



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:

***«Создание интерактивной среды в трехмерном движке
Unreal Engine 5»***

Студент РК6-73Б

(Подпись, дата)

Козлов В.В.

И.О. Фамилия

Руководитель

(Подпись, дата)

Витюков Ф.А.

И.О. Фамилия

2025 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

«____» _____ 2022 г.

ЗАДАНИЕ
на выполнение научно-исследовательской работы

по теме: Создание интерактивной среды в трехмерном движке Unreal Engine 5

Студент группы РК6-73Б

Козлов Вадим Владимирович
(Фамилия, имя, отчество)

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

График выполнения НИР: 25% к 5 нед., 50% к 11 нед., 75% к 14 нед., 100% к 16 нед.

Техническое задание: создание интерактивной среды в трехмерном движке Unreal Engine 5.

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на 14 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.):

Дата выдачи задания «3» сентября 2024 г.

Руководитель НИР

(Подпись, дата)

Витюков Ф.А.
И.О. Фамилия

Студент

(Подпись, дата)

Козлов В.В.
И.О. Фамилия

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

АННОТАЦИЯ

В данной работе рассматривается процесс создания интерактивной среды в трехмерном движке Unreal Engine 5. Целью курсового проекта является разработка высококачественной виртуальной среды, способной обеспечить пользователю увлекательный и интерактивный опыт.

В практической части КП была проведена разработка концепции и дизайна интерактивной среды, включающей в себя моделирование объектов, создание текстур и материалов, а также настройку освещения и атмосферы.

В ходе работы были использованы инструменты и возможности Unreal Engine 5, такие как Blueprints для визуального программирования, системы частиц для создания эффектов, а также технологии рендеринга для достижения высокого качества графики. Результатом проекта стала полноценная интерактивная среда, демонстрирующая возможности современного игрового движка и предоставляющая пользователю уникальный опыт взаимодействия с виртуальным миром.

Данная работа подчеркивает значимость использования современных технологий в разработке интерактивных приложений и открывает перспективы для дальнейших исследований в области создания виртуальных сред.

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	3
ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ	5
ВВЕДЕНИЕ.....	6
1. Создание поверхности в TerreSculptor.....	7
2. Создание поверхности в Unreal Engine	10
3. Размещение деревьев и камней и нанесение текстур.....	12
3.1 Замена текстур горных вершин и подножий	12
3.2 Размещение деревьев и камней на сцене.....	13
3.3 Нанесение текстур на модели камней и деревьев	14
4. Настройка отражений и тумана	16
4.1 Создание отражений на воде	16
4.2 Создание и настройка тумана	19
5. Создание палаточного лагеря и места крушения.....	22
5.1 Создание палаточного лагеря	22
5.2 Создание места крушения	23
ЗАКЛЮЧЕНИЕ	24
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	25

ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

UE5 (Unreal Engine 5) – трёхмерный движок Unreal Engine 5.

Terresculptor – редактор 3D-ландшафта.

Epic Games – американская компания, занимающаяся разработкой компьютерных игр и программного обеспечения.

Nanite – технология, позволяющая создавать сцены с высокой детализацией без перегрузки системы.

Blueprint – система визуального программирования в UE5 на основе нодов с данными (события и функции).

QB (Quixel Bridge) – торговая площадка для приобретения и скачивания различных моделей.

PFV (Procedural Foliage Volume) – инструмент, который позволяет автоматически генерировать и размещать растительность (листву, деревья, кустарники и т.д.) в заданной области уровня.

RTV (Runtime Virtual Textures) – технология, которая позволяет эффективно управлять текстурами и материалами в реальном времени, обеспечивая более высокое качество визуализации и производительность.

ВВЕДЕНИЕ

Unreal Engine 5 (UE5) — это один из самых мощных и популярных движков для разработки игр и интерактивных приложений, предлагающий инновационные технологии и инструменты для создания высококачественных трехмерных миров. Он был разработан компанией Epic Games и официально представлен в мае 2020 года.

Одним из ключевых аспектов разработки является создание интерактивных сред, которые могут реагировать на действия пользователя, изменяя своё состояние в зависимости от взаимодействия.

Цели проекта:

- Научиться создавать различные поверхности, а также импортировать их;
- Исследовать настройки материалов различных элементов;
- Проанализировать влияние различных объектов на производительность сцены;
- Понять, в чем отличие различных методов отражения.

Задачи проекта:

- Создать поверхность в TerreSculptor и импортировать ее в Unreal Engine;
- Доделать поверхность непосредственно в UE, добавив актеров;
- Разместить элементы природы, используя PFV;
- Настроить отражения и создать объёмный туман;
- Создать палаточный лагерь и место крушения самолета.

1. Создание поверхности в TerreSculptor

TerreSculptor 2.0 — мощный инструмент для создания ландшафтов, освоив возможности которого, можно создавать впечатляющие и реалистичные поверхности.

Создание поверхности в программе TerreSculptor 2.0 — это многоэтапный процесс, который включает в себя несколько ключевых шагов.

1. Создание нового проекта

Запуск TerreSculptor и создание нового проекта. Установка параметров проекта, такие как размер карты и разрешение.

2. Импорт данных

Если имеются существующие данные (например, высотные карты или текстуры), есть возможность импортировать их в проект. TerraSculptor поддерживает различные форматы файлов.

3. Создание базовой поверхности

Использование инструментов для создания базовой поверхности. Поверхность сделана с помощью генераторов ландшафта, таких как:

Noise (шум) — для создания случайных форм.

Perlin Noise — для более естественных и органических форм.

Fractal — для создания фрактальных ландшафтов.

