Визуализация природных ландшафтов и средневекового поселения на Unreal Engine 5

Студент: Колыхалов Д. В. РК6-84Б

Научный руководитель: Витюков Ф. А.



Введение – Актуальность работы

В современном мире 3D-сцены с рендерингом на Unreal Engine используются в различных сферах, например кинематографе, архитектурных проектах, телевидении, авиационных тренажёрах. Основной проблемой при создании таких сцен является поиск баланса между детализацией самой сцены и оптимизацией 3D-моделей и ландшафта для возможности рендеринга в реальном времени.

Актуальность работы заключается в следующем:

- 1. Созданные 3D сцены с рендерингом на Unreal Engine используются в различных сферах;
- 2. Разработка собственного материала ландшафта позволяет полностью автоматизировать возможность нанесения материала на слои ландшафта;
- 3. Моделирование средневекового поселения способствует сохранению и популяризации культурного наследия, позволяя пользователю погрузиться в атмосферу прошлого средствами интерактивных технологий;
- 4. Работа охватывает сразу несколько направлений: моделирование окружающей среды, дизайн архитектурных объектов, создание анимаций и механик персонажей, что отражает современные требования к специалистам в цифровом производстве;
- 5. Разработанные сцены могут быть использованы для будущих приложений в сферах виртуальной и дополненной реальностях.

Введение – Актуальность работы

Заинтересованность государства в исторических и других интерактивных проектах с трёхмерной графикой. Финансирование ведётся посредством Института Развития Интернета.





Рисунок 1 – Исторический проект СМУТА

Постановка задачи

Цель работы: Создание сцен с природным ландшафтом и средневековым поселением на Unreal Engine 5.

Задачи:

- 1. Создать детализированный природный ландшафт: сгенерировать полигональную сетку карты высот и разработать материал ландшафта;
- 2. Заполнить ландшафт различными природными объектами;
- 3. Создать ландшафт большого размера и средневековое поселение: создать и корректно разместить объекты, разработать материалы;
- 4. Настроить освещение и провести оптимизацию производительности;
- 5. Создать анимированных персонажей окружения;
- 6. Разработать боевую систему для пользователя и персонажей окружения.

Создание ландшафта

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

$$P(x,y) = \sum_{i=0}^{\infty} a_i \cdot \text{interp}(g_i(x,y)),$$

где P(x,y) – высота в точке (x,y); a_i – амплитуда шума на i-м уровне; $g_i(x,y)$ – сетка случайных значений; interp – функция интерполяции (билинейная).

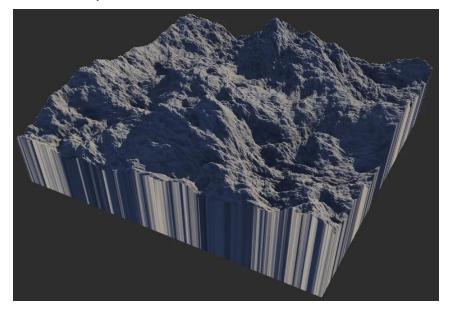


Рисунок 2 - Полученная первоначальная форма ландшафта

Создание ландшафта

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

$$\Delta h = k_i \cdot (h_i - h_j), h_i > h_j,$$

где Δh — изменение высоты; k_i — коэффициент эрозии; h_i , h_i — высота соседних точек.

3. Гидравлическая эрозия моделирует потоки воды, вымывающие почву.

$$h' = h - k_h \cdot \omega \cdot \nabla h,$$

где h' — новая высота после эрозии; k_h — коэффициент гидравлической эрозии; ω — объем воды; ∇h — градиент высоты (скорость потока воды).

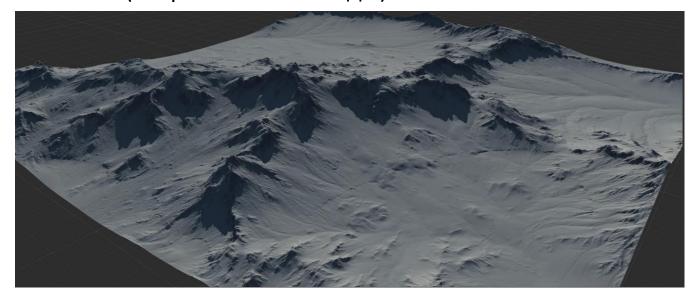


Рисунок 3 - Измененная форма ландшафта после термической и гидравлической эрозий

Создание материала ландшафта

Задача создания материала ландшафта заключалась в следующих этапах:

- 1. Создание отдельных материалов для определенного слоя ландшафта;
- Использование высоты ландшафта для корректного определения позиционирования отдельного материала слоя ландшафта;
- 3. Возможность быстрого редактирования параметров материала;
- 4. Использование виртуального текстурирования для оптимизации сложности шейдеров (shader complexity).

