北京郵電大學
Beijing University of Posts and Telecommunications

Queen Mary
University of London

Undergraduate Project Report 2021/22

Large-scale scene simulation of games in cold-temperate deciduous coniferous forest area based on UE

Name:	Wang Zhiyuan
School:	International School
Class:	2018215114
QMUL Student No.:	190017767
BUPT Student No.:	2018212993
Programme:	e-Commerce Engineering with Law

Date: 12-4-2022

Table of Contents

Large-scale scene simulation of games in cold-temperate deciduous coniferous forest area based on UE

Abstract

Scene design refers to the modelling design of all objects that change with time, except for character modelling. It is an important part of the Game, as well as the Film and Television industries. Scenario simulation refers to simulating the real area for scenario design. In the natural geographical environment, vegetation forms several natural zones according to latitude and terrain height and forms vertical and horizontal distributions according to heat and moisture. This study proposes a method to efficiently produce the natural environment of the big world in the Unreal Engine. The research target selects the cold temperate zone in northern China and uses procedural content generation (PCG) technology to reconstruct the vegetation types and distribution patterns in this area to build a realistic 3D cold temperate game scene. Finally, the system can automatically cover materials, generate corresponding plant assets according to the real terrain, and has a weather system. This study digitizes the distribution law and seasonal performance of landforms and vegetation, and the generated scenes have important research significance for computational geography, digital content production, and game production.

Keywords: Unreal Engine, Game Scene, Simulation, Procedural Generation, Material

Large-scale scene simulation of games in cold-temperate deciduous coniferous forest area based on UE

摘要

场景设计指除角色造型之外的随时间变化的一切物体的造型设计，是游戏与影视产业中重要的一环。场景仿真指模拟现实地区进行场景设计。在自然地理环境中，植被随纬度和地形高度规律形成了数种自然地带，根据热量和水分形成了垂直与水平分布。本研究提出了一种在Unreal Engine中高效制作大世界自然环境的方法，研究目标选择中国北方的寒温带，使用程序内容生成（PCG）技术重建该地区的植被类型和分布模式，建立逼真的3D寒温带游戏场景。最后，系统可以根据真实地形自动覆盖材质与生成对应的植物资产，并带有天气系统。本研究将地貌与植被分布规律与季节表现数字化，所生成的场景对计算地理学、数字内容制作和游戏制作具有重要的研究意义。

关键词：Unreal Engine，游戏场景，仿真，程序化生成，材质

Chapter 1: Introduction

Video game art is a significant type of digital media art. After painting, sculpture, architecture, music, literature, dance, drama, and film, video games are the ninth art category after painting, sculpting, architecture, music, literature, dance, drama, and film. Video games have entered the next generation of 3D with the progress of electrical hardware and game creation technology. Players' expectations for the type and quality of games are rising in tandem with the globalisation of the culture and entertainment industries, and next-generation games have begun to reach the domestic market.

For game developers, in the production of specific game projects, in addition to using the next-generation game production technology represented by normal mapping technology, the most important technical support comes from the game engine.[1]

1.1 Objectives

Game engines and next-generation production technologies are used in the development of next-generation games. The game engine is the key technology, since it provides a platform for game creation and has an impact on the game's design and production process, as well as the final presentation effect. A solid game engine makes game production more easier and quicker. One such engine is the Unreal Engine. Engine (Unreal Engine) is Epic Games' most recent industry-leading gaming engine, which is extensively utilised in Europe and North America.

The release of Unreal Engine 4 provides technical options for the development of next-generation games for domestic game developers. In the domestic game development business, there are still theoretical and practical gaps in the use of Unreal Engine and the fundamental technologies of next-generation game creation.

The development of the next-generation game is a huge project. Due to the length of the paper, the author takes the next-generation game scene in the cold temperate zone of China as the research object, uses Unreal Engine4 as the game development platform, and combines the PCG technology of the next-generation game. The production and implementation of the next-generation big-world game scene design is the purpose of the whole research. Through the author's specific practice, the key implementation technologies of the next-generation big-world game scene are studied and summarized.

1.2 Achievements

To simulate and digitise the natural environment in the cold temperate zone of China, to study the production technology of next-generation game scenes, and to use Unreal Engine as a game development tool to study the realisation of game scene design. It focuses on the in-depth research, discussion, and summary of the production process and implementation technology of the next-generation scene design in Unreal Engine. The main research results of this paper include:

(1) The Unreal Engine's meaning and scope, as well as its current application status at home and abroad, are examined, and the benefits of the Unreal Engine for next-generation game creation are clarified.

(2) Plant model of birch forest built in China's cold temperate zone.

(3) Choosing regional terrain, terrain accuracy, and preprocessing and loading terrain maps into UE5.

(4) Create PBR materials in Unreal Engine, analyses material production and material editor implementation, investigate the specific implementation methods of three types of materials: terrain materials, vegetation materials, and water materials, and look into the role of material function node functions. Summarize the PBR materials' production concepts and procedures.

(5) Fully automated terrain-matching asset generation using PCG technology is possible. Automated generation of different types of vegetation using two different vegetation PCG production methods—PFS and LGT—has been proposed.

Through the author's personal creation, we summarise the key technologies, analyse and summarise the problems and difficulties encountered in the production process, and finally realise the next-generation game scene design work named "White Birch."

Chapter 2: Background

2.1 Terrain Editing in Next-Generation Game Design and Natural Scene Design

2.1.1 Introduction to Next-Generation Games

The term "next-generation game" refers to games that are more advanced than their predecessors.[2]

Unlike traditional game production, the present development of next-generation games frequently need improved art effects and screen performance, which is also one of the most distinguishing elements of next-generation games. Engine rendering, material shading, lighting, PCG, scripting, visual effects (FXs), and scene optimization are examples of related technology. The creation of gaming scenes, for example, necessitates the use of technologies supported by most engines, and is a masterwork of engine rendering technology.

2.1.2 Introduction to Game Terrain Editing

Untiy and UE4 are the current mainstream game production engines in use in the domestic mainstream game development environment, and the local editor is in charge of using the 3D scene model to create maps in the engine, creating sky and light effects, and determining the overall production style of the map. Then you'll need to create textures, edit or create models, create innovative unique artwork, and make technical adjustments. Importing the model into the engine, creating the game scene, debugging the model, debugging the scene, and performing basic run tests are all examples of the task.

2.1.3 Natural scene design

In the gaming centre, game scene design refers to the modelling design of everything but character modelling. The game character, or the primary body controlled by the players, is the game's main body. They encircle the main body and everything connected to it, such as living spaces, furniture, the social environment, the natural environment, and the historical environment.

The gaming scenario reinforces the time-space link and adds to the game's atmosphere. The scene can create a certain atmospheric effect and emotional tone by reflecting the story's regional characteristics, historical generations' style, national culture's characteristics, and the characters' living environment, which is also the difference between scene design and architectural environmental art design.

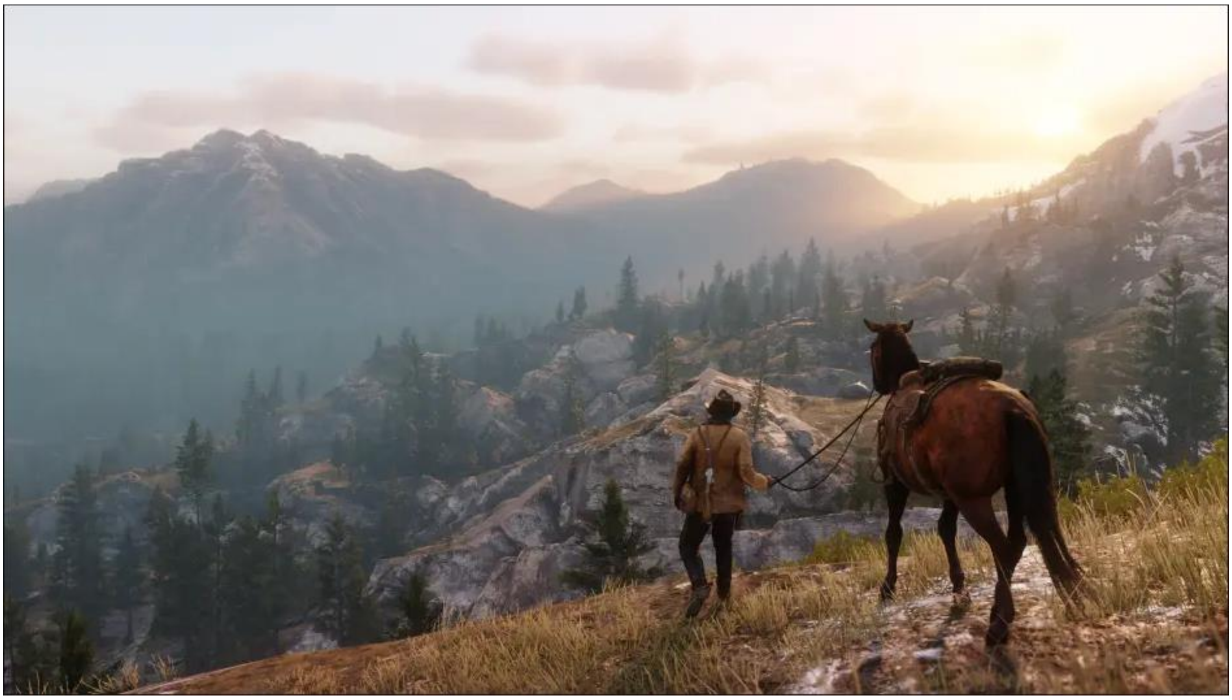


Figure 2.1.1 Game screenshot in *Red Dead Redemption 2*

Scene design's most popular and essential category is natural scenes. Everything that exists in nature is included in the so-called natural scenes, from close-ups of weeds on the side of the road to stunning scenery reaching out to the distant horizon.

Natural scenes are constantly changing in appearance. The variations in natural settings are caused by the changing seasons of sunlight, climate, and weather conditions, as well as the exquisite colours and tints of natural light. The concept of static natural scenery must be abandoned because natural scenes change with the seasons. In the spring, plants sprout new green, filling the air with freshness; in the summer, the sun shines brightly, full of life; in the fall, the mountains and fields are draped in brocade; and in the winter, the land is dressed as an angel in white. Natural beauty and charm can be found in these dynamic shifts.[3]

It is required to recreate natural settings as an important aspect of natural scenes in VR and 3D games. The accurate simulation of scene terrain and scene trees has received a lot of attention.[4]

2.2 Introduction to Unreal Engine

2.2.1 Introduction to Game Engines

The basic technology in game development is game engine technology. Game engines control game components, allowing a complete game to be created. The game engine is responsible for all of the game's functions. Early game engines arose from the concept of code reuse, with a range of reusable components and the game's overall architecture being used to create early game engines.

The current game engine is a complex collection system containing complex subsystems, including: rendering engine (i.e. "renderer," which includes 2D and 3D image engines), physics engine, collision detection system, sound effects, scripting engine, computer animation, artificial intelligence, network engine, and scene management, as well as other components.

2.2.2 Unreal Engine Overview

The Unreal Engine is a game engine developed by the American game company Epic Games. The Unreal Engine is a complete set of development tools for anyone working with real-time technologies. Capabilities range from designing visualisations and cinematic experiences to producing high-quality games on PC, console, mobile, VR, and AR platforms.[5]

The addition of a blueprint visual script editor and a C++ code programming system to Unreal Engine 4 is the most significant update and enhancement. Unreal Engine 4 introduces a new material creation technology called PBR, which is characterised by the use of physically-based colouring approaches to improve the texture of the model material in the picture.

Unreal Engine is used by more than 7.5 million developers globally, according to Unreal Engine's official statistics. The Unreal Engine is a sophisticated rendering computer as well as a game engine. In the realms of TV, film, advertising, architecture, and visual design, it is also utilised by designers in the film and television industries, the construction business, and the advertising industry. According to Daniel Ahmad, a well-known industry expert, an increasing number of Chinese game developers are employing the Unreal Engine 4 engine. The number of developers using Unreal Engine 4 on PC will expand by 280 percent by 2020, while the number of developers using Unreal Engine 4 on mobile devices will increase by 280 percent. The number of employees climbed by 533 percent.

2.3 Introduction to the cold temperate zone in China

The cold temperate zone, also known as the subpolar continental climate, generally refers to the region around 50 °N to 65 °N. Mainly distributed in Eurasia and northern North America, only the northern part of the Greater Khingan Mountains and Heihe City belong to the cold temperate zone in China. The climate type, vegetation species, and weathering forms in this area are obviously different from other landscape areas in my country.

The typical characteristics of cold temperate regions are long and severe winters, short and warm summers, a large annual temperature range, and annual precipitation of 300–600 mm, but with weak evaporation and slow melting, so the relative humidity is high. The cold temperate zone is mostly composed of middle-low mountains and a hilly landform. The surface vegetation is relatively developed. The plants are mainly cold-resistant larch, spruce, and other conifers. The leaves are slender, needle-like, with a thick cuticle. [6]

The cold temperate coniferous forest is a forest type composed of cold-resistant coniferous trees as constructive species, also known as the boreal coniferous forest. The cold temperate coniferous forest is a typical horizontal zonal vegetation type in the cold temperate zone. It is distributed in the northern parts of Eurasia and North America, forming an obvious coniferous forest belt. In the middle and low latitudes and higher mountains, there are also cold-tempered coniferous forests, which constitute the forest vegetation in the vertical belt of the mountains.

The appearance of coniferous forests is often a single tree species. This constitutes a pure forest. The standing trees are straight, the community structure is simple, and the layers are clear. It is primarily made up of spruce (Picca), fir (Abies), larch (Iarix), and pinus (Pinus) species, with needle-like leaves to accommodate the short growing season and low temperature environment. [7] Therefore, the cold temperate forest of the Greater Khingan Mountains in China is called the cold temperate coniferous forest mixed with broad-leaved trees. [8]

The research object of this paper is the representative of the cold temperate zone in China, the Greater Khingan Mountains, mainly to study the bionic research of mixed coniferous and broad-leaved forests, to restore plant species, distribution patterns, seasonal changes and scene performance effects.

