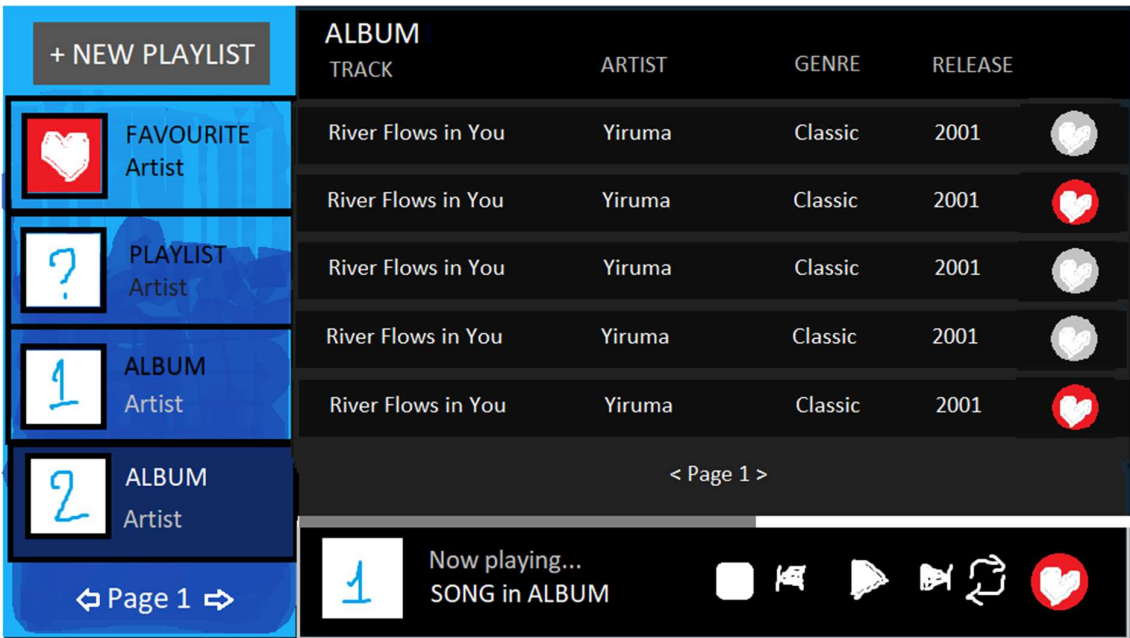


# Design Overview for Advanced GUI Music Player

Name: Nguyen Thi Thanh Minh  
Student ID: 104169617

## Summary of Program

From the GUI Music Player task in 9.1.3D, I'd like to advance it furthermore so that it has all fundamental and some advanced functions of a common music player program.



The sketch of my GUI Music Player with basic and some additional features

## Required Data Types

We need Track and Album class to load albums from the directories into the program.

Table 1: Track class details

Field Name	Type	Notes
id	String	Unique key for each track, combined with album ID
name	String	Track name
location	String	Directory path + formatted track name

Table 2: Album class details

Field Name	Type	Notes
------------	------	-------

<b>id</b>	String	Unique key for each album
<b>artist</b>	String	Album Artist
<b>title</b>	String	Album Title
<b>rls_date</b>	Integer	Album Release date
<b>genre</b>	Integer	Album Genre
<b>artwork</b>	Gosu::Image	Album Artwork image
<b>tracks</b>	Array	Array of Track class data

Then, we need Element class to assign values to things we want to draw with Gosu.

Table 3: Element class details

Field Name	Type	Notes
<b>x</b>	Float	x coordinate of the element
<b>y</b>	Float	y coordinate of the element
<b>width</b>	Float	Element's width
<b>height</b>	Float	Element's height
<b>color</b>	Gosu::Color	Base color
<b>state</b>	Gosu::Color	Changes when hovered/selected
<b>z</b>	Integer	Zorder (layer drawn on)
<b>name</b>	String	For click procedures

ZOrder is a common module in Gosu, which determines the layer things will be drawn on.

