

# 基于 MFC 的音乐播放器设计与实现

范力心 白俊鸽

四川大学锦城学院计算机与软件学院 四川 成都 610000

**【摘要】**音乐播放器一直是生活中不可或缺的一个软件,是播放音乐的媒介部分。一个好的音乐播放器,可以让自己享受到更好的音乐、放松心情。该播放器使用 VS2017 编写,包含打开文件、播放、停止、单曲循环、随机播放、歌单顺序播放等功能。打造一个仅仅属于自己的音乐播放器。

**【关键词】**音乐播放器; MFC; Visual Studio C++

## 1 背景

现代社会的生活节奏越来越快,各个职位不同人群受到的压力不容小觑,而社会生存压力<sup>[1]</sup>引发的各种心理疾病也层出不穷。在这样繁重的压力下,找到能够使人心情愉悦、平缓的途径就显得尤为重要,音乐疗法<sup>[2]</sup>是自古以来都十分有效的方法。

## 2 播放器内容

本音乐播放器主要用于 PC 端本地音乐播放,针对个人单机使用。旨在为个人用户提供无网络状态下播放音乐,更改音乐播放模式,以及按照个人喜好设置播放背景,软件图标,让用户拥有更好的听歌氛围。

## 3 播放器界面及功能设计

### 3.1 总体设计

根据方便简洁的原则,笔者将音乐播放器的所有功能都设计在一个界面上,划分四个不同的功能区域,分别为:文件选择、播放设置、音量控制和歌单列表。四个功能区域都有提示角标,起到引导用户使用的作用。

### 3.2 文件选择区域设计

处在界面的顶部,由文字提示、文本显示框以及打开文件按钮这三个部分组成,点击[打开]按钮,会弹出一个文件对话框,可以选择想要添加到软件中的音乐文件,选中单个音乐文件之后,点击对话框右下角打开,文本显示框中会出现当前选中的音乐名。

### 3.3 播放设置区域设计

该部分总共包含五个按钮和一个下拉选择框。点击

[播放]按钮,会开始播放,文件选择区域文本显示框中显示的那首歌曲;点击[暂停]按钮,歌曲会在当前播放位置停下,按钮上暂停文字改变为继续,点击已经变为[继续]的按钮,歌曲会从之前暂停的位置继续开始播放,按钮上的文字变化为暂停;点击[停止]按钮,歌曲会停止播放,并且文件选择区域的文本显示框中内容将被清空,此时再次点击[播放]按钮,将不会有任何歌曲播放;点击[上一首]按钮,播放歌单中前一首歌曲,若当前歌曲位于歌单第一行,那么弹出当前是第一首歌曲;点击[下一首]按钮,播放歌单中后一首歌曲,若当前歌曲位于歌单最后一行,那么会弹出当前是最后一首歌曲;点击下拉框,会有三个选项,分别单曲循环,随机播放,顺序播放,选择[单曲循环]总是播放当前选定的歌曲,选择[随机播放]将随机播放列表歌曲,选择[顺序播放]将会从当前位置依次播放列表余下歌曲。

### 3.4 音量控制区域设计

这一区域由“音量控制”提示文字、文本框和滚动条组成。文本框中显示的数字随滑块移动而变化,滑块的移动控制播放器音量大小;光标选中文本框,输入 0-100 的整数可以同步控制滑块的位置,以及播放器的音量。

### 3.5 歌单列表区域设计

歌单列表部分包含提示角标文字,以及导入播放器的音乐文件名字组成的播放列表。当打开音乐时,播放列表会增加该音乐,当从外部打开文件夹拖动音乐文件到播放器界面上,播放列表也会增加该文件的名称,并且播放列表中的所有音乐,单击或者双击都可实现立即播放和覆盖播放。所谓覆盖播放就是停止当前播放的音乐,开始播放点击的音乐文件。

