



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 КУРСОВОМУ ПРОЕКТУ

НА ТЕМУ:

***«Разработка программного обеспечения для
визуализации модели Нептуна и его спутников»***

Студент ИУ7-56Б
(Группа)

(Подпись, дата)

А.И.Красильникова
(И.О.Фамилия)

Руководитель курсового проекта

(Подпись, дата)

А.Ю.Быстрицкая
(И.О.Фамилия)

2023 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1. Аналитический раздел	6
1.1. Описание объектов сцены	6
1.2. Анализ способов задания трехмерных моделей	7
1.3. Описание алгоритма построения поверхностной модели для объектов сцены.....	8
1.4. Выбор алгоритма удаления невидимых ребер и поверхностей	10
1.4.1. Алгоритм обратной трассировки лучей	10
1.4.2. Алгоритм Z-буфера	10
1.4.3. Алгоритм Робертса	11
1.5. Анализ методов и алгоритмов закрашивания	12
1.5.1. Модели освещения	13
1.5.2. Простая закрашка.....	14
1.5.3. Закрашка по Гуро.....	15
1.5.4. Закрашка по Фонгу	15
2. Конструкторский раздел.....	17
2.1. Общий алгоритм решения поставленной задачи	17
2.2. Разработка алгоритмов решения поставленной задачи	18
3. Технологический раздел.....	24
3.1. Требования к ПО:	24
3.2. Средства реализации	24
3.3. Описание общей структуры ПО	25
3.4. Используемые типы и структуры данных	26
3.5. Описание интерфейса программы	27
3.6. Результаты работы ПО.....	28

3.6. Результаты работы ПО	28
4. Исследовательский раздел	30
4.1. Постановка задачи	30
4.2. Результаты исследования	31
ЗАКЛЮЧЕНИЕ	34
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	35
ПРИЛОЖЕНИЕ А Графическая часть	36

ВВЕДЕНИЕ

В современном мире стремительного прогресса в области информационных технологий и космических исследований, визуализация становится мощным инструментом для изучения труднодоступных пространств, объектов и расширения границ понимания окружающего нас мира. Планета Нептун – яркий пример такой области для исследования.

Актуальность визуализации модели Нептуна и его спутников обусловлена не только научным интересом, но и потребностью в создании эффективных инструментов для изучения космических тел. Также космическое пространство непрерывно расширяется, что также добавляет значимости этой задаче. Программное обеспечение для визуализации этих объектов поможет в привлечении новых кадров для будущих исследований и расширении границ нашего понимания космоса.

Целью курсовой работы является разработка программного обеспечения для визуализации модели Нептуна и его спутников.

Для достижения поставленной цели необходимо решить следующие задачи:

- описать структуру объектов трехмерной сцены;
- проанализировать существующие алгоритмы трехмерной графики, выбрать наиболее подходящие для визуализации модели Нептуна и его спутников;
- описать структуру программного обеспечения;
- привести схемы выбранных алгоритмов для решения задачи;
- реализовать выбранные алгоритмы;
- разработать программное обеспечение для визуализации модели Нептуна и его спутников.

1. Аналитический раздел

В этом разделе будет проведен анализ предметной области: описаны объекты синтезируемой сцены; изучены и проанализированы существующие методы и алгоритмы для решения поставленной задачи, обоснован выбор тех, которые будут использованы для визуализации сцены.

1.1. Описание объектов сцены

Сцена состоит из источника света, планеты и ее спутников:

- спутники — сферические объекты, имеющие собственную орбиту;
- Нептун - сферический объект, имеющий собственные спутники;
- источником света в данной задаче является Солнце - сферический объект (звезда), пускающий лучи света.

Ниже указаны ключевые характеристики для объектов типа спутник:

- расстояние от центра Нептуна до центра спутника в экранной СК;
- угол α_{xy} — угол между отрезком (начало - центр Нептуна, конец - центр спутника) и плоскостью, параллельной плоскости OXY и проходящей через центр Нептуна;
- угол α_{yz} — угол между отрезком (начало - центр Нептуна, конец - центр спутника) и плоскостью, параллельной плоскости OYZ и проходящей через центр Нептуна;
- цвет поверхности;
- параметр для вычисления количества узлов, использующихся для построения полигональной сетки.

Ниже указаны ключевые характеристики для объекта типа Нептун:

- координаты центра Нептуна в экранной СК;
- радиус;
- параметр для вычисления количества узлов, используемых для построения полигональной сетки.

Ниже указаны ключевые характеристики для объекта типа Солнце:

- координаты центра Солнца в экранной СК;
- радиус;
- параметр для вычисления количества узлов, используемых для построения полигональной сетки.

1.2. Анализ способов задания трехмерных моделей

Отображением формы и размеров объектов являются модели. Обычно используются три формы задания моделей [1]:

- каркасная (проволочная) модель — одна из простейших форм, совокупность вершин и ребер, определяющая форму объекта. Недостаток в том, что представление формы объекта передается не всегда точно;
- поверхностная модель — оболочка объекта, содержит информацию только о внешних геометрических параметрах. Часто встречается в компьютерной графике. Не содержит данных о материале объекта;
- объемная (твердотельная) модель — схожа с поверхностной моделью, содержит данные о материале объекта и с какой стороны поверхности он расположен (т.к. указано направление внутренней нормали).

В таблице 1.1 проведено сравнение этих способов.

Таблица 1.1 — Сравнение способов задания трехмерных объектов

Характеристика	КМ	ПМ	ТМ
Качество изображения	3	2	1
Скорость вычисления	1	2	3
Применимость к сферическим	3	1	2
Сумма оценок	7	5	6

Обозначения: КМ - каркасная модель, ПМ - поверхностная модель, ТМ - твердотельная модель. Число 1 обозначает лучший способ по определенному критерию, 3 - худший.

Для решения данной задачи будет использоваться поверхностный способ задания трехмерных моделей. Этот выбор обусловлен тем, что каркасные модели могут привести к неправильному восприятию формы объекта, а твердотельные модели затрачивают большое количество ресурсом, так как им важен материал объекта.

1.3. Описание алгоритма построения поверхностной модели для объектов сцены

Существует несколько способов представления модели [9].

Аналитический способ — описание модели объекта, доступное в неявной форме, то есть чтобы получить визуальные характеристики, необходимо также вычислять некоторую функцию, зависящую от параметра.

