



Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра алгоритмических языков

Носов Анатолий Владимирович

Модель оптических экспериментов в зеркальной комнате

Отчет о выполнении задания практикума

Москва, 2023

1. Постановка задачи

Зеркальная комната представляется в виде замкнутого контура, каждая стена – плоское или сферическое зеркало.

Для проведения экспериментов необходимо определить для каждой стены комнаты вид зеркала (плоское или сферическое), а для каждого сферического зеркала – направления изгиба и радиус кривизны.

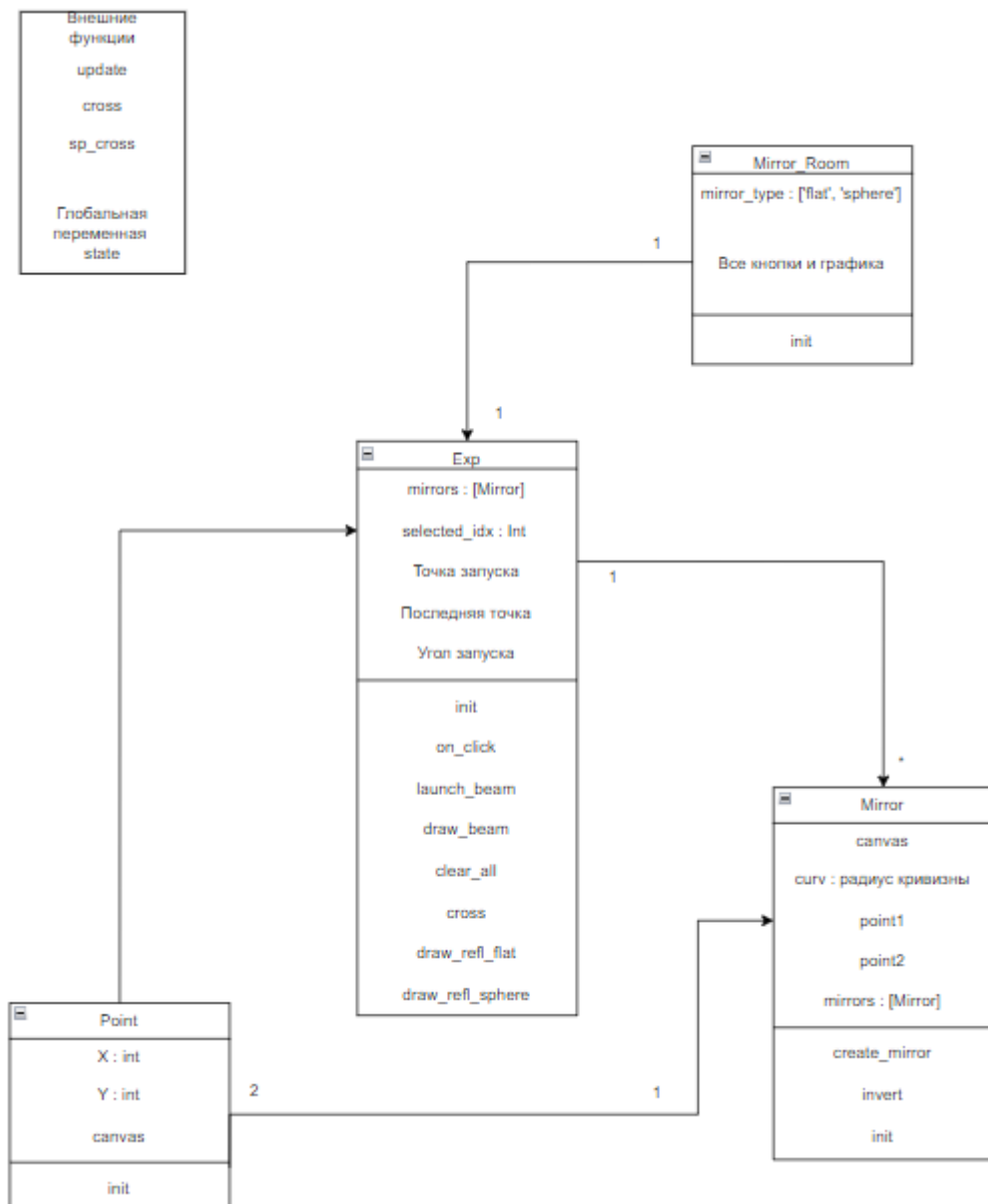
Основная функция программной системы – проведение оптического эксперимента, при котором из некоторой точки комнаты, по какому-то направлению выпускается луч света, и затем показывается его путь внутри комнаты с учетом отражений от зеркал.