作者简介:1.范力心(1999-),女,汉族,四川自贡人,本科,四川大学锦城学院,研究方向:C++程序设计。

2.白俊鸽(指导老师)(1980-),女,汉族,四川成都市人,硕士,讲师,四川大学锦城学院,研究方向:网络安全。

## 4 功能实现

### 4.1 软件开发环境

该软件使用 VS2017 进行编写开发, VS2017 是一款功能强大的编程工具, 能够胜任各种规模和需求的工程。它的全称是 Visual Studio 2017, 可以在 Windows 和 Mac 系统环境中运行。由于使用音乐播放器的人群肯定不全是 IT 行业的从业者, 所以电脑中极有可能没有安装 VS, 并且完全不了解 VS 的使用方法, 因此就需要有一个软件的用户界面程序, 里面必须包含播放器所有功能。界面的封装通常使用 QT 或者 MFC 实现, 在这个软件开发中, 笔者选择使用 MFC 框架结构<sup>[3]</sup>。MFC 全称 Microsoft Foundation Classes, 它以 C++ 类的形式封装了 Windows API 接口。在这个音乐播放器的开发中, 重新自己设计媒体播放器代价较高, 因此笔者在这里直接使用了 MCI 媒体控制接口<sup>[4]</sup>进行播放等控制。

不同区域的划分使用了 MFC 中的 Group Box 控件, 其提示角标的文字设置通过更改 Caption 中的值, 进而完成设定。另外, 关于播放器的背景图片, 笔者在这里运用的方法及其简单。进入项目资源文件夹下, 找到 \*.bmp 类型图片文件, 然后在本地图片文件中选择任意一张自己喜欢的图片, 编辑图片格式为 .bmp, 然后覆盖之前找到的 \*.bmp 文件, 再进入 VS 资源视图打开项目的 Bitmap 文件夹, 对其图片文件进行属性中的 ID 设置, 设置完成就能使用其自带函数 LoadBitmap() 进行图片的加载。关于设置背景图片模糊问题也十分容易解决, 在这里笔者运用的 Windows GDI<sup>[5]</sup> 函数 -SetStretchBltMode() 进行图片拉伸算法设置, 使用 HALFTONE 模式虽然运行的速度相较其他模式更慢, 但经过处理的图片清晰度会得到明显改善。

### 4.2 文件选择区域功能实现

该区域的 Group Box 属性的 Caption 值设置为 [文件选择]。在点击 [打开] 按钮后弹出的文件对话框使用 CFileDialog 类设置一个对象, 通过它调用 DoModal() 函数完成弹出文件对话框动作。此时的程序还需要使用 if(IDCANCEL == \*.DoModal()) 语句来进行状态判定控制, 避免一直重复出现选择文件提示。在打开文件之后, 使用 dlg 对象的 GetPathName() 函数去获取选择文件的路径信息, 并且将其保存在一个 CString 类型的变量中, 这里也可以使用 GetFileName() 函数去获取文件名, 这两个函数区别是第一个包含全部路径以及带后缀的文件名, 第二个只有带扩展名的文件名, 如: test.txt。然后在得到文件路径信息之后, 可以使用 SetDlgItemText() 将获得的字符串显示到该区域的文本显示框中。拖动文件实现打开音乐功能, 与打开按钮的实现过程差不多, 只是

需要将弹出文本框改为获取拖拽文件的路径, 设置一个 wchar\_t 类型的变量 szMusicPath[MAX\_PATH] 来存放文件路径, 然后使用 DragQueryFile() 函数打开文件。

此时我们已经打开一首歌曲了, 但是系统并不知道我们现在已经打开歌曲文件了, 所以我们在类中设置一个 MCIDEVICEID 类型的变量 m\_DeviceID 来记录当前歌曲的 id。当然在这里打开音乐需要使用 MCI 接口的 mciSendCommand() 接口发送请求, 请求成功后才算真正打开了音乐文件, 音乐文件在系统中有挂载。

