|  |  |
| --- | --- |
| Герб МГТУ | **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:*** | |
| ***«Визуализация природных ландшафтов и средневекового поселения»*** | |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | РК6-84Б |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Колыхалов Д. В.** |
| (Группа) | (Подпись, дата) | (Фамилия И.О.) |
| Руководитель ВКР | | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Витюков Ф.А.** |
| (Подпись, дата) | (Фамилия И.О.) |
| Нормоконтролёр | | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Грошев С.В.** |
| (Подпись, дата) | (Фамилия И.О.) |

|  |
| --- |
|  |
| *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  
(Индекс)  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Карпенко А.П.  
(И.О.Фамилия)  
«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ  
на выполнение выпускной квалификационной работы**

|  |  |
| --- | --- |
| Студент группы | РК6-84Б |
| Колыхалов Дмитрий Витальевич | |
| (фамилия, имя, отчество) | |
| Тема квалификационной работы: «Визуализация природных ландшафтов и средневекового поселения» | |

При выполнении ВКР:

|  |  |
| --- | --- |
| Используются / Не используются | Да/Нет |
| 1. Литературные источники и документы, имеющие гриф секретности | Нет |
| 1. Литературные источники и документы, имеющие пометку «Для служебного пользования», иных пометок, запрещающих открытое опубликование | Нет |
| 1. Служебные материалы других организаций | Нет |
| 1. Результаты НИР (ОКР), выполняемой в МГТУ им. Н.Э.Баумана | Нет |
| 1. Материалы по незавершенным исследованиям или материалы по завершенным исследованиям, но ещё не опубликованные в открытой печати | Нет |

Тема квалификационной работы утверждена распоряжением по факультету \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ № \_\_\_\_\_\_\_\_\_\_\_от «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 202\_ г.

**Техническое задание**

**Часть 1.** ***Аналитическая часть***

*Изучить полный цикл создания природных ландшафтов. Изучить средства работы с ландшафтами большого размера, предоставляемые трёхмерным движком Unreal Engine 5. Провести анализ различных методов рельефного текстурирования.*

**Часть 2.** ***Практическая часть 1. Создание ландшафтов***

*Разработать природную сцену небольшого размера с высокой детализацией. Создать ландшафт большого размера, выполнить импорт в движок Unreal Engine 5. Наполнить ландшафт природными объектами. Создать средневековое поселение. Провести оптимизацию отображения.*

**Часть 3.** ***Практическая часть 2. Разработка персонажа окружения***

*Разработать анимированного персонажа окружения. Создать искусственный интеллект, управляющий персонажем окружения.*

|  |
| --- |
| **Оформление выпускной квалификационной работы** |
| Расчетно-пояснительная записка на **52** листах формата A4. |
| Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.): |
| |  | | --- | | *Работа содержит 8 графических листов формата A4.* | |  | |  | |  | |
| Дата выдачи задания: «10» февраля 2025 г. |
| В соответствии с учебным планом выпускную квалификационную работу выполнить в полном объеме в срок до «11» июня 2025 г. |

|  |  |  |
| --- | --- | --- |
| **Руководитель квалификационной работы** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Витюков Ф.А.** |
|  | (Подпись, дата) | (Фамилия И.О.) |
| **Студент** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Колыхалов Д. В.** |
|  | (Подпись, дата) | (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 | РК6-84Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Карпенко А.П. | |
| (Фамилия И.О.) | |
| « » 202 г. | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **КАЛЕНДАРНЫЙ ПЛАН** | | | | | | | | |
| **выполнения выпускной квалификационной работы** | | | | | | | | |
| Студент группы | | | РК6-84Б | | | | | |
| Колыхалов Дмитрий Витальевич | | | | | | | | |
| (фамилия, имя, отчество) | | | | | | | | |
| Тема квалификационной работы: «Визуализация природных ландшафтов и средневекового поселения» | | | | | | | | |
| **№ п/п** | **Наименование этапов выпускной квалификационной работы** | | **Сроки выполнения этапов** | | **Отметка о выполнении** | |
| **план** | **факт** | **Должность** | **ФИО, подпись** |
|  | Задание на выполнение работы. Формулирование проблемы, цели и задач работы | | 10.02.2025 | \_\_\_\_\_\_\_\_\_ | Руководитель ВКР | Витюков Ф.А. |
|  | 1 часть. Аналитическая часть | | 18.02.2025 | \_\_\_\_\_\_\_\_\_ | Руководитель ВКР | Витюков Ф.А. |
|  | Утверждение окончательных формулировок решаемой проблемы, цели работы и перечня задач | | 28.02.2025 | \_\_\_\_\_\_\_\_\_ | Заведующий кафедрой | Карпенко А.П. |
|  | 2 часть. Практическая часть 1 | | 21.04.2025 | \_\_\_\_\_\_\_\_\_ | Руководитель ВКР | Витюков Ф.А. |
|  | 3 часть. Практическая часть 2 | | 23.05.2025 | \_\_\_\_\_\_\_\_\_ | Руководитель ВКР | Витюков Ф.А. |
|  | 1-я редакция работы | | 28.05.2025 | \_\_\_\_\_\_\_\_\_ | Руководитель ВКР | Витюков Ф.А. |
|  | Подготовка доклада и презентации | | 04.06.2025 | \_\_\_\_\_\_\_\_\_ | Студент | Котельникова Е.Ю. |
|  | Заключение руководителя | | 10.06.2025 | \_\_\_\_\_\_\_\_\_ | Руководитель ВКР | Витюков Ф.А. |
|  | Допуск работы к защите на ГЭК (нормоконтроль) | | 17.06.2025 | \_\_\_\_\_\_\_\_\_ | Нормоконтролер | Грошев С.В. |
|  | Внешняя рецензия | | 17.06.2025 | \_\_\_\_\_\_\_\_\_ |  |  |
|  | Защита работы на ГЭК | | 21.06.2025 | \_\_\_\_\_\_\_\_\_ |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Колыхалов Д. В. | Руководитель ВКР | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Витюков Ф.А. |
|  | (подпись, дата) | (Фамилия И.О.) |  | (подпись, дата) | (Фамилия И.О.) |

