**Getting Started - Chapter 6 - Particle Spray**

**@@启程-第六章-粒子喷射@@**

**Particle Spray**

**@@粒子喷射@@**

In this case particles are small sprites emitted in a cluster, or cloud, to simulate used to simulate fire, smoke, water, or even faery dust.

@@在这个例子里粒子是通过一个发射团或发射云，发射出的大量小精灵，用来模拟火焰、烟雾或者尘土。@@

The basis for a cloud of particles is the *ParticleSystem*. With this we specify the number of particles to use

@@粒子云的基础是“粒子系统对象”。通过它我们可以这样设置粒子的数量@@

const particleSystem = new BABYLON.ParticleSystem("particles", 5000, scene); //scene is optional

Particles are emitted from a specifically defined region. Each particle is given a lifetime and when reached it is reused and re-emitted.

@@粒子从一个指定的区域射出。每个粒子都将被分配生存时间，并将在到达生存时间后被回收重新发射。@@

They are given a texture, which obviously governs their appearance,

@@它们将被分配纹理，这纹理显然是用来控制它们的外观的，@@

particleSystem.particleTexture = new BABYLON.Texture("url to image", scene);

The basic emitter region is with a box of given dimensions around a specified point;

@@最基础的发射区是一个盒状区域，这个区域通过指定一个点，然后设置区域边界到点的距离来确定。@@

particleSystem.emitter = new BABYLON.Vector3(-4, 0.8, -6); // the point at the top of the fountain 喷泉顶部的点

particleSystem.minEmitBox = new BABYLON.Vector3(-0.01, 0, -0.01); // minimum box dimensions 盒子的最小范围（左、下、后）

particleSystem.maxEmitBox = new BABYLON.Vector3(0.01, 0, 0.01); // maximum box dimensions 盒子的最大范围（右、上、前）

Up to three colors can be given in the form (r, g, b, a) for red, green, blue, and alpha. Alpha goes from 0 for fully transparent up to 1 for fully opaque. The first two are combined or blended to set the color of the particle. The method of blending can be specified. The third has the property *colorDead* and is of use when recycling of the particles is set to off. This third property is not needed for our fountain.

@@可以以（r，g，b，a）的形式为粒子系统设置三个颜色属性（color1、color2、color3），rgb分别表示红、绿、蓝，a表示不透明度，不透明度为0表示完全透明，不透明度为1表示完全不透明。前两个颜色属性会相互混合产生粒子的实际颜色，混合的方法则是可以定制的。第三个颜色属性则是“死亡颜色”，这种颜色将在关闭粒子回收，而粒子生命时间将要耗尽时使用。在我们的喷泉中用不到第三个属性。@@

particleSystem.color1 = new BABYLON.Color4(0.7, 0.8, 1.0, 1.0);

particleSystem.color2 = new BABYLON.Color4(0.2, 0.5, 1.0, 1.0);

particleSystem.blendMode = BABYLON.ParticleSystem.BLENDMODE\_ONEONE;

The size of a particle and its lifetime are within the range set by minimum and maximum values.

@@粒子的尺寸和生存时间将在最小值和最大值的范围内随机插值产生。@@

particleSystem.minSize = 0.01;

particleSystem.maxSize = 0.05;

particleSystem.minLifeTime = 0.3;

particleSystem.maxLifeTime = 1.5;

We also need to set the number of particles emitted per second. Too fast an emittance rate with a long lifetime can result in gaps in the emission of particles.

@@我们还需要设置每秒的粒子发射数量。太快的粒子发射率和较长的生命时间可能导致粒子发射的中断。@@

particleSystem.emitRate = 1500;

We also set two possible directions for the particles and affect the speed with which the particle travels by giving a minimum and maximum power value and an update speed.

@@我们还为粒子系统设置了两个可能的发射方向（实际方向在之间插值），并且通过设置最低力度、最高力度、更新速度值来影响粒子的运动速度。（更新速度是一种整体的速度调整，它会同时影响粒子的移动速度、发射速度、精灵动画速度）@@

particleSystem.direction1 = new BABYLON.Vector3(-1, 8, 1);

particleSystem.direction2 = new BABYLON.Vector3(1, 8, -1);

particleSystem.minEmitPower = 0.2;

particleSystem.maxEmitPower = 0.6;

particleSystem.updateSpeed = 0.01;

Setting a negative value for graving in the y component ensures the particles fall downwards.

@@给重力属性的y分量设置一个负值，可以使得粒子下落@@

particleSystem.gravity = new BABYLON.Vector3(0, -9.81, 0);

Finally we can start the fountain

@@最后我们可以启动喷泉了@@

particleSystem.start();

Here is an example of our fountain:

@@这是我们喷泉的一个例子：@@

A Basic Particle Fountain

基础粒子喷泉

[https://playground.babylonjs.com/#TC31NV#4](https://playground.babylonjs.com/#4)

const createScene = function () {

    const scene = new BABYLON.Scene(engine);

    const camera = new BABYLON.ArcRotateCamera("Camera", 3 \* Math.PI / 2, Math.PI / 2, 70, BABYLON.Vector3.Zero());

    camera.attachControl(canvas, true);

    const light = new BABYLON.HemisphericLight("hemi", new BABYLON.Vector3(0, 1, 0));

    const fountainProfile = [

        new BABYLON.Vector3(0, 0, 0),

        new BABYLON.Vector3(10, 0, 0),

        new BABYLON.Vector3(10, 4, 0),

        new BABYLON.Vector3(8, 4, 0),

        new BABYLON.Vector3(8, 1, 0),

        new BABYLON.Vector3(1, 2, 0),

        new BABYLON.Vector3(1, 15, 0),

        new BABYLON.Vector3(3, 17, 0)

    ];

    //Create lathe

    const fountain = BABYLON.MeshBuilder.CreateLathe("fountain", {shape: fountainProfile, sideOrientation: BABYLON.Mesh.DOUBLESIDE}, scene);

    fountain.position.y = -5;

    // Create a particle system

    var particleSystem = new BABYLON.ParticleSystem("particles", 5000, scene);

    //Texture of each particle

    particleSystem.particleTexture = new BABYLON.Texture("textures/flare.png", scene);

    // Where the particles come from

    particleSystem.emitter = new BABYLON.Vector3(0, 10, 0); // the starting object, the emitter

    particleSystem.minEmitBox = new BABYLON.Vector3(-1, 0, 0); // Starting all from

    particleSystem.maxEmitBox = new BABYLON.Vector3(1, 0, 0); // To...

    // Colors of all particles

    particleSystem.color1 = new BABYLON.Color4(0.7, 0.8, 1.0, 1.0);

    particleSystem.color2 = new BABYLON.Color4(0.2, 0.5, 1.0, 1.0);

    particleSystem.colorDead = new BABYLON.Color4(0, 0, 0.2, 0.0);

    // Size of each particle (random between...

    particleSystem.minSize = 0.1;

    particleSystem.maxSize = 0.5;

    // Life time of each particle (random between...

    particleSystem.minLifeTime = 2;

    particleSystem.maxLifeTime = 3.5;

    // Emission rate

    particleSystem.emitRate = 1500;

    // Blend mode : BLENDMODE\_ONEONE, or BLENDMODE\_STANDARD

    particleSystem.blendMode = BABYLON.ParticleSystem.BLENDMODE\_ONEONE;

    // Set the gravity of all particles

    particleSystem.gravity = new BABYLON.Vector3(0, -9.81, 0);

    // Direction of each particle after it has been emitted

    particleSystem.direction1 = new BABYLON.Vector3(-2, 8, 2);

    particleSystem.direction2 = new BABYLON.Vector3(2, 8, -2);

    // Angular speed, in radians

    particleSystem.minAngularSpeed = 0;

    particleSystem.maxAngularSpeed = Math.PI;

    // Speed

    particleSystem.minEmitPower = 1;

    particleSystem.maxEmitPower = 3;

    particleSystem.updateSpeed = 0.025;

    // Start the particle system

    particleSystem.start();

    return scene;

}

The fountain in place

@@让喷泉就位@@

Add the Spray

添加喷射效果

https://playground.babylonjs.com/#TC31NV#4

Let's now provide an on-off click event.

@@现在让我们提供一个开关点击事件@@