



Cross-Platform Application Development in C++

Mp3 Player



NAME: M.G.A GEETHANJANA

Overview of the Program Mp3 Player

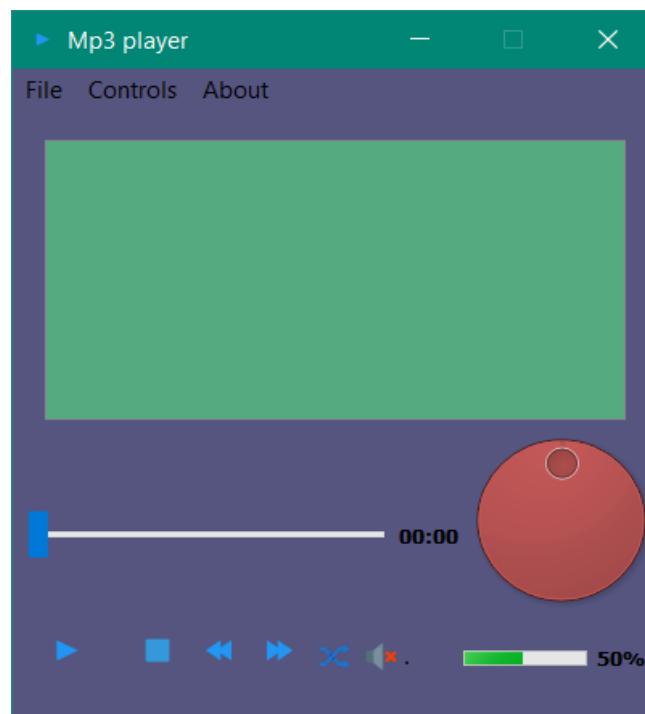
Aim

The main purpose of this project is to develop a basic cross-platform Mp3 player that works. A simple cross-platform Mp3 player with a playlist that has functionalities like Play, Pause, Stop, Next, Previous, Shuffle, and mute using C++ (QT Framework). This program uses the Qt Creator. Qt is a C++ set of tools for the creation of cross-platform applications.

