AudioForma

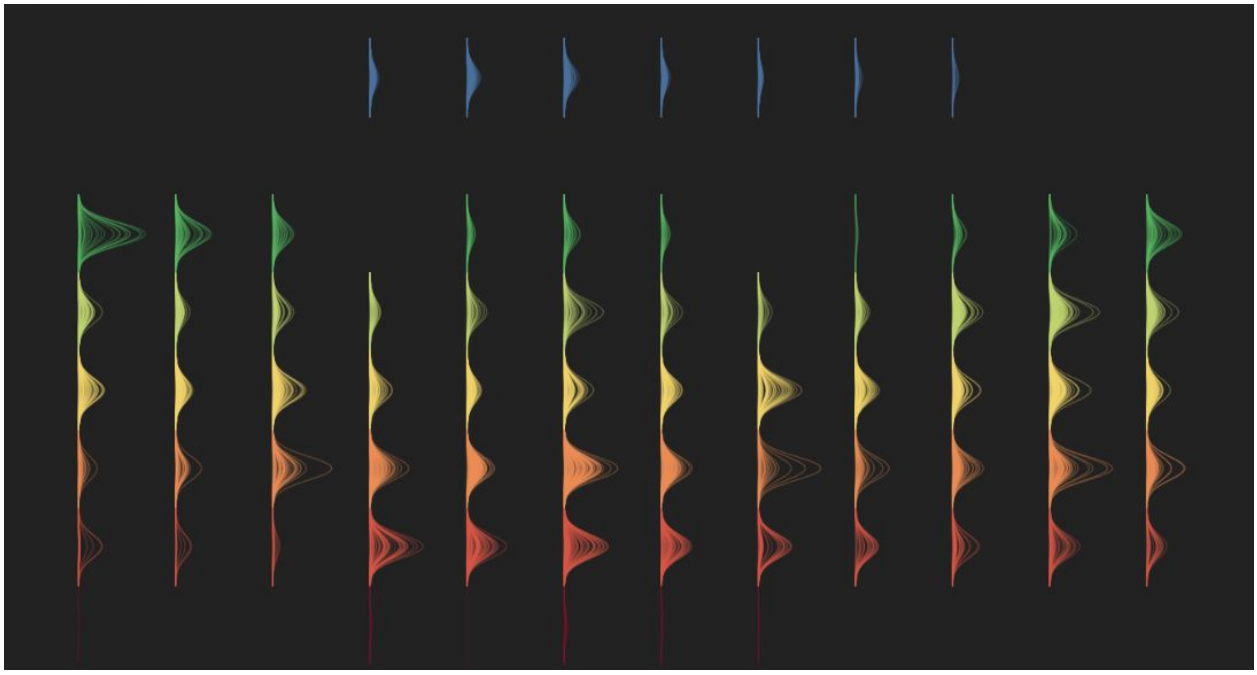
Process Book

Ning Chen, Jasper Croome, Rebecca Lantner

CSCI E-14A: Building Web Apps for Data Analysis

Harvard Extension School Fall 2019

Contents



Introduction

Listening to music is a powerful, sensory experience. A song can move you to tears, brighten your day, motivate you to achieve a goal.

As data-savvy designers, we asked ourselves: how can we translate the auditory power of music into a visual experience? In other words, what is the shape of a song?

We began this project with the concept of a “song fingerprint.” Listening to “Walk This Way” by Aerosmith *feels* very different from listening to Beethoven’s “Moonlight Sonata,” and by that logic, we guessed that the songs would *look* different, too.

With AudioForma, our goal was to create a music visualization web app to add a visual dimension to an audio experience.

Read on to explore our design process.

Design Evolution: Song Fingerprint

In the beginning, there was Librosa.

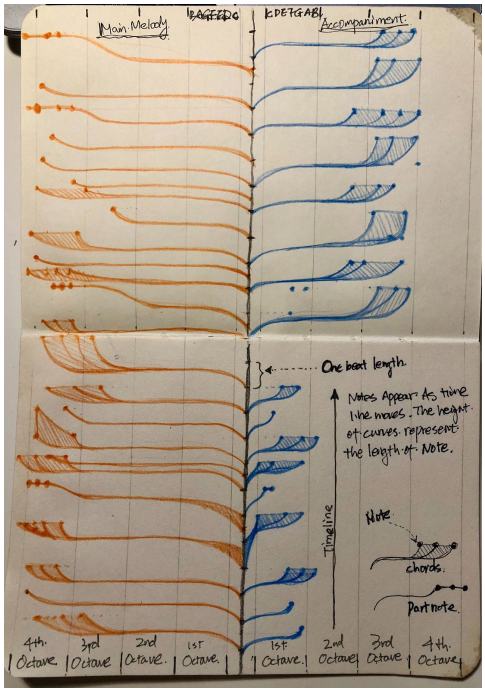
We began by playing around with the Librosa audio analysis library in Python, understanding what data was available for a song, and how we could process and convert audio attributes into visual channels.

