Process Book

Songify

Stefan Popescu, Anne-Marie Rusu, Christopher Williams, Simon Gmür

05/2024

Introduction

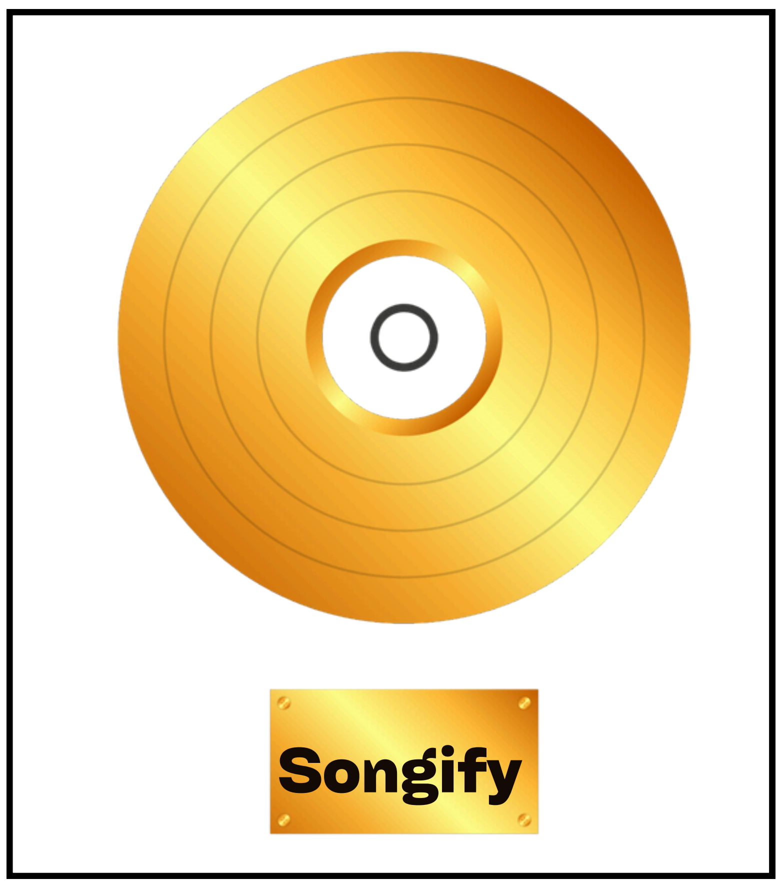
Spotify, Apple Music, Deezer, and all other music streaming platforms evolved hand-in-hand with technological advances. Nowadays, they have meticulously implemented different and very complex recommendation algorithms, making discovering new songs easier than ever. Advanced ML models and huge data sets, constantly collecting and evaluating personal listening behaviors and preferences and trying to please users as best as possible, are unfortunately simultaneously streamlining the diverse categorizations, song-match-making arrangements, and individual proclivities that can be imagined, into certain generalized patterns.

Through our visualization, we aim to provide a new perspective on Spotify top charts data, focusing on metrics that qualitatively describe a song rather than its mere performance in the top charts. Using a dataset of the most popular songs since 2016, our goal is to present the specific characteristics of each song in an interactive manner. Such a presentation would also allow users to intuitively compare individual songs and search/filter for songs containing desired characteristics. With this new tool, we offer a very different approach to sorting and presenting songs, allowing new discoveries in a fashion that diverges from mainstream practices.

Concretely, the presented characteristics that we use to profile songs are as follows:

Danceability, **Energy**, **Instrumentalness**, **Key**, **Liveness**, **Loudness**, **Speechiness**, **Tempo**, **Valence**

To display all of the above, a unique and interactive visualization was designed that allows the user to observe and compare song characteristics at a glance.



Our Idea

Concretely, our idea is to display Spotify charts data in a descriptive way, using a DJ set interface. While having a list of songs to search and select from stylized as a set list or artist list, the user has a DJ booth at their disposal to apply filters to the dataset to visualize their properties. Additionally, the main part of the interface consists of a dynamic visualization of the appropriate characteristics and a small animation assisting in the interpretation of these characteristics. On top of the visuals, we felt the need for audio feedback; so we included song snippets when a song is selected.

