Subject: How to implent a rubber band Class in u++
Posted by westcity1973 on Thu, 25 Oct 2007 12:01:55 GMT
View Forum Message <> Reply to Message

I am now studying CAD. And I need a class of rubber band like AutoCAD. RectTracker is good, but I need a tracker returning an array of points. I try to implent it in Scribble, but it is not very good. Is there any simple method to implement it.

I attach the my modified scribble code .

File Attachments
1) scribble.rar, downloaded 469 times

Subject: Re: How to implent a rubber band Class in u++
Posted by mrjt on Thu, 25 Oct 2007 14:44:30 GMT
View Forum Message <> Reply to Message

Easy

RubberBand class:

```
class RubberBand : public LocalLoop
{
public:
 virtual void MouseMove(Point p, dword keyflags)  { points.Add(p); GetMaster().Refresh(); }
 virtual void LeftUp(Point p, dword keyflags)     { EndLoop(); }
 virtual void RightUp(Point p, dword keyflags)    { EndLoop(); }

 const Vector<Point> & GetPoints()     { return points; }
 void      Clear()      { points.Clear(); }
private:
 Vector<Point> points;
};
```

Test code (band is member variable of type RubberBand):

```
void AWindow::LeftDown(Point p, dword keyflags)
{
 band.Clear();
 band.SetMaster(*this);
 band.Run();
 Refresh();
}

void AWindow::Paint(Draw& w)
{
 const Vector<Point> &p = band.GetPoints();

 w.DrawRect(GetSize(), SColorFace);
```

```
 for (int i = 0; i < p.GetCount()-1; i++)
  w.DrawLine(p[i], p[i+1]);
 if (!band.InLoop() && p.GetCount() > 1)
  w.DrawLine(p[p.GetCount()-1], p[0]);
 w.DrawText(4, 4, AsString(p.GetCount()));
}
```
Hope that helps.

Btw, this is in the wrong forum. General widget forum would have been better.

Subject: Re: How to implent a rubber band Class in u++
Posted by westcity1973 on Fri, 26 Oct 2007 02:49:36 GMT
View Forum Message <> Reply to Message

Thanks, it is very helpful.

Subject: Re: How to implent a rubber band Class in u++
Posted by tojocky on Mon, 07 Jul 2008 16:35:53 GMT
View Forum Message <> Reply to Message

Good example!

But how about to optimize this? When I move mouse and it is in loop, need to ADD only the last line draw but do not repaint all? This situation is when change form sizes too! I thing that a way is to set data in a Draw and every time when calls paint return from draw! Is the standard method about this?

Thank you!

Subject: Re: How to implent a rubber band Class in u++
Posted by tojocky on Fri, 25 Jul 2008 18:02:32 GMT
View Forum Message <> Reply to Message

How can i copy a draw data to another draw data with position?

for example
Thank you!

Subject: Re: How to implent a rubber band Class in u++
Posted by mirek on Sat, 26 Jul 2008 10:03:15 GMT
View Forum Message <> Reply to Message

tojocky wrote on Fri, 25 July 2008 14:02How can i copy a draw data to another draw data with position?

for example
Thank you!

What is "draw data"?

Subject: Re: How to implent a rubber band Class in u++
Posted by tojocky on Sat, 26 Jul 2008 14:18:55 GMT
View Forum Message <> Reply to Message

luzr wrote on Sat, 26 July 2008 13:03tojocky wrote on Fri, 25 July 2008 14:02How can i copy a draw data to another draw data with position?

for example
Thank you!

What is "draw data"?
Sorry for stupid question! In the documentation is this mechanism! I want to save in the variabile image data and on call Paint method to Draw saved Image! ImageMaker is perfect for me!

Subject: Re: How to implent a rubber band Class in u++
Posted by mirek on Sat, 26 Jul 2008 14:49:40 GMT
View Forum Message <> Reply to Message

Are we speaking about "Drawing"?

Mirek

Subject: Re: How to implent a rubber band Class in u++
Posted by tojocky on Sat, 26 Jul 2008 20:45:35 GMT
View Forum Message <> Reply to Message

luzr wrote on Sat, 26 July 2008 17:49Are we speaking about "Drawing"?

Mirek

From Manual point number 6: Image cache

Subject: Re: How to implent a rubber band Class in u++
Posted by tojocky on Sat, 26 Jul 2008 23:02:23 GMT

View Forum Message <> Reply to Message

I tried to optimize this project, but i made this project more slowly and somewhere i have memory leak.
I added in RubberBandClass parameter

Quote:Drawing resultpaint;
I Tried to buffering image on MouseMove and paste buffered image in method Paint method.