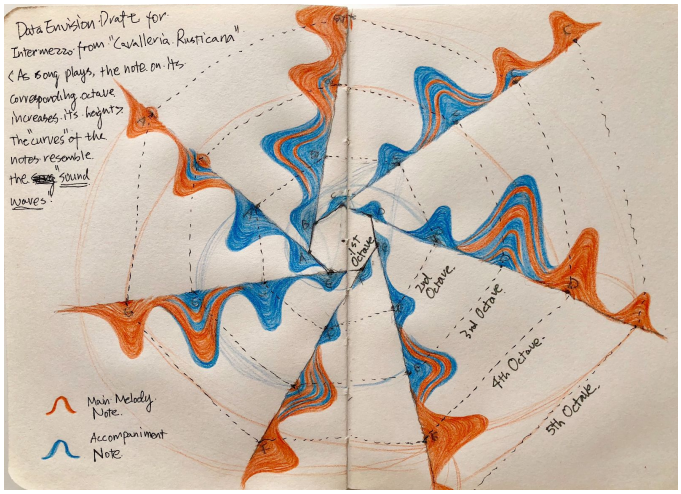
Jasper wrote scripts to manually run audio files through Librosa and store the output in CSV form. The level of detail was extremely granular -- 128 audio points every millisecond, containing note-level attributes like time played, intensity, octave, and others.

We briefly considered using MIDI instead of Librosa, because MIDI files contained additional attributes like instrument played, but we ultimately decided against it. Librosa seemed like a more flexible solution, as it allowed us to retrieve data for any song we wanted, rather than rely on a stock supply of sometimes-untrustworthy MIDI files sourced from the Internet.

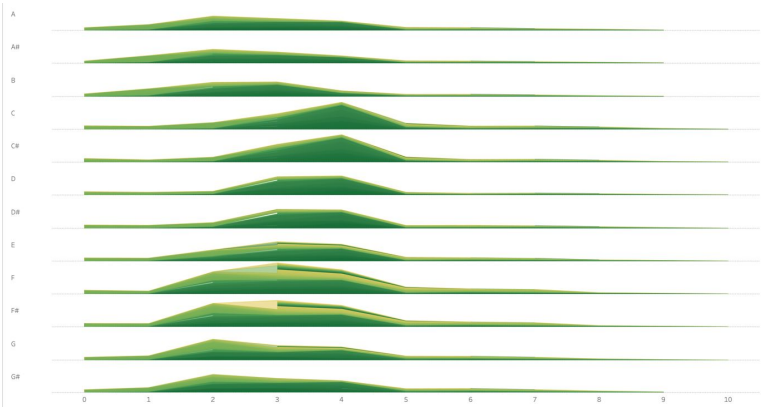
Using Librosa-processed audio data, the song fingerprint was born.

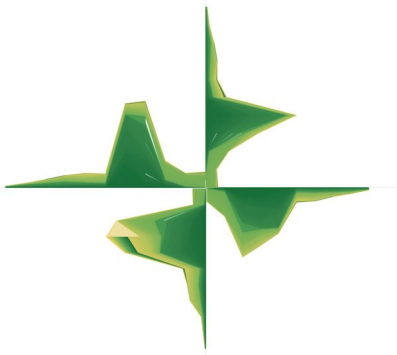
Ning began with sketches.



