UOW MALAYSIA

PART OF THE UNIVERSITY OF WOLLONGONG AUSTRALIA GLOBAL NETWORK

Bachelor of Game Development (Hons)

# XBGT2094N ASSIGNMENT

## DOCUMENTATION

Semester May **2024**

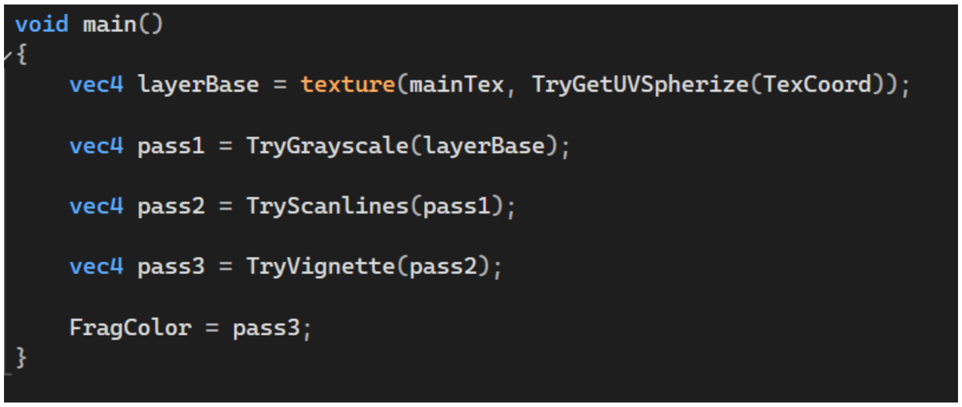
**Joshua Yeoh**
**0135760**

SCHOOL OF COMPUTING & CREATIVE MEDIA

## Themed Post-Processing

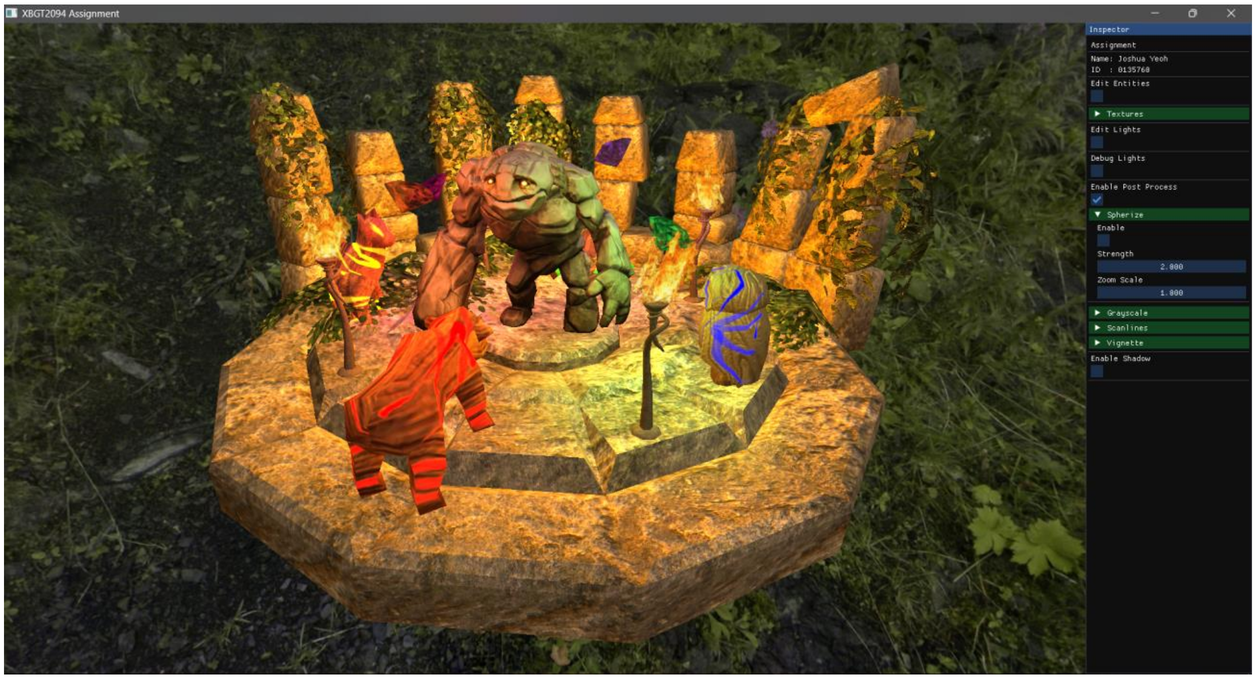
The effects used are Spherize, Zoom, Vignette, Tiling Scanlines, and Greyscale

