Sabina Maddila
Evie Kyritsis
Janelle Wellons
6.111 Project Proposal Draft

Maestro

Introduction

Our project, first inspired by the Theremin -an electronic instrument that produces sounds without physical touch- has become more of a gestured piano. The concept in its simplest form is to make your own music with your hands. The motivation is that we wanted to incorporate as many forms of stimulation and senses that we could implement feasibly- ie sound, visuals, and movement. Our project translates a physical performance into both music and visual art, therefore interacting across multiple media. As a result of combining both visual and audio art forms, we also have applications for music visualization for the deaf and hard of hearing or even the blind in regards to color recognition through sound.

Our system is comprised of three main components: User Input, Sound Generation and Visualization. Some core issues that we foresee are integration, and coordination of all the user inputs which we will discuss further on.

Overview

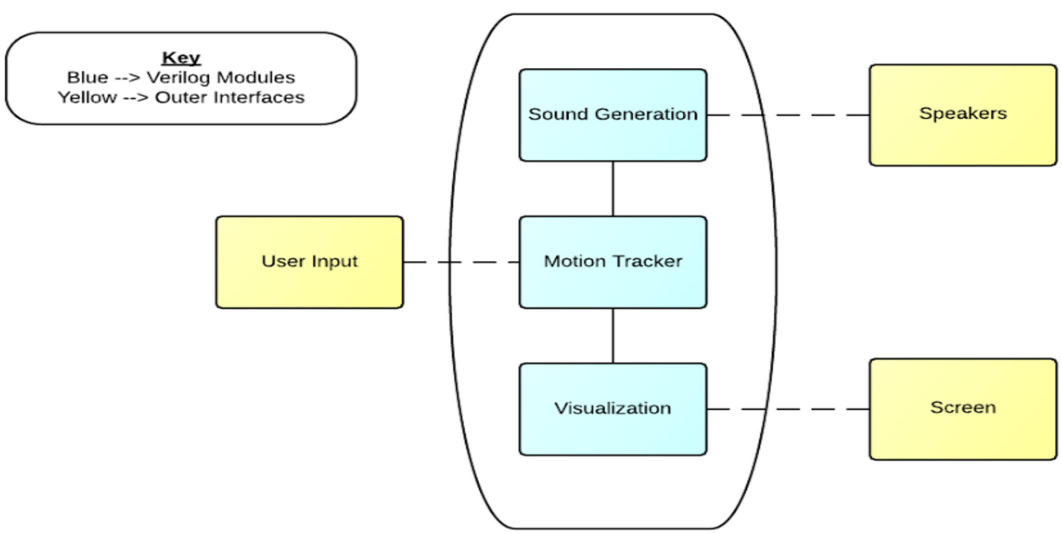

Figure 1: High-Level Block Diagram

Our system is comprised of three main blocks:

Motion Tracker, where inputs from the hardware by the user of the system will be collected and processed. Two main forms of input will be through a VGA camera and accelerometers. Others will be established, such as a microphone, if we ever get to adding a playback feature.

Sound Generation, which takes in the processed data from User Input and uses it to determine and calculate coordinates that will then be used to determine a specific frequency and waveform of the sound as well as for how long the sound should be sustained.

Visualization, which will also take in the processed data from User Input and convert the coordinates into a color. The coordinates from camera tracking will also be used in producing objects that will show the position of the user's hands on the computer screen and will follow the hands in real time as they move about in front of the camera.

The sound generation and visualization fall from the data processed from the user input block. On screen, the user will be able to determine the position of their hands since objects on screen will follow them, and we will incorporate a mode where the wheel that the sound and colors are based on can be shown to the user if they wish. We would like everything to be in real time, and so the use of both the camera and accelerometers to track movements we thought would best capture the user's movements. The whole system is based on a polar coordinate system, with the origin being placed at the center of the computer screen. For features such as playback and volume control, which will be added later on, if time permits, would be controlled through external keypads. These design choices were made so as to be very intuitive for the user.

