

User Interface - Bug #5564

Prevent unnecessary triggers to destroyed widgets

07/27/2021 11:46 AM - Roger Borrello

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			

History

#2 - 07/27/2021 11:57 AM - Roger Borrello

This patch was added in 3821c revisions 12710 and 12714:

```
=== modified file 'src/com/goldencode/p2j/ui/client/FillIn.java'
--- old/src/com/goldencode/p2j/ui/client/FillIn.java      2021-06-02 18:43:57 +0000
+++ new/src/com/goldencode/p2j/ui/client/FillIn.java      2021-07-22 19:51:36 +0000
@@ -1373,11 +1373,15 @@
     @SuppressWarnings("unchecked")
     Frame<O> frame = (Frame<O>) UiUtils.locateFrame(this);

-    if (frame.isInChoose(this))
+    if (frame == null)
+    {
+        return;
+    }
+    else if (frame.isInChoose(this))
     {
         frame.processKeyEvent(ke);
         return;
     }

     // First call all KeyListener objects that may have been registered
     // for this component.
```

It prevented this crash from occurring:

```
[07/20/2021 14:57:57 BST] (com.goldencode.p2j.util.TransactionManager:SEVERE) Abnormal end; original error:
java.lang.NullPointerException
    at com.goldencode.p2j.ui.client.FillIn.processKeyEvent (FillIn.java:1376)
    at com.goldencode.p2j.ui.client.gui.FillInGuiImpl.processKeyEvent (FillInGuiImpl.java:1078)
    at com.goldencode.p2j.ui.chui.ThinClient.processProgressEvent (ThinClient.java:19943)
    at com.goldencode.p2j.ui.client.FocusManager.focusChange (FocusManager.java:1511)
    at com.goldencode.p2j.ui.chui.ThinClient.nextTabStop (ThinClient.java:22579)
    at com.goldencode.p2j.ui.client.FillIn.processKeyEvent (FillIn.java:1700)
    at com.goldencode.p2j.ui.client.gui.FillInGuiImpl.processKeyEvent (FillInGuiImpl.java:1078)
    at com.goldencode.p2j.ui.client.widget.TitledWindow.processKeyEvent (TitledWindow.java:306)
    at com.goldencode.p2j.ui.client.widget.AbstractWidget.processEvent (AbstractWidget.java:2087)
    at com.goldencode.p2j.ui.client.widget.TitledWindow.processEvent (TitledWindow.java:275)
    at com.goldencode.p2j.ui.client.gui.WindowGuiImpl.processEvent (WindowGuiImpl.java:1590)
    at com.goldencode.p2j.ui.chui.ThinClient.processEventsWorker (ThinClient.java:18862)
    at com.goldencode.p2j.ui.chui.ThinClient.pop (ThinClient.java:17594)
    at com.goldencode.p2j.ui.chui.ThinClient.eventBracket (ThinClient.java:17577)
    at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket (ThinClient.java:17499)
    at com.goldencode.p2j.ui.chui.ThinClient.applyWorker (ThinClient.java:17247)
    at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker (ThinClient.java:13956)
    at com.goldencode.p2j.ui.chui.ThinClient.lambda$waitForWorker$63 (ThinClient.java:13408)
```

```

at com.goldencode.p2j.ui.chui.ThinClient.lambda$doInteractive$82(ThinClient.java:18553)
at com.goldencode.p2j.ui.chui.ThinClient.doInteractive(ThinClient.java:18529)
at com.goldencode.p2j.ui.chui.ThinClient.doInteractive(ThinClient.java:18553)
at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:13407)
at com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:13350)
at com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:13299)
at com.goldencode.p2j.ui.ClientExportsMethodAccess.invoke(Unknown Source)
at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:156)
at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:783)
at com.goldencode.p2j.net.Conversation.block(Conversation.java:422)
at com.goldencode.p2j.net.Conversation.waitForMessage(Conversation.java:348)
at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1213)
at com.goldencode.p2j.net.Queue.transact(Queue.java:673)
at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:271)
at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:211)
at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1473)
at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:145)
at com.sun.proxy.$Proxy12.trigger(Unknown Source)
at com.goldencode.p2j.ui.chui.ThinClient.trigger(ThinClient.java:14655)
at com.goldencode.p2j.ui.chui.ThinClient.invokeTriggers(ThinClient.java:21001)
at com.goldencode.p2j.ui.chui.ThinClient.invokeTriggers(ThinClient.java:20737)
at com.goldencode.p2j.ui.chui.ThinClient.processProgressEvent(ThinClient.java:19899)
at com.goldencode.p2j.ui.chui.ThinClient.processEventsWorker(ThinClient.java:18667)
at com.goldencode.p2j.ui.chui.ThinClient.pop(ThinClient.java:17594)
at com.goldencode.p2j.ui.chui.ThinClient.eventBracket(ThinClient.java:17577)
at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:17499)
at com.goldencode.p2j.ui.chui.ThinClient.applyWorker(ThinClient.java:17247)
at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:13956)
at com.goldencode.p2j.ui.chui.ThinClient.lambda$waitForWorker$63(ThinClient.java:13408)
at com.goldencode.p2j.ui.chui.ThinClient.lambda$doInteractive$82(ThinClient.java:18553)
at com.goldencode.p2j.ui.chui.ThinClient.doInteractive(ThinClient.java:18529)
at com.goldencode.p2j.ui.chui.ThinClient.doInteractive(ThinClient.java:18553)
at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:13407)
at com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:13350)
at com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:13299)
at com.goldencode.p2j.ui.ClientExportsMethodAccess.invoke(Unknown Source)
at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:156)
at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:783)
at com.goldencode.p2j.net.Conversation.block(Conversation.java:422)
at com.goldencode.p2j.net.Conversation.waitForMessage(Conversation.java:348)
at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1213)
at com.goldencode.p2j.net.Queue.transact(Queue.java:673)
at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:271)
at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:211)
at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1473)
at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:145)
at com.sun.proxy.$Proxy10.standardEntry(Unknown Source)
at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:383)
at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:169)
at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:250)
at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:444)
at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:144)
at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:313)
...

