

Projet information Visualisation



Visualisation des genres musicaux



Damien STENGEL - Karim CHARLEUX - Wassim BARATLI
- Elias MERDADI - Romain PELLEGRINI

Description du projet

Notre sujet

Comprendre les tendances et les relations entre les genres musicaux et les artistes.

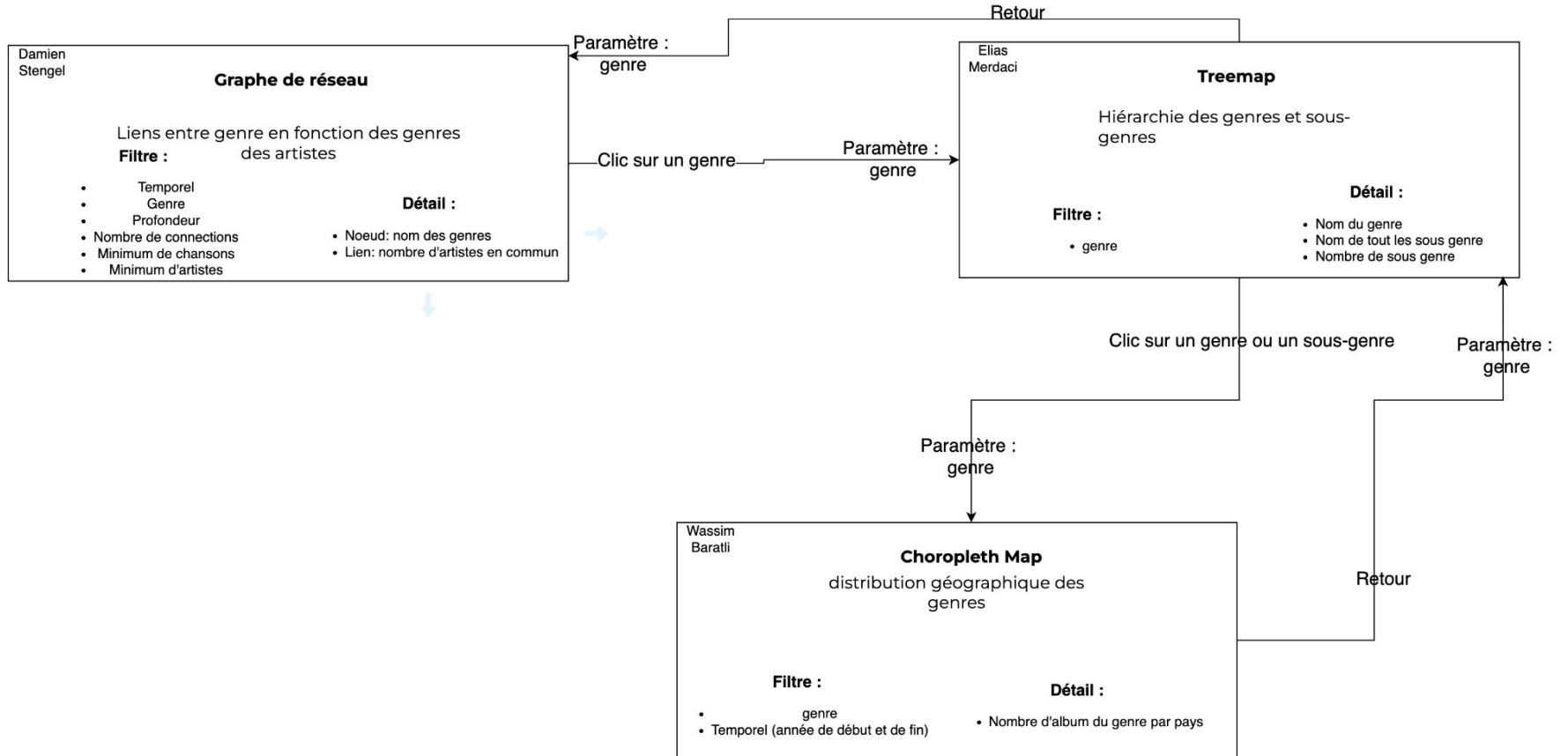
Nos utilisateurs

- Professionnel musical
- Amateur de musique

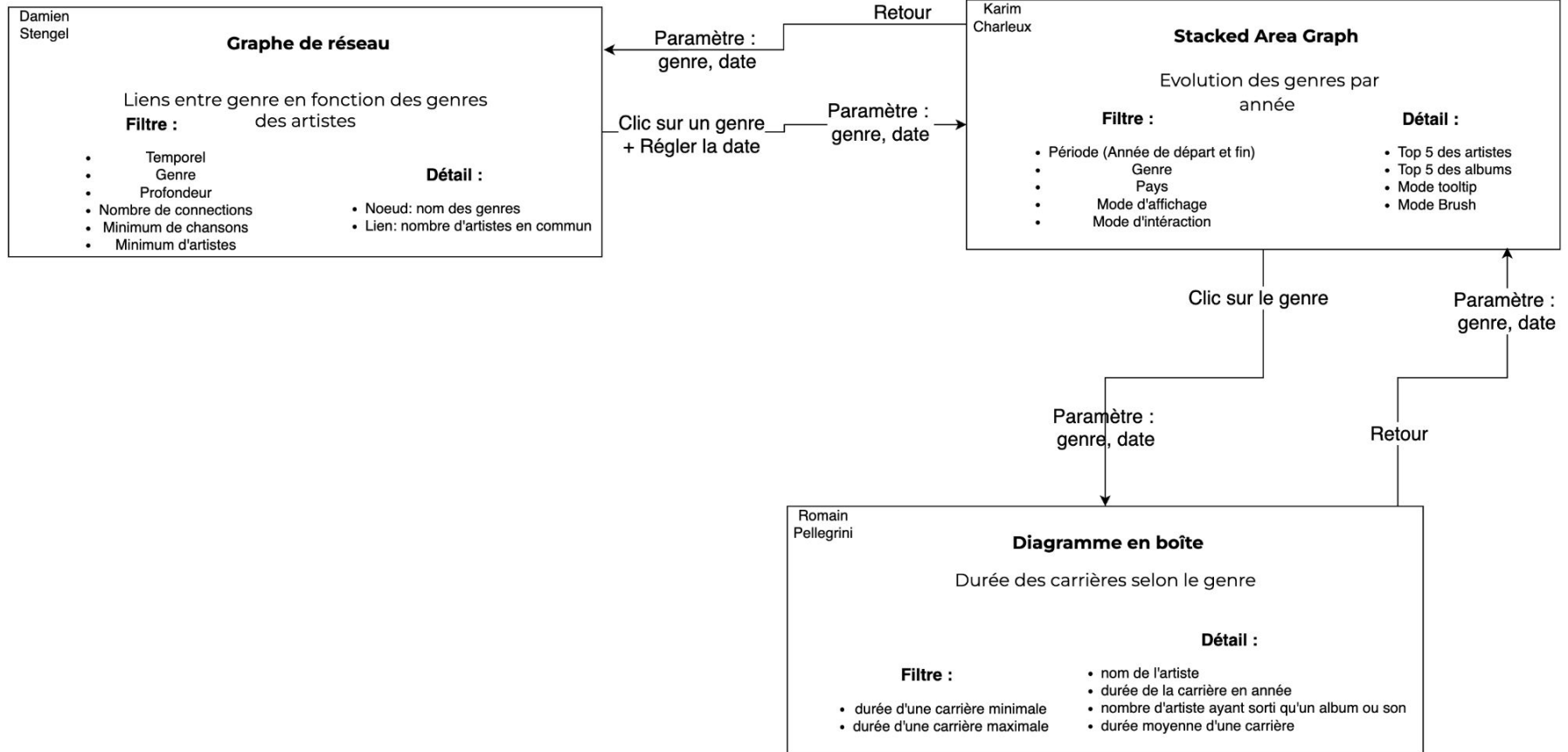
Nos Objectifs

- Identifier
- Comparer
- Associer
- Catégoriser
- Corréler
- Observer

Présentation du workflow des genres



Présentation du workflow des genres et dates



Traitement des données pour le graphe de réseau

Importation des données

Entrée : Fichiers JSON contenant les données des artistes (`artist-without-members.json`) et des chansons (`song.json`).