First, I try to spherize the UV, then grayscale, then add scanlines, then finally add vignette



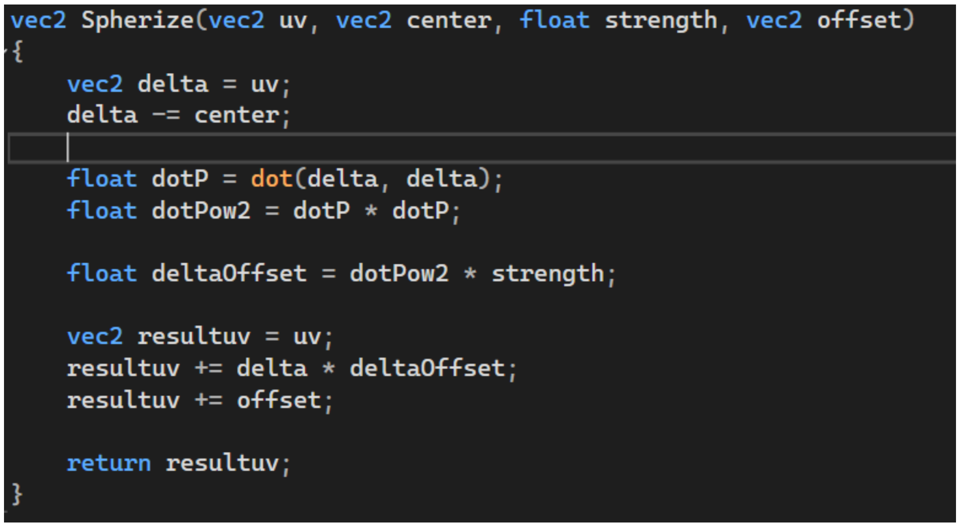
```
void main()
{
    vec4 layerBase = texture(mainTex, TryGetUVSpherize(TexCoord));

    vec4 pass1 = TryGrayscale(layerBase);

    vec4 pass2 = TryScanlines(pass1);

    vec4 pass3 = TryVignette(pass2);

    FragColor = pass3;
}
```

With no post-processing:

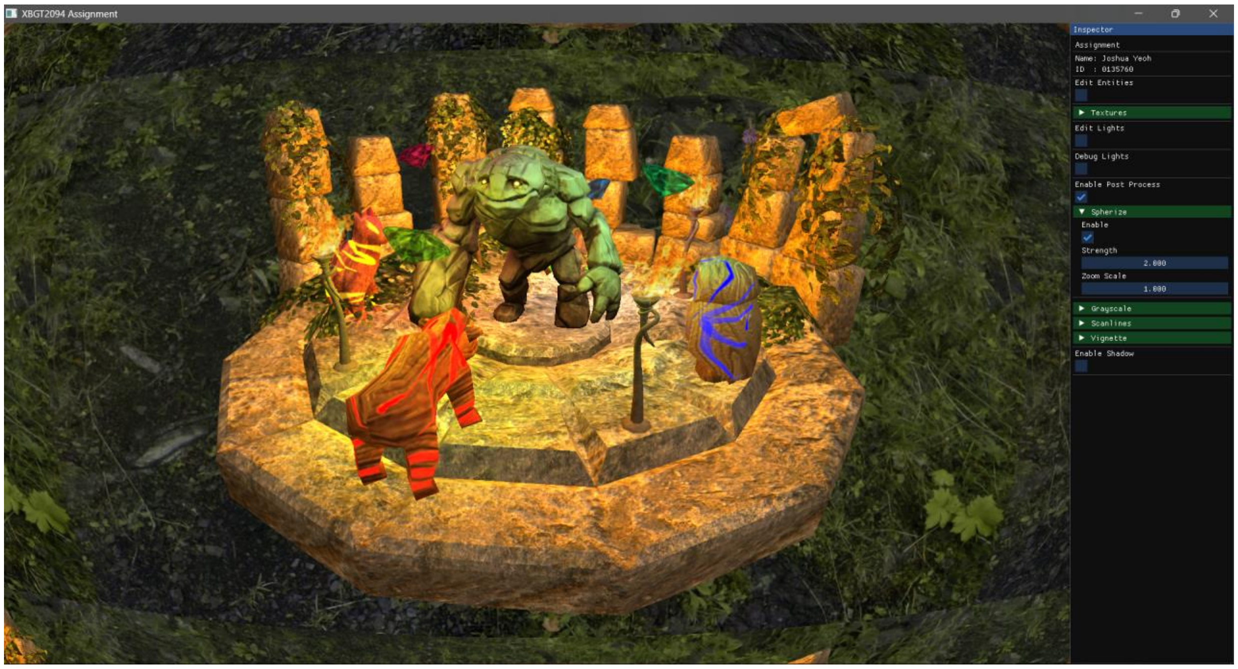


```
Inspector
Assignment
Name: Joshua Yeoh
ID  : 0135760
Edit Entities
☐
▶ Textures
Edit Lights
☐
Debug Lights
☐
Enable Post Process
✓
▼ Spherize
  Enable
  ☐
  Strength
  2.000
  Zoom Scale
  1.000
▶ Grayscale
▶ Scanlines
▶ Vignette
Enable Shadow
☐
```

First is Spherize. I take the texture coordinates and distort them into a spherical shape to get that fisheye effect



