Faculty of Science, Engineering and Technology

Introduction to Programming

Credit Task 7.2: GUI Music Player

Overview

This task allows you to demonstrate the ability to use the Gosu library and API (application programmer interface) to develop a graphical interface application. The application is a Music player which is a graphical interface version of Pass Task 7.1 (Music Player with Menu). The program should read in album and track information from a predefined file. The user should then be able to view the albums and select an album, then view the tracks and select a track to play.

Purpose:	Learn to write a user interface application using a structured programming language. you need to use the Gosu library and API.
Task:	Demonstrate the use of: • File handling • Arrays • Records/Class attributes • Libraries (APIs) in the context of the requirements for the Music application described in this document.
Time:	This task should be completed before the start of week 12.
Resources:	Sobkowicz, M 2015 *Learn Game Programming with Ruby: Bring Your Ideas to Life with Gosu*, The Pragmatic Programmer. Frieder, O. Frieder, G. & Grossman, D. 2013 Computer Science Programming Basics in Ruby, O'Reilly Media (Chapter 6) Flanagan, D. & Matsumoto, Y. 2008 The Ruby Programming Language, O'Reilly. **Pine, C 2014, *Learn to Program (2nd Ed), Chapter 11*, The Pragamatic Programmer (library version – follow the link)**

Submission Details

You must submit the following files to Doubtfire:

- Code for the program, written in Ruby
- A screenshot of the code running

Make sure that your task has the following in your submission:

- Code must follow the coding conventions used in the unit (layout, and use of case).
- You are storing and working with multiple values in an array.
- You are using records and enumeration to store the values.
- The code must run and you must capture a screenshot that shows it working in accordance with the requirements as described here along with any clarifications provided by your tutor.

Instructions

In this task you will build on the skills developed in your other Pass, Credit and Tutorial tasks.

For higher grades within the Credit range additional requirements are required. These are identified below.

You will be given feedback on how well you design your code as well as how well you name your artifacts. There is a minimum requirement for naming and design before your code can be accepted at any of the following grade levels – regardless of how well the code is functioning.

The different grades achievable for this task (within the Pass range) are:

Basic Credit Level – 60 (one star)

Middle Credit Level – 63 (two stars)

Higher Credit level – 65 (three stars)

Top Credit Level – 67 (four stars)

Note: See the Gosu audio API at https://www.rubydoc.info/github/gosu/gosu. You will need to use the following:

Gosu::Song.new(filelocations) – create a new song.

song.play(false) – play the song.

.

Basic Credit Level Requirements - 60

At this level use a combination of a text (Terminal) interface and a GUI interface. For some of the following you will be able to re-use your code from Pass Task 7.1.

The program must read in (from a file) at least 4 albums with up to 15 tracks for each album as well as the file location of each track.

Your application must:

1. Display the albums in the Terminal window including album information such as title and artist and allow the user to select an album, at which point you should display the tracks;
2. The user should be able to select an album and your player should start playing the album's tracks. At this point the user should be able to use buttons on a graphical panel and/or key presses to play, stop or skip tracks (i.e skip to the next track on the album). All other user interaction – as in 1 above - could be text based. You must use an audio API for this component;
3. Whichever track is playing you should display "The track you selected " then the track name " from the Album: " then the album name then " is now playing." This could be in the Terminal window or on the GUI screen.

At this level minimum validation is required. Just make sure your program does not crash if incorrect values are entered and that all fields have an expected value. The GUI for this may look something like the following:

Middle Credit Level Requirements - 63

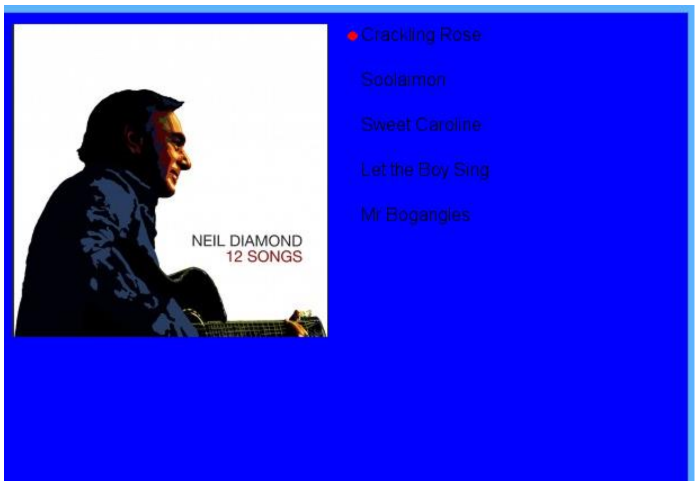
The program must read in (from a file) a single album and up to 15 tracks for the album.

