David Allen

T1A3 - Terminal Application Presentation

Set List Generator

As a working musician, one of the more mundane and time-consuming tasks is creating a fresh song list for every performance.

Some performers choose to play the same songs in the same order at every gig (boring). This is a simple app for those who prefer to keep it interesting.

Features

1. Generate songs from a ‘starter playlist’ ie. a file containing commonly performed songs.
2. Add / Remove songs
3. View playlist
4. Recall last used playlist
5. Exports as JSON

Future features: sort by genre or other parameter

Way, way, way down the line this could be updated to perform tasks such as - automatically creating Spotify playlists, interacting with sheet music/lyrics apps etc.

Using the app

1. User enters their name - app checks if there is a saved playlist, otherwise creates a new one
2. User is asked how many songs they'd like on the list
3. Option to generate entirely from starter playlist
4. Option to add their own songs
5. Option to delete songs
6. Option to shuffle playlist
7. Final playlist is printed to the screen

App logic

user.rb - welcomes user and performs checks

song.rb - creates a new array for user to enter new songs or delete existing ones

setlist.rb - fills out remaining songs from starter playlist and prints final result to the screen

art.rb - stores ASCII logos

generator.rb - actual app to be used

generator.rb

```ruby
require_relative 'setlist'
require_relative 'song'

puts "Welcome to setlist generator"
puts ""
puts "An app for musicians to keep things fresh!"

song = Song.new
loop do
    puts "What would you like to do?"
    choice = gets.chomp.downcase
    if choice == "add"

      puts "Continue adding songs. Hit enter when finished."

        loop do
            print "Title: "
            break if (title = gets.chomp) == ""
            print "Artist: "
            artist = gets.chomp
            song.add(title, artist)
        end

    elsif choice == "remove"
        print "Song to remove: "
        title = gets.chomp
        song.remove(title)

    elsif choice == "view"
        final_list = Setlist.new
        final_list.view(song.push_to_setlist)
    else
        break
    end
end
```

song.rb

```ruby
class Song
    def initialize
        @arr = []
    end

    def add(name, artist)
        @arr << {name: name, artist: artist}
        puts "#{@arr[-1][:name].capitalize} by #{@arr[-1][:artist].capitalize} added to your set list."
    end

    def remove(name)
        @arr.delete_if {|h| h[:name] == name}
    end

    def push_to_setlist
       @arr
    end
end
```

setlist.rb

```ruby
class Setlist
    def view(array)
        array.each_with_index do |item, index|
            puts "#{index + 1}. #{item[:name].capitalize} by #{item[:artist].capitalize}"
        end
    end
end
```

Challenges

TDD has been a very difficult concept to wrap my head around. I'm sure it is truly the best approach and is used in the industry, however at this point in time it feels counter intuitive

Interacting with the file system has been tricky and is still a work in progress

Favourite Parts

Gaining more of an understanding of Classes and Methods has made the code a lot cleaner

Things to do...

- File interaction
- Gems - colorize, tty prompt and more
- Formatting - clearing the screen, new lines etc
- More testing