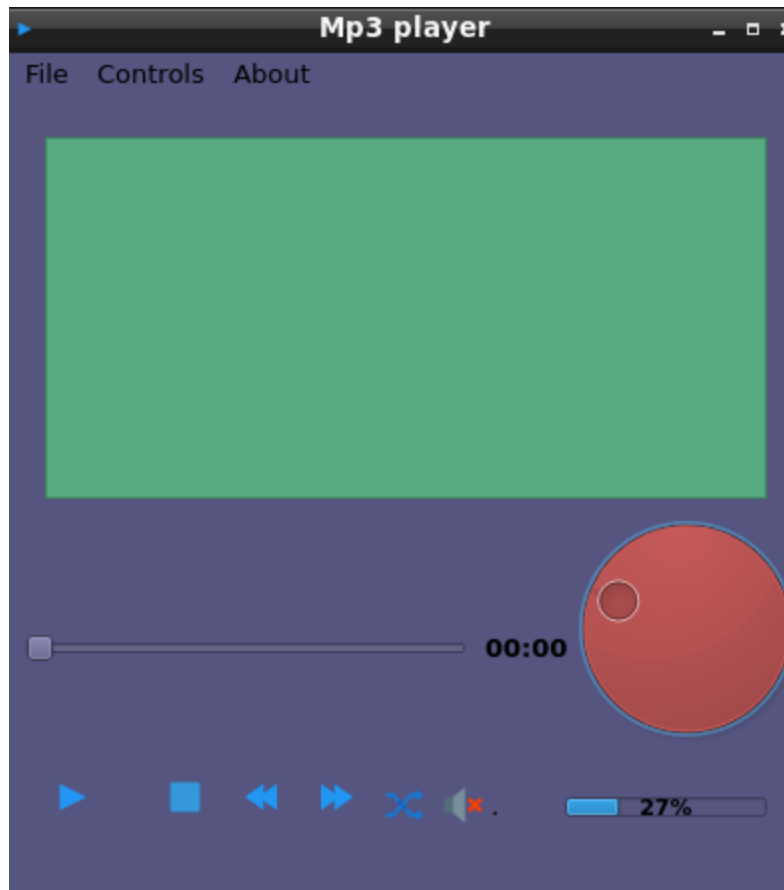
Scope

The Mp3 player supports all audio formats: mp3, ALAC, MIDI, FLAC, APE, and WAV, etc. with all forms of audio formats, audio plays them with greater precision. This Mp3 player has MUTE, STOP, PLAY, PAUSE, and VOLUM methods to command for audios. This one has a Folder OPEN option which can quickly browse and easily identify any favorite music. The main objective is to design a simple Mp3 player with a user-friendly GUI and basic and some advanced functionalities of the Mp3 player.

GUI Design



Windows 10



Ubuntu 20.04.1 LTS

There are several colorful controls on this Mp3 player. (rewind, play, interrupt, pause, forward, repeat, shuffle)And I am using The primary color is blue. You can see the Draggable Search Line above the keys,

From the right hand, click the mute button and the up and down volume Knob on the right side. For open a file, there seems to be an action button at the end.

Functionality

- Open single or multiple songs and play them.

This can be done either using the file button in the toolbar or just simply pressing Ctrl+O as the shortcut. And the current status initialized in a label under the volume bar.

- Play or pause the song on user preference

Once the playlist is loaded the playlist will be played automatically. Users can pause it by either clicking the play button on the toolbar or clicking the GUI button placed at bottom of the window or pressing the Space button. The current status is initialized in a label under the volume bar.

- Stop the song playing

Users can stop it by either clicking the Stop button on the toolbar or clicking the GUI button placed at bottom of the window or pressing the ESC button. The current status is initialized in a label under the volume bar.

- Play next song

Users can do this by either clicking the next button on the toolbar or clicking the GUI button placed at bottom of the window or pressing the PgUp button. The current status is initialized in a label under the volume bar.

- Play the previous song

Users can do this by either clicking the previous button on the toolbar or clicking the GUI button placed at bottom of the window or pressing the PgDn button. The current status is initialized in a label under the volume bar.

- Shuffle the playlist

Users can pause it by either clicking the shuffle button on the toolbar or clicking the GUI button placed under the playlist of the window or pressing the Ctrl+Tab button. And the current status initialized in a label under the volume bar.

- Mute song

Users can mute it by either clicking the mute button on the toolbar or clicking the GUI button placed under the playlist of the window or pressing the M button.

