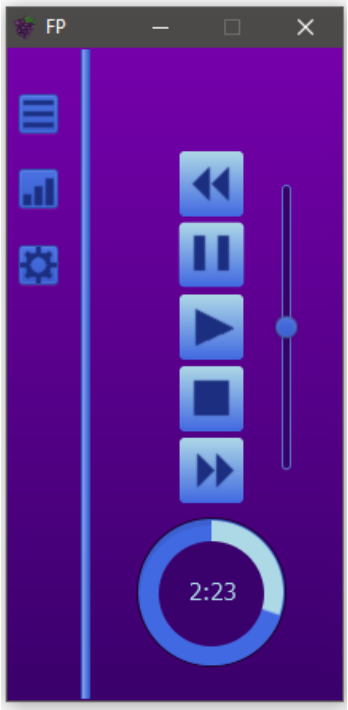
Documentation of FruityPlayer

1. What is FruityPlayer?

FruityPlayer is an audio player written in JavaFX by Grzegorz Rypeść in 2018. Its name comes from looks of the player which set of colors reminds of fruits. It's compatible with Linux systems and Windows. The goal of the project was to make a good looking player with very intuitive interface that has lots of features. These are: audio equalizer, using Gracenote music database to obtain information about tracks, allowing user to manipulate his playlist and to easily switch player's skins, storing settings on a hard drive.

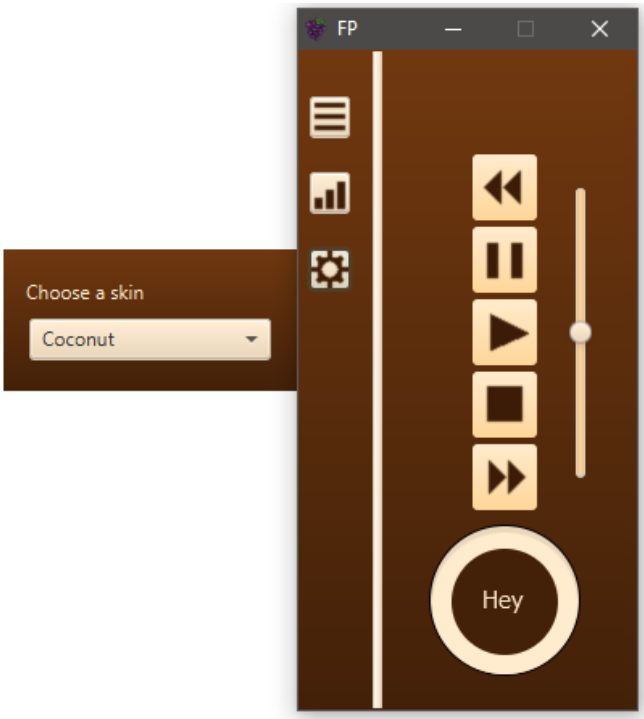
2.User's manual

- Main window consists of 5 buttons placed in the middle that manipulate current track. On the left there are: options, equalizer and settings button. Clicking on each of them will display an additional window or hide it if it's been already shown. Slider on the left allows to change volume and clicking on a progress circle at the bottom will start playing track from the set moment.

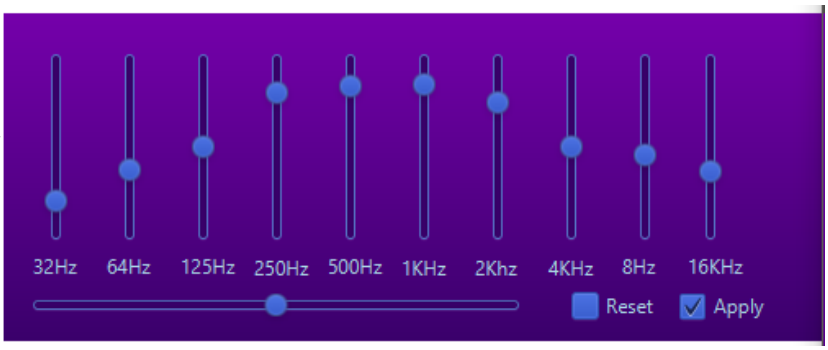


Main window

- Settings window allows user to change skin of the player by selecting one from a list view.



Equalizer window gives the ability to change volume of a certain band.
You can manipulate balance of sound with the bottom slider. Make sure to select apply button to apply changes. Clicking reset button will set all sliders to default.



Playlist manager window displays user's playlist. You can add new tracks with the plus button and you can remove selected track with the x button. Clicking on i button will display an **info window** that contains basic information and an album cover of the track that is currently played.

3. Implementation:

Full set of classes, theirs methods, fields and properties is enclosed in javadocs which you can find in fruitydocs file. Here you will find general information on how the code is grouped.
Classes are divided into 3 groups accordingly to MVC model. Apart from that there is also the main class that launches application.

Model class contains logic of the application that is used by controllers. Model doesn't contain controller and view classes so it's portable to other projects. Additional classes that belong to model are:
Playlist – Keeps tracks of current playlist as a vector of files.
DurationExtended – It's a javafx.util.Duration class with additional method toMinutesAndSeconds which converts duration to formatted and ready to display string.
TrackData – Used to represent information of the track.
Gracenote classes were made by Rich Adams* under Creative Commons Attribution 3.0 Unported license. They make a wrapper to the Gracebote Web API which sends xml queires. I removed from them every method that prints text to console and created my own that returns a string built from received responses.

View consits of 5 .fxml files, CSS stylesheets and png images. Each .fxml file defines each window of the application and is connected to its controller. These files are:
MainStage.fxml
Equalizer.fxml
Info.fxml
PlaylistManager.fxml
Settings.fxml

CSS files define looks of elements specified in .fxml files. Changing skin of the player is simply implemented by loading another CSS file to fxml classes.

Each .fxml file has its own **controller** of simillar name which defines its behaviour. Controllers change the state of model and view. In addition to that ControllerMain performs operations on settings.fruity file that stores settings of the player which are: user's ID required to create queries to Gracenote database and current skin of player. ControllerMain also implements the way the error windows are displayed. These windows are so simple that I put them in controller section instead of view one.

*https://github.com/richadams/php-gracenote

4.License

FruityPlayer is licesed under Apache License 2.0.