



## Ch3 多媒体技术练习

### 1. 本次课目标

掌握发送通知的方法；掌握收发短信的方法；掌握调用摄像头的方法；掌握调用系统相册的方法；掌握播放音频的几类常用方法；掌握播放视频的几类常用方法。

### 2. 本次课任务

#### 2.1. 使用通知

#### 2.2. 发送短信

#### 2.3. 查看系统短信

#### 2.4. 调用摄像头方法一：调用系统相机的功能

#### 2.5. 调用摄像头方法二：直接调用 Camera 硬件进行拍照

#### 2.6. 调用相册

#### 2.7. 播放音频的几个常用方法

#### 2.8. 播放视频的几个常用方法

### 3. 使用通知

新建一个 web 项目 MyNotify，创建包 sf.bzu.org，修改界面 activity\_main.xml 的内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context=".MainActivity">

    <Button android:id="@+id/send_notice"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Send notice"
        android:onClick="btn_click"/>

</LinearLayout>
```

修改界面 MainActivity 的代码，在其中添加方法：

```
public void btn_click(View view) {
    //注意：以下 6 点信息的设置，必须要写小图标，其余的可选择性省略
    NotificationCompat.Builder nb = new NotificationCompat.Builder(this);
    nb.setTitle("设置标题"+System.currentTimeMillis());
    //设置大图标
    nb.setLargeIcon(BitmapFactory.decodeResource(getResources(), R.mipmap.ic_launcher));
    nb.setText("内容文本部分 ");
    nb.setInfo("info 信息"); //设置 info 信息，即设置显示在时间右下角的文字
    nb.setSmallIcon(R.drawable.pic5); //设置小图标
    nb.when(System.currentTimeMillis()); //设置通知时间
    nb.setTicker("滚动提示文字");
    //nb.setOngoing(true); //永远不能手动移除
    nb.setOngoing(false); //可以移除通知
    NotificationManager manager = (NotificationManager) getSystemService(Service.NOTIFICATION_SERVICE);
```



```
num++;    //计数器
manager.notify(num, nb.build());
}
```

运行程序，查看效果。

#### 4. 发送短信

新建项目 **MySms**，用于发送短信。

##### 4.1. 修改布局界面

修改布局界面 **activity\_main.xml** 的代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context="sf.bzu.org.mysms.MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="信息"
            android:textSize="16sp"/>
        <EditText android:id="@+id/et_msg"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="16sp"/>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="电话"
            android:textSize="16sp"/>
        <EditText android:id="@+id/et_phone"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="16sp"/>
    </LinearLayout>

    <Button android:id="@+id/btn_jiekou"
        android:layout_width="match_parent"
```



```
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:text="使用短信接口发信息"
        android:onClick="btn_jiekou_click"/>
<Button android:id="@+id/btn_system"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:text="使用系统短信功能发信息"
        android:onClick="btn_system_click"/>
</LinearLayout>
```

#### 4.2. 修改主界面的交互代码

修改主界面的交互代码 **MainActivity.java** 如下：

```
import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.telephony.PhoneNumberUtils;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import java.util.List;

public class MainActivity extends AppCompatActivity {
    private EditText et_msg ;    //要发送的信息
    private EditText et_phone;  //要发给你的手机号

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        et_msg = (EditText)findViewById(R.id.et_msg);
        et_phone = (EditText)findViewById(R.id.et_phone);
    }

    //使用短信接口发信息按钮的点击事件处理
    public void btn_jiekou_click(View view){
        showMsg("使用短信接口发信息");
        String message = et_msg.getText().toString();
        String phoneNumber = et_phone.getText().toString();

        SmsManager smsManager = SmsManager.getDefault();
        // 判断发送内容字数(一件信息最多 70 字)
```



```
if(message.length() <= 70) {
    smsManager.sendTextMessage(phoneNumber, null, message, null, null);
}else{
    // SmsManger 类中 divideMessage 会将信息按每 70 字分割
    List<String> smsDivs = smsManager.divideMessage(message);
    for(String sms : smsDivs) {
        smsManager.sendTextMessage(phoneNumber, null, sms, null, null);
    }
}
Toast.makeText(this, "信息已发送", Toast.LENGTH_SHORT).show();
}

//使用系统短信功能发信息按钮的点击事件处理
public void btn_system_click(View view){
    showMsg("使用系统短信功能发信息");
    String message = et_msg.getText().toString();
    String phoneNumber = et_phone.getText().toString();

    if(PhoneNumberUtils.isGlobalPhoneNumber(phoneNumber)){
        Intent intent = new Intent(Intent.ACTION_SENDTO, Uri.parse("smsto:"+phoneNumber));
        intent.putExtra("sms_body",message);
        startActivity(intent);
    }
}

//使用信息提示对话框显示信息
private void showMsg(String str){
    Toast.makeText(this, str, Toast.LENGTH_SHORT).show();
}
}
```

