Subject: Re: How to implent a rubber band Class in u++ Posted by tojocky on Sat, 09 Aug 2008 16:43:58 GMT View Forum Message <> Reply to Message

mrjt wrote on Wed, 06 August 2008 12:04I can think of two possible optimisations (besides PolyLine, which is definitely the first one to use)

1- Avoid adding adjacent collinear points. In practice this is a very minor optimisation except for special cases, but it appeals to my sense of correctness.

2- Use a back buffer. Providing you have a static background (one that will not change while the rubber band is being drawn), you can use a back buffer to cache the background + the current rubber band and only draw the last line.

This was a major improvement for me, eliminating flickering entirely.

•••••

Very Hard optimization. Thank you!

Bu in this example is an error!

```
void RubberBand::AddPoint(const Point &newp)
```

```
{
  if (points.GetCount() < 2)
  return points.Add(newp);
  // Get line vectors
  const Point &p = points[points.GetCount()-2];
  Point p1 = points.Top() - p;
  Point p2 = newp - p;
  // If gradient is different, add the new point
  if (p1.x*p2.y - p2.x*p1.y)
   points.Add(newp);
  // Otherwise update last points
  else
  ==>HERE points.Top() = newp;
}
```

The bug is when the newpoint is in the line draws from the last 2 points AND the newpoint is in between from the last 2 points! Corrected code is:

```
void RubberBandClass::AddPoint(const Point &newp) {
    if (points.GetCount() < 2)
        return points.Add(newp);
    // Get line vectors
        const Point &p_second_last = points[points.GetCount()-2];</pre>
```

```
const Point &p_last = points.Top();
Point p1 = p_last - p_second_last;
Point p2 = newp - p_second_last;
// If gradient is different, add the new point
if (p1.x*p2.y - p2.x*p1.y)
 points.Add(newp);
// Otherwise update last points if the newpoint is continue of the last point from the second last
point
else{
 int pos_from_last = 0;
 int pos_1=newp.x, pos_2=p_last.x, pos_3=p_second_last.x;
 if (!((p_last.x == p_second_last.x)\&(newp.x == p_last.x)))
 int pos_1=newp.y, pos_2=p_last.y, pos_3=p_second_last.y;
 if (pos_2 > pos_3)
 if (pos_1 > pos_2) {
  pos_from_last = 1;
 }
 else if (pos 1 < pos 3) {
  pos_from_last = 2;
 }
 }
 else {
 if (pos_1 < pos_3) {
  pos_from_last = 1;
 }
 else if (pos_1 >pos_2){
  pos from last = 2;
 }
 }
 if (pos_from_last) points[points.GetCount()-pos_from_last] = newp;
}
}
```

In base of this example I can write paint method of controls more optimized!

Thanks a lot about back buffer (BackDraw)!

Is other ideas?