Санкт-Петербургский государственный электротехнический университет им. В.И. Ульянова (Ленина)

Разработка алгоритма генерации ландшафта на основе графа связей трехмерных объектов

Выполнил: Скиба Антон Сергеевич, гр. 5304
Руководитель: Геппенер Владимир Владимирович, д.т.н., профессор
Консультант: Шевская Наталья Владимировна, асс. каф. МО ЭВМ

Санкт-Петербург, 2021

Цель и задачи

Актуальность: средства генерации ландшафтов

- Ручное размещение объектов на ландшафте
- Отсутствие программного интерфейса (API)
- Статические размеры ландшафта

Цель: автоматизировать размещение объектов на процедурно созданных участках ландшафта.

Задачи:

1. Провести сравнение аналогов
2. Выбрать и реализовать архитектуру приложения
3. Разработать программный интерфейс
4. Разработать пользовательский интерфейс
5. Исследовать свойства решения

Сравнение аналогов

Обозначения:

Импорт объектов – НП (не поддерживается) / РК (ручное размещение) / АК (автоматическое размещение)

API – РВ (в реальном времени) / ДО (длительные операции) / Нет

Критерий / Сервис	Устойчивость	Импорт объектов	Размер	Доп. ПО	API
World Creator	Да	НП	$2 * 10^8$ x $2 * 10^8$ px	Да	Нет
Instant Terra	Нет	НП	64 000 x 64 000 px	Нет	Нет
World Machine	Да	РК	12 000 x 12 000 px	Нет	ДО
3D Map Generator	Да	НП	1 750 x 1 100 px	Да	Нет

- Устойчивость ко входным данным
- Возможность загрузки объектов
- Предоставление программного интерфейса (API)

Архитектура приложения

Основа – клиент серверная архитектура
Клиент – компонентный подход
Сервер – объектно-ориентированный подход

