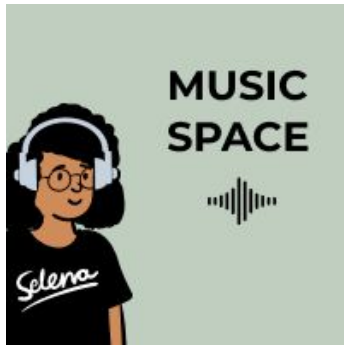


MusicSpace

visualization of music data



Amanda Wu, Alicia Tsai, Tiffany Wang
info247 Spring 2020

Table of Contents

- ❑ Project goals
- ❑ Related work
 - ❑ Related work 1: MusicScape (visual encoding)
 - ❑ Related work 2: An Interactive Visualization of Every Line in Hamilton
 - ❑ Related work 3: Rap Genius
- ❑ Our visualization
 - ❑ Audio Page 1
 - ❑ Audio Page 2
 - ❑ Audio Page 3
 - ❑ Lyrics Page 1
 - ❑ Lyrics Page 2
 - ❑ Lyrics Page 3
- ❑ Data Collection
- ❑ Tools Used
 - ❑ Design and Prototype
 - ❑ Lyrics Visualization
 - ❑ Web Development
 - ❑ Tools Breakdown
- ❑ Implementation Process
 - ❑ Data Collection & Low Fidelity Sketching
 - ❑ Data Processing & EDA
 - ❑ User Feedback
 - ❑ Initial Prototype
 - ❑ User Testing & Refinement
- ❑ User Testing Results
- ❑ Demo
- ❑ Work Allocation
- ❑ Thumbnail image
- ❑ Software create

Project goals

Music Space is a visualization tool based on multiple dimensions of a song, including audio features and lyrics patterns. Users can **generate visualization**, **compare** and **explore** both audio and lyrics data of their favourite song. They can also decompose visualization into small multiples or bring in a different song for further exploration between songs and artists.

We were interested in music visualization after realizing that the current music playing/streaming services mostly provide listeners either a static interface or full-on music videos. The former can be a bit boring and the latter can be distracting and there is no good option in between. Originally, we wanted to create a visualization that is fun, cool and informative targeting the **general music listeners** with the goal of enriching their listening experiences because the added visual element together with the song can engage both of their sight and hearing senses.

After testing our prototype with a few users, we decided to pivot our **target user group** from the general public to **audio enthusiasts and data experts** or any professionals that are interested in an in-depth analysis of music features (e.g. acousticness, tempo, valence) and lyrics patterns since these data might not be too meaningful for a general music listener. Audiences that are interested in spotting trends in music feature and lyrics are also our potential users once we are able to scale the project to include a bigger universe of songs.

The primary goal for Music Space is:

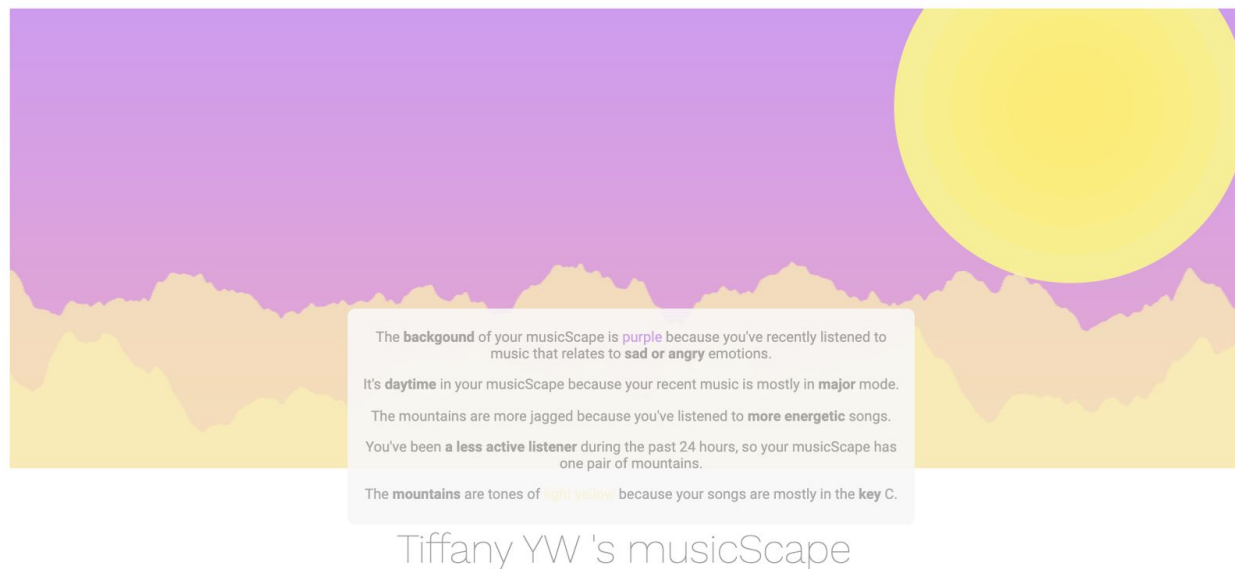
1. To **educate** users on the meanings of audio features by visualization and comparison. The goal is to answer questions such as: What is the valence value of song 1? If the valence of song 2 is higher than song 1, how might that change the entire visualization?
2. To **highlight patterns** from song lyrics and provide insights on the structure of lyrics. The goal is to answer questions such as: What are the most frequent words/sentences in this song? Are pop songs more repetitive than other kinds of songs?

Related work

In order to further explore our options, we have reviewed some current work to get a sense of the existing work in the space and how we can differentiate from them. We have mainly focused on the following 3 designs.

Related work 1: [MusicScape](#) (visual encoding)

