Working with sound samples loaded in memory

INTRODUCTION

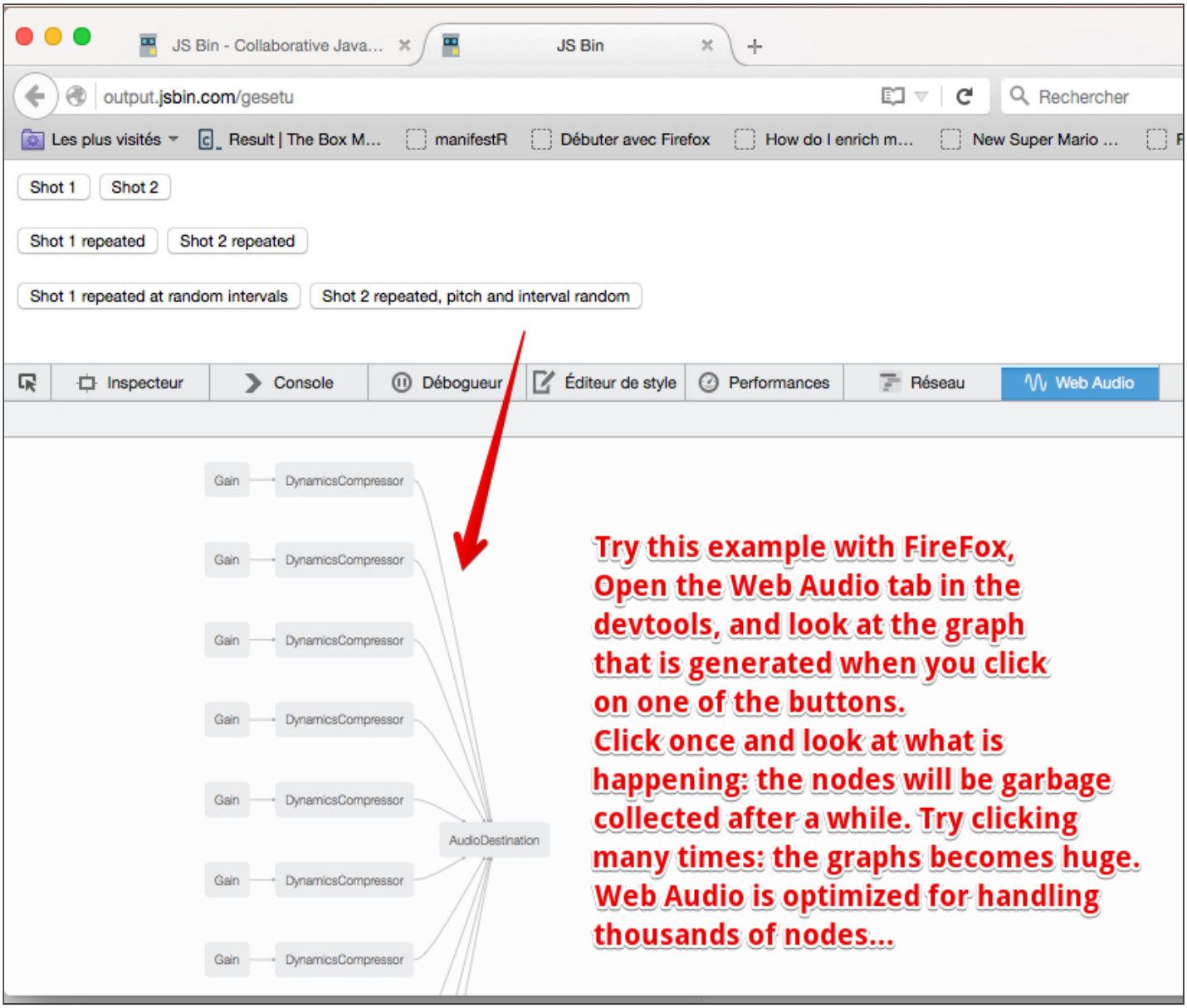
For some applications it may be necessary to work with sound samples loaded in memory, and uncompressed:

- 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.



