

Android程序设计

音视频录制和播放

2019.2.19

isszym sysu.edu.cn

内容

- SD卡
- 录音和音乐播放
- 视频录制和播放

SD卡

方法1、直接访问：

```
private String path = "/sdcard/DCIM/Camera/a.mp3"
```

方法2、先取得路径再访问：

```
public String getSDPath() {  
    File sdDir = null;  
    boolean sdCardExist  
        = Environment.getExternalStorageState()  
            .equals(android.os.Environment.MEDIA_MOUNTED); //判断sd卡是否存在  
    if (sdCardExist) {  
        sdDir = Environment.getExternalStorageDirectory(); //获取跟目录  
        return sdDir.toString();  
    }  
    return null;  
}  
  
private String path = getSDPath() + "/DCIM/Camera/a.mp3"
```

[参考](#)

[参考](#)

音乐播放

项目名: NewMusicPlayer

```
public class MainActivity extends AppCompatActivity {
    private MediaPlayer mediaPlayer = new MediaPlayer();           // 媒体播放器对象
    private String path;                                           // 音乐文件路径
    private boolean isPause;                                       // 暂停状态
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        path="/mnt/sdcard/Music/a1.mp3";                          // 或 path= getSDPath() +"/Music/a1.mp3";
    }
    public void Play(View vw) {
        int position = 0;    File file = new File(path);
        if (!file.exists()) {
            Toast.makeText(this, "音频文件路径错误:"+path, Toast.LENGTH_LONG).show();
            return;
        }
        try {
            mediaPlayer.reset();                                   // 把各项参数恢复到初始状态
            mediaPlayer.setDataSource(path);                       // 设置播放的视频源
            mediaPlayer.prepare();                                 // 进行缓冲
            mediaPlayer.setOnPreparedListener(new PreparedListener(position));
        } catch (Exception e) {
            Toast.makeText(this, "播放音频路径错误"+e.getMessage(), Toast.LENGTH_LONG).show();
        }
    }
}
```

