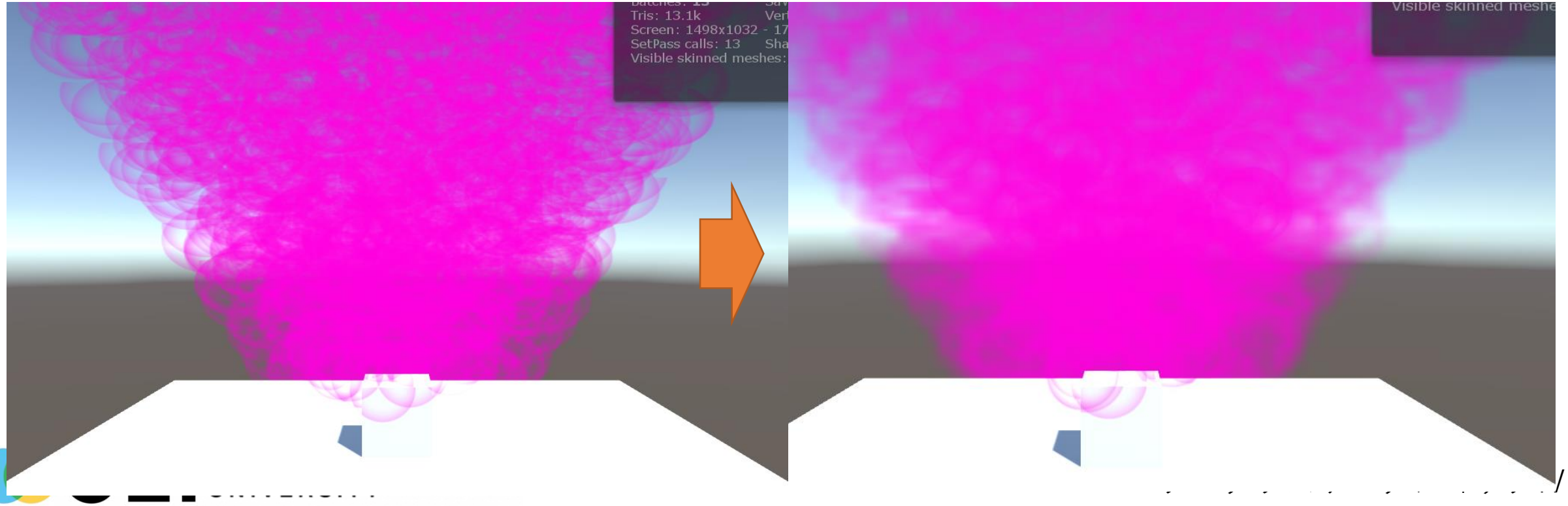
プログラム
ワークショップIV

(7) 縮小バッファ

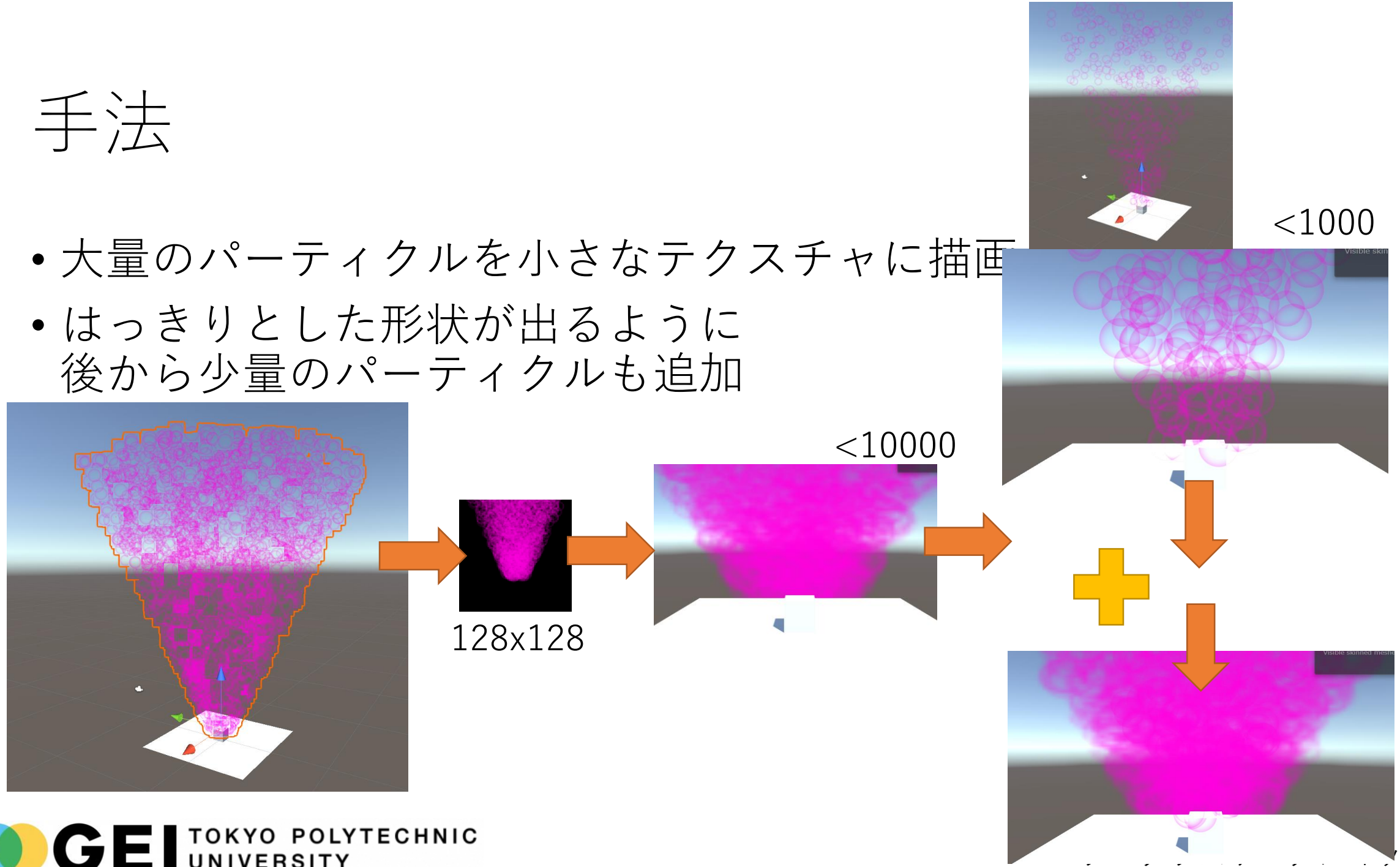
今回の目的

- 半透明をたくさん出したい
  - ボトルネックはたくさんの可能性があるが一つはフィル
    - 小さなバッファに描画して負荷を抑える



手法

- 大量のパーティクルを小さなテクスチャに描画
- はっきりとした形状が出るように
  後から少量のパーティクルも追加



オブジェクトとアセット

シーンを描画
(パーティクル以外)

Layerの設定でパーティクルを
描画しないように(バックミラーや
ブラーの作成を参考に)

