Interactive Techniques

Topics

Processing to input
- Events
- Polling

Basic principles of interactive techniques
- “Input on output”
- Direct manipulation and dragging
Events

1. May mark with time stamp
2. May store state of other devices
   ■ Add continuously changing devices like the mouse
3. May merge events for efficiency
   ■ Combine multiple mouse motion events into a single motion event
4. May reorder events based on priority
   ■ Input events from the keyboard have priority over PostRedisplay
5. May use as a general mechanism by the operating system for notifying the application
Polling

joy.cpp

Gamepads: Buttons, Hats, Axes, Balls

SONY Playstation 3

Microsoft XBOX 360
Polling vs. Events

Polling - Read state of the device directly
- Read when needed
- Read at some regular rate (e.g. 100 Hz)

Events - Notify when device state changes
- Keyboard sends “make” when the key is pressed, and “break” when the key is released

Comparison: Polling vs. Events

Polling - Read state of the device directly
- Good if the device continuously changes state
- Bad because could miss a state change
- Bad because it has high overhead

Events - Notify when a device state changes
- Efficient if the state changes are intermittent
- Need to track state; may lose track of state
- Need to select the events of interest
Graphical Interaction

Bits in input << Bits in output

20-40 words per minute

60 megapixels per sec
Bits in (sensing) > Bits out (motor)

**Big Idea**

Input on Output
Routing Input to Output

- What objects overlap the cursor?
  - Need to implement method for picking objects
  - Normally store objects in a data structure

- How to handle multiple objects?
  - The visible object (usually the last object drawn)

- How to handle object hierarchy?
  - Recursively send events from parents to children
  - Find visible object, send events to parents
DrawImage(x, y, width, height, image);

Document Object Model (DOM)
# Picking

**Simplest method**
- Find if pointer is inside the box

**Better methods**
- **Object specific pick method**: `Obj.pick(x, y)`
- **Object tag buffer**
  - Render tag (int) into a secondary framebuffer
  - Read tag at the pixel location of the mouse
- **OpenGL picking**
  - Tag objects, render all the fragments
  - Return object tag of the fragment under the cursor
Two Types of Buttons

Push button
- Press inside and release inside

Menu button
- Press inside *or outside* and release inside

Flash menu button examples.
- button.swf
- menubutton.swf
Push Button State Transitions

Product of states: Outside vs. Over / Up vs. Down
Three views of the button: Idle, Over, Down

Menu Button State Transitions

Product of states: Outside vs. Over / Up vs. Down
Three views of the button: Idle, Over, Down
Model-View-Controller Design Pattern

\[
\begin{align*}
    m &= \text{new Model();} \\
    m.&\text{addView(v1);} \\
    m.&\text{addView(v2);} \\
    v1 &= \text{new ViewA(m,c);} \\
    c &= \text{new Controller(m);} \\
    v2 &= \text{new ViewB(m,c);} 
\end{align*}
\]
v.onChange = function() {
    v.c.setValue(v.value);
}

v.onChange() = function() {
    c.setValue = function(s) {
        v.c.setValue(v.value);
        c.m.setValue(s);
        c.m.onUpdate();
    }
}

CS148 Lecture 6  Pat Hanrahan, Fall 2011
Model-View-Controller Design Pattern

m.onUpdate = function() {
    for v in m.views:
        v.onUpdate();
}

c.setValue = function(s) {
    c.m.setValue(s);
    c.m.onUpdate();
}

Model-View-Controller Design Pattern

m.onUpdate = function() {
    for v in m.views:
        v.onUpdate();
}

v.onUpdate = function() {
    m.draw(v);
}
Direct Manipulation

Dragging

this.startdrag(true);

- Attaches this to the cursor
- Position of object updated when cursor moves

stopdrag();

Flash dragging examples.
- line.swf, triangle.swf, curve.swf
Represent the world to “afford” action
Selection by pointing (not by typing)
Moving by dragging
Immediate and continuous display of the state
Indicate consequences of “dropping”
- B. Schneiderman (1983)

Things to Remember

Processing input
- Events
- Polling

Basic principles of interactive techniques
- “Input on output”
  - Document/scene model
  - Picking
  - Model-view-controller
- Direct manipulation