



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ

Робототехника и комплексная автоматизация (РК)

КАФЕДРА

Системы автоматизированного проектирования (РК6)

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОМУ ПРОЕКТУ
НА ТЕМУ:
«Визуализация природных ландшафтов и средневекового
поселения»

Студент РК6-74Б

(Подпись, дата)

Колыхалов Д. В.

И.О. Фамилия

Руководитель КП

(Подпись, дата)

Витюков Ф.А.

И.О. Фамилия

2024 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой РК6
А.П. Карпенко

«_____» _____ 2024 г.

ЗАДАНИЕ
на выполнение курсового проекта

по дисциплине: Модели и методы анализа проектных решений

Студент группы РК6-74Б

Колыхалов Дмитрий Витальевич
(Фамилия, имя, отчество)

Направленность КП (учебная, исследовательская, практическая, производственная, др.) учебная
Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения КП: 25% к 6 нед., 50% к 11 нед., 75% к 14 нед., 100% к 16 нед.

Техническое задание: пользуясь различными техниками создания ландшафта в движке Unreal Engine 5, создать фотореалистичный пейзаж. Создать средневековое поселение. Создать NPC.

Оформление курсового проекта:

Расчетно-пояснительная записка на 41 листе формата А4.
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.):
4 графических листа

Дата выдачи задания «3» октября 2024 г.

Руководитель КП

(Подпись, дата)

Витюков Ф.А.
И.О. Фамилия

Студент

(Подпись, дата)

Колыхалов Д. В.
И.О. Фамилия

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре

АННОТАЦИЯ

В данной работе рассмотрены подходы создания реалистичного пейзажа местности, средневекового поселения и NPC. Рассмотрены принципы создания материалов, процедурной генерация биомов, водных объектов, кат-сцены и настройки освещения. Рассмотрены варианты оптимизации.

В расчетно-пояснительной записке 41 страница, 41 рисунок, 3 графических листа.

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	3
ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ.....	6
ВВЕДЕНИЕ.....	7
1. Ландшафт	8
1.1 Создание ландшафта	8
1.2 Импорт ландшафта	10
1.3 Инструменты редактирования ландшафта.....	11
1.3.1 Изменение карты высот – Sculpt.....	12
1.3.2 Возврат к нулевой карте высот – Erase	14
1.3.3 Сглаживание - Smooth.....	14
1.3.4 Выравнивание - Flatten.....	15
1.3.5 Спуск - Ramp	15
1.3.6 Термическая эрозия – Thermal Erosion.....	16
1.3.7 Гидроэрозия – Hydro Erosion	16
1.3.8 Noise	17
2. Материал ландшафта	17
3. Растительность	20
3.1 Painting.....	21
3.2 Placement	22
3.3 Instance Static.....	22
4. Procedural Content Generation	23
5. Средневековое поселение.....	27

6. Настройка неба.....	27
7. Водные объекты.....	28
8. Создание тумана.....	30
9. Настройка освещения	31
10. Оптимизация	31
11. Создание кат-сцены.....	33
12. Создание NPC	33
Результаты.....	37
ЗАКЛЮЧЕНИЕ.....	40
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	41

ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

UE5 – трехмерный движок Unreal Engine 5.

Референс – вспомогательное изображение, которое разработчик изучает перед работой, чтобы точнее передать детали, идеи, получить дополнительную информацию.

3D model – объемная фигура в пространстве, создаваемая в отдельной программе (Blender, 3DS MAX и др.). Для создания модели за основу принимаются чертежи, фотографии и описания модели, опираясь на которые, разработчик создает виртуальную модель.

Текстура – изображение, накладываемое на поверхность 3D модели. Содержит в себе различные свойства, например: цвет, жесткость, металличность, карту нормалей, смещение и др.

Материал – набор специальных настроек, описывающих свойства поверхности объекта, на который накладывается данный материал.

Полигон – базовый компонент, из которого состоит 3D сетка объекта. Полигоны бывают разных типов, например: треугольник, четырехугольник и фигуры с 5 или более вершинами. Наиболее распространение получили треугольно образные полигоны.

ВВЕДЕНИЕ

Unreal Engine — это игровой движок от Epic Games, одной из крупнейших американских компаний по разработке игр и программного обеспечения. И хотя инструмент предназначен в первую очередь для создания видеоигр, он подходит и для производства неигровых проектов в области архитектуры, строительства, автомобильной промышленности, медицины, кинематографа, анимации и других сфер. Редактор движка Unreal Editor создан по принципу «что пользователь видит, то и получит». Это означает, что итоговый результат не будет отличаться от его изображения в 3D-вьюпорте. К тому же редактор очень удобен для использования: все ассеты (модели, источники освещения, визуальные эффекты и так далее) можно сразу разместить в сцене, перетаскив из папок. В целом Unreal Editor можно назвать комплексной системой, состоящей из многочисленных редакторов, которая направлена на то, чтобы сделать процесс разработки максимально цельным.

Целью данной работы является создание реалистичного пейзажа, средневекового поселения и NPC.

Unreal Engine, с его постоянно развивающимися возможностями, предоставляет разработчикам мощные инструменты для достижения этого эффекта. Следует рассмотреть ключевые аспекты, такие как моделирование ландшафта, текстурирование, освещение, постобработка, использование новых возможностей Unreal Engine 5, программирование.

1. Ландшафт

1.1 Создание ландшафта

Для создания и настройки ландшафта в панели выбора следует выбрать Landscape mode. Панель выбора представлена на рисунке 1.

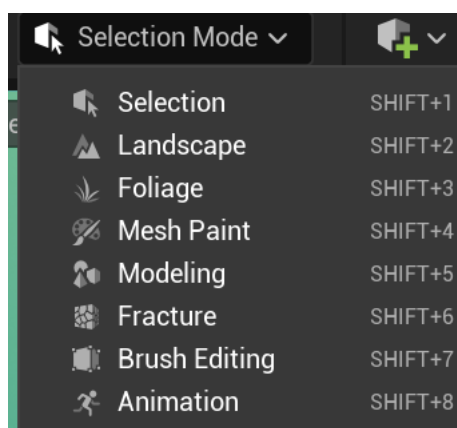


Рис. 1. Панель выбора

В Landscape mode для создания первоначального ландшафта, плоскости, следует выбрать инструмент Manage, что является управлением ландшафта и выбрать New. В инструменте находится две возможности создания ландшафта:

1. Create New – создание нового ландшафта с последующей настройкой рельефа при помощи различных инструментов.
2. Import from file – создание нового ландшафта с первоначально настроенной картой высот, которая может быть получена в сторонних программах.

Оба варианта предоставляет первоначальную настройку ландшафта.

1. Material – добавление первоначального материала ландшафта. Им может быть как однотонный материал одного цвета или автоматериал, который в зависимости от высоты рельефа окрашивает поверхность в той или иной цвет.
2. Location – положение созданного ландшафта относительно точки мира с координатами (0, 0, 0).
3. Rotation - поворот созданного ландшафта относительно точки мира с координатами (0, 0, 0)

4. Scale – размер ландшафта. Для варианта Create New размер ландшафта по z не имеет значения. Чаще всего Scale применяется при импортировании карты высот ландшафта.

5. Section Size – размер в квадратах одной секции. Увеличение размера секции применяется для получения большего контроля над созданным ландшафтом.

6. Number of Components – количество «областей/больших квадратов».

7. Overall Resolution – разрешение ландшафта. Для большей оптимизации используется разрешение 1009 x 1009.

Панель инструмента Manage представлена на рисунке 2. Окно просмотра получаемого ландшафта представлено на рисунке 3.

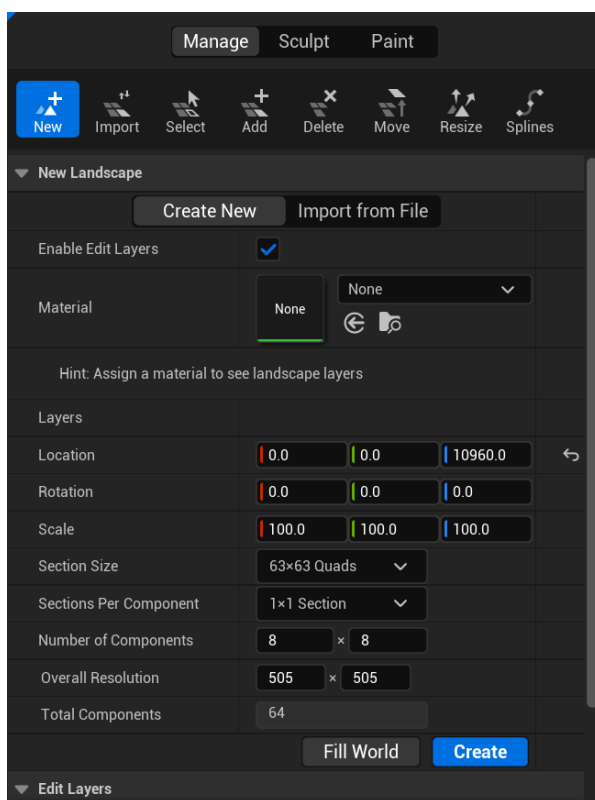


Рис. 2. Инструмент Manage

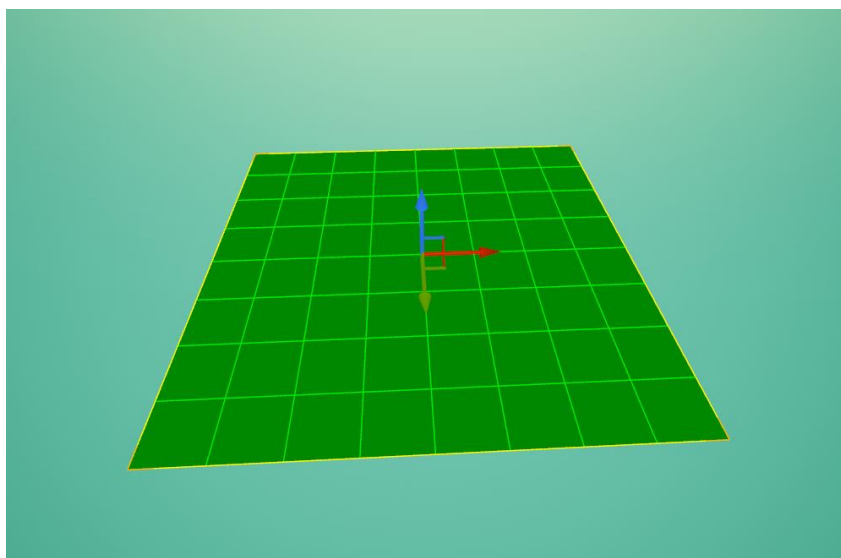


Рис. 3. Окно просмотра получаемого ландшафта

1.2 Импорт ландшафта

Для создания ландшафта удобным и эффективным методом создания ландшафта является импорт его из файла, который представляет собой карту высот, которая может быть получена при генерации ландшафта в сторонних программах, таких как GAEA, и последующая настройка.

