



Universität  
Bremen

University of Bremen

---

# Simulation of Dust Devils and Dust Storms on Mars with Unreal Engine

---

**Hanfang Ma**

Thesis submitted  
within the Bachelor's Program  
Digital Media

Supervisor: Prof. Dr. Gabriel Zachmann  
Second Examiner: Dr. Rene Weller

October 26, 2021

©2021 – HANFANG MA  
All rights reserved.

# **Simulation of Dust Devils and Dust Storms on Mars with Unreal Engine**

## **ABSTRACT**

My thesis focuses on improving the implementation of the phenomena such dust devil and dust storm on the Mars for project VaMEx-VTB, which is aimed to provide a simulation framework for testing the swarm behavior on the Mars terrain[1], using the new version of particle system 'Niagara' in Unreal Engine. The goal of the thesis is to provide a more realistic environment so that the swarm behavior on the Mars will be tested in a more accurate way.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Aim of Paper . . . . .	1
1.3	Structure of the paper . . . . .	2
<b>2</b>	<b>Previous Work and Preparation</b>	<b>3</b>
2.1	Relevance of the topic Natural Phenomenen . . . . .	3
2.1.1	Dust Devil . . . . .	3
2.1.2	Dust Storm . . . . .	6
2.2	Methodology . . . . .	8
2.2.1	Niagara Particle System . . . . .	8
2.2.2	PBR Material System . . . . .	11
2.3	Current state of research and work . . . . .	13
2.3.1	Mars' Whirling Dust Devil from NASA Jet Propulsion Laboratory	13
2.3.2	Large-Eddy Simulation of Dust Devils from Leibniz Universität Hannover (LUH) Institut für Meteorologie und Klimatologie . . .	15
2.3.3	The Martian (film) . . . . .	15
<b>3</b>	<b>Implementation</b>	<b>18</b>
3.1	Introduction of implementation . . . . .	18
3.2	The implementation of dust devil . . . . .	18
3.2.1	Implementation idea . . . . .	18
3.2.2	Implementation process . . . . .	19
3.2.3	Final effect . . . . .	26
3.3	The implementation of dust storm . . . . .	26
3.3.1	Implementation idea . . . . .	26
3.3.2	Implementation process . . . . .	27
3.4	Failure examples . . . . .	35
3.4.1	Failure Version 1. . . . .	35
3.4.2	Failure Version 2. . . . .	35
3.4.3	Failure Version 3. . . . .	36
3.4.4	Failure Version 4. . . . .	36
<b>4</b>	<b>Result</b>	<b>38</b>
4.1	Introduction of result . . . . .	38
4.2	Performance . . . . .	38
4.2.1	FPS without importing special effects . . . . .	38

## Contents

4.2.2	FPS when the particles of the main emitter of dust devil are 1000, 2000, 3000 respectively . . . . .	40
4.2.3	FPS when the particles of main emitter of dust storm are 50, 100, and 200 respectively . . . . .	40
4.3	Visual effects in VaMex . . . . .	40
4.4	Comparison with the old version . . . . .	43
4.4.1	Dust Devils . . . . .	44
4.4.2	Dust Storms . . . . .	44
4.4.3	Summary . . . . .	44
<b>5</b>	<b>Conclusion and Future Work</b>	<b>46</b>
5.1	Conclusion . . . . .	46
5.2	Future Work . . . . .	46
	<b>Bibliography</b>	<b>48</b>
	<b>List of Figures</b>	<b>49</b>
	<b>Acknowledgements</b>	<b>52</b>

# Chapter 1

## Introduction

### 1.1 Motivation

The thesis is a part of the project VaMex, which provides a virtual environment and framework to test the swarm behavior on the Mars terrain. The swarm includes several different elements, such as rovers, UAVs (Unmanned Aerial Vehicles (drones)) and humanoid robots (called as Charlie). The point of the project is to build a suitable and valid environment for autonomous swarm navigation and exploration system which communicate with each other, instead of physical tests, so that much expense and time can be saved.

My thesis focuses on the implementation of dust devil and dust storm, which are two common phenomena on the mars. Both of them can influence the visibility and movement of different project elements. My job is to create those two phenomena as close to reality as possible so that the test of swarm behavior can be more precise and accurate. Besides, my personal career direction is graphic engineer because I am interested in the computer graphic related knowledge such as rendering and animation in the field of animation, film and game industry. I see this bachelor thesis as a good start of my career path.

### 1.2 Aim of Paper

The aim of the thesis is to develop the phenomena mentioned before(dust devil and dust storm) with the game engine – Unreal Engine 4. The first step is to collect accessible information, especially video records and pictures of dust devil and dust storm, to study the simulated animation and research work of other institutions and universities and to see more media work like game and movies related to Martian environment. In this way, the morphological characteristics and movement characteristics of dust devil and dust storm can be summarized, and the difference between them on Mars and the earth is known.

The tool I use is Unreal Engine 4, and the function I use mainly is the latest version of the particle system - Niagara. The main visual effect will base on different emitters and emitter system. Blueprints will also be used when necessary(for example, when the texture of particles is modified). After the characteristics of dust devil and dust storm are derived, the visual effects will be implemented according to the Unreal Engine documentation and tutorials, and they will be tested in the VaMex-VTB project. Several

different versions of those two phenomena will be developed according to the previous research.

Finally, the implemented visual effects will be compared to the former version, and also compared to each other. A best version will be chosen considering different factors.

### **1.3 Structure of the paper**

In this thesis, my exposition will be presented to you in the following order:

I will elaborate on my preparations in Chapter 2 Previous Work and Preparation, which includes:

2.1 Relevance of the topic Natural Phenomenen: The main part is my research on the natural characteristics of dust devil and dust storm. 2.2 Methodology: Describes the software I use and the internal principles I will use.

2.3 Current state of research and work: Here I have listed three examples of research or special effects that are useful and inspiring for my implemntation.

Then, the most important chapter is the Implementation. I will briefly introduce my implementation ideas in chapter 3.1, then explain the implementation process of dust devil and dust storm in chapters 3.2 and 3.3 respectively, and finally show some of my failure cases in chapter 3.4.

In Chapter 4 Result, I will import the final effect into VaMex and explain how I achieved the effect from three different angles.

Finally, in Chapter 5 Conclusion and Future Work. I will make a brief summary of my work and estimate the possible directions for the implementation of special effects in the future.

Then let's get started.

# Chapter 2

## Previous Work and Preparation

### 2.1 Relevance of the topic Natural Phenomenen

Before the implementation, the characteristics of dust devil and sand storm should be observed. In the thesis, the most important observation objects are their morphological characteristics and movement laws. Other factors such as climate, magnetic field etc. will not be discussed in detail.

#### 2.1.1 Dust Devil

##### What is dust devil?

There are different definitions of dust devil in history. One of them briefly and accurately summarizes the appearance characteristics and causes of dust devil: Dust devils are small whirlwinds made visible by entrained dust and sand. They are upward moving, spiraling flows caused by heating of near-surface air by insolation.(Dust devils on Earth and Mars, Matt Balme, Ronald Greeley) [2]

Dust devils will form, when the hot air near the surface rises quickly through the cooler air above it. If certain conditions are met, possible rotation of the formed updraft will occur. This rotation will be intensified as more and more hot air rushes towards the bottom of the vortex. When the rising hot air and the sinking cold air reach a certain balance, the dust devil will start to move and pick up more dust.[3] According to the report, usually dust devil are harmless, but occasionally large-scale dust devil will also cause damage to people and property.[4]

##### Morphological characteristics and movement law of dust devils

1. General structure of dust devil:

According the article Dust devils on Earth and Mars(Matt Balme, Ronald Greeley)[2] , Sinclair, P. C.[6] spilt the vertical structure of a dust devil into three regions:

- The first region is the bottom part which is also called the surface interface region. This region is characterized with heavy particle load. And “vortex boundary layer” is in this region, where turbulent inflow occurs towards the center of the dust devil.
- The second region is the most important part for the implementation later, also the main part of the dust devil – the column of rotating dust. Its look is nearly





Figure 2.1: A dust devil in Arizona  
[5]

vertical. Unlike the first region, dust in this region has little exchange with the surrounding air.

- The third region is the top part. In this region the particles will dissipate into the air and the rotation of the dust devil will decline.

2. Other details:

- Dust devils are at least 5 times higher than they are wide. [Hess and Spillane, 1990]
- Dust devils often tilt toward the direction of motion by about  $10^\circ$ [McGinnigle, 1966; Sinclair, 1973; Mattsson et al., 1993]
- Dust Devils can sometimes be crooked or sinuous because of wind shear. [Matt Balme, Ronald Greeley, 2006]

**Dust devil on the mars**

There are already some images and videos about the dust devils on Mars. Martian dust devils and Earth dust devils are very similar in appearance and characteristics – most of them are both rotating cylinder of wind and sand – but there are still some differences. One of the most notable features is the difference in height. The largest dust devil on Mars can reach a height of 8 kilometers[7] , which is many times higher than that on Earth. The following is a comparison picture[7], you can see the size of the Martian dust devil very intuitively.

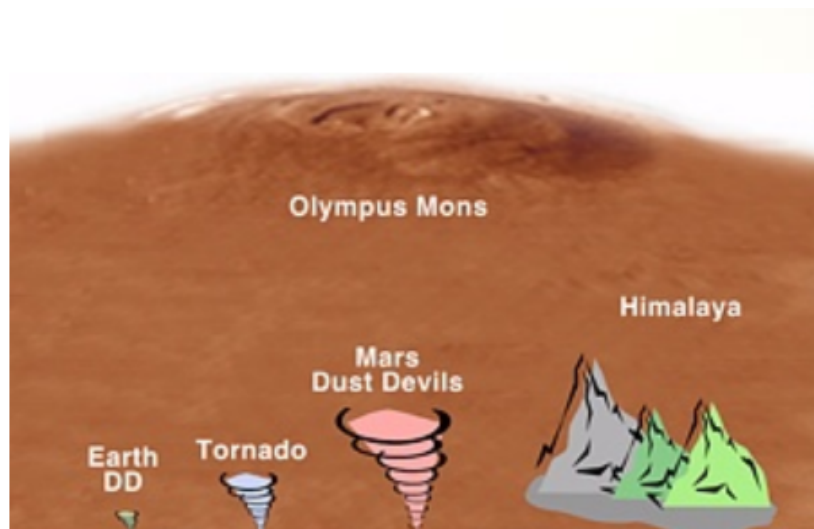


Figure 2.2: Diagram courtesy of Malin Space Science Systems [7]



Figure 2.3: Serpent Dust Devil of Mars  
[5]

### 2.1.2 Dust Storm

#### What is dust storm?

Dust storms are also a common natural phenomenon on Mars and on Earth. Intuitively speaking, a dust storm is a very strong wind, and this gust of wind carries many small particles, usually sand, silt and other materials. These small particles will spin around in the dust storm. Dust storms on the earth tend to spread hundreds of miles and rise more than 10,000 feet with wind speeds of at least 25 miles per hour.[8]



Figure 2.4: A dust storm approaches downtown Phoenix on August 11, 2012. Credit: NOAA[9]

### Morphological characteristics and movement law of dust storms

Visually, dust storms look like moving “wall” composed of dust and debris, which are moved by strong winds caused by thunderstorms. Like the figures showing below.[9] We can also observe from these videos and pictures that sandstorms are often very huge, like a wall that is slowly moving. They will make the surrounding environment visibility becomes very low, with a strong sense of oppression.



Figure 2.5: A sandstorm approaching Al Asad April 27, 2005.

### Dust storms on the mars

Humans have always had many speculations about the environment on Mars. With the exploration of Mars in recent years, people have learned that the dust storms on Mars are very strong. Sometimes the dust storms on Mars are strong enough to be seen by telescopes on Earth.

”Every year there are some moderately big dust storms that pop up on Mars and they cover continent-sized areas and last for weeks at a time,” said Michael Smith, a planetary scientist at NASA’s Goddard Space Flight Center in Greenbelt, Maryland.

However, the dust storm on Mars does not have the ability to trap astronauts like in the movie ”The Martian”. They are usually unable to destroy scientific research equipment, because the strongest Martian dust storm can reach wind speeds of up to 60 miles per hour, which is less than half the intensity of some hurricanes on Earth. But they may still cause some damage to the equipment or reduce the production of solar energy sources.[10]

Other details:

According to an interview from experts[11], we can know that dust storms on the Mars:

- can reach a height of 20 to 30 kilometers,
- can reach a global level, allowing the surface of Mars to be completely covered,
- will cause solar equipment on Mars unusable,
- (in these huge sandstorms) there will be lightning,
- Moving speed is fast (shown in the video material).

