HW5-GPU Programming for video games
(Post processing pipeline)

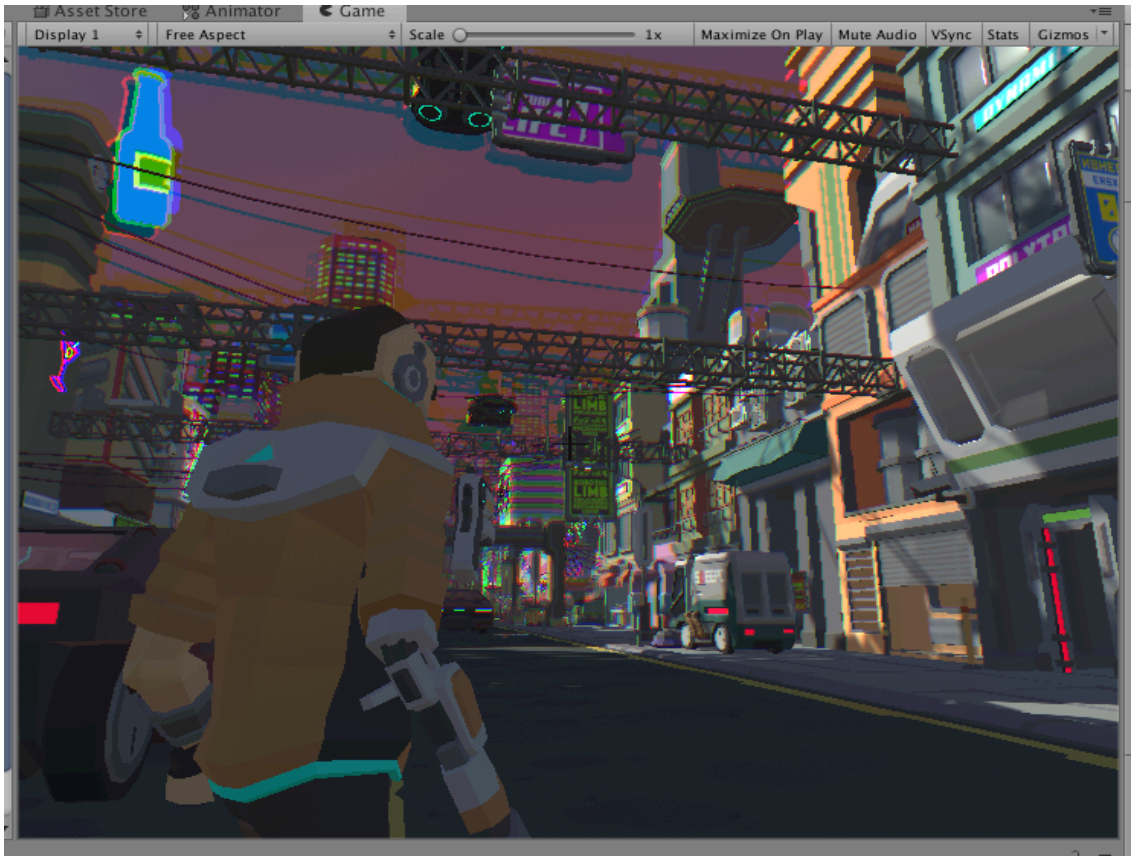
Ankit Arora-anki@gatech.edu

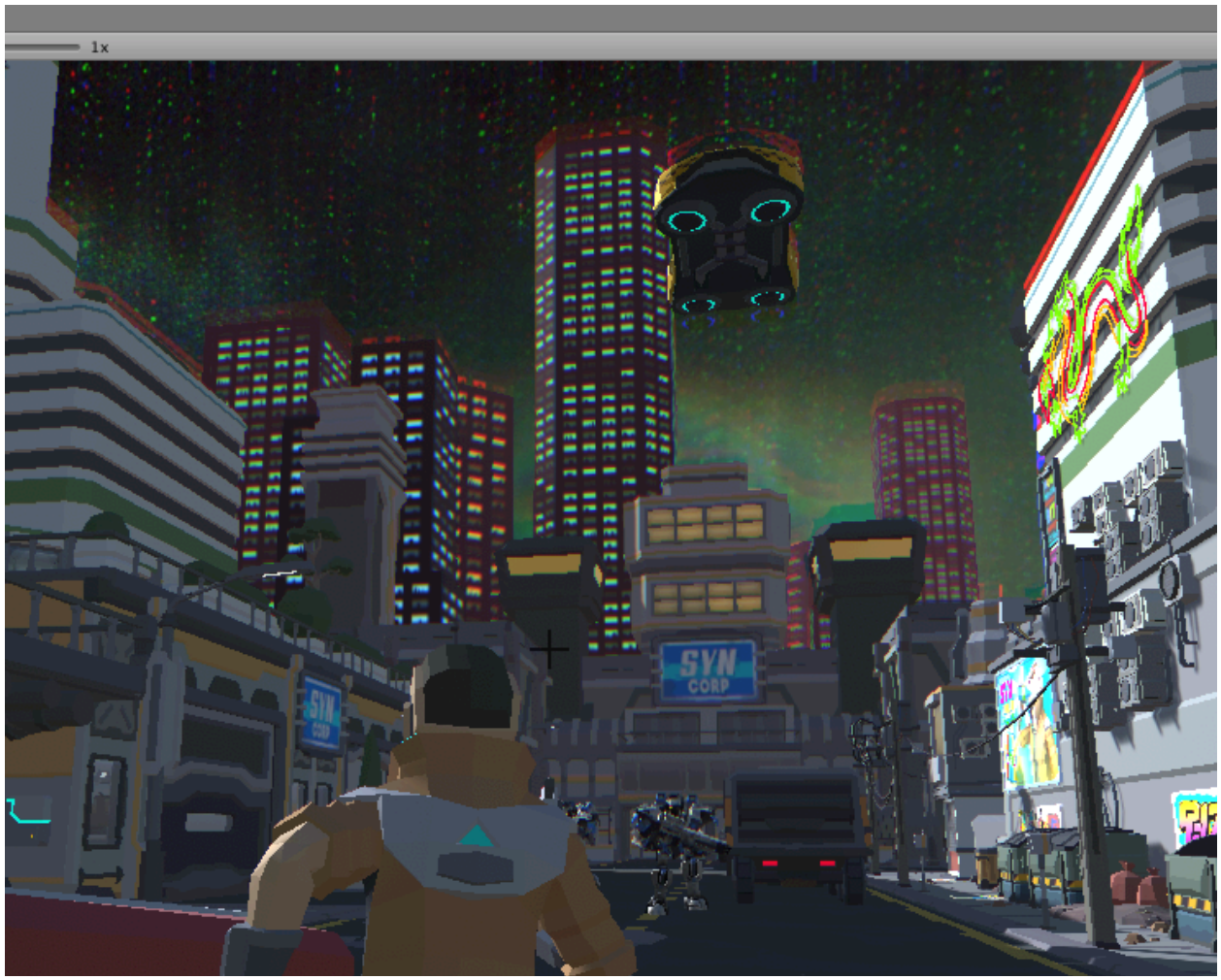
For this assignment, I created RGB miss-representation effect, which changes with depth. Normal is used to find direction of color shift, while depth information is used to for proportional color shift based on the field of depth.