Для импорта данной карты высот в инструменте Manage (рис. 2) следует выбрать import from file. Появившееся око представлено на рисунке 4.

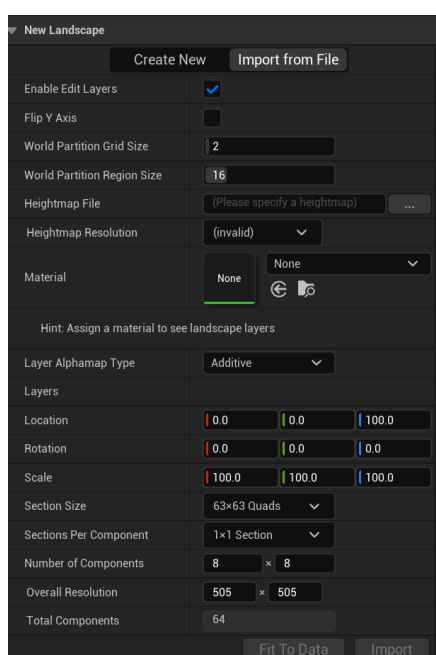


Рис. 4. Инструмент Manage при импорте карты высот

В поле Heightmap File следует выбрать полученную карту высот в сторонней программе. Важно учесть, что полученный ландшафт в Unreal Engine может отличаться от того, что был создан в GAEA. Для этого следует в поле Scale изменить соответствующие значения.

Для изменения Scale следует воспользоваться следующей формулой:

$$\frac{100}{X_{OR}} * X_{GS} = X_S,$$

где X_{OR} - X в Overall Resolution, X_{GS} – Scale, который использовался в GAEA, X_S – полученный Scale. В самом Unreal Engine метрика измеряется в units, 1 unit \approx 1 см. Таблица рекомендованной размерности ландшафта представлена на рисунке 5.

Overall size (vertices)	Quads / section	Sections / component	Component size	Total Components
8129×8129	127	4 (2×2)	254×254	1024 (32×32)
4033×4033	63	4 (2×2)	126×126	1024 (32×32)
2017×2017	63	4 (2×2)	126×126	256 (16×16)
1009×1009	63	4 (2×2)	126×126	64 (8×8)
1009×1009	63	1	63×63	256 (16×16)
505×505	63	4 (2×2)	126×126	16 (4×4)
505×505	63	1	63×63	64 (8×8)
253×253	63	4 (2×2)	126×126	4 (2×2)
253×253	63	1	63×63	16 (4×4)
127×127	63	4 (2×2)	126×126	1
127×127	63	1	63×63	4 (2×2)

Рис. 5. Рекомендованная размерность ландшафта

1.3 Инструменты редактирования ландшафта

Для редактирования созданного ландшафта движок Unreal Engine 5 имеет встроенный набор инструментов, которых находится в Landscape mode -> Sculpt. Инструмент Sculpt и его возможности приведены на рисунке 6.

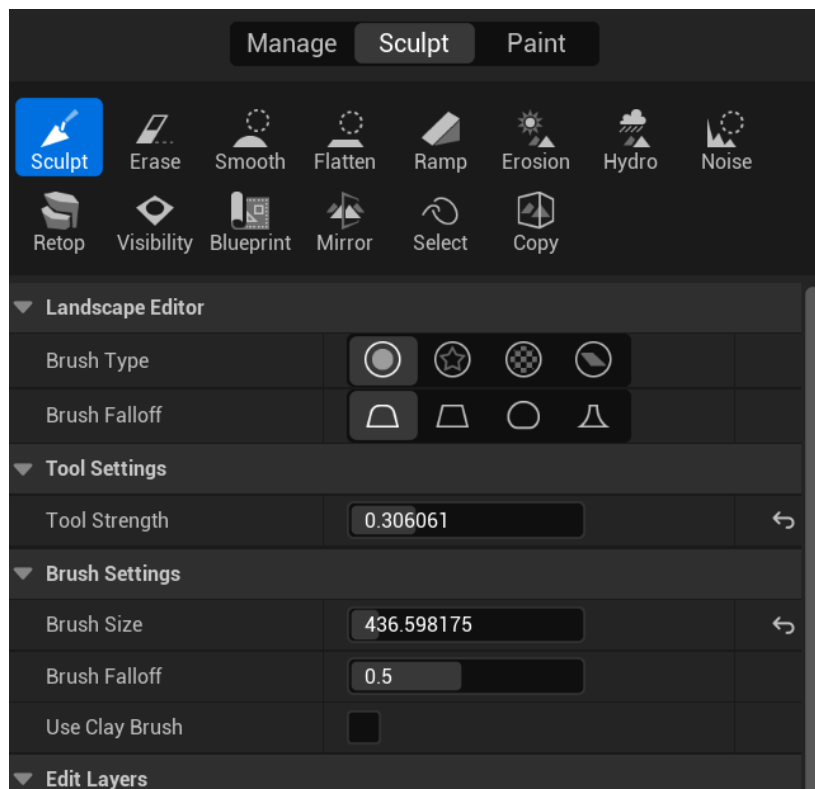


Рис. 6. Инструмент Sculpt

Для изменения ландшафта используется кисть, которая имеет свой ряд параметров, таких как тип, размер, сила, заполняемость и др.

Рассмотрим варианты изменения ландшафта, которые предлагает инструмент Sculpt.

1.3.1 Изменение карты высот – Sculpt

Для наращивания высоты используется простое нажатие по области левой кнопкой мыши. Для уменьшения высоты или для вдавливания вглубь используется сочетание клавиш Shift+ЛКМ.

Примеры работы инструмента представлены на рисунках 7, 8.



Рис. 7. Результат работы Sculpt



Рис. 8. Результат работы Sculpt

1.3.2 Возврат к нулевой карте высот – Erase

Результат работы инструмента Erase представлен на рисунке 9.



Рис. 9. Результат работы Erase

1.3.3 Сглаживание - Smooth

Инструмент Smooth используется для сглаживания карты высот, чтобы переходы между несколькими значениями не были резкими, или для накладывания слоев. Результат работы инструмента Smooth представлен на рисунке 10.



Рис. 10. Результат работы Smooth

1.3.4 Выравнивание - Flatten

Инструмент Flatten используется для выравнивания карты высот. Flatten выравнивает последующие высоты по значению первой высоты, т. е. откуда начиналось движение. Результат работы инструмента Flatten представлен на рисунке 11.



Рис. 11. Результат работы Flatten

1.3.5 Спуск - Ramp

Инструмент Ramp создает спуск между двумя вершинами.

Результат работы инструмента Ramp представлен на рисунке 12.

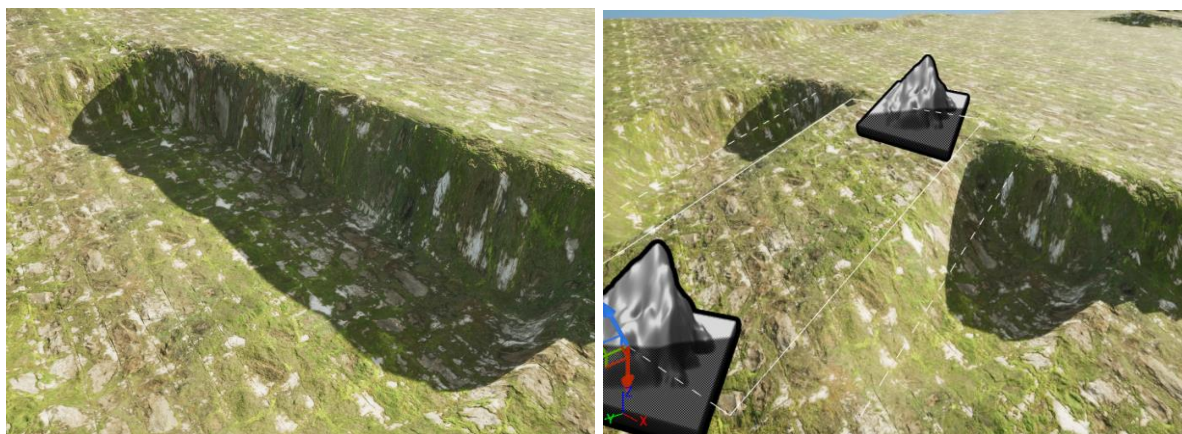


Рис. 12. Результат работы Ramp

1.3.6 Термическая эрозия – Thermal Erosion

Инструмент Thermal Erosion используется для имитации эрозии, вызванной перемещением почвы возвышенностей на более низкие участки. Результат работы инструмента Thermal Erosion представлен на рисунке 13.



Рис. 13. Результат работы Thermal Erosion

1.3.7 Гидроэрозия – Hydro Erosion

Инструмент Hydro Erosion используется для имитации эрозии, вызванной водой. Результат работы инструмента Hydro Erosion представлен на рисунке 14.

