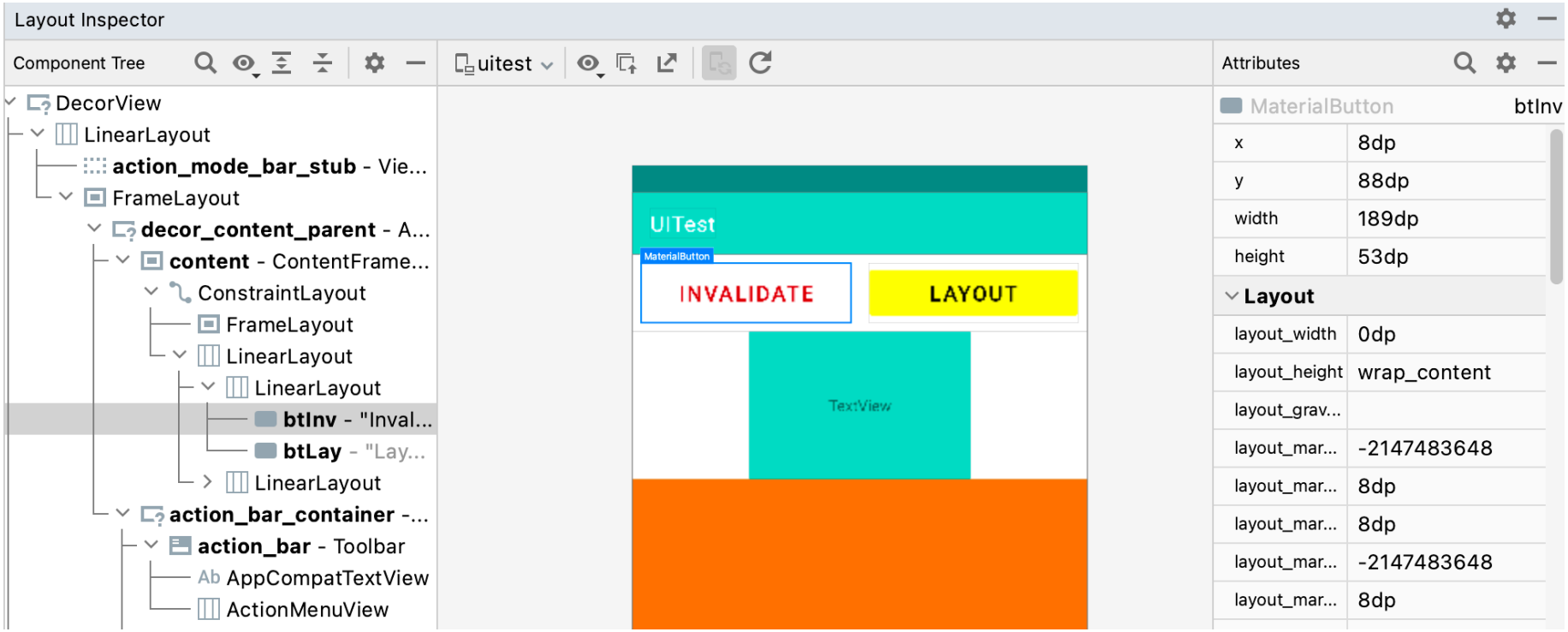
Elementi grafici ed eventi

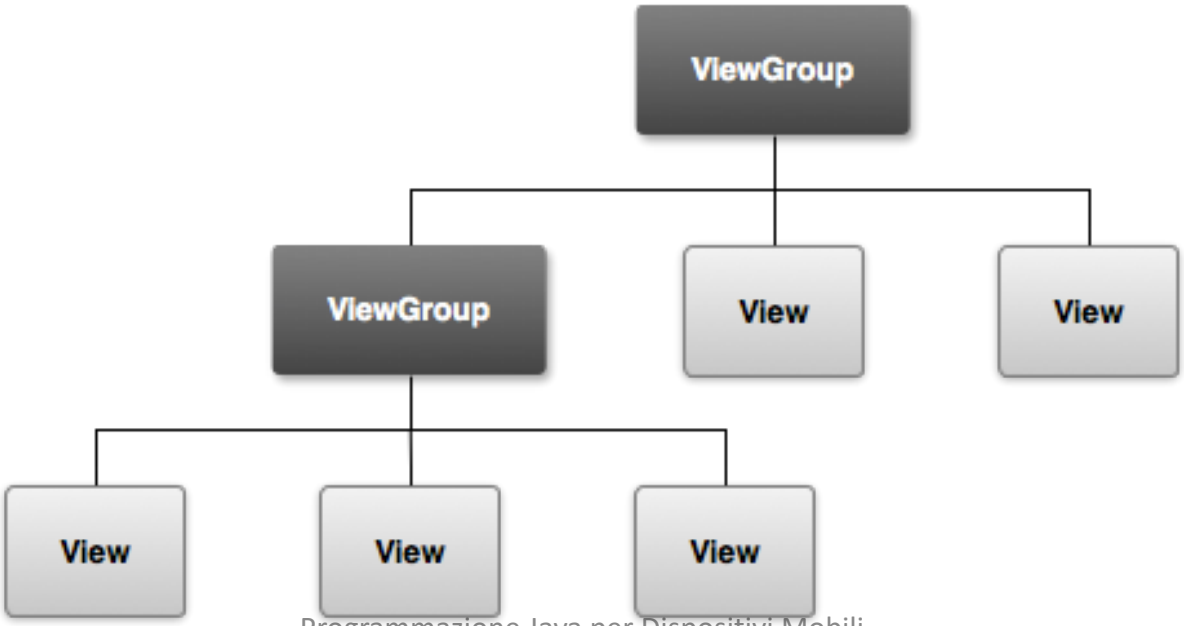
Albero delle view

- Layout Inspector



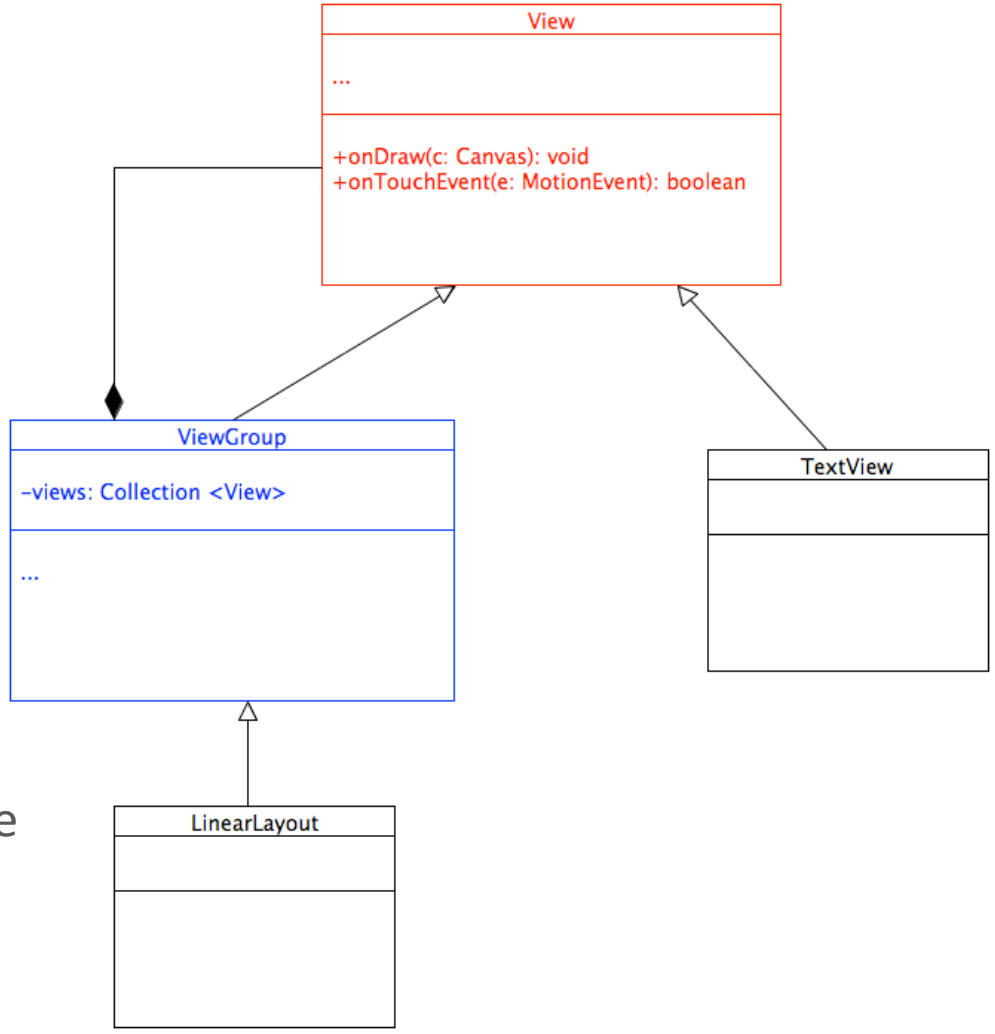
Albero delle view

- La grafica è composta da View e ViewGroup
  - Le view disegnano qualcosa sullo schermo e ci si può interagire
  - Le viewgroup sono contenitori invisibili per view e viewgroup
  - I componenti grafici formano un albero



Composite Pattern

- Le viewgroup estendono view
- Le viewgroup contengono una o più view



- Vantaggi
  - Realizzi collezioni di view
  - Non devo differenziare tra nodi e foglie
  - Posso chiamare un metodo su tutta la collezione

Layout

- Un layout è
  - una specializzazione di ViewGroup
  - posiziona le View contenute in un modo “specifico”

- Layout predefiniti
  - LinearLayout
  - RelativeLayout
  - TableLayout
  - FrameLayout
  - ConstrainLayout

XML e Java

- I file xml genera oggetti java
  - L'operazione di conversione si chiama **inflating**
  - da un layout xml genera un albero di view e viewgroup
- Per ottenere il riferimento all'oggetto uso
  - public View findViewById (int id)
  - metodo di Activity
- Inflating del root nel metodo onCreate
