Live Music Performance with Android

18-551 Spring 2014 – Project Proposal

Michael Nye (mnye), Michael Ryan (mer1)

The Problem

Music performance applications on Android are virtually non-existent. Part of this problem is that historically, Android has not had good audio or touch latency, and music applications require very low latency to feel responsive. Modern Android versions have latency on the order of 100-150 ms. While this isn't yet low enough for a traditional keyboard-like interface, it is low enough to allow for performance aspects with an appropriate interface, with CD-quality audio (44.1kHz, 16-bit samples) in real-time.

The Solution

Firstly, we will provide instruments with the typical set of processing functions. This means, at a minimum, a typical subtractive synthesizer with a set of oscillators, equalization filters and enveloping. It also means providing expected signal chain tools such as compression and reverb. Secondly, to make the tool usable, we will provide an interface for triggering loops and scheduling short (1-4 measure) passages. This schedule can then trigger events in our native code processing engine to produce close to real time audio generation.

What We'll Do

The end goal of this project is a functional, user-friendly synthesizer and music sequencer for Android. The final demo will be an app running on either the Motorala XOOM or a 2013 Nexus 7. Data from the user (touch events, button presses), will be the driving force behind audio production. Audio generation can be handled with a subtractive synthesizer or other methods of production (additive synthesis, granular synthesis, etc) that are to be determined. The Stanford University Synthesis Toolkit (STK) is a candidate for inspiration as a signal processing library that can be implemented on Android.

The project has several substantial components. First, we need to establish an audio generation engine that can run natively and receive input events from Android devices. This functions as a backend to an Android front-end that must allow users to configure the synthesis tool chain and also schedule music. Once prototypes for both are complete, we can expand the application to include a wider variety of synthesis algorithms and an expanded library of effects. With a proper backend design, adding new effects should be relatively simple.