# 实验六 多媒体

学号：2021329600006　　姓名：陈昊天

班级：计算机科学与技术 21（4）班　联系电话：13456982338

实验地点：10-414

实验日期：2023 年 12 月 14 日

## 一、课程目标

1、了解调用摄像头拍照和打开相册中图片的方法。
2、掌握播放本地音频和视频文件的方法。

## 二、作业内容

基本要求：设计开发一个简单的本地视频播放器。从网上下载 3 个视频，保存在 res\raw 目录里，使用 Recycleview 显示视频列表（实际显示图片和文字），点击图片或文字，利用 VideoView 实现视频的播放功能。
选做:模仿今日头条、抖音等软件，滑动视频列表，自动播放当前视频。

## 三、详细设计过程(截图并文字描述设计方法、思路、关键源代码等)

### 3.1 需求分析和规划

功能定义：明确应用的主要功能，即播放本地视频。包括视频的搜索、展示和播放。
用户界面布局：考虑如何有效地展示视频列表。决定使用 RecyclerView ，它提供灵活的布局和高效的滚动性能。
视频播放策略：确定使用 VideoView 和 MediaController 来实现视频播放，因为它们提供了基本的播放控制和易于实现的接口。
权限管理：设计一个流程来请求和验证存储访问权限，这对于访问和播放本地视频文件至关重要。

### 3.2 设计细节

存储访问和视频文件检索：
实现一个递归方法来遍历存储设备，寻找视频文件。
保证这一过程不会阻塞主线程。

视频列表展示：
使用 RecyclerView 搭配 GridLayoutManager 显示视频文件，为用户提供一个直观的网格视图。
每个视频项应显示缩略图和文件名。缩略图的生成应考虑性能和内存占用。

视频播放功能：
当用户选择一个视频时，启动一个新的 Activity（videoPlayerActivity）。
在新的 Activity 中使用 VideoView 加载和播放视频。MediaController 提供基本的播放控制。

用户交互和反馈：
在请求权限和加载视频时，给予用户适当的反馈。
提供一个简洁且直观的用户界面，使用户易于理解如何操作应用。

### 3.3 关键源代码

权限请求机制：

permissionForVideo() 方法
这个方法用于检查应用是否已经获得读取外部存储的权限。如果没有，它会向用户请求该权限。

```
private void permissionForVideo() {
  if ((ContextCompat.checkSelfPermission(getApplicationContext(),
       Manifest.permission.READ_EXTERNAL_STORAGE) !=
       PackageManager.PERMISSION_GRANTED)) {
           if((ActivityCompat.shouldShowRequestPermissionRationale
           (MainActivity.this,
              Manifest.permission.READ_EXTERNAL_STORAGE))) {
       } else {
           ActivityCompat.requestPermissions(MainActivity.this, new
           String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
           REQUEST_PERMISSION);
       }
    } else {
       bolean_permission = true;
       getFile(directory);
       obj_adapter = new MyAdapter(getApplicationContext(),
       fileArrayList);
       myRecyclerView.setAdapter(obj_adapter);
    }
}
```

onRequestPermissionsResult()方法
这个方法处理用户对权限请求的响应。如果用户授予了权限，它将继续获取视频文件并更新适配器。

```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode,     permissions,
    grantResults);
    if (requestCode == REQUEST_PERMISSION) {
        if (grantResults.length > 0 && grantResults[0] ==
        PackageManager.PERMISSION_GRANTED) {
            bolean_permission = true;
            getFile(directory);
            obj_adapter = new MyAdapter(getApplicationContext(),
            fileArrayList);
            myRecyclerView.setAdapter(obj_adapter);
        } else {
            Toast.makeText(this, "Please Allow the Permission",
            Toast.LENGTH_SHORT).show();
        }
    }
}
```