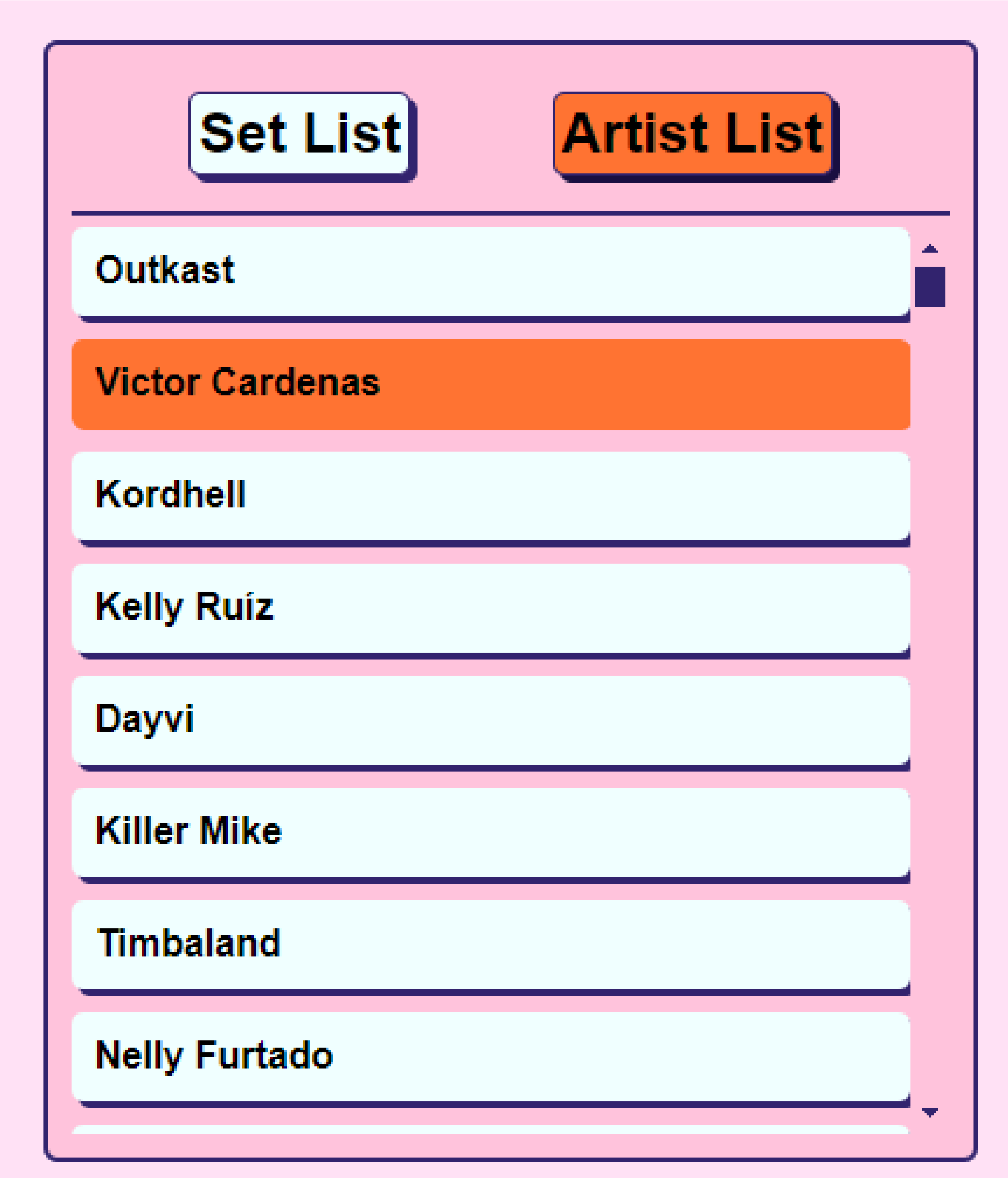
Design

The final implementation meets all our initial wishes and more. All aspects mentioned above and in previous Milestones have been implemented successfully and some extra features added.

Set List/Artist List and Search bar:

The main list of data is shown in a scrollable list, initially set to the “Set List”, showing all songs in the dataset in alphabetical order, and can be toggled to “Artist List” to show the list of artists. The user knows which list they are viewing as the title is in a different color when they click on it.

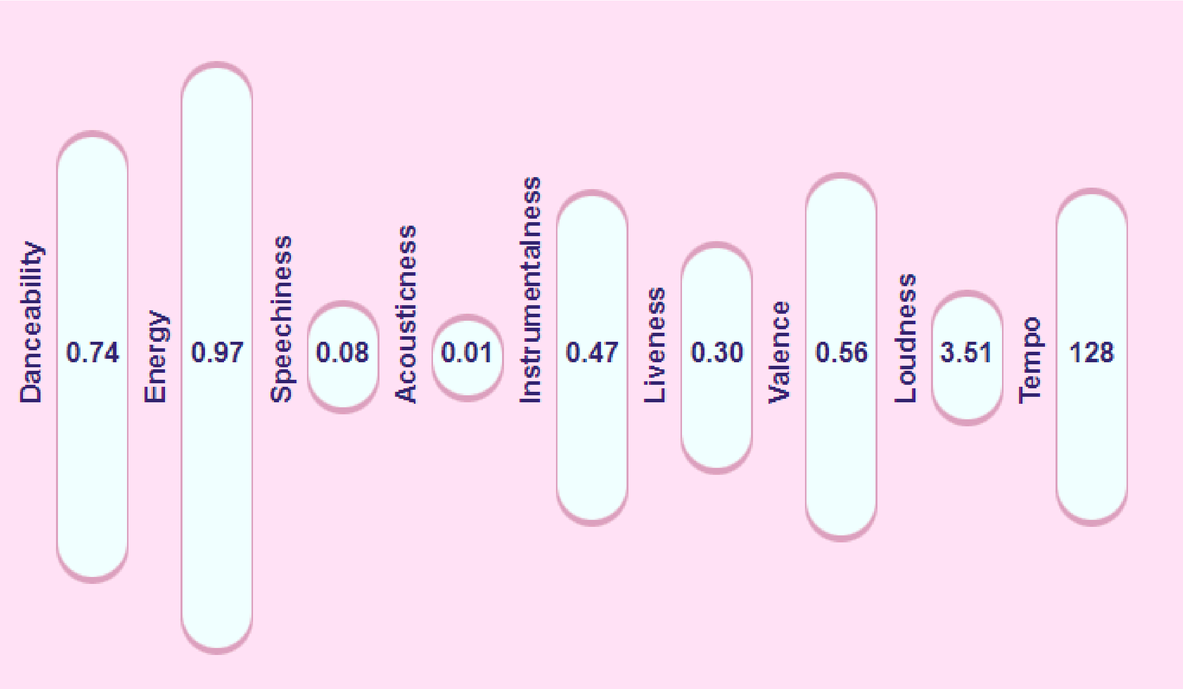
Additionally, the user is able to search for a song or artist, and incremental search progressively shows them results. In “Artist List” mode, the user is only able to search for artists.



