

Subject: : AmigaOS4

Topic: : Shaderjoy 1.21

Re: Shaderjoy 1.16

Author: : kas1e

Date: : 2020/10/12 19:29:04

URL:

@Capehill

Quote:

Ps. Can you reproduce issue 616?

No, works on my side..

But, maybe you remember, there were one shader, which works till ShaderJoy 1.8, and stop working on all versions since 1.9 :

<https://www.shadertoy.com/view/WIBXzy>

```
#define AREA 60
```

```
float r1D(float r)
```

```
{  
    return fract(sin(r*12.43)*519803.43);  
}
```

```
float rand(vec2 p)
```

```
{  
    return fract(sin(dot(p,vec2(12.43,98.21))))*519803.43);  
}
```

```
float map(vec2 v, float s, vec2 z)
```

```
{  
    if (z.x >= 0. && z.y >= 0.)  
    {  
        v*=z;  
        v*=s;  
        return rand(floor(v));  
    }  
}
```

```

if (z.x < 0. && z.y >= 0.)
{
    v*=z;
    v*=s;
    return rand(vec2(ceil(v.x-1.), floor(v.y)));
}
if (z.x < 0. && z.y < 0.)
{
    v*=z;
    v*=s;
    return rand(ceil(v-1.));
}
if (z.x >= 0. && z.y < 0.)
{
    v*=z;
    v*=s;
    return rand(vec2(floor(v.x), ceil(v.y-1.)));
}
}

```

```
float grid2D(vec3 ro, vec3 rd, vec2 st)
```

```

{
    vec3 p;
    vec3 u=ro;

    int j = 1;
    for (int i=0;i<AREA;i+=0)
    {
        if (j > AREA)
            break;
        float h=0.;
        if (rd.x != 0.)
        {
            p.x=floor(u.x)+1.;
            h=(p.x-u.x)/rd.x;
            p.y=h*rd.y+u.y;
        }
        else p.y=floor(u.y)+1.;
        if (p.y>=floor(u.y)+1.)
        {
            p.y=floor(u.y)+1.;
            h=(p.y-u.y)/rd.y;
            p.x=h*rd.x+u.x;
        }
        else{j++;}

        float g=(p.x-u.x)/rd.x;
        p.z=g*rd.z+u.z;
        float t=map(u.xy, 1., sign(st))*abs(sin(iTime)*rand(floor(u.xy))+1.);
        if (u.z<t)
        {

```

```

return length(u-ro);
break;
}
if (p.z<t)
{
float z = p.z-t;
h=z/rd.z;
float x = h*rd.x;
h=x/rd.x;
float y = h*rd.y;
u = p-vec3(x,y,z);
return length(u-ro);
}
u=p;
}
return 0.;
}

```

```
vec3 normal(vec3 p, vec2 st)
```

```

{
float d = length(p);
vec2 e = vec2(.01, 0.);
vec3 n = vec3(map(p.xy-e.xy,1.,sign(st))-map(p.xy+e.xy,1.,sign(st)),
map(p.xy-e.yx,1.,sign(st))-map(p.xy+e.yx,1.,sign(st)),
e.x);
return normalize(n);
}

```

```
#define B vec3(.6,.1,1.)
```

```
#define G vec3(.6,1.,.1)
```

```
void mainImage( out vec4 fragColor, in vec2 fragCoord )
```

```

{
vec2 uv = fragCoord/iResolution.xy*2.-1.;
uv.x*=iResolution.x/iResolution.y;

vec2 st=uv;
vec3 ro=vec3(0.,0.,10.);
vec3 rd=normalize(vec3(abs(st),-1.));

vec3 col=vec3(0.);

float d = grid2D(ro, rd, st);
vec3 p = ro+rd*d;
vec3 n = normal(p,st);
vec3 l = vec3(cos(iTime),sin(iTime),.25)*10.;
vec3 ld = normalize(l-p);
float diff = max(dot(ld,n),0.);
col += diff*p.z+(p.z+1)*.1;
fragColor = vec4(sqrt(col*mix(B,G,p.z*1.5)), 0.);
}

```

Maybe now it's the same issues ? I mean check with previous versions, like 1.16 (at least on my x5000/RadeonRX it bars.frag works on 1.16 version of shaderjoy)

Maybe it can be some memory trashing somewhere which cause some bad data to be send to GPU ? At least that how i can explain why shader which i post works till 1.8 and start freezes after.

If that will be common case, then we can think about what cause it :)