

User Interface - Bug #2850

Bug # 2677 (New): fix drawing and functional differences between P2J GUI and 4GL GUI

menu/simple_sm.p drawing issues in Swing client

11/16/2015 01:59 PM - Greg Shah

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Vadim Gindin	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	GUI Support for a Complex ADM2 App	case_num:	
billable:	No		
vendor_id:	GCD		
Description			

History

#1 - 11/16/2015 02:00 PM - Greg Shah

Using ./menu/simple_sm.p in the Swing client, Vadim reported:

1. After I press on some menu-item in the on of sub-menus of a menubar, root sub-menu in menubar stays "mouse-entered" after body is hidden.
2. The same scenario: sub-menu child is still displayed as focused after it's parent's body is hidden.
3. At the start of the test the warning "Run-time error ... (3269)" is shown and no menubar is displayed in PROGRESS until the user will press any key, but in P2J empty menubar rectangle is drawn.

#2 - 11/18/2015 05:36 PM - Vadim Gindin

- File *fonts_compare.png* added

- File *hidden_menu_elems.png* added

Here are the images

1. Fonts compare *fonts_compare.png* of PROGRESS, P2J SWING, P2J WEB. There is an other font for WEB.
2. MENU bug of showing hidden menu items *hidden_menu_elems.png*

#3 - 11/18/2015 05:41 PM - Vadim Gindin

- File *deleted (hidden_menu_elems.png)*

#4 - 11/18/2015 05:41 PM - Vadim Gindin

- File *deleted (fonts_compare.png)*

#5 - 11/18/2015 05:46 PM - Vadim Gindin

Greg Shah wrote:

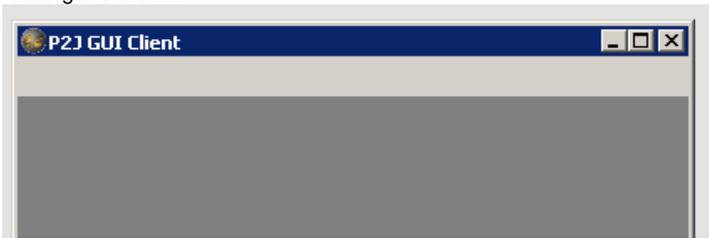
3. At the start of the test the warning "Run-time error ... (3269)" is shown and no menubar is displayed in PROGRESS until the user will press any key, but in P2J empty menubar rectangle is drawn.

The place where that background is drawn is the same as for popup_ext.p: BorderedPanelGuiImpl.draw(). I've checked that at the drawing time menubar is hidden and it does not post PaintEvent's. So there are some other widget posted clipping rectangle, that intersects menubar area.. Once again if the widget is responsible to draw it's background itself, why it is needed to draw some rectangles in BorderedPanelGuiImpl.draw()?

#6 - 11/19/2015 10:03 AM - Vadim Gindin

- File menubar_background.png added

I'm trying to understand how the "menubar's" rectangle get into screen clippings, that are drawn in BorderedPanelGuiImpl.draw(). I recall that at that moment MENUBAR itself is hidden and does not post PaintEvent, so there is some other widget, that posts a PaintEvent with similar rectangle. I.e. right after start, when the warning "Run-time error in WAIT-FOR at line ..." is shown an a messagebar menubar is still hidden but it's background rectangle is drawn:



I've added logging of processed PaintEvent native rectangle and source to TitledWindow.processEvent() method (line 163). Here is what I've got (top, left, bottom, right):

```
PE: [0, 0, 41, 2147483647], source=MessageAreaGuiImpl
PE: [0, 0, 29, 399], source=MessageAreaGuiImpl
PE: [0, 0, 41, 399], source=BorderedPanelGuiImpl
PE: [0, 0, 29, 399], source=BorderedPanelGuiImpl
PE: [0, 0, 2147483647, 2147483647], source=WindowTitleBar
PE: [0, 0, 2147483647, 399], source=WindowTitleBar
PE: [0, 0, 2147483645, 2147483645], source=WindowTitleBar$WindowIcon
PE: [1, 2, 2147483646, 2147483647], source=WindowTitleBar$WindowIcon
PE: [0, 0, 2147483647, 399], source=WindowTitleBar$WindowTitle
PE: [0, 0, 17, 399], source=WindowTitleBar$WindowTitle
PE: [0, 0, 18, 399], source=WindowTitleBar
PE: [0, 0, 13, 15], source=CaptionButton
PE: [2, 382, 15, 397], source=CaptionButton
PE: [0, 0, 13, 15], source=CaptionButton
PE: [2, 364, 15, 379], source=CaptionButton
PE: [0, 0, 13, 15], source=CaptionButton
PE: [2, 348, 15, 363], source=CaptionButton
PE: [1, 19, 18, 399], source=WindowTitleBar$WindowTitle
PE: [1, 19, 18, 345], source=WindowTitleBar$WindowTitle
PE: [0, 0, 541, 408], source=WindowGuiImpl
PE: [23, 4, 484, 403], source=WindowWorkSpace
PE: [23, 4, 484, 403], source=BorderedPanelGuiImpl
PE: [0, 0, 540, 407], source=BorderedPanelGuiImpl
PE: [4, 4, 22, 403], source=WindowTitleBar
PE: [485, 4, 514, 403], source=MessageAreaGuiImpl
PE: [515, 4, 536, 403], source=StatusLineGuiImpl
PE: [485, 4, 514, 403], source=MessageAreaGuiImpl
PE: [485, 4, 514, 403], source=BorderedPanelGuiImpl
PE: [4, 4, 22, 403], source=WindowTitleBar
PE: [5, 6, 20, 21], source=WindowTitleBar$WindowIcon
PE: [5, 23, 22, 349], source=WindowTitleBar$WindowTitle
PE: [6, 386, 19, 401], source=CaptionButton
PE: [6, 368, 19, 383], source=CaptionButton
PE: [6, 352, 19, 367], source=CaptionButton
PE: [23, 4, 484, 403], source=WindowWorkSpace
PE: [23, 4, 484, 403], source=BorderedPanelGuiImpl
PE: [0, 0, 541, 408], source=WindowGuiImpl
PE: [0, 0, 564, 408], source=WindowGuiImpl
PE: [0, 0, 540, 407], source=BorderedPanelGuiImpl
PE: [0, 0, 563, 407], source=BorderedPanelGuiImpl
PE: [508, 4, 537, 403], source=MessageAreaGuiImpl
```

```
PE: [538, 4, 559, 403], source=StatusLineGuiImpl
PE: [508, 4, 537, 403], source=MessageAreaGuiImpl
PE: [538, 4, 559, 403], source=StatusLineGuiImpl
PE: [4, 4, 22, 403], source=WindowTitleBar
PE: [0, 0, 564, 408], source=WindowGuiImpl
PE: [4, 4, 22, 403], source=WindowTitleBar
PE: [4, 4, 22, 403], source=WindowTitleBar
PE: [0, 0, 564, 408], source=WindowGuiImpl
PE: [4, 4, 22, 403], source=WindowTitleBar
PE: [4, 4, 22, 403], source=WindowTitleBar
```

Menubar rectangle starts at top=23px approximately, so I don't see here exact rectangle, but here are some intersecting rectangles and some rectangles seems strange:

```
PE: [0, 0, 41, 2147483647], source=MessageAreaGuiImpl
PE: [0, 0, 29, 399], source=MessageAreaGuiImpl
PE: [0, 0, 41, 399], source=BorderedPanelGuiImpl
PE: [0, 0, 29, 399], source=BorderedPanelGuiImpl
```

Message area resides in the bottom of the window. It looks that rectangles are incorrect. What do you think?

#7 - 11/20/2015 09:44 AM - Constantin Asofiei

Vadim, in which branch are you working this?

Also, please remind me, have you managed to address the sub-menu body repaint I mentioned some time ago?

#8 - 11/20/2015 09:57 AM - Constantin Asofiei

Vadim, I think I know why you are seeing the top light-gray space. Menu-bar gets attached to window (and window is relayout) BEFORE making it visible in GUI. At the time this code is called and warning is shown:

```
if (!hasEnabledFrame())
{
    warnNoEnabledFrames(wnd);
}

wnd.getMenubar().setVisible(true);
```

the window is already layout as if a menu-bar is added. Only later the menu-bar is being drawn (as it is being made visible).

Can you check what happens if the menubar gets set to not-visible via a trigger? Is the window workspace resized and the menubar hidden? If so, the window needs to layout and use the menubar only if the menubar is visible; and if the menubar's visibility gets changed, the window needs to be relayout.

#9 - 11/22/2015 07:54 AM - Vadim Gindin

Constantin Asofiei wrote:

Vadim, in which branch are you working this?

branch 2677a, revno 11067.

Also, please remind me, have you managed to address the sub-menu body repaint I mentioned some time ago?

You probably mean the note 22 in the task [#2736](#). Yes, I'd tried that, but unsuccessfully: sub-menu's aren't drawn. You can see details about it in that task [#2736](#) the latest post (popup_ext.p). More over I think there is another reason in that task.

#10 - 11/23/2015 03:09 AM - Vadim Gindin

- File 1123.diff added

Constantin. You were absolutely right about the reason. I've added isVisible check for MENUBAR to WindowLayout and fixed dynamic attach/detach case for GUI and CHUI. By the way, MENUBAR can be shown only by assigning to window:menubar attribute. visible attribute is not settable for menu. I've added corresponding error. I've also fixed CHUI layout for dynamic case.

