



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»

КАФЕДРА «Системы автоматизированного проектирования (РК-6)»

## РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

*НА ТЕМУ:*

«Разработка ландшафта и 3d-объектов для Real Time  
Strategy окружения на Unreal Engine 5»

Добавлено примечание ([1]): Не уверен что такое  
название темы пойдёт.

Студент РК6-82Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Самарин С.Д.  
(Фамилия И.О.)

Руководитель ВКР

\_\_\_\_\_  
(Подпись, дата)

Витюков Ф.А.  
(Фамилия И.О.)

Нормоконтролёр

\_\_\_\_\_  
(Подпись, дата)

Грошев С.В.  
(Фамилия И.О.)

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  
(Индекс)  
Карпенко А.П.  
(Фамилия И.О.)  
«15» февраля 2024 г.

## ЗАДАНИЕ на выполнение выпускной квалификационной работы бакалавра

Студент группы РК6-82Б

Самарин Савва Денисович  
(фамилия, имя, отчество)

Тема квалификационной работы: «Разработка ландшафта и 3d-объектов для RTS окружения на Unreal Engine 5»

Источник тематики (НИР кафедры, заказ организаций и т.п.): кафедра.

Тема квалификационной работы утверждена распоряжением по факультету РК №\_\_ от «\_\_»  
2024 г.

### Техническое задание

#### Часть 1. Аналитическая часть

Разобрать полный цикл создания статических и анимированных 3d-моделей и сцен. Изучить материалы и средства разработки, предоставляемые трехмерным движком Unreal Engine 5. Исследовать и освоить методы моделирования ландшафта и принципы наложения ландшафтных материалов. Провести анализ экспортации объектов и проблем, возникающих при интеграции в среде Unreal Engine 5. Оптимизировать параметры отрисовки ландшафта.

#### Часть 2. Практическая часть 1. Создание ландшафта.

Создать ландшафт пользуясь инструментами программного обеспечения World Creator 3. Экспортировать и интегрировать ландшафт в среду 3d-движка Unreal Engine 5. Добавить материалы ландшафта. Настроить освещение. Добавить простейшие объекты имитирующие водную поверхность. Создать элементы окружения и растительности.

#### Часть 3. Практическая часть 2. 3d-моделирование.

Создать сцены и 3d-модели объектов наполнения, пройдя полный цикл создания статических моделей. Добавить анимацию объектов в сцене. Создать и настроить частицы шерсти. Выполнить экспорт ассетов и их импорт в движок Unreal Engine 5.

**Часть 4. Практическая часть 3. Создание RTS окружения.**

Настроить навигационную сетку и отладить возможность перемещения контролируемых объектов по созданной карте. Создать модели юнитов и строений. Добавить возможность пользовательского контроля объектов, таких как перемещение в указанное место и взаимодействие выделенных объектов с указанными, такие как сбор ресурсов, атака цели или починка строения.

**Добавлено примечание ([12]):** Не уверен в необходимости добавления Части 4, однако предыдущие пункты не умещают предполагаемой последующей работы.

Если так пойдёт, то нужно это также добавить ниже в таблицу с Календарным Планом?

#### Оформление выпускной квалификационной работы

Расчетно-пояснительная записка на ?? листах формата А4.

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

Работа содержит 8 графических листов формата А4.

Дата выдачи задания: «10» февраля 2024 г.

В соответствии с учебным планом выпускную квалификационную работу выполнить в полном объеме в срок до «11» июня 2024 г.

Руководитель квалификационной работы \_\_\_\_\_  
(Подпись, дата)

**Витюков Ф.А.**  
(Фамилия И.О.)

Студент \_\_\_\_\_  
(Подпись, дата)

**Самарин С.Д.**  
(Фамилия И.О.)

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

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 РК УТВЕРЖДАЮ  
КАФЕДРА РК6 Заведующий кафедрой РК6  
(Индекс)  
ГРУППА РК6-82Б Карпенко А.П.  
(Фамилия И.О.)  
«15» февраля 2024 г.

**КАЛЕНДАРНЫЙ ПЛАН  
выполнения выпускной квалификационной работы**

Студент группы РК6-82Б

Самарин Савва Денисович  
(фамилия, имя, отчество)

Тема квалификационной работы: «Разработка ландшафта и 3d-объектов для RTS окружения на Unreal Engine 5»

№ п/п	Наименование этапов выпускной квалификационной работы	Сроки выполнения этапов		Отметка о выполнении	
		план	факт	Должность	ФИО, подпись
1.	Задание на выполнение работы. Формулирование проблемы, цели и задач работы	10.02.2024	_____	Руководитель ВКР	Витюков Ф.А.
2.	1 часть. Аналитическая часть	18.02.2024	_____	Руководитель ВКР	Витюков Ф.А.
3.	Утверждение окончательных формулировок решаемой проблемы, цели работы и перечня задач	28.02.2024	_____	Заведующий кафедрой	Карпенко А.П.
4.	2 часть. Практическая часть 1	21.04.2024	_____	Руководитель ВКР	Витюков Ф.А.
5.	3 часть. Практическая часть 2	23.05.2024	_____	Руководитель ВКР	Витюков Ф.А.
6.	1-я редакция работы	28.05.2024	_____	Руководитель ВКР	Витюков Ф.А.
7.	Подготовка доклада и презентации	04.06.2024	_____	Студент	Самарин С.Д.
8.	Заключение руководителя	10.06.2024	_____	Руководитель ВКР	Витюков Ф.А.
9.	Допуск работы к защите на ГЭК (нормоконтроль)	17.06.2024	_____	Нормоконтролер	Грошев С.В.
10.	Внешняя рецензия	17.06.2024	_____		
11.	Защита работы на ГЭК	21.06.2024	_____		

Студент Самарин С.Д.  
(подпись, дата) Руководитель ВКР Витюков Ф.А.  
(подпись, дата) (Фамилия И.О.) (Фамилия И.О.)

## **АННОТАЦИЯ**

Работа направлена на изучение и создание RTS (Real Time Strategy) окружения.

В проекте рассмотрено создание карты ландшафта, материалов визуального наполнения местности, RTS камеры с ограничением возможности изменения угла обзора и направления перемещения, объектов растительности и сетки навигации NavMesh, по которой осуществляется возможность передвижения актёров.

Для наполнения игрового окружения были смоделированы и экспортированы 3d-объекты с использованием редактора Blender.

После создания наполнения были настроены элементы пользовательского управления и соответствующего RTS проекту обеспечения, такие как наличие столкновения объектов и возможность навигации при помощи системы NavMesh.

Также в процессе реализации некоторых этапов были произведена оптимизация для улучшения производительности и ресурсоёмкости проекта.

Тип работы: выпускная квалификационная работа.

Тема работы: «Разработка ландшафта и 3d-объектов для RTS окружения на Unreal Engine 5».

Объекты исследований: 3d-моделирование, разработка внутриигровых систем, повышение производительности, разработка RTS окружения.

## **ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

**Game engine (игровой движок)** – набор ключевых компонентов программного обеспечения, используемых для разработки игр и иных 3d-приложений. Как правило, инструменты движка абстрагированы от специфики конкретной игры, но могут учитывать некоторые особенности жанра – они представляют «базис» для разработки, «надстройку» над которым создает его пользователь.

**Unreal Engine 5 (UE5)** – игровой движок, разрабатываемый и поддерживаемый компанией Epic Games.

**3d-model (3d-модель)** – математическое представление объекта в трехмерном пространстве.

**3d-modeling (3d-моделирование)** – процесс создания 3d-модели объекта.

**3d hard-surface modeling (3d-моделирование твердых поверхностей)** – процесс создания 3d-моделей объектов, имеющих гладкие поверхности и острые углы. Как правило, в данную категорию входят неживые объекты, например: механизмы, машины, здания, мебель.

**3d organic modeling (3d-моделирование органических объектов)** – процесс создания 3d-моделей объектов живой природы и объектов, имеющих похожие на них формы. В данную категорию входят такие объекты моделирования, как люди, животные, гуманоиды, растения.

**Текстурирование** – процесс создания текстур объекта.

**High-poly модель (высокополигональная модель)** – максимально детализированная версия 3d-модели. Имеет большое количество полигонов (от нескольких сотен тысяч до миллионов), в основном используется для дальнейшего запекания текстур. Как правило, высокополигональные модели не используются при отрисовке в реальном времени, поскольку для их обработки требуется слишком много вычислительных ресурсов.

**Low-poly модель (низкополигональная модель)** – модель, содержащая относительно низкое количество полигонов (несколько тысяч), при этом сохраняющая основные геометрические свойства объекта. Низкополигональные

модели широко используются при отрисовке в реальном времени, поскольку затраты на их обработку приемлемы для получения достаточно плавного изображения.

**Actor (актёр)** – в рамках движка UE4 любой объект, который может быть размещен на уровне. Базовый класс всех actor'ов – AActor.

**Actor Component** – специальный тип объекта, который может быть присоединен к выбранному actor'у как подобъект (subobject). Как правило, используется для внедрения функциональности, общей для различных actor'ов. Базовый класс – UActorComponent.

**Transform (трансформ)** – данные о местоположении, повороте и масштабе объекта. Представляются матрицей преобразований.

**Scene Component (USceneComponent)** – класс, производный от UActorComponent. Представляет собой компонент, который может иметь свой трансформ на сцене. Данный компонент используется для внедрения функциональности, не требующей геометрического представления.

**Полигональная сетка** – набор вершин, рёбер и граней, определяющий внешний вид многогранного объекта.

**Полигон** – многоугольник, являющийся гранью или набором граней 3d-сетки. Основные типы: треугольник (tri), четырёхугольник (quad) и n-gon (5 или более вершин).

**Sculpting (скульпинг)** – процесс создания и детализации 3d-моделей с помощью 3d-кистей.

**Rendering (рендеринг, отрисовка)** – процесс получения 2d-изображения по имеющейся 3d-модели.

**Polycount** – количество полигонов модели.

**Static Mesh (UStaticMesh)** – класс, представляющий собой статический геометрический объект. Хранит данные о полигональной сетке модели.

**Текстура** – изображение, накладываемое на поверхность 3d-модели. Может содержать одно или несколько свойств поверхности, например: цвет,

жёсткость (roughness), смещение (displacement), направление нормалей (normal map), и т.д.

**Материал** – набор свойств, определяющих поведение света при отражении от поверхности. Материалы также могут использовать одну или несколько текстур.

**Material (UMaterial)** – класс, позволяющий хранить и модифицировать информацию о материале.

**Static Mesh Component (UStaticMeshComponent)** – компонент, который может иметь статический меш и набор материалов, применимых к нему.

**Geometry instancing (инстансинг, дублирование геометрии)** – техника, позволяющая отрисовывать множество однотипных элементов за один проход.

**Instanced Static Mesh Component (UInstancedStaticMeshComponent, ISMC)** – класс, производный от UStaticMeshComponent, позволяющий использовать механизм инстансинга геометрии.

**LOD (Level of Detail, уровни детализации)** – техника, позволяющая подменять разные по детализации версии модели в зависимости от дистанции между камерой и объектом, либо в зависимости от процента площади экрана, занимаемой моделью.

**Ретопология** – процесс изменения топологии полигональной сетки модели с целью ее упрощения/улучшения.

**UV-unwrapping (UV-развертка)** – процесс создания развертки 3d-модели. Представляет собой процесс получения соответствия между координатами поверхности 3d-модели и координатами на текстуре (U по горизонтали, V по вертикали).

**Texture baking (запекание текстур)** – процесс переноса деталей поверхности high-poly модели на набор текстур.

**Hierarchical Instanced Static Mesh Component (HISMC, UHierarchicalInstancedStaticMeshComponent)** – класс, производный от UInstancedStaticMeshComponent, позволяющий использовать LOD.

**Central Processing Unit (CPU)** – центральный процессор.

**Graphics Processing Unit (GPU)** – графический процессор (видеокарта).

**Frames per second (FPS)** – количество кадров в секунду.

**Rendering Hardware Interface (RHI)** – в UE4 надстройка над множеством графических API (например: DirectX, OpenGL, Vulkan), позволяющая писать независимый от платформы код.

**Asset (аксет)** – в контексте компьютерных игр, объект, представляющий собой единицу контента. Игровыми аксессуарами являются 3d-модели, текстуры, материалы, аудиофайлы, и т.д. Как правило, созданием аксессуаров занимаются художники, дизайнеры, музыканты.

**Reference (референс)** – изображение, используемое художником в процессе 3d-моделирования для получения дополнительной информации о моделируемом объекте.

