HTML5 tutorial

Michel Buffa
Wimmics Group, I3S laboratory/INRIA,
University of Nice, France.
buffa@i3s.unice.fr
+33492965103
buffa@i3s.unice.fr

ABSTRACT

This HTML5 tutorial is available at: http://tinyurl.com/77z3t7j

Note about the author: Michel Buffa teaches web technologies at the University of Nice, France, since 1994. He started a HTML5 course last year for master students in computer science[1]. Michel Buffa belongs to the Wimmics research group from I3S/CNRS and INRIA. He is co-author of the semantic wiki SweetWiki and co-supervised recently two PhDs on semantic social network analysis and on semantic enrichment of folksonomies. He also supervises students that use HTML5 at the heart of the software they develop during their PhD, including a JavaScript semantic wiki that uses web sockets and a HTML5 WYSIWYG editor.

The tutorial consists in a presentation (HTML5 slides that include many interactive applications) and lab exercises. Both are downloadable from the tutorial web page. The presentation illustrates the main characteristics of HTML5 and focus on some APIs that have been recently implemented by browsers, such as the new getUserMedia API for real time audio and video streaming. The HTML5 slides cover the following HTML5 tags and APIs: forms, geolocation, File and drag'n'drop APIs, canvas, audio and video tags, getUserMedia API for real time streaming, the sound audio API for real time sound processing and sound synthesis, and the web sockets API for bidirectional, real time communication. Each topic is illustrated by several demonstrations (tiny applications embedded in the slides, with interactive features), most of them developed especially for this tutorial by the author and his students from the University of Nice. All examples came with source code, either available online in the JSbin.com IDE (Figure 1) or as downloadable archives. They include many small code snippets sorted by topic, as well as full featured applications such as a multiplayer arcade game that uses physic simulation or a paint program involving several participants in real time as well as the getUserMedia API for painting with the images coming from the web cam video stream (Figure 2).

The labs exercises are divided in two paths:

1. A "discovery path" composed of around fifty small examples sorted by category (forms, drag'n'drop, canvas, video, web sockets, etc.) Nearly each example is hosted by the JSbin.com online IDE and can be tweaked in real time, facilitating the understanding of the code.
2. A "geek path" that proposes to write step by step a paint program for several participants at the same time, using web sockets. This path involves the installation of the NodeJS web server as well as the installation of some additional modules for NodeJS, for enabling web sockets or JavaScript distributed objects over web sockets. Step 1 starts with a simple paint program that draws when the mouse moves, Step 2 adds the management of more mouse events, Step 3 shows an interesting use of canvases: they can be layered one on top of another. Using relative positioning and the CSS z-index property with different values, they can act as layers in Photoshop. By default canvases are transparent so we can see through them. This step shows how to draw "elastic lines". Step 4, 5 and 6 add different functionalities like drawing in color, drawing different shapes, etc. while Steps 7 and 8 introduce the use of web sockets in order to add collaborative features to the application. Step 7 shows how to write a small chat application using NodeJS and web sockets and include it in the paint program. Step 8 extends the chat in order to transfer paint events from one client to the web socket server, which in turn broadcasts them to other clients. The getUserMedia API is also used in that step so that one can paint with the live picture coming from the web cam (in sync with other participants).

[1] See the HTML5 part of this page http://miageprojet2.unice.fr/Intranet_de_Michel_Buffa/Option_web_2.0_Master_1_informatique_2011