**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)**

**НАПРАВЛЕНИЕ НА ГОСУДАРСТВЕННУЮ ИТОГОВУЮ АТТЕСТАЦИЮ**

**Председателю   
Государственной Экзаменационной Комиссии № \_\_\_\_\_\_\_**

факультета *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*МГТУ им. Н.Э. Баумана

Направляется студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_группы \_\_\_\_\_\_\_\_\_\_\_

на защиту выпускной квалификационной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Декан факультета *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

**Справка об успеваемости**

Студент \_\_\_\_ за время пребывания в МГТУ имени Н.Э. Баумана

с 20 \_\_\_\_ г. по 20 \_\_\_\_ г. полностью выполнил учебный план со следующими оценками: отлично – %, хорошо – %, удовлетворительно – %.

Инспектор деканата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Отзыв руководителя выпускной квалификационной работы**

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО студента) (подпись) (дата)

**АННОТАЦИЯ**

Работа посвящена различных способам и инструментам разработки реалистичных пейзажей и ландшафтов, а также созданию персонажей окружения в Unreal Engine 5.

В данной работе рассмотрены и проиллюстрированы следующие этапы: прохождение полного цикла создания ландшафта, импорт в движок Unreal Engine 5, оптимизация отображения больших ландшафтов с помощью различных механизмов, создание материалов ландшафта, добавление растительности в сцены, настройка освещения, создание водных объектов и интеграция их в природные сцены, создание средневекового поселения, а также разработка и анимация персонажа окружения, настройка искусственного интеллекта персонажа окружения.

Тип работы: выпускная квалификационная работа.

Тема работы: «Визуализация природных ландшафтов и средневекового поселения».

Объекты исследований: создание трехмерных сцен в Unreal Engine 5, повышение производительности, 3d-анимация

**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 предоставляют «базис» для разработки, «надстройку» над которым создает его пользователь.

**UE5** – трехмерный движок Unreal Engine 5.

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

**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 UE5 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.

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

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

**NPC (non-Player Character)** – персонаж окружения, который служит важным средством для создания атмосферы в сцене.

**PCG (Procedural Content Generation)** – процедурный алгоритм, который генерирует данные на основе определенных правил и входных данных.

**Биом** – структурированная зона с уникальными природными характеристиками.

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

**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** – графический объект, который определяет, как поверхность 3D объекта будет выглядеть при рендеринге.

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

**Shader (Шейдер)** – компьютерная программа, выполняющаяся параллельно на графическом процессоре, и служащая для различных графических вычислений, таких как отрисовка 3д моделей, расчёт освещения и так далее.

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

**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.

**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 UE5 надстройка над множеством графических API (например: DirectX, OpenGL, Vulkan), позволяющая писать независимый от платформы код.