## **СОДЕРЖАНИЕ**

<b>ВВЕДЕНИЕ.....</b>	<b>11</b>
1. Создание ландшафта.....	17
1.1. Разработка каркаса ландшафта.....	17
1.2. Настройка материалов.....	20
1.3. Реализация импорта в Unreal Engine .....	24
1.4. Оптимизация сетки ландшафта .....	30
1.5. Добавление растительности и воды.....	34
1.6. Создание сетки NavMesh и добавление RTS камеры .....	38
2. 3d-моделирование .....	40
1.1. Обзор интерфейса.....	40
1.2. Базовые методы моделирования .....	43
1.3. Методы полигонального моделирования объектов .....	46
1.4. Методы «скульптинга» .....	54
1.5. Экспорт и импорт .....	61
3. Создание RTS окружения.....	63
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>64</b>
<b>СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....</b>	<b>65</b>

## ВВЕДЕНИЕ

При создании пользовательских интерфейсов одной из главных задач является достижение удобства потенциального пользователя при взаимодействии с конечным продуктом. Оно, в свою очередь, состоит из многих аспектов. В контексте компьютерных игр можно выделить такие основные аспекты удобства, как простота управления, интуитивность дизайна, комфортная производительность.

Для реализации простой и понятной системы управления необходимо чётко понимать специфику продукта, продумывать различные сценарии использования, исследовать предметную область и рынок, анализировать качество уже придуманных решений. Для достижения комфортной для пользователя производительности важно применять различные способы оптимизации на всех этапах разработки.

В данной работе рассматривается создание и разработка элементов пользовательского интерфейса для компьютерной коллекционной карточной игры. Работа делится на три основные части:

1. создание ландшафта;
2. создание 3d-моделей;
3. разработка RTS системы для проекта;

### *Часть первая. Создание ландшафта*

В рамках первой части была реализована следующая задача: разработка проектов жанра RTS в UE. По ходу выполнения промежуточными этапами стали: разработка ландшафта карты в любом подходящем для этого программном обеспечении, добавление и настройка материалов, играющих роль различных поверхностей ландшафта, например, камня или травы, интегрирование созданных материалов и ландшафта в среде Unreal Engine, добавление растительности, воды, света и RTS камеры, оптимизация сетки ландшафта, создание сетки проходимости NavMesh.

Создание карты является основополагающей задачей значительного числа проектов игровой индустрии и симуляций. Ландшафт прежде всего необходим

**Добавлено примечание ([13]):** Эту часть пока не менял, но такое общее вступление подойдёт 90% работ.

для расположения и перемещения объектов по его поверхности, создания определённого рельефа и биомов. Добавление растительности и освещения могут помочь в создании реалистичных пейзажей, добавлении атмосферы или преград.

Альтернативой созданию карт с помощью различных программ является импорт существующих карт при помощи глобальных картографических и навигационных технологий, однако подобных подход может упираться в невозможность получения необходимого результата или трудоёмкости поиска конкретных параметров, что в полной мере осуществимо средствами современных программных инструментов.

Для разработки проекта жанра RTS был выбран графический движок Unreal Engine 5.2, обладающий значительным набором инструментов для разработки 3D-приложений и совместимостей с другими приложениями в данной области. Значительными преимуществами Unreal Engine является полностью открытый исходных код, гарантирующий одновременно свободную доступность программы и полную возможность изучения внутреннего устройства, активное взаимодействие разработчиков с сообществом, обеспечивающее постоянное совершенствование и оптимизацию работы, а также огромный функционал с мировой известностью, которым пользуются ведущие разработчики игровой индустрии, дизайнеры и кинематографы.

Разработка в Unreal Engine 5 грубо делится на 2 взаимодействующие системы: скрипты Blueprint и модифицированный код C++. Каждая из систем обладает своими преимуществами в зависимости от задач. Написание в коде C++ является абсолютно приоритетным для написания логики событий, начиная от реализации математических алгоритмов и заканчивая работой поведения ИИ. Система Blueprint также позволяет значительно проще и без серьёзных знаний программирования реализовывать значительную часть функционала, приоритетной для написания в коде, однако зачастую это сопряжено с худшей производительностью и в целом это является плохим тоном программирования.

Наилучшее применение данной системы — это написание скриптов для задач, которые сложно осуществить в коде, зачастую такие задачи сопряжены с настройкой материалов или некоторых актёров, например, камеры и её управления. В соответствии с поставленными целями всю работу над проектом рационально выполнять при помощи системы Blueprint, не прибегая к использованию кода.

#### *Часть вторая. 3d-моделирование*

Использование игровых движков в современном мире не ограничено созданием исключительно игровых проектов. При помощи функционала Unreal Engine 5 опытный разработчик в состоянии также создавать фотoreалистичные виды, кинематографические видеоматериалы с аутентичным, объёмным звуковым сопровождение, масштабные сцены, учитывающие базовые законы физики и соударения объектов, интерактивные симуляции для имитации обучения на сложной, дорогостоящей технике и многое другое. Однако, подобные задачи нереализуемы без использования различных элементов и 3D-моделей.

Как правило движок оснащён базовым функционалом, позволяющим добавлять простейшие объекты геометрии, камеры, освещения или ландшафта. Но зачастую, для крупных проектов разработчику приходится создавать конкретные сложные модели, построение которых невозможно без навыков 3D-моделирования. Движок Unreal Engine 5 располагает всем необходимым функционалом создания геометрии для объектов и наложения на них соответствующих материалов и текстур, но на практике лучше всего для 3D-моделирования подходят более специализированные программные продукты, такие как Autodesk Maya, Autodesk 3ds Max или Blender. Более того подобные продукты оснащены встроенными методами экспорта, позволяющего непосредственно переносить созданные модели в среду движка Unreal.

В качестве предмета изучения 3D-моделирования была выбрана программа Blender, предоставляющая продвинутый и доступный функционал, с широкой поддержкой передачи данных между Unreal Engine 5.

В качестве практической части научно исследовательской работы были поставлены задачи воссоздания сцен схожих со сценами «Фабрика» и «Енот», которые представлены ниже на рисунках. В процессе работы над данными сценами требуется прибегнуть к двум разным методиками моделирования.

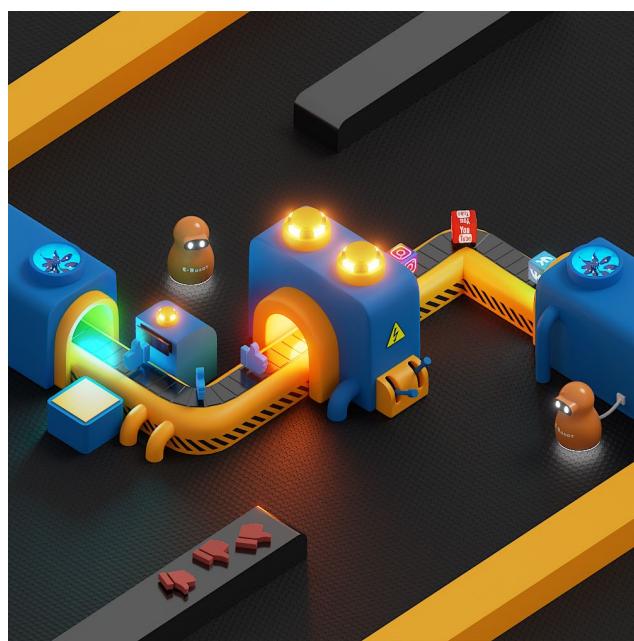


Рисунок 1. Сцена «Фабрика»



Рисунок 2. Сцена «Енот»

В процессе создания данных сцен требуется использовать два разных подхода к моделированию. Первая сцена «Фабрика» в основном состоит из объектов, производных от простых геометрических форм, например, конвейерной ленты, труб, арок и прочего. Для создания данной сцены наиболее рационально использовать базовые методы моделирования, основанные на добавлении модификаторов и ручном редактировании сетки. Вторая сцена «Енот» требует использования такого инструмента как «скульптинг», использующего инструменты работы с сеткой на подобие работы с глиняными скульптурами. Таким образом создание данных двух сцен позволит наиболее полно овладеть разными методиками 3D-моделирования.

### *Часть третья. Создание RTS окружения*



Цели работы:

1. создать, экспорттировать и оптимизировать модели ландшафта в среде Unreal Engine 5;
2. освоить создание 3d-моделей объектов для движка UE5;
3. разработать функционал для RTS окружения проекта.

Для выполнения работы были использованы средства следующих программ:

- Unreal Engine 5 – разработка внутриигровых систем, настройка ландшафта и материалов, интеграция 3d-объектов, создание RTS окружения;
- Blender – создание 3d-моделей объектов;
- World Creator 3 – создание ландшафта и материалов;

**Добавлено примечание ([14]):** Здесь не в полной мере понятно, что писать, поскольку не ясно как много из того что запланировано удастся реализовать в срок и насколько детально.

## 1. Создание ландшафта

### 1.1. Разработка каркаса ландшафта

Для создания RTS карты была составлена приближенная схема расположения значимых объектов и элементов рельефа:

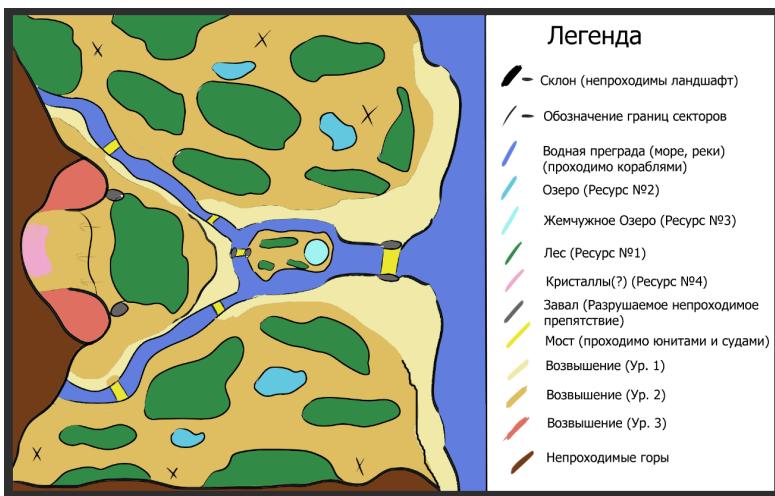


Рисунок 3. Приближенная схема RTS карты

В функционал Unreal Engine 5 входят средства, позволяющие создавать и настраивать ландшафт в самом редакторе, однако стоит отметить что возможности движка в некоторых аспектах серьёзно уступают специализированным программам, таким как Gaea, World Machine, World Creator и прочие. Каждая из вышеперечисленных программ обладает своими особенностями, но в рамках выполнения работы выбор остановился на World Creator 3. Подобные редакторы основывают свою генерацию на широком ряде настраиваемых фильтров, базовых формах и различных модификаторах, позволяющих имитировать различные типы воздействия на местность, накладывать или смешивать между собой разные рельефы, добавлять маски или включать разные случайные формы.

В сравнении с аналогами World Creator 3 обладает рядом функций серьёзно облегчающих тонкую настройку желаемого результата. Самой отличительной чертой является возможность динамического разбиения всего пространства карты на квадратные сегменты указанного числа степени двух. Каждый такой квадрат или кластер квадратов можно вручную делать ниже или выше, регулируя влияние выбранных ячеек на соседние, тем самым допуская теоретически ручное создание любого желаемого ландшафта.

Ранее более тонкую настройку ландшафта можно было осуществлять разве что при помощи наложения масок и подбора необходимых фильтров и модификаций в конкретном локальном месте, что не сопоставимо с трудозатратами при непосредственной манипуляции над сегментами, при том что World Creator 3 не избавился от функций масок. Также в отличии от World Machine и Gaea работает без необходимости указания узлов («нодов») через которые осуществляется всё взаимодействие с ландшафтами. Вместо этого World Creator обладает рядом стандартных параметров настройки и набором добавляемых фильтров.

В процессе создания в первую очередь нужно выбрать стартовый шаблон ландшафта. Каждый из них полностью настраиваем и служит в большей степени как удобный начальный шаг. Далее ландшафт можно изменять, изменения ключи («сиды») генерации, настраивая параметры точности каркасной сетки, а также добавлять различные стили местности и их влияние на общий результат.

Далее после придания необходимой формы накладываются фильтры для добавления эффектов, таких как эрозия разных типов, выравнивание, высушивание, сглаживание острых углов и так далее.

В процессе создания активно использовались фильтры сглаживания пространства, создания «мягких» краёв и эрозии, огрубление обрывов.