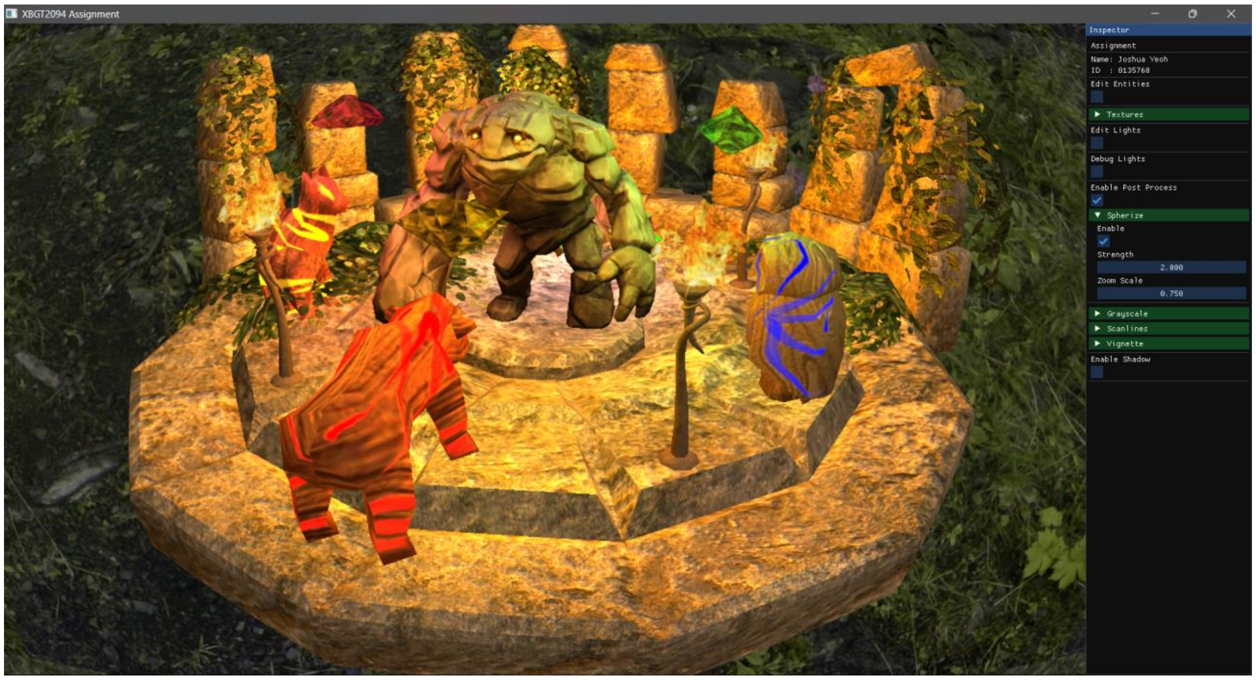
```
vec2 Spherize(vec2 uv, vec2 center, float strength, vec2 offset)
{
    vec2 delta = uv;
    delta -= center;

    float dotP = dot(delta, delta);
    float dotPow2 = dotP * dotP;

    float deltaOffset = dotPow2 * strength;

    vec2 resultuv = uv;
    resultuv += delta * deltaOffset;
    resultuv += offset;

    return resultuv;
}
```



```
Assignment
Name: Joshua Yeoh
ID  : 0135760
Edit Entities ☐
Edit Lights ☐
Debug Lights ☐
Enable Post Process ✓
▼ Spherize
  Enable ✓
  Strength 2.000
  Zoom Scale 1.000
Enable Shadow ☐
```

After I spherize the UV, I zoom in to try and move the weird borders out of the frame a bit

```
vec2 TryGetUVSpherize(vec2 inputUV)
{
    if(!EnablePostProcess) return inputUV;
    if(!EnableSpherize) return inputUV;

    vec2 uvSpherize = Spherize(inputUV, vec2(0.5), SpherizeStrength, vec2(0));

    uvSpherize = Zoom(uvSpherize, vec2(0.5), ZoomScale);

    return fract(uvSpherize);
}
```