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

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

**Rendering (рендеринг)** – процесс преобразования 3D сцены в 2D изображение, которое отображается на экране.

**AI (ИИ)** – искусственный интеллект

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 11](#_Toc191823232)

[1. Создание сцены малого размера 15](#_Toc191823233)

[1.1. Создание ландшафта 15](#_Toc191823234)

[1.2. Импорт ландшафта 17](#_Toc191823235)

[1.3. Инструменты редактирования ландшафта 19](#_Toc191823236)

[1.3.1. Изменение карты высот – Sculpt 20](#_Toc191823237)

[1.3.2. Возврат к нулевой карте высот – Erase 21](#_Toc191823238)

[1.3.3. Сглаживание - Smooth 22](#_Toc191823239)

[1.3.4. Выравнивание - Flatten 22](#_Toc191823240)

[1.3.5. Спуск - Ramp 23](#_Toc191823241)

[1.3.6. Термическая эрозия – Thermal Erosion 23](#_Toc191823242)

[1.3.7. Гидроэрозия – Hydro Erosion 24](#_Toc191823243)

[1.3.8. Noise 24](#_Toc191823244)

[1.4. Растительность 25](#_Toc191823245)

[1.4.1. Painting 27](#_Toc191823246)

[1.4.2. Placement 28](#_Toc191823247)

[1.4.2. Instance Static 28](#_Toc191823248)

[1.5. Настройка неба 29](#_Toc191823249)

[1.6. Создание тумана 30](#_Toc191823250)

[1.7. Создание кат-сцены 31](#_Toc191823251)

[2. Создание большой ландшафтной сцены и средневекового поседения 32](#_Toc191823252)

[2.1. Материал ландшафта 32](#_Toc191823253)

[2.2. Средневековое поселение 34](#_Toc191823254)

[2.3. Водные объекты 35](#_Toc191823255)

[2.4. Настройка освещения 37](#_Toc191823256)

[2.5. Оптимизация 38](#_Toc191823257)

[3. Разработка процедурной генерации биома 40](#_Toc191823258)

[4. Разработка персонажа окружения 44](#_Toc191823259)

[Результаты 49](#_Toc191823260)

[ЗАКЛЮЧЕНИЕ 51](#_Toc191823261)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 52](#_Toc191823262)

ВВЕДЕНИЕ

В современном мире трехмерные ландшафты находят применение во многих сферах. Большие сцены с участием природных пейзажей, созданных с помощью компьютерной графики, повсеместно используются в телевидении, кинематографе, видеоиграх, в качестве демонстрационного материала для различных архитектурных проектов.

В данной работе рассматривается создание природных ландшафтов с использованием трёхмерного движка Unreal Engine 4. Среди всех инструментов Unreal Engine выделяет ориентированность на реалистичность получаемого изображения и возможность работы «в реальном времени», не требуются длительные расчёты как при рендере 3д графики с помощью таких инструментов как 3ds Max, Blender и др. Это позволяет быстро привносить изменения в проекты и создавать интерактивные сцены, способные меняться в зависимости, например, от ввода пользователя.

Работа делится на четыре основные части:

1. Создание сцены малого размера;
2. Создание большой ландшафтной сцены и средневекового поселения;
3. Разработка процедурной генерации биома;
4. Разработка персонажа окружения;

*Часть первая. Создание сцены малого размера.*

Природные сцены могут использоваться как фоны для ведущих телевидения, интерьерные интерактивные картины, рекламные фоны. В таких сценах большую роль играет уровень детализации, ведь в случае, если зритель будет приглядываться к изображению, которое не меняется, он быстрее заметит отсутствие мелких деталей и других погрешностей.

Другим важным аспектом является освещение, зачастую именно оно определяет уровень реалистичности изображения. Грамотное использование прямого и отраженного света, атмосферной перспективы в виде тумана и других природных эффектов может позволить получить картинку, местами почти неотличимую от фотографии.

При создании таких сцен важно ориентироваться на реальные фотографии природы и анализировать их составляющие.

В рамках первой части данной работы рассмотрено создание небольшой природной сцены.

*Часть вторая. Создание большой ландшафтной сцены и средневекового послеления.*

Большие ландшафтные сцены могут использоваться для кинематографа и видеоигр. При создании сцен, по которым будет перемещаться камера, помимо детализации и освещения важную роль играют «наполненность сцены» и оптимизация.

Под «наполненностью» подразумевается равномерное распределение объектов, привлекающих внимание, таких как деревья, растения и камни. Потенциальному зрителю или игроку не должны встречаться области локации, выглядящие пустыми или переполненными. Такие недочёты могут заметно понизить уровень реализма.

Оптимизация – достижение приемлемой производительности на всех целевых платформах при разработке видеоигры. Для сохранения стабильно высокого уровня производительности важно использовать качественные ассеты: 3д модели, использующие оптимальное количество полигонов для нужного уровня детализации. Существует набор способов оптимизации, достигаемых программно:

*Использование LOD (Level Of Detail) -* Для этого необходимо на стадии моделирования создать одну или несколько дополнительных версий модели с пониженной детализацией, а затем, после импорта в движок, настроить зависимость уровня детализации от дистанции между объектом и камерой, либо от процента площади экрана, занимаемой объектом.