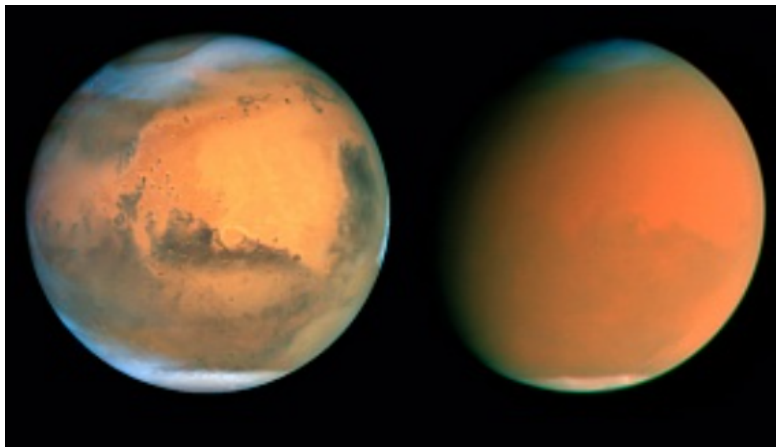


Figure 2.6: The Hubble observed the 2001 dust storm from Earth orbit. [12]

## 2.2 Methodology

### 2.2.1 Niagara Particle System

#### 1. What is Particle System?

In my implementation, the most important tool is undoubtedly the particle system. This is my most basic tool for simulating natural phenomena.

Firstly I will introduce the definition of particle system. Particle systems are often used in games and movies to create natural phenomena and visual effects. They often use any minute sprites, 3D models, or other graphic objects to simulate certain kinds of "fuzzy" phenomena. These special visual effects are often difficult to achieve with other rendering techniques. [13]

In a typical implementation to produce special effect with the help of particle system, there are often two stages: the simulation stage and the rendering stage.

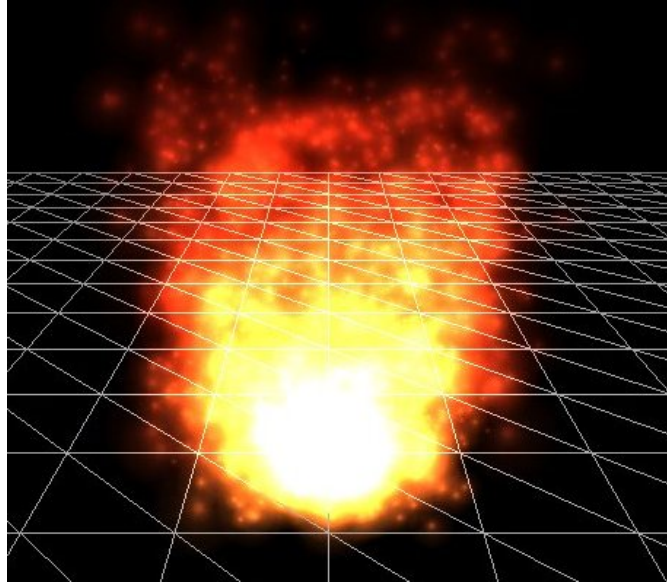


Figure 2.7: A particle system used to simulate a fire, created in 3dengfx.  
[13]

In the simulation phase, the particle system calculates the number of newly created particles and the location of the particles based on the spawning rates and update interval. The parameters of each particle are initialized according to the parameters of the emitter. Particles at the end of their life cycle will be deleted, and the positions and properties of particles that are still alive will be modified accordingly. Usually the collision between particles and particles will be ignored, but particles and objects in the scene will perform collision detection.

Every time the simulation phase is over, each particle will be rendered, usually in the form of a textured billboarded quad (i.e. a quadrilateral that is always facing the viewer).[13]

## 2. What is Niagara VFX System? What is the difference between Cascade and Niagara?

Niagara is a next-generation VFX system developed by Unreal Engine. Compared with the old version of Cascade, although both of them can be used to create dazzling visual effects, Niagara is more powerful in function, and the way of creating and adjusting VFX is very different from Cascade. With Niagara, technical artists can create additional features independently without the help of programmers.[14]

Unreal Engine has enhanced the ability of Niagara particles system to access the bottom layer of rendering, making Niagara a shader program independent of the material system, and many functions can be completed on the front end. But this also means that the functions in the Niagara system are enormous and complex. I only used part

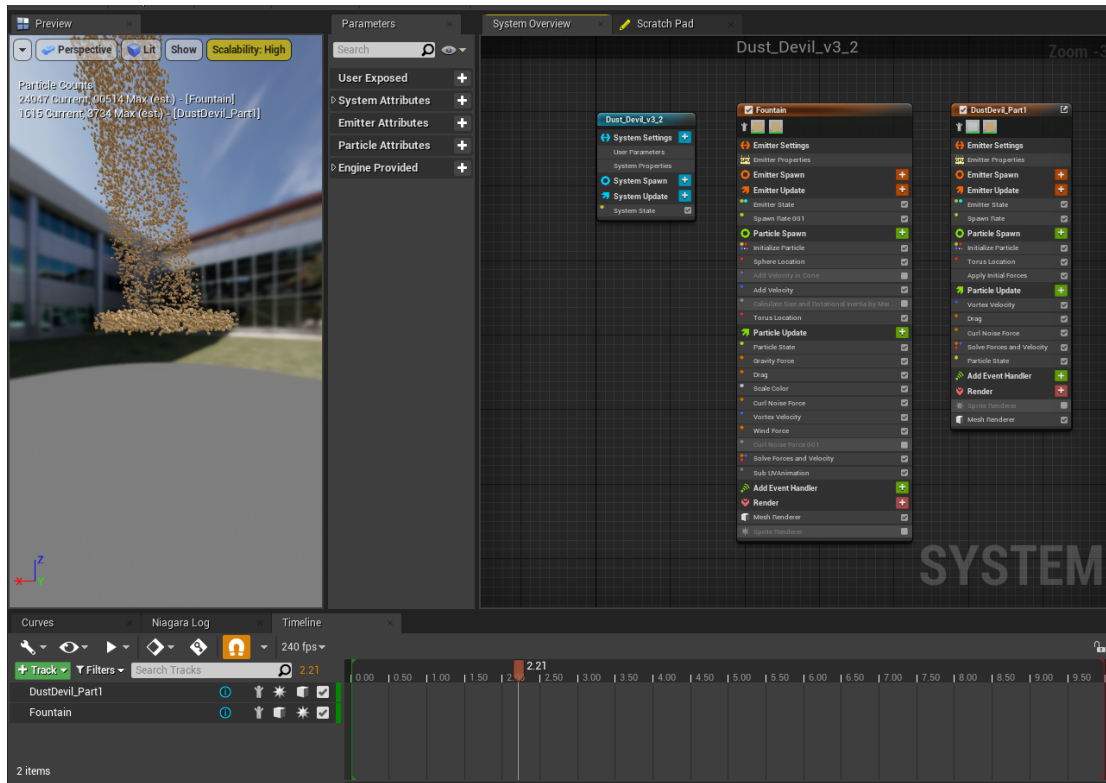


Figure 2.8: An overview of Niagara UI

of the functions in the implementation, so I will focus on the functions I use.<sup>[14]</sup> There are four core components in the Niagara VFX system:

- Systems
- Emitters
- Modules
- Parameters

In my implementation, I mainly used systems and emitters. In general, a system contains multiple emitters. In the system editor, we can modify the parameters of the emitters, or add and delete emitters. A single emitter contains modules, and they are for single purpose, but can be reused. A special feature of the Niagara emitters is that we can create a simulation using the module stack, and then render the simulation in multiple ways. <sup>[14]</sup>

Basically only two renderers are used in my implementation, which are Sprite Renderer and Mesh Renderer. Sprite Renderer is the default Renderer in Niagara. Its characteristic is that every particle is just a flat. The advantage is that the calculation time can be greatly shortened. The disadvantage is that the particles lack a sense of volume,

which can make some certain effects look distorted. Mesh Renderer, as its name, makes the emitted particles present a specific mesh, which will make the special effects more realistic, but at the same time, because more geometric shapes need to be rendered, the rendering speed will be slower. In my implementation, Sprite Renderer is mainly used. For some parts that need to show volume, I use Mesh Renderer.[15]



Figure 2.9: Different Renderers [15]

## 2.2.2 PBR Material System

### 1. What is PBR?

Like many game engines or modeling software nowadays, UE4 uses the PBR material system. The full name of PBR is physically based rendering. This is a method in the field of computer graphics, the purpose is to simulate the flow of light in the real world to render images. A major feature of the PBR method is photorealism.[16]

Physically-based materials will work well in all lighting environments, and the materials do not need to be very complicated and interdependent, so that the interface is more intuitive. [17]

For these reasons, many next-generation games or movies currently use the PBR method.

### 2. What are the basic parameters in PBR Material System?

There are many parameters in PBR material. Four of these parameters are particularly important in use, they are:[17]



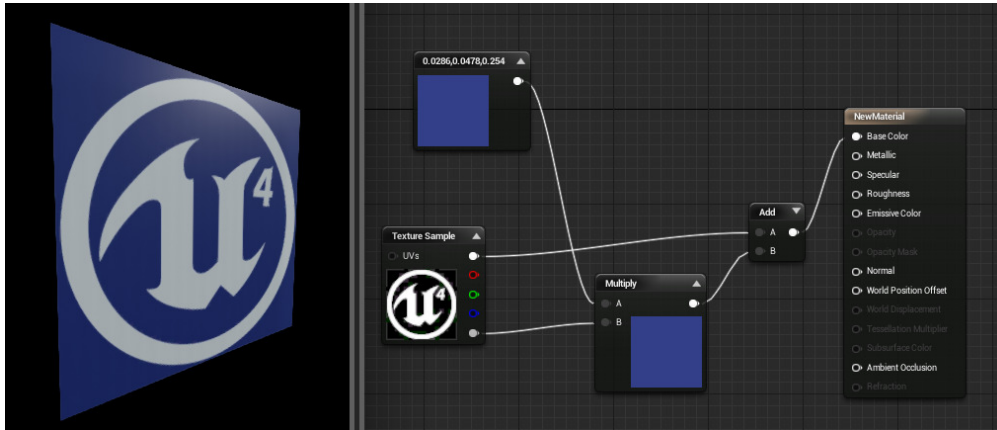


Figure 2.10: The interface of PBR system in UE4 [17]

- **Base Color:** Determines the overall color of the material, using a Vector3 (RGB) value, where the value of each channel is between 0 and 1.
- **Roughness:** Determines the roughness of the material. Rough materials scatter in more directions than smooth materials. The value is between 0 and 1. A roughness of 0 is a specular reflection, and a roughness of 1 is a diffuse reflection surface.

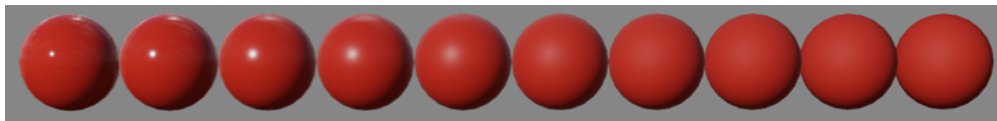


Figure 2.11: Material with different roughness [17]

- **Metallic:** Determines how "metallic" the material is, with values ranging from 0 to 1. For pure surfaces (pure metal, pure stone, pure plastic, etc.) the value is 0 or 1. If the material is mixed, such as rusty metal, the value is between 0 and 1.

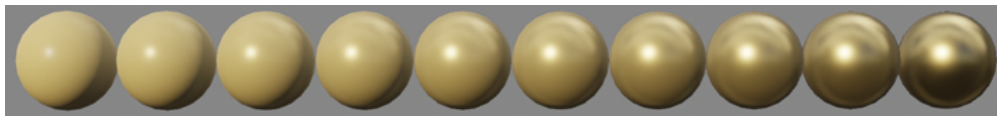


Figure 2.12: Material with different metallic. [17]

- **Specular:** When editing non-metallic surface materials, you need to adjust the value of specular reflection. The default specular value of the material is 0.5.

