

基于 Android 系统的音乐播放器设计与实现*

葛 艳,高占江

(青岛科技大学 信息科学技术学院,山东 青岛 266061)

摘 要: 基于 Android 平台开发音乐播放器。该播放器主要实现了 SD 卡扫描、后台播放、歌手与专辑筛选,歌曲列表管理、歌词同步滚动显示、播放模式选择、皮肤更换、网络下载、桌面 Widget 等功能。对 Android 应用程序的开发环境及工具作了简单介绍,详细介绍了音乐播放器软件界面布局方式、自动音乐扫描机制、歌词同步实现算法以及歌词的搜索与下载等功能模块的设计与实现。对歌词同步滚动显示进行了透彻分析。该音乐播放器通过了 Android 智能手机运行测试,具有较好的集成度和良好的稳定性。

关键词: Android;Java;音乐播放;歌词同步

中图分类号: TP311

文献标识码: A

文章编号: 1674-7720(2014)01-0005-03

Design and implementation of Android-based music player

Ge Yan, Gao Zhanjiang

(School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China)

Abstract: A music player software is developed based on Android platform. The player has mainly realized scanning SD card, background player, singer and the album screening, music list management, synchronized lyrics scrolling display, the playback mode selection, skin replacement, download from network, and desktop Widget functions. Firstly, the Android application development environment and tools are introduced briefly. Secondly, introduces in detail the interface layout, automatic scanning mechanism of music, lyrics and the lyrics synchronization algorithm for the search and download. The lyrics synchronized display scrolling undertook thorough analysis. The music player has been tested in Android intelligent mobile phone. It has good integration and stability.

Key words: Android; Java; music player; lyrics display

随着科技发展的日新月异,人们对移动设备的需求越来越高,手机已不只是通信工具,而是一个多媒体平台。Android 是 Google 公司开发的基于 Linux 平台的开源的移动终端智能操作系统^[1]。Android 系统由操作系统、用户界面和应用程序组成,允许开发人员自由获取和修改源代码。Android 的发布大大丰富了各种手持式设备软件的功能^[2]。

本文基于 Android 平台开发音乐播放器,选择开发个性的播放软件,摒弃单方面追求花哨而带来的系统资源浪费,将各种性能优化,继承播放器的常用功能,满足大多数用户的娱乐需求。该播放器实现 SD 卡扫描、后

台播放、歌手与专辑筛选,歌曲列表管理、歌词同步滚动显示、播放模式选择、皮肤更换、网络下载等功能。此外,还实现桌面的 Widget 功能,使用户在不打开该软件的同时,就可以一键听歌,极大地优化了用户体验。

1 Android 简介

Android 系统分为 Applications、Application Framework、Libraries、Android Runtime、Linux Kernel 5 大层^[3]。本播放软件属于应用软件,只对 Applications 应用层程序的探讨,对具体压缩算法不作深究。

1.1 Android 基本组件

Android 应用程序的组件主要有 4 个,针对智能手机的诸多突发情形,都做出了相应的处理操作^[4]。

(1) Activity: 是应用程序最基本的组件。应用程序的每个页面都由各种 Activity 构成。它是一种可视化的、直

* 基金项目:国家自然科学基金资助项目(61273180);山东省自然科学基金资助项目(ZR2009GQ013, ZR2011FQ005, ZR2011FL010);青岛科技大学科研启动基金项目(0022147);山东省优秀中青年骨干教师国际合作培养项目

接与用户接触的界面元素。

(2)Service:是一种服务组件,运行于程序的后台。该组件对用户是不可见的,在后台提供程序的托管运行。

(3)ContentProvider:是一种内容提供者组件。该组件能够实现应用程序之间的数据共享,并能够监听其共享数据的变化。

(4)BroadcastReceiver:实现应用程序内部数据的传递,也能实现事件的先后顺序触发。

1.2 开发工具

软件开发使用 Eclipse 软件,使用 Android SDK、ADT 的支持,JDK 开发环境,使用 Java 语言作为开发语言,基于 C/S 开发模式。使用 Emulator 调试工具,调试工具提供了断点调试,文件管理,电话短信模拟,在软件开发过程中提供了极大的方便。

2 软件核心功能

该部分详细介绍了播放界面的布局方式、音乐列表自动扫描原理、播放时歌词同步滚动实现机制、歌词搜索与下载机制。

2.1 主页面布局

软件的主播放界面采用线性布局与层叠布局的结合,布局中使用了 Android 的系统控件和自定义的控件,丰富了页面元素,并对每个控件进行了布局设置,下面对应播放主界面的布局:

```
//线性布局方式
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:lrview="http://schemas.android.com/apk/res/com.gao.mymediaplayer01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <FrameLayout //层叠布局方式
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <ImageView //图片控件
            android:id="@+id/background"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:layout_gravity="fill"/>
    //下面是自定义歌词控件的布局控制
    <com.gao.mymediaplayer01.LrcView
        //自定义歌词控件类
        android:id="@+id/lrcTextView" //控件的唯一标识(Id)
        android:layout_width="fill_parent"
        //设置控件的宽度填充父控件
        android:layout_height="160dip"
        //设置控件的高度为特定值
        android:gravity="center"
        //设置控件内容的对齐方式为居中
```

```
android:layout_gravity="center_horizontal"
    //控件的对齐方式为水平居中
android:layout_marginTop="70dip"
    //控件垂直方向上距离顶部的距离
/>.....其他控件.....
```