Траектория луча определяется физическими законами отражения от зеркальных поверхностей.

Цель моделирования – подбор пользователем системы параметров зеркал и луча, при которых луч, отражаясь от зеркальных стен, попадает в выбранную пользователем зону комнаты.

При визуализации оптического эксперимента должен быть показан план комнаты и изображен путь луча в комнате.

2. Диаграмма классов



3. Спецификации интерфейсов

```

class Mirror_Room:
    def __init__(self, master):

```

```

class Exp(Mirror_Room):
    def __init__(self, canvas, control):

```

```

def on_click(self, event): --> handle LMB click
def launch_beam(self):
def draw_beam(self, launch_point, end_point):
    --> change launch and angle point

def clear_all(self):
def cross(self, x1_1, y1_1, x1_2, y1_2,
           x2_1, y2_1, x2_2, y2_2): --> None or [x, y]

def draw_reflection_flat(self, launch_x, launch_y,
                        end_x, end_y,
                        mirror_start, mirror_end):
    --> change launch and angle point

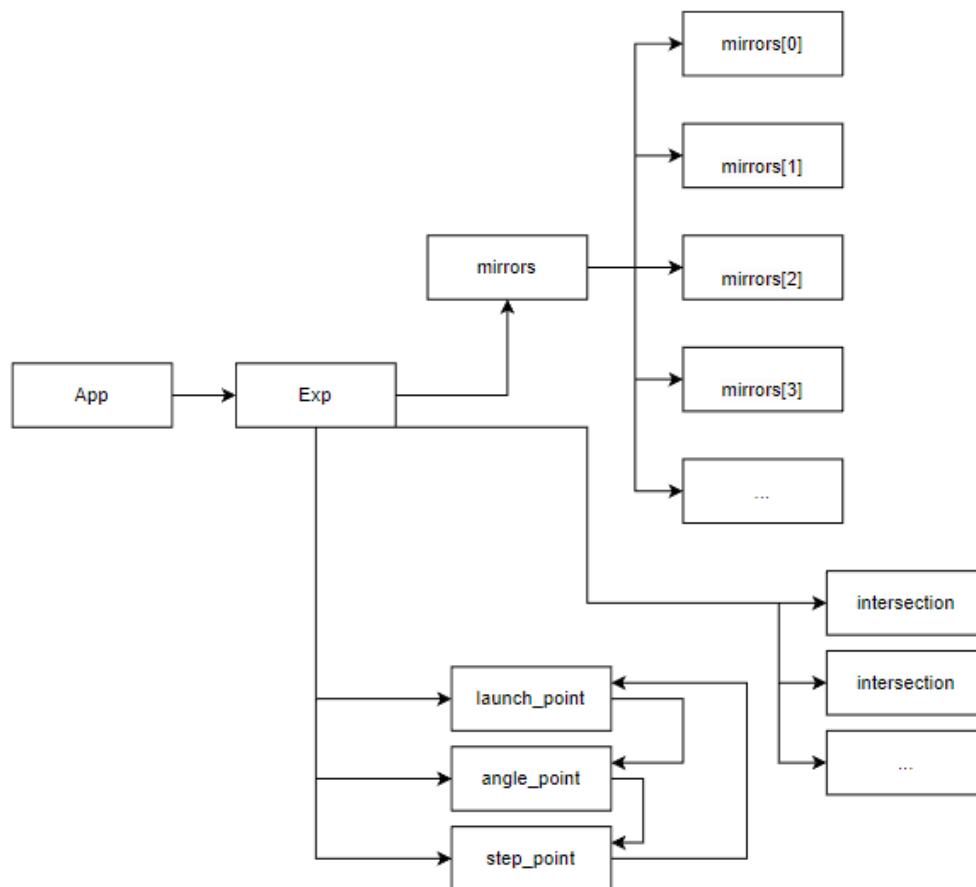
def draw_reflection_sphere(self, launch_x, launch_y,
                        end_x, end_y,
                        center, r,
                        point1_x, point1_y,
                        point2_x, point2_y):
    --> change launch and angle point

class Point:
    def __init__(self, canvas, x, y, w = 2):

class Mirror(Exp):
    def __init__(self, canvas, point1, point2, curv, idx, mirrors):
    def create_mirror(self):
    def invert(self, idx, x, y, r):
    def handle_slider(self, event):

```

4. Диаграмма объектов



5. Инструментальные средства

1. Язык разработки – Python.
2. Среда разработки – Jupyter.
3. Графический интерфейс – tkinter.
4. Используемые библиотеки – tkinter, math.