4. Редактирование поверхности

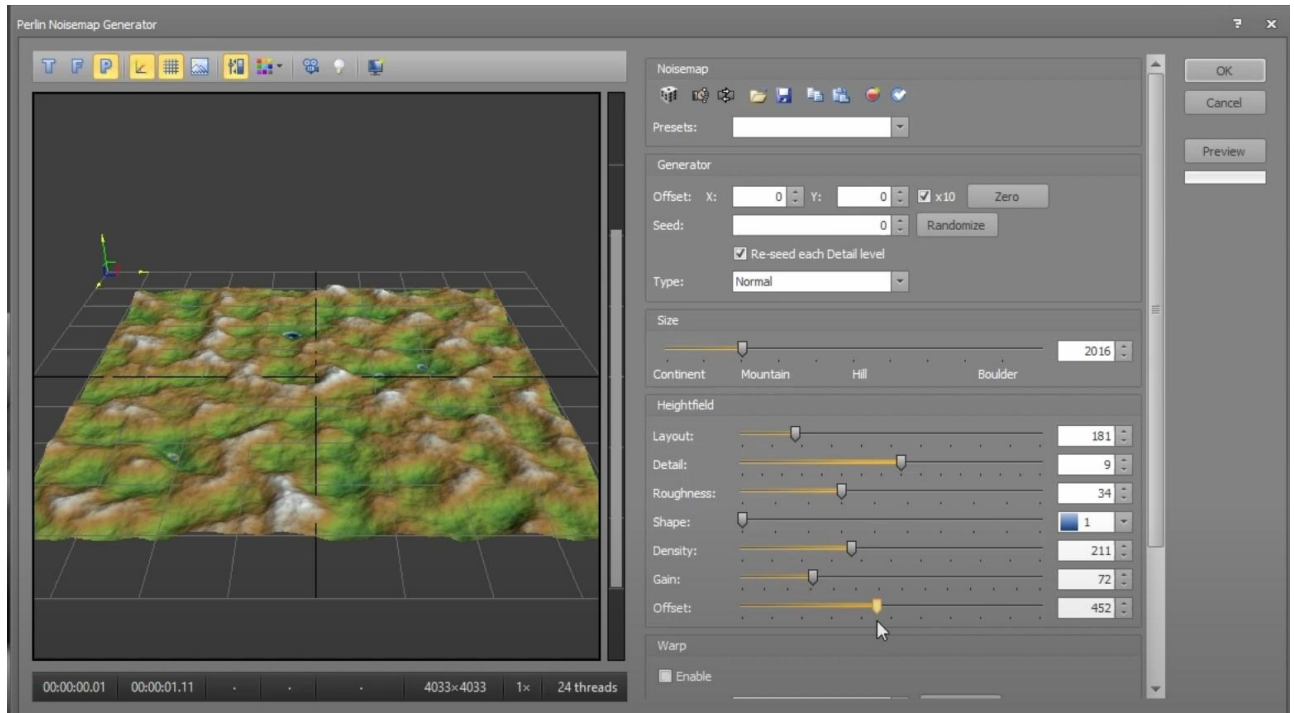


Рис. 1 «Создание поверхности в TerreSculptor с помощью Noise Map Generator»

Инструменты для редактирования:

Sculpting Tools — для ручного изменения высот.

Erosion — для симуляции эрозии и создания более реалистичных форм.

Smoothing — для сглаживания поверхности.

5. Экспорт данных из TerreSculptor и их импорт в Unreal Engine

После окончания работы над проектом, выполняется экспорт созданной поверхности в нужном формате для использования в Unreal Engine 5.

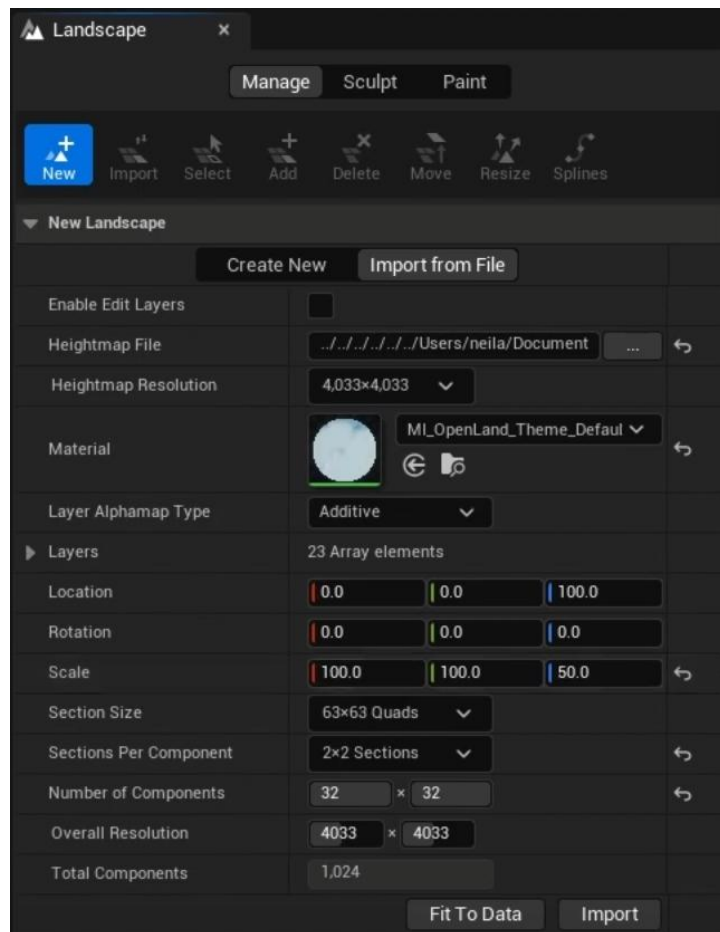


Рис. 2 «Импортирование материала в Unreal Engine»

Нюансы, которые встретились во время работы:

- Параметры генерации: проводились эксперименты с параметрами генерации, которые значительно влияли на конечный результат.
- Производительность: работа с высокими разрешениями потребовала значительных ресурсов. Пришлось искать компромисс между качеством и производительностью.

2. Создание поверхности в Unreal Engine

После импортирования базовой поверхности из TerreSculptor, необходимо доработать ландшафт, добавив несколько элементов. Одним из таких элементов является озеро.

