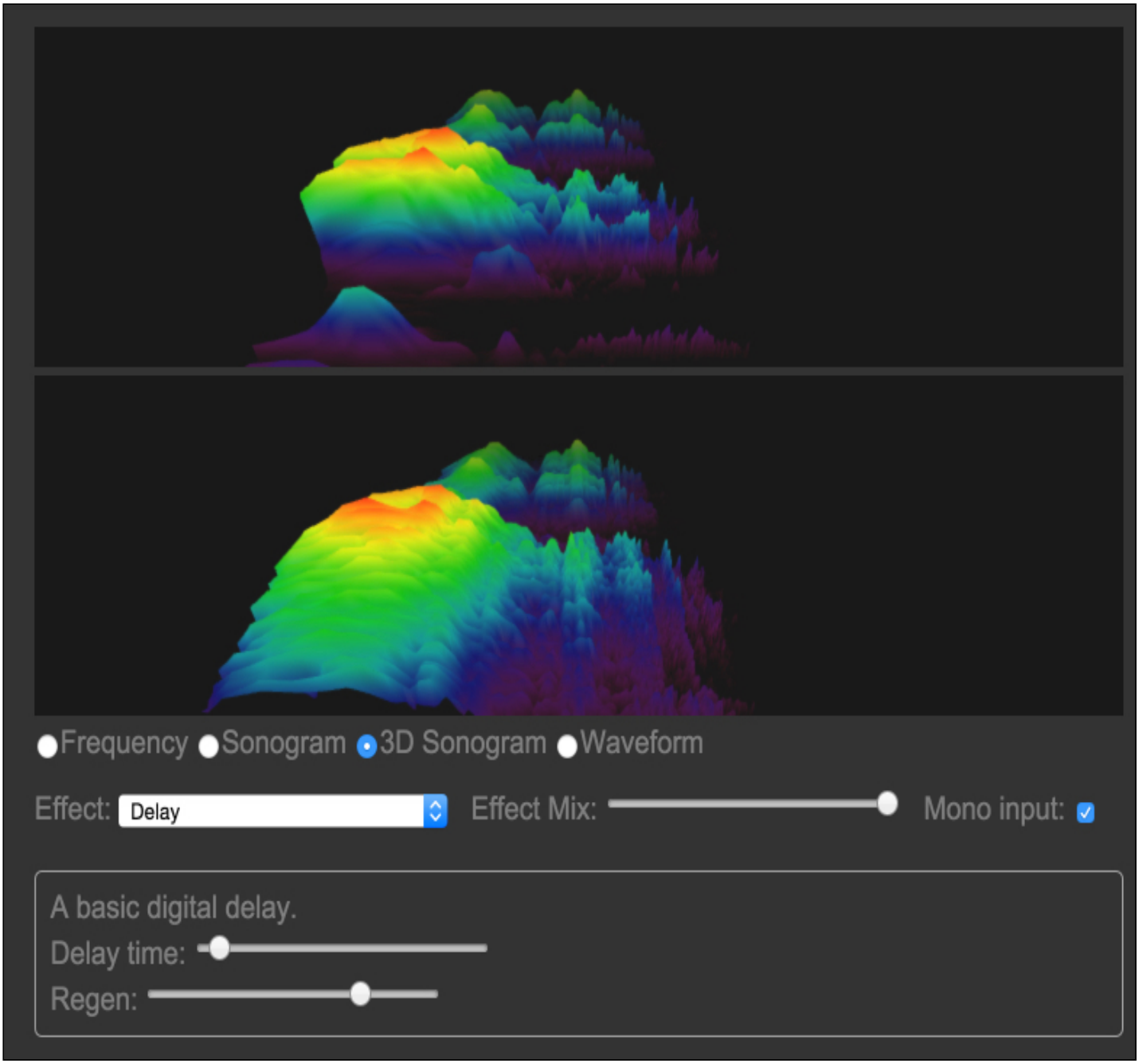
Working with the microphone

Instead of using the getUserMedia API with: `navigator.getUserMedia({video:true}, onSuccess, onError)`, it is also possible to use `{audio:true}` for the first parameter. In that case, only the microphone input will be captured. Notice that `{video:true, audio:true}` is also accepted, if you write a video conferencing system and need to capture both the audio and the video (this is often the case when writing WebRTC applications). The W3C WebRTC is another W3C specification, under development, for P2P audio/video/data Real Time Communication.

