

Working with sound samples loaded in memory

INTRODUCTION

For some applications it may be necessary to work with sound samples loaded in memory, and uncompressed:

- No streaming/decoding in real time means less CPU is used,
- With all samples in memory it's possible to play them in sync with very high accuracy,
- It's possible to make loops, add effects, change the playback rate, etc.
- And of course, if they are in memory and uncompressed, there is no wait time for them to start playing: that are ready to be used!

LET'S TRY SOME DEMOS!

These features will be useful for video games: you often need to have many sounds ready to be played, and by changing the playback rate or the effects, you will be able to make many different sounds, even with a limited number of different samples (for instance, an explosion played at different speed, with different effects).

Try [this example at JSBin](#), click on the different buttons. Only two very small sound samples are used in this example: [shot1.mp3](#) and [shot2.mp3](#). You can get many free sound samples like these at the [freesound.org](#) Web site.

The screenshot shows a web browser window with two tabs: "JS Bin - Collaborative Java..." and "JS Bin". The address bar shows "output.jsbin.com/gesetu". Below the address bar, there are several buttons: "Shot 1", "Shot 2", "Shot 1 repeated", "Shot 2 repeated", "Shot 1 repeated at random intervals", and "Shot 2 repeated, pitch and interval random". Below these buttons, there is a toolbar with icons for "Inspecteur", "Console", "Débogueur", "Éditeur de style", "Performances", "Réseau", and "Web Audio". The "Web Audio" tab is selected, showing a graph of audio nodes. The graph consists of seven "Gain" nodes, each connected to a "DynamicsCompressor" node, which are all connected to an "AudioDestination" node. A red arrow points to the "DynamicsCompressor" nodes. To the right of the graph, there is a red text box with the following text: "Try this example with FireFox, Open the Web Audio tab in the devtools, and look at the graph that is generated when you click on one of the buttons. Click once and look at what is happening: the nodes will be garbage collected after a while. Try clicking many times: the graphs becomes huge. Web Audio is optimized for handling thousands of nodes..."

Try this example with FireFox, Open the Web Audio tab in the devtools, and look at the graph that is generated when you click on one of the buttons. Click once and look at what is happening: the nodes will be garbage collected after a while. Try clicking many times: the graphs becomes huge. Web Audio is optimized for handling thousands of nodes...

Of course, music applications such as Digital Audio Workstations (sort of GarageBand-like apps) will need to play/record/loop music tracks in memory.

Try this impressive DAW that uses free sound samples from freesound.org. Its author calls it "Band in a browser" ([more info on the Web site](#))! Each instrument is a small audio file that contains all the notes played on a real instrument. When you play a song (midi file), the app will play in time the right note of the corresponding instrument audio sample. This is all done with Web Audio and samples loaded in memory:

Applications

Wiki2

Telecharger article

https://vegas.univ-tl

manifestR

manifestR

»

Autres favoris

remixxer.com/app/

Search Sound:

| | | |
|--------------------|------------------------------|-------------------|
| STICKS #1 | untit | Drum Stick #1 |
| Electric Guitar #7 | untitled (pr#27, ch#8) | Fiddle #7 |
| Bass Drum New #1 | untitled (pr#0, ch#9, no#35) | Reed Organ #4 |
| Gtr Cut Down #1 | untitled (pr#0, ch#9, no#49) | Brightness #1 |
| Rhodes #1 | untitled (pr#5, ch#6) | Dist Gt.bass #2 |
| Finger Bass #4 | untitled (pr#33, ch#1) | XR10TIML #1 |
| chh #1 | untitled (pr#0, ch#9, no#42) | Dist Gt.bass #5 |
| Finger Bass #4 | untitled (pr#33, ch#2) | Mute Gtr #2 |
| rsnare #1 | untitled (pr#0, ch#9, no#40) | Thunder #1 |
| Filter Snap #1 | untitled (pr#0, ch#9, no#39) | Door Slam #1 |
| Piano #15 | untitled (pr#0, ch#7) | Bandoneon #7 |
| ohh #1 | untitled (pr#0, ch#9, no#46) | Bird - Sparrow #1 |
| Alto Sax #1 | untitled (pr#65, ch#13) | Banjo #3 |
| Clavinet #8 | untitled (pr#7, ch#10) | Dulcimer #5 |
| Clavinet #8 | untitled (pr#7, ch#11) | Synth Voices #5 |
| Panta Fret bass #1 | untitled (pr#35, ch#12) | Reed Organ #3 |
| | | org:Cathedrl #17 |
| | | Metal Pad #2 |
| | | Timpani #1 |

