



Human Interface Devices

hannes@ru.is

Types of Devices

- Keyboard
- Mouse
- Joystick
- Joypad
- Buttons
- Trackball
- Wheel
- Pedals
- Guitar
- Drums
- Dance pad
- WiiMote
- ...



Interfacing with a HID

- Polling
 - Read a hardware register
 - Read memory-mapped I/O
 - Higher level software interface
- Interrupts
 - Electronic signal switches CPU to process input
 - Processed by Interrupt Service Routine (ISR)
 - Data stored for later processing by game loop
- Wireless (e.g. Bluetooth)
 - Software needs to „talk“ to device
 - Can made to appear like Polling

Types of Input

- Digital Button
- Analog axes and buttons
- Relative axes
- Accelerometers
- 3D orientation
- Cameras
- EEG and EMG

Input Type: Digital Button

- Physical state:

Pressed / down \leftrightarrow Non-Pressed / up

- Electric circuit:

Open \leftrightarrow Closed

- Data (1 bit):

1 \leftrightarrow 0

NOTE: Mapping
between these is not
always the same

- A number of buttons can be packed into a couple of bytes, e.g. `wButton` in `XINPUT_GAMEPAD`

Input Type: Analog axes and buttons

- Range of values
 - Degree of press
 - Position (axes)
- Already digitized
 - E.g. Into 16 bit signed integers
- Example
 - Xbox360 game pad uses 16 bit signed integer for sThumbLX and 8 bit unsigned integer for bLeftTrigger

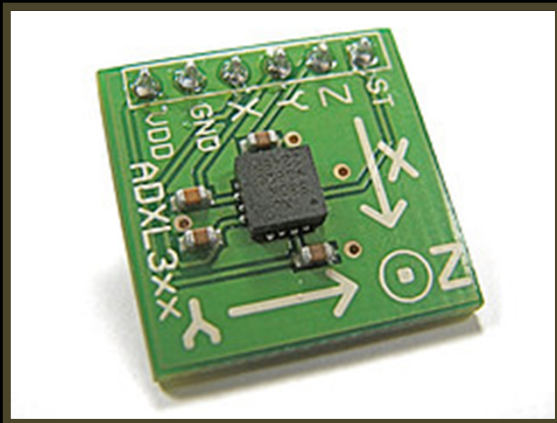


Input Type: Relative Axes

- **Relativity**
 - Absolute devices: Zero position given
 - Relative devices: No clear zero position given
- **Interpretation**
 - Zero means position/input has not changed
 - Non-zero provides info on position/input change
- **Example**
 - Mice, mouse wheel, track balls

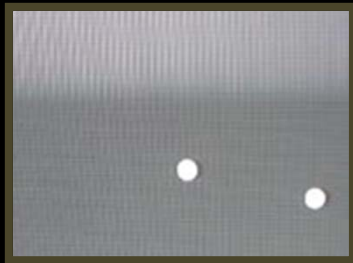
Input Type: Accelerometers (+ orientation)

- Relative analog inputs (like mice)
- Measure acceleration up to $\pm 3g$ along 3 axes
- Often digitized into 3 signed 8-bit integers



Input Type: Cameras

- Infrared Cameras
 - Map external infrared light sources onto 2D image
 - Process image for configuration information
 - E.g. WiiMote contains IR camera and measures position, length and orientation of a line segment between two detected IR lights in the „bar“



Input Type: Cameras (cont)

- Color Cameras
 - Track movements of „blobs“ of known color
- Depth Cameras
 - Store perceived depth in each pixel based on projected infrared pattern (e.g. Kinect)



Input Type: EEG and EMG

- Electroencephalogram (EEG) sensors measure electrical activity of the brain (weak signals)
 - E.g. Converted into levels of relaxation or concentration
- Electromyography (EMG) sensors measure electrical activity produced by muscles (stronger)
 - E.g. for signaling rotation of eyes or mouth opening



Types of Output

- Rumble
 - Motor rotated unbalanced weight
- Force-feedback
 - Motors providing opposing force
- Embedded Audio
- Lights

Game Engine HID System

RAW DATA → **Massaging + Abstraction** → GAME



- Dead Zone
- Analog Signal Filtering
- Event Detection
- Cross-Platform
- Input Re-Mapping
- Context Sensitivity
- Disabling Input

Dead Zone and Filtering

- Have to account for noisy data
- Dead Zone specifies a certain delta within which values get clamped down to Zero
- Filtering can get rid of noise at certain frequencies. Noise is often at much higher frequencies than user input, so „low pass filters“ or simple averaging can help

Event Detection

1. Button up/down event (change in state)

ButtonDowns =

(prevButtonStates ^ ButtonStates) & ButtonStates

2. Chords (multiple simultaneous buttons)

How do you tell chord from individual buttons?

- a. Use button effect plus the chord effect
- b. Delay button effect or effect only upon release
- c. Chord effect pre-empts button effects

3. Tapping and Sequences

Compare ΔT to a $\Delta T_{\max} = 1/f_{\min}$, if greater drop/reset

See code fragment
in book

Cross-Platform

- Hardware abstraction layer
 - Unified naming scheme
 - Map between physical device and abstract controller

Input Re-Mapping

- Possible to change control schema
- But only within certain control classes e.g.
 - Digital Buttons
 - Unidirectional absolute axes $[0, 1]$
 - Bidirectional absolute axes $[-1, 1]$
 - Relative axes $[-1, 1]$ (as offsets)

Context Sensitivity

- How to interpret certain inputs may depend on current activity or mode
- State machines can track current activity or mode

Disable Inputs

- Disabling input can happen at
 - Low Level (efficient but catastrophhic if buggy)
 - High Level (less efficient but more easily tracked)