

# 3. Вершинный шейдер

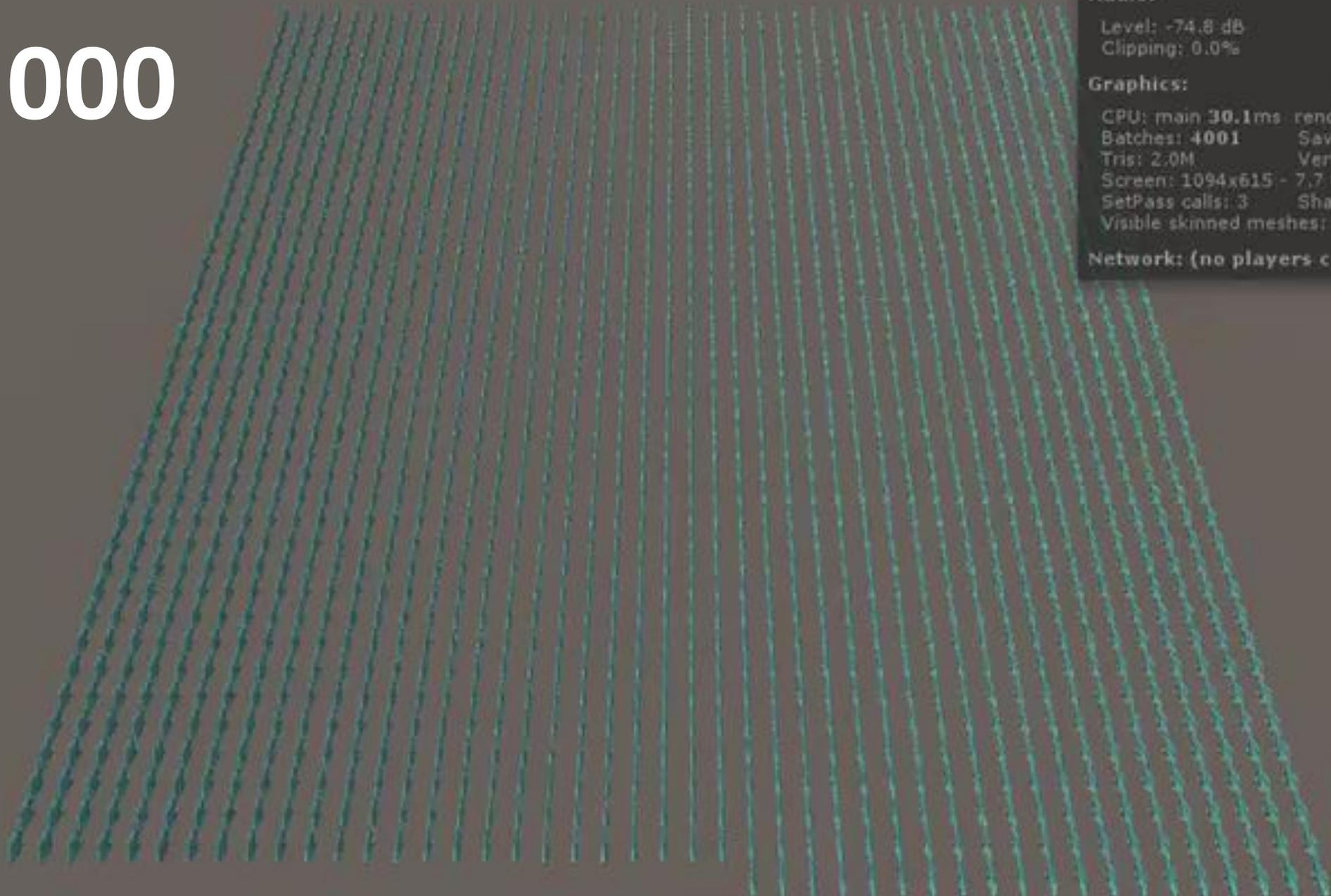
Графика в современных играх





- ▼ cruscarp
  - cruscarp
    - ▼ cruscarp\_skel
      - ▼ Bone
        - ▼ Bone.001
          - ▼ Bone.003
            - Bone.003\_end
        - ▼ Bone.002\_L.001
          - Bone.002\_L.001\_end
        - ▼ Bone.002\_R.001
          - Bone.002\_R.001\_end
        - ▼ Bone\_L.001
          - Bone\_L.001\_end
        - ▼ Bone\_R.001
          - Bone\_R.001\_end
      - ▼ Bone.004
        - ▼ Bone.004\_L.001
          - Bone.004\_L.001\_end
        - ▼ Bone.004\_R.001
          - Bone.004\_R.001\_end
      - ▼ Bone.005
        - ▼ Bone.006
          - Bone.006\_end
      - ▼ bubble
        - bubble\_end
      - ▼ Bone.008
        - Bone.008\_end
      - ▼ Bone.010
        - Bone.010\_end
      - ▼ Bone.012
        - Bone.012\_end

