Starry Night Sequencer

Stephen Wu

Design 3305

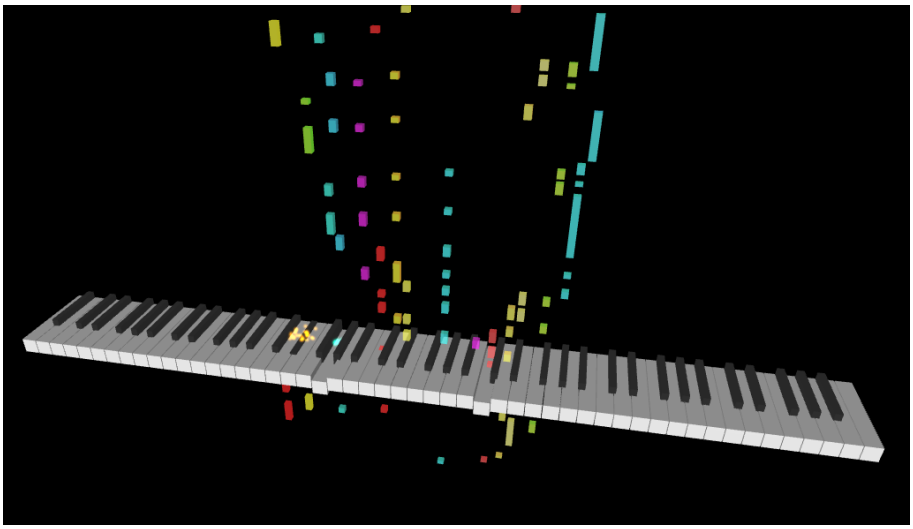
December 3, 2016

https://wustep.github.io/midi/player.html

Background

As an avid fan of programming and music, I sought to combine my interests into a web application that would visualize music. One universal format for sharing aspects of music is MIDI files, which contains information about notes, tempo, duration, volume, and much more. I sought to find libraries or frameworks online that could help me program visualizations of music.

I quickly found MIDI.js[1], a library by Michael Deal, that seemed to have some really interesting visualization and sequencing capabilities. The example immediately caught my attention, showing the library was capable for visualizing individual notes and coloring them, as seen on the right, and even more intricate 3-D animations as seen below[2].



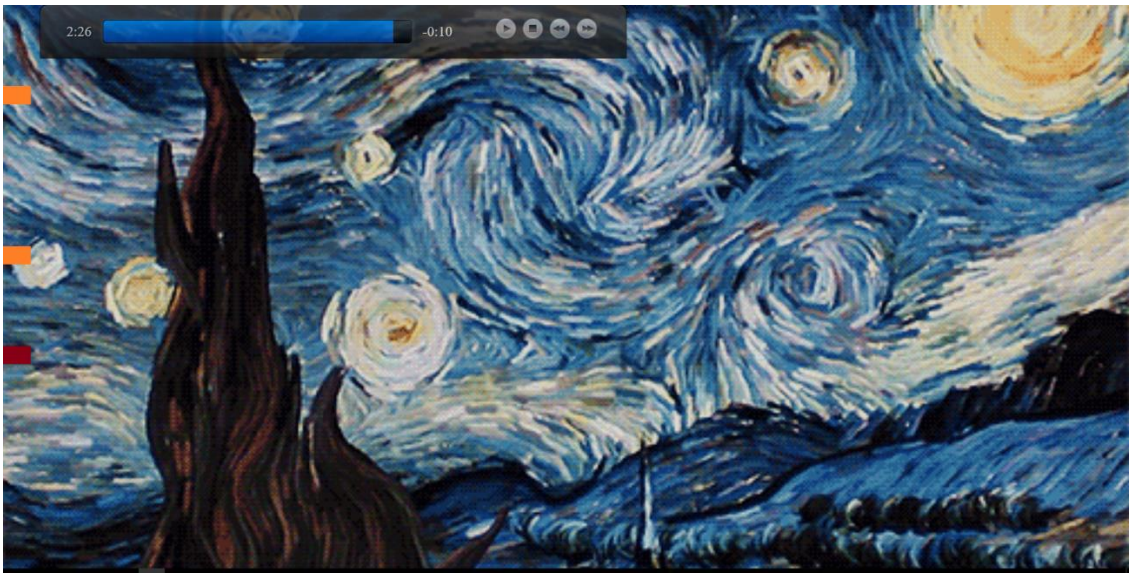
From seeing the colored piano keys example, I decided to use that example and build upon it, initially deciding on having a vertical key visualization paired with various backgrounds and songs. The background I decided on was an animated .GIF version of Van Gogh's *Starry Night*[3], which I unfortunately could not find an original source for.