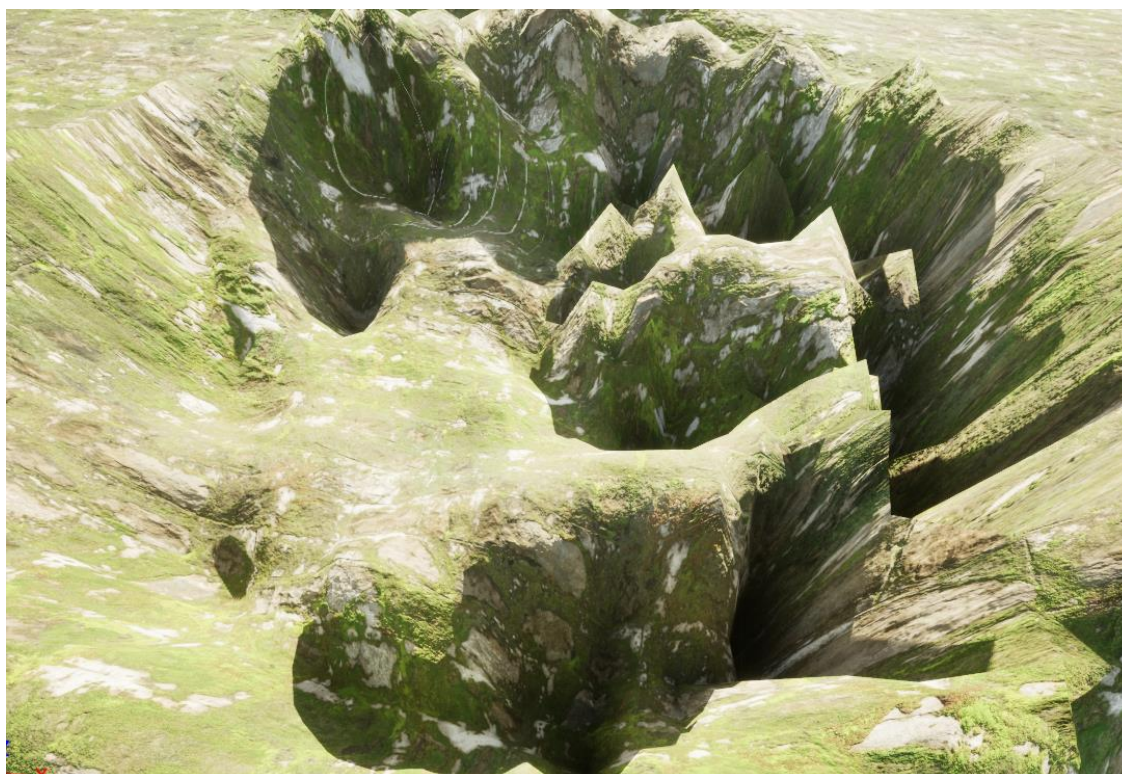


Рис. 14. Результат работы Hydro Erosion

1.3.8 Noise

Инструмент Noise используется для плавного создания углублений посредством шумов. Результат работы инструмента Noise представлен на рисунке 15.



Рис. 15. Результат работы Noise

2. Материал ландшафта

После создания или редактирования ландшафта следует нанести материал на него. Материал ландшафта можно применить на моменте создания самого ландшафта в поле Material в инструменте Manage или после создания ландшафта.

Одними из основных методов нанесения определенного материала на ландшафт является:

1. Ручная раскраска слоев ландшафта
2. Создание авто-материала, который наносит определенные текстуры в зависимости от высоты, скорости перехода ландшафта.

Ручная раскраска слоев ландшафта представляет собой создание мастер материала, который будет содержать в себе несколько материалов с различными текстурами, которые подключены к Layer. Layer используется для создания возможности содержания множества материалов и возможности нанесения

каждого их них на ландшафт. Данный способ не является эффективным, т к придется вручную прокрашивать каждый слой.

Более эффективным является создание авто-материала, который будет раскрашивать слои ландшафта автоматически в зависимости от их высоты, скорости изменения.

Разработанный авто-материал соответствия материала с высотой ландшафта представлен на рисунке 16.

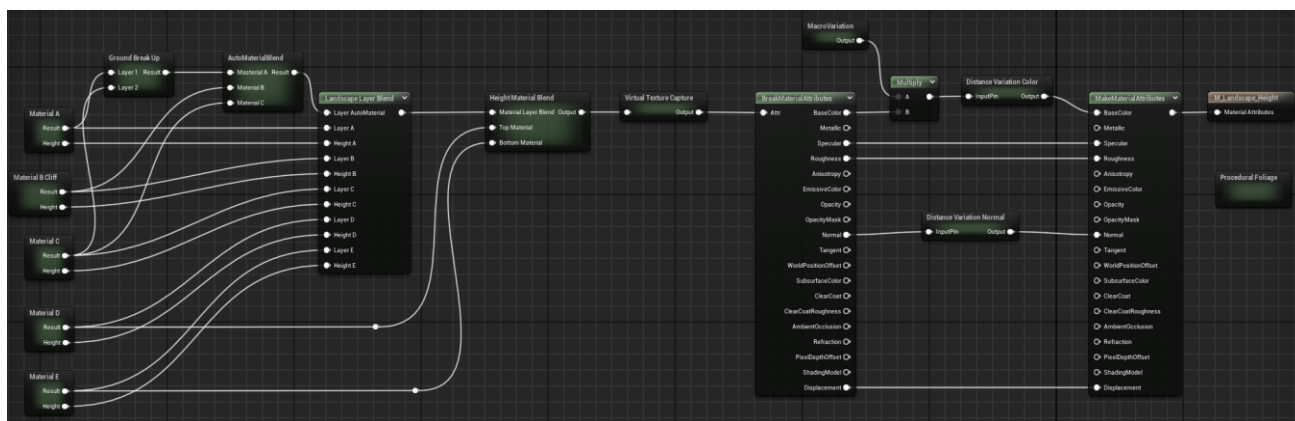


Рис. 16. Авто-материал соответствия материала с высотой ландшафта.

Сами материалы А, В, С, D, Е реализованы по следующему принципу, представленном на рисунке 17.

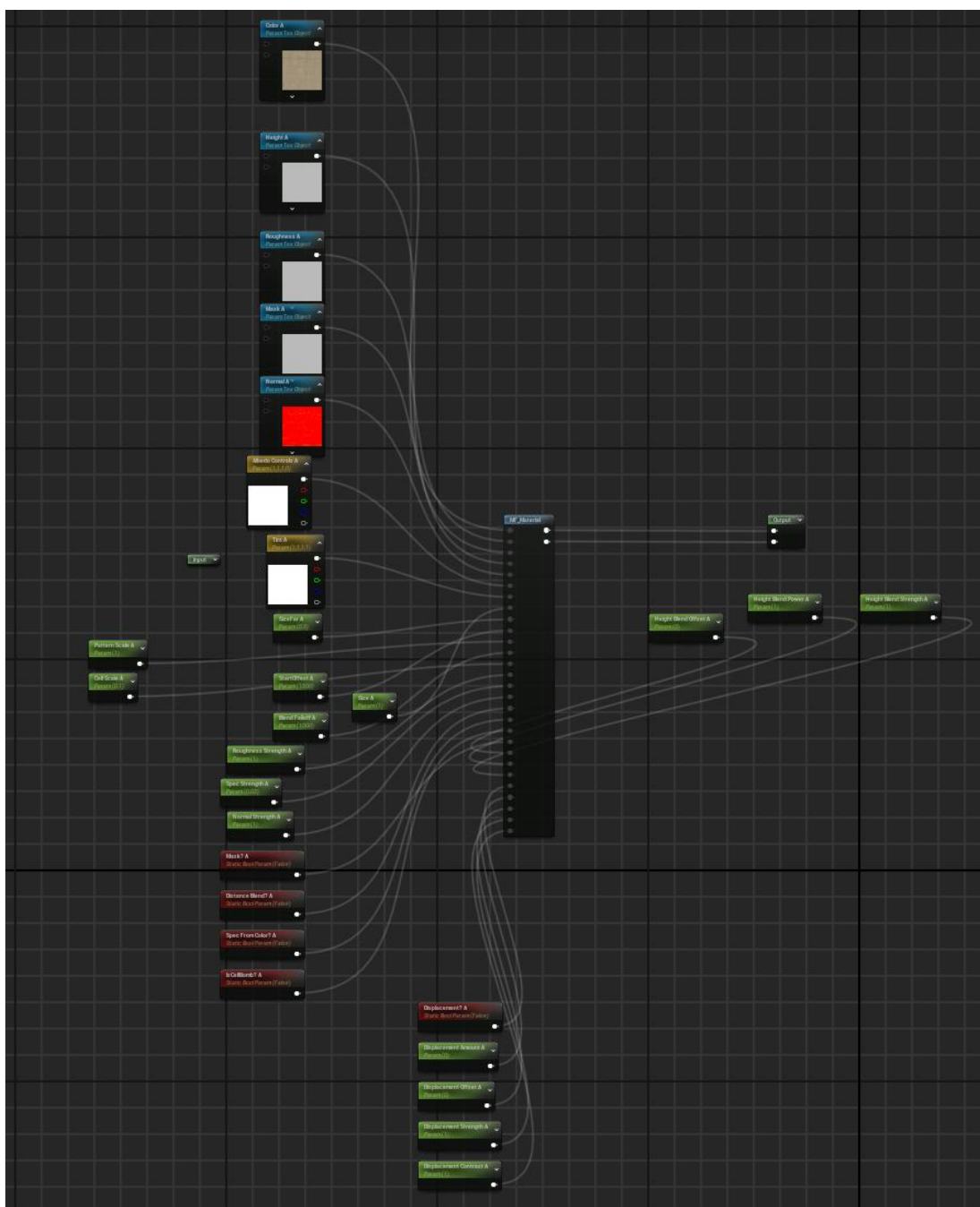


Рис. 17. Принцип реализации материалов А, В, С, D, Е.

