

# 移动平台设计第二次作业报告

## 报告包含三个部分

- 1, 分析问题以及应该用的知识
- 2, 文件的源代码, 以及解释(以注释展现)
- 3, 程序演示截图以及解释

### 1, 分析问题以及应该用的知识

此音乐播放器大致有三个问题:

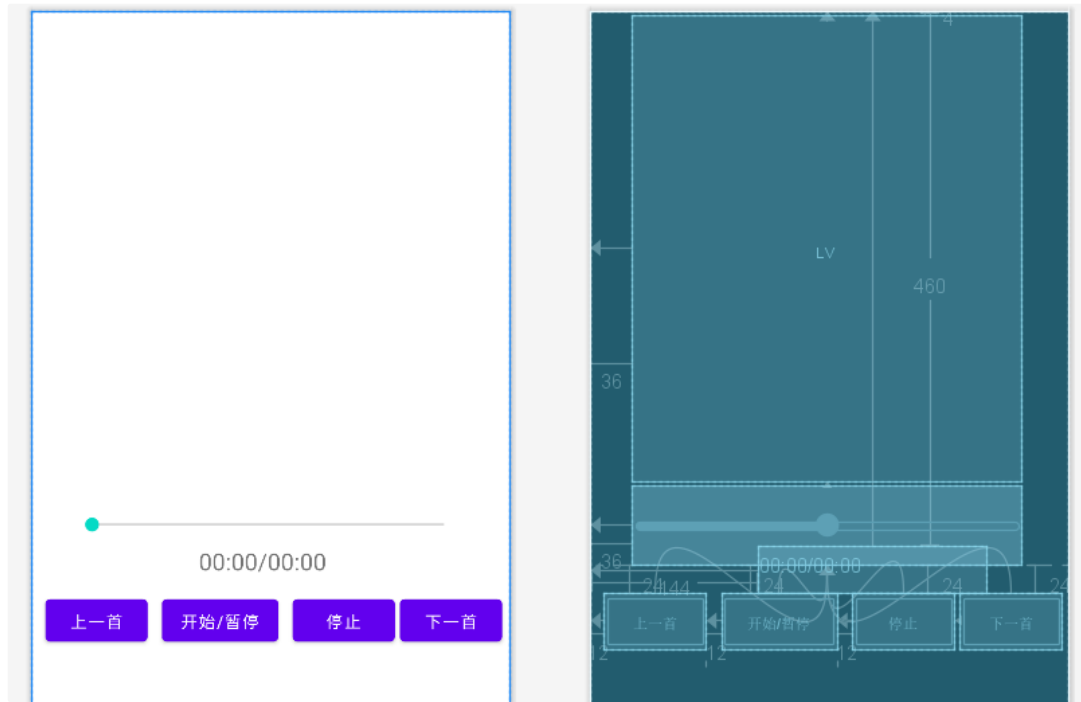
- 1: 能从assets中获得音乐。用ListView, 把所有音乐名展示在界面中。
- 2: 当点击ListView中的音乐名, 可以激活按钮, 然后根据按钮的描述进行播放/暂停 上一首 下一首 停止的操作
- 3: 用seekbar和seekbarHint, 音乐播放的进度, 实时显示在进度条上, 并且通过seekbarHint显示出来, 可以拖动seekbar, 根据seekbarHint, 拖动到想要播放的位置进行播放。

大致用到的知识点(详细以代码注释形式在源代码中给出)

- 1: 通过ListView和ArrayAdapter, 把音乐名先给一个列表, 再通过列表创建adapter, 再把adapter给ListView, 再进行setChoiceMode即可显示。
- 2: 通过ExoPlayer, 通过ListView的监听事件, 获得音乐所在的文件地址, 然后建立Exo播放器, 通过uri为asset:///music/+音乐, 加载媒体资源。然后赋予各个按钮监听事件, 达到各自的作用。
- 3:  
  
seekbar: 用TimerTask创建一个定时任务类, 获得音乐进度, 并赋予seekbar, 在播放, 暂停等会改变音乐播放进度的地方, 设置Timer启用TimerTask, 实现seekbar与音乐同步变化。并且用seekbar的监听事件, 记录拖动的进度, 转换player进度。  
  
seekbarHint: 把音乐播放当前位置转换为分秒格式, 并赋值给seekbarHint来显示, 并把它放在TimerTask里和seekbar一起工作。

### 2, 文件的源代码, 以及解释(以注释展现)

布局截图:



## layout

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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"
    tools:context=".SecondHomeworkActivity">