RT(縮小バッファ)へ
パーティクルを描画

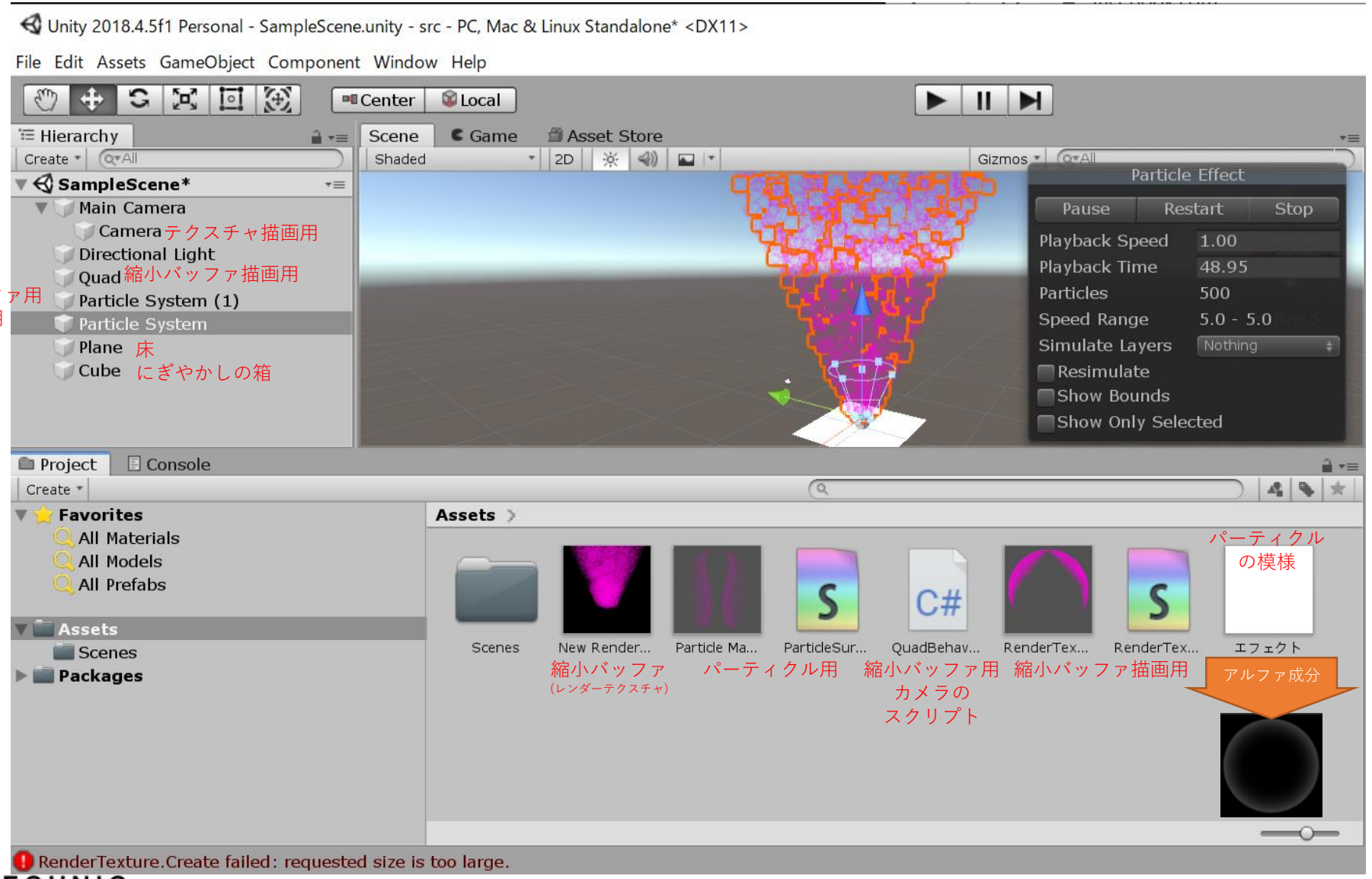
バックミラーやブラーの
作成を参考に

縮小バッファを描画

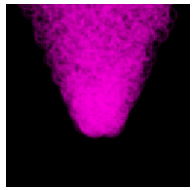
半透明描画を参考に