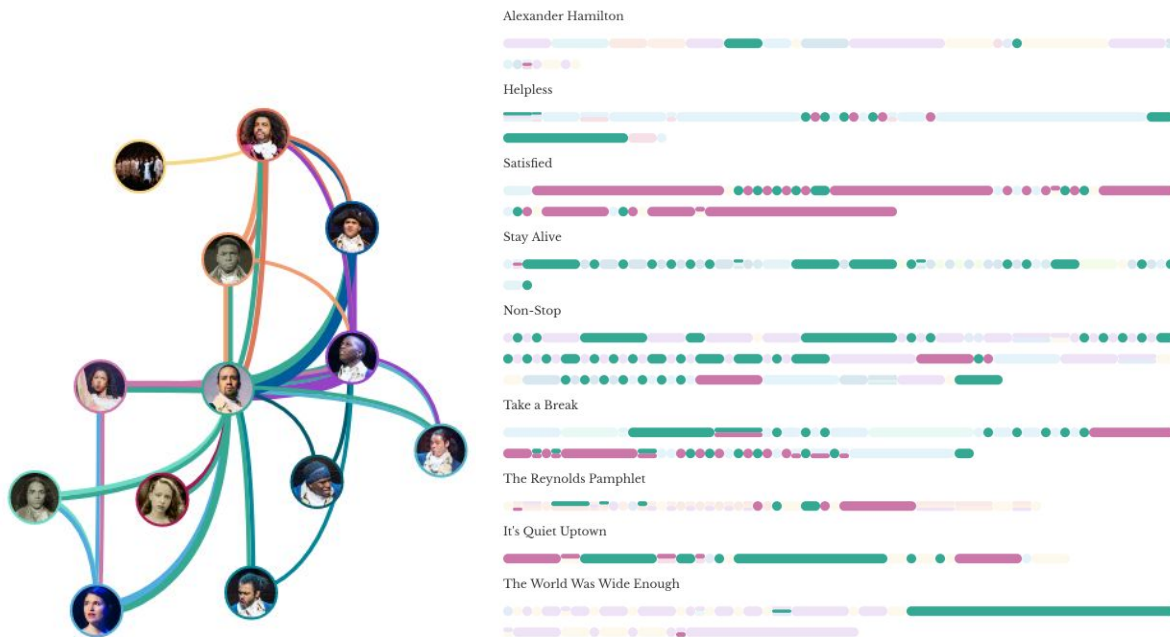
This site is our main source of inspiration. It automatically generates different landscape visual encoding based on user historical listening experiences. The main goal of this website is to help users understand their own listening patterns through visual explanation. The visual effect is quite pleasing and the idea of generating unique visuals for each individual really struck us. However, there are also several downsides of this interface design interface. First of it, there is no interactivity. The page is static and requires no user input to explore more options. It is also hard to navigate through the website because it provides no explanation how it works. Additionally, it heavily relies on user data, which is not easy to acquire. Lastly, though the design is aesthetic, the elements used in this generated image is very limited. And there are not many differences between different listening patterns, and there is no space or comparison in the site. Thus, with the inspirations we got from this interface, we want to expand its features and incorporate more visual encodings.



Related work 2: [An Interactive Visualization of Every Line in Hamilton](#)

This site analyzed the lyrics of the famous musical Hamilton and visualized the lyrics. The main goal of the author is to answer the following two questions: 1) the relationships between the main

characters, and 2) the recurring phrases associated with those characters. The visualization analyzed who sang each line in the musical as well as who that line may have been directed towards. The analysis of co-occurrence of lines among different characters inspires us to investigate the co-occurrence of words in the lyrics.



Related work 3: [Rap Genius](#)

Rap genius is a concept that allows users to explore an artist's use of literary devices. It also visualizes lyrical elements such as complexity, unique rhymes, average syllables per bar and literary distribution. One feature we drew inspiration from is comparing two artists and visualizing their literary differences. The project was particularly good at unpacking literary devices and putting them into categories such as narratives, metaphors, etc and comparing them side by side for 2 artists. We see an analogy in our audio feature data, since there are 9 different features, and we wanted to allow users to see the feature differences between two songs.



Our visualization

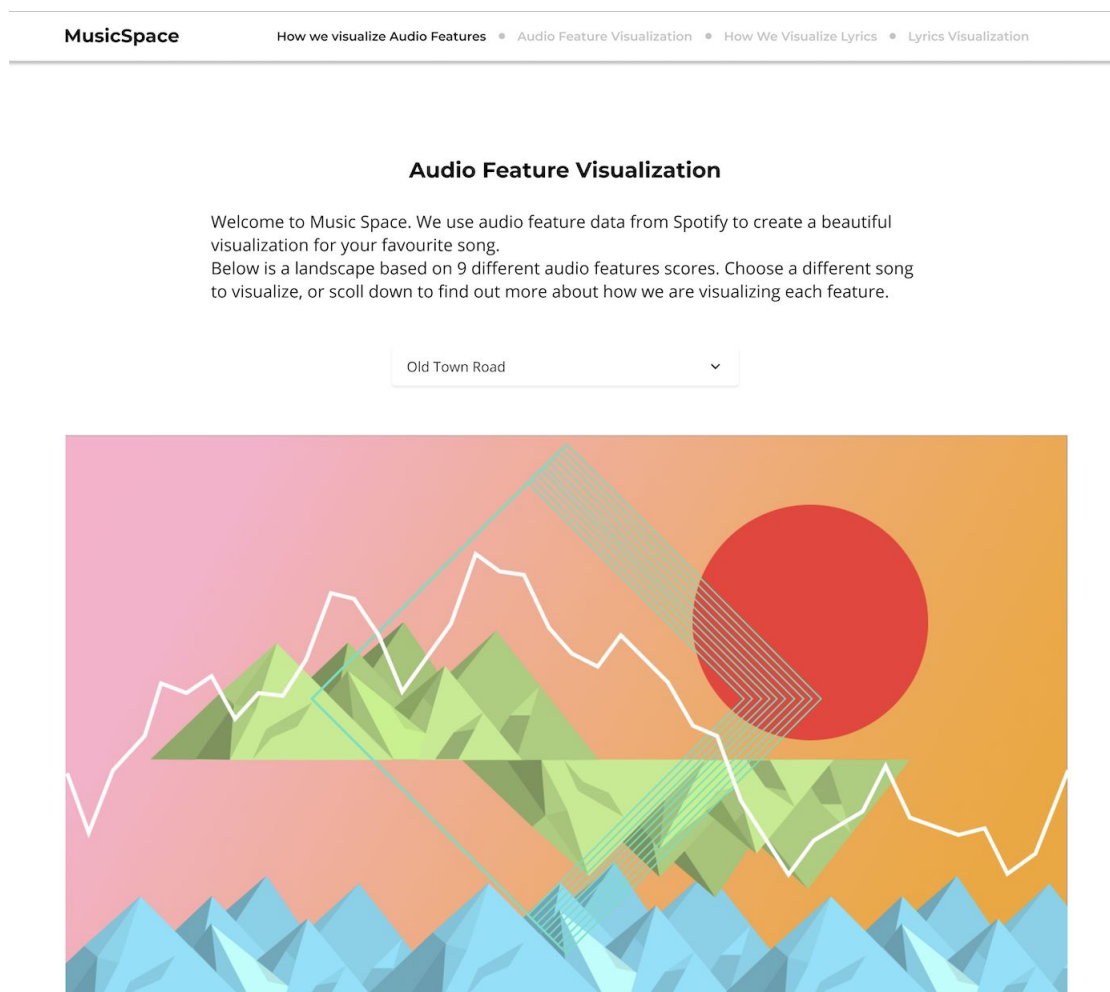
We divided the visualization process into two parts, **audio features** visualization and **lyrics** visualization. In each part, we incorporated **educational content** on our visualization methodology and some **interactive functionality** into our web UI to guide users through a complicated journey.