Chapter 3: Design and Implementation

3.1 Game scene setting and Development environment

The ***game scene*** encompasses all of the activities of the characters in the game. The scene in the game is in charge of explaining the game's time and setting, as well as establishing the game's atmosphere. The performance of picture textures is more important in next-generation games. The portrayal of scene elements, particularly the application of model and material technologies, is both the crucial and the most difficult aspect of gaming scene design. How can the many aspects in the gaming scene, such as topography, vegetation, ambient light, and so on, be assembled in a high-quality and cost-effective manner? This necessitates a thorough examination of the key components and design methodologies used in next-generation game development.

The game scene ***White Birch*** focuses on simulating the topography of the Daxing'an Mountains, depicting undeveloped virgin forests and reflecting seasonal changes as its main purpose, focusing on the forest vegetation system represented by birch trees.

3.1.1 Elements of the game scene

a) Game Terrain: The game terrain is the most basic aspect in the game scenario, and it is where all of the action takes place. Terrain is a broad term for ground shape and landforms, and it refers to the varied ups and downs offered by surface objects as a result of crustal movement and external force erosion. Mountains, plateaus, plains, hills, and basins are the five types of terrain. Rivers, lakes, and waterfalls are generated as a result of external factors such as rain erosion, river erosion, and wind erosion. This research makes use of real-world elevation photographs to recreate real-world terrain.

b) Vegetation: Vegetation is a group of plant communities that cover the surface of the terrain. Plant communities can be classified into two sorts based on their types of vegetation: meadow vegetation and forest vegetation. Vegetation is strongly tied to natural environmental components such as climate, soil, topography, and water conditions in the actual world. The design and production of vegetation in the game scene should also precisely follow the game script in order to set the season and climate.

c) Ambient light: light can express a game scene's ambiance and emotions as well as its outline and shape. The difference between light and shade and brightness in the entire scene is depicted and performed by the light source, and the size of the light region, the opening and closing of space, the change of shape, and other factors in the gaming scene make a rhythmic rhythm with space sequence. [9]

d) Seasonal effect: A season is a span of time that changes dynamically throughout time, and cyclically appearing geographical landscapes alter dramatically. Seasons are a crucial aspect of the environment, and the connection between plant terrain and seasons helps to create more realistic game scenes. The seasonality of the research subjects' cold temperate climate is lengthy and cold in winter, and short and sunny in summer. This research focuses on designing corresponding seasonal effects based on the research terrain zone.

3.1.2 Development Environment

Table 3.1.1: Development hardware environment

CPU	AMD Ryzen 9 5900X 12-Core
Motherboard	B550M-PLUS
RAM	32G DDR4 3200MHz
Graphic card	NVIDIA GeForce RTX 3080Ti
Monitor	LG GSM7706
Storage	SAMSUNG PM9A1

Table 3.1.2: Development hardware environment

Operating System	Windows 10 Professional 64-bit
Vegetation	SpeedTree for UE4
Terrain Generation	World Machine Professional
Elevation Processing	TerreSculptor 2.0
Game Engine	Unreal Engine 4.27/UE5 Preview

3.2 Plant model building

The virgin forest area in the northern part of the Daxing'an Mountains in Inner Mongolia is the best preserved, the only contiguous, and the largest undeveloped virgin forest in China. It is located in the cold-temperature bright coniferous forest belt and extends south to a part of my country. This forest ecological function area retains the virgin forest landscape and native animal and plant resources, and is used for water conservation, soil conservation, carbon and oxygen release, environmental purification, biodiversity protection, and greenhouse gas reduction. The total amount has played a huge and irreplaceable role in ecological balance in terms of alleviating climate warming and is known as the "Jade in the North".[10]

The virgin forest area in the northern part of the Daxing'anling is located on the northwest slope of the northern part of the Daxing'anling Mountains. It includes the 3 ecological function areas of Qigan, Uma, and Yonganshan, the key state-owned forest management bureaus of Inner Mongolia's Daxing'an Mountains. The geographic coordinates are: east longitude 120°01'20"12148'37", north latitude 52°01'42"53°20'00". The climate belongs to the cold, temperate continental monsoon climate. The winter is long and cold and dry, and the summer is short and humid, hot and rainy. The temperature difference between day and night varies greatly among the four seasons, and the plant growth period is short.

3.2.1 Plant species survey

According to the analysis of the characteristics of the main vegetation groups in the Daxing'an Mountains, the Daxing'an Mountains are rich in vegetation groups such as forests, shrubs, grasslands, meadows, swamps, and grass ponds.

Forest: According to the characteristics of plant composition, structure, and appearance, this area can be divided into 3 vegetation subtypes, namely: coniferous forest, mixed coniferous and broad-leaved forest, and broad-leaved forest.

i. **Coniferous forest:** This vegetation subtype is divided into two groups: the larch forest group and the pine forest group, Xing'an larch forest (Larix gmelinii) and the sycamore pine forest (Pinussyloestris var. mongolica Litv).

ii. **Coniferous and broad-leaved mixed forest:** Xing'an larch and birch (Betula platyphylla) mixed forest is only a transitional vegetation type in this area. Although it is unstable, birch trees play an important role in improving the forest soil in this area.

iii. **Broad-leaved forest:** The broad-leaved forest in this area is extremely simple in terms of tree species composition. The most common distribution is Betula platyphylla, followed by aspen forest (Populus davidiana), and other small areas of black birch forest (Betuladahurica).

Shrubs: There is one vegetation subtype, coniferous shrubs, in the virgin forest area in the northern part of the Greater Khingan Mountains. It contains 2 clusters: Pinus pumila and Betula ermanii.

Grassland: The grassland in this area is almost the same as the adjacent grassland in composition. Except for Stipa baicalensis, Filifolium sibiricum and Spodiopogonsibiricus are more common. Other drought-tolerant plants include Platycodonggrandiflorum and Papaver nudicaule. Typical xerophyte grassland grasses include Stipa baikal and Artemisia frigida; xerogenic small shrubs, mainly Thymus dahuricus, are not widely distributed in the meadow steppe in this area, and the changes in composition are not very common. At large, there are two biome groups, namely: Filifolium sibiricum and Xing'an thyme grassland (Thymus dahuricus). [11]

From this, I started to select my target vegetation based on the characteristics of each vegetation biome.

Forest: Four representative tree species were selected, namely: birch, larch, black spruce and spruce, covering coniferous forest, mixed coniferous and broad-leaved forest and broad-leaved forest.

Table 3.1: Tree Type Table

Tree Type	Latin name	Forest Type
White Birch	Betula platyphylla	Broadleaf forest
Xing'an Larch	Larix gmelinii	Coniferous forest, mixed coniferous and broad-leaved forest
Black Alder	Alnus cremastogyne Burk	Broad-leaved forest, mixed coniferous and broad-leaved forest
Spruce	Picea asperata Mast	Broad-leaved forest, mixed coniferous and broad-leaved forest

Shrubs: There is one vegetation subtype, coniferous shrubs, in the virgin forest area in the northern part of the Greater Khingan Mountains. The most representative pine shrub (Pinuspumila) was selected. In addition, forest saplings also constitute an important part of the shrub vegetation, including pine saplings, black saplings and spruce saplings.

Grassland: The distribution of meadow grassland in this area is not very common, and there is little change in composition. There are two groups of groups and two groups, and I chose Filifolium sibiricum. The specific plants are as follows: Filifolium sibiricum, Chrysanthemum chanetii, Stipa baicalensis, Spodiopogonsibiricus, Platycodonggrandiflorum and Bellis perennis L, etc.

Table 3.2 Grassland Type Table

Plant species	Latin name	Phylogeny
Pine bushes	Pinuspumila	Coniferous bush
Xeryl daisy	Filifolium sibiricum	Prairie (Clematis prairie)
Red chrysanthemum	Chrysanthemum chanetii	Prairie (Clematis prairie)
Baikal Stipa	Stipa baicalensis	Prairie (Clematis prairie)
Big oil mango	Spodiopogonsibiricus	Prairie (Clematis prairie)
Bellflower	Platycodongrandiflorum	Prairie (Clematis prairie)

Daisy	Bellis perennis L	Prairie (Clematis prairie)

During the construction of the vegetation model, in order to facilitate the model construction, the vegetation is divided into two categories: trees and other plants.

Trees include 4 types of trees, each type of tree contains several sub-species, including trees of different age groups, such as large trees, medium trees, small trees and young trees. Among them, large trees and medium trees constitute the forest part of the vegetation model, while small trees and young trees belong to the shrub part.

3.2.2 Vertical and Horizontal Distribution of Plants

The height of biological distribution determined by the relationship between ground heights is referred to as the vertical distribution of plant communities. The temperature gradually drops from the foothills to the summit of the mountain. Temperature becomes the limiting factor of distribution from the foothills to the summit of the mountain, and many biological distribution boundaries arise. The mosaicism of plant communities' horizontal structure is its most distinguishing feature. Mosaic occurs when plant individuals are distributed unevenly in the horizontal direction, leading in the establishment of many tiny communities. The inhomogeneity of ecological elements, such as changes non-small topography and micro topography, is linked to the creation of small communities.

Species and number are the most important aspects in this study's vertical distribution of plants. Different species of plants have different quantitative performances at different altitudes as a result of different production ranges, with low-altitude broad-leaved forest, middle-altitude coniferous and broad-leaved mixed forest, and transition to high-altitude coniferous forest as the overall performance.

Plant distance and mosaicism are the most important determinants for the horizontal distribution of vegetation communities in this study (small community). Plant species are gradually enriched and plant communities are developed by using multiple seeding spacing modes in PFS to reflect the generation interval of the same plant.

The distribution quantity of the mixed forest of birch and larch in the Greater Khingan Mountains is different. This study refers to the random sampling data of 6 kinds of mixed forest of birch and larch within a range of 40 meters, and there are 20.5 large-diameter trees, 16.5 middle-diameter grades, and 15 small trees, as shown in the ***Appendix table***. A primary plant model was identified. [12]

3.2.3 Plant Model Pre-processing

3.2.3.1 Pre-processing Meaning

Optimize the scene and fine-tune the plants to meet the basic needs

Naming: Due to the large number of models in the scene, it is convenient for producers to manage clearly;

Coordinates: The coordinates are returned to zero, so that the imported position of the model can be kept consistent;

3.2.3.2 Model Number Pre-processing

Determine species richness from literature, take 50

3.2.3.3 Model Quality Pre-processing

Select the suitable plant assets, which primarily refers to the model's quantity of vertices and triangles. If the model's accuracy is too great, it must be pre-processed for quality, which includes choosing an acceptable LOD layer, using low polymorphism, and so on.

1) Unit setting: Because one unit equals one centimetre in Unreal Engine, the unit must also be set to one centimetre when building in 3D programme to avoid issues during the model import process.
2) Because the number of faces in a single model in the Unreal Engine must be less than 65,000, it's critical to keep the number of model faces under rigorous control during the model creation process so that the finished model may be imported into the engine and run smoothly. The solution for models with too many model faces is to layer the models and import them in order. If the total number of faces in a single model of the building exceeds 65,000, each section of the model is divided and categorised with various IDs, limiting the total number of faces in a single part of the model to less than 65,000.
3) Some unreasonable faces should be avoided in the model, such as faces with 5 sides and more than 5 sides.

3.3 Terrain Elevation

Elevation points are information points marked with elevation values. The distribution expression of elevation points has rich characteristics, and the distribution relationship between important terrain feature units (such as mountains, saddles, and valleys), rivers, roads, and other natural or human-element targets makes the individual elevation points have different significance in the geographical nature.

3.3.1 Region Terrain Selection

When creating or importing terrain, the first thing to consider is the size of the terrain. Oversized terrain will carry more content, and a large amount of content will increase the load on the engine. At the beginning of the project, the appropriate size of the terrain needs to be considered; The maximum level streaming map supported by the Unreal Engine is 8129*8129 pixels, and the corresponding real map is 8129m*8129m.

In this study, Google Earth was used to conduct a preliminary topographic survey. In order to better show the richness of the topography, a piece of topography in the northeastern part of the Greater Khingan Mountains was finally selected, and this area was finally selected as the research object. The terrain is mostly gentle with slopes and is easy to stratify vegetation, including mountains, plains, and two rivers.

3.3.2 Real Terrain Elevation Source

There are many options for terrain elevation sources, the most famous of which is the Google Earth online source. When selecting the elevation source, the source is excluded based on the principle of accuracy first. Initially, the terrain source with poor accuracy (8m/pixel) was used at https://terrain.party/ , and then the terrain source was changed to select high precision (1m/pixel, terrain source: https://portal.opentopography.org). This greatly improves the overall accuracy of the terrain. The effect of the terrain is better, and the realism is greatly increased.

3.3.3 Terrain Elevation Map Processing

The downloaded .geotiff file cannot be directly imported into UE. UE only accepts input in some formats, and the elevation map needs to be stretched on the Z axis for better visual effects. This requires preprocessing of the downloaded terrain elevation map.

This research uses TerreSculptor software. First, import the .geotiff file downloaded by the terrain source OpenTopography, open it in TerreSculptor, select 32-bit grayscale (32-bit grayscale float) when importing, and modify the Z-axis attribute of the terrain elevation map after importing to observe the mountains. The degree of undulation, and choosing a height that matches the prototype. The final output selects the highest quality and outputs a 16-bit grayscale output (16-bit grayscale) to generate a.png file.

3.3.4 Import to Unreal Engine

The landscape editing tool in the Unreal Engine can create the terrain of the game scene in the project. There are two editing modes, namely creating a terrain heightmap and using the relevant sculpting tools to describe the terrain, or generating a terrain heightmap from an external tool and then importing it into the Unreal Engine to create it.

