Handling Touch Events in ViewGroup

Android.

ViewGroup

- A ViewGroup is a special view that can contain other views (called children.)
- ViewGroup is base class for *layouts*, are invisible containers that hold other Views (or other ViewGroups) and define their layout properties.
- Examples: Linear,Relative,Grid Layout,etc.
- Why Extending viewgroup? Complex Layout,with complex touch events.

Well, Lets talk about Touch Events

- dispatchTouchEvent(MotionEvent)- Pass the touch screen motion event down to the target view, or this view if it is the target.
- ViewGroup.onInterceptTouchEvent(MotionEvent) allows a ViewGroup to watch events as they are dispatched to child Views.
- ViewParent.requestDisallowInterceptTouchEvent (boolean) - Call this upon a parent View to indicate that it should not intercept touch events

Structure of methods

```
public boolean dispatchTouchEvent(MotionEvent me){
    return super.dispatchTouchEvent(me);
}
```

- if call super.dispatchTouchEvent(), then layout/view handle that touch event (ie, passing it to its child/view).
- If you don't then you have to dispatch the touch event to child/view by yourself.
- returning boolean means: true = dispatch events to child/view, false don't dispatch event to a view/child.
- If you want to block all touch events to a view, override this method in it's parent view and return false.

Only available in parent/ViewGroup

```
public boolean onInterceptTouchEvent(MotionEvent event){
    return super.onInterceptTouchEvent(event);
}
```

- Normally, this method is overridden in parent/ViewGroup to intercept touch events of its child, before child gets it.
- calling super method, parent/viewGroup handles the touch event by himself.
- returning boolean in parent/ViewGroup method : true means the touch event is intercepted by parent (child won't get any events afterwards), false means events are not intercepted.

```
public boolean onTouchEvent(MotionEvent ev){
    return super.onTouchEvent(ev);
}
```

- Normally we override this method to handle touch events of a view by ourselves using GestureDetector,Gesturelistener etc.
- Calling super method, view itself handle the touch events.
- returning boolean : true means we handled the touch event, false means we didn't (it's just an notification that we handled it or not).

Touch Flow is much more explained in following slides.