Attributs :

- `id_artist_deezer/main_genre`
- `id_artist_deezer/album_genre/duration/publicationDate/country`

Traitement : Ouverture et lecture des fichiers JSON.

Sortie : Données d'artistes et de chansons prêtes pour le traitement suivant.

Traitement des dates

Entrée : Données de dates de publication dans les informations de chaque chanson

Traitement : Extraction et normalisation des dates, par exemple en prenant uniquement l'année partir de la date complète.

Sortie : Dates structurées pour chaque chanson

Traitement des genres

Entrée : Données de genres musicaux bruts pour chaque artiste et chanson.

Traitement :

- Nettoyage des noms de genres avec `clean_genre_name`.
- Conversion des caractères spéciaux en leur version alphabétique.
- Suppression des espaces superflus.

Sortie : Liste nettoyée des genres musicaux.

Génération de la visualisation

Entrée : Objets `nodes` et `links` générés dans l'étape précédente.

Traitement :

- Structure les données dans un format compatible avec D3.js.
- Sauvegarde des résultats dans un fichier JSON pour utilisation avec D3.js.

Sortie : Graphe de visualisation des genres musicaux.

Calcul des connexions

Entrée : `Genres` de chaque artiste et chanson traités.

Traitement :

- Calcul du nombre de chansons et artistes par genre.
- Agrégation des informations par année et par pays.
- Création des connexions entre genres via les artistes qui produisent dans plusieurs genres.

Sortie : Objets `nodes` et `links` avec les connexions pondérées pour la visualisation.

Traitement des données pour le Tree Map

Importation des données

Entrée : Fichiers JSON contenant les données des chansons (`song.json`)

Attributs:
- genre

Traitement : Ouverture et lecture des fichiers JSON.

Sortie : Données des albums et de chansons prêtes pour le traitement suivant.

Traitement des genres

Entrée : Données de genres musicaux bruts pour chaque chanson.

Traitement :

- Conversion des caractères spéciaux en leur version alphabétique.
- Conversion des entités HTML en caractères normés.
- Conversion de toutes les majuscules en minuscules

Sortie : Liste des genres traités

Création des données nécessaires

Entrée : Json contenant tout les sous-genres trouvés sur Internet (`sous-genre.json`)

Traitement :

Comparaison avec la liste des genres pour vérifier qu'au moins chaque genre de la base de données est contenu dans les sous-genres.

Sortie : Liste de sous-genre adaptées à no.

Génération de la visualisation

Entrée : La structure de données contenant les informations sur l'arbre compatible avec DS3.js

Traitement :

Création d'un tree à l'aide de D3.js avec la structure de données reçu.

Sortie : Tree Map des genre parents et sous-genre du genre sélectionnés

Génération de l'arbre

Entrée : Liste des sous-genres traitées ainsi que le genre sélectionnés

Traitement :

- Recherche de tout les genre parent au genre sélectionné.
- Recherche de tous les sous-genre au genre sélectionné
- Vérification de la compatibilité avec D3.js

Sortie : Structure de données contenant le genre sélectionné , ses parents et ses enfants.

Traitement des données pour la ChoroplethMap

Importation des données

Entrée : Fichiers JSON contenant les données des albums (`albums.json`)

Attributs:

- country
- genre
- publicationDate

Traitement : Ouverture et lecture des fichiers JSON.

Sortie : Données d'albums prêtes pour le traitement (pays, genre, année)

Traitement des genres

Entrée : Données du genre de chaque album.

Traitement : Nettoyage des noms de genres (suppression des espaces inutiles, uniformisation en minuscule).

Sortie : Liste nettoyée et standardisée des genres musicaux.

Traitement des dates

Entrée : Données de date pour chaque album.

Traitement :

- Extraction et normalisation de la date de publication.
- Filtrage pour s'assurer que les dates sont dans la plage souhaitée (ex. entre 1950 et 2022).

Sortie : Années de publication pour chaque album, sous forme de données structurées.

Génération de la visualisation

Entrée : Données JSON agrégées (`choropleth_data.json`) et fichier GeoJSON pour la carte du monde.

Traitement :

- Chargement de la carte et des données.
- Application d'une échelle de couleurs pour représenter la fréquence des albums par genre dans chaque pays.
- Ajout d'interactions (tooltips, filtres de genre et d'année).

