Y2.02K

Final Project for Programming Sound @ RISD. Sensor Insturment with Pure Data and Leap Motion Controller. Video - screen recording of Leap Motion Visualizer.

Control

Gestures:

- Move the palm in X, Y, and Z directions
- Open/Close the palm
- Circle/Swipe/Kep Tap/Screen Tap
- Number of fingers

Considering the possible glitches of sensor failures, I deliberately chose to have a consistent organ sound throughout.

Y coordinate controls the pitch of organ.

X, Z coordinates: the XZ-plane was divided into 16 sub-sections and each has either sample or oscillator sound that will play once the palm position triggers.

Finger count tweaks the oscillators in organ.

Sphere radius represents the volume control. Volume is high when you open the palm and low when you close it.

Either circle, swipe, key tap and screen tap will generate a random dial sound.

Hardware

Leap Motion: the technology of yesterday! I have been facinated with gestural controlling for quite a while and this is the first time using this sensor, which tracks hand movements and collect data. In this project, it tracks:

- Palm position
- Sphere Radius
- Finger Count
- Circle
- Swipe
- Key Tap
- Screen Tap

Software

Pd Patches iterated from Leap Motion for Pd by Chikashi Miyama

Besides the patches have been offered already in Miyama's Version 1, I have three layers of patches creates/modified specifically for the Y2.02K project.

- final.pd is the master patch to turn Leap Motion on and off. It also shows live data collected when Leap Motion is on.
    - palm-input.pd connects hand data with sound patches.
        - organ.pd creates organ sound.
        - xz-coord.pd samples sound that to be played based on palm position.
        - volume.pd controls the volume of the palm-input patch
    - gesture-input.pd connects gesture data with dial patch.
        - dial.pd generates a random 0-9 dial sound.