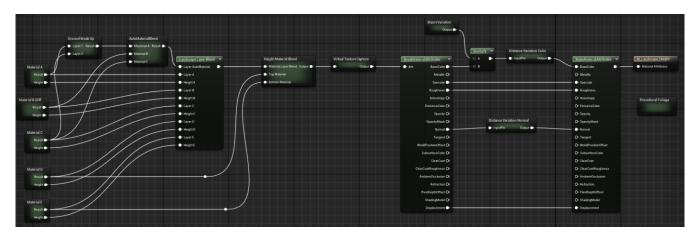


Рисунок 4 - Созданный материал ландшафта

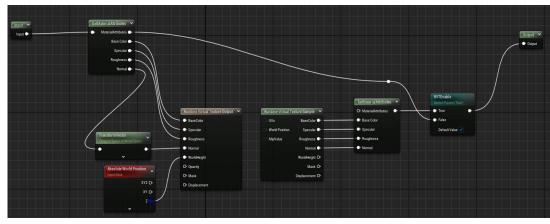


Рисунок 5 – Функция виртуального текстурирования

Растительность

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.

$$d(p_i, p_j) \ge r, \quad \forall i, j, i \ne j,$$

где p_i, p_j – координаты точек распределения объектов; $d(p_i, p_j)$ – расстояние между двумя точками;

r – минимальное расстояние между объектами.

2. Гармонический осциллятор используется для анимации растительности под воздействием ветра. $x(t) = A \cdot e^{-\gamma t} \cos(\omega t + \phi)$,

где x(t) – смещение объекта в момент времени t; A – начальная амплитуда; γ – коэффициент затухания; ω – частота колебаний; ϕ – фазовый сдвиг.

Растительность

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

$$L_o(p,\omega_o) = \int_{\Omega} R_s(p,\omega_i,\omega_o) \cdot L_i(p,\omega_i) \cdot (\mathbf{n} \cdot \omega_i) d\omega_i,$$

где L_i, L_o — входящий и выходящий свет; Ω — сфера направления, откуда может приходить свет ω_i ; R_s — функция рассеивания; p — точка на поверхности; ω_i, ω_o — входящее и исходящее направления света; \mathbf{n} — нормаль поверхности.



Рисунок 6 - Рассеивание света

Водные объекты

1. Спекулярное отражение (модель Фонга)

$$I_{spec} = k_s (\mathbf{R} \cdot \mathbf{V})^n,$$

где k_s — коэффициент зеркального отражения; **R** — отраженный вектор света; **V** — вектор к камере (нормализованный); n — степень блеска (чем выше, тем острее блик).

2. Эффект Френеля – реалистичное смешивание отражения и преломления.

$$F(\theta) = F_0 + (1 - F_0)(1 - \cos \theta)^5,$$

где θ — угол между направлением взгляда и нормалью поверхности; F_0 — отражение при нормальном падении (θ = 0).

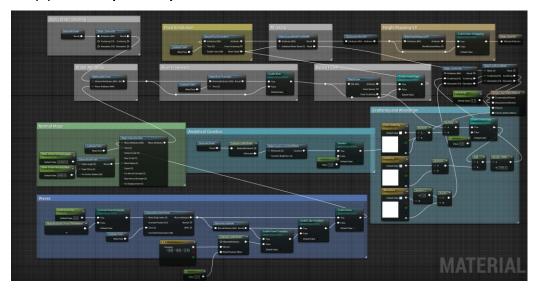


Рисунок 7 - Материал водного объекта



Рисунок 8 - Водный объект

Процедурная генерация

Задача создания процедурной генерации заключалась в следующих этапах:

- 1. Получение данных о ландшафте;
- 2. Рандомизация появляющихся объектов;
- 3. Получении информации о внутренней генерации;
- 4. Выбор и добавление объектов генерации.

$$H(x,y) = f(x,y) + g(x,y),$$

где f(x,y) - базовое расположение точки; g(x,y) - шум, добавляющий детали.

