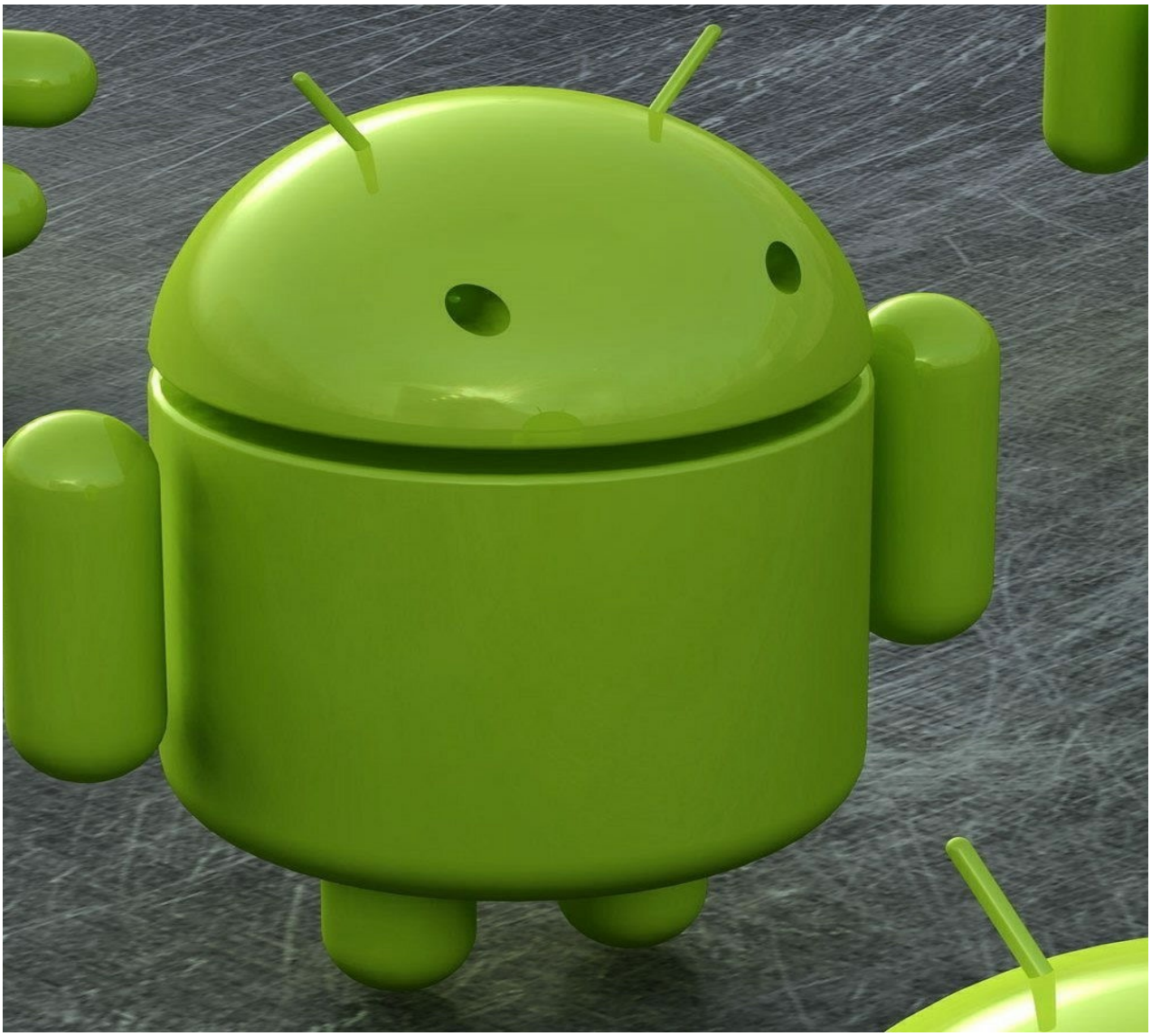


实验六

服务与多线程--简单音乐播放器

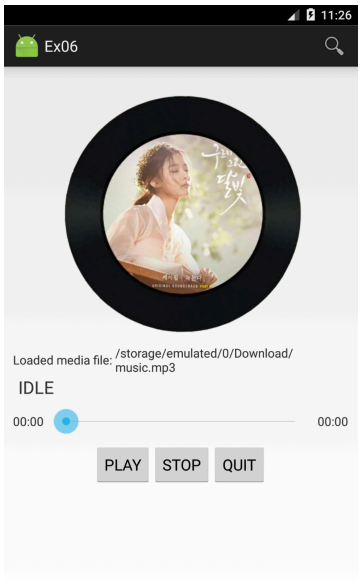
【实验目的】

1. 学会使用 MediaPlayer；
2. 学会简单的多线程编程，使用 Handle 更新 UI；
3. 学会使用 Service 进行后台工作；