#### 4.3. 添加权限

在系统的文件清单 **AndroidManifest.xml** 中添加如下的权限代码，位置及代码如下：

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="sf.bzu.org.mysms">
    <uses-permission android:name="android.permission.SEND_SMS"/>
```

#### 5. 查看系统短信

创建项目 **MyReadSMS**(注意：本项目要在真机中进行测试)。

##### 5.1. 添加背景图片

将背景图片 **bg.9.png** 拷贝到 **res/drawable** 文件夹下。

##### 5.2. 修改布局界面

修改布局界面 **activity\_main.xml** 的内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<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:orientation="vertical"
tools:context=".MainActivity">
<ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@drawable/bg"/>
<TextView
    android:id="@+id/tv_des"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="18dp"
    android:paddingLeft="20dp"
    android:text="读取到的系统短信信息如下:"
    android:textSize="20sp"
    android:visibility="invisible"/>
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:id="@+id/tv_sms"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/tv_des"
        android:lines="20"
        android:paddingLeft="20dp"
        android:paddingTop="10dp"
        android:textSize="16sp" />
    </ScrollView>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginBottom="28dp"
    android:layout_marginLeft="26dp"
    android:layout_marginStart="26dp"
    android:background="#D9D1FA"
    android:onClick="readSMS"
    android:padding="5dp"
    android:text="查看短信"
    android:textSize="30sp"/>
</RelativeLayout>
```

### 5.3. 添加短信实体类 SmsInfo



在 **MainActivity.java** 相同的文件夹下创建实体类 **SmsInfo.java**, 修改内容如下:

```
public class SmsInfo {  
    private int _id;           // 短信的主键  
    private String address;    // 发送地址  
    private int type;         // 类型  
    private String body;      // 短信内容  
    private long date;        // 时间  
    // 构造方法  
    public SmsInfo(int _id, String address, int type, String body, long date) {  
        this._id = _id;  
        this.address = address;  
        this.type = type;  
        this.body = body;  
        this.date = date;  
    }  
    //属性对应的 getter 与 setter 方法  
}
```

#### 5.4. 修改主界面交互代码

修改主界面 **MainActivity** 的交互代码如下:

```
import android.content.ContentResolver;  
import android.database.Cursor;  
import android.net.Uri;  
import android.os.Bundle;  
import android.support.v7.app.AppCompatActivity;  
import android.view.View;  
import android.widget.TextView;  
import java.util.ArrayList;  
import java.util.List;  
public class MainActivity extends AppCompatActivity {  
    private TextView tvSms;  
    private TextView tvDes;  
    private String text = "";  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        tvSms = (TextView) findViewById(R.id.tv_sms);  
        tvDes = (TextView) findViewById(R.id.tv_des);  
    }  
    //点击 Button 时触发的方法  
    public void readSMS(View view) {  
        //查询系统信息的 uri  
        Uri uri = Uri.parse("content://sms/");  
        //获取 ContentResolver 对象  
        ContentResolver resolver = getContentResolver();  
        //通过 ContentResolver 对象查询系统短信
```



```

Cursor cursor = resolver.query(uri, new String[]{ "_id","address",
        "type","body", "date"}, null, null, null);
List<SmsInfo> smsInfos = new ArrayList<SmsInfo>();
if (cursor != null && cursor.getCount() > 0) {
    tvDes.setVisibility(View.VISIBLE);
    while (cursor.moveToNext()) {
        int _id = cursor.getInt(0);
        String address = cursor.getString(1);
        int type = cursor.getInt(2);
        String body = cursor.getString(3);
        long date = cursor.getLong(4);
        SmsInfo smsInfo = new SmsInfo(_id, address, type, body, date);
        smsInfos.add(smsInfo);
    }
    cursor.close();
}
//将查询到的短信内容显示到界面上
for (int i = 0; i < smsInfos.size(); i++) {
    text += "手机号码: " + smsInfos.get(i).getAddress() + "\n";
    text += "短信内容: " + smsInfos.get(i).getBody() + "\n\n";
    tvSms.setText(text);
}
}
}

```

### 5.5. 添加读取短信的权限

在系统的文件清单 **AndroidManifest.xml** 中添加如下的权限代码，位置及代码如下：

```
<uses-permission android:name="android.permission.READ_SMS"/>
```

## 6. 调用摄像头方法一：调用系统相机的功能

创建项目 **MyCamera**。

### 6.1. 修改布局界面

修改布局界面 **activity\_main.xml** 的代码如下：

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/take_photo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Take Photo"/>

    <Button
        android:id="@+id/take_photo_v7"

```



```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Take Photo V7"/>
```

```
<ImageView
```

```
    android:id="@+id/picture"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center_horizontal"/>
```

```
</LinearLayout>
```

## 6.2. 修改主界面的交互代码

修改主界面 **MainActivity.java** 的交互代码如下：

```
import android.Manifest;
import android.app.Activity;
import android.content.ContentValues;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    public static final int TAKE_PHOTO = 1;
    public static final int CROP_PHOTO = 2;
    private static final int PHOTO_REQUEST_CAREMA = 3;
    private Button takePhoto;
    private Button takePhoto_v7;
    private ImageView picture;
    private Uri imageUri;
    private static File tempFile;
```





**@Override**

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    takePhoto = (Button) findViewById(R.id.take_photo);  
    picture = (ImageView) findViewById(R.id.picture);  
    takePhoto_v7 = (Button) findViewById(R.id.take_photo_v7);  
  
    takePhoto.setOnClickListener(this);  
    takePhoto_v7.setOnClickListener(this);  
}
```

**@Override**

```
public void onClick(View v) {  
    switch (v.getId()) {  
        case R.id.take_photo:  
            File outputImage = new File(Environment.getExternalStorageDirectory(),  
                "tempImage" + ".jpg");  
            try {  
                if (outputImage.exists()) {  
                    outputImage.delete();  
                }  
                outputImage.createNewFile();  
            } catch (IOException e) {  
                e.printStackTrace();  
            }  
            imageUri = Uri.fromFile(outputImage);  
            Intent intent = new Intent("android.media.action.IMAGE_CAPTURE");  
            intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);  
            startActivityForResult(intent, TAKE_PHOTO);  
            break;  
        case R.id.take_photo_v7: //v7 支持  
            openCamera(this);  
            break;  
    }  
}
```

