**Freeciv - Bug #857974**

**Qt client with -t augmentedi dumps core (2.6.1+)**

2020-01-30 01:39 PM - Chippo Elder

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
<th>Start date:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td>Marko Lindqvist</td>
<td></td>
</tr>
<tr>
<td>Category:</td>
<td>Client</td>
<td></td>
</tr>
<tr>
<td>Sprint/Milestone:</td>
<td>2.6.2.1</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

I ran 'freeciv-qt -t augmentedi'. I was starting games with random rulesets and random changes to Game options, and if the game started ok, then quit and try again. I was looking for trouble (like a core dump), found it, then got confused with my own note-taking and can't reproduce it, or be sure what I did when it triggered.

Core was generated by 'freeciv-qt'. Program terminated with signal SIGSEGV, Segmentation fault.

```
Core was generated by `freeciv-qt'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0 0x00007f31eb0a5ee4 in QActionPrivate::sendDataChanged() () from /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   [Current thread is 1 (Thread 0x7f31e53c74c0 (LWP 3553034))]
(gdb) bt full
#0 0x00007f31eb0a5ee4 in QActionPrivate::sendDataChanged() () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#1 0x0000563b00347976 in gov_menu::update() (this=0x563b02558800) at
   /usr/include/x86_64-linux-gnu/qt5/QMenuBar/qmrunique.h:211
   gov = 0x563b05206ab0
   revol_gov = 0x563b05206480
   sprite = <optimized out>
   gov_count = 9
   i = 6
   j = 5
#2 0x0000563b00347b84 in gov_menu::update_all() () at menu.cpp:735
   m = <optimized out>
   container = {c = {q_hash = {{d = 0x563b01d07a0, e = 0x563b01d07a0}}, i = {i = {i = 0x563b01fd2d20}, e = {i = {i =
   0x563b01d07a0}}, control = 1}}}
#3 0x0000563b00358089 in real_menus_update () at menu.cpp:580
   #4 real_menus_update() () at menu.cpp:571
   #5 0x0000563b003580be in qt_start_turn() () at menu.cpp:77
   #6 0x0000563b00340601 in qtn_update_timeout_label() () at mapview.cpp:1086
   #7 0x0000563b003d698b in client_handle_packet (type=type@entry=PACKET_BEGIN_TURN,
      packet=packet@entry=0x563b04ed9010) at packhand_gen.c:276
   #8 0x0000563b0039689f in client_packet_input (packet=packet@entry=0x563b04ed9010, type=128) at client_main.c:770
   FUNCTION = "client_packet_input"
   #9 0x0000563b0039d79c in input_from_server (fd=fd@entry=41) at clinet.c:416
   type = PACKET_BEGIN_TURN
   packet = 0x563b04ed9010
   nb = <optimized out>
   FUNCTION = "input_from_server"
   #10 0x0000563b005445c4 in fc_client::server_input(int) (this=0x563b017f8270, sock=41) at fc_client.cpp:462
   #11 0x00007f31eb0a5ee4 in QMetaObject::activate(QObject*, int, int, void**) () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
   #12 0x00007f31eb897315 in QSocketNotifier::activated(int, QSocketNotifier::QPrivateSignal) () at
      /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
   #13 0x00007f31eb8749f1 in QSocketNotifier::event(QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
   #14 0x00007f31eb0aa8e in QAbstractApplication::notify_helper(QObject*, QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #15 0x00007f31eb09e00 in QAbstractApplication::notify(QObject*, QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #16 0x00007f31eb89e9a9 in QCoreApplication::notifyInternal2(QObject*, QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
   #17 0x00007f31eb8b7305 in () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
   #18 0x00007f31eb87f4d in g_main_context_dispatch () at /usr/lib/x86_64-linux-gnu/libglib-2.0.so.0
   #19 0x00007f31eb87f4d in g_main_context_dispatch () at /usr/lib/x86_64-linux-gnu/libglib-2.0.so.0
   #20 0x00007f31eb87f4d in g_main_context_dispatch () at /usr/lib/x86_64-linux-gnu/libglib-2.0.so.0
   #21 0x00007f31eb8b6a5 in QEventDispatcherGlue::processEvents(QFlags<QEventLoop::ProcessEventFlag>) () at
      /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
   #22 0x00007f31eb85d83b in QEventLoop::exec(QFlags<QEventLoop::ProcessEventFlag>) () at
      /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
   #23 0x00007f31eb8653a6 in QCoreApplication::exec() () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
```