Озеро – неотъемлемая часть арктического пейзажа. Инструменты TerreSculptor не позволяют его сделать, поэтому создание воды ведется непосредственно в UE5.

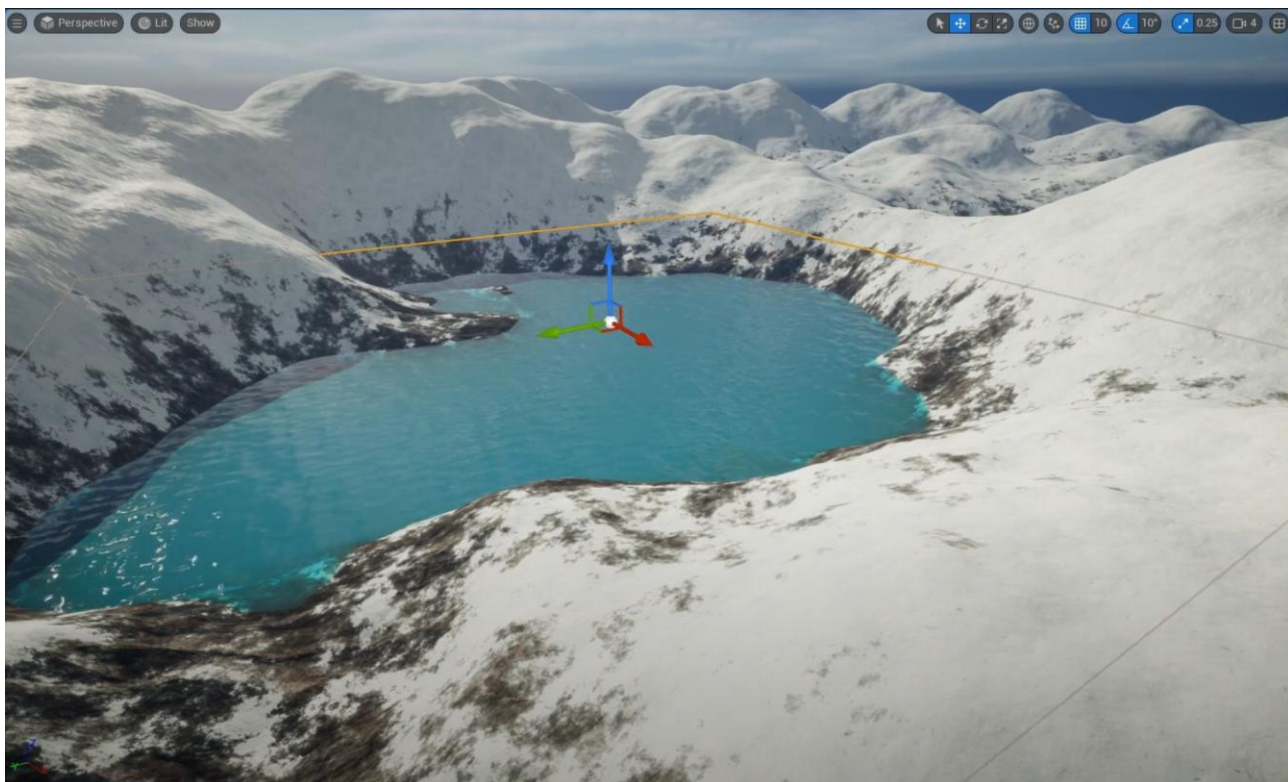


Рис. 3 «Озеро в Unreal Engine с материалом по умолчанию»

Используя актер класса Water body custom, можно создать блок воды в любом удобном месте поверхности.

Озеро создается с материалом по умолчанию, поэтому цвет воды необходимо настроить таким образом, чтобы он соответствовал общей атмосфере сцены. Для этого используется изменение параметров материала, что позволяет его настроить без каких-либо проблем.

