

Разработка программного обеспечения для моделирования водопада

Студент: Артюхин Николай Павлович ИУ7-51Б

Научный руководитель: Барышникова Марина Юрьевна

Москва, 2022 г.

Цель и задачи

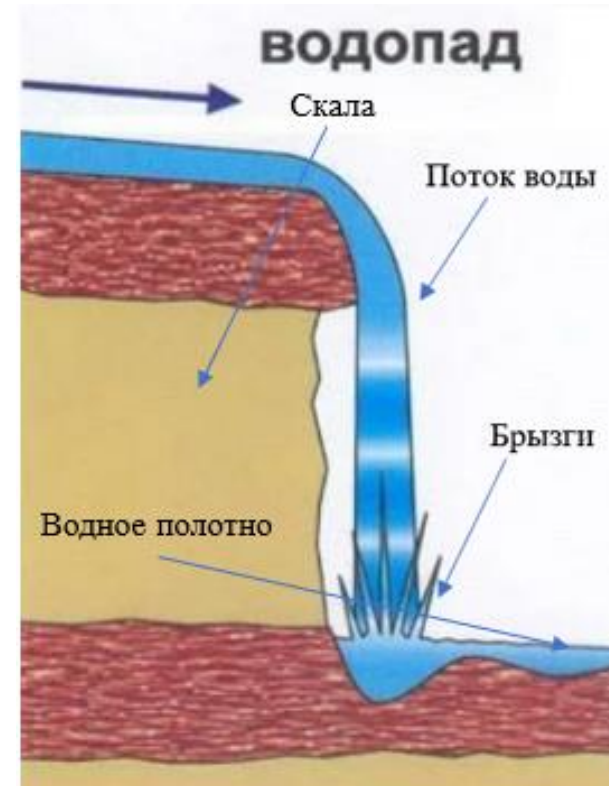
Цель работы – разработать программное обеспечение, обеспечивающее динамическую визуализацию модели искусственного водопада.

Задачи:

- описать структуру трехмерной сцены, включая объекты, из которых она состоит;
- проанализировать существующие алгоритмы, которые можно использовать для моделирования водопада, выбрать наиболее подходящий из них;
- проанализировать алгоритмы удаления невидимых линий и поверхностей и выбрать наиболее подходящий из них;
- реализовать выбранные алгоритмы;
- разработать структуру классов программного обеспечения;
- провести эксперимент по замеру производительности программного обеспечения в зависимости от используемых языков программирования.

Описание объектов сцены

- Скала (уступ водопада)
- Поток воды
- Брызги
- Водное полотно
- Берега
- Камера

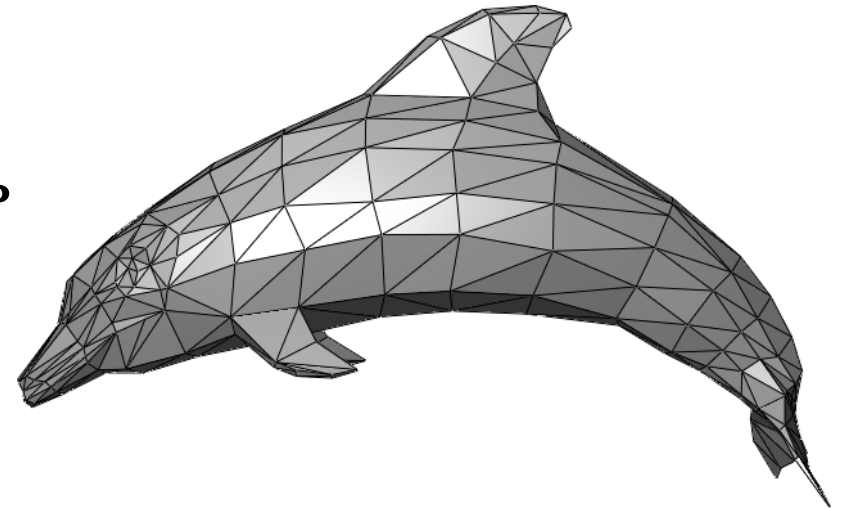


Анализ методов моделирования водопада

	Уравнение Навье-Стокса	Сеточные методы	Система частиц	Комбинированные методы
Эффективность по времени	Низкая	Высокая	Высокая	Высокая
Реалистичность	Высокая	Низкая	Средняя	Высокая
Сложность реализации	Высокая	Низкая	Низкая	Высокая
Возможность реализации брызг	Нет	Нет	Да	Да

Способ представления трехмерной модели

- В рамках данного проекта в качестве представления модели была выбрана полигональная сетка.
- **Полигональная сетка** - это совокупность вершин, рёбер и граней, которые определяют форму многогранного объекта в трехмерной компьютерной графике и объемном моделировании.



Анализ алгоритмов удаления невидимых линий и поверхностей

	Алгоритм Z-буфера	Алгоритм Робертса	Алгоритм художника	Алгоритм Варнока	Алгоритм обратной трассировки лучей
Сложность N – число граней, m – число пикселей	$O(mN)$	$O(N^2)$	$O(mN)$	$O(mN)$	$O(mN)$
Реалистичность	Средняя	Средняя	Низкая	Средняя	Высокая
Сложность реализации	Низкая	Высокая	Высокая	Средняя	Высокая
Эффективность по памяти	Низкая	Средняя	Высокая	Средняя	Высокая
Эффективность по времени	Высокая	Средняя	Низкая	Средняя	Низкая