*Тайлинг (Tiling) ландшафта –* Техника, похожая на LOD, но относящаяся к 3д моделям ландшафта. Ландшафт делится на квадратные «куски», для каждого из которых создается набор LOD-ов. Правильный LOD отображается в зависимости от расстояния между камерой и куском ландшафта. Такая техника позволяет отображать огромные ландшафты с сохранением производительности.

*Использование технологий микрорельефного текстурирования –* создание рельефа поверхностей без создания большого количества полигонов, а с помощью текстур. Повсеместное использование таких техник сильно повышает детализированность сцены без большого ущерба производительности.

В рамках второй части данной работы рассмотрено создание большой ландшафтной сцены и средневекового поселения с использованием технологий оптимизации*.*

*Часть третья. Разработка процедурной генерации биома.*

При работе с огромными ландшафтами добавление растительности и различных 3D объектов в сцену становится затруднительным и времяемким из-за ручного добавления. Для решения данной проблемы используется процедурная генерация с множеством условий и параметров. При правильной реализации она позволяет создавать целые биомы с внутренними объектами, которые зависят от расположения друг друга. Процедурная генерация является одним из способов оптимизации не только создания ландшафтов, но и оптимизацией производительности в сцене.

*Часть четвертая. Разработка пресонажа окружения.*

Создание персонажей окружения является важным аспектом разработки видеоигр, который оказывает значительное влияние на погружение пользователя и общую реалистичность продукта. Посредством персонажей окружения пользователь может исследовать виртуальный мир. Реалистичность персонажей окружения зависит от использованных 3д моделей и анимаций, что, как правило является задачей художников, работающих над продуктом. Однако то, как анимации будут восприниматься пользователем, зависит от выборов разработчика игрового процесса.

При разработке игрового процесса необходимо интегрировать анимации с логикой игры, чтобы движения персонажа были плавными и отзывчивыми. Важно, чтобы все анимации, были синхронизированы с игровыми событиями и могли быстро реагировать на действия игрока.

В рамках третьей части работы рассмотрен процесс настройки и создания анимаций персонажа окружения, создание базового искусственного интеллекта персонажей окружения.

Цели работы:

* 1. создан природный ландшафт;
  2. ландшафт заполнен различными природными объектами;
  3. создано видео, отражающее результат работы;
  4. создана природная сцена большого размера;
  5. разработан материал ландшафта;
  6. разработано средневековое поселение;
  7. настроено индивидуальное освещение;
  8. оптимизирована производительность полученных проектов;
  9. разработана процедурная генерация биома;
  10. создан анимированный персонажа окружения;
  11. создан искусственный интеллект персонажа окружения.

Для выполнения работы были использованы средства следующих программ:

* Unreal Engine 5 – создание сцен и разработка внутриигровых систем;
* GAEA – создание ландшафтов больших размеров.

1. Создание сцены малого размера

1.1. Создание ландшафта

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

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 1. Панель выбора

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

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

Оба варианта предоставляет первоначальную настройку ландшафта.

1. Material – добавление первоначального материала ландшафта. Им может быть как однотонный материал одного цвета или автоматериал, который в зависимости от высоты рельефа окрашивает поверхность в той или иной цвет.
2. Location – положение созданного ландшафта относительно точки мира с координатами (0, 0, 0).
3. Rotation - поворот созданного ландшафта относительно точки мира с координатами (0, 0, 0)
4. Scale – размер ландшафта. Для варианта Create New размер ландшафта по z не имеет значения. Чаще всего Scale применяется при импортировании карты высот ландшафта.
5. Section Size – размер в квадратах одной секции. Увеличение размера секции применяется для получения большего контроля над созданным ландшафтом.
6. Number of Components – количество «областей/больших квадратов».
7. Overall Resolution – разрешение ландшафта. Для большей оптимизации используется разрешение 1009 x 1009.

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

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 2. Инструмент Manage

Изображение выглядит как снимок экрана, зеленый

Автоматически созданное описание

Рис. 3. Окно просмотра получаемого ландшафта

1.2. Импорт ландшафта

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

Для генерации ландшафта было использовано приложение GAEA, которое использует специальные математические алгоритмы и модели, основанные на фракталах, эрозиях и шумах.

1. Шум Перлина – фрактальный алгоритм, задающий базовую форму ландшафта

где - высота в точке ; - амплитуда шума на -м уровне; - сетка случайных значений; – функция интерполяции (билинейная).

2. Термическая эрозия моделирует осыпание почвы и выравнивание крутых склонов.

где - изменение высоты; - коэффициент эрозии; - высота соседних точек.

3. Гидравлическая эрозия моделирует потки воды, вымывающие почву.

где - новая высота после эрозии; - коэффициент гидравлической эрозии; – объем воды; - градиент высоты (скорость потока воды).

