



МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ТЕХНОЛОГИЧЕСКОГО
ОБРАЗОВАНИЯ

Кафедра информационных технологий и электронного обучения

ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4

По дисциплине: Математические основы компьютерной
графики

Выполнил студент 3 курса группы 1.1

Щербинин А. В.

Санкт-Петербург
2022

Постановка задачи:

1. Комплект 1:

См. программу ManySquares из приложенного архива. Программа написана на С и является «ответом», но вам нужно сделать решение на Python. Ожидаемый результат на рисунке 1. Обратите внимание также на начерченные оси. В данном примере создано окно с размерами области рисования 800×600 точек и с размером одной единицы измерения координаты 65 точек для каждой декартовой оси.

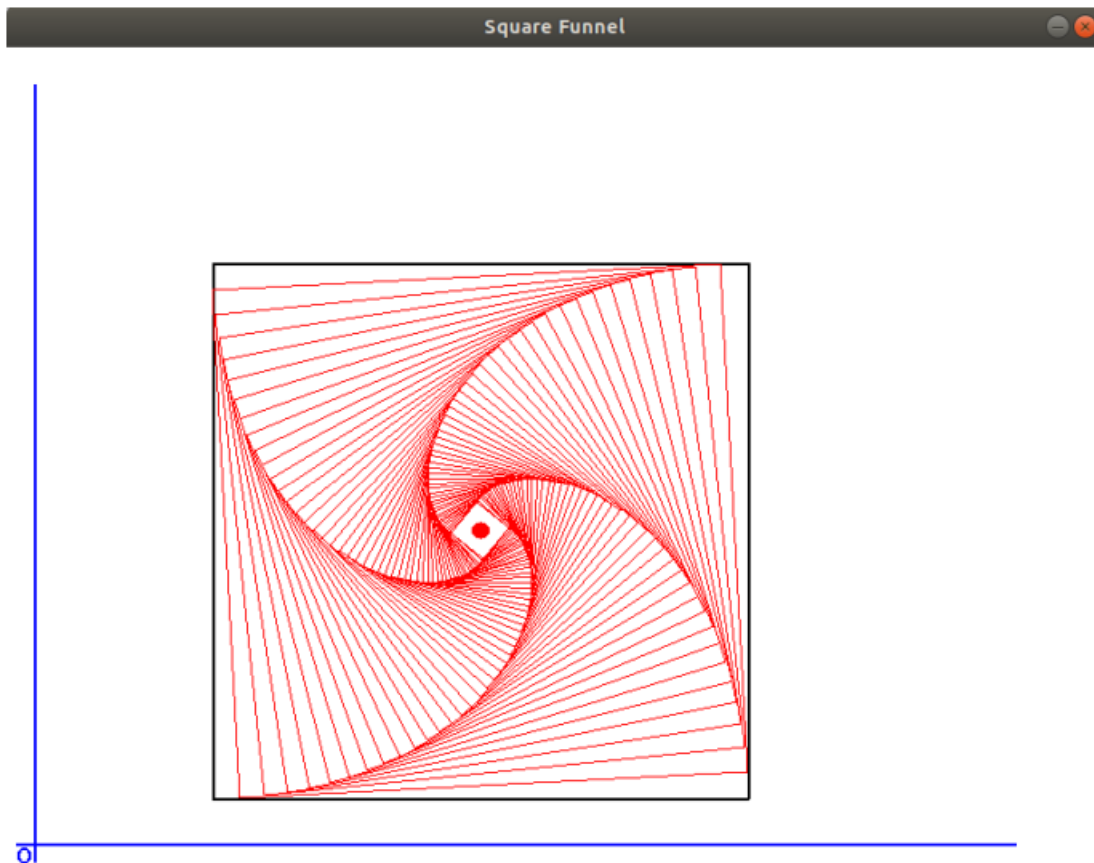


Рис. 1:

Комплект 1: Вращения

Код программы

practice4.py

```
import pygame
import numpy as np
import graphics as g

def main():
    window = g.Drawer(800, 600, 64)
    window.initialize("ManySquares")
    window.draw_text(-20, 7 * 65 + 5, "0", 36, (255, 255, 255), 100)
    window.draw_axes(0, 8 * 65, 0, 7 * 65, 3, (255, 255, 255), 100)
    L = np.array([[2, 0.5], [8, 0.5], [8, 6.5], [2, 6.5]])
```

```

window.draw_line(L, 3, (255, 255, 255), 100)
p = 0.95
q = 1 - p
Lo = np.copy(L)
for i in range(50):
    Lo[0][0] = p * Lo[0][0] + q * Lo[1][0]
    Lo[0][1] = p * Lo[0][1] + q * Lo[1][1]
    Lo[1][0] = p * Lo[1][0] + q * Lo[2][0]
    Lo[1][1] = p * Lo[1][1] + q * Lo[2][1]
    Lo[2][0] = p * Lo[2][0] + q * Lo[3][0]
    Lo[2][1] = p * Lo[2][1] + q * Lo[3][1]
    Lo[3][0] = p * Lo[3][0] + q * Lo[0][0]
    Lo[3][1] = p * Lo[3][1] + q * Lo[0][1]
    window.draw_line(Lo, 3, (255, 0, 0), 100)

main()

FPS = 30
clock = pygame.time.Clock()
running = True
while running:
    for i in pygame.event.get():
        if i.type == pygame.QUIT:
            running = False
            pygame.quit()
        clock.tick(FPS)
        pygame.display.update()