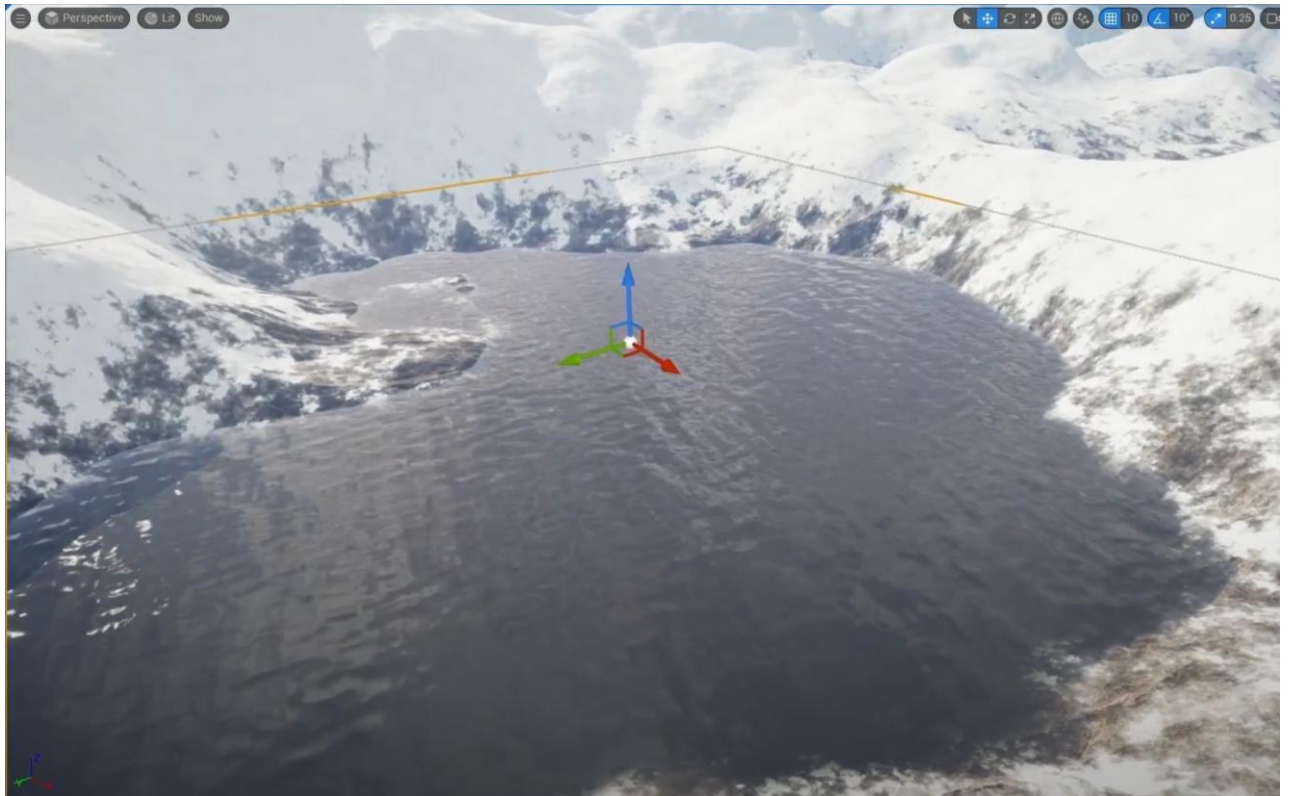


Рис. 4 «Озеро в Unreal Engine с настроенным материалом»

После нанесения правильного материала и выбора его нужных параметров, озеро принимает более тусклый оттенок, а также становится менее прозрачным, что прекрасно вписывается в концепцию арктического пейзажа.

3. Размещение деревьев и камней и нанесение текстур

В этом блоке рассматривается нанесение текстур на поверхности всей сцены, а также создание элементов непосредственно в самой сцене, таких как деревья и камни, с помощью инструмента Procedural Foliage Volume.

3.1 Замена текстур горных вершин и подножий

Для начала, стоит заменить слишком резкие текстуры поверхности на более мягкие. Для этого используется платформа Quixel Bridge – торговая площадка, которая позволяет купить или скачать нужные модели или текстуры. В сцене необходимо заменить два сета текстур: вершины гор заменяются на более мягкий белый цвет, а их подножия – на рыхлый, менее резкий.

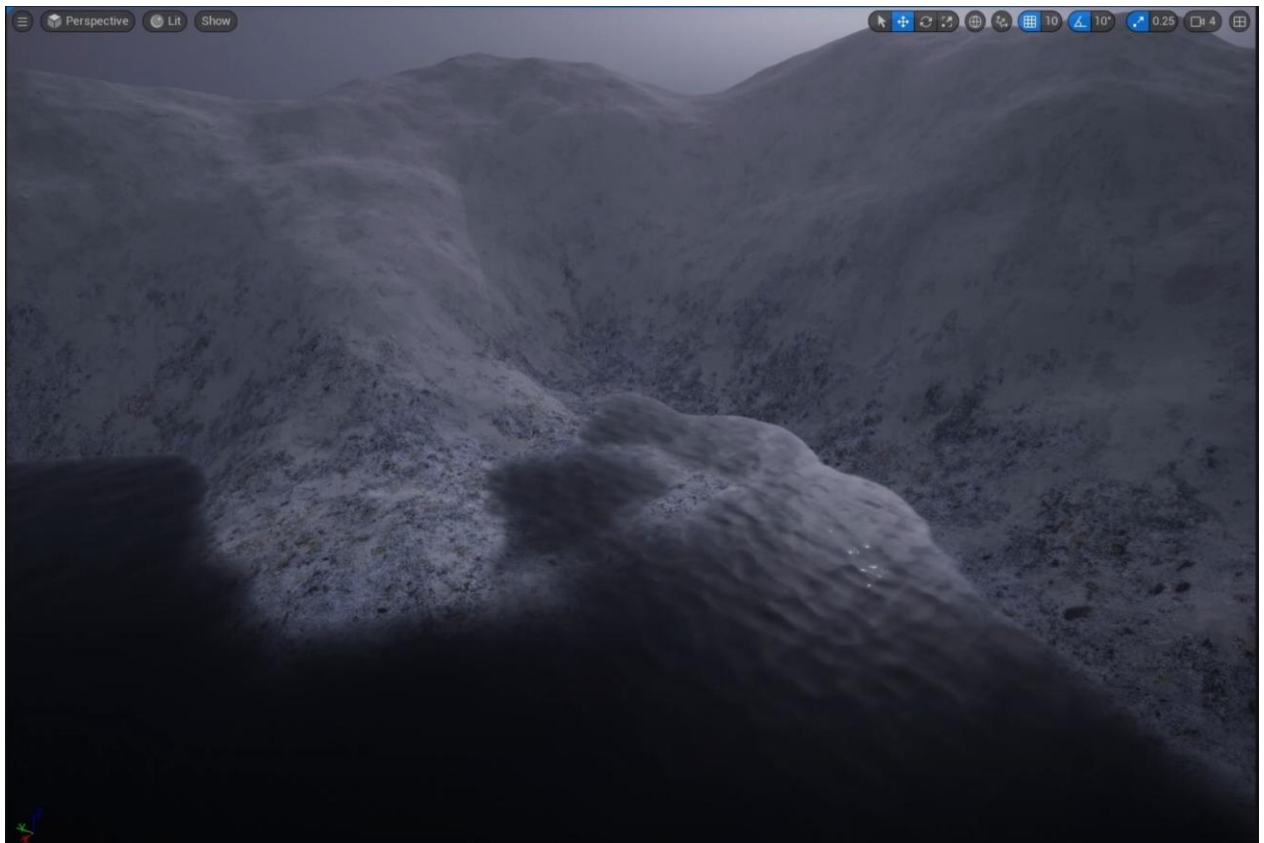


Рис. 5 «Замененные текстуры снега на вершинах гор и у их подножий»

3.2 Размещение деревьев и камней на сцене

Ассеты для деревьев и камней можно также найти в каталоге Quixel Bridge. В параметрах моделей необходимо настроить несколько функций, главными из которых являются **коллизия** (collision) и **прирост** (growth).