2021-08-08

1/3
2021-08-08

Related issues:
- Related to Freeciv - Bug #857869: Qt SEGV in Cancel of Load Game (2.6.1+) Closed
- Related to Freeciv - Bug #858251: SEGV in name_translation_get (2.6.1+, qt) Closed

History
#1 - 2020-01-30 04:31 PM - Marko Lindqvist
Chippo Elder wrote:

```
#5 0x0000563b003580be in qt_start_turn() () at menu.cpp:77
#6 0x0000563b00340601 in qtg_update_timeout_label() () at mapview.cpp:1086
```

This call-chain is showing up in several of your backtraces in different tickets. I think qt_start_turn() gets called when client state is not suitable for it. I don't know why Qt-client has that qt_start_turn() call in qtg_update_timeout_label(). Other clients do not have equivalent call in their update_timeout_label() function.

#2 - 2020-02-01 01:01 PM - Jacob Nevins
- Related to Bug #857869: Qt SEGV in Cancel of Load Game (2.6.1+) added

#3 - 2020-02-10 11:40 PM - Jacob Nevins
- Related to Bug #858251: SEGV in name_translation_get (2.6.1+, qt) added

#4 - 2020-04-22 01:00 PM - Marko Lindqvist
- Category set to Client
- Status changed from New to In Progress
- Sprint/Milestone set to 2.6.3

Marko Lindqvist wrote:

```
I don't know why Qt-client has that qt_start_turn() call in qtg_update_timeout_label(). Other clients do not have equivalent call in their update_timeout_label() function.
```

I considered patching over that design in S2_6, but it turned out that there's no low-impact way to do it, so I'm going to full clean gui API change even in S2_6. As qt_start_turn() call goes away from qtg_update_timeout_label(), any ticket with backtrace with that call chain should be closed if my patch gets accepted.

#5 - 2020-04-22 02:15 PM - Marko Lindqvist
- File 0022-Fix-Qt-client-crash-on-timeout-label-update.patch added
- File 0010-Fix-Qt-client-crash-on-timeout-label-update.patch added
- File 0006-Fix-Qt-client-crash-on-timeout-label-update.patch added
- Status changed from In Progress to Resolved

Patches that reimplement affected code. In any case this is cleaner implementation. I believe it to also avoid the crashes of the old implementation.

**#6 - 2020-04-25 08:20 PM - Marko Lindqvist**
- Status changed from Resolved to Closed
- Assignee set to Marko Lindqvist

**#7 - 2020-11-16 02:10 AM - Marko Lindqvist**
- Sprint/Milestone changed from 2.6.3 to 2.6.2.1

**Files**

<table>
<thead>
<tr>
<th>Patch Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>0022-Fix-Qt-client-crash-on-timeout-label-update.patch</td>
<td>10.9 KB</td>
<td>2020-04-22</td>
<td>Marko Lindqvist</td>
</tr>
<tr>
<td>0010-Fix-Qt-client-crash-on-timeout-label-update.patch</td>
<td>11.3 KB</td>
<td>2020-04-22</td>
<td>Marko Lindqvist</td>
</tr>
<tr>
<td>0006-Fix-Qt-client-crash-on-timeout-label-update.patch</td>
<td>13.4 KB</td>
<td>2020-04-22</td>
<td>Marko Lindqvist</td>
</tr>
</tbody>
</table>