Daniel Dias
Ashana Evans
Principles of UI Software

Milestone 2: Project Implementation

OVERVIEW

Music is a very important part of our lives. People enjoy listening to the music, and many of us find a special pleasure in creating the music. Computers further extended many aspects of our musical experience[1]. The audio signal flow is something you can only hear, but what if you wanted to interact with the audio, actually manipulate the audio? Well, Digital Audio Workstation (DAW) is a software that has a computer, audio interface, digital audio editor software, and some input that needs to be modified. Below are the Pros and Cons of this current solution.

Current Solutions:

Pros:

1. Provides the ability to record, edit, and playback music
2. Offered across almost every operating systems

Cons:

1. Typical Scrollbars and button UI
2. Complicated for novice users
3. Limited features depending on which you use
4. Feedback is lagged when trying to manipulating music
5. Majority are Desktop-based

Noticing the shortcomings of the DAW our team proposes to develop a Native Web Application that allows for the user to interact with their audio through a 3-Dimensional User Interface and multi-touch gestures. We wanted to design and implement a new interaction techniques for accomplishing some interactive task, such as audio editing, that's not well supported by the current "standard" techniques (i.e. scrollbars and buttons).

Proposal Feedback

In Milestone 1 we were given great feedback from the professor on how our originally proposed project did not have a set focus on whether we were enhancing the visualization or the interaction side of a Digital Audio Workstation. He suggested, If we were to do both we needed to ground our work in a specific side to help strengthen our motivation. Therefore, we went back to the drawing board and we came up with a UI that implemented a new interaction techniques for accomplishing audio editing. This is not well supported by the current "standard" techniques (i.e. scrollbars and buttons).
Features of our application include:

1. Immediate Feedback of the audio going through a signal flow
2. 3D Graphics that enhance the experience of manipulating music
3. Multi-touch gestures to help interact between the interfaces and
4. Give it mobility to take with you anywhere.

Application Description

UI (high level): Initially, the user will see a black screen that yields a 3-Dimensional canvas with serval visualizers on top of one another. This is a layman implementation of a digital audio workstation more specifically the audio signal process flow, so all of the user interaction lies between two audio visualizers. The initial visualizers are vertical gray bars that display the audio at around 32 different frequencies. Then, the user may select different processes to add to the signal flow which end up in between the visualizers. All visualizers that manipulate the audio are displayed in a 2D circular shape. The circular nature of the display is to leverage touch gestures when the user is on a mobile device.

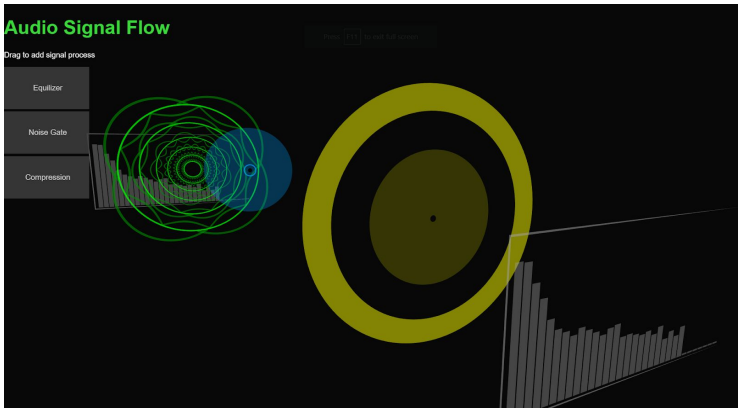
The UI is split into two modes, a multi-view mode designed to display the entire signal flow, and a single display mode to allow a user to modify the audio of the process at that step in the signal flow.

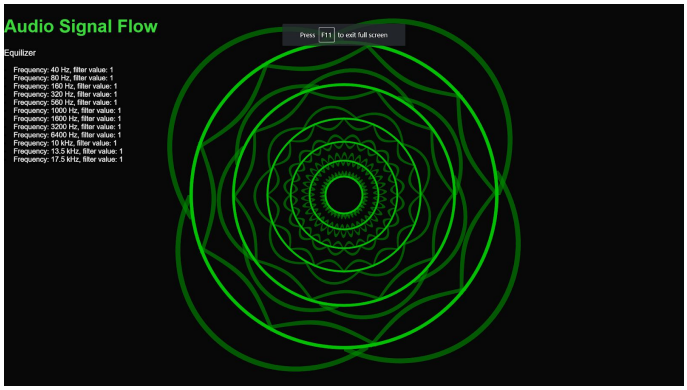
Multi-view screen:

The user will drag over a process to add to the signal flow. To remove a process, they drag it over to the left side of the screen. The user has the ability to scroll through the several visualizers, by dragging on the screen to the left or to the right. The user can also grab a visualizer and drag it around, thus moving the entire chain of processes. To interact with a single process, the user can grab the associated visualizer from the button on the left-side and drag it to the middle of the screen.

Single view:

The user will gain access to a 2-Dimensional canvas that can manipulate audio. Each visualizer will have this same structure. Below is a thorough description of each visualizer and their gestures:





Gestures:

Multi-view: Home Page

- Click and drag:
  - Click off of button - adds a new process
  - Click off of visualizer - scrolls the page
    - Drag to the left side of the screen - deletes the visualizer
    - Drag to the top of the page - Goes into single view mode for that visualizer

Single-view

1. General
    a. Swipe left across 75% of the screen to return to the multi-view.
2. Identity - there are no actions able to be performed on this, except for return.
3. Noise Gate
    a. pan up and down to adjust the threshold
4. Compression
    a. Pan up and down to adjust the threshold
    b. Two-finger rotate to adjust the ratio
5. Equalizer
    a. Click and drag
        i. Inward - maxes the value on the frequency band
        ii. Outward - Zeroes the value on the frequency band

Audio: Automatically built in for testing purposes

Development Environment

References & Technologies

For this application we used the Angular command-line to run the web application. Now, the other technologies we want to use are:

1. **Angular 6:** A TypeScript MVC framework that runs in browser JavaScript engines. Angular dynamic programming feature allows for ease to develop cross-platform application
2. **Three.js:** a cross-browser JavaScript library and Application Programming Interface used to create and display animated 3D computer graphics in a web browser.
3. **Hammer js:** gives us access to mobile gesture events that are not normally found in the browser, including tap, swipe, pan, pinch, press, and rotate. If your audience will be consuming your app on a mobile platform, these events are critical for building a solid user experience.
4. **Node.js:** An execution environment for event-driven server-side and networking applications that can be executed asynchronously from the UI, while handling synchronous UI in the main application.
    a. **NPM-** Package Manager for Node.js

Useful Links

Audio Visualization Resource:
https://github.com/willianjusten/awesome-audio-visualization
Normalize audio, https://www.learndigitalaudio.com/normalize-audio
App based on this code:
https://github.com/makimenko/angular-three-examples/blob/master/src/app/scene/scene.component.ts
2D drawing in Angular, https://teropa.info/blog/2016/12/12/graphics-in-angular-2.html

References

1. Obrenovic, Zeljko. “A flexible system for creating music while interacting with the computer.” *ACM Multimedia* (2005).
2. AngularJS Software, https://angular.io/docs, Last visited November 26, 2018
3. Hammer.JS Software, https://hammerjs.github.io, Last visited November 26, 2018
4. Three.JS Software, https://threejs.org/docs/, Last visited November 26, 2018