**@Override**

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    switch (requestCode) {  
        case TAKE_PHOTO:  
            if (resultCode == RESULT_OK) {  
                Intent intent = new Intent("com.android.camera.action.CROP");  
                intent.setDataAndType(imageUri, "image/*");  
            }  
    }  
}
```



```
        intent.putExtra("scale", true);
        intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);
        startActivityForResult(intent, CROP_PHOTO); // 启动裁剪程序
    }
    break;
case CROP_PHOTO:
    if (resultCode == RESULT_OK) {
        try {
            Bitmap bitmap = BitmapFactory.decodeStream(getContentResolver()
                .openInputStream(imageUri));
            picture.setImageBitmap(bitmap);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
    break;
case PHOTO_REQUEST_CAREMA:
    if (resultCode == RESULT_OK) {
        Intent intent = new Intent("com.android.camera.action.CROP");
        intent.setDataAndType(imageUri, "image/*");
        intent.putExtra("scale", true);
        intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);
        startActivityForResult(intent, CROP_PHOTO); // 启动裁剪程序
    }
    break;
}
}

//android v7.0 版本的拍照处理
public void openCamera(Activity activity) {
    //获取系统版本
    int currentapiVersion = android.os.Build.VERSION.SDK_INT;
    // 激活相机
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    // 判断存储卡是否可以用，可用进行存储
    if (hasSdcard()) {
        SimpleDateFormat timeStampFormat = new SimpleDateFormat("yyyy_MM_dd_HH_mm_ss");
        String filename = timeStampFormat.format(new Date());
        tempFile = new File(Environment.getExternalStorageDirectory(),
            filename + ".jpg");
        if (currentapiVersion < 24) {
            // 从文件中创建 uri
            imageUri = Uri.fromFile(tempFile);
            intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);
        } else {
            //兼容 android7.0 使用共享文件的形式
```



```

ContentValues contentValues = new ContentValues(1);
contentValues.put(MediaStore.Images.Media.DATA, tempFile.getAbsolutePath());
//检查是否有存储权限，以免崩溃
if (ContextCompat.checkSelfPermission(this, Manifest.permission.WRITE_EXTERNAL_STORAGE)
    != PackageManager.PERMISSION_GRANTED) {
    //申请 WRITE_EXTERNAL_STORAGE 权限
    Toast.makeText(this, "请开启存储权限", Toast.LENGTH_SHORT).show();
    return;
}
imageUri = activity.getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI,
contentValues);
intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);
}
}
// 开启一个带有返回值的 Activity，请求码为 PHOTO_REQUEST_CAREMA
activity.startActivityForResult(intent, PHOTO_REQUEST_CAREMA);
}

/** 判断 sdcard 是否被挂载 */
public static boolean hasSdcard() {
    return Environment.getExternalStorageState().equals(
        Environment.MEDIA_MOUNTED);
}
}

```

### 6.3. 添加操作摄像头的权限

在系统的文件清单 **AndroidManifest.xml** 中添加如下的权限代码，位置及代码如下：

```

<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

```

## 7. 调用摄像头方法二：直接调用 Camera 硬件进行拍照

创建项目 **MyCameraDemo**。

### 7.1. 修改布局界面

修改布局界面 **activity\_main.xml** 文件的内容如下：

```

<?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"
    >
    <FrameLayout
        android:id="@+id/camera_preview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_weight="1"
    />

```



<Button

```
    android:id="@+id/button_capture"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="拍照" />
```

</LinearLayout>

## 7.2. 修改主界面的交互代码

修改主界面 **MainActivity.java** 的内容如下：

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import java.io.File;
import java.io.FileOutputStream;

import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.FrameLayout;
import android.content.Context;
import android.content.pm.PackageManager;
import android.hardware.Camera;
import android.hardware.Camera.AutoFocusCallback;
import android.hardware.Camera.PictureCallback;

public class MainActivity extends AppCompatActivity {
    protected static final String TAG = "main";
    private Camera mCamera;
    private CameraPreview mPreview;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mCamera = getCameraInstance();

        // 创建预览类，并与 Camera 关联，最后添加到界面布局中
        mPreview = new CameraPreview(this, mCamera);
        FrameLayout preview = (FrameLayout) findViewById(R.id.camera_preview);
        preview.addView(mPreview);

        Button captureButton = (Button) findViewById(R.id.button_capture);
        captureButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```



```
// 在捕获图片前进行自动对焦
mCamera.autoFocus(new AutoFocusCallback() {

    @Override
    public void onAutoFocus(boolean success, Camera camera) {
        // 从 Camera 捕获图片
        mCamera.takePicture(null, null, mPicture);
    }
});
}

/** 检测设备是否存在 Camera 硬件 */
private boolean checkCameraHardware(Context context) {
    if (context.getPackageManager().hasSystemFeature(
        PackageManager.FEATURE_CAMERA)) {
        // 存在
        return true;
    } else {
        // 不存在
        return false;
    }
}

/** 打开一个 Camera */
public static Camera getCameraInstance() {
    Camera c = null;
    try {
        c = Camera.open();
    } catch (Exception e) {
        Log.d(TAG, "打开 Camera 失败失败");
    }
    return c;
}

private PictureCallback mPicture = new PictureCallback() {

    @Override
    public void onPictureTaken(byte[] data, Camera camera) {
        // 获取 Jpeg 图片，并保存在 sd 卡上
        File pictureFile = new File("/sdcard/" + System.currentTimeMillis()
            + ".jpg");
        try {
            FileOutputStream fos = new FileOutputStream(pictureFile);
            fos.write(data);
        }
    }
}
```



