МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
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**

Основная профессиональная образовательная программа  
Направление подготовки 09.03.01 Информатика и вычислительная техника  
Направленность (профиль) «Технологии разработки программного обеспечения»  
форма обучения – очная

**Практическая работа 4**по дисциплине «Математические основы компьютерной графики»

Обучающегося 3 курса.

Собинина Егора Яковлевича

Санкт-Петербург

2022

Написать программу, аналогичную заданной в задаче 2.2 предыдущего занятия (преобразование координат квадрата) на языке Python. ВСЕ требования к общему внутреннему функционалу задачи, к условям выполнения задачи такие же как в задаче 2.2 от предыдущего занятия.

Программа реализована с помощью подключения собственного пакета.

Структура:

Изображение выглядит как текст

Автоматически созданное описание

main.py:

# импортируем пакет

import pack

if \_\_name\_\_ == "\_\_main\_\_":

    pack.main\_draw()

graphics.py:

import pygame, sys

from pygame.locals import \*

import numpy as np

from . import transform as tr

class Drawer:

    def \_\_init\_\_(self):

        self.X = np.array([

            [2, 2],

            [-2, 2],

            [-2, -2],

            [2, -2]

            ]) \* 100

        self.trans = self.X.T

        self.shifted = tr.shift(self.X)

        self.V = tr.rotate()

        self.M = tr.scale()

    def Color\_method(self):

        self.BLACK = (0, 0, 0)

    def Drawing\_method(self, window, iter):

        for i in range(1, iter + 1):

            X\_i\_transformed = self.V @ (self.M @ self.trans)

            X\_i\_shifted =  tr.shift(X\_i\_transformed)

            pygame.draw.polygon(window, (i\*10, i\*10, i\*10), X\_i\_shifted.T, width=3)

            global center

            center = (i\*10, i\*10, i\*10)

            self.trans = X\_i\_transformed

            self.Color\_method()

            pygame.draw.polygon(window, self.BLACK, self.shifted, width=2)

            pygame.draw.circle(window, center, [400, 400], radius=6)

transform.py:

import numpy as np

def shift(X):

    X\_shifted = X + 400

    return X\_shifted

def scale():

    m\_scale = 0.9

    M = np.array([

        [m\_scale, 0],

        [0, m\_scale]

    ])

    return M

def rotate():

    alpha = -1 \* np.pi / 32

    V = np.array([

        [np.cos(alpha), -(np.sin(alpha))],

        [np.sin(alpha), np.cos(alpha)]

    ])

    return V

\_\_init\_\_.py:

import sys

import pygame, sys

from pygame.locals import \*

from . import graphics as gr

def main\_draw():

    pygame.init()

    pygame.font.init()

    window = pygame.display.set\_mode((800, 800))

    window.fill((255, 255, 255))

    iter = 20

    base = gr.Drawer()

    print(base.\_\_dict\_\_)

    base.Drawing\_method(window, iter)

    FPS = 30

    clock = pygame.time.Clock()

    while True:

        for event in pygame.event.get():

            if event.type == QUIT:

                pygame.quit()

                sys.exit()

            clock.tick(FPS)

            pygame.display.update()