|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Чванова Валерия Олеговна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*фамилия, имя, отчество*

Группа\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ИУ7-55Б\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тип практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_технологическая\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Чванова В. О.\_\_\_**

*подпись, дата фамилия, и.о.*

Руководитель практики **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Куров А. В. \_\_\_**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*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)

« \_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 2022\_ г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Компьютерная Графика\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент группы \_\_\_\_ИУ7-55Б\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Чванова Валерия Олеговна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Тема курсовой работы \_\_\_\_\_\_\_\_Моделирование радуги\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_учебная\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики (кафедра, предприятие, НИР) \_\_\_\_\_\_\_\_\_\_\_\_\_\_кафедра\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения работы: 25% к \_\_\_ нед., 50% к \_\_\_ нед., 75% к \_\_ нед., 100% к \_\_\_ нед.

***Задание:*** Разработать программу для моделирования радуги. В программе предусмотреть возможность определения местоположения наблюдателя и солнца, задание влажности воздуха, возможности перегенерирования расположения молекул. Изображение радуги должно быть реалистичным.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Оформление курсовой работы:***

Расчетно-пояснительная записка на \_\_\_\_\_ листах формата А4.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания « \_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

**Руководитель курсовой работы**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Куров А. В.\_\_\_\_

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** \_\_\_\_Чванова В. О.\_\_\_

(Подпись, дата) (И.О.Фамилия)

**Оглавление**

[Введение 3](#_Toc115072361)

[Постановка задачи 3](#_Toc115072362)

[1. Аналитическая часть 4](#_Toc115072363)

[1.1 Описание объектов сцены 4](#_Toc115072364)

[1.2 Физика радуги 5](#_Toc115072365)

[1.2.1 Ход светового луча в капле дождя 5](#_Toc115072366)

[1.2.2 Общие расчеты в радуге 8](#_Toc115072367)

[1.3 Анализ алгоритмов 12](#_Toc115072368)

[1.3.1 Анализ алгоритмов для определения пересечения луча и массива капель 12](#_Toc115072369)

[1.3.1.1 Алгоритм ЦДА в двумерном пространстве 12](#_Toc115072370)

[1.3.1.2 Геометрический алгоритм 12](#_Toc115072371)

[1.3.1.3 Вывод 13](#_Toc115072372)

[1.3.2 Алгоритм сравнения текущей точки и точек из массива 13](#_Toc115072373)

[1.3.2.1 Алгоритм сравнения текущей точки и точек из массива 13](#_Toc115072374)

[1.3.2.2 Алгоритм бинарного поиска 13](#_Toc115072375)

[2. Конструкторская часть 15](#_Toc115072376)

[2.1 Общий алгоритм решения поставленной задачи 15](#_Toc115072377)

[2.2 Алгоритм определения пересечения луча и массива капель 15](#_Toc115072378)

[2.3 Алгоритм определения конца луча (по координатам начала, углам и одной из координат конца) 16](#_Toc115072379)

[2.4 Алгоритм сравнения текущей точки и точек из массива 17](#_Toc115072380)

[2.5 Вывод 18](#_Toc115072381)

[3. Технологическая часть 19](#_Toc115072382)

[3.1 Выбор языка программирования и среды разработки 19](#_Toc115072383)

[3.2 Промежуточные результаты 20](#_Toc115072384)

[Заключение 21](#_Toc115072385)

[Использованные источники 22](#_Toc115072386)

# Введение

На сегодняшний день компьютерная графика является одним из самых быстрорастущих сегментов в области информационных технологий. В ней сочетаются наука, творчество и технические достижения.

Компьютерная графика применяется во множестве специализаций: в отраслях техники, науки, медицины и архитектуры, трехмерные графические объекты используются для наглядного отображения разнообразной информации и презентации различных проектов.

­­­­­­­Компьютерная графика может быть полезна для отображения физических процессов. Например, иногда возникают ситуации, когда процессов и действующих объектов слишком много, и вручную смоделировать результат достаточно сложно.

Часто в процессе отрисовки изображений происходит большое количество вычислений, поэтому основным направлением в развитии компьютерной графики является ускорение вычислений.

# Постановка задачи

Необходимо разработать программный продукт с пользовательским интерфейсом для моделирования радуги. В программе предусмотреть возможность определения местоположения наблюдателя и солнца, задание влажности воздуха, возможности перегенерирования расположения молекул.

# 1. Аналитическая часть

### 1.1 Описание объектов сцены

Сцена состоит из источника света, множества капель, наблюдателя и плоскости горизонта.

Источник света представляет собой расположенную в бесконечности нематериальную точку (солнце), пускающую лучи света параллельно друг другу в заданную сторону. Лучи обрабатываются по очереди, и после обработки луч исключается из рассмотрения.

Для решения данной задачи была принята система координат, представленная на рис 1.1.1. Луч в данной системе координат задается начальной точкой и тремя углами поворота относительно осей. При этом угол поворота относительно оси ox отсчитывается от оси oz, угол поворота относительно оси oy - от оси ox, угол поворота относительно оси oz - от оси oy (подписи на рис. 1.1.1).