In terms of user experience, we are focusing on first educating the user on what we are doing, then encouraging them to interact with our visualization with small tasks such as comparing songs. We believe this UX approach can reinforce understanding and keep the users engaged.

1. Visualize audio features of a default song

(Audio Feature Education - Step 1)

After the initial simplistic landing page. We welcome the user to our page and provide a brief introduction of the data we're using to visualize. We select a song by default and first display the visualization of the song using 9 audio features combined into one animated image. On the same page users can choose a different song to visualize. Users then can scroll down to learn more about music features that were used to generate this image shown on the page.



2. Decomposition of 9 different features

(Audio Feature Education - Step 2)

We display 9 small multiples of all the audio features we used to visualize the landscape above and provide explanations of what they mean. This step is to help users with understanding of the audio features as well as how the above graph was generated based on their corresponding scores. We also include our justification of our choice of visual encoding, and explain how a different score will change each element visually.

The 9 boxes and their embedded visual encoding changes as the user selects different songs from the preview page. When the user hover over each box, they will see a display of each feature's corresponding score. This is the second step of helping our users understand audio features of each song. To further help them to see the different meaning of the scores, they can scroll down to compare two different tracks with the same feature.

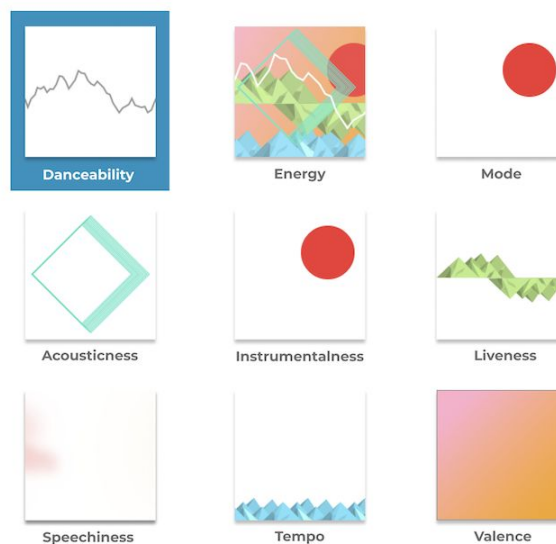
How We Visualize Audio Features

We visualize a song based on 9 audio features populated by Spotify based on the characteristics of the song.

Below is the breakdown of each audio feature we've used in the visualization.

Explore each feature by comparing visualizations from 2 different songs.

Choose a Audio Feature:



Danceability

Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

We use a jagged line to represent danceability because the edges and sharp points remind us of pulse and represents movement. The higher the danceability is, the more jagged the line is.

3. Comparing music features of 2 different songs

(Audio Feature Education - Step 3)

To further reinforce the understanding of each music feature and engage the audience more, we bring our users to our third step - this comparison page. We encourage users to play with each audio feature by choosing two different songs and compare how a different score might change the pictures for those two songs¹. Users can switch between audio features in the previous section and compare across 9 features between 2 songs.

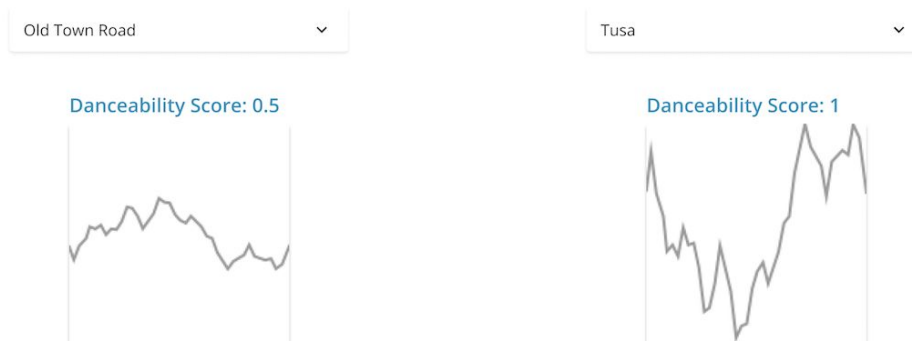
Under each choice of audio feature, we, again, provide a detailed explanation of its meaning defined by Spotify API and how its visual encoding works on our website to generate a unique image for each track.

Danceability

Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

We use a jagged line to represent danceability because the edges and sharp points remind us of pulse and represents movement. The higher the danceability is, the more jagged the line is.

Comparing Audio Features of 2 Songs



¹ See [Appendix 1](#) for specific calculations

4. Part 2: Lyrics visualization, and how we did it

(Lyrics Visualization Education - Step 1)

In order to first help our users get a basic understanding of how we use word frequency and matrix to visualize lyrics in a brand-new way, we need to walk through the process with our users to ensure users with all backgrounds understand how it works. Starting with our first step - here we are giving users a simple explanation of our methodology to visualize lyrics with just 1 sentence. We are highlighting how word frequency drives color choice in the colormap.

On this page, we explain in detail what each row and column represents and how we assign color. Although, our design theme is minimalistic, we intentionally want to use more words here because we feel like it can help with user understanding since this design is not very intuitive.

How We Visualize Lyrics

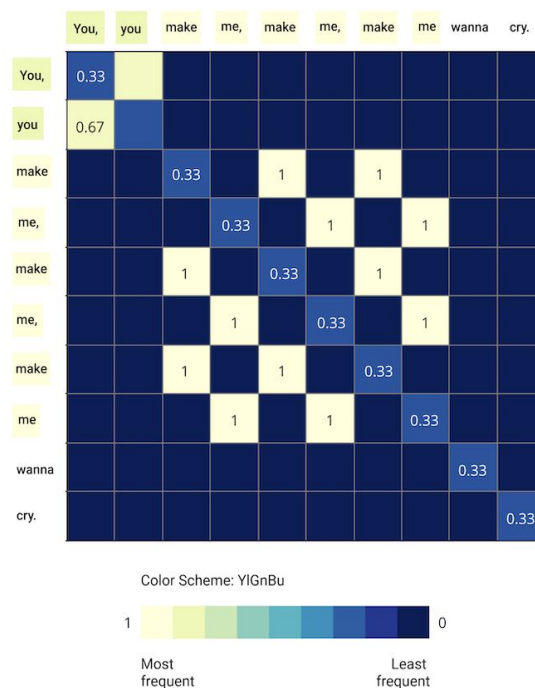
This 10 x 10 matrix represents 10 words of the sentence "You, you make me, make me, make me wanna cry."

