Computer Science and Application 计算机科学与应用, 2015, 5(10), 351-358
Published Online October 2015 in Hans. http://www.hanspub.org/journal/csa
http://dx.doi.org/10.12677/csa.2015.510044

Hans汉斯

Research and Implementation of Intelligent Terminal Music Player Based on Android Platform

Cheng Jiang[1], Zhansheng Chen[1,2*]

[1]School of Applied and Technology, Beijing Union University, Beijing
[2]School of Computer and Information Technology, Beijing Jiaotong University, Beijing
Email: *ldtchenzs@buu.edu.cn

Received: Oct. 5th, 2015; accepted: Oct. 26th, 2015; published: Oct. 30th, 2015

Abstract

Since many music players in the Android market have more functions, fancy appearance and excessive use of resources, we develop a high performance music player. It inherits the common functions, supports MP3, WAV and other common formats, and could satisfy people's basic command. After transplantation test, the music player could run well independently; the interface is simple and easy to use.

Keywords

Android, Music Player, MP3, High Performance

基于Android智能终端音乐播放器的研究与实现

江　城[1]，陈战胜[1,2*]

[1]北京联合大学应用科技学院，北京
[2]北京交通大学计算机与信息技术学院，北京
Email: *ldtchenzs@buu.edu.cn

收稿日期：2015年10月5日；录用日期：2015年10月26日；发布日期：2015年10月30日

*通讯作者。

文章引用：江城，陈战胜．基于 Android 智能终端音乐播放器的研究与实现[J]．计算机科学与应用，2015，5(10): 351-358. http://dx.doi.org/10.12677/csa.2015.510044

摘　要

针对Android市场众多播放器功能多，占用资源过多的现象，开发一款继承播放器常用功能，支持MP3、WAV等常见格式、满足手机空间和上网流量资源匮乏的用户需求的高性能音乐播放器。经移植测试，能够独立运行、界面简洁且便于用户使用。

关键词

Android，音乐播放器，MP3，高性能

1. 引言

Android 平台是目前智能移动终端的主流系统。随着人们生活、工作节奏的加快，乘车、运动、学习等碎片时间的增多，音乐播放器成为人们所关心的必备应用之一，广受大家欢迎。

目前，Android 市场上以酷狗为代表的音乐播放器有着广泛的消费群体，性能品质较优。同时，也存在很多其它不知名的音乐播放器。其中，陈文[1]提出了将音乐播放器和手机传感器相结合，实现当用户翻动手机时来控制音乐播放器的暂停和重播，扩展了传统音乐播放器暂停功能的个性化实现。对于经常手持手机的用户而言的确有效，但是对于将手机置于背包等情况下并未进行测试，并且该播放器未能展示歌词等功能。王秀芳[2]等开发的 YOBO 在线音乐播放器，考虑消费者使用手机在线听歌造成流量过大的情况，并且提供了用户授权、个人信息展示、音乐盒等五大功能，功能相对完善。但是，对于安装即使用的用户而言，用户授权会让用户感觉繁琐，个人信息略显多余，在线听音乐对于流量紧张的用户并不适用。王玉凡[3]设计制作的手机音乐播放器，设计过程描述详细，具有音乐播放器基本功能，但是用户体验感觉不强，不具备歌词展示等功能。谭静[4]等以用户体验和需求出发，在基本功能上增加了了歌词同步显示、播放可选等功能，界面简洁，功能优化，但是并未给出音乐播放器 APP 所占空间大小。针对大多数音乐播放器存在功能多、耗费手机系统资源等弊端，本文在深入学习 Android 多媒体框架结构、框架核心 Open core 和多媒体播放机制，开发一款继承播放器常用功能，支持 MP3、WAV 等常见格式、满足手机空间和上网流量资源匮乏的用户需求的高性能音乐播放器。经移植测试，能够独立运行、界面简洁且便于用户使用。

2. Android 结构组成

Android 操作系统由 Activity、Content Provider、Broadcast Intent receiver 和 Service 四部分组成。通常，一个 Android 应用程序是需要四个模块还是四个模块中的部分模块取决于软件业务的需求。其中，在模块之间进行跳转需要使用 Intent 类的方法 StartIntent()。此外，在应用程序开发过程中布局的作用至关重要，在 Android 中主要有五种布局，分别是 FrameLayout、Linearlayout、AbsoluteLayout、TableLayout 和 RelativeLayout，布局之间可以相互嵌套。

3. 国内外播放器研究现状

目前，Linux 上流行的音视频播放器主要有：QuickTime、Xmms、RealPlayer、Mplayer 和 Xmovie 等，分别介绍如下。

3.1. QuickTime

QuickTime 是苹果公司出品的流式音视频解决方案，是早期流媒体文件格式。QuickTime 文件扩展名为

“.mov”，它由 QuickTime Move 电影、QuickTime 媒体抽象层和 QuickTime 内置媒体服务系统三部分组成。

3.2. Xmms

Xmms 是 X-Window 系统下最流行的多媒体播放器之一，支持 MP3、MOD、WAV 等文件格式。若有插件支持，还可以支持更多的多媒体类型。

3.3. RealPlayer

RealPlayer 在低带宽下支持音频、视频的低损失传输，具有优秀流媒体在线播放能力，其后续版本增加了对微软音视频和便携式设备的支持。

3.4. MPlayer

MPlayer 是 Linux 上的电影播放器，能够播放 MPEG、ASF、WMV、VOB、AVI 和 OGG 等文件，还可以播放 VCD、SVCD、DVD 和 DivX 等格式的电影，功能强大。

3.5. Xmovie

Xmovie 是最早作为电视节目的播放软件。播放界面简洁、支持 MPG、MOV、VOB 和 MP3 等格式文件。

4. 系统总体分析

4.1. 系统需求分析

智能手机大行其道之际，音乐播放功能已经成为各大手机厂商的标配。从用户使用播放器的业务需求出发，经调研分析，本文决定走“界面简洁、功能满足、操作简便”的路线，设计思路如下：

(1) 打开播放器后，可以根据不同分类进行播放列表；
(2) 选定播放文件进行播放；
(3) 播放音乐文件；
(4) 播放的暂停、停止操作；
(5) 音乐的前一首/下一首与当前音乐的切换操作；
(6) 音乐播放的进度表功能；
(7) 音乐播放方式的选择功能；
(8) 音乐文件和名称的显示功能；

4.2. 系统功能分析

经过分析，音乐播放器的功能模块如图 1 所示。

如图 1 所示，手机音乐播放器共包含播放暂停、文件选择、切换歌曲和进度调节四大功能。其中，播放暂停功能可以对手机正在播放的的音乐进行播放和暂停播放。文件选择功能是选择手机里的音乐文件，用列表的方式展示所有的文件，并通过上下滑动可以看到所有文件，选择需要播放的文件。切换歌曲功能是上一首、下一首歌曲按钮功能的实现。调节进度功能是可以通过拖动进度条进行调节音乐播放的进度。

5. 系统详细设计

5.1. 系统开发环境

本文使用 Eclipse 集成开发环境，采用 Java 语言，设计并实现了基于 Android 平台的手机音乐播放器。

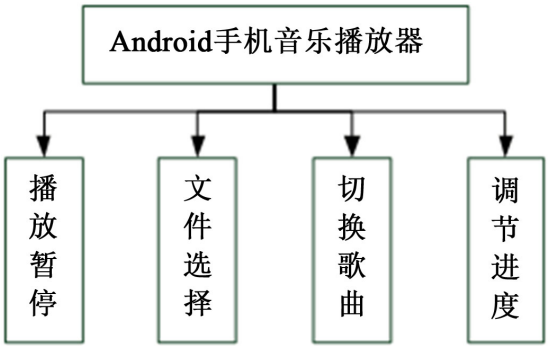

Figure 1. The function chart of mobile music player
图 1. 手机音乐播放器功能图

Android 程序的编译运行有两种方法，一种是利用模拟器 AVD 在电脑上模拟运行；另一种是直接利用 Android 设备进行运行。本文利用 AVD 在电脑上模拟 Android 环境编写代码、调试运行，让后移植到两款以上 Android 设备进行验证检测效果。

