Leap Motion Piano

Patrice Liang pl2279
Matthew Patey mep2167
Vanshil Shah vs2409
Kevin Walters kmw2168

Overview

Leap Motion Piano allows users to produce audio by "playing" an on-screen piano by simulating key presses with their fingers. We display a virtual piano on the VGA screen and use the Leap Motion device as input to the FPGA to detect the user's finger motions. The raw USB data from the device is then processed to correspond to individual key presses, each of which triggers a simple key animation and a unique musical note.

Input

1. Leap Motion USB device

Output

1. VGA screen
2. Audio of piano

Algorithm

1. Gather raw USB data from the Leap Motion controller
2. Filter data to receive only locations of fingers on a 3d plane
3. Create piano animation in software
4. Use the finger location data to display fingers on the screen and simulate piano key presses.
5. Use piano key data to playback audio on an audio device connected using the audio line out jack.

Hardware/Software

The available Leap Motion software API currently presents exposes useful data such as recognized gestures and coordinates of fingers. Leap Motion Piano's hardware component will interface directly with the Leap Motion device (USB) and present an API at a similar level of abstraction as the software API. This will require processing the data from the device to determine the number of objects it's detecting, their coordinates and possibly some predetermined gestures. The software will present and animate the keyboard, obtain information from the hardware API about the position of fingers and use this to render fingers on screen and play sounds when appropriate.

Timeline

Milestone 1

The first milestone will consist of implementing the hardware component using SystemVerilog in order to communicate with the USB device. This will include creating a hardware API to replace the software API that is available for the device so that we can process the input in hardware. We will also be filtering the incoming data at a low level so as to only receive the coordinates of the fingers, since that is all that is needed.

Milestone 2

The second milestone will mainly consist of the purely-software portion of the project. That is, we will implement the on-screen piano graphics, including idle and key-pressed states. At this milestone we will also have finished the sound-generation portion, which will produce the appropriate note when a key is pressed.

Milestone 3

The final milestone will be tying all of this together. Using the low level data we received from milestone 1, and the software components of milestone 2, we will have hardware-level data input received from the USB device mapped to different actions, such as pressing a key. At this point, all main functionality will be completed, and the only remaining steps would be to prepare for the presentation.