
Subject: : AmigaOS4

Topic: : GL4ES: another OpenGL over OpenGL ES2 emulation - some tech. info and porting progress

Re: GL4ES: another OpenGL over OpenGL ES2 emulation - some tech. info and porting progress

Author: : Daytona675x

Date: : 2019/9/20 14:24:19

URL:

@thellier

No, that's not the way to go. The second parameter to `VBOSetArray` must be the index of an array of the respective VBO, namely a value between 0 and $N-1$, where N is the value you used when creating your VBO. Also, your 40 is not the VBO size, it's your vertex-size, your VBO certainly contains more than 1 vertex 😊

In your example N is 3, but you try to falsely set array-layouts for non-existing arrays 3 to 39.

Keep in mind that for this trick the *real* layout of your arrays is not of interest.

What's important is that you use `W3DNEF_UINT8` (otherwise endian conversion would kick in) and the same size (otherwise the driver acts somewhat dumb and always selects complex-slow-copy, which is why you eventually have to add some extra bytes to your VBO (and I suppose an 8 byte divisible size won't hurt neither)) for all arrays of the VBO.

The idea simply is to split the VBO memory temporarily into N sequential raw-byte areas of the same size.

Let's assume your VBO should contain 4 of your vertices. Then it looks like this:

```
ArrayCount=3 (xyz, uvw, rgba)
```

```
VertexSize=40
```

```
VBOsize=4*VertexSize=160
```

But for the trick to work we must ensure that the VBO size is divisible by our ArrayCount, therefore:

```
if(VBOsize % ArrayCount) VBOsize+=(ArrayCount-(VBOsize % ArrayCount)), so  
VBOsize=162
```

So you will firstly create a VBO with 3 arrays and byte-size 162.

In reality your buffer will look like this:

```
..0: xyzuvwrgba  
.40: xyzuvwrgba  
.80: xyzuvwrgba  
120: xyzuvwrgba  
160: bb  
162:
```

However, to make the anti-endian-conv-trick you temporarily make it appear like this though:

.0: bbb
.54: bbb
108: bbb
162:

which is
for(uint32 i=0;i<N;++i) VBOSetArray(vbo_handle,i,W3DNEF_UINT8,FALSE,1,1,i*(vbo_size/N),vbo_size/N);
or here
for(uint32 i=0;i<3;++i) VBOSetArray(vbo_handle,i,W3DNEF_UINT8,FALSE,1,1,i*54,54);