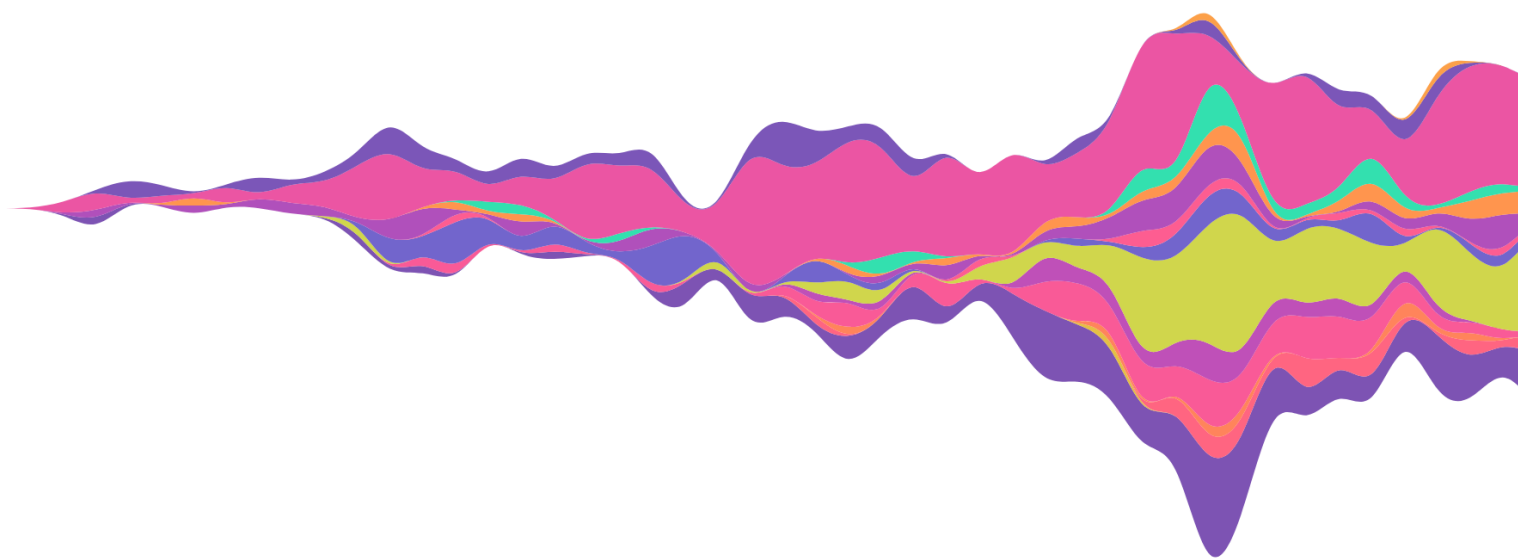


Landscape of genres

Evolution of genres at Montreux Jazz festival



Process book

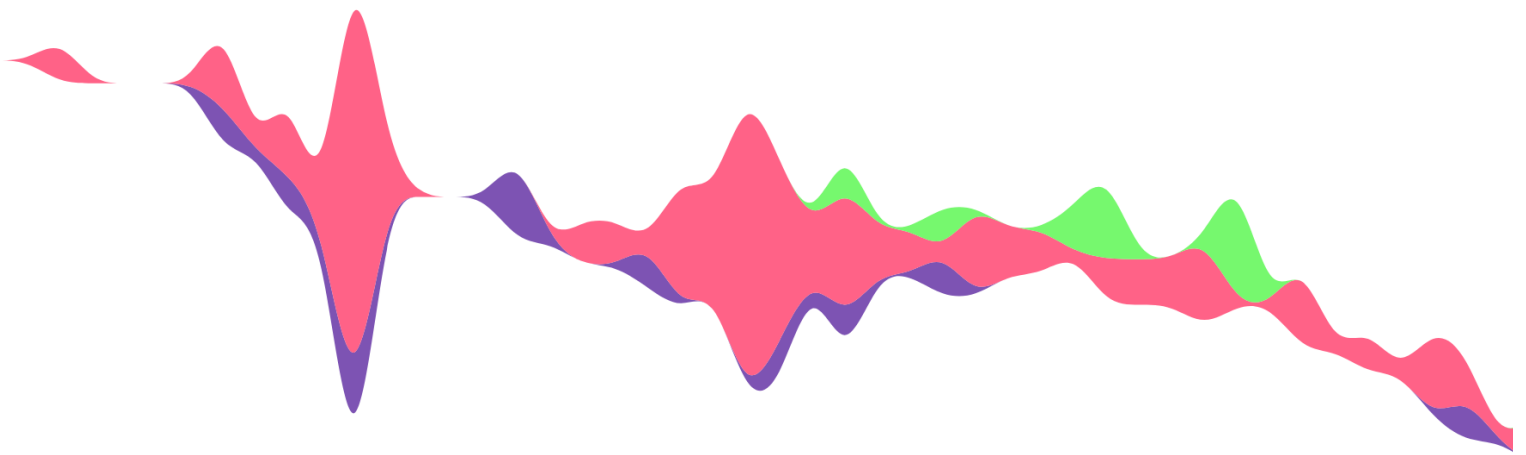
Viktor Sanca

Sebastijan Stevanovic

Dino Mujkic

If you want peace of soul, come to Montreux

Freddie Mercury



Introduction

The main objective of this process book is to guide the reader through the evolution of the project from the spark of idea to the realization and materialization. It covers the project of the Fall semester of 2017. at the course of Data Visualization (COM-480) at EPFL.

The course provided an insight into practical creativity and heuristics that should be followed to make our future visualizations impactful and useful. Moreover, it provided insights into what we should not do as well. We hope that the knowledge obtained will serve its purpose well through the project we will implement.

Overview

Montreux Jazz festival is the second largest annual jazz festival in the world. Since 1966. it represents a point of convergence of the best artists in the world and a source of best tunes for in the summer. Many artists, both famous and aspiring groups and individuals take part in the festival days and enrich the atmosphere with either fresh sounds and interpretations, or simply sound the jazz standards. Throughout the years, MJF has evolved – both in numbers of festival days, venues and artists. Most importantly, the sounds, melodies and genres have flourished and flowed through the years differently.

Motivation

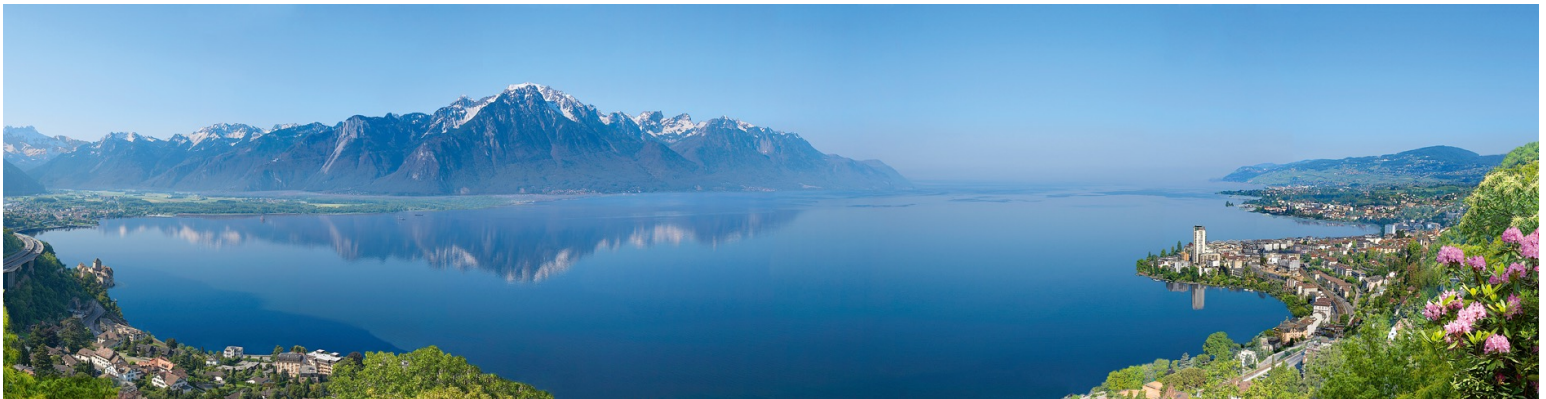
Contrary to the name, Montreux Jazz festival is not exclusively a Jazz festival. The festival welcomes many artists and tunes. This has motivated us to look into the evolution of the genres at the festival throughout the years. We would like to explore the different landscapes of tunes throughout the years. We expect a variety of genres appearing and disappearing through the decades. Through our visualization we would like to capture more than half of a century of music of the festival – a snapshot of evolution of music at Montreux Jazz.

Target audience

Our target audience is anyone interested in finding out more about Montreux Jazz. We would strive to make our visualization accessible and appealing to the wide audience. We hope that we can provoke a thought or at least provide an overview of the decades of MJF music in a simple glance and give method for audience to interact and learn more through it.

Inspiration

We have been captivated by the beauty of the landscape of lake Geneva and the Alps.



Thus we have decided that we need to incorporate a part of the landscape into our future visualization – therefore the title: *Landscape of genres*.

Furthermore, an inspiration came from many legendary artists performing and events happening at Montreux, as well as Montreux Jazz festival. Starting from Queen recording their songs and untimely final album at Mountain Studios at Montreux, to the famous song Smoke on the Water referring to the fortunately not tragical event of Montreux Casino burning down.

An insight in legendary history is something we wanted to provide.

Concept

With previously stated inspiration and goal, in this section we present the conceptual ideas we have come up with during the development process. This part should provide the greatest insight in the evolution of the visualization, both aesthetically as well as through changes in functionalities. At this point we will discuss high level concepts but with clear outcome that should be realized in practice.

We will provide both sketches as well as real demonstration snapshots where appropriate.