Полигональной сеткой — совокупность вершин, граней и ребер, задающих форму многогранного объекта. Этот способ будет использоваться для реализации поставленной задачи в силу его простоты и эффективности.

Хранение информации о полигональной сетке происходит с использованием списка граней, каждый элемент которого является в свою очередь списком вершин, образующих грань.

Рассмотрим сечение сферы плоскостью OXY , в которой лежит окружность. Необходимо изучить генерацию вершин полигональной сетки только для $\frac{1}{4}$ окружности, остальные вершины можно получить отображением благодаря симметрии [1].

Пусть $center$ - центр сферы, R - радиус, $points$ - массив вершин сетки, n - количество вершин между крайними верхней и левой вершинами окружности, i — индекс вершины (индексация происходит с левой крайней вершины) [2].

Тогда определение вершины полигональной сетки для аппроксимации $\frac{1}{4}$ окружности можно описать так:

$$\begin{aligned} temp &:= \sqrt{(2 \cdot R^2 \cdot (1 - \cos(90/(n+1) \cdot i)))} \\ dx &:= \sin(90/(n+1) \cdot i) \cdot R \\ dy &:= \sqrt{temp^2 - dx^2} \\ point_i &:= (center.x - dx, center.y + R - dy, center.z) \end{aligned}$$

Чтобы получить полную полигональную сетку, необходимо получить вершины верхней полуокружности с помощью симметрии, а после поворачивать полуокружность относительно оси OX с шагом $90/(n+1)^\circ$. В результате каждого поворота вычисляются вершины очередной полуокружности, после чего создается список граней.

1.4. Выбор алгоритма удаления невидимых ребер и поверхностей

При выборе алгоритма необходимо учесть особенности поставленной задачи. Речь идет о большом количестве сфер, заданных полигональными сетками. Поэтому основным критерий в выборе оптимального алгоритма удаления невидимых ребер и поверхностей - быстроедействие.

1.4.1. Алгоритм обратной трассировки лучей

Для определения цвета каждого пикселя проводится луч через него от наблюдателя. Траектория таких лучей отслеживается для подтверждения факта пересечения с объектами сцены. Также проверяется пересечение каждого объекта сцены с каждым лучом, а пересечение с Z представляет видимую поверхность для данного пикселя [6].

Положительной стороной алгоритма является возможность использования в параллельных вычислительных системах, так как расчет отдельной точки выполняется независимо от других.

Недостаток этого алгоритма — большое количество необходимых вычислений для синтеза изображения сцены [4].

1.4.2. Алгоритм Z-буфера

Идея алгоритма — использование двух буферов: буфера кадра, в котором хранятся цвет и интенсивность каждого пикселя, и Z-буфера с информацией о координате Z каждого пикселя [3].

Изначально в Z-буфере находятся минимальные значения Z , а в буфере кадра хранятся атрибуты пикселя с описанием фона. Каждый многоугольник преобразуется в растровую форму и записывается в буфер кадра.

В процессе подсчета глубины нового пикселя сравнивается его значение с тем, которое уже лежит в Z-буфере. Если новый пиксель находится ближе к наблюдателю, то он заносится в буфер кадра и Z-буфер корректируется [2].

Преимуществом алгоритма может являться его простота. Также в нем не тратится время на сортировку элементов сцены, что дает преимущество в скорости работы.

Недостатками алгоритма являются возникновение лестничного эффекта и сравнительно большой объем затрачиваемой памяти, однако в данной задаче он незначителен для современных компьютеров.

1.4.3. Алгоритм Робертса

Алгоритм работает в объектном пространстве, решает задачу только с выпуклыми телами.

Алгоритм выполняется в несколько этапов:

- этап подготовки исходных данных;
- удаление невидимых ребер, экранируемых самим телом;
- удаление невидимых ребер, экранируемых другими телами.

Преимуществом алгоритма является высокая точность вычислений за счет работы в объектном пространстве, что отличает его от большинства других алгоритмов.

Главным недостатком является вычислительная трудоемкость алгоритма. При большом количестве объектов сцены этот алгоритм будет недостаточно быстрым, вычислительная сложность возрастает как квадрат количества объектов.

Вывод

Таблица 1.2 — Сравнение алгоритмов удаления ребер и поверхностей

Характеристика	ТЛ	АБ	АР
Качество изображения	2	3	1
Скорость вычисления	3	1	2
Потребляемые ресурсы	1	2	3
Применимость к вращению	2	1	3
Сумма оценок	8	7	9

Обозначения: ТЛ - алгоритм обратной трассировки лучей, АБ - алгоритм Z-буфера, АР - алгоритм Робертса. Число 1 обозначает лучший способ по определенному критерию, 3 - худший.

Для решения поставленной задачи выбран алгоритм Z-буфера из-за приоритетности скорости работы алгоритма.

1.5. Анализ методов и алгоритмов закрашивания

Главный критерий в выборе оптимального алгоритма закрашки в рамках поставленной задачи — быстроедействие.

Основные типы источников света: точечные, прожекторы и бесконечно удаленные (направленные) [5].

В поставленной задаче Солнце не является ни точечным источником света, ни направленным источником света, ни прожектором, поэтому поверхность Солнца будет представляться как множество точечных источников света.

1.5.1. Модели освещения

Существует две модели освещения, используемых в синтезе трехмерных изображений: локальная и глобальная. Модель называется локальной, если она не учитывает перенос света между поверхностями. Иначе модель называют глобальной. Далее будут рассмотрены только локальные модели освещения, поскольку для частой смены кадров важна производительность.

Модель Ламберта

Модель Ламберта моделирует идеальное диффузное освещение. Свет при попадании на поверхность рассеивается равномерно во все стороны. Учитывается ориентация поверхности (N) и направление источника света (L). Этот метод крайне прост в реализации и совершенно не требователен к ресурсам.

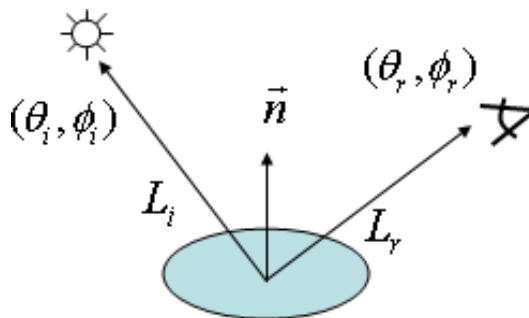


Рисунок 1.1 – Модель освещения Ламберта

Модель освещения по Фонгу

Это комбинация диффузной и зеркальной составляющих. Диффузная рассчитывается по закону Ламберта, а зеркальная придает объектам блеск. Кроме равномерного освещения на материале могут появляться блики. Их расположение на объекте определяется по закону равенства углов падения и отражения. Чем ближе наблюдатель к углам отражения, тем выше яркость соответствующей точки [7].

