

Dong Nai University of Technology

Hiền Tài Là Nguyên Khí Của Quốc Gia

Bài tập 7: Làm quen với các Layout trong Android

By Trần Duy Thành on April 1, 2013

i

10 Votes

Để thiết kế giao diện tốt cho Android thì trước tiên các bạn phải biết sử dụng một số Layout cơ bản trong Android:

1) Cách tạo Layout và kết nối Layout vào Activity

2) Cách sử dụng HierarchyViewer

3) Các Layout cơ bản:

- **FrameLayout**
- **LinearLayout**
- **TableLayout**
- **RelativeLayout**
- **AbsoluteLayout**

*** Đồng thời bạn phải biết kết hợp các Layout này lại với nhau để cho ra được giao diện phức tạp như ý muốn.

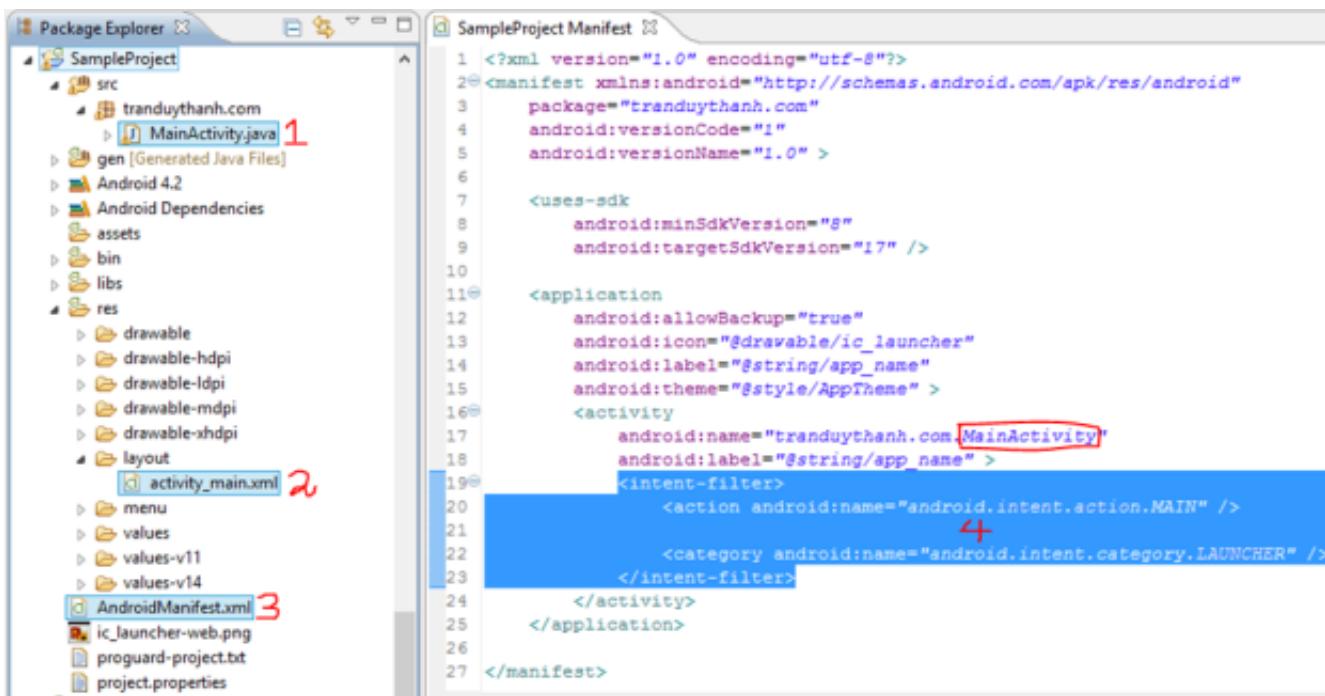
- Tôi chỉ nêu ra một số đặc điểm, không thể nói hết ở đây. Các bạn sẽ từ từ cập nhập sau khi đã quen với Layout.

- Trước tiên bạn cần biết cách tạo một Layout mới và cách kết nối nó vào Activity như thế nào:

1) Cách tạo Layout và kết nối Layout vào Activity:

a) Layout mặc định đầu tiên khi bạn tạo một Android Project:

- Khi bạn tạo một Android Project thì mặc nhiên sẽ có một Activity được chỉ định chạy đầu tiên khi bạn thực thi ứng dụng:



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_0.png)

- Bạn quan sát hình trên: Khi bạn tạo một ứng dụng Android thì mặc nhiên sẽ tạo luôn một Activity để thực thi đầu tiên khi bạn chạy ứng dụng. Ứng với một Activity thì nó sẽ có 1 Layout đi kèm. Trong hình trên thì **MainActivity.java** (số 1) sẽ có layout đi kèm là **activity_main.xml** (số 2 - và bạn nhớ luôn là activity_main này sẽ được tự động tạo ra trong thư mục gen của Android, dựa vào đây để ta kết nối Layout vào Activity).

- Như đã nói ở những bài tập trước: Mọi Activity muốn được triệu gọi thành công trong ứng dụng thì bắt buộc nó phải được khai báo trong **AndroidManifest.xml** (số 3). Bạn double click vào Manifest và nhìn vào vùng số 4, MainActivity được khai báo trong này đồng thời đăng ký là Activity sẽ khởi động lúc ứng dụng được thực thi (xem vùng Tôi bôi màu xanh phần **intent-filter**).