I attached the project!

Help!

File Attachments
1) RubberBandFast.zip, downloaded 373 times

Subject: Re: How to implent a rubber band Class in u++
Posted by mirek on Fri, 01 Aug 2008 06:53:14 GMT
View Forum Message <> Reply to Message

tojocky wrote on Sat, 26 July 2008 19:02I tried to optimize this project, but i made this project more slowly and somewhere i have memory leak.
I added in RubberBandClass parameter

Quote:Drawing resultpaint;
I Tried to buffering image on MouseMove and paste buffered image in method Paint method.

I attached the project!

Help!

After fixing apparent bugs, I see no leak.

Anyway, Drawing will not speed this up. It would be more efficient to just add some "DrawRubber" method to your RubberBandClass.

Mirek

Subject: Re: How to implent a rubber band Class in u++
Posted by tojocky on Fri, 01 Aug 2008 20:33:48 GMT
View Forum Message <> Reply to Message

But i see that virtual memory increasing in example wich i modified... but in the preview original example the memory is constant!
Sorry for spent your time! but can you show me in this example?

Thanks!

Subject: Re: How to implent a rubber band Class in u++
Posted by tojocky on Sat, 02 Aug 2008 07:49:27 GMT
View Forum Message <> Reply to Message

I found corrected project in uppdev repository. but virtual memory increasing anyway and speed is slowly!
I uploaded the original project... the virtual memory is constant and faster.
Is the Method DrawingDraw so slowly?
Thanks!

File Attachments
1) RubberBand.7z, downloaded 364 times

Subject: Re: How to implent a rubber band Class in u++
Posted by tojocky on Mon, 04 Aug 2008 16:56:06 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 01 August 2008 09:53tojocky wrote on Sat, 26 July 2008 19:02I tried to optimize this project, but i made this project more slowly and somewhere i have memory leak.
I added in RubberBandClass parameter

Quote:Drawing resultpaint;
I Tried to buffering image on MouseMove and paste buffered image in method Paint method.

I attached the project!

Help!

After fixing apparent bugs, I see no leak.

Anyway, Drawing will not speed this up. It would be more efficient to just add some "DrawRubber" method to your RubberBandClass.

Mirek
How can I speed up if I will have a lot of points?

Subject: Re: How to implent a rubber band Class in u++
Posted by mirek on Mon, 04 Aug 2008 17:39:24 GMT
View Forum Message <> Reply to Message

tojocky wrote on Mon, 04 August 2008 12:56luzr wrote on Fri, 01 August 2008 09:53tojocky wrote on Sat, 26 July 2008 19:02I tried to optimize this project, but i made this project more slowly and

somewhere i have memory leak.
I added in RubberBandClass parameter

Quote:Drawing resultpaint;
I Tried to buffering image on MouseMove and paste buffered image in method Paint method.

I attached the project!

Help!

After fixing apparent bugs, I see no leak.

Anyway, Drawing will not speed this up. It would be more efficient to just add some "DrawRubber" method to your RubberBandClass.

Mirek
How can I speed up if I will have a lot of points?

Let us put it into perspective.

How much points? How slow? How much virtual memory? What computer?

I had patience to create 3000 points band, noticing NO slowdown, VM at 6MB (normal).

(But I must admit I have top-end rig now).

Mirek

Subject: Re: How to implent a rubber band Class in u++
Posted by tojocky on Mon, 04 Aug 2008 20:33:58 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 04 August 2008 20:39

Let us put it into perspective.

How much points? How slow? How much virtual memory? What computer?

I had patience to create 3000 points band, noticing NO slowdown, VM at 6MB (normal).

(But I must admit I have top-end rig now).

Mirek

Sorry,
OS: Windows XP SP3;
Compiler: MSC8 Debug

UPP Version: SVN 318

1. RubberBand, 5000point: The virtual memory increase from 6.8Mb to 7Mb
2. RubberBandFast, 1500 points: The virtual memory increase from 6.8 to 85Mb

The RubberBandFast is more slowly!

Subject: Re: How to implent a rubber band Class in u++
Posted by mirek on Tue, 05 Aug 2008 13:44:47 GMT
View Forum Message <> Reply to Message

The problem is

```
iw.DrawDrawing(0, 0, old_size.cx, old_size.cy, resultpaint);
```

creates very deep recursion in Drawing definition (other drawing within drawin is stored as operation). That results in slow speed and huge VM.