6. Сценарий работы

Предложенный интерфейс содержит в себе несколько кнопок:

1. Reg mirror - по двум кликам будет строиться плоское зеркало/
2. S mirror - по двум кликам будет строиться сферическое зеркало, появится кнопка для изменения направления зеркала (Invert) и слайдер для изменения радиуса. Каждый слайдер и кнопка привязаны к конкретному зеркалу.
3. Exit - закрытие приложения.

Ход работы программы:

1. Пользователь выбирает тип зеркала используя кнопки в левой части приложения.

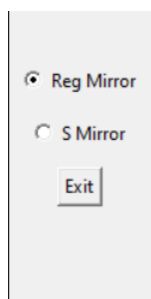


Рис. 1: Визуальное представление п.1

2. По нажатию ЛКМ создается точка, от которой будет строиться зеркало. Повторное нажатие в любое место построит зеркало между двумя точками.
3. Далее для построения нового зеркала требуется всего одно нажатие, т.к. начальной точкой будет считаться конечная точка предыдущего зеркала.

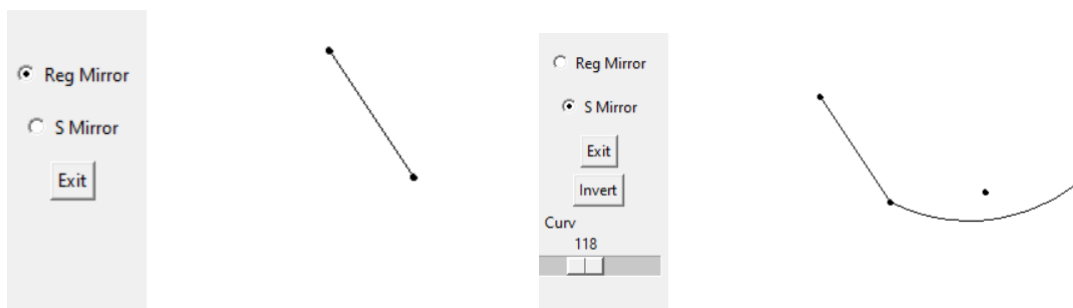


Рис. 2: Визуальное представление п.2, 3

4. При достаточной близости новой точки к самой первой контур автоматически замыкается, в консоли появляется вывод "ready".

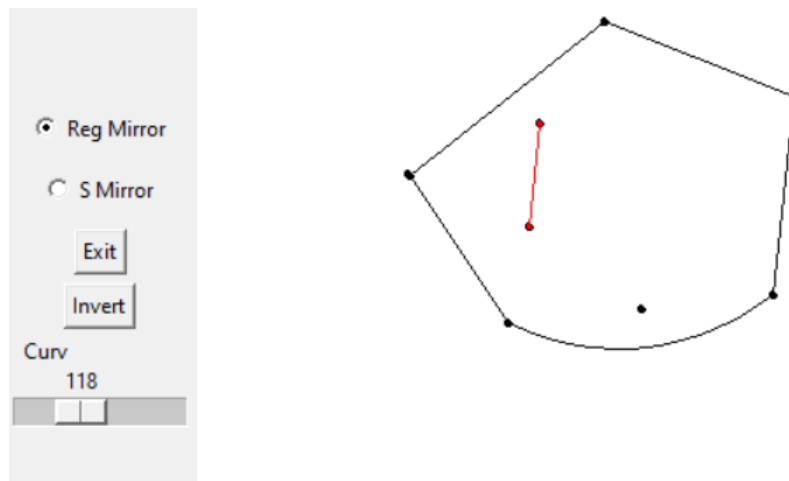


Рис. 3: Визуальное представление п.5

5. Следующие два клика пользователя задают начало и направление луча.
6. Следующий клик задает круглую область, по попаданию в которую луча, программа остановит симуляцию.

