Christopher Kruger, Christopher Youngclaus, Mark Tuhacek
Prof. Yousef
CPE 553
28 October 2022

Project Proposal - **MP3 Player Interface**

Introduction

Our group shares a common interest in music, and we each have experience in either creating music, mixing music, or enjoying local mp3 playback. Since major streaming platforms protect their music, our program would run solely off of local files on one's computer, but would allow one to do everything a normal media player would do. Our goal is to learn about C++ APIs through the integration of local files into our programs, and have a better understanding of what goes on behind the scenes of major streaming platforms at a basic level.

Materials Needed

Creating a GUI will most likely be required for this project, so a software such as Qt will be needed for this task. The mainwindow.h header file allows Qt to interact with the users program and create the constraints for the actual GUI popup window. Also required will be winmm or windows.h, which are the windows multimedia header files. These are the import libraries for multimedia API. For things such as volume control up/down, the API can be used as an interface. Other than that the project will be compiled and run using an IDE such as Visual Studio, and the group will collaborate through a shared repo in GitHub where working copy of the source code will be located.

Expectation and Objective

With the target audience being an average layman, we hope to incorporate a simple yet intuitive GUI for opening audio files. The workflow of our project for the end user is to be able to open an audio file from an aesthetically-pleasing window, and be able to manipulate the file for playback by having an integrated play/pause button, a scrubber, and volume control. The team's expectation is to have a zero-hassle program which is simple to use but is responsive to the user controlling it. The team has used countless audio players bloated with features to the point that simply playing an audio file becomes cluttered with delays from background processes.

Quality Control/Demonstration

To track our progress, the group will prioritize features in the following order: the creation of a window for the user, a GUI including aptly-named buttons, a prompt for users to manually select a specific audio file within their system, a play/pause button, a scrubber, and volume control. Special end goals, given enough time, include the ability to add effects on top of songs (such as a basic reverb function, pitch shifting, etc) or to make a queue of songs to play by incorporating a suitable data structure.

For a demonstration of the project once completed, the team will show the functions to select a song, show playback features, and manipulate the volume. The team will also compare the program to the built in Windows media application to compare both speed, aesthetics, and intuition. A high quality project is the team's main concern, and quality control of the program includes, but is not limited to: maximizing audio fidelity, minimizing features considered "bloat", and reacting promptly to the user's button inputs.