

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчёт по лабораторной работе №7

Специальность ПО

Выполнил

В. С. Юрашевич,

студент группы ПО-11

Проверил

И. Д. Козик

Брест 2025

Цель работы: освоить возможности языка программирования Python в разработке оконных приложений.

Ход работы

Задание 1: Построение графических примитивов и надписей. Изобразить прямоугольник, вращающийся в плоскости формы вокруг одной из своих вершин.

Код программы

```
import pygame
import pygame_gui
import math
import os
import datetime

pygame.init()

pygame.display.set_caption("Управление вращением прямоугольника")

window_size = (800, 600)

screen = pygame.display.set_mode(window_size)

manager = pygame_gui.UIManager(window_size)

clock = pygame.time.Clock()

WHITE = (255, 255, 255)

RED = (255, 0, 0)

rect_width = 100

rect_height = 50

x, y = 400, 300

angle = 0

rotation_speed = 1

pivot = "bottom_left"

is_paused = False

take_screenshot = False

rotation_speed_label = pygame_gui.elements.UILabel(
    relative_rect=pygame.Rect(20, 0, 200, 20),
```

```

        text="Скорость вращения",
        manager=manager
    )
rotation_speed_slider = pygame_gui.elements.UIHorizontalSlider(
    relative_rect=pygame.Rect(20, 20, 200, 20),
    start_value=rotation_speed,
    value_range=(0, 10),
    manager=manager
)
rect_width_label = pygame_gui.elements.UILabel(
    relative_rect=pygame.Rect(20, 40, 200, 20),
    text="Ширина прямоугольника",
    manager=manager
)
rect_size_slider_width = pygame_gui.elements.UIHorizontalSlider(
    relative_rect=pygame.Rect(20, 60, 200, 20),
    start_value=rect_width,
    value_range=(50, 200),
    manager=manager
)
rect_height_label = pygame_gui.elements.UILabel(
    relative_rect=pygame.Rect(20, 80, 200, 20),
    text="Высота прямоугольника",
    manager=manager
)
rect_size_slider_height = pygame_gui.elements.UIHorizontalSlider(
    relative_rect=pygame.Rect(20, 100, 200, 20),
    start_value=rect_height,
    value_range=(25, 100),
    manager=manager
)

```

```

pivot_selector = pygame_gui.elements.UISelectionList(
    relative_rect=pygame.Rect(20, 140, 200, 120),
    item_list=["top_left", "top_right", "bottom_left", "bottom_right"],
    manager=manager
)

pause_button = pygame_gui.elements.UIButton(
    relative_rect=pygame.Rect(20, 270, 90, 40),
    text="Пауза",
    manager=manager
)

screenshot_button = pygame_gui.elements.UIButton(
    relative_rect=pygame.Rect(130, 270, 90, 40),
    text="Скриншот",
    manager=manager
)

def draw_rotating_rect(surface, x, y, width, height, angle, pivot):
    half_width = width // 2
    half_height = height // 2
    points = [
        (-half_width, -half_height),
        (half_width, -half_height),
        (half_width, half_height),
        (-half_width, half_height)
    ]
    if pivot == "top_left":
        pivot_x, pivot_y = -half_width, -half_height
    elif pivot == "top_right":
        pivot_x, pivot_y = half_width, -half_height
    elif pivot == "bottom_left":
        pivot_x, pivot_y = -half_width, half_height
    elif pivot == "bottom_right":

```

```

        pivot_x, pivot_y = half_width, half_height
    else:
        pivot_x, pivot_y = -half_width, half_height
    rotated_points = []
    for point in points:
        px = point[0] - pivot_x
        py = point[1] - pivot_y
        rx = px * math.cos(math.radians(angle)) - py * math.sin(math.radians(angle))
        ry = px * math.sin(math.radians(angle)) + py * math.cos(math.radians(angle))
        rotated_points.append((rx + x + pivot_x, ry + y + pivot_y))
    pygame.draw.polygon(surface, RED, rotated_points)

running = True
while running:
    time_delta = clock.tick(60) / 1000.0
    screen.fill(WHITE)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

        if event.type == pygame.USEREVENT:
            if event.user_type == pygame_gui.UI_BUTTON_PRESSED:
                if event.ui_element == pause_button:
                    is_paused = not is_paused
                    pause_button.set_text("Продолжить" if is_paused else "Пауза")
                elif event.ui_element == screenshot_button:
                    take_screenshot = True

            manager.process_events(event)

    manager.update(time_delta)

    rotation_speed = rotation_speed_slider.get_current_value()
    rect_width = int(rect_size_slider_width.get_current_value())
    rect_height = int(rect_size_slider_height.get_current_value())
    selection = pivot_selector.get_single_selection()

```

```

if selection:
    pivot = selection

if not is_paused:
    angle += rotation_speed

    if angle >= 360:
        angle -= 360

draw_rotating_rect(screen, x, y, rect_width, rect_height, angle, pivot)

manager.draw_ui(screen)

pygame.display.flip()

if take_screenshot:
    filename =
f"screenshot_{datetime.datetime.now().strftime('%Y%m%d_%H%M%S')}.png"

    pygame.image.save(screen, os.path.join(os.getcwd(), filename))

    print(f"Скриншот сохранён: {filename}")

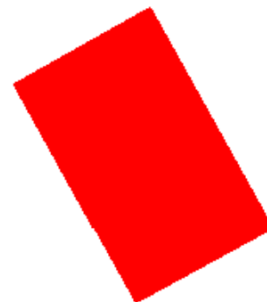
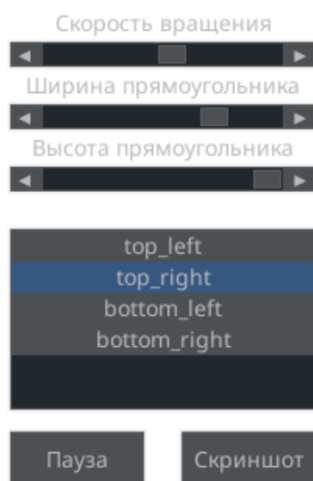
    take_screenshot = False

pygame.quit()

```

Рисунок с результатами работ

🤖 Управление вращением прямоугольника



Задание 2: Реализовать построение заданного типа фрактала по варианту. Везде, где это необходимо, предусмотреть ввод параметров, влияющих на внешний вид фрактала. Множество Жюлиа.

Код программы

```
import pygame

import pygame_gui

import numpy as np

pygame.init()

pygame.display.set_caption("Множество Жюлиа")

width, height = 800, 600

screen = pygame.display.set_mode((width, height))

manager = pygame_gui.UIManager((width, height))

clock = pygame.time.Clock()

WHITE = (255, 255, 255)

c_real = -0.7

c_imag = 0.27015

max_iter = 300

input_c_real = pygame_gui.elements.UITextEntryLine(
    relative_rect=pygame.Rect((20, 20), (100, 30)),
    manager=manager
)

input_c_real.set_text(str(c_real))

input_c_imag = pygame_gui.elements.UITextEntryLine(
    relative_rect=pygame.Rect((20, 60), (100, 30)),
    manager=manager
)

input_c_imag.set_text(str(c_imag))

input_max_iter = pygame_gui.elements.UITextEntryLine(
    relative_rect=pygame.Rect((20, 100), (100, 30)),
    manager=manager
```

```

)

input_max_iter.set_text(str(max_iter))

relabel = pygame_gui.elements.UILabel(
    relative_rect=pygame.Rect((130, 20), (100, 30)),
    text="Действ. часть c",
    manager=manager
)

imlabel = pygame_gui.elements.UILabel(
    relative_rect=pygame.Rect((130, 60), (100, 30)),
    text="Мним. часть c",
    manager=manager
)

iterlabel = pygame_gui.elements.UILabel(
    relative_rect=pygame.Rect((130, 100), (100, 30)),
    text="Макс. итераций",
    manager=manager
)

generate_button = pygame_gui.elements.UIButton(
    relative_rect=pygame.Rect((20, 140), (210, 30)),
    text='Построить фрактал',
    manager=manager
)

def julia_set(width, height, c_real, c_imag, max_iter):
    image = np.zeros((height, width))
    for x in range(width):
        for y in range(height):
            zx = 1.5 * (x - width / 2) / (0.5 * width)
            zy = (y - height / 2) / (0.5 * height)
            z = complex(zx, zy)
            c = complex(c_real, c_imag)

```



```

        iteration = 0

        while abs(z) <= 2 and iteration < max_iter:

            z = z * z + c

            iteration += 1

        image[y, x] = iteration

    return image

def draw_julia_set(image):

    surface = pygame.Surface((width, height))

    for x in range(width):

        for y in range(height):

            color_value = image[y, x]

            color = (color_value % 256, color_value % 256, color_value % 256)

            surface.set_at((x, y), color)

    return surface

image_surface = draw_julia_set(julia_set(width, height, c_real, c_imag, max_iter))

running = True

while running:

    time_delta = clock.tick(60) / 1000.0

    for event in pygame.event.get():

        if event.type == pygame.QUIT:

            running = False

        if event.type == pygame_gui.UI_BUTTON_PRESSED:

            if event.ui_element == generate_button:

                try:

                    c_real = float(input_c_real.get_text())

                    c_imag = float(input_c_imag.get_text())

                    max_iter = int(input_max_iter.get_text())

                    image = julia_set(width, height, c_real, c_imag, max_iter)

                    image_surface = draw_julia_set(image)

                except ValueError:

                    print("Ошибка: Введите корректные числовые значения.")

