

Estudi i aplicació de Tone JS

Roger Hurtado Manzano

PMUD - 2024/25

Introducció i objectius

ToneJS és una llibreria de JavaScript centrada en la incorporació d'elements i temes musicals a la web. Abans de fer aquesta recerca no coneixia la llibreria, però em va cridar l'atenció la possibilitat de treballar amb una eina relacionada amb la música, ja que és un tema que he gaudit tota la meua vida. Escolto una gran quantitat de música molt variada, toco la guitarra desde que tinc memòria i no fa gaire he començat a aprendre a tocar el piano, per aquest motiu, apart de l'objectiu principal de recerca, he escollit com a demo pràctica la programació d'un teclat web utilitzant els recursos facilitats per aquesta llibreria.

Aprofitaré aquest treball per a entrar una mica més en profunditat en què és ToneJS i que ofereix, quin és el seu origen, qui és l'autor darrere de la llibreria, i alguns aspectes més tècnics com quines són algunes de les eines principals que podem utilitzar i com implementar-les. Degut a que estem tractant amb una llibreria, la recerca d'informació teòrica no és molt extensa, a més a més, ToneJS no és precisament una llibreria molt gran. Tenint això en compte, tractaré alguns exemples de codi en favor a evitar que aquest treball tingui una extensió massa curta. Complementaré aquests exemples amb l'explicació de com els he aprofitat en el meu cas particular per a dotar de més possibilitats el projecte del piano web, i així veure casos d'ús reals de la llibreria i fer més interessants les parts de codi.

Què és ToneJS?

ToneJS és una llibreria de JavaScript i un framework de Web Audio, que facilita la creació de música interactiva al navegador. Escrita en TypeScript, va ser llançada oficialment al 2014. Les seves aplicacions van més enllà de simplement incorporar música de fons o mostrar un reproductor d'àudio, la propietat principal que ofereix és la interactivitat, la capacitat d'associar els elements d'una pàgina web amb el so que volguem, ja sigui una nota, un acord, o un àudio qualsevol. Això ens dona una gran varietat de possibilitats, podem programar des d'un complex joc musical, fins a una simple web on cada element reproduceix un so al fer click a sobre.

La llibreria està basada en Web Audio API, una eina amb el mateix objectiu, però més complicada d'utilitzar i sense moltes opcions que ToneJS té programades per a evitar repetir l'escriptura del mateix codi constantment. Una de les avantatges que ofereix, és que està pensat per a ser amigable tant amb músics com amb programadors d'àudio web. Incorpora diverses eines que veurem més endavant com l'ús de sintetitzadors, sincronització global d'esdeveniments, samplers, aplicació d'efectes i molt més.

Autor de la llibreria

L'encarregat del desenvolupament d'aquesta llibreria és Yotamm Mann, músic i creador d'instruments. Va estudiar música i ciències de la computació al Center for New Music and Audio Technologies de la Universitat de Califòrnia a Berkeley. Un dels seus objectius és fomentar el progrés de la música mitjançant la tecnologia. El motiu que va inspirar la creació de ToneJS, va ser agilitzar el procés d'implementació d'un joc musical anomenat ECHO, podem provar el seu funcionament a la web oficial <http://tonenotone.com/ECHO/>. Inicialment utilitzava Web Audio API, però es va adonar que havia de repetir constantment molts fragment de codi i que la API no utilitzava tot el potencial que podia tenir. Així va començar a desenvolupar ToneJS, i més endavant va realitzar un nou projecte que va funcionar com a test del funcionament de l'API a gran escala, podem trobar aquest treball anomenat Jazz Computer a la web omònima <http://jazz.computer/>.

Github pages

Una eina molt útil per a hostejar una pàgina web que no inclou backend és github pages. L'autor de ToneJS utilitza aquest servei amb la pàgina web oficial de la llibreria (<https://tonejs.github.io/>) i altres projectes. Aprofitant que és un servei gratuït, l'incorporarem al nostre projecte per a que qualsevol persona pugui accedir al projecte del piano web i a una altra eina que veurem més endavant.

Repositoris addicionals

A part de la llibreria principal, disposem d'un conjunt extra de repositoris per a complementar el seu ús. Tant aquests extres com ToneJS, els podem trobar a github a l'enllaç <https://github.com/Tonejs>. Els complements amb més utilitat són el repository d'audio i la llibreria MIDI, ambdós els utilitzarem en el projecte del piano web.

Repositori d'àudio

El principal element de ToneJS és la música, però no sempre tenim la capacitat ni el temps per a crear o buscar els nostres instruments o sons. En comptes d'afegir els nostres fitxers mp3 al projecte, ToneJS ens proporciona un repositori complet de diferents tipus d'audio, escales musicals en diversos instruments, efectes sonors i sorolls de fons.