Dataset

We have been provided with the Montreux Jazz festival database. It contains information on many entities: artists, concerts, venues. Since obtaining the dataset has required an NDA, no detailed insight will be provided on the dataset itself. A short overview of some parts and contents of such database is publicly available at the official MJF website: <http://www.montreuxjazz.com/concerts-database>.

The database has been exported to JSON for a more convenient information access and processing.

In general terms, from the information about the concerts throughout the years we have managed to extract the information about the main genres, artists and subgenres appearing at a specific venue. We have aggregated such data and processed it to show the main genres, as well as subgenres conveniently throughout the years.

Additional information has been required in order to construct the links between main genres and subgenres. This information has been extracted and constructed from different sources online, mainly: <http://www.musicgenreslist.com/>. More details about data processing and suitable formats and transformations are available in the source code.

The exploratory data analysis has been conducted both through Python in a convenient matter utilizing Pandas library for efficient JSON exploration, as well as through Javascript in more practical parts of implementation. In this way we have become acquainted with the data and what pieces of information are available and what would be a feasible scope of the project.

It has to be noted that we did not seek a dataset to suit our skill and necessarily to be easy, but we have endeavored on a challenge of adapting to whatever we would get and trying to figure out what we can build out of the dataset would be meaningful. We have not expected that to be a straightforward road, and we thank the professor for meaningful insights about MJF. In the specified timeframe, however, we did not receive timely responses from the dataset holders and stakeholders that could have provided further insights. Nevertheless, we feel that we have managed to utilize the provided dataset well for our topic and purpose, and that we have managed to develop more on course of this project.

In terms of the datasets used, data about the concerts and artists was the main source of information, mainly the information on main genres and subgenres.

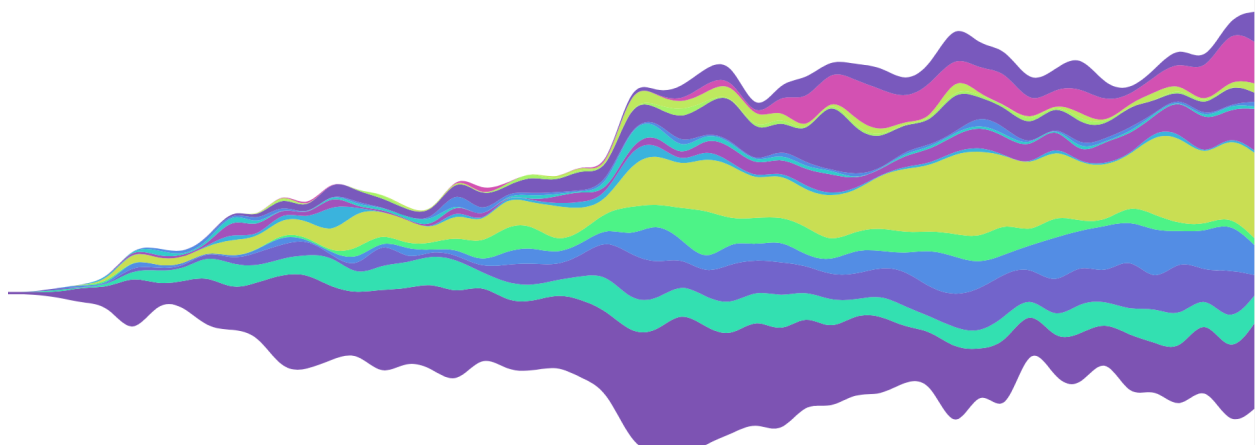
Designs

In this section we provide more insight in the design process. We include the sketches and elaborate on the evolution of the visualization from the idea, to the high level concept on paper to a prototype in code.

Different steps have provided us with insights on how we can improve and what are the pitfalls of such approaches. Often we have made changes due to the richer information provided by lower (more specific) implementations. We will outline changes and deviations from initial plans and ideas with concise explanations.

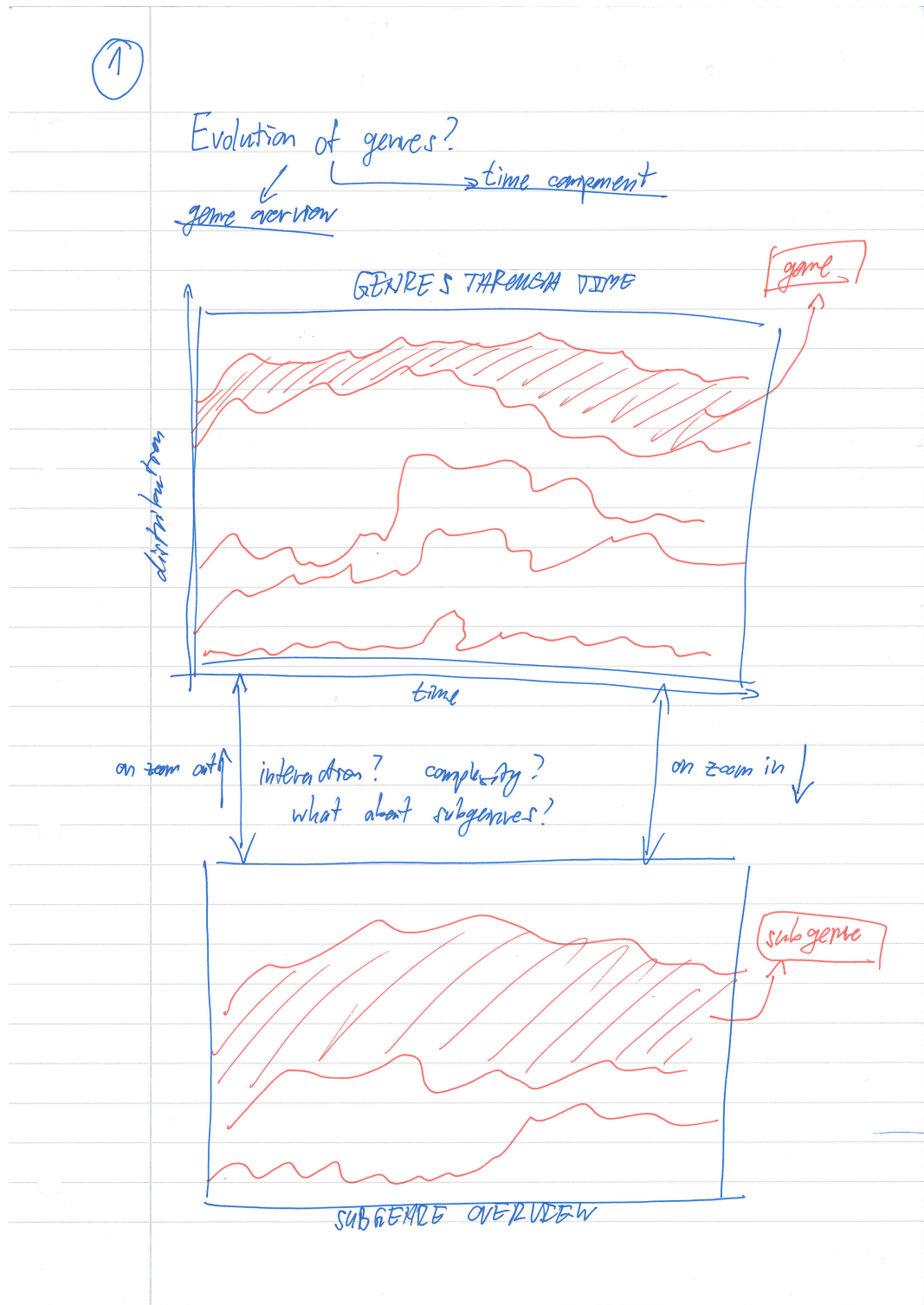
The main inspiration for the genre evolution visualization was the lakeside of Montreux: Lake Geneva and Swiss Alps. To imitate this landscape, as well as to effectively visualize the flow and evolution of genres we have decided to use streamgraphs – since they are a variation of a stacked bar chart. With appropriate effects and use of d3 tools such as interpolation we were sure that we could make this initial concept a reality.

Three key ideas have guided us: compact and clear overview of genres and subgenres, attaining high level of information transfer, as well as a visually appealing result in the end.



Initial high level concept

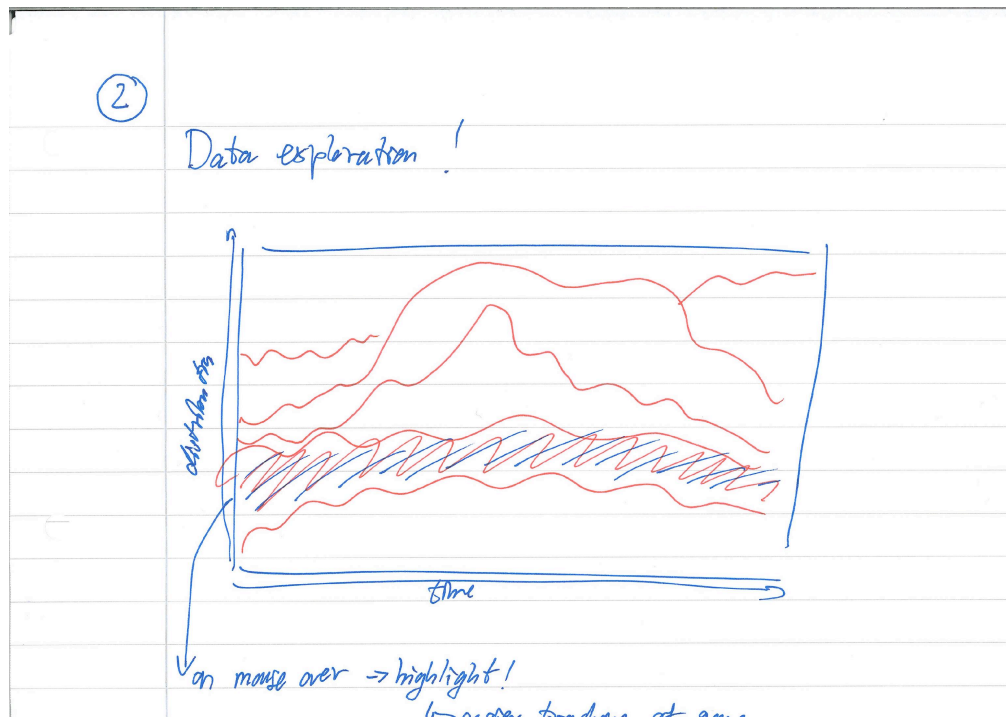
Our initial high level concept has tackled two of the key ideas – an overview of genres through distribution and presence in comparison to other genres, as well as a time component where more than half a century worth of data is concisely presented.