В результате создания авто-материала был получен Material Instance, который возможно использовать для текстурирования ландшафта. На рисунке 18 представлены полученные возможности созданного Material Instance.



Рис. 18. Возможности Material Instance

3. Растительность

Для добавления растительности в сцену движок Unreal Engine 5 имеет множество инструментов. Рассмотрим инструмент Foliage mode. Для выбора данного инструмента в панели выбора следует переключить режим на Foliage mode. Выбор режима Foliage mode представлен на рисунке 19.

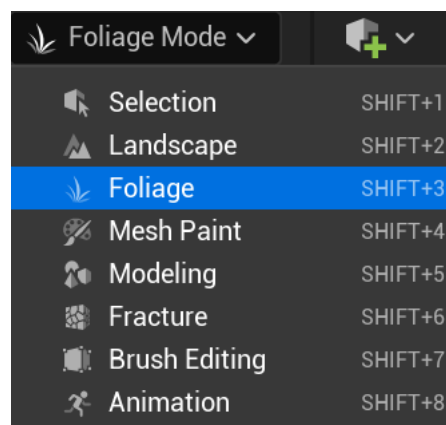


Рис. 19. Выбор Foliage mode

В появившемся окне Foliage mode для возможности нанесения растительности следует перенести статическую сетку модели в поле Foliage.

Нанесение растительности на поверхность ландшафта происходит за счет использования кисти, параметры которой можно редактировать. Окно Foliage mode представлено на рисунке 20.

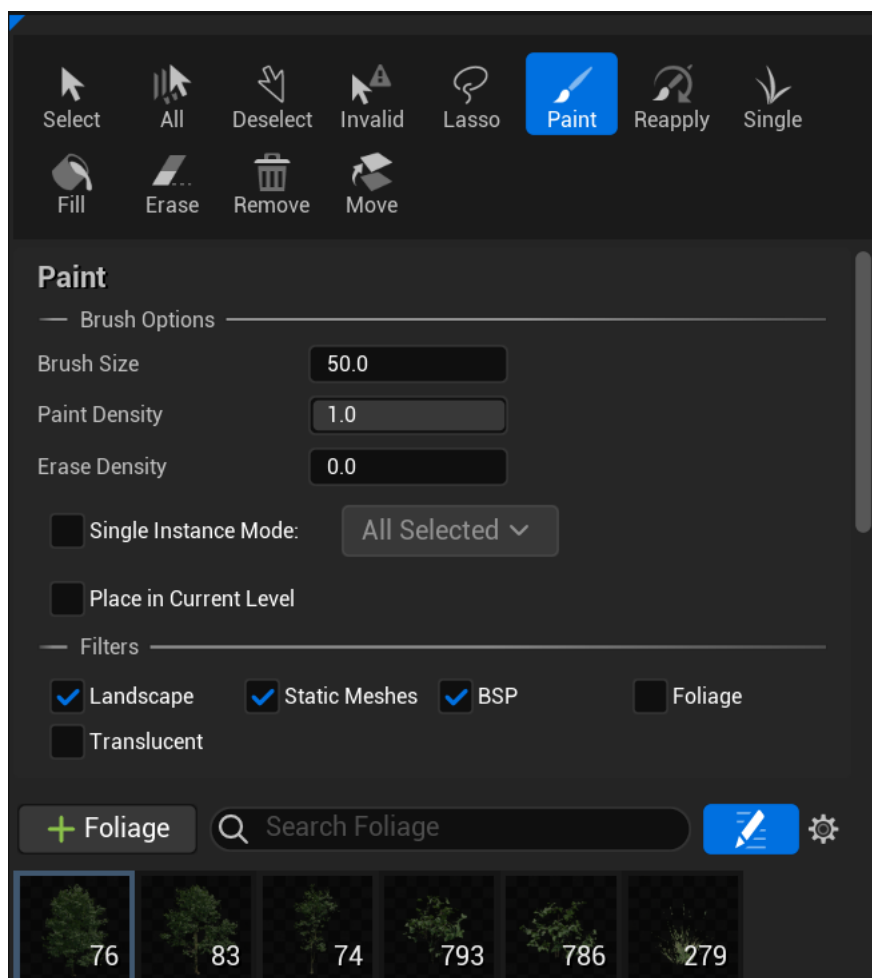


Рис. 20. Окно Foliage mode

В Unreal Engine 5 есть возможность нанесения сразу нескольких объектов, для этого следует выбрать необходимые объекты в панели Foliage и сделать их активными для нанесения. После того, как были выбраны необходимые объекты, инструмент Foliage предлагает параметры, которые могут быть изменены в зависимости от цели, которая необходима.

3.1 Painting

Настройка Painting используется для изменения параметров «рисования» на какой-либо области. Она включает в себя настройку плотности, размера. Настройка Painting представлена на рисунке 21.

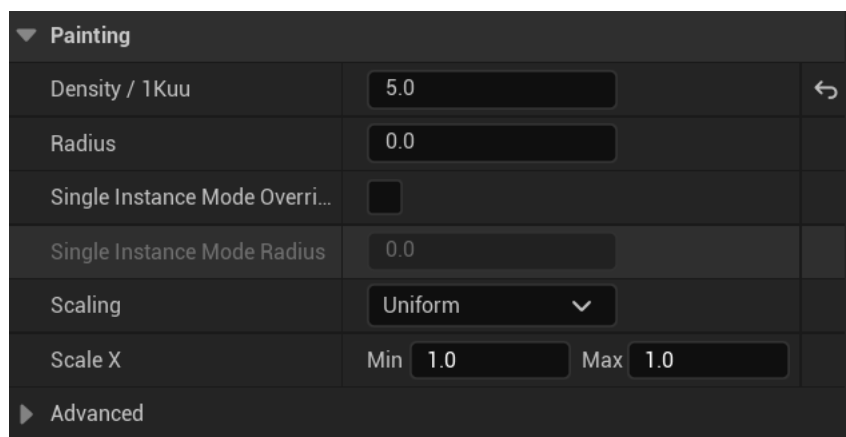


Рис. 21. Настройка Painting

3.2 Placement

Немаловажной настройкой Foliage mode является настройка позиционирования объекта. В данной категории представляется возможность выравнивания объектов по нормали, добавления смещения по оси Z, которое используется для того, чтобы объект полностью находился на поверхности ландшафта. Настройка Placement представлена на рисунке 22.

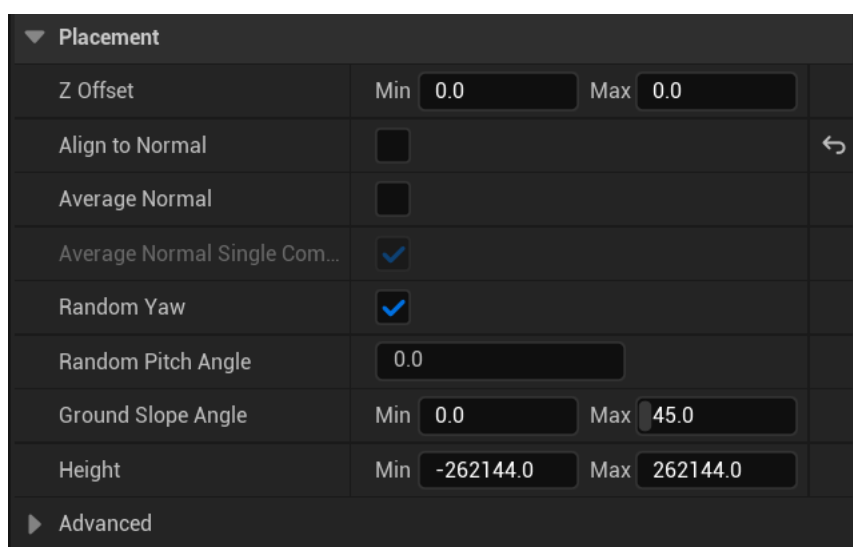


Рис. 22. Настройка Placement

3.3 Instance Static

Данная настройка предназначена для изменения теней, коллизий объекта. Настройка Instance Static представлена на рисунке 23.

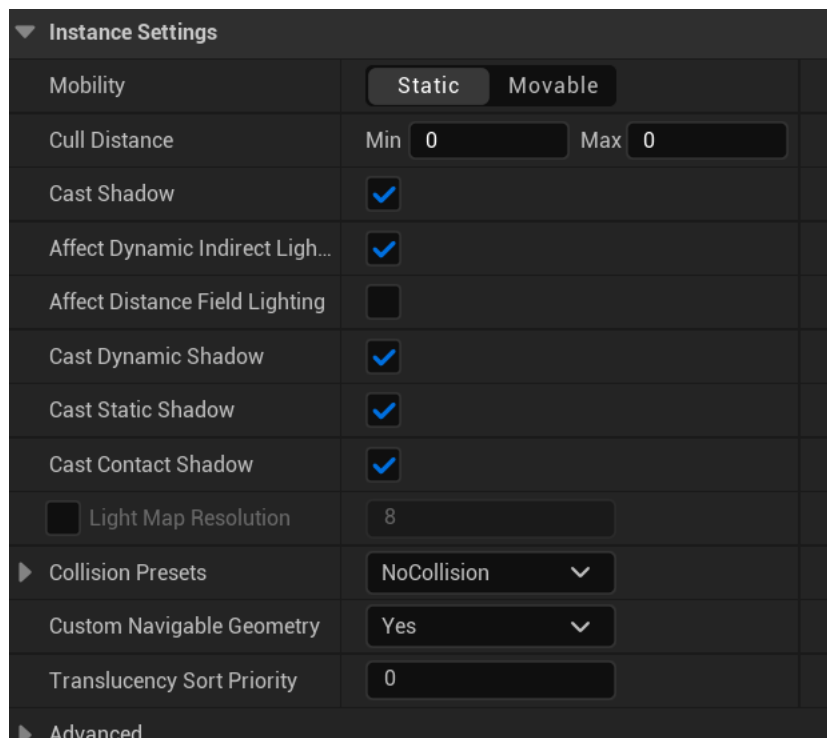


Рис. 24. Настройка Instance Static

4. Procedural Content Generation

При работе с огромными ландшафтами добавление растительности и различных 3D объектов в сцену становится затруднительным и времязатратным из-за ручного добавления. Unreal Engine имеет множество вариантов решения данной проблемы, одним из них является Procedural Content Generation (PCG).

