Subject: : AmigaOS4
Topic: : RadeonHD 3.x bug (?): more than ~256mb of used GPU (second chunk) memory cause a heavy lockup/crash.
Re: RadeonHD 3.x bug (?): more than ~256mb of used GPU (second chunk) memory cause a heavy lockup/crash.
Author: : kas1e
Date: : 2019/9/28 8:28:30
URL:

@All

An another progress: I create just pure MiniGL example which also fill the GPU memory with little textures , and it also the same happy crash/lockup. There is test code for anyone to try, compile it like "gcc test.c -o test -lGL -lauto"

```
 #include <stdlib.h>
#include <GL/gl.h>

#include <proto/exec.h>
#include <proto/intuition.h>



GLuint MakeTexture(void)
{

// 100 : eat 34 mb of GPU
// 200 : eat 68 mb of GPU
// 300 : eat 101 mb of GPU
// 400 : eat 135 mb of GPU
// 500 : eat 168 mb of GPU
// 600 : eat 202 mb of GPU
// 700 : eat 235 mb of GPU
// 750 : eat 252 mb of GPU

// 764 : fill 256 mb of GPU , but not overbound to 257 mb at moment
// 765 : now we cross the line of 256 mb => CRASH

   int num_of_textures = 764;

   int width = 256;
   int height = 256;
   unsigned char *data;
```

```
    data = malloc(width*height*3);

    GLuint TextureID[num_of_textures];

    glEnable(GL_TEXTURE_2D);

    for(int a=0;a<num_of_textures;a++)
    {
        glGenTextures( 1, &TextureID[a]);
        glBindTexture( GL_TEXTURE_2D, TextureID[a]);
        glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width, height,0, GL_RGB, GL_UNSIGNED_BYTE, data);

    }

    free(data);
}



struct Interface *getInterface(char *libname)
{
    struct Library *lib;

    lib = IExec->OpenLibrary(libname, 0);
    if (lib)
        return IExec->GetInterface(lib, "main", 1, NULL);

    return 0;
}

void dropInterface(struct Interface *iface)
{
    if (iface)
    {
        struct Library *lib = iface->Data.LibBase;
        IExec->DropInterface(iface);
        IExec->CloseLibrary(lib);
    }
}



/*  Main Loop
 *  Open window with initial window size, title bar,
 *  RGBA display mode, and handle input events.
 */
int main(int argc, char** argv)
{
    struct MiniGLIFace *IMiniGL = 0;
    IMiniGL = (struct MiniGLIFace *)getInterface("minigl.library");
    if (!IMiniGL)
```

```
    return 20;

  struct GLContextIFace *IGL = IMiniGL->CreateContextTags(
        MGLCC_Width,        500,
        MGLCC_Height,        200,
        MGLCC_Windowed,        TRUE,
        MGLCC_CloseGadget,    TRUE,
        MGLCC_Buffers,         2,
        MGLCC_PixelDepth,    16,
        TAG_DONE);
  if (IGL)
  {
    mglMakeCurrent(IGL);

    glViewport(0,0, 500, 200);

    MakeTexture();

    BOOL bRun = TRUE;

    struct Window *win = mglGetWindowHandle();
    if (!IIntuition->ModifyIDCMP(win, IDCMP_CLOSEWINDOW))
      bRun = FALSE;

    while (bRun)
    {
      struct IntuiMessage *imsg;

      glFlush();

      mglSwitchDisplay();

      struct Window *win = mglGetWindowHandle();

      while ((imsg = (struct IntuiMessage *)IExec->GetMsg(win->UserPort)))
      {
        if (imsg->Class == IDCMP_CLOSEWINDOW)
          bRun = FALSE;
        IExec->ReplyMsg((struct Message *)imsg);
      }
    }

    mglDeleteContext();
  }

  dropInterface((struct Interface *)IMiniGL);

  return 0;
}
```

~~At least on my setup when i fill GPU memory by 764 textures, i fill whole 256 mb, but didn't cross the line. Once i~~ put 765 textures (so, it need a little bit more than 256mb), then crash lockup.

Will be very intersting if anyone can try the same on X1000. And just put there, let's say 1000 textures, and watch in Sysmon how many memory it will eat. Then try 2000 and so on, to see, when it start to crash on x1000 (if it will not crash when cross 256mb line).

I also tried to run some minigl games, till the moment when i fill 256mb chunk in GPU memory, and the same crash.

So taking in account that it happens and with minigl and with ogles2, we can made a safe assumtion that its or Warp3dNova, or , RadeonHD. And issue probably RadeonHD as this is the code which break those 256mb bariers.

I talk to ptitSeb about, he of course can't know how it all done, but he have a guess, that it is some limit in the GPU memory to CPU memory mapping.

Now need to know details about x1000 tests of that minigl example (so everyone can compile it). I.e. will it crash when you fill more than 256 mb, and if not, will it crash when fill 512mb, or when.