Промежуточный результат создания ландшафта без наложения материалов представлен на изображениях ниже:

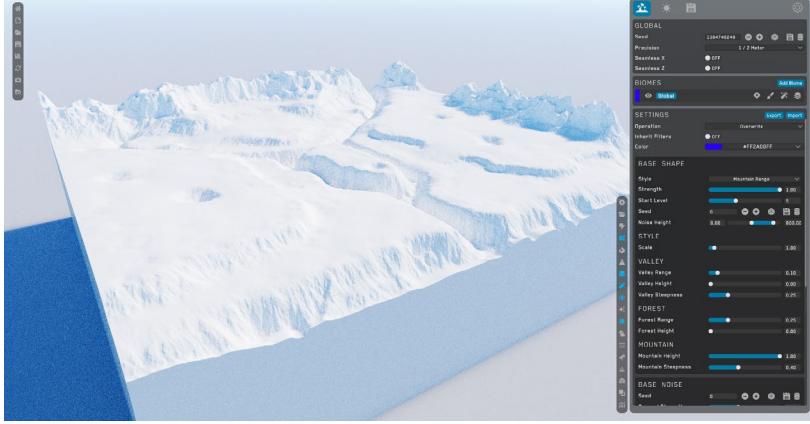


Рисунок 4. Промежуточный ландшафт в World Creator 3, вид сбоку

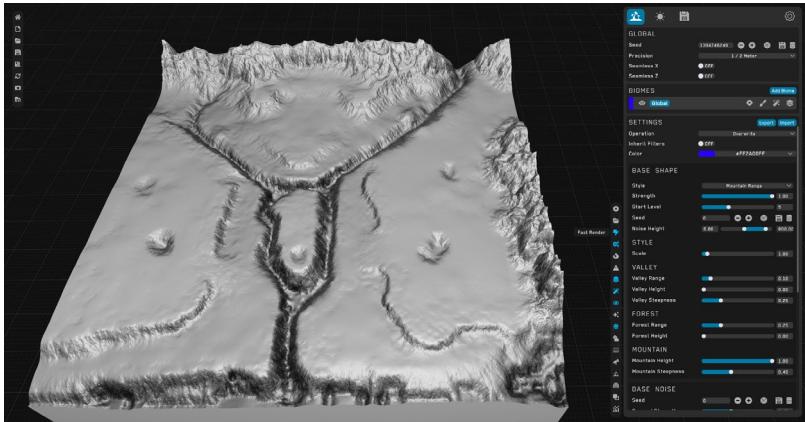


Рисунок 5. Промежуточный ландшафт в World Creator 3, вид сверху под наклоном в фильтре быстрого рендерера

Белый вид местности является стандартным представлением при отсутствии каких-либо покрывающих материалов.

## **1.2. Настройка материалов**

Зачастую добавлять материалы можно и перед полным определением формы ландшафта, поскольку наиболее удобные и универсальные правила, по которым определяется область для используемого материала, будут менять своё положение вместе с изменением рельефа, предоставляя более качественную картину.

Чтобы определить область, к которой будет определён материал необходимо задать ряд правил распределения. Данные правила либо учитывают имеющийся рельеф, либо напрямую исходят из указанной им маски.

Первый способ позволяет учитывать крутизну склонов, впадины, обрывы, холмы и плоскости, ограничивать зоны по высоте или жёсткости, добавление различных случайных «шумов» по всей площади и многое другое. Данный способ наиболее подходит для задания биомов характерных определённом рельефу, например, задание гор, снега, равнинной земли или травы, впадин или морского дна, песка на берегу водоёмов и так далее.

Второй способ с использованием масок может использоваться для всех вышеперечисленных целей, однако, он не универсален и не будет менять положение области каждый раз, когда происходит изменение ландшафта. Данный способ наиболее рационален для выделения нестандартных областей, например, расширение береговой линии в определённых местах, добавления области лесного настила и другие.

Помимо этого, каждое правило распределения обладает набором модификаторов. Они могут влиять на конечный вид области самыми разными способами начиная от добавления случайных шумов по краям и заканчивая полным инвертированием или добавлением масок таких как, например, диаграмма Вороного произвольного дробления.

После установления области необходимо установить параметры материала.

Существуют два основных способа – при помощи градиентов и при помощи текстур.

Для градиентного метода имеется 5 ячеек, каждая из которых может принять градиент цветовой палитры и любую указанную область.

Текстурный метод требует импортировать в программу 5 ключевых компонентов материала поверхности: карту цветов, карту нормалей, карту жёсткости, карту затенений, карту смещений. Одновременным преимуществом и недостатком этого метода перед градиентным является возможность и необходимость точного задания параметров того как материал ляжет на указанную поверхность, в то время как градиентный метод может дать быстрые результаты.

В рамках практики были использован текстурный метод, ресурсы для которого установлены при помощи библиотеки Quixel Megascans, предоставляющей доступ пользователям Unreal Engine к текстурам, считанным с реальных объектов в высоком разрешении. Были добавлены 6 текстур материалов – лесного настила, снега, гор, песка, каменной поверхности и травы. Для добавления текстуры в материал текстурным методом требуется перенести необходимые вышеупомянутые карты в ячейки World Creator 3.

Для создания областей леса была использована пользовательская маска, с размытием по краям. Для снега выбрана определённая высота, случайная девиация по краям (distortion), размытие (blur), случайные шумы (noise offset) и простые потоки (simple flow), симулирующие осадки. Для гор выбрана высота, параметр жёсткости отвечающий за крутизну, простые потоки, размытие и шумы по краям. Для песка выбрана пользовательская маска со случайной девиацией по краям и размытием. Для каменной поверхности выбрана крутизна. Для травы выбран параметр распределения ровности местности. Иерархия наложения в соответствии с порядком перечисления, где лесной настил находится в верху списка и накладывается поверх остальных материалов.

Результат наложения материалов представлен ниже:

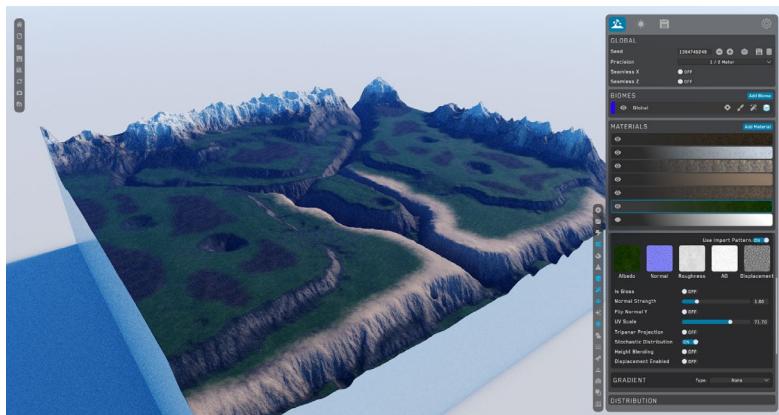


Рисунок 6. Итоговый ландшафт в World Creator 3, вид сбоку



Рисунок 7. Итоговый ландшафт в World Creator 3, вид сверху под наклоном

### **1.3. Реализация импорта в Unreal Engine**

Перед импортом в Unreal Engine требуется определить необходимые для экспорта элементы внутри World Creator 3. World Creator в отличии от World Machine и Gaea не обладает возможностью установления меток экспорта на конкретные узлы за отсутствием таковых в рабочей среде программы. Вместо этого существует отдельная вкладка экспорта, позволяющая экспортировать следующие параметры:

- 1) Цветовая карта (Color Map), содержащая все имеющиеся цвета на ландшафте.
- 2) Карта высот (Height Map), состоящая из монохромного изображения, где степень белого определяет высоту.
- 3) Сетка (Mesh).
- 4) Карта отражаемости (Metalness Map)
- 5) Карта нормалей (Normal Map), цвета на которой отражают различные углы на поверхности ландшафта.
- 6) Карта жёсткости (Roughness Map)
- 7) Splat карта (Splat Map), монохромное изображение отражающую весовую маску, эквивалентную области распределения отдельного материала.

В процессе работы над проектом сначала были экспортированы карта цветов, высот, нормалей и жёсткости, представляющие минимальный необходимый набор для успешного импорта ландшафта в среду Unreal Engine. Однако, в дальнейшем это выявило ряд недостатков, в частности более низкое качество и отсутствие возможности дальнейшей настройки внутри редактора Unreal.

Второй способ включал в себя экспорт карты высот и Splat карт, а также всех материалов, ранее установленных при помощи библиотеки Quixel Megascans.

Для успешного экспорта любых карт в Unreal Engine требуется соблюдать ряд стандартов размерности.

Карта высот должна передаваться в формате raw, глубиной 16 бит, с порядком Little Endian в единичном канале R красного цвета. Также следует включить параметр автоматической нормализации. Поскольку точность ландшафта была изначально задана в 1 квадрат каждые 1/2 метра, общим числом 1024x1024, редактор World Creator 3 автоматически предлагает размер экспортируемой карты 2048x2048. Ближайшая принимаемая величина в Unreal Engine 5 – 2041x2041, которая связана с внутренним устройством движка и по этой причине рекомендуется экспорттировать материалы в соответствующем формате. Также для экспорта Splat Map следует указать Single Channel. Остальные параметры экспорта можно оставить по умолчанию для прочих карт.

После экспорта всех необходимых карт в редакторе Unreal Engine 5 требуется произвести импорт в разделе landscapes в панели Import From File и указать карту высот, после чего в случае верно заданных параметров импорта сгенерируется ландшафт без материалов. Результаты импорта ландшафта в базовый уровень Unreal Engine 5:

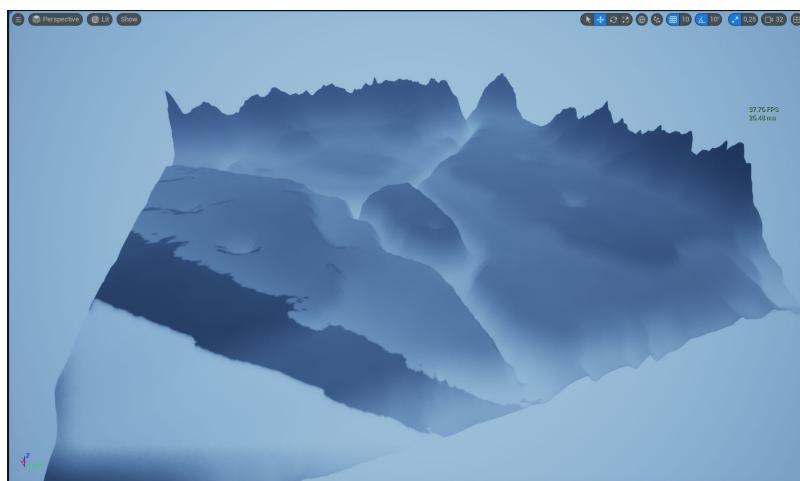


Рисунок 8. Результат импорта карты высот в Unreal Engine

Для создания материала требуется создать Blueprint класс. Согласно конвенции имён в Unreal Engine, файлы материалов должны иметь префикс «M\_». Файл назван M\_Terrain2K.

Чтобы включить все 6 вышеупомянутых текстур в состав материала требуется составить следующую схему в редакторе Blueprint:

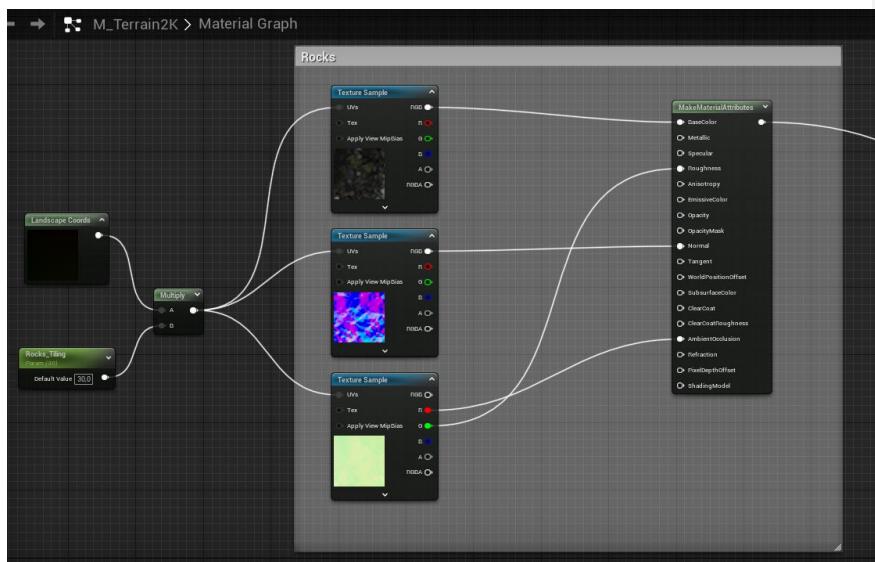


Рисунок 9. Схема включения одной текстуры в рабочее поле системы Blueprint

В данной схеме программа определяет UV координаты ландшафта, которые затем умножаются на изменяемый параметр «тайлинга», ответственного за размер текстур и передаёт эти параметры в 3 необходимых карты имеющегося материала: карту цветов, нормалей и затемнений, которые затем передают результирующие значения в узел создающий понятный для среды набор атрибутов материала. Также в строке Sampler Source для каждой карты необходимо задать параметр Shared Wrap, что позволяет использовать несколько ресурсов одновременно.

После выполнения схожих действий для всех 6 материалов каждому выделяется определённый слой при помощи узла Landscape Layer Blend, с

которым позже можно будет взаимодействовать и настраивать параметры смещивания между собой. Каждому слою были присвоены соответствующие имена на английском языке. Для всех слоёв кроме травы был выбран режим смещивания LB Weight Blend. Для травы – LB Height Blend для предотвращения смещивания слоя травы с слоем каменной поверхности. После этого материалы смешения слоёв передаются выходному узлу материала.

После создания основного класса материала удобно использовать не базовый объект, а создать его инстанцию (material instance), который наследует весь функционал родительского материала, но не оказывает влияние на него при изменении параметров.