```

```
manager.process_events(event)

manager.update(time_delta)

screen.fill(WHITE)

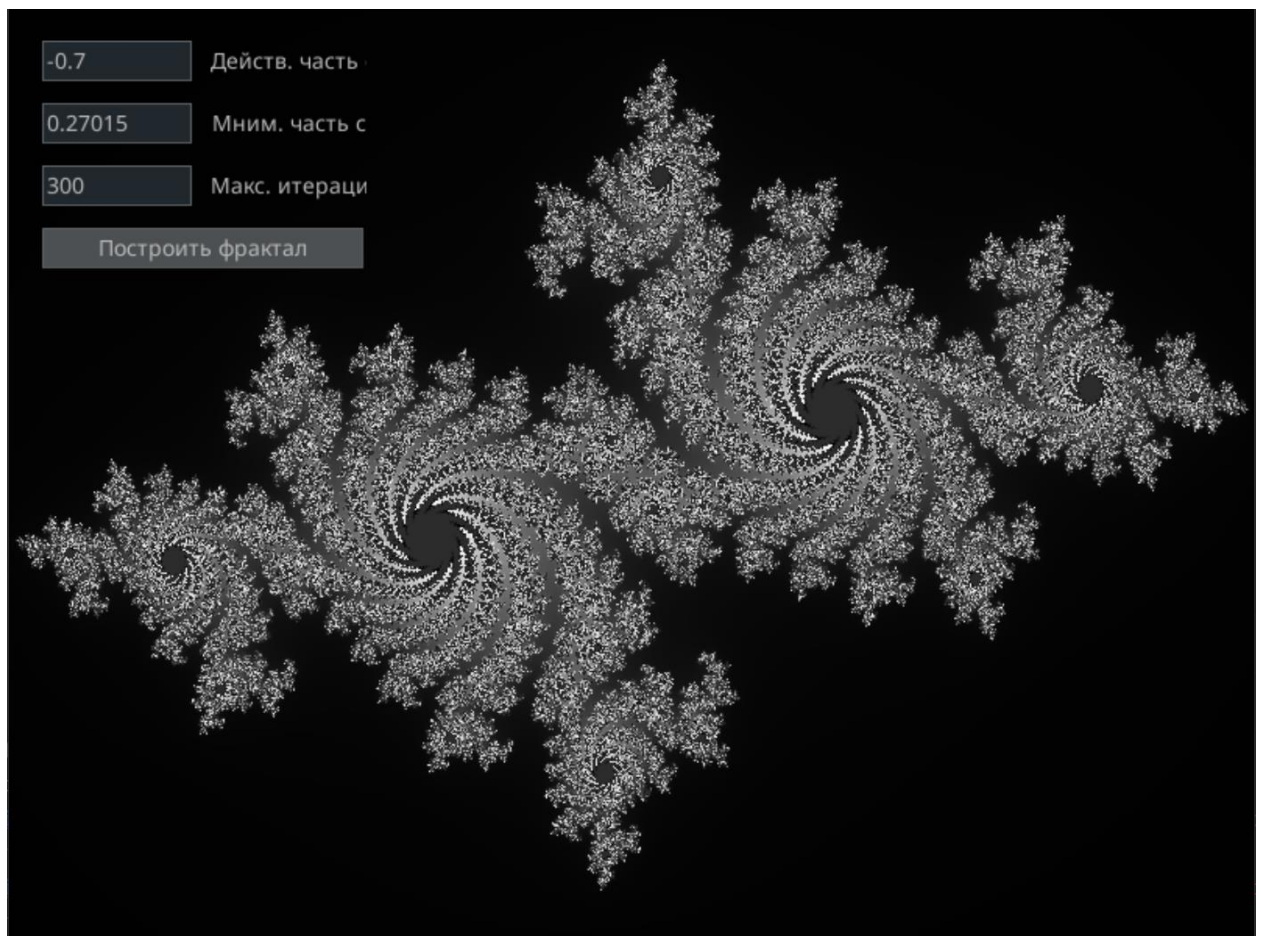
screen.blit(image_surface, (0, 0))

manager.draw_ui(screen)

pygame.display.flip()

pygame.quit()
```

Рисунок с результатами работ



Вывод: освоил возможности языка программирования Python в разработке оконных приложений.