

# 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)
- Конечные размеры

**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

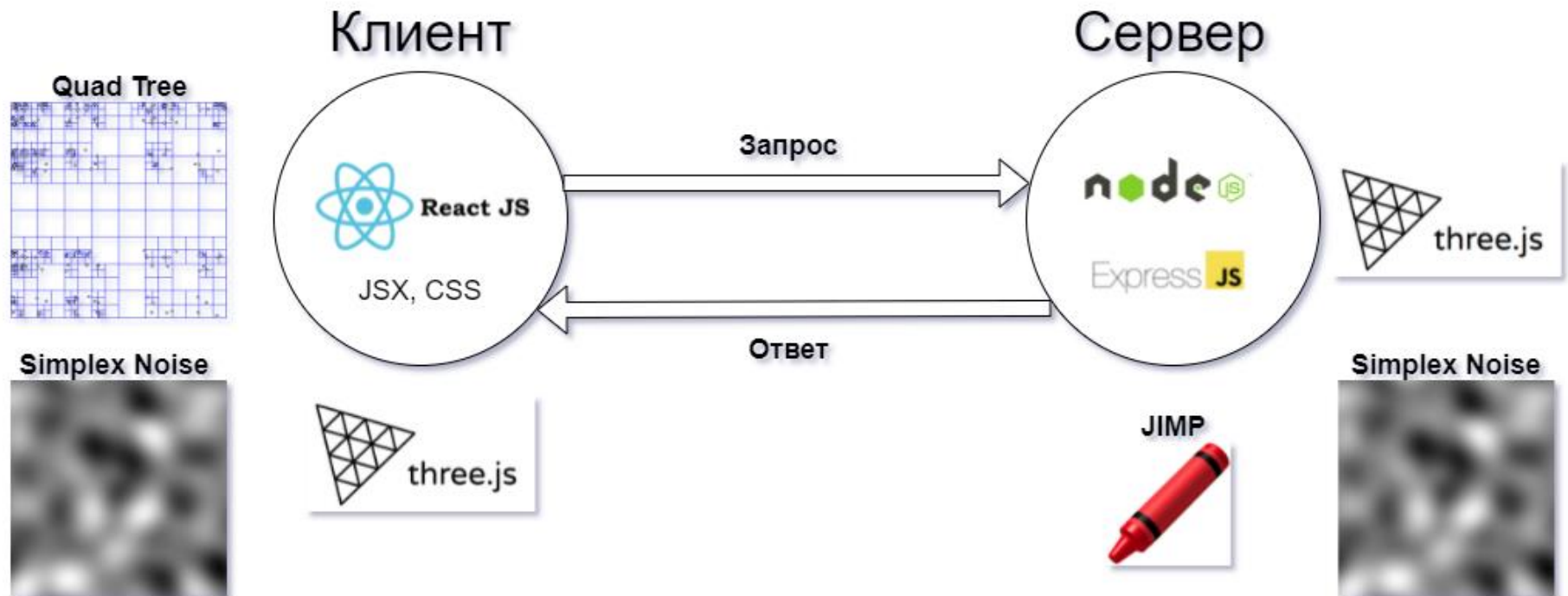
Сервис должен быть устойчивым ко входным данным, предоставлять возможность загрузки объектов, предоставлять API для задания параметров объектов и процедурного создания ландшафта.