Table 4: Zorder module details

Value	Notes
<b>BACKGROUND</b>	The background non-interactive shapes
<b>PLAYER</b>	Lower interactive elements like album panels, track panels...
<b>UI</b>	Interactive elements on the top

In addition, the State module will help tell which state the element is currently in to draw an overlay color on.

Table 5: Zorder module details

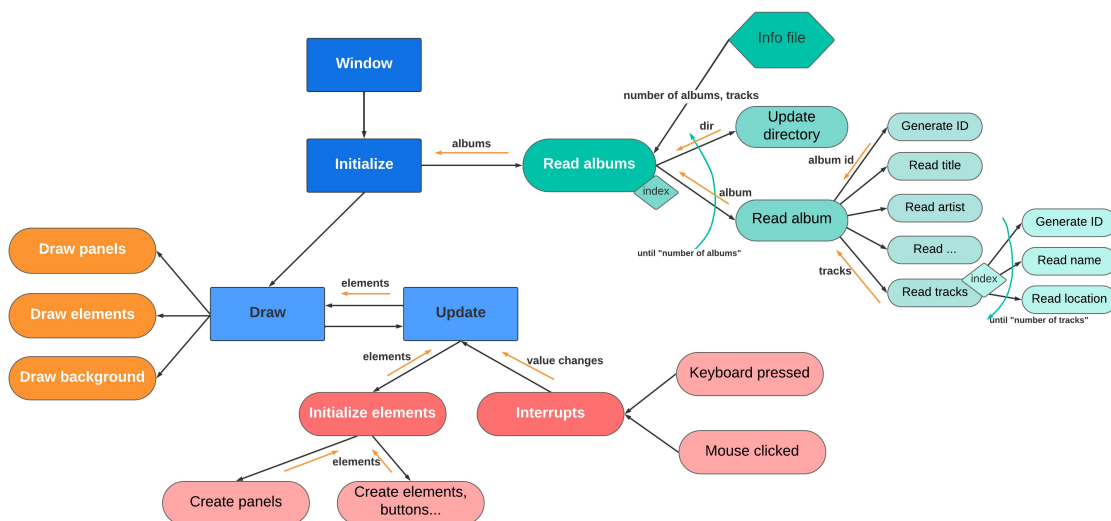
Value	Notes
<b>NORMAL</b>	Transparent overlay
<b>HOVERED</b>	Faint white overlay
<b>SELECTED</b>	Vivid white overlay

## Overview of Program Structure

Specifically, I want to implement these features:

- Display playlists in 4-panel page, allowing iterating within limit of pages number.
- Automatically detect and load albums in predefined format.
- Each element hovered by the mouse will be highlighted. Selected playlist and playing track will be highlighted more brightly.
- When selecting a playlist, its tracks will be displayed with information (name, artist, genre, release). Upon clicking on a track, it will start playing (stop any previously playing track), and show the playbar with buttons.
- Allow the user to play/resume, pause, stop, repeat the current track.
- Auto-playing, skip/rewind to next/previous track in the selected playlist (including albums, favourite list and custom playlists)
- Add the current track to the favourite list (the first playlist) with the heart-shaped button on the playbar. On the tracks panel, there are also non-interactive heart-shaped elements indicating whether the respective track is in the favourite playlist.
- Create a new custom playlist with the "NEW" button on the top left corner. Upon clicking, the program will enter the Creative Mode, which allows the user to click on wanted tracks without interrupting the possibly playing track. When done, click on that button, now named "OK".
- Sort albums by artist (by alphabet), genre (by genre order: Pop, Classic, Jazz, Rock) or release date (from earliest to latest) without changing the order of favourite and custom playlists.
- Adjust the volume of the player by clicking on the volume levels or pressing arrow left/right keys
- Display the elapsed time over the total time of the song with a progress bar and numbers.

## General Hierarchy Structure Chart



*The End*