Music Player Project

1 Objective

In this project, we want you to create a Java application with GUI interface. You will mainly uses Swing components for creating a user interface (GUI) and then implementing the functions of the application using java programming. This project entitled as (Music Player) and this project as a one of two project assigned for this course.

2 Description

In this section, we show you the prototype of the application in GUI and data.

Note: All the GUI here is use for explanation the application, student can freely design the application as you want, but make sure the GUI is easy to use and the application has all required functions.

2.1 User Interface

Your application will has GUI like the suggested interface shown in Figure 1. However, you can create different GUI using traditional menu, or any design you like but still you have to include the entire functions which are listed in the next section.

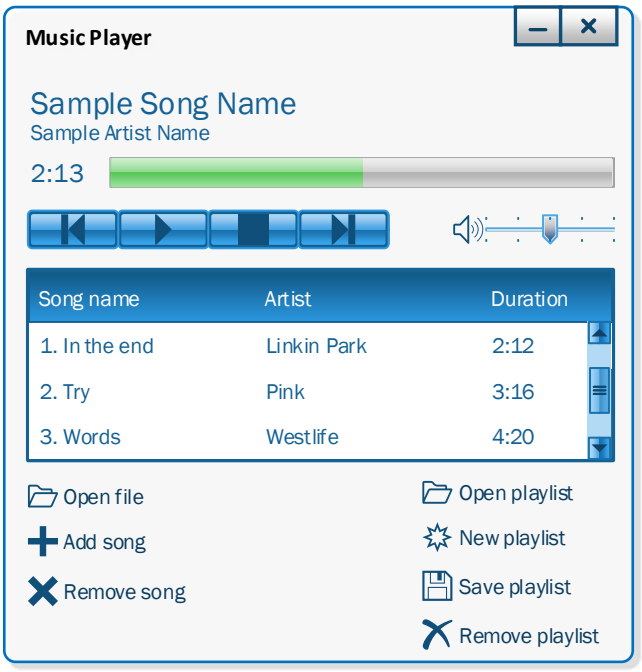

Figure 1: Music player sample user interface

2.2 Functions

The application will have 2 groups of functions:

- Playing song functions:
  - The application support music file of mp3 format.
  - The application must support Play, pause, and stop, next song, previous song operations by clicking on the buttons or using shortcut Ctrl + keyboard key.

Example: Ctrl + P to pause/play, Ctrl + S to stop, Ctrl + B to play previous song, Ctrl + F to play next song.

- Application must show the current playing time, song file name (title), and the current playing position in the song. The time must be update continuously while playing.
- Change the volume of the song.

❖ Playlist functions:

The playlist file format is **text file** and you can define the file structure of the saved playlist by yourself. For example to save the song name, artist, and the song file path in the file are shown as follows :

In the end; Linkin Park; c:\music\in_the_end.mp3

Try; Pink; c:\music\try.mp3

- Show the songs in the playlist
  - Show the title, artist and duration of each song
  - The list of the song can be scrollable.
  - Can select multiple song from the list to remove them using one click
- Open file:
  - Select **Open file.**
  - Choose 1 music file from JFileChooser.
  - The new playlist is created, and the song is added to this playlist.
- Add song to a playlist:
  - Select **Add song to playlist.**
  - Choose song(s) from JFileChooser which only shows .mp3 files.
- Remove song to a playlist:
  - Choose the song (Can choose multiple songs by using Ctrl key).
  - Select **Remove song(s)**.
- Create new playlist:
  - Select **New playlist**
  - Close the current playlist and remove the songs in the list.
- Open playlist:
  - Select **Open playlist.**
  - Choose 1 playlist file from dialog box which only shows .playlist files.
  - The new playlist is created, and the song is added to this playlist.
- Save the playlist:
  - If the playlist is new playlist the open JFileChooser and choose file path to save
  - If the playlist file is already create, just update content of the file.
- Remove current playlist:
  - Select **Remove playlist**.
  - Delete the playlist file.
  - There will be no song in list.

Hint:

- To play .mp3 file, you can use external library or use *Java Media Foundation* (JMF) library.
- You may have to create threads to update GUI and playing file.

3 Extra Credit

In this assignment you can earn extra credit by adding more functions for your application such as:

- Designing friendlier user interface.
- Adding more functions to the program.
  - User can choose the starting time to play (jump in the song and play)
  - User and move the position of the song in playlist.
- Handling abnormal case, such as: the choosing file is not the mp3 format, the file in the playlist does not exist (with Music Player)
- Support more music files type.
- Change the skin (Look-and –Feel) of GUI.

4 Project Submission

Here are the details for how to submit your project:

1. Create an executable JAR file named **musicplayer.jar** that contains your runnable application.
2. Create a new directory using your student IDs. Example: **20150001_20150002**
3. Put the **jar file** and your **Eclipse project folder** into the directory.
4. Put also **README.txt** file to the directory. This file should contain student name (in English), the version of Java you used as well as any special info we might need to know about your program (For example: let us know if you did some extra things).
5. ZIP this directory and submit it via i-Class.

Notes:

- All the **comments** on the code and the names of variables, classes **must be in English**.
- Please take care to **remove any platform dependencies**, such as hardcoded path names or dependence on a particular look-and-feel that may not exist on all platforms. Also, if you use any images in your application, please make sure that you included these images in your JAR file and that your code will refer to them and load them properly when they're in this JAR file (see this page for some details on how to include and load images from within a JAR file).
- Please test your program on other by running the jar file directly, not only run on eclipse, before you submit.

5 Grade

Grading for this assignment will follow this breakdown:

- ✓ 50% functionality (Are all functions run correctly?)
- ✓ 30% architectural design, coding style (Are your classes, methods, fields are reasonable?)
- ✓ 20% commenting (Can I understands your app just by reading your code?)
- ✓ 20% extra credit (Is your application interesting to use?)

Please let the LA or the Professor know if you have any questions or something you don't understand by visiting by asking question on e-class, after lab class or email.

If you have questions about the project requirement, I suggest you post it in public so other student can see them too but if you have a question about your project implementation, then post the question in private.

Good luck!