Падающий и отраженный лучи лежат в одной плоскости с нормалью к отражающей поверхности в точке падения (рис. 1.2). Нормаль делит угол между лучами на две равные части.

L – вектор источника света, R – вектор отраженного луча,

V – вектор на наблюдателя

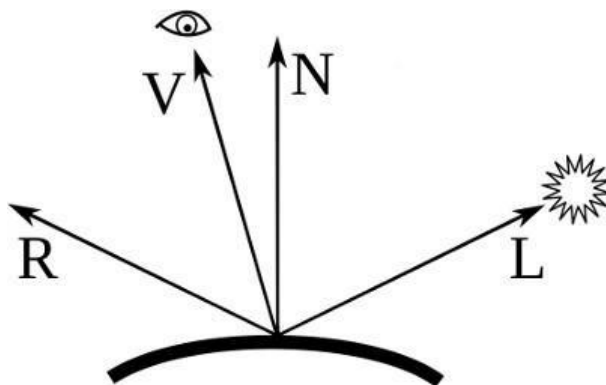


Рисунок 1.2 – Модель освещения Фонга

Выбрана модель Ламберта, так как поверхности небесных тел не обладают зеркальными эффектами, а фоновая составляющая сравнительно мала.

1.5.2. Простая закраска

Вся грань закрашивается одним уровнем интенсивности. Расчет происходит по закону Ламберта.

При использовании данного метода все плоскости будут закрашены однотонно.

Преимущество метода заключается в простоте реализации и нетребовательности к ресурсам, однако переходы между гранями модели оказываются ярко выраженными.

1.5.3. Закраска по Гуро

Идея: закрашивание каждой грани плавно переходящими оттенками, вычисление которых происходит методом интерполяции цветов примыкающих граней.

Основные этапы метода:

- для каждой грани вычисляется вектор нормали;
- для каждой вершины вычисляется нормаль как среднее между нормальями всех граней, пересекающихся в этой вершине;
- определяется интенсивность освещения в вершинах пропорционально косинусу угла между направлением света и нормалью в вершине;
- очередной многоугольник закрашивается с помощью линейной интерполяции значений интенсивностей в вершинах сначала вдоль каждого ребра, затем между ребрами вдоль каждой сканирующей строки.

Достоинство метода - высокое качество построения зеркальных бликов и матовых поверхностей, а недостатками являются относительно большие вычислительные затраты.

1.5.4. Закраска по Фонгу

Идея: основа закрашки по Фонгу – билинейная интерполяция векторов нормалей. Изображение получается реалистичным, зеркальные блики выглядят правдоподобнее, чем в методе закрашки по Гуро, однако требует больше вычислительных затрат для каждой точки поверхности за счет интерполяции значения векторов нормалей.

Вывод

В таблице 1.3 произведено сравнение алгоритмов закрашивания полигонов.

Таблица 1.3 — Сравнение алгоритмов закрашивания полигонов

Характеристика	ЗЗ	ЗГ	ЗФ
Качество изображения	3	2	1
Скорость вычисления	1	2	3
Применимость к вращениям	1	2	3
Сумма оценок	5	6	7

Обозначения: ЗЗ - простая закрашка, ЗГ - закрашка по Гуро, ЗФ - закрашка по Фонгу. Число 1 обозначает лучший способ по определенному критерию, 3 - худший.

Для поставленной задачи приоритетна скорость выполнения, поэтому выбран алгоритм простой закрашки полигонов.

Вывод

В данном разделе были описаны объекты сцены, рассмотрены алгоритмы удаления невидимых линий и поверхностей, методы закрашивания поверхностей. В качестве алгоритма удаления невидимых ребер и поверхностей был выбран Z-буфер, методом закрашки – простой.

2. Конструкторский раздел

В данном разделе описан общий алгоритм решения поставленной задачи, приведены разработанные методы и алгоритмы визуализации сцены, а также используемые типы и структуры данных.

2.1. Общий алгоритм решения поставленной задачи



Общий алгоритм:

1. определить видимость объектов относительно других объектов сцены;
2. определить интенсивность освещенности для каждого небесного тела;
3. визуализировать сцену на основе вычисленных параметров;
4. повторить пункты 1-3 в случае изменения положения сцены, перед этим применив к объектам соответствующую операцию преобразования.

2.2. Разработка алгоритмов решения поставленной задачи

Модифицированный алгоритм Z-буфера

Алгоритм Z-буфера модифицируется так, что в буфере находятся координаты z только для центров сфер.

Небесные тела не пересекаются, по координатам их центра можно однозначно определить их расположение относительно экранной плоскости. Также будут отображаться только те точки сетки сферы, глубина которых не меньше глубины ее центра.

Чтобы определить видимость ребер, аппроксимирующих окружность орбиты, используется схожий алгоритм: если координата z точек ребра окружности больше координаты z центра Нептуна, или точки конца и начала ребра окружности не лежат в окружности, лежащей в сечении Нептуна плоскостью OXY в экранной СК, то ребро окружности отображается. Иначе ребро окружности невидимо.

На рисунке 2.1 представлена схема модифицированного алгоритма Z-буфера.