Outkast
Victor Cardenas
Kordhell
Kelly Ruíz
Dayvi
Killer Mike
Timbaland
Nelly Furtado

Bar Chart:

When clicking on a song from the Set List, the song statistics are shown in a bar chart in the center area of the page. For visual appeal, the bars update with a smooth transition and change color when hovered over. The bars are clickable and will apply a filter in the DJ Mixer corresponding to that value (or range of values in the case of the toggle buttons). When an artist is selected from the Artist List, the average statistics for that artist are shown.

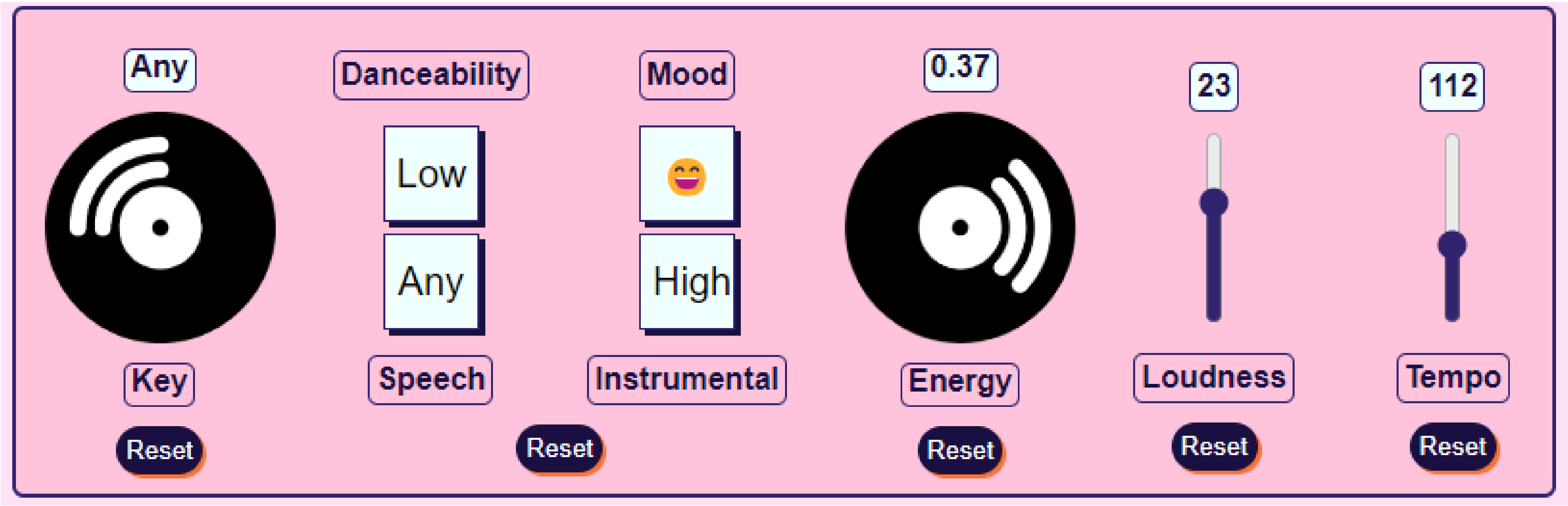


Filtering and Reset Buttons:

Due to the large variety of features the dataset is able to be filtered on, there are several reset buttons of varying granularity allowing the user to quickly and easily change filtering of the data without too much mouse manipulation. By clicking on a "Reset" button, any characteristic can be removed from consideration while filtering, called "Any".

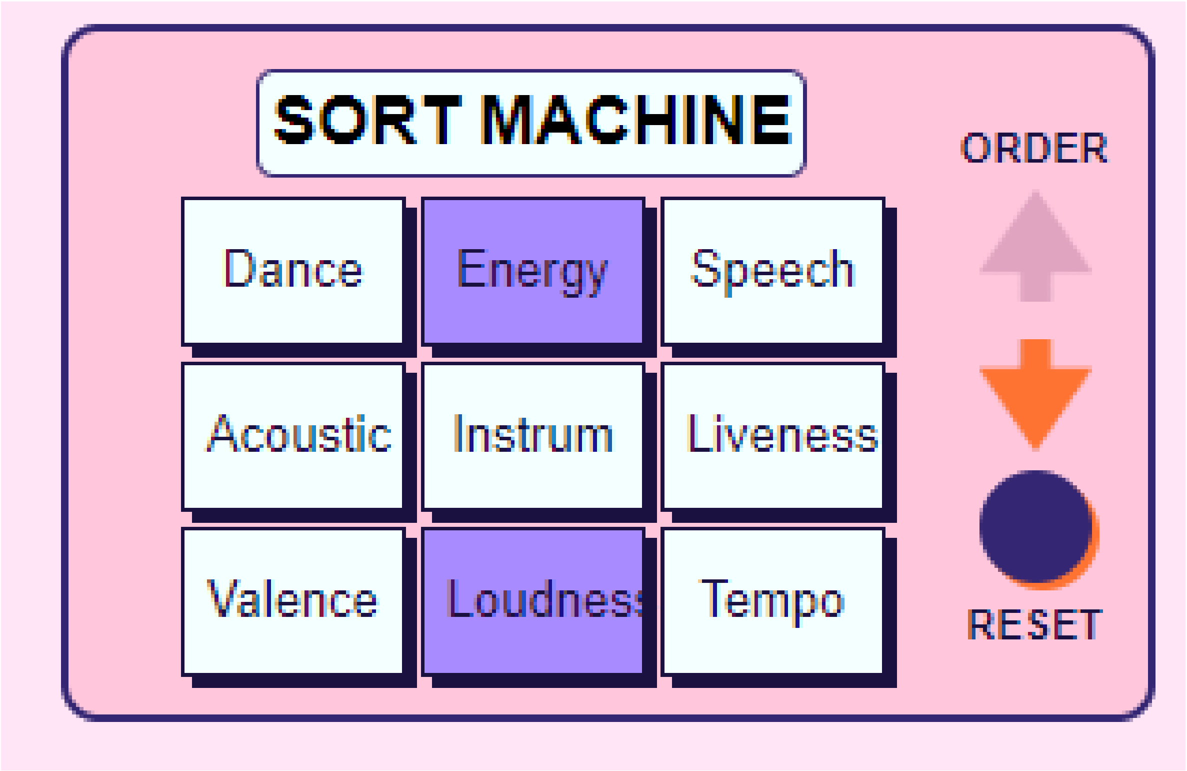
DJ Mixer:

Every characteristic can be manipulated from the DJ Mixer and the song list where characteristic visualization, precise values, and supporting animation react immediately. Some are adjustable through sliders, some with toggle buttons, and some by turning on DJ turntables. These different control types were carefully fitted to the characteristics and their values' distributions. The controls also integrate interaction, such as changing color when hovered over, to allow more visual clarity. For example, when clicking on a bar in the bar chart, it sets the corresponding filter in the DJ Mixer and changes the color of the label temporarily to attract attention to it.



Sort Machine:

This component allows the user to filter and sort songs and artists in the Set List/Artist List based on multiple features. While certain criteria combinations can result in finding no matching songs, which is perfectly acceptable, other combinations can result in finding many matches. Therefore, we implemented a sorting system, and only the best overall matches will appear in the song list according to the selected order. The same intuitive handling applies when the user toggles into "Artist List" mode instead of the "Set List" mode. The only difference is that artists' average characteristics are displayed and no song snippet can be played.

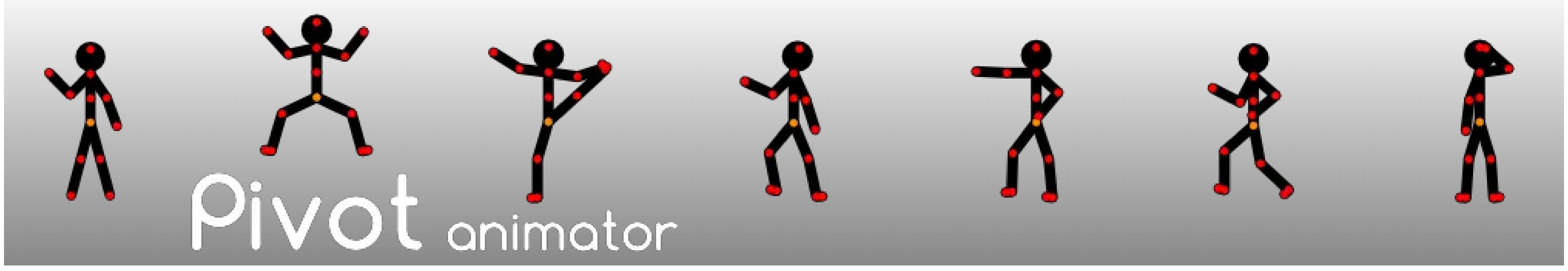


Stickman animation:

We aim to visually represent each selected song through an animated stickman who dances according to the song's characteristics. The dance moves change based on the song's danceability, energy, valence, and tempo. We categorize danceability, energy, and valence into several thresholds, resulting in a potential combination of 27 unique animations. Additionally, the tempo dictates the speed at which these animations are played.

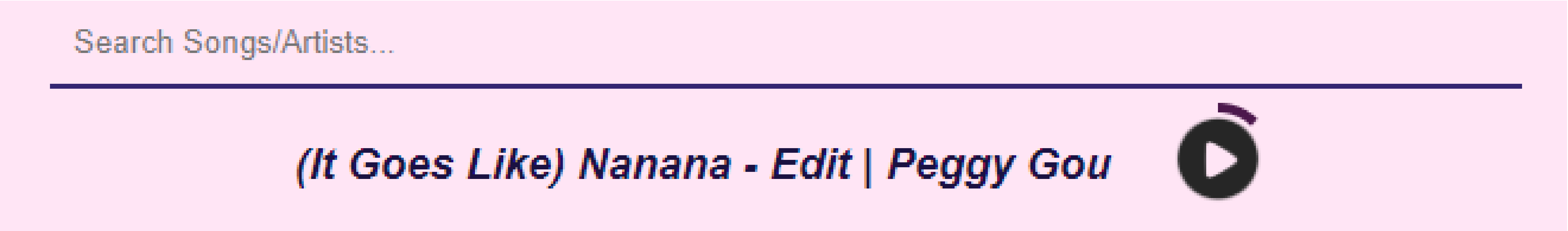
For danceability, the level of skill depicted in the animation ranges from someone with no dance experience to a professional dancer, depending on the song's danceability score. Similarly, the energy metric influences how vigorous the dance appears. And the valence variable affects how happy the dancer appears.

Initially, we planned to use anime.js (found here: anime.js) to animate the stickman directly in SVG format within the HTML. However, the complexity of creating 27 distinct animations by manually programming the movement of SVGs proved too time-consuming. Instead, we turned to Pivot Animator (Pivot Animator), a tool that simplifies the creation of SVG animations through a graphical interface. This application allows for detailed frame-by-frame animation by adjusting the joints of the figure. Once created, these animations can be saved as .piv files for further modification or, more importantly for our project, exported as .webp files for seamless video playback.



Challenges

We have implemented one of the challenges described in Milestone 2. This was playing the song samples of the songs selected by the user, using the Spotify Web API. In detail, when a user clicks on a song from the Set List, an API call is sent for the corresponding track ID. If the song exists in the API, then a play/pause button appears next to the song and the user can listen to the sample. The button also has a progress bar to show the length of the sample. Unfortunately not all songs in the dataset used have a corresponding sample, however, most of them do.



