Maestro

Team 4
Evie Kyritsis, Sabina Maddila, Janelle Wellons
The Theremin

- Early electronic musical instrument controlled without physical contact by the thereminist
- Has two metal antennas
  - Senses the relative position of the user’s hands
  - Controls oscillators for frequency with one hand
  - Amplitude (volume) with the other
Overview

Our system is comprised of three components:

- **Motion Tracking**
  - Processes incoming user input data via NTSC camera
  - Outputs cartesian and polar coordinates

- **Sound Generation**
  - Takes in polar coordinate from Motion Tracker
  - Takes in data from accelerometers
  - Calculates and produces a tone
  - Outputs to ac97 on FPGA to external speakers

- **Visualization**
  - Takes in the cartesian coordinate from motion tracker
  - Generates objects that ‘follow’ the user’s hands on the computer monitor
Motion Tracking

Inputs:
- NTSC Camera Data

Outputs:
- Polar coordinate
- Cartesian coordinate

Modules:
- Decoder that takes in NTSC Data -> Ycrbc format
- Ycrbc -> RGB
- Prepare data for loading into ZBT Ram Memory
- Thresholds color values to identify object & produce coordinate

Implementation:
- Gloves, brightly colored fingers
- Begin with shading tests & crosshair tracking
- Play around with color values to identify our colored gloves
Sound Generation

Inputs:
- Polar Coordinate
- Accelerometer Data
- Keypad Input

Outputs:
- Tone
- ac97 on FPGA to external speakers

Modules:
- Frequency Calculation
- Sine Wave FFT Generator
- Output to ac97
- Debouncer

Implementation:
- Volume controlled by external keypad
- Begin by prototyping formula for sine waves in python first
- Will test not using the motion tracker but will instead have coordinates fed in by other means
- 3-axis accelerometer with SPI digital interface
Visualization

Inputs
- Cartesian coord

Outputs
- VGA output to computer monitor

Modules
- Generate XVGA signals
- Sprite Generator
- Send to VGA output

Implementation
- Use the same wheel described in Sound Gen, in order to represent certain tones as a color
- Will have objects that follow the hands
- Start with testing separate from Motion Tracker, will integrate later on
- Tricky bit: figuring out the timing between receiving the camera input and outputting the response
Timeline

- 5-Nov: Finish Prototyping Sound Gen
- 5-Nov: Acquire Threshold Values for Camera Input
- 12-Nov: Design and Create Simple Sprites that respond to Labkit Input
- 12-Nov: Generating Simple Sine Waves using FFT from Labkit Input
- 19-Nov: Basic Tracking with NTSC Camera; Outputting Coordinates
- 25-Nov: Implement Background Graphics
- 25-Nov: Integrate with Accelerometer Input and Incorporate Tamper
- 25-Nov: Coordinate with Visualization Block with Timing Outputs
- 30-Nov: Sprites Following Hands (Responding to Motion Tracker Block)
- 7-Dec: Integration
- 7-Dec: Debugging!
Questions?