```
        fos.close();
    } catch (Exception e) {
        Log.d(TAG, "保存图片失败");
    }
}
};
```

**@Override**

```
protected void onDestroy() {
    // 回收 Camera 资源
    if(mCamera!=null){
        mCamera.stopPreview();
        mCamera.release();
        mCamera=null;
    }
    super.onDestroy();
}
```

### 7.3. 添加预览处理类

直接调用 **Camera** 硬件进行拍照时，要自己处理预览功能，因此要在与 **MainActivity.java** 相同的包下创建一个 **CameraPreview.java** 类，并修改该类文件的内容如下：

```
import java.io.IOException;
```

```
import android.content.Context;
```

```
import android.hardware.Camera;
```

```
import android.util.Log;
```

```
import android.view.SurfaceHolder;
```

```
import android.view.SurfaceView;
```

```
/**
```

```
 * 定义一个预览类
```

```
 */
```

```
public class CameraPreview extends SurfaceView implements SurfaceHolder.Callback {
```

```
    private static final String TAG = "main";
```

```
    private SurfaceHolder mHolder;
```

```
    private Camera mCamera;
```

```
    public CameraPreview(Context context, Camera camera) {
```

```
        super(context);
```

```
        mCamera = camera;
```

```
        // 通过 SurfaceView 获得 SurfaceHolder
```

```
        mHolder = getHolder();
```

```
        // 为 SurfaceHolder 指定回调
```

```
        mHolder.addCallback(this);
```

```
        // 设置 Surface 不维护自己的缓冲区，而是等待屏幕的渲染引擎将内容推送到界面 在 Android3.0 之后弃用
```

```
        mHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
```



```
}

public void surfaceCreated(SurfaceHolder holder) {
    // 当 Surface 被创建之后，开始 Camera 的预览
    try {
        mCamera.setPreviewDisplay(holder);
        mCamera.startPreview();
    } catch (IOException e) {
        Log.d(TAG, "预览失败");
    }
}

public void surfaceDestroyed(SurfaceHolder holder) {

}

public void surfaceChanged(SurfaceHolder holder, int format, int w, int h) {
    // Surface 发生改变的时候将被调用，第一次显示到界面的时候也会被调用
    if (mHolder.getSurface() == null){
        // 如果 Surface 为空，不继续操作
        return;
    }

    // 停止 Camera 的预览
    try {
        mCamera.stopPreview();
    } catch (Exception e){
        Log.d(TAG, "当 Surface 改变后，停止预览出错");
    }

    // 在预览前可以指定 Camera 的各项参数

    // 重新开始预览
    try {
        mCamera.setPreviewDisplay(mHolder);
        mCamera.startPreview();

    } catch (Exception e){
        Log.d(TAG, "预览 Camera 出错");
    }
}
```

#### 7.4. 添加操作摄像头的权限

在系统的文件清单 **AndroidManifest.xml** 中添加如下的权限代码，位置及代码如下：

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```



## 8. 调用相册

创建项目 **MyAlbum**。

### 8.1. 修改布局界面

修改布局界面 **activity\_main.xml** 文件的内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context=".MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:text="点击选择图片" />

    <ImageView
        android:id="@+id/image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```

### 8.2. 修改主界面的交互代码

修改主界面 **MainActivity.java** 的内容如下：

```
import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.provider.MediaStore;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {

    //调用系统相册-选择图片
    private static final int IMAGE = 1;
    //所需权限 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```





```
}

public void onClick(View v) {
    //调用相册
    Intent intent = new Intent(Intent.ACTION_PICK,
        android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    //获取图片路径
    if (requestCode == IMAGE && resultCode == Activity.RESULT_OK && data != null) {
        Uri selectedImage = data.getData();
        String[] filePathColumns = { MediaStore.Images.Media.DATA };
        Cursor c = getContentResolver().query(selectedImage, filePathColumns, null, null, null);
        c.moveToFirst();
        int columnIndex = c.getColumnIndex(filePathColumns[0]);
        String imagePath = c.getString(columnIndex);
        showImage(imagePath);
        c.close();
    }
}

//加载图片
private void showImage(String imaePath){
    Bitmap bm = BitmapFactory.decodeFile(imaePath);
    ((ImageView)findViewById(R.id.image)).setImageBitmap(bm);
}
}
```

### 8.3. 添加操作摄像头的权限

在系统的文件清单 **AndroidManifest.xml** 中添加如下的权限代码，位置及代码如下：

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

## 9. 播放音频的几类常用方法

创建项目 **MyAudio**。

### 9.1. 修改布局界面

修改布局界面 **activity\_main.xml** 文件的内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="sf.bzu.org.myaudio.MainActivity">
```



```
<Button android:id="@+id/btn_mediaPlayer"
    android:layout_width='match_parent'
    android:layout_height='wrap_content'
    android:text="MediaPlayer 播放 RAW 音频"
    android:textSize="16sp"
    android:onClick="btn_mediaPlayer_click"/>
<Button android:id="@+id/btn_mediaPlayer_yuan"
    android:layout_width='match_parent'
    android:layout_height='wrap_content'
    android:text="MediaPlayer 播放音频源音频"
    android:textSize="16sp"
    android:onClick="btn_mediaPlayer_yuan_click"/>
```

```
<Button android:id="@+id/btn_soundPool"
    android:layout_width='match_parent'
    android:layout_height='wrap_content'
    android:text="SoundPool 播放音频"
    android:textSize="16sp"
    android:onClick="btn_soundPool_click"/>
```