Each row and column of the matrix corresponds to a word in the lyrics. A grid cell is filled in if its row has the same word as its column.

The color of the cell is determined by the frequency of words in the song.

How we assign color:

1. Normalize # of count for each word in the matrix ($\# \text{ count} / \text{max count}$)
2. Plot the matrix with colormap YlGnBu based on the relative score from 0-1

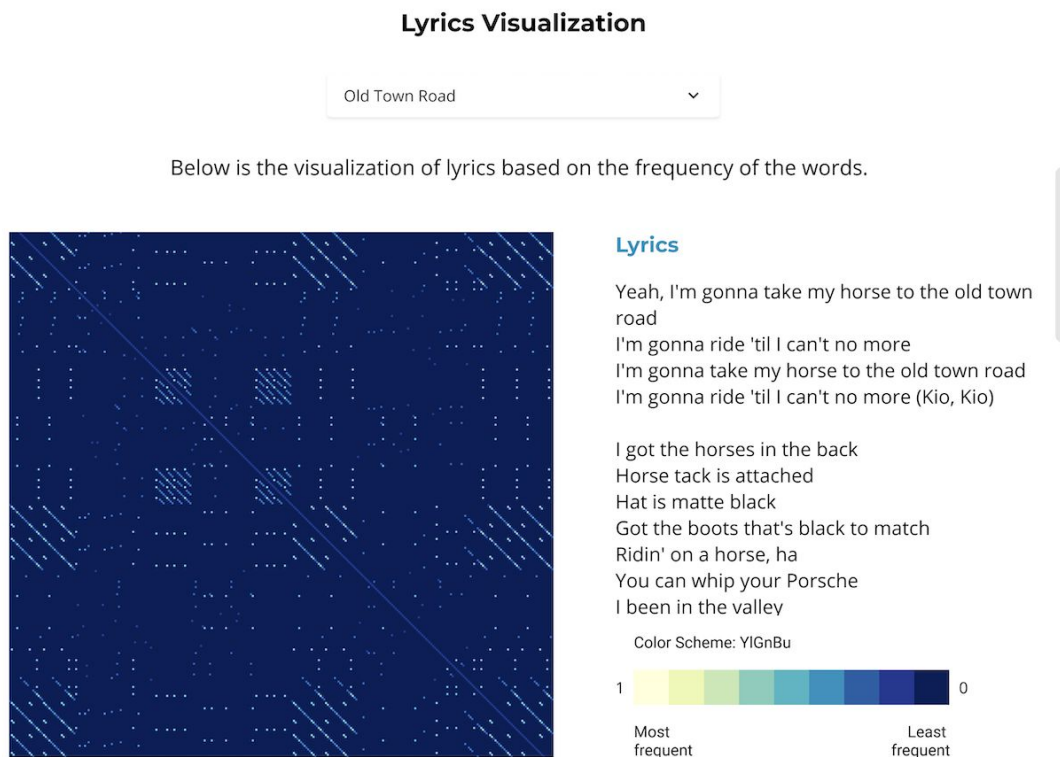


5. Displaying the overall lyrics visualization

(Lyrics Visualization Education - Step 2)

We display the overall lyrics visualization of a selected song with the lyrics on the side. This gives the user an overview of the general pattern and structure of the lyrics. Users can switch between songs to see the structural difference of each song.

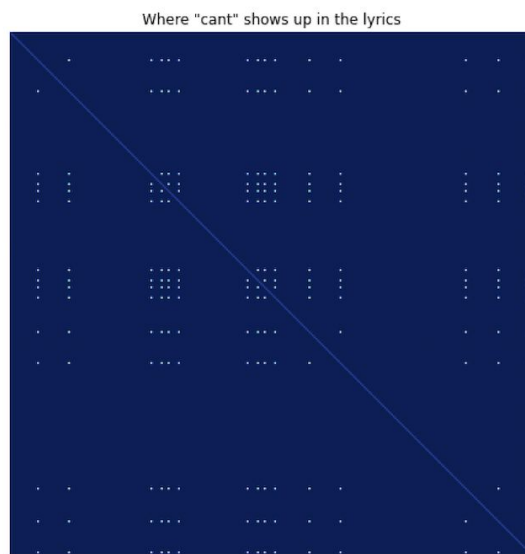
This step is to not only give the user a big picture, but also aims to deepen their understanding of our methodology used to visualize lyrics.



6. Drilling into some details of the lyrics

(Lyrics Visualization Education - Step 3)

Lastly, to help the users better unpack the lyrics even further, we provide more details on the frequency distribution on the top word and top sentence. Users can also see the top 10 frequent words of the song. This way, we have achieved our goal of educating our users on understanding our method used here - going from an explanation to an overview and lastly converging again to details.



Most Frequent Words

	# of word count	Normalized count
1. cant	210	0.689
2. im	156	0.511
3. gonna	132	0.433
4. i	132	0.433
5. my	132	0.433
6. the	90	0.295
7. horse	56	0.184
8. no	56	0.184
9. me	56	0.184
10. nothin	56	0.184

Color Scheme: YlGnBu



Most Frequent Sentences

1. im gonna ride til i cant no more
2. you cant tell me nothin
3. cant nobody tell me nothin
4. yeah im gonna take my horse to the old town road

Color Scheme: YlGnBu



Data Collection

Audio Features

The data is collected from the [Spotify API](#) using Python. We collected the [audio features](#) of each song from Spotify. Features collected from the API are listed below:

Feature	Description
Duration	The duration of the track in milliseconds.
Key	The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation . E.g. 0 = C, 1 = C #/D ♭, 2 = D, and so on. If no key was detected, the value is -1.
Mode	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
Time Signature	An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).
Acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
Danceability	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
Energy	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
Instrumentalness	Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
Liveness	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
Loudness	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.
Speechness	Speechiness detects the presence of spoken words in a track. The more exclusively

	speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
Valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
Tempo	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
Lyrics	The lyrics of the song.

Lyrics Data

Lyrics data is collected from the [Genius Lyrics website](#) by scraping the lyrics page with Python code. For the lyrics text, we removed the punctuation and the text indicating the section of the songs such as [Chorus], [Verse 1], etc. An example of lyrics scraped from the Genius website is shown below.

[Chorus]

I tell a nigga don't dick ride, don't blick ride
Leave it to the double thick thighs, twin sisters

...

[Verse 1]

Now we catch him at the chicken spot, up a couple chops
Pop that nigga with a hundred shots, ra-ta-ta-ta

...

[Chorus]

I tell a nigga don't dick ride, don't blick ride
Leave it to the double thick thighs, twin sisters

...

