

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

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. Реализация архитектуры приложения
3. Разработать пользовательский интерфейс
4. Разработать программный 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	Устойчивость	Импорт объектов	Размер	Доп. ПО	API
Сервис					
World Creator	Да	НП	$2 * 10^8 \times 2 * 10^8 \text{ px}$	Да	Нет
Instant Terra	Нет	НП	$64\,000 \times 64\,000 \text{ px}$	Нет	Нет
World Machine	Да	РК	$12\,000 \times 12\,000 \text{ px}$	Нет	ДО
3D Map Generator	Да	НП	$1\,750 \times 1\,100 \text{ px}$	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

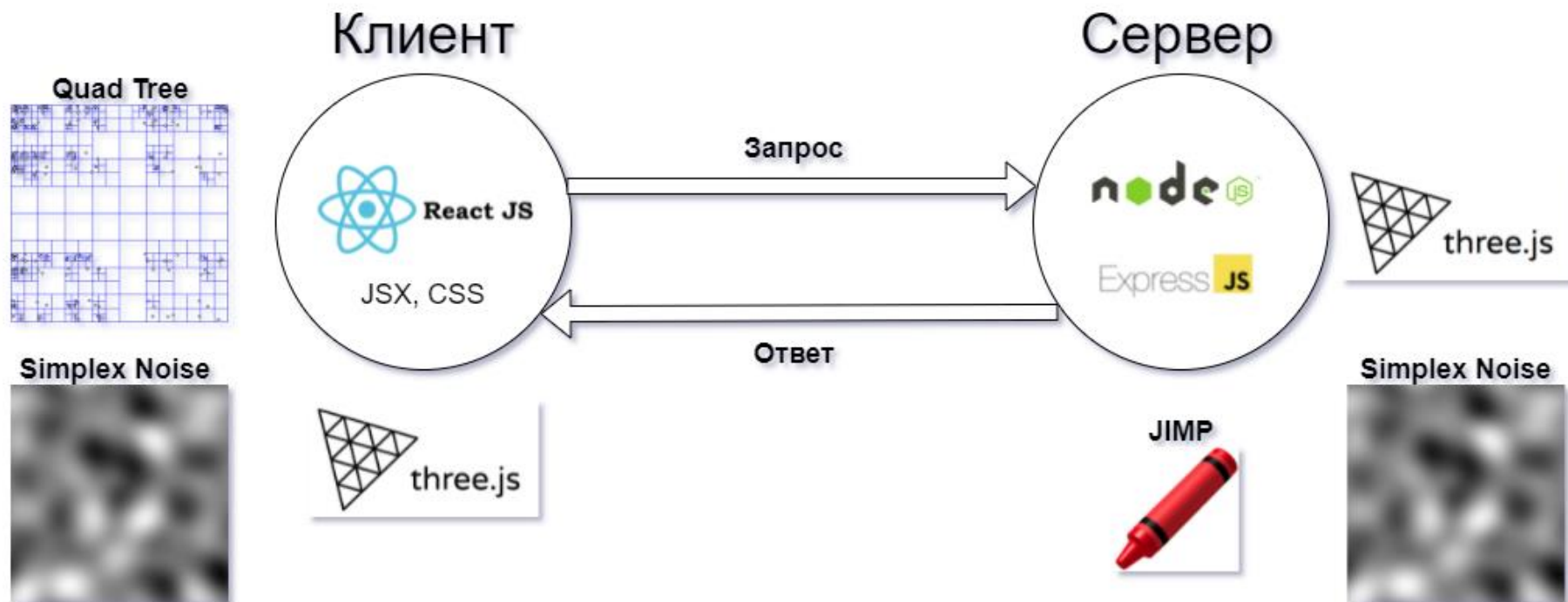


Рисунок 1 – Архитектура приложения

Программный интерфейс (API)