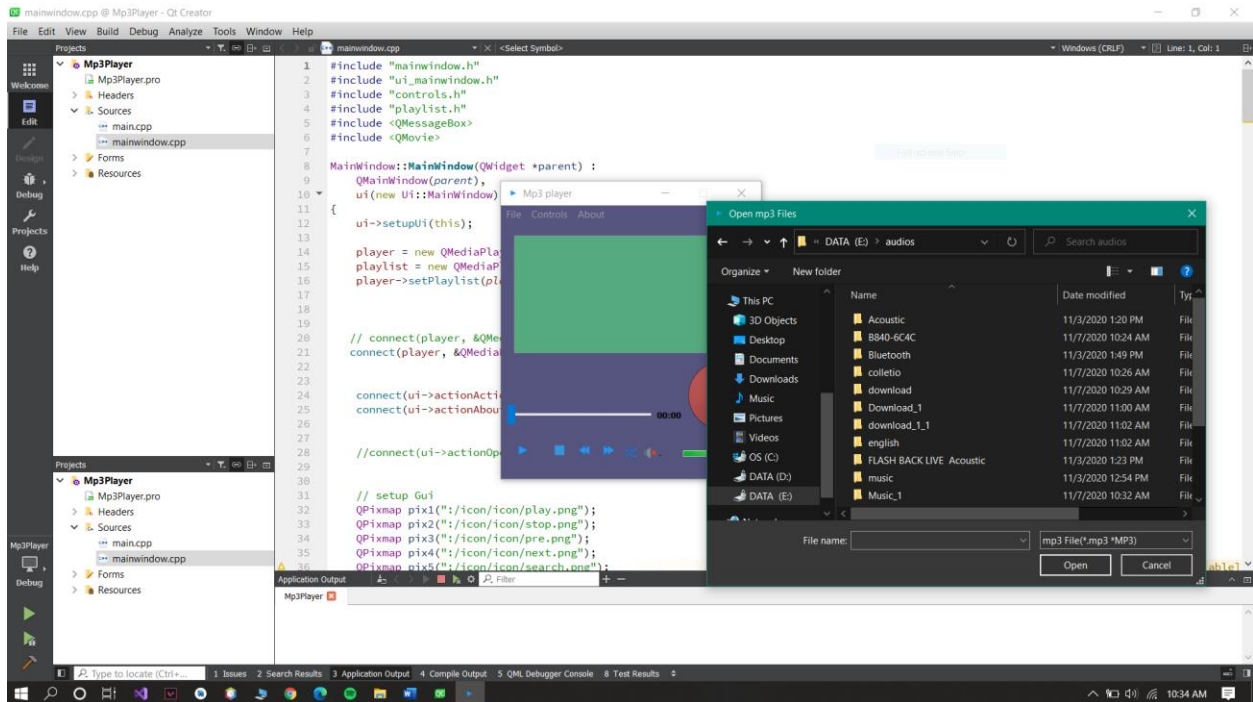
- Change volume

Use can change the volume by using the Dial widget placed under the playlist and can see the progress and current volume level on the progress bar under it.

Shortcuts I have also applied shortcuts to certain alternatives which I mentioned earlier.



Figure 1 - Shortcuts



working with files

Known Issues

When you drag and drop an audio file, we need to manually pick an audio file if it does not work from the choice I offered.

When playing the song, I tried to display the Album art of the song in the player if available, or otherwise play an equalizer gif to give more attractiveness to a player, for some reason it's not working, and I couldn't resolve that

When loading the songs to the playlist it shows the full file path instead of the file name. This is because I used QStringList to store the file paths, but to get the file name it can be easily done by using QString and QFile::FileInfo::fileName(); but it does not support for QStringList. I have tried many methods to do that, but it would not work. I have commented on those methods I tried in on_actionOpen_triggered().

Supported Platforms

This Mp3 player running without even any difficulties on the following platforms.

- Ubuntu 20.04.1 LTS
- Windows 10

The screenshots below display the software going.