[Verse 2]

You're mad I'm back, big mad
He's mad, she's mad, big sad

...

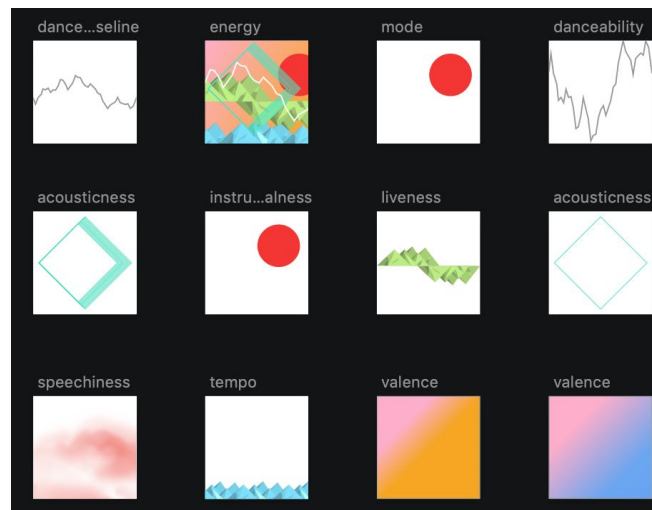
[Outro]

(Jah, ayy, you know that boy)

Tools Used

Design and Prototype

Feature visualization: for feature visualization we used pen and pencil to sketch out some initial ideas and then used Sketch to create SVG mockups like below:



We exported the SVGs to Observable to adjust the color/size/quantity of each visual encoding to finalize on how each feature score will change the image. We used Python code to process the raw feature scores and produced transformation scale/variables for each feature. We then use D3 and html in Observable to generate small multiples of 9 svgs for each song and layer the 9 features to generate a main visualization svg.

Website Prototype: we used Figma to create an interactive prototype for usability testing and modeled our web UI after the prototype.

Lyrics Visualization

The co-occurrence matrix of a song is calculated using Python code. We first generate a symmetric matrix with dimension (#words, #words) for the song and populate zero for all the entries in the matrix. For each unique word in the lyrics, we retrieve the indices where the word appears in the lyrics. Then, we perform a permutation for all the indices and color the corresponding entry of the matrix. The value of the color is determined by the frequency of the word in the lyrics. The frequency is normalized to range from 0 to 1. Finally, the Matplotlib package is used to plot the final co-occurrence matrix.

For most frequent words, we generated a list and count of top 10 words using Python code and created a bar graph for each song using Tableau.

Web Development

For the web development part, we started with a bootstrap template based on our discussed layout. However, as we get more feedback from the users and we iterate many times on the web layout. Thus, we ended up writing in html basically from scratch. Although, we relied on css on most of the display of the components. For the more difficult styling, we chose to do it on D3, as it is a more customizable coding language. As we also want to include some interactivity with the users we added some javascript and jQuery code as well. Additionally, since we did some interesting visualizations using Python, tableau, D3, and figma, we have embedded their code & images into the website.

Tools Breakdown

Activity	Components	Tool
Design	Sketching	Pen and paper
	Graphics	Illustrator, Sketch, Figma
	Presentation	Google Slides
Data	Input	Python API data extraction
	EDA	Tableau, Python
	Manipulation	Python
Web Implementation	Layout	HTML, CSS, Javascript, JQuery
	Graphics	Figma
	Visualization	D3.js, Figma, Tableau
	Animations	D3.js

Implementation Process

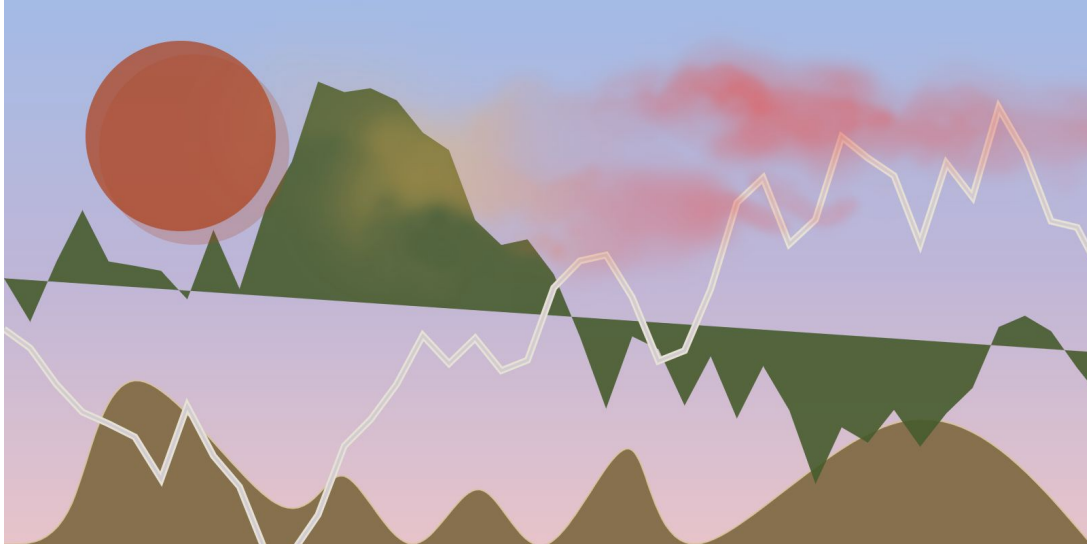
1. Data Collection & Low Fidelity Sketching

During our ideation we were fairly certain that we wanted to do something with music. In order to figure out what kind of story we want to tell to our users, we first started a data collection process from multiple data sources to get as much data as we can. After settling down on spotify audio data and genius lyrics data, we discussed many ways we can put those two datasets into use. After exploring some related work, we settled down on our current idea. Thus, in our early meetings, we created some sketches to decide on the best narrative (see below).



2. Data Processing & EDA

We then started to get to know our data through data processing and EDA. In order to enrich our story-telling, we played around with tableau to see if we saw any interesting patterns and what other features we could add to our final deliverable. From there, we decided that the patterns that we are looking for don't quite align with the visualization outputs constrained by quantitative data, so we decided to use more tools, such as D3, fima, sketch, to explore the possibilities. Luckily, the exploration from those tools were quite pleasing. See screenshot of initially animated audio feature graph using d3:



3. User Feedback (in-class presentation)

Thanks to the in-class presentation, we got some valuable feedback from our potential users. The user feedback we got helped us to find the most effective way to visualize our data and to focus on conveying information that matters the most to our audience. Some interesting feedback (see below) and insights we got from our instructor and classmate feedback were the interpretability of our visual encodings and their relationships to each song track; Thus, we have decided to focus our prototype aiming to solving the issues that our users proposed.