# Архитектура приложения

Основа – клиент серверная архитектура

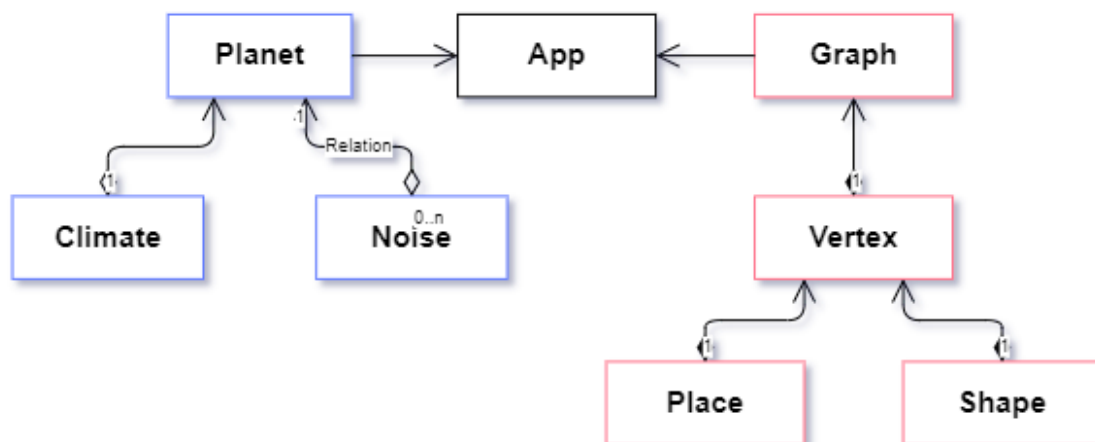
Клиент – компонентный подход

Сервер – объектно-ориентированный подход



# Программный интерфейс (API)

UML – диаграмма классов

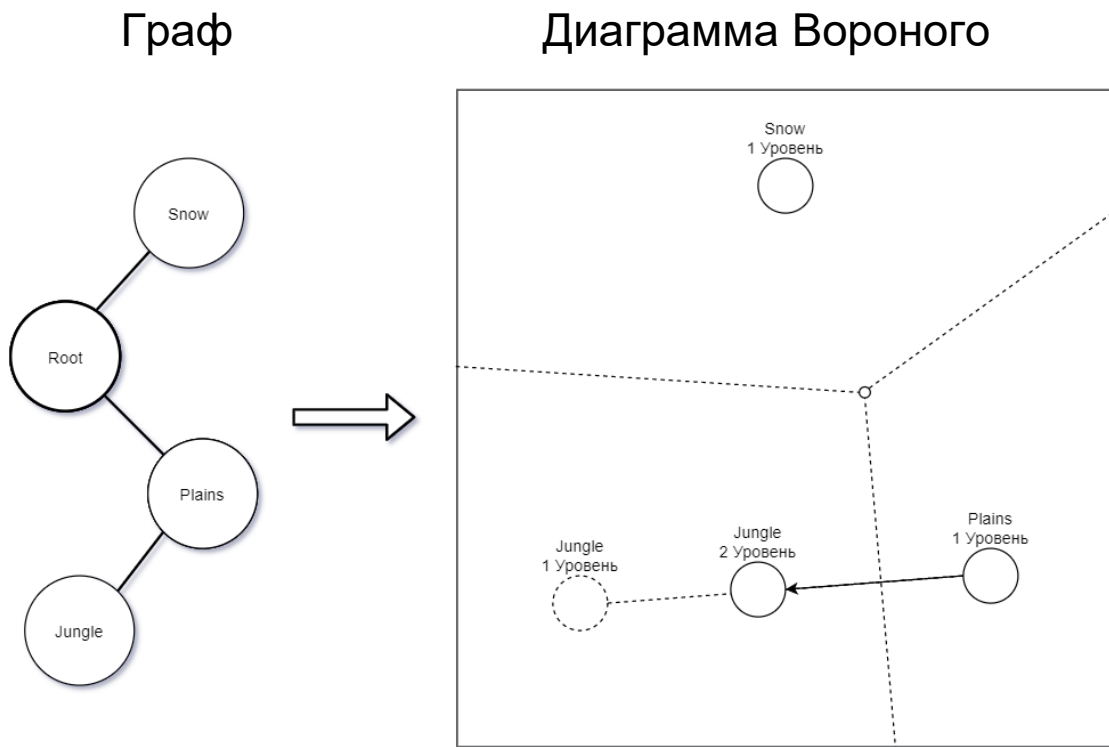


Серверная часть делится на две категории

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.



**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)

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

Параметры:

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



- Ровность (flatness):  $v'_{ij} = v_{ij}^f$ , где  $v_{ij}$  — значение в точке  $[i, j]$ ,  $f$  — значение параметра

# Программный интерфейс (API)

Формирование ландшафта. Размещение объектов.