oz

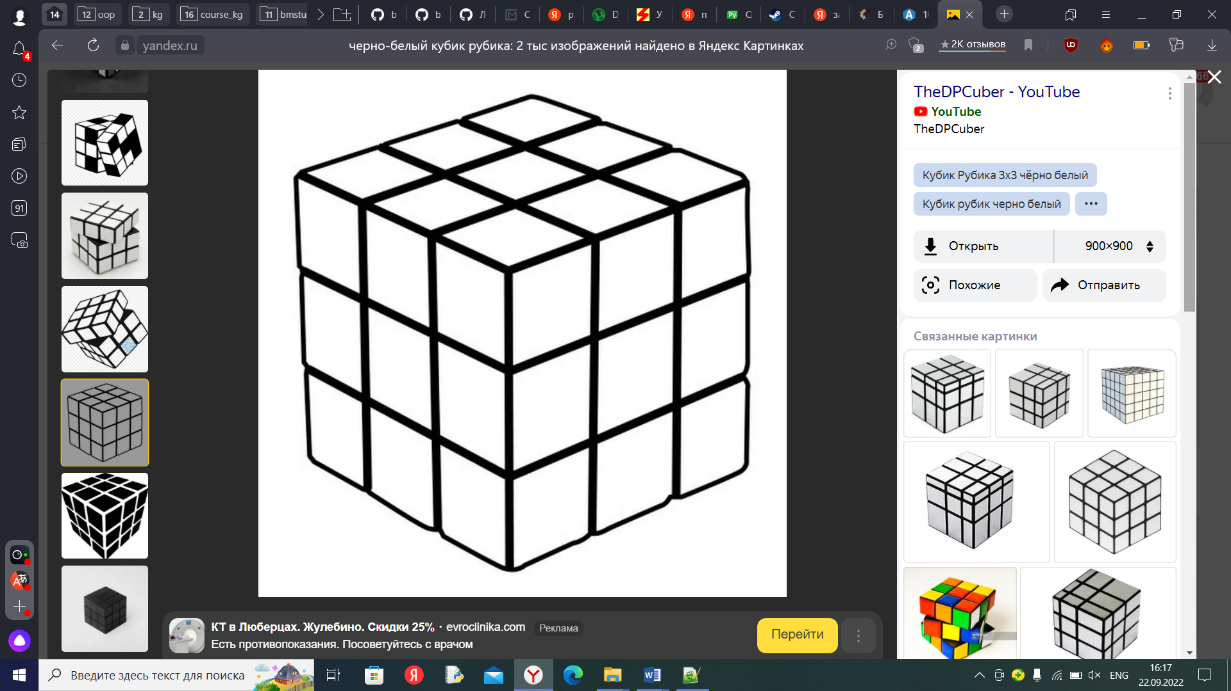
x - 0⁰

ox

y - 0⁰

oy

z - 0⁰



Изменяется z

Изменяется x

Изменяется y

рис. 1.1.1 рис. 1.1.2

Множество капель – множество материальных точек в пространстве. Генерируется и хранится в виде отсортированного трехмерного массива (рис. 1.1.2). Поскольку x, y, z могут изменяться не только на единицу, в каждой ячейке такого массива хранятся координаты капли в пространстве (x\_, y\_, z\_).

Наблюдатель представляет собой материальную точку (глаз наблюдателя) и считается точечным приемником.

Плоскость горизонта – это некая ограничивающая плоскость, на которой для наблюдателя строится радуга (на самом деле изображение радуги формируется в глазу). Она располагается перпендикулярно плоскости, созданной наблюдателем и лучами. Предполагается, что за этой плоскостью не расположено никаких объектов.

### 1.2 Физика радуги

#### 1.2.1 Ход светового луча в капле дождя

На дождевую каплю радиусом R падает пучок параллельных световых лучей. Прицельным параметром луча называется отношение ξ = Q/R, где Q - расстояние от данного луча до параллельной ему прямой, проходящей через центр капли. Будем сначала полагать, что все световые лучи имеют одну и ту же длину волны, т. е. будем рассматривать преломление и отражение света в капле без учета его дисперсии. Капля имеет форму шара. Сферическая симметрия капли приводит к тому, что все лучи с одинаковым прицельным параметром (эти лучи показаны на (рис. 1.2.1.1)) будут описывать внутри капли аналогичные траектории и будут выходить из капли под одним и тем же углом к первоначальному направлению. Траектория каждого луча лежит в плоскости, проходящей через исходное положение данного луча и параллельную ему прямую, проведенную через центр капли. Поэтому будем рассматривать двумерную задачу, изображая ход световых лучей в упомянутой плоскости.



рис. 1.2.1.1 рис. 1.2.1.2

На (рис. 1.2.1.2) показан ход луча, имеющего прицельный параметр Q/R. Легко видеть, что Q/R= sin α, где α - угол падения луча на поверхность капли. Так как треугольники АОВ и ВОС равнобедренные, то ∠ОАВ = ∠АВО и ∠OBC = ∠BСO. Угол падения равен углу отражения, поэтому ∠ABO = ∠OBС. Обозначим все эти углы через β. Картина хода луча симметрична относительно прямой OOˊ. В точках А и С световой луч, преломляясь, поворачивается на угол α - β (в каждой из этих точек). В точке В световой луч поворачивается на угол 180° - 2β. Таким образом, выходящий из капли луч СС1 оказывается повернутым относительно исходного направления на угол 2(α - β) + (180⁰ - 2β) = 180⁰ + 2α - 4β. Это есть угол 180⁰ - γ. Отсюда следует, что γ = 4β - 2α. Выразим угол γ через прицельный параметр луча, падающего на каплю. Закон преломления в точке А имеет вид: sinα\*sinβ = η, где η - показатель преломления воды. Используя γ = 4β - 2α, получаем . Или, иначе, . Итак, , а с учетом того, что , (1).

По мере увеличения прицельных параметров лучей от нуля до единицы угол γ растет от нуля до некоторого максимального значения, а затем уменьшается, снова обращаясь в нуль. Найдем максимальное значение угла γ.