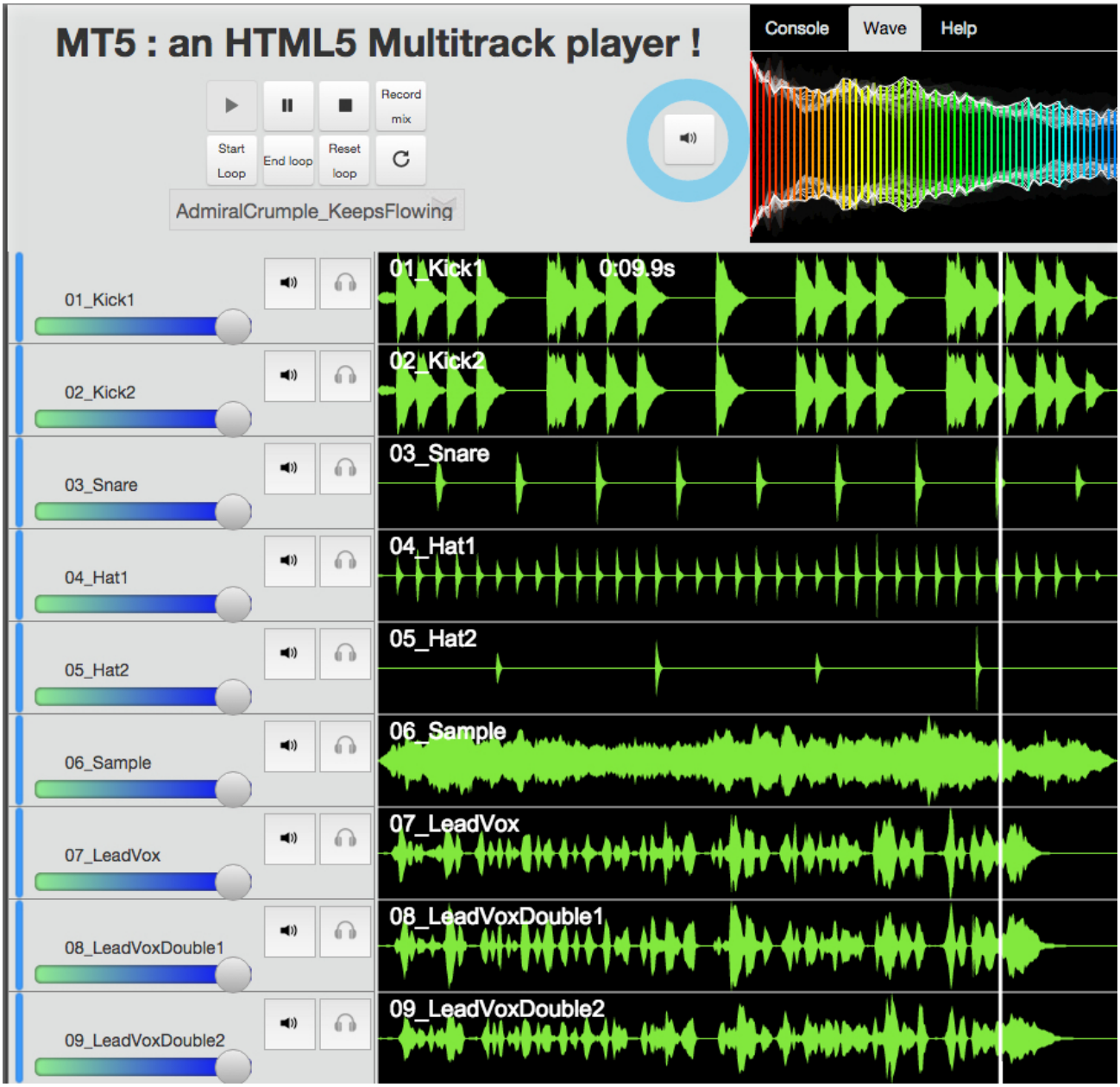
Of course, music applications such as Digital Audio Workstations (sort of GarageBand-like apps) will need to play/record/loop music tracks in memory.

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:


michel
remixxer.com/app/
Applications
Wiki2
Telecharger article
https://vegas.univ-tl
manifestR
manifestR
Autres favoris
Load
Save
Save
Imp
TO-I
How
Song
MID
Current Channel:
Sound:
STICKS #1
Lock Sound
Sound Properties
30
Volume
0
Octave
Mute Channel
Solo Mode
Filters
On
Track Properties
STICKS #1
untit
Electric Guitar #7
untitled (pr#27, ch#8)
Bass Drum New #1
untitled (pr#0, ch#9, no#35)
Gtr Cut Down #1
untitled (pr#0, ch#9, no#49)
Rhodes #1
untitled (pr#5, ch#6)
Finger Bass #4
untitled (pr#33, ch#1)
chh #1
untitled (pr#0, ch#9, no#42)
Finger Bass #4
untitled (pr#33, ch#2)
rsnare #1
untitled (pr#0, ch#9, no#40)
Filter Snap #1
untitled (pr#0, ch#9, no#39)
Piano #15
untitled (pr#0, ch#7)
ohh #1
untitled (pr#0, ch#9, no#46)
Alto Sax #1
untitled (pr#65, ch#13)
Clavinet #8
untitled (pr#7, ch#10)
Clavinet #8
untitled (pr#7, ch#11)
Fanta Fret bass #1
untitled (pr#35, ch#12)
Search Sound:
Drum Stick #1
Fiddle #7
Reed Organ #4
Brightness #1
Dist Gt.bass #2
XR10TIML #1
Dist Gt.bass #5
Mute Gtr #2
Thunder #1
Door Slam #1
Bandoneon #7
Bird - Sparrow #1
Banjo #3
Dulcimer #5
Synth Voices #5
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 explositon 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