视频检索和展示：

getFile(File directory) 方法
这个方法递归地检索指定目录（及其子目录）下的所有 .mp4 视频文件，并将它们添加到 fileArrayList。

```
public ArrayList<File> getFile(File directory) {
    File listFile[] = directory.listFiles();
    if (listFile != null && listFile.length > 0) {
        for (int i = 0; i < listFile.length; i++) {
            if (listFile[i].isDirectory()) {
                getFile(listFile[i]);
            } else {
                bolean_permission = false;
                if (listFile[i].getName().endsWith(".mp4")) {
                    for (int j = 0; j < fileArrayList.size(); j++) {
                        if(fileArrayList.get(j).getName().equals(list
                            File[i].getName())) {
                            bolean_permission = true;
                        }
                    }
                    if (bolean_permission) {
                        bolean_permission = false;
```

```
                } else {
                    fileArrayList.add(listFile[i]);
                }
            }
        }
    }
    return fileArrayList;
}
```

视频播放处理:

playerVideo()方法
这个方法在 `videoPlayerActivity` 中负责设置 `VideoView` 和 `MediaController`，并开始播放用户选择的视频。

```
private void playerVideo() {
    MediaController mediaController = new MediaController(this);
    mediaController.setAnchorView(videoView);
    videoView.setMediaController(mediaController);
    videoView.setVideoPath(String.valueOf(MainActivity.fileArrayLi
    st.get(position)));
    videoView.requestFocus();
    videoView.setOnPreparedListener(new
    MediaPlayer.OnPreparedListener() {
        @Override
        public void onPrepared(MediaPlayer mp) {
            videoView.start();
        }
    });

    videoView.setOnCompletionListener(new
        MediaPlayer.OnCompletionListener() {
        @Override
        public void onCompletion(MediaPlayer mp) {
            videoView.setVideoPath(String.valueOf(MainActivity.file
                ArrayList.get(position = position + 1)));
            videoView.start();
        }
    });
}
```

### 3.4 运行结果

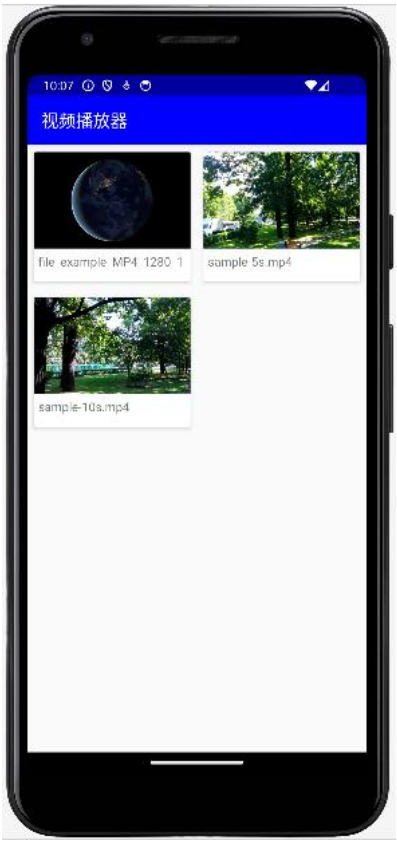


图 3.1 视频展示界面


图 3.2 视频播放界面

## 四、学习体会：

在完成这次多媒体实验的过程中，我深刻体会到了理论知识与实践技能相结合的重要性。在处理 Android 应用中的视频播放和列表展示方面，我不仅将课堂上学到的理论知识应用到了实际项目中，而且还在实践中加深了对这些理论的理解。通过这个项目，我意识到了 Android 权限机制的重要性，这对于保护用户隐私和数据安全至关重要。我现在对为什么需要这些权限以及如何在应用中妥善管理它们有了更深的认识。

在用户界面设计方面，使用 RecyclerView 和自定义适配器来展示视频列表的过程，不仅加强了我对这些控件的了解，也提高了我在数据展示方面的技能。实现视频文件的递归检索和展示，我学到了如何高效处理大量数据，这提高了我的编程技巧和对性能优化的认识。在项目开发过程中，我还学习到了如何妥善处理错误，并通过用户反馈来增强应用的稳定性和用户友好性。

这次实验不仅提升了我的技术能力，而且在解决实际问题和创新思维方面也给了我极大的锻炼。通过规划、设计、实现并测试一个完整的应用，我对 Android 应用的整体架构有了更全面的认识，对我的未来学习和职业发展产生积极的影响。