# 2 000



## Statistics

### Audio:

Level: -74.8 dB  
Clipping: 0.0%

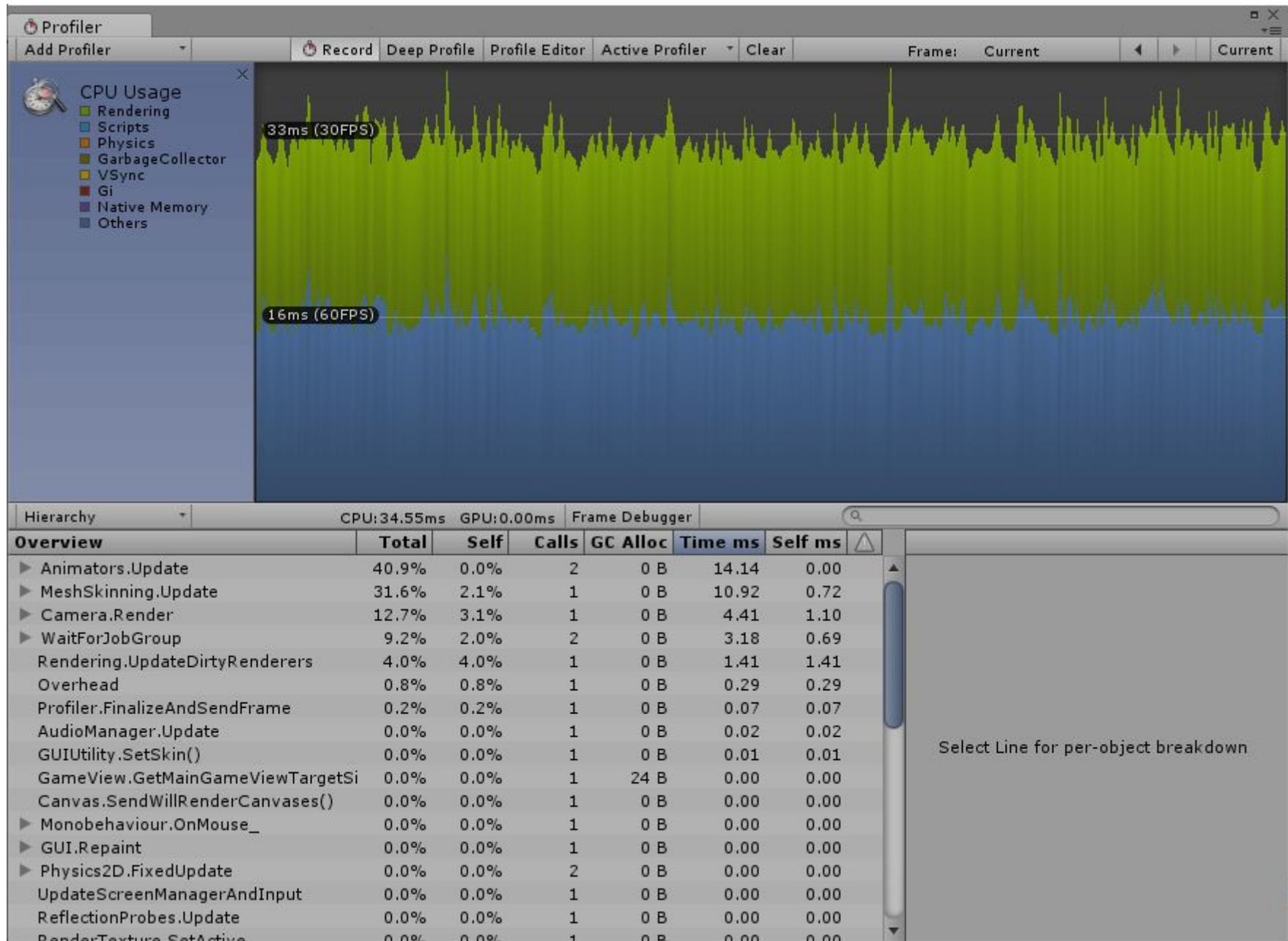
DSP load: 0.2%  
Stream load: 0.0%

### Graphics:

33.2 FPS (30.1ms)

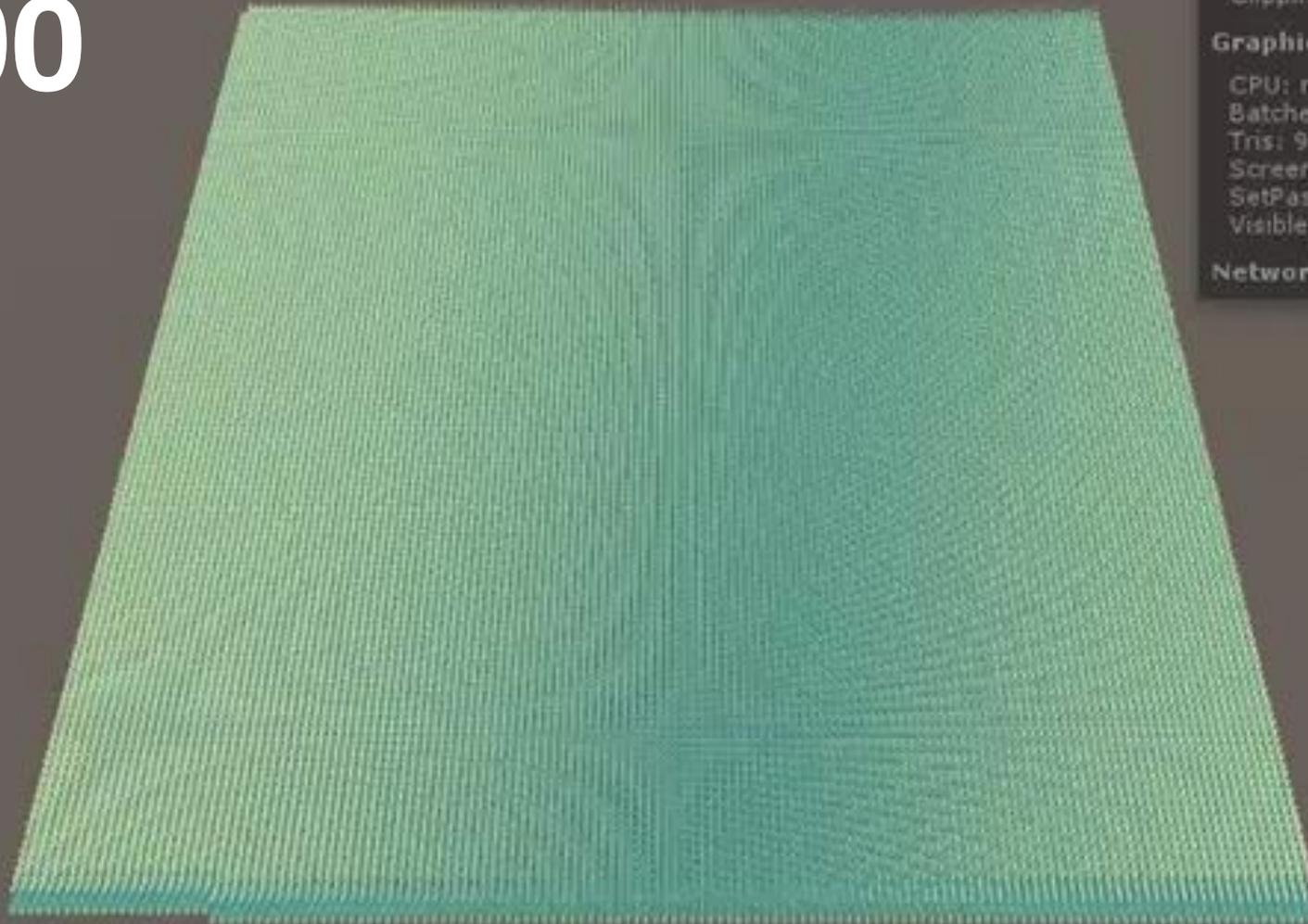
CPU: main 30.1ms render thread 8.3ms  
Batches: 4001 Saved by batching: 0  
Tris: 2.0M Verts: 1.6M  
Screen: 1094x615 - 7.7 MB  
SetPass calls: 3 Shadow casters: 0  
Visible skinned meshes: 2000 Animations: 0