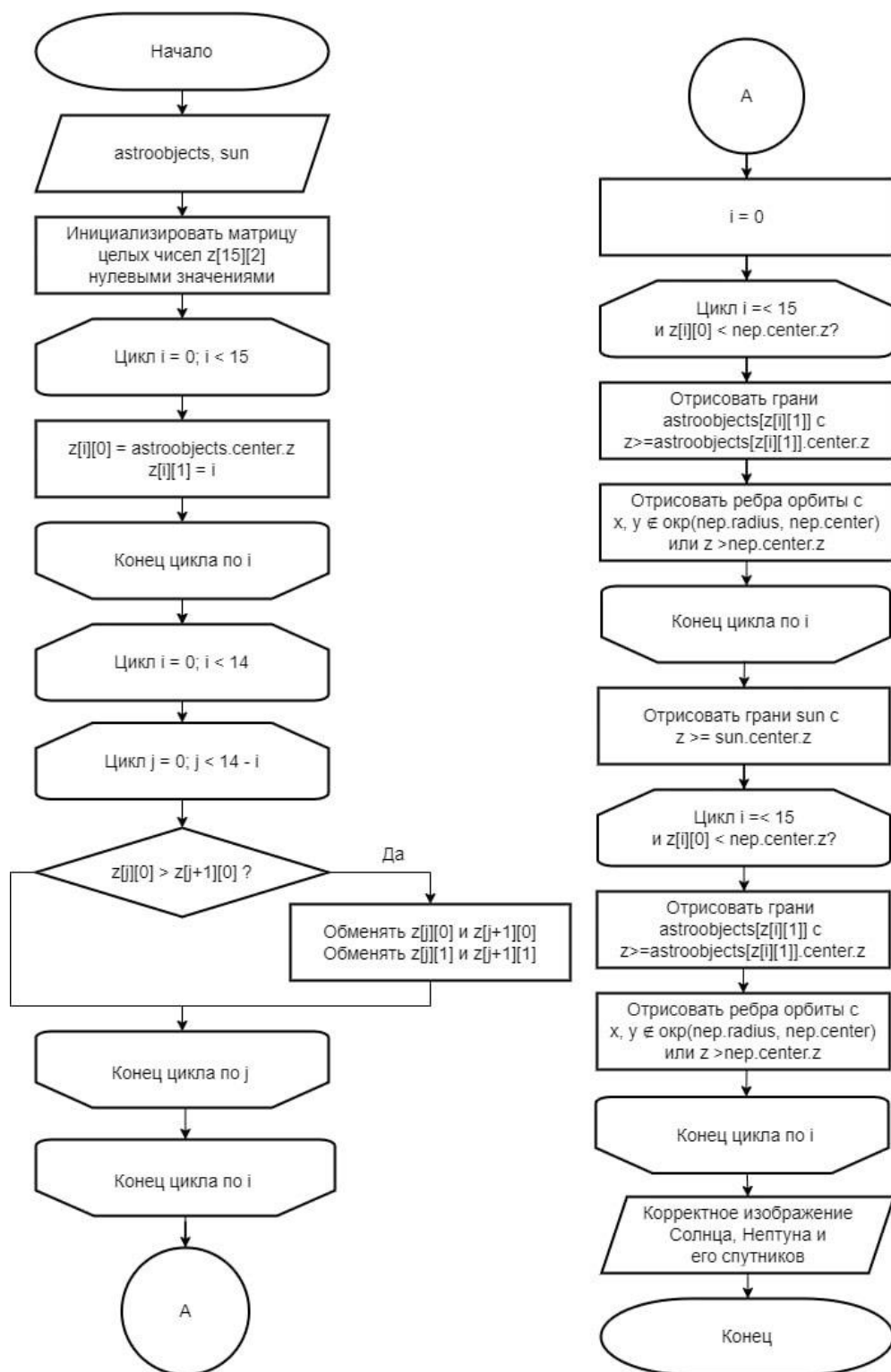


Рисунок 2.1 — Схема алгоритма Z-буфера

Алгоритм простой закрашки

Вся грань объектов закрашивается одним уровнем интенсивности. Расчет происходит по закону Ламберта.

Цвет поверхности вычисляется в зависимости от ее положения относительно источника света. Освещенность пикселя находится по формуле, в которой вычисляется произведение силы источника света и косинуса угла, под которым он светит на точку:

$$I_d = k_d \cos(L, N) i_d,$$

где:

I_d – рассеянная составляющая освещенности в точке;

k_d – свойство материала воспринимать рассеянное освещение;

i_d – мощность рассеянного освещения;

L – направление из точки на источник;

N – вектор нормали

На рисунке 2.2 представлена схема алгоритма простой закрашки.

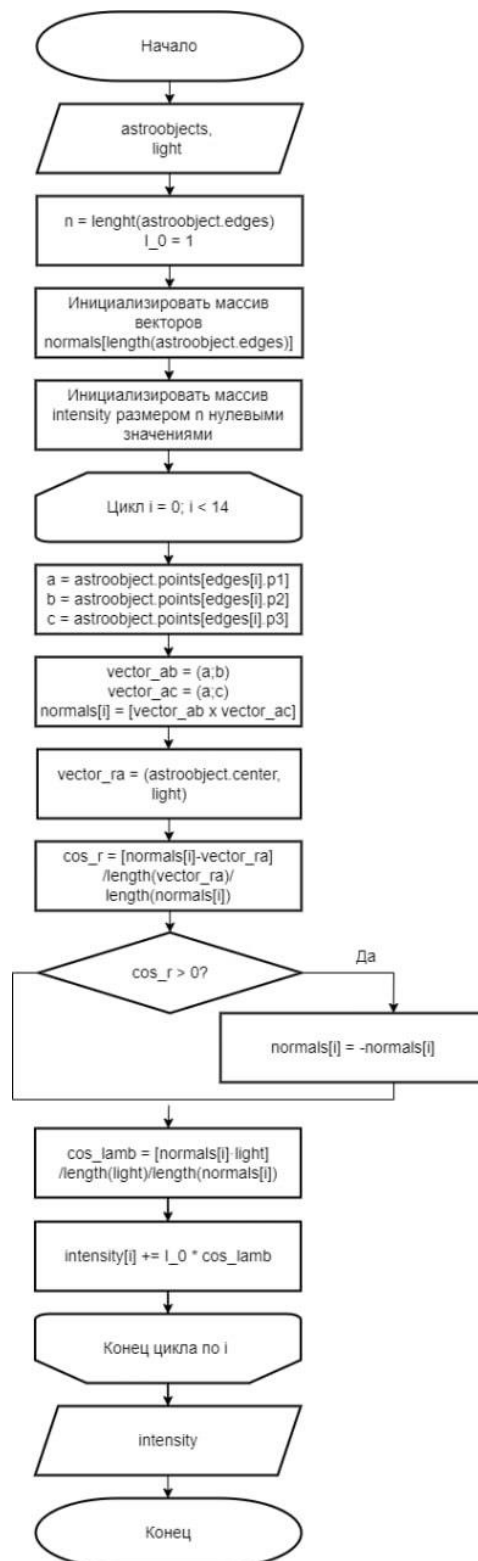


Рисунок 2.2 — Схема алгоритма простой закраски

Алгоритм построения полигональной сетки

При помощи массивов точек и ребер образуются полигоны, составляющие полигональную сетку.

