

User Interface - Bug #6074

Movable widget problem

02/16/2022 02:37 PM - Vladimir Tsichevski

Status:	Review	Start date:	
Priority:	Normal	Due date:	
Assignee:	Alexandru Lungu	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			
Related issues:			
Related to User Interface - Bug #6077: Movable widget problem, when using bot...			New
Related to User Interface - Bug #6167: Mouse event processing refactoring/imp...			New

History

#2 - 02/16/2022 02:38 PM - Vladimir Tsichevski

- File 6074.mp4 added

The following example defines a movable frame. Users can drag this frame with left mouse button withing the window.

The problem: every **odd** attempt to move frame fails: then the user selects and drags the frame nothing visible happens. See the video 6074.mp4 attached.

The problem exist both in Swing and WEB.

```
DEFINE FRAME movableFrame
  WITH 1 DOWN
  OVERLAY
  SIZE-PIXELS 81 BY 81
  BGCOLOR 8.

ASSIGN FRAME movableFrame:MOVABLE = TRUE.

ON START-MOVE OF FRAME movableFrame
DO:
  MESSAGE "startX: " FRAME movableFrame:X " startY: " FRAME movableFrame:Y.
END.

ON END-MOVE OF FRAME movableFrame
DO:
  MESSAGE "endX: " FRAME movableFrame:X " endY: " FRAME movableFrame:Y.
END.

VIEW FRAME movableFrame.
WAIT-FOR CLOSE OF THIS-PROCEDURE.
```

#3 - 02/16/2022 02:39 PM - Vladimir Tsichevski

- Status changed from New to WIP
- Assignee set to Vladimir Tsichevski

#5 - 02/17/2022 11:41 AM - Vladimir Tsichevski

This problem is a regression after the change 13461.

Alexandru, take a look, please.

I've tried your example movable-frame.p, it seems to work correctly even before the 13461 change. What exactly did not work in this test example before your change?

Your example differs from mine in this issue in that in my example I do **not** set the movable frame as selectable.

#6 - 02/17/2022 02:13 PM - Vladimir Tsichevski

- Related to Bug #6077: Movable widget problem, when using both mouse buttons added

#7 - 02/21/2022 12:40 PM - Alexandru Lungu

- % Done changed from 0 to 100
- Status changed from WIP to Review

3821c/rev. 13461 had some changes regarding selection. If the user presses anywhere in the window, the selected widgets are deselected. 3821c/rev. 13384 was fixing movable-frame.p. The problem at that time was exactly the same you describe here: odd attempts to move the frame fail.

The issue I see here is that by pressing the movableFrame, it gets selected (although it is not selectable). In FWD, in order to move a widget, it must be both selected and movable. After rev. 13461, the frame got automatically deselected because of MouseEmptySelector which wasn't taking in account that the frame was selected because it presumed that it is enough not to be selectable. Added the fix in 3821c/rev. 13549.

#8 - 02/21/2022 01:01 PM - Greg Shah

Vladimir/Eugenie/Hynek: Please review.

#9 - 02/21/2022 01:36 PM - Eugenie Lyzenko

Code review for 13549.

Seems OK. But I have a question.

If the frame is not **selectable** how can it become **selected**?

What if we press inside selected frame that has selected widgets inside? Should not we deselect everything in this case (firing SE_EMPTY_SELECTION)?

#10 - 02/21/2022 02:20 PM - Vladimir Tsichevski

- Assignee changed from Vladimir Tsichevski to Alexandru Lungu

#11 - 02/21/2022 02:22 PM - Vladimir Tsichevski

Greg Shah wrote:

Vladimir/Eugenie/Hynek: Please review.

The change looks good, and it fixes the original issue.

I would extract the ((BaseConfig) widgetEff.config()) expression into a local variable in order not to call it twice in one expression.

#12 - 02/21/2022 03:36 PM - Vladimir Tsichevski

- File 6074-12.mp4 added

Got NPE while testing movable-frame.p here:

```
MouseDirectManipulation.mouseReleased(MouseEvent) line: 630
MouseHandler.processWidgetActions(int, MouseEvent) line: 575
MouseHandler.handleMouseEvent(int, MouseEvent) line: 334
GuiWebDriver(AbstractGuiDriver<F>).handleMouseEvent(int, MouseEvent) line: 3750
WindowManager.processWindowEvent(Event, TopLevelWindow<?>) line: 1936
WindowGuiImpl.processEvent(Event) line: 1615
ThinClient.processProgressEvent(Event) line: 19760
ThinClient.processEventsWorker() line: 19094
...
```

The startPoint field is null here.