- Tiếp theo bạn Double – click vào tập tin **MainActivity.java**:

```

1 package tranduythanh.com;
2 import android.os.Bundle;
3 import android.app.Activity;
4 import android.view.Menu;
5 public class MainActivity extends Activity {
6
7     @Override
8     protected void onCreate(Bundle
9             savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13     public boolean onCreateOptionsMenu(Menu menu) { }
14 }
15
16
17
18
19

```

(http://duythanhcse.files.wordpress.com/2013/04/7_layout_1.png)

- Nhìn vào hàm **onCreate**, bạn thấy dòng lệnh : **setContentView(R.layout.activity_main)**; chính là dòng lệnh dùng để kết nối Layout vào Activity. Bạn nhớ là activity_main phải được tự động tạo ra trong **gen** như hình bên dưới:

```

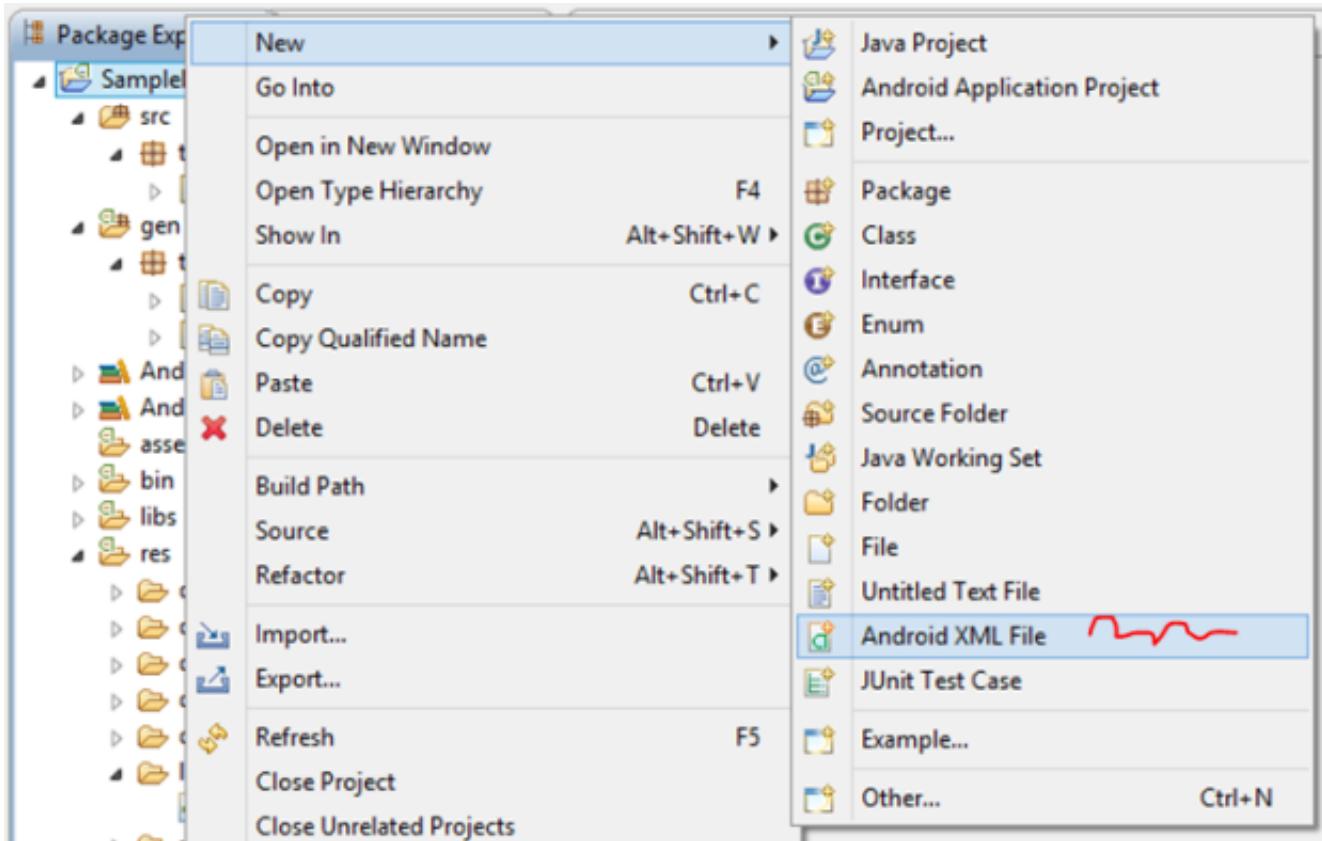
8 package tranduythanh.com;
9
10 public final class R {
11     public static final class array { }
12     public static final class attr { }
13     public static final class drawable { }
14     public static final class id { }
15     public static final class layout {
16         public static final int activity_main=0x7f030000;
17     }
18     public static final class menu { }
19     public static final class string { }
20     public static final class style { }
21 }
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70

```

(http://duythanhcse.files.wordpress.com/2013/04/7_layout_2.png)

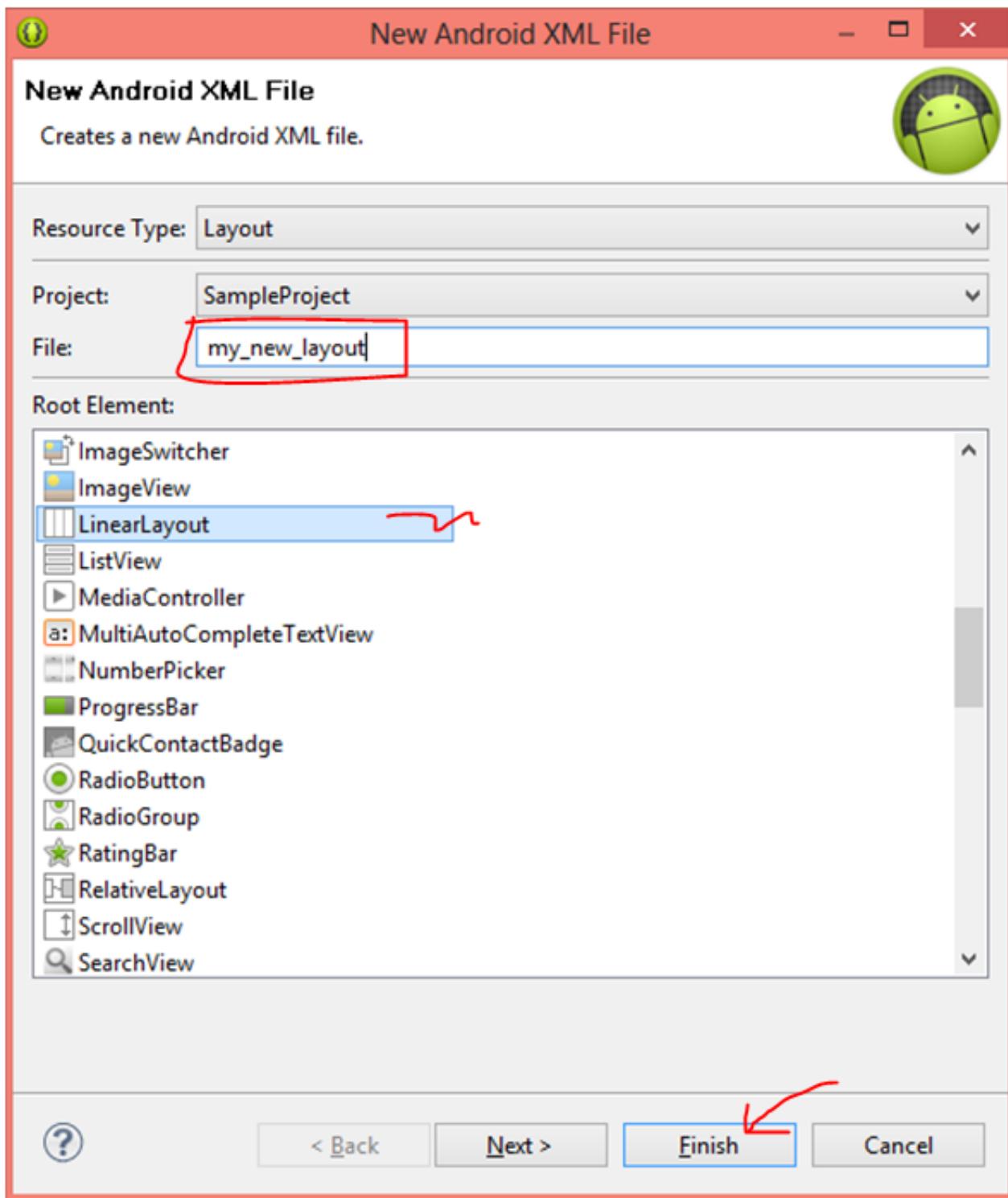
- b) Đổi Layout mặc định bằng một Layout khác bất kỳ:

- Để tạo một Layout mới, bạn click chuột phải vào **Project**/ chọn **New**/ chọn **Android XML File**:



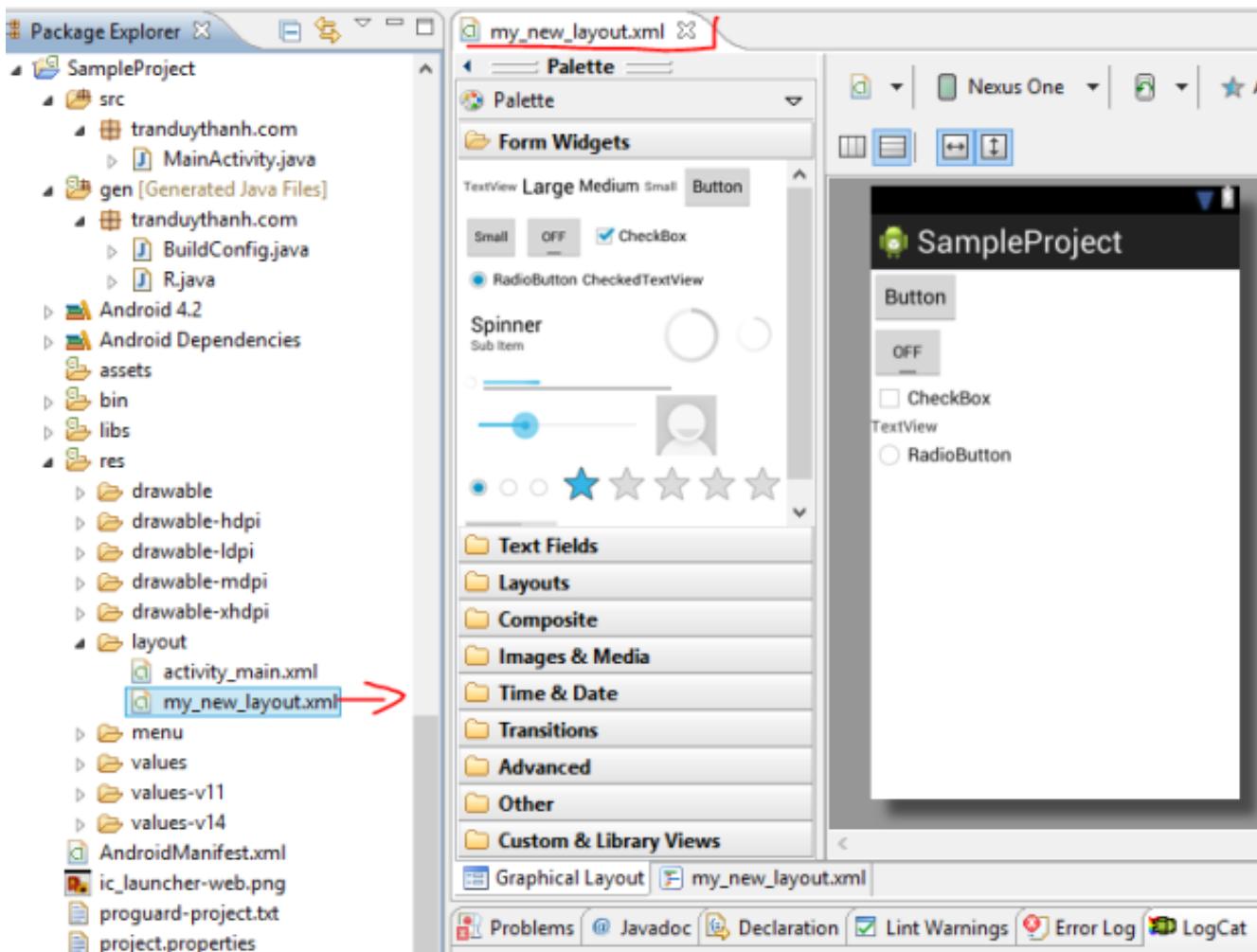
(http://duythanhcse.files.wordpress.com/2013/04/7_layout_3.png)

- Khi bạn chọn Android XML File thì màn hình bên dưới xuất hiện: Đặt tên cho Layout, chọn kiểu Layout rồi nhấn nút Finish:



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_41.png)

- Ở trên Tôi đặt tên là : **my_new_layout**, sau khi nhấn nút Finish bạn quan sát Package Explorer:



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_5.png)