We have decided to make a stacked chart – it would contain all the information necessary about genres for each year. This choice has been made since the goal is to present and compare the same category of data over time, where stacked charts have been one of the effective approaches to take. Nevertheless, there are many variants of charts to explore, and we have considered to use area charts, since data is rich enough on the x-axis to make a reasonable area chart and not use traditional bar charts.

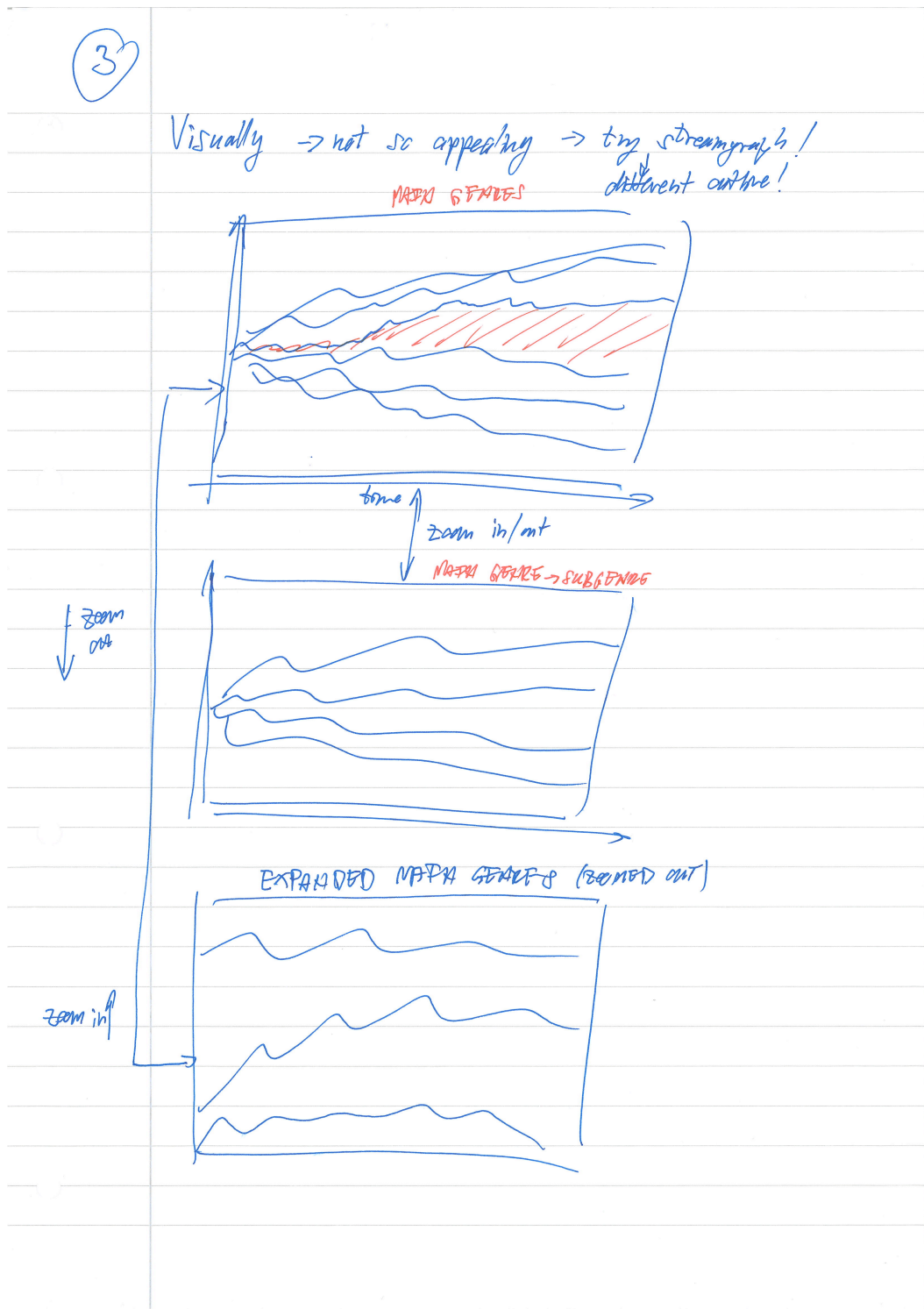
Another issue that had to be tackled is how to represent a more detailed information about genres – its subgenres. To be more consistent we have decided to keep the same chart to manage the focus of attention of the user to one chart that already had many possible details (complying with some of the Ben Schneiderman's principles).

The subgenres would follow the same pattern of genres – a streamgraph, while they would transition on action between each other. The interaction of choice was the zoom in/zoom out events, to capture the intuitive diving deeper into each genre and displaying subgenre information. On zooming in we would show more information, while on zooming out we would return to the default main genres overview.



In order to draw focus to evolution of a single genre, we need to devise a methodology to both capture attention and focus of a user, as well as to mitigate the effect of disproportional genre distribution, where certain genres decay so that they are almost invisible.

We have decided that on hovering over certain flow of genres we need to highlight the whole flow. This way we reduce the information and expose the user more easily the required piece of information. With the ability to focus on selected genre over time, user would be able to explore more reasonably even the streams that are barely present, or that decay to almost invisible proportions. This we plan to implement using opacity, by reducing the opacity of all the other elements, and increasing or preserving the opacity of a selected stream.



Further considering the key point – aesthetically and visually pleasing visualization that would entice users to look deeper in it, after the initial high level prototype using an area chart we have concluded that it might not be the best option available to use. This is a personal judgement and evaluation, and since the inspiration did come from the scenery of Montreux, we did try out different variant of stacked data charts – a streamgraph.

Utilizing the same concepts outlined before are possible with a streamgraph, so this variation would not introduce technical differences in previous considerations.

Another point we have come up with is to add an additional layer of display: on zooming out from the main genres display, we would expand the streamgraph to look similar to an area chart – by using different outline options for streamgraphs. This idea has been motivated by potential gain of visibility of evolution of certain genres, which are not as present during the history of the festival.

At this point we have several interactions with the visualization:

- hovering over a stream of genre would focus it – reduce surrounding opacity
- zooming in on a selected stream – genre, would transition to its subgenres
- hovering over a stream of subgenre has the same behavior as with main genres
- zooming out from subgenres returns to the main genre display
- zooming out additionally from main genres transitions to an area chart
- zooming in on an area chart representation of main genres returns back to an initial overview of a streamgraph

When discussing potential use-cases for this visualization, one consideration was that this visualization might be used on various devices – especially mobile devices such as phones or tablets with touch interface. Zooming in would not be as intuitive to interact with the visualization.

Moreover, even in a Web environment – if our visualization is not standalone in the page, there would be a difficulty of integration if zooming in would be left as is. This difficulty arises from the way that zooming would be implemented – using a mouse wheel action. Since we would use mouse wheel both to scroll up and down the page, performing the same action over our visualization would bring a confusing behavior to the user – breaking the familiarity with the environment, as well as one of the Schneiderman's rules.

Therefore, we have decided to replace the zoom action in favor of clicks. We will consider different combinations of single clicks and double clicks to achieve the same result in a more convenient way both for use on the Web on any devices, but more specifically to be more touch screen device friendly.

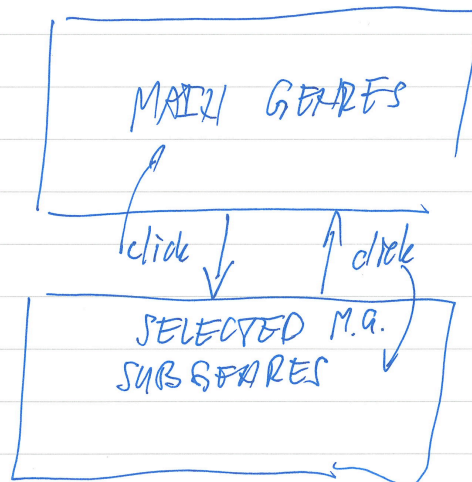
④

zoom → not a good choice → touch devices / portability
→ unstable / web browser scroll
→ overlapping!

→ overlapping events! . zoom . move / hover

zoom → click

→ DISCARD ZOOM OUT
↓
2 EVENTS



Another important thing we have noticed in the prototype implementation are the overlapping events occurring in D3. If two events would occur at the same time, they would often induce an unexpected behavior. Therefore we have explored the options of event blocking and disabling, as well as chaining and using queues to mitigate this effect.