Network: (no players connected)





20 000



Statistics

**Audio:**

Level: -74.8 dB  
Clipping: 0.0%

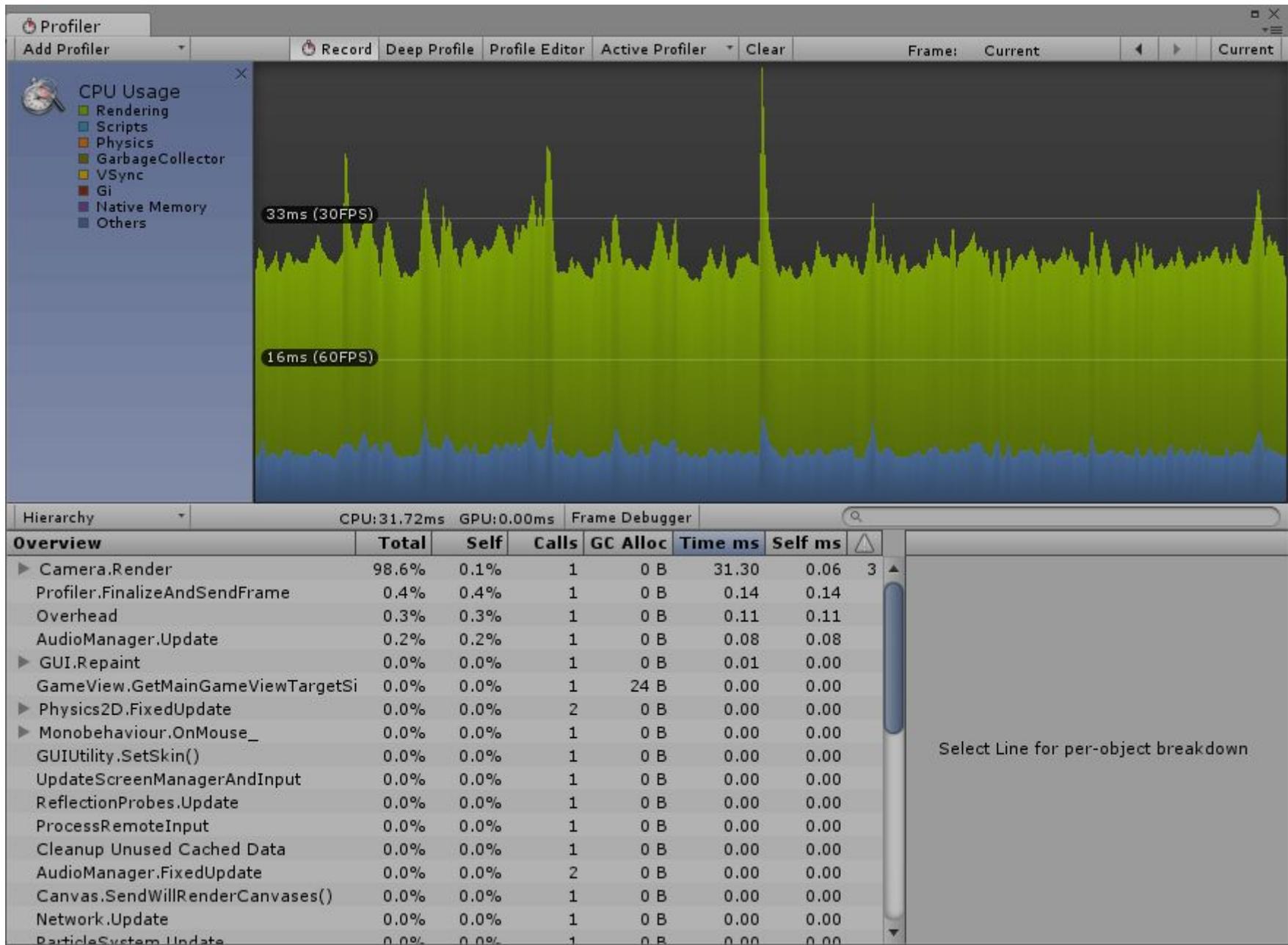
DSP load: 0.2%  
Stream load: 0.0%

**Graphics:**

38.5 FPS (26.0ms)

CPU: main 26.0ms render thread 9.3ms  
Batches: 20001 Saved by batching: 0  
Tris: 9.8M Verts: 8.2M  
Screen: 1094x615 - 7.7 MB  
SetPass calls: 2 Shadow casters: 0  
Visible skinned meshes: 0 Animations: 0