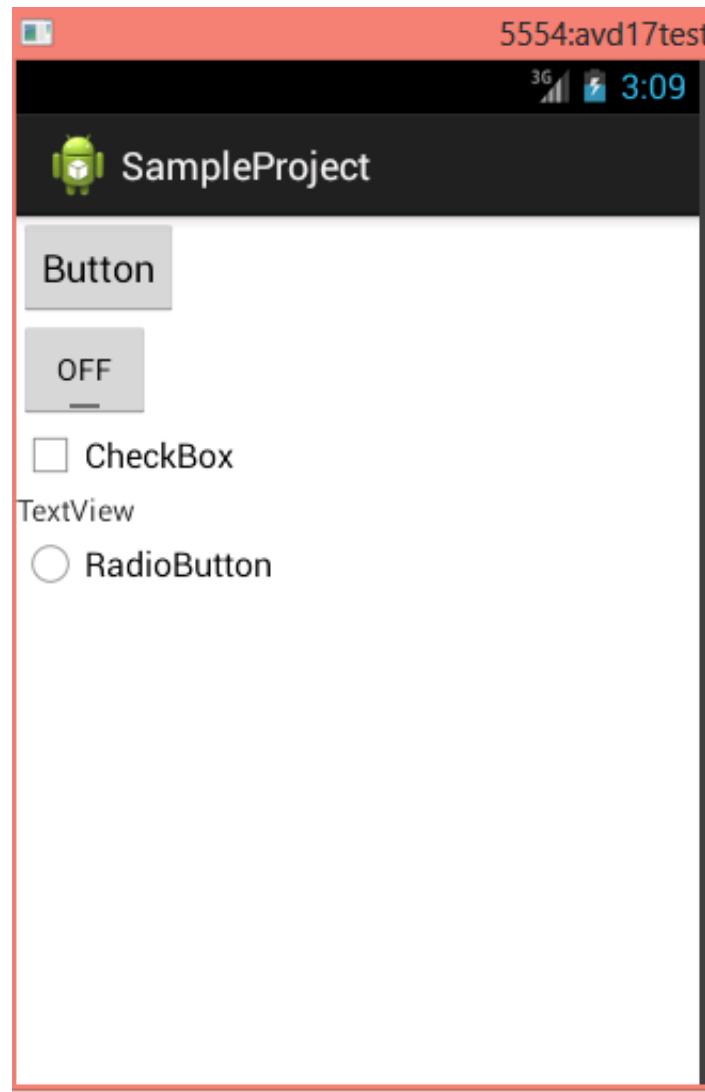
- Bạn thấy Tôi double – click vào layout : **my_new_layout** và kéo thả một số control vào giao diện như hình bên trên (bạn nhớ là **my_new_layout** phải được tự động xuất hiện trong **gen** – bạn tự kiểm tra).

- Nay bạn vào lại MainActivity.java. Sửa lại dòng lệnh setContentView thành:

```
protected void onCreate(Bundle
    savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.my_new_layout);
}
```

(http://duythanhcse.files.wordpress.com/2013/04/7_layout_6.png)

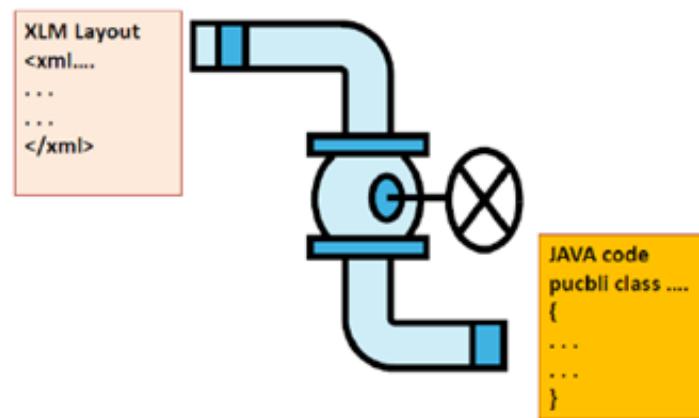
- Khởi động chương trình và bạn thấy ứng dụng sẽ chạy cái Layout mới này chứ không phải Layout cũ nữa:



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_7.png)

2) Cách sử dụng HierarchyViewer:

- Như bạn đã biết, một Layout phải được kết nối vào Activity nào đó thông qua hàm **setContentView**, Android sẽ có cơ chế dịch XML thành Java code:



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_8.png)

- Ta có thể dùng **HierarchyViewer** để hiển thị cấu trúc UI của màn hình hiện tại trên emulator hoặc thiết bị thật:

Bạn vào thư mục **SDK** của bạn / vào thư mục **Tools**/ rồi chạy tập tin **monitor.bat** :



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_9.png)

3) Các Layout cơ bản:

a) FrameLayout:

- Là loại Layout cơ bản nhất, đặc điểm của nó là khi gắn các control lên giao diện thì các control này sẽ luôn được “Neo” ở góc trái trên cùng màn hình, nó không cho phép chúng ta thay đổi vị trí của các control theo một Location nào đó.

- Các control đưa vào sau nó sẽ đè lên trên và che khuất control trước đó (trừ khi ta thiết lập transparent cho control sau):

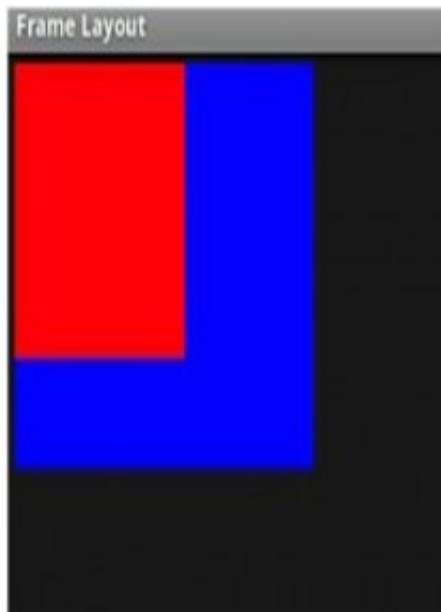
- Bạn xem đoạn cấu trúc XML dưới này:

```
<?xml version="1.0" encoding="utf-8"?>

<FrameLayout android:id="@+id/mainlayout" android:layout_height="fill_parent"
    android:layout_width="fill_parent" android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android">
</FrameLayout>

<ImageView android:layout_height="wrap_content" android:layout_width="wrap_content"
    android:padding="5px" android:src="@drawable/blue"/>
```

```
<ImageView android:layout_height="wrap_content" android:layout_width="wrap_content"  
        android:padding="5px" android:src="@drawable/red"/>  
  
</FrameLayout>
```



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_10.png)- Bạn thấy đó, hình màu đỏ và màu xanh luôn được “neo” ở góc trái màn hình. Hình màu đỏ đưa vào sau sẽ đè lên trên hình màu xanh.

