in *fill_grid_sprite_array()* [tilespec.c::5181]: assertion 't->sprites.player[plrid].grid_borders [pedge->type][1] != NULL' failed (2.6.1+, gtk3.22, civ2civ3_earth)

2020-01-31 08:27 AM - Chippo Elder

**Status:** Closed  **Start date:**

**Priority:** Normal  **Due date:**

**Assignee:** Jacob Nevins  **% Done:** 0%

**Category:** Client  **Estimated time:** 0.00 hour

**Sprint/Milestone:** 2.6.2

**Description**

It happened as I expected to see the city dialog for my first city appear.

2: Loading tileset "amplio2".
2: Loading tileset "delta2".
2: Loading tileset "amplio2".
2: Loading tileset "amplio_earth".

[New Thread 0x7fffca7fc700 (LWP 226314)]
[Thread 0x7fffca7fc700 (LWP 226314) exited]
[New Thread 0x7fffca7fc700 (LWP 226358)]

0: in *fill_grid_sprite_array()* [tilespec.c::5181]: assertion 't->sprites.player[plrid].grid_borders [pedge->type][1] != NULL' failed.
0: Please report this message at https://www.hostedredmine.com/projects/freeciv

Thread 1 "freeciv-gtk3.22" received signal SIGABRT, Aborted.
raise (sig=<optimized out>) at ../sysdeps/unix/sysv/linux/raise.c:50
50  ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) bt full
#0  raise (sig=<optimized out>) at ../sysdeps/unix/sysv/linux/raise.c:50
#1  0x000055555555fede in fc_assert_fail
(file=entry=0x555555577ec1 "tilespec.c", function=function@entry=0x5555555781d40 <__FUNCTION__.32224>
"fill_grid_sprite_array", line=line@entry=5181, assertion=assertion@entry=0x5555555781d40 "t->sprites.player[plrid].grid_borders [pedge->type][1] != NULL", message=0x55555557ad4ec "nologmsg:%%") at log.c:523
level = LOG_FATAL
#2  0x000055555555fede in fc_assert_fail
(file=entry=0x555555577ec1 "tilespec.c", function=function@entry=0x5555555781d40 <__FUNCTION__.32224>
"fill_grid_sprite_array", line=line@entry=5181, assertion=assertion@entry=0x5555555781d40 "t->sprites.player[plrid].grid_borders [pedge->type][1] != NULL", message=0x55555557ad4ec "nologmsg:%%") at log.c:523
#3  0x000055555555fede in fc_assert_fail
(file=entry=0x555555577ec1 "tilespec.c", function=function@entry=0x5555555781d40 <__FUNCTION__.32224>
"fill_grid_sprite_array", line=line@entry=5181, assertion=assertion@entry=0x5555555781d40 "t->sprites.player[plrid].grid_borders [pedge->type][1] != NULL", message=0x55555557ad4ec "nologmsg:%%") at log.c:523

2021-08-09 1/5
# History

## #1 - 2020-02-01 12:59 PM - Jacob Nevins

So, we were trying to draw borders, and we couldn't find an appropriate border sprite for one of the players. (The second one, but maybe the first was NULL.) And this happened when we were popping up a city dialog (which I think ought not to show any terrain that wasn't already on the main map?)

This rings a faint bell. One way this can happen is if something goes wrong with producing player-specific coloured border sprites from the generic ones when player colours are initialised. Haven't got proof that that went wrong here. 

Haven't tried reproducing yet, but I've played a fair amount of civ2civ3_earth in the past and don't recall encountering trouble here.

## #2 - 2020-02-02 11:40 PM - Marko Lindqvist

This is trivially reproducible. civ2civ3_earth ruleset + tileset does not show any borders.

Jacob Nevins wrote:

And this happened when we were popping up a city dialog (which I think ought not to show any terrain that wasn't already on the main map?)

I think the thing here is that there's no territory, and thus no borders, before the first city is founded. As soon as one founds the city, borders should appear.

## #3 - 2020-02-02 11:58 PM - Marko Lindqvist

This only happens if one loads amplio_earth tileset from the ruleset's suggested tileset dialog. It does not happen if client is started with -t amplio_earth, nor does it happen if one refuses to load amplio_earth from the dialog but changes the tileset only once in game.

## #4 - 2020-02-05 11:01 PM - Jacob Nevins

Yes, now I can reproduce this from cazfi's instructions. (For completeness: with freeiciv-gtk3.22, and selecting the ruleset from the client dropdown rather than with "-r" or anything; I haven't checked if those things are necessary.)

## #5 - 2020-02-05 11:07 PM - Jacob Nevins

I noticed in passing recently that the client pops up the tileset dialog on receiving the PACKET_RULESET_CONTROL packet (start of ruleset data), whereas PACKET_RULESETS_READY (end of ruleset data) feels like it would be safer? Maybe that's relevant? Haven't looked in detail.

## #6 - 2020-02-05 11:51 PM - Jacob Nevins

tilespec_reread() is protected from the most obvious trouble of running at this point by a check for game.client.ruleset_ready, FALSE in this situation.

I feel the code path involved (game.client.ruleset_ready FALSE, game_fully_initialized FALSE) might be unique to the situation of loading a ruleset-preferred tileset, so it's plausible there's a bug specific to it.
So, the special conditions prevent tilespec_reread() calling tileset_player_init() for all players; and it is not consistently called later.

It is called for a few later-numbered players (but not the human), as a side effect of receiving PACKET_PLAYER_INFO. I'm guessing (but haven't proved) that doesn't happen for all players because PLAYER_INFO is an is-info packet, and nothing changed about the player from the server side.

The Qt client allows you to change the current tileset from the pregame screen, which feels like it ought to be an analogous situation.

But it doesn't run into the same problem, because it passes game_fully_initialized == client.conn.established, which is TRUE at this point. So it completely reinitialises tileset stuff from the player colours.

So, perhaps the answer is simply to pass game_fully_initialized as TRUE when switching tileset because of ruleset preference. It appears to mean something like "do we have a network connection", and if we're thinking about a specific ruleset we must do.

(Probably this is a spot fix we can apply for the next release, but it feels like we ought to be able to reduce the number of code paths and boolean conditions involved here...)

Perhaps the answer is simply to pass game_fullyInitialized as TRUE when switching tileset because of ruleset preference

There was one other minor tweak needed to avoid a crash.

This seems to work for me. (I also tested with auto-accept.) I haven't thoroughly tested other tileset-loading scenarios but I didn't break it on a quick play around.

The patch mostly, but not entirely, only affects the ruleset-overriding-tileset case.

There was another minor tweak needed to avoid a crash.

This seems to work for me. (I also tested with auto-accept.) I haven't thoroughly tested other tileset-loading scenarios but I didn't break it on a quick play around.

The patch mostly, but not entirely, only affects the ruleset-overriding-tileset case.

...this area was last touched for gna bug #21231, and my patch more or less reverts that. But I think/hope there have probably been rearrangements since then that make my version safe (gna bug #21116, sort of thing).

This seems to work for me. (I also tested with auto-accept.) I haven't thoroughly tested other tileset-loading scenarios but I didn't break it on a quick play around.

The patch mostly, but not entirely, only affects the ruleset-overriding-tileset case.

There was another minor tweak needed to avoid a crash.

This seems to work for me. (I also tested with auto-accept.) I haven't thoroughly tested other tileset-loading scenarios but I didn't break it on a quick play around.

The patch mostly, but not entirely, only affects the ruleset-overriding-tileset case.