**Network: (no players connected)**



```
1 Shader "Test/FishAnimationShader" {
2
3     Properties {
4
5         _MainTex ("Texture", 2D) = "white" {}
6         _Length ("Length", float) = 1.0
7         _PivotY ("PivotY", float) = 1.0
8         _Offset ("Offset", Vector) = (0.0, 0.0, 0.0)
9         _SinOffset ("SinOffset", Vector) = (0.0, 0.0, 0.0)
10    }
11
12    SubShader {
13
14        Pass { ... }
15
16        ...
17
18        ...
19
20        ...
21
22        ...
23
24        ...
25
26        ...
27
28        ...
29
30        ...
31
32        ...
33
34        ...
35
36        ...
37
38        ...
39
40        ...
41
42        ...
43
44        ...
45
46        ...
47
48        ...
49
50        ...
51
52        ...
53
54        ...
55
56        ...
57
58        ...
59
60        ...
61
62        ...
63
64        ...
65
66        ...
67
68        ...
69
70    }
71
72 }
```

Inspector Lighting

FishAnimationMaterial

Shader Test/FishAnimationShader

Texture

Tiling X 1 Y 1

Offset X 0 Y 0



Length 0.3

PivotY 0.1

Offset X 0.01 Y 0 Z 0 W 1

SinOffset X 0.05 Y 5 Z 0 W 5

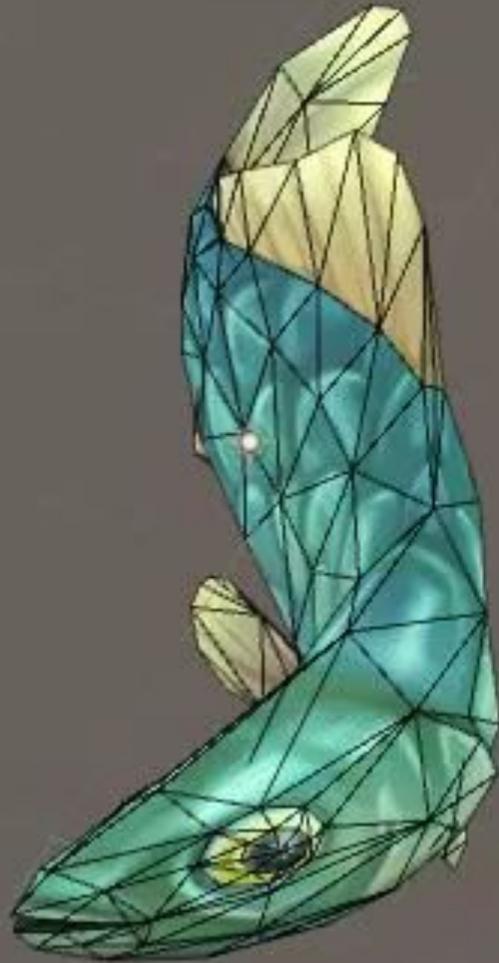
Render Queue From Shader 2000

```
47 vertOutput vert(appdata v)
48 {
49     vertOutput o;
50
51     fixed4 offset = _Offset * cos(_Time.y * _Offset.w);
52     offset.x += _SinOffset.x * sin(_Time.y * _SinOffset.w + (v.vertex.y + _PivotY) * _SinOffset.y);
53     offset *= (v.vertex.y + _PivotY) / _Length;
54
55     v.vertex += offset;
56
57
58     o.vertex = UnityObjectToClipPos(v.vertex);
59     o.uv = TRANSFORM_TEX(v.uv, _MainTex);
60     return o;
61 }
```