    <ListView
        android:id="@+id/LV"
        android:layout_width="335dp"
        android:layout_height="400dp"
        android:layout_marginStart="36dp"
        android:layout_marginTop="4dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <SeekBar
        android:id="@+id/SB"
        android:layout_width="335dp"
        android:layout_height="68dp"
        android:layout_marginStart="36dp"
        android:layout_marginTop="4dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/LV" />

    <Button
        android:id="@+id/prev"
        android:layout_width="44dp"
        android:layout_height="44dp"
        android:layout_marginStart="24dp"
        android:layout_marginTop="24dp"
        android:text="上一首"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/pause"
        android:layout_width="44dp"
        android:layout_height="44dp"
        android:layout_marginStart="24dp"
        android:layout_marginTop="24dp"
        android:text="开始/暂停"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/stop"
        android:layout_width="44dp"
        android:layout_height="44dp"
        android:layout_marginStart="24dp"
        android:layout_marginTop="24dp"
        android:text="停止"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/next"
        android:layout_width="44dp"
        android:layout_height="44dp"
        android:layout_marginStart="24dp"
        android:layout_marginTop="24dp"
        android:text="下一首"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/time"
        android:layout_width="100dp"
        android:layout_height="24dp"
        android:layout_marginStart="24dp"
        android:layout_marginTop="24dp"
        android:text="00:00/00:00"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<Button
    android:id="@+id/Bbegin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="12dp"
    android:layout_marginTop="24dp"
    android:text="开始/暂停"
    app:layout_constraintStart_toEndOf="@+id/Bbefore"
    app:layout_constraintTop_toBottomOf="@+id/SB" />
```

```
<Button
    android:id="@+id/Bstop"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="12dp"
    android:layout_marginTop="24dp"
    android:text="停止"
    app:layout_constraintStart_toEndOf="@+id/Bbegin"
    app:layout_constraintTop_toBottomOf="@+id/SB" />
```

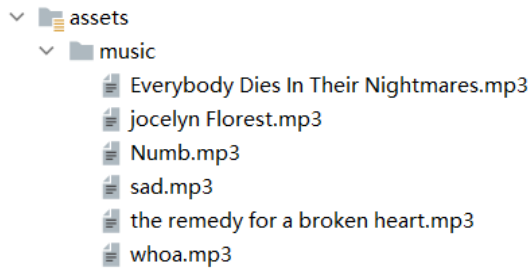
```
<Button
    android:id="@+id/Bbefore"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="12dp"
    android:layout_marginTop="24dp"
    android:text="上一首"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/SB" />
```

```
<Button
    android:id="@+id/Bnext"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="4dp"
    android:layout_marginTop="24dp"
    android:text="下一首"
    app:layout_constraintStart_toEndOf="@+id/Bstop"
    app:layout_constraintTop_toBottomOf="@+id/SB" />
```

```
<TextView
    android:id="@+id/SBHint"
    android:layout_width="196dp"
    android:layout_height="40dp"
    android:layout_marginStart="144dp"
    android:layout_marginTop="460dp"
    android:text="00:00/00:00"
    android:textSize="20sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

## assets中的音乐文件:



## activity

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.SeekBar;
import android.widget.TextView;
import com.google.android.exoplayer2.ExoPlayer;
import com.google.android.exoplayer2.MediaItem;
import com.google.android.exoplayer2.Player;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Timer;
import java.util.TimerTask;

public class SecondHomeworkActivity extends AppCompatActivity {
    // listview的组件
    ListView listView;
    ArrayAdapter<String> adapter;
    List<String> music_list = new ArrayList<>(); //给adapter用的list

    // 下面是exoplayer的组件
    SeekBar SB; //进度条
    TextView SBHint; //音乐时长提示
    Button Bbegin; //播放按钮
    Button Bstop;
    Button Bbefore;
```

```

Button Bnext;
ExoPlayer player; //播放器
private Timer timer; //定时器
private boolean prepared; //播放器是否准备好

String mName;

ListView lv;