Attached diff contains my current changes.

#11 - 11/24/2015 07:18 AM - Greg Shah

Code Review 1123.diff

1. I don't think it is a good idea to require that the status line and message line implementations must calculate whether or not the menubar exists to determine their location. The containing window should have a method that reports the correct location for each one. Actually, I don't understand why the location changes when the menubar is present. The screen dimensions should not be different and the status/message lines are at the same location no matter if the menubar exists or not.

2. You have mixed in your other changes (right mouse issues and toggle-box issues). I don't think the right mouse changes are safe, so these cannot be included.

Constantin: please review.

#12 - 11/24/2015 07:29 AM - Constantin Asofiei

Beside what Greg noted:

- in ChUI, message area, status area, frames (and menubar) are all attached to the same parent (in other terms, they are all siblings). So the change in WindowChuiImpl doesn't look correct, as the contentPane will contain everything, not just frames
- in ChUI, we don't have (yet) a container separation between frames and message area/status area/menubar. This requires more testing, to determine if a frame can overlay the last 3 rows (message + status) in the screen. If it can not, then ChUI needs a "workspace" equivalent, as GUI has, to separate the frames.
- as Greg mentioned, the location/height of the status/message area is constant. Only the height available for frame placement is reduced when the menubar is attached in ChUI.

#13 - 11/24/2015 08:54 AM - Vadim Gindin

So should I fix it now or postpone that?

#14 - 11/24/2015 08:58 AM - Greg Shah

1. Separate out the changes for toggle-box/right mouse issues.
2. Fix the rest based on the code reviews.

#15 - 11/24/2015 12:18 PM - Vadim Gindin

- *File 1124.diff added*

All 3 points in this task are fixed. See attached diff. I've separated issues from other tasks.

#16 - 11/24/2015 02:10 PM - Greg Shah

- *Assignee set to Vadim Gindin*

Code Review 1124.diff

My only question is regarding the addition of the `Window.currentFocus()` code. This seems wrong. If I am understanding it properly, as long as there is ever a popup menu in the widget list for a window, only that popup menu will ever be focusable.

Please explain.

#17 - 11/24/2015 02:35 PM - Vadim Gindin

Greg Shah wrote:

Code Review 1124.diff

My only question is regarding the addition of the `Window.currentFocus()` code. This seems wrong. If I am understanding it properly, as long as there is ever a popup menu in the widget list for a window, only that popup menu will ever be focusable.

Please explain.

It is just a such behaviour: while popup menu is opened, tab order isn't working and navigation is available only inside menu using arrows. It probably relates to [#2849](#) task for ignoring right mouse button: when we do not set a focus to the button, on which we clicked using right mouse button, we will

not be able to find current focused item in menu.. I can also move it at this point from the current update to [#2849](#) changes set.

P.S.
Another bug for CHUI: Focused element of MENUBAR must be drawn as Color.MESSAGES, but it is drawn as Color.NORMAL. I recall that Color.MESSAGES=Color.NORMAL.withReverse().

#18 - 11/24/2015 02:42 PM - Greg Shah

It is just a such behaviour: while popup menu is opened, tab order isn't working and navigation is available only inside menu using arrows. It probably relates to [#2849](#) task for ignoring right mouse button: when we do not set a focus to the button, on which we clicked using right mouse button, we will not be able to find current focused item in menu.

I have 2 concerns:

1. When the popup menu is not open, the currentFocus() will be broken.
2. I don't think that the popup menu should be treated as a widget contained by the window. It is more like the combo-box drop-down. It temporarily appears and when it is open it owns the interaction with the user.

#19 - 11/24/2015 03:01 PM - Vadim Gindin

Greg Shah wrote:

It is just a such behaviour: while popup menu is opened, tab order isn't working and navigation is available only inside menu using arrows. It probably relates to [#2849](#) task for ignoring right mouse button: when we do not set a focus to the button, on which we clicked using right mouse button, we will not be able to find current focused item in menu.

I have 2 concerns:

1. When the popup menu is not open, the currentFocus() will be broken.

In that case w.isVisible()==false and therefore super.currentFocus() will be called. Isn't it?

2. I don't think that the popup menu should be treated as a widget contained by the window. It is more like the combo-box drop-down. It temporarily appears and when it is open it owns the interaction with the user.

The difference is in the fact, than Menu holds it's own widgets tree of menu widgets, that are also focusable.

#20 - 11/24/2015 04:40 PM - Greg Shah

In that case `w.isVisible()==false` and therefore `super.currentFocus()` will be called. Isn't it?

Yes, you're probably right.

2. I don't think that the popup menu should be treated as a widget contained by the window. It is more like the combo-box drop-down. It temporarily appears and when it is open it owns the interaction with the user.

The difference is in the fact, than Menu holds it's own widgets tree of menu widgets, that are also focusable.