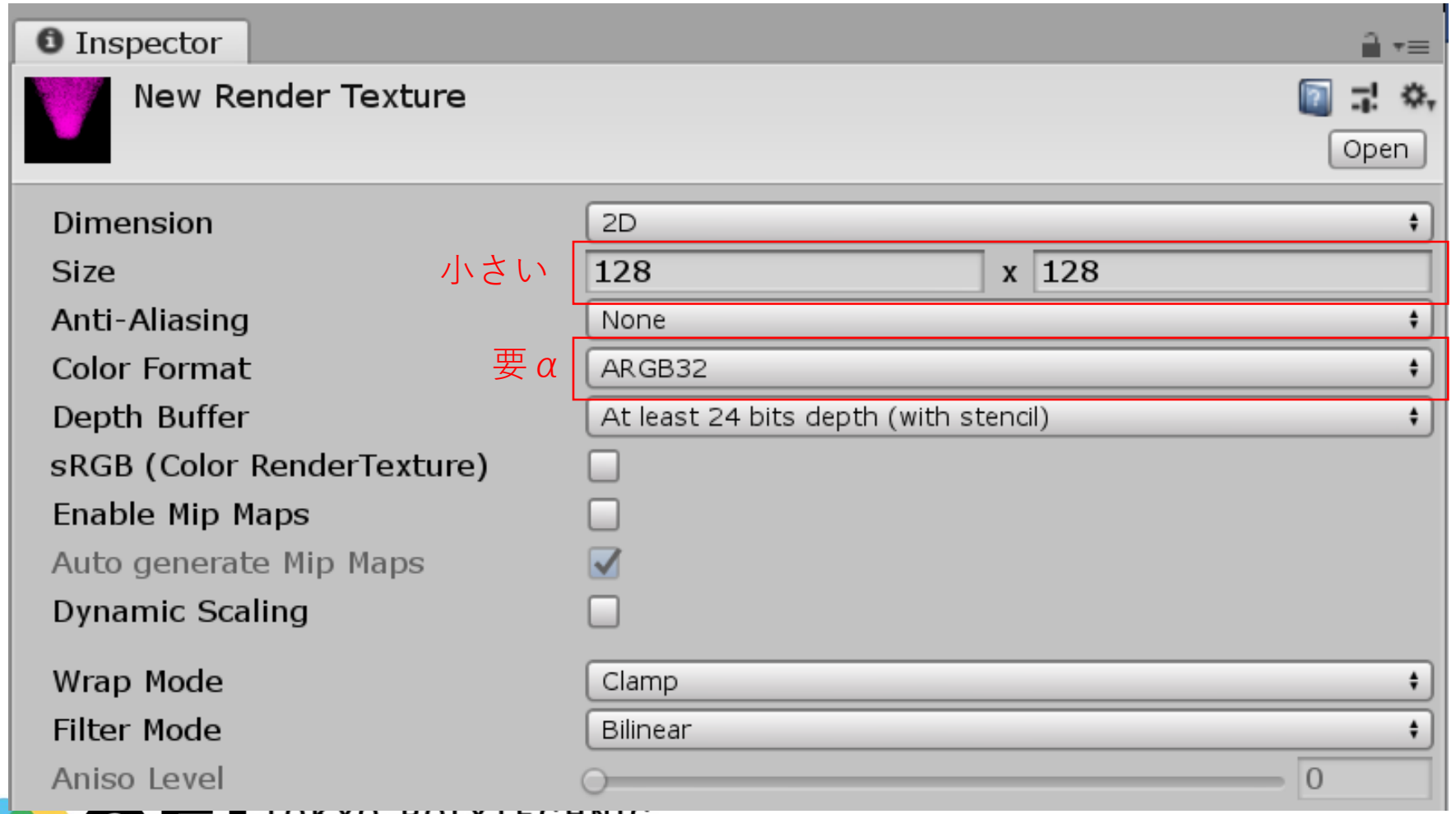
詳細なパーティクル
を追加

詳しく知りたいときは "Unity Shuriken"
で検索



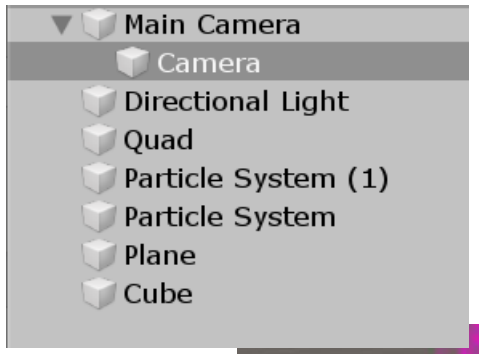
縮小バッファ

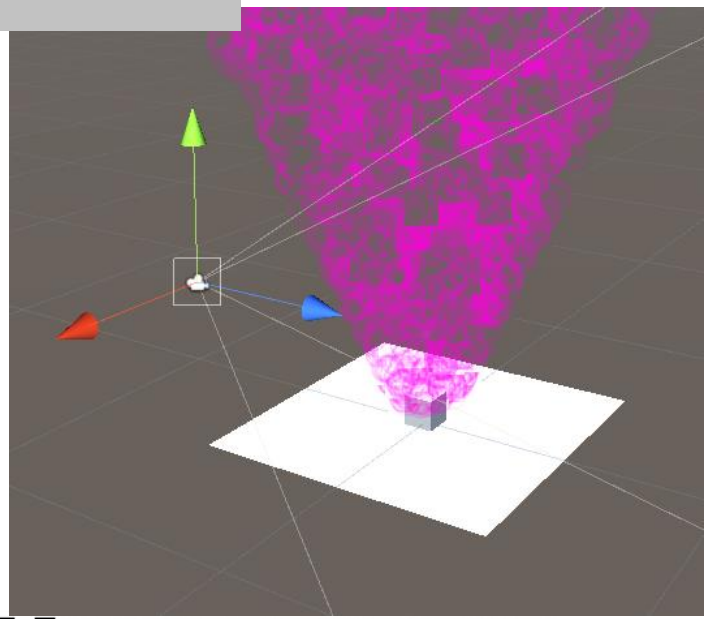