51

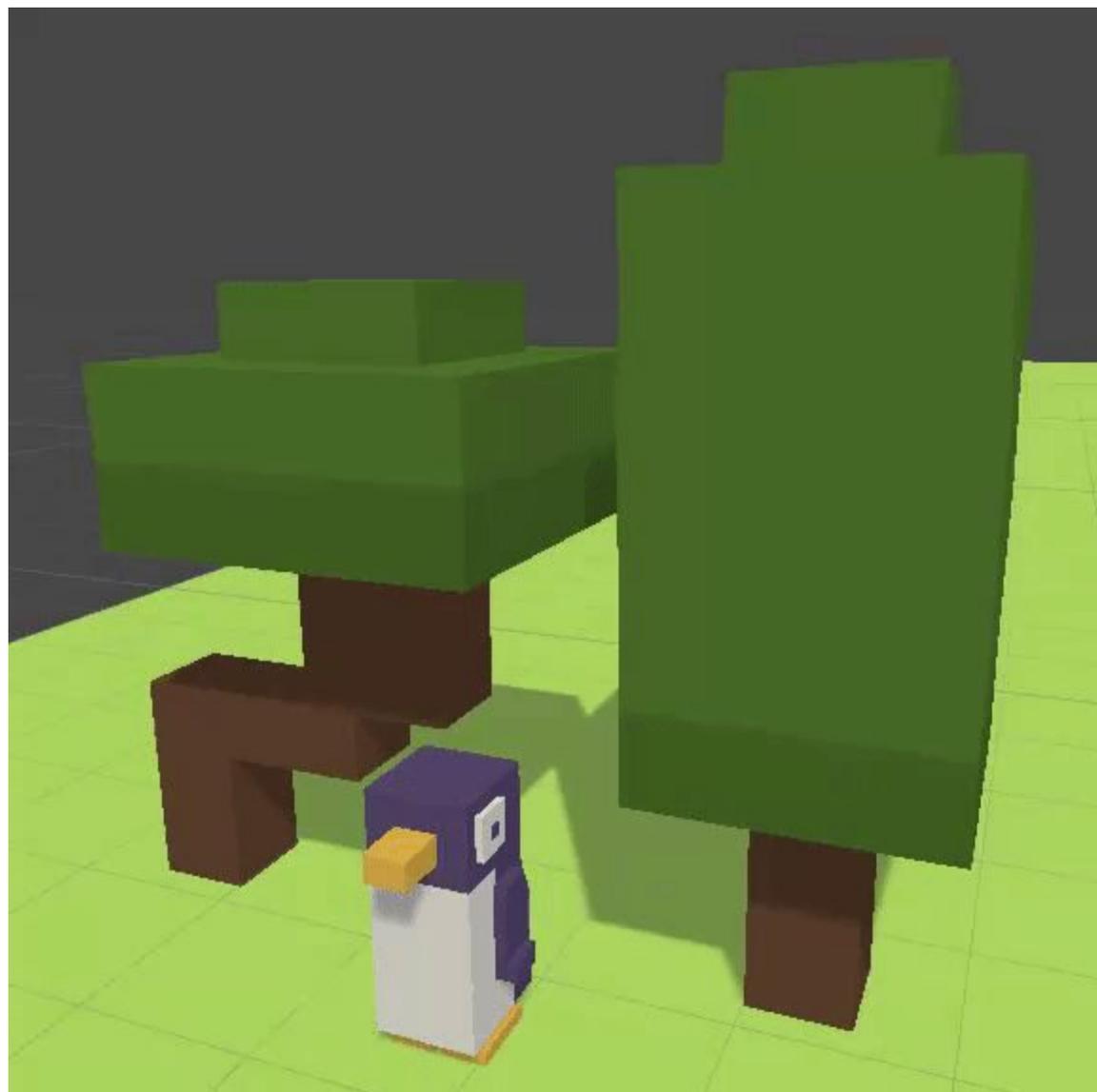
```
fixed4 offset = _Offset * cos(_Time.y * _Offset.w);
```





53

```
offset *= (v.vertex.y + _PivotY) / _Length;
```



```
1 Shader "Studdy Road/Bending Tree" {
2
3     Properties {
4         _MainTex("Texture", 2D) = "white" {}
5         _BendAmplitude("Bend amplitude", Vector) = (0.0, 0.0, 0.0, 0.0)
6         _BendSpeed("Bend speed", float) = 1.0
7     }
8
9     SubShader { ...
36     }
37
38     Fallback "Diffuse"
39 }
```

