

Design and Implementation of Music Player Based on Android

Jiwen Yu, Yifu Zhang, Zhongyi Xu, Ruixia Sun*

School of Electrical Engineering, Anhui Polytechnic University, Wuhu Anhui
Email: *sunrx_1@126.com

Received: May 8th, 2019; accepted: May 21st, 2019; published: May 28th, 2019

Abstract

In view of the problems existing in music player on the Android market at home and abroad, function and performance of the system is firstly analyzed in the paper according to user survey. Then, the overall design of the system is carried out based on Android, and the detailed design and implementation of each function module is carried out. The key technologies and difficulties, such as database, message processing mechanism are mainly studied. At last, function and performance test of the music player software are given, and the results are satisfactory. The software has great practical value, which has good user experience, fast response, simple and generous play interface.

Keywords

Music, Android, Message Processing Mechanism, Database, JAVA

基于Android的音乐播放软件设计与实现

余纪文, 张一夫, 许忠义, 孙瑞霞*

安徽工程大学电气工程学院, 安徽 芜湖
Email: *sunrx_1@126.com

收稿日期: 2019年5月8日; 录用日期: 2019年5月21日; 发布日期: 2019年5月28日

摘要

针对目前国内外Android市场上音乐播放软件存在的问题, 本文根据用户多方面调查, 对系统的功能及性能进行了需求分析。基于Android平台设计和开发了一个音乐播放客户端, 对各个功能模块进行详细的设计与实现。重点对数据库、消息处理机制等核心技术及难点进行了研究。功能以及性能测试得到了

*通讯作者。

满意的结果。用户体验好，响应快，播放界面简洁大方，具有较大的实用价值。

关键词

音乐, Android, 消息处理机制, 数据库, JAVA

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着智能手机和终端的普及，相继出现了各类 APP。基于 Android 平台开发的手机应用有着基本的通用性，可以在各个厂商的手机中普遍安装和使用[1]。目前，国内外主流的基于 Android 平台的音乐播放软件有天天动听、酷狗音乐、QQ 音乐等。但大多数一味的追求外观漂亮，功能强大，造成用户视觉疲劳，手机资源占用太多，多任务操作时响应较慢，用户体验较差[2] [3]。本文旨在研究开发一款基于 Android 平台的实用的个性化音乐播放客户端。

2. 系统需求分析

1) 总体需求分析

针对 Android 手机用户，设计与实现一款简洁大方、易于操作的个性化音乐播放客户端，能够支持 MP3、WAV 等多种音频格式。

2) 功能需求分析

基本播放控制功能：播放/暂停、选择上一首/下一首、选择播放模式、显示歌词、拖动播放进度条、打开文件管理、后台播放。

侧滑菜单功能：扫描歌曲、切换播放模式、换背景、睡眠、退出、系统设置。

歌曲索引及定位功能：当内存中歌曲数量较多时，本文提供两种查找方式：a) 关键字查找：根据用户输入的文字进行关键字索引。b) 九键首字母查找：以字首字母为索引，并定位到相应歌曲。

3) 性能需求分析

为了在基于 Android 的手机上达到良好的用户体验，要求：a) 响应速度快，歌曲播放/暂停、选择上一首/下一首，拖动进度条等操作均不能超过 5 s；b) 人性化的播放界面，体现出简洁大方、易于操作的特点；c) 无缝衔接。程序运行时，要保证一些突发事件的处理，如电话、短信等，场景切换过程中需要做到无缝衔接[2] [4]。

3. 系统设计与实现

3.1. 软件架构设计

图 1 为本文系统的软件架构图，分为音乐播放界面层和音频处理层两层。界面层与用户直接交互。音频处理层对音乐文件进行处理，播放 SD 卡中的 mp3、wav、midi、wma 等音乐文件。首先在数据提取层将音乐文件从 SD 卡中提取出来，传送到数据处理层进行封装，并将其传送到解码层，使用相应的解码器解码文件，并进行播放。双箭头表示两方进行数据交互，单箭头表示单向传输数据。