UML–диаграмма классов

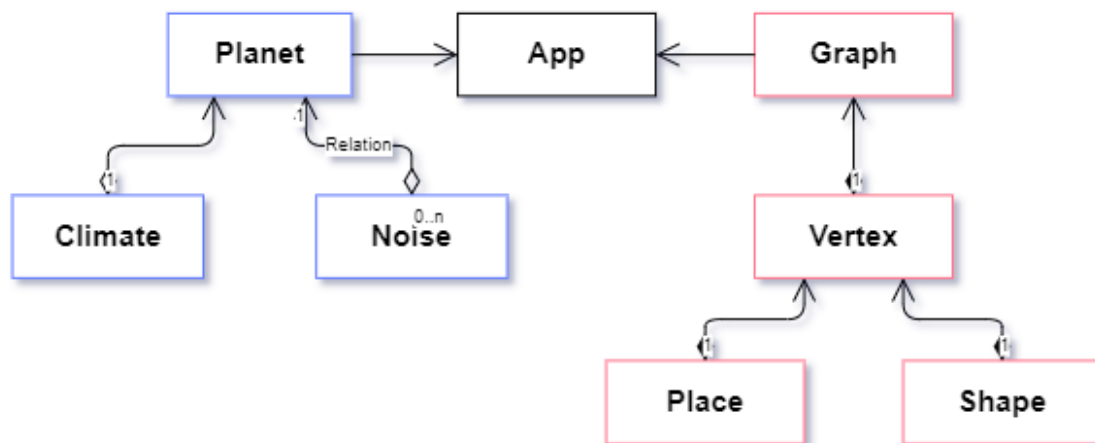


Рисунок 2 – UML-диаграмма серверной стороны

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.

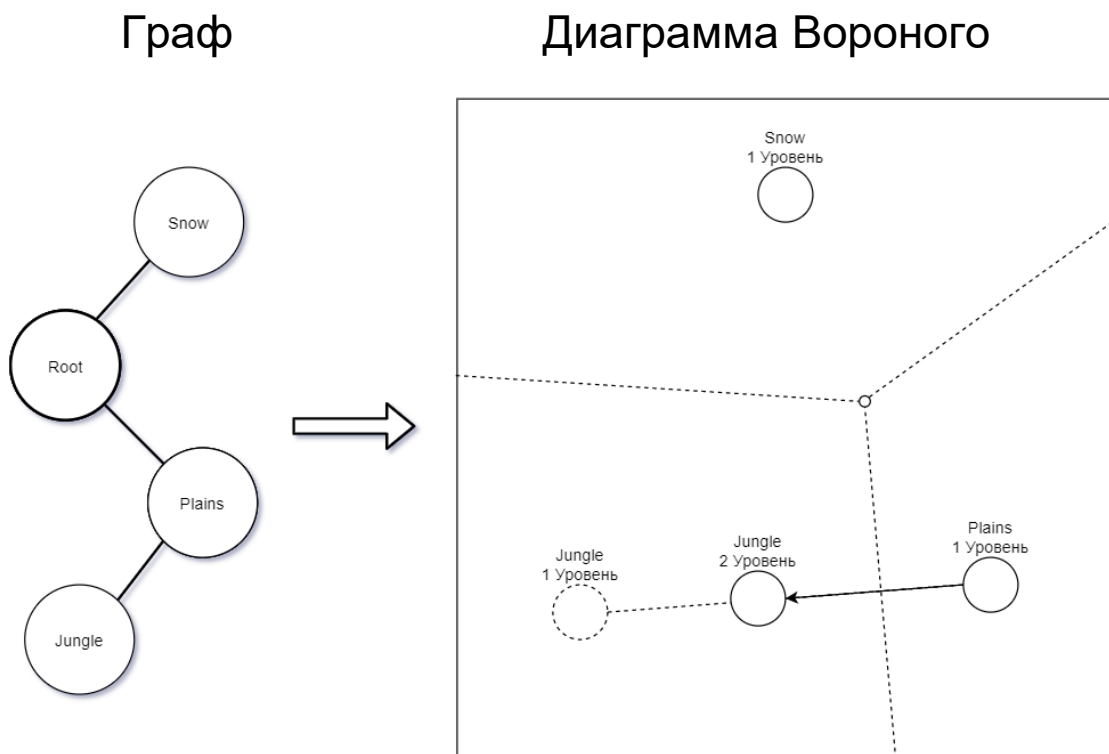


Рисунок 3 – Алгоритм создания климатической карты

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

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

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

Программный интерфейс (API), карта высот

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

Параметры:

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

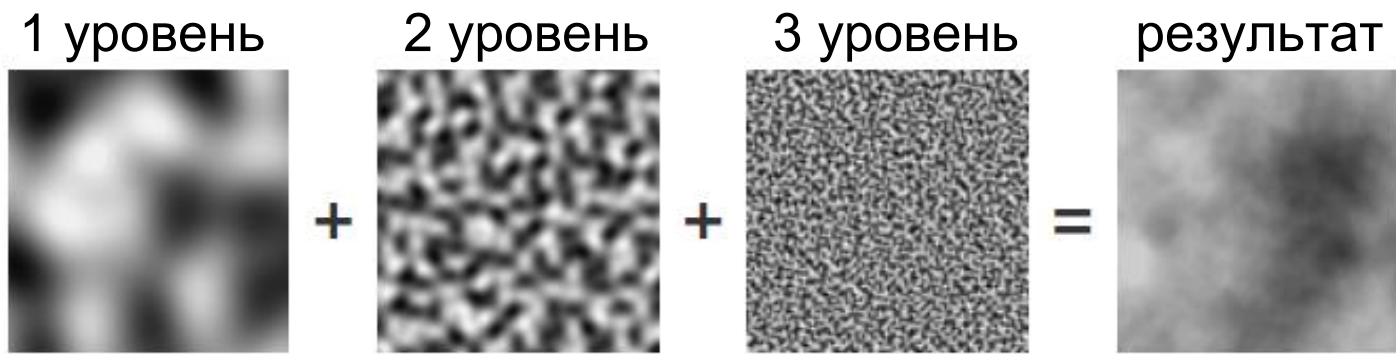


Рисунок 4 – Алгоритм наложения шума с разной частотой

- Ровность (flatness): $V_{x,y,z}^{i,j} = (V_{x,y,z}^{i,j})^f$, 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), размещение объектов

Параметры:

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

Алгоритм:

1. Значение шума: $S_{x,y,z}^{i,j} = \text{Noise}_{x,y,z} \left(\text{seed} = ID, \text{frequency} = \frac{1}{\text{clust}} \right)$, где ID – идентификатор фигуры, clust – значение кластеризации, $\text{Noise}_{x,y,z}$ – значение шума в точке пространства $\{x, y, z\}$ для (i, j) шага участка.
2. Дистанция: $D = \max(0.01, 1 - \text{sat}) * 4^l$, где l – уровень детализации, а sat – значение насыщенности
3. Тестирование значения $\text{check} = \text{test}(S_{x,y,z}^{i,j}, D, F, Z)$, 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 $False$

Клиентская сторона

Делится на две основные страницы:

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

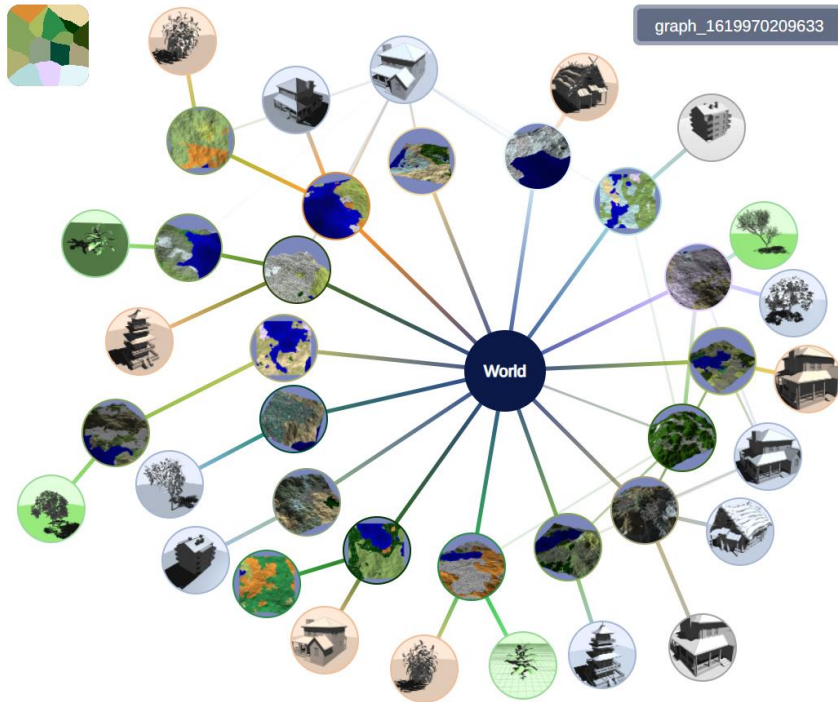


Рисунок 5 – Страница графа

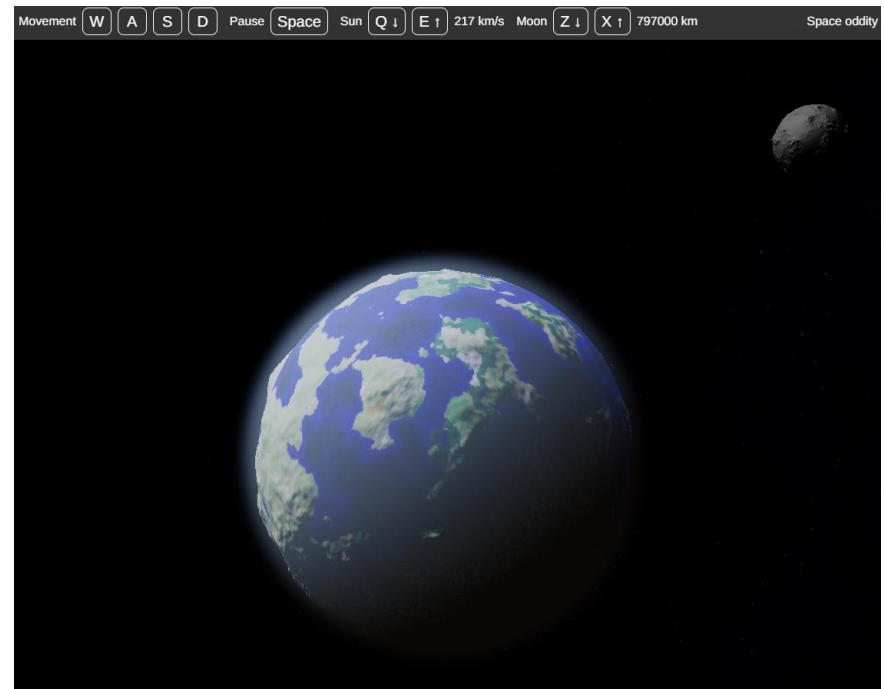


Рисунок 6 – Страница визуализации

Клиентская сторона. Интерфейс.