Select Landscape on the UE interface, select the corresponding terrain size (8129*8129 is used in this study), and then select to import the terrain map (.png) from the outside to import the terrain. The imported terrain comes with a default material, which then needs to be replaced with the terrain material.

3.4 Terrain Material

3.4.1 Unreal Engine Blueprint Editor

The blueprint editor, which gives a full solution, is the most representative aspect of the Unreal Engine. It is distinguished by the fact that it encapsulates several programming languages into modules made up of nodes, and users just drag these functions into it. To achieve related functions, modules can be joined together using a graphical interface. When compared to other game engines that use code to design and perform these functions, the blueprint approach makes it easier and faster to run the related operations.

Blueprints are a type of object-oriented programming. Developers must first comprehend the logical concept of the player entering a specific location to activate anything, such as the player automatically initiating the timeline to control the weather system at the start of the game, and how to communicate it. This feature can be accomplished thanks to the sequence. The logic behind blueprint thinking is as follows:

(1) The results obtained

(2) What caused this reaction? The result can be used to determine the cause of the occurrence.

(3) The implementation of such a function, as well as associated function nodes

3.4.2 Unreal Engine Material Editor

The Unreal Engine's material editor is a blueprint function that is implemented in the graphical editing shader. Its qualities are based on the PBR idea (physically based rendering). The performance of each material in the PBR framework is determined by a natural-laws algorithm that can accurately depict the object's material properties under the impact of the game environment and lighting, and this material property is based on the light and shadow parameters. As a result, the visual performance of the PBR material can imitate the material in real-world lighting conditions. PBR gives material production a unified standard thanks to a consistent specification based on physical attributes.

The elements of the PBR material are mainly composed of five parameters: (This can be placed in the appendix)

(1) basic color: a pure texture map (pure colour map) devoid of light and shadow information that represents the inherent texture or colour of the object's surface;

(2) Normal map: The convex and concave effect produced by changing the light on the surface of the object cannot express the convex and concave structure of the object's surface, so when the line of sight is close to the horizontal, the normal map cannot express its effect.

(3) Roughness: Roughness defines the smoothness of the material, which affects the absorption and reflection of light hitting the surface of the object. The simplest roughness is represented by a 1-dimensional array: 0 represents the absolute smoothness of the surface of the object, and 1 represents Completely rough, more complex materials are represented by a combination of multiple maps and arrays.

(4) Surface metallicity: Metallicity determines whether the property of the material is metal or non-metal and controls the relationship between the surface of the object and ambient light. Metals are characterised by a strong reflection of visible light, while non-metals only reflect a small fraction of the light when illuminated by a light source. The performance of metallicity can also be controlled by a simple array, but contrary to roughness, in a 1-dimensional array, 1 indicates the absolute metallicity of the object, and 0 indicates that the object is absolutely non-metallic.

(5) Highlights: The Unreal Engine's default highlight value is 0.5.In most cases, the default value of highlights is generally not adjusted. It is mainly used for translucent materials such as water, glass, etc.=

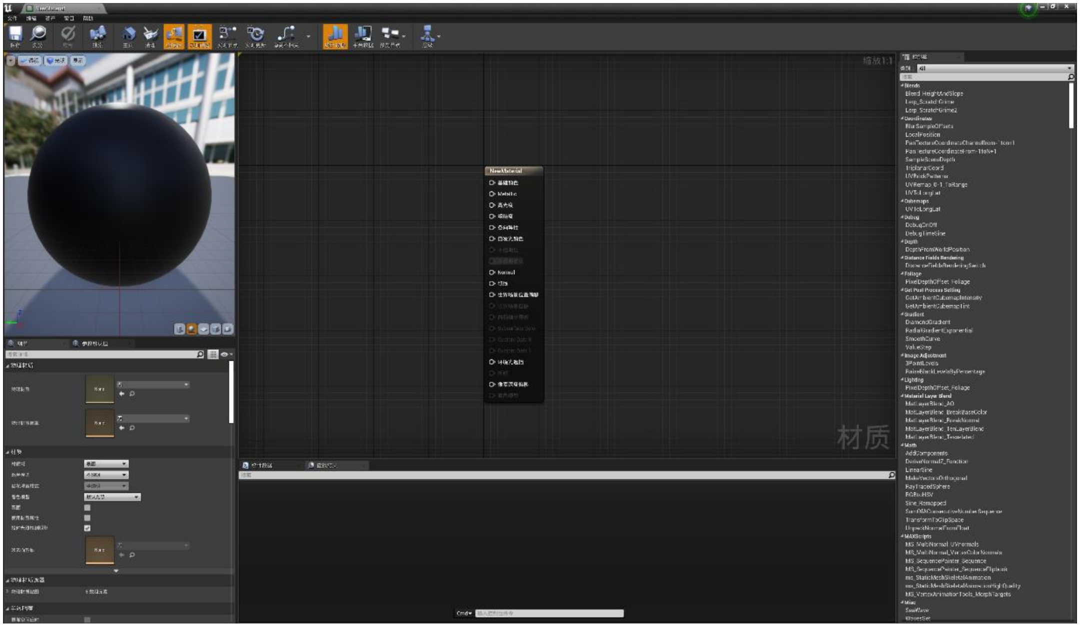


Figure 3.4.1 Material editor panel in Unreal Engine 4

The Unreal Engine that applies PBR uses a large number of visualisation nodes to control the material production instead of the previous method of combining multiple textures, which is also the biggest difference from the traditional next-generation game material production. The main external maps used in the material production of the Unreal Engine are the basic texture map, normal map, and roughness map, and the connection between the visual nodes is used to form the material. The benefit of this is that the producers do not have to spend energy and time drawing lighting, shadows, and other effects to the texture, and can instead focus on adjusting material properties. This working method not only improves the production efficiency but also improves the performance effect. It is also more realistic and believable than the traditional next-generation game materials.

3.4.3 Definition

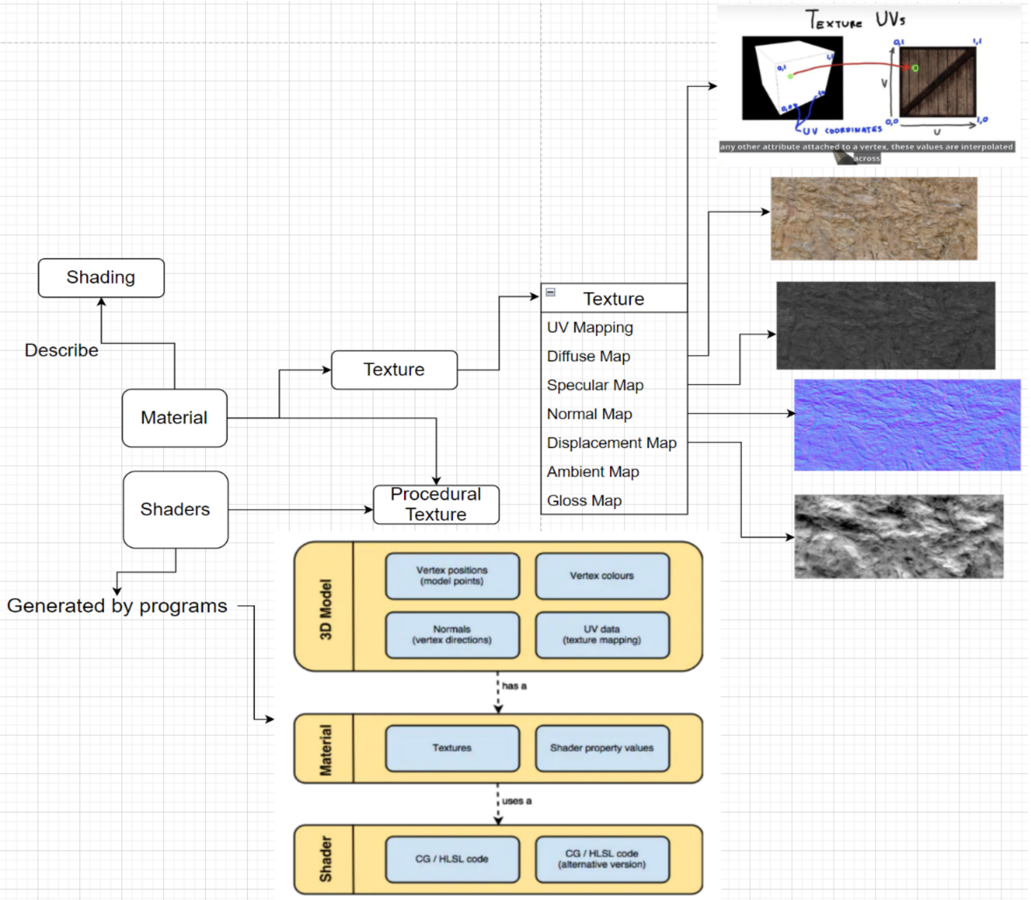


Figure 3.4.1 Game screenshot in *Red Dead Redemption 2*

In the process of creating materials, this study adopts the workflow of Material Functions -> Material Editor -> Material Instance. And create a material (parent material) first, and then create several material instances to apply to different models after adjustment.

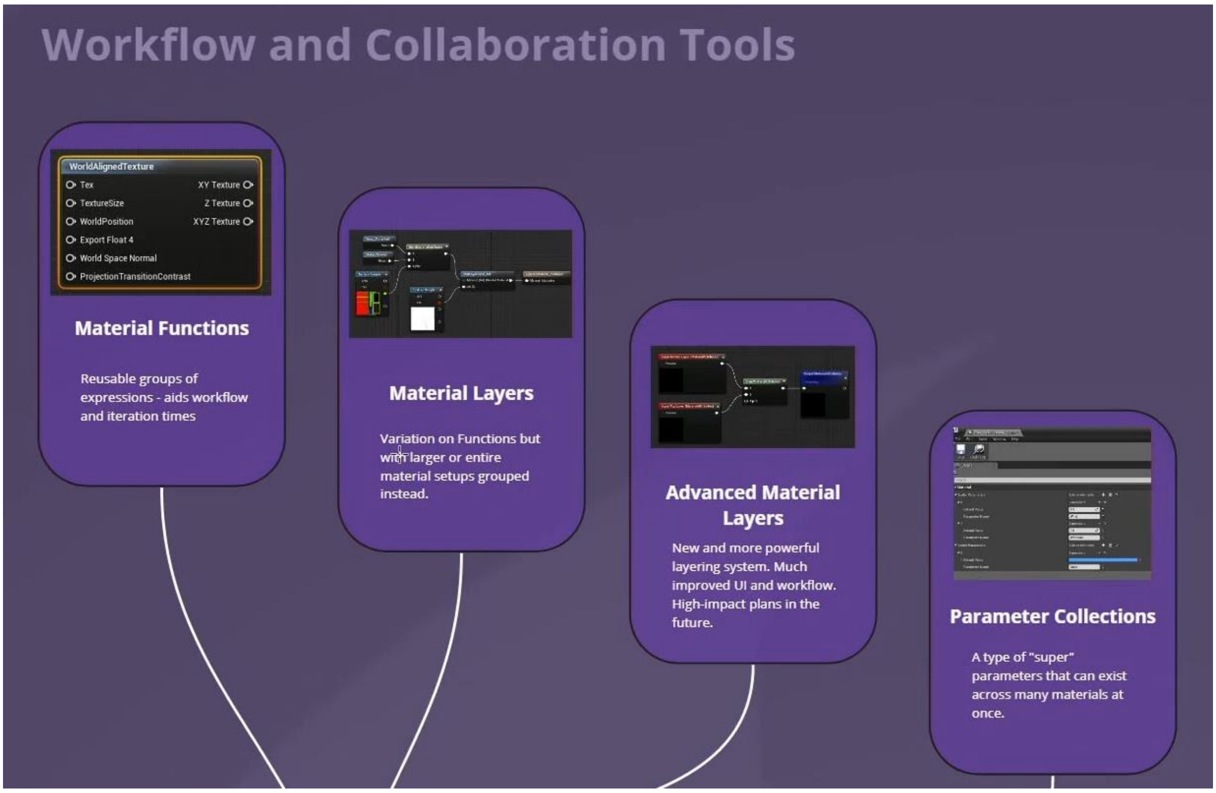
A material is often composed of the following four elements:

1) Material Function: A reusable set of several expressions that are combined

2) Material Layers: Material functions with extra functionality, usually with more content

3) Advanced Material Layers: More powerful layering system, more concise UI and workflow.

4) Parameter Collections: Parameter collections, a collection of variables similar to global variables, can play a role in multiple materials at the same time.



Workflow and Collaboration Tools

Material Functions

Reusable groups of expressions - aids workflow and iteration times

Material Layers

Variation on Functions but with larger or entire material setups grouped instead.