////////////////////////////////////
////////////////////////////////////
//                                listview的创建
public void initView() {
    //ListView初始化
    listView = findViewById(R.id.LV);
    music_list = getMusic(); //获取assets/pic目录下所有音乐名
    adapter = new ArrayAdapter<String>(
        SecondHomeworkActivity.this, android.R.layout.simple_list_item_1,
        music_list);
    listView.setAdapter(adapter);
    listView.setChoiceMode( ListView.CHOICE_MODE_SINGLE );
    listView.setOnItemClickListener(listener);
    //为按钮设置监听
    Bbegin=(Button)findViewById(R.id.Bbegin);
    Bstop=(Button)findViewById(R.id.Bstop);
    Bbefore=(Button)findViewById(R.id.Bbefore);
    Bnext=(Button)findViewById(R.id.Bnext);
    Bstop.setEnabled(false);
    Bbefore.setEnabled(false);
    Bnext.setEnabled(false);
    Bbegin.setEnabled(false);
    //为进度条设置监听
    SB = (SeekBar) findViewById(R.id.SB);
    SBHint = (TextView) findViewById(R.id.SBHint);
}

public List<String> getMusic() {    //给music_list一个放有所有音乐文件名的list, 然后给adapter
    List<String> mList = new ArrayList<>();
    try {
        String[] fNames = getAssets().list("music"); //获取assets/music目录下所有文件名
        for (String fn : fNames) {
            mList.add(fn);
        }
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
    return mList;
}

```

```

// ListView监听器
AdapterView.OnItemClickListener listener = new
AdapterView.OnItemClickListener() {
    //设置给list的监控, 当点击音乐时, 把音乐名提取, 然后新建播放器, 并进入READY状态
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        lv = (ListView) view.getParent();
        lv.setSelector(R.color.purple_200); //设置高亮背景色
        mName = parent.getItemAtPosition(position).toString(); //获得选中项音乐
名称
        initExoPlayer(); //建播放器
        Bstop.setEnabled(true);
        Bbefore.setEnabled(true);
        Bnext.setEnabled(true);
        Bbegin.setEnabled(true);
    }
};

////////////////////////////////////
////////////////////////////////////
//                                     下面为exoplayer

public void initExoPlayer(){
    //建播放器, 并设置成STATE_READY (用prepare())
    player = new ExoPlayer.Builder(SecondHomeworkActivity.this).build();
    MediaItem mediaItem = MediaItem.fromUri( "asset:///music/"+mName);
    player.setMediaItem(mediaItem); //加载媒体资源
    player.setRepeatMode(ExoPlayer.REPEAT_MODE_ONE); //单曲循环
    player.prepare(); //会触发下面的监听器 state会变成STATE_READY
}

Player.Listener listener0 = new Player.Listener() { //播放器状态监听器
    @Override
    public void onPlaybackStateChanged(int playbackState) {
        if (playbackState == ExoPlayer.STATE_READY) { //播放器准备好了
            prepared = true;
            long realDurationMillis = player.getDuration(); //得到音乐的总时长
            SB.setMax((int) realDurationMillis); //设置seekbar的最大值
        }
    }
};

View.OnClickListener listener1 = new View.OnClickListener() { //开始, 暂停按钮
    @Override
    public void onClick(View v) {
        if ( player.isPlaying() ) {
            player.pause();
            timer.cancel(); //停止定时器
        } else {

```

```

        player.play();
        timer = new Timer();
        timer.schedule(new ProgressUpdate(), 300, 500);
    }
}

};

View.OnClickListener listener2= new View.OnClickListener() {    //停止按钮
    @Override
    public void onClick(View v) {
        player.stop();
        player.release();    //停止并释放播放资源
        SB.setProgress(0);    //重置进度条显示
        SBHint.setText("00.00/00.00");
    }
};

View.OnClickListener listener3= new View.OnClickListener() {    //上一首按钮
    @Override
    public void onClick(View v) {
        int index = music_list.indexOf(mName);    //获取当前播放歌曲的索引
        listView.setItemChecked(index - 1, true);    //取消之前选中的 ListView 高亮
        状态, 然后让新的选中项高亮
        int position = music_list.indexOf(mName);
        player.pause();
        player.release();
        timer.cancel();
        if (position > 0) {
            // 暂停当前音乐
            mName = music_list.get(position - 1);
        }
        else {
            mName=music_list.get(music_list.size()-1);
        }

        initExoPlayer();
        player.play();
        timer = new Timer();
        timer.schedule(new ProgressUpdate(), 300, 500);
    }
};

View.OnClickListener listener4= new View.OnClickListener() {    //下一首按钮
    @Override
    public void onClick(View v) {
        int position = music_list.indexOf(mName);
        player.pause();
        timer.cancel();

        if (position < music_list.size() - 1) {

```

```

        mName = music_list.get(position + 1);
    }
    else {
        mName=music_list.get(0);
    }

    initExoPlayer();
    player.play();
    timer = new Timer();
    timer.schedule(new ProgressUpdate(), 300, 500);