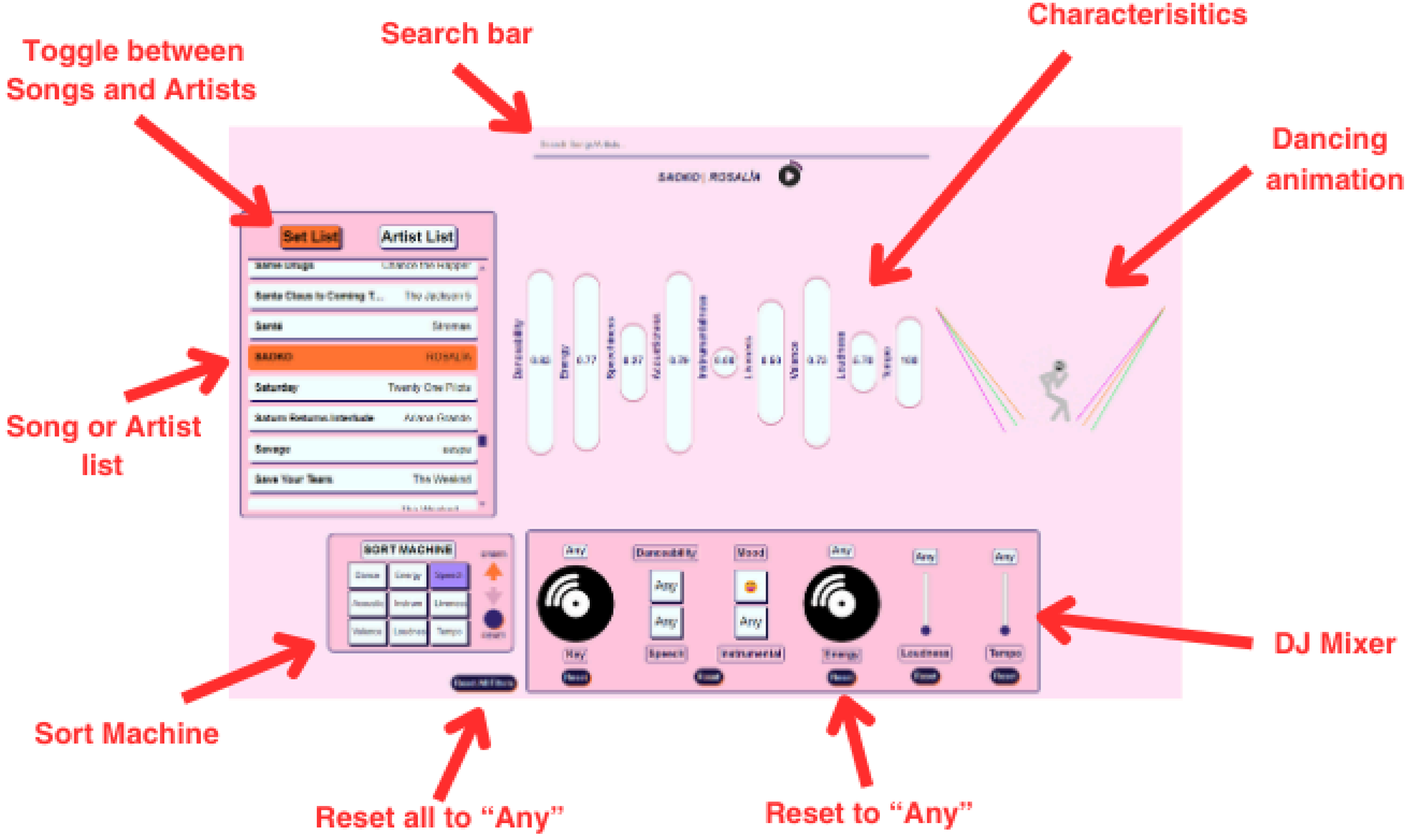
(It Goes Like) Nanana - Edit | Peggy Gou

