
Subject: RectTracker & MaxRect

Posted by [Dochy](#) on Thu, 04 Sep 2008 19:18:05 GMT

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

Hi,

I have problem with RectTracker. I used RectTracker with ALIGN_NULL for user-specific begin of drawing rect. I want to mark part of drawn image. MaxRect is good for limit to bottom and to right, but top and left coordinates aren't limited.

So in view with borders (0,0,10,10) i can't get rect (5,5,20,20) but only (5,5,10,10) - I expected it and want it

But I can get (-10,-10,5,5) - if (5,5) is origin - and I don't want this .

Am I wrong or there is error?

Thanks

I naturally can check and handle it, but I think that there must be better way

```
void LeftDown(Point p, dword keyflags){
    // Vyrez z obrazku
    RectTracker tr(*this);
    Size sz = GetSize();
    tr.Dashed().Animation(40).MaxRect(RectC(0,0,GetSize().cx,GetSize().cy));
    vyrez=RectC(p.x,p.y,0,0);
    vyrez=tr.Track(vyrez,ALIGN_NULL,ALIGN_NULL);
    Refresh();
};
```

Edit:

I made modified copy of RectTracker class, that correspond to my need now. But I think that this class (RectTracker) should be modified or there can be made a new class for clearly defined pixel based operations. Maybe someday I try it. Now I'm beginner only

Subject: Re: RectTracker & MaxRect

Posted by [cbpporter](#) on Sat, 06 Sep 2008 19:09:12 GMT

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

Try using a different align. ALIGN_NULL means you have full freedom to drag in any direction.

Subject: Re: RectTracker & MaxRect

Posted by [Dochy](#) on Sun, 07 Sep 2008 09:42:16 GMT

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

Quote:Try using a different align. ALIGN_NULL means you have full freedom to drag in any

direction.

I know. But it is what I want. Anywhere in view area begin and anywhere end. Problem is only, that I want restricted position of end to border of view, but now there is restriction to right and bottom only. I solved it by simply modification of RectTracker::MouseMove

```
if(tx == ALIGN_NULL) {  
    rect.right = min(org.right - op.x + p.x, maxrect.right);  
    if (rect.right < rect.left) {  
        Swap(rect.right, rect.left);  
        rect.InflateHorz(1);  
    }  
}
```

to:

```
if(tx == ALIGN_NULL) {  
    pom.x = min(org.right - op.x + p.x, maxrect.right);  
    rect.right = max(pom.x, maxrect.left);  
    if (rect.right < rect.left) {  
        Swap(rect.right, rect.left);  
    }  
}
```

But it is not clear enough. There are variations of +-1 pixel depending of draw direction. Now it's adequate for me, but not as final solution.

Thanks for interest.