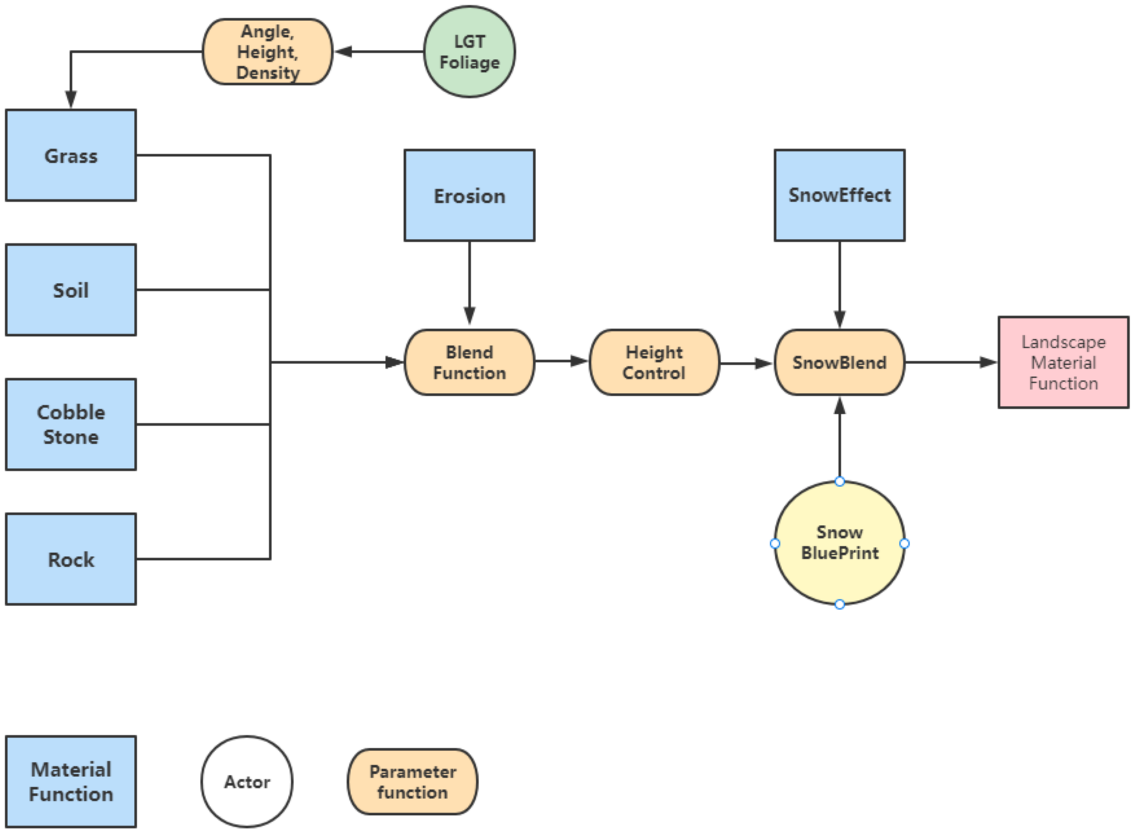
Advanced Material Layers

New and more powerful layering system. Much improved UI and workflow. High-impact plans in the future.

Parameter Collections

A type of "super" parameters that can exist across many materials at once.

The material instance system will select a parent material and inherit the parent function, and then extract the instance through a simplified material interface. You can change the parameters within the material instance to achieve different effects. Material Instances have additional Material Instance UI and are displayed in another window. It is worth noting that the material instance is run after compilation, so when modifying the material instance, you can often see the modification effect being modified in the scene in real time. Additionally, material instances can still be inherited and modified.


Angle,
Height,
Density
LGT
Foliage
Grass
Soil
Cobble
Stone
Rock
Erosion
SnowEffect
Blend
Function
Height
Control
SnowBlend
Landscape
Material
Function
Snow
BluePrint
Material
Function
Actor
Parameter
function

3.4.4 Introduction to Multilayer Mixed Materials

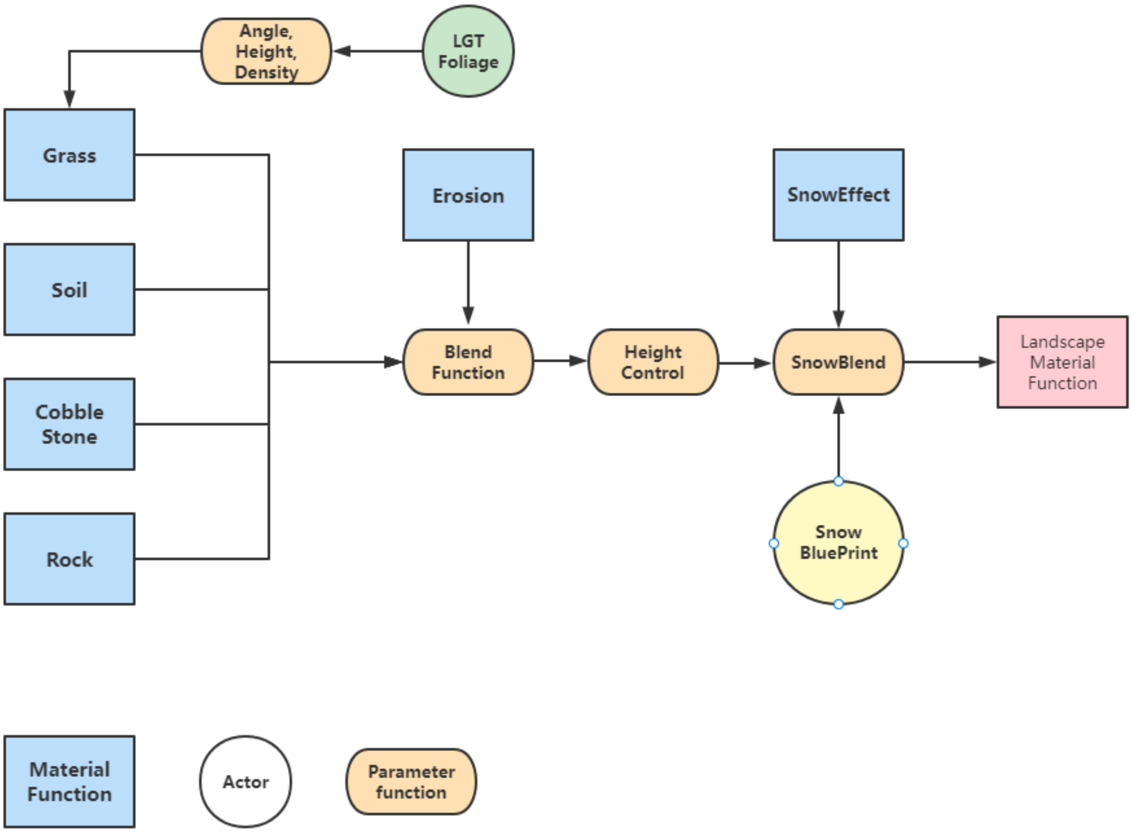


Figure 3.4.2 Landscape Material Function panel in *White Birch*

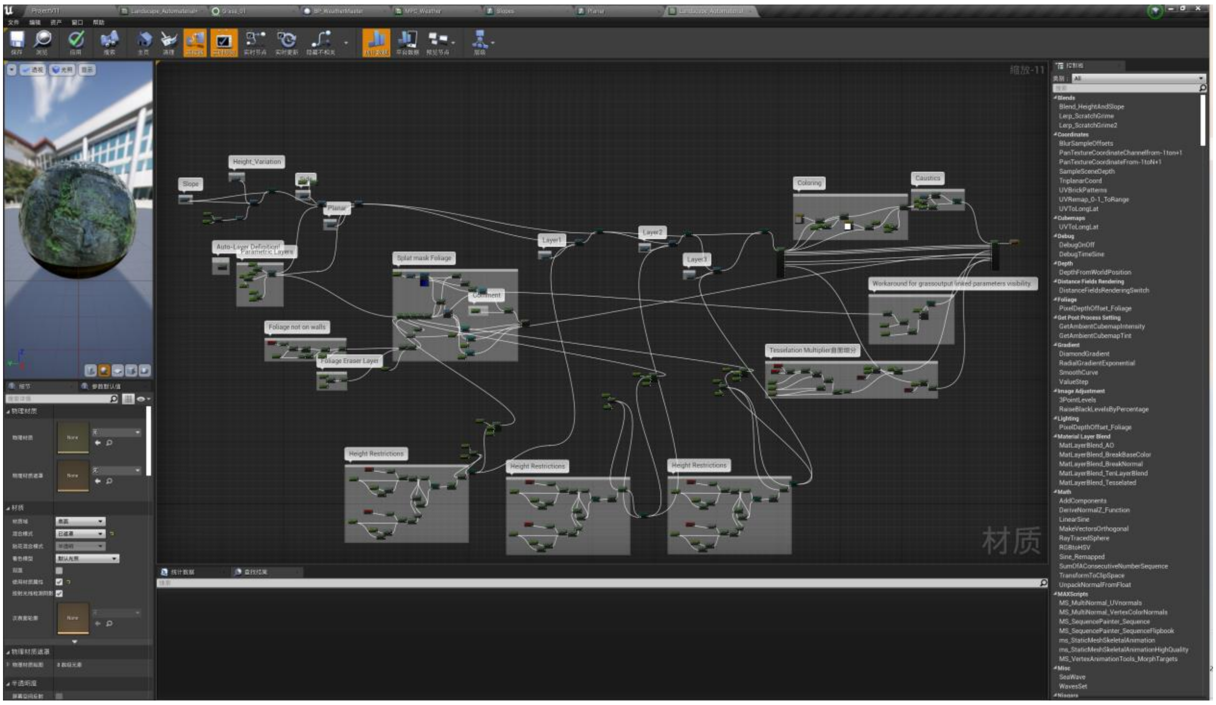


Figure 3.4.2 Landscape Material Function panel in *White Birch*

Terrain material is the most difficult material to express in the Unreal Engine, and writing terrain material functions is the focus of this research. When the character is active in the game scene, it is surrounded by terrain materials all the time. If the terrain material cannot express the realism of the next-generation game scene, it will be detrimental to the atmosphere of the entire game scene. The difficulty in making is that, unlike ordinary materials, terrain materials are often mixed materials composed of multiple materials. To obtain a multi-layer mixed material function, it is necessary to write each layer of material functions separately, and finally mix them into the terrain function. When considering the performance effect of a single material, also consider the performance effect after the material is mixed.

The material functions used in this project are divided into the following three categories:

1. Material functions, corresponding to the materials on each terrain type, such as grass, cobblestones, and cliffs.
2. Functional functions (expressions) that represent functional relationships between material functions, such as tessellation functions, blending functions, and height limit functions
3. The material output node corresponds to the last output node of the function. This node exists in the package of each material function or at the end of the terrain material function.

Here, the planar (grass) layer is used as an example of the material function to introduce the creation process of the first step of the terrain material function.

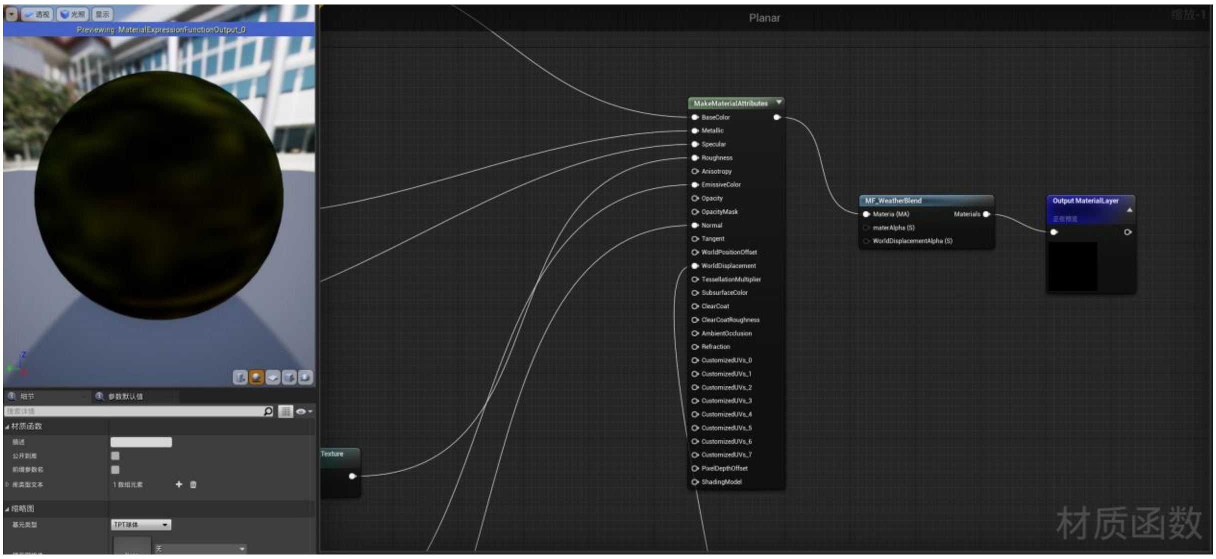


Figure 3.4.3 Planer Layer panel in *White Birch*

The Make Material Attributes panel shows sources that accept various attributes, which together make up the output of a material. BaseColor, Metallic, Specular, Roughness, EmissiveColor, Nomal, and WorldDisplacement are among the properties shown.

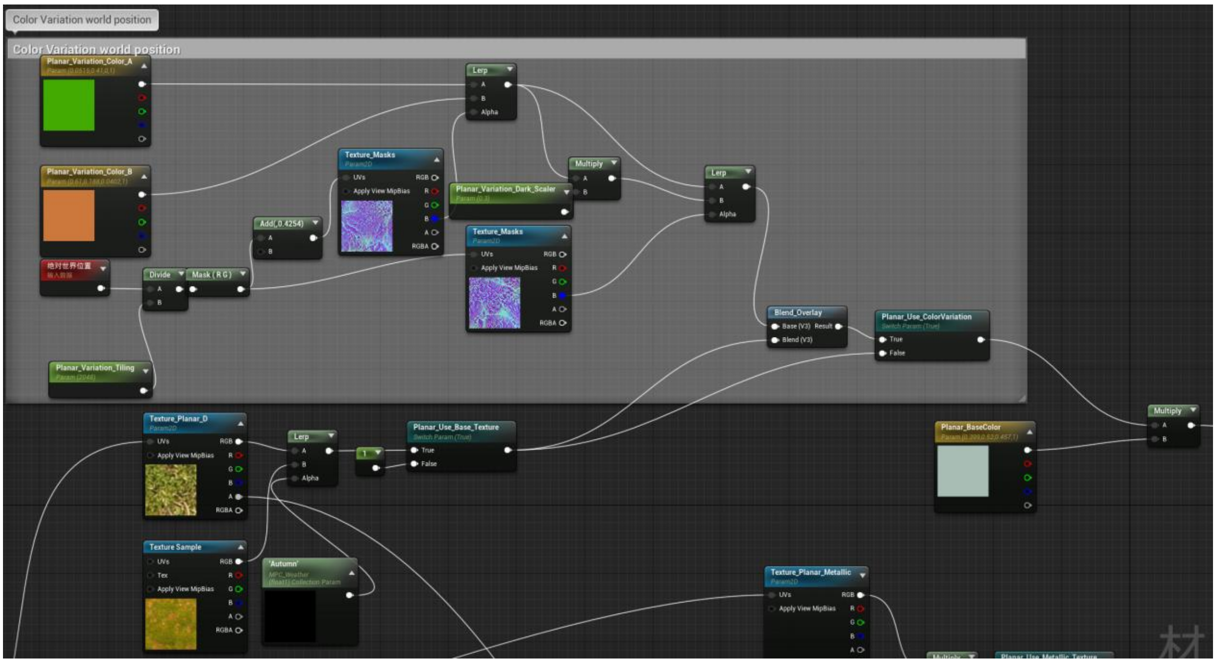


Figure 3.4.4 Planer Layer BaseColor part

Figure 3.4.4 shows the composition of the blueprint input to the Planer BaseColor node, including two texture samples, corresponding to the texture changes of autumn and non-autumn grass. (The one in the annotation panel is the WorldPositionOffset of the grass texture, which controls the dynamic snow).

