U++ - Bug #484
Automated sorting of layout items (based on graphical position) in layout designer

06/13/2013 12:26 PM - Miroslav Fidler

Status:	Approved	**Start date:**	06/13/2013
Priority:	Normal	**Due date:**	
Assignee:	Miroslav Fidler	**% Done:**	0%
Category:	IDE	**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour

Description

History

#1 - 08/13/2013 08:37 AM - Sender Ghost

- File 484_uppsrc.diff added

- Status changed from New to Patch ready

Added possible implementation.

#2 - 08/13/2013 09:33 AM - Sender Ghost

- File deleted (484_uppsrc.diff)

#3 - 08/13/2013 09:34 AM - Sender Ghost

- File 484_uppsrc.diff added

#4 - 08/13/2013 08:45 PM - Sender Ghost

The only drawback of proposed sorting algorithm is that it doesn't use information about alignment.

#5 - 08/13/2013 11:34 PM - Sender Ghost

- File 484_uppsrc2.diff added

Added second version of the patch.
Now, the sorting algorithm uses information about alignment.

#6 - 08/20/2013 12:05 AM - Sender Ghost

- File 484_uppsrc3.diff added

Added third version of the patch.

The algorithm is based on second version of the patch, but uses **IndexSort2** function, instead of **GetSortOrder** and additional **Array<LayoutItem>**. Also I added the check for count of layout items less than two, where sorting is not necessary.

#7 - 08/20/2013 12:20 AM - Sender Ghost

- File deleted (484_uppsrc3.diff)

#8 - 08/20/2013 12:21 AM - Sender Ghost

- File 484_uppsrc3.diff added

#9 - 08/21/2013 09:46 AM - Miroslav Fidler

- Status changed from Patch ready to Ready for CR

In the end, I have used slightly different code - RectLess needed to be made more tolerating in vertical axis to correctly accomodate labels, which made it non-transitive, hence selection sort used instead. Also, we want only to sort selected items...

#10 - 08/21/2013 03:58 PM - Sender Ghost

I have a different use case, where my version of changes is alright.
But if you need my opinion about your changes, then I have some issues to talk about:
1. Your changes is not relative to selected item (cursor.Top()). There will be different order of selection after sort, as well as different list's position of selected item.
2. The sort algorithm is ambiguous, i.e. there are cases, where next try will sort differently, e.g. for following layout:

```
LAYOUT(TestLayout, 356, 76)
  ITEM(LineEdit, edit, LeftPosZ(188, 132).TopPosZ(4, 19))
  ITEM(Label, label, SetLabel(t_("Label")).LeftPosZ(324, 28).TopPosZ(4, 16))
  ITEM(DataPusher, data1, LeftPosZ(4, 64).TopPosZ(20, 24))
  ITEM(DataPusher, data2, LeftPosZ(4, 64).TopPosZ(48, 24))
END_LAYOUT
```

#11 - 09/02/2013 03:09 PM - Miroslav Fidler

- Assignee changed from Miroslav Fidler to Sender Ghost

OK, based on suggested issue and some initial experience (you would not believe how often I need this :) I have made following changes:

Changed sorting algorithm to bubblesort, this should at least make it stable (IMO does not change the order if it can be interpreted as fine).

Also changed the algorithm to actually collect all items after the cursor item and place cursor to the last item at the end (which IME is what I want to happen).

#12 - 09/02/2013 10:28 PM - Sender Ghost

- File 484_uppsrc_r6303.diff added
- Status changed from Ready for CR to Patch ready
- Assignee changed from Sender Ghost to Miroslav Fidler

Miroslav Fidler wrote:

> *Changed sorting algorithm to bubblesort, this should at least make it stable*

Your algorithm(s) doesn't sort following layout (when all items selected (**Ctrl + A**)):

```
LAYOUT(TestLayout, 356, 76)
  ITEM(Label, label, SetLabel(t_("Label")).LeftPosZ(324, 28).TopPosZ(4, 16))
  ITEM(DataPusher, data1, LeftPosZ(4, 64).TopPosZ(20, 24))
  ITEM(LineEdit, edit, LeftPosZ(188, 132).TopPosZ(4, 19))
  ITEM(DataPusher, data2, LeftPosZ(4, 64).TopPosZ(48, 24))
END_LAYOUT
```

Miroslav Fidler wrote:

> *Also changed the algorithm to actually collect all items after the cursor item and place cursor to the last item at the end (which IME is what I want to happen).*

I still consider changing the cursor to different item, than selected one, as an issue (i.e., distraction of user's attention).

I created new patch, which fixes mentioned issues. It also contains the fix for item selection from ArrayCtrl with **Ctrl + mouse click**.

#13 - 09/03/2013 07:46 PM - Miroslav Fidler

- Assignee changed from Miroslav Fidler to Sender Ghost

Well, I understand your concerns, but for my usage scenario (and I have used it 4 times today, saved me about 20 minutes of work) the comparison really needs to be "tolerant", which obviously creates ambiguity, but I rather deal with that (you can e.g. sort by parts) than not to be able to use the function at all.

What problem is there with Ctrl+Mouseclick?

#14 - 09/04/2013 02:57 AM - Sender Ghost

- File 484_uppsrc2_r6303.diff added

- Assignee changed from Sender Ghost to Miroslav Fidler

Miroslav Fidler wrote:

> *the comparison really needs to be "tolerant", which obviously creates ambiguity, but I rather deal with that (you can e.g. sort by parts) than not to be able to use the function at all.*

Ok, I understood, that you need this specific function for your work. I also found it useful, but with different algorithm (sorting by top-left position).

Miroslav Fidler wrote:

> *What problem is there with Ctrl+Mouseclick?*

It collects the same items, instead of removing them, when exists (like LayDes::SelectOne method does for graphical representation), which may lead to crash inside of LayDes:SortItems method (e.g. just click many times for the same **ArrayCtrl** item and try to use "Sort by position" function).

In conclusion, I could suggest to use two algorithms:
1. Sort by position.

2. Sort by top-left position.

I think, there are no need for top-right, bottom-left, bottom-right functions, because the size increases from top to bottom and left to right (like **ArrayCtrl** item list graphical representation), which is ascending (two dimensional) sorting (with top preference over left position, in this case).

The second patch for r6303 contains the same fixes, but with two functions.

#15 - 09/04/2013 07:54 PM - Miroslav Fidler

- Assignee changed from Miroslav Fidler to Sender Ghost

OK, one last try: by making RectLess less tolerant, it sorts your layout too, while retaining ability to sort what I need... Obviously, there still will be some ambiguous cases, but these will by "human eye" ambiguous as well.

Thanks for spotting the ctrl+click bug.

#16 - 09/04/2013 07:54 PM - Miroslav Fidler

- Status changed from Patch ready to Ready for CR

#17 - 09/04/2013 09:41 PM - Sender Ghost

- Status changed from Ready for CR to Approved
- Assignee changed from Sender Ghost to Miroslav Fidler

Ok, now the algorithm is able to sort layout(s) for test.
As I understood, changing an item to last cursor position after sort - is intentional feature.

Thanks.

Files

484_uppsrc.diff	2.09 KB	08/13/2013	Sender Ghost
484_uppsrc2.diff	1.99 KB	08/13/2013	Sender Ghost
484_uppsrc3.diff	1.92 KB	08/19/2013	Sender Ghost
484_uppsrc_r6303.diff	2.08 KB	09/02/2013	Sender Ghost
484_uppsrc2_r6303.diff	3.22 KB	09/04/2013	Sender Ghost