Inspector Lighting

BendingTree

Shader Studdy Road/Bending Tree

Texture

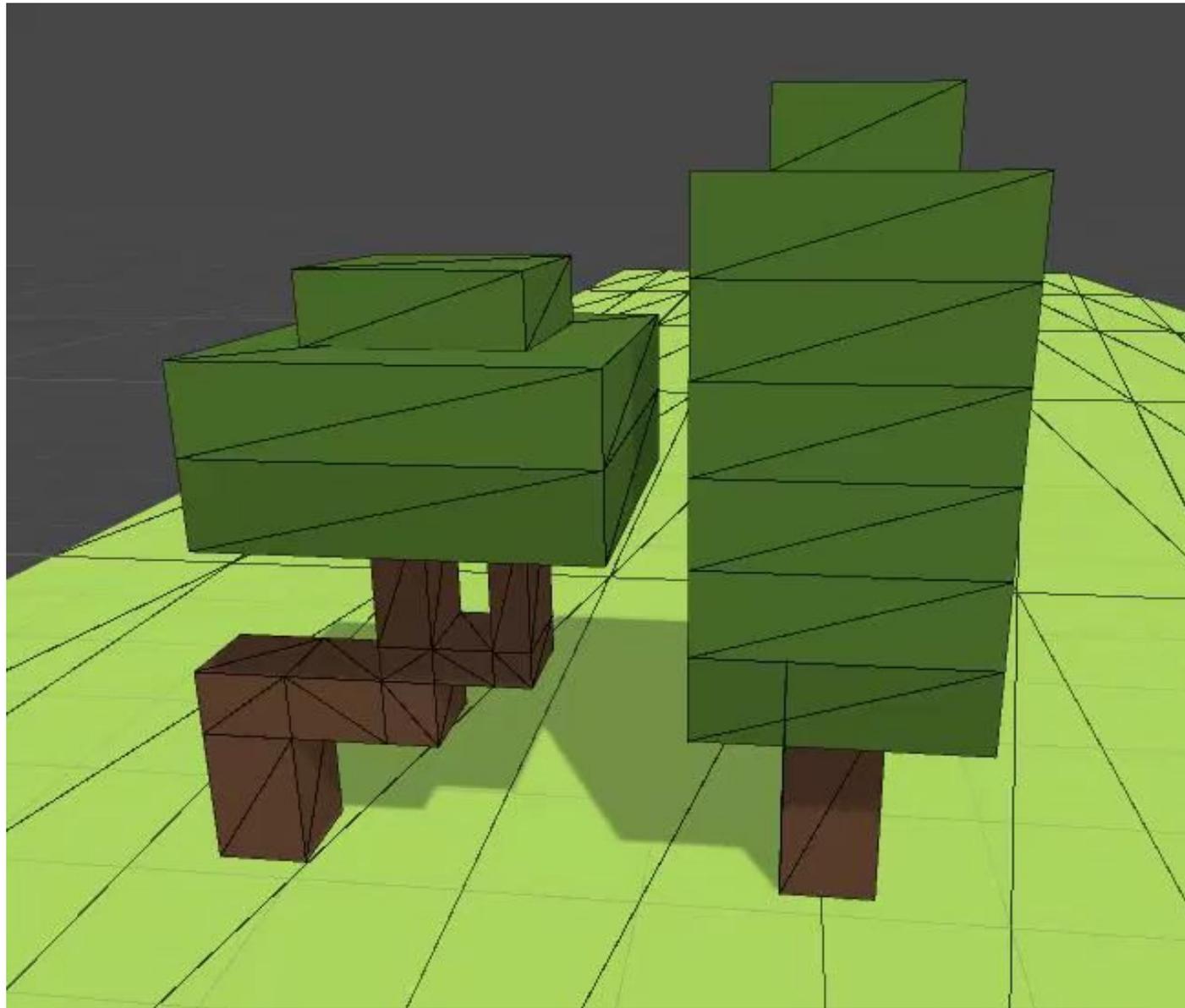
Tiling X 1 Y 1  
Offset X 0 Y 0



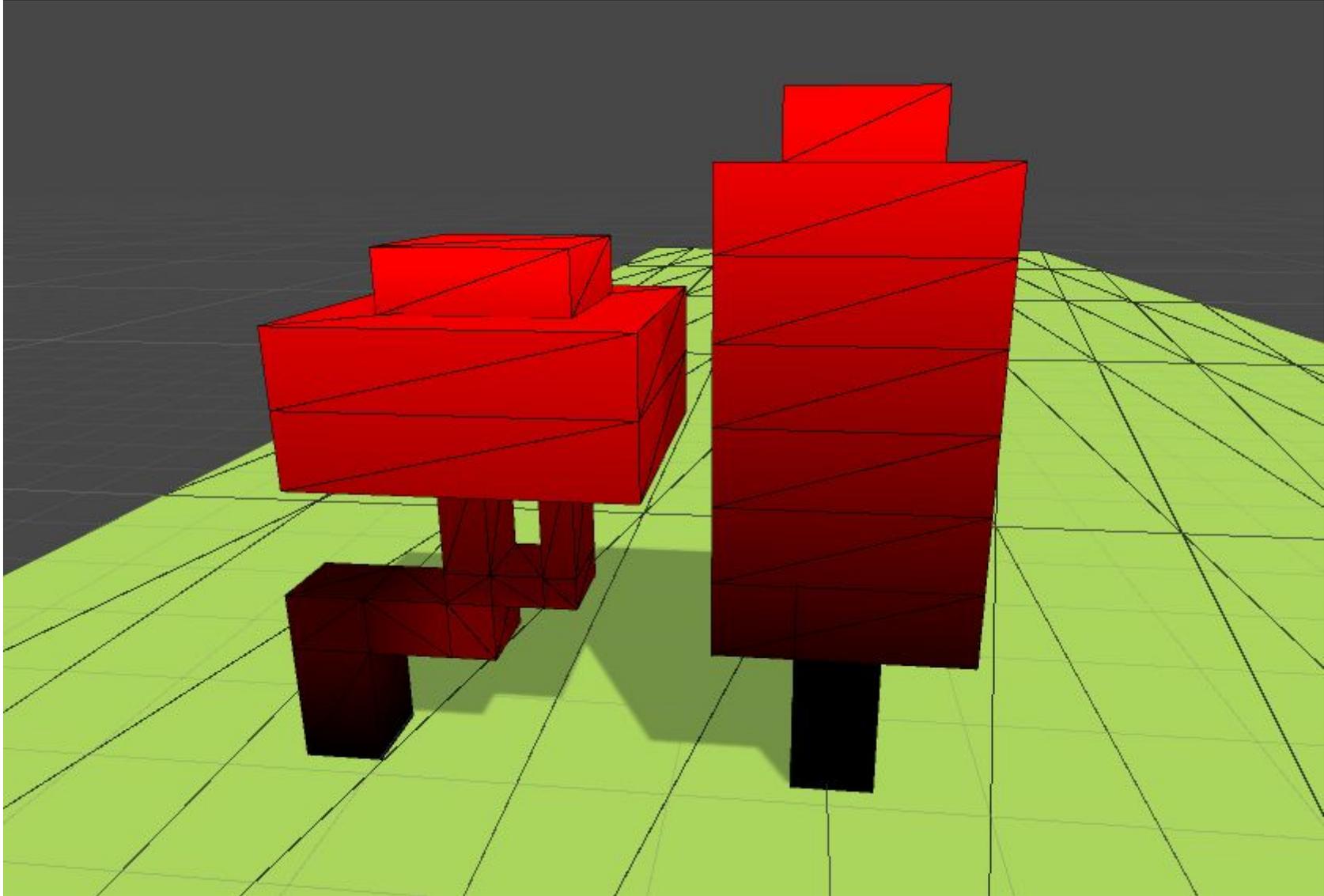
Bend amplitude X 0.05 Y 0.01 Z 0.02 W 0

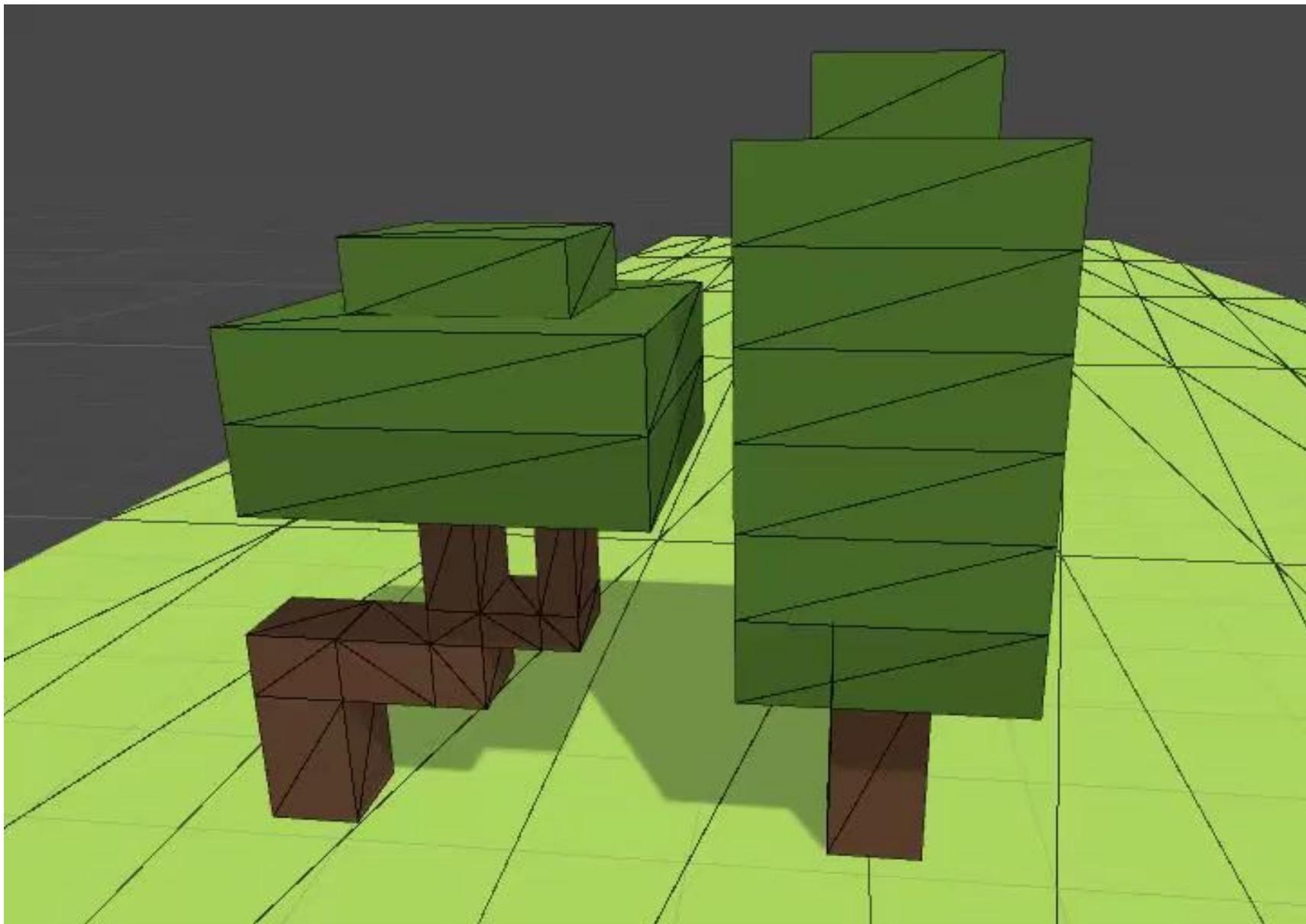
Bend speed 5

Render Queue From Shader 2000

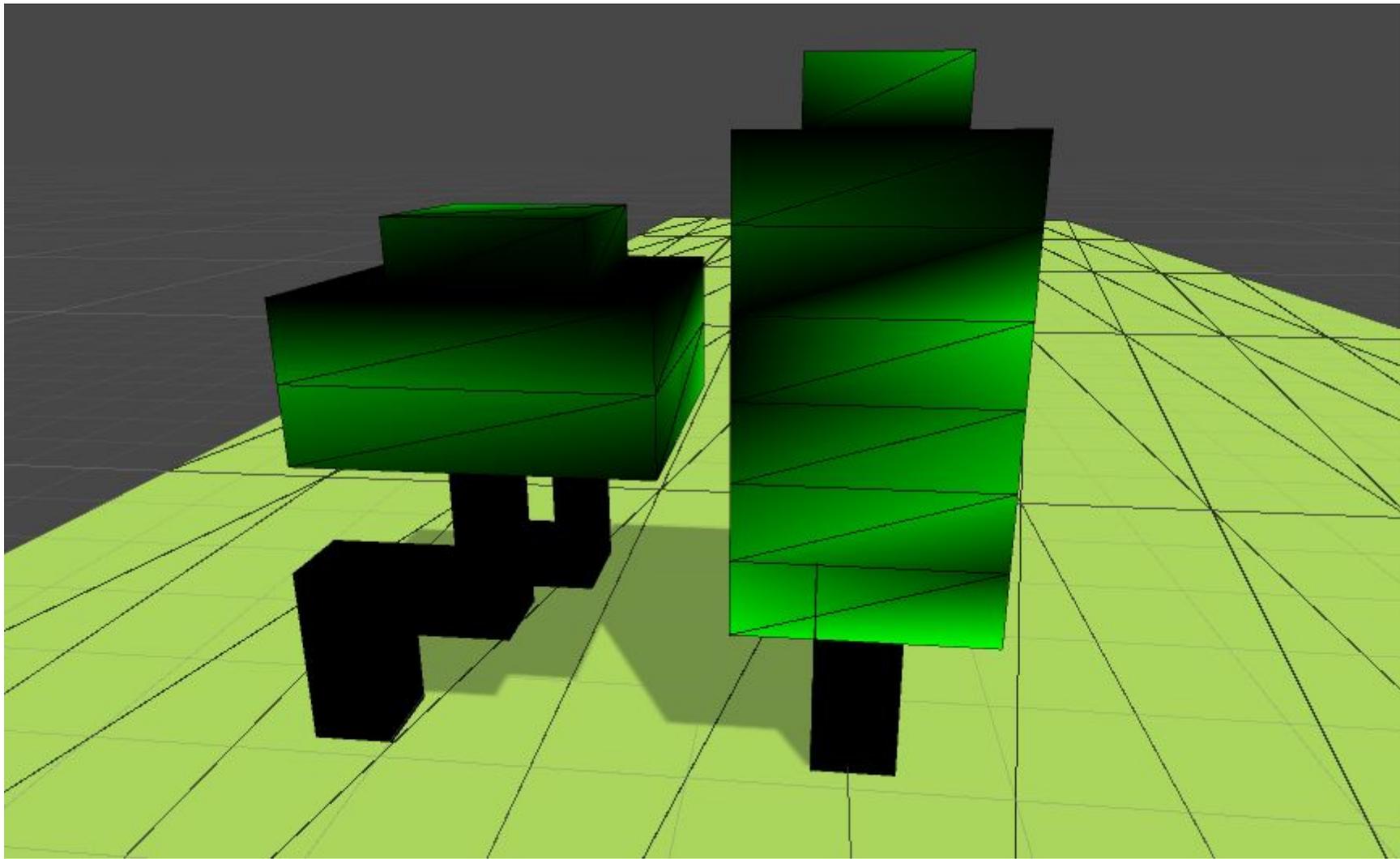


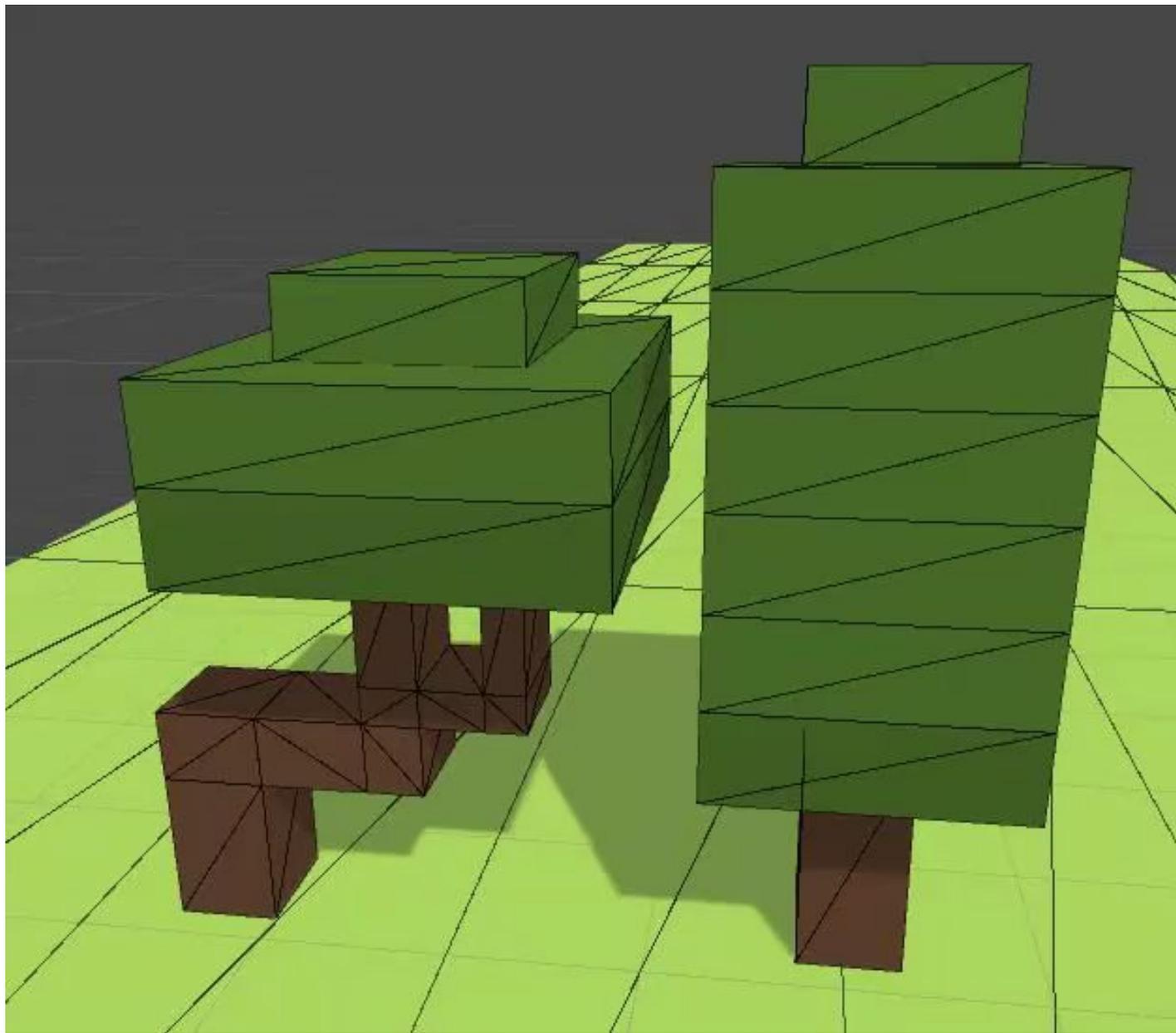
```
v.vertex += _BendAmplitude * sin(_Time.y * _BendSpeed);
```





```
v.vertex += _BendAmplitude * v.color.r * sin(_Time.y * _BendSpeed);
```





```
v.vertex += _BendAmplitude * v.color.r * sin(_Time.y * _BendSpeed + v.color.g);
```

```
CGPROGRAM
```

```
#pragma surface surf Lambert vertex:vert addshadow
```

```
struct Input {
```

```
    float2 uv_MainTex;
```

```
};
```

```
sampler2D _MainTex;
```

```
float4 _BendAmplitude;
```

```
float _BendSpeed;
```

```
void vert (inout appdata_full v, out Input o) {
```

```
    UNITY_INITIALIZE_OUTPUT(Input, o);
```

```
    v.vertex += _BendAmplitude * v.color.r * sin(_Time.y * _BendSpeed + v.color.g);
```

```
}
```

```
void surf (Input IN, inout SurfaceOutput o) {
```

```
    o.Albedo = tex2D(_MainTex, IN.uv_MainTex).rgb;
```

```
}
```

```
ENDCG
```