## 2.3 Current state of research and work

Today, there are already many research institutions or movies/games trying to simulate the dust devils and dust storm on Earth and on Mars. Some of these simulations are very close to real phenomenon. Here I have selected some works as a reference for me to use UE4 to simulate the dust devils and dust storms on Mars.

### 2.3.1 Mars' Whirling Dust Devil from NASA Jet Propulsion Laboratory

A skinny 'dust devil' on the dust-covered Amazonis Planitia region of northern Mars is recorded and NASA has animated a dust devil based on it, to visually show what such a dust devil looks like in the Martian atmosphere. The image was captured by the High Resolution Imaging Science Experiment (HiRISE) camera on NASA's Mars Reconnaissance Orbiter on March 14, 2012[18]

In this animation, we can see the morphological characteristics of the dust devils on Mars: the height is much higher than the dust devil on the earth, and the shape is slender, showing a certain degree of curvature. At the bottom there is a vortex wider than the cylinder.

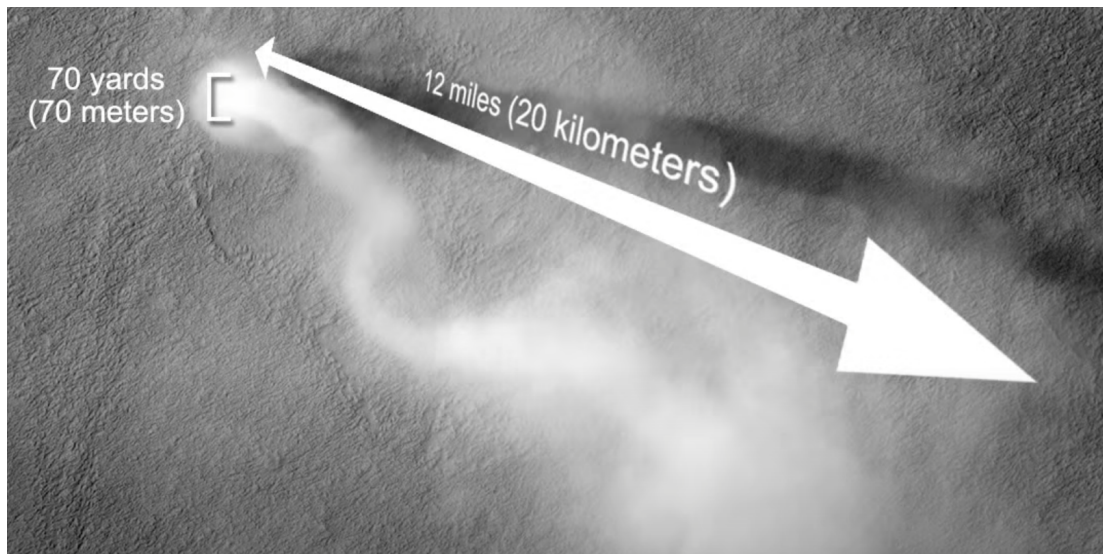


Figure 2.13: Original photo by HiRISE camera on NASA's Mars Reconnaissance Orbiter. [18]



Figure 2.14: The animation by NASA.

[18]

### 2.3.2 Large-Eddy Simulation of Dust Devils from Leibniz Universität Hannover (LUH) Institut für Meteorologie und Klimatologie

This animation is from Institut für Meteorologie und Klimatologie, Leibniz Universität Hannover (LUH). The researchers showed in this video that in a time period of 10 minutes, how a large dust devil in the convective boundary layer develops. The dust devil here is visualized by virtual dust on the surface of the model. The rendering process is completed by VAPOR.[19]

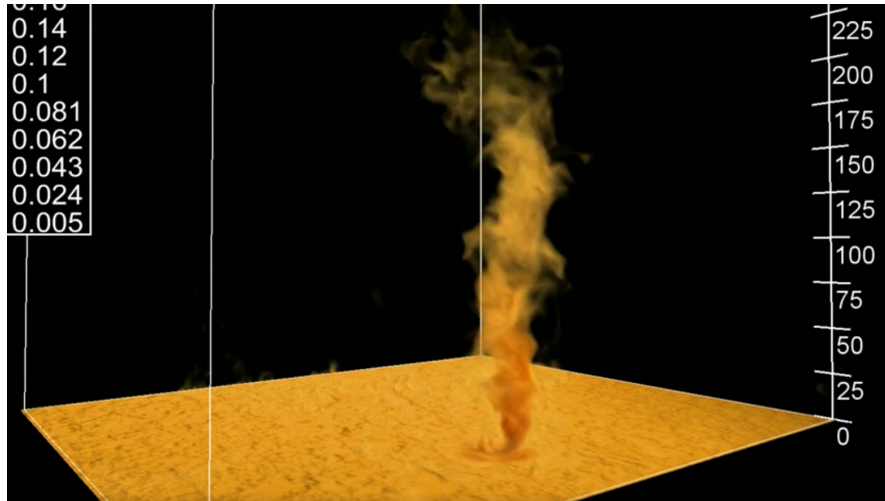


Figure 2.15: Large-eddy simulation of dust devils [19]

In this animation, we can see more clearly how the inside of the dust devil is formed and how it moves. The simulated dust devil in the video presents the characteristics mentioned in the previous investigation: the particles at the bottom are the densest, the concentration of particles gradually decreases from bottom to top, and the particles at the top dissipate.

### 2.3.3 The Martian (film)

In recent years, a film about Mars has attracted much attention - The Martian. This film depicts the lonely life of an astronaut on Mars and how NASA tried to rescue him. This movie is not only a fascinating plot, but also excellent in the visual effects depicting the Martian scene. In order to promote the potential of space exploration, NASA assisted in the production of this film. They provided hundreds of real images of Mars to the production team, which also made the Mars scenes in the film particularly realistic.[20] So the dust devils and sand storms in this movie have also become a good reference when I implement visual effects.



Figure 2.16: VFX Breakdown of dust devils in The Martian  
[21]

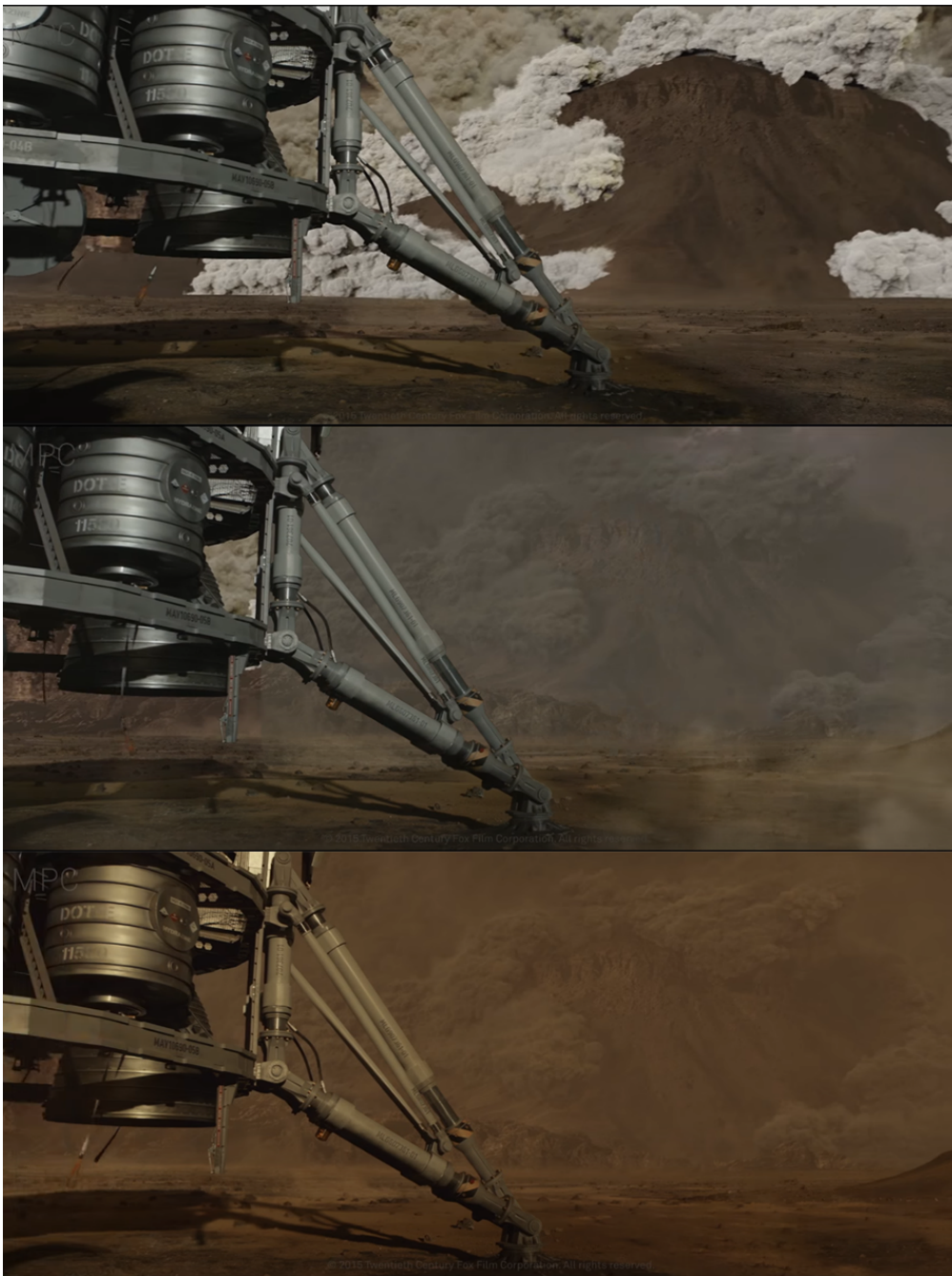


Figure 2.17: VFX Breakdown of dust storms in The Martian  
[21]

# Chapter 3

## Implementation

### 3.1 Introduction of implementation

In this chapter, I will follow the steps to introduce in detail how I implemented dust devil and dust storm. The whole process is completed in UE4.

The introduction of each phenomenon will be divided into these three parts:

- Implementation idea: According to the research and reference mentioned above, which parts of each special effect are composed.
- Implementation process: What parameters are in each emitter and which parameters are important.
- Optimization and comparison: After general steps, I will adjust the parameters and compare different versions to choose the best version.

### 3.2 The implementation of dust devil

#### 3.2.1 Implementation idea

Here we will review the important morphological characteristics of dust devil:

- Dust devils consists of three parts: the vortex at the bottom, the cylinder and the dissipated dust at the top.
- Dust devils are at least 5 times higher than they are wide.
- Dust devils will have a  $10^\circ$  tilt.
- Dust devils can sometimes be crooked or sinuous because of wind shear.
- The dust devil on Mars can be as high as 8 kilometers.

Based on the above characteristics, I preliminarily determined that the dust devil will consist of at least two emitters: one is the vortex at the bottom and the other is the column of the main body. We can achieve the effect of dissipating top particles by adjusting the death period of the particles. Other characteristics need to be manifested with different forces and different parameters, which I will discuss in detail in the next

part. In addition, if you observe the video materials of dust devils, you can easily find that, because dust devil is dust that is rolled up by the wind, there will often be some foggy sand dust around it, as shown in the figure. So there will be a third emitter to present this foggy sand.



Figure 3.1: There will be foggy sand around the dust devil. - from Youtube channel National Geographic "Martian Dust Devils — National Geographic" [22]

### 3.2.2 Implementation process

#### Dust devil

##### 1. First Emitter: The vortex at the bottom

**Initialization:** First we need to initialize the number of particles. From the previous images and researches, we can see that the density of the bottom dust is relatively high, so we set the Spawn Rate to 1000 in Emitter Update.

**Emission method:** The dust at the bottom is swirling, so we definitely need to change the default emission method of particles: add Torus Location to Initialize Particle to make the particles emit in torus.

**Update:** In addition, in order to increase the rotation speed of the vortex, we need to add another speed in Particle Update: Vortex Velocity, which can make the rotation speed of the bottom vortex look more natural.

**Others:** To prevent the vortex from diverging too much, Drag must be added. Make the particles more natural: add Curl Noise.



