



**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)**

Генерация ландшафта

**Студент: Баранов Николай Алексеевич ИУ7-51Б
Руководитель: Кострицкий А. С.**

2024

Цель и задачи

Цель курсовой работы – разработать программу для генерации ландшафта и его визуализации. В качестве параметров генерации принять значения высот в точках, размер ландшафта и шаг задания точек.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) формализовать задачу;
- 2) проанализировать и выбрать алгоритмы для создания изображения и генерации ландшафта;
- 3) спроектировать программу для генерации ландшафта и его визуализации;
- 4) выбрать средства реализации и реализовать спроектированную программу;
- 5) исследовать зависимость скорости генерации ландшафта от параметров генерации.

Объекты сцены

Сцена состоит из следующих объектов:

- 1) источник света, расположенный на бесконечности — задаётся 2 углами;
- 2) камера — задаётся положением в пространстве, фокусным расстоянием, дальностью видимости, а также 3 векторами, задающими системы координат камеры;
- 3) ландшафт — задаётся регулярной картой высот, так как параметрами генерации являются значения высот в точках и шаг задания точек, и множеством полигонов.

Алгоритмы генерации ландшафта

Сравнительная таблица алгоритмов генерации ландшафта

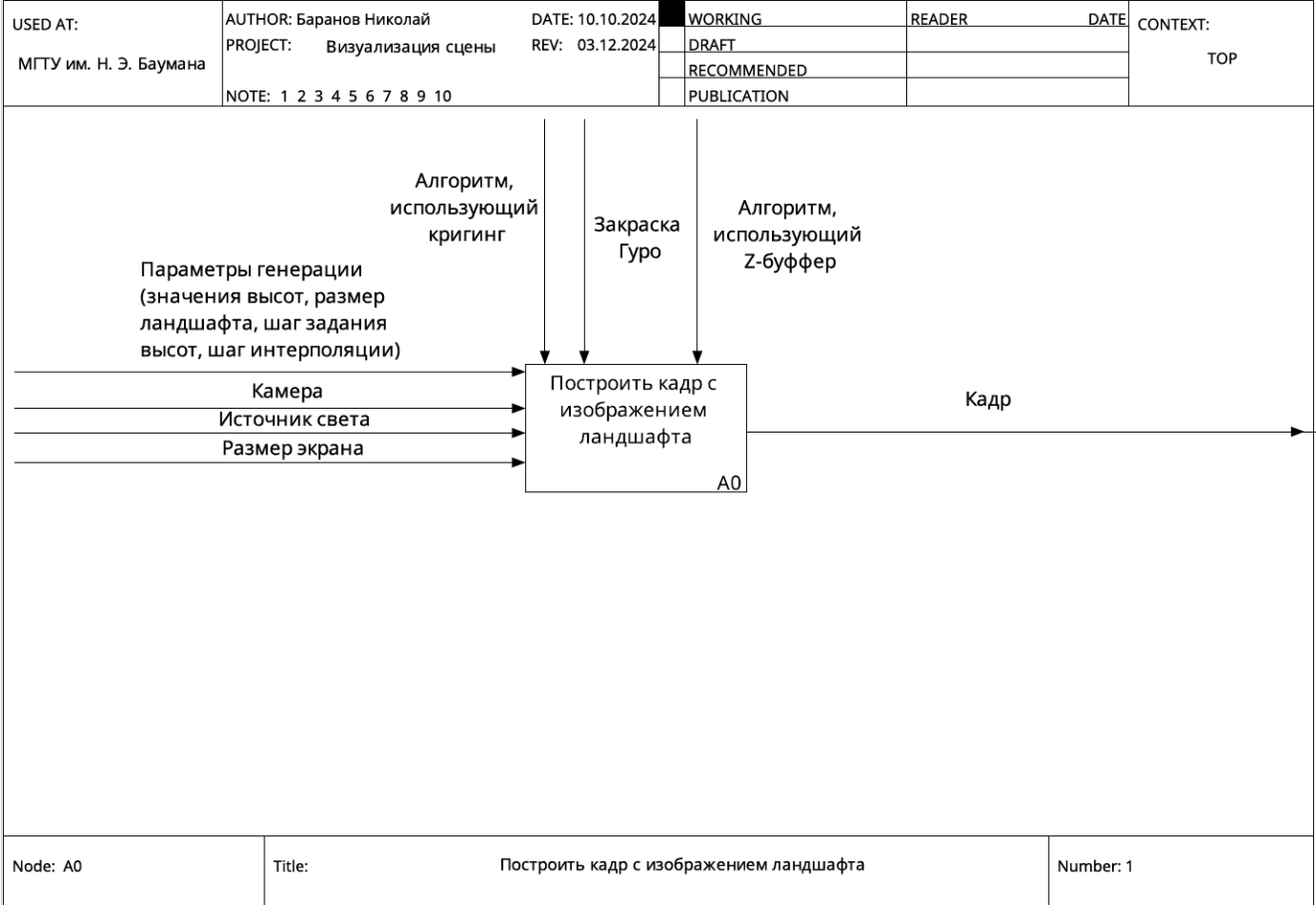
Критерий сравнения	Алгоритм diamond-square	Алгоритм, использующий билинейную интерполяцию	Алгоритм, использующий кригинг
Нужен генератор случайных чисел	Да	Нет	Нет
Проблемы с генерацией значений на границах	Да	Нет	Нет
Значение зависит от всех точек	Нет	Нет	Да