4. 学会使用 Service 与 Activity 进行通信。

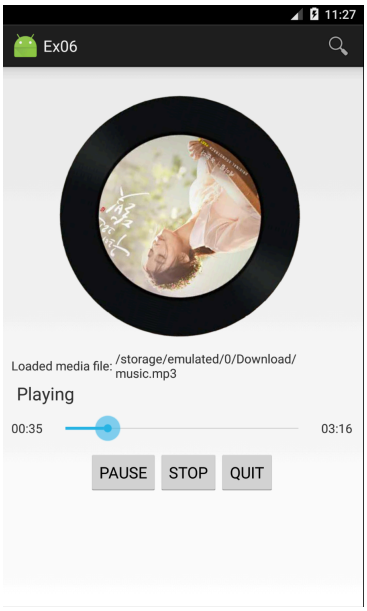
【实验内容】

实现一个简单的播放器，要求功能有：

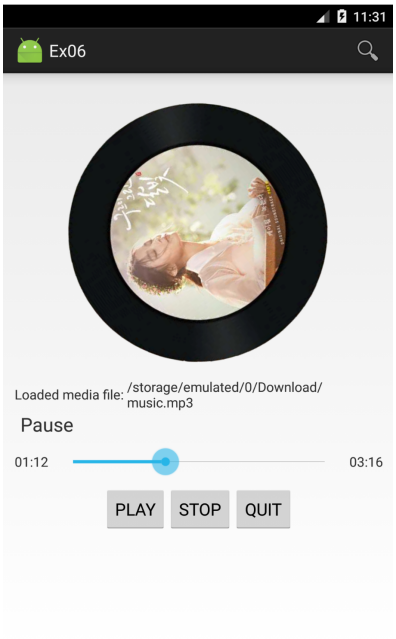
1. 播放、暂停，停止，退出功能；
2. 后台播放功能；
3. 进度条显示播放进度、拖动进度条改变进度功能；
4. 播放时图片旋转，显示当前播放时间功能；