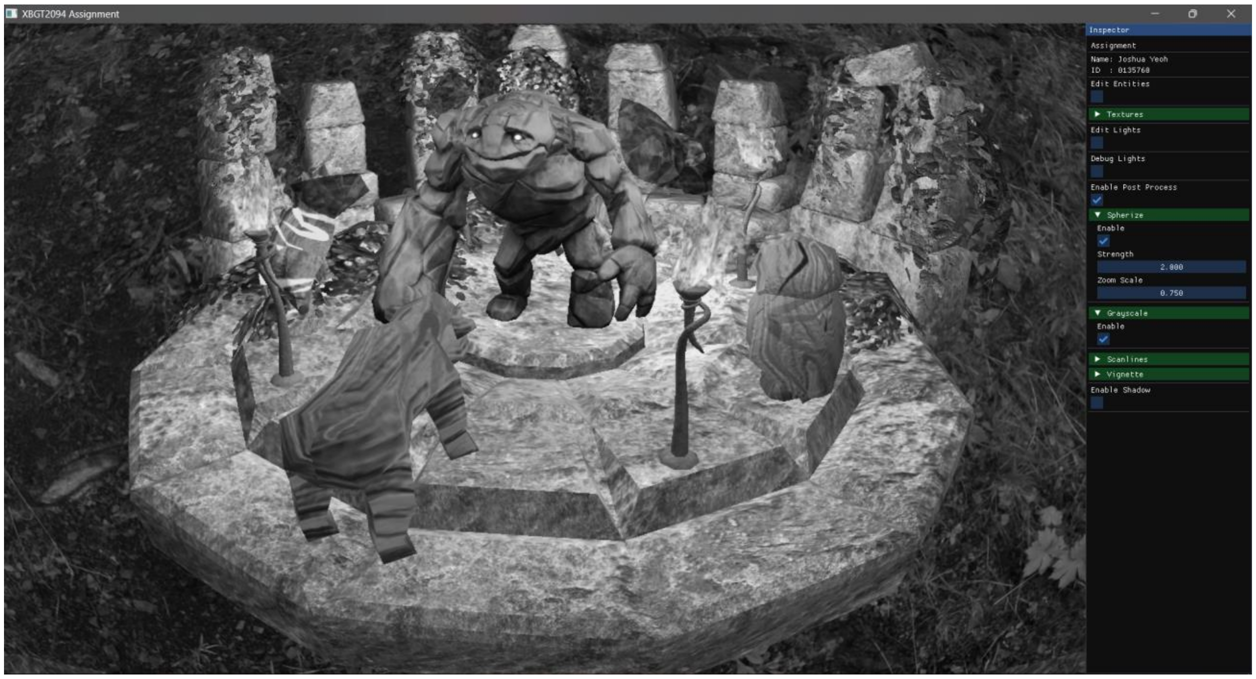


Name: Joshua Yeoh
ID  : 0135760

Edit Entities ☐
▶ Textures
Edit Lights ☐
Debug Lights ☐
Enable Post Process ✓
▼ Spherize
Enable ✓
Strength 2.000
Zoom Scale 0.750
▶ Grayscale
▶ Scanlines
▶ Vignette
Enable Shadow ☐

And then I add grayscale effect

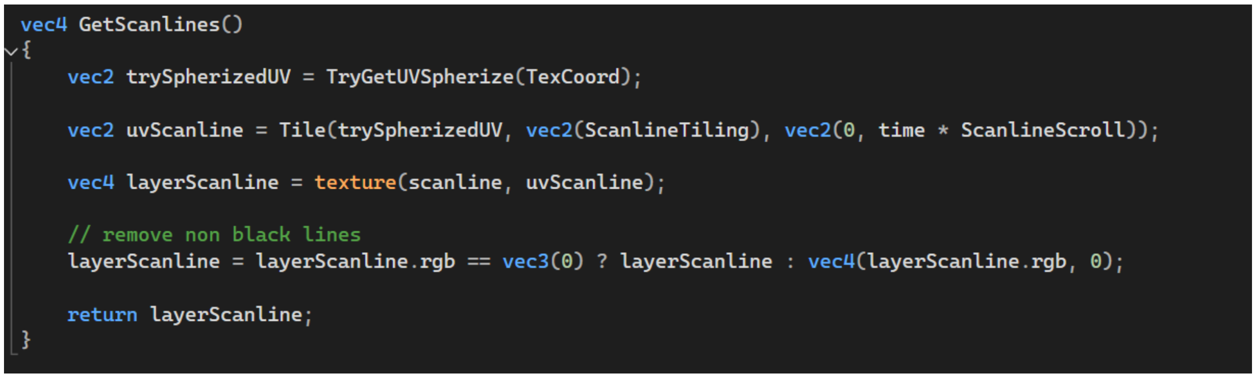
```
vec4 Grayscale(vec4 layer)
{
    float sum = layer.r * 0.299 + layer.g * 0.587 + layer.b * 0.114;

    return vec4(vec3(sum), layer.a);
}
```