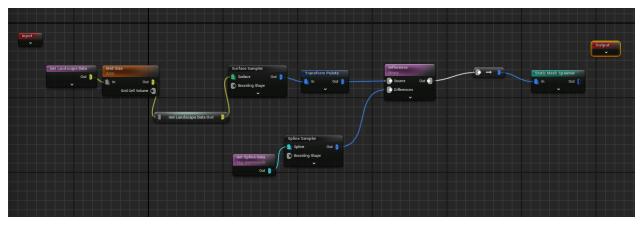


Рисунок 9 - Граф внешней процедурной генерации леса

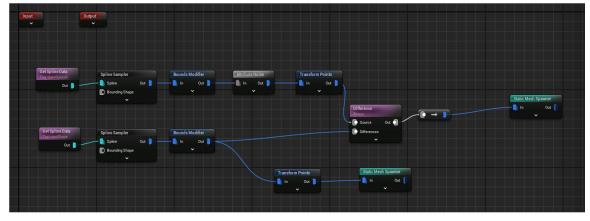


Рисунок 10 - Граф внутренней процедурной генерации

Средневековое поселение

Для создания средневекового поселения были выделены следующие участки:

- 1. Жилая зона;
- 2. Зона торговли;
- 3. Зона скотоводства и земледелия.



Рисунок 11 - 3D модели жилой зоны



Рисунок 12 - 3D модели торговой зоны

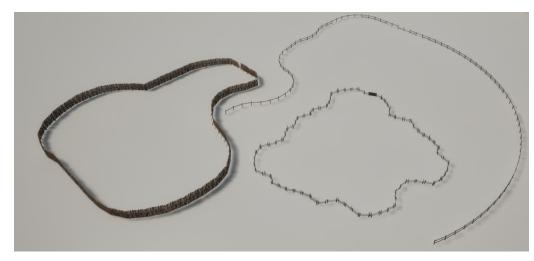


Рисунок 13 - 3D модели зоны скотоводства

Настройка освещения сцены

Движок Unreal Engine имеет обязательные компоненты сцены для настройки освещения.

- 1. Направленный свет (DirectionalLight);
- 2. Туман (ExponentialHeightFog);
- 3. Атмосфера неба (SkyAtmosphere);
- 4. Небесный свет (SkyLight).



Рисунок 14 - Компоненты освещения сцены

Оптимизация

Для решения проблемы производительности:

- 1. Использовался метод уровней детализации (LOD), который основан на уменьшении количества полигонов 3D объекта;
- 2. Проведена оптимизация сложности шейдеров (shader complexity).

Таблица 1 - Сравнение производительности разных LOD

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



Рисунок 15 – Результат оптимизации shader complexity

Персонаж окружения

Для реализации персонажа окружения было создано дерево поведения со следующими возможностями:

- 1. Find Random Location поиск случайной;
- 2. Find Player Location поиск главного персонажа;
- 3. Chasing Player преследование главного персонажа.

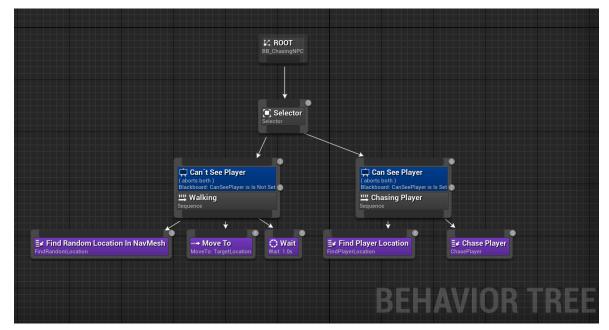


Рисунок 16 - Дерево поведения персонажа окружения

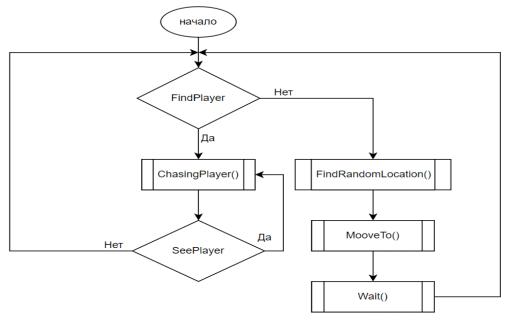


Рисунок 17 – Блок-схема персонажа окружения

Боевая система

Для реализации боевой системы были выполнены следующие действия:

- 1. Настроены анимации перемещения персонажа;
- 2. Разработаны механики захвата цели, комбинации атак, системы здоровья, уклонений, выносливости, инвентаризации;
- 3. Разработаны эффекты и звуковое сопровождение боевой системы;
- 4. Разработаны вражеские персонажи окружения.