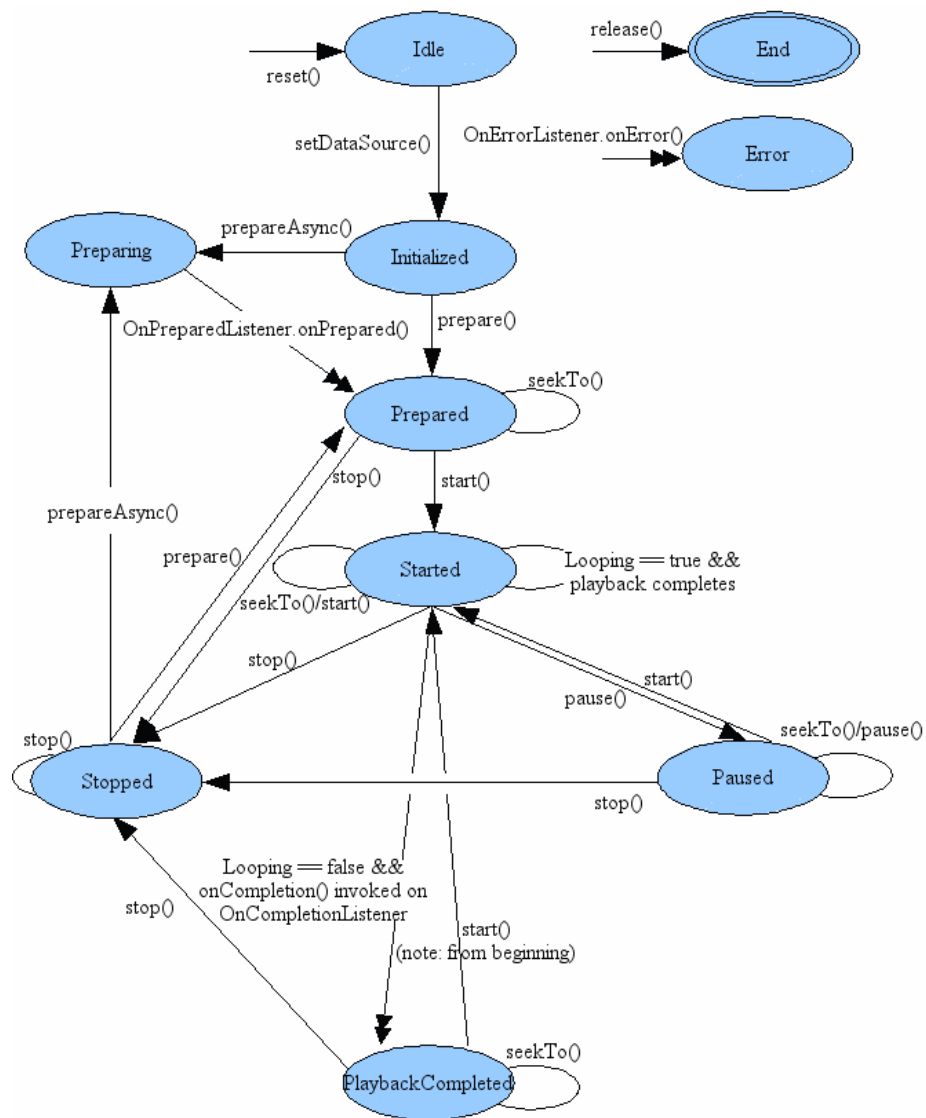
```
public void Pause(View vw) {
    if (mediaPlayer != null && mediaPlayer.isPlaying()) {
        mediaPlayer.pause();
        isPause = true;
    }
}
public void Stop(View vw) {
    if (mediaPlayer != null) {
        mediaPlayer.stop();
        mediaPlayer.release();
    }
}
@Override
public void onDestroy() {
    if (mediaPlayer != null) {
        mediaPlayer.stop();
        mediaPlayer.release();
    }
}
private final class PreparedListener implements MediaPlayer.OnPreparedListener {
    private int positon;
    public PreparedListener(int positon) {
        this.positon = positon;
    }
    @Override
    public void onPrepared(MediaPlayer mp) {    // 当音乐准备好的时候开始播放
        mediaPlayer.start();                    // 开始播放
        if (positon > 0) {                        // 如果音乐不是从头播放
            mediaPlayer.seekTo(positon);
        }
    }
}
```

```
public String getSDPath() {  
    File sdDir = null;  
    boolean sdCardExist =  
Environment.getExternalStorageState().equals(android.os.Environment.MEDIA_MOUNTED);  
    if(sdCardExist) {  
        sdDir = Environment.getExternalStorageDirectory();//获取跟目录  
    }  
    return sdDir.toString();  
}  
}
```

[参考](#) [参考](#)

```
mp.reset();  
mp = create(MainActivity.this, R.raw.a1);  
mp.setAudioStreamType(AudioManager.STREAM_MUSIC);  
mp.setVolume(12f, 12f); //声音调不了  
mp.start();
```

[参考](#)



```
<?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:orientation="horizontal">

    <Button
        android:text="Play"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="Play" />

    <Button
        android:text="pause"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="Pause" />

    <Button
        android:text="stop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/btnPause"
        android:onClick="Stop" />
</LinearLayout>
```



