







检查器

Sphere sss 静态的

标签 Untagged 图层 Default

Transform

位置	X 0	Y 16.5	Z 0.072683
旋转	X 0	Y 0	Z 0
缩放	X -6.469674	Y -6.469674	Z -6.469674

Sphere (Mesh Filter)

网格 Sphere

Mesh Renderer

Materials 1

元素 0 yeee

光照

投射阴影 开启

接受阴影

贡献全局照明

接收全局照明 光照探测器

探测器

光照探测器 混合探测器

反射探测器 Blend Probes

锚点覆盖 无 (变换)

其他设置

运动矢量 每对象运动

动态遮挡

Sphere Collider

编辑碰撞器

是触发器

材质 无 (物理材质)

中心 X 0 Y 0 Z 0

半径 0.5

Yeee (Material)

Shader World of Zero/Hologram Edit...

添加组件

层级

01_CreateAttractor

- Directional Light
- Sphere
- Spheres
- Spheres (1)
- Spheres (2)
- Sphere sss**
- Cubee1 (2)
- Cubee1 (3)
- Cubee1 (4)
- Cubee1 (5)
- Sphere (1)
- Cubee1 (6)
- Cubee1 (7)
- Cubee1 (8)
- Cubee1 (9)

项目

Assets > Mesh > Materials

Favorites

- Assets
 - AudioPeer
 - Materials
 - Mesh
 - Materials
 - Prefabs
 - Scenes
 - Scripts
 - Textures
- Packages

experiment hologram hologramRays Horizontal Scene_Materia...

Yeee (Material)

```
38 public bool _emissionBuffer;
39 public bool _rotateBuffer;
40 public float _rotateSpeed;
41 private float _rotateTangentObjects;
42 public bool _scaleYOnAudio;
43 public bool _scaleBuffer;
44 [Range(0, 1)]
45 public float _thresholdScale;
46 public float _scaleStart;
47 public Vector2 _scaleMinMax;
48
49 // Start is called before the first frame update
50 void Awake()
51 {
52     _innerCircle = new Vector4(0, 0, 0, _innerCircleRadius);
53     _outterCircle = new Vector4(0, 0, 0, _outerCircleRadius);
54     _tangentCircle = new Vector4[_circleAmount];
55     _tangentObject = new GameObject[_circleAmount];
56     _material = new Material[_circleAmount];
57     for (int i = 0; i < _circleAmount; i++)
58     {
59         GameObject tangentInstance = (GameObject)Instantiate(_circlePrefab);
60         _tangentObject[i] = tangentInstance;
61         _tangentObject[i].transform.parent = this.transform;
62
63         _material[i] = new Material(_materialBase);
64         _material[i].EnableKeyword("_EMISSION");
65         if (_tangentObject[i].GetComponent<MeshRenderer>())
66         {
67             _tangentObject[i].GetComponent<MeshRenderer>().material = _material[i];
68         }
69         else
70         {
71             _tangentObject[i].transform.GetChild(0).GetComponent<MeshRenderer>().material = _material[i];
72         }
73     }
74 }
75
76 void PlayerInput()
77 {
78     _tsL = new Vector2(Input.GetAxis("Horizontal"), Input.GetAxis("Vertical"));
79     _tsLSmooth = new Vector2(
80         _tsLSmooth.x * (1 - _movementSmooth) + _tsL.x * _movementSmooth,
81         _tsLSmooth.y * (1 - _movementSmooth) + _tsL.y * _movementSmooth);
82 }
```

```
40 public _channel channel = new _channel ();
41
42
43 //Audio64
44 float[] _freqBand64 = new float[64];
45 float[] _bandBuffer64 = new float[64];
46 float[] _bufferDecrease64 = new float[64];
47 float[] _freqBandHighest64 = new float[64];
48 //audio band64 values
49 [HideInInspector]
50 public float[] _audioBand64, _audioBandBuffer64;
51
52
53 // Use this for initialization
54 void Start ()
55 {
56     _audioProfile = 0.5f;
57     _audioBand = new float[8];
58     _audioBandBuffer = new float[8];
59     _audioBand64 = new float[64];
60     _audioBandBuffer64 = new float[64];
61     _audioSource = GetComponent<AudioSource> ();
62     AudioProfile (_audioProfile);
63
64     //Microphone input
65
66     if (_useMicrophone)
67     {
68         if (Microphone.devices.Length > 0)
69         {
70             _selectedDevice = Microphone.devices[0].ToString();
71             _audioSource.outputAudioMixerGroup = _mixerGroupMicrophone;
72             _audioSource.clip = Microphone.Start(_selectedDevice, true, 10, AudioSettings.outputSampleRate);
73         }
74         else
75         {
76             _useMicrophone = false;
77         }
78     }
79     if (!_useMicrophone)
80     {
81         _audioSource.outputAudioMixerGroup = _mixerGroupMaster;
82         _audioSource.clip = _audioClip;
83     }
84
85     _audioSource.Play();
```