CS 335 Exercise 10: Audio Player Metadata

1 Introduction

In this assignment you will extend the audio player project to extract and display artist and title metadata.

2 Preparing the Class Repository

2.1 Update the Master Branch

Before you begin this exercise, make sure you return to the `master` branch by checking it out via SourceTree or the command line. To ensure you have all demo code available to you, after checking out, pull any changes your instructor might have posted since the last time you were on the master branch.

2.2 Create a New Branch and Project and Push to Personal

After updating the master branch, return to the `audioplayer` branch. From there, create a new branch named `audioplayer-metadata`. This will be the starting point for this exercise.

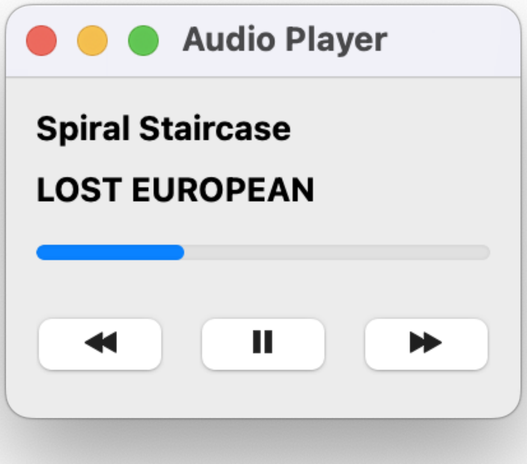
3 Features

After opening an audio file, the application will show the artist and track names above the progress bar. The application playing back a file is shown in Figure 1.

4 Additional Details

You will only need to make minor modifications to the player program. The metadata is available only after the file has been loaded. This means that the code to extract and display the metadata must be added to the `MainWindow::updateMediaStatus` member function, specifically to the `QMediaPlayer::LoadedMedia` case.

You can retrieve the metadata via the `QMediaPlayer::metaData()` member function. The `QMediaMetaData` class provides several tags in a key–value format. Since we only need to retrieve strings, the easiest member function to use is `QMediaMetaData::stringValue()`. The key must be a member of the `QMediaMetaData::Key` enum. The title will use the `QMediaMetaData::Title` key and the artist will be accessible via the `QMediaMetaData::ContributingArtist` key. Unfortunately, Qt as of now does not seem to support accessing the album cover art, although there is



Audio Player

Spiral Staircase

LOST EUROPEAN

Figure 1: Audio Player Application Showing Metadata

a `QMediaMetaData::CoverArtImage` key. Once extracted, you will need to update the labels that you have added to the user interface file.

5 Submit Your Work

Commit your changes locally. Again, make sure to use an appropriate commit message. With your working directory being the top–level of the repo, create the archive with:

```
$ git archive -o audioplayer-metadata.zip HEAD audioplayer-metadata
```

Then upload the `audioplayer-metadata.zip` file to canvas. Submit the pull request (repo URL and commit ID) as the upload comment.

6 Work Summary

- Create the `audioplayer-metadata` branch
- Modify the user interface layout to add labels for artist and track
- After the audio file is loaded, update the artist and track labels with the extracted metadata

- Commit and push your work to the personal remote
- Submit the pull request and archive to Canvas