Matt Kent

mwkent

Grading Criteria:

Synth:

Minimum: Saw tooth wave at a fixed frequency

Fully working: Saw tooth wave oscillator (monophonic) that takes frequency from pitch and note on/off messages

Extra credit: polyphonic synth; takes frequency from pitch and note on/off messages from an external keyboard

TouchOSC:

(Original) I will program controls on a pre-loaded TouchOSC layout to alter the vocoder output.

Minimum: 5 controls with documentation for use

(New) I decided to use a TouchOSC interface for the synthesizer. However, I did not provide an interface for controlling the vocoder because we had a working polyphonic synth, which provided enough depth to the sound on its own.

I made sendOSC.srp that would take midi input from the midi keyboard and send OSC messages to either the same or a different computer. I wrote server.srp, which tested receiving OSC messages. I also worked on the synthesizer, which would play notes based on the OSC messages that the program received. OSC messages would be sent from a device using TouchOSC or from the MIDI keyboard in conjunction with sendOSC.srp. I also wrote vocodermodulator.c, which dealt with applying bandpass filters and volume detectors to the voice.

These programs worked fine. The synthesizer could play up to 12 notes at once, but we could have increased the number if we chose. The only issue was when OSC messages would be dropped because of the network connection, but that could not be helped. We made it so that you could hit enter when the program was running, which would stop all of the saws from playing in case a note off message was dropped.

Our project could have been better with a TouchOSC interface to control some of the parameters of the code. However, this would have added extra depth to our project that was not all that necessary because there was enough going on between the polyphonic synthesizer and voice combination.