Выбор API для отрисовки изображения

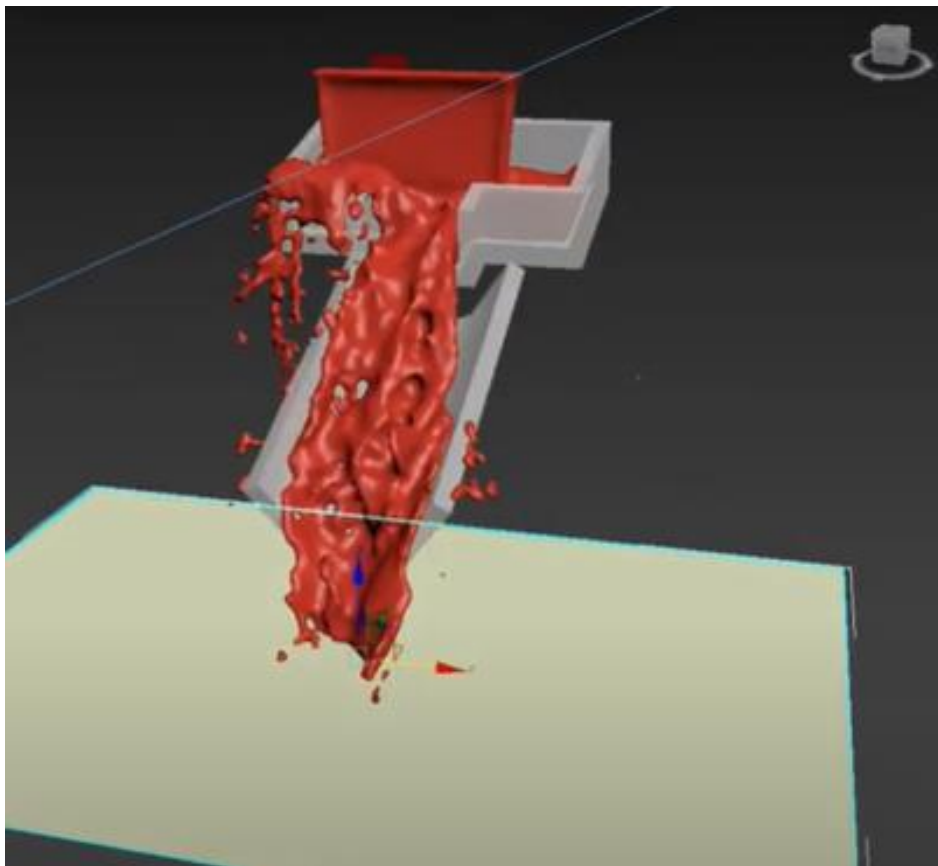
	Vulkan	OpenGL	DirectX
Кроссплатформенность	Да	Да	Нет (только Windows)
Открытый код	Да	Да	Нет
Библиотека для ЯП Python	Нет	Да	Да
Поддержка видеокарт предыдущего поколения	Да	Да	Нет

Выбор модели освещения

	Модель Ламберта	Модель Фонга
Реалистичность	Средняя	Высокая
Сложность реализации	Низкая	Высокая