Чтобы не создавать каждое дерево вручную, уместно воспользоваться утилитой **PFV** (ProceduralFoliageVolume), которая позволяет создать большое количество моделей в выбранной области.

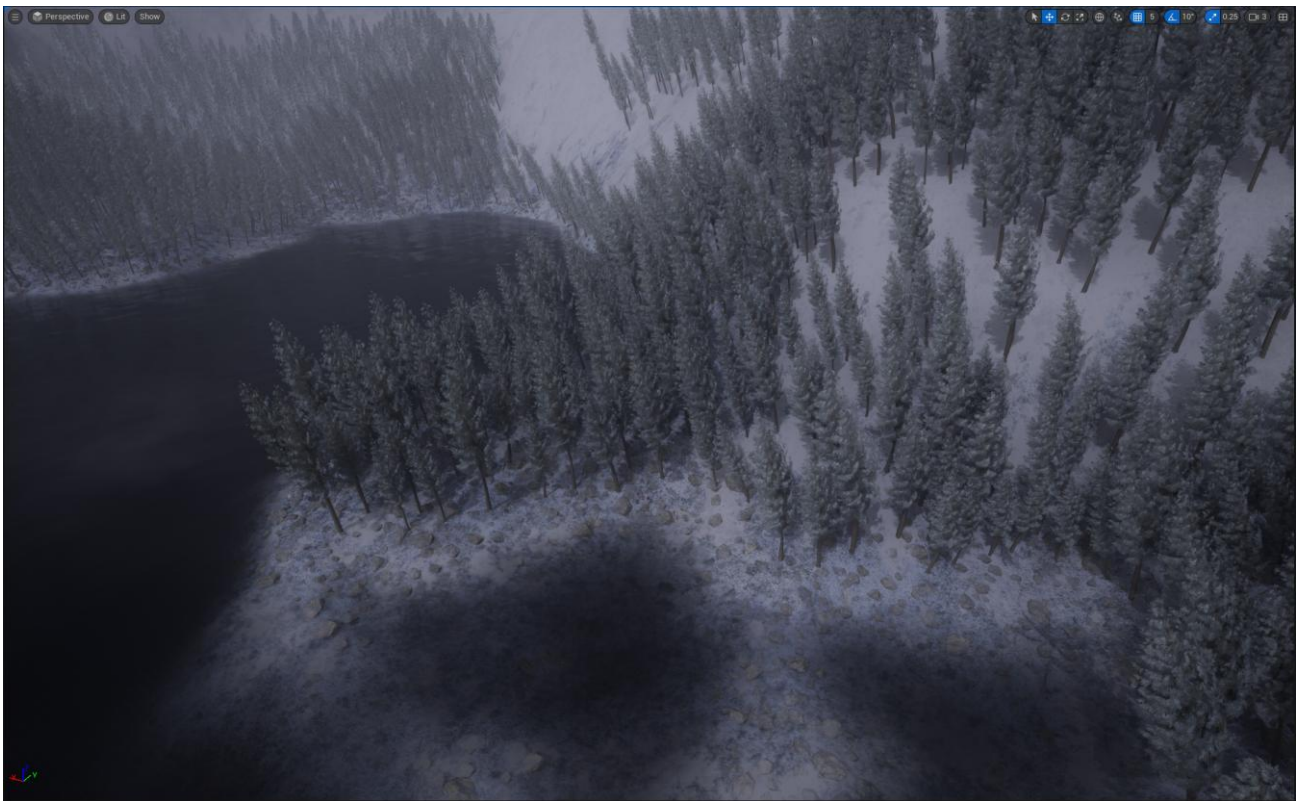


Рис. 6 «Деревья и камни, размещенные с помощью PFV»

3.3 Нанесение текстур на модели камней и деревьев

Моделям деревьев и камней также нужно присвоить заснеженные текстуры, чтобы они вписывались в общую арктическую атмосферу.

В Unreal Engine система материалов организована с использованием концепции родительских и дочерних материалов, что позволяет эффективно управлять и переиспользовать материалы в проекте.

Что касается деревьев – необходимо подняться на уровень **родительского материала**, изменив при этом летние текстуры на зимние.