All properties except BaseColor can be adjusted by using parameter functions to form adjustable variables. The following are screenshots of Metallic, Specular, Roughness, EmissiveColor, Nomal and WorldDisplacement properties.For space reasons, the production methods of the Slope and Side layers are the same. They will not be repeated here.

After obtaining the three-layer material, use the Blend function to blend it, and the result is output to the Auto layer. The Auto layer is responsible for controlling the mixing strength and action angle between layers. For example, grass is generated on the ground with an inclination angle of less than 10 degrees, and cobblestones are generated when the slope angle is greater than 10 degrees but less than 30 degrees. And finally, output to the erosion layer.

Finally, the erosion effect is added to the material. The erosion effect in this study is the rain erosion effect, which refers to the use of noise sampling to add noise simulation to the material layer. Erosion beautification is the basic function of terrain materials. Erosion will change the performance of the normal of the material layer, and will also change the weight distribution of the surface material layer. The erosion effect in this study acts on the normal height of the material layer and does not act on the terrain elevation map.

Following completing the production of the terrain material, the following is the drawing of the terrain material. First of all, it is necessary to create a new material instance for the material, put the corresponding texture into the corresponding material layer, adjust the parameters, and clearly separate the multiple layers at the visual level. This process is very time-consuming and energy-consuming. Simpler materials are more convenient in the drawing process, such as desert terrain, than terrains with rich layers, such as multi-material terrains that combine elements such as rocks, land, grass, snow, etc. A relatively high requirement, that is, visually, it is necessary to clearly display the various material layers, and to reasonably transition the connection between the material layers, so that the terrain material has a sense of integrity.

3.4.5 Introduction to the Parameters of the Terrain Material Function

<table>
<tr><td>Name of the Layer</td><td>Description</td></tr>
<tr><td>Planar</td><td>The texture responsible for generating grass, which is generated on flat land with a slope of fewer than 10 degrees. There is blending with the Slope layer between 10 and 15 degrees. It is also responsible for the automatic generation of vegetation and rocks (based on Landscape Grass Type).

Important parameters that can be adjusted:
<table>
<tr><th>Parameter</th><th>Effect</th></tr>
<tr><td>Texture</td><td>Specify the corresponding Texture to be placed on this layer (Albedo, Normal and Roughness)</td></tr>
<tr><td>BaseColor</td><td>Specifies the base color for this layer. Used to input the initial material blend function.</td></tr>
<tr><td>Metallic</td><td>Specifies the degree of metallization of the layer, the larger the value, the closer the texture is to the metal.</td></tr>
<tr><td>Roughness</td><td>Specifies the roughness of the layer. The larger the value, the rougher the texture and the less reflected light.</td></tr>
<tr><td>Specular</td><td>Specifies the specular reflection strength of this layer. The larger the value, the greater the specular reflection strength.</td></tr>
<tr><td>UV_Scale</td><td>Specifies the UV scaling degree of the texture of this layer. The larger the value, the smaller the image of the layer will be, and it will be tiled in the entire terrain; otherwise, the smaller the value, the larger the image of the layer will be. Used to adjust the degree of material repetition</td></tr>
</table>
</td></tr>
</table>

Slope	The texture responsible for generating cobblestones, which is generated on slopes with a slope greater than 10 degrees and less than 30 degrees and is mixed with the Side layer between 30 degrees and 43 degrees. There is blending with the Slope layer between 10 and 15 degrees. (It has the same type of parameters as in Planar)
Side	The texture responsible for generating the cliff, which is generated on slopes with a slope greater than 30 degrees, and blends with the Side layer between 30 degrees and 43 degrees. Reserved with an excuse (It has the same type of parameters as in Planar)
Foliage Eraser	Responsible for eliminating grass generated by LGT, this layer only supports manual elimination.
Caustics	The erosion layer is responsible for giving the rain erosion effect to the automatically generated material, which is closer to the real landform.
Auto	The Auto layer mixes all the above layers and sets several parameters to adjust the automatic generation effect, including the mixing strength, mixing distance, generation slope, plant growth slope, plant growth on the wall, etc. Set two parameters Blend Bias and Blend sharpness to control the blend distance and blend strength. Add two Distance scales to control the material duplication at close and far distances. Added Perlin noise to materials to avoid grass map duplication issues at long distances.

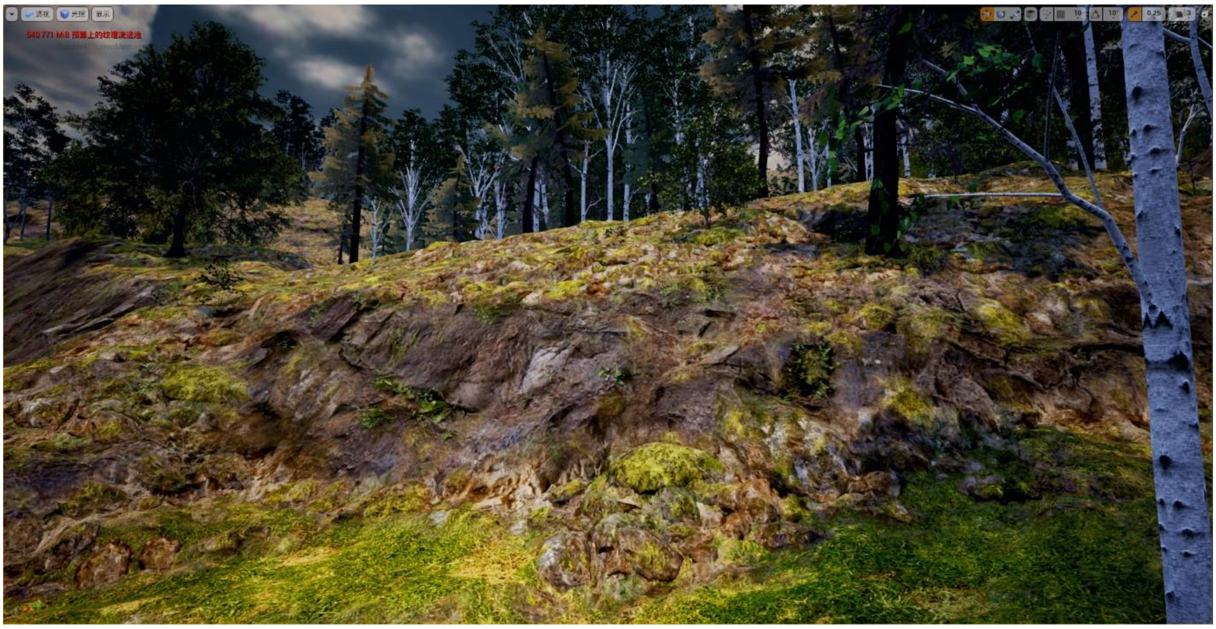


Figure 3.4.5 Terrain final blending effect illustration

3.5 Procedural Content Generation (PCG)

Procedural generation is a method of generating data automatically through algorithms rather than manually, combining human-generated assets and algorithms with the randomness and processing power of computer generation. In computer graphics, it is often used to create textures and 3D models. In video games, it is used to automatically create a lot of content within the game. Depending on the implementation, the advantages of procedural generation over manual generation may include smaller file sizes and larger amounts of content, but the biggest advantage of procedural generation is that it can save a lot of human resources and time.

The PCG algorithm used in this article is based on the built-in components of the Unreal Engine: Procedural Foliage Spawner and Landscape Grass Type, which realise the fully automated generation of plant assets.

3.5.1 Asset Classification

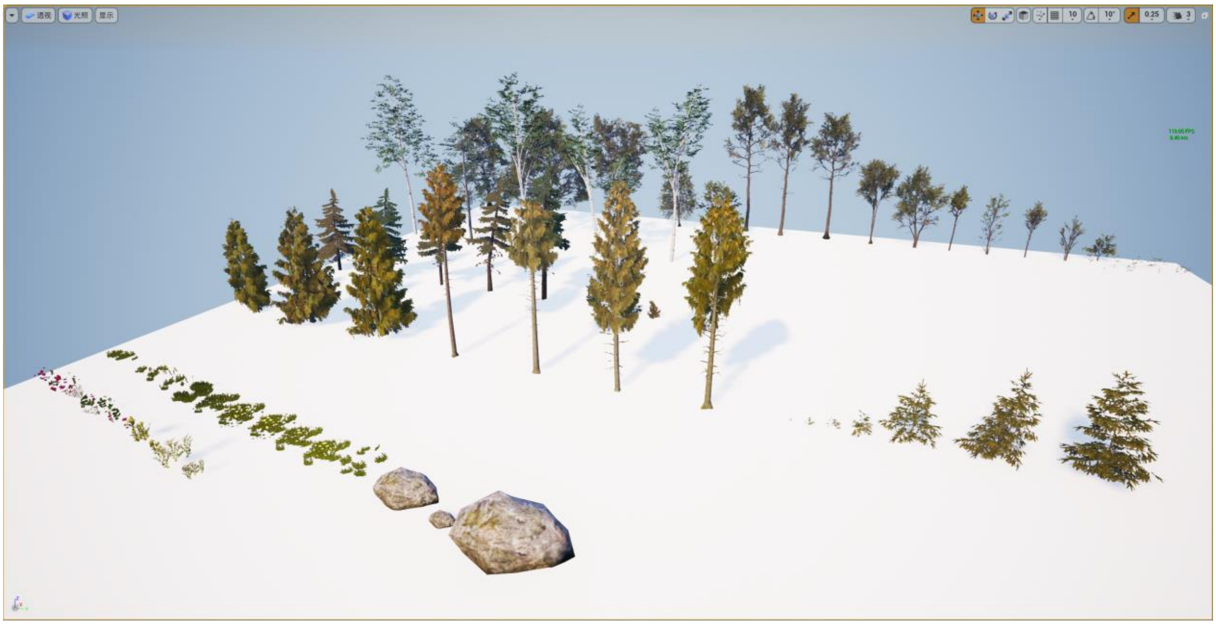


Figure 3.4.5 Terrain final blending effect illustration

Forest

22 kinds of Heikai trees (8 adult trees, 6 small trees, 6 seedlings, 2 sprouts)

5 species of birch

7 larch (5 adult trees, 3 seedlings)

14 spruce (7 mature trees, 7 seedlings)

Grass 7 Grass 5 Blossoms 3 Rocks

Shrubs 8 rose-hip 3 perilla

All assets are from the Unreal Megascan store

3.5.2 Procedural Generation of Plants

Selecting the corresponding PCG algorithm according to different types of assets has a greater contribution to the fidelity of the scene. PFS simulates the growth process of plants, from seeding to adulthood, and finally to the formation of plant clusters. It is worth noting that by adjusting the parameters of PFS reasonably, the characteristics of the horizontal distribution of tree clusters can be effectively reflected. This study uses PFS for the construction of the forest, classifying 48 mesh assets of four types of trees into three iterations of the PFS algorithm.

For the generation of shrubs and grasslands, LGT is used in this study, which is characterised by the ability to reasonably control the density and the degree of cluster density, but lacks the simulation of growth and spread. Therefore, LGT is more suitable for assets such as grass and shrubs that are dense and cannot reflect horizontal distribution. Unlike PFS, LGT only works when placed on the terrain texture layer.

3.5.2.1 Procedural Foliage Spawner（PFS）

The Procedural Foliage Tool is a component that comes with the Unreal Engine 4. Using the Procedural Foliage Tool, you can create, set up, and mass-generate the trees that make up the entire forest. You can control how vegetation is placed in the level, as well as how it grows and walks, by adjusting various properties of foliage type objects.In addition, the Procedural Foliage Generator allows you to generate a variety of quilt types.

Use the PFS tool that comes with UE to add a PFS volume within the selected range, and the specified trees will be generated within the volume by the algorithm. To use PFS, the foliage type of each tree must be added to the PFS. Important parameters for each tree type in PFS are as follows:

Parameter Name	**Value**	**Effect**
AlignToNormal (Boolean)	False	Controls whether the plant is perpendicular to the growing surface.
Ground inclination angle (float)	Birch [0,15] Black Adler [21,25] Spruce [0,5] + [25,30] Pine [12,23]	Plant instances are only placed on slopes that form a special angle from vertical. Control where plants grow, control the degree of mixing between plants.
Number of steps (integer)	[5,8]	The number of times a species is given age and made to spread seeds
Initial seed density (float)	0.1	Number of seeds sown within 10*10 meters
Average Diffusion Distance (float)	[50,500]	Average distance between walk instances and their seeds
Diffusion variance (float)	[20,150]	Specifies the difference between the shot distance and the average
Step by step (integer)	[1,5]	The number of seeds the instance propagates in a single step of the simulation
Distribution seed (integer)	[50,500]	Determine initial placement of seeds

Can grow in shadows (Boolean)	True	Seeds of this type ignore shadow radius when spawned with other plant types
Maximum initial age (float)	[0,1]	Allow new seeds to be created with age greater than 0
Max age (float)	[50,500]	Specifies the maximum age of a seed, after which the instance can continue to scatter seeds, but cannot grow any more.

