
Subject: : AmigaOS4

Topic: : The MiniGL thread

Re: The MiniGL thread

Author: : Daytona675x

Date: : 2019/3/26 13:38:45

URL:

@Raziel

Okay, as far as I can see the following happens:

1. if there's an NPOT texture incoming, then the game creates an **empty** texture with the next power of two edge size. So e.g. a 200x100 texture would become 256x128. So far so good.
2. but the game doesn't scale the texture data accordingly but simply puts the tex-data unscaled at position 0,0.
3. and because the texture was created as an empty texture, a huge part of it is actually undefined!

Now take all that and add a slightly false texture-coordinate generation, voila: you have a randomly colored (fully black or white are eventually most likely depending on driver) line like the one you see.

And yes, the wrap-mode doesn't really have the expected impact here, because we are in mid of the texture data, not at the edges (although a GL_REPEAT most likely makes things even worse here).

You could now try to locate the texture coordinate generation code for that sky-cube and fix it.

Or as a simple slow and dirty workaround you could try adding the following code after line 96 - and eventually improve it later 😊

```
if(!OpenGLContext.NPOTSupported) {
    if(rect.left==0 && rect.top==0 && rect.right==surface->w && rect.bottom==surface->h && (internalWidth!=width || internalHeight!=height)) {
        glTexSubImage2D(GL_TEXTURE_2D,0,1,1,surface->w,surface->h,internalFormat,sourceFormat,surface->getPixels());
        glTexSubImage2D(GL_TEXTURE_2D,0,1,0,surface->w,1,internalFormat,sourceFormat,surface->getPixels());
        glTexSubImage2D(GL_TEXTURE_2D,0,0,1,surface->w,surface->h,internalFormat,sourceFormat,surface->getPixels());
    }
}
```