Далее инстанция добавляется в соответствующее единственное поле материала ландшафта. Теперь чтобы материал отобразился требуется сгенерировать всю необходимую информацию карты. Для этого в режиме Landscape Mode в разделе Paint для каждого из допустимых 6 слоёв нужно сгенерировать информацию слоя с опцией Weight-Blended Layer. Затем перейдя во вкладку Manage, а затем в Import следует импортировать все Splat карты соответствующих слоёв вместе с картой высот, тем самым заново генерируя ландшафт. Результат импортирования материалов представлен ниже:

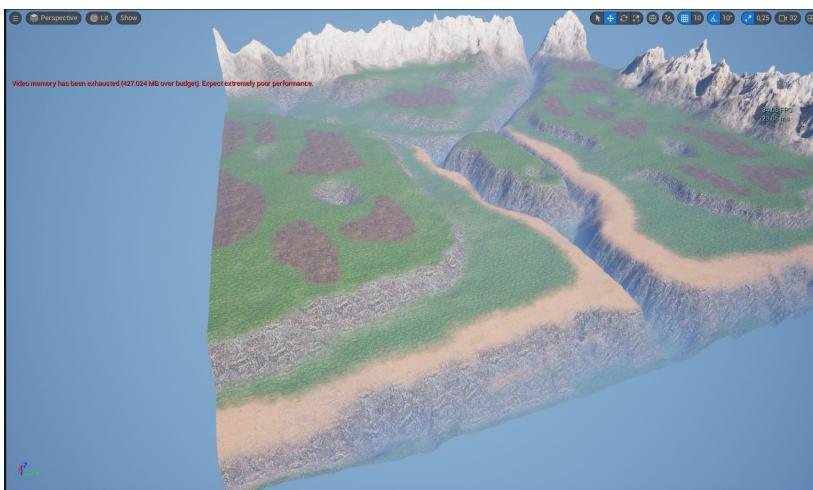


Рисунок 10. Импортированный ландшафт в Unreal Engine 5

На полученном результате можно легко заметить регулярность накладываемых текстур. Существует ряд способов, добавляющих нерегулярности в процесс наложения текстур делая их повторение значительно менее заметным. В процессе работы наиболее оптимальным способом было выбрано добавление схожей текстуры с отличным «тайлингом». Для этого сначала требуется добавить текстуру вышеупомянутым способом через передачу трёх карт и задания атрибутов материала. Далее был добавлен сегмент, отвечающий за шумы, которые будут определять процесс смешивания двух слоёв. После этого два материала смешиваются при помощи звена Blend Material Attributes с использованием маски из составленных ранее шумов. После чего результат смешивания перенаправляется в слой травы. Результаты представлены ниже:

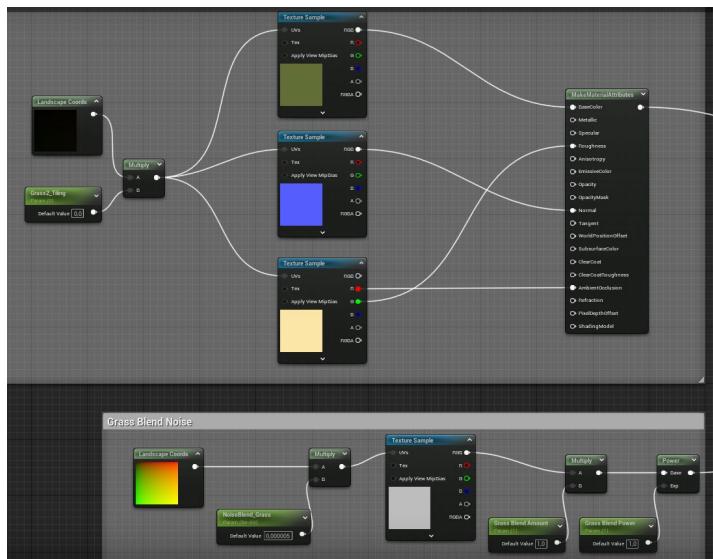


Рисунок 11. Blueprint схема добавления текстуры и шумов

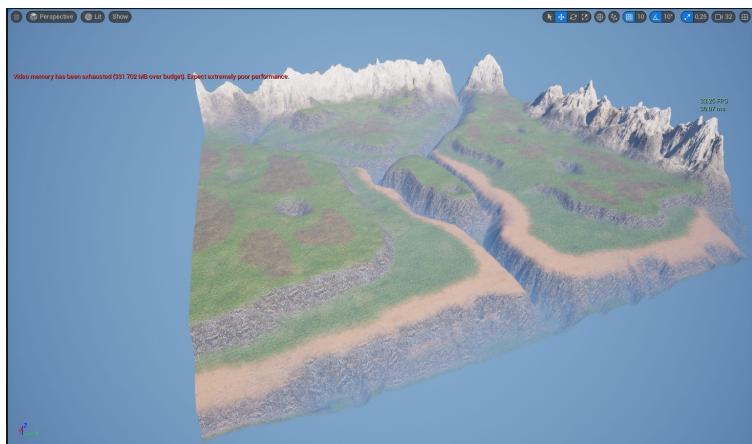


Рисунок 12. Ландшафт после добавления нерегулярностей

#### 1.4. Оптимизация сетки ландшафта

Из вышеперечисленных материалов легко заметить низкие параметры производительности даже при крупных планах. Зачастую показатель FPS (Frames Per Second) находится в диапазоне от 30 до 40. В данном проекте на это в значительной мере влияют показатели GPU (Graphical Processor Unit), освещение и ландшафт.

Основной проблемой, связанной с ландшафтом, является результирующая сетка карты после экспорта. Если переключится в Wireframe mode будет получена следующая картина, указывающая на значительное перенасыщение сетки:

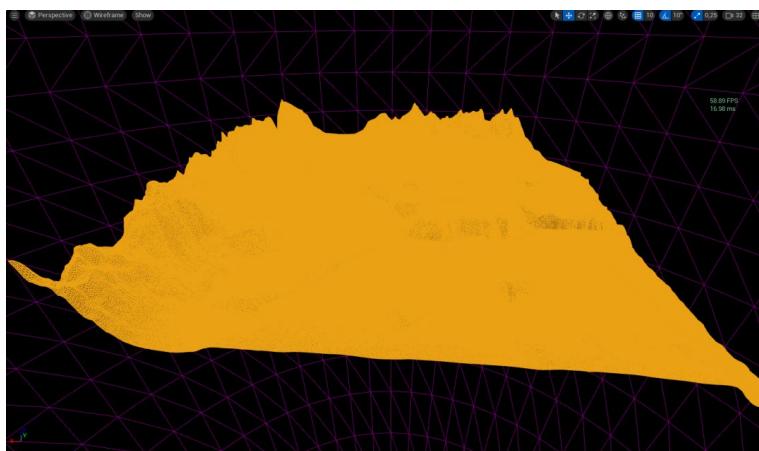


Рисунок 13. Сетка ландшафта до оптимизации, 59 FPS

Подобная детальность является абсолютно излишней и более того не учитывает положение наблюдателя для оптимизации отрисовки при помощи LODов (Levels Of Detail), которые регулируют насколько детальной является сетка, где более высокие уровни отвечают за более дешёвое и грубое распределение. Стоит отметить что использование грубой сетки неминуемо ведёт к созданию более квадратному и примитивному ландшафту, поэтому требуется соблюдать баланс производительности и качества.

По умолчанию редактор Unreal Engine 5 устанавливает параметр максимального числа LODов равное -1, что автоматически допускает использование максимально возможного числа LOD. Однако, остальные параметры по умолчанию устанавливают приоритетным самый низкоуровневый и детальный LOD 0. Были изменены следующие настройки:

- 1) LOD 0 Screen Size: 0,5 → 1,15
- 2) LOD 0: 1,25 → 1,9
- 3) Other LODs: 3,0 → 1,5

Ниже представлены результаты до и после изменения сетки в режиме Unlit с целью минимизировать влияние света на продуктивность в данных тестах:

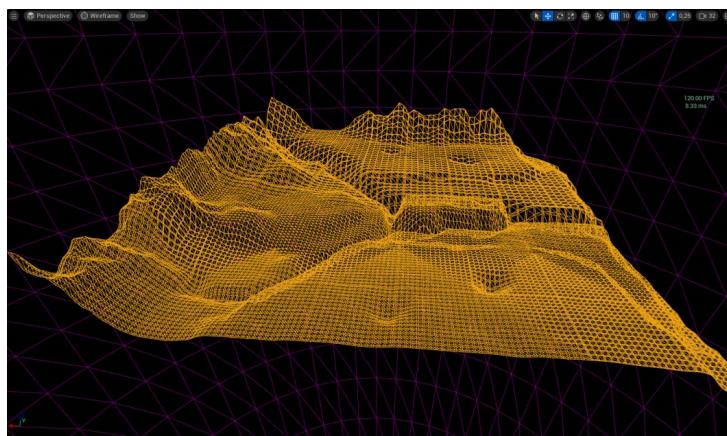


Рисунок 14. Сетка ландшафта после оптимизации, максимальные 120 FPS  
Далее представлен сам ландшафт для сравнения влияния упрощения сетки:

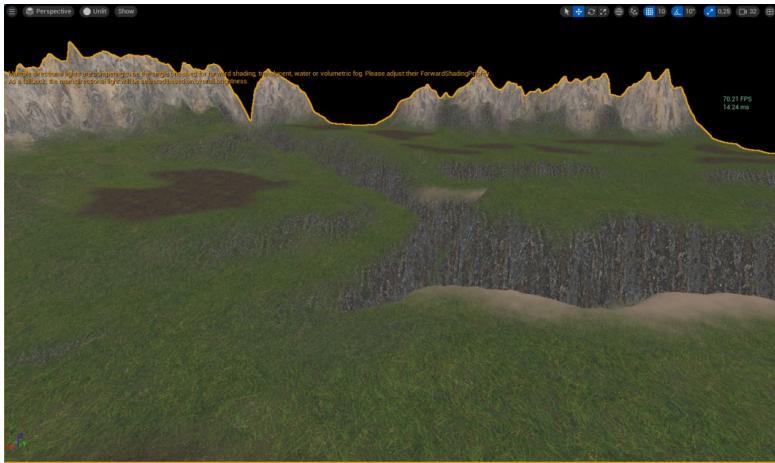


Рисунок 15. Ландшафт до оптимизации, 70 FPS

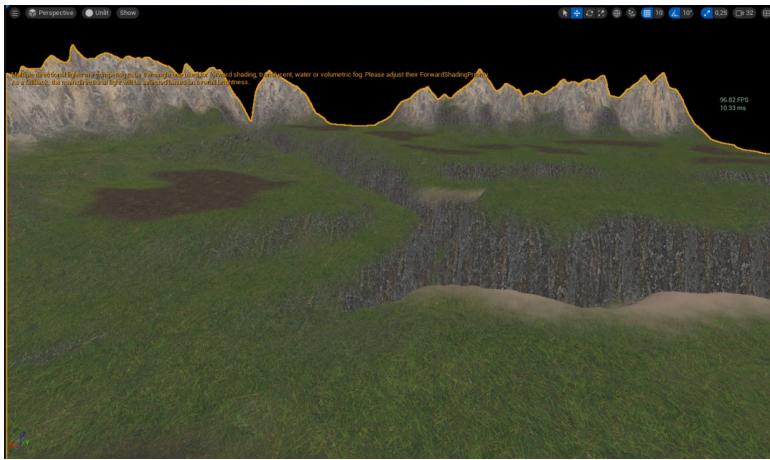


Рисунок 16. Ландшафт после оптимизации, 97 FPS

Как видно влияние на производительность существенная, при визуально  
качество не страдает. Для удобства ниже приведена таблица со сравнением  
производительности до и после оптимизации сетки в разных режимах:

№	Wireframe mode	Unlit mode	Lit mode
1	40→70	66→79	20→25

2	47→96	70→97	23→27
3	59→120	110→120	32→43
4	44→80	69→90	23→26

## 1.5. Добавление растительности и воды

Для водных препятствий в Unreal Engine существует большой набор плагинов. Для создания воды рек и моря был использован стандартный плагин Water, при помощи которого можно поместить плоскость Water Body Custom, имитирующую поверхность воды.

Для создания воды в озёрах были созданы собственные материалы в которых были реализованы возможности по отражению света, перемещению водной глади и имитации прозрачности на глубине. После создания класса Blueprint для материала было создано два алгоритма перемещения двух волн разных по величине, направлению и скорости для предотвращения регулярности. Параметры Water 1 и Water 2 отвечают за величину волн, XSpeed1, XSpeed2 – за скорость волн по направлению X, YSpeed1, YSpeed2 – за скорость волн по направлению Y.

