

User Interface - Bug #2970

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

rework menus to use OverlayWindow

01/26/2016 11:42 AM - Greg Shah

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Hynek Cihlar	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	Cleanup and Stabilization for GUI	case_num:	
billable:	No		
vendor_id:	GCD		
Description			
Related issues:			
Related to User Interface - Bug #2837: GUI combo-box drop-down needs to be ab...			Closed

History

#1 - 01/26/2016 11:44 AM - Greg Shah

Sub-menus and popup menus must be contained in an OverlayWindow so that their contents can draw outside of the bounds of the "real" containing window (when needed).

Please see [#2837](#) for details of how this was resolved for combo-box drop-downs. The same approach should be used here.

#2 - 03/08/2016 11:35 AM - Greg Shah

- Assignee set to Eugenie Lyzenko

#3 - 03/23/2016 05:26 PM - Greg Shah

- Target version changed from Milestone 12 to Milestone 16

#4 - 05/19/2016 06:18 PM - Hynek Cihlar

- Status changed from New to WIP

- Assignee changed from Eugenie Lyzenko to Hynek Cihlar

#5 - 05/29/2016 05:55 PM - Hynek Cihlar

2970a rebased against trunk 11039, now at revision 11045.

#6 - 06/03/2016 07:35 AM - Hynek Cihlar

2970a rebased against trunk 11041, now at revision 11049.

#7 - 06/07/2016 07:32 PM - Hynek Cihlar

Task branch 2970a 11058 is the reworked menus ready for review. The revision has passed ChUI and Swing GUI regression testing. I am still having some minor issues with Web GUI, but as Swing GUI already works OK I only expect small changes to come.

#8 - 06/08/2016 09:37 AM - Hynek Cihlar

Task branch 2970a 11059 resolves the issues mentioned in previous note and passes Web GUI regression testing. Please review.

#9 - 06/08/2016 10:15 AM - Greg Shah

Code Review Task Branch 2970a Revision 11059

This is a very nice improvement. I think the menu implementation is cleaner and easier to understand with this approach. I also like the addition of the `GuiDriver.withSelectedWindow()` and the `AbstractContainer.cycleFocus()` code.

1. Why was the `KA_CURSOR_RIGHT` event processing removed from `SubMenu.processKeyEvent()`? Is it only needed for GUI and the `SubMenuItemImpl` body inner class event processing handles it?
2. The constructor for `WindowTitlePopupGuiImpl` should be at the top of the class, even though it is private (it is a documented exception in our coding standards).
3. Please move the private void `initEditorItems(boolean textEmpty)` to the bottom of `EditorPopupGuiImpl`.
4. Please make the import `java.awt.event.MouseEvent`; explicit again in `OverlayWindow`. Also please add a comment that the AWT dependency is being imported explicitly for use of event processing and AWT support in general is not in use here.

#10 - 06/08/2016 10:41 AM - Constantin Asofiei

Hynek, beside what Greg noted, the changes in 2970a/11059 are OK, I like that they are not invasive so they do not affect the ChUI menu widgets.

You have a TODO: remove for `@SubMenuItemImpl.isDescendantFocused`.

#11 - 06/08/2016 11:25 AM - Hynek Cihlar

Greg Shah wrote:

Code Review Task Branch 2970a Revision 11059

1. Why was the `KA_CURSOR_RIGHT` event processing removed from `SubMenuItemImpl.processKeyEvent()`? Is it only needed for GUI and the `SubMenuItemImpl` body inner class event processing handles it?

Good question. This was removed because the condition was never executed. The same condition a couple of lines higher makes the method return early. Also there didn't seem to be a valid use case for it, it was probably copy&pasted from `MenuItem`.

2. The constructor for `WindowTitlePopupGuiImpl` should be at the top of the class, even though it is private (it is a documented exception in our coding standards).
3. Please move the private void `initEditorItems(boolean textEmpty)` to the bottom of `EditorPopupGuiImpl`.
4. Please make the import `java.awt.event.MouseEvent`; explicit again in `OverlayWindow`. Also please add a comment that the AWT dependency is being imported explicitly for use of event processing and AWT support in general is not in use here.

2-4 fixed.

The changes committed to 2970a revision 11060.

#12 - 06/08/2016 11:25 AM - Hynek Cihlar

Constantin Asofiei wrote:

You have a TODO: remove for @SubMenuGuiImpl.isDescendantFocused.

The unused method removed in 2970a revision 11060. Thanks.

#13 - 06/08/2016 01:11 PM - Eugenie Lyzenko

Code review for 11060 revision.

