



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)

ФАКУЛЬТЕТ \_\_\_\_\_ «Информатика и системы управления» \_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии» \_\_\_\_\_

## **ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

Студент \_\_\_\_\_ Артюхин Николай Павлович \_\_\_\_\_  
*фамилия, имя, отчество*

Группа \_\_\_\_\_ ИУ7-41Б \_\_\_\_\_

Тип практики \_\_\_\_\_ технологическая \_\_\_\_\_

Название предприятия \_\_\_\_\_ МГТУ им. Н. Э. Баумана, каф. ИУ7 \_\_\_\_\_

Студент \_\_\_\_\_ Артюхин Н. П. \_\_\_\_\_  
*подпись, дата* *фамилия, и.о.*

Руководитель практики \_\_\_\_\_ Кузов А. В. \_\_\_\_\_  
*подпись, дата* *фамилия, и.о.*

Оценка \_\_\_\_\_

**«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

---

УТВЕРЖДАЮ

Заведующий кафедрой ИУ7

И.В. Рудаков

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

## З А Д А Н И Е

**на прохождение производственной практики**

**технологическая**

---

Тип практики

Студент Артюхин Николай Павлович 2 курса группы ИУ7-41Б  
Фамилия Имя Отчество № курса индекс группы

в период с 30.06.2022 г. по 20.07.2022 г.

Предприятие: МГТУ им. Н.Э. Баумана (кафедра ИУ7)

Руководитель практики от кафедры: Кузов Андрей Владимирович  
(Фамилия Имя Отчество полностью, должность)

Задание:

- 1. Начать разработку программного обеспечения для моделирования водопада.**
- 2. Решить вопрос о способе представления объектов, проанализировать и выбрать алгоритмы для их обработки.**
- 3. Закрепить знания и навыки, полученные в ходе лекционных занятий по пройденным курсам.**

Дата выдачи задания « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Руководитель практики от кафедры \_\_\_\_\_ / \_\_\_\_\_ /

Студент \_\_\_\_\_ / \_\_\_\_\_ /

## Оглавление

Введение.....	4
1. Аналитический раздел .....	5
1.1 Модель классического водопада .....	5
1.2 Формализация модели водопада .....	6
1.3 Анализ и выбор метода визуализации водопада.....	7
1.4 Анализ и выбор формы задания трехмерной модели .....	13
1.5 Анализ и выбор алгоритма удаления невидимых ребер и поверхностей .	15
1.6 Анализ и выбор модели освещения.....	22
1.7 Анализ существующего программного обеспечения.....	24
1.8 Вывод .....	26
2. Конструкторский раздел .....	26
2.1 Метод визуализации водопада, основанный на системе частиц.....	25
2.2 Алгоритм удаления невидимых ребер и поверхностей, использующий z-буфер.....	28
2.3 Модель освещения Ламберта .....	30
2.4 Визуализация изображения скалы.....	31
2.5 Выбор используемых типов и структур данных.....	31
2.6 Структура программы.....	32
2.7 Диаграмма классов .....	33
2.8 Вывод .....	36
3. Технологический раздел .....	36
3.1 Требования к программному обеспечению .....	36
3.2 Выбор языка программирования и среды разработки .....	36
3.3 Интерфейс программного обеспечения .....	37
3.4 Вывод .....	39
Заключение.....	39
Список использованных источников .....	40

## Введение

Компьютерная графика (машинная графика) – это область деятельности, где компьютеры и специальное программное обеспечение используются в качестве инструмента для синтеза (создания), анализа и обработки изображений. В современном мире область применения компьютерной графики довольно широка. Машинная графика чаще всего используется при создании компьютерных игр и в кинематографе.

Методы отрисовки постоянно развиваются, также появляются новые способы. Сегодня наибольшее внимание уделяется алгоритмам получения реалистических изображений, это одна из основных задач компьютерной графики. Данные алгоритмы являются одними из самых затратных по времени и памяти, потому что они должны учитывать физические явления: отражение, преломление, рассеивание, поглощение света. Чем выше точность алгоритмов, то есть чем качественнее полученное на выходе алгоритма изображение, тем выше их сложность, что обычно приводит к увеличению затрат по времени и по памяти.

При создании динамической сцены на каждом временном промежутке необходимо производить расчеты заново, что приводит к большому числу вычислений и временным затратам.

Одна из самых сложных тем для моделирования – жидкости. На сегодняшний день существует серьезная необходимость в эффективной и качественной отрисовке морей, океанов, озер, рек и множества других водоемов [1].

Темой данной работы является моделирование водопада с учетом брызг, которые возникают у его подножия при падении воды.

Целью данной работы является анализ алгоритмов, которые можно использовать для моделирования водопада, выбор наиболее эффективных из них

и лучше всего справляющихся с поставленной задачей, их практическая реализация и при необходимости адаптация к условиям задачи.

## 1. Аналитический раздел

### 1.1 Модель классического водопада

Водопад – это падение воды в реке с уступа, пересекающего речное русло. В отличие от речных порогов, для водопадов характерны резкий перепад высоты речного дна и отвесность падения. Угол падения воды в классических водопадах составляет примерно 90 градусов. Высота водопада должна быть более 1 метра.

Образование водопадов в природе на реке — это очень долгий временной процесс. Обычно водопады возникают при резком перепаде высот, который появился еще до того, как там потекла река, или из-за того, что водяной поток размывает мягкие осадочные породы, вследствие чего происходит обрушение почвы и на границе мягкой и твердой пород образуется вертикальный уступ, с которого и начинает падать вода (рисунок 1.1).



Рисунок 1.1 - Модель водопада

Для более качественного и реалистичного изображения при построении модели водопада нужно учитывать не только сам поток воды, но и брызги, которые образуется при ударе потока об воду и брызги от самого потока воды (рисунок 1.2).

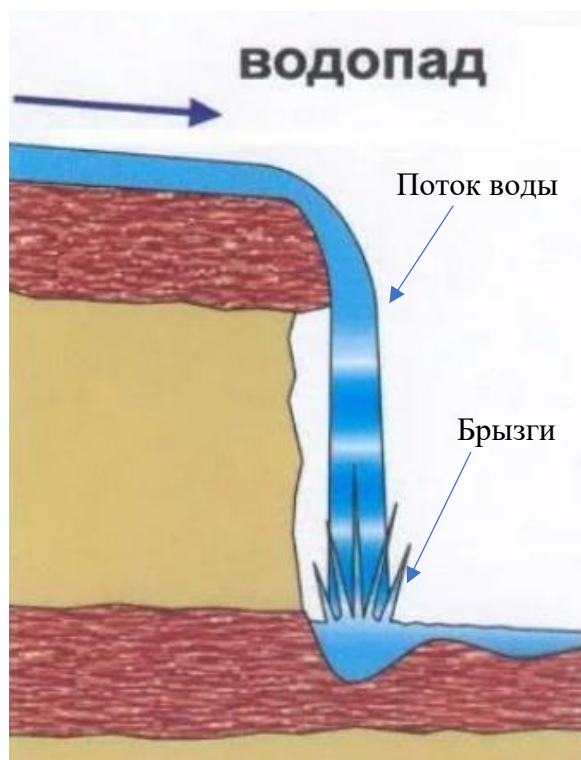


Рисунок 1.2 - Схема водопада

## 1.2 Формализация модели водопада

Модель водопада будет задаваться следующими характеристиками:

- Высота – высота уступа, с которого падает вода (число типа float в интервале от 1 до 30 метров)
- Ширина – ширина уступа, по которому течет вода (число типа float в интервале от 1 до 10 метров)
- Скорость – скорость падения воды с уступа (число типа float)

Частью сцены будет являться скала, играющая роль уступа, с которого падает вода. Она будет задаваться геометрическим объектом – параллелепипедом, ширину и высоту которого сможет изменять пользователь.

### 1.3 Анализ и выбор метода визуализации водопада

Водопад – это поток воды с частицами брызг. Для моделирования текучей воды было разработано несколько основных методов, которые будут приведены далее.