Существующее ПО

Autodesk 3ds Max



Blender

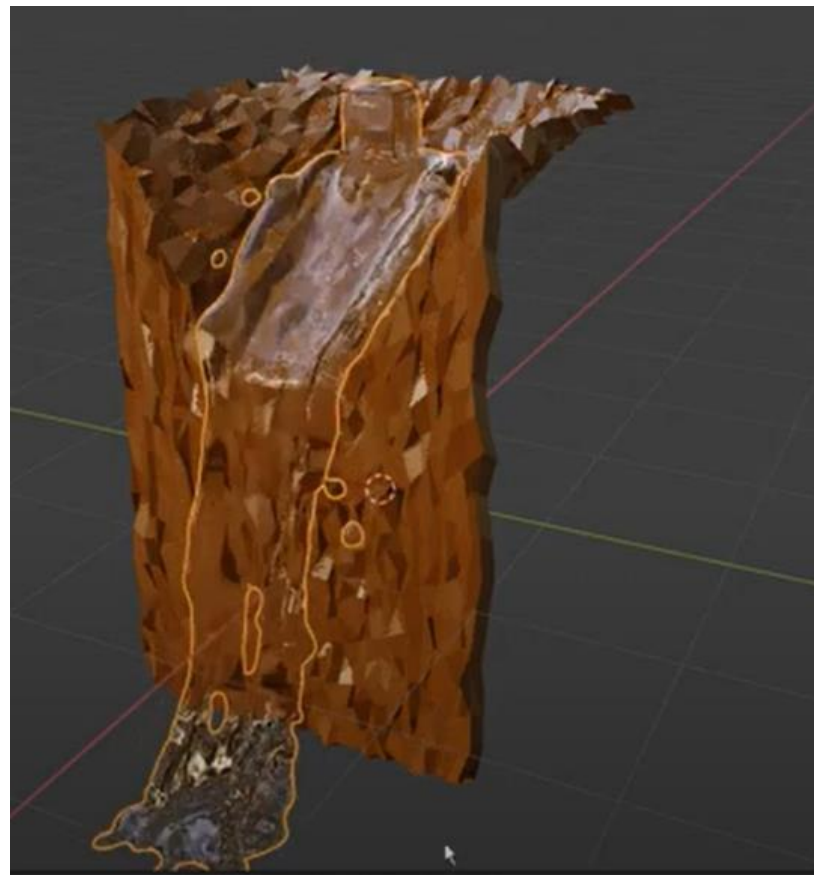


Схема алгоритма движения частиц водопада

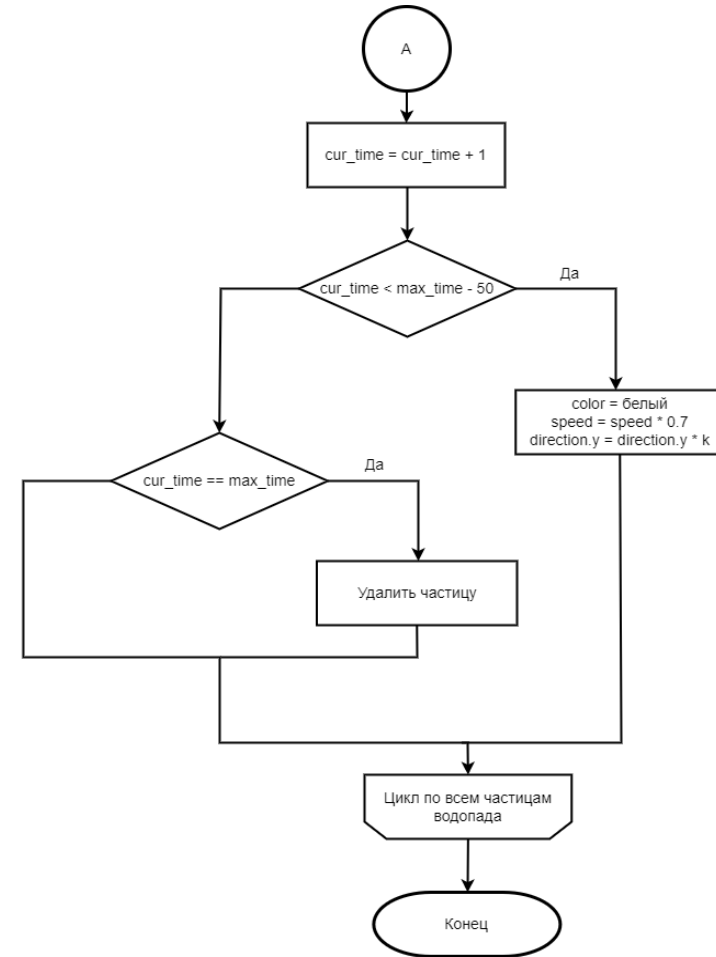
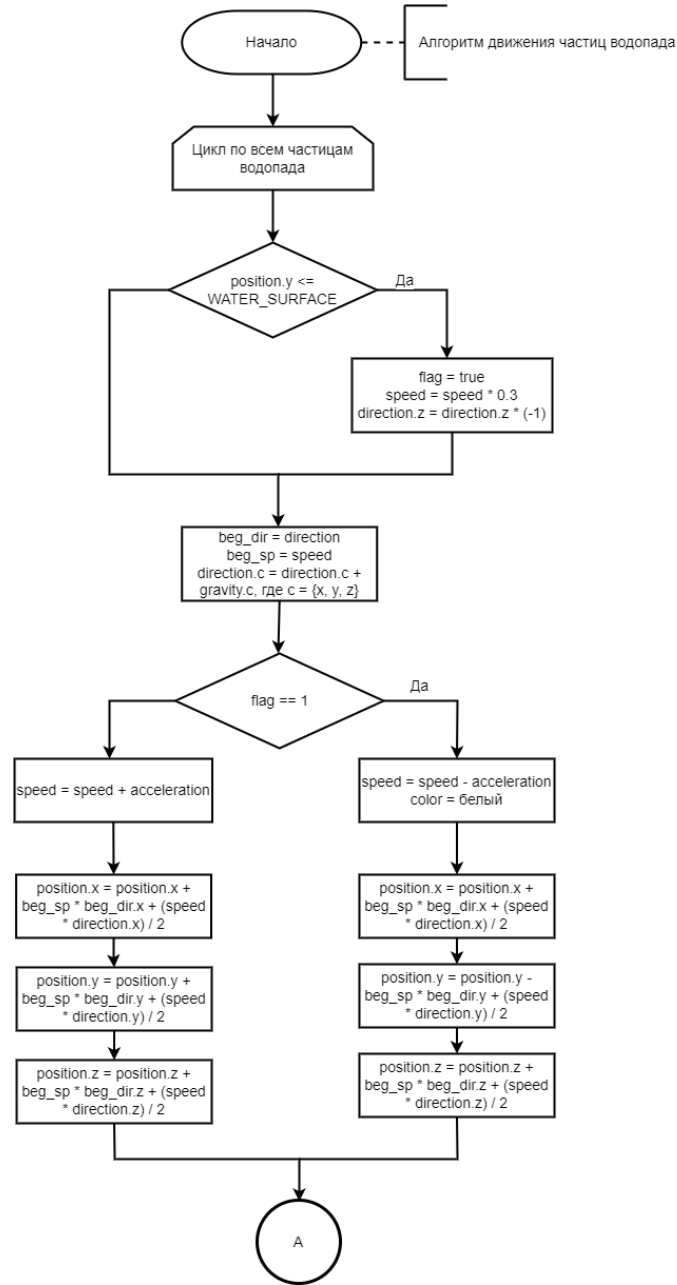