```
<Button android:id="@+id/btn_audioTrack"
    android:layout_width='match_parent'
    android:layout_height='wrap_content'
    android:text="AudioTrack 播放音频"
    android:textSize="16sp"
    android:onClick="btn_audioTrack_click"/>
```

```
<Button android:id="@+id/btn_ringtones"
    android:layout_width='match_parent'
    android:layout_height='wrap_content'
    android:text="Ringtone 播放音频"
    android:textSize="16sp"
    android:onClick="btn_ringtones_click"/>
```

```
</LinearLayout>
```

## 9.2. 修改主界面的交互代码

修改主界面 **MainActivity.java** 的内容如下：

```
import android.media.AudioAttributes;
import android.media.AudioFormat;
import android.media.AudioManager;
import android.media.AudioTrack;
import android.media.MediaPlayer;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.media.SoundPool;
import android.net.Uri;
```



```
import android.os.Build;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Toast;

import java.io.IOException;
import java.io.InputStream;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;

public class MainActivity extends AppCompatActivity {
    //MediaPlayer 变量定义区域开始
    private MediaPlayer mMediaPlayer=null;    //MediaPlayer 实例
    private boolean bMediaPlayerFlag = false;    //MediaPlayer 是否已经播放：默认未播放
    private MediaPlayer mMediaPlayer1 = null;    //MediaPlayer 实例 1
    private boolean bMediaPlayerFlag1 = false;    //MediaPlayer 是否已经播放：默认未播放
    //MediaPlayer 变量定义区域结束

    //SoundPool 变量定义区域开始
    private SoundPool mSoundPool = null;    //SoundPool 实例
    private boolean bSoundPoolFlag = false;    //SoundPool 播放状态：默认未播放
    private int soundId;    //soundPool 播放文件的 id
    private int streamId;    //要播放 soundPool 的 streamId
    private boolean bLoaded = false;
    //SoundPool 变量定义区域结束

    //AudioTrack 变量定义区域开始
    private InputStream mFile;
    private byte mBuffer[];
    private short mLen;
    AudioTrack mTrack;

    int format, channels, rate, bits, dataSize;
    int bufsize;
    boolean isRecordPlaying;
    //AudioTrack 变量定义区域结束

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        initSoundPool();
    }
}
```



//MediaPlayer 播放按钮事件处理

//1.播放 Raw 文件夹下的元数据“我们不一样”，这里只演示播放与停止方法功能

```
public void btn_mediaPlayer_click(View view){
    if(mMediaPlayer==null) { //实例未创建，则创建实例
        mMediaPlayer = MediaPlayer.create(this, R.raw.buyiyang);
    }
    if(!bMediaPlayerFlag){ //未播放
        bMediaPlayerFlag = true; //修改播放状态为播放
        mMediaPlayer.start();
    }else{
        bMediaPlayerFlag = false; //修改播放状态为未播放
        mMediaPlayer.stop();
        mMediaPlayer = null;
    }
}
```

//2: MediaPlayer 播放音频源文件

```
public void btn_mediaPlayer_yuan_click(View view) throws Exception{
```

// 用 的 源 文 件 ( “ 感 谢 有 你 ” ) :

<http://dl.stream.qqmusic.qq.com/C400002MuGJI1qea1b.m4a?vkey=3F744A5A67374775B44E3C1D054FD63C841B9D6FFDE8779908730C019399B10346D08EF06C816305AE34221AFA368BFCB40128F74396F024&guid=1484455433&uin=804156311&fromtag=66>

```
if(mMediaPlayer1==null) { //实例未创建，则创建实例
```

//1.直接使用 raw 播放

```
//mMediaPlayer1 = MediaPlayer.create(this, R.raw.buyiyang);
```

//2.设置网络资源播放

```
mMediaPlayer1 = new MediaPlayer(); //实例化
```

```
mMediaPlayer1.reset(); //重置
```

```
//mMediaPlayer1.setDataSource("http://dl.stream.qqmusic.qq.com/C400002MuGJI1qea1b.m4a?vkey=3F744A5A67374775B44E3C1D054FD63C841B9D6FFDE8779908730C019399B10346D08EF06C816305AE34221AFA368BFCB40128F74396F024&guid=1484455433&uin=804156311&fromtag=66");
```

```
mMediaPlayer1.setDataSource("http://mp3.9ku.com/m4a/411603.m4a"); //伤不起
```

//3.设置 assets 资源播放

```
//AssetFileDescriptor fd = getAssets().openFd("buyiyang.mp3");
```

```
//mMediaPlayer1.reset();
```

```
//mMediaPlayer1.setDataSource(fd.getFileDescriptor(), fd.getStartOffset(), fd.getLength());
```

//需使用异步缓冲

```
//mMediaPlayer1.prepareAsync(); //经过测试使用这个方法不可以使用
```

```
mMediaPlayer1.prepare(); //经过测试，使用这个方法可以使用
```

```
}
```

```
if(!bMediaPlayerFlag1){ //未播放
```

```
bMediaPlayerFlag1 = true; //修改播放状态为播放
```

```
mMediaPlayer1.start();
```

```
}else{
```



```
bMediaPlayerFlag1 = false; //修改播放状态为未播放
mMediaPlayer1.stop();
mMediaPlayer1 = null;
}
}