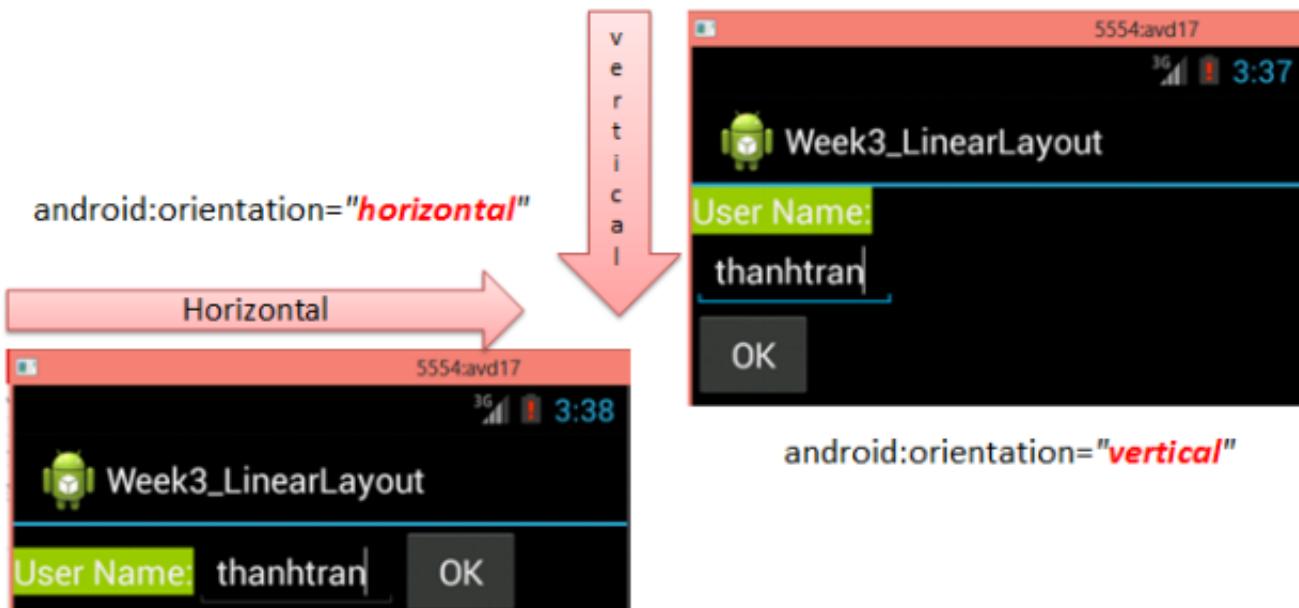
Chú ý 2 dòng lệnh bên dưới:

- android:src="@drawable/blue"
- android:src="@drawable/red"

Là do ta kéo 2 cái hình tên là **blue** và **red** vào thư mục drawable của ứng dụng

b) LinearLayout

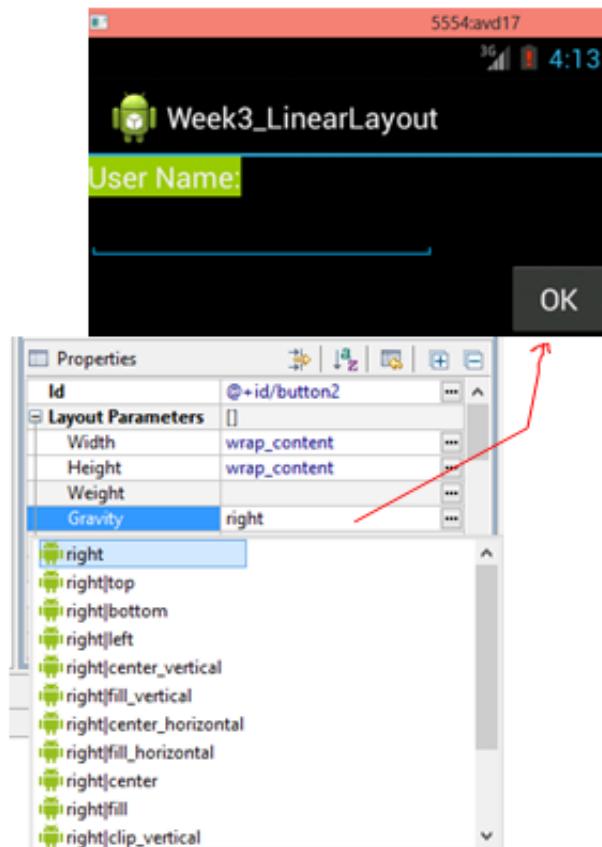
- Layout này cho phép sắp xếp các control theo 2 hướng trên giao diện: Hướng từ trái qua phải và hướng từ trên xuống dưới.



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_11.png)

- Bạn có thể dùng margin, gravity, weight để hỗ trợ cho việc thiết kế. Ở đây Tôi không có thời gian nhiều nên chỉ nói đặc điểm chính của LinearLayout, các bạn tự tìm hiểu thêm.

