Proposal for a "Collaborative Real-Time Music Making" System

Colin Sullivan
Music 256a
Final Project Proposal

Fall 2011

Overview

The idea behind this project is to allow multiple people to simultaneously create the same piece of music together, each using her/his own abstract interface to a single musical instrument. There are two settings which this would be applicable:

1. All users are present in the same physical space and are all using their own interfaces but are listening to the same audio output device.
2. All users are located physically far apart and the interface devices also double as audio output devices, which will (theoretically) be playing the same sound as all the others in real-time.

Motivation

To facilitate music making for people who may not have other means to do so. Providing users with a unique experience of creativity and community.

Description

Ideally, the final product would be an interesting interface that is enjoyable to make music with. I imagine something like Brian Eno and Peter Chilvers' "Bloom"[1], where the UI is very minimal yet intuitive, and some of the power of the instrument lies in its modest aesthetic. As users manipulate the controls

[1] Bloom: http://www.generativemusic.com/index.html

of the instrument, they will be able to see the manipulations that others are performing, and will be able to interact with them as well.

As these manipulations occur there is a centralized synthesizer that is generating the audio and playing it in real-time. Therefore, another aspect of the system is that it must feel responsive to the user (i.e. they must hear the changes immediately upon making them in the interface).

Design & Architecture

UI

As simplicity is a central goal, manipulating shapes on a screen seems to be a user experience that would be intuitive to most people. The entire "canvas" will represent a loop of audio time, the x-axis representing time and y-axis for pitch. Different timbres will be be represented by placing different shapes/sizes on the canvas, and they will have different manipulatable properties depending on which type of shape they are[2].

To develop this application, I think it makes the most sense to utilize the new wave of UI standards in the browser. This will not only ensure that the application is widely distributable, but will allow for the UI to be run on low-cost devices such as small touch-screen tablets. Each of these UI clients will connect (via WebSockets) to a single server which will handle all of the audio processing and synchronization[3].

Backend

For the audio processing server, I will develop in C++ using elements from the "Synthesis Toolkit"[4]. This audio will either be played directly from the server, or will be transmitted back to the clients as raw audio buffers for playing on their own machines[5]. These two cases would be useful in different situations; If this application is used in an art installation where all of the users are in the same room, it makes sense to just have the server rendering the audio out to its own DAC. On the other hand, if the intention is to allow people to play music remotely with eachother, then the audio would need to be sent back over the WebSocket. Playing the audio directly on the server would clearly be more efficient, as the raw audio buffers wouldn't need to be transmitted over the TCP socket. Either way, this is a few lines of code and could be experimented with at a later time.

[2] UI Mockup: https://docs.google.com/drawings/d/1AeDupyQLsePJazCg58aobDYARffH7eDnZB4r8iKoJZA/edit
[3] Architecture Overview Diagram: Architecture_Overview.svg
[4] The Synthesis Toolkit in C++: https://ccrma.stanford.edu/software/stk/
[5] Architecture Tradeoff Diagram: Architecture_Tradeoff.svg

Testing

The best "measure of goodness" that I can think of is the results of a survey asking people who have used the application if they felt like they were making music successfully, if they felt like they were collaborating with others successfully, etc.

Milestones

At minimum, this project could result in a interface where users click on a well-defined grid point in the canvas to add a dot there. When the playhead gets to that dot, a single note will play. People can collaboratively add these dots, so in effect there is only a single timbre in the entire instrument.

At maximum, the result could basically be what I have described throughout the document. There are multiple timbre options, and the items can be manipulated to change the sound that they make when the playhead moves over them.