MPX
Multi-Pointer X
X only supports one mouse and one keyboard.
MPX fixes that.
Peter Hutterer
“whot”
peter@cs.unisa.edu.au

Wearable Computers Lab
School of CIS
University of South Australia
Adelaide, Australia
Multi-user toolkits suck.
Let’s put multiple cursors into X!
later...
Each pointer device has a cursor.
Each pointer device acts like a core pointer.
Each pointer device still acts like a XI pointer.
Each cursor can have different shapes.
Each cursor can be queried.
Each cursor can be warped.
Enter/Leave events are available for all cursors.
Each keyboard can have different focus.
Dynamic pointer-keyboard pairing.
We haven’t broken anything.
We haven’t broken anything.

Except a GUI paradigm that exists since 1965.
Differences between X and MPX
Differences between X and MPX

- Event generation & delivery.
- Mouse cursor rendering.
- Multiple cursor shape handling.
- New protocol requests/events.
- Multiple keyboard foci.
Event delivery in X:

- Driver passes data up with device.
- XI event is constructed with device.
- Core event is constructed with core pointer.
- Events are enqueued in EQ.

- Event is processed from there.
Mouse cursor rendering:

- NVIDIA GeForce 7950
  - 48 pixel pipes, 16 vertex shaders,
  - 76.8 GB per second of memory bandwidth and
  - 24 billion texels per second of fill rate on a single card
Mouse cursor rendering:

- NVIDIA GeForce 7950
  - 48 pixel pipes, 16 vertex shaders,
  - 76.8 GB per second of memory bandwidth and
  - 24 billion texels per second of fill rate on a single card
- One single mouse cursor, max size 64x64 pixels
Mouse cursor rendering:

- Switch off damage
- Restore back buffer
- Switch damage on
- Switch off damage
- Copy back buffer
- Render cursor shape
- Switch on damage

Calling multiple times segfaults.

May remove other cursors

Calling multiple times segfaults.

Calling multiple times segfaults.

Calling multiple times segfaults.

of all cursors that need to
may need to remove other cursors

of all cursors that need to
list of cursors may have changed

Calling multiple times segfaults.
Mouse cursor rendering:

- Switch off damage
- Restore back buffer
- Switch damage on
- Copy back buffer
- Render cursor shape
- Switch on damage

Calling multiple times segfaults.
May remove other cursors
Calling multiple times segfaults.

This is a pain!

Calling multiple times segfaults.
of all cursors that need to
may need to remove other cursors
of all cursors that need to
list of cursors may have changed
Calling multiple times segfaults.
API break between mipointer and misprite
  • Now passes device data across the layers.

Further API break coming soon.
  • Will improve memory usage
  • All graphics drivers will need to adjust!
Cursor shapes in X:

- The cursor can have one shape per window.
- Shapes can be inherited.
Cursor shapes in X:

- NULL
- NULL && cursorIsNone
- 0x8a234af
Cursor shapes in MPX:

- NULL
- NULL && cursorIsNone
- 0x8a234af
- device-specific 0x8a413232
- device-specific NULL
Cursor shapes in MPX:

- Cursor can have one shape per window.
- Shape can be inherited.
- Device can have one shape per window.
- Device’s cursor shape can be inherited.
- Device’s cursor shape can be inherited from cursor.
New protocol requests:

- QueryDevicePointer
- WarpDevicePointer
- DefineDeviceCursor
- ChangePointerKeyboardPairing
New protocol events:

• DeviceEnterNotify
• DeviceLeaveNotify
• (PairingChangedNotify)
Problems
Problem:

- Approx 30 calls in the core protocol do not have a defined state any more.
- Race conditions

Solution:

- For legacy apps, provide a `SetPointerBehaviour` call to change standard behaviour inside X.
  
  - `FollowSinglePointer`, `DevicePointerOnly`, `SinglePress`
Problem:

• We now rely more on window manager support.

Solution:

• Get the WM people to adopt it.
Problem:

- Writing applications is not easy anymore, the number of input devices can change at any time.

Solution

- Get toolkits to adopt it.
- Make heavy use of SetPointerBehaviour call.
Things to think of:

- Floor control (was there already)
- Relative device events
- Multi-user cut and paste
- Mouse cursor restacking
- Gesture events
It’s not ready yet.
Thanks to the X.org Foundation for sponsorship.

Thanks to Keith Packard and Daniel Stone for organising my travel & accommodation.
Peter Hutterer
peter@cs.unisa.edu.au