Алгоритмы удаления невидимых линий и поверхностей

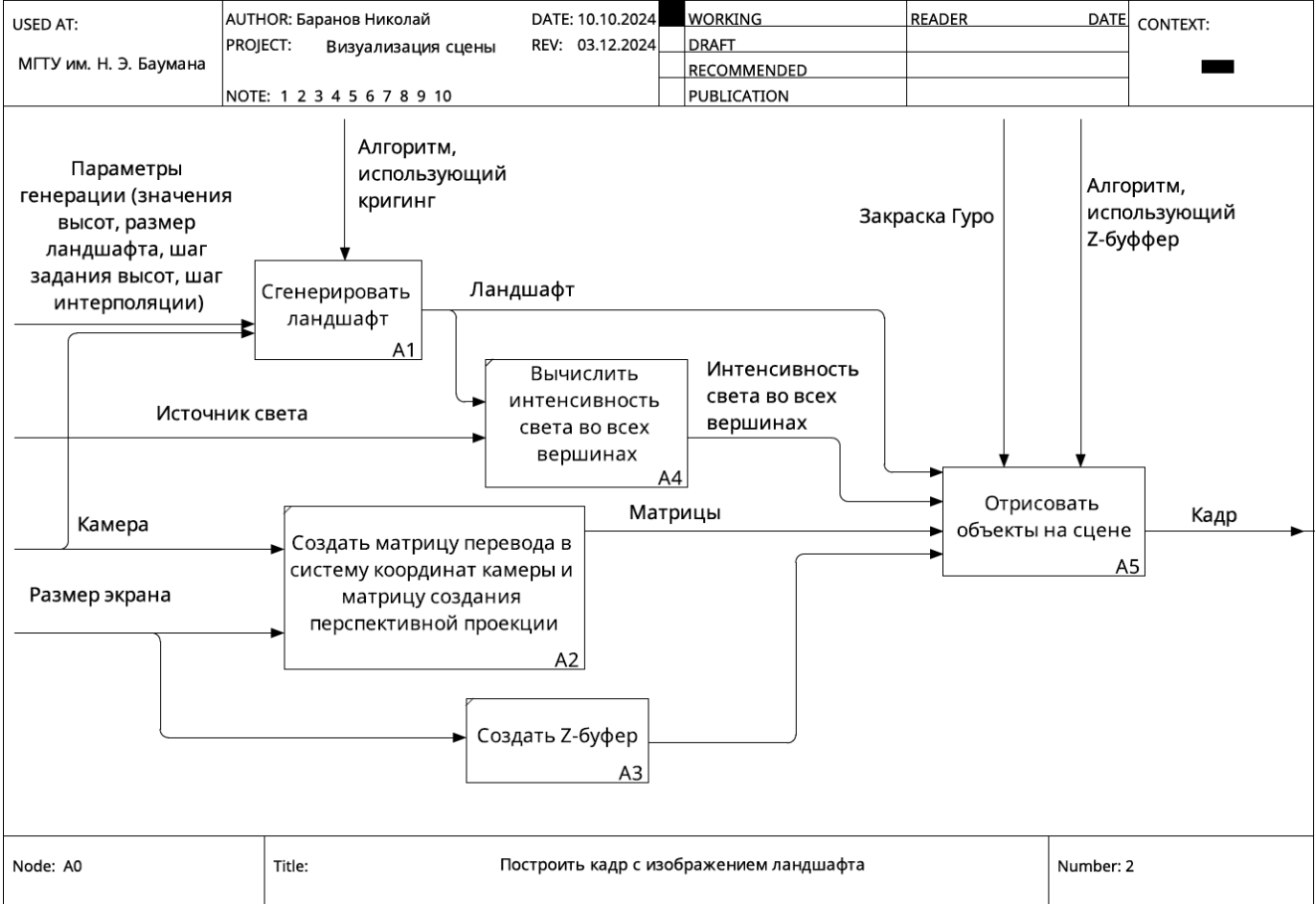
Сравнительная таблица алгоритмов удаления невидимых линий и поверхностей