5.2. 系统开发界面实现

本文首先构思手机音乐播放器的布局显示界面，决定采用 RelativeLayout 布局中的列表方式实现音乐文件的呈现，具体如图 2 所示。播放器播放界面如图 3 所示。

5.3. 系统开发工作流程

本文在音乐播放器主界面实现的基础上，遵循如图 4 所示的系统开发工作流程，便于提升开发效率。

5.4. 关键操作代码实现

本小节主要介绍播放界面功能的实现代码，主要包括音乐播放的暂停、播放、上/下一首和进度调节四部分，分别介绍如下。

(1) 音乐播放的实现代码

```
void start() {
        if (state == STOP) {
            play();
         } else if (state == PAUSE) {
            mediaPlayer.start();
            state = PLAYING;
         }
play.setImageResource(R.drawable.pause);
    }
```

其中，play()函数需要显示音乐专辑的图片、音乐标题、作者以及时间等相关信息，是音乐播放器播放功能的核心，具体代码如图 5 所示。

(2) 音乐暂停的实现代码

```
    private void pause() {
     if (mediaPlayer.isPlaying()) {
          mediaPlayer.pause();
```

```
                state = PAUSE;
            }
        play.setImageResource(R.drawable.play);
            }
```

Figure 2. The main interface of mobile music player
图 2. 手机音乐播放器主界面

Figure 3. The play interface of mobile music player
图 3. 手机音乐播放器播放界面

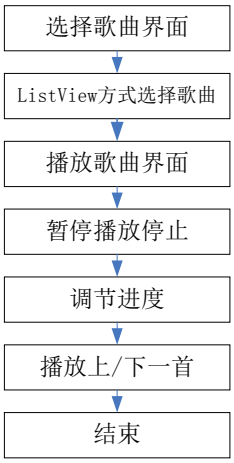

Figure 4. System development process
图 4. 系统开发流程

```
private void play() {
    ContentResolver    cr = getContentResolver();
    long album_id = cursor.getLong(cursor.getColumnIndex(MediaALBUM_ID));
    Uri sArtwork    Uri = UriparseUri("content://media/external/audio/albumart");
    Uri    uri= ContentUriswithAppendedId(sArtworkUri, album_id);
    InputStream in = null;
    ImageView album_image = (ImageView) findViewById(R.id.album_iamge);
    try {    in = cr.openInputStream(uri);
             Bitmap bitmap = BitmapFactorydecodeStream(in);
             album_image.setImageBitmap(bitmap);
       } catch (FileNotFoundException e) {
    album_image.setImageResource(R.drawable.disc1);}
    isRunning = false;
    String path = cursor.getString(cursor.getColumnIndexOrThrow(MediaDATA));
    try {    if (mediaPlayer == null ||state == STOP) {
                mediaPlayer = new MediaPlayer();
                mediaPlayer.setOnPreparedListener(new OnPreparedListener() {
                public void onPrepared(MediaPlayer mp) {
                   mediaPlayer.start();
                   state = PLAYING;
                   isRunning = true;
                   mUIThread = new UIThread();
                   mUIThread.start();
              }   } );    // end of PreparedListener
             mediaPlayer.setOnCompletionListener(new OnCompletionListener() {
                public void onCompletion(MediaPlayer mp) {
                   isRunning = false;
                   next();    }   } );
             } else {    mediaPlayer.reset();    }
             mediaPlayer.setDataSource(path);
             mediaPlayer.prepare();
       } catch (Exception e) {   e.printStackTrace();   }
}
```