Mirek

Subject: Re: How to implent a rubber band Class in u++
Posted by tojocky on Tue, 05 Aug 2008 15:37:51 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 05 August 2008 16:44The problem is

```
iw.DrawDrawing(0, 0, old_size.cx, old_size.cy, resultpaint);
```

creates very deep recursion in Drawing definition (other drawing within drawin is stored as operation). That results in slow speed and huge VM.

Mirek

What another method will be more faster than DrawDrawing? May be DrawImage?

Subject: Re: How to implent a rubber band Class in u++
Posted by mirek on Tue, 05 Aug 2008 17:54:24 GMT
View Forum Message <> Reply to Message

tojocky wrote on Tue, 05 August 2008 11:37luzr wrote on Tue, 05 August 2008 16:44The problem is

```
iw.DrawDrawing(0, 0, old_size.cx, old_size.cy, resultpaint);
```

creates very deep recursion in Drawing definition (other drawing within drawin is stored as operation). That results in slow speed and huge VM.

Mirek

What another method will be more faster than DrawDrawing? May be DrawImage?

You are limited here by the speed of DrawLine.

Anyway, 3000 points sounds like quite a lot for rubber-band. Maybe you could try to simplifify it a bit?

Mirek

Subject: Re: How to implent a rubber band Class in u++
Posted by tojocky on Tue, 05 Aug 2008 22:29:38 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 05 August 2008 20:54tojocky wrote on Tue, 05 August 2008 11:37luzr wrote on Tue, 05 August 2008 16:44The problem is

```
iw.DrawDrawing(0, 0, old_size.cx, old_size.cy, resultpaint);
```

creates very deep recursion in Drawing definition (other drawing within drawin is stored as operation). That results in slow speed and huge VM.

Mirek

What another method will be more faster than DrawDrawing? May be DrawImage?

You are limited here by the speed of DrawLine.

Anyway, 3000 points sounds like quite a lot for rubber-band. Maybe you could try to simplifify it a bit?

Mirek

I want to understand how can I optimized in drawing and to use in future! In future I can have a control that Paint will have more operations and will be more optimized to save image and set it in Paint method.

Subject: Re: How to implent a rubber band Class in u++
Posted by mirek on Wed, 06 Aug 2008 06:34:04 GMT
View Forum Message <> Reply to Message

Well, one method that comes to mind is to use DrawPolyline.

IMO, significant bottleneck here is that each line drawn means one system call to Win32 API. Using Polyline this would be reduced to just single call.

Mirek

Subject: Re: How to implent a rubber band Class in u++
Posted by mrjt on Wed, 06 Aug 2008 09:04:05 GMT
View Forum Message <> Reply to Message

I 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.

```
class RubberBand : public LocalLoop
{
public:
 virtual void MouseMove(Point p, dword keyflags)  { AddPoint(p); GetMaster().Refresh(); }
 virtual void LeftUp(Point p, dword keyflags)     { EndLoop(); }
 virtual void RightUp(Point p, dword keyflags)    { EndLoop(); }

 const Vector<Point> & GetPoints()     { return points; }
 void      Clear()      { points.Clear(); }
private:
 Vector<Point> points;

 void AddPoint(const Point &newp);
};

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
  points.Top() = newp;
}
```

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.

```
 RubberBand  band;
 BackDraw  back;
 Size   backsz;

 void LeftDown(Point p, dword keyflags)
 {
  band.Clear();
  band.SetMaster(*this);
  Size sz = GetSize();
  if (sz != backsz) {
      back.Create(sz);
    backsz = sz;
  }
  Paint(back);
  band.Run();
  Refresh();
 }

 void Paint(Draw& w)
 {
  const Vector<Point> &p = band.GetPoints();
  if (!band.InLoop()) {
   // Normal painting
   w.DrawRect(GetSize(), White);
   if (p.GetCount() >= 2) {
    w.DrawPolyline(p);
    w.DrawLine(p[p.GetCount()-1], p[0]);
   }

  }
  else {
                    // RubberBand painting
   if (p.GetCount() >= 2)
    back.DrawLine(p[p.GetCount()-2], p.Top());
   back.Put(w, 0, 0);
   w.DrawText(4, 4, AsString(p.GetCount()));
  }
 }
```

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

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];
 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?
