I was running a separate server and client (qt) and I quit the server with a 'quit' command and got the following dump in the client:

chippo@chippo-Aspire-V3-731:~/Downloads/git_clones/freeciv/freeciv-30$ ASAN_OPTIONS="abort_on_error=1:disable_coredump=0:unmap_shadow_on_exit=1:detect_leaks=0" ./fcgui --gui qt

Running ./client/freeciv-qt

2: Loading tileset "hexemplio".
AddressSanitizer:DEADLYSIGNAL

=================================================================
==1931591==ERROR: AddressSanitizer: SEGV on unknown address 0x00000000e1 (pc 0x000000916834 bp 0x7ffbab81c80 sp 0x7ffbab81c60 T0)
==1931591==The signal is caused by a READ memory access.
==1931591==Hint: address points to the zero page.
#0 0x916833 in player_has_real_embassy /home/chippo/Downloads/git_clones/freeciv/freeciv-30/common/player.c:214:10
#1 0x9167cc in player_has_embassy /home/chippo/Downloads/git_clones/freeciv/freeciv-30/common/player.c:203:14
#2 0x729bf8 in plr_widget::nation_selected(QItemSelection const&, QItemSelection const&) /home/chippo/Downloads/git_clones/freeciv/freeciv-30/client/gui-qt/plrdlg.cpp:454:11
#3 0x7962f4 in plr_widget::qt_static_metacall(QObject*, QMetaObject::Call, int, void**) /home/chippo/Downloads/git_clones/freeciv/freeciv-30/client/gui-qt/meta_plrdlg.cpp:313:21
#4 0x7fb3f5a753f7 in QMetaObject::activate(QObject*, int, int, void**) (/usr/lib/x86_64-linux-gnu/libQt5Core.so.5+0x2b13f7)
#5 0x7fb3f5a050c3 in QItemSelectionModel::selectionChanged(QItemSelection const&, QItemSelection const&) (/usr/lib/x86_64-linux-gnu/libQt5Core.so.5+0x2410c3)
#6 0x7fb3f5a0a501 in QItemSelectionModel::select(QModelIndex const&, QFlags<QItemSelectionModel::SelectionFlag>) (/usr/lib/x86_64-linux-gnu/libQt5Core.so.5+0x243300)
#7 0x7fb3f5a07531 in QItemSelectionModel::setCurrentIndex(QModelIndex const&, QFlags<QItemSelectionModel::SelectionFlag>) (/usr/lib/x86_64-linux-gnu/libQt5Core.so.5+0x243531)
#8 0x7fb3f54fca2c in update_unqueue /home/chippo/Downloads/git_clones/freeciv/freeciv-30/client/update_queue.c:320:5
#9 0x6b87e8 in QtPrivate::QSlotObject<void (mr_idle::*)(), QtPrivate::List<>, void>::impl(int, QtPrivate::ObjectBase*, QObject*, void*, bool) (/usr/include/x86_64-linux-gnu/qt5/QtCore/qobjectdefs_impl.h:185:13)
#10 0x6b8707 in QtPrivate::QSlotObject<void (mr_idle::*)(), QtPrivate::List<>, void>::impl(int, QtPrivate::ObjectBase*, QObject*, void*, bool) (/usr/include/x86_64-linux-gnu/qt5/QtCore/qobjectdefs_impl.h:141:17)

The dump was taken at the line player_has_real_embassy where it says:"#0 0x916833 in player_has_real_embassy /home/chippo/Downloads/git_clones/freeciv/freeciv-30/common/player.c:214:10".

I'm not sure what exactly is the problem. It seems to be related to the server and the client quitting, but I'm not sure how to fix it. If you need any more information, please let me know.
sigs = {__val = {32, 0 <repeats 15 times>}}

#2 0x000000000004fa427 in ()
#3 0x000000000004f8e01 in ()
#4 0x00000000004e0989 in ()
#5 0x00000000004e0613 in __asan::ReportDeadlySignal(__sanitizer::SignalContext const&) ()
#6 0x00000000004dffc3 in __asan::AsanOnDeadlySignal(int, void*, void*) ()
#7 0x00000000004f8e01 in ()
#8 0x00000000004e0989 in ()
#9 0x00000000004e0613 in __asan::ReportDeadlySignal(__sanitizer::SignalContext const&) ()
#10 0x000000000004fa427 in ()

