

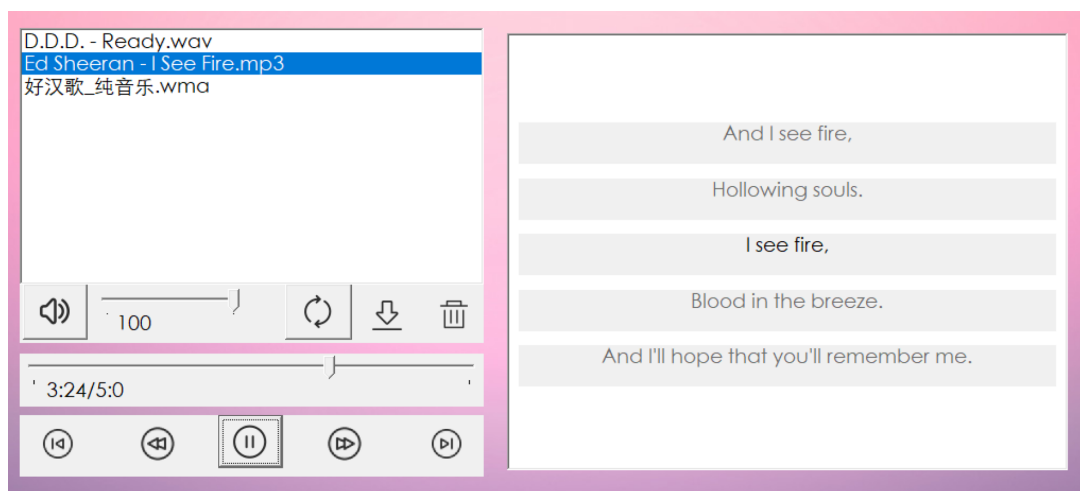
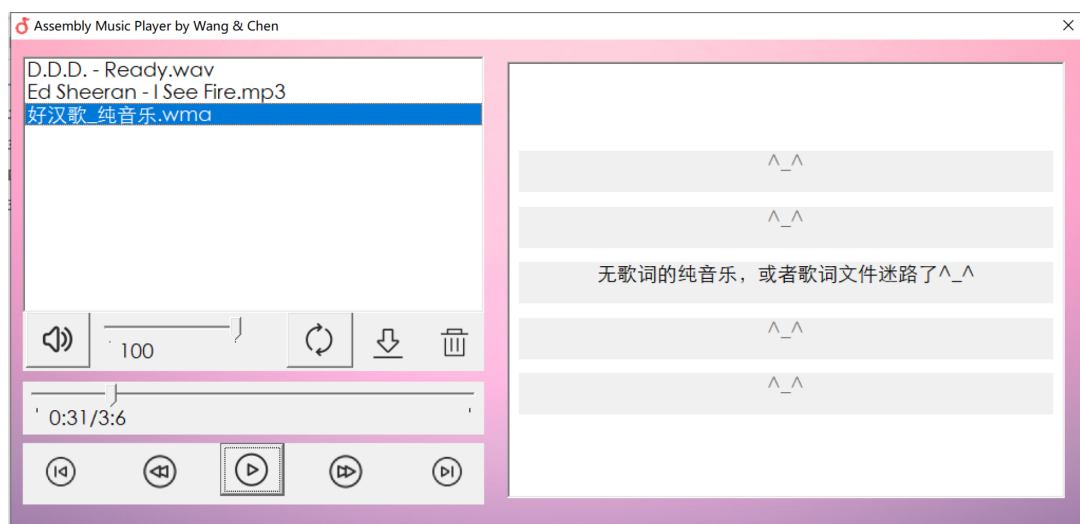
汇编小组作业-5音乐播放器

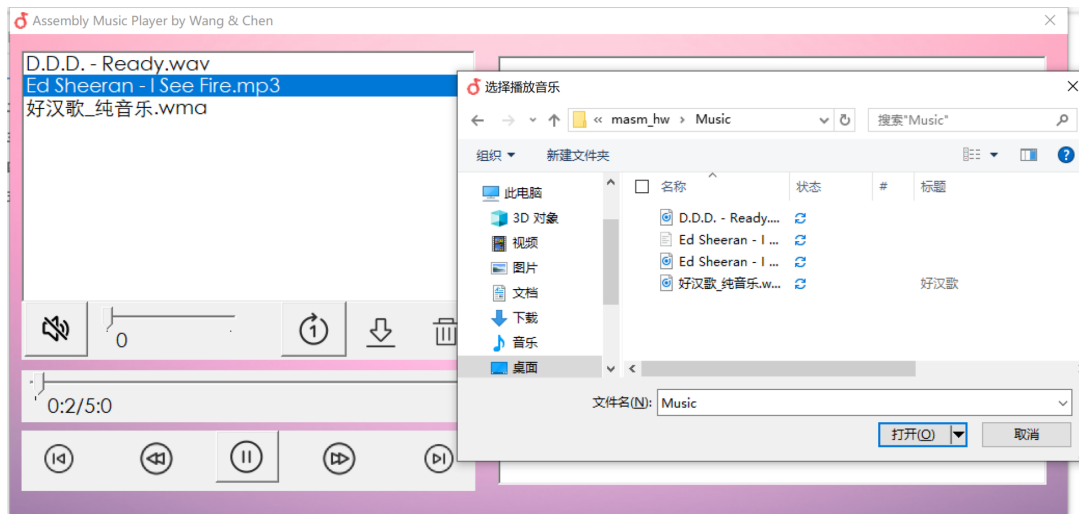
2016010539 王世杰 2017013631 陈语凝

一、项目简介

本次音乐播放器的MASM汇编开发，采用模块化与函数化的设计思路，实现了以下功能的实现（示意图见下方）：

1. mp3、wma、wav等文件格式的解析播放；
2. 音量控制，包括滑块控制与静音按钮；
3. 播放控制，包括播放与暂停、进度条、前进与后退5秒按钮；
4. 歌曲目录的管理，包括本地导入、列表删除、点选播放、按列表顺序前后切歌等；
5. 列表播放顺序的设置，包括列表顺序、单曲循环、列表随机三种；
6. 滚动歌词的实现，包括通用.lrc格式文件的解析呈现、无歌词音乐显示设置、歌词滚动播放等。