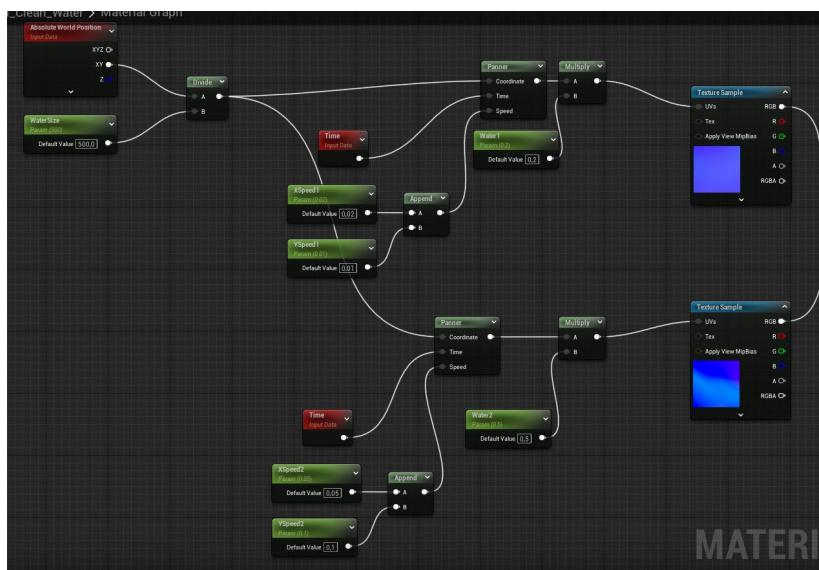


Рисунок 17. Фрагмент схемы Blueprint реализации хождения волн

После добавляется параметры DepthOpacity и FadeDistance, которые затем передаются в узел DepthFade позволяющий установить атрибут прозрачности

воды. Параметры отражаемости и жёсткости также передаются в виде параметров.

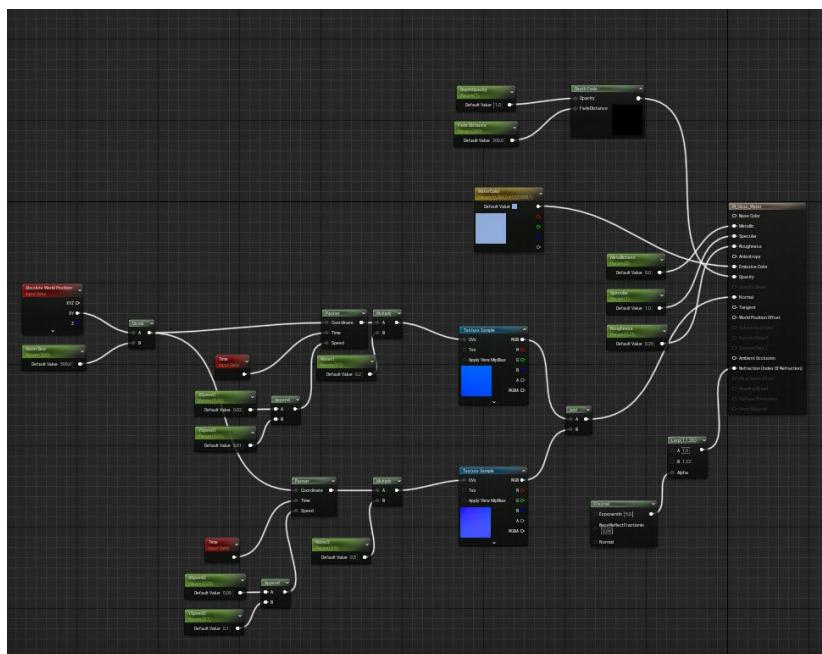


Рисунок 18. Конечная схема Blueprint материалов воды

Стоит также отметить что при установлении какой-либо константы в виде параметра её значение может быть в дальнейшем изменено в дочерних инстанциях материала, не требуя изменения самого материала для точной настройки конкретного объекта общего типа.

В качестве объекта, использующего материал воды, наилучшим образом подходит куб или плоскость. Далее для четырёх внешних озёр были наложены плоскости с одним материалом воды, в то время как для центрального озера был создана собственная инстанция и определены иные параметры для выделения из числа прочих. Результат добавления озёр:



Рисунок 19. Ландшафт после добавления водных объектов

Для добавления растительности в среде Unreal Engine 5 существует собственный режим Foliage, позволяющий насыщать местность любыми объектами растительности, от травы до деревьев. Добавить подобные объекты можно с помощью экспорта из сторонних программных продуктов, позволяющих создавать статическую сетку растительности и прочую информацию необходимую движку, или с помощью существующих наборов и плагинов. Ранее упомянутая библиотека Quixel Megascans обладает рядом наборов доступных для свободного использования, однако для поставленной задачи было принято использовать менее качественные объекты в угоду производительности.

Для помещения растительности необходимо во вкладке Foliage перетащить имеющиеся статические сетки растительности в требуемое поле, а затем настроить масштаб объектов и насыщенность их посадки. Добавление происходит в режиме кисти позволяя в реальном времени добавлять регулируемое количество деревьев необходимых для выполнения задания.

После установки всех необходимых объектов были отключены динамические тени для улучшения производительности.



Рисунок 20. Ландшафт после добавления растительности

## 1.6. Создание сетки NavMesh и добавление RTS камеры

Для создания сетки в среде движка Unreal существует класс объектов NavMesh. Для выполнения задачи создания сетки навигации необходимыми являются актёры NavMeshBoundsVolume, отвечающий за области создания сетки, и NavMeshModifies, позволяющий модифицировать или удалять определённые области существующей сетки.

При добавлении актёра NavMeshBoundsVolume автоматически добавляется его менеджер – RecastNavMesh-Default, в котором имеются все возможности задания параметров, по которым генерируется сетка, указывая предельную высоту двух соседних ячеек сетки, плавность и прочее.

Актёр NavMeshModifies по умолчанию удаляет область сетки проходимости вокруг себя, что удобно использовать для ограничения доступа к объектам типа озёр, леса или определённым местам.

Также существует актёр NavLinkProxy, позволяющий другим движущимся актёрам перепрыгивать между двух указанных точек связи, что может быть реализовано в создании моста для перехода.



Рисунок 21. Результаты создания NavMesh сетки навигации, выделено зелёным

Для добавления RTS камеры был использован плагин Phoenix-Dev. Данный плагин содержит настроенную RTS камеру с функцией плавного перемещения при помощи передвижения указателя мыши или установленных четырёх клавиш, фиксированного вертикального угла обзора, возможностью кругового вращения и изменения расстояния от камеры до поверхности в заданном пределе.



Рисунок 22. Демонстрация положения RTS камеры на максимальном удалении

## 2. 3d-моделирование

### 1.1. Обзор интерфейса

Стандартный интерфейс Blender представляет из себя ряд интерактивных окон. По центру располагается главное окно взаимодействия - Viewport.

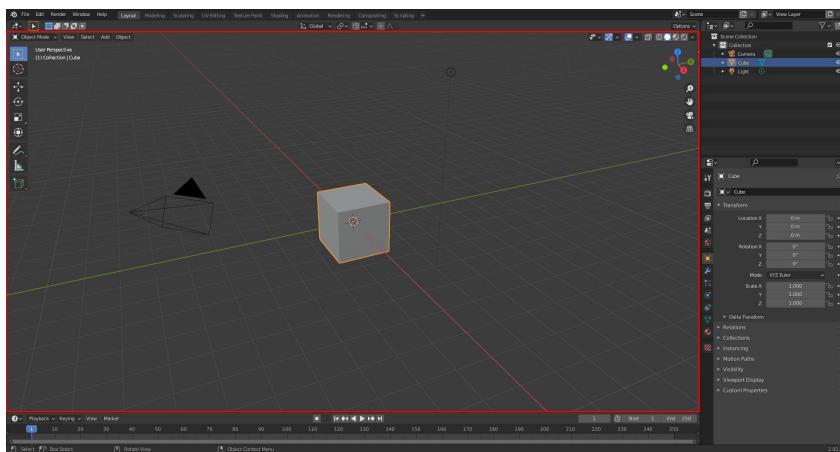


Рисунок 23. Окно Viewport Blender выделено красным

Данное окно предоставляет доступ к результату четырёх режимов отображения:

- 1) Wireframe – режим отображения сетки объектов, в которой видны вертексы и грани построения полигонов.
- 2) Solid – режим отображения твердотельных объектов без освещения и материалов.
- 3) Material Preview – режим отображения материалов, но без корректного рендеринга и освещения.
- 4) Rendered – режим отображения рендеринга во Viewport. Стоит отметить, что по умолчанию в данном режиме есть отличия с конечным рендером, вызываемом опцией Render → Render Image или по нажатию горячей клавиши F12, которые можно настроить в настройках Render Viewport.

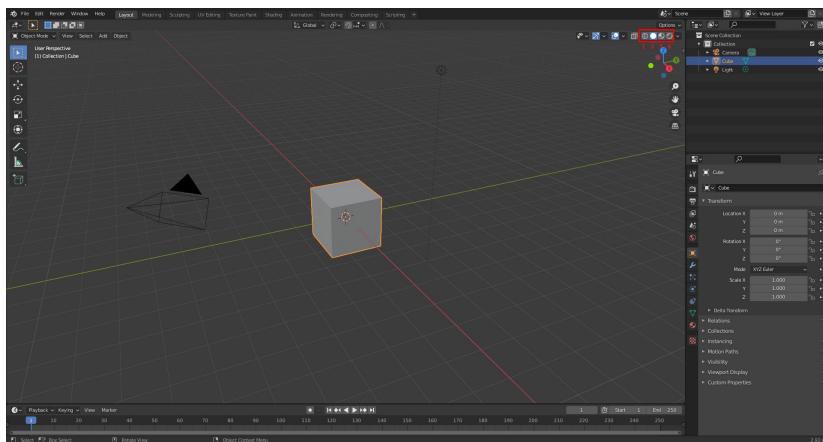


Рисунок 24. Режимы отображения, выделены красным

По умолчанию, помимо четырёх режимов отображения, можно переключаться между десятью режимами рабочей среды. Наиболее значимые из них:

- 1) Layout – стандартный режим взаимодействия с 3D-объектами. В данном режиме удобно создавать базовые геометрические формы и объекты, выделять, перемещать, масштабировать и настраивать группы существующих моделей.
- 2) Modeling – режим непосредственного моделирования сетки выделенных объектов.
- 3) Sculpting – режим «скульптуинга», позволяющий модифицировать сетку объектов по методам схожим с работой с глиной.
- 4) UV Editing – режим изменения и расположения UV-развёртки объектов, которая отвечает за то, какие полигоны будут получать те или иные части накладываемой текстуры.
- 5) Texture Paint – режим создания текстур на самом объекте. В данном режиме представляется возможность рисовать, имитируя кисточку, добавлять смазывания и оттягивания имеющихся цветов текстуры, заливать области и накладывать маски.
- 6) Shading – режим настройки материала для выделенного объекта.

Из важных элементов также стоит отметить наличие навигационного окна в правом верхнем углу. Данное окно предоставляет доступ к объектам, их иерархии, некоторым свойствам, подсвеченным индикаторами и фильтрами, позволяющими влиять на отображение и взаимодействие с объектами, например, выключать их во Viewport, в Render или отключать возможность их выделения.

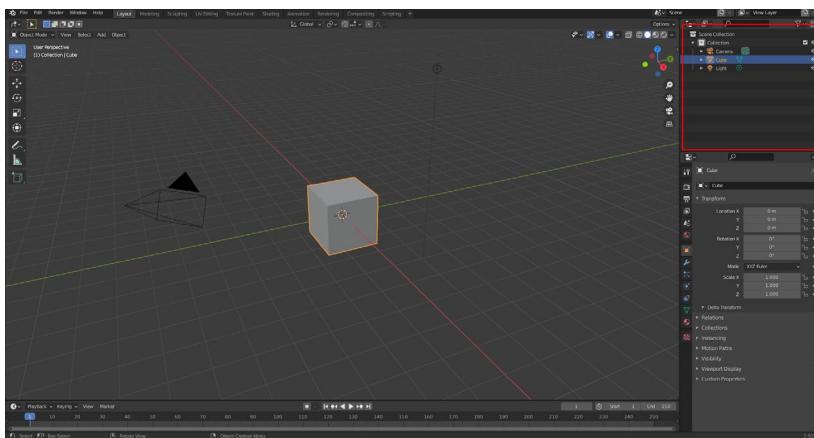


Рисунок 25. Окно навигации, выделено красным

## 1.2. Базовые методы моделирования

Предварительно для осуществления любого взаимодействия, требуется создать базовые объекты. В блендере по умолчанию для создания объектов можно использовать комбинацию клавиш «Shift + A». По нажатию данной комбинации на месте курсора всплывает окно выбора создаваемого объекта. Самые часто используемые объекты находятся в меню Mesh, Curve, Light и Camera.

