Report Milestone 2 - Website @ Playlist Explorer (playlist-alchemy.herokuapp.com)

Our project aims to explore the identity of Spotify Playlists. Beneath the surface, playlists are a stranger beast than one might expect. It represents a perfect pocket of complexity to be explored visually. The goal of our visualizations is to allow Spotify users to gain insight into the patterns hiding in their playlists.

The website offers two options : if the user owns a Spotify account, he can login to import one of his own playlists. Otherwise, he can select which playlist he wants to explore among a selection displayed on the main page.

Task 1 - Design the basic skeleton of the webpage. Courses Needed : Web Development Lectures (1&2) . Tools : Material Design Lite framework

Task 2 - Implement a login portal to Spotify according to the **Authorization Code Flow** in order to get access tokens and client credentials. Resources needed : Documentation about how to make authorized request to the Spotify API (Spotify for Developers Website)

Task 3 - Build a layout to be able to embed playlists into the website using **Spotify Playlists Widgets** and implement buttons to select playlists to visualize.
Ressources needed : Documentation on Spotify widgets & Javascript lectures.

Task 4 - Implement the Javascript code to make AJAX requests to the Spotify API in order to import the Playlist data from the Spotify Website. Resources needed: Ajax documentation on server requests

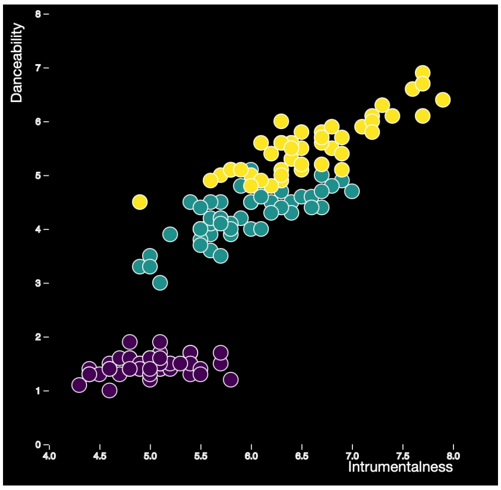
We consider the first visualization our minimum viable product, it is dedicated to the Playlist Audio Features. The user can interact with a scatter plot displaying the songs according to the values of their audio features. He will be the main actor of the visualization by selecting which feature goes along each axis of the plot to be able to explore what he wants. Here is a putative list of the audio-features the user can select from : key, tempo, valence, danceability, loudness, speechiness, acousticness, instrumentalness, liveness.

The goal is to both discover the core features of a given playlist and to compare different playlists. As such, the user can display different playlists in the same plot to visually compare them.

Task 5 - After the user is done selecting the playlists to analyze from his Spotify account, clicking an ‘analyze’ button will trigger a request to import the corresponding audio features data from the server. The button also scrolls the page down from playlist selection to playlist visualization. The requested features are plotted. We will be using D3.js. Resources needed: D3.js lectures - Javascript lectures

Task 6 - Implement the whole interface allowing the user to interact with the plot.
Resources needed: D3.js lectures - Javascript lectures

I mainly feature scatter plots and histograms, these are not the most sophisticated plots but they provide an effective way to explore audio data.



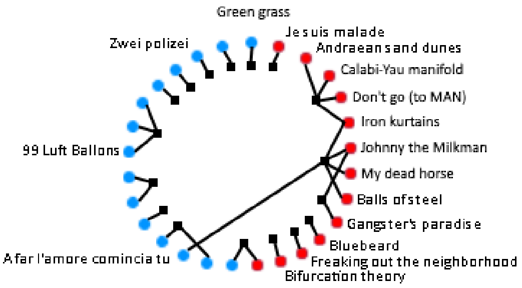
To go further

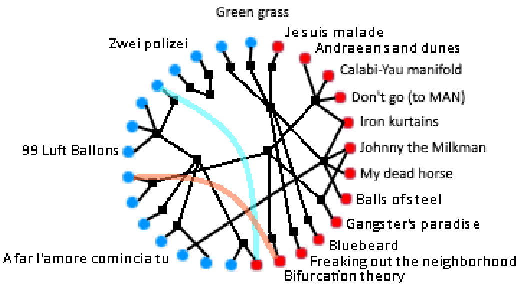
Task 7 - Artists interactive graph
<u>Resources needed</u>: D3, Interactions, Graphs, Artists API
What can artists tell us about the relations between the tracks of a playlist?

The visualization: First, the user selects one or more playlists. Then, all tracks in the playlists are displayed in a circle, with their names. The artists of each track are displayed inside the circle, linked to their respective track by a straight line. The user can then perform a "search" step iteratively: all related artists of the artists found in the previous iteration are retrieved and added to the visualization, linked to the previous artists, forming an undirected graph (we assume relatedness to be symmetric). Once a path has been found from one track to another, a curved line is drawn between them, color-coded depending on its length (number of edges).

Extra: The complete network can be toggled off to leave only the established links between tracks. The circle of tracks can be color-coded depending on the playlist they belong to. Artist nodes can be clicked to display a small profile and a link to their Spotify page or personal website. The visualization is drawn using a force-directed algorithm. The visualization can also be set to auto-run to avoid clicking the button repeatedly.

Motivation: We suspect that artists represent a complex latent variable which captures subtle relationships not visible from more explicit features. As a bonus, brushing and linking can be implemented between this and other views (e.g. the audio features view) in order to contrast this implicit measure with more explicit measures. It's also fun, since the iteration aspect actively involves the user and might reveal surprising information.





Task 8: Visualization of the timeline of a playlist - Cut for space, implement if time allows

Motivation: Sometime, a playlist has a certain flow, evolving in a coherent manner from one kind of music to another.
NOTE: Unfortunately, due to a problem with Material, much of the website does not display, including the visualization...