**中山大学移动信息工程学院本科生实验报告**

**（2017年秋季学期）**

课程名称：移动应用开发 任课教师： 郑贵峰

|  |  |  |  |
| --- | --- | --- | --- |
| 年级 | 2015 | 专业（方向） | 移动信息工程（M2） |
| 学号 | 15352193 | 姓名 | 梁公勋 |
| 电话 | 15066299571 | Email | 1104695389@qq.com |
| 开始日期 |  | 完成日期 |  |

# 实验题目

服务与多线程--简单音乐播放器

# 实现内容

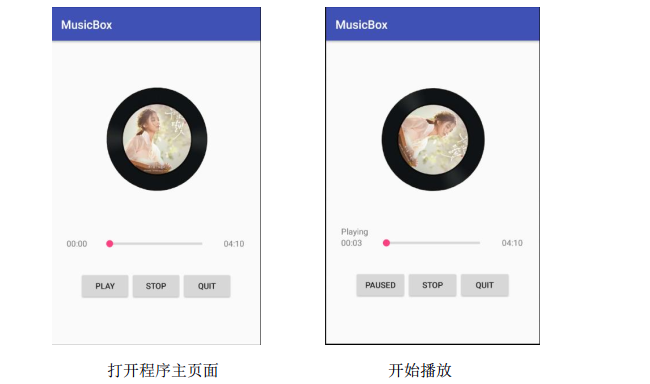
实现一个简单的播放器，要求功能有：

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；

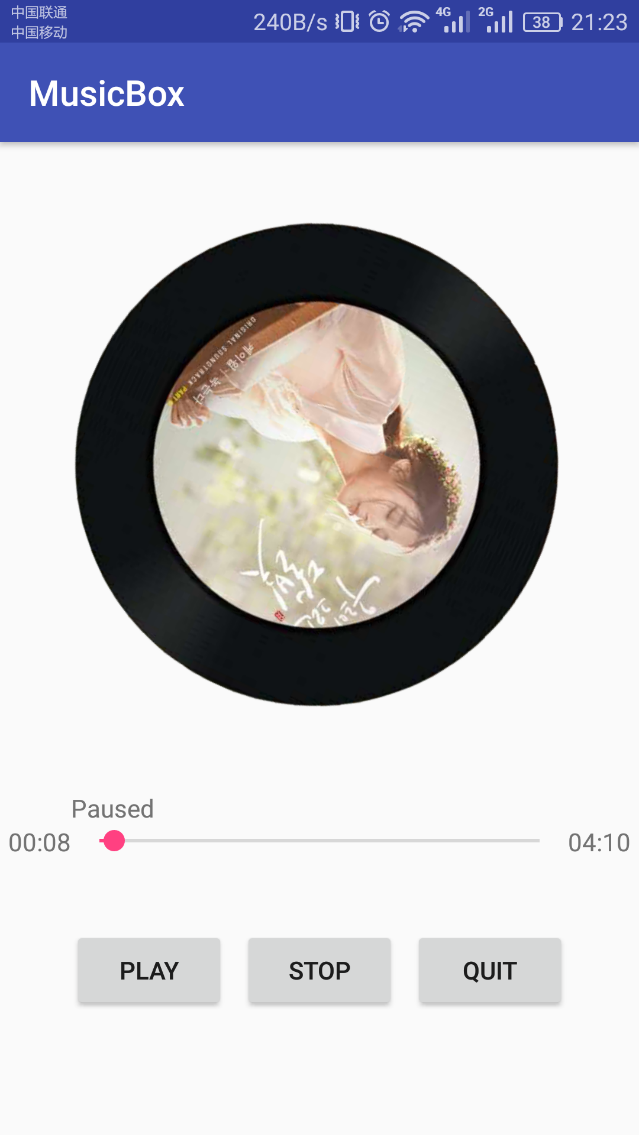
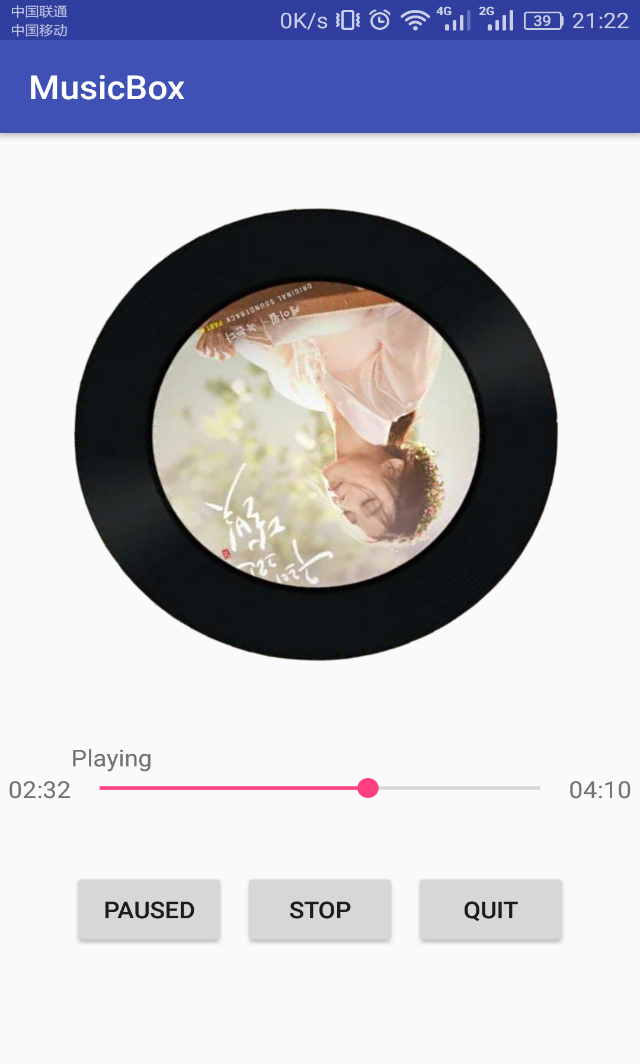
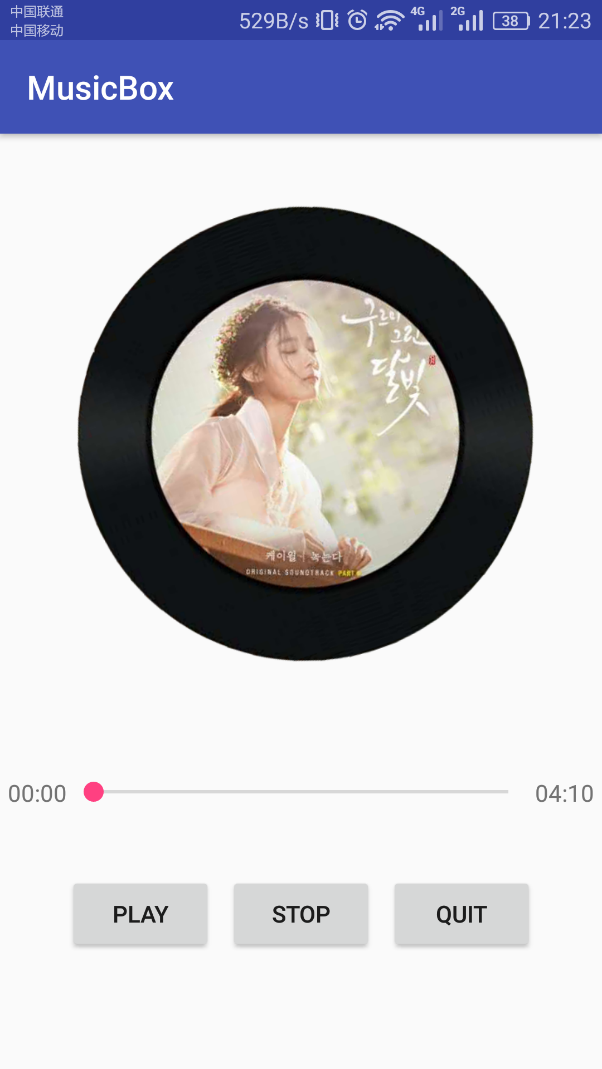
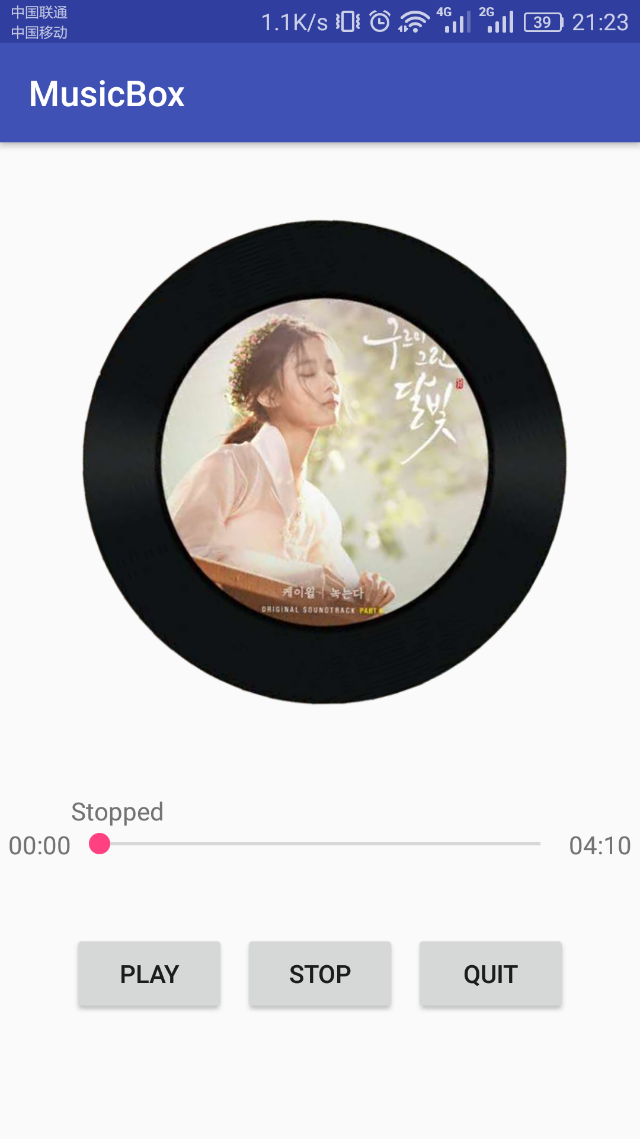
4. 播放时图片旋转，显示当前播放时间功能；





# 课堂实验结果

1. 实验截图

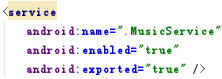
1. 实验步骤以及关键代码

**一：Service以及MediaPlayer部分**

这次实验中我们要创建一个service后台运行，并在其中放置一个MediaPlayer播放音乐。

首先，新建一个service类，这里我们不要新建Java文件，而是直接新建一个service。这样，与service注册有关的代码会自动补全，而不需要我们自己填写。

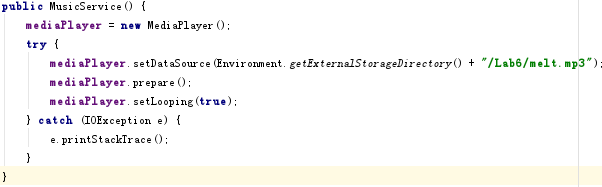
Androidmanifest文件中与service注册有关的代码如下：



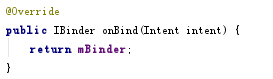
我们创建的MusicService是继承自Service类中的。创建完成后，我们可以重写其构造函数，还有一个必须重载的函数onBind。Service中一共有两个变量，一个是我们自定义的一个继承自IBinder类的对象，这个对象复制Activity与Service之间的通信。还有一个是MediaPlayer的对象，用来播放音乐。



我们首先在构造函数中，初始化MediaPlayer对象，用以播放音乐。使用setDataSource函数，设置播放源的路径，这里设置的是手机内部内存的路径。调用prepare函数，是MediaPlayer处于就绪状态。调用setLooping函数，使播放处于循环状态。



然后是在Service中必须重载的函数IBinder，我们就返回我们负责通信的那个自定义类的对象



Service中最后一部分就是我们自定义的IBinder类，负责与音乐播放相关的一些控制。

这个类必须要重载父类中的onTransact函数，与这个函数对应的是transact函数。onTransact函数是在service中重载，负责接收自Activity中传入的数据，并返回service中的数据给Activity。而在activity中的是transact函数，负责接收service中传来的数据。发送activity中的数据。

这个函数一个有四个形参。第一个int变量是识别码，用来区分传递消息的场合。第二个和第三个形参都是Parcel类型，这个类型代表了内存中的一段缓冲区，用来发送数据和接受数据。

在如下的我们重载的这个函数中，使用switch函数，根据识别码code的不同，分别处理。

接下来讲解不同code情况下，这个onTransact函数的处理。



Code为100是回传数据用的，不对MediaPlayer作任何处理。这里我们使用一个Bundle容器类，存放我们要发送的数据。包括，代表音乐是否正在播放的布尔变量，音乐的总时间和音乐播放的当前时间。然后使用writeBundle函数将其写入缓冲区reply，发送给Activity。

Code为101是处理播放及暂停键，当activity以识别码101向service发送数据时，即为点击了play/pause按钮。然后我们使用mediaplayer中的isPlayong函数判断一下当前音乐的播放状态，如果是正在播放，就使用pause函数将其暂停，如果不是就使用start函数开始播放音乐。

Code为102是处理停止键。当stop键被点击，activity就好会识别码102向service中发送数据。在这种情况下，我们调用mediaPlayer中的stop函数停止播放音乐。然后使用prepa函数重新使之恢复就绪状态，并使用seekTO函数，将其播放时间设回到0.

Code为103,是处理我们拖动进度条改变音乐进度的情况。当activity发生进度条的改变时，就会向service发送一个数据，也就是seekbar的位置数据。然后我们直接根据这个位置，使用seekTo函数调整音乐的播放时间即可。

**二、获取读取手机内存的权限。**

在播放音乐时我们需要从内存中获取音乐文件，所以要使App具有读取手机内存的权限。首先在注册文件中添加如下代码请求读取内存权限。



接下来还有在activity中动态获取读取内存的权限。我们在activity中调用如下函数获取读取手机内存的权限。首先我们得到读取内存权限的状态，然后判断这个权限是否被授权了，如果没有被授权，就需要请求其权限。

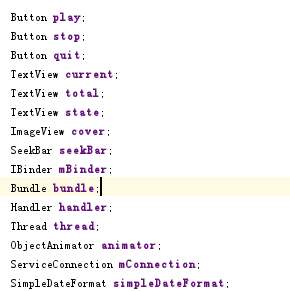


**三、activity中的变量以及与service建立连接的部分。**

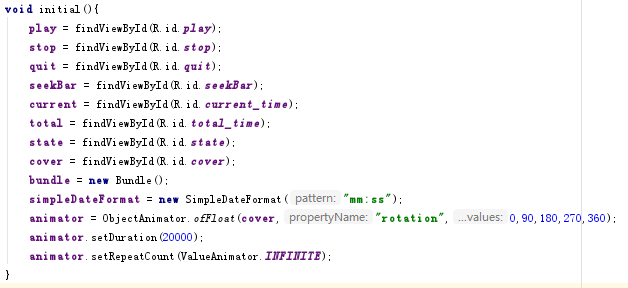
首先，先列一下我们activity中除去控件变量之外用到的变量。

mBinder是activicity与service通信的类。bundle是负责接收从service中回调的bundle容器。handle负责处理UI更新，thread计算UI参数。animatot是负责图片旋转的动画类。mConnection则是使activity与service建立通信而用。SimpleDataFormat则是格式化字符串为我们输出的时间。