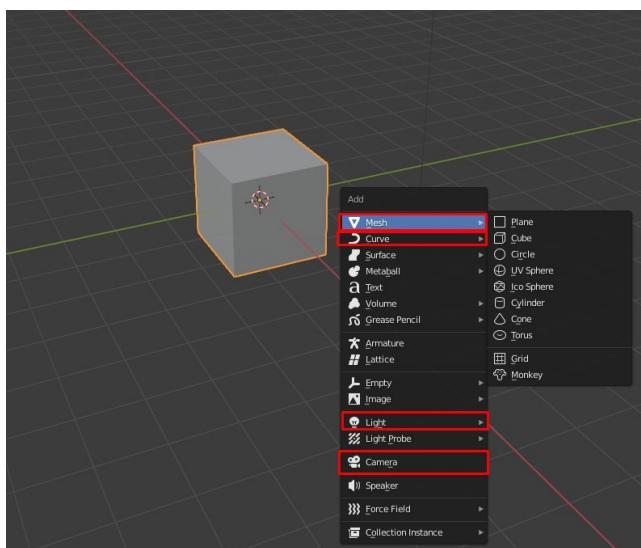


Рисунок 26. Меню добавления объектов

В меню Mesh пользователю предлагается выбрать из набора следующих простых геометрических объектов:

- 1) Plane – квадратная, односторонняя плоскость с четырьмя вертексами по краям. Стоит заметить, что по умолчанию данный объект на рендере будет отображаться лишь, с одной стороны.
- 2) Cube – объёмный куб с восемью вертексами по краям.
- 3) UV Sphere – сфера, по умолчанию состоящая из 32 вертикальных и 16 горизонтальных огибающих рёбер.
- 4) Cylinder – цилиндр, основания которого по умолчанию задаются как круги с 32 вертексами.

Остальные геометрические объекты представляют меньший интерес за счёт меньшего практического смысла. Также во время создания объектов в левом нижнем углу появляется вкладка меню последнего действия, позволяющая изменять параметры только что созданного объекта. Изменения включают в себя размер объекта в метрах, количество сегментов, расположение и вращение в пространстве и другие.

Меню Curve позволяет создавать кривые Безье, которые могут быть использованы для задания определённых линий для расположения объектов с помощью модификатора Align, а также для выдавливания их в объёмные трубы.

В меню Light доступен выбор четырёх типов источников света:

- 1) Point – точечный источник света, который имеет одну точку испускания света, яркость и направление.
- 2) Sun – источник имитирующий солнце, играющий роль большого объекта, все лучи которого падают на объекты под единым углом в независимости от местонахождения этих объектов на сцене.
- 3) Spot – источник света с определённым телесным углом.
- 4) Area – плоскость произвольной формы, одна сторона которой испускает свет.

Опция Camera позволяет добавлять камеры, через которые в последствии может осуществляться рендер сцены.

В дальнейшем созданные объекты можно изменять при помощи опций в панели слева или при помощи горячих клавиш. При нажатии на горячую клавишу Tab или при переходе в режим Modeling пользователю становится доступен функционал изменения сеток выделенных объектов.

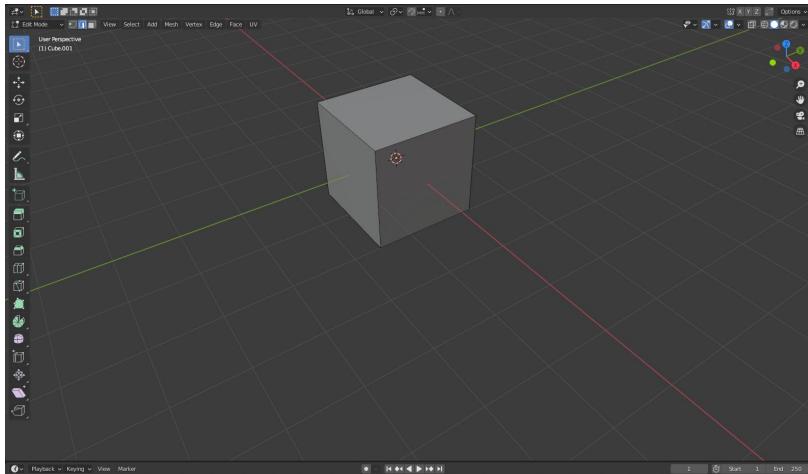


Рисунок 27. Окно Layout в режиме редактирования сетки

В данном режиме в ранее выделенных объектах можно выбирать отдельные вертексы, грани или рёбра. Переключения между режимами выделения подобъектов доступно в панели в левом верхнем углу или по нажатию клавиш «1», «2» или «3» соответственно. При этом после перехода в режим редактирования прежние функции масштабирования и перемещения теперь применимы и к выделенным подобъектам.

В этом режиме удобно пользоваться функцией X-Ray, дающую сквозную видимость через объект. Данная функция позволяет выделять подобъекты тела, скрытые за его гранями без необходимости вращаться вокруг тела.

Среди наиболее часто применяемых модификаций стоит выделить функции масштабирования на клавишу «S», перемещения на клавишу «G», выдавливание объектов на клавишу «E» и удаление объектов на клавишу «X». Более продвинутые функции Inset на клавишу «I», функция Bevel на клавишу «B» и функция Loop Cut на комбинацию клавиш «Ctrl + R» позволяют создавать на выделенном ребре отмасштабированное внутреннее ребро, фаску или набор граней по периметру соответственно.

### **1.3. Методы полигонального моделирования объектов**

Используя вышеупомянутые представляется возможным создать базовый «блокинг» сцены «Фабрика». Под «блокингом» подразумевается первоначальное расположение геометрических объектов, схожих по форме с конечным вариантом в сцене.

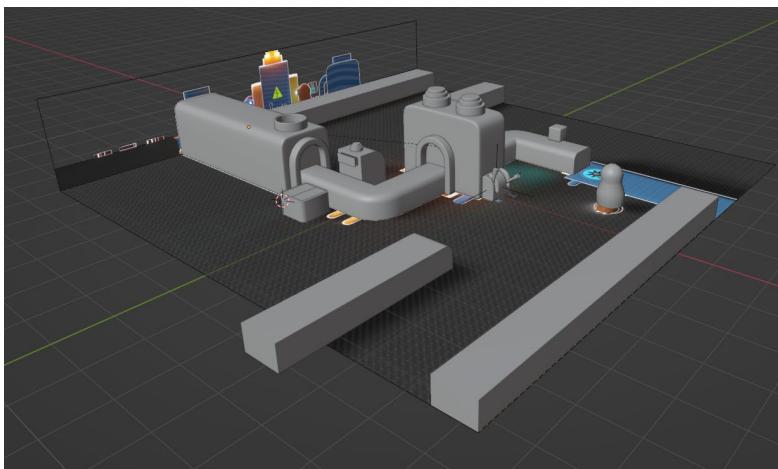


Рисунок 28. «Блокинг» сцены «Фабрика»

Создание арок требует добавление окружности, удаление нижнего полукруга и выдавливания нижних вертексов перпендикулярно вниз. Создание корпуса робота потребовало добавления UV-сферы и последовательного удаления, добавления и масштабирования нижних подобъектов. Для создания конвейера потребовалось последовательное выдавливание передней грани куба и добавление фаски на всём протяжении по бокам.

Для остальных элементов необходимо использование настройки модификаторов.



Рисунок 29. Окно модификаторов, выделено красным

В данном окне возможно присвоение объектам различного набора из списка доступных модификаторов. Одними из главных преимуществ модификаторов являются строгая иерархия применения по их взаиморасположению и тот факт, что модификаторы не влияют непосредственно на конечную сетку объекта, до тех пор, пока они не были применены и их эффекты не были перенесены на сетку.

Для создания фасок на объектах необходимо добавить модификатор Bevel, который имеет опции задания размеров фаски и количества сегментов, образуемых при построении. Также при создании некоторых объектов на данном этапе могут понадобится модификаторы Solidify добавления объёма фигуре перпендикулярно её нормали, Mirror отражения сетки объекта вокруг определённых осей с центром в указанной точке и модификатор Subdivision Surface, делающий более дробную сетку объекта, для того чтобы сглаживание объектов производилось более плавно.

На втором этапе работы была добавлена конвейерная лента, винты и созданы проёмы для арок.

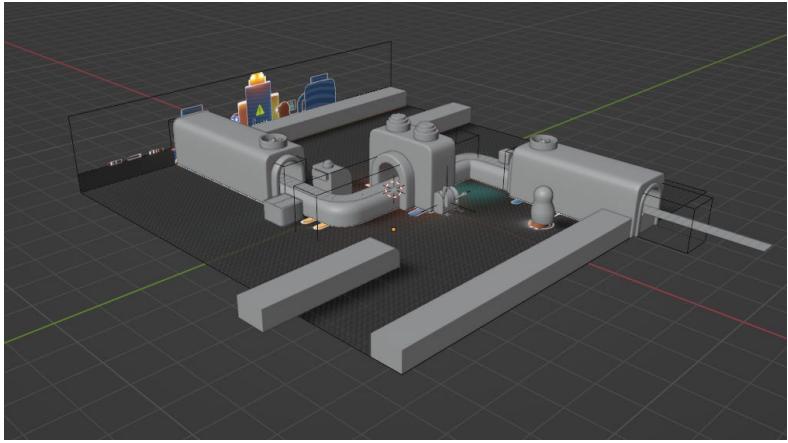


Рисунок 30. Сцена «Фабрика» на втором этапе моделирования

Для создания направления ленты была создана единая прямоугольная пластина и на её основе создан массив элементов при помощи модификатора Array. Направление для ленты задаётся посредством модификатора Curve, берущего за основу существующую кривую и располагающую объекты вдоль осей выбранной кривой, которая была создана на основе ранее созданного конвейера.

Для создания вентиляторов была создана сфера и из неё получен окружлый стержень. Листы винтов были созданы при помощи добавления плоскости Plane и затем изменены через модификаторы Subdivision Surface и Simple Deform, позволяющий искривлять объект вокруг одной осей. Затем данные листы были размножены копированием и закреплены модификатором Align к центру стержня.

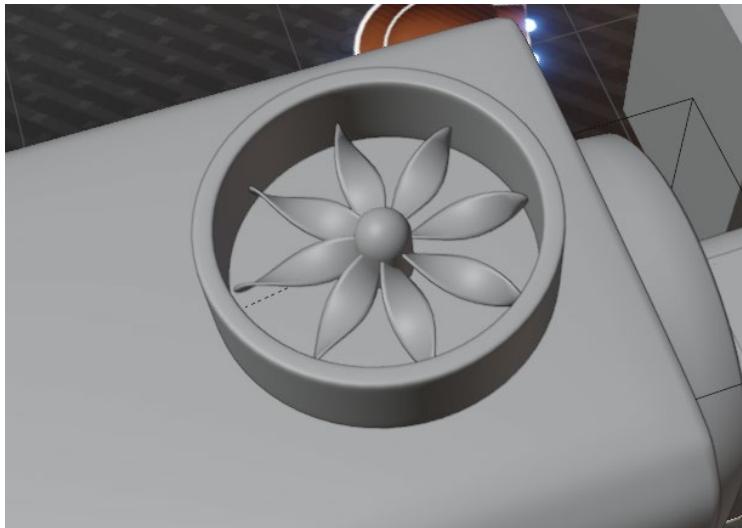


Рисунок 31. Объект вентилятор

Для создания отверстий был использован модификатор Boolean. Данный модификатор позволяет совершать взаимодействия между двумя объектами – объектом который использует модификатор и объектом, который указан в настройках данного модификатора. У этого модификатора присутствует 3 режима работы. Intersect оставляет лишь пересечение двух объектов, Union совмещает два объекта, Difference выполняет вычитание одного объекта из другого, при этом второй объект перестаёт быть твердотельным и оставляет после себя лишь остов для возможности его перемещения и изменения конечного объекта вычитания. Последний режим был применён для создания арок при помощи вычитания скопированных арок.

На третьем этапе были созданы трубы и провод зарядки для робота путём создания кривой и добавления модификатора Solidify. Маска робота и глаза с использованием уже известных методов изменения сетки объектов. Также был добавлен элемент «Лайк» над конвейером, созданный путём выдавливания и соединения вертексов по контуру объекта.

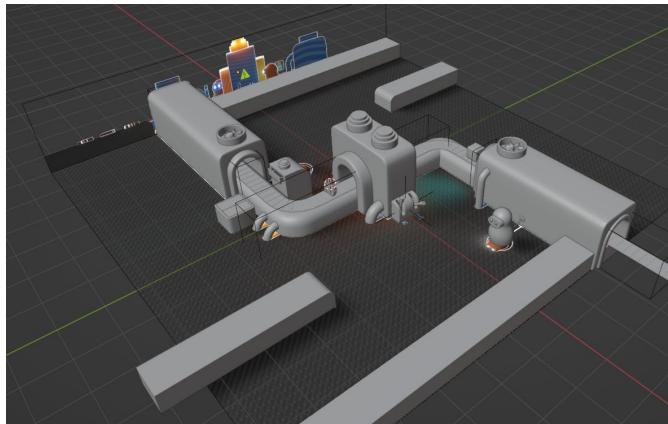


Рисунок 32. Сцена «Фабрика» на третьем этапе моделирования

На четвёртом этапе было добавлено освещение без материалов. Внутри арок были добавлены объекты Area Light, также был создан источник света Sun над сценой. Также была добавлена камера для корректировки угла обзора сцены для корректного построения освещения с учётом требуемого результата сцены «Фабрика». Освещение сцены производится только в режиме отображения Render.



Рисунок 33. Сцена «Фабрика» на четвёртом этапе моделирования

Стоит отметить что обзор через установленную активную камеру осуществляется нажатием клавиши «0» на NumPad. При этом камера даёт ортогональный обзор.