## Параметры:

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

## Алгоритм:

1. Формируется сущность Noise для объекта на основе id и кластеризации:  
 $frequency = 1 / clustering$
2. В зависимости от насыщенности формируется расстояние между объектами:  
 $d = \max(0.01, 1 - saturation) * 4^l$ , где  $d$  — расстояние,  $l$  — уровень детализации
3. Тестирование значения
  1. Проверяется дистанция
  2. Значение шума проверяется на прохождение порога заполнения
  3. Проверяется биом объекта на совпадение с биомом местности в рассматриваемой точке
4. При прохождении всех проверок, ключ объекта добавляется в массив для рендеринга на клиентской стороне.



# Клиентская сторона.

Делится на две основные страницы:

- Конфигурация графа, рисунок 1
- Визуальное представление ландшафта, рисунок 2

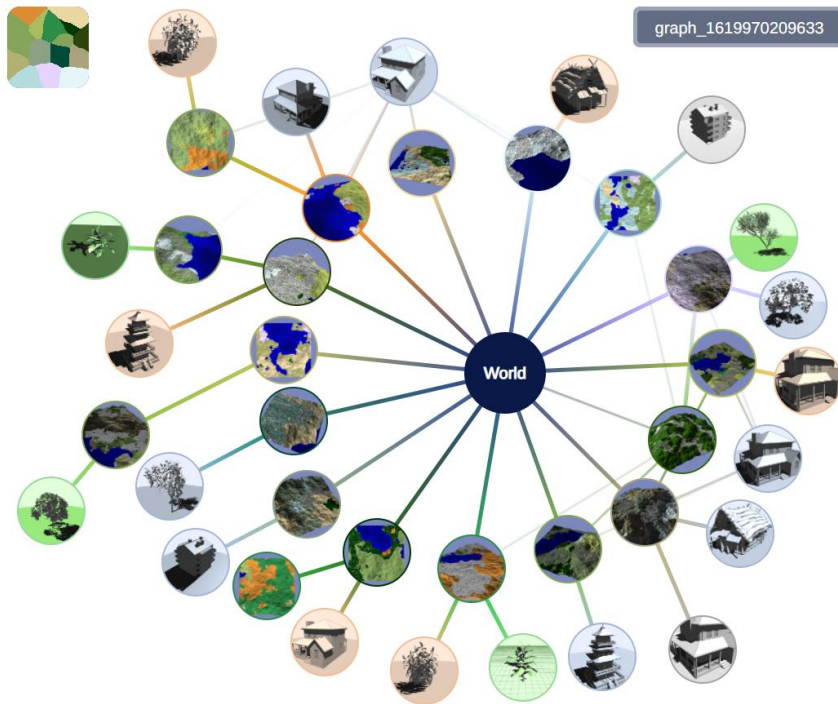


Рисунок 1 – Страница графа

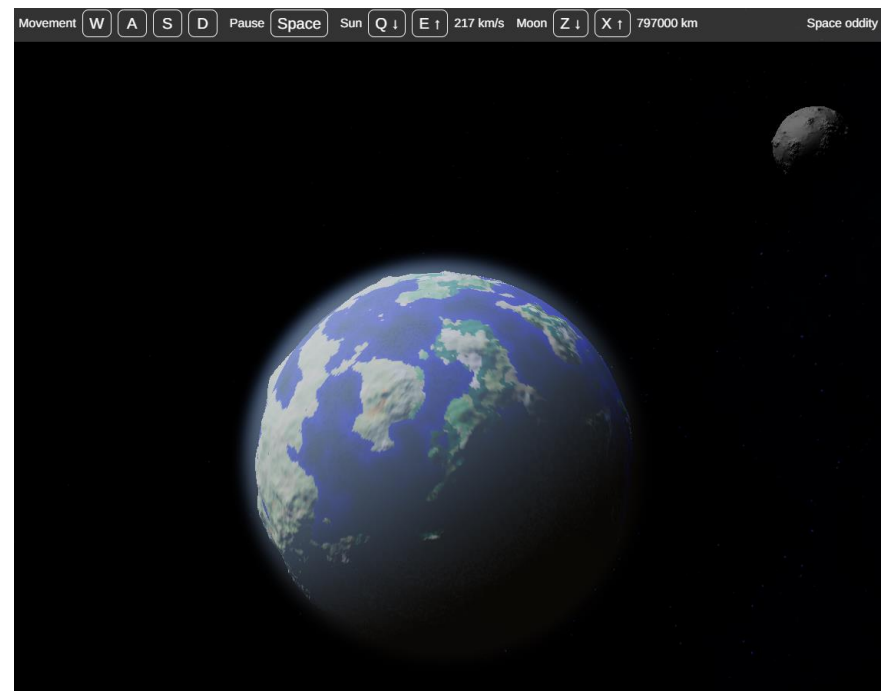
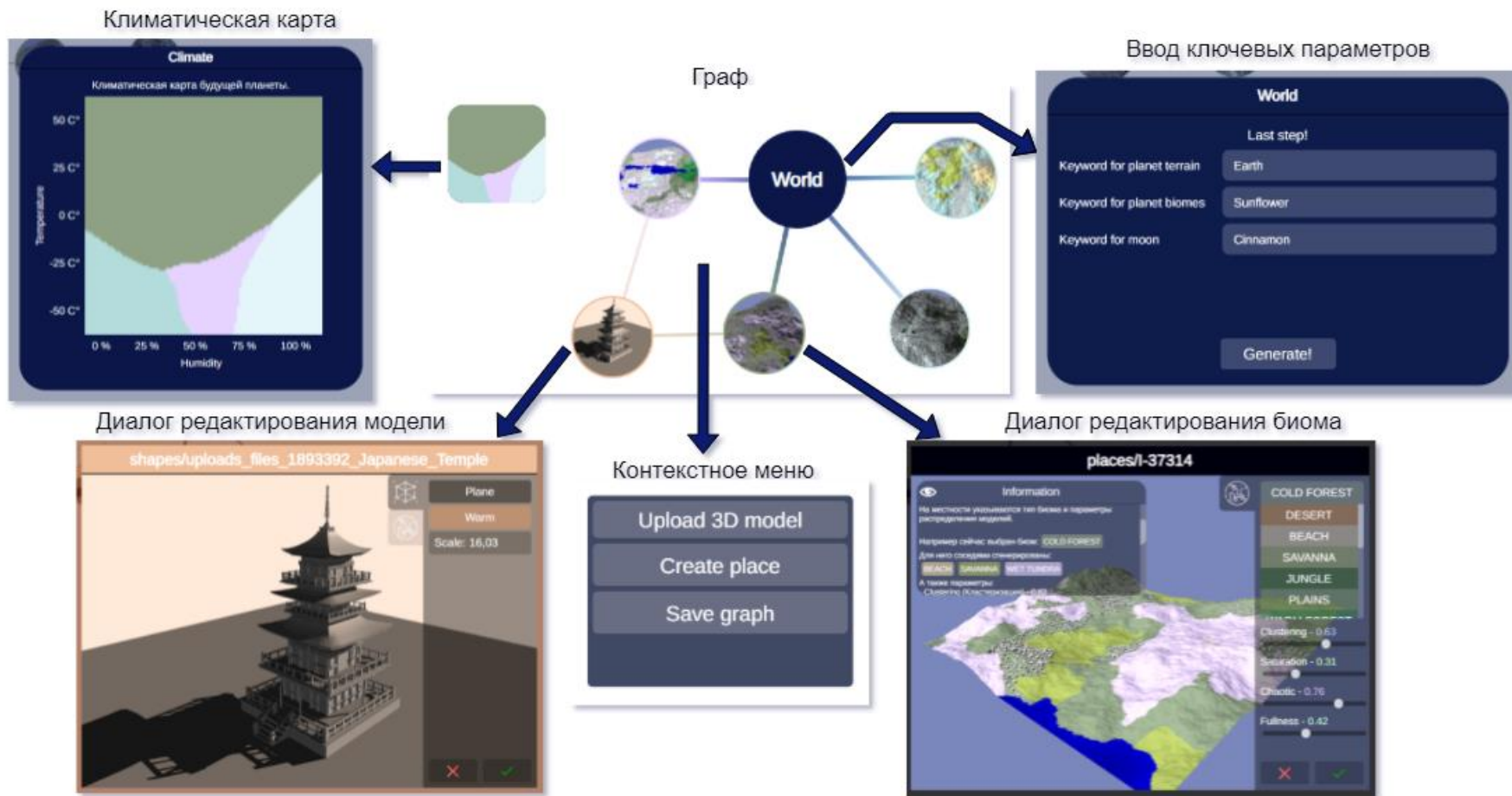


Рисунок 2 – Страница визуализации

# Клиентская сторона. Интерфейс.

## Конфигурация графа



# Клиентская сторона. Алгоритмы.

Визуальное представление ландшафта.