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

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рис. 4. Инструмент Manage при импорте карты высот

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

Для изменения Scale следует воспользоваться следующей формулой:

,

где - в Overall Resolution, – Scale, который использовался в GAEA, – полученный Scale. В самом Unreal Engine метрика измеряется в units, 1 unit 1 см. Таблица рекомендованной размерности ландшафта представлена на рисунке 5.

*Изображение выглядит как текст, снимок экрана

Автоматически созданное описание*

Рис. 5. Рекомендованная размерность ландшафта

1.3. Инструменты редактирования ландшафта

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

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рис. 6. Инструмент Sculpt

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

Рассмотрим варианты изменения ландшафта, которые предлагает инструмент Sculpt.

1.3.1. Изменение карты высот – Sculpt

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

Примеры работы инструмента представлены на рисунках 7, 8.

Изображение выглядит как облако, небо, на открытом воздухе, пейзаж

Автоматически созданное описание

Рис. 7. Результат работы Sculpt

Изображение выглядит как трава, на открытом воздухе, растение, природа

Автоматически созданное описание

Рис. 8. Результат работы Sculpt

1.3.2. Возврат к нулевой карте высот – Erase

Результат работы инструмента Erase представлен на рисунке 9.

Изображение выглядит как на открытом воздухе, облако, небо, трава

Автоматически созданное описание

Рис. 9. Результат работы Erase

1.3.3. Сглаживание - Smooth

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

Изображение выглядит как на открытом воздухе, небо, облако, трава

Автоматически созданное описание

Рис. 10. Результат работы Smooth

1.3.4. Выравнивание - Flatten

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

Изображение выглядит как трава, на открытом воздухе, растение, природа

Автоматически созданное описание

Рис. 11. Результат работы Flatten

1.3.5. Спуск - Ramp

Инструмент Ramp создает спуск между двумя вершинами.

Результат работы инструмента Ramp представлен на рисунке 12.

Изображение выглядит как трава, на открытом воздухе, вода, Водоросли

Автоматически созданное описание Изображение выглядит как трава, снимок экрана, растение, на открытом воздухе

Автоматически созданное описание

Рис. 12. Результат работы Ramp

1.3.6. Термическая эрозия – Thermal Erosion

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

Изображение выглядит как на открытом воздухе, облако, небо, пейзаж

Автоматически созданное описание Изображение выглядит как на открытом воздухе, небо, пейзаж, облако

Автоматически созданное описание

Рис. 13. Результат работы Thermal Erosion

1.3.7. Гидроэрозия – Hydro Erosion

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

Изображение выглядит как растение, природа, пейзаж, трава

Автоматически созданное описание

Рис. 14. Результат работы Hydro Erosion

1.3.8. Noise

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

Изображение выглядит как кратер, трава, на открытом воздухе, природа

Автоматически созданное описание

Рис. 15. Результат работы Noise

1.4. Растительность

Для добавления растительности в сцену движок Unreal Engine 5 имеет множество инструментов, например, Foliage mode.

Для добавления растительности движок Unreal Engine имеет специальный инструмент, Foliage mode, который основан на следующих методах:

1. Распределение объектов (Poisson Disk Sampling) используется для равномерного распределения объектов растительности без пересечений.

где - координаты точек распределения объектов; - расстояние между двумя точками;

– минимальное расстояние между объектами.

1. Гармонический осциллятор используется для анимации растительности под воздействием ветра.

где - смещение объекта в момент времени ; – начальная амплитуда; – коэффициент затухания; – частота колебаний; – фазовый сдвиг.

1. Рассеивание света (Subsurface Scattering (SSS)) используется для пропуска части света через листву.

где - входящий и выходящий свет; - функция рассеивания; – нормаль поверхности;

- входящее и исходящее направления света; – точка на поверхности; – сфера направления, откуда может приходить свет .

Для выбора данного инструмента в панели выбора следует переключить режим на Foliage mode. Выбор режима Foliage mode представлен на рисунке 16.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 16. Выбор Foliage mode

В появившемся окне Foliage mode для возможности нанесения растительности следует перенести статическую сетку модели в поле Foliage. Нанесение растительности на поверхность ландшафта происходит за счет использования кисти, параметры которой можно редактировать. Окно Foliage mode представлено на рисунке 17.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 17. Окно Foliage mode

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

1.4.1. Painting

Настройка Painting используется для изменения параметров «рисования» на какой-либо области. Она включает в себя настройку плотности, размера. Настройка Painting представлена на рисунке 18.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рис. 18. Настройка Painting

1.4.2. Placement

Немаловажной настройкой Foliage mode является настройка позиционирования объекта. В данной категории представляется возможность выравнивания объектов по нормали, добавления смещения по оси Z, которое используется для того, чтобы объект полностью находился на поверхности ландшафта. Настройка Placement представлена на рисунке 19.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рис. 19. Настройка Placement

1.4.2. Instance Static

Данная настройка предназначена для изменения теней, коллизий объекта. Настройка Instance Static представлена на рисунке 20.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 20. Настройка Instance Static

1.5. Настройка неба

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

HDRIBackdop является Actor-ом, поэтому для добавления его в сцену в панели быстрого добавления следует выбрать Place Actor Panel. Окно быстрого добавления представлена на рисунке 21.