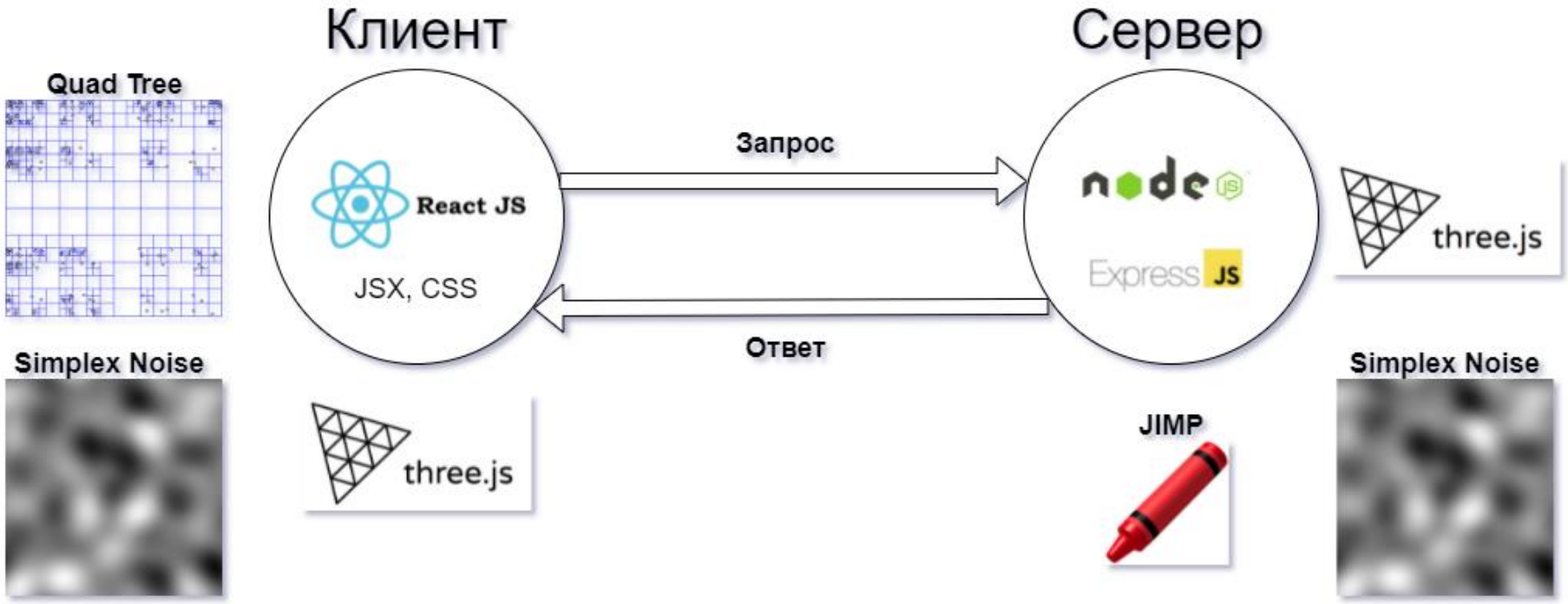

Рисунок 1 – Архитектура приложения

Программный интерфейс (API)

UML–диаграмма классов

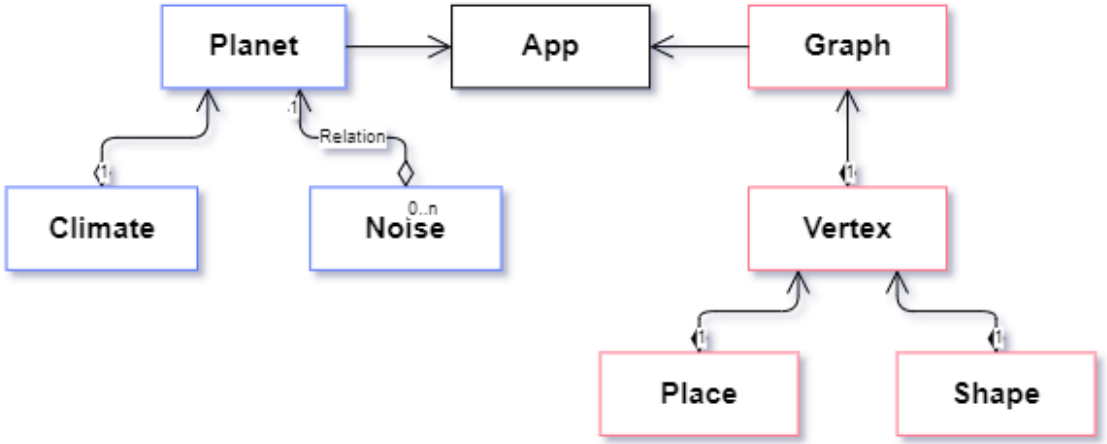

Рисунок 2 – UML-диаграмма серверной стороны

Серверная часть делится на две категории

1. Обработка взаимодействия с графом: Graph, Vertex, Place, Shape.
2. Формирование ландшафта: Planet, Climate, Noise.

Программный интерфейс (API), климатическая карта

Алгоритм формирования климатической карты на основе графа.