```

However, the root cause is most likely that we are still processing events that should have already completed processing before the window dies.

Constantin:

It is possible for a widget/window to get destroyed in a trigger, and for additional events to remain on the event queue, or to be raised for now-dead widgets (like LEAVE or FOCUS events, depending from where in ThinClient the trigger is invoked).

Constantin Asotiei wrote:

The trigger which deletes the FILL-IN is invoked from this stacktrace:

```
$Proxy12.trigger(int, int, int, int, long, int, ScreenBuffer[]) line: not available
ThinClient.trigger(int, int, int, int, long, int, ScreenBuffer, EventList) line: 14699
ThinClient.invokeTriggers(Widget, Widget, long, int, boolean, Integer, Integer) line: 21051
ThinClient.invokeTriggers(Widget, Widget, int, boolean) line: 20787
ThinClient.processProgressEvent(Event) line: 19949
FocusManager.focusChange(Widget, boolean, boolean) line: 1511
ThinClient.nextTabStop(Widget) line: 22629
FillInGuiImpl(FillIn<O,C>).processKeyEvent(KeyInput) line: 1706
FillInGuiImpl.processKeyEvent(KeyInput) line: 1079
WindowGuiImpl(TitledWindow<O>).processKeyEvent(KeyInput) line: 306
FillInGuiImpl(AbstractWidget<O>).processEvent(Event) line: 2087
WindowGuiImpl(TitledWindow<O>).processEvent(Event) line: 275
WindowGuiImpl.processEvent(Event) line: 1590
ThinClient.processEventsWorker() line: 18912
ThinClient.pop() line: 17638
ThinClient.eventBracket(boolean, Runnable) line: 17621
ThinClient.eventDrawingBracket(Widget<?>, boolean, boolean, Runnable) line: 17543
ThinClient.applyWorker(Event) line: 17291
ThinClient.waitForWorker(EventList, int, int, ScreenBuffer[], boolean, boolean, boolean, BlockingOperation, boolean, int) line: 140
```

The NPE is not in another iteration of TC.processEventsWorker(), but in the logic of TC.processProgressEvent.

Maybe a more generic protection should be added to TC.processProgressEvent. But more investigation is needed for this. I would assume that if any of the event's widget (like source and other for KeyInput) gets destroyed by the trigger, to raise some exception which is caught in TC.processEventsWorker, to not allow logic to continue if the event's widget gets destroyed.

### #3 - 07/27/2021 12:03 PM - Roger Borrello

Issue #5552 has been updated by Hynek Cihlar.

Are the key input events already on the event queue, when the widget is being destroyed? If this is the case, then this could also be an issue of ordering of UI events. I.e. frame destroy should be enqueued after those key input events. But this should be experimentally confirmed with the native 4GL environment.

**#4 - 07/27/2021 01:12 PM - Constantin Asofiei**

Roger Borrello wrote:

Issue #5552 has been updated by Hynek Cihlar.

Are the key input events already on the event queue, when the widget is being destroyed? If this is the case, then this could also be an issue of ordering of UI events. I.e. frame destroy should be enqueued after those key input events. But this should be experimentally confirmed with the native 4GL environment.

In this case, the widget gets destroyed when the trigger (for the TAB key input event) is invoked in `ThinClient.processProgressEvent`, and the FILL-IN continues with the processing via the `src.processKeyEvent(evt);`. So we are not in another iteration of `TC.processEventsWorker` (i.e. another event being processed).

**#5 - 07/27/2021 01:25 PM - Hynek Cihlar**

Constantin Asofiei wrote:

Roger Borrello wrote:

Issue #5552 has been updated by Hynek Cihlar.

Are the key input events already on the event queue, when the widget is being destroyed? If this is the case, then this could also be an issue of ordering of UI events. I.e. frame destroy should be enqueued after those key input events. But this should be experimentally confirmed with the native 4GL environment.

In this case, the widget gets destroyed when the trigger (for the TAB key input event) is invoked in `ThinClient.processProgressEvent`, and the FILL-IN continues with the processing via the `src.processKeyEvent(evt);`. So we are not in another iteration of `TC.processEventsWorker` (i.e. another event being processed).

The question is, whether the key event dispatching should be fully finished before the destroy is commenced. I don't know the answer for this, this is something to experiment with.