Oregon State University

CS_557_X001_W2022 COMPUTER GRAPHICS SHADERS

Project #6

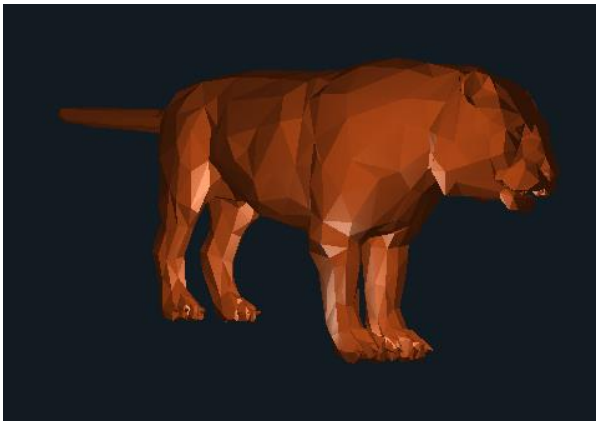
Professor: Mike Bailey
Student: Chengxu Xu
(xucheng@oregonstate.edu)

For the coolest tiger, my idea was to try to simulate a real-world tiger pattern and add some interesting parts like a broken tiger. Here I used uStripes to differentiate the range of stripes and broken effects, and used the previous uNoiseAmp and uNoiseFreq to implement irregular stripes to simulate the real world tiger look as much as possible.