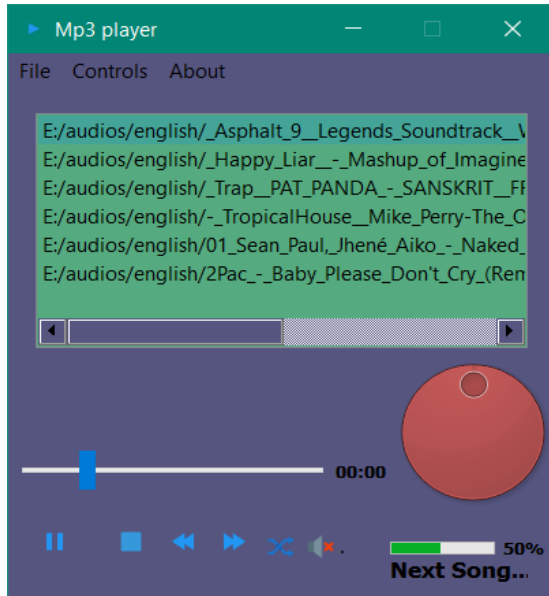


Figure 2 - Windows 10

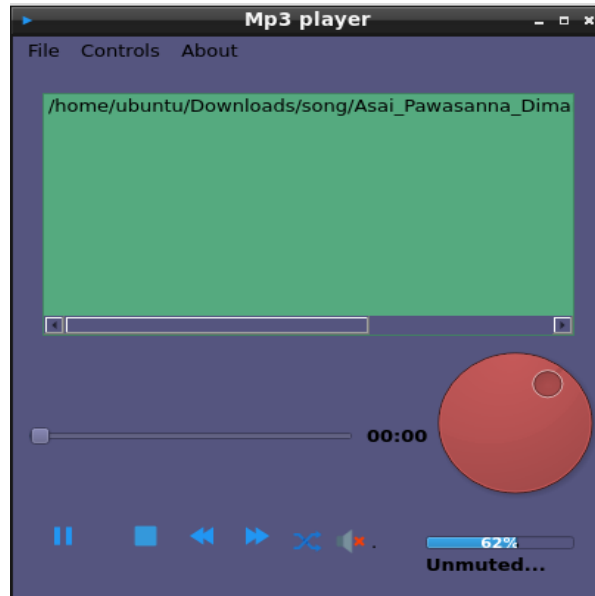


Figure 3 - Ubuntu 20.04.1 LTS

On these two systems, Mp3 Player checked and performed 