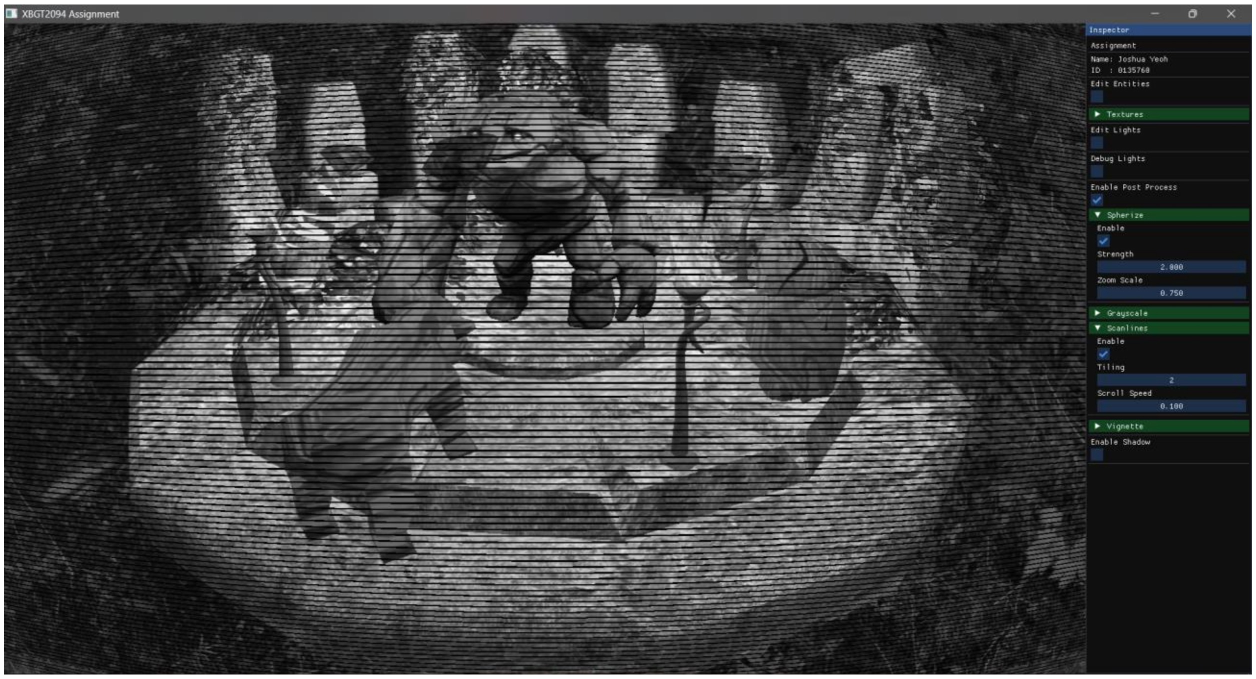
Name: Joshua Yeoh
ID : 0135768

Edit Entities ☐
▶ Textures
Edit Lights ☐
Debug Lights ☐
Enable Post Process ✓
▼ Spherize
Enable ✓
Strength 2.000
Zoom Scale 0.750
▼ Grayscale
Enable ✓
▶ Scanlines
▶ Vignette
Enable Shadow ☐

Then I add a scanline texture overlay. If Spherize effect is also on, then the scanline uv will also distort. To make the scanlines scroll up or down, I used the Tile effect and used the time value to move the y axis.



```
vec4 GetScanlines()
{
    vec2 trySpherizedUV = TryGetUVSpherize(TexCoord);

    vec2 uvScanline = Tile(trySpherizedUV, vec2(ScanlineTiling), vec2(0, time * ScanlineScroll));

    vec4 layerScanline = texture(scanline, uvScanline);

    // remove non black lines
    layerScanline = layerScanline.rgb == vec3(0) ? layerScanline : vec4(layerScanline.rgb, 0);

    return layerScanline;
}
```