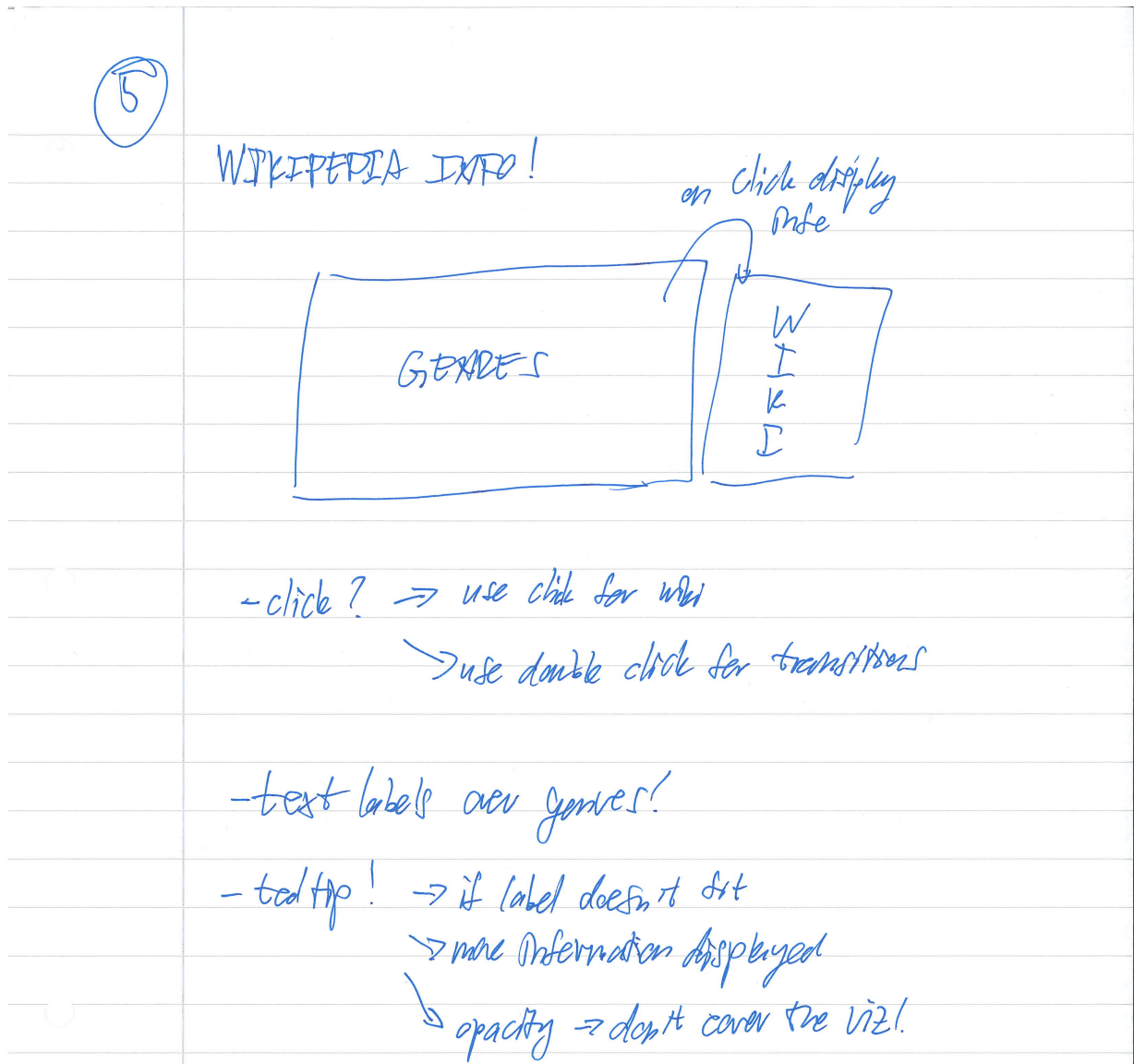
In practical sense, in 4th iteration we have discarded one option in the interaction with the visualization: zooming out from the main genre streamgraph to area chart. Such event adds more complexity and confusion to the visualization, takes away the focus and consistency and does not add more information than we can via other methods. Changing the event from zooming to clicking would also prove very difficult to devise for such case, where single click/double click can be combined for 2 transitions.

To overcome the lack of a general and clearer display of genres over time, we are going to implement tooltips that will appear while hovering over certain genres. Therefore on hovering on a stream of genre or subgenre:

- the opacity of other streams becomes lower
- a tooltip appears, showing more detailed information about the piece of a flow of genre on which the mouse is currently over:
 - genre name – in case it is not visible on the streamgraph
 - year
 - number of concerts that year of selected genre

Additionally for the tooltip:

- It has to be semi-transparent, not to block out the view of a streamgraph
- We need to take care of edge cases, not to display out of visible part of screen
- Incorporate some visual elements of genre, such as display color



Furthermore, at 5th iteration we have considered adding an additional component to complement our visualization. Since we are showing and evolution of many genres, of which most people know basic information about, we will show a Wikipedia article of the selected genre.

The practical considerations that have changed about the Wikipedia is the placement of the widget. Initial placement was conceived to be below the visualization, not to interfere with the flow across the page and to have a more fluid and complete design. Albeit, the focus of the user and practicality would suffer greatly – user would have to scroll to see more information about genre, and depending on the screen size and resolution or input device, that would present a tedious task. To mitigate this issue, we made a compromise on design and added the widget on the right side of the visualization itself.

In practical sense, since Wikipedia widget will probably use Web API calls, we need to consider how and when the content will be loaded:

- on hovering over genre: this action is very frequent, same genre is commonly chosen, as well as playing around the visualization is possible. We cannot interpret hovering as an intention of the user to see detailed content
- on mouse click: this is a deliberate action by the user, thus we select this option. To transition to more or less detailed streamgraphs we will then use double clicks.

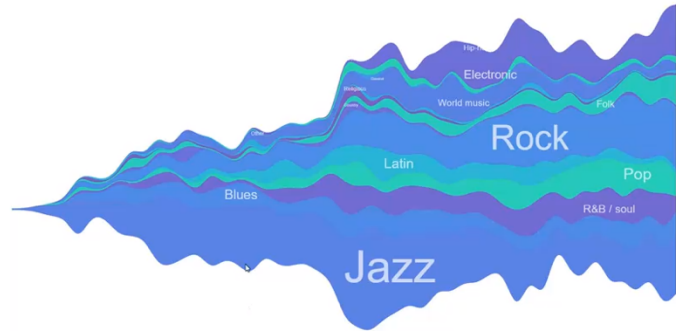
Implementation model

During different iterations over high level conceptual model outlined previously we have used practical implementation using D3 in order to test our hypotheses and look out for practical pitfalls that might arise from our choices.

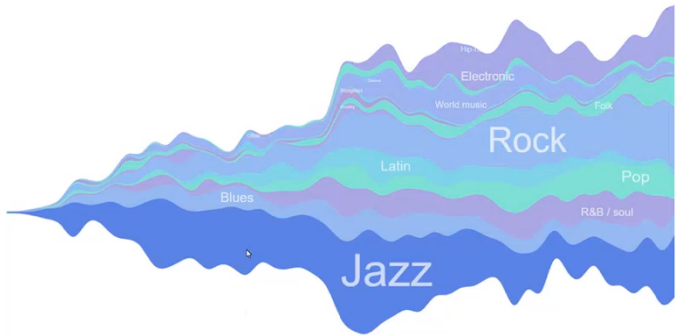
We have managed to implement our high level concepts completely, and we will demonstrate in this subsection via descriptions and screenshots how such implementations work using real coded examples.

Conceptual flow of interaction

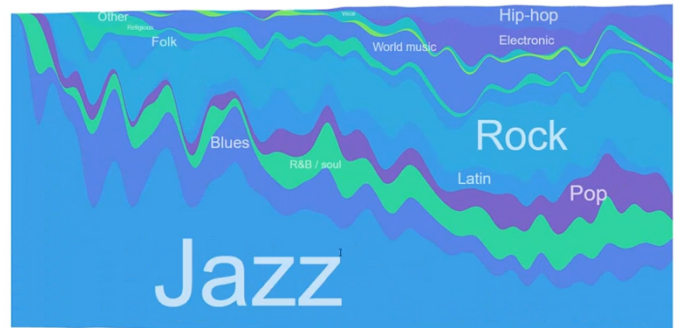
1. Complete visualization is shown



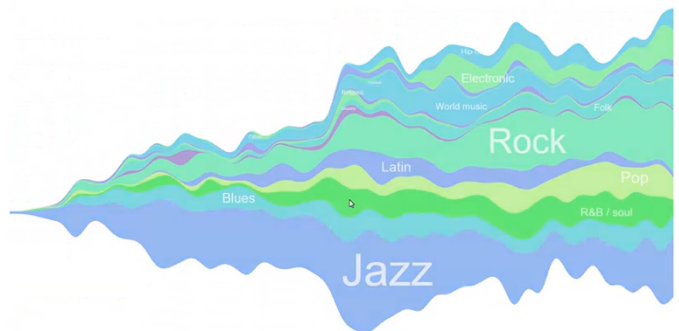
2. On mouse hover a flow is highlighted



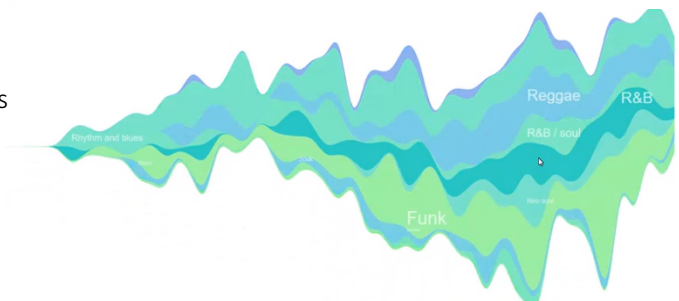
3. Zooming out shows a wide overview



4. Zooming in returns to the overview
Genre is selected by hovering over



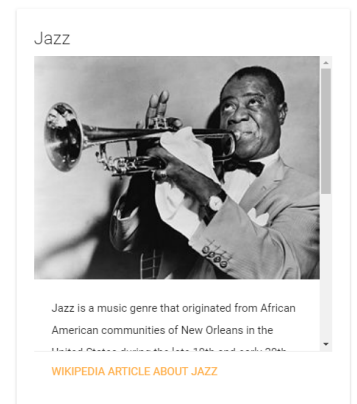
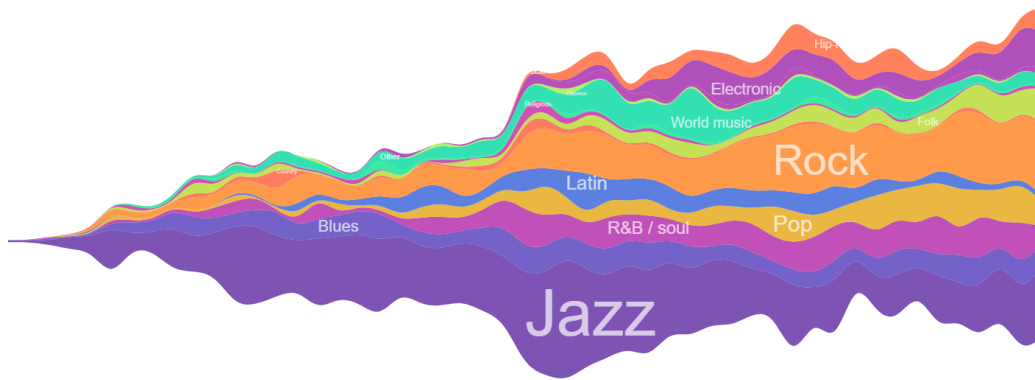
5. Zooming in makes a transition to subgenres
Interaction elements stay similar
Zooming out brings us to previous step



