ASSIGNMENT-10

Event listeners

An event listener is an interface in the View class that contains a single callback method. These methods will be called by the Android framework when the View to which the listener has been registered is triggered by user interaction with the item in the UI.

Included in the event listener interfaces are the following callback methods:

onClick()

From View.OnClickListener. This is called when the user either touches the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses the suitable "enter" key or presses down on the trackball.

onLongClick()

From View.OnLongClickListener. This is called when the user either touches and holds the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses and holds the suitable "enter" key or presses and holds down on the trackball (for one second).

onFocusChange()

From View.OnFocusChangeListener. This is called when the user navigates onto or away from the item, using the navigation-keys or trackball.

onKey()

From View.OnKeyListener. This is called when the user is focused on the item and presses or releases a hardware key on the device.

onTouch()

From View.OnTouchListener. This is called when the user performs an action qualified as a touch event, including a press, a release, or any movement gesture on the screen (within the bounds of the item).

onCreateContextMenu()

From View.OnCreateContextMenuListener. This is called when a Context Menu is being built (as the result of a sustained "long click"). See the discussion on context menus in the Menus developer guide.

These methods are the sole inhabitants of their respective interface. To define one of these methods and handle your events, implement the nested interface in your Activity or define it as an anonymous class. Then, pass an instance of your implementation to the respective View.set...Listener() method. (E.g., call setOnClickListener() and pass it your implementation of the OnClickListener.)

Event handlers

If you're building a custom component from View, then you'll be able to define several callback methods used as default event handlers. In the document about Custom View Components, you'll learn some of the common callbacks used for event handling, including:

onKeyDown(int, KeyEvent) - Called when a new key event occurs.

onKeyUp(int, KeyEvent) - Called when a key up event occurs.

onTrackballEvent(MotionEvent) - Called when a trackball motion event occurs.

onTouchEvent(MotionEvent) - Called when a touch screen motion event occurs.

onFocusChanged(boolean, int, Rect) - Called when the view gains or losses focus.

There are some other methods that you should be aware of, which are not part of the View class, but can directly impact the way you're able to handle events. So, when managing more complex events inside a layout, consider these other methods:

Activity.dispatchTouchEvent(MotionEvent) - This allows your Activity to intercept all touch events before they are dispatched to the window.

ViewGroup.onInterceptTouchEvent(MotionEvent) - This 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 with onInterceptTouchEvent(MotionEvent).