A Web-Based Approach to Virtual Guitar Amplification

Richard Marquez
Computer Science Department
Winona State University
Winona, MN 55987
RMarquez14@winona.edu

***Abstract*— Guitar amplifiers take the electrical signal from an instrument and strengthen it to be pushed out through a speaker system. They are often prohibitively expensive or overly complex. A new system has been developed to make virtual guitar amplification available through the web. This allows users to be able to access virtual guitar amplification more quickly and easily than through a traditional digital audio workstation.**

Keywords—audio; signal processing; web; amplifier; guitar

I. INTRODUCTION

A. Background Information

An electric guitar uses magnetic "pickups" to convert the mechanical vibrations of its ferrous strings into an electrical signal. This signal is too weak to drive a speaker by itself so it is first sent through an amplifier. A guitar amplifier amplifies the signal so that it can produce sound through one or more speakers. An amplifier also provides a set of basic sliding controls including: volume—the loudness of the output; gain—the sensitivity of the input; tone—the amount of treble present; and reverb—simulation of a reverberant environment.

Physical guitar amplifiers are often expensive and physically unwieldy which can be prohibitive traits for hobbyist players. Increases in computer performance have led to the proliferation of software-based virtual guitar amplification. However, virtual guitar amplifiers can also be expensive and overly complex, often requiring special training as a part of a full-scale digital audio workstation (DAW).

The chart in Fig. 1 shows the popularity of various digital audio workstations as voted on by readers of Ask.Audio, a resource for educational materials by digital music makers. GarageBand and its associated paid version, Logic Pro, together make up some of the most popular DAWs [1] due to their relative simplicity and availability on macOS. The relatively low bar to entry of these DAWs is still prohibitive to many players who do not have access to an Apple computer, have no interest in recording software, or do not have the time to download, install, and learn how to operate the software.

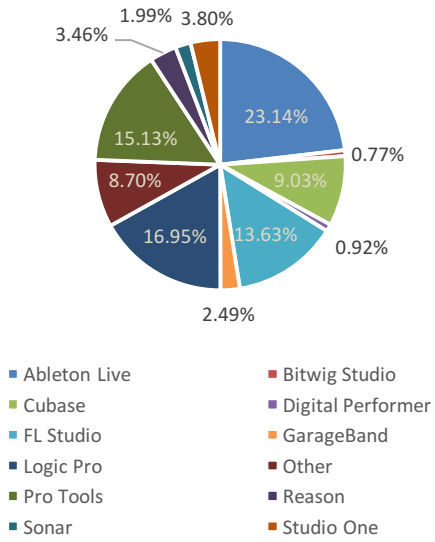

Figure. 1. Popularity of digital audio workstations

B. Research Purpose

When I began learning to play guitar I was in the situation of needing an amplifier. I did not want to make the financial commitment of purchasing a physical amp. The alternative was to use a virtual amp, with GarageBand seeming to be the best available choice given my circumstances: I had a Mac computer and was searching for a free option. I was in no need of the slew of professional recording features, as shown in Fig. 2, that dominate the application. As such, the learning curve was unnecessarily steep and nearly destroyed the effort altogether.

That personal story is a demonstration of the need for a simpler and accessible virtual guitar amplifier. The goal of this project is to build such a software that makes straightforward guitar amplification available in the browser space.

A web-based guitar amplifier makes virtual guitar amplification available through the web browser. Virtual guitar amplification will be able to be achieved more quickly and easily through a web application dedicated solely to guitar amplification, rather than a traditional full-featured desktop DAW.

Proceedings of the 17th Winona Computer Science Undergraduate Research Seminar, April 26, 2017, Winona, MN, US.

Figure. 2. GarageBand example project

II. METHODOLOGY

A. Development Strategy

This is a software development project that used industry standard web technologies to develop a single-page web application that provides standard guitar amplifier functionality given an audio input. Functionality includes amplification of the input signal as well as controls for volume, gain, tone, and reverb. Fig. 3 shows the final user interface. It features adjustable circular controls similar to physical amplifiers for the preceding effects. It also includes a waveform display of the output signal.

HTML5 and CSS3 form the base of technology used for front-end development. The Bootstrap framework was used to provide page structure and responsive capabilities. The audio waveform was drawn using the standard HTML5 canvas element. The JavaScript language and associated web APIs were used to develop the back-end of the application. A combination of jQuery and standard JavaScript were used to connect the data fed from the back-end into the front-end, and vice versa. The standard Web Audio API was used to interface with audio input and output [2].

