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;
  }
 }
}
```