На рисунке 2.3 представлена схема алгоритма построения полигональной сетки.

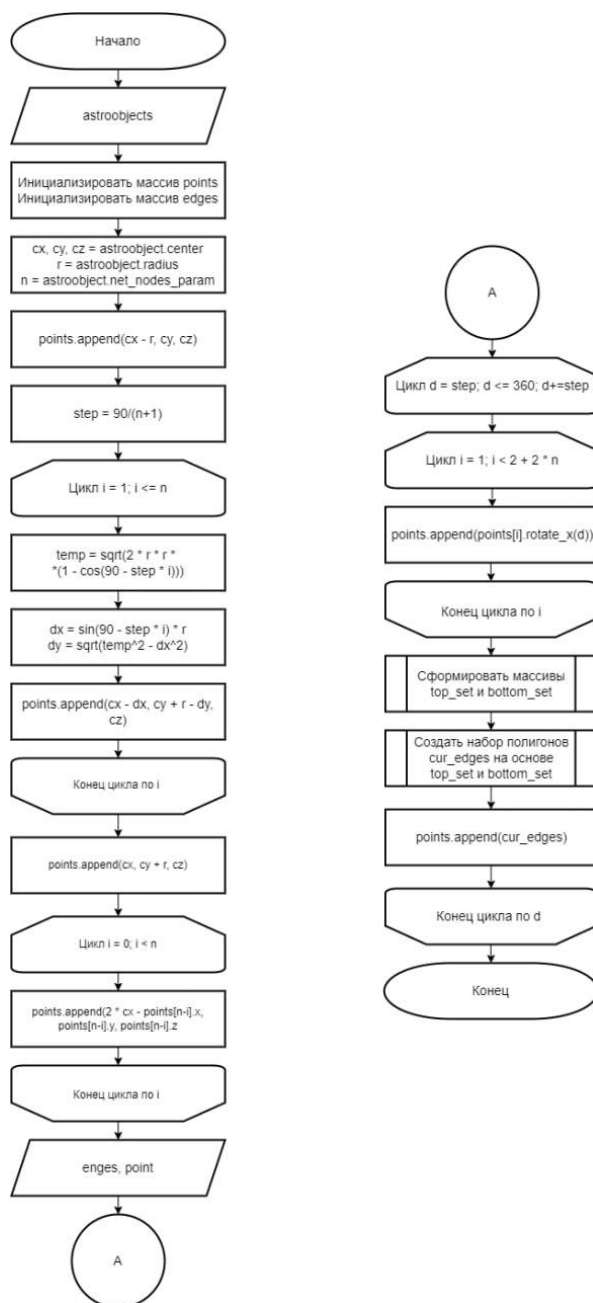


Рисунок 2.3 — Схема алгоритма построения полигональной сетки

Вывод

В этом разделе были созданы схемы реализуемых алгоритмов, определены используемые типы и структуры данных, а также описана структура ПО.

3. Технологический раздел

В данном разделе будут описаны требования к ПО, обоснован выбор языка программирования и среды обработки, приведена диаграмма основных классов, приведены средства реализации и пользовательский интерфейс, а также приведены листинги кода.

3.1. Требования к ПО:

Интерфейс программы должен позволять пользователю при помощи мыши выполнять перемещение, приближение, удаление и поворот наблюдателя (камеры).

Программа должна удовлетворять следующим требованиям:

- отклик на действия пользователя дается за комфортное для человека время;
- на любые действия пользователя программа реагирует корректно.

3.2. Средства реализации

В качестве языка программирования для реализации работы был выбран язык C++. Такой выбор сделан потому, что этот язык предоставляет весь необходимый функционал для решения поставленной задачи, а также для него существует среда разработки QT Creator со встроенным графическим редактором QT Design, позволяющим создать пользовательский интерфейс.

3.3. Описание общей структуры ПО

На рисунке 2.4 представлена общая структура ПО.