Схема алгоритма z-буфера

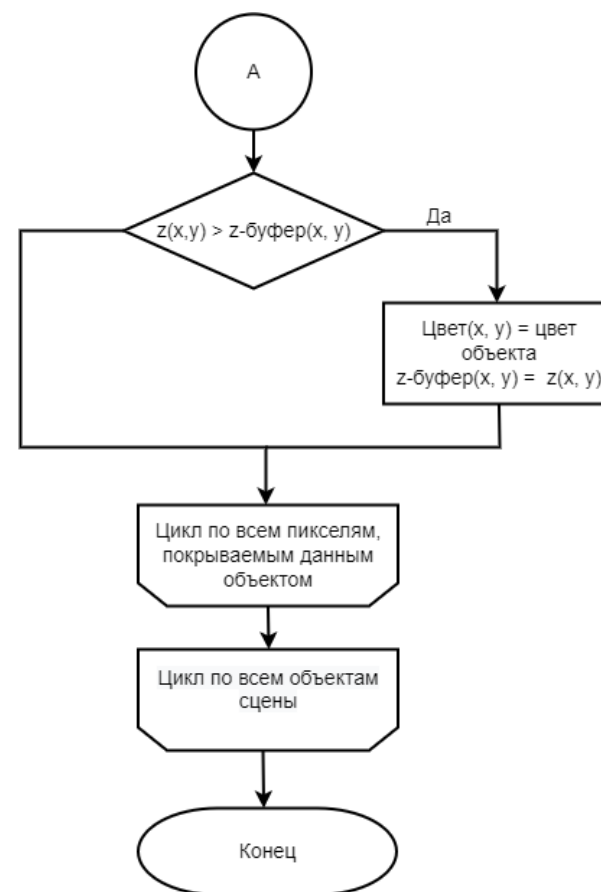
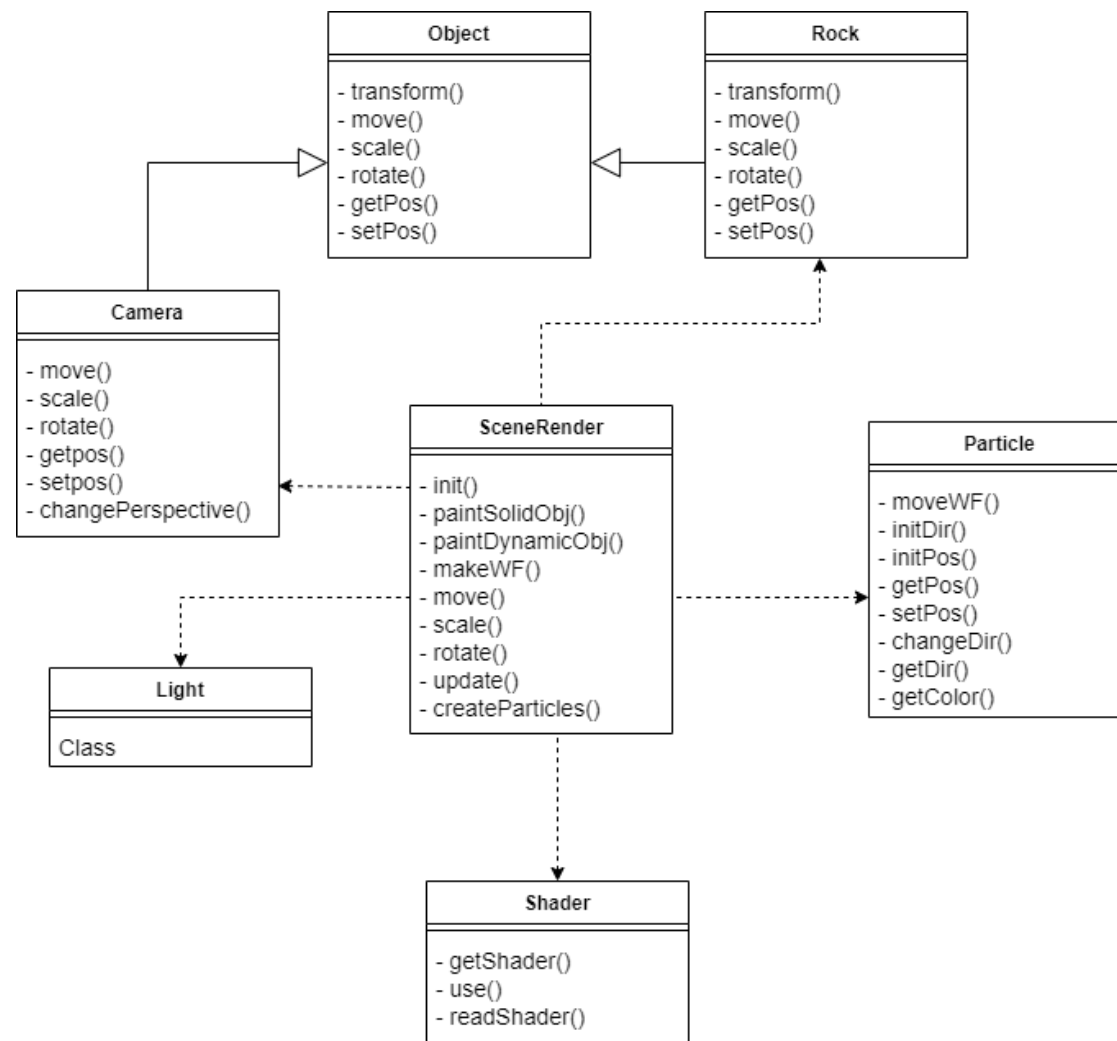
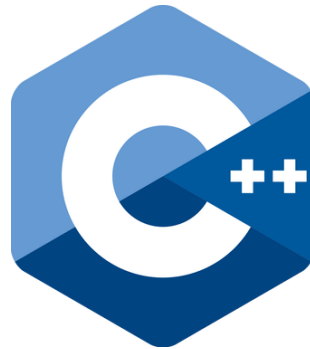
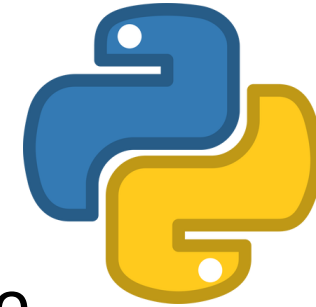