#### 1.3.1 Метод, основанный на уравнении Навье-Стокса

Уравнение Навье-Стокса – это тип нелинейных дифференциальных уравнений в частных производных, которые описывают движение жидких и газообразных сред [2].

$$\frac{d\vec{v}}{dt} = -(\vec{v}\nabla)\vec{v} + \nu\Delta\vec{v} - \frac{1}{\rho}\nabla p + \vec{f}, \quad (1.1)$$

где  $\vec{v}$  – векторное поле скорости,  $t$  – время,  $\nabla$  – оператор набла,  $\nu$  – коэффициент вязкости,  $\Delta$  – векторный оператор Лапласа,  $\rho$  – коэффициент плотности жидкости,  $p$  – давление,  $\vec{f}$  – векторное поле массовых сил (внешние силы, применяемые к жидкости).

Формула 1.1 позволяет вычислить скорость изменения скорости жидкости в точке.

Данное уравнение пока не имеет решения в общем виде, поэтому при разработке алгоритмов используются частные решения. Уравнение Навье-Стокса обычно применяется для математического моделирования сложных моделей природных явлений.

Методы, основанные на этом уравнении, трудно реализуются из-за сложной математики и большого количества трудоемких вычислений, которые придется производить компьютеру при обработке формул. Таким образом, данные методы являются очень неэффективными по времени, и подробное их рассмотрение не имеет смысла.

### 1.3.2 Сеточные методы

Сетки довольно часто используются при моделировании жидкостей. В каждой ячейке регулярной кубической сетки хранится информация о точке поверхности, задающей картину. Сеточные методы неэффективны по памяти. Качество получаемого изображения зависит от количества ячеек, чем больше ячеек, тем выше качество изображения.

Преимущества сеточных методов:

- простота визуализации результата симуляции;
- точно известны границы целевой области (возможность встраивания водяного потока в нужное место).

Наиболее распространенные сеточные методы будут рассмотрены далее.

#### Метод, основанный на уравнении Эйлера

В данном методе используется уравнение Эйлера невязкой частицы [3]:

$$\frac{d\vec{u}}{dt} = -(u\nabla)\vec{u} + \frac{\vec{f}}{\rho} - \frac{\nabla p}{\rho}, \quad (1.2)$$

с учетом ограничения несжимаемости:

$$\nabla \cdot \vec{u} = 0, \quad (1.3)$$

где  $\vec{u} = [u, v, w]^T$  – поле скорости жидкости,  $p$  – давление,  $t$  – время,  $\rho$  – плотность жидкости,  $\vec{f}$  – поле внешних сил.

С помощью данного метода можно моделировать крупномасштабные трехмерные жидкости.

В методе, основанном на уравнении Эйлера, используется гибридное представление сетки, состоящее из правильных кубических ячеек поверх слоя высоких ячеек.



## **Метод моделирования пузырьков воздуха в воде**

Данный метод является физически обоснованным методом расчета размеров пузырьков воздуха. Он основан на скорости замерзания и давлении и состоит из двух процессов: процесса моделирования замерзания и процесса образования пузырьков. В отличие от большинства других методов, этот метод может представлять большое количество мелких пузырьков, которые не могут быть представлены с помощью решений имитационных сеток [4].

### **1.3.3 Методы, основанные на частицах**

Метод частиц основывается на использовании трилинейной интерполяции для каждой частицы для определения ее скорости движения. Каждая частица перемещается в соответствии с определенным инерциальным физическим уравнением.

Преимущество данного метода в том, что он имеет довольно небольшие затраты по ресурсам компьютера. Большое количество частиц позволяет достичь максимальной точности изображения, но чем больше частиц, тем больше нагрузка на компьютер.

Главным недостатком данного подхода является невозможность точно определить границы, в которых должен находиться водяной поток, а это приводит к появлению дополнительных проверок; также нет возможности создавать извивающиеся водяные потоки.

Наиболее распространенные методы, основанные на частицах, будут рассмотрены далее.

### **Метод, использующий диаграммы Вороного**

Основным компонентом данного метода является аппроксимация геометрии пены путем обработки частиц пузырьков как участков взвешенной диаграммы Вороного. Информация о связности, предоставляемая диаграммой

Вороного, позволяет нам точно моделировать различные эффекты взаимодействия между пузырьками.

Пусть  $P$  – точка, заданная в  $R^3$ . Тогда для любой точки  $p$ , принадлежащей  $P$ , ячейка Вороного  $V_p$  точки  $p$  определяется как место точки в  $R^3$ , имеющих  $p$  в качестве ближайшего соседа в  $P$ :

$$V_p = \{x \in R, \forall q \in P: |x - p| < |x - q|\} \quad (1.4)$$

Каждая ячейка Вороного – выпуклая, а граница состоит из выпуклых граней меньшего размера. Совокупность ячеек Вороного и их граней образует комплекс ячеек в  $R^3$ , что и называется диаграммой Вороного [5].

Используя ячейки Вороного и веса (взвешенную диаграмму Вороного), можно решить проблему потери объема при моделировании пены, что является общей проблемой во многих подходах.

Данный метод может быть использован для моделирования жидкости, например, для водопада.

### **Метод полу-Лагранжа**

Метод полу-Лагранжа, объединенный с новым подходом расчета жидкости вокруг объектов, позволяет эффективно решать уравнения движения жидкости, сохраняя при этом достаточно деталей, чтобы получить реалистичное изображение.

Высококачественная поверхность получается из результирующего поля скоростей с использованием нового адаптивного метода для создания неявной поверхности [6].

### **Метод, основанный на уравнениях движения частиц по криволинейной траектории.**

В данном методе при моделировании каждая частица считается независимой и для каждой вычисляется ее место в соответствии с физическим

уравнением. Частицы случайным образом распределяются по линии течения воды.

В системе частиц каждая отдельно взятая частица будет рассматриваться в независимости от остальных частиц. В качестве приоритетного способа реализации модели частицы, падающей с вершины уступа водопада, используется подход, основанный на криволинейном равноускоренном движении.

Таким образом, можно получить качественный результат, используя известные физические законы для движения частицы.

Для моделирования потока воды в качестве основного уравнения берется закон равноускоренного движения [7]:

$$y = y_0 + v_0 t + \frac{at^2}{2}, \quad (1.5)$$

где  $y_0$  – начальное положение частицы,  $v_0$  – начальная скорость движения частицы,  $t$  – время,  $a$  – ускорение движения частицы.

При этом скорость частицы при равноускоренном движении вычисляется следующим образом:

$$v = v_0 + at, \quad (1.6)$$

при  $a = \text{const}$ .

Ускорение вычисляется следующим образом:

$$a = \frac{dv}{dt} \quad (1.7)$$

*Нахождение скорости, в зависимости от направления, которое было задано частице.*

Вектор направления в каждый момент времени вычисляется, путем сложения вектора направления движения и вектора гравитации. При этом получается новый вектор направления (резльтирующий вектор), тогда скорость будет вычисляться следующим образом:

$$\vec{v} = v * \vec{R}, \quad (1.8)$$

где  $\vec{v}$  – вектор скорости частицы,  $v$  – скалярная скорость частицы,  $\vec{R}$  – результирующий вектор вектора направления и вектора гравитации.

### 1.3.4 Комбинированные методы

Комбинированные методы – это методы, полученные в результате объединения методов, основанных на частицах и на сетках. Совместное использование двух методов помогает одновременно с потоком воды моделировать и другие виды водяных потоков. Данный метод является предпочтительным при создании полноценной системы.

Наиболее распространённые комбинированные методы рассмотрены далее.

#### Метод, напрямую использующий методы на основе сеток и частиц

Области жидкости, которые не могут быть представлены полем высоты (брызги от удара воды; волны, разбивающиеся о скалы), представляются частицами и превращаются в брызги и пену.

Все частицы рассматриваются как простые невзаимодействующие точечные массы, которые обмениваются массой и моментом с жидкостью поля высоты [8].

#### Метод, моделирующий пузырьки воздуха и пену

Данный метод основывается на уравнении гидродинамики сглаженных частиц (SPH). Здесь жидкость превращается в набор частиц  $i$  с положением  $x_i$  и скоростью  $v_i$ . Обычно количество частиц  $A_i$  аппроксимируется функцией

сглаживания, которая интерполирует  $A_i$ , используя конечный набор точек выборки  $j$ , находящихся на расстоянии  $h$  - окрестность частицы. Функция сглаживания определяется следующим образом [9]:

$$A_i = \sum_j \frac{m_j}{\rho_j} A_j W(x_i - x_j, h), \quad (1.9)$$

где  $m_j$  – масса  $j$ ,  $\rho_j$  – ее плотность,  $W(x_i - x_j, h)$  – функция ядра (кубический сплайн).