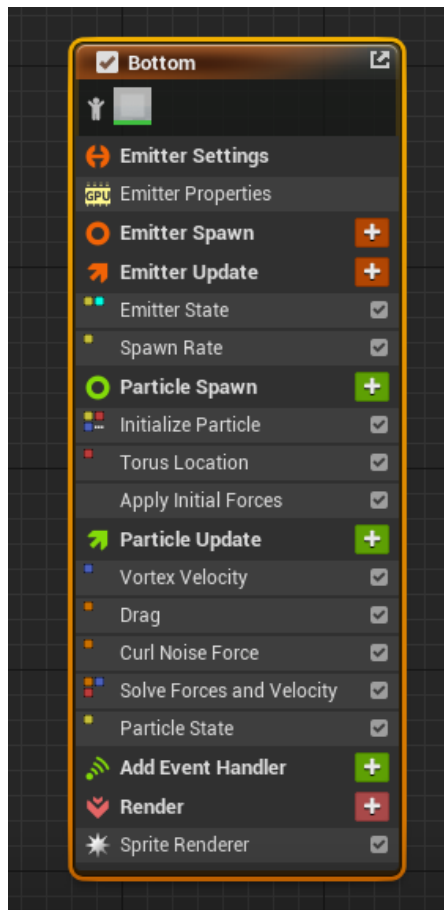


Figure 3.2: The parameters of the first emitter.

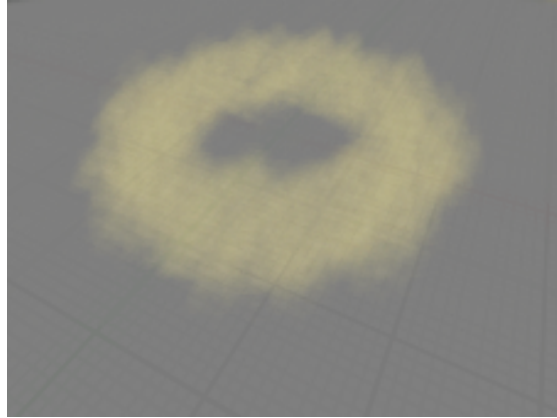


Figure 3.3: Visual effect of the first emitter.

## 2. Second Emitter: The main cylinder

**Initialization:** Like the first emitter, the number of particles in this emitter is also very dense, so I set the Spawn Rate to 3000. But the difference is that the top particles should be more sparse, so the lifetime of the particles cannot be the same. So I set the lifetime mode of particles as random, and there must be a certain gap between the maximum and minimum values (the larger the gap is, the sparser are the top particles, because the top particles are those with a longer lifetime. When the number of all particles is constant, larger gap between the maximum and the minimum means that the number of particles with a longer lifetime will decrease.) Here I set the minimum lifetime to 3 and the maximum lifetime to 6, making the effect of dissipating look closer to the real dust devil.

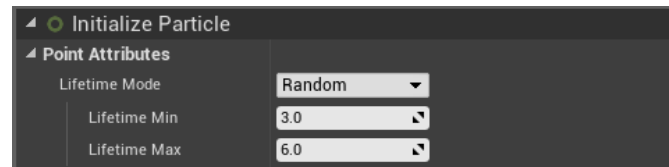


Figure 3.4: Initialize Particle

**Emission method:** Like the first emitter, the main cylinder's emission method cannot be the default. It also needs to be emitted in a vortex (Torus Location) with a radius smaller than the radius of the bottom vortex. In addition, I also added a spherical emission method (Sphere Location), which can make the bottom of the cylinder look more volumetric and more realistic. In the initial phase of the particle (Particle Spawn) there is a very important value: Add Velocity, this velocity allows particles to emit at an initial velocity. The shape of the cylinder needs to be maintained, so I only added speed in the direction of the Z axis. In order to make the effect more natural, Random

Range Vector is also added to the speed, so that the value of the speed is set to random.

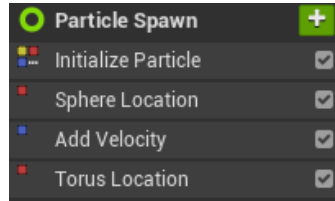


Figure 3.5: The parameters of Particle Spawn.

**Update:** In the Particle Update section, I added a lot of values to achieve the characteristics of dust devil. Vortex Velocity allows the emitter to present a three-dimensional cylindrical effect, which is very important. Wind Force and Curl Noise Force can make the cylinder have a certain degree of tilt and swing changes. Other values such as Drag and Gravity Force added some details and minor changes.

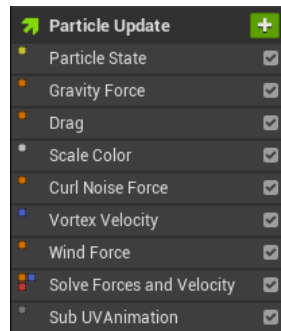


Figure 3.6: Parameters of the Particle Update.

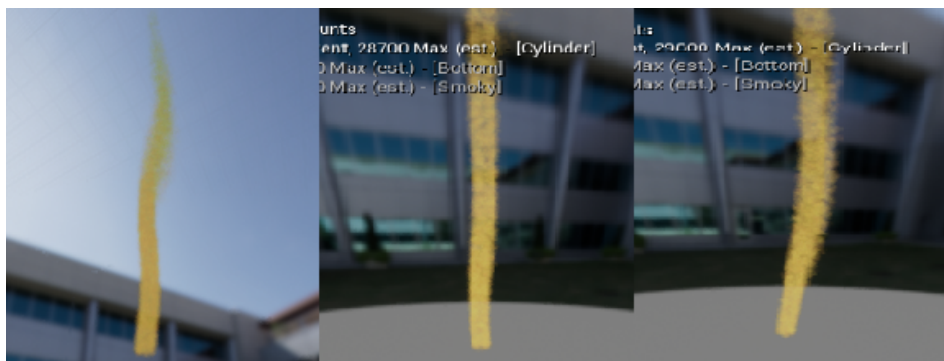


Figure 3.7: Visual effect of the second emitter.

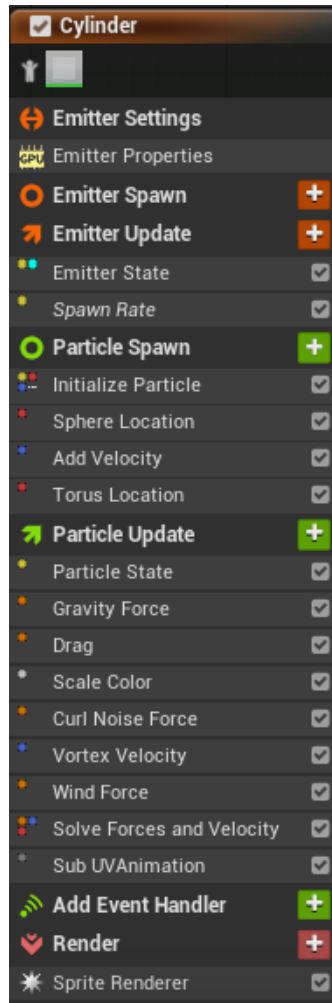


Figure 3.8: Parameters of the second emitter.

### 3. Third Emitter: Smoky sand

**Initialization:** The number of particles in the third emitter will not be that many, so I set the Spawn Rate to 40. However, because of the fog-like effect, the size of the particles needs to be adjusted to a relatively large value (if the particles are too small, the effect will be too grainy and distorted) For a natural effect, the size of the particles is also randomly distributed. The minimum value is 375 and the maximum value is 450, which are much larger than the other two emitters.

**Emission method:** Because the fog-like dust is not emitted in a concentrated manner, but looks relatively scattered (they are mainly emitted from the cylindrical body to the surrounding air), so I added the Cylinder Location for the emission method. Similarly, they also need to be initialized speed, also used Add Velocity(only in z axis). However, due to the influence of air friction and other forces, their speed will appear to be slower

than that of the main cylinder.

**Update:** The parameter type of Particle Update of the third emitter is basically the same as that of the second emitter. There will be some small modification according to the actual effect.

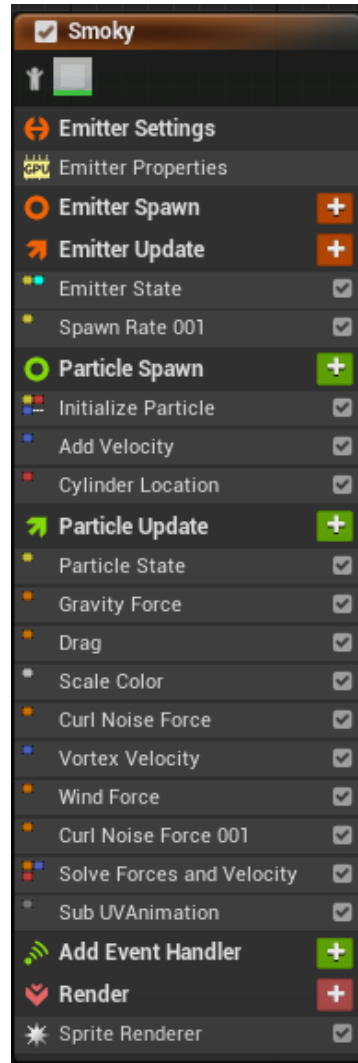


Figure 3.9: Parameters of the third emitter.

#### 4. Material

When implementing the dust devil effect, I only used Sprite Renderer, because there is no obvious volume effect in dust devil. Also because the number of particles is large, using Sprite Renderer can make the calculating speed faster.

After considering different materials, I finally found that using the Material M\_smoke\_subUV



Figure 3.10: Visual effect of the third emitter.

from UE4 has a good effect. Three of the emitters all use M\_smoke\_subUV, and the colors are ordinary sand colors, but the specific color adjustment needs to be adjusted in VaMex, which I will mention in the later chapter. I also adjusted the Sub Image Size in Sub UV to X=8, Y=8, and added Sub UVAnimation to the Particle Update in the second and third emitters to make the dynamic effect better.

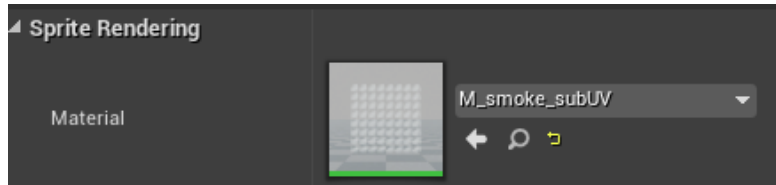


Figure 3.11: M\_smoke\_subUV Material

### 3.2.3 Final effect

Combining the three emitters and the material, I put the niagara system into Level to see the effect. Figure is what it looks like in the Level.



Figure 3.12: Visual effect of the whole niagara system

## 3.3 The implementation of dust storm

### 3.3.1 Implementation idea

As before, let's review the morphological characteristics of dust storm:

- The most critical morphological feature: the appearance of a dust storm is like a moving “wall”, and the dust storm on Mars moves very quickly.

- Dust storms are full of sand, which look like smoke and fog.
- There are small particles flying in the dust storm.
- There are also relatively large objects scattered in the dust storm, such as rocks and mud.
- Sometimes in dust storm is accompanied by lightning.

So in dust storm, I created a total of **6 emitters**:

- The first is the moving “wall”, which is also the core of this system.
- The second is the misty sand rising around the “wall”, which is very similar to the third emitter “smoky sand” of dust devil.
- The third and fourth are both lightnings.
- The fifth and sixth are the micro-particles and rocks suspended in the dust storm.

### 3.3.2 Implementation process

#### 1. First Emitter: “Moving Wall”

**Initialization:** Although Sand Wall looks very large, the number of emitted particles is not that many as expected. Because fog-like effect will make the size of particles larger, it would look too crowded when there are too many particles. So I set the number of initial particles to 100, which is much smaller than the number of dust devil.

**Emission method:** I used Sphere Location and Box Location to emit Sand Wall. Radius is also relatively large (the materials show that the dust storm on Mars is huge). What’s more, according to the videos and articles I mentioned in the previous chapter, the dust storm on Mars moves very fast, so I added Add Velocity and adjusted the value according to the moving speed of dust storm on the video.

**Update:** In Particle Update, I added Acceleration Force, so that dust storm can move faster and faster over time, which looks closer to the real situation.

#### 2. Second Emitter: Smoky sand

The basic parameters of smoky sand and sand wall are similar.

In Particle Update, I added Curl Noise Force to make it look more natural. The transparency and brightness of the particles are adjusted to be very small, so that it looks transparent, because the visibility of the smoky sand is greater than that of the sand wall.