Changes since Milestone 1

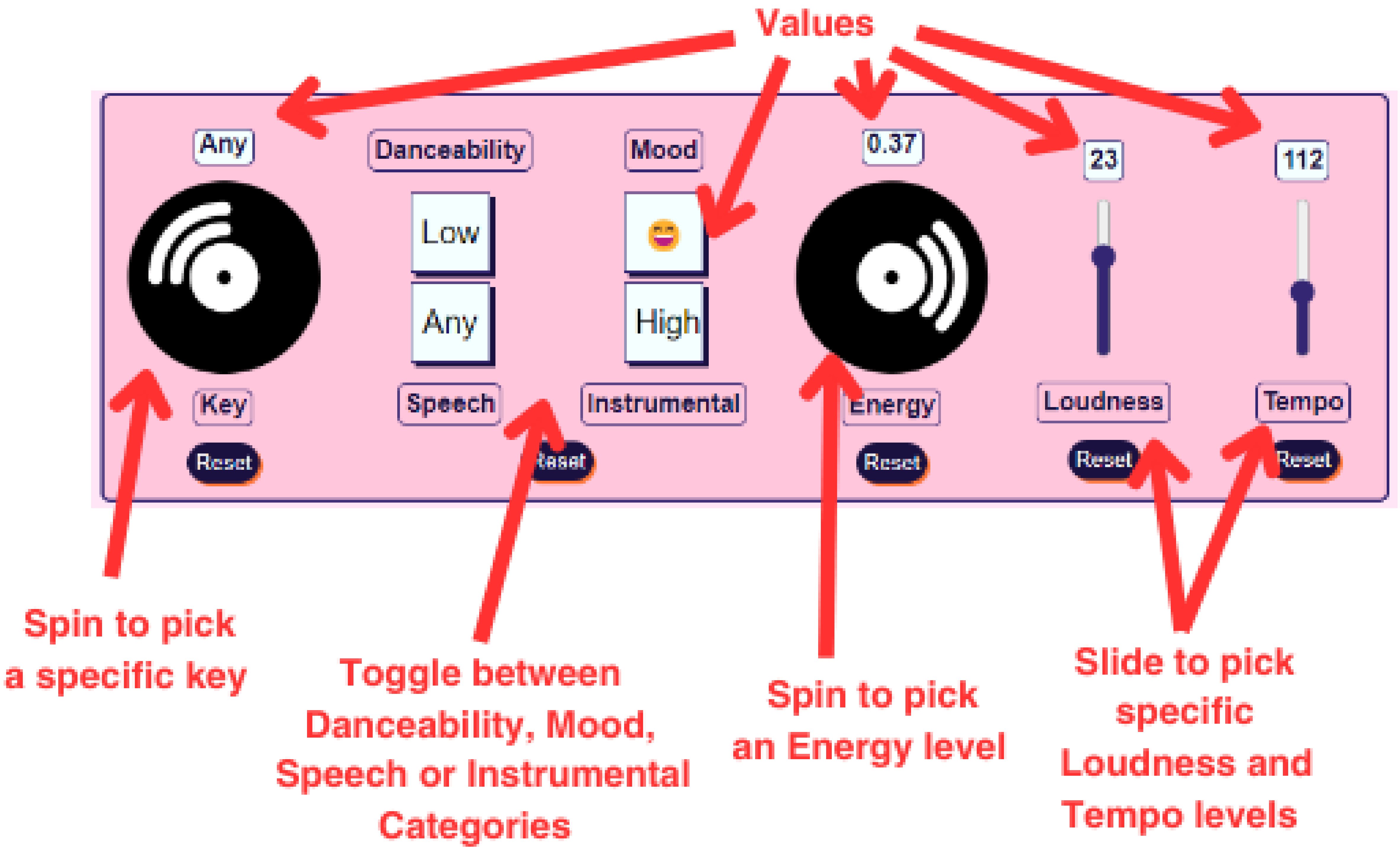
Since Milestone 1, there have been some changes in our implementation, due to functionality and usability reasons. Firstly, we removed some features from the dataset that had missing titles or values, or which were replaceable with similar features,to make our visualization more consistent and easier to understand. Firstly, we decided to incorporate all the songs from the dataset in a scrollable list instead of showing only the top N songs to not limit the amount of data the user can access. The discs now represent filtering instead of scrolling through the song and the song/artist toggle button moved to be on top of the list container to be more visually coherent.