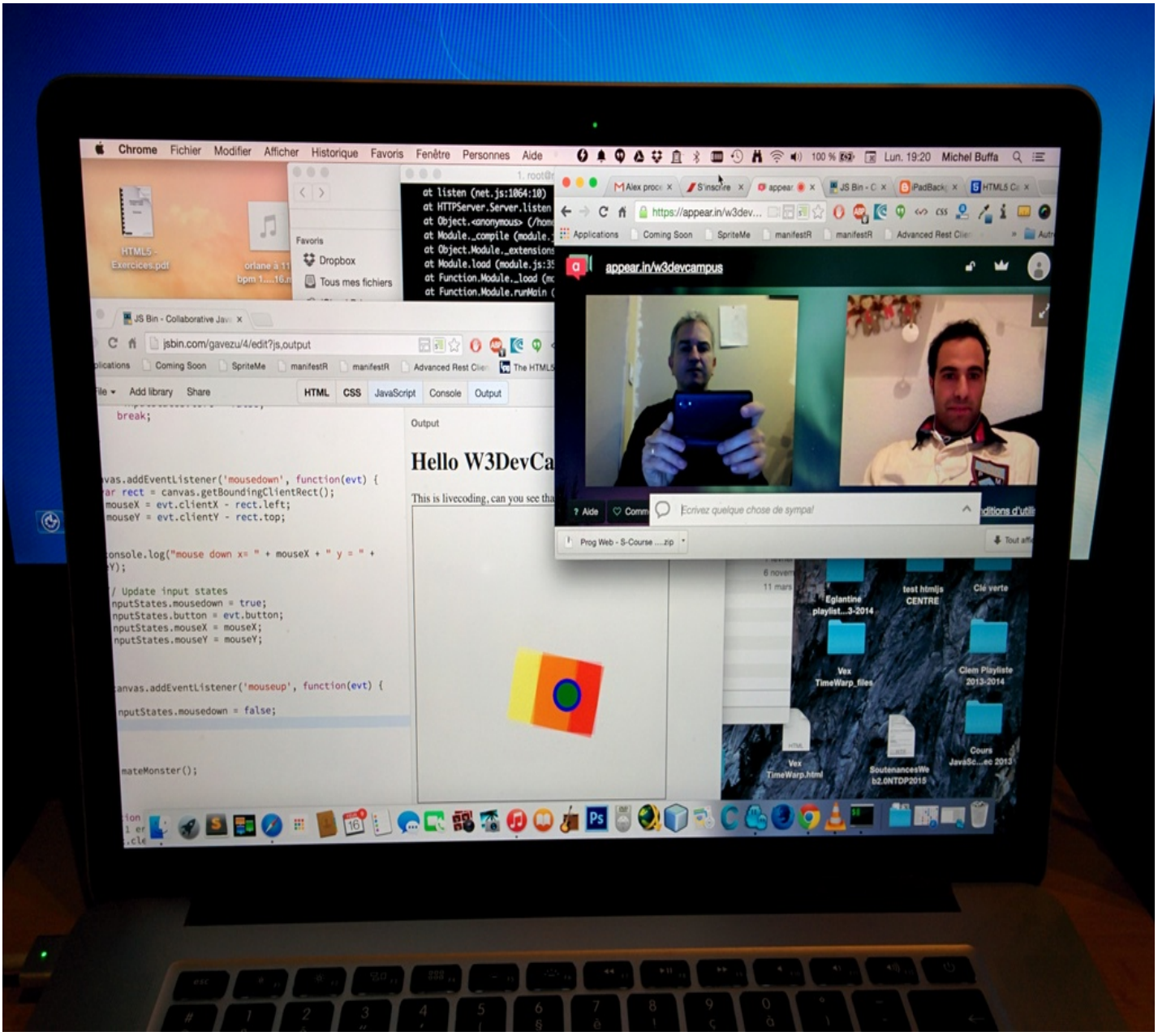
Apart from videoconferencing, microphone input will be used for music Web apps, from the WebAudio API. This API focuses on real time sound processing and music synthesis. This API will be covered in the advanced W3Cx HTML5 course.

Do try some nice WebRTC applications like Appear.in audio and video conferencing tool. Also check out the WebAudio demonstrations written by Chris Wilson, in particular the one called "Input effects".

Below is an example of real time audio processing of the microphone input using getUserMedia and WebAudio APIs



appear.in: a free WebRTC video conferencing tool. It uses the getUserMedia API for video and audio.


Chrome Fichier Modifier Afficher Historique Favoris Fenêtre Personnes Aide
Lun. 19:20 Michel Buffa
HTML5 - Exercices.pdf
Favoris
Dropbox
Tous mes fichiers
at listen (net.js:1064:10)
at HTTPServer.Server.listen
at Object.<anonymous> (/hom
at Module._compile (module.
at Object.Module._extensions
at Module.load (module.js:3
at Function.Module._load (mo
at Function.Module.runMain (
https://appear.in/w3dev...
Applications
Coming Soon
SpriteMe
manifestR
manifestR
Advanced Rest Clie
appear.in/w3devcampus
JS Bin - Collaborative Jav
jsbin.com/gavezu/4/edit?js,output
Add library
Share
HTML
CSS
JavaScript
Console
Output
break;
vas.addEventListener('mousedown', function(evt) {
ar rect = canvas.getBoundingClientRect();
mouseX = evt.clientX - rect.left;
mouseY = evt.clientY - rect.top;
onsole.log("mouse down x= " + mouseX + " y = " +
Y);
/ Update input states
nputStates.mousedown = true;
nputStates.button = evt.button;
nputStates.mouseX = mouseX;
nputStates.mouseY = mouseY;
anvas.addEventListener('mouseup', function(evt) {
nputStates.mousedown = false;
mateMonster();
Output
Hello W3DevCa
This is livecoding, can you see tho
Aide
Écrivez quelque chose de sympa!
Prog Web - S-Course ....zip
Eglantine playlist...3-2014
test htmljs CENTRE
Clé verte
Vex TimeWarp_files
Clem Playliste 2013-2014
Vex TimeWarp.html
SoutenancesWeb2.0NTDP2015
Cours JavaSc...ec 2013