Критерий сравнения	Алгоритм Робертса	Алгоритм плавающего горизонта	Алгоритм Варнока	Алгоритм Вейлера-Азертон	Алгоритм, использующий z-буфер	Алгоритм художника	Алгоритм обратной трассировки лучей
Пространство работы	Объектное	Изображения	Изображения	Объектное	Изображения	Изображения	Изображения
Сложность	$O(N^2)$	$O(CN)$	$O(CN)$	$O(N^2)$	$O(CN)$	$O(CN)$	$O(CN)$
Возможность вычислить интенсивность света	+	-	+	+	+	+	+
Обрабатываемые объекты	Выпуклые полигональные	Заданные аналитически	Любые	Полигональные	Любые	Полигональные	Любые

Формализация задачи с учётом выбранных алгоритмов



Формализация процесса создания кадра с изображением ландшафта



Алгоритм, использующий кригинг

Значение высоты в точке: $z_k(x_0) = \sum_{i=1}^n \lambda_i * z(x_i)$

Ковариация: $\text{Cov}(x_i, x_j) = C_{ij} = \text{Cov}(h) = 1 - e^{-\frac{h}{a}}$

Система уравнений:

$$\begin{cases} \lambda_1 * C_{11} + \lambda_2 * C_{12} + \dots + \lambda_n * C_{1n} - C_{10} = 0 \\ \lambda_1 * C_{21} + \lambda_2 * C_{22} + \dots + \lambda_n * C_{2n} - C_{20} = 0 \\ \dots \\ \lambda_1 * C_{n1} + \lambda_2 * C_{n2} + \dots + \lambda_n * C_{nn} - C_{n0} = 0 \end{cases}$$

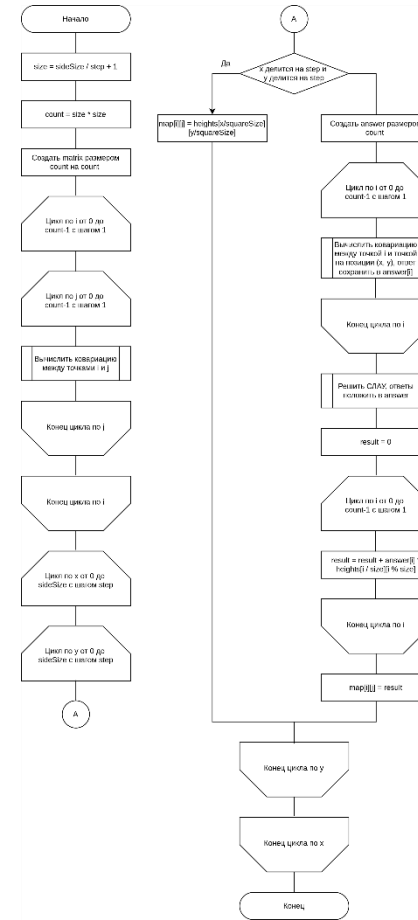
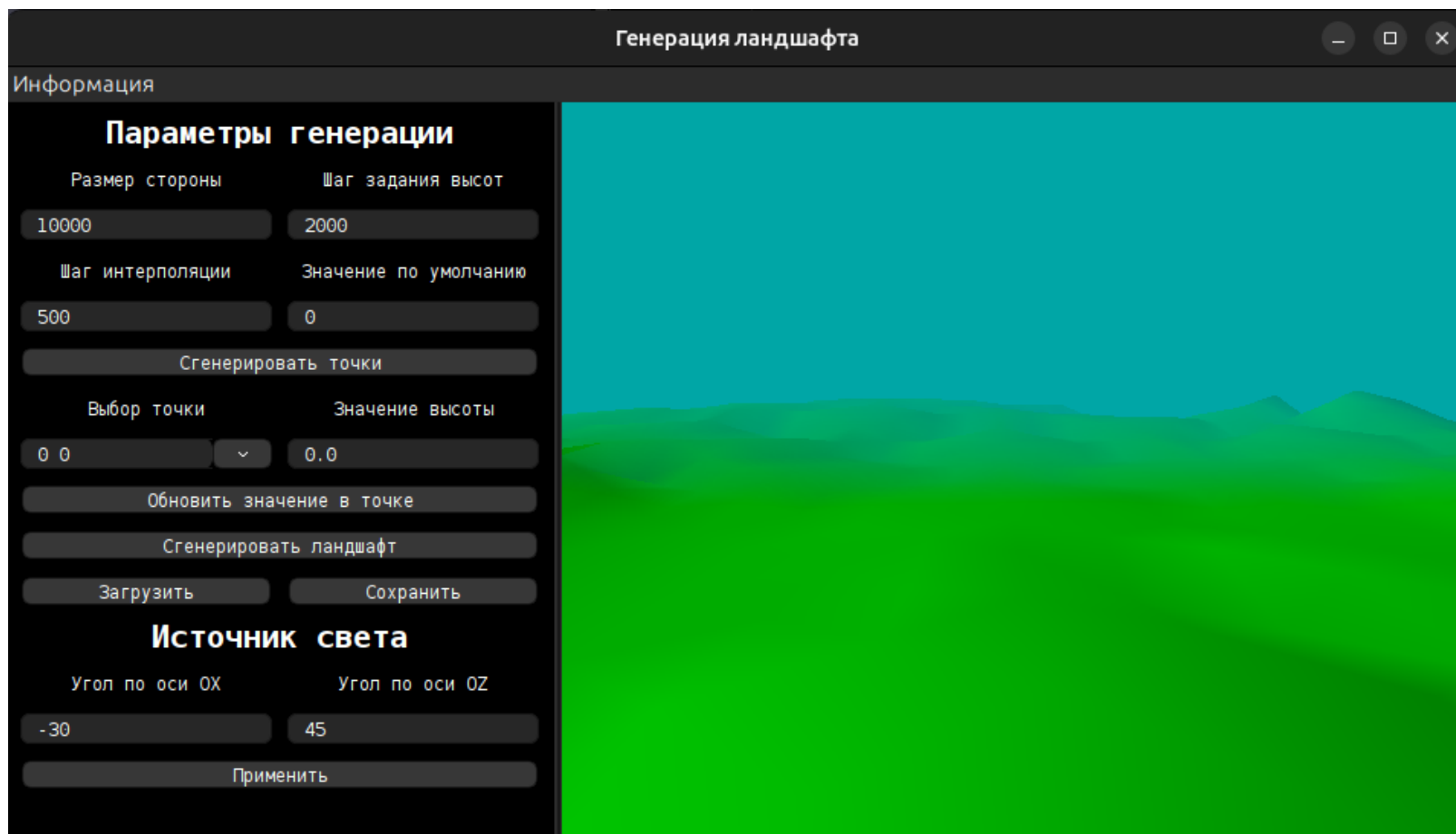