Граф Диаграмма Вороного

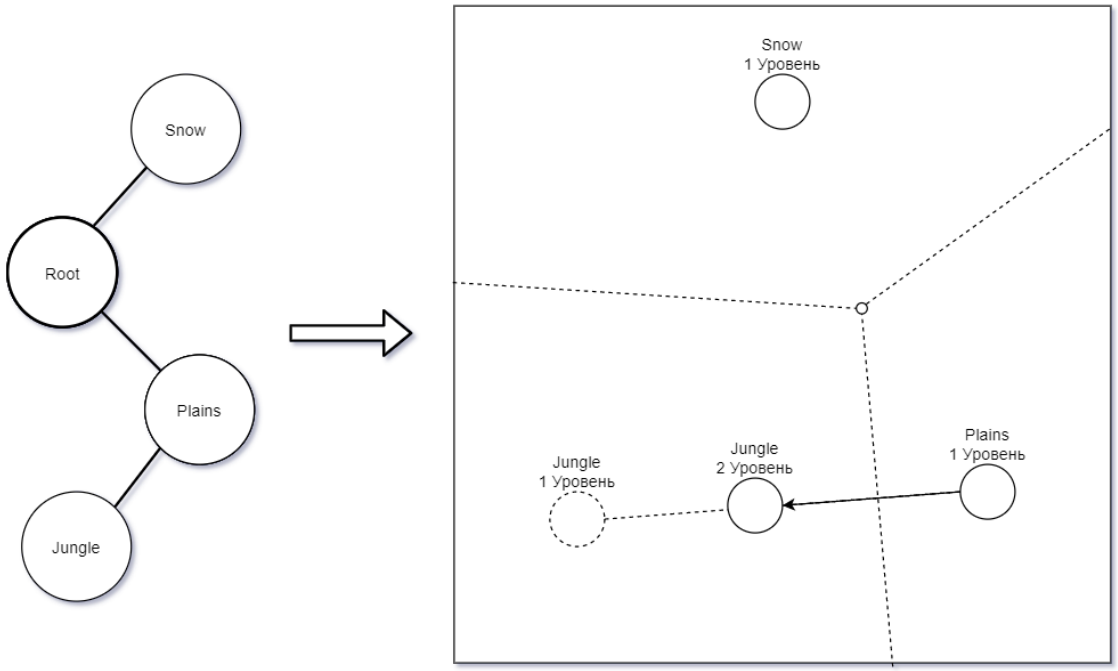

Рисунок 3 – Алгоритм создания климатической карты

Метрика Минковского: $\mathbf{d}(\boldsymbol{p},\boldsymbol{q}) = \left(\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} |\boldsymbol{p_i} - \boldsymbol{q_i}|^{\boldsymbol{k}}\right)^{\frac{\mathbf{1}}{\boldsymbol{k}}}$, где

$k = 1$: Манхэттенское расстояние: $\mathrm{d}(p,q) = \left|\left|p - q\right|\right| = \sum_{i=1}^{n} |p_i - q_i|$

$k = 2$: Евклидова метрика: $\mathrm{d}(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$

Программный интерфейс (API), карта высот

Формирование ландшафта. Основная карта высот. Класс “Noise”.

Параметры:

- Зерно (seed)
- Частота (frequency) и октавы (octaves)

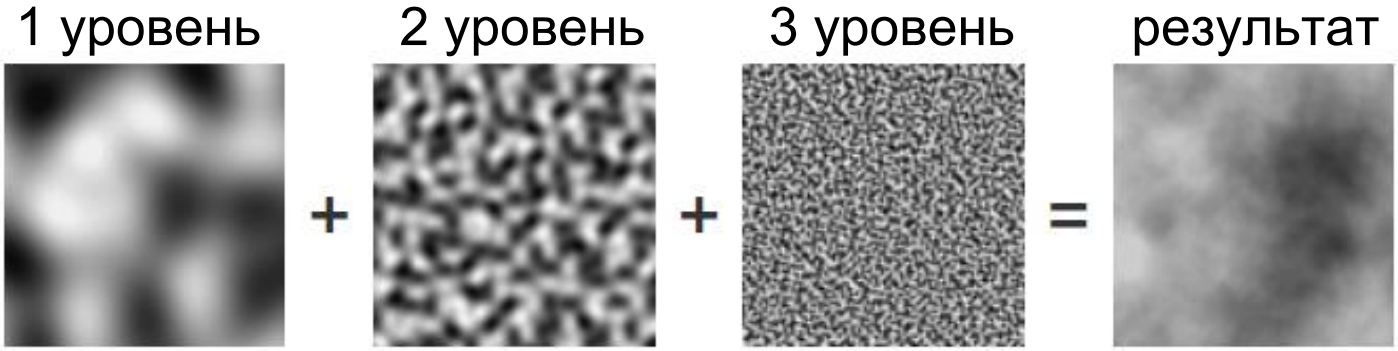

Рисунок 4 – Алгоритм наложения шума с разной частотой

- Ровность (flatness): $V_{x,y,z}^{i,j} = (V_{x,y,z}^{i,j})\text{^}f$, где $V_{x,y,z}^{i,j} -$ значение шума в точке пространства $\{x, y, z\}$ для (i, j) шага участка, f $-$ значение параметра

Программный интерфейс (API), алгоритм размещения объектов

Параметры:

- Кластеризация (clustering)
- Насыщенность (saturation)
- Заполнение (fullness)
- Биом (zone)

Алгоритм:

1. Значение шума: $S_{x,y,z}^{i,j} = Noise_{x,y,z}(seed = ID, frequency = \frac{1}{clust})$, где ID − идентификатор фигуры, $clust$ − значение кластеризации, $Noisex,y,z$ − значение шума в точке пространства {x, y, z} для (i, j) шага участка.
2. Дистанция: $D = \max(0.01, 1 - sat) * 4^l$, где l − уровень детализации, а sat − значение насыщенности
3. Тестирование значения $check = test(S_{x,y,z}^{i,j}, D, F, Z)$, где F − значение заполнения, Z − тип биома. Проверяется по 3 критериям
    1. Дистанция: $i * j \,\%\, D == 0$
    2. Заполнение: $S_{x,y,z}^{i,j} > F$
    3. Биом: $Z_{x,y,z} == Z_{ID}$
4. If (*check*) then return *ID* else return *False*

Пользовательский интерфейс

Делится на две основные страницы:

- Конфигурация графа, рис. 5
- Визуальное представление ландшафта, рис. 6

Рисунок 5 – Страница графа

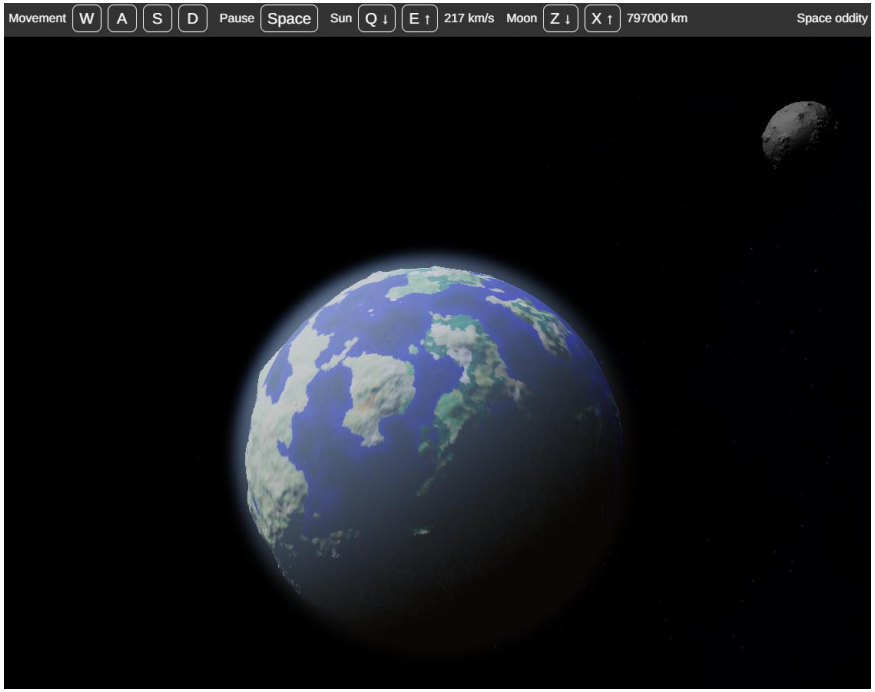

Рисунок 6 – Страница визуализации

Пользовательский интерфейс, компоненты

Конфигурация графа

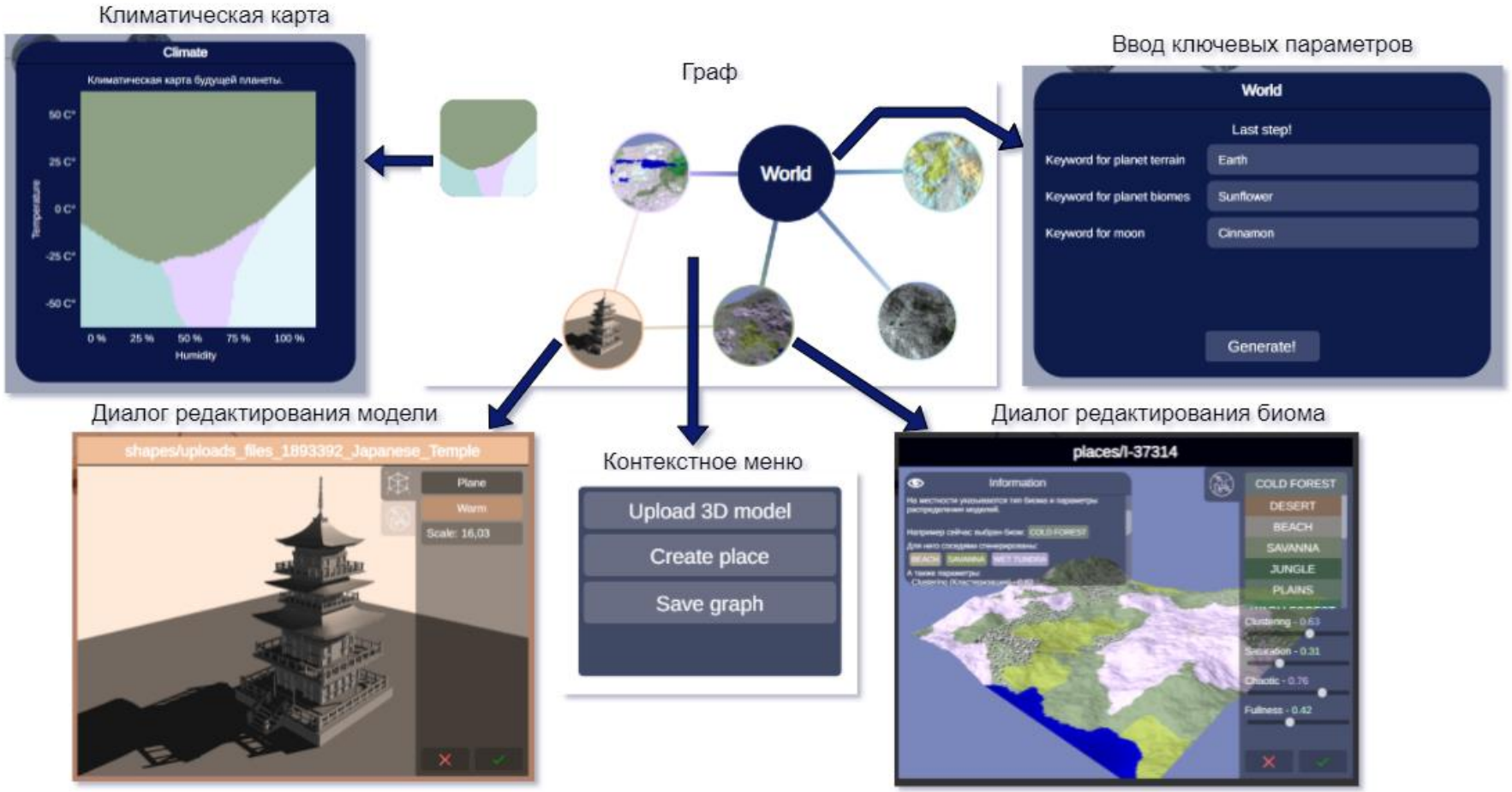

Рисунок 7 – Основные компоненты

Пользовательский интерфейс, алгоритмы

Визуальное представление ландшафта.

1) Для оптимизации рендеринга ландшафта используется структура дерева квадрантов, рис. 8.

Задача вставки элемента: $\boldsymbol{O(\log n)}$

Количество потенциальных участков: $\boldsymbol{4^l}$, где l – уровень детализации.