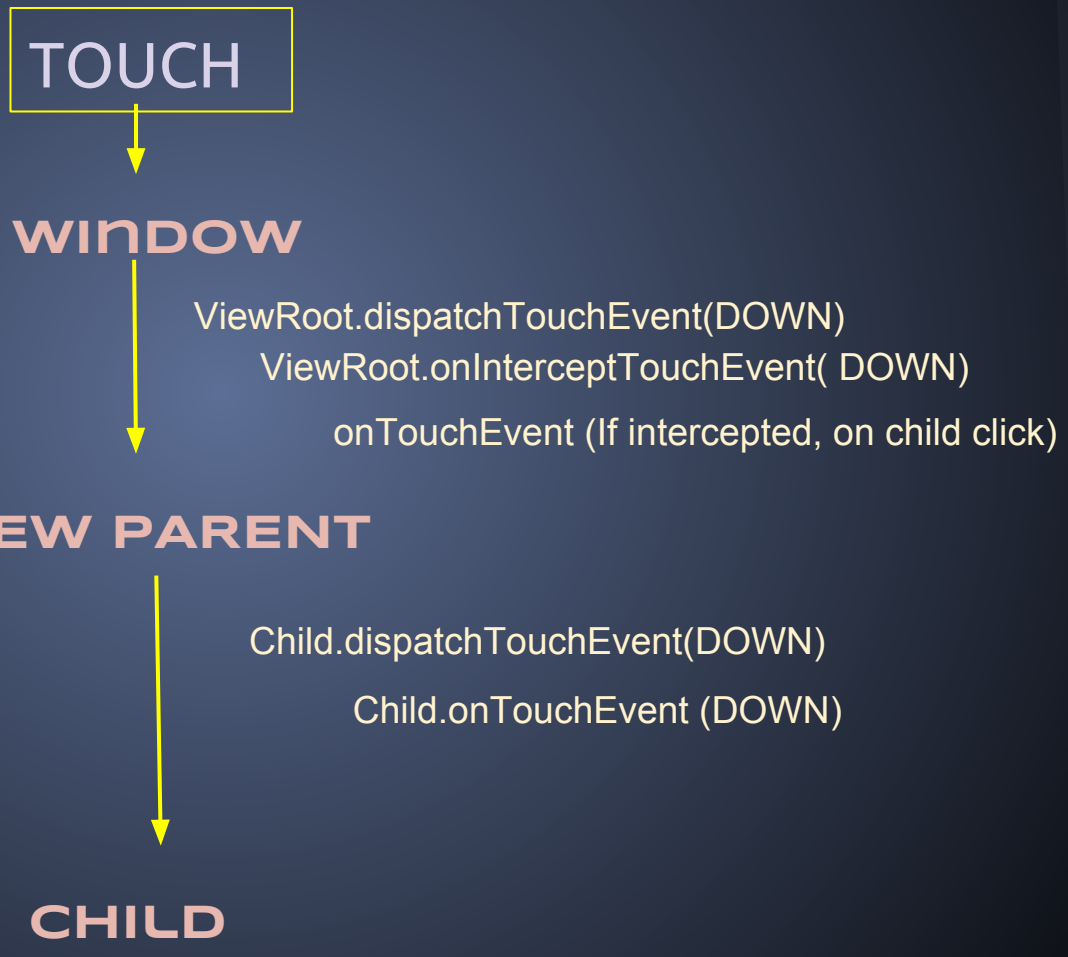
Touch Flow:-

Touch events always flow from Parent ->Child (w/o overriding any methods)

DEFAULT:

==========

Without overriding any methods.



Example: TOUCHING ON LAYOUT

DOWN -> MOVE -> UP event (Child won't get any event)

Events in order

1. Layout.dispatchTouchEvent(Down)
2. Layout.onInterceptTouchEvent(Down)
   (ViewGroup always get down event)
3. Layout.onTouchEvent(Down)
4. Layout.dispatchTouchEvent(Move)
5. Layout.onTouchEvent(Move)
6. Layout.dispatchTouchEvent(Up)
7. Layout.onTouchEvent(Up)

Example: TOUCHING ON CHILD VIEW

DOWN -> MOVE -> UP event

Events in order: (Layout means parent ViewGroup)

1. Layout.dispatchTouchEvent(Down)
2. Layout.onInterceptTouchEvent(Down)

3. Child.dispatchTouchEvent(Down)
4. Child.onTouchEvent(Down)

==========

5. Layout.dispatchTouchEvent(Move)
6. Layout.onInterceptTouchEvent(Move)

(continue..)

7. Child.dispatchTouchEvent(Move)
8. Child.onTouchEvent(Move)

==========

9. Layout.dispatchTouchEvent(Up)
10. Layout.onInterceptTouchEvent(Up)

11. Child.dispatchTouchEvent(Move)
12. Child.onTouchEvent(Move)

dispatchTouchEvent returns false ?

By Layout

Touching on Layout

- Will Get dispatchTouchEvent(me),then onInterceptTouchEvent(Down only), onTouchEvent(me).

Touching on Child

- Child won't get any events, only layout.

(me= any MotionEvent)

Conclusion

(Event won't be dispatched to child views)If you want events not to be dispatched to child views, return false in dispatchTouchEvent method.

onInterceptTouchEvent returns true

By Layout

Touching on Layout

- Doesn't affect child, bcoz its not child event

Touching on Child

- Intercepted @ DOWN event
  - Layout gets ALL event, child gets nothing.
- Intercepted @ MOVE event
  - Layout & child get DOWN, layout get MOVE & dispatches CANCEL event to child, layout gets rest of events.

- Intercepted @ MOVE event
  - Layout & Child will get DOWN,MOVE event, layout get UP & intercepts, dispatches CANCEL event to child.

CONCLUSION

When viewParent intercepts an events, it dispatches CANCEL event to child, takes ownership of rest of Touch events.

onTouchEvent returns true ?

By Layout

Touching on Layout

- Works normal, gets all events.

Touching on Child

- Child gets all events, but viewParent (Layout) can only spy the events(get only onDispatch & onIntercept Touch events).

Conclusion

Works just like without overriding any methods.

onInterceptTouchEvent returns false

By Layout

Touching on Layout

- return false @ DOWN event
  - Gets only DOWN event(dispatch,intercept,onTouch)
- return false @ MOVE/UP event
  - Works normal like without overriding.

Touching on Child

- Works normal like without overriding.

RequestDisallowInterceptTouch=true

When it is set true, the parent can't intercept any touch events from child

Conclusion:

Will get dispatchTouchEvent() on parent, but not onInterceptTouch event.

Gesture Detection

- Detects various gestures and events using the supplied Motion Events.

- Example:-
  - create an gestureDetector

    ~~GestureDetector gd=new GestureDetector(GestureListener);~~

    (deprecated)

    or

    GestureDetector gd=new GestureDetector(context,gl);
- In onTouch,
  - gd.onTouchEvent(me).

Gesture Listener

Gesture listener is used to notify when gestures occur.

- OnDoubleTapListener
  - to notify when a double-tap or a confirmed single-tap occur.
- OnGestureListener
  - to notify when gestures occur.
- SimpleOnGestureListener
  - only want to listen for a subset of all the gestures

Example of Gestures.

- onDown(MotionEvent e)
- onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY)
- onLongPress(MotionEvent e)
- onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY)
- onShowPress(MotionEvent e)
- onSingleTapUp(MotionEvent e)