Really cool idea, I would love to do this for my music! Just one critique: It seems like the visualization would provide for a lot of noise and be somewhat difficult to interpret, is there any way you could combine certain features to make a less cluttered visualization?

I love the concept - it definitely seems like it would be an improvement on traditional music visualizers!

4. Initial Prototype

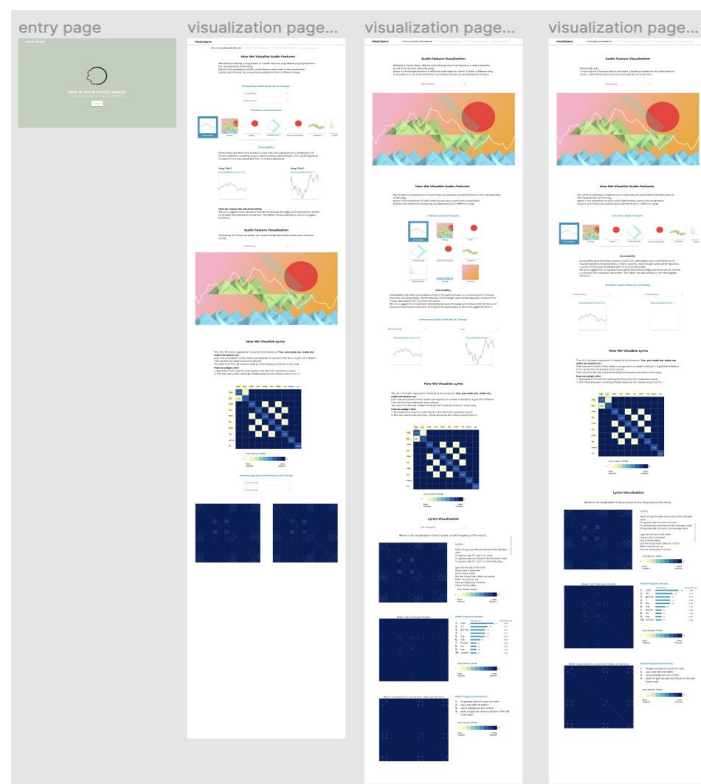
Based on our Figma sketch and our user interviews (classmates & instructor), we began to create user flows and design the overall structure of the website. We spent a lot of time on coming up with the most intuitive meanwhile aesthetic visual encoding for each audio feature and display them on D3. We have also spent quite a lot of time on the lyric matrix generation as we want to create a novel to visualize lyrics where it provides more meaning on the pattern matching side of lyrics. Below is a screenshot of our initial website user flow using figma.



After the basic flow was finalized, we created a MVP website that housed each of the visualizations together to create a coherent user journey.

5. User Testing & Refinement

We conducted our usability test with the initial website we created based on the framework above and we got value feedback. The main takeaway from the user tests was also our problem on interpretability - how can we make sure our users understand the information displayed. We then went through another iteration of website framework design (see below for the 3 versions we came up with). With many rounds of discussion (meeting with instructor) we have finally decided to go with the middle design.



Usability Testing & Results

Goal

Our goal is to enhance the user's knowledge of the music and allow them to explore audio features and lyrics patterns of a song.

In order to find out the effectiveness of our visualization, we designed a test that evaluates the user's understanding of music features and lyrics features **before and after** interacting with our website².

Additionally, we want to evaluate the heuristics of our design to see if:

1. We've used the most meaningful visual encoding methods
2. We've successfully supported exploring and explaining the data.

Method

For our usability test, we used a [pre-test survey](#), a [task-oriented usability test](#), and a [post-test survey](#). We asked the participant to listen to a music track and show them the lyrics of the song before our test.

We've created a pre-test survey with 5 questions asking the participant to give us answers on the estimation of some quantitative features of the song and a free-response describing their general impression of the song.

We then guide them through a 3-step usability test with goals defined for each step, we make observations of the participants and record if they've successfully completed the goal.

Lastly, we give them a post-test survey, repeating the same question of quantitative features of the song and record the accuracy after giving them the visualization. In addition, we ask the participants to rank the visual encoding for audio features from most effective to least effective, and ask them for general comments on things they liked/found confusing. We conclude with a question asking if the participant had felt like they've learned something from our visualization and if they've felt like our visualization has enhanced their experience with the music overall.

² Due to the time and technical constraint, we have decided to focus our beginning to end visualization of music features on one song - "Old Town Road", as it has been the #1 song on the billboard for the longest time.

Participants

We used a convenience sample from our own network of family and friends. Because we think our target audience is the general public or any music listener, the convenience sample is appropriate here³.

Results

Quantitative Measures - Pre vs. Post Test

	Total Answer Correct from pre-test:	Total Answer Correct from post-test:
Participant 1	1/4	4/5
Participant 2	1/4	3/5
Participant 3	0/4	4/5

All 3 participants did better in the post test in answering music feature questions after interaction with our visualization.

All 3 participants think that the visualization has taught them something new about this song.

Qualitative Measures - Usability Study

Scenario 1: Can you find the valence value of this song? Can you describe what the score means and what it tells us about the song?

- **Participant 1:** 0.507, valence is the emotion of the song, this song is more negative since the color is blue-purple. (Participant was reading the color instead of the score for valence.)
- **Participant 2:** NA
- **Participant 3:** 0.507 neutral sound (1 is energetic, 0 is depressing)

Scenario 2: Can you tell us if a different song with a high liveness score will change the graph?

- **Participant 1:** It will add more mountains to the picture
- **Participant 2:** different color scale
- **Participant 3:** more color gradient

Scenario 3: Can you find the 3rd most frequent word in the lyrics?

- **Participant 1:** gonna
- **Participant 2:** cant, gonna, horses
- **Participant 3:** gonna

³ After our decision to pivot to a different target audience, it is important for us to re-run the test with our target audience in the future to ensure that we are achieving our goal with the new audience group.

Takeaways & Revisions

Visual Encoding

General issues we've found with our use of visual encoding is that users tend to focus too much on the pictures of each feature instead of reading the explaining text we've included.

This is important because we've also found out that, if a user is able to understand what the feature is, they tend to agree with the visual encoding we've chosen. For example, danceability is a feature that's self-explanatory and 2 out of 3 users ranked it as the most informative and most effective visual encoding.

With this discovery, we've decided to change our website to add explainability. We've tried different methods including adding the pictures of extremas to each feature and allowing users to compare 2 songs on the same feature. We eventually decided to keep the comparing functionality, as it engages users better, and actually helps them to read the text when they start seeing differences between songs.

