Subject: RectTracker filled with black
Posted by cbpporter on Wed, 13 Feb 2008 05:54:18 GMT
View Forum Message <> Reply to Message

Hi!

I've been using RectTracker for a while now and I'm quite happy with it because it removes the need for manual mouse move and up handling, the actual drawing, managing capture, etc. But the limitation of the resulting rect to a specified quadrant is no longer suited to my need, so I've added 4 little lines to allow the use of ALIGN_NULL for tx and ty parameters in RectTracker::TrackRect to mean that the resulting rect is freed from constrains, even though by it's definition, it could possibly be "empty".

The problem is that if it is outside one given quadrant, the drawn rect becomes filled with black, as opposed to just an outline. Does anybody have any idea why this happens?

Here is my moddified RectTracker:

```
void RectTracker::MouseMove(Point, dword)
{
 Point p = GetMousePos();
 rect = org;
 if(tx == ALIGN_CENTER && ty == ALIGN_CENTER) {
  int x = org.left - op.x + p.x;
  int y = org.top - op.y + p.y;
  if(x + org.Width() > maxrect.right)
   x = maxrect.right - org.Width();
  if(x < maxrect.left)
   x = maxrect.left;
  if(y + org.Height() > maxrect.bottom)
   y = maxrect.bottom - org.Height();
  if(y < maxrect.top)
   y = maxrect.top;
  rect = RectC(x, y, org.Width(), org.Height());
 }
 else {
  if(tx == ALIGN_LEFT) {
   rect.left = max(org.left - op.x + p.x, maxrect.left);
   rect.left = minmax(rect.left, rect.right - maxsize.cx, rect.right - minsize.cx);
  }
  if(tx == ALIGN_RIGHT) {
   rect.right = min(org.right - op.x + p.x, maxrect.right);
   rect.right = minmax(rect.right, rect.left + minsize.cx, rect.left + maxsize.cx);
  }
  if(ty == ALIGN_TOP) {
   rect.top = max(org.top - op.y + p.y, maxrect.top);
   rect.top = minmax(rect.top, rect.bottom - maxsize.cy, rect.bottom - minsize.cy);
  }
```

```
  if(ty == ALIGN_BOTTOM) {
   rect.bottom = min(org.bottom - op.y + p.y, maxrect.bottom);
   rect.bottom = minmax(rect.bottom, rect.top + minsize.cy, rect.top + maxsize.cy);
  }
  if(tx == ALIGN_NULL)
   rect.right = min(org.right - op.x + p.x, maxrect.right);
  if(ty == ALIGN_NULL)
   rect.bottom = min(org.bottom - op.y + p.y, maxrect.bottom);
  if(keepratio) {
   int cy = org.Width() ? rect.Width() * org.Height() / org.Width() : 0;
   int cx = org.Height() ? rect.Height() * org.Width() / org.Height() : 0;
   if(tx == ALIGN_BOTTOM && ty == ALIGN_RIGHT) {
    Size sz = rect.Size();
    if(cx > sz.cx)
     rect.right = rect.left + cx;
    else
     rect.bottom = rect.top + cy;
   }
   else
   if(tx == ALIGN_RIGHT)
    rect.bottom = rect.top + cy;
   else
   if(ty == ALIGN_BOTTOM)
    rect.right = rect.left + cx;
  }
 }
 if(rect != o) {
  rect = Round(rect);
  if(rect != o) {
   DrawRect(o, rect);
   sync(rect);
   o = rect;
  }
 }
}
```

Subject: Re: RectTracker filled with black
Posted by mirek on Wed, 13 Feb 2008 07:44:19 GMT
View Forum Message <> Reply to Message

IMO a good idea. Do you think you can post the code to test it? (A "testcase"

Sure, it is perhaps simple, but I might understand something wrong...

Mirek

Subject: Re: RectTracker filled with black
Posted by mrjt on Wed, 13 Feb 2008 11:13:15 GMT
View Forum Message <> Reply to Message

DrawDragRect doesn't work properly with backwards rectangles.

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

KeepRatio won't work properly, but since it only works correctly with ALIGN_RIGHT/ALIGN_BOTTOM anyway I don't see this as a major problem.

For a test case just change the default behaviour in RectTracker example (ie no control keys) to ALIGN_NULL/ALIGN_NULL.

I love looking through Upp source, there is always something new and cool to find . For instance, what is the purpose of ViewDraw? Could I use it to do drawing over a TopWindow full of controls? I've been looking for a way to do that for a while.

