Opinion

Off the Staff: An Experiment in Visualizing Notes from Music Scores

Nicholas Rougeux1,*

1Web Designer and Artist, Wheeling, IL, USA
*Correspondence: rougeux@gmail.com
https://doi.org/10.1016/j.patter.2020.100055

An introduction to the work of web designer and artist Nick Rougeux, who created the cover art for this edition of *Patterns*. Here he explains his method for visualizing music scores and highlights the importance of exploring data through imagery to create new ways for his audience to engage with the stories told by the numbers.

The talent of reading music has always escaped me, which is a little ironic considering I grew up in a musical family. However, I’ve always enjoyed how sheet music looks so I took a shot at visualizing the notes from musical scores, and the result is a series of posters called “Off the Staff.” The beautiful complexity of sheet music begs for experimental notation, and I’m hardly the first, but I wanted to bring my own interpretation to the process.

The aim was to find a way to represent an entire musical score in a single image—not necessarily as an infographic but something fans of the music would enjoy having on their wall because they’re familiar with the music. My work is meant for those who want to connect with stories in a new way, whether that’s a music score, a modern reproduction of a brochure for antique typographical machinery, or a timeline of spacewalk milestones since the dawn of modern space exploration. My aim is to take an existing structure and visualize the raw data in a radically different way to see what we can gain from a new perspective.

Initial Ideas

I explored a wide range of visualization options, first plotting all the markings from a PDF of a score with the least frequently used in the center to the most frequently used in the outer ring. Then I experimented with treemaps, horizontal plots, triangles, grouping by measure, notes written out as text, radial charts, and timelines.

One of the approaches I take when building a new series is to test my ideas with my audience on Twitter. One repre-

servation that got a good response was built with every measure in “Moonlight Sonata” represented by a “flower.” Petals were arranged based on when notes occur—first note in the center, then send as a larger ring, and so on. Petal color was based on pitch/octave, and the number of petal was based on duration (longer note = more petals in a ring).

Scores as Circles

I finally settled on representing a score as a circle. When I first tweeted it, it quickly became my most popular tweet at the time, and it allowed me to break away from the short efficient five-bar staff representation of notes, showing notes on the same scale.

In each image, an individual dot represents a note in the score of a piece of music. The cover for this edition of *Patterns* is Vivaldi’s “Four Seasons,” but my visualizations range from Mendelssohn’s “Wedding March” to Rimsky-Korsakov’s “Flight of the Bumblebee.” Pitch is indicated by the distance from the center of the image, while the time at which the note occurs is given by the angle from the 12 o’clock position. The size of the dot indicates the duration of the note, and the color of the dot is different for each instrument.

To take a simple illustration, Figure 1 demonstrates the evolution of the poster for “Twinkle Twinkle Little Star.” You can see how single instruments are represented using a single color, and multiple instruments are represented using multiple colors in order to add an extra dimension to the visual.

I created the first series of images in September 2016, parsing the music and painstakingly constructing each representation. I had wanted to extend my visualizations to other pieces of music, but the process had bottlenecks that zapped my motivation. However, in 2017, I was approached by the MuseScore team to partner with them on OpenScore (https://www.kickstarter.com/projects/openscore/openscore-join-the-sheet-music-revolution), a Kickstarter campaign to liberate public domain sheet music by making it available in open formats on any device. They wanted to use the images I generated as cover images for each score. This was an exciting project for me, and I was happy to get involved, but in order to complete the process efficiently, we needed to streamline the process for creating the images, which meant finding a way to eliminate the knots in my method.

Developing the Technique

My original method involved several unnecessary steps, mostly because I was avoiding writing code to do the parsing. I preferred to use existing tools to convert data, even if doing so required extra steps.

The original process was:

1. Export XML from MuseScore.
2. Convert XML to CSV with Luxon:
   1. Use a Windows virtual machine.
5. Being a Mac user, I had to download the tool.
6. The web-based tool had a 4 mb size limit, and several scores had so many notes that resulted in XML files beyond that limit, which meant I had to use the downloadable tool that only worked in Windows. Being a Mac user, I had to use a Windows virtual machine each time to convert XML to CSV.