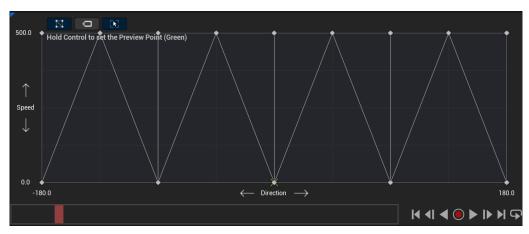


Рисунок 18 – Граф анимаций перемещений

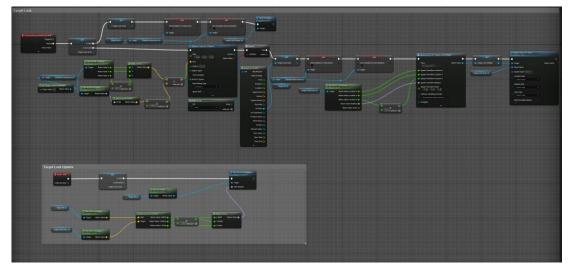


Рисунок 19 – Механика захвата цели

Боевая система

Для реализации механики системы инвентаризации были разработаны следующие классы:

- 1. UBaseOfInventoryItem класс получения данных о подбираемом предмете из таблицы данных;
- 2. IInterfaceForInteractionWithObject класс для взаимодействия с объектом на сцене;
- 3. AltemPickup класс для представления объекта на сцене;
- 4. UComponentOfPlayerInventory класс, реализующий основные функции системы инвентаризации;
- 5. Классы для реализации визуальной составляющей системы инвентаризации (HUD, MainMenu, InteractionWidget, MainInventory, InventoryItemSlot, InventoryHint).

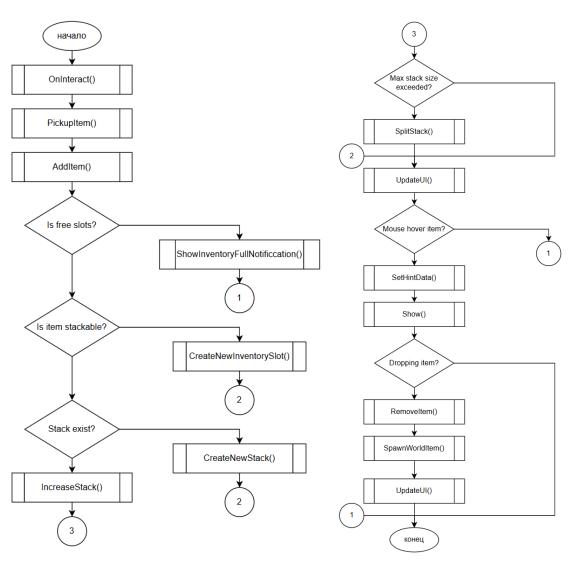


Рисунок 20 – Блок-схема системы инвентаризации

Результаты разработки



Рисунок 21 - Детализированная природная сцена

Результаты разработки



Рисунок 22 – Вид на средневековое поселение

Результаты разработки



Рисунки 23-25 – Персонажи окружения, боевая система и элементы инвентаря





Заключение

В рамках выполнения работы достигнута поставленная **цель** создания сцен с природным ландшафтом и средневековым поселением на Unreal Engine 5.

Для достижения цели были выполнены следующие задачи:

- 1. Создана детализированная сцена природного ландшафта: сгенерирована полигональная сетка карты высот, разработан материал ландшафта;
- 2. Ландшафт заполнен различными природными объектами;
- 3. Создан ландшафт большого размера и средневековое поселение: созданы и корректно размещены объекты, разработаны материалы.
- 4. Настроено освещение и проведена оптимизация производительности;
- 5. Созданы анимированные персонажи окружения;
- 6. Разработана боевая система для пользователя и персонажей окружения.