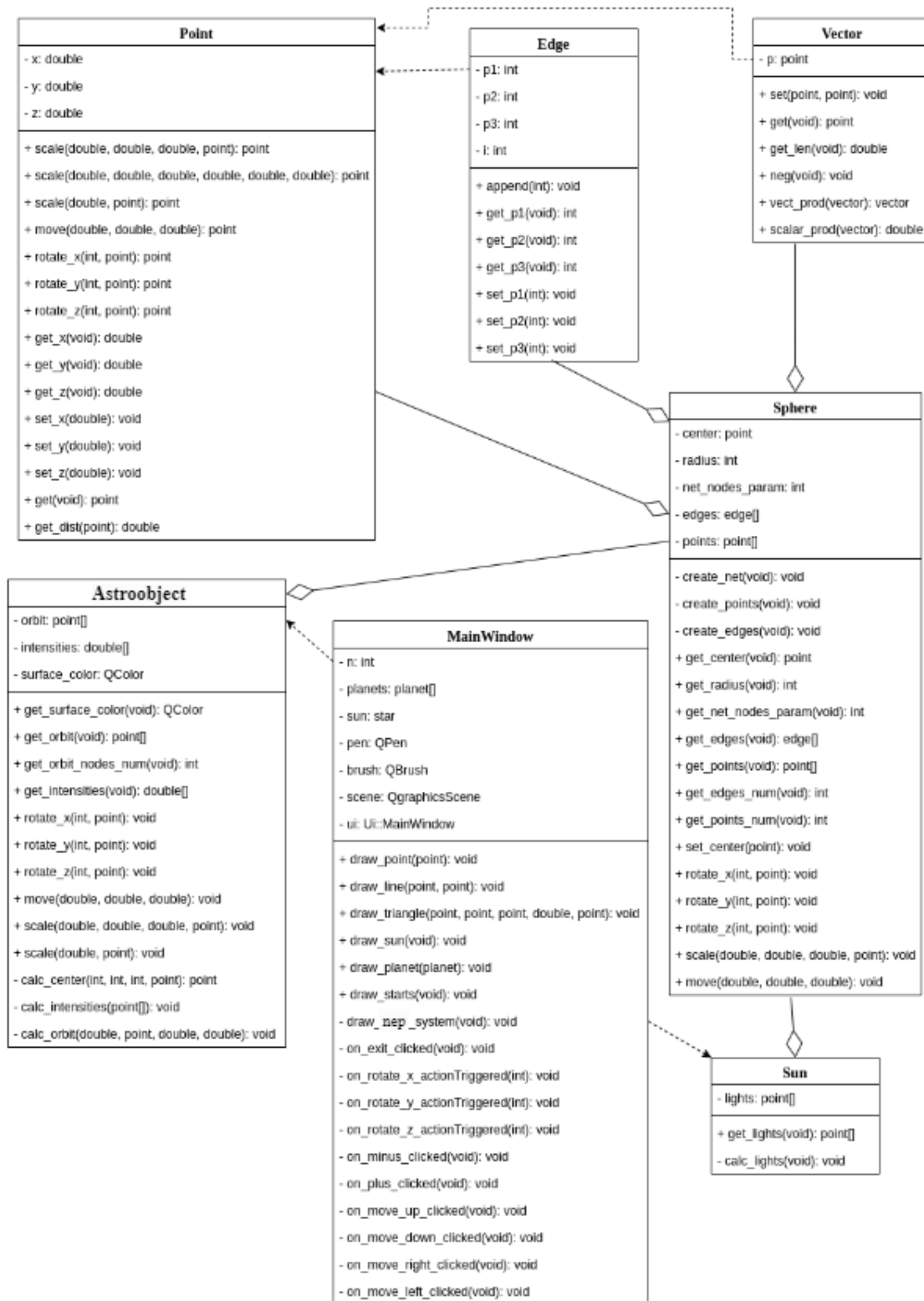


Рисунок 2.4 — Общая структура ПО

3.4. Используемые типы и структуры данных

В таблице 2.1 представлены типы и структуры данных, которые необходимо реализовать для разрабатываемого программного обеспечения.

Таблица 2.1 — Используемые типы и структуры данных

Тип данных	Структура
Точка	Координаты X, Y, Z
Вектор	Начальная и конечная точки
Грань	Индексы трёх точек, входящих в грань
Сфера	Координаты центра, радиус, параметр узлов сетки, массив узлов, массив граней
Спутник	Все параметры сферы, массив точек орбиты, цвет поверхности, список интенсивностей граней
Нептун	Все параметры сферы, цвет поверхности, список интенсивностей граней
Солнце	Все параметры сферы, список координат источников света
Сцена	Библиотечные классы

3.5. Описание интерфейса программы

Программа запускается с помощью команд `make && ./nep_system`.

На рисунке 3.1 представлен интерфейс программного продукта. Три верхние кнопки в боковом меню справа используются для поворота фигуры вокруг оси X, Y или Z (по отдельности). Повороты учитывают перспективное отображение объектов. Следующие две кнопки реализуют масштабирование относительно центра Нептуна: вперед и назад. Следующие 4 кнопки реализуют перемещение экрана. Далее можно задать цвет кодом RGB для определенного спутника по его индексу (начиная от ближайшего к Нептуну). Также можно отменить заданные пользователем цвета соответствующей кнопкой. Последняя кнопка – выход из программы.

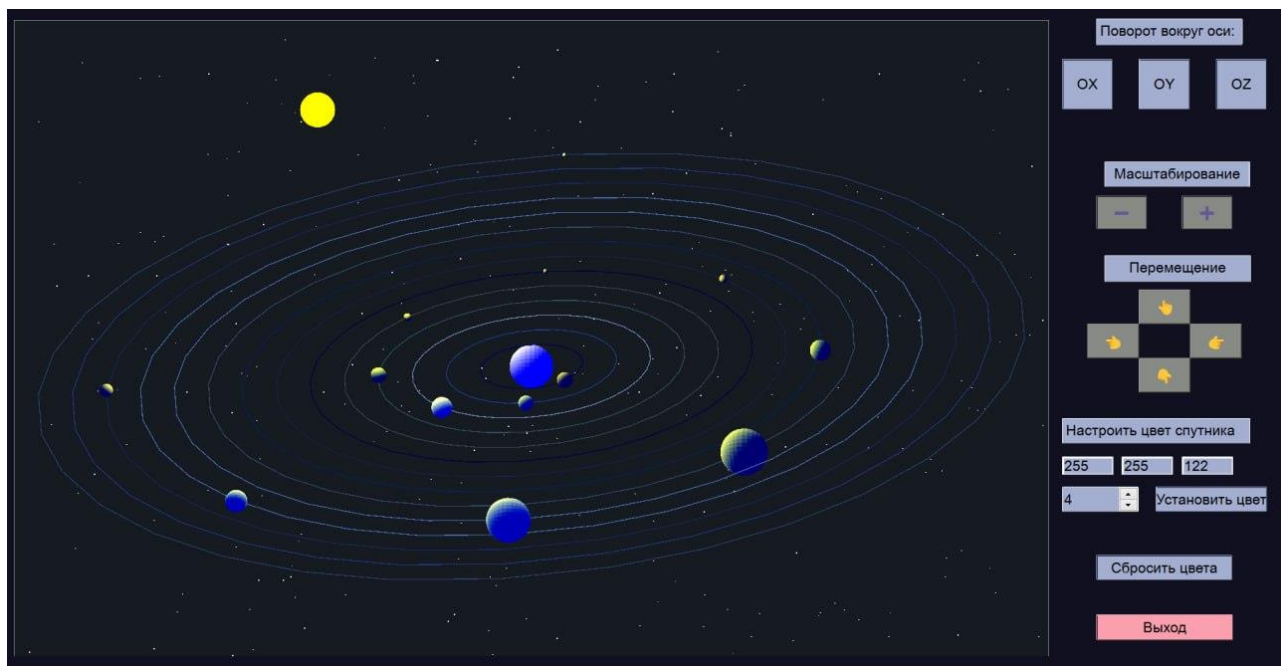


Рисунок 3.1 — Интерфейс программного продукта

3.6. Результаты работы ПО

На рисунке 3.2 приведен пример, когда девятый спутник Нептуна (Halimede) выделен розовым цветом.

