

《Android应用开发》课程

实验报告

姓名 苏桐渤
学号 2019212212236
班级 软工192

实验名称 多媒体的使用 - 音乐器播放
实验日期 2021/11/26

实验目的

了解和掌握如何获取系统中所有的音乐信息，掌握控制音频播放的基本方法，熟练使用 MediaPlayer 控制音频播放。

实验内容和要求

- (1) 创建音乐播放器的基本布局；
- (2) 获取系统中所有的 MP3 音频数据并显示在 ListView 组件中。
- (3) 实现对列表中音频文件的播放控制。

实验过程及运行结果

一、创建显示歌曲列表 ListView 相关的控件

- 自定义控件-布局

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="40dp">
```

```

<TextView
    android:id="@+id/music_item_text_view_id"
    android:layout_width="20dp"
    android:layout_height="match_parent" />

<TextView
    android:id="@+id/music_item_text_view_title"
    android:layout_width="150dp"
    android:layout_height="match_parent" />

<TextView
    android:id="@+id/music_item_text_view_artist"
    android:layout_width="150dp"
    android:layout_height="match_parent" />

<TextView
    android:id="@+id/music_item_text_view_duration"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

</LinearLayout>

```

- 自定义类，关联每条音乐的信息

```

public class MusicItem {
    private String id;
    private String title;
    private String artist;
    private String duration;
    private String path;

    ...
}

```

- 适配器

```

public class MusicAdapter extends ArrayAdapter<MusicItem> {

    private int resourceId;

    public MusicAdapter(@NonNull Context context, int resource, @NonNull
List<MusicItem> objects) {
        super(context, resource, objects);
        this.resourceId = resource;
    }

    ...
}

```

- listview

```

musicAdapter = new MusicAdapter(MainActivity.this, R.layout.music_item,
this.list);
listView = (ListView) findViewById(R.id.main_list_view);
listView.setAdapter(musicAdapter);

```

二、获取手机中的音频文件

- 申请权限

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

- 读取音频文件

```
Cursor cursor =  
getContentResolver().query(MediaStore.Audio.Media.EXTERNAL_CONTENT_URI, null,  
null, null, MediaStore.Audio.Media.DEFAULT_SORT_ORDER);
```

- 加载到listview中

```
list.add(new MusicItem(String.valueOf(++id), title, artist, getLength(duration),  
path));
```

三、点击事件：完成音乐的播放和暂停

- 播放、暂停、上一首、终止、下一首

```
public void onClick(View view) {  
    switch (view.getId()) {  
        case R.id.main_button_play:  
            videoPlay();  
            break;  
        case R.id.main_button_pause:  
        case R.id.main_button_stop:  
            if (videoView.isPlaying()) {  
                videoView.pause(); // 暂停播放  
            }  
            break;  
        case R.id.main_button_prev:  
            if (current > 0) {  
                videoView.setVideoPath(list.get(--current).getPath());  
                videoPlay();  
            }  
            break;  
        case R.id.main_button_next:  
            if (current < list.size() - 1) {  
                videoView.setVideoPath(list.get(++current).getPath());  
                videoPlay();  
            }  
            break;  
    }  
}
```

- 点击listview 中的音乐，实现跳转播放

```

listview.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long
1) {
        current = i;
        videoView.setVideoPath(list.get(current).getPath());
        show.setText(list.get(i).getTitle());
        videoPlay();
    }
});

```

问题分析与总结

- 权限问题

```

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

```

```

if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
    init();
} else {
    Toast.makeText(this, "拒绝权限将无法使用程序", Toast.LENGTH_SHORT).show();
    finish();
}

```

- 顺序问题

要先获得读取文件的权限，再读取文件加载listview，再从列表中获取音乐进行播放