The information read from the file should include:

- Album title
- Artist
- Artwork file name (place your artwork in an /images folder under the main folder)
- The number of tracks
- The title of each track
- The file location of each track (again, use a sub-directory/folder to store the song files.)

At this level user interaction must be entirely through a GUI. Your GUI interface should show a single album using either a text description, artwork or both. Users should be able to click on the Album information (i.e the artwork) and the tracks will be listed and the first track start playing. The tracks should continue to play in order until either they are finished or the program is stopped. The currently playing track must be indicated somehow (e.g the track could be highlighted or display a simple text message ‘Now playing ..’). You must use the Gosu audio API for this component.

Your GUI may look something like the following:



High Credit Level Requirements - 65

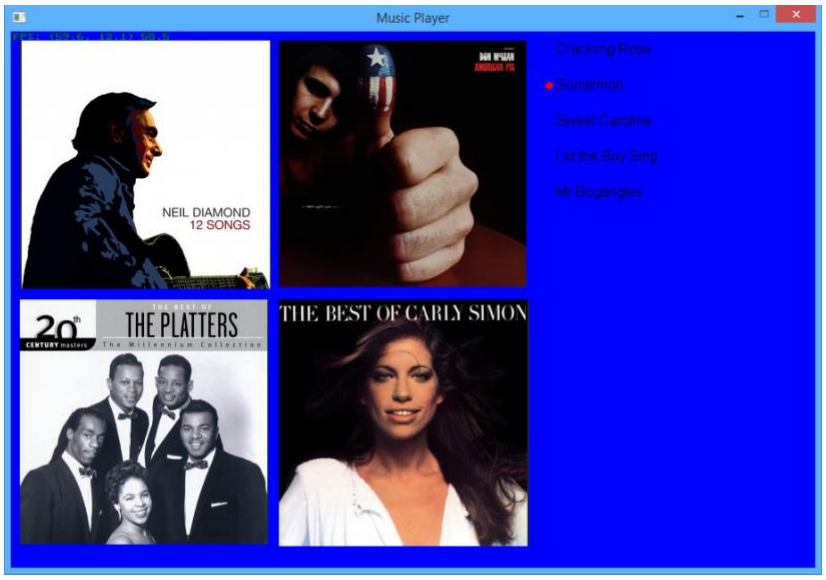
At this level your program must read in (from a file) at least four albums and up to 15 tracks for each album.

The information read from the file should include:

- Number of Albums
- Album title
- Artist
- Artwork file name (place your artwork in an /images folder under the main folder where you run the ./build.sh script)
- The number of tracks
- The title of each track
- The file location of each track.

At this level user interaction must be entirely through a GUI. Your GUI interface should show all the albums using either a text description, artwork or both. Users should be able to click on any Album information (i.e the artwork) and the tracks will be listed. When an album is clicked on the first track should start playing. The program should keep playing tracks in order until all tracks have been played or the user has clicked on another album, in which case that album’s tracks should start playing. The program should indicate somehow which track is currently playing. The GUI interface may look something like the following:

You must use the Gosu audio API for this component.



Top Credit Level Requirements - 67

At this level your program must read in (from a file) at least four albums and up to 15 tracks for each album.

The information read from the file should include:

- Number of Albums
- Album title
- Artist
- Artwork file name (place your artwork in an /images folder under the main folder where you run the program)
- The number of tracks
- The title of each track
- The file location of each track

At this level user interaction must be entirely through a GUI. Your GUI interface should show all the albums using either a text description, artwork or both. Users should be able to click on any Album information (i.e the artwork) and the tracks will be listed. The user should then be able to click on a track to play that track. The currently playing track must be indicated somehow (e.g the track could be highlighted or display a simple text message 'Now playing ..'). If the user clicks on another track (for the current album or another album) then any currently playing track should be stopped and the most recently selected track start playing.

You must use the Gosu audio API for this component.

Custom Project Extensions

You may wish to extend on the requirements above for your custom project. You would need to discuss this with your tutor as to what is required for different grade levels. Some possible extensions are:

Possible custom project Distinction level Extensions:

- Allow users to page through multiple pages of albums.
- Allow sorting of albums based on year recorded, genre ,etc.

Possible Custom project High Distinction extensions:

- Allow users to select tracks from various albums to create playlists. Users can then select from the playlists to play a pre-selected sequence of tracks.
- Some combination of all the above extensions.

End of Task