上面是页面布局的部分代码,最后一个控件 com.gao.mymediaplayer01.LrcView 使用的是自定义的控件,目的是显示歌词信息并能够根据歌曲当前播放时间匹配歌词的当前行索引,实现歌词的实时动态刷新显示。实现的效果如图 1 所示。

2.2 音乐扫描

Android 系统提供了一种类似关系表的结构来把应用程序的数据暴露给外界,并把每个这种表使用唯一的标识符 URI 来标识^[2]。Android 系统对外部存储设备的媒体文件进行了统一管理,把每个音乐文件的 ID、时长、艺术家等相关信息全部存放在这个表中,使用 Contentprovider 来访问这个唯一的标识符 URI 便可以查询到在用户的 SD 卡中的所有音乐文件,实现代码如下:

```
musicCursor=this.getContentResolver().query(MediaStore.Audio.Media.EXTERNAL_CONTENT_URI,
    new String[]{MediaStore.Audio.Media.TITLE, //歌曲标题
        MediaStore.Audio.Media.DURATION, //歌曲时长
        MediaStore.Audio.Media.ARTIST, //歌曲的作者
        MediaStore.Audio.Media._ID, //歌曲在 SD 卡上的唯一标识
        MediaStore.Audio.Media.DISPLAY_NAME, //歌曲显示的名字
        MediaStore.Audio.Media.DATA //歌曲文件的路径
    }, null, null, null);
```

根据查询条件可以得到所有歌曲的游标指针,这里的查询条件可以设置指定的艺术家和指定的专辑,从而可以查询指定的艺术家和专辑的特定歌曲列表。将得到的游标数据传递到 ListAdapter 适配器中,其中可以设置列表项显示的内容,之后为这个 ListView 控件设置单击事件。

2.3 歌词同步滚动实现机制

音乐播放时实现歌词同步滚动显示是本音乐播放器的一个特色。下面详细介绍实现歌词同步滚动的具体流程:

(1)歌词 LRC 文件一般存放在与该歌曲相同的位置,通过歌曲在 SD 卡上的 DATA 属性获得 LRC 文件的位置,使用输入缓冲流 BufferedReader 进行读取,关键是每一次读取歌词文件的一行,因为歌词文件的每一行是一个或者多个时刻和歌词内容的连接,这里把每行歌词抽象为一个对象,整个歌词文件看成是所有对象的 List 集



图 1 播放主界面

合, 每个对象是由时间属性和歌词内容属性共同组成, 在读取每一行歌词并将其转换成歌词类的一个实例时, 对不同的表示形式作了不同的处理:

[00:25.93]和[00:25]两种时间形式的处理。对其统一格式后判断时间段的长度, 采用不同的函数处理。在实际使用时将其转换为毫秒保存到对象的时间的私有变量中, 内容保存到 LRC 内容的成员变量中。

对于在一行歌词中多个时间段表示同一种歌词内容的情形, 首先对整行歌词中的字符“]”统一替换为某个特殊的字符, 将整行内容根据这个特殊字符进行分割得到 string 类型的数组: String []splitLrc_data=str.split("@"); 可以得到数组长度减一个数量的歌词对象, 每个歌词对象时间域为分割得到的数组的内容, 内容域都是数组最后一个元素的值, 最后将所有这些歌词对象存放到 List<LrcContent>当中。

(2)对 List 歌词对象按照其时间变量进行排序, 排序采用冒泡排序算法, 其核心代码如下:

```
for(int i=1;i<=count-1;i++)
for(int j=0;j<count-i;j++)
{
    //排序依据是歌词对象的时间变量
    if (LrcList.get (j).getLrc_time ()>LrcList.get (j+1).
getLrc_time())
    {
        LrcContent tempLrcContent=LrcList.set (j,LrcList.get (j+
1));
        LrcList.set (j+1,tempLrcContent);
    }
}
```

(3)利用冒泡排序算法最终得到按时间先后排好序的歌词对象的 List, 根据歌曲当前播放时间选择当前需要显示的歌词行的索引, 具体操作如下。

当前歌曲时间小于第一个歌词对象的时间时, 设定要显示歌词的行的索引为 1;

当前歌曲时间大于第一个歌词对象的时间时, 要循环判断出当前歌词时间大于第 N 个歌词对象的时间并且要小于第 N+1 个歌词对象的时间, 设定要显示的歌词的行的索引为 N;