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

Задача вставки элемента:  $O(\log n)$

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

● Положение камеры пользователя  
— Граница основного участка  
— Шаг детализации

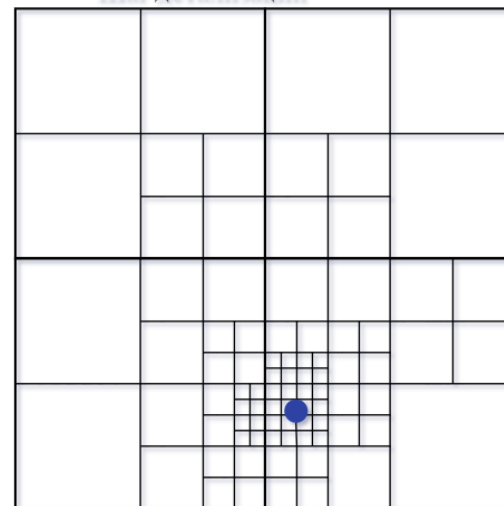


Рисунок 1 – Страница графа

2) Преобразование куба в сферу, рисунок 2.

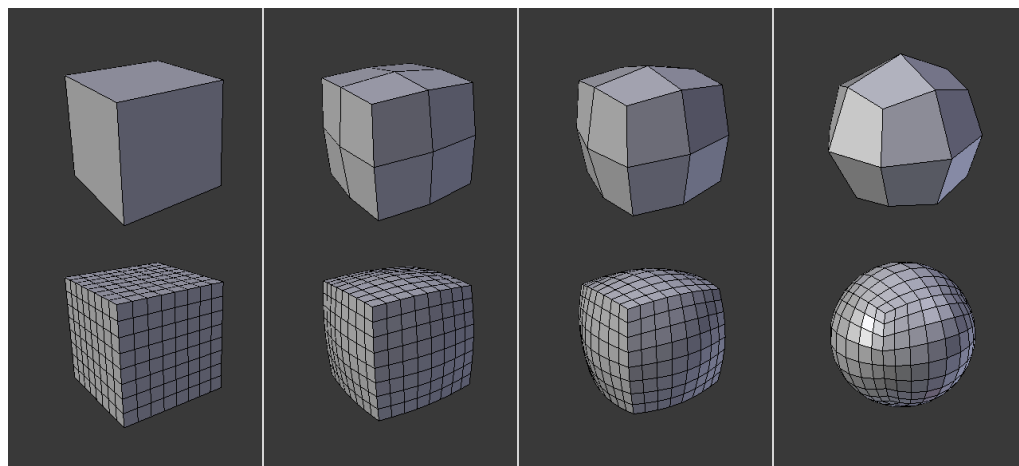
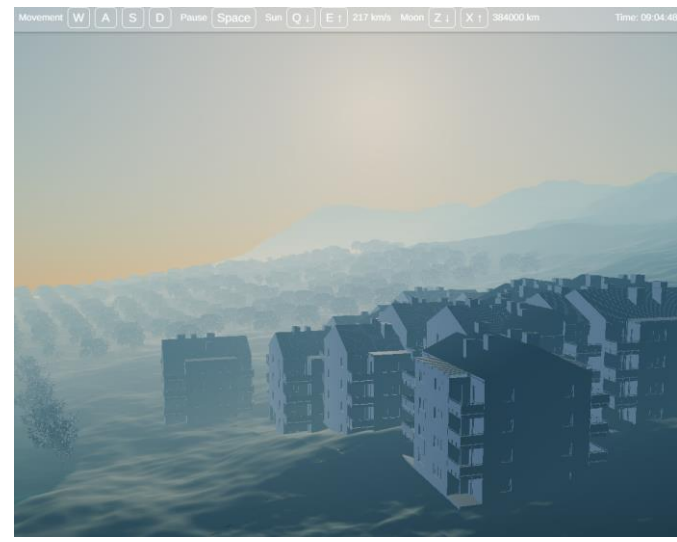
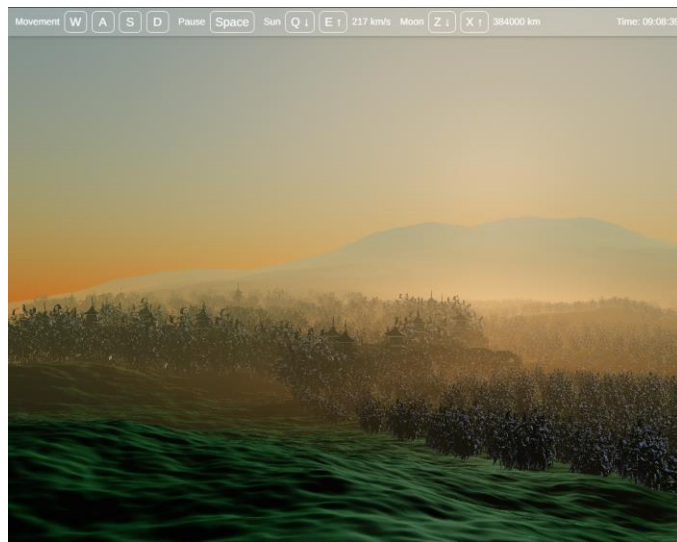
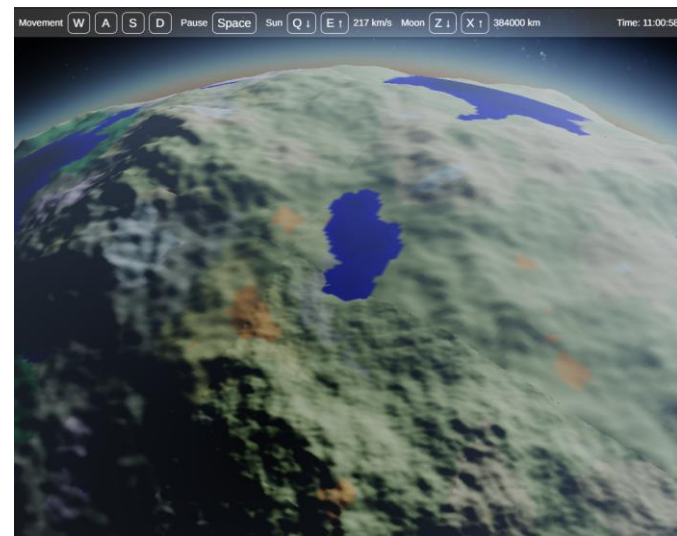
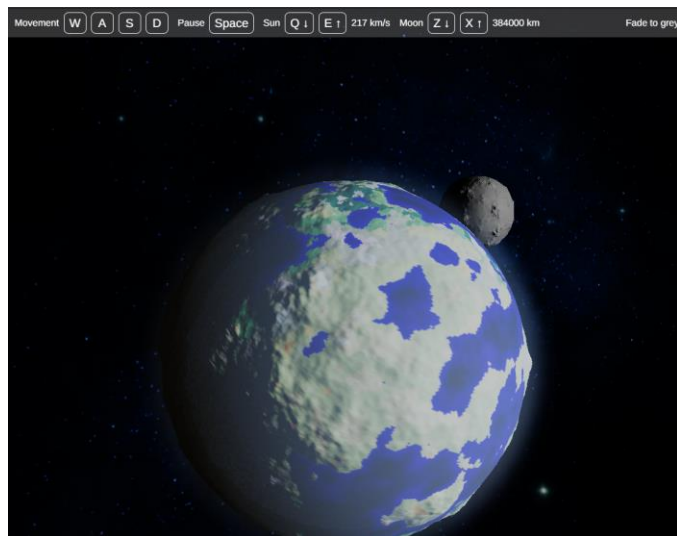


Рисунок 2 – Кубическая сфера

# Исследование свойств решения

- Достигается устойчивость с помощью методов градиентного шума
- Программный интерфейс предоставляет механизмы конфигурации графа
  - Запись и чтение конфигурации вершин
  - Запись и чтение конфигурации графа
- Программный интерфейс предоставляет механизмы процедурного создания участков ландшафта
  - Создание климатической карты
  - Создание карты местности
  - Позиционирование объектов

# Результаты работы алгоритма



# 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>