Digital Theremin Synthesizer and Visualizer

6.111 Final Project
November 15, 2007

Alexander Spicer, Daniel Rodgers, Jeffrey Chang
Project Overview

- Functionality of end product:
  - Musical instrument with Theremin-style input
  - Audio effects synthesizer
  - Real-time music visualizer

- Should be fun to use and entertaining to watch!
Project Overview, cont.

- **System inputs:**
  - Movement of user’s hands (while wearing gloves)
  - External audio input (e.g., mp3 player)
  - Computer keyboard (to select/apply audio effects)

- **System outputs:**
  - Playback of music from mp3 player, via speakers
  - Tones/effects generated from Theremin input
  - Real-time music visualization on LCD monitor
Project Overview, cont.

- Our system will have 3 main modules:
  - Input module
    - Processes input from the video camera
  - Audio effects module
    - Applies audio filters as specified
  - Visualizer module
    - Displays position of hands, plus visualization of music
Top-level Block Diagram

Project Top Level

Labkit switches

Debouncer

Video Camera Input

Video Processing (Alexander Spicer)
- Determine hand positions.
- Convert right hand x-coordinates to frequencies
- Convert left hand y-coordinates to volume

Audio Processing (Daniel Rodgers)
- Sound Generation
- Echo/reverb
- Flanging/Chorus
- Vibrato
- FFT
- Harmonizer

Visualizer (Jeffrey Chang)
- Pitch
- Volume
- Rhythm
- Multiple modes
- Background/Foreground
- Graph of Hand Position

Keyboard Decoding (Dan Rodgers)

ac97 audio stream

Audio Output

Video Output

15-bit representation of frequency between 0 - ~22000 Hz
8-bit volume

x and y coordinates of hands

15-bit representation

//
8

//
8

//
8

//
8

//
8

//
8

//
8

//
8

//
8

//
8
Video Processing Module

- Gloves on Hands
  - Red on Right, Blue on Left
- Control pitch range with input switches
- Calculate linear scale for pitch
- Harmonizer system
  - Rounds note to closest note that matches the harmonics of background music. (FFT)
**Block Diagram: Video Processing**

- **Video Camera Input**
  - NTSC Video Decoder (J. Castro)
    - Convert camera data to Y/Cr/Cb
  - Labkit switches
  - Debouncer

- **Video Processing**
  - NTSC to ZBT
    - Store Luminance data in ram
  - Chrominance Calculations
    - Calculate positions of left and right hands based on chrominance information (Cb/Cr)
  - Volume/Pitch Calculations
    - Convert linear hand position to logarithmic pitch and linear volume based on switch settings for scale.

- **Video Output**
  - Show averaged positions of left and right hands over B/W luminance video
  - Frequency
  - Volume
Block Diagram: Video Processing

Video Camera Input
- NTSC Video Decoder (J. Castro)
  - Convert camera data to Y/Cr/Cb

Debouncer

Chrominance Calculations
- Calculate positions of left and right hands based on chrominance information (Cb/Cr)

Volume/Pitch Calculations
- Convert linear hand position to logarithmic pitch and linear volume based on switch settings for scale.

NTSC to ZBT
- Store Luminance data in ram

Video Output
- Show averaged positions of left and right hands over B/W luminance video

Labkit switches

Frequency

Volume
Block Diagram: Audio Effects

- **FFT**
- **Harmonizer**
- **Tone Generator**
- **Chorus**
- **Flanging**
- **Delay**
- **Reverb**
- **Audio Stream Combination**

Connections:
- External audio
- Frequency
- Volume
- FFT Data
- Modified Theremin Audio
- Combined Audio
- External Audio
Audio Effects Module

- Responsible for generating audio given frequency and volume from Theremin

- Four different effects can be added to Theremin audio – Chorus, Flange, Delay, and Reverb

- Computes FFT of external audio and stores result in BRAM

- Outputs combined audio stream to AC97

- Advanced Theremin tone generation
  - More realistic tones through additive synthesis.
Visualizer Module

- Hand positions will be displayed as vertical and horizontal bars on left and bottom of screen, respectively

- The visualizations will take into account the source of the audio (external vs. Theremin), and will also utilize stereo data (left vs. right speaker)

- Extra feature, if time permits: Save screenshot into BRAM for later retrieval
Visualizer Module, cont.

- Three different modes that the user can select from:
  - Mode 1: “Fire”
    - Uses frequency data from the FFT, pitch varies horizontally, intensity varies vertically, has pixels that gradually drop from the peaks
  - Mode 2: “Kaleidoscope”
    - Primarily depends on volume, rhythmically jumps from one configuration of shapes to another, symmetrical around the center
  - Mode 3: “Rainbow Ladder”
    - Primarily depends on pitches detected, vertical position changes with pitch, color changes with time (duration of note)
# Timeline and Milestones

<table>
<thead>
<tr>
<th>Feature</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bare-minimum working system</td>
<td>All</td>
<td>11/18</td>
</tr>
<tr>
<td>Basic audio effects (e.g. echo)</td>
<td>drodgers</td>
<td>11/25</td>
</tr>
<tr>
<td>Displaying hand position on screen</td>
<td>jchang1</td>
<td>11/25</td>
</tr>
<tr>
<td>Color detection, frequency scaling/centering</td>
<td>aspicer</td>
<td>11/25</td>
</tr>
<tr>
<td>Having a functional FFT</td>
<td>drodgers</td>
<td>12/2</td>
</tr>
<tr>
<td>One mode with simple visualization</td>
<td>jchang1</td>
<td>12/2</td>
</tr>
<tr>
<td>More complex audio filters, or sampling</td>
<td>aspicer</td>
<td>12/2</td>
</tr>
<tr>
<td>Additional audio effects (e.g. instruments)</td>
<td>drodgers</td>
<td>12/9</td>
</tr>
<tr>
<td>Multiple modes, more complex visualizations</td>
<td>jchang1</td>
<td>12/9</td>
</tr>
<tr>
<td>Harmonizer (discretizing input to match)</td>
<td>aspicer</td>
<td>12/9</td>
</tr>
<tr>
<td>Final report and presentation</td>
<td>All</td>
<td>12/12</td>
</tr>
</tbody>
</table>
Thank you!

Questions?