Android音乐播放器设计文档

前几天写了一个Android视频播放器，觉得不是太完善，于是又做了一个Android音乐播放器，播放的是《美丽之物》的口琴版，我个人感觉非常好听，这次用到了MediaPlayer,感觉还挺好用的，功能挺强。

好了，我们进入正题，我个人习惯先写界面然后再添加功能，所以首先我们看一下我的界面：



界面做的比较简单，最上方有4个按钮，分别是“播放”，“暂停”，“从头开始”，“循环播放”，中间有一个进度条，音乐播放时进度条会跟着动，拖动进度条可以控制音乐播放的位置，最下面左边会显示当前时间，右边显示了总时间。

界面写好了，我们要开始写逻辑部分了，首先我们当然要初始化各组件，然后我们需要初始化MediaPlayer,API中说有两种方法可以初始化MediaPlayer,一种是直接用MediaPlayer.create()方法来初始化，这样就不用setDataSourse了,这里需要将mp3文件存放在res下的raw文件夹下，调用的时候可以直接用R.raw.文件名调用，后缀名可以省略，就像这样：

mediaplayer=MediaPlayer.create(this,R.raw.prettything);

但很不幸的事发生了，用这种方法初始化MediaPlayer，可以build成功，也可以生成apk，但安装后打开apk会闪退，我在网上查了查也没找到有用的方法，所以我用了另外一种方法来初始化，就是先new一个MediaPlayer的对象，然后再调用setDataSourse()设置MediaPlayer的音乐的路径，就像这样：

mediaplayer=new MediaPlayer();

AssetFileDescriptor fileDescriptor=getAssets().openFd("prettything.mp3");  
mediaplayer.setDataSource(fileDescriptor.getFileDescriptor(),fileDescriptor.getStartOffset(), fileDescriptor.getLength());

这次我们不是把mp3文件存到raw文件夹下，而是把文件存到assets文件夹下，然后用上述方法setDataSourse()。

接下来我们还要调用MediaPlayer的prepare方法来准备：

mediaplayer.prepare();

这样，MediaPlayer就算初始化好了。

现在MediaPlayer已经准备好了，我们来添加一下播放暂停等功能，我们给各按钮设置事件监听器，当点击播放按钮时，MediaPlayer就会调用start()方法播放音乐，当点击暂停按钮时MediaPlayer就会调用pause()方法，当点击重新开始按钮时，MediaPlayer就会调用seekTo(0)方法将音乐进度调整为0，当点击循环播放按钮时，MediaPlayer就会调用setLooping(true)将MediaPlayer设置为循环播放。

到现在，已经可以实现播放音乐的功能，但我不想就这样简单的播放音乐，我想要显示音乐的总时间，音乐的当前时间，还需要进度条来显示播放进度并可以拖动进度条更改音乐的时间，就像大多数音乐播放器做的那样。

我们先来显示音乐的总时间，我们可以调用MediaPlayer的getDuration()方法得到音乐的总时间，这个总时间单位是毫秒，我们需要将时间变为秒，然后根据秒数转化为多少分钟多少秒的格式，然后显示到屏幕上，就像这样：

int totalmsecond=mediaplayer.getDuration();  
int totalsecond=Math.*round*(totalmsecond/1000);  
String str = String.*format*("%02d:%02d", totalsecond / 60, totalsecond % 60);

totaltime.setText(str);

然后，我们还需要设置进度条的最大值，我们这里用音乐时间的毫秒数作为最大值：

seekbar.setMax(totalmsecond);

接下来，我们想让进度条随着音乐的播放自动移动，而且进度条下面会显示当前播放的时间，就像所有的音乐播放器做的那样，所以，这里我们需要新建一个线程来监听当前播放的位置，并将当前播放的位置转化为当前时间并显示在进度条上，就像这样：

handler=new Handler();  
runnable=new Runnable() {  
 @Override  
 public void run() {  
  
 int currentmsecond=mediaplayer.getCurrentPosition();  
 int currendsecond=Math.*round*(currentmsecond/1000);  
  
 String str=String.*format*("当前时间 %02d:%02d",currendsecond/60,currendsecond%60);  
 currenttime.setText(str);  
  
 seekbar.setProgress(currentmsecond);  
  
 handler.postDelayed(runnable,1000);  
 }  
};

首先，我们可以直接调用MediaPlayer的getCurrentPosition方法得到当前播放位置的毫秒数，然后将毫秒数转化为秒数，然后将秒数变为多少分多少秒的格式，显示在当前时间上，并给进度条设置进度为当前毫秒数，然后过1秒之后继续调用runnable,实现持续监控。

这样用于监控的新线程就写好了，现在当音乐播放的时候就可以用这个线程监控音乐播放，更新当前时间和进度条进度了。我们还希望拖动进度条可以使音乐转到相应位置，我们可以给进度条添加进度改变监听器，当进度发生改变时，音乐就会跳转到改变后的进度，就像这样：

seekbar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
 mediaplayer.seekTo(seekBar.getProgress());  
 }  
  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
  
 }  
  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {  
   
 }  
});

但我很快就发现了一个问题，这样写的时候音乐会出现卡顿，我很快就明白了，我们设置的监听器是当进度条发生改变的时候进行音乐的跳转，而即使我们不拖动进度条，进度条也会自动改变，这是就会触发监听器，从而造成卡顿，解决这个问题也比较简单，我们之前用的是OnProgressChanged方法，这个方法只要当进度条改变就会调用，造成了卡顿，所以我们不用这个方法，我们可以看到，监听器中还有两个方法，onStartTracking和onStopTrackingTouch,这两个方法分别在开始拖动进度条和结束拖动进度条时调用，我们这里使用onStopTracki ngChanged方法，当结束拖动进度条时，调用seekTo方法将音乐转到当前进度条的位置，就像这样：

seekbar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
  
 }  
  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
  
 }  
  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {  
 mediaplayer.seekTo(seekBar.getProgress());  
  
 }  
});

采用这种方法，只有当人为的拖动进度条的时候才会触发监听器，当音乐自动播放的时候不会触发监听器，所以播放的时候不会受影响；当人为拖动进度条的时候就会跳转到相应位置，这样进度条的功能就实现了。

这样，一个音乐播放器就做完了。