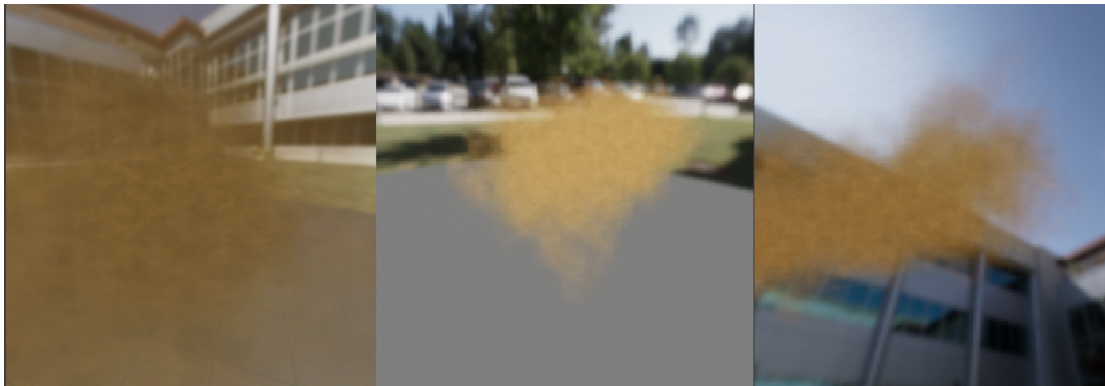


Figure 3.13: Visual effect of the first emitter.

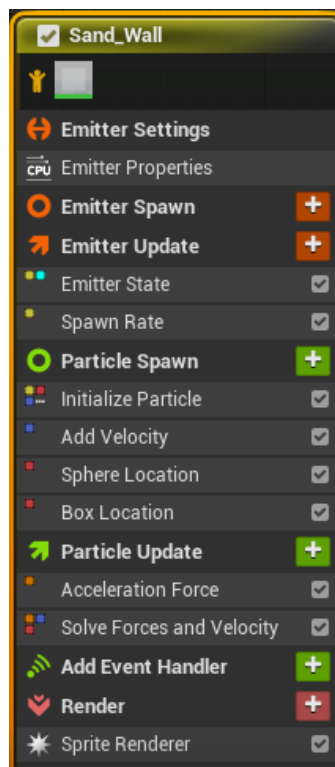


Figure 3.14: Parameters of the first emitter.



Figure 3.15: Visual effect of the second emitter.

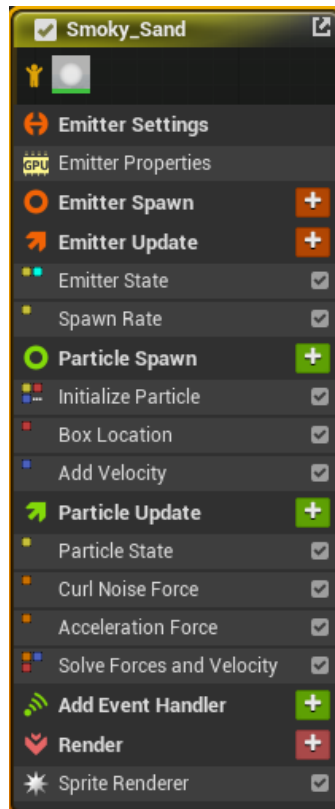


Figure 3.16: Parameters of the second emitter.

### 3. Third and Fourth Emitters: Lightning

**Initialization:** First of all, we must explain that the Renderer used by Lightning is different from the one we used earlier: Ribbon Renderer. This Renderer is generally used to render objects like silk and satin, and it is suitable here to show the effect of lightning. And because the default effect in Ribbon Renderer will let the object look too smooth, we can turn off the Mode option of Tessellation in Ribbon Renderer to make the object has a more hard appearance.

Lightning is generated in an instant, so I added Spawn Burst Instantaneous in Emitter

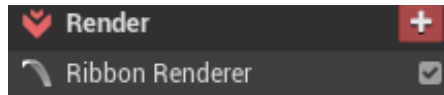


Figure 3.17: Ribbon Renderer

Update, and the time cannot be too long.

**Emission method:** The method of emission is also different from the previous method. In Emitter Update I added Beam Emitter Setup. Here we can turn on Use Beam Tangents to adjust the shape of the lightning until it looks better.

**Update:** We all know that in nature, lightning flashing from light to dim happens in a short period of time. Therefore, in our emitter, the interval between lightning emission must also be relatively short.

We need to change the Life Cycle Mode in Emitter State to Self, so that the life cycle of particles does not need to follow the entire system. Then we adjust the life cycle of the particle to a relatively small value. At the same time, we also need to change the number of loops in Emitter Update, and adjust it repeatedly until the effect looks more credible. After those steps, the flashing effect of lightning appears.

What's more: In order to make the lightning have some jitter effect: I added Jitter

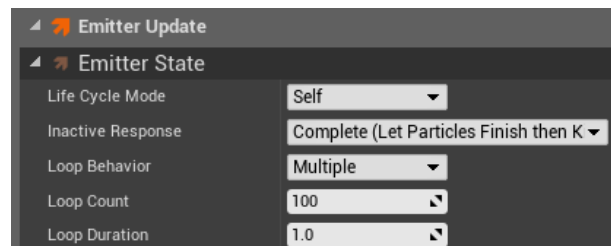


Figure 3.18: Emitter Update

Position in Particle Update. Similarly, Curl Noise Force is also to make the effect look more random.

Note: The creation of Lightning refers to many suggestions in this Tutorial: UE4 Niagara Particle System Quick Introduction.[\[23\]](#)

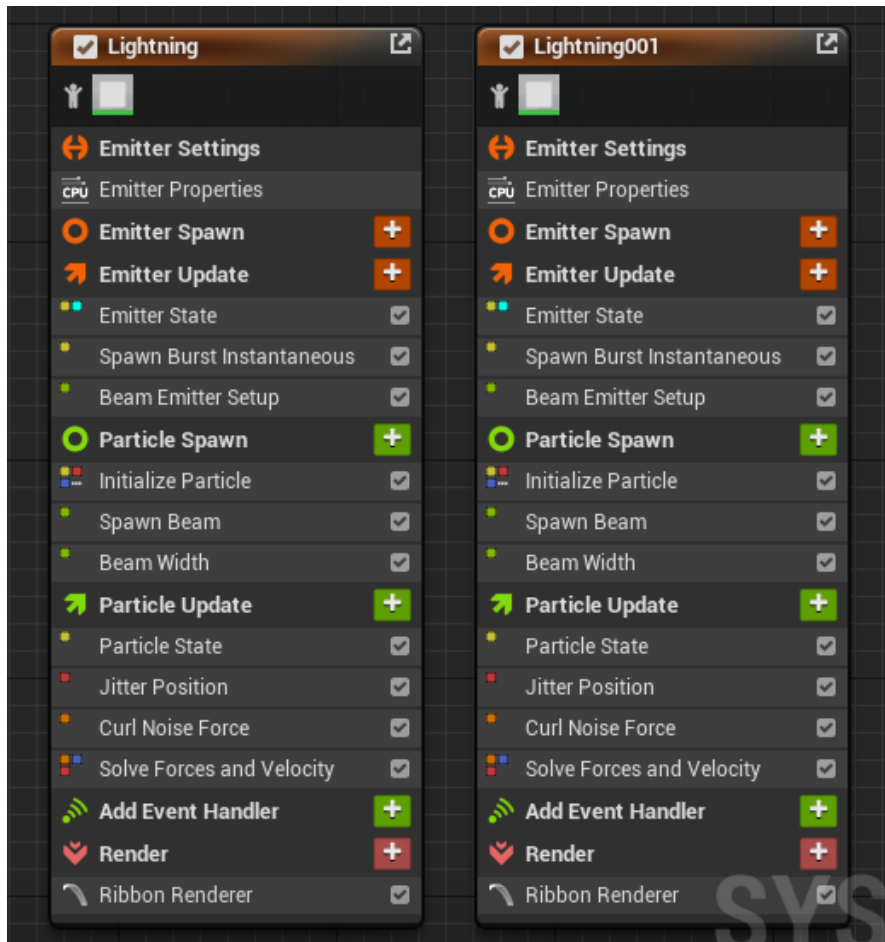


Figure 3.19: Parameters of the lightning emitter.



Figure 3.20: Visual effect of the lightning emitter.

#### 4. Fifth and Sixth Emitters: Micro-particles and rocks

**Initialization:**Micro-particles and rocks are to increase the details of the overall special effect. The overall trajectory of micro-particles follows the Sand Wall. The biggest difference is that the size of the particles is much smaller and the number is larger(The number of micro-particles here is 3000).

And the rock is more volumetric, so here I use Mesh Renderer. Among the Particle Mesh option of Mesh Renderer, I chose MaterialSphere, which is closer to the actual rock. The initial number of rocks will also be smaller, because too many particles that need to be rendered in the Mesh Renderer will cause the running speed to be very slow, here I set it to 50.

**Emission Method:**The emission method of both emitters is Box Location, and there are initial speed: Add Velocity and initial Force: Apply Initial Forces.

**Update:**In the Particle Update, Rock has added Curl Noise Force and Wind Force to make the trajectory of the rocks more random, and the effect is better.

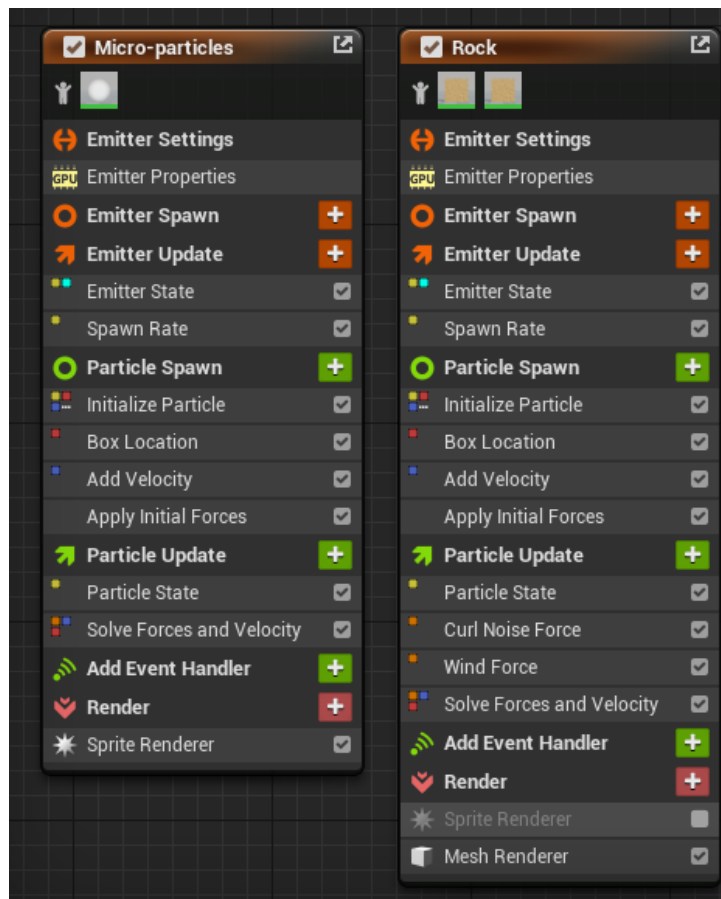


Figure 3.21: Parameters of the micro-particles and rock emitter.



Figure 3.22: Visual effect of the micro-particles and rock emitter. (With Sand Wall in the background)

#### 4. Material

Sand Wall and Smoky Sand still use the M\_smoke\_subUV Material from UE4. Micro Particle uses the UE4 default Material of Sprite Renderer: DefaultSpriteMaterial. For the lightning texture I created a simple material: M\_Lightning. The Blend Mode needs to be set to Translucent, and the Shading Mode is set to Unit. The blueprint is shown in the figure.



Figure 3.23: Overview of M\_Lightning.

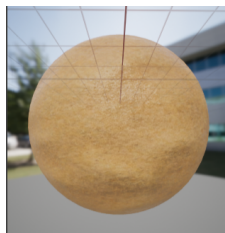


Figure 3.24: Overview of M\_Sand.

In order to show the volume of Rocks, I created a new material M\_Sand and found free PBR textures on the Internet.[\[24\]](#)The blueprint and effect are shown in the figure.