}

};

```

```

SeekBar.OnSeekBarChangeListener listenerSB =
    new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int progress,
                                       boolean fromUser) {

        }
        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {
            timer.cancel();
            SBHint.setText("正在快进");
        }
        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {
            player.seekTo(SB.getProgress());
            timer=new Timer();
            timer.schedule(new ProgressUpdate(),300,500);
        }
    };

```

`private class ProgressUpdate extends TimerTask` {*//ProgressUpdate定时任务类：定时刷新SeekBar进度条*

```

    @Override
    public void run() {
        runOnUiThread( new Runnable() {
            @Override
            public void run() {
                if(player.isPlaying()){//正在播放时显示 播放位置和seekbar进度
                    long position = player.getContentPosition();//获取媒体播放的当前位置
                    (毫秒)

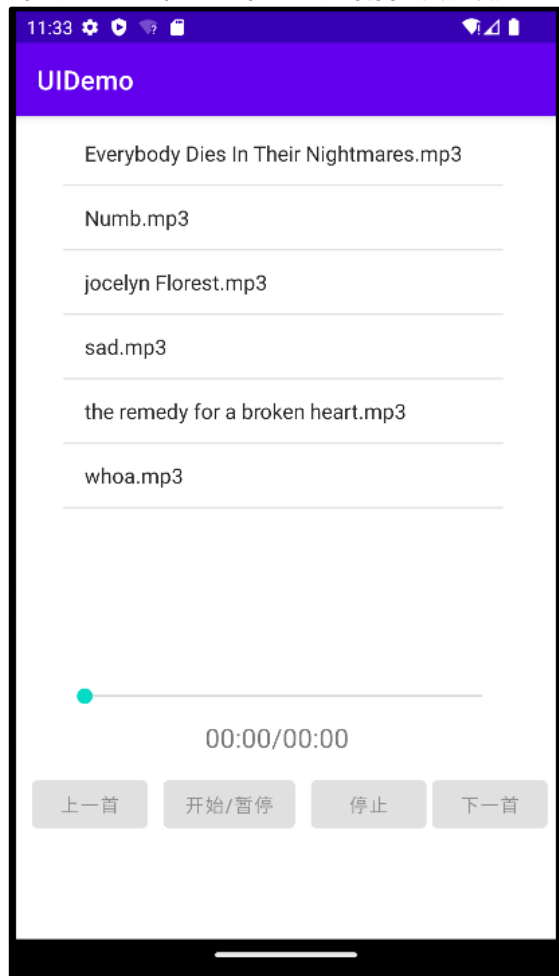
                    SB.setMax((int)player.getDuration());//seekBar设置最大进度
                    SB.setProgress((int) position);//SeekBar设置当前进度
                    SBHint.setText(
format(position)+'/'+format(player.getDuration()));//显示当前音乐时间
                }}
            });
        }
    }

```

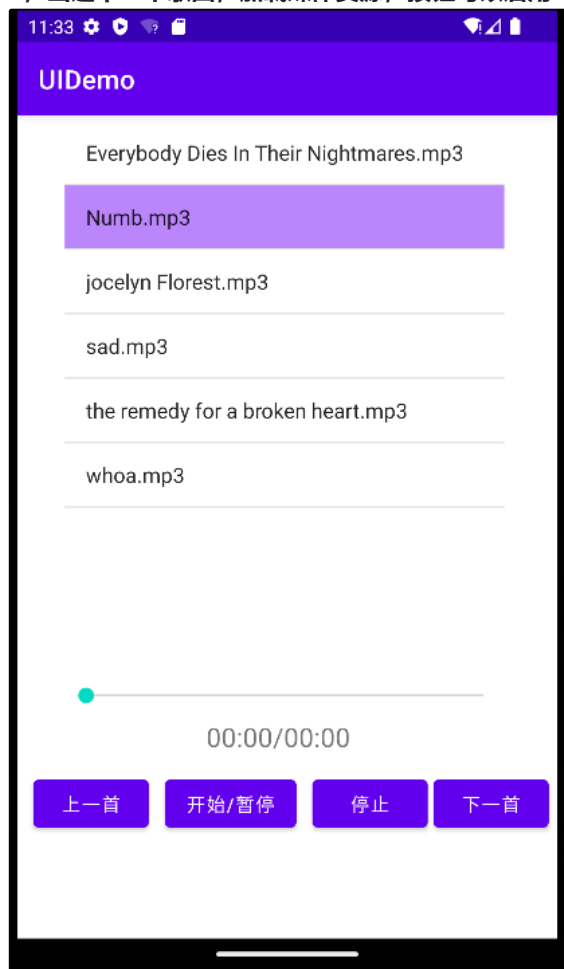




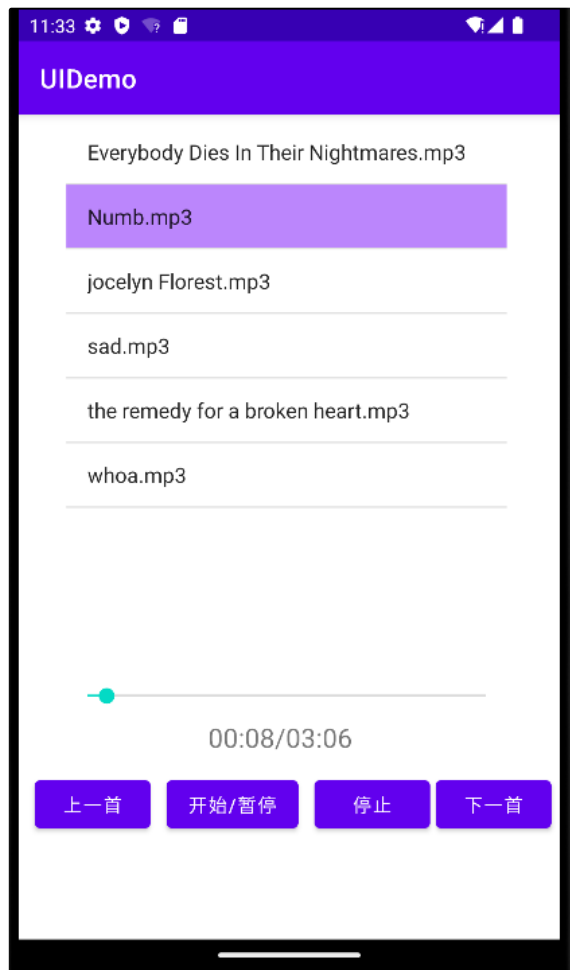
1, 刚进入APP, Exo播放器未加载音乐资源按钮无法使用



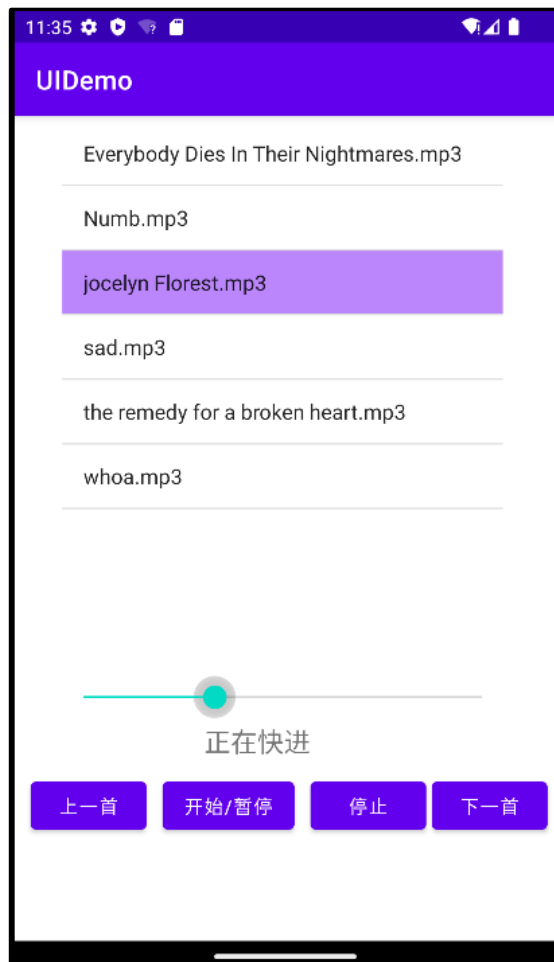
2, 当选中一个歌曲, 加载媒体资源, 按钮可以启用



3, 当点击开始时, 音乐开始播放, seekbar开始走动, seekbarHint显示最大时长和当前时长



4, 当拖动进度条, 显示正在快进, 松开后, 音乐跳转至拖动位置播放



5, 点击停止时, 释放当前音乐资源, 需要重新点击歌曲名, 加载资源。(无法通过图片演示)

6, 点击上一首和下一首, 会先暂停并释放当前资源, 然后上一首/下一首资源, 并开始播放。(无法通过图片演示)