3.5.2.2 Landscape Grass Type – LGT

The procedural grass tool is a component that comes with the Unreal Engine 4. By using the procedural grass tool, set the landscape grass type actor to the landscape terrain actor to fill the terrain with dense grass. After creating the LGT type, create an empty variable and add the grass mesh to this variable, then set various properties of this variable to adjust.

Table 3.5.2 Grassland Type Table

Property Name	Value	Additional Details
Grass Density	400.0	Because we want this to look like grass, we must spawn a lot of Static Meshes to make the Landscape appear densely covered in grass.
Use Grid	Enabled	To make the Static Meshes look more naturally placed, this offset is their placement position.
Random Rotation	Enabled	Giving the Static Meshes used for the plants and grasses a random rotation makes sure that the same side of the Static Mesh used is not seen all the time, adding to the visual variety of the scene.
Align to Surface	Enabled	This makes sure that the Static Mesh used conforms to the surface of the Landscape terrain.

Then use the Landscape Grass Output component to bind the corresponding terrain material layer, which in this case is the Planer layer, to output the grass.

3.6 Weather System

The weather system is one of the focuses of this research, and adding the interaction effects of weather changes and the environment can greatly increase the authenticity of the environment. The weather system is a global blueprint class that holds all the environmental effects in the scene that are automatically triggered by blueprints, such as dynamic snow, seasonality, and day and night cycles.

The weather system includes a collection of several functions that control the behaviour of different actors in the scene, which together constitute the weather system. For example, the direction and angle of sunlight control day and night; the intensity of sunlight, the colour of clouds, and the colour of the atmosphere together control sunny and dark nights. He has the following features: Day-Night cycle, random thunder, rain, snow, water, ice.

- User based Weather starting randomization
- Curves allowing crossfading weather patterns for minutes, hours or endless
- Particles collide with Scene (all), absorb light (all) & blend background color (snow).

Weather	**Sunlight Intensity**	**Volume Cloud**	**Sound Effect**	**Particle Effects FX**
Sunny	Warming	Thinned.	Birdsong and wind	Bloom
Cloudy	Weakened	Thicker and moves faster.	Wind	-
Rain	Weakened	Thicker and darken in colour	Wind and Rain	Water drop effects and water drop splash effects
Snow	Remains Unchanged	Remains unchanged	Wind and Snow	Falling snow effects and snow effects

In the weather system, the properties of each Actor can be adjusted to achieve the desired effect.

Core: Weather Random Time, Always Rain, Always Snow, Always Sunny, Always Day Time, Always Night, Dark Clouds

Snow: Max Snow Falling, Snow Blend Max, D Snow Color Of Day, D Snow Color Of Night

Rain: Rain Color Of Day, Rain Color Of Night, Max Rain Falling, Max Rain Fog Falling, Rain Drops Blend Max, Get Wet Roughness, Get Wet Specular, Thunder Waiting Time, Thunder Min Volume, Thunder Max Volume

Sun: Sun Intensity, Sun Intensity Of Rain or Snow. Sun Temperature of Daytime, Sun Temperature of Night

Fog: Fog Density Of Sunny, Fog Density Of Rain or Snow, Fog Height Falloff Of Sunny, Fog Height Falloff Rain or Snow

Wind: Wind Speed Of Snow or Rain, Wind Speed Of Sunny, Wind Weight Of Snow or Rain, Wind Weight Of Sunny

SkyLight: Sky Light Intensity Of Day, Sky Light Intensity Of Night

3.6.1 Dynamic Snow

The ***dynamic snow*** here refers to the construction of the interactivity between the snow effect and the snow. The snow effect also includes the interaction with the scene vegetation assets. For details, please refer to 3.7.2: Vegetation Snow Effect.

The dynamic snow in this study is divided into two parts, which are the realization of the surface snow effect and the interaction between the characters and the snow.

The principle of surface snow is to mix a layer of snow material on the planer layer in advance. The snow material changes the WorldPositionOffset of the planer material layer to achieve the snow effect. The material is controlled by the parameter set MPC_Weather, and the gradual accumulation is realized under the control of the weather blueprint. snow effect The effect is that as the weather timeline turns to winter, the grass layer gradually turns white and rises, finally forming a snow effect.

This study is known as the ***Trail System***. The principle is to set the acquisition range within a certain range around the characters in the scene, and bind them to a certain material layer, and at the same time, bind the collectors on the characters. When the character moves in the scene, the collector collects the movement track of the character and sends it to the collection range. The collection range is judged according to the track, and the material layer is modified according to the track. The performance is that when the character walks, the snow material layer will leave the movement track of the character behind.

3.6.2 Seasonality

The realization of the seasonal effect of the scene depends on the blueprint timeline system. The seasons alternate normally in the timeline system, and the four seasons are expressed as spring, summer, autumn, and winter. The duration of the seasons is in line with the seasonal characteristics of the cold temperate zone in northern China, with long and cold winters and short and sunny summers. The effects of the changing seasons on the environment vary in performance.

The scene weather system is independent of the time axis. The change of scene weather is a random blueprint event, and the weather sequence obtained by each simulation may be inconsistent. The weather is generally divided into three types: sunny, rainy, and cloudy. These weathers have different scene component settings, such as post-processing, lighting effects, and volumetric fog. The transition of weather is affected by seasonal changes, such as no rain in winter, no snow in other seasons, etc.

Spring: The sun is full, the weather is sunny, rainy and cloudy.

Summer: sunshine and more prone to rain.

Autumn: The leaves of the scene plants change color, and the color temperature of the overall scene increases.

Winter: Add a snow effect to the scene, including two stages, which are the snow accumulation process of the scene and the snow melting process of the scene.

3.6.3 Circadian cycle

The day and night cycle of the scene is independent of the seasonal change timeline and has a separate timeline control. The most important thing in the day-night cycle change of the scene is to adjust the lighting effect of the scene and render the scene's atmosphere of day and night. This includes setting volumetric fog, sun light intensity, sky light intensity, color temperature, and DOF adjustments.

3.7 Game Scene Construction

3.7.1 Steps to Build The Scene

flow chart:

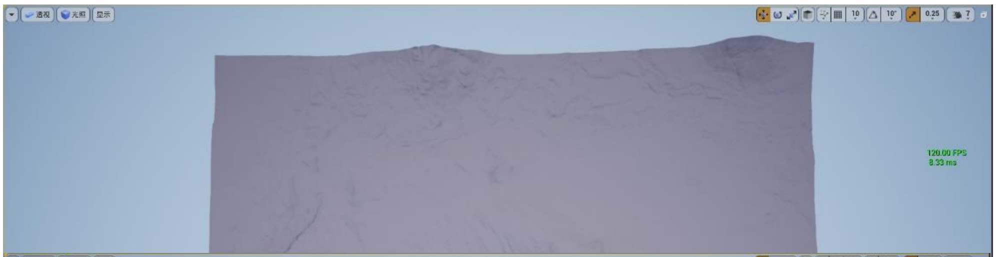
1. New map, remove all light sources and add weather system.

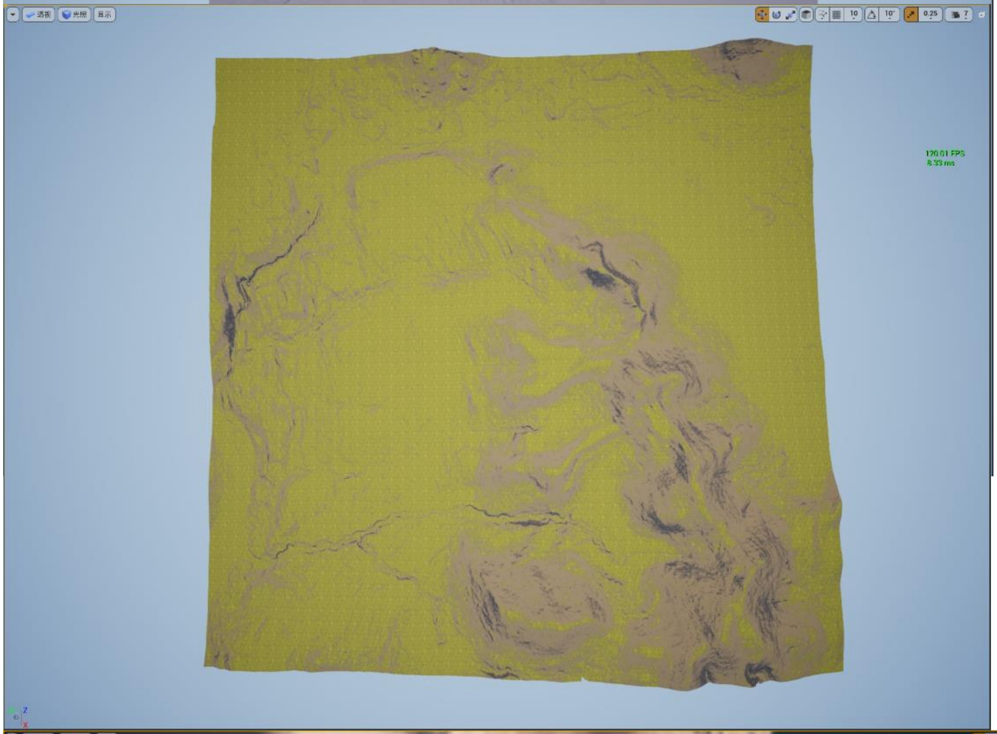
2. Create a new terrain, select the appropriate size and import the terrain elevation map

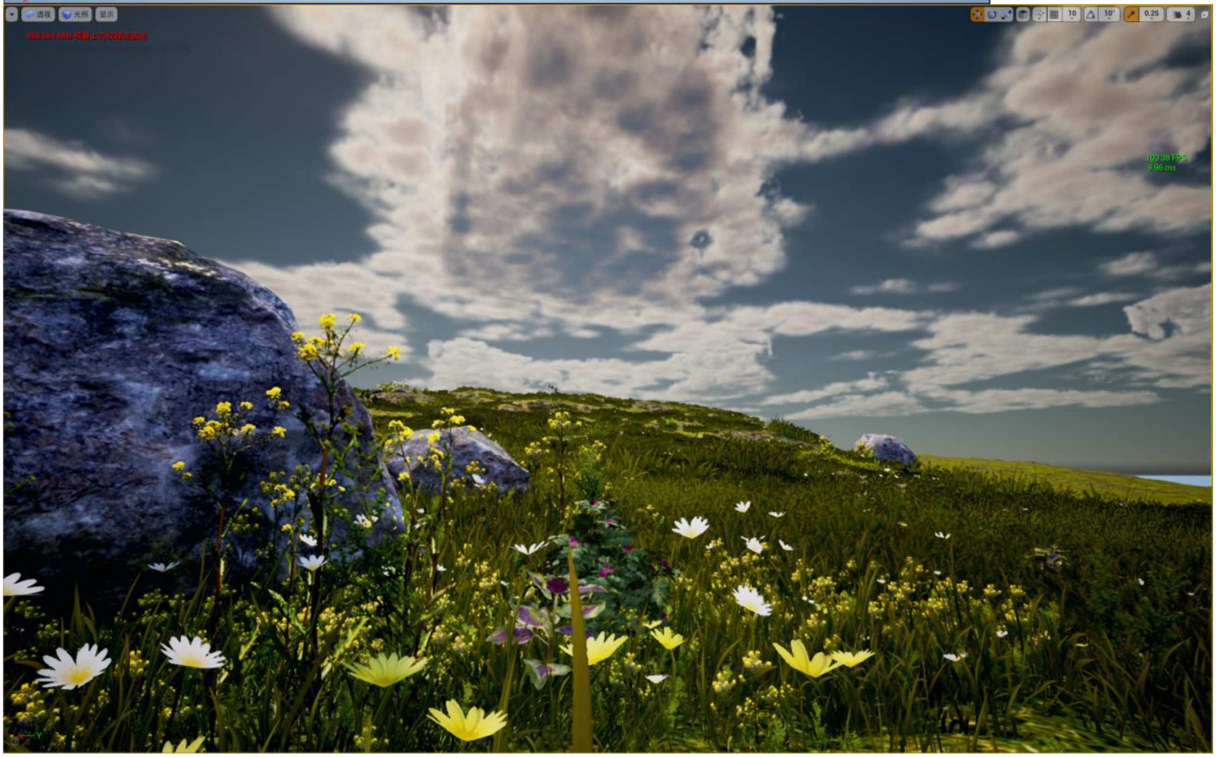
3. Bind the terrain material instance to the terrain elevation map and wait for the material to load.