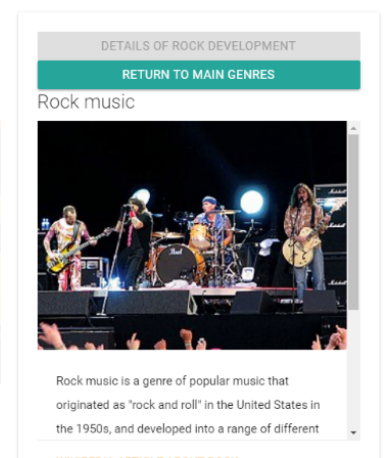
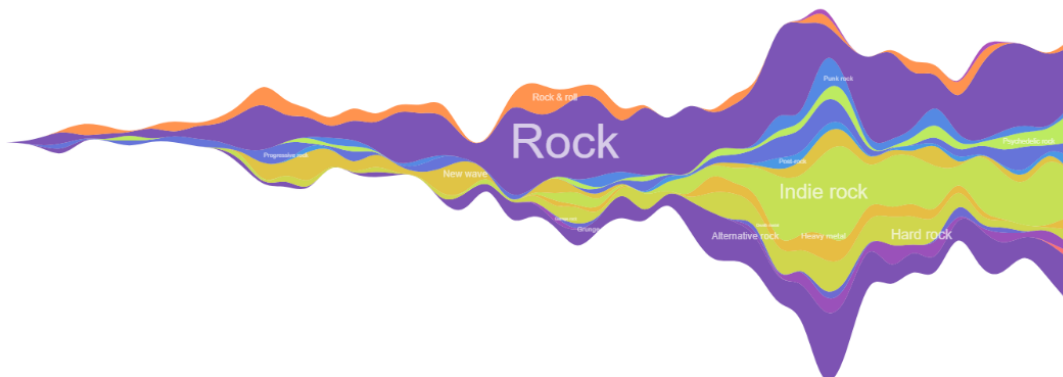
As we have outlined before, during the course of such tests, we have made several changes to our final implementation model:

- we have discarded mouse wheel/zoom events in favor of clicks
- we have discarded step number 3 in Conceptual flow of interaction
- we have learned more about the behavior and usability:
 - in terms of d3 events, especially overlapping events
 - events queuing and disabling while other events happen
- learned more about practical interaction with the visualization
- noticed the pitfalls of bad positioning of Wikipedia widget

Wikipedia widget itself needed to be constrained to the height of the visualization. Unfortunately that meant placing the content of the widget in the box that would contain a slider. This is a compromise that had to be taken to have a consistent visualization, since articles and images obtained from Wikipedia vary in length.



An important step to consider was the user orientation when accessing selected main genre subgenres. Without any context user would not know in some cases what is the main genre from which he entered from, as well as not knowing intuitively how to return. For those reasons buttons and additional information have been added in the Wikipedia widget, to add more strength to the context of the visualization.



Differences from the initial proposal

We have expected that we will have more expert input in our case, so that our visualization can target a specific need from a musicologist. Since we did not even obtain the most recent dataset and the time constraints of the project, we had to deviate from this initial plan, both for reasons of complexity and round trip time of the information and feedback. Then we focused purely on how to present such data to everyday user curious to learn more about Montreux Jazz Festival and its rich cultural development over the years through music genres.

Visualizing interactions between genres and artists

Design

Having completed our first implementation of visualizing the evolution of genre through the history, we turned our focus to presenting the interactions between different genres and artists in a time-series representation over the years. The concept behind this second visualization had two clear objectives – a representation of the interactions between different genres, more specifically which genres tended to be played at the same concerts, and showcasing which artists have over the years worked or interacted with one or multiple genres.

To accomplish this task, we initially attempted several different tree and graph representations which we found most suitable to represent the numerous links and interactions between genres and artists. Through a high-level design approach and sketching we initially opted to represent the data in a circle graph which provided flexibility for the numerous artist nodes and their individual interactions with genres.

However, through testing noticed that this kind of a representation led to a lot of cluttering where there were numerous artists who only interacted with a single genre and thus the lines within the graph had a lot of nearby singular connections which diminished the ability to really see the interactions between different genres. Through additional research, we were able to design and implement a form of a hybrid circular graph where nodes who are only connected to one other node in the circle (i.e. artists playing within only one genre), would be placed on the outside of the original circle graph on a larger, encapsulating outside circle graph. Through this approach, the visualization benefitted not only from increased clarity and information gain, but an aesthetically pleasing array of links flowing into and out of the main nodes representing the different genres.

This improved circular graph representation draws the user in by allowing them to follow the different emerging paths from the nodes representing genres as well as seeing which genres are more or less common. This type of representation was able to signify to a great extent the information encapsulated by our previous visualization by showing the evolution of the presence of individual genres through time. Furthermore, we were now able to represent genres which tend to rarely combine with others at the jazz festival, most notably Electronic music, by showing this collection of nodes completely outside the main graph, adding not only to further information gain, but quite an intriguing concept for the user as well.

Interaction

A major issue for this visualization however remained the design and implementation of interactions as any changes would be computationally heavy. Nonetheless, through iterative design and testing, we reached the following set of functionalities and interactions to make the visualization more engaging and useful:

- Clicking on any node will extract it from the main circular graph and focus on it and its connections. This allows the user to explore the many connections of each genre as well as all of the genres an artist has worked with
- Zooming in or zooming out of the graph to explore it further in depth will appropriately change the font size of node labels for better visual aesthetics and clarity
- Clicking on the 'reset' button in the top right will reset the view of the visualization to its original position as an easy way for the user to reset their position
- When a node is selected, if the user clicks on black space or the 'reset' button, the visualization will first zoom out to its original position and then reposition all of the nodes in the graph. This is extremely visually pleasing and clearly displays to the user the interactives as well as the changes within the graph
- Clicking on a year on the time-series tooltip at the bottom of the visualization will display the data for the relevant year and rebuild the graph representation

All of these interactions lead to an improved user experience as well as provide engaging and valuable means to explore the dataset in depth. The goal of these visualizations is also to further highlight the pleasant and enjoyable manner in which our data has been presented, showing the user the power of data visualizations as well as their intricate beauty.

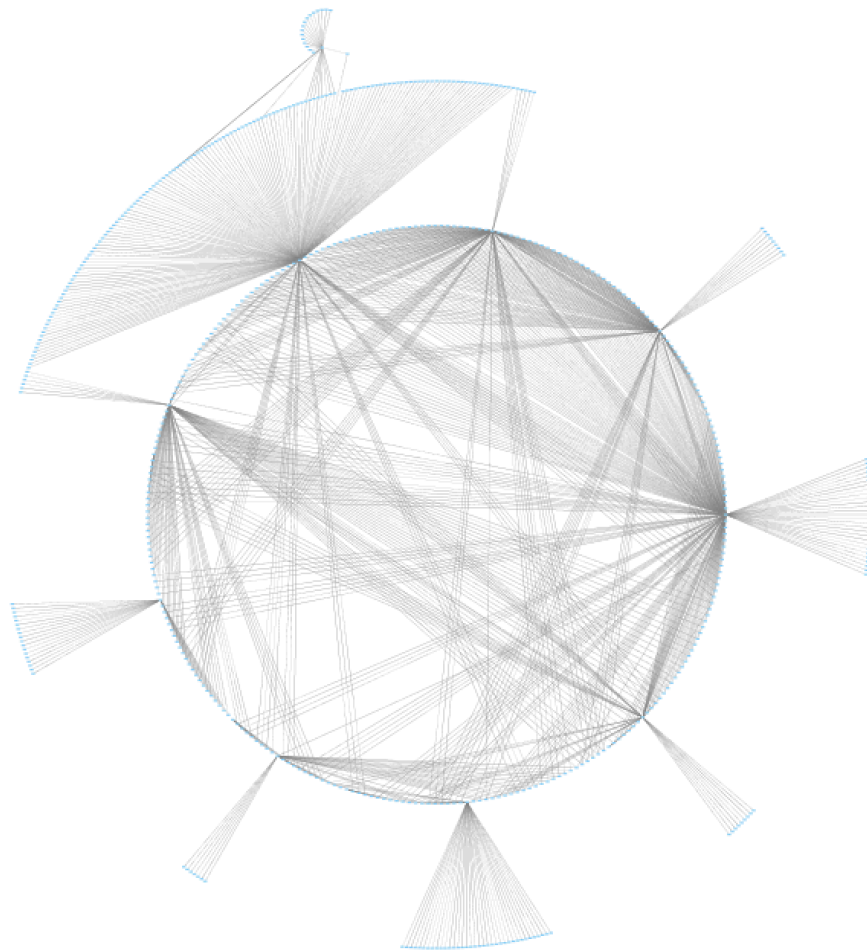
Furthermore, we also provide the user with an easy to use search bar to locate their favorite artists within particular years. The reason we chose to make the search bar year specific is that our visualization works in a time-series manner and when searching for an artist, the user would nonetheless have to specify the year they are interested in to obtain the relevant graph representation for it. Additionally, we feel that the above-mentioned interactions are highly suitable for this search bar functionality as the user can benefit from all of the visual and contextual information without the need to find the specific artist's node themselves in the graph.