其余的则是需要与布局文件中的控件绑定的变量。

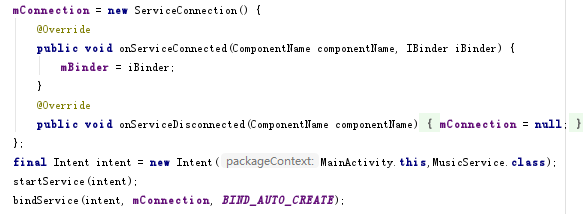


然后是在初始化函数中初始化了一部分变量。首先是将控件类与布局文件中的控件绑定。另外，我们为simpleDataFormat设置我们要展示的时间格式。然后为animator设置动画效果。使用ofFloat函数设置动画效果。这个函数的第一个形参是设置动画的控件，第二个是动画类型，rotation是旋转动画。最后是一串数字代表动画的变化范围。然后使用setDuration设置动画时间，使用setRepeatCount函数设置动画重复次数为无数次。



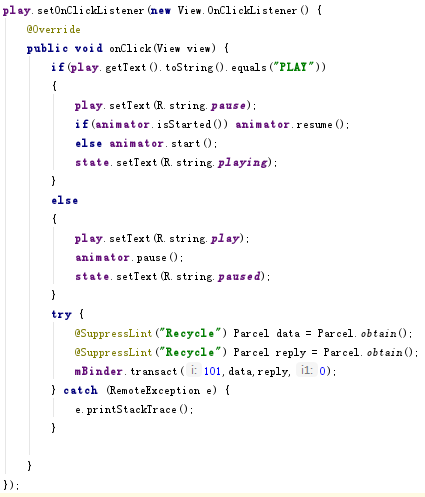
接下来是将activity与service进行绑定。我们为mConnection new一个serviceConnection对象。在这个对象中重写onServiceConnected函数和onServiceDisconnected函数。绑定成功后会回调onServiceConnected函数，然后我们就使用IBinder获取service对象，以进行通信。

然后使用intent，并放进去我们要跳转到的服务。使用startService函数启动service，然后使用bindService函数，将activity和service绑定。



**四、activity中绑定的各种监听器**

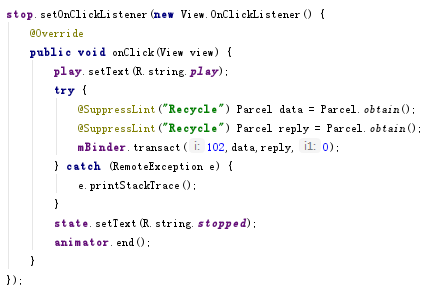
首先是为play/pause按钮绑定点击监听器。因为这个按钮有播放和暂停两种功能，所以监听到点击事件后，要分情况处理。判断的方法则是直接读取按钮上的文字进行判断。如果是点击的播放功能，我们先将这个按钮的文字修改为PAUSED,然后判断是否启用了动画效果，如果启用了就将其动画效果也暂停，然后将展示播放状态的那个TextView设为Playing。



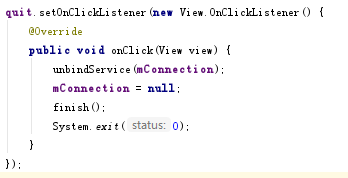
如果是点击暂停的功能，则将按钮文字设为PLAY，恢复动画的播放，并将播放状态修改为Paused。