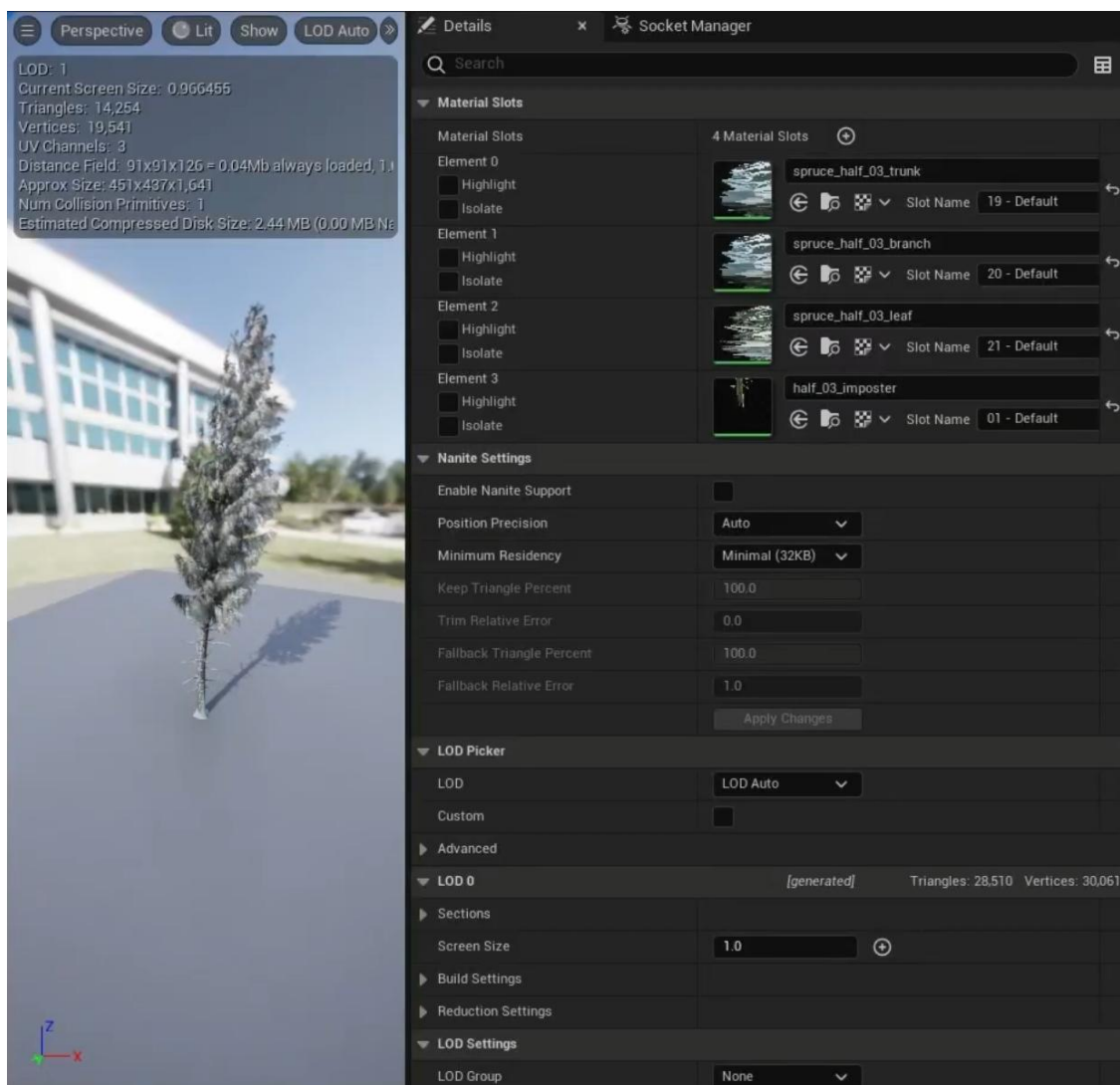


Рис. 7 «Окно деталей материала деревьев»

Использование родительских и дочерних материалов — это мощный инструмент для художников и разработчиков, позволяющий создавать сложные и разнообразные визуальные эффекты в Unreal Engine.

RVT (Runtime Virtual Textures) в Unreal Engine — это система, которая позволяет эффективно управлять текстурами и их отображением в реальном времени, улучшая производительность и качество графики в играх и приложениях. Основная идея RVT заключается в том, чтобы объединить текстуры в виртуальные текстуры, которые могут динамически обновляться и использоваться для рендеринга.

Для изменения текстур камней как раз используется **Runtime Virtual Textures**, которая позволяет настроить внешний вид текстур с помощью графа.

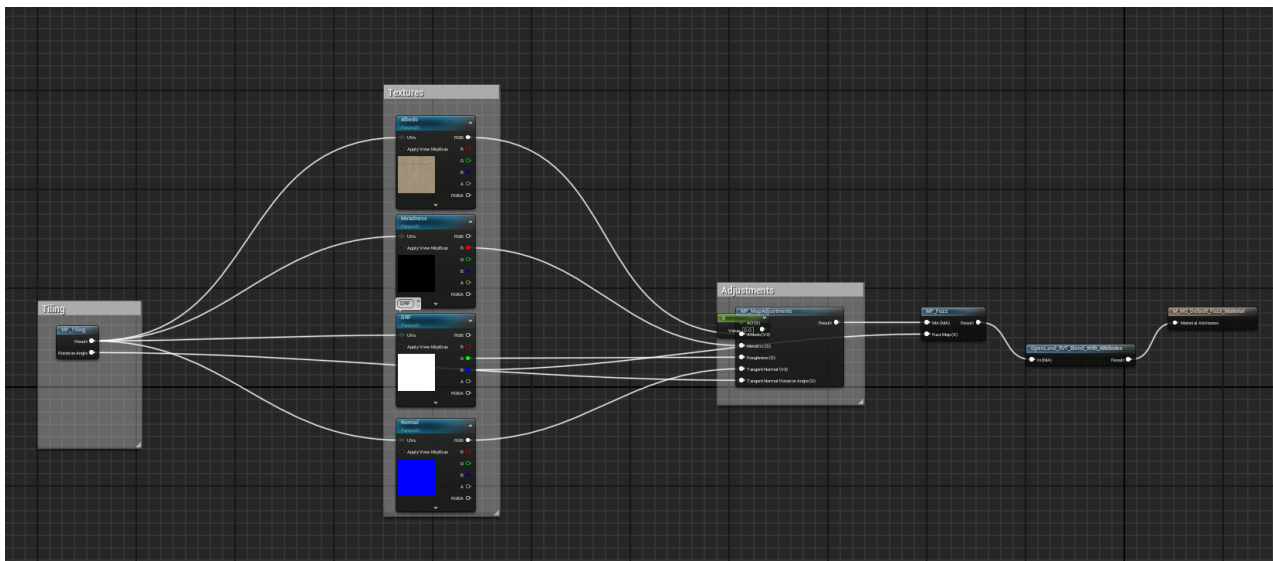


Рис. 8 «Граф RVT для изменения текстур камней»

Для использования RVT в Unreal Engine необходимо создать RVT-активы, настроить их в материалах и применить к объектам в сцене. В итоге это позволило значительно улучшить визуальное качество и общую производительность сцены.

4. Настройка отражений и тумана

4.1 Создание отражений на воде

В Unreal Engine существует несколько **методов отражения**, каждый из которых имеет свои особенности, преимущества и недостатки. Можно выделить три основных метода: Lumen, Screen Space Reflections (SSR) и Ray Tracing.