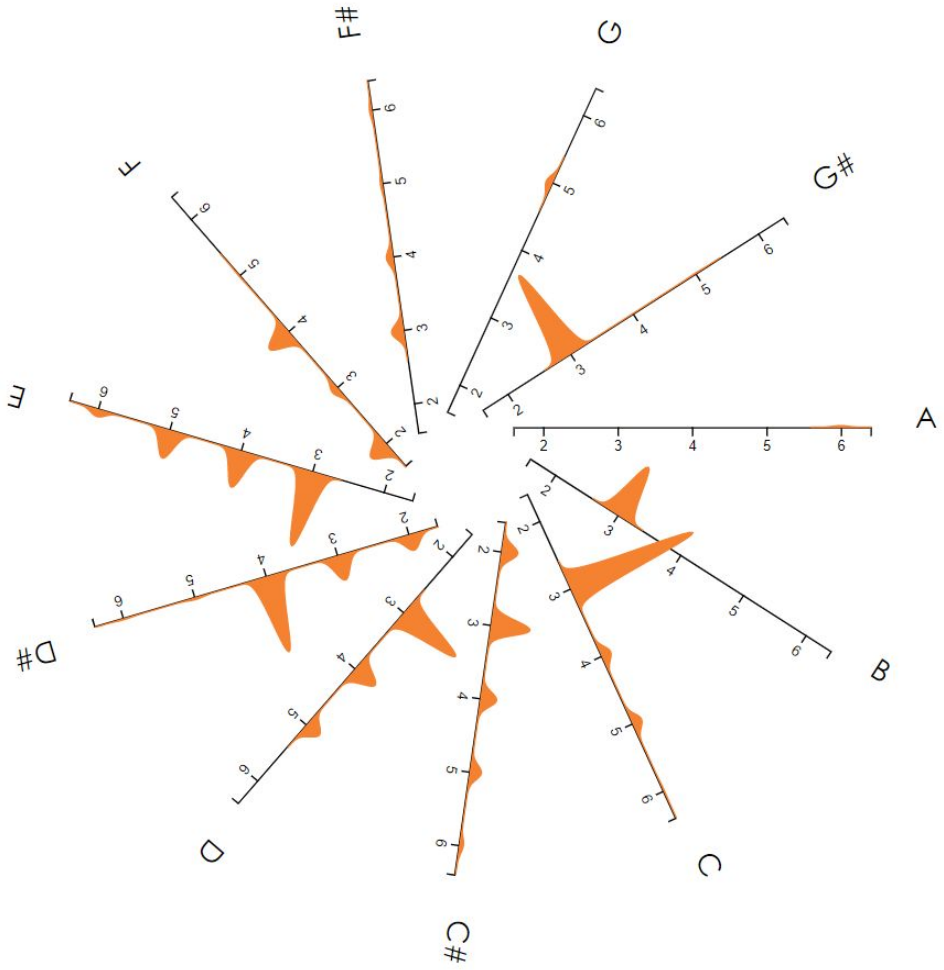


Jasper created prototypes in Tableau to ensure that, using actual audio data, we could achieve our on-paper vision.