Схема алгоритма, использующего кригинг

Выбранные средства реализации

- Язык: Java;
- Модульное тестирование: JUnit5.

Интерфейс программы



Модульное тестирование

В качестве меры, используемой при тестировании программы, было выбрано покрытие кода. Процент покрытия кода составил 18%.

Пример теста

```
@Test
void negativeHeights2() {
    try {
        heights.getLast().removeLast();
        new Generator(landscape, heights, sideSize, squareSize, step);
        fail();
    } catch (IllegalArgumentException ignored) {
    }
}
```

Пример вывода

```
> Task :test
com.github.N1ckBaran0v.program.math.GaussTest
Test solve() PASSED
Test negative() PASSED
com.github.N1ckBaran0v.program.scene.GeneratorTest
Test negativeSquareSize() PASSED
Test run() PASSED
Test negativeHeights1() PASSED
Test negativeHeights2() PASSED
Test negativeHeights3() PASSED
Test negativeStep() PASSED
Test negativeSideSize() PASSED
SUCCESS: Executed 9 tests in 1s
```

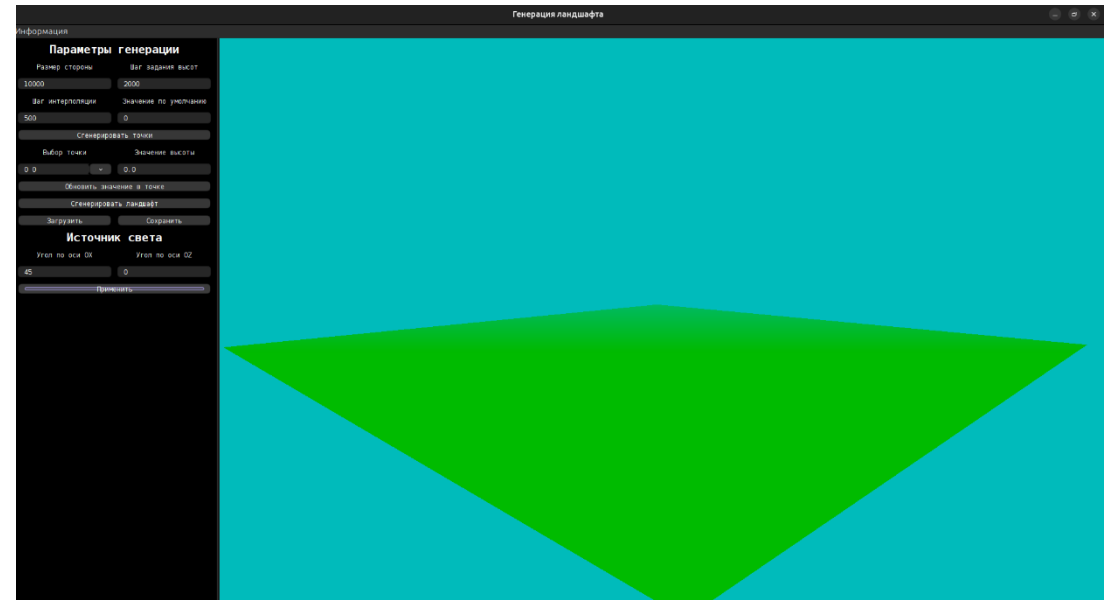
Функциональное тестирование

Этапы функционального тестирования:

- 1) определение классов эквивалентности функционального тестирования;
- 2) составление входных данных для каждого класса эквивалентности;
- 3) генерация ландшафта и получение изображения для каждого тестового случая;
- 4) визуальная оценка результата.

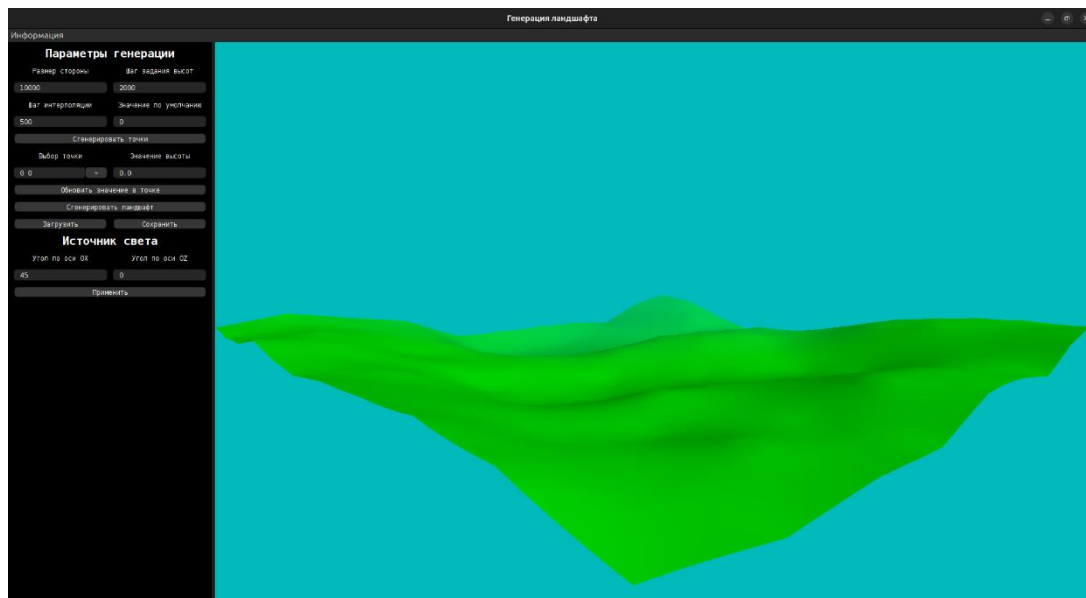
Классы эквивалентности функционального тестирования:

- 1) все значения высот равны 0;
- 2) значения высот сгенерированы с использованием шума Перлина;
- 3) «квадратный» холм;
- 4) «круглый» холм;
- 5) на главных диагоналях значения высот равны 1000, в остальных вершинах – 0.