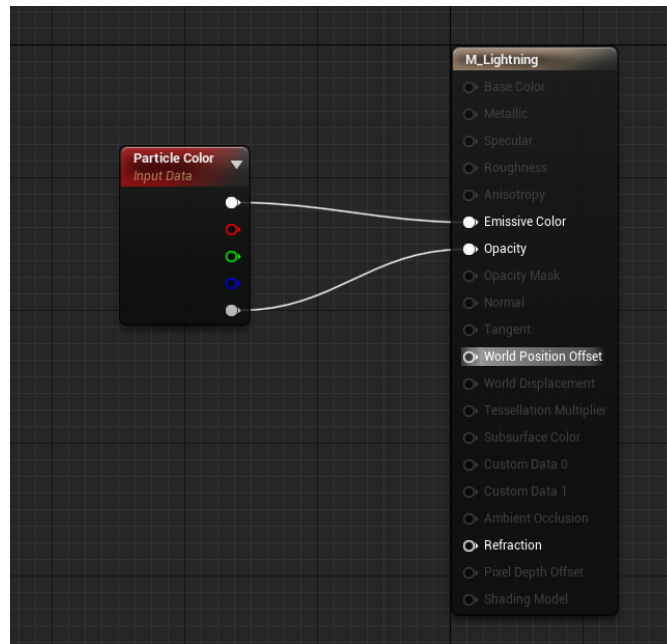


Figure 3.25: Blueprint of M\_Lightning.

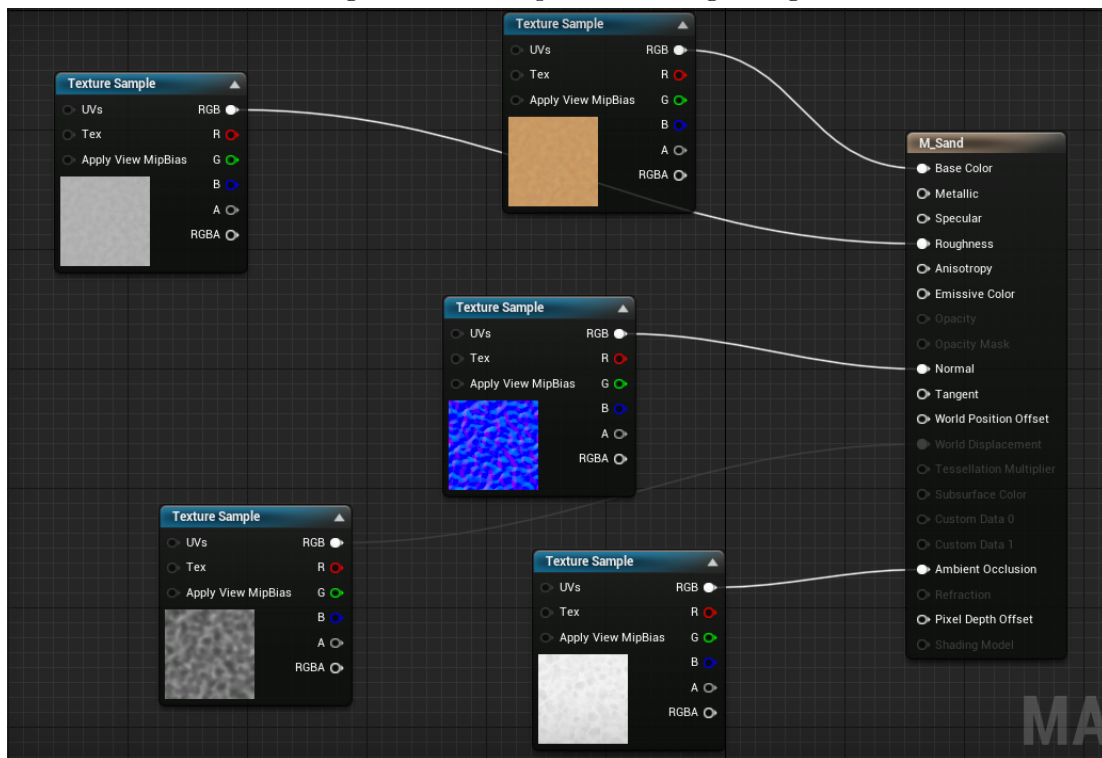


Figure 3.26: Blueprint of M\_Sand.

### Final effect

After the steps above, we achieved the effect as follow.

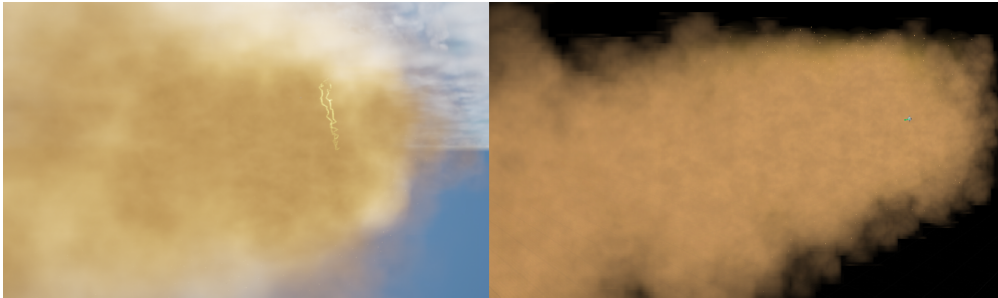


Figure 3.27: Visual effect of dust storm.

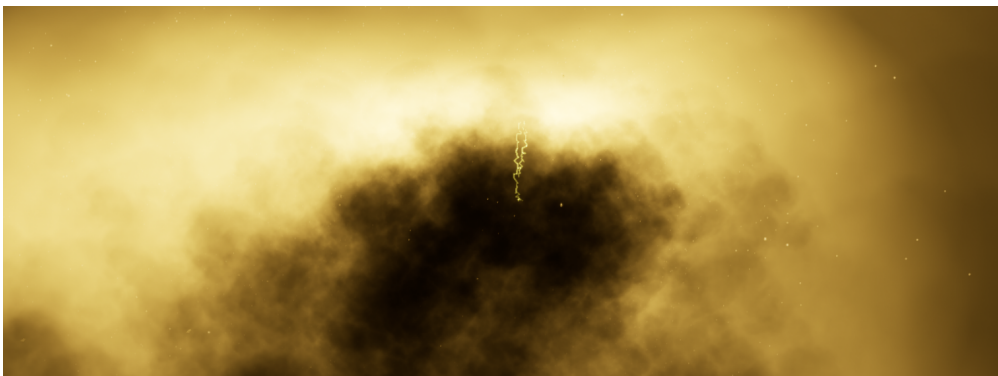


Figure 3.28: Visual effect of dust storm.

## 3.4 Failure examples

Here I will show some failure cases(dust devils) in the process. Although those results are not very good, they have given me a deeper understanding of the Niagara System through trial and error, and making the final effect be closer to reality.

### 3.4.1 Failure Version 1.

The velocity of this case is somehow incorrect. Too much velocity in x and y axis, which makes the effect far from reality.(Figure 3.29)

### 3.4.2 Failure Version 2.

The visual effect of this version is good but the real dust devils look more like smoke. You can see clearly there are many particles within it, which is not the case in reality.(Figure 3.30)



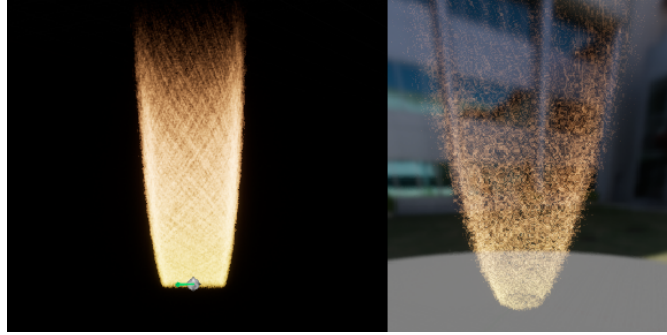


Figure 3.29: Failure version one

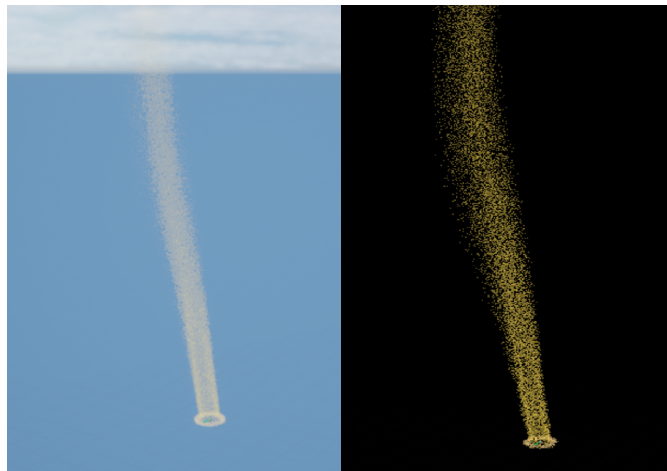


Figure 3.30: Failure version two

### 3.4.3 Failure Version 3.

The general look of this version is fine, but it lacks dynamic. The moving direction is too stiff.(Figure 3.31)

### 3.4.4 Failure Version 4.

The effect is closest to the final one. But it lacks the smoky sand around it.(Figure 3.32)

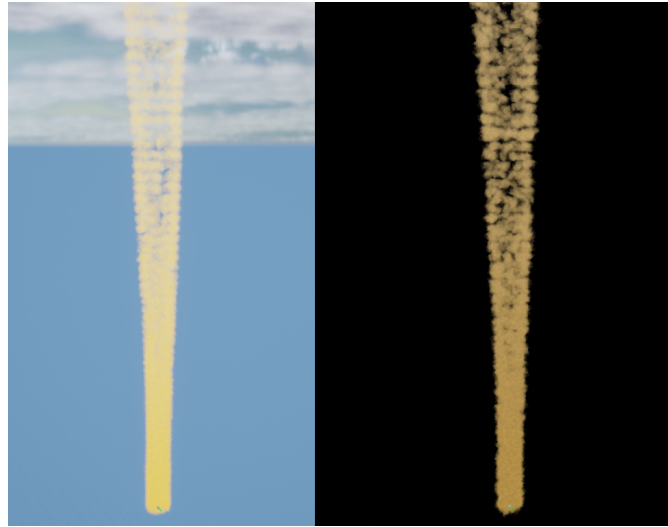


Figure 3.31: Failure version three



Figure 3.32: Failure version four

# Chapter 4

## Result

### 4.1 Introduction of result

In this chapter, we will apply the final effect obtained in the previous chapter to actual use to determine the final effect. I import both Niagara systems into the VaMex project and see how it works. I will explain the final result from three perspectives:

- Performance
- Visual effects in VaMex
- Comparison with the old version

### 4.2 Performance

In the process of producing effects, performance is an unavoidable topic. FPS is one of the most important but easily overlooked indicators. Here I tested the FPS in VaMex project under several conditions:

- FPS without importing special effects
- FPS when the particles of the main emitter of dust devil are 1000, 2000, 3000 respectively
- FPS when the particles of main emitter of dust storm are 50, 100, and 200 respectively

#### 4.2.1 FPS without importing special effects

As the figure 4.1 shows, if there is no VFX or Niagara System in the VaMex, FPS is really stable, ranging from 112 - 119.

This range can be used as our control variable control group, which allows us to see more clearly how the performance is after adding special effects.

