点击,双击等