- Ta có thể dùng Properties hỗ trợ sẵn trong Eclipse để thiết lập các thuộc tính cho control:
- Ví dụ như để căn lề các control trên giao diện ta dùng **layout_gravity**:



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_12.png)

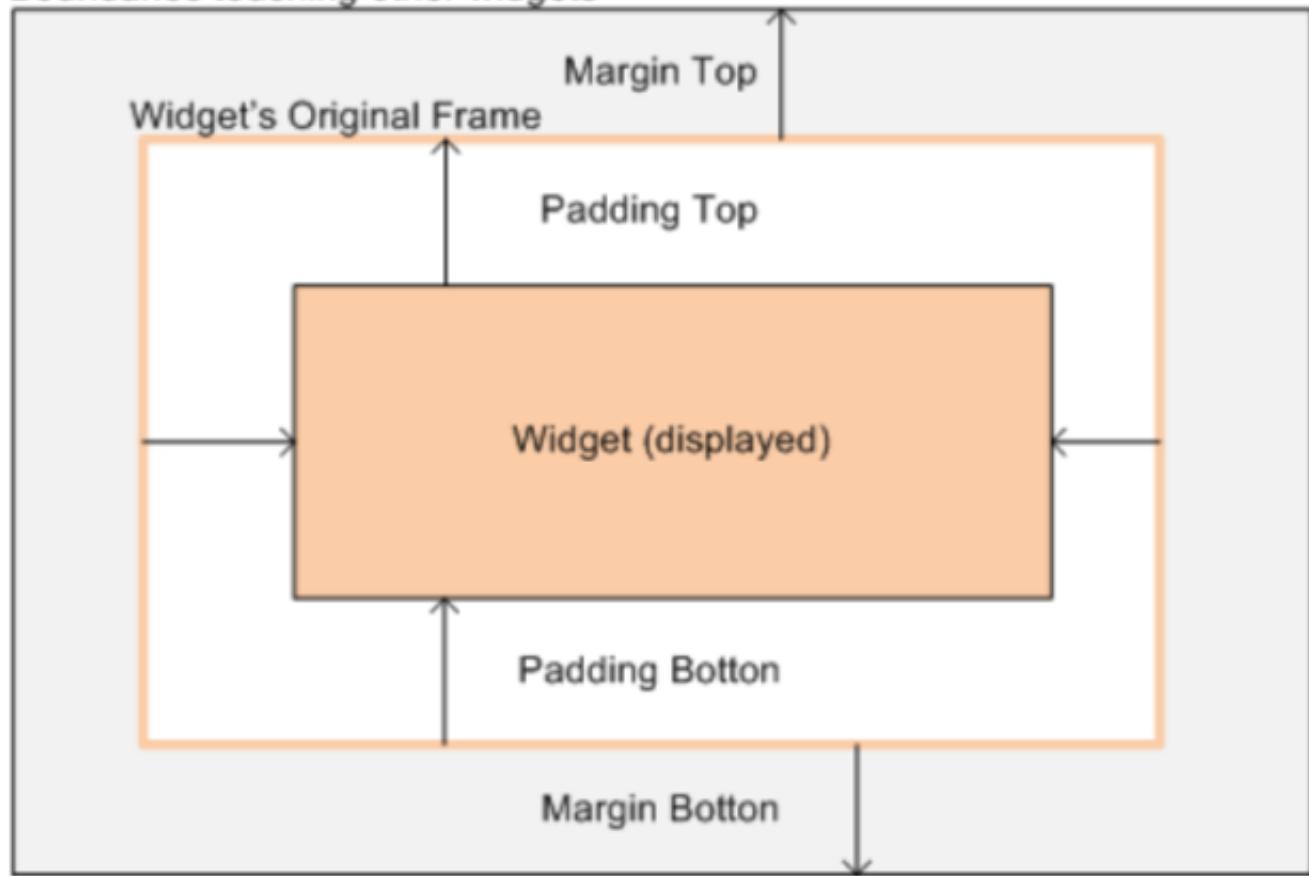
- hay để căn lề nội dung bên trong của control dùng **gravity**:



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_13.png)

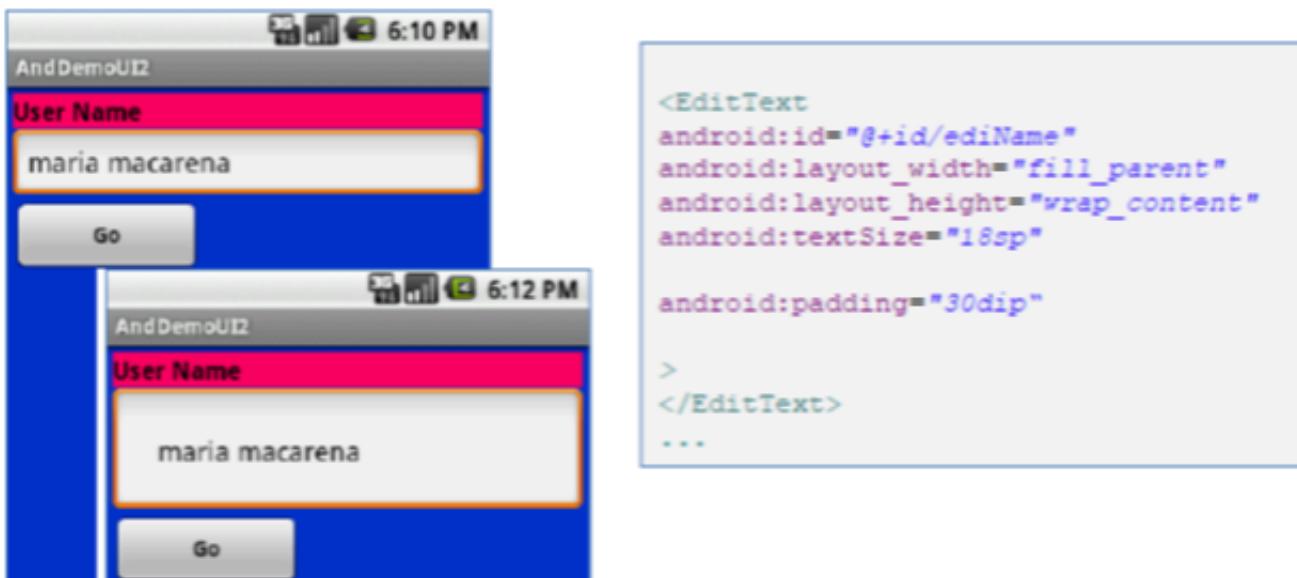
- Bạn cũng phải biết so sánh sự khác biệt giữa **Padding** và **Margin**:

Boundaries touching other widgets



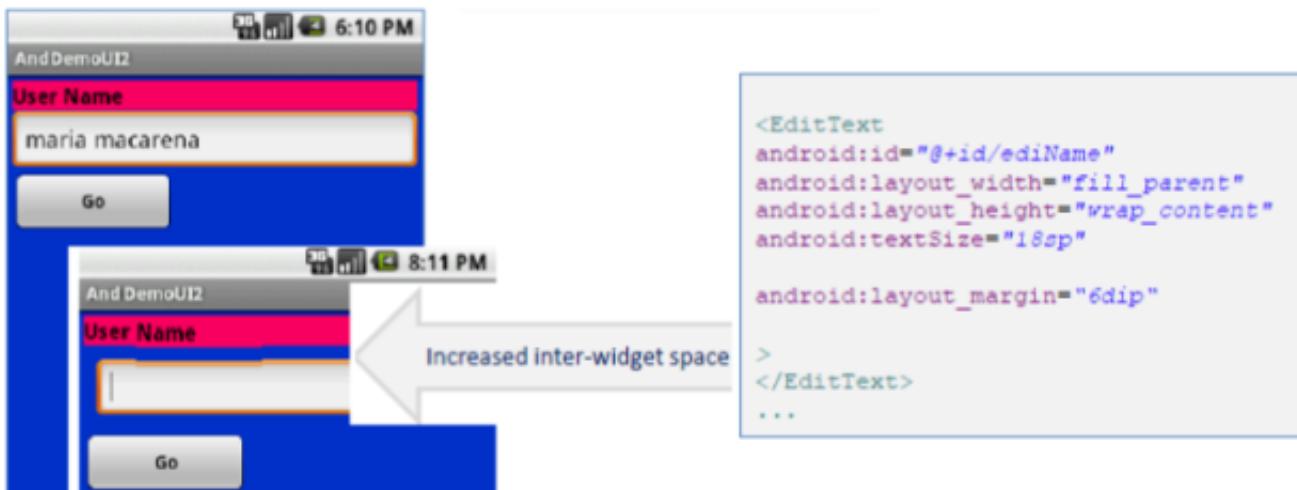
(http://duythanhcse.files.wordpress.com/2013/04/7_layout_14.png)

- Ví dụ thay đổi Padding (internal spacing – khoảng cách giữa nội dung bên trong so với đường viền của control):



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_15.png)

- Ví dụ về đổi Margin (external spacing – khoảng cách giữa control này với control khác):



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_16.png)

c) TableLayout

- Cho phép sắp các control theo dạng lưới (dòng và cột)

- TableLayout sẽ xem dòng nào có số lượng control nhiều nhất để xác định rằng nó có bao nhiêu cột (lấy dòng có số lượng control nhiều nhất làm số cột chuẩn).

0	1		
0	1		
0	1	2	3

(http://duythanhcse.files.wordpress.com/2013/04/7_layout_17.png)

- Như vậy theo hình trên thì bạn phải nói là TableLayout này có 4 cột, 3 dòng.

- Dùng **layout_span** để trộn các cột:

```
<TableRow>
    <TextView android:text="URL:" />
    <EditText
        android:id="@+id/entry"
        android:layout_span="3" />
</TableRow>
```

(http://duythanhcse.files.wordpress.com/2013/04/7_layout_19.png)

- Dùng **layout_column** để di chuyển vị trí của control đến một cột nào đó trên 1 dòng:

The diagram shows a 2x5 grid of cells. A horizontal double-headed arrow above the first three columns is labeled "android:layout_span='3'". A red arrow below the second and third columns is labeled "android:layout_columns='2'".

Label (URL)	EditText	EditText-span	EditText-span	
Column 0	Column 1	Column 2	Column 3	
		Button Cancel	Button OK	

(http://duythanhcse.files.wordpress.com/2013/04/7_layout_20.png)- Dùng **stretchColumns** để dãn đều các control, các cell (ta thường dùng dấu "*"):

```
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:stretchColumns="*"
    >
```

(http://duythanhcse.files.wordpress.com/2013/04/7_layout_21.png)d) **RelativeLayout**:

- RelativeLayout cho phép sắp xếp các control theo vị trí tương đối giữa các control khác trên giao diện (kể cả control chưa nó). Thường nó dựa vào Id của các control khác để sắp xếp theo vị trí tương đối. Do đó khi làm RelativeLayout bạn phải chú ý là đặt Id control cho chuẩn xác, nếu sau khi Layout xong mà bạn lại đổi Id của các control thì giao diện sẽ bị xáo trộn (do đó nếu đổi ID thì phải đổi luôn các tham chiếu khác sao cho khớp với Id bạn mới đổi).

- Dưới đây là ví dụ về cách sử dụng RelativeLayout (bạn để ý những dòng tô đậm):

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#ff0000ff"
    android:padding="10px" >

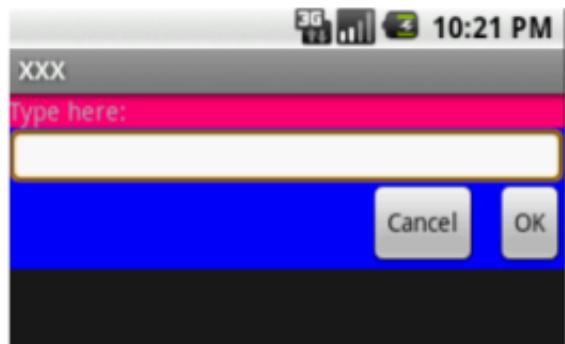
    <TextView android:id="@+id/label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#ffff0077"
        android:text="Type here:" />