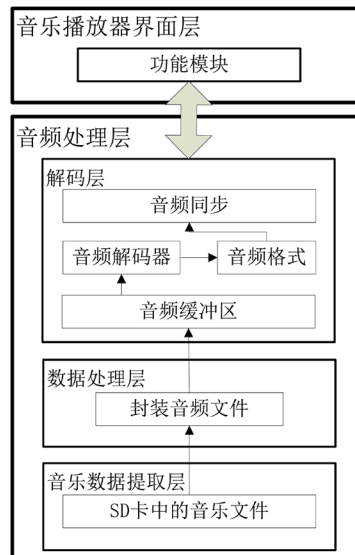


Figure 1. Software architecture
图 1. 软件架构图

3.2. 系统功能设计

根据功能需求分析, 对系统进行功能设计。整个音乐播放软件划分为四大功能模块: 播放控制模块、播放列表模块、文件管理模块、搜索模块。系统功能结构图如图 2 所示。

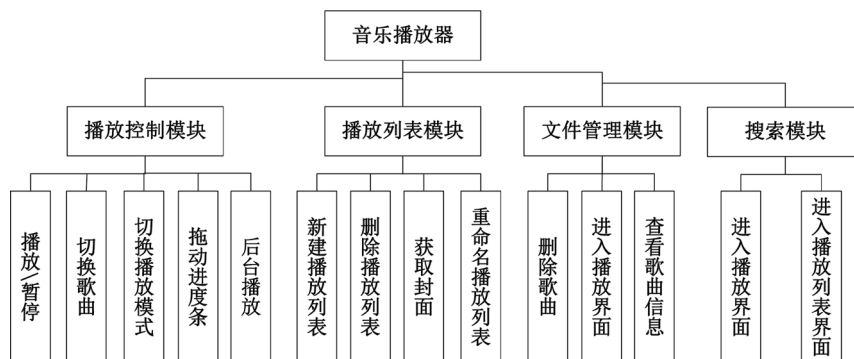


Figure 2. System functional structure diagram
图 2. 系统功能结构图

根据功能结构图, 设计出其内部实现的框架结构图, 如图 3 所示。图中虚线代表依赖于关系。

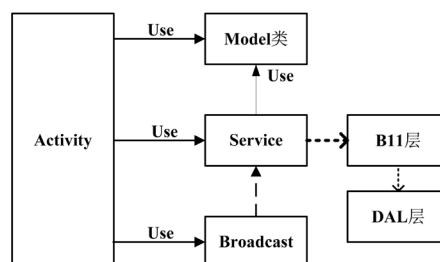


Figure 3. System framework diagram
图 3. 系统框架结构图

- 1) Activity: 与用户进行交互的界面;
- 2) Model 类: 存放歌曲信息的类;
- 3) Service: 用户不可见, 实现歌曲的播放/暂停等功能;
- 4) Broadcast: 用来实现消息的广播;
- 5) B11 层: 用来实现接口;
- 6) DAL: 依赖于 B11 层, 存放数据库的操作类, 用来创建、实现、删除数据库。

3.3. Android 开发环境的搭建

本文开发的音乐播放软件是在 Android 平台上, Java 语言编程, 所需软件和开发工具有 JDK、Android Studio、Android SDK。

环境变量的设置: 将 JDK 开发包中的 bin 文件夹的绝对路径、SDK 中的 tools 文件的绝对路径放到“Path”变量中; 添加 JDK 中的 lib 以及 demo 文件夹的路径到“CLASSPATH”变量。

建立一个新的工程, 包括三个文件夹:

- 1) java 文件夹: 存放工程的 Java 文件。
- 2) res 文件夹: 包含整个工程的所有资源, drawable(图标)、layout(布局文件)、values(常量)等。
- 3) AndroidManifest.xml 文件: 包括了程序中要用到的全部活动、服务、接收器, 跟 Android 安全相关的文件[5] [6]。

3.4. 系统界面设计与实现

根据系统需求分析, 界面的设计: 要求界面简洁, 因此采用线性布局的方式; 使用触控方式对界面进行操作; 对不同的功能使用不同的屏幕表示。

- 1) 播放控制主界面
设计模型如图 4 所示。

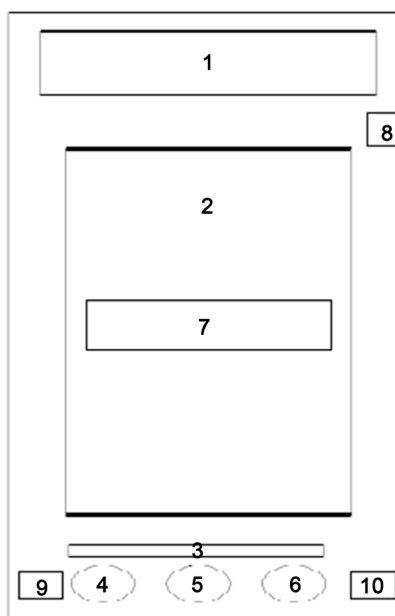


Figure 4. Layout of playback control home interface
图 4. 播放控制主界面布局

数字说明:

- ① TextView 控件, 显示歌词信息, 如歌曲及艺术家名称等。
- ② 显示歌词, 通过匹配的歌词在此处滚动播放。
- ③ SeekBar 控件, 进度条。
- ④ Button 控件, 用户点击该按钮切换到上一首歌曲。
- ⑤ Button 控件, 播放/暂停按钮。
- ⑥ Button 控件, 下一首按钮。
- ⑦ 无歌词提示文本, 平时处于隐藏状态, 当没有找到匹配歌曲时显示。
- ⑧ 声音设置按钮。
- ⑨ 我的最爱快捷键, 将该歌曲加入“我的最爱”表中。
- ⑩ Button 控件, 当前歌曲表单对话框按钮。

2) 播放列表界面

使用 ListView、TextView、ImageView 进行线性布局, 其中 TextView 显示列表名称, 如图 5 所示。

3) 侧滑菜单界面

主要使用 ListView、TextView、ImageButton、Button、EditText、ImageView 进行布局, 使用第三方的开源库 SlidingMenu 来实现侧滑菜单的功能。侧滑菜单界面布局如图 6 所示。

```
mSlidingMenu = new SlidingMenu(this);
mSlidingMenu.setTouchModeAbove(SlidingMenu.TOUCHMODE_MARGIN);
mSlidingMenu.setMode(SlidingMenu.RIGHT); mSlidingMenu.setShadowWidthRes(R.dimen.shadow_width); //设置阴影的宽度
mSlidingMenu.setShadowDrawable(R.drawable.shadow); //滑动菜单的阴影效果
mSlidingMenu.setBehindOffsetRes(R.dimen.slidingmenu_offset); //设置偏移量
mSlidingMenu.setFadeDegree(0.35f); mSlidingMenu.attachToActivity(this, SlidingMenu.SLIDING_CONTENT); //设置布局
mSlidingMenu.setMenu(R.layout.frame_menu);
```

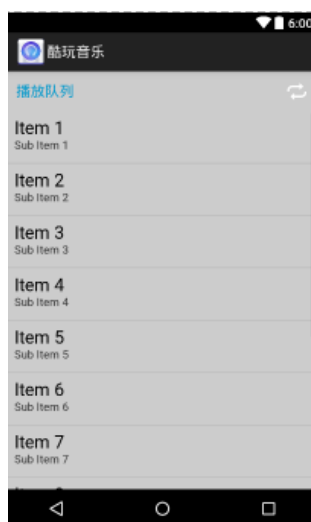


Figure 5. Layout of playlist interface

图 5. 播放列表界面布局



Figure 6. Layout of side slide menu interface
图 6. 侧滑菜单界面布局

4) 搜索界面

主要使用 EditText、ImageView、ListView 进行线性布局，如图 7 所示。

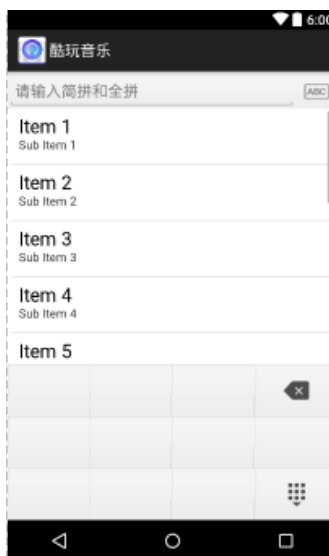


Figure 7. Layout of search interface
图 7. 搜索界面布局

3.5. 播放功能的实现

在 Android 中要实现某个功能，首先需要实现与用户进行交互的控件，对用户的触摸发生响应；其次是各个界面之间的切换，需要对进程的生命周期进行控制和管理等。音乐播放功能主要使用 MediaPlayer 类来完成。

1) 创建与销毁

使用 new 或 create() 创立一个该类的实例对象。release() 释放不需要的内存。

2) 初始化

首先使用重载的 `setDataSource()` 将对象转变到 `initialized` 形态, 接着转变为 `prepared` 形态。此时 `MediaPlayer` 对象获得歌曲时长等信息, 使用 `setVolume` 设定播放器的音量。

3) 播放、暂停与停止

当对象是 `prepared` 状态时, 用 `start()` 转变成 `started` 状态, 用 `pause()` 转变成 `paused` 状态。调用 `isPlaying()` 判别对象是不是位于 `started` 状态。

