

Android 开发实验 5 实验报告

13331251 王嘉威

一.实验目的

- 1.学会使用 MediaPlayer;
- 2.学会简单的多线程编程,使用 Handle 更新 UI;
- 3.学会使用 Service 进行后台工作;
- 4.学会使用 Service 与 Activity 进行通信。

二.实验内容

实现一个简单的播放器,要求功能有:

- 1.播放、暂停功能;
- 2.进度条显示播放进度、拖动进度条改变进度功能;
- 3.后台播放功能;
- 4.停止功能;
- 5.退出功能。

三.实验步骤

1.编写主 Activity 的 xml 样式文件:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text=""
            android:id="@+id/state"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/time"/>
        <SeekBar
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/process"/>
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">
            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="PLAY/PAUSE"
                android:id="@+id/pp"/>
            <Button
                android:layout_width="wrap_content"
```

2.创建 MusicService 类,在内部创建播放器 MediaPlayer 及 Binder,在其构造函数初始化.

```
public class MusicService extends Service {
    public static MediaPlayer mp = new MediaPlayer();

    public MusicService() {
        try {
            mp.setDataSource("/data/Wonderful Rush.mp3");
            mp.prepare();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public class MyBinder extends Binder {
        MusicService getService() { return MusicService.this; }
    }

    public final IBinder binder = new MyBinder();

    @Override
    public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        return binder;
    }
}
```

3. 创建主 Activity 类,在 onCreate()中设置播放控件的事件监听. 绑定 MusicService.

```
private void bindServiceConnection() {
    Intent intent = new Intent(MainActivity.this, MusicService.class);
    bindService(intent, sc, this.BIND_AUTO_CREATE);
}
```

```
private ServiceConnection sc = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName componentName, IBinder iBinder) {
        musicService = ((MusicService.MyBinder)iBinder).getService();
    }

    @Override
    public void onServiceDisconnected(ComponentName componentName) { musicService = null; }
};
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    timeView = (TextView)findViewById(R.id.time);
    pp = (Button)findViewById(R.id.pp);
    stop = (Button)findViewById(R.id.stop);
    quit = (Button)findViewById(R.id.quit);
    bar = (SeekBar)findViewById(R.id.process);
    state = (TextView)findViewById(R.id.state);
    bindServiceConnction();

    bar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
            if (fromUser) {
                musicService.mp.seekTo(seekBar.getProgress());
            }
        }

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {

        }

        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {

        }
    });
}

```

4) 在 Activity 类中创建 handler 及线程以控制 UI 的变化.

```

private android.os.Handler handler = new android.os.Handler();

private Runnable myThread;

```

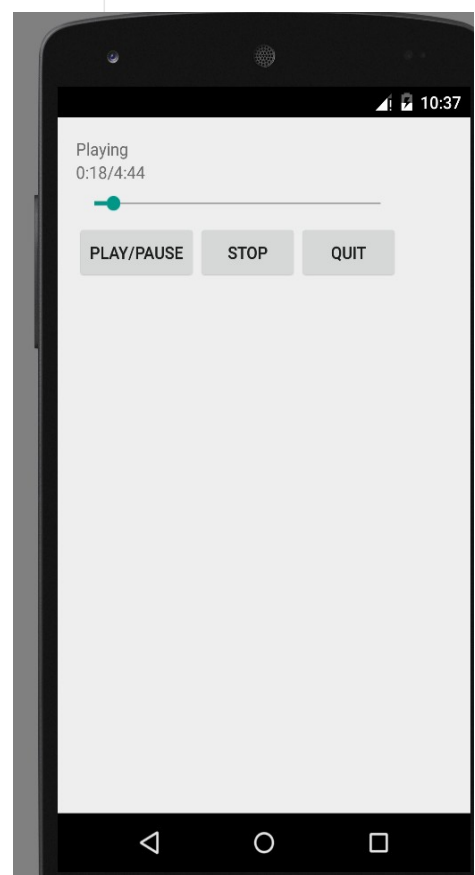
```

myThread = new Runnable() {
    public void run() {
        timeView.setText(time.format(musicService.mp.getCurrentPosition()) + "/" + time.format(musicService.mp.getDuration()));
        bar.setProgress(musicService.mp.getCurrentPosition());
        handler.postDelayed(myThread, 100);
    }
};

