

# 21M.385 Lecture Notes

---

## Week 5

### Types of Input Devices

- Game controllers – gamepads, some with analog sticks, computer keyboards.
- Specialized controllers – steering wheels
- Button controllers - Guitar Hero / Rock Band guitar and drums, Launchpad controller.
- Phone/Tablet touch controllers
  - Lots of flexibility – touch/tap, continuous control, and gestures.
  - the display/feedback surface is the same as the control surface - when used well, this can be a nice advantage.
- Motion controllers
  - Sony Eyetoy
  - Wii (huge surprise for the games industry)
  - Sony Move
  - Kinect (at the time, was the most successful HW launch ever)
  - Leap Motion

### Kinect (xbox 360 version)

- Two main technologies:
  - Depth sensing: creating a depth map, a 2D image where each pixels is a depth value, using the technique *structured light*. This happens inside the Kinect camera.
  - Skeletal tracking – a few different approaches (e.g., machine learning). This happens in software on the PC.
- Microsoft developed skeletal tracking for the Xbox and PC – but this cannot be used on other OSs.
- Primesense developed drivers and skeletal tracking that is available as compiled libraries (not open source). Apple bought Primesense in Nov 2013 and all support has been discontinued.
- Sadly, getting Kinect to run on modern OSs is becoming more difficult due to the lack of support.
- To get Kinect to talk to Kivy, use Synapse, written by Ryan Challinor. Kinect talks to Synapse which sends skeletal data via OSC (open sound control) to Kivy and the module kinect.py

### Leap Motion

- Similar to Kinect, but sensing area is smaller: 2x2x2 foot area above the sensor.
- Uses infrared lights and two infrared cameras create the depth map using stereoscopic image processing
- Skeletal detection for up to two hands. Lots of tracking and filtering to smooth over problematic cases (occlusion, interference)
- Leap is mostly focused on VR today, but the sensor is still great for motion tracking and creating music apps!
- Has python bindings as part of the installation package.
- We will use leap motion SDK “V2”. Documentation is here:  
<https://developer.leapmotion.com/documentation/v2/python/index.html>
- Try out the Leap Visualizer App.

### Leap in Kivy

- `import leaputil` and `import Leap` to get started.
- Create the leap controller with `Leap.Controller()`.
- On every `on_update()`, get a “frame” of data from the controller with `frame()` and find out what’s going on.

- You can either use the Leap API directly or you can use my helper function provided by `leaputil`.
- The coordinate system is relative to the Leap Motion sensor, and units are in millimeters.
- It can sometimes be easier to normalize coordinates from the interaction space to a unit-coordinate system (0-1 in x, y, and z).
- Use `Cursor3D` object (in `gfxutil` module) to plot normalized hand or finger positions. The most stable point appears to be the hand's palm position (`MainWidget2`).
- You can track two hands (`MainWidget3`).
- You can track the 5 fingers on each hand, but the data appears to be less stable (`MainWidget4`).

### Gestures

- Continuous control vs. discrete events.
- Make a separate class for a gesture detector. It's a useful abstraction. Use callbacks to communicate back results.
- `ClapGesture` example:
  - Simple state machine with two states: hands-together, hands-apart.
  - Use of hysteresis (or debouncing) to avoid chatter at boundary.

### Kinect in Kivy (if you choose to try Kinect for your final project)

- Run Synapse. Make sure your Kinect is plugged into USB and plugged into power.
- Create a `Kinect` object (from module `kinect`).
- Use `kinect.add_joint()` to register a joint for tracking.
- Remember to call `Kinect.on_update()` every frame
- Query for data using `kinect.get_joint()`
- There are three options for coordinate systems:
  - `kinect.kBody`: All values are in millimeters, relative to the torso joint.
  - `kinect.kWorld`: All values are in millimeters, relative to the Kinect camera
  - `kinect.kScreen`: x, y are in screen pixels (0,0 is top left), z is like `kWorld`.

### Final Projects

- Start thinking of ideas.
- I have a HW budget to purchase sensors or whatever you may want for your final project. But you will need to return this HW at the end of the term.
- Pitch sessions – get ideas flowing, start thinking about who has what ideas and who you may want to partner with.
- Projects should have 2 or 3 people per project.

MIT OpenCourseWare  
<https://ocw.mit.edu>

21M.385 Interactive Music Systems  
Fall 2016

For information about citing these materials or our Terms of Use, visit: <https://ocw.mit.edu/terms>.