



《移动终端开发技术》 电子教案

第三单元 播放多媒体文件

所属专业（教研室）： 计算机软件技术

制定人： 陈媛媛

合作人：

制定时间： 2018年2月

日照职业技术学院



单元标题	播放多媒体文件	单元教学学时	2 课时
		在整体设计中的位置	第 32 次
授课班级		上课地点	一体化教室
上课时间	周 月 日第 节		
教学目标	能力目标	知识目标	素质目标
	能够在应用中播放音频和视频。	1、掌握在 Android 中播放音频的方法； 2、掌握在 Android 中播放视频的方法。	1、养成积极主动学习意识； 2、养成勤于动手的习惯。
教学重点、难点	教学重点：在应用中播放音频和视频 教学难点：在应用中播放音频和视频		
教学方法	采用反转课堂教学模式，课前学生学习微课了解知识点，课上采用教师引导、演示，学生分组练习、讨论等教学方法。 运用多媒体、AndroidStudio 开发环境、实训助手、教学平台等辅助授课。		
课前需掌握的知识点	1、MediaPlayer 类中一些较为常用的控制方法 setDataSource ()、prepare ()、start ()、pause ()、reset ()、seekTo ()、stop ()、release () 等的作用。 2、VideoView 类中一些较为常用的控制方法 setVideoPath ()、start ()、pause ()、resume ()、seekTo ()、isPlaying ()、getDuration () 等的作用。		
教学任务分解	任务一、播放音频 任务二、播放视频		
教学总结			



一、情景导入

手机上最常见的休闲方式毫无疑问就是听音乐和看电影了。而 Android 在播放音频和视频方面也是做了相当不错的支持，它提供了一套较为完整的 API，使得开发者可以很轻松地编写一个简易的音频或视频播放器。

二、课前测试

1、MediaPlayer 类中设置要播放的音频文件的位置使用的方法是（ ）。

- A. setDataSource () B. prepare ()
C. start () D. pause ()

答案 A

2、MediaPlayer 类中开始或继续播放音频使用的方法是（ ）。

- A. setDataSource () B. prepare ()
C. start () D. pause ()

答案 C

3、MediaPlayer 类中从指定的位置开始播放音频使用的方法是（ ）。

- A. setDataSource () B. prepare ()
C. start () D. seekTo ()

答案 D

4、VideoView 类中设置要播放的视频文件的位置使用的方法是（ ）。

- A. setVideoPath () B. resume ()
C. start () D. seekTo ()

答案 A

5、VideoView 类中将视频从头开始播放使用的方法是（ ）。

- A. setVideoPath () B. resume ()
C. start () D. seekTo ()

答案 B

三、知识讲解

1、MediaPlayer

老师引导,在 Android 中,MediaPlayer 是用于播放音频和视频的。MediaPlayer 对多种格式的音频文件提供了非常全面的控制方法,从而使得播放音乐的工作变得十分简单。

- setDataSource(): 设置要播放的音频文件的位置
- prepare(): 在开始播放之前调用这个方法完成准备工作
- start(): 开始或继续播放音频
- pause(): 暂停播放音频
- reset(): 将 MediaPlayer 对象重置到刚刚创建的状态
- seekTo(): 从指定位置开始播放音频
- stop(): 停止播放音频,调用该方法后 MediaPlayer 对象无法在播放音频



- `release()`: 释放掉与 `MediaPlayer` 对象相关的资源
- `isPlaying()`: 判断当前 `MediaPlayer` 是否正在播放音频
- `getDuration`: 获取载入的音频文件的时长

常用方法讲解完成后, 需要讲解使用 `MediaPlayer` 播放音频的步骤, 具体请参加教材。

2、VideoView

播放视频文件与播放音频文件类似, 与音频播放相比, 视频的播放需要使用视觉组件将影像展示出来。在 Android 中, 播放视频主要使用 `VideoView` 或者 `SurfaceView`, 其中 `VideoView` 组件播放视频最简单, 它将视频的显示和控制集于一身, 因此, 借助它就可以完成一个简易的视频播放器。

- `setVideoPath()`: 设置要播放的视频文件的位置
- `start()`: 开始或继续播放视频
- `pause()`: 暂停播放视频
- `resume()`: 将视频重头开始播放
- `seekTo()`: 从指定位置开始播放视频
- `isPlaying()`: 判断当前是否正在播放视频
- `getDuration()`: 获取载入的视频文件的时长

常用方法讲解完成, 然后讲解 `VideoView` 播放视频的步骤, 最后通过一个案例来演示视频播放。

任务一、播放音频

(1) 创建项目 `PlayAudioTest`

(2) 修改 `activity_main.xml` 中的代码

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
    <Button
        android:id="@+id/play"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Play" />
    <Button
        android:id="@+id/pause"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Pause" />
    <Button
        android:id="@+id/stop"
```



```
android:layout_width="0dp"  
android:layout_height="wrap_content"  
android:layout_weight="1"  
android:text="Stop" />  
</LinearLayout>
```

(3) 修改 MainActivity 中的代码