然后通过IBinder的transact方法，向service发送请求，区别码为101，然后service根据区别码进行处理。

接下来是停止按钮stop的点击事件。监听到点击事件后使用transact向service发送区别码102，然后service会根据区别码进行停止播放的处理。接着将播放文字修改为Stopped，然后停止动画播放。



然后是退出按钮quit的点击事件。监听到点击事件后，解除activity与service的绑定，然后将连接变量mConnection置空，结束这个activity，退出APP。



最后是进度条seekbar的点击事件。Seekbar的监听器可以监听三种点击事件，一个是进度条的进度改变，第二个是开始点击进度条，第三是结束点击进度条。这里我们只用到了第一个。

在这个点击事件中，首先根据变量b判断是否是人为拖动，而不是使用setProgress设置的进度条改变。如果是人为拖动，我们将进度条位置写入缓冲区，然后使用transact函数发送给service。Service中会根据这个的区别码设置音乐播放进度。其余两个点击事件的方法空置就好



**五、使用handle和thread更新UI。**

在APP中，我们要对UI进行实时的更新，主要包括进度条的改变和播放时间的变化。

首先，新建一个thread，计算UI更改的数据。这个thread中是一个while的死循环，以保证它能一直处于运行状态。然后调用thread的sleep函数使其休眠0.1s，这样更新UI的频率就是0.1s更新一次。

接着，如果mBinder不为空，使用transact函数与service通信，获取service中的数据，然后使用readBundle将其中的数据读取到activity中的bundle中。

然后使用handle中的obtainMessage函数更新UI。

最后还要调用start函数启用这个线程。



接着我们还有新建一个handle对象，并重写其中的handleMessage方法，进行UI的更新。首先我们读取bundle中与音乐是否正在播放的布尔变量。如果音乐正在播放，我们将play/paused按钮的文字设为paused。然后读取bundle中的音乐正播放到的时间，使用seekbar的setProgress函数，设置进度条的更新。读取bundle中的音乐总时间，使用setMax为seekbar设置总长度。

然后分别使用这两个时间新建一个表示日期的Data对象。使用我们之前设置好的格式化字符串格式的simpleDataFormat将其从时间转化为字符串。然后使用这两个字符串为当前当前时间和总时间设置文本。

最后如果音乐不是正在播放，就将play/paused按钮的文字设为play。



1. 实验遇到困难以及解决思路

**1、activity无法从service中回调数据**

因为这个问题，seekbar不能动。后来发现，我所有的transact函数都用了同一个data和reply缓冲区。后来，每次使用transact之前都新建一个单独的缓冲区调用就可以了。

**2、拖动seekbar与更新UI中对seekbar的进度条修改冲突。**

就是我拖动seekbar时，因为UI更新的原因，总是会跳回到音乐播放点。后来，直接在拖动的监听事件中写入修改音乐播放进度。但是还有判断是不是人为拖动的才可以，不然音乐就播放不了。这里还需要用到进度改变的单击事件中的一个判断是不是人为改变的布尔变量。

# 课后实验结果

如果有，就是那个半成品的动态读取手机内存权限吧。

## 实验思考及感想

在这次实验中，我们学习了thread和四大组件service的使用，也算是初步了解了音乐播放APP后台运行的原理。对于APP中的音乐播放类mediaPlayer也算是会使用了。不得不感叹，安卓中对这些类的封装真的是强大。另外，这次还使用seekbar，所以对seekbar也熟悉了很多。还有学习使用了动画效果类ObjectAnimator，这个类用起来倒是和html中的动画有些相

似。

最后感觉，其实一些细节上还是可以优化的，不过这次没时间了。

作业要求：

1. 命名要求: 学号\_姓名\_实验编号，例如15330000\_林XX\_lab1。
2. 实验报告提交格式为pdf。
3. 实验内容不允许抄袭，我们要进行代码相似度对比。如发现抄袭，按0分处理。