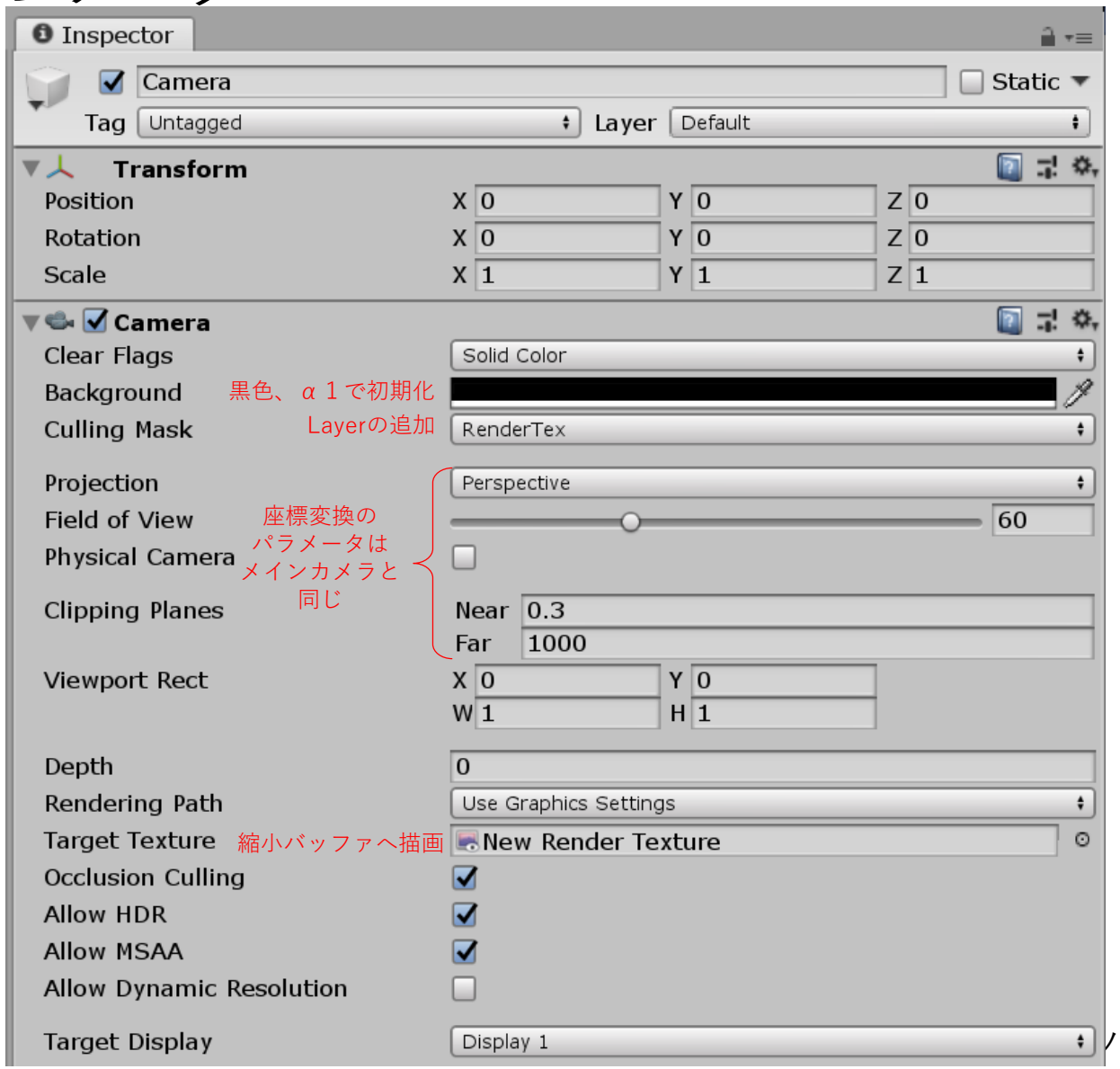


一辺が1/10程度だと、
面積的には1/100になる

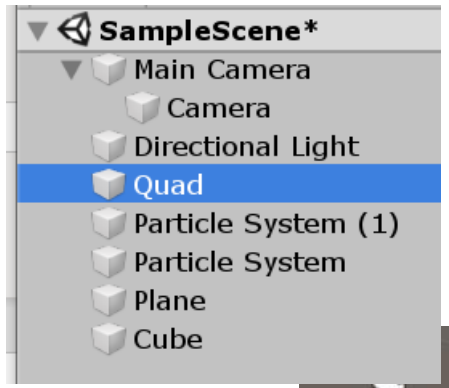
テクスチャ作成用カメラ

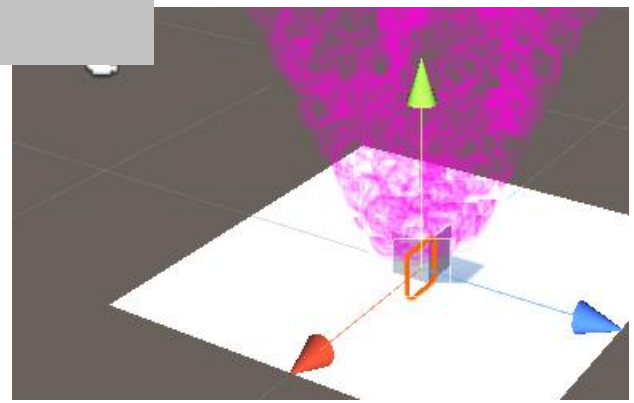