```
public class MainActivity extends Activity implements OnClickListener {  
    private Button play;  
    private Button pause;  
    private Button stop;  
    private MediaPlayer mediaPlayer = new MediaPlayer();  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        play = (Button) findViewById(R.id.play);  
        pause = (Button) findViewById(R.id.pause);  
        stop = (Button) findViewById(R.id.stop);  
        play.setOnClickListener(this);  
        pause.setOnClickListener(this);  
        stop.setOnClickListener(this);  
        initMediaPlayer(); // 初始化 MediaPlayer  
    }  
    private void initMediaPlayer() {  
        try {  
            File file = new File(Environment.getExternalStorageDirectory(),  
                "music.mp3");  
            mediaPlayer.setDataSource(file.getPath()); // 指定音频文件的路径  
            mediaPlayer.prepare(); // 让 MediaPlayer 进入到准备状态  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
    @Override  
    public void onClick(View v) {  
        switch (v.getId()) {  
            case R.id.play:  
                if (!mediaPlayer.isPlaying()) {  
                    mediaPlayer.start(); // 开始播放  
                }  
            break;  
            case R.id.pause:  
                if (mediaPlayer.isPlaying()) {  
                    mediaPlayer.pause(); // 暂停播放  
                }  
            break;  
        }  
    }  
}
```



```
    }  
    break;  
    case R.id.stop:  
        if (mediaPlayer.isPlaying()) {  
            mediaPlayer.reset(); // 停止播放  
            initMediaPlayer();  
        }  
        break;  
    default:  
        break;  
    }  
}  
  
@Override  
protected void onDestroy() {  
    super.onDestroy();  
    if (mediaPlayer != null) {  
        mediaPlayer.stop();  
        mediaPlayer.release();  
    }  
}
```

可以看到，在类初始化的时候我们就创建了一个 MediaPlayer 的实例，然后在 onCreate() 方法中调用了 initMediaPlayer() 方法为 MediaPlayer 对象进行初始化操作。在 initMediaPlayer() 方法中，首先是通过创建一个 File 对象来指定音频文件的路径，从这里可以看出，我们需要事先在 SD 卡的根目录下放置一个名为 music.mp3 的音频文件。后面依次调用了 setDataSource() 方法和 prepare() 方法为 MediaPlayer 做好了播放前的准备。

接下来我们看一下各个按钮的点击事件中的代码。当点击 Play 按钮时会进行判断，如果当前 MediaPlayer 没有正在播放音频，则调用 start() 方法开始播放。当点击 Pause 按钮时会判断，如果当前 MediaPlayer 正在播放音频，则调用 pause() 方法暂停播放。当点击 Stop 按钮时会判断，如果当前 MediaPlayer 正在播放音频，则调用 reset() 方法将 MediaPlayer 重置为刚刚创建的状态，然后重新调用一遍 initMediaPlayer() 方法。最后在 onDestroy() 方法中，我们还需要分别调用 stop() 和 release() 方法，将与 MediaPlayer 相关的资源释放掉。这样一个简易版的音乐播放器就完成了。

任务二、播放视频

- (1) 创建项目 PlayVideoTest
- (2) 修改 activity_main.xml 中的代码

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"
```



```
android:layout_width="match_parent"
android:layout_height="match_parent" >
    <LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content" >
    <Button
android:id="@+id/play"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="Play" />
    <Button
android:id="@+id/pause"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="Pause" />
    <Button
android:id="@+id/replay"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="Replay" />
</LinearLayout>
<VideoView
android:id="@+id/video_view"
android:layout_width="match_parent"
android:layout_height="wrap_content" />
</LinearLayout>
```

(3) 修改 MainActivity 中的代码

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener{
    private VideoView videoView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        videoView = (VideoView) findViewById(R.id.video_view);
        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button replay = (Button) findViewById(R.id.replay);
        play.setOnClickListener(this);
        pause.setOnClickListener(this);
        replay.setOnClickListener(this);
```



```
        if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(MainActivity.this, new
String[]{ Manifest.permission. WRITE_EXTERNAL_STORAGE }, 1);
        } else {
            initVideoPath(); // 初始化 MediaPlayer
        }
    }

    private void initVideoPath() {
        File file = new File(Environment.getExternalStorageDirectory(), "movie.mp4");
        videoView.setVideoPath(file.getPath()); // 指定视频文件的路径
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions,
int[] grantResults) {
        switch (requestCode) {
            case 1:
                if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                    initVideoPath();
                } else {
                    Toast.makeText(this, "拒绝权限将无法使用程序", Toast.LENGTH_SHORT).show();
                    finish();
                }
                break;
            default:
        }
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.play:
                if (!videoView.isPlaying()) {
                    videoView.start(); // 开始播放
                }
                break;
            case R.id.pause:
                if (videoView.isPlaying()) {
                    videoView.pause(); // 暂停播放
                }
                break;
        }
    }
}
```



```
case R.id.replay:
    if (videoView.isPlaying()) {
        videoView.resume(); // 重新播放
    }
    break;
}

@Override
protected void onDestroy() {
    super.onDestroy();
    if (videoView != null) {
        videoView.suspend();
    }
}
```

(4) 在 AndroidManifest.xml 中声明权限

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

四、演示作品

各小组演示汇报作品，教师与其他小组总结评价作品，各小组课后完善作品并提交到教学平台。

五、知识巩固

总结知识点，使用教学平台中的随堂练习题巩固本所学知识。

拓展
作业

在应用中播放音频和视频，并能够在手机上调试应用。

教学
后记