Implementation progress

We will now show how our implementation progressed from a very basic graph representation, to a fully functional, interacting and eye pleasing visualization.

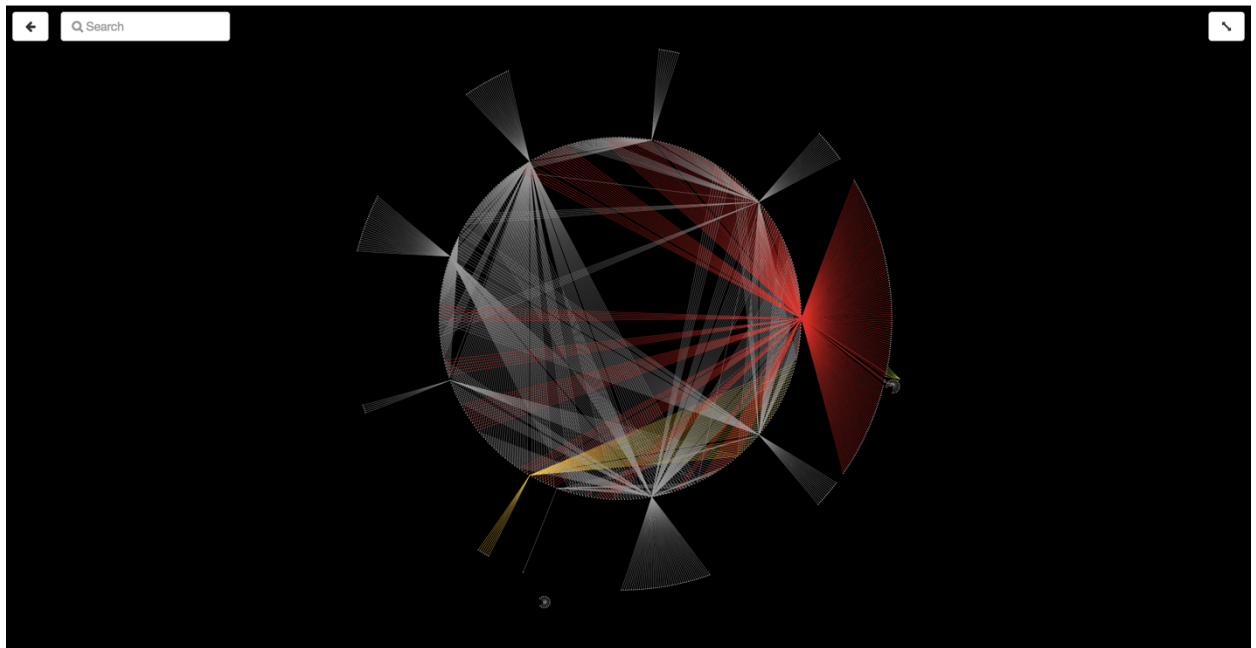
We have chosen to use Cytoscape.js, an open-source graph library written in JS which allows user to interact with the graph. Firstly, we had to preprocess our dataset and generate files for each festival, containing all genres, bands, musicians and connections between them for that specific year.

After that, we used a desktop version of Cytoscape to generate positions for each node (genre, band or musician). This application offers several numbers of layout algorithms and we tried all of them and choosed to one that we thought as most appropriate (Circular layout). In picture below we can see how graph looked like in Cytoscape application after applying the layout algorithm.

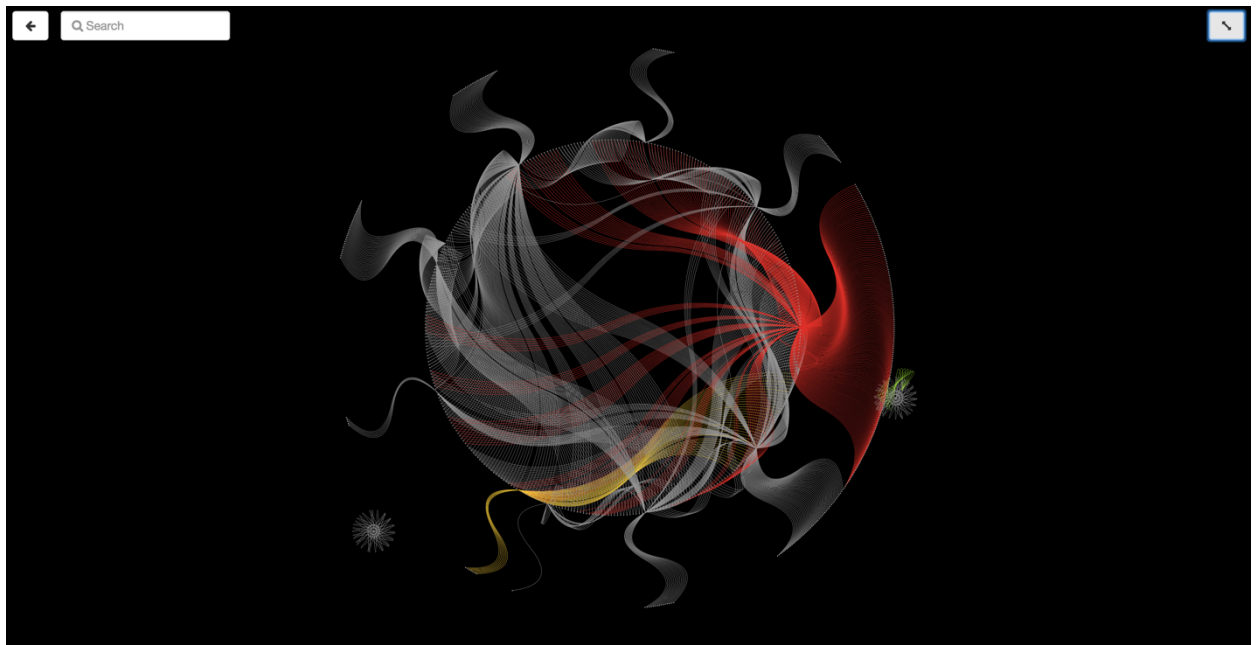


After applying layout algorithm to all files, we now had updated files, with nodes position for each year.

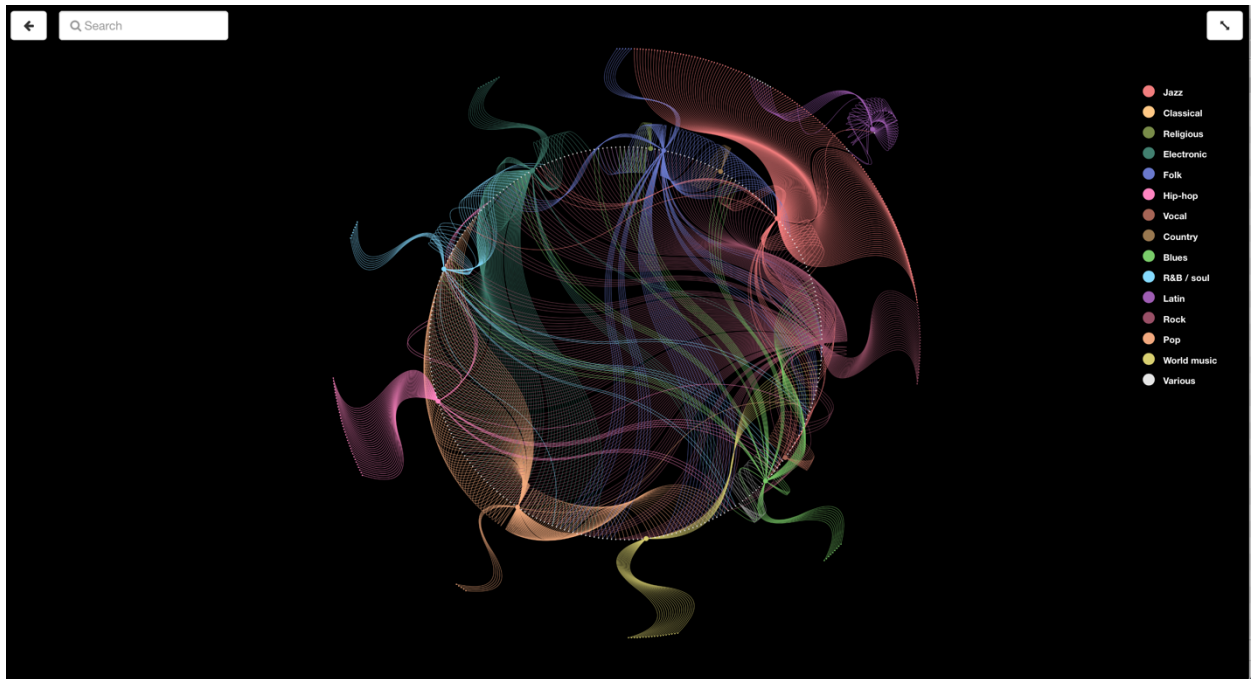
We then started to work on our interactive version of graph. We started to play around and looked for the most pleasing visualization. Some of the iteration steps can be seen below.