Конфигурация графа

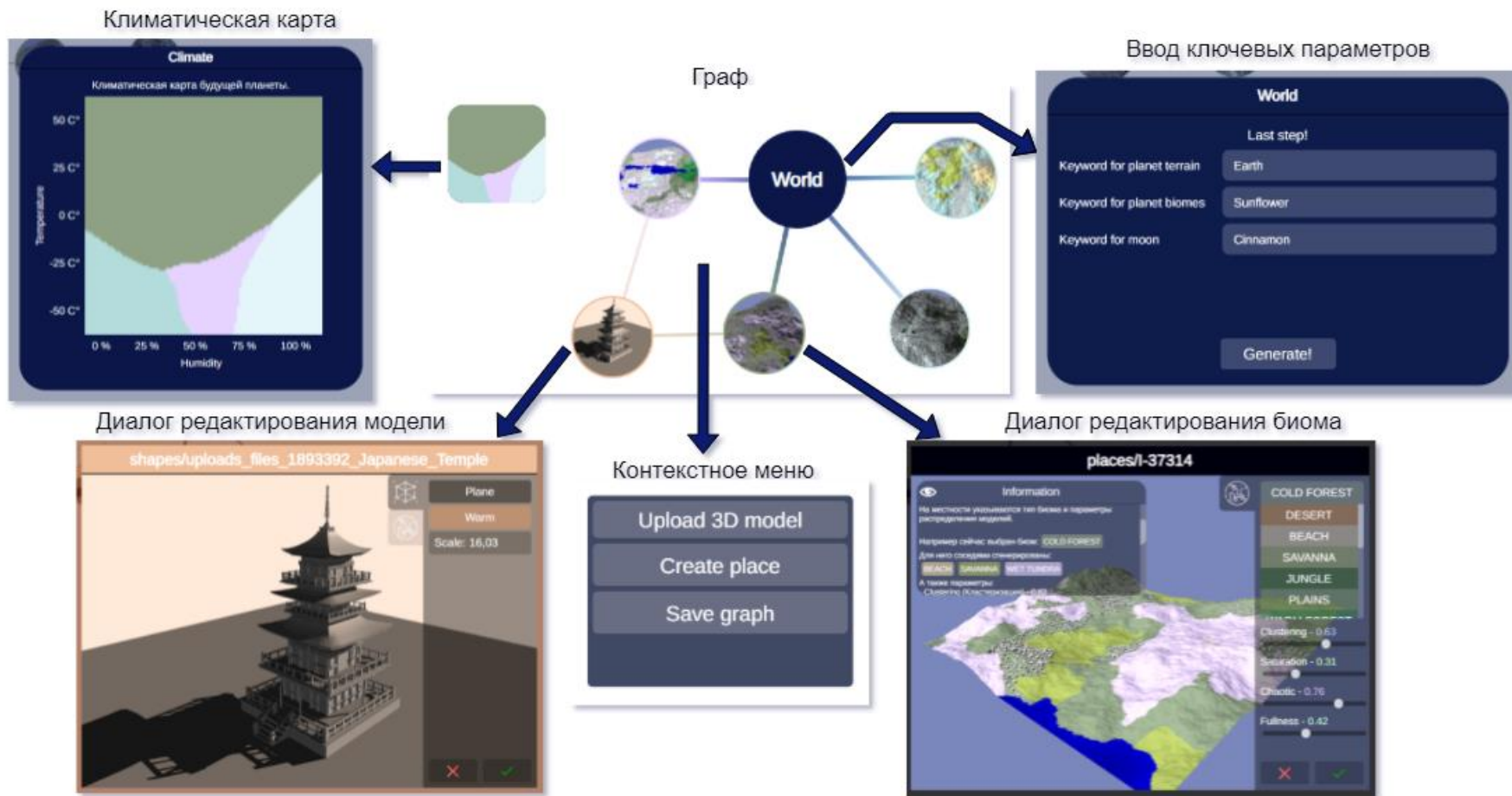


Рисунок 7 – Пользовательский интерфейс

Клиентская сторона, алгоритмы

Визуальное представление ландшафта.

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

Задача вставки элемента: $O(\log n)$

Количество потенциальных участков: 4^l , где l — уровень детализации.



Рисунок 8 – Страница графа

2) Преобразование куба в сферу, рис. 9.

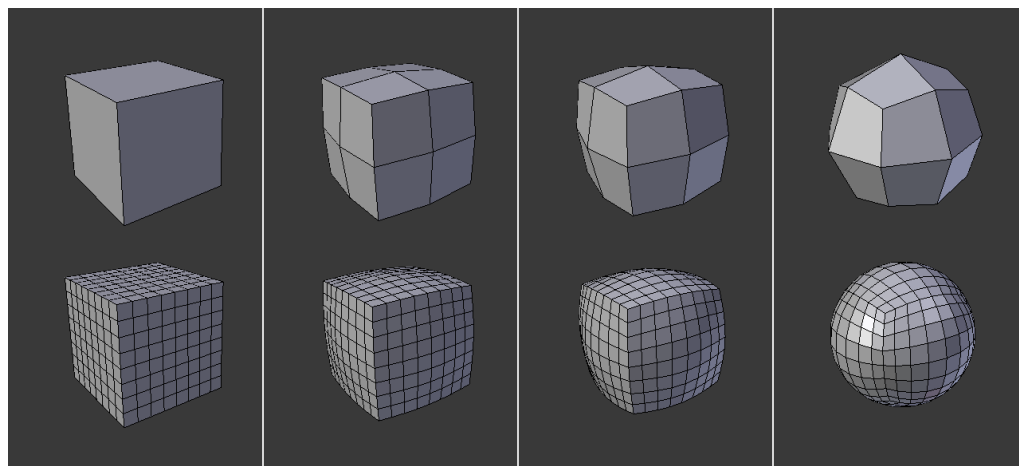


Рисунок 9 – Кубическая сфера

Исследование свойств решения

- Достигается устойчивость с помощью чистых функций градиентного шума
- 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

