

Thiết kế giao diện trên Android

- Trong Android, dùng Activity để hiển thị màn hình.
- Mỗi activity sẽ chứa các View theo dạng cấu trúc cây, nghĩa là một Layout gốc chứa các view/layout con bên trong hoặc chỉ có 1 view duy nhất. (lưu ý Layout cũng là một view nhé)
- Có thể thiết kế giao diện trong code java hoặc trong file xml trong thư mục layout.

1. Layout mẫu của helloworld

HTML Code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

</LinearLayout>
```

2. Một số thuộc tính cơ bản

- Layout_width, layout_height: chiều rộng của view (fill_parent là to bằng kích thước của layout chứa view này, wrap_content là vừa đủ nội dung cần hiển thị của view)
- Orientation: với LinearLayout, việc sắp xếp các view là nằm kề nhau theo hàng ngang hoặc hàng dọc, ta khai báo orientation để chọn sắp theo kiểu nào (horizontal/vertical)
- Gravity: thuộc tính này qui định các view nằm bên trong layout sẽ đặt theo vị trí nào so với layout(trung tâm, trái, phải, trên dưới...)
- Weight: để các view phân chia tỉ lệ diện tích hiển thị trên màn hình (tỉ lệ tính theo weight của từng view trên tổng số weight, các view ko khai báo weight thì sẽ xem qua width và height)

4. Giao diện với LinearLayout

* Với LinearLayout (LL), các view bên trong nó được đặt kề nhau theo hàng ngang hoặc hàng dọc (cần lưu ý đặc điểm này)

* Với ví dụ vừa rồi, ta thấy cách phân tích như sau:

- Nguyên tắc chủ yếu là phân nhóm các View liên tiếp kề nhau (như 3 TextView kề nhau theo hàng dọc hoặc hàng ngang trong ví dụ trên) vào trong một LL, phân rã từ lớn đến nhỏ.
- Như vậy màn hình gồm 1 LL lớn bao bên ngoài, nhìn thấy bên trong chia thành 2 phần trên dưới rõ ràng vậy thuộc tính của LL này là dạng dọc, sau đó chia đôi ra và phân tích tiếp.
- Phần bên trên lại chia thành 2 nữa theo hàng ngang à là một LL dạng ngang, lại chia đôi: một bên là 1 ImageView (vì chỉ có 1 view nên ko cần bỏ vào trong LL), một bên lại là 1 LL chứa 3 TextView theo hàng dọc.
- Nửa bên dưới ta thấy rõ ràng chứa 3 TextView kề nhau theo hàng ngang à cho vào 1 LL dạng ngang là xong. J

FrameLayout: các view bên trong được quy định vị trí bằng khoảng cách so với biên trái và trên so với layout, các view có thể đè lên nhau.

- RelativeLayout: các view được thiết kế dựa trên quan hệ giữa chúng với nhau và với layout chứa chúng.

- AbsoluteLayout: dành cho bạn nào làm nhiều với C#, nhưng ko khuyến khích với Android nhé.

- Hạn chế độ sâu của cây

- Với các Layout phức tạp, đừng dùng RelativeLayout

- Nên chèn vào dữ liệu tạm để xem trước layout hiển thị ra sao bên tab layout (trong eclipse), nhưng xong rồi thì nhớ xóa dữ liệu tạm đi.

- FrameLayout có vấn đề với background

- Muốn tìm thuộc tính gì, bấm "android:" rồi đợi suggestion số ra xem.

Phần mềm lấy thông tin và hiển thị các File ảnh trong thẻ SD card

Đầu tiên chúng ta nói qua về thẻ SD Card, thẻ SD Card trong điện thoại thì không còn xa lạ gì với chúng ta, nhưng trong thiết bị mô phỏng thì thẻ SD Card được mô phỏng và sử dụng như thế nào? Thẻ SD Card dùng cho bộ mô phỏng sẽ được tạo khi chúng ta khởi tạo AVD, như trong máy mình sẽ nằm ở : C:\Documents and Settings\Trungh0\android\avd\Avd_2.2.avd với tên sdcard.img. Chúng ta có thể mở file này, thêm xóa sửa đổi dữ liệu thông qua chương trình UntraIso hoặc thông qua chương trình hỗ trợ của Android là File Eplorer.

Tiếp theo là đến việc lập trình: Chúng ta sẽ viết ctrình đọc toàn bộ file ảnh trong thẻ nhớ và hiển thị ảnh cũng như tên file ảnh lên màn hình Android dưới dạng 1 GridView (là một kiểu hiển thị các List Item theo dạng mạng lưới)

B1: Khởi tạo Project: name: LoadImg, MainActivity: MainActivity

B2: Thiết kế giao diện:

-Chúng ta cần 1 file xml để chứa giao diện bao gồm 1 ImageView và 1 TextView để hiển thị ảnh và tên ảnh, tạo file img.xml và code như sau:

PHP Code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">
    <ImageView android:src="@drawable/icon" android:layout_width
    ="150dp" android:id="@+id/imageView1" android:layout_height="150
    dp"></ImageView>
    <TextView android:text="TextView" android:id="@+id/textView1"
    android:layout_width="wrap_content" android:layout_height="wrap_
    content"></TextView>
</LinearLayout>
```

-File main.xml sẽ chứa 1 GridView để show ảnh và tên ảnh lên (chúng ta sẽ để Grid view hiển thị theo 2 cột)



Code: chú ý dòng numberOfColumns

PHP Code:

```
<?xml version="1.0" encoding="utf-8"?>
<GridView xmlns:android="http://schemas.android.com/apk/res/andr
oid"
    android:id="@+id/gridview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:numColumns="2"
    android:verticalSpacing="10dp"
    android:horizontalSpacing="10dp"
    android:stretchMode="columnWidth"
    android:gravity="center"
/>
```

B3: Tạo file Item.java, class này chứa các đối tượng để lưu thuộc tính của ảnh:

PHP Code:

```

package trungh0.study;

import android.graphics.Bitmap;

public class Item {
    String name;
    Bitmap img;

    public Item(String name, Bitmap img) {
        this.name=name;
        this.img=img;
    }

    public String getName(){
        return name;
    }
    public void setName(String name){
        this.name=name;
    }

    public Bitmap getImg(){
        return img;
    }
    public void setImg(Bitmap img){
        this.img=img;
    }
}

```

B4: Tạo file Image_Adapter.java để tạo một Adapter truyền dữ liệu từ một ArrayList các Item vào GridView

Code:

PHP Code:

```

package trungh0.study;

import java.util.List;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;

public class Image_Adapter extends BaseAdapter {
    private Context mContext;

```

```

private List<Item> list_item;
public Image_Adapter(Context c,List<Item> obj) {
    mContext = c;
    list_item=obj;
}

public int getCount() {
    return list_item.size();
}
public Object getItem(int position){
    return null;
}

public long getItemId(int position) {
    return 0;
}

@Override
public View getView(int position, View convertView, ViewGroup
parent) {
    View v;
    if(convertView==null){
        LayoutInflater inflater = LayoutInflater.from(mConte
xt);
        v = inflater.inflate(R.layout.img, null);

        TextView tv = (TextView)v.findViewById(R.id.textView1
);
        tv.setText(list_item.get(position).getName());
        ImageView iv = (ImageView)v.findViewById(R.id.imageV
iew1);
        iv.setImageBitmap(list_item.get(position).getImg());

    }
    else
    {
        v = convertView;
    }

    return v;
}
}

```

Chú ý: Hàm getCount() khi chúng ta return lại giá trị bao nhiêu thì sau đó ở GridView sẽ hiển thị bấy nhiêu Item, ở đây mình cho hiển thị hết những ảnh

nào đọc được nên để `list.getsize()` 🤔

B5: Quan trọng nhất là đây, chúng ta có nhiệm vụ phải đọc thông tin các file ảnh trong thẻ nhớ lưu chúng trong 1 list, sau đó sẽ dùng Adapter để đẩy chúng vào GridView

Code đọc các file ảnh trong thẻ nhớ:

PHP Code:

```
Uri uri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;
String[] cursor_cols = {
    MediaStore.Images.Media.DISPLAY_NAME,
    MediaStore.Images.Media.DATA
};

//String where = MediaStore.Images.Media. + " =1";
Cursor cursor = getContentResolver().query(uri, cursor_cols, null, null, null);
int i=0;
while (i<cursor.getCount()){
    cursor.moveToNext();
    String path = cursor.getString(cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA));
    Bitmap img=BitmapFactory.decodeFile(path);
    String name= cursor.getString(cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.DISPLAY_NAME));
    Log.d("TB:" , path + " "+name);
    list.add(new Item(name, img));
    i++;
}
```

chúng ta cũng có thể làm tương tự với các loại file media khác như music, video thông qua thư viện MediaStore.

Tuy nhiên một vấn đề nảy sinh ra là sẽ thường mất vài giây cho việc đọc và hiển thị các file ảnh lên màn hình Android, khi đó người dùng mở ứng dụng sẽ dễ tưởng máy bị treo. Cách giải quyết là ta sẽ đưa công việc mất thời gian này vào một thread chạy ngầm, và trên màn hình chính sẽ hiển thị một ProgressDialog báo hiệu chương trình vẫn đang chạy. Khi nào dữ liệu được load xong, ProgressDialog này sẽ biến mất và hình ảnh của chúng ta sẽ hiển thị.

Chúng ta sẽ code như sau:

PHP Code:

```

package trungh0.study;

import java.util.ArrayList;
import java.util.List;

import android.app.Activity;
import android.app.ProgressDialog;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.provider.MediaStore;
import android.util.Log;
import android.widget.GridView;

public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    private List<Item> list;
    private ProgressDialog pd;
    private GridView gv;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        gv=(GridView)findViewById(R.id.gridview);
        list=new ArrayList<Item>();

        pd = ProgressDialog.show(this, "Loading... ", "Please Wait!!!", true,false);
        myThread mt=new myThread();
        Thread th=new Thread(mt);
        th.start();

    }

    public class myThread implements Runnable{
        String output="";
        @Override
        public void run() {
            // TODO Auto-generated method stub
            Uri uri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;
            String[] cursor_cols = {

```

```

        MediaStore.Images.Media.DISPLAY_NAME,
        MediaStore.Images.Media.DATA
    };

    //String where = MediaStore.Images.Media. + " =1";
    Cursor cursor = getContentResolver().query(uri, cursor_cols, null, null, null);
    int i=0;
    while (i<cursor.getCount()){
        cursor.moveToNext();
        String path = cursor.getString(cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA));
        Bitmap img=BitmapFactory.decodeFile(path);
        String name= cursor.getString(cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.DISPLAY_NAME));
        Log.d("TB:" , path + " "+name);
        list.add(new Item(name, img));
        i++;
    }
    handler.sendMessage(0);
}

private Handler handler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        pd.dismiss();
        Image_Adapter ia=new Image_Adapter(getApplicationContext(), list);
        gv.setAdapter(ia);
    }
};
}
}

```

Bước 6: nhẹ nhàng nhất, save lại, dùng untralIso add thêm vào thẻ nhớ ảnh vài em hotgirl, Ctrl+F11 và chiêm ngưỡng thôi

Kết quả nè :

Kim Na Na nhé 😊)



Lập trình Android cơ bản: xử lý với GridView

Tôi muốn hiển thị một loạt các ảnh đã có trong thư mục lên một gridView để xem tổng thể. Khi đó, tôi dùng GridView đơn giản để hiển thị như sau:

1. Chuẩn bị một số ảnh cần hiển thị.
2. Ta sẽ tạo một gridView như một ma trận để hiển thị từng ảnh trên các ô đó trong main.xml như sau:

```
<GridView
    android:id="@+id/gridview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:columnWidth="90dp"
    android:numColumns="auto_fit" //số cột.Tự động phụ thuộc vào số image
    có trong folder
    android:verticalSpacing="10dp" //khoảng cách theo chiều dọc giữa hai ảnh
    liên tiếp
```

```
    android:horizontalSpacing="10dp" //khoảng cách theo chiều ngang của hai ảnh  
    kề nhau  
    android:stretchMode="columnWidth"  
    android:gravity="center" />
```

3. Trong Activity ta sẽ sửa lại code như sau:

```
public class GridViewActivity extends Activity {  
    /** Called when the activity is first created. */  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
        GridView gridView = (GridView) findViewById(R.id.gridview);  
        gridView.setAdapter(new ImageAdapter(this));  
    }  
  
    public class ImageAdapter extends BaseAdapter {  
        private Context mContext;  
        public ImageAdapter(Context c) {  
            mContext = c;  
        }  
        public int getCount() {  
            return mThumbIds.length;  
        }  
        public Object getItem(int position) {  
            return null;  
        }  
        public long getItemId(int position) {  
            return 0;  
        }  
        // create a new ImageView for each item referenced by the Adapter  
        public View getView(int position, View convertView, ViewGroup parent) {  
            ImageView imageView;  
            if (convertView == null) { // if it's not recycled, initialize some attributes  
                imageView = new ImageView(mContext);  
                imageView.setLayoutParams(new GridView.LayoutParams(85, 85));  
                imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);  
                imageView.setPadding(8, 8, 8, 8);  
            } else {  
                imageView = (ImageView) convertView;  
            }  
            imageView.setImageResource(mThumbIds[position]);  
            return imageView;  
        }  
    }  
}
```

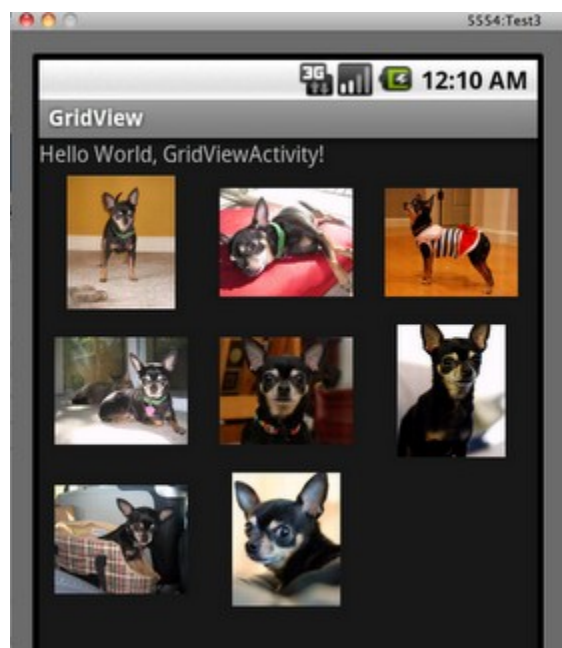
```

// references to our images
private Integer[] mThumbIds = {
    R.drawable.sample_0, R.drawable.sample_1,
    R.drawable.sample_2, R.drawable.sample_3,
    R.drawable.sample_4, R.drawable.sample_5,
    R.drawable.sample_6, R.drawable.sample_7
};
}
}

```

OK, vậy là xong.