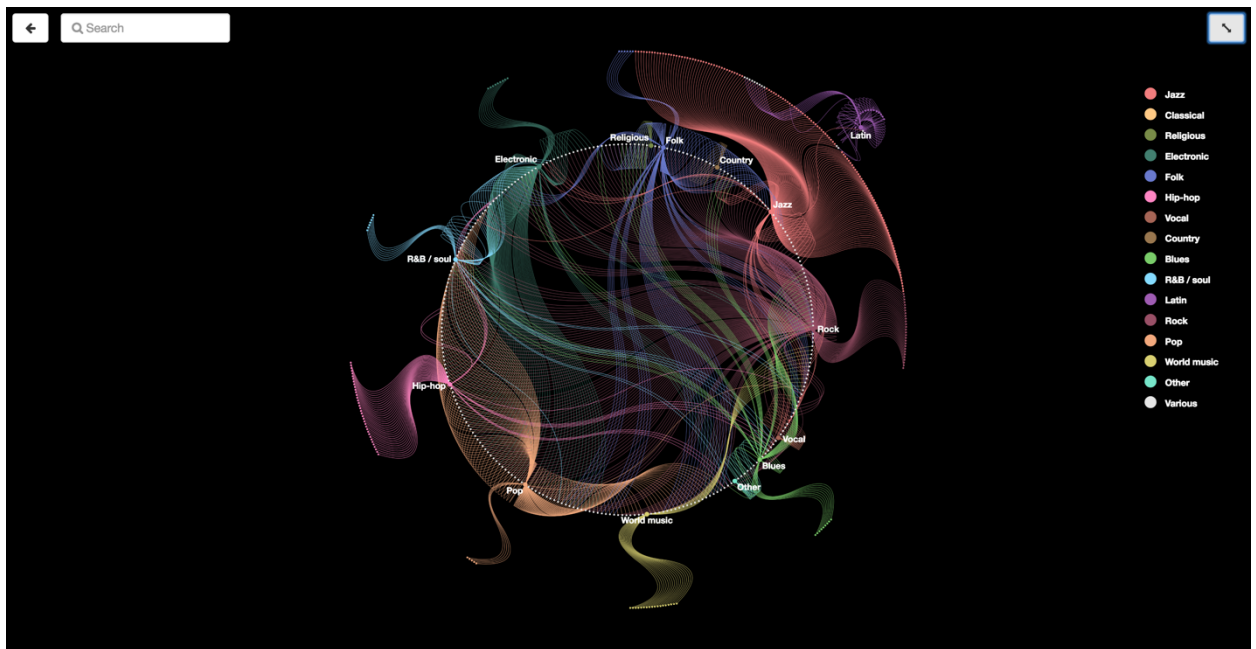
Adding colors to nodes and edges



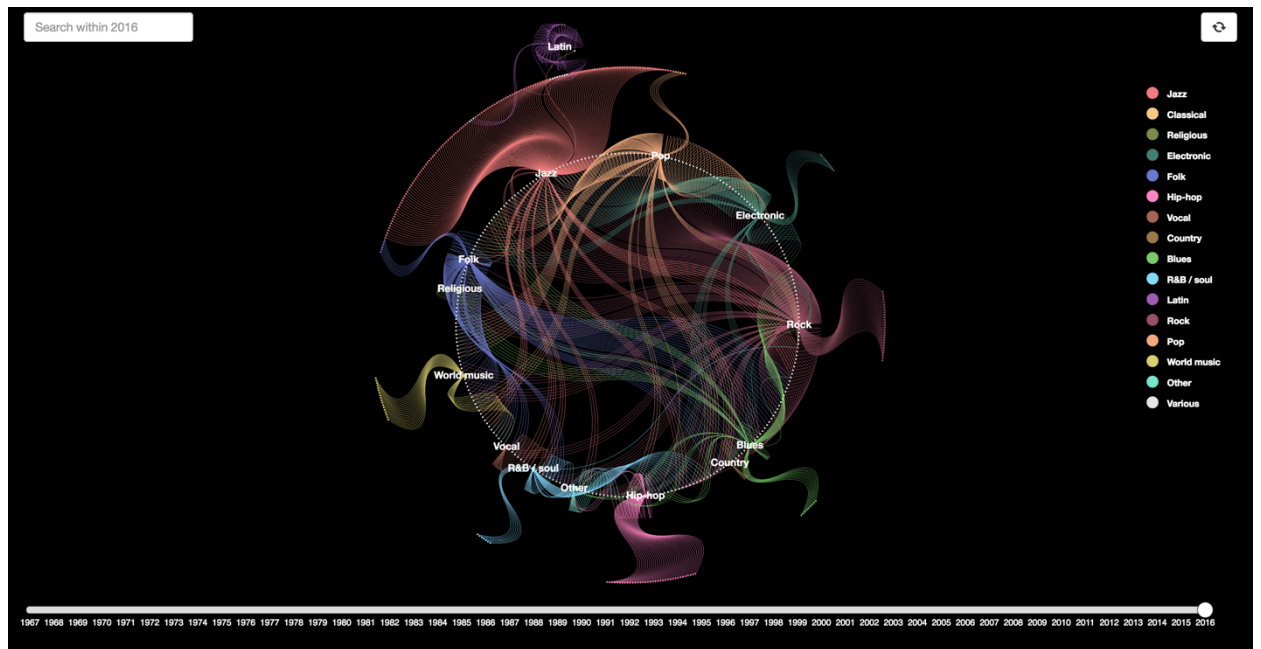
Making edges curved



Choosing different colors for all types of genre and adding legend



Making genre labels bigger



Adding slider to be able to change years

Implementation

In this section we will further discuss implementational details, based on the final implementation model in the previous section.

We will cover technical details: libraries utilized, the functionalities and reasoning behind it, as well as the overview and uses of the visualization.

Technical details

We will start by an honest remark that we have explored the option of visualizing with existing libraries and wrappers around D3. As expected, this approach lacked in any customizability which we required for implementing our visualization – we have decided to take the harder but more fruitful path. We have developed our visualization using the latest version of D3.js library (D3v4), while trying to use the best practices given by ECMAScript 6.

We process our data in Javascript. The website itself is utilizing JQuery and Materialize library (<http://materializecss.com/>).

To show information about genres, we utilize simple wrapper around Wikipedia/DBPedia API (<http://okfnlabs.org/wikipediajs/>).

Visualization is utilizing a helper library for D3 used for calculating the best size and positions for the text labels inside defined areas (<https://github.com/curran/d3-area-label>).

For second visualization, we used graph library for analysis and visualisation Cytoscape.js (<http://js.cytoscape.org>). Also we used desktop version of Cytoscape application (<http://www.cytoscape.org>) for easier data manipulating.

Finally, the website is hosted on github pages, while the data is stored in private github repository to comply with the NDA.

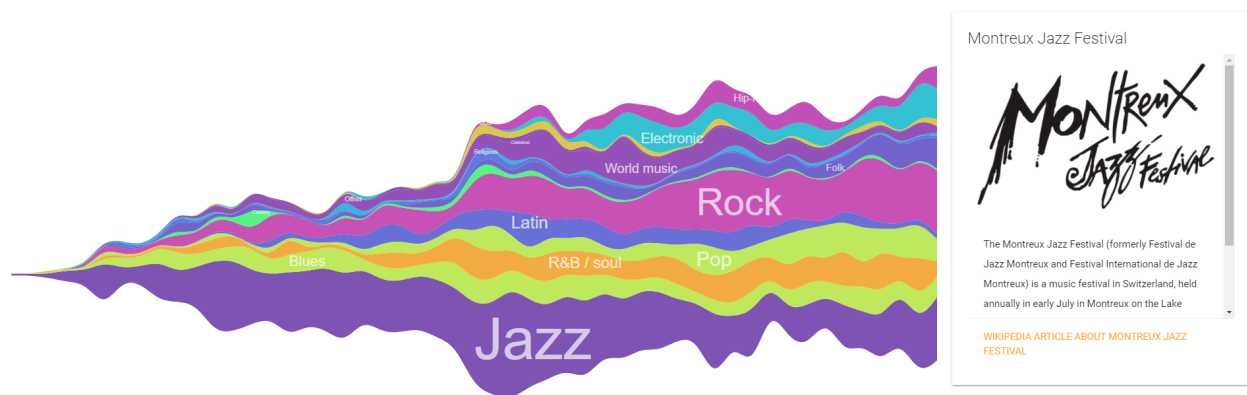
Overview and Functionality

We will describe the intended and possible usage of our visualization through screenshots of the actual page. More information would also be available in the screencast. The main features are: genre and subgenre overviews over time, detailed information provided by the tooltips and Wikipedia information about selected genre.

Initially displayed visualization

Initial visualization shows a streamgraph of main genre presence over the years at Montreux Jazz festival. It is intended to spread over the complete width of the Web page display to capture the fluid flow of genre evolution. In order to transition smoothly between the years interpolation is performed between the years to provide a smooth and flowing transition.

On the right hand side there is an element containing Wikipedia information obtained using Wikipedia/DBPedia API. It contains title, an image if defined, short description and a link towards Wikipedia if the user would like to learn more about the genre. Clicking on the link opens a new tab instead of loading it in the same page. The height of the container is roughly similar to the height of the SVG visualization to integrate well together, since as previously mentioned we have concluded that for seamless use such information needs to be accessible and within the visual range of the user. For those reasons it is necessary that a container for Wikipedia content has a scroll bar.



The containers for both SVG and Wikipedia are reactive to change in size, although some tweaking needs to be done to ensure that the Wikipedia container is displayed properly after certain minimization of the size.

