
Subject: capture mouse and limit all events to window
Posted by [slashupp](#) on Tue, 17 Jan 2017 08:19:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm writing a little game and for it to be functional I need full control over the mouse.
I want to capture the mouse and contain it within a window regardless of how far the mouse is moved.
e.g.: if the mouse is moved far to the right I want to show the pointer as 'stuck' within the right edge
of the window but still responding to vertical movement; all mouse-events(move, clicks, wheel, etc) must
come to my window

how can I achieve this, SetCapture() does not work for this
how/should/can I implement a MouseHook to achieve this?

Also:
how can I set the mouse-pointer to a specific point within my window?
e.g.: if want to (based on some hotkey) place the mousepointer at the center of my window

Subject: Re: capture mouse and limit all events to window
Posted by [slashupp](#) on Fri, 20 Jan 2017 07:14:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

I managed to achieve exclusive grab of mouse by modifying Ctrl:

In CtrlCore.h

Added attribute:

```
[1049] bool bcaptureLock;
```

Added parameters:

```
[1050] bool SetCapture(bool bLock=false);
```

```
[1051] bool ReleaseCapture(bool bUnlock=false);
```

In Ctrl.cpp (ctor)

```
[596] bcaptureLock=false;
```

In CtrlMouse.cpp (changed lines marked with: ///mais)

```
[301]
```

```
Image Ctrl::MEvent0(int e, Point p, int zd)
```

```
{  
    GuiLock __;  
    LLOG("MEvent0 " << Name() << " event: " << FormatIntHex(e, 0) << " point:" << p);  
    Ptr<Ctrl> _this = this;  
    mousepos = p;  
    dword mm = 0;  
    if((e & ACTION) == DOUBLE)
```

```
mm |= K_MOUSEDOUBLE;
if((e & ACTION) == TRIPLE)
mm |= K_MOUSETRIPLE;
```

```
if (bcaptureLock) return MouseEventH(e, p, zd, GetMouseFlags() | mm); ///mais...
```

```
Rect view = GetView();
if(mouseCtrl != this) {
    ...
```

```
[587]
Image Ctrl::DispatchMouseEvent(int e, Point p, int zd) {
    GuiLock __;
#ifdef defined(flagWINGL) || defined(flagLINUXGL)
    if(!IsEnabled() && this != (Ctrl*) &infoPanel)
        return Image::Arrow();
#else
    if(!IsEnabled())
        return Image::Arrow();
#endif
```

```
if(captureCtrl && captureCtrl->bcaptureLock) ///mais...
{
    if (captureCtrl!=this) return captureCtrl->DispatchMouseEvent(e, p, zd);
    return captureCtrl->MouseEvent0(e, p, zd);
}
```

```
if(captureCtrl && captureCtrl != this && captureCtrl->IsMouseActive())
    ...
```

```
[618]
bool Ctrl::SetCapture(bool bLock) { ///mais...
    GuiLock __;
    ReleaseCtrlCapture();
    if(!GetTopCtrl()->SetWndCapture()) return false;
    captureCtrl = mouseCtrl = this;
    bcaptureLock=bLock; ///mais...
    return true;
}
```

```
[628]
bool Ctrl::ReleaseCapture(bool bUnlock) { ///mais...
    GuiLock __;
    if (bcaptureLock) bcaptureLock=!bUnlock; ///mais...
    return this == captureCtrl && ReleaseCtrlCapture();
```

```
}
```

[634]

```
bool Ctrl::ReleaseCtrlCapture() {  
    GuiLock __;  
    if (captureCtrl && captureCtrl->bcaptureLock) return false; ///mais...  
    if(captureCtrl) {  
        captureCtrl->CancelMode();  
        Ctrl *w = captureCtrl->GetTopCtrl();  
        captureCtrl = NULL;  
        if(w->HasWndCapture()) {  
            w->ReleaseWndCapture();  
            return true;  
        }  
    }  
    captureCtrl = NULL;  
    return false;  
}
```

All mouse events now goes to my control exclusively (for testing I use K_ESCAPE hotkey to release)

These changes 'should' not affect any other uses of the functions. - mirek, what say you?

I'm looking at 'XWarpPointer()' to force pointer to remain physically within my control's view-area (don't know what MSWindows would require)

Adding test-code:

File Attachments

1) [main.cpp](#), downloaded 222 times

Subject: Re: capture mouse and limit all events to window

Posted by [slashupp](#) on Sun, 22 Jan 2017 12:02:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

@mirek:

I made the above changes to my copy of upp-svn10672 and done make & make install and is using the resultant theide without problem - there shouldn't be. (In these changes I've replaced the phrase 'lock' with 'exclusive')

Would it be possible to make the changes part of the official source?

Subject: Re: capture mouse and limit all events to window

Posted by [slashupp](#) on Mon, 29 Apr 2019 07:16:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

This applies to Linux-only.

To position the mouse-pointer anywhere in the window, or centered on a specific control in the window

I came up with this 'hack':

Header:

```
#ifndef _xfunc_h_
#define _xfunc_h_
```

```
#include <CtrlCore/CtrlCore.h>
```

```
namespace XFUNC
{
```

```
void set_Xmouse_pos(int x, int y);
void set_Xmouse_pos(Upp::Ctrl *pCtrl);
```

```
} //namespace XFUNC
```

```
#endif
```

Implementation:

```
#include "xfunc.h"
```

```
namespace XFUNC
{
```

```
#include <X11/Xlib.h>
```

```
void set_Xmouse_pos(int x, int y) //Ctrl *pCtrl)
{
    Upp::GuiLock __;
    static Display *disp=XOpenDisplay(0);
    if (disp)
```

```

{
    Window wroot=XRootWindow(disp, 0);
    XWarpPointer(disp, 0, wroot, 0, 0, 0, 0, x, y);
    XFlush(disp);
}
}

```

```

void set_Xmouse_pos(Upp::Ctrl *pCtrl)
{
    if (!pCtrl) return;
    Upp::GuiLock __;
    int x, y;
    Upp::Rect r=pCtrl->GetScreenRect();
    x=(r.left+r.Width()/2);
    y=(r.top+r.Height()/2);
    static Display *disp=XOpenDisplay(0);
    if (disp)
    {
        Window wroot=XRootWindow(disp, 0);
        XWarpPointer(disp, 0, wroot, 0, 0, 0, 0, x, y);
        XFlush(disp);
    }
}

```

```

} //namespace XFUNC

```

Hope this helps someone

Subject: Re: capture mouse and limit all events to window

Posted by [mirek](#) on Thu, 04 Jul 2019 09:06:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

This should not be mixed with "capture" concept. It is already hard to make capture consistent across platforms (win/x11/macos) and this would add another level of complication.

What we need instead IMO is to encapsulate platform specific apis like warp.

Mirek