Subject: Re: RectTracker filled with black
Posted by cbpporter on Wed, 13 Feb 2008 11:40:58 GMT
View Forum Message <> Reply to Message

Quote:DrawDragRect doesn't work properly with backwards rectangles.
This seems to be the problem. Works fine with the swapped coords. I'm a little worried about the precision of the end-point coordinate, which seems of by a pixel in both dimensions (seen more clearly when Image::Cross() cursor is used, and end-point is mouse coord), but I can't tell for sure. U++ Rect is extremely counter intuitive for me and I have to battle years of experience with completely different rectangle semantics, so until I get used to this, it is really hard to make sense of code which uses rect.

Quote:For a test case just change the default behaviour in RectTracker example (ie no control keys) to ALIGN_NULL/ALIGN_NULL.
Forgot about that one. At least I don't have to write a testcase .

Quote:For instance, what is the purpose of ViewDraw?
I found it this morning before work too. I didn't look into it because I was already having a headache with all those rects, but you can find a lot of such classes in U++. I don't know if even Mirek remembers them all?

Subject: Re: RectTracker filled with black
Posted by cbpporter on Wed, 13 Feb 2008 15:12:26 GMT
View Forum Message <> Reply to Message

I found a subtle lack of functionality from fixing the black rect. By swapping the coordinates, you are basically mirroring your rect so that it winds up in the forth quadrant and get the coordinates as if you would translate that rect to a different position. This is Ok if you want just to draw it, but there is no way to determine the relationship of you initial mouse position and the final one. Since Track returns a rect, I want to be able to determine the relative position of the coordinates only with the information in the rect, so it must return values where left > right or top > bottom. This can easily be fixed by moving the extra check and swap from MouseMove to DrawRect.

There is another fix that must be done. The relationship between the edge of the rect and the mouse position is not uniform as you drag across quadrants, both in the modified version and in the original one. I can fix this too easily, but I need to know how Mirek wants it to behave.

Q4: Rect edge is one pixel distance from mouse on both dimensions
Q2: Rect edge is identical to mouse
Q1&2: Rect edge is one pixel distance from mouse only on one dimension

What should the convention be?

Subject: Re: RectTracker filled with black
Posted by cbpporter on Wed, 20 Feb 2008 08:33:57 GMT
View Forum Message <> Reply to Message

Since nobody commented on the coordinate relation between mouse cursor position/rect coordinates and the way the drawn rect visually "touches" the cursor (and also KeepRation(true) seems to add another difference of 1 pixel to the extent of the visual rect), I guess it's safe to assume that this is not an issue for anybody else and it will remain as is.

Then here are the proposed changed to RectTracker, so that ALIGN_NULL will work:

```
void RectTracker::MouseMove(Point, dword)
{
 Point p = GetMousePos();
 rect = org;
 if(tx == ALIGN_CENTER && ty == ALIGN_CENTER) {
  int x = org.left - op.x + p.x;
  int y = org.top - op.y + p.y;
  if(x + org.Width() > maxrect.right)
   x = maxrect.right - org.Width();
  if(x < maxrect.left)
   x = maxrect.left;
  if(y + org.Height() > maxrect.bottom)
   y = maxrect.bottom - org.Height();
  if(y < maxrect.top)
```

```
  y = maxrect.top;
 rect = RectC(x, y, org.Width(), org.Height());
}
else {
 if(tx == ALIGN_LEFT) {
  rect.left = max(org.left - op.x + p.x, maxrect.left);
  rect.left = minmax(rect.left, rect.right - maxsize.cx, rect.right - minsize.cx);
 }
 if(tx == ALIGN_RIGHT) {
  rect.right = min(org.right - op.x + p.x, maxrect.right);
  rect.right = minmax(rect.right, rect.left + minsize.cx, rect.left + maxsize.cx);
 }
 if(ty == ALIGN_TOP) {
  rect.top = max(org.top - op.y + p.y, maxrect.top);
  rect.top = minmax(rect.top, rect.bottom - maxsize.cy, rect.bottom - minsize.cy);
 }
 if(ty == ALIGN_BOTTOM) {
  rect.bottom = min(org.bottom - op.y + p.y, maxrect.bottom);
  rect.bottom = minmax(rect.bottom, rect.top + minsize.cy, rect.top + maxsize.cy);
 }
 if(tx == ALIGN_NULL) {
  rect.right = min(org.right - op.x + p.x, maxrect.right);
 }
 if(ty == ALIGN_NULL) {
  rect.bottom = min(org.bottom - op.y + p.y, maxrect.bottom);
 }
 if(keepratio) {
  int cy = org.Width() ? rect.Width() * org.Height() / org.Width() : 0;
  int cx = org.Height() ? rect.Height() * org.Width() / org.Height() : 0;
  if(tx == ALIGN_BOTTOM && ty == ALIGN_RIGHT) {
   Size sz = rect.Size();
   if(cx > sz.cx)
    rect.right = rect.left + cx;
   else
    rect.bottom = rect.top + cy;
  }
  else
  if(tx == ALIGN_RIGHT)
   rect.bottom = rect.top + cy;
  else
  if(ty == ALIGN_BOTTOM)
   rect.right = rect.left + cx;
 }
}
if(rect != o) {
 rect = Round(rect);
 if(rect != o) {
  DrawRect(o, rect);
```

```
   sync(rect);
   o = rect;
  }
 }
}


void RectTracker::DrawRect(Rect r1, Rect r2)
{
 if(ty < 0) {
  r1.left = r1.right - 1;
  r2.left = r2.right - 1;
 }
 if(tx < 0) {
  r1.top = r1.bottom - 1;
  r2.top = r2.bottom - 1;
 }
 Rect c = clip & GetMaster().GetSize();
 if (r1.right < r1.left) Swap(r1.left, r1.right);
 if (r1.bottom < r1.top) Swap(r1.top, r1.bottom);
 if (r2.right < r2.left) Swap(r2.left, r2.right);
 if (r2.bottom < r2.top) Swap(r2.top, r2.bottom);
 if(animation) {
  int nanim = (GetTickCount() / animation) % 8;
  DrawDragRect(GetMaster(), Rect(0, 0, 0, 0), r2, c, width, color, sGetAniPat(pattern, nanim));
  DrawDragRect(GetMaster(), r1, Rect(0, 0, 0, 0), c, width, color, sGetAniPat(pattern, panim));
  panim = nanim;
 }
 else
  DrawDragRect(GetMaster(), r1, r2, c, width, color, pattern);
}
```