Для этого надо найти максимум функции γ(ξ), ξ определяемой выражением (1). Для этого функцию γ(ξ) дифференцируем и затем приравниваем производную нулю. Производная функция γ(ξ) есть . Приравняв ее нулю, получим уравнение: . Решая это уравнение, находим значение ξˊ, при котором функция γ(ξ) принимает максимальное значение. Находим, что (2.2). При η = 4/3 имеем ξˊ = 0,861. Подставляя (2) в (1), получаем выражение для максимального угла между падающим на каплю и выходящим из нее лучами: (1.3). При η = 1,3330 имеем γ\_max = 42,08⁰.



рис. 1.2.1.3 рис 1.2.1.4

На (рис. 1.2.1.3) представлена для желтых лучей зависимость угла γ от прицельного параметра ξ. На дождевую каплю падают лучи со всевозможными (от 0 до 1) прицельными параметрами. Они выходят из капли под разными углами γ. Естественно, что наблюдатель увидит более яркими те лучи, которые будут иметь меньшую расходимость. Такими являются лучи, попадающие в район максимума кривой, изображенной на (рис. 1.2.1.3), т. е. лучи, для которых γ = 42⁰. По выражению Ньютона, именно эти лучи «более сгущены».

Сгущение выходящих из капли лучей вблизи угла γ = 42⁰ хорошо демонстрирует (рис. 1.2.1.4), на котором показаны рассчитанные траектории световых лучей, характеризующихся разными прицельными параметрами (траектории получены для η = 1,3330).

Таким образом, для желтого цвета угол выходящего из капли луча можно принять равным 42,08⁰. Найдем эти углы для остальных цветов радуги.



рис. 1.2.1.5

Для красного луча: η = 1,3311, .

Для зеленого луча: η = 1,3345, .

Для синего луча: η = 1,3374, .

Для фиолетового луча: η = 1,3428, .

#### 1.2.2 Общие расчеты в радуге

Для наблюдателя, находящегося на относительно ровной земной поверхности, радуга появляется при условии, что угловая высота солнца над горизонтом не превышает 42°. Чем ниже солнце, тем больше угловая высота вершины радуги и тем, следовательно, больше наблюдаемый участок радуги.



Рис 1.2.2.1

Лучи, образующие первичную радугу, формируют конус. Его вершина находится в глазу наблюдателя (на рисунке 1.2.2.1 точка O), а ось параллельна лучам солнца и направлена в противосолнечную точку (O1 - она находится ниже линии горизонта CD). ∠AOО1 = ѱ - угловая высота солнца над горизонтом. Из рисунка видно, что радуга представляет собой окружность конуса, ось которого есть OO1, γ - угол между осью конуса и любой из его образующих. Разумеется, наблюдатель видит не всю указанную окружность, а только ту часть ее (на рисунке участок CBD), которая находится над линией горизонта. Заметим, что ∠АОВ = ϕ есть угол, под которым наблюдатель видит вершину радуги, a ∠AOD = α - угол, под которым он видит основание радуги. Очевидно, что ϕ + ѱ = γ. Таким образом, положение радуги по отношению к ландшафту зависит от положения наблюдателя по отношению к солнцу, а угловые размеры радуги определяются высотой солнца над горизонтом. При передвижениях наблюдателя указанный конус, а значит, и радуга соответствующим образом перемещаются.

Из формулы ϕ + ѱ = γ следует, что ϕ = γ - ѱ. Для основной радуги угол γ равен 42° (для желтого участка радуги). Отсюда ясно, почему земной наблюдатель не может любоваться основной радугой, если высота солнца над горизонтом превышает 42°. Иное дело, если наблюдатель находится в самолете: в этом случае он может увидеть радугу в виде полной окружности. Но где бы ни находился наблюдатель (на земной поверхности или над нею), он всегда есть центр ориентированного на солнце конуса с углом раствора около 42° для основной радуги.



Рис 1.2.2.2

Радуга имеет вид дуги, наблюдаемой под углом 42° к прямой линии, проходящей через наблюдателя и солнце. Для простоты будем полагать, что солнце находится у самой линии горизонта и что завеса дождя имеет вид отвесной стены, плоскость которой перпендикулярна к направлению солнечных лучей. На (рис. 1.2.2.2) дан разрез плоскостью, совпадающей с плоскостью поверхности, на которой стоит наблюдатель: MN - завеса дождя, О - наблюдатель, O1 - противосолнечная точка. На рисунке заштрихована область, в пределах которой в глаз наблюдателя попадают лучи, каждый из которых испытал в какой-то капле дождя отражение и двукратное преломление; вне этой области такие лучи к данному наблюдателю не попадают. Лучи, приходящие к наблюдателю от дождевых капель, находящихся справа от C и слева от D, заметно ослаблены из-за относительно большой расходимости. Наиболее яркими будут лучи, приходящие к наблюдателю от границы заштрихованной области, т. е. от капель вблизи точек С и D, поскольку расходимость этих лучей минимальна. Итак, если бы в спектре солнца была представлена лишь одна длина волны, наблюдатель увидел бы радугу в образе узкой светящейся дуги. При этом небо под дугой будет представляться наблюдателю несколько более светлым, чем небо над дугой. Капли дождя, конечно, не висят в воздухе неподвижно, а довольно быстро падают вниз. Поэтому та или иная капля участвует в формировании изображения радуги лишь в течение очень короткого промежутка времени - тогда, когда направление на нее от наблюдателя составляет с направлением OO1 угол, близкий к 42°. Затем капля выходит из игры, но на смену ей приходят новые и новые капли. Капли сменяют друг друга быстро, так что глаз наблюдателя не улавливает происходящей смены; в результате наблюдатель видит неподвижное изображение радуги.

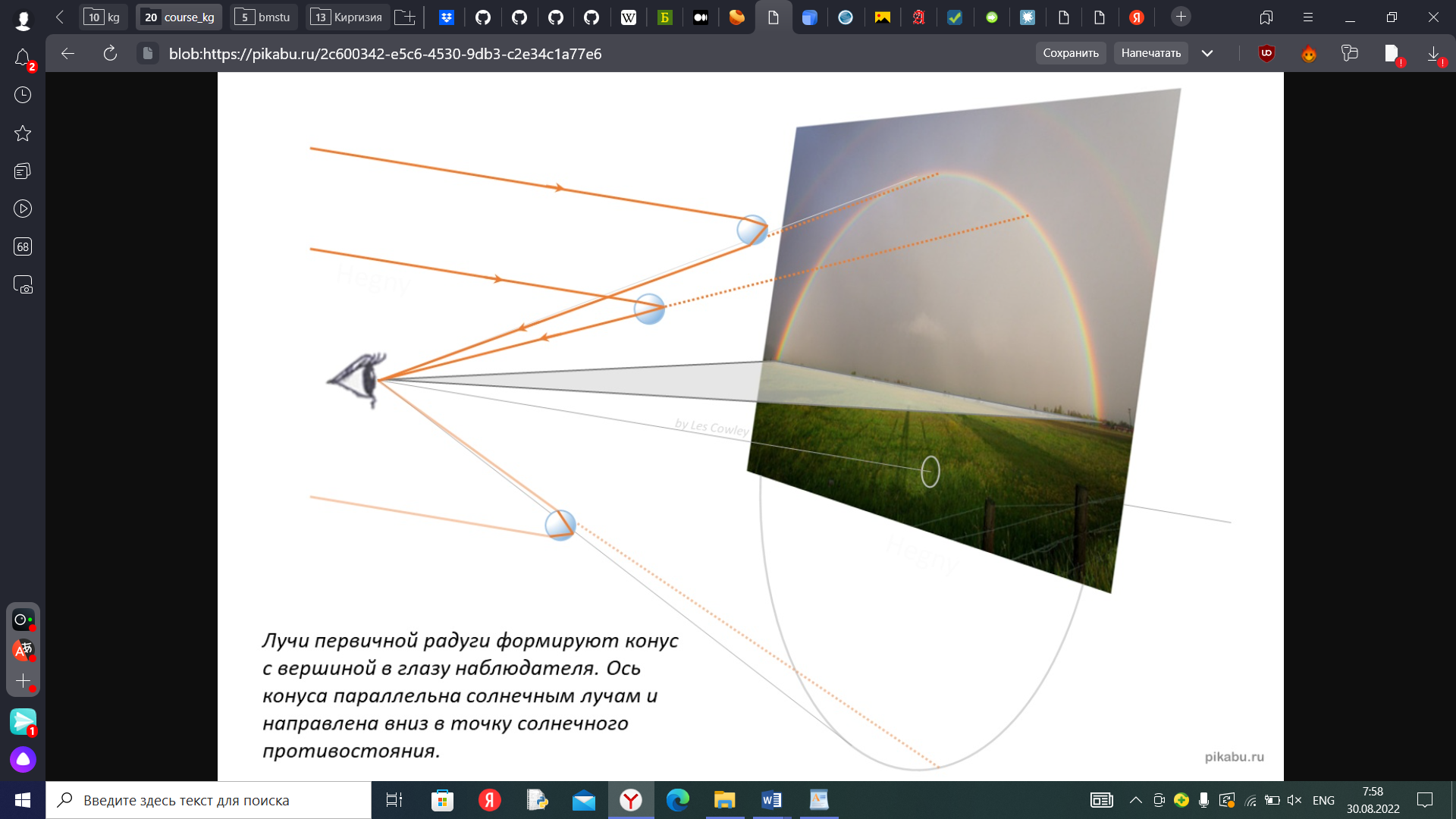


рис. 1.2.2.3

Миллионы капель, находящиеся рядом с поверхностью этого конуса, отражают свет в глаз наблюдателя и формируют изображение цветной дуги. Капли не обязательно должны находится радом, они могут быть на расстоянии несколько метров или даже километров друг от друга. Расстояние не имеет значения, радуга будет выглядеть точно так же.

### 1.3 Анализ алгоритмов

#### 1.3.1 Анализ алгоритмов для определения пересечения луча и массива капель

##### 1.3.1.1 Алгоритм ЦДА в двумерном пространстве

Алгоритм цифрового дифференциального анализатора (ЦДА) использует достаточно общий принцип, известный в математике: изучение какого-либо явления на основе дифференциального уравнения или системы таких уравнений, описывающей это явление. Поскольку прямая линия на плоскости описывается уравнением вида AX+BY+C=0, где A, B, C - коэффициенты этого уравнения, то производная dY/dX является постоянной. Заменив дифференциалы конечными разностями, получим: (1), где Xн, Yн и Xк, Yк - координаты начальной и конечной точек отрезка.

Ордината очередного пикселя Yi+1 может быть вычислена по известной ординате предыдущего пикселя Yi следующим образом: (2). Подставив dY из (1), получим нужную ординату пикселя. Остается определить величину приращения X. В рассматриваемых здесь алгоритмах большее из приращений (X или Y) выбирается в качестве единицы растра, а приращение вдоль другой координатной оси подлежит определению. Если же поступить по-другому (меньшее из приращений взять равным единице), то отрезок на экране может получиться «дырявым», то есть состоящим из отдельных точек, не расположенных вплотную друг к другу.

Если расширить этот алгоритм до трехмерного пространства, его можно использовать для решения поставленной задачи.

##### 1.3.1.2 Геометрический алгоритм

Поставленную задачу можно решить, рассматривая каждую каплю в отдельности, и определяя принадлежность заданной точки прямой в пространстве, исходя из геометрических соображений. Например, определяя углы отрезка, проведенного из начала луча в заданную точку, и сравнивая их с углами исходного луча.