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

Рис. 21. Панель быстрого добавления

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

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

1.6. Создание тумана

Для большей реалистичности полученной сцены был создан материал тумана. Схема полученного материала тумана представлена на рисунке 22.

Изображение выглядит как снимок экрана, текст, диаграмма, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 22. Материал тумана

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

1.7. Создание кат-сцены

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

Для создания кат-сцены движок Unreal Engine предоставляет Level Sequence Actor, который находится в панели добавления Actor. На рисунке 23 представлен интерфейс редактора Sequencer.

Изображение выглядит как Мультимедийное программное обеспечение, Графическое программное обеспечение, программное обеспечение, Редактирование

Автоматически созданное описание

Рис. 23. Интерфейс редактора Sequencer

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

2. Создание большой ландшафтной сцены и средневекового поседения

2.1. Материал ландшафта

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

Одними из основных методов нанесения определенного материала на ландшафт является:

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

Ручная раскраска слоев ландшафта представляет собой создание мастер материала, который будет содержать в себе несколько материалов с различными текстурами, которые подключены к Layer. Layer используется для создания возможности содержания множества материалов и возможности нанесения каждого их них на ландшафт. Данный способ не является эффективным, т к придется вручную прокрашивать каждый слой.

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

Разработанный авто-материал соответствия материала с высотой ландшафта представлен на рисунке 24.

Изображение выглядит как снимок экрана, схема, диаграмма

Автоматически созданное описание

Рис. 24. Авто-материал соответствия материала с высотой ландшафта.

Сами материалы A, B, C, D, E реализованы по следующему принципу, представленном на рисунке 25.

Изображение выглядит как снимок экрана, 3D-моделирование, Графическое программное обеспечение, Цифровая сборка

Контент, сгенерированный ИИ, может содержать ошибки.

Рис. 25. Принцип реализации материалов A, B, C, D, E.

В результате создания авто-материала был получен Material Instance, который возможно использовать для текстурирования ландшафта. На рисунке 26 представлены полученные возможности созданного Material Instance.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рис. 26. Возможности Material Instance

2.2. Средневековое поселение

Для создания средневекового поселения были использованы:

1. Библиотека Quixel Megascans
2. Marketplace Fab

Созданные 3D модели зданий представлены на рисунке 27.

Изображение выглядит как строительство, дом, трава, на открытом воздухе

Автоматически созданное описание

Рис. 27. 3D модели зданий

Для текстурирования полученных 3D объектов были созданы необходимые материалы стен и крыши. Примеры созданных материалов представлено на рисунках 28 и 29.

Изображение выглядит как текст, снимок экрана, Графическое программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 28. Материал стены

Изображение выглядит как снимок экрана, текст, диаграмма, карта

Автоматически созданное описание

Рис. 29. Материал крыши

2.3. Водные объекты

Unreal Engine 5 имеет инструменты для добавления и последующей работы с водными объектами. Одним из решений добавления водных объектов является подключение плагина Water. Данный плагин предоставляет набор водных инструментов и методов рендеринга, позволяющих легко добавлять океаны, реки, озера или пользовательские водоемы, которые создаются в ландшафте и взаимодействуют с игровым процессов.

Для добавления водного объекта с использованием плагина в меню Place Actor Panel -> Water следует выбрать конкретный водный объект, который должен быть добавлен в сцену. Список доступных водных объектов и результат работы Water River представлены на рисунках 30 и 31.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 30. Список доступных водных объектов при подключении Water плагина.

Изображение выглядит как вода, природа, дюна, пляж

Автоматически созданное описание

Рис. 31. Результат работы Water River

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

Для адаптации водного объекта в сцену был изменен водный материал, предоставляемый плагином. Полученный материал представлен на рисунке 32.

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

Рис. 32. Материал воды

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

2.4. Настройка освещения

Движок Unreal Engine имеет обязательные компоненты сцены для настройки освещения.

1. Направленный свет (DirectionalLight)

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

1. Туман (ExponentialHeightFog)

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

1. Атмосфера неба (SkyAtmosphere)

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

1. Небесный свет (SkyLight)

Захватывает свет неба и освещает текстурой неба всю сцену.

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

2.5. Оптимизация

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

LOD означает уровень детализации. Для одного объекта может быть несколько уровней детализации. Самый низкий уровень детализации (LOD0) является наиболее подробным, когда вы находитесь близко к объекту. Наивысший уровень детализации (LODx) обеспечивает наименьшее количество деталей, когда объект находится очень далеко от камеры и вы не можете разглядеть детали. Unreal поддерживает 8 уровней детализации, так что между самым низким и самым высоким уровнями детализации может быть множество шагов.