Per al projecte del piano web, utilitzarem uns fitxers que contenen les notes d'una octava, amb els sostinguts i bemols, d'un piano. No és estrictament necessari tenir totes les notes que volem utilitzar, com veurem més endavant, el propi ToneJS inclou una opció que calcula totes les notes a partir d'un o més fitxers. Podem accedir a aquests fitxers mitjançant enllaços, ja sigui una carpeta sencera o un arxiu en particular. Trobem els audios del piano i altres que utilitzarem per a veure exemples a les seves respectives direccions, <https://tonejs.github.io/audio/salamander/> i <https://tonejs.github.io/audio/loop/chords.mp3>.

Llibreria MIDI

Els fitxers d'extensió *.midi* són molt utilitzats per a grabacions d'instruments musicals electrònics i en el context de la música en general, però ToneJS no està preparat per a llegir aquest tipus d'arxius, per aquest motiu, podem utilitzar una altra llibreria que ens ofereix ToneJS anomenada MIDI, podem trobar tota la informació d'aquesta llibreria a <https://github.com/Tonejs/MIDI>.

La funció principal de MIDI és proporcionar una eina per a codificar el contingut d'un arxiu *.midi* a un format JSON que ToneJS és capaç d'interpretar, permetent així que poguem reproduir aquests fitxers al navegador o realitzar qualsevol altre funció que necessitem. Això és possible ja que el que ens retorna aquesta funció és tot el conjunt de notes del fitxer, i la seva separació temporal.

Farem ús d'aquesta funció en el piano web per a afegir la possibilitat de reproduir una cançó de manera automàtica al clicar l'opció corresponent, la peça que farem servir és el preludi nº1 en Do major de Bach. MIDI ens proporciona aquest fitxer a l'enllaç https://github.com/Tonejs/Midi/tree/master/examples/bach_846.midi.

Exemples de codi

De la mateixa manera que amb qualsevol altre llibreria, podem utilitzar ToneJS amb node escrivint a un terminal `npm install tone` i afegint-lo al fitxer Java Script amb `import * as Tone from "tone";`. O bé podem utilitzar un Content Delivery Network, el proporcionat a la web oficial és <https://unpkg.com/tone>, tot i que podem trobar la llibreria a més llocs.

Des de abril de 2018 els navegadors no permeten la reproducció automàtica d'audio si no s'ha interactuat prèviament amb la pàgina, ja sigui fer click o premer una tecla. Per això mateix, es recomana incloure al fitxer `.js` la crida `await Tone.loaded();` per a assegurar que tot el contingut carrega de manera correcta un cop s'ha realitzat la interacció.

A continuació veurem alguns exemples de com utilitzar ToneJS. Per a veure el resultat de cada fragment de codi sense haver de crear i executar un arxiu cada vegada, he programat una web senzilla que permet executar el que peguem dins del quadre de text. Podem accedir-hi mitjançant l'enllaç a github pages <https://nero-xii.github.io/ToneJS-Research-Work/test.html>.

Hello Tone

La funció més bàsica que podem fer és la de generar un so, el "Hello World" d'aquesta llibreria. El primer és definir un nou instrument, un sintetitzador és la opció més bàsica que incorpora per defecte ToneJS, i ara tenim dues opcions. La primera és declarar la nota i el temps de reproducció amb un "triggerAttackRelease". Per a aquest cas, farem sonar un C4, que correspon la 4^a nota de Do del piano, durant 1/8 part d'una nota rodona, una corchea.

La segona opció consisteix en declarar un instant amb "now()", fer sonar la nota en aquell instant amb "triggerAttack", i després aturar la reproducció amb "triggerRelease" en l'instant now+t, sent t el nombre de segons que nosaltres vulguem.

En el piano web utilitzarem la primera opció per a simular l'efecte d'apretar una tecla, ja sigui al clicar sobre ella o al presionar la tecla que li assignem del teclat.

Sampler

Si anem un pas més enllà, trobem l'opció de definir els nostres propis instruments. Això ho podem fer utilitzant un "sampler", on especificarem la nota a la que correspon cada arxiu d'audio que utilitzem, la duració base de la nota (release) i la direcció a partir de la qual llegim els fitxers, en aquest cas, utilitzarem l'enllaç al repositori d'audio per a preparar les notes del piano web. Per a provar-ho executem un "triggerAttackRelease" fent l'acord de mi menor i canviant la duració a 4 segons.

Si ens fixem en el codi, podem veure que de les tres notes de l'acord, només coincideix una amb les que hem declarat prèviament (Eb == D#), però això no suposa un inconvenient, ni tan sols en el cas que cap de les notes fos coincident. Això és degut a que el sampler realitza un procés d'interpolació a partir de les notes que rep, per a així generar les que falten. Tot i així, s'ha comprovat que és recomanable utilitzar un mínim de 4 notes per a tocar notes d'una octava per sobre i per sota, sinò tot i que la nota resultant serà correcta, pot no assemblar-se gaire al instrument que volem recrear.