Modules

Motion Tracker

This module processes all of the user inputs and sends off the relevant signals to the other two modules. First we have tracking of the fingers, which will be done using a glove that has the top portions of fingers brightly colored so as to be able to produce simultaneous notes. Identification of the hands will be done via VGA input by an NTSC video camera. Taking in the movements of the user's hands, the module will generate polar coordinates that will be mapped to a position on the screen. These coordinates will be sent off to the Sound generation and visualization modules. Some challenges will be to track just fingers, so we will start with whole hand and then refine down to fingers.

This module will also be processing data received from accelerometers that will be placed at the palms of the user's hands. They will sense movement of the hand to sustain note duration. If someone wanted the sound to stop, they would stop moving their hand. To sustain a note, one would need to keep their hands vibrating back and forth within the range of the note. Challenges will be filtering the data, and determining the appropriate thresholds for movements. We will also need to wire up an analog to digital converter in order to take in the signals from the accelerometer and use them on the FPGA board.

Gloves with brightly colored fingers, and accelerometers in the palms that are directly wired to the board (via long wires to the gloves) will be the main source of input to this module and to the system as a whole.

Sound Generation

Sound generation is in charge of determining and producing all sounds of the system. It is comprised of sub parts that need to be determined individually. The module takes in the inputs from Motion Tracker, as well as possible keypad inputs, and uses these to map the data to a sound wheel which will be discussed under the pitch and tone. Once the sound is calculated, the output goes through the ac47 on the FPGA board and then eventually to external speakers.

Pitch/Tone

The actual frequency of the sound will be built off of a sine wave. We will be using the ac47 on the FPGA labkit. Taking in the polar coordinates from the motion tracker module, a tone will be produced depending on the position on the screen. The reason for polar coordinates, is that we will define a sound wheel that will fill the computer screen. The sound wheel is comprised of concentric circles, and one can imagine that a pie slice of this wheel corresponds to one specific note, like a C. This part of sound generation will define this wheel, by having range of frequencies that will map to specific pie slices of the wheel. So the coordinates received from motion tracker will produce a position that is mapped onto these concentric circles that represent the sound frequencies. As you move radially in and out from the origin, going from smaller to bigger circles, the octave of the note changes. The physical representation of a sound circle allows for smooth transitions between notes, since the physical coordinates determined by the camera tracking will go into a formula into calculating the specific sound.

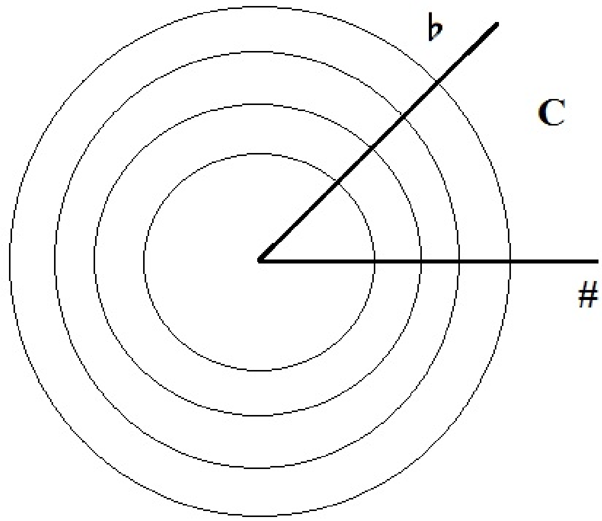

Figure 2: Sound Wheel

Tamber

We want our sound to sound nice, and so we want to add the feature of having the sound represent another instrument, like a violin, or change it in other ways. To accomplish this, we will have to add more sine waves on top of the one that we are already producing.