Ta có kết quả chạy như sau:



[Download toàn bộ code tại đây.](#)

Lập trình với ListView

Đề bài: Làm ứng dụng hiển thị List Phone Book:

B1: Tạo file phonebook.xml trong thư mục res/layout và tạo ra giao diện của một hàng trong list Phone Book bao gồm các item: 1 ImageView để chứa Avatar của người dùng, 3 TextView chứa tên, số điện thoại và Email người dùng.



Jenny Nguyen

0913-223-498

jenny_in_love98@yahoo.com

Ta làm như sau:

Code:

Code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >

    <ImageView
        android:id="@+id/imgAvatar"
        android:layout_width="70dip"
        android:layout_height="70dip"
        android:scaleType="fitCenter"
        android:src="@drawable/icon"

    />

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        >

        <TextView
            android:id="@+id/tvName"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textStyle="bold"

        />

        <TextView
            android:id="@+id/tvPhone"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"

        />

        <TextView
```

```

        android:id="@+id/tvEmail"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    />

</LinearLayout>

</LinearLayout>

```

B2: Tạo giao diện cho file main.xml: Chứa một listview để hiển thị các PhoneBook. Chúng ta làm như sau:

Code:

Code:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <ListView
        android:id="@+id/listPhone"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    />

</LinearLayout>

```

B3: Vào src/name package/ tạo file PhoneBook.java: Là file định nghĩa một class PhoneBook với các thuộc tính: mAvatar, mPhone, mName, mEmail :

Code:

Code:

```

public class PhoneBook {
    private Bitmap mAvatar;
    private String mName;
    private String mPhone;
    private String mEmail;

    public PhoneBook(Bitmap avatar, String name, String phone,
String email) {
        mAvatar = avatar;
        mName = name;
        mPhone = phone;
        mEmail = email;
    }
}

```

```

    }

    public void setAvatar(Bitmap avatar) {
        mAvatar = avatar;
    }
    public Bitmap getAvatar() {
        return mAvatar;
    }

    public void setName(String name) {
        mName = name;
    }
    public String getName() {
        return mName;
    }

    public void setPhone(String phone) {
        mPhone = phone;
    }
    public String getPhone() {
        return mPhone;
    }

    public void setEmail(String email) {
        mEmail = email;
    }
    public String getEmail() {
        return mEmail;
    }
}

```

B4: Vào src/name package/ tạo file PhoneBookAdapter.java: tạo một lớp Adapter để truyền dữ liệu từ một ArrayList vào ListView:

Code:

Code:

```

package trungh0.study;

import java.util.List;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;

```

```
public class PhoneBookAdapter extends BaseAdapter {
    private Context mContext;
    private List<PhoneBook> mListPhoneBook;

    public PhoneBookAdapter(Context context, List<PhoneBook>
list) {
        mContext = context;
        mListPhoneBook = list;
    }

    @Override
    public int getCount() {
        return mListPhoneBook.size();
    }

    @Override
    public Object getItem(int pos) {
        return mListPhoneBook.get(pos);
    }

    @Override
    public long getItemId(int pos) {
        return pos;
    }

    @Override
    public View getView(int pos, View convertView, ViewGroup
parent) {
        // get selected entry
        PhoneBook entry = mListPhoneBook.get(pos);

        // inflating list view layout if null
        if(convertView == null) {
            LayoutInflater inflater =
LayoutInflater.from(mContext);
            convertView =
inflater.inflate(R.layout.phonebook.xml, null);
        }

        // set avatar
        ImageView ivAvatar =
(ImageView)convertView.findViewById(R.id.imgAvatar);
        ivAvatar.setImageBitmap(entry.getAvatar());
    }
}
```

```

        // set name
        TextView tvName =
        (TextView)convertView.findViewById(R.id.tvName);
        tvName.setText(entry.getName());

        // set phone
        TextView tvPhone =
        (TextView)convertView.findViewById(R.id.tvPhone);
        tvPhone.setText(entry.getPhone());

        // set email
        TextView tvEmail =
        (TextView)convertView.findViewById(R.id.tvEmail);
        tvEmail.setText(entry.getEmail());

        return convertView;
    }
}

```

B5: Mở file MainActivity.java và sửa:

Code:

```

package trungh0.study;

import java.util.ArrayList;
import java.util.List;
import android.app.Activity;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.widget.ListView;

public class MainActivity extends Activity {

    private ListView lvPhone;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        lvPhone = (ListView) findViewById(R.id.listPhone);

        List<PhoneBook> listPhoneBook = new
        ArrayList<PhoneBook>();
    }
}

```



```

        listPhoneBook.add(new
PhoneBook(BitmapFactory.decodeResource(getResources(),
        R.drawable.kimnana), "Kim Nana", "0947-896-668",
"Umbala_Kimnana@gmail.com"));

        listPhoneBook.add(new
PhoneBook(BitmapFactory.decodeResource(getResources(),
        R.drawable.ellytran), "Elly Tran", "0986-686-868",
"EllyTran@gmail.com"));

        listPhoneBook.add(new
PhoneBook(BitmapFactory.decodeResource(getResources(),
        R.drawable.thanhthuy), "Thanh Thuy", "0902-001-002",
"ThanhThuy@gmail.com"));

        listPhoneBook.add(new
PhoneBook(BitmapFactory.decodeResource(getResources(),
        R.drawable.thuhuong), "Thu Huong", "0986-999-999",
"ThuHuong9x@gmail.com"));

        listPhoneBook.add(new
PhoneBook(BitmapFactory.decodeResource(getResources(),
        R.drawable.thuylinh), "Thuy Linh", "0986-006-228",
"hot_girl9x@gmail.com"));

        listPhoneBook.add(new
PhoneBook(BitmapFactory.decodeResource(getResources(),
        R.drawable.trang), "Thuy Trang", "0986-677-028",
"bagiabanmaybay@gmail.com"));







        PhoneBookAdapter adapter = new PhoneBookAdapter(this,
listPhoneBook);
        lvPhone.setAdapter(adapter);
    }
}

```

Chú ý: Các file R.drawable.thanhthuy,R.drawable.thuhuong... là các file ảnh được mình thêm vào trong thư mục: res\drawable-mdpi

B6: Kết quả:

Phone Book

- | | |
|---|---|
|  | Kim Nana
0947-896-668
Umbala_Kimnana@gmail.com |
|  | Elly Tran
0986-686-868
EllyTran@gmail.com |
|  | Thanh Thuy
0902-001-002
ThanhThuy@gmail.com |
|  | Thu Huong
0986-999-999
ThuHuong9x@gmail.com |
|  | Thuy Linh
0986-006-228
hot_girl9x@gmail.com |
|  | Thuy Trang
0986-677-028
bagiabanmaybay@gmail.com |

Còn đây 4 em vô danh 😊)
3G 8:58 PM



Lập trình với các thành phần giao diện cơ bản

Trong bài này mình sẽ giới thiệu với các bạn cách lập trình với các thành phần giao diện cơ bản như TextView, EditText và Button

Ví dụ của chúng ta: Có một EditText cho chúng ta nhập Text vào, có một Button là Xử lí thì sẽ hiển thị dòng Text ta đã nhập vào lên một TextView

B1: Khởi tạo project: cài này tùy do các bạn đặt, đã hướng dẫn trong bài HelloWorld

B2: Xây dựng giao diện: mở file res/layout/main.xml và làm như sau:

Code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
```

```

>
<EditText
    android:id="@+id/editText"
    android:hint="Nhập Text vào đây" // Đoạn này sẽ hiện dòng
chữ "Nhập Text vào đây" ở dạng chữ mờ.
    android:layout_width="fill_parent"
    android:layout_height="60dp"
/>
<TextView
    android:id="@+id/textView"
    android:layout_width="fill_parent"
    android:layout_height="60dp"
    android:text="Text đã nhập: "
/>
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Xử lý"
/>
</LinearLayout>

```

-Chú ý: android:hint=" " dùng để hiện thị chuỗi mờ trong Edit Text.

Bước 3: Xử lý bắt sự kiện, mở file MainActivity.java và làm như sau:

Code:

```

package HelloAndroid.Example;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.view.View.OnClickListener;

public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        //Khai báo các biến dùng

```

```

        final TextView
tv=(TextView) findViewById(R.id.textView);
        final EditText
ed=(EditText) findViewById(R.id.editText);
        Button bt=(Button) findViewById(R.id.button);

        //Thiết lập sự kiện button click
        View.OnClickListener btOnClickListener=new
OnClickListener() {

                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        tv.setText(ed.getText().toString());
                // Set text nhận được tại EditText cho TextView

                }

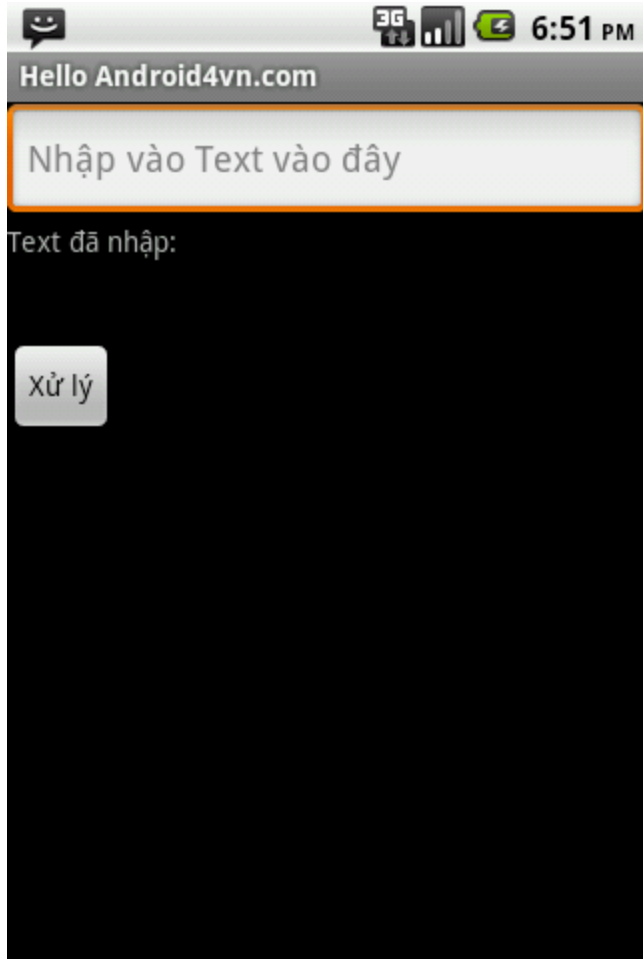
        };

        //Set sự kiện cho Button
        bt.setOnClickListener(btOnClickListener);

    }
}

```

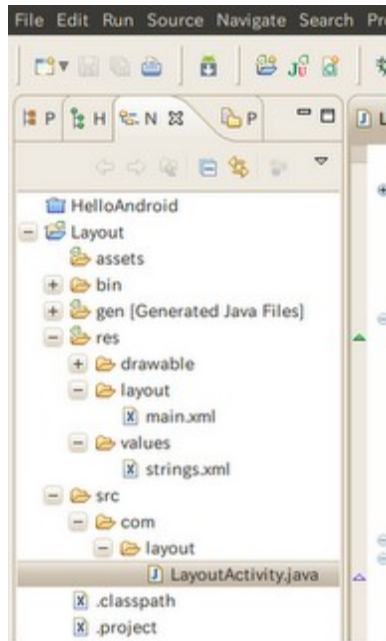
B4: Save, Ctrl+ F11 và xem kết quả:



- **Lập trình Android cơ bản: xử lý layout**

Thiết lập xong môi trường lập trình Android, ta có thể tiến hành xử lý layout của Android. Tất nhiên, việc thiết kế layout không phải là mục đích trong bài này. Tôi chỉ hướng dẫn cách xử lý các đối tượng vô cùng đơn giản trên layout mà thôi.

Ta tạo một project Layout mới với các thành phần như sau:



Các file quan trọng nhất trong project này là các file sau:

1. *string.xml*: file thiết lập các tên đối tượng đặt trong layout. Khi biên dịch, trình biên dịch sẽ tìm đến file này để đọc tên biến và so sánh với các đối tượng có trong layout để tạo ra file *R.class*. *R.class* sẽ được sinh ra để tạo ra mối liên hệ giữa biến và đối tượng đồng nhất
2. *main.xml*: dùng để thiết kế giao diện cho app.
3. *LayoutActivity.java*: file code của chương trình.

Đầu tiên, ta tiến hành thêm các biến trong *string.xml* như sau:

Ta thêm 2 biến kiểu *string* là 「*init_msg*」 「*btn_label*」. Hai biến này sẽ được định nghĩa trong code của *string.xml* như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Gioi thieu ve Layout tren
Android</string>
    <string name="init_msg">Xin hay nhap ho va ten</string>
    <string name="btn_label">Click</string>
    <string name="app_name">Layout</string>
</resources>
```

Tiếp đến, ta sửa giao diện trong *main.xml* bằng cách thêm mới 「*EditText*」 「*Button*」. Việc này có thể thực hiện bằng cách *drag&drop* hoặc thực hiện trực tiếp trên code như sau:

Ta cần thêm các đối tượng có tính chất sau:

- **TextView** (là đối tượng có sẵn)

id——@+id/text

text——@string/init_msg

- **EditText**

id——@+id/edit

· **Button**

id——@+id/button

text——@string/btn_label

Chú ý: Ta cần để ý đến các tên đối tượng được định nghĩa sẵn là text, init_msg, edit, button, btn_label sẽ được dùng để code trong chương trình.

Code trong main.xml sẽ như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:id="@+id/text"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/init_msg"
        />
    <EditText
        android:id="@+id/edit"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        />
    <Button
        android:id="@+id/button"
        android:layout_width="100px"
        android:layout_height="wrap_content"
        android:text="@string/btn_label"
        />
</LinearLayout>
```

Bước cuối cùng là ta sửa code cho *LayoutActivity.java*. Ngoài việc tạo 3 đối tượng đã thêm vào, ta còn phải chú ý đến việc tạo event click.