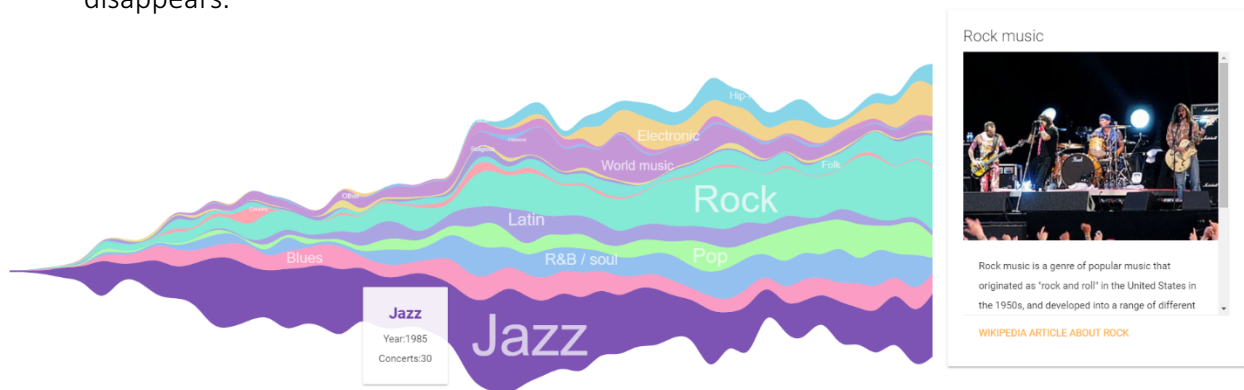
Interactions on main genres display

Several interactions are possible on the main genres display:

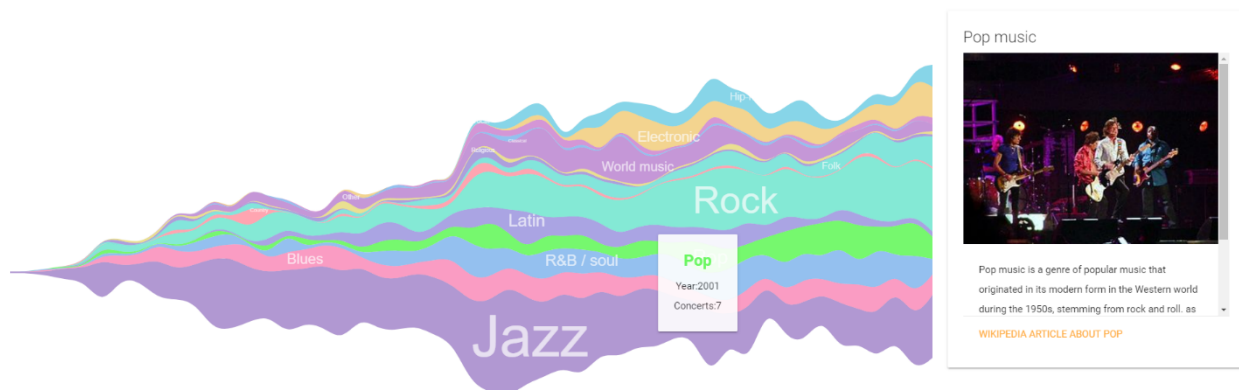
- highlighting the genre flow
- displaying more information about genre in certain point of time
- displaying Wikipedia information about selected genre
- transition to subgenres of a selected genre

To highlight the flow of genre, it is necessary to hover over the genre to explore. It highlights the genre and brings up a popup which displays the genre name in the same color as the flow, as well as the year and number of concerts having that genre at that point in time. This also enables to read genres with small or less visible labels

The popup is semi-transparent to allow genre evolution observation without completely blocking some portion of the display. Since text and chart are separate elements of an SVG, an effort has been taken to make all the functionalities as smooth no matter whether they have been performed on a textual label of genre or the graphical elements of a streamgraph. On hovering out of any genres, the popup disappears.



On a single click on the either genre label or area of a streamgraph, a request is sent to Wikipedia API to show more details in the info-box.



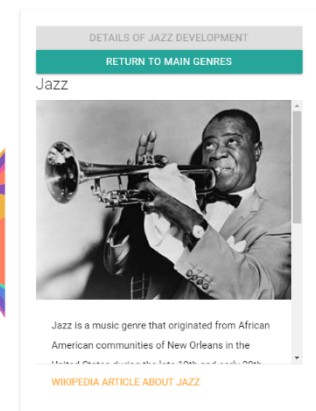
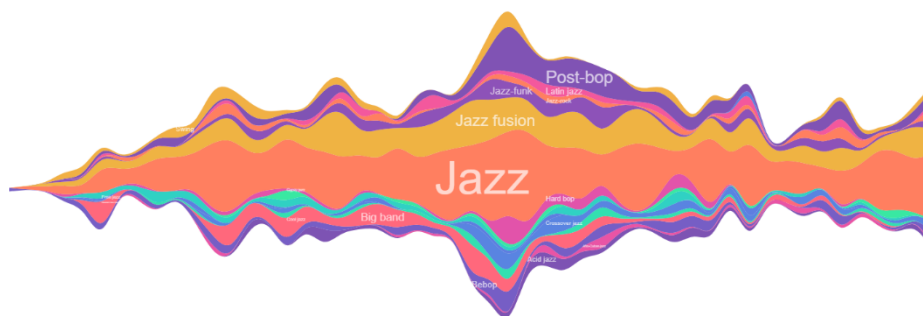
In order to transition to more detailed graphical display of subgenres evolution of a selected genre, it is necessary to perform a double click. The effects are double:

- Wikipedia box refreshes to be about the selected main genre
- Graph seamlessly transforms into another streamgraph, displaying subgenre information in a similar fashion as the main genres
- Wikipedia box has added elements: name of the main genre and a button to return to the main display – to enable easier usage and provide context

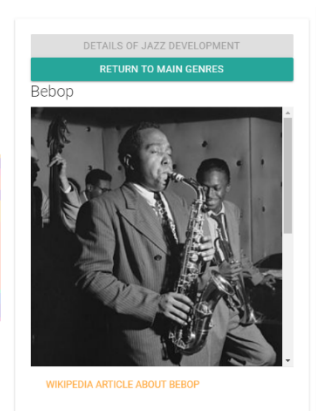
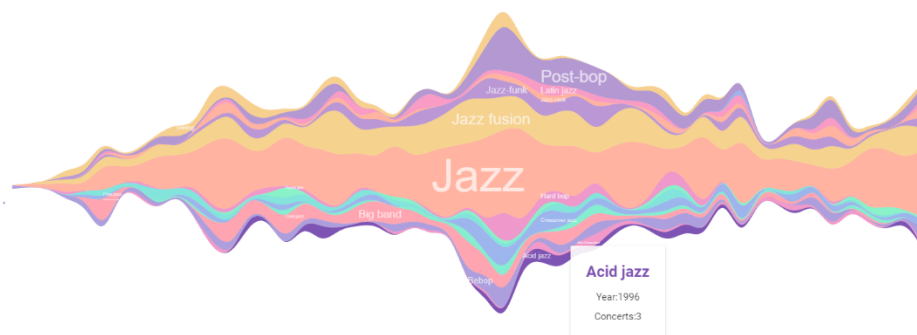
Interactions on subgenre display

As previously mentioned we have transitioned with a double click to a subgenre display. The Wikipedia information box now has two additional elements:

- *Details of \$main_genre development* – name of the main genre
- *Return to main genres* button – which resets the display back to main genres



The interactions are otherwise similar to the interactions possible at main genres display, keeping consistency in design and display – by having user apply the same knowledge about the graph as he already learned (utilizing one of the Schneiderman's rules).



By either clicking on the button in the Wikipedia box, or double clicking on the subgenres to invoke familiar transition, we are returned to the main display.

By implementing only features that have went through several iterations and evaluations in higher level prototypes, we have managed to construct a visualization using the least possible effort that would happen if we had to make significant corrections at this point of time with the detailed implementation.

Evaluation

In this section we discuss the results we have obtained, the data insights we have achieved, pitfalls to look for as well as the prospect for the future improvements and work.

More details about practical aspects are in the code and comments to point out to more practical details and insights of implementation in Javascript or D3.

Data processing can be observed in the code – it uses the latest D3 and ECMAScript 6 practices. For practical reasons and the NDA, we did not disclose some features and processing of the elements present in the data. We did however document the code by detailed comments.

Data insights

We have started this project without many knowledge and insight in Montreux Jazz Festival. We very much liked the prospect of working with an unique dataset and some of us have experience in music. Accordingly, we did not know what to expect as an outcome of this visualization – the evolution of genres.

We can observe that over the years there are two tendencies:

- number of concert has increased since the inception of the festival
- the diversity of musical styles and genres has started in the 1970s

From the 1970s we can observe emergence of the most present and constant genres: Jazz, Rock, Pop, Blues, Latin, Electronic, World Music and Hip-Hop. Other genres evolve and are present in different measure depending on the year, such as Country music, which was popular in the late 1970s, or the emergence of World Music in the mid-1990s to early 2000s.

Overall, we can notice the growth of the festival, both in terms of concerts, as well as in the cultural and musical richness. We can observe the musical trends, as well as more subtle details – in terms of same tendencies in subgenres. We leave to our users to explore all the intricacies of the dataset, and we hope that this visualization can provide a good insight in rich history and hopefully prosperous future of the Montreux Jazz Festival.

Future work

This visualization has many possible improvements that can be implemented. A more detailed insights could be extracted from the dataset and displayed in the tooltips. Another possibility would be adding on the x-scale (time scale) a reel of pictures from previous festivals to be able to convey additionally the atmosphere of the festival. Unfortunately, we could not obtain any pictures from the data we were presented with currently.

Some improvements could be done on the visualization side, improving the technique of label calculation, position and color.

The data processing technique is offloaded to the client, with a possibility for performance issues on low performance devices. In practice, we have tested the visualization both on mobile phones and lower end devices without any issues, but there is a delay until the visualization is set and loaded. To mitigate this issue it is possible to have a dedicated server to process the data and then to push the final result to the user side, or using a database system which would execute a query returning only the necessary results, completely removing the need of any data preprocessing on the client side. For the scope of this project, as well as demonstraing the skills attained in ECMAScript 6 and D3, we have decided to keep the processing step at this specific level.

With further insights from experts and musicologists, we could possibly create an even better and insightful visualization that would be not only visually interesting to non-experts, but potentially provide deeper and practical insight into the history of the festival.