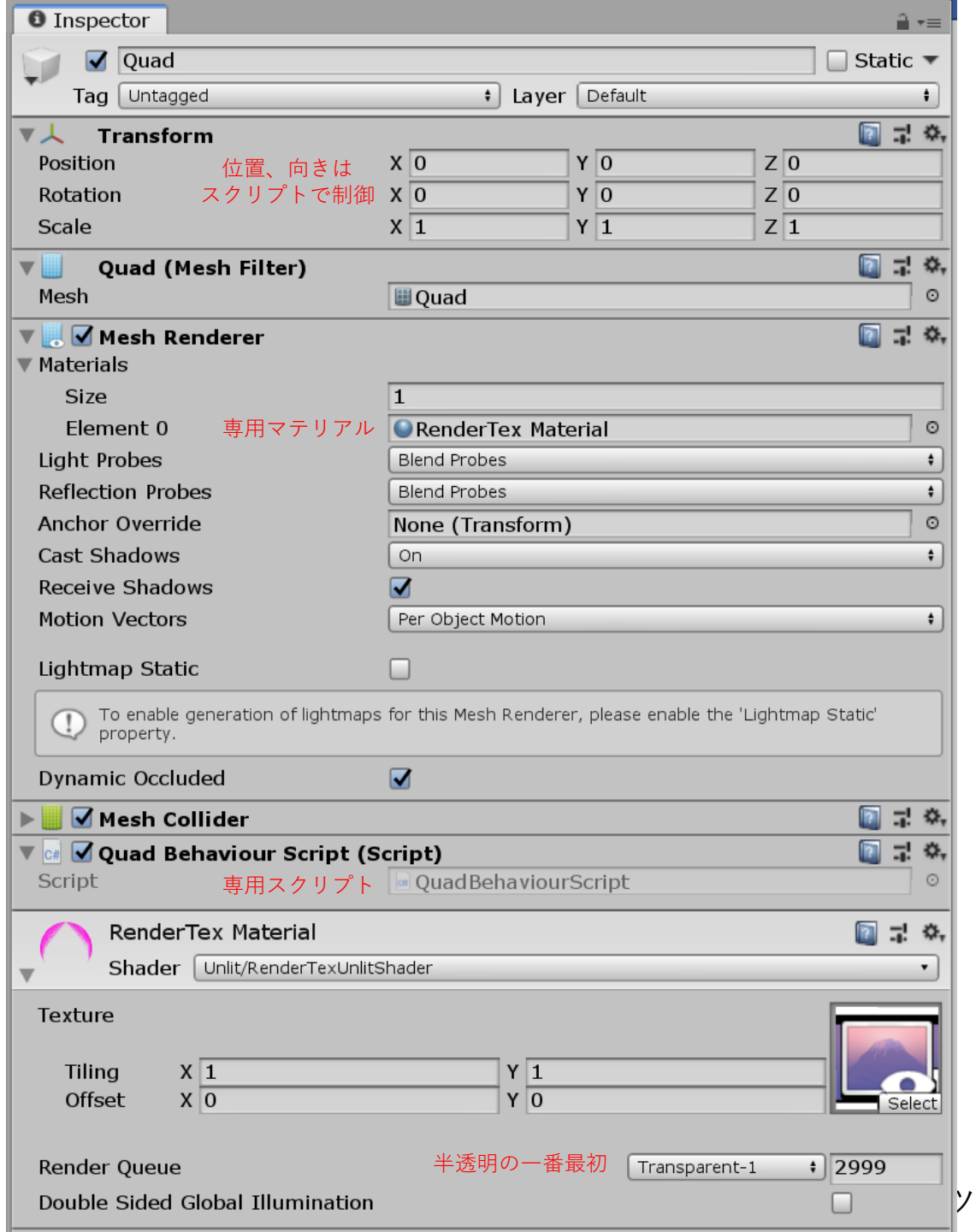




縮小バッファ描画


SampleScene*
Main Camera
Camera
Directional Light
Quad
Particle System (1)
Particle System
Plane
Cube