Результаты работы алгоритма

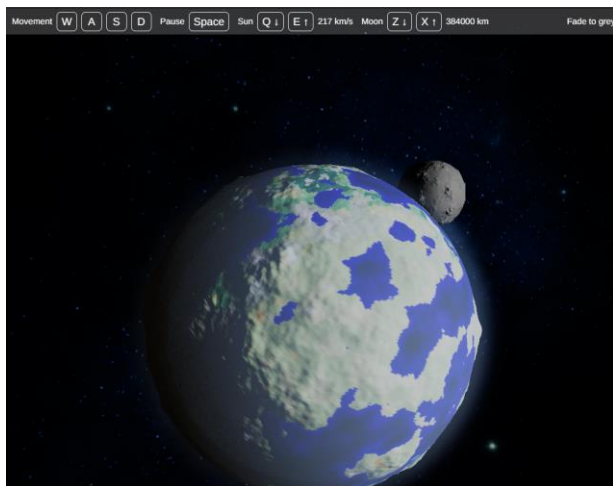


Рисунок 10, А

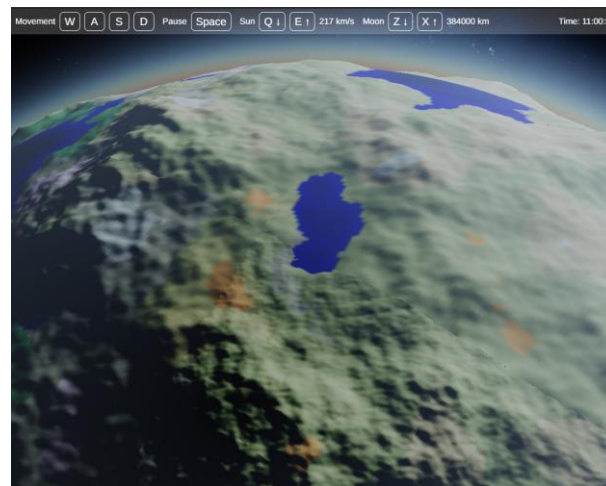


Рисунок 10, Б

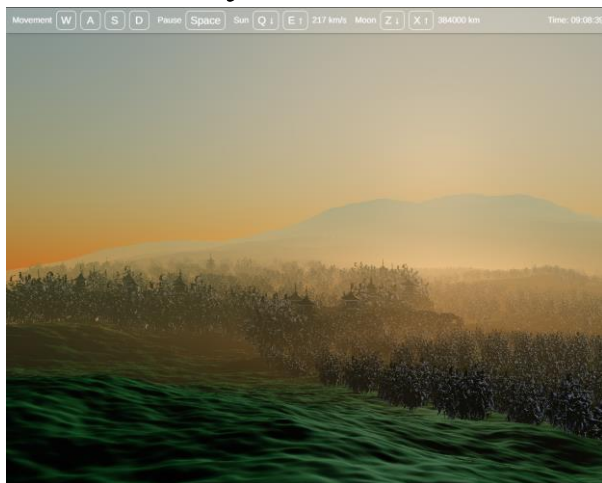


Рисунок 10, В

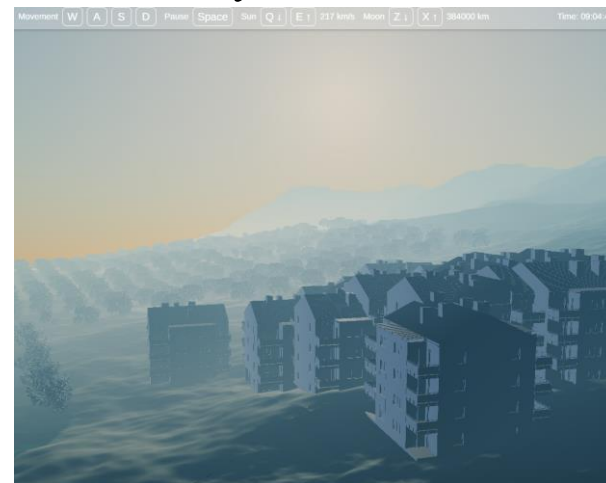


Рисунок 10, Г

Заключение

- Было произведено сравнение аналогов, на основе которого сформировались основные требования к сервису.
- Реализован программный интерфейс за счет использования клиент-серверной архитектуры
- На основе структуры дерева квадрантов был разработан алгоритм детализации ландшафта.
- 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.

Апробация работы

- Репозиторий проекта

<https://github.com/AntonSkiba/SpaceWorld>

- Развернутое приложение в сети Интернет

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