Utilization and Standardization of Web Audio

There is so much content on the internet today it's inconceivable. The hardware used in a computer system allows a wide array of interaction with this content. Eyes survey and process information on a screen. The tactile sense controls the hands that press keys, move the mouse, tap a touchscreen, or apply pressure to a stylus. However, there is one extremely important way of interacting with a computer that is often overlooked. Using the auditory sense to perceive sounds that are emitted from a speaker adds an entirely different dimension to interfacing with technology. There are innumerable applications for audio when considering the utility of the internet. The web is a massive communal space, and stored on it are countless files with sonic information. Having web standards for audio is pivotal for the data to be easy to use, change, move, or even create.

The W3C Audio Working Group was charted in 2011. Its official mission statement is to "support the features required by advanced interactive applications including the ability to process and synthesize audio" (Web Audio Processing). They realized that it was crucial to have the open web be an easy and efficient platform for handling audio. Adding high quality, low capacity sound to applications, games, videos, and music, will make the future of the web a better place. Developing specifications which define a client-side script API will add advanced capabilities. The goal was to make it easy for the public to use digital audio on the internet.

In 2012 the Web Audio API was drafted. This specification "describes a high-level JavaScript API for processing and synthesizing audio in web applications" (Web Audio API). It was also made to be easily used in conjunction with other APIs and elements on the web. For example, to start music on a pageload it can be combined with XMLHttpRequest. For gaming, it would be linked with a 2D Canvas or 3D WebGL object. The Web Audio API was intentionally

made versatile to be able to adapt. This way it would be adopted by the most programmers and continue the standardization effort. W3C wanted it to be powerful enough to provide “some of the mixing, processing, and filtering tasks that are found in modern desktop audio production applications” (API). One of the most important of these features is the modular routing capabilities. It allows for connections between the AudioNode objects. Each node can have inputs, outputs, or both. All routing is done in an AudioContext container. A source and destination are specified, and the data is retrieved. Other nodes can be added to allow for volume manipulation, overlaying filters, crossfading, or even setting the time signature. These are all the building blocks of loading and utilizing audio on a webpage. It will minimize dependence on plugins and embedded sound. It equips developers with a standard set of tools that will help keep audio playback simple and error-free. Another advantageous feature is the ability to process spatialized audio. Maybe back in the day of 8-bit, side-scrolling games just needed music and a couple of sound effects. Today, the web has thousands of interactive, three-dimensional games that take place in a virtual environment. With present day stereo audio, there is a right and left speaker. The Web Audio API adjusts balance and volume to give the illusion of a three-dimensional soundstage while playing a video game. Using a complex panning algorithm, it receives spatial data from the game and produces sounds depending on if the source is to the right or left. It can tell the distance from the source of the sound, and manipulate volume accordingly. It can even use a Doppler shift to “realistically simulate moving sources. This depends on source and listener velocity vectors, speed of sound, and Doppler factor” (Web API). Also, certain sound snippets are triggered when the player performs a certain action such as firing a gun, starting a car, or walking with loud footsteps. Game developers wouldn’t put so

much effort into stellar sound design if it was difficult to implement on the internet. Having standards for using stereo sound in video games helps make the web a more viable medium.

HTML5 is becoming more prevalent every day. Although it was finalized only a year ago in October 2014, it is becoming the main specification on the web. It expands on the older version in some key areas to provide a vastly expanded functionality. These improvements only further the pursuit of audio standardization on the internet. For example, the HTML5 specification introduced the <audio> and <video> tags. A MediaElementAudioSourceNode is created using that AudioContext container, and houses the content. This made it so much easier for a beginning coder to insert small sound or video clips into their site. Before this plugins like Flash or ActiveX had to be used, and there was no standard way of playing video and audio in the webpage. Now there is a convenient media element built into HTML5 with consistent filetypes. These tags support .mp4, .webM, .ogg, and .wav audio types, and have been optimized to work with most browsers. This is paramount because these are the most common formats that can be played with any universal media player. Some filetypes cause complications. FLAC, lossless sound, can't be played in Apple iTunes. If FLAC was allowed in the <audio> tag, someone could accidentally download it and put it on their iOS device. Then, when the user goes to listen to that song it would cause an error. This type of dysfunctionality is what standards strive to eliminate. By normalizing and regulating which types of audio can be on the web, it makes using ears to interact with computers that much easier. In addition, these new audio and video elements come with a plethora of methods and properties which allow more manipulation. Code can be added to present control buttons, autoplay on page load, loop the content, modify playback speed, and many more inventive functions. Little tweaks to existing functionality like this will make development friendlier and more accessible.