Lyrics Analysis

With the testing of lyrics visualization matrix we found out that users with a data science background have a much easier time understanding the methodology of the visualization, and have a higher chance of answering questions on lyrics correctly. Users without a data science background still struggle with understanding even after reading the educational content.

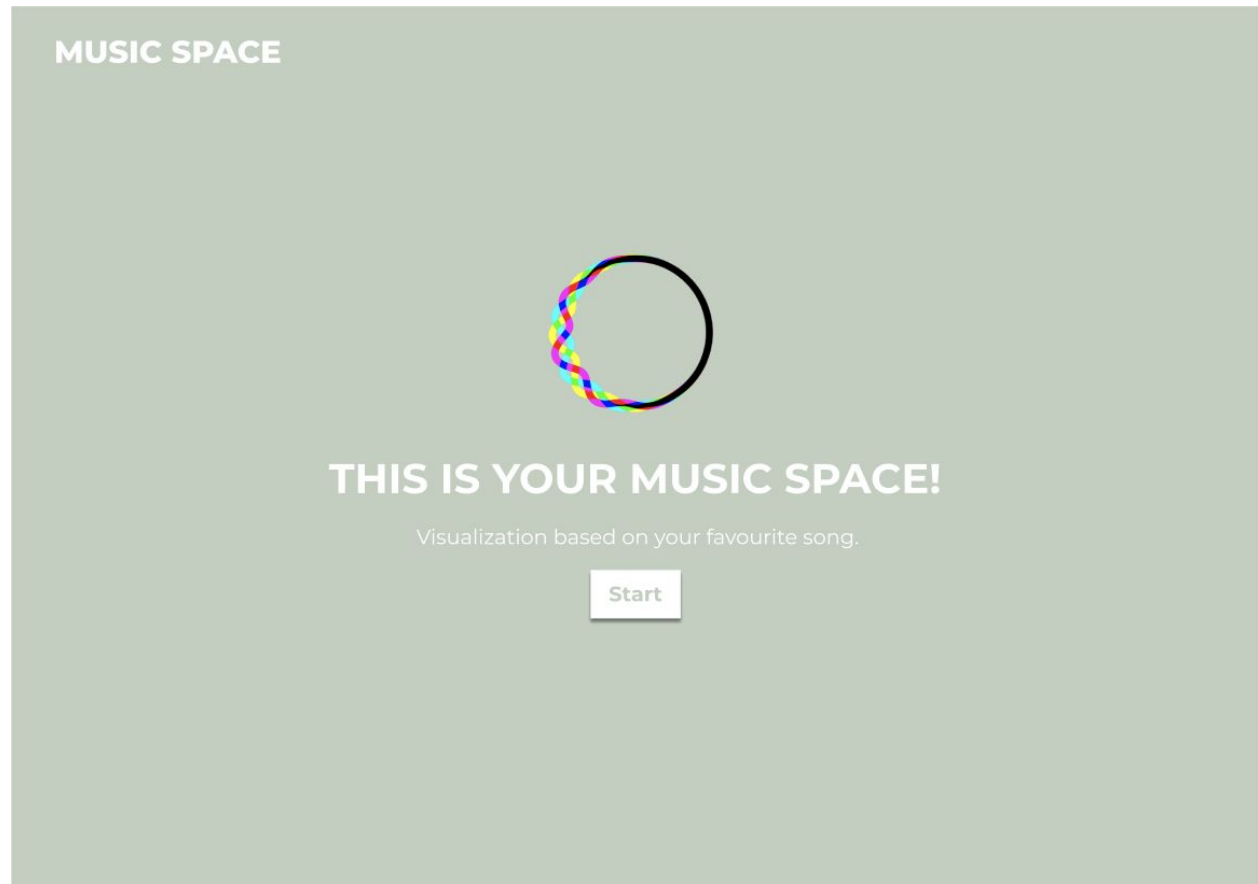
This has also prompted us to consider changing the target audience to people with professional background in data and/or music analysis.

Other

We've also found out that there are some discoverability issues with our website, since most users are not scrolling down to see more content. We've added a top navigation that changes state as the user scrolls to different positions to indicate progress.

We've also received feedback on how the feature matrix is "confusing" and at a glance, users didn't know how to interact with it. We've decided to add more conversational text to guide the user through the process, and we've changed the display order of our content to provide more guidance throughout the website.

Demo

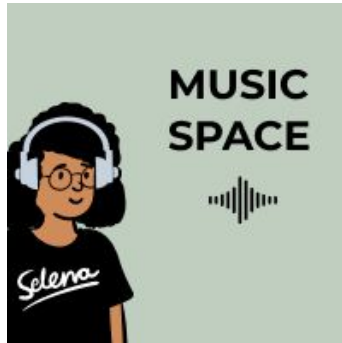


[demo.mov](#)

Work Allocation

Project Components	Sub Components	Tiffany	Amanda	Alicia
Data Preparation	Data Research	33%	33%	33%
	Data Collection & Cleaning	10%	10%	80%
	EDA	33%	33%	33%
Visual Encoding Design	Mapping features to graphics	50%	50%	0%
	Generate abstract visual encodings	50%	50%	0%
	Spotify API audio feature visual encoding	45%	45%	10%
Lyrics Visualization Design	Generate co-occurrence matrix	10%	10%	80%
	Visualize co-occurrence matrix	20%	10%	70%
	Calculate lyrics statistics	40%	0%	60%
User Testing	Interview & Heuristic Design	80%	10%	10%
	Interviews	33%	33%	33%
	Usability Testing	33%	33%	33%
Website Design	UX and UI design	70%	20%	10%
	D3 Animation	25%	75%	0%
	Web development	0%	100%	0%
Presentation & Writeup	Presentation	33%	33%	33%
	Report	33%	33%	33%

Thumbnail image



Software created

Github: <http://github.com/amandaywu/musicSpace> (all files included)