    <EditText android:id="@+id/entry"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/label" />
```

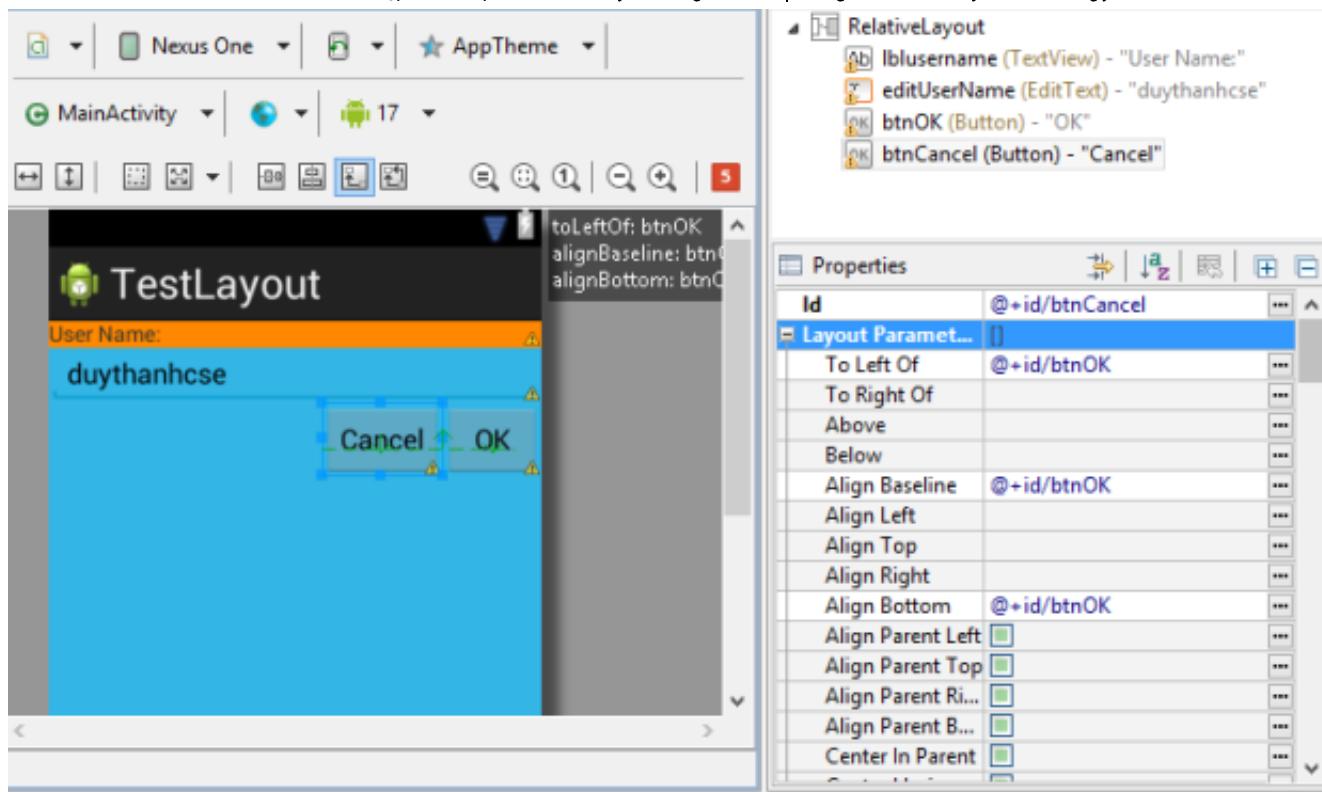
```
<Button
    android:id="@+id/ok"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/entry"
    android:layout_alignParentRight="true"
    android:layout_marginLeft="10px"
    android:text="OK" />

<Button
    android:text="Cancel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toLeftOf="@+id/ok"
    android:layout_alignTop="@+id/ok" />

</RelativeLayout>
```



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_22.png)- Ta có thể sử dụng công cụ trong Eclipse để thiết kế:



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_23.png)e) AbsoluteLayout:

- Cho phép thiết lập các control giao diện theo vị trí tùy thích:

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
    android:id="@+id/myAbsoluteLayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <Button
        android:id="@+id/myButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        android:layout_x="120px"
        android:layout_y="32px"
        >
    </Button>
</AbsoluteLayout>
```



(http://duythanhcse.files.wordpress.com/2013/04/7_layout_24.png)

- Như vậy các Tôi đã giới thiệu sơ qua cách tạo Layout và cách sử dụng một số Layout cụ thể, tùy từng trường hợp mà các bạn ứng dụng vào. Thông thường ứng với mỗi một ứng dụng cụ thể thì bạn phải kết hợp nhiều loại Layout lại với nhau.
- Bạn cần hiểu rõ các loại Layout để giúp ích cho quá trình thiết kế giao diện
- Bài tập sau Tôi sẽ hướng dẫn các bạn cách sử dụng một số control cơ bản và cách viết sự kiện cho Button như thế nào.

Chúc các bạn thành công.

Posted in: 2. Xử lý giao diện người dùng

7 Responses

Van Qui April 8, 2013 at 8:43 am | [Reply](#)

Thầy ôi...những bài lập trình thiết bị di động mà thầy post lên hay quá à...hix..em học ở cs2 , em rất thích học lập trình thiết bị di động nhưng trên này k có dạy....ò...Thầy có thể tạo 1 loạt những video dạy môn này cho những sv thiếu điều kiện như em cùng học, Thầy nha!tks Thầy rất nhiều.

Phạm Minh Ánh April 13, 2013 at 1:36 am | [Reply](#)

Cảm ơn thầy rất nhiều .bài viết của thầy em thấy rất hay .

dao duy tung April 23, 2013 at 9:19 am | [Reply](#)

thank thay.em thay day la bai viet bo ich.de hieu va chi tiet.em cung doc rat nhieu bai roi nhung cua thay ka hay nhat,sau sac hon. e mog thay co nhieu bai hon nua.va mog thay day em mot bai ve su dung eclipse de lap trinh cac ohim tat.va cac loi hay gap khi lap trinh va cach khac phuc

Quang Pham April 27, 2013 at 10:54 am | [Reply](#)

Thưa Thầy, tại sao em kéo thả control vào layout thì nó không tự động xuất hiện trong gen (em toàn phải insert trong main activity khi code báo lỗi). Mình khắc phục lỗi không tự động xuất hiện trong gen bằng cách nào?

Đại Vũ June 24, 2013 at 5:13 pm | [Reply](#)

Cảm ơn thầy, bài giảng rất chi tiết và dễ hiểu. Chúc thầy và gia đình luôn mạnh khỏe và thành công !

Hữu Nhân July 22, 2013 at 3:41 pm | [Reply](#)

Thầy ơi em muốn sắp xếp các edittext thẳng hàng từ trên xuống thì phải làm sao ạ

kaka August 3, 2013 at 9:50 am | [Reply](#)

cám ơn thầy bài viết rất hữu ích

Blog at WordPress.com. The Retro-Fitted Theme.