Subject: Re: RectTracker filled with black
Posted by mrjt on Wed, 20 Feb 2008 09:55:49 GMT

View Forum Message <> Reply to Message

This would be my suggested version:

```
void RectTracker::MouseMove(Point, dword)
{
 Point p = GetMousePos();
 rect = org;
 if(tx == ALIGN_CENTER && ty == ALIGN_CENTER) {
  int x = org.left - op.x + p.x;
  int y = org.top - op.y + p.y;
  if(x + org.Width() > maxrect.right)
   x = maxrect.right - org.Width();
```

```
 if(x < maxrect.left)
  x = maxrect.left;
 if(y + org.Height() > maxrect.bottom)
  y = maxrect.bottom - org.Height();
 if(y < maxrect.top)
  y = maxrect.top;
 rect = RectC(x, y, org.Width(), org.Height());
}
else {
 if(tx == ALIGN_LEFT) {
  rect.left = max(org.left - op.x + p.x, maxrect.left);
  rect.left = minmax(rect.left, rect.right - maxsize.cx, rect.right - minsize.cx);
 }
 if(tx == ALIGN_RIGHT) {
  rect.right = min(org.right - op.x + p.x, maxrect.right);
  rect.right = minmax(rect.right, rect.left + minsize.cx, rect.left + maxsize.cx);
 }
 if(ty == ALIGN_TOP) {
  rect.top = max(org.top - op.y + p.y, maxrect.top);
  rect.top = minmax(rect.top, rect.bottom - maxsize.cy, rect.bottom - minsize.cy);
 }
 if(ty == ALIGN_BOTTOM) {
  rect.bottom = min(org.bottom - op.y + p.y, maxrect.bottom);
  rect.bottom = minmax(rect.bottom, rect.top + minsize.cy, rect.top + maxsize.cy);
 }
 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);
  }
 }
 if(ty == ALIGN_NULL) {
  rect.bottom = min(org.bottom - op.y + p.y, maxrect.bottom);
  if (rect.bottom < rect.top) {
   Swap(rect.bottom, rect.top);
   rect.InflateVert(1);
  }
 }
 if(keepratio) {
  int cy = org.Width() ? rect.Width() * org.Height() / org.Width() : 0;
  int cx = org.Height() ? rect.Height() * org.Width() / org.Height() : 0;
  if(tx == ALIGN_BOTTOM && ty == ALIGN_RIGHT) {
   Size sz = rect.Size();
   if(cx > sz.cx)
    rect.right = rect.left + cx;
   else
    rect.bottom = rect.top + cy;
```

```
   }
   else
   if(tx == ALIGN_RIGHT)
    rect.bottom = rect.top + cy;
   else
   if(ty == ALIGN_BOTTOM)
    rect.right = rect.left + cx;
  }
 }
 if(rect != o) {
  rect = Round(rect);
  if(rect != o) {
   DrawRect(o, rect);
   sync(rect);
   o = rect;
  }
 }
}
```

It fixes the cursor alignment problem. I also believe that this is the correct place to modify the coordinates, as doing so before drawing means that the final rectangle is inaccurate.