JetBrains WebStorm was used as the primary IDE for development with an educational license. Git was used for version control, and the source of the project is available publicly on GitHub. The web application was tested on the Safari, Chrome, and Firefox browsers on macOS and Windows systems to ensure cross-compatibility.

An iterative approach to development was taken. Amplification and output of the input signal form the core of the application. Once the core was complete the effects controls for volume, gain, tone, and reverb were added. The waveform display was developed after all of the preceding functionality and serves primarily as a UI element.

B. Software Internals

Connecting the guitar to the computer with the 1/4 inch to USB cable makes it appear as a standard audio input device. The browser's audioContext object is used access the input and output streams of the chosen audio devices (the guitar and system speakers, respectively). Audio nodes are created and attached to the audioContext's graph to process the signal.

1) Tone

Tone indicates the amount of treble present in the signal. A biquad filter is used to achieve this effect. A highshelf biquad filter node is created and connected to the input audio device's stream (the beginning of the signal chain) to modify the range of frequencies that are attenuated [3]. Selecting a high tone

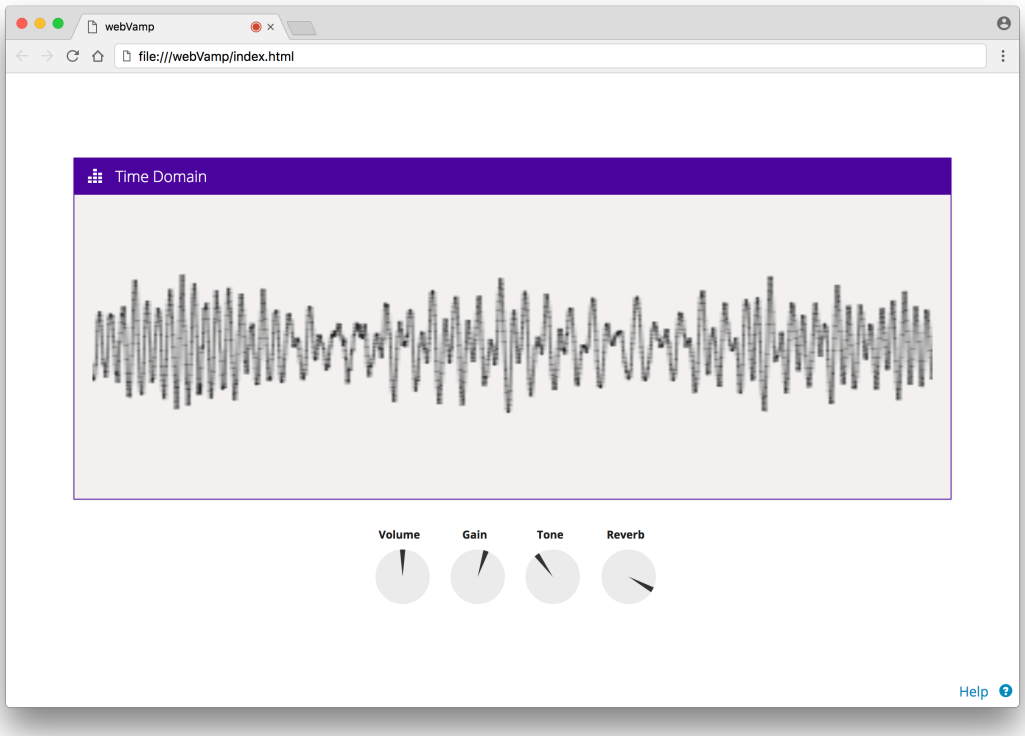

Figure. 3. Finished UI of web-based amp

value from the UI knob will raise the threshold of attenuated frequencies; i.e. the higher the tone, the more treble.

2) Reverb

The reverb effect controls simulation of a reverberant environment. This typically involves complex calculations or an intricate set of interconnecting audio nodes. To simplify this process, the soundbank-reverb library (developed by mmckegg on GitHub) was used. A reverb node from that library is created and connected to the tone node. The UI's reverb knob adjusts the time property of the reverb node as it is spun; i.e. the higher the reverb, the longer it lasts.

3) Amplification and Volume

The amplification and volume of the signal can be considered one and the same from a programmatic point of view. A gain node is created that takes the reverb node as its input. The reverb node is used instead of the direct input stream so that the volumes of the effects are also taken into account. The gain property of the gain node used to represent the volume of the entire signal chain. The UI's volume knob changes the value of the gain property as it is spun.