1. Lumen

Lumen — это система глобального освещения и отражений, представленная в Unreal Engine 5. Она предназначена для работы в реальном времени и обеспечивает высокое качество освещения и отражений.

Преимущества:

- **Динамическое освещение:** Lumen поддерживает динамическое освещение и может адаптироваться к изменениям в сцене, что делает его идеальным для игр с открытым миром.
- **Качество:** обеспечивает реалистичные отражения и освещение, включая сложные взаимодействия света с поверхностями.
- **Простота использования:** Lumen интегрирован в движок и не требует сложной настройки, что упрощает процесс разработки.

Недостатки:

- **Производительность:** может быть более требовательным к ресурсам по сравнению с другими методами, особенно на менее мощных системах.
- **Ограничения:** в некоторых случаях может не обеспечивать такое же качество отражений, как трассировка лучей.

2. Screen Space Reflections (SSR)

SSR — это метод, который использует информацию о пикселях, уже отрендеренных на экране, для создания отражений. Он работает только с теми объектами, которые видны в текущем кадре.

Преимущества:

- **Производительность:** SSR обычно менее требователен к ресурсам по сравнению с трассировкой лучей, что делает его подходящим для игр с высокими требованиями к производительности.
- **Простота:** Легко интегрируется в существующие материалы и не требует сложной настройки.

Недостатки:

- **Ограничения по видимости:** SSR не может отражать объекты, которые не находятся в поле зрения камеры, что может привести к артефактам и недостаткам в отражениях.
- **Качество:** Качество отражений может быть ниже, чем у других методов, особенно в сложных сценах.

3. Ray Tracing

Трассировка лучей — это метод, который использует физические модели света для создания реалистичных отражений, теней и освещения. Этот метод требует поддержки аппаратного обеспечения, такого как NVIDIA RTX.

Преимущества:

- **Высокое качество:** обеспечивает реалистичные отражения, включая отражения от прозрачных и полупрозрачных материалов, а также сложные эффекты, такие как глобальное освещение.

- **Точная симуляция:** позволяет точно моделировать поведение света, что делает его идеальным для фотореалистичных сцен.

Недостатки:

- **Производительность:** трассировка лучей требует значительных вычислительных ресурсов и может заметно снизить производительность, особенно на старых или менее мощных системах.
- **Сложность настройки:** настройка трассировки лучей может быть более сложной и требовать дополнительных знаний.



Рис. 9 «Отражения в воде, созданные методом Ray Tracing»

Для создания отражений в воде был использован метод **Ray Tracing**, так как он делает отражения максимально реалистичными и саму сцену более кинематографичной.

4.2 Создание и настройка тумана

В Unreal Engine существуют два основных типа тумана: плоский и объемный. Каждый из этих типов тумана имеет свои особенности, применения и визуальные эффекты.

Плоский туман (2D Fog)

Плоский туман представляет собой эффект, который накладывается на сцену и создает иллюзию тумана или дыма, но не взаимодействует с геометрией в 3D-пространстве. Он обычно используется для создания атмосферы и улучшения визуального восприятия.

Преимущества:

- **Производительность:** плоский туман менее требователен к ресурсам, так как он не требует сложных расчетов для взаимодействия с геометрией.
- **Простота настройки:** легко настраивается и может быть быстро добавлен в сцену для создания эффекта тумана.

Недостатки:

- **Ограниченная реалистичность:** плоский туман не взаимодействует с объектами в сцене, что может привести к менее реалистичному восприятию, особенно в сложных сценах.
- **Отсутствие глубины:** плоский туман не создает эффекта глубины, который можно было бы ожидать от настоящего тумана.