  - public void setContentView(int id)

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main_layout);
}
```

Attraversamento dell'albero

- Usando il composite pattern ad ogni Activity è associato l'albero delle view
  - Posso chiamare un metodo sulla radice ed a cascata viene chiamato su tutto l'albero
  - **draw!!**
  - L'attraversamento dell'albero cambia l'ordine di visualizzazione

- Android garantisce che
  - i nodi sono attraversati partendo dal root
  - i nodi più vicini al root sono attraversati prima
  - i nodi di pari livello sono attraversati secondo l'ordine di definizione

Processo di draw

- Processo distribuito in tre passi
  - Determino le misure delle view sullo schermo
  - Determino il posizionamento delle view
  - Disegno le view

- Eseguo tutto in un singolo thread per non aver problemi di contesa
  - inibisco il controllo degli oggetti grafici dagli altri thread

Misura

- Il parent chiama **measure(int, int)**
  - Viene eseguito in metodo **onMeasure**
  - Può essere chiamato più volte
- Il parent comunica alle view se la loro misura
  - UNSPECIFIED – misura libera
  - EXACTLY – misura esatta
  - AT_MOST – misura massima
- View.MeasureSpec
  - makeMeasureSpec – metodo per fornire il tipo di misura
- Anche i layout devono calcolare la propria misura in base a quella delle view
  - setMeasuredDimension

Posizionamento delle view

- Si attraversa l'albero una seconda volta per determinare la posizione delle view nota la loro dimensione

- Il parent chiama **layout(int, int, int, int)**
  - Viene eseguito in metodo onLayout
  - coordinate del vertici top-left e bottom-right

- Le informazioni circa il layout sono memorizzate in istanze della classe ViewGroup.LayoutParams
  - Ogni layout definisce la sua con i parametri specifici

Disegno delle view

- Viene chiamato su tutto l'albero il metodo **draw()**
  - questo chiama il metodo **onDraw**

- Tutte le view note le dimensioni e nota la posizione si disegnano

Ridisegnare le view

- public void **requestLayout** ()
  - richiede di ricominciare la fase di disegno partendo dal ricalcolo delle misure
- public void **invalidate** ()
  - serve a far ridisegnare le view
- public void **postInvalidate** ()
  - se sono in un thread differente da quello di grafica

Gestione Eventi

Overriding

- Per **gestire eventi** direttamente nella sottoclasse
  - View o Activity