二、开发环境

1. 操作系统：Windows 10
2. 开发语言：基于x86的汇编语言
3. 汇编器：MASM32_v11.0
4. 开发软件：WinAsm（包括其图形界面辅助编辑器）
5. 依赖库：详见.asm文件头部

三、实现原理

- 通过windows32 API进行桌面程序编写

通过查询微软官方文档，对Windows API进行调用，实现相关功能。

- 依赖windows窗口语句进行函数间信息传递与控件控制

初始化完成后即进入每0.02秒进行一次的事件循环，检测到收到的消息在窗口呈现出来，包括更改控件图片、进度条位置调整，歌词刷新等。

- 同一逻辑的多种控制方式实现

进度条等窗口控件与实际逻辑的绑定，可以使多种调整方式实现效果相同，同时添加新的调整方式也更加容易。

- 播放列表的记录

通过在本地储存播放列表文件，并在打开软件时自动读取，实现了记忆性的播放列表播，也顺利实现了重启恢复功能，用户友好性高。

- GUI的实现

为了提高开发效率，让软件界面更加美观，我们使用rc文件对项目资源和UI模板进行了设置，通过资源模板文件和汇编代码的结合，我们实现了逻辑与界面分离，有利于模块化开发。

四、难点与创新点

- 歌词文件的解析、显示、滚动播放

不同的.lrc文件格式略有不同，但都是“([时间]内容)*”的格式，我们字符串匹配查询到“[”的位置，维护每一句歌词的位置，在循环查询中对比实际位置所匹配的那一句歌词以及前后各两句歌词，实时显示，便有了“滚动”的视觉效果。

.lrc文件的编码格式各异，UTF-8等也很多，但WinAsm的解析功能仅支持GB2312，特此说明，使用其他会有乱码出现。

- 使用按钮与进度条进行播放进程控制

可以通过进度条调整播放进度，其主要原理是，在拖动进度条放手后，确定进度条的位置，随后根据进度条的位置发送MCI命令控制播放进度，随后根据播放进度设置时间格式等。

得益于模块化设计，控制只需改变其中一部分变量，即可实现全局控制。在事件循环中，不断检测发来的消息，当接收到WM_HSCROLL类型的消息，判断是音量条还是进度条，在循环检测中发送信息修改相关数据，用户可以以多种方式接收到变化。

- 尽可能模块化、函数化开发

之前的汇编小作业相对简单，熟练掌握基础语法、寄存器管理基本都能完成，而本项目在功能很少时即一度多达1900行代码，后续调试与功能添加难度颇大，模块化与函数化后，清晰很多，可复用性、可移植性明显增强；信息模块与逻辑模块分离后，理解难度明显降低。

五、具体分工

项目考虑两人协作无需过多版本控制，合并均为手动，未使用Git等版本控制软件。具体分工为：

	学习与知识储备阶段	主要实现阶段	迭代与总结阶段
王世杰	1.win32窗口api的学习、整理与总结 2.VisualStudio2017调试环境配置 3.WinAsm软件初探，开发环境比较 4.C语言demo实现	1.音乐文件格式解析 2.进度条的多种控制方式维护 3.不同循环方式播放的实现 4.GUI卡停不同问题的调试与解决	1.实现播放前进与后退一段 2.模块化、对象化检查与完善 3.鲁棒性测试与增强
陈语凝	1.WinAsm资源文件 (.rc) 学习与使用 2.程序所需美术材料的收集与处理 3.C语言demo实现	1.文件目录访问、文件加载、列表恢复实现 2.进度条与音量条与逻辑绑定 3.滑条与文字显示绑定 4.歌词文件读取、解析与显示	1.歌词滚动播放，可使用编码模式测试与标注 2.实现原理概述 3.鲁棒性测试与增强

六、体验、学习与感想

此次大作业与计网FTP撞车（要感谢助教与老师修改DDL，收获大了很多！），时间相对有限，学习资料较少，但使用WinAsm对理解与使用win32API帮助很大，同时也体验了汇编代

码的模块化、函数化全局设置可以有效减少代码量，降低调试与功能迭代的难度。

由于masm的资料不是特别多，因此我们的很多功能都是通过先查阅WINDOWS API的C语言函数，在C语言版本下实现后，手动翻译成汇编代码，经过这样的过程，我们发现得到的代码可读性和逻辑性都比较强。

总体来说，通过这一次的汇编大作业，我们对于汇编了解更加深刻，同时也对于Windows程序开发的理解更加深入。

七、参考资料

《汇编语言 基于x86处理器 第7版》

《Intel汇编语言程序设计》

win32 API 大全

<http://www.esk365.com/zxsc/prg/winapia/>

WinAsm相关资料

<https://www.jb51.net/softjc/191426.html>

一些思路优秀的Github 汇编项目

<https://github.com/chenyijie2016/MusicPlayer-ASM>

<https://github.com/ThssSE/MusicPlayer#下载功能的实现>

<https://github.com/pieceoffolly/EMP>