[1] MIDI.js – Sequencing in JavaScript – https://galactic.ink/midi-js/
[2] Euphony 3D Piano – http://qiao.github.com/euphony/
[3] Moving Van Gogh - http://weheartit.com/entry/146867318/via/yulyet_loveinthewronglane

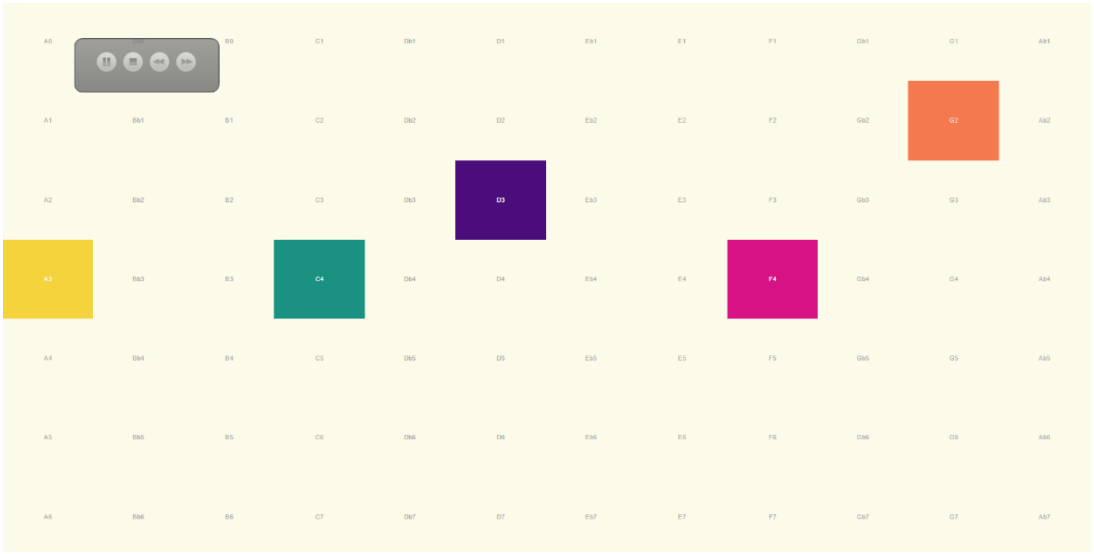
Process

From there, I initially built a simple combination of the two elements—The Van Gogh and the side piano keys—messing around with CSS styling and HTML elements. I started out with the default pieces provided by MIDI.js's example. Sinding's *Rustles of Spring Op-32 No-3*, Tchaikovsky's *Arabian Dance*, Tchaikovsky's *Waltz of the Flowers*, Granados's *Oriental*, and Joplin's *The Entertainer*. So the songs would play and the side piano keys would play colors based on the notes played in the songs.

The result looked something like the picture below. As the song played, the left side would light up. The colors followed something of a rainbow pattern (from notes A to G). When the note was released, the color would be released as well.



But I wanted to go further and really intertwine the music with the space, so I modified the keys to be a grid system, also seeing how the grids looked without a background.



I noticed that the Starry Night animation was both fast and jerky, so I used an online tool from EzGif.com[4] to slow down the animation by 25% and loop fluidly through reversal of the .GIF file. The tool quickly helped remove the jerkiness of the animation which can be seen in the original .GIF and improved the visualization by slowing down the speed of the frames.

After that, I decided to modify the key colors to be more fitting of Van Gogh, with a resource found online, a Van Gogh color pack gallery by Gabe Mott[5].

For each of the 12 keys, I designed a map of specific colors from the palette above, using the tool HTML Image Color Picker[6] to get the hexadecimal codes. This led to more fitting colors for the background.

```
var starryNightColors = ["#02092B", "#374F74", "#6D96BE", "#616C67", "#91AEAA",
"#766B3C", "#A0A778", "#A28A31", "#C6C267", "#CFAA28", "#EDDC57", "#FDF66F"];
```

[4] GIF Optimizer – http://ezgif.com/optimize
[5] huedoku Color Pack: Van Gogh Gallery – http://huedoku.com/huedoku-color-pack-van-gogh-gallery/
[6] HTML Image Color Picker – http://imagecolorpicker.net/

Result during a song:

From there, I decided to look into other possible visualization ideas. After some consulting and trial and error, I realized that it might be interesting to show the notes staying in place after being played, creating columns of notes that had been played. I built a draggable player that could access this second visualization scheme. The fifth button on the panel below allowed users to swap between the normal view (disappearing colors) and this new view (colors that stayed in place throughout the song).

Result of second visualization scheme during a song:

Piece: Joplin's *The Entertainer*

The less transparent boxes are notes that are being played at that frame of the song. This led to a more interesting visualization as the user could see which notes of the 88 keys were played in all. At the end of the piece, a user could see all the notes that were and weren't played, as well as which scales were more frequently used.

Finally, I selected a new set list by finding MIDI files from various sources online and converting them to Base64 (as needed for the program) through an online Encoder / Decoder[7].

The final song list became:

- Tchaikovsky – Arabian Dance
- Shirō Sagisu – Will of the Heart[8]
- Erik Satie – Gymnopedie[9]
- Debussy – Reverie[10]

There was no exact method for selecting the songs. The first was one of the defaults from MIDI.js, and the 2nd and 4th were selected by friends. They simply seemed fitting to the backdrop and had visualizations that were improved by extended durations of notes.

Conclusion

Prompted by the results of the project, I hope to continue experimenting with different methods, backdrops, songs, and shapes to visualize music in the future. The project demonstrates one of many ways to visualize MIDI files through web applications, and hopefully shows a unique way of listening and looking at music, enhancing what the listener feels or imagines.

The final result can be found at https://wustep.github.io/midi/player.html and its source at https://github.com/wustep/midi. I hope you enjoyed!

- Stephen Wu

[7] Base 64 Encoder / Decoder – http://www.freeformatter.com/base64-encoder.html
[8] Will of the Heart MIDI – http://www.midishrine.com/index.php?id=189
[9] Gymnopedie MIDI – http://www.kunstderfuge.com/satie.htm
[10] Reverie MIDI – https://tirolmusic.blogspot.com/2007/10/reverie.html