PCG – это набор инструментов для создания собственного процедурного контента и инструментов внутри Unreal Engine. PCG предоставляет техническим художникам, дизайнерам и разработчикам возможность создавать быстрые итеративные инструменты и контент любой сложности, начиная от утилит активов, таких как задания или генерация биомов и заканчивая целыми мирами.

PCG в Unreal Engine представляется в виде графа, который начинается с получения информации о ландшафте и заканчивается спавнером статических сеток различных объектов. Пример простейшей реализации PCG представлена на рисунке 25.

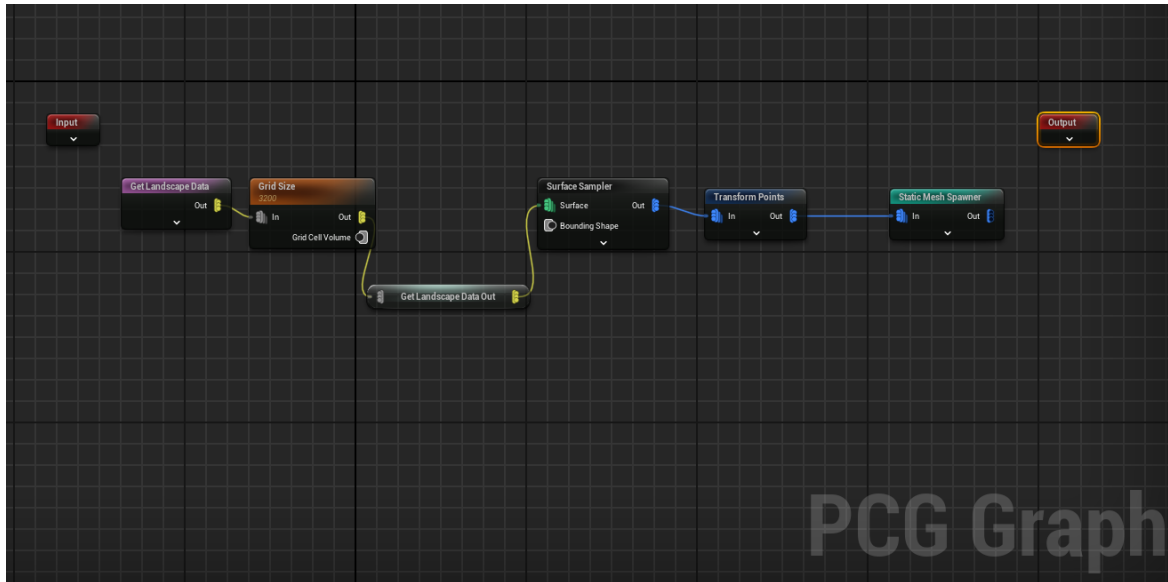


Рис. 25. Простейший пример PCG

В данном примере блок Grid Size используется для оптимизации полученного биома. Благодаря этому блоку объекты будут появляться только в определенной близости к игровому персонажу. Для того, чтобы данные изменения вступили в силу в настройках PCG объекта, расположенном на сцене, следует включить опцию is Partitioned и Generation Triggered переключить в Generate at Runtime. В самих настройках PCGWorldActor следует включить опцию Treat Editor Viewport as Generation Source.

Блоки Surface Sampler и Transform Point используются для придания реалистичности появляющихся объектов, таких как смещение относительно каждого соседнего объекта, смещение относительно ландшафта, выравнивание, размерность, угол поворота по оси.

Пример работы данного PCG Graph представлен на рисунке 26. Для наглядности блок Grid Size отключен.



Рис. 26. Пример работы PCG Graph

Полученный результат не является достаточным ведь возможно, что внутри леса находится какие-либо другие объекты, например озеро или город. Сама область внутреннего объекта может быть задана при помощи сплайна. Дополненный PCG Graph леса представлен на рисунке 27.

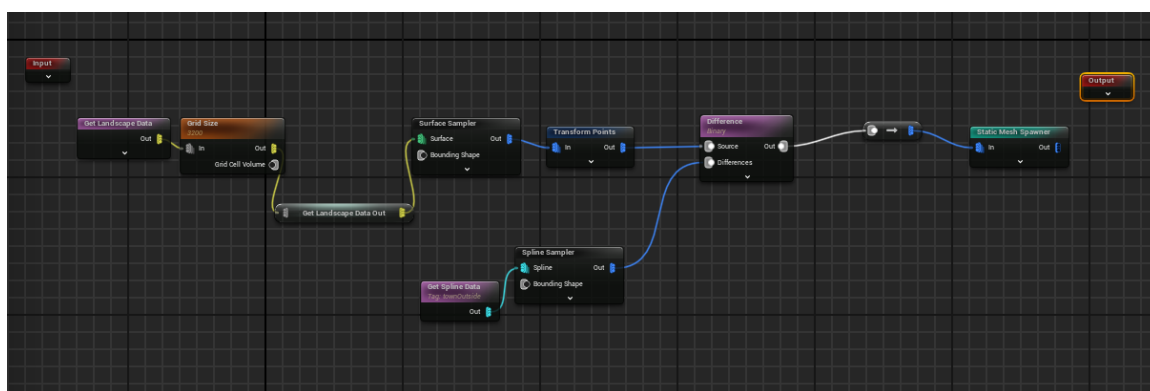


Рис. 27. Дополненный PCG Graph леса.

Дополнением является то, что теперь PCG Graph леса получает информацию о находящемся сплайне внутри и спавнит объекты за пределами сплайна в своей допустимой области. Для реализации самого поселения необходимо создать еще один PCG Graph, который будет спавнить объекты внутри определенной области, а именно созданного сплайна. Для большей реализации и возможностей также был добавлен сплайн дороги. Реализация PCG Graph поселения представлена на рисунке 28.

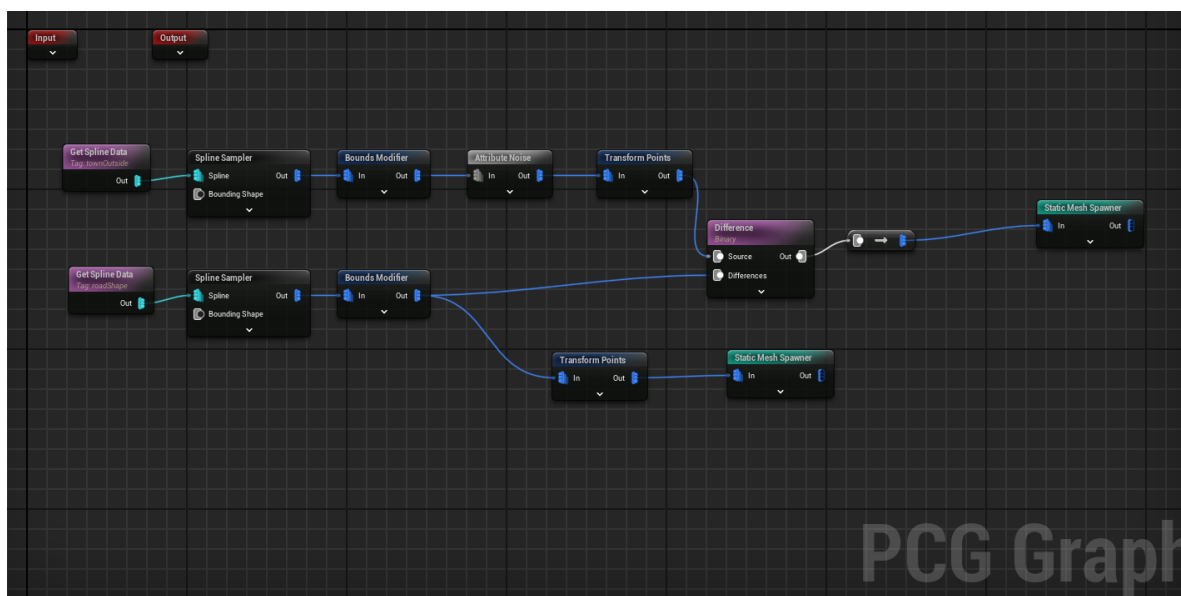


Рис. 28. Реализация PCG Graph поселения

Результат работы данных PCG Graph-ов представлен на рисунке 29.



Рис. 29. Результат работы PCG Graph-ов.

5. Средневековое поселение

Для создания средневекового поселения были использованы:

1. Библиотека Quixel Megascans
2. Marketplace Fab

Созданные 3D модели зданий представлены на рисунке 30.



Рис. 30. 3D модели зданий

6. Настройка неба

Для получения большей реалистичности создаваемой сцены было изменено небо, т. к. само небо предоставляемое Unreal Engine не является фотореалистичным. Для изменения был задействован плагин предоставляемой Unreal Engine – HDRIBackdrop. Для установки данного плагина в проект Unreal Engine следует в панели Edit -> Plugins найти плагин HDRIBackdrop и подключить, после перезапустить проект.

HDRIBackdrop является Actor-ом, поэтому для добавления его в сцену в панели быстрого добавления следует выбрать Place Actor Panel. Окно быстрого добавления представлена на рисунке 31.