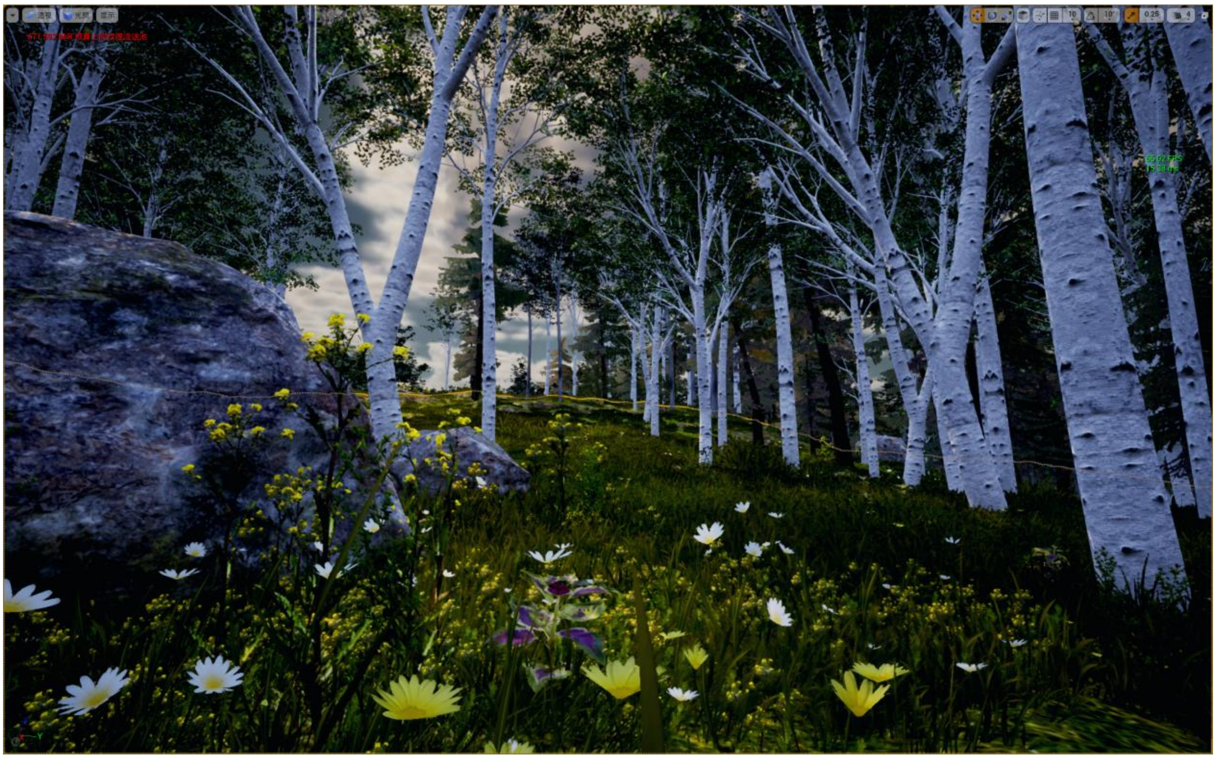
4. Add three corresponding PFS components, click Simulate Generate respectively, and wait for the loading to complete.

5. Done.


透视
光照
显示
10
10°
0.25
7
120.00 FPS
8.33 ms


透视
光照
显示
10
10°
0.25
7
120.01 FPS
8.33 ms
Z
X

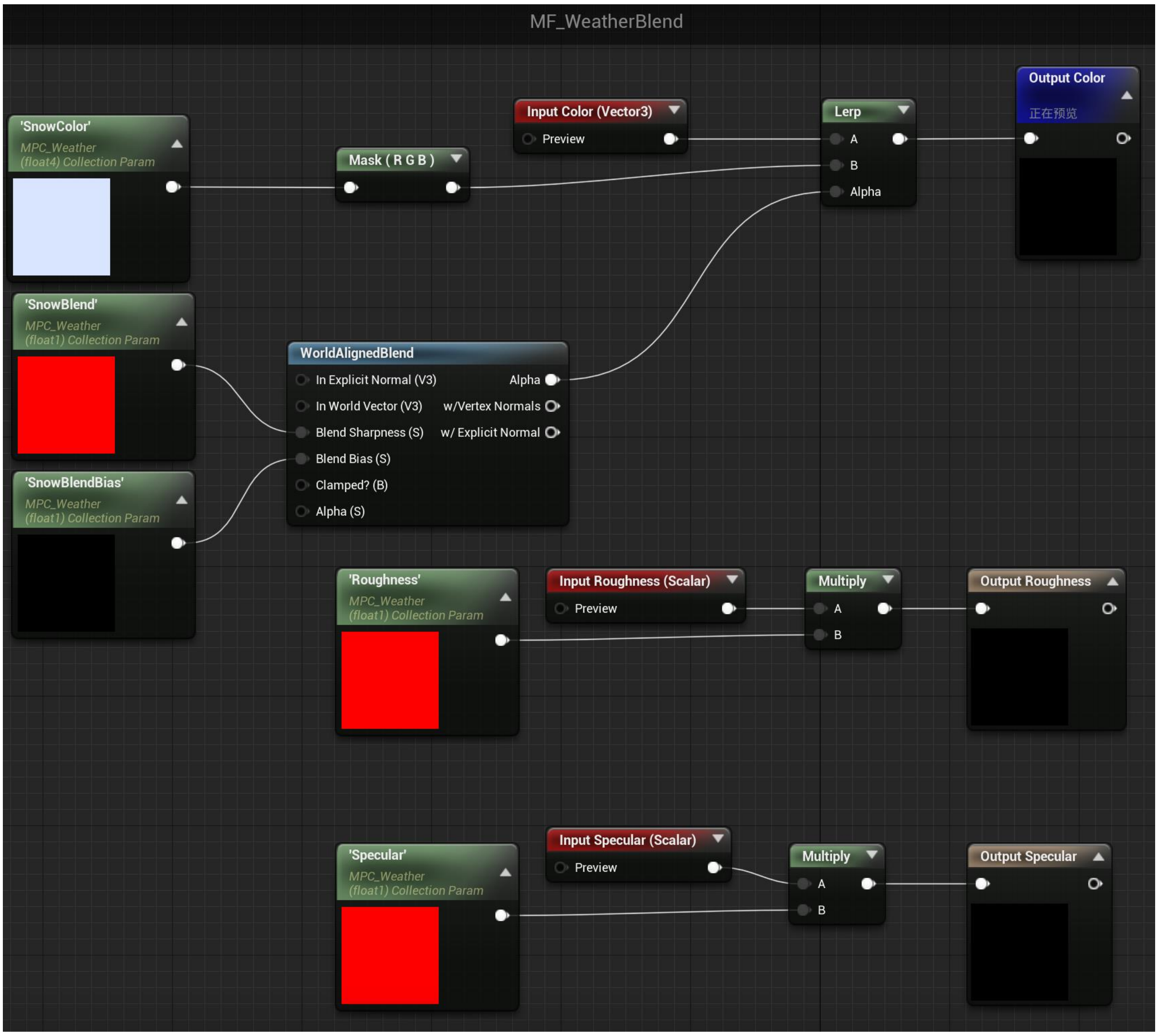

透视
光照
显示
10
10°
0.25
4



3.7.2 Vegetation Snow Effect

This project mainly studies boreal landforms, so one of the key points of the project is the production of vegetation snow effects. Until the mid-term, making snow effects was mainly about adding falling snow FXs, sound effects, and snow effects. The Snow Effect is a complex blueprint function that contains a child function and a mother function. The mother function controls parameters such as the snowfall curve and controls the occurrence of the entire snowfall behavior; at the same time, the child function needs to be linked to the output nodes of each material in advance to achieve the effect of controlling the color (grass material, leaf material, cobblestone material, etc.).

I used the WorldAlignedBlend method to blend white directly onto the material's surface to create a snow effect. Most vegetation materials can be used with this method. This type of map saves a lot of computing resources compared to applying the 3D snow effect to the plant material, which is beneficial to the scene's fluency.



3.7.3 Scene Lighting and Postprocessing

3.7.3.1 Lighting Effect

Atmospheric effects: light, volumetric clouds, volumetric fog, particle effects, and lighting effects.

Creating a good atmospheric effect can add beauty to the scene, and lighting constitutes the most important part of the scene design. Several atmospheric effects were used in the scene to add the unique climate effect of the Greater Khingan Mountains.

Light Source	**Effect**
Directional Light	A directional light source simulates light coming from an infinite source. The shadows cast by this light source are parallel, so it is suitable for simulating sunlight.
Sky light	Sky light collects the distant part of the level and applies it to the scene as a light source. Even if the sky comes from the atmosphere, clouds at the top of the sky box or distant mountains, the appearance of the sky and its illumination / reflection will match
Volumetric Cloud	Using a two-dimensional volume cloud map, a volume cloud is

	generated by connecting Perlin noise nodes.
Exponential Height Fog	The density of exponential height fog is higher at the lower position on the map, while the density is lower at the higher position. Its transition is very smooth, and there will be no obvious switching with the increase of altitude. Can highlight the altitude effect.
Niagara FX	Use niagarafx to write special effects of rain and snow.
Post Process Component	Add lens flares, add lens effects, add volumetric effects, and add lens effects bloom and exposure

3.7.3.2 Post Processing

Post-processing is often used for 3D rendering, especially in video games. Instead of rendering 3D objects directly to the display, the scene is first rendered to a buffer in video memory. Post-processing filters are then applied to the image buffer using pixel shaders and various vertex shaders before displaying it on the screen. Some post-processing effects also require multipass, gamma input, vertex manipulation, and depth buffer access. The use of post-processing necessitates understanding of the overall effect of the image (as each 3D object is typically rendered individually).Such effects include:

- Ambient Occlusion (HBAO, Screen Space Ambient Occlusion (SSAO), Reflections, and so on)
- Antialiasing (FXAA, AGAA, SMAA, MLAA, and custom antialiasing methods-not sample-sized AA like MSAA and SSAA)
- Bloom
- Curves of Color
- DOF (Depth of Field)

Post-processing can easily create global effects, which play a great role in creating a scene's atmosphere. For example, in

In this study, a variety of post-processing methods were used, including XXX, of which the most important

3.8 Scene Optimization

Scene performance optimization: increase the smoothness.

Scene performance optimization is the top priority of the entire project, and it is also the most difficult part of the project. In UE5, using a lot of expensive plant assets in the same scene will make the scene's FPS (frames per second) very low. Use several effective optimization methods to stabilize the frame rate above 30fps to ensure the basic scene browsing experience is the project pursuit. Since this project needs to use a large number of plant assets, if the scene performance optimization is not set, the scene fluency will be poor. After starting to optimize the scene, the overall performance optimization of the scene progresses.

3.8.1 Level of Detail (LOD)

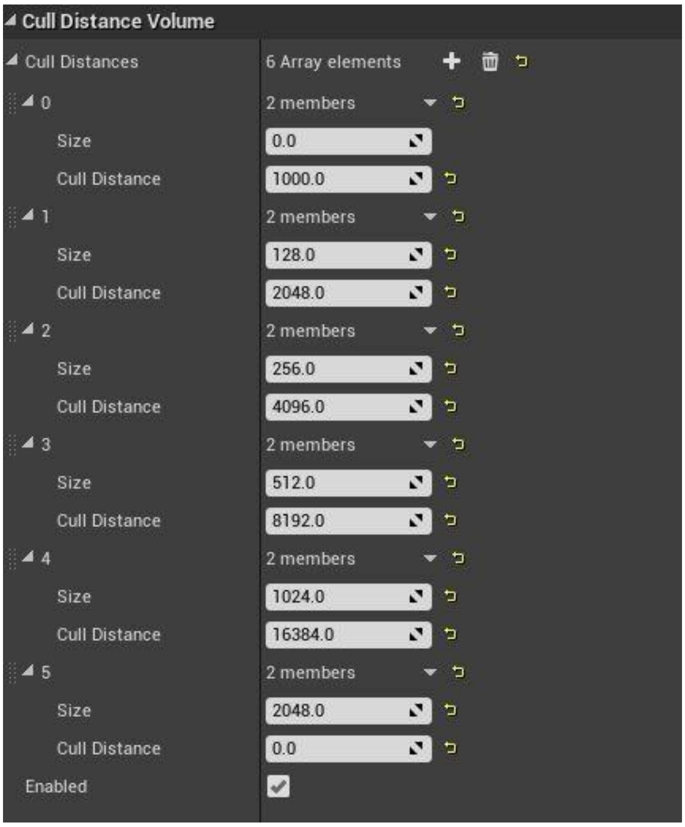
Set LOD for plant materials that appear in a large number of scenes. Set the position and importance of the nodes of the plant asset model in the display environment, determine the resource allocation for object rendering, and reduce non-important objects. To obtain efficient rendering operations, the number of faces and the degree of detail must be high. Take the Black Alder plant asset, as an example:

LOD Layers	Screen Size	Triangles	Vertices
LOD 0	(0.99,5]	423,509	439,689
LOD 1	(0.8,0.99]	203,065	259,446
LOD 2	(0.4,0.6]	143,457	204,793
LOD 3	(0.3,0.4]	50,491	59,268
LOD 4	(0.15,0.3]	28,020	37,956
LOD 5	(0.15,0.3]	7,111	9,054
LOD 6	(0,0.15]	9	9

It can be seen that as the proportion of objects on the screen decreases, the number of LOD layers increases, and the number of triangles and vertices decreases rapidly. Scene computing resources are optimized. After adding LODs, the FPS of the same scene was significantly improved.