```

graphics.py

```

import pygame as pg

class Drawer:
    # Class constructor
    def __init__(self, res_x: int, res_y: int, color_depth: int):
        self.instance = pg.display.set_mode((res_x, res_y), color_depth)

    # Initialize 'pygame' graphics separately
    def initialize(self, caption: str):
        pg.init()
        pg.font.init()
        pg.display.set_caption(caption)

```

```

# Put a text on the graphic surface
def draw_text(self, x: float, y: float, text: str, size, color, offset:
float):
    f = pg.font.SysFont('Times New Roman', size)
    self.instance.blit(f.render(text, True, color),
                        (x + offset, y + offset))

# Draw a line
def draw_line(self, Mat,
               width: int, color, offset):
    pg.draw.lines(self.instance, color, True, Mat * 65 + offset, width)

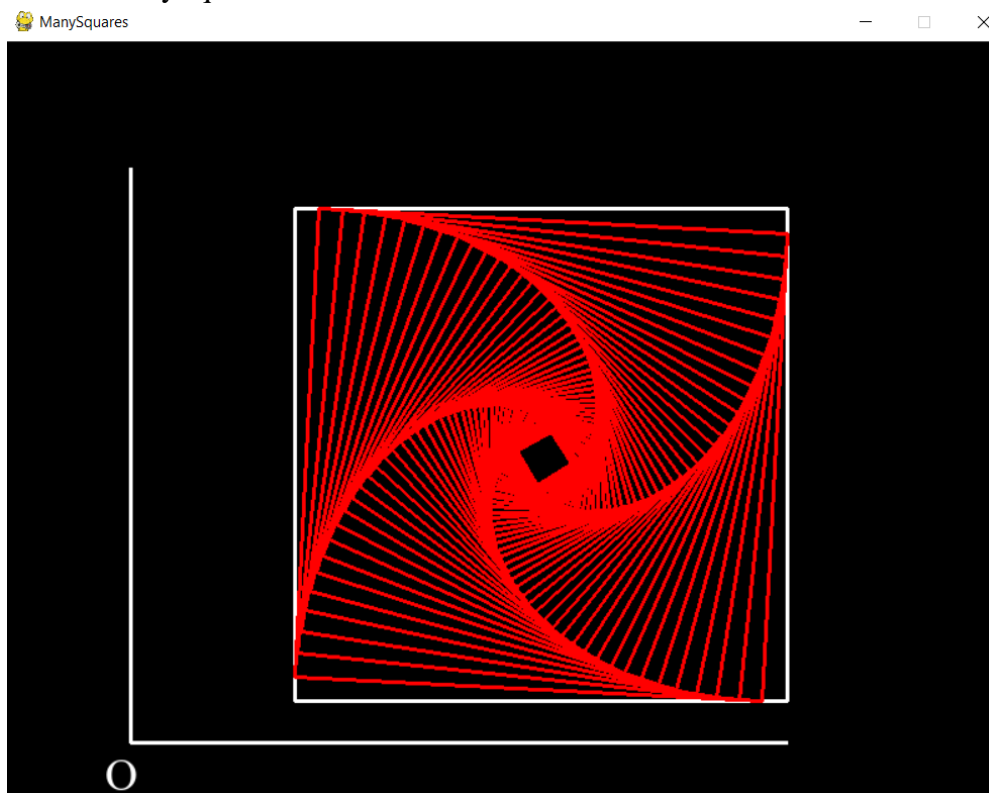
# Draw axes Ox and Oy with point O
def draw_axes(self, x_min: float, x_max: float, y_min: float,
               y_max: float, width: int, color, offset: float):
    pg.draw.lines(self.instance, color, True, ([x_min + offset, y_max +
offset], [x_max + offset, y_max + offset]), width)
    pg.draw.lines(self.instance, color, True, ([offset, y_min + offset],
[offset, y_max + offset]), width)

```

Результаты выполнения программы

Комплект 1: Задания

ManySquares:



К отчёту прилагаются два файла: practice4.py и модуль graphics.py, внутри которого описывается класс Drawer.