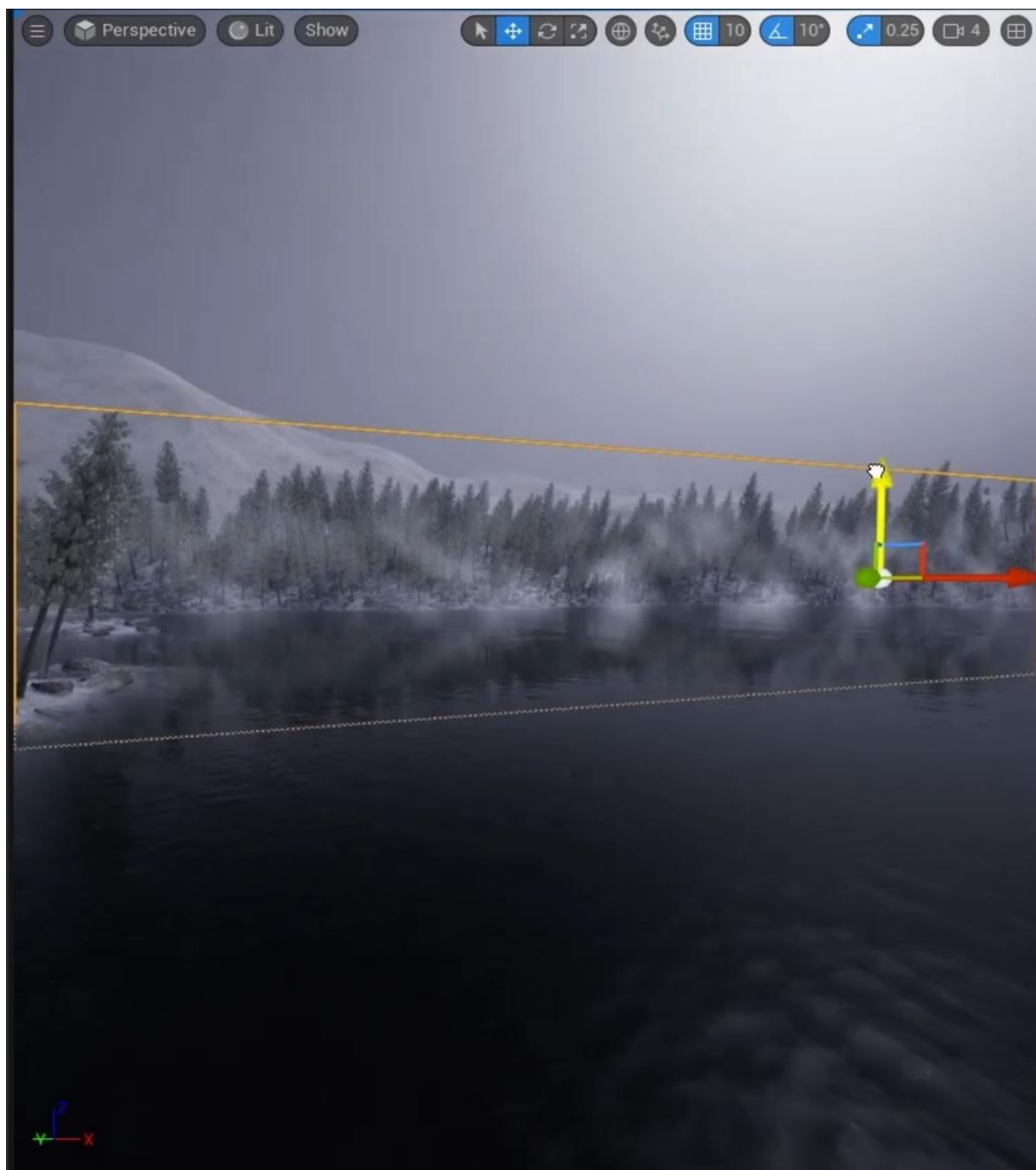


Рис. 10 «Плоский туман в Unreal Engine»

Объемный туман (3D Fog)

Объемный туман создает эффект тумана, который взаимодействует с геометрией в 3D-пространстве. Он может заполнять пространство между объектами и создавать более реалистичное восприятие глубины и атмосферы.

Преимущества:

- **Реалистичность:** объемный туман создает более правдоподобные эффекты, так как он взаимодействует с освещением и геометрией, создавая ощущение глубины.
- **Динамика:** объемный туман может изменяться в зависимости от положения камеры и объектов в сцене, что добавляет динамичности.

Недостатки:

- **Производительность:** объемный туман более требователен к ресурсам, так как требует сложных расчетов для рендеринга.
- **Сложность настройки:** настройка объемного тумана может быть более сложной и требовать больше времени для достижения желаемого эффекта.



Рис. 11 «Объемный туман в Unreal Engine»

5. Создание палаточного лагеря и места крушения

5.1 Создание палаточного лагеря

В сцене необходимо разбавить естественную природу несколькими искусственными объектами. Для создания палаточного лагеря использовались материалы из **Quixel Bridge**, такие как палета, фургон, бочка, коробка и брезент.

Большинство моделей огня сильно **нагружают сцену** вследствие огромного количества частиц самого пламени и дыма. Чтобы число кадров не сильно падало, был использован оптимизированный огонь из **UE Starter Kit**.



Рис. 12 «Сцена палаточного лагеря»

5.2 Создание места крушения

Для создания места крушения использовалось все из вышеописанного: **Quixel Bridge** для модели самолета, **ProceduralFoliageVolume** для камней под самолетом, огонь из **UE Starter Kit**.

Новое в этой сцене – **снежная буря**. Она была создана с помощью плагина Niagara, а также отредактирована в разделе графа материала.



Рис. 13 «Сцена крушения самолета и снежной бури»

ЗАКЛЮЧЕНИЕ

Создание интерактивной среды в Unreal Engine 5 — это многогранный процесс, который требует сочетания художественных и технических навыков. На каждом этапе, от проектирования до программирования, разработчики сталкиваются с уникальными вызовами, которые требуют креативного подхода и глубокого понимания инструментов движка.

Unreal Engine 5 предоставляет разработчикам мощные инструменты, такие как Nanite и Lumen, которые позволяют создавать высококачественные и реалистичные миры. Использование Blueprints упрощает процесс программирования, позволяя сосредоточиться на создании увлекательного игрового опыта.

В ходе проекта были выполнены все поставленные цели и задачи:

- Исследовано создание поверхности (на примере программы TerreSculptor);
- Проведена работа по настройке материалов каждого элемента в сцене;
- Проанализировано влияние различных объектов (плоский и объемный туманы, различные виды огня, разные методы отражений) на производительность сцены;
- Исследованы важнейшие виджеты, встроенные в Unreal Engine: инструмент PFV и технология RVT.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Unreal Engine 5 Documentation // Unreal Engine Documentation URL:
<https://dev.epicgames.com/community/learning/tutorials/DYE1/unreal-engine-5-1-unreal-engine-5>. Дата обращения: 14.09.2024;
2. Introduction to Materials // Unreal Engine Documentation URL:
<https://dev.epicgames.com/community/learning/tutorials/9d0a/unreal-engine-introduction-to-materials>. Дата обращения: 01.11.2024;
3. Procedural Foliage Tool // Unreal Engine Documentation URL:
<https://dev.epicgames.com/documentation/en-us/unreal-engine/procedural-foliage-tool-in-unreal-engine>. Дата обращения: 22.10.2024;
4. Visual Effects in Niagara for Unreal Engine 5 // Unreal Engine Documentation URL:
<https://dev.epicgames.com/documentation/en-us/unreal-engine/creating-visual-effects-in-niagara-for-unreal-engine>. Дата обращения: 12.01.2025;
5. Geometry instancing // Wikipedia, the free encyclopedia URL:
https://en.wikipedia.org/wiki/Geometry_instancing. Дата обращения: 17.10.2024;
6. Quixel Bridge // Quixel Bridge market URL:
<https://quixel.com/bridge>. Дата обращения: 19.12.2024;
7. Unreal Engine 5 Documentation // Unreal Engine Documentation URL:
<https://docs.unrealengine.com/>. Дата обращения: 10.09.2024;