Subject: [Feature request] Relative positioning
Posted by rafiwui on Fri, 11 Aug 2017 07:51:32 GMT
View Forum Message <> Reply to Message

Working on an app with resizable option I discovered that I am not able to resize/reposition the ctrls relative to their original set size/position.
What I am thinking about is a sizing/positioning option so the ctrls get resized/positioned like the following:
1. Original size

2. X-Size 1.5x

3. X-Size 2x

4. Y-Size 1.5x

5. X-Size 1.5; Y-Size 2x

Source code for these layouts (a bit formatted for clean review of the important things):

```
LAYOUT(button_bar, 450, 50)
 ITEM(Button, generate, LeftPosZ( 10,  90).TopPosZ(10, 30))
 ITEM(Button, previous, LeftPosZ(130,  90).TopPosZ(10, 30))
 ITEM(Button, next,     LeftPosZ(230,  90).TopPosZ(10, 30))
 ITEM(Button, cancel,   LeftPosZ(350,  90).TopPosZ(10, 30))
END_LAYOUT

LAYOUT(button_bar_biggerX, 675, 50)
 ITEM(Button, generate, LeftPosZ( 15, 135).TopPosZ(10, 30))
 ITEM(Button, previous, LeftPosZ(195, 135).TopPosZ(10, 30))
 ITEM(Button, next,     LeftPosZ(345, 135).TopPosZ(10, 30))
 ITEM(Button, cancel,   LeftPosZ(525, 135).TopPosZ(10, 30))
END_LAYOUT

LAYOUT(button_bar_biggestX, 900, 50)
 ITEM(Button, generate, LeftPosZ( 20, 180).TopPosZ(10, 30))
 ITEM(Button, previous, LeftPosZ(260, 180).TopPosZ(10, 30))
 ITEM(Button, next,     LeftPosZ(460, 180).TopPosZ(10, 30))
 ITEM(Button, cancel,   LeftPosZ(700, 180).TopPosZ(10, 30))
END_LAYOUT

LAYOUT(button_bar_biggerY, 450, 75)
 ITEM(Button, generate, LeftPosZ( 10,  90).TopPosZ(15, 45))
 ITEM(Button, previous, LeftPosZ(130,  90).TopPosZ(15, 45))
 ITEM(Button, next,     LeftPosZ(230,  90).TopPosZ(15, 45))
 ITEM(Button, cancel,   LeftPosZ(350,  90).TopPosZ(15, 45))
END_LAYOUT
```

```
LAYOUT(button_bar_bigger_mixed, 675, 100)
 ITEM(Button, generate, LeftPosZ( 15, 135).TopPosZ(20, 60))
 ITEM(Button, previous, LeftPosZ(195, 135).TopPosZ(20, 60))
 ITEM(Button, next,     LeftPosZ(345, 135).TopPosZ(20, 60))
 ITEM(Button, cancel,   LeftPosZ(525, 135).TopPosZ(20, 60))
END_LAYOUT
```

Like you can see in the code it is basically a pretty simple thing, because everything is scaled up/positioned multiplied by the same factor: the scaling factor of the resized window.
If you can give me a hint where to find the method(s) called when resizing the window, maybe I can fit the library to my needs and/or contribute a solution.

Please correct me if there is already something to achieve that but I didn't find anything.

See this thread also.

File Attachments

1) upp_autoresize1.PNG, downloaded 672 times
2) upp_autoresize2.PNG, downloaded 663 times
3) upp_autoresize3.PNG, downloaded 659 times
4) upp_autoresize4.PNG, downloaded 639 times
5) upp_autoresize5.PNG, downloaded 663 times

Subject: Re: [Feature request] Relative positioning
Posted by koldo on Mon, 14 Aug 2017 06:59:40 GMT
View Forum Message <> Reply to Message

Dear Daniel

Now I understand. However I do not realize now how to implement it in a simple manner.
The only way I remember would be using Splitter (watch this), blocking in any way the user to resize them by himself.

Subject: Re: [Feature request] Relative positioning
Posted by rafiwui on Mon, 14 Aug 2017 07:29:16 GMT
View Forum Message <> Reply to Message

Quote:The only way I remember would be using Splitter (watch this), blocking in any way the user to resize them by himself.
Just checked it out but it doesn't work. It works the same way the standard resizing works.