Диаграмма классов



Выбор языков программирования и сред разработки

- Языки программирования: Python, C++
- C++ поддерживает объектно-ориентированное программирование, имеет хорошую документацию, строго типизированный (защита от неконтролируемых ошибок).
- Python поддерживает объектно-ориентированное программирование, предоставляет все необходимые графические библиотеки для решения поставленной задачи, имеет хорошую документацию.
- Разработка интерфейса: Qt Designer
- Среды разработки: PyCharm, Clion

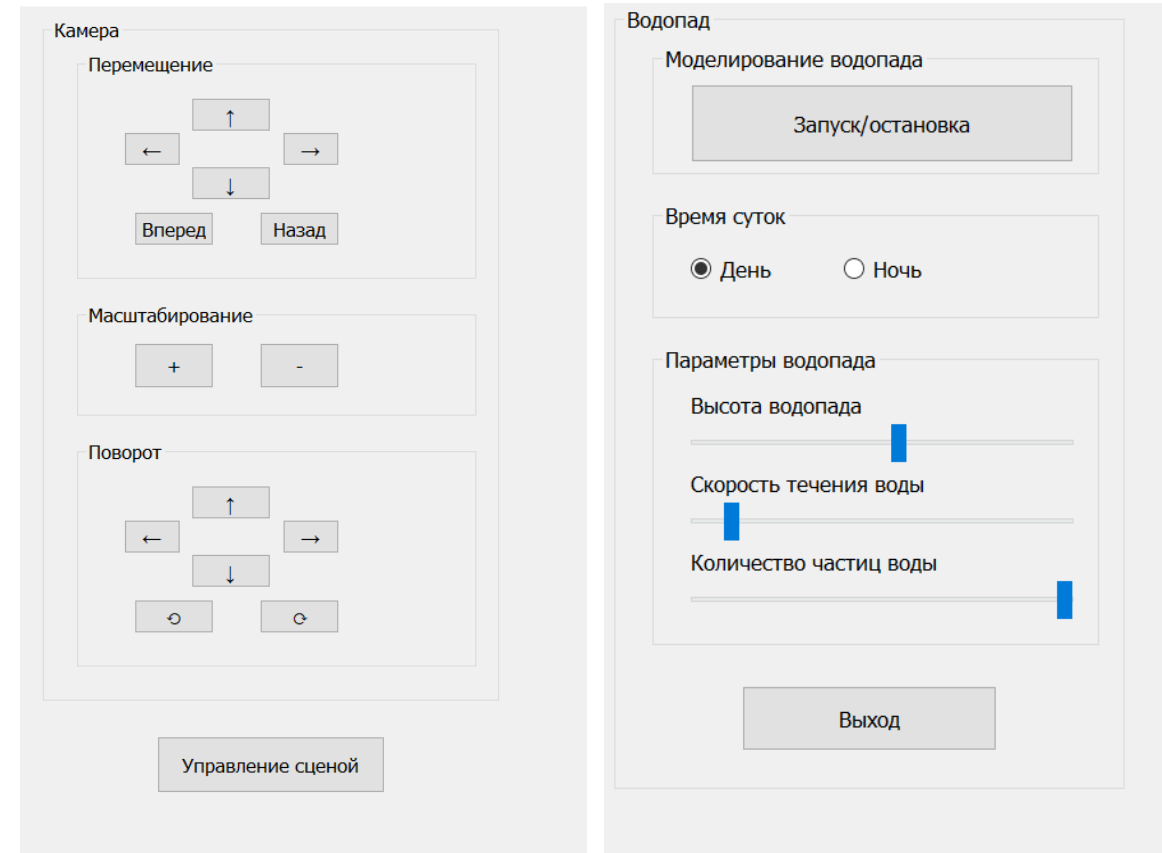


Интерфейс программы

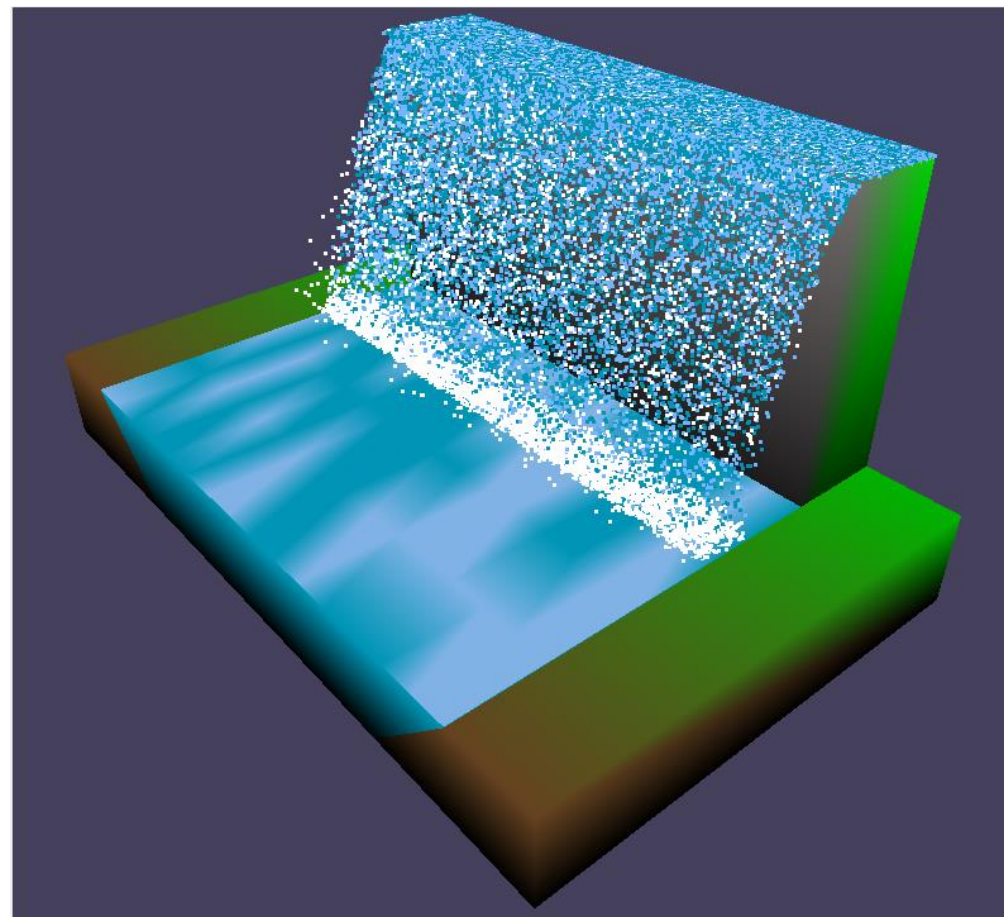
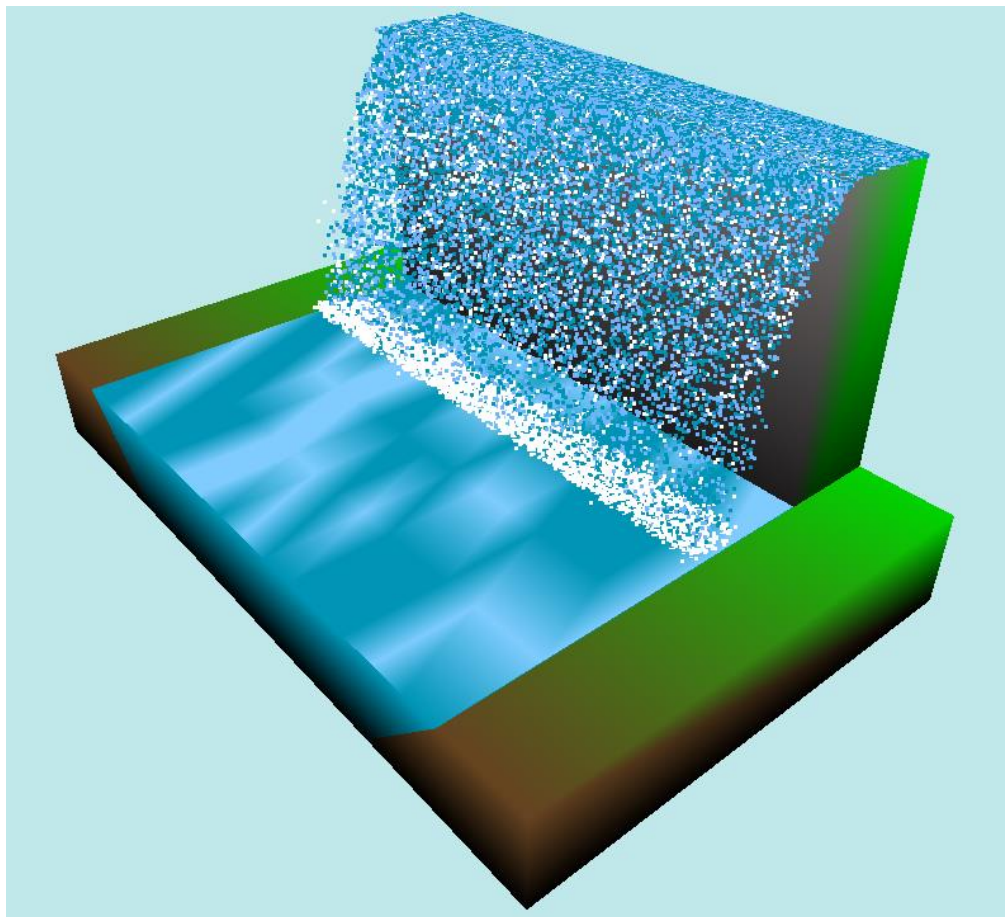
Пользователь может:

- запустить/остановить моделирование водопада;
- выбрать дневное или ночное освещение;
- изменить высоту водопада;
- изменить скорость течения воды;
- изменить количество частиц воды;
- управлять камерой/сценой.

Корректность ввода данных проверяется автоматически за счет средств, предоставляемых интерфейсом Qt Designer.



Примеры работы программы



Эксперимент

- **Цель эксперимента** – проведение тестирования производительности при создании сцен с различной степенью нагрузки на программное обеспечение, полностью написанное на языке программирования Python, и программное обеспечение, написанное на Python и C++.

Нагрузка на ПО будет меняться в зависимости от количества частиц, из которых состоит водопад.