В данном методе этапы моделирования пузырьков воздуха и пены рассматриваются отдельно. В нем используется сила сопротивления для совмещения обоих этапов без особого увеличения вычислительной нагрузки и функция насыщения, которая зависит от объема.

При достижении поверхности пузырьки преобразуются в пену, а затем удаляются через определенное время.

## **Вывод**

Для моделирования водопада выбран метод, основанный на частицах.

Данный метод позволяет смоделировать не только сам водопад (основной поток воды), который состоит из большого количества частиц, но и брызги и пену, которые тоже состоят из частиц воды. Также у данного метода довольно небольшие затраты по времени и по памяти по сравнению с другими рассмотренными выше методами. Качество изображения можно повысить путем увеличения количества частиц.

## **1.4 Анализ и выбор формы задания трехмерной модели**

### **1.4.1 Основные формы задания моделей**

1. Каркасная (проволочная) модель

Хранится информация только о вершинах и ребрах объекта.

Преимущество – простая.

Недостаток – далеко не всегда передает правильную информацию о моделируемом объекте (все ребра видимы, при наличии отверстий не понятно какие грани связывает отверстие).

## 2. Поверхностная модель

Поверхностная модель объекта является оболочкой объекта, она пустая внутри. Хранятся только информация о внешних геометрических параметрах объекта. Данный тип модели чаще всего используется в компьютерной графике. При этом могут использоваться различные типы поверхностей, ограничивающих объект, такие как полигональные модели, поверхности второго порядка и другие.

Преимущество – довольно точно передает представление о форме моделируемого объекта.

Недостаток – не подходит для создания изделий, так как нет информации о материале.

## 3. Объемная (твердотельная) модель

Твердотельная модель содержит информацию о том, где находится материал, а где пустота. Это делается с помощью указания направления внутренней нормали.

Преимущество – наиболее полное представление о моделируемом объекте.

Недостаток – довольно сложное представление объекта, для большинства задач компьютерной графики данная модель является избыточной.

## Вывод

Для представления скалы (уступа водопада) была выбрана поверхностная модель, так как каркасные модели могут привести к неправильному восприятию

формы моделируемого объекта, а реализация твердотельной модели потребует большого количества ресурсов на воспроизведение деталей, которые не являются необходимыми для решения поставленной задачи.

#### **1.4.2 Способы задания поверхностной модели**

##### **1. Аналитический способ**

Данный способ характеризуется описанием модели объекта, доступной в неявной форме, то есть для получения визуальных характеристик необходимо дополнительно вычислять некоторую функцию, которая зависит от некоего параметра.

##### **2. Полигональная сетка**

Данный способ характеризуется совокупностью вершин, граней и ребер, которые определяют форму многогранного объекта в трехмерной компьютерной графике.

#### **Вывод**

В качестве способа задания поверхностной модели была выбрана полигональная сетка, чтобы избежать дополнительных вычислительных затрат, которые есть в аналитическом способе.

### **1.5 Анализ и выбор алгоритма удаления невидимых ребер и поверхностей**

Перед выбором алгоритма удаления невидимых ребер выделим несколько свойств, которыми должен обладать выбранный алгоритм, чтобы обеспечить оптимальную работу и реалистичное изображение.

Свойства:

- Алгоритм может работать как в объектном пространстве, так и в пространстве изображений.
- Алгоритм должен быть эффективным по времени и по памяти.

- Алгоритм должен иметь достаточно высокую реалистичность изображения.

### 1.5.1 Алгоритм, использующий Z-буфер

Суть алгоритма – это использование двух буферов: буфера кадра, в котором хранятся атрибуты каждого пикселя, и Z-буфера, в котором хранятся информация о координате Z для каждого пикселя.

Начальная инициализация буферов:

- z-буфер заполняется минимально возможным значением Z;
- буфер кадра заполняется фоновым значением цвета.

Каждый многоугольник преобразуется в растровую форму и записывается в буфер кадра.

В процессе подсчета глубины нового пикселя, он сравнивается с тем значением, которое уже лежит в z-буфере. Если новый пиксель расположен ближе к наблюдателю, чем предыдущий, то он заносится в буфер кадра и происходит корректировка z-буфера [10, с. 321].

Для решения задачи вычисления глубины Z каждый многоугольник описывается уравнением  $ax + by + cz + d = 0$ . При  $c = 0$  многоугольник для наблюдателя вырождается в линию.

Для некоторой сканирующей строки  $y = \text{const}$  можно рекуррентно высчитывать  $z1$  (глубину пикселя) для каждого  $x1 = x + dx$ :

$$z1 = z - a/c, \quad (1.10)$$

при  $dx = 1$  (шаг раstra).

Для невыпуклых многогранников предварительно потребуется удалить нелицевые грани.

Преимущества алгоритма:

- простота реализации;
- отсутствие сортировок;



- линейный рост времени работы алгоритма.

Недостатки алгоритма:

- сложная реализация эффекта прозрачности;
- неэффективность по памяти (большой расход памяти).

Для экономии памяти можно использовать алгоритм построчного сканирования, использующий z-буфер. (в z-буфере будет храниться одна сканирующая строка).

### 1.5.2 Алгоритм Робертса

Данный алгоритм работает в объектном пространстве, решая задачу только с выпуклыми телами в виде проволочной (каркасной) модели.

Этапы алгоритма Робертса [10, с. 250]:

- **этап 0** - подготовка исходных данных.

На данном этапе должна быть задана информация о телах. Для каждого тела сцены должна быть сформирована матрица тела  $V$ . Размерность матрицы тела –  $4 * n$ , где  $n$  – число граней тела. Каждый столбец матрицы представляет собой четыре коэффициента уравнения плоскости  $ax + by + cz + d = 0$ , проходящей через очередную грань тела.

Матрица тела представляется в следующем виде:

$$[V] = \begin{pmatrix} a_1 & a_2 & & a_n \\ b_1 & b_2 & & b_n \\ c_1 & c_2 & \dots & c_n \\ d_1 & d_2 & & d_n \end{pmatrix} \quad (1.11)$$

Матрица тела должна быть сформирована корректно (любая точка, расположенная внутри тела, должна располагаться по положительную сторону от каждой грани тела), иначе если для очередной грани условие не выполняется, соответствующий столбец матрицы тела умножается на -1.

- **этап 1** – удаление ребер, экранируемых сами телом.

На данном этапе рассматривается вектор взгляда наблюдателя  $E = \{0 \ 0 \ -1 \ 0\}$ .

Для определения невидимых граней достаточно умножить вектор  $E$  на матрицу тела  $V$ . Отрицательные компоненты полученного вектора будут соответствовать невидимым граням.

Если тело на сцене единственное, то работа алгоритма на этом заканчивается.

- **этап 2** – удаление невидимых ребер, экранируемых другими телами сцены.

На данном этапе для определения невидимых точек ребра требуется построить луч, соединяющий точку наблюдения с точкой на ребре. Точка будет невидимой, если луч на своём пути встречает в качестве преграды рассматриваемое тело.

- **этап 3** – удаление невидимых частей новых ребер, появившихся при протыкании тел.

Если тела пересекаются, т.е. протыкают друг друга, то образуются новые ребра в пересечении, их нужно проверить на экранирование самим телом (этап 1) и другими телами сцены (этап 2).

Преимущества алгоритма:

- высокая точность вычислений.

Недостатки алгоритма:

- сложность алгоритма увеличивается нелинейно –  $O(N^2)$ , где  $N$  – число объектов;
- тела сцены должны быть выпуклыми (алгоритм усложняется, так как необходима проверка на выпуклость и если тело невыпуклое, то его необходимо разбить на выпуклые тела);
- сложность реализации алгоритма.

Таким образом, Алгоритм Робертса не подходит для решения поставленной задачи из-за высокой сложности реализации алгоритма, что приведет к низкой производительности.

### **1.5.3 Алгоритм, использующий список приоритетов (алгоритм художника)**

Данный алгоритм работает аналогично тому, как художник рисует картину, то есть сцена строится, начиная с самых дальних объектов и заканчивая самыми близкими к наблюдателю.