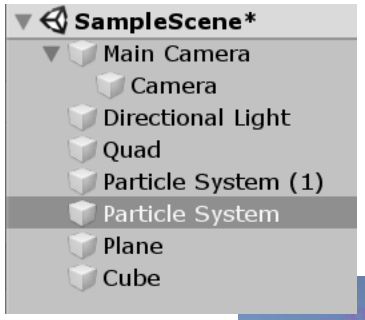


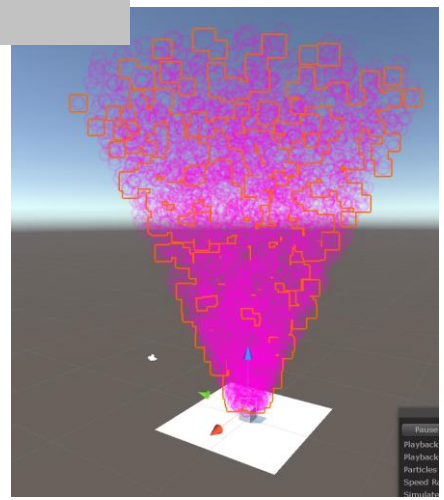

Inspector
Quad
Static
Tag
Untagged
Layer
Default
Transform
Position
位置、向きは
Rotation
スクリプトで制御
Scale
X 0 Y 0 Z 0
X 0 Y 0 Z 0
X 1 Y 1 Z 1
Quad (Mesh Filter)
Mesh
Quad
Mesh Renderer
Materials
Size
1
Element 0
専用マテリアル
RenderTex Material
Light Probes
Blend Probes
Reflection Probes
Blend Probes
Anchor Override
None (Transform)
Cast Shadows
On
Receive Shadows
Motion Vectors
Per Object Motion
Lightmap Static
To enable generation of lightmaps for this Mesh Renderer, please enable the 'Lightmap Static' property.
Dynamic Occluded
Mesh Collider
Quad Behaviour Script (Script)
Script
専用スクリプト
QuadBehaviourScript
RenderTex Material
Shader
Unlit/RenderTexUnlitShader
Texture
Tiling
Offset
X 1 Y 1
X 0 Y 0
Select
Render Queue
半透明の一番最初
Transparent-1
2999
Double Sided Global Illumination

縮小バッファカメラ用スクリプト

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
    0 references
5   public class QuadBehaviourScript : MonoBehaviour
6   {
7       // Start is called before the first frame update
        0 references
8       void Start()
9       {
10          Camera cam = Camera.main;
11
12          // 親を変更
13          transform.parent = cam.transform;
14
15          // 位置の調整
16          float distance = cam.farClipPlane - 3.0f;// farClipPlane までの距離でよいはずだが、少し短くないと表示されない
17          transform.localPosition = new Vector3(0,0, distance);最遠方に配置
18          transform.localRotation = new Quaternion();
19          // 画面にピッタリ一致するようにカメラからの距離から大きさを逆算
20          float inv_aspect = (float)Screen.width / (float)Screen.height;
21          float s = Mathf.Tan(0.5f * cam.fieldOfView * Mathf.Deg2Rad) * 2.0f * distance;
22          transform.localScale = new Vector3(inv_aspect*s, s, 1);
23      }
24
25      // Update is called once per frame
        0 references
26      void Update()
27      {
28
29      }
30  }
```

パーティクル

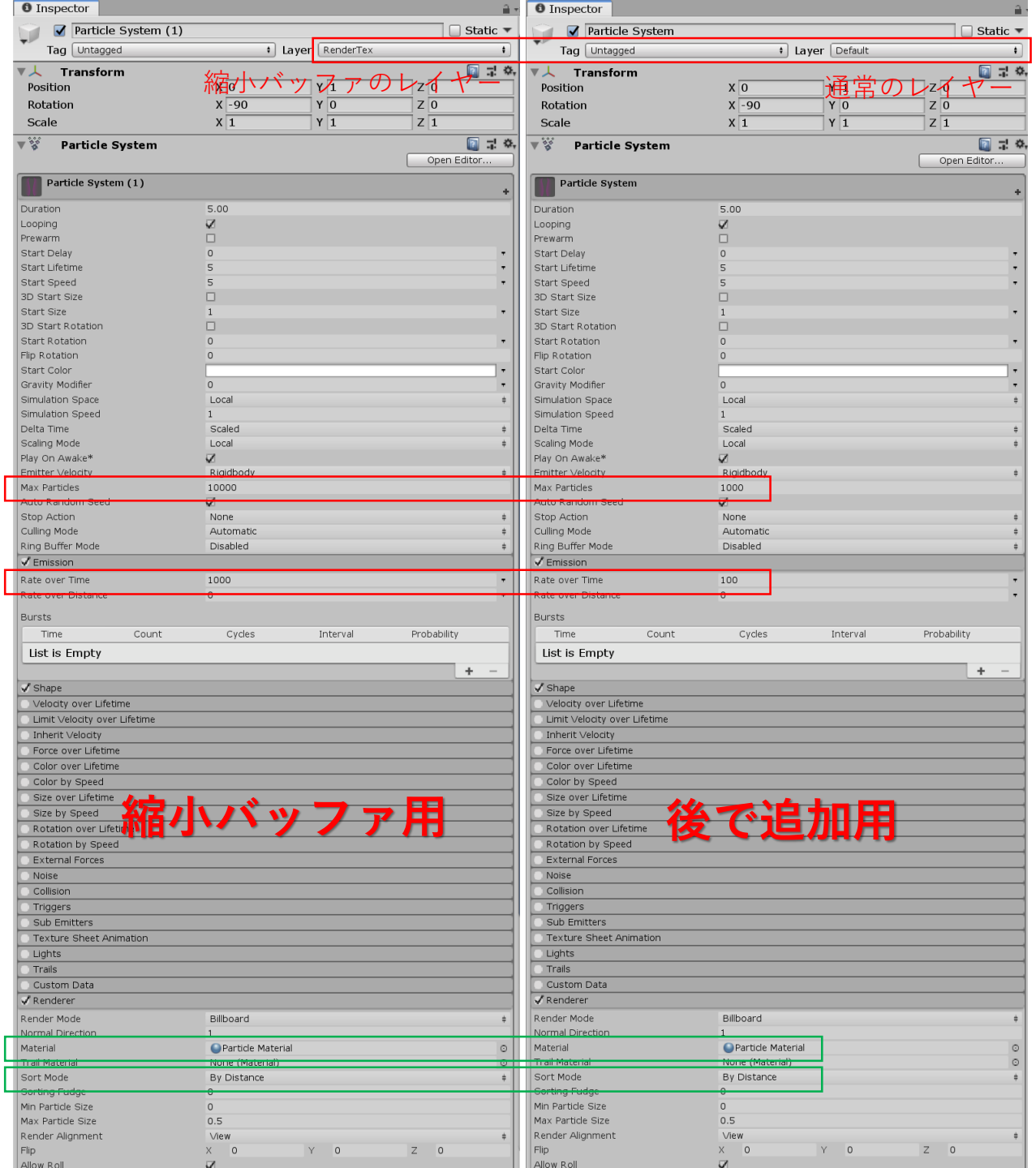




最大数

一度の放出数

同じマテリアル
距離でソート



シェーダ

$$color = \alpha_{src} color_{src} + (1 - \alpha_{src}) color_{dest}$$

$$\alpha = (1 - \alpha_{src}) \alpha_{dest}$$

複数の半透明を重ねた際に

$$c = \alpha_2 c_2 + (1 - \alpha_2)\{\alpha_1 c_1 + (1 - \alpha_1) c_{dest}\}$$
$$= \alpha_2 c_2 + (1 - \alpha_2)\alpha_1 c_1 + (1 - \alpha_2)(1 - \alpha_1) c_{dest}$$

なるが、これを

$c = c_{\text{縮小バッファ}} + \alpha_{\text{縮小バッファ}} c_{dest}$

とできるように色とαの合成の式を立てた