100 percent without any problem. The pictures below show the app's operating process on both systems.

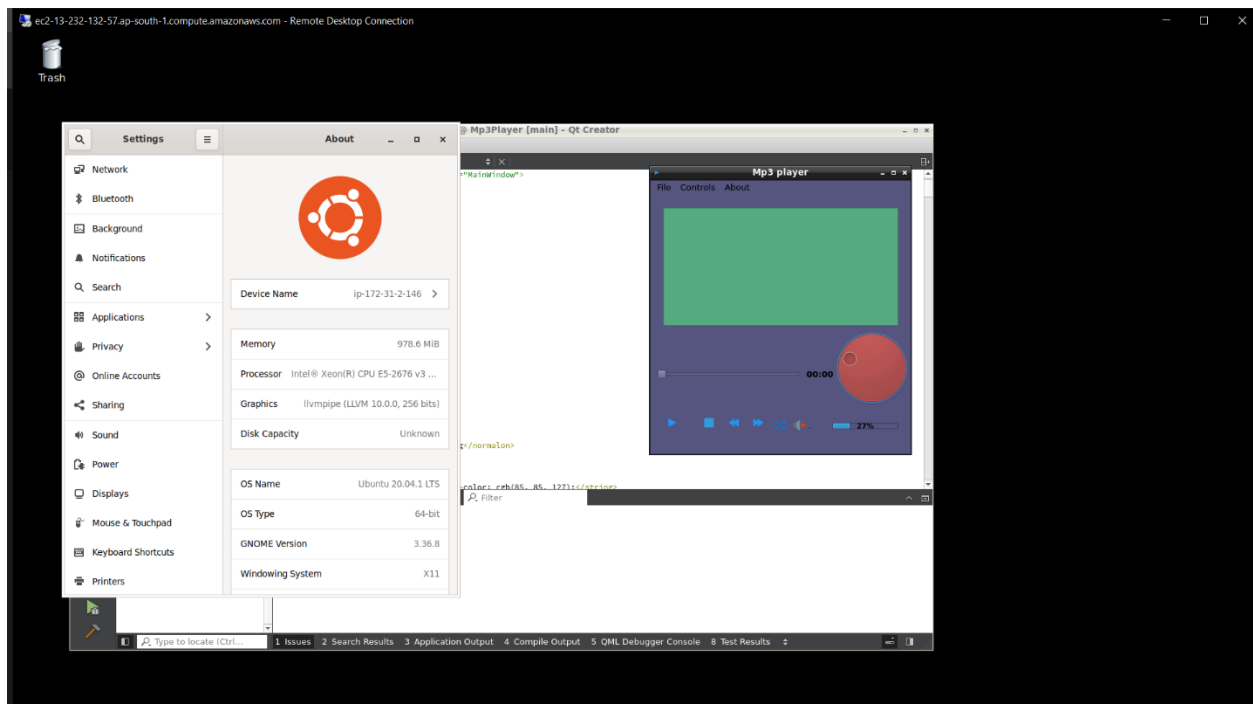
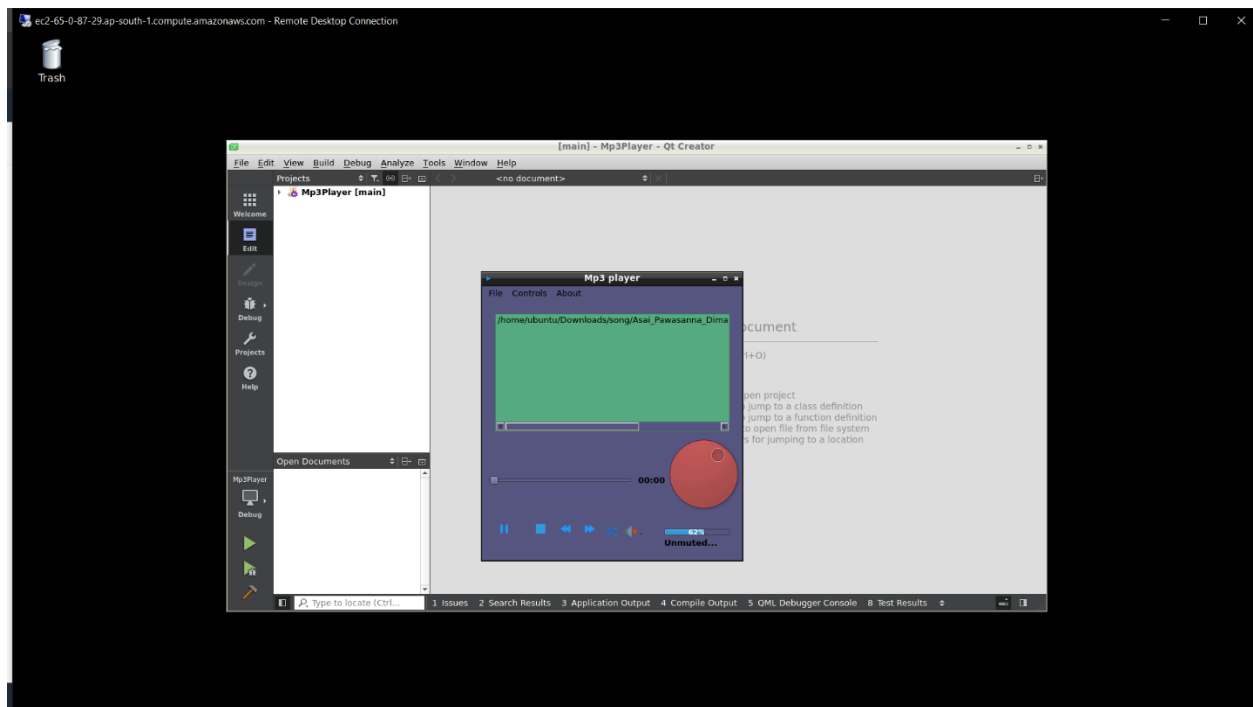


