министерство НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
Российской федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Тюменский государственный университет»

институт математики и компьютерных наук

Кафедра программного обеспечения

Курсовая работа по направлению

«Математическое обеспечение и администрирование

информационных систем»

на тему «Разработка приложения для расчета характеристик и визуализации сложных физических систем в волновой оптике»

Выполнил: студент 2 курса

22 МОАИС 184-2 группы

Юсуф М.А.

Научный руководитель:

доцент кафедры ПО

Гаврилова Н.М.

Тюмень 2020

Оглавление

[ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ 4](#_Toc55743119)

[1.1 Общие сведения о модели распространения лучей и основные определения 4](#_Toc55743120)

[1.2 Геометрическая оптика 5](#_Toc55743121)

[1.3 Волновая оптика. Используемые законы. 7](#_Toc55743122)

[ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ 8](#_Toc55743123)

[2.1Используемые технологии 8](#_Toc55743124)

[2.2 Реализованные классы и методы 8](#_Toc55743125)

[2.3 Математическая модель 18](#_Toc55743126)

[2.4 Алгоритм отрисовки 19](#_Toc55743127)

[ГЛАВА 3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 19](#_Toc55743128)

[Заключение 20](#_Toc55743129)

[Список литературы 21](#_Toc55743130)

[Приложение 1 21](#_Toc55743131)

**ВВЕДЕНИЕ**

Курсовая

В последнее время всё больше компаний в игровой индустрии начинает использовать технологии Ray-tracing, эти технологии отображают реалистичное распространение лучей в пространстве и из-за этого передаваемое изображение становится более реалистичным и приятным для пользователя. Но эта технология не описывает реальную картину распространения лучей света, она строит ход лучей, приближенный к реальности.

Цель работы: разработать приложение для визуализации и моделирования распространения лучей света в пространстве.

Для достижения цели необходимо выполнить следующие задачи:

1. Изучить необходимые разделы по оптике(физика)
2. Разработать математическую модель;
3. Реализовать приложение-визуализатор;
4. Провести вычислительный эксперимент;

Объектом исследования являются лучи света и их распространение в различных средах.

# ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

# 1.1 Общие сведения о модели распространения лучей и основные определения

На данный момент физиками было подсчитана и сформулирована модель распространения лучей. Её начинают изучать школьники в 7 классе в качестве ознакомления и продолжают углублённо в 10 – 11 классах. Оптика разделяется на 3 раздела: геометрическая, волновая и квантовая. В дальнейшем в курсовой работе будут использоваться следующие определения:

**Оптика** - раздел физики, рассматривающий явления, связанные с распространением электромагнитных волн видимого, инфракрасного и ультрафиолетового диапазонов спектра. Оптика описывает свойства света и объясняет связанные с ним явления.

**Геометрическая оптика** (оптика луча) - раздел оптики, который не занимается рассмотрением вопроса о природе света, а основывается лишь на эмпирических законах его распространения. Центральное понятие геометрической оптики, с помощью которого описывается распространение света, — световой луч

**Волновая оптика** – раздел оптики, рассматривающий свет как электромагнитные волны. Явления интерференции и дифракции света служат опытным подтверждением его волновой природы.

**Квантовая оптика** – раздел оптики, изучающий взаимодействие света с веществом, при котором проявляются корпускулярные свойства вещества.

**Световой луч** – траектория, вдоль которой распространяется свет.

**Источник света** – тело, которое излучает световые лучи, имеющее пренебрежительно малый размер.

**Свет** - это электромагнитные волны, длины волн которых лежат для среднего глаза человека в пределах от 400 до 760 нм. В этих пределах свет называется **видимым**.

**Относительный показатель преломления** (относительный коэффициент преломления) – отношение показателя преломления света второй среды к показателю первой, обозначается как n.

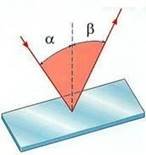
# 1.2 Геометрическая оптика

Основные законы геометрической оптики были известны задолго до установления физической природы света.

**Закон прямолинейного распространения света**: в оптически однородной среде свет распространяется прямолинейно. Опытным доказательством этого закона могут служить резкие тени, отбрасываемые непрозрачными телами при освещении светом источника достаточно малых размеров («точечный источник»). Другим доказательством может служить известный опыт по прохождению света далекого источника сквозь небольшое отверстие, в результате чего образуется узкий световой пучок. Этот опыт приводит к представлению о световом луче как о геометрической линии, вдоль которой распространяется свет. Таким образом, геометрическая оптика, опирающаяся на представление о световых лучах, есть предельный случай волновой оптики при → 0. Границы применимости геометрической оптики будут рассмотрены в разделе о дифракции света.

На границе раздела двух прозрачных сред свет может частично отразиться так, что часть световой энергии будет распространяться после отражения по новому направлению, а часть пройдет через границу и продолжит распространяться во второй среде.

**Закон независимости световых лучей**: Распространение световых лучей в среде происходит независимо друг от друга.

**Закон отражения света**: Падающий и отраженный лучи, а также перпендикуляр к границе раздела двух сред, восстановленный в точке падения луча, лежат в одной плоскости (плоскость падения). Угол отражения 𝛽 равен углу падения 𝛼 (рис. 1.2.1):

(рис. 1.2.1: отражение света)

**Закон преломления света***:*Падающий и преломленный лучи, а также перпендикуляр к границе раздела двух сред, восстановленный в точке падения луча, лежат в одной плоскости. Отношение синуса угла падения *𝛼* к синусу угла преломления *𝛽* есть величина, постоянная для двух данных сред https://mega-talant.com/uploads/files/301962/82154/87201_html/images/82154003.png(рис. 1.2.2):

(рис. 1.2.2: закон преломления света)

Где n – отношение коэффициента преломления второй среды, к первой. n = n2/n1.

Закон преломления был экспериментально установлен голландским ученым В. Снеллиусом в 1621г.

Законы отражения и преломления находят объяснение в волновой физике. Согласно волновым представлениям, преломление является следствием изменения скорости распространения волн при переходе из одной среды в другую.

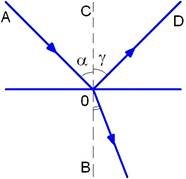
**Физический смысл показателя преломления** – отношение скорости распространения волн в первой среде 𝑣1 к скорости их распространения во второй среде 𝑣2:

https://mega-talant.com/uploads/files/301962/82154/87201_html/images/82154005.png

Абсолютный показатель преломления равен отношению скорости света 𝑐 в вакууме к скорости света 𝑣 в среде:

https://mega-talant.com/uploads/files/301962/82154/87201_html/images/82154006.png

Согласно уравнениям, 𝑛21 показывает, во сколько раз меняется скорость света при его переходе из одной среды в другую, a 𝑛 – при переходе из вакуума в среду. В этом заключается физический смысл показателей преломления.



(Рис.1.2.3. Законы отражения и преломления: 𝛾 = 𝛼; 𝑛1𝑠𝑖𝑛𝛼 = 𝑛2𝑠𝑖𝑛𝛽.)

# 1.3 Волновая оптика. Используемые законы.

**Волновая оптика** – раздел оптики, объясняющий оптические явления на основе волновой природы света.

Волновая оптика описывает такие оптические явления, как интерференция, дифракция, поляризация, дисперсия. В курсовой работе этот раздел оптики будет использоваться только для подсчёта количества энергии светового пучка.

**Зако́н Бугера — Ламберта — Бера** (также просто закон Бугера) — физический закон, определяющий ослабление параллельного монохроматического пучка света при распространении его в поглощающей среде. Закон выражается следующей формулой (рис. 1.3.1):

(рис. 1.3.1: закон Бугера)

где I — интенсивность света, прошедшего слой вещества толщиной I0 — интенсивность света на входе в вещество, k λ — показатель поглощения

**Коэффициент отражения** (ρ) – количество отраженного света, по сравнению с количеством преломлённого и поглощенного света. Выражается следующей формулой (рис. 1.3.2):

(рис. 1.3.2: коэффициент отражения границы двух сред)

# ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

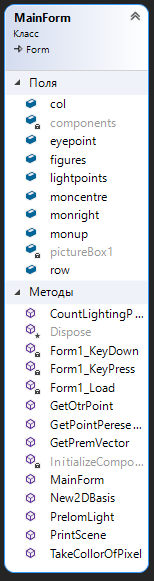
# 2.1Используемые технологии

Программа написана в среде разработки Microsoft Visual Studio на языке программирования C#, для операционной системы Windows с использованием интерфейса программирования приложений WindowsForms.

# 2.2 Реализованные классы и методы

В ходе написания приложения были реализованы следующие классы: MainForm, Figure, box, Sphere, Point, Flat, Vector, LightPoint, PixelColor, InfoAboutLightsAtPoint, PointAndFigure

Класс MainForm представляет собой основной класс, в нём происходят все основные вычисления и обрисовывается изображение (диаграмма 1):



(Диаграмма 1.Класс MainForm)

Для класса MainForm были реализованы поля и методы, представленные в таблице 1.

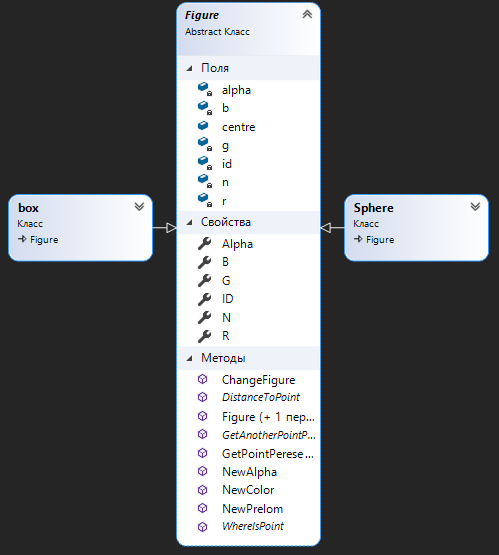
|  |  |  |
| --- | --- | --- |
| Методы класса MainForm | | |
| Form1\_Load(object sender, EventArgs e) | | Инициализирует объекты(фигуры, источники света), отрисовывает изображение на экран. |
| Form1\_KeyDown(object sender, KeyEventArgs e) | | Производит перемещение/вращение камеры в пространстве. |
| PrintScene() | | Метод, который отрисовывает изображение на экран |
| PointAndFigure GetPointPeresechFigur(Point A, Point B, List<Figure> figures) | | Возвращает точку и фигуру пересечения прямой между точками A и B и какой-либо фигурой из списка figures |
| Point New2DBasis(Vector v1, Vector v2, Point p3) | | Возвращает координаты точки p3 в базисе векторов v1 v2 |
| Point GetOtrPoint(Point p1, Point p2, Flat flat) | | Возвращает точку, лежащую на отраженном луче от плоскости flat |
| Vector GetPremVector(Point p1, Point p2, Flat flat, double N1 , double N2) | | Возвращает вектор, полученый путём преломления вектора, лежащего между точками p1p2, плоскостью flat и коэфицентами преломления сред N1 N2 |
| InfoAboutLightsAtPoint PrelomLight(Point p,Point p1, Figure f, Vector v, List<Figure> oldfigures , List<LightPoint> oldlights) | | Возвращает информацию о свете, попадающего в точку p, через преломлённый луч v из точки p1. |
| PixelColor TakeCollorOfPixel(int numcol, int numrow, List<LightPoint> oldlights, List<Figure> oldfigures, Point moncentre, Point monup, Point monright, Point eyepoint) | | Метод, возвращающий цвет пикселя на мониторе, где: numcol – номер столбца пикселя на мониторе, numrow – номер строки пикселя на мониторе, moncentre? Monup, monleft – координаты плоскости монитора в пространстве. |
| Поля класса MainForm | | |
| Point moncentre | Точка центра монитора | |
| Point monup | Точка монитора сверху посередине | |
| Point monright | Точка монитора справа посередине | |
| List<Figure> figures | Список всех фигур | |
| List<LightPoint> lightpoints | Список всех источников света | |
| Point eyepoint | Точка глаз наблюдателя | |
| const int row = 600 | Количество строк пикселей в форме | |
| const int col = 800 | Количество столбцов пикселей в форме | |

(Таблица 1. Класс MainForm)

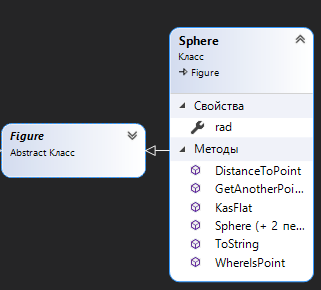
Класс Figure является абстрактным классом. Он родитель для классов фигур(box, Sphere).диаграмма 2.

Класс box представляет собой описание объекта – коробка. Этот объект представляет собой коробку с указанными размерами граней и направленную направляющими перпендикулярными векторами. Диаграмма 3.

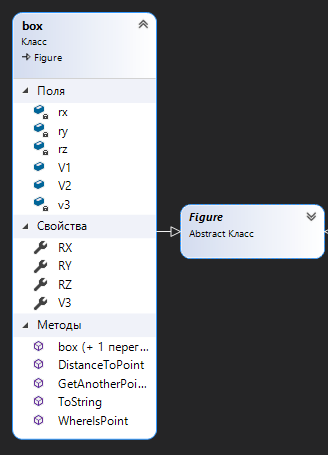
Класс Sphere представляет собой описание объекта – шар. Он имеет только центр фигуры и радиус. Диаграмма 4.



(Диаграмма 2.Класс Figure)



(Диаграмма 4.Класс Sphere)



(Диаграмма 3. Класс box)

|  |  |
| --- | --- |
| Свойства класса Figure | |
| double Alpha | Коэффициент поглощения материала, из которого сделан объект. Открыт для чтения, закрыт для записи. |
| double N | Коэффициент преломления материала, из которого состоит объект. Открыт для чтения, закрыт для записи. |
| byte R | Красная составляющая объекта. Открыт для чтения, закрыт для записи. |
| byte G | Зелёная составляющая объекта. Открыт для чтения, закрыт для записи. |
| byte B | Синяя составляющая объекта. Открыт для чтения, закрыт для записи. |
| int ID | Численный идентификатор объекта. Открыт для чтения, закрыт для записи. |
| Конструкторы класса Figure | |
| Figure() | Конструктор по умолчанию. Задаёт все свойства по нулям, кроме N и Alpha. N устанавливает на 10000, Alpha на 10^7 |

|  |  |
| --- | --- |
| Figure(Point Centre, int id) | Конструктор, устанавливающий точку центра и идентификатор фигуры. |
| Методы класса Figure | |
| void ChangeFigure(byte r, byte g, byte b, Point centre,double n, double alpha) | Устанавливает свойства на указанные. |
| void NewColor(byte r, byte g, byte b) | Устанавливает новый цвет |
| void NewAlpha(double alpha) | Устанавливает новый коэффициент поглощения |
| void NewPrelom(double n) | Устанавливает новое преломление |
| abstract Point GetAnotherPointPrelom(Point p, Vector v) | Абстрактный метод, возвращающий вторую точку пересечения прямой с указанным вектором v через точку p и стенками фигуры. |
| abstract double DistanceToPoint(Point point) | Абстрактный метод, возвращающий расстояние от точки до фигуры. |
| abstract Flat WhereIsPoint(Point p) | Абстрактный метод, возвращающий плоскость фигуры, на которой лежи точка. |

(Таблица 2. Класс Figure)

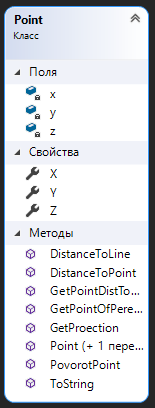
|  |  |
| --- | --- |
| Свойства класса box: Figure | |
| double RX | Расстояние от центра фигуры до её стороны по направляющему вектору V1. |
| double RY | Расстояние от центра фигуры до её стороны по направляющему вектору V2. |
| double RZ | Расстояние от центра фигуры до её стороны по направляющему вектору V3. |
| Vector V1 | Направляющий вектор V1 |
| Vector V2 | Направляющий вектор V2 |
| Vector V3 | Направляющий вектор V3 |
| Конструкторы класса box | |
| box(double rx, double ry, double rz, Vector v1, Vector v2) | Конструктор, в котором объявляются направляющие векторы и размеры коробки. |
| box(double rx, double ry, double rz, Vector v1, Vector v2, Point Centre, int id) : base(Centre, id) | Конструктор, в котором помимо свойств объекта задаётся её центр и идентификатор. |

(Таблица 3. Класс box)

|  |  |
| --- | --- |
| Свойства класса Sphere | |
| double rad | Радиус шара. Открыт для чтения, закрыт для записи. |
| Конструкторы класса Sphere | |
| Sphere(Point centre, double radius, int ID) : base(centre, ID) | Конструтокр, в котором можно задать центр шара, его радиус, идентификатор. |
| Sphere(Sphere sphere) | Конструктор, предназначенный для клонирования объекта |
| Методы класса Sphere | |
| Flat KasFlat(Point p) | Возвращает плоскость, касательную к сфере в точке |

(Таблица 4. Класс Sphere)

Класс Point реализует объект – точка. Является неотъемлемым классом программы. (Диаграмма 5):



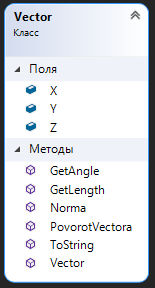
(Диаграмма 5.Класс Point)

Для класса Point были реализованы свойства, поля и методы, представленные в таблице 5

|  |  |
| --- | --- |
| Свойства класса Point | |
| Int X | X составляющая точки. |
| Int Y | Y составляющая точки. |
| Int Z | Z составляющая точки. |
| Конструкторы класса Point | |
| Point() | Конструктор по умолчанию. X,Y,Z = 0. |
| Point(double X, double Y, double Z) | Конструктор, задающий координаты точки. |
| Методы класса Point | |
| Point GetProection(Flat flat, Point point) | Метод, возвращающий проекцию точки point на плоскость flat |
| Point GetPointOfPeresech(Flat flat, Vector v, Point M) | Возвращает точку пересечения прямой из точки M с направляющим вектором v и плоскости flat. |
| Point PovorotPoint(double ox, double oy, double oz) | Поворачивает точку в пространстве на угол ox, oy, oz вокруг осей ox, oy, oz. |
| Point GetPointDistToPoint(double distance, Vector v) | Возвращает новую точку, отложенную на прямой от данной точки с направляющим вектором v на дистанции от данной – distance. |
| double DistanceToPoint(Point point) | Возвращает расстояние от одной точки до другой |
| double DistanceToLine(Point p1, Point p2) | Возвращает расстояние от обращаемой точки до прямой, образованной точками p1, p2. |

(Таблица 5. Класс Point)

Класс Vector является представлением объекта – вектор. (Диаграмма 6)



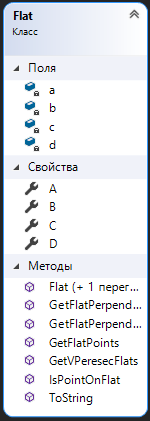
(Диаграмма 6. Класс Vector)

