



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 №3

По дисциплине: Математические основы компьютерной
графики

Выполнил студент 3 курса группы 1.1

Щербинин А. В.

Санкт-Петербург
2022

Постановка задачи:

1. Комплект 1:

- 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 $\alpha = \pi/32$. Начальные координаты квадрата

$$X = \begin{pmatrix} 2 & 2 \\ -2 & 2 \\ -2 & -2 \\ 2 & -2 \end{pmatrix}^T \times 100 = \begin{pmatrix} 2 & -2 & -2 & 2 \\ 2 & 2 & -2 & -2 \end{pmatrix} \times 100.$$

Матрица масштабирования:

$$M = \begin{pmatrix} m & 0 \\ 0 & m \end{pmatrix}$$

Матрица вращения:

$$V = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

Комбинированное преобразование:

$$X^{transformed} = V \cdot M \cdot X$$

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.

- 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 :

$$r = b + 2 \cdot a \cdot \cos(\theta)$$

$$x = r \cdot \cos(\theta)$$

$$y = r \cdot \sin(\theta)$$

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

Комплект 1: Вращения

Код программы

```
import numpy as np
import pygame as pg
import math
from pprint import pprint as pp
import sys
from pygame.locals import *
```

```
# Переменные цвета
WHITE = (255, 255, 255)
BLACK = (0, 0, 0)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)

# Инициализируем игровую библиотеку, игровое окно и один из шрифтов для
надписей в окне
pg.init()
pg.font.init()
window = pg.display.set_mode((0, 0), FULLSCREEN)
window.fill((255, 255, 255))
my_game_font = pg.font.SysFont('Fira Code', 40)
pg.display.update()

def main():
    diag = 500
    menu(diag)
    FPS = 30
    clock = pg.time.Clock()
    while True:
        pressed_keys = pg.key.get_pressed()
        for event in pg.event.get():
            if (event.type == QUIT) or (pressed_keys[K_ESCAPE]):
                pg.quit()
                sys.exit()
            if pressed_keys[K_1]:
                window.fill(WHITE)
                lab3_1(diag)
            if pressed_keys[K_2]:
                window.fill(WHITE)
                lab3_2(diag)
            if pressed_keys[K_0]:
                window.fill(WHITE)
                menu(diag)
            if pressed_keys[K_UP]:
                diag += 50
                window.fill(WHITE)
                menu(diag)
            if pressed_keys[K_DOWN]:
                diag -= 50
                window.fill(WHITE)
                menu(diag)
        clock.tick(FPS)
        pg.display.update()
```

```

# "Главное меню"
def menu(d):
    window.blit(my_game_font.render("'1-2' - выбор задания", True, BLACK),
                (5, 5))
    window.blit(my_game_font.render("'0' - возвращение в меню", True, BLACK),
                (5, 45))
    window.blit(my_game_font.render("'UP', 'DOWN' - изменить смещение по осям
x, y", True, BLACK),
                (5, 85))
    window.blit(
        my_game_font.render("'Escape' - выход из программы", True, BLACK),
        (5, 125))
    window.blit(
        my_game_font.render(f"Оси x и y смещаются на {d}", True, BLACK),
        (5, 165))
    pg.display.update()

# Самостоятельная работа 1
def lab3_1(d):
    m = 0.9
    n = 20
    a = math.pi / 32
    X = np.array([[2, -2, -2, 2], [2, 2, -2, -2]]) * 100 + d
    M = np.array([[m, 0], [0, m]])
    V = ([math.cos(a), -math.sin(a)], [-math.sin(a), math.cos(a)])
    print("X:")
    mat_out(X)
    print("M:")
    mat_out(M)
    print("V:")
    mat_out(V)
    pg.draw.lines(window, BLUE, True, X.transpose(), 3)
    Xtransform = np.copy(X)
    for i in range(1, n + 1):
        Xtransform = V @ M @ Xtransform
        mat_out(Xtransform)
        pg.draw.lines(window, RED, True, Xtransform.transpose(), 3)
    pg.display.update()

# Самостоятельная работа 2
def lab3_2(d):
    a = 210
    b = 180
    h = 1
    O = 0
    r = b + 2 * a * math.cos(O)
    x = r * math.cos(O)

```

```

y = r * math.sin(0)
snail = np.array([[x, y]])
0 += h
while 0 <= 360:
    r = b + 2 * a * math.cos(0)
    x = r * math.cos(0)
    y = r * math.sin(0)
    snail = np.append(snail, [[x, y]], axis = 0)
    0 += h
print(snail)
pg.draw.lines(window, GREEN, False, snail + d, 3)
pg.display.update()

# Функция вывода матрицы
def mat_out(n):
    print("-----")
    pp(n)
    print("-----\n")

main()

```

Результаты выполнения программы

Меню программы. Есть возможность выбрать задание, нажав на соответствующую цифру, вернуться в меню, менять значение смещения по двум осям (по умолчанию 500), выйти из программы:

'1-2' - выбор задания

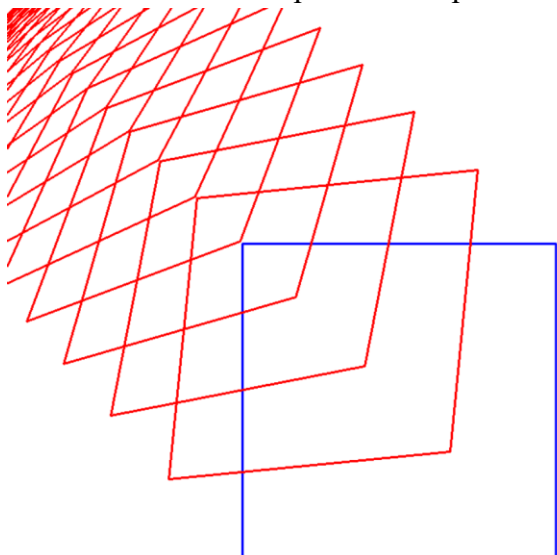
'0' - возвращение в меню

'UP', 'DOWN' - изменить смещение по осям x, y

'Escape' - выход из программы

Оси x и y смещаются на 500

1.1 Масштабирование и вращение квадрата:



1.2 Последовательность соединённых между собой линий в виде улитки Паскаля:

