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 3**по дисциплине «Математические основы компьютерной графики»

Обучающегося 3 курса.

Собинина Егора Яковлевича

Санкт-Петербург

2022

1.1 Напишите программу, которая строит квадрат, затем масштабирует его с коэффициентом m = 0.9 с помощью матрицы масштабирования, а потом вращает этот уменьшенный квадрат на угол α = π/32. Начальные координаты квадрата

Матрица масштабирования:

Матрица вращения:

Комбинированное преобразование:

Затем рассчитайте середину начального отрезка и с помощью матрицы преобразования найдите середину нового отрезка. Прорисуйте на экране с помощью библиотеки pygame 20 таких комбинированных преобразований. Подберите знак α так, чтобы вращение на экране было против часовой стрелки.

Ответ:

import sys

import numpy as np

import pprint as pprn

import pygame, sys

from pygame.locals import \*

# инициализируем

pygame.init()

pygame.font.init()

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

window.fill((255, 255, 255))

shift = 400

# начальные координаты

X = np.array([

    [2, 2],

    [-2, 2],

    [-2, -2],

    [2, -2]

]) \* 100

print(f"Массив координат отрезков X = ")

pprn.pprint(X)

X\_shifted = X + shift

# скалирование

m\_scale = 0.9

M = np.array([

    [m\_scale, 0],

    [0, m\_scale]

])

# углы поворота

alpha = -1 \* np.pi / 32

V = np.array([

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

    [np.sin(alpha), np.cos(alpha)]

])

# алгоритм отрисовки

iter = 20

X\_i = X.T

for i in range(1, iter + 1):

    X\_i\_transformed = V @ (M @ X\_i)

    X\_i\_shifted = X\_i\_transformed + shift

    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)

    X\_i = X\_i\_transformed

# отрисовка начального полигона и круга в центре

pygame.draw.polygon(window, (0, 0, 0), X\_shifted, width=2)

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

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()

1.2 Нарисуйте на экране последовательность соединённых между собой линий в виде улитки Паскаля, используя полярные координаты (r, θ) на плоскости. Формулы для полярного радиуса r и декартовых координат x и y:

Подберите параметры a, b и шаг по полярному углу θ. Используйте pygame.draw.lines().

Ответ:

import sys

import numpy as np

import pprint as pprn

import pygame

from pygame.locals import \*

#рассчетные формулы

#a = 1

#b = 1

#r = b + 2 \* a \* cos(phi)

#x = r \* cos(phi)

#y = r \* sin(phi)

#x\_end = cos(phi) \* (1 + 2 \* cos(phi))

#y\_end = cos(phi) \* (1 + 2 \* sin(phi))

#scale - изменение маштаба

#shift - перенос вправо-вниз

scale = 100

shift = 400

#инициализация pygame и окна

pygame.init()

pygame.font.init()

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

window.fill((255, 255, 255))

coord\_shift = 400

#создание 255 значений phi в промежутке [-2\*pi/3, 2\*pi/3]

phi = np.linspace(-2\*np.pi/3, 2\*np.pi/3, 255)

for i in range(len(phi)):

    if i < (len(phi)-1):

        #для сокращения количества хранимаой информации будем создавать элемент np.array "налету"

        start = np.array([np.cos(phi[i]) \* (1 + 2 \* np.cos(phi[i])), np.sin(phi[i]) \* (1 + 2 \* np.cos(phi[i]))])

        finish = np.array([np.cos(phi[i+1]) \* (1 + 2 \* np.cos(phi[i+1])), np.sin(phi[i+1]) \* (1 + 2 \* np.cos(phi[i+1]))])

        start, finish = (start \* scale) + shift, (finish \* scale) + shift

        pygame.draw.line(window, (155, i, i), start, finish, 3)

#условие "незакрывания" окна

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()