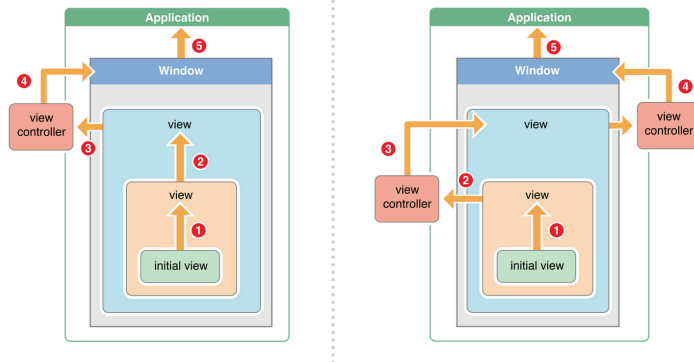


## iOS hitTest-点击事件分发分析

的解释:

Figure 2-2 The responder chain on iOS



For the app on the left, the event follows this path:

1. The **initial view** attempts to handle the event or message. If it can't handle the event, it passes the event to its **superview**, because the initial view is not the top most view in its view controller's view hierarchy.
2. The **superview** attempts to handle the event. If the superview can't handle the event, it passes the event to its superview, because it is still not the top most view in the view hierarchy.
3. The **topmost view** in the view controller's view hierarchy attempts to handle the event. If the topmost view can't handle the event, it passes the event to its view controller.
4. The **view controller** attempts to handle the event, and if it can't, passes the event to the window.
5. If the **window object** can't handle the event, it passes the event to the **singleton app object**.
6. If the **app object** can't handle the event, it discards the event.

那么,UIView是如何判定这个事件是否是自己应该处理的呢? iOS系统检测到一个触摸操作时会打包一个UIEvent对象,并放入Application的队列,Application从队列中取出事件后交给UIWindow来处理,UIWindow会使用hitTest:withEvent:方法来递归的寻找操作初始点所在的view,这个过程成为`hit-test view`.

继续看文档中的说明:

The hitTest:withEvent: method returns the hit test view for a given CGPoint and UIEvent. The hitTest:withEvent: method begins by calling the pointInside:withEvent: method on itself. If the point passed into hitTest:withEvent: is inside the bounds of the view, pointInside:withEvent: returns YES. Then, the method recursively calls hitTest:withEvent: on every subview that returns YES.

If the point passed into hitTest:withEvent: is not inside the bounds of the view, the first call to the pointInside:withEvent: method returns NO, the point is ignored, and hitTest:withEvent: returns nil. If a subview returns NO, that whole branch of the view hierarchy is ignored, because if the touch did not occur in that subview, it also did not occur in any of that subview's subviews. This means that any point in a subview that is outside of its superview can't receive touch events because the touch point has to be within the bounds of the superview and the subview. This can occur if the subview's clipsToBounds property is set to NO.

翻译下,差不多就是:

1. 首先调用当前视图的pointInside:withEvent:方法判断触摸点是否在当前视图内;
2. 若pointInside:withEvent:方法返回NO,说明触摸点不在当前视图内,则当前视图的hitTest:withEvent:返回nil.
3. 若pointInside:withEvent:方法返回YES,说明触摸点在当前视图内,则遍历当前视图的所有子视图(subviews),调用子视图的hitTest:withEvent:方法重复前面的步骤,子视图的遍历顺序是从top到bottom,即从subviews数组的末尾向前遍历,直到有子视图的hitTest:withEvent:方法返回非空对象或者全部子视图遍历完毕.
4. 若第一次有子视图的hitTest:withEvent:方法返回非空对象,则当前视图的hitTest:withEvent:方法就返回此对象,处理结束;
5. 若所有子视图的hitTest:withEvent:方法都返回nil,则当前视图的hitTest:withEvent:方法返回当前视图自身(self).

处理过程:

