Subject: Raster::Line segfaults ... sometimes.
Posted by rainbowsally on Mon, 22 Dec 2014 07:04:36 GMT
View Forum Message <> Reply to Message

Adding a static counter somehow fixed the issue (for me). And then considering that something might be choking on the cpu usage in this thread, I tried a 'usleep()' which also works, and seems to be more consistent.

I am multithreading in linux. You might not have this problem.

In uppsrc/Draw/Raster.cpp I was getting seg faults ("illegal memory access") so I put a counter in the function noted below. And just adding a static counter fixed the problem.

Huh????

[Update: Well, it fixed it a little more than half the time.]

If you are having trouble with RasterTest try this. I can't see how it could do anything but add a delay (unless there's a bug in BLITZ), but I don't want to experiment any more with this at this time.

File: uppsrc/Draw/RasterLine.cpp

```
void Raster::Line::MakeRGBA() const
{
// -rs added these two lines
//  static int cnt;
//  cnt++;
// -rs replaced the above with this ONE line.. another "experiment".
    usleep(500); // half a millisecond

 ASSERT(fmtdata && raster);
 int cx = raster->GetWidth();
// ...
```

Tested, 5 times, no crashes. I'm done for now.

The half millisecond delay gives the OS plenty of time to pass a time slice to other threads. Dunno what's causing this or where the best place to deal with this is, but it might be that a "io pause" something like this should be in the main event loop. At least in Linux.

May look into this later.