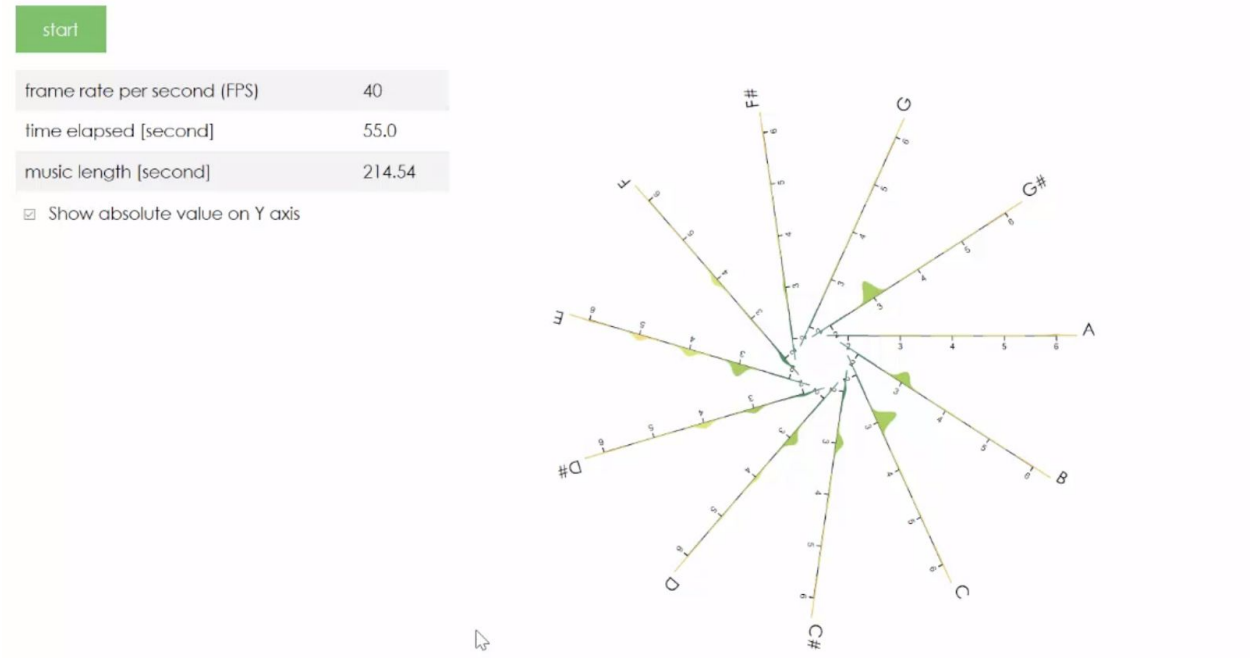




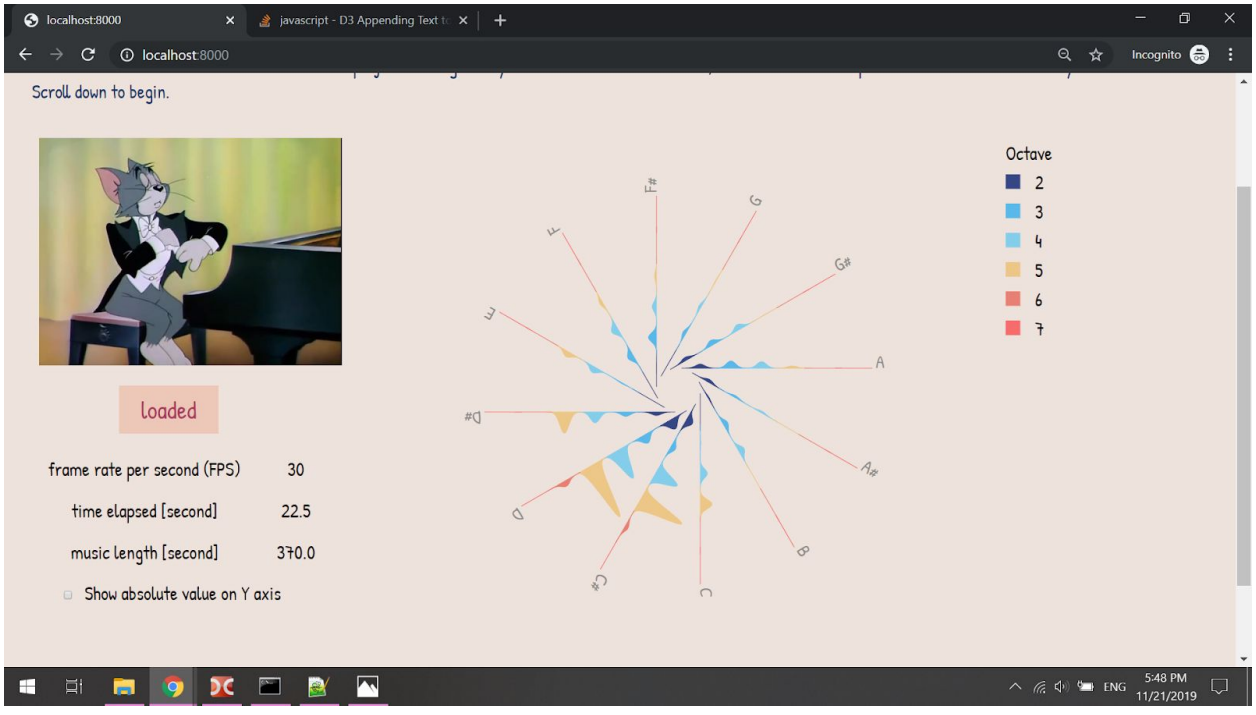
Then the D3 implementation began.



Alongside the frontend work, we hooked up the backend to play the visualization in time with the audio. The image below is a freeze frame of the notes and octaves active at the 55-second mark of a song.

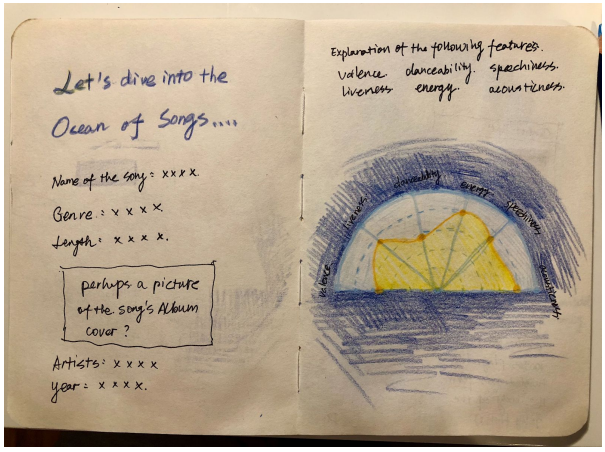


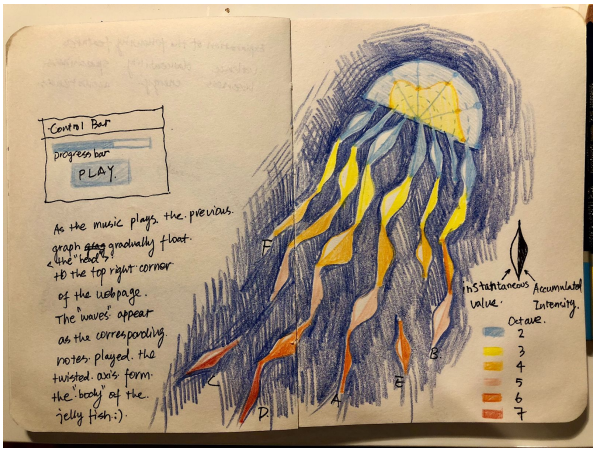
More updates followed, focusing on the UI.



While we were happy with the radial design, we perceived a shortcoming in its inability to capture the shape of the song in its entirety. The real-time movement was neat, but there was no lasting, holistic “fingerprint” once the audio stopped playing.