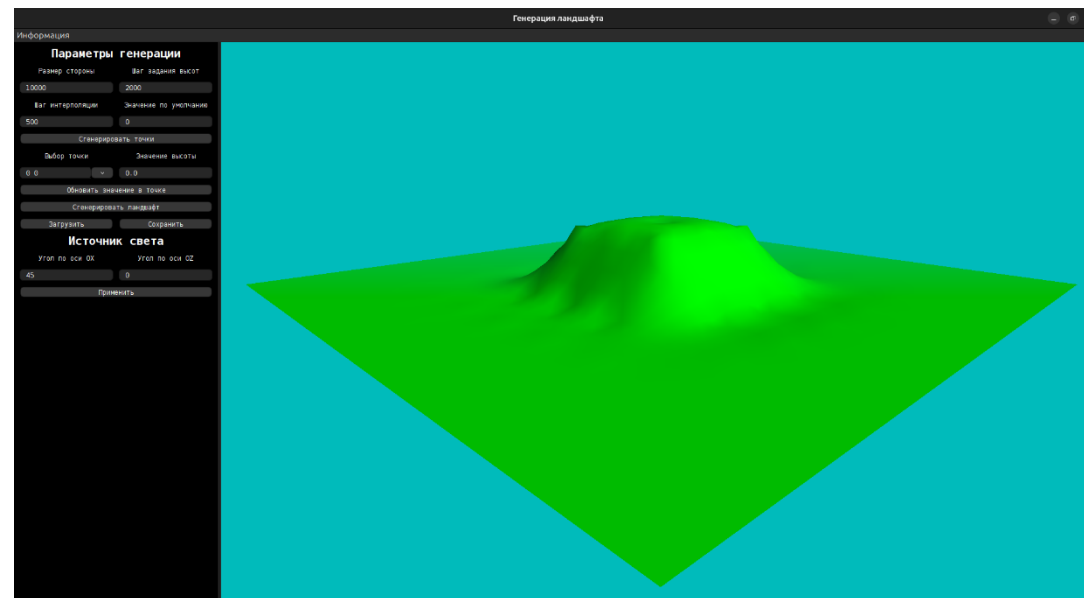


Пример работы программы для теста 1 – плоский ландшафт

Функциональное тестирование

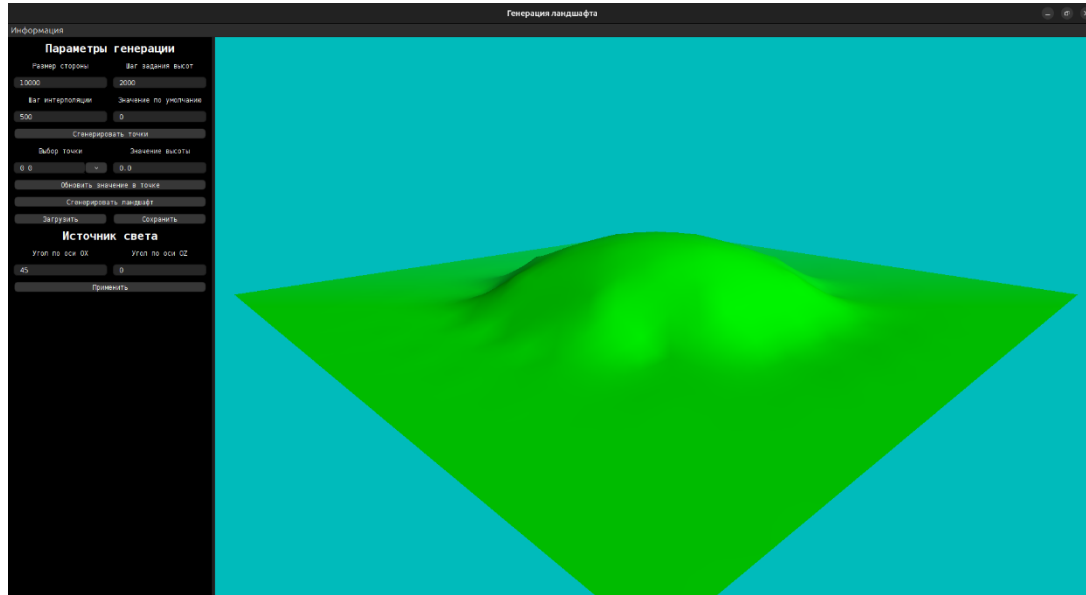


Пример работы программы для теста 2 — значения сгенерированы с помощью шума Перлина

