实现的功能：广告页自动轮播，并可以手动轮播，且能加载sd卡下指定路径下的图片文件。

1. 完成广告页的页面布局

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center" >

<Gallery

android:id="@+id/gallery1"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_centerHorizontal="true"

android:layout\_centerVertical="true" />

</RelativeLayout>

1. 编写填充到Gallery中的Adapter对象，由于要获取指定路径的图片，要对BaseAdapter 的方法进行简单的改写。

public class ImageAdapter extends BaseAdapter {

private Context context;

private int[] imageId = {R.drawable.ad1,R.drawable.ad2,R.drawable.ad3};//设置默认的图片，保证没有获取到指定路径的图片时，也能显示默认图片。

private ImageView[] images=null;

//默认构造方法

public ImageAdapter(Context context){

this.context = context;

}

//外部ImageView数组的构造方法

public ImageAdapter(Context context,ImageView[] images){

this.context = context;

this.images=images;

}

@Override

public int getCount() {

// TODO Auto-generated method stub

if(images==null)

return Integer.MAX\_VALUE;//设置为最大，这样能解决当图片轮到最后时，能继续从第一张开始。

else

return Integer.MAX\_VALUE;

}

@Override

public Object getItem(int position) {

// TODO Auto-generated method stub

if(images==null)

return imageId[position];

else

return images[position];

}

@Override

public long getItemId(int position) {

// TODO Auto-generated method stub

return position;

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

// TODO Auto-generated method stub

if(images==null){

ImageView imageView = new ImageView(context);

imageView.setBackgroundResource(imageId[position%(imageId.length)]);

return imageView;

}else{

return images[position%(images.length)];

}

}

}

布局对应的activity中，声明Gallery控件：

final List<File> templist=getFileList(path);

if(templist==null){

imageadapter=new ImageAdapter(this);

}else{

imageadapter=new ImageAdapter(this,filepathToImageview(templist));

}

gallery=(Gallery) findViewById(R.id.gallery1);

//　设置Gallery组件的Adapter对象

gallery.setAdapter(imageadapter);

/\*\*storage/emulated/0/rice/image

\* 遍历查询SD卡中的图片，并将其保存为file格式,如果图片少于三张，会出现不可预知的黑屏现象

\*/

public static List<File> getFileList(String strPath) {

System.out.println("获取"+strPath+"路径");

File dir = new File(strPath);

File[] files = dir.listFiles(); // 该文件目录下文件全部放入数组

List<File> filelist = new ArrayList<File>();

if (files != null) {

System.out.println("路径不为空");

for (int i = 0; i < files.length; i++) {

String fileName = files[i].getName();

if (files[i].isDirectory()) { // 判断是文件还是文件夹

getFileList(files[i].getAbsolutePath()); // 获取文件绝对路径

} else if (fileName.endsWith("jpg") || fileName.endsWith("png")) { //

// 判断文件名是否以.jpg结尾

String strFileName = files[i].getAbsolutePath();

filelist.add(files[i]);

} else {

continue;

}

}

System.out.println("filelist的长度"+filelist.size());

}

if(files == null){

System.out.println("files为空");

filelist=null;

}

return filelist;

}

/\*\*\*\*

\*把路径转为imageview[]

\*

\*\*\*\*/

private ImageView[] filepathToImageview(List<File> filelist){

ImageView[] images=new ImageView[filelist.size()];

for(int i=0;i<filelist.size();i++){

String s = filelist.get(i).getAbsolutePath();

System.out.println("实际获取图片："+s);

Bitmap bmtemp =BitmapFactory.decodeFile(filelist.get(i).getAbsolutePath());

System.out.println("bmtemp为："+bmtemp);

ImageView image=new ImageView(this);

image.setImageBitmap(bmtemp);

System.out.println("image为："+image);

images[i]=image;

System.out.println("images["+i+"]为："+images[i]);

}

return images;

}

1. 为了达到自动轮播的效果，还要在oncreat中设置一下

handler = new Handler()

{

public void handleMessage(Message msg)

{

if(msg.what == 0x123)

{

//设置gallery显示第几张图片

gallery.setSelection(current);

}

}

};

timer = new Timer();

timer.schedule(new TimerTask()

{

@Override

public void run() {

current++;

if(current>=5) //5是存图片的数组，数组的长度

{

current = 0;//此处如果需要gallery循环播放，就需要这么写

}

Message msg = new Message();

msg.what = 0x123;

handler.sendMessage(msg);

}

}, 0, 3000);