4) Gain

The gain control modifies the sensitivity of the input. To achieve this effect, another gain node was created that resides slightly askew to rest of the graph. It is connected to both the reverb node and the input stream. The UI's gain knob changes the value of the node's gain property as it is spun; i.e. the more gain, the more sensitive the input.

5) Waveform

The audio waveform is displayed in the time domain. The amplitude of the signal is represented on the y-axis and time on the x-axis. Because of the relatively small amount of signal processing taking place within the application there were no performance issues resulting in signal buffer lag. A script processor is attached to the input stream to redraw the graph whenever audio is processed. It is drawn on an HTML5 canvas element.

C. Testing

1) Test Process

A survey was executed to assess the hypothesis. The user was tasked with achieving a usable tone from either GarageBand or the new system given an electric guitar. The user may have had experience with the instrument, but may not have had experience with any digital audio workstation software.

The user was provided with an electric guitar, 1/4-inch audio to USB cable, and a MacBook Pro (Early 2015) with GarageBand 10.1.6 and the web amp system installed. The user was asked to obtain a usable tone from either GarageBand or the new system. The user was shown links on the desktop to the appropriate software. No further instruction was given throughout the survey.

It should be noted that the time taken for GarageBand to be downloaded and installed (an unnecessary step for the web amp) was not part of the test. GarageBand 10.1.6 is a 956 MB package which takes approximately 19 minutes to download on a 7 Mbit/s connection [4].

2) Usability Survey

The time it took to acquire a usable tone was recorded. The user was then asked to rate the ease of use of the software on a scale from 1 (being extremely difficult) to 5 (extremely easy). The user was also asked to rate how likely they were to recommend the software to a beginner guitar player on a scale from 1 (extremely unlikely) to 5 (extremely likely).

These metrics were initially recorded on paper by the user and were then transferred into a Microsoft Excel spreadsheet. As suggested by the Nielsen Norman Group for user experience research [5] there was a total of twenty users in an effort to achieve statistically significant results. Half of the pool tested GarageBand, and the other half tested the new system.

III. Results and Analysis

There were a total of twenty users used to gather information about the software usability. Ten of the users tested GarageBand, and ten users tested the new web-based system. No users had prior experience playing guitar or using any type of digital audio workstation.

A. Time to Amplification

The overall mean time it took to achieve amplification, as shown in Fig. 4, was 1.3 minutes for the web amp and 5.1 minutes for GarageBand. As noted earlier, the approximately 19 minutes it takes to download and install GarageBand (an unnecessary step for the web amp) was not part of the test. Even without that added time a t-test showed a statistically significant difference between the results with a p-value < 0.05.

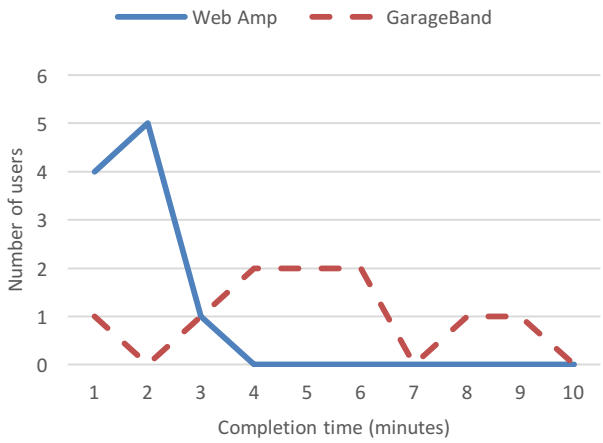

Figure 4. Time to completion results

B. Ease of Use

The overall mean rating for the ease of use of the software, as shown in Fig. 5, was a score of 4.6/5 for the web amp and 2.8/5 for GarageBand. A t-test showed a statistically significant difference between the results with a p-value < 0.05. to 5 (extremely likely).

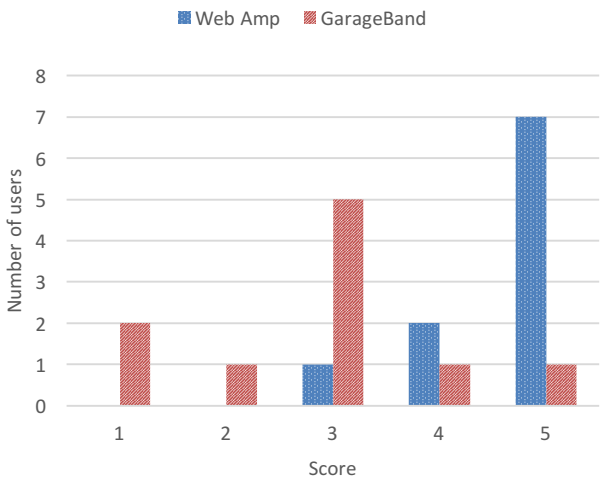

Figure 5. Ease of use results

C. *Likelihood of Recommendation*

As shown in Fig. 6, the overall mean rating for the likelihood the user would recommend the software to a beginner guitar player was highly skewed in favor of the web amp, with a score of 4.8/5, as opposed to GarageBand with a score of 2.8/5. A t-test showed a statistically significant difference between the results with a p-value < 0.05.

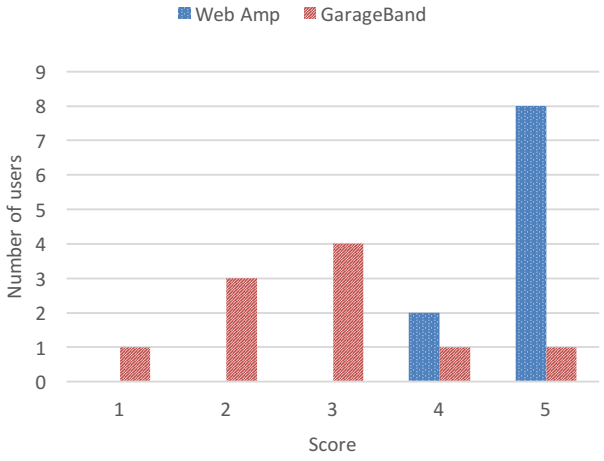

Figure 6. Likelihood of recommendation results

IV. Conclusion

The web amp software was developed as planned and used to compare usability with GarageBand as a traditional digital audio workstation. The statistically significant data acquired through the survey supports the initial hypothesis that a web-based amp with dedicated functionality can be used to achieve virtual guitar amplification more quickly and easily than through a traditional DAW.

The general accessibility of the functionality is also vastly increased by the web amp. GarageBand is available only on macOS systems and requires a large download. The web amp requires no download and is available on any system with a web browser that supports the Web Audio API (e.g. Chrome 53.0, Firefox 36).

V. Further Work

This project is open-source and available on GitHub (https://github.com/richard92m/webVamp). The next step is to have the application hosted online and made available for public usage.

The web amp should work on mobile devices given the appropriate hardware connections and a browser that supports the Web Audio API. This was not tested during the course of this research project.

This web-based virtual guitar amplifier stands as a nice proof of concept and exploration of the Web Audio API; however, there is a wide open space in the market for a full-fledged DAW in the browser. Soundtrap (http://soundtrap.com) is an example of the type of applications that will be coming in the near future. This open-source project may be expanded upon.

Acknowledgment

I would like to thank Dr. Narayan Debnath, Dr. Sudharsan Iyengar, and Dr. Mingrui Zhang for advising and reviews throughout this project. I would also like to thank the Winona State University Computer Science Department for this opportunity.

References

[1] R. Sethi, "The top 11 most popular DAWs," in *Ask.Audio*, 2015. [Online]. Available: https://ask.audio/articles/the-top-11-most-popular-daws-you-voted-for. Accessed: Feb. 7, 2017.

[2] I. Hickson, "Media Capture and Streams," in *W3C*, 2016. [Online]. Available: https://www.w3.org/TR/mediacapture-streams/. Accessed: Feb. 7, 2017.

[3] B. Smus, *Web audio API*. Sudbury, MA, United States: O'Reilly Media, Inc, USA, 2013.

[4] Akamai, "Q4 2016 State of the Internet/Connectivity Executive Summary," in *Akamai*, 2016. [Online]. Available: https://www.akamai.com/us/en/our-thinking/state-of-the-internet-report/index.jsp. Accessed: Apr. 1, 2017.

[5] J. Nielsen, "How many test users in a usability study?" in *Nielsen Norman Group*, 2012. [Online]. Available: https://www.nngroup.com/articles/how-many-test-users/. Accessed: Feb. 10, 2017.

[6] R. G. Lyons, *Understanding digital signal processing*, 3rd ed. Boston, MA, United States: Prentice Hall, 2010.