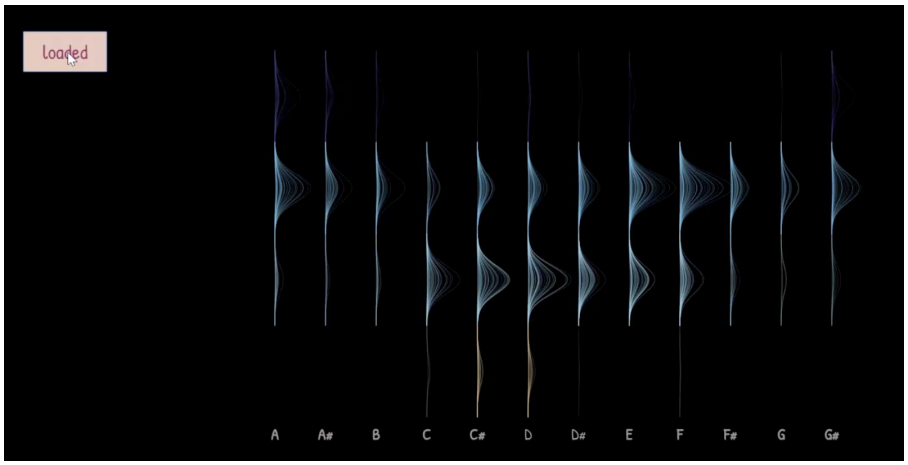
So Ning returned to the sketchbook.



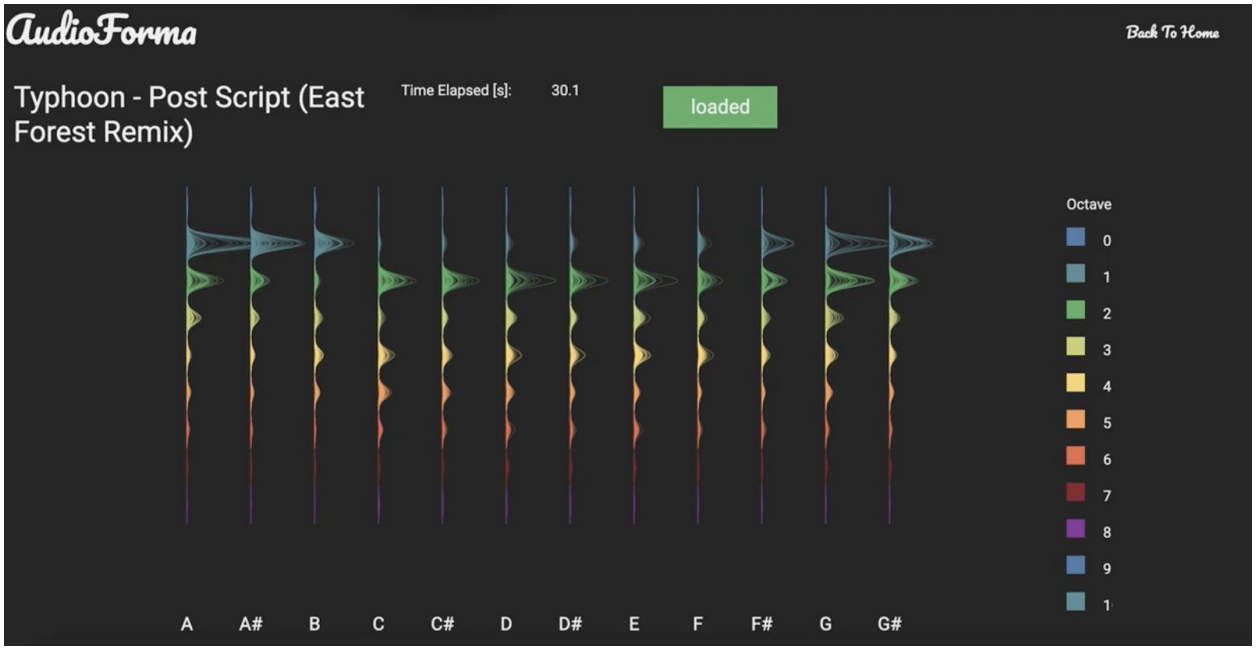


Around this time, Jasper made a data breakthrough. Rather than relying on Librosa, we were now able to hook up our app to the Spotify API and bask in its wealth of audio info and 30-second song samples.

Integrating this newfound Spotify data trove, Ning adapted her visualization to account for a song’s note history. Below is the result of 30 seconds of audio, each curve representing a note’s intensity at a given time and accumulating into a stacked, Gaussian fingerprint.



