Multitouch-Events in OS X

Dominik Stengele
2011.05.26
2do

- Create and post Multitouch-Events in OS X

- Problem: NOT DOCUMENTED 😞
  Event delivering only roughly explained (some parts in Legacy-Documents)
  Never updated for Multitouch
How

- Analyse event delivering in OS X
- Capture Multitouch-Events from trackpad
- Disassemble captured events and analyse the structure
- Create own events
- Post created events
How events are delivered

- Trackpad
  - IOKit
  - Window Server
    - Target Process
      - Carbon Event Manager
      - NSApplication
        - NSEvent
      - Window
        - View
          - sendEvent:
            - mouseDown:
              - keyDown
            - touchesBegan:
              - ...

- Run Loop Port

- LowLevel
- CGEvent
- NSEvent

D. Stengele
2011.05.26
How to catch events

Trackpad Driver -> IOKit -> Window Server -> Target Process -> Carbon Event Manager

Event Tap

NSEvent

Why not catching NSEvent?
⇒ Only exists inside Applications
⇒ No way to publish to other Applications

D. Stengele 2011.05.26
How to catch events

// Specify event mask
CGEventMask eventMask =
    NSEventMaskGesture | NSEventMaskMagnify |
    NSEventMaskSwipe | NSEventMaskRotate |
    NSEventMaskBeginGesture | NSEventMaskEndGesture;
    // The use of these NSEvent-Flags for the mask is NOT officially supported

// Create event tap
CFMachPortRef eventTap = CGEventTapCreate(
    kCGHIDEventTap, // Event tap is placed at the point where events enter the window server
    kCGHeadInsertEventTap, // Event tap should be inserted before any pre-existing event tap
    kCGEventTapOptionDefault, // Event tap is an active filter
    eventMask, // Event mask (as specified above)
    cgEventCallbackFunction, // Callback-Function
    NULL); // User data

if (eventTap)
{
    // Create a run loop source
    CFRunLoopSourceRef runLoopSource = CFMachPortCreateRunLoopSource(kCFAllocatorDefault, eventTap, (CFIndex)0);

    // Add run loop source to current run loop
    CFRunLoopAddSource(CFRunLoopGetCurrent(), runLoopSource, kCFRunLoopCommonModes);

    // Enable event tap
    CGEventTapEnable(eventTap, true);
}

CGEventRef cgEventCallbackFunction(CGEventTapProxy proxy, CGEventType type, CGEventRef event, void* refcon);
How to analyse events

1. \[
\text{typedef \ uint32_t\ CGEventField;} \\
\text{int64_t\ CGEventGetIntegerValueField(CGEVENTRef\ event,\ CGEventField\ field);} \\
\text{double\ CGEventGetDoubleValueField(CGEVENTRef\ event,\ CGEventField\ field);} \\
\]

2. \[
\text{Flat representation without embedded pointers} \\
\Rightarrow\ \text{Contains anything we need} \\
\]

\[
\text{NSEVENT} \Rightarrow\ \text{Contains Multitouch data} \\
\]

D. Stengele 2011.05.26
Status

✓ Analyse event delivering in OS X
✓ Capture Multitouch-Events from trackpad
• Disassemble captured events and analyse the structure
➡ Create own events
➡ Post created events
How?

- Disassemble captured events and analyse the structure
  ⇒ Analyse raw data (byte buffer)
References

- Quartz Event Services Reference
- Carbon Event Manager Reference (Legacy Document)
- Cocoa Event-Handling Guide
- A lot of Headers ...
Thank you!