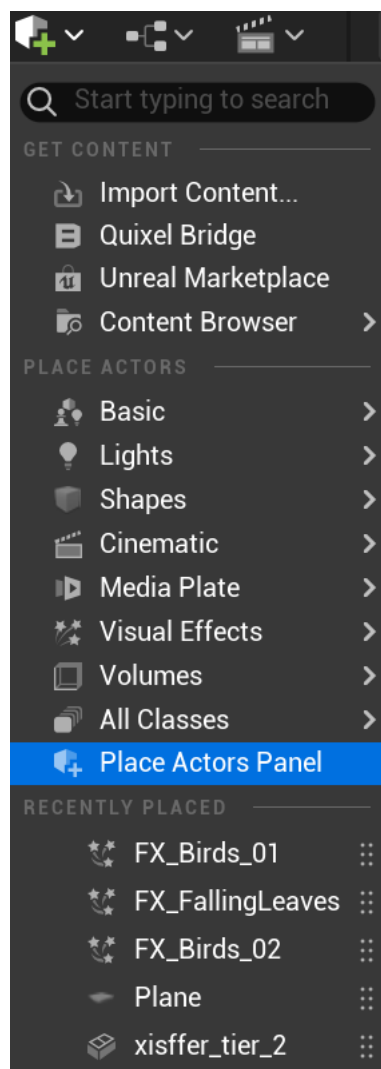


Рис. 31. Панель быстрого добавления

После добавления HDRIBackdoor в сцену в панели деталей следует заменить HDRI карту на желаемую и изменить размер Scale. Изменение размера требуется для того, чтобы в различных ракурсах можно было увидеть результат.

HDRIBackdoor имеет настройки не только для изменения неба, но и самого солнечного света в целом, для этого в панели деталей HDRIBackdoor следует выбрать SkyLight.

7. Водные объекты

Unreal Engine 5 имеет инструменты для добавления и последующей работы с водными объектами. Одним из решений добавления водных объектов является подключение плагина Water. Данный плагин предоставляет набор водных инструментов и методов рендеринга, позволяющих легко добавлять океаны, реки,

озера или пользовательские водоемы, которые создаются в ландшафте и взаимодействуют с игровым процессом.

Для добавления водного объекта с использованием плагина в меню Place Actor Panel -> Water следует выбрать конкретный водный объект, который должен быть добавлен в сцену. Список доступных водных объектов и результат работы Water River представлены на рисунках 32 и 33.

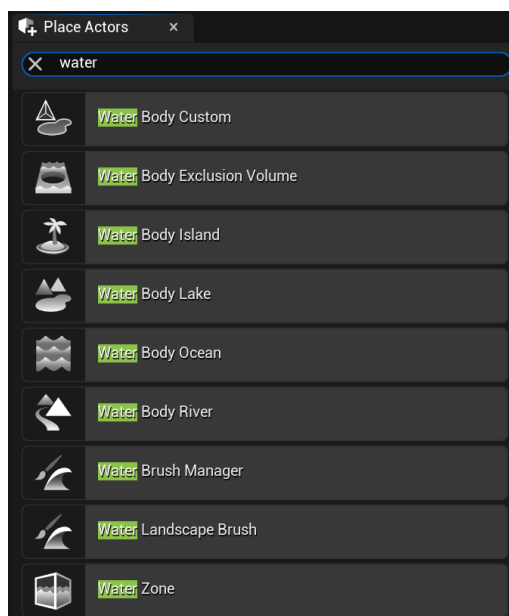


Рис. 32. Список доступных водных объектов при подключении Water плагина.

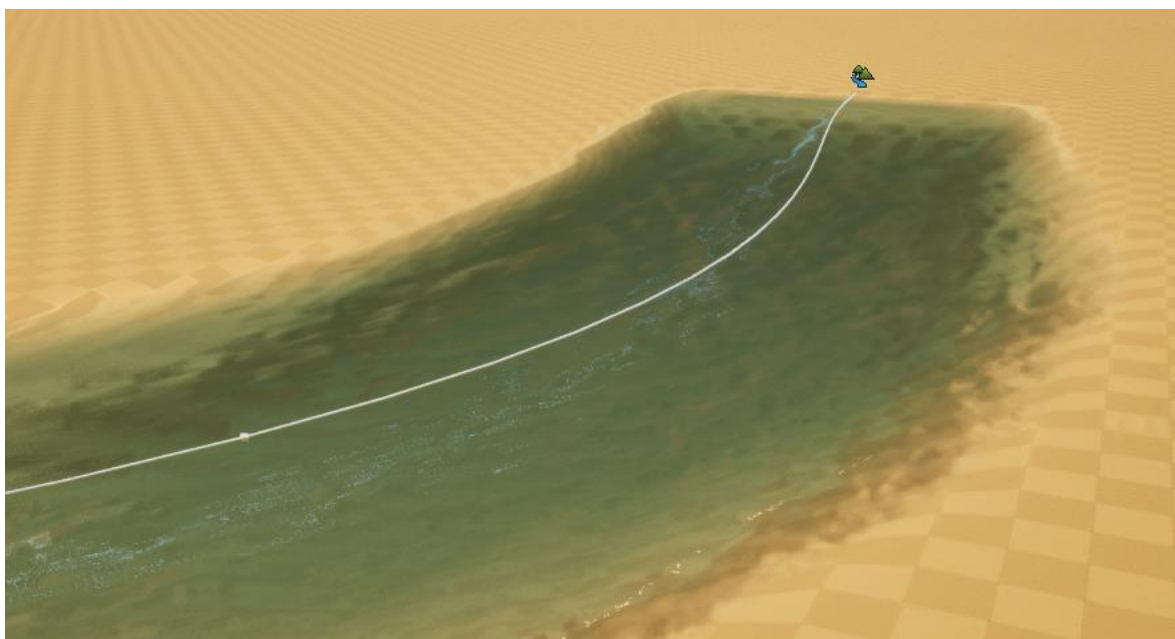


Рис. 33. Результат работы Water River

Данный плагин предоставляет разработчику сцены полную возможность настройки вводного объекта посредством использования сплайнов, изменения параметров водного объекта в определенной точке сплайна, например: глубины, ширины, скорости и др. Также присутствует возможность изменения параметров в материале воды для получения наилучшего визуального эффекта воды.

Другим возможным решением добавления водных объектов является создание углубления при помощи инструментов редактирования ландшафта, последующим созданием поверхности и добавления на нее водного материала, который возможно найти среди бесплатных ассетов, предоставленных в Fab.

8. Создание тумана

Для большей реалистичности полученной сцены был создан материал тумана. Схема полученного материала тумана представлена на рисунке 34.

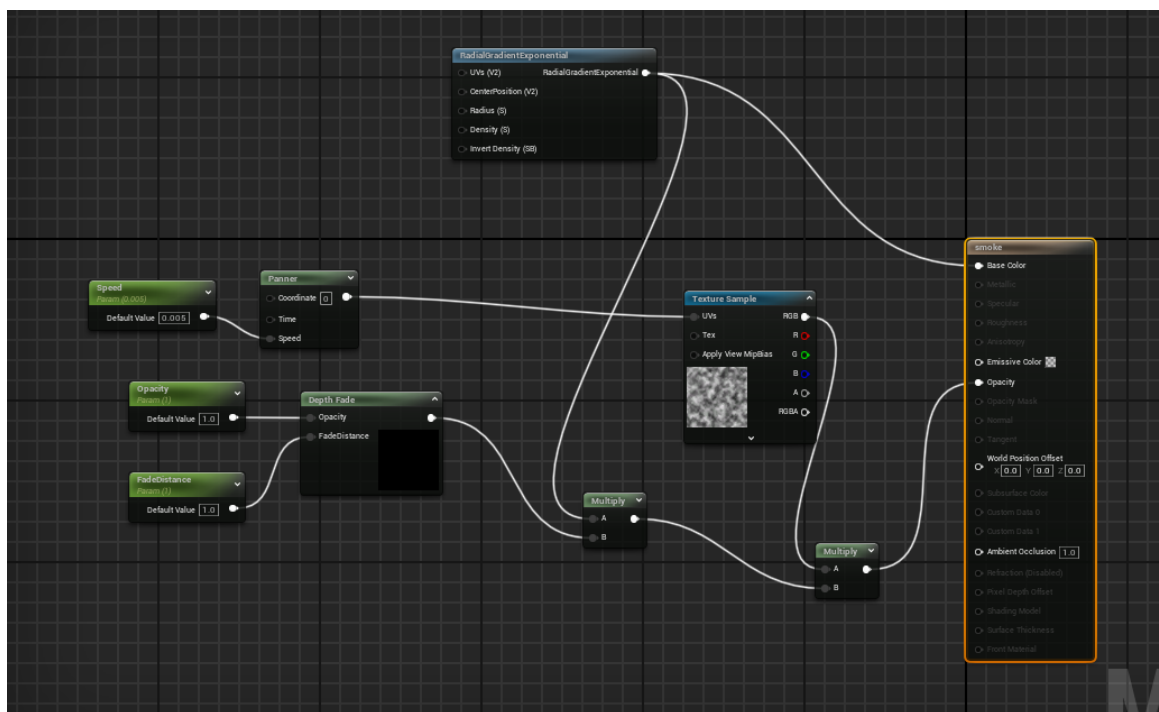


Рис. 34. Материал тумана

Сам материал был добавлен на созданную поверхность, которая была размещена вдоль горных объектов.

9. Настройка освещения

Движок Unreal Engine имеет обязательные компоненты сцены для настройки освещения.

1. DirectionalLight

Световой компонент с параллельными лучами, который обеспечивает равномерное освещение любой поражаемой поверхности (например, Солнца). Влияет на все объекты в определенном объеме, имеющие значение световой массы.

2. ExponentialHeightFog

Используется для создания эффектов запотевания, таких как облака, но плотность которых зависит от высоты тумана.

3. SkyAtmosphere

Представляет собой материал атмосферы планеты и имитирующий небо и рассеяние света в нем.

4. SkyLight

Захватывает свет неба и освещает текстурой неба всю сцену.