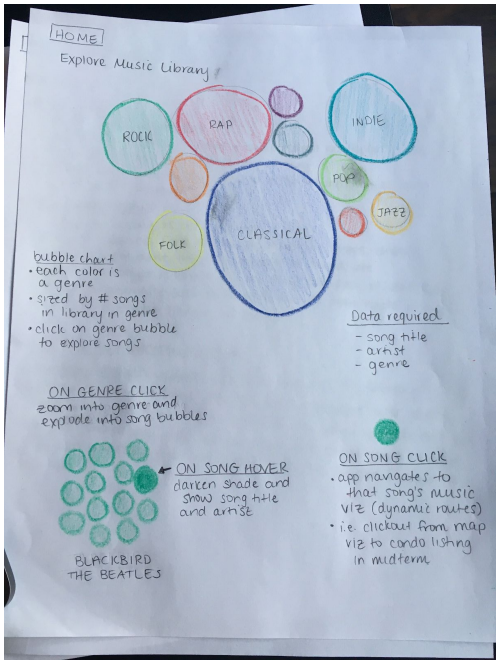
After some more HTML and CSS updates, voilà! AudioForma was born.



Design Evolution: Song Library

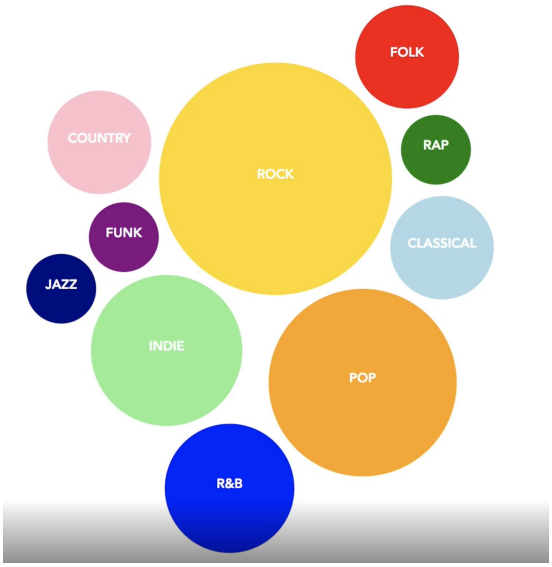
Concurrently, we were working on another visual experience -- one where the user could explore a music library and select a song to hear and view in detail. This visualization hinged on song “metadata” like title, artist, and genre.

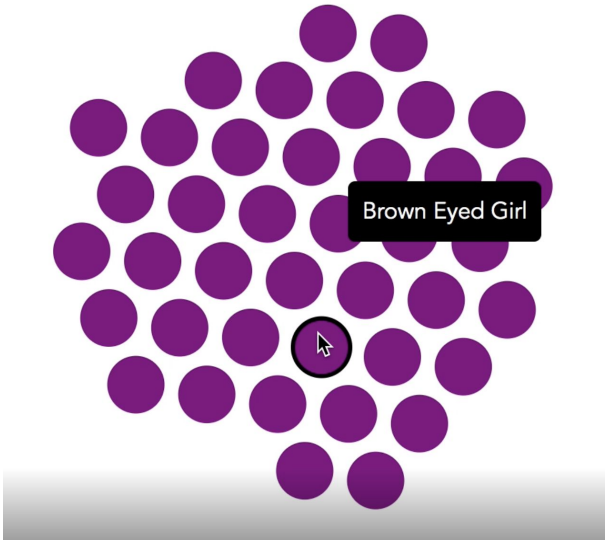
Again, the concept began with a sketch.



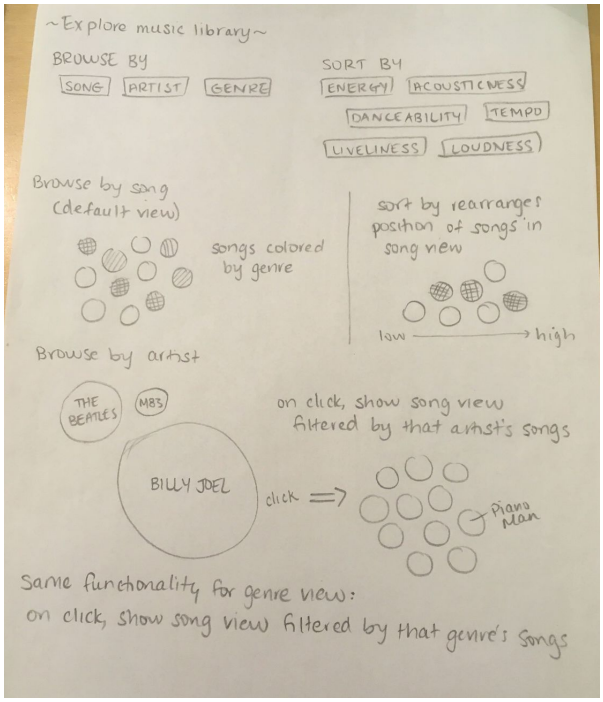
Rebecca’s initial plan was for three views at different levels of aggregation. The song view was the most granular, with each song rendered as a bubble that clicked out to its respective song fingerprint. Two additional aggregate views grouped songs by artist or genre, allowing the user to filter down to a subset of songs on click. Bubbles were colored by genre and sized by number of songs in that bubble group.

