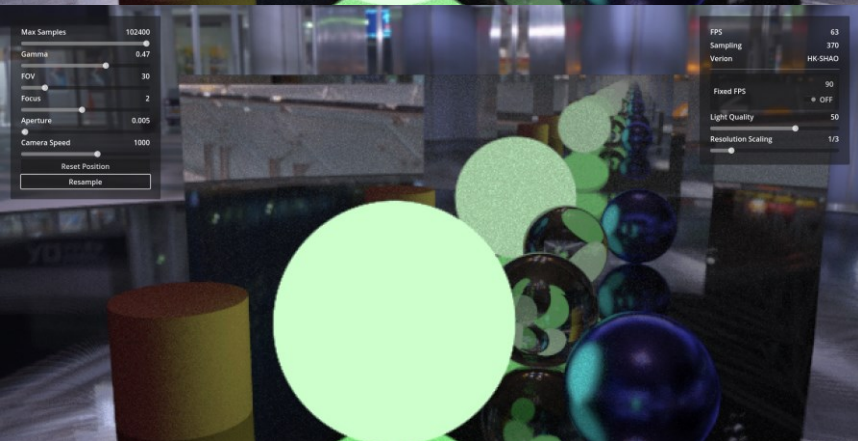
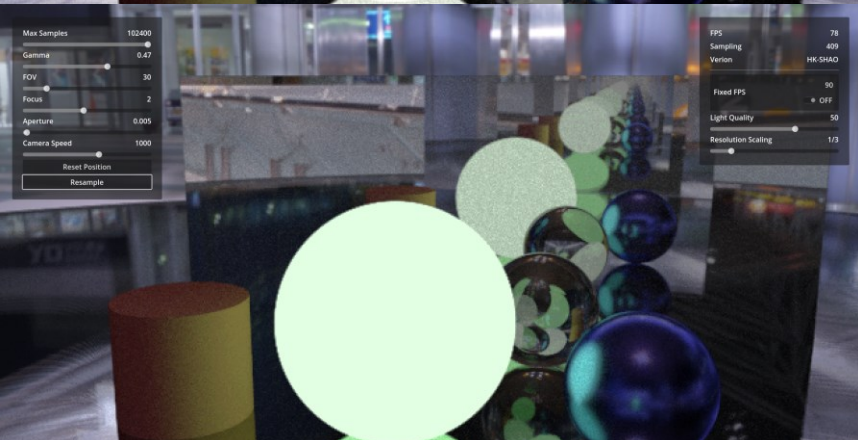


```
vec3 ACES Tone Mapping(vec3 color, float adapted_lum){
    const float a = 2.51f;
    const float b = 0.03f;
    const float c = 2.43f;
    const float d = 0.59f;
    const float e = 0.14f;
    color *= adapted_lum;
    return (color * (a * color + b)) / (color * (c * color + d) + e);
}
```

```
vec3 HDR(vec3 color) {
    return vec3(1) - exp(-color);
}
```

```
vec3 tone_mapping(vec3 color) {
    const float a = 0.15;
    const float b = 0.50;
    const float c = 0.10;
    const float d = 0.20;
    const float e = 0.02;
    const float f = 0.30;
    const float w = 11.2;
    const float exposure = 3.0;
    color *= exposure;
    color = ((color * (a * color + c * b) + d * e) / (color * (a * color + b) + d * f)) - e / f;
    float white = ((w * (a * w + c * b) + d * e) / (w * (a * w + b) + d * f)) - e / f;
    color /= white;
    return color;
}
```

None Tone Mapping



```
vec3 RRTAndODTFit(vec3 v)
{
    vec3 a = v * (v + 0.0245786) - 0.000090537;
    vec3 b = v * (0.983729 * v + 0.4329510) + 0.238081;
    return a / b;
}

vec3 ACESFitted(vec3 color)
{
    color *= 5.0;
    color = (transpose(ACESInputMat) * color);
    // Apply RRT and ODT
    color = RRTAndODTFit(color);
    color = (transpose(ACESOutputMat) * color);
    return(color);
}
```

```
// https://github.com/TheRealMJP/BakingLab/blob/master/BakingLab/ACES.hlsl
const mat3 ACESInputMat = mat3(
    vec3(0.59719, 0.35458, 0.04823),
    vec3(0.07600, 0.90834, 0.01566),
    vec3(0.02840, 0.13383, 0.83777)
);
```

```
const mat3 ACESOutputMat = mat3(
    vec3( 1.60475, -0.53108, -0.07367),
    vec3(-0.10208,  1.10813, -0.00605),
    vec3(-0.00327, -0.07276,  1.07602)
);
```

```
vec3 ACES_tone_mapping(vec3 color) {
    const float a = 0.15;
    const float b = 0.50;
    const float c = 0.10;
    const float d = 0.20;
    const float e = 0.02;
    const float f = 0.30;
    const float w = 11.2;
    const float exposure = 5.0;
    color *= exposure;
    color = ((color * (a * color + c * b) + d * e) / (color * (a * color + b) + d * f)) - e / f;
    float white = ((w * (a * w + c * b) + d * e) / (w * (a * w + b) + d * f)) - e / f;
    color /= white;
    return color;
}
```

```
vec3 HDR(vec3 color) {
    return vec3(1) - exp(-color);
}
```