Для более точной настройки освещения Unreal Engine предоставляет дополнительные возможности. Одной из них является Post Process Volume. Данный эффект позволяет определять общий внешний вид сцены с помощью комбинированного выбора свойств и функций, влияющих на окраску, тональное отображение, освещение и многое другое.

10. Оптимизация

При использовании библиотеки готовых ассетов Quixel Megascans полученные 3D объекты являются очень детализированными, что не всегда является верным решением при разработке динамических сцен, где важную роль играет производительность. Для этого существует система LOD.

LOD означает уровень детализации. Для одного объекта может быть несколько уровней детализации. Самый низкий уровень детализации (LOD0) является наиболее подробным, когда вы находитесь близко к объекту.

Наивысший уровень детализации (LODx) обеспечивает наименьшее количество деталей, когда объект находится очень далеко от камеры и вы не можете разглядеть детали. Unreal поддерживает 8 уровней детализации, так что между самым низким и самым высоким уровнями детализации может быть множество шагов.

Оптимизация за счет LOD системы возможна, когда самым низким уровнем детализации становится не LOD0, а LOD2 – LOD4. Данное решение уменьшается размерность сетки объекта, из-за чего становится меньше треугольников при рендере модели в сцене.

Изменение минимального LOD-а представлено на рисунке 35.

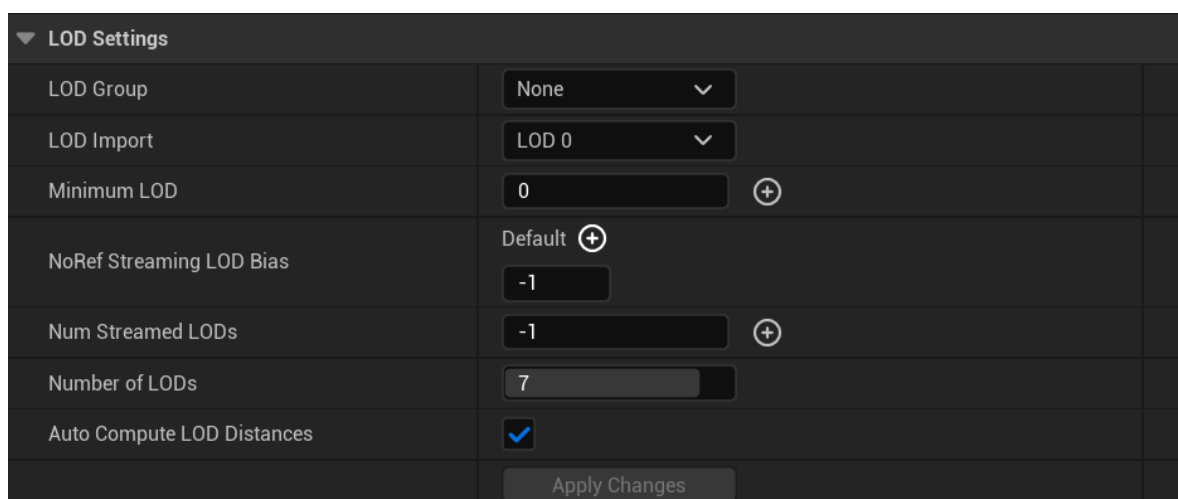


Рис. 35. Изменение минимального LOD

Сравнение производительности 10 близко стоящих объектов при использовании Lumen и определенных уровнях детализации представлено в таблице 1.

LOD	Кол-во треугольников	FPS
0	3,039,796	12
1	2,309, 645	14
2	1,022,547	25
3	954,495	30
4	96,289	60

Таблица 1. Сравнение производительности при использовании LOD

11. Создание кат-сцены

Кат-сцена – это эпизод, который прерывает стандартный геймплей и используется для кинематографичного изложения сюжета.

Для создания кат-сцены движок Unreal Engine предоставляет Level Sequence Actor, который находится в панели добавления Actor. На рисунке 36 представлен интерфейс редактора Sequencer.

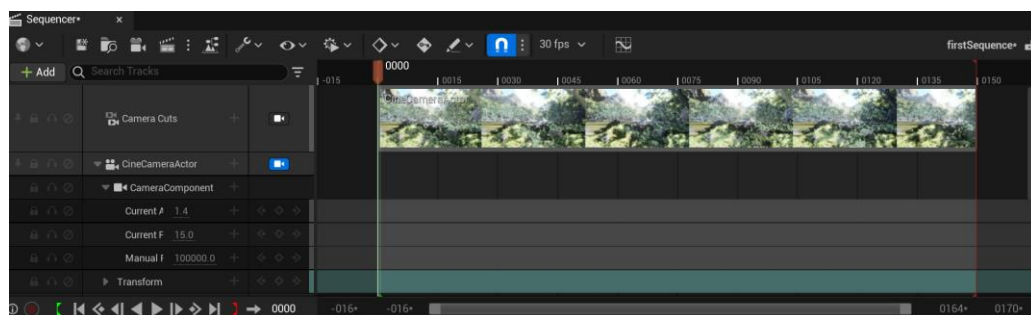


Рис. 36. Интерфейс редактора Sequencer

Для создания кат-сцены следует переместить камеру в необходимое место на сцене и добавить маркер в редакторе Sequencer.

12. Создание NPC

NPC – неигровой персонаж, который служит для важным средством создания атмосферы в сцене. В Unreal Engine логика неигрового персонажа осуществляется за счет Behavior Tree – дерево поведения, компонентом которого служит Blackboard. Blackboard – это ресурс, который хранит все данные, необходимые для конкретного контроллера AI, на который будет ссылаться дерево поведения.

Для реализации NPC было выбрано три его основных поведения:

1. Find Random Location – данное поведение реализуется случайную ходьбу по карте неигрового персонажа.
2. Find Player Location – поведение, которое реализует нахождение главного персонажа на сцене. Персонаж будет являться найденным если NPC его увидел в определенном поле своего зрения.

3. Chasing Player – поведение реализующее преследование главного персонажа. Преследование продолжается до тех пор, по NPC видит главного персонажа в своей определенной области.

Разработанное дерево поведения NPC представлено на рисунке 37.

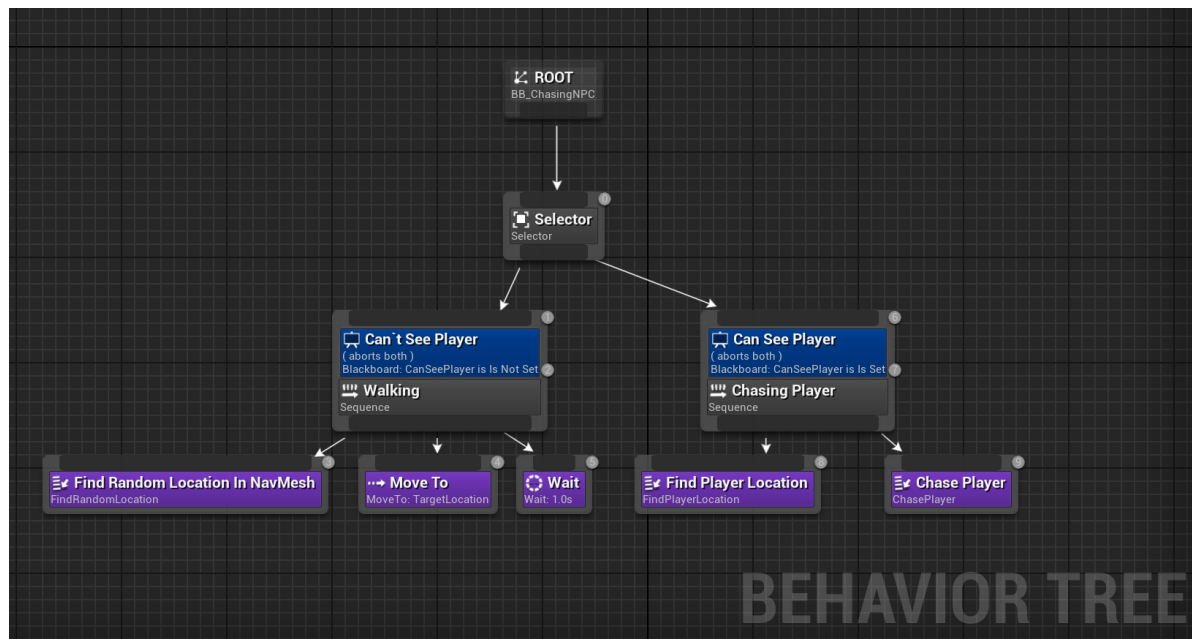


Рис. 37. Дерево поведения NPC

В ходе реализации NPC были созданы необходимые классы на языке программирования C++ с использованием технологии Blueprints для контроля необходимых анимаций персонажа.

1. ChasingNPC – класс NPC;
2. FindRandomLocation – класс поиска случайно локации;
3. FindPlayerLocation – класс поиска главного персонажа;
4. ChasePlayer – класс преследования главного персонажа;
5. ChasingNPC_AIController – класс служащий для контроля действий NPC.

