

Lightning effects technique comparison in unity 3D

Syauqi Hidayatul Hakim^{a,1,*}, Wahyu Andhyka Kusuma^{b,2}

^{ab} Univesity of Muhammadiyah Malang,

¹syauqi_437238@webmail.umm.ac.id; ²kusuma.wahyu.a@gmail.com

* *corresponding author*

ABSTRACT

Game developer team especially indie game developer mostly has small number of members and sometimes unbalanced team composition. Sometimes it has more artist than programmer, sometimes it is the opposite. At the same time, consumer will choose game that has a beautiful graphics. This demands of good graphic game can cause problem to unbalanced developer team. Therefore it is needed to know techniques that can utilize many artist or many programmer to create good graphic games. In this articles, we introduce some techniques to create special lightning effects of characters skills in Unity3D. Some of them can be created with good artistic skill. It means that this technique make use of your artist. The other technique created procedurally using Line Renderer from Unity 3D. This technique is friendly for artist since it utilize programmer. At the end of this article, all of those technique will be tested in their performance of its lightness and speed.

Keywords:

Unity 3D

Lightning

Game Effects

I. Introduction

Game consumer are always demanding a good game. They prefer realistic game graphically but processing lightly at the same time. Take lightning as an example. Lightning in game not always used as scene effects but can also used as skill effects as was introduced by Zhang and Hu [1]. But the nature of electricity is complicated and chaotic [2]. Thus it is hard to make lightning effects the way developer wanted.

Unity 3D [3][4] is a game engine that has been used by many game developer. In Unity 3D, there is a built-in particle system that can be utilized to make game effects with almost no code. It is up to the user familiarity of the system to make many kinds of effects. While it is true, not all of lightning characteristics can be applied using particle system. There is other way to create lightning effects in Unity, that is generated procedurally from mesh or line renderer in Unity.

In this article there will be four technique to create lightning skill effects. First is lightning which introduced by Zhang and Hu. Second is particle system with addition of noise module from Unity 3D. Third and fourth will be procedurally generated using Line Renderer in Unity 3D. At the end of this article, the performance of all of those technique will be compared.

II. Method

There are four tehcnique that will be introduced. The first two will use particle system while the other two will use Line Renderer from Unity with difference approach.

A. Particle System

There are two technique that use particle systems of Unity 3D to create lightning skill effects. Firstly is Zhang and Hu lightning. The particle system is customized so that all particle will be emitted from the center to every direction spherically. The second technique is using noise module from particle system component. The noise will make each particle looks chaotic. Noise module in Unity is based on two type of noise, curl noise for it's movement and perlin noise as parameters [5]. The parameters are Strength to adjust speed and range, frequency to change how chaotic it is, and octaves as the layers of noise. Adjusting strength and frequency only is enough to mimic lightning effects, while octaves can be pretty heavy in performance. Other thing can be added to make it more similar to lightning like colors and lighting.



B. Line Renderer

Line renderer in Unity 3d works like line in computer graphic generally. There are dots or vertices (in Unity 3d case, it is called elements) and lines. Define the position of two vertices and Unity will generate the lines automatically. Thus, the technique will modify those position over time to create effects similar to lightning. With addition of lighting and color, it is enough to make the lightning realistic.

C. Test

Average CPU consumption each seconds and average FPS rates each seconds will be measured and compared. Each lightning effects firstly will be created equal. Like the number lightning objects. One objects is not enough to see the different of performance. Number of strike (particle) each seconds in each lightning objects will be adjusted equally.

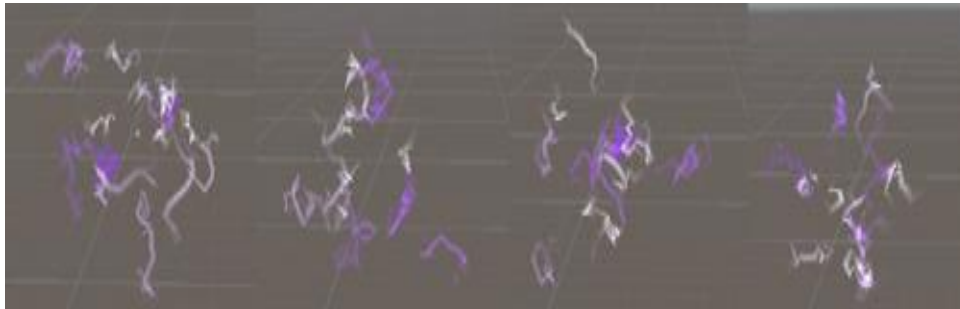


Fig. 1. Lightning skill effect using particle with noise module

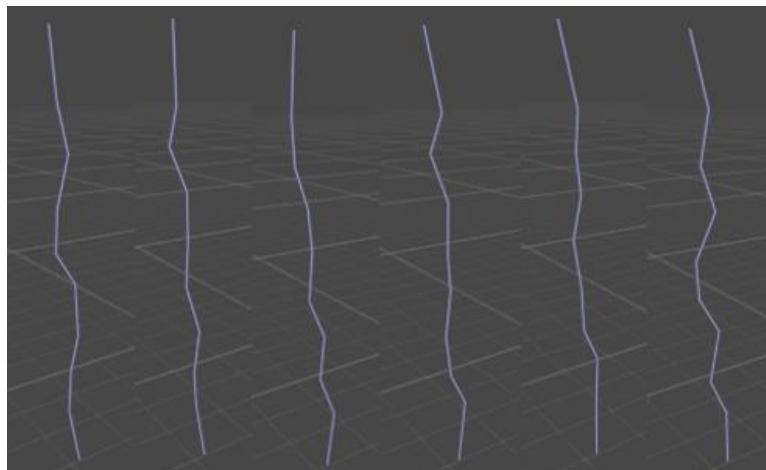


Fig. 2. Lightning skill effect using unity line renderer version 1

Table 1. Script for lightning effect using line renderer version 2

```

public class LineOne : MonoBehaviour {
    private LineRenderer lr;
    public int Segment;
    void Start () {
        lr = GetComponent<LineRenderer> ();
    }
    void Update () {
        lr.positionCount = Segment;
        for(int i = 0 ; i < Segment ; i++){
            lr.SetPosition (i, new Vector3 (0 + Random.Range(-.3f, .3f), i +
            Random.Range(.2f, .5f), 0));
        }
    }
}