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

Изменение минимального LOD-а представлено на рисунке 33.

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 33. Изменение минимального LOD

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

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

Таблица 1. Сравнение производительности при использовании LOD

3. Разработка процедурной генерации биома

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

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

PCG в Unreal Engine представляется в виде графа, который начинается с получения информации о ландшафтt и заканчивающийся спавнером статических сеток различных объектов.

где - базовое расположение точки; - шум, добавляющий детали.

Пример простейшей реализации PCG представлена на рисунке 34.

Изображение выглядит как снимок экрана, 3D-моделирование, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 34. Простейший пример PCG

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

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

Пример работы данного PCG Graph представлен на рисунке 35. Для наглядности блок Grid Size отключен.

Изображение выглядит как растение

Автоматически созданное описание

Рис. 35. Пример работы PCG Graph

Полученный результат не является достаточным ведь возможно, что внутри леса находится какие-либо другие объекты, например озеро или город. Сама область внутреннего объекта может быть задана при помощи сплайна. Дополненный PCG Graph леса представлен на рисунке 36.

Изображение выглядит как снимок экрана, 3D-моделирование, Компьютерная игра, Программное обеспечение для видеоигр

Автоматически созданное описание

Рис. 36. Дополненный PCG Graph леса

Дополнением является то, что теперь PCG Graph леса получает информацию о находящемся сплайне внутри и спавнит объекты за пределами сплайна в своей допустимой области. Для реализации самого поселения необходимо создать еще один PCG Graph, который будет спавнить объекты внутри определенной области, а именно созданного сплайна. Для большей реализации и возможностей также был добавлен сплайн дороги. Реализация PCG Graph поселения представлена на рисунке 37.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рис. 37. Реализация PCG Graph поселения

Результат работы данных PCG Graph-ов представлен на рисунке 38.

Изображение выглядит как червь, шаблон

Автоматически созданное описание со средним доверительным уровнем

Рис. 38. Результат работы PCG Graph-ов.

4. Разработка персонажа окружения

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

Для реализации NPC было выбрано три его основных поведения:

1. Find Random Location – данное поведение реализуется случайную ходьбу по карте неигрового персонажа.
2. Find Player Location – поведение, которое реализует нахождение главного персонажа на сцене. Персонаж будет являться найденным если NPC его увидел в определенном поле своего зрения.
3. Chasing Player – поведение реализующее преследование главного персонажа. Преследование продолжается до тех пор, по NPC видит главного персонажа в своей определенной области.

Разработанное дерево поведения NPC представлено на рисунке 39.

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, 3D-моделирование

Автоматически созданное описание

Рис. 39. Дерево поведения персонажа окружения

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

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

Блок-схема персонажа окружения представлена на рисунке 40.

Изображение выглядит как диаграмма, План, Технический чертеж, линия

Автоматически созданное описание

Рис. 40. Блок-схема персонажа окружения

Листинг 1. Реализация функции поиска случайно локации

|  |
| --- |
| EBTNodeResult::Type UBTTask\_FindRandomLocation::ExecuteTask(UBehaviorTreeComponent& OwnerComp, uint8\* NodeMemory)  {  // get AI controller and it`s NPC  if (auto\* const cont = Cast<AChasingNPC\_AIController>(OwnerComp.GetAIOwner()))  {  if (auto\* const npc = cont->GetPawn())  {  // obtain npc location to use as an origin  auto const Origin = npc->GetActorLocation();  // get the navigation system and generate a random location  if (auto\* const NavSys = UNavigationSystemV1::GetCurrent(GetWorld()))  {  FNavLocation Loc;  if (NavSys->GetRandomPointInNavigableRadius(Origin, SearchRadius, Loc))  {  OwnerComp.GetBlackboardComponent()->SetValueAsVector(GetSelectedBlackboardKey(), Loc.Location);  }  // finish with success  FinishLatentTask(OwnerComp, EBTNodeResult::Succeeded);  return EBTNodeResult::Succeeded;  }  }  }  return EBTNodeResult::Failed;  } |