1) First trial – GPU19ColorMissRepresentation (post processing effect)

```
float4 Frag(VaryingsDefault i) : SV_Target
{

    float shh = _Shift*(cos(_Speed * _Time.y) - 1);
    float4 original = SAMPLE_TEXTURE2D(_MainTex, sampler_MainTex, i.texcoord);
    float4 origina;
    float4 dn_enc = SAMPLE_TEXTURE2D(_CameraDepthNormalsTexture, sampler_CameraDepthNormalsTexture, i.texcoord);
    float depth = dot(float2(1.0, 1/255.0),dn_enc.zw)+_depth;

    float3 n = DecodeViewNormalStereo(dn_enc);
    float3 display_n = 0.5 * (1 + n);
    origina.r = SAMPLE_TEXTURE2D(_MainTex, sampler_MainTex, i.texcoord+depth*float2(n.x*shh / _ScreenParams.x,n.z*shh / _ScreenParams.y)).r;
    origina.b = SAMPLE_TEXTURE2D(_MainTex, sampler_MainTex, i.texcoord-depth*float2(n.x*shh / _ScreenParams.x,n.z*shh / _ScreenParams.y)).b;
    origina.g = SAMPLE_TEXTURE2D(_MainTex, sampler_MainTex, i.texcoord).g;//+depth*float2(n.x*-shh / _ScreenParams.y,n.z*shh / _ScreenParams.x)).b;
    return float4(origina.rgb,1);
    //return(float4(lerp(origina.rgb,original.rgb,0.5*(cos(_Speed * _Time.y) - 1)),1));
    //return(float4(lerp(original.rgb,depth.xxx,0.5*(cos(_Speed * _Time.y) - 1)),1));

}
```

Effect give different color miss representation per depth shown below





Gives colorful stars

2) Second trial - GPU19ColorMissRepresentationBlend (Post processing effect). Tried to blend color per color channel asynchronously. Gives much better results, similar to ‘spider-verse’ effect, scene looks more comical. Also acts as a motion blurr.

```
float4 Frag(VaryingsDefault i) : SV_Target
{

    float shh = _Shift;//*(cos(_Speed * _Time.y) + 1);
    float4 original = SAMPLE_TEXTURE2D(_MainTex, sampler_MainTex, i.texcoord);
    float4 origina;
    float4 dn_enc = SAMPLE_TEXTURE2D(_CameraDepthNormalsTexture, sampler_CameraDepthNormalsTexture, i.texcoord);
    float depth = dot(float2(1.0, 1/255.0),dn_enc.zw)+1;//*10*(cos(_Speed * _Time.y) + 1);

    float3 n = DecodeViewNormalStereo(dn_enc);
    float3 display_n = 0.5 * (1 + n);

    origina.r = SAMPLE_TEXTURE2D(_MainTex, sampler_MainTex, i.texcoord+depth*float2(n.x*shh / _ScreenParams.x,n.z*shh / _ScreenParams.y)).r;
    origina.b = SAMPLE_TEXTURE2D(_MainTex, sampler_MainTex, i.texcoord-depth*float2(n.x*shh / _ScreenParams.x,n.z*shh / _ScreenParams.y)).b;
    origina.g = SAMPLE_TEXTURE2D(_MainTex, sampler_MainTex, i.texcoord+depth*float2(-n.x*shh / _ScreenParams.x,n.z*shh / _ScreenParams.y)).g;

    origina.r = lerp(original.r,origina.r,0.3*depth*(cos(_Speed * (_Time.y+2)) + 1));
    origina.g = lerp(original.g,origina.g,0.3*depth*(cos(_Speed * (_Time.y+4)) - 1));
    origina.b = lerp(original.b,origina.b,0.3*depth*(cos(_Speed * (_Time.y+6)) + 1));



    return float4(origina.rgb,1);

}
```

Results


SYN
CORP
PINKKITTY


Display 1
Free Aspect
Scale
1x
Maximize On Play
Mute Audio
VSync
Stats
Gizmos
ROBOTIC
LIMB