//初始化 SoundPool,
// 1.因为 SoundPool 是异步的,所以要先在 create 方法中进行初始化与 load
// 2.然后再在点击按钮事件中调用
// 3.还要注意 SoundPool 播放的时间特别短,大概就 1s 左右
private void initSoundPool(){
    if(mSoundPool == null) {
        //mSoundPool = new SoundPool(100, AudioManager.STREAM_MUSIC, 5);//构建对象
        //当前系统的 SDK 版本大于等于 21(Android 5.0)时
        if (Build.VERSION.SDK_INT >= 21) {
            SoundPool.Builder builder = new SoundPool.Builder();
            //传入音频数量
            builder.setMaxStreams(1);
            //AudioAttributes 是一个封装音频各种属性的方法
            AudioAttributes.Builder attrBuilder = new AudioAttributes.Builder();
            //设置音频流的合适的属性
            attrBuilder.setLegacyStreamType(AudioManager.STREAM_MUSIC);
            //加载一个 AudioAttributes
            builder.setAudioAttributes(attrBuilder.build());
            mSoundPool = builder.build();
        }
        //当系统的 SDK 版本小于 21 时
        else { //设置最多可容纳 10 个音频流,音频的品质为 5
            mSoundPool = new SoundPool(10, AudioManager.STREAM_SYSTEM, 5);
        }
    }
    soundId= mSoundPool.load(this,R.raw.buyiyang,1);//加载资源,得到 soundId

    mSoundPool.setOnLoadCompleteListener(new SoundPool.OnLoadCompleteListener() {
        @Override
        public void onLoadComplete(SoundPool soundPool, int sampleId, int status) {
            Toast.makeText(MainActivity.this, "" + sampleId, Toast.LENGTH_SHORT).show();
            bLoaded = true;
        }
    });
}

//SoundPool 播放按钮事件处理
public void btn_soundPool_click(View view) throws Exception{
    if(bLoaded){
        if(!bSoundPoolFlag){
```



```
bSoundPoolFlag =true;

streamId= mSoundPool.play(soundId, 1.0f, 1.0f, 0, 0, 1.0f);//播放，得到 StreamId
}else{
    bSoundPoolFlag = false;
    mSoundPool.stop(streamId);//暂停
    //mSoundPool.release();    //释放资源
}
}else{
    Toast.makeText(this, "SoundPool 还未加载完成", Toast.LENGTH_SHORT).show();
}
}

//AudioTrack 播放按钮事件处理
public void btn_audioTrack_click(View view){
    playStream();
    //playStatic();
}
/**
 * -- 流式播放音频（用于播放大音频）
 */
void playStream() {
    try {
        mFile = getAssets().open("raw/10.wav");
    } catch (Exception e) {
    }
    if (mTrack == null && mFile != null) {
        try {
            bufsize = AudioTrack.getMinBufferSize(8000, AudioFormat.CHANNEL_OUT_STEREO,
AudioFormat.ENCODING_PCM_16BIT);
            mTrack = new AudioTrack(AudioManager.STREAM_MUSIC
, 8000, //每秒 8K 个点(采样率)
AudioFormat.CHANNEL_OUT_STEREO, //双声道
AudioFormat.ENCODING_PCM_16BIT, //一个采样点 16 比特-2 个字节(设置音频数据块是 8 位还是 16 位-采样精度)
bufsize,
AudioTrack.MODE_STREAM);
        } catch (Exception e) {
        }
    }
    if (mTrack != null) {
        mTrack.play();
        int readResult = 0;
        byte buffer[] = new byte[bufsize];
        while (true) {
            try {
                readResult = mFile.read(buffer, 0, bufsize);
            } catch (IOException e1) {
            }
        }
    }
}
```



```
    }
    if (readResult != -1) {
        // AudioTrack.PLAYSTATE_PAUSED (暂停状态),
        mTrack.write(buffer, 0, readResult);
    } else {
        break;
    }
}

// -- 输出流可以重复使用
try {
    //      mFile.close();
    mFile.reset();
    readHeader(mFile);
} catch (IOException e) {
    e.printStackTrace();
}

// -- 重置 mTrack 状态
mTrack.stop();
// -- 释放底层资源。
//mTrack.release();
}
}

/**
 * -- 静态播放 (需要提前获取输入流, 延迟低, 用于播放小音频)
 */
void playStatic() {
    try {
        mFile = getAssets().open("raw/paobujieshu.wav"); //可以切换其它的 wav 文件查看效果
    } catch (Exception e) {
    }
    if (mTrack == null && mFile != null) {
        try {
            readHeader(mFile);
            mBuffer = new byte[1024 * 1024];
            mLen = (short) mFile.read(mBuffer);
            // AudioTrack.MODE_STREAM(不能重复播放) ;AudioTrack.MODE_STATIC(可以重复播放)
            mTrack = new AudioTrack(AudioManager.STREAM_MUSIC
                , 8000, //每秒 8K 个点(采样率)
                AudioFormat.CHANNEL_OUT_STEREO, //双声道
                AudioFormat.ENCODING_PCM_16BIT, //一个采样点 16 比特-2 个字节(设置音频数据块是 8 位还是 16 位-采样精度)
                mLen,
                AudioTrack.MODE_STATIC);
        } catch (Exception e) {
        }
    }
}
```



```
}
    if (mTrack != null) {
        short written = (short) mTrack.write(mBuffer, 0, mLen);
        mTrack.play();
    }
}
/**
 * -- 获取音频 wav 文件头信息
 * @param wavStream
 * @throws IOException
 */
public void readHeader(InputStream wavStream) throws IOException {