4) 进度条

通过 `seekTo()` 方法来控制对象的媒体时间, `onSeekCompleteListener` 来监听进度条的播放进度。

5) 错误处理

当播放歌曲时会发生各种错误, 例如文件格式错误, 或用户操作错误等。为了应对这些情况, `MediaPlayer` 对象设置了 `onErrorListener`。有错误时, 就用 `onError()` 使对象进入到 `error` 状态。如果想恢复播放器的状态, 就使用 `reset()` 使对象再次进入 `idle` 状态。

4. 关键技术分析

4.1. 数据库设计与分析

数据库设计会直接影响到系统运行的效果。本文用到的数据库主要是 `SQLite`, 一个支持 `SQL` 语言标准的开源的 `Android` 数据库, 只需要几百 `KB` 的内存, 可以在任意操作系统中使用。

4.1.1. 数据库存储设计

根据系统的功能结构图, 以及对所需数据的分析,

数据库主要设计以下表格:

1) 歌曲信息表, 主要用来存放歌曲 `id`, 歌曲名称, 艺术家, 专辑路径, 大小, 歌词等信息, 如表 1 所示。

Table 1. Song information table
表 1. 歌曲信息表

属性	字段类型	说明	主键	空值
<code>_ID</code>	<code>INTEGER</code>	歌曲 <code>id</code> 号	<code>Primary key</code>	不允许
<code>NAME</code>	<code>TEXT</code>	歌曲名称		允许
<code>ARTIST</code>	<code>TEXT</code>	艺术家		允许
<code>ALBUM</code>	<code>TEXT</code>	专辑		允许
<code>SIZE</code>	<code>LONG</code>	大小		允许
<code>PATH</code>	<code>TEXT</code>	歌曲路径		允许
<code>TYPE</code>	<code>TEXT</code>	歌曲列表名		允许

2) 已有播放列表, 如表 2 所示。

Table 2. Existing playlists
表 2. 已有播放列表

属性	字段类型	说明	主键	空值
<code>_ID</code>	<code>INTEGER</code>	播放列表 <code>id</code> 号	<code>Primary key</code>	不允许
<code>NAME</code>	<code>TEXT</code>	播放列表名		允许

3) 当前播放列表，与表 2 类似。

以上数据库之间的关系如图 8 所示。

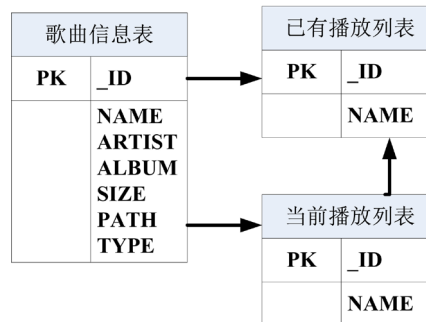


Figure 8. Diagram of three tables
图 8. 三表关系图

4.1.2. 数据库的实现

在开发过程中，通过继承 SQLiteOpenHelper 类来建立和管理数据库。参数 context 用来判断打开还是创建数据库，name 表示数据库名称，factory 用来创建游标，version 代表当前 database 的版本号。

通过构造方法来建立数据库：数据库不存在时，建立且打开数据库，再执行创立表的操作；数据库存在时，检测从构造函数传入的版本号，使用 onUpgrade()更新数据库和版本号，再重新调用 onCreate 创建新的数据库信息。关键代码：

```
public class DatabaseHelper extends SQLiteOpenHelper {
    private static SQLiteDatabase mDb;
    private static DatabaseHelper mHelper;
    private static final int DB_VERSION = 3;
    private static final String DB_NAME = "musicstore_new";
    private static final String TABLE_ALBUM = "album_info";
    private static final String TABLE_ARTIST = "artist_info";
    private static final String TABLE_MUSIC = "music_info";
    private static final String TABLE_FOLDER = "folder_info";
    private static final String TABLE_FAVORITE = "favorite_info";
    /**创建主界面中的五个表格*/
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table " + TABLE_MUSIC + " (_id INTEGER PRIMARY KEY
        AUTOINCREMENT," + " songid integer, albumid integer, duration integer, musicname varchar(10), " +
        "artist char, data char, folder char, musicnamekey char, artistkey char, favorite integer)");
        db.execSQL("create table " + TABLE_ALBUM + " (_id INTEGER PRIMARY KEY
        AUTOINCREMENT, " + "album_name char, album_id integer, number_of_songs integer, album_art
        char)");
        .....
    }
}
```



```

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    if (newVersion > oldVersion) {
        db.execSQL("DROP TABLE IF EXISTS +TABLE_ARTIST);
        db.execSQL("DROP TABLE IF EXISTS +TABLE_ALBUM);
        db.execSQL("DROP TABLE IF EXISTS +TABLE_MUSIC);
        db.execSQL("DROP TABLE IF EXISTS"+TABLE_FOLDER);
        onCreate(db);/*创建新的数据库信息*/
    }
}
.....
}