```
package com.layout;
import android.app.Activity;
import android.os.Bundle;
import android.text.Editable;
import android.view.View;
import android.view.View.*;
import android.widget.*;

public class LayoutActivity extends Activity {
    private Button button;
    private EditText edit;
    private TextView text;
```



```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    button = (Button) this.findViewById(R.id.button);
    edit = (EditText) this.findViewById(R.id.edit);
    text = (TextView) this.findViewById(R.id.text);

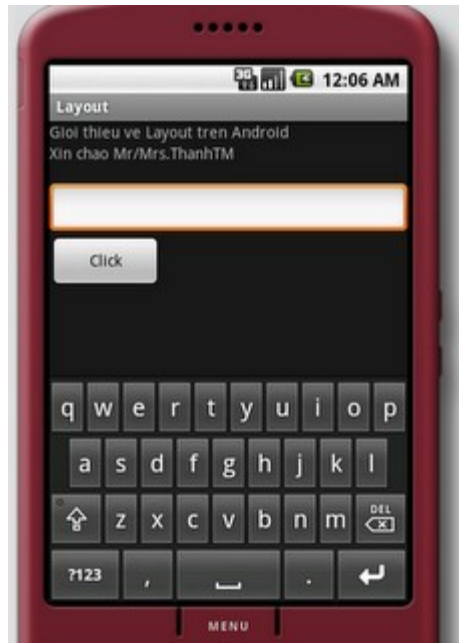
    button.setOnClickListener(new MyClickAdapter());
}

class MyClickAdapter implements OnClickListener {
    @Override
    public void onClick(View v) {
        Editable s = edit.getText();
        text.setText("Xin chào Mr/Mrs." + s);
    }
}

```

OK. Nếu bạn làm đúng các bước trên, chắc chắn sẽ chạy.
 Kết quả của tôi như sau:





Xong. Vậy là ta đã biết cách để xử lý button, text và label trên Android.
Chúc các bạn thành công.

Download toàn bộ code [tại đây](#).

ThanhTM

Lập trình Android cơ bản: xử lý Tab



Trong Android, tab được sử dụng rộng rãi và là thành phần không thể thiếu trong xử lý hiển thị. Trong bài này, tôi sẽ tìm hiểu cách lập trình tab và cách chỉnh sửa tab trên Android. Tôi sẽ trình bày theo các mục sau:

1. Nguyên tắc tạo tab
2. Tap trên top screen
3. Tap ở bottom screen
4. Tap ở left screen
5. Tab ở right screen

Nguyên tắc tạo tab

Khi tạo tab trong Android, ta cần phải tuân theo một nguyên tắc để tạo layout cho tab. Ví dụ, để tạo tab trên top màn hình, ta sẽ tạo layout như sau:

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

<TabHost

    xmlns:android="http://schemas.android.com/apk/res/android"

    android:id="@android:id/tabhost"

    android:background="#0094FF"

    android:layout_width="fill_parent"

    android:layout_height="fill_parent">

    <LinearLayout

        android:orientation="vertical"

        android:layout_width="fill_parent"

        android:layout_height="fill_parent">

        <!-- Tab -->

        <TabWidget

            android:id="@android:id/tabs"

            android:layout_width="fill_parent"

            android:layout_height="wrap_content" />

        <!-- Seperator -->

        <FrameLayout

            android:background="#222222"
```

```

        android:layout_width="fill_parent"

        android:layout_height="1dp" />

<!-- Noi dung tab -->

<FrameLayout

        android:id="@android:id/tabcontent"

        android:layout_weight="1"

        android:layout_width="fill_parent"

        android:layout_height="fill_parent" />

</LinearLayout>

</TabHost>

```

Để tạo tab, ta phải định nghĩa layout là tab bằng tag TabHost. Trong id phần này ta phải chỉ định là android:id/tabhost. Sau đó, trong TabHost, ta định nghĩa LinearLayout rồi trong đó định nghĩa tag TabWidget, id là @android:id/tabs, nội dung của tab đặt trong FrameLayout có id là @android:id/tabcontent.

Nếu định nghĩa layout theo nguyên tắc trên thì ta có thể tạo tab và hiển thị tab theo ý muốn.

Tiếp theo, ta cần tạo các lớp TabActivity để thực hiện chức năng của tab.

Tap trên top screen

Để thử nghiệm, ta sẽ lập trình đơn giản tab trên top màn hình của Android.

Định nghĩa main.xml layout:

```

<?xml version="1.0" encoding="utf-8"?>
<TabHost
    android:id="@android:id/tabhost"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <LinearLayout
        android:layout_width="match_parent"
        android:id="@+id/linearLayout1"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <TabWidget
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@android:id/tabs">

```

```

</TabWidget>

<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@android:id/tabcontent">
</FrameLayout>
</LinearLayout>
</TabHost>

```

Trong Main Activity ta code như sau:

```

import android.app.TabActivity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TabHost;

public class TabBarActivity extends TabActivity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        TabHost tabHost = getTabHost();
        TabHost.TabSpec spec;
        Intent intent;

        intent = new Intent().setClass(this, FirstActivity.class);
        spec = tabHost.newTabSpec("First").setIndicator("First")
            .setContent(intent);
        tabHost.addTab(spec);

        intent = new Intent().setClass(this, SecondActivity.class);
        spec = tabHost.newTabSpec("Second").setIndicator("Second")
            .setContent(intent);
        tabHost.addTab(spec);

        intent = new Intent().setClass(this, ThirdActivity.class);
        spec = tabHost.newTabSpec("Third").setIndicator("Third")
            .setContent(intent);
        tabHost.addTab(spec);

        intent = new Intent().setClass(this, FourthActivity.class);
        spec = tabHost.newTabSpec("Fourth").setIndicator("Fourth")
            .setContent(intent);
        tabHost.addTab(spec);
    }
}

```

```
}  
}
```

Tạo layout tabbar hiển thị chồng liên tiếp tab_test.xml

```
<?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="fill_parent">  
    <TextView  
        android:id="@+id/txtDisplayedTab"  
        android:layout_width="fill_parent"  
        android:layout_height="fill_parent"  
        android:textAppearance="?android:attr/textAppearanceLarge"  
        android:text="TextView"  
        android:gravity="center|center_vertical">  
    </TextView>  
  
</RelativeLayout>
```

Định nghĩa Activity cho từng tab như sau:

FirstActivity.java

```
import android.app.Activity;  
import android.os.Bundle;  
import android.widget.TextView;  
  
public class FirstActivity extends Activity  
{  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        // TODO Auto-generated method stub  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.tab_test);  
  
        TextView txtView = (TextView) findViewById(R.id.txtDisplayedTab);  
        txtView.setText("First Tab is Selected");  
    }  
}
```

Các Activity này sẽ được gọi trực tiếp trong main Activity. Tiếp tục tạo các Activity cho tab2, tabe, tab4 như trên theo các Activity: SecondActivity.java, ThirdActivity.java, FourthActivity.java.

Trong AndroidManifest.xml, ta đặt các tab này trong application để gọi khi click tab.