```

Table 2. Script for lightning effect using line renderer version 2

```

public class LineTwo : MonoBehaviour {
    private LineRenderer lr;
    private int pos = 1;
    public int MaxSegment;

    void Start () {
        lr = GetComponent<LineRenderer> ();
        InvokeRepeating ("Strike", Random.Range (.5f, 1f), Random.Range (.01f, .1f));
    }
    void Strike () {
        lr.positionCount = pos;
        lr.SetPosition (pos - 1 , new Vector3 (pos-1 + (Random.Range (-.5f, .5f)),
        Random.Range(-.5f, .5f), Random.Range(-.5f, .5f)));
        pos += 1;
        if(pos > MaxSegment){
            pos = 1;
        }
    }
}

```



Fig. 3. Lightning skill effect using unity line renderer version 2

III. Result and Discussion

The lower value in CPU consumption and the higher value in FPS rate, means the lighter and faster the performance is. Using only simple configuration on particle system with lightning picture like what Zhang and Hu do is the lightest and fastest. But the result of this technique depends on the picture used. So, it only work for specific effects. On the other hand, noise module can create a realistic lightning effects while still light and fast as long as we know which parameters will rise the work on processing. What the line renderer do is very specific but highly customizable. Yet, it can be very heavy. But to be noted that, even without making it emits the same amount of particle like the other two, one line is enough to make lightning effects visually. So, it's performance can still be improved.

Table 3. Performance for each lightning effect technique

Lightning effect type	Average CPU Consumption	Average FPS Rate
Zhang and Hu	1.21 ms	827.82 fps
Noise Module	1.43 ms	787.53 fps
Line Renderer 1	1.83 ms	65.21 fps
Line Renderer 2	1.94 ms	318.35 fps

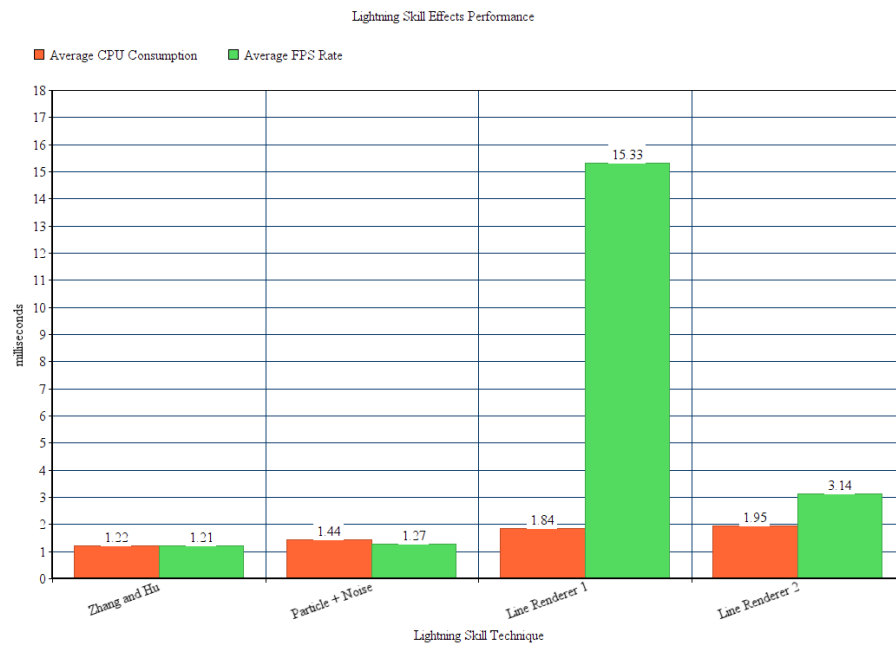


Fig. 4. Performance Comparison

IV. Conclusion

This articles is about four type of ways to create lightning skill effects. All of them are created using two objects in Unity 3d, that is particle system and line renderer. Each of them has different performance on CPU and FPS. It is up to the developer to use which technique that suit their needs and resources the most.

Acknowledgment

The author would like to say a lot of praise to Allah SWT who has provided ease for authors in completing this journal for the smooth task of seminar courses and Scientific Writing. Also thanks to friends who have supported the author in completing this research.

References

- [1] B. Zhang and W. Hu. "Game Special Effect Simulation Based on Particle System of Unity3D". IEEE Computer Society. vol 17. pp 595-598. 2017.
- [2] P. Kruszewski. "A probabilistic technique for the synthetic imagery of lightning". Computer & Graphics 1999. vol 23. pp 287-293. 1999.
- [3] S. Wang, Z. Mao, C. Zeng, H. Gong, S. Li. and B. Chen, 2010, June. A new method of virtual reality based on Unity3D. In *2010 18th international conference on Geoinformatics* (pp. 1-5). IEEE.
- [4] R.H Creighton., 2010. *Unity 3D game development by example: A Seat-of-your-pants manual for building fun, groovy little games quickly*. Packt Publishing Ltd.
- [5] K. Jones and R. Kettlewell. "The Unity particle system: features and tips". Unite Europe 2017. Amsterdam. 2017.