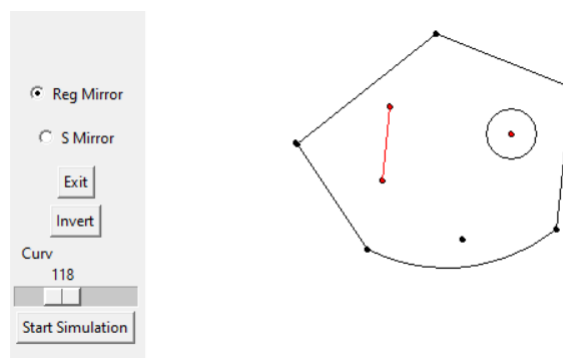


Рис. 4: Визуальное представление п.6

7. Появляется кнопка "Start simulation", нажатие которой строит путь луча до отражения. Для продолжения симуляции нужно повторно нажать на кнопку.
8. По завершению симуляции на экране выводится текст "End of simulation" и появляется кнопка "New simulation", позволяющая запустить новую симуляцию.

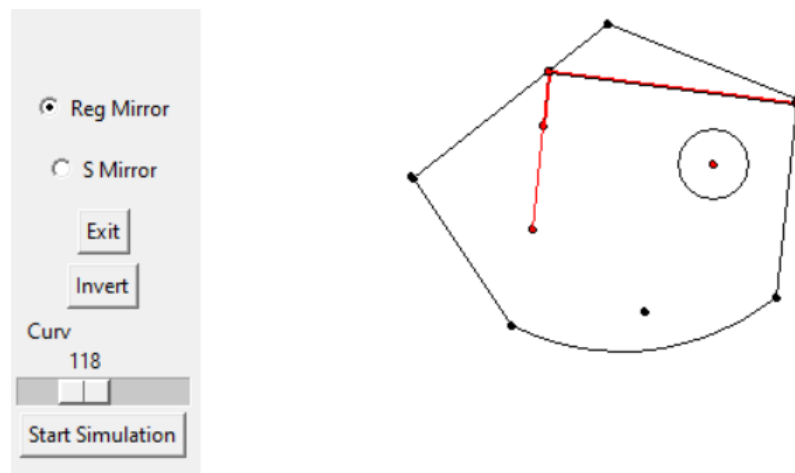


Рис. 5: Визуальное представление п.7

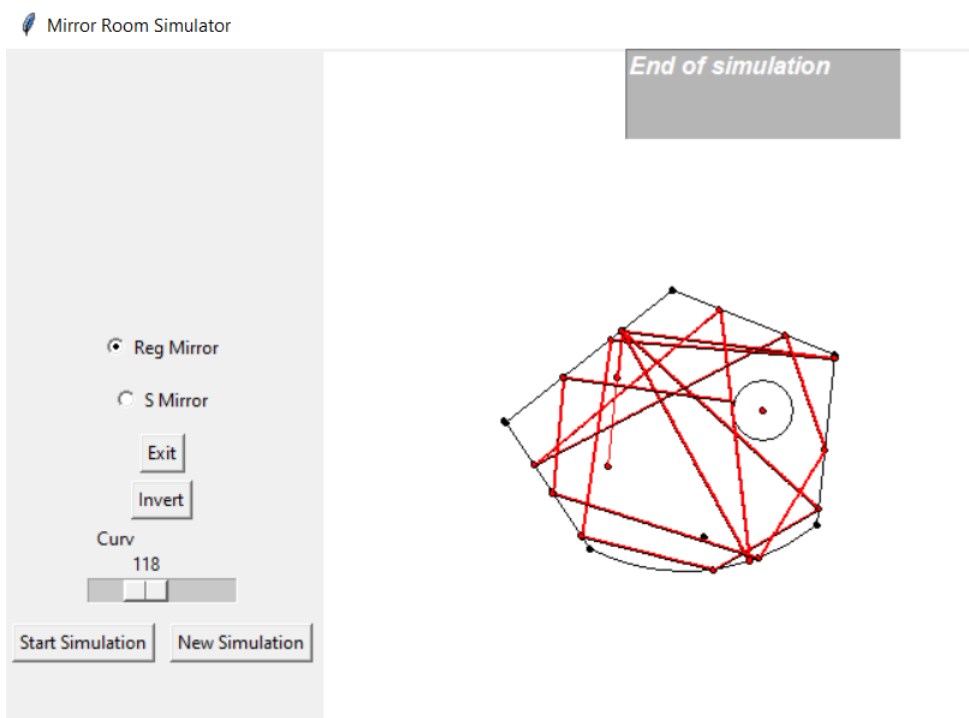


Рис. 6: Визуальное представление п.8