In the initial D3 implementation, Rebecca hardcoded a library of song, artist, and genre data in order to build the basic functionality.

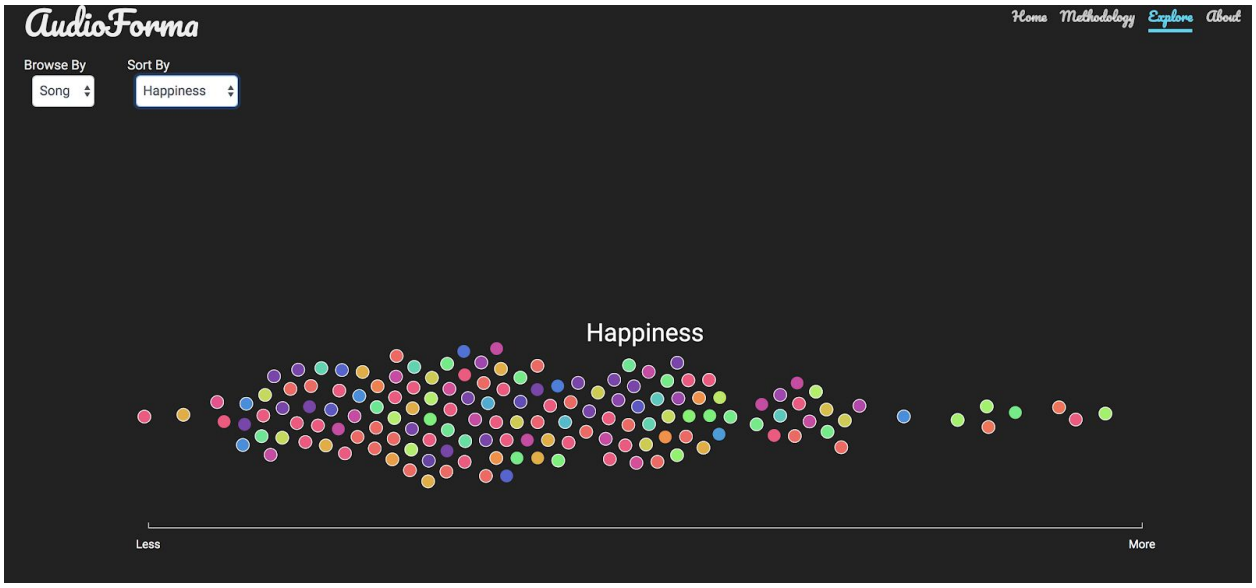


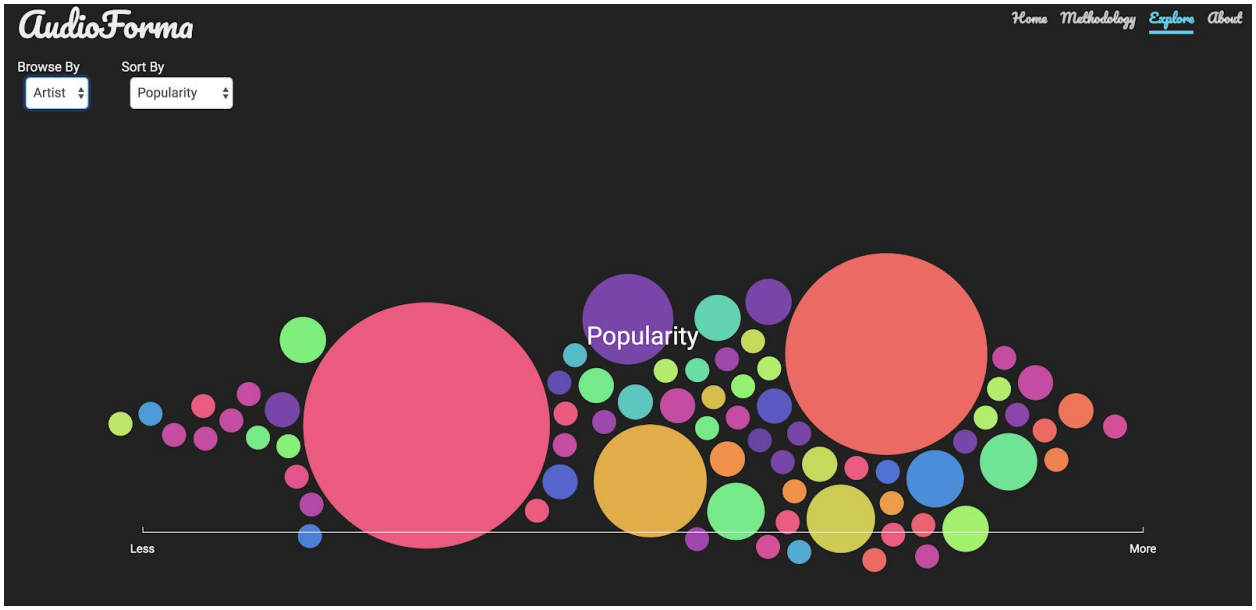


The Spotify integration introduced new data options. We pulled in unique attributes like “acousticness” and “danceability” and used them to sort the song bubbles. To navigate through these new features, we created two different selectors for the user to browse, sort, and choose a song. An early sketch describes this intended functionality.