##### 1.3.1.3 Вывод

Оба алгоритма нельзя назвать эффективными. В геометрическом алгоритме нам придется рассматривать каждую каплю в отдельности, а их может быть огромное количество. В алгоритме ЦДА необходимо будет рассматривать каждую точку луча, что может оказаться очень неэффективным при длинных лучах.

Для решения проблемы можно перенести точку старта для алгоритма ЦДА к началу множества капель, а весь предыдущий путь пропустить, используя алгоритм нахождения конца луча. Таким образом, мы будем анализировать только ту часть луча, которая уже попала во влажный воздух. Временные затраты сведутся к минимуму.

#### 1.3.2 Алгоритм сравнения текущей точки и точек из массива

##### 1.3.2.1 Алгоритм сравнения текущей точки и точек из массива

За счет способа хранения массива капель, алгоритм определения принадлежности данной точки массиву можно сделать достаточно эффективным.

Необходимо решить задачу поиска элемента в отсортированном массиве 3 раза. Наиболее эффективный метод – бинарный поиск. С его помощью сначала определяем, есть ли в данном массиве точки с заданной координатой z. Если такие находятся, применяем второй раз бинарный поиск – определяем, есть ли среди точек с заданной координатой z точки с заданной координатой y. Если такие находятся, применяем третий раз бинарный поиск – для точек с заданными координатами z, y, x. Таким образом сложность алгоритма можно оценить как .

##### 1.3.2.2 Алгоритм бинарного поиска

Бинарный поиск производится в упорядоченном массиве.  
При бинарном поиске искомый элемент сравнивается со средним элементом в массиве. Если они равны, то поиск успешен. В противном случае поиск осуществляется аналогично в левой или правой частях массива.  
Бинарный поиск также называют поиском методом деления отрезка пополам или дихотомии.

Количество шагов поиска определится как ln(n)↑, где n-количество элементов, ↑ — округление в большую сторону до ближайшего целого числа.

На каждом шаге осуществляется поиск середины отрезка по формуле mid = (left + right)/2. Если искомый элемент равен элементу с индексом mid, поиск завершается. В случае если искомый элемент меньше элемента с индексом mid, на место mid перемещается правая граница рассматриваемого отрезка, в противном случае — левая граница.

# 2. Конструкторская часть

### 2.1 Общий алгоритм решения поставленной задачи

**Основной алгоритм**:

1. Задаем расположение глаза – (x\_eye, y\_eye, z\_eye). В данной задаче глаз можно считать точечным приемником.

2. Генерируем массив расположения капель. Каждая точка в заданном диапазоне (в параллелепипеде) попадает в массив капель с вероятностью 10%.

3. Задаем плоскость горизонта. Считаем горизонт параллельным плоскости (oy, oz). То есть, задаем точку на оси x, где виден горизонт.

4. Пускаем солнечные лучи (углы: alpha\_x, alpha\_y, alpha\_z = 90, 0, 110).

5. Перебираем все лучи. Для каждого луча:

5.1. Определяем, где луч должен кончиться (чтобы не обрабатывать лишнее) – самая дальняя используемая точка (находим координаты конца луча).

5.2. Определяем, пересекается ли текущий луч с какой-нибудь из капель.

5.3. Если пересечения нет, данный луч дальше не обрабатывается.

5.4. Если пересечение есть, раскладываем общий луч на цветовые - учитываем дисперсию, с учетом четырех возможных вариантов вхождения луча в каплю (по четвертям, т.к. изначально рассматривается вариант q\_y = 0..0.5, q\_z = 0..0.5) для каждого находим углы выходящего из капли луча.

Находим координаты конца луча. Определяем, пересекается ли луч с глазом (с учетом раствора глаза). Если пересечение есть, продлеваем луч до пересечения с плоскостью горизонта, находим координаты точки, в которой будет видна радуга. С учетом яркости отрисовываем видимую точку радуги.

### 2.2 Алгоритм определения пересечения луча и массива капель

**Псевдокод:**

1. Получение исходных данных об отрезке, массива капель
2. Пропуск части луча, где нет капель – с помощью алгоритма определения конца луча
3. Проверка вырожденности отрезка, если вырожден, то переход к 7.
4. dx = Xк-Xн, dy = Yк-Yн, dz = Zк-Zн
5. Δx = |dx|, Δy = |dy|, Δz = |dz|
6. Если Δx > Δy и Δx > Δz,  
       то l\_points = Δx  
   иначе если Δy > Δx и Δy > Δz,

то l\_points = Δy

иначе l\_points = Δz

1. dx = dx/ l\_points, dy = dy/ l\_points, dz = dz/ l\_points
2. X=Xн, Y=Yн, Z=Zн
3. Цикл от i=1 до i= l\_points +1 с шагом 1.

* Сравнение текущей точки и точек из массива. Если такая точка есть в массиве, то пересечение найдено, переход к пн.9
* Вычисление координат следующей точки: X=X+dx, Y=Y+dy, Z=Z+dz.

1. Конец

### 2.3 Алгоритм определения конца луча (по координатам начала, углам и одной из координат конца)

**Псевдокод:**

Если задана координата coord = Х:

1. Переводим углы луча из нашей системы углов в углы (0, 0, 0)

2. Если новая координата Х совпадает с начальной х0,

то завершаем функцию, возвращаются значения х0, y0, z0

3. Сдвигаем отрезок в начало координат: coord -= x0

4. Поворачиваем отрезок вокруг оси oz:

x = coord \* cos(alpha\_z)

y = coord \* sin(alpha\_z)

z = 0

5. Поворачиваем отрезок вокруг оси oy:

x = x \* cos(alpha\_y) + z \* sin(alpha\_y)

z = -1 \* x \* sin(alpha\_y) + z \* cos(alpha\_y)

6. Поворачиваем отрезок вокруг оси ox:

y = y \* cos(alpha\_x) - z \* sin(alpha\_x)

z = y \* sin(alpha\_x) + z \* cos(alpha\_x)

7. Вытягиваем отрезок в coord/x раз

z \*= coord/x

y \*= coord/x

x \*= coord/x

8. Переносим отрезок в начальную точку

x += x0

y += y0

z += z0

При заданных координатах Y и Z выполняются аналогичные действия.