Visual User Manual

General Layout



Setting constraints on the characteristics



Values

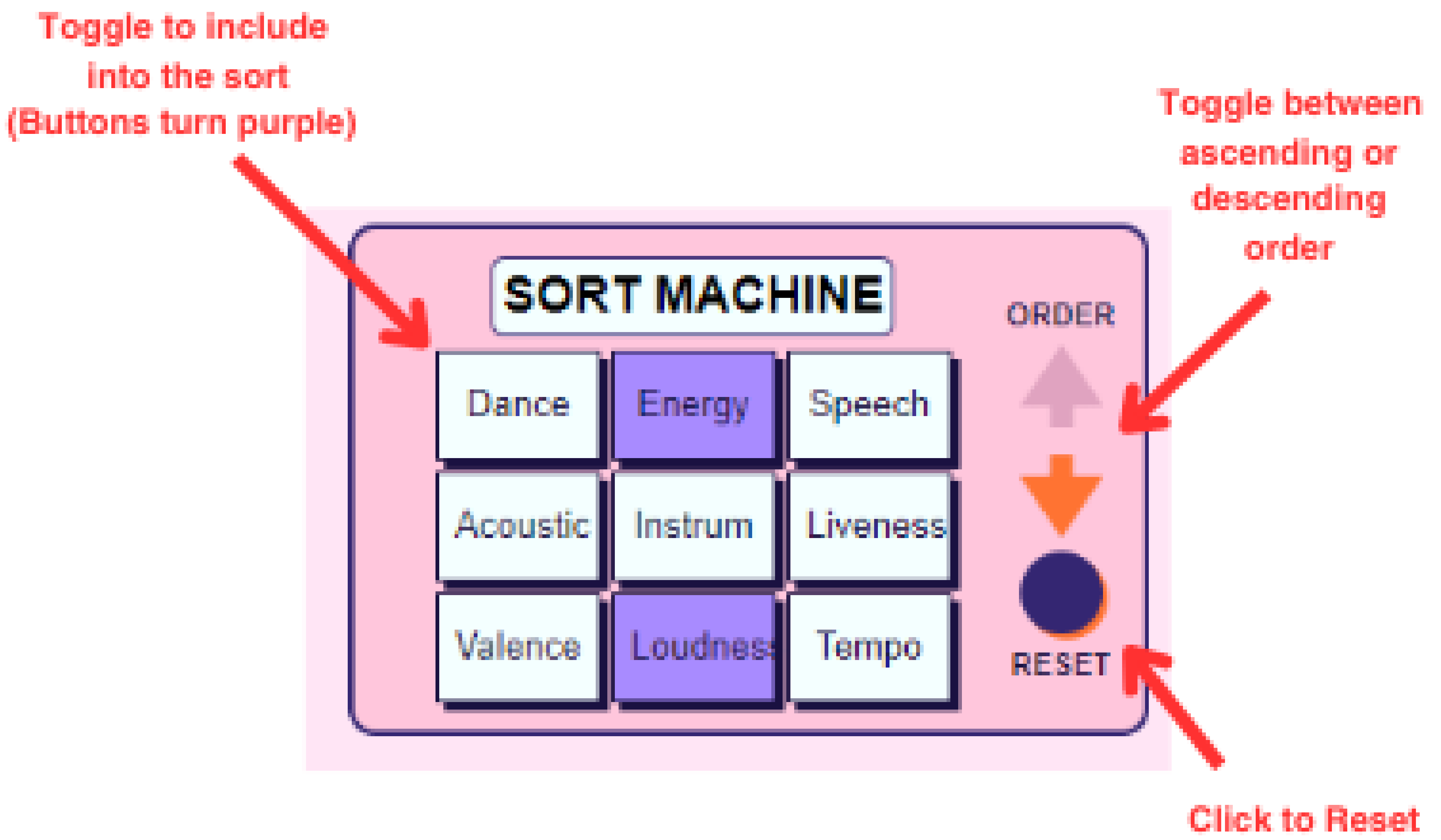
Spin to pick a specific key

Toggle between Danceability, Mood, Speech or Instrumental Categories

Spin to pick an Energy level

Slide to pick specific Loudness and Tempo levels

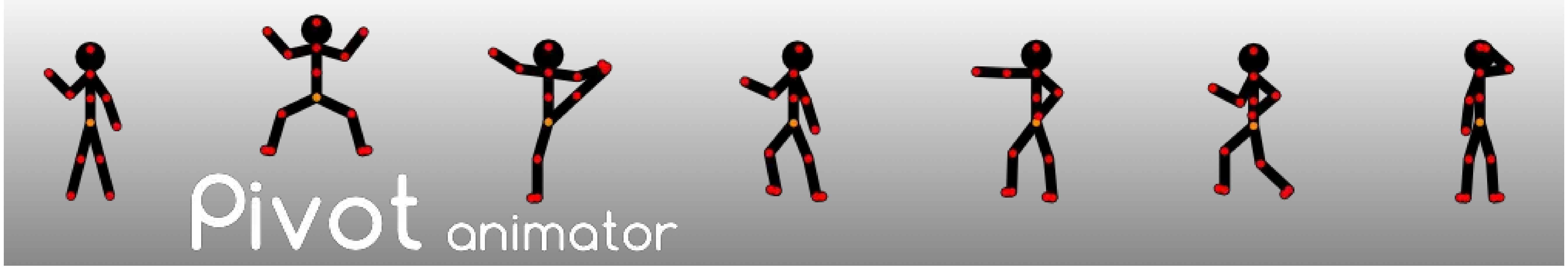
Sort Machine



Toggle to include into the sort (Buttons turn purple)

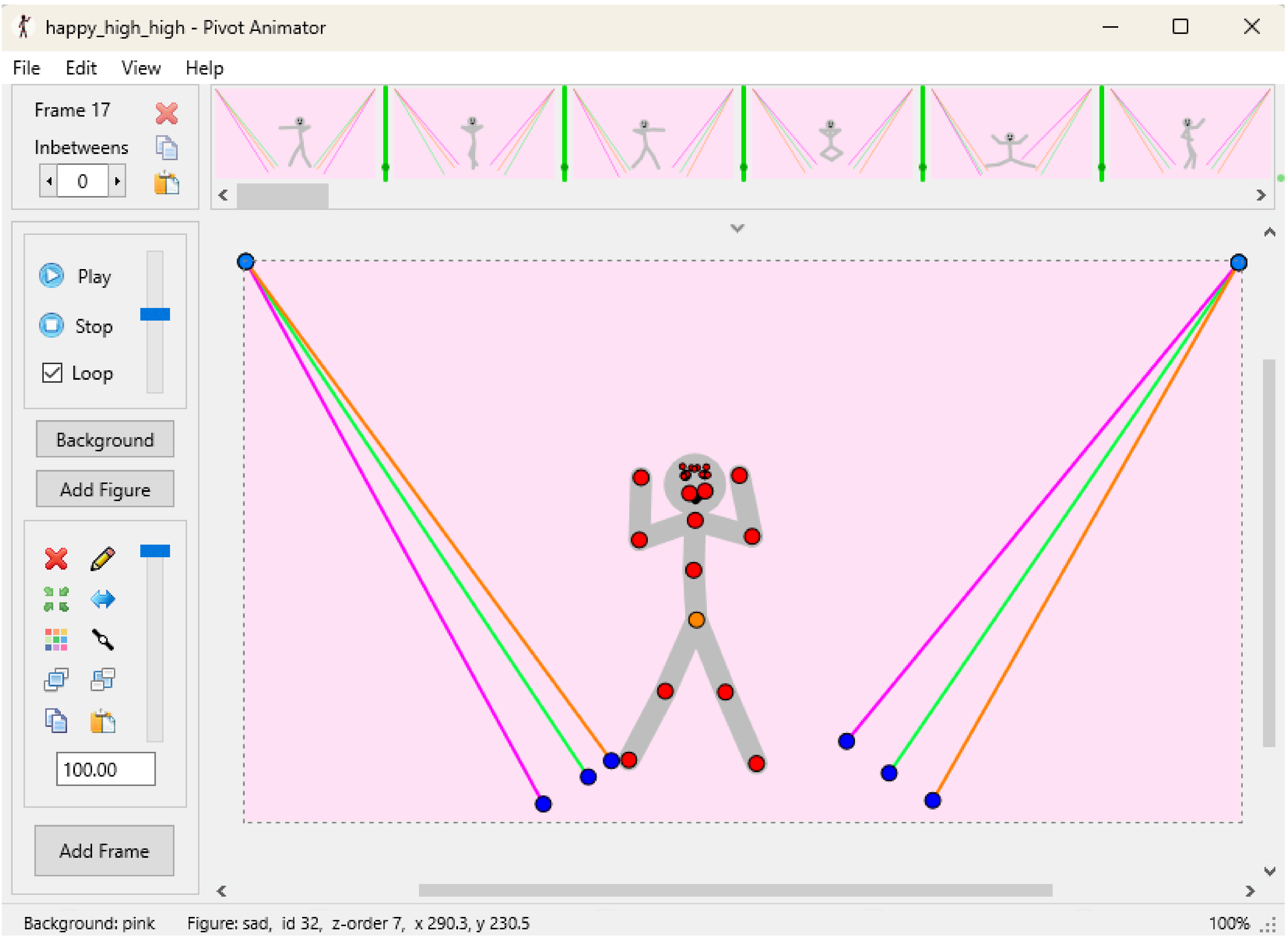
Toggle between ascending or descending order