3.8.2 Culling Distance

Set the mesh culling distance (Cull Distance):



Cull Distance Volume

Cull Distances 6 Array elements

0 2 members
Size 0.0
Cull Distance 1000.0

1 2 members
Size 128.0
Cull Distance 2048.0

2 2 members
Size 256.0
Cull Distance 4096.0

3 2 members
Size 512.0
Cull Distance 8192.0

4 2 members
Size 1024.0
Cull Distance 16384.0

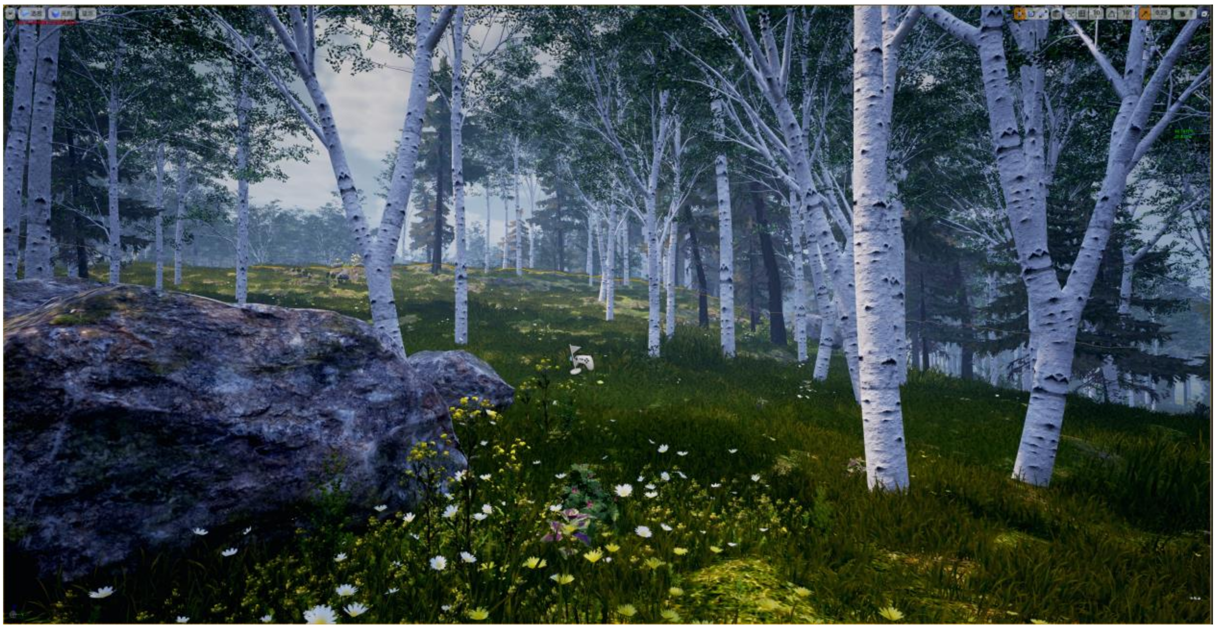
5 2 members
Size 2048.0
Cull Distance 0.0

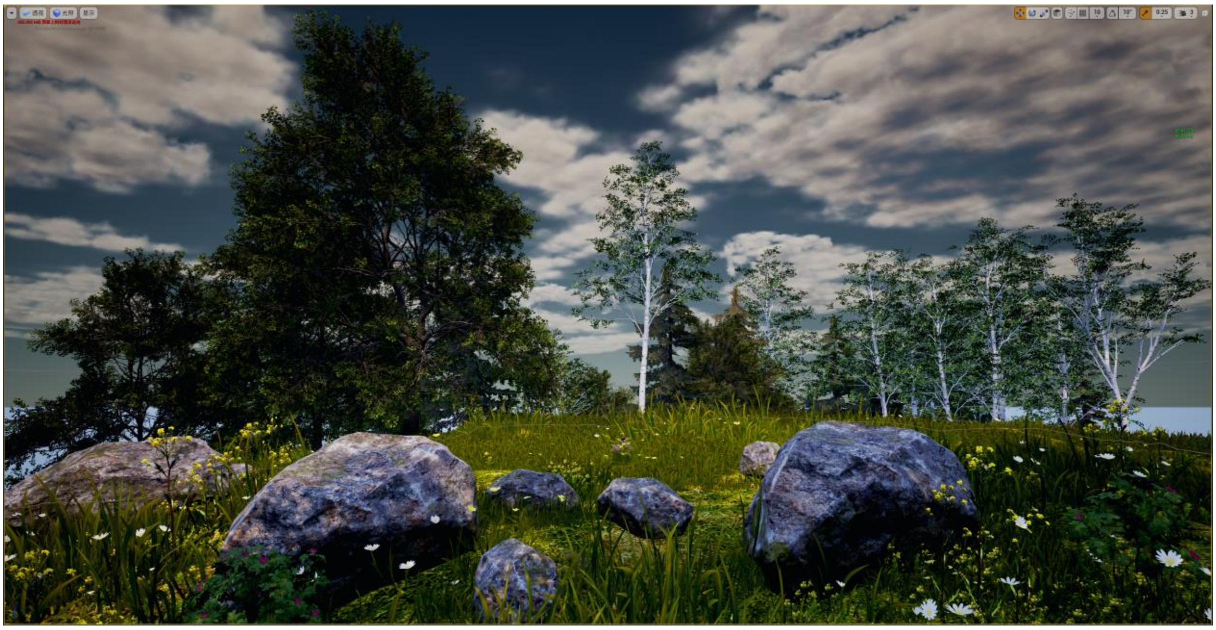
Enabled ☑

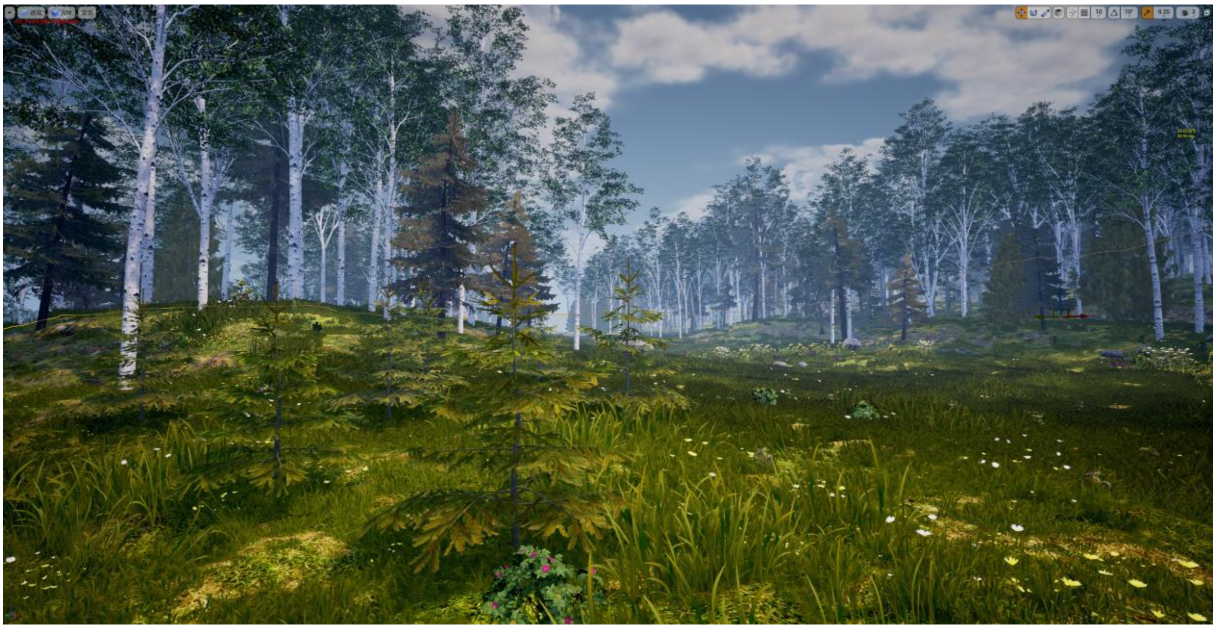
When the camera is sufficiently removed from a mesh component, the mesh component can be deemed uninteresting and hence culled. Mesh components are automatically culled from the scene after a specified distance from the camera when the cull distance is set, saving computational efficiency. The culling distance can usually be set to two values in LGT and PFS: the start culling distance and the end culling distance. When the mesh hits the start culling distance, it begins to gradually cull and then totally culls when it reaches the end culling distance, resulting in a smooth scene gradien

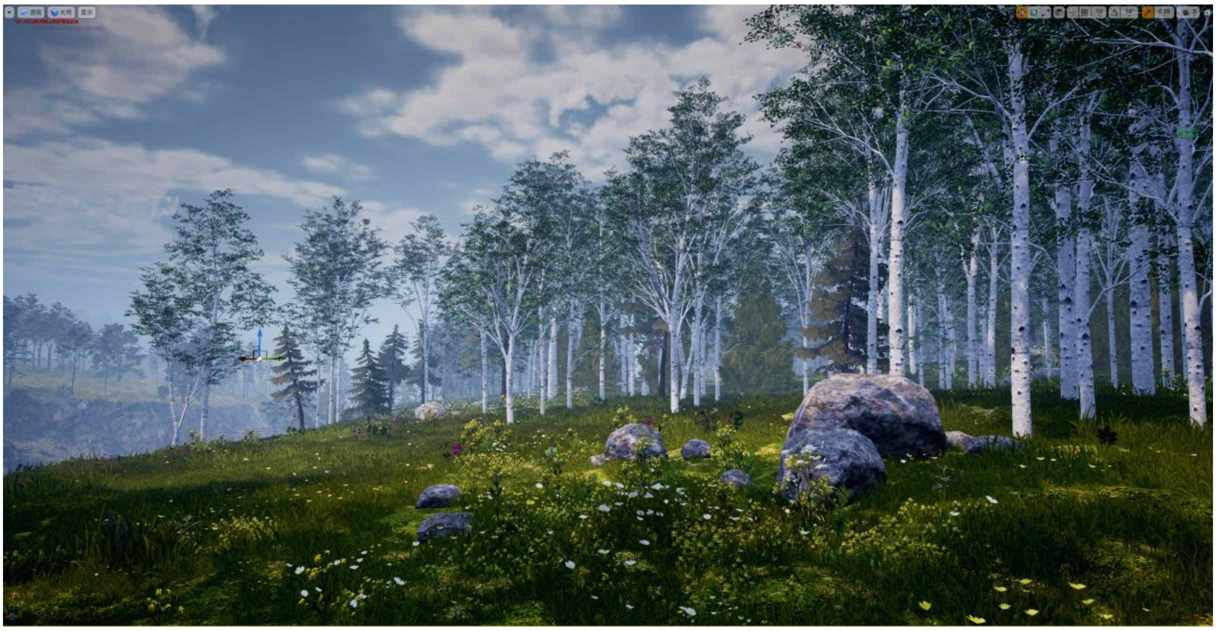
Chapter 4: Results and Discussion

4.1 Results









4.2 Highlight

4.2.1 Seasonality

4.2.2 Interactive Snow

4.3 Performance Evaluation

Frame rate complexity analysis

Shader Complexity Analysis

Chapter 5: Conclusion and Further Work

5.1 Conclusion

5.2 Innovation

The implementation scheme is based on the ProceduraLFoliageSpawner that comes with the native UE, and some modifications have been made. Compared with the current more popular schemes, such as Houdini, it is relatively simple as a whole. Its advantages are that

- Based on the native UE with minimal changes
- Vegetation generation rules are simple to modify and extend.
- Streamable, making it ideal for open worlds
- In addition, we've created some tools for artists that can be generated with a single click once the configuration is complete and can be iterated relatively quickly.

5.3 Further Work

Runtime virtual texture

Increase running frame rate

References

[1]谢飏霄. Unreal Engine 下的次世代游戏场景设计研究[D].浙江工业大学,2018.

[2]游艺网教育部．网络游戏角色制作[M]．北京：清华大学出版社，2015.

[3]杨为一.美工系列讲座第 4 讲——3D 游戏制作之自然场景（一）[J].程序员：游戏创造,2005(6):68-73

[4]叶津.复杂地貌园林地形景观叠加区域三维仿真[J].计算机仿真,2019,36(11):207-210.

[5]Unreal Engine, https://www.unrealengine.com/zh-CN/featuresUnreal Engine 产品介绍,2022.4.27

[6]李博,马晓阳,李真玉.我国东北寒温带景观区地球化学特征[J].黑龙江自然资源,2021(2):26-27

[7]朱道光,曾昭文,王继丰,崔福星,倪红伟.中国寒温带针叶林生物多样性的研究进展[J].国土与自然资源研究,2014(1):90-92

[8]周以良. 中国大兴安岭植被[M]. 科学出版社, 1991.

[9]赵麽.探索次世代游戏场景设计当中的光影运用与表现[J.才智,2013(30):336.

[10]谢振光.内蒙古大兴安岭北部原始林区植物区系及区系地理成分研究[J].林业资源管理,2017,0(2):88-92

[11]高占军.内蒙古大兴安岭北部原始林区植物区系组成及主要植被群系特征分析[J].内蒙古林业调查设计,2020,43(2):65-68

Acknowledgement

On the occasion of the completion of my undergraduate dissertation, looking back on the four years of study at Beijing University of Posts and Telecommunications has brought me a rich and profound learning and life experience, which will become my future, whether in professional exploration or in life. The most precious spiritual wealth Here, I would like to express my heartfelt thanks to the teachers, family members, and seniors who have helped and taught me.

I would like to thank my graduate tutor and my teacher for their teaching in the past five months. Under the guidance of my teacher, I am gradually understanding the research methods and research thinking of academic research. The teacher's serious and rigorous attitude towards the undergraduate program has benefited me a lot, and it has helped me a lot to complete the graduation program. When I have doubts, the teacher will always answer my questions. The teacher's rigorous thinking and logic, truth-seeking and pragmatic research attitude, and unique style of innovation are worthy of my lifelong learning.

Thanks to an upmaster I met on the BiliBili platform, 宇宙丨YZ who lent a helping hand when I encountered technical difficulties in my graduation project. It was his encouragement and support that allowed me to successfully overcome the weather system. Problems with interactive snow fusion

Thanks to my friend Liu Zihao, who gave me technical support during the completion of the project and answered some of my questions, including but not limited to the vegetation lighting model.

Thanks to Chen Longxiang and Qi Hangjiang, during our senior year, we got together with the same vision for the game industry and had a wonderful time.

Thanks to my family for their encouragement, support, and meticulous care, I have successfully completed my four-year undergraduate study journey. I will continue to move forward with their expectations of me in the future.

At the same time, I would like to thank Mr. Ling Ma and Mr. XX for giving me a lot of directional guidance, whether it is teaching and preaching on weekdays, or guiding in the opening and defense of the thesis.

Finally, I would like to thank my classmates who have spent four years at Beijing University of Posts and Telecommunications for providing very useful opinions for my graduation project and graduation thesis, so that I can successfully complete the production of the graduation project and the writing of the graduation thesis.

I dedicate this document to all those who helped me, supported me, and cared about me.

Appendix

Risk and environmental impact assessment