

# ToneJS

...

CREATE MUSIC IN YOUR BROWSER

<https://github.com/Nero-XII/ToneJS-Research-Work>

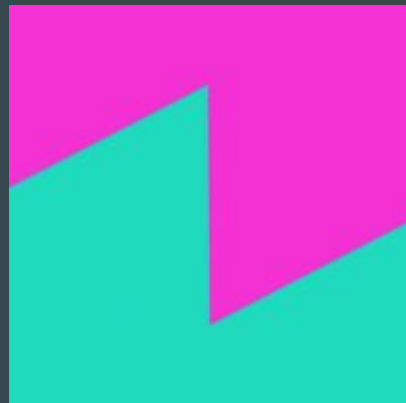
ToneJS-slides.pdf

# Index

- Que és ToneJS?
- Autor de la llibreria
- Instal·lació
- Hello Tone
- Audio Library
- Sampler Example
- Effects Example
- Signals Example
- MIDI Library
- Web Piano

# Que és ToneJS?

- Web Audio framework per a crear música interactiva al navegador
- Basat en Web Audio API
- Amigable amb músics i programadors d'audio web
- Fet amb TypeScript
- Publicada al 2014
- Sincronització global, sintetitzadors, samplers, efectes...



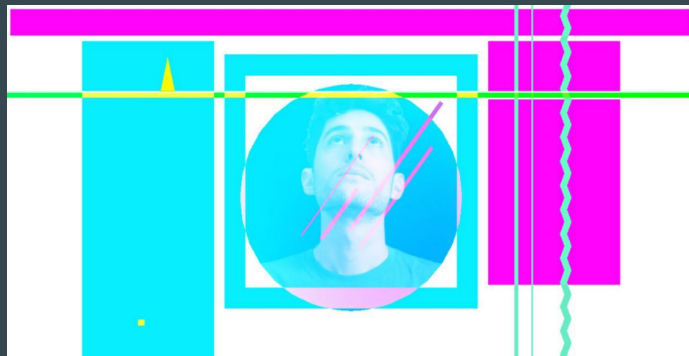
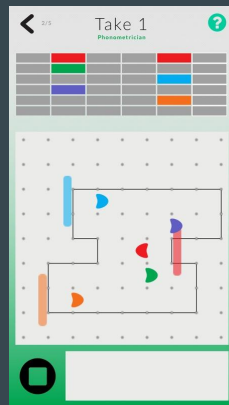
# Autor de la llibreria

- **Yotam Mann**
- Informàtic i músic
- Progrés de la **música** mitjançant la **tecnologia**
- Facilitar la creació d'Echo:

<http://tonenotone.com/ECHO/>

- Primer test a gran escala de la llibreria a jazz.computer:

<http://jazz.computer/>



# Instal·lació

- Utilitzant npm:
  - npm `install` tone
  - `import * as Tone from "tone";`
- Directament a un fitxer HTML:
  - `<script src="https://unpkg.com/tone"></script>`
- Recomanat: `await Tone.loaded();`
- <https://tonejs.github.io/>
- <https://github.com/Tonejs/Tone.js>

# Hello Tone

- Attack & Release:
  - `const synth = new Tone.Synth().toDestination();`
  - `synth.triggerAttackRelease("C4", "8n");`
- Attack then Release:
  - `const synth = new Tone.Synth().toDestination();`
  - `const now = Tone.now();`
  - `synth.triggerAttack("C4", now);`
  - `synth.triggerRelease(now + 1);`
- <https://nero-xii.github.io/ToneJS-Research-Work/test.html>

# Audio Repository

- Notes de piano
  - Mostres de bateria
  - Instruments MIDI
  - Acords de diversos instruments
  - Efectes sonors
  - Sorolls de fons
  - ...
- 
- <https://github.com/Tonejs/audio>





# Sampler Example

```
- const sampler = new Tone.Sampler({  
-   urls: {  
-       C4: "C4.mp3",  
-       "D#4": "Ds4.mp3",  
-       "F#4": "Fs4.mp3",  
-       A4: "A4.mp3",  
-   },  
-   release: 1,  
-   baseUrl: "https://tonejs.github.io/audio/salamander/",  
- }).toDestination();  
- Tone.loaded().then(() => {  
-   sampler.triggerAttackRelease(["Eb4", "G4", "Bb4"], 4);  
- });
```

# Effects Example

- `const player = new Tone.Player({`
  - `url: "https://tonejs.github.io/audio/loop/chords.mp3",`
  - `autostart: true,`
  - `});`
- Distortion:
  - `const distortion = new Tone.Distortion(0.2).toDestination();`
  - `player.connect(distortion);`
- Filter:
  - `const filter = new Tone.Filter(400, "lowpass").toDestination();`
  - `player.connect(filter);`
- FeedbackDelay:
  - `const feedbackDelay = new Tone.FeedbackDelay(0.125, 0.75).toDestination();`
  - `player.connect(feedbackDelay);`

## Signals Example

- Podem imaginar un Signal com un llapis que dibuixa una línia en un gràfic. Pots controlar com puja o baixa aquesta línia (freqüència, volum, etc.), la velocitat amb què ho fa (temps) i la seva forma (lineal, exponencial, etc.).
- Oscil·lador:
  - `const osc = new Tone.Oscillator().toDestination();`
  - `osc.frequency.value = "C4";`
  - `osc.frequency.rampTo("C2", 2);`
  - `osc.start().stop("+2");`

# MIDI Library

- Converteix fitxers midi a un format JSON interpretable per ToneJS
- Instal·lació:
  - `npm install @tonejs/midi`
  - `<script src="https://unpkg.com/@tonejs/midi"></script>`
- Ús bàsic:
  - `const midi = await Midi.fromUrl("path/to/midi.mid")`
  - `midi.tracks.forEach(track => {`
  - `track.notes.forEach(note => {`
  - `synth.triggerAttackRelease(note.name, note.duration, note.time + now, note.velocity)`
  - `});});`
- <https://github.com/Tonejs/Midi>

# Web Piano



- <https://nero-xii.github.io/ToneJS-Research-Work/>