Sortie : Carte interactive où chaque pays est coloré en fonction du nombre d'albums de chaque genre et des filtres d'années.

Agrégation des données pour la visualisation

Entrée : Données traitées avec `country`, `genre`, et `year`.

Traitement :

- Calcul du nombre d'albums par genre, année et pays.
- Création d'une structure de données hiérarchique permettant de lier chaque pays à ses genres et à la répartition des albums par année..

Sortie : Fichier JSON contenant les données agrégées, structurées par `country > genre > year` avec les comptages.

Traitement des données pour la Stacked Area Chart

Importation des albums

Entrée : Fichiers JSON contenant les données des albums (**albums.json**)

Attributs:

- **_id**
- **id_artist**
- **country**
- **publicationDate**
- **cover > small**

Traitement : Ouverture et préparation des informations des albums

Sortie : Données d'albums prêtes pour le traitement (id, id_artist, country, date, cover)

Importation des artistes

Entrée : Fichiers JSON contenant les données des artistes (**artist.json**)

Attributs:

- **_id**
- **name**
- **country**
- **picture > small**

Traitement : Ouverture et préparation des informations des artistes

Sortie : Données d'artistes prêtes pour le traitement (id, name, country, picture)

Importation des sons

Entrée : Fichiers JSON contenant les données des artistes (**song.json**)

Attributs:

- **_id**
- **publicationDate**
- **album_genre**
- **rank**
- **id_album**

Traitement : Ouverture et préparation des informations des sons

Sortie : Données des sons prêtes pour le traitement (id, id_album, date, genre, rank)

Génération de la visualisation

Entrée : Données JSON agrégées (**genre_evolution.json**)

Traitement :

Préparation des données :

- Conversion des dates, calcul des totaux par année, 2 types de visualisation

Configuration du graphique :

- Définition des échelles (temps, valeurs), création des axes, configuration des couleurs par genre

Création des interactions :

- Tooltip au survol, filtres de période, genre
- Affichage des top 5 albums/artistes

Sortie : Graphique à aires empilées interactif avec des légende des contrôle de filtrage, tooltips informatifs et deux top 5 dynamique

Agrégation des données pour la visualisation

Entrée : Données traitées avec les **albums**, **artistes**, et **sons**.

Traitement :

Pour chaque année, agrégation par genre :

- Comptage du nombre de chansons (count)
- Calcul de la somme des ranks (rank_sum)
- Calcul du rank moyen (rank_avg)
- Création des statistiques albums
- Création des statistiques artistes

Sortie : Fichier JSON structuré par année > genre pour chaque genre statistiques globales, albums et artistes

Traitement des données pour le diagramme en boîtes

Importation des données

Entrée : Fichiers JSON contenant les données des artistes (`artist-without-members.json`) et des chansons (`song.json`).

Attributs:

- `id_artist_deezer/name`
- `id_artist_deezer/album_genre/publicationDate`

Traitement : Ouverture et lecture des fichiers JSON.

Sortie : Données d'artistes et de chansons prêtes pour le traitement suivant.

Traitement des dates

Entrée : Données de dates de publication dans les informations de chaque chanson

Traitement : Effacement des dates incorrectes exemple 0000-00-00

Sortie : Liste de `dates` pour chaque `artistes` et son `genre`

Traitement des artistes

Entrée : Nom de l'artiste dans `artist-without-members.json` à partir de l'id récupéré dans `song.json`

Traitement :

- Nom : "Unknow Artist" si l'artiste ne possède pas d'identifiant

Sortie : Liste nettoyée des `artistes`.

Génération du diagramme

Entrée : Objets json avec la durée de la carrière.

Traitement :

- Structure les données dans un format compatible avec D3.js.
- Sauvegarde des résultats dans un fichier JSON pour utilisation avec D3.js.

Sortie : Diagramme en boîtes avec la durée des carrières des artistes du genre sélectionné

Création du json

Entrée : Liste des dates par artiste et liste des artistes

Traitement :

- ajout d'un champ `first_song_date` pour la date du premier titre
- ajout d'un champ `last_song_date` pour la date du dernier titre
- calcul de la différence entre le premier et dernier titre

Sortie : Objets json avec la durée de la carrière