```
1   Shader "Custom/NewSurfaceShader"
2   {
3       Properties
4       {
5           _MainTex("Texture", 2D) = "white" {}
6           _Color("Color", Color) = (1,0,0,1)
7       }
8           SubShader
9           {
10              Tags { "RenderType" = "Transparent" "Queue" = "Transparent" }
11              Blend SrcAlpha OneMinusSrcAlpha, Zero OneMinusSrcAlpha
12              Pass
13              {
14                  CGPROGRAM
15                  #pragma vertex vert
16                  #pragma fragment frag
17
18                  #include "UnityCG.cginc"
19                  struct appdata
20                  {
21                      float4 vertex : POSITION;
22                      float4 uv : TEXCOORD0;
23                  };
24                  struct v2f
25                  {
26                      float2 uv : TEXCOORD0;
27                      float4 vertex : SV_POSITION;
28                      fixed4 color : COLOR;
29                  };
30                  sampler2D _MainTex;
31                  float4 _MainTex_ST;
32                  fixed4 _Color;
33
34                  v2f vert(appdata v)
35                  {
36                      v2f o;
37                      o.vertex = UnityObjectToClipPos(v.vertex);
38                      o.uv = TRANSFORM_TEX(v.uv, _MainTex);
39                      return o;
40                  }
41
42                  fixed4 frag(v2f i) : SV_Target
43                  {
44                      return tex2D(_MainTex, i.uv) * _Color;
45                  }
46                  ENDCG
47              }
48          }
49  }
```

縮小バッファ描画用シェーダ

- 前頁の計算を反映させる合成方法
  - 深度バッファへの書き込みをしないようにする方が良いですね…

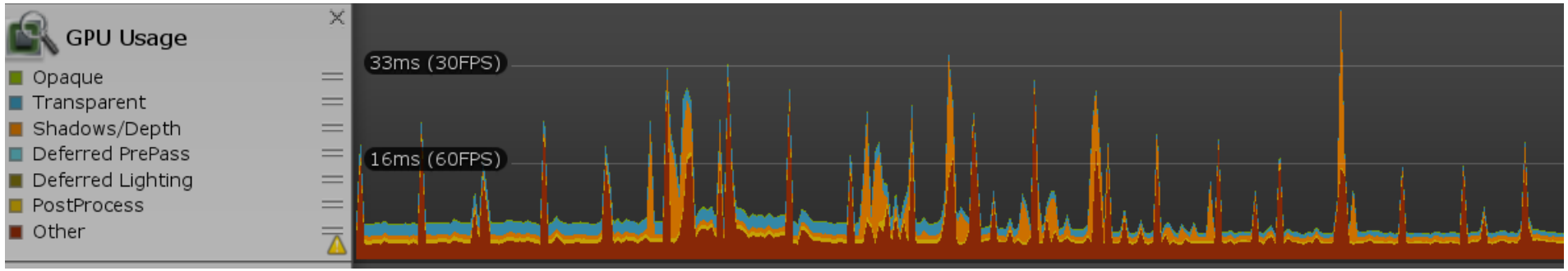
```
Shader "Unlit/RenderTexUnlitShader"
{
    Properties
    {
        _MainTex ("Texture", 2D) = "white" {}
    }
    SubShader
    {
        Tags { "RenderType"="Transparent" }
        Blend One SrcAlpha
        LOD 100

        Pass
        {
            CGPROGRAM
            #pragma vertex vert
            #pragma fragment frag

            #include "UnityCG.cginc"

            struct appdata
            {
                float4 vertex : POSITION;
                float2 uv : TEXCOORD0;
            };

            struct v2f
            {
                float2 uv : TEXCOORD0;
                float4 vertex : SV_POSITION;
            };

            sampler2D _MainTex;
            float4 _MainTex_ST;

            v2f vert (appdata v)
            {
                v2f o;
                o.vertex = UnityObjectToClipPos(v.vertex);
                o.uv = TRANSFORM_TEX(v.uv, _MainTex);
                return o;
            }

