Input Technology

Topics

Buttons and keyboards
Pointing
  ■ D-pad and digital joystick
  ■ Trackball
  ■ Mechanical and optical mice
  ■ Multitouch
Layered model of input
Beyond keyboard and mice
  ■ Game controllers
Keyboards
Key cap

Separating layer (with hole)

Top conductive layer

Bottom conductive layer
### Row/Column Scanning

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Q</td>
<td>W</td>
<td>E</td>
<td>R</td>
<td>T</td>
<td>Y</td>
</tr>
<tr>
<td>R2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Closeup**

- **C1**:
  - R1
  - R2

- **C2**:
  - R1
  - R2
One Key Down

C1

C2

R1

R2

One Key Down

C1

C2

R1

R2
3 Keys Down

C1

C2

R1

R2
Keyboard Matrix (16 x 8)

16 X 8 IBM Keyboard Matrix (columns are marked 'a' to 'p' and rows are marked '1' to '8')

<table>
<thead>
<tr>
<th></th>
<th>a1</th>
<th>a2</th>
<th>a3</th>
<th>a4</th>
<th>a5</th>
<th>a6</th>
<th>a7</th>
<th>a8</th>
<th>a9</th>
<th>a10</th>
<th>a11</th>
<th>a12</th>
<th>a13</th>
<th>a14</th>
<th>a15</th>
<th>a16</th>
</tr>
</thead>
<tbody>
<tr>
<td>b1</td>
<td>esc</td>
<td>F4</td>
<td>G</td>
<td>F5</td>
<td>H</td>
<td>F6</td>
<td>*</td>
<td>#0</td>
<td>.</td>
<td>Arrow up</td>
<td>L Alt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b2</td>
<td>L Shift</td>
<td>Tab</td>
<td>Cap Lock</td>
<td>F3</td>
<td>T</td>
<td>Bk</td>
<td>Space</td>
<td>Y</td>
<td>F7</td>
<td>#4</td>
<td>#5</td>
<td>#6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b3</td>
<td>L Ctrl</td>
<td>~</td>
<td>F1</td>
<td>F2</td>
<td>5</td>
<td>F9</td>
<td>6 =</td>
<td>-</td>
<td>F8</td>
<td>Del</td>
<td>Ins</td>
<td>Page Up</td>
<td>Home</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>F/0</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>F11</td>
<td>F12</td>
<td>Page Down</td>
<td>End</td>
<td>Prnt Scrn</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b5</td>
<td>Q</td>
<td>W</td>
<td>E</td>
<td>R</td>
<td>U</td>
<td>I</td>
<td>P</td>
<td>O</td>
<td>#7</td>
<td>#8</td>
<td>#9</td>
<td>#+</td>
<td>Scrl Lock</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b6</td>
<td>A</td>
<td>S</td>
<td>D</td>
<td>F</td>
<td>\</td>
<td>J</td>
<td>K</td>
<td>:</td>
<td>L</td>
<td>#1</td>
<td>#2</td>
<td>#3</td>
<td>#Enter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b7</td>
<td>R Ctrl</td>
<td>R Shift</td>
<td>Z</td>
<td>X</td>
<td>C</td>
<td>V</td>
<td>Enter</td>
<td>M</td>
<td>,</td>
<td>.</td>
<td>#num</td>
<td>Loc</td>
<td>#/</td>
<td>#*</td>
<td>Pause</td>
<td></td>
</tr>
<tr>
<td>b8</td>
<td>B</td>
<td>Space</td>
<td>N</td>
<td>/</td>
<td>Arrow Down</td>
<td>Arrow Right</td>
<td>Arrow Left</td>
<td>R Alt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Standard PS/2 keyboard has 104 keys

CS148 Lecture 5

Keys → Scan Codes

Make (onPress) and Break (onRelease) codes

http://www.computer-engineering.org/ps2keyboard/

CS148 Lecture 5

Pat Hanrahan, Fall 2011
Keys (Scan Codes) ≠ Characters

Special keys - interpreted by the OS or App
- F1, ..., F12
- Insert, Delete, Home, ...

Duplicated keys
- Numbers on keypad vs. keyboard
- Left-shift, Right-shift, Left-cmd, Right-cmd, ...