Duration of Sound

The sound will fade in and out according to accelerometer data rather than abruptly turn on and off. The sound will be sustained for as long as the hands are moving. When the accelerometers receive data under a certain threshold, the sound stops/fades out. A challenge will again be thresholding the data for the accelerometer correctly, as we have to account for the hand moving to change notes. And the user may not want to play all the in-between notes to get to the final, desired notes. The accelerometer will have to only detect a back-and forth movement. Any movement in one direction should not produce a sound in order to change notes.

Volume

Volume will be adjusted by a switchboard that we will make out of a circuit board and LEDs or a simple keypad.

Visualization

Color Wheel

The same map of concentric circles for the sound will also map the colors. Colors and sounds are both made of different frequencies so in our research, we discovered the colors that corresponded to specific sounds. There are calculators that convert tones to colors, and we will need to implement the formula. As moving radially changed the octave of the note, moving radially changes the hue or brightness of the color. Each note corresponds to a true color. For example, a C is in the red frequency domain. The same inputs from motion tracker for sound generation will be fed into visualization except for keypad inputs

Objects following the hands

We wish to have two objects that follow the hands on the screen, so that a user may know where they are on the wheel of color and sound. The objects themselves may have the added feature of changing colors themselves based on position on the color wheel. We will start with implementing circles most likely, then moving on to more complicated shapes if we have time. The challenge will be to implement the smooth transition of colors through the use of blending.

Background

While colors are changing with the objects, the background itself will change color, perhaps in complementary fashion or on it's own color formula. But we do wish to have a dynamic background. If it has objects, we will try to implement scrolling which will be a challenge in and of itself.

Implementation Process

We have a pretty straight forward method for dividing the work. One person will be in charge of one module for now, unless our plans change. For the first week will focus on ordering and planning the hardware that we will need and prototyping our system. Then, once we hash out what is feasible and determine what needs to be changed, we will begin implementation in Verilog. What will be key is designing our modules so that they may be tested and simulated individually so that debugging goes as smoothly as possible. Especially since we do not want the sound gen and visualization module testing to based solely on the data processed in the motion tracker module when debugging. That way we minimize time needed to compile on the FPGA board itself.

Our initial implementation plan is as follows:

Week	Tasks
10/30-11/2	begin things that should be done by 11/5
	incorporate ideas+topics covered with meetings w staff
	think of parts that we want to prototype
	think of inputs and outputs
	order components
11/5/2014	define algorithm for colors and sound
	tracking with a camera
	accelerometer choice and wiring
11/12/2014	take input of camera tracking to other modules as inputs
	focus on producing the sound
	foucs on producing the objects that will follow hands on screen
	inteface with accelerometer (wire it up) + calculations
11/19/2014	averaging data from user input of camera tracking
	integrating acceleromter and camera input data
	taking accelermometer data to correspond to sound duration
	objects changing color
11/26/2014	finish implementation, integrate modules
11/30/2014	done with project
12/7/2014	done with debugging

For video tracking, we will first be starting with identifying and tracking one hand. Then attempting identify and track two hands. After this we will need to determine if we can track object of smaller size (like fingers), and then see if we can determine the position of an object by it's relative size to the camera. Additionally we will need to determine an accelerator of our choice, interface with it using the FPGA, and then reliably read its output.

For sound production, we will need to properly map the video input to a frequency that we will then synthesize. After this we will need to determine the duration and additionally synthesized harmonics for the sound output.

For visualizations, we will need to properly map the video input to a video output. Then we will need to project our output based on the location of the users hands and 'color' of the synthesized sound.

Conclusion

Beginning with prototyping and organizing the hardware requirements in the first week, we hope to have a basic framework of our system completed in 2 and a half weeks, with the remainder of the time focused on finetuning the rest - leaving a week for hashing out any bugs or unforeseen problems. The only main challenges we expect to have are coordination and integration of the modules and data from user inputs. The actual content and creation of the modules we predict will be straightforward.

We are excited to be creating what we think will be a fun experience in performance art, and hope that other future projects will be inspired by the culmination of sound, color, and movement that is Maestro!