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.technotalkative.tabbarexample"  
    android:versionCode="1"
```

```

        android:versionName="1.0">
<uses-sdk android:minSdkVersion="8" />

<application android:icon="@drawable/icon"
android:label="@string/app_name">
    <activity android:name=".TabBarActivity"
        android:label="@string/app_name">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

<activity android:name="FirstActivity">
</activity>

<activity android:name="SecondActivity">
</activity>

<activity android:name="ThirdActivity">
</activity>

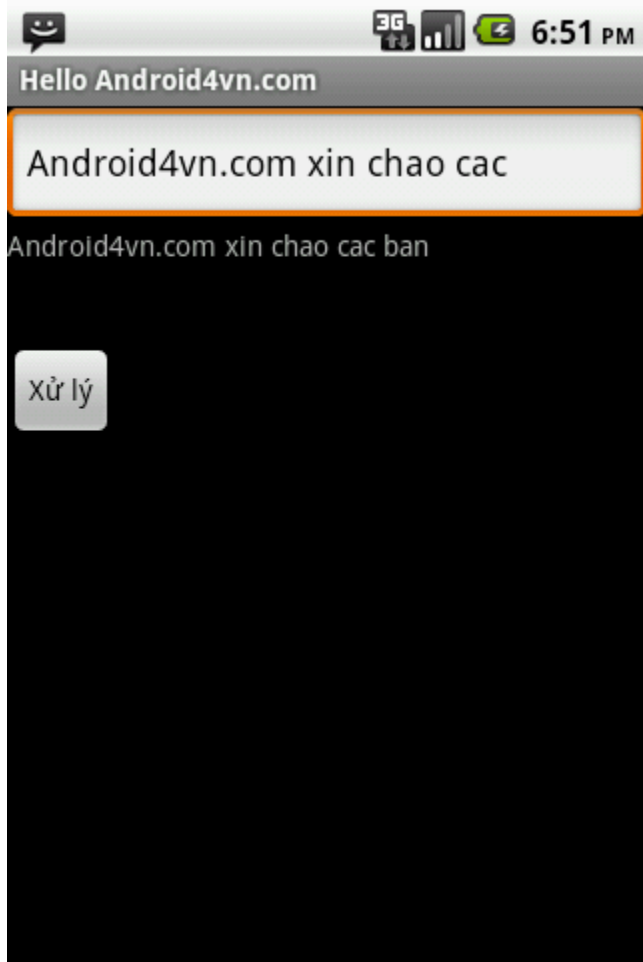
<activity android:name="FourthActivity">
</activity>
</application>
</manifest>

```

Kết quả chạy như sau:



Download toàn bộ code [tại đây](#)



Tạo menu trong android

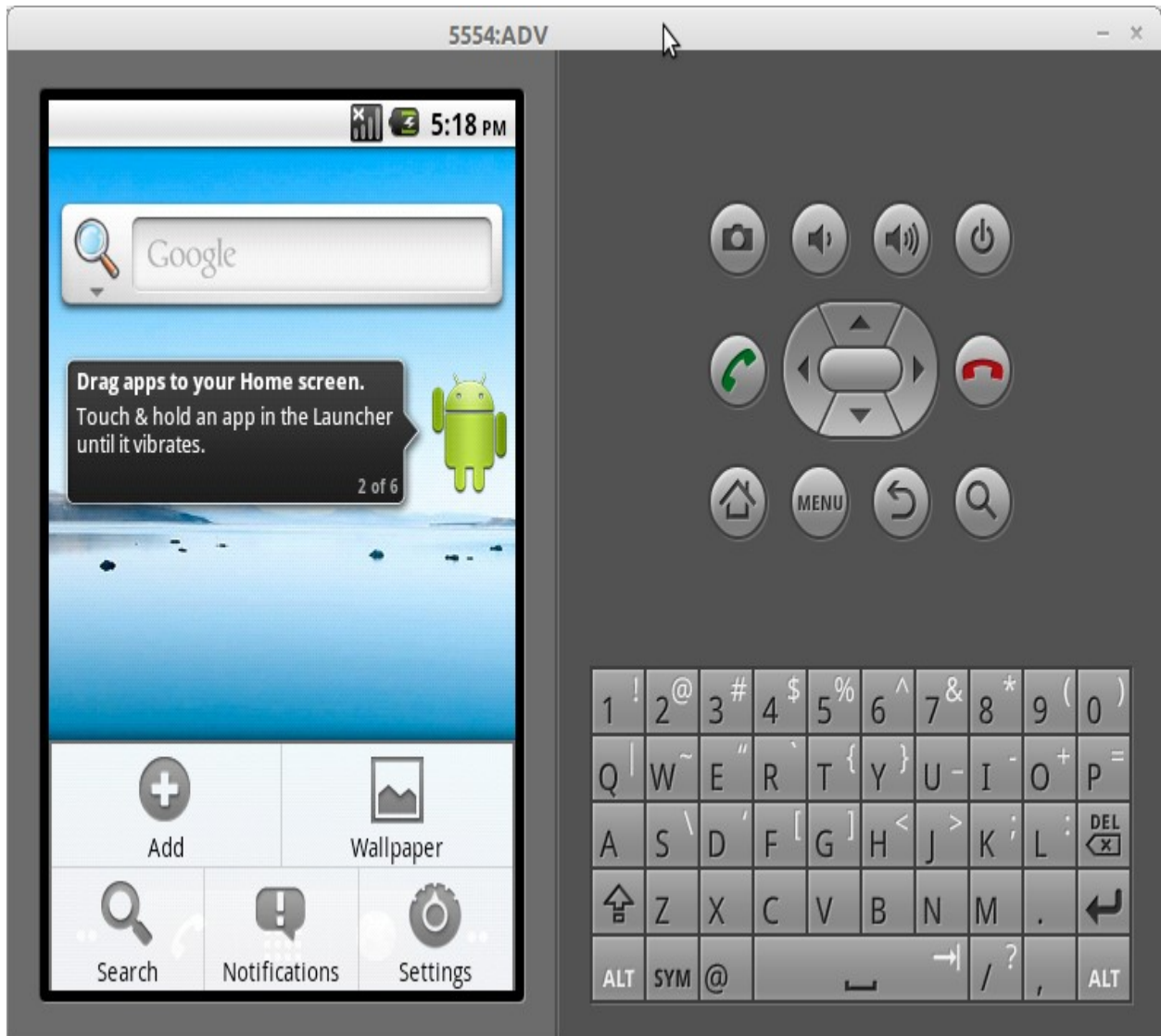
Chào các bạn.

Hôm nay, chúng ta sẽ cùng tìm hiểu và làm 1 demo nhỏ về menu trong android.

Menu là gì hẳn các bạn cũng đã biết rồi, menu giúp user có thể dễ dàng truy cập đến các chức năng khác của ứng dụng hơn.

Trong các ứng dụng thì menu là 1 phần không thể thiếu.

Ví dụ về một menu:



Bây giờ chúng ta sẽ làm 1 demo nhỏ về menu:

B1, Tạo 1 project với tên mà bạn thích ở đây ví dụ mình tạo project là DemoMenu.

B2, Trong phần res bạn tạo 1 folder với tên là menu (chuột phải res -> New -> Folder)

B3, Trong thư mục menu bạn tạo 1 file android xml với tên bất kỳ ví dụ là menu.xml (chuột phải vào folder menu -> New -> Other -> Android -> Android XML file)

B4, Bạn viết code vào trong file menu.xml đây chính là phần giao diện của menu sẽ hiện ra. Mỗi thành phần trong menu được coi là 1 item.

Mỗi item thì sẽ có 2 thành phần chính là icon,title tất nhiên là có thể có 1 trong 2 hoặc không có gì cả

Bạn viết code xml cho file menu như sau

PHP Code:

```
<?xml version="1.0" encoding="utf-8"?>
<menu
  xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/menu_icon"
    android:icon="@drawable/icon"
  />
  <item
    android:id="@+id/menu_text"
    android:title="android4vn.com"
  />

  <item
    android:id="@+id/menu_text_and_icon"
    android:icon="@drawable/icon"
    android:title="android4vn.com"
  />

  <item
    android:id="@+id/menu_none"
  />
</menu>
```

B5, Bạn vào file activity ở đây là DemoActivity và override 2 phương thức là onCreateOptionsMenu và onOptionsItemSelected

- phương thức onCreateOptionsMenu : được gọi khi menu được tạo ra
- phương thức onOptionsItemSelected : được gọi khi user chọn 1 item trong menu

Chú ý: để tiện và tránh sai sót thì khi viết các phương thức override thì bạn vào Source -> Override / Implement methods.. rồi chọn các phương thức cần override ở đây là onCreateOptionsMenu và onOptionsItemSelected

PHP Code:

```
package android.demomenu;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;

public class DemoMenuActivity extends Activity {
    @Override
```

```

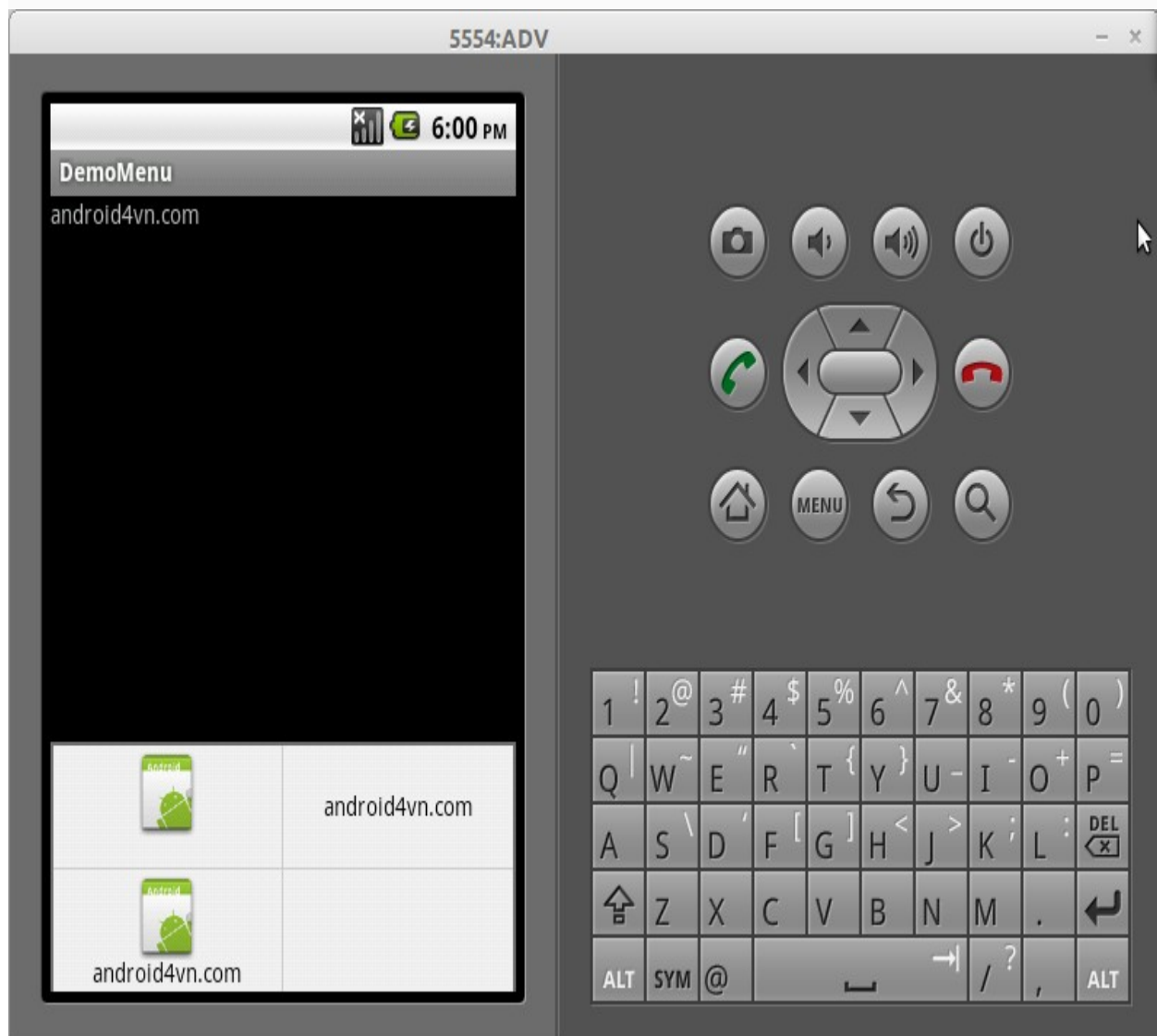
public boolean onOptionsItemSelected(MenuItem item) {
    // TODO Auto-generated method stub
    switch (item.getItemId()) {
        case R.id.menu_icon:
            Toast.makeText(this, "Bạn chọn item chỉ có icon :D",
Toast.LENGTH_LONG).show();
            break;
        case R.id.menu_text:
            Toast.makeText(this, "Bạn chọn item chỉ có text :)",
Toast.LENGTH_LONG).show();
            break;
        case R.id.menu_text_and_icon:
            Toast.makeText(this, "Bạn chọn item có text và icon
:X", Toast.LENGTH_LONG).show();
            break;
        case R.id.menu_none:
            Toast.makeText(this, "Bạn chọn item không có gì cả :
(", Toast.LENGTH_LONG).show();
            break;
        default:
            break;
    }
    return true;
}

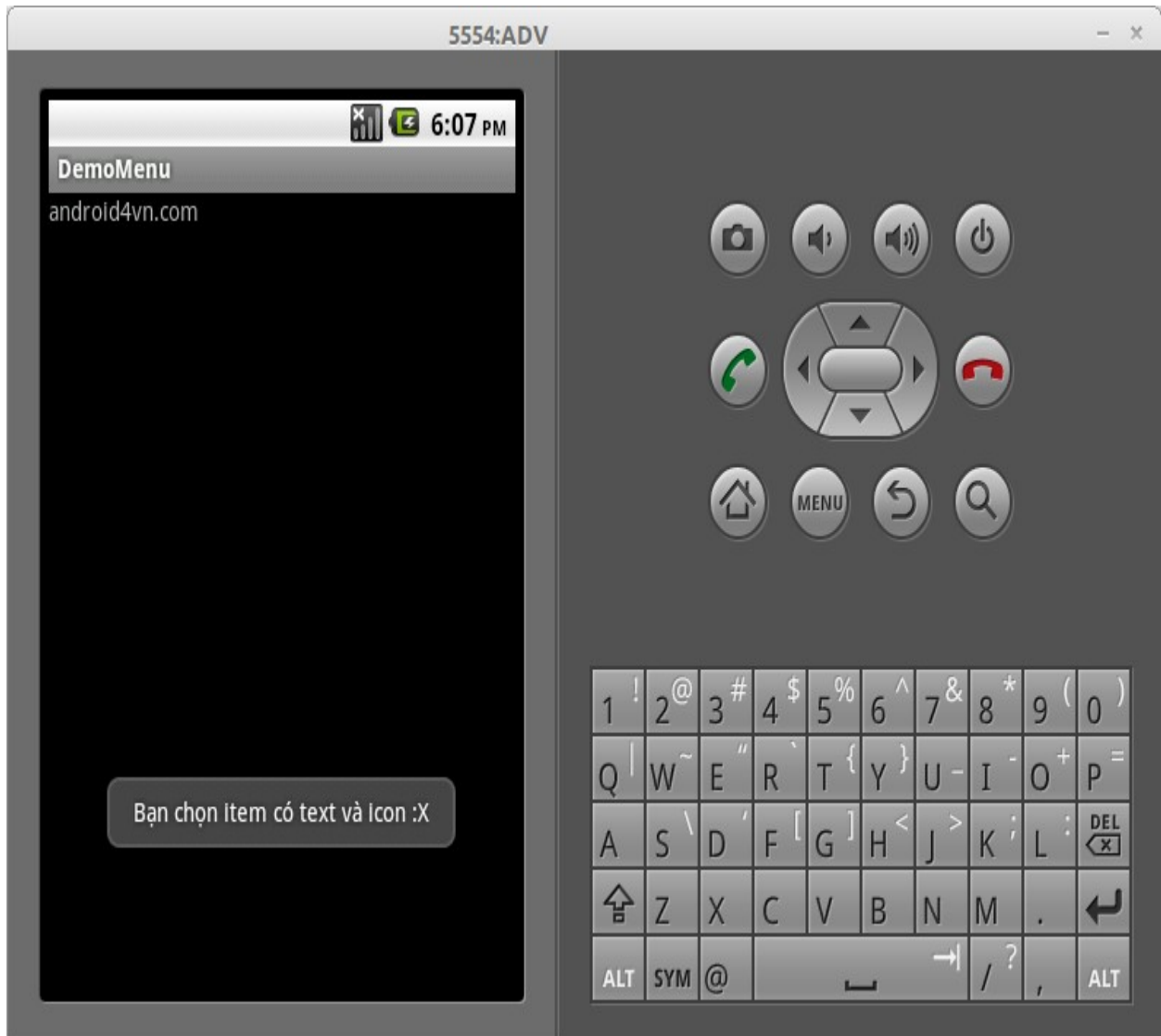
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // TODO Auto-generated method stub
    super.onCreateOptionsMenu(menu);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu, menu);
    return true;
}