Figure 5. Play() function
图 5. Play()函数

6. 系统测试与评价

本文采用 AVD 在电脑编程实现后，移植到小米 2 和华为荣耀 6 plus 分别进行测试，界面美观，使用便捷且功能满足需求，占用系统资源少，运行速度极快。图 6(a)~图 6(c)展示了小米 2 真机运行的效果。

(a) (b) (c)

(d) (e)

Figure 6. Music player portable test effect
图 6. 音乐播放器移植测试效果

其中，图 6(a)显示了在真机上成功安装 APP 后的显示图标。图 6(b)为程序启动后，显示手机中所有可用的音乐文件，采用列表的形式进行显示。图 6(c)为手机播放器的播放界面，界面简洁大方且便于用户操作，包含的信息主要有：上一首，暂停播放，停止，下一首四个按钮，分别实现音乐播放器的播放音乐功能。同时，进度条能够显示音乐播放的进度，方便用户通过拖动进度条调节音乐播放的进度，增强用户体验。

图 6(d)展示了本软件占用机身内存仅有 168 KB，相较之的播放器软件动则八九十 M 大大减少了内存空间。图 6(e)所示为本软件运行时所占资源和虾米音乐所占资源进行比较，不足十分之一，极大的体现了本软件在运行速度和优化资源的突破和创新。

7. 结束语

本文基于 Android 手机平台音乐播放器软件的开发，实现了播放器的基本功能，采用 Java 语言，SQLite 数据库支持组合 SharePreference 配置文件的方式进行数据管理，实现了音乐播放器的系统编程。

综上所述，本文开发的音乐播放器针对用户和开发的音乐播放器具有如下特点：

(1) 该音乐播放器测试分为模拟器和不同型号的真机模拟进行测试，效果不错。

(2) 该音乐播放器占用内存仅仅 168K，如图 6(d)所示，以虾米音乐对比，仅占其十分之一。

(3) 该音乐播放器可以配备静态图片辅以歌词，便于用户听看同步。

(4) 该音乐播放器辅以音轨，不仅考虑用户体验性，而且便于用户随心听曲。

(5) 该音乐播放器针对手机内存资源匮乏，上网流量紧张的普通用户。

(6) 该音乐播放器的歌曲来自于手机内部，需要用户通过蓝牙、数据线或 WIFI 导入手机内部。对于 WIFI 普及的情况，结合小众用户，具有一定的市场应用价值。

本软件经测试，功能完备，界面简洁，适用方便，能够满足用户的基本需求。在不同的机型上进行测试时均表现出色，运行速度，打开速度，优化资源等方面都远远超过其他音乐播放软件，避免了界面花哨、功能庞大带来的资源浪费问题，软件实用性大为提升。

在今后的研究中，将增加音乐分享的功能，并将其升级为集音频、视频于一体的综合多媒体播放器。

基金项目

北京联合大学“启明星”大学生科技创新项目(12222994701, 12222994501)，北京联合大学新起点计划项目资助(zk10201303)，北京市职业院校教师素质提高工程资助项目。

参考文献 (References)

[1] 陈文, 王琳燕 (2015) 个性化 Android 手机音乐播放器的实现. *电脑编程技巧与维护*, **16**, 39-41.

[2] 王秀芳, 杨阳 (2011) 基于 Android 的 YOBO 在线音乐播放器. *科学技术与工程*, **11**, 2506-2509, 2518.

[3] 王玉凡 (2014) 基于 Android 平台的手机音乐播放器的设计与实现. *河北软件职业技术学院学报*, **1**, 57-60.

[4] 谭静, 黄甫道辉 (2014) 基于用户体验的 Android 手机音乐播放器设计与实现. *软件*, **11**, 42-44.