To reproduce: click with mouse right of movable frame and drag left to the movable frame.
See the 6074-12.mp4.

#13 - 02/21/2022 03:37 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

Got NPE while testing movable-frame.p here:

This happens with WEB only.

#14 - 02/21/2022 03:41 PM - Vladimir Tsichevski

- File 6074-14.mp4 added

Also, after you drag the frame once, you can drag it by a point outside the frame area: by the point left or below of the frame.

See 6074-14.mp4.

#15 - 02/22/2022 08:21 AM - Vladimir Tsichevski

The change I am working on in #6013 seems to fix [#6074-12](#) and [#6074-14](#).

#16 - 02/22/2022 08:41 AM - Alexandru Lungu

Eugenie Lyzenko wrote:

If the frame is not **selectable** how it can become **selected**?

This was the same reasoning I had when implementing MouseEmptySelector. It seems that, for the moving process, FWD requires the widget to be selected and movable: MouseDirectManipulation.getMovableWidgetsBounds and MouseDirectManipulation.getWidgetsToMove. In this particular case, when dragging, the frame gets selected without being selectable.

What if we press inside selected frame that has selected widgets inside? Should not we deselect everything in this case(firing SE_EMPTY_SELECTION)?

Just tested; in this 4GL scenario no empty selection is fired and the inner widgets are not deselected.

Vladimir Tsichevski wrote:

I would extract the ((BaseConfig) widgetEff.config()) expression into a local variable in order not to call it twice in one expression.

Fixed in 3821c/rev. 13555.

#17 - 02/22/2022 10:03 AM - Vladimir Tsichevski

Alexandru Lungu wrote:

Vladimir Tsichevski wrote:

I would extract the `((BaseConfig) widgetEff.config())` expression into a local variable in order not to call it twice in one expression.

Fixed in 3821c/rev. 13555.

I meant extracting to a variable of the `BaseConfig` type at the beginning of the code: this eliminates one more config reference, two Java class casts and one explicit instanceof usage:

```
@Override
public void mousePressed(MouseEvent e)
{
    final BaseConfig config = widgetEff.config();
    if (isFrame && !config.selectable && !config.selected)
    {
        EventManager.postEvent(new DirectManipulationEvent(widgetEff,
                                                            Keyboard.SE_EMPTY_SELECTION));

        final TitledWindow<?> topWindow = widgetEff.ancestor();
        if (topWindow != null)
        {
            Collection<? extends Widget> selectedWidgets = topWindow.getAllSelectedWidgets();
            for (Widget w : selectedWidgets)
            {
                if (config.selected)
                {
                    EventManager.postEvent(new DirectManipulationEvent(w, UiUtils.locateFrame(w),
                                                                    true, Keyboard.SE_DESELECTION));
                }
            }
        }
    }
}
```

#18 - 02/22/2022 10:25 AM - Alexandru Lungu

Vladimir Tsichevski wrote:

I meant extracting to a variable of the BaseConfig type at the beginning of the code: this eliminates one more config reference, two Java class casts and one explicit instanceof usage:

[...]

Done in 3821c/rev. 13557. Note that the second widget configuration is for the selected widget, not the parent container: if (((BaseConfig) w.config()).selected) instead of if (config.selected).

#19 - 02/22/2022 12:39 PM - Vladimir Tsichevski

Alexandru Lungu wrote:

Vladimir Tsichevski wrote:

I meant extracting to a variable of the BaseConfig type at the beginning of the code: this eliminates one more config reference, two Java class casts and one explicit instanceof usage:

[...]

Done in 3821c/rev. 13557. Note that the second widget configuration is for the selected widget, not the parent container: if (((BaseConfig) w.config()).selected) instead of if (config.selected).

Yes, you are correct here, thanks!

#20 - 03/14/2022 01:31 PM - Vladimir Tsichevski

- Related to Bug #6167: Mouse event processing refactoring/improvement added

Files

6074.mp4	80.6 KB	02/16/2022	Vladimir Tsichevski
----------	---------	------------	---------------------

6074-12.mp4
6074-14.mp4

11.8 KB
271 KB

02/21/2022
02/21/2022

Vladimir Tsichevski
Vladimir Tsichevski