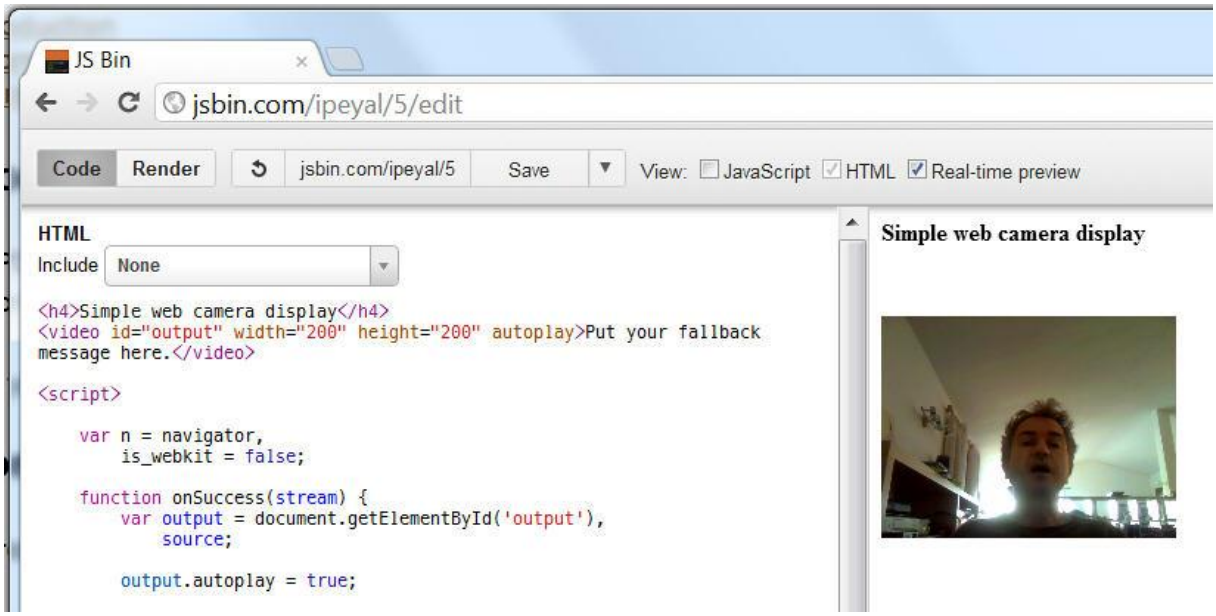

Figure 1: a small example in the JSbin.com online IDE that shows how to use the HTML5 getUserMedia API for real time video streaming. Users can tweak the code and see in real time the results.

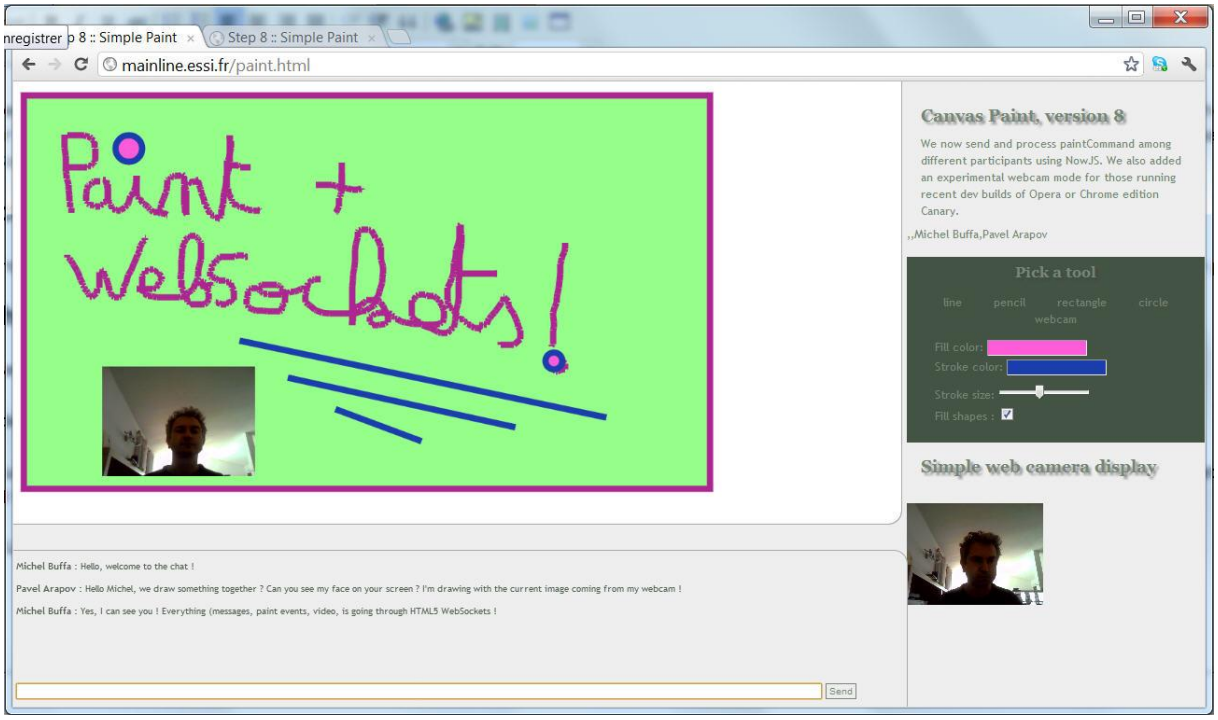

Figure 2: a multi participant paint program that uses web sockets and the getUsermedia API for real time video streaming from the webcam. It includes a chat (bottom) and uses the NodeJS server as well as some additional modules for sharing objects between client code and server code.