Keys ≠ Characters

Modifier keys
- [SHIFT] [OPTION/ALT] [CTRL] [CMD] [FN]
Keys ≠ Characters

Modifier keys
■ [SHIFT] [OPTION/ALT] [CTRL] [CMD] [FN]

[Shift]

[Option]
Orange = Dead Keys

CS148 Lecture 5
Pat Hanrahan, Fall 2011
Keys ≠ Characters

Modifier keys

[SHIFT] [OPTION/ALT] [CTRL] [CMD] [FN]

[Option-A]

CS148 Lecture 5
Pat Hanrahan, Fall 2011

GLUT Keyboard Interface

ASCI keys

glutKeyboardFunc(func) // onDown
glutKeyboardUpFunc(func) // onUp
    func(unsigned char key, int mousex, int mousey);

Non-ASCI keys (F1, ..., F12, INSERT, ...)

glutSpecialKeyFunc(func) // onDown
glutSpecialKeyUpFunc(func) // onUp
    func(unsigned char key, int mousex, int mousey);

Int glutGetModifiers()
    GLUT_ACTIVE_SHIFT | GLUT_ACTIVE_CTRL | GLUT_ACTIVE_ALT

N.B. Need to track key state
Flash Keyboard Interface

Key class
- `addMouseListener(proc)` - call `onUp` and `onDown`
- `getAscii()` - return ASCII for the last key pressed
- `getCode()` - return virtual key code for last key
- `isDown(key)` - return state of key
- `isToggled(key)` - return change in state of key

Note difference between
state (Up, Down) vs. transition (onDown, onUp)
Layered Model of Input

device abstraction

transformation

signal coding

sensing

physical properties

user action

Keyboard

Characters

Scan Codes

Switches

Keys

user action
D-pad

Famicom Controller (1983)
Atari CX40 Joystick

Just 5 switches!

\ o5 o4 o3 o2 o1/  
\ o9 o8 o7 o6 /

pin #
1 Up
2 Down
3 Left
4 Right
5 unused
6 Button
7 unused
8 Ground
9 unused

4-way Joystick (Just 4 switches)
Demo

Trackball and Mouse use Rotary Encoders
Trackball

Sensing: Rotary Encoder

High
Sensing: Forward Rotation

![Diagram of a sensor detecting forward rotation]

Sensing: Backward Rotation

![Diagram of a sensor detecting backward rotation]

CS148 Lecture 5

Pat Hanrahan, Fall 2011
Solution: Use Two Detectors

Sensing: Rotary Encoder
Sensing: Rotary Encoder

Coding:
HH -> LH: dx = 1
HH -> HL: dx = -1

High
Low

CS148 Lecture 5
Pat Hanrahan, Fall 2011

Mouse. Engelbart and English ~1964
Optical Mouse

1st generation (Xerox)
- Led + photosensor over a grid of lines

2nd generation (Agilent)
- CMOS imager + DSP
  - 1500 frames per second
  - 16 x 16 pixel resolution
  - 300 counts per inch

Mouse Cam

[Image of software interface]

http://www.bidouille.org/hack/mousecam/index.php

CS148 Lecture 5  Pat Hanrahan, Fall 2011
Multitouch

Capacitive Sensing - Button

http://elm-chan.org/works/capsens/report_e.html

Pat Hanrahan, Fall 2011
Self-Capacitance

Capacitive button

Whirlpool Microwave Oven

Invented by B. Stumpe at CERN 1972
Mutual Capacitive Sensing

Scan x and y to sense multiple points at once (ITO = Indium Tin Oxide = transparent conductor)

http://walkermobile.com
Multitouch Wall Displays

Perceptive Pixel

CS148 Lecture 5

Pat Hanrahan, Fall 2011
Gamepads
Analog Joystick

Gamepads

SONY Playstation 3

Microsoft XBOX 360
New Input Technologies

iPhone Sensors

Camera(s)
Microphone
Touchscreen
Accelerometer
Proximity
Ambient light
GPS
Compass
Nintendo Wii Controller

Sensors
Accelerometers
IR sensor

Microsoft Kinect

Sensors
Camera
Depth camera (Z)
Multi-Array Mic.

CS148 Lecture 5
Pat Hanrahan, Fall 2011
Things to Remember

Layered model of input

Keys and keyboards
- Keys are not characters
- D-pad and digital joysticks are just switches

Position and location
- Quadrature encoding
- Mechanical mice and trackballs
- Optical mice
- Multitouch

Emerging devices
- Many sensors: Wii, iPhone, ...