Например, задана координата конца X. Тогда отрезок ((0, 0, 0), (X – x0, 0, 0)) просто поворачивается на заданные углы, вытягивается в (X - х0) / x\_new раз и переносится в точку с координатами (x0, y0, z0). Аналогично при заданных других координатах.

### 2.4 Алгоритм сравнения текущей точки и точек из массива

**Псевдокод:**

1. Получение исходных данных: точки и массива arr
2. i = 0, j = 0, k = 0
3. l = 0, r = len(arr)-1
4. Пока r > l:

4.1. m = (r + l) / 2

4.2. Если z точки из массива (arr[m][j][k][2]) < z точки, то l = m

иначе r = m

1. Если z точки из массива (arr[m][j][k][2]) != z точки, то точки нет в массиве, переход к пн. 15
2. i = m
3. l = 0, r = len(arr[i])-1
4. Пока r > l:

8.1. m = (r + l) / 2

8.2. Если y точки из массива (arr[i][m][k][1]) < y точки, то l = m

иначе r = m

1. Если y точки из массива (arr[i][m][k][1]) != y точки, то точки нет в массиве, переход к пн. 15
2. j = m
3. l = 0, r = len(arr[i][j])-1
4. Пока r > l:

12.1. m = (r + l) / 2

12.2. Если x точки из массива (arr[i][j][m][0]) < x точки, то l = m

иначе r = m

1. Если x точки из массива (arr[i][j][m][0]) != x точки, то точки нет в массиве, переход к пн. 15
2. k = m
3. Если k != 0, то точка есть в массиве
4. Конец

### 2.5 Вывод

Описанный алгоритм работает очень долго. Поэтому моделирование всех физических процессов (со всевозможными упрощениями) - очень затратная по времени операция. Если не задаваться целью смоделировать всю физику радуги, то радугу можно моделировать, основываясь на геометрических построениях. Можно сказать, что если мы нарисуем конус с осью от глаза, наклоненной под тем же углом, что и лучи, и углом при острие 2\*42 градуса, то для каждой капли, лежащей на этом конусе, найдется луч, который отразится в этой капле одной из своих цветовых составляющих по направлению к вершине конуса. То есть, достаточно знать расположение глаза и горизонта и углы падения лучей, чтобы смоделировать радугу исходя из геометрических построений.

# 3. Технологическая часть

В данной части рассматривается выбор средств реализации программного обеспечения.

### 3.1 Выбор языка программирования и среды разработки

В качестве языка программирования для написания программного продукта был выбран язык *python*, так как он:

1. Является объектно-ориентированным. Объектно-ориентированный подход позволяет четко структурировать программу, достаточно легко расширять и модифицировать ее, использовать общий интерфейс взаимодействия с разными объектами, а также уменьшить дублирование кода.
2. Предоставляет широкий набор библиотек.
3. Изучался в рамках курса «Программирование», что уменьшает временные затраты на предварительное знакомство с языком.

В качестве среды разработки была выбрана среда *Qt Creator*. Она обладает редактором кода, отладчиком, а также широким функционалом по работе с интерфейсами.

### 3.2 Промежуточные результаты

На рис. 3.2.1 представлен промежуточный результат работы программы – отрисовано множество капель, плоскость горизонта, исходные лучи, лучи, прошедшие два преломления и одно отражение в каплях, лучи, попавшие в глаз наблюдателя.