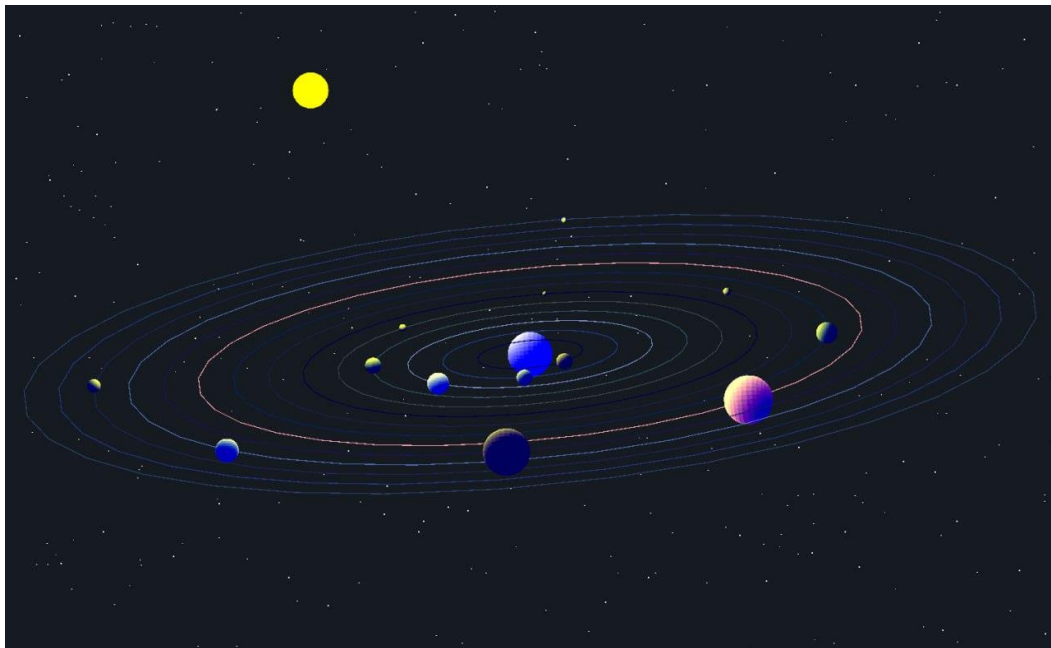


Рисунок 3.2 — Выделен девятый спутник

На рисунке 3.3 приведен пример, когда девятый спутник Нептуна (Halimede) выделен розовым цветом, а четвертый (Naiad) – желтым.

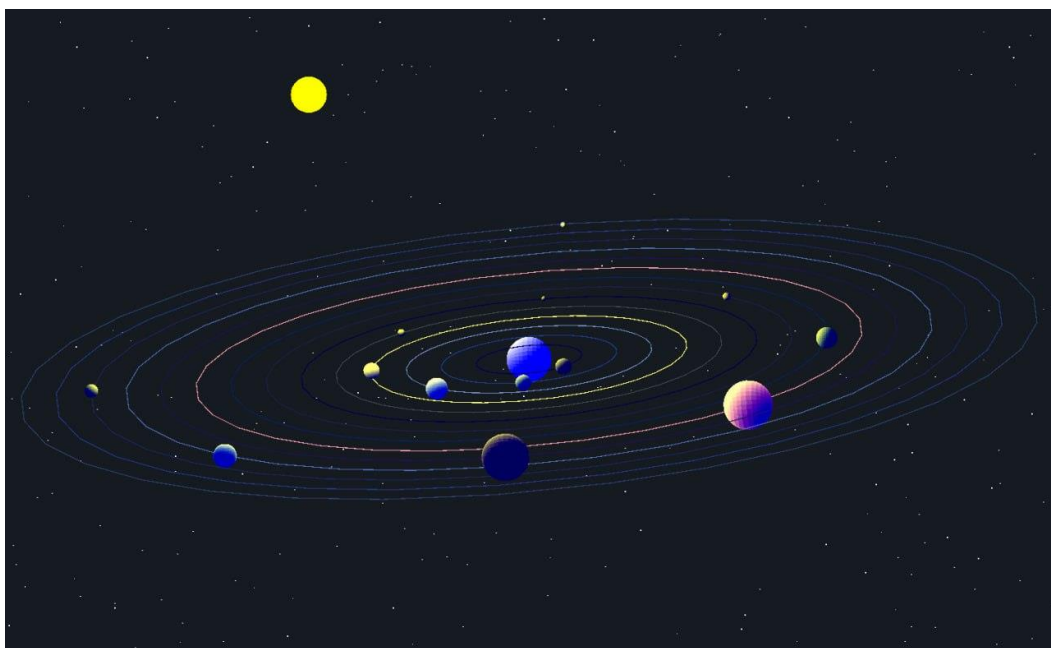


Рисунок 3.3 — Выделен девятый и четвертый спутники

Вывод

В данном разделе были рассмотрены средства реализации, описаны основные моменты программной реализации и рассмотрен интерфейс программного продукта.

4. Исследовательский раздел

В данном разделе приведено описание планирования исследований и их результаты.

Технические характеристики машины, на которой выполнялось тестирование:

- операционная система: Ubuntu Linux 20.04 64-bit [6] [7];
- оперативная память: 16 Gb;
- процессор: AMD(R) Ryzen(TM) 5 4500U CPU @ 2.3 CHz [8].

4.1. Постановка задачи

Целью исследования является определение зависимости времени генерации изображения модели Солнечной системы в зависимости от количества полигонов, аппроксимирующих поверхности объектов сцены. Количество полигонов для всех объектов является одинаковым. Исследования проводится при параметрах сетки (см 1.1.2), принимающих значение от 0 до 10.

Также необходимо определить зависимость времени регенерации изображения при осуществлении операции поворота. Для исследования рассматривается поворот вокруг оси ОУ на 5 градусов.

Во время тестирования устройство было подключено к блоку питания и не нагружено никакими приложениями, кроме встроенных приложений окружения, окружением и системой тестирования. Оптимизация компилятора была отключена.

По результатам исследования составляются сравнительные таблицы, а также строятся графики зависимостей.

4.2. Результаты исследования

В таблице 4.1 приведены усредненные результаты десяти исследований в табличном виде.

На рисунке 4. приведены усредненные результаты десяти исследования в графическом виде.