Click to Reset



Animation

We aim to visually represent each selected song through an animated stickman who dances according to the song's characteristics. The dance moves change based on the song's danceability, energy, valence, and tempo. We categorize danceability, energy, and valence into several thresholds, resulting in a potential combination of 27 unique animations. Additionally, the tempo dictates the speed at which these animations are played.



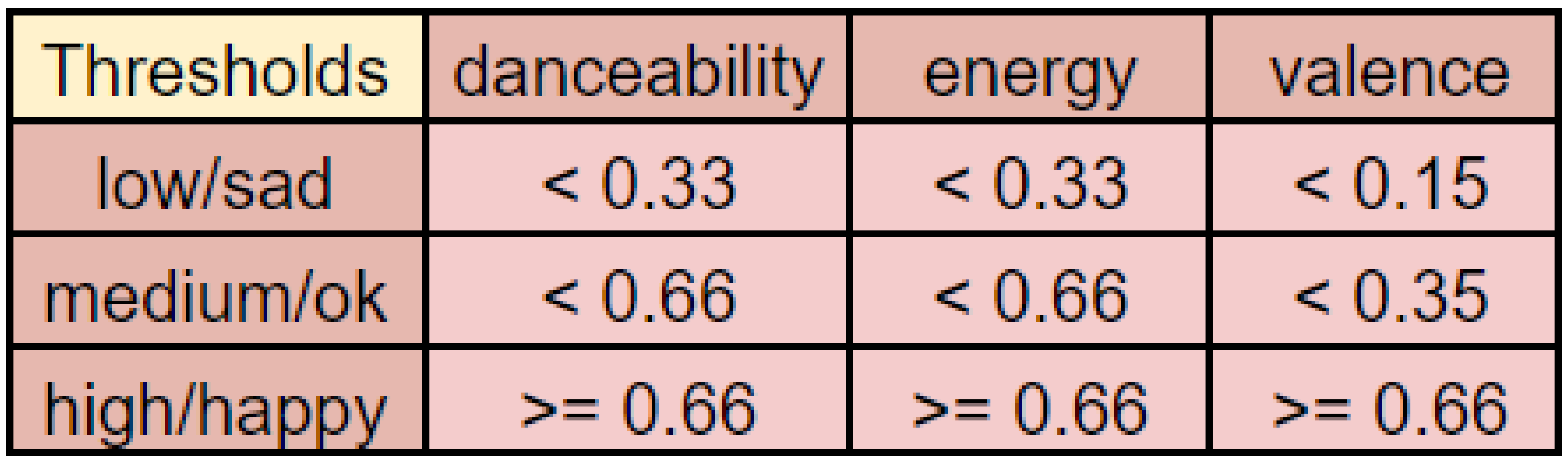
Frame 17

Inbetweens 0

☑ Loop

100.00

Background: pink   Figure: sad, id 32, z-order 7, x 290.3, y 230.5   100%



Thresholds	danceability	energy	valence
low/sad	< 0.33	< 0.33	< 0.15
medium/ok	< 0.66	< 0.66	< 0.35
high/happy	>= 0.66	>= 0.66	>= 0.66

For danceability, the level of skill depicted in the animation ranges from someone with no dance experience to a professional dancer, depending on the song's danceability score. Similarly, the energy metric influences how vigorous the dance appears. And the valence variable affects how happy the dancer appears. Initially, we planned to use anime.js (found here: anime.js) to animate the stickman directly in SVG format within the HTML. However, the complexity of creating 27 distinct animations by manually programming the movement of SVGs proved too time-consuming. Instead, we turned to Pivot Animator (Pivot Animator), a tool that simplifies the creation of SVG animations through a graphical interface. This application allows for detailed frame-by-frame animation by adjusting the joints of the figure. Once created, these animations can be saved as .piv files for further modification or, more importantly for our project, exported as .webp files for seamless video playback.

Peer Assessment:

We all contributed equally to the project and are extremely proud of the work we achieved together. Our collaborative efforts resulted in a project that not only met our goals but also reflected our combined skills and dedication.

Individual Contributions:

Anne-Marie: Focused on writing the project report, developing the csvManipulator.js, styling the website with style.css, crafting the index.html, and designing visual interactions including clickable elements, as well as visual clarity of our website.

Christopher: Responsible for creating the stickman animations, contributing to index.html, managing our GitHub repository, and ensuring all animations were smoothly integrated into the project.

Simon: Took charge of compiling and formatting the project report, editing index.html to ensure clarity and professionalism in its presentation.

Stefan: Worked on csvManipulator.js for data handling, processed the data needed for the project, integrated the Spotify API, and managed data interfacing within index.html, as well as considerable contributions to csvManipulator.

Appendix

Here are some