Website: <http://amandaywu.github.io/musicSpace>

Observables 1: <https://observablehq.com/@amandawyq/musicspace-audio-features>

Observables 2: <https://observablehq.com/d/faf851b129f06ea8>

Observables 3: <https://observablehq.com/@yuexitwang/audio-visualization>

Observables 4: <https://observablehq.com/@yuexitwang/audio-features>

Appendix 1 -Music Feature Score Transformation Calculation

Feature Name	Visual Encoding	Transformation Formula
danceability	Using jagged line, difference danceability affects the scale of the line	danceability_scale = danceability/0.5 (scaling it according to baseline for the height of the jagged line)
energy	Using transparency of the entire svg	opacity_baseline = 0.3 adjusted_opacity = energy+opacity_baseline
mode	Appearance of a sun if major, and a moon if minor	If mode = 1 then sun, if mode = 0 then moon
speechiness	The amount of animated cloud to display	If speechiness <= 0.33, cloud level = low; if 0.33<speechiness<=0.66, cloud level = mid; else cloud level = high
acousticness	The amount of line square box to display	Start with 10 lines, for each increment of 0.1 in acousticness, decrease 1 line
instrumentalness	The magnitude of the sun/moon	max_radius = 150 min_radius = 2 radius = (1-instrumentalness)*max_radius If instrumentalness is 1, then use min_radius
liveness	The scale of the green mountains	liveness_scale = liveness/0.5 #scale for the green mountains based on baseline
valence	The gradient of the background color, changing the offsets to adjust amount of warm/cool color to display	If valence >=0.5, use the pink-orange gradient, if valence< 0.5, use pink-blue gradient offsets = 100-np.absolute(50-valence*100)
tempo	The amount of waves to display	If tempo<=60 npm, display 1 set of waves, if 60<tempo<=120, display 2 sets of waves, if 120<tempo<=180, display 3 sets of waves, if tempo > 180, display 4 sets of waves

Appendix 2 -Usability Test Scripts, Measures and Raw Data

Test Script:

- Give the participant a 30s overview of our project - what we are visualizing, what data we're using, and which song we will be playing.
- Play the song to the participant and show them the lyrics.
- Give participant the pre-test
- Show participant the prototype website, give them 2-3 mins to explore the interface and ask them to complete 3 tasks (do not show them how to do it unless they get stuck, then give them hints on where they might want to look):
 - a. **Scenario 1:** Can you find the valence value of this song? Can you describe what the score means and what it tells us about the song?
 - b. **Scenario 2:** Can you tell us if a different song with a high liveness score will change the graph?
 - c. **Scenario 3:** Can you find the 3rd most frequent word in the lyrics?
- Record if the participants have successfully completed the tasks. Also record places where the participants got stuck or find confusing.
- Give the participant the post-test.
- Ask for any questions or additional comments. Thank the participant for their time.

Pre-test:

1. From a scale of 1 to 100, with 1 being the least acoustic and 100 being the most acoustic, what do you think the song's acousticness is?
2. From a scale of 1 to 100, with 1 being the least danceable and 100 being the most danceable, what do you think the song's danceability is? (Danceability: how suitable a track is for dancing)
3. Can you tell if the song is in a major or minor mode?
4. What do you think is the most frequent word in the lyrics?
5. What are the 3 words you would use to describe the song's vibe? (users can use words below if they have difficulty coming up with words.)

Soothe	Excite	Relax
Stimulate	Calm	Enlighten
Frighten	Focus	Invigorate

Rejuvenate	Restore	Improve
Heal	Empower	Stir
Incite	Exhilarate	Uplift

Post-test:

1. From a scale of 1 to 100, with 1 being the least acoustic and 100 being the most acoustic, what do you think the song's acousticness is?
2. From a scale of 1 to 100, with 1 being the least danceable and 100 being the most danceable, what do you think the song's danceability is? (Danceability: how suitable a track is for dancing)
3. Can you tell if the song is in a major or minor mode?
4. What do you think is the most frequent word in the lyric?
5. From the lyrics visualization, how many times did the most frequent sentence appear?
6. From the audio feature visual encoding part, choose the one visual that you think relates to the audio feature the most, and the one that relates the least.
7. From the audio feature visual encoding part, choose the one visual that you think has communicated the information to you the most, and the one that communicates the least.
8. Any additional information about the song you would like to see/find out?
9. Do you feel like this visualization taught you something new about this song?
10. Do you think this visualization has enhanced your music experience?

Raw Interview response data

Participant 1:

Question	Pre-test Answers	Post-test Answers
1.	80	5
2.	75	90
3.	Major	Major
4.	"I"	"cant"
5.	Country, repetitive, classic pop	30
6.	-	Most: liveness, danceability

		Least: instrumentalness
7.	-	Most: liveness, danceability Least: acousticness, valence
8.	-	Amount of instruments, amount of tracks in the song.
9.	-	Yes. dissection of the music makes her want to visualize her fav song too.
10.	-	No, it has only enhanced her music knowledge.

Participant 2:

Question	Pre-test Answers	Post-test Answers
1.	65-70	57.8
2.	50	90.7
3.	major	major
4.	"horse"	"cant"
5.	Heal, soothe, rejuvenate	6
6.	-	liveliness(most); mode (least)
7.	-	energy(most); valence(least)
8.	-	Instrumental vs Vocal
9.	-	Yes
10.	-	No change really

Participant 3:

Question	Pre-test Answers	Post-test Answers
1.	1	1
2.	20	90
3.	minor	major
4.	"ride"	"can't"

5.	Stimulate, repetitive, excite	15
6.	-	danceability (most), speechiness (least)
7.	-	danceability (most), visual encoding (least)
8.	-	spectral content
9.	-	yes
10.	-	yes

Other Design Feedback

Landing page

- Participant 2: was very pleased by the landing page, she likes the simplistic design, color scheme, and the animated circle movement - “It makes me feel very chill and relaxed”. She also says the design keeps the website a little mysterious for her and prompts her to discover more by scrolling down.
- Participant 3: Ricky was impressed by the landing page. He thinks it has a nice simple design and the graphic is well displayed.

Audio features section

- Participant 1: thinks that features like liveness (mountain) and danceability (jagged line) are things that are easy to understand and conveys a lot of information whereas features like acousticness is **confusing** (as a music feature) and she struggles to see the relation between acousticness and saturation.
- Participant 2: thinks the design and the concept of visual encoding defined by us were very refreshing. However, she still found it **confusing** at first when she sees the landing page of this section.
- Participant 3: Ricky thinks it is very interesting to visualize music. He would like to know more about how the scores were calculated and the algorithms used to calculate them. For example, how do you calculate beats per minute (through a wavelet transformation?).

Lyrics section

- Participant 1: has a hard time understanding the matrix even with some explaining text and illustration showing how the matrix is generated. Had a particularly hard time understanding that the matrix is symmetrical.
- Participant 2: thinks our representation of the lyrics was very interesting and “something I have never seen before”. She was able to pick up the meaning of the matrix pretty fast after seeing the explanation matrices underneath. However, it was not intuitive for her that we provided explanations after the first main lyric matrix. She was stuck on the lyrics landing page for a while trying to figure out the meaning by herself before she realized there were explanations following that matrix.

- Participant 3: Ricky wonders if the matrix is positive semidefinite (which is not) and he wonders if any of the NLP techniques can be applied. The way that the matrix is calculated is presented very clearly but it should have numbers on the scale. The frequency is color-coded but does not have the actual numbers.