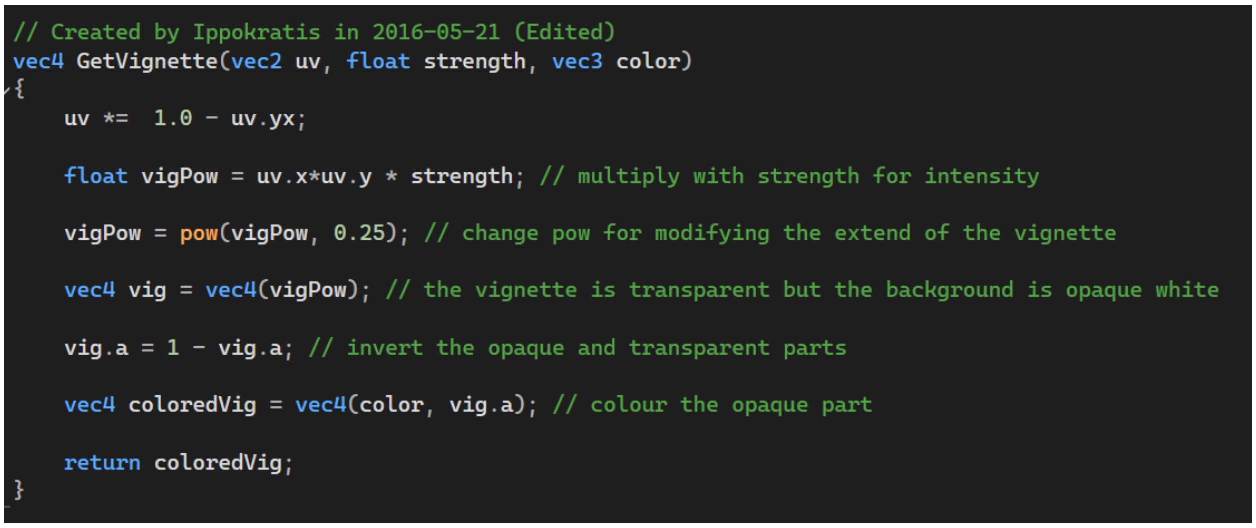


```
Assignment
Name: Joshua Yeoh
ID  : 0135760
Edit Entities
☐
▶ Textures
Edit Lights
☐
Debug Lights
☐
Enable Post Process
✓
▼ Spherize
  Enable
  ✓
  Strength
  2.000
  Zoom Scale
  0.750
▶ Grayscale
▼ Scanlines
  Enable
  ✓
  Tiling
  2
  Scroll Speed
  0.100
▶ Vignette
Enable Shadow
☐
```

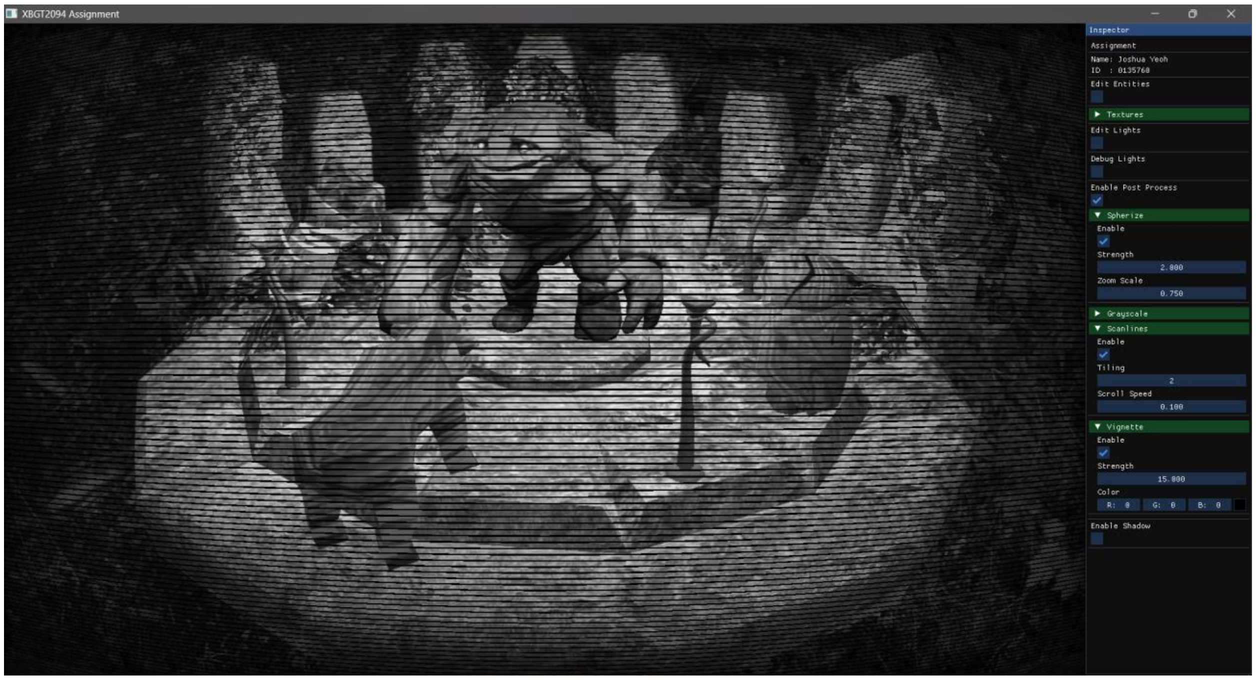
Lastly, I add the vignette effect for darker edges.