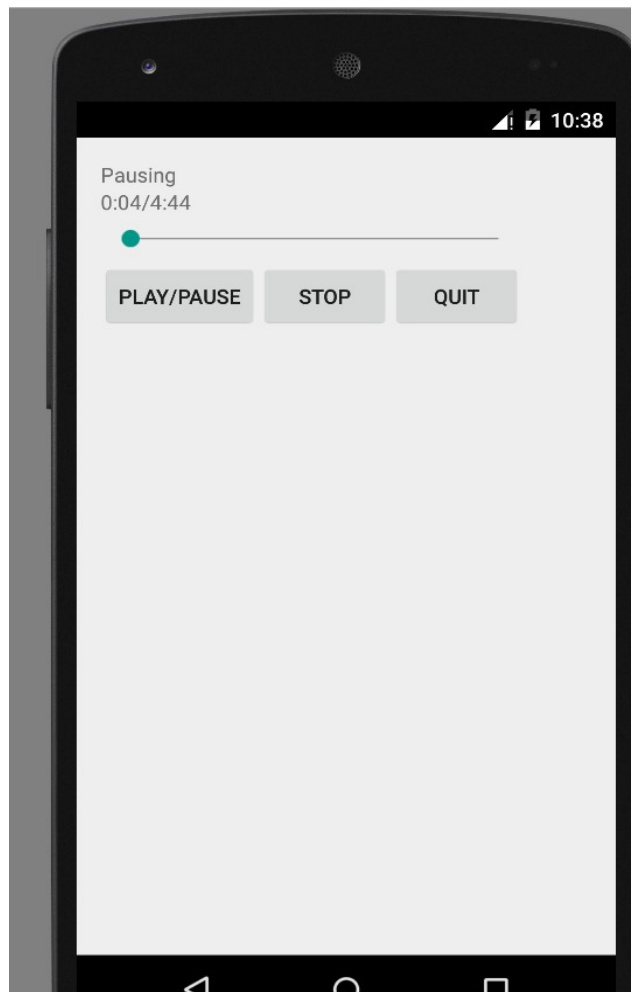
```

四.实验结果

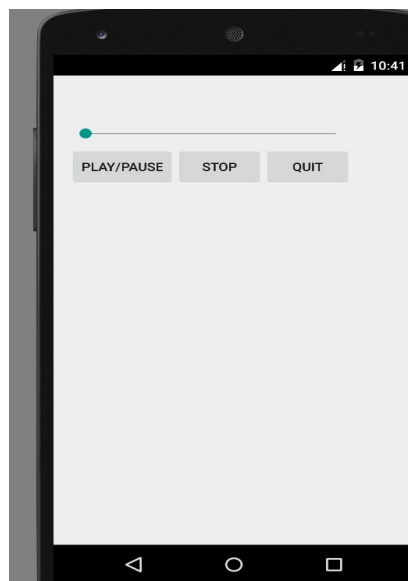
播放时:



暂停:



停止:



五.实验总结

这次实验遇到的问题不少,最大的一个是 MusicService 调用 stop()方法后,我按实验文档上做的调用了 prepare(),然而还是无法重新播放,我调试了程序,发现 prepare()后连 getDuration()都出错了,我上网查了下错误原因,说是 MusicService 没有 prepare(),可我明明调用了 prepare()方法,并且调试到这步还没错误产生,我十分纳闷,只好将 stop()改成了 pause(),再调用了 seekTo(0)代替,我会继续去寻找错误原因,若是解决了会成为我日后 android 开发的一个教训.

本次实验参考了以下网站:

深入 Android Player 的使用方法详解:

<http://www.jb51.net/article/38722.htm>

Android 的 MediaPlayer 架构介绍

<http://blog.csdn.net/eustoma/article/details/6706492>