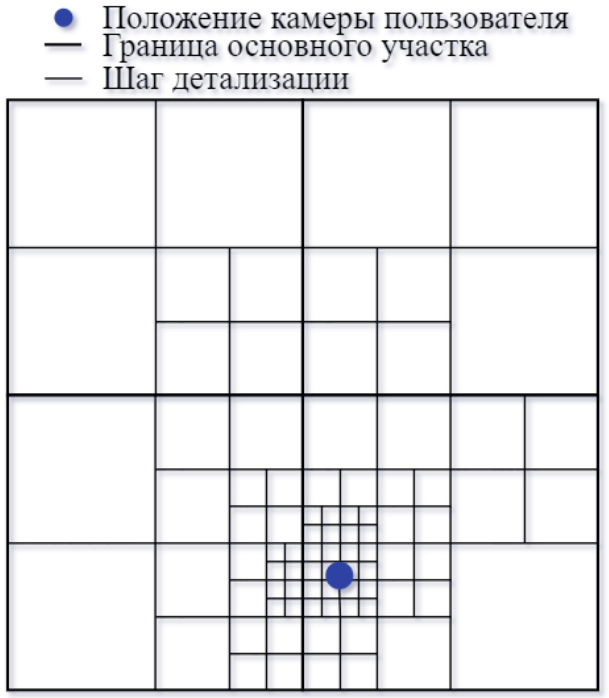

Рисунок 8 – Дерево квадрантов

2) Преобразование куба в сферу, рис. 9.

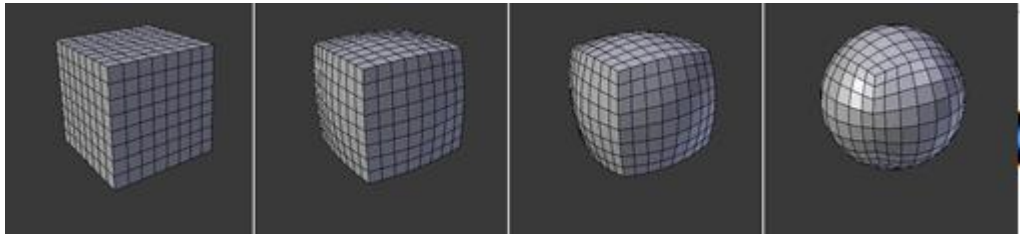

Рисунок 9 – Кубическая сфера

Исследование свойств решения

- Достигается устойчивость с помощью чистых функций градиентного шума
  - Расстояние Хемминга между изображениями карт высот и карт размещения объектов равно нулю.
- Программный интерфейс предоставляет механизмы конфигурации графа
  - Запись (≈ 16 мс) и чтение (≈ 21 мс) конфигурации вершин
  - Запись (≈ 113 мс) и чтение (≈ 161 мс) конфигурации графа
- Программный интерфейс предоставляет механизмы процедурного создания участков ландшафта
  - Создание климатической карты (≈ 65 мс)
  - Создание карты местности (≈ 155 мс, для одного участка)
  - Позиционирование объектов (≈ 184 мс, для одного участка)

Объем памяти, используемый программным интерфейсом:

- Для создания графа: ≈ 26.9 мб
- Для создания ландшафта ≈ 114.5 мб

Время считалось как среднее из 100 итераций отдельного действия.

Память как максимальное значение, используемое компьютером при работе приложения.