IMO if you really need to have the final rectangle backwards I think you should compare and switch it after getting the final rect back.

Subject: Re: RectTracker filled with black
Posted by cbpporter on Wed, 20 Feb 2008 12:28:46 GMT
View Forum Message <> Reply to Message

Quote:IMO if you really need to have the final rectangle backwards I think you should compare and switch it after getting the final rect back.
Well I don't know how one could achieve that, since the only info you get is the final rect and the stating point, so not enough to determine if it is backwards or not.

Quote:It fixes the cursor alignment problem. I also believe that this is the correct place to modify the coordinates, as doing so before drawing means that the final rectangle is inaccurate.
The rect's I'm modifying in drawrect are not the original, so I don't think this affect the result in any way. I've written a test program, and the coordinates always seem to be accurate regarding the mouse position.

Subject: Re: RectTracker filled with black
Posted by mirek on Wed, 20 Feb 2008 12:57:11 GMT
View Forum Message <> Reply to Message

mrjt wrote on Wed, 20 February 2008 04:55This would be my suggested version:

Applied. Thanks.

Mirek

Subject: Re: RectTracker filled with black
Posted by mrjt on Wed, 20 Feb 2008 13:13:10 GMT
View Forum Message <> Reply to Message

Quote:Well I don't know how one could achieve that, since the only info you get is the final rect and the stating point, so not enough to determine if it is backwards or not.
This should do the trick:
Rect original = rect;
rect = tr.Track(original, ALIGN_NULL, ALIGN_NULL);
if (rect.left < original.left) Swap(rect.left, rect.right);
if (rect.top < original.top) Swap(rect.top, rect.bottom);

Subject: Re: RectTracker filled with black
Posted by cbpporter on Wed, 20 Feb 2008 13:19:22 GMT
View Forum Message <> Reply to Message

Rect original = rect;
And what value would this original rect have? I usually set it up as Rect(mousex, mousey, mousex, mousey), and any other value I used ends up extending the rectangle in a way I don't want to.

Subject: Re: RectTracker filled with black
Posted by mrjt on Wed, 20 Feb 2008 13:52:52 GMT
View Forum Message <> Reply to Message

original = Rect(mousex, mousey, mousex, mousey)

Subject: Re: RectTracker filled with black
Posted by cbpporter on Wed, 20 Feb 2008 13:58:33 GMT
View Forum Message <> Reply to Message

mrjt wrote on Wed, 20 February 2008 15:52original = Rect(mousex, mousey, mousex, mousey)
You're right! It was under my nose all this time . Anyway, I don't really care which one will be included, I can manage with both. But I need to know, so that I can use it an focus on more important parts (like SVG export).

Subject: Re: RectTracker filled with black
Posted by cbpporter on Sun, 24 Feb 2008 21:21:49 GMT
View Forum Message <> Reply to Message

Mirek, could you please tell me which version will you include in U++ sources. Tomorrow in Monday, a brand new week and I would like to start by getting rid with all the silly coordinate messing and do something more serious .

Subject: Re: RectTracker filled with black
Posted by mirek on Tue, 26 Feb 2008 20:13:03 GMT
View Forum Message <> Reply to Message

well, the recent one long listing

Wed, 20 February 2008 04:55
mrjt

Unless told otherwise.

Mirek

Subject: Re: RectTracker filled with black
Posted by cbpporter on Thu, 28 Feb 2008 10:12:32 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 26 February 2008 22:13well, the recent one long listing

Wed, 20 February 2008 04:55
mrjt

Unless told otherwise.

Mirek
OK, I'm giving up on RectTracker and doing it manually with mouse events and GetCapture(). I just can't get it to behave as as want it.

Subject: Re: RectTracker filled with black
Posted by cbpporter on Fri, 29 Feb 2008 14:08:56 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Thu, 28 February 2008 12:12
OK, I'm giving up on RectTracker and doing it manually with mouse events and GetCapture(). I just can't get it to behave as as want it.

OK, I seems that U++ is having an undesired effect on me. I can no longer choose the easy way so easily . I was writing my manual code, and even though it was easy and worked fine, I felt it as highly redundant since there was already that RectTracker class.
So I sat down and managed to adapt it to my needs. I'm not posting the result, because I think it is not that general, it is more suited for graphics applications with precise cursor interaction, and also you have to adjust the coordinates returned by Track a little.

But there are still a couple of thing I would like to note:
1. I used ViewDraw and strangely, the drawings remain on my form even after it got covered by other windows. Does ViewDraw feature any cache or backbuffer?

2. In the code proposed by mrjt, I have found that the InflateVert(1) and InflateHorz(1) introduce a 1 pixel imprecision between mouse cursor, rect coordinates and visual representation of the drawn drag rect.

I have added include the test case I use to test coordinates.

File Attachments

1) RectTest.rar, downloaded 429 times

Subject: Re: RectTracker filled with black
Posted by mrjt on Fri, 29 Feb 2008 14:20:31 GMT
View Forum Message <> Reply to Message

I'm aware of the imprecision, but I wasn't sure what to do about it. It's linked to how rectangles are drawn, ie. inclusive top-left-coords, exclusive bottom-right coords.

Changing it either way is wrong, so I chose the way that means if you draw a rect of the coords it perfectly matches the (visible) cursor position.

James

Subject: Re: RectTracker filled with black
Posted by cbpporter on Fri, 29 Feb 2008 14:34:28 GMT
View Forum Message <> Reply to Message

mrjt wrote on Fri, 29 February 2008 16:20I'm aware of the imprecision, but I wasn't sure what to do about it. It's linked to how rectangles are drawn, ie. inclusive top-left-coords, exclusive bottom-right coords.

Changing it either way is wrong, so I chose the way that means if you draw a rect of the coords it perfectly matches the (visible) cursor position.

James

Yes, that makes sense in these conditions and I guess is OK with U++ rects. I'll inherit my class from RectTracker so that I can use the trackers like in the atachment.

File Attachments
1) RectTest.exe, downloaded 445 times

Subject: Re: RectTracker filled with black
Posted by mirek on Mon, 03 Mar 2008 19:48:54 GMT
View Forum Message <> Reply to Message

I was thinking about this issue a bit.

Would not it be better instead of ALIGN_NULL to allow negative "MinSize" ?

Mirek

Subject: Re: RectTracker filled with black
Posted by cbpporter on Tue, 04 Mar 2008 09:24:25 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 03 March 2008 21:48I was thinking about this issue a bit.

Would not it be better instead of ALIGN_NULL to allow negative "MinSize" ?

Mirek
Well I chose ALIGN_NULL because I was trying to keep it in line with the rest of the API and make it as unintrusive as possible.

But anything will do. I just want it to work right. Right now I'm using a slightly "hacked" version of RectTracker, which might become a private ShapeTracker before the next release, because I don't want to have to change things every time I download a new U++.

This class has to cover the following points:
1. Allow full mouse movement
2. Be uniform in it's interpretation of coordinates. Current rect tracker doesn't seem to be.
3. Be precise. Again, not the case with current. An experienced user will get annoyed very quickly when they have to drag the mouse one extra pixel depending on the quadrant to align some stuff.

And in the future, I'll need it to track arrows, so that you can visually design relationships between objects. Nothing fancy, just a line with a small arrow head.

Subject: Re: RectTracker filled with black
Posted by cbpporter on Thu, 20 Mar 2008 13:35:14 GMT

View Forum Message <> Reply to Message

Would it be possible to remove the static from uint64 sGetAniPat(uint64 src, int pos) in localloop.cpp.

And this is the wrong place to ask, but how do I get the real name of the StdFont()?

Subject: Re: RectTracker filled with black
Posted by cbpporter on Thu, 20 Mar 2008 14:31:08 GMT
View Forum Message <> Reply to Message

And another question: Why is void DrawDragRect(Draw& w, const Rect& rect1, const Rect& rect2, const Rect& clip, int n,
Color color, uint64 pattern) so complicated . Couldn't one use just simple DrawRects with NOTXOR drawing mode? I'm asking because I'm implementing custom dragable shapes.

Subject: Re: RectTracker filled with black
Posted by mirek on Sat, 22 Mar 2008 19:33:36 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Thu, 20 March 2008 10:31And another question: Why is void DrawDragRect(Draw& w, const Rect& rect1, const Rect& rect2, const Rect& clip, int n,
Color color, uint64 pattern) so complicated . Couldn't one use just simple DrawRects with NOTXOR drawing mode? I'm asking because I'm implementing custom dragable shapes.

Because of flickering.

Mirek