Таблица 4.1 — Зависимость процессорного времени генерации изображения и поворота объектов от кол-ва полигонов

Кол-во полигонов	Параметр сетки	Генерация, μs	Поворот, μs
10	0	2 762	2 516
54	1	10 211	7 442
105	2	26 676	19 157
238	3	56 570	41 739
414	4	114 768	83 666
550	5	203 093	151 683
806	6	327 034	280 007
990	7	613 546	470 203
1258	8	1 100 283	791 337
1558	9	1 718 542	1 310 187
1932	10	2 443 060	2 067 922

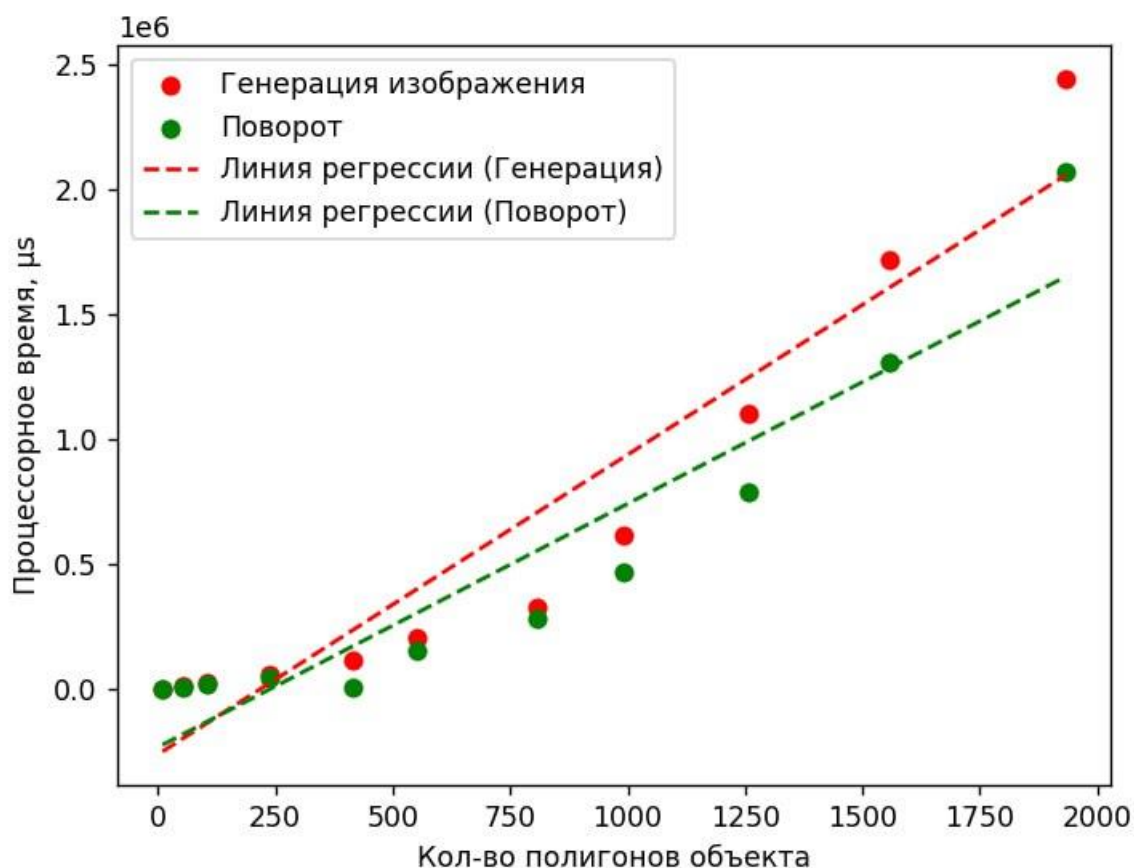


Рисунок 4.1 — Зависимость процессорного времени генерации изображения и поворота объектов от кол-ва полигонов объекта.

Вывод

По результатам исследования можно сделать вывод, что время генерации изображения и время поворота объектов сцены зависит от количества полигонов, аппроксимирующих поверхности объектов, причем зависимость стремится к квадратичной.

Также можно заметить, что время поворота объектов практически совпадает с временем генерации изображения, т. е. эта операция сравнима по трудоемкости с перерисовкой всей сцены.

На основе этих данных с целью дальнейшего улучшения продукта можно также провести анализ, какой угол поворота в зависимости от количества полигонов в системе нужно задавать для того, чтобы время поворота было комфортным для человеческого глаза, и анимация становилась более плавной.

Таким образом, чем больше полигонов, аппроксимирующих поверхности объектов сцены, тем большее требуется процессорное время работы системы, однако, чем больше полигонов, тем лучше качества изображения.

ЗАКЛЮЧЕНИЕ

При выполнении курсовой работы были решены следующие задачи:

- описаны структуры объектов трехмерной сцены;
- проанализированы существующие алгоритмы трехмерной графики, выбраны наиболее подходящие для визуализации модели Нептуна и его спутников;
- описана структура программного обеспечения;
- приведены схемы выбранных алгоритмов для решения задачи;
- реализованы выбранные алгоритмы;
- разработано программное обеспечение для визуализации модели Нептуна и его спутников.

Цель курсовой работы достигнута - было реализовано программное обеспечение для визуализации модели Нептуна и его спутников. Пользователь может масштабировать, перемещать и поворачивать модель Нептуна и его спутников.

Разработанный продукт позволяет просматривать модель Нептуна и его спутников, предоставляя возможность получить представление о ней несмотря на то, что в действительности человек не может ее увидеть.

Также над программным продуктом было проведено исследование, определяющее зависимости процессорного времени генерации изображения и поворота объектов на заданный угол от количества полигонов, аппроксимирующих поверхности объектов. Было определено, что эти зависимости близки к квадратичным.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Компьютерная графика: учебное пособие. — В.Е. Васильев, А.В. Морозов, 2005. — С. 11–27.
- [2] Computer graphics: theory and applications. — Tosiyasu L.Kunii, 1983. — С. 35–45.
- [3] Алгоритмические основы машинной графики. (Procedural Elements for Computer Graphics). — Роджерс Д. Ф., 1989. — С. 233 – 360.
- [4] Компьютерная графика: учебное пособие. — Т.О. Перемитина, 2012. — С. 79–92.
- [5] Типы источников света. — [Электронный ресурс]. Режим доступа: <https://math.hws.edu/graphicsbook/c4/s1.html>., 18.07.2023.
- [6] Ubuntu. — [Электронный ресурс]. Режим доступа: <https://nauchniestati.ru/spravka/operaczionnaya-sistema-ubuntu>., 16.09.2023.
- [7] Linux. — [Электронный ресурс]. Режим доступа: <https://progr-system.ru/wp-content/uploads/OS>., 13.09.2023.
- [8] Процессор AMD Ryzen(TM) 5. — [Электронный ресурс]. Режим доступа: <https://shop.lenovo.ru/product/81YM007FRU/>., 21.09.2023.
- [9] Лисяк В. Основы компьютерной графики: 3D-моделирование и 3D-печать. Litres, 2022.

ПРИЛОЖЕНИЕ А

Графическая часть

Презентация к курсовой работе.