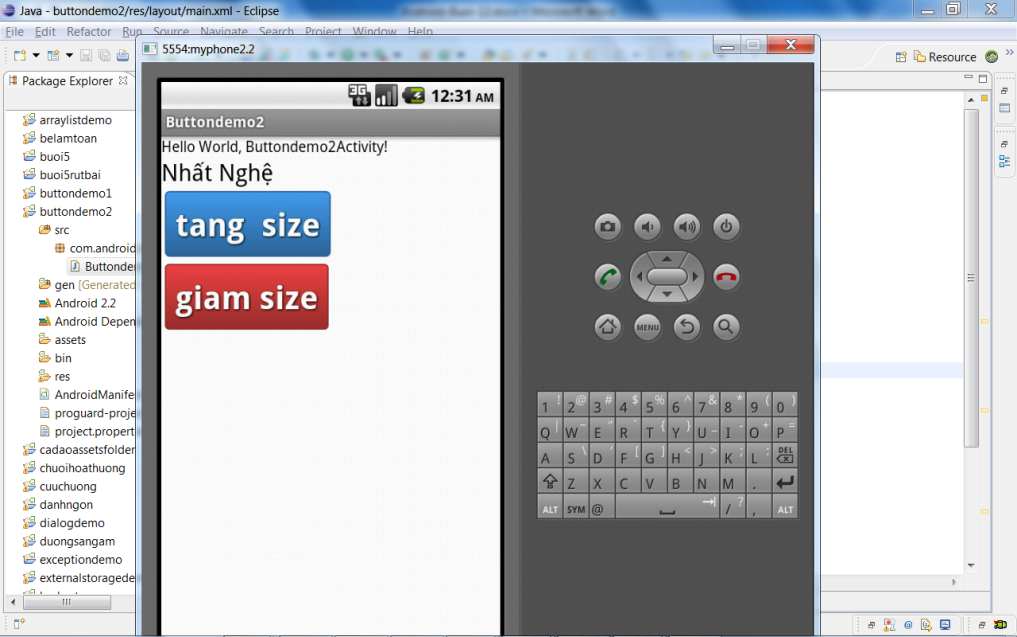
# Nút đẹp



Trong drawable tạo xml với root là selector tên nutdo.xml (xem thêm trong buttondemo2 có sẵn một loạt nút xanh,tim,den)

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

<selector xmlns:android=*"http://schemas.android.com/apk/res/android"* >

<item android:state\_pressed=*"true"* >

<shape>

<solid

android:color=*"#ef4444"* />

<stroke

android:width=*"1dp"*

android:color=*"#992f2f"* />

<corners

android:radius=*"3dp"* />

<padding

android:left=*"10dp"*

android:top=*"10dp"*

android:right=*"10dp"*

android:bottom=*"10dp"* />

</shape>

</item>

<item>

<shape>

<gradient

android:startColor=*"#ef4444"*

android:endColor=*"#992f2f"*

android:angle=*"270"* />

<stroke

android:width=*"1dp"*

android:color=*"#992f2f"* />

<corners

android:radius=*"3dp"* />

<padding

android:left=*"10dp"*

android:top=*"10dp"*

android:right=*"10dp"*

android:bottom=*"10dp"* />

</shape>

</item>

</selector>

Tạo luôn style trong string.xml

<style name=*"ButtonText"*>

<item name=*"android:layout\_width"*>fill\_parent</item>

<item name=*"android:layout\_height"*>wrap\_content</item>

<item name=*"android:textColor"*>#ffffff</item>

<item name=*"android:gravity"*>center</item>

<item name=*"android:layout\_margin"*>3dp</item>

<item name=*"android:textSize"*>30dp</item>

<item name=*"android:textStyle"*>bold</item>

<item name=*"android:shadowColor"*>#000000</item>

<item name=*"android:shadowDx"*>1</item>

<item name=*"android:shadowDy"*>1</item>

<item name=*"android:shadowRadius"*>2</item>

</style>

Cấu hình nút

<Button

android:id=*"@+id/button2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"giam size"*

android:background=*"@drawable/nutdo"*

style=*"@style/ButtonText"*

/>

## Vấn đề tăng giảm kích thước font (textSize) (dp và sp)

The default method setTextSize(float) assumes you're inputting sp units (scaled pixels), while the getTextSize() method returns an exact pixel size.

Hàm setText đưa vô là đơn vị sp. Hàm gettext trả về exact pixel

Giải quyết (nút giảm , nút tăng khác là x+1 thôi)

**float** x=tv.getTextSize();

tv.setTextSize(TypedValue.*COMPLEX\_UNIT\_PX*,x-1);

tv.setText(String.*valueOf*(x));

Xem thêm

Firstly, you should be using Density Independent Pixels - dip or dp - and not Scaled Pixels - sp.

The density-independent pixel is equivalent to one physical pixel on a 160 dpi screen. If your resources are sized in dip they will get automatically scaled on a screens with different densities; this is why the stars specified in your XML look OK. Scaled Pixels are similar but should be used for text resources only since they also reflect text scaling options.

Scale-independent Pixels (sp) this is like the dp unit, but it is also scaled by the user's font size preference. It is recommend you use this unit when specifying font sizes, so they will be adjusted for both the screen density and user's preference.

Density independent pixel (dip) A virtual pixel unit that applications can use in defining their UI, to express layout dimensions or position in a density-independent way.

2 loại này đều là trừu tượng, 1 dp tương đương 1px pixel trên màng hình 160dpi. Sp giống dp nhưng có thể co giản bởi tùy chỉnh của người dùng. (tức là người dùng chỉnh font trên máy to ra cho dể nhìn thì nó sẽ được to ra. Nên dùng sp để chỉnh font chữ để vừa phục vụ cho mật độ màn hình vừa tốt cho người sử dụng muốn chỉnh lại)