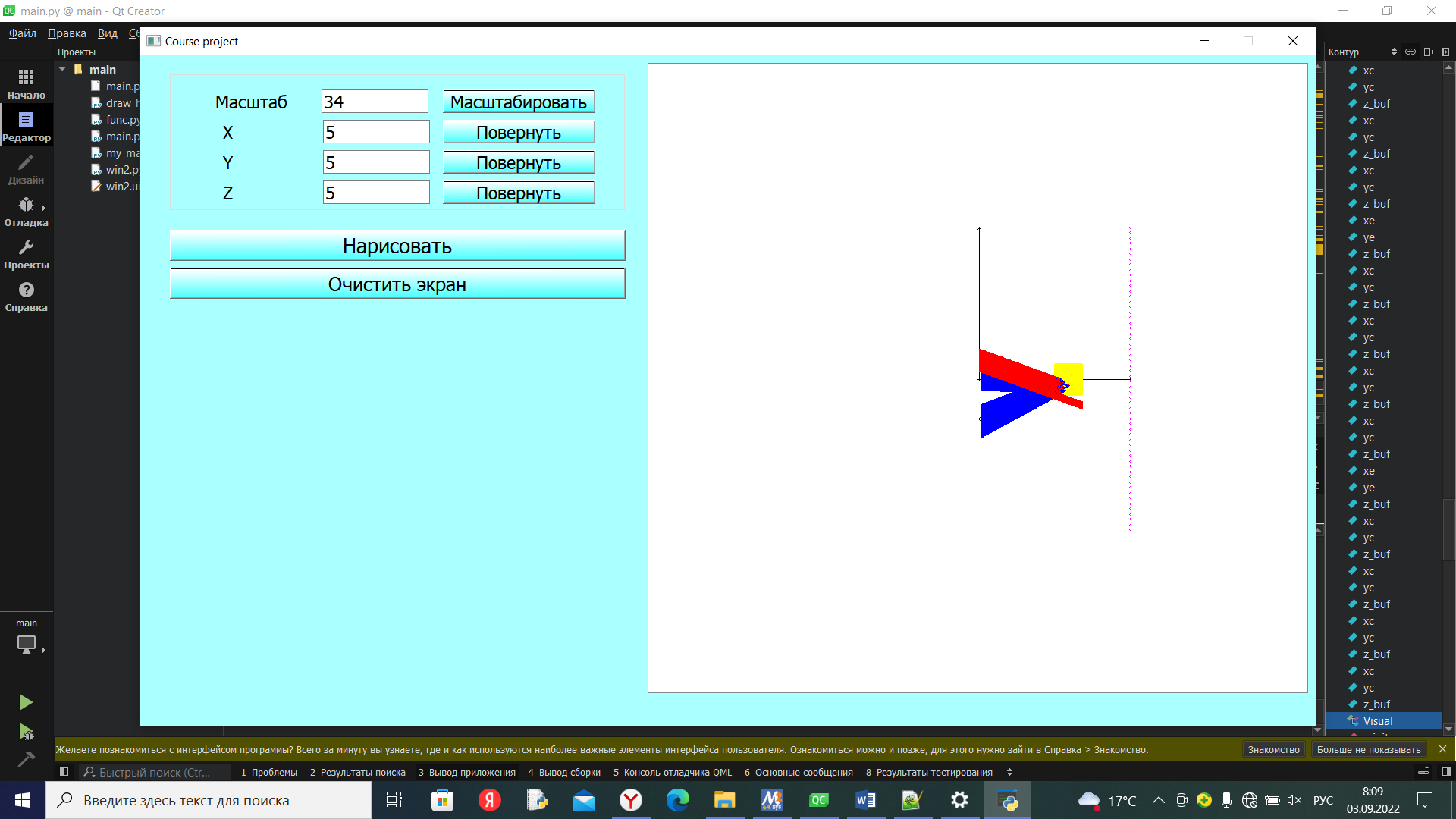
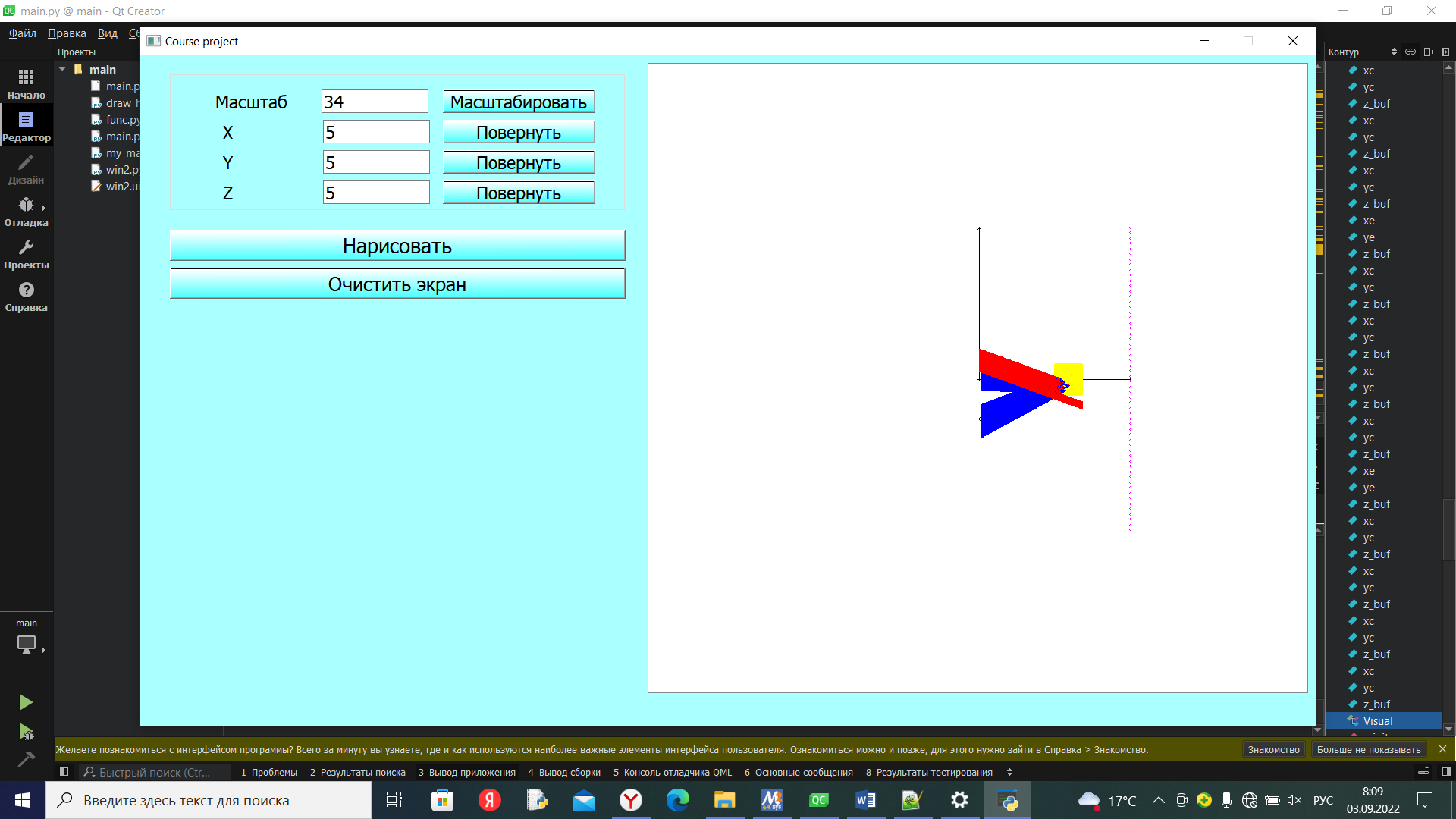
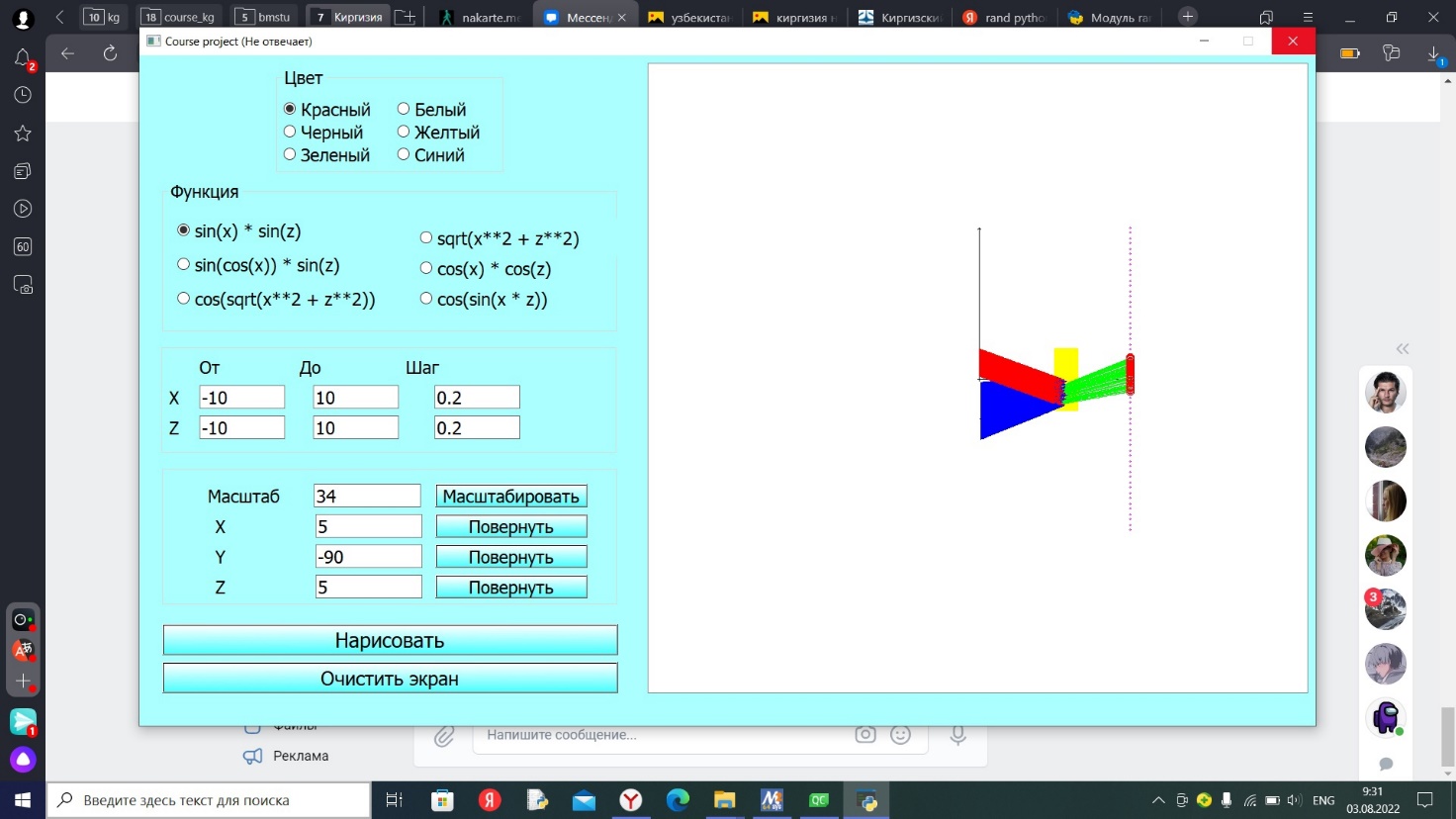