Листинг 2. Функция, реализующая поиск главного персонажа

|  |
| --- |
| EBTNodeResult::Type UBTTask\_FindPlayerLocation::ExecuteTask(UBehaviorTreeComponent& OwnerComp, uint8\* NodeMemory)  {  // get player character  if (auto\* const Player = UGameplayStatics::GetPlayerCharacter(GetWorld(), 0))  {  // get player location to use as an origin  auto const PlayerLocation = Player->GetActorLocation();  if (SearchRandom)  {  FNavLocation Loc;  // get the navigation system and generate a random location near the player  if (auto\* const NavSys = UNavigationSystemV1::GetCurrent(GetWorld()))  {  // try to get a random location near the player  if (NavSys->GetRandomPointInNavigableRadius(PlayerLocation, SearchRadius, Loc))  {  OwnerComp.GetBlackboardComponent()->SetValueAsVector(GetSelectedBlackboardKey(), Loc.Location);  FinishLatentTask(OwnerComp, EBTNodeResult::Succeeded);  return EBTNodeResult::Succeeded;  }  }  }  else  {  OwnerComp.GetBlackboardComponent()->SetValueAsVector(GetSelectedBlackboardKey(), PlayerLocation);  FinishLatentTask(OwnerComp, EBTNodeResult::Succeeded);  return EBTNodeResult::Succeeded;  }  }  return EBTNodeResult::Failed;  } |

Листинг 3. Функция, реализующая преследование главного персонажа

|  |
| --- |
| EBTNodeResult::Type UBTTask\_ChasePlayer::ExecuteTask(UBehaviorTreeComponent& OwnerComp, uint8\* NodeMemory)  {  // get target location from blackboard via the NPC`s controller  if (auto\* const cont = Cast<AChasingNPC\_AIController>(OwnerComp.GetAIOwner()))  {  auto const PlayerLocation = OwnerComp.GetBlackboardComponent()->GetValueAsVector(GetSelectedBlackboardKey());  // move to the player`s location  UAIBlueprintHelperLibrary::SimpleMoveToLocation(cont, PlayerLocation);  //finish with success  FinishLatentTask(OwnerComp, EBTNodeResult::Succeeded);  return EBTNodeResult::Succeeded;  }  return EBTNodeResult::Failed;  } |

Результаты

В ходе выполнения работы были получены две природные сцены:

1. Реалистичный пейзаж, представленный на рисунках 41, 42;
2. Сцена размером 4х4 км с детализированным средневековым поселением, представленная на рисунках 43, 44.

**Изображение выглядит как облако, пейзаж, на открытом воздухе, трава

Автоматически созданное описание**

Рис. 41. Реалистичный пейзаж

**Изображение выглядит как на открытом воздухе, небо, пейзаж, трава

Автоматически созданное описание**

Рис. 42. Реалистичный пейзаж

Изображение выглядит как на открытом воздухе, облако, небо, растение

Автоматически созданное описание

Рис. 43. Средневековое поселение

Изображение выглядит как на открытом воздухе, небо, дерево, облако

Автоматически созданное описание

Рис. 44. Средневековое поселение

# **ЗАКЛЮЧЕНИЕ**

В рамках выпускной квалификационной работы были выполнены следующие задачи:

* 1. создан природный ландшафт;
  2. ландшафт заполнен различными природными объектами;
  3. создано видео, отражающее результат работы;
  4. создана природная сцена большого размера;
  5. разработан материал ландшафта;
  6. разработано средневековое поселение;
  7. настроено индивидуальное освещение;
  8. оптимизирована производительность полученных проектов;
  9. разработана процедурная генерация биома;
  10. создан анимированный персонажа окружения;
  11. создан искусственный интеллект персонажа окружения.

В результате работы были рассмотрены подходы создания реалистичного пейзажа местности, средневекового поселения и персонажа окружения. Изучены принципы создания материалов, процедурной генерация биомов, водных объектов, кат-сцены и настройки освещения. Рассмотрены варианты оптимизации. Изучены методы работы с плагинами Unreal Engine. Изучены средства разработки на языке C++, технология Blueprints предоставляемые движком Unreal Engine 5. Приобретены знания об организации проектов.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Epic Games Unreal Engine Documentation / Epic Games [Электронный ресурс] // Epic Games Developer: [сайт]. — URL: https://dev.epicgames.com/documentation/ru-ru/unreal-engine/unreal-engine-5-5-documentation (дата обращения: 15.10.2024);
2. Unreal Engine Building natural environments / Unreal Engine [Электронный ресурс] // YouTube: [сайт]. — URL: https://www.youtube.com/watch?v=gbj1qgPOl3E (дата обращения: 19.10.2024);
3. Том Шэннон Unreal Engine 4 для дизайна и визуализации [Текст] / Том Шэннон — 1-е изд.. — Москва: Bombora, 2021 — 368 c;
4. Epic Games Unreal Engine / Epic Games [Электронный ресурс] // GitHub: [сайт]. — URL: https://github.com/EpicGames (дата обращения: 7.11.2024);
5. Максименкова Ольга Вениаминовна, Веселко Никита Игоревич Программирование в Unreal Engine 5. Основы визуального языка Blueprint. [Текст] / Максименкова Ольга Вениаминовна, Веселко Никита Игоревич — 1-е изд.. — Москва: Эксмо, 2023 — 320 c.