Последним пятым этапом объектам сцены добавляются материалы и текстуры. Редактирование текстур осуществляется в меню Shading. После создания нового материала для тела появляется доступ к его свойствам.

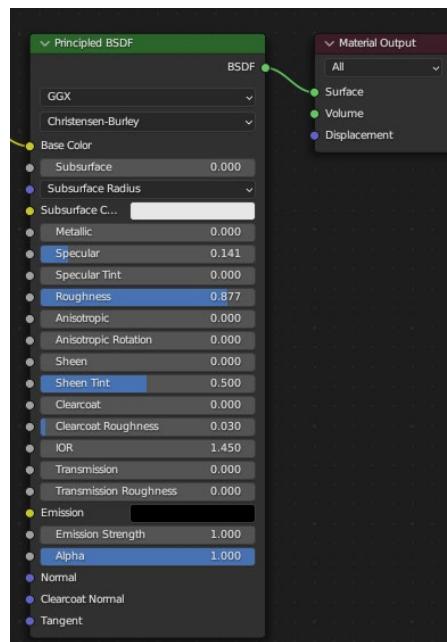


Рисунок 34. Свойства материала в меню Shading

Наиболее важные параметры:

- 1) Base Color – отвечает за цвет объекта.
- 2) Metallic – отражает металличность объекта, позволяя имитировать свойства отражения света материалов из металла.
- 3) Specular – отвечает за блики.
- 4) Roughness – отвечает за грубоность поверхности объекта. Низкие значения сделают объект идеально гладким на вид, позволяя отражать больше света.
- 5) Emission – цвет эмиссионного света материала.

## 6) Emission Strength – сила эмиссии.

Для создания простых материалов достаточно указать определённый цвет и параметры отражения света.

Для того чтобы добавить текстуру на объект достаточно перенести изображение текстуры в меню снизу и подключить его к узлу Base Color. Далее текстуре требуется назначить способ, по которому текстура накладывается на объект. Для этого создается панель Mapping и Texture Coordinate. Конечный результат создания текстуры для куба представлен на рисунке ниже.

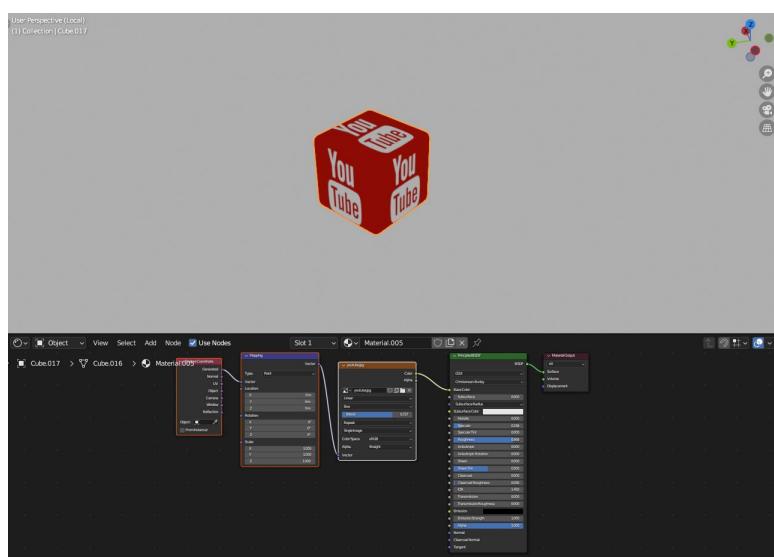


Рисунок 35. Результат создания материала для куба «YouTube»

Для создания материалов для робота требуется использования масок и эмиссии света. Поскольку робот в процессе работы был преобразован в единый объект, для того чтобы добавить на его части разные цвета необходимо воспользоваться картами масок. Глаза робота не были преобразованы в подобъект робота и поэтому им достаточно задать параметры Emission. Для светящихся цилиндров был применён подход создания различных материалов и определения для них разных наборов вертексов тела. Один набор вертексов получил простой материал, имеющий оранжевый цвет, а другой – чёрный

материал, излучающий жёлтый свет. Ниже представлены окончательный рендер данной сцены в Blender.

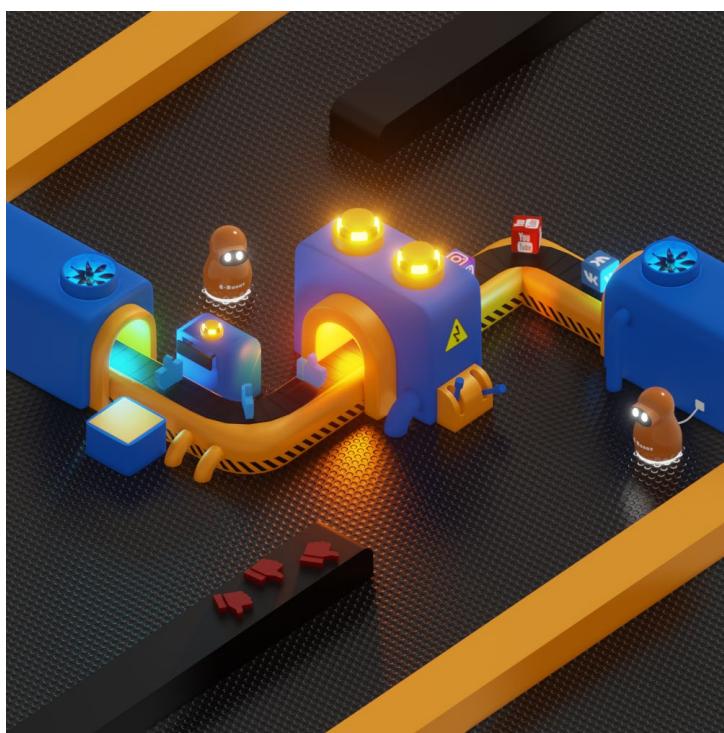


Рисунок 36. Окончательный рендер сцены «Фабрика» в Blender

## 1.4. Методы «скульптинга»

Для создания сцены «Енот» упомянутые выше методы не подойдут поскольку форма объекта не представляет собой подобие стандартной геометрической фигуры. Для создания подобных объектов в Blender существует режим Sculpting.

Скульптинг – это процесс 3D-моделирования, при котором используется инструменты воздействия на сетку объекта с помощью инструментов схожих с теми что используются при обработке глиняных скульптур.

Первоначально для моделирования были созданы несколько сфер для головы, тела, ушей, глаз, носа и лап енота. Далее данные сферы были растянуты в более схожие с результатом формы. Также был создан куб для коробки и цилиндр для хвоста. Для их построения можно использовать инструменты упомянутые в прошлой секции. Хвост был создан при помощи добавления цилиндра и последующего выдавливания масштабирования передней грани цилиндра. Коробка была создана путём масштабирования сторон, удаление верхней грани и вытягивания боковых граней в стороны. После придания примитивным объектам подходящей формы имеет смысл переходить к непосредственно к «скульптингу».

Основными инструментами «скульптинга» в процессе работы были функции:

- 1) Draw Sharp – тонкая, острая кисть позволяющая наращивать либо вдавливать материал в месте нажатия.
- 2) Clay Strips – инструмент схожий с наложением материала шпателем. При нажатии на объект наращивает либо вдавливает слои в месте нажатия.
- 3) Smooth – инструмент сглаживания поверхности.
- 4) Pinch – инструмент сжатия соседних объектов в центр места нажатия.
- 5) Grab – инструмент позволяющий тянуть за объект для добавления деформаций по направлению.

Стоит заметить, что для удобной работы с инструментами «скульптинга» необходимо создать более дробную сетку для объекта, поскольку данные инструменты могут работать некорректно при разряженной сетки и создавать слишком острые изменения. По этой причине при переходе в режим Sculpting требуется перейти во вкладку Remesh и уменьшить размер вокселей, а затем нажать кнопку Remesh.

При помощи данных инструментов были добавлены углубления в местах для ушей и рта, наращены массы в местах бровей, вытянуты лапы и наращены плечи при помощи инструмента Clay Strips. Результат первоначального «скульптинга» представлен ниже.



Рисунок 37. Первый этап моделирования сцены «Енот»

Далее отельные объекты ушей, торса и лап были совмещены с исходным объектом головы и сглажены на стыках при помощи инструмента Smooth. Были проделаны углубления для носа, при помощи инструмента Draw Sharp были созданы дуги вокруг глаз и более острая обводка рта.



Рисунок 38. Второй этап моделирования сцены «Енот»

Как упоминалось ранее, для удобной работы с инструментами «скульптуинга» необходимо увеличение числа элементов в сетке объекта. Ниже представлена сетка объекта после преобразований.

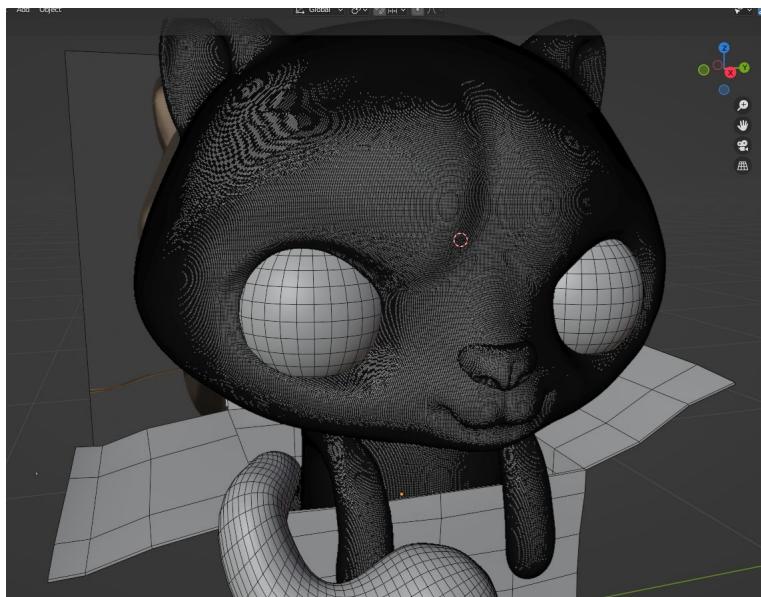


Рисунок 39. Сетка объекта до применения «запечки»

При этом серьёзно страдает производительность обработки и рендера объекта. Чтобы избежать проблем с производительностью можно воспользоваться функцией «запечки» рельефа объекта в определённую текстуру, карты рельефа и нормалей которой можно будет использовать для имитации детального объекта на менее полигональной, упрощённой модели. После применения метода «запечки» и применения полученных текстур на новой модели был получен следующий результат:

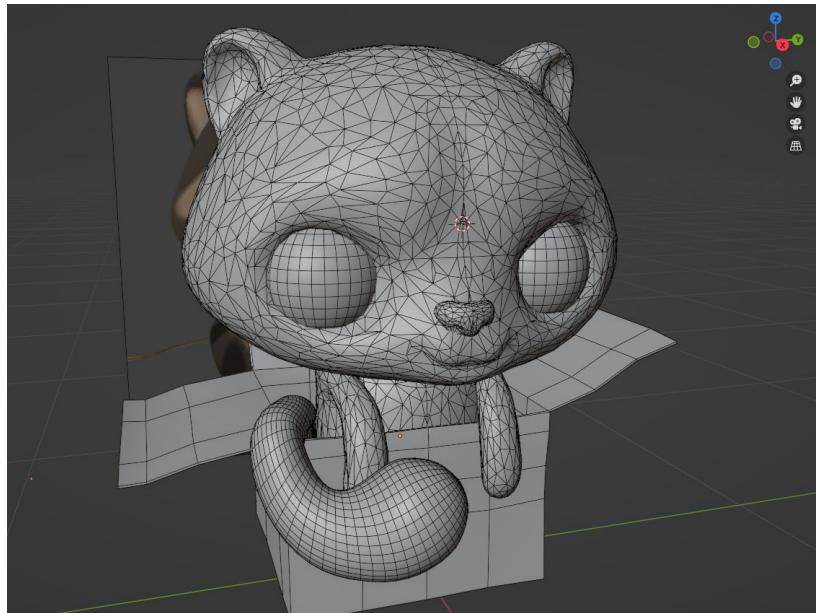


Рисунок 40. Сетка объекта после применения «запечки»



Рисунок 41. Вид объекта после применения «запечки»

Далее во вкладке UV Editing была автоматически получена UV-сетка объекта при помощи функции UV → Smart UV project, а затем произведена функция unwrap для расположения сетки на плоскости для дальнейшего наложения текстур.

После создания UV развертки можно создать материал. Далее используя кисти во вкладке Texture Paint можно покрасить енота в необходимые цвета, а затем присвоить материалу необходимые параметры взаимодействия со светом во вкладке Shading. Для добавления текстур на коробку во вкладке Texture Paint существует способ покраски через трафарет, дающий возможность добавлять одинаковые рисунки отпечатков лап на объекте.