视频播放

项目名: **VideoPlayer**

```
public class MainActivity extends AppCompatActivity {  
    private final String TAG = "main";  
    private EditText et_path;  
    private SurfaceView sv;  
    private MediaPlayer mediaPlayer;  
    private SeekBar seekBar;  
    private int currentPosition = 0;  
    private boolean isPlaying;  
    private Button btn_play, btn_pause, btn_replay, btn_stop;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        seekBar = (SeekBar) findViewById(R.id.seekBar);  
        sv = (SurfaceView) findViewById(R.id.sv);  
        et_path = (EditText) findViewById(R.id.et_path);  
        btn_play = (Button) findViewById(R.id.btn_play);  
        btn_pause = (Button) findViewById(R.id.btn_pause);  
        btn_replay = (Button) findViewById(R.id.btn_replay);  
        btn_stop = (Button) findViewById(R.id.btn_stop);  
        btn_play.setOnClickListener(click);  
        btn_pause.setOnClickListener(click);  
        btn_replay.setOnClickListener(click);  
        btn_stop.setOnClickListener(click);  
    }  
}
```

```
private View.OnClickListener click = new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.btn_play:
                play(0);
                break;
            case R.id.btn_pause:
                pause();
                break;
            case R.id.btn_replay:
                replay();
                break;
            case R.id.btn_stop:
                stop();
                break;
            default:
                break;
        }
    }
};

protected void stop() {
    if (mediaPlayer != null && mediaPlayer.isPlaying()) {
        mediaPlayer.stop();
        mediaPlayer.release();
        mediaPlayer = null;
        btn_play.setEnabled(true);
        isPlaying = false;
    }
}
```

```

protected void play(final int msec) {
    String path = et_path.getText().toString().trim();
    File file = new File(path);
    if (!file.exists()) {
        Toast.makeText(this, "视频文件路径错误", Toast.LENGTH_LONG).show();
        return;
    }
    try {
        mediaPlayer = new MediaPlayer();
        mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
        mediaPlayer.setDataSource(file.getAbsolutePath());
        mediaPlayer.setDisplay(sv.getHolder()); // 设置显示视频的SurfaceHolder
        Log.i(TAG, "开始装载");
        mediaPlayer.prepareAsync();
        mediaPlayer.setOnPreparedListener(new OnPreparedListener() {

            @Override
            public void onPrepared(MediaPlayer mp) {
                Log.i(TAG, "装载完成");

                // 首先取得video的宽和高
                int vWidth = mediaPlayer.getVideoWidth();
                int vHeight = mediaPlayer.getVideoHeight();

                // 该LinearLayout的父容器 android:orientation="vertical" 必须
                LinearLayout linearLayout = (LinearLayout) findViewById(R.id.LL);
                int lw = linearLayout.getWidth();
                int lh = linearLayout.getHeight();
            }
        });
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

```
// 如果video的宽或者高超出了当前屏幕的大小, 则要进行缩放
float wRatio = (float) vWidth / (float) lw;
float hRatio = (float) vHeight / (float) lh;

// 选择大的一个进行缩放
float ratio = Math.max(wRatio, hRatio);
vWidth = (int) Math.ceil((float) vWidth / ratio);
vHeight = (int) Math.ceil((float) vHeight / ratio);

// 设置surfaceView的布局参数
ViewGroup.LayoutParams lp= sv.getLayoutParams();
lp.width= vWidth;
lp.height= vHeight;
sv.setLayoutParams(lp);

mediaPlayer.start();
mediaPlayer.seekTo(msec); // 按照初始位置播放
// 设置进度条的最大进度为视频流的最大播放时长
seekBar.setMax(mediaPlayer.getDuration());
```

```

        // 开始线程, 更新进度条的刻度
new Thread() {
    @Override
    public void run() {
        try {
            isPlaying = true;
            while (isPlaying) {
                int current = mediaPlayer
                    .getCurrentPosition();
                seekBar.setProgress(current);

                sleep(500);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}.start();
btn_play.setEnabled(false);
});

mediaPlayer.setOnCompletionListener(new OnCompletionListener() {
    @Override
    public void onCompletion(MediaPlayer mp) {
        // 在播放完毕被回调
        btn_play.setEnabled(true);
    }
});

```

```

mediaPlayer.setOnErrorListener(new OnErrorListener() {
    @Override
    public boolean onError(MediaPlayer mp, int what, int extra) {
        // 发生错误重新播放
        play(0);
        isPlaying = false;
        return false;
    }
});
} catch (Exception e) {
    e.printStackTrace();
}
}

protected void replay() {
    if (mediaPlayer != null && mediaPlayer.isPlaying()) {
        mediaPlayer.seekTo(0);
        Toast.makeText(this, "重新播放", Toast.LENGTH_LONG).show();
        btn_pause.setText("暂停");
        return;
    }
    isPlaying = false;
    play(0);
}
}

```