```
public class CustomView extends View {
    public CustomView(Context context) {
        super(context);
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            Log.d("CustomView", "Touch ricevuto");
            return true; // evento gestito
        }
        return false; // passa ad altri
    }
}
```

Override di metodi di View

- Gestione di eventi di interazione con View
  - metodi da sovrascrivere
- Gestione del tocco
  - public boolean onTouchEvent (MotionEvent event)
- Gestione della pressione dei tasti.
  - boolean onKeyDown (int keyCode, KeyEvent event)
    boolean onKeyUp(int keyCode, KeyEvent event)
    boolean onKeyLongPress (int keyCode, KeyEvent event)
    public boolean onKeyPreIme (int keyCode, KeyEvent event)
    boolean onKeyMultiple (int keyCode, int repeatCount, KeyEvent event)
    public boolean onKeyShortcut (int keyCode, KeyEvent event)
    public boolean onKeyUp (int keyCode, KeyEvent event)
- Gestione del focus
  - public void onWindowFocusChanged (boolean hasWindowFocus)

Gestione degli eventi fuori dalla View

- Activity
  - public boolean **dispatchTouchEvent** (MotionEvent ev)
    - intercettare tutti gli eventi di touch prima che vengano inviati alla finestra contenuta
- ViewGroup/Layout
  - public boolean **dispatchTouchEvent** (MotionEvent ev)
  - public boolean **onInterceptTouchEvent** (MotionEvent ev)
    - permette di specificare se gli eventi di touch debbano essere elaborati nel metodo onTouch() di ognuna delle View
- Interfaccia ViewParent
  - public void **requestDisallowInterceptTouchEvent** (boolean disallowIntercept)
    - per impedire l'invocazione del precedente metodo onInterceptTouchEvent()

Listener di eventi

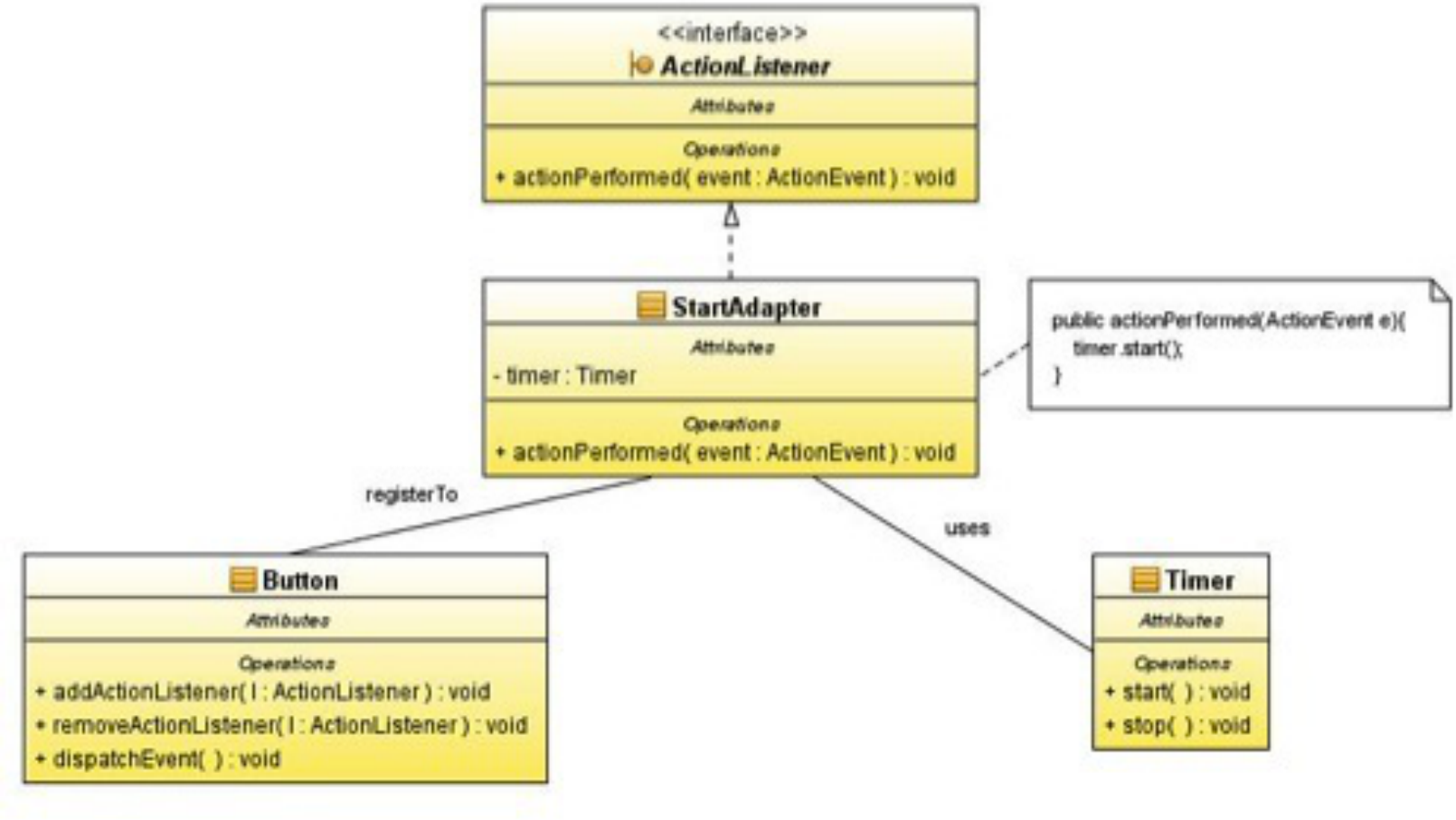
- Classe Inner Anonima:

```
Button button = new Button(getActivity());

button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Esegui azione quando il bottone viene cliccato
    }
});
```

- Lambda:

```
Button button = new Button(getActivity());

button.setOnClickListener(v -> {
    // Esegui azione quando il bottone viene cliccato
});
```

Delegation Model



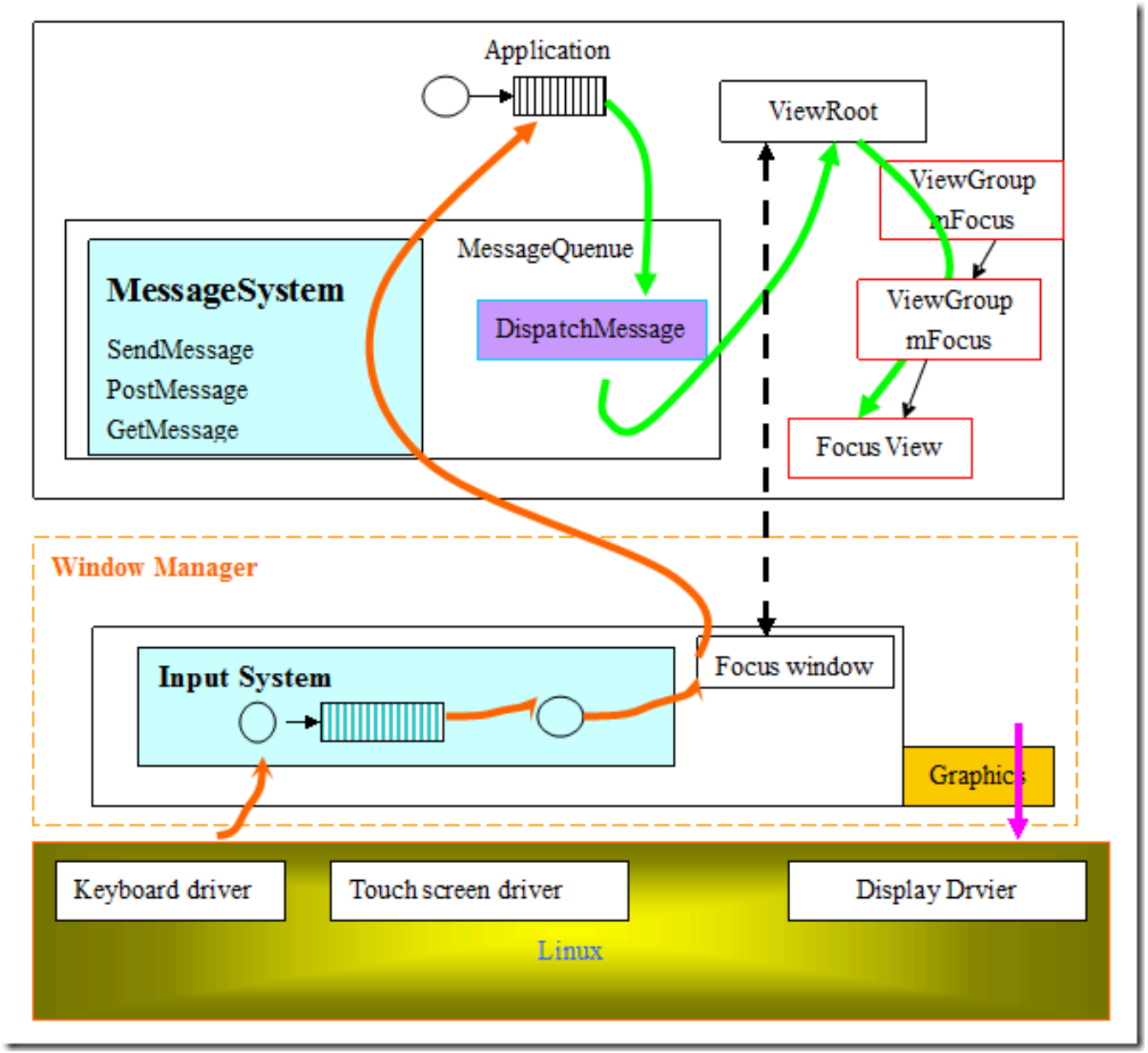
- Disaccoppiare il generatore dell'evento dal gestore

Listener della classe View

Template dei metodi setter per i Listener:

public void setOn<Evento>Listener (View.On<Evento>Listener l)

Nome evento	Interfaccia Listener	Descrizione
click	View.OnClickListener	Evento di selezione di un componente.
long click	View.OnLongClickListener	Evento di hold, ovvero di selezione prolungata di un componente.
focus change	View.OnFocusChangeListener	Evento di acquisizione o perdita del focus da parte di un componente.
key	View.OnKeyListener	Evento di selezione di un tasto.
touch	View.OnTouchListener	Evento di touch.
createContextMenu	View.OnCreateContextMenuListener	Evento di creazione del menu contestuale.


Application
ViewRoot
ViewGroup
mFocus
MessageQueue
MessageSystem
SendMessage
PostMessage
GetMessage
DispatchMessage
ViewGroup
mFocus
Focus View
Window Manager
Input System
Focus window
Graphics
Keyboard driver
Touch screen driver
Display Drvier
Linux