## Chapter 4 Result

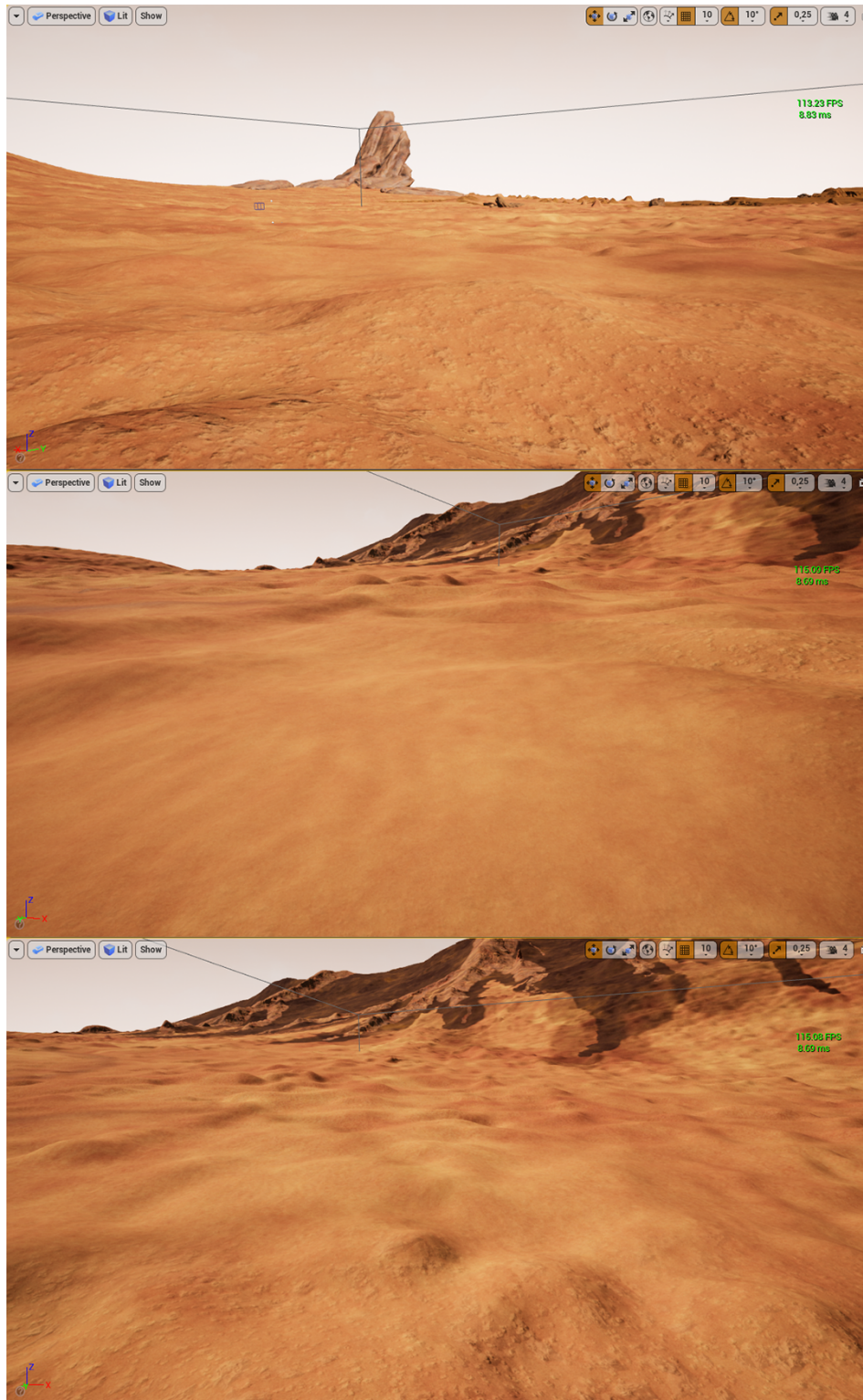


Figure 4.1: FPS without importing special effects

#### **4.2.2 FPS when the particles of the main emitter of dust devil are 1000, 2000, 3000 respectively**

Here, the main emitter of Dust Devil is Cylinder. I adjusted its particle number to 1000, 2000, 3000 respectively, and observed their FPS, and got the following range:

- 1000 Particles : FPS 103-119
- 2000 Particles : FPS 103-118
- 3000 Particles : FPS 103-118

It can be seen that the overall FPS value of VaMex is lower than that of VaMex without the Niagara system. But overall there is not much difference in FPS for different particle numbers.

In Figure 4.2 we can see the different visual effects of different particle numbers and their FPS.

#### **4.2.3 FPS when the particles of main emitter of dust storm are 50, 100, and 200 respectively**

In my detection situation, the most important emitter in Dust Storm is Sand Wall. In Dust Storm, the particle size in the main emitter is relatively large, so the number is relatively small. Here I took 3 suitable values: 50, 100 and 200. Below are their FPS ranges.

- 50 Particles : FPS 113-119
- 100 Particles : FPS 113-119
- 200 Particles : FPS 98-112

We can see that when the number of particles is 50 and 100, the FPS is almost the same as the project value when no special effects are imported. But when the number of particles becomes 200, the value of FPS drops a lot. Therefore, the number of particles in the main emitter in a dust storm is more appropriate to be between 50-100.

In Figure 4.3 we can see the different visual effects of different particle numbers and their FPS.

### **4.3 Visual effects in VaMex**

In the project VaMex, we can see the actual effects of dust devil and dust storm as shown in the Figure 4.4.

It can be seen that in VaMex, the height of the dust devil has a certain effect on visibility. The impact of dust storm is greater. As shown in the video data, a dust storm on Mars will severely reduce the visibility of the environment, will also block the sun, and will

Chapter 4 Result

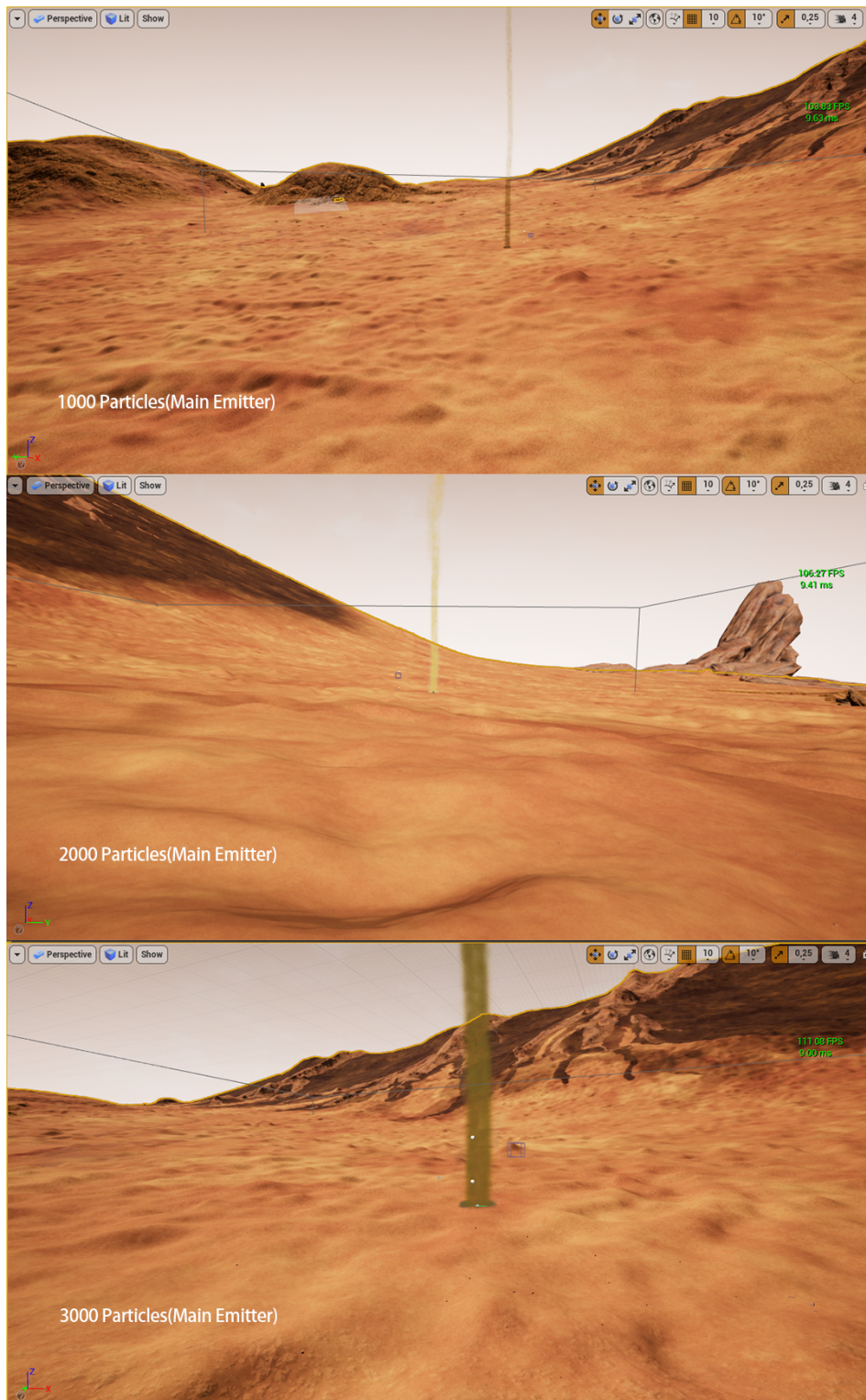


Figure 4.2: FPS with dust devils

## Chapter 4 Result



Figure 4.3: FPS with dust storms

be accompanied by lightning, which will have a great impact on the collection of solar energy on Mars.

Also because the robots in VaMex will visualize the Martian landscape data and use these data to communicate with each other, dust devils and dust storm will also affect the collection and interaction of data.

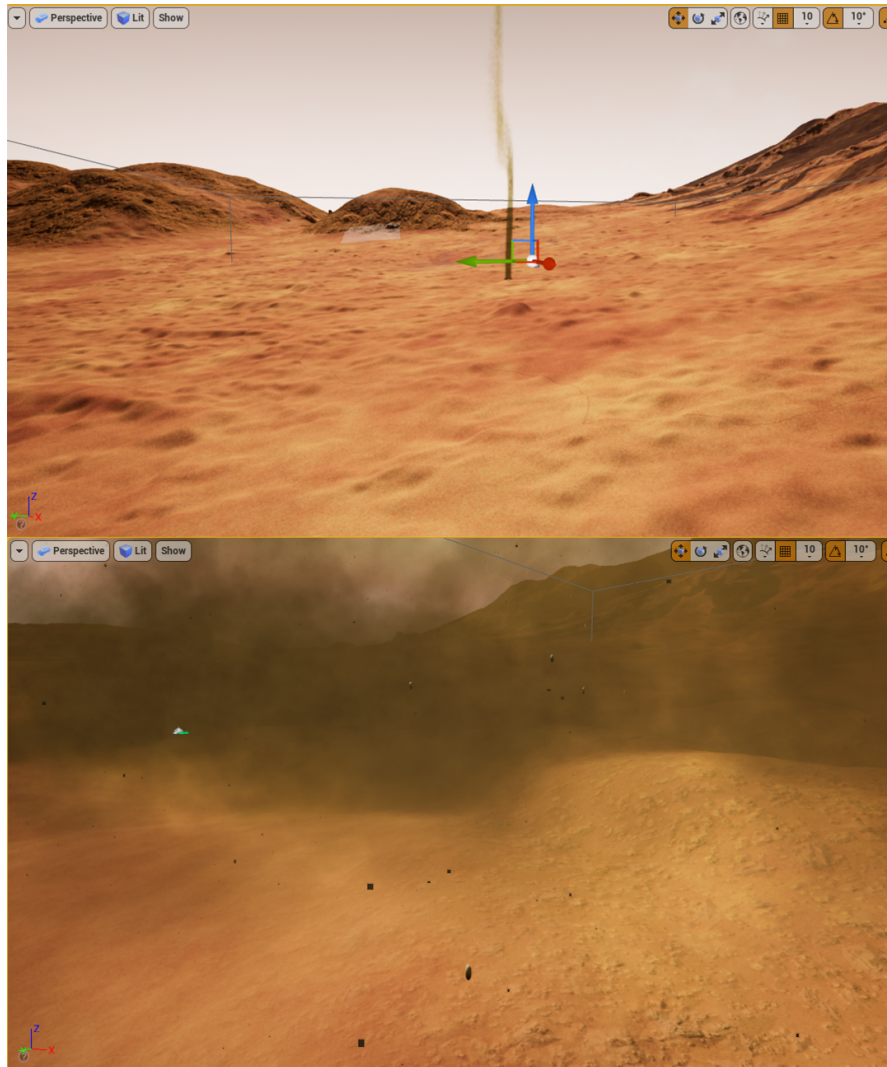


Figure 4.4: Visual Effect

#### 4.4 Comparison with the old version

On the introduction website of VaMex-VTB[1], we can find the previous special effects of the old version. In this part we will elaborate a simple comparison.



### 4.4.1 Dust Devils

In the old version of dust devil, the number of particles in general is not very large, and the particles at the top are more densely visually, and the movement speed of the particles is more regular and stable.

And my dust devil particles are more and denser, and the speed of the particles is faster. And the top particle has the effect of dissipating. Also, my dust devil is higher in height, has a certain slope, and swings more irregularly.