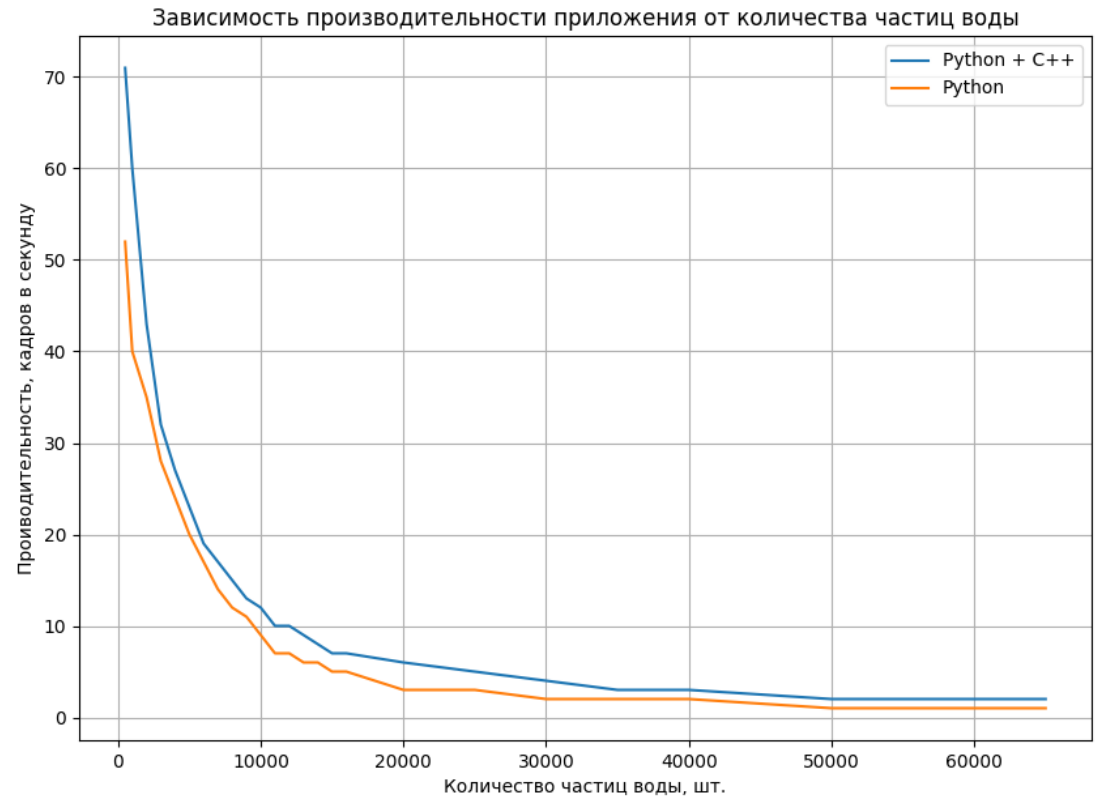
- **Результаты эксперимента**

Справа представлена таблица зависимости производительности ПО, написанного полностью на Python, и ПО, написанного на Python (графика) и C++ (алгоритм движения частиц, алгоритм z-буфера), в зависимости от количества частиц в водопаде.

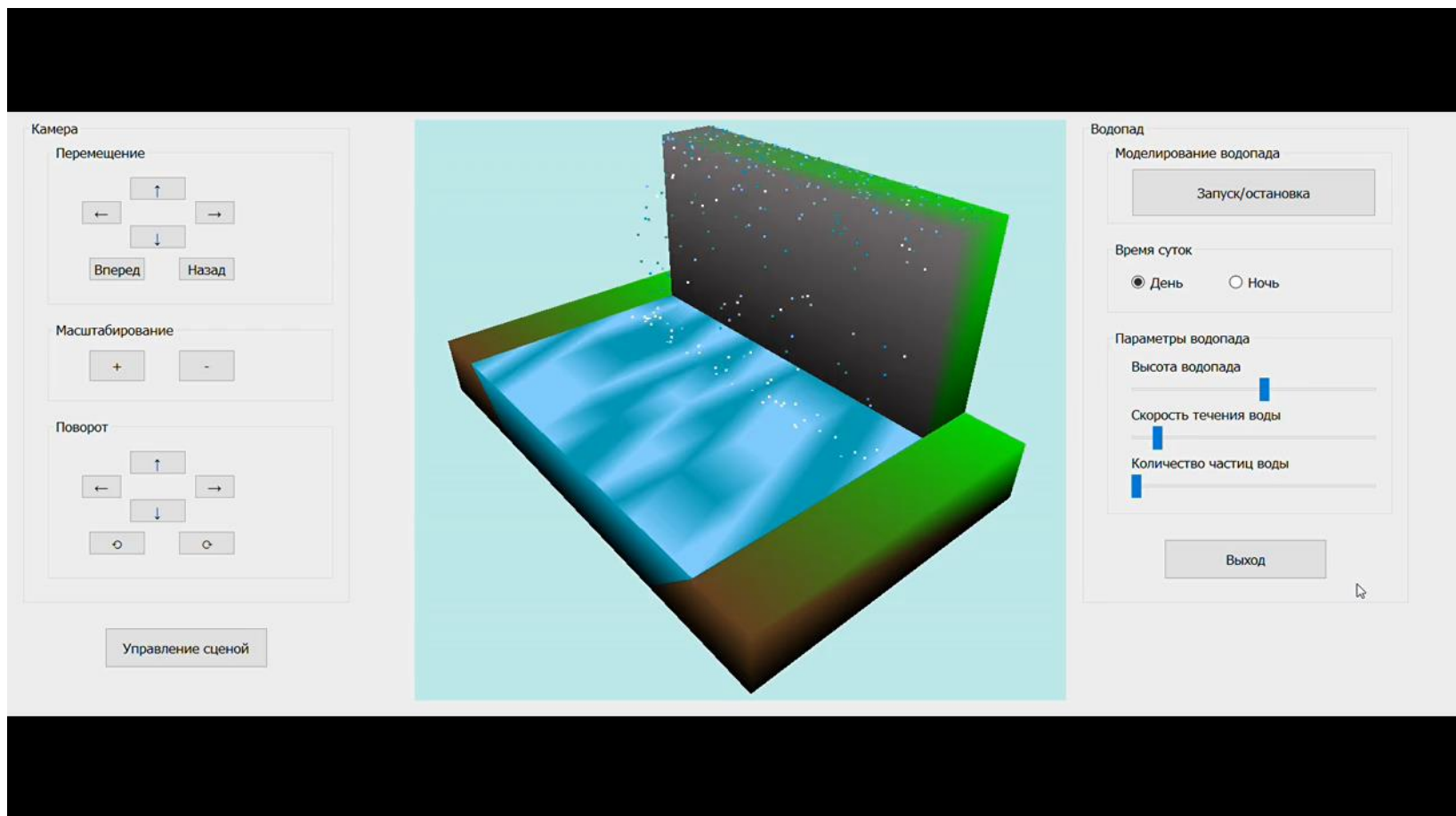
Количество частиц, штук	Производительность, кадров в секунду	
	Python и C++	Python
500	71	52
1000	60	40
2000	43	35
3000	32	28
4000	27	24
5000	23	20
6000	19	17
7000	17	14
8000	15	12
9000	13	11
10000	12	9
15000	7	5
20000	6	3
25000	5	3
30000	4	2
35000	3	2
40000	3	2
50000	2	1
65000	2	1