    ByteBuffer buffer = ByteBuffer.allocate(44);
    buffer.order(ByteOrder.LITTLE_ENDIAN);
    wavStream.read(buffer.array(), buffer.arrayOffset(), buffer.capacity());
    buffer.rewind();
    buffer.position(buffer.position() + 20);
    format = buffer.getShort(); // --格式
    channels = buffer.getShort(); // --声道
    rate = buffer.getInt(); // --采样率
    buffer.position(buffer.position() + 6);
    bits = buffer.getShort(); // --采样精度
    while (buffer.getInt() != 0x61746164) { // "data" marker
        int size = buffer.getInt();
        wavStream.skip(size);
        buffer.rewind();
        wavStream.read(buffer.array(), buffer.arrayOffset(), 8);
        buffer.rewind();
    }
    dataSize = buffer.getInt(); // -- 数据长度
    Log.d("TAG", "readHeader: format=" + format + "--channels=" + channels + "--rate=" + rate + "--bits=" + bits + "--dataSize="
+ dataSize);
}

//Ringtone 播放按钮事件处理
public void btn_ringtone_click(View view){
    Uri uri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_RINGTONE) ;
    Ringtone mRingtone = RingtoneManager.getRingtone(this,uri);
    mRingtone.play();
}

@Override
protected void onDestroy() {
    super.onDestroy();
}
```





```
mSoundPool.release(); //释放 SoundPool 的资源
```

```
}  
}
```

### 9.3. 添加操作摄像头的权限

在系统的文件清单 **AndroidManifest.xml** 中添加如下的权限代码，位置及代码如下：

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>  
<uses-permission android:name="android.permission.INTERNET"/>
```

## 10. 播放视频的几类常用方法

创建项目 **MyVideo**。

### 10.1. 修改布局界面

修改布局界面 **activity\_main.xml** 文件的内容如下：

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    tools:context="sf.bzu.org.myvideo.MainActivity">  
  
    <Button android:id="@+id/btn_system"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:textSize="16sp"  
        android:text="使用系统自带的播放器"  
        android:onClick="btn_system_click"/>  
    <Button android:id="@+id/btn_videoView"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:textSize="16sp"  
        android:text="使用 VideoView"  
        android:onClick="btn_videoView_click"/>  
    <Button android:id="@+id/btn_mediaPlayer"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:textSize="16sp"  
        android:text="使用 MediaPlayer 和 SurfaceView"  
        android:onClick="btn_mediaPlayer_click"/>  
    <VideoView android:id="@+id/video_view"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent" />  
</LinearLayout>
```

### 10.2. 修改主界面的交互代码

修改主界面 **MainActivity.java** 的内容如下：

```
import android.content.Intent;  
import android.media.MediaPlayer;
```



```
import android.net.Uri;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.MediaController;
import android.widget.Toast;
import android.widget.VideoView;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    //使用系统自带的播放器按钮事件处理
    public void btn_system_click(View view) throws Exception{
        //Uri uri = Uri.parse(Environment.getExternalStorageDirectory().getPath()+"/Test_Movie.m4v");
        /******这个操作需要在 DDMS 中添加文件操作权限，并添加 root 权限，操作步骤如下：
         * 1.打开 sdk\platform-tools，进入这个文件夹
         * 2.按着 shift 键+右击-->在右键菜单中选择“在此处打开命令窗口”
         * 3.使用 adb shell 命令
         * 4.接着使用 su，转换为超级用户
         * 5.使用 chmod -R 777 /storage 为 storage 文件添加操作权限
         * 6.使用 exit 命令退出 su 用户，再使用 exit 命令退出 adb
         * 7.使用 adb root，为运行的模拟器添加 root 权限
         * 8.在 ddms 中，找到 storage/0 文件夹
         * 9.使用 pull file to device 命令按钮，将本地的文件 xiatian.mkv 上传到 ddms 中的 storage/0 文件夹
         * 10.再使用下面的这段代码，就可以进行全屏播放了
         *****/
        Uri uri = Uri.parse(Environment.getExternalStorageDirectory().getPath()+"/xiatian.mkv");
        //Uri uri = Uri.parse("android.resource://" + getApplicationContext().getPackageName() + "/" + R.raw.xiatian);
        //调用系统自带的播放器
        Intent intent = new Intent(Intent.ACTION_VIEW);
        Log.v("URI:.....", uri.toString());
        intent.setDataAndType(uri, "video/mp4");
        try {
            startActivity(intent);
        } catch (Exception ex){
            ex.printStackTrace();
            Toast.makeText(this, "没有合适的播放器", Toast.LENGTH_SHORT).show();
        }
    }

    //上面为使用本机的文件
    //下面的内容是直接播放网络文件
```



```
// String video = "http://www.nandudu.com/hls/course/video/2/test.m3u8";
// Intent openVideo = new Intent(Intent.ACTION_VIEW);
// openVideo.setDataAndType(Uri.parse(video), "video/*");
// startActivity(openVideo);
}

//使用 VideoView 按钮事件处理
public void btn_videoView_click(View view){
    //Uri uri = Uri.parse(Environment.getExternalStorageDirectory().getPath()+"/Test_Movie.m4v");
    Uri uri = Uri.parse("android.resource://" + getApplicationContext().getPackageName() + "/" + R.raw.xiatian);
    VideoView videoView = (VideoView)this.findViewById(R.id.video_view);
    videoView.setMediaController(new MediaController(this));
    videoView.setVideoURI(uri);
    videoView.start();
    videoView.requestFocus();
}