With the app connected to users' Spotify libraries, we added white outlines to identify the song bubbles that had audio samples available. In the final viz, the user can click on those bubbles to view songs' unique fingerprint visualizations.





About the Team

Tasks were divided among the team, with Ning tackling the song fingerprint visualization, Rebecca taking point on the song library visualization, and Jasper building the backend data processes and web app structure.

SongShape Project Plan

Ning Chen, Jasper Croome, Rebecca Lantner
CSCI E-14A: Building Web Apps for Data Analysis

	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	COMMENTS
1	**Song Visualization:** design unique musical fingerprints						
1.1	Brainstorm & gather inspiration	ALL	11/1/19	11/11/19	10	100%	
1.2	Finalize data requirements: attributes needed for viz, methodology to process audio files	ALL	11/1/19	11/11/19	10	100%	
1.3	Initial design prototyping (sketches, Tableau)	NC & JC	11/1/19	11/11/19	10	100%	
1.4	D3 implementation: static "point in time" song shape	NC	11/6/19	11/18/19	12	100%	NC: I am trying to integrate dynamic effect with the sound wave shapes, so that as the music plays, the corresponding keys are highlighted as their waves increase.
1.5	D3 implementation: "moving" song shape (figure out how to "play" the song & viz simultaneously)	NC	11/12/19	11/26/19	14	100%	JC: I know this needs to utilize "onClick" and the time field from the data, I'm hoping to either leverage the youtube or spotify API to play the music without the issue of royalties or copyright infringement.
1.6	Finalize processing of CSV song library	JC	11/1/19	11/26/19	25	100%	pivoted to using spotify data
1.7	**Complete initial working draft of viz in D3**	NC & RL	**11/26/19**	**11/26/19**	0	100%	
1.8	Iterate on draft viz (design & interactivity)	ALL	11/26/19	12/10/19	14	100%	
1.9	Apply visualization technique to all songs	RL & NC	11/26/19	12/10/19	14	100%	
1.10	**Complete final visualization**	ALL	**12/10/19**	**12/10/19**	0	100%	
2	**Summary Visualization:** explore the song viz library						
2.1	Decide on view (small multiples? interactive bubble chart? etc.)	ALL	11/6/19	11/11/19	5	100%	RL: current concept is a bubble chart, colored by genre and sized by song count. On genre click, category expands into individual song bubbles; on song click, app routes to that song's viz JC - I like this a lot. We could even build small multiples within the bubbles!
2.2	Prototype designs (sketches, Tableau)	ALL	11/6/19	11/11/19	5	100%	
2.3	Gather additional metadata for songs in final library (genre, etc.) if needed	RL	11/11/19	11/18/19	7	100%	
2.4	D3 implementation	RL	11/14/19	11/26/19	12	100%	
2.5	**Complete initial working draft of viz in D3**	RL	**11/26/19**	**11/26/19**	0	100%	
2.6	Iterate on design if needed	ALL	11/26/19	12/5/19	9	100%	
2.7	**Complete final visualization**	ALL	**12/5/19**	**12/5/19**	0	100%	
3	**Web App:** build out essential backend connections						
3.1	Decide on app structure (from user perspective)	ALL	11/9/19	11/14/19	5	100%	App Structure complete
3.2	Create framework for web app (dynamic routes, etc.)	JC	11/14/19	11/21/19	7	100%	dynamic routes complete
3.2.1	Add MetaData to App	JC	11/23/19	11/23/19	0	100%	added connection to spotify API
3.2.2	Connect to Users personal spotify data, using Oauth	JC	11/23/19	11/30/19	7	100%	
3.3	Front end design	JC	11/21/19	11/26/19	5	100%	
3.4	**Complete basic functionality**	ALL	**11/26/19**	**11/26/19**	0	100%	
3.5	**Complete final web app including D3 visualizations**	ALL	**12/10/19**	**12/10/19**	0	100%	
4	**Additional App Functionality:** build out "nice-to-haves" if time allows						
4.1	Create user profiles: login, logout, save songs	TBD	11/26/19	12/3/19	7	0%	
4.2	Create ability for users to upload their own songs	TBD	12/3/19	12/10/19	7	0%