            fixed4 frag (v2f i) : SV_Target
            {
                // sample the texture
                fixed4 col = tex2D(_MainTex, i.uv);
                return col;
            }
            ENDCG
        }
    }
}
```

実行結果

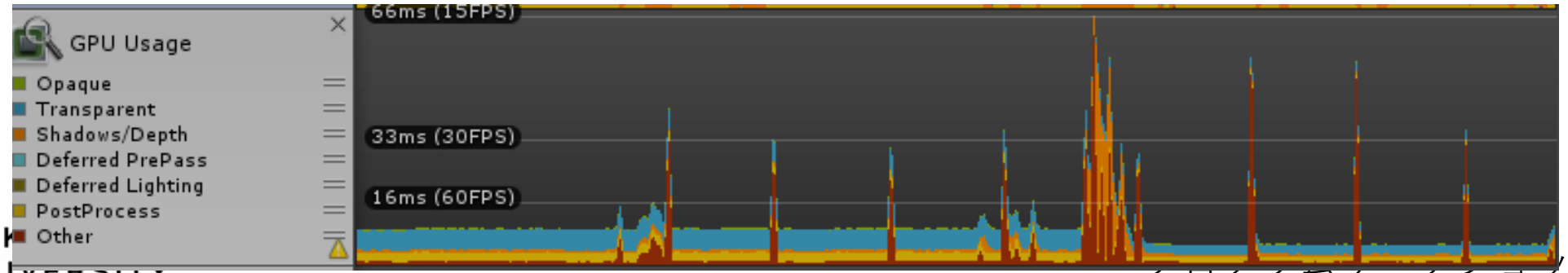
- デスクトップではこの設定では、さほど差は出ない
  - パーティクルを100倍にすると、大きな差が出るが、見た目がよろしくない

1024 x 1024　　128 x 128

Radeon RX 560

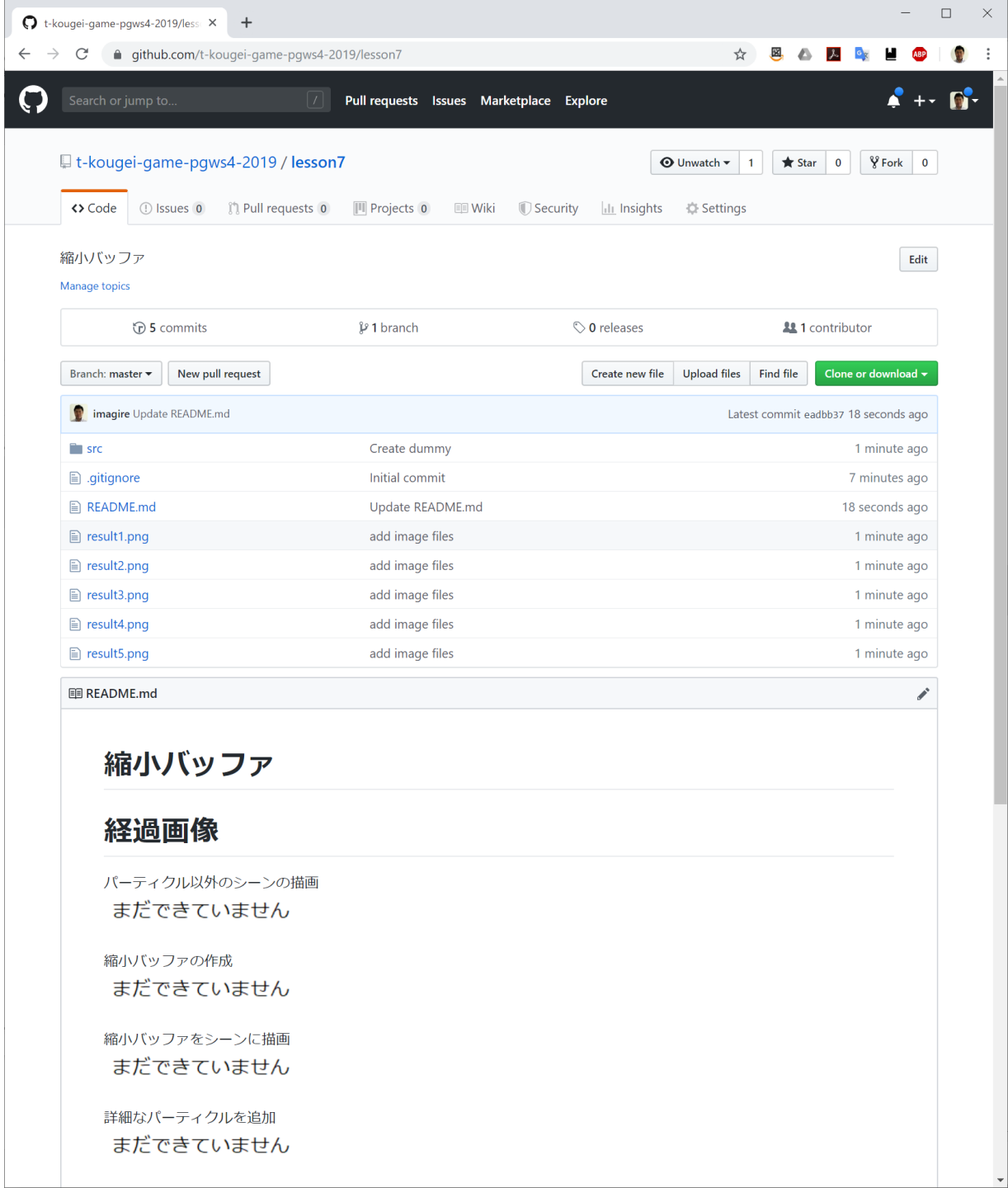


GeForce MX250



やってみよう

- できる限りすごいパーティクルエフェクトを作ってみよう
  - ライティング
  - 複数種類の追加



KOGEI TOKYO POLYTECHNIC UNIVERSITY