Наиболее распространенная реализация алгоритма – сортировка по глубине ( $Z_{min}$ ), которая заключается в том, что произвольное множество граней сортируется по ближнему расстоянию от наблюдателя, причем первым в списке окажется самый дальний от наблюдателя [10, с. 329].

Затем отсортированные грани выводятся на экран в порядке от самой дальней до самой ближней.

Данный метод работает лучше для построения сцен, в которых отсутствуют пересекающиеся грани.

Преимущества алгоритма:

- достаточно эффективен по памяти (затрачивает меньше памяти, чем алгоритм, использующий Z-буфер).

Недостатки алгоритма:

- сложность реализации при пересечении граней на сцене (добавление проверок (тестов), разбиение многоугольников (граней) при циклическом экранировании);
- неэффективен по времени (из-за отрисовки невидимых граней);
- недостаточно высокая реалистичность изображения.

Таким образом, данный алгоритм не подходит для решения поставленной задачи из-за недостаточно высокой реалистичности изображения и неэффективности по времени, так как он отрисовывает все грани, включая невидимые.

### **1.5.4 Алгоритм Варнока**

Алгоритм Варнока основан на разбиении картинной плоскости на части, для каждой из которых исходная задача может быть решена достаточно просто. Алгоритм Варнока работает в пространстве изображения.

В пространстве изображения рассматривается окно и решается вопрос о том, пусто ли оно, или его содержимое достаточно просто для визуализации. Если это не так, то окно разбивается на фрагменты до тех пор, пока содержимое фрагмента не станет достаточно простым для визуализации или его размер не достигнет требуемого предела разрешения (обычно в качестве такого предела выбирают окно размером в 1 пиксель) [10, с. 291].

Сравнивая область с проекциями всех граней, можно выделить случаи, когда изображение, получающееся в рассматриваемой области, определяется сразу:

- проекция ни одной грани не попадает в область;
- проекция только одной грани содержится в области;
- проекция только одной грани пересекает область. В этом случае проекции грани разбивают всю область на две части, одна из которых соответствует этой проекции;
- существует грань, проекция которой охватывает данную область, и эта грань расположена к картинной плоскости ближе, чем все остальные грани, проекции которых пересекают данную область. В данном случае область соответствует ближайшей грани.

Если ни один из рассмотренных трех случаев не имеет места, то снова разбиваем область, например, на четыре равные части и проверяем выполнение этих условий для каждой из частей. Те части, для которых ни одно из условий не выполнилось, разбиваются снова и т. д.

Преимущества алгоритма:

- эффективен по времени, если рассматриваемая область содержит мало информации (тогда картинная плоскость будет разбита на минимальное число частей и алгоритм справится со своей задачей довольно быстро).

Недостатки алгоритма:

- алгоритм работает только в пространстве изображений;
- неэффективен по времени, если область содержит много информации (придется разбивать картинную плоскость на большое число частей, и анализировать каждую из них).

Таким образом, данный алгоритм не подходит для решения поставленной задачи, так как работает только в пространстве изображений и может быть неэффективен по времени.

### **1.5.5 Алгоритм обратной трассировки лучей**

Суть алгоритма - наблюдатель видит объект с помощью испускаемого света, который согласно законам оптики доходит до наблюдателя некоторым путем. Алгоритм называется «обратной трассировкой лучей», так как более эффективно отслеживать пути лучей не от источника к наблюдателю, а в обратном направлении, то есть от наблюдателя к объекту [10, с. 361].

Преимущества алгоритма:

- высокая реалистичность получаемого изображения;
- вычислительная сложность не сильно зависит от сложности сцены.

Недостатки алгоритма:

- неэффективность по времени из-за большого количества вычислений – низкая производительность.

Таким образом, данный алгоритм не подходит для решения поставленной задачи, так он неэффективен по времени.

### **Вывод**

Для удаления невидимых линий выбран алгоритм, использующий Z-буфер. Данный алгоритм довольно просто реализуется и достаточно эффективен по времени из-за отсутствия сортировок, также он позволяет получить

достаточно высокую реалистичность изображения, что нужно для моделирования водопада.

## 1.6 Анализ и выбор модели освещения

Модели освещения используются для вычисления интенсивности света в данной точке на поверхности модели. Все модели освещения делятся на простые (локальные) и глобальные. В простых моделях освещения учитывается свет только от источников и ориентация поверхности, а в глобальных моделях учитывают еще и свет, отраженный от других поверхностей сцены (вторичные источники).

Простая модель представляет собой сумму трех составляющих света: рассеянная, диффузная и зеркальная.

Рассеянная составляющая света (фоновое освещение) присутствует в любой точке сцены, не зависит от пространственных координат освещаемой точки и источника и описывается следующей формулой:

$$I_p = i_p * k_p, \quad (1.12)$$

где  $i_p$  – интенсивность рассеянного света (обычно задается константой для всей сцены),  $k_p$  – коэффициент диффузного отражения рассеянного света (свойство материала воспринимать фоновое освещение).

### 1.6.1 Модель Ламберта

Модель Ламберта моделирует идеальное диффузное освещение, свет при попадании на поверхность рассеивается равномерно во все стороны [11].

При такой модели освещения учитывается только ориентация поверхности ( $\vec{n}$  – вектор нормали в данной точке поверхности), направление источника света ( $\vec{S}$  – вектор падения света),  $\theta$  – угол между векторами  $\vec{n}$  и  $\vec{S}$  (рисунок 1.3).

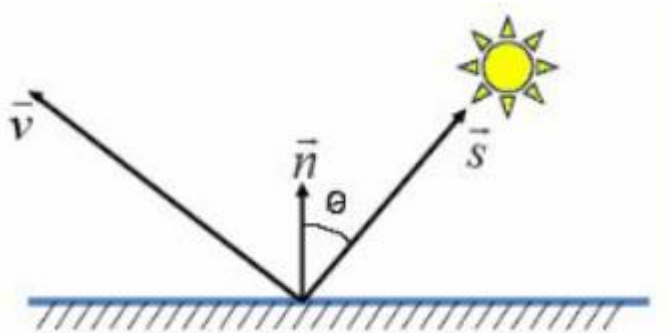


Рисунок 1.3 – модель Ламберта

Формула для расчета диффузной составляющей света:

$$I_d = k_d * i_d * \cos(\vec{S}, \vec{n}), \quad (1.13)$$

где  $k_d$  – коэффициент диффузной составляющей света,  $i_d$  – интенсивность диффузного освещения.

Эта модель является одной из самых простых моделей освещения и часто является основой для большинства других моделей. Поверхность при использовании данной модели будет выглядеть одинаково яркой со всех направлений, без бликов на телах сцены.

Преимущество данной модели: удобство для анализа свойств других моделей, за счет ее легкого выделения из любой другой модели.

### 1.6.2 Модель Фонга

Модель Фонга - классическая модель освещения, она добавляет в модель Ламберта зеркальную составляющую, то есть она представляет собой комбинацию диффузной и зеркальной составляющих. При использовании данной модели освещения кроме равномерного освещения на материале могут появляться блики. Местонахождение блика на объекте определяется из закона равенства углов падения и отражения. Чем ближе наблюдатель к углам отражения, тем выше яркость соответствующей точки [11].

Падающий луч ( $\vec{S}$ ) и отраженный луч ( $\vec{r}$ ) лежат в одной плоскости с нормалью ( $\vec{n}$ ) к отражающей поверхности в точке падения,  $\vec{v}$  - вектор, направленный на наблюдателя (рисунок 1.4). Нормаль делит угол между лучами на две равные части.

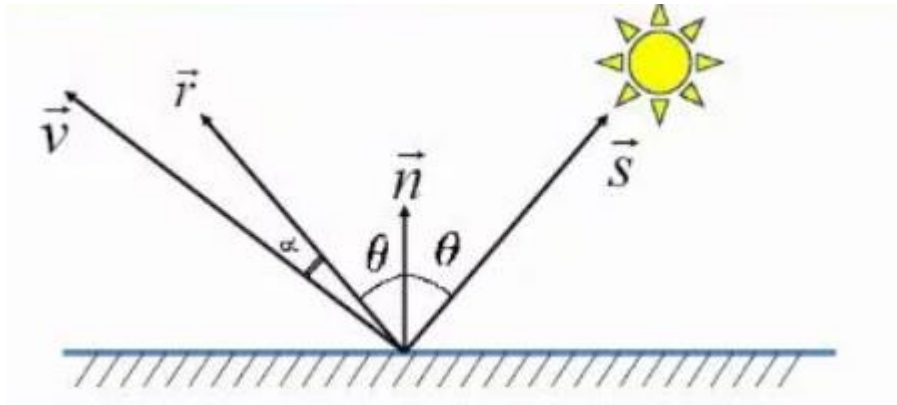


Рисунок 1.4 – модель Фонга

Формула для расчета зеркальной составляющей света в модели Фонга:

$$I_3 = k_3 * i_3 * \cos(\vec{r}, \vec{v})^p, \quad (1.14)$$

где  $k_3$  – коэффициент зеркального отражения,  $i_3$  – интенсивность зеркального освещения,  $p$  – степень аппроксимации.

## Вывод

В качестве модели освещения была выбрана модель, так как она более проста в реализации, чем модель Фонга. Также для модели Ламберта необходимо производить меньше вычислений, чем для модели Фонга, следовательно модель Ламберта более эффективна по времени, чем модель Фонга.

## 1.7 Анализ существующего программного обеспечения



В данном разделе будет рассмотрено самое распространенное существующее программное обеспечение, которое можно использовать для моделирования водопада.

### 1.7.1 Autodesk 3ds Max

Autodesk 3dsMax — профессиональное программное обеспечение для 3D-моделирования, анимации и визуализации при создании игр и проектировании [12]. В настоящее время разрабатывается и издается компанией Autodesk.

3ds Max располагает большим количеством средств для создания разнообразных по форме и сложности трёхмерных компьютерных моделей, реальных или фантастических объектов окружающего мира, с использованием разнообразных техник и механизмов.

Недостаток данного ПО – для коммерческих целей программа платная, но существует бесплатная лицензия для обучения.

Пример создания водопада в данной программе показан ниже на рисунке 1.5.

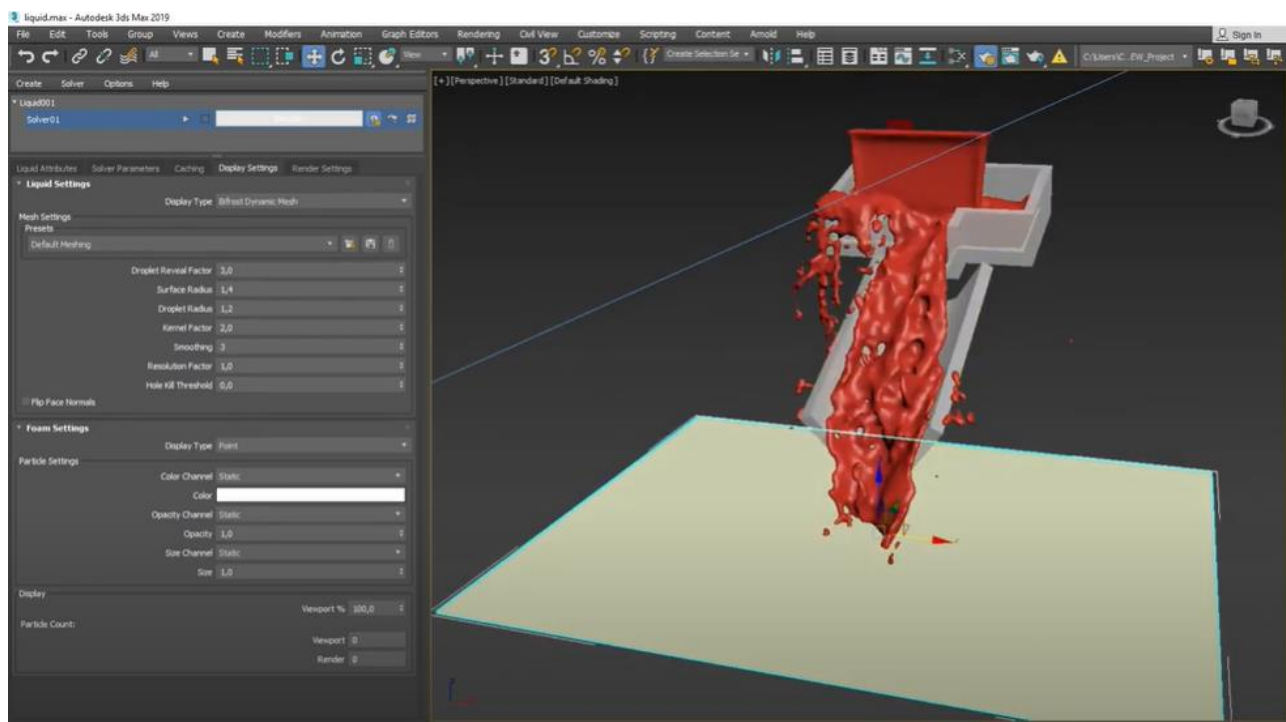


Рисунок 1.5 – пример реализации водопада в программе Autodesk 3ds Max

### 1.7.2 Blender

Blender – профессиональное свободное, кроссплатформенное и открытое программное обеспечение для создания трёхмерной компьютерной графики, включающее в себя средства моделирования, анимации, симуляции, рендеринга, монтажа видео со звуком [13].

С помощью данного ПО можно смоделировать водопад, причем он будет выглядеть очень реалистично (рисунок 1.6).

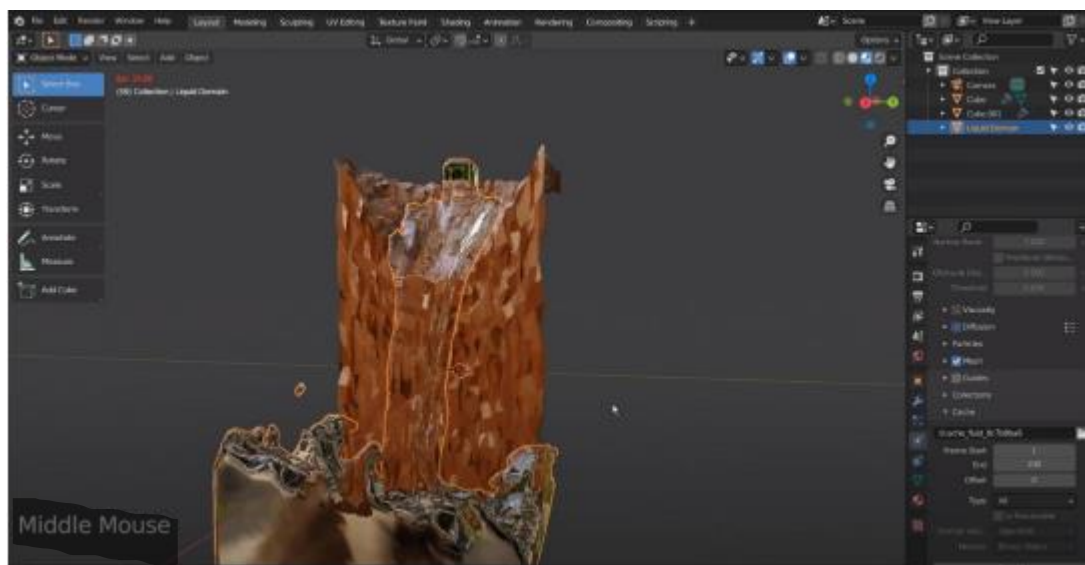


Рисунок 1.6 – пример реализации водопада в программе Blender

## 1.8 Вывод

В данном разделе были описаны и проанализированы методы визуализации текучей воды, с помощью которых можно смоделировать водопад, методы удаления невидимых линий и поверхностей, модели освещения. В качестве алгоритма визуализации водопада выбран метод, основанный на системе частиц, для удаления невидимых линий и поверхностей выбран алгоритм, использующий z-буфер, в качестве модели освещения выбрана модель Ламберта. Также в данном разделе было рассмотрено существующее ПО, которое можно применять для моделирования водопада, и была формализована модель водопада.

## 2. Конструкторский раздел

## **2.1 Метод визуализации водопада, основанный на системе частиц**

### **Алгоритм движения частиц**

В качестве метода для моделирования основного потока водопада был выбран метод, основанный на уравнениях движения частиц по криволинейной траектории - рассмотрен в аналитическом разделе.

### **Моделирование водяного облака брызг, образующегося при ударе о воду**

Так как при ударе воды о водяную поверхность образуется облако брызг, то его необходимо смоделировать для повышения реалистичности водопада. Полученная система частиц при реализации основного потока водопада позволяет довольно просто реализовать облако брызг.

При достижении поверхности водоема, частица отражается, теряет примерно 70% от своей скорости, и перекрашивается в белый цвет, чтобы отличаться от частиц основного потока водопада. Также у нее изменяется направление движения (в векторе направления движения частицы координата  $z$  умножается на  $-1$ ).

### **Моделирование водяных брызг от основного потока водопада**

С помощью системы частиц довольно просто смоделировать частицы брызг, которые отходят от водопада. Чтобы показать, что частица стала брызгом, ее скорость уменьшается на 30% от текущей и изменяются значения вектора направления (с помощью умножения на определенные коэффициенты).

Каждая частица имеет определенное время жизни. Частицы, которые прошли стадии водопада и брызг должны быть удалены.

Входные данные:

- position – массив из 3 вещественных чисел (положение частицы в трехмерном пространстве – координаты  $x$ ,  $y$ ,  $z$ );

- speed – скорость частицы;
- acceleration – ускорение частицы;
- direction – массив из 3 вещественных чисел (направления движения частицы по осям  $Ox$ ,  $Oy$ ,  $Oz$ );
- gravitation – массив из 3 вещественных чисел (направление гравитации по осям  $Ox$ ,  $Oy$ ,  $Oz$ );
- color – цвет частицы, массив из 3 целых чисел (в формате RGB);
- max\_time – максимальное время жизни частицы (при достижении максимального времени жизни частица должна исчезнуть с экрана);
- cur\_time – текущее время жизни частицы.

Выходные данные:

- position (измененное положение частицы).

Схема алгоритма представлена ниже (рисунок 2.1).

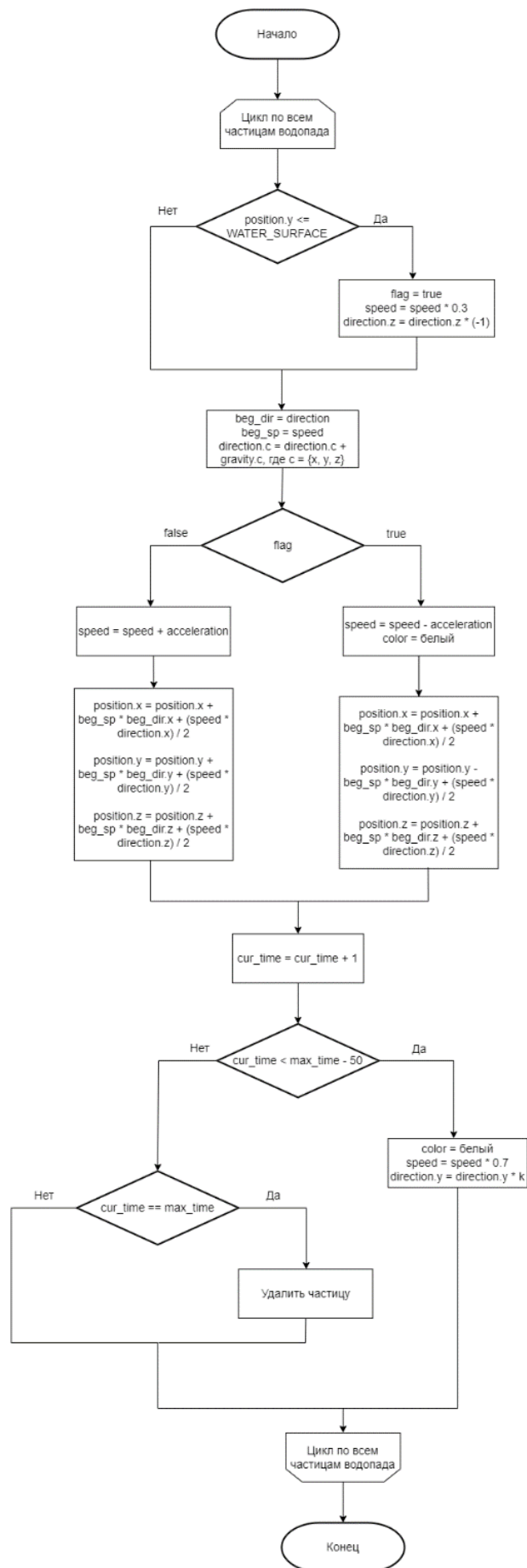


Рисунок 2.1 – схема алгоритма движения частицы водопада

## 2.2 Алгоритм удаления невидимых ребер и поверхностей, использующий Z-буфер

Z-буфер — это отдельный буфер глубины, используемый для запоминания координаты  $z$  или глубины каждого видимого пиксела в пространстве изображения (размер z-буфера равен размеру экрана приложения).

Буфер кадра используется для запоминания атрибутов (цвета/интенсивности) каждого пиксела в пространстве изображения.

Вычислять глубину каждой частицы воды не придется, так как она будет вычислена в методе визуализации водопада (`position.z`).

Для вычисления глубины скалы можно использовать уравнения плоскостей, несущих грани скалы (параллелепипеда).

$$Ax + By + Cz + D = 0 \quad (2.1)$$

Если  $C = 0$ , то многоугольник находится параллельно оси  $z$  и линии взгляда наблюдателя, в этом случае изображается видимое ребро многоугольника.

Если ребро многоугольника не вертикально, то  $z$  можно вычислить следующим образом:

$$\frac{z - z_1}{z_2 - z_1} = \frac{x - x_1}{x_2 - x_1} \quad (2.2)$$

Иначе, если ребро многоугольника не вертикально, то  $z$  можно вычислить следующим образом:

$$\frac{z - z_1}{z_2 - z_1} = \frac{y - y_1}{y_2 - y_1} \quad (2.3)$$

Иначе, произошло попадание в вершину многоугольника, глубину которой можно получить из исходных данных.

Вычисление глубины каждого пиксела на сканирующей строке ( $y=\text{const}$ ) можно проделать пошаговым способом. Грань при этом рисуется последовательно, то есть строка за строкой.

$$\Delta z = z_1 - z = -\frac{A*x_1 + B*y + D}{c} + \frac{Ax + By + D}{c} = -\frac{A}{c} * (x_1 - x) = -\frac{A}{c} * \Delta x = -\frac{A}{c} \quad (2.4)$$

В формуле (2.4)  $\Delta x = 1$  – шаг растра.

Общая схема алгоритма, использующего  $z$  – буфер представлена ниже (рисунок 2.2).

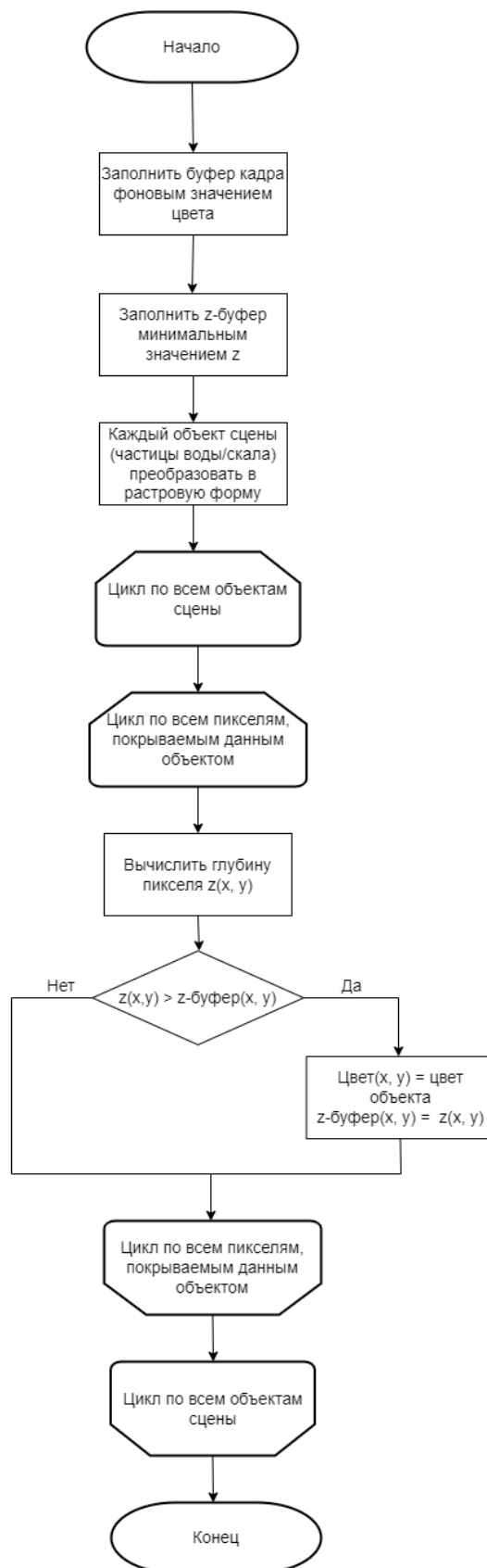


Рисунок 2.2 – схема алгоритма, использующего z – буфер

## 2.3 Модель освещения Ламберта



Данная модель вычисляет цвет поверхности в зависимости от того, как на нее светит источник света. Согласно данной модели, освещенность в точке вычисляется по следующей формуле:

$$I_d = k_d * i_d * \cos(\vec{S}, \vec{n}), \quad (2.5)$$

где  $I_d$  – диффузная (рассеянная) составляющая света,  $k_d$  – коэффициент диффузной составляющей света (способность материала воспринимать диффузное освещение),  $i_d$  – интенсивность диффузного освещения,  $\vec{S}$  – вектор падения света,  $\vec{n}$  – вектор нормали в точке поверхности.

## 2.4 Визуализация изображения скалы

Скалу удобно представить в виде массива точек, ее ограничивающих, и связей между ними.

Скала – прямоугольный параллелепипед (8 вершин). Каждая точка (вершина) задается тремя параметрами – координаты  $x$ ,  $y$ ,  $z$ .

Ширина и длина скалы задается константами. Пользователь может изменять высоту скалы.

## 2.5 Выбор используемых типов и структур данных

Для разрабатываемого программного обеспечения необходимо использовать следующие типы и структуры данных:

- точка (вершина) – массив из 3 вещественных чисел (float) – координаты  $x$ ,  $y$ ,  $z$ ;
- параметры скалы (уступа водопада):
  - высота – вещественное число (float);
  - длина – вещественное число (float);
  - ширина – вещественное число (float);

- параметры водопада (потока воды):
  - скорость водопада – вещественное число (float);
  - количество частиц – целое число (int);
- цвет – массив из 3 чисел (RGB)
- источник света - естественный, то есть расположен в бесконечности (задается максимальной интенсивностью и направлением);
- частица – объект класса Particle (положение частицы (массив координат x, y, z – float), направление движения частицы (массив из направлений по осям 0x, 0y, 0z – float), скорость частицы (float), текущее и максимальное время жизни (int), цвет частицы (в формате RGB))
- водопад (поток воды) – массив частиц;
- скала (уступ водопада) – объект класса Rock (массив точек с координатами вершин – float и массив граней - связей (по индексам точек) – int, цвет (в формате RGB), нормаль к каждой грани);

## 2.6 Структура программы

Условное разбиение реализуемых классов на группы:

- Трехмерные объекты
  - Object – абстрактный класс.
  - Rock – скала (уступ водопада), реализует работу со скалой, трехмерные преобразования и генерацию.
  - Particle – частица водопада, реализует работу с частицей, ее движение, генерацию, хранит текущее положение частицы в пространстве, направление движения, скорость, текущее время жизни, максимальное время жизни, цвет.
- Камера
  - Camera – класс для работы с камерой, хранит позицию камеры.
- Сцена
  - Scene – класс, характеризующий набор объектов и их свойства.

- Источник света
  - Light – класс, хранящий максимальную интенсивность и направление источника света.

Алгоритмы визуализации:

- Rock – отрисовка скалы.
- Particle – отрисовка частицы воды.

Интерфейс:

- Взаимодействие пользователя с интерфейсом производится через диалоговые окна приложения, взаимодействующие с классом Scene.

## 2.7 Диаграмма классов

Диаграмма классов для данного программного обеспечения представлена ниже (рисунок 2.3).

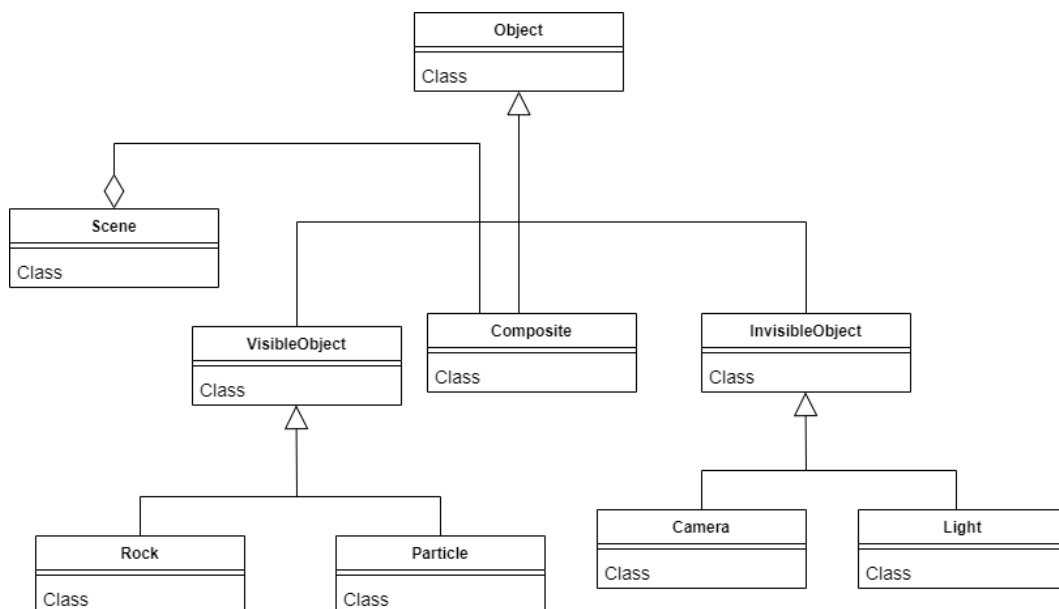


Рисунок 2.3 – диаграмма классов

## **2.8 Вывод**

В данном разделе были рассмотрены алгоритмы, которые будут использоваться для решения поставленной задачи, построены их схемы, также описаны используемые типы и структуры данных, описана структура основных классов программного обеспечения и построена их uml-диаграмма.

## **3. Технологический раздел**

### **3.1 Требования к программному обеспечению**

Программа должна предоставлять следующий функционал:

- визуальное отображение сцены (модели водопада);
- изменение параметров водопада в интерактивном режиме - высоты водопада, скорости течения воды;
- изменение количества частиц воды в водопаде;
- изменение естественного источника света за счет выбора времени суток (день/ночь);
- перемещение, масштабирование, вращение сцены;
- запуск и остановка работы модели водопада.

Требования к программе:

- Программа должна корректно реагировать на любые действия пользователя;
- Время отклика программы на действие пользователя не должно превышать 1 секунды.

### **3.2 Выбор языка программирования и среды разработки**

На данный момент существует очень большое количество языков и сред программирования, многие из которых обладают достаточно высокой эффективностью и удобны в использовании.

Для разработки данного программного обеспечения был выбран язык C++ [14]. Это обусловлено следующими факторами:

- C++ поддерживает объектно-ориентированное программирование;
- данный язык программирования является строго типизированным, что позволяет защититься от неконтролируемых ошибок;
- язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности;
- C++ имеет хорошую документацию.

В качестве среды разработки была выбрана Visual Studio 2022.

Преимущества данной среды разработки:

- высокая производительность, возможность эффективной работы над проектами любого размера и сложности;
- современная среда разработки;
- поддерживает весь основной функционал: отладчик, точки останова, сборки, клиент для Git и т.д.;
- данная среда разработки работает под Linux, macOS и Windows.

В качестве среды для разработки интерфейса был выбран Qt Designer.

Он предоставляет возможность для создания качественного интерфейса и удобен в использовании, так как имеет встроенный редактор выводимого окна приложения.

