Subject: Rect_ operators overloading suggestion
Posted by chickenk on Fri, 04 Jan 2008 17:25:45 GMT
View Forum Message <> Reply to Message

I would like to make a suggestion. I may have missed an important point so my suggestion would not make sense; don't hesitate to blame me for that.

I can see these overloads:
Rect_& operator+=(Sz sz){ Offset(sz); return *this; }
Rect_& operator+=(Pt p) { Offset(p); return *this; }
[...]
Rect_& operator-=(Sz sz){ Offset(-sz); return *this; }
Rect_& operator-=(Pt p) { Offset(-p); return *this; }
I agree with the Pt-argumented functions but I think it would make more sense for Sz-argumented functions to modify the size of the Rectangle instead of translating it...

For example, the following code could be used (I've not tested it):
Rect_& operator+=(Sz sz){ SetSize(Size()+sz); return *this; }
[...]
Rect_& operator-=(Sz sz){ SetSize(Size()-sz); return *this; }
What do you think about it ?

regards,
Lionel

Subject: Re: Rect_ operators overloading suggestion
Posted by mirek on Sat, 05 Jan 2008 10:33:40 GMT
View Forum Message <> Reply to Message

chickenk wrote on Fri, 04 January 2008 12:25I would like to make a suggestion. I may have missed an important point so my suggestion would not make sense; don't hesitate to blame me for that.

I can see these overloads:
Rect_& operator+=(Sz sz){ Offset(sz); return *this; }
Rect_& operator+=(Pt p) { Offset(p); return *this; }
[...]
Rect_& operator-=(Sz sz){ Offset(-sz); return *this; }
Rect_& operator-=(Pt p) { Offset(-p); return *this; }
I agree with the Pt-argumented functions but I think it would make more sense for Sz-argumented functions to modify the size of the Rectangle instead of translating it...

For example, the following code could be used (I've not tested it):
Rect_& operator+=(Sz sz){ SetSize(Size()+sz); return *this; }
[...]
Rect_& operator-=(Sz sz){ SetSize(Size()-sz); return *this; }
What do you think about it ?

regards,
Lionel

I think this classically ambiguos case.... I see good reasons for current overload as well for the proposed one...

Mirek

Subject: Re: Rect_ operators overloading suggestion
Posted by chickenk on Sat, 05 Jan 2008 19:04:30 GMT
View Forum Message <> Reply to Message

Thank you Mirek for your answer.

Unless it is too personal I'd like to know about the reasons for current implementation. It would fill an empty space in my brain

Thanks a lot
Lionel

Subject: Re: Rect_ operators overloading suggestion
Posted by mirek on Sat, 05 Jan 2008 22:29:24 GMT
View Forum Message <> Reply to Message

chickenk wrote on Sat, 05 January 2008 14:04Thank you Mirek for your answer.

Unless it is too personal I'd like to know about the reasons for current implementation. It would fill an empty space in my brain

Thanks a lot
Lionel

Well, Size can also have "offset" meaning. Adding Size to top-left point of Rect you get bottom-right point. Therefore it seems logical to interpret adding a Size as offset.

Meanwhile, r.Inflate(sz) is not much more verbose and certainly has quite clear meaning

Mirek

Subject: Re: Rect_ operators overloading suggestion
Posted by mdelfede on Sat, 05 Jan 2008 22:44:42 GMT

View Forum Message <> Reply to Message

BTW, in CAD terminology 'offset' would mean 'inflate' or 'deflate', not 'move' as upp interpretation

I got half an hour to get the upp meaning

Ciao

Max

Subject: Re: Rect_ operators overloading suggestion
Posted by chickenk on Sun, 06 Jan 2008 17:57:30 GMT
View Forum Message <> Reply to Message

Well before reading your answer I ended up using Deflate/Inflate actually. It makes my code nicely readable indeed. That's really enough for my needs, at least for now !

Thank you for your explanation.

Lionel