1. gui/MenuItemGuiImpl.java. The boolean flag mouseEntered is removed from onFocusGained() method. Does it mean the pop-up menu can not have the focus after displaying sometimes?
2. gui/SubMenuGuiImpl.java. The boolean flag mouseEntered usage is reworked. Please explain the idea behind the change and current approach of this flag usage.
3. widget/AbstractContainer.java removed code to find out the menu behind container. How this functionality will now work?
4. widgetbrowser/WidgetBrowserAspect.java needs to remove commented out code or put additional TODO comments.
5. client/SubMenu.java does not remove menu on right key. This is OK when current menu item has submenu item to show on right key reaction. What if not?

#14 - 06/08/2016 01:19 PM - Vadim Gindin

I've just tried to run menu/simple_sm.p and I got infinite loop. Did you run that test?

#15 - 06/08/2016 01:35 PM - Vadim Gindin

Some additional questions:

1. OverlayWindow.draw() contains the following code.

```
if (!isVisible())  
{  
    return;  
}
```

As I understand draw() method must not be called if isVisible() == false.

2. SubMenuGuiImpl.setVisible() makes all child items visible. But it is not necessary when sub-menu body isn't shown. At some moment we'd discussed that case and came to solution, when child items become visible at the body show moment. Just a remark. I'm not sure if it good for new approach.

Eugenie Lyzenko wrote:

Code review for 11060 revision.

1. gui/MenuItemGuiImpl.java. The boolean flag mouseEntered is removed from onFocusGained() method. Does it mean the pop-up menu can not have the focus after displaying sometimes?

I moved the flag mouseEntered to mouseEntered() method. It didn't make much sense to have the flag set in onFocusGained(), this way it was redundant with AbstractWidget.hasFocus().

2. gui/SubMenuGuiImpl.java. The boolean flag mouseEntered usage is reworked. Please explain the idea behind the change and current approach of this flag usage.

I am not sure what usage do you refer to, could you maybe point to a specific piece of code?

3. widget/AbstractContainer.java removed code to find out the menu behind container. How this functionality will now work?

I am glad you point this out. I am a bit worried that I can brake something with this change. I wasn't able to find any valid use case for this and with the menus now being in separate window, this seemed to be redundant. Can you describe the functionality you are referring to?

4. widgetbrowser/WidgetBrowserAspect.java needs to remove commented out code or put additional TODO comments.

Thanks for pointing this out, I will yet remove this change as it was only for debugging purposes.

5. client/SubMenu.java does not remove menu on right key. This is OK when current menu item has submenu item to show on right key reaction. What if not?

The removed code didn't have any effect and there doesn't seem to be a use case for it anyway.

#17 - 06/08/2016 02:30 PM - Hynek Cihlar

Vadim Gindin wrote:

I've just tried to run menu/simple_sm.p and I got infinite loop. Did you run that test?

Thanks for catching this. There was indeed an unhandled exception when there was a popup menu not attached anywhere.

I have committed a fix to 2670a revisions 11061 and 11062.

#18 - 06/08/2016 03:11 PM - Eugenie Lyzenko

2. gui/SubMenuGuiImpl.java. The boolean flag mouseEntered usage is reworked. Please explain the idea behind the change and current approach of this flag usage.

I am not sure what usage do you refer to, could you maybe point to a specific piece of code?

Previously mouseEntered become true when submenu gained focus or be shown and become false when submenu lost focus. So this is after respective event happening. Now this flag is changing in mouse event mouse(Entered/Exited) - it is a bit different approach. The flag is heavily used within class file. So I would like to know why the flag setting approach was changed, what was wrong with old way and what is the idea behind now.

3. widget/AbstractContainer.java removed code to find out the menu behind container. How this functionality will now work?

I am glad you point this out. I am a bit worried that I can brake something with this change. I wasn't able to find any valid use case for this and with the menus now being in separate window, this seemed to be redundant. Can you describe the functionality you are referring to?

If the container that holds menu(in OverlayWindow I guess) will run currentFocus() and focus() methods how it will find and return the menu widget reference if there is one?

Eugenie Lyzenko wrote:

2. gui/SubMenuGuiImpl.java. The boolean flag mouseEntered usage is reworked. Please explain the idea behind the change and current approach of this flag usage.

I am not sure what usage do you refer to, could you maybe point to a specific piece of code?

Previously mouseEntered become true when submenu gained focus or be shown and become false when submenu lost focus. So this is after respective event happening. Now this flag is changing in mouse event mouse(Entered/Exited) - it is a bit different approach. The flag is heavily used within class file. So I would like to know why the flag setting approach was changed, what was wrong with old way and what is the idea behind now.