//使用 MediaPlayer 按钮事件处理，未解决的问题：进行第 2 个界面后，马上返回主界面
public void btn_mediaPlayer_click(View view) throws Exception{
    Intent intent = new Intent(this, MediaPlayerActivity.class);
    startActivity(intent);
}
}
```

### 10.3. 创建 MediaPlayerActivity 并修改 activity\_media\_player.xml 文件

为了使用 **MediaPlayer** 进行视频播放，创建一个新的名为 **MediaPlayerActivity** 的 **Activity**，修改布局界面 **activity\_media\_player.xml** 文件的内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context="sf.bzu.org.myvideo.MediaPlayerActivity">

    <SurfaceView android:id="@+id/video_surface"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

### 10.4. 修改 MediaPlayerActivity.java 文件内容

修改 **MediaPlayerActivity.java** 类文件的内容如下：

```
import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Environment;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
```



```
import java.io.File;
import java.io.IOException;

import android.media.MediaPlayer;
import android.util.Log;
import android.view.MotionEvent;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
import android.view.View;

/**
 * 该实例中使用 MediaPlayer 完成播放，同时界面使用 SurfaceView 来实现 *
 * 这里我们实现 MediaPlayer 中很多状态变化时的监听器 *
 * 使用 MediaPlayer 时，也可以使用 MediaController 类，但是需要实现 MediaController.mediaController 接口
 * 实现一些控制方法。 *
 * 然后，设置 controller.setMediaPlayer(),setAnchorView(),setEnabled(),show()就可以了，这里不再实现
 * @author Administrator *
 */
public class MediaPlayerActivity extends AppCompatActivity{
    private MediaPlayer mediaPlayer;
    private SurfaceView surfaceView;
    //读取本地文件
    private File file=new File("/storage/0/", "xiatian.mkv");
    private String myFileName = Environment.getExternalStorageDirectory().getPath()+"/xiatian.mkv";
    //访问网络视频
    private String uri='http://112.253.22.157/17/z/z/y/u/zzyuasjwufnqerzvyxgkuigrkcatxr/hc.yinyuetai.com/D046015255134077DDB3ACA0D7E68D45.flv';

    private static final int REQUEST_EXTERNAL_STORAGE = 1;
    private static String[] PERMISSIONS_STORAGE = {
        Manifest.permission.READ_EXTERNAL_STORAGE,
        Manifest.permission.WRITE_EXTERNAL_STORAGE
    };

    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        this setContentView(R.layout.activity_media_player);

        surfaceView = (SurfaceView) findViewById(R.id.video_surface);
        mediaPlayer = new MediaPlayer();
        //获取 SurfaceHolder 可以通过该接口来操作 SurfaceView 中的 Surface
        SurfaceHolder surfaceHolder = surfaceView.getHolder();
        //设置 MediaPlayer 的准备监听
        mediaPlayer.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
            @Override
```



```
public void onPrepared(MediaPlayer mp) {
    //准备完成后播放
    mediaPlayer.start();
}
});

surfaceHolder.addCallback(new SurfaceHolder.Callback() {
    //当 SurfaceView 中 Surface 创建时回掉
    //该方法表示 Surface 已经创建完成，可以在该方法中进行绘图操作
    @Override
    public void surfaceCreated(SurfaceHolder holder) {
        mediaPlayer.reset();
        try {
            //设置视屏文件图像的显示参数
            mediaPlayer.setDisplay(holder);

            /*** 1. file.getAbsolutePath()本地视频,
             * 在 android6.0 以上的系统中播放本地视频时，只在文件清单中添加权限是不行的
             * 要使用如下的代码进行权限检查，并打开权限
             * 使用这段代码时，会弹出权限允许的提示对话框，确定后，即可以播放了
             * *****/

            int permission = ActivityCompat.checkSelfPermission(MediaPlayerActivity.this, Manifest.permission.WRITE_
EXTERNAL_STORAGE);

            if (permission != PackageManager.PERMISSION_GRANTED) {
                // We don't have permission so prompt the user
                ActivityCompat.requestPermissions(
                    MediaPlayerActivity.this,
                    PERMISSIONS_STORAGE,
                    REQUEST_EXTERNAL_STORAGE
                );
            }
            Log.i("MediaPlayerActivity", myFileName);
            mediaPlayer.setDataSource(myFileName);
            mediaPlayer.prepare();
            //2.uri 网络视频,网络视频要结合 mediaPlayer.prepareAsync 使用
            //mediaPlayer.setDataSource(MediaPlayerActivity.this, Uri.parse(uri));
            //prepare();表示准备工作同步进行，（准备工作在 UI 线程中进行）
            //当播放网络视频时，如果网络不要 会报 ARN 所以不采用该方法
            //mediaPlayer.prepare();
            //异步准备 准备工作在子线程中进行 当播放网络视频时候一般采用此方法
            //mediaPlayer.prepareAsync();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
});
```



```
//当 SurfaceView 的大小发生改变时候触发该方法
@Override
public void surfaceChanged(SurfaceHolder holder, int format, int width, int height) {

}
//Surface 销毁时回掉
//当 Surface 销毁时候，同时把 MediaPlayer 也销毁
@Override
public void surfaceDestroyed(SurfaceHolder holder) {
    if (mediaPlayer!=null) {
        mediaPlayer.stop();
        //释放资源
        mediaPlayer.release();
    }
}
});
//设置 surfaceView 点击监听
surfaceView.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
                if (mediaPlayer.isPlaying()) {
                    mediaPlayer.pause();
                } else {
                    mediaPlayer.start();
                }
                break;
        }
        //返回 True 代表事件已经处理了
        return true;
    }
});
}
```

### 10.5. 添加操作摄像头的权限

在系统的文件清单 **AndroidManifest.xml** 中添加如下的权限代码，位置及代码如下：

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```