Patterns 1, June 12, 2020 © 2020

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
files. Plus, I was using a computer that was 7 years old, which meant just turning it on took a while. Booting up Windows was a very slow process.

3. Upload select CSV files to Google Sheets: Copying and pasting tens of thousands of cells into Google Sheets was a slow process and prone to failing—even with the speediest of browsers.

4. Combine CSV files with Google Sheets: The process of combining CSV files with Google Sheets was the most time-consuming step of the original process and had the risk for inaccuracy. I used Google Sheets because it was easier to get the lookup formulas working to combine the three CSV files for each score (notes, measures, and parts) into one final file.

5. Import CSV into NodeBox to generate images.

When MuseScore approached me, a member of their team, Peter Jonas, was kind enough to write a script to eliminate Google Sheets entirely and the risk of inaccuracy. The script, midicsv-process (https://github.com/shoogle/midicsv-process), not only parses data from a MIDI file but accounts for tempo changes so each note has an accurate timestamp.

The new process is:

1. Export MIDI from MuseScore.
2. Generate CSV from MIDI file with midicsv-process.
3. Import CSV into NodeBox to generate images.

The result is a much cleaner set of data, complete with note names like D4# (a D# in the fourth octave) for verification. I can’t thank Peter enough for writing this script. Thanks, Peter!

Prior to this script, I wasn’t able to plot notes based on when they were played because timestamps weren’t included in the XML export. The closest to a timestamp was a value that indicated how far each note was placed in each measure of the sheet music. That meant I could tell when notes were played simultaneously, but tempo changes weren’t factored in, and I had to cobble together some concatenation formulas to generate timestamp-like values based on several values for each note.

Once I had those timestamp-like values, I chose to space them out equidistant around a circle. While this produced nice graphics, they weren’t as accurate as I wish they were. The difference made by having the true timestamps in the XML export is significant for the visualizations of Vivaldi’s “Four Seasons”; to me it feels like night and day. The first images were very chaotic and contained the risk of calculation errors. The second set of images, with the true timestamps, are much more structured and consistent. Patterns of the music are more apparent, and overall, they’re more visually appealing.

Animations

Ever since I created the first images, I have wanted to see them in motion, partly because I thought they would be fun to watch, partly for debugging them to make sure they’re accurate, but also so others could understand them better. Having timestamps for each note made animations a reality. With some extra work (and more than a few headaches), I was able to bring the illustrations to life and to show how the images were made. Combining timestamps with durations of notes even allowed me to make each circle/note grow for as long as it’s played.

MuseScore generates MP3 files for each score uploaded to their community, so once an animation is generated, I was able to sync it with the audio to produce the final animations. The animations really bring these images to life, and I enjoy watching each one. All are available in my “Off the Staff” YouTube playlist.

NodeBox

All of these animations were created in NodeBox and combined with the audio in Premiere. NodeBox has a great feature to use animation frame numbers as values (Figure 2). Videos are exported at 30 frames per second (fps). Knowing this, I was able to feed in the note data and determine which notes should be visible and even how much of it was played at any given frame. Then, it was just a matter of rendering all the frames. I wanted the animations to be high quality, so they’re all rendered at 30 fps at 4K resolution, which means each one does take several h and sometimes days to render.

Future Plans

I’ve been working with MuseScore to create many more images for OpenScore since the Kickstarter succeeded and continue to do so as more scores become available. I can’t wait to see what kinds of images come out of them. I will also continue to build new series using other data sources, and I hope you will
Nick Rougeux is a Chicago-based web designer and artist. From a young age, his fascination with the digital world led to a healthy obsession with data visualization and fractal artwork, which has been published in publications and exhibitions around the world. Since his start in the digital art and online worlds, simplicity and clarity have played key roles in what he has created. His website, c82.net, was born out of a desire to display the digital art he created. Over time, it evolved into a playground for personal experiments, a vehicle for voicing his opinion on art and the web, and a place to chronicle the processes for creating his projects.

WEB RESOURCES

C82, https://c82.net/midicsv-process, https://github.com/shoogle/midicsv-process

OpenScore, https://www.kickstarter.com/projects/openscore/openscore-join-the-sheet-music-revolution