Synthesis ToolKit + FluidSynth

mus256a / cs 476a | fall 2010
Nicholas J. Bryan
Synthesis Toolkit in C++ (STK)

- open source audio signal processing and algorithmic synthesis
- realtime, cross platform
- used for prototyping and production
  - most of chuck's unit generators are internally from STK
- can be used with RtAudio/RtMidi or with any other audio drivers that gives access to the audio buffer
STK overview

- **generators** - ADSR, sine, noise, saw, square, etc.
- **instruments** - physical models, sampler
- **effects** - chorus, echo, reverberation, pitch shift
- **filters**
  - delay, biquad, interpolated delay, FIR, IIR filters
  - no filter design, just implementation
- **midi I/O**
- **audio I/O**
sample STK code - reverberation

#include "NRev.h"

NRev * g_rev = new Nrev(5.0);

for( int i = 0; i < nBufferFrames; i++ )
{
    SAMPLE temp = .5*(buffy[i*2] + buffy[i*2 + 1]);
    temp = g_rev->tick(temp);
    buffy[i*2] = buffy[i*2 + 1] = temp;
}
what is FluidSynth?

- open source project for real-time wavetable synthesis (application and API)
- cross platform
- renders SoundFonts (SoundFont 2 specification)
- realtime audio effects such as reverberation, modulation, etc.
- render MIDI files
what is SoundFont?

- file format for sample based instrument sounds or wavetable playback
  - a collection of audio files / pre-recorded sounds
  - Example: piano.sf2
- used in conjunction with MIDI and/or General MIDI
  - specifies instrument to MIDI program change
    http://en.wikipedia.org/wiki/General_MIDI
- allows you to synthesize pitch/volume with only a limited number of pre-recorded sounds via wavetable synthesis
why is this awesome?

- FluidSynth + SoundFonts
  - allow you to flexibly synthesize instruments using pre-recorded sounds
  - allows you to separate out synthesis control and the actual sound
  - changing an instrument sound in just one line
    - load("piano.sf2) vs. load("violin.sf2)
what else?

- audio driver
  - could replace RtAudio
- midi driver
  - could replace RtMidi
- wavetable synthesis with midi control
- midi sequencer
  - real-time or from MIDI files
the API

- C style API
- load SoundFont
  - fluid_synth_sfload()
- trigger MIDI Events
  - fluid_synth_noteon(), fluid_synth_noteoff(), etc.
- process audio
  - fluid_synth_write_float()
- recommendation: create a C++ object to wrap the fluidsynth mechanisms you desire
//Setup
m_settings = new_fluid_settings();

m_synth = new_fluid_synth( m_settings );

// Trigger note
fluid_synth_noteon( m_synth, channel, pitch, velocity );

// Render buffer
fluid_synth_write_float( m_synth, numFrames, buffer, 0, 2, buffer, 1, 2 );
compiling fluidsynth

• OMG...it might take a while

• numerous dependencies for OSX
  • Fink, libgnugetopt, readline5, libflac8-dev, libsndfile1-dev, and glib2-dev

• watch out!
sample fluidsynth + stk project
project brainstorming

choose a limited subset:

audio input, audio output, OSC networking, graphics, hid control (joystick, computer keyboard, mouse, music keyboard), midi file playback, social (twitter, facebook, myspace, etc.), audio synthesis, audio effects/processing, other

idea:

repeat: