Game Special Effect Simulation Based on Particle System of Unity3D

Bingqing Zhang
School of Computer Science
Communication University of China
Beijing, China
bq_zhang_cuc@163.com

Wenfeng Hu
School of Computer Science
Communication University of China
Beijing, China
huwf@cuc.edu.cn

***Abstract*—Along With the progress of computer games, a considerable variety of game has appeared. There is an increasing demand of digital entertainment. It is common for game players to pursuit a better game interaction experience and visual experience. Game art designers pay more attention to achieve realistic effects. In these years, particle effects are widely used in game development. In this article, through the analysis of the basic principle and application method of Unity3D particle system, we design and implement common game scene effects and characters skills effects, mainly simulation and realize the effect of rain fall and design the Special "lightning" effects of characters skills.**

Keywords—Unity 3D; Particle system; Game effects

I. INTRODUCTION

As the game industry vigorous development, game production level is higher and higher, the performance of the game scene and the special effects becomes more and more close to reality.

Game effects, it can be understood as the special effect in the game , one of the most direct feeling is the game of light and shadow, it has a lot of specific segmentation in the game , for example, characters special skills effects, simulation waterfalls or deciduous effects, UI effects, etc. In some large standalone game, we can often see very lifelike scene simulation, like illumination effect, fluid simulation, imitation of petals flying and emulation of weather changes. In the shooting games, it frequently showed some effects like explosion and the bullet trajectories, these interactive elements are difficult to experience personally in real life, but it can be simulated in the virtual game world.

In recent years, more and more game players prefer game visual experience, game special effects become the main means to improve the game visual experience and attract people attention, It provides good atmosphere for players to appreciate the art of the game and help game production promote game easily.

Unity3D,as the mainstream of game engine, has a complete particle system, it can help game developers make brilliant game effects, in this article ,we designed two kinds of game special effects include scene effects and character skills effects, provided concrete realization method of the rain effects and special lighting effects.

II. BASIC PRINCIPLES OF PARTICLE SYSTEM

Particle system is not a static system. Over time, the particles in the system are constantly changing the shape, moving continuously, and new particles are added to the system, while the old particles will disappear. Each particle in the system has a certain life cycle, including: "produce", "activity" and "death". Each parameter related to the particle is controlled by a random process. Because of the characteristics of the particle system, it has a good effect on the construction of "fuzzy" objects[1].

The process of establishing the particle system is as follows:

- The hypothesis of particles composition. Particles in a particle system have a continuous or discrete ways in its space, and run constantly, has distribution in space and time.
- The hypothesis of the relationship between particles . One is the particle in particle system does not intersect with any other objects in the scene, the other is non-intersect and impenetrable between particles[2].
- The hypothesis of particle properties. Each particle in the system is not abstract, they all have a number of properties.
- The life mechanism of particles. Each particle in a particle system has a lifetime.
- The motion mechanism of particle. Particles is always moving in their lifetime.
- The algorithm of drawing particle.

III. PATICLE SYSTEM OF UNITY3D

A particle system is a technique in game physics, motion graphics, and computer graphics that uses a large number of very small sprites, 3D models, or other graphic objects to simulate certain kinds of "fuzzy" phenomena, which are otherwise very hard to reproduce with conventional rendering techniques-usually highly chaotic systems, natural phenomena, or processes caused by chemical reactions[3]. Particle system is not a simple static system, with the passage of time, some particles constantly change shape, continuously move, and also have new particles to join the system.

In order to simulate the process of growth and death, each particle is endowed with certain life cycle, it will go through birth, growth, aging and death. At the same time, in order to make the object has a good randomness, some parameters related to the particle will be controlled by random data[4]. In simple terms, the main functions of the particle system includes three modules, they are particle emitters, impact part and renderer, The emitter launch particles, the impact part simulate element which affect particles motion state ,such as wind or gravity. The renderer structure particles information including some basic properties such as position, direction, scale, color, UV offset into geometry data, and then drawn by calculation, make them appear on the screen.

Particle system is a dynamic process modeling method, as shown in the Fig.1 below.

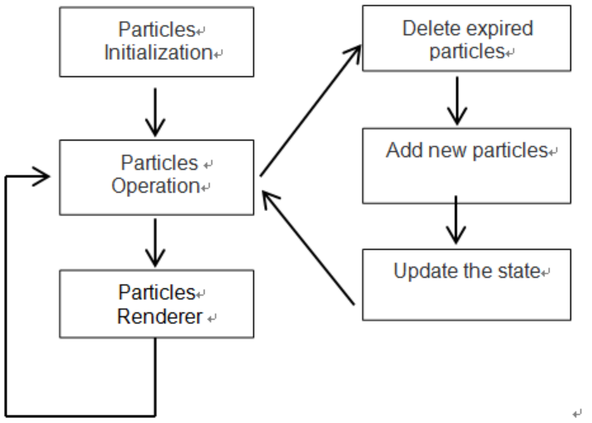


Fig.1. The structure of Unity3D particle system.

According to the above, we can see that system initialization mainly allocates memory space for particle system, sets the particle parameters such as particle volume and particle life . Particles operation contains the characteristics of simulation objects. Particles renderer translates the physical representation of the particle system into graphical representation[5].

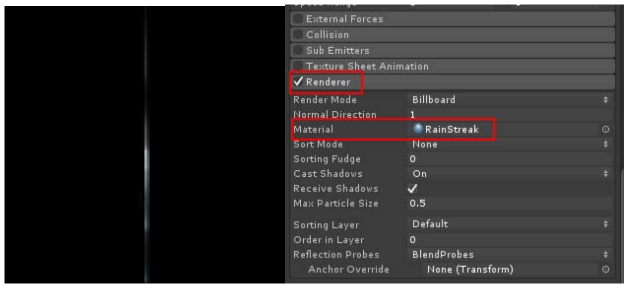
IV. Development Of Game Effects

The main visual effect in game is game scenes effect and characters skills effect, in most of game scenes, there blend different elements according to different art style. Here are concrete implementation process of two kinds of game effects.

A. *Development of "rain fall" effect*

This effect design can be broken down into two parts, one part simulate the process of rainfall, the other part make the animation of ripples .

First of all , create a new particle system in Unity3D, add a new material, apply the Fig. 2(a) below to this material named "RainStreak", and then assign it to our system renderer material as shown in Fig.2(b) below:

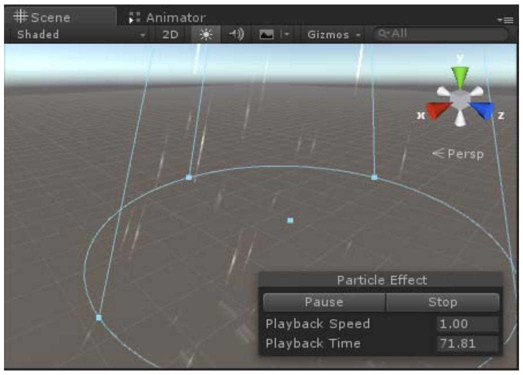


```
Render Mode          Billboard
Normal Direction     1
Material             RainStreak
Sort Mode            None
Sorting Fudge        0
Cast Shadows         On
Receive Shadows      ✓
Max Particle Size    0.5
Sorting Layer        Default
Order in Layer       0
Reflection Probes    BlendProbes
  Anchor Override      None (Transform)
```

(a) (b)

Fig.2. (a) The original picture of rain streak.(b) The operation panel of particle system

So now, each particle looks like raindrop, after this ,we need to adjust the basic parameters of particles, including launch position, direction, speed and particle size , etc. When we finished, we will get a complete rain model as shown in Fig.3 below.



```
Playback Speed    1.00
Playback Time     71.81
```

Fig.3 A simple model of rain drop

The environment of rainy day is very complicated in real life. The interaction between the rain and the surrounding objects may produce many kinds of visual effects. When the rain fell on the ground, there would appear splash water, we need to simulate ripple effect through the animation. Create a new plane, assign Fig.4 below to this plane, and then add animation component.

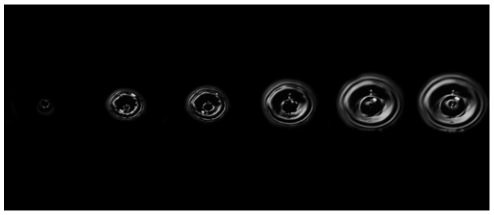
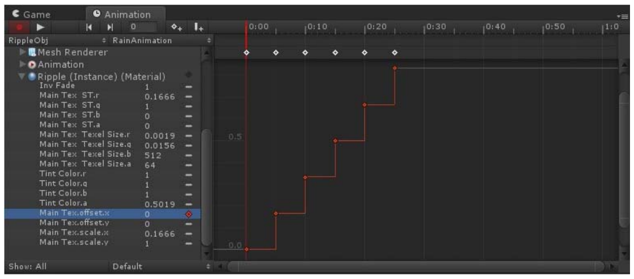


Fig.4 The orginal picture of splash

Using the animation editor, make the key frame animation with Fig.4, adjust its sequence and playback speed until it produces realistic ripple animation. The animation editor is shown in Fig.5 below.



```
RippleObj                      RainAnimation
  Mesh Renderer
  Animation
  Ripple (Instance) (Material)
    Inv Fade                   1
    Main Tex  ST.r             0.1666
    Main Tex  ST.q             1
    Main Tex  ST.b             0
    Main Tex  ST.a             0
    Main Tex  Texel Size.r     0.0019
    Main Tex  Texel Size.q     0.0156
    Main Tex  Texel Size.b     512
    Main Tex  Texel Size.a     64
    Tint Color.r               1
    Tint Color.q               1
    Tint Color.b               1
    Tint Color.a               0.5019
    Main Tex.offset.x          0
    Main Tex.offset.y          0
    Main Tex.scale.x           0.1666
    Main Tex.scale.y           1
Show: All                      Default
```

Fig.5 The animation editor of ripple

When we finished ripple animation, we need to write a script to control the stop time of this animation, the script named "RippleDestroy" :

```
using System. Collections;
public class RippleDestroy : MonoBehaviour
{
    public void DestroyMe()
    {
      Destroy(gameObject);
    }
}
```

So now ,we can run our system, it shows us random ripple like below:

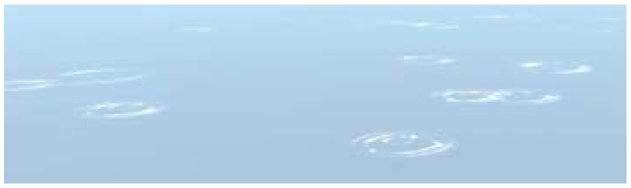


Fig.6 Final effect of ripple

We need this animation random plays on the ground, control it by script below:

```
public class rippleFX : MonoBehaviour {
    public GameObject rippleObj;
    int ti =0;
    void Update () {
     ti++;
     if (ti >= 5) {
       GameObject tempObj = Instantiate(rippleObj) as GameObject;
       tempObj.transform.parent = gameObject.transform;
       tempObj.GetComponent<Animation>().Play();
       tempObj.transform.position = transform.position+new Vector3(Random.Range(10,10),0,Random.Range(10,-10));
         }
       }
    }
```

Add a sky box for this system. The end result is shown below in Fig.7.

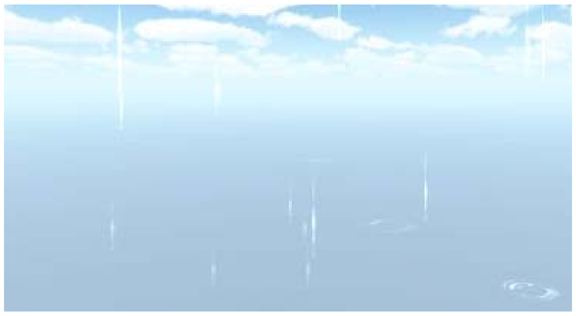


Fig.7 "rain fall"effect

B. *Development of "special lighting"effect*

The most common graphical effects in RPG games or fighting games is characters skill effect, such effects are usually have special dazzling visual effects, each game hero has different skills, each kind of skill has different effects, when the players control the hero, the effect is shocking and the experience infinite pleasure. In this paper, we take "special lighting" effect as an example, studying how to develop dynamic lighting effect by particle system of Unity3D.

Before we design this effect, we should understand some concept of particles. Once the particles are generated, the life cycle is given, it will decrease with the motion of the particle. When the cycle is reduced to 0, the particles die and they should be removed from the system[1]. In the actual application, we also can use other ways to measure the particles of death. For example, when the color and transparency of the particle is lower than the system setting value or the movement of the particles is beyond the specified area, these conditions can be considered as the actual requirements that the particles have died and these particles can be removed from the system[6]. As shown in figure below, it is particle motion process.

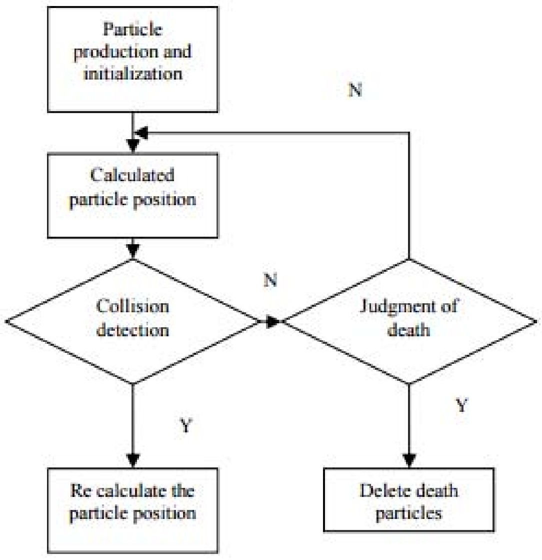

Particle production and initialization
N
Calculated particle position
N
Collision detection
Judgment of death
Y
Y
Re calculate the particle position
Delete death particles

Fig.8 The particle motion process.

Create a new particle system in Unity3D, we need to design a radiation effect that spread around the center, launch multiple lighting from one point. So we should adjust the basic information of the particle emitter, including life cycle, emission speed, maximum particle quantity and so on. One of the important steps is to change the shape of particle system to sphere, and then narrow its radius so that we can simulate launching point as shown below:

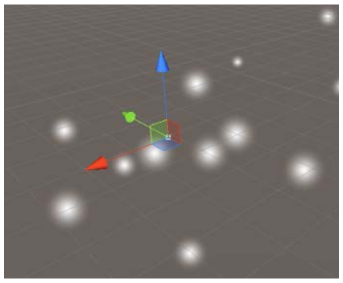
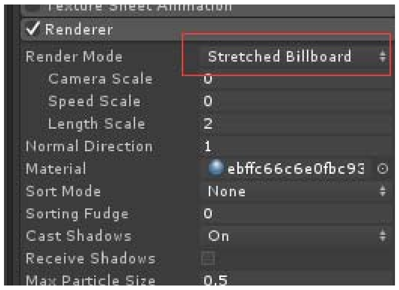


Fig.9 launching point of lighting effect

Another important factor to consider is render mode. Unity3D has five kinds of render mode, we need choose the suitable mode according to different situations. Based on this, we choose the stretched billboard as our render mode. As shown in figure 10.



```
✓ Renderer
Render Mode            Stretched Billboard
   Camera Scale        0
   Speed Scale         0
   Length Scale        2
Normal Direction       1
Material               ebffc66c6e0fbc93
Sort Mode              None
Sorting Fudge          0
Cast Shadows           On
Receive Shadows        ☐
Max Particle Size      0.5
```

Fig.10 Modify render mode

This mode has big different from default mode, it can comprehensively reflect the state of lighting motility, from this, we can see that each particle seem to be stretcher and become more longer. Now the system can be used to simulate the motion of lighting.

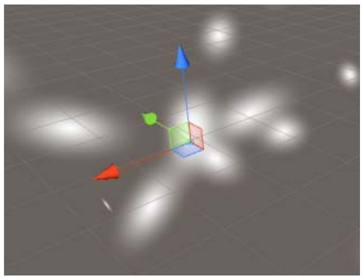
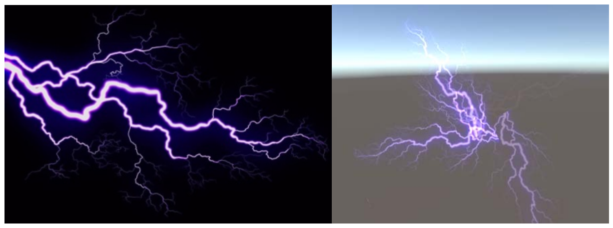


Fig.11 The Effect of stretched billboard

According to the needs of effect, next step we will design and implement the art of lighting effect. Create a new material ball, set Fig.12(a) as the material, and then apply it to render material of particle system, so each default particle looks like lighting as shown in the Fig.12(b). If the requirements should change in later part of the design, in order to meet the design requirements, the developers can be able to adjust the system parameters, such as size, quantity, color, etc.



(a) (b)

Fig.12 (a) The orginal picture of lighting . (b) The final effect of "special lighting" in Unity3D

V. CONCLUSIONS

Game effect, as a major section of computer graphic art, is being applied intensively in fields of game developing projects. It is difficult to see some special effects in real life , but players can enjoy in the game. Unity3D particle system is very perfect and mature, it has good stability and excellent interactive interface, provide amazing user experiences for game developers.

This paper discussed how to carry out two kinds of common effects based on Unity3D particle system and reached the desired objective of the study. Therefore, add new interaction to this effect system will become a study emphasis in further research.

REFERENCES

[1] Lei Shi, Wenyong Wang, "The status quo and realization of Waterfall simulation based on particle system".International Conference on Computer Science and Network Technology(ICCSNT), 2015 4th, pp. 91-93.

[2] Lei Shi, Wenyong Wang , "Study on algorithm for firework simulation based on particle system" International Conference on Computer Science and Network Technology(ICCSNT), 2013 3rd, pp. 231-234.

[3] William T.Reeves, "Particle System:A Technique for Modeling a class of Fuzzy Object" ACM Transactions on Gtaphics,April 1983.

[4] XiaoJuan Wang, "Simulation of dynamic firework based on particle system",Journal of Qinghai University,Vol .27 No 4,Aug 2009.

[5] Qing Yang, "The research and implementation of dynamic simulation based on particle system",Xidian University,2010.

[6] Yao Xiong, "3D special effect simulation based on particle system of unity3D",Software Guide,Vol.11 No.11,Nov 2012.