iOS事件传递及响应过程 - By Dorayo

`---》事件到来---》事件分发---》事件响应`

事件 (Events)

官方文档 【Events (iOS)】

1. `触摸事件（Touch Events）` （单点触摸、多点触摸及各种手势）
2. 晃动事件 (Motion Events) （重力、加速度等传感器）
3. 远程控制事件(Remote-Control Events) (线控耳机、airplay等)

事件分发 (Event Delivery)

触摸->硬件中断->UIKit封装成UIEvent对象(针对触摸时间)->当前运行的应用程序的事件队列->UIApplication对象->key window->Hit-Testing View 或 First Responder

目的：

> *The ultimate goal of these event paths is to find an object that can handle and respond to an event. Therefore, UIKit first sends the event to the object that is best suited to handle the event. For touch events, that object is the hit-test view, and for other events, that object is the first responder.*

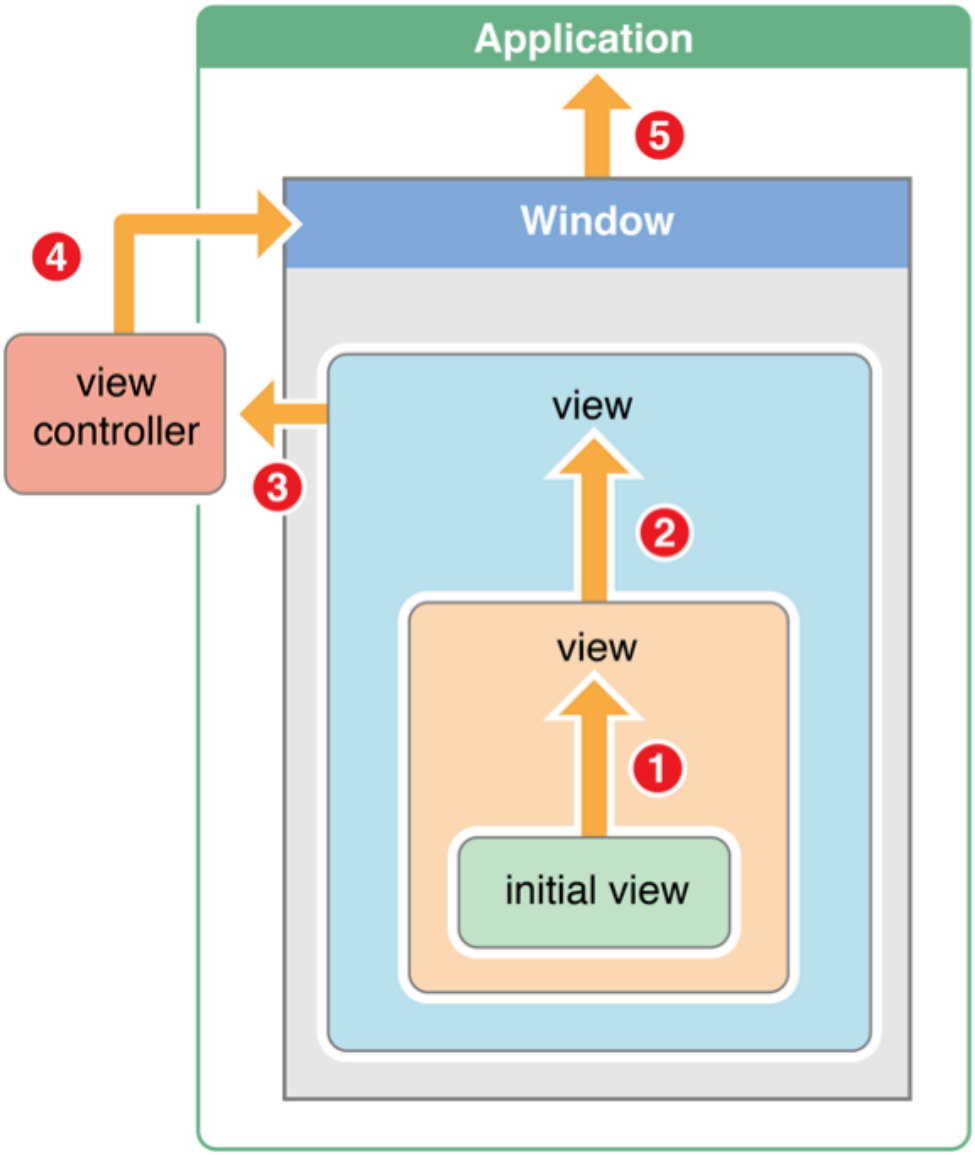
方法：

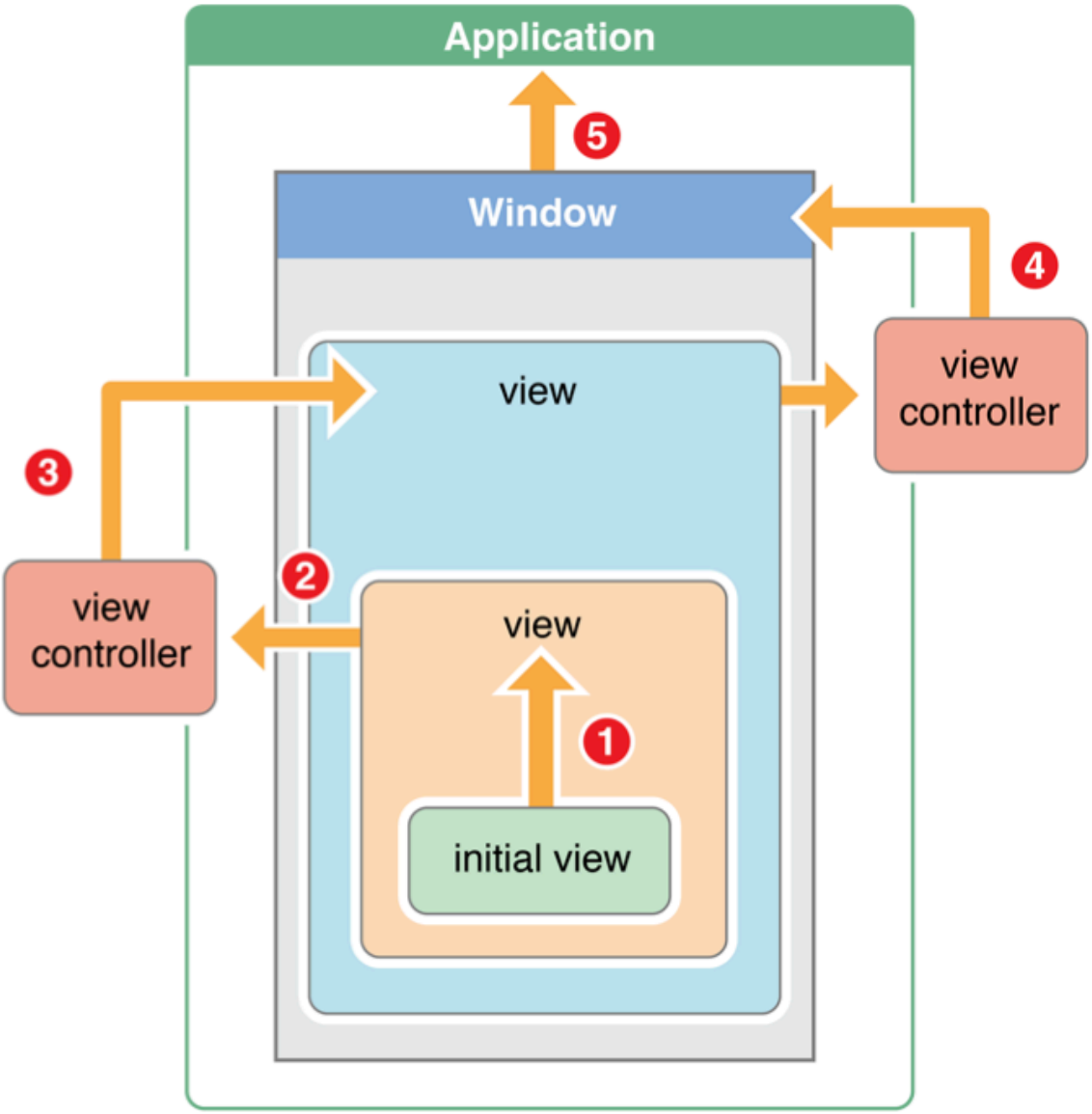
- Hit-Testing Returns the View Where a Touch Occurred （**Touch**）

The hit-test view is given the first opportunity to handle a touch event. If the hit-test view cannot handle an event, the event travels up that view's chain of responders as described in "`The Responder Chain Is Made Up of Responder Objects`" until the system finds an object that can handle it.

- The Responder Chain Is Made Up of Responder Objects (**Motion Or R-C**)

事件响应


Application
5
Window
4
view
controller
3
view
2
view
1
initial view



注意：

> *hitTest:withEvent:方法将会忽略隐藏(hidden=YES)的视图，禁止用户操作(userInteractionEnabled=YES)的视图，以及alpha级别小于0.01(alpha<0.01)的视图。如果一个子视图的区域超过父视图的bound区域(父视图的clipsToBounds 属性为NO，这样超过父视图bound区域的子视图内容也会显示)，那么正常情况下对子视图在父视图之外区域的触摸操作不会被识别,因为父视图的pointInside:withEvent:方法会返回NO,这样就不会继续向下遍历子视图了。当然，也可以重写pointInside:withEvent:方法来处理这种情况。*