The main vignette code was created by Ippokratis in 2016-05-21 from shadertoy.com.

I just added the ability to change the vignette colour.



```
// Created by Ippokratis in 2016-05-21 (Edited)
vec4 GetVignette(vec2 uv, float strength, vec3 color)
{
    uv *=  1.0 - uv.yx;

    float vigPow = uv.x*uv.y * strength; // multiply with strength for intensity

    vigPow = pow(vigPow, 0.25); // change pow for modifying the extend of the vignette

    vec4 vig = vec4(vigPow); // the vignette is transparent but the background is opaque white

    vig.a = 1 - vig.a; // invert the opaque and transparent parts

    vec4 coloredVig = vec4(color, vig.a); // colour the opaque part

    return coloredVig;
}
```



```
Assignment
Name: Joshua Yeoh
ID  : 0135768
Edit Entities  ☐
▶ Textures
Edit Lights  ☐
Debug Lights  ☐
Enable Post Process  ✓
▼ Spherize
  Enable  ✓
  Strength  2.000
  Zoom Scale  0.750
▶ Grayscale
▼ Scanlines
  Enable  ✓
  Tiling  2
  Scroll Speed  0.100
▼ Vignette
  Enable  ✓
  Strength  15.000
  Color  R: 0  G: 0  B: 0
Enable Shadow  ☐
```

The effects can be individually toggled and tweaked in the inspector.

## Credits

Simple vignette effect by Ippokratis

https://www.shadertoy.com/view/lsKSWR

Forest skyboxes by Emil Persson

https://opengameart.org/content/forest-skyboxes

3TD Fantasy Ruins Pack by Ron Kapaun

https://opengameart.org/content/3td-fantasy-ruins-pack

Animal Figurines by Clint Bellanger

https://opengameart.org/content/animal-figurines

Gems by Clint Bellanger

https://opengameart.org/content/gems

fantasy torch by rubberduck

https://opengameart.org/content/fantasy-torch-pack-1

Flame Particle by Keith333

https://opengameart.org/content/flame-particle-set-4-in-total

Normal Map Online by Christian Petry and Am Gänsweiher

https://cpetry.github.io/NormalMap-Online/

Learn OpenGL and Shadow Maps (Directional Lights) Tutorial by Victor Gordan

https://learnopengl.com/Advanced-Lighting/Shadows/Shadow-Mapping

https://youtu.be/9g-4aJhCnyY?si=dAAKu7lS9MZcFn7L

(When I was attempting renderer feature)