OS: Windows 10, **GPU**: Sapphire Radeon RX 5700 XT**, CPU**: AMD Ryzen 5 3600

Результаты работы алгоритма

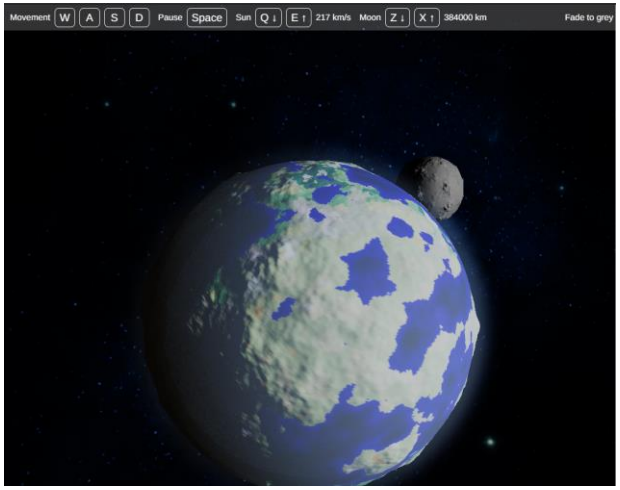

Рисунок 10, А

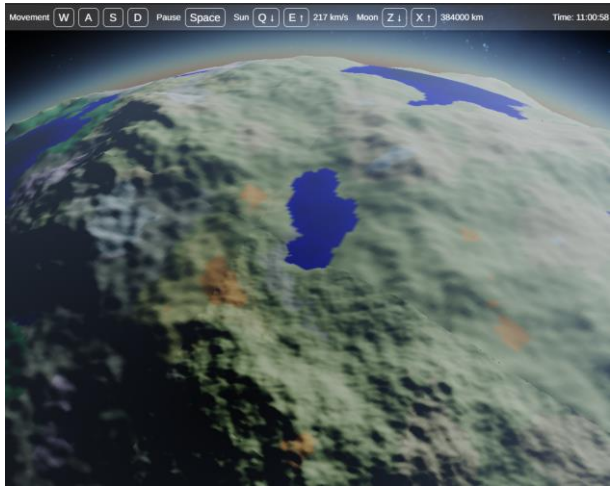

Рисунок 10, Б

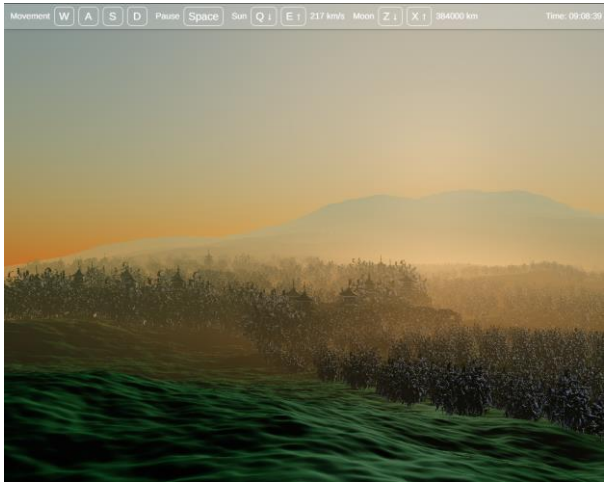

Рисунок 10, В

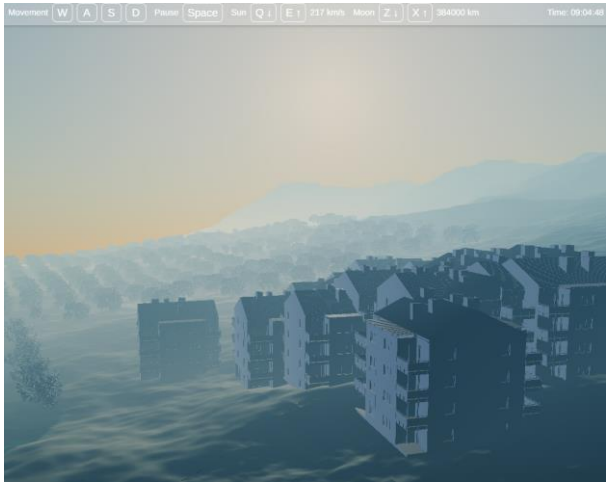

Рисунок 10, Г

Заключение

- Было произведено сравнение аналогов, на основе которого сформировались основные требования к сервису.
- Была выбрана и реализована клиент-серверная архитектура, а также выбраны основные подходы для разработки клиентской и серверной стороны
- Реализован программный интерфейс
  - Был разработан и протестирован алгоритм построения климатической карты на основе связного ациклического графа
  - Был разработан алгоритм генерации ландшафта и позиционирования объектов на нем на основе алгоритма симплексного градиентного шума.
- Реализован пользовательский интерфейс как прототип клиентского приложения для взаимодействия с программным интерфейсом.
  - Были разработаны основные страницы с компонентами формирования входных данных и визуализацией алгоритма.
  - На основе структуры дерева квадрантов был разработан алгоритм детализации ландшафта.
- Были произведены замеры приложения по использованию памяти, а также по времени обработки основных запросов.

Цель была полностью достигнута. Пользовательский интерфейс демонстрирует работу алгоритма автоматизации размещения объектов на ландшафте.

Апробация работы

- Репозиторий проекта

  https://github.com/AntonSkiba/SpaceWorld
- Развернутое приложение в сети Интернет

  https://https://4da7ac80.ngrok.io