

# HenCoder Plus 讲义

---

## 触摸反馈原理全解析、手写触摸反馈算法

---

### 自定义单 View 的触摸反馈

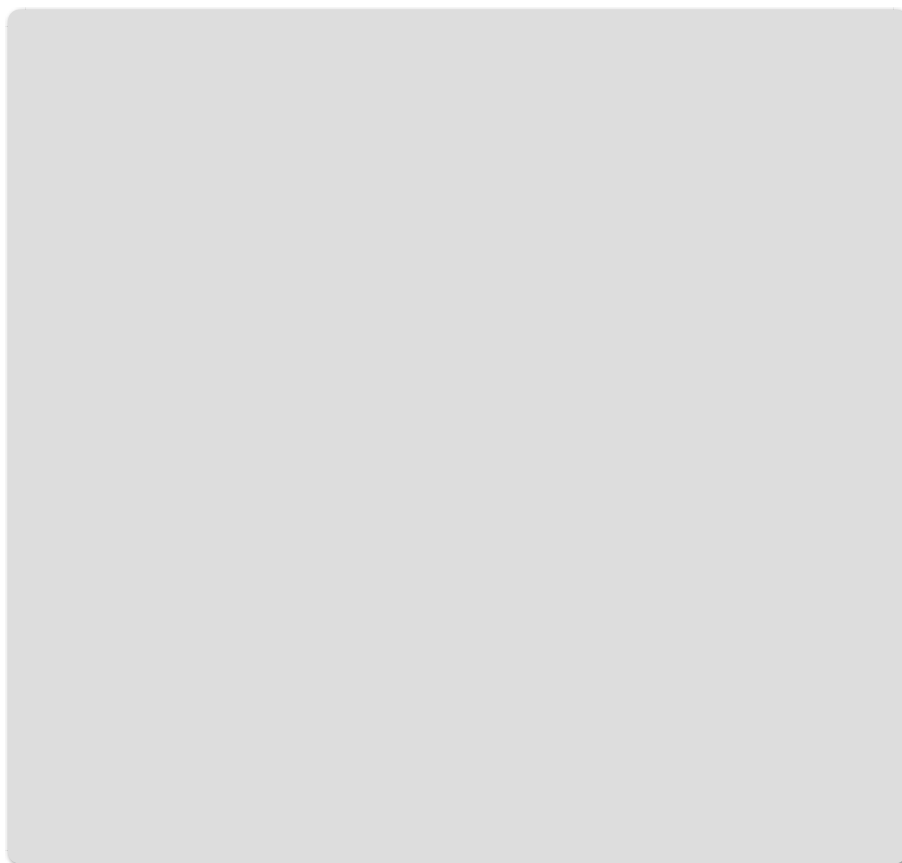
- 重写 onTouchEvent(), 在方法内部定制触摸反馈算法
  - 是否消费事件取决于 ACTION\_DOWN 事件是否返回 true
  - MotionEvent
    - getActionMasked() 和 getAction() 怎么选?
      - 选 getActionMasked()。因为到了多点触控时代, getAction() 已经不够准确
      - 那为什么有些地方 (包括 Android 源码里) 依然在用 getAction()? 因为它们的场景不考虑多点触控
    - POINTER\_DOWN / POINTER\_UP: 多点触控时的事件
      - getActionIndex(): 多点触控时用到的方法
      - 关于 POINTER\_DOWN、POINTER\_UP 和 getActionIndex(), 后面多点触控的课程里会详细讲

### View.onTouchEvent() 的源码逻辑

- 当用户按下 (ACTION\_DOWN) :
  - 如果不在滑动控件中, 切换至按下状态, 并注册长按计时器
  - 如果在滑动控件中, 切换至预按下状态, 并注册按下计时器
- 当进入按下状态并移动 (ACTION\_MOVE) :
  - 重绘 Ripple Effect
  - 如果移动出自己的范围, 自我标记本次事件失效, 忽略后续事件
- 当用户抬起 (ACTION\_UP) :

- 如果是按下状态并且未触发长按，切换至抬起状态并触发点击事件，并清除一切状态
- 如果已经触发长按，切换至抬起状态并清除一切状态
- 当事件意外结束（ACTION\_CANCEL）：
  - 切换至抬起状态，并清除一切状态

Tool Tip: 新版 Android 加入的「长按提示」功能。



## 自定义 ViewGroup 的触摸反馈

- 除了重写 onTouchEvent()，还需要重写 onInterceptTouchEvent()
- onInterceptTouchEvent() 不用在第一时间返回 true，而是在任意一个事件里，需要拦截的时候返回 true 就行
- 在 onInterceptTouchEvent() 中除了判断拦截，还要做好拦截之后的工作的准备工作（主要和 onTouchEvent() 的代码逻辑一致）

## 触摸反馈的流程



- Activity.dispatchTouchEvent()
  - 递归: ViewGroup(View).dispatchTouchEvent()

- ViewGroup.onInterceptTouchEvent()
- child.dispatchTouchEvent()
- super.dispatchTouchEvent()
  - View.onTouchEvent()
- Activity.onTouchEvent()

## View.dispatchTouchEvent()

- 如果设置了 onTouchListener，调用 onTouchListener.onTouch()
  - 如果 onTouchListener 消费了事件，返回 true
  - 如果 onTouchListener 没有消费事件，继续调用自己的 onTouchEvent()，并返回和 onTouchEvent() 相同的结果
- 如果没有设置 onTouchListener，同上

## ViewGroup.dispatchTouchEvent()

- 如果是用户初次按下 (ACTION\_DOWN)，清空 TouchTargets 和 DISALLOW\_INTERCEPT 标记
- 拦截处理
- 如果不拦截并且不是 CANCEL 事件，并且是 DOWN 或者 POINTER\_DOWN，尝试把 pointer（手指）通过 TouchTarget 分配给子 View；并且如果分配给了新的子 View，调用 child.dispatchTouchEvent() 把事件传给子 View
- 看有没有 TouchTarget
  - 如果没有，调用自己的 super.dispatchTouchEvent()
  - 如果有，调用 child.dispatchTouchEvent() 把事件传给对应的子 View（如果有的话）
- 如果是 POINTER\_UP，从 TouchTargets 中清除 POINTER 信息；如果是 UP 或 CANCEL，重置状态

## TouchTarget

- 作用：记录每个子 View 是被哪些 pointer（手指）按下的
- 结构：单向链表

## 问题和建议？

课上技术相关的问题，都可以去群里和大家讨论，对于比较通用的、有价值的问题，可以去我们的知识星球提问。

具体技术之外的问题和建议，都可以找丢物线（微信：diuwuxian），丢丢会为你解答技术以外的一切。



## 觉得好？

如果你觉得课程很棒，欢迎给我们好评呀！<https://ke.qq.com/comment/index.html?cid=381952>

一定要是你真的觉得好，再给我们好评。不要仅仅因为对丢物线的支持而好评（报名课程已经是你最大的支持了，再不够的话 B 站多来点三连我也很开心），另外我们也坚决不做好评返现等任何的交易。我们只希望，在课程对你有帮助的前提下，可以看到你温暖的评价。

## 更多内容：

- 网站：<https://hencoder.com>；<https://kaixue.io>
- 各大搜索引擎、微信公众号、微博、知乎、掘金、哔哩哔哩、YouTube、西瓜视频、抖音、快手、微视：统一账号「扔物线」，我会持续输出优质的技术内容，欢迎大家关注。
- 哔哩哔哩快捷传送门：<https://space.bilibili.com/27559447>

大家如果喜欢我们的课程，还请去扔物线的哔哩哔哩，帮我素质三连，感谢大家！