There is no functional difference, the change makes the code more readable.

Consider the following condition from the menu logic.

```
if (SubMenuGuiImpl.this.mouseEntered)
{
    ...
}
```

Would you expect the condition above to hold true when mouse is entered the widget? I suppose so. Previously mouseEntered set to true didn't really mean "mouse entered the widget" but instead "widget has focus". This was confusing and making the code harder to read and maintain.

3. widget/AbstractContainer.java removed code to find out the menu behind container. How this functionality will now work?

I am glad you point this out. I am a bit worried that I can brake something with this change. I wasn't able to find any valid use case for this and with the menus now being in separate window, this seemed to be redundant. Can you describe the functionality you are referring to?

If the container that holds menu(in OverlayWindow I guess) will run currentFocus() and focus() methods how it will find and return the menu widget reference if there is one?

Yes, but does the container need to return the menu widget from currentFocus()? In what case?

#20 - 06/08/2016 05:33 PM - Vadim Gindin

Did you also test popup_ext.p and system popups (window title popup, scroll popup and editor popup)?

#21 - 06/09/2016 02:38 AM - Hynek Cihlar

Vadim Gindin wrote:

Did you also test popup_ext.p and system popups (window title popup, scroll popup and editor popup)?

Vadim, thanks for your testing efforts.

I tried to run as many tests as I could in uast/menu but apparently I missed some due to high number of failing tests there (I think we should look at these failures).

I have committed more fixes to revision 11065 and I don't see any new failures from running uast/menu tests.

I am also rerunning ChUI regression tests. The changes also touch ChUI code, though they should be safe.

#22 - 06/09/2016 11:36 AM - Hynek Cihlar

2970a 11065 has passed ChUI regression testing, there are just the usual failed tests not related to this issue.

If there is no more input, the branch is ready to be merged to trunk.

#23 - 06/09/2016 11:59 AM - Eugenie Lyzenko

Hynek Cihlar wrote:

Eugenie Lyzenko wrote:

There is no functional difference, the change makes the code more readable.

Consider the following condition from the menu logic.

[...]

Would you expect the condition above to hold true when mouse is entered the widget? I suppose so. Previously mouseEntered set to true didn't really mean "mouse entered the widget" but instead "widget has focus". This was confusing and making the code harder to read and maintain.

...

Yes, but does the container need to return the menu widget from currentFocus()? In what case?

Both can trigger the regression when the menu is first drawing. The mouse pointer will not be over menu, correct? Will the menu consider mouseEntered as true? Will it be focused and having highlight item?

#24 - 06/09/2016 12:18 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

The mouse pointer will not be over menu, correct? Will the menu consider mouseEntered as true?

When mouse pointer is not positioned over a menu item, mouseEntered of the respective item will be false.

Will it be focused and having highlight item?

When an item is focused, it is highlighted.

#25 - 06/09/2016 12:29 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

Eugenie Lyzenko wrote:

The mouse pointer will not be over menu, correct? Will the menu consider mouseEntered as true?

When mouse pointer is not positioned over a menu item, mouseEntered of the respective item will be false.

Will it be focused and having highlight item?

When an item is focused, it is highlighted.

So you mean it is possible to have focused menu with mouseEntered == false, right? And the current implementation does properly handle this case?
If the answer is yes I have no other concerns.

#26 - 06/09/2016 12:42 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

Hynek Cihlar wrote:

Eugenie Lyzenko wrote:

The mouse pointer will not be over menu, correct? Will the menu consider `mouseEntered` as `true`?

When mouse pointer is not positioned over a menu item, `mouseEntered` of the respective item will be `false`.

Will it be focused and having highlight item?

When an item is focused, it is highlighted.

So you mean it is possible to have focused menu with `mouseEntered == false`, right?

Yes.

And the current implementation does properly handle this case?

Yes, the tests show it does.

#27 - 06/09/2016 12:52 PM - Greg Shah

Please merge to trunk.

#28 - 06/09/2016 01:21 PM - Hynek Cihlar

2970a merged to trunk as revision 11042 and archived.

#29 - 06/09/2016 01:22 PM - Hynek Cihlar

- *% Done changed from 0 to 100*

#30 - 06/09/2016 01:22 PM - Greg Shah

- *Status changed from WIP to Closed*

#31 - 11/16/2016 12:22 PM - Greg Shah

- *Target version changed from Milestone 16 to Cleanup and Stabilization for GUI*