The key point here is that we don't want to spread widget-specific behavior (e.g. for menus) into other non-related classes (e.g. window). It makes it hard to understand how menus work. It makes it hard to find all the behavior related to menus. It makes those classes very dependent upon one another, which makes things fragile and more likely to break.

Constantin/Hynek: thoughts?

#21 - 11/25/2015 03:18 AM - Hynek Cihlar

Greg Shah wrote:

In that case `w.isVisible()==false` and therefore `super.currentFocus()` will be called. Isn't it?

Yes, you're probably right.

2. I don't think that the popup menu should be treated as a widget contained by the window. It is more like the combo-box drop-down. It temporarily appears and when it is open it owns the interaction with the user.

The difference is in the fact, than Menu holds it's own widgets tree of menu widgets, that are also focusable.

The key point here is that we don't want to spread widget-specific behavior (e.g. for menus) into other non-related classes (e.g. window). It

makes it hard to understand how menus work. It makes it hard to find all the behavior related to menus. It makes those classes very dependent upon one another, which makes things fragile and more likely to break.

Constantin/Hynek: thoughts?

I think Greg is right. Menu should be rendered in its own top-level window, otherwise we will have the same issue as with combo drop-down - the menu won't be able to draw outside of its logical parent window.

#22 - 11/25/2015 09:50 AM - Vadim Gindin

OK, I understand.

I have collected changes for 2-3 tasks and I would want to commit some of them, that are approved. All issues in this task are approved. Can I commit them (the latest diff) excluding Menu change?

Talking about ability to draw menu with opened sub-menus exceeding window sizes, is there a separate task for this already? Is it solved for combo-box or other widget, that has the same issue?

#23 - 11/25/2015 10:23 AM - Hynek Cihlar

Vadim Gindin wrote:

Talking about ability to draw menu with opened sub-menus exceeding window sizes, is there a separate task for this already? Is it solved for combo-box or other widget, that has the same issue?

There is an issue for combo-box drop-down, see [#2837](#).

#24 - 11/25/2015 11:52 AM - Vadim Gindin

- File 2850_2511.diff added

Here is the latest diff on merged branch.

#25 - 11/26/2015 03:39 AM - Vadim Gindin

Yesterday, I've fixed CHUI bug with menubar:

Another bug for CHUI: Focused element of MENUBAR must be drawn as Color.MESSAGES, but it is drawn as Color.NORMAL. I recall that Color.MESSAGES=Color.NORMAL.withReverse().

So once again can I commit the last change (diff in the note [#24](#))?

#26 - 11/27/2015 12:04 PM - Vadim Gindin

- File 2711_rev10976.diff added

Here is the current changes diff (base on revision 10976 of the branch 2677b). Changes solve 3 bugs, described in the note 1 of this task. Please review and approve commit.

#27 - 11/27/2015 01:06 PM - Greg Shah

Go ahead and check these into 2677b.

#28 - 11/27/2015 01:29 PM - Vadim Gindin

Committed with revno 10978.

#29 - 11/27/2015 02:08 PM - Greg Shah

Are there any issues mentioned in this task which are not yet fixed AND which are not part of some other open task?

#30 - 11/27/2015 02:24 PM - Vadim Gindin

There are the following issues:

1. CHUI. Dynamic attach/detach menubar (MENUBAR layout problems). It correlates with the task [#2557](#)
2. CHUI. Fix pop-up menu layout (popup-ext.p).
3. GUI. The case when opened sub-menus can reside outside of the window. It correlates with the task [#2837](#).

#31 - 12/12/2015 06:15 AM - Greg Shah

CHUI. Dynamic attach/detach menubar (MENUBAR layout problems). It correlates with the task [#2557](#)

It was my understanding that this was a GUI-only issue as noted in [#2557](#).

CHUI. Fix pop-up menu layout (popup-ext.p).

This is still broken? Is it already being worked in another task?

If it is broken and not being worked, please create a new task for this and post the task number here.

I plan to close this task once that task is created, unless you know of a reason I should not.

#32 - 12/14/2015 02:22 AM - Vadim Gindin

I've created the task [#2933](#) for CHUI layout errors.

#33 - 12/14/2015 08:47 AM - Greg Shah

- % Done changed from 0 to 100

- Status changed from New to Closed

#34 - 11/16/2016 12:12 PM - Greg Shah

- Target version changed from Milestone 12 to GUI Support for a Complex ADM2 App

Files

menubar_background.png	1.67 KB	11/19/2015	Vadim Gindin
1123.diff	14.1 KB	11/23/2015	Vadim Gindin
1124.diff	9.14 KB	11/24/2015	Vadim Gindin
2850_2511.diff	4.96 KB	11/25/2015	Vadim Gindin
2711_rev10976.diff	4.52 KB	11/27/2015	Vadim Gindin