FreeSound.org

SoundCloud.com

Play

REC

Load

120
Tempo

Quantize

1/4
Quantize

Metronome

Register

Piano

Full Screen

Log in

version: 0.7 (ogg)

tracks: 16/16

channels: 16

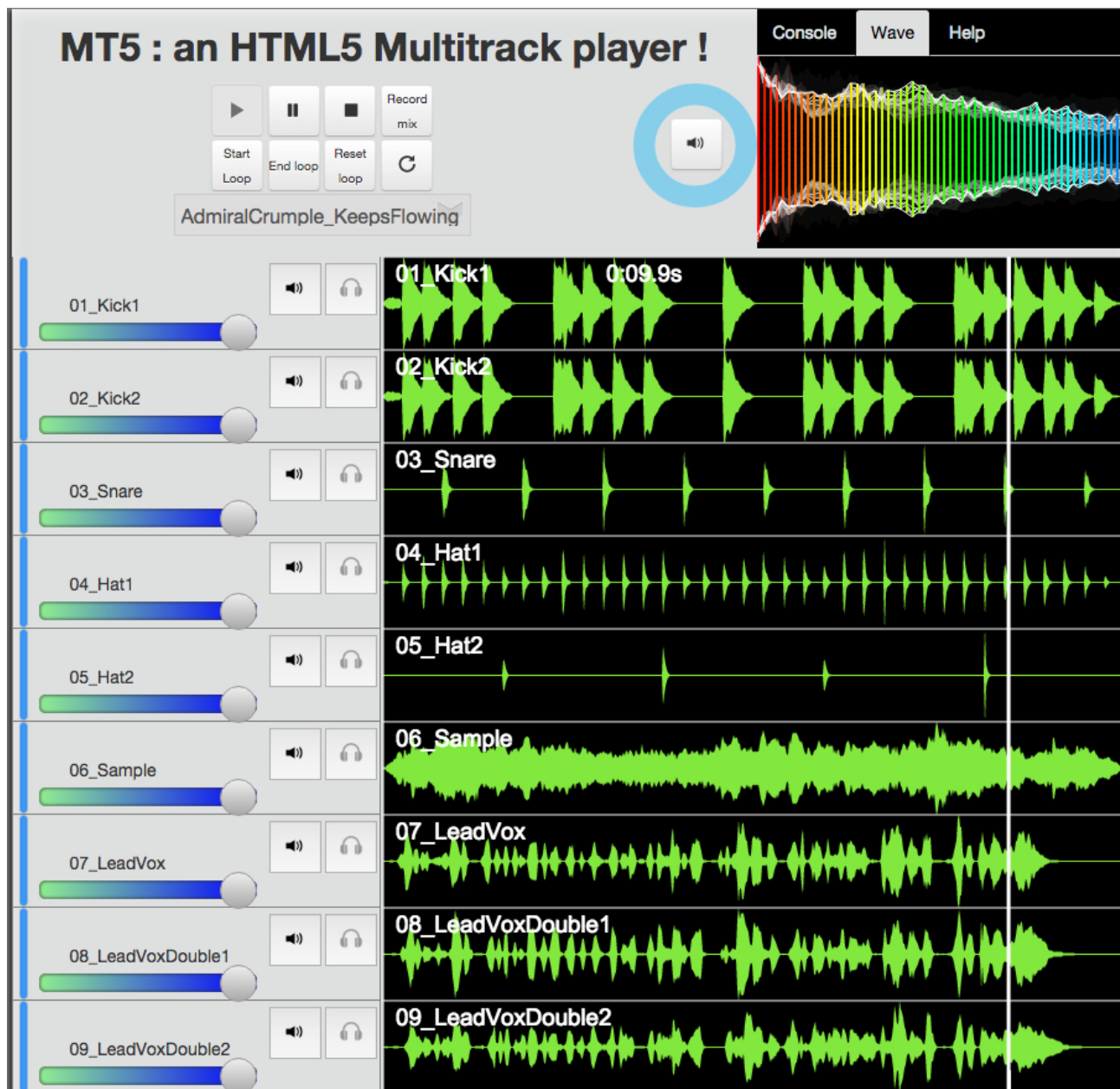
sounds: 14/14

used memory: 860.6 Kb

tsvetozar@gmail.com

The author of this course wrote a multitrack audio player: it loads different mp3 files corresponding to different instruments and play/loop them in sync.

You can try it or [get the sources on GitHub](#). The documentation is in the help menu.



Try also this small demonstration that uses the [Howler.js library](#) for loading sound samples in memory and playing them using WebAudio (we'll discuss this lib later). Click on the main window and notice how fast the sound effects are played. Click as fast as you can!

[Try the explosion demo at JSBin](#):

Particle based explosion system based on <http://www.gameplaypassion.com/blog/explosion-effect-html5-canvas/> by Michel Buffa... it uses the HowlerJS library for loading and playing sounds using WebAudio.

BOOM!

Basic Explosion