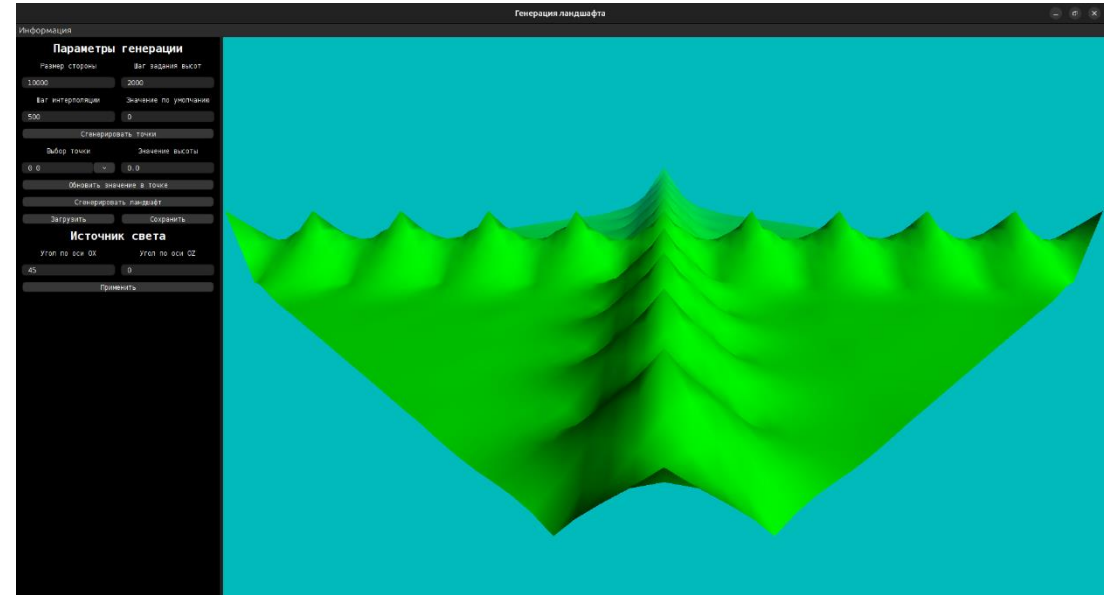


Пример работы программы для теста 3 — «квадратный» холм

Функциональное тестирование

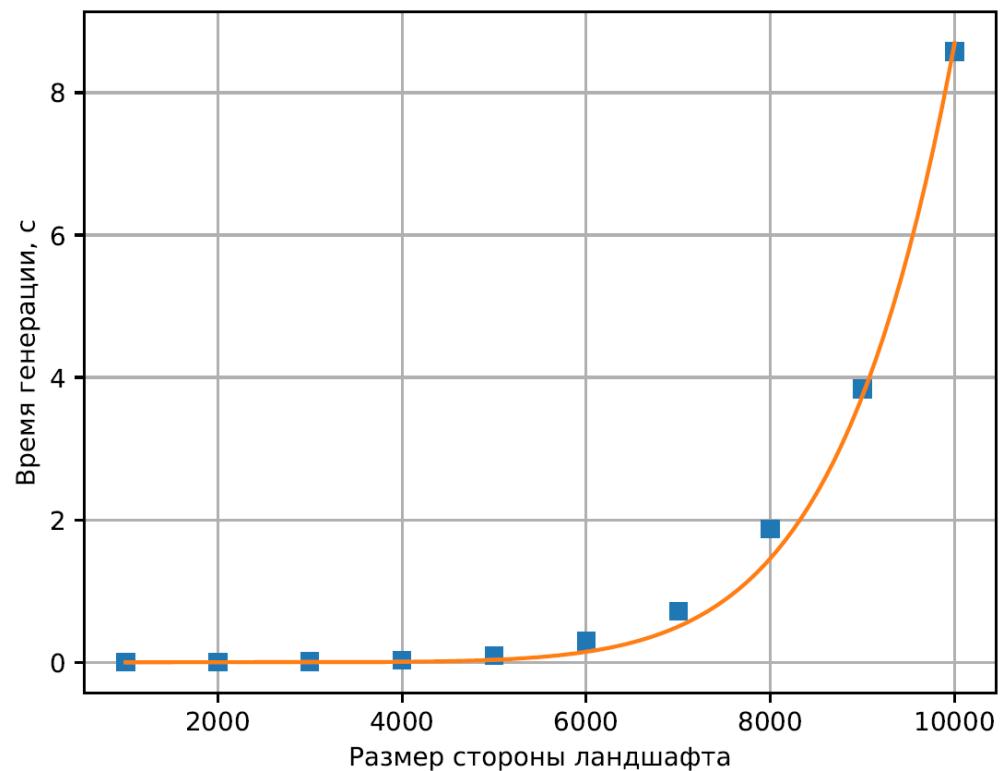


Пример работы программы для
теста 4 – «круглый» холм

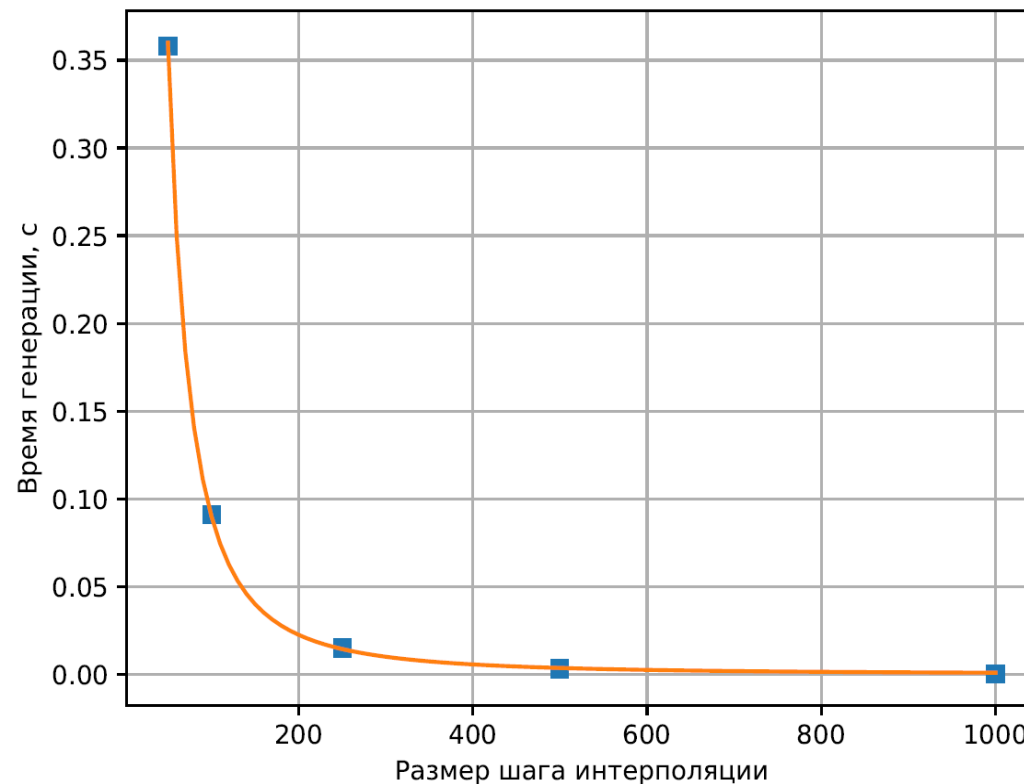


Пример работы программы для теста 5 – 1000 на
главных диагоналях, 0 в остальных вершинах

Зависимость скорости генерации ландшафта от параметров генерации

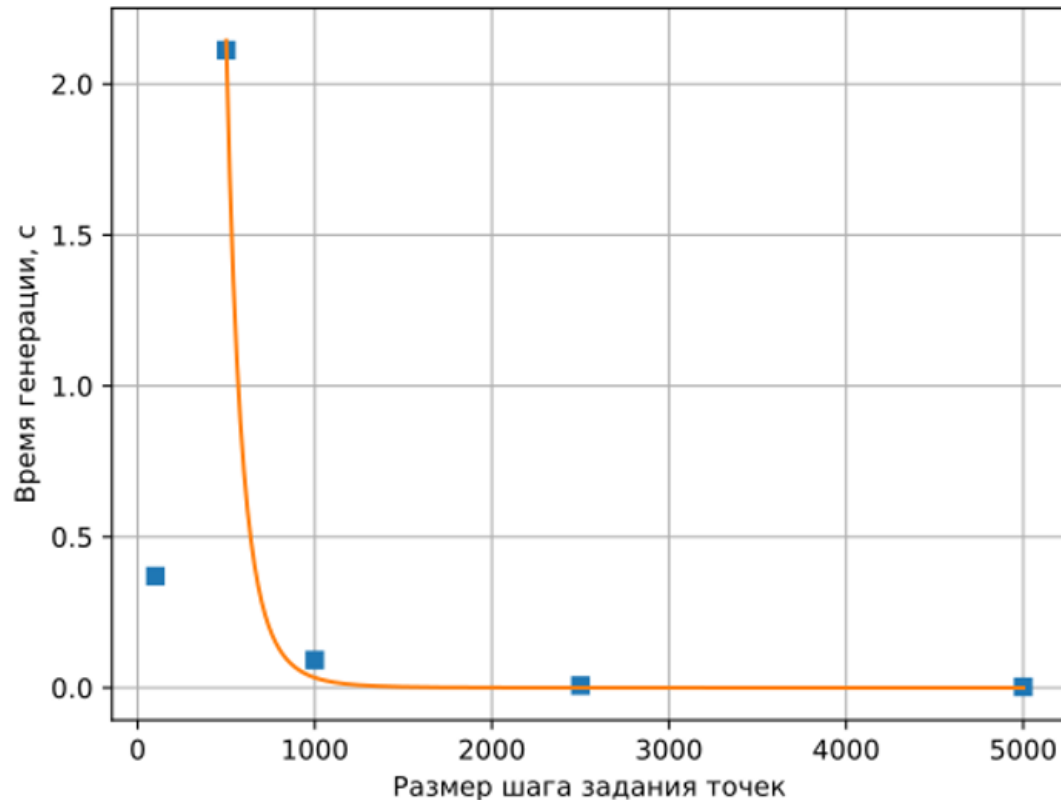


Зависимость времени генерации ландшафта от размера стороны ландшафта



Зависимость времени генерации ландшафта от размера шага интерполяции

Зависимость скорости генерации ландшафта от параметров генерации



Зависимость времени генерации ландшафта от размера шага задания точек

Заключение

В ходе выполнения курсовой работы была достигнута цель – была разработана программа для генерации ландшафта и его визуализации. Были выполнены все поставленные цели:

- 1) была формализована задача;
- 2) были проанализированы и выбраны алгоритмы для создания изображения и генерации ландшафта;
- 3) была спроектирована программа для генерации ландшафта и его визуализации;
- 4) были выбраны средства реализации, а также была реализована спроектированная программа;
- 5) была исследована зависимость скорости генерации ландшафта от параметров генерации.