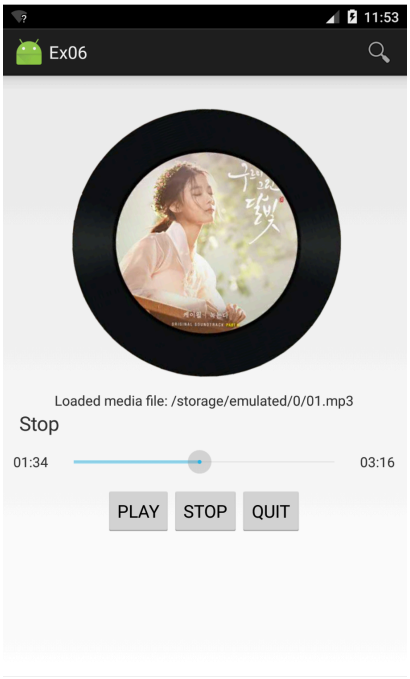
打开程序主页面



开始播放



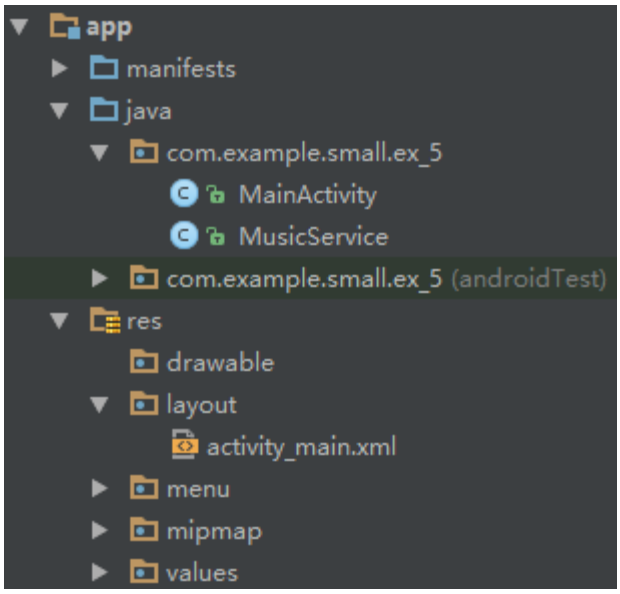
暂停



停止

【参考内容】

1. 参考文件目录如下：



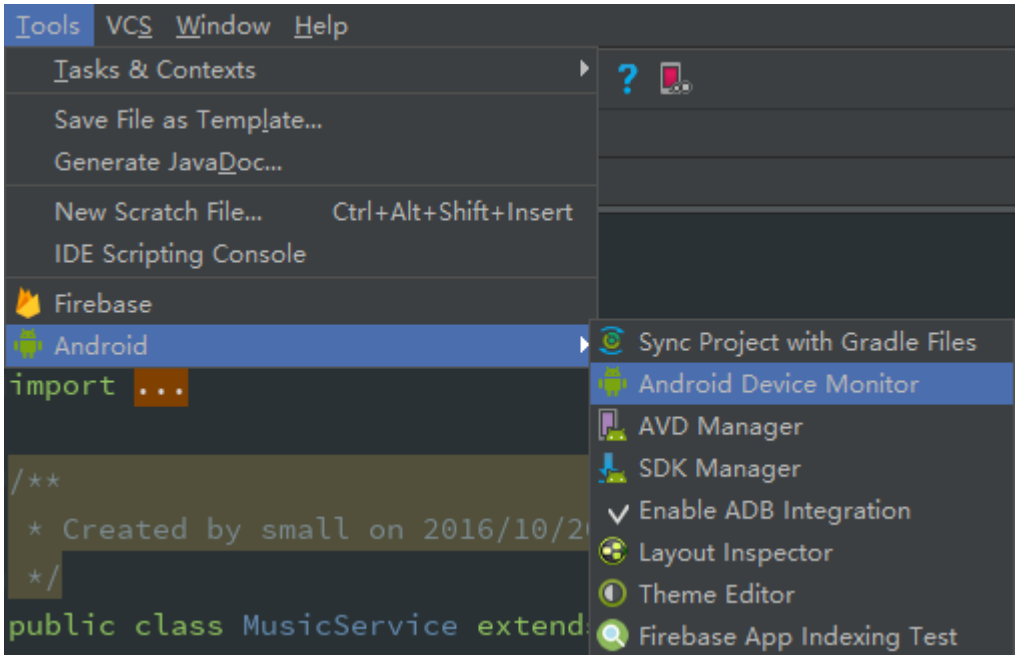
2. MediaPlayer 介绍

常用方法：

函数	功能	使用时机
setDataSource(String)	设置音频文件路径 进入初始化状态	MediaPlayer 对象已创建
prepare()	进入就绪状态	已初始化或停止
start()	进入播放状态	已就绪
pause()	进入暂停状态	正在播放
stop()	进入停止状态	正在播放或暂停
isPlaying()	检查是否正在播放	任意正常状态
getCurrentPosition()	获取当前已播放的毫秒数	已就绪
getDuration()	获取文件的时间长度(毫秒)	已就绪
release()	停止播放并释放资源	任何时候

3. 向虚拟机添加文件

首先打开 Android Device Monitor，如下图：



使用自己手机进行调试的同学，注意下把文件拷到内置 SD 卡而不是外置 SD 卡会比较方便。要使用外置的 SD 卡时，注意下文件路径的获取。这是相关的路径获取方法:http://blog.sina.com.cn/s/blog_5da93c8f0102vcam.html

4. 使用 MediaPlayer

创建对象：初始化：

注意下获取的文件路径，若是使用模拟器的如下，若是使用自己手机的内置 SD 卡则使用：`Environment.getExternalStorageDirectory() + "/data/K.Will-Melt.mp3`

```
try {
    mPlayer.setDataSource(filepath);
    mPlayer.prepare();
    state = STATE.PREPARED;
    result = true;
} catch (IOException e) {
    e.printStackTrace();
}
```

播放/暂停：

```
if (mp.isPlaying()) {
    mp.pause();
} else {
    mp.start();
}
```

停止：

```
if (mp != null) {
    mp.stop();
    try {
        mp.prepare();
        mp.seekTo(0);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

5. Service 的使用

创建 service 类，实现 MediaPlayer 的功能。

注意在 AndroidManifest.xml 文件里注册 Service：

```
<service android:name=".MusicService" android:exported="true"/>
```

通过 Binder 来保持 Activity 和 Service 的通信(写在 service 类)：

```
public final IBinder binder = new MyBinder();
public class MyBinder extends Binder {
    MusicService getService() {
        return MusicService.this;
    }
}
```

在 Activity 中调用 bindService 保持与 Service 的通信(写在 activity 类)：
Activity 启动时绑定 Service：

```
Intent intent = new Intent(this, MusicService.class);
bindService(intent, sc, BIND_AUTO_CREATE);
```

bindService 成功后回调 onServiceConnected 函数，通过 IBinder 获取 Service 对象，实现 Activity 与 Service 的绑定：

```
private ServiceConnection sc = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName name, IBinder service) {
        ms = ((MusicService.MyBinder)service).getService();
    }
```

停止服务时，必须解除绑定，写入退出按钮中：

```
mHandler.removeCallbacks(mRunnable);
unbindService(sc);
try {
    MainActivity.this.finish();
    System.exit(0);
} catch (Exception e) {
    e.printStackTrace();
}
```

此时，在 Activity 的 onCreate 方法中执行上述与 Service 通信的方法后，即可实现后台播放。点击退出按钮，程序会退出，音乐停止；返回桌面，音乐继续播放。

6. Handler 的使用

Handler 与 UI 是同一线程，这里可以通过 Handler 更新 UI 上的组件状态，Handler 有很多方法，这里使用比较简便的 post 和 postDelayed 方法。

使用 Seekbar 显示播放进度，设置当前值与最大值：

```
seekBar.setProgress(ms.mp.getCurrentPosition());
seekBar.setMax(ms.mp.getDuration());
```

定义 Handler：run 函数中进行更新 seekbar 的进度在类中定义简单日期格式，用来显示播放的时间，用 time.format 来格式所需要的数据，用来监听进度条的滑动变化：

```
private SimpleDateFormat time = new SimpleDateFormat("mm:ss");
```

```
seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
```

```
Handler mHandler = new Handler();
Runnable mRunnable = new Runnable() {
    @Override
    public void run() {
```

【检查内容】

1. 布局显示是否正常
2. 播放，暂停，停止功能是否可用，界面显示是否正常
3. 是否可以后台播放
4. 播放时是否显示当前播放时间，位置，以及图片是否旋转

【提交说明】

1、deadline：下一次实验课前一天晚上 12 点 （不是两周！！！）

2、提交位置：ftp://222.200.185.18:1890/ 对应文件夹下

3、命名与目录要求： 附件命名及格式要求：学号_姓名 _labX.zip（姓名中文拼音均可）
重复提交命名格式要求：学号_姓名 _labX_Vn.zip 目录结构：

```
14331111_huashen_lab1 --
                       |
                       --  lab1实验报告.pdf
                       |
                       --  lab1_code（包含项目代码文件）
```

其中项目代码文件为项目文件夹，提交之前先 clean