Effects

Per a complementar els sons que reproduïm, podem fer ús de diferents efectes. Aquests efectes són una part essencial de la música, els trobem en els amplificadors i en programes d'edició, inclús existeixen els pedals, que són eines dedicades a la manipulació del so que reben, aplicant tot tipus d'efectes. Una manera senzilla de provar-ho amb ToneJS, és utilitzant un "player", que s'encarrega de reproduir el so que li assignem. Utilitzarem el fitxer *chords.mp3* que ens proporciona el repositori d'audio. Després, només cal especificar l'efecte i connectar el player a aquest.

Signals

Els signals són un concepte que pot semblar una mica més complicat d'entendre. Una manera de veure'ls és com un llapis que dibuixa una línia en un gràfic. Podem controlar com puja o baixa aquesta línia (freqüència, volum, etc.), la velocitat amb què ho fa (temps) i la seva forma (lineal, exponencial, etc.). Per a provar el seu funcionament farem ús d'un "oscil·lador" per a modificar la freqüència del so. Només necessitem definir l'oscil·lador, assignar-li una nota i especificar quina és la nota a la que volem arribar i el temps que trigarà.

MIDI

Anteriorment hem explicat quia era la funció de la llibreria MIDI, ara anem a veure com utilitzar-la amb un exemple bàsic.

Primer hem de llegir el fitxer des de una direcció, en aquest cas cal que tinguem el fitxer descarregat, no ens serveix simplement incloure la ruta al repositori de github. Després llegim totes les pistes que tingui l'arxiu i per a cada una busquem les notes. Finalment executem un "triggerAttackRelease" per a cada nota. Podem observar que a part de la nota i la duració, MIDI ens proporciona l'instant en que s'ha de reproduir i la seva velocitat, i això ho podem afegir al disparador. Aquest és un codi molt bàsic que reproduceix un arxiu tal qual el llegeix, però si vulguéssim podríem jugar una mica amb el seu funcionament.

Web piano

Una vegada em vist el que podem fer amb ToneJS, arriba el moment de posar-ho en pràctica amb un projecte una mica més elaborat. Com s'ha comentat a la introducció, la idea és crear un teclat de piano web interactiu. Passant ràpidament per sobre, he dissenyat un teclat bàsic mitjançant html i css, en el qual trobem 21 notes blanques i 15 negres amb un identificador únic cada una que correspon a la nota que tenen assignada. Just a sobre del teclat hi ha una barra amb diferents opcions que aporten algunes funcions extres, i tant aquests botons com les tecles tenen una petita animació per a ser més agradables a la vista. Fent ús de github pages, qualsevol pot accedir al projecte a través de l'enllaç <https://nero-xii.github.io/ToneJS-Research-Work/>.

Podem interactuar amb el piano tant amb el teclat com amb el ratolí. Per a que cada tecla faci un soroll diferent, utilitzem "triggerAttackRelease" especificant la nota que s'ha activat. El so base es el de un piano, que hem recreat utilitzant un sampler, però si fem click a la barra d'opcions on fica *Piano*, canviarà a *Casio* i el so passarà a ser el de un sintetitzador de la mateixa marca, també generat amb un sampler.

L'opció *Clair de Lune*, reproduïx l'inici de la cançó omònima de Debussy. Per a fer-ho, he definit una variable que inclou cada nota i quan s'ha de reproduir. Això m'ha permès programar que el piano fes l'animació de tocar cada nota que sona per simular que s'està tocant ell mateix.

Per a l'opció de *Bach 846*, el resultat és la reproducció de la peça que hem comentat al parlar de la llibreria MIDI. En aquest cas no es veu reflectit el que està sonant a les tecles de piano, però estic segur que amb una mica més d'investigació es pot aconseguir.

Finalment trobem l'opció *Metronome*, que activa un metrònom creat a partir d'un bucle de "triggerAttackRelease" i algunes eines més. L'ús del metrònom per a practicar una peça o simplement saber quina és la velocitat de cada nota és utilitzat per la majoria de músics. De base tenim un bpm (Beats Per Minute) de 120, però podem modificar-ho a través de la consola del navegador amb la comanda `Tone.Transport.bpm.value = newValue;`, on *newValue* correspon al valor que indica la peça o el que nosaltres volgüem.

Conclusions

Aquest ha estat un treball que he disfrutat realitzant, com he dit a la introducció, m'agrada molt la música, i aprendre a utilitzar aquesta eina m'ha semblat molt interessant.

En la meua opinió, ToneJS és una llibreria molt útil per a afegir interactivitat musical a una pàgina web, aporta una gran varietat d'opcions i no és difícil d'utilitzar. La documentació sobre la llibreria és molt completa, i el fet d'incloure altres eines com MIDI o el repositori d'audio demostra que l'autor tenia un gran interès i motivació a l'hora de crear ToneJS.

Si haguèssim de buscar un punt negatiu, comentaria que són necessaris alguns coeixements musicals bàsics per a poder utilitzar totes les funcions que implementa la llibreria, i pot ser que alguns conceptes siguin una mica difícils d'entendre a la primera, però com he dit, la documentació és molt completa, i en qualsevol cas, podem trobar una quantitat il·limitada d'informació de teoria musical i altres aspectes que abarca la llibreria a través d'internet, sense cap problema ni dificultat.