Figure 4 - Ubuntu

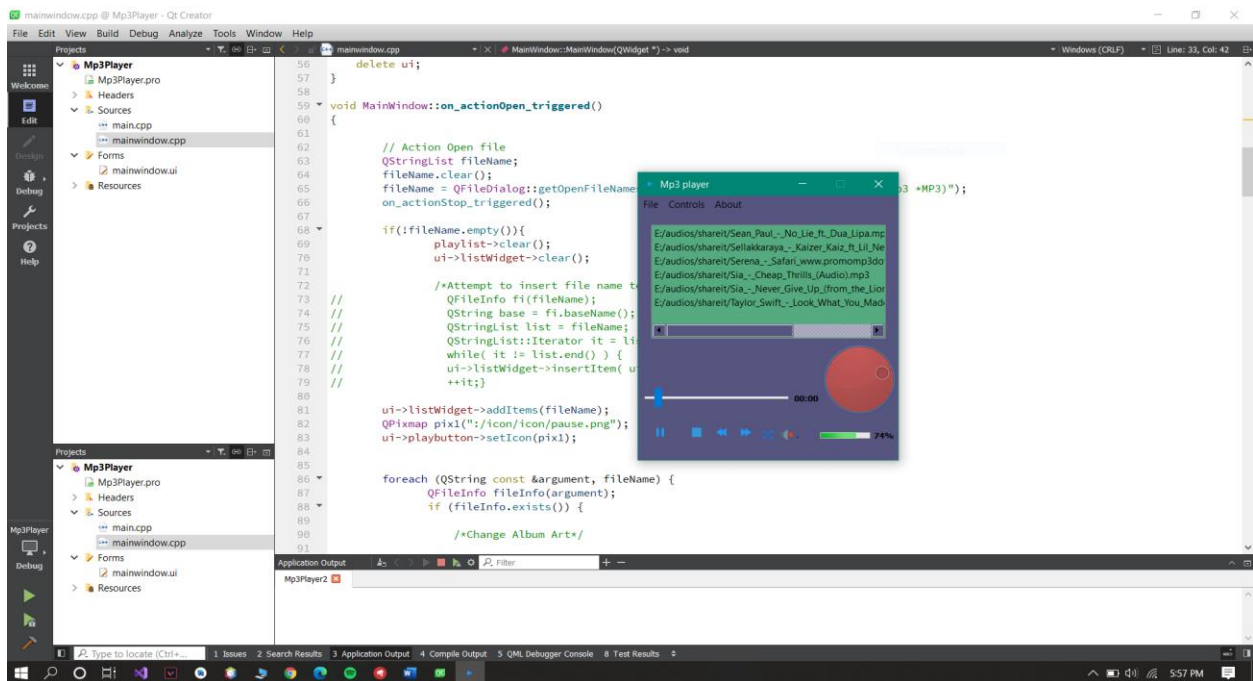
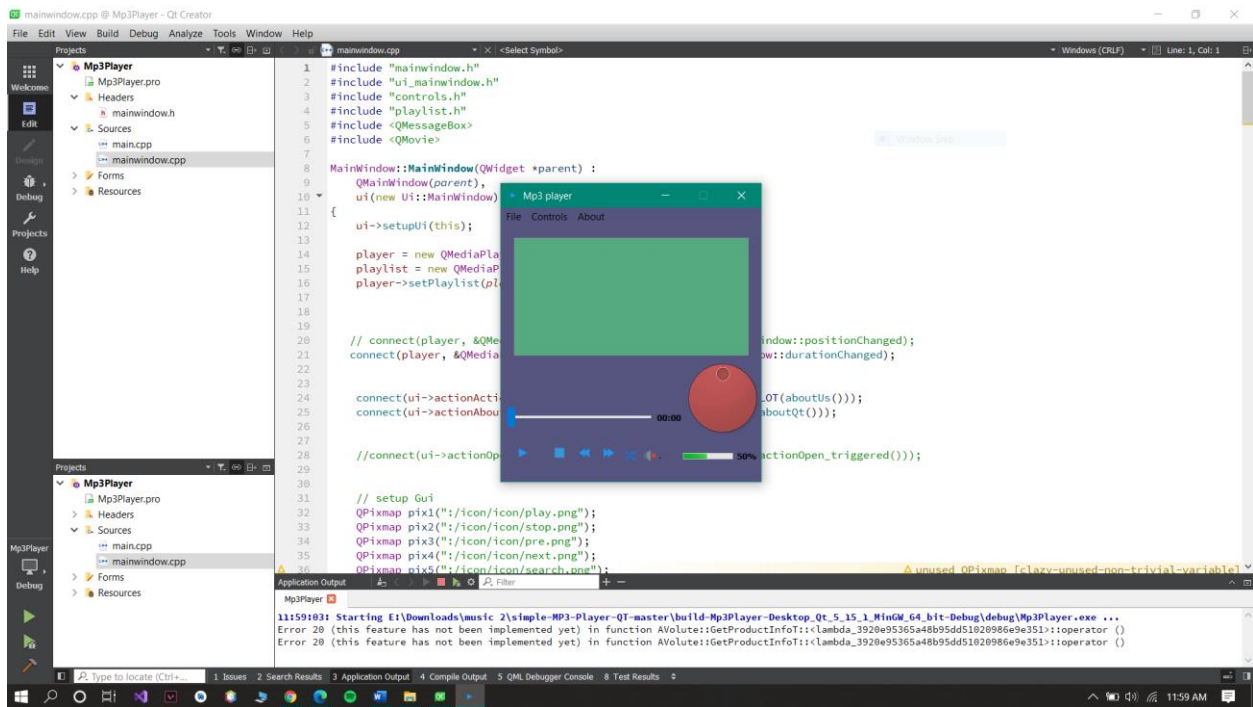


Figure 5 - Windows 10

Thank You.