当前歌曲时间大于最后一个歌词行的时间时, 设定要显示的歌词行的索引为歌词文件的数量。

(4)最后, 得到当前显示歌词的索引后, 使用自定义文本控件高亮显示当前歌词行, 其余歌词行非高亮显示。在后台 Service 中设定刷新频率为 50 ms, 每 50 ms 获得当前歌曲时间进度, 更新当前行的索引, 获得一个时刻的当前行的索引后, 从 Service 中利用广播机制将其发送到前台 Activity 中, 在 Activity 接收到当前索引后, 将自定义的歌词控件重新绘出。

设置 lrcView.setIndex (lrcIndex) 后, 使用 lrcView.invalidate () 强制使歌词控件重新绘画, 此时绘出的高亮行为当前索引行, 且显示在屏幕的中央, 其余行显示为

非高亮行, 绘画歌词的算法核心代码如下:

```
canvas.drawText (lrcList.get (Index).getLrc_body (), width/
2, high/2,
CurrentPaint);
float tempY=high/2; //屏幕垂直方向中央的高度
//画出本句之前的句子
for(int i=Index-1;i>=0;i--){
tempY=tempY-TextHigh;
canvas.drawText (lrcList.get (i).getLrc_body (), width/2,
tempY, NotCurrentPaint);}
tempY=high/2;
//画出本句之后的句子
for(int i=Index+1;i<=lrcList.size()-1;i++){
tempY=tempY+TextHigh;
canvas.drawText (lrcList.get (i).getLrc_body (), width/2,
tempY, NotCurrentPaint);}
```

按照上述步骤, 在后台 Service 中设定刷新频率 (一般为 50~200 ms) 可以实现动态的显示歌词。歌词滚动效果如图 1 所示。

2.4 歌词的搜索与下载

要下载一首歌的歌词信息, 应该由这首歌曲的歌手和歌曲名共同决定, 所以利用当前播放歌曲的歌手和歌曲名称作为参数进行歌词的搜索, 这里使用百度音乐盒提供的歌词服务器来进行下载。下载流程如下:

(1)首先将歌曲的歌手和歌曲名称进行 UTF-8 编码的转换如下:

```
titleName=URLEncoder.encode (titleName, "UTF-8");
singerName=URLEncoder.encode (singerName, "UTF-8");
```

(2)其次, 将参数传递到搜索链接中:

```
strUrl="http://box.zhangmen.baidu.com/x? op=12&count=
1&title="+titleName+"$$$"+singerName+"$$$$$";
```

此链接指向的是一个 xml 类型的文件, 该文件包含对该歌曲及歌词等信息的描述, 使用 I/O 流读取该文件, 如果该文件内容不为空, 可以从中获取到该歌词在服务器中的 Id (即 LyricId), 根据这个 Id, 如果这个 Id 不为空, 进而可以得到该歌词文件的 URL 链接地址:

```
lyricURLStr = "http://box.zhangmen.baidu.com/bdlrc/" +
lyricId/100+"/"+lyricId+".lrc";
```

根据该 URL 使用 I/O 流将歌词文件下载到本地就可以完成歌词的下载。

(3)最后, 歌词下载完成后, 获取歌词的保存路径, 调用解析歌词文件的方法进行解析实现歌词的滚动显示。

本文介绍了基于 Android 平台的音乐播放器的设计方案和关键技术。详细介绍了音乐播放器软件界面布局方式、自动音乐扫描机制、歌词同步实现算法以及歌词的搜索与下载等功能模块的设计与实现。对歌词同步滚

动显示进行了透彻分析。该音乐播放器集扫描 SD 卡, 音乐列表显示、播放、后台播放、上一首、下一首、音量调节歌手选择、专辑选择、最近播放、最经常播放、歌词同步滚动显示、快进快退、播放模式选择、更换皮肤、音乐文件操作、网络下载、桌面 Widget 等功能于一体, 功能较完善。通过在 Android 智能手机对音乐播放器进行了功能测试。该音乐播放器性能良好, 运行流畅。

参考文献

- [1] 周时伟, 谢维波. 基于 Android 的智能家居终端设计与实现[J]. 微型机与应用, 2012, 31(14): 10-13.
- [2] 曾建平, 邵艳洁. Android 系统架构及应用程序开发研究[J].

微计算机信息, 2011, 27(9): 1-3.

- [3] 樊新, 高曙. 基于智能移动终端的安全检查系统设计与实现[J]. 微型机与应用, 2012, 31(20): 87-92.
- [4] 刘安战, 贾晓辉. 基于 Android 的私密短信系统设计与实现[J]. 微型机与应用, 2012, 31(17): 51-56.

(收稿日期: 2013-09-13)

作者简介:

葛艳, 女, 1975 年生, 博士, 副教授, 硕士生导师, 主要研究方向: 计算机应用技术, 智能控制理论与应用。

高占江, 男, 1989 年生, 硕士研究生, 主要研究方向: 软件工程。

电子技术应用网
APPLICATION OF ELECTRONIC TECHNIQUE
www.ChinaAET.com