рис. 3.2.1

# Заключение

В результате проделанной работы была изучена физика радуги, смоделирована и запрограммирована часть, касающаяся движения лучей. Данная работа позволила закрепить приобретенные в течение учебного процесса знания и навыки в областях компьютерной графики и проектирования программного обеспечения, а также лучше изучить возможности языка программирования python и среды разработки Qt Creator.

При дальнейшей работе планируется завершить моделирование физики радуги, максимально оптимизируя вычислительные процессы; смоделировать и запрограммировать упрощенную модель радуги, основанную на геометрических построениях; добиться реалистичного изображения радуги.

# Использованные источники

1. Куров А.В., Курс лекций по компьютерной графике в МГТУ им. Н.Э.Баумана, 2022.
2. Как устроена радуга [Электронный ресурс]. — Режим доступа: <https://pikabu.ru/story/kak_ustroena_raduga_chast_1_pervichnaya_raduga_7489102> (дата обращения 05.07.2022)
3. Астронет. Еще раз о радуге [Электронный ресурс]. — Режим доступа: <http://www.astronet.ru/db/msg/1175762/page1.html> (дата обращения 07.07.2022)
4. [Левитин А. В.](https://www.wikidata.org/wiki/Q21694518) Глава 4. Метод декомпозиции: Бинарный поиск // [Алгоритмы. Введение в разработку и анализ](https://www.wikidata.org/wiki/Q21694522) — М.: [Вильямс](https://www.wikidata.org/wiki/Q21694521), 2006. — С. 180—183. — 576 с.