One of the most futuristic applications of audio is video chatting. Audiences were captivated in *Back to the Future II* when Marty McFly is fired by his boss on a large conference screen in his living room. Humans in 1989 imagined a world where a communal internet could be used to transmit synchronized video and dialogue over long distance. Today, this technology is available on not only computer, but phones also. With so many different devices, there has to be a standard way of audio and video through a transmission medium. The HTML Media Capture specification extends the HTML form and gives access to the devices media capture mechanism. Microphone and camera data are given to the HTMLInputElement with a capture attribute. It was designed to be simple and declarative. Keeping streams organized allows for all members of a chat to control incoming/outgoing audio and video. All a user must do is click a mute button to halt the audio stream being passes to the capture attribute. Having the implementation be this elastic has allowed for an explosion in this type of software. Skype, Oovoo, Google Hangouts, and Chatroullete are all successful applications that have applied this mechanism. Since the telephone, sound has been a vital part of the way society communicates from a distance. Being able to associate facial movements with a conversation adds a whole new dimension to electronic transmission. Having web standards that govern how video chatting operates is only beneficial for its effectiveness.

It also reinvents the world of music and audio engineering, forming a global effort towards innovation. Not long ago, digital sound had not existed. Everything was analog and stored on physical mediums. In the modern age, the internet has changed the way we share, listen to, and use audio. Spotify, Last.fm, Apple Music, Pandora, and Soundcloud are all huge conglomerates of music hosted completely on the internet. Audio is becoming a very social form of media. Who is driving this aural renaissance? It’s not a reclusive, elite, upper-class of web

geniuses. Making audio on the internet accessible has been a foremost objective in all the progress thus far. Look in the mirror to see who can be a potential primary mover for this normalization movement. Give a man a plugin, he'll listen for a pageload. Teach a man to use audio on the web, and he'll develop for a lifetime. W3C understood the moral of that age-old proverb, and made the future of internet sound flexible. Having a comprehensive API gives anyone the reference they need tailor functionality to their needs. MP3 isn't going anywhere soon either. In other industries and forms of media, filetypes and compatibility is a mess. Having just a handful of widely accepted filetypes for sound extends the current era of development. As always, the general public will embrace HTML5 audio and push it to its fullest capabilities. Music streaming services, video chat, web-based games, voice recognition, electronic music production platforms, and so many more applications rely on sound to interact with the user. Using our sense of hearing to receive and process audio will only become more customary as technology advances. It will be essential to keep its implementation standardized and normalized to achieve an accessible web.

Works Cited

"HTML Audio/Video DOM Reference." W3C, 28 Oct. 2014. Web. 13 Oct. 2015. <http://www.w3schools.com/tags/ref_av_dom.asp>.

"HTML Media Capture." *HTML Media Capture*. Ed. Anssi Kostiainen. 9 Sept. 2014. Web. 13 Oct. 2015. <http://www.w3.org/TR/html-media-capture/>.

"Web Audio API." *Web Audio API*. Ed. Chris Rogers. 2 Aug. 2012. Web. 10 Oct. 2015. <http://www.w3.org/TR/webaudio/>.

"Web Audio Processing: Use Cases and Requirements." Ed. Joe Berkovitz. W3C, 29 Jan. 2013. Web. 10 Oct. 2015. <http://www.w3.org/TR/webaudio-usecases/>.