Figure 4.5: The comparison of two dust devils

### 4.4.2 Dust Storms

The old version of dust storm uses a more moderate version. The suspended particles seem not to be disturbed by too much external force, and they move regularly and slowly in the air. The overall effect is foggy and still has a certain degree of visibility.

My dust storm looks more violent. The speed of the forward movement is very fast, and the movement speed of the internal particles and sand is more violent and irregular. Visibility is very low. Also it is accompanied by lightning.

### 4.4.3 Summary

I cannot compare the different versions with any judgmental words. Natural phenomena on Mars due to different climates will also change over time. My version provides more reference versions for the project.

In Addition, I upgraded the VaMex project from UE4.25 to UE4.26 and uploaded it in the GitLab.

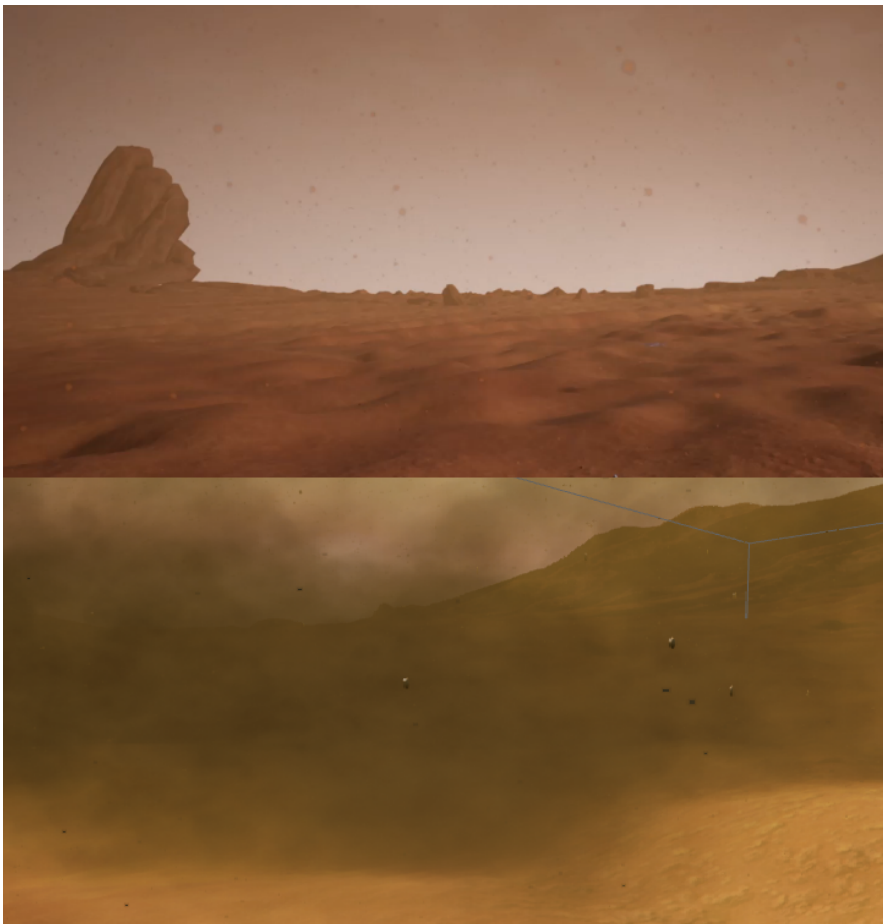


Figure 4.6: The comparison of two dust storms

# Chapter 5

## Conclusion and Future Work

### 5.1 Conclusion

Here I briefly summarize the work of my entire thesis.

First of all, I explained my motivation and purpose for making this paper. It is to add more usage scenarios to the project VaMex and add more possibilities to the exploration of Mars in the project.

Then I elaborated on my research on these two natural phenomena on Mars, summarized their morphological characteristics and other useful researches for my implementation. I also described the software I use and their inner principles. Of course, I also gave some previous research and works, they are very good, and gave me a good inspiration for the implementation.

Regarding implementation, I respectively stated my implementation idea according to different natural phenomena, how I set and adjusted the parameters in different emitters, and how their materials were completed. Of course I also gave the final effects(including those that failed).

I imported the effect obtained from the previous step into the VaMex project, observed the actual visual effect, and tested their performance. I also compared the effect I got with the old version. Then I uploaded the new version in GitLab under the name Dust\_Devil, which is updated to the UE 4.26.

### 5.2 Future Work

Mars exploration is still a very hot topic. Not only scientists and space agencies of various countries, but now entrepreneurs are also joining the ranks of exploring Mars. Therefore, it is of great significance to visually and physically simulate the scene of Mars, which can save time and money to better explore Mars.

According to the existing data, the natural phenomena and landscapes on Mars are far more than dust devil and dust storm. Others such as dunes, yardangs etc. also have visual significance. With the advancement of science, future work may add more different and interesting natural phenomena on Mars.

In addition, advances in various software will also affect future work. During the time I finished this paper, UE5 was officially released, and its rendering effect was unprecedentedly good. Of course, the Niagara system has also been upgraded, with many new functions. Various other modeling softwares such as blender and Unity are constantly

## *Chapter 5 Conclusion and Future Work*

updated in special effects or graphics rendering, and the effect is getting better and better.

I believe that with the advancement of technology, the field exploration and visualization of Mars will develop rapidly.

# Bibliography

- [1] Vamex-vtb. <https://cgvr.cs.uni-bremen.de/research/vamex-vtb/>.
- [2] Ronald Greeley Matt Balme. Dust devils on earth and mars. *Reviews of Geophysics*, September 2006. doi: 10.1029/2005RG000188.
- [3] Dust devils in northern arizona. <https://www.weather.gov/fgz/DustDevil>.
- [4] dust devil. *Glossary of Meteorology*. American Meteorological Society, 2016.
- [5] "dust devil" from wikipedia. [https://en.wikipedia.org/wiki/Dust\\_devil](https://en.wikipedia.org/wiki/Dust_devil), .
- [6] P. C. Sinclair. General characteristics of dust devils. *J. Appl. Meteorol*, 8:32–45, 1969.
- [7] Dust devil - smithsonian national air and space museum. <https://airandspace.si.edu/exhibitions/exploring-the-planets/online/solar-system/mars/wind/dust-devils.cfm>, .
- [8] Dust storms chapter. *Emergency Management Plan*, October 2013.
- [9] What is a dust storm? - noaa. <https://scijinks.gov/dust-storm/>.
- [10] Kathryn Mersmann. The fact and fiction of martian dust storms. *National Aeronautics And Space Administration.*, September 2016.
- [11] Mega dust storms — mars. [https://www.youtube.com/watch?v=JKBk\\_Kfucs4](https://www.youtube.com/watch?v=JKBk_Kfucs4).
- [12] Global storms - smithsonian national air and space museum. <https://airandspace.si.edu/exhibitions/exploring-the-planets/online/solar-system/mars/wind/storms.cfm>, .
- [13] "particle system" from wikipedia. [https://en.wikipedia.org/wiki/Particle\\_system](https://en.wikipedia.org/wiki/Particle_system), .
- [14] Niagara overview. <https://docs.unrealengine.com/4.27/en-US/RenderingAndGraphics/Niagara/Overview/>, .
- [15] Niagara renderers. <https://docs.unrealengine.com/4.27/en-US/RenderingAndGraphics/Niagara/EmitterReference/RenderModules/>, .
- [16] "physically based rendering" from wikipedia. [https://en.wikipedia.org/wiki/Physically\\_based\\_rendering](https://en.wikipedia.org/wiki/Physically_based_rendering), .

## Bibliography

- [17] Physically based materials - unreal engine 4.27 documentation. [https://en.wikipedia.org/wiki/Physically\\_based\\_rendering](https://en.wikipedia.org/wiki/Physically_based_rendering).
- [18] Mars' whirling dust devil - nasa jet propulsion laboratory. <https://www.jpl.nasa.gov/videos/mars-whirling-dust-devil>, .
- [19] Large-eddy simulation of dust devils - animation of dust devils in the convective boundary layer using a virtual dust source. <https://av.tib.eu/media/9352>, .
- [20] The martian (film) from wikipedia. [https://en.wikipedia.org/wiki/The\\_Martian\\_\(film\)](https://en.wikipedia.org/wiki/The_Martian_(film)), .
- [21] Mpc the martian vfx breakdown. <https://www.youtube.com/watch?v=5CPbISkPqKk>, .
- [22] Martian dust devils — national geographic. <https://www.youtube.com/watch?v=oTLZci5dp44>, .
- [23] Ue4 niagara particle system quick introduction. <https://www.aboutcg.org/courseDetails/1194/introduce>, .
- [24] Pbr cg textures ground sand and beach (ground 0024). <https://www.texturecan.com/details/215/>, .

# List of Figures

2.1	A dust devil in Arizona . . . . .	4
2.2	Diagram courtesy of Malin Space Science Systems . . . . .	5
2.3	Serpent Dust Devil of Mars . . . . .	6
2.4	A dust storm approaches downtown Phoenix on August 11, 2012. Credit: NOAA[9] . . . . .	6
2.5	A sandstorm approaching Al Asad April 27, 2005. . . . .	7
2.6	The Hubble observed the 2001 dust storm from Earth orbit. . . . .	8
2.7	A particle system used to simulate a fire, created in 3dengfx. . . . .	9
2.8	An overview of Niagara UI . . . . .	10
2.9	Different Renderers . . . . .	11
2.10	The interface of PBR system in UE4 . . . . .	12
2.11	Material with different roughness . . . . .	12
2.12	Material with different metallic. . . . .	12
2.13	Original photo by HiRISE camera on NASA's Mars Reconnaissance Orbiter. . . . .	13
2.14	The animation by NASA. . . . .	14
2.15	Large-eddy simulation of dust devils . . . . .	15
2.16	VFX Breakdown of dust devils in The Martian . . . . .	16
2.17	VFX Breakdown of dust storms in The Martian . . . . .	17
3.1	There will be foggy sand around the dust devil. - from Youtube channel National Geographic "Martian Dust Devils — National Geographic" . . . . .	19
3.2	The parameters of the first emitter. . . . .	20
3.3	Visual effect of the first emitter. . . . .	21
3.4	Initialize Particle . . . . .	21
3.5	The parameters of Particle Spawn. . . . .	22
3.6	Parameters of the Particle Update. . . . .	22
3.7	Visual effect of the second emitter. . . . .	22
3.8	Parameters of the second emitter. . . . .	23
3.9	Parameters of the third emitter. . . . .	24
3.10	Visual effect of the third emitter. . . . .	25
3.11	M.smoke_subUV Material . . . . .	26
3.12	Visual effect of the whole niagara system . . . . .	26
3.13	Visual effect of the first emitter. . . . .	28
3.14	Parameters of the first emitter. . . . .	28
3.15	Visual effect of the second emitter. . . . .	29
3.16	Parameters of the second emitter. . . . .	29
3.17	Ribbon Renderer . . . . .	30

*List of Figures*

3.18	Emitter Update . . . . .	30
3.19	Parameters of the lightning emitter. . . . .	31
3.20	Visual effect of the lightning emitter. . . . .	31
3.21	Parameters of the micro-particles and rock emitter. . . . .	32
3.22	Visual effect of the micro-particles and rock emitter.(With Sand Wall in the background) . . . . .	33
3.23	Overview of M.Lightning. . . . .	33
3.24	Overview of M.Sand. . . . .	33
3.25	Blueprint of M.Lightning. . . . .	34
3.26	Blueprint of M.Sand. . . . .	34
3.27	Visual effect of dust storm. . . . .	35
3.28	Visual effect of dust storm. . . . .	35
3.29	Failure version one . . . . .	36
3.30	Failure version two . . . . .	36
3.31	Failure version three . . . . .	37
3.32	Failure version four . . . . .	37
4.1	FPS without importing special effects . . . . .	39
4.2	FPS with dust devils . . . . .	41
4.3	FPS with dust storms . . . . .	42
4.4	Visual Effect . . . . .	43
4.5	The comparison of two dust devils . . . . .	44
4.6	The comparison of two dust storms . . . . .	45



# Acknowledgements

First and foremost, I am deeply indebted to my supervisor Rene Weller, who has helped me all the time when I do my thesis. Second, I am really grateful for my college Marc who spent a lot of time helping me to import my VFX into the project.