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
}
}

```

Bây giờ bạn hãy chạy chương trình (Ctrl + F11)
Ấn vào phím menu trên bàn phím của emu để tạo menu
Một vài hình ảnh demo.





[Source code download](#)

AutoCompleteTextView trong Android

AutoCompleteTextView trong Android gần giống như 1 EditText để cho user có thể nhập thông tin vào đó.

Tuy nhiên, AutoComplete có khả năng đưa ra các suggestion (gợi ý) để user có thể chọn lựa đảm bảo cho việc nhập dữ liệu chính xác.

Dưới đây, chúng ta sẽ thử làm 1 demo nhỏ.

B1, Tạo 1 project giả sử với tên là : DemoAutoCompleteTextView
(Vào File->New->Project->Android->Android Project gõ tên project và tên package chú ý là tên package phải có dạng aaa.bbb)

B2, Vào layout main để tạo AutoCompleteTextView (vào res->layout->main.xml ở phía dưới file chọn main.xml để gõ tạo giao diện bằng tay)

Tạo 1 AutoCompleteTextView như sau:

file: main.xml

PHP Code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/
android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <AutoCompleteTextView
        android:id="@+id/ac_demo"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
    />
    <Button
        android:id="@+id/btn_ok"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="OK"
    />
    <TextView
        android:id="@+id/tv_output"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    />
</LinearLayout>
```

android:id đây là định danh cho AutoComplete để sau này mình có thể gọi ra được đúng nó.

B3, Mở file activity ra để viết code xử lý

Đầu tiên chúng ta cần tạo ra 1 mảng các từ để AutoCompleteTextView có thể dựa vào đó mà đưa ra suggestion.

PHP Code:

```
private static final String[] COUNTRIES =
{
    "Belgium", "France", "Italy", "Germany", "Spain", "Viet Nam",
    "China", "Japan", "Korean", "Russian", "Canada", "Afghanistan",
    "Albania", "Algeria", "American Samoa", "Andorra", "Angola", "Ar
gentina", "Armenia", "Aruba", "Australia", "Austria", "Azerbaija
n", "Zimbabwe"
};
```

Sau đó chúng ta sẽ tạo ra 1 ArrayAdapter và set adapter này cho AutoCompleteTextView

file: DemoAutoCompleteTextViewActivity

PHP Code:

```
package demo.autocomplete;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.TextView;

public class DemoAutoCompleteTextViewActivity extends Activity implements OnClickListener{

    private static final String[] COUNTRIES =
    {
        "Belgium", "France", "Italy", "Germany", "Spain", "Viet Nam", "China", "Japan", "Korean", "Russian", "Canada", "Afghanistan", "Albania", "Algeria", "American Samoa", "Andorra", "Angola", "Argentina", "Armenia", "Aruba", "Australia", "Austria", "Azerbaijan", "Zimbabwe"
    };
    private Button btn_ok;
    private TextView tv_output;
    private AutoCompleteTextView ac_txt;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_dropdown_item_1line, COUNTRIES);
        ac_txt = (AutoCompleteTextView) findViewById(R.id.ac_demo);

        ac_txt.setThreshold(1);
        ac_txt.setDropDownHeight(200);
        ac_txt.setAdapter(adapter);

        btn_ok = (Button) findViewById(R.id.btn_ok);
        btn_ok.setOnClickListener(this);

        tv_output = (TextView) findViewById(R.id.tv_output);
```

```

    }

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        if(v.getId() == R.id.btn_ok)
        {
            tv_output.setText(ac_txt.getText().toString());
        }
    }
}

```

Ở trong đoạn code này có xử lý thêm sự kiện user nhấn vào nút ok. Khi đó chúng ta sẽ lấy ra đoạn text ở trong autoCompleteTextView và set nó vào TextView :

PHP Code:

```
tv_output.setText(ac_txt.getText().toString());
```

Câu lệnh

PHP Code:

```
ac_txt.setThreshold(1);
```

có nghĩa là khi user gõ vào 1 ký tự thì bắt đầu đưa ra suggestion, bạn có thể set nó nhiều hơn.

Câu lệnh

PHP Code:

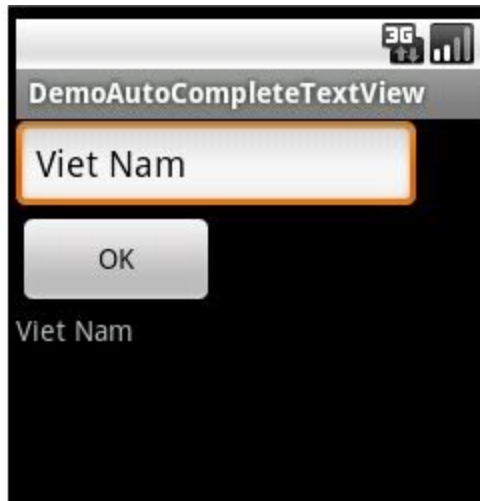
```
ac_txt.setDropDownHeight(200);
```

là để set chiều cao của ô gợi ý số ra.

AutoCompleteTextView tìm kiếm rất tốt ví dụ ở đây có từ khóa là "Viet Nam" thì chỉ cần user gõ vào Nam thì cũng sẽ đưa ra gợi ý là Viet Nam, nhưng nếu mà gõ vào "iet" thì sẽ không có gợi ý là "Viet Nam"

Đến đây bạn có thể bắt đầu chạy project được rồi:





Source code: <http://www.mediafire.com/?2v4ea0yr0aew9>