### **3.3 Интерфейс программного обеспечения**

Кнопки приложения и их функции:

- кнопка запуск/остановка – запускает или останавливает моделирование водопада, то есть течение потока воды;
- кнопки переключения день/ночь – задают текущее время суток;
- слайдер «высота водопада» – изменяет высоту водопада, чем правее слайдер, тем больше высота;

- слайдер «скорость течения воды» - изменяет скорость течения воды, чем правее слайдер, тем больше скорость;
- слайдер «количество частиц» - изменяет количество частиц воды в основном потоке воды, чем правее слайдер, тем больше частиц;
- стрелки вверх/вниз/влево/вправо и кнопки вперед/назад в блоке «камера/перемещение» - перемещают камеру;
- кнопки +/- в блоке «камера/масштабирование» - приближают/уменьшают изображение;
- стрелки вверх/вниз/влево/вправо/против часовой/по часовой – осуществляют поворот камеры.

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

Также есть возможность взаимодействовать с программой при помощи стрелок вправо/влево на клавиатуре или при помощи мышки (нажать на сцену и не отпуская кнопку плавно ввести курсор вправо/влево).

Интерфейс программы представлен ниже (рисунок 3.1).



Рисунок 3.1 – интерфейс программы

### **3.4 Вывод**

В данном разделе были рассмотрены требования к разрабатываемому программному обеспечению, был выбран язык программирования и среда разработки и был рассмотрен интерфейс приложения.

## **Заключение**

Во время выполнения производственной практики были проанализированы методы визуализации текущей воды, которые можно использовать при моделировании водопада, были рассмотрены основные алгоритмы удаления невидимых ребер и поверхностей, основные методы освещения и существующее программное обеспечение для моделирования водопадов. Также были проанализированы достоинства и недостатки рассмотренных алгоритмов и выбраны наиболее оптимальные из них для решения поставленной задачи.

Проделанная работа помогла закрепить полученные навыки в области компьютерной графики и проектирования программного обеспечения. Программное обеспечение спроектировано так, чтобы можно было достаточно легко добавлять новые объекты без изменения уже написанного кода.

## Список использованных источников

1. Куров А. В., Завалин А. А. Визуализация водной поверхности на основе спектрального метода [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/n/vizualizatsiya-vodnoy-poverhnosti-na-osnove-spektralnogo-metoda> (дата обращения 30.06.22)
2. Koptev A. V. Method of solution construction for Navier-Stokes Equations [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/n/metod-postroeniya-resheniy-uravneniy-navie-stoksa/viewer> (дата обращения 01.07.22)
3. Chentanez N., Muller M. Real-Time Eulerian Water Simulation Using a Restricted Tall Cell Grid [Электронный ресурс]. Режим доступа: <https://matthias-research.github.io/pages/publications/tallCells.pdf> (дата обращения 01.07.22)
4. Iwasaki K. Visual Simulation of Freezing Ice with Air Bubbles [Электронный ресурс]. Режим доступа: [https://www.researchgate.net/publication/262171575\\_Visual\\_simulation\\_of\\_freezing\\_ice\\_with\\_air\\_bubbles](https://www.researchgate.net/publication/262171575_Visual_simulation_of_freezing_ice_with_air_bubbles) (дата обращения 01.07.22)
5. Busaryev O. Animating Bubble Interactions in a Liquid Foam [Электронный ресурс]. Режим доступа: [https://www.researchgate.net/publication/254461110\\_Animating\\_Bubble\\_I  
nteractions\\_in\\_a\\_Liquid\\_Foam](https://www.researchgate.net/publication/254461110_Animating_Bubble_Interactions_in_a_Liquid_Foam) (дата обращения 02.07.22)
6. Foster N., Fedkiw R. Practical Animation of Liquids [Электронный ресурс]. Режим доступа: <http://physbam.stanford.edu/~fedkiw/papers/stanford2001-02.pdf> (дата обращения 02.07.22)
7. Сивухин Д. В. Общий курс физики. — М.: Физматлит, 2005. — Т. I. Механика. — С. 37. — 560 с.
8. Chentanez N., Muller M. Real-time Simulation of Large Bodies of Water with Small Scale Details [Электронный ресурс]. Режим доступа:



- <https://matthias-research.github.io/pages/publications/hfFluid.pdf> (дата обращения 02.07.22)
9. Ihmsen M. Animation of air bubbles with SPH [Электронный ресурс]. Режим доступа: [https://cg.informatik.uni-freiburg.de/publications/2011\\_GRAPP\\_airBubbles.pdf](https://cg.informatik.uni-freiburg.de/publications/2011_GRAPP_airBubbles.pdf) (дата обращения 02.07.22)
  10. Роджерс Д., Алгоритмические основы машинной графики: пер. с англ.— М.: Мир, 1989.— 512 с.: ил.
  11. Модели освещения. [Электронный ресурс]. Режим доступа: <https://devburn.ru/2015/09/> (дата обращения: 06.07.2022).
  12. Документация по Autodesk 3ds Max [Электронный ресурс]. Режим доступа: [https://ru.wikibooks.org/wiki/3ds\\_Max](https://ru.wikibooks.org/wiki/3ds_Max) (дата обращения: 07.07.2022)
  13. Документация по Blender [Электронный ресурс]. Режим доступа: <https://docs.blender.org/manual/ru/dev/> (дата обращения: 07.07.2022)
  14. Документация по C++ [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/cpp/cpp/> (дата обращения: 13.07.2022)
  15. Документация по Visual Studio [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/visualstudio/windows/> (дата обращения: 13.07.2022)
  16. Документация по Qt Designer [Электронный ресурс]. Режим доступа: <https://doc.qt.io/qt-5/qtdesigner-manual.html> (дата обращения: 13.07.2022)



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)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## *К КУРСОВОЙ РАБОТЕ*

### *НА ТЕМУ:*

*«Разработка программного обеспечения для  
моделирования водопада»*

Студент ИУ7-51Б  
(Группа)

Н.П. Артюхин  
(Подпись, дата) (И.О.Фамилия)

Руководитель курсовой работы

М.Ю. Барышникова  
(Подпись, дата) (И.О.Фамилия)

2022 г.

**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)**

---

УТВЕРЖДАЮ  
Заведующий кафедрой ИУ7  
(Индекс)  
И.В. Рудаков  
(И.О.Фамилия)  
«      »      20      г.

**З А Д А Н И Е**  
**на выполнение курсовой работы**

по дисциплине Компьютерная графика

Студент группы ИУ7-51Б

Артюхин Николай Павлович  
(Фамилия, имя, отчество)

Тема курсовой работы Разработка программного обеспечения для моделирования  
водопада

Направленность КР (учебная, исследовательская, практическая, производственная, др.)  
учебная

Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения работы: 25% к 4 нед., 50% к 7 нед., 75% к 12 нед., 100% к 14 нед.

**Задание** Разработать программное обеспечение, обеспечивающее возможность динамической визуализации модели водопада, состоящей из падающего с обрыва потока воды и брызг от него. Интерфейс программы должен позволять пользователю управлять параметрами водопада в интерактивном режиме, а именно изменять высоту водопада, скорость течения воды и количество частиц, а также изменять положение камеры и естественный источник света за счет выбора времени суток (день/ночь). Должна быть также реализована возможность перемещения, масштабирования и вращения сцены.

**Оформление курсовой работы:**

2.1. Расчетно-пояснительная записка на 25-30 листах формата А4.

Расчетно-пояснительная записка должна содержать постановку введение, аналитическую часть, конструкторскую часть, технологическую часть, экспериментально-исследовательский раздел, заключение, список литературы, приложения.

2.2. Перечень графического материала (плакаты, схемы, чертежи и т.п.). На защиту проекта должна быть представлена презентация, состоящая из 15-20 слайдов. На слайдах должны быть отражены: постановка задачи, использованные методы и алгоритмы, расчетные соотношения, структура комплекса программ, диаграмма классов, интерфейс, характеристики разработанного ПО, результаты проведенных исследований.

Дата выдачи задания «      »      20      г.

**Руководитель курсовой работы**

М. Ю. Барышникова  
(Подпись, дата) (И.О.Фамилия)

**Студент**

Н.П. Артюхин  
(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.