**实 验 报 告**

**课程名称**  移 动 应 用 系统

**实验项目**  实验三：多媒体设计与开发

**实验仪器** 计算机一台、Android手机一部

**系 别**  软件工程

**专 业** 软件工程

**班级/学号**  软工1801 /2018011192

**学生姓名** 刘宏远

**实验日期**  2020/11/15

**成 绩**

**指导教师**  郝 保 水

# 实验三 多媒体设计与开发

1. 实验目的
   1. 了解Android Service的基本概念；
   2. 理解并掌握使Service使用方法;
   3. 掌握MediaPlayer的使用方法。

2实验要求

1. 课前预习实验内容，并查找相关资料。
2. 按照实验步骤完成各个相关内容。
3. 撰写实验报告。
   1. 实验报告格式必须符合学校要求（例如必须采用学校规定的实验封面）；
   2. 写出实验详细步骤，包括主要采用的技术方案、相关分析和核心代码。注意：不要简单地近包括截屏和代码，完整代码可以作为附录放在实验报告结尾；
   3. 总结实验中遇到的问题、分析和解决方法。
   4. 写出心得体会与收获等。

3实验步骤

主要分为几个大步骤进行实现。

1. 需求分析，调研播放器具有哪些功能，确定界面。
2. 系统设计，对系统进行模块划分，确定技术方案。
3. 编码，编写代码实现各项功能。
4. 测试。

3.1需求分析

播放器可以播放音乐。我们这里仅实现基本功能。

* 主要功能
* 用户选择音乐文件
* 创建播放列表
* 多种播放模式：单首一次、单首多次、多首一次、多首多次，顺序播放、随机播放等。
* 界面设计

界面可以参考目前主流的播放器界面。当然也可自行设置其他界面。下图给出了一个简单的界面。



图 4‑1音乐播放器界面

3.2系统设计

歌单（随机播放、顺序播放）：

//开始播放

buttonStart.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

try {

Log.v(TAG, "start");

mediaPlayer.reset();

AssetManager assetManager = getAssets();

AssetFileDescriptor assetFileDescriptor = assetManager.openFd("Carly Rae Jepsen - I Really Like You.mp3");

mediaPlayer.setDataSource(assetFileDescriptor.getFileDescriptor(), assetFileDescriptor.getStartOffset(), assetFileDescriptor.getLength());

mediaPlayer.prepare();

mediaPlayer.start();

buttonPause.setEnabled(true);

buttonStop.setEnabled(true);

buttonLoop.setEnabled(true);

} catch (IllegalArgumentException e) {

e.printStackTrace();

} catch (IllegalStateException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

});

//暂停播放

buttonPause.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (mediaPlayer.isPlaying()) {

buttonPause.setText("Play");

mediaPlayer.pause();

} else {

buttonPause.setText("Pause");

mediaPlayer.start();

}

}

});

//停止播放

buttonStop.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (mediaPlayer.isPlaying())

mediaPlayer.stop();

}

});

//循环播放

buttonLoop.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Log.v(TAG, "Looping");

boolean loop = mediaPlayer.isLooping();

mediaPlayer.setLooping(!loop);

if (!loop)

txtLoopState.setText("循环播放");

else

txtLoopState.setText("一次播放");

}

});

}

基本播放（播放、暂停）：

其他按钮（上一首、下一首）：

进度条（可拖动到任意位置）：

歌曲文件（数据库）：

界面布局（相对布局）：

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto" android:layout\_width="match\_parent"

android:layout\_height="match\_parent" android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"

tools:showIn="@layout/activity\_main" tools:context=".MainActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/txtLoopState"

/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/buttonStart"

android:text="@string/action\_start"

android:layout\_below="@id/txtLoopState"

/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/buttonPause"

android:text="@string/action\_pause"

android:layout\_below="@id/buttonStart"

/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/buttonStop"

android:text="@string/action\_stop"

android:layout\_below="@id/buttonPause"

/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/buttonLoop"

android:text="@string/action\_loop"

android:layout\_below="@id/buttonStop"

/>

</RelativeLayout>

3.3编码实现

主要步骤：

1. 创建工程
2. 编写各界面布局、菜单等资源文件
3. 编写界面处理代码

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);

fab.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH\_LONG)

.setAction("Action", null).show();

}

});

mediaPlayer = new MediaPlayer();

/\*如果需要播放完停止，则需要注册OnCompletionListener监听器

在本示例中，用户可以选择是否循环播放，如果选择了循环播放，则播放完后会自动转到Started状态，再次播放

mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {

@Override

public void onCompletion(MediaPlayer mp) {

Log.v(TAG,"setOnCompletionListener");

mp.release();

}

});\*/

final TextView txtLoopState = (TextView) findViewById(R.id.txtLoopState);

final Button buttonStart = (Button) findViewById(R.id.buttonStart);

final Button buttonPause = (Button) findViewById(R.id.buttonPause);

final Button buttonStop = (Button) findViewById(R.id.buttonStop);

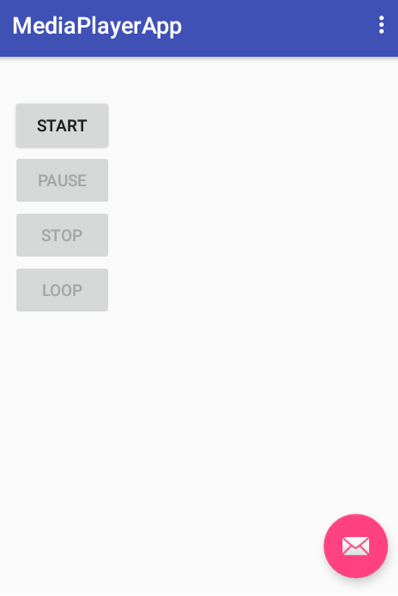
final Button buttonLoop = (Button) findViewById(R.id.buttonLoop);

buttonPause.setEnabled(false);

buttonStop.setEnabled(false);

buttonLoop.setEnabled(false);

3.4系统测试



4.问题分析（也可以放在实验步骤中写，即写了相关技术或代码后，运行时发现问题，对问题进行分析讨论）

写出问题涉及的相关技术或知识点、问题分析、解决办法等。

歌曲文件无法导入数据库，百度解决

歌曲播放无声音，百度解决

5.实验心得与体会

实验收获、心得、总结等。

内容不少于500字。

收获：学会了MediaPlayer的基本使用，包括其一系列的回调函数，了解了一般播放器的功能和界面。

心得：实践出真知

总结：音乐播放要合理利用其回调函数

实验报告一定要注意：

1.不要抄袭雷同，一旦发现，不区分抄袭者和被抄袭者，一律0分；

注意，一定不要雷同，不要抄袭！

注意，一定不要雷同，不要抄袭！

注意，一定不要雷同，不要抄袭！

2.格式一定要正确，不要缺少封皮、实验目的、实验内容步骤、心得等；

3，不要代码截屏，我通过github看大家代码，不用为了显得篇幅长而在实验报告中大量粘贴代码；

4.按时提交，过了时间点后系统不允许提交。