```

4.2. 消息处理机制

由于音乐播放界面和其他控件之间的通讯十分频繁，通过 Android 的 Handler 消息机制来处理。当处理视图对象时，需要对控件进行实例化，然后将控件 ID 与对应的配置文件进行绑定，并设置监听器进行监听，最终通过重写 `onClick()` 响应用户的操作，并进行状态刷新[2] [7]。消息处理的流程：

- 1) 封装消息对象，将指定的 Handler 以及各种回调函数等进行封装；
- 2) 使用 Handler 的 `sendMessage()` 来发送消息；
- 3) 把消息加入到 Looper 的消息队列；
- 4) Looper 则将循环的从消息队列中拿出消息，并进行处理，且删除处理过的消息。

以“扫描歌曲”为例，在布局文件中定义按钮“scanBtn”：

```

<Button
    android:id="@+id/scanBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="70dip"
    android:text="开始扫描"
    android:textSize="22sp"
    android:textColor="@color/white" />

```

在 activity 中，`onClick()` 响应用户的操作：

```

public void onClick(View v) {
    switch (v.getId()) {
        case R.id.keyboard_switcher:
            if (mIsT9Keyboard) { // 若 T9 隐藏，打开 T9
                mKeyboardSwitcherIv.setImageResource(R.drawable.keyboard_switch);
                mSearchInputEt.setHint("请输入简拼或全拼");
                mSearchInputEt.setInputType(InputType.TYPE_NULL);
                mInputMethodManager.hideSoftInputFromWindow( mSearchInputEt.getWindowToken(), 0); // 隐藏输入法
                mKeyboardLayout.setVisibility(View.VISIBLE);
            }
    }
}

```

```

    break;
  }
}

```

5. 系统测试

测试建立在硬件系统良好的前提下，根据用户对系统功能及界面的需求制定相应的测试用例，得到测试结果[7] [8]。测试环境：Android 模拟器，4.0.1，手机。

5.1. 功能测试

播放和进度条测试：效果图如图 9 所示，播放进度条在播放时可以自行进行刷新，并显示播放时间，左边显示为已经播放时间，右边显示剩余时间。上面是 Seekbar 正常状态，下面是其被选中时的效果。

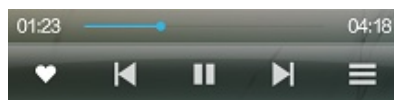


Figure 9. Rendering of playback and progress bar
图 9. 播放和进度条效果图

后台播放测试：从图 10 可以看出后台播放时出现的后台图标，以及用户可以在下拉框中看到播放器的播放基本控制键，方便用户进行操作。

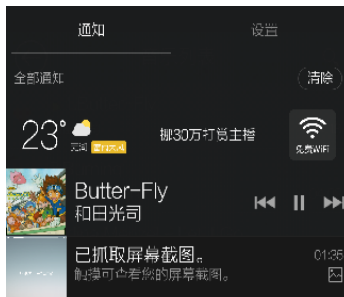


Figure 10. Rendering of background playback
图 10. 后台播放效果图

播放列表测试：图 11 中显示的常驻快捷播放栏，上方是当前数据库中的音乐列表，当前播放的歌曲有图片提示。

侧滑菜单测试：如图 12 所示，当用户向左滑动时，可以看到出现的菜单栏。

常驻菜单栏及主界面测试：如图 13，主界面将用到的五个表以按钮的形式呈现给用户，点击相应的按钮即可进入对应的列表。下方的菜单栏和图 12 中的菜单栏是一致的。

播放界面效果图如图 14 所示，达到了预期效果。

5.2. 性能测试

本文对手机预装音乐播放软件进行了 100 次测试。音乐播放流畅，音质清晰，系统的稳定性满足用户需要。提取每十次测试的系统响应时间平均数据，画出折线图，并与市场上其他音乐播放软件进行对比，结果如图 15 所示。实折线表示 Android 市场上一般音乐播放软件的响应时间，虚折线表示本文系统的响应时间。可以看出，系统的响应时间达到了要求，并体现了其响应迅速的优势[9]。