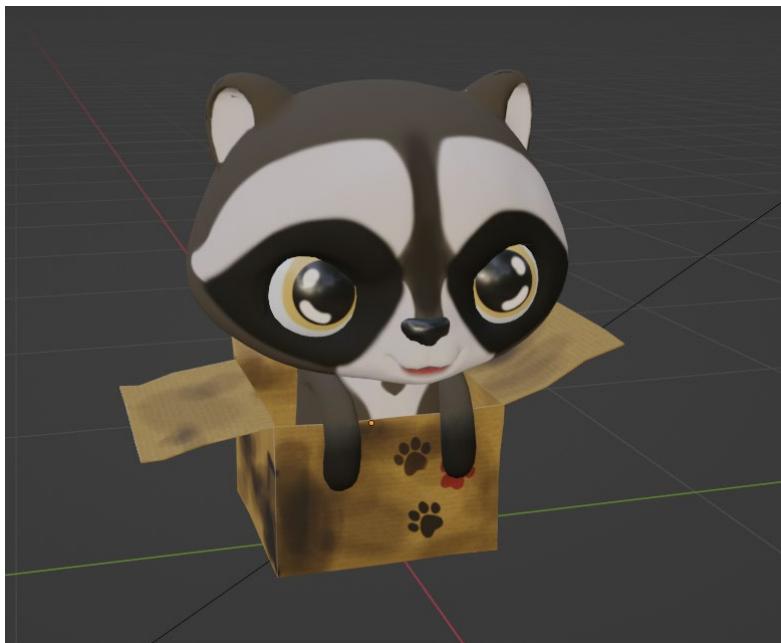


Рисунок 42. Сцена «Енот» после наложения материалов и текстур

Далее необходимо преступить к созданию шерсти. Шерсть в Blender создаётся при помощи добавления Particle Setting в меню Particles, доступном в правом нижнем углу. После выбора опции Hair программа создаёт на всём объекте отдельные элементы похожие на стержни. В настройках можно изменять

различные параметры, отвечающие за количество элементов, их скученность, «грязноту», длину, кривизну и многие другие.

Для того чтобы выделять определённые зоны в которых будут появляться элементы волос необходимо переключится на вкладку Weight Paint в левом верхнем углу вместо Object Mode и затем при помощи кисти добавить или удалить области распределения. После добавления нескольких зон и присвоения им различных систем волос для имитации поведения разнородной шерсти можно приступить к укладыванию волос по направлению. Для этого требуется перейти во вкладку Particle Edit в левом верхнем углу. Данный режим предоставляет доступ к инструментам схожим с причёсыванием, позволяя укладывать или расpusшать, удалять или добавлять, удлинять или укорачивать шерсть в месте нажатия. После применения данных инструментов был получен конечный результат. Более того в сцену было добавлено несколько вспомогательных объектов и освещение схожее с требуемым исходником.



Рисунок 43. Сцена «Енот» после окончательного рендера в Blender

## **1.5. Экспорт и импорт**

Для переноса файлов из Blender в Unreal Engine 5 существует множество методов со своими плюсами и минусами.

Первый метод заключается в экспорте файлов в формате FBX. Преимущество данного метода состоит в возможности точной настройки элементов экспорта и настройке качества передачи данных. Основным минусом является невозможность передачи объектов освещения, некоторых внутренних настроек объектов и неспособность переносить модификаторы. Последнее может быть несущественно поскольку после воссоздание конечной сцены в Blender все модификаторы можно без осложнений применить к геометрии и воспользоваться упомянутым экспортом.

Второй метод использует файлы glTF2.0. Данный формат имеет меньшее число опций, но отличается тем что экспортирует данные в сыром формате, который хранит больше информации и позволяет переносить элементы освещения и более корректно устанавливать материалы для объектов.

Третий способ передачи данных состоит в установке взаимной связи между приложениями Blender и Unreal Engine 5 при помощи плагина Send2UE. Данный плагин устанавливается для Blender и открывает доступ к меню Pipeline, в котором можно настроить пути и способы передачи данных из Blender. Для корректной работы со стороны Unreal Engine 5 требуется открыть доступ удалённого изменения в настройках проекта и добавить несколько плагинов взаимодействия с Python, которые установлены по умолчанию в новых версиях Unreal Engine 5.

После создания базовой сцены и импорта полученных материалов в Unreal Engine 5 были получены конечные сцены рендера в среде Unreal Engine 5.

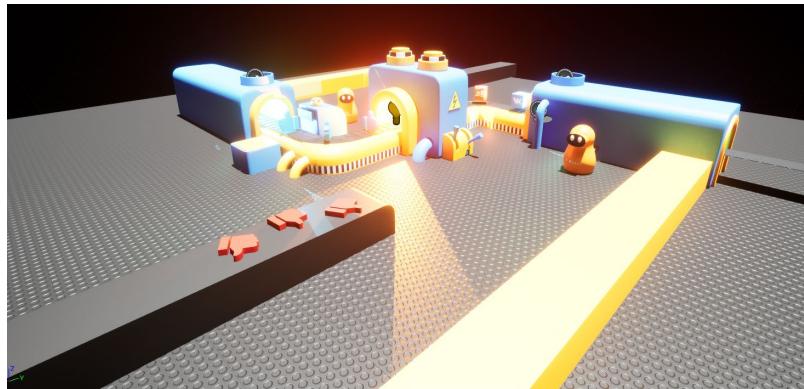


Рисунок 44. Рендер сцены «Фабрика» в Unreal Engine 5



Рисунок 45. Рендер сцены «Енот» в Unreal Engine 5

Стоит отметить что даже при корректном экспорте всех материалов из среды в Blender в Unreal Engine 5 сцена не будет выглядеть идентично в обоих программах. Это связано как с внутренними механизмами рендеринга, так и с разным отношением данных программ к различным элементам, например, к шерсти.

### **3. Создание RTS окружения**

<TODO>

## **ЗАКЛЮЧЕНИЕ**

В результате работы были получены навыки работы с GitHUB, World Creator 3, Gaea и World Machine. Улучшен опыт разработки приложений на Unreal Engine. Дополнены знания механизмов оптимизации статических объектов.

Также были освоены навыки 3D-моделирования объектов при помощи различных методик моделирования. Изучены и освоены механизмы настройки освещения, создания и модификации материалов, добавления элементов волос. Получены углублённые знания работы системы рендера Eevee и Cycles, систем распределения UV-развёрток и методов их ручной и автоматической настройки. Разработаны и успешно экспортированы две сцены в среду разработки движка Unreal Engine 5.

**В конечном итоге были разработаны все стадии и механизмы RTS системы с возможностью пользовательского управления, а также возможностью добавления новых карт с использованием созданного RTS окружения.**

## **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Modifying the Navigation Mesh // Unreal Engine Documentation URL:  
<https://docs.unrealengine.com/5.1/en-US/overview-of-how-to-modify-the-navigation-mesh-in-unreal-engine/>. Дата обращения: 16.09.2023;
2. Basic Navigation// Unreal Engine Documentation URL:  
<https://docs.unrealengine.com/5.1/en-US/basic-navigation-in-unreal-engine/>. Дата обращения: 16.09.2023;
3. Landscape Material Layer Blending // Unreal Engine Documentation URL:  
<https://docs.unrealengine.com/5.1/en-US/landscape-material-layer-blending-in-unreal-engine/>. Дата обращения: 16.09.2023;
4. Landscape Quick Start Guide // Unreal Engine Documentation URL:  
<https://docs.unrealengine.com/5.0/en-US/landscape-quick-start-guide-in-unreal-engine/>. Дата обращения: 16.09.2023;
5. Creating Landscapes// Unreal Engine Documentation URL:  
<https://docs.unrealengine.com/5.2/en-US/creating-landscapes-in-unreal-engine/>. Дата обращения: 16.09.2023;
6. Landscape Edit Layers// Unreal Engine Documentation URL:  
<https://docs.unrealengine.com/5.2/en-US/landscape-edit-layers-in-unreal-engine/>. Дата обращения: 16.09.2023;
7. Landscape Collision Guide // Unreal Engine Documentation URL:  
<https://docs.unrealengine.com/5.2/en-US/landscape-collision-guide-in-unreal-engine/>. Дата обращения: 16.09.2023;
8. Unreal Engine 4 Documentation // Unreal Engine Documentation URL:  
<https://docs.unrealengine.com/>. Дата обращения: 16.09.2023;
9. Unreal Engine 5 : How to REMOVE/HIDE Texture REPETITION with Natural Variations in UE5 Tutorial // YouTube, video sharing platform URL:  
<https://www.youtube.com/watch?v=j1AekcZeOKc>. Дата обращения: 16.09.2023;
10. How to Paint Landscape with Megascans Materials in Unreal Engine 5 // YouTube, video sharing platform URL:

<https://www.youtube.com/watch?v=eW0iLoO6QrM>. Дата обращения: 16.09.2023;

11. Как настроить свет в Unreal Engine 5. Outdoor Light Tutorial UE5 // YouTube, video sharing platform URL:

<https://www.youtube.com/watch?v=49O3S4VQZvc>. Дата обращения: 16.09.2023;

12. Create a Photorealistic Landscape in Unreal Engine 5 // YouTube, video sharing platform URL:

<https://www.youtube.com/watch?v=WhZxDpxvMwY>. Дата обращения: 16.09.2023;

13. UE5 Create an RTS Camera with Movement, Rotating and Zoom - Tutorial - Adjust to terrain // YouTube, video sharing platform URL:

<https://www.youtube.com/watch?v=-nvWVY-oVIY>. Дата обращения: 16.09.2023;

14. Setting Up and Editing a Good Navmesh in Unreal Engine 5 - The Designer's Recourse // YouTube, video sharing platform URL:

<https://www.youtube.com/watch?v=POVVHSAIT7E>. Дата обращения: 16.09.2023;

15. How to export World Creator landscapes to Unreal Engine // YouTube, video sharing platform URL:

<https://www.youtube.com/watch?v=3YcVLql4fxQ>. Дата обращения: 16.09.2023;

16. How to HIDE Texture REPETITION in Unreal Engine - UE4 Tutorial // YouTube, video sharing platform URL:

<https://www.youtube.com/watch?v=yCRzOdo4b68>. Дата обращения: 16.09.2023;

17. Building natural environments in Unreal Engine // YouTube, video sharing platform URL: <https://www.youtube.com/watch?v=gbj1qgPOl3E>. Дата обращения: 16.09.2023;

- 18.UE4 Simple Water Material 1 Unreal Engine 4.26 (Tutorial) // YouTube, video sharing platform URL:  
<https://www.youtube.com/watch?v=52RR8jIrDhQ>. Дата обращения: 16.09.2023;
- 19.Quickly Add Landscapes to Blender using World Creator 2022 - 1/3 Basics & Filters // YouTube, video sharing platform URL:  
<https://www.youtube.com/watch?v=hBmonmwcWnQ&t=310s>. Дата обращения: 16.09.2023;
- 20.Chapter 2. Terrain Rendering Using GPU-Based Geometry Clipmaps // NVidia Developer URL:  
<https://developer.nvidia.com/gpugems/gpugems2/part-i-geometric-complexity/chapter-2-terrain-rendering-using-gpu-based-geometry>. Дата обращения: 16.09.2023;
21. 01/19 – Введение. Курс «Енот». Скульптинг персонажа в Blender. Knower School. // YouTube, video sharing platform URL:  
[https://www.youtube.com/watch?v=fDOotm6bxUY&list=PLn6DikVGbeEgMvn\\_JJyX1Rnrt3Wlj0rvk&index=1](https://www.youtube.com/watch?v=fDOotm6bxUY&list=PLn6DikVGbeEgMvn_JJyX1Rnrt3Wlj0rvk&index=1). Дата обращения: 15.11.2023.
22. 01/39 – Интерфейс Blender. Курс «Фабрика». Обучение Blender с нуля. Knower School. // YouTube, video sharing platform URL:  
[https://www.youtube.com/watch?v=Q8Q8HBj-Ua8&list=PLn6DikVGbeEiJFNb2\\_wfV2zg4BDm8xvsQ&index=1](https://www.youtube.com/watch?v=Q8Q8HBj-Ua8&list=PLn6DikVGbeEiJFNb2_wfV2zg4BDm8xvsQ&index=1). Дата обращения: 15.11.2023.
23. Send to Unreal - Quickstart | Unreal Engine // YouTube, video sharing platform URL:  
[https://www.youtube.com/watch?v=apa9EXI2KZA&list=PLZlv\\_N0\\_O1gZfQaN9qXynWll7bzX8H3t&index=1](https://www.youtube.com/watch?v=apa9EXI2KZA&list=PLZlv_N0_O1gZfQaN9qXynWll7bzX8H3t&index=1). Дата обращения: 15.11.2023.
24. Hair Rendering // Unreal Engine Documentation URL:  
<https://docs.unrealengine.com/5.2/en-US/hair-rendering-in-unreal-engine/>.  
Дата обращения: 15.11.2023.

25. Generating Groom Textures // Unreal Engine Documentation URL:  
<https://docs.unrealengine.com/5.2/en-US/generating-groom-textures-in-unreal-engine/>. Дата обращения: 15.11.2023.
26. How to HIDE Texture REPETITION in Unreal Engine - UE4 Tutorial //  
YouTube, video sharing platform URL:  
<https://www.youtube.com/watch?v=yCRzOdo4b68>. Дата обращения:  
15.11.2023.