Листинг 1. Реализация функции поиска случайно локации

```
EBTNodeResult::Type
UBTTask_FindRandomLocation::ExecuteTask(UBehaviorTreeComponent&
OwnerComp, uint8* NodeMemory)
{
    // get AI controller and it`s NPC
```

```

        if (auto* const cont =
Cast<AChasingNPC_AIController>(OwnerComp.GetAIOwner()))
        {
            if (auto* const npc = cont->GetPawn())
            {
                // obtain npc location to use as an origin
                auto const Origin = npc->GetActorLocation();

                // get the navigation system and generate a random
location
                if (auto* const NavSys =
UNavigationSystemV1::GetCurrent(GetWorld()))
                {
                    FNavLocation Loc;
                    if (NavSys-
>GetRandomPointInNavigableRadius(Origin, SearchRadius, Loc))
                    {
                        OwnerComp.GetBlackboardComponent()-
>SetValueAsVector(GetSelectedBlackboardKey(), Loc.Location);
                    }

                    // finish with success
                    FinishLatentTask(OwnerComp,
EBTNodeResult::Succeeded);
                    return EBTNodeResult::Succeeded;
                }
            }
        }

        return EBTNodeResult::Failed;
    }

```

Листинг 2. Функция, реализующая поиск главного персонажа

```

EBTNodeResult::Type
UBTTask_FindPlayerLocation::ExecuteTask(UBehaviorTreeComponent&
OwnerComp, uint8* NodeMemory)
{
    // get player character
    if (auto* const Player =
UGameplayStatics::GetPlayerCharacter(GetWorld(), 0))
    {
        // get player location to use as an origin
        auto const PlayerLocation = Player->GetActorLocation();
        if (SearchRandom)
        {
            FNavLocation Loc;

            // get the navigation system and generate a random
location near the player

```

```

        if (auto* const NavSys =
UNavigationSystemV1::GetCurrent(GetWorld()))
        {
            // try to get a random location near the
player
            if (NavSys-
>GetRandomPointInNavigableRadius(PlayerLocation, SearchRadius,
Loc))
            {
                OwnerComp.GetBlackboardComponent()-
>SetValueAsVector(GetSelectedBlackboardKey(), Loc.Location);
                FinishLatentTask(OwnerComp,
EBTNodeResult::Succeeded);
                return EBTNodeResult::Succeeded;
            }
        }
    }
    else
    {
        OwnerComp.GetBlackboardComponent()-
>SetValueAsVector(GetSelectedBlackboardKey(), PlayerLocation);
        FinishLatentTask(OwnerComp,
EBTNodeResult::Succeeded);
        return EBTNodeResult::Succeeded;
    }
}
return EBTNodeResult::Failed;
}

```

Листинг 3. Функция, реализующая преследование главного персонажа

```

EBTNodeResult::Type
UBTTask_ChasePlayer::ExecuteTask(UBehaviorTreeComponent&
OwnerComp, uint8* NodeMemory)
{
    // get target location from blackboard via the NPC`s
controller
    if (auto* const cont =
Cast<AChasingNPC_AIController>(OwnerComp.GetAIOwner()))
    {
        auto const PlayerLocation =
OwnerComp.GetBlackboardComponent()-
>GetValueAsVector(GetSelectedBlackboardKey());

        // move to the player`s location
        UAIBlueprintHelperLibrary::SimpleMoveToLocation(cont,
PlayerLocation);

        //finish with success
        FinishLatentTask(OwnerComp, EBTNodeResult::Succeeded);
        return EBTNodeResult::Succeeded;
    }
}

```

```
}  
    return EBTNodeResult::Failed;  
}
```

Результаты

В ходе выполнения курсового проекта были созданы два проекта:

1. Реалистичный пейзаж, представленный на рисунках 38, 39;
2. Сцена размером 4x4 км с детализированным средневековым поселением, представленная на рисунках 40, 41.

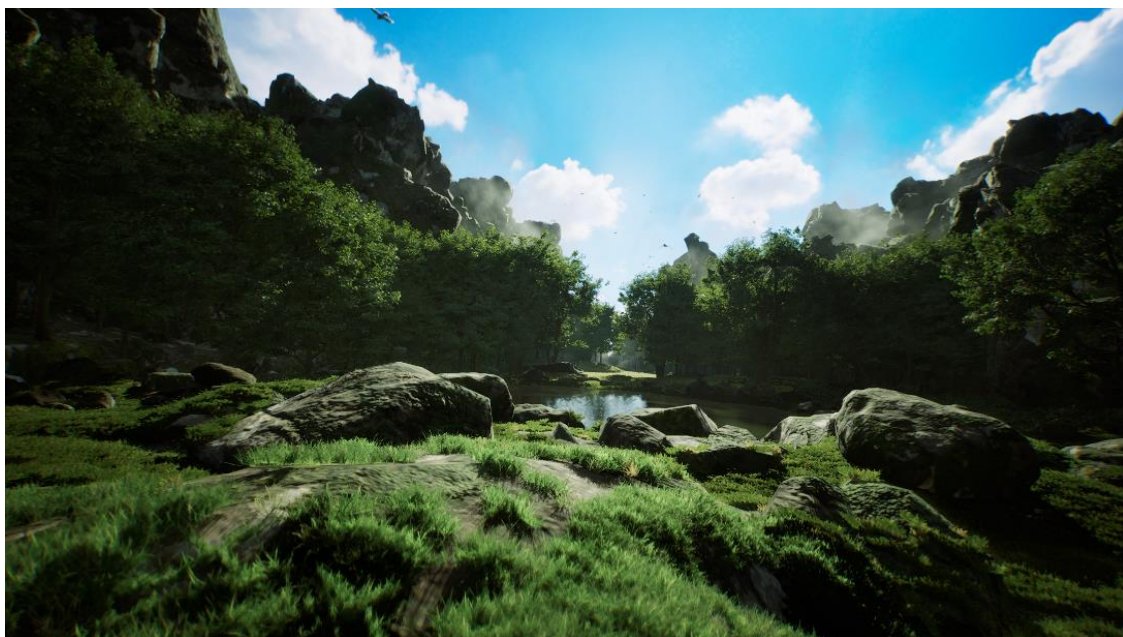


Рис. 38. Реалистичный пейзаж

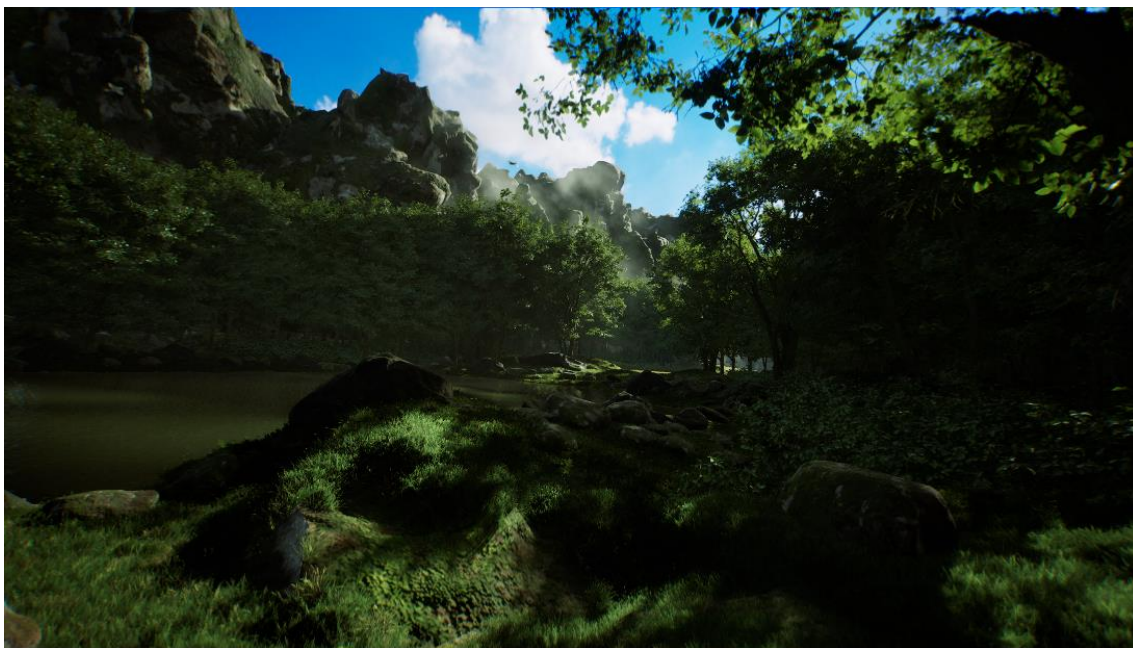


Рис. 39. Реалистичный пейзаж



Рис. 40. Средневековое поселение



Рис. 41. Средневековое поселение

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта были рассмотрены подходы создания реалистичного пейзажа местности, средневекового поселения и NPC. Изучены принципы создания материалов, процедурной генерация биомов, водных объектов, кат-сцены и настройки освещения. Рассмотрены варианты оптимизации. Изучены методы работы с плагинами Unreal Engine. Необходимо отметить, что для комфортной работы в Unreal Engine необходим производительный компьютер.

В процессе работы были выполнены следующие задачи:

1. Изучены методы построение ландшафта;
2. Изучены методы добавления растительности;
3. Изучены методы добавления водных объектов;
4. Подключение сторонних плагинов и работа с ними;
5. Настройка освещения;
6. Создана процедурная генерация биомов;
7. Создание материалов и наложение их на объекты сцены;
8. Создан фотореалистичный и детализированный пейзаж;
9. Разработан NPC;
10. Создано средневековое поселение;
11. Проведена оптимизация полученных сцен;
12. Создана кат-сцена получившегося пейзажа.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Epic Games Unreal Engine Documentation / Epic Games [Электронный ресурс] // Epic Games Developer: [сайт]. — URL: <https://dev.epicgames.com/documentation/ru-ru/unreal-engine/unreal-engine-5-5-documentation> (дата обращения: 15.10.2024);
2. Unreal Engine Building natural environments / Unreal Engine [Электронный ресурс] // YouTube: [сайт]. — URL: <https://www.youtube.com/watch?v=gbl1qgPOl3E> (дата обращения: 19.10.2024);
3. Том Шэннон Unreal Engine 4 для дизайна и визуализации [Текст] / Том Шэннон — 1-е изд.. — Москва: Vombora, 2021 — 368 с;
4. Epic Games Unreal Engine / Epic Games [Электронный ресурс] // GitHub: [сайт]. — URL: <https://github.com/EpicGames> (дата обращения: 7.11.2024);
5. Максименкова Ольга Вениаминовна, Веселко Никита Игоревич Программирование в Unreal Engine 5. Основы визуального языка Blueprint. [Текст] / Максименкова Ольга Вениаминовна, Веселко Никита Игоревич — 1-е изд.. — Москва: Эксмо, 2023 — 320 с.