So I would like to know what methods are called when resizing to try make my own solution ;)

Subject: Re: [Feature request] Relative positioning
Posted by cbpporter on Wed, 16 Aug 2017 08:24:46 GMT
View Forum Message <> Reply to Message

Hi rafiwui,

Layouts and automatic positioning is good enough for a lot of common task but sometimes you do need to go in and write your own positioning code.

You need to override this method in your window:

virtual void   Layout();

In this method you should query the size of the parent view-port, determine the center and align your 4 controls as in the pictures.

Subject: Re: [Feature request] Relative positioning
Posted by rafiwui on Wed, 13 Sep 2017 13:49:07 GMT
View Forum Message <> Reply to Message

I finished my work on my AutoResizer. So if anyone wants to use it simply include and inherit from AutoResizeTopWindow instead of TopWindow.
So here is the solution I came up with:

```
#ifndef _AutoResizeTopWindow_h_
#define _AutoResizeTopWindow_h_

#include <CtrlLib/CtrlLib.h>
using namespace Upp;

class AutoResizeTopWindow : public TopWindow
{
public:
 virtual void Layout()
 {
  Size currentSize = GetSize();
  if(firstSizing)
   startSize = currentSize;

  float sizeXChange = (float)currentSize.cx / (float)startSize.cx;
  float sizeYChange = (float)currentSize.cy / (float)startSize.cy;

  Ctrl* pChild = GetFirstChild();
  for(int i = 0; pChild != NULL; i++)
  {
   if(firstSizing)
    startRects.Add(pChild->GetRect());
```

```
   float rectLeft = startRects[i].left * sizeXChange;
   float rectTop = startRects[i].top * sizeYChange;
   float rectSizeX = startRects[i].Width() * sizeXChange;
   float rectSizeY = startRects[i].Height() * sizeYChange;

   pChild->SetRect(rectLeft, rectTop,
      rectSizeX, rectSizeY);

   pChild = pChild->GetNext();
  }
  firstSizing = false;
 }

private:
 bool firstSizing = true;
 Size startSize;
 Vector<Rect> startRects;
};

#endif
```

File Attachments

1) AutoResizeTopWindow.h, downloaded 276 times

Subject: Re: [Feature request] Relative positioning
Posted by koldo on Wed, 13 Sep 2017 16:26:34 GMT
View Forum Message <> Reply to Message

Thank you Daniel for showing us your work.

Subject: Re: [Feature request] Relative positioning
Posted by rafiwui on Thu, 14 Sep 2017 10:27:30 GMT
View Forum Message <> Reply to Message

Working with my code I realized that there is a big weak point: No widget that is not a direct child of the TopWindow is getting resized.
So I reworked my script and came up with this (IMO better) solution:

```
#ifndef _AutoResizeCtrl_h_
#define _AutoResizeCtrl_h_

#include <CtrlLib/CtrlLib.h>
```

```
using namespace Upp;

#include <type_traits>

template<typename T>
class AutoResizeCtrl : public T
{
   // Make sure the base class inherits or is Ctrl
   static_assert(std::is_base_of<Ctrl, T>::value, "T must inherit from Ctrl");

private:
   bool initialization;
   bool addedChildren;
   Rect initializationRect;
   Vector<Rect> childrenInitializationRects;

public:
   AutoResizeCtrl<T>()
      : T()
   {
      initialization = true;
      addedChildren = false;
   }

   virtual void Layout()
   {
      Rect currentRect = GetRect();
      if(initialization)
         initializationRect = currentRect;

      float sizeXChange = (float)currentRect.Width() / (float)initializationRect.Width();
      float sizeYChange = (float)currentRect.Height() / (float)initializationRect.Height();

 Ctrl* pChild = GetFirstChild();
 for(int i = 0; pChild != NULL; i++)
 {
         if(initialization)
    {
  childrenInitializationRects.Add(pChild->GetRect());
  addedChildren = true;
    }

    float rectLeft = childrenInitializationRects[i].left * sizeXChange;
    float rectTop = childrenInitializationRects[i].top * sizeYChange;
    float rectSizeX = childrenInitializationRects[i].Width() * sizeXChange;
    float rectSizeY = childrenInitializationRects[i].Height() * sizeYChange;

    pChild->SetRect(rectLeft, rectTop,
```

```
            rectSizeX, rectSizeY);
        pChild->Layout();
        pChild = pChild->GetNext();
    }
    initialization = !addedChildren;
      }
};

#endif
```

For an example how to use it please check the attached package.

File Attachments

1) AutoResizeExample.7z, downloaded 273 times