```

/**
 * 暂停或继续
 */
protected void pause() {
    if (btn_pause.getText().toString().trim().equals("继续")) {
        btn_pause.setText("暂停");
        mediaPlayer.start();
        Toast.makeText(this, "继续播放", Toast.LENGTH_LONG).show();
        return;
    }
    if (mediaPlayer != null && mediaPlayer.isPlaying()) {
        mediaPlayer.pause();
        btn_pause.setText("继续");
        Toast.makeText(this, "暂停播放", Toast.LENGTH_LONG).show();
    }
}
}

```

```

<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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

```

```

<EditText
    android:id="@+id/et_path"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="/mnt/sdcard/Movies/a1.mp4" />

```

```

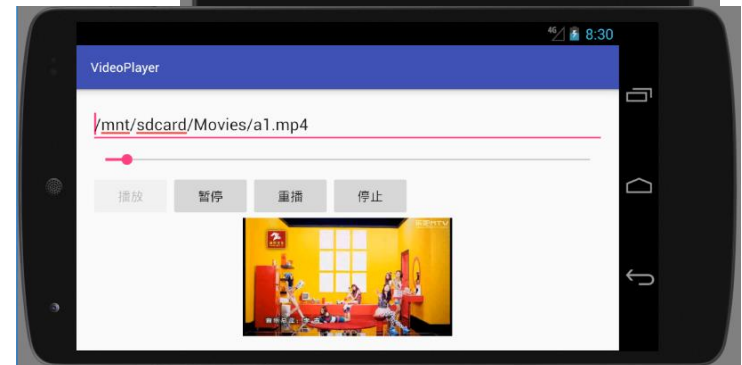
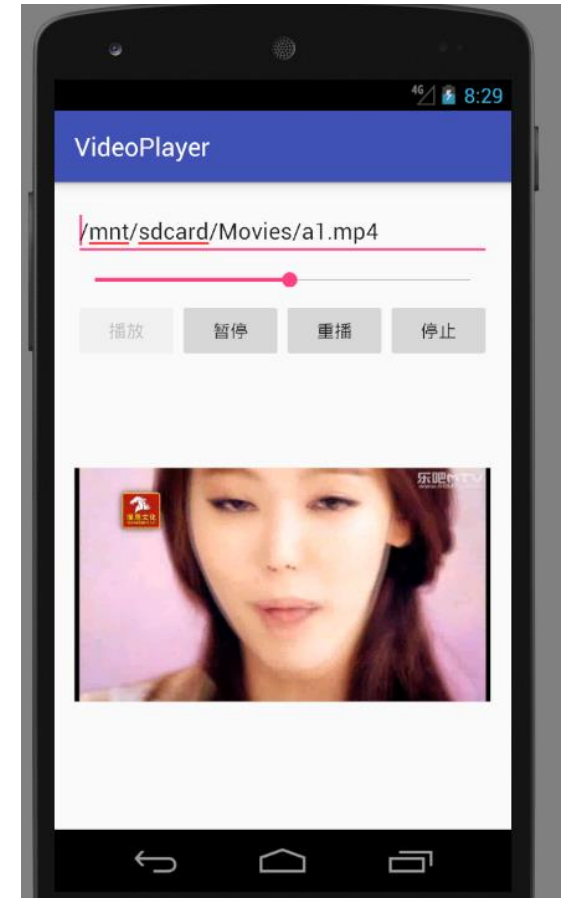
<SeekBar
    android:id="@+id/seekBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

```

```

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >

```




```
<Button
    android:id="@+id/btn_play"
    android:layout_width="0dip"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="播放" />
```

```
<Button
    android:id="@+id/btn_pause"
    android:layout_width="0dip"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="暂停" />
```

```
<Button
    android:id="@+id/btn_replay"
    android:layout_width="0dip"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="重播" />
```

```
<Button
    android:id="@+id/btn_stop"
    android:layout_width="0dip"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="停止" />
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:gravity="center"
    android:id="@+id/ll">
    <SurfaceView
        android:id="@+id/sv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    </LinearLayout>
</LinearLayout>
```