### 4.3 播放设置区域功能实现

这部分的按钮较多, 还有一个下拉选框。首先播放按钮是调用的 mciSendCommand() 函数, 将挂载的音乐文件设置为播放状态, 由参数 MCI\_PLAY 控制, 并且歌曲的播放起始位置设定为 0, 就会从头开始播放。暂停 / 播放按钮通过调用 GetDlgItemText() 函数来获取当前按钮上的文字信息, 保存到变量 str 中, 然后用 if (str == L" 暂停 ") 和 else if (str == L" 继续 ") 进行判定, 决定调用 mciSendCommand() 中的参数是 MCI\_PAUSE, 还是 MCI\_RESUME。停止按钮部分就更简单了, 也是调用 mciSendCommand(), MCI\_STOP 停止播放和 MCI\_CLOSE 关闭文件。上一首和下一首按钮类似, 主要是通过设置一个类对象, 然后通过 GetText() 函数和 SelectString() 函数进行获取指定行字符串和选中指定行的操作, 以及 GetCount() 函数和 GetCurSel() 函数获取列表总行数和当前选中列表项的行号。

下拉选框是在定时器的回调函数中实现的, 为该控件设置一个消息响应函数, 然后在函数中使用 SetTimer() 定时器函数, 然后在回调函数中检查音乐播放进度, 当音乐播放结束时, 根据选中的播放模式调用函数, 单曲模式直接调用播放按钮函数, 顺序模式调用下一首函数, 随机播放需要写一个随机函数, 通过 rand() 函数产生在列表总行数内的随机整数, 进而进行随机音乐播放。

### 4.4 音量控制区域功能实现

音量控制使用的 Slider Control 滑块控件以及文本编辑框实现。为滑块控件添加消息响应函数, 在对滑块进行初始化之后, 在该函数中设置一个 MCI\_DGV\_SETAUDIO\_PARMS 的变量 mciSetVolume, 然后通过 mciSetVolume.dwItem = MCI\_DGV\_SETAUDIO\_VOLUME 语句对系统音量媒体进行使用, 并且通过 mciSendCommand() 进行音量设置, 该函数此时的接口参数有 MCI\_SETAUDIO、MCI\_DGV\_SETAUDIO\_VALUE 和 MCI\_DGV\_SETAUDIO\_ITEM。关于文本编辑框中显示内容的动态变化, 笔者在这里为该控件设置了一个变量 m\_volume, 然后使用 m\_volume = m\_slider.GetPos() 语句将其与滑块的位置绑定在

一起, 这样编辑框中的值, 就会随着滑块的滑动位置而动态变化。

#### 4.5 歌单列表区域功能实现

歌单列表中是导入播放器的音乐文件路径信息, 通过 Listbox Control 控件显示。这部分功能尤为简洁, 只有单击选中列表项, 然后通过 SetDlgItemText() 将选中项的字符串写到打开文件区域的文本框中, 并且使用 mciSendCommand() 发送打开文件请求, 再对播放按钮函数进行调用。当然, 在对文本编辑框进行字符串写入时, 需要先对其原有内容进行清除操作, 也就是编辑框内容置空, 这里直接使用 SetDlgItemText(IDC\_filename, L"") 语句进行操作。同样的, 在调用播放按钮函数的时候, 要事先关闭当前正在播放中的音乐文件。

#### 5 结束语

该音乐播放器虽然能够被其他用户下载到个人电脑上作为音乐播放媒介使用, 且软件操作简单明了, 但是

只能对已经下载到本地的音乐文件进行播放, 另外对于当前播放音乐的歌词滚动部分也还未能实现。从用户使用的体验感这方面来看, 播放器还有很多值得改善的地方, 也还缺少了许多交互功能。再往后的软件开发设计中, 尽量从用户的角度出发, 不断完善自己的软件, 努力做到功能详尽、实用。

#### 【参考文献】

- [1] 冯莉. 个体化时代城市青年的社会压力及其应对 [J]. 中国青年研究, 2014(02).
- [2] 孙耶宾. 音乐疗法与健康教育思想的相融合研究——评《音乐与健康》[J]. 中国食用菌, 2020(03).
- [3] 刘红, 校云芳. MFC 框架程序剖析 [J]. 电脑知识与技术, 2012(12).
- [4] 乔建魁. 媒体控制接口 (MCI) 编程应用 [J]. 电脑编程技巧与维护, 2005(11).
- [5] 张莹. 基于 GDI 开发 windows 动态绘图优化解决方案 [J]. 硅谷, 2011(14).