Для класса Vector были реализованы следующие поля, конструкторы и методы:

|  |  |
| --- | --- |
| Поля класса Vector | |
| Int X | X составляющая вектора. |
| Int Y | Y составляющая вектора. |
| Int Z | Z составляющая вектора. |
| Методы класса Vector | |
| double GetLength() | Вернуть длину вектора. |
| Double GetAngle(Vector v) | Вернуть угол между двумя векторами. |
| Void Norma() | Отнормировать вектор. |
| Vector PovorotVectora(double ox, double oy, double oz) | Повернуть вектор на угол ox, oy, oz вокруг осей ox, oy, oz. |

(Таблица 6. Класс Vector)

Класс Flat представляет собой объект - плоскость (диаграмма 7).



(Диаграмма 7. Класс Flat)

Для класса Flat были разработаны следующие конструкторы, поля, свойства, методы:

|  |  |
| --- | --- |
| Конструктор класса Flat | |
| Flat() | Конструктор по умолчанию. A,B,C = 1, D=0 |
| Flat(double A, double B, double C, double D) | Конструктор, в котором можно задать коэффициенты A,B,C,D |
| Методы класса Flat | |
| Flat GetFlatPoints(Point p0, Point p1, Point p2) | Метод, возвращающий плоскость, образованную тремя точками. |
| Vector GetVPeresecFlats(Flat f) | Метод, возвращающий вектор, пересечения двух плоскостей. |
| Flat GetFlatPerpendLine(Point p0, Point p1) | Возвращает плоскость, перпендикулярную прямой через точку p0. |
| bool IsPointOnFlat(Point p) | Проверка на принадлежность точки плоскости. |

(Таблица7. Класс Flat)

Остальные классы являются простыми классами для хранения информации об объектах, либо хранения других объектов.

# 2.3 Математическая модель

Для получения цвета отдельного пикселя на мониторе производится расчет координат этого пикселя в пространстве, строится прямая через точку глаз наблюдателя и точку пикселя. Полученная прямая рассматривается как луч света. Производится проверка на пересечение луча с фигурами при помощи соответствующего алгоритма.

Если луч из точки глаз через точку пикселя не пересекается ни с одной фигурой, то считается что цвет пикселя, через который был построен луч, примет значения (125, 125, 125), что является серым цветом.

Если же луч пересекается с какой-либо фигурой, проверяется прозрачность фигуры:

Фигура полупрозрачна – строится дополнительный преломлённый луч, для него высчитывается точка выхода из фигуры, строится новый преломлённый выходной луч, производится проверка на пересечении с остальными фигурами и, если пересекается – то берётся цвет и сила излучения точки пересечения, если нет, то берётся серый цвет с силой излучения 1. Для получения конечного цвета в изначальной точке используя закон Бугера рассчитывается конечная яркость, берётся среднее арифметическое между цветом самого объекта и цветом точки пересечения выходного луча, и умножается на силу излучения.

Фигура непрозрачна – пропускается предыдущий шаг.

Строится отраженный луч по соответствующему алгоритму. Проверяется его пересечение с другими фигурами. Если не пересекается – передаётся серый цвет и сила излучения, равная 0. Иначе – проверяется цвет точки пересечения, сила света в этой точке, используя закон Бугера сила уменьшается и берётся среднее арифметическое между первоначальной точкой и точкой пересечения отраженного луча. Сила света складывается.

Таким образом цвет пикселя получает значение точки пересечения исходного луча, помноженную на силу излучения.

По описанному выше алгоритму происходит вычисление цвета для каждого пикселя в рабочей области.

# 2.4 Алгоритм отрисовки

Алгоритм отрисовки

# ГЛАВА 3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

# Заключение

В ходе курсовой работы было создано оконное приложение, позволяющее визуализировать и моделировать распространение света в пространстве.

Были выполнены все поставленные задачи:

1. Список задач

Разрабатываемое приложение обладает перспективами, какими – может потом в диплом можно доработать.

# Список литературы

# Приложение 1