ALPHA RELEASE

FretJS - Fretboard Visualization in Vanilla JavaScript

Library Usage

FretJS is a visualizer for fretted string instruments such as guitars, basses, violins. It renders notes as dots on the fretboard. It is simple in that dots are rendered by string/row position like a matrix, and it is up to the user to elaborate, making the library very flexible. Developers and musicians can use the library to draw chord diagrams, play live songs from midi files, even use as digital input to a touch guitar game.

Features

FretJS is currently capable of drawing any number of notes concurrently and/or sequentially. All elements are available to be styled via user specifications. Nice animations exist for notes lifespan like fade-out for live animating. Callbacks are available onBirth and onDeath of a dot to allow a user to do cool things like play the notes tone, or provide additional complexity. Additional non-DOM helper functions are included to do cool things easily with the library like drawing notes within the CAGED system by filtering notes by fret indices, playing MIDI files, adding synesthesia-inducing colours to notes, etc!

Web App

Heroku Link: https://intense-caverns-82434.herokuapp.com/

The web app features some immediately useful cases of the library.

1. **Render a chord natively with library.** Using just the library api we render the individual notes of a c major chord

2. **Play any scale.** Choose any scale and root note and watch it play live up the fretboard. Callbacks are used to play sound, and the Tonal.js library is used to get musical data.

3. **CAGED System.** Filter out notes by fret indices. This is useful to filter notes into a specific shape of the CAGED system on a particular area of the fretboard. *Only applicable to standard tuning.*

4. **Play MIDI file:** View a live animation of a song in a MIDI file played on the guitar fretboard. Very useful for seeing how to play a song since MIDI files are so prevalent and there exist countless piano visualizers but few guitar visualizers for the task,

JS Objects

A Fretboard Class Object is created for each fretboard visualization you want. This class is instantiated with the id of the figure for the visualization <figure id="jimmy"/>. The Fretboard class holds all library functions to do with drawing the fretboard and dots. It can be styled. The rendering of dots is done through the fretboard figure being subdivided like a matrix by its strings and frets and drawing a dot by given coordinates.

The Dot Class Object is the other object and represents a note. At minimum, a dot is of the form {stringIndex: int, fretIndex: int} and is sufficient for fretboard.drawDot() to render it with default styling. The Dot Class object also has property values that provide more functionality is dot.dotFill which specifies dot color and dot.noteName which specifies text rendered upon dot. This dot class is passed into the onBirth and onDeath callback of fretboard.drawDot() so providing non-library-specific properties like dot.pitch can help out to play the tone onBirth.

API

- fretboard = Fretboard("#id of figure element")

Instantiate an instance of the Fretboard class to control every figure with a fretboard visual

- fretboard.drawBoard()

Render fretboard in figure

- fretboard.drawDot(dot, lifespan, onBirth=(dot)=>{}, onDeath=(dot)=>{})

Render dot for lifespan (if provided) with callbacks triggered during lifecycle

- fretboard.drawDots(dots, lifespan, onBirth=(dot)=>{}, onDeath=(dot)=>{})

Render list of dots

- fretboard.drawStrings()

Draw strings on fretboard

- fretboard.eraseDots()

Clear all dots from fretboard

Moving Forward

- Richer fretboard visualization with more customization details.
    - Adding details like numbering frets
    - Adding shadows and borders
    - Potentially animating string motion

- Improving live animation functionality and midi file support
    - Add visualizations for incoming midi notes, to visualize time dimension