Результаты эксперимента

- Справа представлен график зависимости производительности ПО, написанного полностью на Python, и ПО, написанного на Python (графика) и C++ (алгоритм движения частиц, алгоритм z-буфера), в зависимости от количества частиц в водопаде.
- По графику, полученному в результате эксперимента, видно, что производительность ПО (количество кадров в секунду) уменьшается экспоненциально при линейном увеличении количества частиц в водопаде.
- Также можно увидеть, что ПО, написанное на Python и C++ показало более высокую производительность, чем ПО, полностью написанное на Python.

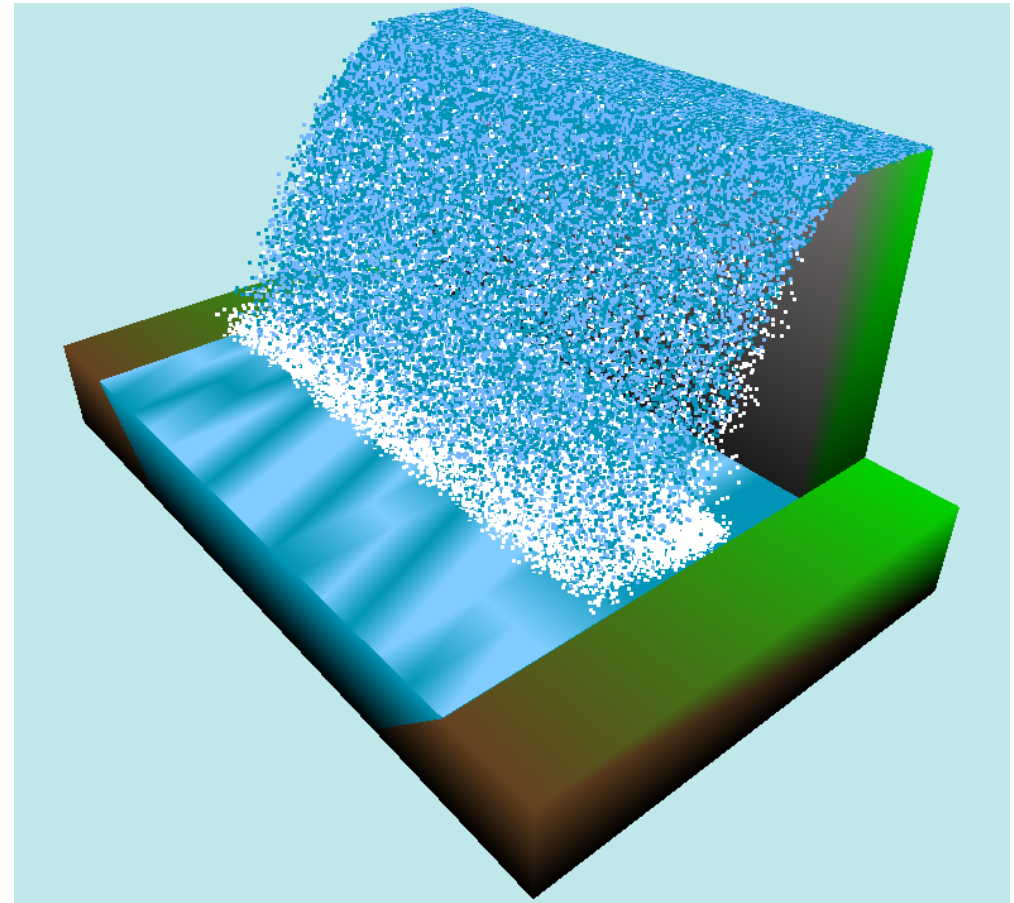


Реалистичность водопада, в зависимости от числа частиц



Реалистичность водопада, в зависимости от числа частиц

- На рисунке представлен водопад из 200000 частиц.
- Видно, что вода стала почти непрерывной.
- При этом за реалистичность изображения приходится платить производительностью приложения.



Заключение

Цель курсовой работы была достигнута: разработано программное обеспечение, обеспечивающее динамическую визуализацию модели искусственного водопада.

Все поставленные **задачи** были решены:

- описана структура трехмерной сцены;
- проанализированы алгоритмы необходимые для моделирования водопада;
- реализованы выбранные алгоритмы;
- разработана структура классов ПО;
- проведен эксперимент по замеру производительности программного обеспечения в зависимости от используемых языков программирования.