(index = {r = 539352, c = 24992, i = 8460121226083921741, m = 0x72})

qvar = {d = {data = {c = -128 '\200', uc = 128 '\200', s = 1152, sc = -128 '\200', us = 1152, i = 132224, u = 132224, l = 107614700700800, ul = 107614700700800, b = 128, d = 5.3168726603753199e-310, f = 1.85285288e-40, real = 5.3168726603753199e-310, ll = 107614700700800, ull = 107614700700800, o = 0x61e000020480, ptr = 0x61e000020480}, type = 31, is_shared = 0, is_null = 0}}

indexes = {<QListSpecialMethods<QModelIndex>> = {<No data fields>}, {p = {static shared_null = {ref = {atomic = {_q_value = {<std::__atomic_base<int>> = {static _S_alignment = 4, _M_i = -1}, <No data fields>}}}, alloc = 0, begin = 0, end = 0, array = (0x0)}, d = 0x6110006c2900}, d = 0x6110006c2900}}

res = {static null = {<No data fields>}, d = 0x7fb3f5b2a680 <QArrayData::shared_null>}

sp = {static null = {<No data fields>}, d = 0x6030000d00f0}

etax = {static null = {<No data fields>}, d = 0x7fb3f5b2a680 <QArrayData::shared_null>}
esci = {static null = {<No data fields>}, d = 0x7fb3f5b2a680 <QArrayData::shared_null>}
elux = {static null = {<No data fields>}, d = 0x7fb3f5b2a680 <QArrayData::shared_null>}
egold = {static null = {<No data fields>}, d = 0x7fb3f5b2a680 <QArrayData::shared_null>}
egov = {optimized out}
cult = {optimized out}
nl = {optimized out}

sorted_list_a = {<QList<QString>> = {<QListSpecialMethods<QString>> = {<No data fields>}, (p = {static shared_null = {ref = {atomic = {_q_value = {<std::__atomic_base<int>> = {static _S_alignment = 4, _M_i = -1}, <No data fields>}}}, alloc = 0, begin = 0, end = 0, array = (0x0)}, d = 0x6110006c2900)}, d = 0x7fb3f5b2c760 <QListData::shared_null>}, d = 0x7fb3f5b2c760 <QListData::shared_null>}}, <No data f ields>}

sorted_list_b = {<QList<QString>> = {<QListSpecialMethods<QString>> = {<No data fields>}, (p = {static shared_null = {ref = {atomic = {_q_value = {<std::__atomic_base<int>> = {static _S_alignment = 4, _M_i = -1}, <No data fields>}}}, alloc = 0, begin = 0, end = 0, array = (0x0)}, d = 0x7fb3f5b2c760 <QListData::shared_null>}, d = 0x7fb3f5b2c760 <QListData::shared_null>}, d = 0x7fb3f5b2c760 <QListData::shared_null>}}, <No data f ields>}

entry_exist = false

pplayer = 0xe1

me = 0x0

research = 0x16747f4 <research_array+97812>

pcity = {optimized out}

added = {optimized out}

state = {optimized out}

tech_id = {optimized out}

a = {optimized out}

b = {optimized out}

my_research = {optimized out}

#11 0x000000000007962f5 in plr_widget::qt_static_metacall(QObject*, QMetaObject::Call, int, void**) (_o=0xex, _c=<optimized out>, _id=<optimized out>, _a=0x1l) at meta_plrdlg.cpp:313

_t = 0x0el
mr_idle::idling() after the client is supposedly left the game. It tries to execute callbacks registered while the game was still running, and which try to update widgets like it still was in a game.

Unfortunately there's no easy way to implement "catch all" solution. It seems we need to make necessary checks, per ticket, inside the running callbacks.

Did you happen to be global observer at the time? For this crash to happen client can't be attached to a player. That can happen at least when one is global observer. I'm not sure if it can happen because client is already leaving the game (if you weren't global observer, it would be the only explanation).

The crash also required player with no research selected to be selected in the player dialog. The fact that crash happens just when client is leaving the game is probably explained by removal of the other players from the player dialog. When previously selected player is removed, new one gets to be the selected one, and eventually that hits player that has no research selected.

Attached patch should fix at least the crash seen in the backtrace. There's no guarantees it doesn't still crash later in the codepath.

Attached patch should fix at least the crash seen in the backtrace. There's no guarantees it doesn't still crash later in the codepath.