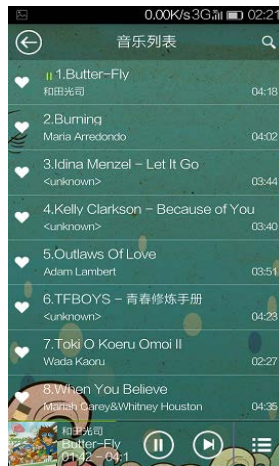


Figure 11. Playlist rendering
图 11. 播放列表效果图



Figure 12. Rendering of side slide menu bar
图 12. 侧滑菜单栏效果图



Figure 13. Main interface effect diagram
图 13. 主界面效果图



Figure 14. Play interface effect diagram
图 14. 播放界面效果图

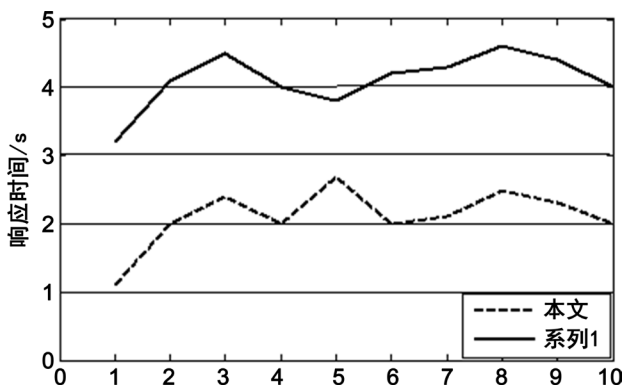


Figure 15. Comparisons of system response time
图 15. 系统响应时间对比折线图

6. 结论

在分析国内外已有 Android 平台音乐播放软件的基础上, 结合对用户的调查, 得出本文音乐播放软件的功能和性能需求, 对系统进行了总体的设计, 并对各个模块进行详细的设计, 讨论了数据库、消息处理机制等关键技术。以 Android 为平台, JAVA 语言为主导, 实现了 UI 界面及功能。最后对系统进行了功能及性能上的测试, 并与市场上音乐播放软件进行了对比。测试表明, 该音乐播放软件界面清晰简单, 操作方便, 占用手机资源较少, 响应速度快, 达到了预期的目标, 具有较大的市场空间。实际应用中, 还可以在流畅交互方面进一步研究完善[10]。

基金项目

安徽省大学生创新创业训练项目(201710363159)。

参考文献

- [1] 郭金尚, 等. Android 经典项目案例开发实战宝典[M]. 北京: 清华大学出版社, 2013: 22-30.
- [2] 徐艳菲, 吴铁峰. 基于 Android 的音视频播放器的研究与设计[J]. 微处理机, 2017(6): 74-77.
- [3] 张亚杰. 基于 Android 平台的移动终端应用程序的研究与开发[D]: [硕士学位论文]. 郑州: 郑州大学, 2013:

16-24.

- [4] 韩文智, 骆文亮. Android 平台的移动 APP 开发方法与应用研究[J]. 四川理工学院学报(自科版), 2015, 28(3): 22-26.
- [5] 张晶. 基于 Android 平台的物流系统设计研究[J]. 制造业自动化, 2013, 35(4): 153-155.
- [6] Chung, S.M. and Wu, C.T. (2017) Designing Music Games and Mobile Apps for Early Music Learning. *Serious Games and Edutainment Applications*, 86-100. https://doi.org/10.1007/978-3-319-51645-5_3
- [7] 陈浩, 李本富. 基于 Android 云计算的移动点餐系统[J]. 计算机系统应用, 2012, 21(8): 148-151.
- [8] 唐敏. 基于 Android 平台的通讯帮手的设计与开发[J]. 计算机科学, 2012, 39(S1): 573-576.
- [9] 倪红军. 基于 MVP 模式的 Android 应用开发研究[J]. 电子设计工程, 2018, 26(11): 12-15+19.
- [10] Nathan, K.S., Arun, M. and Kannan, M.S. (2017) EMOSIC—An Emotion Based Music Player for Android. *IEEE International Symposium on Signal Processing & Information Technology*, 271-276. <https://doi.org/10.1109/ISSPIT.2017.8388671>

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2161-8801, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: csa@hanspub.org