Kaltura link: https://media.oregonstate.edu/media/t/1_b5bkw7p3

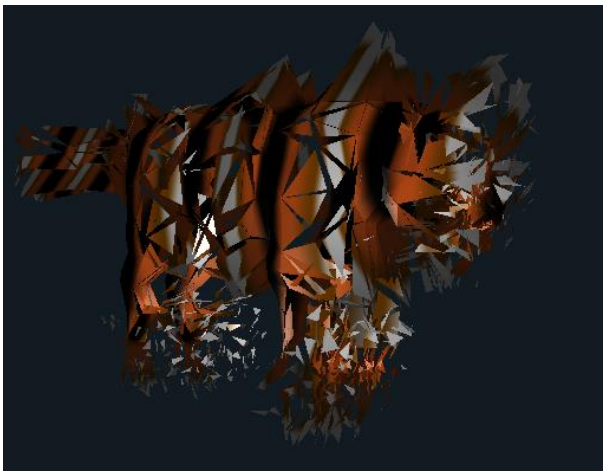
Screen Shots：

Original



adjust uStrips

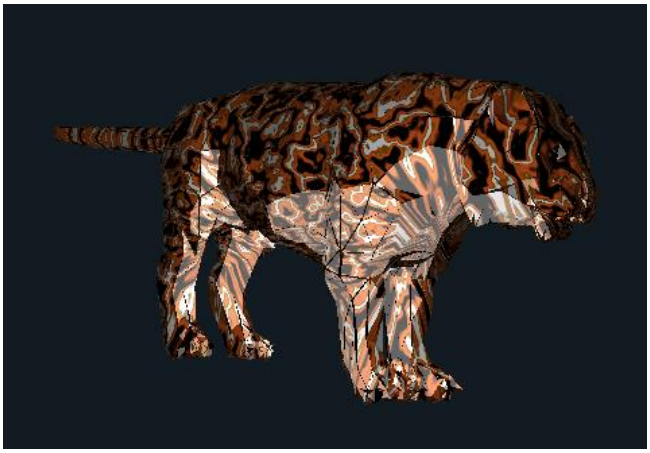
adjust uSkinSmmothness



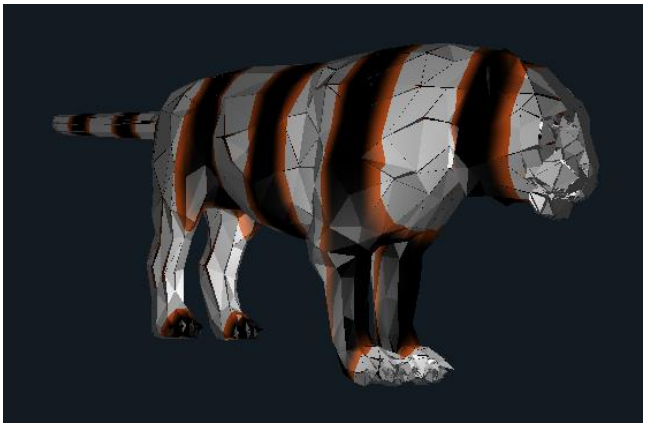
adjust uNoiseAmp

adjust uNoiseFreq

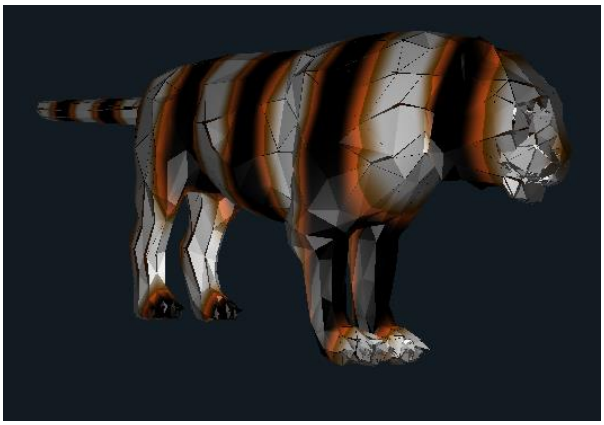
adjust uShininess



adjust WhiteStrips



adjust WhiteBlend



adjust OrangeStrips

adjust OrangeBlend

adjust StripMix

Key snippets:

Special parameter predefined:

```
uNoiseAmp <0. 0. 3.>        \
uNoiseFreq <0. 1. 10.>      \
uStripes <0. 0. 30.>        \
```

```
WhiteStrips <0. .2 1.>      \
WhiteBlend <0. .4 1.>       \
OrangeStrips <0. .4 1.>     \
OrangeBlend <0. .8 1.>      \
StripMix <0. .4 1.>
```

Color and stripe interval predefined:

```
const vec3 blackColor = vec3(0.,0.,0.);
const vec3 orangeColor = vec3(250./255.,100./255.,30./255.);
const vec3 mudColor = vec3(150./255.,80./255.,0.);
const vec3 whiteColor = vec3(1.,1.,1.);
uniform float WhiteStrips = .1;
uniform float WhiteBlend = .3;
uniform float OrangeStrips = .45;
uniform float OrangeBlend = .65;
uniform float StripMix = .4;
```

Adjust different stripe intervals and their colors with uStripes:

```
float dist = abs(int((vST.t + n) *uStripes) +.5 - (vST.t + n) * uStripes);
if(int((vST.t + n) * uStripes) %2 == 0  ){
    //distance from the center, goes between 0. -> .5
    dist*=2.;
    if(dist <= WhiteStrips){
        finalColor = whiteColor;
    }else if(dist > WhiteStrips && dist <= WhiteBlend){
        float t = smoothstep(WhiteStrips,WhiteBlend,dist);
        finalColor = mix(whiteColor,mudColor,t);
    }else if(dist > WhiteBlend && dist <= OrangeStrips){
        finalColor = mudColor;
    }else if(dist > OrangeStrips && dist <= OrangeBlend){
        float t = smoothstep(OrangeStrips,OrangeBlend,dist);
        finalColor = mix(mudColor,orangeColor,t);
    }else{
        finalColor = orangeColor;
    }
```

Adjusting the reflective part with uKa, uKd and uKs:

```
vec3 ambient = uKa * finalColor;
float d = max( dot(Normal,Light), 0. );
vec3 diffuse = uKd * d * finalColor;
float s = 0.;
if( dot(Normal,Light) > 0. ){ // only do specular if the light can see the point
    vec3 ref = normalize( 2. * Normal * dot(Normal,Light) - Light );
    s = pow( max( dot(Eye,ref),0. ), uShininess );
}
vec3 specular = uKs * s * uSpecularColor.rgb;

gl_FragColor = vec4( ambient.rgb + diffuse.rgb + specular.rgb, 1. );
```

Irregular stripes with uNoiseAmp and uNoiseFreq:

```
vec4 nv = texture(Noise2,uNoiseFreq * vST);
float n = nv.r + nv.g + nv.b + nv.a;     //
n = n - 2.;                              //
n*= uNoiseAmp;
```

Wave breaking effect by combining uStripes and uSkinSmoothness:

```
vec4 newPosition = gl_Vertex;
if(int((vST.t + n) * uStripes) %2 == 0  ){
    float dist = abs(int((vST.t + n) *uStripes) +.5 - (vST.t + n) * uStripes);
   newPosition.xyz +=  (normalize( gl_NormalMatrix * gl_Normal ) * (.5-dist))/ uSkinSmoothness;
}
```